

STATA CAUSAL INFERENCE AND TREATMENT-EFFECTS ESTIMATION REFERENCE MANUAL RELEASE 19



A Stata Press Publication
StataCorp LLC
College Station, Texas



® Copyright © 1985–2025 StataCorp LLC
All rights reserved
Version 19

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845

ISBN-10: 1-59718-420-9

ISBN-13: 978-1-59718-420-5

This manual is protected by copyright. All rights are reserved. No part of this manual may be reproduced, stored in a retrieval system, or transcribed, in any form or by any means—electronic, mechanical, photocopy, recording, or otherwise—without the prior written permission of StataCorp LLC unless permitted subject to the terms and conditions of a license granted to you by StataCorp LLC to use the software and documentation. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document.

StataCorp provides this manual “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. StataCorp may make improvements and/or changes in the product(s) and the program(s) described in this manual at any time and without notice.

The software described in this manual is furnished under a license agreement or nondisclosure agreement. The software may be copied only in accordance with the terms of the agreement. It is against the law to copy the software onto DVD, CD, disk, diskette, tape, or any other medium for any purpose other than backup or archival purposes.

The automobile dataset appearing on the accompanying media is Copyright © 1979 by Consumers Union of U.S., Inc., Yonkers, NY 10703-1057 and is reproduced by permission from CONSUMER REPORTS, April 1979.

Stata, **STATA**, Stata Press, Mata, **MATA**, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC.

Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations.

StataNow is a trademark of StataCorp LLC.

Other brand and product names are registered trademarks or trademarks of their respective companies.

For copyright information about the software, type `help copyright` within Stata.

The suggested citation for this software is

StataCorp. 2025. *Stata 19*. Statistical software. StataCorp LLC.

The suggested citation for this manual is

StataCorp. 2025. *Stata 19 Causal Inference and Treatment-Effects Estimation Reference Manual*. College Station, TX: Stata Press.

Contents

Intro	Introduction to causal inference and treatment-effects estimation	1
Causal inference commands	Introduction to causal inference commands	16
cate	Conditional average treatment-effects estimation	24
cate postestimation	Postestimation tools for cate	86
DID intro	Introduction to difference-in-differences estimation	104
didregress	Difference-in-differences estimation ⁺	119
didregress postestimation	Postestimation tools for didregress and xtdidregress	157
eteffects	Endogenous treatment-effects estimation	175
eteffects postestimation	Postestimation tools for eteffects	187
etpoisson	Poisson regression with endogenous treatment effects	192
etpoisson postestimation	Postestimation tools for etpoisson	206
etregress	Linear regression with endogenous treatment effects	210
etregress postestimation	Postestimation tools for etregress	238
gencohort	Create a cohort variable	243
hdidregress	Heterogeneous difference in differences ⁺	245
hdidregress postestimation	Postestimation tools for hdidregress and xthdidregress ⁺	267
latebalance	Check balance after lateeffects estimation ⁺	283
latebalance density	Covariate balance density ⁺	287
latebalance overid	Test for covariate balance ⁺	290
latebalance summarize	Covariate-balance summary statistics ⁺	292
lateffects	Local average treatment-effect estimation ⁺	294
lateffects postestimation	Postestimation tools for lateffects ⁺	307
lateoverlap	Overlap plots ⁺	310
mediate intro	Introduction to causal mediation analysis	314
mediate	Causal mediation analysis (one mediator)	329
mediate postestimation	Postestimation tools for mediate	369
mediate multiple	Causal mediation analysis (two mediators) ⁺	381
mediate multiple postestimation	Postestimation tools for mediate ⁺	425
stteffects	Treatment-effects estimation for observational survival-time data	432
stteffects intro	Introduction to treatment effects for observational survival-time data	433
stteffects ipw	Survival-time inverse-probability weighting	453
stteffects ipwra	Survival-time inverse-probability-weighted regression adjustment	463
stteffects postestimation	Postestimation tools for stteffects	493
stteffects ra	Survival-time regression adjustment	503
stteffects wra	Survival-time weighted regression adjustment	514
tebalance	Check balance after teffects or stteffects estimation	522
tebalance box	Covariate balance box	535
tebalance density	Covariate balance density	540
tebalance overid	Test for covariate balance	545
tebalance summarize	Covariate-balance summary statistics	550
teffects	Treatment-effects estimation for observational data	556

teffects intro	Introduction to treatment effects for observational data	557
teffects intro advanced	Advanced introduction to treatment effects for observational data	570
teffects aipw	Augmented inverse-probability weighting	584
teffects ipw	Inverse-probability weighting	608
teffects ipwra	Inverse-probability-weighted regression adjustment	617
teffects multivalued	Multivalued treatment effects	627
teffects nnmatch	Nearest-neighbor matching	637
teffects postestimation	Postestimation tools for teffects	653
teffects psmatch	Propensity-score matching	661
teffects ra	Regression adjustment	670
telasso	Treatment-effects estimation using lasso	682
telasso postestimation	Postestimation tools for telasso	704
teoverlap	Overlap plots	707
xthdidregress	Heterogeneous difference in differences for panel data ⁺	714
xthdidregress postestimation	Postestimation tools for xthdidregress ⁺	736
xtswitchdid	Difference in differences with switching treatments for panel data ⁺	737
xtswitchdid postestimation	Postestimation tools for xtswitchdid ⁺	766
Glossary		773
Subject and author index		781

Cross-referencing the documentation

When reading this manual, you will find references to other Stata manuals, for example, [\[U\] 27 Overview of Stata estimation commands](#); [\[R\] regress](#); and [\[D\] reshape](#). The first example is a reference to chapter 27, *Overview of Stata estimation commands*, in the *User's Guide*; the second is a reference to the `regress` entry in the *Base Reference Manual*; and the third is a reference to the `reshape` entry in the *Data Management Reference Manual*.

All the manuals in the Stata Documentation have a shorthand notation:

[GSM]	<i>Getting Started with Stata for Mac</i>
[GSU]	<i>Getting Started with Stata for Unix</i>
[GSW]	<i>Getting Started with Stata for Windows</i>
[U]	<i>Stata User's Guide</i>
[R]	<i>Stata Base Reference Manual</i>
[ADAPT]	<i>Stata Adaptive Designs: Group Sequential Trials Reference Manual</i>
[BAYES]	<i>Stata Bayesian Analysis Reference Manual</i>
[BMA]	<i>Stata Bayesian Model Averaging Reference Manual</i>
[CAUSAL]	<i>Stata Causal Inference and Treatment-Effects Estimation Reference Manual</i>
[CM]	<i>Stata Choice Models Reference Manual</i>
[D]	<i>Stata Data Management Reference Manual</i>
[DSGE]	<i>Stata Dynamic Stochastic General Equilibrium Models Reference Manual</i>
[ERM]	<i>Stata Extended Regression Models Reference Manual</i>
[FIN]	<i>Stata Financial Statistics Reference Manual</i>
[FMM]	<i>Stata Finite Mixture Models Reference Manual</i>
[FN]	<i>Stata Functions Reference Manual</i>
[G]	<i>Stata Graphics Reference Manual</i>
[H2OML]	<i>Machine Learning in Stata Using H2O: Ensemble Decision Trees Reference Manual</i>
[IRT]	<i>Stata Item Response Theory Reference Manual</i>
[LASSO]	<i>Stata Lasso Reference Manual</i>
[XT]	<i>Stata Longitudinal-Data/Panel-Data Reference Manual</i>
[META]	<i>Stata Meta-Analysis Reference Manual</i>
[ME]	<i>Stata Multilevel Mixed-Effects Reference Manual</i>
[MI]	<i>Stata Multiple-Imputation Reference Manual</i>
[MV]	<i>Stata Multivariate Statistics Reference Manual</i>
[PSS]	<i>Stata Power, Precision, and Sample-Size Reference Manual</i>
[P]	<i>Stata Programming Reference Manual</i>
[RPT]	<i>Stata Reporting Reference Manual</i>
[SP]	<i>Stata Spatial Autoregressive Models Reference Manual</i>
[SEM]	<i>Stata Structural Equation Modeling Reference Manual</i>
[SVY]	<i>Stata Survey Data Reference Manual</i>
[ST]	<i>Stata Survival Analysis Reference Manual</i>
[TABLES]	<i>Stata Customizable Tables and Collected Results Reference Manual</i>
[TS]	<i>Stata Time-Series Reference Manual</i>
[I]	<i>Stata Index</i>
[M]	<i>Mata Reference Manual</i>

Description

This entry provides an introduction to causal inference and treatment-effects estimation. It presents concepts, frameworks, and assumptions that researchers consider when they wish to draw causal inferences in their analyses.

For information on Stata commands that estimate treatment effects and that are specifically designed for causal inference, see [\[CAUSAL\]](#) **Causal inference commands**.

For more in-depth introductions to causal inference, see [Imbens and Rubin \(2015\)](#), [Robins and Greenland \(1992\)](#), [Hernán and Robins \(2020\)](#), and [Pearl \(2009\)](#).

Remarks and examples

Remarks are presented under the following headings:

Motivation: Causation versus association
Causal inference workflow
Potential-outcomes framework
Treatment-effect estimands
Assumptions required in potential-outcomes framework
Relaxing causal assumptions
Causal diagrams
Importance of identification before estimation

Motivation: Causation versus association

Research may be driven by the desire to evaluate causation or association. Causal questions explore changes in the outcome when we change a variable under our control or examine what would happen to the outcome if a variable of interest had not changed. For example:

- Does receiving a treatment cure the illness?
- What would have happened to the inflation rate if the Federal Reserve had not increased interest rates?
- Does smoking reduce fetal growth?
- Does raising the minimum wage decrease unemployment?

In contrast, associative questions observe patterns in data. For example:

- How does the cure rate between patients who received treatment and those who did not receive treatment compare?
- Is there a correlation between interest rates and inflation?
- Is there a difference in the mean birthweight of infants born to mothers who smoke versus those born to mothers who do not smoke?
- What is the difference in the unemployment rate between states that have implemented an increase in minimum wage and those that have not?

The first set of questions asks what happens when there is an intervention or imagine a scenario where a variable changes versus does not change. The second set of questions observe only the pattern without intervention.

To examine some of the considerations we must face when performing causal inference, we consider a hypothetical study, conducted by a software company called Statanium, that examines the relationship between the number of breaks taken by software developers and their productivity. The company wants to find out whether the number of breaks impacts productivity. As a user of Statanium and an expert in causality, you are hired to advise the company whether it should encourage developers to take additional breaks during their workday. The question the company is interested in is causal because it aims to compare the productivity when the number of breaks is increased versus the productivity when the number of breaks (and other possible factors) remains unchanged. If the two outcomes of productivity are different, then the action of increasing the number of breaks has a causal effect. In causal inference literature, the action of increasing the number of breaks is referred to as a treatment or an intervention.

You could estimate an association between increased breaks and productivity via correlation or many other statistical methods that estimate dependence. However, the interest is in estimation of causal effect. The well-known expression “association is not causation” suggests that for any given amount of association, only some part or none of it is causal. Thus, a challenge in causal inference is to identify and eliminate relationships that are only associative. To perform causal inference in our case, we want to create a hypothetical scenario where the number of breaks is increased and all other factors that may influence productivity remain fixed. Then we can determine the causal effect of the additional break.

When it is possible, randomized experiments are a popular method to estimate causal effects because the treatment (number of breaks) is controlled by the experimenter. Therefore, the treatment assignment is guaranteed to be random and unrelated to all other factors that may determine the outcome. This can be represented using causal diagrams, as in [figure 1](#), where T represents the treatment, Y represents the productivity or outcome, and X represents all other factors. The absence of an arrow between X and T indicates that X does not affect T .

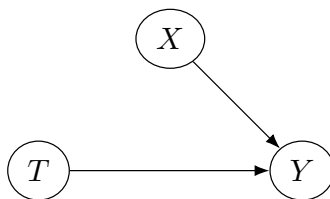


Figure 1.

However, randomized experiments are not always feasible, and causal effects need to be inferred from observational data. In the observational data case, the experimenter does not have control over the treatment assignment, and the assumption that all other factors are held constant in both the observed and hypothetical worlds may not hold. This is due to the presence of confounding factors (for example, job satisfaction) that affect both the treatment and outcome. This is reflected in the causal diagram in [figure 2](#) by arrows from X to both T and Y .

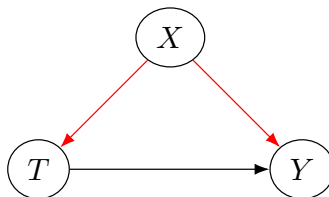


Figure 2.

Here the measured total association between T and Y contains both causal path $T \rightarrow Y$ and association path $T \leftarrow X \rightarrow Y$, highlighted in red. To identify the causal effect and make further causal inference, you need to eliminate the association represented by the red path by accounting or adjusting for other confounding factors.

The following highlights the significance of adjusting for confounding variables. Suppose that in [figure 3](#) you plot the productivity of software developers as a function of the average number of breaks.

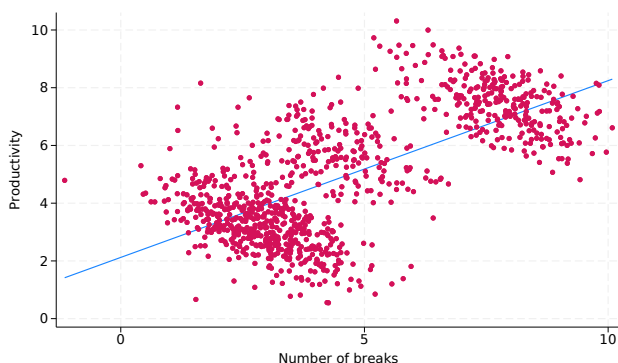


Figure 3.

The plot shows a positive association between the number of breaks and productivity, implying that taking more breaks leads to higher productivity. The questions of interest are whether this association can be considered causal and whether Statanium should motivate its hardworking developers to take more breaks during the workday. The conclusion that more breaks are beneficial would be valid if the model assumed is as shown in [figure 1](#). However, as an expert, you believe that there are confounding factors, such as workload or job satisfaction, that need to be considered, as shown in [figure 2](#). To account for the effect of job satisfaction, in [figure 4](#), you plot the productivity of software developers against the average number of breaks they take, considering different levels of job satisfaction—dissatisfied (orange), satisfied (red), and highly satisfied (yellow).

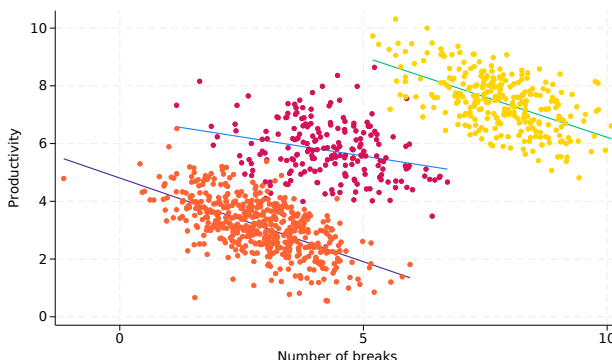


Figure 4.

The plot shows that when job satisfaction is accounted for, the effect of the number of breaks on productivity changes from positive to negative. This phenomenon is known as Simpson's paradox (Blyth 1972), where the overall effect appears to be positive but, when it is adjusted for a confounder, the effect sign changes direction.

The above example illustrates the fundamental difference between causation and association. For example, if for some reason the researcher does not account for the job satisfaction or it is not observed, then the estimated association cannot be interpreted as causal.

Below, we introduce the workflow, popular causal inference frameworks, and assumptions that allow researchers to draw causal inferences rather than merely find associations.

Causal inference workflow

The causal inference literature recommends the following three-phase workflow (Pearl 2009; Imbens 2020; and Heckman and Pinto 2022) when a research question is causal in nature:

1. Hypothetical modeling: Researchers make assumptions about relationships among variables based on their understanding and expertise. These assumptions are related not only to the treatment variable and the outcome of interest but also to any variables that might be related to the treatment or the outcome. The assumptions regarding these relationships cannot be tested from data; therefore, the validity of these assumptions must come from previous theory or the researcher's own expertise.
2. Causal effect identification: Based on the assumptions made in the first phase, the researcher tries to determine whether the causal effect can be identified.
3. Parameter estimation: If the answer to the second phase is positive, the researcher can then use various estimation techniques, such as those provided by the commands discussed in [CAUSAL] **Causal inference commands**, to estimate the causal effect.

Potential-outcomes framework

The potential-outcomes framework is one of the most commonly used theories for understanding and evaluating causal inference. The foundation of this framework is motivated by the idea of a natural experiment (Imbens 2020), which focuses on finding settings and identification strategies under which the assignment of a treatment can be considered as good as random even though the data are observational rather than from a randomized experiment. The roots of the potential-outcomes framework trace back to

a seminal paper by [Splawa-Neyman \(1923\)](#), and the framework was formally introduced in [Rubin \(1974\)](#). [Fisher \(1925\)](#) built upon Neyman’s ideas and introduced the idea of physical randomization, which formally defines the concept of a treatment-assignment mechanism. For further reading, see [Imbens and Rubin \(2015\)](#).

As discussed earlier, the goal of causal inference is to estimate the change in an outcome as the treatment varies. In the Statanium example, we now assume that the treatment (number of breaks) and the outcome (productivity) are binary. That is, $T_i = 1$ means the Statanium developer takes additional breaks, and $T_i = 0$ otherwise. Similarly, $Y_i = 1$ if a developer’s productivity increases, and $Y_i = 0$ otherwise. Thus, we are interested in estimating the change in productivity if the developer takes additional breaks, $T_i = 1$, versus if the developer does not take additional breaks, $T_i = 0$. However, for a given developer, we can observe only one outcome. This is known as the fundamental problem of causal inference ([Holland 1986](#)). The potential-outcomes framework provides tools and assumptions to solve this problem.

The important concepts in potential outcomes are the unit, treatment, and outcome. A unit is the research object to which treatment is assigned. It can be, for example, a person, a company, a school, or a county. Treatment is the action that we apply to a unit. In this entry, treatment is denoted as T . For binary treatment, the units to which treatment is applied ($T = 1$) are called the treated group, and the units to which the treatment is not applied ($T = 0$) are called the control group. The potential outcome of treatment with value t for unit i is denoted by $Y_i(T_i = t)$ or $Y_i(t)$. In our running example, the unit is the developer, treatment is taking additional breaks, and the outcome is whether productivity improves.

The observed outcome is related to the potential outcome through $Y_i = T_i Y_i(1) + (1 - T_i) Y_i(0)$. That is, if the unit receives the treatment $T_i = 1$, then $Y_i = Y_i(1)$ and $Y_i(0)$ otherwise.

A potential outcome that is not observed is called a counterfactual outcome. For example, for the developer who took additional breaks, $Y_i(1)$ is the observed outcome and $Y_i(0)$ is the counterfactual outcome.

Treatment-effect estimands

Individual treatment effect (ITE). For each developer or unit i , the ITE is defined as

$$Y_i(T_i = 1) - Y_i(T_i = 0)$$

Because only one of the potential outcomes is observed, we cannot identify this quantity directly from the data without making assumptions about the unobserved counterfactuals and the assignment to treatment. However, estimates of individual effects can be useful for providing insight into how a treatment may affect an individual. For instance, in epidemiology, estimated ITEs could help determine whether a treatment is likely to be helpful for a particular patient.

We can also define and estimate effects that allow us to draw interesting causal inferences for the population instead of for each individual.

Average treatment effect (ATE). The ATE is also known as the average causal effect. The ATE at the population level can be defined as the mean difference in potential outcomes when units received a treatment versus when units did not receive any treatment,

$$\text{ATE} = E[Y(1) - Y(0)] = E[Y(1)] - E[Y(0)] \quad (1)$$

where $Y(t) = Y(T = t)$, for $t = \{0, 1\}$. We say that the ATE of treatment T on outcome Y exists if $E[Y(1)] \neq E[Y(0)]$.

The ATE provides an estimate of the expected average effect in the population and can therefore be used to answer many interesting questions that could help policymakers make decisions. In our Statanium example, the ATE is the expected difference in productivity if all developers took extra breaks versus none taking extra breaks. Statanium could determine whether extra breaks are beneficial and set break policies based on the estimated ATE. Similarly, we interpret the ATE for a couple of our motivating causal questions. When investigating whether a treatment cures an illness, we could interpret the ATE as the expected difference in the proportion of individuals who were cured from an illness when everyone received the treatment versus when no one received the treatment. When evaluating a raise in minimum wage, we could interpret the ATE as the expected change in unemployment rate when the minimum wage is raised for everyone versus when the minimum wage stays the same.

When the ATE is of interest, we must use an appropriate method that leads to an estimate of this quantity. We might be tempted to use association difference to estimate ATE, where association difference is the conditional mean difference of outcomes between treatment and not treated units. The treatment T and Y are associated if $E[Y|T = 1] \neq E[Y|T = 0]$. Thus, one might try to estimate the causal quantity in (1) with the statistical quantity $E[Y|T = 1] - E[Y|T = 0]$. For the Statanium example, this quantity could be estimated from the data by contrasting the sample average of the developers that took the treatment with the ones that did not. Mathematically,

$$\frac{\sum_{i=1}^N Y_i T_i}{\sum_{i=1}^N T_i} - \frac{\sum_{i=1}^N Y_i (1 - T_i)}{\sum_{i=1}^N (1 - T_i)}$$

However, in general, ATE and association difference are different; otherwise, association would be causation. Figure 5 highlights the causation–association difference.

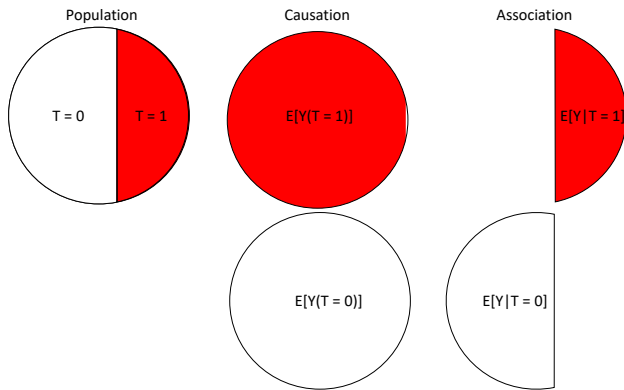


Figure 5.

To infer causation, we imagine that each treatment or intervention is applied to the entire population, and the difference between the red and white circles is observed in the same population. In contrast, to infer association, we condition on $T = t$ and estimate the difference between subsets of populations.

Average treatment effect on the treated (ATET). For the treated group, the treatment-effect estimand is the ATET:

$$\text{ATET} = E[Y(1)|T = 1] - E[Y(0)|T = 1]$$

The ATET is useful when researchers are interested in the effect on those who received the treatment. This effect may be of particular interest when the goal is to understand how a treatment performs for the subpopulation at which the treatment was targeted. The effect is a comparison with what would have happened in this subpopulation if they had not received the treatment. In our Statanium example, we estimate the expected difference in productivity for developers who took extra breaks compared with the productivity of these developers if they had not taken extra breaks. When we investigate whether a treatment cures an illness, the ATET is focused only on those who received the treatment. For this group, what is the expected difference in the proportion of individuals who were cured when given the treatment versus the proportion cured if they had not received the treatment? When evaluating a raise in minimum wage, we might be interested in the effect for states that enacted a minimum wage increase. The ATET is the expected difference in unemployment rate for these states compared with a situation where these states did not raise minimum wage.

Additional estimands. In specific situations, there are several other treatment-effect estimands that may be of interest and that can be defined in the potential-outcomes framework. We briefly mention a few here.

Sometimes, it is assumed that the effect of T on Y may involve both a direct effect and an indirect effect such that T has an effect on another variable M , known as a mediator, and that M in turn has an effect on Y . In the Statanium example, we might believe that increased breaks could improve the developers' focus while working and that improved focus leads to increased productivity. In such a situation, comparisons of average direct treatment effects and average indirect treatment effects may be of interest. See [Robins and Greenland \(1992\)](#), [VanderWeele \(2015\)](#), and [Pearl and MacKenzie \(2018\)](#) for discussions of causal mediation analysis and definitions of applicable direct and indirect effect estimands.

Recently, there has been a surge of interest in estimating the treatment effect when it differs between subgroups, also known as the heterogeneous treatment effect ([Athey and Imbens 2016](#); [Künzel et al. 2019](#); and [Nie and Wager 2021](#)). At the subgroup level, the treatment-effect estimand is called conditional average treatment effect (CATE),

$$\text{CATE} = E[Y(1)|X = x] - E[Y(0)|X = x]$$

where $Y(t)|X = x$ for $t = \{0, 1\}$ are the potential outcomes of the subgroup $X = x$.

Assumptions required in potential-outcomes framework

At this point, it is natural to ask under which conditions treatment effects can be estimated from observational data. For illustration purposes, our focus will be on ATE. We are interested in assumptions for which

$$\text{ATE} = E[Y(1)] - E[Y(0)] = E[Y|T = 1] - E[Y|T = 0]$$

A causal quantity, that is, ATE, is “identifiable” if it can be computed from a statistical quantity $E[Y|T = t]$. By statistical quantity, we mean an object that can be estimated from data.

In the potential-outcome framework, commonly used assumptions are the stable unit treatment value assumption (SUTVA), unconfoundedness assumption, and overlap assumption.

SUTVA. The SUTVA, along with consistency, states that for a given unit, the treatment of other units does not affect the outcome of the treatment received by that unit. Consequently, there are two different sources in which SUTVA could be violated. The first source is a violation of the consistency condition, which might not hold in some studies (Cole and Frangakis 2009; and Schwartz, Gatto, and Campbell 2011). Usually, the problem arises from the vagueness of the assigned treatment. For example, for the Statanium example, if the treatment is additional breaks, this may be one additional break or three additional breaks. If we observe only whether a treatment has been assigned, the counterfactual $Y(T = t)$ is not well defined because different numbers of breaks have different causal effects. The second source of violation arises if some units are influenced by the assignment of the treatment of other units. For example, if some developers in the control group noticed that the developers in the treatment group are less productive, they might change their lifestyle and start taking fewer breaks, which can lead to an increase in productivity. Typically, interference can occur because of spillover effects or noncompliance or because the units are members of a social network. For details, see Hernán and Robins (2020, chap. 3).

Unconfoundedness assumption. The unconfoundedness assumption goes by many names, including the conditional-independence assumption, the ignorability assumption, and the exchangeability assumption. The impetus behind this assumption is to make the treatment and control group comparable within strata defined by X . It states that the probability of a positive outcome in the control group (white group in the figure above) would be the same as the probability of a positive outcome in the treatment group (red group) had units in the control group received the treatment given to those in the treatment group. In other words, under unconfoundedness, if by accident the treatment were given to the white group instead of the red group, then the ATE would remain the same. Mathematically, this is represented by $(Y(1), Y(0)) \perp\!\!\!\perp T|X$, where $\perp\!\!\!\perp$ denotes (conditional) independence and X are potential confounders. It is important to differentiate between $Y(t) \perp\!\!\!\perp T|X$, which utilizes potential outcomes, and $Y \perp\!\!\!\perp T|X$. The unconfoundedness assumption does not imply that $Y \perp\!\!\!\perp T|X$. On the contrary, if the ATE is not zero, then Y and T are associated. In the context of the Statanium example, the underlying intuition is that for two developers i and j , their potential outcomes should be independent of the treatment assignment $P(Y_i(0), Y_i(1)|T = t_i, X) = P(Y_j(0), Y_j(1)|T = t_j, X)$, for $t_i, t_j \in \{0, 1\}$.

Overlap assumption. Finally, the overlap or positivity assumption $P(T = t|X = x) > 0$ implies that the treatment assignment should be stochastic. For example, if the developers who are highly satisfied are always assigned treatment $T = 1$, then there is no meaning in studying the treatment $T = 0$. In contrast to unconfoundedness, the overlap assumption can sometimes be verified from the data. For details, see Hernán and Robins (2020, chap. 12).

When these three main assumptions in the potential-outcome framework are satisfied, then the ATE is identified, and estimation methods available in the `teffects` suite of commands can be used.

Relaxing causal assumptions

One of the crucial questions in causal inference is whether all confounders have been accounted for in the study. Unfortunately, the unconfoundedness is not testable from data. There are observable and unobservable confounders that, unaccounted for, will lead to incorrect conclusions. This problem of unobserved confounders or endogeneity is usually addressed using estimators that account for an endogenous treatment or instrumental-variable method.

Discussions of these issues can be found in Imbens and Rubin (2015). For a general treatment, see Wooldridge (2010) and Angrist and Pischke (2009).

The potential-outcome framework is not restricted only to models in which identification relies on conditional unconfoundedness, which is sometimes referred to as selection on observables. It also allows us to recover the ATET by controlling for individual and time-varying unobservables, without the need to control for covariates. One example of this is the difference-in-differences method, which can be used to estimate ATET by comparing the change in outcomes between the treatment and control groups over time.

Causal diagrams

In *Motivation: Causation versus association*, we used causal diagrams, specifically directed acyclic graphs (DAGs), to represent our assumptions about causal relationships. DAGs are helpful in phases 1 and 2 described in *Causal inference workflow*. The intellectual predecessor of causal diagrams, or more generally, the structural causal models (SCMs) framework, goes back to the pioneering work of geneticist Wright (1921, 1934) and econometricians Frisch and Waugh (1933) and Haavelmo (1944) and the references therein. Early econometricians were attempting to conceptualize the fact that, unlike correlation, the regression of a variable Y on T has a natural direction and is different from the regression of T on Y . This distinction led to the development of causal frameworks by Pearl (2009) and Heckman and Pinto (2022). Furthermore, this differentiation is linked to the concept known as “the ladder of causality” (Pearl and MacKenzie 2018), where the lowest rung signifies association and higher rungs address causal queries. Here our focus is on causal diagrams. We provide only a brief introduction and explore the usefulness of causal diagrams in determining causal-effect identification before estimating treatment effects using one of Stata’s estimation commands. We will mention some of the common terminology used in SCM, but for more detailed descriptions of these terms and additional details on SCM, we refer you to Pearl (2009), Peters, Janzing, and Schölkopf (2017), Bareinboim et al. (2022), and the references therein.

DAGs are characterized by nodes and directed edges between the nodes (shown as circles and arrows, respectively, in diagrams below), which represent causal relationships. The absence of edges between nodes indicates that there is no direct causal effect between those nodes. Unlike path diagrams in SEM, independent unobserved error terms are not depicted in DAGs.

For the Statanium example, we capture our assumptions using the DAG in the left panel of figure 6.



Figure 6. (left) Unconfoundedness and (right) Instrumental variable

Here X is a confounder in the causal relationship for T and Y , and because X is observed, we can control or adjust for it by including it as a covariate in our model when we estimate the causal effect. With this adjustment, we eliminate associative paths and identify the causal relationship between T and Y . In the causal diagram literature, the path $T \leftarrow X \rightarrow Y$ is known as a backdoor path because it contains an arrow that goes to the back of node T (hence, the name “backdoor”). If we do not condition on X , we say that we are leaving the path open, and we cannot estimate the causal effect. In this situation, we meet what is known as the backdoor path criterion because, by adjusting for X , we close all paths entering T .

In comparison, in the DAG shown on the right of figure 6, X is an unobserved confounder, which is indicated by shading the node for this variable. In this case, we cannot condition on X by including it as a covariate when we estimate the causal effect. Thus, the backdoor path criterion is not satisfied, and the causal effect of T on Y cannot be identified in this way. Fortunately, in this case, the assumed DAG

still allows us to estimate the causal effect because Z can be used as an instrumental variable. A proper instrumental variable can be included as a predictor of the treatment to eliminate unobserved sources of confounding such as that from X here. Z can be used as a proper instrumental variable because there is no direct effect of Z on the outcome Y and there are no unobserved confounders between Z and X or Z and Y .

For our simple DAGs, we were able to evaluate identification easily by considering the relationships among the few variables. For more complex situations, there exist several algorithms and criteria that are designed to check and achieve causal effect identification using a graphical approach. We do not discuss those algorithms here; we instead refer the interested reader to [Shpitser and Pearl \(2008\)](#), [Pearl \(2009\)](#), [Maathuis and Colombo \(2015\)](#), [Perković et al. \(2015\)](#), and [van der Zander, Liśkiewicz, and Textor \(2019\)](#). These algorithms can select variables that form a valid adjustment set that can be used for treatment-effect estimation. Once the valid adjustment set is selected, researchers can use their preferred treatment-effect estimation method from the list of commands in [\[CAUSAL\] Causal inference commands](#).

Below, we demonstrate how a DAG can be used in the phases discussed in the [Causal inference workflow](#). Consider a study that examines the effect on five-year mortality of polycystic kidney disease (PKD) among patients undergoing peritoneal dialysis (PD). Recall that in phase 1 of the workflow, we construct a hypothetical model that represents researchers’ assumptions. We begin our causal analysis by drawing a DAG that represents our assumptions about the causal effects we are interested in estimating and the relationship among all variables, observed and unobserved, that may affect the variables of interest. Here we borrow the hypothetical model from [Evans et al. \(2012\)](#), illustrated in [figure 6](#).

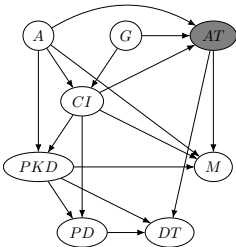


Figure 6.

The goal of phase 2 is to check whether the causal effect of the treatment variable PKD on mortality M is identifiable. For the variable definitions, see [table 2](#). The gray node in the graph indicates that the variable AT is not observed.

Table 2. Variable definitions

PKD - polycystic kidney disease; treatment variable
M - mortality; outcome variable
A - patient age
G - patient gender
C - patient’s comorbidities summary index
PD - indicator for patients undergoing peritoneal dialysis
AT - type of medical assistance
DT - type of peritoneal dialysis

Note that if all variables in the causal graph are observed, then the causal effect can be identified using backdoor path criteria (Pearl 2009).

However, in our example, AT is not observed, and there are unblocked noncausal (backdoor) paths that hinder identification:

P1: $PKD \leftarrow CI \rightarrow AT \rightarrow M$

P2: $PKD \leftarrow A \rightarrow AT \rightarrow M$

P3: $PKD \leftarrow CI \leftarrow G \rightarrow AT \rightarrow M$

Note that, despite a common misconception, conditioning on all observed variables does not mitigate the identification issue. For example, consider the path $PKD \rightarrow DT \leftarrow AT$. This type of path is known as a v-structure, and DT is called a collider. PKD and AT are unrelated because the collider closes or blocks the path. However, we would not want to include DT in the model because conditioning on the collider DT introduces a selection bias through $PKD \rightarrow DT \leftarrow AT \rightarrow M$. For more examples, see [Importance of identification before estimation](#). Interestingly, the above noncausal paths P1, P2, and P3 can be blocked if we condition on variables CI and A. In fact, CI and A are the necessary minimal sufficient set that makes the causal effect of PKD to M identifiable. Now that we know that the causal effect is identified, we are ready to move to phase 3, estimation of the treatment effect. In this case, CI and A can be used as covariates in, for instance, the inverse-probability weighting estimator. For details, see [\[CAUSAL\] teffects ipw](#).

```
. teffects ipw (M) (PKD CI A)
(output omitted)
```

Importance of identification before estimation

In this section, we provide numerical examples that reemphasize the importance of the identification step. If a causal effect is not identified, the results of any treatment-effect estimation method are unreliable. We illustrate this point with two simple examples, which also aim to clear up the common misconception that it is harmless to control for more variables in treatment-effect estimation. For more examples, the reader is referred to Cinelli, Forney, and Pearl (2024) and Hünernmund, Louw, and Caspi (2021).

► Example 1

The presumed relationship among variables in [figure 7](#) is well known for introducing collider bias, sometimes called selection bias, in the estimated effect of T on Y . In the causal diagram literature, the represented DAG is known as a v-structure, and the variable X as a collider. A v-structure $T \rightarrow X \leftarrow Y$ has an interesting characteristic that the causal effect of T on Y is 0 because the collider X blocks the path. If we adjust for X , we introduce a selection bias because the adjustment results in an induced association between T and Y .

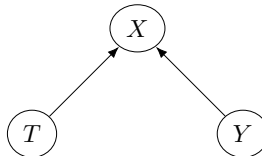


Figure 7.

Suppose the data come from the data-generation process as

$$\begin{aligned}T &:= \epsilon_T \\Y &:= \epsilon_Y \\X &:= 2 \times T - 0.5 \times Y + \epsilon_X\end{aligned}$$

where $\epsilon_T, \epsilon_Y, \epsilon_X \sim N(0, 1)$. The regression of Y on T and X shows that T and X are for prediction purposes if we were to determine this on the basis of their p -values. However, the X is a bad control in terms of answering a causal question about the effect of T on Y .

```
. regress Y T X
```

Source	SS	df	MS	Number of obs	=	1,000
Model	187.350778	2	93.675389	F(2, 997)	=	111.51
Residual	837.504999	997	.840025074	Prob > F	=	0.0000
				R-squared	=	0.1828
Total	1024.85578	999	1.02588166	Adj R-squared	=	0.1812
				Root MSE	=	.91653

	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
T	.8088171	.0607918	13.30	0.000	.6895225	.9281117
X	-.3942266	.0263994	-14.93	0.000	-.4460314	-.3424219
_cons	.0345212	.0289833	1.19	0.234	-.0223541	.0913965

Fortunately, the true causal effect 0 is recovered if we do not control for X .

```
. regress Y T
```

Source	SS	df	MS	Number of obs	=	1,000
Model	.025504326	1	.025504326	F(1, 998)	=	0.02
Residual	1024.83027	998	1.02688404	Prob > F	=	0.8748
				R-squared	=	0.0000
Total	1024.85578	999	1.02588166	Adj R-squared	=	-0.0010
				Root MSE	=	1.0134

	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
T	.0049212	.0312266	0.16	0.875	-.0563562	.0661986
_cons	.0347662	.0320452	1.08	0.278	-.0281175	.0976498

◀

► Example 2

Here we demonstrate another type of bias that can be induced in the estimated effect of T on Y .

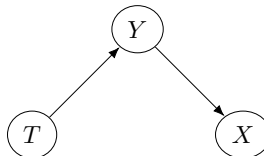


Figure 8.

Compared with the previous [example](#), the harmfulness for controlling X is not explicit. Recall that in the previous [section](#), we mentioned that the nodes for unobserved error terms (ϵ_T, ϵ_Y) are omitted from the causal graph, but they are there. If we imagine that the error term for Y is there, then Y would be a

collider because $T \rightarrow Y \leftarrow \epsilon_Y$. We also see $Y \rightarrow X$, so we can say that X is a descendant of the collider Y . It turns out that by controlling on the descendant X of collider Y , we induce an association between T and ϵ_Y (Pearl 2009, sec. 11.3). This conditioning opens a backdoor path between T and Y , which makes the causal effect of T on Y unidentified.

Suppose the data arise from the following data-generation process:

$$\begin{aligned}T &:= \epsilon_T \\ Y &:= 2 \times T + \epsilon_Y \\ X &:= -0.5 \times Y + \epsilon_X\end{aligned}$$

Then the causal effect of T on Y is 2. Similar to the previous example, even though X is good for prediction purposes, it is a bad control for causality.

```
. regress Y T X
```

Source	SS	df	MS	Number of obs	=	1,000
Model	4420.49975	2	2210.24987	F(2, 997)	=	2631.17
Residual	837.504998	997	.840025073	Prob > F	=	0.0000
				R-squared	=	0.8407
				Adj R-squared	=	0.8404
Total	5258.00474	999	5.26326801	Root MSE	=	.91653

Y	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
T	1.626137	.0379613	42.84	0.000	1.551644	1.700631
X	-.3942266	.0263994	-14.93	0.000	-.4460314	-.3424219
_cons	.0345212	.0289833	1.19	0.234	-.0223541	.0913965

Again, the true causal effect is recovered if we do not control for X .

```
. regress Y T
```

Source	SS	df	MS	Number of obs	=	1,000
Model	4233.17448	1	4233.17448	F(1, 998)	=	4122.35
Residual	1024.83027	998	1.02688404	Prob > F	=	0.0000
				R-squared	=	0.8051
				Adj R-squared	=	0.8049
Total	5258.00474	999	5.26326801	Root MSE	=	1.0134

Y	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
T	2.004921	.0312266	64.21	0.000	1.943644	2.066199
_cons	.0347662	.0320452	1.08	0.278	-.0281175	.0976498

These examples demonstrate that, before estimating a treatment effect, we must determine whether and how the effect can be identified based on our assumed relationships among variables. Once we have evaluated identification, we can select an appropriate method for estimating the causal effect. In these examples, we used linear regression. Actually, many of Stata's regression commands can be used for estimating treatment effects, provided that identification assumptions hold. Stata also offers estimation commands that are specifically designed for estimating treatment effects in various situations. For more information on these specialized commands, see [CAUSAL] [Causal inference commands](#).

References

- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Athey, S., and G. W. Imbens. 2016. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences* 113: 7353–7360. <https://doi.org/10.1073/pnas.1510489113>.
- Bareinboim, E., J. D. Correa, D. Ibeling, and T. Icard. 2022. “On Pearl’s hierarchy and the foundations of causal inference”. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, edited by H. Geffner, R. Dechter, and J. Y. Halpern, vol. 36: 507–556. New York: ACM Books. <https://doi.org/10.1145/3501714.3501743>.
- Blyth, C. R. 1972. On Simpson’s paradox and the sure-thing principle. *Journal of the American Statistical Association* 67: 364–366. <https://doi.org/10.2307/2284382>.
- Cinelli, C., A. Forney, and J. Pearl. 2024. A crash course in good and bad controls. *Sociological Methods and Research* 53: 1071–1104. <https://doi.org/10.1177/00491241221099552>.
- Cole, S. R., and C. E. Frangakis. 2009. Commentary: The consistency statement in causal inference: A definition or an assumption? *Epidemiology* 20: 3–5.
- Evans, D., B. Chaix, T. Lobbedez, C. Verger, and A. Flahault. 2012. Combining directed acyclic graphs and the change-in-estimate procedure as a novel approach to adjustment-variable selection in epidemiology. *BMC Medical Research Methodology* 12: art. 156. <https://doi.org/10.1186/1471-2288-12-156>.
- Fisher, R. A. 1925. *Statistical Methods for Research Workers*. Edinburgh: Oliver and Boyd.
- Frisch, R., and F. V. Waugh. 1933. Partial time regressions as compared with individual trends. *Econometrica* 1: 387–401. <https://doi.org/10.2307/1907330>.
- Haavelmo, T. 1944. The probability approach in econometrics. *Econometrica* 12 (Suppl.): 1–115. <https://doi.org/10.2307/1906935>.
- Heckman, J. J., and R. Pinto. 2022. Causality and econometrics. Working Paper 29787, National Bureau of Economic Research. <https://doi.org/10.3386/w29787>.
- Hernán, M. A., and J. M. Robins. 2020. *Causal Inference: What If*. Boca Raton, FL: CRC Press.
- Holland, P. W. 1986. Statistics and causal inference. *Journal of the American Statistical Association* 81: 945–960. <https://doi.org/10.2307/2289064>.
- Hünernmund, P., B. Louw, and I. Caspi. 2021. Double machine learning and automated confounder selection—a cautionary tale. arXiv:2108.11294 [econ.EM], <https://doi.org/10.48550/arXiv.2108.11294>.
- Imbens, G. W. 2020. Potential outcome and directed acyclic graph approaches to causality: Relevance for empirical practice in economics. *Journal of Economic Literature* 58: 1129–1179. <https://doi.org/10.1257/jel.20191597>.
- Imbens, G. W., and D. B. Rubin. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. New York: Cambridge University Press. <https://doi.org/10.1017/CBO9781139025751>.
- Künzel, S. R., J. S. Sekhon, P. J. Bickel, and B. Yu. 2019. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences* 116: 4156–4165. <https://doi.org/10.1073/pnas.1804597116>.
- Maathuis, M. H., and D. Colombo. 2015. A generalized back-door criterion. *Annals of Statistics* 43: 1060–1088. <https://doi.org/10.1214/14-AOS1295>.
- Nie, X., and S. Wager. 2021. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika* 108: 299–319. <https://doi.org/10.1093/biomet/asaa076>.
- Pearl, J. 2009. *Causality: Models, Reasoning, and Inference*. 2nd ed. Cambridge: Cambridge University Press. <https://doi.org/10.1017/CBO9780511803161>.
- Pearl, J., and D. MacKenzie. 2018. *The Book of Why: The New Science of Cause and Effect*. New York: Basic Books.
- Perković, E., J. Textor, M. Kalisch, and M. H. Maathuis. 2015. “A complete generalized adjustment criterion”. In *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*, 682–691. Amsterdam: Association for Uncertainty in Artificial Intelligence. <https://auai.org/uai2015/proceedings/papers/155.pdf>.
- Peters, J., D. Janzing, and B. Schölkopf. 2017. *Elements of Causal Inference: Foundations and Learning Algorithms*. Cambridge, MA: MIT Press.

- Robins, J. M., and S. Greenland. 1992. Identifiability and exchangeability for direct and indirect effects. *Epidemiology* 3: 143–155. <https://doi.org/10.1097/00001648-199203000-00013>.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- Schwartz, S., N. M. Gatto, and U. B. Campbell. 2011. “What would have been is not what would be: Counterfactuals of the past and potential outcomes of the future”. In *Causality and Psychopathology: Finding the Determinants of Disorders and Their Cures*, edited by P. E. Shrout, K. M. Keyes, and K. Ornstein, 25–46. Oxford: Oxford University Press. <https://doi.org/10.1093/oso/9780199754649.003.0006>.
- Shpitser, I., and J. Pearl. 2008. Complete identification methods for the causal hierarchy. *Journal of Machine Learning Research* 9: 1941–1979.
- Splawa-Neyman, J. 1923. On the application of probability theory to agricultural experiments. Essay on principles. *Polish Agricultural Forestry Journal* 10: 29–42.
- van der Zander, B., M. Liśkiewicz, and J. Textor. 2019. Separators and adjustment sets in causal graphs: Complete criteria and an algorithmic framework. *Artificial Intelligence* 270: 1–40. <https://doi.org/10.1016/j.artint.2018.12.006>.
- VanderWeele, T. J. 2015. *Explanation in Causal Inference: Methods for Mediation and Interaction*. New York: Oxford University Press.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.
- Wright, S. 1921. Correlation and causation. *Journal of Agricultural Research* 20: 557–585.
- . 1934. The method of path coefficients. *Annals of Mathematical Statistics* 5: 161–215.

Description

In this entry, we give you an overview of the estimation commands in Stata that are designed for causal inference. We provide important details about each command so that readers can select the one that best fits their data and research needs.

Here we assume that you are familiar with causal inference and the most common assumptions. For an introduction to these concepts, see [\[CAUSAL\] Intro](#).

Remarks and examples

Below, we introduce Stata commands that are specifically designed for causal inference. For each command, we provide information on the type of data required and the necessary assumptions. In addition, we outline the type of statistics that can be estimated—typically one or more of the average treatment effect (ATE), the average treatment effect on the treated (ATET), or the potential-outcome means (POM). We also indicate the type of outcome variable (continuous, binary, count, fractional, or nonnegative) and the type of treatment (binary, multivalued, or continuous) that each command supports. Finally, we note which models must be specified: a model for the outcome, a model for the treatment, both, or none.

Remarks are presented under the following headings:

- [teffects](#)
- [stteffects](#)
- [telasso](#)
- [cate](#)
- [lateffects](#)
- [Difference in differences](#)
- [Endogenous treatment](#)
- [Causal mediation](#)
- [Extended regression models](#)
- [margins](#)

teffects

The `teffects` suite of commands is useful for estimating treatment effects from cross-sectional data. These commands rely on the stable unit treatment value assumption (SUTVA), unconfoundedness (conditional-independence) assumption, and overlap assumption.

The commands in the `teffects` suite and the type of estimator provided by each are as follows:

<code>teffects ra</code>	Regression adjustment
<code>teffects ipw</code>	Inverse-probability weighting
<code>teffects ipwra</code>	Inverse-probability-weighted regression adjustment
<code>teffects aipw</code>	Augmented inverse-probability weighting
<code>teffects nnmatch</code>	Nearest-neighbor matching
<code>teffects psmatch</code>	Propensity-score matching

Details on the available estimands, types of outcomes and treatments supported, and the models to be specified are given below:

Command	Estimand	Outcome types	Treatment types	Models specified
<code>teffects ra</code>	ATE ATET POM	continuous binary count fractional nonnegative	binary multivalued	outcome
<code>teffects ipw</code>	ATE ATET POM	continuous binary count fractional nonnegative	binary multivalued	treatment
<code>teffects ipwra</code>	ATE ATET POM	continuous binary count fractional nonnegative	binary multivalued	outcome treatment
<code>teffects aipw</code>	ATE ATET POM	continuous binary count fractional nonnegative	binary multivalued	outcome treatment
<code>teffects psmatch</code>	ATE ATET	continuous binary count fractional nonnegative	binary	treatment
<code>teffects nnmatch</code>	ATE ATET	continuous binary count fractional nonnegative	binary	outcome*

*`nnmatch` includes covariates for modeling the outcome but does not require specification of a functional form for the outcome model.

For further information on these commands and the properties of the estimators that they implement, see [\[CAUSAL\] teffects intro](#).

stteffects

The `stteffects` suite of commands is useful for estimating treatment effects from survival-time data. These commands rely on the SUTVA, unconfoundedness (conditional-independence) assumption, and overlap assumption. They also rely on an assumption that the correct adjustment is made for censoring.

The commands in the `stteffects` suite and the type of estimator provided by each are as follows:

<code>stteffects ra</code>	Survival-time regression adjustment
<code>stteffects wra</code>	Survival-time weighted regression adjustment
<code>stteffects ipw</code>	Survival-time inverse-probability weighting
<code>stteffects ipwra</code>	Survival-time inverse-probability-weighted regression adjustment

Details on the available estimands, types and treatments supported, and the models to be specified are given below:

Command	Estimand	Treatment types	Models specified
<code>stteffects ra</code>	ATE	binary	outcome
	ATET	multivalued	
	POM		
<code>stteffects wra</code>	ATE	binary	outcome
	ATET	multivalued	censoring
	POM		
<code>stteffects ipw</code>	ATE	binary	treatment
	ATET	multivalued	censoring
	POM		
<code>stteffects ipwra</code>	ATE	binary	outcome
	ATET	multivalued	treatment
	POM		censoring (optional)

For further information on these commands and the properties of the estimators that they implement, see [\[CAUSAL\] stteffects intro](#).

telasso

The `telasso` command is useful for estimating treatment effects from cross-sectional data and using lasso to select from among many potential control variables to be included in the model. This estimator relies on the SUTVA, unconfoundedness (conditional-independence) assumption, and overlap assumption.

`telasso` allows a continuous, binary, count, or nonnegative outcome and requires a binary treatment variable. Models are specified for both the outcome and the treatment. The ATE, ATET, or POM may be requested.

For further information on this command and the properties of the estimator that it implements, see [\[CAUSAL\] telasso](#).

cate

The `cate` suite of commands is useful for estimating the average treatment effects conditional on a set of variables, known as conditional average treatment effects (CATEs). `cate` provides three different CATE estimates: individualized average treatment effects (IATEs), group average treatment effects (GATEs), and sorted group average treatment effects (GATESS). IATEs are treatment effects conditional on observation-level characteristics. There is one IATE for each observation in the data. GATEs are treatment effects conditional on prespecified groups. There is a treatment effect for each group. GATESSs are average treatment effects for a prespecified number of groups. The groups are determined by quantiles of individual-level treatment-effects values. Estimating CATEs allows us to study the treatment-effect heterogeneity and evaluate the treatment-assignment policy.

These commands rely on the SUTVA, unconfoundedness (conditional-independence) assumption, and overlap assumption.

`cate` allows a continuous outcome and requires a binary treatment variable. `cate` estimates a CATE function, an outcome model, and a treatment-assignment model. The CATE function is estimated by the partialing-out (PO) estimator or the augmented inverse-probability weighting (AIPW) estimator via random forest or parametric regression. The outcome and the treatment models can be estimated using cross-fitting via lasso, random forest, or parametric regression.

For further information on this command and the properties of the estimator that it implements, see [\[CAUSAL\] cate](#).

lateffects

The `lateffects` suite of commands is useful for estimating treatment effects from cross-sectional data when an instrument is available. Specifically, the instrumental variable may be the treatment assignment (in an experimental setting) or a variable that encourages, persuades, or otherwise motivates individuals to self-select into a certain treatment status (in an observational-data setting). The estimand of interest is the average treatment effect for compliers, also known as the local average treatment effect (LATE).

These commands rely on the SUTVA, the unconfoundedness (conditional-independence) assumption, and the overlap assumption. Note that, for a LATE, these assumptions are defined with respect to the propensity score of the instrument rather than the propensity score of the treatment indicator. Also, the instrument should affect the outcome only through treatment, an instrumental-variable exclusion restriction. Additionally, there must be no defiers, understood as individuals who would act contrary to their treatment assignment.

The commands in the `lateffects` suite and the type of estimator provided by each are as follows:

<code>lateffects ipwra</code>	Inverse-probability-weighted regression adjustment
<code>lateffects kappa</code>	Normalized kappa
<code>lateffects balancing</code>	Normalized covariate balancing

Details on the available estimands; the types of outcomes, treatments, and instruments supported; and the models to be specified are given below:

Command	Estimand	Outcome types	Treatment types	Instrument types	Models specified
lateffects ipwra	LATE	continuous binary count fractional	binary	binary	outcome treatment instrument
lateffects kappa	LATE	continuous	binary	binary	instrument
lateffects balancing	LATE	continuous	binary	binary	instrument

Difference in differences

The difference-in-differences suite of commands is useful for estimating treatment effects from data in which some of the units are observed both before and after a treatment and some units remain untreated. The difference-in-differences suite comprises the following commands:

didregress	Difference in differences
xtdidregress	Difference in differences for panel data
hdidregress	Heterogeneous difference in differences
xthdidregress	Heterogeneous difference in differences for panel data
gencohort	Create a cohort variable for heterogeneous difference in differences
xtswitchdid	Difference in differences with switching treatments for panel data

The `didregress` and `hdidregress` commands estimate treatment effects for repeated cross-sectional data, while `xtdidregress` and `xthdidregress` estimate treatment effects for panel data. The `didregress` and `xtdidregress` commands estimate a single ATET. The `hdidregress` and `xthdidregress` commands allow for heterogeneous treatment effects and report separate ATETs for each time and treatment cohort.

These estimators apply in settings where treatments are binary and absorbing. They rely on the SUTVA, unconfoundedness (conditional-independence) assumption, and overlap assumption. In addition, they rely on an assumption of parallel trends in the treatment and control groups.

`xtswitchdid` estimates event-study effects for panel data where treatments may be multivalued and switching (nonabsorbing). It relies on assumptions of parallel trends and no anticipation.

For further information on these commands and the properties of the estimators that they implement, see [\[CAUSAL\] DID intro](#).

Endogenous treatment

The `et` commands are useful for estimating treatment effects from cross-sectional data in cases where the unconfoundedness (conditional-independence) assumption is violated because treatment assignment is not independent of the potential outcomes. The `et` commands comprise the following:

<code>eteffects</code>	Endogenous treatment-effects estimation
<code>etpoisson</code>	Poisson regression with endogenous treatment effects
<code>etregress</code>	Linear regression with endogenous treatment effects

Details on the estimands, types of outcomes and treatments supported, and the models to be specified are given below:

Command	Estimand	Outcome types	Treatment types	Models specified
<code>eteffects</code>	ATE	continuous	binary	outcome treatment
	ATET	binary		
	POM	count		
		fractional nonnegative		
<code>etpoisson</code>	ATE	count	binary	outcome treatment
	ATET	nonnegative		
	POM			
<code>etregress</code>	ATE	continuous	binary	outcome treatment
	ATET			
	POM			

Note that `eteffects` provides the ATE, ATET, and POM directly. `etregress` estimates the ATE directly, while the ATET and POM can be obtained from `margins` after estimation. For `etpoisson`, ATE, ATET, and POM can all be obtained from `margins` after estimation.

For further information on these commands and the properties of the estimators that they implement, see [\[CAUSAL\] eteffects](#), [\[CAUSAL\] etpoisson](#), and [\[CAUSAL\] etregress](#).

Causal mediation

The `mediate` command is useful for estimating direct, indirect, and total treatment effects from cross-sectional data in some cases where the treatment may affect an outcome both directly and indirectly. An indirect effect is one in which the treatment affects another variable, called a mediator, and the mediator in turn affects the outcome.

The `mediate` command fits causal mediation models with one or two mediators. The single-mediator version allows both outcome and mediator variables to be continuous, binary, count, and nonnegative. The treatment may be binary, multivalued, or continuous. The two-mediator version fits parallel as well as sequential causal mediation models and provides a decomposition into all path-specific effects. Outcome and mediator variables may be continuous, whereas the treatment may be binary or continuous.

These estimators rely on the SUTVA, unconfoundedness (conditional-independence) assumption, and overlap assumption.

`mediate`, when used with a single mediator, provides estimates of the following statistics:

Estimand

natural indirect effect (NIE)
 natural direct effect (NDE)
 pure natural indirect effect (PNIE)
 total natural direct effect (TNDE)
 total causal effect (TE)

`mediate`, when used with two mediators, provides estimates of the following statistics:

Estimand

natural indirect effect through both mediators alone (NIE_1 and NIE_2)
 natural indirect effect through both (sequential) mediators (NIE_{12})
 mediator-specific natural indirect effect ($MS-NIE_1$ and $MS-NIE_2$)
 natural direct effect (NDE)
 total causal effect (TE)

For further information on this command and the properties of the estimators that it implements, see [\[CAUSAL\] `mediate`](#) and [\[CAUSAL\] `mediate multiple`](#).

Extended regression models

The extended regression model (ERM) suite of commands is designed to account for treatment (exogenous or endogenous), endogenous covariates, and nonrandom sample selection one at a time or in combination. Commands are available for both cross-sectional and panel data. The following commands are comprised in the ERM suite:

<code>eregress</code>	Extended linear regression
<code>eintreg</code>	Extended interval regression
<code>eprobit</code>	Extended probit regression
<code>eoprobit</code>	Extended ordered probit regression
<code>xteregress</code>	Extended linear regression for panel data
<code>xteintreg</code>	Extended interval regression for panel data
<code>xteprobit</code>	Extended probit regression for panel data
<code>xteoprobit</code>	Extended ordered probit regression for panel data

`eregress` and `xteregress` fit models for continuous outcomes. `eintreg` and `xteintreg` fit models for interval-censored outcomes. `eprobit` and `xteprobit` fit models for binary outcomes. `eoprobit` and `xteoprobit` fit models for ordinal outcomes. All commands allow binary and multivalued treatments.

After fitting a model that accounts for endogenous or exogenous treatment with one of the ERM commands, you can use `estat teffects` to estimate the ATE, ATET, or POM.

For further information on these commands and the properties of the estimators that they implement, see [\[ERM\] Intro 1](#).

Other commands in Stata provide some of the features found in the ERM commands. For instance, when you account only for endogenous covariates, `eregress` and `ivregress` provide equivalent parameter estimates. Instrumental-variable commands—`ivregress`, `ivprobit`, `ivpoisson`, and `ivtobit`—are designed to account for endogeneity (unobserved confounding) and provide consistent parameter estimates in this situation. Thus, these commands are used when the goal is causal inference. In some cases, a parameter estimated by these commands can be directly interpreted as the causal effect of interest, and in other cases, postestimation commands can be used to obtain the ATE, ATET, and POM.

margins

The `margins` command is available after many estimation commands in Stata. When a researcher has determined that appropriate assumptions have been satisfied for performing causal inference, many estimation commands can be used in combination with `margins` to estimate the ATE, ATET, and POM. As a simple example, you may type

```
. regress y c.x##i.trt, vce(robust)
```

to fit a linear regression of `y` on treatment `trt` and adjusted for covariate `x`. To estimate the POM, you could type

```
. margins trt, vce(unconditional)
```

The ATE is a contrast of the POM, and `margins` uses the `r.` operator to request such a contrast:

```
. margins r.trt, vce(unconditional)
```

The `margins` command can be used similarly after other estimation commands, and the results can be interpreted causally when proper assumptions for causal inference have been met.

For more information on `margins`, see [\[R\] margins](#).

Also see

[\[CAUSAL\] Intro](#) — Introduction to causal inference and treatment-effects estimation

[\[CAUSAL\] Glossary](#)

[Description](#)
[Options](#)
[References](#)

[Quick start](#)
[Remarks and examples](#)
[Also see](#)

[Menu](#)
[Stored results](#)

[Syntax](#)
[Methods and formulas](#)

Description

`cate` estimates conditional average treatment effects (CATES), which are average treatment effects (ATES) conditional on a set of variables for which the treatment effects may vary. Estimating CATES allows us to study treatment-effect heterogeneity and evaluate treatment-assignment policies.

`cate` provides three different CATE estimates: individualized average treatment effects (IATES), group average treatment effects (GATES), and sorted group average treatment effects (GATESS). IATES are treatment effects conditional on observation-level characteristics; there is one IATE for each observation in the data. GATES are treatment effects conditional on prespecified groups; there is a treatment effect for each group. GATESS are treatment effects for a prespecified number of groups, where the groups are determined by the quantiles of the IATES.

To estimate CATES, `cate` fits an outcome model and a treatment-assignment model. These models can be fit using cross-fitting via lasso, random forest, or parametric regression. The CATES themselves can be estimated using a partialing-out (PO) estimator or an augmented inverse-probability weighting (AIPW) estimator, either via random forest or linear regression.

Quick start

Estimate the IATE function for outcome `y` and treatment `treat`, conditioning on covariates `x1-x5` and `i.group1`, using the PO estimator, and report the ATE

```
cate po (y x1-x5 i.group1) (treat)
```

Same as above, but add variables `w1-w100` as control variables in the outcome and treatment models

```
cate po (y x1-x5 i.group1) (treat), controls(w1-w100)
```

Same as above, but use the AIPW estimator

```
cate aipw (y x1-x5 i.group1) (treat), controls(w1-w100)
```

Estimate the GATES for the groups defined by variable `group2`

```
cate aipw (y x1-x5 i.group1) (treat), controls(w1-w100) group(group2)
```

Same as above, but reestimate the GATES for groups defined by variable `group1` without refitting the IATE function

```
cate, reestimate group(group1)
```

Divide the data into five groups based on quintiles of the IATE estimates, and estimate the GATESS for those groups

```
cate aipw (y x1-x5 i.group1) (treat), controls(w1-w100) group(5)
```

Same as above, but divide the data into four groups (quartiles of the IATE estimates), and reestimate the GATESS for these groups

```
cate, reestimate group(4)
```

Perform cross-fitting with five folds instead of the default ten folds

```
cate aipw (y x1-x5 i.group1) (treat), controls(w1-w100) xfolds(5)
```

Perform random forest for the outcome and treatment models

```
cate aipw (y x1-x5 i.group1) (treat), controls(w1-w100)      ///
omethod(rforest) tmethod(rforest)
```

Same as above, but use the out-of-bag prediction-based algorithm instead of cross-fitting

```
cate aipw (y x1-x5 i.group1) (treat), controls(w1-w100)      ///
omethod(rforest) tmethod(rforest) oob
```

Use linear regression to fit the outcome model, logit regression to fit the treatment model, and linear regression to fit the CATE model

```
cate aipw (y x1-x5 i.group1) (treat),                        ///
omethod(regress) tmethod(logit) cmethod(regress)
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Conditional average treatment effects

Syntax

Partialing-out estimator

```
cate po (ovar catevarlist) (tvar) [if] [in] [ , options ]
```

Augmented inverse-probability weighting estimator

```
cate aipw (ovar catevarlist) (tvar) [if] [in] [ , options ]
```

ovar is a continuous outcome of interest.

catevarlist specifies the covariates of the CATE model—the conditioning variables for the treatment effects. *catevarlist* may contain factor variables; see [U] 11.4.3 **Factor variables**.

tvar must be a binary variable representing the treatment levels.

<i>options</i>	Description
Model	
<code>controls(<i>varlist</i>)</code>	specify the control variables for the outcome and treatment models
<code>* group(<i>varname</i>)</code>	compute the GATE for each group defined by <i>varname</i>
<code>* group(#)</code>	divide the data into # groups based on IATES and compute the GATES for each group
<code>rseed(#)</code>	set random-number seed
<code>xfolds(#)</code>	use cross-fitting algorithm with # folds; default is <code>xfolds(10)</code>
Method	
<code>omethod(<i>om_spec</i>)</code>	specify estimation method for outcome model; default is <code>omethod(lasso)</code>
<code>tmethod(<i>tm_spec</i>)</code>	specify estimation method for treatment model; default is <code>tmethod(lasso)</code>
<code>cmethod(<i>cm_spec</i>)</code>	specify estimation method for CATE model; default is <code>cmethod(rforest)</code>
Advanced	
<code>† reestimate</code>	reestimate GATES or GATESs with a new specification in <code>group()</code> and without refitting IATE function
<code>‡ oob</code>	use out-of-bag prediction-based algorithm instead of cross-fitting
<code>treatcontrols(<i>varlist</i>)</code>	use variables in <i>varlist</i> as controls for treatment model instead of variables specified in <code>controls()</code> (AIPW estimator only)
<code>pstolerance(#)</code>	set tolerance for overlap assumption; default is <code>pstolerance(1e-5)</code>
<code>osample(<i>newvar</i>)</code>	generate <i>newvar</i> to identify observations that violate the overlap assumption
<code>rflistwise</code>	omit observations with missing covariate values when random forest is used for all models
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>[no]log</code>	suppress iteration log
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

*Only one of `group(varname)` or `group(#)` may be specified.

†reestimate may be specified with `group(#)` only if `group(#)` has been specified in the previous `cate` estimation. If `reestimate` is specified with `group(new_varname)`, where `new_varname` is different than the previous `group(varname)`, then `new_varname` must have been a factor variable in `catevarlist` in the previous `cate` estimation.

‡oob may not be specified with `group(#)` or `xfolds()`. oob is only allowed if the random forest method has been specified for both the outcome and the treatment models (`omethod(rforest)` and `tmethod(rforest)`).

`collect` is allowed; see [U] 11.1.10 Prefix commands.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

<i>om_spec</i>	Description
<code>lasso[, <i>lasso_options</i>]</code>	use linear lasso to fit the outcome model; the default
<code>sqrlasso[, <i>lasso_options</i>]</code>	use square-root lasso to fit the outcome model
<code>rforest[, <i>rforest_options</i>]</code>	use random forest to fit the outcome model
<code>regress</code>	use linear regression to fit the outcome model

<i>tm_spec</i>	Description
<code>lasso[, <i>lasso_options</i>]</code>	use logit lasso to fit the treatment model; the default
<code>rforest[, <i>rforest_options</i>]</code>	use random forest to fit the treatment model
<code>logit</code>	use logit regression to fit the treatment model
<code>probit</code>	use probit regression to fit the treatment model

<i>cm_spec</i>	Description
<code>rforest[, <i>rforest_options</i>]</code>	use random forest to fit the CATE model; the default
<code>regress</code>	use linear regression to fit the CATE model

<i>rforest_options</i>	Description
<code>samprate(#)</code>	specify sampling rate for observations; default is <code>samprate(0.5)</code>
<code>ntrees(#)</code>	specify number of trees in the forest; default is <code>ntrees(2000)</code>
<code>cintrees(#)</code>	specify number of trees in each group to compute the confidence intervals; default is <code>cintrees(2)</code>
<code>splitminobs(#)</code>	specify minimum number of observations to split a node; default is <code>splitminobs(6)</code>
<code>splitmeanvars(#)</code>	specify mean number of variables to be split in each node; default is <code>splitmeanvars(ceil(sqrt(<i>p</i>) + 20))</code> with <i>p</i> as the dimension of <i>catevarlist</i>
<code>nohonest</code>	do not use an honest tree
<code>honestrate(#)</code>	set sampling rate for honest tree; default is <code>honestrate(0.5)</code>

Options

Model

`controls(varlist)` specifies the control variables for the outcome and treatment models. *catevarlist* and the specified control variables are the covariates in the outcome and the treatment models. If no control variables are specified, then the variables specified in *catevarlist* will be the only covariates for both models.

`group(varname)` computes the GATE for each group defined by the levels of *varname*. The ATE for each level in the group variable will be estimated. The grouping variable will be added as a factor variable in *catevarlist*. Only one of `group(varname)` or `group(#)` may be specified.

`group(#)` computes the GATESS by dividing the observations into # groups. The groups are generated from the quantiles of the estimates of the IATEs. The observations are sorted based on the IATEs and grouped into # levels. For example, if we specify `group(4)`, the data would be divided into four groups. The first group will contain observations with IATE estimates greater than the 75th percentile of the overall IATE estimates, the second group will contain observations that lie between the 50th and the 75th percentiles, the third group will contain those between the 25th and 50th percentiles, and the last group will contain those below the 25th percentile. Once the groups are formed, *cate* computes the ATE for each group. Only one of `group(#)` or `group(varname)` may be specified.

`rseed(#)` sets the random-number seed. `rseed(#)` is equivalent to typing `set seed #` prior to running *cate*. Random numbers are used to produce split samples for cross-fitting. To reproduce results, you must either use this option or use `set seed`. See [R] [set seed](#).

`xfolds(#)` specifies the number of folds for cross-fitting. The default is `xfolds(10)`; that is, cross-fitting is done by randomly dividing the original data into 10 folds.

Method

`omethod(om_spec)` specifies the estimation method for the outcome model. *om_spec* may be `lasso[ , lasso_options]`; `sqrlasso[ , lasso_options]`; `rforest[ , rforest_options]`; or `regress`. The default is `omethod(lasso)`.

`lasso[ , lasso_options]` specifies that linear lasso be used to fit the outcome model. *lasso_options* are `selection()`, `grid()`, `stop()`, `cvtolerance()`, `bictolerance()`, `tolerance()`, and `dtolerance()`; see [LASSO] [lasso options](#). If `selection()` is not specified, then `selection(plugin)` is assumed; that is, the plugin penalty parameter is used.

`sqrlasso[ , lasso_options]` specifies that square-root lasso be used to fit the outcome model. *lasso_options* are `selection()`, `grid()`, `stop()`, `cvtolerance()`, `bictolerance()`, `tolerance()`, and `dtolerance()`; see [LASSO] [lasso options](#). If `selection()` is not specified, then `selection(plugin)` is assumed; that is, the plugin penalty parameter is used.

`rforest[ , rforest_options]` specifies that random forest be used to fit the outcome model. *rforest_options* are `samprate()`, `ntrees()`, `cintrees()`, `splitminobs()`, `splitmeanvars()`, `nohonest`, and `honestrate()`.

`samprate(#)` sets the sampling rate for observations when drawing the random sample for each tree. The sampling is without replacement. The sampling rate must be in the range (0, 1). The default is `samprate(0.5)`, meaning that half of the estimation sample is used to construct each tree. Using a random sample to construct each tree makes the random forest more robust to the overfitting issues.

`ntrees(#)` sets the number of trees in the random forest. The default is `ntrees(2000)`. Using more trees in the random forest usually implies more stable estimates, but it also requires longer computational time.

`cintrees(#)` sets the number of trees in each group or bag when using the bootstrap of little bags to compute the confidence intervals of the random forest's predictions. Each tree in the same bag draws a random sample from the same half-size sample, which allows us to estimate the variance of the random forest's prediction. The default is `cintrees(2)`, two trees in each bag.

`splitminobs(#)` sets the minimum number of observations to perform a split in a node. A node must have at least `#` observations to be split. The default is `splitminobs(6)`.

`splitmeanvars(#)` sets the mean number of variables to be split in each node. In each tree node in a random forest, only a random subset of variables is searched to find the best splitting variable and value. The number of variables in this subset is also random and equals $\max(\min(m, p), 1)$, where p is the dimension of *catevarlist* and m follows a Poisson distribution with mean `#`. The default is `splitmeanvars(ceil(sqrt( $p$ ) + 20))`.

`nohonest` specifies not to use an honest tree. Honest splitting in an honest tree is the critical feature that allows us to make inferences on the random forest's prediction. Confidence intervals and standard errors for the random forest's prediction cannot be estimated when `nohonest` is specified.

`honestrate(#)` specifies the fraction of the sample used for splitting the honest tree. For a random sample S drawn to create a tree, an honest tree divides sample S into two disjoint parts A and B . Part A is used to split the tree and part B is used to label the tree. `honestrate()`

specifies the fraction of the sample S to be used as part A . Honest splitting is the key feature that allows inference on the random forest's prediction. The default is `honestrate(0.5)`, where half of the data in S is used to split the tree and the other half is used to label the tree.

`regress` specifies that linear regression be used to fit the outcome model. Thus, this option imposes a parametric assumption on the outcome model.

`tmethod(tm_spec)` specifies the estimation method for the treatment model. `tm_spec` may be `lasso[ , lasso_options]`, `rforest[ , rforest_options]`, `logit`, or `probit`. The default is `tmethod(lasso)`.

`lasso[ , lasso_options]` specifies that logit lasso be used to fit the treatment model. `lasso_options` are `selection()`, `grid()`, `stop()`, `cvtolerance()`, `bictolerance()`, `tolerance()`, and `dtolerance()`; see [LASSO] [lasso options](#). If `selection()` is not specified, then `selection(plugin)` is assumed; that is, the plugin penalty parameter is used.

`rforest[ , rforest_options]` specifies that random forest be used to fit the treatment model. `rforest_options` are `samprate()`, `ntrees()`, `cintrees()`, `splitminobs()`, `splitmeanvars()`, `nohonest`, and `honestrate()`; see [rforest options](#).

`logit` specifies that a logit model be used to fit the treatment model. Thus, this option imposes a parametric assumption on the treatment model.

`probit` specifies that a probit model be used to fit the treatment model. Thus, this option imposes a parametric assumption on the treatment model.

`cmethod(cm_spec)` specifies the estimation method for the CATE model. `cm_spec` may be `rforest[ , rforest_options]` or `regress`. The default is `cmethod(rforest)`.

`rforest[ , rforest_options]` specifies that random forest be used to fit the CATE model. `rforest_options` are `samprate()`, `ntrees()`, `cintrees()`, `splitminobs()`, `splitmeanvars()`, `nohonest`, and `honestrate()`; see [rforest options](#).

`regress` specifies that linear regression be used to fit the CATE model. Thus, this option assumes a parametric assumption on the CATE model.

Advanced

`reestimate` reestimates GATES or GATESS with a new specification in `group()`. It is much faster than estimating GATES or GATESS from scratch because it uses existing results from the previous `cate` estimation for the estimates of the IATE function. The typical usages of this option are the following:

`reestimate` with the `group(new_varname)` option reestimates the GATES with the group variable `new_varname`. This syntax requires that the group variable `new_varname` was specified as a factor variable in `catevarlist` in the previous `cate` estimation. For example, after using `cate` to estimate the GATE for each level of group variable `group1`,

```
cate (y x1 x2 i.group2) (treat), group(group1)
```

we can estimate the GATE for each level of group variable `group2`:

```
cate, reestimate group(group2)
```

Notice above that `group2` is specified as a factor variable in `catevarlist`.

`reestimate` with the `group(#)` option reestimates the GATESS with `#` data-driven groups. This syntax requires that option `group(#)` has been specified in the previous `cate` estimation. For example, after using `cate` to estimate the GATESS with 4 data-driven groups,

```
cate (y x1 x2) (treat), group(4)
```

we can estimate the GATESS with 5 data-driven groups:

```
cate, reestimate group(5)
```

Notice that we specified option `group(#)` in the previous `cate` specification.

`oob` uses the out-of-bag prediction-based algorithm instead of cross-fitting. It requires that the random forest is used to fit the outcome model and the treatment model; that is, both `omethod(rforest)` and `tmethod(rforest)` must be specified with `oob`. The out-of-bag prediction-based algorithm is generally faster than cross-fitting under the same setup. However, this algorithm does not allow computing the GATESS; thus, `oob` may not be combined with `group(#)`.

`treatcontrols(varlist)` specifies that *varlist* be used as controls in the treatment model instead of the variables specified in `controls()`. This option is only allowed for the AIPW estimator. If `treatcontrols()` is not specified, then the variables in *catevarlist* and the variables specified in `controls()` are used as covariates in both the outcome and the treatment models. If `treatcontrols(varlist)` is specified, then the treatment model instead uses *varlist* as covariates.

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `cate` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` generates indicator variable *newvar* identifying observations that violate the overlap assumption.

`rflistwise` specifies that listwise deletion be used when the random forest method is used for all models. By default, when `omethod(rforest)`, `tmethod(rforest)`, and `cmethod(rforest)` are all specified, observations with missing covariate values will be used because the random forest method can use missing covariate values in estimation. See [Generalized random forest](#) in [Methods and formulas](#).

When `rflistwise` is specified, observations with missing covariate values are not used to estimate the CATES. If one of `omethod()`, `tmethod()`, or `cmethod()` does not use `rforest`, then observations with missing covariate values will not be used for estimating the CATES.

Reporting

`level(#)`; see [\[R\] Estimation options](#).

`[no]log` displays or suppresses a log showing the progress of the estimation. The log is displayed by default unless you used `set iterlog off` to suppress it; see `set iterlog` in [\[R\] set iter](#).

display_options: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no!stretch`; see [\[R\] Estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

Introduction

What is a CATE?

Different versions of the CATE

Overview of the cate suite

Workflows

Workflow 1: Exploiting the IATE function heterogeneity

Workflow 2: Prespecified group hypothesis testing

Workflow 3: Data-driven group hypothesis testing

Workflow 4: Evaluation of counterfactual policies

Workflow 5: Evaluating policies designed using the IATE estimates

Examples

Example 1: Explore treatment-effect heterogeneity

Example 2: Add high-dimensional controls

Example 3: Estimate the ATEs over prespecified groups

Example 4: Estimate the ATEs over values of a continuous variable

Example 5: Use the AIPW estimator

Example 6: Data-driven group hypothesis testing

Example 7: Flexible models

Example 8: Treatment-assignment policy evaluation

Introduction

Treatment effects estimate the causal effect of a treatment on an outcome. This effect may be constant or it may vary across different subpopulations. For example, a labor economist may want to know if the earnings of immigrants and nonimmigrants are affected differently by a job training program and, if so, by how much. An online shopping company may want to know the effect of a price discount on purchasing behavior for customers with different demographic characteristics such as age or income. A medical team may want to measure the effect of smoking on stress levels for individuals in different age groups.

The ATE is a popular way to summarize the treatment effects by taking the mean of the effects over the population. The ATE characterizes the whole distribution of treatment effects when the treatment effect is constant across the population. However, when the treatment effects are heterogeneous and the units react differently to the same treatment, estimating only the mean of treatment effects may mask the underlying mechanism of how the treatment affects different units. For example, the estimated ATE may be close to zero when some groups experience positive effects while other groups experience adverse effects.

In contrast to the ATE, the CATES help us better understand the heterogeneous nature of treatment effects. Like the ATE, the CATES are averages of treatment effects, but unlike the ATE, the averages are taken over population subgroups. Imagine that we have a microscope to observe the treatment effects. The ATE only allows us to look at the effects at the most coarse precision, but the CATES allow us to zoom in on particular parts of the population. Furthermore, once we understand the heterogeneity of treatment effects, we can evaluate different treatment-assignment policies that may shed light on which policy would result in better overall outcomes for different groups in the population.

In summary, the advantage of studying the CATES is, at least, two-fold:

1. It improves understanding of the treatment-effect heterogeneity.
2. It builds a foundation to optimize the assignment to treatment.

What is a CATE?

So what is a CATE precisely? Under the potential outcome framework, we define $y_i(1)$ to be the potential outcome if unit i is treated and define $y_i(0)$ to be the potential outcome if unit i is not treated. $\mathbf{x}_i$ is a vector of characteristics for unit i . The CATE is defined as

$$\text{CATE} \equiv \tau(\mathbf{x}) = E\{y_i(1) - y_i(0) | \mathbf{x}_i = \mathbf{x}\}$$

That is, the CATE is the expectation of the difference between the treated and untreated potential outcomes conditional on the characteristics, $\mathbf{x}_i$, being equal to $\mathbf{x}$.

We can identify $\tau(\mathbf{x})$ via either a partial linear or a fully interactive model. Here we build intuition by focusing on identification of the CATE via the partial linear model. For notational simplicity, we drop the subscript i indicating the i th observation to refer to a random variable. We refer to the observed outcome as y and the binary treatment indicator as d .

In the simplest case, when the treatment effects are constant or when we are interested in estimating the ATE, the partial linear model is

$$\begin{aligned} y &= d * \tau + g(\mathbf{x}, \mathbf{w}) + \epsilon \\ d &= f(\mathbf{x}, \mathbf{w}) + u \end{aligned}$$

Here we divide the variables into two groups: $\mathbf{x}$ and $\mathbf{w}$. We will differentiate the two shortly. The outcome model is partial linear because the observed outcome is a sum of the treatment effects $d * \tau$, a nonparametric function $g(\mathbf{x}, \mathbf{w})$, and the error term ϵ . The treatment assignment is modeled by the function $f(\mathbf{x}, \mathbf{w})$ and an additive error term u . By definition, we can write the potential outcome models as

$$\begin{aligned} y(1) &= \tau + g(\mathbf{x}, \mathbf{w}) + \epsilon \\ y(0) &= g(\mathbf{x}, \mathbf{w}) + \epsilon \end{aligned}$$

Thus, τ characterizes the ATE.

$$\text{ATE} \equiv E\{y(1) - y(0)\} = \tau$$

Now let's go one step further. Suppose the treatment effects are heterogeneous and depend on $\mathbf{x}$. We can rewrite the outcome model as

$$y = d * \tau(\mathbf{x}) + g(\mathbf{x}, \mathbf{w}) + \epsilon$$

where $\mathbf{x}$ is a vector of conditioning variables for the treatment effects. $\tau(\mathbf{x})$ is a function of $\mathbf{x}$ that interacts with the treatment d . Notice that $\tau(\mathbf{x})$ is a function of $\mathbf{x}$ but not of $\mathbf{w}$. $\mathbf{w}$ is an optional vector of additional control variables for the outcome and treatment-assignment models, which can potentially be high-dimensional. This model is flexible and general because it does not impose parametric assumptions on $\tau(\mathbf{x})$ or $g(\mathbf{x}, \mathbf{w})$. The potential outcomes now become

$$\begin{aligned} y(1) &= \tau(\mathbf{x}) + g(\mathbf{x}, \mathbf{w}) + \epsilon \\ y(0) &= g(\mathbf{x}, \mathbf{w}) + \epsilon \end{aligned}$$

Thus, the CATE is $\tau(\mathbf{x})$.

$$E\{y(1) - y(0) | \mathbf{x}\} = \tau(\mathbf{x})$$

If we impose parametric assumptions such as $\tau(\mathbf{x}) = \mathbf{x}'\beta$ and $g(\mathbf{x}, \mathbf{w}) = \mathbf{x}'\gamma_1 + \mathbf{w}'\gamma_2$, we can estimate this special model by running a regression of y on $\mathbf{x}$, $\mathbf{w}$, and the interaction between d and $\mathbf{x}$. However, this parametric assumption may be too strong and hard to satisfy with our data. In the implementation of `cate`, we focus on a more general case that does not impose a parametric form on $\tau(\mathbf{x})$ or the nuisance parameters $g(\mathbf{x}, \mathbf{w})$ and $f(\mathbf{x}, \mathbf{w})$. The parameter of interest, $\tau(\mathbf{x})$, can be estimated nonparametrically via the generalized random forest proposed in [Athey, Tibshirani, and Wager \(2019\)](#). The nuisance parameters can be estimated using lasso or random forest. However, if we want to impose a parametric assumption on either $\tau(\mathbf{x})$ or the nuisance parameters, `cate` can also fit such models.

Different versions of the CATE

The granularity of the conditional set $\mathbf{x}_i$ determines the subpopulation. Thus, the CATE has different versions or names depending on the definition of the conditional set $\mathbf{x}_i$.

IATE: At the finest level, when $\mathbf{x}_i$ refers to the characteristics of a specific observation, the CATE measures the expected treatment effect for individuals with the same characteristics as this observation. This version of CATE is also called IATE. In literature, people often refer to IATE as CATE, even though IATE is a special version of CATE. In our terminology, we use IATE when we refer to the finest level of CATE.

Under the [unconfoundedness assumption](#), it is possible to identify and estimate $\tau(\mathbf{x})$ if we are willing to impose some restrictions on $\tau(\mathbf{x})$. In particular, when $\mathbf{x}$ is a low-dimensional vector and $\tau(\mathbf{x})$ is smooth enough as defined in [Athey, Tibshirani, and Wager \(2019\)](#), we can nonparametrically estimate the IATE and provide confidence intervals using the generalized random forest proposed in [Athey, Tibshirani, and Wager \(2019\)](#).

GATE: If $\mathbf{x}_i$ is a prespecified grouping, denoted by G_i , the CATE measures the ATE for each group. This version of CATE is also called GATE.

In particular, the GATE is defined as

$$\tau(g) = E\{y_i(1) - y_i(0) | G_i = g\}$$

where G_i is a prespecified grouping and g is a specific group. The GATES are coarser than the IATES because they focus on group effects instead of individual effects.

For the GATES, we must specify the group variable, G_i , before analyzing the data to avoid p -value hacking, as discussed in [Head et al. \(2015\)](#).

GATES: Sometimes, we do not have a group variable to specify but still want to understand the underlying treatment-effect heterogeneity. In such cases, we can discover the groups in a data-driven way by using the sorted IATES. This version of CATE is known as GATES.

The groups are generated by the quantiles of the IATE estimates. For example, let's say we want to divide the data into four groups. The first group will consist of the observations with IATE estimates greater than the 75th percentile of the overall IATE estimates, the second group will include observations whose estimates lie between the 50th and 75th percentiles, the third group will contain observations with estimates between the 25th and 50th percentile, and the last group will contain observations with estimates below the 25th percentile.

Once we have the groups as above, we can estimate the GATES as usual.

The `cate` command estimates IATEs, GATEs with prespecified groups, and GATESSs, for which groups are determined in a data-driven way. `cate` helps us answer questions about treatment heterogeneity, such as the following:

1. Are the treatment effects heterogeneous?
2. How do the treatment effects vary with some variables?
3. Do the treatment effects vary between prespecified groups?
4. Do the data discover groups where treatment effects are different?

Additionally, we can use the estimates obtained from `cate` to measure the effect of counterfactual treatment-assignment policies on the outcome by using postestimation commands such as `estat policyeval` and `estat ate`. Suppose a hypothetical treatment-assignment policy assigns some individuals in the sample to be treated and some others not to be treated. Using `estat policyeval` and `estat ate` postestimation commands, we may answer questions such as the following:

1. If we implement such a policy, how would the average outcome in the population change?
2. Which policy is better among a candidate set of policies?

Overview of the cate suite

Here we outline the Stata commands to estimate, predict, visualize, and make inferences about the CATES. In particular, these Stata commands can be grouped into the following categories:

Estimation: `cate po` estimates the IATE function by using the PO estimator discussed in Nie and Wager (2021) via the generalized random forest proposed in Athey, Tibshirani, and Wager (2019). This method is the default and is also known as causal forest.

`cate aipw` estimates the IATE function by using the AIPW estimator discussed in Knaus (2022) and Kennedy (2023) via the honest regression random forest proposed in Wager and Athey (2018) by default.

`cate` with option `group(varname)` estimates the GATEs by taking the means of the AIPW scores implied by the model (the estimates of the individual-level treatment effects) over the group variable `varname`. This method is discussed in Semenova and Chernozhukov (2021) and Knaus (2022).

`cate` with option `group(#)` estimates the GATESSs for `#` groups with levels based on the rankings of the IATE estimates. This method is discussed in Chernozhukov et al. (2006). Once the groups are discovered, the GATESSs are estimated as the GATEs.

Prediction: `predict` predicts the IATE function $\tau(\mathbf{x})$, its standard errors, and the lower and upper bounds of the pointwise confidence intervals. The estimates of standard errors and confidence intervals are computed using the bootstrap of little bags proposed in Athey, Tibshirani, and Wager (2019).

Visualization: `categraph histogram` plots the histogram of estimated IATEs and shows how the IATEs are distributed. This histogram can serve as a preliminary visualization of the treatment-effect heterogeneity.

`categraph gateplot` plots the estimate of the GATE or GATES and its confidence interval for each group. It visualizes the trend of the GATE or GATES function.

`categraph iateplot` plots the IATE function with varying values of $\mathbf{x}$.

If $\mathbf{x}$ is a vector, we can allow one variable to vary, fix the values of the other variables, and then use `categraph iateplot` to plot the function.

Inference: `estat heterogeneity` tests whether the treatment effects are heterogeneous using the method proposed by Chernozhukov et al. (2006).

`estat gatetest` tests whether the estimated GATES or GATESS are equal across the groups.

`estat classification` compares the means of a variable in the group with the largest treatment effect and the group with the smallest treatment effect. It is used to compare the properties of the subpopulations with the largest and smallest effects.

`estat ate` computes the ATE for a subpopulation. This command can be useful for policy evaluations.

`estat projection` fits a linear regression of the estimated IATE on a vector of variables. It provides a linear approximation of the IATE function.

`estat series` fits a nonparametric series regression of the estimated IATE on a vector of variables using B-spline, piecewise polynomial spline, or polynomial basis. It provides a nonparametric approximation of the IATE function, as discussed in Semenova and Chernozhukov (2021).

`estat policyeval` evaluates and compares the prespecified treatment-assignment policy. In particular, it computes the value of a treatment-assignment policy or compares the difference of two policies' values if specified.

Workflows

Here we provide possible workflows that may be useful, depending on the question of interest. Workflows 1 to 3 help us to answer questions regarding the treatment-effects heterogeneity, and workflows 4 and 5 help us to evaluate treatment-assignment policies. Below, we list the questions of interest for each workflow. Then, we will discuss the details of the workflows.

1. Understand treatment-effect heterogeneity:

- **Workflow 1: Exploiting the IATE function heterogeneity**

Given an estimate of the IATE function, are the treatment effects heterogeneous?

- **Workflow 2: Prespecified group hypothesis testing**

We have some prespecified groups and we want to test whether the treatment effects are the same across these groups or study how the effects differ across them.

- **Workflow 3: Data-driven group hypothesis testing**

We do not know the groups for which the treatment effects may vary, so we ask the data to discover these groups and study whether certain variables may be correlated with the treatment-effect heterogeneity.

2. Policy evaluation:

- **Workflow 4: Evaluation of counterfactual policies**

We want to evaluate some prespecified treatment-assignment policies and compare them.

- **Workflow 5: Evaluating policies designed using the IATE estimates**

We want to evaluate a policy that is designed based on the IATE estimates. For example, we want to treat all units with an IATE greater than a fixed cost and evaluate the effect of such a policy on the outcome.

Workflow 1: Exploiting the IATE function heterogeneity

1. Suppose the outcome variable is `y`, the CATE covariates are `x1-x5`, the treatment variable is `treat`, and the control variables are `w1-w100`. Estimate the IATE function. (We demonstrate using the PO estimator but could also use the AIPW estimator.)

```
cate po (y x1-x5) (treat), controls(w1-w100)
```

2. Plot the histogram of IATE estimates.

```
categraph histogram
```

3. Test whether the effects are heterogeneous.

```
estat heterogeneity
```

4. Regress the estimated IATE function on variables that may impact treatment effects to understand the mechanism underlying the treatment-effect heterogeneity.

```
estat projection x1-x5
```

5. Estimate the ATE for a subpopulation of interest. Suppose we suspect that the variable `x1` positively affects the treatment effects; we can estimate the ATE for the subpopulation where `x1` is greater than 0.8.

```
estat ate if x1 > 0.8
```

6. Plot the IATE function for `x1` while the other variables are fixed at specific values, such as their means.

```
estat iateplot x1, at((mean) x2-x5)
```

Workflow 2: Prespecified group hypothesis testing

1. Estimate the GATE function over the levels of group variable `gvar`. (We demonstrate using the AIPW estimator but could also use the PO estimator.)

```
cate aipw (y x1-x5) (treat), controls(w1-w100) group(gvar)
```

2. Visualize the GATE estimates and their confidence intervals.

```
categraph gateplot
```

3. Test whether the GATES are the same across groups.

```
estat gatetest
```

Workflow 3: Data-driven group hypothesis testing

1. Estimate the GATES function for five groups created by dividing data into groups based on rankings of the IATE estimates. (We demonstrate using the AIPW estimator but could also use the PO estimator.)

```
cate aipw (y x1-x5) (treat), controls(w1-w100) group(5)
```

2. Visualize the GATES estimates and their confidence intervals.

```
categraph gateplot
```

3. Test whether the GATESs are the same across groups.

```
estat gatetest
```

4. Compare the mean of `x1` in the groups with the smallest and largest treatment effects.

```
estat classification x1
```

Workflow 4: Evaluation of counterfactual policies

1. Estimate the IATE function. (We demonstrate using the PO estimator but could also use the AIPW estimator.)

```
cate po (y x1-x5) (treat), controls(w1-w100)
```

2. Estimate the average outcome in the population for a potential policy. Suppose the `policy1` variable stores the treatment assignments for each observation under a counterfactual policy; we estimate the average outcome for `policy1`.

```
estat policyeval policy1
```

3. Compare the average outcomes in the population for multiple potential policies. Suppose the `policy2` variable stores the treatment assignments for an alternative policy; we compare average outcomes for `policy1` and `policy2`.

```
estat policyeval policy1 policy2
```

Workflow 5: Evaluating policies designed using the IATE estimates

1. Split the sample into training and testing data.

```
splitsample, split(0.6 0.4) generate(group)
```

2. Estimate the IATE function using the training data, `group = 1`. (We demonstrate using the PO estimator but could also use the AIPW estimator.)

```
cate po (y x1-x5) (treat) if group == 1, controls(w1-w100)
```

3. Predict the IATE function in the testing data, `group = 2`; the predicted IATE function is used to construct the policy rule.

```
predict tauhat if group == 2
```

4. Estimate the IATE function using the testing data; the AIPW scores will be to compute the ATE in the testing sample.

```
cate po (y x1-x5) (treat) if group == 2, controls(w1-w100)
```

5. Estimate the ATE for the entire testing sample.

```
estat ate if group == 2
```

6. Estimate the ATE for a subset of units in the testing sample based on the IATE predictions. For instance, estimate the ATE for units with predicted IATEs greater than 50 in the testing sample.

```
estat ate if group == 2 & tauhat >= 50
```

Examples

In the following examples, we illustrate how to use `cate` to study treatment-effect heterogeneity and to evaluate treatment-assignment policies. In particular, in examples 1 to 7, we demonstrate commands to evaluate the effects of 401(k) program eligibility on net financial wealth. Suppose that we want to answer the following questions:

1. Are the effects of 401(k) eligibility on net wealth heterogeneous? In other words, do the treatment effects vary across individuals or groups?
2. If the treatment effects are heterogeneous, how do they vary across levels of prespecified group variables, such as income category, home ownership, or education level?
3. Do the data discover groups in which the treatment effects are particularly high or low?

In [example 8](#), we demonstrate commands to evaluate the effects of two types of lung transplants on patients' health outcomes. Supposing a doctor has a treatment-assignment recommendation rule, we want to evaluate the overall outcomes if this treatment-assignment rule is implemented.

Example 1: Explore treatment-effect heterogeneity

Suppose we want to estimate the effect of 401(k) eligibility (`e401k`) on net financial assets (`asset`) using data reported by [Chernozhukov and Hansen \(2004\)](#). These data are from a sample of households in the 1990 Survey of Income and Program Participation (SIPP). The data contain information on the head of the household: income level category (`incomecat`), age (`age`), years of education (`educ`), whether they receive pension benefits (`pension`), marital status (`married`), whether they participate in an IRA (`ira`), whether they own a home (`ownhome`), and whether there are two earners in the same household (`twoearn`).

We believe that the treatment effects of `e401k` on `asset` could vary based on `incomecat`, `age`, `educ`, `pension`, `married`, `ira`, `ownhome`, and `twoearn`, which we denote as $\mathbf{x}$. `asset(1)` represents the potential outcomes (net financial assets) of being eligible for a 401(k), and `asset(0)` represents the potential outcomes of not being eligible for a 401(k). We want to estimate the effects of 401(k) eligibility on assets conditional on the variables $\mathbf{x}$. In other words, we are interested in estimating the effects as a function of $\mathbf{x}$. Precisely, we want to estimate

$$E\{\text{asset}(1) - \text{asset}(0)|\mathbf{x}\}$$

This version of CATEs is also known as individualized average treatment effects (IATEs) because $\mathbf{x}$ refers to individual characteristics. In the syntax of `cate`, $\mathbf{x}$ is referred to as the *catevarlist*.

In this example, we use the PO estimator in the partial linear model to estimate the IATE function. Without assuming any additional control variables, the partial linear model for `asset` is

$$\text{asset} = \text{e401k} * \tau(\mathbf{x}) + g(\mathbf{x}) + \epsilon$$

where $\tau(\mathbf{x})$ is a function of $\mathbf{x}$ that interacts with the treatment `e401k`, $g(\mathbf{x})$ is a nonparametric nuisance function, and ϵ is the error term for the outcome. The treatment-assignment model for the treatment `e401k` is

$$\text{e401k} = f(\mathbf{x}) + u$$

where $f(\mathbf{x})$ is a nonparametric nuisance function and u is an error term for the treatment.

The potential outcomes are

$$\text{asset}(1) = \tau(\mathbf{x}) + g(\mathbf{x}) + \epsilon$$

$$\text{asset}(0) = g(\mathbf{x}) + \epsilon$$

Thus, the function $\tau(\mathbf{x})$ identifies the IATE function.

$$\tau(\mathbf{x}) = E\{\text{asset}(1) - \text{asset}(0)|\mathbf{x}\}$$

Notice that we do not assume any functional form of $\tau(\mathbf{x})$, and it can be as simple as a linear model or any arbitrary function of $\mathbf{x}$. Here we want the data to tell us what this function $\tau(\mathbf{x})$ looks like instead of assuming a specific functional form. We can use `cate` to estimate the function $\tau(\mathbf{x})$ nonparametrically via the generalized random forest proposed in [Athey, Tibshirani, and Wager \(2019\)](#); this method is also known as causal forest and is the default method used by `cate`.

First, we open the `assets3` data. To save some typing later, we define a global macro, `catecovars`, to represent the IATE conditioning variables $\mathbf{x}$.

```
. use https://www.stata-press.com/data/r19/assets3
  (Excerpt from Chernozhukov and Hansen (2004))

. global catecovars age educ i.(incomecat pension married twoearn ira ownhome)
```

We are ready to fit the model using cate. We specify po to use the partialing-out estimator. We specify the outcome variable `asset` and `catevarlist` (the `x` variables) in the first set of parentheses and the treatment-assignment variable `e401k` in the second set. We also specify the `rseed()` option to make the results reproducible. The first portion of the output is

```
. cate po (assets $catecovars) (e401k), rseed(12345671)
Cross-fit fold 1 of 10 ...
Performing lasso for outcome assets ...
Performing lasso for treatment e401k ...
(output omitted)
Cross-fit fold 10 of 10 ...
Performing lasso for outcome assets ...
Performing lasso for treatment e401k ...
```

This iteration log corresponds to the cross-fitting process that is used to fit the outcome model for `assets` and the treatment model for `e401k`. To estimate the IATE function $\tau(\mathbf{x})$, the PO estimator needs to partial out the nuisance functions $g(\mathbf{x})$ and $f(\mathbf{x})$. To do this, cate needs to estimate the expectation of the outcome and the treatment variable conditional on `x`. By default, the lasso for the linear model is used to estimate the outcome `assets`, and the lasso for the logit model is used to estimate the treatment `e401k`. To guard against errors when fitting the nuisance functions (variable selection errors when using lasso and prediction errors when using random forest), cate uses cross-fitting. By default, ten-fold cross-fitting is used. See [Methods and formulas](#) for details.

We can also use other alternatives, such as a random forest or a parametric model, to estimate the outcome and the treatment models. Here, we use the default lasso for both models.

The remaining output is

```
Performing random forest for IATE ...
Estimating AIPW scores ...
Estimating ATE ...
Conditional average treatment effects      Number of observations      = 9,913
Estimator:      Partialing out              Number of folds in cross-fit = 10
Outcome model:  Linear lasso                 Number of outcome controls  = 17
Treatment model: Logit lasso                 Number of treatment controls = 17
CATE model:     Random forest                 Number of CATE variables    = 17
```

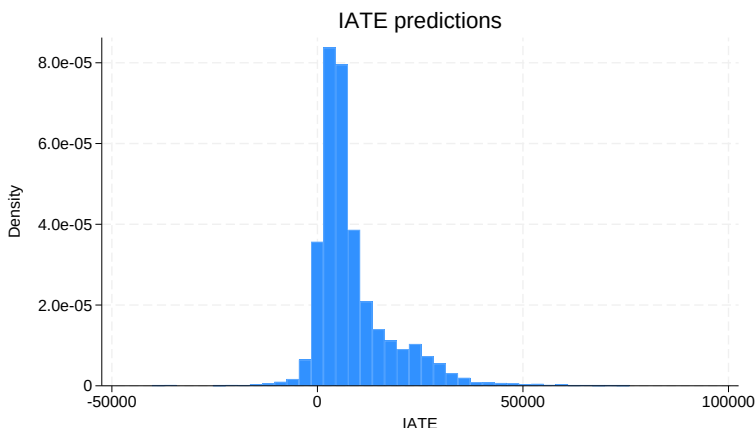
assets	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
e401k						
(Eligible						
vs						
Not elig..)	7937.182	1153.017	6.88	0.000	5677.309	10197.05
POmean						
e401k						
Not eligi..	14016.38	833.4423	16.82	0.000	12382.87	15649.9

We see that a random forest is used to estimate the IATE function once the cross-fitting is finished. It then estimates the AIPW scores, not to be confused with the AIPW estimator. The AIPW scores are doubly robust estimates of individual-level treatment effects. The average of these AIPW scores is the ATE. The estimated ATE indicates that if everyone in the population is eligible for a 401(k), the net financial assets

will, on average, be \$7,937 larger than the net financial assets if no one is eligible for a 401(k). The potential outcome mean indicates that net financial assets are expected to be \$14,016 if no one is eligible for a 401(k).

In addition to the ATE that we see in the output, `cate` also estimates the IATE function $\tau(\mathbf{x})$, and we can use it to predict the treatment effects for each observation. We can use `categraph histogram` to draw a histogram of the predicted $\tau(\mathbf{x})$ function and see its distribution.

```
. categraph histogram
(bin=39, start=-40204.13, width=2975.4332)
```



The graph shows that treatment effects are mostly positive but have a fat right tail. Thus, the ATE may underestimate the effect of 401(k) eligibility on assets for some groups.

Although the histogram above allows us to inspect the distribution of treatment effects visually, we should not use it as conclusive evidence to support treatment-effect heterogeneity. For example, when the number of observations in the sample is small or when the number of the CATE conditioning variables $\mathbf{x}$ is very large, we will likely see a well-spread histogram of IATE predictions due to the estimation noise even if the actual CATE function is constant.

To statistically test whether the treatment effects are heterogeneous, we use `estat heterogeneity`.

```
. estat heterogeneity
Treatment-effects heterogeneity test
H0: Treatment effects are homogeneous
      chi2(1) =   4.11
Prob > chi2 = 0.0427
```

We find evidence against the null hypothesis that the treatment effects are homogeneous.

To further explore the heterogeneity of treatment effects, we want to know whether a variable positively or negatively affects the treatment effects. One way to do this is by linearly projecting the AIPW scores, the individual-level treatment effects estimated by `cate`, on the variables of interest. We use `estat projection`.

```
. estat projection
Treatment-effects linear projection
```

Number of obs =	9,913
F(11, 9901) =	4.90
Prob > F =	0.0000
R-squared =	0.0045
Adj R-squared =	0.0034
Root MSE =	1.146e+05

	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
age	205.1206	117.9809	1.74	0.082	-26.14605	436.3873
educ	-442.4583	488.4721	-0.91	0.365	-1399.963	515.0466
incomecat						
1	-2439.222	2013.522	-1.21	0.226	-6386.136	1507.692
2	1874.817	2295.155	0.82	0.414	-2624.154	6373.788
3	5707.689	3298.341	1.73	0.084	-757.7313	12173.11
4	18194.6	5398.391	3.37	0.001	7612.651	28776.54
pension Receives ..	3817.355	2454.437	1.56	0.120	-993.8419	8628.553
married Married	-2399.333	3403.066	-0.71	0.481	-9070.035	4271.37
twoearn Yes	-1428.041	4347.025	-0.33	0.743	-9949.094	7093.013
ira Yes	-2438.404	3619.217	-0.67	0.500	-9532.807	4656
ownhome Yes	3162.649	1669.587	1.89	0.058	-110.081	6435.379
_cons	232.7251	8072.023	0.03	0.977	-15590.08	16055.53

Without specifying any variables, `estat projection` projects the AIPW scores on all the variables defined in `x`. In other words, it performs a regression of $\tau(\mathbf{x})$ on the conditioning variables in our model. We can interpret the coefficients as the effects of variables on the linear approximation of treatment effects. For example, the coefficient for 4. `incomecat` is 18,195. We can say that being in the highest income category increases the 401(k) eligibility effects on assets by \$18,195 over being in the lowest income category if the treatment effects are linearly approximated by the variables defined in `x`.

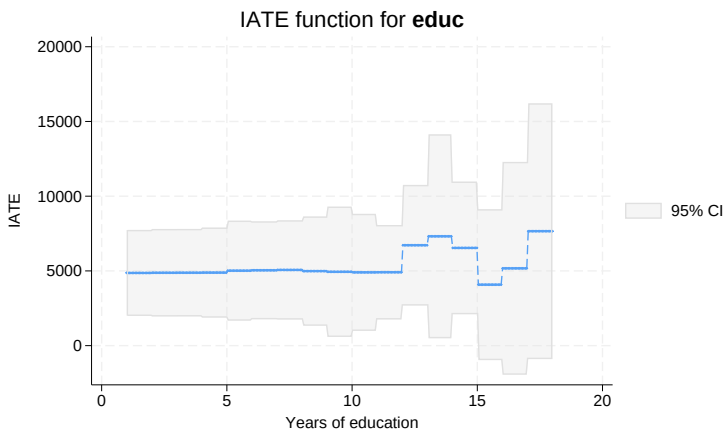
We can also plot the estimated IATE function by allowing one variable to vary and fixing the other variables to set values. For example, we can plot the function $\tau(\mathbf{x})$ by allowing `educ` to vary and fixing the values of the other variables.

We use `categraph iateplot`. We specify the variable `educ` to allow the IATE function to vary with `educ`. By default, the continuous variables, such as `age`, are fixed at their sample means, and the factor variables are fixed at their base levels.

```
. categraph iateplot educ
```

Note: IATE estimated at fixed values of covariates other than `educ`.

Variable	Statistic	Value	Type
age	mean	41.05891	continuous
incomecat	base	0	factor
ira	base	0	factor
married	base	0	factor
ownhome	base	0	factor
pension	base	0	factor
twoearn	base	0	factor



`categraph iateplot` plots the prediction of the IATE function at different education levels while holding other variables fixed. It also plots the 95% pointwise confidence interval for each prediction. Below 10 years of education, the effects seem constant. The treatment effects are larger for people with 12 to 14 years of education and then vary for 15 or more years of education. However, with the wide confidence intervals for the IATEs, especially at the higher education levels, we cannot conclude that the IATEs vary across education levels when the other variables are fixed at these levels.

Example 2: Add high-dimensional controls

In the previous example, the control variables for the outcome and the treatment-assignment model coincide with `catevarlist` or the $\mathbf{x}$ variables. In this example, we want to allow more flexible models for the outcome and the treatment-assignment by adding high-dimensional controls.

The partial linear model with high-dimensional controls is defined as

$$\text{asset} = \text{e401k} * \tau(\mathbf{x}) + g(\mathbf{x}, \mathbf{w}) + \epsilon$$

$$\text{e401k} = f(\mathbf{x}, \mathbf{w}) + u$$

where $\mathbf{w}$ is a vector of additional control variables. The nuisance functions $g(\mathbf{x}, \mathbf{w})$ and $f(\mathbf{x}, \mathbf{w})$ now depend on both $\mathbf{x}$ and $\mathbf{w}$.

The potential outcomes are now

$$\begin{aligned}\text{asset}(1) &= \tau(\mathbf{x}) + g(\mathbf{x}, \mathbf{w}) + \epsilon \\ \text{asset}(0) &= g(\mathbf{x}, \mathbf{w}) + \epsilon\end{aligned}$$

Thus, the function $\tau(\mathbf{x})$ identifies the IATE function.

$$\tau(\mathbf{x}) = E\{\text{asset}(1) - \text{asset}(0) | \mathbf{x}\}$$

Here we want to include the interactions between the continuous variables (age and educ) and the factor variables (incomecat, pension, married, twoearn, ira, and ownhome) as controls in the outcome and treatment-assignment models. We define a global macro, `controls`, to represent the interaction terms.

```
. global fvars incomecat pension married twoearn ira ownhome
. global controls c.(educ age)#i.($fvars)
```

We add the `controls()` option to our `cate` command to include the additional control variables in both the outcome and the treatment model. We also specify option `nolog` to suppress the iteration log.

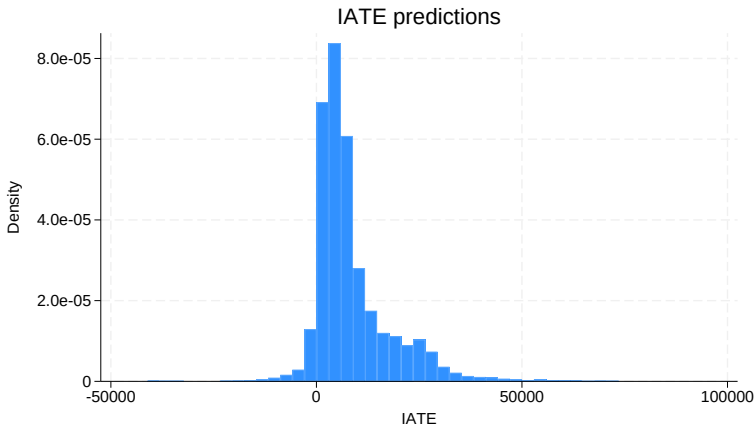
```
. cate po (assets $catecovars) (e401k), rseed(12345671) controls($controls)
> nolog
Conditional average treatment effects      Number of observations      = 9,913
Estimator:      Partialing out              Number of folds in cross-fit = 10
Outcome model:   Linear lasso                 Number of outcome controls  = 47
Treatment model: Logit lasso                  Number of treatment controls = 47
CATE model:      Random forest                 Number of CATE variables    = 17
```

assets	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE e401k (Eligible vs Not elig..)	8107.563	1144.817	7.08	0.000	5863.763	10351.36
POMean e401k Not eligi..	13902.88	838.5924	16.58	0.000	12259.27	15546.49

The header shows that there are now 47 control variables for both the outcome and the treatment model. After accounting for these controls, the ATE indicates that if everyone in the population were eligible for a 401(k), the net financial assets would be \$8,108 more than if no one in the population were eligible. This is a larger estimated effect of 401(k) eligibility on financial assets than the ATE of \$7,937 estimated in [example 1](#).

We can again use `categraph histogram` to visualize the distribution of the predicted IATE function.

```
. categraph histogram
(bin=39, start=-41089.941, width=2938.3059)
```



The IATE predictions are primarily positive, and the distribution has a fat right tail. This may imply that the ATE underestimates the treatment effects for some groups in the population. To statistically test if the treatment effects are heterogeneous, we use `estat heterogeneity`.

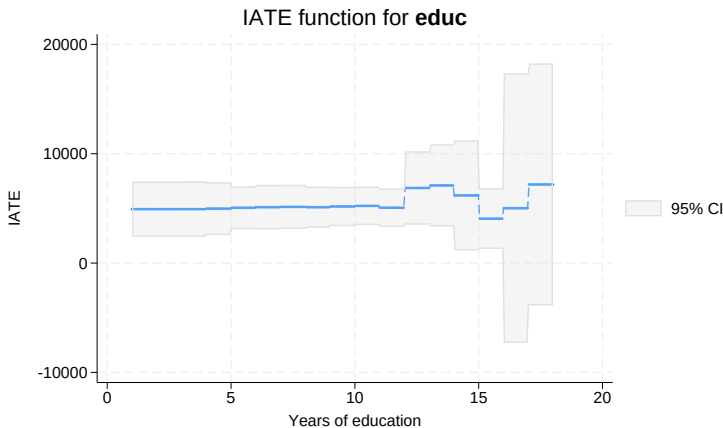
```
. estat heterogeneity
Treatment-effects heterogeneity test
H0: Treatment effects are homogeneous
      chi2(1) =    4.19
Prob > chi2 = 0.0406
```

We find evidence against the null hypothesis that the treatment effects are homogeneous, which is the same conclusion as in [example 1](#).

Finally, we can use `categraph iateplot` to plot the IATE function by allowing one variable to vary and fixing the other variables at some values. We plot the IATE function with respect to the level of education (`educ`).

```
. categraph iateplot educ
Note: IATE estimated at fixed values of covariates other than educ.
```

Variable	Statistic	Value	Type
age	mean	41.05891	continuous
incomecat	base	0	factor
ira	base	0	factor
married	base	0	factor
ownhome	base	0	factor
pension	base	0	factor
twoearn	base	0	factor



The graph shows that the treatment effects are larger for people with 12 to 14 years of education while holding other variables fixed. Again, the confidence intervals for the IATEs are wide, especially at the highest education levels.

Example 3: Estimate the ATEs over prespecified groups

In examples 1 and 2, we learned that the treatment effects of 401(k) eligibility on financial assets are heterogeneous. To further characterize this heterogeneity, we want to know how the ATEs vary across population groups defined by variables such as income category or home ownership.

In general, we refer to a group variable as G and a specific level of the group variable as g . We want to estimate the ATE conditional on belonging to group g , that is, $G = g$. We are interested in estimating

$$\tau(g) = E\{\text{asset}(1) - \text{asset}(0) | G = g\}$$

The function $\tau(g)$ is referred to as the GATE function. In our case, the first G variable of interest is the income category (`incomecat`). We will consider the home ownership indicator (`ownhome`) later.

We use `table` to report the minimum, maximum, and median income for the five income categories.

```
. table incomecat, stat(min income) stat(max income) stat(median income) nototal
```

	Minimum value	Maximum value	Median
Income category			
0	0	17196	12240
1	17214	26523	21735
2	26526	37275	31482
3	37296	53841	44379
4	53844	242124	69612

Levels 0 and 1 are low-income groups, levels 2 and 3 are middle-income groups, and level 4 is the high-income group.

The GATES are summaries of the IATE function over the groups defined by variable G . Because we already estimated the IATE function in example 2, there is no need to estimate it again. By specifying the option `reestimate`, we can reuse the IATE function and only reestimate the effects reported, here the GATES for `incomecat`. With this option, `cate` will require less computational time than estimating the GATES from scratch. We specify `group(incomecat)` to estimate GATES for the income categories.

```
. cate, group(incomecat) reestimate
```

```
Estimating GATE ...
```

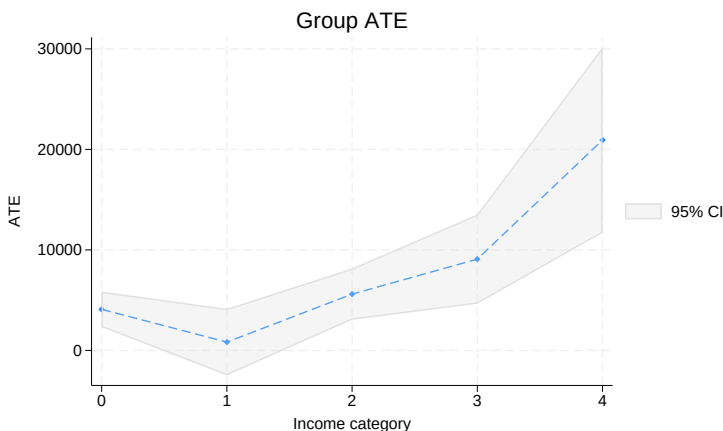
Conditional average treatment effects	Number of observations	= 9,913
Estimator: Partialing out	Number of folds in cross-fit	= 10
Outcome model: Linear lasso	Number of outcome controls	= 47
Treatment model: Logit lasso	Number of treatment controls	= 47
CATE model: Random forest	Number of CATE variables	= 17

assets	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
GATE						
incomecat						
0	4089.228	900.535	4.54	0.000	2324.212	5854.244
1	830.3422	1687.517	0.49	0.623	-2477.13	4137.815
2	5602.296	1300.555	4.31	0.000	3053.256	8151.336
3	9084.531	2265.143	4.01	0.000	4644.933	13524.13
4	20929.77	4706.377	4.45	0.000	11705.44	30154.1
ATE						
e401k						
(Eligible						
vs						
Not elig..)	8107.563	1144.817	7.08	0.000	5863.763	10351.36
POmean						
e401k						
Not eligi..	13902.88	838.5924	16.58	0.000	12259.27	15546.49

The results show both the ATE and the GATES. For example, the GATE estimate for the high-income group (level 4) is \$20,930. For those in the high-income group, being eligible for a 401(k) is expected to increase net financial assets by \$20,930 compared with the net financial assets if not eligible for a 401(k). In contrast, the GATE estimate for the lowest income group (level 0) is only \$4,089. In other words, people who earn more benefit more from working for a company with a 401(k) plan. The ATE estimate indicates that the treatment effects for the population are expected to be \$8,108. The variation in the estimated GATES across income categories indicates that using the ATE alone does not fully characterize the treatment effects.

We can use `categraph gateplot` to visualize the GATE estimates and see if there is any trend.

```
. categraph gateplot
```



The graph illustrates the upward trend between the income group and the treatment effects.

To further test whether the treatment effects are heterogeneous across income groups, we use `estat gatetest`.

```
. estat gatetest
```

Group treatment-effects heterogeneity test

H0: Group average treatment effects are homogeneous

```
( 1) [GATE]0bn.incomecat - [GATE]1.incomecat = 0
```

```
( 2) [GATE]0bn.incomecat - [GATE]2.incomecat = 0
```

```
( 3) [GATE]0bn.incomecat - [GATE]3.incomecat = 0
```

```
( 4) [GATE]0bn.incomecat - [GATE]4.incomecat = 0
```

```
chi2(4) = 21.84
```

```
Prob > chi2 = 0.0002
```

We find evidence that the group treatment effects are not homogeneous.

To compare the treatment-effects difference between groups, we use contrast.

```
. contrast r.incomecat
warning: cannot perform check for estimable functions.
Contrasts of marginal linear predictions
Margins: asbalanced
```

	df	chi2	P>chi2
GATE			
incomecat			
(1 vs 0)	1	2.90	0.0884
(2 vs 0)	1	0.91	0.3388
(3 vs 0)	1	4.20	0.0404
(4 vs 0)	1	12.35	0.0004
Joint	4	21.84	0.0002

	Contrast	Std. err.	[95% conf. interval]	
GATE				
incomecat				
(1 vs 0)	-3258.886	1912.767	-7007.84	490.0682
(2 vs 0)	1513.068	1581.899	-1587.397	4613.534
(3 vs 0)	4995.303	2437.588	217.7184	9772.887
(4 vs 0)	16840.54	4791.758	7448.869	26232.22

The output shows the difference in each group's ATE compared with the lowest income group (level 0). Except for income group 1, we see that the difference in GATEs increases as income level increases, which corresponds with our conjecture that people who earn more benefit more from being eligible for a 401(k).

Similarly, we can estimate the GATES for home ownership. We again specify option `reestimate` to reuse the `cate` estimation results and specify option `group(ownhome)` to estimate the GATES for home ownership categories.

```
. cate, group(ownhome) reestimate
Estimating GATE ...
Conditional average treatment effects      Number of observations      = 9,913
Estimator:      Partialing out              Number of folds in cross-fit = 10
Outcome model:   Linear lasso                 Number of outcome controls  = 47
Treatment model: Logit lasso                  Number of treatment controls = 47
CATE model:      Random forest                 Number of CATE variables    = 17
```

assets	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
GATE						
ownhome						
0	3319.503	795.4385	4.17	0.000	1760.472	4878.534
1	10858.26	1742.672	6.23	0.000	7442.69	14273.84
ATE						
e401k						
(Eligible						
vs						
Not elig..)	8107.563	1144.817	7.08	0.000	5863.763	10351.36
POmean						
e401k						
Not eligi..	13902.88	838.5924	16.58	0.000	12259.27	15546.49

Among people who own a home, being eligible for a 401(k) is expected to increase their net financial assets by \$10,858 compared with the net financial assets if not eligible for a 401(k). This effect is substantially larger than \$8,108, the ATE in the population.

We use `estat gatetest` to test whether the GATES are heterogeneous.

```
. estat gatetest
Group treatment-effects heterogeneity test
H0: Group average treatment effects are homogeneous
( 1) [GATE]0bn.ownhome - [GATE]1.ownhome = 0
      chi2(1) = 15.49
      Prob > chi2 = 0.0001
```

We find evidence that the GATES are not homogeneous.

Finally, we use `contrast` to further quantify the difference of the effects between the groups.

```
. contrast r.ownhome
warning: cannot perform check for estimable functions.
Contrasts of marginal linear predictions
Margins: asbalanced
```

	df	chi2	P>chi2
GATE			
ownhome	1	15.49	0.0001

	Contrast	Std. err.	[95% conf. interval]	
GATE ownhome (Yes vs No)	7538.76	1915.627	3784.201	11293.32

For homeowners, the GATE of 401(k) eligibility on financial assets is \$7,539 more than that for the nonhomeowners.

Example 4: Estimate the ATEs over values of a continuous variable

In [example 3](#), we estimated the ATEs over levels of categorical variables. Sometimes, however, we want to estimate the ATEs over values of a continuous variable. For example, we may want to know how the effects of 401(k) eligibility vary with income (not with income categories).

For a continuous variable Z and a specific value z of variable Z , we are interested in estimating

$$\tau(z) = E\{\text{asset}(1) - \text{asset}(0)|Z = z\}$$

[Semenova and Chernozhukov \(2021\)](#) proposed the use of nonparametric series regression to approximate the function $\tau(z)$. Specifically, they suggest running a series regression of the AIPW scores estimated in `cate` on the variable Z . To do this after `cate`, we use `estat series`. We specify `income` after `estat series` to indicate that we want to estimate the ATEs over values of `income`. We also specify the `graph` option to plot the estimated function. To reduce the impact of outliers (very high incomes) on estimation, we restrict the sample to incomes less than or equal to \$150,000, which is the 99th percentile of incomes in the sample. We also specify the option `knots(5)` to choose five knots in the generated B-spline terms.

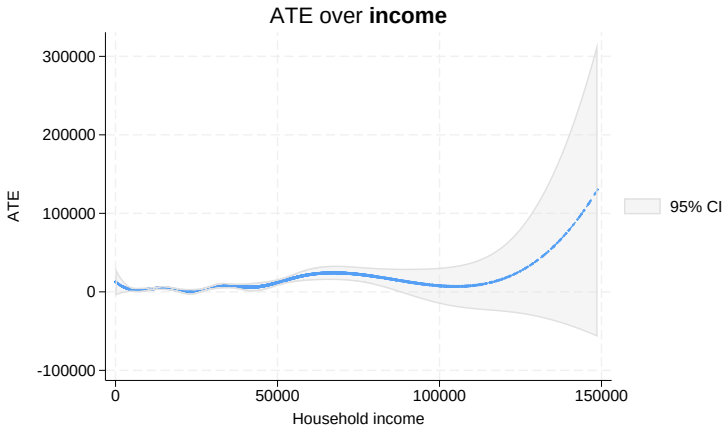
```
. estat series income if income <= 150000, graph knots(5)
Computing approximating function
Computing average derivatives
Nonparametric series regression for IATE
Cubic B-spline estimation      Number of obs      =      9,884
                               Number of knots            =           5
```

	Effect	Robust std. err.	z	P> z	[95% conf. interval]	
income	.1966117	.0521898	3.77	0.000	.0943216	.2989018

Note: Effect estimates are averages of derivatives.

The estimate shows the marginal effect of income on the 401(k) eligibility treatment effects on the net financial assets. Thus, the average marginal effect of a change in income on the treatment effects is \$0.20, indicating that people who earn more benefit more from working in a company with a 401(k) plan.

Because we specified the `graph` option, we obtain the following graph that illustrates how the treatment effect changes with income.



Each point in the plot corresponds to the estimated ATE at a given income level. It also plots the 95% confidence interval. The graph shows an upward trend between income level and the treatment effects, especially at higher income levels. However, the confidence intervals for the ATEs are also wide at the higher income levels. Compared with the `categraph` `gateplot` we used for the income category in [example 3](#), the series graph reveals a more nuanced and nonlinear relationship between income and the treatment effects.

Example 5: Use the AIPW estimator

In addition to the PO estimator in a partial linear model, we can also estimate the IATE function in a fully interactive model by using the AIPW estimator. For example, let's estimate the IATE function as in [example 2](#) but using the AIPW estimator.

The IATE function we want to estimate is

$$\tau(\mathbf{x}) = E\{\text{asset}(1) - \text{asset}(0) | \mathbf{x}\}$$

where `asset(1)` and `asset(0)` are the treated and untreated potential outcomes, respectively. $\mathbf{x}$ are the treatment-effects covariates.

The fully interactive model is

$$\begin{aligned}\text{asset}(1) &= g_1(\mathbf{x}, \mathbf{w}) + \epsilon_1 \\ \text{asset}(0) &= g_0(\mathbf{x}, \mathbf{w}) + \epsilon_2 \\ \text{e401k} &= f(\mathbf{x}, \mathbf{w}_2) + u\end{aligned}$$

where $\mathbf{w}$ and $\mathbf{w}_2$ are vectors of additional control variables for the outcome and treatment models, respectively. By default, $\mathbf{w}_2$ is the same as $\mathbf{w}$, but it can be different if specified. $g_1(\mathbf{x}, \mathbf{w})$ and $g_0(\mathbf{x}, \mathbf{w})$ are the nonparametric models for the treated and untreated potential outcomes, respectively. ϵ_1 and ϵ_2 are the error terms. $f(\mathbf{x}, \mathbf{w}_2)$ is a nonparametric model for the treatment, and u is the error term. The fully interactive model allows the treatment effect to interact with both $\mathbf{x}$ and $\mathbf{w}$; thus, it is more general than the partial linear model. Note that estimating $\tau(\mathbf{x})$ now requires that we also estimate three nuisance functions: $g_1(\mathbf{x}, \mathbf{w})$, $g_0(\mathbf{x}, \mathbf{w})$, and $f(\mathbf{x}, \mathbf{w}_2)$.

We specify `aipw` after `cate` to invoke the AIPW estimator. The first portion of the output is

```
. cate aipw (assets $catecovars) (e401k), rseed(12345671) controls($controls)

Cross-fit fold 1 of 10 ...
Estimating lasso for outcome assets if e401k = 0 ...
Estimating lasso for outcome assets if e401k = 1 ...
Performing lasso for treatment e401k ...

      :
      (output omitted)
      :

Cross-fit fold 10 of 10 ...
Estimating lasso for outcome assets if e401k = 0 ...
Estimating lasso for outcome assets if e401k = 1 ...
Performing lasso for treatment e401k ...
```

The iteration log shows that two potential outcome models and a treatment model are fit using ten-fold cross-fitting. By default, the potential outcome models are estimated using a linear lasso and the treatment model is estimated with a logit lasso.

The remaining portion of the output is

```
Estimating AIPW scores ...
Estimating random forest for IATE ...
Estimating ATE ...

Conditional average treatment effects      Number of observations      = 9,913
Estimator:      Augmented IPW              Number of folds in cross-fit =   10
Outcome model:  Linear lasso                Number of outcome controls  =   47
Treatment model: Logit lasso                Number of treatment controls =   47
CATE model:     Random forest                Number of CATE variables    =   17
```

assets	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
e401k						
(Eligible						
vs						
Not elig..)	8164.364	1151.125	7.09	0.000	5908.2	10420.53
POmean						
e401k						
Not eligi..	13910.87	842.0945	16.52	0.000	12260.39	15561.34

We see that random forest is used to estimate the IATE function once the cross-fitting is finished. Then the AIPW scores implied by the fully interactive model are computed, and the ATE is an average of the AIPW scores. The estimated ATE indicates that if everyone in the population is eligible for a 401(k), the net financial assets will, on average, be \$8,164 more than if no one in the population is eligible for a 401(k). The \$8,164 ATE estimate is similar to the \$8,108 result in [example 2](#).

Both the PO and the AIPW estimators are Neyman orthogonal, implying that the ATE estimation results are robust in response to the machine learning estimation errors in the outcome and the treatment model. However, the AIPW estimator is asymptotically more efficient than the PO estimator (see [Kennedy \[2023\]](#)). In addition, the AIPW estimator enjoys a doubly robust property, meaning that only one of the outcome model or the treatment-assignment model needs to be correctly specified to consistently estimate the ATE (see [Chernozhukov et al. \[2018\]](#)).

Another advantage of the AIPW estimator over the PO estimator is that it allows us to use different control variables in the outcome and the treatment model. For example, suppose we want to add the square of age as an additional control in the treatment-assignment model. In `cate`, we specify the option `treatcontrols($controls c.age#c.age)` to use `c.age#c.age` as the additional control variable in the treatment model.

```
. cate aipw (assets $catecovars) (e401k), rseed(12345671) controls($controls)
> treatcontrols($controls c.age#c.age) nolog
```

Conditional average treatment effects	Number of observations	= 9,913
Estimator: Augmented IPW	Number of folds in cross-fit	= 10
Outcome model: Linear lasso	Number of outcome controls	= 47
Treatment model: Logit lasso	Number of treatment controls	= 48
CATE model: Random forest	Number of CATE variables	= 17

assets	Robust					
	Coefficient	std. err.	z	P> z	[95% conf. interval]	
ATE						
e401k						
(Eligible						
vs						
Not elig..)	8164.364	1151.125	7.09	0.000	5908.2	10420.53
POmean						
e401k						
Not eligi..	13910.87	842.0945	16.52	0.000	12260.39	15561.34

The ATE estimate does not change in this case even though the outcome model has one extra control variable. The results in this example imply that our results are robust to a different control specification.

We can use the same set of postestimation commands to explore, summarize, and visualize the treatment effects after AIPW estimation as we did after PO estimation. The conclusions for our example would be similar; thus, we will not repeat those postestimation commands here.

Example 6: Data-driven group hypothesis testing

In [example 3](#), we summarized the heterogeneous treatment effects by estimating the ATEs for the prespecified groups of income category and home ownership. We emphasize that these group variables must be prespecified before the data collection and analysis to avoid *p*-value hacking as discussed in [Head et al. \(2015\)](#). This scenario is suitable when researchers know a priori across which groups they would like to explore treatment-effect heterogeneity.

However, we sometimes do not know which variable is linked to the heterogeneity of treatment effects; we want the data to discover these variables. [Chernozhukov et al. \(2006\)](#) suggest ranking the treatment effects first and then performing a classification analysis based on the groups induced by the treatment-effects ranking.

For example, suppose we estimate the IATE of 401(k) eligibility on net financial assets for each observation, and we divide the data into four groups based on the IATE prediction's ranking. The first group is the people in the top 25% of the treatment effects in the data, and the last group is the people in the bottom 25% of the treatment effects in the data. We want to know whether the mean income differs for the groups with the largest and the smallest treatment effects.

In `cate`, we specify the `group(#)` option to rank the data based on the estimated IATEs. In the following example, we specify `group(4)` to divide the data into four groups based on the IATE ranking. We also specify the option `xfolds(5)` to use five-fold cross-fitting. The first portion of the output is

```
. cate po (assets $catecovars) (e401k), rseed(12345671) controls($controls)
> group(4) xfolds(5)
Cross-fit fold 1 of 5 ...
Performing lasso for outcome assets ...
Performing lasso for treatment e401k ...
Estimating IATE rankings ...
Estimating AIPW scores ...
:
(output omitted)
:
Cross-fit fold 5 of 5 ...
Performing lasso for outcome assets ...
Performing lasso for treatment e401k ...
Estimating IATE rankings ...
Estimating AIPW scores ...
```

The iteration logs show that the IATE rankings are computed using cross-fitting. This procedure is necessary because it avoids the overfitting issues by using one sample to estimate the IATE function and a different sample to predict the IATE function.

The remaining output is as follows:

```
Performing random forest for IATE ...
Estimating AIPW scores ...
Estimating sorted GATE ...
Conditional average treatment effects      Number of observations      = 9,913
Estimator:      Partialing out              Number of folds in cross-fit =    5
Outcome model:  Linear lasso                 Number of outcome controls  =   47
Treatment model: Logit lasso                 Number of treatment controls =   47
CATE model:     Random forest                Number of CATE variables   =   17
```

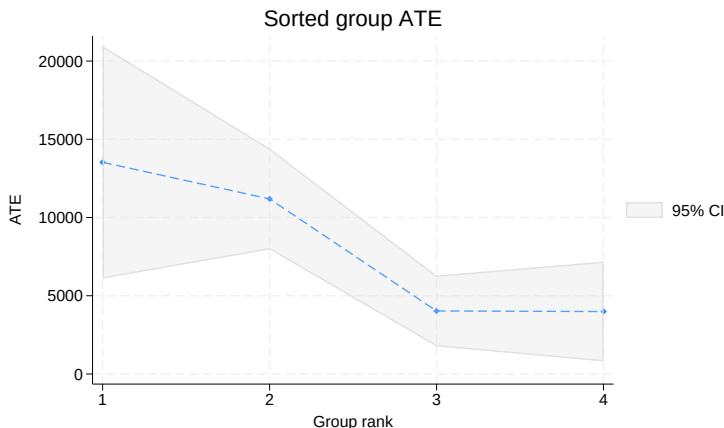
assets	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
GATES						
rank						
1	13529.88	3792.728	3.57	0.000	6096.266	20963.49
2	11190.14	1646.548	6.80	0.000	7962.962	14417.31
3	4026.967	1154.876	3.49	0.000	1763.452	6290.481
4	3993.897	1627.408	2.45	0.014	804.236	7183.558
ATE						
e401k (Eligible vs Not elig..)	8183.327	1148.204	7.13	0.000	5932.888	10433.77
POmean						
e401k Not eligi..	13881.65	840.706	16.51	0.000	12233.89	15529.4

The remaining output shows the ATEs for the groups sorted by IATE predictions, also known as the GATESs. The GATES for the group with the largest treatment effects is \$13,530. The results show a substantial difference of GATESs estimates between the most and the least affected groups.

In some cases, however, the highest-ranking group does not necessarily have a greater GATES estimate than the lowest-ranking group because the rankings are generated using cross-fitting to avoid overfitting. The rankings depend on the IATE estimates' quantiles in each cross-fitting fold but not the whole sample. Thus, an observation with a higher ranking implies only that this observation has greater IATE estimates compared with other observations in a particular fold; it does not necessarily mean that it has greater IATE estimates compared with the full sample. If the treatment effects are genuinely homogeneous, we would observe GATES estimates that are similar across ranking levels. See [Methods and formulas](#) for details.

We can visualize the GATES estimates using `categraph gateplot`.

```
. categraph gateplot
```



To test whether the treatment effects are homogeneous across the group, we use `estat gatetest`. We specify levels 1 and 4 to compare the groups with the largest and smallest effects.

```
. estat gatetest 1 4
Sorted group treatment-effects heterogeneity test
H0: Sorted group average treatment effects are homogeneous
( 1) [GATES]1bn.rank - [GATES]4.rank = 0
      chi2(1) = 5.34
      Prob > chi2 = 0.0209
```

We find evidence that the GATES are not homogeneous across the groups with the largest and smallest effects.

We can use `estat classification` to test whether the means of some variables are different between the groups with the largest and smallest effects. For example, we can test whether the income level has different means in groups ranked 1 and 4.

```
. estat classification income
Classification t test with equal variances
```

Group	Obs	Mean	Std. err.	Std. dev.	[95% conf. interval]	
1	2,480	62522.61	513.3117	25562.71	61516.05	63529.17
4	2,475	26420.44	367.1063	18263.31	25700.57	27140.31
Combined	4,955	44489.74	406.6722	28626.37	43692.48	45287
diff		36102.17	631.2817		34864.58	37339.76

```

diff = mean(1) - mean(4)                                t = 57.1887
HO: diff = 0                                             Degrees of freedom = 4953
Ha: diff < 0                                           Ha: diff != 0
Pr(T < t) = 1.0000                                Pr(|T| > |t|) = 0.0000      Ha: diff > 0
Pr(T > t) = 0.0000

```

The income levels are higher in the group with the largest effects than in the group with the smallest effects.

We can do a similar classification analysis for `age` and `ownhome`.

```
. estat classification ownhome
Classification t test with equal variances
```

Group	Obs	Mean	Std. err.	Std. dev.	[95% conf. interval]	
1	2,480	.8584677	.0070009	.3486401	.8447396	.8721959
4	2,475	.4056566	.0098718	.4911179	.3862986	.4250145
Combined	4,955	.6322906	.0068507	.4822304	.6188603	.645721
diff		.4528112	.0120983		.4290932	.4765291

```

diff = mean(1) - mean(4)                                t = 37.4278
HO: diff = 0                                             Degrees of freedom = 4953
Ha: diff < 0                                           Ha: diff != 0
Pr(T < t) = 1.0000                                Pr(|T| > |t|) = 0.0000      Ha: diff > 0
Pr(T > t) = 0.0000

```

```
. estat classification age
Classification t test with equal variances
```

Group	Obs	Mean	Std. err.	Std. dev.	[95% conf. interval]	
1	2,480	45.16815	.1808494	9.006225	44.81351	45.52278
4	2,475	35.26747	.2175367	10.82231	34.8409	35.69405
Combined	4,955	40.22281	.1579318	11.1171	39.91319	40.53242
diff		9.90067	.2828416		9.346176	10.45517

```

diff = mean(1) - mean(4)                                t = 35.0043
HO: diff = 0                                             Degrees of freedom = 4953
Ha: diff < 0                                           Ha: diff != 0
Pr(T < t) = 1.0000                                Pr(|T| > |t|) = 0.0000      Ha: diff > 0
Pr(T > t) = 0.0000

```

More people in group 1 own homes than in group 4, and people are older in group 1 than people in group 4.

The analysis suggests that individuals who are older, have a home, and have a higher income see a more substantial effect of 401(k) eligibility on net financial assets. Without any ex-ante assumptions about the effect of eligibility, we can learn from the data which subpopulations defined by covariates would benefit more with greater access to a 401(k).

Finally, we can specify the group(#) option with reestimate if we want to divide the data into a different number of ranking groups but want to avoid recomputing the IATE function. It is much faster than estimating GATESS from scratch. Here, we specify the group(2) option to divide the data into two ranking levels.

```
. cate, group(2) reestimate
Estimating sorted GATE ...
Conditional average treatment effects   Number of observations   = 9,913
Estimator:          Partialing out      Number of folds in cross-fit =    5
Outcome model:      Linear lasso         Number of outcome controls  =   47
Treatment model:    Logit lasso          Number of treatment controls =   47
CATE model:         Random forest        Number of CATE variables    =   17
```

assets	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
GATES						
rank						
1	12360.48	2067.996	5.98	0.000	8307.28	16413.68
2	4010.449	997.4399	4.02	0.000	2055.502	5965.395
ATE						
e401k (Eligible vs Not elig..)	8183.327	1148.204	7.13	0.000	5932.888	10433.77
POmean						
e401k Not eligi..	13881.65	840.706	16.51	0.000	12233.89	15529.4

Example 7: Flexible models

In the CATE estimation, we need to specify the estimation methods in three different models: the outcome model, the treatment-assignment model, and the CATE model. The outcome and treatment models are the nuisance parameters, which we are not interested in making inferences about. The CATE model is the object of interest, and we want to make inferences on IATES, ATES, GATES, and GATESS.

In the previous examples, we used the lasso linear model for the outcome model, the lasso logit model for the treatment model, and the random forest for the IATE function $\tau(\mathbf{x})$, the default in cate. Sometimes, however, we want to use different techniques to explore data based on different assumptions. For the outcome and treatment models, we can use a semi-parametric method such as lasso, a nonparametric method such as random forest, or a purely parametric method such as linear or logistic regression. For the IATE function, we can use either a nonparametric method, such as random forest, or a parametric method, such as linear regression. We can try different models to see how sensitive the results are to the modeling methods and assumptions.

In the case of our 401(k) eligibility example, we will try the parametric models, which are estimated using parametric methods, and the nonparametric models, which are estimated using random forest.

Parametric models like linear regression are easier to compute and interpret than nonparametric models such as a random forest. However, the assumptions of a parametric model are less likely to be satisfied. Nevertheless, we can use it as a benchmark. For example, suppose we assume a linear model for the outcome, a logit model for the treatment, and the linear model for the IATE. Under these assumptions, the IATE function is

$$\tau(\mathbf{x}) = \mathbf{x}'\beta$$

Thus, the outcome and treatment models under these parametric assumptions become

$$\begin{aligned} \text{asset} &= \text{e401k} * (\mathbf{x}'\beta) + \mathbf{x}'\gamma_1 + \epsilon \\ \Pr(\text{e401k} = 1|\mathbf{x}) &= \frac{\exp(\mathbf{x}'\gamma_2)}{1 + \exp(\mathbf{x}'\gamma_2)} \end{aligned}$$

In cate, we specify the `omethod(regress)` option to use linear regression for the outcome, the `tmethod(logit)` option to use a logit model for the treatment, and the `cmethod(regress)` option to use linear regression for the IATE function.

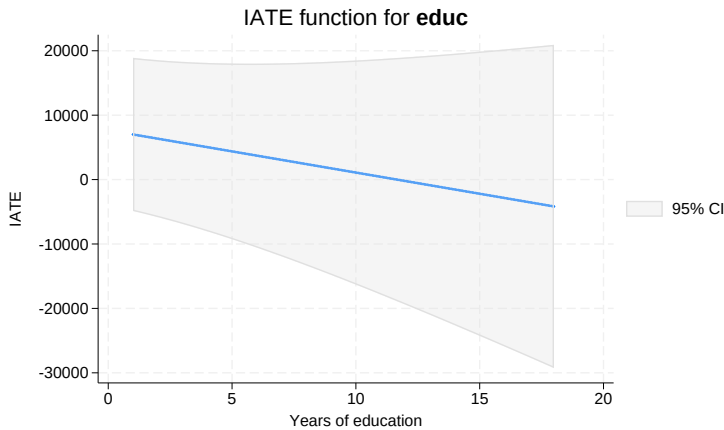
```
. cate po (assets $catecovars) (e401k), rseed(12345671)
> omethod(regress) tmethod(logit) cmethod(regress) nolog
Conditional average treatment effects      Number of observations      = 9,913
Estimator:      Partialing out              Number of folds in cross-fit =   10
Outcome model:  Linear regression            Number of outcome controls  =   17
Treatment model: Logit                      Number of treatment controls =   17
CATE model:     Linear regression            Number of CATE variables    =   17
```

assets	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE e401k (Eligible vs Not elig..)	7904.218	1155.565	6.84	0.000	5639.351	10169.08
POmean e401k Not eligi..	13977.45	831.0932	16.82	0.000	12348.54	15606.37

We can interpret the results as before. If we plot the IATE function over one variable, such as `educ`, we see a straight line, which is expected due to the parametric assumption on the IATE function $\tau(\mathbf{x})$.

```
. categraph iateplot educ
Note: IATE estimated at fixed values of covariates other than educ.
```

Variable	Statistic	Value	Type
age	mean	41.05891	continuous
incomecat	base	0	factor
ira	base	0	factor
married	base	0	factor
ownhome	base	0	factor
pension	base	0	factor
twoearn	base	0	factor



Interestingly, the graph shows a downward trend between the years of education and the IATEs, and the estimates of the IATEs are not different from zero (the confidence intervals include zero for all education levels). This conclusion is very different from that in examples 1 and 2 when we use the random forest to estimate the IATE function. It may imply that the parametric assumptions on the IATE function are too strong.

In contrast to the pure parametric model, we can use the random forest in the outcome, the treatment, and the IATE models. Random forest allows us to model flexibly without imposing restrictive assumptions. Another advantage is that we can use the out-of-bag predictions from the random forest to avoid using cross-fitting, which may be time consuming.

We specify the `omethod(rforest)` and `tmethod(rforest)` options to use the random forest model for both the outcome and the treatment models. The default method for the IATE estimation is already a random forest, so we do not need to specify the `cmethod()` option here. In addition, to use the out-of-bag prediction-based algorithm, we specify the `oob` option.

```
. cate po (assets $catecovars) (e401k), rseed(12345671)
> omethod(rforest) tmethod(rforest) oob

Performing random forest for outcome assets ...
Performing random forest for treatment e401k ...
Performing random forest for IATE ...
Estimating AIPW scores ...
Estimating ATE ...

Conditional average treatment effects      Number of observations      = 9,913
Estimator:      Partialing out              Number of folds in cross-fit =    1
Outcome model:  Random forest                Number of outcome controls  =   17
Treatment model: Random forest                Number of treatment controls =   17
CATE model:     Random forest                 Number of CATE variables    =   17
```

assets	Robust		z	P> z	[95% conf. interval]	
Coefficient	std. err.					
ATE						
e401k						
(Eligible						
vs						
Not elig..)	8225.258	1173.862	7.01	0.000	5924.53	10525.99
POmean						
e401k						
Not eligi..	14016.34	850.1257	16.49	0.000	12350.13	15682.56

The results are a little different from those in examples 1 and 2. This is expected because, in examples 1 and 2, we used the lasso methods for the outcome and treatment models.

We can test whether the treatment effects are heterogeneous using estat heterogeneity.

```
. estat heterogeneity

Treatment-effects heterogeneity test
HO: Treatment effects are homogeneous

      chi2(1) =   4.07
Prob > chi2 = 0.0437
```

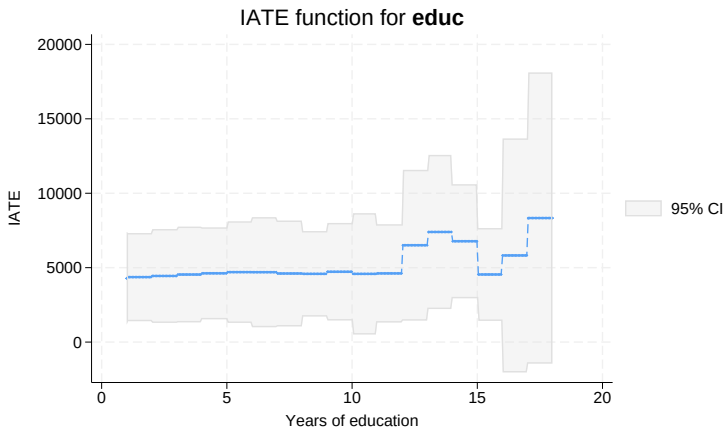
We find evidence that the treatment effects are not homogeneous.

To compare with the parametric model, we can also plot the IATE function with respect to education (educ).

```
. categraph iateplot educ

Note: IATE estimated at fixed values of covariates other than educ.
```

Variable	Statistic	Value	Type
age	mean	41.05891	continuous
incomecat	base	0	factor
ira	base	0	factor
married	base	0	factor
ownhome	base	0	factor
pension	base	0	factor
twoearn	base	0	factor



Compared with the parametric IATE function, the graph shows a nonlinear form, and the treatment effects are more substantial for people with 12 to 15 years of education while holding other variables fixed.

The random forest method puts less restrictive assumptions on the data generating process (DGP) than the lasso method, and thus, random forest may model the real-world data better. However, the caveats are that the random forest can not handle high-dimensional controls like lasso, and it takes much longer to compute when there are many observations in the data. In contrast, lasso can be considered a semi-parametric model by approximating a function using a set of basis functions. It can approximate the DGP reasonably well when the underlying function is sparse, meaning only a few terms among high-dimensional controls have nonzero coefficients.

Example 8: Treatment-assignment policy evaluation

In examples 1 to 7, we use `cate` to study the treatment-effects heterogeneity from different perspectives, such as IATES, GATES, and GATESS. Sometimes, researchers are not interested in the treatment-effects heterogeneity itself but instead want to use the estimated treatment effects to evaluate a treatment-assignment policy.

We want to evaluate the policy by answering questions such as the following:

1. If we implement such a policy, what is the average outcome of the population?
2. If we have alternative policies, which one is better?

For the first question, we compute the average outcome if the treatment is assigned according to the policy, which is also known as the policy's value. Precisely, the value of the treatment-assignment policy, $\Pi(\pi)$, is defined as

$$\Pi(\pi) = E\{\pi_i y_i(1) + (1 - \pi_i) y_i(0)\}$$

where $\pi_i \in [0, 1]$ is a prespecified treatment-assignment probability for the i th observation. Thus, π_i is referred to as the policy. $y_i(1)$ and $y_i(0)$ are the potential outcomes for being treated or not treated, respectively.

For the second question, we compute the value difference between the two policies,

$$\Pi(\pi_A) - \Pi(\pi_B)$$

where π_A and π_B are two different treatment-assignment policies.

We illustrate how to use `cate` to evaluate a hypothetical treatment-assignment policy that assigns patients to two types of lung transplants. Bilateral lung transplant (BLT) is usually associated with a higher death rate in the short term after the operation but with a more significant improvement in the quality of life compared with a single lung transplant (SLT). Suppose a doctor has a simple treatment-assignment rule, which assigns a patient to BLT if the patient's walking distance is greater than 500 meters in six minutes and if the patient does not have diabetes. The doctor wants to evaluate this policy by answering the following two questions:

1. What would be the average outcome if this policy is implemented?
2. Is this policy better than the treatment assignment observed in the data?

We have a fictional dataset (`lung.dta`) inspired by [Koch, Vock, and Wolfson \(2018\)](#). An individual's forced expiratory volume in one second (FEV1) measures a patient's quality of life. The outcome of interest is the percentage of FEV1 that a patient has relative to a healthy person with similar characteristics, FEV1% (`fev1p`), measured one year after the operation. The treatment variable (`transtype`) indicates whether the treatment is BLT or SLT.

To open the dataset and describe it, we type

```
. use https://www.stata-press.com/data/r19/lung, clear
(Fictional data on lung transplant)
. describe *, short
```

Variable name	Storage type	Display format	Value label	Variable label
agep	byte	%10.0g		Patient age (years)
bmip	double	%10.0g		Patient body mass index
diabetesp	byte	%12.0g	lbdiaab	Patient diabetes status
heightp	double	%10.0g		Patient height (cm)
o2amt	double	%10.0g		Oxygen delivered
karn	byte	%8.0g	lbyes	Karnofsky score > 60
lungals	double	%10.0g		Lung allocation score
racep	byte	%8.0g	lbrace	Patient race
sexp	byte	%8.0g	lbsex	Patient gender
lifesvent	byte	%8.0g	lbyes	Life support ventilator needed
assisvent	byte	%8.0g	lbyes	Assisted ventilation needed
centervol	double	%10.0g		Center volume
walkdist	double	%10.0g		Walking distance in 6 minutes
o2rest	byte	%8.0g	lbyes	Oxygen needed at rest
aged	byte	%10.0g		Donor age (years)
raced	byte	%8.0g	lbrace	Donor race
bmip	double	%10.0g		Donor body mass index
smoked	byte	%8.0g	lbyes	Donor if has history of smoking
cmv	byte	%8.0g	lbyes	Positive cytomegalovirus test
deathcause	byte	%8.0g	lbyes	Cause of death - traumatic brain injury
diabetesd	byte	%12.0g	lbdiaab	Donor diabetes status
expandd	byte	%8.0g	lbyes	Expanded donor needed
heightd	double	%10.0g		Donor height (cm)
sexd	byte	%8.0g	lbsex	Donor gender
distd	int	%10.0g		Donor to treatment center distance
lungpo2	double	%10.0g		Lung P02
lungalloc	byte	%8.0g	lballo	Lung allocation status
hratio	double	%10.0g		Height ratio
ischemict	double	%10.0g		Ischemic time
genderm	byte	%19.0g	lbgm	Matching gender status
racem	byte	%17.0g	lbrm	Matching race status
transtype	byte	%8.0g	lbttau	Lung transplant type
fev1p	double	%10.0g		Percentage of predicted value of FEV1

Thirty-one variables measure characteristics of the patients and donors. To construct *catevarlist* and the control variables, we want to use these variables and the interactions among them. The following commands create global macro *catecovars* to represent the covariates in the IATE function and global macro controls to represent the additional control variables in the outcome and treatment models.

```
. global cvars bmip heightp o2amt lungals centervol walkdist bmip heightd
> distd lungpo2 hratio ischemict
. global fvars diabetesp karn racep sexp lifesvent assisvent o2rest raced
> smoked cmv deathcause diabetesd expandd sexd lungalloc genderm racem
. global catecovars c.($cvars) i.($fvars)
. global controls c.($cvars)#i.($fvars)
```

The hypothetical policy variable `policy1` assigns a patient to BLT if the patient’s walking distance is more than 500 meters and if the patient does not have diabetes.

```
. generate policy1 = walkdist > 500 & !diabetesp & !missing(walkdist)
```

To evaluate `policy1`, we first need to use `cate` and estimate the potential outcomes for each individual.

```
. cate aipw (fev1p $catecovars) (transtype), rseed(12345671) controls($controls)
> nolog
Conditional average treatment effects      Number of observations      = 937
Estimator:      Augmented IPW              Number of folds in cross-fit = 10
Outcome model:   Linear lasso               Number of outcome controls  = 454
Treatment model: Logit lasso               Number of treatment controls = 454
CATE model:      Random forest              Number of CATE variables    = 46
```

fev1p	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE transtype (BLT vs SLT)	37.5243	.1646795	227.86	0.000	37.20153	37.84707
POmean transtype SLT	46.49502	.2025403	229.56	0.000	46.09805	46.892

The FEV1% if all the patients were to be assigned to BLT is expected to be 38 percentage points higher than the 46% average expected if all patients were to be assigned to SLT.

The ATE is a special version of policy evaluation. The ATE estimates the difference in average outcomes between the two policies: everyone is treated versus everyone is untreated. Here is an illustration.

We create the variable `treatall` representing a policy that assigns each patient to BLT treatment. In contrast, variable `treatnone` represents a policy that assigns each patient to SLT treatment.

```
. generate treatall = 1
. generate treatnone = 0
```

We can use `estat policyeval` to compare these treatment policies.

```
. estat policyeval treatall treatnone
```

Treatment-assignment policy evaluation					Number of obs = 937	
	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
Value						
policy						
treatall	84.01932	.3085432	272.31	0.000	83.41459	84.62406
treatnone	46.49502	.2025403	229.56	0.000	46.09805	46.892
Contrast						
policy						
(treatall						
vs						
treatnone)	37.5243	.1646795	227.86	0.000	37.20153	37.84707

The value of a policy is the average outcome if the policy is implemented. For example, the value of `treatnone` is 46.5, which means that the expected FEV1% is 46.5% if all patients are assigned to SLT. By definition, the value of `treatnone` corresponds to the potential outcome mean if the treatment status is SLT, and in the `cate` output, the `P0mean` for SLT is indeed 46.5. That is,

$$\Pi(\text{treatnone}) = E\{0 * \text{fev1p}(1) + 1 * \text{fev1p}(0)\} = E\{\text{fev1p}(0)\}$$

Similarly, the value of `treatall` is the potential outcome if all patients are assigned to BLT. That is,

$$\Pi(\text{treatall}) = E\{1 * \text{fev1p}(1) + 0 * \text{fev1p}(0)\} = E\{\text{fev1p}(1)\}$$

The contrast is the difference in the values of the two policies. The contrast between `treatall` and `treatnone` is 37.5, which means that the FEV1% is 37.5% higher if all patients are assigned to BLT over SLT. By definition, the contrast between `treatall` and `treatnone` is the ATE, and the ATE estimate in the `cate` output is indeed 37.5. That is,

$$\Pi(\text{treatall}) - \Pi(\text{treatnone}) = E\{\text{fev1p}(1) - \text{fev1p}(0)\} = \text{ATE}$$

Now we are ready to evaluate the hypothetical policy `policy1`, which assigns a patient to BLT if the patient's walking distance is greater than 500 meters and the patient does not have diabetes.

```
. estat policyeval policy1
```

Treatment-assignment policy evaluation					Number of obs = 937	
	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
Value						
policy						
policy1	72.66426	.714435	101.71	0.000	71.26399	74.06452

The average FEV1% is 72.6% if this policy is implemented. We can compare this with the observed treatment policy (transtype).

```
. estat policyeval policy1 transtype
```

Treatment-assignment policy evaluation

Number of obs = 937

	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
Value						
policy	72.66426	.714435	101.71	0.000	71.26399	74.06452
policy1	66.53891	.5149955	129.20	0.000	65.52954	67.54828
transtype						
Contrast						
policy						
(policy1						
vs						
transtype)	6.125348	.9130896	6.71	0.000	4.335725	7.91497

The average FEV1% is 6.1% higher if policy1 is implemented compared with the actual treatment-assignment policy transtype.

Stored results

cate stores the following in e():

Scalars

e(N)	number of observations
e(n_xfolds)	number of folds for cross-fitting
e(k_controls_om)	number of controls in the outcome model
e(k_controls_tm)	number of controls in the treatment model
e(k_cate_covars)	number of covariates in the CATE model
e(gateslevel)	number of sub-levels in the data-driven group if group(#) specified
e(samprate_om)	sampling rate for outcome model, if omethod(rforest) specified
e(ntrees_om)	number of trees for outcome model, if omethod(rforest) specified
e(cintrees_om)	number of trees in each group for outcome model, if omethod(rforest) specified
e(splitminobs_om)	minimum number of observations to split a node for outcome model, if omethod(rforest) specified
e(splitmeanvars_om)	mean number of variables to be split in each node for outcome model, if omethod(rforest) specified
e(honestrate_om)	sampling rate for honest tree for outcome model, if omethod(rforest) specified
e(samprate_tm)	sampling rate for treatment model, if tmethod(rforest) specified
e(ntrees_tm)	number of trees for treatment model, if tmethod(rforest) specified
e(cintrees_tm)	number of trees in each group for treatment model, if tmethod(rforest) specified
e(splitminobs_tm)	minimum number of observations to split a node for treatment model, if tmethod(rforest) specified
e(splitmeanvars_tm)	mean number of variables to be split in each node for treatment model, if tmethod(rforest) specified
e(honestrate_tm)	sampling rate for honest tree for treatment model, if tmethod(rforest) specified
e(samprate_cm)	sampling rate for CATE model, if cmethod(rforest) specified
e(ntrees_cm)	number of trees for CATE model, if cmethod(rforest) specified
e(cintrees_cm)	number of trees in each bag for CATE model, if cmethod(rforest) specified
e(splitminobs_cm)	minimum number of observations to split a node for CATE model, if cmethod(rforest) specified
e(splitmeanvars_cm)	mean number of variables to be split in each node for CATE model, if cmethod(rforest) specified
e(honestrate_cm)	sampling rate for honest tree for CATE model, if cmethod(rforest) specified

Macros

<code>e(cmd)</code>	<code>cate</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(gate_var)</code>	GATE variable
<code>e(cate_covars)</code>	name of CATE covariates
<code>e(covars_om)</code>	covariates for outcome model
<code>e(covars_tm)</code>	covariates for treatment model
<code>e(lasso_selection_om)</code>	lasso selection method for outcome model
<code>e(lasso_selection_tm)</code>	lasso selection method for treatment model
<code>e(estimator)</code>	name of estimator
<code>e(omethod)</code>	estimation method for outcome model
<code>e(tmmethod)</code>	estimation method for treatment model
<code>e(cmethod)</code>	estimation method for CATE model
<code>e(title)</code>	title in estimation output
<code>e(predict)</code>	program used to implement predict
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(rngstate)</code>	random-number state used
<code>e(properties)</code>	b V
<code>e(marginsnotok)</code>	predictions disallowed by margins

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals
-----------------------	--

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

The methods and formulas are presented under the following headings:

- PO for the partial linear model*
 - PO IATE estimator*
 - PO GATE estimator with prespecified groups*
 - PO GATES estimator with data-driven groups*
- AIPW for the fully interactive model*
 - AIPW IATE estimator*
 - AIPW GATE estimator with prespecified groups*
 - AIPW GATES estimator with data-driven groups*
- Generalized random forest*
 - Honest tree*
 - Honest random forest*
 - Confidence intervals*
 - Missing values*

We can group the estimation methods by the outcome model. The outcome model can be expressed as a partial linear or fully interactive model. For the partial linear model, the estimation method is the PO estimator proposed in [Nie and Wager \(2021\)](#). The fully interactive model's estimation methods are

built around the AIPW estimator discussed in [Knaus \(2022\)](#) and [Kennedy \(2023\)](#). We will briefly discuss the features of these two estimators and compare them. For notational simplicity, we drop the subscript i indicating the i th observation to refer to a random variable.

PO for the partial linear model

The partial linear model is

$$\begin{aligned} y &= d * \tau(\mathbf{x}) + g(\mathbf{x}, \mathbf{w}) + \epsilon \\ d &= f(\mathbf{x}, \mathbf{w}) + u \end{aligned}$$

where y is the outcome variable, d is a binary treatment, $\mathbf{x}$ is a vector of covariates for the IATE function, $\mathbf{w}$ are optional controls for both the outcome and the treatment models, and $g(\mathbf{x}, \mathbf{w})$ and $f(\mathbf{x}, \mathbf{w})$ are nuisance functions for the outcome and the treatment, respectively. ϵ and u are the error terms.

Let $y(1)$ be the potential outcome when the unit is treated and $y(0)$ be the potential outcome when the unit is not treated. By definition, the potential outcomes are

$$\begin{aligned} y(1) &= \tau(\mathbf{x}) + g(\mathbf{x}, \mathbf{w}) + \epsilon \\ y(0) &= g(\mathbf{x}, \mathbf{w}) + \epsilon \end{aligned}$$

Thus, the IATE function is $\tau(\mathbf{x}) = E\{y(1) - y(0)|\mathbf{x}\}$.

To estimate $\tau(\mathbf{x})$, we need to partial-out the nuisance functions $g(\mathbf{x}, \mathbf{w})$ and $f(\mathbf{x}, \mathbf{w})$. To do this, we need to construct residuals of the outcome and the treatment that are independent of $\mathbf{w}$ and $\mathbf{x}$. First, we take the expectation of y conditional on $\mathbf{x}$ and $\mathbf{w}$. That is,

$$E(y|\mathbf{x}, \mathbf{w}) = f(\mathbf{x}, \mathbf{w})\tau(\mathbf{x}) + g(\mathbf{x}, \mathbf{w})$$

Note that unconfoundedness implies that $E(\epsilon|\mathbf{x}, \mathbf{w}) = E(u|\mathbf{x}, \mathbf{w}) = 0$, and therefore we have $E(d|\mathbf{x}, \mathbf{w}) = f(\mathbf{x}, \mathbf{w})$. Thus, subtracting $E(y|\mathbf{x}, \mathbf{w})$ from y removes the term $g(\mathbf{x}, \mathbf{w})$.

$$y - E(y|\mathbf{x}, \mathbf{w}) = \{d - f(\mathbf{x}, \mathbf{w})\}\tau(\mathbf{x}) + \epsilon$$

We can estimate $E(y|\mathbf{x}, \mathbf{w})$ and $f(\mathbf{x}, \mathbf{w})$ via lasso, random forest, or parametric regression. Then we use the residuals to estimate $\tau(\mathbf{x})$. Let $h(\mathbf{x}, \mathbf{w}) = E(y|\mathbf{x}, \mathbf{w})$, and let $\hat{h}(\mathbf{x}, \mathbf{w})$ and $\hat{f}(\mathbf{x}, \mathbf{w})$ be the estimates for $h(\mathbf{x}, \mathbf{w})$ and $f(\mathbf{x}, \mathbf{w})$, respectively. We construct the PO version of y and d as

$$\begin{aligned} \tilde{y} &= y - \hat{h}(\mathbf{x}, \mathbf{w}) \\ \tilde{d} &= d - \hat{f}(\mathbf{x}, \mathbf{w}) \end{aligned}$$

One way we can estimate $\tau(\mathbf{x})$ is by solving the local moment condition

$$\sum_{i=1}^N [\alpha(x_i) \tilde{d}_i \{\tilde{y}_i - \tilde{d}_i \tau(\mathbf{x})\}] = 0$$

where $\alpha(x_i)$ defines the local weights that attach more weight to observations that are close to $\mathbf{x}$. We use the generalized random forest proposed in [Athey, Tibshirani, and Wager \(2019\)](#) to solve this moment condition. For details on causal forest, see [Generalized random forest](#).

Another way to estimate $\tau(\mathbf{x})$ is by assuming a linear functional form for the CATE, $\tau(\mathbf{x}) = \mathbf{x}\beta$, and estimating β using linear regression.

Next we discuss the PO estimator for IATES, GATES, and GATESs.

PO IATE estimator

In practice, the partialled-out residuals $\tilde{d}$ and $\tilde{y}$ are constructed using the out-of-sample prediction. In particular, $\hat{h}(\mathbf{x}_i, \mathbf{w}_i) = \hat{h}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i)$ is the out-of-sample prediction of y_i , with $\hat{h}^{(-i)}(\mathbf{x}, \mathbf{w})$ estimated using data that exclude observation i . Similarly, $\hat{f}(\mathbf{x}_i, \mathbf{w}_i) = \hat{f}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i)$ is the out-of-sample prediction of d_i .

The residuals are based on the out-of-sample prediction.

$$\begin{aligned}\tilde{y}_i^{(-i)} &= y_i - \hat{h}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i) \\ \tilde{d}_i^{(-i)} &= d_i - \hat{f}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i)\end{aligned}$$

Thus, in practice, we solve the moment condition

$$\sum_{i=1}^N \left[\alpha(x_i) \tilde{d}_i^{(-i)} \left\{ \tilde{y}_i^{(-i)} - \tilde{d}_i^{(-i)} \tau(\mathbf{x}) \right\} \right] = 0$$

We can construct the out-of-sample prediction using the cross-fitting technique. We split the data into K folds, define the main sample as the observations in the k th fold, and define the auxiliary sample as the observations not in the k th fold. We estimate all the nuisance functions using the auxiliary sample and get the out-of-sample predictions in the main sample. After circulating through all the folds, we eventually compute the out-of-sample predictions for the full sample. For details of the cross-fitting version of the PO estimator for the IATE, see algorithm 1 below.

Algorithm 1: PO for the IATE using cross-fitting

1. Define the input.
 - (a) Set the number of cross-fitting folds K .
 - (b) Set the estimation method for the outcome model. It can be one of random forest, lasso, square-root lasso, or linear regression.
 - (c) Set the estimation method for the treatment model. It can be one of probability forest, lasso, probit, or logit.
2. Do the cross-fitting to construct the residuals.
 - (a) Randomly split the data into K folds.
 - (b) For each fold $k = 1$ to K , do the following:
 - i. Define the main sample S^M as the observations in the k th fold and the auxiliary sample S^A as the observations not in the k th fold.
 - ii. Construct outcome residuals.
 - A. Using the auxiliary sample S^A , train the outcome model $\hat{h}^A(\mathbf{x}, \mathbf{w})$.

- B. Based on $\hat{h}^A(\mathbf{x}, \mathbf{w})$, predict the outcome in the main sample S^M . Denote the prediction as $\hat{y}^M$.
- C. Compute outcome residuals in the main sample $\tilde{y}^M = y - \hat{y}^M$.
- iii. Construct treatment residuals.
 - A. Using the auxiliary sample S^A , train the treatment model $\hat{f}^A(\mathbf{x}, \mathbf{w})$.
 - B. Based on $\hat{f}^A(\mathbf{x}, \mathbf{w})$, predict the outcome in the main sample S^M . Denote the prediction as $\hat{d}^M$.
 - C. Compute treatment residuals in the main sample $\tilde{d}^M = d - \hat{d}^M$.
- 3. Using the full sample, estimate the function $\tau(\mathbf{x})$ via linear regression or via generalized random forest as in [Athey, Tibshirani, and Wager \(2019\)](#):

$$\sum_{i=1}^N \left[\alpha(x_i) \tilde{d}_i^{(-i)} \left\{ \tilde{y}_i^{(-i)} - \tilde{d}_i^{(-i)} \tau(\mathbf{x}) \right\} \right] = 0$$

Cross-fitting can be applied to generic machine learning techniques. However, by construction, it is computationally demanding.

When the outcome and treatment models are estimated using random forest, we can use a particular case of cross-fitting that saves computational time. That is, we can use the out-of-bag predictions to construct the residuals. The out-of-bag prediction for an observation is constructed using only the trees in the random forest that do not contain this observation. See [Generalized random forest](#) for more details. This procedure is equivalent to cross-fitting but has a faster computation time. For details of the out-of-bag prediction-based PO estimator for the IATE, see algorithm 2 below.

Algorithm 2: PO for the IATE using out-of-bag prediction

1. Construct outcome residuals.
 - (a) Use the full sample, fit a regression forest for the outcome model, and denote it as $\hat{h}(\mathbf{x}, \mathbf{w})$.
 - (b) Based on $\hat{h}(\mathbf{x}, \mathbf{w})$, compute the out-of-bag prediction for the full sample, and denote it as $\hat{y}^{(\text{oob})}$.
 - (c) Compute the outcome residuals for the full sample: $\tilde{y} = y - \hat{y}^{(\text{oob})}$.
2. Construct treatment residuals.
 - (a) Use the full sample, fit a probability forest for the treatment model, and denote it as $\hat{f}(\mathbf{x}, \mathbf{w})$.
 - (b) Based on $\hat{f}(\mathbf{x}, \mathbf{w})$, compute the out-of-bag prediction for the full sample, and denote it as $\hat{d}^{(\text{oob})}$.
 - (c) Compute the treatment residuals for the full sample: $\tilde{d} = d - \hat{d}^{(\text{oob})}$.
3. Using the full sample, estimate the function $\tau(\mathbf{x})$ via linear regression or via generalized random forest:

$$\sum_{i=1}^N \left[\alpha(x_i) \tilde{d}_i^{(-i)} \left\{ \tilde{y}_i^{(-i)} - \tilde{d}_i^{(-i)} \tau(\mathbf{x}) \right\} \right] = 0$$

`cate po` implements both algorithms 1 and 2, and the cross-fitting-based PO estimator in algorithm 1 is the default. In algorithm 2, the `oob` option specifies to use the out-of-bag prediction, and it also requires that options `omethod()` and `tmethod()` be specified with random forest.

In practice, we need to specify the following points in `cate po` for the IATE estimation:

1. The outcome variable y and the treatment variable d .
2. The treatment-effects conditioning variables $\mathbf{x}$, which correspond to `catevarlist` in the syntax of `cate`.
3. The `controls()` option with the control variables $\mathbf{w}$, which are empty by default.
4. The `cmethod()` option with the estimation method for the IATE function $\tau(\mathbf{x})$, which can be either random forest or linear regression. The default is random forest.
5. The `omethod()` option with the estimation method for $E(y|\mathbf{x}, \mathbf{w})$, which can be lasso, random forest, or linear regression. The default is lasso.
6. The `tmethod()` option with the estimation method for the treatment model $f(\mathbf{x}, \mathbf{w})$, which can be logit, probit, lasso, or random forest. The default is lasso.

At minimum, we must specify points 1 and 2 and use the default settings for the other points.

Here is a general note on choosing between random forest, lasso, and parametric regression: Random forest is suitable when the number of covariates is low-dimensional relative to the number of observations and the function is smooth enough. Lasso is suitable when the model can be approximated by a sparse function, which can be very useful in the presence of high-dimensional controls. Parametric regression is easy to compute and interpret but imposes strong assumptions that may be too hard to satisfy with real-world data. In terms of computational speed, lasso is generally faster than random forest.

PO GATE estimator with prespecified groups

The GATE estimator is constructed by regressing the AIPW scores implied by the partial linear model on the group dummy variables, following [Semenova and Chernozhukov \(2021\)](#). In the partial linear model, the AIPW score is defined as

$$\Gamma_i = \hat{\tau}^{(-i)}(\mathbf{x}_i) + \frac{d_i - \hat{f}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i)}{\hat{f}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i) \{1 - \hat{f}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i)\}} \{y_i - \hat{\mu}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i, d_i)\}$$

where

1. $\hat{\tau}^{(-i)}(\mathbf{x}_i)$ is the out-of-bag prediction of IATE produced by `predict` after `cate po`. If CATE is estimated by linear regression, then $\hat{\tau}^{(-i)}(\mathbf{x}_i)$ is the linear prediction.
2. d_i is the treatment indicator.
3. $\hat{f}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i)$ is the prediction of propensity scores. These out-of-sample predictions are already produced in the cross-fitting algorithm 1 or the out-of-bag prediction algorithm 2.
4. y_i is the outcome.
5. $\hat{\mu}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i, d_i)$ is the out-of-sample prediction of the outcome mean conditional on both the controls and the treatment status. It is defined as

$$\hat{\mu}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i, d_i) = \hat{h}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i) + \{d_i - \hat{f}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i)\} \hat{\tau}^{(-i)}(\mathbf{x}_i)$$

where $\hat{h}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i)$ is the out-of-sample prediction of $E(y|\mathbf{x}, \mathbf{w})$, which are already computed in either algorithm 1 or 2 when estimating the IATE function.

Here we derive $\hat{\mu}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i, d_i)$. By the definition of the partial linear model, $h(\mathbf{x}, \mathbf{w}) = E(y|\mathbf{x}, \mathbf{w}) = \tau(\mathbf{x})f(\mathbf{x}, \mathbf{w}) + g(\mathbf{x}, \mathbf{w})$. Denote $\mu_0(\mathbf{x}, \mathbf{w})$ as the conditional expected value of the potential outcome when it is not treated. It is defined as

$$\mu_0(\mathbf{x}, \mathbf{w}) = g(\mathbf{x}, \mathbf{w}) = h(\mathbf{x}, \mathbf{w}) - \tau(\mathbf{x})f(\mathbf{x}, \mathbf{w})$$

Denote $\mu_1(\mathbf{x}, \mathbf{w})$ as the conditional expected value of the potential outcome when it is treated. It is defined as

$$\mu_1(\mathbf{x}, \mathbf{w}) = \mu_0(\mathbf{x}, \mathbf{w}) + \tau(\mathbf{x}) = h(\mathbf{x}, \mathbf{w}) + \{1 - f(\mathbf{x}, \mathbf{w})\}\tau(\mathbf{x})$$

Thus, the cross-fitting version of $\mu(\mathbf{x}, \mathbf{w}, d)$ is defined as above.

We already computed all the necessary terms when fitting the IATE function, so it is computationally convenient to compute the AIPW scores. Once we get the AIPW scores, computing the GATE is easy. Just run a linear regression of the AIPW scores on the group indicators. For details on estimating the GATE in the partial linear model, see algorithm 3 below.

Algorithm 3: PO for the GATE with a prespecified group

1. Set the group indicator variable G .
2. Run either algorithm 1 or algorithm 2 to get the following terms:
 - (a) $\widehat{y}_i^{(-i)} = \widehat{h}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i)$
 - (b) $\widehat{d}_i^{(-i)} = \widehat{f}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i)$
3. Compute the out-of-bag prediction of the IATE $\widehat{\tau}^{(-i)}(\mathbf{x}_i)$.
4. Compute the AIPW scores for the partial linear model:

$$\Gamma_i = \widehat{\tau}^{(-i)}(\mathbf{x}_i) + \frac{d_i - \widehat{f}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i)}{\widehat{f}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i) \{1 - \widehat{f}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i)\}} \{y_i - \widehat{\mu}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i, d_i)\}$$

with

$$\widehat{\mu}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i, d_i) = \widehat{h}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i) + \{d_i - \widehat{f}^{(-i)}(\mathbf{x}_i, \mathbf{w}_i)\} \widehat{\tau}^{(-i)}(\mathbf{x}_i)$$

5. Run ordinary least-squares (OLS) regression of the AIPW scores Γ_i on the group dummy indicators based on G . The coefficients estimated on these indicators are the GATES.

PO GATES estimator with data-driven groups

Suppose we do not know which groups we should condition on when computing the GATES. We can ask the data to discover the groups based on the sorted estimates of the IATES. The groups are generated by the quantiles of IATE estimates. For example, if we want to divide the IATES into four groups, the first group will be observations with IATE estimates greater than the 75th percentile of the overall IATE estimates; the second group will be observations with estimates between the 50th and 75th percentiles; the third group will be between the 25th and 50th percentiles; and the last group will be below the 25th percentile.

We use the cross-fitting technique to generate the group's ranking to avoid overfitting. Suppose we split the data into K folds. For each fold, we do the following. Observations in the k th fold are defined as the main sample, and observations in the other folds are defined as the auxiliary sample. First, we train an IATE model using the auxiliary sample. Second, we predict the IATE function $\tau(\mathbf{x})$ in the main sample and denote the prediction as $\hat{\tau}(\mathbf{x})^{(k)}$. The ranking in the main sample depends on the quantile of $\hat{\tau}(\mathbf{x})^{(k)}$. After circulating all the folds, we divide the full sample into the prespecified number of groups.

Because of the nature of cross-fitting, using the data to discover groups is time consuming. In addition, it requires a large sample because we need to fit a separate random forest model for the IATE in each fold.

Once the group is discovered, we run an OLS regression of the AIPW scores, which is generated in the cross-fitting procedure, to the group indicator. For details of the GATES estimator with the data-driven group for the partial linear model, see algorithm 4. For a discussion of using the sorted effects to generate the group, see Chernozhukov et al. (2006, sec. E) and Golub Capital Social Impact Lab (2023, chap. 4).

Algorithm 4: GATES estimator for data-driven group for partial linear model

1. Define the input.
 - (a) Set number of cross-fitting folds K .
 - (b) Set the estimation method for the outcome model. It can be one of random forest, lasso, square-root lasso, or linear regression.
 - (c) Set the estimation method for the treatment model. It can be one of random forest, lasso, logit, or probit.
 - (d) Set the number of groups G to divide.
2. Do the cross-fitting to construct the AIPW scores and group ranking.
 - (a) Randomly split the data into K folds.
 - (b) For each fold $k = 1$ to K , do the following:
 - i. Define the main sample S^M as the observations in the k th fold and the auxiliary sample S^A as the observations not in the k th fold.
 - ii. Construct the outcome prediction.
 - A. Using the auxiliary sample S^A , train the outcome model $\hat{h}^A(\mathbf{x}, \mathbf{w})$.
 - B. Based on $\hat{h}^A(\mathbf{x}, \mathbf{w})$, predict the outcome in the main sample S^M . Denote the prediction as $\hat{h}^M(\mathbf{x}_i, \mathbf{w}_i)$.
 - iii. Construct the treatment prediction.
 - A. Using the auxiliary sample S^A , train the treatment model $\hat{f}^A(\mathbf{x}, \mathbf{w})$.
 - B. Based on $\hat{f}^A(\mathbf{x}, \mathbf{w})$, predict the propensity score in the main sample S^M . Denote the prediction as $\hat{f}^M(\mathbf{x}_i, \mathbf{w}_i)$.
 - iv. Construct the IATE ranking.
 - A. Using the auxiliary sample S^A , fit the IATE model $\hat{\tau}(\mathbf{x})^A$ using algorithm 1.
 - B. Based on $\hat{\tau}(\mathbf{x})^A$, predict $\tau(\mathbf{x})$ in the main sample and denote it as $\hat{\tau}(\mathbf{x})^M$.
 - C. Generate the ranking in the main sample using the quantiles of $\hat{\tau}(\mathbf{x})^M$.

v. Construct the AIPW scores in the main sample:

$$\Gamma_i^M = \hat{\tau}^M(\mathbf{x}_i) + \frac{d_i - \hat{f}^M(\mathbf{x}_i, \mathbf{w}_i)}{\hat{f}^M(\mathbf{x}_i, \mathbf{w}_i) \{1 - \hat{f}^M(\mathbf{x}_i, \mathbf{w}_i)\}} \{y_i - \hat{\mu}^M(\mathbf{x}_i, \mathbf{w}_i, d_i)\}$$

with

$$\hat{\mu}^M(\mathbf{x}_i, \mathbf{w}_i, d_i) = \hat{h}^M(\mathbf{x}_i, \mathbf{w}_i) + \{d_i - \hat{f}^M(\mathbf{x}_i, \mathbf{w}_i)\} \hat{\tau}^M(\mathbf{x}_i)$$

3. Regress the AIPW scores on the group dummies based on the IATE quantiles to estimate the GATESS.

AIPW for the fully interactive model

The fully interactive model is

$$y(1) = g_1(\mathbf{x}, \mathbf{w}) + \epsilon_1$$

$$y(0) = g_0(\mathbf{x}, \mathbf{w}) + \epsilon_0$$

$$d = f(\mathbf{x}, \mathbf{w}_2) + u$$

where $g_1(\mathbf{x}, \mathbf{w})$ and $g_0(\mathbf{x}, \mathbf{w})$ are the models for the potential outcomes $y(1)$ and $y(0)$, respectively. ϵ_1 and ϵ_0 are the error terms. $\mathbf{w}_2$ is a vector of control variables for the treatment model. By default, $\mathbf{w}_2$ is equal to $\mathbf{w}$. However, it can be different if specified. The other terms, d , $\mathbf{x}$, $\mathbf{w}$, and $f(\mathbf{x}, \mathbf{w}_2)$, are the same as seen in the partial linear model. The fully interactive model allows the treatment effect to interact with both $\mathbf{x}$ and $\mathbf{w}$; thus, it is more general than the partial linear model.

By definition, the IATE function $\tau(\mathbf{x})$ is

$$\tau(\mathbf{x}) = E\{y(1) - y(0)|\mathbf{x}\} = E\{g_1(\mathbf{x}, \mathbf{w}) - g_0(\mathbf{x}, \mathbf{w})|\mathbf{x}\}$$

Intuitively, we can regress $g_1(\mathbf{x}, \mathbf{w}) - g_0(\mathbf{x}, \mathbf{w})$ on $\mathbf{x}$ to estimate $\tau(\mathbf{x})$. This method is also known as regression adjustment (RA). However, RA is vulnerable to machine learning mistakes made when estimating $g_1(\mathbf{x}, \mathbf{w})$ and $g_0(\mathbf{x}, \mathbf{w})$. Similarly, the inverse-probability weighting (IPW) estimator is also a bad choice. In contrast, the classical AIPW estimator, known as a doubly robust estimator, is Neyman orthogonal (see Chernozhukov et al. [2018] and Knaus [2022]). The AIPW version of the potential outcomes are

$$y(1)_{\text{AIPW}} = g_1(\mathbf{x}, \mathbf{w}) + \frac{I(d=1)\{y - g_1(\mathbf{x}, \mathbf{w})\}}{f(\mathbf{x}, \mathbf{w}_2)}$$

$$y(0)_{\text{AIPW}} = g_0(\mathbf{x}, \mathbf{w}) + \frac{I(d=0)\{y - g_0(\mathbf{x}, \mathbf{w})\}}{1 - f(\mathbf{x}, \mathbf{w}_2)}$$

Thus, $\tau(\mathbf{x})$ can also be written as

$$\tau(\mathbf{x}) = E\{y(1)_{\text{AIPW}} - y(0)_{\text{AIPW}}|\mathbf{x}\}$$

Given the estimates of $g_1(\mathbf{x}, \mathbf{w})$, $g_0(\mathbf{x}, \mathbf{w})$, and $f(\mathbf{x}, \mathbf{w}_2)$, let $\widehat{y(1)}_{\text{AIPW}}$ and $\widehat{y(0)}_{\text{AIPW}}$ be estimates of $y(1)_{\text{AIPW}}$ and $y(0)_{\text{AIPW}}$, respectively. Let $\widehat{\Gamma} = \widehat{y(1)}_{\text{AIPW}} - \widehat{y(0)}_{\text{AIPW}}$ be an estimate of the AIPW scores. We estimate $\tau(\mathbf{x})$ by solving the local moment condition

$$\sum_{i=1}^N [\alpha(\mathbf{x}_i) \{\widehat{\Gamma}_i - \tau(\mathbf{x})\}] = 0$$

where $\alpha(\mathbf{x}_i)$ defines the local weights that attach more weight to observations that are close to $\mathbf{x}$. To solve this moment condition, we use the honest regression forest proposed in [Wager and Athey \(2018\)](#) and implement it using the generalized random forest in [Athey, Tibshirani, and Wager \(2019\)](#). For details on honest regression forest, see [Generalized random forest](#).

AIPW IATE estimator

To guard against machine learning estimation bias in the nuisance function, we need to construct the AIPW scores using the out-of-sample prediction. We estimate the IATE function $\tau(\mathbf{x})$ by solving

$$\sum_{i=1}^N \left[\alpha(\mathbf{x}_i) \left\{ \hat{\Gamma}_i^{(-i)} - \theta(\mathbf{x}) \right\} \right] = 0$$

where $\hat{\Gamma}_i^{(-i)}$ are predictions of the AIPW scores for the i th observation via estimation of the nuisance function using observations excluding the i th observation.

We can construct the out-of-sample prediction using the cross-fitting technique. We split the data into K folds, define the main sample as the observations in the k th fold, and define the auxiliary sample as the observations not in the k th fold. We estimate all the nuisance functions by using the auxiliary sample and get the out-of-sample predictions in the main sample. After circulating through all the folds, we eventually compute the out-of-sample predictions for the full sample. For details of the cross-fitting version of the AIPW estimator for the IATE, see algorithm 5 below.

Algorithm 5: AIPW estimator for the IATE using cross-fitting

1. Define the input.
 - (a) Set the number of cross-fitting folds K .
 - (b) Set the estimation method for the outcome model. It can be one of random forest, lasso, square-root lasso, or linear regression.
 - (c) Set the estimation method for the treatment model. It can be one of random forest, lasso, logit, or probit.
2. Do the cross-fitting to construct the AIPW scores.
 - (a) Randomly split the data into K folds.
 - (b) For each fold $k = 1$ to K , do the following:
 - i. Define the main sample S^M as the observations in the k th fold and the auxiliary sample S^A as the observations not in the k th fold.
 - ii. Construct $g_1(\mathbf{x}, \mathbf{w})$.
 - A. Using the auxiliary sample S^A and treated observations ($d_i = 1$), train the outcome model $\hat{g}_1^A(\mathbf{x}, \mathbf{w})$.
 - B. Based on $\hat{g}_1^A(\mathbf{x}, \mathbf{w})$, predict the treated potential outcome in the main sample S^M . Denote the prediction as $\widehat{y(1)}^M$.
 - iii. Construct $g_0(\mathbf{x}, \mathbf{w})$.
 - A. Using the auxiliary sample S^A and untreated observations ($d_i = 0$), train the outcome model $\hat{g}_0^A(\mathbf{x}, \mathbf{w})$.
 - B. Based on $\hat{g}_0^A(\mathbf{x}, \mathbf{w})$, predict the untreated potential outcome in the main sample S^M . Denote the prediction as $\widehat{y(0)}^M$.

- iv. Construct the propensity score.
 - A. Using the auxiliary sample S^A , train the outcome model $\hat{f}^A(\mathbf{x}, \mathbf{w})$.
 - B. Based on $\hat{f}^A(\mathbf{x}, \mathbf{w})$, predict the propensity score in the main sample S^M . Denote the prediction as $\hat{d}^M$.
- v. Construct the AIPW score in the main sample S^M as

$$\Gamma_i^M = \left[\widehat{y(1)}_i^M + \frac{d_i \left\{ y_i - \widehat{y(1)}_i^M \right\}}{\hat{d}_i^M} \right] - \left[\widehat{y(0)}_i^M + \frac{(1 - d_i) \left\{ y_i - \widehat{y(0)}_i^M \right\}}{1 - \hat{d}_i^M} \right]$$

3. Fit an honest random forest regression of Γ on $\mathbf{x}$ using the generalized random forest in [Athey, Tibshirani, and Wager \(2019\)](#) or fit a linear regression of Γ on $\mathbf{x}$.

Cross-fitting can be applied to generic machine learning techniques. However, by construction, it is computationally demanding.

When the outcome and treatment models are estimated using random forest, we can use a particular case of cross-fitting that saves computational time. That is, we can use the out-of-bag predictions to construct the residuals. The out-of-bag prediction for an observation is constructed using only the trees in the random forest that do not contain this observation. See [Generalized random forest](#) for more details. This procedure is equivalent to cross-fitting but has a faster computation time. For details of the out-of-bag prediction-based PO estimator for the IATE, see algorithm 6 below.

Algorithm 6: AIPW estimator for the IATE using out-of-bag prediction

1. Construct $g_1(\mathbf{x}, \mathbf{w})$.
 - (a) Using the full sample and the treated observations ($d_i = 1$), fit a regression forest for the outcome model. Denote it as $\hat{g}_1(\mathbf{x}, \mathbf{w})$.
 - (b) Based on $\hat{g}_1(\mathbf{x}, \mathbf{w})$, predict the treated potential outcome for the full sample using the out-of-bag prediction. Denote it as $\widehat{y(1)}^{(\text{oob})}$.
2. Construct $g_0(\mathbf{x}, \mathbf{w})$.
 - (a) Using the full sample and the untreated observations ($d_i = 0$), fit a regression forest for the outcome model. Denote it as $\hat{g}_0(\mathbf{x}, \mathbf{w})$.
 - (b) Based on $\hat{g}_0(\mathbf{x}, \mathbf{w})$, predict the treated potential outcome for the full sample using the out-of-bag prediction. Denote it as $\widehat{y(0)}^{(\text{oob})}$.
3. Construct the propensity score.
 - (a) Using the full sample, fit a probability forest for the treatment model. Denote it as $\hat{f}(\mathbf{x}, \mathbf{w})$.
 - (b) Based on $\hat{f}(\mathbf{x}, \mathbf{w})$, predict the propensity score for the full sample using the out-of-bag prediction. Denote it as $\hat{d}^{(\text{oob})}$.
4. Construct the AIPW scores as

$$\Gamma_i = \left[\widehat{y(1)}_i^{(\text{oob})} + \frac{d_i \left\{ y_i - \widehat{y(1)}_i^{(\text{oob})} \right\}}{\hat{d}_i^{(\text{oob})}} \right] - \left[\widehat{y(0)}_i^{(\text{oob})} + \frac{(1 - d_i) \left\{ y_i - \widehat{y(0)}_i^{(\text{oob})} \right\}}{1 - \hat{d}_i^{(\text{oob})}} \right]$$

5. Fit a random forest regression of Γ on $\mathbf{x}$ using the generalized random forest in [Athey, Tibshirani, and Wager \(2019\)](#) or fit a linear regression of Γ on $\mathbf{x}$.

`cate aipw` implements both algorithms 5 and 6, and the cross-fitting-based AIPW estimator in algorithm 5 is the default. The `oob` option specifies to use the out-of-bag prediction in algorithm 6, and it requires that options `omethod()` and `tmethod()` be specified with random forest.

In practice, we must specify the following points in `cate aipw` for the IATE estimation:

1. The outcome variable y and the treatment variable d .
2. The treatment-effects conditioning variables $\mathbf{x}$, which correspond to `catevarlist` in the syntax of `cate`.
3. The `controls()` option with the control variables $\mathbf{w}$, which are empty by default.
4. The `cmethod()` option with the estimation method for the IATE function $\tau(\mathbf{x})$, which can be either random forest or linear regression. The default is random forest.
5. The `omethod()` option with the estimation method for $g_1(\mathbf{x}, \mathbf{w})$ and $g_0(\mathbf{x}, \mathbf{w})$ can be lasso, random forest, or linear regression. The default is lasso.
6. The `tmethod()` option with the estimation method for the treatment model $f(\mathbf{x}, \mathbf{w})$, which can be logit, probit, lasso, or random forest. The default is lasso.

At minimum, we must specify points 1 and 2 and use the default settings for the other points.

Both PO and AIPW are consistent estimators of $\tau(\mathbf{x})$ under similar regularity conditions, and they are Neyman orthogonal in the sense that the estimates are robust to the machine learning mistakes made in the nuisance parameters, such as $g(\mathbf{x}, \mathbf{w})$ or $f(\mathbf{x}, \mathbf{w})$.

The AIPW estimator is more efficient than the PO estimator. That is, in large samples, the AIPW estimates of the IATE function are more precise than the PO estimates (see [Kennedy \[2023\]](#)). In addition, the AIPW estimator has double robustness; that is, the estimator is still consistent even if the outcome model or the treatment model is misspecified (see [Chernozhukov et al. \[2018\]](#)).

However, the PO estimator is more robust than the AIPW estimator when there are some nearly perfect predictions of the propensity score $f(\mathbf{x}, \mathbf{w})$. More precisely, the AIPW estimator needs to compute the inverse of propensity score $f(\mathbf{x}, \mathbf{w})$ or $1 - f(\mathbf{x}, \mathbf{w})$, which lies between 0 and 1. Thus, the AIPW estimator may be undefined whenever the propensity score $f(\mathbf{x}, \mathbf{w})$ estimates are close to 0 or 1. In contrast, the PO estimator does not have these issues.

AIPW GATE estimator with prespecified groups

The GATE estimator for the fully interactive model is the OLS estimate for the AIPW scores on the group indicator, following [Semenova and Chernozhukov \(2021\)](#).

In the full interactive model, the AIPW score is defined as

$$\Gamma_i = \left[\widehat{y(1)}_i^{(-i)} + \frac{d_i \left\{ y_i - \widehat{y(1)}_i^{(-i)} \right\}}{\widehat{d}_i^{(-i)}} \right] - \left[\widehat{y(0)}_i^{(-i)} + \frac{(1 - d_i) \left\{ y_i - \widehat{y(0)}_i^{(-i)} \right\}}{1 - \widehat{d}_i^{(-i)}} \right]$$

where

1. y_i is the observed outcome.
2. d_i is the observed treatment indicator.
3. $\widehat{y(1)}_i^{(-i)}$ is the out-of-sample prediction of the treated potential outcome.
4. $\widehat{y(0)}_i^{(-i)}$ is the out-of-sample prediction of the untreated potential outcome.
5. $\widehat{d}_i^{(-i)}$ is the out-of-sample prediction of the propensity score.

The scores can be obtained using either the cross-fitting AIPW estimator in algorithm 5 or the out-of-bag prediction-based AIPW estimator in algorithm 6. After the AIPW scores are obtained, the GATE estimator is just the OLS estimate of scores on the group indicator. For details of estimating the GATE in the fully interactive model, see algorithm 7 below.

Algorithm 7: GATE estimator with a prespecified group

1. Select the group variable G .
2. Run either algorithm 5 or algorithm 6 to obtain the AIPW scores $\Gamma_i^{(-i)}$.

$$\Gamma_i^{(-i)} = \left[\widehat{y(1)}_i^{(-i)} + \frac{d_i \left\{ y_i - \widehat{y(1)}_i^{(-i)} \right\}}{\widehat{d}_i^{(-i)}} \right] - \left[\widehat{y(0)}_i^{(-i)} + \frac{(1 - d_i) \left\{ y_i - \widehat{y(0)}_i^{(-i)} \right\}}{1 - \widehat{d}_i^{(-i)}} \right]$$

3. Run OLS of Γ on the group indicators based on G .

AIPW GATES estimator with data-driven groups

In the fully interactive model, the procedure to compute the GATESs with data-driven groups is very similar to that for the partial linear model. The only difference is using the AIPW scores implied by the fully interactive model. For details of estimating the GATESs with data-driven groups for the fully interactive model, see algorithm 8.

Algorithm 8: GATES estimator with data-driven groups in fully interactive model

1. Define the input.
 - (a) Set the number of cross-fitting folds K .
 - (b) Select the estimation method for the outcome and treatment models.
2. Perform cross-fitting to construct the AIPW scores.
 - (a) Randomly split the data into K folds.
 - (b) For each fold $k = 1$ to K , do the following:
 - i. Define the main sample S^M as the observations in folds k and the auxiliary sample S^A as the observations not in folds k .

- ii. Construct $g_1(\mathbf{x}, \mathbf{w})$.
 - A. Using the auxiliary sample S^A and treated observations ($d_i = 1$), train the outcome model $\hat{g}_1^A(\mathbf{x}, \mathbf{w})$.
 - B. Based on $\hat{g}_1^A(\mathbf{x}, \mathbf{w})$, predict the treated potential outcome in the main sample S^M . Denote the prediction as $\widehat{y(1)}^M$.
- iii. Construct $g_0(\mathbf{x}, \mathbf{w})$.
 - A. Using the auxiliary sample S^A and untreated observations ($d_i = 0$), train the outcome model $\hat{g}_0^A(\mathbf{x}, \mathbf{w})$.
 - B. Based on $\hat{g}_0^A(\mathbf{x}, \mathbf{w})$, predict the untreated potential outcome in the main sample S^M . Denote the prediction as $\widehat{y(0)}^M$.
- iv. Construct the propensity score.
 - A. Using the auxiliary sample S^A , train the outcome model $\hat{f}^A(\mathbf{x}, \mathbf{w})$.
 - B. Based on $\hat{f}^A(\mathbf{x}, \mathbf{w})$, predict the propensity score in the main sample S^M . Denote the prediction as $\hat{d}^M$.
- v. Construct the AIPW score in the main sample S^M as

$$\Gamma_i^M = \left[\widehat{y(1)}_i^M + \frac{d_i \left\{ y_i - \widehat{y(1)}_i^M \right\}}{\hat{d}_i^M} \right] - \left[\widehat{y(0)}_i^M + \frac{(1 - d_i) \left\{ y_i - \widehat{y(0)}_i^M \right\}}{1 - \hat{d}_i^M} \right]$$

- vi. Construct the IATE ranking.
 - A. Using the auxiliary sample S^A , train the IATE model $\hat{\tau}(\mathbf{x})^A$ using algorithm 5.
 - B. Based on $\hat{\tau}(\mathbf{x})^A$, predict $\tau(\mathbf{x})$ in the main sample, and denote it as $\hat{\tau}(\mathbf{x})^M$.
 - C. Generate the ranking in the main sample based on the quantiles of $\hat{\tau}(\mathbf{x})^M$.

3. Run an OLS regression of the AIPW scores on the generated groups' indicator dummies.

Generalized random forest

The generalized random forest, proposed in [Athey, Tibshirani, and Wager \(2019\)](#), solves the moment condition

$$E \left\{ \psi_{\theta(\mathbf{x})}(\mathbf{o}_i) | \mathbf{x}_i = \mathbf{x} \right\} = 0$$

where $\theta(\mathbf{x})$ is a parameter of interest, $\mathbf{o}_i$ is a vector of variables including the outcome variable and some covariates, and $\mathbf{x}_i$ is a vector of covariates in the function $\theta(\mathbf{x})$.

The form of $\psi_{\theta(\mathbf{x})}(\mathbf{o}_i)$ varies depending on the context. For example, in the regression or probability random forest, the parameter of interest is $\theta(\mathbf{x}) = E(y|\mathbf{x})$, and the moment condition is

$$E \{ y_i - \theta(\mathbf{x}) | \mathbf{x}_i = \mathbf{x} \} = 0$$

In the causal random forest, the parameter of interest is $\theta(\mathbf{x}) = \tau(\mathbf{x})$, and the moment condition is

$$E \left[\tilde{d}_i \left\{ \tilde{y}_i - \theta(\mathbf{x}) \tilde{d}_i \right\} \middle| \mathbf{x}_i = \mathbf{x} \right] = 0$$

where $\tilde{d}$ and $\tilde{y}$ are partialled-out residuals of d and y discussed in the PO estimator.

To estimate $\theta(\mathbf{x})$, generalized random forest solves the empirical moment condition

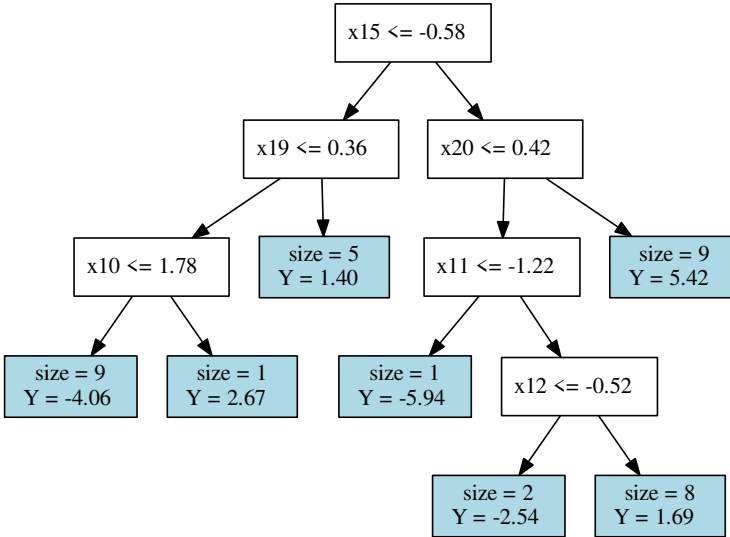
$$\sum_{i=1}^N \alpha_i(\mathbf{x}) \psi_{\theta(\mathbf{x})}(\mathbf{o}_i) = 0$$

where $\alpha_i(\mathbf{x})$ defines the local weights that attach more weight to observations close to x . Generalized random forest obtains $\alpha_i(\mathbf{x})$ by averaging the neighborhood implied by a forest. See [Athey, Tibshirani, and Wager \(2019, sec. 2\)](#) for a detailed discussion.

The generalized random forest is an ensemble of honest trees, which we explain next.

Honest tree

A tree is an algorithm that divides the data into different parts, such that each part consists of observations that are as similar as possible. The following graph illustrates a regular regression tree:



For example, in the above graph, we first look at variable x_{15} to determine how to divide the data. If its value is smaller than or equal to -0.58 , that observation goes to the left; otherwise, it goes to the right. Each time we split the data, it is represented by a node. We recursively continue this procedure until we hit a “leaf” node, which means we cannot find a variable to split the data.

The leaf nodes in the graph are represented as a blue rectangle. A leaf node defines a part of the tree partition. For example, the leftmost leaf has 9 observations (size = 9 and $Y = -4.06$ in the graph). We can travel to this leaf by finding observations satisfying three conditions: $x_{15} \leq -0.58$, $x_{19} \leq 0.36$, and $x_{10} \leq 1.78$. These three conditions correspond to the three nodes from the top to the bottom, enabling us to travel from the top to the particular leaf.

In each leaf node, the mean of the outcome within a leaf is used to label the leaf or attach a value to it. Thus, constructing a tree involves two main steps: splitting the tree and labeling the tree. More precisely, tree construction involves the following:

1. Splitting the tree: For each node, find a splitting variable x_s and a value v_s such that the resulting divisions are as different as possible. In particular, we find x_s and v_s by solving

$$(x_s, v_s) = \operatorname{argmin}_{(x_m, v_m) \in \mathbf{x} \times D(x_m)} \{\operatorname{cost}(x_m \leq v_m) + \operatorname{cost}(x_m > v_m)\}$$

where $\operatorname{cost}(\cdot)$ is a cost function that characterizes the node's homogeneity and its definition varies depending on the context. For example, in the regression forest, the cost function is the mean squared error of the outcome; in the probability forest, the cost function is the Gini index for the outcome; in the causal forest, the cost function is the squared sum of the influence function implied by the causal forest moment condition. For details, see [Athey, Tibshirani, and Wager \(2019, sec. 2\)](#).

2. When there are not enough observations in any node, the algorithm stops searching. The `splitminobs()` option specifies the minimum number of observations needed to perform a split.
3. Labeling the tree: In each leaf, attach a value or a label to the leaf. The labeling formula varies depending on the context. In the regression or probability forest, the label is the outcome's mean within a leaf. In the causal forest, the label consists of the means of the outcome, the treatment, and their interactions. For details, see [Athey, Tibshirani, and Wager \(2019, sec. 2\)](#).

An honest tree differs from a regular tree by dividing the sample into two subsamples. One is used to split the tree, and the other is used to label the leaves. For details, see algorithm 9 below.

Algorithm 9: Honest tree

1. Split the data into two parts: A and B . The `honestrate()` option specifies the fraction of the data used to construct A .
2. Use sample A to split the tree.
3. Use sample B to label the tree.

Honest random forest

The honest random forest is an ensemble of honest trees with some requirements on the randomness of the tree. There are two main requirements: first, each tree must use a random subsample of the data; second, for each node in the tree, only a random subset of the variables may be searched to find the best variable to split. For more details on the honest random forest, see algorithm 10 below.

Algorithm 10: Honest random forest

1. Define B as the number of trees, which can be specified in the `ntrees()` option.
2. For each tree $b = 1$ to B , do the following:
 - (a) Draw a random sample S_b of the full data. The `samprate()` option specifies the fraction of the sample.

- (b) Based on S_b , construct an honest tree using algorithm 9. In each node of the honest tree, only a random subset of variables is searched to find the best variable to split. The mean number of this variable set follows a Poisson distribution with expectation μ . The `splitmeanvars()` option specifies μ .
3. Use the forest to construct local weights $\alpha_i(\mathbf{x})$, and then estimate $\theta(\mathbf{x})$ by solving the moment condition $\sum_{i=1}^N \alpha_i(\mathbf{x}) \psi_{\theta(\mathbf{x})}(\mathbf{o}_i) = 0$.

Confidence intervals

The confidence intervals of the estimate $\theta(\mathbf{x})$ in generalized random forest are constructed using the delta method, as discussed in [Athey, Tibshirani, and Wager \(2019, sec. 4\)](#). In particular, the confidence intervals with significance level α are defined as

$$\lim_{n \rightarrow \infty} E \left[\theta(\mathbf{x}) \in \left\{ \hat{\theta}(\mathbf{x}) \pm \Phi^{-1}(1 - \alpha/2) \hat{\sigma}(\mathbf{x}) \right\} \right] = \alpha$$

where $\Phi^{-1}(\cdot)$ is the inverse function of the Gaussian cumulative distribution function and $\hat{\sigma}(\mathbf{x})$ is an estimate of the standard errors of $\hat{\theta}(\mathbf{x})$.

The variance of $\theta(\mathbf{x})$ is defined as

$$\text{Var}\{\theta(\mathbf{x})\} = \xi' V(\mathbf{x})^{-1} H(\mathbf{x}) V(\mathbf{x})^{-1} \xi$$

where $V(\mathbf{x}) = \frac{\partial E\{\psi_{\theta(\mathbf{x})}(\mathbf{o}) | \mathbf{x} = \mathbf{x}\}}{\partial \theta(\mathbf{x})}$, ξ is the subvector selector, and $H(\mathbf{x}) = \text{Var}\left\{\sum_{i=1}^N \alpha_i \psi_{\theta(\mathbf{x})}(\mathbf{o}_i)\right\}$. See [Athey, Tibshirani, and Wager \(2019, sec. 4\)](#) for a detailed discussion on $V(\mathbf{x})$ and ξ . We use the bootstrap of little bags to estimate $H(\mathbf{x})$. See algorithm 11 for details.

Algorithm 11: Bootstrap of little bags

1. Define the number of trees in each bag, l , which can be specified in the `cintrees()` option.
2. Define the number of trees in the forest, B , which can be specified in the `ntrees()` option.
3. For each bag, $g = 1, \dots, \lceil B/l \rceil$, draw a random half-sample $H_g \subset \{1, \dots, n\}$ of size $\lceil n/2 \rceil$, where n is the total number of observations.
4. For each tree, $b = 1, \dots, B$, draw a random sample $I_b \subseteq H_{\lceil b/l \rceil}$, and build an honest tree using I_b . This step implies that the trees in the same bag are drawing samples from the same half-sample H_g .
5. Define $\Psi_b\{\theta(\mathbf{x})\} = \sum_{i=1}^n \alpha_{bi}(\mathbf{x}) \psi_{\theta(\mathbf{x})}(\mathbf{o}_i)$, where $\alpha_{bi}(\mathbf{x})$ is the tree-level local weight. Similarly, define $\Psi\{\theta(\mathbf{x})\} = \sum_{i=1}^n \alpha_i(\mathbf{x}) \psi_{\theta(\mathbf{x})}(\mathbf{o}_i)$, where $\alpha_i(\mathbf{x})$ is the forest-level local weight. Estimate $H(\mathbf{x})$ as

$$\widehat{H}(\mathbf{x}) = E \left\{ \left(\frac{1}{l} \sum_{b=1}^l \Psi_b - \Psi \right)^2 \right\} - \frac{1}{l-1} E \left\{ \frac{1}{l} \sum_{b=1}^l \left(\Psi_b - \frac{1}{l} \sum_{b=1}^l \Psi_b \right)^2 \right\}$$

where the expectation is taken over the bags.

See [Athey, Tibshirani, and Wager \(2019, sec. 4.1\)](#) for a detailed discussion on the consistency of the bootstrap of little bags.

Missing values

Random forest can handle missing values in covariates. When facing a missing value of a covariate, there are three possible scenarios for splitting a node. All three scenarios are computed for every variable in the random subset of variables to search, and the one with the best cost function is used. For more discussion, see [Twala, Jones, and Hand \(2008\)](#).

1. Scenario 1: All observations that are not missing in a specific covariate are sent to the left node, and all the other observations are sent to the right node.

Missing values can provide information on the dependent variable, and potentially, a missing value of a covariate can be used to split a node in the tree.

2. We use the optimal nonmissing value of the specific covariate (the value that minimizes the cost) to split observations and then proceed as follows:
 - (a) Scenario 2: The observations with missing values are sent to the right node.
 - (b) Scenario 3: The observations with missing values are sent to the left node.

In `cate`, the observations with missing covariate values are kept if the random forest is used in all the models, that is, when `omethod()`, `tmethod()`, and `cmethod()` all use `rforest`. Specifying the `rflistwise` option will drop the observations with missing covariate values when all the models are estimated using random forest.

In contrast, if one of `omethod()`, `tmethod()`, or `cmethod()` does not use `rforest`, the observations with missing covariates will be dropped before any computation. This is because estimation methods such as `lasso`, `regress`, and `probit` will predict missing values if one of the covariates is missing, and the predicted missing values make the dependent variable of the IATE estimation missing. Because random forest cannot handle missing dependent variable values, these observations will be dropped eventually.

References

- Athey, S., J. Tibshirani, and S. Wager. 2019. Generalized random forests. *Annals of Statistics* 47: 1148–1178. <https://doi.org/10.1214/18-AOS1709>.
- Cerulli, G. 2025. *Optimal policy learning using Stata*. *Stata Journal* 25: 309–343.
- Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. B. Hansen, W. K. Newey, and J. M. Robins. 2018. Double/debiased machine learning for treatment and structural parameters. *Econometrics Journal* 21: C1–C68. <https://doi.org/10.1111/ectj.12097>.
- Chernozhukov, V., M. Demirer, E. Duflo, and I. Fernández-Val. 2006. Generic machine learning inference on heterogeneous treatment effects in randomized experiments, with an application to immunization in India. NBER Working Paper 24678, National Bureau of Economic Research. <https://doi.org/10.3386/w24678>.
- Chernozhukov, V., and C. B. Hansen. 2004. The effects of 401(k) participation on the wealth distribution: An instrumental quantile regression analysis. *Review of Economics and Statistics* 86: 735–751. <https://doi.org/10.1162/0034653041811734>.
- Golub Capital Social Impact Lab. 2023. *Machine Learning-based Causal Inference Tutorial*. Stanford, CA, <https://bookdown.org/stanfordgsbsilab/ml-ci-tutorial/hte-i-binary-treatment.html>.
- Head, M. L., L. Holman, R. Lanfear, A. T. Kahn, and M. D. Jennions. 2015. The extent and consequences of p-hacking in science. *PLOS Biology* 13: e1002106. <https://doi.org/10.1371/journal.pbio.1002106>.
- Kennedy, E. H. 2023. Towards optimal doubly robust estimation of heterogeneous causal effects. *Electronic Journal of Statistics* 17: 3008–3049. <https://doi.org/10.1214/23-EJS2157>.

- Knaus, M. C. 2022. Double machine learning–based programme evaluation under unconfoundedness. *Econometrics Journal* 25: 602–627. <https://doi.org/10.1093/ectj/utac015>.
- Koch, B., D. M. Vock, and J. Wolfson. 2018. Covariate selection with group lasso and doubly robust estimation of causal effects. *Biometrics* 74: 8–17. <https://doi.org/10.1111/biom.12736>.
- Nie, X., and S. Wager. 2021. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika* 108: 299–319. <https://doi.org/10.1093/biomet/asaa076>.
- Semenova, V., and V. Chernozhukov. 2021. Debiased machine learning of conditional average treatment effects and other causal functions. *Econometrics Journal* 24: 264–289. <https://doi.org/10.1093/ectj/utaa027>.
- Twala, B. E. T. H., M. C. Jones, and D. J. Hand. 2008. Good methods for coping with missing data in decision trees. *Pattern Recognition Letters* 29: 950–956. <https://doi.org/10.1016/j.patrec.2008.01.010>.
- Wager, S., and S. Athey. 2018. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association* 113: 1228–1242. <https://doi.org/10.1080/01621459.2017.1319839>.

Also see

[CAUSAL] **cate postestimation** — Postestimation tools for cate

[CAUSAL] **teffects aipw** — Augmented inverse-probability weighting

[CAUSAL] **telasso** — Treatment-effects estimation using lasso

[U] **20 Estimation and postestimation commands**

Postestimation commands

Remarks and examples

Also see

predict

Stored results

estat

Methods and formulas

catergraph

References

Postestimation commands

The following postestimation commands are of special interest after `cate`:

Command	Description
<code>estat heterogeneity</code>	test for treatment-effects heterogeneity
<code>estat gatetest</code>	test for group treatment-effects heterogeneity
<code>estat ate</code>	compute the average treatment effect (ATE) for a subpopulation
<code>estat projection</code>	fit a linear projection of the individualized average treatment effect (IATE) estimates on variables
<code>estat series</code>	fit a nonparametric series regression of the IATE estimates on variables
<code>estat policyeval</code>	evaluate treatment-assignment policy
<code>estat tassigneval</code>	synonym of <code>estat policyeval</code>
<code>estat classification</code>	perform classification analysis of the data-driven groups
<code>catergraph histogram</code>	histogram of the IATE predictions
<code>catergraph gateplot</code>	plot of the group average treatment effect (GATE) or sorted GATE (GATES) estimates
<code>catergraph iateplot</code>	plot of the IATE function estimates

The following postestimation commands are also available:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of parameters
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of parameters
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
<code>predict</code>	predict the IATE function or its confidence intervals
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of parameters
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` predicts the observation-level IATE function, the standard errors, or the IATE confidence intervals.

Menu for predict

Statistics > Postestimation

Syntax for predict

Syntax for predicting the IATE or the standard errors

```
predict [type] newvar [if] [in] [ , iate stdp ]
```

Syntax for predicting the IATE confidence intervals

```
predict [type] newvarll [type] newvarul [if] [in] , ci [level(#)]
```

`newvarll` and `newvarul` specify new variables for the lower and upper bounds of confidence intervals, respectively.

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Options for predict

`iate`, the default, predicts the IATE function point estimates for each observation. The prediction can be from either a random forest or a parametric regression, which depends on the specification of `cmethod()` in `cate`. If the `cmethod(rforest)` option is specified in `cate`, the IATE prediction is computed using the generalized random forest. If the `cmethod(regress)` option is specified in `cate`, the IATE prediction is computed using a parametric linear regression. By default, a random forest prediction of the IATE function is computed.

`stdp` predicts the standard errors of the predictions of the IATE function. For the IATE predictions based on random forest, the standard errors are computed using a bootstrap of little bags, and for the IATE predictions based on linear regression, they are computed using the parametric delta method.

`ci` predicts the confidence intervals of the predictions of the IATE function.

`level(#)`, available only with `ci`, sets the confidence levels of the confidence intervals; the default is `level(95)`.

estat

Description for estat

`estat heterogeneity` tests the null hypothesis that the treatment effects are homogeneous.

`estat gatetest` tests the null hypothesis that the ATEs are the same among the specified subgroups. This command is allowed only when the `group()` option is specified in `cate`.

`estat ate` computes the ATE for a subpopulation defined by an `if` or `in` qualifier.

`estat projection` fits a linear projection of the estimated IATE function on specified variables.

`estat series` performs nonparametric series regression of the estimated IATE function on specified variables using a B-spline, piecewise polynomial spline, or polynomial basis.

`estat policyeval` evaluates the prespecified treatment-assignment rule. In particular, it computes the value of the policy and compares the difference of the two policies' values if two policies are specified.

`estat tassigneval` is a synonym of `estat policyeval`.

`estat classification` performs a classification analysis of the groups constructed based on the sorted IATE estimates. It performs a two-sample t test to compare the mean of a variable between the group with the largest treatment effects and the group with the smallest treatment effects. It is only allowed when the `group()` option is specified in `cate`.

Menu for estat

Statistics > Postestimation

Syntax for estat

Perform test of treatment-effects heterogeneity

```
estat heterogeneity
```

Perform test of group treatment-effects heterogeneity

```
estat gatetest [grnumlist] [ , gatetest_option ]
```

grnumlist is a numlist that specifies the group levels to be tested. If none are specified, all levels are used.

Compute the ATE for a subpopulation

```
estat ate [if] [in] [ , ate_options ]
```

Fit a linear projection of the IATE estimates on variables

```
estat projection [varlist] [if] [in] [ , projection_options ]
```

If *varlist* is not specified, then *catevarlist* specified in `cate` will be used.

Fit a nonparametric series regression of the IATE estimates on variables

```
estat series indepvarsseries [if] [in] [ , series_options ]
```

indepvars_{series} is a list of independent variables for which a basis function will be formed.

Evaluate treatment-assignment policy

```
estat policyeval policyvar1 [policyvar2] [if] [in] [ , policy_options ]
```

policyvar₁ and *policyvar₂* are variables specifying the probability of assigning each observation to treatment. If two *policyvars* are specified, `estat policyeval` computes the values of each policy and their difference.

Perform classification analysis of the data-driven groups

```
estat classification varname [if] [in] [ , classification_options ]
```

<i>gatetest_option</i>	Description
<code>mtest(<i>mtest_options</i>)</code>	test each condition separately

<i>mtest_options</i>	Description
<code>noadjust</code>	no adjustment is to be made; the default
<code>bonferroni</code>	Bonferroni's method
<code>holm</code>	Holm's method
<code>sidak</code>	Šidák's method

Specifying `mtest` without an argument is equivalent to `mtest(noadjust)`.

<i>ate_options</i>	Description
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<i>display_options</i>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

<i>projection_options</i>	Description
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> can be one of <code>ols</code> or <code>robust</code>
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>noconstant</code>	suppress the constant term
<code>post</code>	post the results as the estimation results
<i>display_options</i>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

<i>series_options</i>	Description
Model	
<code>bspline</code>	use a third-order B-spline basis; the default
<code>bspline(#)</code>	use a B-spline basis of order #
<code>spline</code>	use a third-order piecewise polynomial spline basis
<code>spline(#)</code>	use a piecewise polynomial spline basis of order #
<code>polynomial</code>	use a polynomial basis
<code>polynomial(#)</code>	use a polynomial basis of order #
<code>asis(varlist)</code>	include <i>varlist</i> in model as specified; do not use in basis
<code>nointeract(seriesvarlist)</code>	use <i>seriesvarlist</i> in basis without interactions
<code>criterion(crittype)</code>	criterion to use; <i>crittype</i> may be <code>cv</code> , <code>gcv</code> , <code>aic</code> , <code>bic</code> , or <code>mallows</code>
<code>knots(#)</code>	use a piecewise polynomial spline or B-spline basis function with # knots
<code>knotsmat(matname)</code>	use knots in matrix <i>matname</i> for piecewise polynomial spline or B-spline estimation
<code>distinct(#)</code>	minimum number of distinct values allowed in continuous covariates; default is <code>distinct(10)</code>
SE	
<code>vce(vctype)</code>	<i>vctype</i> can be one of <code>ols</code> or <code>robust</code> ; default is <code>vce(robust)</code>
Graph	
* <code>graph[(seriesgraph_opts)]</code>	plot the prediction of conditional ATEs (CATEs)
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary regression coefficients
<i>display_options</i>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

* When `graph` or `graph()` is specified, only one variable is allowed in *indepvars*_{series}.

<i>seriesgraph_opts</i>	Description
<code>noci</code>	do not plot the confidence intervals
<code>cateopts(scatter_opts)</code>	affect rendition of the predicted CATE point estimates
<code>ciopts(area_opts)</code>	affect rendition of the confidence interval
<i>twoway_options</i>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>

<i>scatter_opts</i>	Description
<i>connect_options</i>	change the look of lines or connecting method
<i>marker_options</i>	change the look of markers (color, size, etc.)

<i>policy_options</i>	Description
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<i>display_options</i>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

<i>classification_options</i>	Description
<code>unequal</code>	data have unequal variances
<code>welch</code>	use Welch's approximation
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>

Options for estat

Options for `estat` are presented under the following headings:

[Options for estat gatetest](#)
[Options for estat ate](#)
[Options for estat projection](#)
[Options for estat series](#)
[Options for estat policyeval](#)
[Options for estat classification](#)

Options for estat gatetest

`mtest[(mtest_options)]` specifies that tests be performed for each condition separately. *mtest_options* specifies the method for adjusting *p*-values for multiple tests and can be the following:

`noadjust` specifies that no adjustment is to be made.

`bonferroni` specifies that Bonferroni's method be used.

`holm` specifies that Holm's method be used.

`sidak` specifies that Šidák's method be used.

Specifying `mtest` without an argument is equivalent to specifying `mtest(noadjust)`.

Options for estat ate

`level(#)`; see [\[R\] Estimation options](#).

display_options: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [\[R\] Estimation options](#).

Options for estat projection

`vce(vcetype)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`ols`) or that are robust to some kinds of misspecification (`robust`); see [\[R\] vce_option](#). The default is `vce(robust)`.

`vce(ols)` uses the standard variance estimator for ordinary least-squares regression.

`level(#)`; see [\[R\] Estimation options](#).

`noconstant` suppresses the constant term.

`post` posts the results as the estimation results, so all postestimation commands after `regress` will be available; see [\[R\] regress postestimation](#).

display_options: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

Options for estat series

Model

`bspline` specifies that a third-order B-spline be selected as the basis. It is the default basis.

`bspline(#)` specifies that a B-spline of order `#` be used as the basis. The order may be 1, 2, or 3.

`spline` specifies that a third-order piecewise polynomial spline be selected as the basis.

`spline(#)` specifies that a piecewise polynomial spline of order `#` be used as the basis. The order may be 1, 2, or 3.

`polynomial` specifies that a polynomial be selected as the basis.

`polynomial(#)` specifies that a polynomial of order `#` be used as the basis. The order may be an integer between 1 and 16.

`asis(varlist)` specifies that variables in *varlist* be included as independent variables in the model without any transformation. No B-spline, piecewise polynomial spline, or polynomial basis function will be formed from these variables. Variables in *varlist* may not be specified in *indepvars*_{series}.

`nointeract(seriesvarlist)` specifies that the terms in the basis function formed from variables in *seriesvarlist* not be interacted with the terms of the basis function formed from other variables in *indepvars*_{series}. Covariates specified in *seriesvarlist* must be in *indepvars*_{series}.

`criterion(crittype)` specifies that *crittype* be used to select the optimal number of terms in the basis function. *crittype* may be one of the following: `cv` (cross-validation), `gcv` (generalized cross-validation), `aic` (Akaike's information criterion), `bic` (Schwarz's Bayesian information criterion), or `mallows` (Mallows's C_p). The default is `criterion(cv)`.

`knots(#)` specifies that a piecewise polynomial spline or B-spline basis function with `#` knots be used. The minimum number of knots must be an integer greater than or equal to 1. The maximum number of knots is either 4,096 or two-thirds of the sample size, whichever is smaller.

`knotsmat(matname)` specifies that the knots for each continuous covariate be the values in each row of *matname*. The number of knots should be the same for each covariate, and there must be as many rows as there are continuous covariates. If rows of *matname* are not labeled with varnames, then rows are assumed to be in the order of *indepvars*_{series}.

`distinct(#)` specifies the minimum number of distinct values allowed in continuous variables. By default, continuous variables that enter the basis through either *indepvars*_{series} or *seriesvarlist* are required to have at least 10 distinct values. Continuous variables with few distinct values provide little information for determining an appropriate basis function and may produce unreliable estimates.

SE

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`) and that assume homoskedasticity (`ols`); see [R] [vce_option](#). The default is `vce(robust)`.

Graph

`graph[ (seriesgraph_opts) ]` plots the prediction of CATES. *seriesgraph_opts* may be the following:

`noci` specifies not to plot the confidence intervals.

`cateopts(scatter_opts)` affects the rendition of the predicted CATE point estimates. *scatter_opts* may be the following:

connect_options specify how points on a graph are to be connected; see [G-3] [connect_options](#).

marker_options affect the rendition of markers drawn at the plotted points, including their shape, size, color, and outline; see [G-3] [marker_options](#).

`ciopts(area_options)` affects the rendition of the confidence intervals; see [G-3] [area_options](#).

twoway_options are any of the options documented in [G-3] [twoway_options](#), excluding `by()`. These include options for titling the graph (see [G-3] [title_options](#)) and for saving the graph to disk (see [G-3] [saving_option](#)).

When `graph` or `graph()` is specified, only one variable is allowed in *indepvars*_{series}.

Reporting

`level(#)`; see [R] [Estimation options](#).

`aequations` specifies that the auxiliary regression coefficients be reported. By default, only the average marginal effects of the covariates on the treatment effects are reported.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

Options for estat policyeval

`level(#)`; see [R] [Estimation options](#).

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

Options for estat classification

`unequal` specifies that the unpaired data not be assumed to have equal variances.

`welch` specifies that the approximate degrees of freedom for the test be obtained from Welch's (1947) formula rather than from Satterthwaite's (1946) approximation formula, which is the default when `unequal` is specified. Specifying `welch` implies `unequal`.

`level(#)`; see [R] [Estimation options](#).

categraph

Description for categraph

categraph histogram plots the histogram of the IATE predictions.

categraph gateplot plots the GATE or GATES estimates and their confidence intervals.

categraph iateplot plots the IATE function and the pointwise confidence intervals when one variable is varying and the other variables are fixed at specific values.

Menu for categraph

Statistics > Postestimation

Syntax for categraph

Histogram of the IATE predictions

```
categraph histogram [ if ] [ in ] [ , histogram_options ]
```

Plot of the GATE or GATES estimates

```
categraph gateplot [ , gateplot_options ]
```

Plot of the IATE function estimates

```
categraph iateplot xvar [ if ] [ in ] [ , iateplot_options ]
```

xvar is a variable name specified in catevarlist of cate. It can be a regular variable or a factor variable, but interaction and product notations are not allowed.

gateplot_options	Description
Main	
level(#)	set confidence level; default is level(95)
nocl	do not plot the confidence intervals
Scatter options	
gateopts(scatter_opts)	affect rendition of the predicted GATE or GATES point estimates
CI options	
ciopts(area_opts)	affect rendition of the confidence interval
Y axis, X axis, Titles, Legend, Overall	
twoway_options	any options other than by() documented in [G-3] twoway_options
scatter_opts	
Description	
connect_options	change the look of lines or connecting method
marker_options	change the look of markers (color, size, etc.)

<i>iateplot_options</i>	Description
Model	
* <code>range(#_min #_max)</code>	plot IATE function over <i>xvar</i> equal to <code>#_min</code> to <code>#_max</code> ; the default sets minimum and maximum of <i>xvar</i> in the current dataset
* <code>range(varname)</code>	plot IATE function over <i>xvar</i> equal to minimum and maximum of <i>varname</i>
* <code>n(#)</code>	evaluate at <code>#</code> points; default is 300 points
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>at(atspec)</code>	set values for all <i>catevarlist</i> except <i>xvar</i>
IATE plot	
<code>iateopts(scatter_opts)</code>	affect rendition of the predicted IATE point estimates
CI	
<code>noci</code>	do not plot the confidence intervals
<code>ciopts(area_opts)</code>	affect rendition of the confidence interval
Y axis, X axis, Titles, Legend, Overall	
<code>twoway_options</code>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>

*`range()` and `n()` are not allowed if *xvar* is a factor variable in *catevarlist* of *cate*.

Options for categraph

Options for categraph are presented under the following headings:

- [Options for categraph histogram](#)
- [Options for categraph gateplot](#)
- [Options for categraph iateplot](#)

Options for categraph histogram

histogram_options are options in [R] [histogram](#).

Options for categraph gateplot

Main
<code>level(#)</code> ; see [R] Estimation options .
<code>noci</code> specifies not to plot the confidence intervals.
Scatter options
<code>gateopts(scatter_opts)</code> affects the rendition of the predicted GATE or GATES point estimates. <i>scatter_opts</i> may be the following:
<i>connect_options</i> specify how points on a graph are to be connected; see [G-3] connect_options .
<i>marker_options</i> affect the rendition of markers drawn at the plotted points, including their shape, size, color, and outline; see [G-3] marker_options .
CI options
<code>ciopts(area_options)</code> affects the rendition of the confidence intervals; see [G-3] area_options .

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] *twoway_options*, excluding `by()`. These include options for titling the graph (see [G-3] *title_options*) and for saving the graph to disk (see [G-3] *saving_option*).

Options for `categraph iateplot`

Model

`range(#_min #_max)` or `range(varname)` plots the IATE function over *xvar* between `#_min` and `#_max` or between the minimum or maximum of *varname*, respectively, while holding other variables in *catevarlist* of *cate* fixed at some values. The default sets `#_min` and `#_max` to the minimum and maximum of *xvar* in the current dataset. See `at()` below for details on fixing values for the variables other than *xvar*.

`n(#)` evaluates the IATE function at `#` points. The points on *xvar* are evenly spaced between the minimum and the maximum specified in `range()`. The other variables in *catevarlist* are fixed at some values specified in `at()`. The default is `n(300)`.

`level(#)`; see [R] *Estimation options*.

`at(atspec)` specifies values for all the covariates (except *xvar*) in *catevarlist* of *cate* to be treated as fixed.

atspec may contain one or more of the following specifications:

(*stat*) *varlist*

varname = #

where

1. Variable names (whether in *varlist* or *varname*) must be the covariates in *catevarlist* other than *xvar* in the *cate* estimation.
2. Variable names may be continuous variables or factor variables.
3. *varlist* may also be one of three standard lists:
 - (a) `_all` (all covariates),
 - (b) `_factor` (all factor-variable covariates), or
 - (c) `_continuous` (all continuous covariates).

4. *stat* may be any of the following:

<i>stat</i>	Description	Variables allowed
mean	means (default for continuous variables)	continuous
base	base level (default for factor variables)	factors
median	medians	continuous
p1	1st percentile	continuous
p2	2nd percentile	continuous
...	3rd–49th percentiles	continuous
p50	50th percentile (same as median)	continuous
...	51st–97th percentiles	continuous
p98	98th percentile	continuous
p99	99th percentile	continuous
min	minimums	continuous
max	maximums	continuous
zero	fixed at zero	continuous

IATE plot

`iateopts(scatter_opts)` affects the rendition of the predicted IATE point estimates. `scatter_opts` may be the following:

`connect_options` specify how points on a graph are to be connected; [G-3] [connect_options](#).

`marker_options` affect the rendition of markers drawn at the plotted points, including their shape, size, color, and outline; see [G-3] [marker_options](#).

CI

`noci` specifies not to plot the confidence intervals.

`ciopts(area_options)` affects the rendition of the confidence intervals; see [G-3] [area_options](#).

Y axis, X axis, Titles, Legend, Overall

`twoway_options` are any of the options documented in [G-3] [twoway_options](#), excluding `by()`. These include options for titling the graph (see [G-3] [title_options](#)) and for saving the graph to disk (see [G-3] [saving_option](#)).

Remarks and examples

For an overview of cate postestimation tools and the examples that demonstrate how to use the cate command and its postestimation tools, see details in [Remarks and examples](#) in [CAUSAL] [cate](#).

The `estimates` commands after the `cate` command work the same as they do after other estimation commands with only one difference: `estimates save filename` saves three files, not just one. `filename.ster`, `filename.stgrf`, and `filename.stxer` are saved. See [R] [estimates](#) for details.

Stored results

`estat heterogeneity` stores the following in `r()`:

Scalars

<code>r(p)</code>	two-sided p -value
<code>r(df)</code>	test constraints degrees of freedom
<code>r(chi2)</code>	χ^2

Matrices

<code>r(b)</code>	coefficient vector in the best linear prediction of IATE
<code>r(V)</code>	variance–covariance matrix of the estimators

`estat gatetest` stores the following in `r()`:

Scalars

<code>r(p)</code>	two-sided p -value
<code>r(df)</code>	test constraints degrees of freedom
<code>r(chi2)</code>	χ^2
<code>r(drop)</code>	1 if constraints were dropped, 0 otherwise

`estat ate` stores the following in `r()`:

Scalars

<code>r(N)</code>	number of observations
-------------------	------------------------

Matrices

<code>r(b)</code>	coefficient vector
<code>r(V)</code>	variance–covariance matrix of the estimators
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals

`estat projection` stores the following in `r()`:

Scalars

<code>r(N)</code>	number of observations
<code>r(mss)</code>	model sum of squares
<code>r(df_m)</code>	model degrees of freedom
<code>r(rss)</code>	residual sum of squares
<code>r(df_r)</code>	residual degrees of freedom
<code>r(r2)</code>	R^2
<code>r(r2_a)</code>	adjusted R^2
<code>r(F)</code>	F statistic
<code>r(rmse)</code>	root mean squared error
<code>r(ll)</code>	log likelihood under additional assumption of independent and identically distributed normal errors
<code>r(ll_0)</code>	log likelihood, constant-only model
<code>r(rank)</code>	rank of <code>r(V)</code>

Matrices

<code>r(b)</code>	coefficient vector
<code>r(V)</code>	variance–covariance matrix of the estimators
<code>r(beta)</code>	standardized coefficients
<code>r(V_modelbased)</code>	model-based variance
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals

`estat projection` with the `post` option stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(mss)</code>	model sum of squares
<code>e(df_m)</code>	model degrees of freedom
<code>e(rss)</code>	residual sum of squares
<code>e(df_r)</code>	residual degrees of freedom
<code>e(r2)</code>	R^2
<code>e(r2_a)</code>	adjusted R^2
<code>e(F)</code>	F statistic
<code>e(rmse)</code>	root mean squared error
<code>e(ll)</code>	log likelihood under additional assumption of i.i.d. normal errors
<code>e(ll_0)</code>	log likelihood, constant-only model
<code>e(rank)</code>	rank of <code>e(V)</code>

Macros

<code>e(cmd)</code>	<code>regress</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(model)</code>	ols
<code>e(title)</code>	title in estimation output when <code>vce()</code> is not <code>ols</code>
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	b V
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsok)</code>	predictions allowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators
<code>e(beta)</code>	standardized coefficients
<code>e(V_modelbased)</code>	model-based variance

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

`estat series` stores the following in `r()`:

Scalars

<code>r(N)</code>	number of observations
<code>r(converged)</code>	1 if converged, 0 otherwise
<code>r(order)</code>	order of basis function
<code>r(rank)</code>	rank of <code>r(V)</code>

Matrices

<code>r(b)</code>	coefficient vector
<code>r(V)</code>	variance–covariance matrix of the estimators
<code>r(V_modelbased)</code>	model-based variance
<code>r(ilog)</code>	iteration log (up to 20 iterations)
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals

`estat policyeval` stores the following in `r()`:

Scalars

`r(N)` number of observations

Macros

`r(policy_var1)` first policy variable name
`r(policy_var2)` second policy variable name

Matrices

`r(b)` coefficient vector
`r(V)` variance–covariance matrix of the estimators
`r(table)` matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals

`estat classification` stores the following in `r()`:

Scalars

`r(N_1)` sample size n_1
`r(N_2)` sample size n_2
`r(p_l)` lower one-sided p -value
`r(p_u)` upper one-sided p -value
`r(p)` two-sided p -value
`r(se)` estimate of standard error
`r(t)` t statistic
`r(sd_1)` standard deviation for population 1
`r(sd_2)` standard deviation for population 2
`r(sd)` combined standard deviation
`r(mu_1)` $\bar{x}_1$ mean for population 1
`r(mu_2)` $\bar{x}_2$ mean for population 2
`r(df_t)` degrees of freedom
`r(level)` confidence level

`categraph iateplot` stores the following in `r()`:

Macros

`r(xvar)` variable allowed to vary
`r(vtype_list)` types of variables other than `r(xvar)`
`r(vname_list)` names of variables other than `r(xvar)`
`r(stat_list)` statistics of variables other than `r(xvar)`

Matrices

`r(at)` matrix of values from the `at()` options

Methods and formulas

Methods and formulas are presented under the following headings:

IATE predictions

Test of treatment-effects heterogeneity

Test of group-level treatment-effects heterogeneity

ATE for a subsample

Linear or nonparametric series projection of the IATE on variables

Treatment-assignment policy evaluation

Classification analysis

For notational simplicity, we drop the subscript i indicating the i th observation to refer to a random variable.

IATE predictions

`predict` predicts the IATEs, their standard errors, or the IATE confidence intervals. The IATEs can be estimated by either a generalized random forest or a parametric linear regression, which is specified in the `cmethod()` option of `cate`. For details of the random-forest-based IATE predictions, their standard errors, and the confidence intervals, see [Generalized random forest](#) in [\[CAUSAL\] cate](#). For details on linear-regression-based predictions and their standard errors, see the discussions in [Methods and formulas](#) of [\[R\] predict](#); the confidence intervals are obtained via the delta method.

Test of treatment-effects heterogeneity

`estat heterogeneity` tests the null hypothesis that the treatment effects are homogeneous. In particular, it implements the test proposed in [Chernozhukov et al. \(2006\)](#). Let $\tau_0(\mathbf{x})$ be the true IATE function, $\hat{\tau}(\mathbf{x})$ be an estimate of the IATE function, $\bar{\tau}$ be $\mathbf{E}\{\hat{\tau}(\mathbf{x})\}$, and $\tilde{\tau}$ be the sample average of $\hat{\tau}(\mathbf{x})$. Then the best linear prediction of $\tau_0(\mathbf{x})$ conditional on $\hat{\tau}(\mathbf{x})$ is given by

$$\tau_0(\mathbf{x}) = \gamma_1 \bar{\tau} + \gamma_2 \{\hat{\tau}(\mathbf{x}) - \bar{\tau}\} + \epsilon$$

where ϵ is the error term.

If $\gamma_2 = 0$, it implies that the $\hat{\tau}(\mathbf{x})$ predictions are pure noise, and it also means that $\tau_0(\mathbf{x})$ is constant or homogeneous. Thus, to test the null hypothesis that the treatment effects are homogeneous, we perform a Wald test of $\gamma_2 = 0$.

In the partialing-out estimator, the coefficients of γ_1 and γ_2 can be identified by fitting the following regression:

$$y - \hat{h}(\mathbf{x}, \mathbf{w}) = \gamma_1 \tilde{\tau} \{d - \hat{f}(\mathbf{x}, \mathbf{w})\} + \gamma_2 \{\hat{\tau}(\mathbf{x}) - \tilde{\tau}\} \{d - \hat{f}(\mathbf{x}, \mathbf{w})\} + \epsilon$$

where y is the outcome variable, $\hat{h}(\mathbf{x}, \mathbf{w})$ estimates $\mathbf{E}(y|\mathbf{x}, \mathbf{w})$, d is the treatment variable, and $\hat{f}(\mathbf{x}, \mathbf{w})$ estimates $\mathbf{E}(d|\mathbf{x}, \mathbf{w}) \equiv \mathbf{P}(d = 1|\mathbf{x}, \mathbf{w})$.

In the augmented inverse-probability weighting (AIPW) estimator, the best linear prediction of $\tau_0(\mathbf{x})$ conditional on $\hat{\tau}(\mathbf{x})$ can be obtained by regressing the AIPW scores implied by the full interactive model on $\tilde{\tau}$ and $\hat{\tau}(\mathbf{x}) - \tilde{\tau}$.

Test of group-level treatment-effects heterogeneity

`estat gatetest` tests the null hypothesis that the ATEs are the same among the specified subgroup levels. It performs Wald tests on the GATE estimates' coefficients. For details of Wald tests, see [Methods and formulas](#) in [\[R\] test](#).

ATE for a subsample

`estat ate` computes the ATE for a subsample by taking the average of the AIPW scores implied by the model over the subsample, which is proposed in [Chernozhukov et al. \(2018\)](#) and [Knaus \(2022\)](#). For details of the AIPW scores in the partial linear and the fully interactive models, see [Methods and formulas](#) in [\[CAUSAL\] cate](#).

Linear or nonparametric series projection of the IATE on variables

`estat projection` computes the linear projection of the IATE function on the specified variables. Similarly, `estat series` computes the nonparametric series projection of the IATE function on the basis functions formed by the specified variables. The linear projection is a special case of the series projection that uses the basis functions as the variables themselves. Thus, we only need to discuss the methods for `estat series`, because `estat projection` is just a special case.

`estat series` implements the methods proposed by [Semenova and Chernozhukov \(2021\)](#) by running a series regression of the AIPW scores implied by the model on the basis functions formed by the specified variables. For details of the AIPW scores in the partial linear and the fully interactive models, see [Methods and formulas](#) in [\[CAUSAL\] cate](#). For a discussion of nonparametric series regression, see [Methods and formulas](#) in [\[R\] npregress series](#).

Treatment-assignment policy evaluation

`estat policyeval` or `estat tassigneval` evaluates treatment-assignment policies. Suppose a treatment-assignment rule assigns individuals to be treated or not treated. We want to evaluate this treatment-assignment rule by answering questions such as the following:

1. If we implement such a rule, what is the average outcome of the population?
2. Furthermore, if we have two different rules, which is better?

For the first question, we compute the average of the outcome if the treatment is assigned according to a rule. We estimate

$$\Pi(\pi) = \mathbf{E}[\pi(\mathbf{x})y(1) + \{1 - \pi(\mathbf{x})\}y(0)]$$

where $y(1)$ is the potential outcome when it is treated, $y(0)$ is the potential outcome when it is not treated, and $\pi(\mathbf{x}) \in [0, 1]$ is a prespecified treatment-assignment probability, which is also known as a policy. $\Pi(\pi)$ is also called the value of the policy π .

For the second question, we compute the difference of the values between two policies, π_1 and π_2 . In particular, we compute the contrast of the values between the two treatment-assignment policies.

$$\Pi(\pi_1) - \Pi(\pi_2)$$

For details of the potential outcomes in the partial linear and the fully interactive models, see [Methods and formulas](#) in [\[CAUSAL\] cate](#).

Classification analysis

`estat classification` performs a classification analysis of the groups constructed based on the sorted IATE estimates. It performs a two-sample t test to compare the mean of a variable between the least and the most affected groups. For details of t tests on the equality of means, see [Methods and formulas](#) in [\[R\] ttest](#).

References

Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. B. Hansen, W. K. Newey, and J. M. Robins. 2018. Double/debiased machine learning for treatment and structural parameters. *Econometrics Journal* 21: C1–C68. <https://doi.org/10.1111/ectj.12097>.

- Chernozhukov, V., M. Demirer, E. Duflo, and I. Fernández-Val. 2006. Generic machine learning inference on heterogeneous treatment effects in randomized experiments, with an application to immunization in India. NBER Working Paper 24678, National Bureau of Economic Research. <https://doi.org/10.3386/w24678>.
- Knaus, M. C. 2022. Double machine learning-based programme evaluation under unconfoundedness. *Econometrics Journal* 25: 602–627. <https://doi.org/10.1093/ectj/utac015>.
- Satterthwaite, F. E. 1946. An approximate distribution of estimates of variance components. *Biometrics Bulletin* 2: 110–114. <https://doi.org/10.2307/3002019>.
- Semenova, V., and V. Chernozhukov. 2021. Debiased machine learning of conditional average treatment effects and other causal functions. *Econometrics Journal* 24: 264–289. <https://doi.org/10.1093/ectj/utaa027>.
- Welch, B. L. 1947. The generalization of ‘student’s’ problem when several different population variances are involved. *Biometrika* 34: 28–35. <https://doi.org/10.2307/2332510>.

Also see

[CAUSAL] **cate** — Conditional average treatment-effects estimation

[U] **20 Estimation and postestimation commands**

Description

Difference in differences (DID) is a method to estimate the average effect of a treatment on those who receive the treatment. The method can be applied to two types of observational data: repeated cross-sections, in which different individuals are observed at different time points, and panel data, in which the same individuals are observed over time. We provide commands that estimate the average treatment effect on the treated (ATET). The effect may be the same for all (homogeneous) or may differ across time and across groups (heterogeneous). `didregress` and `xtdidregress` are for homogeneous treatment effects; `hdidregress` and `xthdidregress` are for heterogeneous treatment effects.

`xtswitchdid` estimates event-study effects for panel data when treatments may not be binary and switching. In other words, treatments are allowed to take more than two discrete values, and treated groups can move between different levels of treatment over time or out of treatment altogether.

`didregress` and `xtdidregress` estimate the ATET of a binary or continuous treatment on a continuous outcome by fitting a linear model with time fixed effects and group or panel fixed effects. These commands also estimate difference in difference in differences (DDD), in which we augment the DID framework to include additional control groups to obtain the ATET.

`hdidregress` and `xthdidregress` estimate ATETs that vary over time and over treatment cohorts. Treatment cohorts are groups that are subject to intervention at different points in time. As in the homogeneous case, we can fit a linear model with time fixed effects and group or panel fixed effects, but we also incorporate interactions of the treatment with time and treatment cohorts. We fit these models using the two-way fixed-effects (TWFE) estimator. `hdidregress` and `xthdidregress` additionally allow for regression adjustment (RA), inverse-probability weighting (IPW), and augmented inverse-probability weighting (AIPW) to estimate the ATETs. See [\[CAUSAL\] teffects intro](#) for a discussion of RA, AIPW, and IPW estimators.

`xtswitchdid` estimates event-study effects for panel data using the heterogeneous DID estimator of [de Chaisemartin and D'Haultfœuille \(2026\)](#), which allows for multivalued and switching, or nonabsorbing, treatment variables. These event-study effects can be considered the average effect of being exposed to a treatment weakly stronger than a group's initial treatment state after a given number of periods. These effects are dynamic, meaning they can include the effect of treatment values from past periods. In the binary, absorbing case, the event-study effects in `xtswitchdid` can be compared with the dynamic aggregation of ATETs after `xthdidregress` that are estimated by using `estat aggregation, dynamic`.

Remarks and examples

Remarks are presented under the following headings:

- Introduction*
- Intuition for estimating effects*
- DID with heterogeneous treatment effects*
- Standard error considerations*
- Different types of data and specification*
 - Specifying groups and time as binary indicators*
 - Excluding group and time effects*
 - Exploring treatment-effect heterogeneity*
- DID with switching treatments for panel data*
- Conclusion*

This entry presents the intuition and some of the technical details for the estimators in `didregress`, `xtdidregress`, `hdidregress`, `xthdidregress`, and `xtswitchdid` and the diagnostics available after estimation. See [\[CAUSAL\] didregress](#), [\[CAUSAL\] didregress postestimation](#), [\[CAUSAL\] hdidregress](#), [\[CAUSAL\] xthdidregress](#), [\[CAUSAL\] hdidregress postestimation](#), [\[CAUSAL\] xtswitchdid](#), and [\[CAUSAL\] xtswitchdid postestimation](#) for details on the syntax and worked examples. For a more complete discussion and references on homogeneous DID, see [Angrist and Pischke \(2009\)](#), [Blundell and Dias \(2009\)](#), [Imbens and Wooldridge \(2009\)](#), [Lechner \(2011\)](#), [Angrist and Pischke \(2015\)](#), [Abadie and Cattaneo \(2018\)](#), and [Wing, Simon, and Bello-Gomez \(2018\)](#). For more details on heterogeneous DID, see [Roth et al. \(2022\)](#) and [de Chaisemartin and D’Haultfœuille \(2023\)](#) and the references therein. For details on event-study effects for panel DID with switching treatments, see [de Chaisemartin and D’Haultfœuille \(2026\)](#).

Introduction

DID is one of the most venerable causal inference methods used by researchers. DID provides estimates of treatment effects on those that receive a treatment. Examples of such treatment effects are the effect of a minimum-wage increase on employment for those that see their minimum wage increase or the effect of water pollution on health outcomes for those that were exposed to water pollution.

Unlike the treatment-effects estimators described in [\[CAUSAL\] teffects](#), which are applied to data from one cross-section, DID allows us to consider variation over time. For example, DID allows for panel data, for which we observe the same unit over time. It also works for repeated cross-sections, for which individuals sampled within groups differ across time periods but the groups sampled are the same. An appealing feature of DID is that our model specification does not require us to control for individual characteristics to identify treatment effects, something that is fundamental to the estimators described in [\[CAUSAL\] teffects](#).

When thinking about ATETs in a DID framework, we compare a group before and after the treatment, perhaps by looking at a graph such as the one below:



Figure 1.

A treatment occurred in the year 2010. This might be a government policy, a change in medicine dosage, or any other intervention of interest. We would like to know if the treatment had a causal effect. It is clear from the graph that the outcome of interest changed after 2010. Is this due to the treatment or is something else occurring? Perhaps there are unobserved time effects that affect the treatment group after the treatment. For instance, there could have been a change in weather conditions or an economic downfall that affected the treatment group but was not captured in the model or the data. If this is the case, it does not suffice to look at the treatment group before and after the policy. DID addresses this by finding a control group, that is, a group that was subject to the same unobserved time effects but was not exposed to the treatment. Comparing the treatment group with the control group before and after the treatment might give us a better understanding of whether the treatment made a difference. A graph looking at a treatment and control group might look like this:

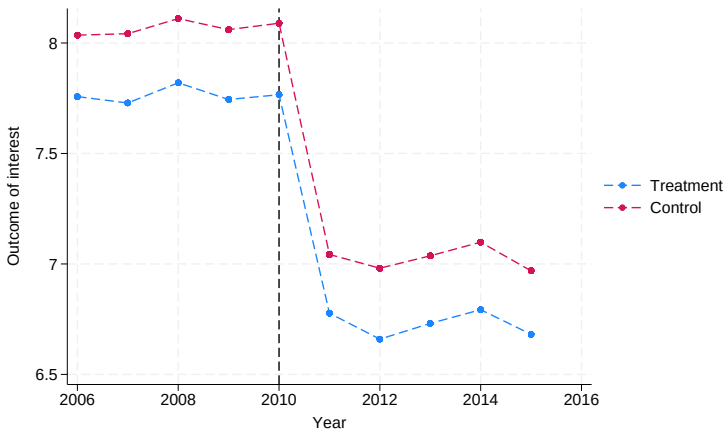


Figure 2.

For both the treatment and the control group, we see that there was a decrease in the mean of the outcome after 2010. Therefore, the decrease we saw in the treatment group cannot be attributable entirely to the treatment. (In fact, these are simulated data, and we know the treatment has no effect.)

In a DID setup, if the treated group had not received the treatment, we would expect the treated and control groups to experience the same trends. A treatment effect implies a systematic deviation from a common trend that can be observed graphically. This is what we observe:

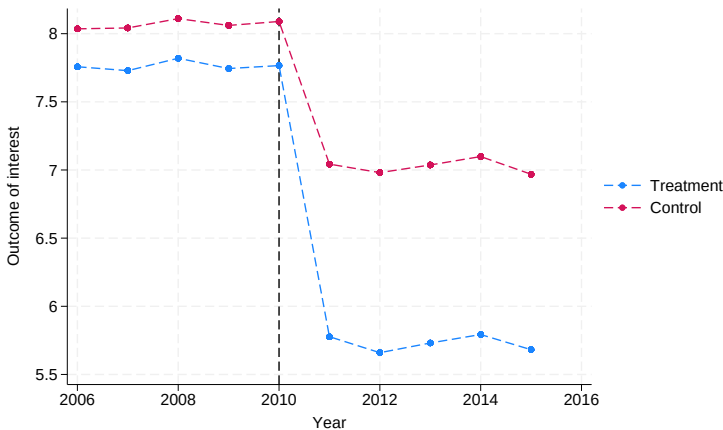


Figure 3.

Here both groups experienced a decrease after 2010, but the treatment-group decrease was more substantial. The difference in the decreases across groups may indicate the effect of the treatment.

Researchers may motivate their analysis with such graphs. However, graphical evidence is not enough. We need statistical validation, so we fit a model.

The ATET is consistently estimated by differencing the average outcome for the treatment and control groups over time to eliminate time-invariant unobserved characteristics and also differencing the average outcome of these groups to eliminate time-varying unobserved effects common to both groups. These two differences give the method its name and highlight its intuitive appeal. More appealing is the fact that you can get the effect of interest, the ATET, from one parameter in a linear regression.

When talking about DID, people cite [Snow \(1849\)](#) and [Snow \(1855\)](#) as the first known applications. Snow claimed that cholera was not transmitted by contaminated air or contaminated blood, as was thought by some academics of his time. Snow hypothesized the disease was communicated via water that had been polluted with sewage. Below, he describes how he came up with an idea for a natural experiment to validate his hypothesis:

In Thomas Street, Horsleydown, there are two courts close together, consisting of a number of small houses or cottages, inhabited by poor people. The houses occupy one side of each court or alley—the south side of Trusscott’s Court, and the north side of the other, which is called Surrey Buildings, being placed back to back, with an intervening space, divided into small back areas, in which are situated the privies of both the courts, communicating with the same drain, and there is an open sewer which passes the further end of both courts. Now, in Surrey’s buildings the cholera has committed fearful devastation, whilst in the adjoining court there has been but one fatal case, and another case that ended in recovery. In the former court the slops of dirty water poured down by the inhabitants into a channel in front of the houses got into the well from which they obtained their water, this being the only difference

In the first edition ([1849](#)) of the text, Snow reports the deaths from cholera from September 23, 1848, to August 25, 1849, for five London districts. The number of deaths is higher in the South and East districts relative to the other three districts, arising from the source of their water supply. Snow obtains a clear motivation for his theory. In the second edition ([1855](#)), he collects data before and after a pump with contaminated water in Broadstreet, London, is closed. It is then that he can compare a treated with a control group before and after a treatment to establish a treatment effect.

John Snow (1813–1858) was born in York, England. From age 14, he worked as an apprentice and assistant to surgeons in northeast England and Yorkshire. In 1836, Snow moved to London; he was admitted to the Royal College of Surgeons in 1838 and the Royal College of Physicians in 1850. He made outstanding contributions to the adoption of anesthesia and is considered one of the originators of modern epidemiology. Snow died following a stroke in 1858.

Snow calculated dosages for ether and chloroform. He personally administered chloroform to Queen Victoria for the births of her last two children, which helped obstetric anesthesia gain wider acceptance.

Snow was skeptical of the miasma theory that cholera was caused by foul air. His essay *On the Mode of Communication of Cholera* was first published in 1849 and then greatly enlarged in 1855 with the results of his very detailed investigation of the role of water supply in the epidemic of 1854 in the Soho district of London. Snow identified the source of the outbreak as the public water pump on Broad Street (now Broadwick Street), leading the local council to remove the pump handle. It was later discovered that the well had been dug very close to an old cesspit. He also mapped the clustering of cholera cases around the pump and related mortality to water sources, clearly showing higher deathrates in areas supplied by the Southwark and Vauxhall Waterworks Company, which was taking water from sewage-polluted sections of the River Thames. Snow is widely regarded as a pioneer in public health, epidemiology, and medical geography.

Intuition for estimating effects

We can build our intuition about the causal inference implied by the DID setup by using the potential-outcomes framework described in [\[CAUSAL\] Intro](#), [\[CAUSAL\] teffects intro](#), and [\[CAUSAL\] teffects intro advanced](#). We consider individual-level data for which we sample different individuals at different points in time, that is, a repeated cross-section. The treatment occurs at the group level. For example, the treatment may occur at the state, county, or hospital level. All individuals in a given state, county, or hospital either are treated or are controls at a given point in time. We index individuals by i , groups by g , and time by t . We are interested in the effect of a treatment, $D_{igt} \in \{0, 1\}$, on an outcome, Y_{igt} . Suppose the potential-outcome mean of an individual in group g at time t that does not receive the treatment is given by the following:

$$E\{Y_{igt}(0) \mid g, t\} = \gamma_g + \gamma_t \quad (1)$$

Above, γ_g denotes the group effects, and γ_t denotes the time effects. Also suppose the potential-outcome mean for someone who receives the treatment is given by the following:

$$E\{Y_{igt}(1) \mid g, t\} = \gamma_g + \gamma_t + \delta \quad (2)$$

The potential outcomes described above allow us to think of the regression model

$$Y_{igt} = \gamma_g + \gamma_t + D_{gt}\delta + \epsilon_{igt}$$

A regression estimate of δ , the coefficient on the indicator of treatment, consistently estimates the ATET in this simplified framework, if we meet the overlap assumption, the stable unit treatment value assumption, and conditional independence (described in [\[CAUSAL\] Intro](#) and [\[CAUSAL\] teffects intro advanced](#)), plus one additional assumption.

To introduce this additional assumption, we think it is sometimes more intuitive to look at a two-period, two-group example. In this case, $g \in \{0, 1\}$, where 0 is the control group and 1 is the treatment group, and $t \in \{0, 1\}$, where 0 is the period before the treatment and 1 occurs after the treatment. To guarantee a consistent estimate of the ATET, we need to make the parallel-trends assumption:

$$E(Y_{ig1}(0) | D_{gt} = 1) - E(Y_{ig0}(0) | D_{gt} = 1) = E(Y_{ig1}(0) | D_{gt} = 0) - E(Y_{ig0}(0) | D_{gt} = 0)$$

The parallel-trends assumption is stated in terms of the potential outcomes of not being treated, $Y_{igt}(0)$, conditional on treatment, D_{gt} . It implies that if the treated had not received the treatment, the groups defined by $D_{gt} = 1$ and $D_{gt} = 0$ would have the same trends. For this to be true, we need group effects to be time invariant and time effects to be group invariant. The simple framework described in (1) and (2) satisfies the parallel-trends assumption.

The parallel-trends assumption has a graphical representation. Let's think again about the case with multiple time periods. The parallel-trends assumption implies the paths of the outcome variable over time are parallel between the control and treatment groups prior to the date of the treatment. We can visually check this assumption by plotting the means of the outcome over time for both groups or by visualizing the results of the linear trends model. For instance, we might use a graph like the one in [figure 2](#), where we plotted the means over time. After fitting a model using `didregress` and `xtdidregress`, you can get both the mean outcome plot and the trends plot by typing

```
. estat trendplots
```

Another way to think about the parallel-trends assumption in the pretreatment period is that treatment and control groups do not change their behavior in anticipation of the treatment. We can think of the parallel-trends assumption as implying that there should be no treatment effect in anticipation of the treatment. To test this assumption, we could fit a Granger-type causality model where we augment our model with dummies for each pretreatment period for the treated observations. A joint test of the coefficients on these dummies against 0 can be used as a test of the null hypothesis that no anticipatory effects have taken place. We can perform this test by typing

```
. estat granger
```

DID with heterogeneous treatment effects

When we introduced DID estimation above, we built our intuition for a case in which the treatment effect is the same for every group and in which the treatment effect does not change over time. We imposed this when we wrote

$$\begin{aligned} E\{Y_{igt}(0) | g, t\} &= \gamma_g + \gamma_t \\ E\{Y_{igt}(1) | g, t\} &= \gamma_g + \gamma_t + \delta \end{aligned}$$

Also, we discussed the model using two groups and two time periods, a 2×2 model. But nothing precludes us from thinking of a set of treatment effects, say, δ_{gt} , that varies over multiple groups, g , and time periods, t . More importantly, when we assume that the effect is homogeneous but the true model is heterogeneous, our treatment-effect estimates are going to be inconsistent, as was shown by [de Chaisemartin and D'Haultfœuille \(2020\)](#), [Borusyak, Jaravel, and Spiess \(2021\)](#), [Sun and Abraham \(2021\)](#), and [Goodman-Bacon \(2021\)](#).

Goodman-Bacon (2021) characterizes the ATET estimate obtained from a DID of the form

$$Y_{igt} = \gamma_g + \gamma_t + D_{gt}\delta + \epsilon_{igt}$$

when there are multiple time periods and treatment occurs at different points in time as a deviation from the 2×2 framework. He shows the estimated coefficient, $\hat{\delta}$, is a weighted average of contrasts between groups treated at different points in time and groups that are not treated. When contrasts occur over time between treated groups and groups that are never treated or not yet treated, the comparisons are informative. When contrasts occur between groups that are both already treated, this component of the weighted average is not a treatment effect and thus introduces bias. The bias can be significant and change not only the magnitude but also the sign of the coefficient.

Goodman-Bacon (2021) decomposes $\hat{\delta}$ and estimates all contrasts and weights that lead to the homogeneous estimate. For balanced-panel datasets, after `xtddidregress`, you can get the decomposition of the ATET by typing

```
. estat bdecomp
```

The result will let you know if the homogeneity assumption is warranted. This would be the case if the coefficients of the different contrasts are similar to the ATET estimate. If the contrasts differ substantially from the ATET estimate, our estimates are misleading.

If the decomposition suggests the homogeneous treatment effect is unwarranted or you do not want to impose homogeneity in treatment, you may use `xthdidregress`. If you are concerned about heterogeneity, you may choose from two workflows. With balanced panels, you might choose to fit the model for homogeneous treatment effects first and use `estat bdecomp` to determine whether effects are heterogeneous. With any panel or repeated cross-sectional data, you can fit a heterogeneous treatment-effects model and then use `estat aggregation` to ascertain whether effects are heterogeneous.

The treatment effects you are estimating using `hddidregress` and `xthdidregress` are of the form

$$\text{ATET}(c, t)$$

where c corresponds to the moment in time when a group receives the treatment, a cohort, and t corresponds to time. For example, a policy might be administered at the state level starting in 1995 for some states and in 1998 for other states. In this case, we would have treatment effects of the form $\text{ATET}(1995, t)$ and $\text{ATET}(1998, t)$. We are saying that the effect of the policy is different for states treated in 1995 than it is for those treated in 1998. Also, we are saying that the effect changes over time. Thus, if our sample goes from 1993 to 2000, for the 1995 cohort, we have treatment effects of the form $\text{ATET}(1995, 1993)$, $\text{ATET}(1995, 1994)$, ..., $\text{ATET}(1995, 2000)$.

As is the case for homogeneous treatment, we need to satisfy a parallel-trends assumption and no anticipation of treatment. Both of these assumptions now need to hold for each cohort. As with the homogeneous treatment effects, we provide graphical diagnostics and tests. For a test of parallel trends, we have the command

```
. estat ptrends
```

For graphical inspection, you may type

```
. estat atetplot
```

which results in a graph for each cohort before and after treatment that lets you see whether there is no treatment effect, before treatment, for each cohort.

For a discussion of these assumptions and the requirements needed for consistency of the heterogeneous treatment implemented in `hdidregress` and `xthdidregress`, see [Wooldridge \(2021\)](#), [Callaway and Sant’Anna \(2021\)](#), and [Roth et al. \(2022\)](#).

Standard error considerations

While a standard linear regression model can be used to estimate the ATET in the homogeneous case, when assuming homogeneous treatment, the best estimate of the standard error requires some consideration. Many standard-error estimates have been proposed, and each one performs differently depending on the type of DID model being fit and the structure of the data. Below, we provide a discussion of some of the issues centered on the available standard-error estimates for `hdidregress`, `didregress`, `xthdidregress`, and `xtdidregress`. For a more complete discussion of inference for the homogeneous treatment-effect estimators, see [Cameron and Miller \(2015\)](#) and [MacKinnon \(2019\)](#) and the references therein. For a more complete discussion of the heterogeneous case, see [Wooldridge \(2021\)](#), [Callaway and Sant’Anna \(2021\)](#), and [Roth et al. \(2022\)](#). We begin our discussion with `didregress` and `xtdidregress`.

[Bertrand, Duflo, and Mullainathan \(2004\)](#) show that the standard errors for DID estimates are inconsistent if they do not account for the serial correlation of the outcome of interest. Because the outcomes studied usually vary at the group and time levels, it makes sense to correct for serial correlation. The authors show that using cluster-robust standard errors at the group level where treatment occurs provides correct coverage in the presence of serial correlation when the number of groups is not too small. [Bester, Conley, and Hansen \(2011\)](#) further show that using cluster-robust standard errors and using critical values of a t distribution with $G - 1$ degrees of freedom, where G is the number of groups, is asymptotically valid for a fixed number of groups and a growing sample size. In other words, you do not need the number of groups to be arbitrarily large, that is, to grow asymptotically. Cluster-robust standard errors with $G - 1$ degrees of freedom are the default standard errors of `didregress` and `xtdidregress`.

The results of [Bertrand, Duflo, and Mullainathan \(2004\)](#) and [Bester, Conley, and Hansen \(2011\)](#) demonstrate that we could still obtain reliable standard errors even when the number of groups was not large. But what about data with a very small number of groups? Several simulation and theoretical results suggest that cluster-robust standard errors may still have poor coverage when the number of groups is very small or when the number of treated groups is small relative to the number of control groups. These scenarios with small group sizes are common in DID studies, and another method of standard error estimation should be considered in these situations.

When the number of groups is small, we provide three alternatives. The first alternative is to use the wild cluster bootstrap that imposes the null hypothesis that the ATET is 0. [Cameron, Gelbach, and Miller \(2008\)](#) and [MacKinnon and Webb \(2018\)](#) show that the wild cluster bootstrap provides better inference than using cluster-robust standard errors with $t(G - 1)$ critical values. The second alternative comes from [Imbens and Kolesár \(2016\)](#), who show that with a small number of groups, you may use bias-corrected standard errors with the degrees-of-freedom adjustment proposed by [Bell and McCaffrey \(2002\)](#). For the third alternative, you may use aggregation-type methods like those proposed by [Donald and Lang \(2007\)](#); they show that their method works well when the number of groups is small but the number of individuals in each group is large.

When the disparity between treatment and control groups is large, for example, because there is only one treated group or because the group sizes vary greatly, cluster-robust standard errors and the other methods mentioned above underperform. Yet the bias-corrected and cluster-bootstrap methods provide an improvement over the cluster-robust standard errors.

What we said above for `didregress` and `xtdidregress` applies to `hdidregress` and `xthdidregress` when the underlying estimator is a linear regression. Considerations are different when we talk about the RA, AIPW, and IPW estimators implemented by the heterogeneous DID commands. These last three estimators can be understood as estimates from a multiple-equation model, in which each $ATET(c, t)$ could have been estimated separately. The standard errors of the estimates are, by default, equivalent to what you would obtain using a method of moments estimator and clustering at the group level where the treatment is administered; see [R] [gmm](#).

Because heterogeneous DID has multiple equations and parameters, you may want to consider inference of all the $ATET(c, t)$ estimates simultaneously: uniform inference. For such a scenario, you may use the bootstrap procedure suggested by [Callaway and Sant’Anna \(2021\)](#) by typing

```
. estat sci
```

after `hdidregress` and `xthdidregress` to obtain simultaneous confidence intervals.

Different types of data and specification

We allow for DID estimation of ATETs with two types of data: repeated cross-sections and panel data. The type of estimator to use depends on the assumptions you would like to make about the model and what type of data you have. Below, we walk you through the different types of models you may want to fit and how to fit them with `didregress`, `hdidregress`, `xtdidregress`, and `xthdidregress`.

Suppose you have a repeated cross-section of individuals i over a period of time t . The treatment is implemented at the state level, s , and it is denoted by D_{st} , where $D_{st} = 1$ for all observations that are subject to the treatment in state s at time t . The indexing of the treatment makes clear that all individuals in the state at a given time are either treated or untreated. The model to be fit is given by

$$y_{ist} = \gamma_s + \gamma_t + \mathbf{z}_{ist}\beta + D_{st}\delta + \epsilon_{ist} \quad (3)$$

In the expression above, y_{ist} is the outcome, y in our data; $\mathbf{z}_{ist}$ are individual-level characteristics, given by `z1` and `z2`; and D_{st} is given by the variable `d`. To obtain the ATET, we type

```
. didregress (y z1 z2) (d), group(state) time(year)
```

Within the first set of parentheses, we list the outcome and covariates $\mathbf{z}_{ist}$ from (3). In the second set of parentheses, we specify the treatment variable. Group and time effects are included in the model through the `group()` and `time()` options, respectively. Also, by default, cluster-robust standard errors are computed at the state level. The command above is equivalent to typing

```
. regress y z1 z2 i.year i.state d, vce(cluster state)
```

The model we fit above assumes the treatment effect is homogeneous. Yet we may want to allow the treatment effect to vary over treatment-time cohort and over time. If this is the case, we may type

```
. hdidregress twfe (y) (d), group(state) time(year)
```

or, instead of `twfe`, we could specify `ra`, `ipw`, or `aipw`. All of these estimators will provide multiple cohort-time treatment effects instead of one ATET parameter. Notice that we excluded `z1` and `z2` in the specification above. Depending on the estimator we select, we could include them in either of the sets of parentheses. If we include them only in the first parentheses, we are modeling the outcome. If we include them in the second parentheses, we are modeling the treatment. If we include regressors in both parentheses, we are modeling the treatment and the outcome. This is equivalent to what we specify when we use the estimators discussed in [\[CAUSAL\] teffects ra](#), [\[CAUSAL\] teffects ipw](#), and [\[CAUSAL\] teffects aipw](#).

`didregress` also allows us to specify a DDD model for situations in which we would like to augment the DID framework to include another control group. For instance, starting from (3), let's assume the treatment occurs at the state level but also varies for people older than 65 versus people who are younger, defined by a binary variable g . The DDD model is now given by

$$y_{isgt} = \gamma_s + \gamma_g + \gamma_t + \gamma_s \gamma_t + \gamma_g \gamma_t + \gamma_s \gamma_g + \mathbf{z}_{isgt} \beta + D_{sgt} \delta + \epsilon_{isgt}$$

To fit this model, we would type

```
. didregress (y z1 z2) (d), group(state g) time(year)
```

Although the model has a large set of interactions and looks much more complex than the DID model, the only difference in what we type is the addition of g to the set of group variables. The cluster-robust standard errors are computed at the highest level of clustering, in this case, *state*.

In some cases, data do not include a time component but rather multiple grouping variables across which differences may be taken. You could fit DDD models for three groups or DID for two groups. For DDD with *state*, *g1*, and *g2*, you could type

```
. didregress (y z1 z2) (d), group(state g1 g2)
```

Or you could fit a DID by typing

```
. didregress (y z1 z2) (d), group(g1 g2)
```

With these last two specifications, diagnostics that rely on time variables are not available.

With panel data, the basic model is given by

$$y_{ist} = \gamma_i + \gamma_t + \mathbf{z}_{ist} \beta + D_{st} \delta + \epsilon_{ist} \quad (4)$$

Again, i denotes the individual (panel), t is the time period, and s is a group-level index. Individuals are assumed to be nested within the group, and treatment occurs at the group and time levels.

The specification is analogous to the one in (3), except that instead of γ_s we have γ_i . Suppose that our data now include a panel identifier variable, *id*, that corresponds to i in (4). To fit the model above, we would type

```
. xtset id
. xtddidregress (y z1 z2) (d), group(state) time(year)
```

In this model, the group variable is different from the *xtset* identifying variable. The group variable defines the level of clustering, whereas the *xtset* variable defines the panel identifier. In other words, we are fitting a fixed-effects model with individual fixed effects, not one with state fixed effects. Of course, the group and *xtset* variables could be the same, but they do not need to be.

If we want to assume heterogeneous treatment, we must *xtset* our data with respect to the panel variable and with respect to time. The heterogeneous estimates rely on first differences of the outcome variable. For the RA estimator, you would type

```
. xtset id time
. xthdidregress ra (y z1 z2) (d), group(state)
```

Again, the panel variable and the group variable need not be the same, but the group and panel variable need to be related.

Specifying groups and time as binary indicators

The cross-sectional and panel-data models discussed above for homogeneous treatment are referred to as generalized DID models by [Wing, Simon, and Bello-Gomez \(2018\)](#). They are a generalization of the two-period, two-group specification that is usually discussed in the literature. This is also the way [Angrist and Pischke \(2009\)](#) write the model.

It is not uncommon, however, to write the DID model in terms of binary indicators. Instead of having group effects and time effects in the model for all groups and times in the data, you could instead include only a posttreatment effect and a treatment-group effect. To do this, you would create and include an indicator in your model that takes the value of 1 if the observation belongs to a treated group and 0 otherwise as well as an indicator that is 1 if the period considered occurs after the treatment. You can specify this model by using the `nogteffects` option and including your own indicators.

Say, for example, you create a time-constant indicator, `gtreated`, that is 1 if a group is treated and 0 otherwise, as well as an indicator, `post`, that is 1 after the policy was implemented and 0 before. You would type

```
. didregress (y z1 z2 i.post i.gtreated) (d), group(state) time(year) nogteffects
```

The syntax, in which we include indicators and use `nogteffects`, is equivalent for `xtdidregress`.

The model you specify in this case is given by

$$y_{ist} = 1 \{s = \text{treated}\} + 1 \{t = \text{post}\} + \mathbf{z}_{ist}\beta + D_{st}\delta + \epsilon_{ist}$$

where $1 \{s = \text{treated}\}$ is an indicator function that is 1 if group s receives the treatment and 0 otherwise. Similarly, $1 \{t = \text{post}\}$ indicates that we are in the period for which the treatment is active. In the expression above, we still specify the `group()` and `time()` options to obtain the correct standard errors, to validate the assumptions of the DID specification, and to obtain group- and time-level statistics.

Excluding group and time effects

`didregress` and `xtdidregress` allow you to forgo the group and time effects that are generated by default. In this case, you would have to add your own group and time effects or omit them entirely by using the `nogteffects` option, which excludes group and time effects from your specification. For DDD models, you may also use the `nointeract` option, which will exclude group and time interactions from your model. Combining both options excludes group and time effects altogether. You may type

```
. didregress (y z) (d), group(s g) time(t) nogteffects nointeract
```

You would then fit

$$y_{isgt} = \beta_0 + \mathbf{z}_{isgt}\beta + D_{sgt}\delta + \epsilon_{ist}$$

You could still include group and time interactions by adding them in the first set of parentheses.

Exploring treatment-effect heterogeneity

The heterogeneous treatment-effect estimators assume that the treatment effect changes over cohorts and over time. Yet we may want to see how the treatment effects vary only in one of those dimensions. For instance, we may want to look at the average of the treatment effects for the 1995 cohort $ATET(1995, t)$ within all time periods. We could also average within cohorts at a given point in time to get a treatment effect for a particular year. Or even more, we might average over cohort and over time to get back one single ATET. After estimation, we can do this using the `estat aggregation` command; see [CAUSAL] `hdidregress postestimation`.

For the `twfe` estimator, we can additionally decide at the estimation stage what level of heterogeneity we want to impose to our model. We can decide whether we want to allow for time heterogeneity or cohort-level heterogeneity using the `hettype()` option. For instance, if we wanted to disregard time heterogeneity but model cohort heterogeneity, we would type

```
. hdidregress ..., ... hettype(cohort)
```

DID with switching treatments for panel data

The concept of an ATET becomes less useful and harder to define when treatments can be nonbinary and nonabsorbing (switching). However, we can still use the principle of difference in differences to learn something about the effect of treatment variables on outcomes of interest in these cases.

`xtswitchdid` applies the panel-data DID method of [de Chaisemartin and D’Haultfœuille \(2026\)](#) to estimate event-study effects. Consider a case where a treated group experiences the treatment path $(0, 1, 1, 2, 2)$ over five time periods, meaning that the group starts at treatment value 0, then moves to treatment value 1 in the second period, and finally to treatment value 2 in the fourth period. It is natural to consider the effect, $\delta_{g,\ell}$, of group g having experienced this treatment path relative to the counterfactual treatment path $(0, 0, 0, 0, 0)$, in which the group stayed at its initial treatment value. For group g , whose first treatment switch away from initial treatment value was ℓ periods ago, we write

$$\delta_{g,\ell} = E(Y_{g,t_g+\ell} - Y_{g,t_g+\ell}(0, 0, 0, 0, 0) | \mathbf{D})$$

where t_g is the first period in which g had a treatment different from its initial treatment and $\mathbf{D}$ is a stacked vector of observed treatment paths across all groups.

However, we are interested not only in this group but also in any group that has experienced a switch away from its initial treatment group. Additionally, we restrict attention to groups for which treatment either rises above or falls below the initial treatment level but not both. [de Chaisemartin and D’Haultfœuille \(2026\)](#) define the event-study parameter δ_ℓ as an aggregation of parameters $\delta_{g,\ell}$ across groups that experience a treatment shock and for which there exists at least one suitable control group. In this case, a suitable control group for g at exposure ℓ is one that shares the same initial treatment value and has not switched away from that treatment as of period $t_g + \ell$.

We do not observe the counterfactual, untreated outcome for group g , but we can use a DID method to estimate $\delta_{g,\ell}$. The difference $Y_{g,t_g+\ell} - Y_{g,t_g}$ is itself differenced with the average of the differences $Y_{g',t_g+\ell} - Y_{g',t_g}$ across relevant control groups g' . This difference is then, with appropriate weights, aggregated across all groups g with available controls at exposure ℓ to give an estimate $\hat{\delta}_\ell$. We can interpret our estimate as the average effect of being exposed to a treatment weakly stronger than the initial treatment level after ℓ periods.

`xtswitchdid`, by default, returns analytic standard errors clustered at the group level. Bootstrap standard errors are also available.

We may be interested in focusing only on switching groups with a specific treatment path. Using the `path()` option, we could type

```
. xtswitchdid ..., ... path(0 1 1 2 2)
```

This would give us a set of event-study effects that can be readily interpreted as the effect of experiencing the treatment path (0, 1, 1, 2, 2) up to a given exposure, relative to remaining at treatment level 0.

As with other DID estimators, `xtswitchdid` relies on a parallel trends assumption, as well as an assumption of no anticipation. These assumptions can be tested using placebo effects by typing

```
. estat ptrends
```

after estimation using `xtswitchdid`.

Conclusion

`didregress` and `xtdidregress` compute the ATET using DID and DDD for panel data and repeated cross-sections. Additionally, `hdidregress` and `thdidregress` estimate ATETs that vary over time and over treatment cohorts, while `xtswitchdid` estimates event-study effects for treatments that can be nonbinary and nonabsorbing. All four commands offer standard error computations that address some of the issues that researchers face, such as a small number of groups. After you fit models with these commands, diagnostics and tests to validate the assumptions and internal validity of the DID and DDD results are available.

References

- Abadie, A., and M. D. Cattaneo. 2018. Econometric methods for program evaluation. *Annual Review of Economics* 10: 465–503. <https://doi.org/10.1146/annurev-economics-080217-053402>.
- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- . 2015. *Mastering Metrics: The Path from Cause to Effect*. Princeton, NJ: Princeton University Press.
- Bell, R. M., and D. F. McCaffrey. 2002. Bias reduction in standard errors for linear regression with multi-stage samples. *Survey Methodology* 28: 169–181.
- Bertrand, M., E. Duflo, and S. Mullainathan. 2004. How much should we trust difference-in-differences estimates? *Quarterly Journal of Economics* 119: 249–275. <https://doi.org/10.1162/003355304772839588>.
- Bester, C. A., T. G. Conley, and C. B. Hansen. 2011. Inference with dependent data using cluster covariance estimators. *Journal of Econometrics* 165: 137–151. <https://doi.org/10.1016/j.jeconom.2011.01.007>.
- Blundell, R. W., and M. C. Dias. 2009. Alternative approaches to evaluation in empirical microeconomics. *Journal of Human Resources* 44: 565–640. <https://doi.org/10.1007/s10258-002-0010-3>.
- Borusyak, K., X. Jaravel, and J. Spiess. 2021. Revisiting event study designs: Robust and efficient estimation. arXiv:2108.12419 [econ.EM], <https://doi.org/10.48550/arXiv.2108.12419>.
- Callaway, B., and P. H. C. Sant'Anna. 2021. Difference-in-differences with multiple time periods. *Journal of Econometrics* 225: 200–230. <https://doi.org/10.1016/j.jeconom.2020.12.001>.
- Cameron, A. C., J. B. Gelbach, and D. L. Miller. 2008. Bootstrap-based improvements for inference with clustered errors. *Review of Economics and Statistics* 90: 414–427. <https://doi.org/10.1162/rest.90.3.414>.
- Cameron, A. C., and D. L. Miller. 2015. A practitioner's guide to cluster-robust inference. *Journal of Human Resources* 50: 317–372. <https://doi.org/10.3368/jhr.50.2.317>.
- Clarke, D., D. Pailanir, S. Athey, and G. W. Imbens. 2024. On synthetic difference-in-differences and related estimation methods in Stata. *Stata Journal* 24: 557–598.

- de Chaisemartin, C., and X. D’Haultfœuille. 2020. Two-way fixed effects estimators with heterogeneous treatment effects. *American Economic Review* 110: 2964–2996. <https://doi.org/10.1257/aer.20181169>.
- . 2023. Two-way fixed effects and differences-in-differences with heterogeneous treatment effects: A survey. *Econometrics Journal* 26: C1–C30. <https://doi.org/10.1093/ectj/utac017>.
- . 2026. Difference-in-differences estimators of intertemporal treatment effects. *Review of Economics and Statistics* 108: 863–880. https://doi.org/10.1162/rest_a_01414.
- Donald, S. G., and K. Lang. 2007. Inference with difference-in-differences and other panel data. *Review of Economics and Statistics* 89: 221–233. <https://doi.org/10.1162/rest.89.2.221>.
- Goodman-Bacon, A. 2021. Difference-in-differences with variation in treatment timing. *Journal of Econometrics* 225: 254–277. <https://doi.org/10.1016/j.jeconom.2021.03.014>.
- Imbens, G. W., and M. Kolesár. 2016. Robust standard errors in small samples: Some practical advice. *Review of Economics and Statistics* 98: 701–712. https://doi.org/10.1162/REST_a_00552.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Kaspereit, T. 2021. Event studies with daily stock returns in Stata: Which command to use? *Stata Journal* 21: 462–497.
- Lechner, M. 2011. The estimation of causal effects by difference-in-difference methods. *Foundations and Trends in Econometrics* 4: 165–224. <https://doi.org/10.1561/08000000014>.
- MacKinnon, J. G. 2019. How cluster-robust inference is changing applied econometrics. *Canadian Journal of Economics* 52: 851–881. <https://doi.org/10.1111/caje.12388>.
- MacKinnon, J. G., and M. D. Webb. 2018. The wild bootstrap for few (treated) clusters. *Econometrics Journal* 21: 114–135. <https://doi.org/10.1111/ectj.12107>.
- Roodman, D., J. G. MacKinnon, M. Ø. Nielsen, and M. D. Webb. 2019. Fast and wild: Bootstrap inference in Stata using `boottest`. *Stata Journal* 19: 4–60.
- Roth, J., P. H. C. Sant’Anna, A. Bilinski, and J. Poe. 2022. What’s trending in difference-in-differences? A synthesis of the recent econometrics literature. arXiv:2201.01194 [econ.EM], <https://doi.org/10.48550/arXiv.2201.01194>.
- Snow, J. 1849. *On the Mode of Communication of Cholera*. London: Churchill.
- . 1855. *On the Mode of Communication of Cholera*. 2nd ed. London: Churchill.
- Sun, L., and S. Abraham. 2021. Estimating dynamic treatment effects in event studies with heterogeneous treatment effects. *Journal of Econometrics* 225: 175–199. <https://doi.org/10.1016/j.jeconom.2020.09.006>.
- Wing, C., K. Simon, and R. A. Bello-Gomez. 2018. Designing difference in difference studies: Best practices for public health policy research. *Annual Review of Public Health* 39: 453–469. <https://doi.org/10.1146/annurev-publhealth-040617-013507>.
- Wooldridge, J. M. 2021. Two-way fixed effects, the two-way Mundlak regression, and difference-in-differences estimators. Working paper, Department of Economics, Michigan State University, East Lansing, MI. <https://doi.org/10.2139/ssrn.3906345>.

Also see

[CAUSAL] [teffects](#) — Treatment-effects estimation for observational data

[CAUSAL] [teffects intro](#) — Introduction to treatment effects for observational data

[CAUSAL] [teffects intro advanced](#) — Advanced introduction to treatment effects for observational data

⁺This command includes features that are part of [StataNow](#).

Description	Quick start	Menu	Syntax
Options	Remarks and examples	Stored results	Methods and formulas
Acknowledgment	References	Also see	

Description

`didregress` estimates the average treatment effect on the treated (ATET) from observational data by difference-in-differences (DID) or difference-in-difference-in-differences (DDD). The ATET of a binary or continuous treatment on a continuous outcome is estimated by fitting a linear model with time and group fixed effects. The DID and DDD estimation performed by `didregress` can be applied to data comprising repeated cross-sections in which different groups of individuals are observed at each time period.

`xtdidregress` estimates the ATET from observational data by DID or DDD for panel data. The ATET of a binary or continuous treatment on a continuous outcome is estimated by fitting a linear model with time and panel fixed effects.

Quick start

DID estimate of the ATET of `treat1` on outcome `y1` modeled using covariates `x1` and `x2`, and `grpvar1` and `tvar` fixed effects, with the treatment occurring at the `grpvar1` and `tvar` levels

```
didregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar)
```

Same as above, but compute wild cluster–bootstrap *p*-values and confidence intervals with `grpvar1` as the clustering variable

```
didregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar) ///
wildbootstrap
```

Aggregate data at the `grpvar1` and `tvar` levels to estimate the ATET

```
didregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar) ///
aggregate(standard)
```

Same as above, but use the [Donald and Lang \(2007\)](#) method to compute the ATET and standard errors

```
didregress (y1 x1 x2) (treat1), group(grpvar1) time(tvar) ///
aggregate(dlang)
```

DDD estimate of the ATET of `treat2` on outcome `y2` modeled using covariates `x1` and `x2` and fixed effects defined by two-way interactions for `grpvar1`, `grpvar2`, and `tvar`, with the treatment occurring at the `grpvar1`, `grpvar2`, and `tvar` levels

```
didregress (y2 x1 x2) (treat2), group(grpvar1 grpvar2) time(tvar)
```

DID estimate of ATET of `treat3` on outcome `y3` using `xtset` data; `y3` modeled using covariates `x1` and `x2`, and individual (panel) and `tvar` fixed effects, with the treatment occurring at the `grpvar1` and `tvar` levels

```
xtdidregress (y3 x1 x2) (treat3), group(grpvar1) time(tvar)
```

Menu

didregress

Statistics > Causal inference/treatment effects > Continuous outcomes > Difference in differences (DID) > DID

xtdidregress

Statistics > Causal inference/treatment effects > Continuous outcomes > Difference in differences (DID) > Panel-data DID (FE)

Syntax

DID for repeated cross-sectional data

```
didregress (ovar omvarlist) (tvar [ , continuous]) [if] [in] [weight],
    group(groupvars) [time(timevar) options]
```

DID for panel data

```
xtdidregress (ovar omvarlist) (tvar [ , continuous]) [if] [in] [weight],
    group(groupvars) [time(timevar) options]
```

ovar is the outcome of interest.

omvarlist specifies the covariates in the outcome model and may contain factor variables; see [\[U\] 11.4.3 Factor variables](#).

tvar must be a binary variable indicating observations subject to treatment or a continuous variable measuring treatment intensity.

groupvars are categorical variables that indicate the group level at which the treatment occurs. At least one group variable must be specified. If *timevar* is specified, at most two group variables may be specified. If *timevar* is not specified, at most three group variables may be specified.

timevar is a time variable. It must be specified if *groupvar* has only one variable.

<i>options</i>	Description
Model	
* <code>group(<i>groupvars</i>)</code>	specify group variables
‡ <code>time(<i>timevar</i>)</code>	specify time variable
<code>nointeract</code>	exclude <code>group()</code> and <code>time()</code> interactions
<code>noteffects</code>	do not include group and time effects in the model
<code>aggregate(<i>aggmethod</i>)</code>	aggregate to the levels defined by interacting <i>groupvars</i> and <i>timevar</i>
<code>wildbootstrap[(<i>wildopts</i>)]</code>	compute confidence intervals and <i>p</i> -values with the wild bootstrap
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>cluster</code> <i>clustvar</i> , <code>robust</code> , <code>hc2</code> , or <code>bootstrap</code>
+ <code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may also be <code>hc2</code> [<i>clustvar</i>] or <code>hc3</code> [<i>clustvar</i>]
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
<code>coeflegend</code>	display legend instead of statistics

⁺This feature is part of [StataNow](#).

*`group(groupvars)` is required.

‡`time(timevar)` is required when only one group is specified.

For `xtdidregress`, a panel variable must be specified using `xtset`; see [\[XT\] xtset](#).

`by`, `collect`, and `statsby` are allowed; see [\[U\] 11.1.10 Prefix commands](#).

`fweights`, `awweights`, and `pweights` are allowed; see [\[U\] 11.1.6 weight](#).

`pweights` are not allowed with `vce(hc3)`.

`coeflegend` does not appear in the dialog box.

See [\[U\] 20 Estimation and postestimation commands](#) for more capabilities of estimation commands.

<i>aggmethod</i>	Description
<code>standard</code>	aggregate data and fit model
<code>dlang[, <i>dlopt</i>]</code>	aggregate data and fit model using Donald and Lang method

<i>wildopts</i>	Description
<code>errorweight(<i>edtype</i>)</code>	specify the error weight type <i>edtype</i> ; default is <code>errorweight(rademacher)</code>
<code>reps(#)</code>	perform # wild bootstrap replications; default is <code>reps(1000)</code>
<code>rseed(#)</code>	set random-number seed to #
<code>blocksize(#)</code>	perform wild bootstrap in blocks of # replications

Options

Model

`group(groupvars)` specifies group variables. It indicates the group level at which the treatment occurs. *groupvars* may be, for example, states, counties, or hospitals. *groupvars* define the group levels for which group effects are included in the model used to perform DID estimation and for which group interactions are included in the model used to perform DDD estimation. `group()` also defines the level clustering for the default cluster-robust standard errors. `group()` is required.

You may specify at most two group variables if you also specify a time variable in the `time()` option or at most three group variables if no time variable is specified.

`time(timevar)` specifies the time variable. You may specify `time()` when one or two group variables are specified in the `group()` option. `time()` is required when only one group variable is specified in the `group()` option.

`nointeract` excludes *groupvars* and *timevar* interactions from being included in the model. By default, `didregress` and `xtdidregress` include group and time interactions for your specification if there is more than one group variable.

`nogteffects` specifies to not include group and time effects. By default, `didregress` adds group and time dummies to the regression specification. By default, `xtdidregress` adds time dummies to the fixed-effects specification.

`aggregate(aggmethod)` fits the model by aggregating data at the *groupvars* and *timevar* levels. *aggmethod* may be either `standard` or `dlang[, dlopt]`.

`standard` specifies that aggregation is performed using the standard aggregation method. In this case, a regression model is fit of the original outcome on covariates that vary within levels of the *groupvars* and *timevar* interaction. The estimates of the group-time level effects from this regression are used to construct a new dependent variable. These effects along with the remaining covariates are then aggregated to the level of the of the *groupvars* and *timevar* interaction. The final results are obtained by regressing the estimated group-time level effects on the remaining covariates with this aggregated dataset and estimating group-level cluster-robust standard errors.

`dlang[, dlopt]` aggregates data using methods proposed by [Donald and Lang \(2007\)](#). *dlopt* may be either `constant` or `varying`.

`constant` requests that standard errors be estimated using the standard ordinary least-squares method, as suggested by [Donald and Lang \(2007\)](#). With this method, as with the `standard` aggregation method, a single regression model is fit in the first step, so the coefficient estimates are constant across levels of the *groupvars* and *timevar* interaction.

`varying` specifies that the aggregation method allow for varying coefficients on the covariates. Specifically, in this case, separate regression models of the original outcome on covariates that vary within levels of *groupvars* and *timevar* are fit for each level of the *groupvars* and *timevar* interaction. Thus, this method allows the coefficients on these variables to vary. The constant from each of these regressions forms the new dependent variable. The final regression and standard error computations are equivalent to those used by the `constant` method.

`wildbootstrap[ (wildopts) ]` computes confidence intervals and p -values with the wild bootstrap. The wild bootstrap is constructed imposing the null hypothesis that the ATET is 0; that is, it is a restricted wild bootstrap. Confidence intervals are computed separately from the p -values. The bounds of the confidence interval are computed using a bisection optimization algorithm described in [Methods and formulas](#). *wildopts* are `errorweight(edtype)`, `reps(#)`, `rseed(#)`, and `blocksize(#)`.

`errorweight(edtype)` defines the error weight used to draw residuals from the wild bootstrap. *edtype* is one of `rademacher` (the default), `mammen`, `webb`, `normal`, or `gamma`.

`rademacher` multiplies the residuals at each bootstrap replication with a randomly generated variable that takes the value of 1 with probability 0.5 and the value of -1 with probability 0.5. `errorweight(rademacher)` is the default.

`mammen` multiplies the residuals at each bootstrap replication with a randomly generated variable that takes the value of $1 - \phi$ with probability $\phi/\sqrt{5}$ and ϕ otherwise, where $\phi = (1 + \sqrt{5})/2$.

`webb` multiplies the residuals at each bootstrap replication with a randomly generated variable that takes the values $-\sqrt{3}/2$, $-\sqrt{2}/2$, $-\sqrt{1}/2$, $\sqrt{1}/2$, $\sqrt{2}/2$, and $\sqrt{3}/2$, each with probability $1/6$.

`normal` multiplies the residuals at each bootstrap replication with a randomly generated normal distribution variable with the first four moments given by 0, 1, 0, and 3.

`gamma` multiplies the residuals at each bootstrap replication with a randomly generated gamma distribution variable with shape parameter 4 and scale parameter $1/2$.

`reps(#)` performs # wild bootstrap replications. The default is `reps(1000)`.

`rseed(#)` sets the random-number seed to #.

`blocksize(#)` specifies that the wild bootstrap be performed in blocks, with # replications per block. The wild bootstrap computation requires a matrix with dimensions $(\text{\# groups}) \times (\text{\# replications})$. If this is too large, you can reduce the matrix to $(\text{\# groups}) \times (\text{\# block size})$ and loop $(\text{\# replications})/(\text{\# block size})$ times. When the same random seed is set, using a different block size does not change the numerical results; it only modifies the computation method. The block size must be less than or equal to the number of bootstrap replications.

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that allow for intragroup correlation (`cluster clustvar`), that are robust to some kinds of misspecification (`robust`), that are bias-corrected cluster-robust using the degrees-of-freedom adjustment proposed by [Bell and McCaffrey \(2002\)](#) (`hc2` and `hc3`), and that use bootstrap sampling done at the group level (`bootstrap`); see [\[R\] vce_option](#).

`vce(cluster clustvar)` specifies that standard errors allow for intragroup correlation within the groups defined by *clustvar*, relaxing the usual requirement that the observations be independent. `vce(cluster groupvar1)`, where *groupvar1* is the first variable specified in the `group(groupvars)` option, is the default. See [Methods and formulas](#) in [\[R\] regress](#) for details on how cluster-robust VCEs are calculated. Also see [Methods and formulas](#) in [\[R\] areg](#) and [Methods and formulas](#) below for details on how degrees of freedom are calculated.

`vce(robust)` with `didregress` specifies that standard errors allow for heteroskedasticity in the variance of the observations. Intragroup correlations, however, are not allowed. See [Methods and formulas](#) in [\[R\] regress](#) for details on how heteroskedasticity-robust VCEs are calculated.

`vce(robust)` with `xtdidregress` is equivalent to specifying `vce(cluster groupvar1)`, where `groupvar1` is the first variable specified in the `group(groupvars)` option.

`vce(hc2 [clustvar])` and `vce(hc3 [clustvar][, hansen])` specify bias-corrected cluster-robust standard errors with the degrees-of-freedom adjustment proposed by [Bell and McCaffrey \(2002\)](#), with the additional scale adjustment of [Hansen \(2025b\)](#) if suboption `hansen` is specified with `vce(hc3)`. If `clustvar` is not specified, the standard errors are clustered using the first variable specified in the `group(groupvars)` option. As with `vce(hc2)` and `vce(hc3)` with the `regress` command, the residuals are rescaled by the projection matrix to improve the small-sample properties of the variance estimates. For more details, see [Methods and formulas](#) below and [Methods and formulas](#) in [\[R\] regress](#). Specifying the optional `clustvar` in `vce(hc2)` and the `vce(hc3 [clustvar][, hansen])` option are part of StataNow.

Reporting

`level(#)`; see [\[R\] Estimation options](#).

`aequations` specifies that the results for the outcome-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`display_options:` `nocl`, `novalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no stretch`; see [\[R\] Estimation options](#).

The following option is available with `didregress` and `xtdidregress` but is not shown in the dialog box:

`coeflegend`; see [\[R\] Estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

- [Introduction](#)
- [DID estimation](#)
- [Graphical diagnostics and tests](#)
- [Specifying a 2-by-2 DID](#)
- [Standard error considerations](#)
- [Default cluster-robust standard errors](#)

Introduction

DID is one of the most venerable causal inference methods used by researchers. DID estimates the average treatment effect on the treated group (ATET).

To obtain the ATET using DID, one must compute the difference of the mean outcome for the treatment and the control groups before and after the treatment. This eliminates time-invariant unobservable group characteristics that confound the ATET; however, this is not enough to identify an effect. There may be time-varying unobservable confounders with an effect on the treatment group even after we control for time-invariant unobservable group characteristics. DID eliminates time-varying confounders by including a control group that is subject to the same time-varying confounders as the treatment group.

The ATET is then consistently estimated, differencing the mean outcome for the treatment and control groups over time to eliminate time-invariant unobservable characteristics and also differencing the mean outcome of these groups to eliminate time-varying unobservable effects common to both groups. These two differences give the DID method its name and highlight its intuitive appeal. More appealing is the fact that you can get the effect of interest, the ATET, from one parameter in a linear regression.

Below, we illustrate how to use `didregress` and `xtdidregress`. For more information about the methods used below, see [\[CAUSAL\] DID intro](#). For general discussions of the DID methodology, see [Angrist and Pischke \(2009, 2015\)](#), [Blundell and Dias \(2009\)](#), [Imbens and Wooldridge \(2009\)](#), [Lechner \(2011\)](#), [Abadie and Cattaneo \(2018\)](#), and [Wing, Simon, and Bello-Gomez \(2018\)](#) and the references therein.

DID estimation

▷ Example 1: Fitting a DID model

A health provider is interested in studying the effect of a new hospital admissions procedure on the satisfaction of patients. The provider has monthly data on patients from January to July. The new admissions procedure was implemented in April by hospitals that were under new management. Of the 46 hospitals in the study, 18 implemented the new procedure.

The health provider will use a DID regression to analyze the effect of the new admissions procedure on the hospitals that participated in the program. The outcome of interest is patient satisfaction, `satis`, which is recorded as an average of the responses to a set of four questions asked to patients. `satis` may take values between 0 and 10, where 10 is the greatest possible level of satisfaction and 0 is utter disappointment. The procedure variable marks the treated observations; it is 1 if a surveyed individual was admitted to the hospital using the new procedure after March and 0 otherwise. To get the ATET on the outcome `satis`, we type

```
. didregress (satis) (procedure), group(hospital) time(month)
```

The first set of parentheses is used to specify the outcome of interest followed by the covariates in the model. In this case, there are no covariates, just the outcome, `satis`. The second set of parentheses is used to specify the binary variable that indicates the treated observations, `procedure`. The `group()` and `time()` options are used to construct group and time fixed effects that are included in the model. The variable specified in `group()` is also important because it defines the level of clustering for the default cluster-robust standard errors; in this case, we cluster at the hospital level. The results from this command are as follows:

```
. use https://www.stata-press.com/data/r19/hospdd
(Artificial hospital admission procedure data)
. didregress (satis) (procedure), group(hospital) time(month)
```

Treatment and time information

Time variable: month

Control: procedure = 0

Treatment: procedure = 1

	Control	Treatment
Group hospital	28	18
Time		
Minimum	1	4
Maximum	1	4

Difference-in-differences regression

Number of obs = 7,368

Data type: Repeated cross-sectional

(Std. err. adjusted for 46 clusters in hospital)

satis	Robust		t	P> t	[95% conf. interval]	
	Coefficient	std. err.				
ATET procedure (New vs Old)	.8479879	.0321121	26.41	0.000	.7833108	.912665

Note: ATET estimate adjusted for group effects and time effects.

The first table gives information about the treated and control groups and about treatment timing. The first segment with the title Group tells us the number of treated and control hospitals: 28 hospitals were using the old procedure and 18 hospitals were using the new one. The second segment of the table gives information about the first time we observe hospitals in the control group and the first time we observe the treatment (the new admission procedure) for hospitals in the treatment group. In this example, all hospitals that adopted the new procedure did so in April, time period 4. If some hospitals had adopted the policy later, the minimum and maximum time of first treatment would differ.

The ATET is 0.85, almost a 1-point increase in satisfaction relative to the case where none of the treated hospitals enacted the new procedure. In other words, if the hospitals that implemented the new admission procedure had not done so, their satisfaction ratings would be lower by almost one point on average.

We now explore whether the trajectories of `satis` are parallel for the control and treatment groups prior to the date when the new procedure was implemented. We are checking what is known as the parallel-trends or common-trends assumption, an important assumption of the DID model, as discussed in [CAUSAL] DID intro. A visual diagnostic of this assumption can be obtained by plotting the means of the outcome over time for both groups or by visualizing the results of the linear-trends model. We can perform both of these diagnostic checks by using `estat trendplots`. To obtain figure 1 below, we type

```
. estat trendplots
```

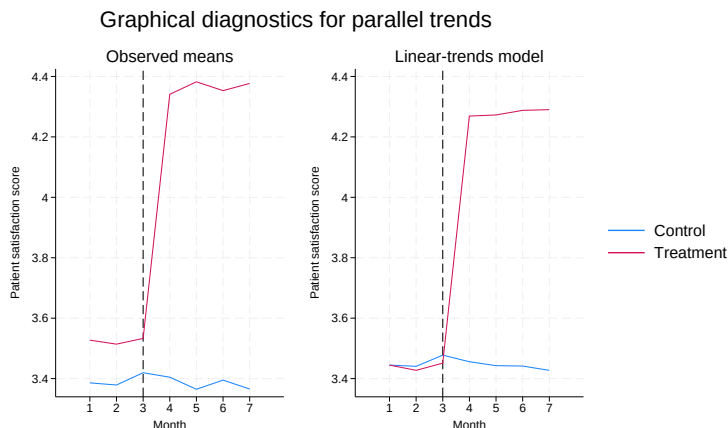


Figure 1.

The graph seems to indicate that the parallel-trends assumption is satisfied. Prior to the policy implementation, treated and control hospitals followed a parallel path.

We could also perform a test to see if the trajectories are parallel by augmenting our original model to include variables representing time trends before and after the treatment for both groups of hospitals. The linear-trends model estimates a coefficient for the differences in linear trends prior to treatment, and if that coefficient is 0, the linear pretreatment trends are parallel. Otherwise, identification of the ATET may become questionable.

We can perform this test by using `estat ptrends`:

```
. estat ptrends
Parallel-trends test (pretreatment time period)
H0: Linear trends are parallel
F(1, 45) = 0.55
Prob > F = 0.4615
```

We do not have evidence to reject the null hypothesis of parallel trends in this case. Both the test and the graphical analysis support the parallel-trends assumption and, therefore, our ATET estimate.

► Example 2: Fitting a DDD model

The results in [example 1](#) could come into question if they could be the consequence of other unobserved variables rather than the consequence of the new hospital admissions procedure. The health provider administrators believe that responses to the survey are related to the frequency of individuals' hospital visits. The patients may have unobserved characteristics that affect both how frequently they visit the hospital and how they feel about the admissions procedure. In other words, there might be unobserved characteristics that confound the effect of the new hospital admissions procedure. The administrators decide to obtain the ATET by using a DDD model. They want to estimate the average treatment effect on patients who visit the hospital with high or very high frequencies.

To do this, we will first create a new variable `hightrt` to be our new treatment identifier. Observations are now marked as treated (`hightrt = 1`) if hospital visit frequency by an individual is high or very high (`frequency = 3` or `4`) and if the hospital implemented the new admissions procedure in April.

```
. generate hightrt = procedure==1 & (frequency==3 | frequency==4)
. label define trt 0 "Untreated" 1 "Treated"
. label values hightrt trt
```

The DDD model will incorporate both hospital and frequency of usage effects as well as their interaction with time effects. To fit the model, we incorporate a new group variable, `frequency`:

```
. didregress (satis) (hightrt), group(hospital frequency) time(month)
(output omitted)
```

Treatment and time information

Time variable: month

Control: hightrt = 0

Treatment: hightrt = 1

	Control	Treatment
Group		
hospital	28	18
frequency	2	2
Time		
Minimum	1	4
Maximum	1	4

Triple-differences regression

Number of obs = 7,368

Data type: Repeated cross-sectional

(Std. err. adjusted for 46 clusters in hospital)

satis	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
ATET						
hightrt						
(Treated						
vs						
Untreated)	.764154	.0402603	18.98	0.000	.6830655	.8452425

Note: ATET estimate adjusted for group effects, time effects, and group- and time-effects interactions.

The omitted output after the command corresponds to the factor-variable interactions that include the base categories. This is common when you fit DDD models that by default include group interactions and group and time interactions.

The first table above has information on the second group variable, `frequency`, for which low and medium frequencies are controls and high and very high frequencies are treated. The second table shows that the ATET is now smaller, but the policies still increase satisfaction.

◀

► Example 3: DID for panel-data model

Moser and Voena (2012) look at the effect of compulsory licensing on domestic inventions. Compulsory licensing allows firms in developing countries to produce foreign inventions without the consent of foreign patent owners. Having access to foreign technology may discourage domestic inventions, but it could also enhance local production.

Moser and Voena consider legislation that occurred during World War I called Trading With the Enemy Act (TWEA). By 1919, German-owned patents were systematically licensed to US companies. In Moser and Voena (2012), the treated observations correspond to a subclass in the chemical industry that was granted at least one of the TWEA patents after 1918, as reported by United States Patent and Trademark Office (USPTO). A subclass is a group of firms in an industry that employ similar technologies, as defined by the USPTO.

The outcome of interest is the number of patents granted to inventors from the US in that subclass, `uspatents`. This measures domestic innovation. Moser and Voena also include the number of non-TWEA patents granted to the subclass that were from foreign inventors, `fpatents`. `fpatents` measure innovation in the subclass that is not from US inventors. We observe the same subclasses at each point in time from 1875 to 1939. Thus, we have a panel dataset.

To fit the model, we first `xtset` our data at the subclass level, `classid`:

```
. use https://www.stata-press.com/data/r19/patents
(Excerpt from Moser and Voena (2012))
. xtset classid
Panel variable: classid (balanced)
```

Below, we fit a DID model for the number of patents granted to US inventors in a subclass, controlling for the number of non-TWEA patents granted to foreign inventors. The treatment indicator `gotpatent` is 1 if the subclass received a TWEA patent after 1918 and is 0 otherwise. The model includes year fixed effects and subclass fixed effects.

```
. xtddidregress (uspatents fpatents) (gotpatent), group(classid) time(year)
```

Treatment and time information

Time variable: year

Control: gotpatent = 0

Treatment: gotpatent = 1

	Control	Treatment
Group		
classid	6912	336
Time		
Minimum	1875	1919
Maximum	1875	1919

Difference-in-differences regression

Number of obs = 471,120

Data type: Longitudinal

(Std. err. adjusted for 7,248 clusters in classid)

uspatents	Robust		t	P> t	[95% conf. interval]	
	Coefficient	std. err.				
ATET						
gotpatent						
(Patent						
vs						
None)	.150516	.0356081	4.23	0.000	.0807137	.2203183

Note: ATET estimate adjusted for covariates, panel effects, and time effects.

The ATET is 0.15, which means that in subclasses that were awarded one or more patents, domestic inventors produced an average of 0.15 additional patents after the TWEA compared with the scenario in which no patents are awarded for those subclasses.



Graphical diagnostics and tests

► Example 4: DID diagnostic graphs and tests

As illustrated in [example 1](#), when conducting a DID study, it is common to complement the regression analysis with graphical diagnostics and tests that provide evidence of whether an estimated effect can be given a causal interpretation. As discussed in [\[CAUSAL\] DID intro](#), we would like to observe that the treated and control groups had mean outcomes that evolved similarly to each other over time prior to the treatment. This is usually referred to as a parallel-trends or common-trends assumption. We would also like to ascertain that neither the control nor the treatment group changed their behavior in anticipation of the treatment. This is assessed using a Granger-type test.

Below, we use simulated data to illustrate the diagnostics and tests available after `didregress` and `xtddidregress`. Simulated data helps us know exactly what we should expect and how to interpret it.

Suppose we have a panel dataset with 10 time points, denoted by $t1$, where a treatment has taken place between $t1 = 5$ and $t1 = 6$. We have a set of covariates, $x1$ and $x2$, and an outcome, $y1$. The outcome could be something like patient satisfaction, as in [example 1](#), or the number of patents filed by US inventors, as in [example 3](#).

We fit the model:

```
. use https://www.stata-press.com/data/r19/parallel1t
(Simulated data to test parallel-trends assumption)
. xtset id1
Panel variable: id1 (unbalanced)
. xtdidregress (y1 c.x1##c.x2) (treated1), group(id1) time(t1)
Treatment and time information
Time variable: t1
Control:      treated1 = 0
Treatment:    treated1 = 1
```

	Control	Treatment
Group		
id1	102	98
Time		
Minimum	1	6
Maximum	1	6

Difference-in-differences regression
Data type: Longitudinal

Number of obs = 2,000

(Std. err. adjusted for 200 clusters in id1)

		Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
ATET							
treated1							
(Treated							
vs							
Untreated)		.5069426	.0220218	23.02	0.000	.4635166	.5503686

Note: ATET estimate adjusted for covariates, panel effects, and time effects.

Is this result valid? We can first explore the assumption of parallel trends graphically, comparing the trajectories of the outcome variable for the control and treatment groups prior to the date of treatment. We can check this assumption by plotting the means of the outcome over time for both groups or by visualizing the results of the linear-trends model. We can perform both of these diagnostic checks by using `estat trendplots`.


```
. estat trendplots
```

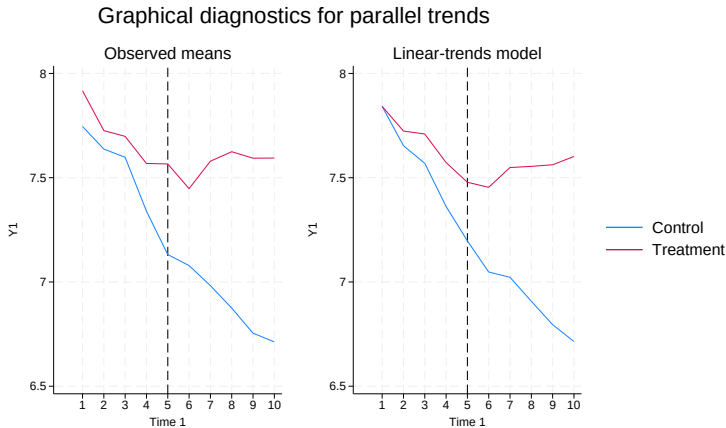


Figure 2.

Looking at the plotted observed means (left side of figure 2), the outcome trajectories in the control and treatment groups prior to the treatment are somewhat different. While we can observe declining trends in both groups, it looks as though the decline is progressing more rapidly in the control group, especially between time points three and five. We can get a clearer picture of this by looking at the results of the linear-trends model on the right side of figure 2. The group-level trajectories are shown with respect to a common reference point, $t_1 = 1$, which makes it easy to discern whether they are parallel. In this case, we can see that they are not. The differences between the treatment and control groups are growing larger over time up to $t_1 = 5$ (posttreatment time periods are not relevant when assessing the parallel-trends assumption). Judging by figure 2 alone, we should be concerned about whether the parallel-trends assumption holds for our effect estimate.

A more formal way to assess whether the pretreatment trajectories are parallel is to perform a test on the linear-trends model coefficient that captures the differences in the trends between treated and controls. If the pretreatment trends are actually linear in both groups, then this coefficient will be 0 because there are no differences in the slopes between the two groups. Thus, by testing this coefficient against 0, we have a test of the null hypothesis that the pretreatment period trajectories are parallel. We can perform this test by using `estat ptrends`:

```
. estat ptrends
Parallel-trends test (pretreatment time period)
H0: Linear trends are parallel
F(1, 199) = 39.97
Prob > F = 0.0000
```

We reject the null hypothesis of the linear trends being parallel.

We can also think of nonparallel as an indication of an anticipatory treatment effect. We saw that the trends were not parallel before the treatment took place, which could indicate a treatment effect even before the treatment is implemented. Thus, another way to state our parallel-trends assumption is that there should be no treatment effect in anticipation of the treatment. To test this assumption, we could fit a Granger-type causality model where we augment our model with dummies that indicate future treatment status for each time period prior to the treatment. A joint test of the coefficients on these dummies against 0 can be used as a test of the null hypothesis that no anticipatory effects have taken place. We can perform this test by using `estat granger`:

```
. estat granger
Granger causality test
H0: No effect in anticipation of treatment
F(4, 199) = 18.17
Prob > F = 0.0000
```

The result suggests that we reject the null hypothesis of no anticipatory effects prior to treatment. Thus, based on the results of both `estat ptrends` and `estat granger`, we conclude that we should be concerned about identification of ATET.

Notice that the parallel-trends F test consumes only 1 numerator degree of freedom, while the Granger causality F test consumes 4. That is because we have five pretreatment time periods and thus four coefficients to test. In this example, both tests would be appropriate, but the parallel-trends test has higher statistical power. However, as we will see below, the less-powered Granger test is more flexible and can be used in situations where the differences between treatment and control groups are nonlinear in pretreatment periods.

◀

➤ Example 5: DID diagnostics and tests with nonlinear mean outcome differences

We now look at a different outcome and policy using the same simulated dataset as in [example 4](#). We fit the same model but with the new variables:

```
. xtset id2
Panel variable: id2 (balanced)
. xtdidregress (y2 c.z1##c.z2) (treated2), group(id2) time(t2)
Treatment and time information
Time variable: t2
Control:      treated2 = 0
Treatment:    treated2 = 1
```

		Control	Treatment
Group	id2	480	520
Time	Minimum	1	6
	Maximum	1	6

Difference-in-differences regression Number of obs = 10,000
Data type: Longitudinal

(Std. err. adjusted for 1,000 clusters in id2)						
y2	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
ATET treated2 (Treated vs Untreated)	.2636651	.0097188	27.13	0.000	.2445936	.2827367

Note: ATET estimate adjusted for covariates, panel effects, and time effects.

Here is the diagnostic plot produced by estat trendplots:

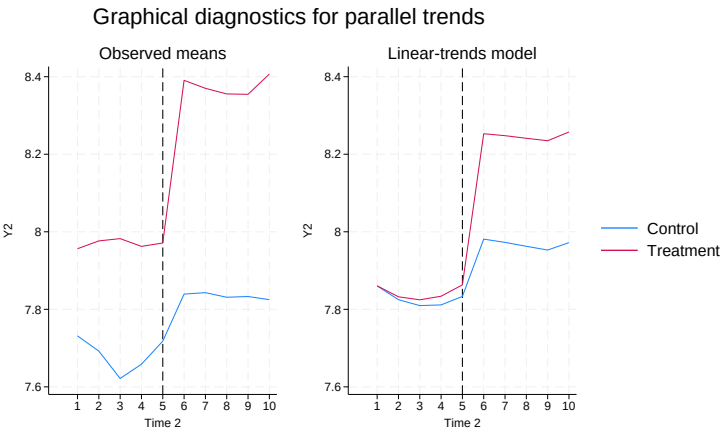


Figure 3.

Inspecting the observed means over the pretreatment time periods, we can see that no considerable change has occurred in the outcome of the treatment group over time. For the control group, on the other hand, we observe a somewhat U-shaped trajectory in advance of the treatment. However, if we look at the plot from the linear-trends model, both trajectories appear somewhat U-shaped, and the trends appear to be almost parallel. Indeed, if we use `estat ptrends`, we fail to reject the null hypothesis of parallel trends:

```
. estat ptrends
Parallel-trends test (pretreatment time period)
H0: Linear trends are parallel
F(1, 999) = 2.13
Prob > F = 0.1446
```

The Granger causality test can handle cases in which the trajectories are nonlinear and for which `estat ptrends` will fail to reject the null hypothesis when it should:

```
. estat granger
Granger causality test
H0: No effect in anticipation of treatment
F(4, 999) = 9.86
Prob > F = 0.0000
```

We correctly reject the null hypothesis of no effect in anticipation of the treatment.



► Example 6: Diagnostics and tests when parallel-trends assumption is satisfied

Finally, we look at a case where the assumption of parallel trends is satisfied. Again, we use the simulated dataset from [example 4](#). We fit the following model:

```
. xtset id3
Panel variable: id3 (balanced)
. xtdidregress (y3 c.w1##c.w2) (treated3), group(id3) time(t3)
Treatment and time information
Time variable: t3
Control:      treated3 = 0
Treatment:    treated3 = 1
```

		Control	Treatment
Group			
	id3	502	498
Time			
	Minimum	1	6
	Maximum	1	6

Difference-in-differences regression
Data type: Longitudinal

Number of obs = 10,000

(Std. err. adjusted for 1,000 clusters in id3)

y3	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
ATET treated3 (Treated vs Untreated)	.4996049	.0102458	48.76	0.000	.4794991	.5197107

Note: ATET estimate adjusted for covariates, panel effects, and time effects.

Using estat trendplots, we obtain

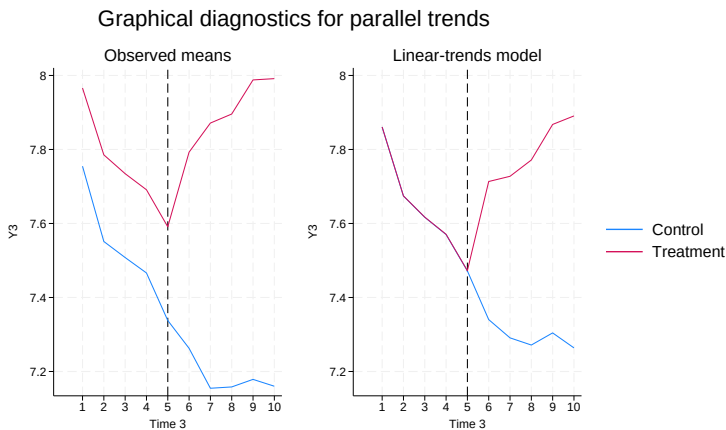


Figure 4.

This time, the trajectories of the observed means appear to be parallel before the treatment occurs. Both follow a declining trend up to the last pretreatment time point. Looking at the results from the linear-trends model, the pretreatment trajectories appear to be the same. Using estat ptrends yields the following result:

```
. estat ptrends
Parallel-trends test (pretreatment time period)
H0: Linear trends are parallel
F(1, 999) = 0.00
Prob > F = 0.9688
```

We cannot reject the null hypothesis of parallel linear trends. Likewise, using estat granger, we do not reject the null hypothesis of the absence of anticipatory effects:

```
. estat granger
Granger causality test
H0: No effect in anticipation of treatment
F(4, 999) = 0.52
Prob > F = 0.7220
```

► Example 7: Time-specific treatment effects

As we have seen in the previous examples, we can use `estat granger` to test for treatment effects occurring prior to the point at which treatment has been administered. Beyond the particular purpose of this test, however, it can be instructive to inspect not only pretreatment effects but also posttreatment effects that are allowed to vary over time. That is, rather than relying on a single treatment-effect estimate that is assumed to be constant over time, it can be instructive to check whether treatment effects change as time unfolds. We can fit a model that includes lags and leads of an indicator for the time period in which the treatment began. The coefficients on the lags can be used to assess whether there is any change in the treatment effect during posttreatment time periods. We can use the postestimation command `estat grangerplot` to fit such a model and plot the resulting coefficients. This model is also sometimes referred to as an event study model in the literature; see [Schmidheiny and Siegloch \(2019\)](#) and [Clarke and Tapia-Schythe \(2021\)](#).

We start by fitting the following DID model:

```
. use https://www.stata-press.com/data/r19/hospdd
(Artificial hospital admission procedure data)
. didregress (satis) (procedure), group(hospital) time(month)
Treatment and time information
Time variable: month
Control:      procedure = 0
Treatment:    procedure = 1
```

	Control	Treatment
Group hospital	28	18
Time Minimum	1	4
Maximum	1	4

Difference-in-differences regression
Data type: Repeated cross-sectional

Number of obs = 7,368

(Std. err. adjusted for 46 clusters in hospital)

satis	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
ATET procedure (New vs Old)	.8479879	.0321121	26.41	0.000	.7833108	.912665

Note: ATET estimate adjusted for group effects and time effects.

We can now simply use `estat grangerplot` to plot pre- and postintervention treatment effects:

```
. estat grangerplot
```

This yields the following graph:

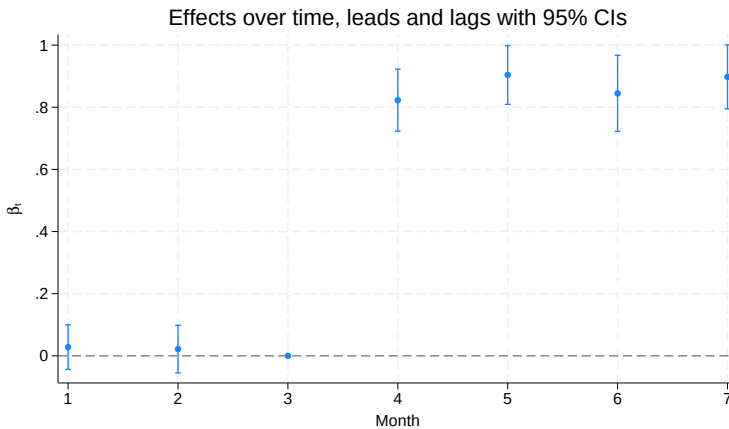


Figure 5.

By default, `estat grangerplot` uses all available leads and lags, and in this case, we have estimates for two leads and four lags. For identification purposes, we must omit one of the indicator variables. By default, `estat grangerplot` omits the first lead, which sets the coefficient to 0 that corresponds to the time period prior to treatment administration. Thus, the plotted effects are normalized with respect to that time period. If we wanted, we could change the baseline period by using the `baseline()` option. We can see that the coefficients on the leads that correspond to the first two months are both close to 0, which shows that there are no substantial effects in the pretreatment era. The posttreatment effects for months 4 to 7 range between around 0.8 and 0.9. When we account for the uncertainty of these estimates, as indicated by the plotted 95% confidence intervals, it appears as though the treatment effects are rather stable over time in this example.

Rather than just plotting the coefficients, we could be interested in looking at a table with the numeric results. We can obtain the underlying results by specifying the verbose option. We are also using the nodraw option to see only the numeric results and not draw the graph again:

```
. estat grangerplot, nodraw verbose
```

Linear regression, absorbing indicators	Number of obs	=	7,368
Absorbed variable: hospital	No. of categories	=	46
	F(12, 45)	=	94.68
	Prob > F	=	0.0000
	R-squared	=	0.5334
	Adj R-squared	=	0.5298
	Root MSE	=	0.7240

(Std. err. adjusted for 46 clusters in hospital)

satis	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
month						
February	-.007044	.0263953	-0.27	0.791	-.0602068	.0461188
March	.0335696	.0255925	1.31	0.196	-.0179764	.0851156
April	.0187852	.0250623	0.75	0.457	-.0316927	.0692632
May	-.0211152	.0269569	-0.78	0.438	-.0754092	.0331788
June	.0091208	.0179016	0.51	0.613	-.026935	.0451766
July	-.0203444	.0306266	-0.66	0.510	-.0820296	.0413407
_lead3	.027897	.035569	0.78	0.437	-.0437426	.0995367
_lead2	.0217322	.0380076	0.57	0.570	-.054819	.0982833
_lag0	.8228153	.0494933	16.62	0.000	.7231307	.9224999
_lag1	.9040498	.0469682	19.25	0.000	.8094511	.9986486
_lag2	.844724	.0608006	13.89	0.000	.7222654	.9671826
_lag3	.8978885	.0511588	17.55	0.000	.7948494	1.000928
_cons	3.433074	.0198449	173.00	0.000	3.393104	3.473044

The coefficients on the variables that begin with _lead and _lag are the ones that are plotted in our graph.

As an interesting side note, notice that our original DID model is a special case of the model that we fit with `estat grangerplot`. Specifically, it is the model that includes only a single lead and no lags. To reproduce our original DID result, we can use the `nleads()` and `nlags()` options:

```
. estat grangerplot, nodraw verbose nleads(1) nlags(0)
Linear regression, absorbing indicators      Number of obs      = 7,368
Absorbed variable: hospital                No. of categories = 46
                                           F(7, 45)           = 138.73
                                           Prob > F            = 0.0000
                                           R-squared           = 0.5333
                                           Adj R-squared       = 0.5299
                                           Root MSE            = 0.7238

                                (Std. err. adjusted for 46 clusters in hospital)
```

satis	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
month						
February	-.0096077	.0184317	-0.52	0.605	-.0467311	.0275158
March	.0219686	.018251	1.20	0.235	-.0147907	.0587279
April	-.0032839	.0221028	-0.15	0.883	-.0478013	.0412335
May	-.0094027	.0232399	-0.40	0.688	-.0562103	.0374048
June	-.0038375	.0190634	-0.20	0.841	-.0422332	.0345581
July	-.0111941	.0230029	-0.49	0.629	-.0575244	.0351361
_lag0	.8479879	.0321121	26.41	0.000	.7833108	.912665
_cons	3.444675	.011354	303.39	0.000	3.421807	3.467543

As we can see, the results match the ones we obtained earlier with `didregress`.

Notice also that the model used by `estat granger` is a special case as well; it is the model with all available leads and no lags. To re-create the test result from `estat granger`, we use the `post` option to store the results:

```
. estat grangerplot, nodraw verbose post nlags(0)
Linear regression, absorbing indicators      Number of obs      = 7,368
Absorbed variable: hospital                No. of categories  = 46
                                           F(9, 45)           = 113.78
                                           Prob > F            = 0.0000
                                           R-squared           = 0.5333
                                           Adj R-squared       = 0.5298
                                           Root MSE            = 0.7239

                               (Std. err. adjusted for 46 clusters in hospital)
```

satis	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
month						
February	-.007044	.0263899	-0.27	0.791	-.0601959	.0461079
March	.0335696	.0255873	1.31	0.196	-.0179658	.085105
April	.0002573	.0243547	0.01	0.992	-.0487955	.0493101
May	-.0058616	.0249982	-0.23	0.816	-.0562105	.0444874
June	-.0002964	.0199414	-0.01	0.988	-.0404605	.0398678
July	-.007653	.0251828	-0.30	0.763	-.0583738	.0430679
_lead3	.027897	.0355617	0.78	0.437	-.043728	.099522
_lead2	.0217322	.0379998	0.57	0.570	-.0548033	.0982676
_lag0	.8673694	.0424929	20.41	0.000	.7817844	.9529544
_cons	3.433074	.0198408	173.03	0.000	3.393113	3.473035

We can now test whether the coefficients on the leads are jointly 0:

```
. test _lead3 _lead2
( 1) _lead3 = 0
( 2) _lead2 = 0
      F( 2, 45) = 0.33
      Prob > F = 0.7239
```

The test result indicates that we cannot reject the null hypothesis that the coefficients are jointly 0. We fit our original model again to compare the results with the ones from `estat granger`:

```
. quietly didregress (satis) (procedure), group(hospital) time(month)
. estat granger
Granger causality test
H0: No effect in anticipation of treatment
F(2, 45) = 0.33
Prob > F = 0.7239
```

As we can see, the results are the same. Using `estat grangerplot` followed by `test` allows us to perform additional tests of lags and leads that are not available through `estat granger`.

Specifying a 2-by-2 DID

▷ Example 8: Specifying a 2-by-2 DID

`didregress` and `xtdidregress` by default fit generalized DID models, also known as two-way fixed-effects models. Yet DID models sometimes are viewed from a two-period and two-group perspective, a 2-by-2 DID. You can also fit a 2-by-2 DID using `didregress` and `xtdidregress`. In fact, you will get equivalent results using a 2-by-2 DID or a generalized DID. The generalized DID is the default because it allows for a wider range of specifications that would not be feasible within the 2-by-2 framework, such as cases when the intervention occurs at different points in time.

In [example 1](#), we got the ATET for hospitals that instituted a new admissions procedure. We typed the following:

```
. use https://www.stata-press.com/data/r19/hospdd
. didregress (satis) (procedure), group(hospital) time(month)
```

This implies that we are regressing `satis` on `procedure` and indicators for hospitals and for months. The indicators are created and added as regressors to our model by default. To fit a 2-by-2 model, we need to omit the hospital and month indicators and instead add an indicator for the period after treatment and an indicator for groups that are treated.

This is what we do below. We create the treatment-group indicator based on the fact that the variable `procedure` is 1 for individuals that experienced the new procedure and 0 otherwise; this variable identifies treated hospitals. We create the posttreatment indicator based on the fact that the new procedures are established after March, which has a value of 3 in the data.

```
. bysort hospital: egen treated = max(procedure)
. generate post = month>3
```

We now fit

```
. didregress (satis i.treated i.post) (procedure), nogteffects
> group(hospital) time(month)
```

Treatment and time information

Time variable: month

Control: procedure = 0

Treatment: procedure = 1

	Control	Treatment
Group		
hospital	28	18
Time		
Minimum	1	4
Maximum	1	4

Difference-in-differences regression
 Data type: Repeated cross-sectional
 Number of obs = 7,368
 (Std. err. adjusted for 46 clusters in hospital)

satis	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
ATET procedure (New vs Old)	.8479879	.0320051	26.50	0.000	.7835263	.9124494

Note: ATET estimate adjusted for covariates.

We add the new indicators to our list of covariates and use the `nogteffects` option to exclude the group and time indicators that were included by default.

The point estimates are identical. What changes are the standard errors. They change because we have a different number of regressors.



Standard error considerations

► Example 9: Standard errors and data aggregation

In [example 3](#), we had 336 treated subclasses and 7,248 subclasses in total. However, many studies face the challenge of having very few elements per group. In fact, sometimes the data analyzed consist of only two groups—the treatment group in which treatment is administered to members of the group and the controls group in which no treatment is administered. Think, for example, of an analysis at the state level where one state is treated and one state is controlled. As discussed in [\[CAUSAL\] DID intro](#), these scenarios with few elements pose a challenge for inference. For a good discussion on these issues, see [MacKinnon \(2019\)](#).

`didregress` and `xtdidregress` provide alternatives in such cases. One alternative is to compute standard errors by using the wild cluster–bootstrap. Another alternative is to use bias-corrected clustered standard errors with the degrees-of-freedom adjustment proposed by [Bell and McCaffrey \(2002\)](#). A final alternative is to aggregate your data and then compute effects and standard errors such as proposed by [Donald and Lang \(2007\)](#) or use bias-corrected standard errors.

Below, we explore these options using simulated data. We created a dataset with 2,000 individuals and five time periods. The treatment occurs at the county level and there are six counties. Two of the counties, county 1 and county 2, receive the treatment and the remaining four counties do not. The true value of the ATET is -1.0 . As we see below, the groups are unbalanced.

```
. use https://www.stata-press.com/data/r19/smallg
(Simulated data with a small number of groups)
. tab county
```

County	Freq.	Percent	Cum.
1	715	7.15	7.15
2	2,570	25.70	32.85
3	3,410	34.10	66.95
4	2,285	22.85	89.80
5	920	9.20	99.00
6	100	1.00	100.00
Total	10,000	100.00	

First, we fit the model using the default standard errors, which perform well with many balanced groups but not necessarily well with data like these. We compare the other results with these results.

```
. didregress (outcome x i.b) (treated), group(county) time(year)
```

Treatment and time information

Time variable: year

Control: treated = 0

Treatment: treated = 1

	Control	Treatment
Group		
county	4	2
Time		
Minimum	2011	2013
Maximum	2011	2013

Difference-in-differences regression

Number of obs = 10,000

Data type: Repeated cross-sectional

(Std. err. adjusted for 6 clusters in county)

outcome	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
ATET						
treated (Treated vs Untreated)	-.9394987	.0884134	-10.63	0.000	-1.166773	-.7122247

Note: ATET estimate adjusted for covariates, group effects, and time effects.

For this draw, the point estimates are reasonably close to the true value of -1.0 , and the true values are contained inside the confidence interval.

Next, we use the wild cluster–bootstrap. For a good introduction to the methodology, see [Cameron, Gelbach, and Miller \(2008\)](#), [MacKinnon and Webb \(2018\)](#), and [Roodman et al. \(2019\)](#). The wild cluster–bootstrap works well in scenarios like the one above, where there are few groups.

We use the wild cluster bootstrap to construct p -values and confidence intervals, imposing the null hypothesis that the ATET is 0. We describe both computations in [Methods and formulas](#). Here are the results:

```
. didregress (outcome x i.b) (treated), group(county) time(year)
> wildbootstrap(rseed(123) errorweight(webb))

Performing 1,000 replications for p-value for constraint
treated = 0 ...
Computing confidence interval for treated
  Lower bound: .....10.....20..... done (26)
note: lower-bound CI achieved  $F(-1.25) = 0.0240$ , but target is  $F(x) = .025$ .
note: the sorted bootstrap  $t$  statistics have at least two tied values
      adjacent to the  $t$  statistic under the null; this prevents the CI bound
      from converging to the target.
  Upper bound: .....10.....20..... done (25)
Treatment and time information
Time variable: year
Control:      treated = 0
Treatment:    treated = 1
```

	Control	Treatment
Group		
county	4	2
Time		
Minimum	2011	2013
Maximum	2011	2013

```
DID with wild-cluster bootstrap inference          Number of obs = 10,000
                                                    Replications = 1,000

Data type:    Repeated cross-sectional
Error weight: webb
```

outcome	Coefficient	t	P> t	[95.10% conf. interval]	
ATET					
treated					
(Treated vs Untreated)	-.9394987	-10.63	0.020	-1.248532	-.5549851

```
Note: 95.10% confidence interval is wider than requested.
Note: ATET estimate adjusted for covariates, group effects, and time effects.
```

Above, we first see the iterations used to find the confidence interval lower bound and upper bound. The optimization algorithm sometimes converges to a level that is below the test size. In such cases, the confidence interval is conservative. In the example above, instead of a 95% confidence interval, you obtain a 95.1% confidence interval. If there are t statistics that have exactly the same value across bootstrap replications, the algorithm will not solve exactly for the requested confidence level. We also use `errorweight()` with error weight `webb` to compute the wild bootstrap. This is best in cases with less than 10 groups, as suggested by [Roodman et al. \(2019\)](#). The results from the wild bootstrap suggest more uncertainty than the default confidence interval.

Now we look at two data aggregation methods. In both cases, data aggregation occurs in two steps. In the first step, we regress the outcome on the set of covariates that vary at the individual, group, and time levels. We then estimate the group–time fixed effects from this procedure to use in a second stage as the new dependent variable. In the second stage, we aggregate the remaining covariates and the new dependent variable at the group and time levels and run a regression. This aggregation strategy is described in more detail in [Bertrand, Duflo, and Mullainathan \(2004\)](#), [Donald and Lang \(2007\)](#), and [Cameron and Miller \(2015\)](#). We also describe the aggregation methods in *Methods and formulas*.

Below, we aggregate the data as described above and estimate bias-corrected standard errors by using the degrees-of-freedom adjustment suggested by [Bell and McCaffrey \(2002\)](#). This method of getting standard errors is computationally intensive, so it is well suited for cases where the dimensionality of the problem is reduced via aggregation.

```
. didregress (outcome x i.b) (treated), group(county) time(year)
> aggregate(standard) vce(hc2)
```

Computing degrees of freedom ...

Treatment and time information

Time variable: year

Control: treated = 0

Treatment: treated = 1

		Control	Treatment
Group			
	county	4	2
Time			
	Minimum	2011	2013
	Maximum	2011	2013

Difference-in-differences regression

Number of obs = 30

No. of clusters = 6

Data type: Repeated cross-sectional

Aggregation: Standard

outcome	Robust HC2		t	P> t	[95% conf. interval]	
	Coefficient	std. err.				
ATET treated (Treated vs Untreated)	-.9958521	.1373277	-7.25	0.017	-1.566242	-.4254624

Note: ATET estimate adjusted for covariates, group effects, and time effects.

We see that the confidence intervals are again wider than with the default standard errors. It is also worth noticing that we went from 10,000 observations to 30. This is because we aggregated at the county and year levels. We have six counties and five years.

We could also aggregate our data and compute the t statistics using the degrees of freedom proposed by [Donald and Lang \(2007\)](#). This gives us

```
. didregress (outcome x i.b) (treated), group(county) time(year)
> aggregate(dlang)
```

Treatment and time information

Time variable: year

Control: treated = 0

Treatment: treated = 1

	Control	Treatment
Group		
county	4	2
Time		
Minimum	2011	2013
Maximum	2011	2013

Difference-in-differences regression

Number of obs = 30

Data type: Repeated cross-sectional

Aggregation: Donald-Lang

outcome	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
ATET						
treated						
(Treated						
vs						
Untreated)	-.9958521	.1224496	-8.13	0.000	-1.248576	-.7431287

Note: ATET estimate adjusted for covariates, group effects, and time effects.

Again, we see wider confidence intervals.

Whenever you have few elements for each group and the groups are unbalanced, as in this example, you should be careful not to base your conclusions solely on the default cluster-robust standard errors. As was mentioned in [\[CAUSAL\] DID intro](#), you should validate your conclusions by using one of the standard error computations suggested above.

Default cluster–robust standard errors

► Example 10: Default cluster–robust standard errors with didregress and xtddidregress

We have stated that `didregress` was designed to handle repeated cross-sections and `xtddidregress` was designed to handle panel datasets. However, you can use `xtddidregress` when you have repeated cross-sections. For instance, for the hospital dataset in [example 1](#), you may have typed

```
. use https://www.stata-press.com/data/r19/hospdd
(Artificial hospital admission procedure data)
. xtset hospital
Panel variable: hospital (unbalanced)
. xtddidregress (satis) (procedure), group(hospital) time(month)
Treatment and time information
Time variable: month
Control:      procedure = 0
Treatment:    procedure = 1
```

	Control	Treatment
Group hospital	28	18
Time		
Minimum	1	4
Maximum	1	4

Difference-in-differences regression Number of obs = 7,368
Data type: Longitudinal

(Std. err. adjusted for 46 clusters in hospital)

		Robust			
satis	Coefficient	std. err.	t	P> t	[95% conf. interval]
ATET procedure (New vs Old)	.8479879	.0320138	26.49	0.000	.7835088 .9124669

Note: ATET estimate adjusted for panel effects and time effects.

You will get the same point estimate as you would get with `didregress` but different standard errors. The reason is that `xtddidregress` with cluster–robust standard errors does not count the group variables as regressors in the degrees-of-freedom correction used to compute standard errors. It relies on the asymptotic theory of fixed-effects regression, where the number of group effects are expected to grow with the sample size. Put differently, `xtddidregress` is using `xtreg, fe` to compute the default cluster–robust standard errors. `didregress` uses `areg`, which assumes the number of groups is fixed and counts them in the degrees-of-freedom computation.

As a practical matter, these standard errors are going to be close to each other when the number of observations per cluster is large, as is the case for this example. As the number of observations per cluster becomes smaller, however, the standard errors will differ more, with standard errors of `didregress` tending to be larger.

Which estimator to use depends on the assumptions you would like to make about your data. If you think your data should be treated as panel data, with the number of groups growing with the sample size, then even if you have repeated cross-sections, you should use `xtdidregress`.

More in-depth discussions about cluster-robust standard errors can be found in [Cameron and Miller \(2015\)](#) and [Wooldridge \(2010\)](#).



Stored results

`didregress` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_clust)</code>	number of clusters
<code>e(tmin)</code>	minimum of first observed treatment time across groups
<code>e(tmax)</code>	maximum of first observed treatment time across groups
<code>e(N_reps)</code>	number of bootstrap replications
<code>e(df_r)</code>	residual degrees of freedom
<code>e(blocksize)</code>	block size used in wild bootstrap computations

Macros

<code>e(cmd)</code>	<code>didregress</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(treatment)</code>	indicator for treated observations
<code>e(treatment_type)</code>	binary or continuous
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(wb_weight)</code>	wild bootstrap error weight
<code>e(datatype)</code>	data type
<code>e(groupvars)</code>	group variables
<code>e(clustvar)</code>	name of cluster variable
<code>e(timevar)</code>	time variable
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b</code> <code>V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by margins
<code>e(_contrast_not_ok)</code>	prediction disallowed by contrast

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance-covariance matrix of the estimators
<code>e(group_count)</code>	matrix with number of groups and treatment time
<code>e(adj_ci)</code>	adjusted confidence intervals when <code>vce(hc3, hansen)</code> is specified
<code>e(adj_p)</code>	adjusted <i>p</i> -values when <code>vce(hc3, hansen)</code> is specified
<code>e(fwboot)</code>	wild bootstrap estimates
<code>e(wboot)</code>	wild bootstrap statistics and constraint
<code>e(aggmethod)</code>	aggregation method

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices	
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

`xtdidregress` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_clust)</code>	number of clusters
<code>e(tmin)</code>	minimum of first observed treatment time across groups
<code>e(tmax)</code>	maximum of first observed treatment time across groups
<code>e(N_reps)</code>	number of bootstrap replications
<code>e(df_r)</code>	residual degrees of freedom
<code>e(blocksize)</code>	block size used in wild bootstrap computations

Macros

<code>e(cmd)</code>	<code>xtdidregress</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(treatment)</code>	indicator for treated observations
<code>e(treatment_type)</code>	binary or continuous
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(wb_weight)</code>	wild bootstrap error weight
<code>e(datatype)</code>	data type
<code>e(groupvars)</code>	group variables
<code>e(clustvar)</code>	name of cluster variable
<code>e(panelvar)</code>	panel variable
<code>e(timevar)</code>	time variable
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(_contrast_not_ok)</code>	prediction disallowed by <code>contrast</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators
<code>e(group_count)</code>	matrix with number of groups and treatment time
<code>e(adj_ci)</code>	adjusted confidence intervals when <code>vce(hc3, hansen)</code> is specified
<code>e(adj_p)</code>	adjusted <i>p</i> -values when <code>vce(hc3, hansen)</code> is specified
<code>e(fwboot)</code>	wild bootstrap estimates
<code>e(wboot)</code>	wild bootstrap statistics and constraint
<code>e(aggmethod)</code>	aggregation method

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices	
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

Methods and formulas are presented under the following headings:

DID for repeated cross-sectional data
DDD model
DID and DDD models with longitudinal data
Aggregation estimators
Wild bootstrap confidence intervals and p-values
Bias-corrected clustered standard errors

DID for repeated cross-sectional data

The DID model for repeated cross-sectional data fit by `didregress` is given by

$$y_{ist} = \gamma_s + \gamma_t + \mathbf{z}_{ist}\boldsymbol{\beta} + D_{st}\delta + \epsilon_{ist} \quad (1)$$

where i is the observation-level index, s is a group-level index, and t is a time-level index. For example, we might have yearly repeated cross-sectional data for individuals living in different states; in this case, i denotes the individual, s the state, and t the year. In (1) above, γ_s are group fixed effects and γ_t are time fixed effects. $\mathbf{z}_{ist}$ are the covariates, and ϵ_{ist} is the error term. D_{st} is the treatment that varies at the group and time levels. D_{st} could be binary or continuous.

If y_{ist} is y in the data, then $\mathbf{z}_{ist}$ are $z1$ and $z2$, D_{st} is d , the group is `state`, and time is `year`. To fit the model, you would type

```
didregress (y z1 z2) (d), group(state) time(year)
```

which is equivalent to typing

```
areg y z1 z2 i.year d, absorb(state) vce(cluster state)
```

The methods and formulas for this model are those of a linear regression and can be found in [Methods and formulas](#) in [R] `areg`.

DDD model

The DDD model is given by

$$y_{isgt} = \gamma_s + \gamma_g + \gamma_t + \gamma_s\gamma_t + \gamma_g\gamma_t + \gamma_s\gamma_g + \mathbf{z}_{isgt}\boldsymbol{\beta} + D_{sgt}\delta + \epsilon_{isgt} \quad (2)$$

where i is the observation-level index, s and g are group-level indices, and t is a time-level index. For example, we might have yearly repeated cross-sectional data for older and younger individuals living in different states. In this case, i denotes the individual, s the state, g age group, and t the year. In (2) above, γ_s are group s fixed effects, γ_g are group g fixed effects, and γ_t are time fixed effects. $\mathbf{z}_{isgt}$ are the covariates, and ϵ_{isgt} is the error term. D_{sgt} denotes the treatment that varies at the group s , group g , and time levels. D_{sgt} could be binary or continuous.

Say you now want to fit a DDD, where the treatment occurs for group2 in some of the states, state. You would type

```
didregress (y z1 z2) (d), group(state group2) time(year)
```

didregress would construct the group and time interactions in (2) and fit a linear regression. The methods and formulas for the specification are the same as those in [Methods and formulas](#) in [R] areg.

DID and DDD models with longitudinal data

The DID model for longitudinal data fit by xtddidregress is given by

$$y_{ist} = \alpha_i + \gamma_t + \mathbf{z}_{ist}\boldsymbol{\beta} + D_{st}\delta + \epsilon_{ist} \quad (3)$$

where i is the observation-level index, s is a group-level index, and t is a time-level index. For example, we might have a longitudinal dataset of individuals over time living in a given state; in this case, i denotes the individual, s the state, and t the year. In (3) above, α_i are individual fixed effects and γ_t are time fixed effects. $\mathbf{z}_{ist}$ are the covariates, and ϵ_{ist} is the error term. D_{st} denotes the treatment that varies at the group and time levels. D_{st} could be binary or continuous. Individuals, i , are assumed to be nested within the group; thus, group effects are subsumed by the individual effects.

If y_{ist} is y in the data, then $\mathbf{z}_{ist}$ are $z1$ and $z2$, D_{st} is d , the group is $state$, time is $year$, and individuals are denoted by id . To fit the model, you would type

```
xtset id year
xtddidregress (y z1 z2) (d), group(state) time(year)
```

which is equivalent to typing

```
xtreg y z1 z2 i.year d, fe vce(cluster state)
```

The methods and formulas for this model are those of a within estimator computed by xtreg, fe and can be found in [Methods and formulas](#) in [XT] xtreg.

The DDD model is given by

$$y_{isgt} = \alpha_i + \gamma_t + \gamma_t\gamma_s + \gamma_t\gamma_g + \mathbf{z}_{ist}\boldsymbol{\beta} + D_{st}\delta + \epsilon_{isgt}$$

Aggregation estimators

To discuss the aggregation estimators, it is instructive to rewrite the covariate vector $\mathbf{z}_{ist}$ as consisting of group and time invariant components $\mathbf{z}_{1st}$ and time-, group-, and individual-varying components $\mathbf{z}_{2ist}$. The DID model can now be expressed as

$$\begin{aligned} y_{ist} &= \gamma_s + \gamma_t + \mathbf{z}_{1st}\beta_1 + \mathbf{z}_{2ist}\beta_2 + D_{st}\delta + \epsilon_{ist} \\ y_{ist} &= \mathbf{z}_{2ist}\beta_2 + C_{st} + \epsilon_{ist} \end{aligned} \quad (4)$$

$$C_{st} = \gamma_s + \gamma_t + \mathbf{z}_{1st}\beta_1 + D_{st}\delta + \nu_{st} \quad (5)$$

The standard and `dlang`, constant aggregation methods regress y_{ist} on $\mathbf{z}_{2ist}$ in (4) and then obtain an estimate of C_{st} , $\widehat{C}_{st}$. After we regress y_{ist} on $\mathbf{z}_{2ist}$, we get an estimate of the group–time effects $\widehat{C}_{st}$ from $y_{ist} - \mathbf{z}_{2ist}\widehat{\beta}_2 - \widehat{\epsilon}_{ist}$. These effect estimates come from using `predict` with option `d` after `areg`. We then aggregate the data at the s and t levels. With the aggregated data, we run a regression of $\widehat{C}_{st}$ on $\mathbf{z}_{1st}$ and on D_{st} using group and time fixed effects in (5). The `dlang` method computes the ordinary least-squares standard errors from this second stage. The `standard` method uses cluster–robust standard errors clustered at the group level by default.

The `dlang`, varying method runs a regression for each group defined by s and t and obtains an estimate of C_{st} as the set of constants for each regression. Whereas in the first case β_2 was the same vector across the sample, in this case, we will get a different estimate of the slope coefficient for each group. The second step is the same as for `dlang`, constant.

Wild bootstrap confidence intervals and p-values

The wild cluster–bootstrap is a bit different from the conventional bootstrap in that it keeps the covariates fixed and constructs the new dependent variables using the residuals from the original regression. The wild bootstrap procedure proceeds as follows:

1. Fit a restricted model, where the ATET is constrained to be 0. In other words, imposing the null hypothesis that the ATET is 0. Obtain the predicted values, $\widehat{\mathbf{y}}_r$, and residuals, $\widehat{\mathbf{u}}_r$, from the restricted model, where the subscript r refers to the bootstrap replicates. Also fit the unrestricted model and compute the t statistic, t , testing the null hypothesis $\text{ATET} = 0$.
2. At each of the subsequent $B - 1$ bootstrap steps, compute $\mathbf{y}^* = \widehat{\mathbf{y}}_r + \widehat{\mathbf{u}}_r \odot (\mathbf{K}\mathbf{w})$, where $\mathbf{w}$ is a column vector of length S , the number of clusters, containing the wild bootstrap weights, and $\mathbf{K}$ is an $N \times S$ matrix with elements $k_{i,j} = 1$ if observation i , $i = 1, \dots, N$, is in group j and 0 otherwise. The operator $\odot$ is the Hadamard product that performs elementwise multiplication. By default, $w_j = 1$ with probability 0.5 and $w_j = -1$ with probability 0.5, that is, the `rademacher` error weights.

Alternatively, use one of the following error weights w :

- `mammen`, which is $1 - \phi$ with probability $\phi/\sqrt{5}$ and ϕ otherwise, where $\phi = (1 + \sqrt{5})/2$.
 - `webb`, which takes the values $-\sqrt{3/2}$, $-\sqrt{2/2}$, $-\sqrt{1/2}$, $\sqrt{1/2}$, $\sqrt{2/2}$, and $\sqrt{3/2}$, each with probability $1/6$.
 - `normal`, which is a normal distribution with the first four moments given by 0, 1, 0, and 3.
 - `gamma`, which is a gamma distribution with shape parameter 4 and scale parameter $1/2$.
3. For each bootstrap sample, compute the unrestricted model and the null hypothesis t statistic, t^* . Include the observed t statistic, t in this set.

4. Given the B bootstrapped t statistics t_b^* , $b = 1, \dots, B$, compute the wild bootstrap lower-tail p -value

$$P_l(t) = \frac{1}{B} \sum_{b=1}^B I(|t| < |t_b^*|)$$

and the upper-tail p -value

$$P_u(t) = \frac{1}{B} \sum_{b=1}^B I(|t| > |t_b^*|)$$

The reported equal-tail p -value is $P_{et}(t) = 2 \min\{P_l(t), P_u(t)\}$.

5. Compute the wild bootstrap confidence interval by searching for the lower limit and upper limit separately using the bisection root-finding algorithm. In doing so, find the quantities $ATET = c_l$ and $ATET = c_u$, $c_l < c_u$, such that $(100 - L)/2\%$ of the bootstrapped distribution of the t statistics testing $ATET = c$ is contained in either the lower or the upper tail. By default, $L = 95$.

The search algorithm does not always generate an $L\%$ confidence interval. One reason for this situation is if $B(100 - L)/200$ is not an integer, then there is not a slot in the sorted vector $\mathbf{t}^*$ that corresponds to the upper- or lower-tail area $(100 - L)/200$. See [Roodman et al. \(2019, 8\)](#) for a good discussion.

Also, although a low-probability event, if at least one of the computed $\mathbf{t}^*$ is (numerically) equal to the t testing $ATET = c$, or if there are ties between other computed values in $\mathbf{t}^*$ adjacent to the t testing $ATET = c$, then the algorithm will not achieve the optimal coverage. In this case, we choose the confidence interval that is more conservative than requested. The achieved confidence level is reported in the ATET coefficient table.

The chance of ties in $\mathbf{t}^*$ increases if the number of clusters, S , is small and the rademacher error weights are used, because there are 2^S possible combinations of the two values 1 and -1 in the weight vector $\mathbf{w}$. For example, if $S = 10$, then there are 1,024 possible combinations.

Bias-corrected clustered standard errors

The formulas for the bias-corrected VCEs implemented by `vce(hc2 [clustvar])` and `vce(hc3 [clustvar])` can be found in [Methods and formulas](#) in [\[R\] regress](#). The formulas for the critical values used in confidence intervals, the adjusted degrees of freedom, and the scaled t statistics implemented by the `hansen` suboption can be found in [Methods and formulas](#) in [\[R\] areg](#) and [Methods and formulas](#) in [\[R\] regress](#).

Acknowledgment

The algorithm used to compute the wild bootstrap was made faster using techniques described in [Roodman et al. \(2019\)](#). We thank David Roodman for bringing this to our attention.

References

- Abadie, A., and M. D. Cattaneo. 2018. Econometric methods for program evaluation. *Annual Review of Economics* 10: 465–503. <https://doi.org/10.1146/annurev-economics-080217-053402>.
- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist’s Companion*. Princeton, NJ: Princeton University Press.
- . 2015. *Mastering Metrics: The Path from Cause to Effect*. Princeton, NJ: Princeton University Press.
- Bell, R. M., and D. F. McCaffrey. 2002. Bias reduction in standard errors for linear regression with multi-stage samples. *Survey Methodology* 28: 169–181.
- Bertrand, M., E. Duflo, and S. Mullainathan. 2004. How much should we trust difference-in-differences estimates? *Quarterly Journal of Economics* 119: 249–275. <https://doi.org/10.1162/003355304772839588>.
- Blundell, R. W., and M. C. Dias. 2009. Alternative approaches to evaluation in empirical microeconomics. *Journal of Human Resources* 44: 565–640. <https://doi.org/10.1007/s10258-002-0010-3>.
- Cameron, A. C., J. B. Gelbach, and D. L. Miller. 2008. Bootstrap-based improvements for inference with clustered errors. *Review of Economics and Statistics* 90: 414–427. <https://doi.org/10.1162/rest.90.3.414>.
- Cameron, A. C., and D. L. Miller. 2015. A practitioner’s guide to cluster-robust inference. *Journal of Human Resources* 50: 317–372. <https://doi.org/10.3368/jhr.50.2.317>.
- Clarke, D., and K. Tapia-Schythe. 2021. Implementing the panel event study. *Stata Journal* 21: 853–884.
- Donald, S. G., and K. Lang. 2007. Inference with difference-in-differences and other panel data. *Review of Economics and Statistics* 89: 221–233. <https://doi.org/10.1162/rest.89.2.221>.
- Hansen, B. E. 2025a. Jackknife standard errors for clustered regression. Unpublished manuscript, University of Wisconsin, <https://users.ssc.wisc.edu/~bhansen/papers/tcauchy.pdf>.
- . 2025b. Standard errors for difference-in-difference regression. *Journal of Applied Econometrics* 40: 291–309. <https://doi.org/10.1002/jae.3110>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Lechner, M. 2011. The estimation of causal effects by difference-in-difference methods. *Foundations and Trends in Econometrics* 4: 165–224. <https://doi.org/10.1561/08000000014>.
- MacKinnon, J. G. 2019. How cluster-robust inference is changing applied econometrics. *Canadian Journal of Economics* 52: 851–881. <https://doi.org/10.1111/caje.12388>.
- MacKinnon, J. G., and M. D. Webb. 2018. The wild bootstrap for few (treated) clusters. *Econometrics Journal* 21: 114–135. <https://doi.org/10.1111/ectj.12107>.
- Moser, P., and A. Voena. 2012. Compulsory licensing: Evidence from the Trading with the Enemy Act. *American Economic Review* 102: 396–427. <https://doi.org/10.1257/aer.102.1.396>.
- Roodman, D., J. G. MacKinnon, M. Ø. Nielsen, and M. D. Webb. 2019. *Fast and wild: Bootstrap inference in Stata using boottest*. *Stata Journal* 19: 4–60.
- Schmidheiny, K., and S. Sieglösch. 2019. On event study designs and distributed-lag models: Equivalence, generalization and practical implications. IZA Discussion Paper 12079, Institute of Labor Economics (IZA). <https://docs.iza.org/dp12079.pdf>.
- Wing, C., K. Simon, and R. A. Bello-Gomez. 2018. Designing difference in difference studies: Best practices for public health policy research. *Annual Review of Public Health* 39: 453–469. <https://doi.org/10.1146/annurev-publhealth-040617-013507>.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.
- Yan, G., and Q. Chen. 2022. *rcm: A command for the regression control method*. *Stata Journal* 22: 842–883.

Also see

[CAUSAL] **didregress postestimation** — Postestimation tools for didregress and xtdidregress

[CAUSAL] **DID intro** — Introduction to difference-in-differences estimation

[CAUSAL] **hdidregress** — Heterogeneous difference in differences⁺

[CAUSAL] **teffects intro** — Introduction to treatment effects for observational data

[CAUSAL] **teffects intro advanced** — Advanced introduction to treatment effects for observational data

[CAUSAL] **xthdidregress** — Heterogeneous difference in differences for panel data⁺

[CAUSAL] **xtswitchdid** — Difference in differences with switching treatments for panel data⁺

[U] **20 Estimation and postestimation commands**

Postestimation commands

Remarks and examples

References

[predict](#)

[Stored results](#)

[Also see](#)

[estat](#)

[Methods and formulas](#)

Postestimation commands

The following postestimation commands are of special interest after `didregress` and `xtdidregress`:

Command	Description
<code>estat trendplots</code>	graphical diagnostics for parallel trends
<code>estat ptrends</code>	parallel-trends test
<code>estat granger</code>	Granger causality test
<code>estat grangerplot</code>	time-specific treatment effects
<code>estat bdecomp</code>	treatment-effect decomposition

The following standard postestimation commands are also available:

Command	Description
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>forecast</code>	dynamic forecasts and simulations
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of parameters
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
<code>predict</code>	linear predictions and residuals
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as linear predictors and residuals.

Menu for predict

Statistics > Postestimation

Syntax for predict

```
predict [type] newvar [if] [in] [ , statistic]
```

statistic	Description
Main	
xb	linear predictor; the default
residuals	residuals

Options for predict

Main

`xb`, the default, calculates the linear predictor. It excludes the effect of the first group or of the panel identifier. All other effects, including the time fixed effects, are included in the linear predictor.

`residuals` calculates the overall residuals. It is the difference of the outcome and the linear predictor, including all group, panel, and time effects. In other words, it is not just the difference of the outcome and the linear predictor.

estat

Description for estat

`estat trendplots` produces two diagnostic plots for assessing the parallel-trends assumption that is required for consistent estimation of the ATET using `didregress` or `xtdidregress`. The first plot consists of two lines showing the mean of the outcome over time for the treatment and the control groups. The second plot augments the DID model to include interactions of time with an indicator of treatment and plots the predicted values of this augmented model for the treatment and control groups. Both plots include a vertical line one period before treatment.

`estat ptrends` performs a test of whether the linear trends in the outcome variable are parallel between control and treatment groups during the pretreatment period.

`estat granger` performs a test of whether treatment effects can be observed in anticipation of the treatment.

`estat grangerplot` produces a graph of time-specific treatment effects by plotting coefficients from leads and lags of the treatment indicator variable.

`estat bdecomp` performs a treatment-effect (Bacon) decomposition into all 2-by-2 treatment-effect components when there are multiple cohorts, each with a different treatment timing. Optionally, the components can be plotted against their weights. `estat bdecomp` requires that the data be strongly balanced. That is, units defined by the group variable must have the same number of observations, and observation times must be the same across all units.

Menu for estat

Statistics > Postestimation

Syntax for estat

Graphical diagnostics for parallel trends

```
estat trendplots [ , trend_options plot_options ]
```

Parallel-trends test

```
estat ptrends
```

Granger causality test

```
estat granger
```

Time-specific treatment effects

```
estat grangerplot [ , grangerplot_options ]
```

Treatment-effect decomposition

```
estat bdecomp [ , bdecomp_options ]
```

<i>trend_options</i>	Description
<code>omeans</code>	draw graph showing observed means
<code>omeans(plot_options)</code>	draw observed-means graph and affect its rendition
<code>ltrends</code>	draw graph showing linear trends
<code>ltrends(plot_options)</code>	draw linear-trends graph and affect its rendition
<code>notitle</code>	suppress overall title
<code>noxline</code>	suppress treatment-time reference line
<code>nocommonlegend</code>	display two individual legends
<code>legendfrom(#)</code>	specify which legend to use

<i>plot_options</i>	Description
---------------------	-------------

Plot	
<code>cline_options</code>	affect rendition of the plotted trend lines; see [G-3] <i>cline_options</i>
<code>line1opts(cline_options)</code>	affect rendition of the line for controls
<code>line2opts(cline_options)</code>	affect rendition of the line for treated
Y axis, X axis, Titles, Legend, Overall	
<code>twoway_options</code>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>

<i>grangerplot_options</i>	Description
<code>nleads(#)</code>	number of leads
<code>nlags(#)</code>	number of lags
<code>baseline(#)</code>	baseline period
<code>lagview</code>	show lags instead of time units
<code>verbose</code>	display results of the underlying regression
<code>post</code>	post the results of the underlying regression in <code>e()</code>
<code>nocl</code>	do not plot confidence intervals
<code>noyline</code>	suppress <i>y</i> -axis reference line

CI plot	
<code>recastci(plotype)</code>	plot confidence intervals using <i>plotype</i>
<code>ciopts(rcap_options)</code>	affect rendition of confidence intervals
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
Add plots	
<code>addplot(plot)</code>	add other plots to the graph
Y axis, X axis, Titles, Legend, Overall	
<code>twoway_options</code>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>

<i>bdecomp_options</i>	Description
<i>summaryonly</i>	show decomposition summary only
<i>fulldecomponly</i>	show full decomposition only
<i>noheader</i>	suppress the table header in the output
<i>notable</i>	suppress the table in the output
<i>graph</i>	plot individual components against their weights
Plot	
<i>copts(marker_options)</i>	change look of markers for cohorts
<i>tnopts(marker_options)</i>	change look of markers for treated versus never treated
<i>laopts(marker_options)</i>	change look of markers for later versus always treated
<i>leopts(marker_options)</i>	change look of markers for later versus earlier treated
<i>elopts(marker_options)</i>	change look of markers for earlier versus later treated
<i>atetlineopts(cline_options)</i>	change look of the ATET reference line
<i>noatetline</i>	suppress ATET reference line
Y axis, X axis, Titles, Legend, Overall	
<i>twoway_options</i>	any options other than <i>by()</i> documented in [G-3] <i>twoway_options</i>

Options for estat trendplots

omeans, *omeans(plot_options)*, *ltrends*, and *ltrends(plot_options)* specify which graphs are to be included and how they should be individually rendered. The default is *omeans ltrends*, meaning that both graphs are included without any modifications.

omeans specifies that the observed-means graph be included. Specifying *omeans* suppresses the linear-trends model graph unless *ltrends* or *ltrends(plot_options)* is also specified.

omeans(plot_options) specifies that the observed-means graph be included and affects its rendition. Specifying *omeans(plot_options)* suppresses the linear-trends model graph unless *ltrends* or *ltrends(plot_options)* is also specified.

ltrends specifies that the linear-trends model graph be included. Specifying *ltrends* suppresses the observed-means graph unless *omeans* or *omeans(plot_options)* is also specified.

ltrends(plot_options) specifies that the linear-trends model graph be included and affects its rendition. Specifying *ltrends(plot_options)* suppresses the observed-means graph unless *omeans* or *omeans(plot_options)* is also specified.

notitle suppresses the overall title of the rendered graph.

noxline suppresses rendering of the treatment-time reference line.

nocommonlegend suppresses the display of one common legend and renders two individual legends.

legendfrom(#) specifies which legend to use; the default is *legendfrom(1)*, which refers to the legend of the first plot (observed means). *legendfrom(#)* is not allowed with the *nocommonlegend* option.

Plot

cline_options affect the rendition of the plotted trend lines, including their style, size, and color; see [G-3] [cline_options](#).

line1opts(cline_options) affect the rendition of the plotted trend lines for the group of controls, including their style, size, and color; see [G-3] [cline_options](#).

line2opts(cline_options) affect the rendition of the plotted trend lines for the group of treated, including their style, size, and color; see [G-3] [cline_options](#).

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] [twoway_options](#), excluding *by()*. These include options for titling the graph (see [G-3] [title_options](#)) and for saving the graph to disk (see [G-3] [saving_option](#)).

Options for estat grangerplot

nleads(#) specifies the number of leads to be included in the model and plotted. By default, all available leads are included. The number of leads must be greater than 0.

nlags(#) specifies the number of lags to be included in the model and plotted. By default, all available lags are included. The number of lags must be greater than or equal to 0.

baseline(#) specifies the baseline period for which the corresponding lead or lag is omitted. By default, the first lead is omitted, which corresponds to the time period prior to intervention.

lagview specifies to show the values of the *x* axis in terms of lags. If this option is not specified, time values are shown.

verbose specifies to display the output of the underlying regression model.

post posts the results of the underlying regression as estimation results in *e()*.

noci removes the pointwise confidence intervals. The default is to plot the confidence intervals.

noyline suppresses rendering of the reference line.

CI plot

recastci(plotype) specifies that confidence intervals be plotted using *plotype*. *plotype* may be *rarea*, *rbar*, *rspike*, *rcap*, *rcapsym*, *rline*, *rconnected*, or *rscatter*; see [G-2] [graph twoway](#). When *recastci()* is specified, the plot-rendition options appropriate to the specified *plotype* may be used in lieu of *rcap_options*. For details on those options, follow the appropriate link from [G-2] [graph twoway](#).

ciopts(rcap_options) affects the rendition of confidence intervals; see [G-3] [rcap_options](#).

level(#) specifies the confidence level, as a percentage, for confidence intervals. The default is *level(95)* or as set by *set level*; see [U] [20.8 Specifying the width of confidence intervals](#).

Add plots

addplot(plot) provides a way to add other plots to the generated graph; see [G-3] [addplot_option](#).

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] *twoway_options*, excluding `by()`. These include options for titling the graph (see [G-3] *title_options*) and for saving the graph to disk (see [G-3] *saving_option*).

Options for estat bdecomp

summaryonly displays the decomposition summary only.

fulldecomponly displays the results of the full decomposition only.

noheader suppresses the table header in the output.

notable suppresses the results table in the output.

graph specifies to plot the treatment-effect components against their weights.

Plot

copts(marker_options) affect the rendition of the markers for the plotted cohort coefficients, including their style, size, and color; see [G-3] *marker_options*.

tnopts(marker_options) affect the rendition of the markers for the plotted coefficients for treated versus never treated, including their style, size, and color; see [G-3] *marker_options*.

laopts(marker_options) affect the rendition of the markers for the plotted coefficients for later versus always treated, including their style, size, and color; see [G-3] *marker_options*.

leopts(marker_options) affect the rendition of the markers for the plotted coefficients for later versus earlier treated, including their style, size, and color; see [G-3] *marker_options*.

elopts(marker_options) affect the rendition of the markers for the plotted coefficients for earlier versus later treated, including their style, size, and color; see [G-3] *marker_options*.

atetlineopts(cline_options) affect the rendition of the plotted ATET reference line, including its style, size, and color; see [G-3] *cline_options*.

noatetline suppresses rendering of the ATET reference line.

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] *twoway_options*, excluding `by()`. These include options for titling the graph (see [G-3] *title_options*) and for saving the graph to disk (see [G-3] *saving_option*).

Remarks and examples

In the following, we discuss the treatment-effect decomposition approach implemented in `estat bdecomp`. To see examples of `estat trendplots`, `estat ptrends`, and `estat granger`, see examples 1, 4, 5, and 6 in [CAUSAL] [didregress](#). An example of `estat grangerplot` is shown in example 7 in [CAUSAL] [didregress](#).

A simple 2-by-2 DID design is characterized by two time periods (a period before and a period after a treatment or intervention occurs) and two groups (a treatment and a control group). Under this design, all treated units receive the treatment at the same time. A staggered DID design, on the other hand, involves multiple time points at which the treatment is administered. [Goodman-Bacon \(2021\)](#) has shown that, in this case, the ATET is a weighted average of all possible 2-by-2 DID estimators that compare treatment cohorts with each other. Decomposing an overall ATET into these 2-by-2 components provides a diagnostic tool to assess the validity of the two-way fixed-effects DID estimator implemented in `didregress` and `xtdidregress` when treatment timing varies.

Consider the case of a DID design with two treatment times so that the data consist of three groups: an earlier-treated group, a later-treated group, and a never-treated group. Under this scenario, there would be four 2-by-2 comparisons. Two of these come from comparing the group that was treated first with the never-treated group and comparing the group that was treated second with the never-treated group. The third compares the earlier-treated group with the later-treated group (before the later-treated group was treated). The fourth compares the later-treated group with the earlier-treated group (beginning at the point where the earlier-treated group received treatment). This fourth comparison is of particular concern if the 2-by-2 DID estimate substantially differs from the overall ATET estimate and if the corresponding weight is considerably large. In that case, this 2-by-2 estimate induces bias in the overall ATET estimate. The Bacon decomposition implemented in `estat bdecomp` offers insight into which 2-by-2 comparisons are driving the overall ATET estimate. With this decomposition, we can identify sources of variation that potentially bias the overall effect estimate. In cases of suspected bias, users may consider using [hdidregress](#) or [xthdidregress](#) to account for treatment-effect heterogeneity.

The decomposition requires that the data be strongly balanced with respect to the observed times per unit. Specifically, units defined by the group variable that is specified in the `group()` option of `didregress` or `xtdidregress` must have an equal number of observations, and observation times given by the variable specified in the `time()` option of `didregress` or `xtdidregress` must be the same across all units.

For further theoretical and empirical details, see [Goodman-Bacon \(2021\)](#). A summary of the Bacon treatment-effect decomposition is provided in [Baker, Larcker, and Wang \(2022\)](#). Applications of the Bacon decomposition can be found in [Rabideau et al. 2021](#) and [Strasser, Oggenfuss, and Wolter \(2022\)](#).

➤ Example 1: DID treatment-effect decomposition for designs with variation in treatment timing

We use a (fictional) dataset where the units are dog breeds for which registration numbers at the American Kennel Club are observed over the course of 10 years. We wish to find out whether the registration numbers change in response to a breed being featured in a widely shown movie. Here is an excerpt of the data:

```
. use https://www.stata-press.com/data/r19/akc
(Fictional dog breed and AKC registration data)
. list in 981/1000, sepby(breed) abbreviate(10) noobs
```

year	breed	movie	best	registered
2031	Old English Sheepdog	0	0	1324
2032	Old English Sheepdog	0	0	878
2033	Old English Sheepdog	0	0	699
2034	Old English Sheepdog	0	0	645
2035	Old English Sheepdog	0	0	1762
2036	Old English Sheepdog	0	0	1182
2037	Old English Sheepdog	1	0	1816
2038	Old English Sheepdog	1	0	4171
2039	Old English Sheepdog	1	0	3755
2040	Old English Sheepdog	1	0	4054
2031	Otterhound	0	0	692
2032	Otterhound	0	0	598
2033	Otterhound	0	0	670
2034	Otterhound	0	0	1642
2035	Otterhound	0	0	1311
2036	Otterhound	0	0	572
2037	Otterhound	0	0	1167
2038	Otterhound	0	0	620
2039	Otterhound	0	0	511
2040	Otterhound	0	0	1065

We list the data for two breeds, the Old English Sheepdog and the Otterhound. We can see that the Old English Sheepdog is featured in a movie that is released in the year 2037 and, thus, is a treated unit. The Otterhound breed has no such luck and belongs to the group that is never treated. Our outcome variable registered measures the number of registrations at the American Kennel Club. We use xtdidregress to fit a two-way fixed-effects DID model:

```
. xtset breed
Panel variable: breed (balanced)
. xtdidregress (registered) (movie), group(breed) time(year)
Treatment and time information
Time variable: year
Control:      movie = 0
Treatment:    movie = 1
```

	Control	Treatment
Group		
breed	119	22
Time		
Minimum	2031	2034
Maximum	2031	2037

Difference-in-differences regressionNumber of obs = 1,410

Data type: Longitudinal

(Std. err. adjusted for 141 clusters in breed)

registered	Coefficient	Robust std. err.	t	P> t	[95% conf. interval]	
ATET						
movie (1 vs 0)	2129.655	78.16241	27.25	0.000	1975.124	2284.186

Note: ATET estimate adjusted for panel effects and time effects.
Note: Treatment occurs at different times.

We can see that our overall ATET estimate is around 2,130, which is to say that we expect the number of dog registrations to increase by 2,130 if a member of their breed is featured as a movie protagonist. We can now use `estat bdecomp` to decompose this number into its constituent 2-by-2 parts:

```
. estat bdecomp
DID treatment-effect decomposition
ATET = 2129.655
Number of obs = 1,410
Number of groups = 141
Number of cohorts = 4
```

ATET decomposition summary	ATET component	Weight
Treated vs never treated	2166.021	0.970810
Treated earlier vs later	936.70729	0.013157
Treated later vs earlier	906.60588	0.016033

Full ATET decomposition	2x2 coefficient	Weight
Treated vs never treated		
2034 vs never treated	1673.5688	0.157125
2036 vs never treated	1999.2515	0.140290
2037 vs never treated	2315.6702	0.673394
Treated earlier vs later		
2034 vs 2036	580.59722	0.001132
2034 vs 2037	836.17778	0.008488
2036 vs 2037	1291.9333	0.003537
Treated later vs earlier		
2036 vs 2034	542.825	0.001886
2037 vs 2034	831.70833	0.011318
2037 vs 2036	1448.7167	0.002829

Note: Number of cohorts includes never treated.
Note: The ATET reported by **xtdidregress** is a weighted average of the ATET components. If any component is substantially different from the ATET reported by **xtdidregress** and the weight is large, consider accounting for treatment-effect heterogeneity by using **xthdidregress**.

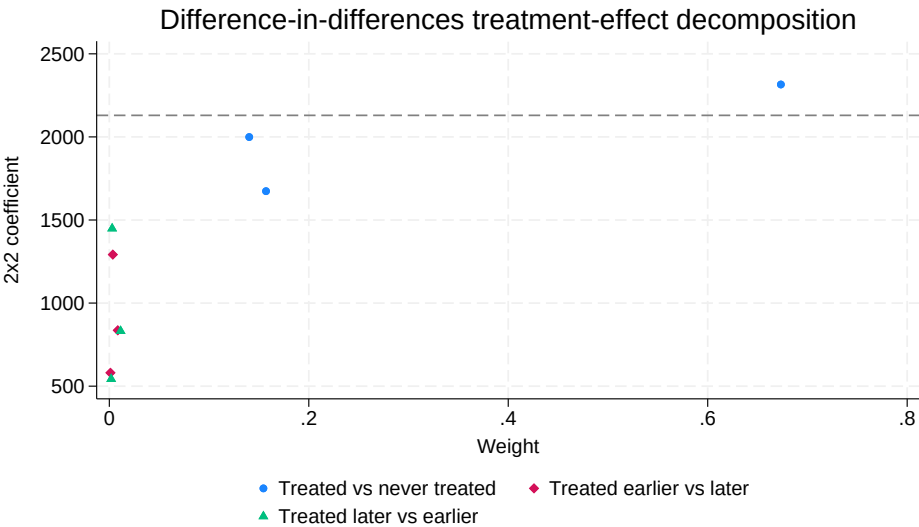
In the table header, we see that we have 1,410 observations from 141 groups (breeds) and that we have four cohorts. The four cohorts consist of groups of breeds that are treated at three different times (2034, 2036, and 2037) and a group that is never treated. The first table shows a decomposition summary where the estimated ATET is decomposed into components due to comparisons between treated and never treated, between earlier treated and later treated, and between later treated and earlier treated. The results in the second table decompose the summary results further into the individual 2-by-2 comparisons.

From the summary results, we see that the component due to comparisons between treated and never treated is close to the overall effect estimate and that its weight is large. Specifically, the weight is around 0.97, which indicates that 97% of the overall ATET estimate is due to comparisons between treated and never treated. The other two components are not close to the overall effect. This would be a potential concern, especially for the later- versus earlier-treated comparisons, if the corresponding weight was considerably large. However, the weights are small here. The later- versus earlier-treated and earlier-versus later-treated components together account for only 3% of the overall ATET.

In the full decomposition results, we see the three components from the summary broken down further. For example, if we look at the treated versus never-treated component, we see that around two thirds of that component is driven by the group that received treatment last. We also see that the 2-by-2 coefficients are increasing, which provides some indication that the treatment effect may be changing over time.

In this example, the number of treatment times is relatively small. However, if that number is large, it can be instructive to plot the 2-by-2 coefficients against their weights. We do this by using the `graph` option of `estat bdecomp`. Because we do not need to see the numerical results again, we also specify the `notable` and `noheader` options:

```
. estat bdecomp, graph notable noheader
```



The blue circles are the 2-by-2 coefficients from the treated versus never-treated component; the red diamonds are the 2-by-2 coefficients from the earlier- versus later-treated component; and the green triangles are the 2-by-2 coefficients from the later- versus earlier-treated component. The triangles and diamonds, while being substantially different from the overall ATET (indicated by the dashed reference line), are all very close to zero weight. This indicates that the contribution of these coefficients to the overall ATET is negligible.



Stored results

`estat ptrends` stores the following results for the test of linear trends in `r()`:

Scalars

<code>r(N)</code>	number of observations
<code>r(F)</code>	test statistic
<code>r(df_r)</code>	number of degrees of freedom of the residuals for the F distribution under H_0
<code>r(p)</code>	p -value
<code>r(df_m)</code>	number of degrees of freedom of the test for the F distribution under H_0

`estat granger` stores the following results for the test of treatment anticipation in `r()`:

Scalars

<code>r(N)</code>	number of observations
<code>r(F)</code>	test statistic
<code>r(df_r)</code>	number of degrees of freedom of the residuals for the F distribution under H_0
<code>r(p)</code>	p -value
<code>r(df_m)</code>	number of degrees of freedom of the test for the F distribution under H_0

`estat grangerplot`, when used with option `post`, stores results from the underlying regression model in `e()` and `r()`.

`estat bdecomp` stores the following results in `r()`:

Scalars

<code>r(N_groups)</code>	number of panels
<code>r(N_obs)</code>	number of observations
<code>r(N_cohorts)</code>	number of cohorts
<code>r(atet)</code>	overall ATET
<code>r(atet_between_b)</code>	between component
<code>r(atet_within_b)</code>	within component
<code>r(atet_between_w)</code>	weight of between component
<code>r(atet_within_w)</code>	weight of within component

Macros

<code>r(ttimes)</code>	times of treatment
------------------------	--------------------

Matrices

<code>r(atet_comp_b)</code>	ATET components
<code>r(atet_comp_w)</code>	weights of ATET components
<code>r(coefs)</code>	individual 2-by-2 coefficients
<code>r(wgts)</code>	weights of individual 2-by-2 coefficients

Methods and formulas

The tests performed with `estat ptrends` and `estat granger` are based on augmented difference-in-differences (DID) models. With `estat ptrends`, we augment the DID model with terms that capture the differences in slopes between treated and controls. With `estat granger`, we augment the model by interacting the dummy variable that marks treated observations with dummy variables for time periods prior to the treatment to capture any potential anticipatory treatment effects.

Let's consider the case of panel data for individuals over time, in which individuals belong to a group s . Groups could be states, occupational categories, districts, etc. Let y_{its} be the outcome of individual i , who belongs to group s , at time t , where $i = 1, \dots, N$, $t = 1, \dots, T$, and $s = 1, \dots, S$.

We can write the DID model for such setups as follows:

$$y_{ist} = \gamma_i + \gamma_t + \mathbf{x}_{ist}\beta + D_{st}\delta + \epsilon_{ist}$$

Here γ_i are individual fixed effects, γ_t are time fixed effects, $\mathbf{x}_{ist}$ are covariates, D_{st} is a variable that is 1 if an individual belongs to a group s that is treated at time t and is 0 otherwise, and ϵ_{ist} is an error term. The coefficient δ represents the average treatment effect on the treated (ATET).

To simplify the exposition below, we rewrite the model as follows:

$$\begin{aligned} y_{ist} &= \gamma_i + \gamma_t + \mathbf{x}_{ist}\beta + D_{st}\delta + \epsilon_{ist} \\ y_{ist} &= \text{DID}_{ist} + \epsilon_{ist} \end{aligned} \tag{1}$$

The linear-trends model that is used for the parallel-trends test with `estat ptrends` augments the above model with two more terms. Let $d_{t,0} = 1(d_t = 0)$ be a variable indicating pretreatment time periods, and let $d_{t,1} = 1(d_t = 1)$ be a variable indicating posttreatment time periods. Also, let w_i be a variable that is 1 if the individual belongs to a treated group and is 0 otherwise. The augmentation terms then consist of two 3-way interactions between $d_{t,0}$, w_i , and t , and $d_{t,1}$, w_i , and t :

$$y_{ist} = \text{DID}_{ist} + w_i d_{t,0} t \zeta_1 + w_i d_{t,1} t \zeta_2 + \epsilon_{ist} \tag{2}$$

Under this specification, the coefficient ζ_1 captures the differences in slopes between treatment group and control group in pretreatment periods, while ζ_2 captures the differences in slopes in posttreatment periods. If ζ_1 is 0, the linear trends in the outcome are parallel during pretreatment periods. The same is true for ζ_2 with respect to the posttreatment period; however, posttreatment differences in trends are not relevant for assessing the parallel-trends assumption. `estat ptrends` uses a Wald test of ζ_1 against 0 to assess whether the linear trends are parallel prior to treatment. Thus, the null hypothesis of this test is that the linear trends are parallel.

`estat granger` performs a Granger-type causality test to assess whether treatment effects are observed prior to the treatment. To illustrate this, suppose the treatment took place at time $t = j$. We could express D_{st} as $D_{st} = 1(t \geq j)w_i$. The Granger-type test augments the model with counterfactual treatment-time indicators. For example, if the treatment occurred at time $j - 1$, then we could construct a new treatment as $1(t_{it} \geq j - 1)w_i$, and if we have sufficient time points, we could construct another counterfactual treatment as $1(t_{it} \geq j - 2)w_i$, and so on. These terms are referred to as leads in the DID literature. The model used by `estat granger` uses the model in (1) and augments it with all leads leaving out one for identification purposes. Let J index the time at which the treatment occurs.

$$y_{ist} = \text{DID}_{ist} + \sum_{j=2}^{J-1} 1(t_{it} \geq j)w_i \lambda_j + \nu_{ist} \tag{3}$$

The test result is then obtained by performing a joint Wald test on the coefficients λ_j . Thus, the null hypothesis for this test is that the coefficients in λ_j are jointly 0, which is to say there are no anticipatory effects.

`estat grangerplot` fits a generalization of the DID model in (1) and plots the estimated coefficients from this model, including their 95% confidence intervals. The model is similar to (3), but this model parameterizes the leads differently and includes lags in addition to leads. Let I_s be the time of treatment, $m < 0$ be the number of time periods prior to I_s , $q \geq 0$ be the number of periods after I_s , and b be the baseline period. The model is

$$y_{ist} = \gamma_i + \gamma_t + \mathbf{x}_{ist}\boldsymbol{\beta} + \sum_{k=m, k \neq b}^q B_{st}^k w_i \lambda_k + \epsilon_{ist}$$

where

$$B_{st}^k = \begin{cases} 1(t_{it} \leq I_s + k), & \text{if } k = m \\ 1(t_{it} = I_s + k), & \text{if } m < k < q \\ 1(t_{it} \geq I_s + k), & \text{if } k = q \end{cases}$$

This yields a model with $|m|$ leads and q lags. By default, `estat grangerplot` uses all available leads and lags. If, without loss of generality, we set the base to the period prior to treatment, $b = -1$ (the default), then with $t = 1, \dots, T$ and $I_s = J$, a maximum of $n_{\text{leads}} = J - 2$ leads and $n_{\text{lags}} = T - J + 1$ lags is available. Notice that, if all available leads and lags are used, B_{st}^k reduces to $B_{st}^k = 1(t_{it} = I_s + k)$ because there are no periods before or after $I_s + k$. With fewer than available leads and lags, that is, $|m| < n_{\text{leads}}$ or $q < n_{\text{lags}}$, notice that the corresponding indicator variables capture the periods beyond the endpoints that correspond to m and q .

At a minimum, the model has to include a single lead. In that case, we have that $m = -1$ and $q = 0$. After omitting the base, we have that $k = 0$ and B_{st}^k reduces to $B_{st}^k = 1(t_{it} \geq I_s)$. Notice that $B_{st}^k w_i$ now yields our original treatment indicator D_{st} . In other words, the model with a single lead and no lags yields the special case of the DID model in (1). Notice also that the model in (3) is a special case, too. It is equivalent to the model fit by `estat grangerplot` with all available leads and no lags. However, (3) uses a different parameterization because the indicator variables are constructed differently. `estat grangerplot` plots the coefficients λ_k against the corresponding time periods.

The `estat trendplots` command produces two plots. The first plot is simply plotting the observed means for each treatment group at each point in time. The second plot is based on the model in (2), which is the model used for the parallel-trends test, but this model centers the continuous time variable around its minimum value:

$$y_{ist} = \text{DID}_{ist} + w_i d_{t,0} \{t - \min(t)\} \zeta_1 + w_i d_{t,1} \{t - \min(t)\} \zeta_2 + \mu_{ist}$$

Centering around the minimum time value provides a common reference point at the first observed time point such that deviations from parallelism are easily detectable. The graph then shows the predicted values from this model, evaluated at all observed time points for each of the treatment groups and at the means of the covariates.

While the formulas above are shown for the case of panel data, these methods work the same way for data that consist of repeated cross-sections.

The `estat bdecomp` command may be used in cases where units get treatment at different times. The command performs a decomposition of the estimated DID treatment effect into components that come from all possible 2-by-2 cohort comparisons. Cohorts are defined by the times of treatment administration and also include a group of never treated and potentially a group of always treated. Cohorts are also referred to as timing groups. `estat bdecomp` results are based on the derivation in [Goodman-Bacon \(2021\)](#).

For a DID model without covariates, we have the two-way fixed-effects model

$$y_{it} = \gamma_i + \gamma_t + D_{it}\delta^{\text{DD}} + \epsilon_{it}$$

where δ^{DD} is the overall treatment effect from the DID model. Let t_i denote the time when unit i receives treatment. If unit i was never treated, we set $t_i = U > T$ and say that the unit belongs to cohort U (where T is the latest time period for which we have data). If unit i was treated in period $k \in \{1, 2, \dots, T\}$, that is, $t_i = k$, we say that the unit belongs to cohort k . Let K denote the set of all cohorts (including U).

[Goodman-Bacon \(2021\)](#) show that we can decompose the estimated $\hat{\delta}^{\text{DD}}$ into the weighted sum of the DID coefficients that comes from all possible 2-by-2 cohort pairs,

$$\hat{\delta}^{\text{DD}} = \sum_{k \neq U} w_{kU} \hat{\delta}_{kU}^{2 \times 2} + \sum_{k \neq U} \sum_{l > k} \left[w_{kl}^k \hat{\delta}_{kl}^{2 \times 2, k} + w_{kl}^l \hat{\delta}_{kl}^{2 \times 2, l} \right]$$

where $\hat{\delta}_{kU}^{2 \times 2}$ is the coefficient from the 2-by-2 DID model that uses cohorts k and U as treatment and control, respectively, and $\hat{\delta}_{kl}^{2 \times 2, k}$ is the coefficient from the model that uses cohorts k and l as treatment and control, respectively. The weights w are given by

$$\begin{aligned} w_{kU} &= \frac{(s_k + s_U)^2 s_{kU} (1 - s_{kU}) \overline{D}_k (1 - \overline{D}_k)}{\hat{V}^D} \\ w_{kl}^k &= \frac{[(s_k + s_l)(1 - \overline{D}_l)]^2 s_{kl} (1 - s_{kl}) \frac{\overline{D}_k - \overline{D}_l}{1 - \overline{D}_l} \frac{1 - \overline{D}_k}{1 - \overline{D}_l}}{\hat{V}^D} \\ w_{kl}^l &= \frac{[(s_k + s_l) \overline{D}_k]^2 s_{kl} (1 - s_{kl}) \frac{\overline{D}_l}{\overline{D}_k} \frac{\overline{D}_k - \overline{D}_l}{\overline{D}_k}}{\hat{V}^D} \end{aligned}$$

where $s_j \equiv \sum_i 1(t_i = j)/N$ is the share of cohort $j \in K$, $s_{ab} \equiv s_a/(s_a + s_b)$ is the relative size of cohort a with respect to cohort b , $\overline{D}_j \equiv \sum_t 1(t \geq j)/T$ is the proportion of time spent in treatment by cohort $j \in K$, and $\hat{V}^D$ is the sample variance of the treatment indicator demeaned with respect to both group and time fixed effects. The weights w sum to unity: $\sum_{k \neq U} w_{kU} + \sum_{k \neq U} \sum_{l > k} [w_{kl}^k + w_{kl}^l] = 1$.

For a model that includes time-varying covariates, $\mathbf{x}_{it}$, a similar decomposition result holds. Consider the model

$$y_{it} = \gamma_i + \gamma_t + \mathbf{x}_{it}\beta + D_{it}\delta^{\text{DD}|\mathbf{x}} + \epsilon_{it}$$

where $\delta^{\text{DD}|\mathbf{x}}$ denotes the covariate-adjusted ATET. In this case, we can decompose $\hat{\delta}^{\text{DD}|\mathbf{x}}$ into a within and a between component,

$$\hat{\delta}^{\text{DD}|\mathbf{x}} = \Omega \hat{\delta}_w^p + (1 - \Omega) \hat{\delta}_b^d$$

To estimate the within and between components, we first fit a linear model that partials the covariates out of D_{it} ; that is,

$$\widetilde{D}_{it} = \widetilde{\mathbf{x}}_{it}\widehat{\Gamma} + \widetilde{d}_{it} = \widetilde{p}_{it} + \widetilde{d}_{it} \quad (4)$$

where $\widetilde{D}_{it}$ and $\widetilde{\mathbf{x}}_{it}$ are respectively the treatment indicator and the covariates demeaned with respect to both group and time fixed effects. The resulting residuals can then be separated into a within component, $\widetilde{d}_{i(k)t}$, and a between component, $\widetilde{d}_{kt}$, so that $\widetilde{d}_{it} = \widetilde{d}_{i(k)t} + \widetilde{d}_{kt}$.

The within component of $\widetilde{d}_{it}$ is defined as $\widetilde{d}_{i(k)t} = (d_{it} - \bar{d}_i) - (\bar{d}_{kt} - \bar{d}_k)$, where $d_{it} = D_{it} - \widehat{\Gamma}\mathbf{x}_{it}$ is the treatment status adjusted by covariates only; and $\bar{d}_i$, $\bar{d}_k$, and $\bar{d}_{kt}$ are respectively the averages of d_{it} by individual, cohort, and cohort by year. This component captures deviations of d_{it} from cohort averages and cohort-by-year averages.

The between component of $\widetilde{d}_{it}$ is defined as $\widetilde{d}_{kt} = (\bar{d}_{kt} - \bar{d}_k) - (\bar{d}_t - \bar{\bar{d}})$, where $\bar{d}_t$ and $\bar{\bar{d}}$ are the averages of d_{it} by year and in the whole sample, respectively.

The within and between components of the DID treatment effect, $\hat{\delta}_w^p$ and $\hat{\delta}_b^d$, can be obtained by separately regressing the outcome variable y_{it} on $\widetilde{d}_{i(k)t}$ and $\widetilde{d}_{kt}$, respectively. Thus, $y_{it} = a_w + \widetilde{d}_{i(k)t}\hat{\delta}_w^p + \nu_{it}$ and $y_{it} = a_b + \widetilde{d}_{kt}\hat{\delta}_b^d + \eta_{it}$. The weight Ω is given by the ratio of the variances $\Omega = \widehat{V}_w^d / \widehat{V}^d$, where $\widehat{V}_w^d$ and $\widehat{V}^d$ are the sample variances of $\widetilde{d}_{i(k)t}$ and $\widetilde{d}_{it}$, respectively.

The between component $\hat{\delta}_b^d$ can be decomposed further into 2-by-2 components

$$\hat{\delta}_b^d = \sum_k \sum_{l>k} w_{kl}^{b|\mathbf{x}} \hat{\delta}_{kl}^{2 \times 2|d}$$

where

$$\hat{\delta}_{kl}^{2 \times 2|d} = \frac{\widehat{V}_{kl}^D \hat{\delta}_{kl}^{2 \times 2} - \widehat{V}_{b,kl}^p \hat{\delta}_{b,kl}^p}{\widehat{V}_{b,kl}^d}$$

and

$$w_{kl}^{b|\mathbf{x}} = (s_k + s_l)^2 \frac{\widehat{V}_{b,kl}^d}{\widehat{V}_b^d}$$

In the formulas above, $\widehat{V}_b^d$ and $\widehat{V}_{b,kl}^d$ are the sample variances of $\widetilde{d}_{kt}$ in the whole sample and in the subsample consisting of cohorts k and l , respectively. $\widehat{V}_{kl}^D$ and $\widehat{V}_{b,kl}^p$ are the sample variances of $\widetilde{D}_{it}$ and $\widetilde{p}_{it}$, respectively, in the subsample consisting of cohorts k and l . Finally, $\hat{\delta}_{kl}^{2 \times 2}$ and $\hat{\delta}_{b,kl}^p$ are the coefficients from a regression of y_{it} on $\widetilde{D}_{it}$ and from a regression of y_{it} on $\widetilde{p}_{it}$, respectively, in the subsample consisting of cohorts k and l .

Thus, the full decomposition for a DID specification with controls is

$$\hat{\delta}^{\text{DD}|\mathbf{x}} = \Omega \hat{\delta}_w^p + (1 - \Omega) \sum_k \sum_{l>k} w_{kl}^{b|\mathbf{x}} \hat{\delta}_{kl}^{2 \times 2|d}$$

References

- Baker, A. C., D. F. Larcker, and C. C. Y. Wang. 2022. How much should we trust staggered difference-in-differences estimates? *Journal of Financial Economics* 144: 370–395. <https://doi.org/10.1016/j.jfineco.2022.01.004>.
- Goodman-Bacon, A. 2021. Difference-in-differences with variation in treatment timing. *Journal of Econometrics* 225: 254–277. <https://doi.org/10.1016/j.jeconom.2021.03.014>.
- Rabideau, B., M. D. Eisenberg, R. Reid, and N. Sood. 2021. Effects of employer-offered high-deductible plans on low-value spending in the privately insured population. *Journal of Health Economics* 76: 102424. <https://doi.org/10.1016/j.jhealeco.2021.102424>.
- Strasser, M., C. Oggenfuss, and S. C. Wolter. 2022. Much ado about nothing? School curriculum reforms and students' educational trajectories. CESifo Working Paper 9912, CESifo Group, Munich, Germany. https://www.cesifo.org/DocDL/cesifo1_wp9912.pdf.

Also see

- [CAUSAL] **didregress** — Difference-in-differences estimation⁺
- [CAUSAL] **DID intro** — Introduction to difference-in-differences estimation
- [U] **20 Estimation and postestimation commands**

[Description](#)

[Options](#)

[Acknowledgment](#)

[Quick start](#)

[Remarks and examples](#)

[References](#)

[Menu](#)

[Stored results](#)

[Also see](#)

[Syntax](#)

[Methods and formulas](#)

Description

`eteffects` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational data when treatment assignment is correlated with the potential outcomes. It allows for continuous, binary, count, fractional, and non-negative outcomes and requires a binary treatment. To control for the endogeneity of the treatment assignment, the estimator includes residuals from the treatment model in the models for the potential outcomes, known as a control-function approach.

Quick start

ATE of binary treatment `treat` using a linear model for outcome `y1` on `x` and the residuals from a probit model for `treat` on `x` and `z`

```
eteffects (y1 x) (treat x z)
```

Same as above, but estimate ATET

```
eteffects (y1 x) (treat x z), atet
```

Same as above, but estimate POMs

```
eteffects (y1 x) (treat x z), pomeans
```

Same as above, and show parameters from auxiliary equations

```
eteffects (y1 x) (treat x z), pomeans aequations
```

ATE of `treat` using an exponential-mean model for `y1`

```
eteffects (y1 x, exponential) (treat x z)
```

Same as above, but for count outcome `y2`

```
eteffects (y2 x, exponential) (treat x z)
```

Same as above, but use a probit model for binary outcome `y3`

```
eteffects (y3 x, probit) (treat x z)
```

Same as above, but use a fractional probit model for `y4` ranging from 0 to 1

```
eteffects (y4 x, fractional) (treat x z)
```

Menu

- Statistics > Causal inference/treatment effects > Endogenous treatment > Control function estimator > Continuous outcomes
- Statistics > Causal inference/treatment effects > Endogenous treatment > Control function estimator > Binary outcomes
- Statistics > Causal inference/treatment effects > Endogenous treatment > Control function estimator > Count outcomes
- Statistics > Causal inference/treatment effects > Endogenous treatment > Control function estimator > Fractional outcomes
- Statistics > Causal inference/treatment effects > Endogenous treatment > Control function estimator > Nonnegative outcomes

Syntax

```
eteffects (ovar omvarlist [ , omodel noconstant ])  
          (tvar tmvarlist [ , noconstant ]) [ if ] [ in ] [ weight ] [ , stat options ]
```

ovar is the *depvar* of the outcome model.
omvarlist is the list of exogenous *indepvars* in the outcome model.
tvar is the binary treatment variable.
tmvarlist is the list of covariates that predict treatment assignment.

<i>omodel</i>	Description
Model	
linear	linear outcome model; the default
fractional	fractional probit outcome model
probit	probit outcome model
exponential	exponential-mean outcome model

<i>stat</i>	Description
Model	
ate	estimate average treatment effect in population; the default
atet	estimate average treatment effect on the treated
<u>pomeans</u>	estimate potential-outcome means

options	Description
Model	
<code>noconstant</code>	suppress constant term
SE/Robust	
<code>vce(vcetype)</code>	<i>vcetype</i> may be <code>robust</code> , <code>cluster clustvar</code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
Advanced	
<code>pstolerance(#)</code>	set tolerance for overlap assumption
<code>osample(newvar)</code>	generate <i>newvar</i> to mark observations that violate the overlap assumption
<code>coeflegend</code>	display legend instead of statistics

omvarlist and *tmvarlist* may contain factor variables; see [U] 11.4.3 Factor variables.

bayesboot, *bootstrap*, *by*, *collect*, *jackknife*, and *statsby* are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the *bootstrap* prefix; see [R] *bootstrap*.

fweights, *iweights*, and *pweights* are allowed; see [U] 11.1.6 *weight*.

coeflegend does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

Model
<code>noconstant</code> ; see [R] Estimation options.
<i>stat</i> is one of three statistics: <i>ate</i> , <i>atet</i> , or <i>pomeans</i> . <i>ate</i> is the default.
<i>ate</i> specifies that the average treatment effect be estimated.
<i>atet</i> specifies that the average treatment effect on the treated be estimated.
<i>pomeans</i> specifies that the potential-outcome means for each treatment level be estimated.
SE/Robust
<code>vce(vcetype)</code> specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (<code>robust</code>), that allow for intragroup correlation (<code>cluster clustvar</code>), and that use bootstrap or jackknife methods (<code>bootstrap</code> , <code>jackknife</code>); see [R] <i>vce_option</i> .
Reporting
<code>level(#)</code> ; see [R] Estimation options.

`aequations` specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no1stretch`; see [R] [Estimation options](#).

Maximization

`maximize_options`: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

`init_specs` is one of

`matname [ , skip copy]`

`# [ , # ...], copy`

Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `eteffects` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption.

The following option is available with `eteffects` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

If you are unfamiliar with treatment-effects estimators for observational data or the `teffects` commands, we recommend that you look at [\[CAUSAL\] teffects intro](#). For the intuition behind some of the concepts discussed below, we recommend that you read *Defining treatment effects* in [\[CAUSAL\] teffects intro advanced](#).

The estimators implemented in `eteffects` extend the regression adjustment (RA) estimators implemented in `teffects ra` to allow for endogenous treatments, that is, when treatment assignment is not independent of outcomes. This endogeneity is a violation of the conditional mean independence assumption used by `teffects ra`, as discussed in *The potential-outcome model* in [\[CAUSAL\] teffects intro advanced](#).

`eteffects` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs). It uses a linear, a probit, a fractional probit, or an exponential-mean model for the potential outcomes and a probit model for treatment assignment. After conditioning on the observable covariates, `eteffects` allows some remaining unobservable components to affect both treatment assignment and the potential outcomes. The treatment assignment process is endogenous because these unobservable components affect both treatment assignment and the potential outcomes.

To control for the endogeneity of the treatment assignment, `eteffects` uses a control-function approach. This method controls for endogeneity by including the residuals from the treatment-assignment model as a regressor in the models for the potential outcome. The implementation in `eteffects` follows Wooldridge (2010), who provides an excellent discussion of the control-function approach that addresses endogeneity problems in a treatment-effects context.

The control-function approach estimates the parameters of the conditional means of the potential outcomes. Sample averages of the conditional means are used to estimate the unconditional ATE, ATET, or POMs. This method is known as RA.

Taken collectively, the estimators implemented in `eteffects` are control-function RA estimators. See *Methods and formulas* below for details about the estimation procedure.

▷ Example 1: Linear outcome estimates for ATE

Suppose we want to know the effect of a mother smoking while pregnant on the birthweight of her infant. We use an extract from Cattaneo (2010) in which `bweight` records the baby's birthweight and `mbsmoke` is the variable (0 or 1) indicating whether a mother smoked while pregnant.

We may believe that birthweight (the potential outcome) is influenced by whether the mother had a prenatal exam in the first trimester, whether the mother is married, the mother's age, whether this is the first birth, and the education level of the father. We may also believe that the smoking decision (the treatment) is influenced by the mother's marital status, the education level of the mother, her age, whether she had a prenatal exam in the first trimester, and whether this baby is her first baby.

Thus we condition on different sets of covariates in the models for treatment assignment and the potential outcomes. In the probit model for smoking status (`mbsmoke`), we condition on marital status (`mmarried`), age (`mage`), mother's education level (`medu`), father's education level (`fedu`), and whether it was the mother's first baby (`fbaby`). We model birthweight (`bweight`) as a linear function of whether the mother had a first-trimester prenatal exam (`prenatal1`), `mmarried`, `mage`, and `fbaby`. We can estimate the ATE of smoking status using one of the `teffects` estimators if we believe that there are no unobservable components that affect both the decision to smoke while pregnant and the potential birthweights.

If we believe there is some unobservable factor that affects both assignment to treatment and the potential outcome, we must select another estimator. For example, we do not observe a mother's health consciousness, which affects both the smoking decision and each potential birthweight through other behaviors such as intake of prenatal vitamins. Under these assumptions, the estimators in `eteffects` consistently estimate the ATE, but the estimators in `[CAUSAL] teffects` yield inconsistent estimates.


```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. eteffects (bweight i.prenatal1 i.mmarried mage i.fbaby)
> (mbsmoke i.mmarried mage i.fbaby medu fedu)

Iteration 0: EE criterion = 4.739e-24
Iteration 1: EE criterion = 2.524e-25

Endogenous treatment-effects estimation      Number of obs = 4,642
Outcome model: linear
Treatment model: probit
```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke (Smoker vs Nonsmoker)	-455.9119	212.4393	-2.15	0.032	-872.2853	-39.53852
POmean						
mbsmoke Nonsmoker	3437.964	31.21145	110.15	0.000	3376.791	3499.138

When no mother smokes, the average birthweight is 3,438 grams. The average birthweight is 456 grams less when all mothers smoke than when no mother smokes.

We can compare these results with those obtained if we ignore the endogeneity of the smoking decision. Below we estimate the ATE using the inverse-probability-weighted regression-adjustment estimator in [\[CAUSAL\] teffects ipwra](#).

```
. teffects ipwra (bweight i.prenatal1 i.mmarried mage i.fbaby)
> (mbsmoke i.mmarried mage i.fbaby medu fedu)

Iteration 0: EE criterion = 3.036e-22
Iteration 1: EE criterion = 3.755e-26

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : IPW regression adjustment
Outcome model  : linear
Treatment model: logit
```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke (Smoker vs Nonsmoker)	-233.6835	25.07695	-9.32	0.000	-282.8335	-184.5336
POmean						
mbsmoke Nonsmoker	3403.191	9.529709	357.11	0.000	3384.513	3421.869

In magnitude, the estimated ATE is more than half the estimate that allows for endogenous treatment assignment. If there is endogeneity, disregarding it underestimates the effect of smoking on birthweight. We show how to test for endogeneity in [example 1](#) of [\[CAUSAL\] eteffects postestimation](#).

➤ Example 2: Estimating the ATET

Continuing [example 1](#), we can use the `atet` option to estimate the ATET.

```
. eteffects (bweight i.prenatal1 i.mmarrried mage i.fbaby)
> (mbsmoke i.mmarrried mage i.fbaby medu fedu), atet

Iteration 0:  EE criterion = 4.659e-24
Iteration 1:  EE criterion = 1.479e-25

Endogenous treatment-effects estimation      Number of obs = 4,642
Outcome model: linear
Treatment model: probit
```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATET mbsmoke (Smoker vs Nonsmoker)	-409.8527	161.4816	-2.54	0.011	-726.3507	-93.35466
POmean mbsmoke Nonsmoker	3547.512	160.0595	22.16	0.000	3233.801	3861.223

In the population of mothers who smoke, the average infant birthweight would be 3,548 grams if none of these mothers smoked. For the mothers who smoke, the average infant birthweight is 410 grams less than if none of these mothers smoked.



► Example 3: Exponential-mean outcomes

We estimate the ATE of living in an urban area on monthly earnings (wage), using a subset of the National Longitudinal Survey in 1980 found in [Wooldridge \(2010\)](#). We assume that once we condition on work experience (exper), whether education level attained is college or higher (college), and IQ (iq), individual wages follow an exponential mean. The variables used to predict residence in an urban area (urban) are college and whether the respondent's father attained a bachelor's degree or higher (fcollege).

```
. use https://www.stata-press.com/data/r19/nlsy80
. eteffects (wage exper iq i.college, exponential nocons)
> (urban i.college fcollege)

Iteration 0: EE criterion = 2.903e-25
Iteration 1: EE criterion = 2.903e-25 (backed up)

Endogenous treatment-effects estimation          Number of obs = 935
Outcome model: exponential
Treatment model: probit
```

		Robust		z	P> z	[95% conf. interval]	
wage		Coefficient	std. err.				
ATE							
	urban						
	(1 vs 0)	481.0465	31.74882	15.15	0.000	418.82	543.2731
P0mean							
	urban						
	0	233.8083	13.51028	17.31	0.000	207.3286	260.288

When everyone lives outside urban areas, wages are \$234 a month on average. Wages are \$481 a month greater, on average, when everyone lives in urban areas.



Stored results

`eteffects` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>eteffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(omodel)</code>	fractional, linear, probit, or exponential
<code>e(stat)</code>	statistic estimated, <code>ate</code> , <code>atet</code> , or <code>pomeans</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vce</i> type specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

The treatment-effects models considered in `eteffects` are given by

$$y_{i0} = E(y_{i0}|\mathbf{x}_i) + \epsilon_{i0} \quad (1)$$

$$y_{i1} = E(y_{i1}|\mathbf{x}_i) + \epsilon_{i1} \quad (2)$$

$$t_i = E(t_i|\mathbf{z}_i) + \nu_i \quad (3)$$

$$y_i = t_i y_{i1} + (1 - t_i) y_{i0} \quad (4)$$

$$E(\epsilon_{ij}|\mathbf{x}_i, \mathbf{z}_i) = E(\epsilon_{ij}|\mathbf{z}_i) = E(\epsilon_{ij}|\mathbf{x}_i) = 0 \quad \text{for } j \in \{0, 1\} \quad (5)$$

$$E(\epsilon_{ij}|t) \neq 0 \quad \text{for } j \in \{0, 1\} \quad (6)$$

where the subscript i denotes individual level observations, y_{i1} is the potential outcome of receiving the treatment, y_{i0} is the potential outcome when the treatment is not received, t_i is the observed binary treatment, and y_i is the observed outcome. Each one of the potential outcomes is determined by its expected value conditional on a set of regressors $\mathbf{x}_i$ and an unobserved random component ϵ_{ij} , for $j \in \{0, 1\}$. Similarly, the treatment is given by its expectation conditional on a set of regressors $\mathbf{z}_i$, which does not need to differ from $\mathbf{x}_i$, and an unobserved component ν_i .

Equations (1)–(5) describe the parametric treatment-effects models in [\[CAUSAL\] teffects](#). Equation (6) adds endogeneity to the framework. It states that the unobservables in the potential-outcome equations are correlated to treatment status. For our birthweight example, this would happen if mothers who do not smoke are more health conscious than those who smoke and if we do not observe health awareness in our data. If we do not observe health awareness, the decision to smoke or not to smoke is not independent of the infant's birthweight.

Equations (3), (5), and (6) are the basis of the control-function estimator implemented by `eteffects`. Equation (5) states that the unobserved components in the potential outcome are independent of $\mathbf{z}_i$. Therefore, the correlation between t_i and the unobserved components must be equivalent to the correlation between ϵ_{ij} and ν_i . Another way of stating this is

$$\text{from (3)} \quad E(\epsilon_{ij}|t_i) = E(\epsilon_{ij}|E(t|\mathbf{z}_i) + \nu_i)$$

$$\begin{aligned} \text{from (5)} \quad &= E(\epsilon_{ij}|\nu_i) \\ &= \nu_i \beta_{2j} \end{aligned}$$

We fit (3) using a probit estimator. We then obtain $\hat{\nu}_i$ as the difference between the treatment and our estimate of $E(t_i|\mathbf{z}_i)$ and use this statistic to compute an estimate of $E(y_{ij}|\mathbf{x}_i, \nu_i, t_i)$ for $j \in \{0, 1\}$. If the outcome is linear, for instance,

$$E(y_{ij}|\mathbf{x}_i, \nu_i, t_i = j) = \mathbf{x}'_i \beta_{1j} + \nu_i \beta_{2j} \quad \text{for } j \in \{0, 1\} \quad (7)$$

For the probit and exponential-mean cases, respectively, we have the following:

$$E(y_{ij}|\mathbf{x}_i, \nu_i, t_i = j) = \Phi(\mathbf{x}'_i \beta_{1j} + \nu_i \beta_{2j}) \quad (8)$$

$$E(y_{ij}|\mathbf{x}_i, \nu_i, t_i = j) = \exp(\mathbf{x}'_i \beta_{1j} + \nu_i \beta_{2j}) \quad (9)$$

The parameters of (3) and (7)–(9), and the ATE, ATET, and POMs are estimated using the generalized method of moments (GMM). The moment equations used in GMM are the sample analogs of $E\{\mathbf{w}'_i \epsilon_i(\theta)\} = 0$, where $\mathbf{w}_i$ are the instruments, $\epsilon_i(\theta)$ are residuals, and θ are the parameters of the model (see [R] gmm). The moment conditions in the GMM estimation for the linear model are given by

$$\frac{1}{n} \sum_{i=1}^n \mathbf{x}'_i (y_i - \mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j}) t_i = 0 \quad (10)$$

$$\frac{1}{n} \sum_{i=1}^n \mathbf{x}'_i (y_i - \mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j}) (1 - t_i) = 0 \quad (11)$$

$$\frac{1}{n} \sum_{i=1}^n \mathbf{z}'_i \left\{ t_i \frac{\phi(\mathbf{z}'_i \widehat{\pi})}{\Phi(\mathbf{z}'_i \widehat{\pi})} - (1 - t_i) \frac{\phi(\mathbf{z}'_i \widehat{\pi})}{1 - \Phi(\mathbf{z}'_i \widehat{\pi})} \right\} = 0 \quad (12)$$

$$\frac{1}{n} \sum_{i=1}^n \left\{ (\mathbf{x}'_i \widehat{\beta}_{10} + \widehat{\nu}_i \widehat{\beta}_{20}) - \widehat{\text{POM0}} \right\} = 0$$

$$\frac{1}{n} \sum_{i=1}^n \left\{ (\mathbf{x}'_i \widehat{\beta}_{11} + \widehat{\nu}_i \widehat{\beta}_{21}) - \widehat{\text{POM0}} - \widehat{\text{ATE}} \right\} = 0 \quad (13)$$

where $\widehat{\nu}_i = t_i - \Phi(\mathbf{z}'_i \widehat{\pi})$, n is the number of observations, and $\widehat{\beta}_{11}, \widehat{\beta}_{10}, \widehat{\beta}_{21}, \widehat{\beta}_{20}, \widehat{\pi}, \widehat{\text{ATE}}$, and $\widehat{\text{POM0}}$ are the parameters. If we want to estimate the ATET, we replace (13) with

$$\frac{1}{n} \sum_{i=1}^n \left\{ (\mathbf{x}'_i \widehat{\beta}_{11} + \widehat{\nu}_i \widehat{\beta}_{21}) \frac{n}{n_t} - \widehat{\text{POM0}} \frac{n}{n_t} - \widehat{\text{ATE}} \right\} = 0$$

and if we want to estimate the potential-outcome means, we replace (13) with

$$\frac{1}{n} \sum_{i=1}^n \left\{ (\mathbf{x}'_i \widehat{\beta}_{11} + \widehat{\nu}_i \widehat{\beta}_{21}) - \widehat{\text{POM1}} \right\} = 0 \quad (14)$$

where $\widehat{\text{ATE}}$ and $\widehat{\text{POM1}}$ are the parameters of the model, and n_t is the number of treated units.

For the exponential-mean outcome model, we replace $\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j}$ with $\exp(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})$ to obtain the residual equations in (10)–(14). For the probit outcome model, we replace (10) and (11) with the following:

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n t_i \mathbf{x}'_i \left\{ y_i \frac{\phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})}{\Phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})} - (1 - y_i) \frac{\phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})}{1 - \Phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})} \right\} &= 0 \\ \frac{1}{n} \sum_{i=1}^n (1 - t_i) \mathbf{x}'_i \left\{ y_i \frac{\phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})}{\Phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})} - (1 - y_i) \frac{\phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})}{1 - \Phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})} \right\} &= 0 \end{aligned}$$

For the remaining equations, $\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j}$ is replaced with $\Phi(\mathbf{x}'_i \widehat{\beta}_{1j} + \widehat{\nu}_i \widehat{\beta}_{2j})$. The fractional probit model uses the same moment conditions as the probit model.

Acknowledgment

We thank Jeffrey Wooldridge of Michigan State University for helpful conversations and advice about the literature and methods in this area.

References

- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cerulli, G. 2014. `ivtreatreg`: A command for fitting binary treatment models with heterogeneous response to treatment and unobservable selection. *Stata Journal* 14: 453–480.
- . 2022. *Econometric Evaluation of Socio-Economic Programs: Theory and Applications*. 2nd ed. Berlin: Springer.
- Drukker, D. M. 2016. A generalized regression-adjustment estimator for average treatment effects from panel data. *Stata Journal* 16: 826–836.
- Lee, M., and C. Han. 2024. Ordinary least squares and instrumental-variables estimators for any outcome and heterogeneity. *Stata Journal* 24: 72–92.
- Terza, J. V., A. Basu, and P. J. Rathouz. 2008. Two-stage residual inclusion estimation: Addressing endogeneity in health econometric modeling. *Journal of Health Economics* 27: 531–543. <https://doi.org/10.1016/j.jhealeco.2007.09.009>.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.
- . 2014. Quasi-maximum likelihood estimation and testing for nonlinear models with endogenous explanatory variables. *Journal of Econometrics* 182: 226–234. <https://doi.org/10.1016/j.jeconom.2014.04.020>.

Also see

- [CAUSAL] **eteffects postestimation** — Postestimation tools for eteffects
- [CAUSAL] **etregress** — Linear regression with endogenous treatment effects
- [CAUSAL] **teffects** — Treatment-effects estimation for observational data
- [R] **gmm** — Generalized method of moments estimation
- [R] **probit** — Probit regression
- [R] **regress** — Linear regression⁺
- [U] **20 Estimation and postestimation commands**

Postestimation commands

The following postestimation command is of special interest after `eteffects`:

Command	Description
<code>estat endogenous</code>	perform tests of endogeneity

The following standard postestimation commands are available after `eteffects`:

Command	Description
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>hausman</code>	Hausman’s specification test
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of parameters
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
<code>predict</code>	treatment effects, conditional means at treatment, propensity scores, etc.
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

predict

Description for predict

predict creates a new variable containing predictions such as treatment effects, conditional means, propensity scores, and linear predictions.

Menu for predict

Statistics > Postestimation

Syntax for predict

```
predict [type] { stub* | newvar | newvars } [if] [in] [ , statistic tlevel ]
```

```
predict [type] stub* [if] [in] , scores
```

statistic	Description
Main	
te	treatment effect; the default
cmean	conditional mean at treatment level
ps	propensity score
xb	linear prediction
psxb	linear prediction for propensity score
xbtotal	linear prediction, using residuals from treatment model

Specify one new variable with te; specify one or two new variables with cmean, ps, and xb.

Options for predict

Main

- te, the default, calculates the treatment effect.
- cmean calculates the conditional mean for the control group. To also obtain the conditional mean for the treatment group, specify two variables. If you want the conditional mean for only the treatment group, specify the tlevel option.
- ps calculates the probability of being in the control group. To also obtain the probability of being in the treatment group, specify two variables. If you want the probability of being in the treatment group only, specify the tlevel option.
- xb calculates the linear prediction for the control group. To also obtain the linear prediction for the treatment group, specify two variables. If you want the linear prediction for only the treatment group, specify the tlevel option.
- psxb calculates the linear prediction for the propensity score.
- xbtotal calculates the linear prediction for the control group, including the residuals from the treatment model as regressors. To also obtain the linear prediction for the treatment group, specify two variables. If you want the linear prediction, including the residuals from the treatment model as regressors, only for the treatment group, specify the tlevel option.

`tlevel` specifies that the statistic be calculated for the treatment group; the default is to calculate the statistic for the control group.

`scores` calculates the score variables. For `eteffects`, this is the same as the residuals in the moment conditions used by the generalized method of moments (see [R] [gmm](#)). For the average treatment effect, the average treatment effect on the treated, and the potential-outcome means, parameter-level scores are computed. For the auxiliary equations, equation-level scores are computed.

estat

Description for estat

`estat endogenous` performs a Wald test to determine whether the estimated correlations between the treatment-assignment and potential-outcome models are different from zero. The null hypothesis is that the correlations are jointly zero. Rejection of the null hypothesis suggests endogeneity.

Menu for estat

Statistics > Postestimation

Syntax for estat

`estat endogenous`

`collect` is allowed with `estat endogenous`; see [U] [11.1.10 Prefix commands](#).

Remarks and examples

► Example 1: Testing for endogeneity

In [example 3](#) of [CAUSAL] `eteffects`, endogeneity could arise if unobservable factors that determine wages are correlated with the decision to live in an urban area. If there is no endogeneity, we would prefer to use one of the `teffects` estimators because they will give us the more efficient standard errors. The control-function approach used by `eteffects` allows us to test for endogeneity.

The control-function approach estimates the correlation between the unobservables of the treatment-assignment and potential-outcome models. If there is no correlation between the unobservables, then there is no endogeneity. We test for correlation, and thus for endogeneity, by typing

```
. use https://www.stata-press.com/data/r19/nlsy80
. eteffects (wage exper iq i.college, exponential nocons)
> (urban i.college fcollege)
(output omitted)
. estat endogenous
Test of endogeneity
H0: Treatment and outcome unobservables are uncorrelated

      chi2( 2) = 275.36
Prob > chi2 = 0.0000
```

We reject the null hypothesis of no endogeneity. This suggests that unobservable factors that determine wages mediate the decision to live in an urban area.

◀

□ Technical note

The estimated correlations between the unobservables of the treatment-assignment and potential-outcome models are auxiliary parameters. They appear under the headings TEOM0 and TEOM1, which refer to treatment residuals (TE) for outcome model 0 (OM0) and outcome model 1 (OM1), when the option `aequations` is specified.

For the model in [example 3](#) of [\[CAUSAL\] eteffects](#) with the `aequations` option, the results are the following:

```
. eteffects (wage exper iq i.college, exponential nocons)
> (urban i.college fcollege), aequations
Iteration 0: EE criterion = 2.903e-25
Iteration 1: EE criterion = 2.903e-25 (backed up)
Endogenous treatment-effects estimation          Number of obs = 935
Outcome model: exponential
Treatment model: probit
```

wage	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
urban (1 vs 0)	481.0465	31.74882	15.15	0.000	418.82	543.2731
POMean						
urban 0	233.8083	13.51028	17.31	0.000	207.3286	260.288
TME1						
college 1	.195811	.1012119	1.93	0.053	-.0025607	.3941827
fcollege	.1069748	.0992075	1.08	0.281	-.0874683	.3014179
_cons	.498012	.056408	8.83	0.000	.3874543	.6085698
OME0						
exper	.0193244	.0085633	2.26	0.024	.0025405	.0361082
iq	.0099473	.0036949	2.69	0.007	.0027053	.0171892
college 1	-.3718598	.2678636	-1.39	0.165	-.8968629	.1531433
OME1						
exper	.0238566	.017597	1.36	0.175	-.0106329	.058346
iq	.0148581	.0113311	1.31	0.190	-.0073505	.0370667
college 1	1.236947	.6401383	1.93	0.053	-.0177013	2.491595
TEOM0						
_cons	-7.771932	.6406251	-12.13	0.000	-9.027534	-6.51633
TEOM1						
_cons	16.7739	4.777519	3.51	0.000	7.410131	26.13766

Among other things, we can use these correlations to test the joint significance of the coefficients on the residuals from the treatment-assignment models. This is equivalent to the endogeneity test in [example 1](#). We type

```
. test [TEOM0]_cons [TEOM1]_cons
( 1)  [TEOM0]_cons = 0
( 2)  [TEOM1]_cons = 0

      chi2( 2) =   275.36
Prob > chi2 =    0.0000
```

□

Also see

[\[CAUSAL\] eteffects](#) — Endogenous treatment-effects estimation

[\[U\] 20 Estimation and postestimation commands](#)

[Description](#)
[Options](#)
[References](#)[Quick start](#)
[Remarks and examples](#)
[Also see](#)[Menu](#)
[Stored results](#)[Syntax](#)
[Methods and formulas](#)

Description

`etpoisson` estimates the parameters of a Poisson regression model in which one of the regressors is an endogenous binary treatment. Both the average treatment effect and the average treatment effect on the treated can be estimated with `etpoisson`.

Quick start

Poisson model of `y` on `x` and endogenous binary treatment `treat` modeled by `x` and `w`

```
etpoisson y x, treat(treat = x w)
```

With robust standard errors

```
etpoisson y x, treat(treat = x w) vce(robust)
```

Average treatment effect after `etpoisson` with the required `vce(robust)` option

```
margins r.treat, vce(unconditional)
```

Same as above, but calculate average treatment effect on the treated

```
margins, vce(unconditional) predict(cte) subpop(if treat==1)
```

Menu

Statistics > Causal inference/treatment effects > Endogenous treatment > Maximum likelihood estimator > Count outcomes

Syntax

```
etpoisson depvar [indepvars] [if] [in] [weight] ,
      treat(depvart = indepvarst [ , noconstant offset(varnameo) ]) [options]
```

<i>options</i>	Description
Model	
* <u>treat</u> ()	equation for treatment effects
<u>noconstant</u>	suppress constant term
<u>exposure</u> (<i>varname</i> _{<i>e</i>})	include ln(<i>varname</i> _{<i>e</i>}) in model with coefficient constrained to 1
<u>offset</u> (<i>varname</i> _{<i>o</i>})	include <i>varname</i> _{<i>o</i>} in model with coefficient constrained to 1
<u>constraints</u> (<i>constraints</i>)	apply specified linear constraints
SE/Robust	
<u>vce</u> (<i>vcetype</i>)	<i>vcetype</i> may be oim, <u>robust</u> , <u>cluster</u> <i>clustvar</i> , opg, <u>bootstrap</u> , or <u>jackknife</u>
Reporting	
<u>level</u> (#)	set confidence level; default is level(95)
<u>irr</u>	report incidence-rate ratios
<u>nocnsreport</u>	do not display constraints
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
<u>intpoints</u> (#)	use # Gauss–Hermite quadrature points; default is intpoints(24)
Maximization	
<u>maximize_options</u>	control the maximization process; seldom used
<u>collinear</u>	keep collinear variables
<u>coeflegend</u>	display legend instead of statistics

*treat() is required.

The full specification is treat(*depvar*_{*t*} = *indepvars*_{*t*} [, noconstant offset(*varname*_{*o*})]).

indepvars and *indepvars*_{*t*} may contain factor variables; see [U] 11.4.3 Factor variables.

depvar, *depvar*_{*t*}, *indepvars*, and *indepvars*_{*t*} may contain time-series operators; see [U] 11.4.4 Time-series varlists.

bayesboot, bootstrap, by, collect, jackknife, rolling, statsby, and svy are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the bootstrap prefix; see [R] bootstrap.

awweights are not allowed with the jackknife prefix; see [R] jackknife.

vce() and weights are not allowed with the svy prefix; see [SVY] svy.

fweights, aweights, iweights, and pweights are allowed; see [U] 11.1.6 weight.

collinear and coeflegend do not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

Model

`treat(depvart = indepvarst [ , noconstant offset(varnameo) ])` specifies the variables and options for the treatment equation. It is an integral part of specifying a treatment-effects model and is required.

The indicator of treatment, *depvar*_{*t*}, should be coded as 0 or 1.

`noconstant`, `exposure`(*varname*_{*e*}), `offset`(*varname*_{*o*}), `constraints`(*constraints*); see [R] [Estimation options](#).

SE/Robust

`vce`(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`, `opg`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

Reporting

`level`(#); see [R] [Estimation options](#).

`irr` reports estimated coefficients transformed to incidence-rate ratios, that is, e^{β_i} rather than β_i . Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated or stored. `irr` may be specified at estimation or when replaying previously estimated results.

`nocnsreport`; see [R] [Estimation options](#).

display_options: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap`(#), `fvwrapon`(*style*), `cformat`(%*fmt*), `pformat`(%*fmt*), `sformat`(%*fmt*), and `nolstretch`; see [R] [Estimation options](#).

Integration

`intpoints`(#) specifies the number of integration points to use for integration by quadrature. The default is `intpoints`(24); the maximum is `intpoints`(128). Increasing this value improves the accuracy but also increases computation time. Computation time is roughly proportional to its value.

Maximization

maximize_options: `difficult`, `technique`(*algorithm_spec*), `iterate`(#), `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance`(#), `ltolerance`(#), `nrtolerance`(#), `nonrntolerance`, and `from`(*init_specs*); see [R] [Maximize](#). These options are seldom used.

Setting the optimization type to `technique`(`bhhh`) resets the default *vcetype* to `vce`(`opg`).

The following options are available with `etpoisson` but are not shown in the dialog box:

`collinear`, `coeflegend`; see [R] [Estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

[Overview](#)
[Basic example](#)
[Average treatment effect \(ATE\)](#)
[Average treatment effect on the treated \(ATET\)](#)

Overview

`etpoisson` estimates the parameters of a Poisson regression model that includes an endogenous binary-treatment variable. The dependent variable must be a Poisson distributed count. The parameters estimated by `etpoisson` can be used to estimate the average treatment effect (ATE) and average treatment effect on the treated (ATET).

We call the model fit by `etpoisson` an endogenous treatment-regression model, although it is also known as an endogenous binary-variable model or as an endogenous dummy-variable model. The endogenous treatment-regression model fit by `etpoisson` is a specific endogenous treatment-effects model; it uses a nonlinear model for the outcome and a constrained normal distribution to model the deviation from the conditional independence assumption imposed by the estimators implemented by `teffects`; see [\[CAUSAL\] teffects intro](#). In treatment-effects jargon, the endogenous binary-variable model fit by `etpoisson` is a nonlinear potential-outcome model that allows for a specific correlation structure between the unobservables that affect the treatment and the unobservables that affect the potential outcomes. See [\[CAUSAL\] etregress](#) for an estimator that allows for a linear-outcome model and a similar model for the endogeneity of the treatment.

More formally, we have an equation for outcome y_j and an equation for treatment t_j :

$$E(y_j | \mathbf{x}_j, t_j, \epsilon_j) = \exp(\mathbf{x}_j \boldsymbol{\beta} + \delta t_j + \epsilon_j)$$

$$t_j = \begin{cases} 1, & \mathbf{w}_j \boldsymbol{\gamma} + u_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

The $\mathbf{x}_j$ are the covariates used to model the outcome, $\mathbf{w}_j$ are the covariates used to model treatment assignment, and error terms ϵ_j and u_j are bivariate normal with mean 0 and covariance matrix

$$\begin{bmatrix} \sigma^2 & \sigma\rho \\ \sigma\rho & 1 \end{bmatrix}$$

The covariates $\mathbf{x}_j$ and $\mathbf{w}_j$ are unrelated to the error terms; in other words, they are exogenous. Note that y_j may be a count or continuous and nonnegative in this specification.

[Terza \(1998\)](#) describes the maximum likelihood estimator used in `etpoisson`. [Terza \(1998\)](#) categorized the model fit by `etpoisson` as an endogenous-switching model. These models involve a binary switch that is endogenous for the outcome. Calculation of the maximum likelihood estimate involves numeric approximation of integrals via Gauss–Hermite quadrature. This is computationally intensive, but the computational costs are reasonable on modern computers.

Basic example

▷ Example 1

In this example, we observe a simulated random sample of 5,000 households. The outcome of interest is the number of trips taken by members of the household in the 24-hour period immediately prior to the interview time.

We have fictional household level data on the following variables: number of trips taken in the past 24 hours (`trips`), distance to the central business district from the household (`cbd`), distance from the household to a public transit node (`ptn`), an indicator of whether there is a full-time worker in the household (`worker`), an indicator of whether the examined period is on a weekend (`weekend`), the ratio of the household income to the median income of the census tract (`realinc`), and an indicator of car ownership (`owncar`). We suspect that unobservables that affect the number of trips also affect the household's propensity to own a car.

We use `etpoisson` to estimate the parameters of a Poisson regression model for the number of trips with car ownership as an endogenous treatment. In subsequent examples, we will use `margins` (see [\[R\] margins](#)) to estimate the ATE and the ATET of car ownership on the number of trips taken by the household. In the `etpoisson` command below, we specify the `vce(robust)` option because we need to specify `vce(unconditional)` when we use `margins` later.

```
. use https://www.stata-press.com/data/r19/trip1
(Household trips, car ownership)

. etpoisson trips cbd ptn worker weekend,
> treat(owncar = cbd ptn worker realinc) vce(robust)

Iteration 0:  Log pseudolikelihood = -14845.147   (not concave)
Iteration 1:  Log pseudolikelihood = -14562.997   (not concave)
Iteration 2:  Log pseudolikelihood = -13655.592   (not concave)
Iteration 3:  Log pseudolikelihood = -12847.219   (not concave)
Iteration 4:  Log pseudolikelihood = -12566.037
Iteration 5:  Log pseudolikelihood = -12440.974
Iteration 6:  Log pseudolikelihood = -12413.485
Iteration 7:  Log pseudolikelihood = -12412.699
Iteration 8:  Log pseudolikelihood = -12412.696
Iteration 9:  Log pseudolikelihood = -12412.696

Poisson regression with endogenous treatment          Number of obs =   5,000
(24 quadrature points)                               Wald chi2(5)  = 397.94
Log pseudolikelihood = -12412.696                    Prob > chi2   = 0.0000
```

	Robust				[95% conf. interval]	
	Coefficient	std. err.	z	P> z		
trips						
cbd	-.0100919	.0020071	-5.03	0.000	-.0140258	-.006158
ptn	-.0204038	.0020289	-10.06	0.000	-.0243805	-.0164272
worker	.692301	.0548559	12.62	0.000	.5847854	.7998166
weekend	.0930517	.034538	2.69	0.007	.0253585	.160745
1.owncar	.5264713	.1124157	4.68	0.000	.3061406	.746802
_cons	-.2340772	.0810812	-2.89	0.004	-.3929934	-.0751609
owncar						
cbd	.007218	.00239	3.02	0.003	.0025337	.0119023
ptn	.0084769	.0024518	3.46	0.001	.0036714	.0132824
worker	.543643	.0504267	10.78	0.000	.4448085	.6424774
realinc	.176479	.0108746	16.23	0.000	.1551652	.1977928
_cons	-.4611246	.0592161	-7.79	0.000	-.5771859	-.3450633
/athrho	.5741169	.0957832	5.99	0.000	.3863852	.7618486
/lnsigma	-.2182037	.0256281	-8.51	0.000	-.2684338	-.1679735
rho	.5183763	.0700449			.3682398	.6421645
sigma	.8039617	.020604			.764576	.8453762

Wald test of indep. eqns. (rho = 0): chi2(1) = 35.93 Prob > chi2 = 0.0000

The Wald test in the header is highly significant, indicating a good model fit. All the covariates are statistically significant, and the Wald test in the footer indicates that we can reject the null hypothesis of no correlation between the treatment errors and the outcome errors.

We can interpret the coefficient on 1.owncar as the logarithm of the ratio of the treatment potential-outcome mean to the control potential-outcome mean. The treatment variable did not interact with any of the outcome covariates, so the effect of each regressor is the same in the two regimes and will cancel from the ratio of potential-outcome means. This means the ratio is equivalent to the exponentiated coefficient on 1.owncar. After discussing the other parameters, we will use 1.incom to obtain this ratio. See [R] [lincom](#) for more information.

The estimated correlation between the treatment-assignment errors and the outcome errors is 0.518, indicating that unobservables that increase the number of trips tend to occur with unobservables that increase the chance of car ownership.

The results for the two ancillary parameters require explanation. `etpoisson` estimates the inverse hyperbolic tangent of ρ ,

$$\operatorname{atanh} \rho = \frac{1}{2} \ln \left(\frac{1 + \rho}{1 - \rho} \right)$$

and $\ln \sigma$ rather than ρ and σ . For numerical stability during optimization, `etpoisson` does not directly estimate ρ and σ .

Now we use `lincom` and the `eform` option to estimate the exponentiated coefficient for `1.owncar`. This corresponds to the ratio of the treatment regime potential-outcome mean to the control regime potential-outcome mean.

```
. lincom [trips]_b[1.owncar], eform
(1) [trips]1.owncar = 0
```

	exp(b)	Std. err.	z	P> z	[95% conf. interval]	
(1)	1.692948	.1903139	4.68	0.000	1.358173	2.110241

The potential-outcome mean for the treatment regime is 1.69 times the potential-outcome mean for the control regime. So the average number of trips in the treatment regime is over one and a half times the average number of trips in the control regime.

By interacting the treatment, `owncar`, with the other regressors, we could estimate different coefficients for the regressors in the treatment and control regimes. In the current model, there are no treatment interactions, so the coefficients are the same in each regime.

◀

Average treatment effect (ATE)

The parameter estimates from `etpoisson` can be used by `margins` to estimate the ATE, the average difference of the treatment and control potential outcomes.

► Example 2

Continuing with [example 1](#), we use `margins` to estimate the ATE of car ownership on the number of trips taken in a 24-hour period.

We can estimate the ATE of car ownership by using the potential-outcome means obtained through the `predict`, `pomean` command and the `margins` command; see [Methods and formulas](#) below and [\[CAUSAL\] etpoisson postestimation](#) for more details about the use of `predict` after `etpoisson`.

The `r.` notation indicates that the potential-outcome means for treatment and control will be contrasted. We specify the `contrast(nowald)` option to suppress the Wald tests that margins displays by default for contrasts.

```
. margins r.owncar, vce(unconditional) contrast(nowald)
Contrasts of predictive margins                                Number of obs = 5,000
Expression: Potential-outcome mean, predict()
```

	Unconditional			
	Contrast	std. err.	[95% conf. interval]	
owncar (1 vs 0)	1.058914	.1922909	.6820309	1.435797

The estimated ATE of car ownership on the number of trips taken is 1.06. The average household will take 1.06 more trips when it owns a car.

◀

Average treatment effect on the treated (ATET)

The parameter estimates from `etpoisson` can be used by `margins` to estimate the ATET, the average difference of the treatment and control potential outcomes in the treated population.

► Example 3

Continuing with the [previous example](#), we use `margins` to estimate the ATET of car ownership on the number of trips taken in a 24-hour period.

We can estimate the ATET of car ownership by using the conditional treatment effect (conditional on exogenous covariates and treatment level) obtained through the `predict, cte` command and the `margins` command; see [Methods and formulas](#) below and [\[CAUSAL\] etpoisson postestimation](#) for more details about the use of `predict` after `etpoisson`.

We estimate the ATET with `margins`. We specify `cte` in the `predict()` option. Estimation is restricted to the treated subpopulation by specifying `owncar` in the `subpop()` option.

```
. margins, predict(cte) vce(unconditional) subpop(owncar)
Predictive margins                                Number of obs = 5,000
                                                Subpop. no. obs = 3,504
Expression: Conditional treatment effect, predict(cte)
```

	Unconditional				[95% conf. interval]	
	Margin	std. err.	z	P> z		
_cons	1.251971	.2059201	6.08	0.000	.8483747	1.655567

The estimated ATET of car ownership on the number of trips taken is 1.25. Thus the average household in the treated population will take 1.25 more trips than it would if it did not own a car. This number is higher than the ATE. In this model, the ATE and ATET will only coincide when there is no correlation between the treatment errors and outcome errors and the exogenous covariates $\mathbf{x}$ have the same distribution in the general population and treated subpopulation. See [Methods and formulas](#) for more details.

◀

Stored results

etpoisson stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_eq_model)</code>	number of equations in overall model test
<code>e(k_aux)</code>	number of auxiliary parameters
<code>e(k_dv)</code>	number of dependent variables
<code>e(df_m)</code>	model degrees of freedom
<code>e(ll)</code>	log likelihood
<code>e(N_clust)</code>	number of clusters
<code>e(chi2)</code>	χ^2
<code>e(chi2_c)</code>	χ^2 for comparison, $\rho = 0$ test
<code>e(n_quad)</code>	number of quadrature points
<code>e(p)</code>	p -value for model test
<code>e(p_c)</code>	p -value for comparison test
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(ic)</code>	number of iterations
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	etpoisson
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(title2)</code>	secondary title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(offset1)</code>	offset for regression equation
<code>e(offset2)</code>	offset for treatment equation
<code>e(chi2type)</code>	Wald; type of model χ^2 test
<code>e(chi2_ct)</code>	Wald; type of comparison χ^2 test
<code>e(vce)</code>	<i>vce</i> type specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(opt)</code>	type of optimization
<code>e(which)</code>	max or min; whether optimizer is to perform maximization or minimization
<code>e(ml_method)</code>	type of ml method
<code>e(user)</code>	name of likelihood-evaluator program
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	b V
<code>e(predict)</code>	program used to implement predict
<code>e(marginsok)</code>	predictions allowed by margins
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(Cns)</code>	constraints matrix
<code>e(ilog)</code>	iteration log (up to 20 iterations)
<code>e(gradient)</code>	gradient vector
<code>e(V)</code>	variance–covariance matrix of the estimators
<code>e(V_modelbased)</code>	model-based variance

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`

matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

Terza (1998) derives the maximum likelihood estimator implemented here. We provide some details of the derivation and then explain how the model is nested in a more general potential-outcomes model. Then the ATE and ATET are derived.

Let $\mathbf{x}_j$ be the covariates used to model the outcome, and let $\mathbf{w}_j$ be the covariates used to model treatment assignment. Define $\mathbf{z}_j = (\mathbf{w}_j, \mathbf{x}_j)$. The vector $\mathbf{z}_j$ contains all the exogenous covariates in the model. When offsets o_j^β are used in the outcome variable equation, the following formulas apply with $\mathbf{x}_j\beta$ changed to $\mathbf{x}_j\beta + o_j^\beta$. Similarly, when offsets o_j^γ are used in the endogenous treatment equation, the following formulas apply with $\mathbf{w}_j\gamma$ changed to $\mathbf{w}_j\gamma + o_j^\gamma$. If offsets are used in either equation, they are included in the vector of exogenous covariates $\mathbf{z}_j$.

For treatment t_j , $\mathbf{z}_j$, and ϵ_j , outcome y_j of this model has conditional mean

$$E(y_j|\mathbf{z}_j, t_j, \epsilon_j) = \exp(\mathbf{x}_j\beta + \delta t_j + \epsilon_j) \quad (1)$$

The probability density function of y_j for this model, conditioned on treatment t_j , $\mathbf{z}_j$, and ϵ_j , is given by

$$f(y_j|\mathbf{z}_j, t_j, \epsilon_j) = \frac{\exp\{-\exp(\mathbf{x}_j\beta + \delta t_j + \epsilon_j)\} \{\exp(\mathbf{x}_j\beta + \delta t_j + \epsilon_j)\}^{y_j}}{y_j!}$$

The treatment t_j is determined by

$$t_j = \begin{cases} 1, & \text{if } \mathbf{w}_j\gamma + u_j > 0 \\ 0, & \text{otherwise} \end{cases}$$

The error terms ϵ_j and u_j are bivariate normal with mean zero and covariance matrix

$$\begin{bmatrix} \sigma^2 & \sigma\rho \\ \sigma\rho & 1 \end{bmatrix}$$

Conditional on ϵ_j , u_j is normal with mean $\epsilon_j\rho/\sigma$ and variance $(1 - \rho^2)$; thus we obtain the following conditional probability density for t_j :

$$\Pr(t_j|\mathbf{z}_j, \epsilon_j) = t_j\Phi\left\{\frac{\mathbf{w}_j\gamma + (\rho/\sigma)\epsilon_j}{\sqrt{1 - \rho^2}}\right\} + (1 - t_j)\left[1 - \Phi\left\{\frac{\mathbf{w}_j\gamma + (\rho/\sigma)\epsilon_j}{\sqrt{1 - \rho^2}}\right\}\right]$$

Φ denotes the standard normal cumulative distribution function. This leads to the following joint density of y_j , t_j , and ϵ_j :

$$f(y_j, t_j, \epsilon_j|\mathbf{z}_j) = f(y_j|\mathbf{z}_j, t_j, \epsilon_j)P(t_j|\mathbf{z}_j, \epsilon_j)f(\epsilon_j)$$

The density of y_j and t_j , conditioned on $\mathbf{z}_j$, is obtained by integrating the above with respect to ϵ_j . Recall that ϵ_j is normal with mean 0 and variance σ^2 .

$$f(y_j, t_j | \mathbf{z}_j) = \int_{-\infty}^{\infty} f(y_j | \mathbf{z}_j, t_j, \epsilon_j) P(t_j | \mathbf{z}_j, \epsilon_j) \frac{1}{\sigma\sqrt{2\pi}} \exp \left\{ - \left(\frac{\epsilon_j}{\sigma\sqrt{2}} \right)^2 \right\} d\epsilon_j$$

$f(y_j, t_j | \mathbf{z}_j)$ cannot be evaluated in a closed form. We change the variable of integration from ϵ_j to $\eta_j = \epsilon_j / (\sigma\sqrt{2})$, which yields

$$f(y_j, t_j | \mathbf{z}_j) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} f(y_j | \mathbf{z}_j, t_j, \sqrt{2}\sigma\eta_j) P(t_j | \mathbf{z}_j, \sqrt{2}\sigma\eta_j) \exp(-\eta_j^2) d\eta_j$$

We approximate this integral by Gauss–Hermite quadrature. Observing a sample of t_j , y_j , and $\mathbf{z}_j$, we calculate the log likelihood as the following:

$$\ln L = \sum_{j=1}^n w_j \ln \{ f(y_j, t_j | \mathbf{z}_j) \}$$

The w_j terms denote optional weights.

In the maximum likelihood estimation, σ and ρ are not directly estimated. Directly estimated are $\ln \sigma$ and $\operatorname{atanh} \rho$:

$$\operatorname{atanh} \rho = \frac{1}{2} \ln \left(\frac{1 + \rho}{1 - \rho} \right)$$

Now we present formulas for the ATE and ATET. First, we nest the endogenous-treatment Poisson regression model in a potential-outcome model. A potential-outcome model specifies what each individual would obtain in each treatment level.

A potential-outcome model that nests the endogenous-treatment Poisson regression fit by `etpoisson` is

$$\begin{aligned} E(y_{0j} | \mathbf{x}_j, \epsilon_j) &= \exp(\mathbf{x}_j \boldsymbol{\beta}_0 + \epsilon_{0j}) \\ E(y_{1j} | \mathbf{x}_j, \epsilon_j) &= \exp(\mathbf{x}_j \boldsymbol{\beta}_1 + \epsilon_{1j}) \\ t_j &= \begin{cases} 1, & \text{if } \mathbf{w}_j \boldsymbol{\gamma} + u_j > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

where y_{0j} is the outcome that person j obtains if person j selects treatment 0, and y_{1j} is the outcome that person j obtains if person j selects treatment 1. This formulation allows differing coefficients for the control ($\boldsymbol{\beta}_0$) and treatment ($\boldsymbol{\beta}_1$) regimes. The constant intercept for the control group is β_{00} . The constant intercept for the treatment group is $\beta_{11} = \beta_{00} + \delta$, where δ is the coefficient for treatment t_j in the outcome (1). The remaining notation was defined above.

We may allow other coefficients to differ across regimes in the outcome (1) by adding interactions between the treatment t_j and covariates $\mathbf{x}_j$ to the model. To be concise, we use two coefficient vectors $\boldsymbol{\beta}_0$ and $\boldsymbol{\beta}_1$ here rather than a single coefficient vector with interactions between the treatment t_j and covariates $\mathbf{x}_j$. The two formulations are equivalent.

We never observe both y_{0j} and y_{1j} , only one or the other. We observe

$$y_j = t_j y_{1j} + (1 - t_j) y_{0j}$$

The vector of error terms $(\epsilon_{0j}, \epsilon_{1j}, u_j)'$ comes from a mean zero trivariate normal distribution with covariance matrix

$$\begin{bmatrix} \sigma^2 & \theta & \sigma\rho \\ \theta & \sigma^2 & \sigma\rho \\ \sigma\rho & \sigma\rho & 1 \end{bmatrix}$$

The parameters σ and ρ were discussed earlier. The parameter θ is the covariance between the two potential outcomes. We cannot estimate θ because we have no observations in which an individual is observed in both potential outcomes. Fortunately, θ is not required for the calculations that we present.

The ATE is the difference in means of the potential outcomes. The mean of each potential outcome accounts for each individual's contribution, regardless of whether that individual selects that treatment level.

The conditional means of the potential outcomes y_{tj} , $t \in (0, 1)$ for exogenous covariates $\mathbf{z}_j$ are

$$E(y_{tj}|\mathbf{z}_j) = \exp\left(\mathbf{x}_j\boldsymbol{\beta}_t + \frac{\sigma^2}{2}\right)$$

We can see that when the coefficients are the same across the regimes, the ratio of potential-outcome means will be equal to $\exp(\delta)$; this is true of the conditional and marginal potential-outcome means.

The difference in potential-outcome means or treatment effect at exogenous covariates $\mathbf{z}_j$ is

$$E(y_{1j} - y_{0j}|\mathbf{z}_j) = \{\exp(\mathbf{x}_j\boldsymbol{\beta}_1) - \exp(\mathbf{x}_j\boldsymbol{\beta}_0)\} \exp\left(\frac{\sigma^2}{2}\right)$$

By the law of iterated expectations, the ATE is

$$\begin{aligned} E(y_{1j} - y_{0j}) &= E\{E(y_{1j} - y_{0j}|\mathbf{z}_j)\} \\ &= E\left[\{\exp(\mathbf{x}_j\boldsymbol{\beta}_1) - \exp(\mathbf{x}_j\boldsymbol{\beta}_0)\} \exp\left(\frac{\sigma^2}{2}\right)\right] \end{aligned}$$

This expectation can be estimated as a predictive margin.

Now we will derive an expression for the ATET.

The conditional means of the potential outcomes y_{tj} , $t \in (0, 1)$ for exogenous covariates $\mathbf{z}_j$ and treatment t_j are

$$E(y_{tj}|\mathbf{z}_j, t_j) = \exp\left(\mathbf{x}_j\boldsymbol{\beta}_t + \frac{\sigma^2}{2}\right) \left\{ \frac{\Phi(\rho\sigma + \mathbf{w}_j\boldsymbol{\gamma})}{\Phi(\mathbf{w}_j\boldsymbol{\gamma})} \right\}^{t_j} \left\{ \frac{1 - \Phi(\rho\sigma + \mathbf{w}_j\boldsymbol{\gamma})}{1 - \Phi(\mathbf{w}_j\boldsymbol{\gamma})} \right\}^{1-t_j}$$

Rather than the conditional potential-outcome means, the conditional mean of the observed outcome may be of interest. The conditional mean of the observed outcome y_j for endogenous treatment indicator t_j and exogenous covariates $\mathbf{z}_j$ is given by

$$E(y_j|\mathbf{z}_j, t_j) = t_j \exp\left(\mathbf{x}_j\boldsymbol{\beta}_1 + \frac{\sigma^2}{2}\right) \frac{\Phi(\rho\sigma + \mathbf{w}_j\boldsymbol{\gamma})}{\Phi(\mathbf{w}_j\boldsymbol{\gamma})} \\ + (1 - t_j) \exp\left(\mathbf{x}_j\boldsymbol{\beta}_0 + \frac{\sigma^2}{2}\right) \frac{1 - \Phi(\rho\sigma + \mathbf{w}_j\boldsymbol{\gamma})}{1 - \Phi(\mathbf{w}_j\boldsymbol{\gamma})}$$

The treatment effect at exogenous covariates $\mathbf{z}_j$ and treatment t_j is

$$E(y_{1j} - y_{0j}|\mathbf{z}_j, t_j) = \\ \left\{ \exp(\mathbf{x}_j\boldsymbol{\beta}_1) - \exp(\mathbf{x}_j\boldsymbol{\beta}_0) \right\} \exp\left(\frac{\sigma^2}{2}\right) \left\{ \frac{\Phi(\rho\sigma + \mathbf{w}'_j\boldsymbol{\gamma})}{\Phi(\mathbf{w}_j\boldsymbol{\gamma})} \right\}^{t_j} \left\{ \frac{1 - \Phi(\rho\sigma + \mathbf{w}_j\boldsymbol{\gamma})}{1 - \Phi(\mathbf{w}'_j\boldsymbol{\gamma})} \right\}^{1-t_j}$$

By the law of iterated expectations, the ATET is

$$E(y_{1j} - y_{0j}|t_j = 1) = E\{E(y_{1j} - y_{0j}|\mathbf{z}_j, t_j = 1)|t_j = 1\} \\ = E\left[\left\{ \exp(\mathbf{x}_j\boldsymbol{\beta}_1) - \exp(\mathbf{x}_j\boldsymbol{\beta}_0) \right\} \exp\left(\frac{\sigma^2}{2}\right) \frac{\Phi(\rho\sigma + \mathbf{w}_j\boldsymbol{\gamma})}{\Phi(\mathbf{w}_j\boldsymbol{\gamma})} \middle| t_j = 1\right]$$

This can be estimated as a predictive margin on the treated subpopulation.

We note that when $\rho = 0$, the correction factor involving Φ will disappear from the ATET. Then the ATE and ATET will be equivalent if the distribution of $\mathbf{x}_j$ under the treated population is identical to the distribution over the entire population.

The probability of y_j conditional on t_j and $\mathbf{z}_j$ is

$$\Pr(y_j = n|\mathbf{z}_j, t_j) = \frac{f(y_j = n, t_j|\mathbf{z}_j)}{\Phi(\mathbf{w}_j\boldsymbol{\gamma})^{t_j} \Phi(-\mathbf{w}_j\boldsymbol{\gamma})^{1-t_j}}$$

As discussed earlier, we approximate $f(y_j, t_j|\mathbf{z}_j)$ using Gauss–Hermite quadrature.

References

- Cerulli, G. 2022. *Econometric Evaluation of Socio-Economic Programs: Theory and Applications*. 2nd ed. Berlin: Springer.
- Hasebe, T. 2020. Endogenous switching regression model and treatment effects of count-data outcome. *Stata Journal* 20: 627–646.
- Terza, J. V. 1998. Estimating count data models with endogenous switching: Sample selection and endogenous treatment effects. *Journal of Econometrics* 84: 129–154. [https://doi.org/10.1016/S0304-4076\(97\)00082-1](https://doi.org/10.1016/S0304-4076(97)00082-1).

Also see

- [CAUSAL] **etpoisson postestimation** — Postestimation tools for etpoisson
- [CAUSAL] **etregress** — Linear regression with endogenous treatment effects
- [R] **heckpoisson** — Poisson regression with sample selection
- [R] **ivpoisson** — Poisson model with continuous endogenous covariates
- [R] **ivprobit** — Probit model with continuous endogenous covariates
- [R] **ivregress** — Single-equation instrumental-variables regression⁺
- [R] **ivtobit** — Tobit model with continuous endogenous covariates
- [R] **poisson** — Poisson regression
- [SVY] **svy estimation** — Estimation commands for survey data
- [U] **20 Estimation and postestimation commands**

Postestimation commands

The following standard postestimation commands are available after `etpoisson`:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of parameters
<code>estat ic</code>	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estat (svy)</code>	postestimation statistics for survey data
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
* <code>hausman</code>	Hausman's specification test
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of parameters
* <code>lrtest</code>	likelihood-ratio test
<code>margins</code>	marginal means, predictive margins, marginal effects, and average marginal effects
<code>marginsplot</code>	graph the results from margins (profile plots, interaction plots, etc.)
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
<code>predict</code>	potential-outcome means, observed-outcome means, conditional treatment effects, etc.
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of parameters
<code>suest</code>	seemingly unrelated estimation
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

*`hausman` and `lrtest` are not appropriate with `svy` estimation results.

predict

Description for predict

predict creates a new variable containing predictions such as counts, conditional treatment effects, probabilities, and linear predictions.

Menu for predict

Statistics > Postestimation

Syntax for predict

```
predict [type] newvar [if] [in] [ , statistic nooffset ]
```

```
predict [type] stub* [if] [in] , scores
```

statistic	Description
Main	
<u>p</u> mean	potential-outcome mean (the predicted count); the default
<u>o</u> mean	observed-outcome mean (the predicted count)
cte	conditional treatment effect at treatment level
pr(<i>n</i>)	probability $\Pr(y_j = n)$
pr(<i>a</i> , <i>b</i>)	probability $\Pr(a \leq y_j \leq b)$
xb	linear prediction
<u>x</u> b <u>t</u> reat	linear prediction for treatment equation

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

Options for predict

Main

`pomean`, the default, calculates the potential-outcome mean.

`omean` calculates the observed-outcome mean.

`cte` calculates the treatment effect, the difference of potential-outcome means, conditioned on treatment level.

`pr( $n$ )` calculates the probability $\Pr(y_j = n)$, where n is a nonnegative integer that may be specified as a number or a variable.

`pr( $a, b$ )` calculates the probability $\Pr(a \leq y_j \leq b)$, where a and b are nonnegative integers that may be specified as numbers or variables;

b missing ($b \geq .$) means $+\infty$;

`pr(20, .)` calculates $\Pr(y_j \geq 20)$;

`pr(20,  $b$ )` calculates $\Pr(y_j \geq 20)$ in observations for which $b \geq .$ and calculates

$\Pr(20 \leq y_j \leq b)$ elsewhere.

`pr(. ,  $b$ )` produces a syntax error. A missing value in an observation of the variable a causes a missing value in that observation for `pr( $a, b$ )`.

`xb` calculates the linear prediction for the dependent count variable, which is $\mathbf{x}_j\boldsymbol{\beta}$ if neither `offset()` nor `exposure()` was specified; $\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j^\beta$ if `offset()` was specified; or $\mathbf{x}_j\boldsymbol{\beta} + \ln(\text{exposure}_j)$ if `exposure()` was specified.

`xbtreat` calculates the linear prediction for the endogenous treatment equation, which is $\mathbf{w}_j\boldsymbol{\gamma}$ if `offset()` was not specified in `treat()` and $\mathbf{w}_j\boldsymbol{\gamma} + \text{offset}_j^\alpha$ if `offset()` was specified in `treat()`.

`nooffset` is relevant only if you specified `offset()` or `exposure()` when you fit the model. It modifies the calculations made by `predict` so that they ignore the offset or exposure variable. `nooffset` removes the offset from calculations involving both the `treat()` equation and the dependent count variable.

`scores` calculates equation-level score variables.

The first new variable will contain $\partial \ln L / \partial (\mathbf{x}_j\boldsymbol{\beta})$.

The second new variable will contain $\partial \ln L / \partial (\mathbf{w}_j\boldsymbol{\gamma})$.

The third new variable will contain $\partial \ln L / \partial \text{atanh } \rho$.

The fourth new variable will contain $\partial \ln L / \partial \ln \sigma$.

margins

Description for margins

margins estimates margins of response for counts, conditional treatment effects, probabilities, and linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

margins [marginlist] [, options]
margins [marginlist] , predict(statistic ...) [predict(statistic ...) ...] [options]

statistic	Description
<u>p</u> omean	potential-outcome mean (the predicted count); the default
<u>o</u> mean	observed-outcome mean (the predicted count)
cte	conditional treatment effect at treatment level
pr(<i>n</i>)	probability $\Pr(y_j = n)$
pr(<i>a</i> , <i>b</i>)	probability $\Pr(a \leq y_j \leq b)$
xb	linear prediction
<u>x</u> b <u>t</u> reat	linear prediction for treatment equation

Statistics not allowed with margins are functions of stochastic quantities other than e(b).

For the full syntax, see [R] margins.

Remarks and examples

The average treatment effect (ATE) and the average treatment effect on the treated (ATET) are the parameters most frequently estimated by postestimation techniques after etpoisson.

You can use the margins command (see [R] margins) after etpoisson to estimate the ATE or ATET. See example 2 of [CAUSAL] etpoisson for an example of ATE estimation. See example 3 of [CAUSAL] etpoisson for an example of ATET estimation.

See example 1 of [CAUSAL] etpoisson for an example using lincom after etpoisson.

Methods and formulas

See Methods and formulas of [CAUSAL] etpoisson for details.

Also see

- [CAUSAL] etpoisson — Poisson regression with endogenous treatment effects
- [U] 20 Estimation and postestimation commands

[Description](#)[Menu](#)[Options for maximum likelihood estimates](#)[Options for control-function estimates](#)[Stored results](#)[References](#)[Quick start](#)[Syntax](#)[Options for two-step consistent estimates](#)[Remarks and examples](#)[Methods and formulas](#)[Also see](#)

Description

`etregress` estimates an average treatment effect (ATE) and the other parameters of a linear regression model augmented with an endogenous binary-treatment variable. Estimation is by full maximum likelihood, a two-step consistent estimator, or a control-function estimator.

In addition to the ATE, `etregress` can be used to estimate the average treatment effect on the treated (ATET) when the outcome may not be conditionally independent of the treatment.

Quick start

ATE and ATET from a linear regression model of `y` on `x` and endogenous binary treatment `treat` modeled by `x` and `w`

```
etregress y x, treat(treat = x w)
```

Same as above, but use a control-function estimator

```
etregress y x, treat(treat = x w) cfunction
```

With robust standard errors

```
etregress y x, treat(treat = x w) vce(robust)
```

Add the interaction between `treat` and continuous covariate `x` using [factor variables](#)

```
etregress y x i.treat#c.x, treat(treat = x w) vce(robust)
```

ATE after `etregress` with the required `vce(robust)` option and endogenous treatment interaction terms

```
margins r.treat, vce(unconditional)
```

Same as above, but calculate ATET

```
margins, vce(unconditional) predict(cte) subpop(if treat==1)
```

Menu

Statistics > Causal inference/treatment effects > Endogenous treatment > Maximum likelihood estimator > Continuous outcomes

Syntax

Basic syntax

```
etregress depvar [indepvars] , treat(depvart = indepvarst) [twostep | cfunction]
```

Full syntax for maximum likelihood estimates only

```
etregress depvar [indepvars] [if] [in] [weight] ,  
  treat(depvart = indepvarst [, noconstant]) [etregress_ml_options]
```

Full syntax for two-step consistent estimates only

```
etregress depvar [indepvars] [if] [in] ,  
  treat(depvart = indepvarst [, noconstant]) twostep [etregress_ts_options]
```

Full syntax for control-function estimates only

```
etregress depvar [indepvars] [if] [in] [weight] ,  
  treat(depvart = indepvarst [, noconstant]) cfunction [etregress_cf_options]
```


<i>etregress_ml_options</i>	Description
Model	
* <u>treat</u> ()	equation for treatment effects
<u>noconstant</u>	suppress constant term
<u>poutcomes</u>	use potential-outcome model with separate treatment and control group variance and correlation parameters
<u>constraints</u> (<i>constraints</i>)	apply specified linear constraints
SE/Robust	
<u>vce</u> (<i>vcetype</i>)	<i>vcetype</i> may be <u>oim</u> , <u>robust</u> , <u>cluster</u> <i>clustvar</i> , <u>opg</u> , <u>bootstrap</u> , or <u>jackknife</u>
Reporting	
<u>level</u> (#)	set confidence level; default is level(95)
<u>first</u>	report first-step probit estimates
<u>hazard</u> (<i>newvar</i>)	create <i>newvar</i> containing hazard from treatment equation
<u>lrmodel</u>	perform the likelihood-ratio model test instead of the default Wald test
<u>nocnsreport</u>	do not display constraints
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<u>maximize_options</u>	control the maximization process; seldom used
<u>collinear</u>	keep collinear variables
<u>coeflegend</u>	display legend instead of statistics
* <u>treat</u> (<i>depvar</i> _{<i>t</i>} = <i>indepvars</i> _{<i>t</i>} [, <u>noconstant</u>]) is required.	

<i>etregress_ts_options</i>	Description
Model	
* <u>treat</u> ()	equation for treatment effects
* <u>twostep</u>	produce two-step consistent estimate
<u>noconstant</u>	suppress constant term
SE	
<u>vce</u> (<i>vcetype</i>)	<i>vcetype</i> may be <u>conventional</u> , <u>bootstrap</u> , or <u>jackknife</u>
Reporting	
<u>level</u> (#)	set confidence level; default is level(95)
<u>first</u>	report first-step probit estimates
<u>hazard</u> (<i>newvar</i>)	create <i>newvar</i> containing hazard from treatment equation
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
<u>coeflegend</u>	display legend instead of statistics
* <u>treat</u> (<i>depvar</i> _{<i>t</i>} = <i>indepvars</i> _{<i>t</i>} [, <u>noconstant</u>]) and <u>twostep</u> are required.	

<i>etregress _cf_options</i>	Description
Model	
* <u>treat()</u>	equation for treatment effects
* <u>cfunction</u>	produce control-function estimate
<u>noconstant</u>	suppress constant term
<u>poutcomes</u>	use potential-outcome model with separate treatment and control group variance and correlation parameters
SE	
<u>vce</u> (<i>vcetype</i>)	<i>vcetype</i> may be <u>robust</u> , <u>bootstrap</u> , or <u>jackknife</u>
Reporting	
<u>level</u> (#)	set confidence level; default is level(95)
<u>first</u>	report first-step probit estimates
<u>hazard</u> (<i>newvar</i>)	create <i>newvar</i> containing hazard from treatment equation
<i>display_options</i>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<i>maximize_options</i>	control the maximization process; seldom used
<u>coeflegend</u>	display legend instead of statistics

**treat*(*depvar*_{*t*} = *indepvars*_{*t*} [, noconstant]) and *cfunction* are required.

indepvars and *indepvars*_{*t*} may contain factor variables; see [U] 11.4.3 Factor variables.

depvar, *indepvars*, *depvar*_{*t*}, and *indepvars*_{*t*} may contain time-series operators; see [U] 11.4.4 Time-series varlists.

bayesboot, *bootstrap*, *by*, *collect*, *fp*, *jackknife*, *rolling*, *statsby*, and *svy* are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the *bootstrap* prefix; see [R] *bootstrap*.

*aweight*s are not allowed with the *jackknife* prefix; see [R] *jackknife*.

twostep, *cfunction*, *vce*(), *first*, *hazard*(), *lrmodel*, and weights are not allowed with the *svy* prefix; see [SVY] *svy*.

*pweight*s, *aweight*s, *fweight*s, and *iweight*s are allowed with both maximum likelihood and control-function estimation; see [U] 11.1.6 *weight*. No weights are allowed if *twostep* is specified.

collinear and *coeflegend* do not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options for maximum likelihood estimates

Model

`treat(depvart = indepvarst [, noconstant])` specifies the variables and options for the treatment equation. It is an integral part of specifying a treatment-effects model and is required.

`noconstant`; see [R] [Estimation options](#).

`poutcomes` specifies that a potential-outcome model with separate variance and correlation parameters for each of the treatment and control groups be used.

`constraints(constraints)`; see [R] [Estimation options](#).

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`, `opg`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

Reporting

`level(#)`; see [R] [Estimation options](#).

`first` specifies that the first-step probit estimates of the treatment equation be displayed before estimation.

`hazard(newvar)` will create a new variable containing the hazard from the treatment equation. The hazard is computed from the estimated parameters of the treatment equation.

`lrmodel`, `nocnsreport`; see [R] [Estimation options](#).

`display_options`: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no1stretch`; see [R] [Estimation options](#).

Maximization

`maximize_options`: `difficult`, `technique(algorithm_spec)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrtolerance`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

Setting the optimization type to `technique(bhhh)` resets the default `vcetype` to `vce(opg)`.

The following options are available with `etregress` but are not shown in the dialog box:

`collinear`, `coeflegend`; see [R] [Estimation options](#).

Options for two-step consistent estimates

Model

`treat(depvart = indepvarst[ , noconstant ])` specifies the variables and options for the treatment equation. It is an integral part of specifying a treatment-effects model and is required.

`twostep` specifies that two-step consistent estimates of the parameters, standard errors, and covariance matrix be produced, instead of the default maximum likelihood estimates.

`noconstant`; see [R] [Estimation options](#).

SE

`vce(vcetype)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`conventional`) and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

`vce(conventional)`, the default, uses the conventionally derived variance estimator for the two-step estimator of the treatment-effects model.

Reporting

`level(#)`; see [R] [Estimation options](#).

`first` specifies that the first-step probit estimates of the treatment equation be displayed before estimation.

`hazard(newvar)` will create a new variable containing the hazard from the treatment equation. The hazard is computed from the estimated parameters of the treatment equation.

display_options: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no1stretch`; see [R] [Estimation options](#).

The following option is available with `etregress` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Options for control-function estimates

Model

`treat(depvart = indepvarst[ , noconstant ])` specifies the variables and options for the treatment equation. It is an integral part of specifying a treatment-effects model and is required.

`cfunction` specifies that control-function estimates of the parameters, standard errors, and covariance matrix be produced instead of the default maximum likelihood estimates. `cfunction` is required.

`noconstant`; see [R] [Estimation options](#).

`poutcomes` specifies that a potential-outcome model with separate variance and correlation parameters for each of the treatment and control groups be used.

SE

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`) and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

Reporting

`level(#)`; see [R] [Estimation options](#).

`first` specifies that the first-step probit estimates of the treatment equation be displayed before estimation.

`hazard(newvar)` will create a new variable containing the hazard from the treatment equation. The hazard is computed from the estimated parameters of the treatment equation.

`display_options`: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

Maximization

`maximize_options`: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

`init_specs` is one of

`matname [ , skip copy]`

`# [ # ... ] copy`

The following option is available with `etregress` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Basic examples](#)

[Average treatment effect \(ATE\)](#)

[Average treatment effect on the treated \(ATET\)](#)

Overview

`etregress` estimates an ATE and the other parameters of a linear regression model that also includes an endogenous binary-treatment variable. In addition to the ATE, the parameters estimated by `etregress` can be used to estimate the ATET when the outcome is not conditionally independent of the treatment.

We call the model fit by `etregress` an endogenous treatment-regression model, although it is also known as an endogenous binary-variable model or as an endogenous dummy-variable model. The endogenous treatment-regression model is a specific endogenous treatment-effects model; it uses a linear model for the outcome and a normal distribution to model the deviation from the conditional independence assumption imposed by the estimators implemented in `teffects`; see [CAUSAL] [teffects intro](#). In treatment-effects jargon, the endogenous binary-variable model is a linear potential-outcome model that allows for a specific correlation structure between the unobservables that affect the treatment and the unobservables that affect the potential outcomes. See [CAUSAL] [etpoisson](#) for an estimator that allows for a nonlinear outcome model and a similar model for the endogeneity of the treatment.

Heckman (1976, 1978) brought this model into the modern literature. Maddala (1983) derives the maximum likelihood and the control-function (CF) estimators of the model. Maddala (1983) also reviews some empirical applications and describes it as an endogenous-switching model. Barnow, Cain, and Goldberger (1981) provide another useful derivation of this model. They concentrate on deriving the conditions for which the self-selection bias of the simple OLS estimator of the treatment effect, δ , is nonzero and of a specific sign. Cameron and Trivedi (2005, sec. 16.7 and 25.3.4) and Wooldridge (2010, sec. 21.4.1) discuss the endogenous binary-variable model as an endogenous treatment-effects model and link it to recent work.

etregress performs CF estimation in one step by using the generalized method of moments (GMM) with stacked moments. See Newey (1984) and Wooldridge (2010, sec. 14.2) for a description of this technique. Many econometric and statistical models can be expressed as conditions on the population moments. The parameter estimates produced by GMM estimators make the sample-moment conditions as true as possible given the data. See [R] gmm for further information on GMM estimation and how Stata performs it. Two-step CF estimation is also supported by etregress.

Formally, the endogenous treatment-regression model is composed of an equation for the outcome y_j and an equation for the endogenous treatment t_j . The variables $\mathbf{x}_j$ are used to model the outcome. When there are no interactions between t_j and $\mathbf{x}_j$, we have

$$\begin{aligned} y_j &= \mathbf{x}_j\boldsymbol{\beta} + \delta t_j + \epsilon_j \\ t_j &= \begin{cases} 1, & \text{if } \mathbf{w}_j\boldsymbol{\gamma} + u_j > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

where $\mathbf{w}_j$ are the covariates used to model treatment assignment, and the error terms ϵ_j and u_j are bivariate normal with mean zero and covariance matrix

$$\begin{bmatrix} \sigma^2 & \rho\sigma \\ \rho\sigma & 1 \end{bmatrix}$$

The covariates $\mathbf{x}_j$ and $\mathbf{w}_j$ are unrelated to the error terms; in other words, they are exogenous. We call this the constrained model because the variance and correlation parameters are identical across the treatment and control groups.

This model can be generalized to a potential-outcome model with separate variance and correlation parameters for the treatment and control groups. The generalized model is

$$\begin{aligned} y_{0j} &= \mathbf{x}_j\boldsymbol{\beta}_0 + \epsilon_{0j} \\ y_{1j} &= \mathbf{x}_j\boldsymbol{\beta}_1 + \epsilon_{1j} \\ t_j &= \begin{cases} 1, & \text{if } \mathbf{w}_j\boldsymbol{\gamma} + u_j > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

where y_{0j} is the outcome that person j obtains if person j selects treatment 0, and y_{1j} is the outcome that person j obtains if person j selects treatment 1. We never observe both y_{0j} and y_{1j} , only one or the other. We observe

$$y_j = t_j y_{1j} + (1 - t_j) y_{0j}$$

In this unconstrained model, the vector of error terms $(\epsilon_{0j}, \epsilon_{1j}, u_j)'$ comes from a mean zero trivariate normal distribution with covariance matrix

$$\begin{bmatrix} \sigma_0^2 & \sigma_{01} & \sigma_0\rho_0 \\ \sigma_{01} & \sigma_1^2 & \sigma_1\rho_1 \\ \sigma_0\rho_0 & \sigma_1\rho_1 & 1 \end{bmatrix}$$

The covariance σ_{01} cannot be identified because we never observe both y_{1j} and y_{0j} . However, identification of σ_{01} is not necessary to estimate the other parameters because all covariates and the outcome are observed in observations from each group. We normalize the treatment error variance to be 1 because we observe only whether an outcome occurs under treatment. More details are found in [Methods and formulas](#).

Rather than showing two separate regression equations, `etregress` reports one outcome equation with interaction terms between the treatment and outcome covariates. `etregress` can fit the constrained and generalized potential-outcome models using either the default maximum likelihood estimator or the one-step CF estimator obtained with option `cfunction`. The two-step CF estimator provides consistent estimates for the constrained model.

Basic examples

When there are no interactions between the treatment variable and the outcome covariates in the constrained model, `etregress` directly estimates the ATE and the ATET.

► Example 1: Basic example

We estimate the ATE of being a union member on wages of women in 1972 from a nonrepresentative extract of the National Longitudinal Survey on young women who were ages 14–26 in 1968. We will use the variables `wage` (wage), `grade` (years of schooling completed), `smsa` (an indicator for living in an SMSA—standard metropolitan statistical area), `black` (an indicator for being African-American), `tenure` (tenure at current job), and `south` (an indicator for living in the South).

We use `etregress` to estimate the parameters of the endogenous treatment-regression model.

```
. use https://www.stata-press.com/data/r19/union3
(NLS Women 14-24 in 1968)

. etregress wage age grade smsa black tenure, treat(union = south black tenure)

Iteration 0:  Log likelihood =  -3140.811
Iteration 1:  Log likelihood = -3053.6629
Iteration 2:  Log likelihood = -3051.5847
Iteration 3:  Log likelihood = -3051.575
Iteration 4:  Log likelihood = -3051.575

Linear regression with endogenous treatment          Number of obs =  1,210
Estimator: Maximum likelihood                      Wald chi2(6)  = 681.89
Log likelihood = -3051.575                          Prob > chi2   = 0.0000
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
wage						
age	.1487409	.0193291	7.70	0.000	.1108566	.1866252
grade	.4205658	.0293577	14.33	0.000	.3630258	.4781058
smsa	.9117044	.1249041	7.30	0.000	.6668969	1.156512
black	-.7882471	.1367078	-5.77	0.000	-1.05619	-.5203048
tenure	.1524015	.0369596	4.12	0.000	.0799621	.2248409
1.union	2.945815	.2749621	10.71	0.000	2.4069	3.484731
_cons	-4.351572	.5283952	-8.24	0.000	-5.387208	-3.315936
union						
south	-.5807419	.0851111	-6.82	0.000	-.7475566	-.4139271
black	.4557499	.0958042	4.76	0.000	.2679771	.6435226
tenure	.0871536	.0232483	3.75	0.000	.0415878	.1327195
_cons	-.8855758	.0724506	-12.22	0.000	-1.027576	-.7435753
/athrho	-.6544347	.0910314	-7.19	0.000	-.832853	-.4760164
/lnsigma	.7026769	.0293372	23.95	0.000	.645177	.7601767
rho	-.5746478	.060971			-.682005	-.4430476
sigma	2.019151	.0592362			1.906325	2.138654
lambda	-1.1603	.1495097			-1.453334	-.8672668

LR test of indep. eqns. (rho = 0): chi2(1) = 19.84 Prob > chi2 = 0.0000

The likelihood-ratio test in the footer indicates that we can reject the null hypothesis of no correlation between the treatment-assignment errors and the outcome errors. The estimated correlation between the treatment-assignment errors and the outcome errors, ρ , is -0.575 . The negative relationship indicates that unobservables that raise observed wages tend to occur with unobservables that lower union membership. We discuss some details about this parameter in the technical note [below](#).

The estimated ATE of being a union member is 2.95. The ATET is the same as the ATE in this case because the treatment indicator variable has not been interacted with any of the outcome covariates, and the correlation and variance parameters are identical across the control and treatment groups.



□ Technical note

The results for the ancillary parameters ρ and σ require explanation. For numerical stability during optimization, `etregress` does not directly estimate ρ or σ . Instead, `etregress` estimates the inverse hyperbolic tangent of ρ ,

$$\operatorname{atanh} \rho = \frac{1}{2} \ln \left(\frac{1 + \rho}{1 - \rho} \right)$$

and $\ln \sigma$. Also `etregress` reports $\lambda = \rho \sigma$, along with an estimate of the standard error of the estimate and the confidence interval.

□

In contrast to the constrained model, `etregress` directly estimates the ATE only when there are no interactions between the treatment variable and the outcome covariates in the unconstrained model.

▷ Example 2: Allowing group-specific variance and correlation

We estimate the ATE of having health insurance on the natural logarithm of total out-of-pocket prescription drug expenditures from a simulated random sample of individuals between the ages of 25 and 64. We will use the variables `lndrug` (natural logarithm of spending on prescription drugs), `age` (age of the individual), `chron` (whether the individual has a chronic condition), `lninc` (natural logarithm of income), `married` (marriage status), and `work` (employment status). Our treatment is whether the person has health insurance, `ins`. We allow the outcome error variance and correlation parameters to vary between the treated (insured) and control (uninsured) groups in this example, rather than constraining them to be equal as in [example 1](#).

We use `etregress` to estimate the parameters of the endogenous treatment-effects model. To estimate separate variance and correlation parameters for each of the control and treatment groups, we specify the `poutcomes` option. We specify the `cfunction` option to use the CF estimator.

```
. use https://www.stata-press.com/data/r19/drugexp
(Prescription drug expenditures)
. etregress lndrug chron age lninc, treat(ins=age married lninc work) poutcomes
> cfunction
Iteration 0: GMM criterion Q(b) = 2.279e-15
Iteration 1: GMM criterion Q(b) = 1.842e-28
Linear regression with endogenous treatment          Number of obs = 6,000
Estimator: Control function
```

	Robust					
	Coefficient	std. err.	z	P> z	[95% conf. interval]	
lndrug						
chron	.4671725	.0319731	14.61	0.000	.4045064	.5298387
age	.1021359	.00292	34.98	0.000	.0964128	.1078589
lninc	.0550672	.0225036	2.45	0.014	.0109609	.0991735
1.ins	-.8598836	.3483648	-2.47	0.014	-1.542666	-.1771011
_cons	1.665539	.2527527	6.59	0.000	1.170153	2.160925
ins						
age	.021142	.0022961	9.21	0.000	.0166416	.0256424
married	.084631	.0359713	2.35	0.019	.0141286	.1551334
lninc	.1023032	.0225009	4.55	0.000	.0582022	.1464041
work	.288418	.0372281	7.75	0.000	.2154522	.3613837
_cons	-.622993	.108795	-5.73	0.000	-.8362273	-.4097587
/athrho0	.4035094	.1724539	2.34	0.019	.0655059	.7415129
/lnsigma0	.3159269	.0500476	6.31	0.000	.2178353	.4140184
/athrho1	.7929459	.2986601	2.66	0.008	.2075829	1.378309
/lnsigma1	.1865347	.0613124	3.04	0.002	.0663646	.3067048
rho0	.3829477	.1471637			.0654124	.6300583
sigma0	1.37153	.0686418			1.243382	1.512885
lambda0	.5252243	.226367			.0815532	.9688954
rho1	.6600746	.1685343			.2046518	.880572
sigma1	1.205066	.0738855			1.068616	1.35894
lambda1	.7954338	.2513036			.3028878	1.28798

Wald test of indep. (rho0 = rho1 = 0): chi2(2) = 8.88 Prob > chi2 = 0.0118

The Wald test reported in the footer indicates that we can reject the null hypothesis of no correlation between the treatment-assignment errors and the outcome errors for the control and treatment groups. The estimate of the correlation of the treatment-assignment errors for the control group (ρ_0) is positive, indicating that unobservables that increase spending on prescription drugs tend to occur with unobservables that increase health insurance coverage. Because ρ_1 is also positive, we make the same interpretation for individuals with insurance. The estimate ρ_1 is larger than the estimate ρ_0 , indicating a stronger relationship between the unobservables and treatment outcomes in the treated group.

The estimated ATE of having health insurance is -0.86 . Note that while the ATE and ATET were the same in [example 1](#), that is not the case here. We show how to calculate the ATET for a potential-outcome model in [example 6](#).

The estimate of the outcome error standard-deviation parameter for the control group (σ_0) is slightly larger than that of the treatment-group parameter (σ_1), indicating a greater variability in the unobservables among the untreated group.

Average treatment effect (ATE)

When there is a treatment variable and outcome covariate interaction, the parameter estimates from `etregress` can be used by `margins` to estimate the ATE, the average difference of the treatment potential outcomes and the control potential outcomes.

► Example 3: Allowing interactions between treatment and outcome covariates, ATE

In [example 1](#), the coefficients on the outcome covariates do not vary by treatment level. The differences in wages between union members and nonmembers are modeled as a level shift captured by the coefficient on the indicator for union membership. In this example, we use factor-variable notation to allow some of the coefficients to vary over treatment level and then use `margins` (see [\[R\] margins](#)) to estimate the ATE. (See [\[U\] 11.4.3 Factor variables](#) for an introduction to factor-variable notation.)

We begin by estimating the parameters of the model in which the coefficients on `black` and `tenure` differ for union members and nonmembers. We specify the `vce(robust)` option because we need to specify `vce(unconditional)` when we use `margins` below.

```
. use https://www.stata-press.com/data/r19/union3
(NLS Women 14-24 in 1968)

. etregress wage age grade smsa i.union#c.(black tenure),
> treat(union = south black tenure) vce(robust)

Iteration 0:  Log pseudolikelihood = -3614.6714
Iteration 1:  Log pseudolikelihood = -3218.8152
Iteration 2:  Log pseudolikelihood = -3057.0115
Iteration 3:  Log pseudolikelihood = -3049.3081
Iteration 4:  Log pseudolikelihood = -3049.2838
Iteration 5:  Log pseudolikelihood = -3049.2838

Linear regression with endogenous treatment
Estimator: Maximum likelihood
Log pseudolikelihood = -3049.2838

Number of obs = 1,210
Wald chi2(8) = 493.40
Prob > chi2 = 0.0000
```

	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
wage						
age	.1489075	.0207283	7.18	0.000	.1082809	.1895342
grade	.4200493	.0377621	11.12	0.000	.3460371	.4940616
smsa	.9232615	.1201486	7.68	0.000	.6877746	1.158748
union#						
c.black						
0	-.6685582	.1444213	-4.63	0.000	-.9516187	-.3854977
1	-1.1831	.2574817	-4.59	0.000	-1.687755	-.6784455
union#						
c.tenure						
0	.168746	.0503107	3.35	0.001	.0701388	.2673532
1	.0836367	.0903669	0.93	0.355	-.0934792	.2607526
1.union	3.342859	.5586863	5.98	0.000	2.247854	4.437864
_cons	-4.42566	.6493003	-6.82	0.000	-5.698265	-3.153055
union						
south	-.5844678	.0833069	-7.02	0.000	-.7477464	-.4211893
black	.4740688	.093241	5.08	0.000	.2913197	.6568178
tenure	.0874297	.0253892	3.44	0.001	.0376678	.1371916
_cons	-.8910484	.0746329	-11.94	0.000	-1.037326	-.7447706
/athrho	-.6733149	.2215328	-3.04	0.002	-1.107511	-.2391185
/lnsigma	.7055907	.0749711	9.41	0.000	.55865	.8525313
rho	-.5871562	.1451589			-.8031809	-.234663
sigma	2.025042	.1518197			1.748311	2.345577
lambda	-1.189016	.3631079			-1.900695	-.4773378

Wald test of indep. eqns. (rho = 0): chi2(1) = 9.24 Prob > chi2 = 0.0024

The results indicate that the coefficients on black differ by union membership and that the coefficient on tenure for nonmembers is positive, while the coefficient on tenure for members is 0. The model fits well overall, so we proceed with interpretation. Because we interacted the treatment variable with two of the covariates, the estimated coefficient on the treatment level is not an estimate of the ATE. Below we use margins to estimate the ATE from these results. We specify the `vce(unconditional)` option to obtain the standard errors for the population ATE instead of the sample ATE. We specify the `contrast(nowald)` option to suppress the Wald tests, which margins displays by default for contrasts.

```
. margins r.union, vce(unconditional) contrast(nowald)
Contrasts of predictive margins                                Number of obs = 1,210
Expression: Linear prediction, predict()
```

	Unconditional			
	Contrast	std. err.	[95% conf. interval]	
union (1 vs 0)	3.042688	.5305151	2.002898	4.082478

The ATE estimate is essentially the same as the one produced by the constrained model in [example 1](#). ◀

We can use the same methods above to obtain the ATE in an unconstrained model.

➤ Example 4: Treatment interactions and group-specific variance and correlation, ATE

In [example 2](#), the coefficients on the outcome covariates do not vary by treatment level. Suppose we believe that the effect of having a chronic condition on out-of-pocket spending differs between the insured and uninsured. Again, we use an interaction term. Because we are using a CF estimator, the variance–covariance of the estimator (VCE) is already robust so we do not specify `vce(robust)`.

```
. use https://www.stata-press.com/data/r19/drugexp
(Prescription drug expenditures)
. etregress lndrug i.ins#i.chron age lninc, treat(ins=age married lninc work)
> poutcomes cfunction
Iteration 0: GMM criterion Q(b) = 2.279e-15
Iteration 1: GMM criterion Q(b) = 2.883e-28
Linear regression with endogenous treatment          Number of obs = 6,000
Estimator: Control function
```

	Robust					
	Coefficient	std. err.	z	P> z	[95% conf. interval]	
lndrug						
ins#chron						
0 1	.3798705	.0720713	5.27	0.000	.2386132	.5211277
1 1	.4957773	.0352571	14.06	0.000	.4266746	.5648801
age	.1022045	.0029228	34.97	0.000	.0964758	.1079331
lninc	.0548917	.0225219	2.44	0.015	.0107497	.0990337
1.ins	-.89703	.3493058	-2.57	0.010	-1.581657	-.2124031
_cons	1.691336	.2531222	6.68	0.000	1.195225	2.187446
ins						
age	.021142	.0022961	9.21	0.000	.0166416	.0256424
married	.084631	.0359713	2.35	0.019	.0141286	.1551334
lninc	.1023032	.0225009	4.55	0.000	.0582022	.1464041
work	.288418	.0372281	7.75	0.000	.2154522	.3613837
_cons	-.622993	.108795	-5.73	0.000	-.8362273	-.4097587
/athrho0	.4046007	.1725597	2.34	0.019	.0663899	.7428115
/lnsigma0	.3157561	.0501956	6.29	0.000	.2173746	.4141376
/athrho1	.7950592	.2992825	2.66	0.008	.2084763	1.381642
/lnsigma1	.1868903	.0614281	3.04	0.002	.0664934	.3072871
rho0	.3838786	.1471308			.0662925	.6308408
sigma0	1.371296	.0688329			1.24281	1.513065
lambda0	.5264111	.2264197			.0826366	.9701856
rho1	.6612655	.1684146			.2055076	.8813184
sigma1	1.205495	.0740512			1.068754	1.359731
lambda1	.7971523	.2514293			.3043599	1.289945

Wald test of indep. (rho0 = rho1 = 0): chi2(2) = 8.90 Prob > chi2 = 0.0117

The results indicate that the coefficient on `chron` differs by whether an individual has insurance. The model fits well overall, so we proceed with interpretation.

Because we interacted the treatment variable with one of the covariates, the estimated coefficient on the treatment level is not an estimate of the ATE. Below we use `margins` to estimate the ATE from these results. We specify the `vce(unconditional)` option to obtain the standard errors for the population ATE instead of the sample ATE. We specify the `contrast(nowald)` option to suppress the Wald tests.

```
. margins r.ins, vce(unconditional) contrast(nowald)
Contrasts of predictive margins                                Number of obs = 6,000
Expression: Linear prediction, predict()
```

	Unconditional			
	Contrast	std. err.	[95% conf. interval]	
ins (1 vs 0)	-.8632045	.3484924	-1.546237	-.1801718

The ATE estimate is similar to the one produced by the constrained model in [example 2](#).



Average treatment effect on the treated (ATET)

When there is a treatment variable and outcome covariate interaction, the parameter estimates from `etregress` can be used by `margins` to estimate the ATET, the average difference of the treatment potential outcomes and the control potential outcomes on the treated population.

➤ Example 5: Allowing interactions between treatment and outcome covariates, ATET

The ATET may differ from the ATE in [example 3](#) because the interaction between the treatment variable and some outcome covariates makes the ATE and the ATET vary over outcome covariate values. Below we use `margins` to estimate the ATET by specifying the `subpop(union)` option, which restricts the sample used by `margins` to union members.

```
. use https://www.stata-press.com/data/r19/union3
(NLS Women 14-24 in 1968)
. etregress wage age grade smsa i.union#c.(black tenure),
> treat(union = south black tenure) vce(robust)
(output omitted)
. margins r.union, vce(unconditional) contrast(nowald) subpop(union)
Contrasts of predictive margins                                Number of obs   = 1,210
                                                                Subpop. no. obs = 253
Expression: Linear prediction, predict()
```

	Unconditional			
	Contrast	std. err.	[95% conf. interval]	
union (1 vs 0)	2.968977	.5358456	1.918739	4.019215

The estimated ATET and ATE are close, indicating that the average predicted outcome for the treatment group is similar to the average predicted outcome for the whole population.



➤ Example 6: Treatment interactions and group-specific variance and correlation, ATET

The ATET may differ from the ATE in [example 4](#) because the interaction between the treatment variable and some outcome covariates makes the ATE and the ATET vary over values of the covariate in the outcome equation. Even if there is no interaction between treatment assignment and a covariate in the outcome equation, the estimated ATE and ATET will differ if the variances of the outcome errors and their correlations with the treatment-assignment errors differ across the control and treatment groups.

We can estimate the ATET of having health insurance by using the conditional treatment effect (conditional on exogenous covariates and treatment level) obtained using the `predict`, `cte` and the `margins` commands; see [Methods and formulas](#) below and [\[CAUSAL\] etregress postestimation](#) for more details about the use of `predict` after `etregress`.

We restrict estimation to the treated subpopulation by specifying the `subpop(ins)` option with `margins`.

```
. use https://www.stata-press.com/data/r19/drugexp
(Prescription drug expenditures)
. etregress lndrug i.ins#i.chron age lninc,
> treat(ins = age married lninc work) poutcomes cfunction
(output omitted)
. margins, predict(cte) subpop(ins) vce(unconditional)
Predictive margins                                Number of obs   = 6,000
                                                    Subpop. no. obs = 4,556
Expression: Conditional treatment effect, predict(cte)
```

	Unconditional				
	Margin	std. err.	z	P> z	[95% conf. interval]
_cons	-.7558373	.3827579	-1.97	0.048	-1.506029 -.0056457

In absolute value, the treatment effect on the treated of -0.76 is smaller than the population average effect of -0.86 that we found in [example 4](#).



Stored results

`etregress` (maximum likelihood) stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_eq_model)</code>	number of equations in overall model test
<code>e(k_aux)</code>	number of auxiliary parameters
<code>e(k_dv)</code>	number of dependent variables
<code>e(df_m)</code>	model degrees of freedom
<code>e(ll)</code>	log likelihood
<code>e(ll_0)</code>	log likelihood, constant-only model (<code>lrmodel</code> only)
<code>e(N_clust)</code>	number of clusters
<code>e(lambda)</code>	estimate of λ in constrained model
<code>e(selambda)</code>	standard error of λ in constrained model
<code>e(sigma)</code>	estimate of σ in constrained model
<code>e(lambda0)</code>	estimate of λ_0 in potential-outcome model
<code>e(selambda0)</code>	standard error of λ_0 in potential-outcome model

e(sigma0)	estimate of σ_0 in potential-outcome model
e(lambda1)	estimate of λ_1 in potential-outcome model
e(selambda1)	standard error of λ_1 in potential-outcome model
e(sigma1)	estimate of σ_1 in potential-outcome model
e(chi2)	χ^2
e(chi2_c)	χ^2 for comparison test
e(p)	p -value for model test
e(p_c)	p -value for comparison test
e(rho)	estimate of ρ in constrained model
e(rho0)	estimate of ρ_0 in potential-outcome model
e(rho1)	estimate of ρ_1 in potential-outcome model
e(rank)	rank of $\mathbf{e}(V)$
e(rank0)	rank of $\mathbf{e}(V)$ for constant-only model
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	etregress
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(hazard)	variable containing hazard
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(title2)	secondary title in estimation output
e(clustvar)	name of cluster variable
e(chi2type)	Wald or LR; type of model χ^2 test
e(chi2_ct)	Wald or LR; type of model χ^2 test corresponding to e(chi2_c)
e(vce)	<i>vcetype</i> specified in vce()
e(vcetype)	title used to label Std. err.
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(method)	m1
e(ml_method)	type of m1 method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V
e(predict)	program used to implement predict
e(footnote)	program used to implement the footnote display
e(marginsok)	predictions allowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(Cns)	constraints matrix
e(ilog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance

Functions

e(sample)	marks estimation sample
-----------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices	
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

`etregress` (two-step) stores the following in `e()`:

Scalars	
<code>e(N)</code>	number of observations
<code>e(df_m)</code>	model degrees of freedom
<code>e(lambda)</code>	λ
<code>e(selambda)</code>	standard error of λ
<code>e(sigma)</code>	estimate of sigma
<code>e(chi2)</code>	χ^2
<code>e(p)</code>	p -value for model test
<code>e(rho)</code>	ρ
<code>e(rank)</code>	rank of $e(V)$
Macros	
<code>e(cmd)</code>	<code>etregress</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(hazard)</code>	variable containing hazard
<code>e(title)</code>	title in estimation output
<code>e(title2)</code>	secondary title in estimation output
<code>e(chi2type)</code>	Wald or LR; type of model χ^2 test
<code>e(vce)</code>	<code>vcetype</code> specified in <code>vce()</code>
<code>e(method)</code>	<code>twostep</code>
<code>e(properties)</code>	<code>b V</code>
<code>e(predict)</code>	program used to implement predict
<code>e(footnote)</code>	program used to implement the footnote display
<code>e(marginsok)</code>	predictions allowed by <code>margins</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>
Matrices	
<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators
Functions	
<code>e(sample)</code>	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

etregress (control-function) stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_aux)</code>	number of auxiliary parameters
<code>e(k_dv)</code>	number of dependent variables
<code>e(lambda)</code>	estimate of λ in constrained model
<code>e(selambda)</code>	standard error of λ in constrained model
<code>e(sigma)</code>	estimate of σ in constrained model
<code>e(lambda0)</code>	estimate of λ_0 in potential-outcome model
<code>e(selambda0)</code>	standard error of λ_0 in potential-outcome model
<code>e(sigma0)</code>	estimate of σ_0 in potential-outcome model
<code>e(lambda1)</code>	estimate of λ_1 in potential-outcome model
<code>e(selambda1)</code>	standard error of λ_1 in potential-outcome model
<code>e(sigma1)</code>	estimate of σ_1 in potential-outcome model
<code>e(chi2_c)</code>	χ^2 for comparison test
<code>e(p_c)</code>	p -value for comparison test
<code>e(rho)</code>	estimate of ρ in constrained model
<code>e(rho0)</code>	estimate of ρ_0 in potential-outcome model
<code>e(rho1)</code>	estimate of ρ_1 in potential-outcome model
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	etregress
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(hazard)</code>	variable containing hazard
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(title2)</code>	secondary title in estimation output
<code>e(chi2_ct)</code>	Wald; type of model χ^2 test corresponding to <code>e(chi2_c)</code>
<code>e(vce)</code>	<i>vce</i> type specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(method)</code>	cfunction
<code>e(properties)</code>	b V
<code>e(predict)</code>	program used to implement predict
<code>e(footnote)</code>	program used to implement the footnote display
<code>e(marginsok)</code>	predictions allowed by margins
<code>e(asbalanced)</code>	factor variables fvset as asbalanced
<code>e(asobserved)</code>	factor variables fvset as asobserved

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

Maddala (1983, 117–122 and 223–228) derives both the maximum likelihood and the CF estimators implemented here. Greene (2012, 890–894) also provides an introduction to the treatment-effects model. Cameron and Trivedi (2005, sections 16.7 and 25.3.4) and Wooldridge (2010, section 21.4.1) discuss the endogenous binary-variable model as an endogenous treatment-effects model and link it to recent work.

Methods and formulas are presented under the following headings:

Constrained model
General potential-outcome model
Average treatment effect
Average treatment effect on the treated

Constrained model

The primary regression equation of interest is

$$y_j = \mathbf{x}_j\boldsymbol{\beta} + \delta t_j + \epsilon_j \quad (1)$$

where t_j is a binary-treatment variable that is assumed to stem from an unobservable latent variable:

$$t_j^* = \mathbf{w}_j\boldsymbol{\gamma} + u_j$$

The decision to obtain the treatment is made according to the rule

$$t_j = \begin{cases} 1, & \text{if } t_j^* > 0 \\ 0, & \text{otherwise} \end{cases}$$

where ϵ and u are bivariate normal with mean zero and covariance matrix

$$\begin{bmatrix} \sigma^2 & \rho\sigma \\ \rho\sigma & 1 \end{bmatrix}$$

Interactions between $\mathbf{x}_j$ and the treatment t_j are also allowed in (1). The likelihood function for this model is given in Maddala (1983, 122). Greene (2000, 180) discusses the standard method of reducing a bivariate normal to a function of a univariate normal and the correlation ρ . The following is the log likelihood for observation j ,

$$\ln L_j = \begin{cases} \ln \Phi \left\{ \frac{\mathbf{w}_j\boldsymbol{\gamma} + (y_j - \mathbf{x}_j\boldsymbol{\beta} - \delta)\rho/\sigma}{\sqrt{1 - \rho^2}} \right\} - \frac{1}{2} \left(\frac{y_j - \mathbf{x}_j\boldsymbol{\beta} - \delta}{\sigma} \right)^2 - \ln(\sqrt{2\pi}\sigma) & t_j = 1 \\ \ln \Phi \left\{ \frac{-\mathbf{w}_j\boldsymbol{\gamma} - (y_j - \mathbf{x}_j\boldsymbol{\beta})\rho/\sigma}{\sqrt{1 - \rho^2}} \right\} - \frac{1}{2} \left(\frac{y_j - \mathbf{x}_j\boldsymbol{\beta}}{\sigma} \right)^2 - \ln(\sqrt{2\pi}\sigma) & t_j = 0 \end{cases}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

In the maximum likelihood estimation, σ and ρ are not directly estimated. Rather $\ln \sigma$ and $\operatorname{atanh} \rho$ are directly estimated, where

$$\operatorname{atanh} \rho = \frac{1}{2} \ln \left(\frac{1 + \rho}{1 - \rho} \right)$$

The standard error of $\lambda = \rho\sigma$ is approximated through the delta method, which is given by

$$\operatorname{Var}(\lambda) \approx \mathbf{D} \operatorname{Var}\{(\operatorname{atanh} \rho \quad \ln \sigma)\} \mathbf{D}'$$

where $\mathbf{D}$ is the Jacobian of λ with respect to $\operatorname{atanh} \rho$ and $\ln \sigma$.

Maddala (1983, 120–122) also derives the CF estimator as a two-step estimator. This estimator is implemented here. We will discuss it and then discuss the one-step CF estimator that is also implemented.

For the two-step estimator, probit estimates of the treatment equation

$$\Pr(t_j = 1 \mid \mathbf{w}_j) = \Phi(\mathbf{w}_j \boldsymbol{\gamma})$$

are obtained in the first stage. From these estimates, the hazard, h_j , for each observation j is computed as

$$h_j = \begin{cases} \phi(\mathbf{w}_j \hat{\boldsymbol{\gamma}}) / \Phi(\mathbf{w}_j \hat{\boldsymbol{\gamma}}) & t_j = 1 \\ -\phi(\mathbf{w}_j \hat{\boldsymbol{\gamma}}) / \{1 - \Phi(\mathbf{w}_j \hat{\boldsymbol{\gamma}})\} & t_j = 0 \end{cases}$$

where ϕ is the standard normal density function. If

$$d_j = h_j(h_j + \mathbf{w}_j \hat{\boldsymbol{\gamma}})$$

then

$$\begin{aligned} E(y_j \mid t_j, \mathbf{x}_j, \mathbf{w}_j) &= \mathbf{x}_j \boldsymbol{\beta} + \delta t_j + \rho \sigma h_j \\ \operatorname{Var}(y_j \mid t_j, \mathbf{x}_j, \mathbf{w}_j) &= \sigma^2 (1 - \rho^2 d_j) \end{aligned}$$

The two-step parameter estimates of $\boldsymbol{\beta}$ and δ are obtained by augmenting the regression equation with the hazard h . Thus the regressors become $[\mathbf{x} \ \mathbf{t} \ h]$, and the additional parameter estimate β_h is obtained on the variable containing the hazard. A consistent estimate of the regression disturbance variance is obtained using the residuals from the augmented regression and the parameter estimate on the hazard

$$\hat{\sigma}^2 = \frac{\mathbf{e}'\mathbf{e} + \beta_h^2 \sum_{j=1}^N d_j}{N}$$

The two-step estimate of ρ is then

$$\hat{\rho} = \frac{\beta_h}{\hat{\sigma}}$$

To understand how the consistent estimates of the coefficient covariance matrix based on the augmented regression are derived, let $\mathbf{A} = [\mathbf{x} \ \mathbf{t} \ h]$ and $\mathbf{D}$ be a square diagonal matrix of size N with $(1 - \hat{\rho}^2 d_j)$ on the diagonal elements. The conventional VCE is

$$\mathbf{V}_{\text{twostep}} = \hat{\sigma}^2 (\mathbf{A}'\mathbf{A})^{-1} (\mathbf{A}'\mathbf{D}\mathbf{A} + \mathbf{Q}) (\mathbf{A}'\mathbf{A})^{-1}$$

where

$$\mathbf{Q} = \hat{\rho}^2 (\mathbf{A}'\mathbf{D}\mathbf{A}) \mathbf{V}_p (\mathbf{A}'\mathbf{D}\mathbf{A})$$

and $\mathbf{V}_p$ is the variance–covariance estimate from the probit estimation of the treatment equation.

The one-step CF estimator is a GMM estimator with stacked moments. See [Newey \(1984\)](#) and [Wooldridge \(2010, sec. 14.2\)](#) for a description of this technique. Many econometric and statistical models can be expressed as conditions on the population moments. The parameter estimates produced by GMM estimators make the sample-moment conditions as true as possible given the data.

Under CF estimation, as in maximum likelihood estimation, we directly estimate $\text{atanh } \rho$ and $\ln \sigma$ rather than ρ and σ , so the parameter vector is

$$\boldsymbol{\theta} = (\boldsymbol{\beta}', \delta, \boldsymbol{\gamma}', \text{atanh } \rho, \ln \sigma)'$$

In this case, we have separate error functions for the treatment assignment

$$u_t(t_j, \mathbf{w}_j, \boldsymbol{\theta}) = \begin{cases} \phi(\mathbf{w}_j \boldsymbol{\gamma}) / \Phi(\mathbf{w}_j \boldsymbol{\gamma}) & t_j = 1 \\ -\phi(\mathbf{w}_j \boldsymbol{\gamma}) / \{1 - \Phi(\mathbf{w}_j \boldsymbol{\gamma})\} & t_j = 0 \end{cases}$$

for the outcome mean

$$u_m(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) = y_j - \mathbf{x}_j \boldsymbol{\beta} - \delta t_j - \rho \sigma u_{t,j}$$

and for the outcome variance

$$u_v(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) = u_{m,j}^2 - \sigma^2 [1 - \rho^2 \{u_{t,j}(u_{t,j} + \mathbf{w}_j \boldsymbol{\gamma})\}]$$

We calculate the hazard, h_j , prior to estimation from a probit regression of the treatment t_j on the treatment covariates $\mathbf{w}_j$. Let $\tilde{\mathbf{z}}_j = (\mathbf{x}_j, t_j, h_j)$. Now we define

$$\mathbf{Z}_j = \begin{bmatrix} \tilde{\mathbf{z}}_j & \mathbf{0} & 0 \\ 0 & \mathbf{w}_j & 0 \\ 0 & \mathbf{0} & 1 \end{bmatrix}$$

and

$$s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) = \mathbf{Z}_j' \begin{bmatrix} u_{m,j} \\ u_{t,j} \\ u_{v,j} \end{bmatrix}$$

The CF estimator $\hat{\boldsymbol{\theta}}$ is the value of $\boldsymbol{\theta}$ that satisfies the sample-moment conditions

$$\mathbf{0} = \frac{1}{N} \sum_i s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta})$$

The Huber/White/robust sandwich estimator is consistent for the VCE. See [Wooldridge \(2010, chap. 14\)](#), [Cameron and Trivedi \(2005, chap. 6\)](#), and [Newey and McFadden \(1994\)](#).

The formula is

$$\hat{\mathbf{V}} = (1/N) \bar{\mathbf{G}} \bar{\mathbf{S}} \bar{\mathbf{G}}'$$

where

$$\bar{\mathbf{G}} = \left\{ (1/N) \sum_i \frac{\partial s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\}^{-1}$$

and

$$\bar{\mathbf{S}} = (1/N) \sum_i s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta})'$$

The matrix $\overline{\mathbf{G}}$ is not symmetric because our estimator comes from stacking the moment conditions instead of optimizing one objective function. The implication is that the robust formula should always be used because, even under correct specification, the nonsymmetric $\overline{\mathbf{G}}$ and the symmetric $\overline{\mathbf{S}}$ converge to different matrices.

General potential-outcome model

Equation (1) can be generalized to a potential-outcome model with separate variance and correlation parameters for the control and treatment groups.

The generalized model is

$$\begin{aligned} y_{0j} &= \mathbf{x}_j \boldsymbol{\beta}_0 + \epsilon_{0j} \\ y_{1j} &= \mathbf{x}_j \boldsymbol{\beta}_1 + \epsilon_{1j} \\ t_j &= \begin{cases} 1, & \text{if } \mathbf{w}_j \boldsymbol{\gamma} + u_j > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

where y_{0j} is the outcome that person j obtains if person j selects treatment 0, and y_{1j} is the outcome that person j obtains if person j selects treatment 1. We never observe both y_{0j} and y_{1j} , only one or the other. We observe

$$y_j = t_j y_{1j} + (1 - t_j) y_{0j}$$

In this unconstrained model, the vector of error terms $(\epsilon_{0j}, \epsilon_{1j}, u_j)'$ comes from a mean zero trivariate normal distribution with covariance matrix

$$\begin{bmatrix} \sigma_0^2 & \sigma_{01} & \sigma_0 \rho_0 \\ \sigma_{01} & \sigma_1^2 & \sigma_1 \rho_1 \\ \sigma_0 \rho_0 & \sigma_1 \rho_1 & 1 \end{bmatrix}$$

The likelihood function for this model is given in [Maddala \(1983, 224\)](#).

$$\begin{aligned} \ln f_j &= \begin{cases} \ln \Phi \left\{ \frac{\mathbf{w}_j \boldsymbol{\gamma} + (y_j - \mathbf{x}_j \boldsymbol{\beta}_1) \rho_1 / \sigma_1}{\sqrt{1 - \rho_1^2}} \right\} - \frac{1}{2} \left(\frac{y_j - \mathbf{x}_j \boldsymbol{\beta}_1}{\sigma_1} \right)^2 - \ln(\sqrt{2\pi} \sigma_1), & t_j = 1 \\ \ln \Phi \left\{ \frac{-\mathbf{w}_j \boldsymbol{\gamma} - (y_j - \mathbf{x}_j \boldsymbol{\beta}_0) \rho_0 / \sigma_0}{\sqrt{1 - \rho_0^2}} \right\} - \frac{1}{2} \left(\frac{y_j - \mathbf{x}_j \boldsymbol{\beta}_0}{\sigma_0} \right)^2 - \ln(\sqrt{2\pi} \sigma_0), & t_j = 0 \end{cases} \\ \ln L &= \sum_{j=1}^n w_j \ln f_j \end{aligned}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution, and w_j is an optional weight. The covariance between ϵ_{0j} and ϵ_{1j} , σ_{01} , cannot be estimated because the potential outcomes y_{0j} and y_{1j} are never observed simultaneously.

As in the constrained model, σ_0 and σ_1 are not directly estimated in the maximum likelihood estimation; rather, $\ln \sigma_0$ and $\ln \sigma_1$ are estimated.

The parameters ρ_0 and ρ_1 are also not directly estimated; rather, $\text{atanh } \rho_0$ and $\text{atanh } \rho_1$ are directly estimated.

The new parameter vector is

$$\boldsymbol{\theta} = (\boldsymbol{\beta}'_0, \boldsymbol{\beta}'_1, \boldsymbol{\gamma}', \tanh \rho_0, \ln \sigma_0, \tanh \rho_1, \ln \sigma_1)'$$

The CF estimator for this potential-outcome model uses new error functions for the outcome mean

$$\begin{aligned} u_m(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) &= y_j - t_j(\mathbf{x}_j\boldsymbol{\beta}_1 + \rho_1\sigma_1u_{t,j}) \\ &\quad - (1 - t_j)(\mathbf{x}_j\boldsymbol{\beta}_0 + \rho_0\sigma_0u_{t,j}) \end{aligned}$$

and for the outcome variances

$$\begin{aligned} u_{v,0}(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) &= (1 - t_j) (u_{m,j}^2 - \sigma_0^2 [1 - \rho_0^2 \{u_{t,j}(u_{t,j} + \mathbf{w}_j\boldsymbol{\gamma})\}]) \\ u_{v,1}(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) &= t_j (u_{m,j}^2 - \sigma_1^2 [1 - \rho_1^2 \{u_{t,j}(u_{t,j} + \mathbf{w}_j\boldsymbol{\gamma})\}]) \end{aligned}$$

These error functions are derived based on the identities

$$\begin{aligned} E(y_j | t_j, \mathbf{x}_j, \mathbf{w}_j) &= t_j(\mathbf{x}_j\boldsymbol{\beta}_1 + \rho_1\sigma_1u_{t,j}) + (1 - t_j)(\mathbf{x}_j\boldsymbol{\beta}_0 + \rho_0\sigma_0u_{t,j}) \\ \text{Var}(y_j | t_j = 0, \mathbf{x}_j, \mathbf{w}_j) &= \sigma_0^2 [1 - \rho_0^2 \{u_{t,j}(u_{t,j} + \mathbf{w}_j\boldsymbol{\gamma})\}] \\ \text{Var}(y_j | t_j = 1, \mathbf{x}_j, \mathbf{w}_j) &= \sigma_1^2 [1 - \rho_1^2 \{u_{t,j}(u_{t,j} + \mathbf{w}_j\boldsymbol{\gamma})\}] \end{aligned}$$

We calculate the hazard, h_j , prior to estimation from a probit regression of the treatment, t_j , on the treatment covariates, $\mathbf{w}_j$. Let $\tilde{\mathbf{Z}}_j = \{\mathbf{x}_j, t_j h_j, (1 - t_j) h_j\}$. Now we define

$$\mathbf{Z}_j = \begin{bmatrix} \tilde{\mathbf{Z}}_j & \mathbf{0} & 0 & 0 \\ 0 & \mathbf{w}_j & 0 & 0 \\ 0 & \mathbf{0} & 1 & 0 \\ 0 & \mathbf{0} & 0 & 1 \end{bmatrix}$$

and

$$s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) = \mathbf{Z}'_j \begin{bmatrix} u_{m,j} \\ u_{t,j} \\ u_{v,0,j} \\ u_{v,1,j} \end{bmatrix}$$

The CF estimator $\hat{\boldsymbol{\theta}}$ is the value of $\boldsymbol{\theta}$ that satisfies the sample-moment conditions

$$\mathbf{0} = \frac{1}{N} \sum_i \mathbf{s}_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta})$$

The Huber/White/robust sandwich estimator is consistent for the VCE. See Wooldridge (2010, chap. 14), Cameron and Trivedi (2005, chap. 6), and Newey and McFadden (1994).

The formula is

$$\widehat{\mathbf{V}} = (1/N) \overline{\mathbf{G}} \overline{\mathbf{S}} \overline{\mathbf{G}}'$$

where

$$\overline{\mathbf{G}} = \left\{ (1/N) \sum_i \frac{\partial s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\}^{-1}$$

and

$$\overline{\mathbf{S}} = (1/N) \sum_i s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta}) s_j(y_j, t_j, \mathbf{x}_j, \mathbf{w}_j, \boldsymbol{\theta})'$$

The matrix $\overline{\mathbf{G}}$ is not symmetric because our estimator comes from stacking the moment conditions instead of optimizing one objective function. The implication is that the robust formula should always be used because, even under correct specification, the nonsymmetric $\overline{\mathbf{G}}$ and the symmetric $\overline{\mathbf{S}}$ converge to different matrices.

Average treatment effect

The ATE is the average difference of the treated potential outcomes and the control potential outcomes.

By the law of iterated expectations, the ATE is

$$\begin{aligned} E(y_{1j} - y_{0j}) &= E\{E(y_{1j} - y_{0j} | \mathbf{x}_j, \epsilon_{0j}, \epsilon_{1j})\} \\ &= E(\mathbf{x}_j \boldsymbol{\beta}_1 + \epsilon_1 - \mathbf{x}_j \boldsymbol{\beta}_0 - \epsilon_0) \\ &= E\{\mathbf{x}_j (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0)\} \end{aligned}$$

This expectation can be estimated as a predictive margin when $\mathbf{x}_j (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0)$ varies in $\mathbf{x}_j$. Otherwise, the ATE is estimated as the coefficient of t_j in the model.

Average treatment effect on the treated

The ATE is the average difference of the treated potential outcomes and the control potential outcomes on the treated population.

The conditional means of the potential outcomes y_{tj} , $t \in (0, 1)$ for exogenous covariates $\mathbf{x}_j$ and treatment covariates $\mathbf{w}_j$ at treatment $t_j = 1$ are

$$E(y_{tj} | \mathbf{x}_j, \mathbf{w}_j, t_j = 1) = \mathbf{x}_j \boldsymbol{\beta}_t + \rho_t \sigma_t \phi(\mathbf{w}_j \boldsymbol{\gamma}) / \Phi(\mathbf{w}_j \boldsymbol{\gamma})$$

By the law of iterated expectations, the ATET is

$$\begin{aligned} E(y_{1j} - y_{0j} | t_j = 1) &= E\{E(y_{1j} - y_{0j} | \mathbf{x}_j, \mathbf{w}_j, t_j = 1)\} \\ &= E\{\mathbf{x}_j (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0) + (\rho_1 \sigma_1 - \rho_0 \sigma_0) \phi(\mathbf{w}_j \boldsymbol{\gamma}) / \Phi(\mathbf{w}_j \boldsymbol{\gamma}) | t_j = 1\} \end{aligned}$$

This expectation can be estimated as a predictive margin on the treated population when $\mathbf{x}_j (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_0)$ varies in $\mathbf{x}_j$ or when the variance and correlation parameters differ by treatment group. Otherwise, the ATET is estimated as the coefficient of t_j in the model.

References

- Barnow, B. S., G. G. Cain, and A. S. Goldberger. 1981. “Issues in the analysis of selectivity bias”. In *Evaluation Studies Review Annual*, edited by E. W. Stromsdorfer and G. Farkas, vol. 5: 123–126. Beverly Hills: Sage.
- Brave, S., and T. Walstrum. 2014. [Estimating marginal treatment effects using parametric and semiparametric methods](#). *Stata Journal* 14: 191–217.
- Brown, G. K., and T. Mergoupis. 2011. [Treatment interactions with nonexperimental data in Stata](#). *Stata Journal* 11: 545–555.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cerulli, G. 2015. [ctreatreg: Command for fitting dose-response models under exogenous and endogenous treatment](#). *Stata Journal* 15: 1019–1045.
- . 2022. *Econometric Evaluation of Socio-Economic Programs: Theory and Applications*. 2nd ed. Berlin: Springer.
- Greene, W. H. 2000. *Econometric Analysis*. 4th ed. Upper Saddle River, NJ: Prentice Hall.
- . 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.
- Heckman, J. J. 1976. “The common structure of statistical models of truncation, sample selection and limited dependent variables and a simple estimator for such models”. In *Annals of Economic and Social Measurement*, edited by S. V. Berg, vol. 5: 475–492. Cambridge, MA: National Bureau of Economic Research.
- . 1978. Dummy endogenous variables in a simultaneous equation system. *Econometrica* 46: 931–959. <https://doi.org/10.2307/1909757>.
- Maddala, G. S. 1983. *Limited-Dependent and Qualitative Variables in Econometrics*. Cambridge: Cambridge University Press. <https://doi.org/10.1017/CBO9780511810176>.
- Nannicini, T. 2007. [Simulation-based sensitivity analysis for matching estimators](#). *Stata Journal* 7: 334–350.
- Newey, W. K. 1984. A method of moments interpretation of sequential estimators. *Economics Letters* 14: 201–206. [https://doi.org/10.1016/0165-1765\(84\)90083-1](https://doi.org/10.1016/0165-1765(84)90083-1).
- Newey, W. K., and D. L. McFadden. 1994. “Large sample estimation and hypothesis testing”. In *Handbook of Econometrics*, edited by R. F. Engle and D. L. McFadden, vol. 4: 2111–2245. Amsterdam: Elsevier. [https://doi.org/10.1016/S1573-4412\(05\)80005-4](https://doi.org/10.1016/S1573-4412(05)80005-4).
- Nichols, A. 2007. [Causal inference with observational data](#). *Stata Journal* 7: 507–541.
- Vella, F., and M. Verbeek. 1998. Whose wages do unions raise? A dynamic model of unionism and wage rate determination for young men. *Journal of Applied Econometrics* 13: 163–183. [https://doi.org/10.1002/\(SICI\)1099-1255\(199803/04\)13:2<163::AID-JAE460>3.0.CO;2-Y](https://doi.org/10.1002/(SICI)1099-1255(199803/04)13:2<163::AID-JAE460>3.0.CO;2-Y).
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

- [CAUSAL] [etregress postestimation](#) — Postestimation tools for etregress
- [CAUSAL] [etpoisson](#) — Poisson regression with endogenous treatment effects
- [ERM] [eregress](#) — Extended linear regression
- [R] [heckman](#) — Heckman selection model
- [R] [probit](#) — Probit regression
- [R] [regress](#) — Linear regression⁺
- [SVY] [svy estimation](#) — Estimation commands for survey data
- [U] [20 Estimation and postestimation commands](#)

[Postestimation commands](#)
[predict](#)
[margins](#)
[Remarks and examples](#)
[Also see](#)

Postestimation commands

The following standard postestimation commands are available after `etregress`:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of parameters
* <code>estat ic</code>	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estat (svy)</code>	postestimation statistics for survey data
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
* <code>hausman</code>	Hausman's specification test
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of parameters
* <code>lrtest</code>	likelihood-ratio test
<code>margins</code>	marginal means, predictive margins, marginal effects, and average marginal effects
<code>marginsplot</code>	graph the results from margins (profile plots, interaction plots, etc.)
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
<code>predict</code>	conditional treatment effects, linear predictions and their SEs, etc.
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of parameters
* <code>suest</code>	seemingly unrelated estimation
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

*`estat ic`, `lrtest`, and `suest` are not appropriate after `etregress`, `twostep` or `etregress`, `cfunction`.
`hausman` and `lrtest` are not appropriate with `svy` estimation results.

predict

Description for predict

predict creates a new variable containing predictions such as linear predictions, conditional treatment effects, standard errors, expected values, and probabilities.

Menu for predict

Statistics > Postestimation

Syntax for predict

After ML, twostep, or cfunction

```
predict [type] newvar [if] [in] [, statistic]
```

After ML or cfunction

```
predict [type] stub* [if] [in], scores
```

statistic	Description
Main	
xb	linear prediction; the default
cte	conditional treatment effect at treatment level
stdp	standard error of the prediction
stdf	standard error of the forecast
yctrtrt	$E(y_j \mid \text{treatment} = 1)$
ycntrtrt	$E(y_j \mid \text{treatment} = 0)$
ptrt	$\text{Pr}(\text{treatment} = 1)$
xbtrtrt	linear prediction for treatment equation
stdptrtrt	standard error of the linear prediction for treatment equation

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.
stdf is not allowed with svy estimation results.

Options for predict

Main

- xb, the default, calculates the linear prediction, $\mathbf{x}_j\mathbf{b}$.
- cte calculates the treatment effect, the difference of potential-outcome means, conditioned on treatment level.
- stdp calculates the standard error of the prediction, which can be thought of as the standard error of the predicted expected value or mean for the observation’s covariate pattern. The standard error of the prediction is also referred to as the standard error of the fitted value.

`stdf` calculates the standard error of the forecast, which is the standard error of the point prediction for one observation. It is commonly referred to as the standard error of the future or forecast value. By construction, the standard errors produced by `stdf` are always larger than those produced by `stdp`; see *Methods and formulas* in [R] [regress postestimation](#).

`yctrtr` calculates the expected value of the dependent variable conditional on the presence of the treatment: $E(y_j \mid \text{treatment} = 1)$.

`ycntrtr` calculates the expected value of the dependent variable conditional on the absence of the treatment: $E(y_j \mid \text{treatment} = 0)$.

`ptrtr` calculates the probability of the presence of the treatment:

$$\Pr(\text{treatment} = 1) = \Pr(\mathbf{w}_j\boldsymbol{\gamma} + u_j > 0).$$

`xbtrtr` calculates the linear prediction for the treatment equation.

`stdptrtr` calculates the standard error of the linear prediction for the treatment equation.

`scores`, not available with `twostep`, calculates equation-level score variables.

The first new variable will contain $\partial \ln L / \partial (\mathbf{x}_j \boldsymbol{\beta})$.

The second new variable will contain $\partial \ln L / \partial (\mathbf{w}_j \boldsymbol{\gamma})$.

Under the constrained model, the third new variable will contain $\partial \ln L / \partial \tanh \rho$.

Under the constrained model, the fourth new variable will contain $\partial \ln L / \partial \ln \sigma$.

Under the general potential-outcome model, the third new variable will contain $\partial \ln L / \partial \tanh \rho_0$.

Under the general potential-outcome model, the fourth new variable will contain $\partial \ln L / \partial \ln \sigma_0$.

Under the general potential-outcome model, the fifth new variable will contain $\partial \ln L / \partial \tanh \rho_1$.

Under the general potential-outcome model, the sixth new variable will contain $\partial \ln L / \partial \ln \sigma_1$.

margins

Description for margins

margins estimates margins of response for linear predictions, conditional treatment effects, expected values, and probabilities.

Menu for margins

Statistics > Postestimation

Syntax for margins

```
margins [marginlist] [ , options ]
margins [marginlist] , predict(statistic ...) [predict(statistic ...) ...] [options]
```

Maximum likelihood and control-function estimation results

statistic	Description
xb	linear prediction; the default
cte	conditional treatment effect at treatment level
<u>y</u> ctr t	$E(y_j \mid \text{treatment} = 1)$
<u>y</u> cn t r t	$E(y_j \mid \text{treatment} = 0)$
<u>p</u> tr t	$\text{Pr}(\text{treatment} = 1)$
<u>x</u> btr t	linear prediction for treatment equation
stdp	not allowed with margins
stdf	not allowed with margins
<u>s</u> td p tr t	not allowed with margins

Two-step estimation results

statistic	Description
xb	linear prediction; the default
<u>p</u> tr t	$\text{Pr}(\text{treatment} = 1)$
<u>x</u> btr t	linear prediction for treatment equation
cte	not allowed with margins
<u>y</u> ctr t	not allowed with margins
<u>y</u> cn t r t	not allowed with margins
stdp	not allowed with margins
stdf	not allowed with margins
<u>s</u> td p tr t	not allowed with margins

Statistics not allowed with margins are functions of stochastic quantities other than $e(b)$.

For the full syntax, see [R] margins.

Remarks and examples

The average treatment effect (ATE) and the average treatment effect on the treated (ATET) are the parameters most frequently estimated by postestimation techniques after `etregress`.

When there are no interactions between the treatment variable and the outcome covariates in the constrained model, `etregress` directly estimates the ATE and the ATET; see [example 1](#) of [\[CAUSAL\] etregress](#).

When there are no interactions between the treatment variable and the outcome covariates in the general potential-outcome model, `etregress` directly estimates the ATE; see [example 2](#) of [\[CAUSAL\] etregress](#).

When there are interactions between the treatment variable and the outcome covariates, you can use `margins` after `etregress` to estimate the ATE. See [example 3](#) and [example 4](#) of [\[CAUSAL\] etregress](#) for examples of ATE estimation.

When there are interactions between the treatment variable and the outcome covariates in the constrained model, you can use `margins` after `etregress` to estimate the ATET. See [example 5](#) of [\[CAUSAL\] etregress](#) for an example of ATET estimation in the constrained model.

In the general potential-outcome model, you can use `margins` after `etregress` to estimate the ATET. See [example 6](#) of [\[CAUSAL\] etregress](#) for an example of ATET estimation in the general potential-outcome model.

Also see

[\[CAUSAL\] etregress](#) — Linear regression with endogenous treatment effects

[\[U\] 20 Estimation and postestimation commands](#)

Description

`gencohort` creates a categorical variable indicating the first time period at which units within a group are treated, that is, a cohort variable. Cohort variables created by `gencohort` are meant to be used as inputs for the `hdidregress` and `xthdidregress` commands.

Quick start

Create a cohort variable with name `mycohort` using group variable `country`, time variable `year`, and binary treatment variable `policy`

```
gencohort mycohort, group(country) time(year) treat(policy)
```

Same as above, but only if outcome variable `y` is positive

```
gencohort mycohort if y>0, group(country) time(year) treat(policy)
```

Menu

Data > Create or change data > Other variable-creation commands > Create cohort variable

Syntax

```
gencohort newvar [if] [in], group(groupvar) time(timevar) treat(tvar)
```

groupvar is a categorical variable that indicates the group level at which the treatment occurs.

timevar is a time variable.

tvar must be a binary variable indicating observations subject to treatment.

Options

Main

`group(groupvar)` specifies a group variable that indicates the group level at which the treatment occurs. *groupvar* may be, for example, states, counties, or hospitals. `group()` is required.

`time(timevar)` specifies the time variable used to define treatment-time cohorts. `time()` is required.

`treat(tvar)` specifies the binary variable that indicates which observations are subject to treatment. `treat()` is required.

Remarks and examples

For examples, see *Remarks and examples* in [CAUSAL] `hdidregress` and *Remarks and examples* in [CAUSAL] `xthdidregress`.

Also see

[CAUSAL] [hdidregress](#) — Heterogeneous difference in differences⁺

[CAUSAL] [xthdidregress](#) — Heterogeneous difference in differences for panel data⁺

⁺This command includes features that are part of [StataNow](#).[Description](#)[Options](#)[Acknowledgments](#)[Quick start](#)[Remarks and examples](#)[References](#)[Menu](#)[Stored results](#)[Also see](#)[Syntax](#)[Methods and formulas](#)

Description

hdiidregress estimates average treatment effects on the treated (ATETs) that may vary over time and over treatment cohorts. Treatment cohorts are groups subject to treatment at different points in time. **hdiidregress** provides four estimators: extended two-way fixed effects (TWFE), regression adjustment (RA), inverse-probability weighting (IPW), and augmented inverse-probability weighting (AIPW). See [\[CAUSAL\] teffects intro](#) for a discussion of RA, IPW, and AIPW estimators.

hdiidregress is for repeated cross-sectional data. For panel data, see [\[CAUSAL\] xthdiidregress](#).

Quick start

Estimate ATETs of treatment **treat** on outcome **y** with group **grpvar** and time **tvar**; use the RA estimator and model **y** using covariate **x**

```
hdiidregress ra (y x) (treat), group(grpvar) time(tvar)
```

Same as above, but use the TWFE estimator

```
hdiidregress twfe (y x) (treat), group(grpvar) time(tvar)
```

Use the IPW estimator and model **treat** using covariate **z**

```
hdiidregress ipw (y) (treat z), group(grpvar) time(tvar)
```

Use the AIPW estimator, model **y** using covariate **x**, and model **treat** using covariate **z**

```
hdiidregress aipw (y x) (treat z), group(grpvar) time(tvar)
```

Same as above, but use the not-yet-treated group as the control group

```
hdiidregress aipw (y x) (treat z), group(grpvar) time(tvar) ///
    controlgroup(notyet)
```

Same as above, but cluster at the county level

```
hdiidregress aipw (y x) (treat z), group(grpvar) time(tvar) ///
    controlgroup(notyet) vce(cluster county)
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Heterogeneous DID > Heterogeneous DID (TWFE)

Statistics > Causal inference/treatment effects > Continuous outcomes > Heterogeneous DID > Heterogeneous DID (RA)

Statistics > Causal inference/treatment effects > Continuous outcomes > Heterogeneous DID > Heterogeneous DID (IPW)

Statistics > Causal inference/treatment effects > Continuous outcomes > Heterogeneous DID > Heterogeneous DID (AIPW)

Syntax

Two-way fixed effects

```
hdidregress twfe (ovar [omvarlist]) (tvar) [if] [in] [weight],
    group(groupvar) time(timevar) [options]
```

Regression adjustment

```
hdidregress ra (ovar [omvarlist]) (tvar) [if] [in] [weight],
    group(groupvar) time(timevar) [options]
```

Inverse-probability weighting

```
hdidregress ipw (ovar) (tvar [tmvarlist]) [if] [in] [weight],
    group(groupvar) time(timevar) [options]
```

Augmented inverse-probability weighting

```
hdidregress aipw (ovar [omvarlist]) (tvar [tmvarlist]) [if] [in] [weight],
    group(groupvar) time(timevar) [options]
```

ovar is a continuous outcome of interest.

omvarlist specifies the covariates in the outcome model and may contain factor variables; see [\[U\] 11.4.3 Factor variables](#).

tvar must be a binary variable indicating observations subject to treatment.

tmvarlist specifies the covariates in the treatment model and may contain factor variables; see [\[U\] 11.4.3 Factor variables](#).

groupvar is a categorical variable that indicates the group level at which the treatment occurs.

timevar is a time variable.

<i>options</i>	Description
Model	
* <u>group</u> (<i>groupvar</i>)	specify group variable
* <u>time</u> (<i>timevar</i>)	specify time variable
<u>controlgroup</u> (<i>cgtype</i>)	specify the type of control group; default is <code>controlgroup(never)</code>
<u>cohortvar</u> (<i>cvar</i> [, <code>replace</code>])	specify the variable name for the generated cohort
<u>usercohort</u> (<i>varname</i>)	specify name of cohort variable to be used for estimation
† <u>basetime</u> (<i>btspec</i>)	specify the type of base time for pretreatment periods; default is <code>basetime(adaptive)</code>
‡ <u>hettype</u> (<i>hetspec</i>)	specify the type of heterogeneity; default is <code>hettype(timecohort)</code>
+‡ <u>over</u> (<i>overspec</i>)	specify the subpopulation over which each ATET is computed; default is <code>over(cohort)</code>
+‡ <u>noxinteract</u>	omit covariate interactions
SE/Robust	
<u>vce</u> (<i>vcetype</i>)	<i>vcetype</i> may be <code>cluster</code> <i>clustvar</i> , <code>robust</code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<u>level</u> (#)	set confidence level; default is <code>level(95)</code>
[<code>no</code>] <code>log</code>	suppress iteration log
<code>nodots</code>	suppress replication dots
<i>display_options</i>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
<u>coeflegend</u>	display legend instead of statistics
<i>cgtype</i>	Description
<code>never</code>	use the never-treated group as the control group; the default
<code>notyet</code>	use the not-yet-treated group as the control group
<i>btspec</i>	Description
<code>adaptive</code>	specify the adaptive base time for pretreatment ATETs; the default
<code>common</code>	specify a common base time for all pretreatment ATETs
<i>hetspec</i>	Description
<code>timecohort</code>	heterogeneous treatment effects over time and cohort; the default
<code>time</code>	heterogeneous treatment effects over time
<code>cohort</code>	heterogeneous treatment effects over cohort

<i>overspec</i>	Description
cohort	use cohort subpopulations when controlling for covariate heterogeneity; the default
timecohort	use time and cohort subpopulations when controlling for covariate heterogeneity

⁺These features are part of [StataNow](#).

*`group()` and `time()` are required.

[†]`basetime()` may be specified only when `ra`, `ipw`, or `aipw` is specified.

[‡]`hettype()` and `noxinteract` may be specified only when `twfe` is also specified. `over()` may be specified only when `twfe` and `hettype(timecohort)` are also specified.

`by`, `collect`, and `statsby` are allowed; see [\[U\] 11.1.10 Prefix commands](#).

`fweights`, `awweights`, and `pweights` are allowed; see [\[U\] 11.1.6 weight](#).

`coeflegend` does not appear in the dialog box.

See [\[U\] 20 Estimation and postestimation commands](#) for more capabilities of estimation commands.

Options

Model

`group(groupvar)` specifies a group variable that indicates the group level at which the treatment occurs. *groupvar* may be, for example, states, counties, or hospitals. `group()` also defines the clusters for the default cluster-robust standard errors. `group()` is required. You may specify only one group variable.

`time(timevar)` specifies the time variable used to define treatment-time cohorts. `time()` is required.

`controlgroup(cgtype)` specifies the type of control group. A control group can be either a never-treated group or a not-yet-treated group. A never-treated group refers to the units that are untreated from the first to the last period. A not-yet-treated group refers to the units that are untreated up to a specific period. *cgtype* can be one of `never`, referring to the never-treated group, or `notyet`, referring to the not-yet-treated group. By default, *cgtype* is `never`.

`cohortvar(cvar [ , replace])` specifies the variable name *cvar* for the generated cohort variable. The cohort variable is a categorical variable indicating the period when the unit is first treated. By default, `_did_cohort` is used as the name of the cohort variable. If `_did_cohort` already exists in the dataset, it is replaced if option `cohortvar()` is not specified.

If suboption `replace` is specified, *cvar* is replaced.

`usercohort(varname)` specifies a variable to be used as a cohort indicator during estimation. By default, a cohort variable is generated using the information in the estimation sample to indicate the period when a unit is first treated. `usercohort()` overrides this default and allows you to provide a cohort indicator. This is useful, for instance, when there are gaps in the estimation sample, but you know a group was treated at the time when the gap is present in the data.

`basetime(btspec)` specifies how the base time is chosen when computing the pretreatment ATETs with the `ra`, `ipw`, or `aipw` estimator. *btspec* is one of `adaptive` (the default) or `common`.

`adaptive` specifies that the base time for pretreatment ATETs be chosen adaptively. The base time for each pretreatment period *t* for cohort *g* is the previous period, *t* − 1.

common specifies that a common base time of $g - 1$ be used for all pretreatment ATETs for cohort g . A long-run violation of the parallel trends assumption is easier to identify when using this common base time.

The base time for posttreatment periods is $g - 1$, regardless of whether the adaptive or common base time is used for pretreatment periods.

`hettype(hetspec)` specifies time or cohort heterogeneity for the `twfe` estimator. By default, treatment is interacted with time and cohort. You may choose to keep one of time or cohort interactions using `hetspec`.

`hetspec` may be one of `timecohort` for heterogeneous treatment effects over both time and cohort, `time` for heterogeneous treatment effects over time only, or `cohort` for heterogeneous treatment effects over cohort only. By default, `hetspec` is `timecohort`.

`over(overspec)` is part of StataNow. It specifies the subpopulation over which treatment effects are averaged for the `twfe` estimator when the heterogeneity type is specified as `timecohort` (`hettype(timecohort)`). When covariates are specified, we need to control for their variation when computing treatment effects. `overspec` may be one of `cohort`, for effects averaged over each cohort, or `timecohort`, for effects averaged over each time period and cohort. The default is `over(cohort)`, meaning effects are averaged over each cohort and adjusted for the time period effect. If there are no covariates, this option has no effect.

`over()` is not allowed when `hettype()` is specified as `time` or `cohort`.

`noxinteract` is part of StataNow. It specifies that the covariates in `omvarlist` not be interacted with the difference-in-difference regressors in the TWFE model. Covariates are still included in the model as levels.

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that allow for intra-group correlation (`cluster clustvar`), that are robust to intragroup correlation among group variable (`robust`), and that use bootstrap or jackknife sampling done at the individual level (`bootstrap`, `jackknife`); see [R] [vce_option](#).

`vce(cluster clustvar)`, the default, uses the variable specified in `group(groupvar)`.

Reporting

`level(#)`; see [R] [Estimation options](#).

`log` and `nolog` specify whether to display the iteration log. The iteration log is displayed by default unless you used `set iterlog off` to suppress it; see `set iterlog` in [R] [set iter](#).

`nodots` suppresses display of the replication dots.

`display_options:` `nocl`, `nopvalues`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

The following option is available with `hdidregress` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

It is common to study the effects of a treatment, for example, a policy or intervention, on a group. `hdidregress` is for data where the treated groups are subject to the treatment at different points in time and they remain exposed to the treatment. For example, a health policy such as an increase in the age to purchase cigarettes is implemented in a given region, and over time, other regions decide to imitate the initiative. Another example is change in work policies across industries. Perhaps airlines implement a minimum number of hours between shifts for safety reasons. The policy is subsequently adopted by other similar industries. Some similar industries may never adopt the policy, remaining untreated, or it might be that all similar industries eventually adopt the policy.

`hdidregress` estimates ATET parameters that change over time and treatment cohorts (groups treated at different points in time). Each one of these ATETs has the same interpretation that the parameters of a two-time two-group difference-in-differences (DID) parameter would have. Because there are multiple DID parameters, we refer to them as heterogeneous treatment effects or as heterogeneous DID. This is in contrast to estimating only one ATET, which assumes there is no variation across time or cohort. If you assume no variation across time or cohort, you may use `didregress`; see [\[CAUSAL\] didregress](#).

`hdidregress` provides four estimators: TWFE, outlined in [Wooldridge \(2021\)](#); RA, IPW, and AIPW, outlined in [Callaway and Sant’Anna \(2021\)](#). Each one of these estimators fits a model for the outcome of interest, a model for the treatment, or a model for both. For example, RA and TWFE model the outcome; IPW models the treatment; and AIPW models both. If the model for the outcome is correctly specified, RA and TWFE are best, with TWFE being more efficient. If the treatment model is correctly specified, IPW should be best. AIPW models both treatment and outcome. If at least one of the models is correctly specified, it provides consistent estimates. Thus, it allows us to misspecify one of the models and still get consistent estimates, a property called double robustness. See [\[CAUSAL\] teffects intro](#) for a discussion of RA, IPW, and AIPW estimators.

`hdidregress` is for repeated cross-sectional data. For panel data, see [\[CAUSAL\] xthdidregress](#). Below, we illustrate how to use `hdidregress`. For a general overview of DID and more information about the methods used below, see [\[CAUSAL\] DID intro](#). For general discussions about the methods, see [Roth et al. \(2022\)](#) and [de Chaisemartin and D’Haultfœuille \(2023\)](#) and the references therein.

► Example 1: Heterogeneous DID for repeated cross-sections

We are interested in knowing if a school-district-level program, Healthy Habits, reduces the body mass index (BMI) for students in the school district. We have fictional data on the Healthy Habits program. This program incorporates more exercise time and augments the intakes of fruits and vegetables. Our data are at the school-district level and include information on whether a school participates in the program, `hhabit`, and the BMI of students in the district, `bmi`. We have repeated samples of students ages 11 to 14 from 40 school districts from the year 2032 to the year 2040.

```
. use https://www.stata-press.com/data/r19/hhabits
(Fictional children BMI and school district data)
```

We are going to use the `aipw` estimator, which allows us to model the outcome and the treatment. If we had selected another estimator and specified the outcome incorrectly, the treatment effects would be inconsistent; see [\[CAUSAL\] teffects aipw](#). With the `aipw` estimator, as long as one of the treatment or outcome model is correctly specified, we will get a consistent estimate of the ATET—a property called double robustness.

We model `hhabit` using the number of parks in the district, `parksd`. We conjecture that school districts with more parks consider exercise spaces more important in their urban planning than those with fewer parks. These districts are therefore more amenable to the Healthy Habits program.

For the outcome variable, we believe that mother’s education, `medu`, is a good predictor of the health habits of children. We also believe that participation in sports, `sports`, affects `bmi`. Finally, we control for whether the student is a `girl` to account for behavioral differences and differences in body types of boys and girls at this age.

In the first set of parentheses, we define the outcome, `bmi`, and any covariates that affect the outcome directly. In the second set of parentheses, we define the observation-level treatment variable, `hhabit`, and the covariates that affect it. After the comma, we must define the group variable in `group()`; this is a required option. The group variable defines at which level the treatment occurs and also identifies the clustering variable, which in this case is `schools`. We also need to specify a time variable in `time()`. We fit the following model:

```
. hdidregress aipw (bmi medu i.girl i.sports) (hhabit parksd),
> group(schools) time(year)
note: variable _did_cohort, containing cohort indicators formed by treatment
      variable hhabit and group variable schools, was added to the dataset
      using the estimation sample.
```

Computing ATET for each cohort and time:

Cohort 2034 (8): done

Cohort 2036 (8): done

Cohort 2038 (8): done

Treatment and time information

Time variable: year

Time interval: 2032 to 2040

Control: `_did_cohort = 0`

Treatment: `_did_cohort > 0`

	<code>_did_cohort</code>
Number of cohorts	4
Number of obs	
Never treated	11355
2034	1231
2036	2097
2038	2042

Heterogeneous treatment-effects regression Number of obs = 16,725
 Estimator: Augmented IPW
 Treatment level: schools
 Control group: Never treated

(Std. err. adjusted for 40 clusters in schools)

Cohort	ATET	Robust std. err.	z	P> z	[95% conf. interval]	
2034						
year						
2033	.6544681	.5946048	1.10	0.271	-.5109359	1.819872
2034	-1.226451	.379168	-3.23	0.001	-1.969607	-.4832957
2035	-2.491842	.4169657	-5.98	0.000	-3.30908	-1.674605
2036	-2.72486	.2363878	-11.53	0.000	-3.188171	-2.261548
2037	-2.786634	.6672867	-4.18	0.000	-4.094492	-1.478776
2038	-3.980456	.2993279	-13.30	0.000	-4.567127	-3.393784
2039	-.604415	.5929199	-1.02	0.308	-1.766517	.5576866
2040	-.6522272	.3640416	-1.79	0.073	-1.365736	.0612812
2036						
year						
2033	.6635794	.3089663	2.15	0.032	.0580167	1.269142
2034	-1.3933	.3871204	-3.60	0.000	-2.152042	-.6345582
2035	.5947865	.4065947	1.46	0.144	-.2021245	1.391697
2036	-1.71427	.4565384	-3.75	0.000	-2.609069	-.8194714
2037	-3.170542	.5221368	-6.07	0.000	-4.193912	-2.147173
2038	-2.967701	.4247053	-6.99	0.000	-3.800108	-2.135294
2039	.0360098	.6868764	0.05	0.958	-1.310243	1.382263
2040	-.957117	.3510986	-2.73	0.006	-1.645258	-.2689763
2038						
year						
2033	-1.434451	.5163232	-2.78	0.005	-2.446426	-.422476
2034	1.010288	.4808165	2.10	0.036	.067905	1.952671
2035	-.3809733	.4336764	-0.88	0.380	-1.230963	.4690169
2036	.5199519	.4849723	1.07	0.284	-.4305763	1.47048
2037	-.0315794	.5863875	-0.05	0.957	-1.180878	1.117719
2038	-3.602114	.3498692	-10.30	0.000	-4.287845	-2.916383
2039	-1.388906	.6765493	-2.05	0.040	-2.714919	-.0628943
2040	-.6222491	.5510466	-1.13	0.259	-1.70228	.4577824

Note: ATET computed using covariates.

Note: Base time for pretreatment ATETs is the previous period.

Notice the note below the command. A variable with the name `_did_cohort` has been generated. Using the group variable and the observation-level treatment, `hdidregress` generated treatment-time cohorts. The new variable creates treatment groups based on the time when a group was first treated. For instance, if two schools adopt the Healthy Habits program in 2034, they are grouped in the 2034 cohort. The variable also contains a category for a control group. In this case, the control group is formed by the schools that never participate in the program. Cohorts are an important input for estimation and for postestimation commands. You do not need to adhere to the default name, `_did_cohort`, and may provide your own name using the `cohortvar()` option.

Next appears a table that gives you a sense of the treatment groups and time. You see the time variable, `year`, and its range, 2032 to 2040. Then we see what defines a treated or a control group. The table after provides group-level information about the cohort-time groups. The first row tells you the number

➤ Example 3: Less heterogeneity; aggregating and summarizing treatment effects

So far, we have allowed treatment effects to change over cohort and over time. But we might want to obtain only one treatment effect for each cohort, abstracting away from time variation within cohorts. You would get this using the postestimation command `estat aggregation`.

```
. estat aggregation, cohort
ATET over cohort                                     Number of obs = 16,725
                                     (Std. err. adjusted for 40 clusters in schools)
```

Cohort	ATET	Robust std. err.	z	P> z	[95% conf. interval]	
2034	-2.065755	.1999412	-10.33	0.000	-2.457633	-1.673877
2036	-1.7781	.4013978	-4.43	0.000	-2.564825	-.9913744
2038	-1.869405	.4650349	-4.02	0.000	-2.780857	-.9579538

Note: Aggregation weights vary across times and cohorts.

Note that aggregation occurs only for the posttreatment periods and not for the pretreatment periods. The 2034 estimate is a weighted average of all the treatment-effect estimates after 2034 for the 2034 cohort; see [\[CAUSAL\] hdidregress postestimation](#) for more details.

Aggregated estimates are easier to digest; now we have 3 treatment effects to analyze instead of 24. For the 2034 cohort, we have a treatment effect of -2.1 . For the 2036 cohort, the effect is -1.8 , and for the 2038 cohort, it is -1.9 . We cannot see how the treatment evolves over time for each cohort, but we have a sense of the average effect over time for each of them.

We could instead want to see the treatment effect at each point in time, abstracting from cohort-level variation.

```
. estat aggregation, time
ATET over time                                     Number of obs = 16,725
                                     (Std. err. adjusted for 40 clusters in schools)
```

Time	ATET	Robust std. err.	z	P> z	[95% conf. interval]	
2034	-1.226451	.379168	-3.23	0.001	-1.969607	-.4832957
2035	-2.491842	.4169657	-5.98	0.000	-3.30908	-1.674605
2036	-2.111619	.3654785	-5.78	0.000	-2.827943	-1.395294
2037	-3.028686	.4278557	-7.08	0.000	-3.867268	-2.190104
2038	-3.449829	.2670184	-12.92	0.000	-3.973176	-2.926483
2039	-.6624494	.44865	-1.48	0.140	-1.541787	.2168884
2040	-.7575068	.2816374	-2.69	0.007	-1.309506	-.2055078

Note: Aggregation weights vary across times and cohorts.

We see the treatment effects for each one of the posttreatment periods. As before, we have the option to look at the effects graphically. We just need to use the graph option.

```
. estat aggregation, time graph
```

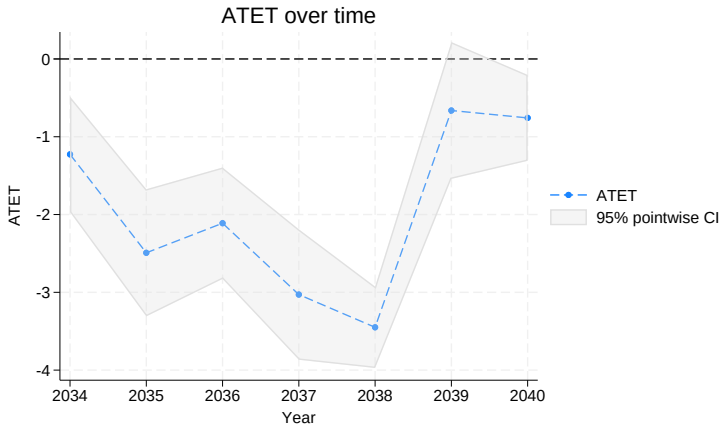


Figure 2. ATETs over time



▷ Example 4: Dynamic treatment effects

We could also ask what the evolution of the treatment effect is after treatment. For instance, we might want to know what happens one period after the onset of treatment, two periods after treatment, and so forth. It might be the case that treatment effects vanish over time or even change patterns. We might also want to see whether, before treatment, we observe a treatment effect or a pattern that might suggest that there is anticipation of treatment. `estat aggregation` allows us to answer these questions by using the `dynamic` option.

```
. estat aggregation, dynamic graph
```

Duration of exposure ATET

Number of obs = 16,725

(Std. err. adjusted for 40 clusters in schools)

Exposure	ATET	Robust std. err.	z	P> z	[95% conf. interval]	
-5	-1.434451	.5163232	-2.78	0.005	-2.446426	-.422476
-4	1.010288	.4808165	2.10	0.036	.067905	1.952671
-3	.1338267	.3091619	0.43	0.665	-.4721195	.739773
-2	-.4256324	.4292553	-0.99	0.321	-1.266957	.4156925
-1	.3727141	.3197563	1.17	0.244	-.2539967	.999425
0	-2.285098	.3827362	-5.97	0.000	-3.035248	-1.534949
1	-2.344265	.3829047	-6.12	0.000	-3.094744	-1.593785
2	-2.045521	.3911543	-5.23	0.000	-2.81217	-1.278873
3	-1.045601	.6840119	-1.53	0.126	-2.38624	.2950372
4	-2.145004	.5952525	-3.60	0.000	-3.311678	-.978331
5	-.604415	.5929199	-1.02	0.308	-1.766517	.5576866
6	-.6522272	.3640416	-1.79	0.073	-1.365736	.0612812

Note: Base time for pretreatment ATETs is the previous period.

Note: Exposure is the number of periods since the first treatment time.

Note: Aggregation weights vary across times and cohorts.

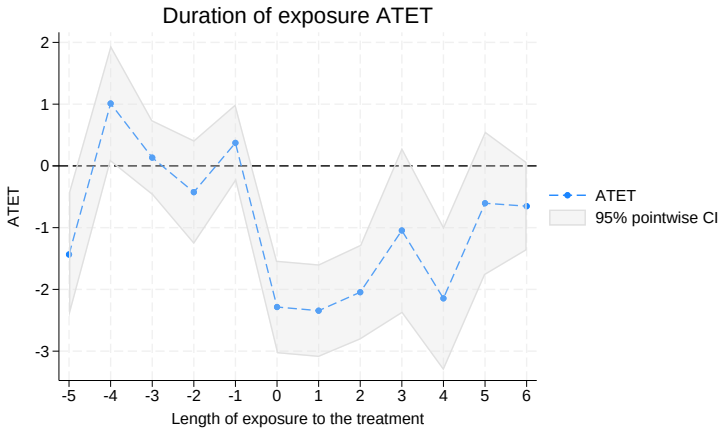


Figure 3. ATET dynamics

In the three periods prior to treatment, there is no effect. This suggests no anticipation to treatment. At the onset, the program reduces bmi, but the effect decreases for school districts that remain for more than four years in the program.



► Example 5: TWFE estimation

The literature on heterogeneous DID started by pointing out the problems that arise when one assumes erroneously that the treatment effects are homogeneous. It suggested that TWFE estimation was inadequate. [Wooldridge \(2021\)](#) suggests that fixed-effects estimation can be used if we extend it to include interactions between treatment-time cohorts and time.

Another important insight of [Wooldridge \(2021\)](#) is that you can use pooled ordinary least squares and add panel-level averages of covariates and obtain the same point estimates as one would get with fixed-effects estimation in the context of DID estimation. This is an extension of the intuition by [Mundlak \(1978\)](#). `xthdidregress` and `hdidregress` fit pooled ordinary least-squares models using these ideas. Below, we present the results we obtain using the `twfe` estimator.

```
. hddidregress twfe (bmi medu i.girl i.sports) (hhabit), group(schools) time(year)
note: variable _did_cohort, containing cohort indicators formed by treatment
      variable hhabit and group variable schools, was added to the dataset
      using the estimation sample.
```

Treatment and time information

Time variable: year

Time interval: 2032 to 2040

Control: **_did_cohort** = 0

Treatment: **_did_cohort** > 0

	_did_cohort
Number of cohorts	4
Number of obs	
Never treated	11355
2034	1231
2036	2097
2038	2042

Heterogeneous treatment-effects regression

Number of obs = 16,725

Data type: Repeated cross-sectional

Estimator: Two-way fixed effects

Treatment level: schools

Control group: Never treated

Heterogeneity: Cohort and time

Over: Cohort

(Std. err. adjusted for 40 clusters in schools)

Cohort	Robust					
	ATET	std. err.	t	P> t	[95% conf. interval]	
2034						
year						
2034	-.8057824	.2723491	-2.96	0.005	-1.35666	-.2549045
2035	-1.951481	.2098279	-9.30	0.000	-2.375898	-1.527064
2036	-2.091438	.2081903	-10.05	0.000	-2.512542	-1.670333
2037	-2.329408	.4674253	-4.98	0.000	-3.274865	-1.383952
2038	-3.623645	.4658056	-7.78	0.000	-4.565826	-2.681464
2039	-.1729334	.7543583	-0.23	0.820	-1.698767	1.3529
2040	-.2267266	.3344035	-0.68	0.502	-.9031216	.4496684
2036						
year						
2036	-1.671963	.3424563	-4.88	0.000	-2.364646	-.9792798
2037	-3.27542	.3496365	-9.37	0.000	-3.982627	-2.568213
2038	-2.995124	.2853544	-10.50	0.000	-3.572308	-2.41794
2039	-.0792949	.5152787	-0.15	0.878	-1.121544	.9629547
2040	-.9852905	.1856743	-5.31	0.000	-1.360852	-.6097289
2038						
year						
2038	-3.389082	.154181	-21.98	0.000	-3.700942	-3.077221
2039	-.7309226	.5173441	-1.41	0.166	-1.77735	.3155046
2040	-.6942153	.3558485	-1.95	0.058	-1.413987	.0255563

Note: ATET computed using covariates.

The output is almost the same as the one for the ra estimator in [example 1](#). There are a couple of noteworthy differences. First, the estimator fits an extended TWFE regression. Second, the ATET parameters are shown for each cohort only at the time of treatment exposure and for the periods thereafter but not for the pretreatment periods. As discussed in [Wooldridge \(2021\)](#), these are the parameters identified using the parallel-trends assumption he derives.

As we did before, we could use `estat` aggregation to explore different ways of looking at our treatment effects and `estat atetplot` to visualize the estimated ATETs.



► Example 6: Reducing model complexity

When we fit the `aipw` model, we had to estimate ATET parameters for each cohort over time. The complexity of the model grows with the number of cohorts and the number of time periods. As is described in [Methods and formulas](#), the `aipw` estimator uses a different subset of the data to obtain each parameter. To get a reliable estimator of each parameter, you need sufficient data for each subsample. Sometimes, there are few observations for a given cohort in a given set of time periods.

We can ameliorate this problem by reducing the amount of heterogeneity we assume. For the `twfe` estimator, the complexity of the model comes from the interactions between the observation-level treatment with cohort and time and the interactions between the observation-level treatment, cohort, time, and covariates. This allows us to decide which interactions to include in our model. We could, for instance, allow for heterogeneity at the cohort level instead of at the cohort and time level. We use the `hettype()` option with the argument `cohort` to do this:

```
. hdidregress twfe (bmi medu i.girl i.sports) (hhabit), group(schools)
> time(year) hettype(cohort)
note: variable _did_cohort, containing cohort indicators formed by treatment
      variable hhabit and group variable schools, was added to the dataset
      using the estimation sample.
```

Treatment and time information

Time variable: year

Time interval: 2032 to 2040

Control: `_did_cohort = 0`

Treatment: `_did_cohort > 0`

	<code>_did_cohort</code>
Number of cohorts	4
Number of obs	
Never treated	11355
2034	1231
2036	2097
2038	2042

Heterogeneous treatment-effects regression Number of obs = 16,725
 Data type: Repeated cross-sectional
 Estimator: Two-way fixed effects
 Treatment level: schools
 Control group: Never treated
 Heterogeneity: Cohort

(Std. err. adjusted for 40 clusters in schools)

Cohort	ATET	Robust std. err.	t	P> t	[95% conf. interval]	
2034	-1.619553	.2223114	-7.29	0.000	-2.069221	-1.169886
2036	-1.832602	.1954433	-9.38	0.000	-2.227924	-1.437281
2038	-1.739144	.2152765	-8.08	0.000	-2.174582	-1.303706

Note: ATET computed using covariates.

You fit a regression model with fewer terms and obtain treatment effects only at the cohort level. You could also have the treatment effect change over time but not over cohort by typing `hettype(time)`.

For the estimators proposed by [Callaway and Sant’Anna \(2021\)](#), heterogeneity is built in, so we need to estimate all the ATET parameters.



► Example 7: Defining your own cohort

By default, `hdidregress` creates a cohort variable based on the estimation sample. Yet this might be inadequate if a researcher has more information than is provided in the dataset. Suppose that our dataset looked something like this for school district 1:

```
. list schools year hhabit in 100/105, noobs sepby(schools)
```

schools	year	hhabit
1	2033	No
1	2033	No
1	2035	Yes
1	2035	Yes
1	2035	Yes
1	2035	Yes

There is no information for the year 2034. If the school district participated in the healthy habits program in 2034, it should belong to the 2034 cohort. However, `hdidregress` has no information about the year 2034 in the estimation sample and will classify school district 1 as belonging to the 2035 cohort. `hdidregress`’s inability to determine the proper cohort is not exclusive to situations with gaps in your repeated cross-section. In fact, Stata excludes observations in your sample if any of the variables used during estimation are missing. If all observations for the time period in which a group is first treated are omitted because of missing values, `hdidregress` cannot assign the group to the appropriate cohort.

If you have information about the cohort values, instead of letting the command create a cohort variable, you can provide the cohort variable with the `usercohort()` option. Suppose you had a cohort variable, `mycohort`; then you could type

```
. hdidregress twfe (bmi) (hhabit), group(schools) time(year) usercohort(mycohort)
```


Another possibility is to generate the cohort variable `mycohort` yourself using the `gencohort` command; this is helpful when you have missing information on covariates or the outcome but have enough information about the treatment. Suppose you had missing information about the outcome variable `bmi` but had information about the treatment variable. Below, we drop information about the 2034 cohort to illustrate the point.

```
. replace bmi = . if year==2034 & schools==1
(44 real changes made, 44 to missing)
```

These observations for year 2034 would not be used during estimation, but we have enough information in them to create our own cohort variable.

```
. gencohort mycohort, treat(hhabit) time(year) group(schools)
. sort schools year bmi
. list schools year hhabit bmi mycohort in 100/105, noobs sepby(schools)
```

schools	year	hhabit	bmi	mycohort
1	2033	No	34.34539	2034
1	2033	No	36.46486	2034
1	2034	Yes	.	2034
1	2034	Yes	.	2034
1	2034	Yes	.	2034
1	2034	Yes	.	2034

The `mycohort` variable can now be specified in the `usercohort()` option of `hdidregress` to properly treat school district 1 as belonging to cohort 2034.



Stored results

`hdidregress` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_clust)</code>	number of clusters
<code>e(tmin)</code>	first time period
<code>e(tmax)</code>	last time period
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(xinteract)</code>	1 if covariate interactions are included, 0 otherwise

Macros

<code>e(cmd)</code>	<code>hdidregress</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(clustvar)</code>	name of cluster variable
<code>e(control_group)</code>	control group
<code>e(het_type)</code>	heterogeneity type for <code>twfe</code> estimator
<code>e(over_type)</code>	subpopulation over which each ATET is computed for <code>twfe</code> estimator
<code>e(cohortvar)</code>	name of cohort variable
<code>e(usercohort)</code>	name of user-specified cohort variable
<code>e(ovar)</code>	name of outcome variable
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(marginsnotok)</code>	predictions disallowed by margins
<code>e(timevar)</code>	time variable
<code>e(treatname)</code>	name of treatment variable

e(basetime)	type of pretreatment base time
e(estat_cmd)	program used to implement estat
e(vce)	vcetype specified in vce()
e(vcetype)	title used to label Std. err.
e(method)	estimator method
e(properties)	b V
Matrices	
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
e(cohort_count)	matrix with cohort count information
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

Methods and formulas are presented under the following headings:

[Introduction](#)
[The RA, IPW, and AIPW estimators](#)
[The TWFE estimator](#)

Introduction

`hdidregress` for repeated cross-sectional data implements the RA, IPW, and AIPW estimators, outlined in [Callaway and Sant’Anna \(2021\)](#), and the TWFE estimator, outlined in [Wooldridge \(2021\)](#).

To reveal how the heterogeneous treatment effects evolve across cohorts and time, we are interested in estimating the ATET for each combination of cohort and time. Cohorts are defined by the time a group is treated, where time is denoted by t , where $t = 1, \dots, T$. We denote a cohort by g and the individuals in our sample by i , where $i = 1, \dots, N$. Let G_{ig} be an indicator that equals one if unit i is first treated at time g . Then the units in cohort g can be denoted by $G_{ig} = 1$. When a unit i is never treated, we denote $G_{i0} = 1$. Thus, cohort 0 indicates all the units that are never treated. We assume that once a unit is treated, it will remain treated. We also define d_{it} as an indicator for treatment of unit i at time t .

Let $\theta(g, t)$ be the ATET for cohort g at time t , which is defined as

$$\theta(g, t) = \mathbf{E} \{ y_t(g) - y_t(0) | G_g = 1 \} \quad (\text{ATET})$$

where $y_t(g)$ is the potential outcome at time t for those first treated at time g , $y_t(0)$ is the potential outcome for those that are never treated, and G_g equals 1 if a unit belongs to cohort g . All the four estimators provided in `hdidregress` estimate $\theta(g, t)$ in equation (ATET). We cannot directly estimate $\theta(g, t)$ using equation (ATET) because the potential outcomes $y_t(g)$ and $y_t(0)$ are not observable.

Next, we will describe the RA, IPW, and AIPW estimators.

The RA, IPW, and AIPW estimators

To estimate the ATET for cohort g at time t , the RA, IPW, and AIPW estimators transform the estimation into a classical two groups and two periods difference-in-differences setup. Thus, we need to restrict the data to an estimation sample with only two groups and only two periods based on the values of g and t . For the two groups, one group comprises all observations in cohort g ; the other group comprises untreated observations not in cohort g , also known as a control group. For the two periods, one period is the data in time t ; the other period is a period when cohort g is not treated, also known as base time.

There are two ways to define the control group. One way is to use the units that are never treated as the control group. Let C^{NEV} be an indicator that equals one if a unit belongs to the never-treated group. In particular, $C^{\text{NEV}} = G_0$. Another way is to use the units not in cohort g and not yet treated at time t as the control group. Let $C^{\text{NY}}_{g,t}$ be an indicator that equals one if a unit belongs to the not-yet-treated group by time t . In particular, $C^{\text{NY}}_{g,t} = (1 - G_g)(1 - d_t)$. To simplify, we indicate control, in both cases, as $C^*_{g,t}$.

The definitions of the RA, IPW, and AIPW estimators depend on the definition of $C^*_{g,t}$, which can either be C^{NEV} or $C^{\text{NY}}_{g,t}$. However, regardless of the control group's choice, the estimators' definitions can always be written using the general notation $C^*_{g,t}$.

There are also two ways to define the base time. One way is to adaptively choose the base time for the pretreatment periods. When the adaptive method is used to compute the ATET for cohort g at time t , for the pretreatment periods, the base time is $t - 1$; for the posttreatment periods, the base time is $g - 1$. Another way is to use a common base time $g - 1$ for both pretreatment and posttreatment periods. The common base time is useful for identifying a violation of the parallel trends assumption in event studies as discussed in Roth (2024). To simplify the notation, we indicate the base time in both cases as t_0 .

For each unit i in the pooled sample, we observe $\{\tau_i, y_{i,\tau_i}, \mathbf{x}_{i,\tau_i}, d_{i,\tau_i}, \mathbf{z}_{i,\tau_i}\}$, where y_i is the outcome, $\mathbf{x}_i$ are pretreatment covariates for the outcome model, d_i is a treatment indicator, $\mathbf{z}_i$ are covariates for the treatment assignment model, and $\tau_i \in \{1, \dots, T\}$ is a categorical variable indicating the time when unit i is observed. Let T_t equal one if the unit is observed at time t and zero otherwise.

The estimands also require the following notation,

$$\begin{aligned} m_{g,s}^{\text{treat}}(\mathbf{x}) &= \mathbf{E}(y|\mathbf{x}, G_g = 1, \tau = s) \\ m_{g,s,t}^{\text{comp}}(\mathbf{x}) &= \mathbf{E}(y|\mathbf{x}, C^*_{g,t} = 1, \tau = s) \\ w_{g,s}^{\text{treat}} &= \frac{T_s G_g}{\mathbf{E}(T_s G_g)} \\ w_{g,s,t}^{\text{comp}}(\mathbf{z}) &= \frac{\frac{T_s p_{g,t}(\mathbf{z}) C^*_{g,t}}{1 - p_{g,t}(\mathbf{z})}}{\mathbf{E}\left\{\frac{T_s p_{g,t}(\mathbf{z}) C^*_{g,t}}{1 - p_{g,t}(\mathbf{z})}\right\}} \end{aligned}$$

where $p_{g,t}(\mathbf{z})$ is defined by

$$p_{g,t}(\mathbf{z}) = \Pr(G_g = 1 | \mathbf{z}, G_g + C^*_{g,t} = 1) \quad (\text{Pz})$$

and the superscript refers to the group we are conditioning on, either the treated group (treat) or the control or comparison group (comp).

The RA estimand is

$$\theta_{\text{RA}}(g, t) = \mathbf{E} \left(\frac{G_g}{\mathbf{E}(G_g)} \left[\{m_{g,t}^{\text{treat}}(\mathbf{x}) - m_{g,g-1}^{\text{treat}}(\mathbf{x})\} - \{m_{g,t,t}^{\text{comp}}(\mathbf{x}) - m_{g,g-1,t}^{\text{comp}}(\mathbf{x})\} \right] \right) \quad (\text{RA})$$

The IPW estimand is

$$\theta_{\text{IPW}}(g, t) = \mathbf{E} \left\{ \left(w_{g,t}^{\text{treat}} - w_{g,g-1}^{\text{treat}} \right) y \right\} - \mathbf{E} \left[\{w_{g,t,t}^{\text{comp}}(\mathbf{z}) - w_{g,g-1,t}^{\text{comp}}(\mathbf{z})\} y \right] \quad (\text{IPW})$$

The AIPW estimand is

$$\begin{aligned} \theta_{\text{AIPW}}(g, t) = \mathbf{E} \left(\frac{G_g}{\mathbf{E}(G_g)} \left[\{m_{g,t}^{\text{treat}}(\mathbf{x}) - m_{g,g-1}^{\text{treat}}(\mathbf{x})\} - \{m_{g,t,t}^{\text{comp}}(\mathbf{x}) - m_{g,g-1,t}^{\text{comp}}(\mathbf{x})\} \right] \right) \\ + \mathbf{E} \left[w_{g,t}^{\text{treat}} \{y - m_{g,t}^{\text{treat}}(\mathbf{x})\} - w_{g,g-1}^{\text{treat}} \{y - m_{g,g-1}^{\text{treat}}(\mathbf{x})\} \right] \\ - \mathbf{E} \left[w_{g,t,t}^{\text{comp}}(\mathbf{z}) \{y - m_{g,t,t}^{\text{comp}}(\mathbf{x})\} - w_{g,g-1,t}^{\text{comp}}(\mathbf{z}) \{y - m_{g,g-1,t}^{\text{comp}}(\mathbf{x})\} \right] \end{aligned} \quad (\text{AIPW})$$

Under some regularity conditions, [Callaway and Sant’Anna \(2021\)](#) showed that the estimand for RA, IPW, and AIPW is the same as $\theta(g, t)$ in equation (ATET). In other words,

$$\theta(g, t) = \theta_{\text{RA}}(g, t) = \theta_{\text{IPW}}(g, t) = \theta_{\text{AIPW}}(g, t)$$

Furthermore, the estimands in equations (AIPW) are estimable because they are all based on observed variables. The identification of the estimators sheds light on how to estimate $\theta(g, t)$. The estimator can be generally divided into three steps:

1. Restrict the sample to time t and t_0 , and keep only the units in cohort g or in control group $C_{g,t}^*$. When option `basetime(adaptive)` is specified, $t_0 = g - 1$ if $t \geq g$ or $t_0 = t - 1$ if $t < g$. When option `basetime(common)` is specified, $t_0 = g - 1$.
2. Use a parametric model to estimate the nuisance functions.
 - a. For outcomes: linear regression to estimate $m_{g,t}^{\text{treat}}(\mathbf{x})$, $m_{g,t_0}^{\text{treat}}(\mathbf{x})$, $m_{g,s,t}^{\text{comp}}(\mathbf{x})$, and $m_{g,s,t_0}^{\text{comp}}(\mathbf{x})$.
 - b. For propensity: logit regression to estimate $p_{g,t}(\mathbf{z})$.
 - c. For probability weights: $w_{g,t}^{\text{treat}}$, w_{g,t_0}^{treat} , $w_{g,s,t}^{\text{comp}}(\mathbf{z})$, and $w_{g,s,t_0}^{\text{comp}}(\mathbf{z})$ to estimate using propensity scores T_t and G_g .
3. Plug in the nuisance function estimates into the estimating equation in equations (IPW), (AIPW), or (RA). Notice that the expectation operator $\mathbf{E}(\cdot)$ is replaced by the sample average.

The variance–covariance matrix for the estimates is computed using the influence-function approach proposed in [Callaway and Sant’Anna \(2021\)](#). The influence function approach is numerically equivalent to the generalized method of moments approach. However, it is much faster because it avoids computing the covariance matrix for the parameters in the nuisance functions. For more discussions on influence functions, see [Hampel et al. \(1986\)](#), [Newey and McFadden \(1994\)](#), and [Jann \(2020\)](#).

The TWFE estimator

A TWFE estimator for repeated cross-sections fits

$$y_i = \alpha_h + \gamma_t + \mathbf{x}_i\beta + d_i\tau + \epsilon_i$$

Above, h denotes the group level at which treatment occurs. [Wooldridge \(2021\)](#) extends this model to incorporate interactions between the observation-level treatment, d_i , cohort, G_{ig} , posttreatment periods, and covariates. We define indicators for posttreatment periods as f_s with s going from q to T , where q is the first time period we observe treatment. For instance, f_q equals 1 if we are in time-period q and 0 otherwise. If `noxinteract` has not been specified, the extended fixed-effects model is given by

$$y_i = \eta + \sum_{g=q}^T G_{ig}\theta_g + \sum_{s=2}^T f_s\gamma_s + \sum_{g=q}^T \sum_{s=g}^T d_i G_{ig} f_s \tau_{gs} + \sum_{g=q}^T G_{ig} \mathbf{x}_i \boldsymbol{\theta}_g + \sum_{s=2}^T f_s \mathbf{x}_i \boldsymbol{\gamma}_s + \sum_{g=q}^T \sum_{s=g}^T d_i G_{ig} f_s \mathbf{x}_i \boldsymbol{\tau}_{gs} + \mathbf{x}_i \boldsymbol{\beta} + \epsilon_i \quad (\text{TWFE})$$

We can fit equation (TWFE) using pooled ordinary least squares or a within estimator. We are going to use the estimator proposed by [Mundlak \(1978\)](#). This gives the same point estimates as using the within estimator with h as the panel level for the parameters in equation (TWFE) but has different degrees of freedom because of the additional terms added by the Mundlak approach. Unlike within estimation, the Mundlak approach works for both repeated cross-sectional data and panel data. Also, it has good properties to obtain partial effects under various data-generating processes, as pointed out in [Wooldridge \(2019\)](#).

Given this model, the cohort-time treatment effects are a function of τ_{gs} , $\boldsymbol{\tau}_{gs}^x$, and the covariates. We can compute these effects using `margins` by typing

```
. margins, dydx(d) at(year=q ... year=T) over(cohort) vce(unconditional)
```

where `d` is the treatment indicator, `year` indicates treatment times at which treatment will be evaluated using `at()`, and `cohort` is the treatment-time cohorts.

If `over(timecohort)` was specified with `hdidregress`, we can obtain effects by typing

```
. margins, dydx(d) over(year cohort) vce(unconditional)
```

We use `vce(unconditional)` to account for the variation in the covariates. When there are no covariates specified or when `noxinteract` is specified, `vce(unconditional)` has no effect. Furthermore, if we include covariates but specify `noxinteract`, our equation reduces to

$$y_i = \eta + \sum_{g=q}^T G_{ig}\theta_g + \sum_{s=2}^T f_s\gamma_s + \sum_{g=q}^T \sum_{s=g}^T d_i G_{ig} f_s \tau_{gs} + \mathbf{x}_i\beta + \epsilon_i$$

In this case, τ_{gs} are the treatment effects.

In practice, `hdidregress` computes the treatment effects analytically rather than by use of `margins`. Specifically, a modified Mundlak regression model is fit. The modified regression interacts treatment indicators with demeaned covariates instead of the covariates themselves. By default, the covariates are demeaned by cohort-specific covariate means. If `over(timecohort)` is specified, the covariates are demeaned by cohort-time-specific means. Treatment-effect parameters can be estimated as coefficients of this regression rather than as linear combinations of regression coefficients.

The modified Mundlak regression is treated as being fit following a set of first-stage regressions of each covariate on cohort indicators. GMM-style standard errors account for variation in these first-stage regressions and are equivalent to the standard errors produced by `margins` with the `vce(unconditional)` option.

With the `hettype()` option, we reduce the complexity of equation (TWFE). In particular, if we ask for `hettype(time)`, we have

$$y_i = \eta + \sum_{g=q}^T G_{ig} \theta_g + \sum_{s=2}^T f_s \gamma_s + \sum_{s=q}^T d_i f_s \tau_s + \sum_{g=q}^T G_{ig} \mathbf{x}_i \theta_g^{\mathbf{x}} + \sum_{s=2}^T f_s \mathbf{x}_i \gamma_s^{\mathbf{x}} + \sum_{s=q}^T d_i f_s \mathbf{x}_i \tau_s^{\mathbf{x}} + \mathbf{x}_i \beta + \epsilon_i$$

Now treatment varies over time but not over cohort. If we use the `hettype(cohort)` option, we have

$$y_i = \eta + \sum_{g=q}^T G_{ig} \theta_g + \sum_{s=2}^T f_s \gamma_s + \sum_{g=q}^T d_i G_{ig} \tau_g + \sum_{g=q}^T G_{ig} \mathbf{x}_i \theta_g^{\mathbf{x}} + \sum_{s=2}^T f_s \mathbf{x}_i \gamma_s^{\mathbf{x}} + \sum_{g=q}^T d_i G_{ig} \mathbf{x}_i \tau_g^{\mathbf{x}} + \mathbf{x}_i \beta + \epsilon_i$$

Now treatment varies over cohort but not over time.

When the `controlgroup(notyet)` option is specified, the G_{ig} indicator excludes the last treated cohort. As discussed in Wooldridge (2021), when every group is eventually treated, we cannot identify the treatment effect for this cohort. It is therefore sensible to use the last treated cohort as a control group. When some of the units in our sample are never treated, we can always identify all cohorts, and the `twfe` estimator will always revert to using `controlgroup(never)`.

Acknowledgments

We thank Fernando Rios-Avila of the Levy Economics Institute of Bard College for helpful conversations and advice and for his contributions in developing heterogeneous treatment-effects software and accompanying material. His community-contributed commands `csdid` and `drdid` implement some of the methods in this entry and a few others we have yet to include.

We thank Pedro H. C. Sant’Anna of Emory University for helpful conversations and advice and for his continuing contributions to the heterogeneous treatment-effects literature.

We thank Jeff Wooldridge of Michigan State University for helpful conversations about extended two-way fixed-effects estimation.

We also thank Austin Nichols of Amazon for his advice, for his insights, and for organizing the [WHYDC](#) causal inference conference.

Stata has an active research community adding features to the area of heterogeneous DID. We would like to acknowledge their previous and ongoing contributions to this area: `csdid` by Fernando Rios-Avila; `did2s` by Kyle Butts; `did_multiplegt` by Clement de Chaisemartin and Xavier D’Haultfœuille; `did_imputation` by Kirill Borusyak, Xavier Jaravel, and Jann Spiess; `eventstudyinteract` by Liyang Sun and Sarah Abraham; `jwdid` by Fernando Rios-Avila; and many more. Type `search heterogeneous difference in differences` to see Stata’s official and community-contributed features for heterogeneous DID.

References

- Callaway, B., and P. H. C. Sant’Anna. 2021. Difference-in-differences with multiple time periods. *Journal of Econometrics* 225: 200–230. <https://doi.org/10.1016/j.jeconom.2020.12.001>.
- de Chaisemartin, C., and X. D’Haultfœuille. 2023. Two-way fixed effects and differences-in-differences with heterogeneous treatment effects: A survey. *Econometrics Journal* 26: C1–C30. <https://doi.org/10.1093/ectj/utac017>.
- Hampel, F. R., E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. 1986. *Robust Statistics: The Approach Based on Influence Functions*. New York: Wiley. <https://doi.org/10.1002/9781118186435>.
- Jann, B. 2020. Influence functions continued. A framework for estimating standard errors in reweighting, matching, and regression adjustment. Working Papers 35, University of Bern Social Sciences. <https://ideas.repec.org/p/bss/wpaper/35.html>.
- Mundlak, Y. 1978. On the pooling of time series and cross section data. *Econometrica* 46: 69–85. <https://doi.org/10.2307/1913646>.
- Newey, W. K., and D. L. McFadden. 1994. “Large sample estimation and hypothesis testing”. In *Handbook of Econometrics*, edited by R. F. Engle and D. L. McFadden, vol. 4: 2111–2245. Amsterdam: Elsevier. [https://doi.org/10.1016/S1573-4412\(05\)80005-4](https://doi.org/10.1016/S1573-4412(05)80005-4).
- Roth, J. 2024. Interpreting event-studies from recent difference-in-differences methods. ArXiv Working Paper No. arXiv:2401.12309, <https://doi.org/10.48550/arXiv.2401.12309>.
- Roth, J., P. H. C. Sant’Anna, A. Bilinski, and J. Poe. 2022. What’s trending in difference-in-differences? A synthesis of the recent econometrics literature. arXiv:2201.01194 [econ.EM], <https://doi.org/10.48550/arXiv.2201.01194>.
- Wooldridge, J. M. 2019. Correlated random effects models with unbalanced panels. *Journal of Econometrics* 211: 137–150. <https://doi.org/10.1016/j.jeconom.2018.12.010>.
- . 2021. Two-way fixed effects, the two-way Mundlak regression, and difference-in-differences estimators. Working paper, Department of Economics, Michigan State University, East Lansing, MI. <https://doi.org/10.2139/ssrn.3906345>.

Also see

- [CAUSAL] **hdidregress postestimation** — Postestimation tools for hdidregress and xthdidregress⁺
- [CAUSAL] **DID intro** — Introduction to difference-in-differences estimation
- [CAUSAL] **didregress** — Difference-in-differences estimation⁺
- [CAUSAL] **gencohort** — Create a cohort variable
- [CAUSAL] **teffects intro** — Introduction to treatment effects for observational data
- [CAUSAL] **teffects intro advanced** — Advanced introduction to treatment effects for observational data
- [CAUSAL] **xthdidregress** — Heterogeneous difference in differences for panel data⁺
- [CAUSAL] **xtswitchdid** — Difference in differences with switching treatments for panel data⁺
- [U] **20 Estimation and postestimation commands**

estat

Description for estat

`estat ptrends` tests that all pretreatment periods are equal to zero.

`estat atetplot` plots the coefficients of ATET for each cohort across different periods.

`estat aggregation` aggregates the cohort-period ATETs to characterize the heterogeneity of treatment effects. Aggregation may be within cohorts, time periods, time exposed to treatment, or within cohort and time periods. You may display the output of `estat aggregation` simultaneously as a table and a graph. The default is the tabular output.

`estat moderation` is part of StataNow. It reports moderating effects of covariates in two-way fixed-effects models. These effects can be interpreted as the marginal effects of each covariate on the ATETs reported by `hdidregress` and `xthdidregress`. Factor variables may be specified but interaction terms are not allowed.

`estat sci` provides the simultaneous confidence intervals for ATETs using the multiplier bootstrap method proposed in [Callaway and Sant'Anna \(2021\)](#). It may not be used after estimation using the TWFE estimator.

Menu for estat

Statistics > Postestimation

Syntax for estat

Tests that all pretreatment periods are zero

```
estat ptrends
```

Plot coefficients for ATETs

```
estat atetplot [cohort_list] [, atetplot_options]
```

Aggregate ATETs

```
estat aggregation [, aggregation_options]
```

Moderating effects

```
estat moderation [varlist] [, level(#)]
```

Simultaneous confidence intervals

```
estat sci [, level(#) sci_options]
```

cohort_list is a subset of all the cohorts when estimating the ATETs. By default, the *cohort_list* contains all the cohorts. *cohort_list* is not allowed when the TWFE estimator is combined with the `hettype(time)` or `hettype(cohort)` option.

varlist may contain factor variables; see [\[U\] 11.4.3 Factor variables](#).

<i>atetplot_options</i>	Description
<code>level(#)</code>	set confidence level
* <code>sci[(<i>sci_options</i>)]</code>	use multiplier bootstrap to compute the simultaneous confidence intervals
Graph options	
Main	
<code>noci</code>	do not plot the confidence intervals
* <code>preteopts(<i>scatter_opts</i>)</code>	affect rendition of the pretreatment scatterplot
<code>postteopts(<i>scatter_opts</i>)</code>	affect rendition of the posttreatment scatterplot
<code>[no]zeroline</code>	suppress the y -axis reference line passing through zero
<code>zerolineopts(<i>refline_opts</i>)</code>	affect rendition of the y -axis reference line passing through zero
† <code>[no]cohortline</code>	suppress the x -axis reference line passing through the time when the treatment began for each cohort
† <code>cohortlineopts(<i>refline_opts</i>)</code>	affect rendition of the x -axis reference line passing through the time when the treatment began for each cohort
CI plot	
<code>ciopts(<i>area_opts</i>)</code>	affect rendition of the confidence interval
Y axis, X axis, Titles, Legend, Overall	
† <code>byopts(<i>byopts</i>)</code>	affect rendition of the graph by cohorts
<code>twoway_options</code>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>

*These options are not allowed for the TWFE estimators.

†These options are not allowed when the TWFE estimator is combined with `hettype(time)` or `hettype(cohort)`.

<i>aggregation_options</i>	Description
<code>overall</code>	aggregate ATETs within cohorts and time periods; the default
<code>dynamic[(<i>event_list</i>)]</code>	aggregate ATETs within exposures to the treatment
<code>time[(<i>time_list</i>)]</code>	aggregate ATETs within time periods
<code>cohort[(<i>cohort_list</i>)]</code>	aggregate ATETs within cohorts
* <code>weights(<i>weight_type</i>)</code>	use <i>weight_type</i> to aggregate ATETs; default is <code>weights(timecohort)</code>
<code>[no]graph</code>	whether to suppress or display the aggregation plot; <code>nograph</code> is the default
<code>graph[(<i>graph_opts</i>)]</code>	affect rendition of the aggregation plot
<code>level(#)</code>	set confidence level
* <code>sci[(<i>sci_options</i>)]</code>	use multiplier bootstrap to compute the simultaneous confidence intervals

Only one of `overall`, `dynamic()`, `cohort()`, or `time()` is allowed.

*These options are not allowed after the TWFE estimator.

<i>sci_options</i>	Description
<code>rseed(#)</code>	set random-number seed to #
<code>reps(#)</code>	perform # multiplier bootstrap replications; default is <code>reps(999)</code>

`sci` or `sci()` is not allowed after the TWFE estimator in `hdidregress` and `xthdidregress`. In addition, it is not allowed after estimation with bootstrap or jackknife standard errors for RA, IPW, and AIPW estimators.

By default, `estat atetplot` plots the pointwise CIs.

Main

`noci` removes plots of the CIs. The default is to plot the CIs.

`preteopts(scatter_opts)` affects the rendition of the scatterplot for pretreatment periods. This option is not allowed after the TWFE estimator in `hdidregress` and `xthdidregress`. `scatter_opts` may be the following:

`connect_options` specify how points on a graph are to be connected; [G-3] [connect_options](#).

`marker_options` affect the rendition of markers drawn at the plotted points, including their shape, size, color, and outline; see [G-3] [marker_options](#).

`postteopts(scatter_opts)` affects the rendition of the scatterplot for posttreatment periods. `scatter_opts` may be the following:

`connect_options` specify how points on a graph are to be connected; [G-3] [connect_options](#).

`marker_options` affect the rendition of markers drawn at the plotted points, including their shape, size, color, and outline; see [G-3] [marker_options](#).

`nozeroline` suppresses the *y*-axis reference line passing through zero. After estimation with `hdidregress` and the RA, IPW, or AIPW estimator, the default is to plot this reference line. After estimation with the TWFE estimator, the default is not to plot this reference line.

`zerolineopts(refline_opts)` affects the rendition of the reference line passing through zero. `refline_opts` may be the following:

`style(addedlinestyle)` specifies the overall style of the added line, which includes `[no]extend` and `lstyle(linestyle)` documented below. See [G-4] [addedlinestyle](#). The `[no]extend` and `lstyle()` options allow you to change the added line's attributes individually, but `style()` is the starting point.

You need not specify `style()` just because there is something that you want to change, and in fact, most people seldom specify the `style()` option. You specify `style()` when another style exists that is exactly what you desire or when another style would allow you to specify fewer changes to obtain what you want.

`extend` and `noextend` specify whether the line should extend through the plot region's margin and touch the axis; see [G-3] [region_options](#). Usually, `noextend` is the default, and `extend` is the option, but that is determined by the overall `style()` and, of course, the scheme; see [G-4] [Schemes intro](#).

`lstyle(linestyle)`, `lpattern(linepatternstyle)`, `lwidth(linewidthstyle)`, `lalign(linealignmentstyle)`, and `lcolor(colorstyle)` specify the look of the line; see [G-2] [graph twoway line](#).

`nocohortline` suppresses the *x*-axis reference line passing through the time when the treatment began for each cohort. The default is to plot this reference line. This option is not allowed after the TWFE estimator.

`cohortlineopts(refline_opts)` affects the rendition of the reference line passing through the time when the treatment began for each cohort. This option is not allowed after the TWFE estimator. *refline_opts* may be the following:

`style(addedlinestyle)` specifies the overall style of the added line, which includes `[no]extend` and `lstyle(linestyle)` documented below. See [G-4] [addedlinestyle](#). The `[no]extend` and `lstyle()` options allow you to change the added line's attributes individually, but `style()` is the starting point.

You need not specify `style()` just because there is something that you want to change, and in fact, most people seldom specify the `style()` option. You specify `style()` when another style exists that is exactly what you desire or when another style would allow you to specify fewer changes to obtain what you want.

`extend` and `noextend` specify whether the line should extend through the plot region's margin and touch the axis; see [G-3] [region_options](#). Usually, `noextend` is the default, and `extend` is the option, but that is determined by the overall `style()` and, of course, the scheme; see [G-4] [Schemes intro](#).

`lstyle(linestyle)`, `lpattern(linepatternstyle)`, `lwidth(linewidthstyle)`, `lalign(linealignmentstyle)`, and `lcolor(colorstyle)` specify the look of the line; see [G-2] [graph twoway line](#).

CI plot

`ciopts(area_options)` affects the rendition of the CIs; see [G-3] [area_options](#).

Y axis, X axis, Titles, Legend, Overall

`byopts(byopts)` affects the rendition of the graph combined by cohorts. For *byopts*, see [G-3] [by_option](#). This option is not allowed after the TWFE estimator.

twoway_options are any of the options documented in [G-3] [twoway_options](#), excluding `by()`. These include options for titling the graph (see [G-3] [title_options](#)) and for saving the graph to disk (see [G-3] [saving_option](#)).

Options for estat aggregation

`overall` aggregates ATETs within all the cohorts and time periods; it is the default.

`dynamic` or `dynamic(event_list)` aggregates ATETs within exposure to the treatment. For example, two periods of exposure to the treatment means two periods after the treatment started. Specifying `dynamic` implies aggregating ATETs within all the estimable exposures to the treatment.

`dynamic(event_list)` aggregates ATETs within the exposure to the treatment specified by *event_list*. *event_list* is a [numlist](#) specifying length of exposures to the treatment.

`time` or `time(time_list)` aggregates ATETs within time periods. Specifying `time` implies aggregating ATETs within all the estimable time periods.

`time(time_list)` aggregates ATETs within the time specified by *time_list*. *time_list* is a set of time periods used during estimation.

`cohort` or `cohort(cohort_list)` aggregates ATETs within cohort. Specifying `cohort` implies aggregating ATETs within all the estimable cohorts.

`cohort(cohort_list)` aggregates ATETs within the cohorts specified by *cohort_list*. *cohort_list* is a set of cohorts used during estimation.

`weights(weight_type)` defines the weight type used to aggregate the different cohort-time treatment effects. *weight_type* may be one of the following:

`timecohort`, the default, constructs individual weights using the subset of the data used to compute each ATET(g, t). These subsets vary by time and cohort and so do the weights.

`cohort` constructs individual weights using all subsets of the data to compute the proportion of observations in each cohort. Because the subsets are pooled over all time periods, `cohort` weights vary across cohorts but not over time.

`nograph` and `graph` specifies whether to suppress or display the plot of aggregation of ATETs. `nograph` is the default.

`graph(graph_opts)` affects the rendition of the aggregation plot. *graph_opts* may be the following:

`noci` removes plots of the CIs. The default is to plot the CIs.

connect_options specify how points on a graph are to be connected; [G-3] [connect_options](#).

marker_options affect the rendition of markers drawn at the plotted points, including their shape, size, color, and outline; see [G-3] [marker_options](#).

`ciopts(area_options)` affects the rendition of the CIs; see [G-3] [area_options](#).

twoway_options are any of the options documented in [G-3] [twoway_options](#), excluding `by()`. These include options for titling the graph (see [G-3] [title_options](#)) and for saving the graph to disk (see [G-3] [saving_option](#)).

`level(#)` specifies the confidence level, as a percentage, for CIs. The default is `level(95)` or as set by `set level`; see [U] [20.8 Specifying the width of confidence intervals](#).

`sci` or `sci(sci_options)` plots the simultaneous confidence intervals (SCIs) using the multiplier bootstrap method proposed in [Callaway and Sant'Anna \(2021\)](#). SCIs simultaneously cover the true values of aggregations of ATETs with a predefined probability level. By default, specifying `sci` implies using 999 bootstrap replications to construct the SCIs.

`sci(sci_options)` specifies the number of replications and the seed for the multiplier bootstrap when computing SCIs. *sci_options* may be `rseed(#)` or `reps(#)`. For the definition of these options, see [Options for estat sci](#).

`sci` or `sci()` is not allowed after the TWFE estimator in `hddidregress` and `xthddidregress`.

By default, `estat` aggregation plots the pointwise CIs if option `graph()` is specified.

Option for estat moderation (StataNow)

`level(#)` specifies the confidence level, as a percentage, for CIs. The default is `level(95)` or as set by `set level`; see [U] [20.8 Specifying the width of confidence intervals](#).

Options for estat sci

`level(#)` specifies the confidence level, as a percentage, for CIs. The default is `level(95)` or as set by `set level`; see [U] [20.8 Specifying the width of confidence intervals](#).

`rseed(#)` sets the random-number seed. Specifying this option makes the results reproducible because the critical values are drawn from a bootstrap sample.

`reps(#)` specifies the number of bootstrap replications to get the critical values of the test. The default is `reps(999)`.

Remarks and examples

For examples of the `estat` commands above, see [CAUSAL] [hddidregress](#) and [CAUSAL] [xthddidregress](#). Both entries have examples that illustrate how the estimation and postestimation commands work together.

Stored results

`estat ptrends` stores the following in `r()`:

Scalars

<code>r(F)</code>	F statistic
<code>r(chi2)</code>	χ^2
<code>r(df)</code>	test constraints degrees of freedom
<code>r(p)</code>	two-sided p -value
<code>r(df_r)</code>	residual degrees of freedom
<code>r(drop)</code>	1 if constraints were dropped, 0 otherwise

`estat atetplot` stores the following in `r()`:

Matrices

<code>r(table)</code>	matrix containing test statistics and critical values
-----------------------	---

`estat aggregation` stores the following in `r()`:

Scalars

<code>r(reps)</code>	number of replications
----------------------	------------------------

Macros

<code>r(agg_type)</code>	aggregation type
<code>r(weights)</code>	cohort or timecohort

Matrices

<code>r(b)</code>	coefficient vector
<code>r(V)</code>	variance–covariance matrix of the estimators
<code>r(table)</code>	matrix containing test statistics and critical values

`estat moderation` stores the following in `r()`:

Macros

<code>r(covariates)</code>	covariates for which moderating effects were computed
----------------------------	---

Matrices

<code>r(b)</code>	coefficient vector
<code>r(V)</code>	variance–covariance matrix of the estimators
<code>r(table)</code>	matrix containing test statistics and critical values

`estat sci` stores the following in `r()`:

Scalars

<code>r(reps)</code>	number of replications
----------------------	------------------------

Matrices

<code>r(table)</code>	matrix containing coefficients, bootstrap standard errors, and SCIs
-----------------------	---

Methods and formulas

Methods and formulas are presented under the following headings:

Test for all pretreatment period ATETs being zero
Aggregations for the RA, IPW, and AIPW estimators
Overall aggregation
Dynamic aggregation
Cohort aggregation
Time aggregation
Aggregations for the TWFE estimator
Moderating effects (StataNow)
SCIs

Test for all pretreatment period ATETs being zero

`estat ptrends` tests that all pretreatment period ATETs are zero. This should be satisfied if both parallel trends and no anticipation of treatment hold for the pretreatment period.

For the RA, IPW, and AIPW estimators, `estat ptrends` is equivalent to a Wald test of all the pretreatment ATET estimates equaling zero. For methods and formulas on the Wald test, see [Methods and formulas](#) in [R] `test`.

Below, we will use the notation from [Methods and formulas](#) in [CAUSAL] `xthdidregress`. For the TWFE estimator, we fit the augmented model:

$$y_{it} = \eta + \sum_{g=q}^T G_{ig} \alpha_g + \sum_{s=q}^T f_s \gamma_s + \sum_{g=2}^{q-1} \sum_{s=q}^{q-1} d_{it} G_{ig} f_s \omega_{gs} + \sum_{g=q}^T \sum_{s=g}^T d_{it} G_{ig} f_s \delta_{gs} + \epsilon_{it}$$

We then jointly test if the ω_{gs} terms are zero by using `test`.

Note that `estat ptrends` after the TWFE estimator is applicable only when `hettype(timecohort)` has been specified.

Aggregations for the RA, IPW, and AIPW estimators

We denote $\text{ATET}(g, t)$ as $\theta(g, t)$. These are the parameters computed during estimation. Instead of looking at all of these parameters, we can aggregate them to explore heterogeneity in different dimensions. We denote aggregations of ATETs as θ . Regardless of whether we use cohort, time, dynamic, or overall aggregation, we can always write θ as a weighted sum of $\theta(g, t)$ as follows,

$$\theta = \sum_{g \in \mathbf{G}} \sum_{t=t_1}^{\mathbf{T}} w(g, t) \theta(g, t)$$

where $\mathbf{G}$ denotes the set of cohort values except the never-treated group and $\mathbf{T}$ denotes the maximum number of the time periods. t_1 is the first period when the treatment starts. $w(g, t)$ is the weight attached to each $\theta(g, t)$.

$w(g, t)$ depends on the aggregation type and the weight type used in computation. These are the types of aggregations:

1. Overall aggregation: aggregate all the treatment effects across cohorts and posttreatment periods.
2. Dynamic aggregation: aggregate the treatment effects given a fixed length of exposure to the treatment. Commonly referred to as event studies.

3. Cohort aggregation: aggregate the treatment effects within a specific cohort across the posttreatment periods.
4. Time aggregation: aggregate the treatment effects at a specific time across the cohorts.

For each type of aggregation, there are two different weight types:

- (a) Weights $w(g, t)$ that differ by cohort and time. The weights are determined by the data used to compute each $\theta(g, t)$, which varies by cohort and time. Therefore, cohorts with more observations at a given point in time will receive more weight. This accommodates situations in which the relative size of cohorts or the number of observations for a given cohort changes over time. The `weights(timecohort)` option specifies this weighting type. For notational simplicity, we denote this framework as weighting type (a).
- (b) Weights $w(g, t)$ that differ only by cohort. Weights that vary by cohort but not over time are reasonable if the groups sampled at each point in time have similar sizes. These weights are also reasonable if the sizes of the groups sampled over time change but the relative proportions of the groups are roughly the same. The weights $w(g, t)$ in this case are computed using observations in a cohort over all time periods as a proportion of all other observations. [Callaway and Sant'Anna \(2021\)](#) propose this weighting type. The `weights(cohort)` option specifies this weighting type. For notational simplicity, we denote this framework as weighting type (b).

In the common case where the groups sampled over time are of similar sizes, both weighting types give equivalent results.

Before going into details about each type of aggregation combined with each weighting type, we need to explain some notations for later use.

1. $G \in \mathbf{G}$ is the random variable that indicates the cohort. g denotes a specific cohort, and $\mathbf{I}(G_i = g)$ is an indicator function that equals 1 if the i th observation belongs to cohort g and 0 otherwise.
2. $T \in \{t_1, \dots, \mathbf{T}\}$ is the random variable that indicates the time period. t is a specific time period when a unit is observed, and $\mathbf{I}(T_i = t)$ equals 1 if the i th observation is observed at time t and 0 otherwise.
3. Let $S_i(g, t)$ be a function equal to 1 if the i th observation is observed in a time period that is used in estimating $\theta(g, t)$ and 0 otherwise. For a repeated cross-section, $S_i(g, t) = \mathbf{I}(T_i = t \text{ or } T_i = g - 1)$, which equals 1 if the i th observation is observed in time t or time $g - 1$ and 0 otherwise. For panel data, $S_i(g, t) = \mathbf{I}(T_i = g - 1)$, which equals 1 if the i th observation is observed in time $g - 1$ and 0 otherwise.
4. Let $N(g, t)$ be the number of observations in cohort g at time t , given by

$$N(g, t) = \sum_{i=1}^N \mathbf{I}(G_i = g) \mathbf{I}(T_i = t)$$

where N is the total number of observations.

5. Let $M(g, t)$ be the number of observations in cohort g for which $S(g, t) = 1$. Thus, $M(g, t) = N(g, t) + N(g, g - 1)$ for a repeated cross-section, and $M(g, t) = N(g, g - 1)$ for panel data. Generally,

$$M(g, t) = \sum_{i=1}^N \mathbf{I}(G_i = g) \mathbf{I}(S_i(g, t) = 1)$$

6. Let N_g be the number of observations belonging to cohort g in the sample,

$$N_g = \sum_{i=1}^N \mathbf{I}(G_i = g)$$

Overall aggregation

The overall aggregation aggregates all the treatment effects across all the treated cohorts in the post-treatment periods.

The overall aggregation under weighting type (a) is defined as

$$\theta_o^{(a)} = \sum_{g \in \mathbf{G}} \sum_{t=t_1}^{\mathbf{T}} w_o^{(a)}(g, t) \theta(g, t)$$

where the overall aggregation weight $w_o^{(a)}(g, t)$ is

$$w_o^{(a)}(g, t) = \frac{1}{\kappa_o^{(a)}} \mathbf{I}(t \geq g) \Pr(G = g, S(g, t) = 1 | G \leq \mathbf{T})$$

and $\kappa_o^{(a)} = \sum_{g' \in \mathbf{G}} \sum_{t'=t_1}^{\mathbf{T}} \mathbf{I}(t' \geq g') \Pr(G = g', S(g', t') = 1 | G \leq \mathbf{T})$. The term $\Pr(G = g, S(g, t) = 1 | G \leq \mathbf{T})$ is the proportion of cohort- g observations with $S(g, t) = 1$ among all the treated cohorts. The term $\kappa_o^{(a)}$ is a rescaling factor to ensure that the sum of the aggregation weights equals one. We estimate the overall weight $w_o^{(a)}(g, t)$ as

$$\widehat{w_o^{(a)}}(g, t) = \frac{\mathbf{I}(t \geq g) M(g, t)}{\sum_{g' \in \mathbf{G}} \sum_{t'=t_1}^{\mathbf{T}} \mathbf{I}(t' \geq g') M(g', t')}$$

$\theta_o^{(a)}$ is computed when the `overall` and `weight(timecohort)` options are specified.

The overall aggregation under weighting type (b) is defined as

$$\theta_o^{(b)} = \sum_{g \in \mathbf{G}} \sum_{t=t_1}^{\mathbf{T}} w_o^{(b)}(g, t) \theta(g, t)$$

where $w_o^{(b)}(g)$ is

$$w_o^{(b)}(g, t) = \frac{1}{\kappa_o^{(b)}} \mathbf{I}(t \geq g) \Pr(G = g | G \leq \mathbf{T})$$

and $\kappa_o^{(b)} = \sum_{g' \in \mathbf{G}} \sum_{t'=t_1}^{\mathbf{T}} \mathbf{I}(t' \geq g') \Pr(G = g' | G \leq \mathbf{T})$. The term $\Pr(G = g | G \leq \mathbf{T})$ is the proportion of cohort g among all the treated cohorts in the full sample. The overall aggregation weight is estimated as

$$\widehat{w_o^{(b)}}(g, t) = \frac{\mathbf{I}(t \geq g) N_g}{\sum_{g' \in \mathbf{G}} \sum_{t'=t_1}^{\mathbf{T}} \mathbf{I}(t' \geq g') N_{g'}}$$

$\theta_o^{(b)}$ is computed when the `overall` and `weight(cohort)` options are specified.

Dynamic aggregation

One popular question in a DID setup with multiple time periods has to do with the dynamics of treatment effects: how do the average treatment effects vary with the length of exposure to the treatment? In the literature, this is also known as an event study. Let $e = t - g$ be the length of exposure to the treatment.

Under weighting type (a), the dynamic aggregation given e is defined as

$$\theta_d^{(a)}(e) = \sum_{g \in \mathbf{G}} w_d^{(a)}(g, g + e) \theta(g, g + e)$$

where $w_d^{(a)}(g, g + e)$ is defined as

$$w_d^{(a)}(g, g + e) = \frac{1}{\kappa_d^{(a)}} \mathbf{I}(g + e \leq \mathbf{T}) \Pr(G = g, S(g, g + e) = 1 | G + e \leq \mathbf{T})$$

and $\kappa_d^{(a)} = \sum_{g' \in \mathbf{G}} \mathbf{I}(g' + e \leq \mathbf{T}) \Pr(G = g', S(g', g' + e) = 1 | G + e \leq \mathbf{T})$. $\kappa_d^{(a)}$ is a rescaling factor to ensure the sum of the aggregation weights equals one. We estimate the weight $w_d^{(a)}(g, g + e)$ as

$$\widehat{w_d^{(a)}}(g, g + e) = \frac{\mathbf{I}(g + e \leq \mathbf{T}) M(g, g + e)}{\sum_{g' \in \mathbf{G}} \mathbf{I}(g' + e \leq \mathbf{T}) M(g', g' + e)}$$

$\theta_d^{(a)}(e)$ is computed when the `dynamic` and `weight(timecohort)` options are specified.

Under weighting type (b), the dynamic aggregation given e is defined as

$$\theta_d^{(b)}(e) = \sum_{g \in \mathbf{G}} w_d^{(b)}(g, g + e) \theta(g, g + e)$$

where

$$w_d^{(b)}(g, g + e) = \mathbf{I}(g + e \leq \mathbf{T}) \Pr(G = g | G + e \leq \mathbf{T})$$

We estimate the weight $w_d^{(b)}(g, g + e)$ as

$$\widehat{w_d^{(b)}}(g, g + e) = \frac{\mathbf{I}(g + e \leq \mathbf{T}) N_g}{\sum_{g' \in \mathbf{G}} \mathbf{I}(g' + e \leq \mathbf{T}) N_{g'}}$$

$\theta_d^{(b)}(e)$ is computed when the `dynamic` and `weight(cohort)` options are specified.

Cohort aggregation

To account for the heterogeneity of treatment effects across cohorts, we consider the cohort aggregation.

Under weighting type (a), the cohort aggregation for cohort g is defined as

$$\theta_c^{(a)}(g) = \sum_{t=t_1}^T w_c^{(a)}(g, t) \theta(g, t)$$

where the aggregation weight $w_c^{(a)}(g, t)$ is

$$w_c^{(a)}(g, t) = \frac{1}{\kappa_c^{(a)}} \mathbf{I}(t \geq g) \Pr(G = g, S(g, t) = 1)$$

with $\kappa_c^{(a)} = \sum_{t'=t_1}^T \mathbf{I}(t' \geq g) \Pr(G = g, S(g, t') = 1)$. We estimate the weight $w_c^{(a)}(g, t)$ as

$$\widehat{w_c^{(a)}}(g, t) = \frac{\mathbf{I}(t \geq g)M(g, t)}{\sum_{t'=2}^T \mathbf{I}(t' \geq g)M(g, t')}$$

$\theta_c^{(a)}(g)$ is computed when the `cohort` and `weight(timecohort)` options are specified.

Under weighting type (b), the cohort aggregation for cohort g is defined as

$$\theta_c^{(b)}(g) = \sum_{t=t_1}^T w_c^{(b)}(g) \theta(g, t)$$

where $w_c^{(b)}$ is

$$w_c^{(b)}(g, t) = \frac{1}{T - g + 1} \mathbf{I}(t \geq g)$$

$\theta_c^{(b)}(g)$ is computed when the `cohort` and `weight(cohort)` options are specified.

Time aggregation

Time effects characterize treatment-effect heterogeneity over time. Under weighting type (a), the average effect of participating in the treatment in period t (among cohorts that are treated by time t) is

$$\theta_t^{(a)}(t) = \sum_{g \in \mathbf{G}} w_t^{(a)}(g, t) \theta(g, t)$$

where $w_t^{(a)}(g, t)$ is

$$w_t^{(a)}(g, t) = \frac{1}{\kappa_t^{(a)}} \mathbf{I}(t \geq g) \Pr(G = g, S(g, t) = 1 | G \leq t)$$

with $\kappa_t^{(a)} = \sum_{g' \in \mathbf{G}} \mathbf{I}(t \geq g') \Pr(G = g', S(g', t) = 1 | G \leq t)$. We estimate $w_t^{(a)}(g, t)$ as

$$\widehat{w_t^{(a)}}(g, t) = \frac{\mathbf{I}(t \geq g)M(g, t)}{\sum_{g' \in \mathbf{G}} \mathbf{I}(t \geq g')M(g', t)}$$

$\theta_t^{(a)}(t)$ is computed when the `time` and `weight(timecohort)` options are specified.

Under weighting type (b), the time aggregation is defined as

$$\theta_t^{(b)}(t) = \sum_{g \in \mathbf{G}} w_t^{(b)}(g, t) \theta(g, t)$$

where $w_t^{(b)}(g, t)$ is

$$w_t^{(b)}(g, t) = \mathbf{I}(t \geq g) \Pr(G = g | G \leq t)$$

We estimate $w_t^{(b)}(g, t)$ as

$$\widehat{w_t^{(b)}}(g, t) = \frac{\mathbf{I}(t \geq g)N_g}{\sum_{g' \in \mathbf{G}} \mathbf{I}(t \geq g')N_{g'}}$$

$\theta_t^{(b)}(t)$ is computed when the `time` and `weight(cohort)` options are specified.

The variance–covariance matrix for the estimates of θ is computed using the influence function approach outlined in section 4.2 of [Callaway and Sant’Anna \(2021\)](#).

When the `sci` option is specified, the SCIs are computed using the multiplier bootstrap proposed in section 4.2 in [Callaway and Sant’Anna \(2021\)](#).

Aggregations for the TWFE estimator

Aggregation after TWFE uses [margins](#) after the Mundlak estimation of the model. Let `treat` denote the observation-level treatment, `cohort` denote the variable that contains treat-time cohorts, and `exposure` denote a variable that indicates the time exposed to treatment.

For estat aggregation, overall:

```
. margins r.treat, subpop(if treat==1) vce(unconditional)
```

For estat aggregation, cohort:

```
. margins, subpop(if treat==1) dydx(treat) over(cohort) vce(unconditional)
```

For estat aggregation, time:

```
. margins, subpop(if treat==1) dydx(treat) over(time) vce(unconditional)
```

For estat aggregation, dynamic:

```
. margins, subpop(if treat==1) dydx(treat) over(exposure) vce(unconditional)
```

Moderating effects (StataNow)

After using the TWFE estimator to model the outcome using a set of covariates, `estat moderation` reports estimates of moderating effects, which can be interpreted as the marginal changes in the ATET with respect to each covariate. For factor variables, the marginal effect is the difference between the margin for the specified level and the margin for the base level.

Estimation of moderating effects uses [margins](#) after the Mundlak estimation of the model. Let `treat` denote the observation-level treatment, `cohort` denote the variable that contains treat-time cohorts, `time` denote the time variable, and `x1` and `x2` denote covariates that appear in the model.

When `hettype(timecohort)` and `over(cohort)` are specified, the moderating effects are computed as follows:

```
. margins r.treat, dydx(x1 x2) over(cohort) at(time=q ... time=T)
vce(unconditional)
```

When `hettype(timecohort)` and `over(timecohort)` are specified:

```
. margins r.treat, dydx(x1 x2) over(cohort time) vce(unconditional)
```

When `hettype(cohort)` is specified:

```
. margins r.treat, dydx(x1 x2) over(cohort) vce(unconditional)
```

When `hettype(time)` is specified:

```
. margins r.treat, dydx(x1 x2) over(time) vce(unconditional)
```

By default, estimates are returned for each covariate that appears in the model. If *varlist* is specified with *estat moderation*, effects are returned only for the specified covariates. Note that if a covariate *x1* appears as both a factor variable and a continuous variable in the model, moderating effects for *x1* cannot be computed. *estat moderation* is not allowed after estimation with the *noxiinteract* option.

SCIs

After the RA, IPW, and AIPW estimators, *estat sci* can provide the SCIs that are guaranteed to cover all the ATETs with a specified probability. *estat sci* computes the SCIs using the multiplier bootstrap approach outlined in section 4.1 in Callaway and Sant'Anna (2021).

Unlike the traditional bootstrap, the multiplier bootstrap resamples the influence functions (which are already computed in the estimation step). Thus, the multiplier bootstrap is much faster than the traditional bootstrap because there is no need to recompute the estimators.

The influence function is a linear representation of the estimator. Let $\hat{\theta}(g, t)$ be the RA, IPW, and AIPW estimators, and denote $\theta(g, t)$ the true ATET for cohort *g* at time *t*. Then the linear representation of these estimators can be written as

$$\hat{\theta}(g, t) - \theta(g, t) = \frac{1}{n} \sum_{i=1}^n \psi_{g,t}(\mathbf{w}_i) + o_p(1)$$

where $\psi_{g,t}()$ is the influence function, *n* is the sample size of the estimation sample for $\hat{\theta}(g, t)$, $\mathbf{w}_i$ are the data, and $o_p(1)$ is a term that vanishes to zero in probability as *n* grows. For a more detailed discussion on influence functions, see section 4.1 in Callaway and Sant'Anna (2021).

Denote $\hat{\theta}$ as estimates of all the ATETs, and let $\hat{\Psi}$ be estimates of the influence functions for $\hat{\theta}$. Let $\hat{\theta}^b$ be the *b*th bootstrap draw, which is defined as

$$\hat{\theta}^b = \hat{\theta} + \frac{1}{n} \sum_{i=1}^n V_i \cdot \hat{\Psi}_i$$

where $\{V_i\}$ is a Bernoulli draw with $P(V = 1 - \eta) = \eta/\sqrt{5}$, $P(V = \eta) = 1 - \eta/\sqrt{5}$ and $\eta = (\sqrt{5} + 1)/2$. Then the SCIs can be computed in the following steps:

1. Draw *B* samples of $\{V_i\}_{i=1, \dots, n}$, and compute $\hat{\theta}^b$ using each sample.
2. Compute the bootstrap diagonal of $\Sigma^{1/2}$ as

$$\hat{\Sigma}_{g,t}^{1/2} = \frac{q_{0.75}(g, t) - q_{0.25}(g, t)}{z_{0.75} - z_{0.25}}$$

where $q_p(g, t)$ is the *p*th sample quantile of $\hat{R}_{g,t}^b = \sqrt{n} \{ \hat{\theta}^b(g, t) - \hat{\theta}(g, t) \}$ in *B* draws and z_p is the *p*th sample quantile of standard normal distribution.

3. For each bootstrap draw, compute the *t* test^{*b*} as

$$t \text{ test}^b = \max_{(g,t)} |\hat{R}_{g,t}^b| \hat{\Sigma}_{g,t}^{-1/2}$$

4. Compute the critical values $\hat{c}_{1-\alpha/2}$ as the $1 - \alpha/2$ quantile of the *B* draws of *t* test^{*b*}.
5. Construct the simultaneous bootstrap confidence intervals for $\hat{\theta}(g, t)$ as

$$\hat{C}(g, t) = \left\{ \hat{\theta}(g, t) - \hat{c}_{1-\alpha/2} \hat{\Sigma}_{g,t}^{1/2} / \sqrt{n}, \quad \hat{\theta}(g, t) + \hat{c}_{1-\alpha/2} \hat{\Sigma}_{g,t}^{1/2} / \sqrt{n} \right\}$$

Reference

Callaway, B., and P. H. C. Sant’Anna. 2021. Difference-in-differences with multiple time periods. *Journal of Econometrics* 225: 200–230. <https://doi.org/10.1016/j.jeconom.2020.12.001>.

Also see

[CAUSAL] **hdidregress** — Heterogeneous difference in differences⁺

[CAUSAL] **xthdidregress** — Heterogeneous difference in differences for panel data⁺

[CAUSAL] **DID intro** — Introduction to difference-in-differences estimation

[U] **20 Estimation and postestimation commands**

⁺This command is part of [StataNow](#).

[Description](#)
[References](#)

[Syntax](#)
[Also see](#)

[Remarks and examples](#)

[Methods and formulas](#)

Description

The `latebalance` postestimation commands produce diagnostic statistics, test statistics, and diagnostic plots to assess whether a `lateffects` command balanced the covariates over treatment-assignment levels. `latebalance` is not available after `lateffects` balancing.

Syntax

`latebalance subcommand ... [ , options ]`

<i>subcommand</i>	Description
summarize	compare means and variances in raw and balanced data
overid	overidentification test
density	kernel density plots for raw and balanced data

Remarks and examples

This entry provides an overview of the commands in `latebalance`. We recommend that you read this entry before proceeding to [\[CAUSAL\] latebalance summarize](#), [\[CAUSAL\] latebalance overid](#), or [\[CAUSAL\] latebalance density](#) for command-specific syntax and details.

A covariate is said to be balanced when its distribution does not vary over treatment-assignment levels.

Covariates are balanced in experimental data with full compliance because treatment assignment is independent of the covariates because of the study design. In contrast, covariates must be balanced by weighting or matching in observational data if treatment assignment is related to the covariates that also affect the outcome of interest.

The estimators implemented in `lateffects` use a model to make the potential outcomes independent of the treatment assignment conditional on covariates. If the model is well specified, it should balance the covariates. Balance diagnostic techniques and tests check the specification of the conditioning method used by a `lateffects` estimator.

`latebalance` implements three methods to check for balance after `lateffects`: `latebalance summarize`, `latebalance overid`, and `latebalance density`. `latebalance overid` implements a formal test, while the other two methods are exploratory diagnostic techniques.

[Austin \(2009, 2011\)](#) and [Guo and Fraser \(2015, sec. 5.52\)](#) provide introductions to covariate balance. [Imai and Ratkovic \(2014\)](#) derived a test for balance implemented in `latebalance overid`. For an illustration, see [example 2](#) of [\[CAUSAL\] lateffects](#).

Methods and formulas

Methods and formulas are presented under the following headings:

[Introduction](#)

[Testing the propensity-score model specification](#)

Introduction

For covariate x , we observe values $\{x_1, x_2, \dots, x_N\}$. Define a treatment-assignment indicator variable for treatment levels 0 and 1 as $t_i \in \{0, 1\}$, for $i = 1, \dots, N$, and frequency weights as $\{w_1, w_2, \dots, w_N\}$. The sample mean and variance of x for level t are

$$\hat{\mu}_x(t) = \frac{\sum_i^N I(t_i = t) w_i x_i}{N_t} \quad \text{and}$$

$$\hat{\sigma}_x^2(t) = \frac{\sum_i^N I(t_i = t) w_i \{x_i - \hat{\mu}_x(t)\}^2}{N_t - 1}$$

where $N_t = \sum_i^N w_i I(t_i = t)$, and

$$I(t_i = t) = \begin{cases} 1 & \text{if } t_i = t \\ 0 & \text{otherwise} \end{cases}$$

As shown in [Austin \(2011\)](#), the standardized differences for covariate x between level t and the control t_0 are computed as

$$\delta_x(t) = \frac{\hat{\mu}_x(t) - \hat{\mu}_x(t_0)}{\sqrt{\frac{\hat{\sigma}_x^2(t) + \hat{\sigma}_x^2(t_0)}{2}}}$$

The variance ratio is $\rho_x(t) = \{\hat{\sigma}_x^2(t)\} / \{\hat{\sigma}_x^2(t_0)\}$.

The kernel density estimates from `latebalance` density are computed by `kdensity` using the normalized instrument inverse propensity scores for each treatment level.

Testing the propensity-score model specification

We estimate the probability of treatment conditioned on a set of covariates with a propensity-score model. [Imai and Ratkovic \(2014\)](#) derive a test for whether the estimated propensity score balances the covariates. The score equations for parameters of the propensity-score model define an exactly identified generalized method of moments (GMM) estimator. [Imai and Ratkovic \(2014\)](#) use the conditions imposed by mean balance as overidentifying conditions. A standard GMM test for the validity of the overidentifying conditions is then a test for covariate balance. See [\[R\] gmm](#) for a discussion of this overidentifying test, which is known as Hansen's J test in the econometrics literature.

Here are the details about the score equations and the overidentifying balance conditions. Recall from [Methods and formulas](#) of [\[CAUSAL\] teffects aipw](#) that we have the first-order condition of the treatment model

$$\frac{1}{N} \sum_{i=1}^N \mathbf{s}_{\text{tm},i}(\mathbf{x}_i, \hat{\gamma}) = 0$$

For a two-level treatment-effects model with conditional treatment $\tilde{t}$ and control t_0 , the score is

$$\mathbf{s}_{\text{tm},i}(\mathbf{x}_i, \gamma) = \frac{I(t_i = \tilde{t})}{p(\mathbf{x}_i, \tilde{t}, \gamma)} \frac{\partial p(\mathbf{x}_i, t, \gamma)}{\partial \gamma'} - \left\{ \frac{I(t_i = t_0)}{1 - p(\mathbf{x}_i, \tilde{t}, \gamma)} \right\} \frac{\partial p(\mathbf{x}_i, \tilde{t}, \gamma)}{\partial \gamma'} \bigg|_{\gamma = \hat{\gamma}}$$

The score reduces to

$$\mathbf{s}_{\text{tm},i}(\mathbf{x}_i, \hat{\gamma}) = \left[\frac{I(t_i = \tilde{t}) - p(\mathbf{x}_i, \tilde{t}, \gamma)}{p(\mathbf{x}_i, \tilde{t}, \gamma) \{1 - p(\mathbf{x}_i, \tilde{t}, \gamma)\}} \right] \frac{\partial p(\mathbf{x}_i, \tilde{t}, \gamma)}{\partial \gamma'} \bigg|_{\gamma = \hat{\gamma}}$$

The corresponding covariate balancing moment conditions are

$$\mathbf{w}_{\text{tm},i}(\mathbf{x}_i, \gamma) = \left[\frac{I(t_i = \tilde{t}) - p(\mathbf{x}_i, \tilde{t}, \gamma)}{p(\mathbf{x}_i, \tilde{t}, \gamma) \{1 - p(\mathbf{x}_i, \tilde{t}, \gamma)\}} \right] \mathbf{x}_i$$

We stack the moment conditions

$$\begin{aligned} \mathbf{g}_{\text{tm}}(\mathbf{X}, \gamma) &= \frac{1}{N} \sum_{i=1}^N \begin{Bmatrix} \mathbf{s}_{\text{tm},i}(\mathbf{x}_i, \gamma) \\ \mathbf{w}_{\text{tm},i}(\mathbf{x}_i, \gamma) \end{Bmatrix} \\ &= \frac{1}{N} \sum_{i=1}^N \mathbf{g}_{\text{tm},i}(\mathbf{x}_i, \gamma) \end{aligned}$$

The overidentified GMM estimator is then

$$\tilde{\gamma} = \underset{\gamma}{\operatorname{argmin}} N \mathbf{g}_{\text{tm}}(\mathbf{X}, \gamma)' \mathbf{W}_{\text{tm}}(\mathbf{X}, \gamma)^{-1} \mathbf{g}_{\text{tm}}(\mathbf{X}, \gamma) \quad (1)$$

where

$$\mathbf{W}_{\text{tm}}(\mathbf{X}, \gamma) = \frac{1}{N} \sum_{i=1}^N E_T \{ \mathbf{g}_{\text{tm},i}(\mathbf{x}, \gamma) \mathbf{g}_{\text{tm},i}(\mathbf{x}, \gamma)' \}$$

and the expectation is taken with respect to treatment distribution. The weight matrix $\mathbf{W}_{\text{tm}}(\mathbf{X}, \gamma)$ is computed explicitly (Imai and Ratkovic 2014), and (1), written as a maximization problem, is solved using `m1`.

Finally, Hansen's J statistic is evaluated at its minimum,

$$J = N \mathbf{g}_{\text{tm}}(\mathbf{X}, \tilde{\gamma})' \mathbf{W}_{\text{tm}}(\mathbf{X}, \tilde{\gamma})^{-1} \mathbf{g}_{\text{tm}}(\mathbf{X}, \tilde{\gamma})$$

and is asymptotically distributed as a χ^2 with degrees of freedom d ,

$$d = \operatorname{rank} \{ \mathbf{W}_{\text{tm}}(\mathbf{X}, \tilde{\gamma}) \} - \operatorname{rank} \left[\frac{1}{N} \sum_{i=1}^N E_T \{ \mathbf{s}_{\text{tm},i}(\mathbf{x}_i, \tilde{\gamma}) \mathbf{s}_{\text{tm},i}(\mathbf{x}_i, \tilde{\gamma})' \} \right]$$

References

Austin, P. C. 2009. Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. *Statistics in Medicine* 28: 3083–3107. <https://doi.org/10.1002/sim.3697>.

- . 2011. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate Behavioral Research* 46: 399–424. <https://doi.org/10.1080/00273171.2011.568786>.
- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: Sage.
- Imai, K., and M. Ratkovic. 2014. Covariate balancing propensity score. *Journal of the Royal Statistical Society, B ser.*, 76: 243–263. <https://doi.org/10.1111/rssb.12027>.
- Jann, B. 2021. [Relative distribution analysis in Stata](#). *Stata Journal* 21: 885–951.

Also see

- [CAUSAL] [latebalance density](#) — Covariate balance density⁺
- [CAUSAL] [latebalance overid](#) — Test for covariate balance⁺
- [CAUSAL] [latebalance summarize](#) — Covariate-balance summary statistics⁺
- [CAUSAL] [lateffects](#) — Local average treatment-effect estimation⁺

⁺This command is part of [StataNow](#).

Description	Quick start	Menu	Syntax	Options
Remarks and examples	Stored results	Reference	Also see	

Description

latebalance density produces kernel density plots that are used to check for covariate balance after estimation by [lateeffects ipwra](#) or [lateeffects kappa](#).

Quick start

Kernel density plot of x1 after lateeffects

```
latebalance density x1
```

Same as above, but rescale the kernel bandwidth by a factor of 2

```
latebalance density x1, bwidth(*2)
```

Menu

Statistics > Postestimation

Syntax

```
latebalance density varname [ , options ]
```

<i>options</i>	Description
Main	
<code>kernel</code> (<i>kernel</i>)	specify the kernel function; default is <code>kernel(epanechnikov)</code>
<code>line#opts</code> (<i>line_options</i>)	twoway line options for density line number #
<code>bwidth</code> (*#)	rescale default bandwidth
Y axis, X axis, Titles, Legend, Overall	
<i>twoway_options</i>	any options other than <code>by()</code> documented in [G-3] twoway_options
By options	
<code>byopts</code> (<i>by_option</i>)	how subgraphs are combined, labeled, etc.
collect is allowed; see [U] 11.1.10 Prefix commands .	

kernel	Description
<code>epanechnikov</code>	Epanechnikov kernel function; the default
<code>triangle</code>	triangle kernel function
<code>epan2</code>	alternative Epanechnikov kernel function
<code>biweight</code>	biweight kernel function
<code>cosine</code>	cosine trace kernel function
<code>gaussian</code>	Gaussian kernel function
<code>parzen</code>	Parzen kernel function
<code>rectangle</code>	rectangle kernel function

Options

Main

`kernel` (*kernel*) specifies the kernel function for use in calculating the kernel density estimates. The default kernel is `kernel(epanechnikov)`.

`line#opts` (*line_options*) specifies the line pattern, width, color, and overall style of density line number `#`. The line numbers are in the same order as the treatment levels specified in `e(tlevels)`.

`bwidh(*#)` specifies the factor by which the default bandwidths are to be rescaled. A bandwidth is the half-width of the kernel, the width of the density window around each point. Each kernel density plot has its own bandwidth, and by default, each kernel density plot uses its own optimal bandwidth; see [\[R\] `kdensity`](#). `bwidh()` rescales each plot’s optimal bandwidth by the specified amount.

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [\[G-3\] *twoway_options*](#), excluding `by()`. These include options for titling the graph (see [\[G-3\] *title_options*](#)) and for saving the graph to disk (see [\[G-3\] *saving_option*](#)). `latebalance density` uses `by()` to differentiate between raw and weighted or matched samples, and some *twoway_options* will be repeated for `by` graph and might be better specified as `byopts()`.

By options

`byopts` (*by_option*) is as documented in [\[G-3\] *by_option*](#). `byopts()` affects how the subgraphs are combined, labeled, etc. `byopts()` generally affects the entire graph, and some *by_options* may be better specified as *twoway_options*; see [\[G-3\] *twoway_options*](#).

Remarks and examples

When the distribution of a covariate does not vary over the assigned treatment levels, the covariate is said to be balanced. `latebalance density` produces kernel density plots of a covariate over assigned treatment levels for the raw data and the weighted data produced by `lateeffects`. If the weighted-sample kernel density plots of the covariate are the same over the assigned treatment levels, the covariate is balanced in the weighted sample.

We recommend that you read [\[CAUSAL\] *latebalance*](#) before proceeding because it provides an introduction to covariate balance. For an illustration, see [example 2](#) of [\[CAUSAL\] *lateeffects*](#).

Stored results

latebalance density stores the following in `r()`:

Scalars

<code>r(bwc_adj)</code>	bandwidth for control in weighted sample
<code>r(Nc_adj)</code>	observations on control in weighted sample
<code>r(bwt_adj)</code>	bandwidth for treated in weighted sample
<code>r(Nt_adj)</code>	observations on treated in weighted sample
<code>r(bwc_raw)</code>	bandwidth for control in raw sample
<code>r(Nc_raw)</code>	observations on control in raw sample
<code>r(bwt_raw)</code>	bandwidth for treated in raw sample
<code>r(Nt_raw)</code>	observations on treated in raw sample

Macros

<code>r(kernel)</code>	name of kernel
------------------------	----------------

Reference

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.

Also see

[CAUSAL] [latebalance](#) — Check balance after lateeffects estimation⁺

[CAUSAL] [lateeffects](#) — Local average treatment-effect estimation⁺

[CAUSAL] [lateoverlap](#) — Overlap plots⁺

⁺This command is part of [StataNow](#).

[Description](#)
[Remarks and examples](#)

[Quick start](#)
[Stored results](#)

[Menu](#)
[References](#)

[Syntax](#)
[Also see](#)

[Options](#)

Description

latebalance overid performs a test for covariate balance after estimation by [lateffects ipwra](#) or [lateffects kappa](#).

Quick start

Test for covariate balance after lateffects estimation

```
latebalance overid
```

Same as above, but test for balance only in base covariates and exclude interaction terms

```
latebalance overid, bconly
```

Menu

Statistics > Postestimation

Syntax

```
latebalance overid [ , bconly nolog iterate(#) ]
```

collect is allowed; see [\[U\] 11.1.10 Prefix commands](#).

Options

Main

bconly specifies that only the base covariates be included in the test for balance. By default, the powers and interactions specified by factor-variable notation in the lateffects model are also included in the test for balance.

nolog suppresses the display of the optimization search log.

iterate(#) sets the maximum number of iterations to # in the generalized method of moments estimator used to compute the test statistic.

Remarks and examples

When the distribution of a covariate is the same for all assigned treatment levels, the covariate is said to be balanced. latebalance overid performs a test to see whether the covariates are balanced after lateffects. If the treatment-assignment model is well specified, instrumental-variables propensity-score functions of the covariates from the model are balanced.

We recommend that you read [\[CAUSAL\] latebalance](#) for an introduction to covariate balance and an overview of the implemented methods.

Stored results

latebalance overrides stores the following in `r()`:

Scalars

<code>r(p)</code>	<i>p</i> -value
<code>r(df)</code>	overidentifying constraints, test degrees of freedom
<code>r(chi2)</code>	χ^2 statistic

References

- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Imai, K., and M. Ratkovic. 2014. Covariate balancing propensity score. *Journal of the Royal Statistical Society, B ser.*, 76: 243–263. <https://doi.org/10.1111/rssb.12027>.

Also see

- [CAUSAL] [latebalance](#) — Check balance after lateffects estimation⁺
- [CAUSAL] [lateffects](#) — Local average treatment-effect estimation⁺
- [CAUSAL] [lateoverlap](#) — Overlap plots⁺

⁺This command is part of [StataNow](#).[Description](#)[Remarks and examples](#)[Quick start](#)[Stored results](#)[Menu](#)[Reference](#)[Syntax](#)[Also see](#)[Option](#)

Description

`latebalance summarize` reports diagnostic statistics that are used to check for covariate balance over treatment-assignment groups after estimation by [lateffects ipwra](#) or [lateffects kappa](#).

Quick start

Raw and weighted standardized differences and variance ratios of all covariates from the most recently fitted `lateffects` model

```
latebalance summarize
```

Same as above, but report statistics only for covariates `x1` and `x2`

```
latebalance summarize x1 x2
```

Baseline means and variances for units assigned to treatment and control

```
latebalance summarize, baseline
```

Menu

Statistics > Postestimation

Syntax

```
latebalance summarize [varlist] [ , baseline ]
```

varlist may contain factor variables; see [\[U\] 11.4.3 Factor variables](#).

`collect` is allowed; see [\[U\] 11.1.10 Prefix commands](#).

Option

Main

`baseline` specifies that `latebalance summarize` report means and variances by treatment level.

Remarks and examples

When the distribution of a covariate is the same for all assigned treatment levels, the covariate is said to be balanced. `latebalance summarize` reports diagnostic statistics to check for covariate balance after `lateffects`.

We recommend that you read [\[CAUSAL\] latebalance](#) before proceeding because it provides an introduction to covariate balance. For an illustration, see [example 2](#) of [\[CAUSAL\] lateffects](#).

Stored results

latebalance summarize stores the following in `r()`:

Matrices

<code>r(size)</code>	number of observations in the raw and weighted samples
<code>r(table)</code>	table of covariate statistics

Reference

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.

Also see

[CAUSAL] [latebalance](#) — Check balance after lateffects estimation⁺

[CAUSAL] [lateffects](#) — Local average treatment-effect estimation⁺

[CAUSAL] [lateoverlap](#) — Overlap plots⁺

⁺This command is part of [StataNow](#).[Description](#)[Options](#)[Acknowledgments](#)[Quick start](#)[Remarks and examples](#)[References](#)[Menu](#)[Stored results](#)[Also see](#)[Syntax](#)[Methods and formulas](#)

Description

`lateffects` estimates local average treatment effects (LATEs) by using weighting estimators. Outcomes may be continuous, binary, count, or fractional. The treatment is binary. `lateffects` provides three estimators: inverse-probability-weighted regression adjustment (IPWRA), normalized kappa, and normalized covariate balancing.

Quick start

LATE for outcome *y*, with treatment *t*, and instrument propensity score for *z* modeled using *x1* and *x2* via the normalized kappa estimator

```
lateffects kappa (y) (t) (z x1 x2)
```

Same as above, but use a normalized covariate-balancing estimator

```
lateffects balancing (y) (t) (z x1 x2)
```

LATE of treatment *t* via IPWRA estimation using a linear model for outcome *y* on *x1* and *x2*, a logistic model for *t* on *x1* and *w*, and a logistic model for the instrument propensity score of *z* on *x1*, *x2*, and *w*

```
lateffects ipwra (y x1 x2) (t x1 w) (z x1 x2 w)
```

Same as above, but use probit models for the treatment and the instrument propensity score

```
lateffects ipwra (y x1 x2) (t x1 w, probit) (z x1 x2 w, probit)
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Local average treatment effects

Statistics > Causal inference/treatment effects > Binary outcomes > Local average treatment effects

Statistics > Causal inference/treatment effects > Count outcomes > Local average treatment effects

Statistics > Causal inference/treatment effects > Fractional outcomes > Local average treatment effects

Statistics > Causal inference/treatment effects > Nonnegative outcomes > Local average treatment effects

Syntax

Inverse-probability-weighted regression adjustment

```
lateffects ipwra (ovar [omvarlist, omodel noconstant])
    (tvar [tmvarlist, tmodel noconstant])
    (iv [ivvarlist, ivmodel noconstant]) [if] [in] [weight] [, options]
```

Normalized kappa

```
lateffects kappa (ovar) (tvar)
    (iv [ivvarlist, ivmodel noconstant]) [if] [in] [weight] [, options]
```

Normalized covariate balancing

```
lateffects balancing (ovar) (tvar)
    (iv [ivvarlist, ivmodel noconstant]) [if] [in] [weight] [, options]
```

ovar is a binary, count, continuous, fractional, or nonnegative outcome of interest.

omvarlist specifies the covariates in the outcome model.

tvar is a binary variable indicating observed treatment status.

tmvarlist specifies the covariates for the treatment-status model.

iv is a binary instrumental variable indicating treatment assignment.

ivvarlist specifies covariates for the instrument propensity-score model.

<i>omodel</i>	Description
---------------	-------------

Model

linear	linear outcome model; the default
logit	logistic outcome model
probit	probit outcome model
poisson	count outcome model
flogit	fractional logistic outcome model
fprobit	fractional probit outcome model

omodel specifies the model for the outcome variable. *omodel* is available only with ipwra.

<i>tmodel</i>	Description
---------------	-------------

Model

logit	logistic treatment-status model; the default
probit	probit treatment-status model

tmodel specifies the model for treatment status. *tmodel* is available only with ipwra.

<i>ivmodel</i>	Description
Model	
<code>logit</code>	logistic instrument propensity-score model; the default
<code>probit</code>	probit instrument propensity-score model
<i>ivmodel</i> specifies the model for the instrument propensity score.	
<i>options</i>	Description
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>cluster <i>clustvar</i></code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
Advanced	
<code>pstolerance(#)</code>	set tolerance for overlap assumption
<code>osample(<i>newvar</i>)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>coeflegend</code>	display legend instead of statistics

omvarlist, *tmvarlist*, and *ivvarlist* may contain factor variables; see [U] 11.4.3 Factor variables.

`bayesboot`, `bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the `bootstrap` prefix; see [R] `bootstrap`.

`fweights`, `awweights`, `iweights`, and `pweights` are allowed; see [U] 11.1.6 weight.

`coeflegend` does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] `vce_option`.

Reporting

`level(#)`; see [R] `Estimation options`.

`aequations` specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no!stretch`; see [R] `Estimation options`.

Maximization

`maximize_options`: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `lateffects` will exit with an error if an observation has an estimated propensity score smaller than `#` or larger than `1 - #`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption.

The following option is available with `lateffects` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

A local average treatment effect (LATE) is an average treatment effect for a subpopulation. We would usually prefer to identify an effect for the entire population, but in many instances, this is not feasible. In the LATE framework, we cannot identify a treatment effect for the population because of unobservable differences between treated units and untreated units. Unaccounted for, unobservable differences confound any causal effect we would like to identify (see [\[CAUSAL\] Intro](#)). [Imbens and Angrist \(1994\)](#) and [Angrist, Imbens, and Rubin \(1996\)](#) illustrate how, if there is a binary variable that splits the population into treated and untreated units, as if they were randomly assigned, we may identify a treatment effect for those that comply with the treatment assignment, a LATE. Because the effect is identified only for compliers, sometimes the estimand is referred to as the complier average treatment effect.

In an experimental setting, where units are assigned to treatment or control, “compliance” is readily understood as a unit’s adherence to the treatment status to which it was assigned, for instance, those assigned to be treated who take the medicine and those assigned to be controls who take the placebo. In a context of observational data, however, units often self-select into their treatment status, and the treatment-assignment variable of interest can only encourage, persuade, or otherwise motivate units into a treatment status. Some units, the “always takers”, will opt for the treatment regardless of whether they receive the motivation. Others, the “never takers”, will always opt out of the treatment. “Defiers” will opt out of treatment when motivated and opt in when motivation is absent. Finally, “compliers” opt for the treatment when they are motivated and opt out when they are not; this is the target population of the LATE estimate, which we now define.

In the LATE framework, treatment can be thought of as a potential outcome. To exemplify, let d denote binary treatment and let z be the binary instrument that assigns or motivates units into their treatment status. We can express treatment as

$$d = d(1)z + d(0)(1 - z)$$

That is, we observe $d(1)$ if $z = 1$ and $d(0)$ if $z = 0$. $d(1)$ might be 1 or it might be 0. Realizations of the random variable d for which $d(1) = 1$ when $z = 1$ can arise from two types of behavior. They could be those that always opt into treatment so that $d(1) = 1$ regardless of z . Alternatively, they could be realizations for which $d(1) = 1$ only when $z = 1$. Similarly, realizations for which $d(0) = 0$ when $z = 0$ either never opt into treatment (so that $d(0) = 0$ regardless of z) or opt out only when

not assigned to treatment. In this framework, compliers are those that modify their behavior based on treatment assignment. The instrument affects their decisions. For them, $d(1) = 1$ when $z = 1$ and $d(0) = 0$ when $z = 0$, or equivalently, $d(1) > d(0)$.

The LATE estimand, in the absence of covariates, is defined by

$$\tau_{\text{LATE}} = \frac{E(y|z=1) - E(y|z=0)}{E(d=1|z=1) - E(d=1|z=0)}$$

which, under the assumptions in [Imbens and Angrist \(1994\)](#) and [Angrist, Imbens, and Rubin \(1996\)](#), is proved to be a treatment effect conditional on compliance:

$$\tau_{\text{LATE}} = \frac{E(y|z=1) - E(y|z=0)}{E(d=1|z=1) - E(d=1|z=0)} = E\{y(1) - y(0) | d(1) > d(0)\}$$

where y corresponds to the outcome of interest, $y(1)$ is the potential outcome when a unit is treated, and $y(0)$ is the potential outcome when a unit is untreated.

Without covariates, one can estimate the expression above using two-stage least squares with `ivregress`. [Imbens and Angrist \(1994\)](#) and [Angrist, Imbens, and Rubin \(1996\)](#) illustrate how the LATE estimand derived from a potential-outcomes framework, as described in [\[CAUSAL\] Intro](#), is equivalent to the instrumental-variables regression, which is usually thought of from a simultaneous-equation perspective. In the LATE framework, z satisfies the assumptions of an instrumental variable.

Recent literature (see [Śłoczyński \[2020\]](#) and [Blandhol et al. \[2022\]](#)) suggests that, when controlling for covariates, using two-stage least squares does not necessarily lead to an estimate of τ_{LATE} . [Śłoczyński, Uysal, and Wooldridge \(2022, 2025\)](#) propose weighting estimators that can be interpreted as estimates of

$$\tau_{\text{LATE}} = \frac{E_x \{E(y|z=1, \mathbf{x}) - E(y|z=0, \mathbf{x})\}}{E_x \{E(d=1|z=1, \mathbf{x}) - E(d=1|z=0, \mathbf{x})\}}$$

where $\mathbf{x}$ is a random vector of covariates and E_x denotes an expectation with respect to the covariates. The `lateffects` command provides three estimators that construct weighted estimates for the four conditional expectations above and then average over the data. The weights are constructed so that the instrument, once we condition on the covariates, is as good as if it were randomly assigned.

You can choose from the IPWRA, normalized kappa, or normalized covariate-balancing estimator. IPWRA models the outcome, the treatment, and the instrument. The outcome may be modeled using linear, logistic, probit, Poisson, fractional logistic, or fractional probit regression. The treatment and instrument may be modeled using logistic or probit regression. The normalized kappa estimator models only the instrument using logistic or probit regression to estimate a propensity score for the instrument. The normalized covariate-balancing estimator also models only the instrument using logistic or probit regression, but this estimator forces covariate balance through additional moment conditions. See [Methods and formulas](#) for a description of these estimators.

For all three estimators, we require conventional identification assumptions for treatment effects: the stable unit treatment value assumption (SUTVA), the unconfoundedness (conditional-independence) assumption, and the overlap assumption. See [\[CAUSAL\] Intro](#) for descriptions of these assumptions. Note that, for a LATE, these assumptions are defined with respect to the propensity score of the instrument rather than the propensity score of the treatment indicator. We also require that the instrument affects the outcome only through treatment, an instrumental-variable exclusion restriction. Additionally, we must guarantee that there are no defiers, understood as individuals that would act contrary to their treatment assignment.

Below, we illustrate how to use the `lateffects` command to obtain LATE estimates and how to check for violations of testable assumptions. For more discussion on LATE estimation and theory, see [Imbens and Angrist \(1994\)](#); [Angrist, Imbens, and Rubin \(1996\)](#); [Angrist and Pischke \(2009, 2015\)](#); and [Imbens and Rubin \(2015\)](#) and the references therein.

► Example 1: LATE estimation for labor outcomes

We consider the effect of schooling on wages. Schooling is an individual choice that is not independent of unobservable characteristics that affect wages, such as ability. In other words, schooling is endogenous. To address the endogeneity problem, [Card \(1995\)](#) proposes proximity to a four-year college as an instrument. He argues that proximity to a four-year college affects wages only through the effect it has on schooling decisions. For instance, he suggests that living in a college town lowers the cost of attending college, making the decision to pursue additional schooling more likely.

We use data provided by [Śłoczyński, Uysal, and Wooldridge \(2025\)](#) to revisit part of the analysis in [Card \(1995\)](#). We study the effect of having some post-secondary education (`somecol`) on the log of hourly wages (`lwage`). In this context, `somecol` is considered the treatment, while living near a four-year college (`nearc4`) is the instrumental variable acting as the treatment assignment. To build some intuition, we first use the normalized kappa estimator without covariates. We type

```
. use https://www.stata-press.com/data/r19/card95
(National Longitudinal Survey of Young Men)
```

In the `lateffects` command, we specify a first set of parentheses with a model for the outcome and a second set with a model for the treatment status. The kappa estimator computes weighted means for both models, where the weights are constructed using instrument propensity scores. In the third set of parentheses, we specify the instrument propensity-score model. Because we have no covariates, the propensity score is just the mean of `nearc4`.

```
. lateffects kappa (lwage) (somecol) (nearc4)
Iteration 0: EE criterion = 6.298e-30
Iteration 1: EE criterion = 3.286e-30

Local average treatment effect          Number of obs = 3,010
Estimator:      Normalized kappa
Outcome model:  Weighted mean
Treatment model: Weighted mean
IV pscore model: Logit
```

lwage	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
LATE somecol (Yes vs No)	1.278672	.2203624	5.80	0.000	.8467691	1.710574

In the coefficient table of our output, we see the estimate of the LATE. This is the average treatment effect of pursuing some college education on the log of wages for compliers (for those that will pursue some college education if they live near a four-year college and will not do so otherwise). Thus, for the complier subpopulation, we expect average log wages to be 1.28 higher if everyone has some college education than if no one has college education.

We would obtain the same result using `ivregress`:

```
. ivregress 2sls lwage (somecol=nearc4), vce(robust)
```

Instrumental-variables 2SLS regression	Number of obs	=	3,010
	Wald chi2(1)	=	33.67
	Prob > chi2	=	0.0000
	Root MSE	=	.69383

lwage	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
somecol	1.278672	.2203624	5.80	0.000	.8467691	1.710574
_cons	5.615699	.1117247	50.26	0.000	5.396723	5.834676

Endogenous: somecol

Exogenous: nearc4

As was demonstrated in [Imbens and Angrist \(1994\)](#) and [Angrist, Imbens, and Rubin \(1996\)](#), without covariates, two-stage least squares is LATE.

We now add covariates. In particular, we follow the basic specification in [Card \(1995\)](#). In his specification, he includes binary indicators for race (`black`), living in the south (`south`), living in a standard metropolitan area (`smsa`), and living in a metropolitan area in the initial survey wave (`smsa66`); a categorical variable for the four regions of the United States (`region`); and work experience (`exper`), which enters the model quadratically.

```
. lateffects kappa (lwage) (somecol)
> (nearc4 i.(black south smsa smsa66 region) c.exper##c.exper)
```

Iteration 0: EE criterion = 8.456e-20
Iteration 1: EE criterion = 1.836e-29

Local average treatment effect	Number of obs = 3,010
Estimator: Normalized kappa	
Outcome model: Weighted mean	
Treatment model: Weighted mean	
IV pscore model: Logit	

lwage	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
LATE						
somecol (Yes vs No)	.3328798	.2237742	1.49	0.137	-.1057095	.7714691

The LATE indicates average log wages are 0.33 higher among compliers when they all pursue some college education than when none of them do. If we had used two-stage least squares, the coefficient associated with `somecol` would be 0.69, which, given the presence of covariates, cannot be interpreted as a LATE. Also, the confidence interval for the LATE parameter in the `lateffects` results includes 0, which would not be the case if we had fit the model using `ivregress`.

► Example 2: Verifying model assumptions

For us to interpret our estimate as causal, we need to verify that the LATE assumptions hold. Using `latebalance`, we may obtain diagnostics and tests that let us ascertain if the instrument is as good as randomly assigned once we control for the covariates in our model. If this is the case, after weighting with the instrument propensity scores, group characteristics should be equivalent between those assigned to treatment and those assigned to control.

First, we use `latebalance summarize`. The standardized-difference and variance-ratio results are each presented in two columns. The first column corresponds to the raw data, and the second column presents the statistics computed using the instrument propensity-score weights. If the instrument propensity-score weights have balanced the distributions of our covariates, their weighted mean differences should be close to 0 and their weighted variance ratios should be close to 1. The table at the top additionally reports the treated and control sample sizes, which should be similar after weighting.

```
. latebalance summarize
```

Covariate balance summary

Number of observations	Raw	Weighted
Total	3,010	3,010
Assigned to treatment	2,053	1,515.395
Assigned to control	957	1,494.605

	Standardized differences		Variance ratio	
	Raw	Weighted	Raw	Weighted
black				
Yes	-.1586972	.0275062	.8277845	1.037878
south				
Yes	-.484023	-.0130109	.8972617	.995531
smsa				
Yes	.7722047	-.0029765	.5854675	1.002687
smsa66				
Yes	1.079367	-.0064253	.7272248	1.003921
region				
Midwest	.0797295	-.0076143	1.094144	.9917265
South	-.5598127	.002059	.9172754	1.000646
West	.1599008	.0599971	1.501104	1.169778
exper	-.1312633	-.0373534	.9005343	1.038114
exper#				
exper	-.134808	-.0248842	.7968106	1.008667

The table of diagnostics suggests that the group characteristics have been balanced. For a visual inspection of this balance, we could look at the full distributions of characteristics among controls and treated both in the data and after weighting. We can do this for each covariate. To illustrate with the variable `exper`, we type

```
. latebalance density exper
```

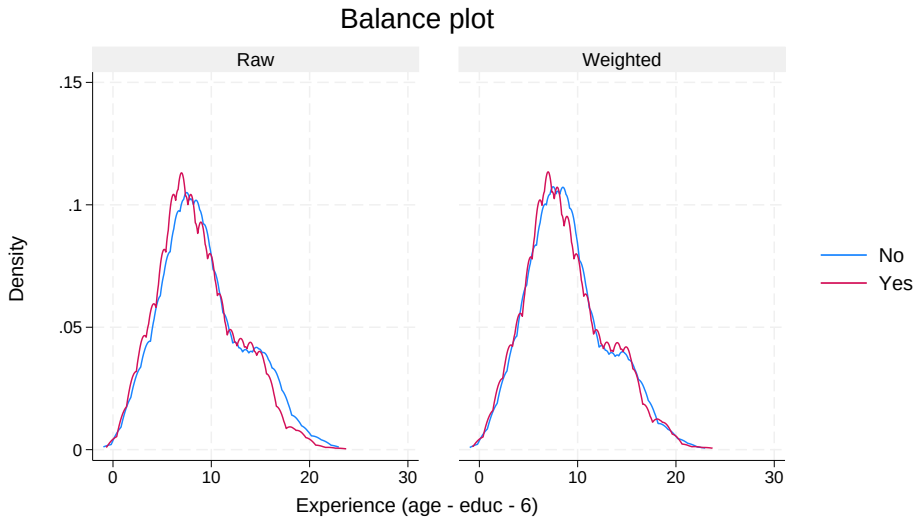


Figure 1.

The distributions of `exper` appear to be more balanced after weighting, as our diagnostics had already suggested.

We can also readily verify the overlap assumption. It states that there is a positive probability of observations being assigned to treatment and control groups once we control for covariates. The `lateoverlap` postestimation command plots estimated densities of the probability of being assigned to treatment or control, allowing us to verify the assumption. A violation of the overlap assumption would be reflected in a plot where observations are bunched toward 0, or toward 1, or toward both extremes, with few treated and control observations sharing the same regions of the support. We type

```
. lateoverlap
```

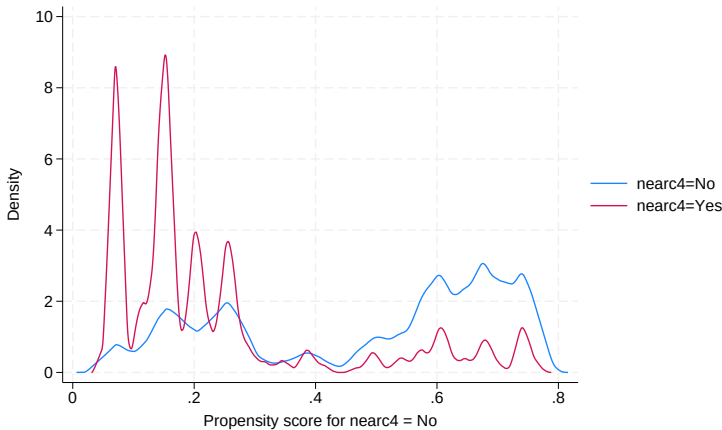


Figure 2.

The graph does not suggest that violations of the overlap assumption are a concern.



Stored results

`lateffects` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(N_clust)</code>	number of clusters
<code>e(converged)</code>	1 if converged, 0 otherwise
<code>e(late)</code>	local average treatment-effect estimate

Macros

<code>e(cmd)</code>	<code>lateffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(tmodel)</code>	logit, probit, or Weighted mean
<code>e(omodel)</code>	linear, logit, probit, poisson, flogit, fprobit, or Weighted mean
<code>e(ivpscmmodel)</code>	logit or probit
<code>e(estimator)</code>	ipwra, kappa, or balancing
<code>e(wexp)</code>	weight expression
<code>e(wtype)</code>	weight type
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>

<code>e(estat_cmd)</code>	program used to implement estat
<code>e(marginsnotok)</code>	predictions disallowed by margins
Matrices	
<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators
Functions	
<code>e(sample)</code>	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

Below, we provide methods and formulas for the weighting estimators implemented in `lateffects`. For a detailed discussion of the estimators and the underlying theory, see [Śłoczyński, Uysal, and Wooldridge \(2022, 2025\)](#).

The IPWRA, normalized kappa, and normalized covariate-balancing estimators provided by `lateffects` compute propensity-score weights for cross-sectional data, with observations $i = 1, \dots, N$. The instrument propensity scores estimate the probability of a binary instrument z conditional on covariates. Let $\mathbf{x}_z$ correspond to covariates used to model the instrument propensity scores defined by the expression

$$G(\mathbf{x}_{zi}\hat{\gamma}) = \hat{P}(z_i = 1 | \mathbf{x}_{zi})$$

$G(\mathbf{x}_{zi}\hat{\gamma})$ is estimated via a probit or logistic regression for the IPWRA and normalized kappa estimators and by using a covariate-balancing propensity score, described below, for the normalized covariate-balancing estimator. $\hat{\gamma}$ are the parameters fit using these estimators. To simplify notation, we denote $G(\mathbf{x}_{zi}\hat{\gamma}) \equiv G_{1i}$. Similarly, $1 - G(\mathbf{x}_{zi}\hat{\gamma}) \equiv G_{0i}$.

In what follows, y corresponds to the outcome variable, and d is a binary variable indicating treatment status. A covariate vector used to model the outcome is given by $\mathbf{x}_y$, and a covariate vector used to model treatment status is given by $\mathbf{x}_d$.

The estimators of the LATE defined below are functions of estimates themselves. Thus, they all use `gmm` to obtain correct standard errors.

Methods and formulas are presented under the following headings:

IPWRA estimator
Normalized kappa estimator
Normalized covariate-balancing estimator

IPWRA estimator

The IPWRA estimator provided by `lateffects ipwra` is computed via the following steps:

1. Compute $\hat{P}(d_i = 1 | \mathbf{x}_{di}, z = 0)$ using weights $1/G_{0i}$; denote the estimator $\Lambda(\mathbf{x}_{di} \hat{\delta}_0) \equiv \Lambda_{0i}$.
2. Compute $\hat{P}(d_i = 1 | \mathbf{x}_{di}, z = 1)$ using weights $1/G_{1i}$; denote the estimator $\Lambda(\mathbf{x}_{di} \hat{\delta}_1) \equiv \Lambda_{1i}$.
3. Compute $\hat{E}(y_i | \mathbf{x}_{yi}, z = 0)$ using weights $1/G_{0i}$; denote the estimator $m(\mathbf{x}_{yi} \hat{\beta}_0) \equiv m_{0i}$.
4. Compute a weighted $\hat{E}(y_i | \mathbf{x}_{yi}, z = 1)$ using weights $1/G_{1i}$; denote the estimator $m(\mathbf{x}_{yi} \hat{\beta}_1) \equiv m_{1i}$.
5. Compute the LATE via $\hat{\tau}_{\text{LATE}} = \{n^{-1} \sum_{i=1}^n (m_{1i} - m_{0i})\} / \{n^{-1} \sum_{i=1}^n (\Lambda_{1i} - \Lambda_{0i})\}$.

Normalized kappa estimator

The normalized kappa estimator provided by `lateffects kappa` is given by

$$\hat{\tau}_{\text{LATE}} = \frac{\left(\sum_{i=1}^n \frac{z_i}{G_{1i}}\right)^{-1} \sum_{i=1}^n \frac{y_i z_i}{G_{1i}} - \left(\sum_{i=1}^n \frac{1-z_i}{G_{0i}}\right)^{-1} \sum_{i=1}^n \frac{y_i (1-z_i)}{G_{0i}}}{\left(\sum_{i=1}^n \frac{z_i}{G_{1i}}\right)^{-1} \sum_{i=1}^n \frac{d_i z_i}{G_{1i}} - \left(\sum_{i=1}^n \frac{1-z_i}{G_{0i}}\right)^{-1} \sum_{i=1}^n \frac{d_i (1-z_i)}{G_{0i}}}$$

Normalized covariate-balancing estimator

The normalized covariate-balancing estimator provided by `lateffects balancing` is computed using the same formula as the normalized kappa estimator above, but to estimate the instrument propensity scores, the balancing estimator solves the sample analog of the moment condition

$$E \left[\frac{d\mathbf{x}_z}{G(\mathbf{x}_z\gamma)} - \frac{(1-d)\mathbf{x}_z}{\{1 - G(\mathbf{x}_z\gamma)\}} \right] = 0$$

Acknowledgments

We thank Derya Uysal from the Department of Economics at the University of Munich for her presentation on weighting estimators for LATE and their properties at the Stata Economics Virtual Symposium of 2023, [Abadie's kappa and weighting estimators of the local average treatment effect](#). We would also like to thank Tymon Słoczyński from the Department of Economics and International Business School (IBS) at Brandeis University, Derya Uysal, and Jeffrey M. Wooldridge for their `kappaLate` command.

References

- Angrist, J. D., G. W. Imbens, and D. B. Rubin. 1996. Identification of causal effects using instrumental variables. *Journal of the American Statistical Association* 91: 444–455. <https://doi.org/10.2307/2291629>.
- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- . 2015. *Mastering Metrics: The Path from Cause to Effect*. Princeton, NJ: Princeton University Press.

- Blandhol, C., J. Bonney, M. Mogstad, and A. Torgovitsky. 2022. When is TSLS actually LATE? NBER Working Paper 29709, National Bureau of Economic Research. <https://doi.org/10.3386/w29709>.
- Card, D. 1995. “Using geographic variation in college proximity to estimate the return to schooling”. In *Aspects of Labour Market Behaviour: Essays in Honour of John Vanderkamp*, edited by L. N. Christofides, E. K. Grant, and R. Swidinsky, 201–222. Toronto, Canada: University of Toronto Press.
- Imbens, G. W., and J. D. Angrist. 1994. Identification and estimation of local average treatment effects. *Econometrica* 62: 467–475. <https://doi.org/10.2307/2951620>.
- Imbens, G. W., and D. B. Rubin. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. New York: Cambridge University Press. <https://doi.org/10.1017/CBO9781139025751>.
- Słoczyński, T. 2020. When should we (not) interpret linear IV estimands as LATE. Working paper, Department of Economics, Brandeis University. https://people.brandeis.edu/~tslocz/Sloczynski_paper_LATE.pdf.
- Słoczyński, T., S. D. Uysal, and J. M. Wooldridge. 2022. Doubly robust estimation of local average treatment effects using inverse probability weighted regression adjustment. IZA Discussion Paper 15727, Institute for the Study of Labor (IZA). <https://docs.iza.org/dp15727.pdf>.
- . 2025. Abadie’s kappa and weighting estimators of the local average treatment effect. *Journal of Business and Economic Statistics* 43: 164–177. <https://doi.org/10.1080/07350015.2024.2332763>.

Also see

[CAUSAL] **lateffects postestimation** — Postestimation tools for lateffects⁺

[U] **20 Estimation and postestimation commands**

⁺Postestimation features after lateffects are part of [StataNow](#).

[Postestimation commands](#)
[Methods and formulas](#)

[estat](#)
[Reference](#)

[Remarks and examples](#)
[Also see](#)

[Stored results](#)

Postestimation commands

The following postestimation commands are of special interest after lateffects:

Command	Description
lateoverlap	overlap plots
latebalance	check balance of covariates
estat compliers	covariate averages for compliers

The following standard postestimation commands are also available:

Command	Description
estat summarize	summary statistics for the estimation sample
estat vce	variance–covariance matrix of the estimators (VCE)
estimates	cataloging estimation results
etable	table of estimation results
lincom	point estimates, standard errors, testing, and inference for linear combinations of parameters
nlcom	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
predictnl	point estimates, standard errors, testing, and inference for generalized predictions
test	Wald tests of simple and composite linear hypotheses
testnl	Wald tests of nonlinear hypotheses

estat

Description for estat compliers

`estat compliers` estimates the means of covariates used to estimate a local average treatment effect (LATE) alongside the mean of those same covariates for the complier subpopulation. `estat compliers` also provides an option to obtain the kappa weights used to compute the complier means, also referred to as Abadie's kappa.

Menu for estat compliers

Statistics > Postestimation

Syntax for estat compliers

```
estat compliers [lateffectsvarlist] [, genkappa(newvar) ]
```

lateffectsvarlist is a list of covariates from `lateffects` estimation.

Option for estat compliers

`genkappa(newvar)` generates a variable with name *newvar* containing the weights used to compute complier covariate averages. These weights are also known as kappa weights or Abadie's kappa.

Remarks and examples

`estat compliers` helps us characterize the average of covariates in the complier subpopulation, relative to the overall population. Given that we are estimating a treatment effect for compliers, we would like to know how compliers compare with the population. `estat compliers` additionally provides the option of generating a variable that will store Abadie's kappa. As shown by [Abadie \(2003\)](#), we can use the kappa weights to characterize features other than the mean for the compliers subpopulation. Thus, the kappa weights are an object of interest in their own right.

Stored results

`estat compliers` stores the following in `r()`:

Matrices

<code>r(kappa)</code>	matrix of covariate means and complier covariate means
-----------------------	--

Methods and formulas

`estat compliers` estimates the means of the covariates in its first column and provides the mean of those covariates for the complier subpopulation in the second column. The complier subpopulation average, for each of the k covariates, is an estimator of

$$\frac{E(\kappa_i x_{ik})}{E(\kappa_i)}$$

where the computation replaces expected values by sample averages and where κ_i is defined by

$$\kappa_i = 1 - \frac{d_i(1 - z_i)}{1 - p(z_i = 1 | \mathbf{x}_{zi})} - \frac{(1 - d_i)z_i}{p(z_i = 1 | \mathbf{x}_{zi})}$$

In the expression above, d_i refers to a realization of the observed treatment, z_i refers to a realization of the binary instrument, and $\mathbf{x}_{zi}$ is a vector of covariates used to model the instrument propensity score, $p(z_i = 1 | \mathbf{x}_{zi})$. In the expression above, $p(z_i = 1 | \mathbf{x}_{zi})$ is replaced by the logistic or probit regression estimate of the propensity score used during estimation.

Reference

Abadie, A. 2003. Semiparametric instrumental variable estimation of treatment response models. *Journal of Econometrics* 113: 231–263. [https://doi.org/10.1016/S0304-4076\(02\)00201-4](https://doi.org/10.1016/S0304-4076(02)00201-4).

Also see

[CAUSAL] **lateffects** — Local average treatment-effect estimation⁺

[CAUSAL] **latebalance** — Check balance after lateffects estimation⁺

[CAUSAL] **lateoverlap** — Overlap plots⁺

[U] **20 Estimation and postestimation commands**

⁺This command is part of [StataNow](#).

[Description](#)[Options](#)[Also see](#)[Quick start](#)[Remarks and examples](#)[Menu](#)[Stored results](#)[Syntax](#)[References](#)

Description

One of the assumptions required to use the `lateffects` estimators is the overlap assumption, which states that each individual has a positive probability of receiving each treatment-assignment level. `lateoverlap`, a postestimation command, plots the estimated densities of the probability of getting each treatment-assignment level. These plots can be used to check whether the overlap assumption is violated.

Quick start

Visually check whether the overlap assumption is violated

```
lateoverlap
```

Same as above, but use the Epanechnikov kernel function

```
lateoverlap, kernel(epanechnikov)
```

Menu

Statistics > Postestimation

Syntax

```
lateoverlap [ , treat_options kden_options ]
```

<i>treat_options</i>	Description
Main	
<code>ptlevel(<i>treat_level</i>)</code>	calculate predicted probabilities for treatment-assignment level <i>treat_level</i> ; default is the first treatment-assignment level
<code>tlevels(<i>treatments</i>)</code>	specify conditioning treatment-assignment levels; default is all treatment-assignment levels
<code>nolabel</code>	use treatment-assignment level values and not value labels in legend and axis titles
<i>kden_options</i>	Description
Main	
<code>kernel(<i>kernel</i>)</code>	specify kernel function; default is <code>kernel(triangle)</code>
<code>n(#)</code>	estimate densities using # points; default is <code>e(N)</code> , the number of observations in the estimation sample
<code>bwidth(#)</code>	half-width of kernel
<code>at(<i>var_x</i>)</code>	estimate densities using the values specified by <i>var_x</i>
Kernel plots	
<code>line#opts(<i>cline_options</i>)</code>	affect rendition of density for conditioning treatment #
Add plots	
<code>addplot(<i>plot</i>)</code>	add other plots to the generated graph
Y axis, X axis, Titles, Legend, Overall	
<i>twoway_options</i>	any options other than by() documented in [G-3] <i>twoway_options</i>
<i>kernel</i>	Description
<code>triangle</code>	triangle kernel function; the default
<code>epanechnikov</code>	Epanechnikov kernel function
<code>epan2</code>	alternative Epanechnikov kernel function
<code>biweight</code>	biweight kernel function
<code>cosine</code>	cosine trace kernel function
<code>gaussian</code>	Gaussian kernel function
<code>parzen</code>	Parzen kernel function
<code>rectangle</code>	rectangle kernel function

collect is allowed; see [U] 11.1.10 Prefix commands.

Options

Main

`ptlevel(treat_level)` specifies that predicted probabilities be calculated for treatment-assignment level *treat_level*. The default is `ptlevel(first)`, where *first* is the first treatment-assignment level.

`tlevels(treatments)` specifies the observations for which to obtain predicted probabilities. By default, all treatment-assignment levels are used. Specify *treatments* as a space-delimited list.

For instance,

```
. lateoverlap, ptlevel(1) tlevels(0 1)
```

says to predict the probability of getting treatment-assignment level 1 for those subjects who actually obtained treatment-assignment levels 0 or 1.

`no label` specifies that treatment-assignment level values and not value labels be used in legend and axis titles.

`kernel(kernel)` specifies the kernel function for use in calculating the kernel density estimates. The default kernel is the triangle kernel (`triangle`).

`n(#)` specifies the number of points at which the density estimate is to be evaluated. The default is $e(N)$, the estimation sample size.

`bw width(#)` specifies the half-width of the kernel, the width of the density window around each point. If `bw width()` is not specified, the “optimal” width is calculated and used; see [R] [kdensity](#). The optimal width is the width that would minimize the mean integrated squared error if the data were Gaussian and a Gaussian kernel were used, so it is not optimal in any global sense. In fact, for multimodal and highly skewed densities, this width is usually too wide and oversmooths the density ([Silverman 1986](#)).

`at(var_x)` specifies a variable that contains the values at which the density should be estimated. This option allows you to more easily obtain density estimates for different variables or different subsamples of a variable and then overlay the estimated densities for comparison.

Kernel plots

`line#opts(cline_options)` affect the rendition of the plotted kernel density estimates. See [G-3] [cline_options](#).

Add plots

`addplot(plot)` provides a way to add other plots to the generated graph. See [G-3] [addplot_option](#).

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] [twoway_options](#), excluding `by()`. These include options for titling the graph (see [G-3] [title_options](#)) and for saving the graph to disk (see [G-3] [saving_option](#)).

Remarks and examples

`lateoverlap` plots the estimated densities of the probability of getting each treatment-assignment level after `lateeffects`.

These plots can be used to check whether the overlap assumption is violated. The overlap assumption is satisfied when there is a chance of seeing observations in both the control and the treatment-assignment groups at each combination of covariate values.

The overlap assumption is required by the estimators implemented in `lateffects`. Intuitively, when the overlap assumption is violated, we cannot predict, or otherwise account for, the unobserved outcomes for some units, specifically, for units assigned to treatment with characteristics (that is, covariate values) that are absent in the control group and for units assigned to control with characteristics that cannot be found in the treated group.

There is evidence that the overlap assumption is violated when an estimated density has too much mass around 0 or 1; see [Busso, DiNardo, and McCrary \(2014\)](#). An implication of this point is that when the overlap assumption is violated, the estimated densities will have relatively little mass in the regions in which they overlap.

For an illustration, see [example 2](#) of [\[CAUSAL\] lateffects](#).

Stored results

`lateoverlap` stores the following in `r()`:

Scalars

<code>r(bwidthj)</code>	kernel bandwidth for treatment-assignment level j
<code>r(nj)</code>	number of points at which the estimate was evaluated for treatment-assignment level j
<code>r(scalej)</code>	density bin width for treatment-assignment level j

Macros

<code>r(kernel)</code>	name of kernel
------------------------	----------------

References

- Busso, M., J. DiNardo, and J. McCrary. 2014. New evidence on the finite sample properties of propensity score reweighting and matching estimators. *Review of Economics and Statistics* 96: 885–897. https://doi.org/10.1162/REST_a_00431.
- Silverman, B. W. 1986. *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.

Also see

[\[CAUSAL\] lateffects](#) — Local average treatment-effect estimation⁺

Description

Causal inference aims to estimate the causal effect of a treatment on an outcome. Causal mediation analysis further explores the causal effect by evaluating how that effect may arise. In particular, the total effect can be decomposed into a direct effect and one or more indirect effects. An indirect effect is an effect in which the treatment leads to a change in an intervening variable, a mediator, and that change then leads to a change in an outcome. Causal mediation analysis explores whether and to what extent the effect of a treatment on an outcome is due to a change in the mediator.

In this entry, we provide a conceptual introduction to causal mediation analysis and the corresponding research process. We also introduce the `mediate` command for fitting causal mediation models.

For syntax and further details on fitting causal mediation models with one mediator, see [\[CAUSAL\] mediate](#). For syntax and further details on fitting causal mediation models with two mediators, see [\[CAUSAL\] mediate multiple](#).

Remarks and examples

Remarks are presented under the following headings:

[Introduction](#)

[Approaches to mediation analysis](#)

[Workflow for causal mediation](#)

[Potential outcomes, effects, and decompositions](#)

[One mediator](#)

[Two parallel mediators](#)

[Two sequential mediators](#)

Introduction

Causal inference is an essential goal in many research areas and aims at identifying and quantifying causal effects. For example, we might wish to find out whether physical exercise leads to an improvement in self-perceived well-being, and if so, to what extent. Causality in this context typically means that there is some cause T that has an effect on some outcome Y . We could visualize this relation with a simple causal diagram:



Figure 1

If T is a measure of exercise and Y is well-being, then under certain assumptions, we could use the above causal model to identify the total effect of exercise on well-being (by means of a randomized controlled trial, for instance). However, a question that we cannot answer empirically with our simple causal model is why exercise may increase well-being. Perhaps exercising causes an increase in certain chemicals or hormones in the human body, which in turn affects perceptions of well-being. To assess such intermediary effects, we need to expand our simple causal model by adding variables that lie on the causal pathway between T and Y :



Figure 2

Suppose that, in our exercise example, the variable M represents the production of a certain chemical in the human body. With this new model, we now hypothesize that exercising leads to the production of this chemical, which in turn leads to an increase in well-being. However, it might be unrealistic to assume that the effect of exercise on well-being hinges exclusively on the production of that chemical. Perhaps we would like to allow for the possibility that exercise has an effect on well-being beyond its path through the mediating variable, and so a better model might be

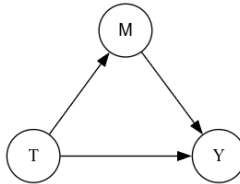


Figure 3

Here we include a direct path from T to Y in addition to the indirect path of T to Y via M . In other words, we assume that exercise produces a particular chemical that affects well-being, but we also allow for the possibility of a direct effect of exercise on well-being that is not related to the chemical. This is the classical mediation model that decomposes the total effect into a direct and an indirect effect. Causal mediation analysis aims to identify these direct and indirect effects and give them a causal interpretation.

Causal mediation analysis can extend beyond this basic model. We could have multiple mediators of interest. For instance, we could also hypothesize that increasing exercise leads to increased strength, which in turn leads to an increase in well-being. Again, we assume that exercise may affect well-being in ways other than changes in the chemical and in strength. Our model now can be represented by the following diagram in which M_1 represents the chemical and M_2 represents strength.

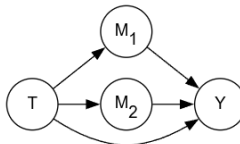


Figure 4

We can now decompose the total effect into a direct effect and two indirect effects.

In figure 4, the mediators are known as parallel mediators. Neither mediator is assumed to predict the other. Perhaps we also hypothesize that the change in the chemical can lead to a change in strength. In this case, the mediators are known as sequential mediators. In our diagram, we now add an arrow from M_1 to M_2 .

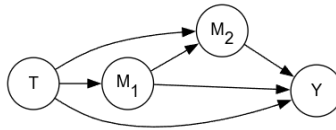


Figure 5

Here there are four path-specific effects of interest: the direct effect from T to Y (that is, the effect of T on Y not mediated by either M_1 or M_2); one effect through M_1 alone; one through M_2 alone; and one through both M_1 and M_2 . These four effects combined will sum to the total causal effect.

In each of these situations, we can use causal mediation analysis to estimate the direct and indirect effects of interest to better understand the effect of T on Y .

Approaches to mediation analysis

Mediation analysis can be performed in a variety of ways. For the one mediator case, the classical approach of [Baron and Kenny \(1986\)](#) fits two linear regression models, one for M and one for Y , and estimates direct, indirect, and total effects as functions of the coefficients. Estimation can be simplified by fitting the models for M and Y simultaneously via structural equation modeling as discussed in [\[SEM\] Example 42g](#). In Stata, you can use `sem` to fit linear models for the outcome and mediator, and you can then use `estat teffects` to obtain a decomposition of direct and indirect effects based on the results from `sem`. Similarly, models with two parallel or sequential mediators can be fit with `sem`, and `estat teffects` and `nlcom` can be used to estimate direct and indirect effects. Note that this classical approach relies on the specification of a particular model at the outset of the process.

Another approach to mediation analysis is based on the potential-outcomes framework. The potential outcomes are values of the outcome that would be obtained under different conditions, such as when the treatment occurs. Differences in potential outcomes yield direct, indirect, and total effects of interest. This is the approach typically referred to as causal mediation analysis and is the one implemented in `mediate`.

The causal mediation framework allows much flexibility. In this framework, it is common to allow the mediator and the treatment to interact; thus, we do not assume that the effect of a mediator on the outcome is the same for the treated and untreated groups. The total effect of the treatment on the outcome can be decomposed into direct and indirect effects in multiple ways, and the researcher can study the decomposition or decompositions that answer the research questions of interest. The effects are defined in a model-free manner, so the researcher can select an estimation method that is appropriate for his or her data and then compute estimates of the effects of interest.

When the outcome and the mediators are modeled using linear regression and there is no treatment–mediator interaction (and no mediator–mediator interaction in the two-mediator case), the classical approach and causal mediation via the potential-outcomes framework will lead to the same results.

Workflow for causal mediation

The general workflow for researchers performing causal mediation analysis is as follows:

1. Specify your research question.
2. Identify the treatment, mediators, and outcome to be analyzed.
3. Determine which effect decomposition or decompositions can be used to answer your research question.
4. Evaluate whether assumptions for causal interpretation are appropriate.
5. Select a method for estimating the causal effects of interest.
6. Interpret the results.

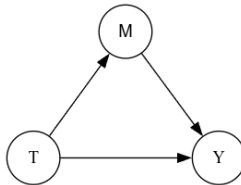
In our introductory discussion, we provided simple examples of step 2 by using exercise, chemical production, strength, and well-being. Below, we will briefly provide a conceptual introduction to the remaining steps.

Potential outcomes, effects, and decompositions

Causal mediation analysis allows us to estimate a variety of effects that are defined in terms of potential-outcome means. Here we discuss the potential-outcome means that can be estimated in both the one-mediator and two-mediator cases, and we show how different effects can be estimated from these to answer a variety of research questions. To explore these concepts, we will use a simple example and demonstrate how to use the `mediate` command.

One mediator

With one mediator, we are interested in decomposing the total effect of a treatment on the outcome into the indirect effect through a mediator and the direct effect.



To explore causal mediation for this case, we use a treatment variable t , a mediator variable m , and an outcome variable y .

Causal mediation relies on estimating the total, indirect, and direct effects of interest from potential-outcome means. Therefore, we will first consider the four potential-outcome means. To obtain these with `mediate`, we specify our outcome, mediator, and treatment in parentheses followed by the `pomeans` option.

```
. mediate (y) (m) (t), pomeans
Iteration 0: EE criterion = 2.804e-27
Iteration 1: EE criterion = 1.878e-28

Causal mediation analysis                                Number of obs = 2,000
Outcome model:      Linear
Mediator model:      Linear
Mediator variable: m
Treatment type:      Binary
```

y	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
PMeans						
Y0M0	31.04239	.2603784	119.22	0.000	30.53205	31.55272
Y1M0	29.7409	.242952	122.41	0.000	29.26473	30.21708
Y0M1	26.95702	.1644953	163.88	0.000	26.63462	27.27943
Y1M1	26.2634	.1281546	204.94	0.000	26.01222	26.51458

Note: Outcome equation includes treatment-mediator interaction.

Here the first potential-outcome mean, labeled Y0M0, is the population-average value of the outcome that would be expected if everyone was untreated. In this notation, Y0 implies it is the value of y when the treatment is set to 0, and M0 implies the mediator is set to its value that would occur when the treatment is 0. Similarly, the last potential-outcome mean, labeled Y1M1, is the population-average value of the outcome that would be expected if everyone was given the treatment.

The potential-outcome means labeled Y1M0 and Y0M1 are known as cross-world potential-outcome means. Y1M0 is the expected value of the outcome when everyone is treated but counterfactually experiences the value of the mediator associated with being untreated. Y0M1 is the expected value of the outcome when everyone is untreated but counterfactually experiences the value of the mediator associated with being treated.

From these four potential-outcome means, we can estimate the total, indirect, and direct effects. These effects are reported by `mediate` by default.

```
. mediate (y) (m) (t)
Iteration 0: EE criterion = 2.804e-27
Iteration 1: EE criterion = 1.262e-28

Causal mediation analysis                                Number of obs = 2,000
Outcome model:      Linear
Mediator model:      Linear
Mediator variable: m
Treatment type:      Binary
```

	y	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NIE	t						
(Yes vs No)		-3.477502	.2286525	-15.21	0.000	-3.925653	-3.029352
NDE	t						
(Yes vs No)		-1.301482	.1701042	-7.65	0.000	-1.63488	-.9680837
TE	t						
(Yes vs No)		-4.778984	.287326	-16.63	0.000	-5.342133	-4.215836

Note: Outcome equation includes treatment-mediator interaction.

The total effect, labeled TE, is the difference in potential-outcome means when everyone is treated versus when everyone is untreated ($Y1M1 - Y0M0$). It has the same interpretation as an average treatment effect that is commonly reported with other causal inference methods. We expect the average of y to be 4.78 less in the population when everyone is treated versus when no one is treated.

What is unique to causal mediation is the ability to better understand this effect in terms of the mediator. The value labeled NIE is the estimated natural indirect effect, sometimes called the total natural indirect effect, which is the portion of the total effect that can be attributed to a change in t leading to a change in m , which in turn leads to a change in y . The value labeled NDE is the estimated natural direct effect, sometimes called the pure natural direct effect, which is the portion of the total effect not attributed to mediation through m . The total effect is the sum of the natural indirect effect and the natural direct effect, and we can see that in this case, the indirect effect is larger (in absolute value) than the direct effect, accounting for over half of the total effect.

Causal mediation allows for the effect of the mediator to differ for treated and untreated groups by allowing for interaction between the treatment and mediator in the model for the outcome, as noted at the bottom of this output. This leads to two possible ways to decompose the total effect into direct and indirect effects. The decomposition above includes the interaction effect in the indirect effect and isolates the direct effect. [Nguyen, Schmid, and Stuart \(2021\)](#) recommend using this decomposition when the researcher assumes a direct effect exists and is questioning whether any mediation effect via m exists.

The second decomposition can be obtained as follows:

```
. mediate (y) (m) (t), pnietnde te
Iteration 0: EE criterion = 2.804e-27
Iteration 1: EE criterion = 1.036e-28

Causal mediation analysis                                Number of obs = 2,000
Outcome model:      Linear
Mediator model:      Linear
Mediator variable: m
Treatment type:      Binary
```

		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
PNIE	t						
(Yes vs No)		-4.085363	.2791863	-14.63	0.000	-4.632558	-3.538168
TNDE	t						
(Yes vs No)		-.6936208	.1656556	-4.19	0.000	-1.0183	-.3689417
TE	t						
(Yes vs No)		-4.778984	.287326	-16.63	0.000	-5.342133	-4.215836

Note: Outcome equation includes treatment-mediator interaction.

Here the value labeled PNIE is the estimated pure natural indirect effect, and the value labeled TNDE is the estimated total natural direct effect. These two sum to the same total effect that we obtained previously. This decomposition includes the interaction effect in the direct effect and isolates the indirect effect. [Nguyen, Schmid, and Stuart \(2021\)](#) recommend using this decomposition when the researcher assumes that there is a mediating effect via *m* and is questioning whether any additional effects exist.

When the research question does not make a prior assumption about a direct or indirect effect, [Nguyen, Schmid, and Stuart \(2021\)](#) note that both of these decompositions can be reported to characterize the mediating effects.

Researchers may also want to investigate what the direct effect would be if the mediator is set to a specific value of interest. The controlled direct effect provides this information and can be estimated by using `estat cde` after `mediate`.

```
. estat cde, mvalue(5)
Controlled direct effect                                Number of obs = 2,000
Mediator variable: m
Mediator value = 5
```

		CDE	Delta-method std. err.	z	P> z	[95% conf. interval]	
t							
(Yes vs No)		-2.876864	.4969117	-5.79	0.000	-3.850793	-1.902935

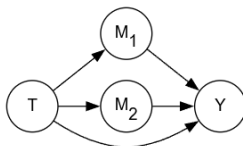
Here we estimate the effect of *t* on *y* when *m* is set to 5.

Causal inference based on estimates like those shown above requires a number of assumptions. In particular, we assume that there is no unobserved confounding in the treatment–outcome relationship, in the mediator–outcome relationship, or in the treatment–mediator relationship. In addition, we assume there are no confounders in the mediator–outcome relationship that are caused by the treatment.

In this section, we provided a preview of the types of research questions, the related effects that can be estimated, the assumptions, and the syntax of the `mediate` command that can be used to fit models to perform causal mediation analysis with one mediator. `mediate` also allows for specifying covariates in the outcome and mediator models and for modeling mediators and outcomes that are continuous, binary, or count. For more conceptual and technical details, command syntax, and worked examples using `mediate` with one mediator, see [\[CAUSAL\] mediate](#) and the references provided there.

Two parallel mediators

When we evaluate two mediators in causal mediation analysis, we can decompose the total effects into a direct effect and indirect effects via each mediator. When two mediators are parallel, there are three pathways of interest from the treatment to the outcome.



We will extend our example above to include two mediators, `m1` and `m2`, as we explore the potential-outcomes framework, effects of interest, and the `mediate` command with two parallel mediators.

We first consider the potential-outcome means when mediators are parallel and when we allow for interactions between the treatment and each of the mediators as well as an interaction between the two mediators. This is the most flexible version of the causal mediation model with parallel mediators. We specify the outcome, mediators, and treatment and request potential-outcome means in the `mediate` command similarly to the specification for the one-mediator case. We also add the `tinteraction` and `minteraction` options to include the interactions.

```
. mediate (y) (m2) (m1) (t), tinteraction minteraction pomeans
Iteration 0: EE criterion = 2.526e-26
Iteration 1: EE criterion = 8.897e-29
Causal mediation analysis                               Number of obs = 2,000
Mediation type: Parallel
Mediator 1:      m1
Mediator 2:      m2
Treatment type: Binary
```

y		Robust		z	P> z	[95% conf. interval]	
		Coefficient	std. err.				
P0means							
	000	31.64364	.3251299	97.33	0.000	31.0064	32.28088
	001	31.4691	.4037414	77.94	0.000	30.67778	32.26041
	010	28.3664	.1809759	156.74	0.000	28.01169	28.7211
	011	28.07937	.16393	171.29	0.000	27.75807	28.40066
	100	29.89799	.3377696	88.52	0.000	29.23597	30.56
	101	29.66508	.372723	79.59	0.000	28.93455	30.3956
	110	26.91666	.2504554	107.47	0.000	26.42578	27.40755
	111	26.44074	.1378333	191.83	0.000	26.1706	26.71089

Note: Outcome equation includes treatment-mediator interactions and mediator-mediator interaction.

While we had only four potential-outcome means with one mediator, we now have eight. The first one, labeled 000, is the population-average value of y that would be expected if everyone was untreated. The last one, labeled 111, is the population-average value of y that would be expected if everyone was treated. In between, we have cross-world potential outcomes that are similar to those in the one mediator case. The first number corresponds to y , the second to m_1 , and the third to m_2 , where 0 corresponds to untreated and 1 corresponds to treated in each of these positions. So the potential-outcome mean labeled 001 is the expected value of y when everyone is untreated and experiences the value of m_1 associated with being untreated and the value of m_2 associated with being treated. The other cross-world potential-outcome means can be interpreted similarly.

From these potential-outcome means, we can estimate the total, indirect, and direct effects.

```
. mediate (y) (m2) (m1) (t), tinteraction minteraction
Iteration 0: EE criterion = 2.526e-26
Iteration 1: EE criterion = 1.137e-28

Causal mediation analysis                                Number of obs = 2,000
Mediation type: Parallel
Mediator 1:      m1
Mediator 2:      m2
Treatment type: Binary
```

	y	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE	00	-1.745654	.1767601	-9.88	0.000	-2.092097	-1.39921
	10	-1.449735	.2693148	-5.38	0.000	-1.977582	-.9218876
	01	-1.804018	.2952766	-6.11	0.000	-2.38275	-1.225287
	11	-1.638622	.139265	-11.77	0.000	-1.911576	-1.365667
NIE1	00	-3.277243	.3619214	-9.06	0.000	-3.986596	-2.56789
	10	-2.981325	.3508021	-8.50	0.000	-3.668884	-2.293765
	01	-3.389729	.3789566	-8.94	0.000	-4.132471	-2.646988
	11	-3.224333	.4005567	-8.05	0.000	-4.00941	-2.439256
NIE2	00	-.174545	.1703176	-1.02	0.305	-.5083614	.1592713
	10	-.2329098	.2139202	-1.09	0.276	-.6521856	.186366
	01	-.2870311	.1642122	-1.75	0.080	-.6088811	.0348188
	11	-.4759182	.2112993	-2.25	0.024	-.8900573	-.0617791
TE	t (Yes vs No)	-5.202896	.3531395	-14.73	0.000	-5.895037	-4.510756

Note: Outcome equation includes treatment-mediator interactions and mediator-mediator interaction.

The total effect, TE, is interpreted just like the total effect in the one-mediator case. However, we now can estimate four natural direct effects and four natural indirect effects through each mediator. The natural direct effects (NDE) here range from -1.45 to -1.80 . The first of these, labeled 00, is the estimated direct effect when both mediators are at their values associated with being untreated. The NIE1 section reports natural indirect effects through the m1 mediator, which range from -2.98 to -3.39 . The 00 indirect effect is the effect via m1 if everyone was untreated and if m2 is set to its value associated with being untreated. Other indirect effects are interpreted similarly. The NIE2 section reports natural indirect effects through the m2 mediator, which range from -0.17 to -0.48 . From these results, there are six possible ways to decompose the total effect into a direct effect, an indirect effect through m1, and an indirect effect through m2. While we do not show all of these decompositions in this introduction, reviewing all the reported effects, we see that the indirect effect through m1 is larger than both the direct effect and the indirect effect through m2.

In addition to the natural direct and indirect effects, we can also estimate controlled direct effects, as we did in the case of one mediator. This estimate is useful when the research question asks what the direct effect would be when the mediators are set to specific values.

```
. estat cde, mvalue(m1=3 m2=5)
Controlled direct effect
Mediator variables: m1 m2
Mediator values:
  m1 = 3
  m2 = 5
```

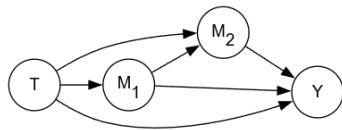
Number of obs = 2,000

	CDE	Delta-method std. err.	z	P> z	[95% conf. interval]	
t (Yes vs No)	-1.485701	.7919253	-1.88	0.061	-3.037846	.0664441

Here we estimate the direct effect of t on y when setting m1 to 3 and m2 to 5.

Two sequential mediators

When two mediators are sequential, there are now four pathways that may be of interest from the treatment to the outcome.



Continuing with our example above, we have now introduced a path from m1 to m2, which means we can estimate an indirect effect that goes through both m1 and m2 in addition to the indirect effect through m1 alone and the indirect effect through m2 alone.

With this additional causal pathway, there are more potential-outcome means that can be estimated. The most flexible version of a model for sequential mediators allows for interactions between the treatment and mediators and between the mediators in the outcome equation and allows for an interaction between the treatment and the first mediator in the equation for the second mediator. Here we estimate all the potential-outcome means in this most flexible case. In the `mediate` command, we include the `sequential` option to specify that the mediators are sequential and the `meqtinteraction` option to include the interaction in the mediator equation.

```
. mediate (y) (m2) (m1) (t), sequential tinteraction minteraction
> meqtinteraction pomeans

Iteration 0: EE criterion = 6.373e-21
Iteration 1: EE criterion = 2.656e-28

Causal mediation analysis                                Number of obs = 2,000
Mediation type: Sequential
Mediator 1:      m1
Mediator 2:      m2
Treatment type: Binary
```

	y	Robust		z	P> z	[95% conf. interval]	
		Coefficient	std. err.				
POmeans							
	0000	31.30634	.2935023	106.66	0.000	30.73109	31.8816
	0001	31.20478	.3270818	95.40	0.000	30.56371	31.84584
	0010	31.2244	.3084594	101.23	0.000	30.61984	31.82897
	0011	31.11898	.3576589	87.01	0.000	30.41798	31.81997
	0100	28.0291	.1958977	143.08	0.000	27.64515	28.41305
	0101	27.86208	.1574547	176.95	0.000	27.55347	28.17068
	0110	27.90262	.1665546	167.53	0.000	27.57618	28.22906
	0111	27.72925	.1619862	171.18	0.000	27.41176	28.04673
	1000	29.16931	.2600319	112.18	0.000	28.65966	29.67896
	1001	29.03378	.2511767	115.59	0.000	28.54148	29.52608
	1010	29.04938	.2429515	119.57	0.000	28.5732	29.52556
	1011	28.9087	.2744599	105.33	0.000	28.37077	29.44663
	1100	26.18799	.2941772	89.02	0.000	25.61141	26.76456
	1101	25.91105	.2332539	111.09	0.000	25.45388	26.36822
	1110	25.97183	.2590678	100.25	0.000	25.46407	26.47959
	1111	25.68436	.2425794	105.88	0.000	25.20892	26.15981

Note: Outcome equation includes treatment-mediator interactions and mediator-mediator interaction, and mediator 2 equation includes treatment-mediator interaction.

This produces 16 potential-outcome means. Again the first one, labeled 0000, is the population-average value of y that would be expected if everyone was untreated. The last one, labeled 1111, is the population-average value of y that would be expected if everyone was treated. The difference in these two leads to the total effect. The remaining potential-outcome means correspond to various cross-world potential outcomes. From these 16 potential-outcome means, we can estimate 8 of each type of direct and indirect effects.

```
. mediate (y) (m2) (m1) (t), sequential tinteraction minteraction
> meqtinteraction
```

Iteration 0: EE criterion = 6.373e-21

Iteration 1: EE criterion = 5.935e-28

Causal mediation analysis

Number of obs = 2,000

Mediation type: Sequential

Mediator 1: m1

Mediator 2: m2

Treatment type: Binary

		Robust		z	P> z	[95% conf. interval]	
y	Coefficient	std. err.					
NDE							
000	-2.137034	.1755708	-12.17	0.000	-2.481146	-1.792921	
100	-1.841115	.2914441	-6.32	0.000	-2.412335	-1.269895	
010	-2.175025	.1867005	-11.65	0.000	-2.540951	-1.809098	
001	-2.170996	.1970605	-11.02	0.000	-2.557228	-1.784765	
110	-1.93079	.2240868	-8.62	0.000	-2.369992	-1.491587	
101	-1.951029	.1944125	-10.04	0.000	-2.33207	-1.569987	
011	-2.210278	.284549	-7.77	0.000	-2.767984	-1.652573	
111	-2.044882	.1971614	-10.37	0.000	-2.431311	-1.658453	
NIE1							
000	-3.277243	.3619214	-9.06	0.000	-3.986596	-2.56789	
100	-2.981325	.3508021	-8.50	0.000	-3.668884	-2.293765	
010	-3.321785	.3659441	-9.08	0.000	-4.039022	-2.604548	
001	-3.342699	.3726794	-8.97	0.000	-4.073137	-2.612261	
110	-3.07755	.3644594	-8.44	0.000	-3.791877	-2.363223	
101	-3.122732	.3817002	-8.18	0.000	-3.87085	-2.374613	
011	-3.389729	.3789566	-8.94	0.000	-4.132471	-2.646988	
111	-3.224333	.4005567	-8.05	0.000	-4.00941	-2.439256	
NIE2							
000	-.0819393	.0646655	-1.27	0.205	-.2086815	.0448028	
100	-.1199303	.0962299	-1.25	0.213	-.3085375	.0686768	
010	-.126481	.0623618	-2.03	0.043	-.2487078	-.0042542	
001	-.0858009	.068813	-1.25	0.212	-.2206718	.04907	
110	-.2161557	.0956439	-2.26	0.024	-.4036142	-.0286971	
101	-.1250831	.104926	-1.19	0.233	-.3307343	.0805681	
011	-.1328312	.0670555	-1.98	0.048	-.2642576	-.0014048	
111	-.2266847	.1099928	-2.06	0.039	-.4422665	-.0111028	
NIE12							
000	-.101568	.100049	-1.02	0.310	-.2976605	.0945245	
100	-.1355305	.1244552	-1.09	0.276	-.3794582	.1083972	
010	-.1670238	.0977036	-1.71	0.087	-.3585193	.0244717	
001	-.1054296	.1019476	-1.03	0.301	-.3052432	.0943841	
110	-.2769374	.1246697	-2.22	0.026	-.5212856	-.0325893	
101	-.1406833	.1288855	-1.09	0.275	-.3932943	.1119277	
011	-.1733739	.0984035	-1.76	0.078	-.3662413	.0194934	
111	-.2874664	.1298441	-2.21	0.027	-.5419561	-.0329768	
TE							
t							
(Yes vs No)	-5.62198	.452873	-12.41	0.000	-6.509595	-4.734365	

Note: Outcome equation includes treatment-mediator interactions and mediator-mediator interaction, and mediator 2 equation includes treatment-mediator interaction.

The total effect is estimated to be -5.62 . The natural direct effects are reported in the NDE section and range from -1.84 to -2.21 . The natural indirect effects through m1 only are reported in the NIE1 section and range from -2.98 to -3.39 . The natural indirect effects through m2 only are reported in the NIE2 section and range from -0.08 to -0.23 . The natural indirect effects through both m1 and m2 are reported in the NIE12 section and range from -0.10 to -0.29 . There are 24 possible ways to decompose the total effect into one direct effect, one indirect effect through m1, one indirect effect through m2, and one indirect effect through m1 and m2. We do not explore each of these decompositions in this introduction. However, a review of these effects indicates that the largest proportion of the total effect can be attributed to the indirect effect through m1 alone.

Sometimes, the research question allows us to focus on a more coarse decomposition. For instance, if we wanted to focus on the effect through m1 alone and on any effect that involves m2, we could estimate mediator-specific natural effects which isolate the effect through m1 and combine the effects through m2 alone with those through both m1 and m2. This decomposition of effects is known as type-1 mediator-specific effects, and we specify the `mseffects(m1)` option in `mediate` to estimate them.

```
. mediate (y) (m2) (m1) (t), sequential tinteraction minteraction
> meqtinteraction mseffects(m1)

Iteration 0: EE criterion = 6.373e-21
Iteration 1: EE criterion = 2.787e-28

Causal mediation analysis                                Number of obs = 2,000
Mediation type: Sequential
Mediator 1:      m1
Mediator 2:      m2
Treatment type: Binary
MS effects:      Type 1
```

y		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
MS-NDE							
	00	-2.137034	.1755708	-12.17	0.000	-2.481146	-1.792921
	01	-2.210278	.284549	-7.77	0.000	-2.767984	-1.652573
	10	-1.841115	.2914441	-6.32	0.000	-2.412335	-1.269895
	11	-2.044882	.1971614	-10.37	0.000	-2.431311	-1.658453
MS-NIE1							
	00	-3.277243	.3619214	-9.06	0.000	-3.986596	-2.56789
	01	-3.389729	.3789566	-8.94	0.000	-4.132471	-2.646988
	10	-2.981325	.3508021	-8.50	0.000	-3.668884	-2.293765
	11	-3.224333	.4005567	-8.05	0.000	-4.00941	-2.439256
MS-NIE2							
	00	-.1873689	.1647989	-1.14	0.256	-.5103689	.1356311
	01	-.299855	.1591053	-1.88	0.059	-.6116956	.0119857
	10	-.2606136	.218082	-1.20	0.232	-.6880465	.1668193
	11	-.5036221	.2198351	-2.29	0.022	-.934491	-.0727532
TE							
	t						
(Yes vs No)		-5.62198	.452873	-12.41	0.000	-6.509595	-4.734365

Note: Outcome equation includes treatment-mediator interactions and mediator-mediator interaction, and mediator 2 equation includes treatment-mediator interaction.

We now obtain only four estimates each of mediator-specific direct effects, indirect effects through `m1` alone, and indirect effects involving `m2`. The estimates in the section labeled `MS-NIE1` are the mediator-specific natural indirect effects through `m1`. The estimates in the section labeled `MS-NIE2` are the mediator-specific natural indirect effects through both pathways involving `m2`.

It is similarly possible to estimate mediator-specific effects that isolate the pathway through `m2` alone, which are known as type-2 mediator-specific effects, by specifying the `mseffects(m2)` option.

As in the one-mediator case, causal inference based on estimates like those shown here requires making assumptions of no unobserved confounding. The assumptions are natural extensions of those required with one mediator.

Above, we have previewed of the effects that can be estimated in the two-mediator case and shown how those effects grow quickly in number as we increase the number of mediators. This complexity is reduced as we allow for less flexibility by omitting interactions in our models or when the mediators are parallel rather than sequential. For more conceptual and technical details, command syntax, and worked examples using `mediate` with two mediators, see [\[CAUSAL\] mediate multiple](#) and the references provided there.

References

- Baron, R. M., and D. A. Kenny. 1986. The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of Personality and Social Psychology* 51: 1173–1182. <https://doi.org/10.1037/0022-3514.51.6.1173>.
- Nguyen, T. Q., I. Schmid, and E. A. Stuart. 2021. Clarifying causal mediation analysis for the applied researcher: Defining effects based on what we want to learn. *Psychological Methods* 26: 255–271. <https://doi.org/10.1037/met0000299>.

Also see

[\[CAUSAL\] mediate](#) — Causal mediation analysis (one mediator)

[\[CAUSAL\] mediate multiple](#) — Causal mediation analysis (two mediators)⁺

[Description](#)

[Options](#)

[Acknowledgments](#)

[Quick start](#)

[Remarks and examples](#)

[References](#)

[Menu](#)

[Stored results](#)

[Also see](#)

[Syntax](#)

[Methods and formulas](#)

Description

`mediate` fits causal mediation models and estimates effects of a treatment on an outcome. The treatment effect can occur both directly and indirectly through another variable, a mediator.

In this entry, we describe how `mediate` can be used to perform causal mediation analysis with one mediator variable. The outcome and mediator variables may be continuous, binary, or count. The treatment may be binary, multivalued, or continuous.

The estimated direct, indirect, and total effects have a causal interpretation provided that assumptions pertaining to causal mediation models are met.

For an introduction to causal mediation analysis, see [\[CAUSAL\] mediate intro](#). For the use of `mediate` to fit causal mediation models with two mediators, see [\[CAUSAL\] mediate multiple](#).

Quick start

Fit the mediation model with continuous outcome `y1`, continuous mediator `m1`, and categorical treatment `t1`, and estimate the total effect, natural direct effect, and natural indirect effect

```
mediate (y1) (m1) (t1)
```

Same as above, but with covariates in both the outcome and the mediator equations

```
mediate (y1 x1 x2) (m1 x1 x3) (t1)
```

Same as above, but with probit model for binary outcome `y2` and Poisson model for count mediator `m2`

```
mediate (y2 x1 x2, probit) (m2 x1 x3, poisson) (t1)
```

Same as above, but estimate only the natural indirect effect (NIE)

```
mediate (y2 x1 x2, probit) (m2 x1 x3, poisson) (t1), nie
```

Same as above, but also estimate potential-outcome means

```
mediate (y2 x1 x2, probit) (m2 x1 x3, poisson) (t1), nie pomeans
```

Fit the mediation model with continuous treatment `t2`, and evaluate at values 0 and 4 of the treatment with 0 as the control

```
mediate (y2 x1 x2, probit) (m2 x1 x3, poisson) (t2, continuous(0 4))
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Causal mediation

Statistics > Causal inference/treatment effects > Binary outcomes > Causal mediation

Statistics > Causal inference/treatment effects > Count outcomes > Causal mediation

Statistics > Causal inference/treatment effects > Nonnegative outcomes > Causal mediation

Syntax

```
mediate (ovar [ omvarlist , omodel noconstant ] )  
        (mvar [ mmvarlist , mmodel noconstant ] )  
        (tvar [ , continuous(numlist) ] ) [ if ] [ in ] [ weight ] [ , stat options ]
```

ovar is a continuous, binary, or count outcome of interest.
omvarlist specifies the covariates in the outcome model.
mvar is the mediator variable and may be continuous, binary, or count.
mmvarlist specifies the covariates in the mediator model.
tvar is the treatment variable and may be binary, multivalued, or continuous.

<i>omodel</i>	Description
Model	
linear	linear model; the default
<u>expmean</u>	exponential-mean model
logit	logistic regression model
probit	probit regression model
poisson	Poisson model

omodel specifies the model for the outcome variable.

<i>mmodel</i>	Description
Model	
linear	linear model; the default
<u>expmean</u>	exponential-mean model
logit	logistic regression model
probit	probit regression model
poisson	Poisson model

mmodel specifies the model for the mediator variable.

The logit outcome model may not be combined with the linear or expmean mediator model; probit rather than logit may be used in these cases.

<i>stat</i>	Description
Stat	
<i>Pearl's labeling of effects</i>	
<i>nie</i>	natural indirect effect
<i>nde</i>	natural direct effect
<i>te</i>	total effect
<i>pnie</i>	pure natural indirect effect
<i>tnde</i>	total natural direct effect
<i>ATE labeling of effects</i>	
<i>aite</i>	average indirect treatment effect; synonym for <i>nie</i>
<i>adte</i>	average direct treatment effect; synonym for <i>nde</i>
<i>ate</i>	total average treatment effect; synonym for <i>te</i>
<i>aitec</i>	average indirect treatment effect with respect to controls; synonym for <i>pnie</i>
<i>adtet</i>	average direct treatment effect with respect to the treated; synonym for <i>tnde</i>
<i>pomeans</i>	potential-outcome means
<i>all</i>	all effects and potential-outcome means
Multiple effects may be specified; default is <i>nie nde te</i> .	
<i>options</i>	Description
Options	
<i>nointeraction</i>	exclude interaction of mediator and treatment
<i>control(# label)</i>	specify the level of <i>tvar</i> that is the control; default is first treatment level
SE/Robust	
<i>vce(vctype)</i>	<i>vcetype</i> may be <i>robust</i> , <i>cluster clustvar</i> , <i>bootstrap</i> , or <i>jackknife</i>
<i>nose</i>	do not estimate standard errors
Reporting	
<i>level(#)</i>	set confidence level; default is <i>level(95)</i>
<i>ateterms</i>	use ATE terminology to label effects
<i>aequations</i>	display auxiliary-equation results
<i>nolegend</i>	suppress table legend
<i>display_options</i>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
<i>optimization_options</i>	control the optimization process; seldom used
Advanced	
<i>force</i>	force estimation when the number of treatment groups exceeds 10
<i>coeflegend</i>	display legend instead of statistics

omvarlist and *mmvarlist* may contain factor variables; see [U] 11.4.3 Factor variables.

bayesboot, *bootstrap*, *by*, *collect*, *jackknife*, and *statsby* are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the *bootstrap* prefix; see [R] *bootstrap*.

pweights, *fweights*, and *iweights* are allowed; see [U] 11.1.6 *weight*.

coeflegend does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

Model

noconstant; see [R] Estimation options.

continuous (*numlist*) specifies that the treatment variable is continuous; *numlist* specifies the values at which the potential-outcome means are to be evaluated, where the first value in the list is taken as the control.

Options

nointeraction excludes the interaction between the treatment and the mediator; by default, the model includes the treatment–mediator interaction.

control (*#* | *label*) specifies the level of *tvar* that is the control. The default is the first treatment level. You may specify the numeric level *#* (a nonnegative integer) or the label associated with the numeric level. *control*() may not be specified with continuous treatments.

Stat

stat specifies the statistics to be estimated. You may select from among five effects, each of which can be labeled according to terminology used by Pearl and others or by ATE terminology. In addition to effects, you may request that potential-outcome means be reported. The default is *nie nde te*.

stat may be one or more of the following:

<i>stat</i>	Definition
<i>nie</i>	natural indirect effect
<i>nde</i>	natural direct effect
<i>te</i>	total effect
<i>pnie</i>	pure natural indirect effect
<i>tnde</i>	total natural direct effect
<i>aite</i>	average indirect treatment effect; synonym for <i>nie</i>
<i>adte</i>	average direct treatment effect; synonym for <i>nde</i>
<i>ate</i>	average treatment effect; synonym for <i>te</i>
<i>aitec</i>	average indirect treatment effect with respect to controls; synonym for <i>pnie</i>
<i>adtet</i>	average direct treatment effect with respect to the treated; synonym for <i>tnde</i>
<i>pomeans</i>	potential-outcome means

all specifies that all effects and potential-outcome means be estimated; specifying *all* is equivalent to specifying *nie nde te pnie tnde pomeans*. When option *ateterms* is specified, *all* is equivalent to specifying *aite adte ate aitec adtet pomeans*.

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

`nose` suppresses calculation of the variance–covariance matrix and standard errors.

Reporting

`level(#)`; see [R] [Estimation options](#).

`ateterms` specifies that ATE terminology be used to label effects. `ateterms` is strictly a labeling option.

This option may not be specified on replay.

`aequations` specifies that the estimation results for the outcome model and the mediator model be displayed. By default, they are not displayed.

`nolegend` suppresses the display of the table legend.

`display_options`: `noci`, `novalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

Optimization

`optimization_options`: `conv_maxiter()`, `conv_ptol()`, `conv_vtol()`, `tracelevel()`, and `[no]log`. See [M-5] [optimize\(\)](#).

`conv_maxiter(#)` specifies the maximum number of iterations. The default is the number set using [set_maxiter](#), which by default is 300.

`conv_ptol(#)` specifies the convergence criteria for the parameters. The default is `conv_ptol(1e-6)`.

`conv_vtol(#)` specifies the convergence criteria for the gradient. The default is `conv_vtol(1e-7)`.

`tracelevel(tracelevel)` allows you to display additional information about the iterative process in the iteration log. `tracelevel` may be `none`, `value`, `tolerance`, `step`, `params`, or `gradient`. See [tracelevel](#) in [M-5] [optimize\(\)](#) for details.

`log` and `nolog` specify whether to display the iteration log. The iteration log is displayed by default unless you used `set_iterlog off` to suppress it; see `set_iterlog` in [R] [set_iter](#).

Advanced

`force` forces estimation when the number of treatment groups exceeds 10. By default, only 10 groups are allowed for multivalued treatments. Do not use the `force` option if the treatment is continuous; instead, use the `continuous()` option.

The following option is available with `mediate` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

Introduction

- Research questions
- Potential outcomes and effect decompositions
- Evaluating assumptions for causal inference
- Estimation of effects

Technical overview of causal mediation

- Mediation analysis in the potential-outcomes framework
- Total, direct, and indirect effects
- Comparison of potential outcomes and classical mediation analysis
- Accounting for treatment–mediator interaction
- Assumptions for causal identification

Examples

- Example 1: A simple causal mediation model
- Example 2: Including covariates and relaxing the no-interaction assumption
- Example 3: Referring to treatment effects using an alternative naming scheme
- Example 4: Causal mediation model with a binary mediator
- Example 5: Causal mediation model with a binary outcome
- Example 6: Causal mediation model with a binary mediator and binary outcome
- Example 7: Causal mediation model with a count mediator
- Example 8: Causal mediation model with an exponential-mean outcome
- Example 9: Causal mediation model with multivalued treatment
- Example 10: Causal mediation model with continuous treatment
- Example 11: Estimating controlled direct effects
- Example 12: Estimating treatment effects on different scales

Introduction

In this entry, we discuss practical and theoretical aspects of causal mediation with one mediator, and we provide examples of fitting these models with the `mediate` command. For a general introduction to causal mediation, we recommend that you first read [\[CAUSAL\] mediate intro](#).

The simplest causal mediation model can be represented by the following causal diagram:

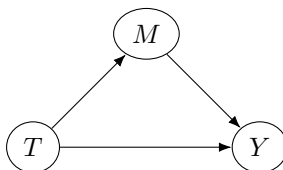


Figure 1

Here we are not only interested in estimating the causal effect of some cause or treatment T on the outcome Y but also interested in better understanding that effect by decomposing it into the indirect effect that is mediated through M and the direct effect. Suppose we wish to analyze the effect of physical exercise (X) on self-perceived well-being (Y). We also want to investigate whether some or all of this effect can be explained because increased exercising leads to an increase in the production of a certain chemical (M) in the body, which in turn leads to improved well-being.

The process of performing causal mediation analysis involves specifying a research question, determining the effects of interest that allow you to answer your research question, verifying that the assumptions for causal interpretation are met, fitting the mediation model, and interpreting the results. In the following sections, we give a conceptual overview of the research questions, effects, assumptions, and estimation with specific focus on the case of one mediator. If you have two mediators, see [\[CAUSAL\] mediate multiple](#).

Research questions

The effects to be estimated in a causal mediation analysis are determined by the research goals. Here are some types of research questions that may arise:

1. A scenario in which the primary interest is to determine whether there is an indirect effect and, if so, to quantify it. In our example above, we might assume there will be some direct impact of exercise on well-being, but we also wonder if and to what extent there is an indirect effect, such as exercise increasing production of a chemical which in turn increases well-being. In this case, we would be interested in decomposing the effects according to [ATE decomposition 1](#).
2. A scenario in which the primary interest is to determine whether there is a direct effect and, if so, to quantify it. Continuing with our example, perhaps we expect an indirect effect, but we also wish to determine if there are any other ways in which exercise causes changes in well-being. In this case, we would be interested in decomposing the effects according to [ATE decomposition 2](#).
3. A scenario in which the primary interest is to determine how the total effect can be decomposed into direct and indirect effects, with focus remaining on all effects and not just direct or just indirect effects. In our example, we simply want to explore the breakdown of the total effect of exercise on well-being into all possible direct and indirect effects. In this case, we would likely be interested in looking at both decompositions, [ATE decomposition 1](#) and [ATE decomposition 2](#).
4. A scenario in which we want to determine the effect of the treatment on the outcome when the mediator is set to a specific value. In our example, we might want to know the effect of exercise on well-being for individuals whose level of this particular chemical is 10, which is the mean value in the population. In this case, we would be interested in [controlled direct effects](#).

Potential outcomes and effect decompositions

Below, we introduce statistics that may be of interest when performing causal mediation analysis. Many of these statistics have a variety of names in the causal mediation literature. See, for instance, [Robins and Greenland \(1992\)](#), [VanderWeele \(2015\)](#), and [Pearl and MacKenzie \(2018\)](#) for some of the various terminology.

1. **Potential-outcome means.** These estimate the population-average value of the outcome that would be expected if everyone was given the treatment (denoted here as $Y[1, M(1)]$) or if everyone was given the control (denoted $Y[0, M(0)]$). In our example, $Y[1, M(1)]$ is the expected average well-being if everyone exercises, and $Y[0, M(0)]$ is the expected average well-being if no one exercises.

In addition, there are two cross-world potential outcomes. These are a bit less intuitive because they correspond to situations that do not exist for any individual in the population. The first is the expected value of the outcome when everyone is treated but counterfactually experiences the value of the mediator associated with being untreated (denoted $Y[1, M(0)]$). The second is the expected value of the outcome when everyone is untreated but counterfactually experiences the value of the mediator associated with being treated (denoted $Y[0, M(1)]$). In our example, $Y[1, M(0)]$ is the expected well-being if everyone was treated but experiencing the chemical level as if untreated. $Y[0, M(1)]$ is the expected value if everyone was untreated but experiencing the chemical level as if treated.

The `mediate` command reports these potential-outcome means when the `pomeans` option is specified.

2. **Total effect (TE).** This estimates the average difference in outcomes that we expect when everyone receives the treatment versus when no one receives the treatment. In our case, it estimates the improvement in well-being that we would expect if everyone exercises versus if no one exercises.

The total effect is also referred to as the average treatment effect (ATE), the total average treatment effect, or the marginal total effect.

The `mediate` command reports this statistic when the `te` option or its synonym `ate` is specified.

The total effect can be decomposed into direct and indirect effects in two ways when we allow for a treatment–mediator interaction.

Decomposition 1. This decomposition separates the direct effect under the untreated mediator condition from the total indirect effect. [Nguyen, Schmid, and Stuart \(2021\)](#) recommend using this decomposition when a direct effect is assumed and the researcher is questioning whether a mediation effect also exists. In our example, we would be interested in this decomposition if we expect that exercise has a direct effect on well-being but want to determine whether a portion of the total effect can be attributed to the increase in the chemical (and if so, how much of the total effect is due to this mediation effect).

3. **Natural direct effect (NDE).** This estimates the average direct effect of the treatment on the outcome when the mediator is held at its value associated with being untreated. It is the difference $Y[1, M(0)] - Y[0, M(0)]$.

This effect is sometimes referred to as the pure natural direct effect or the average direct treatment effect (ADTE). All remaining effects of the treatment on the outcome are included in the natural indirect effect.

The `mediate` command reports this statistic when the `nde` option or its synonym `adte` is specified.

4. **Natural indirect effect (NIE).** This estimates the average indirect effect through a mediator. It is the difference $Y[1, M(1)] - Y[1, M(0)]$.

This effect is sometimes referred to as the total natural indirect effect, causal mediation effect, or average indirect treatment effect (AITE).

The `mediate` command reports this statistic when the `nle` option or its synonym `aite` is specified.

Decomposition 2. This decomposition separates the indirect effect under the untreated condition from the total direct effect. [Nguyen, Schmid, and Stuart \(2021\)](#) recommend using this decomposition when an indirect effect is assumed and the researcher is questioning whether a direct effect also exists. In our example, we would be interested in this decomposition if we believe that exercise increases production of the chemical which in turn increases well-being but want to determine if there is also some change in well-being that is not caused by this mediation effect (and if so, how much of the total effect is not due to the mediation effect).

5. **Pure natural indirect effect (PNIE).** This estimates the average indirect effect of a mediator under the untreated/control condition. It is the difference $Y[0, M(1)] - Y[0, M(0)]$.

This is sometimes referred to as the average indirect treatment effect with respect to controls (AITEC). All remaining effects of the treatment on the outcome are included in the total natural direct effect.

The `mediate` command reports this statistic when the `pnle` option or its synonym `aitec` is specified.

6. **Total natural direct effect (TNDE).** This estimates the average direct treatment effect when the mediator is held at its value associated with being treated. It is the difference $Y[1, M(1)] - Y[0, M(1)]$.

This effect is sometimes referred to as the average direct treatment effect with respect to the treated (ADTET).

The `mediate` command reports this statistic when the `tnde` option or its synonym `adtet` is specified.

When no prior assumptions are made about the existence of direct and indirect effects, [Nguyen, Schmid, and Stuart \(2021\)](#) recommend reporting both Decomposition 1 and Decomposition 2.

7. **Controlled direct effects.** These are the direct effects when the mediator is controlled by setting it to a specific value. After fitting your model with `mediate`, you can estimate the average controlled direct effect with the mediator set to your selected value by using `estat cde`; see [Example 11: Estimating controlled direct effects](#). In our well-being example, controlled direct effects provide the direct effect of exercise on well-being when the chemical is assumed to be a specific value.

Evaluating assumptions for causal inference

Before proceeding to estimation and interpretation of the effects of interest, we need to verify that it is reasonable to give them a causal interpretation in our particular research context.

General assumptions for causal inference are discussed in [\[CAUSAL\] Intro](#), and more precise definitions in the context of mediation are provided in [Assumptions for causal identification](#) below.

To evaluate whether assumptions of causality are met for our mediation model, we must first consider all potential variables, both observed and unobserved, that could affect the relationships among our treatment, mediator, and outcome. If we anticipate that there are confounders (variables that affect both an outcome and a predictor), we must determine whether these confounders will lead to biased results in the estimation of effects from our mediation analysis. In particular, we want to assume that

1. There is no unobserved confounding in the treatment–outcome relationship, and observed confounders are included as covariates in the outcome model.
2. There is no unobserved confounding in the mediator–outcome relationship, and observed confounders are included as covariates in the outcome model.
3. There is no unmeasured confounding in the treatment–mediator relationship, and observed confounders are included as covariates in the mediator model.
4. There are no confounders in the mediator–outcome relationship that are caused by the treatment. No variable exists that affects both the mediator and the outcome and that itself is caused by the treatment.

Estimation of effects

When assumptions are met, the `mediate` command can be used to estimate the causal parameters of interest.

While the effects derived under the potential-outcomes framework required no particular model, we now need to decide how to model our data to obtain estimates.

We first select models for the outcome and the mediator. Outcomes can be continuous, binary, or counts and can be modeled using a linear, exponential-mean, logistic, probit, or Poisson model. Mediators can also be continuous, binary, or counts and can also be modeled using a linear, exponential-mean, logistic, probit, or Poisson model. Covariates can be included in the outcome and mediator models. The treatment may be binary, categorical (multivalued), or continuous.

As a simple example of the `mediate` command, say that we have a binary outcome `y`, a continuous mediator `m`, and a binary treatment `t`. We can fit a mediation model by typing

```
. mediate (y, probit) (m, linear) (t)
```

The first set of parentheses specifies a model for the outcome. The second set of parentheses specifies the model for the mediator. The third set of parentheses defines the treatment. By default, the TE and its decomposition into NDE and NIE are reported in the output.

If we would instead like to see the second type of decomposition, we can obtain the TE, PNIE, and TNDE by typing

```
. mediate (y, probit) (m, linear) (t), te pnie tnde
```

Many combinations of models and effects can be obtained. See [Examples](#) below for additional syntax examples as well as interpretation of the results.

Technical overview of causal mediation

Above, we provided a conceptual introduction to the concepts in causal mediation analysis. Here we more formally define the potential-outcomes framework; explain total, direct, and indirect effects; and introduce the assumptions necessary for causal inference. If you are familiar with the technical aspects of causal mediation and are ready to see `mediate` demonstrated, go directly to [Examples](#).

Mediation analysis in the potential-outcomes framework

The potential-outcomes framework is commonly employed for identifying causal effects. If we go back to the model associated with [figure 1](#) and assume T is a binary treatment, we can identify two sets of potential outcomes, $Y_i(1)$ and $Y_i(0)$. $Y_i(t)$ is the outcome that would be realized if the i th individual were exposed to treatment level t .

Consider a randomized experiment where the experimental group exercises while the control group spends the same amount of time in a resting state. The outcome is subjective well-being that is measured after exercising/resting. If it were possible to observe an individual in both states at the same time, we would observe one outcome value under treatment, $Y_i(1)$, and one value under the control condition, $Y_i(0)$. Then the treatment effect would be the difference $\tau_i = Y_i(1) - Y_i(0)$. In other words, there is a potential outcome for each treatment level that could be administered. Averaging the difference over all individuals in the sample would yield an estimate of the ATE $\tau = E[Y_i(1) - Y_i(0)] = E[Y_i(1)] - E[Y_i(0)]$.

However, it is not possible to observe the same individual under both conditions at the same time; we can only observe one of these while the other is missing. If an individual is treated, we observe $Y_i(1)$, and if not, we observe $Y_i(0)$. This has been coined the “fundamental problem of causal inference” ([Holland 1986](#)). Much of the treatment effects and causal inference literature deals with the question of how to estimate an ATE in the presence of this problem.

In a simple experiment where treatment is randomly assigned, the potential outcomes are independent of treatment assignment and the missing potential outcomes are missing completely at random. In this case, the average of the treatment group outcomes are a valid estimate of $E[Y_i(1)]$, and the average of the control group outcomes are a valid estimate of $E[Y_i(0)]$. Then $\hat{\tau} = \hat{E}[Y_i(1)] - \hat{E}[Y_i(0)]$ where $\hat{E}[Y_i(t)] = 1/N_t \sum_{i=1}^{N_t} 1(T_i = t)Y_i$ is a valid estimator of the ATE. This estimation strategy follows from the identification result that $E[Y_i(t)] = E(Y_i|T_i = t)$ such that $\tau = E[Y_i(1)] - E[Y_i(0)] = E[Y_i|T_i = 1] - E[Y_i|T_i = 0]$.

With observational rather than experimental data, however, the potential outcomes are not independent of the treatment assignment process, and the causal effect is not identifiable without imposing further assumptions such as conditional independence. Stata’s `teffects` suite of commands provides a variety of estimators from this class of treatment-effects estimators.

For further information about identification and estimation in the context of causal models as well as an overview of estimators implemented in Stata, see [\[CAUSAL\] Intro](#). Here we focus on causal inference and potential outcomes specifically for mediation analysis. In this situation, we have another set of potential outcomes, $M_i(1)$ and $M_i(0)$, because M is also affected by the treatment. That is, we can only observe $M_i(1)$ for the group of individuals who were treated, and we can only observe $M_i(0)$ for the controls. If we let t denote the treatment level with respect to the outcome and let t' be the treatment level with respect to the mediator, then the potential outcomes become $Y_i[t, M_i(t')]$.

Similar to the nonmediation case above, we can define a treatment effect as a difference between potential outcomes. The treatment effect is identified if

$$E[Y_i(t, M_i(t'))] = E_{M_i|T_i=t'} E[Y_i|M_i, T_i = t]$$

where $E_{M_i|T_i=t'}$ is the expectation of the mediator conditional on the treatment taking on the value t' and where $E[Y_i|M_i, T_i = t]$ is the expectation of the outcome conditional on the mediator and treatment taking on the value t .

Total, direct, and indirect effects

In mediation analysis, we are interested not only in the total treatment effect but also in its decomposition into direct effects and indirect effects.

Notice that if $t = t'$ for a given potential outcome, the resulting potential outcome is equivalent to $Y_i(t)$. Assuming again a binary treatment, we have that

$$\tau = E[Y_i(1)] - E[Y_i(0)] = E[Y_i(1, M_i(1))] - E[Y_i(0, M_i(0))]$$

In the context of mediation analysis, this treatment effect is also referred to as the total effect.

The total effect can be decomposed further into direct and indirect effects using contrasts between potential-outcome means. The contrasts yielding direct and indirect effects use potential outcomes for which $t \neq t'$, which means we set the treatment level to t and set the mediator to its potential value under treatment level t' .

The natural indirect effect is then defined as

$$\delta(t) \equiv E[Y_i(t, M_i(1))] - E[Y_i(t, M_i(0))], \quad t \in \{0, 1\}$$

Notice that here we “switch” the treatment from on to off in its effect on the mediator but keep the treatment fixed at value t in its effect on the outcome. This natural indirect effect is also sometimes referred to as the causal mediation effect (Imai, Keele, and Tingley 2010).

Likewise, the natural direct effect can be defined as

$$\zeta(t) \equiv E[Y_i(1, M_i(t))] - E[Y_i(0, M_i(t))], \quad t \in \{0, 1\}$$

Comparison of potential outcomes and classical mediation analysis

For those familiar with classical mediation analysis for linear models, it may be helpful to see how the calculation of total, direct, and indirect effects in the potential-outcomes framework relates to the classical product-of-coefficients approach.

We first write our mediation model corresponding with figure 1 as

$$Y_i = \beta_0 + \beta_1 M_i + \beta_2 T_i + \epsilon_i$$

$$M_i = \alpha_0 + \alpha_1 T_i + \nu_i$$

where ϵ_i and ν_i are uncorrelated error terms with means 0 and variances σ_ϵ^2 and σ_ν^2 , respectively.

Let's consider the indirect effect $\delta(1)$. To calculate $\delta(1)$ in the potential-outcomes framework, we need estimates for the potential-outcome means $E[Y_i(1, M_i(1))]$ and $E[Y_i(1, M_i(0))]$. Intuitively, what we want is a world where everyone in the population is exposed to the treatment, that is, $Y_i(1)$, but where we can switch the treatment on and off in regard to the effect of the treatment on the mediator, that is, $M_i(1)$ and $M_i(0)$. The difference when going from the treatment switched on to the treatment switched off will inform us about the effect of the treatment on the outcome that goes through the mediator. First, we write the above model in reduced form:

$$\begin{aligned} E[Y_i | M_i, T_i] &= \beta_0 + \beta_1(\alpha_0 + \alpha_1 T_i) + \beta_2 T_i \\ &= \beta_0 + \beta_1 \alpha_0 + \beta_1 \alpha_1 T_i + \beta_2 T_i \end{aligned}$$

This yields the conditional expectation $E[Y_i | M_i, T_i]$ that we can observe from the data.

To obtain the potential-outcome means, we can modify the reduced-form model by replacing M_i with the expectation of M_i that we would observe if T_i had taken on the value t' for every unit in the population. That is,

$$E[Y_i(t, M_i(t'))] = \beta_0 + \beta_1 E[M_i(t')] + \beta_2 t, \quad t \in \{0, 1\}$$

Thus, to compute the potential-outcome mean $E[Y_i(1, M_i(1))]$, we must set the treatment T_i to 1 in both the outcome and the mediator equations. In other words, we fix both t and t' at 1:

$$\begin{aligned} E[Y_i(1, M_i(1))] &= \beta_0 + \beta_1 E[M_i(t')] + \beta_2 t, \quad t = t' = 1 \\ &= \beta_0 + \beta_1 \alpha_0 + \beta_1 \alpha_1 \times 1 + \beta_2 \times 1 \\ &= \beta_0 + \beta_1 \alpha_0 + \beta_1 \alpha_1 + \beta_2 \end{aligned}$$

However, to compute $E[Y_i(1, M_i(0))]$, we need to set treatment T_i to 1 in the outcome equation and need to set it to 0 in the mediator equation. Specifically, we fix $t' = 0$ and $t = 1$:

$$\begin{aligned} E[Y_i(1, M_i(0))] &= \beta_0 + \beta_1 E[M_i(t')] + \beta_2 t, \quad t = 1; t' = 0 \\ &= \beta_0 + \beta_1 \alpha_0 + \beta_1 \alpha_1 \times 0 + \beta_2 \times 1 \\ &= \beta_0 + \beta_1 \alpha_0 + \beta_2 \end{aligned}$$

Calculating the difference between these two potential-outcome means yields the indirect treatment effect

$$\begin{aligned} \delta(1) &= (\beta_0 + \beta_1 \alpha_0 + \beta_1 \alpha_1 + \beta_2) - (\beta_0 + \beta_1 \alpha_0 + \beta_2) \\ &= \beta_0 + \beta_1 \alpha_0 + \beta_1 \alpha_1 + \beta_2 - \beta_0 - \beta_1 \alpha_0 - \beta_2 \\ &= \beta_1 \alpha_1 \end{aligned}$$

In this case of a linear model, the indirect treatment effect is the product of the treatment coefficient from the mediator equation and the mediator coefficient from the outcome equation. This is congruent with the indirect effect definition in the product-of-coefficients method for mediation as proposed by the classical mediation literature; see [Baron and Kenny \(1986\)](#).

Accounting for treatment–mediator interaction

Notice that the indirect effect we estimated above would be the same if we had estimated $\delta(0)$ instead. Thus far, we assumed that the effect of the mediator on the outcome is the same for both treatment groups. Presumably, a more realistic assumption would be to allow the mediator effects to vary by treatment. This can be achieved by including a treatment–mediator interaction term.

When we allow an interaction, $\delta(0) \neq \delta(1)$. Now we have two indirect effects, one with respect to treatment [$\delta(1)$] and one with respect to controls [$\delta(0)$]. In the following, we will refer to $\delta(1)$ as the NIE and to $\delta(0)$ as the PNIE.

To illustrate computation of the NIE under inclusion of a treatment–mediator interaction, we write a new model

$$Y_i = \beta_0 + \beta_1 M_i + \beta_2 T_i + \beta_3 M_i T_i + \epsilon_i$$

$$M_i = \alpha_0 + \alpha_1 T_i + \nu_i$$

Here, $NIE \equiv E[Y_i(1, M_i(1)) - Y_i(1, M_i(0))]$, whereas $PNIE \equiv E[Y_i(0, M_i(1)) - Y_i(0, M_i(0))]$.

As before, to calculate NIE, we need potential-outcome means $E[Y_i(1, M_i(1))]$ and $E[Y_i(1, M_i(0))]$. Writing the model in reduced form, we get

$$\begin{aligned} E[Y_i | M_i, T_i] &= \beta_0 + \beta_2 T_i + \beta_1(\alpha_0 + \alpha_1 T_i) + \beta_3 T_i(\alpha_0 + \alpha_1 T_i) \\ &= \beta_0 + \beta_2 T_i + (\beta_1 + \beta_3 T_i)(\alpha_0 + \alpha_1 T_i) \end{aligned}$$

Fixing the values for the treatment in both equations accordingly, we have potential-outcome means

$$\begin{aligned} E[Y_i(1, M_i(1))] &= \beta_0 + \beta_2 \times 1 + (\beta_1 + \beta_3 \times 1)(\alpha_0 + \alpha_1 \times 1) \\ &= \beta_0 + \beta_2 + (\beta_1 + \beta_3)(\alpha_0 + \alpha_1) \end{aligned}$$

and

$$\begin{aligned} E[Y_i(1, M_i(0))] &= \beta_0 + \beta_2 \times 1 + (\beta_1 + \beta_3 \times 1)(\alpha_0 + \alpha_1 \times 0) \\ &= \beta_0 + \beta_2 + (\beta_1 + \beta_3)\alpha_0 \end{aligned}$$

Taking the difference yields the NIE

$$E[Y_i(1, M_i(1))] - E[Y_i(1, M_i(0))] = (\beta_1 + \beta_3)\alpha_1$$

We could proceed similarly for the other direct and indirect treatment effects. In this case with treatment–mediator interaction, we also have two direct treatment effects. We have $NDE \equiv E[Y_i(1, M_i(0)) - Y_i(0, M_i(0))]$ and $TNDE \equiv E[Y_i(1, M_i(1)) - Y_i(0, M_i(1))]$.

Notice that both treatment-effect decompositions—NIE and NDE as well as PNIE and TNDE—sum to the total treatment effect (or, as we will call it, the TE).

$$TE \equiv E[Y_i(1, M_i(1)) - Y_i(0, M_i(0))]$$

For further details and discussion on the different direct and indirect effects, as well as a discussion on the differences between causal inference and traditional mediation approaches, see [Nguyen, Schmid, and Stuart \(2021\)](#).

Assumptions for causal identification

The above discussion shows that the estimands of interest are the result of contrasts between potential-outcome means, which are conditional expectations of the outcome with respect to counterfactuals in both the outcome and the mediator equations. In other words, once we can estimate $E[Y_i(t, M_i(t'))]$, we can estimate all direct and indirect treatment effects of interest.

The general form for causal mediation potential-outcome means, which includes covariates X_i , can be written as the integral of the conditional expectation of the outcome with respect to the conditional distribution of the mediator (see [Imai, Keele, and Tingley \[2010\]](#)):

$$E[Y_i(t, M_i(t')) | X_i = x] = \int E[Y_i | M_i = m, T_i = t, X_i = x] dF[m | T_i = t', X_i = x]$$

This is a general, nonparametric solution that applies regardless of the underlying outcome and mediator models. Stata's `mediate` command uses analytical solutions of this integral for a variety of parametric outcome and mediator model combinations. Also, while so far we assumed a binary treatment for simplicity purposes, this approach generalizes straightforwardly to multivalued as well as continuous treatments.

As is the case with nonmediation causal inference, there are assumptions to be met for the estimated effects to be given a causal interpretation. Most notably, a crucial assumption in the nonmediation case is the conditional independence assumption, also known as conditional ignorability assumption, unconfoundedness, or selection on observables. This assumption states that potential outcomes are independent of treatment assignment after conditioning on a set of observed covariates that affect both the outcome and the selection into treatment (see [Imbens \[2004\]](#)). Intuitively, we have a model that resembles an experiment once we account for observable characteristics. More formally, we have that

$$Y_i(t) \perp T_i | X_i$$

In the mediation case, however, we have an additional selection process because “selection” into the mediator is also typically not based on random assignment. This leads to the following two conditional independence assumptions:

$$\begin{aligned} \{Y_i[t, m], M_i(t')\} &\perp T_i | X_i = x \\ Y_i[t, m] &\perp M_i(t') | T_i = t', X_i = x \end{aligned}$$

The first assumption states that treatment assignment is independent of potential outcomes and potential mediators after conditioning on observed (pretreatment) covariates, or confounders. The second assumption states that potential mediators are independent of the potential outcomes given the observed treatment and observed (pretreatment) covariates. Because these assumptions are being made sequentially, this has also been coined the sequential ignorability assumption ([Imai, Keele, and Tingley 2010](#)).

Similarly, there is an additional overlap assumption with causal mediation models. In the nonmediation case, the overlap assumption states that each individual has a positive probability of receiving each treatment:

$$0 < \Pr(T_i = t | X_i = x), \quad t \in \{0, 1\}$$

In the mediation case, the same principle applies to the mediator:

$$0 < p(M_i(t) = m | T_i = t, X_i = x), \quad t \in \{0, 1\}$$

Finally, as is the case with nonmediation treatment-effects models, causal mediation models rely on the stable unit treatment-value assumption, which states that potential outcomes do not depend on treatments assigned to other individuals. For a detailed overview of effect identification and assumptions for causal mediation analysis, see [Nguyen et al. \(2022\)](#).

Examples

Example 1: A simple causal mediation model

Suppose we wish to find out whether exercise affects perceptions of well-being among some population of individuals. To the extent that there is such a causal relationship, we also wish to find out why exercise affects well-being.

We have fictional data from a randomized controlled trial with individuals randomized into two groups—one group performs physical exercise and the other group spends the same amount of time in a resting state. Subjective well-being is measured before and after treatment sessions. In addition, the level of the (fictional) hormone bonotonin is measured. The researchers wish to determine whether exercise leads to an increase in bonotonin levels, which in turn has a positive effect on subjective well-being. Here is an excerpt from our dataset:

```
. use https://www.stata-press.com/data/r19/wellbeing
(Fictional well-being data)
. list wellbeing bonotonin exercise age gender in 1/5, abbreviate(12)
```

	wellbeing	bonotonin	exercise	age	gender
1.	71.73816	196.5467	Control	58	Male
2.	68.66573	195.8572	Exercise	38	Female
3.	71.05155	228.6035	Exercise	53	Female
4.	69.44469	206.6651	Exercise	44	Female
5.	75.62035	261.6855	Exercise	28	Female

To estimate the treatment effects with `mediate`, we specify `wellbeing` as the outcome variable in the first set of parentheses, `bonotonin` as the mediator variable in the second set of parentheses, and `exercise` as the binary treatment variable in the third set of parentheses. Although inclusion of a treatment–mediator interaction is commonly recommended, we specify the `nointeraction` option here to omit the interaction and fit the simplest model possible.

```
. mediate (wellbeing) (bonotonin) (exercise), nointeraction
Iteration 0: EE criterion = 1.627e-25
Iteration 1: EE criterion = 3.061e-28

Causal mediation analysis                                Number of obs = 2,000
Outcome model:      Linear
Mediator model:      Linear
Mediator variable:  bonotonin
Treatment type:      Binary
```

wellbeing	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NIE exercise (Exercise vs Control)	9.694617	.377312	25.69	0.000	8.955099	10.43413
NDE exercise (Exercise vs Control)	2.996658	.2109357	14.21	0.000	2.583231	3.410084
TE exercise (Exercise vs Control)	12.69127	.4005769	31.68	0.000	11.90616	13.47639

Note: Outcome equation does not include treatment–mediator interaction.

In the header of the output, we see that `mediate` fit linear models (the default) for both the outcome and the mediator. Three treatment-effect estimates are reported in the table. TE is the total effect of exercise on well-being and is estimated to be 12.7. The interpretation is the same as for the ATE in the nonmediation case: if everyone in the population exercised, their well-being would be, on average, 12.7 points higher than their well-being would be if no one exercised. The decomposition of the TE into direct and indirect effects is of primary interest. The NIE is estimated to be 9.7, whereas the NDE is estimated to be 3.0. These sum to the total effect of 12.7. The indirect effect is much larger than the direct effect, indicating that the effect of exercise on well-being is largely due to exercise affecting bonotonin levels, which in turn affect well-being. The direct effect of 3.0 is the effect of exercise on well-being beyond the effect through bonotonin.

Instead of comparing estimates of the direct and indirect effects, we might ask what proportion of the total effect is due to mediation. We can answer this question by using `estat proportion`.

```
. estat proportion
```

Proportion mediated

Number of obs = 2,000

wellbeing	Proportion	Robust std. err.	z	P> z	[95% conf. interval]	
exercise (exercise vs control)	.7638805	.0154928	49.31	0.000	.7335151	.7942459

The indirect effect via bonotonin accounts for 76% of the effect of physical activity on well-being, and the remaining 24% is due to other mechanisms.

Example 2: Including covariates and relaxing the no-interaction assumption

The previous example was somewhat unrealistic. For causal inference, we must evaluate the potential of confounding. With causal mediation models, there are three types of confounders we should consider: treatment–outcome confounders, treatment–mediator confounders, and mediator–outcome confounders. A treatment–outcome confounder, for example, is a variable that affects both the selection into treatment and the outcome. If confounders exist and we observe them in our data, we can add them as covariates to the model to prevent biased results.

Above, we noted that the `wellbeing` data come from a randomized controlled trial. In this case, we do not have to worry about treatment–outcome and treatment–mediator confounders because treatment assignment is random. We do, however, need to consider variables such as `age`, `gender`, and `hstatus` (a person’s health status) that affect both the mediator and the outcome. We include these variables as covariates in the model for well-being. We also make our model a bit more realistic by including baseline well-being in the outcome equation and baseline bonotonin level in the mediator equation. In addition, we omit the `nointeraction` option to allow the bonotonin coefficients to vary across treatment groups.

```
. mediate (wellbeing age gender i.hstatus basewell)
>         (bonotonin basebono)
>         (exercise)

Iteration 0: EE criterion = 1.664e-25
Iteration 1: EE criterion = 1.192e-28

Causal mediation analysis                                Number of obs = 2,000

Outcome model:      Linear
Mediator model:      Linear
Mediator variable:  bonotonin
Treatment type:      Binary
```

wellbeing	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NIE exercise (Exercise vs Control)	9.941404	.2307909	43.08	0.000	9.489062	10.39375
NDE exercise (Exercise vs Control)	3.08372	.1684778	18.30	0.000	2.753509	3.41393
TE exercise (Exercise vs Control)	13.02512	.2356989	55.26	0.000	12.56316	13.48709

Note: Outcome equation includes treatment-mediator interaction.

The interpretation of the treatment effects is the same as before. The total effect of exercise on well-being is 13.0. Of this effect, 3.1 is attributed to the direct effect, while the remaining 9.9 is due to the indirect path via bonotonin. These results are similar to our simpler model above.

We find that the expected effect of exercise on well-being is 13.0, but what is the expected well-being when everyone exercises? When no one exercises? We can estimate four such potential-outcome means by specifying the `pomeans` option:

```
. mediate (wellbeing age gender i.hstatus basewell)
>         (bonotonin basebono)
>         (exercise), pomeans

Iteration 0: EE criterion = 1.660e-25
Iteration 1: EE criterion = 1.473e-28

Causal mediation analysis                                Number of obs = 2,000
Outcome model:      Linear
Mediator model:      Linear
Mediator variable:   bonotonin
Treatment type:      Binary
```

wellbeing	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
POMeans						
YOM0	56.94195	.2300492	247.52	0.000	56.49107	57.39284
Y1M0	60.02567	.2571311	233.44	0.000	59.52171	60.52964
YOM1	66.78952	.2642177	252.78	0.000	66.27167	67.30738
Y1M1	69.96708	.232508	300.92	0.000	69.51137	70.42278

Note: Outcome equation includes treatment-mediator interaction.

Y1M1 is an estimate of the potential-outcome mean $E[Y_i(1, M_i(1))]$. If everyone in the population exercised, we would expect the average of well-being to be around 70. The values labeled Y1M0 and YOM1 are estimates of the “cross-world” potential-outcome means $E[Y_i(1, M_i(0))]$ and $E[Y_i(0, M_i(1))]$. For these, we set different counterfactuals in the outcome and mediator equations. In this case, the Y1M0 estimate tells us the expected average well-being if, for the outcome equation, we assume that everyone in the population exercised, but we assume that no one exercised in regard to the effect of treatment on the mediator. If we compare the Y1M1 and Y1M0 estimates, we imagine a world where everyone received the treatment, except that the treatment is switched on and off in its effect on the mediator. The difference between these is $69.96708 - 60.02567 = 9.94141$, which is our NIE reported above.

By default, the TE, NIE, and NDE are computed, but we can request specific effects. For example, we could estimate only the NIE by typing

```
. mediate (wellbeing age gender i.hstatus basewell)
>         (bonotonin basebono)
>         (exercise), nie
```


Alternatively, we could estimate all available effects and potential-outcome means at once by specifying the all option:

```
. mediate (wellbeing age gender i.hstatus basewell)
>         (bonotonin basebono)
>         (exercise), all
Iteration 0: EE criterion = 1.668e-25
Iteration 1: EE criterion = 1.532e-28
Causal mediation analysis                      Number of obs = 2,000
Outcome model:      Linear
Mediator model:      Linear
Mediator variable:   bonotonin
Treatment type:      Binary
```

wellbeing	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
POmeans						
YOM0	56.94195	.2300492	247.52	0.000	56.49107	57.39284
Y1M0	60.02567	.2571311	233.44	0.000	59.52171	60.52964
YOM1	66.78952	.2642177	252.78	0.000	66.27167	67.30738
Y1M1	69.96708	.232508	300.92	0.000	69.51137	70.42278
NIE exercise (Exercise vs Control)	9.941404	.2307909	43.08	0.000	9.489062	10.39375
NDE exercise (Exercise vs Control)	3.08372	.1684778	18.30	0.000	2.753509	3.41393
PNIE exercise (Exercise vs Control)	9.84757	.2318329	42.48	0.000	9.393186	10.30195
TNDE exercise (Exercise vs Control)	3.177554	.1800896	17.64	0.000	2.824585	3.530523
TE exercise (Exercise vs Control)	13.02512	.2356989	55.26	0.000	12.56316	13.48709

Note: Outcome equation includes treatment-mediator interaction.

Here we obtain estimates for two additional effects, PNIE and TNDE, which provide a different decomposition of the TE into direct and indirect effects. In this case, PNIE and TNDE are similar to NIE and NDE, respectively, because the coefficient on the treatment-mediator interaction term is quite small in the model for well-being. We can see the results for the underlying models, including this small coefficient of 0.002, if we add the `aequations` option to our `mediate` command.

Example 3: Referring to treatment effects using an alternative naming scheme

The effects we have discussed so far are sometimes referred to by different names. The default naming conventions originate in the works of Pearl and others. However, we can instead use terminology more closely tied to ATEs if we specify the `ateterms` option:

```
. mediate (wellbeing age gender i.hstatus basewell)
>         (bonotonin basebono)
>         (exercise), all ateterms

Iteration 0: EE criterion = 1.668e-25
Iteration 1: EE criterion = 1.532e-28

Causal mediation analysis                                Number of obs = 2,000

Outcome model:      Linear
Mediator model:      Linear
Mediator variable:  bonotonin
Treatment type:      Binary
```

wellbeing	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]		
POmeans							
Y0M0	56.94195	.2300492	247.52	0.000	56.49107	57.39284	
Y1M0	60.02567	.2571311	233.44	0.000	59.52171	60.52964	
Y0M1	66.78952	.2642177	252.78	0.000	66.27167	67.30738	
Y1M1	69.96708	.232508	300.92	0.000	69.51137	70.42278	
AITE							
exercise (Exercise vs Control)	9.941404	.2307909	43.08	0.000	9.489062	10.39375	
ADTE							
exercise (Exercise vs Control)	3.08372	.1684778	18.30	0.000	2.753509	3.41393	
AITEC							
exercise (Exercise vs Control)	9.84757	.2318329	42.48	0.000	9.393186	10.30195	
ADTET							
exercise (Exercise vs Control)	3.177554	.1800896	17.64	0.000	2.824585	3.530523	
ATE							
exercise (Exercise vs Control)	13.02512	.2356989	55.26	0.000	12.56316	13.48709	

Note: Outcome equation includes treatment-mediator interaction.

Using this notation, ATE can be decomposed into AITE and ADTE or into AITEC and ADTET. Notice that the estimates are the same as in the previous example; they now just have different names.

Example 4: Causal mediation model with a binary mediator

In the previous examples, both outcome and mediator variables were continuous. We now look at the case where the mediator variable is binary. To this end, we use the binary variable `bbonotonin`, an indicator of higher bonotonin levels after exercise, where improvement is defined as an increase of at least 10%. We could use a probit or a logit model for this mediator; we choose a logit model:

```
. mediate (wellbeing age gender i.hstatus basewell)
>         (bbonotonin, logit)
>         (exercise)

Iteration 0:  EE criterion = 8.253e-18
Iteration 1:  EE criterion = 8.223e-18  (backed up)

Causal mediation analysis                                Number of obs = 2,000
Outcome model:      Linear
Mediator model:      Logit
Mediator variable:  bbonotonin
Treatment type:      Binary
```

wellbeing	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NIE						
exercise (Exercise vs Control)	4.41435	.4666635	9.46	0.000	3.499706	5.328994
NDE						
exercise (Exercise vs Control)	8.429238	.5696256	14.80	0.000	7.312792	9.545683
TE						
exercise (Exercise vs Control)	12.84359	.3712965	34.59	0.000	12.11586	13.57132

Note: Outcome equation includes treatment-mediator interaction.

Direct and indirect effect estimates differ from previous results because we used a different bonotonin measure as our mediator variable. However, because we still have the continuous well-being outcome, the interpretation of the effects is the same as before. Here we estimate a total effect of 12.8 with direct and indirect effects of 8.4 and 4.4, respectively. That is, we expect an increase of 12.8 in well-being due to treatment, of which 4.4 is due to an increase in bonotonin levels whereas the remaining 8.4 is due to other mechanisms.

Example 5: Causal mediation model with a binary outcome

Interpretation of effects did not change with a binary mediator, but interpretation does change when we specify a different type of outcome.

To demonstrate, we return to the continuous mediator but use a binary outcome variable. The outcome `bwellbeing` indicates higher well-being and is defined as an increase in well-being of at least 10% compared with the baseline measurement. Using `bwellbeing` as the outcome variable and specifying a probit outcome model, we get

```
. mediate (bwellbeing age gender i.hstatus, probit)
>         (bonotonin basebono, linear)
>         (exercise)

Iteration 0: EE criterion = 2.177e-25
Iteration 1: EE criterion = 9.728e-29

Causal mediation analysis                                Number of obs = 2,000
Outcome model:      Probit
Mediator model:      Linear
Mediator variable:  bonotonin
Treatment type:      Binary
```

bwellbeing	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NIE exercise (Exercise vs Control)	.2346259	.0145763	16.10	0.000	.2060568	.263195
NDE exercise (Exercise vs Control)	.033732	.0237585	1.42	0.156	-.0128338	.0802978
TE exercise (Exercise vs Control)	.2683579	.0200872	13.36	0.000	.2289877	.3077281

Note: Outcome equation includes treatment-mediator interaction.

We interpret the effects as expected differences measured on the probability scale, sometimes referred to as risk differences. The TE of 0.27 indicates that if everyone in the population exercised, we would expect the probability of increased well-being to be 0.27 higher than the probability of increased well-being if no one exercised. In other words, the chance of experiencing an increase in well-being goes up by 27 percentage points when exposed to the exercise treatment. We can see that about 23 points are due to the indirect path via bonotonin, and about 3 points are due to other mechanisms.

Example 6: Causal mediation model with a binary mediator and binary outcome

We could also have the case where both the outcome and the mediator are binary. Here we use a logit model for both:

```
. mediate (bwellbeing age gender i.hstatus, logit)
>         (bbonotonin, logit)
>         (exercise)

Iteration 0: EE criterion = 4.223e-16
Iteration 1: EE criterion = 2.037e-30

Causal mediation analysis                                Number of obs = 2,000
Outcome model:      Logit
Mediator model:      Logit
Mediator variable:   bbonotonin
Treatment type:      Binary
```

bwellbeing	Robust					
	Coefficient	std. err.	z	P> z	[95% conf. interval]	
NIE						
exercise						
(Exercise						
vs						
Control)	.0959618	.0288699	3.32	0.001	.0393778	.1525457
NDE						
exercise						
(Exercise						
vs						
Control)	.1676141	.0358902	4.67	0.000	.0972706	.2379577
TE						
exercise						
(Exercise						
vs						
Control)	.2635759	.0212488	12.40	0.000	.221929	.3052228

Note: Outcome equation includes treatment-mediator interaction.

The interpretation is again in terms of differences in probabilities. We observe a TE of around 0.26, which is partially due to the indirect effect via bbonotonin (0.10) and partially due to other mechanisms (0.17).

Example 7: Causal mediation model with a count mediator

We use a fictional dataset on birthweights and demonstrate how to perform causal mediation analysis when using a Poisson model for a count mediator.

We now pretend to have observational data instead of experimental data. The sample includes women who gave birth to a child. We wish to find out whether socioeconomic status and education of the mother affects the child’s health. The outcome variable is the birthweight of the baby (bweight), and the treatment variable is whether or not the mother has a college degree (college). The mediator variable is the number of cigarettes smoked per day during pregnancy (ncigs). The hypothesis is that women with a higher educational degree are likely to smoke fewer cigarettes and that smoking during pregnancy has negative effects on birthweight. Here is an excerpt from the dataset:

```
. use https://www.stata-press.com/data/r19/birthweight, clear
(Fictional birthweight data)
. list bweight ncigs college ses sespar age in 1/5
```

	bweight	ncigs	college	ses	sespar	age
1.	3621	1	No	5.3581	3.308523	29
2.	3278	0	Yes	9.556957	4.376035	38
3.	3073	1	No	3.980829	6.580275	39
4.	3306	0	Yes	11.17643	12.12075	30
5.	4517	0	Yes	9.026146	4.738766	28

We fit a linear model for the outcome `bweight` and a Poisson model for the mediator `ncigs`, and we specify `college` as the binary treatment variable. Because we have fully observational data, where selection into treatment is no longer completely random, we have to be concerned about all confounder types as mentioned in [Example 2: Including covariates and relaxing the no-interaction assumption](#). We specify several potential confounders as covariates in both equations.

We do not assume that the adverse effects of smoking are different between women with a college degree and women without a college degree. Therefore, we use the `nointeraction` option.

```
. mediate (bweight sespar c.age##c.age)
>         (ncigs sespar c.age##c.age, poisson)
>         (college), nointeract

Iteration 0: EE criterion = 1.939e-21
Iteration 1: EE criterion = 1.937e-21 (backed up)

Causal mediation analysis                                Number of obs = 2,000
Outcome model:      Linear
Mediator model:      Poisson
Mediator variable:   ncigs
Treatment type:      Binary
```

	bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.					
NIE							
college (Yes vs No)	167.3075	21.36134	7.83	0.000	125.4401	209.175	
NDE							
college (Yes vs No)	347.3375	34.44561	10.08	0.000	279.8253	414.8496	
TE							
college (Yes vs No)	514.645	28.65043	17.96	0.000	458.4912	570.7988	

Note: Outcome equation does not include treatment-mediator interaction.

As before, the type of model we use for the mediator does not affect the interpretation of the estimated treatment effects. Effects are expected differences on the scale of the outcome variable. The TE indicates that if all women had a college degree, the average birthweight of newborn babies would be almost 515 grams higher than the average birthweight if no woman had a college degree. Of this weight increase, around 167 grams are due to women with higher educational degrees smoking less, while 347 grams are due to other mechanisms.

Example 8: Causal mediation model with an exponential-mean outcome

Here we use an exponential-mean model for the outcome bweight.

```
. mediate (bweight sespar c.age##c.age, expmean)
> (ncigs sespar c.age##c.age, poisson)
> (college), nointeract
Iteration 0: EE criterion = 3.250e-13
Iteration 1: EE criterion = 1.159e-17

Causal mediation analysis                                Number of obs = 2,000
Outcome model:      Exponential mean
Mediator model:      Poisson
Mediator variable:  ncigs
Treatment type:      Binary
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NIE						
college (Yes vs No)	198.978	23.53279	8.46	0.000	152.8546	245.1014
NDE						
college (Yes vs No)	320.3318	34.47792	9.29	0.000	252.7563	387.9072
TE						
college (Yes vs No)	519.3098	28.70435	18.09	0.000	463.0503	575.5693

Note: Outcome equation does not include treatment-mediator interaction.

Because we are still modeling a continuous outcome, the interpretation does not change. The TE is about 519 grams, of which 199 grams are due to women with a college degree smoking less.

Example 9: Causal mediation model with multivalued treatment

So far we have only dealt with treatments that are binary. However, experiments often have more than two treatment arms, or an observational treatment could consist of multiple categories. Then we would refer to the treatment as multivalued.

To demonstrate, we return to our well-being data and use treatment variable mexercise, which captures three treatment groups: a control group, a group where individuals exercised for 45 minutes, and a group where individuals exercised for 90 minutes. Such a design would allow the researcher to find out whether and how the duration of exercise affects bonotonin levels and thereby well-being. Here we use a linear model for both the outcome and the mediator, and we include the multivalued treatment mexercise as our treatment variable:

```
. use https://www.stata-press.com/data/r19/wellbeing, clear
(Fictional well-being data)
. mediate (wellbeing age gender i.hstatus basewell)
> (bonotonin basebono)
> (mexercise)

Iteration 0: EE criterion = 1.697e-25
Iteration 1: EE criterion = 2.577e-26

Causal mediation analysis                                Number of obs = 2,000
Outcome model:      Linear
Mediator model:      Linear
Mediator variable:  bonotonin
Treatment type:      Multivalued
```

wellbeing	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NIE						
mexercise (45 minutes vs Control)	5.128899	.3505171	14.63	0.000	4.441898	5.815899
(90 minutes vs Control)	9.780537	.2880877	33.95	0.000	9.215895	10.34518
NDE						
mexercise (45 minutes vs Control)	1.197498	.1750038	6.84	0.000	.8544965	1.540499
(90 minutes vs Control)	3.051084	.2071236	14.73	0.000	2.645129	3.457039
TE						
mexercise (45 minutes vs Control)	6.326396	.3894269	16.25	0.000	5.563134	7.089659
(90 minutes vs Control)	12.83162	.2967962	43.23	0.000	12.24991	13.41333

Note: Outcome equation includes treatment-mediator interaction.

We now have two effects per estimand because we compare the two treated groups to the control group. Starting with the TE, we expect nearly a 13-point increase in well-being if everyone in the population exercised for 90 minutes. Of these 13 points, around 10 points are due to the increase in bonotonin levels and 3 points are due to other mechanisms. The results for the 45-minute treatment arm, though expectedly smaller in magnitude, are interpreted similarly.

Example 10: Causal mediation model with continuous treatment

Instead of a binary or multivalued treatment, we could have a continuous treatment variable. With continuous treatments, we have to specify at least two values, one to be the treatment and another to be the control. We return to our birthweight data and use socioeconomic status (*ses*) as our continuous treatment variable. Here are some summary statistics for *ses*:


```
. use https://www.stata-press.com/data/r19/birthweight
(Fictional birthweight data)
. summarize ses
```

Variable	Obs	Mean	Std. dev.	Min	Max
ses	2,000	7.804412	2.287496	1.304026	16.27844

We can see that `ses` ranges from around 1 to 16 and has a mean of about 8. These values, however, do not tell us much because the variable is measured on an arbitrary scale. Therefore, we standardize it so that the resulting variable has a mean of 0 and a standard deviation of 1:

```
. generate std_ses = (ses-r(mean))/r(sd)
```

We will use the new variable, `std_ses`, as our treatment variable. We include the `continuous()` option within the third set of parentheses where we define the treatment. This option tells `mediate` to treat the variable as continuous and to use the values specified within the option as the control and treatment points. The first value is the control, and the remaining values are treatments that are compared with the control. Here we will specify one standard deviation below the mean as our control value and one standard deviation above the mean as our treatment value:

```
. mediate (bweight sespar c.age##c.age, expmean)
>         (ncigs sespar c.age##c.age, poisson)
>         (std_ses, continuous(-1 1)), nointeract

Iteration 0:  EE criterion = 1.470e-12
Iteration 1:  EE criterion = 1.986e-17

Causal mediation analysis                                Number of obs = 2,000

Outcome model:      Exponential mean
Mediator model:      Poisson
Mediator variable:   ncigs
Treatment type:      Continuous

Continuous treatment levels:
  0: std_ses = -1 (control)
  1: std_ses = 1
```

bweight	Robust					
	Coefficient	std. err.	z	P> z	[95% conf. interval]	
NIE std_ses (1 vs 0)	171.3015	14.68778	11.66	0.000	142.514	200.089
NDE std_ses (1 vs 0)	170.0598	32.14841	5.29	0.000	107.05	233.0695
TE std_ses (1 vs 0)	341.3613	31.73741	10.76	0.000	279.1571	403.5655

Note: Outcome equation does not include treatment-mediator interaction.

Even though we used a continuous treatment variable, we interpret the results as before: if everyone in the population had a socioeconomic status one standard deviation above the mean, the birthweight of newborn children would be about 341 grams higher than the birthweight if everyone's status value is one standard deviation below the mean. Of these 341 grams, roughly half is due to women with a higher status smoking less, and the other half is due to other mechanisms.

We could also evaluate the treatment effects at more than two values. Here we use the mean (0) of the standardized variable as the base, and we evaluate the treatment effects at -2 , -1 , 1 , and 2 :

```
. mediate (bweight sespar c.age##c.age, expmean)
>         (ncigs sespar c.age##c.age, poisson)
>         (std_ses, continuous(0 -2 -1 1 2)), nointeract

Iteration 0: EE criterion = 1.470e-12
Iteration 1: EE criterion = 2.767e-17

Causal mediation analysis                                Number of obs = 2,000

Outcome model:      Exponential mean
Mediator model:      Poisson
Mediator variable:   ncigs
Treatment type:      Continuous

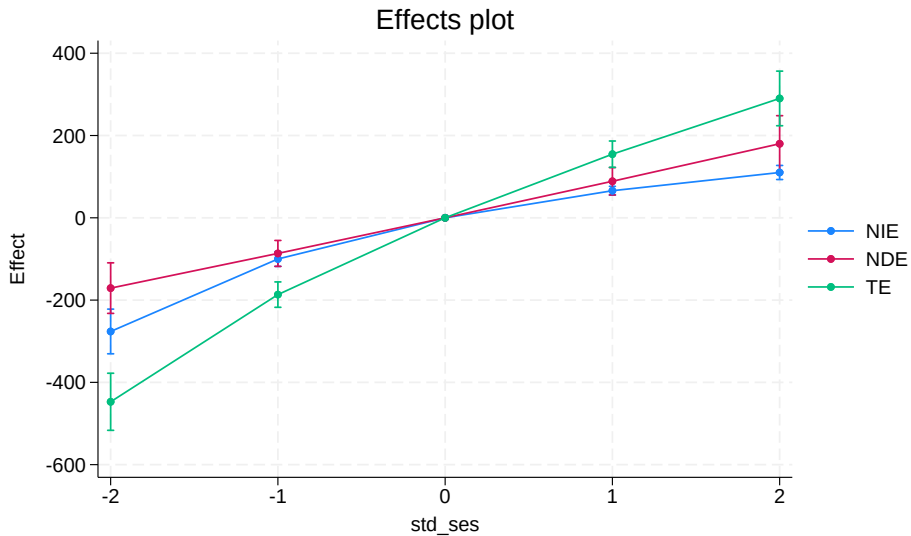
Continuous treatment levels:
0: std_ses = 0 (control)
1: std_ses = -2
2: std_ses = -1
3: std_ses = 1
4: std_ses = 2
```

		Robust				
bweight	Coefficient	std. err.	z	P> z	[95% conf. interval]	
NIE						
std_ses						
(1 vs 0)	-276.2757	27.69004	-9.98	0.000	-330.5471	-222.0042
(2 vs 0)	-100.1155	9.170566	-10.92	0.000	-118.0894	-82.14148
(3 vs 0)	65.84585	5.423096	12.14	0.000	55.21678	76.47493
(4 vs 0)	110.1346	8.724232	12.62	0.000	93.03538	127.2337
NDE						
std_ses						
(1 vs 0)	-170.9012	31.33649	-5.45	0.000	-232.3196	-109.4828
(2 vs 0)	-86.56069	16.08129	-5.38	0.000	-118.0794	-55.04193
(3 vs 0)	88.83929	16.94031	5.24	0.000	55.6369	122.0417
(4 vs 0)	180.0172	34.77372	5.18	0.000	111.8619	248.1724
TE						
std_ses						
(1 vs 0)	-447.1769	35.41401	-12.63	0.000	-516.5871	-377.7667
(2 vs 0)	-186.6761	15.73291	-11.87	0.000	-217.5121	-155.8402
(3 vs 0)	154.6851	16.31969	9.48	0.000	122.6991	186.6712
(4 vs 0)	290.1517	33.85571	8.57	0.000	223.7958	356.5077

Note: Outcome equation does not include treatment-mediator interaction.

We now get four effect estimates for each treatment effect, which capture the expected differences in the outcome with respect to the control point. With multiple effect estimates, it can be convenient to plot the results. We use the postestimation command `estat effectsplot` to do so:

```
. estat effectsplot
```



For more information about `estat effectsplot`, see [\[CAUSAL\] mediate postestimation](#).

Example 11: Estimating controlled direct effects

Controlled direct effects (CDEs) are different from the other estimands we have dealt with so far. Here, rather than having potential outcomes of the form $Y_i(t, M_i(t'))$, we have potential outcomes of the form $Y_i(t|M_i = m)$. That is, we have potential outcomes for each treatment level that are evaluated at set values of the mediator. Thus, CDEs only use the results of the outcome equation. Assuming a binary treatment, the CDE for value m of the mediator is $\text{CDE}(m) = Y_i(1|M_i = m) - Y_i(0|M_i = m)$. CDEs can be estimated using the postestimation command `estat cde`.

To demonstrate, we begin by fitting a mediation model using the well-being data:

```
. use https://www.stata-press.com/data/r19/wellbeing, clear
(Fictional well-being data)
. mediate (bwellbeing age gender i.hstatus, probit)
>         (bbonotonin, probit)
>         (exercise)

Iteration 0:  EE criterion = 4.326e-19
Iteration 1:  EE criterion = 6.955e-32

Causal mediation analysis                                Number of obs = 2,000
Outcome model:      Probit
Mediator model:      Probit
Mediator variable:  bbonotonin
Treatment type:      Binary
```

bwellbeing	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NIE						
exercise (Exercise vs Control)	.0962832	.0288905	3.33	0.001	.0396588	.1529076
NDE						
exercise (Exercise vs Control)	.1672304	.0358936	4.66	0.000	.0968803	.2375805
TE						
exercise (Exercise vs Control)	.2635137	.0212346	12.41	0.000	.2218946	.3051327

Note: Outcome equation includes treatment-mediator interaction.

We fit probit models for the outcome and the mediator, but the type of model is not important here; we can use estat cde after any model fit with mediate.

What is the TE if everyone in the population has a mediator value of 0 (no improvement in bonotonin levels)? To find out, we estimate CDE(0) by specifying the mvalue(0) option with estat cde:

```
. estat cde, mvalue(0)

Controlled direct effect                                Number of obs = 2,000
Mediator variable: bbonotonin
Mediator value = 0
```

	CDE	Delta-method std. err.	z	P> z	[95% conf. interval]	
exercise (Exercise vs Control)	.1605355	.039731	4.04	0.000	.0826641	.2384068

This CDE is around 0.16. The probability of increased well-being when everyone exercises is 0.16 higher than the probability of increased well-being when no one exercises, provided that no one in the population had at least a 10% increase in bonotonin levels.

We could perform the same analysis specifying multiple values for the mediator. Here we wish to estimate both CDE(0) and CDE(1):

```
. estat cde, mvalue(0 1)
Controlled direct effect                                Number of obs = 2,000
Mediator variable: bbonotonin
Mediator values:
  1._at: bbonotonin = 0
  2._at: bbonotonin = 1
```

	Delta-method		z	P> z	[95% conf. interval]	
	CDE	std. err.				
exercise0_at (Exercise vs Control) 1	.1605355	.039731	4.04	0.000	.0826641	.2384068
(Exercise vs Control) 2	.2224479	.0493025	4.51	0.000	.1258166	.3190791

If we “switch on” the mediator, the CDE is higher by around 0.06 points. We could also estimate this difference directly by using the contrast option:

```
. estat cde, mvalue(0 1) contrast
Controlled direct effect                                Number of obs = 2,000
Mediator variable: bbonotonin
Mediator values:
  1._at: bbonotonin = 0
  2._at: bbonotonin = 1
```

	Delta-method		z	P> z	[95% conf. interval]	
	CDE	std. err.				
_at#exercise (2 vs 1) (Exercise vs Control)	.0619124	.0630241	0.98	0.326	-.0616126	.1854373

See [\[CAUSAL\] mediate postestimation](#) for further information about estat cde.

Example 12: Estimating treatment effects on different scales

The mediate command estimates treatment effects on the natural scale of the outcome variable. However, some researchers may want to present their estimated effects on a different scale such as on the odds-ratio or risk-ratio scale if the outcome variable is binary or on the incidence-rate-ratio scale if the outcome variable is a count. The postestimation commands estat rr, estat or, and estat irr transform estimated treatment effects onto these different scales.

To see how this works, we first fit the following model with a binary outcome variable. We could use a probit or logit model for the outcome variable; here we are using probit:

```
. mediate (bwellbeing age gender i.hstatus, probit)
>         (bbonotonin, probit)
>         (exercise), all
Iteration 0: EE criterion = 4.326e-19
Iteration 1: EE criterion = 7.545e-32
Causal mediation analysis                      Number of obs = 2,000
Outcome model:      Probit
Mediator model:      Probit
Mediator variable:  bbonotonin
Treatment type:      Binary
```

bwellbeing	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
POmeans						
YOM0	.3014153	.0146737	20.54	0.000	.2726554	.3301752
Y1M0	.4686457	.0328309	14.27	0.000	.4042984	.5329931
YOM1	.3536121	.0381972	9.26	0.000	.2787469	.4284773
Y1M1	.5649289	.0154435	36.58	0.000	.5346603	.5951976
NIE						
exercise (Exercise vs Control)	.0962832	.0288905	3.33	0.001	.0396588	.1529076
NDE						
exercise (Exercise vs Control)	.1672304	.0358936	4.66	0.000	.0968803	.2375805
PNIE						
exercise (Exercise vs Control)	.0521968	.0348642	1.50	0.134	-.0161357	.1205293
TNDE						
exercise (Exercise vs Control)	.2113169	.041136	5.14	0.000	.1306917	.291942
TE						
exercise (Exercise vs Control)	.2635137	.0212346	12.41	0.000	.2218946	.3051327

Note: Outcome equation includes treatment-mediator interaction.

Given that our outcome variable is binary and our outcome model is probit, the potential-outcome means are averaged probabilities, and because the treatment effects are differences between potential-outcome means, the estimated effects can be interpreted as risk differences. For example, the natural indirect effect is the difference between potential-outcome means Y1M1 and Y1M0:

```
. display _b[P0means:Y1M1] - _b[P0means:Y1M0]
.09628322
```

If, instead of interpreting the effect on the risk-difference scale, we wanted to interpret it on the risk-ratio scale, we could simply compute the ratio of the potential-outcome means:

```
. display _b[P0means:Y1M1] / _b[P0means:Y1M0]
1.2054499
```

This is what `estat rr` is doing:

```
. estat rr
```

Transformed treatment effects					Number of obs = 2,000	
bwelbeing	Risk ratio	Robust std. err.	z	P> z	[95% conf. interval]	
NIE exercise (Exercise vs Control)	1.20545	.0746555	3.02	0.003	1.06766	1.361023
NDE exercise (Exercise vs Control)	1.554817	.132328	5.19	0.000	1.315937	1.837062
PNIE exercise (Exercise vs Control)	1.173172	.1157431	1.62	0.105	.966905	1.423442
TNDE exercise (Exercise vs Control)	1.597595	.1778207	4.21	0.000	1.284469	1.987055
TE exercise (Exercise vs Control)	1.874254	.1043578	11.28	0.000	1.680482	2.09037

The total treatment effect is now decomposed into multiplicative components NIE and NDE as well as PNIE and TNDE. That is, taking their product, rather than their sum, will yield the total effect.

Similarly, we can express all effects on the odds-ratio scale by using estat or:

```
. estat or
Transformed treatment effects                                Number of obs = 2,000
```

bwellbeing	Odds ratio	Robust std. err.	z	P> z	[95% conf. interval]	
NIE exercise (Exercise vs Control)	1.472222	.1708025	3.33	0.001	1.172788	1.848106
NDE exercise (Exercise vs Control)	2.044157	.3042049	4.80	0.000	1.527008	2.736449
PNIE exercise (Exercise vs Control)	1.267908	.1933291	1.56	0.120	.9403671	1.709534
TNDE exercise (Exercise vs Control)	2.373558	.4231302	4.85	0.000	1.673622	3.366217
TE exercise (Exercise vs Control)	3.009452	.281479	11.78	0.000	2.505377	3.614945

Here the total average treatment effect is 3 on the odds-ratio scale and is composed of odds ratios 1.47 and 2.04 in regard to NIE and NDE, respectively, and of odds ratios 1.27 and 2.37 in regard to PNIE and TNDE. Typically, the treatment effects can be interpreted more intuitively on the risk-difference scale, but there may be applications where transforming them to the risk-ratio or odds-ratio scale is desirable.

Notice that estat rr, estat or, and estat irr require estimation of potential-outcome means with mediate. If the fitted model does not contain potential-outcome mean estimates, these estat commands will refit the model. The reestimation does not affect the results, but computation takes longer. See [CAUSAL] mediate postestimation for further information about estat rr, estat or, and estat irr.

Stored results

`mediate` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(interact)</code>	1 if treatment–mediator interaction included, 0 otherwise
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>mediate</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(mvar)</code>	name of mediator variable
<code>e(tvar)</code>	name of treatment variable
<code>e(omodel)</code>	linear, logit, probit, poisson, or expmean
<code>e(mmodel)</code>	linear, logit, probit, poisson, or expmean
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(tvartype)</code>	binary, multivalued, or continuous
<code>e(control)</code>	control level
<code>e(vce)</code>	<i>vce</i> type specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	b V
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals
-----------------------	--

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

`mediate` fits causal mediation models and estimates direct, indirect, and total treatment effects. Using the potential-outcomes framework, the estimated treatment effects are the result of contrasts between potential-outcome means. Without loss of generality, let T_i be a binary treatment, $t \in \{0, 1\}$, for observations $i = 1, \dots, N$, and let Y_i be the outcome and M_i be the mediator variable. The potential-outcome means are

$$\text{POM}_{t,t'} \equiv E[Y_i(t, M_i(t'))]$$

The treatment effects are then defined as follows:

$$\begin{aligned} \text{NIE} &\equiv E[Y_i(1, M_i(1)) - Y_i(1, M_i(0))] \\ \text{NDE} &\equiv E[Y_i(1, M_i(0)) - Y_i(0, M_i(0))] \\ \text{PNIE} &\equiv E[Y_i(0, M_i(1)) - Y_i(0, M_i(0))] \\ \text{TNDE} &\equiv E[Y_i(1, M_i(1)) - Y_i(0, M_i(1))] \\ \text{TE} &\equiv E[Y_i(1, M_i(1)) - Y_i(0, M_i(0))] \end{aligned}$$

Synonyms for NIE, NDE, PNIE, TNDE, and TE are AITE, ADTE, AITEC, ADTET, and ATE, respectively.

The potential-outcome means are the result of an integral of the conditional expectation of the outcome with respect to the conditional distribution of the mediator (Imai, Keele, and Tingley 2010):

$$E[Y_i(t, M_i(t')) | X_i = x] = \int E[Y_i | M_i = m, T_i = t, \mathbf{X}_i = \mathbf{x}] dF[m | T_i = t', \mathbf{X}_i = \mathbf{x}] \quad (1)$$

They are estimated as the sample average

$$\widehat{\text{POM}}_{t,t'} = \frac{1}{N} \sum_{i=1}^N Y_i(t, M_i(t')) | \mathbf{X}_i = \mathbf{x}$$

The estimated treatment effects are then the result of differences between estimated potential-outcome means.

`mediate` uses analytical solutions for the integral in (1) for a variety of parametric outcome and mediator model combinations. Let $\mathbf{X}_i = \{\mathbf{W}_i, \mathbf{Z}_i\}$, the index function of the outcome model is

$$\eta_i^Y = \beta_0 + \beta_1 T_i + \beta_2 M_i + \beta_3 T_i M_i + \mathbf{W}_i \boldsymbol{\gamma} \quad (2)$$

and that of the mediator model is

$$\eta_i^M = \alpha_0 + \alpha_1 T_i + \mathbf{Z}_i \boldsymbol{\zeta} \quad (3)$$

where $\mathbf{W}_i$ and $\mathbf{Z}_i$ are potentially overlapping sets of covariates. If the `nointeraction` option is used, η_i^Y reduces to the simpler function where $\beta_3 = 0$. Depending on which model is specified, the expected values of the outcome and mediator follow these functional forms:

Model	Link function
linear	η_i
exponential mean	e^{η_i}
Poisson	e^{η_i}
logit	$\Pi(\eta_i)$
probit	$\Phi(\eta_i)$

Π and Φ are the cumulative logistic and cumulative normal distribution functions, respectively. Between the outcome and mediator models, all combinations of the above functional forms are allowed with the exception of logit outcome models in combination with linear or exponential-mean mediator models.

mediate uses the estimated coefficients from (2) and (3) to estimate $\widehat{\text{POM}}_{t,t'}$. Calculation of $\widehat{\text{POM}}_{t,t'}$ depends on the combination of functional forms of the outcome and mediator models. We define the following terms, where t represents counterfactual values for the treatment with respect to the outcome equation and t' represents treatment counterfactuals in regard to the mediator equation:

$$\begin{aligned}\nu_t &= \beta_0 + \beta_1 t + \mathbf{W}_t \boldsymbol{\gamma} \\ \xi_{t'} &= \alpha_0 + \alpha_1 t' + \mathbf{Z}_{t'} \boldsymbol{\zeta} \\ \kappa_t &= \beta_2 + \beta_3 t\end{aligned}$$

If the outcome model is linear, we have

$$E[Y_i(t, M_i(t'))] = \nu_t + \Theta_m(\xi_{t'})\kappa_t$$

where $\Theta_m(\cdot)$ denotes the identity function if the mediator model is linear, the cumulative normal distribution function if probit, the cumulative logistic distribution function if logit, and the exponential function if exponential mean or Poisson. In this case, exponential mean and Poisson are synonyms when specifying the mediator model; notice, though, that this is not necessarily the case for other model combinations.

If the outcome model is probit and the mediator model is linear or exponential mean, we have

$$E[Y_i(t, M_i(t'))] = \Phi \left[\frac{\nu_t + \Gamma_m(\xi_{t'})\kappa_t}{\sqrt{1 + \kappa_t^2 \sigma_m^2}} \right]$$

where $\Gamma_m(\cdot)$ denotes the identity function if the mediator model is linear and denotes the exponential function if it is exponential mean, and σ_m^2 is the error variance pertaining to the mediator model.

For probit and logit outcome models in combination with probit and logit mediator models, the potential outcomes are

$$E[Y_i(t, M_i(t'))] = \Lambda_y(\nu_t + \kappa_t)\Lambda_m(\xi_{t'}) + \Lambda_y(\nu_t)\{1 - \Lambda_m(\xi_{t'})\}$$

where $\Lambda_y(\cdot)$ is the cumulative normal distribution function if the outcome model is probit and the cumulative logistic distribution function if the outcome model is logit. $\Lambda_m(\cdot)$ denotes the cumulative normal distribution function if the mediator model is probit and the cumulative logistic distribution function if the mediator model is logit.

Regarding the outcome model, notice that, unlike the case of the mediator model, Poisson and exponential mean always refer to the same model. Thus, we can use the terms Poisson and exponential mean interchangeably in regard to the outcome model. The potential outcomes in the case of the exponential-mean outcome model and the linear or exponential-mean mediator model are

$$E[Y_i(t, M_i(t'))] = e^{\nu_t + \kappa_t \Gamma_m(\xi_{t'}) + (\kappa_t^2 \sigma_m^2)/2}$$

where $\Gamma_m(\cdot)$ is the identity function if the mediator model is linear and is the exponential function if the mediator model is exponential mean. For probit and logit mediator models, the potential outcomes are

$$E[Y_i(t, M_i(t'))] = \Lambda_m(\xi_{t'})e^{\nu_t + \kappa_t} + \{1 - \Lambda_m(\xi_{t'})\}e^{\nu_t}$$

If the outcome model is exponential mean, probit, or logit, and if the mediator model is Poisson, the potential outcomes are

$$E[Y_i(t, M_i(t'))] = \sum_{j=0}^K \Psi_y(\nu_t + \kappa_t j) \frac{e^{j\xi_{t'}} e^{-\xi_{t'}}}{j!}$$

where j indexes the counts of the mediator variable, and $\Psi_y(\cdot)$ denotes the exponential function if the outcome model is exponential mean, denotes the cumulative normal distribution function if the outcome model is probit, and denotes the cumulative logistic distribution function if the outcome model is logit.

`mediate` uses a method of moments estimator, also known as an estimating equations estimator, to estimate all auxiliary and effect parameters as well as their variance–covariance matrix. For more information about the underlying `gmm` command, see [R] [gmm](#).

The postestimation commands `estat or`, `estat rr`, and `estat irr` calculate treatment effects on different scales. If the outcome variable is binary and the model for the outcome variable is `logit` or `probit`, `estat or` computes marginal treatment effects on the odds-ratio scale, whereas `estat rr` computes marginal treatment effects on the risk-ratio scale. If the model for the outcome variable is `poisson` or `expmean`, `estat irr` computes marginal treatment effects on the incidence-rate–ratio scale. Let $Y_{tM_{i'}}'$ be a shorthand for $E[Y_i(t, M_i(t'))]$; the treatment effects on risk-ratio and incidence-rate–ratio scales are ratios of potential-outcome means:

$$\begin{aligned}\text{NIE}^{\text{RR}} &\equiv Y_{1M_1}/Y_{1M_0} \\ \text{NDE}^{\text{RR}} &\equiv Y_{1M_0}/Y_{0M_0} \\ \text{PNIE}^{\text{RR}} &\equiv Y_{0M_1}/Y_{0M_0} \\ \text{TNDE}^{\text{RR}} &\equiv Y_{1M_1}/Y_{0M_1} \\ \text{TE}^{\text{RR}} &\equiv Y_{1M_1}/Y_{0M_0}\end{aligned}$$

For logit and probit outcome models, $Y_{tM_{i'}}'$ are probabilities, and so the treatment effects on odds-ratio scale are

$$\begin{aligned}\text{NIE}^{\text{OR}} &\equiv Y_{1M_1}/(1 - Y_{1M_1})/\{Y_{1M_0}/(1 - Y_{1M_0})\} \\ \text{NDE}^{\text{OR}} &\equiv Y_{1M_0}/(1 - Y_{1M_0})/\{Y_{0M_0}/(1 - Y_{0M_0})\} \\ \text{PNIE}^{\text{OR}} &\equiv Y_{0M_1}/(1 - Y_{0M_1})/\{Y_{0M_0}/(1 - Y_{0M_0})\} \\ \text{TNDE}^{\text{OR}} &\equiv Y_{1M_1}/(1 - Y_{1M_1})/\{Y_{0M_1}/(1 - Y_{0M_1})\} \\ \text{TE}^{\text{OR}} &\equiv Y_{1M_1}/(1 - Y_{1M_1})/\{Y_{0M_0}/(1 - Y_{0M_0})\}\end{aligned}$$

Notice that, with all three of these scales, the total effect is the product of direct and indirect effects, rather than their sum.

CDEs use only the results of the outcome equation. Rather than having potential outcomes of the form $Y_i(t, M_i(t'))$, here we have potential outcomes $Y_i(t|M_i = m)$. That is, we have potential outcomes for each treatment level t that are evaluated at value m of the mediator. $\text{CDE}(m)$ then is the average of the differences between potential outcomes. For binary treatment, $\text{CDE}(m)$ is defined as $Y_i(1|M_i = m) - Y_i(0|M_i = m)$. Letting Y_{tm} be a shorthand for $Y_i(t|M_i = m)$, we have that

$$\begin{aligned}\text{CDE}(m) &\equiv Y_{1m} - Y_{0m} \\ \text{CDE}(m)^{\text{RR}} &\equiv Y_{1m}/Y_{0m} \\ \text{CDE}(m)^{\text{IRR}} &\equiv Y_{1m}/Y_{0m} \\ \text{CDE}(m)^{\text{OR}} &\equiv Y_{1m}/(1 - Y_{1m})/\{Y_{0m}/(1 - Y_{0m})\}\end{aligned}$$

Acknowledgments

Stata has an active research community adding features to the area of causal mediation. We would like to acknowledge their previous and ongoing contributions to the area: `paramed` by Hanhua Liu, Richard Emsley, Graham Dunn, Tyler VanderWeele, and Linda Valeri; `medeff` by Raymond Hicks and Dustin Tingley; `rwrmed` by Ariel Linden, Chuck Huber, and Geoffrey T. Wodtke; and many more. Type `search causal mediation` to see Stata's official and community-contributed features for causal mediation.

References

- Baron, R. M., and D. A. Kenny. 1986. The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of Personality and Social Psychology* 51: 1173–1182. <https://doi.org/10.1037//0022-3514.51.6.1173>.
- Holland, P. W. 1986. Statistics and causal inference. *Journal of the American Statistical Association* 81: 945–960. <https://doi.org/10.2307/2289064>.
- Imai, K., L. Keele, and D. Tingley. 2010. A general approach to causal mediation analysis. *Psychological Methods* 15: 309–334. <https://doi.org/10.1037/a0020761>.
- Imbens, G. W. 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. <https://doi.org/10.1162/003465304323023651>.
- Nguyen, T. Q., I. Schmid, E. L. Ogburn, and E. A. Stuart. 2022. Clarifying causal mediation analysis: Effect identification via three assumptions and five potential outcomes. *Journal of Causal Inference* 10: 246–279. <https://doi.org/10.1515/jci-2021-0049>.
- Nguyen, T. Q., I. Schmid, and E. A. Stuart. 2021. Clarifying causal mediation analysis for the applied researcher: Defining effects based on what we want to learn. *Psychological Methods* 26: 255–271. <https://doi.org/10.1037/met0000299>.
- Pearl, J., and D. MacKenzie. 2018. *The Book of Why: The New Science of Cause and Effect*. New York: Basic Books.
- Robins, J. M., and S. Greenland. 1992. Identifiability and exchangeability for direct and indirect effects. *Epidemiology* 3: 143–155. <https://doi.org/10.1097/00001648-199203000-00013>.
- VanderWeele, T. J. 2015. *Explanation in Causal Inference: Methods for Mediation and Interaction*. New York: Oxford University Press.

Also see

- [CAUSAL] [mediate postestimation](#) — Postestimation tools for mediate
- [CAUSAL] [mediate intro](#) — Introduction to causal mediation analysis
- [CAUSAL] [mediate multiple](#) — Causal mediation analysis (two mediators)⁺
- [CAUSAL] [teffects](#) — Treatment-effects estimation for observational data
- [CAUSAL] [teffects ra](#) — Regression adjustment
- [SEM] [sem](#) — Structural equation model estimation command
- [U] [20 Estimation and postestimation commands](#)

Postestimation commands

Remarks and examples

[predict](#)

Stored results

[estat](#)

Also see

Postestimation commands

The following postestimation commands are of special interest after `mediate`:

Command	Description
estat proportion	proportion mediated
estat cde	controlled direct effects
estat or	effects on the odds-ratio scale
estat rr	effects on the risk-ratio scale
estat irr	effects on the incidence-rate–ratio scale
estat effectsplot	effects plot

The following standard postestimation commands are also available:

Command	Description
estat summarize	summary statistics for the estimation sample
estat vce	variance–covariance matrix of the estimators (VCE)
estimates	cataloging estimation results
etable	table of estimation results
lincom	point estimates, standard errors, testing, and inference for linear combinations of parameters
nlcom	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
predict	treatment effects, conditional means, etc.
predictnl	point estimates, standard errors, testing, and inference for generalized predictions
test	Wald tests of simple and composite linear hypotheses
testnl	Wald tests of nonlinear hypotheses

predict

Description for predict

predict creates a new variable (or variables) containing predictions such as treatment effects, conditional means, linear predictions, and expected values.

Menu for predict

Statistics > Postestimation

Syntax for predict

```
predict [type] { stub* | newvar | newvars } [if] [in]
    [ , effect_statistic tlevel(treat_level) ]

predict [type] { stub* | newvar | newvars } [if] [in]
    [ , po_statistic polevels(t,t') ]

predict [type] newvar [if] [in] [ , fitted_statistic ]
```

<i>effect_statistic</i>	Description
Main	
n <i>ie</i>	natural indirect effect; the default
n <i>de</i>	natural direct effect
t <i>e</i>	total effect
p <i>nie</i>	pure natural indirect effect
t <i>nde</i>	total natural direct effect
i <i>te</i>	indirect treatment effect; synonym for n <i>ie</i>
d <i>te</i>	direct treatment effect; synonym for n <i>de</i>
t <i>te</i>	total treatment effect; synonym for t <i>e</i>
i <i>tec</i>	indirect treatment effect with respect to controls; synonym for p <i>nie</i>
d <i>tet</i>	direct treatment effect with respect to the treated; synonym for t <i>nde</i>
<i>po_statistic</i>	Description
Main	
c <i>mean</i>	conditional mean at treatment levels
<i>fitted_statistic</i>	Description
Main	
x <i>b</i>	linear prediction for outcome model
m <i>edxb</i>	linear prediction for mediator model
m <i>u</i>	expected values for outcome model
m <i>edmu</i>	expected values for mediator model

If you do not specify `tlevel()` and only specify one new variable, then *effect_statistics* assume `tlevel()` specifies the first noncontrol treatment level. You specify one or $t - 1$ new variables with *effect_statistic*, where t is the number of treatment levels.

If you do not specify `polevels()` and only specify one new variable, then `polevels(c, c)` is assumed, where c is the control group. You specify one or d new variables with *cmean*, where d is the number of potential outcomes.

You specify one new variable with *fitted_statistic*.

Options for predict

Main

nie, the default, calculates the natural indirect effect for each noncontrol treatment level or for the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you must specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

nde calculates the natural direct effect for each noncontrol treatment level or for the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you must specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

te calculates the total effect for each noncontrol treatment level or for the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you must specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

pnie calculates the pure natural indirect effect for each noncontrol treatment level or for the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you must specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

tnde calculates the total natural direct effect for each noncontrol treatment level or for the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you must specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

ite calculates the indirect treatment effect for each noncontrol treatment level or for the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you must specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

dte calculates the direct treatment effect for each noncontrol treatment level or for the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you must specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

tte calculates the total treatment effect for each noncontrol treatment level or for the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you must specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

itec calculates the indirect treatment effect with respect to controls for each noncontrol treatment level or for the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you must specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

`dtet` calculates the direct treatment effect with respect to the treated for each noncontrol treatment level or for the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you must specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

`tlevel(treat_level)` specifies the treatment level for prediction.

`cmean` calculates the conditional mean for each potential outcome $Y(t, M(t'))$ or the potential outcome specified in `polevels()`. If you specify the `polevels()` option, you must specify only one new variable; otherwise, you must specify a new variable for each potential outcome.

`polevels(t, t')` specifies the values of the treatment for which potential outcomes are to be calculated. The first value, t , refers to the value that the treatment is set to in the outcome equation; the second value, t' , refers to the value of the treatment in the mediator equation.

`xb` calculates the linear prediction for the outcome model.

`medxb` calculates the linear prediction for the mediator model.

`mu` calculates the expected values of the dependent variable of the outcome model.

`medmu` calculates the expected values of the dependent variable of the mediator model.

estat

Description for estat

`estat proportion` calculates the indirect effect as a proportion of the total effect.

`estat cde` calculates controlled direct effects.

`estat or` calculates effects on the odds-ratio scale after `mediate` with the `logit` or `probit` outcome model.

`estat rr` calculates effects on the risk-ratio scale after `mediate` with the `logit` or `probit` outcome model.

`estat irr` calculates effects on the incidence-rate–ratio scale after `mediate` with the `poisson` or `expmean` outcome model.

`estat effectsplot` plots the estimated effects. Typically, this is useful if there are more than two treatment groups in the case of a multivalued treatment or if a continuous treatment is evaluated at more than two points. By default, `estat effectsplot` plots the effects estimated in the previous `mediate` command.

Menu for estat

Statistics > Postestimation

Syntax for estat

Proportion mediated

```
estat proportion [ , prop_options ]
```

Controlled direct effects

```
estat cde, mvalue(numlist) [ cde_options ]
```

Effects on the odds-ratio scale

```
estat or [ , scale_options ]
```

Effects on the risk-ratio scale

```
estat rr [ , scale_options ]
```

Effects on the incidence-rate–ratio scale

```
estat irr [ , scale_options ]
```

Effects plot

```
estat effectsplot [ , effectsplot_options ]
```

<i>prop_options</i>	Description
<u>level</u> (#)	set confidence level; default is level(95)
<u>percent</u>	display percentage instead of proportion
<u>force</u>	force calculations to proceed in case of conflicting signs
<u>nolegend</u>	suppress table legend
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

collect is allowed; see [U] 11.1.10 Prefix commands.

<i>cde_options</i>	Description
* <u>mvalue</u> (<i>numlist</i>)	value of the mediator variable
<u>rr</u>	controlled direct effect on risk-ratio scale
<u>or</u>	controlled direct effect on odds-ratio scale
<u>irr</u>	controlled direct effect on incidence-rate-ratio scale
<u>level</u> (#)	set confidence level; default is level(95)
<u>contrast</u>	differences of controlled direct effects
<u>nolegend</u>	suppress table legend
<u>atmeans</u>	controlled direct effect at the means of covariates
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

* mvalue(*numlist*) is required.

collect is allowed; see [U] 11.1.10 Prefix commands.

<i>scale_options</i>	Description
<u>level</u> (#)	set confidence level; default is level(95)
<u>nolegend</u>	suppress table legend
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

`estat or`, `estat rr`, and `estat irr` require estimation of potential-outcome means with `mediate`.

If no potential-outcome means were estimated, `estat or`, `estat rr`, and `estat irr` will refit the model in the background; the reestimation does not affect the results, but computation takes longer.

collect is allowed; see [U] 11.1.10 Prefix commands.

<i>effectsplot_options</i>	Description
<i>When mediate had Pearl's labeling of effects</i>	
<code>nie</code>	plot natural indirect effects
<code>nde</code>	plot natural direct effects
<code>te</code>	plot total effects
<code>pnie</code>	plot pure natural indirect effects
<code>tnde</code>	plot total natural direct effects
<i>When mediate had ATE labeling of effects</i>	
<code>aite</code>	plot average indirect treatment effects
<code>adte</code>	plot average direct treatment effects
<code>ate</code>	plot average treatment effects
<code>aitec</code>	plot average indirect treatment effects with respect to controls
<code>adtet</code>	plot average direct treatment effects with respect to the treated
Main	
<code>noci</code>	do not plot confidence intervals
Plot	
<code><i>plot_options</i></code>	affect rendition of all effect plots
<code>plot#opts(<i>plot_options</i>)</code>	affect rendition of #th effect plot
<code>recast(<i>plotype</i>)</code>	plot effects using <i>plotype</i>
CI plot	
<code><i>ciopts(rcap_options)</i></code>	affect rendition of confidence intervals
<code><i>ci#opts(rcap_options)</i></code>	affect rendition of #th confidence interval plot
<code>recastci(<i>plotype</i>)</code>	plot confidence intervals using <i>plotype</i>
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
Add plots	
<code>addplot(<i>plot</i>)</code>	add other plots to the graph
Y axis, X axis, Titles, Legend, Overall	
<code><i>twoway_options</i></code>	any options other than by() documented in [G-3] <i>twoway_options</i>
<i>plot_options</i>	Description
<code><i>marker_options</i></code>	change look of markers (color, size, etc.)
<code><i>marker_label_options</i></code>	add marker labels; change look or position
<code><i>cline_options</i></code>	change look of the line

Options for estat proportion

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [\[U\] 20.8 Specifying the width of confidence intervals](#).

`percent` specifies to calculate percentages. By default, `estat proportion` calculates proportions.

`force` forces calculations to proceed in case of conflicting signs. By default, `estat proportion` issues an error message if opposite signs among indirect, direct, and total effects are detected. In that case, the result is typically not interpretable in a meaningful way.

`nolegend` suppresses the display of the table legend.

display_options: `noci`, `nopvalues`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [\[R\] Estimation options](#).

Options for estat cde

`mvalue(numlist)` specifies the value of the mediator variable at which to evaluate the controlled direct effect. If the causal mediation model contained a continuous treatment variable, only a single value may be specified. `mvalue()` is required.

`rr` specifies to calculate controlled direct effect on the risk-ratio scale after `mediate` with the logit or probit outcome model.

`or` specifies to calculate controlled direct effect on the odds-ratio scale after `mediate` with the logit or probit outcome model.

`irr` specifies to calculate controlled direct effect on the incidence-rate–ratio scale after `mediate` with the poisson or `expmean` outcome model.

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [\[U\] 20.8 Specifying the width of confidence intervals](#).

`contrast` specifies to calculate differences of controlled direct effects between evaluations at different points of the mediator, where the base effect is the one defined by the first value in `mvalue()`; this option requires at least two evaluation points to be specified in `mvalue()`.

`nolegend` suppresses the display of the table legend.

`atmeans` specifies to evaluate the controlled direct effect at the means of covariates. By default, the counterfactual predictions are averaged over the covariates.

display_options: `noci`, `nopvalues`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [\[R\] Estimation options](#).

Options for estat or, estat rr, and estat irr

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [\[U\] 20.8 Specifying the width of confidence intervals](#).

`nolegend` suppresses the display of the table legend.

display_options: `noci`, `nopvalues`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [\[R\] Estimation options](#).

Options for estat effectsplot

`nie`, `nde`, `te`, `pnie`, `tnde`, `aite`, `adte`, `ate`, `aitec`, and `adtet` specify to plot the respective treatment effects. For these effects to be plotted, they must be part of the model estimates. By default, `estat effectsplot` plots the effects estimated in the previous `mediate` command.

Main

`noci` removes plots of the pointwise confidence intervals. The default is to plot the confidence intervals.

Plot

plot_options affects the rendition of all effect plots. The *plot_options* can affect the size and color of markers, whether and how the markers are labeled, and whether and how the points are connected; see [G-3] *marker_options*, [G-3] *marker_label_options*, and [G-3] *cline_options*.

These settings may be overridden for specific plots by using the `plot#opts()` option.

`plot#opts(plot_options)` affects the rendition of the *#th* effect plot. The *plot_options* can affect the size and color of markers, whether and how the markers are labeled, and whether and how the points are connected; see [G-3] *marker_options*, [G-3] *marker_label_options*, and [G-3] *cline_options*.

`recast(plotype)` specifies that effects be plotted using *plotype*. *plotype* may be `scatter`, `line`, `connected`, `bar`, `area`, `spike`, `dropline`, or `dot`; see [G-2] **graph twoway**. When `recast()` is specified, the plot-rendition options appropriate to the specified *plotype* may be used in lieu of *plot_options*. For details on those options, follow the appropriate link from [G-2] **graph twoway**.

CI plot

`ciopts(rcap_options)` affects the rendition of confidence intervals; see [G-3] *rcap_options*.

These settings may be overridden for specific confidence interval plots with the `ci#opts()` option.

`ci#opts(rcap_options)` affects the rendition of the *#th* confidence interval; see [G-3] *rcap_options*.

`recastci(plotype)` specifies that confidence intervals be plotted using *plotype*. *plotype* may be `rarea`, `rbar`, `rspike`, `rcap`, `rcapsym`, `rline`, `rconnected`, or `rscatter`; see [G-2] **graph twoway**. When `recastci()` is specified, the plot-rendition options appropriate to the specified *plotype* may be used in lieu of *rcap_options*. For details on those options, follow the appropriate link from [G-2] **graph twoway**.

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] 20.8 Specifying the width of confidence intervals.

Add plots

`addplot(plot)` provides a way to add other plots to the generated graph; see [G-3] *addplot_option*.

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] *twoway_options*, excluding `by()`. These include options for titling the graph (see [G-3] *title_options*) and for saving the graph to disk (see [G-3] *saving_option*).

Remarks and examples

Below we provide examples for `predict`. To see an example of `estat proportion`, see [A simple causal mediation model](#) in `[CAUSAL] mediate`. To see an example of `estat cde`, see [Estimating controlled direct effects](#) in `[CAUSAL] mediate`. To see an example of `estat effectsplot`, see [Causal mediation model with continuous treatment](#) in `[CAUSAL] mediate`. An example of `estat rr` and `estat or` is shown in [Estimating treatment effects on different scales](#) in `[CAUSAL] mediate`.

► Example 1: Predicting individual-level direct, indirect, and total effects

We can use `predict` to make a variety of predictions from the fitted mediation model, such as individual-level direct, indirect, and total effects; potential outcomes; and linear predictions and expected values of the outcome and mediator. Suppose we have the following mediation model with binary outcome and binary mediator:

```
. use https://www.stata-press.com/data/r19/wellbeing
(Fictional well-being data)

. mediate (bwellbeing age gender i.hstatus basewell, logit)
>         (bbonotonin, logit)
>         (exercise)

Iteration 0:  EE criterion = 8.253e-18
Iteration 1:  EE criterion = 6.067e-33

Causal mediation analysis                                Number of obs = 2,000

Outcome model:      Logit
Mediator model:      Logit
Mediator variable:   bbonotonin
Treatment type:      Binary
```

bwellbeing	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
NIE exercise (Exercise vs Control)	.1052971	.0170666	6.17	0.000	.0718472	.1387471
NDE exercise (Exercise vs Control)	.1524917	.0208284	7.32	0.000	.1116689	.1933146
TE exercise (Exercise vs Control)	.2577889	.0143	18.03	0.000	.2297613	.2858164

Note: Outcome equation includes treatment-mediator interaction.

Using `predict` without options yields estimated individual-level natural indirect effects:

```
. predict nie
(option nie assumed; natural indirect effect)
```

We could go ahead and predict individual-level direct and total effects by using options `nde` and `te`, respectively:

```
. predict nde, nde
. predict te, te
```

Here is an excerpt from the data showing the predicted effects for five individuals:

```
. list nie nde te in 1/5
```

	nie	nde	te
1.	.0504899	.2496191	.3001091
2.	.1693522	.1037404	.2730926
3.	.2145208	.3612216	.5757424
4.	.0265223	.1576028	.1841251
5.	.2005004	.3735286	.574029

We can see that the indirect and direct effects sum to the total effect for each individual. The differences in effects between individuals are due to their differences in covariates. Had we fit the model without covariates, the predicted effects would be constant over the sample.

If we look at the sample means of the newly generated variables `nie`, `nde`, and `te`, we can see that their averages match the estimates from `mediate` for NIE, NDE, and TE, respectively:

```
. summarize nie nde te
```

Variable	Obs	Mean	Std. dev.	Min	Max
nie	2,000	.1052971	.0883299	.000014	.2529227
nde	2,000	.1524917	.1354418	.0001044	.3877526
te	2,000	.2577889	.2037248	.0001184	.5757825



► Example 2: Predicting potential outcomes

In addition to individual-level effects, we can also predict individual-level potential outcomes by using the `cmean` option. By default, `predict` with `cmean` will compute the potential outcomes for the control level of the treatment variable. For example, if the treatment variable is binary and takes on the values 0 and 1, where 0 is the control level, we will predict potential outcomes $Y_i[0, M_i(0)]$:

```
. predict po_y0m0, cmean
```

We can also target other potential outcomes by using the `polevels()` option. For instance, to compute potential outcomes $Y_i[1, M_i(0)]$, we specify option `polevels(1,0)`:

```
. predict po_y1m0, cmean polevels(1,0)
```

If we wish to predict all potential outcomes at once, we can use the `stub*` notation:

```
. predict po_*, cmean
```


In this case, there are four potential outcomes available, so Stata creates four new variables. Using `describe`, we can also see that the new variables are labeled according to the estimated potential outcome:

```
. describe po_?
Variable      Storage      Display      Value
  name         type         format      label      Variable label
-----
po_1          float        %9.0g              Conditional mean, Y[0,M(0)]
po_2          float        %9.0g              Conditional mean, Y[1,M(0)]
po_3          float        %9.0g              Conditional mean, Y[0,M(1)]
po_4          float        %9.0g              Conditional mean, Y[1,M(1)]
```



Stored results

`estat proportion` stores the following results in `r()`:

- Scalars
 - `r(N)` number of observations
- Macros
 - `r(title)` title in estimation output
- Matrices
 - `r(b)` vector of estimated proportions or percentages
 - `r(V)` variance–covariance matrix of the estimates
 - `r(table)` matrix containing the estimates with their standard errors, test statistics, *p*-values, and confidence intervals

`estat cde` stores the following results in `r()`:

- Scalars
 - `r(N)` number of observations
- Macros
 - `r(title)` title in estimation output
- Matrices
 - `r(b)` vector of estimated controlled direct effects or their contrasts
 - `r(V)` variance–covariance matrix of the estimates
 - `r(table)` matrix containing the estimates with their standard errors, test statistics, *p*-values, and confidence intervals

`estat or`, `estat rr`, and `estat irr` store the following results in `r()`:

- Scalars
 - `r(N)` number of observations
 - `r(level)` confidence level
- Matrices
 - `r(b)` vector of transformed treatment effects (log scale)
 - `r(V)` variance–covariance matrix of the estimates
 - `r(table)` matrix containing the estimates with their standard errors, test statistics, *p*-values, and confidence intervals

Also see

[\[CAUSAL\] mediate](#) — Causal mediation analysis (one mediator)

[\[U\] 20 Estimation and postestimation commands](#)

⁺The two-mediator version of this command is part of [StataNow](#).

Description	Quick start	Menu	Syntax
Options	Remarks and examples	Stored results	Methods and formulas
Reference	Also see		

Description

`mediate` fits causal mediation models and estimates effects of a treatment on an outcome. The treatment effect can occur both directly and indirectly through other variables, mediators.

In this entry, we describe how `mediate` can be used to perform causal mediation analysis with two mediator variables. The relationship between the mediators can take one of two forms. Causal mediation models where there is no causal relationship between mediators are known as parallel mediation models. Causal mediation models where the mediators are causally ordered, with one mediator predicting the other, are known as sequential mediation models.

`mediate` estimates direct, indirect, and total effects. The natural direct and indirect effects reflect the finest possible decomposition of total effects into their path-specific components. Because the number of effects is potentially large with two mediators, `mediate` also offers coarser decompositions.

For an introduction to causal mediation analysis, see [\[CAUSAL\] mediate intro](#). For the use of `mediate` to fit causal mediation models with one mediator, see [\[CAUSAL\] mediate](#).

Quick start

Fit a parallel mediation model with two mediators `m1` and `m2`, outcome variable `y`, and treatment `t`

```
mediate (y) (m2) (m1) (t)
```

Same as above, but with covariates in the outcome and mediator equations

```
mediate (y x1 x2) (m2 x1 x3) (m1 x1 x2 x3) (t)
```

Fit a sequential mediation model where `m1` is a predictor of `m2`

```
mediate (y x1 x2) (m2 x1 x3) (m1 x1 x2 x3) (t), sequential
```

Same as above, but add treatment–mediator interactions in the equation for `y`

```
mediate (y x1 x2) (m2 x1 x3) (m1 x1 x2 x3) (t), sequential tinteraction
```

Same as above, but add the mediator–mediator interaction in the equation for `y`

```
mediate (y x1 x2) (m2 x1 x3) (m1 x1 x2 x3) (t), sequential tinteraction ///
minteraction
```

Same as above, but add the treatment–mediator interaction in the equation for `m2`

```
mediate (y x1 x2) (m2 x1 x3) (m1 x1 x2 x3) (t), sequential tinteraction ///
minteraction meqtinteraction
```

Same as above, but add covariate interactions with the treatment and mediators in the outcome equation

```
mediate (y x1 x2 i.t#c.(x1 x2) c.m2#c.(x1 x2) c.m1#c.(x1 x2))          ///
        (m2 x1 x3) (m1 x1 x2 x3) (t),                                  ///
        sequential tinteraction minteraction meqtinteraction
```

Sequential mediation model with type-1 mediator-specific effects

```
mediate (y x1 x2) (m2 x3) (m1 x1) (t), sequential mseffects(m1)
```

Fit sequential mediation model with continuous treatment t_2 , and evaluate at values 0 and 3 of the treatment with 0 as the control

```
mediate (y x1 x2) (m2 x3) (m1 x1) (t2, continuous(0 3)), sequential
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Causal mediation

Syntax

```
mediate (ovar [omvarlist, noconstant])
        (mvar2 [mmvarlist2, noconstant])
        (mvar1 [mmvarlist1, noconstant])
        (tvar [, continuous(numlist)] [if] [in] [weight] [, stat options])
```

ovar is a continuous outcome of interest.

omvarlist specifies the covariates in the outcome model.

*mvar*₂ is a continuous mediator variable (the second mediator for sequential mediators).

*mmvarlist*₂ specifies the covariates in the *mvar*₂ model.

*mvar*₁ is a continuous mediator variable (the first mediator for sequential mediators).

*mmvarlist*₁ specifies the covariates in the *mvar*₁ model.

tvar is the treatment variable and may be binary or continuous.

<i>stat</i>	Description
Stat	
<code>nie</code>	natural indirect effects
<code>nde</code>	natural direct effects
<code>te</code>	total effect
<code>pomeans</code>	potential-outcome means

Multiple effects may be specified. The default is `nie nde te`. You may not specify `nie`, `nde`, or `te` in combination with `pomeans`.

<i>options</i>	Description
Options	
<code>parallel</code>	fit a parallel mediation model; the default
<code>sequential</code>	fit a sequential mediation model
<code>tinteraction (mvar₁ mvar₂)</code>	include interactions of mediators and treatment in outcome model
<code>minteraction</code>	include interaction among mediators in outcome model
<code>meqtinteraction</code>	include interaction of first mediator and treatment in second mediator model
<code>mseffects (mvar₁ mvar₂)</code>	estimate mediator-specific natural effects for mediator <code>mvar₁</code> or mediator <code>mvar₂</code>
<code>control (# label)</code>	specify the level of <code>tvar</code> that is the control; default is first treatment level
SE/Robust	
<code>vce (vcetype)</code>	<code>vcetype</code> may be <code>robust</code> , <code>cluster clustvar</code> , <code>bootstrap</code> , or <code>jackknife</code>
<code>nose</code>	do not estimate standard errors
Reporting	
<code>level (#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>nolegend</code>	suppress table legend
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
<code>optimization_options</code>	control the optimization process; seldom used
Advanced	
<code>coeflegend</code>	display legend instead of statistics

`omvarlist` and `mmvarlist` may contain factor variables; see [U] 11.4.3 Factor variables.

`bayesboot`, `bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the `bootstrap` prefix; see [R] bootstrap.

`pweights`, `fweights`, and `iweights` are allowed; see [U] 11.1.6 weight.

`coeflegend` does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

Model

`noconstant`; see [R] Estimation options.

`continuous(numlist)` specifies that the treatment variable is continuous; *numlist* specifies the values at which the potential-outcome means are to be evaluated, where the first value in the list is taken as the control.

Options

`parallel` fits a parallel mediation model where there is no causal relationship between mediators; this is the default.

`sequential` fits a sequential mediation model where there is a causal order among mediators. In this case, the mediators should be specified such that *mvar*₁ is assumed to predict *mvar*₂.

`tinteraction[(mvar1 | mvar2)]` includes the interaction between the treatment and one or both mediators in the model for the outcome. If one of the mediator names, *mvar*₁ or *mvar*₂, is specified, only the interaction with that mediator is included. If `tinteraction` is specified without a mediator name, the model includes treatment–mediator interactions for both mediator variables.

`minteraction` includes the interaction between two mediators in the model for the outcome.

When `minteraction` is specified with `tinteraction`, the three-way interaction between the treatment and both mediators is also included in the model for the outcome.

`meqinteraction` includes the interaction between the treatment and the first mediator in the model for the second mediator. This option is available only when the `sequential` option is specified to fit a sequential mediation model.

`mseffects(mvar1 | mvar2)` specifies that mediator-specific natural effects for the specified mediator be reported. Specifying *mvar*₁ yields type-1 mediator-specific natural effects. Specifying *mvar*₂ yields type-2 mediator-specific natural effects. This option is available only when the `sequential` option is specified to fit a sequential mediation model.

`control(# | label)` specifies the level of *tvar* that is the control. The default is the first treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with continuous treatments.

Stat

`stat` specifies the statistics to be estimated. In addition to natural effects, you may request that potential-outcome means be reported. The default is `nie nde te`.

`stat` may be one or more of the following:

<i>stat</i>	Definition
<code>nie</code>	natural indirect effects
<code>nde</code>	natural direct effects
<code>te</code>	total effect
<code>pomeans</code>	potential-outcome means

Multiple effects may be specified. You may not specify `nie`, `nde`, or `te` in combination with `pomeans`.

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

`nose` suppresses calculation of the variance–covariance matrix and standard errors.

Reporting

`level(#)`; see [R] [Estimation options](#).

`aequations` specifies that the estimation results for the outcome model and the mediator model be displayed. By default, they are not displayed.

`nolegend` suppresses the display of the table legend.

`display_options`: `noci`, `nopvalues`, `noomit`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

Optimization

`optimization_options`: `conv_maxiter()`, `conv_ptol()`, `conv_vtol()`, `tracelevel()`, and `[no]log`. See [M-5] [optimize\(\)](#).

`conv_maxiter(#)` specifies the maximum number of iterations. The default is the number set using `set_maxiter`, which by default is 300.

`conv_ptol(#)` specifies the convergence criteria for the parameters. The default is `conv_ptol(1e-6)`.

`conv_vtol(#)` specifies the convergence criteria for the gradient. The default is `conv_vtol(1e-7)`.

`tracelevel(tracelevel)` allows you to display additional information about the iterative process in the iteration log. `tracelevel` may be `none`, `value`, `tolerance`, `step`, `params`, or `gradient`. See [tracelevel](#) in [M-5] [optimize\(\)](#) for details.

`log` and `nolog` specify whether to display the iteration log. The iteration log is displayed by default unless you used `set_iterlog off` to suppress it; see `set_iterlog` in [R] [set iter](#).

Advanced

The following option is available with `mediate` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

- Introduction*
- Sequential mediators*
- Modeling and estimation*
- Mediator-specific natural effects*
- Parallel mediators*
- Identification assumptions*
- Examples*
 - Example 1: Parallel causal mediation model*
 - Example 2: Treatment–mediator interactions*
 - Example 3: Mediator–mediator interaction*
 - Example 4: Accounting for confounding variables*
 - Example 5: Additional interactions*
 - Example 6: Sequential causal mediation model*
 - Example 7: Sequential model with treatment–mediator interactions*
 - Example 8: Including the mediator–mediator interaction*
 - Example 9: Complete set of interactions*
 - Example 10: Estimating mediator-specific natural effects*
 - Example 11: Adding interactions of covariates with mediators and treatment*
 - Example 12: Sensitivity analysis*
 - Example 13: Estimating controlled direct effects*
 - Example 14: Continuous treatment*

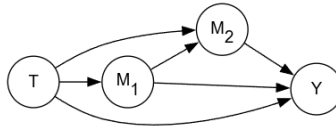
Introduction

In this entry, we discuss practical and theoretical aspects of causal mediation with two mediators, and we provide examples of fitting these models with the `mediate` command. For a general introduction to causal mediation, we recommend that you first read [\[CAUSAL\] mediate intro](#).

In causal mediation analysis with two mediators, we first need to consider the relationship among the mediators. When one mediator is presumed to predict the other, we refer to them as sequential mediators. When there is no presumption of one mediator predicting the other, we refer to them as parallel mediators. The `mediate` command allows us to estimate total, direct, and indirect effects of interest when mediators are sequential or parallel. The estimates are derived from a potential-outcomes framework, and the framework for the two-mediator case is an extension of the framework for the one-mediator case that is described in [\[CAUSAL\] mediate](#). With two mediators, however, the number and complexity of potential outcomes increases. In the following sections, we provide an introduction into the intricacies of causal mediation with multiple mediators that is more technical than the one in [\[CAUSAL\] mediate intro](#). For both sequential and parallel mediation models, we formally define the potential outcomes and the effects and discuss how to estimate them. We also show all of the ways that the total effects can be decomposed into the direct and indirect effects. We follow the derivations and definitions presented in [Daniel et al. \(2015\)](#). You may prefer to read this technical introduction and then the examples demonstrating `mediate`, or if you are already familiar with technical concepts of mediation analysis for multiple mediators, you may prefer to go directly to [Examples](#) and then refer back to the definitions and decompositions as they relate to the examples at hand.

Sequential mediators

Suppose we have the following causal model involving treatment T , first mediator M_1 , second mediator M_2 , and outcome Y :



There are four path-specific effects of interest: the direct effect from T to Y (that is, the effect of T on Y not mediated by either M_1 or M_2); one effect through M_1 alone; one through M_2 alone; and one through both M_1 and M_2 . These four effects combined will sum to the total causal effect.

We define the effects based on potential outcomes. When we had a single mediator, our potential outcomes took the form $Y_i\{t, M_i(t')\}$, where t can be counterfactually set in two places. Multiple mediators increase the complexity and number of the potential outcomes. With these two mediators, and given the two are causally ordered, our potential outcomes of interest take on the form $Y_i[t, M_{1i}(t'), M_{2i}\{t'', M_{1i}(t''')\}]$. Notice that we can counterfactually set t in four places now: with respect to the outcome, with respect to the first mediator, with respect to the second mediator, and with respect to the first mediator as component of the potential outcomes of the second mediator. The direct and indirect effects are derived as expected differences in these potential outcomes. This leads to eight variations of each path-specific effect rather than the two variations (pure and total) in the single-mediator case. With 4 path-specific effects of interest and 8 variations of each one, we can estimate a maximum of 32 effects (plus the total effect) for 2 causally ordered mediators.

With a single mediator, there were two possible decompositions of the total effect into direct and indirect effects. With multiple mediators, there are many ways that total effect can be decomposed into sums of its path-specific components. In fact, with k mediators, there are $2^k!$ decompositions. Thus, for models with a single mediator, we have only two decompositions. However, in the case with 2 sequential mediators, we have 24 distinct decompositions of the total causal effect. This becomes infeasible quickly if we want to add more mediators. With only 3 mediators, the number of distinct decompositions grows to 40,320, and with 4 mediators, we would be looking at over 20 trillion decompositions.

The following table shows an overview of all possible effect estimands as well as their corresponding potential-outcome means contrast. When you use `mediate` to estimate effects, you can refer to this table to see how each effect is defined in terms of the potential outcomes.

Table 1

Path	Effect	Definition
Direct (through neither M_1 nor M_2)	NDE-000	$E(Y[1, M_1(0), M_2\{0, M_1(0)\}] - Y[0, M_1(0), M_2\{0, M_1(0)\}])$
	NDE-100	$E(Y[1, M_1(1), M_2\{0, M_1(0)\}] - Y[0, M_1(1), M_2\{0, M_1(0)\}])$
	NDE-010	$E(Y[1, M_1(0), M_2\{1, M_1(0)\}] - Y[0, M_1(0), M_2\{1, M_1(0)\}])$
	NDE-001	$E(Y[1, M_1(0), M_2\{0, M_1(1)\}] - Y[0, M_1(0), M_2\{0, M_1(1)\}])$
	NDE-110	$E(Y[1, M_1(1), M_2\{1, M_1(0)\}] - Y[0, M_1(1), M_2\{1, M_1(0)\}])$
	NDE-101	$E(Y[1, M_1(1), M_2\{0, M_1(1)\}] - Y[0, M_1(1), M_2\{0, M_1(1)\}])$
	NDE-011	$E(Y[1, M_1(0), M_2\{1, M_1(1)\}] - Y[0, M_1(0), M_2\{1, M_1(1)\}])$
	NDE-111	$E(Y[1, M_1(1), M_2\{1, M_1(1)\}] - Y[0, M_1(1), M_2\{1, M_1(1)\}])$
Indirect through M_1 only	NIE ₁ -000	$E(Y[0, M_1(1), M_2\{0, M_1(0)\}] - Y[0, M_1(0), M_2\{0, M_1(0)\}])$
	NIE ₁ -100	$E(Y[1, M_1(1), M_2\{0, M_1(0)\}] - Y[1, M_1(0), M_2\{0, M_1(0)\}])$
	NIE ₁ -010	$E(Y[0, M_1(1), M_2\{1, M_1(0)\}] - Y[0, M_1(0), M_2\{1, M_1(0)\}])$
	NIE ₁ -001	$E(Y[0, M_1(1), M_2\{0, M_1(1)\}] - Y[0, M_1(0), M_2\{0, M_1(1)\}])$
	NIE ₁ -110	$E(Y[1, M_1(1), M_2\{1, M_1(0)\}] - Y[1, M_1(0), M_2\{1, M_1(0)\}])$
	NIE ₁ -101	$E(Y[1, M_1(1), M_2\{0, M_1(1)\}] - Y[1, M_1(0), M_2\{0, M_1(1)\}])$
	NIE ₁ -011	$E(Y[0, M_1(1), M_2\{1, M_1(1)\}] - Y[0, M_1(0), M_2\{1, M_1(1)\}])$
	NIE ₁ -111	$E(Y[1, M_1(1), M_2\{1, M_1(1)\}] - Y[1, M_1(0), M_2\{1, M_1(1)\}])$
Indirect through M_2 only	NIE ₂ -000	$E(Y[0, M_1(0), M_2\{1, M_1(0)\}] - Y[0, M_1(0), M_2\{0, M_1(0)\}])$
	NIE ₂ -100	$E(Y[1, M_1(0), M_2\{1, M_1(0)\}] - Y[1, M_1(0), M_2\{0, M_1(0)\}])$
	NIE ₂ -010	$E(Y[0, M_1(1), M_2\{1, M_1(0)\}] - Y[0, M_1(1), M_2\{0, M_1(0)\}])$
	NIE ₂ -001	$E(Y[0, M_1(0), M_2\{1, M_1(1)\}] - Y[0, M_1(0), M_2\{0, M_1(1)\}])$
	NIE ₂ -110	$E(Y[1, M_1(1), M_2\{1, M_1(0)\}] - Y[1, M_1(1), M_2\{0, M_1(0)\}])$
	NIE ₂ -101	$E(Y[1, M_1(0), M_2\{1, M_1(1)\}] - Y[1, M_1(0), M_2\{0, M_1(1)\}])$
	NIE ₂ -011	$E(Y[0, M_1(1), M_2\{1, M_1(1)\}] - Y[0, M_1(1), M_2\{0, M_1(1)\}])$
	NIE ₂ -111	$E(Y[1, M_1(1), M_2\{1, M_1(1)\}] - Y[1, M_1(1), M_2\{0, M_1(1)\}])$
Indirect through M_1 and M_2	NIE ₁₂ -000	$E(Y[0, M_1(0), M_2\{0, M_1(1)\}] - Y[0, M_1(0), M_2\{0, M_1(0)\}])$
	NIE ₁₂ -100	$E(Y[1, M_1(0), M_2\{0, M_1(1)\}] - Y[1, M_1(0), M_2\{0, M_1(0)\}])$
	NIE ₁₂ -010	$E(Y[0, M_1(1), M_2\{0, M_1(1)\}] - Y[0, M_1(1), M_2\{0, M_1(0)\}])$
	NIE ₁₂ -001	$E(Y[0, M_1(0), M_2\{1, M_1(1)\}] - Y[0, M_1(0), M_2\{1, M_1(0)\}])$
	NIE ₁₂ -110	$E(Y[1, M_1(1), M_2\{0, M_1(1)\}] - Y[1, M_1(1), M_2\{0, M_1(0)\}])$
	NIE ₁₂ -101	$E(Y[1, M_1(0), M_2\{1, M_1(1)\}] - Y[1, M_1(0), M_2\{1, M_1(0)\}])$
	NIE ₁₂ -011	$E(Y[0, M_1(1), M_2\{1, M_1(1)\}] - Y[0, M_1(1), M_2\{1, M_1(0)\}])$
	NIE ₁₂ -111	$E(Y[1, M_1(1), M_2\{1, M_1(1)\}] - Y[1, M_1(1), M_2\{1, M_1(0)\}])$

Source: Definitions from [Daniel et al. \(2015\)](#).

This defines all eight estimands for each of the four path-specific effects. Each estimand is defined as the expected value of a particular contrast of potential outcomes.

The first block shows estimands related to the natural direct effect. Notice that the contrasts of potential outcomes take on the form $Y[1, M_1(t'), M_2\{t'', M_1(t''')\}] - Y[0, M_1(t'), M_2\{t'', M_1(t''')\}]$. That is, for each difference in potential outcomes, the first argument is fixed to 1 in the first potential outcome and is fixed to 0 in the second, while the remaining arguments consist of permutations of 0s and 1s. Fixing the first argument is what makes them direct effects. Zooming in on the first effect, NDE-000, we see the corresponding contrast is $Y[1, M_1(0), M_2\{0, M_1(0)\}] - Y[0, M_1(0), M_2\{0, M_1(0)\}]$. Here all the remaining arguments are fixed at 0, which is why it is labeled 000. This is an extension of the pure

natural direct effect because the potential values of the mediators are based on fixing the treatment at the level it would naturally take had there been no treatment. However, unlike in the single-mediator case where we have only 2 variations of effects, pure and total, here we have 8 types of effects: 000, 100, 010, 001, 110, 101, 011, and 111. The NDE-111 effect can be considered an extension of the total natural direct effect because the potential mediators are held at the level that they would be observed at had everyone in the population been exposed to treatment.

The second block of estimands reflects natural indirect effects through M_1 alone. Notice that here we fix the second argument of the potential outcomes to 1 and 0, while the remaining arguments again consist of 0/1 permutations. For example, the $\text{NIE}_1\text{-100}$ natural indirect effect is defined based on the contrast $Y[1, M_1(1), M_2\{0, M_1(0)\}] - Y[1, M_1(0), M_2\{0, M_1(0)\}]$, where the free arguments are 1, 0, and 0, hence the label 100.

Similarly, the third block shows the indirect effects through the second mediator alone. Here it is the third argument of the potential outcomes that is fixed. For example, the $\text{NIE}_2\text{-110}$ effect is defined as based on the contrast $Y[1, M_1(1), M_2\{1, M_1(0)\}] - Y[1, M_1(1), M_2\{0, M_1(0)\}]$.

Finally, the NIE_{12} effects are indirect effects through both the first and second mediator. The natural $\text{NIE}_{12}\text{-111}$ indirect effect, for instance, is defined based on the contrast $Y[1, M_1(1), M_2\{1, M_1(1)\}] - Y[1, M_1(1), M_2\{1, M_1(0)\}]$.

The effects discussed above constitute a single decomposition of the total causal effect into its path-specific components. That is, if we let TE be the total effect, we have that

$$\text{TE} = \text{NDE-000} + \text{NIE}_1\text{-100} + \text{NIE}_2\text{-110} + \text{NIE}_{12}\text{-111}$$

Recall that, in the single-mediator case, we had only two decompositions (pure natural direct effect plus total natural indirect effect and total natural direct effect plus pure natural indirect effect). With 2 sequential mediators, however, there are 24 ways to decompose the total effect into a sum of path-specific effects. The possible decompositions of the total effect into sums of its path-specific components are listed in the table below. When you use `mediate` to fit a model with two sequential mediators, you can refer back to this table to see all the ways the reported effects can be summed to the total effect.

Table 2

Decomposition	Total effect
1	$\text{NDE-000} + \text{NIE}_1\text{-100} + \text{NIE}_2\text{-110} + \text{NIE}_{12}\text{-111}$
2	$\text{NDE-000} + \text{NIE}_1\text{-100} + \text{NIE}_2\text{-111} + \text{NIE}_{12}\text{-110}$
3	$\text{NDE-000} + \text{NIE}_1\text{-110} + \text{NIE}_2\text{-100} + \text{NIE}_{12}\text{-111}$
4	$\text{NDE-000} + \text{NIE}_1\text{-101} + \text{NIE}_2\text{-111} + \text{NIE}_{12}\text{-100}$
5	$\text{NDE-000} + \text{NIE}_1\text{-111} + \text{NIE}_2\text{-100} + \text{NIE}_{12}\text{-101}$
6	$\text{NDE-000} + \text{NIE}_1\text{-111} + \text{NIE}_2\text{-101} + \text{NIE}_{12}\text{-100}$
7	$\text{NDE-100} + \text{NIE}_1\text{-000} + \text{NIE}_2\text{-110} + \text{NIE}_{12}\text{-111}$
8	$\text{NDE-100} + \text{NIE}_1\text{-000} + \text{NIE}_2\text{-111} + \text{NIE}_{12}\text{-110}$
9	$\text{NDE-010} + \text{NIE}_1\text{-110} + \text{NIE}_2\text{-000} + \text{NIE}_{12}\text{-111}$
10	$\text{NDE-010} + \text{NIE}_1\text{-111} + \text{NIE}_2\text{-000} + \text{NIE}_{12}\text{-101}$
11	$\text{NDE-001} + \text{NIE}_1\text{-101} + \text{NIE}_2\text{-111} + \text{NIE}_{12}\text{-000}$
12	$\text{NDE-001} + \text{NIE}_1\text{-111} + \text{NIE}_2\text{-101} + \text{NIE}_{12}\text{-000}$
13	$\text{NDE-110} + \text{NIE}_1\text{-000} + \text{NIE}_2\text{-010} + \text{NIE}_{12}\text{-111}$
14	$\text{NDE-110} + \text{NIE}_1\text{-010} + \text{NIE}_2\text{-000} + \text{NIE}_{12}\text{-111}$
15	$\text{NDE-101} + \text{NIE}_1\text{-000} + \text{NIE}_2\text{-111} + \text{NIE}_{12}\text{-010}$
16	$\text{NDE-101} + \text{NIE}_1\text{-001} + \text{NIE}_2\text{-111} + \text{NIE}_{12}\text{-000}$
17	$\text{NDE-011} + \text{NIE}_1\text{-111} + \text{NIE}_2\text{-000} + \text{NIE}_{12}\text{-001}$
18	$\text{NDE-011} + \text{NIE}_1\text{-111} + \text{NIE}_2\text{-001} + \text{NIE}_{12}\text{-000}$
19	$\text{NDE-111} + \text{NIE}_1\text{-000} + \text{NIE}_2\text{-010} + \text{NIE}_{12}\text{-011}$
20	$\text{NDE-111} + \text{NIE}_1\text{-000} + \text{NIE}_2\text{-011} + \text{NIE}_{12}\text{-010}$
21	$\text{NDE-111} + \text{NIE}_1\text{-010} + \text{NIE}_2\text{-000} + \text{NIE}_{12}\text{-011}$
22	$\text{NDE-111} + \text{NIE}_1\text{-001} + \text{NIE}_2\text{-011} + \text{NIE}_{12}\text{-000}$
23	$\text{NDE-111} + \text{NIE}_1\text{-011} + \text{NIE}_2\text{-000} + \text{NIE}_{12}\text{-001}$
24	$\text{NDE-111} + \text{NIE}_1\text{-011} + \text{NIE}_2\text{-001} + \text{NIE}_{12}\text{-000}$

Source: Decompositions from [Daniel et al. \(2015\)](#).

Modeling and estimation

So far, we have concerned ourselves only with the nonparametric part of identification of treatment effects based on potential outcomes but not with how to estimate these. In what follows, we use linear models with interaction effects to approximate our effects of interest.

Given the causal mediation model with two causally ordered mediators, and ignoring covariates, we can specify the following system of three equations:

$$\begin{aligned}
 Y &= \beta_0 + \beta_1 T + \beta_2 M_1 + \beta_3 M_2 + \beta_4 T M_1 + \beta_5 T M_2 + \beta_6 M_1 M_2 + \beta_7 T M_1 M_2 + \epsilon_Y \\
 M_2 &= \gamma_0 + \gamma_1 T + \gamma_2 M_1 + \gamma_3 T M_1 + \epsilon_{M_2} \\
 M_1 &= \alpha_0 + \alpha_1 T + \epsilon_{M_1}
 \end{aligned}$$

Here Y is our outcome variable, M_1 and M_2 are our first and second mediators, respectively, and T is our treatment variable. Error terms ϵ_{M_1} , ϵ_{M_2} , and ϵ_Y are assumed to have zero mean and be independent. Parameters $\beta \equiv (\beta_0, \dots, \beta_7)$, $\gamma \equiv (\gamma_0, \dots, \gamma_3)$, and $\alpha \equiv (\alpha_0, \alpha_1)$ are to be estimated. Notice that the equations contain all possible interactions among treatment and mediators to retain model flexibility. Looking at the equation for the first mediator (M_1), we see that this includes only the treatment variable on the right-hand side. The equation for M_2 contains M_1 , T , and a $T \times M_1$ interaction term. That is, we allow

the effect of the first mediator on the second to vary by treatment group. Finally, the outcome equation contains both mediators and treatment as well as their interactions, a mediator–mediator interaction term, and the three-way interaction $T \times M_1 \times M_2$.

To estimate the coefficients for each of the equations, we can simply fit each equation separately and solve for the coefficients by ordinary least squares (OLS). We can then work out the form for our potential-outcomes of interest. Plugging the equations for M_1 and M_2 into the outcome equation, we get

$$\begin{aligned} Y[t, M_1(t'), M_2\{t'', M_1(t''')\}] = & \beta_0 + \beta_1 t + (\beta_2 + \beta_4 t)(\alpha_0 + \alpha_1 t' + \epsilon_{M_1}) + \\ & (\beta_3 + \beta_5 t)(\gamma_0 + \gamma_1 t'' + (\gamma_2 + \gamma_3 t'')(\alpha_0 + \alpha_1 t''' + \epsilon_{M_1}) + \epsilon_{M_2}) + \\ & (\beta_6 + \beta_7 t) \left\{ (\alpha_0 + \alpha_1 t' + \epsilon_{M_1})(\gamma_0 + \gamma_1 t'' + (\gamma_2 + \gamma_3 t'')(\alpha_0 + \alpha_1 t''' + \epsilon_{M_1}) + \epsilon_{M_2}) \right\} + \epsilon_Y \end{aligned}$$

Replacing error terms with their expectations yields

$$\begin{aligned} E(Y[t, M_1(t'), M_2\{t'', M_1(t''')\}]) = & \beta_0 + \beta_1 t + (\beta_2 + \beta_4 t)(\alpha_0 + \alpha_1 t') + \\ & (\beta_3 + \beta_5 t)(\gamma_0 + \gamma_1 t'' + (\gamma_2 + \gamma_3 t'')(\alpha_0 + \alpha_1 t''')) + \\ & (\beta_6 + \beta_7 t) \left\{ (\alpha_0 + \alpha_1 t')(\gamma_0 + \gamma_1 t'' + \right. \\ & \left. (\gamma_2 + \gamma_3 t'')(\alpha_0 + \alpha_1 t''')) \right\} + \\ & (\beta_6 + \beta_7 t)(\gamma_2 + \gamma_3 t'')E[\epsilon_{M_1} \epsilon_{M_1}] \\ = & \beta_0 + \beta_1 t + (\beta_2 + \beta_4 t)(\alpha_0 + \alpha_1 t') + \\ & (\beta_3 + \beta_5 t)(\gamma_0 + \gamma_1 t'' + (\gamma_2 + \gamma_3 t'')(\alpha_0 + \alpha_1 t''')) + \\ & (\beta_6 + \beta_7 t) \left\{ (\alpha_0 + \alpha_1 t')(\gamma_0 + \gamma_1 t'' + \right. \\ & \left. (\gamma_2 + \gamma_3 t'')(\alpha_0 + \alpha_1 t''')) \right\} + \\ & (\beta_6 + \beta_7 t)(\gamma_2 + \gamma_3 t'')\sigma_{M_1}^2 \end{aligned} \quad (1)$$

At this point, we could simply plug in values for t , t' , t'' , and t''' according to which potential-outcome mean we wish to compute. For example, to compute natural direct effect NDE-000, we need the difference of potential outcome means $E[Y_i(1, M_1(0), M_2(0, M_1(0)))] - E[Y_i(0, M_1(0), M_2(0, M_1(0)))]$. Taking the first, we have

$$\begin{aligned} E[Y_i(1, M_1(0), M_2(0, M_1(0)))] = & \beta_0 + \beta_1 + (\beta_2 + \beta_4)\alpha_0 + (\beta_3 + \beta_5)(\gamma_0 + \gamma_2\alpha_0) + \\ & (\beta_6 + \beta_7) \left\{ \alpha_0(\gamma_0 + \gamma_2\alpha_0) + \gamma_2\sigma_{M_1}^2 \right\} \end{aligned}$$

And for the second potential-outcome mean, we have

$$\begin{aligned} E[Y_i(0, M_1(0), M_2(0, M_1(0)))] = & \beta_0 + \beta_2\alpha_0 + \beta_3(\gamma_0 + \gamma_2\alpha_0) + \\ & \beta_6 \left\{ \alpha_0(\gamma_0 + \gamma_2\alpha_0) + \gamma_2\sigma_{M_1}^2 \right\} \end{aligned}$$

We could now plug in our OLS estimates for β , γ , and α for both potential-outcome means and then take the difference to get an estimate of NDE-000.

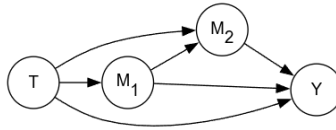
In addition to the coefficients, however, we also need to plug in an estimate of $\sigma_{M_1}^2$, which is the residual variance of the M_1 equation. Notice that our reduced form in (1) contains the expected value of the product of two error terms. Following Daniel et al. (2015), we can think of the error terms as arising from the fact that $M_1(t)$ can be evaluated at a different t when $t' \neq t'''$. If we set $t' = 0$ and $t''' = 1$, for example, we can write the potential outcomes for mediator M_1 as

$$\begin{aligned} M_1(t') &= M_1(0) = \alpha_0 + \epsilon_{M_1}^{t'} \\ M_1(t''') &= M_1(1) = \alpha_0 + \alpha_1 + \epsilon_{M_1}^{t'''} \end{aligned}$$

Here, to calculate the expected value of the product $\epsilon_{M_1}^{t'} \epsilon_{M_1}^{t'''}$, we would need to have information about their covariance, which is not available from the data. Therefore, this conditional correlation between $M_1(0)$ and $M_1(1)$ has to be set by assumption. A reasonable assumption is that the unobservables $\epsilon_{M_1}^{t'}$ and $\epsilon_{M_1}^{t'''}$ are the same regardless of how treatment is set counterfactually. Under this assumption, $\epsilon_{M_1}^{t'}$ and $\epsilon_{M_1}^{t'''}$ are perfectly correlated with correlation coefficient $\rho_{M_1^{t'} M_1^{t'''}} = 1$, and the expected value of the error term product yields $\sigma_{M_1}^2$. However, the value of $\rho_{M_1^{t'} M_1^{t'''}}$ can be varied to perform a sensitivity analysis. After we run `mediate`, this can be done using the `estat smsensitivity` postestimation command. [Example 12: Sensitivity analysis](#) shows an example of this type of sensitivity analysis.

Mediator-specific natural effects

As we have seen, a causal mediation model with two causally ordered mediators and a full set of interactions yields a complex set of estimands. One option to reduce this complexity is to perform a coarser decomposition. So far, we have focused on the finest possible decomposition along four path-specific effects. However, we could combine some of the paths, which reduces the number of estimands at the expense of the decomposition being coarser. Consider again the following causal diagram:



The four path-specific effects we considered so far are $T \rightarrow Y$, $T \rightarrow M_1 \rightarrow Y$, $T \rightarrow M_2 \rightarrow Y$, and $T \rightarrow M_1 \rightarrow M_2 \rightarrow Y$. However, if, for example, we were primarily interested in the indirect effect $T \rightarrow M_1 \rightarrow Y$, we could combine the paths $T \rightarrow M_2 \rightarrow Y$ and $T \rightarrow M_1 \rightarrow M_2 \rightarrow Y$. Because here we put a focus on M_1 , that is, the first mediator in the causal sequence, the resulting indirect effects are referred to as type-1 mediator-specific effects. Likewise, if our focus is on the second mediator M_2 , we can combine the paths $T \rightarrow M_1 \rightarrow Y$ and $T \rightarrow M_1 \rightarrow M_2 \rightarrow Y$ to obtain type-2 mediator-specific effects. Here are the definitions of all type-1 and type-2 mediator-specific natural effects that result from this coarser decomposition:

Table 3

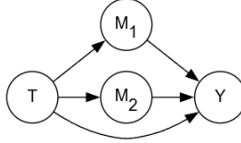
Effect	Definition
MS ¹ -NDE-00	$E(Y[1, M_1(0), M_2\{0, M_1(0)\}] - Y[0, M_1(0), M_2\{0, M_1(0)\}])$
MS ¹ -NDE-01	$E(Y[1, M_1(0), M_2\{1, M_1(1)\}] - Y[0, M_1(0), M_2\{1, M_1(1)\}])$
MS ¹ -NDE-10	$E(Y[1, M_1(1), M_2\{0, M_1(0)\}] - Y[0, M_1(1), M_2\{0, M_1(0)\}])$
MS ¹ -NDE-11	$E(Y[1, M_1(1), M_2\{1, M_1(1)\}] - Y[0, M_1(1), M_2\{1, M_1(1)\}])$
MS ¹ -NIE ₁ -00	$E(Y[0, M_1(1), M_2\{0, M_1(0)\}] - Y[0, M_1(0), M_2\{0, M_1(0)\}])$
MS ¹ -NIE ₁ -01	$E(Y[0, M_1(1), M_2\{1, M_1(1)\}] - Y[0, M_1(0), M_2\{1, M_1(1)\}])$
MS ¹ -NIE ₁ -10	$E(Y[1, M_1(1), M_2\{0, M_1(0)\}] - Y[1, M_1(0), M_2\{0, M_1(0)\}])$
MS ¹ -NIE ₁ -11	$E(Y[1, M_1(1), M_2\{1, M_1(1)\}] - Y[1, M_1(0), M_2\{1, M_1(1)\}])$
MS ¹ -NIE ₂ -00	$E(Y[0, M_1(0), M_2\{1, M_1(1)\}] - Y[0, M_1(0), M_2\{0, M_1(0)\}])$
MS ¹ -NIE ₂ -01	$E(Y[0, M_1(1), M_2\{1, M_1(1)\}] - Y[0, M_1(1), M_2\{0, M_1(0)\}])$
MS ¹ -NIE ₂ -10	$E(Y[1, M_1(0), M_2\{1, M_1(1)\}] - Y[1, M_1(0), M_2\{0, M_1(0)\}])$
MS ¹ -NIE ₂ -11	$E(Y[1, M_1(1), M_2\{1, M_1(1)\}] - Y[1, M_1(1), M_2\{0, M_1(0)\}])$
MS ² -NDE-00	$E(Y[1, M_1(0), M_2\{0, M_1(0)\}] - Y[0, M_1(0), M_2\{0, M_1(0)\}])$
MS ² -NDE-01	$E(Y[1, M_1(0), M_2\{1, M_1(0)\}] - Y[0, M_1(0), M_2\{1, M_1(0)\}])$
MS ² -NDE-10	$E(Y[1, M_1(1), M_2\{0, M_1(1)\}] - Y[0, M_1(1), M_2\{0, M_1(1)\}])$
MS ² -NDE-11	$E(Y[1, M_1(1), M_2\{1, M_1(1)\}] - Y[0, M_1(1), M_2\{1, M_1(1)\}])$
MS ² -NIE ₁ -00	$E(Y[0, M_1(1), M_2\{0, M_1(1)\}] - Y[0, M_1(0), M_2\{0, M_1(0)\}])$
MS ² -NIE ₁ -01	$E(Y[0, M_1(1), M_2\{1, M_1(1)\}] - Y[0, M_1(0), M_2\{1, M_1(0)\}])$
MS ² -NIE ₁ -10	$E(Y[1, M_1(1), M_2\{0, M_1(1)\}] - Y[1, M_1(0), M_2\{0, M_1(0)\}])$
MS ² -NIE ₁ -11	$E(Y[1, M_1(1), M_2\{1, M_1(1)\}] - Y[1, M_1(0), M_2\{1, M_1(0)\}])$
MS ² -NIE ₂ -00	$E(Y[0, M_1(0), M_2\{1, M_1(0)\}] - Y[0, M_1(0), M_2\{0, M_1(0)\}])$
MS ² -NIE ₂ -01	$E(Y[0, M_1(1), M_2\{1, M_1(1)\}] - Y[0, M_1(1), M_2\{0, M_1(1)\}])$
MS ² -NIE ₂ -10	$E(Y[1, M_1(0), M_2\{1, M_1(0)\}] - Y[1, M_1(0), M_2\{0, M_1(0)\}])$
MS ² -NIE ₂ -11	$E(Y[1, M_1(1), M_2\{1, M_1(1)\}] - Y[1, M_1(1), M_2\{0, M_1(1)\}])$

Source: Definitions from [Daniel et al. \(2015\)](#).

In the table above, the effects that start with MS¹ correspond to type-1 mediator-specific effects and those that start with MS² to type-2 mediator-specific effects. In [Example 10: Estimating mediator-specific natural effects](#), we illustrate how to estimate mediator-specific effects.

Parallel mediators

The estimand complexity is also reduced if we have a causal mediation model with two parallel mediators, which are not causally ordered. In this case, we have at most 4 estimands over 3 effects for a total of 12 estimands. The complexity is reduced because mediator potential outcomes are no longer nested. That is, instead of potential outcomes of the form $Y\{t, M_1(t'), M_2\{t'', M_1(t''')\}\}$, with a parallel rather than sequential design, we simply have $Y\{t, M_1(t'), M_2(t'')\}$. Consider the following causal diagram:



In contrast to the sequential mediation model we have discussed so far, with the parallel mediation model, we have only three path-specific effects of interest: the effect through neither M_1 nor M_2 (that is, the natural direct effect), the effect through M_1 only, and the effect through M_2 only. Using linear models with interactions, and again ignoring covariates, we can specify our system of equations as follows:

$$\begin{aligned}
 Y &= \beta_0 + \beta_1 T + \beta_2 M_1 + \beta_3 M_2 + \beta_4 T M_1 + \beta_5 T M_2 + \beta_6 M_1 M_2 + \beta_7 T M_1 M_2 + \epsilon_Y \\
 M_2 &= \gamma_0 + \gamma_1 T + \epsilon_{M_2} \\
 M_1 &= \alpha_0 + \alpha_1 T + \epsilon_{M_1}
 \end{aligned}$$

Again, we can use OLS to estimate the coefficients of the above equations and work out the form for the potential-outcome means of interest. Plugging the equations for M_1 and M_2 into the outcome equation yields

$$\begin{aligned}
 E\{Y\{t, M_1(t'), M_2(t'')\}\} &= \beta_0 + \beta_1 t + \beta_2(\alpha_0 + \alpha_1 t' + \epsilon_{M_1}) + \beta_3(\gamma_0 + \gamma_1 t'' + \epsilon_{M_2}) + \\
 &\quad \beta_4 t(\alpha_0 + \alpha_1 t' + \epsilon_{M_1}) + \beta_5 t(\gamma_0 + \gamma_1 t'' + \epsilon_{M_2}) + \\
 &\quad \beta_6(\alpha_0 + \alpha_1 t' + \epsilon_{M_1})(\gamma_0 + \gamma_1 t'' + \epsilon_{M_2}) + \\
 &\quad \beta_7 t(\alpha_0 + \alpha_1 t' + \epsilon_{M_1})(\gamma_0 + \gamma_1 t'' + \epsilon_{M_2}) + \epsilon_Y
 \end{aligned}$$

Simplifying and replacing error terms with their expectations yields

$$\begin{aligned}
 E[Y\{t, M_1(t'), M_2(t'')\}] &= \beta_0 + \beta_1 t + (\beta_2 + \beta_4 t)(\alpha_0 + \alpha_1 t') + (\beta_3 + \beta_5 t)(\gamma_0 + \gamma_1 t'') + \\
 &\quad (\beta_6 + \beta_7 t)(\alpha_0 + \alpha_1 t')(\gamma_0 + \gamma_1 t'')
 \end{aligned}$$

Suppose now that we wanted to calculate the natural direct effect NDE-00. This effect is defined as the contrast $E[Y\{1, M_1(0), M_2(0)\}] - E[Y\{0, M_1(0), M_2(0)\}]$. We can write out each potential-outcome mean as

$$\begin{aligned}
 E[Y\{1, M_1(0), M_2(0)\}] &= \beta_0 + \beta_1 t + (\beta_2 + \beta_4 t)\alpha_0 + (\beta_3 + \beta_5 t)\gamma_0 + (\beta_6 + \beta_7 t)\alpha_0\gamma_0 \\
 E[Y\{0, M_1(0), M_2(0)\}] &= \beta_0 + \beta_2\alpha_0 + \beta_3\gamma_0 + \beta_6\alpha_0\gamma_0
 \end{aligned}$$

We could now simply plug in our OLS estimates of β , α , and γ to calculate NDE-00. The following table provides an overview of all parallel mediation estimands:

Table 4

Path	Effect	Definition
Direct (through neither M_1 nor M_2)	NDE-00	$E[Y\{1, M_1(0), M_2(0)\} - Y\{0, M_1(0), M_2(0)\}]$
	NDE-10	$E[Y\{1, M_1(1), M_2(0)\} - Y\{0, M_1(1), M_2(0)\}]$
	NDE-01	$E[Y\{1, M_1(0), M_2(1)\} - Y\{0, M_1(0), M_2(1)\}]$
	NDE-11	$E[Y\{1, M_1(1), M_2(1)\} - Y\{0, M_1(1), M_2(1)\}]$
Indirect through M_1	NIE ₁ -00	$E[Y\{0, M_1(1), M_2(0)\} - Y\{0, M_1(0), M_2(0)\}]$
	NIE ₁ -10	$E[Y\{1, M_1(1), M_2(0)\} - Y\{1, M_1(0), M_2(0)\}]$
	NIE ₁ -01	$E[Y\{0, M_1(1), M_2(1)\} - Y\{0, M_1(0), M_2(1)\}]$
	NIE ₁ -11	$E[Y\{1, M_1(1), M_2(1)\} - Y\{1, M_1(0), M_2(1)\}]$
Indirect through M_2	NIE ₂ -00	$E[Y\{0, M_1(0), M_2(1)\} - Y\{0, M_1(0), M_2(0)\}]$
	NIE ₂ -10	$E[Y\{1, M_1(0), M_2(1)\} - Y\{1, M_1(0), M_2(0)\}]$
	NIE ₂ -01	$E[Y\{0, M_1(1), M_2(1)\} - Y\{0, M_1(1), M_2(0)\}]$
	NIE ₂ -11	$E[Y\{1, M_1(1), M_2(1)\} - Y\{1, M_1(1), M_2(0)\}]$

Source: Definitions from [Daniel et al. \(2015\)](#).

Because we have fewer estimands under the parallel design, we also have fewer decompositions. The number of distinct decompositions in this case is six:

Table 5

Decomposition	Total effect
1	NDE-00 + NIE ₁ -10 + NIE ₂ -11
2	NDE-00 + NIE ₁ -11 + NIE ₂ -10
3	NDE-10 + NIE ₁ -00 + NIE ₂ -11
4	NDE-01 + NIE ₁ -11 + NIE ₂ -00
5	NDE-11 + NIE ₁ -00 + NIE ₂ -01
6	NDE-11 + NIE ₁ -01 + NIE ₂ -00

Source: Decompositions from [Daniel et al. \(2015\)](#).

Notice that these decompositions also apply to the mediator-specific effects discussed in the previous section.

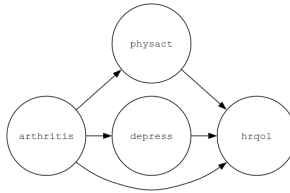
Identification assumptions

Assumptions that allow identification of causal effects are principally the same as in the single-mediator case; see *Evaluating assumptions for causal inference* in [CAUSAL] **mediate**. Perhaps most prominently, there must not be any unmeasured confounding anywhere in the system. That is, no treatment–mediator confounding, no treatment–outcome confounding, no mediator–mediator confounding, and no mediator–outcome confounding. In other words, all model equations have to be correctly specified. For a detailed discussion of identification assumptions for models with multiple mediators, see [Daniel et al. \(2015, sec. 4\)](#).

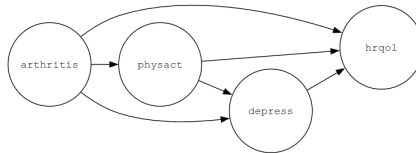
Examples

Example 1: Parallel causal mediation model

Suppose we wish to evaluate whether suffering from arthritis affects an individuals' health-related quality of life. If so, we also wish to disentangle, at least partially, the causal mechanisms through which the arthritis affects quality of life. We consider two mediator variables: physical activity and depression. We hypothesize that suffering from arthritis leads to a reduction of physical activity, which in turn leads to a reduction of health-related quality of life. We also hypothesize that arthritis leads to an increase in depression, which also leads to a reduction in health-related quality of life. Ignoring other covariates, we could depict our causal model in the following causal diagram:



We have three causal pathways from arthritis (arthritis) to health-related quality of life (hrqol): a direct path, a path through physical activity (physact), and path through depression (depress). Notice that there is no path between the two mediators. That is, our hypothesized causal model does not assume causal ordering among mediators. We refer to this design as parallel mediation. We could instead argue that there is a causal path between the two mediators. For instance, we could hypothesize the following model,



where we assume that physical activity, or lack thereof, affects depression. In this case, we have four causal pathways from arthritis to hrqol: one direct path, one indirect path through physact, one indirect path through depress, and one indirect path through both physact and depress. We refer to this design as sequential mediation.

We start with a parallel design. The following shows an excerpt of a fictional dataset where `hrqol` is a scale measuring health-related quality of life, `physact` records hours of physical activity in the last month, `depress` is a scale measuring depression, and `arthritis` is an indicator for having arthritis. The dataset also contains confounding variables `male` and `age`, which are used in later examples.

```
. use https://www.stata-press.com/data/r19/qoflife
Artificial quality-of-life data

. list hrqol depress physact arthritis male age in 1/5
```

	hrqol	depress	physact	arthritis	male	age
1.	33.58646	6.161322	4	No	Yes	34
2.	33.09237	6.321835	4	Yes	Yes	35
3.	40.68521	5.355282	9	No	Yes	30
4.	34.04324	6.025885	3	No	No	38
5.	28.16133	6.586061	0	No	Yes	48

To estimate the total treatment effect of arthritis on quality of life as well as the three path-specific effects, we use `mediate` with four sets of parentheses. The first set specifies the outcome equation, the middle two specify the mediator equations, and the last one specifies our treatment variable. Because we do not include covariates in this example, we simply specify the outcome, mediator, and treatment variables accordingly:

```
. mediate (hrqol) (depress) (physact) (arthritis)
Iteration 0: EE criterion = 1.537e-27
Iteration 1: EE criterion = 8.303e-30

Causal mediation analysis                                Number of obs = 2,000
Mediation type: Parallel
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Binary
```

hrqol	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE arthritis (Yes vs No)	-1.71165	.1496673	-11.44	0.000	-2.004993	-1.418308
NIE1 arthritis (Yes vs No)	-1.814969	.1817366	-9.99	0.000	-2.171166	-1.458772
NIE2 arthritis (Yes vs No)	-1.252368	.2507925	-4.99	0.000	-1.743912	-.7608239
TE arthritis (Yes vs No)	-4.778988	.2912224	-16.41	0.000	-5.349773	-4.208202

The output shows estimates of the total effect (TE), the natural direct effect (NDE), and both natural indirect effects (NIE1 and NIE2). The total effect is the average treatment effect that is decomposed into direct and indirect effects. It follows the usual interpretation of average treatment effects: if everyone in the population had arthritis, health-related quality of life would on average be 4.8 points lower than if no one in the population had arthritis. The indirect effect NIE1 estimates the indirect effect of arthritis

on hrqol through the physact mediator. That is, 1.8 points of the 4.8-point total expected decrease in hrqol are due to reduced physical activity. The NIE2 estimate captures the indirect effect through depress and is slightly smaller than the other indirect effect. A reduction of around 1.3 points in hrqol is due to depression. Finally, the natural direct effect (NDE) captures causal mechanisms other than the ones we observed with our two mediators. That is, from the 4.8-point average decrease in the outcome, 1.7 points are due to mechanisms other than physical activity and depression.

By default, mediate shows only the estimated treatment effects. However, we could also take a look at the parameter estimates from the individual equations by replaying the results with the aequations option:

```
. mediate, aequations
Causal mediation analysis                               Number of obs = 2,000
Mediation type: Parallel
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Binary
```

hrqol	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE arthritis (Yes vs No)	-1.71165	.1496673	-11.44	0.000	-2.004993	-1.418308
NIE1 arthritis (Yes vs No)	-1.814969	.1817366	-9.99	0.000	-2.171166	-1.458772
NIE2 arthritis (Yes vs No)	-1.252368	.2507925	-4.99	0.000	-1.743912	-.7608239
TE arthritis (Yes vs No)	-4.778988	.2912224	-16.41	0.000	-5.349773	-4.208202
hrqol arthritis Yes	-1.71165	.1496673	-11.44	0.000	-2.004993	-1.418308
physact	.8053262	.0741201	10.87	0.000	.6600534	.950599
depress	-3.117997	.5559884	-5.61	0.000	-4.207714	-2.028279
_cons	47.15107	3.59214	13.13	0.000	40.1106	54.19154
depress arthritis Yes	.4016579	.0255639	15.71	0.000	.3515536	.4517623
_cons	6.040968	.0236416	255.52	0.000	5.994632	6.087305
physact arthritis Yes	-2.253707	.2307478	-9.77	0.000	-2.705964	-1.801449
_cons	3.386249	.2171971	15.59	0.000	2.96055	3.811947

These estimates are typically not of substantive interest. However, it is always a good idea to check these and see whether the coefficient estimates appear reasonable in light of given theoretical considerations.

Example 2: Treatment–mediator interactions

The previous example was somewhat simplistic because our model did not include any interactions. We have two options related to treatment and mediator interactions in the case of parallel mediation: we can include treatment–mediator interactions as well as mediator–mediator interactions in the outcome equation. This adds flexibility to the underlying model equations at the cost of additional estimand complexity.

Continuing with the current example, we can allow the effects of physical activity and depression on quality of life to differ between treatment groups. We include these treatment–mediator interactions in the outcome equation by specifying the `tinteraction` option:

```
. mediate (hrqol) (depress) (physact) (arthritis), tinteraction
Iteration 0: EE criterion = 1.158e-25
Iteration 1: EE criterion = 3.275e-28
Causal mediation analysis                               Number of obs = 2,000
Mediation type: Parallel
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Binary
```

hrqol		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE	00	-1.863452	.1961826	-9.50	0.000	-2.247963	-1.478941
	10	-2.7116	.3099477	-8.75	0.000	-3.319086	-2.104114
	01	-.7317365	.3513065	-2.08	0.037	-1.420285	-.0431885
	11	-1.579885	.1577089	-10.02	0.000	-1.888988	-1.270781
NIE1	00	-1.582646	.1686346	-9.39	0.000	-1.913164	-1.252129
	11	-2.430795	.3236353	-7.51	0.000	-3.065108	-1.796481
NIE2	00	-1.616457	.2614895	-6.18	0.000	-2.128967	-1.103947
	11	-.4847413	.2181954	-2.22	0.026	-.9123964	-.0570863
TE arthritis (Yes vs No)		-4.778988	.2912224	-16.41	0.000	-5.349773	-4.208202

Note: Outcome equation includes treatment–mediator interactions.

The interpretation of the total effect is the same as before. However, we now have four distinct natural direct-effect estimates and two distinct indirect-effect estimates for each mediator. Differences among these estimates are due to differences in corresponding potential-outcome means, as shown in [table 4](#).

The natural direct effect labeled 00 is the estimated direct effect when both `physact` and `depress` are at their values associated with being untreated (`arthritis` = 0). Similarly, the natural direct effect labeled 10 is the estimated direct effect when `physact` is at its value associated with being treated and `depress` is at its value associated with being untreated. The other natural direct effects are interpreted similarly. Based on this specification, the natural direct effects range from -0.73 to -2.71 .

Before discussing the effect estimates in more detail, we will explore options for limiting interactions and including other interactions. With the `tinteraction` option, interactions with the treatment variable are added for both mediators. However, we can specify that only one of these interactions should be included. For example, if we want to add an interaction term only for physical activity, we specify the `tinteraction()` option with the name of the mediator in parentheses:

```
. mediate (hrqol) (depress) (physact) (arthritis), tinteraction(physact)
Iteration 0: EE criterion = 1.580e-27
Iteration 1: EE criterion = 5.149e-29
Causal mediation analysis                               Number of obs = 2,000
Mediation type: Parallel
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Binary
```

hrqol		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE	00	-1.562983	.2001687	-7.81	0.000	-1.955307	-1.17066
	10	-1.735797	.148358	-11.70	0.000	-2.026574	-1.445021
	01	-1.562983	.2001687	-7.81	0.000	-1.955307	-1.17066
	11	-1.735797	.148358	-11.70	0.000	-2.026574	-1.445021
NIE1	00	-1.799643	.175755	-10.24	0.000	-2.144116	-1.455169
	11	-1.972457	.2491519	-7.92	0.000	-2.460785	-1.484128
NIE2	00	-1.243548	.2369214	-5.25	0.000	-1.707905	-.7791903
	11	-1.243548	.2369214	-5.25	0.000	-1.707905	-.7791903
TE arthritis (Yes vs No)		-4.778988	.2912224	-16.41	0.000	-5.349773	-4.208202

Note: Outcome equation includes treatment-mediator interactions.

Now the two estimates for the NIE2 indirect effect are the same. Omitting the interaction of treatment with the `depress` mediator leads to a single NIE2 effect (reported as `NIE2-00 = NIE2-11`). For NIE1, we still have two distinct effect estimates due to the `arthritis × physact` interaction.

Example 3: Mediator–mediator interaction

In addition to treatment–mediator interactions, we can include a mediator–mediator interaction if we think that the effects of the mediators on the outcome depend on one another. To do so, we add the `minteraction` option:

```
. mediate (hrqol) (depress) (physact) (arthritis), tinteraction minteraction
Iteration 0: EE criterion = 2.526e-26
Iteration 1: EE criterion = 1.137e-28
Causal mediation analysis                               Number of obs = 2,000
Mediation type: Parallel
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Binary
```

hrqol		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE	00	-1.745654	.1767601	-9.88	0.000	-2.092097	-1.39921
	10	-1.449735	.2693148	-5.38	0.000	-1.977582	-.9218876
	01	-1.804018	.2952766	-6.11	0.000	-2.38275	-1.225287
	11	-1.638622	.139265	-11.77	0.000	-1.911576	-1.365667
NIE1	00	-3.277243	.3619214	-9.06	0.000	-3.986596	-2.56789
	10	-2.981325	.3508021	-8.50	0.000	-3.668884	-2.293765
	01	-3.389729	.3789566	-8.94	0.000	-4.132471	-2.646988
	11	-3.224333	.4005567	-8.05	0.000	-4.00941	-2.439256
NIE2	00	-.174545	.1703176	-1.02	0.305	-.5083614	.1592713
	10	-.2329098	.2139202	-1.09	0.276	-.6521856	.186366
	01	-.2870311	.1642122	-1.75	0.080	-.6088811	.0348188
	11	-.4759182	.2112993	-2.25	0.024	-.8900573	-.0617791
TE	arthritis (Yes vs No)	-5.202896	.3531395	-14.73	0.000	-5.895037	-4.510756

Note: Outcome equation includes treatment–mediator interactions and mediator–mediator interaction.

We now have four distinct estimates each for natural direct and both natural indirect effects. That is, we have twelve distinct estimates plus the total effect. This is the maximum number of distinct estimates we can have under the parallel mediation design.

The four natural direct effects are interpreted in the same way as in [example 1](#). The estimates of the natural indirect effects through `physact` are in the section labeled `NIE1`. The first one, labeled `00`, is the natural indirect effect via `physact` if no one was treated (`arthritis` = 0), and `depress` is set to the value associated with being untreated. Similarly, the value labeled `10` is the natural indirect effect via `physact` if everyone was treated (`arthritis` = 1), and `depress` is set to the value associated with being untreated. The other `NIE1` natural indirect effects are interpreted similarly. The `NIE2` section reports similar natural indirect effects through `depress` while setting values for `arthritis` and `physact`. Based on this model, indirect effects via `physact` range from -2.98 to -3.39 , and indirect effects via `depress` range from -0.17 to -0.47 .

There are six ways to decompose the total effect into sums of these natural direct and indirect effects as shown in [table 5](#).

We can check the coefficient estimates for each of the included interaction terms by replaying the results with the `aequations` option:

```
. mediate, aequations
Causal mediation analysis                               Number of obs = 2,000
Mediation type: Parallel
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Binary
```

hrqol		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE	00	-1.745654	.1767601	-9.88	0.000	-2.092097	-1.39921
	10	-1.449735	.2693148	-5.38	0.000	-1.977582	-.9218876
	01	-1.804018	.2952766	-6.11	0.000	-2.38275	-1.225287
	11	-1.638622	.139265	-11.77	0.000	-1.911576	-1.365667
NIE1	00	-3.277243	.3619214	-9.06	0.000	-3.986596	-2.56789
	10	-2.981325	.3508021	-8.50	0.000	-3.668884	-2.293765
	01	-3.389729	.3789566	-8.94	0.000	-4.132471	-2.646988
	11	-3.224333	.4005567	-8.05	0.000	-4.00941	-2.439256
NIE2	00	-.174545	.1703176	-1.02	0.305	-.5083614	.1592713
	10	-.2329098	.2139202	-1.09	0.276	-.6521856	.186366
	01	-.2870311	.1642122	-1.75	0.080	-.6088811	.0348188
	11	-.4759182	.2112993	-2.25	0.024	-.8900573	-.0617791
TE arthritis (Yes vs No)		-5.202896	.3531395	-14.73	0.000	-5.895037	-4.510756

hrqol							
arthritis							
Yes	2.526337	4.33281	0.58	0.560	-5.965815	11.01849	
physact	.7034825	.0376614	18.68	0.000	.6296675	.7772976	
depress	-.85535	.407575	-2.10	0.036	-1.654182	-.0565176	
arthritis#							
c.physact							
Yes	-1.002342	.2613421	-3.84	0.000	-1.514564	-.4901213	
arthritis#							
c.depress							
Yes	-.6335682	.6635123	-0.95	0.340	-1.934028	.666892	
c.physact#							
c.depress	.1242639	.0077999	15.93	0.000	.1089765	.1395514	
arthritis#							
c.physact#							
c.depress							
Yes	.1441887	.046566	3.10	0.002	.0529211	.2354563	
_cons	31.88665	2.635727	12.10	0.000	26.72072	37.05258	
depress							
arthritis							
Yes	.4016579	.0255639	15.71	0.000	.3515536	.4517623	
_cons	6.040968	.0236416	255.52	0.000	5.994632	6.087305	
physact							
arthritis							
Yes	-2.253707	.2307478	-9.77	0.000	-2.705964	-1.801449	
_cons	3.386249	.2171971	15.59	0.000	2.96055	3.811947	

Note: Outcome equation includes treatment-mediator interactions and mediator-mediator interaction.

Looking at the hrqol outcome equation, we see the coefficient estimates for these terms. Also notice that, with both the `tinteraction` and `minteraction` options specified, the outcome equation includes the three-way interaction of the two mediators and the treatment, which allows the mediator-mediator interaction to vary by treatment group.

Example 4: Accounting for confounding variables

For causal inference, we must evaluate the potential of confounding. Under the parallel mediation design, there are three types of confounders that we need to consider: treatment-outcome confounders, treatment-mediator confounders, and mediator-outcome confounders. For the sequential model, we must additionally consider mediator-mediator confounders. A treatment-outcome confounder, for example, is a variable that affects both the selection into treatment and the outcome. If confounders exist, we need to add them as covariates to the model to prevent biased results.

In this example, we consider age to be a treatment–outcome confounder and male to be a treatment–mediator confounder. Thus, we include age as a covariate in our outcome equation and male as a covariate in our mediator equations. We also specify the `aequations` option to inspect their coefficients:

```
. mediate (hrqol age) (depress i.male) (physact i.male) (arthritis),
> tinteraction minteraction aequations

Iteration 0:  EE criterion = 4.054e-26
Iteration 1:  EE criterion = 5.196e-30

Causal mediation analysis                                Number of obs = 2,000
Mediation type: Parallel
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Binary
```

hrqol		Robust		z	P> z	[95% conf. interval]	
Coefficient		std. err.					
NDE	00	-2.923315	.0983242	-29.73	0.000	-3.116027	-2.730603
	10	-2.328291	.131225	-17.74	0.000	-2.585487	-2.071094
	01	-3.024462	.15042	-20.11	0.000	-3.31928	-2.729645
	11	-2.410038	.0614085	-39.25	0.000	-2.530397	-2.28968
NIE1	00	-2.439151	.2229985	-10.94	0.000	-2.87622	-2.002082
	10	-1.844127	.1837655	-10.04	0.000	-2.204301	-1.483953
	01	-2.481129	.2301443	-10.78	0.000	-2.932204	-2.030054
	11	-1.866705	.1901424	-9.82	0.000	-2.239377	-1.494033
NIE2	00	-.589078	.0943036	-6.25	0.000	-.7739097	-.4042464
	10	-.6902256	.1139496	-6.06	0.000	-.9135627	-.4668885
	01	-.6310562	.0940434	-6.71	0.000	-.8153779	-.4467346
	11	-.7128041	.1142426	-6.24	0.000	-.9367155	-.4888926
TE	arthritis (Yes vs No)	-5.480245	.251458	-21.79	0.000	-5.973094	-4.987397

hrqol							
arthritis							
Yes	-1.160197	1.875236	-0.62	0.536	-4.835591	2.515197	
physact	.7273916	.0122129	59.56	0.000	.7034547	.7513284	
depress	-1.372237	.174576	-7.86	0.000	-1.7144	-1.030074	
arthritis#							
c.physact							
Yes	-.1328243	.0598191	-2.22	0.026	-.2500676	-.015581	
arthritis#							
c.depress							
Yes	-.1585676	.2871678	-0.55	0.581	-.721406	.4042709	
c.physact#							
c.depress	.0340505	.0037065	9.19	0.000	.0267859	.041315	
arthritis#							
c.physact#							
c.depress							
Yes	-.015736	.0117052	-1.34	0.179	-.0386777	.0072057	
age	-.307173	.0031765	-96.70	0.000	-.3133989	-.3009472	
_cons	50.26891	1.147244	43.82	0.000	48.02036	52.51747	
depress							
arthritis							
Yes	.4710086	.0273445	17.22	0.000	.4174144	.5246029	
male							
Yes	-.2003285	.0271024	-7.39	0.000	-.2534482	-.1472087	
_cons	6.104774	.0240335	254.01	0.000	6.057669	6.151878	
physact							
arthritis							
Yes	-2.617407	.2337319	-11.20	0.000	-3.075514	-2.159301	
male							
Yes	1.050597	.2306835	4.55	0.000	.5984654	1.502728	
_cons	3.05163	.2330003	13.10	0.000	2.594958	3.508302	

Note: Outcome equation includes treatment-mediator interactions and mediator-mediator interaction.

The age and male coefficients appear to have considerable size. Indeed, if we look at the effect estimates, we see substantial increases (in absolute value) in estimates of natural direct effect, decreases in estimates of natural indirect effects through physical activity (NIE1), and slight increases in natural indirect effects through depression (NIE2).

Example 5: Additional interactions

It is conceivable that the effect of age on quality of life may be moderated by one or both of the mediators. To include additional interactions of covariates and mediators, we simply add them by using factor-variable notation:

```
. mediate (hrqol age c.depress#c.age c.physact#c.age)
>         (depress i.male)
>         (physact i.male)
>         (arthritis), tinteraction minteraction

Iteration 0:  EE criterion = 1.762e-25
Iteration 1:  EE criterion = 8.149e-30

Causal mediation analysis                                Number of obs = 2,000
Mediation type: Parallel
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Binary
```

hrqol		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE	00	-2.93327	.0986156	-29.74	0.000	-3.126553	-2.739987
	10	-2.2664	.1321111	-17.16	0.000	-2.525333	-2.007467
	01	-3.13574	.1548724	-20.25	0.000	-3.439284	-2.832196
	11	-2.441183	.0633247	-38.55	0.000	-2.565297	-2.317068
NIE1	00	-2.559244	.2330217	-10.98	0.000	-3.015958	-2.102529
	10	-1.892373	.1842448	-10.27	0.000	-2.253486	-1.53126
	01	-2.596165	.2396821	-10.83	0.000	-3.065933	-2.126396
	11	-1.901607	.1889659	-10.06	0.000	-2.271974	-1.531241
NIE2	00	-.4994956	.0965284	-5.17	0.000	-.6886879	-.3103033
	10	-.7019653	.1137533	-6.17	0.000	-.9249177	-.4790128
	01	-.5364167	.0960709	-5.58	0.000	-.7247123	-.3481211
	11	-.7111994	.1138654	-6.25	0.000	-.9343716	-.4880273
TE arthritis (Yes vs No)		-5.536843	.2526692	-21.91	0.000	-6.032065	-5.04162

Note: Outcome equation includes treatment-mediator interactions and mediator-mediator interaction.

Here we included interactions of both mediators with age. Had we used the `aequations` option, we would see that the coefficients on these interaction terms are both close to 0. Consequently, the above results do not change in any meaningful way compared with those of the previously fitted model.

Example 6: Sequential causal mediation model

We now move on to a sequential mediation model, that is, a causal mediation model with two causally ordered mediators. In this case, the total causal effect is decomposed into four path-specific effects: the natural direct effect that goes through neither of the mediators (NDE), a natural indirect effect through only the first mediator (NIE1), a natural indirect effect through only the second mediator (NIE2), and a natural indirect effect through both of the mediators (NIE12).

We start with a model without interactions and without covariates, in which case we will have only a single estimand per effect. The syntax of the `mediate` command is similar to that for the parallel mediation model, but we will have to be cognizant about the causal order between mediators. The specification follows the causal sequence from right to left, where the causal sequence is $Y \leftarrow M_2 \leftarrow M_1 \leftarrow T$. Thus, if we distinguish between the first and second mediator such that the first mediator is a predictor of the second, then we specify the first mediator right next to the treatment and the second mediator next to the outcome.

In the following example, we hypothesize that the causal sequence is arthritis affects physical activity, which in turn affects depressive symptoms. Thus, we specify `physact` as our first mediator right next to the treatment `arthritis`, and we specify `depress` as our second mediator next to the outcome `hrqol`. We also specify the `sequential` option to fit a sequential model:

```
. mediate (hrqol) (depress) (physact) (arthritis), sequential
Iteration 0:  EE criterion = 1.539e-27
Iteration 1:  EE criterion = 5.798e-29
Causal mediation analysis                               Number of obs = 2,000
Mediation type: Sequential
Mediator 1:      physact
Mediator 2:      depress
Treatment type:  Binary
```

hrqol	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE arthritis (Yes vs No)	-1.71165	.1496673	-11.44	0.000	-2.004993	-1.418308
NIE1 arthritis (Yes vs No)	-1.814969	.1817366	-9.99	0.000	-2.171166	-1.458772
NIE2 arthritis (Yes vs No)	-.5203296	.1197562	-4.34	0.000	-.7550474	-.2856118
NIE12 arthritis (Yes vs No)	-.7320385	.1400059	-5.23	0.000	-1.006445	-.457632
TE arthritis (Yes vs No)	-4.778988	.2912224	-16.41	0.000	-5.349773	-4.208202

The output provides estimates of all four path-specific effects as well as the total effect. The total effect of -4.78 decomposes into the natural direct effect of -1.71 , an indirect effect through `physact` alone of -1.81 , an indirect effect through `depress` alone of -0.52 , and an indirect effect through both `physact` and `depress` of -0.73 . The interpretation of these effects is the same as before.

Example 7: Sequential model with treatment–mediator interactions

We now include interactions of both mediator variables with the treatment. While this is often a reasonable thing to do, it comes at the expense of increased complexity with respect to the number of estimands. To fit the model including these interactions, we specify the `tinteraction` option:

```
. mediate (hrqol) (depress) (physact) (arthritis), sequential
> tinteraction

Iteration 0: EE criterion = 1.158e-25
Iteration 1: EE criterion = 5.666e-28

Causal mediation analysis                                Number of obs = 2,000
Mediation type: Sequential
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Binary
```

hrqol		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE							
	000	-1.863452	.1961826	-9.50	0.000	-2.247963	-1.478941
	100	-2.7116	.3099477	-8.75	0.000	-3.319086	-2.104114
	010	-1.393251	.2242743	-6.21	0.000	-1.83282	-.9536811
	001	-1.201938	.2449106	-4.91	0.000	-1.681954	-.7219218
	110	-2.241399	.203996	-10.99	0.000	-2.641224	-1.841574
	101	-2.050086	.1664379	-12.32	0.000	-2.376298	-1.723873
	011	-.7317365	.3513065	-2.08	0.037	-1.420285	-.0431885
	111	-1.579885	.1577089	-10.02	0.000	-1.888988	-1.270781
NIE1							
	000	-1.582646	.1686346	-9.39	0.000	-1.913164	-1.252129
	111	-2.430795	.3236353	-7.51	0.000	-3.065108	-1.796481
NIE2							
	000	-.6715998	.1293398	-5.19	0.000	-.9252152	-.4179843
	111	-.2013986	.0907096	-2.22	0.026	-.3791862	-.0236111
NIE12							
	000	-.9448568	.1463968	-6.45	0.000	-1.231789	-.6579242
	111	-.2833427	.1292444	-2.19	0.028	-.536657	-.0300283
TE arthritis (Yes vs No)							
		-4.778988	.2912224	-16.41	0.000	-5.349773	-4.208202

Note: Outcome equation includes treatment–mediator interactions.

We now estimate eight natural direct effects and two distinct indirect effects each for NIE1, NIE2, and NIE12. For each of the indirect effects, we have one labeled 000, which is the indirect-effect estimate evaluated as if no one in the population has arthritis, and the one labeled 111 is the indirect effect evaluated as if everyone in the population has arthritis. For example, the estimated natural direct effects via only `physact` are $NIE1-000 = -1.58$ and $NIE1-111 = -2.43$. Here the effects $NIE1-000$ can be interpreted analogously to a pure natural indirect effect, with everything set to values corresponding to no treatment. Similarly, $NIE1-000$ is analogous to a total natural indirect effect with everything set to values corresponding to treatment. If we had a good theoretical or even practical argument for focusing on either pure or total natural effects, we could pick estimates of interest accordingly.

Example 8: Including the mediator–mediator interaction

Including the mediator–mediator interaction allows for additional flexibility in that the effect of one mediator on the outcome depends on the other mediator and vice versa. However, adding the mediator–mediator interaction increases complexity further. To include the interaction, we add the `minteraction` option:

```
. mediate (hrqol) (depress) (physact) (arthritis), sequential
> tinteraction minteraction

Iteration 0:  EE criterion = 6.373e-21
Iteration 1:  EE criterion = 2.803e-28

Causal mediation analysis                                Number of obs = 2,000
Mediation type: Sequential
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Binary
```

hrqol		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE	000	−2.138797	.1763549	−12.13	0.000	−2.484447	−1.793148
	100	−1.842878	.2916825	−6.32	0.000	−2.414566	−1.271191
	010	−2.163046	.178523	−12.12	0.000	−2.512945	−1.813148
	001	−2.172913	.1984872	−10.95	0.000	−2.561941	−1.783885
	110	−1.921357	.2153885	−8.92	0.000	−2.34351	−1.499203
	101	−1.953287	.194975	−10.02	0.000	−2.335431	−1.571144
	011	−2.197162	.2759688	−7.96	0.000	−2.738051	−1.656273
	111	−2.031766	.1850413	−10.98	0.000	−2.39444	−1.669091
NIE1	000	−3.277243	.3619214	−9.06	0.000	−3.986596	−2.56789
	100	−2.981325	.3508021	−8.50	0.000	−3.668884	−2.293765
	010	−3.323979	.3679973	−9.03	0.000	−4.04524	−2.602717
	001	−3.342994	.3728914	−8.97	0.000	−4.073848	−2.61214
	110	−3.082289	.3685075	−8.36	0.000	−3.80455	−2.360028
	101	−3.123369	.3820944	−8.17	0.000	−3.87226	−2.374478
	011	−3.389729	.3789566	−8.94	0.000	−4.132471	−2.646988
	111	−3.224333	.4005567	−8.05	0.000	−4.00941	−2.439256
NIE2	000	−.0725194	.0703347	−1.03	0.303	−.2103728	.065334
	101	−.0967685	.0893702	−1.08	0.279	−.2719309	.0783938
	011	−.1192547	.0672674	−1.77	0.076	−.2510964	.012587
	111	−.1977329	.0879458	−2.25	0.025	−.3701034	−.0253623
NIE12	000	−.1020257	.100273	−1.02	0.309	−.2985572	.0945059
	101	−.1361412	.1249654	−1.09	0.276	−.3810689	.1087865
	011	−.1677764	.0977465	−1.72	0.086	−.3593561	.0238032
	111	−.2781854	.125105	−2.22	0.026	−.5233866	−.0329842
TE	arthritis (Yes vs No)	−5.59604	.4280766	−13.07	0.000	−6.435055	−4.757025

Note: Outcome equation includes treatment–mediator interactions and mediator–mediator interaction.

Notice that, when `tinteraction` is specified to include treatment–mediator interactions, adding `minteraction` will also add the three-way interaction of both mediators and the treatment. Therefore, we actually add two interaction terms in this case. We see that the indirect effects related to the second mediator `depress` (NIE2 and NIE12) become quite small and fairly close to zero. Perhaps the somewhat larger results that we obtained before (for example, NIE2 equals -0.52 in the model without interactions) were rather spurious and the result of nonflexible specifications of the structural equations.

Example 9: Complete set of interactions

Finally, we add an interaction of the treatment and first mediator to the equation of the second mediator by specifying the `meqtinteraction` option. We now estimate all 32 defined-effect estimands that are shown in [table 1](#) (plus the total effect, of course).

```
. mediate (hrqol) (depress) (physact) (arthritis), sequential
> tinteraction minteraction meqtinteraction

Iteration 0: EE criterion = 6.373e-21
Iteration 1: EE criterion = 5.935e-28

Causal mediation analysis                                Number of obs = 2,000

Mediation type: Sequential
Mediator 1:      physact
Mediator 2:      depress
Treatment type:  Binary
```

hrqol		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE	000	-2.137034	.1755708	-12.17	0.000	-2.481146	-1.792921
	100	-1.841115	.2914441	-6.32	0.000	-2.412335	-1.269895
	010	-2.175025	.1867005	-11.65	0.000	-2.540951	-1.809098
	001	-2.170996	.1970605	-11.02	0.000	-2.557228	-1.784765
	110	-1.93079	.2240868	-8.62	0.000	-2.369992	-1.491587
	101	-1.951029	.1944125	-10.04	0.000	-2.33207	-1.569987
	011	-2.210278	.284549	-7.77	0.000	-2.767984	-1.652573
	111	-2.044882	.1971614	-10.37	0.000	-2.431311	-1.658453
NIE1	000	-3.277243	.3619214	-9.06	0.000	-3.986596	-2.56789
	100	-2.981325	.3508021	-8.50	0.000	-3.668884	-2.293765
	010	-3.321785	.3659441	-9.08	0.000	-4.039022	-2.604548
	001	-3.342699	.3726794	-8.97	0.000	-4.073137	-2.612261
	110	-3.07755	.3644594	-8.44	0.000	-3.791877	-2.363223
	101	-3.122732	.3817002	-8.18	0.000	-3.87085	-2.374613
	011	-3.389729	.3789566	-8.94	0.000	-4.132471	-2.646988
	111	-3.224333	.4005567	-8.05	0.000	-4.00941	-2.439256
NIE2	000	-.0819393	.0646655	-1.27	0.205	-.2086815	.0448028
	100	-.1199303	.0962299	-1.25	0.213	-.3085375	.0686768
	010	-.126481	.0623618	-2.03	0.043	-.2487078	-.0042542
	001	-.0858009	.068813	-1.25	0.212	-.2206718	.04907
	110	-.2161557	.0956439	-2.26	0.024	-.4036142	-.0286971
	101	-.1250831	.104926	-1.19	0.233	-.3307343	.0805681
	011	-.1328312	.0670555	-1.98	0.048	-.2642576	-.0014048
	111	-.2266847	.1099928	-2.06	0.039	-.4422665	-.0111028

NIE12							
	000	-.101568	.100049	-1.02	0.310	-.2976605	.0945245
	100	-.1355305	.1244552	-1.09	0.276	-.3794582	.1083972
	010	-.1670238	.0977036	-1.71	0.087	-.3585193	.0244717
	001	-.1054296	.1019476	-1.03	0.301	-.3052432	.0943841
	110	-.2769374	.1246697	-2.22	0.026	-.5212856	-.0325893
	101	-.1406833	.1288855	-1.09	0.275	-.3932943	.1119277
	011	-.1733739	.0984035	-1.76	0.078	-.3662413	.0194934
	111	-.2874664	.1298441	-2.21	0.027	-.5419561	-.0329768
TE							
arthritis							
(Yes vs No)		-5.62198	.452873	-12.41	0.000	-6.509595	-4.734365

Note: Outcome equation includes treatment-mediator interactions and mediator-mediator interaction, and mediator 2 equation includes treatment-mediator interaction.

The estimates of NIE_2 and NIE_{12} remain close to zero, and the NDE and NIE_1 estimates do not seem to change in a meaningful way either. Here we could conclude that the effect of arthritis on health-related quality of life is largely due to its effect on physical activity (NIE_1) as well as other causal mechanisms (NDE) but not so much due to the pathway through depression.

There are 24 ways we can decompose the total effect into direct and indirect effects reported in the output above; see [table 2](#).

Example 10: Estimating mediator-specific natural effects

A potential way of reducing the estimand complexity is to obtain a coarser decomposition. This can be done by focusing on the indirect effect of one mediator variable while combining the effect through the other mediator with the indirect effect that runs through both mediators. We refer to the resulting effects as mediator-specific effects.

To estimate mediator-specific effects, we first determine which mediator we wish to focus on. Focusing on the first mediator in the causal sequence, `physact` in this case, results in type-1 mediator-specific effects. Choosing the second mediator (`depress`) yields type-2 mediator-specific effects. To estimate type-1 effects, we use the `mseffects(physact)` option:

```
. mediate (hrqol age) (depress i.male) (physact i.male) (arthritis),
> sequential mseffects(physact)
Iteration 0: EE criterion = 1.339e-27
Iteration 1: EE criterion = 2.898e-29
Causal mediation analysis                               Number of obs = 2,000
Mediation type: Sequential
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Binary
MS effects:      Type 1
```

hrqol	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
MS-NDE arthritis (Yes vs No)	-2.48409	.0594904	-41.76	0.000	-2.600689	-2.367491
MS-NIE1 arthritis (Yes vs No)	-1.938463	.1639594	-11.82	0.000	-2.259817	-1.617108
MS-NIE2 arthritis (Yes vs No)	-.9174464	.1036528	-8.85	0.000	-1.120602	-.7142907
TE arthritis (Yes vs No)	-5.339999	.2326946	-22.95	0.000	-5.796072	-4.883926

We no longer have an effect that captures the pathway through both mediators because this is subsumed in the MS-NIE₂ effect now. Here we did not include any interactions, which is why we have only a single estimate per effect. However, even if we had included all interactions as we did in the previous example, we would have to deal with a total of only 12 estimands rather than 32, as described in the top half of [table 3](#).

Example 11: Adding interactions of covariates with mediators and treatment

In some of the previous examples, we added treatment–mediator and mediator–mediator interactions to allow for model flexibility. However, if our model contains covariates, we can make it even more flexible by adding interactions of the covariates with both the treatment and the mediators. To do so, we use factor-variable notation and add the desired interactions to the equations of interest:

```
. mediate (hrqol age i.male i.male#c.(depress physact) i.male#i.arthritis)
>         (depress i.male i.male#c.(physact i.arthritis))
>         (physact i.male i.male#i.arthritis)
>         (arthritis), sequential tinteraction minteraction meqtinteraction

Iteration 0:  EE criterion = 7.489e-21
Iteration 1:  EE criterion = 1.034e-28

Causal mediation analysis                                     Number of obs = 2,000
Mediation type: Sequential
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Binary
```

hrqol		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE	000	-2.861964	.0939964	-30.45	0.000	-3.046194	-2.677735
	100	-2.217973	.1583486	-14.01	0.000	-2.52833	-1.907615
	010	-2.949311	.0976019	-30.22	0.000	-3.140608	-2.758015
	001	-2.972083	.1109218	-26.79	0.000	-3.189485	-2.75468
	110	-2.297695	.0963058	-23.86	0.000	-2.486451	-2.108939
	101	-2.317705	.0815672	-28.41	0.000	-2.477574	-2.157837
	011	-3.047135	.1569048	-19.42	0.000	-3.354663	-2.739607
	111	-2.386283	.0686456	-34.76	0.000	-2.520826	-2.25174
NIE1	000	-2.465474	.2288064	-10.78	0.000	-2.913926	-2.017022
	100	-1.821483	.1850826	-9.84	0.000	-2.184238	-1.458728
	010	-2.484133	.2312894	-10.74	0.000	-2.937452	-2.030815
	001	-2.490889	.233858	-10.65	0.000	-2.949243	-2.032536
	110	-1.832517	.1875503	-9.77	0.000	-2.200109	-1.464925
	101	-1.836512	.1893983	-9.70	0.000	-2.207726	-1.465298
	011	-2.506735	.2357752	-10.63	0.000	-2.968846	-2.044624
	111	-1.845883	.191579	-9.64	0.000	-2.22137	-1.470395
NIE2	000	-.2491834	.045908	-5.43	0.000	-.3391614	-.1592055
	100	-.3365307	.0588508	-5.72	0.000	-.4518762	-.2211853
	010	-.2678428	.0454731	-5.89	0.000	-.3569683	-.1787172
	001	-.2119778	.0394603	-5.37	0.000	-.2893185	-.1346371
	110	-.347565	.0585846	-5.93	0.000	-.4623886	-.2327414
	101	-.2870304	.0513105	-5.59	0.000	-.3875972	-.1864636
	011	-.2278234	.0390462	-5.83	0.000	-.3043525	-.1512942
	111	-.2964007	.051029	-5.81	0.000	-.3964157	-.1963858

NIE12							
	000	-.328122	.0694224	-4.73	0.000	-.4641874	-.1920565
	100	-.4382404	.0793344	-5.52	0.000	-.5937329	-.2827478
	010	-.3535372	.0704798	-5.02	0.000	-.4916751	-.2153993
	001	-.2909163	.0618135	-4.71	0.000	-.4120686	-.1697641
	110	-.4532698	.0803072	-5.64	0.000	-.6106689	-.2958706
	101	-.38874	.0704634	-5.52	0.000	-.5268457	-.2506343
	011	-.3135178	.0627361	-5.00	0.000	-.4364783	-.1905573
	111	-.4021055	.0713222	-5.64	0.000	-.5418943	-.2623166
TE							
arthritis							
(Yes vs No)		-5.433117	.2491783	-21.80	0.000	-5.921498	-4.944737

Note: Outcome equation includes treatment-mediator interactions and mediator-mediator interaction, and mediator 2 equation includes treatment-mediator interaction.

Here we include covariates age and male in our outcome equation and covariate male in both mediator equations. We also added the interaction term male $\times$ arthritis to the equation of the first mediator (physact), added male $\times$ arthritis and male $\times$ physact to the equation of the second mediator (depress), and added terms male $\times$ arthritis, male $\times$ depress, and male $\times$ physact to the outcome equation.

Example 12: Sensitivity analysis

In *Modeling and estimation*, we discussed performing sensitivity analysis on the unobserved correlation parameter related to the conditional covariance of $M_1(0)$ and $M_1(1)$. We can use the `estat smsensitivity` postestimation command to perform the analysis. By default, `mediate` assumes that the error terms related to $M_1(0)$ and $M_1(1)$ are identical and thus perfectly correlated (that is, $\rho = 1$). By varying the value of ρ , we can find out how sensitive our results are to the assumed correlation. By default, `estat smsensitivity` evaluates the effect estimates at values of $\rho = 0$ and $\rho = 0.5$ and compares them to the estimates we obtained from `mediate` using $\rho = 1$. We could evaluate other values of ρ by specifying them using the `rho()` option. Here, however, we use the default values:

```
. estat smsensitivity
Sequential mediation sensitivity analysis
Number of obs = 2,000
Correlation assumed in fitted model: rho = 1
Evaluating effects at rho = 0.0 ...
                        rho = 0.5 ...
```

	rho=0.00	rho=0.50	rho=1.00
NDE			
000	-2.900 (0.108)	-2.881 (0.100)	-2.862 (0.094)
100	-2.256 (0.156)	-2.237 (0.156)	-2.218 (0.158)
010	-2.983 (0.113)	-2.966 (0.104)	-2.949 (0.098)
001	-3.010 (0.126)	-2.991 (0.118)	-2.972 (0.111)
110	-2.331 (0.094)	-2.315 (0.094)	-2.298 (0.096)
101	-2.356 (0.078)	-2.337 (0.078)	-2.318 (0.082)
011	-3.081 (0.169)	-3.064 (0.162)	-3.047 (0.157)
111	-2.420 (0.066)	-2.403 (0.066)	-2.386 (0.069)
NIE1			
000	-2.465 (0.229)	-2.465 (0.229)	-2.465 (0.229)
100	-1.821 (0.185)	-1.821 (0.185)	-1.821 (0.185)
010	-2.484 (0.231)	-2.484 (0.231)	-2.484 (0.231)
001	-2.491 (0.234)	-2.491 (0.234)	-2.491 (0.234)
110	-1.833 (0.188)	-1.833 (0.188)	-1.833 (0.188)
101	-1.837 (0.189)	-1.837 (0.189)	-1.837 (0.189)
011	-2.507 (0.236)	-2.507 (0.236)	-2.507 (0.236)
111	-1.846 (0.192)	-1.846 (0.192)	-1.846 (0.192)

NIE2				
000	-0.260	-0.254	-0.249	
	(0.046)	(0.046)	(0.046)	
100	-0.343	-0.340	-0.337	
	(0.059)	(0.059)	(0.059)	
010	-0.278	-0.273	-0.268	
	(0.046)	(0.046)	(0.045)	
001	-0.222	-0.217	-0.212	
	(0.039)	(0.039)	(0.039)	
110	-0.354	-0.351	-0.348	
	(0.059)	(0.059)	(0.059)	
101	-0.293	-0.290	-0.287	
	(0.051)	(0.051)	(0.051)	
011	-0.238	-0.233	-0.228	
	(0.039)	(0.039)	(0.039)	
111	-0.303	-0.300	-0.296	
	(0.051)	(0.051)	(0.051)	
NIE12				
000	-0.328	-0.328	-0.328	
	(0.069)	(0.069)	(0.069)	
100	-0.438	-0.438	-0.438	
	(0.079)	(0.079)	(0.079)	
010	-0.354	-0.354	-0.354	
	(0.070)	(0.070)	(0.070)	
001	-0.291	-0.291	-0.291	
	(0.062)	(0.062)	(0.062)	
110	-0.453	-0.453	-0.453	
	(0.080)	(0.080)	(0.080)	
101	-0.389	-0.389	-0.389	
	(0.070)	(0.070)	(0.070)	
011	-0.314	-0.314	-0.314	
	(0.063)	(0.063)	(0.063)	
111	-0.402	-0.402	-0.402	
	(0.071)	(0.071)	(0.071)	
TE				
Arthritis				
(Yes vs No)	-5.477	-5.455	-5.433	
	(0.254)	(0.251)	(0.249)	

The results do not change in a meaningful way over the different values of ρ . The assumption of perfect correlation is inconsequential in this case.

See [\[CAUSAL\] mediate multiple postestimation](#) for further information about `estat smsensitivity`.

Example 13: Estimating controlled direct effects

Controlled direct effects (CDEs) are different from the other estimands we have discussed so far. Here, rather than having potential outcomes of the form $Y_i[t, M_{1i}(t'), M_{2i}\{t'', M_{1i}(t''')\}]$, we have potential outcomes $Y_i(t|M_{1i} = m_{1i}, M_{2i} = m_{2i})$. That is, we have potential outcomes for each treatment level that are evaluated at set values of the mediators. Thus, CDEs use only the results of the outcome equation. Assuming a binary treatment, the CDE for values m_1 and m_2 of the two mediators is $\text{CDE}(\mathbf{m}) = Y_i(1|M_{1i} = m_1, M_{2i} = m_2) - Y_i(0|M_{1i} = m_1, M_{2i} = m_2)$. CDEs can be estimated using the `estat cde postestimation` command.

To demonstrate, we begin by fitting the following sequential mediation model:

```
. mediate (hrqol) (depress) (physact) (arthritis), sequential
> tinteraction minteraction

Iteration 0: EE criterion = 6.373e-21
Iteration 1: EE criterion = 2.803e-28

Causal mediation analysis                                Number of obs = 2,000

Mediation type: Sequential
Mediator 1:      physact
Mediator 2:      depress
Treatment type:  Binary
```

hrqol		Robust		z	P> z	[95% conf. interval]	
		Coefficient	std. err.				
NDE	000	-2.138797	.1763549	-12.13	0.000	-2.484447	-1.793148
	100	-1.842878	.2916825	-6.32	0.000	-2.414566	-1.271191
	010	-2.163046	.178523	-12.12	0.000	-2.512945	-1.813148
	001	-2.172913	.1984872	-10.95	0.000	-2.561941	-1.783885
	110	-1.921357	.2153885	-8.92	0.000	-2.34351	-1.499203
	101	-1.953287	.194975	-10.02	0.000	-2.335431	-1.571144
	011	-2.197162	.2759688	-7.96	0.000	-2.738051	-1.656273
	111	-2.031766	.1850413	-10.98	0.000	-2.39444	-1.669091
NIE1	000	-3.277243	.3619214	-9.06	0.000	-3.986596	-2.56789
	100	-2.981325	.3508021	-8.50	0.000	-3.668884	-2.293765
	010	-3.323979	.3679973	-9.03	0.000	-4.04524	-2.602717
	001	-3.342994	.3728914	-8.97	0.000	-4.073848	-2.61214
	110	-3.082289	.3685075	-8.36	0.000	-3.80455	-2.360028
	101	-3.123369	.3820944	-8.17	0.000	-3.87226	-2.374478
	011	-3.389729	.3789566	-8.94	0.000	-4.132471	-2.646988
	111	-3.224333	.4005567	-8.05	0.000	-4.00941	-2.439256
NIE2	000	-.0725194	.0703347	-1.03	0.303	-.2103728	.065334
	101	-.0967685	.0893702	-1.08	0.279	-.2719309	.0783938
	011	-.1192547	.0672674	-1.77	0.076	-.2510964	.012587
	111	-.1977329	.0879458	-2.25	0.025	-.3701034	-.0253623
NIE12	000	-.1020257	.100273	-1.02	0.309	-.2985572	.0945059
	101	-.1361412	.1249654	-1.09	0.276	-.3810689	.1087865
	011	-.1677764	.0977465	-1.72	0.086	-.3593561	.0238032
	111	-.2781854	.125105	-2.22	0.026	-.5233866	-.0329842
TE	arthritis (Yes vs No)	-5.59604	.4280766	-13.07	0.000	-6.435055	-4.757025

Note: Outcome equation includes treatment-mediator interactions and mediator-mediator interaction.

We now use `estat cde` to estimate CDEs. We specify at least one value of interest for each mediator in the `mvalue()` option. Here we evaluate the CDE at `physact = 0` and `depress = 7`:

```
. estat cde, mvalue(physact=0 depress=7)
Controlled direct effect                                Number of obs = 2,000
Mediator variables: physact depress
Mediator values:
  physact = 0
  depress = 7
```

	CDE	Delta-method std. err.	z	P> z	[95% conf. interval]	
arthritis (Yes vs No)	-1.908641	.3501075	-5.45	0.000	-2.594839	-1.222443

The estimated effect is -1.9 , which means that, if everyone in the population had a physical activity score of 0 and a depression score of 7, we would expect a difference in quality of life of 1.9 points as a direct consequence of arthritis.

Rather than evaluating CDEs at single points, we could also provide lists of evaluation points. For example, we might be interested in the CDE for which we fix one mediator at a single value but fix the other mediator at a range of values. In the following example, we fix physact at 0 and let the values for depress range from 0 to 10:

```
. estat cde, mvalue(physact=0 depress=(0(1)10))
Controlled direct effect                                Number of obs = 2,000
Mediator variables: physact depress
Mediator values:
  1._at: physact = 0
           depress = 0
  2._at: physact = 0
           depress = 1
  3._at: physact = 0
           depress = 2
  4._at: physact = 0
           depress = 3
  5._at: physact = 0
           depress = 4
  6._at: physact = 0
           depress = 5
  7._at: physact = 0
           depress = 6
  8._at: physact = 0
           depress = 7
  9._at: physact = 0
           depress = 8
 10._at: physact = 0
           depress = 9
 11._at: physact = 0
           depress = 10
```

	CDE	Delta-method std. err.	z	P> z	[95% conf. interval]	
arthritis@_at						
(Yes vs No) 1	2.526337	4.33281	0.58	0.560	-5.965815	11.01849
(Yes vs No) 2	1.892769	3.669792	0.52	0.606	-5.299891	9.085429
(Yes vs No) 3	1.259201	3.006991	0.42	0.675	-4.634394	7.152795
(Yes vs No) 4	.6256324	2.344593	0.27	0.790	-3.969686	5.220951
(Yes vs No) 5	-.0079359	1.683074	-0.00	0.996	-3.306699	3.290828
(Yes vs No) 6	-.6415041	1.024135	-0.63	0.531	-2.648772	1.365764
(Yes vs No) 7	-1.275072	.3813991	-3.34	0.001	-2.022601	-.5275439
(Yes vs No) 8	-1.908641	.3501075	-5.45	0.000	-2.594839	-1.222443
(Yes vs No) 9	-2.542209	.9900416	-2.57	0.010	-4.482655	-.6017629
(Yes vs No) 10	-3.175777	1.648723	-1.93	0.054	-6.407214	.0556602
(Yes vs No) 11	-3.809345	2.310171	-1.65	0.099	-8.337196	.7185059

We can see that our specification expands to pairs of values at which the CDE is evaluated. That is, the CDE is evaluated at pairs (physact = 0, depress = 0), (physact = 0, depress = 1), (physact = 0, depress = 2), and so on.

See [CAUSAL] [mediate multiple postestimation](#) for further information about estat cde.

Example 14: Continuous treatment

Instead of a binary treatment, we can use a continuous treatment variable. With continuous treatments, we have to specify evaluation points of the treatment variable at which to evaluate the effects. The potential-outcome mean notation extends straightforwardly to this case where we simply replace the zeros and ones from the binary case with the evaluation points of interest from the continuous treatment variable. With continuous treatments, we have to specify two values, one to be taken as the treatment level and another to be taken as the control level.

Here we are using variable arth_cont as our treatment variable, which can be thought of as a measure of severity of arthritis. We wish to estimate natural direct and indirect effects related to contrasts in potential outcomes that are defined by the set control and treatment levels of the continuous variable. Here we set 0 as our control level (that is, no arthritis) and 10 as our level of treatment (that is, moderately severe arthritis) using the continuous() option:

```
. mediate (hrqol) (depress) (physact) (arth_cont, continuous(0 10)), sequential
Iteration 0: EE criterion = 1.219e-27
Iteration 1: EE criterion = 2.897e-29
Causal mediation analysis                                     Number of obs = 2,000
Mediation type: Sequential
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Continuous
Continuous treatment levels:
  0: arth_cont = 0 (control)
  1: arth_cont = 10
```

hrqol	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE arth_cont (1 vs 0)	-1.350019	.3748461	-3.60	0.000	-2.084703	-.6153338
NIE1 arth_cont (1 vs 0)	-1.091863	.3515679	-3.11	0.002	-1.780923	-.4028024
NIE2 arth_cont (1 vs 0)	-.5607094	.1522004	-3.68	0.000	-.8590167	-.2624022
NIE12 arth_cont (1 vs 0)	-.7201396	.2306259	-3.12	0.002	-1.172158	-.2681211
TE arth_cont (1 vs 0)	-3.72273	.9267735	-4.02	0.000	-5.539173	-1.906288

The results are interpreted the same way as in the binary treatment case. For example, the total effect here is around -3.7 . If everyone in the population had `arth_cont` = 0 versus everyone had `arth_cont` = 10, we would expect an average decrease in health-related quality of life by 3.7 points. This total effect decomposes into direct and indirect effects that are interpreted similarly.

Stored results

`mediate` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(mstype)</code>	type of mediator-specific effects
<code>e(tinteraction)</code>	1 if treatment–mediator interactions included, 0 otherwise
<code>e(minteraction)</code>	1 if mediator–mediator interaction included, 0 otherwise
<code>e(meqtinteraction)</code>	1 if treatment–mediator interaction in mediator equation, 0 otherwise
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>mediate</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(mvar1)</code>	name of first mediator variable
<code>e(mvar2)</code>	name of second mediator variable
<code>e(tvar)</code>	name of treatment variable
<code>e(med_type)</code>	parallel or sequential
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(tvartype)</code>	binary or continuous
<code>e(control)</code>	control level
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals
-----------------------	--

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

Two-mediator case
Sequential mediation sensitivity analysis
CDEs

Two-mediator case

The two-mediator version of `mediate` fits causal mediation models with two mediators that can be causally ordered and provides the finest possible decomposition into all path-specific effects. Coarser decompositions are also available in form of mediator-specific effects. We follow the derivations and definitions discussed in [Daniel et al. \(2015\)](#).

With the potential-outcomes framework, the estimated treatment effects are the result of contrasts between potential-outcome means. Without loss of generality, let T_i be a binary treatment, $t \in \{0, 1\}$, for observations $i = 1, \dots, N$, and let Y_i be the outcome, $M_{i,1}$ the first mediator variable, and $M_{i,2}$ the second mediator variable. The potential-outcome means for models with causally ordered mediators take on the form

$$E(Y_i[t, M_{i,1}(t'), M_{i,2}\{t'', M_{i,1}(t''')\}])$$

The potential-outcome means are defined as the result of an integral of the conditional expectation of the outcome with respect to conditional distributions of mediators and covariates. This integral can be written as

$$\begin{aligned} E(Y[t, M_1(t'), M_2\{t'', M_1(t''')\}]) &= \int_x \int_{m_1} \int_{m'_1} \int_{m_2} E(Y \mid X = x, T = t, M_1 = m_1, M_2 = m_2) \\ &\quad \times p(m_2 \mid x, t'', m'_1) \times p(m'_1 \mid x, m_1, t'') \times p(m_1 \mid x, t') \\ &\quad \times p(x) dm_2 dm'_1 dm_1 dx \end{aligned}$$

where $p(\cdot)$ denotes conditional densities. For models without causally ordered mediators, the potential-outcome means simplify to

$$E[Y_i\{t, M_{i,1}(t'), M_{i,2}(t'')\}]$$

Consequently, the integral from above simplifies to

$$\begin{aligned} E[Y\{t, M_1(t'), M_2(t'')\}] &= \int_x \int_{m_1} \int_{m_2} E[Y \mid X = x, T = t, M_1 = m_1, M_2 = m_2] \\ &\quad \times p(m_2 \mid x, t'') \times p(m_1 \mid x, t') \times p(x) dm_2 dm_1 dx \end{aligned}$$

`mediate` uses a system of linear equations to approximate these integrals and estimate the potential-outcome means. For example, the general case of causal mediation with two causally ordered mediators and all possible treatment–mediator and mediator–mediator interactions (ignoring covariates here for simplicity) yields the following system of equations:

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 T_i + \beta_2 M_{i,1} + \beta_3 M_{i,2} + \beta_4 T_i M_{i,1} + \beta_5 T_i M_{i,2} + \beta_6 M_{i,1} M_{i,2} + \\ &\quad \beta_7 T_i M_{i,1} M_{i,2} + \epsilon_{Y_i} \\ M_{i,2} &= \gamma_0 + \gamma_1 T_i + \gamma_2 M_{i,1} + \gamma_3 T_i M_{i,1} + \epsilon_{M_{i,2}} \\ M_{i,1} &= \alpha_0 + \alpha_1 T_i + \epsilon_{M_{i,1}} \end{aligned}$$

Error terms ϵ_Y , ϵ_{M_1} , and ϵ_{M_2} are assumed independent with zero mean and variances σ_Y^2 , $\sigma_{M_1}^2$, and $\sigma_{M_2}^2$. Plugging the M_1 and M_2 equations into the outcome equation yields the expectation

$$\begin{aligned} E(Y[t, M_1(t'), M_2\{t'', M_1(t''')\}]) = & \beta_0 + \beta_1 t + (\beta_2 + \beta_4 t)(\alpha_0 + \alpha_1 t') + \\ & (\beta_3 + \beta_5 t)(\gamma_0 + \gamma_1 t'' + (\gamma_2 + \gamma_3 t''')(\alpha_0 + \alpha_1 t''')) + \\ & (\beta_6 + \beta_7 t)[(\alpha_0 + \alpha_1 t')\{\gamma_0 + \gamma_1 t'' + \\ & (\gamma_2 + \gamma_3 t''')(\alpha_0 + \alpha_1 t''')\}] + \\ & (\beta_6 + \beta_7 t)(\gamma_2 + \gamma_3 t''')E(\epsilon_{M_1}^{t'} \epsilon_{M_1}^{t''''}) \end{aligned} \quad (2)$$

Using estimates of β , γ , and α , obtained by OLS, the estimated potential-outcome mean is the sample average

$$\widehat{\text{POM}}_{t,t',t'',t'''} = \frac{1}{N} \sum_{i=1}^N Y_i[t, M_{i,1}(t'), M_{i,2}\{t'', M_{i,1}(t''')\}] \mid \mathbf{X}_i = \mathbf{x}$$

Estimates of direct, indirect, and total effects can then be obtained by taking differences between potential-outcome means. The effect definitions and corresponding decompositions are shown in tables 1, 2, 4, and 5.

`mediate` uses a method of moments estimator, also known as an estimating equations estimator, to estimate all auxiliary and effect parameters as well as their variance–covariance matrix. For more information about the underlying `gmm` command, see [R] [gmm](#).

Sequential mediation sensitivity analysis

The reduced form in (2) contains the expected value of the product of error terms $\epsilon_{M_1}^{t'}$ and $\epsilon_{M_1}^{t''''}$ that is to be computed for sequential models that contain treatment–mediator as well as mediator–mediator interactions. Following Daniel et al. (2015), the error terms can be thought of as arising from the fact that $M_1(t)$ can be evaluated at different t when $t' \neq t''$. If we set $t' = 0$ and $t'' = 1$, for example, potential outcomes for mediator M_1 are

$$\begin{aligned} M_1(t') &= M_1(0) = \alpha_0 + \epsilon_{M_1}^{t'} \\ M_1(t''') &= M_1(1) = \alpha_0 + \alpha_1 + \epsilon_{M_1}^{t''''} \end{aligned}$$

Information about the residual covariance, which would be needed to compute the error product, is not available from the data. Therefore, this conditional correlation between $M_1(0)$ and $M_1(1)$ has to be set by assumption. We can write the expected value of the error product as $E(\epsilon_{M_1}^{t'} \epsilon_{M_1}^{t''''}) = E(\epsilon_{M_1}^{t'})E(\epsilon_{M_1}^{t''''}) + \rho_{M_1^{t'} M_1^{t''''}} \sigma_{M_1^{t'}} \sigma_{M_1^{t''''}}$, where $\rho_{M_1^{t'} M_1^{t''''}}$ is the correlation and $\sigma_{M_1^{t'}}$ and $\sigma_{M_1^{t''''}}$ are standard deviations. By assumption, we have that $\sigma_{M_1^{t'}} = \sigma_{M_1^{t''''}}$ and $E(\epsilon_{M_1}^{t'}) = E(\epsilon_{M_1}^{t''''}) = 0$. Thus, if we were to assume perfect correlation, that is, $\rho_{M_1^{t'} M_1^{t''''}} = 1$, we are left with $\sigma_{M_1^{t'}} \sigma_{M_1^{t''''}} = \sigma_{M_1}^2$. At the other extreme, assuming independence implies that $\rho_{M_1^{t'} M_1^{t''''}} = 0$ and thus that $E(\epsilon_{M_1}^{t'} \epsilon_{M_1}^{t''''}) = 0$. `mediate` assumes $\rho_{M_1^{t'} M_1^{t''''}} = 1$ and replaces $E(\epsilon_{M_1}^{t'} \epsilon_{M_1}^{t''''})$ with $\sigma_{M_1}^2$. The `estat smsensitivity` postestimation command can be used to perform a sensitivity analysis where the value of $\rho_{M_1^{t'} M_1^{t''''}}$ can be varied between 0 and 1.

CDEs

CDEs are differences in potential-outcome means between treatments where the mediators are fixed at certain values of interest. The potential outcomes in this case are $Y_i(t|M_{1i} = m_{1i}, M_{2i} = m_{2i})$ and require use of only the outcome equation. With binary treatment $T \in \{0, 1\}$, the CDEs for values $\mathbf{m} \equiv \{m_1, m_2\}$ of the two mediators are $\text{CDE}(\mathbf{m}) = Y_i(1|M_{1i} = m_1, M_{2i} = m_2) - Y_i(0|M_{1i} = m_1, M_{2i} = m_2)$. The potential outcomes $Y_i(t|M_{1i} = m_1, M_{2i} = m_2)$ reflect the conditional expectation $E(Y_i|T = t, M_{1i} = m_1, M_{2i} = m_2)$, which is computed by taking the linear prediction of the outcome equation while fixing T at 0 and 1 and the mediator values at m_1 and m_2 .

Reference

Daniel, R. M., B. L. De Stavola, S. N. Cousens, and S. Vansteelandt. 2015. Causal mediation analysis with multiple mediators. *Biometrics* 71: 1–14. <https://doi.org/10.1111/biom.12248>.

Also see

[CAUSAL] **mediate multiple postestimation** — Postestimation tools for mediate⁺

[CAUSAL] **mediate intro** — Introduction to causal mediation analysis

[CAUSAL] **mediate** — Causal mediation analysis (one mediator)

[SEM] **sem** — Structural equation model estimation command

[U] **20 Estimation and postestimation commands**

⁺Postestimation features after the two-mediator version of `mediate` are part of [StataNow](#).

[Postestimation commands](#)

[predict](#)

[estat](#)

[Remarks and examples](#)

[Stored results](#)

[Also see](#)

Postestimation commands

The following postestimation commands are of special interest after `mediate` with two mediators:

Command	Description
estat cde	controlled direct effects
estat smsensitivity	sequential mediation sensitivity analysis

The following standard postestimation commands are also available:

Command	Description
estat summarize	summary statistics for the estimation sample
estat vce	variance–covariance matrix of the estimators (VCE)
estimates	cataloging estimation results
etable	table of estimation results
lincom	point estimates, standard errors, testing, and inference for linear combinations of parameters
nlcom	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
predict	treatment effects, conditional means, etc.
predictnl	point estimates, standard errors, testing, and inference for generalized predictions
test	Wald tests of simple and composite linear hypotheses
testnl	Wald tests of nonlinear hypotheses

predict

Description for predict

predict creates a new variable (or variables) containing predictions such as treatment effects, conditional means, and linear predictions.

Menu for predict

Statistics > Postestimation

Syntax for predict

```
predict [type] { stub* | newvars } [if] [in] [ , effect_statistic ]
```

```
predict [type] newvar [if] [in] [ , fitted_statistic ]
```

effect_statistic	Description
Main	
nie	natural indirect effect; the default
nde	natural direct effect
te	total effect
cmean	conditional mean at treatment levels

fitted_statistic	Description
Main	
xb	linear prediction for outcome model
med1xb	linear prediction for first mediator model
med2xb	linear prediction for second mediator model

You specify the number of indirect effects, the number of direct effects, one total effect, or the number of potential outcomes for the model (depending on the selected statistic) with *effect_statistic*.

You specify one new variable with *fitted_statistic*.

Options for predict

Main

nie, the default, calculates natural indirect effects. You must specify a new variable for each indirect effect or use the *stub** wildcard syntax.

nde calculates natural direct effects. You must specify a new variable for each direct effect or use the *stub** wildcard syntax.

te calculates the total effect.

`cmean` calculates the conditional mean for each potential outcome $Y[t, M_1(t'), M_2\{t'', M_1(t''')\}]$ (sequential case) or $Y\{t, M_1(t'), M_2(t'')\}$ (parallel case). You must specify a new variable for each potential outcome or use the `stub*` wildcard syntax.

`xb` calculates the linear prediction for the outcome model.

`med1xb` calculates the linear prediction for the first mediator model.

`med2xb` calculates the linear prediction for the second mediator model.

estat

Description for estat

`estat cde` calculates controlled direct effects.

`estat smsensitivity` performs a sensitivity analysis for sequential mediation models.

Menu for estat

Statistics > Postestimation

Syntax for estat

Controlled direct effects

`estat cde, mvalue(mvaluespec) [cde_options]`

Sequential mediation sensitivity analysis

`estat smsensitivity [, sens_options]`

<i>cde_options</i>	Description
* <code>mvalue(<i>mvaluespec</i>)</code>	values of the mediator variables
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>contrast</code>	differences of controlled direct effects
<code>nolegend</code>	suppress table legend
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

* `mvalue(mvaluespec)` is required.

`collect` is allowed; see [U] 11.1.10 Prefix commands.

<i>sens_options</i>	Description
<code>rho(<i>numlist</i>)</code>	values of rho at which to evaluate effects
<code>format(<i>%fmt</i>)</code>	specify format for effects and standard errors
<code>nose</code>	do not estimate standard errors

`collect` is allowed; see [U] 11.1.10 Prefix commands.

Options for estat cde

`mvalue(mvaluespec)` specifies the values of the mediator variables at which to evaluate the controlled direct effect. If the causal mediation model contained a continuous treatment variable, only a single value per mediator may be specified. `mvalue()` is required.

`mvaluespec` may contain one or more of the following specifications:

- `(stat) varlist`
- `varname = #`
- `varname = (numlist)`

`varname` must be one of the mediators in the previously fit model. `stat` can be any of the following:

<i>stat</i>	Description
mean	means
median	medians
p1	1st percentile
p2	2nd percentile
...	3rd–49th percentiles
p50	50th percentile (same as median)
...	51st–97th percentiles
p98	98th percentile
p99	99th percentile
min	minimums
max	maximums
zero	fixed at zero

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [\[U\] 20.8 Specifying the width of confidence intervals](#).

`contrast` specifies to calculate differences of controlled direct effects between evaluations at different points of the mediators, where the base effect is the one defined by the first value pair in `mvalue()`; this option requires at least two evaluation points per mediator to be specified in `mvalue()`.

`nolegend` suppresses the display of the table legend.

`display_options:` `noci`, `nopvalues`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [\[R\] Estimation options](#).

Options for estat smsensitivity

`rho(numlist)` specifies the values of rho at which to evaluate effects and potential-outcome means.

`format(%fmt)` specifies the format for displaying effects and their standard errors.

`nose` suppresses calculation of standard errors for the effects evaluated at a given rho(`numlist`). `nose` is implied if the preceding specification of `mediate` was issued with the `nose` option.

Remarks and examples

Below, we provide examples for `predict`. To see an example of `estat cde`, see [Estimating controlled direct effects](#) in [CAUSAL] **mediate multiple**. To see an example of `estat smsensitivity`, see [Sensitivity analysis](#) in [CAUSAL] **mediate multiple**.

➤ Example 1: Predicting individual-level direct, indirect, and total effects

We can use `predict` to make a variety of predictions from the fitted mediation model, such as individual-level direct, indirect, and total effects; potential outcomes; and linear predictions of the outcome and mediator. Suppose we have the following mediation model:

```
. use https://www.stata-press.com/data/r19/qoflife
(Artificial quality-of-life data)

. mediate (hrqol c.age##c.age)
> (depress age i.male)
> (physact age i.male)
> (arthritis), sequential

Iteration 0: EE criterion = 3.447e-27
Iteration 1: EE criterion = 1.090e-29

Causal mediation analysis                                Number of obs = 2,000
Mediation type: Sequential
Mediator 1:      physact
Mediator 2:      depress
Treatment type: Binary
```

hrqol	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
NDE arthritis (Yes vs No)	-2.487514	.0581771	-42.76	0.000	-2.601539	-2.373489
NIE1 arthritis (Yes vs No)	-2.075981	.1567195	-13.25	0.000	-2.383146	-1.768817
NIE2 arthritis (Yes vs No)	-.3989949	.058067	-6.87	0.000	-.5128041	-.2851858
NIE12 arthritis (Yes vs No)	-.5471164	.060534	-9.04	0.000	-.6657608	-.4284719
TE arthritis (Yes vs No)	-5.509607	.2237507	-24.62	0.000	-5.94815	-5.071064

Using `predict` without options yields estimated natural indirect effects. Notice that we have three indirect effects here, so we need to specify three new variables. Or we can use the `stub*` wildcard syntax, which will generate as many variables as needed:

```
. predict nie_*
(option nie assumed; natural indirect effects)
```

Inspecting the generated variables, we can see that we have one variable for each indirect effect:

```
. describe nie_*
```

Variable name	Storage type	Display format	Value label	Variable label
nie_1	float	%9.0g		NIE1, arthritis: Yes vs No
nie_2	float	%9.0g		NIE2, arthritis: Yes vs No
nie_3	float	%9.0g		NIE12, arthritis: Yes vs No

We can predict direct and total effects by using the `nde` and `te` options, respectively. In this case, these are just single variables:

```
. predict nde, nde
. predict te, te
```

If we look at the sample means of the newly generated variables, we can see that their averages match the estimates from `mediate` for NIE1, NIE2, NIE12, NDE, and TE, respectively:

```
. summarize nie_* nde te
```

Variable	Obs	Mean	Std. dev.	Min	Max
nie_1	2,000	-2.075981	0	-2.075981	-2.075981
nie_2	2,000	-.3989949	0	-.3989949	-.3989949
nie_3	2,000	-.5471163	0	-.5471163	-.5471163
nde	2,000	-2.487514	0	-2.487514	-2.487514
te	2,000	-5.509607	0	-5.509607	-5.509607

In this case, the predictions are constant because our model did not include any interactions.



Stored results

`estat cde` stores the following results in `r()`:

Scalars	
<code>r(N)</code>	number of observations
Macros	
<code>r(title)</code>	title in estimation output
Matrices	
<code>r(b)</code>	vector of estimated controlled direct effects or their contrasts
<code>r(V)</code>	variance–covariance matrix of the estimates
<code>r(table)</code>	matrix containing the estimates with their standard errors, test statistics, <i>p</i> -values, and confidence intervals
<code>r(mvalues)</code>	matrix containing the mediator values at which the controlled direct effects are evaluated

`estat smsensitivity` stores the following results in `r()`:

Scalars	
<code>r(N)</code>	number of observations
<code>r(k_rho)</code>	number of ρ evaluation points
<code>r(rho_#)</code>	#th ρ evaluation point
Matrices	
<code>r(b_rho_#)</code>	#th vector of estimated effects
<code>r(V_rho_#)</code>	#th variance–covariance matrix of the estimates

Also see

[CAUSAL] **mediate multiple** — Causal mediation analysis (two mediators)⁺

[U] 20 **Estimation and postestimation commands**

Description

`stteffects` estimates average treatment effects, average treatment effects on the treated, and potential-outcome means using observational survival-time data. The available estimators are regression adjustment, inverse-probability weighting, and more efficient methods that combine regression adjustment and inverse-probability weighting.

For a brief description and example of each estimator, see [Remarks and examples](#) in [\[CAUSAL\] stteffects intro](#).

Syntax

```
stteffects subcommand ... [ , options ]
```

<i>subcommand</i>	Description
ra	regression adjustment
ipw	inverse-probability weighting
ipwra	inverse-probability-weighted regression adjustment
wra	weighted regression adjustment

Also see

[\[CAUSAL\] stteffects intro](#) — Introduction to treatment effects for observational survival-time data

Description

This entry provides an overview of the treatment-effects estimators that use observational survival-time data and are implemented in `stteffects`. It also provides an overview of the potential-outcomes framework and its application to survival-time data and to the interpretation of the treatment-effects parameters estimated.

The `stteffects` command estimates average treatment effects (ATEs), average treatment effects on the treated (ATETs), and potential-outcome means (POMs). Each of these effect parameters is discussed in this entry. `stteffects` implements a variety of estimators for the ATE, ATET, and POM. The treatment effects can be estimated using regression adjustment (RA), inverse-probability weights (IPW), inverse-probability-weighted regression adjustment (IPWRA), and weighted regression adjustment (WRA). This entry also provides some intuition for the estimators and discusses the tradeoffs between them.

Remarks and examples

Remarks are presented under the following headings:

Introduction

A quick tour of the estimators

Regression adjustment

Inverse-probability weighting

Combinations of RA and IPW

Weighted regression adjustment

Average treatment effect on the treated

Comparison of treatment-effects estimators

Assumptions and tradeoffs

The conditional independence assumption

The sufficient overlap assumption

The correct adjustment for censoring assumption

Assumptions for the ATET

Specification diagnostics and tests

Multivalued treatments

Introduction

The `stteffects` command estimates treatment effects using observational survival-time data.

For some intuition about the methods implemented in the `stteffects` command, consider the following question: Does smoking decrease the time to a second heart attack in the population of women aged 45–55 who have had one heart attack? Three aspects of this question stand out.

1. For ethical reasons, these data will be observational.
2. This question is about the time to an event, and such data are commonly known as survival-time data or time-to-event data. These data are nonnegative and, frequently, right-censored.
3. Many researchers and practitioners want an effect estimate in easy-to-understand units of time.

[Aspect 1](#) is one of the most common reasons for using observational data, and [aspect 2](#) focuses interest on survival-time data.

We are most concerned with [aspect 3](#) because it helps us define and understand the effect of interest. In particular, we would like to know the average change in time to a second heart attack that would occur in the population if all women smoked instead of if no women smoked. This effect is an ATE.

We must solve a missing-data problem to estimate the ATE. The ATE is the population average of the contrast in outcomes when everyone gets the treatment and when no one gets the treatment. Formally, we write this as

$$\text{ATE} = E(t_1 - t_0)$$

where t_1 is the survival time when a subject gets the treatment and t_0 is the survival time when a subject does not get the treatment. For each treatment level, there is a potential outcome that would be observed if a subject received that treatment level: t_1 is the potential outcome that would occur if someone gets the treatment and t_0 is the potential outcome that would occur if someone does not get the treatment. The missing-data problem arises because each subject receives only one treatment level, and so we observe only one of the two potential outcomes.

Much of the survival-time literature uses a hazard ratio as the effect of interest. The ATE has three advantages over the hazard ratio as an effect measure.

1. The ATE measures the effect in the same time units as the outcome instead of in relative conditional probabilities.
2. The ATE is much easier to explain to nontechnical audiences.
3. The models used to estimate the ATE can be much more flexible. Hazard ratios are useful for population effects when they are constant, which occurs when the treatment enters linearly and the distribution of the outcome has a proportional hazards form. Neither linearity in treatment nor proportional hazards form is required for the ATE, and neither is imposed on the models fit by the estimators implemented in `stteffects`.

The estimators implemented in `stteffects` use the common missing-data techniques of regression modeling, weighting, and combinations thereof to account for data lost to censoring and to unobserved potential outcomes.

Here we note only a few contributions and entry points to the vast literature on estimating ATEs. The use of potential outcomes to define treatment effects has proved extraordinarily useful; see [Holland \(1986\)](#), [Rubin \(1974\)](#), and [Heckman \(1997\)](#). [Cameron and Trivedi \(2005, chap. 25\)](#), [Wooldridge \(2010, chap. 21\)](#), and [Vittinghoff et al. \(2012, chap. 9\)](#) provide excellent general introductions to estimating ATEs.

□ Technical note

Left-truncation would be another type of missing data. The estimators implemented in `stteffects` do not adjust for left-truncation, so `stteffects` cannot be used with delayed-entry data.

`stteffects` cannot be used with time-varying covariates or multiple-record data because these add a repeated-measure structure that significantly complicates the estimation problem.

□

A quick tour of the estimators

The `stteffects` command implements five estimators of treatment effects. We introduce each one by showing the basic syntax used to apply it to a common example dataset. See each command's entry for detailed information.

We have some fictional data on the time to a second heart attack among women aged 45–55 years. The treatment, smoking, is stored in the 0/1 indicator `smoke`. These data also contain each woman’s age at the time of her first heart attack (`age`), and indices of her exercise level (`exercise`), diet quality (`diet`), and education attainment (`education`) prior to her first heart attack.

Like `streg` and other survival-time commands, `stteffects` uses the outcome variable and the failure indicator computed by `stset`. In this dataset, `atime` is the observed time in years to the second heart attack, and `fail` is the 0/1 indicator that a second heart attack was observed and recorded in `atime`. (When `fail` is 1, `atime` records the time to the second attack; when `fail` is 0, `atime` records a censored observation of the time to the second attack.)

We begin our examples by first reading in the data and then specifying the raw outcome and failure variables to `stset`.

```
. use https://www.stata-press.com/data/r19/sheart
(Time to second heart attack (fictional))
. stset atime, failure(fail)
Survival-time data settings
      Failure event: fail!=0 & fail<.
Observed time interval: (0, atime]
Exit on or before: failure
```

2,000	total observations	
0	exclusions	

2,000	observations remaining, representing	
1,208	failures in single-record/single-failure data	
3,795.226	total analysis time at risk and under observation	
	At risk from t =	0
	Earliest observed entry t =	0
	Last observed exit t =	34.17743

The output indicates that 1,208 of the 2,000 observations record actual time to a second heart attack. The remaining observations were censored. Now that we have `stset` the data, we can use `stteffects`.

Regression adjustment

Regression modeling of the outcome variable is a venerable approach to solving the missing-data problem in treatment-effects estimation. Known as the regression-adjustment (RA) estimator, this method uses averages of predicted outcomes to estimate the ATE. If the outcome model is well specified, this approach is surprisingly robust.

► Example 1: RA estimation

We now use `stteffects ra` to estimate the ATE by RA. We model the outcome as a function of age, exercise, diet, and education, and we specify that smoke is the treatment variable.

```
. stteffects ra (age exercise diet education) (smoke)
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 1.006e-14
Iteration 1:  EE criterion = 2.302e-25
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model   : Weibull
Treatment model : none
Censoring model: none
```

	_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE							
smoke (Smoker vs Nonsmoker)		-1.956657	.3331787	-5.87	0.000	-2.609676	-1.303639
POMean							
smoke Nonsmoker		4.243974	.2620538	16.20	0.000	3.730358	4.75759

When all women in the population smoke, the average time to a second heart attack is estimated to be 1.96 years less than when no women smoke. The estimated average time to a second heart attack when no women smoke is 4.24 years.

The output reports that a Weibull model was used for the outcome. The other outcome models available are exponential, gamma, and log normal. See [example 2](#) in [CAUSAL] `stteffects ra` for an application of the gamma parameterization to this model.

The ratio of the ATE to control-level POM measures the importance of the effect. In this example, when all women smoke, the time to a second heart attack falls by an estimated 46% relative to the case in which none of them smoke. See [example 3](#) in [CAUSAL] `stteffects ra` for an example that uses `nlcom` to compute a point estimate and a confidence interval for this ratio.

◀

Unlike the IPW estimator discussed in the next section, RA does not model treatment assignment or the censoring process. Treatment assignment is handled by fitting separate models for each treatment level and averaging the predicted outcomes. As is standard in the survival-time literature, the censoring term in the log-likelihood function accounts for censoring; see [Kalbfleisch and Prentice \(2002, chap. 3\)](#), [Cameron and Trivedi \(2005, chap. 17\)](#), [Cleves, Gould, and Marchenko \(2016, chap. 13\)](#), and [Wooldridge \(2010, chap. 22\)](#).

See [CAUSAL] `stteffects ra` for further discussion of this command and the RA estimator.

Inverse-probability weighting

Sometimes researchers are more comfortable modeling treatment assignment than the outcome. Inverse-probability-weighted (IPW) estimators use weighted averages of the observed outcome to estimate the POMs and the ATE. The weights correct for the missing data. When there is no censoring, the missing potential outcome is the only missing data, and the weights are constructed from a model of treatment assignment. When the data may be censored, the weights must control for censoring and the missing potential outcome. In this case, IPW estimators construct the weights from two models, one for the censoring time and one for treatment assignment.

▷ Example 2: IPW estimation

Here we use `stteffects ipw` to estimate the effect of smoking on the time to a second heart attack. The model of assignment to the treatment smoke depends on age, exercise, diet, and education. The time-to-censoring model also depends on age, exercise, diet, and education.

```
. stteffects ipw (smoke age exercise diet education)
> (age exercise diet education)

      Failure _d: fail
      Analysis time _t: atime

Iteration 0:  EE criterion = 2.042e-18
Iteration 1:  EE criterion = 2.488e-30

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit
Censoring model: Weibull
```

_t	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE smoke (Smoker vs Nonsmoker)	-2.187297	.6319837	-3.46	0.001	-3.425962	-.9486314
POmean smoke Nonsmoker	4.225331	.517501	8.16	0.000	3.211047	5.239614

When all women in the population smoke, the average time to a second heart attack is estimated to be 2.19 years less than when no women smoke. The estimated average time to a second heart attack when no women smoke is 4.23 years. When all women smoke, the average time to a second heart attack falls by an estimated 52% relative to the case when no women smoke.

The estimates have changed; however, the interpretation is the same as for the RA estimator because the IPW and RA estimators are estimating the same population effects. Under correct model specification, the estimates will differ in finite samples, but the size of these differences will decrease as the sample size gets larger. For the case at hand, the estimated ATE and control-level POM are roughly similar to those produced by the RA estimator using the Weibull model for the outcome.

Recall that IPW estimators are weighted averages of observed outcomes and that the weights control for the missing outcomes. Weights in survival-time data have two components: one for the missing potential outcome and one for data lost to censoring. We used a logit model for treatment assignment, so the component of the weights that controls for the missing potential outcome comes from the estimated logit parameters. We used a Weibull model for the time to censoring, so the component of the weights that controls for data lost to censoring comes from the estimated Weibull parameters.



Using weighting from an estimated treatment-assignment model to control for the missing potential outcome is standard in the treatment-effects literature; for example, see [CAUSAL] [teffects intro advanced](#), Wooldridge (2010, chap. 21), Vittinghoff et al. (2012, chap. 9), Hirano, Imbens, and Ridder (2003), Cattaneo (2010), and Cattaneo, Drukker, and Holland (2013). Modeling the time to censoring is specific to the survival-time treatment-effects literature; see Bai, Tsiatis, and O'Brien (2013) and Robins and Rotnitzky (2006). See *Methods and formulas* in [CAUSAL] [stteffects ipwra](#) for more details.

See [CAUSAL] [stteffects ipw](#) for further discussion of this command and the IPW estimator.

Combinations of RA and IPW

More efficient estimators are obtained by combining IPW and RA, due to Wooldridge (2007) and Wooldridge (2010, chap. 21) and denoted by IPWRA. Unlike the estimators discussed in Wooldridge (2010, chap. 21), both the treatment and the outcome models must be correctly specified to estimate the ATE.

The IPWRA estimator uses estimated weights that control for missing data to obtain missingness-adjusted regression coefficients that are used to compute averages of predicted outcomes to estimate the POMs. The estimated ATE is a contrast of the estimated POMs. These weights always involve a model for treatment assignment. You choose whether to account for censoring by including a term in the log-likelihood function or whether to use weights that also account for the data lost to censoring.

► Example 3: Likelihood-adjusted-censoring IPWRA estimation

We model the outcome (time to a second heart attack) as a function of age, exercise, diet, and education. We model assignment to the treatment smoke as a function of the same covariates.

```
. stteffects ipwra (age exercise diet education)
> (smoke age exercise diet education)

      Failure _d: fail
      Analysis time _t: atime

Iteration 0:  EE criterion = 6.476e-15
Iteration 1:  EE criterion = 1.097e-26

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : IPW regression adjustment
Outcome model   : Weibull
Treatment model : logit
Censoring model : none
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE smoke (Smoker vs Nonsmoker)	-1.592494	.4872777	-3.27	0.001	-2.54754	-.637447
POMean smoke Nonsmoker	4.214523	.2600165	16.21	0.000	3.7049	4.724146

The estimated ATE of -1.59 and control-level POM of 4.21 are similar to the reported values of -1.96 and 4.24 in [example 1](#).

We did not specify a model for the time to censoring, so censoring is handled by including a term in the log-likelihood function in the Weibull outcome model. We denote this likelihood-adjusted-censoring (LAC) version of the IPWRA estimator by LAC-IPWRA.



► Example 4: Weighted-adjusted-censoring IPWRA estimation

Instead of including a term in the log-likelihood function, the weighted-adjusted-censoring IPWRA (WAC-IPWRA) estimator uses estimated weights to adjust for censoring. We model the time to a second heart attack as a function of age, exercise, diet, and education; we model assignment to the treatment smoke as a function of the same covariates; and we model the time to censoring as a function of age, exercise, and diet.

```
. stteffects ipwra (age exercise diet education)
> (smoke age exercise diet education) (age exercise diet)

Failure _d: fail
Analysis time _t: atime
Iteration 0: EE criterion = 2.797e-13
Iteration 1: EE criterion = 2.021e-25
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : IPW regression adjustment
Outcome model  : Weibull
Treatment model: logit
Censoring model: Weibull
```

_t	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
smoke						
(Smoker						
vs						
Nonsmoker)	-2.037944	.6032549	-3.38	0.001	-3.220302	-.855586
POMean						
smoke						
Nonsmoker	4.14284	.4811052	8.61	0.000	3.199891	5.085789

The estimated ATE of -2.04 and control-level POM of 4.14 are similar to the reported values of -1.96 and 4.24 in [example 1](#).

The weights for censoring are constructed from the estimated parameters because we specified a time-to-censoring model.

◀

Under correct specification, both versions of the IPWRA estimator estimate the same ATE and control-level POM as estimated by RA and IPW.

The addition of the time-to-censoring model makes the WAC-IPWRA somewhat less robust than the LAC-IPWRA estimator. Weighting methods to control for censoring also place more restrictive assumptions on the censoring process. For example, the censoring time must be random, otherwise it would be impossible to construct the weights. In [Assumptions and tradeoffs](#) below, we discuss the tradeoffs among the estimators and the assumptions that each requires. For the moment, we note that we believe the LAC-IPWRA estimator is more robust than the WAC-IPWRA estimator.

See [\[CAUSAL\]](#) `stteffects ipwra` for further discussion of this command and the IPWRA estimator.

Weighted regression adjustment

When estimating the parameters of an outcome model, the weighted regression-adjustment (WRA) estimator uses weights instead of a term in the log-likelihood function to adjust for censoring. These weights are constructed from a model for the censoring process. The estimated parameters are subsequently used to compute averages of predicted outcomes that estimate the POMs. A contrast of the estimated POMs estimates the ATE.

► Example 5: WRA estimation

We model the time to a second heart attack as a function of age, exercise, diet, and education; we specify that smoke is the treatment; and we model the time to censoring as a function of age, exercise, and diet.

```
. stteffects wra (age exercise diet education) (smoke) (age exercise diet)
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 7.116e-15
Iteration 1:  EE criterion = 5.984e-27
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : weighted regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: Weibull
```

_t	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
smoke						
(Smoker						
vs						
Nonsmoker)	-2.152014	.4986005	-4.32	0.000	-3.129253	-1.174775
POMean						
smoke						
Nonsmoker	4.079273	.4379517	9.31	0.000	3.220903	4.937642

The estimated ATE of -2.15 and control-level POM of 4.08 are similar to the reported values of -1.96 and 4.24 in [example 1](#). Like the other estimators discussed, the WRA estimators estimate the same effect parameters as the RA estimator, so the interpretation is the same.



In many survival-time applications, using weights to adjust for censoring is probably less robust than just including a term in the log-likelihood function for the outcome model. The model used to construct the weights is just as complicated as the outcome model, and including the term in the log-likelihood function places fewer restrictions on the censoring process, as discussed in [The correct adjustment for censoring assumption](#) below.

See [\[CAUSAL\] stteffects wra](#) for further discussion of this command and the WRA estimator.

Average treatment effect on the treated

Intuitively, the average treatment effect on the treated (ATET) is the effect in a well-defined, at-risk subpopulation. Sometimes the subpopulation that gets the treatment defines such an at-risk subpopulation. For example, we may want to know the average change in time to a second heart attack among female smokers aged 45–55 who have had a heart attack if they all became nonsmokers. This effect is the ATET.

Below, we use `stteffects ra` to estimate the ATET by RA.

```
. stteffects ra (age exercise diet education) (smoke), atet
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 1.006e-14
Iteration 1:  EE criterion = 2.985e-26
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: none
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET						
smoke (Smoker vs Nonsmoker)	-1.527476	.2489203	-6.14	0.000	-2.015351	-1.039602
POMean						
smoke Nonsmoker	3.436937	.2217808	15.50	0.000	3.002255	3.87162

Now, all effects are calculated only for the subpopulation of women aged 45–55 years who smoke after their first heart attack. If no women in the subpopulation were to smoke, the average time to a second heart attack would be 3.44 years. When all women in the subpopulation smoke (the observed behavior), the average time to a second heart attack is estimated to be 1.53 years less than if no women in the subpopulation had smoked. In other words, if we could somehow turn all smokers in the subpopulation into nonsmokers, the average time to a second heart attack would be 3.44 years instead of 1.91 years ($3.44 - 1.53 = 1.91$).

These point estimates are a little different than those for the ATE and the control-level POM in the full population of women aged 45–55 years who have had one heart attack. The difference indicates that this particular health cost of smoking may be smaller among women who choose to smoke than in the full population.

Comparison of treatment-effects estimators

We can classify the estimators implemented in `stteffects` into five categories: 1) estimators based on a model for the outcome variable; 2) estimators based on models for the treatment assignment and the censoring time; 3) estimators based on models for the outcome variable and the treatment assignment; 4) estimators based on models for the outcome variable, the treatment assignment, and the censoring time; and 5) estimators based on models for the outcome variable and the censoring time.

Because there are several categories of estimators, the user must decide whether to model the outcome, the probability of treatment, the time to censoring, or some combination thereof.

Each category of estimator contains a variety of choices about the functional forms for the models.

We now provide some intuition behind each category of estimator and discuss the relationships.

1. When modeling only the outcome, separate outcome models for each treatment level account for treatment assignment, and censoring is adjusted for in the log-likelihood function. This approach is used in the RA estimators.
2. Some researchers would rather avoid modeling the outcome. Some estimators use weighted averages of the observed outcome to estimate the effect. When estimating treatment effects from observational survival-time data, the weights used must account for treatment assignment and censoring. Models for treatment assignment and time to censoring are used to construct the weights. This approach is used in the IPW estimators.
3. When seeking a more efficient estimator, it is natural to model both the outcome and the treatment and to adjust for censoring in the outcome model. This approach is used in the LAC-IPWRA estimators.
4. When seeking a more efficient estimator, another natural approach is to model both the outcome and the treatment and to adjust for censoring by weights that come from a time-to-censoring model. This approach is used in the WAC-IPWRA estimators.
5. We could modify approach 1 to model the outcome and the time to censoring so that censoring is handled by weighting and its own model instead of by likelihood adjustment. This approach is used in the WRA estimators.

While researcher preferences over what to model largely dictate the approach selected, we quickly note two points that could affect which approach works best. First, we can adjust for censoring by weighting only when censoring time is random. Second, weighting estimators become unstable if the weights get too large.

In the next section, we elaborate on the assumptions needed and the tradeoffs among the approaches to estimation.

Assumptions and tradeoffs

The estimators implemented in `stteffects` require three assumptions: conditional independence, sufficient overlap, and correct adjustment for censoring.

The conditional independence assumption

All estimators implemented in `stteffects` require the potential outcomes to be independent of the treatment assignment after conditioning on the covariates. Randomized experiments and the Heckman selection model are two motivating frameworks for the conditional independence assumption.

When the treatment is assigned randomly, the randomization ensures that the potential outcomes are independent of the treatment assignment. In observational data, the treatment is not randomly assigned. However, many important questions can only be answered using observational data because it would be unethical to randomly allocate hazardous treatments, for example, smoking. The conditional independence assumption in observational data says that treatment assignment is as good as random after conditioning on the covariates.

We can also understand conditional independence from a modeling framework. The Heckman selection model specifies that each of the potential outcomes and the treatment assignment process are functions of observable covariates and unobservable errors. The potential outcomes are conditionally independent of the treatment assignment when the unobservable errors in the treatment-assignment process are independent of the unobservable errors in each of the potential-outcome processes. See *The CI assumption* in [CAUSAL] **teffects intro advanced** for a detailed example.

Both frameworks lead to the same conclusion: we need to observe and to condition on a sufficient number of covariates.

Essentially, all the estimators in `stteffects` are equally susceptible to violations of the conditional independence assumption. No one estimator is any more robust to the conditional independence assumption than any other one.

Estimating the ATE among the subpopulation of those who get the treatment requires a significantly weaker version of the CI assumption; see *Assumptions for the ATET* below.

For more details about the conditional independence assumption, see *The CI assumption* in [CAUSAL] **teffects intro advanced**, and see Rosenbaum and Rubin (1983), Heckman (1997), Imbens and Wooldridge (2009), Cameron and Trivedi (2005, sec. 25.2), Wooldridge (2010, chap. 21), and Vittinghoff et al. (2012, chap. 9).

The sufficient overlap assumption

The sufficient overlap assumption requires that each individual have a sufficiently positive probability of being assigned to each treatment level. We believe that the RA estimator is more robust than the other estimators to near violations of the sufficient overlap condition, under correct model specification.

The overlap condition has no specification test, but using `teoverlap` and then summarizing the predicted treatment probabilities presents good diagnostics of overlap problems.

The correct adjustment for censoring assumption

The correct adjustment for censoring assumption has two parts. First, either the censoring time must be fixed or the process must be conditionally-on-covariates independent of the potential outcomes and the treatment-assignment process. This assumption is standard in survival analysis; see, for example, Kalbfleisch and Prentice (2002, chap. 3).

Second, the method used to adjust to censoring must be correct. For the RA and LAC-IPWRA estimators, which use likelihood-adjusted censoring, the second assumption is no more restrictive than assuming correct specification of the outcome model. For the IPW, WAC-IPWRA, and WRA estimators, which adjust by weighting, the second assumption requires that the censoring be random and that the censoring process be correctly modeled.

Under correct specification, all the estimators in `stteffects` perform well. However, we believe that estimators that use likelihood adjustment instead of weighting are more robust for three reasons.

1. The estimators that use weighting to adjust for censoring cannot handle fixed censoring processes. If the censoring process is not random, the weights are not well defined.
2. The estimators that use weighting to adjust for censoring do not allow the random censoring process to vary by treatment level.
3. The estimators that use weighting to adjust for censoring require an additional sufficient overlap condition: the probability of not being censored must be sufficiently greater than 0 or else the weights that adjust for censoring get too large.

While the estimators that use WAC instead of LAC require a few more assumptions, some researchers are more comfortable modeling the treatment and censoring than the outcome. In this case, the IPW or WAC-IPWRA estimator would be the estimator of choice.

See [Specification diagnostics and tests](#) below for information about testing these assumptions.

Assumptions for the ATET

We noted in [Average treatment effect on the treated](#) that the ATET is sometimes more interesting than the ATE. We can also estimate the ATET under less restrictive versions of the conditional independence assumption and the sufficient overlap assumption than those required for the ATE.

While ATE estimation requires that the potential outcomes for both the treated and the not treated be conditionally independent of treatment assignment, ATET estimation requires that only the not treated potential outcome be conditionally independent of treatment assignment.

This weaker version of conditional independence allows the gains from the treatment to be related to treatment assignment, after conditioning on the covariates. We can estimate the ATET, but not the ATE, if some unobserved factor increases (or decreases) the likelihood of assignment to the treatment, increases (or decreases) the time to event in the treatment group, and has no effect on the time to event when not in the treatment group.

For example, suppose that smoking is an acquired taste and that individuals who acquire the taste for smoking more easily are less adversely affected by smoking and otherwise similar to everyone else when not smoking. Taste for smoking is unobservable, and our data have no measure of this variable. In this case, we could estimate the ATET but not the ATE.

The weaker version of the sufficient overlap assumption only requires that each individual in the treated subpopulation have a positive probability of not getting treated. In contrast, ATE estimation requires that each individual in the population have a positive probability of getting each treatment level. In particular, we can estimate the ATET, but not the ATE, when some individuals in the population have zero chance of getting the treatment. For example, we could estimate the ATET, but not the ATE, if some women will never smoke for religious reasons.

Even when the conditions for ATE estimation hold, the ATE and ATET may differ. Finding that the ATET is significantly different from the ATE does not mean that the ATE is incorrectly estimated.

See [Heckman \(1997\)](#) and [Wooldridge \(2010, 911–912\)](#) for more information about the assumptions necessary to estimate the ATET.

Specification diagnostics and tests

After `stteffects ipw` and `stteffects ipwra`, some specification checks for the treatment-assignment model and the overlap condition are available.

The checks for the treatment-assignment model are known as balance checks. When the covariate distributions are invariant to the treatment level, the covariates are said to be balanced. The concept of balanced covariates comes from the experimental literature, in which random treatment assignment ensures that the covariates are balanced.

In observational data, the covariates are almost never balanced in the raw data. Weighting methods can be viewed as using a treatment-assignment model to balance the covariates. If the treatment-assignment model is well specified, the weights constructed from this model will balance the covariates. One of the nice features of balance checks is that they do not depend on the outcome or its distribution. This fact is especially useful for survival-time outcomes because censoring of the outcome has no effect on the balance checks, so the balance checks implemented in `tebalance` work without modification.

Conditional on the treatment-assignment model being well specified, we can use the estimated probabilities of treatment, known as the propensity scores, to look for signs that the overlap condition is violated. These checks depend only on the estimated treatment probabilities and are not affected by any censoring of the outcome, so the methods implemented in `teoverlap` work without modification.

We begin examining our model by using `tebalance summarize` after refitting the models used by the LAC-IPWRA estimator.

```
. quietly stteffects ipwra (age exercise diet education)
> (smoke age exercise diet education)
. tebalance summarize
```

	Raw	Weighted
Number of obs =	2,000	2,000.0
Treated obs =	738	994.1
Control obs =	1,262	1,005.9

	Standardized differences		Variance ratio	
	Raw	Weighted	Raw	Weighted
age	-.3122094	-.0184574	.8547308	.9370065
exercise	-.4975269	-.0458412	.4966778	.8342339
diet	-.2479756	.0021802	.7937645	1.095347
education	-.4801442	-.0216366	.6015139	.978078

The weighted standardized differences are much closer to 0 than the raw standardized differences, and the weighted variance ratios are much closer to 1 than the raw variance ratios. These results indicate that the model-based treatment weights balanced the covariates; see [\[CAUSAL\] tebalance](#) and [\[CAUSAL\] tebalance summarize](#) for details.

The diagnostics presented by `tebalance` summarize are not a formal test. However, we can use `tebalance overid` to conduct a formal test of the hypothesis that the weights constructed from the treatment-assignment model balanced the covariates.

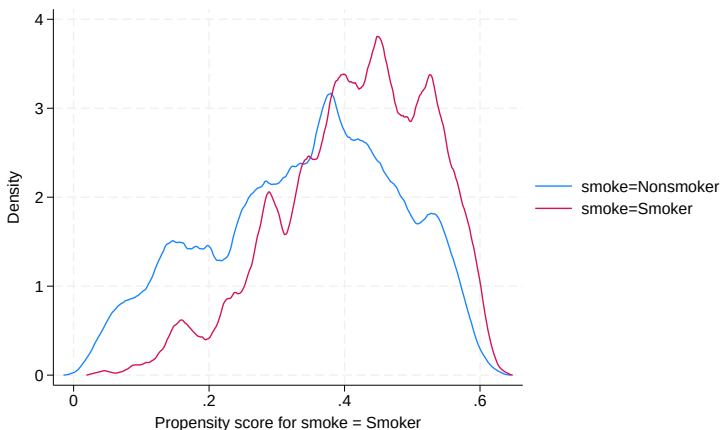
```
. tebalance overid
Iteration 0: Criterion = .22681884
Iteration 1: Criterion = .22692316 (backed up)
Iteration 2: Criterion = .23090157
Iteration 3: Criterion = .23114612
Iteration 4: Criterion = .23256286
Iteration 5: Criterion = .23286304
Iteration 6: Criterion = .23335857
Iteration 7: Criterion = .2335567
Iteration 8: Criterion = .2335671
Iteration 9: Criterion = .23356711

Overidentification test for covariate balance
H0: Covariates are balanced
      chi2(5)      = 3.28142
      Prob > chi2   = 0.6567
```

There is no significant evidence against the null hypothesis. The interpretation is that we do not reject the null hypothesis that the treatment-assignment model is well specified; see [\[CAUSAL\] tebalance](#) and [\[CAUSAL\] tebalance overid](#) for details.

Given that we do not reject the treatment-assignment model, we can use this model to look for evidence that the overlap condition is violated. We begin by using `teoverlap`.

```
. teoverlap, ptlevel(Smoker)
```



The densities of the propensity scores for the smokers and nonsmokers appear to have the same support, indicating that there is no violation of the overlap condition. The only indicator of a possible problem is that the support of the density for nonsmokers gets very close to 0. This problem would affect ATE estimation but not ATET estimation, as discussed in [Assumptions and tradeoffs](#). To further investigate, we compute and summarize the predicted propensity score by treatment level.

```
. predict ps1, ps tlevel(Smoker)
```

```
. summarize ps1 if smoke == 0
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps1	1,262	.3410001	.1381673	.014819	.6161401

```
. summarize ps1 if smoke == 1
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps1	738	.4168805	.1107557	.0454891	.6216282

To interpret these results, recall that ATE estimation requires that the minimum propensity score for each treatment level be sufficiently greater than 0 and that the maximum propensity score for each treatment level be sufficiently less than 1. Also recall that ATET estimation only requires that the maximum propensity score for each treatment level be sufficiently less than 1.

For ATE estimation, only the minimum predicted propensity score for nonsmokers presents a challenge, and 0.015 is probably not too small. For ATET estimation, neither maximum causes concern.

For information about choosing among the `stteffects` estimators and their functional forms for the different models, see [Model choice](#) under *Remarks and examples* in [\[CAUSAL\] teffects intro advanced](#).

Multivalued treatments

`stteffects` can estimate treatment effects for multivalued treatments; here we provide some examples. See [\[CAUSAL\] teffects multivalued](#) for an introduction to interpreting effects from multivalued treatments.

► Example 6: Multivalued ATE estimation

We have another fictional dataset that records the time to a second heart attack among women aged 45–55 years. In this dataset, `atime` is the observed time in years to the second heart attack, and `fail` is the 0/1 indicator that a second heart attack was observed and recorded in `atime`. (When `fail` is 1, `atime` records the time to the second attack; when `fail` is 0, `atime` records a censored observation of the time to the second attack.)

These data also contain the age at the time of the first heart attack (`age`), and indices of each woman's exercise level (`exercise`), diet quality (`diet`), and education attainment (`education`) prior to her first heart attack.

The treatment, smoking, is stored in the categorical variable `smoke`, which has the following value labels. The women who never smoked are labeled as N; the women who previously smoked but quit before their first heart attack are labeled as B; the women who previously smoked but quit after their first heart attack are labeled as A; and the women who continued to smoke after their first heart attack are labeled as S.

We begin by first reading in the data and then reviewing information previously stored using `stset`.

```
. use https://www.stata-press.com/data/r19/sheartm, clear
(Time to second heart attack (fictional))

. stset
-> stset atime, failure(fail)
Survival-time data settings

      Failure event: fail!=0 & fail<.
Observed time interval: (0, atime]
      Exit on or before: failure
```

10,000	total observations
0	exclusions

10,000	observations remaining, representing
9,741	failures in single-record/single-failure data
27,999.155	total analysis time at risk and under observation
	At risk from t = 0
	Earliest observed entry t = 0
	Last observed exit t = 17.40826

We continue by tabulating the treatment variable `smoke`.

```
. tabulate smoke
```

Smoking level	Freq.	Percent	Cum.
N	3,167	31.67	31.67
B	2,263	22.63	54.30
A	1,924	19.24	73.54
S	2,646	26.46	100.00
Total	10,000	100.00	

We see that 31.67% of the women never smoked, 22.63% of the women previously smoked but quit before their first heart attack, 19.24% of the women previously smoked but quit after their first heart attack, and 26.46% of the women continued to smoke after their first heart attack.

We now use `stteffects ra` to estimate the ATE by RA. We model the outcome as a function of age, exercise, diet, and education, and we specify that `smoke` is the treatment variable.

```
. stteffects ra (age exercise diet education) (smoke)
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 1.390e-20
Iteration 1:  EE criterion = 1.375e-29
Survival treatment-effects estimation      Number of obs      =      10,000
Estimator      : regression adjustment
Outcome model   : Weibull
Treatment model : none
Censoring model : none
```

_t	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
smoke						
(B vs N)	-.4129793	.0317	-13.03	0.000	-.47511	-.3508485
(A vs N)	-1.281031	.032866	-38.98	0.000	-1.345447	-1.216614
(S vs N)	-2.167359	.0338994	-63.93	0.000	-2.233801	-2.100917
POmean						
smoke						
N	3.745919	.0289014	129.61	0.000	3.689273	3.802565

The average time to a second heart attack is 0.41 years sooner when all the women smoked at some point but quit before their first heart attack than when all the women never smoked. The average time to a second heart attack is 1.28 years sooner when all the women smoked at some point but quit after their first heart attack than when all the women never smoked. The average time to a second heart attack is 2.17 years sooner when all the women continued to smoke after their first heart attack than when all the women never smoked.



► Example 7: Multivalued ATET estimation

In the at-risk subpopulation of women who continued to smoke, we want to estimate the effect of continuing to smoke (S) versus quitting after the first heart attack (A). Below we estimate the ATETs by RA, specifying A to be the control level and S to be the treatment level.

```
. stteffects ra (age exercise diet education) (smoke), atet control(A) tlevel(S)

      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 1.390e-20
Iteration 1:  EE criterion = 1.372e-29

Survival treatment-effects estimation      Number of obs      =      10,000
Estimator      : regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: none
```

	_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET							
	smoke						
	(N vs A)	1.290123	.0377552	34.17	0.000	1.216125	1.364122
	(B vs A)	.8748349	.0239595	36.51	0.000	.8278751	.9217946
	(S vs A)	-.8869257	.0272301	-32.57	0.000	-.9402958	-.8335557
POmean							
	smoke						
	A	2.500108	.0217833	114.77	0.000	2.457413	2.542802

The parameter (S vs A) is the one of interest. The estimate implies that the average time to a second heart attack among women who continue to smoke is 0.89 years sooner when they all continue to smoke than when they all quit smoking after their first heart attack.



Acknowledgments

We thank Xiaofei Bai of Servier, Matias Cattaneo of Princeton University, Barbara Sianesi of the Institute for Fiscal Studies, and Jeff Wooldridge of Michigan State University for helpful conversations about methods and the literature.

References

- Bai, X., A. A. Tsiatis, and S. M. O'Brien. 2013. Doubly robust estimators of treatment-specific survival distributions in observational studies with stratified sampling. *Biometrics* 69: 830–839. <https://doi.org/10.1111/biom.12076>.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Cleves, M. A., W. W. Gould, and Y. V. Marchenko. 2016. *An Introduction to Survival Analysis Using Stata*. Rev. 3rd ed. College Station, TX: Stata Press.

- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: Sage.
- Heckman, J. J. 1997. Instrumental variables: A study of implicit behavioral assumptions used in making program evaluations. *Journal of Human Resources* 32: 441–462. <https://doi.org/10.2307/146178>.
- Hirano, K., G. W. Imbens, and G. Ridder. 2003. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica* 71: 1161–1189. <https://doi.org/10.1111/1468-0262.00442>.
- Holland, P. W. 1986. Statistics and causal inference. *Journal of the American Statistical Association* 81: 945–960. <https://doi.org/10.2307/2289064>.
- Imbens, G. W. 2000. The role of the propensity score in estimating dose–response functions. *Biometrika* 87: 706–710. <https://doi.org/10.1093/biomet/87.3.706>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Kalbfleisch, J. D., and R. L. Prentice. 2002. *The Statistical Analysis of Failure Time Data*. 2nd ed. New York: Wiley.
- Robins, J. M., and A. Rotnitzky. 2006. “Inverse probability weighting in survival analysis”. In *Survival and Event History Analysis*, edited by N. Keiding and P. K. Andersen, 266–271. Chichester, UK: Wiley.
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.
- Royston, P. 2015. *Estimating the treatment effect in a clinical trial using difference in restricted mean survival time*. *Stata Journal* 15: 1098–1117.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2007. Inverse probability weighted estimation for general missing data problems. *Journal of Econometrics* 141: 1281–1301. <https://doi.org/10.1016/j.jeconom.2007.02.002>.
- . 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

- [CAUSAL] **stteffects postestimation** — Postestimation tools for stteffects
- [CAUSAL] **teffects intro advanced** — Advanced introduction to treatment effects for observational data
- [ST] **streg** — Parametric survival models
- [ST] **stset** — Declare data to be survival-time data
- [U] **20 Estimation and postestimation commands**

[Description](#)
[Options](#)
[References](#)

[Quick start](#)
[Remarks and examples](#)
[Also see](#)

[Menu](#)
[Stored results](#)

[Syntax](#)
[Methods and formulas](#)

Description

`stteffects ipw` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational survival-time data with random time to censoring. Estimation is by inverse-probability weighting (IPW). IPW estimators use weighted averages of the observed outcome. The estimated weights correct for missing data on the potential outcomes and for censored survival times. `stteffects ipw` offers several choices for the functional forms of the treatment model and the time-to-censoring model. Binary and multivalued treatments are accommodated.

See [\[CAUSAL\] stteffects intro](#) for an overview of estimating treatment effects from observational survival-time data.

Quick start

Specify time as observed failure time and `fail` as failure indicator

```
stset time, failure(fail)
```

ATE of binary `treat2` on time by IPW using a logistic model of `treat2` on `x` and `w` and using `x` and `w` in a Weibull model for the censoring time

```
stteffects ipw (treat2 x w) (x w)
```

Same as above, but estimate the ATET

```
stteffects ipw (treat2 x w) (x w), atet
```

ATE of `treat2` on time by IPW using a probit model of `treat2` on `x` and `w` and using `x` and `w` in a gamma model for the censoring time

```
stteffects ipw (treat2 x w, probit) (x w, gamma)
```

ATE for treatment levels 2 and 3 of three-valued treatment `treat3`

```
stteffects ipw (treat3 x w) (x w)
```

Same as above, and specify that `treat3 = 3` is the control level using the value label “MyControl” for 3

```
stteffects ipw (treat3 x w) (x w), control("MyControl")
```

Menu

Statistics > Causal inference/treatment effects > Survival outcomes > Inverse-probability weighting (IPW)

Syntax

```
stteffects ipw (tvar tmvarlist [ , tmoptions ]) (cmvarlist [ , cmoptions ])
[if] [in] [ , stat options ]
```

tvar must contain integer values representing the treatment levels.
tmvarlist specifies the variables that predict treatment assignment in the treatment model.
cmvarlist specifies the variables that predict censoring in the censoring model.

tmoptions	Description
Model	
logit	logistic treatment model; the default
probit	probit treatment model
hetprobit(varlist)	heteroskedastic probit treatment model
noconstant	suppress constant from treatment model
cmoptions	Description
Model	
weibull	Weibull; the default
exponential	exponential
gamma	two-parameter gamma
lnormal	lognormal
ancillary(avarlist [, noconstant])	specify variables used to model ancillary parameter
noconstant	suppress constant from censoring model
stat	Description
Stat	
ate	estimate average treatment effect in population; the default
atet	estimate average treatment effect on the treated
pomeans	estimate potential-outcome means

<i>options</i>	Description
SE/Robust <code>vce(vcetype)</code>	<i>vcetype</i> may be <code>robust</code> , <code>cluster clustvar</code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting <code>level(#)</code> <code>aequations</code> <code>noshow</code> <i>display_options</i>	set confidence level; default is <code>level(95)</code> display auxiliary-equation results do not show st setting information control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization <i>maximize_options</i> <code>iterinit(#)</code>	control the maximization process; seldom used specify starting-value iterations; seldom used
Advanced <code>pstolerance(#)</code> <code>osample(newvar)</code> <code>control(# label)</code> <code>tlevel(# label)</code> <code>coeflegend</code>	set tolerance for the overlap assumption identify observations that violate the overlap assumption specify the level of <i>tvar</i> that is the control specify the level of <i>tvar</i> that is the treatment display legend instead of statistics

You must `stset` your data before using `stteffects`; see [ST] `stset`.

tmvarlist, *cmvarlist*, and *avarlist* may contain factor variables; see [U] 11.4.3 Factor variables.

`bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the `bootstrap` prefix; see [R] `bootstrap`.

`fweights`, `iweights`, and `pweights` may be specified using `stset`; see *Weights* under *Remarks and examples* in [ST] `stset`.

However, weights may not be specified if you are using the `bootstrap` prefix.

`coeflegend` does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

Model

`ancillary(avarlist [ , noconstant ])` specifies the variables used to model the ancillary parameter.

By default, the ancillary parameter does not depend on covariates. Specifying `ancillary(avarlist, noconstant)` causes the constant to be suppressed in the model for the ancillary parameter.

`noconstant`; see [R] *Estimation options*.

Stat

stat is one of three statistics: `ate`, `atet`, or `pomeans`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`atet` specifies that the average treatment effect on the treated be estimated.

`pomeans` specifies that the potential-outcome means for each treatment level be estimated.

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

Reporting

`level(#)`; see [R] [Estimation options](#).

`aequations` specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`noshow` prevents `stteffects ipw` from showing the key `st` variables. This option is rarely used because most people type `stset`, `show` or `stset`, `noshow` to permanently set whether they want to see these variables mentioned at the top of the output of every `st` command; see [ST] [stset](#).

display_options: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

Maximization

maximize_options: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

`init_specs` is one of

`matname [ , skip copy]`

`# [ , # ...], copy`

`iterinit(#)` specifies the maximum number of iterations used to calculate the starting values. This option is seldom used.

Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `stteffects` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption.

`control(#|label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with the statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(#|label)` specifies the level of `tvar` that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `stteffects` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

If you are not familiar with the framework for treatment-effects estimation from observational survival-time data, please see [\[CAUSAL\] stteffects intro](#).

IPW estimators use contrasts of weighted averages of observed outcomes to estimate treatment effects. The estimated weights correct for data that are missing because each subject is only observed after receiving one of the possible treatment levels and because some survival-time outcomes are censored.

The IPW estimators implemented in `stteffects ipw` use a three-step approach to estimating the ATE:

1. Estimate the parameters of a treatment-assignment model, and compute the component of the estimated weights that accounts for data missing because each subject is only observed after receiving one of the possible treatment levels.
2. Estimate the parameters of a time-to-censoring model, and compute the component of the estimated weights that accounts for data lost to censoring.
3. Use the estimated weights to compute weighted averages of the outcomes for each treatment level.

To estimate the ATET, we use different weights in step 2.

The time to censoring must be random to use `stteffects ipw` because the model in step 2 is not well defined if the time to censoring is fixed. See [\[CAUSAL\] stteffects intro](#) for more details. For information about estimators that accommodate a fixed time to censoring, see [\[CAUSAL\] stteffects ra](#) and [\[CAUSAL\] stteffects ipwra](#).

Here we note only a few entry points to the vast literature on IPW estimators. [Hirano, Imbens, and Ridder \(2003\)](#), [Imbens \(2000, 2004\)](#), [Imbens and Wooldridge \(2009\)](#), [Rosenbaum and Rubin \(1983\)](#), [Robins and Rotnitzky \(2006\)](#), [Wooldridge \(2002, 2007\)](#), [Cameron and Trivedi \(2005, chap. 25\)](#), [Wooldridge \(2010, chap. 21\)](#), and [Vittinghoff et al. \(2012, chap. 9\)](#) provide excellent general introductions to estimating ATEs and to the IPW estimators in particular.

Like `streg` and other survival-time commands, `stteffects ipw` uses the outcome variable and the failure indicator computed by, and optionally weights specified with, `stset`. `stteffects ipw` is not appropriate for data with time-varying covariates, also known as multiple-record survival-time data, or for delayed-entry data.

► Example 1: Estimating the ATE

Suppose we wish to study the effect of smoking on the time to a second heart attack among women aged 45–55 years. In our fictional `sheart` dataset, `atime` is the observed time in years to a second heart attack or censoring, and `fail` is the 0/1 indicator that a second heart attack was observed. (When `fail` is 1, `atime` records the time to the second heart attack; when `fail` is 0, `atime` records a censored observation of the time to a second heart attack.) We previously `stset` these data; see [A quick tour of the estimators](#) in [\[CAUSAL\] stteffects intro](#).

The treatment, smoking, is stored in the 0/1 indicator `smoke`. These data also contain age at the time of the first heart attack (`age`), and indices of the level of exercise (`exercise`), diet quality (`diet`), and education (`education`) prior to the first heart attack.

We can use `stteffects ipw` to estimate the ATE. We model treatment assignment using the default logit model with covariates on `age`, `exercise`, and `education`. We model the time to censoring using the default Weibull model with covariates on `age`, `exercise`, `diet`, and `education`.

```

. use https://www.stata-press.com/data/r19/sheart
(Time to second heart attack (fictional))
. stteffects ipw (smoke age exercise education) (age exercise diet education)

      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 2.042e-18
Iteration 1:  EE criterion = 1.890e-30
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit
Censoring model: Weibull

```

_t	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
smoke						
(Smoker						
vs						
Nonsmoker)	-2.22226	.6307573	-3.52	0.000	-3.458522	-.9859983
POMean						
smoke						
Nonsmoker	4.235569	.5210937	8.13	0.000	3.214244	5.256894

When every woman smoked in the population of women aged 45–55 years who have had a heart attack, the average time to a second heart attack is estimated to be 2.22 years less than when no women in the population of interest smoked. The estimated average time to a second heart attack when no women in the population of interest smoked is 4.24 years.

The ratio of the ATE to the control-level POM measures the importance of the effect. In this example, when every woman smoked, the average time to a second heart attack falls by an estimated 52% relative to the case when none of them smoked. See [example 3](#) in [\[CAUSAL\] stteffects ra](#) for an example that uses `nlcom` to compute a point estimate and a confidence interval for this ratio.

► Example 2: Different treatment and censoring models

Instead of a logit model for the treatment assignment, we could have used a probit or a heteroskedastic probit model. Instead of a Weibull model for the censoring time, we could have used an exponential, a gamma, or a lognormal model. For a quick comparison, we now estimate the ATE using a probit model for the treatment assignment and using a gamma model for the censoring time.

```
. stteffects ipw (smoke age exercise education, probit)
> (age exercise diet education, gamma)

      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 3.534e-15
Iteration 1:  EE criterion = 5.203e-27

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
Censoring model: gamma
```

_t	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
smoke (Smoker vs Nonsmoker)	-2.646808	.8368254	-3.16	0.002	-4.286956	-1.006661
POmean						
smoke Nonsmoker	4.702301	.7404567	6.35	0.000	3.251033	6.15357

The estimated ATE of -2.65 and control-level POM of 4.70 are similar to the values of -2.22 and 4.24 reported in [example 1](#).

► Example 3: Estimating the ATET

Intuitively, the ATET measures the effect of the treatment on an at-risk subpopulation. Sometimes the subpopulation that gets the treatment defines such an at-risk subpopulation. The ATET has the added benefit that it can be estimated under weaker conditions than the ATE; see [Assumptions and tradeoffs](#) under *Remarks and examples* in [\[CAUSAL\] stteffects intro](#).

```
. stteffects ipw (smoke age exercise education) (age exercise diet education),
> atet

      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 2.042e-18
Iteration 1:  EE criterion = 1.017e-31

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit
Censoring model: Weibull
```

_t	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATET						
smoke (Smoker vs Nonsmoker)	-1.846136	.5076872	-3.64	0.000	-2.841185	-.8510877
P0mean						
smoke Nonsmoker	3.543788	.474395	7.47	0.000	2.613991	4.473585

When every woman in the subpopulation smoked, the average time to a second heart attack is estimated to be 1.85 years less than when no women in the subpopulation smoked. The estimated average time to a second heart attack when no women in the subpopulation smoked is 3.54 years.



Stored results

`stteffects ipw` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>stteffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(dead)</code>	<code>_d</code>
<code>e(depvar)</code>	<code>_t</code>
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>ipw</code>
<code>e(tmodel)</code>	treatment model: <code>logit</code> , <code>probit</code> , or <code>hetprobit</code>
<code>e(cmodel)</code>	censoring model: <code>weibull</code> , <code>exponential</code> , <code>gamma</code> , or <code>lognormal</code>
<code>e(stat)</code>	statistic estimated: <code>ate</code> , <code>atet</code> , or <code>pomeans</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

The methods and formulas for the IPW estimators implemented in `stteffects ipw` are given in [Methods and formulas](#) of [CAUSAL] `stteffects ipwra`.

References

- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Bai, X., A. A. Tsiatis, and S. M. O'Brien. 2013. Doubly robust estimators of treatment-specific survival distributions in observational studies with stratified sampling. *Biometrics* 69: 830–839. <https://doi.org/10.1111/biom.12076>.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: Sage.
- Hirano, K., G. W. Imbens, and G. Ridder. 2003. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica* 71: 1161–1189. <https://doi.org/10.1111/1468-0262.00442>.
- Imbens, G. W. 2000. The role of the propensity score in estimating dose–response functions. *Biometrika* 87: 706–710. <https://doi.org/10.1093/biomet/87.3.706>.
- . 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. <https://doi.org/10.1162/003465304323023651>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Robins, J. M., and A. Rotnitzky. 2006. “Inverse probability weighting in survival analysis”. In *Survival and Event History Analysis*, edited by N. Keiding and P. K. Andersen, 266–271. Chichester, UK: Wiley.
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2002. Inverse probability weighted M-estimators for sample selection, attrition, and stratification. *Portuguese Economic Journal* 1: 117–139. <https://doi.org/10.1007/s10258-002-0008-x>.
- . 2007. Inverse probability weighted estimation for general missing data problems. *Journal of Econometrics* 141: 1281–1301. <https://doi.org/10.1016/j.jeconom.2007.02.002>.
- . 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

- [CAUSAL] **stteffects postestimation** — Postestimation tools for stteffects
- [CAUSAL] **stteffects intro** — Introduction to treatment effects for observational survival-time data
- [ST] **streg** — Parametric survival models
- [ST] **stset** — Declare data to be survival-time data
- [U] **20 Estimation and postestimation commands**

[Description](#)
[Options](#)
[References](#)

[Quick start](#)
[Remarks and examples](#)
[Also see](#)

[Menu](#)
[Stored results](#)

[Syntax](#)
[Methods and formulas](#)

Description

`stteffects ipwra` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational survival-time data by inverse-probability-weighted regression adjustment (IPWRA). IPWRA estimators use missingness-adjusted regression coefficients to compute averages of treatment-level predicted outcomes. Contrasts of these averages estimate the treatment effects. `stteffects ipwra` offers several choices for the functional forms of the outcome model, of the treatment model, and of the optional time-to-censoring model. Binary and multivalued treatments are accommodated.

See [\[CAUSAL\] stteffects intro](#) for an overview of estimating treatment effects from observational survival-time data.

Quick start

Specify time as observed failure time and fail as failure indicator

```
stset time, failure(fail)
```

ATE of binary treatment `treat2` estimated by IPWRA using a Weibull model for time on `x1` and `x2` and a logistic model for `treat2` on `x1` and `w`

```
stteffects ipwra (x1 x2) (treat2 x1 w)
```

Same as above, but estimate the ATET

```
stteffects ipwra (x1 x2) (treat2 x1 w), atet
```

Gamma model for time and probit model for `treat2`

```
stteffects ipwra (x1 x2, gamma) (treat2 x1 w, probit)
```

ATE for each level of three-valued treatment `treat3`

```
stteffects ipwra (x1 x2) (treat3 x1 w)
```

Same as above, and specify that `treat3 = 3` is the control level using the value label “MyControl” for 3

```
stteffects ipwra (x1 x2) (treat3 x1 w), control("MyControl")
```

ATE of `treat2` estimated by IPWRA using a Weibull model for time on `x1` and `x2`, a logistic model for `treat2` on `x1` and `w`, and a Weibull model for the time to censoring with covariates `x1` and `x2`

```
stteffects ipwra (x1 x2) (treat2 x1 w) (x1 x2)
```

Gamma model for time, probit model for `treat2`, and gamma model for censoring

```
stteffects ipwra (x1 x2, gamma) (treat2 x1 w, probit) (x1 x2, gamma)
```

Menu

Statistics > Causal inference/treatment effects > Survival outcomes > Regression adjustment with IPW

Syntax

```
stteffects ipwra (omvarlist [ , omoptions ]) (tvar tmvarlist [ , toptions ])
[ (cmvarlist [ , cmoptions ]) ] [ if ] [ in ] [ , stat options ]
```

omvarlist specifies the variables that predict the survival-time variable in the outcome model.
tvar must contain integer values representing the treatment levels.
tmvarlist specifies the variables that predict treatment assignment in the treatment model.
cmvarlist specifies the variables that predict censoring in the censoring model.

<i>omoptions</i>	Description
Model	
<u>weibull</u>	Weibull; the default
<u>exponential</u>	exponential
<u>gamma</u>	two-parameter gamma
<u>lnormal</u>	lognormal
ancillary(<i>avarlist</i> [, <u>noconstant</u>])	specify variables used to model ancillary parameter
<u>noconstant</u>	suppress constant from outcome model

<i>toptions</i>	Description
Model	
logit	logistic treatment model; the default
probit	probit treatment model
hetprobit(<i>varlist</i>)	heteroskedastic probit treatment model
<u>noconstant</u>	suppress constant from treatment model

<i>cmoptions</i>	Description
Model	
<u>weibull</u>	Weibull; the default
<u>exponential</u>	exponential
<u>gamma</u>	two-parameter gamma
<u>lnormal</u>	lognormal
ancillary(<i>avarlist</i> [, <u>noconstant</u>])	specify variables used to model ancillary parameter
<u>noconstant</u>	suppress constant from censoring model

<i>stat</i>	Description
Stat	
ate	estimate average treatment effect in population; the default
atet	estimate average treatment effect on the treated
<u>pomeans</u>	estimate potential-outcome means

<i>options</i>	Description
SE/Robust <code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>cluster <i>clustvar</i></code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting <code>level(#)</code> <code>aequations</code> <code>noshow</code> <i>display_options</i>	set confidence level; default is <code>level(95)</code> display auxiliary-equation results do not show st setting information control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization <i>maximize_options</i> <code>iterinit(#)</code>	control the maximization process; seldom used specify starting-value iterations; seldom used
Advanced <code>pstolerance(#)</code> <code>osample(<i>newvar</i>)</code> <code>control(# <i>label</i>)</code> <code>tlevel(# <i>label</i>)</code> <code>coeflegend</code>	set tolerance for the overlap assumption identify observations that violate the overlap assumption specify the level of <i>tvar</i> that is the control specify the level of <i>tvar</i> that is the treatment display legend instead of statistics

You must `stset` your data before using `stteffects`; see [ST] [stset](#).

`omvarlist`, `tmvarlist`, `cmvarlist`, and `avarlist` may contain factor variables; see [U] [11.4.3 Factor variables](#).

`bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see [U] [11.1.10 Prefix commands](#).

Weights are not allowed with the `bootstrap` prefix; see [R] [bootstrap](#).

`fweights`, `iweights`, and `pweights` may be specified using `stset`; see [Weights](#) under *Remarks and examples* in [ST] [stset](#).

However, weights may not be specified if you are using the `bootstrap` prefix.

`coeflegend` does not appear in the dialog box.

See [U] [20 Estimation and postestimation commands](#) for more capabilities of estimation commands.

Options

Model

`ancillary(avarlist [ , noconstant ])` specifies the variables used to model the ancillary parameter.

By default, the ancillary parameter does not depend on covariates. Specifying `ancillary(avarlist, noconstant)` causes the constant to be suppressed in the model for the ancillary parameter.

`ancillary()` may be specified for the model for survival-time outcome, for the model for the censoring variable, or for both. If `ancillary()` is specified for both, the varlist used for each model may be different.

`noconstant`; see [R] [Estimation options](#).

Stat

stat is one of three statistics: *ate*, *atet*, or *pomeans*. *ate* is the default.

ate specifies that the average treatment effect be estimated.

atet specifies that the average treatment effect on the treated be estimated.

pomeans specifies that the potential-outcome means for each treatment level be estimated.

SE/Robust

vce (*vcetype*) specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (*robust*), that allow for intragroup correlation (*cluster* *clustvar*), and that use bootstrap or jackknife methods (*bootstrap*, *jackknife*); see [R] [vce_option](#).

Reporting

level (*#*); see [R] [Estimation options](#).

aequations specifies that the results for the outcome-model or treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

noshow prevents *stteffects ipwra* from showing the key *st* variables. This option is rarely used because most people type *stset*, *show* or *stset*, *noshow* to permanently set whether they want to see these variables mentioned at the top of the output of every *st* command; see [ST] [stset](#).

display_options: *nocl*, *novalues*, *noomitted*, *vsquish*, *noemptycells*, *baselevels*, *allbaselevels*, *nofvlabel*, *fvwrap* (*#*), *fvwrapon* (*style*), *cformat* (*%fmt*), *pformat* (*%fmt*), *sformat* (*%fmt*), and *nolstretch*; see [R] [Estimation options](#).

Maximization

maximize_options: *iterate* (*#*), *[no]log*, and *from* (*init_specs*); see [R] [Maximize](#). These options are seldom used.

init_specs is one of

matname [, *skip copy*]

[, *#...*], *copy*

iterinit (*#*) specifies the maximum number of iterations used to calculate the starting values. This option is seldom used.

Advanced

pstolerance (*#*) specifies the tolerance used to check the overlap assumption. The default value is *pstolerance* (*1e-5*). *stteffects* will exit with an error if an observation has an estimated propensity score smaller than that specified by *pstolerance* ().

osample (*newvar*) specifies that indicator variable *newvar* be created to identify observations that violate the overlap assumption.

control (*#* | *label*) specifies the level of *tvar* that is the control. The default is the first treatment level. You may specify the numeric level *#* (a nonnegative integer) or the label associated with the numeric level. *control* () may not be specified with the statistic *pomeans*. *control* () and *tlevel* () may not specify the same treatment level.

`tlevel(#|label)` specifies the level of *tvar* that is the treatment for the statistic *atet*. The default is the second treatment level. You may specify the numeric level *#* (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic *atet*. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `stteffects` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

If you are not familiar with the framework for treatment-effects estimation from observational survival-time data, please see [\[CAUSAL\] stteffects intro](#).

IPWRA estimators use estimated weights to obtain missingness-adjusted outcome-regression parameters. The missingness-adjusted outcome-regression parameters are used to compute averages of treatment-level predicted outcomes. Contrasts of these averages estimate the treatment effects.

The estimated weights account for the missing potential outcome and, optionally, for data lost to censoring. The weights are estimated using a treatment-assignment model and, optionally, a model for the censoring time. A term in the estimator for the outcome-regression parameters accounts for data lost to censoring when estimated weights are not used.

There are two versions of the IPWRA estimator because there are two methods of accounting for the data lost to censoring.

1. IPWRA estimators that adjust for censoring by including a term in the likelihood function for the outcome-model parameters are known as likelihood-adjusted-censoring IPWRA (LAC-IPWRA) estimators.
2. IPWRA estimators that adjust for censoring by weighting the likelihood function for the outcome-model parameters by estimated inverse-probability-of-censoring weights are known as weighted-adjusted-censoring IPWRA (WAC-IPWRA) estimators.

The LAC-IPWRA estimators require fewer assumptions than the WAC-IPWRA estimators. Outlining the steps performed by LAC-IPWRA and WAC-IPWRA estimators allows us to be more specific about the tradeoffs between the estimators.

LAC-IPWRA estimators use a three-step approach to estimating treatment effects:

1. Estimate the parameters of a treatment-assignment model and compute inverse-probability-of-treatment weights.
2. Obtain the treatment-specific predicted mean outcomes for each subject by using the weighted maximum likelihood estimators. Estimated inverse-probability-of-treatment weights are used to weight the maximum likelihood estimator. A term in the likelihood function adjusts for right-censored survival times.
3. Compute the means of the treatment-specific predicted mean outcomes. Contrasts of these averages provide the estimates of the ATETs. By restricting the computations of the means to the subset of treated subjects, we can obtain the ATETs.

WAC-IPWRA estimators use a four-step approach to estimating treatment effects:

1. Estimate the parameters of a treatment-assignment model and compute inverse-probability-of-treatment weights.
2. Estimate the parameters of a time-to-censoring model and compute inverse-probability-of-censoring weights.
3. Obtain the treatment-specific predicted mean outcomes for each subject by using the weighted maximum likelihood estimators. Estimated inverse-probability-of-treatment weights and inverse-probability-of-censoring weights are used to weight the maximum likelihood estimator. The inverse-probability-of-censoring weights account for right-censored survival times.
4. Compute the means of the treatment-specific predicted mean outcomes. Contrasts of these averages provide the estimates of the ATEs. By restricting the computations of the means to the subset of treated subjects, we can obtain the ATETs.

The WAC-IPWRA estimators require that the censoring time be random and that the time-to-censoring model be well specified. The implemented WAC-IPWRA estimators also require that the time-to-censoring process not vary by treatment level. The LAC-IPWRA estimators do not require these extra assumptions because they use a likelihood term instead of weights to adjust for the data lost to censoring.

Here we note only a few entry points to the vast literature on estimators that combine IPW and RA methods. [Hirano, Imbens, and Ridder \(2003\)](#), [Imbens \(2000, 2004\)](#), [Imbens and Wooldridge \(2009\)](#), [Rosenbaum and Rubin \(1983\)](#), [Robins and Rotnitzky \(1995, 2006\)](#), [Robins, Rotnitzky, and Zhao \(1995\)](#), [Wooldridge \(2002, 2007\)](#), [Cameron and Trivedi \(2005, chap. 25\)](#), [Wooldridge \(2010, chap. 21\)](#), and [Vittinghoff et al. \(2012, chap. 9\)](#) provide excellent general introductions to estimating ATEs and to the IPWRA estimators in particular.

Like `streg` and other survival-time commands, `stteffects ipwra` uses the outcome variable and the failure indicator computed by, and optionally weights specified with, `stset`. `stteffects ipwra` is not appropriate for data with time-varying covariates, also known as multiple-record survival-time data, or for delayed-entry data.

► Example 1: Estimating the ATE by LAC-IPWRA

Suppose we wish to study the effect of smoking on the time to a second heart attack among women aged 45–55 years. In our fictional `sheart` dataset, `atime` is the observed time in years to a second heart attack or censoring, and `fail` is the 0/1 indicator that a second heart attack was observed. (When `fail` is 1, `atime` records the time to the second heart attack; when `fail` is 0, `atime` records a censored observation of the time to a second heart attack.) We previously `stset` these data; see [A quick tour of the estimators](#) in [\[CAUSAL\] stteffects intro](#).

The treatment, smoking, is stored in the 0/1 indicator `smoke`. These data also contain age at the time of the first heart attack (`age`), and indices of the level of exercise (`exercise`), diet quality (`diet`), and education (`education`) prior to the first heart attack.

We can use `stteffects ipwra` to estimate the ATE. We model the mean survival time using the default Weibull model, controlling for age, exercise, diet, and education. We model treatment assignment using the default logit model with covariates age, exercise, and education. We do not specify a time-to-censoring model so that we obtain the LAC estimator.

```
. use https://www.stata-press.com/data/r19/sheart
(Time to second heart attack (fictional))

. stteffects ipwra (age exercise diet education) (smoke age exercise education)

      Failure _d: fail
      Analysis time _t: atime

Iteration 0:  EE criterion = 7.439e-15
Iteration 1:  EE criterion = 1.730e-26

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : IPW regression adjustment
Outcome model  : Weibull
Treatment model: logit
Censoring model: none
```

	_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE							
	smoke (Smoker vs Nonsmoker)	-1.591874	.4837332	-3.29	0.001	-2.539973	-.643774
POMean							
	smoke Nonsmoker	4.214263	.2598689	16.22	0.000	3.704929	4.723597

When every woman smoked in the population of women aged 45–55 years who have had a heart attack, the average time to a second heart attack is estimated to be 1.59 years less than when no women in the population of interest smoked. The estimated average time to a second heart attack when no women in the population of interest smoked is 4.21 years.

The ratio of the ATE to the control-level potential-outcome mean (POM) measures the importance of the effect. In this example, when all women smoked, the time to the second heart attack falls by an estimated 38% relative to the case in which no women smoked. See [example 3](#) in [\[CAUSAL\] stteffects ra](#) for an example that uses [nlcom](#) to compute a point estimate and a confidence interval for this ratio.



▶ Example 2: Different outcome and treatment models

Instead of a Weibull model for the outcome model, we could have used an exponential, a gamma, or a lognormal model. Instead of a logit model for the treatment assignment, we could have used a probit or a heteroskedastic probit model. This example uses a gamma model for the outcome and a probit model for the treatment assignment.

```
. stteffects ipwra (age exercise diet education, gamma)
> (smoke age exercise education, probit)

Failure _d: fail
Analysis time _t: atime
Iteration 0: EE criterion = 2.644e-13
Iteration 1: EE criterion = 2.153e-23
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : IPW regression adjustment
Outcome model   : gamma
Treatment model : probit
Censoring model : none
```

_t	Robust					
	Coefficient	std. err.	z	P> z	[95% conf. interval]	
ATE						
smoke						
(Smoker						
vs						
Nonsmoker)	-1.387303	.4786032	-2.90	0.004	-2.325348	-.4492583
POmean						
smoke						
Nonsmoker	3.97986	.2258474	17.62	0.000	3.537207	4.422512

The estimated ATE of -1.39 and control-level POM of 3.98 are similar to the values of -1.59 and 4.21 that we obtained in [example 1](#).

◀

➤ Example 3: Estimating the ATE by WAC-IPWRA

Rather than using LAC, we may want to specify a time-to-censoring model. We now use `steffects ipwra` to estimate the ATE by WAC-IPWRA. We use the same specification of the outcome and treatment models that we used in [example 1](#). However, now we specify a time-to-censoring model, using the default Weibull model with covariates age, exercise, diet, and education.

```
. steffects ipwra (age exercise diet education) (smoke age exercise education)
> (age exercise diet education)

Failure _d: fail
Analysis time _t: atime
Iteration 0: EE criterion = 3.673e-18
Iteration 1: EE criterion = 8.511e-31
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : IPW regression adjustment
Outcome model  : Weibull
Treatment model: logit
Censoring model: Weibull
```

_t	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
smoke						
(Smoker						
vs						
Nonsmoker)	-2.285057	.7318456	-3.12	0.002	-3.719448	-.8506656
POMean						
smoke						
Nonsmoker	4.385841	.6427521	6.82	0.000	3.12607	5.645612

The estimated ATE of -2.29 differs from the ATE of -1.59 estimated by LAC-IPWRA, but the estimates of the control-level POM are similar between the two models: 4.39 for the WAC compared with 4.21 for the LAC.



➤ Example 4: Estimating the ATET by LAC-IPWRA

Intuitively, the ATET measures the effect of the treatment on an at-risk subpopulation. Sometimes the subpopulation that gets the treatment defines such an at-risk subpopulation. The ATET has the added benefit that it can be estimated under weaker conditions than the ATE; see *Assumptions and tradeoffs* under *Remarks and examples* in [\[CAUSAL\] stteffects intro](#).

```
. stteffects ipwra (age exercise diet education)
> (smoke age exercise education), atet

      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 2.815e-19
Iteration 1:  EE criterion = 1.938e-31

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : IPW regression adjustment
Outcome model   : Weibull
Treatment model : logit
Censoring model : none
```

_t	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATET						
smoke						
(Smoker						
vs						
Nonsmoker)	-1.775107	.3437506	-5.16	0.000	-2.448846	-1.101368
POmean						
smoke						
Nonsmoker	4.062424	.2779877	14.61	0.000	3.517578	4.60727

When all women in the subpopulation smoked, the average time to a second heart attack is estimated to be 1.78 years less than when no women in the subpopulation of interest smoked. If no women in the subpopulation of interest smoked, the average time to a second heart attack is 4.06 years.



Stored results

`stteffects ipwra` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>stteffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(dead)</code>	<code>_d</code>
<code>e(depvar)</code>	<code>_t</code>
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>ipwra</code>
<code>e(omodel)</code>	outcome model: <code>weibull</code> , <code>exponential</code> , <code>gamma</code> , or <code>lognormal</code>
<code>e(tmodel)</code>	treatment model: <code>logit</code> , <code>probit</code> , or <code>hetprobit</code>
<code>e(cmodel)</code>	censoring model: <code>weibull</code> , <code>exponential</code> , <code>gamma</code> , or <code>lognormal</code> (if specified)
<code>e(stat)</code>	statistic estimated: <code>ate</code> , <code>atet</code> , or <code>pomeans</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vce</i> type specified in <code>vce()</code>
<code>e(vctype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b</code> <code>V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

Methods and formulas are presented under the following headings:

- Introduction*
- Regression-adjusted estimators*
- Weighted-adjusted-censoring assumptions*
- Weighted regression-adjusted estimators*
- Inverse-probability-weighted estimators*
 - Uncensored data*
 - Inverse-probability-weighted regression-adjustment estimators*
 - Weighted-adjusted-censoring IPWRA*
 - Likelihood-adjusted-censoring IPWRA*
 - Functional-form details*

Introduction

This section presents the methods and formulas used by the estimators implemented in `stteffects ra`, `stteffects wra`, `stteffects ipw`, and `stteffects ipwra`. This section assumes that you are familiar with the concepts and intuition from the estimators discussed in [\[CAUSAL\] teffects intro advanced](#).

Each of the estimators implemented in `stteffects` has a multistep logic but is implemented as one step by simultaneously solving the estimating equations that define each step. This one-step estimating-equation approach provides consistent point estimates and a consistent variance–covariance of the estimator (VCE); see [Newey \(1984\)](#), [Wooldridge \(2010\)](#), and [Drukker \(2014\)](#).

Survival-time treatment-effects estimators handle two types of missing data. First, only one of the potential outcomes is observed, as is standard in causal inference. Second, the potential outcome for the received treatment may be censored. The data missing because of censoring may be handled by an outcome model, a censoring model, or both, just like the data missing due to observing only one potential outcome.

□ Technical note

Delayed entry would be a third type of missing data. The left-truncation process caused by delayed entry would also need to be modeled to estimate ATE parameters. The estimators implement in `stteffects` do not allow for delayed entry because they do not have a method for modeling how the left-truncation process selects the sample, conditional on the covariates.

□

All the implemented estimators are combinations of regression-adjustment (RA) and inverse-probability-weighted (IPW) techniques. RA estimators use an outcome model to account for the missing potential outcome and for censoring. IPW estimators use models for treatment assignment and censoring to construct weights that account for the missing potential outcome and for censoring.

The remainder of this section provides technical details about how the estimators in `stteffects` were implemented. We provide details only for the two-treatment-level case to simplify the formulas. We provide outlines for how the extensions to the multiple-treatment-level case were implemented.

Regression-adjusted estimators

We begin with the RA estimators implemented in `stteffects ra`. The RA estimators have the following logic:

- RA1. For each treatment level $\tau \in \{0, 1\}$, estimate by maximum likelihood (ML) the parameters β_τ of a parametric model for the survival-time outcome t in which $F(t|\mathbf{x}, \tau, \beta_\tau)$ is the distribution of t conditional on covariates $\mathbf{x}$ and treatment level τ . Denote the estimates β_τ by $\widehat{\beta}_{\text{ra}, \tau}$.
- RA2. Use the estimated $\widehat{\beta}_{\text{ra}, \tau}$ and the functional form implied by $F(t|\mathbf{x}, \tau, \beta_\tau)$ to estimate the mean survival time, conditional on $\mathbf{x}$ and treatment level τ , for each sample observation, denoted by $\widehat{E}(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ra}, \tau})$. Conditional independence of the treatment and the survival-time potential outcomes ensures that $E(t|\mathbf{x}, \tau, \beta_\tau) = E(t_\tau|\mathbf{x}, \beta_\tau)$, where t_τ is the potential survival-time outcome corresponding to treatment level τ . Under correct model specification, sample averages of $\widehat{E}(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ra}, \tau})$ consistently estimate the POM for treatment level τ , denoted by POM_τ .
- RA3. A contrast of the estimated POMs estimates the ATE.

If estimating an ATET, step RA2 is modified to use only the treated observations when estimating the POMs. A contrast of these POMs then estimates the ATET.

The contribution of the i th observation to the log likelihood that is maximized in step RA1 is

$$L_{\text{ra}}(t_i, \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ra}, \tau}) = \varpi_i(\tau_i == \tau) \left[(1 - c_i) \ln\{f(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ra}, \tau})\} + c_i \ln\{1 - F(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ra}, \tau})\} \right] \quad (1)$$

where ϖ_i is the observation-level weight, c_i is the 0/1 indicator for whether the survival-time observation on person i was censored, and $f(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ra}, \tau})$ is the density corresponding to distribution $F(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ra}, \tau})$. The first term inside the curly braces in (1) accounts for the noncensored observations, and the second term inside the curly braces accounts for the censored observations.

The RA estimators for the POMs simultaneously solve estimating equations (2a) through (2d) for $\widehat{\beta}_{\text{ra},0}$, $\widehat{\beta}_{\text{ra},1}$, $\widehat{\text{POM}}_{\text{ra},0}$, and $\widehat{\text{POM}}_{\text{ra},1}$.

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ra}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ra},0}, F) = \mathbf{0} \quad (2a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ra}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ra},1}, F) = \mathbf{0} \quad (2b)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ra},0}) - \widehat{\text{POM}}_{\text{ra},0} \right\} = 0 \quad (2c)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ra},1}) - \widehat{\text{POM}}_{\text{ra},1} \right\} = 0 \quad (2d)$$

where

$\mathbf{s}_{\text{ra}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ra},0}, F) = \frac{\partial L_{\text{ra}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ra},0})}{\partial \widehat{\beta}_{\text{ra},0}}$ is the vector of score equations from the ML estimator for $\widehat{\beta}_{\text{ra},0}$ based on survival-time model F ,

$\mathbf{s}_{\text{ra}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ra},1}, F) = \frac{\partial L_{\text{ra}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ra},1})}{\partial \widehat{\beta}_{\text{ra},1}}$ is the vector of score equations from the ML estimator for $\widehat{\beta}_{\text{ra},1}$ based on survival-time model F ,

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ra},0})$ is the predicted mean survival time assuming treatment level 0 for observation i conditional on $\mathbf{x}_i$, and

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ra},1})$ is the predicted mean survival time assuming treatment level 1 for observation i conditional on $\mathbf{x}_i$.

The ATE is estimated by replacing (2d) with

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ra},1}) - \widehat{\text{POM}}_{\text{ra},0} - \widehat{\text{ATE}}_{\text{ra}} \right\} = 0 \quad (3)$$

and the ATET is estimated by replacing (2c) and (3) with

$$1/N_1 \sum_{i=1}^N \varpi_i (\tau_i == 1) \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ra},0}) - \widehat{\text{POM}}_{\text{ra},\text{cot},0} \right\} = 0$$

$$1/N_1 \sum_{i=1}^N \varpi_i (\tau_i == 1) \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ra},1}) - \widehat{\text{POM}}_{\text{ra},\text{cot},0} - \widehat{\text{ATET}}_{\text{ra}} \right\} = 0$$

where $N_1 = \sum_{i=1}^N (t_i == 1)$ and $\widehat{\text{POM}}_{\text{ra},\text{cot},0}$ is the estimated conditional-on-treatment POM for treatment level 0.

Asymptotic standard errors for estimating equation estimators, also known as exactly identified generalized method of moments estimators, are standard in the literature; see [Newey \(1984\)](#), [Newey and McFadden \(1994\)](#), [Tsiatis \(2006\)](#), and [Wooldridge \(2010\)](#). These standard errors always have a robust structure and have been generalized to cluster-robust standard errors (see [Wooldridge \[2010\]](#)).

The score equations and the functional form for the predicted mean survival time depend on the model for survival-time outcome F . We provide these details below, under [Functional-form details](#).

Weighted-adjusted-censoring assumptions

All estimators that permit you to model the time to censoring are subject to three assumptions:

1. The censoring time must be random.
2. The censoring time must be from a known distribution.
3. The distribution of the censoring time cannot vary by treatment level.

We call these three requirements the WAC assumptions. If the WAC assumptions are violated, you can use either an RA estimator or the LAC version of the IPWRA estimator.

□ Technical note

We now describe how the observed survival-time outcome t is generated from the random censoring time t_c , the received treatment τ , and the potential-outcome survival times t_0 and t_1 under the WAC assumptions. First, each potential outcome is either censored or not censored.

$$\begin{aligned}\tilde{t}_0 &= t_c(t_0 \geq t_c) + t_0\{1 - (t_0 \geq t_c)\} \\ \tilde{t}_1 &= t_c(t_1 \geq t_c) + t_1\{1 - (t_1 \geq t_c)\}\end{aligned}$$

Under the WAC assumptions, t_c is a random variable from a known distribution, and t_c does not vary by treatment level.

Next, the received treatment $\tau \in \{0, 1\}$ determines which, possibly censored, potential outcome is observed.

$$t = (1 - \tau)\tilde{t}_0 + \tau\tilde{t}_1$$

The 0/1 indicator for whether the observed t was censored, denoted by c , is given by

$$c = (1 - \tau)(t_0 \geq t_c) + \tau(t_1 \geq t_c)$$

□

Weighted regression-adjusted estimators

As is standard in the survival literature, the RA estimators account for censored survival times by adding a term to the log-likelihood function for censored observations [see (1)]. In contrast, weighted regression-adjustment (WRA) estimators use weights to account for censored observations and are subject to the [WAC assumptions](#).

Wooldridge (2007) and Lin (2000) derived estimators for the regression parameters that maximize a weighted objective function of the uncensored observations. Each observation-level weight is the inverse of the probability of not being censored. Like the RA estimators, the WRA estimators use averages of the predicted mean survival times to estimate treatment-effect parameters.

The WRA estimators have the following logic.

- WRA1. Estimate by ML the parameters γ of a parametric survival-time model for the time to censoring t_c , in which $F_c(t_c|\mathbf{w}, \gamma)$ is the distribution of t_c conditional on covariates $\mathbf{w}$. Note that the censoring process does not vary by treatment level and that we only observe t_c when the observed potential outcome was censored. Denote the estimates of γ by $\hat{\gamma}$.
- WRA2. For each treatment level $\tau \in \{0, 1\}$, estimate by weighted maximum likelihood (WML) the β_τ parameters of a parametric survival-time model, denoted by $F(t|\mathbf{x}, \tau, \beta_\tau)$, where t is the survival-time outcome and $\mathbf{x}$ are the covariates. The weights are the inverse of the estimated probabilities of not being censored, $1/\{1 - F_c(t_c|\mathbf{w}, \hat{\gamma})\}$, and only the uncensored observations are used. Denote the estimates of β_τ by $\hat{\beta}_{\text{wra}, \tau}$.
- WRA3. Use the estimated $\hat{\beta}_{\text{wra}, \tau}$ and the functional form implied by $F(t|\mathbf{x}, \tau, \beta_\tau)$ to estimate the mean survival time, conditional on $\mathbf{x}$ and treatment level τ , for each sample observation, denoted by $\hat{E}(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{wra}, \tau})$. Conditional independence of the treatment and the survival-time potential outcomes ensures that $E(t|\mathbf{x}, \tau, \beta_\tau) = E(t_\tau|\mathbf{x}, \beta_\tau)$, where t_τ is the potential survival-time outcome corresponding to treatment level τ . Under correct model specification, sample averages of $\hat{E}(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{wra}, \tau})$ consistently estimate the POM for treatment level τ , denoted by POM_τ .
- WRA4. A contrast of the estimated POMs estimates the ATE.

If estimating an ATET, step WRA3 is modified to use only the treated observations when estimating the POMs. A contrast of these POMs then estimates the ATET.

The contribution of the i th observation to the log likelihood that is maximized in step WRA1 is

$$L_{c, \text{wra}}(t_i, \mathbf{w}_i, \hat{\gamma}) = \varpi_i [c_i \ln\{f_c(t_i|\mathbf{w}_i, \hat{\gamma})\} + (1 - c_i) \ln\{1 - F_c(t_i|\mathbf{w}_i, \hat{\gamma})\}] \quad (4)$$

where ϖ_i is the observation-level weight, c_i is the 0/1 indicator for whether the survival-time observation on person i was censored, t_i is the observed failure time, and $f_c(t_i|\mathbf{w}_i, \hat{\gamma})$ is the density corresponding to conditional time-to-censoring distribution $F_c(t_i|\mathbf{w}_i, \hat{\gamma})$. When $c_i = 1$, t_i is the time to censoring. When $c_i = 0$, the censoring time is not observed; we only know that it is greater than the observed t_i . The first term accounts for the observations in which t_i is observed to be the censoring time, and the second term accounts for the observations in which the censoring time is greater than the observed t_i .

The contribution of the i th observation to the log likelihood that is maximized in step WRA2 is

$$L_{\text{wra}}(t_i, \mathbf{x}_i, \tau, \hat{\beta}_{\text{wra}, \tau}) = \varpi_i (\tau_i == \tau) \left[\frac{(1 - c_i)}{\{1 - F_c(t_i|\mathbf{w}_i, \hat{\gamma})\}} \right] \ln\{f(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{wra}, \tau})\} \quad (5)$$

where $f(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{wra}, \tau})$ is the density corresponding to distribution $F(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{wra}, \tau})$. Equation (5) does not contain a term that adjusts for censoring; see (1) for a comparison. Rather, it uses inverse-probability weights to account for both the censored and the uncensored observations.

The WRA estimators for the POMs simultaneously solve estimating equations (6a) through (6e) for $\widehat{\gamma}$, $\widehat{\beta}_{\text{wra},0}$, $\widehat{\beta}_{\text{wra},1}$, $\widehat{\text{POM}}_{\text{wra},0}$, and $\widehat{\text{POM}}_{\text{wra},1}$.

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{wra}}(t_i, \mathbf{w}_i, \widehat{\gamma}, F_c) = \mathbf{0} \quad (6a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{wra}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{wra},0}, F) = \mathbf{0} \quad (6b)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{wra}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{wra},1}, F) = \mathbf{0} \quad (6c)$$

$$1/N \sum_{i=1}^N \varpi_i \{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{wra},0}) - \widehat{\text{POM}}_{\text{wra},0} \} = 0 \quad (6d)$$

$$1/N \sum_{i=1}^N \varpi_i \{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{wra},1}) - \widehat{\text{POM}}_{\text{wra},1} \} = 0 \quad (6e)$$

where

$\mathbf{s}_{\text{wra}}(t_i, \mathbf{w}_i, \widehat{\gamma}, F_c) = \frac{\partial L_{c,\text{wra}}(t_i, \mathbf{w}_i, \widehat{\gamma})}{\partial \widehat{\gamma}}$ is the vector of score equations from the ML estimator for $\widehat{\gamma}$ based on survival-time model F_c ,

$\mathbf{s}_{\text{wra}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{wra},0}, F) = \frac{\partial L(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{wra},0})}{\partial \widehat{\beta}_{\text{wra},0}}$ is the vector of score equations from the WML estimator for $\widehat{\beta}_{\text{wra},0}$ based on survival-time model F ,

$\mathbf{s}_{\text{wra}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{wra},1}, F) = \frac{\partial L(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{wra},1})}{\partial \widehat{\beta}_{\text{wra},1}}$ is the vector of score equations from the WML estimator for $\widehat{\beta}_{\text{wra},1}$ based on survival-time model F ,

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{wra},0})$ is the predicted mean survival time assuming treatment level 0 for observation i conditional on $\mathbf{x}_i$, and

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{wra},1})$ is the predicted mean survival time assuming treatment level 1 for observation i conditional on $\mathbf{x}_i$.

The observation-level scores $\mathbf{s}_{\text{wra}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{wra},0}, F)$ and $\mathbf{s}_{\text{wra}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{wra},1}, F)$ also depend on c_i , $\mathbf{w}_i$, $\widehat{\gamma}$, and F_c , but we ignored this dependence to simplify the notation.

The ATE is estimated by replacing (6e) with

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{wra},1}) - \widehat{\text{POM}}_{\text{wra},0} - \widehat{\text{ATE}}_{\text{wra}} \right\} = 0 \quad (7)$$

and the ATET is estimated by replacing (6e) and (7) with

$$\begin{aligned} 1/N_1 \sum_{i=1}^N \varpi_i (\tau_i == 1) \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{wra},0}) - \widehat{\text{POM}}_{\text{wra},\text{cot},0} \right\} &= 0 \\ 1/N_1 \sum_{i=1}^N \varpi_i (\tau_i == 1) \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{wra},1}) - \widehat{\text{POM}}_{\text{wra},\text{cot},0} - \widehat{\text{ATET}}_{\text{wra}} \right\} &= 0 \end{aligned}$$

where $\widehat{\text{POM}}_{\text{wra},\text{cot},0}$ is the estimated conditional-on-treatment POM.

Inverse-probability-weighted estimators

IPW estimators are weighted averages of the observed outcome. The weights correct for missing data due to unobserved potential outcomes and censoring. Each weight is the inverse of the probability that a given value is observed. Observed values that were not likely to be observed have higher weights.

When the outcome variable is never censored, the missing data are the unobserved potential outcome and an observation's weight is the inverse of a treatment probability. When the outcome may be censored, the censoring is an additional source of missing data. In this case, an observation's weight is the inverse of the joint probability that an observation is uncensored and has a particular treatment level.

To define this joint probability, the censoring time must be random. In practice, we make the [WAC assumptions](#).

As is standard in the survival-time literature, we assume that the censoring-time process is independent of treatment assignment after conditioning on the covariates. This conditional independence assumption implies that the probability that observation i receives treatment level 1 and is not censored is the product of the probability that i gets treatment level 1 and the probability that i is not censored at time t_i , which we denote by

$$p(\mathbf{z}_i, \alpha) \{1 - F_c(t_i | \mathbf{w}_i, \gamma)\}$$

where

$p(\mathbf{z}_i, \alpha)$ is the modeled probability that i gets treatment level 1, conditional on covariates $\mathbf{z}_i$ with parameters α , and

$F_c(t_i | \mathbf{w}_i, \gamma)$ is the survival-time model for the censoring time, conditional on covariates $\mathbf{w}_i$ with parameters γ , and evaluated at time t_i .

[Bai, Tsiatis, and O'Brien \(2013\)](#) formally derive these weights to control jointly for the missing potential outcome and censoring.

The IPW estimators have the following logic.

- IPW1. Estimate by ML the parameters γ of a parametric survival-time model for the time to censoring, in which $F_c(t_c | \mathbf{w}, \gamma)$ is the distribution of censoring time, conditional on covariates $\mathbf{w}$. Denote the estimates of γ by $\widehat{\gamma}$.
- IPW2. Estimate by ML the parameters α of a parametric model for the probability of treatment model $p(\mathbf{z}_i, \alpha)$. Denote the estimates of α by $\widehat{\alpha}$.

IPW3. Use the $\widehat{\gamma}$ estimated in IPW1 and the $\widehat{\alpha}$ estimated in IPW2 to construct inverse-probability weights by (8a) for treatment level 1 and by (8b) for treatment level 0.

$$\omega_{i,1} = \frac{(\tau_i == 1)(c_i == 0)}{p(\mathbf{z}_i, \widehat{\alpha})\{1 - F_c(t_i|\mathbf{w}_i, \widehat{\gamma})\}} \quad (8a)$$

$$\omega_{i,0} = \frac{(\tau_i == 0)(c_i == 0)}{[\{1 - p(\mathbf{z}_i, \widehat{\alpha})\}\{1 - F_c(t_i|\mathbf{w}_i, \widehat{\gamma})\}]} \quad (8b)$$

IPW4. Use the estimated weights to estimate each POM_{τ} by a weighted average of the uncensored observations on the observed potential outcome.

The contribution of the i th observation to the log likelihood that is maximized in [step IPW1](#) is

$$L_{c,\text{ipw}}(t_i, \mathbf{w}_i, \widehat{\gamma}) = \varpi_i [c_i \ln\{f_c(t_i|\mathbf{w}_i, \widehat{\gamma})\} + (1 - c_i) \ln\{1 - F_c(t_i|\mathbf{w}_i, \widehat{\gamma})\}]$$

where the definitions and intuition are as described after (4).

The contribution of the i th observation to the log likelihood that is maximized in [step IPW2](#) is

$$L_{p,\text{ipw}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}) = \varpi_i [(\tau_i == 1) \ln\{p(\mathbf{z}_i, \widehat{\alpha})\} + \{1 - (\tau_i == 1)\} \ln\{1 - p(\mathbf{z}_i, \widehat{\alpha})\}]$$

where $p(\mathbf{z}_i, \widehat{\alpha})$ is the model for the probability that i gets treatment level 1.

The IPW estimators for the POMs simultaneously solve estimating equations (9a) through (9d) for $\widehat{\gamma}$, $\widehat{\alpha}$, $\widehat{\text{POM}}_{\text{ipw},0}$, and $\widehat{\text{POM}}_{\text{ipw},1}$.

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipw}}(t_i, \mathbf{w}_i, \widehat{\gamma}, F_c) = \mathbf{0} \quad (9a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipw}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \mathbf{0} \quad (9b)$$

$$1/N \sum_{i=1}^N \varpi_i \omega_{i,0} (t_i - \widehat{\text{POM}}_{\text{ipw},0}) = 0 \quad (9c)$$

$$1/N \sum_{i=1}^N \varpi_i \omega_{i,1} (t_i - \widehat{\text{POM}}_{\text{ipw},1}) = 0 \quad (9d)$$

where

$\mathbf{s}_{\text{ipw}}(t_i, \mathbf{w}_i, \widehat{\gamma}, F_c) = \frac{\partial L_{c,\text{ipw}}(t_i, \mathbf{w}_i, \widehat{\gamma})}{\partial \widehat{\gamma}}$ is the vector of score equations from the ML estimator for $\widehat{\gamma}$ based on survival-time model F_c , and

$\mathbf{s}_{\text{ipw}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \frac{\partial L_{p,\text{ipw}}(\tau_i, \mathbf{z}_i, \widehat{\alpha})}{\partial \widehat{\alpha}}$ is the vector of score equations from the ML estimator for $\widehat{\alpha}$ based on probability model p .

The literature on IPW estimators discusses using normalized versus unnormalized weights, with normalized weights doing better in simulation studies; see [Busso, DiNardo, and McCrary \(2014\)](#) for example. The way that weights enter moment equations (9c) and (9d) implies that they are normalized, because the scale of the weights does not affect the estimates.

The estimated ATE is computed as

$$\widehat{\text{POM}}_{\text{ipw},1} - \widehat{\text{POM}}_{\text{ipw},0} = \widehat{\text{ATE}}_{\text{ipw}}$$

The estimated ATET uses weights

$$\omega_{i,\text{cot},1} = \frac{(\tau_i == 1)(c_i == 0)}{[\{1 - F_c(t_i | \mathbf{w}_i, \hat{\gamma})\}]} \quad (10a)$$

for treatment level 1 and

$$\omega_{i,\text{cot},0} = \frac{p(\mathbf{z}_i, \hat{\alpha})(\tau_i == 0)(c_i == 0)}{[\{1 - p(\mathbf{z}_i, \hat{\alpha})\}\{1 - F_c(t_i | \mathbf{w}_i, \hat{\gamma})\}]} \quad (10b)$$

for treatment level 0, and replaces (9c) and (9d) with

$$1/N_1 \sum_{i=1}^N \varpi_i \omega_{i,\text{cot},0} (t_i - \widehat{\text{POM}}_{\text{ipw},\text{cot},0}) = 0 \quad (11a)$$

$$1/N_1 \sum_{i=1}^N \varpi_i \omega_{i,\text{cot},1} (t_i - \widehat{\text{POM}}_{\text{ipw},\text{cot},1}) = 0 \quad (11b)$$

and then computes

$$\widehat{\text{POM}}_{\text{ipw},\text{cot},1} - \widehat{\text{POM}}_{\text{ipw},\text{cot},0} = \widehat{\text{ATET}}_{\text{ipw}}$$

These IPW estimators can be viewed as weighted IPW estimators and are thus related to those in [Hirano, Imbens, and Ridder \(2003\)](#).

Uncensored data

As mentioned, when the outcome variable is never censored, the missing data are the unobserved potential outcome and an observation's weight is the inverse of a treatment probability. In the never-censored case, the IPW estimators are identical to those implemented in `teffects ipw`; see *IPW estimators* under *Methods and formulas* in `[CAUSAL] teffects aipw`.

`stteffects ipw` computes the estimator for never-censored data when a censoring model is not specified and there are no censored observations in the sample. In the never-censored case, the following changes are made to the IPW estimator for the POMs and the ATE.

1. `Step IPW1` is not performed.
2. The weights in (8a) and (8b) for the POMs and the ATE are replaced with (12a) for treatment level 1 and (12b) for treatment level 0.

$$\omega_{i,1} = \frac{(\tau_i == 1)}{p(\mathbf{z}_i, \hat{\alpha})} \quad (12a)$$

$$\omega_{i,0} = \frac{(\tau_i == 0)}{\{1 - p(\mathbf{z}_i, \hat{\alpha})\}} \quad (12b)$$

3. Only moment conditions (9b), (9c), and (9d) are used.

The following changes also are made to the IPW estimator for the ATET.

1. [Step IPW1](#) is not performed.
2. The weights in (10a) and (10b) are replaced with (13a) for treatment level 1 and (13b) for treatment level 0.

$$\omega_{i,\text{cot},1} = (\tau_i == 1) \quad (13a)$$

$$\omega_{i,\text{cot},0} = \frac{p(\mathbf{z}_i, \hat{\alpha})(\tau_i == 0)}{\{1 - p(\mathbf{z}_i, \hat{\alpha})\}} \quad (13b)$$

3. Only moment conditions (9b) (11a), and (11b) are used.

Inverse-probability-weighted regression-adjustment estimators

IPWRA estimators are averages of treatment-specific predicted conditional means that were made using missingness-adjusted regression parameters. These estimators are Wooldridge's IPWRA for survival-time outcomes; see [Wooldridge \(2010, chap. 21\)](#) and [Wooldridge \(2007\)](#).

The censored observations can be handled either by weighting under the WAC assumptions to obtain the WAC-IPWRA estimator or by adding a term to the log-likelihood function (which we call likelihood-adjusted censoring) to obtain the LAC-IPWRA estimator. Correspondingly, there are two versions of formulas for the IPWRA estimator.

1. When a censoring model is specified, `stteffects ipwra` uses the formulas for the WAC-IPWRA estimator given in [Weighted-adjusted-censoring IPWRA](#).
2. When a censoring model is not specified, `stteffects ipwra` uses the formulas for the LAC-IPWRA given in [Likelihood-adjusted-censoring IPWRA](#), below.

The WAC-IPWRA estimator requires that some observations be censored and that the WAC assumptions hold; see [Weighted-adjusted-censoring assumptions](#), above. The LAC-IPWRA estimator handles the case in which no observations are censored and requires the weaker independent censoring assumptions, which allows for fixed censoring times.

Weighted-adjusted-censoring IPWRA

When a censoring model is specified, `stteffects ipwra` uses the formulas for the WAC-IPWRA estimator to obtain the model-based weights that account for censoring. For notational conciseness and to reinforce its dependence on random censoring, we denote the WAC-IPWRA estimator by `IPWRAR` in lists and formulas. The WAC-IPWRA estimators have the following logic.

- IPWRAR1. Estimate by ML the parameters γ of a parametric survival-time model for the time to censoring, in which $F_c(t_c|\mathbf{w}, \gamma)$ is the censoring-time distribution, conditional on covariates $\mathbf{w}$. We denote the estimates of γ by $\hat{\gamma}$.
- IPWRAR2. Estimate by ML the parameters α of a parametric model for the probability of treatment model $p(\mathbf{z}_i, \alpha)$. We denote the estimates of α by $\hat{\alpha}$.
- IPWRAR3. For each treatment level $\tau \in \{0, 1\}$, estimate by WML the parameters β_τ of a parametric model for the survival-time outcome t , in which $F(t|\mathbf{x}, \tau, \beta_\tau)$ is the distribution of t conditional on covariates $\mathbf{x}$ and treatment level τ . For the ATE, the weights are those in equations (8a) and (8b). For the ATET, the weights are those in equations (10a) and (10b). We denote the estimates of $\beta_{\text{ipwrar}, \tau}$ by $\hat{\beta}_\tau$.
- IPWRAR4. Use the estimated $\hat{\beta}_{\text{ipwrar}, \tau}$ and the functional form implied by $F(t|\mathbf{x}, \tau, \beta_\tau)$ to estimate the mean survival time, conditional on $\mathbf{x}$ and treatment level τ , for each sample observation, denoted by $\hat{E}(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{ipwrar}, \tau})$. Conditional independence of the treatment and the survival-time potential outcomes ensures that $E(t|\mathbf{x}, \tau, \beta_\tau) = E(t_\tau|\mathbf{x}, \beta_\tau)$, where t_τ is the potential survival-time outcome corresponding to treatment level τ . Under correct model specification, sample averages of $\hat{E}(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{ipwrar}, \tau})$ consistently estimate the POM for treatment level τ , denoted by POM_τ .

The contribution of the i th observation to the log likelihood that is maximized in [step IPWRAR1](#) is

$$L_{c, \text{ipwrar}}(t_i, \mathbf{w}_i, \hat{\gamma}) = \varpi_i [c_i \ln\{f_c(t_i|\mathbf{w}_i, \hat{\gamma})\} + (1 - c_i) \ln\{1 - F_c(t_i|\mathbf{w}_i, \hat{\gamma})\}]$$

where the definitions and intuition are as described after (4).

The contribution of the i th observation to the log likelihood that is maximized in [step IPWRAR2](#) is

$$L_{p, \text{ipwrar}}(\tau_i, \mathbf{z}_i, \hat{\alpha}) = \varpi_i [(\tau_i == 1) \ln\{p(\mathbf{z}_i, \hat{\alpha})\} + \{1 - (\tau_i == 1)\} \ln\{1 - p(\mathbf{z}_i, \hat{\alpha})\}]$$

where $p(\mathbf{z}_i, \hat{\alpha})$ is the model for the probability that i gets treatment level 1.

The weights and the parameters in [step IPWRAR3](#) used to estimate the ATE differ from those used to estimate the ATET. For the ATE, the contribution of the i th observation to the log likelihood that is maximized in step IPWRAR3 is

$$L_{\text{ipwrar}}(t_i, \mathbf{x}_i, \tau, \hat{\beta}_{\text{ipwrar}, \text{ate}, \tau}) = \varpi_i \omega_{i, \tau} \ln\{f(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{ipwrar}, \text{ate}, \tau})\}$$

where $\omega_{i, 1}$ is given in (8a), $\omega_{i, 0}$ is given in (8b), and $f(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{ipwrar}, \text{ate}, \tau})$ is the density corresponding to distribution $F(t_i|\mathbf{x}_i, \tau, \hat{\beta}_{\text{ipwrar}, \text{ate}, \tau})$. Like WRA, only the uncensored observations are used because the weights account for censoring.

The IPWRAR estimators for the POMs simultaneously solve estimating equations (14a) through (14f) for $\widehat{\gamma}$, $\widehat{\alpha}$, $\widehat{\beta}_{\text{ipwrrar,ate},0}$, $\widehat{\beta}_{\text{ipwrrar,ate},0}$, $\widehat{\text{POM}}_{\text{ipwrrar},0}$, and $\widehat{\text{POM}}_{\text{ipwrrar},1}$.

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{w}_i, \widehat{\gamma}, F_c) = \mathbf{0} \quad (14a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrrar}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \mathbf{0} \quad (14b)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwrrar,ate},0}, F) = \mathbf{0} \quad (14c)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwrrar,ate},1}, F) = \mathbf{0} \quad (14d)$$

$$1/N \sum_{i=1}^N \varpi_i \{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ipwrrar,ate},0}) - \widehat{\text{POM}}_{\text{ipwrrar},0} \} = 0 \quad (14e)$$

$$1/N \sum_{i=1}^N \varpi_i \{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwrrar,ate},1}) - \widehat{\text{POM}}_{\text{ipwrrar},1} \} = 0 \quad (14f)$$

where

$\mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{w}_i, \widehat{\gamma}, F_c) = \frac{\partial L_{c,\text{ipwrrar}}(t_i, \mathbf{w}_i, \widehat{\gamma})}{\partial \widehat{\gamma}}$ is the vector of score equations from the ML estimator for $\widehat{\gamma}$ based on survival-time model F_c ,

$\mathbf{s}_{\text{ipwrrar}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \frac{\partial L_{p,\text{ipwrrar}}(\tau_i, \mathbf{z}_i, \widehat{\alpha})}{\partial \widehat{\alpha}}$ is the vector of score equations from the ML estimator for $\widehat{\alpha}$ based on probability model p ,

$\mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwrrar,ate},0}, F) = \frac{\partial L_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwrrar,ate},0})}{\partial \widehat{\beta}_{\text{ipwrrar,ate},0}}$ is the vector of score equations from the ML estimator for $\widehat{\beta}_{\text{ipwrrar,ate},0}$ based on survival-time model F ,

$\mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwrrar,ate},1}, F) = \frac{\partial L_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwrrar,ate},1})}{\partial \widehat{\beta}_{\text{ipwrrar,ate},1}}$ is the vector of score equations from the ML estimator for $\widehat{\beta}_{\text{ipwrrar,ate},1}$ based on survival-time model F ,

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ipwrrar,ate},0})$ is the predicted mean survival time assuming treatment level 0 for observation i conditional on $\mathbf{x}_i$, and

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwrrar,ate},1})$ is the predicted mean survival time assuming treatment level 1 for observation i conditional on $\mathbf{x}_i$.

The ATE is estimated by replacing (14f) with

$$1/N \sum_{i=1}^N \varpi_i \{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwrrar,ate},1}) - \widehat{\text{POM}}_{\text{ipwrrar},0} - \widehat{\text{ATE}}_{\text{ipwrrar}} \} = 0$$

For the ATET, the contribution of the i th observation to the weighted log likelihood that is maximized in step IPWRAR3 is

$$L_{\text{ipwrrar}}(t_i, \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwrrar,atet},\tau}) = \varpi_i \omega_{i,\text{cot},\tau}(\tau_i == \tau) \ln\{f(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwrrar,atet},\tau})\}$$

where $\omega_{i,\text{cot},1}$ is given in (10a), $\omega_{i,\text{cot},0}$ is given in (10b), and $f(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwrrar,atet},\tau})$ is the density corresponding to distribution $F(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwrrar,atet},\tau})$.

The WAC-IPWRA estimators for the conditional-on-treatment POMs simultaneously solve estimating equations (15a) through (15f) for $\widehat{\beta}_{\text{ipwrrar,atet},0}$, $\widehat{\beta}_{\text{ipwrrar,atet},1}$, $\widehat{\gamma}$, $\widehat{\alpha}$, $\widehat{\text{POM}}_{\text{ipwrrar,cot},0}$, and $\widehat{\text{POM}}_{\text{ipwrrar,cot},1}$.

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{w}_i, \widehat{\gamma}, F_c) = \mathbf{0} \quad (15a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrrar}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \mathbf{0} \quad (15b)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwrrar,atet},0}, F) = \mathbf{0} \quad (15c)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwrrar,atet},1}, F) = \mathbf{0} \quad (15d)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i|\mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ipwrrar,atet},0}) - \widehat{\text{POM}}_{\text{ipwrrar,cot},0} \right\} = 0 \quad (15e)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i|\mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwrrar,atet},1}) - \widehat{\text{POM}}_{\text{ipwrrar,cot},1} \right\} = 0 \quad (15f)$$

where

$\mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{w}_i, \widehat{\gamma}, F_c) = \frac{\partial L_{c,\text{ipwrrar}}(t_i, \mathbf{w}_i, \widehat{\gamma})}{\partial \widehat{\gamma}}$ is the vector of score equations from the ML estimator for $\widehat{\gamma}$ based on survival-time model F_c ,

$\mathbf{s}_{\text{ipwrrar}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \frac{\partial L_{p,\text{ipwrrar}}(\tau_i, \mathbf{z}_i, \widehat{\alpha})}{\partial \widehat{\alpha}}$ is the vector of score equations from the ML estimator for $\widehat{\alpha}$ based on probability model p ,

$\mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwrrar,atet},0}, F) = \frac{\partial L_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwrrar,atet},0})}{\partial \widehat{\beta}_{\text{ipwrrar,atet},0}}$ is the vector of score equations from the WML estimator for $\widehat{\beta}_{\text{ipwrrar,atet},0}$ based on survival-time model F ,

$\mathbf{s}_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwrrar,atet},1}, F) = \frac{\partial L_{\text{ipwrrar}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwrrar,atet},1})}{\partial \widehat{\beta}_{\text{ipwrrar,atet},1}}$ is the vector of score equations from the WML estimator for $\widehat{\beta}_{\text{ipwrrar,atet},1}$ based on survival-time model F ,

$\widehat{E}(t_i|\mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ipwrrar,atet},0})$ is the predicted mean survival time assuming treatment level 0 for observation i conditional on $\mathbf{x}_i$, and

$\widehat{E}(t_i|\mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwrrar,atet},1})$ is the predicted mean survival time assuming treatment level 1 for observation i conditional on $\mathbf{x}_i$.

The ATET is estimated by replacing (15f) with

$$\frac{1}{N} \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwra}, \text{atet}, 1}) - \widehat{\text{POM}}_{\text{ipwra}, \text{cot}, 0} - \widehat{\text{ATET}}_{\text{ipwra}} \right\} = 0$$

Likelihood-adjusted-censoring IPWRA

When a censoring model is not specified, `stteffects ipwra` uses the formulas for the LAC-IPWRA estimator that add a term to the log-likelihood function. For notational conciseness and to reinforce its use of an additional term in the log likelihood, we denote the LAC-IPWRA estimator by IPWRAL in lists and formulas.

The methods and formulas for the LAC-IPWRA estimator differ in three ways from those for the WAC-IPWRA estimator.

1. No censoring model is specified, so LAC-IPWRA does not perform a version of [step IPWRAR1](#) and it does not use the moment equations (14a).
2. The weights only depend on the treatment level and treatment assignment probabilities, not on the censoring.
3. The WML estimator for β_τ includes a term for censored observations and censored observations are used. Recall that for the WAC-IPWRA estimator, the weights used in the WML estimator for β_τ account for the censoring, and the censored observations are not used in the WML estimator.

The LAC-IPWRA estimators have the following logic.

- IPWRAL1. Estimate by ML the parameters α of a parametric model for the probability of treatment model $p(\mathbf{z}_i, \alpha)$.
- IPWRAL2. For each treatment level $\tau \in \{0, 1\}$, estimate by WML the parameters β_τ of a parametric model for the survival-time outcome t in which $F(t | \mathbf{x}, \tau, \beta_\tau)$ is the distribution of t conditional on covariates $\mathbf{x}$ and treatment level τ . The weights depend only on the treatment level and the treatment-assignment probabilities. For the ATE, the weights are those in (12a) and (12b). For the ATET, the weights are those in (13a) and (13b). We denote the estimates of β_τ by $\widehat{\beta}_{\text{ipwral}, \tau}$.
- IPWRAL3. Use the estimated $\widehat{\beta}_{\text{ipwral}, \tau}$ and the functional form implied by $F(t | \mathbf{x}, \tau, \beta_\tau)$ to estimate the mean survival time, conditional on $\mathbf{x}$ and treatment level τ , for each sample observation, denoted by $\widehat{E}(t_i | \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral}, \tau})$. Conditional independence of the treatment and the survival-time potential outcomes ensures that $E(t | \mathbf{x}, \tau, \beta_\tau) = E(t_\tau | \mathbf{x}, \beta_\tau)$, where t_τ is the potential survival-time outcome corresponding to treatment level τ . Under correct model specification, sample averages of $\widehat{E}(t_i | \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral}, \tau})$ consistently estimate the POM for treatment level τ , denoted by POM_τ .

The contribution of the i th observation to the log likelihood that is maximized in [step IPWRAL1](#) is

$$L_{p, \text{ipwral}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}) = \varpi_i \{ (\tau_i == 1) \ln\{p(\mathbf{z}_i, \widehat{\alpha})\} + \{1 - (\tau_i == 1)\} \ln\{1 - p(\mathbf{z}_i, \widehat{\alpha})\} \}$$

where $p(\mathbf{z}_i, \widehat{\alpha})$ is the model for the probability that i gets treatment level 1.

The weights and the parameters in [step IPWRA2](#) used to estimate the ATE differ from those used to estimate the ATET. For the ATE, the contribution of the i th observation to the log likelihood that is maximized in step IPWRA2 is

$$L_{\text{ipwral}}(t_i, \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,ate},\tau}) = (\tau_i == \tau) \varpi_i \omega_{i,\tau} \left\{ (1 - c_i) \ln\{f(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,ate},\tau})\} \right. \\ \left. c_i \ln\{1 - F(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,ate},\tau})\} \right\}$$

where $\omega_{i,1}$ is given in (12a), $\omega_{i,0}$ is given in (12b), and $f(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,ate},\tau})$ is the density corresponding to distribution $F(t_i|\mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,ate},\tau})$. Unlike the WRA estimator, the censored observations are used, and there is a term in the likelihood function to account for censoring.

The LAC-IPWRA estimators for the POMs simultaneously solve estimating equations (16a) through (16e) for $\widehat{\alpha}$, $\widehat{\beta}_{\text{ipwral,ate},0}$, $\widehat{\text{POM}}_{\text{ipwral},0}$, and $\widehat{\text{POM}}_{\text{ipwral},1}$.

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwral}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \mathbf{0} \quad (16a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwral,ate},0}, F) = \mathbf{0} \quad (16b)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwral,ate},1}, F) = \mathbf{0} \quad (16c)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i|\mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ipwral,ate},0}) - \widehat{\text{POM}}_{\text{ipwral},0} \right\} = 0 \quad (16d)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i|\mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwral,ate},1}) - \widehat{\text{POM}}_{\text{ipwral},1} \right\} = 0 \quad (16e)$$

where

$\mathbf{s}_{\text{ipwral}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \frac{\partial L_{p,\text{ipwral}}(\tau_i, \mathbf{z}_i, \widehat{\alpha})}{\partial \widehat{\alpha}}$ is the vector of score equations from the ML estimator for $\widehat{\alpha}$ based on probability model p ,

$\mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwral,ate},0}, F) = \frac{\partial L_{\text{ipwral}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwral,ate},0})}{\partial \widehat{\beta}_{\text{ipwral,ate},0}}$ is the vector of score equations from the WML estimator for $\widehat{\beta}_{\text{ipwral,ate},0}$ based on survival-time model F ,

$\mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwral,ate},1}, F) = \frac{\partial L_{\text{ipwral}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwral,ate},1})}{\partial \widehat{\beta}_{\text{ipwral,ate},1}}$ is the vector of score equations from the WML estimator for $\widehat{\beta}_{\text{ipwral,ate},1}$ based on survival-time model F ,

$\widehat{E}(t_i|\mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ipwral,ate},0})$ is the predicted mean survival time assuming treatment level 0 for observation i conditional on $\mathbf{x}_i$, and

$\widehat{E}(t_i|\mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwral,ate},1})$ is the predicted mean survival time assuming treatment level 1 for observation i conditional on $\mathbf{x}_i$.

The ATE is estimated by replacing (16e) with

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i|\mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwral,ate},1}) - \widehat{\text{POM}}_{\text{ipwral},0} - \widehat{\text{ATE}}_{\text{ipwral}} \right\} = 0$$

For the ATET, the contribution of the i th observation to the WML function that is maximized in [step IPWRAL2](#) is

$$L_{\text{ipwral}}(t_i, \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,atet},\tau}) = (\tau_i == \tau) \varpi_i \omega_{i,\text{cot},\tau} \left\{ (1 - c_i) \ln \{ f(t_i | \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,atet},\tau}) \} \right. \\ \left. c_i \ln \{ 1 - F(t_i | \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,atet},\tau}) \} \right\}$$

where $\omega_{i,\text{cot},1}$ is given in (13a), $\omega_{i,\text{cot},0}$ is given in (13b), and $f(t_i | \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,atet},\tau})$ is the density corresponding to distribution $F(t_i | \mathbf{x}_i, \tau, \widehat{\beta}_{\text{ipwral,atet},\tau})$. Again unlike the WRA, the censored observations are used, and there is a term in the likelihood function to account for censoring.

The LAC-IPWRA estimators for the conditional-on-treatment POMs simultaneously solve estimating equations (17a) through (17e) for $\widehat{\alpha}$, $\widehat{\beta}_{\text{ipwral,atet},0}$, $\widehat{\beta}_{\text{ipwral,atet},1}$, $\widehat{\text{POM}}_{\text{ipwral,cot},0}$, and $\widehat{\text{POM}}_{\text{ipwral,cot},1}$.

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwral}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \mathbf{0} \quad (17a)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwral,atet},0}, F) = \mathbf{0} \quad (17b)$$

$$1/N \sum_{i=1}^N \mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwral,atet},1}, F) = \mathbf{0} \quad (17c)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ipwral,atet},0}) - \widehat{\text{POM}}_{\text{ipwral,cot},0} \right\} = 0 \quad (17d)$$

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwral,atet},1}) - \widehat{\text{POM}}_{\text{ipwral,cot},1} \right\} = 0 \quad (17e)$$

where

$\mathbf{s}_{\text{ipwral}}(\tau_i, \mathbf{z}_i, \widehat{\alpha}, p) = \frac{\partial L_{p,\text{ipwral}}(\tau_i, \mathbf{z}_i, \widehat{\alpha})}{\partial \widehat{\alpha}}$ is the vector of score equations from the ML estimator for $\widehat{\alpha}$ based on probability model p ,

$\mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwral,atet},0}, F) = \frac{\partial L_{\text{ipwral}}(t_i, \mathbf{x}_i, 0, \widehat{\beta}_{\text{ipwral,atet},0})}{\partial \widehat{\beta}_{\text{ipwral,atet},0}}$ is the vector of score equations from the WML estimator for $\widehat{\beta}_{\text{ipwral,atet},0}$ based on survival-time model F ,

$\mathbf{s}_{\text{ipwral}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwral,atet},1}, F) = \frac{\partial L_{\text{ipwral}}(t_i, \mathbf{x}_i, 1, \widehat{\beta}_{\text{ipwral,atet},1})}{\partial \widehat{\beta}_{\text{ipwral,atet},1}}$ is the vector of score equations from the WML estimator for $\widehat{\beta}_{\text{ipwral,atet},1}$ based on survival-time model F ,

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 0, \widehat{\beta}_{\text{ipwral,atet},0})$ is the predicted mean survival time assuming treatment level 0 for observation i conditional on $\mathbf{x}_i$, and

$\widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwral,atet},1})$ is the predicted mean survival time assuming treatment level 1 for observation i conditional on $\mathbf{x}_i$.

The ATET is estimated by replacing (17e) with

$$1/N \sum_{i=1}^N \varpi_i \left\{ \widehat{E}(t_i | \mathbf{x}_i, \tau = 1, \widehat{\beta}_{\text{ipwral,atet},1}) - \widehat{\text{POM}}_{\text{ipwral,cot},0} - \widehat{\text{ATET}}_{\text{ipwral}} \right\} = 0$$

Functional-form details

In this section, we specify the functional forms for the conditional distribution function used in the survival-time outcome model F , the conditional distribution function used in the survival-time censoring model F_c , and the conditional distribution used to model the treatment probabilities p .

You may choose among the same set of conditional distribution functions for either F or F_c : `exponential`, `weibull`, `lnormal`, or `gamma`.

Name	Cumulative	Density	Mean
exponential	$1 - \exp(-\lambda_i t_i)$	$\lambda_i \exp(-\lambda_i t_i)$	$1/\lambda_i$
Weibull	$1 - \exp\{-(\lambda_i t_i)^{s_i}\}$	$s_i t_i^{s_i-1} \lambda_i \exp\{-(\lambda_i t_i)^{s_i}\}$	$(1/\lambda_i) \Gamma\{(s_i + 1)/s_i\}$
log normal	$\Phi\{(\ln(t_i) - \lambda_i)/s_i\}$	$(1/(s_i t_i)) \phi\{(\ln(t_i) - \lambda_i)/s_i\}$	$\exp(\lambda_i + s_i^2/2)$
gamma	$\text{gammap}\{s_i, (s_i t_i/\lambda_i)\}$	$(s_i^{s_i} t_i^{s_i-1}) / \{\lambda_i^{s_i} \Gamma(s_i)\} \exp(-s_i t_i/\lambda_i)$	λ_i

where the following table specifies how λ_i and s_i are parameterized in terms of the covariates $\mathbf{x}_i$ and the ancillary covariates $\tilde{\mathbf{x}}_i$, respectively.

Name	λ_i	s_i
exponential	$\exp(-\mathbf{x}_i \boldsymbol{\beta})$	
Weibull	$\exp(-\mathbf{x}_i \boldsymbol{\beta})$	$\exp(\tilde{\mathbf{x}}_i \tilde{\boldsymbol{\beta}})$
log normal	$\mathbf{x}_i \boldsymbol{\beta}$	$\exp(\tilde{\mathbf{x}}_i \tilde{\boldsymbol{\beta}})$
gamma	$\exp(\mathbf{x}_i \boldsymbol{\beta})$	$\exp(-2\tilde{\mathbf{x}}_i \tilde{\boldsymbol{\beta}})$

For the treatment-assignment models, the `probit` model uses the standard normal distribution, the `logit` uses the standard logistic distribution, the `hetprobit` model uses

$$\Phi\{\mathbf{z}_1 \boldsymbol{\alpha}_1 / \exp(\mathbf{z}_2 \boldsymbol{\alpha}_2)\}$$

and the multinomial logit uses

$$p(\mathbf{z}, t) = \exp(\mathbf{z} \boldsymbol{\alpha}_t) / \{1 + \sum_{k=1}^q \exp(\mathbf{z} \boldsymbol{\alpha}_k)\}$$

where the notation is defined below.

In the `hetprobit` model, $\mathbf{z}_1$ are the covariates specified in the treatment-assignment specification, $\mathbf{z}_2$ are the covariates specified in the `hetprobit()` option, and $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ are the corresponding coefficients.

In the multinomial logit model, $\mathbf{z}$ are the covariates specified in the treatment-assignment specification and $\boldsymbol{\alpha}_k$ are the coefficients; see [R] [mlogit](#) for further details.

References

- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Bai, X., A. A. Tsiatis, and S. M. O'Brien. 2013. Doubly robust estimators of treatment-specific survival distributions in observational studies with stratified sampling. *Biometrics* 69: 830–839. <https://doi.org/10.1111/biom.12076>.

- Busso, M., J. DiNardo, and J. McCrary. 2014. New evidence on the finite sample properties of propensity score reweighting and matching estimators. *Review of Economics and Statistics* 96: 885–897. https://doi.org/10.1162/REST_a_00431.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Drukker, D. M. 2014. Using gmm to solve two-step estimation problems. *The Stata Blog: Not Elsewhere Classified*. <https://blog.stata.com/2014/12/08/using-gmm-to-solve-two-step-estimation-problems/>.
- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: Sage.
- Hirano, K., G. W. Imbens, and G. Ridder. 2003. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica* 71: 1161–1189. <https://doi.org/10.1111/1468-0262.00442>.
- Imbens, G. W. 2000. The role of the propensity score in estimating dose–response functions. *Biometrika* 87: 706–710. <https://doi.org/10.1093/biomet/87.3.706>.
- . 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. <https://doi.org/10.1162/003465304323023651>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Lin, D. Y. 2000. Linear regression analysis of censored medical costs. *Biostatistics* 1: 35–47. <https://doi.org/10.1093/biostatistics/1.1.35>.
- Newey, W. K. 1984. A method of moments interpretation of sequential estimators. *Economics Letters* 14: 201–206. [https://doi.org/10.1016/0165-1765\(84\)90083-1](https://doi.org/10.1016/0165-1765(84)90083-1).
- Newey, W. K., and D. L. McFadden. 1994. “Large sample estimation and hypothesis testing”. In *Handbook of Econometrics*, edited by R. F. Engle and D. L. McFadden, vol. 4: 2111–2245. Amsterdam: Elsevier. [https://doi.org/10.1016/S1573-4412\(05\)80005-4](https://doi.org/10.1016/S1573-4412(05)80005-4).
- Robins, J. M., and A. Rotnitzky. 1995. Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association* 90: 122–129. <https://doi.org/10.2307/2291135>.
- . 2006. “Inverse probability weighting in survival analysis”. In *Survival and Event History Analysis*, edited by N. Keiding and P. K. Andersen, 266–271. Chichester, UK: Wiley.
- Robins, J. M., A. Rotnitzky, and L. P. Zhao. 1995. Analysis of semiparametric regression models for repeated outcomes in the presence of missing data. *Journal of the American Statistical Association* 90: 106–121. <https://doi.org/10.1080/01621459.1995.10476493>.
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2002. Inverse probability weighted M-estimators for sample selection, attrition, and stratification. *Portuguese Economic Journal* 1: 117–139. <https://doi.org/10.1007/s10258-002-0008-x>.
- . 2007. Inverse probability weighted estimation for general missing data problems. *Journal of Econometrics* 141: 1281–1301. <https://doi.org/10.1016/j.jeconom.2007.02.002>.
- . 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

[CAUSAL] **stteffects postestimation** — Postestimation tools for stteffects

[CAUSAL] **stteffects intro** — Introduction to treatment effects for observational survival-time data

[ST] **streg** — Parametric survival models

[ST] **stset** — Declare data to be survival-time data

[U] **20 Estimation and postestimation commands**

Postestimation commands

The following postestimation commands are of special interest after `stteffects`:

Command	Description
<code>teoverlap</code>	overlap plots
<code>tebalance</code>	check balance of covariates

The following standard postestimation commands are also available:

Command	Description
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>hausman</code>	Hausman’s specification test
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of parameters
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
<code>predict</code>	propensity scores, censored survival probability, etc.
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

predict

Description for predict

predict creates a new variable containing predictions such as treatment effects, conditional means, propensity scores, linear predictions, and log square roots of latent variances.

Menu for predict

Statistics > Postestimation

Syntaxes for predict

Syntaxes are presented under the following headings:

- Syntax for predict after stteffects ipw
- Syntax for predict after stteffects ipwra
- Syntax for predict after stteffects ra
- Syntax for predict after stteffects wra

Syntax for predict after stteffects ipw

```
predict [type] { stub* | newvar | newvars } [if] [in]
    [, statistic tlevel(treat_level)]

predict [type] stub* [if] [in], scores
```

statistic	Description
Main	
ps	propensity score; the default
censurv	censored survival probability
xb	linear prediction for propensity score
cxb	linear prediction for censoring model
lnsigma	log square root of latent variance (for treatment model hetprobit())
clnshape	log of conditional latent shape (for censoring distribution Weibull, log normal, or gamma)

If you do not specify tlevel() and only specify one new variable, ps assumes tlevel() specifies the first treatment level.

If you do not specify tlevel() and only specify one new variable, xb and lnsigma assume tlevel() specifies the first noncontrol treatment level.

You specify one or t new variables with ps, where t is the number of treatment levels.

You specify one or $t - 1$ new variables with xb and lnsigma.

You specify one new variable with censurv, cxb, and clnshape.

Syntax for predict after stteffects ipwra

```
predict [type] { stub* | newvar | newvars } [if] [in]
      [, statistic tlevel(treat_level)]
```

```
predict [type] stub* [if] [in], scores
```

statistic	Description
Main	
te	treatment effect; the default
cmean	conditional mean at treatment level
ps	propensity score
censurv	censored survival probability
xb	linear prediction for outcome model
cxb	linear prediction for censoring model
psxb	linear prediction for propensity score
lnshape	log of conditional latent shape (for outcome distribution Weibull, log normal, or gamma) at treatment level
clnshape	log of conditional latent shape (for censoring distribution Weibull, log normal, or gamma)
pslnsigma	log square root of latent variance (for treatment model hetprobit()) for propensity score

If you do not specify `tlevel()` and only specify one new variable, `te` and `psxb` assume `tlevel()` specifies the first non-control treatment level.

If you do not specify `tlevel()` and only specify one new variable, `cmean`, `ps`, `xb`, and `pslnsigma` assume `tlevel()` specifies the first treatment level.

You specify one or t new variables with `cmean`, `ps`, `xb`, and `lnshape`, where t is the number of treatment levels.

You specify one or $t - 1$ new variables with `te`, `psxb`, and `pslnsigma`.

You specify one new variable with `censurv`, `cxb`, and `clnshape`.

Syntax for predict after stteffects ra

```
predict [type] { stub* | newvar | newvars } [if] [in]
      [, statistic tlevel(treat_level)]

predict [type] stub* [if] [in], scores
```

statistic	Description
Main	
te	treatment effect; the default
cmean	conditional mean at treatment level
xb	linear prediction for outcome model
lnshape	log of conditional latent shape (for outcome distribution Weibull, log normal, or gamma) at treatment level

If you do not specify `tlevel()` and only specify one new variable, `te` assumes `tlevel()` specifies the first noncontrol treatment level.

If you do not specify `tlevel()` and only specify one new variable, `cmean`, `xb`, and `lnshape` assume `tlevel()` specifies the first treatment level.

You specify one or t new variables with `cmean`, `xb`, and `lnshape`, where t is the number of treatment levels.

You specify one or $t - 1$ new variables with `te`.

Syntax for predict after stteffects wra

```
predict [type] { stub* | newvar | newvars } [if] [in]
      [, statistic tlevel(treat_level)]

predict [type] stub* [if] [in], scores
```

statistic	Description
Main	
te	treatment effect; the default
cmean	conditional mean at treatment level
censurv	censored survival probability
xb	linear prediction for outcome model
cxb	linear prediction for censoring model
lnshape	log of conditional latent shape (for outcome distribution Weibull, log normal, or gamma) at treatment level
clnshape	log of conditional latent shape (for censoring distribution Weibull, log normal, or gamma)

If you do not specify `tlevel()` and only specify one new variable, `te` assumes `tlevel()` specifies the first noncontrol treatment level.

If you do not specify `tlevel()` and only specify one new variable, `cmean`, `xb`, and `lnshape` assume `tlevel()` specifies the first treatment level.

You specify one or t new variables with `cmean`, `xb`, and `lnshape`, where t is the number of treatment levels.

You specify one or $t - 1$ new variables with `te`.

You specify one new variable with `censurv`, `cxb`, and `clnshape`.

Options for predict

Options are presented under the following headings:

Options for predict after stteffects ipw
Options for predict after stteffects ipwra
Options for predict after stteffects ra
Options for predict after stteffects wra

Options for predict after stteffects ipw

Main

ps, the default, calculates the propensity score of each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

censurv calculates the survivor probability from the time-to-censoring model. (In other words, it calculates the probability that an outcome is not censored.) This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.

xb calculates the propensity score linear prediction at each noncontrol level of the treatment or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

cxb calculates the linear prediction of the censoring model. This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.

lnsigma calculates the log square root of the latent variance. This option is valid only when treatment model `hetprobit()` is used. You need to specify only one new variable.

clnshape calculates the log of the conditional latent shape parameter of the censoring distribution. This option is valid when censoring distribution Weibull, log normal, or gamma is used. You need to specify only one new variable.

tlevel (*treat_level*) specifies the treatment level for prediction.

scores calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the censoring and propensity-score equations.

The j th new variable will contain the scores for the j th parameter in the coefficient table if $j \leq t$, where t is the number of treatment levels. Otherwise, it will contain the scores for fitted equation $j - t$ following the first t parameters in the coefficient table.

Options for predict after stteffects ipwra

Main

te, the default, calculates the treatment effect for each noncontrol treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

cmean calculates the conditional mean for each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`ps` calculates the propensity score of each treatment level or the treatment level specified in `tlevel()`.

If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`censurv` calculates the survivor probability from the time-to-censoring model. (In other words, it calculates the probability that an outcome is not censored.) This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.

`xb` calculates the outcome model linear prediction at each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`cxb` calculates the linear prediction of the censoring model. This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.

`psxb` calculates the propensity score linear prediction at each noncontrol level of the treatment or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

`lnshape` calculates the log of the conditional latent shape parameter for each treatment level or the treatment level specified in `tlevel()`. This option is valid when outcome distribution Weibull, log normal, or gamma is used. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`clnshape` calculates the log of the conditional latent shape parameter for the censoring distribution. This option is valid when censoring distribution Weibull, log normal, or gamma is used. You need to specify only one new variable.

`pslnsigma` calculates the log square root of the latent variance for the propensity score. This option is valid only when treatment model `hetprobit()` is used. You need to specify only one new variable.

`tlevel(treat_level)` specifies the treatment level for prediction.

`scores` calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the outcome, censoring, and propensity-score equations.

The j th new variable will contain the scores for the j th parameter in the coefficient table if $j \leq t$, where t is the number of treatment levels. Otherwise, it will contain the scores for fitted equation $j - t$ following the first t parameters in the coefficient table.

Options for predict after stteffects ra

Main

`te`, the default, calculates the treatment effect for each noncontrol treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

`cmean` calculates the conditional mean for each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

xb calculates the outcome model linear prediction at each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

lnshape calculates the log of the conditional latent shape parameter for each treatment level or the treatment level specified in `tlevel()`. This option is valid when the outcome distribution Weibull, log normal, or gamma is used. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

tlevel (*treat_level*) specifies the treatment level for prediction.

scores calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the outcome equations.

The j th new variable will contain the scores for the j th parameter in the coefficient table if $j \leq t$, where t is the number of treatment levels. Otherwise, it will contain the scores for fitted equation $j - t$ following the first t parameters in the coefficient table.

Options for predict after stteffects wra

Main

te, the default, calculates the treatment effect for each noncontrol treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

cmean calculates the conditional mean for each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

censurv calculates the survivor probability from the time-to-censoring model. (In other words, it calculates the probability that an outcome is not censored.) This option is allowed only if a censoring model is specified at estimation time. You need to specify only one new variable.

xb calculates the outcome model linear prediction at each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

lnshape calculates the log of the conditional latent shape parameter for each treatment level or the treatment level specified in `tlevel()`. This option is valid when the outcome distribution Weibull, log normal, or gamma is used. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

clnshape calculates the log of the conditional latent shape parameter of the censoring distribution. This option is valid when the censoring distribution Weibull, log normal, or gamma is used. You need to specify only one new variable.

tlevel (*treat_level*) specifies the treatment level for prediction.

scores calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the outcome and censoring equations.

The j th new variable will contain the scores for the j th parameter in the coefficient table if $j \leq t$, where t is the number of treatment levels. Otherwise, it will contain the scores for fitted equation $j - t$ following the first t parameters in the coefficient table.

Remarks and examples

Checking model specification is the most frequent reason for postestimation computation after `stteffects`. `teoverlap` provides a graphical method for checking the overlap assumption; see [\[CAUSAL\] teoverlap](#). Summarizing the estimated probabilities provides another check. Recall that the reciprocals of these estimated probabilities are used as weights by some of the estimators. If the estimated probabilities are too small, the weights get too large and the estimators become unstable.

We estimate the average treatment effect of smoking on the time to a second heart attack by inverse-probability weighting; see [example 1](#) of [\[CAUSAL\] stteffects ipw](#) for background.

```
. use https://www.stata-press.com/data/r19/sheart
(Time to second heart attack (fictional))
. stteffects ipw (smoke age exercise education) (age exercise diet education)
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 2.042e-18
Iteration 1:  EE criterion = 1.890e-30
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit
Censoring model: Weibull
```

_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
smoke (Smoker vs Nonsmoker)	-2.22226	.6307573	-3.52	0.000	-3.458522	-.9859983
P0mean						
smoke Nonsmoker	4.235569	.5210937	8.13	0.000	3.214244	5.256894

Below, we compute the estimated probabilities of being a Nonsmoker and store them in `ps0`. Likewise, the estimated probabilities of being a Smoker are stored in `ps1`.

```
. predict ps0 ps1, ps
```

The overlap condition requires that each of these probabilities be sufficiently greater than 0 and less than 1 for every individual; see [Assumptions and tradeoffs](#) under *Remarks and examples* in [\[CAUSAL\] stteffects intro](#).

In practice, we know that weighting estimators perform poorly when the weights become too large. This approach requires that the probability of being a Nonsmoker not be too small among Nonsmokers and that the probability of being a Smoker not be too small among Smokers. Below, we summarize these probabilities.

```
. summarize ps0 if fail==1 & smoke==0
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps0	716	.6712529	.138754	.3872543	.9840293

```
. summarize ps1 if fail==1 & smoke==1
```

Variable	Obs	Mean	Std. dev.	Min	Max
ps1	492	.4101277	.1101277	.0850604	.6125538

The minimum probability of being a Nonsmoker among Nonsmokers is 0.39. The minimum probability of being a Smoker among Smokers is 0.09. Neither minimum seems too small.

Estimating survival-time treatment effects also uses weights to adjust for censored outcomes; see [\[CAUSAL\] stteffects intro](#). Thus we require that the probability of an uncensored failure also be sufficiently greater than 0. Below, we compute the estimated probabilities of failure and summarize them among those that fail.

```
. predict fprob2, censurv
```

```
. summarize fprob if fail==1
```

Variable	Obs	Mean	Std. dev.	Min	Max
fprob2	1,208	.7246067	.2143543	.0364246	.9999086

The minimum probability of 0.04 does not appear too small.

□ Technical note

The previous discussion builds on the intuition that the weights used in a weighting estimator should not be too large.

This technical note goes a little further by explicitly computing the weights and using them to replicate the inverse-probability-weighted point estimate for the Nonsmoker potential-outcome mean.

We now compute the weights using the predicted probabilities computed in the examples above and then use [mean](#) to compute the weighted average that estimates the potential-outcome mean for Nonsmokers.

```
. generate double ipw0 = 1/(ps0*fprob)
```

```
. mean _t [pw=ipw0] if smoke==0 & fail==1
```

Mean estimation Number of obs = 716

	Mean	Std. err.	[95% conf. interval]	
_t	4.235569	.5820212	3.092894	5.378244

The weights account for data lost to the Smoker potential outcome or to censoring by increasing the importance of observations that were observed to be Nonsmoker failure times even though they were not likely to be observed.

The point estimate matches that reported by `stteffects ipw`; the standard errors differ because `mean` takes the estimated weights as given. See [Inverse-probability-weighted estimators](#) under *Methods and formulas* in [\[CAUSAL\] stteffects ipwra](#).

References

- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Bai, X., A. A. Tsiatis, and S. M. O'Brien. 2013. Doubly robust estimators of treatment-specific survival distributions in observational studies with stratified sampling. *Biometrics* 69: 830–839. <https://doi.org/10.1111/biom.12076>.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: Sage.
- Imbens, G. W. 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. <https://doi.org/10.1162/003465304323023651>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

- [CAUSAL] **tebalance** — Check balance after teffects or stteffects estimation
- [CAUSAL] **teoverlap** — Overlap plots
- [CAUSAL] **stteffects ipw** — Survival-time inverse-probability weighting
- [CAUSAL] **stteffects ipwra** — Survival-time inverse-probability-weighted regression adjustment
- [CAUSAL] **stteffects ra** — Survival-time regression adjustment
- [CAUSAL] **stteffects wra** — Survival-time weighted regression adjustment
- [U] **20 Estimation and postestimation commands**

[Description](#)
[Options](#)
[References](#)

[Quick start](#)
[Remarks and examples](#)
[Also see](#)

[Menu](#)
[Stored results](#)

[Syntax](#)
[Methods and formulas](#)

Description

`stteffects ra` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational survival-time data by regression adjustment (RA). RA uses averages of treatment-specific predicted mean survival times to estimate mean survival times for each potential outcome. Contrasts of these predicted mean survival times estimate the treatment effects. `stteffects ra` offers several choices for the model used to predict mean survival time. Binary and multivalued treatments are accommodated.

See [\[CAUSAL\] stteffects intro](#) for an overview of estimating treatment effects from observational survival-time data.

Quick start

Specify time as observed failure time and fail as failure indicator

```
stset time, failure(fail)
```

ATE from a Weibull model for time on `x1` and `x2` with binary treatment `treat2`

```
stteffects ra (x1 x2) (treat2)
```

Same as above, but estimate the ATET

```
stteffects ra (x1 x2) (treat2), atet
```

Same as above, but estimate the potential-outcome means

```
stteffects ra (x1 x2) (treat2), pomeans
```

ATE of `treat2` using a gamma model for time

```
stteffects ra (x1 x2, gamma) (treat2)
```

ATE for each level of three-valued treatment `treat3`

```
stteffects ra (x1 x2) (treat3)
```

Same as above, and specify that `treat3 = 3` is the control level using the value label “MyControl” for 3

```
stteffects ra (x1 x2) (treat3), control("MyControl")
```

Menu

Statistics > Causal inference/treatment effects > Survival outcomes > Regression adjustment

Syntax

```
stteffects ra (omvarlist [ , omoptions ]) (tvar) [if] [in] [ , stat options ]
```

omvarlist specifies the variables that predict the survival-time variable in the outcome model.
tvar must contain integer values representing the treatment levels.

<i>omoptions</i>	Description
Model	
<u>weibull</u>	Weibull; the default
<u>exponential</u>	exponential
<u>gamma</u>	two-parameter gamma
<u>lnormal</u>	lognormal
<u>ancillary</u> (<i>avarlist</i> [, <u>noconstant</u>])	specify variables used to model ancillary parameter
<u>noconstant</u>	suppress constant from outcome model

<i>stat</i>	Description
Stat	
<u>ate</u>	estimate average treatment effect in population; the default
<u>atet</u>	estimate average treatment effect on the treated
<u>pomeans</u>	estimate potential-outcome means

<i>options</i>	Description
SE/Robust	
<u>vce</u> (<i>vcetype</i>)	<i>vcetype</i> may be <u>robust</u> , <u>cluster</u> <i>clustvar</i> , <u>bootstrap</u> , or <u>jackknife</u>

Reporting	
<u>level</u> (#)	set confidence level; default is level(95)
<u>aequations</u>	display auxiliary-equation results
<u>noshow</u>	do not show st setting information
<i>display_options</i>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

Maximization	
<i>maximize_options</i>	control the maximization process; seldom used
<u>iterinit</u> (#)	specify starting-value iterations; seldom used

Advanced	
<u>control</u> (# <i>label</i>)	specify the level of <i>tvar</i> that is the control
<u>tlevel</u> (# <i>label</i>)	specify the level of <i>tvar</i> that is the treatment
<u>coeflegend</u>	display legend instead of statistics

You must `stset` your data before using `stteffects`; see [ST] [stset](#).

`omvarlist` and `avarlist` may contain factor variables; see [U] [11.4.3 Factor variables](#).

`bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see [U] [11.1.10 Prefix commands](#).

Weights are not allowed with the `bootstrap` prefix; see [R] [bootstrap](#).

`fweights`, `iweights`, and `pweights` may be specified using `stset`; see [Weights](#) under *Remarks and examples* in [ST] [stset](#).

However, weights may not be specified if you are using the `bootstrap` prefix.

`coeflegend` does not appear in the dialog box.

See [U] [20 Estimation and postestimation commands](#) for more capabilities of estimation commands.

Options

Model

`ancillary(avarlist [ , noconstant ])` specifies the variables used to model the ancillary parameter.

By default, the ancillary parameter does not depend on covariates. Specifying `ancillary(avarlist, noconstant)` causes the constant to be suppressed in the model for the ancillary parameter.

`noconstant`; see [R] [Estimation options](#).

Stat

`stat` is one of three statistics: `ate`, `atet`, or `pomeans`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`atet` specifies that the average treatment effect on the treated be estimated.

`pomeans` specifies that the potential-outcome means for each treatment level be estimated.

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use `bootstrap` or `jackknife` methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

Reporting

`level(#)`; see [R] [Estimation options](#).

`aequations` specifies that the results for the outcome-model or treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`noshow` prevents `stteffects ra` from showing the key `st` variables. This option is rarely used because most people type `stset`, `show` or `stset`, `noshow` to permanently set whether they want to see these variables mentioned at the top of the output of every `st` command; see [ST] [stset](#).

display_options: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `novlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

Maximization

`maximize_options`: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

`init_specs` is one of

```
matname [ , skip copy]
# [ , # ...], copy
```

`iterinit(#)` specifies the maximum number of iterations used to calculate the starting values. This option is seldom used.

Advanced

`control(#|label)` specifies the level of *tvar* that is the control. The default is the first treatment level.

You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with the statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(#|label)` specifies the level of *tvar* that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `stteffects` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

If you are not familiar with the framework for treatment-effects estimation from observational survival-time data, please see [\[CAUSAL\] stteffects intro](#).

RA estimators use contrasts of the averages of treatment-specific predicted mean outcomes to estimate treatment effects. RA estimators use a two-step approach to estimating treatment effects:

1. For each treatment level, fit a model of the survival-time outcome on the same set of covariates.
2. Compute the averages of the predicted outcomes for each subject within each treatment level.

These averages estimate the potential-outcome means (POMs). Contrasts of these averages estimate the ATEs. By restricting the computations of the averages to the subset of treated subjects, we obtain estimates of the ATETs.

Here we note only a few entry points to the vast literature on RA estimators. [Imbens \(2004\)](#), [Imbens and Wooldridge \(2009\)](#), [Cameron and Trivedi \(2005, chap. 25\)](#), [Wooldridge \(2010, chap. 21\)](#), and [Vittinghoff et al. \(2012, chap. 9\)](#) provide excellent general introductions to estimating ATEs and to RA estimators in particular.

Like `streg` and other survival-time commands, `stteffects ra` uses the outcome variable and the failure indicator computed by, and optionally weights specified with, `stset`. `stteffects ra` is not appropriate for data with time-varying covariates, also known as multiple-record survival-time data, or for delayed-entry data.

► Example 1: Estimating the ATE

Suppose we wish to study the effect of smoking on the time to a second heart attack among women aged 45–55 years. In our fictional sheart dataset, `atime` is the observed time in years to a second heart attack or censoring, and `fail` is the 0/1 indicator that a second heart attack was observed. (When `fail` is 1, `atime` records the time to the second heart attack; when `fail` is 0, `atime` records a censored observation of the time to a second heart attack.) We previously `stset` these data; see [A quick tour of the estimators](#) in [\[CAUSAL\] stteffects intro](#).

The treatment, smoking, is stored in the 0/1 indicator `smoke`. These data also contain age at the time of the first heart attack (`age`), and indices of the level of exercise (`exercise`), diet quality (`diet`), and education (`education`) prior to the first heart attack.

We can use `stteffects ra` to estimate the ATE by RA. We model the mean survival time using the default Weibull model, controlling for age, exercise, diet, and education, and we specify that `smoke` is the treatment variable.

```
. use https://www.stata-press.com/data/r19/sheart
(Time to second heart attack (fictional))
. stteffects ra (age exercise diet education) (smoke)

      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 1.006e-14
Iteration 1:  EE criterion = 2.302e-25

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model   : Weibull
Treatment model : none
Censoring model : none
```

_t	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
smoke						
(Smoker						
vs						
Nonsmoker)	-1.956657	.3331787	-5.87	0.000	-2.609676	-1.303639
POmean						
smoke						
Nonsmoker	4.243974	.2620538	16.20	0.000	3.730358	4.75759

When every woman smoked in the population of women aged 45–55 years who have had a heart attack, the average time to a second heart attack is estimated to be 1.96 years less than when no women in the population of interest smoked. The estimated average time to a second heart attack when no women in the population of interest smoked is 4.24 years. In other words, if every woman in the population of interest smoked, then the average time to a second heart attack would fall by an estimated 46% relative to the case when no women smoked.

► Example 2: Changing the outcome model

Instead of a Weibull model for the outcome model, we could have used an exponential, a gamma, or a lognormal model. By way of comparison, we use a gamma model and the same covariates to estimate the ATE.

```
. stteffects ra (age exercise diet education, gamma) (smoke)
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 6.213e-25
Iteration 1:  EE criterion = 2.152e-31
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model   : gamma
Treatment model : none
Censoring model : none
```

	_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE							
smoke (Smoker vs Nonsmoker)		-1.801787	.2924388	-6.16	0.000	-2.374956	-1.228617
POMean							
smoke Nonsmoker		3.994327	.2258257	17.69	0.000	3.551717	4.436937

The estimated ATE of -1.80 and control-level POM of 3.99 are similar to those of -1.96 and 4.24 obtained from the Weibull model in [example 1](#). The ratio of the estimated ATE to the control-level POM indicates a 45% reduction instead of the 46% reduction obtained from the Weibull model.



► Example 3: Estimating the ratio of the ATE to the control-level POM

The ratio of the ATE to the control-level POM measures the importance of the effect. In [example 1](#), we computed the point estimate of this ratio from the output, but we were left without a confidence interval. In this example, we use `nlcom` to compute a point estimate and a confidence interval.

Below, we refit the model from [example 1](#), specifying the `coeflegend` option to learn the parameter names. We use the parameter names in `nlcom` to estimate the ratio of the ATE to the control-level POM.

```
. stteffects ra (age exercise diet education) (smoke), coeflegend
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 1.006e-14
Iteration 1:  EE criterion = 2.302e-25
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model   : Weibull
Treatment model : none
Censoring model : none
```

_t	Coefficient	Legend
ATE		
smoke		
(Smoker		
vs		
Nonsmoker)	-1.956657	_b[ATE:r1vs0.smoke]
POMean		
smoke		
Nonsmoker	4.243974	_b[POMean:0.smoke]

```
. nlcom _b[ATE:r1vs0.smoke] / _b[POMean:0.smoke]
      _nl_1:  _b[ATE:r1vs0.smoke] / _b[POMean:0.smoke]
```

_t	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
_nl_1	-.4610437	.0598709	-7.70	0.000	-.5783885	-.3436988

The output shows that when every woman smoked, the average time to a second heart attack falls by an estimated 46% relative to the case when no women smoked, as we computed earlier. We also obtain a 95% confidence interval of 34% to 58% for this estimate.



► Example 4: Estimating the ATET

Intuitively, the ATET measures the effect of the treatment on an at-risk subpopulation. Sometimes the subpopulation that gets the treatment defines such an at-risk subpopulation. The ATET has the added benefit that it can be estimated under weaker conditions than the ATE; see [Assumptions and tradeoffs](#) in [\[CAUSAL\] stteffects intro](#).

```
. stteffects ra (age exercise diet education) (smoke), atet
      Failure _d: fail
      Analysis time _t: atime
Iteration 0:  EE criterion = 1.006e-14
Iteration 1:  EE criterion = 2.985e-26
Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: none
```

	_t	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET							
smoke							
(Smoker							
vs							
Nonsmoker)		-1.527476	.2489203	-6.14	0.000	-2.015351	-1.039602
POmean							
smoke							
Nonsmoker		3.436937	.2217808	15.50	0.000	3.002255	3.87162

When every woman in the subpopulation smoked, the average time to a second heart attack is estimated to be 1.53 years less than when no women in the subpopulation smoked. The estimated average time to a second heart attack when no women in the subpopulation smoked is 3.44 years.

◀

► Example 5: Fixed or random censoring time

The time to censoring in survival-time data can be random or deterministic, although it must be independent of treatment assignment and the potential outcomes; see [Kalbfleisch and Prentice \(2002, chap. 3\)](#) for the standard case and see [The correct adjustment for censoring assumption](#) under [Assumptions and tradeoffs](#) in [\[CAUSAL\] stteffects intro](#) for the treatment-effects case.

The RA estimator and the likelihood-adjusted-censoring version of the inverse-probability-weighted RA estimator can accommodate a fixed time to censoring; see [The correct adjustment for censoring assumption](#) in [\[CAUSAL\] stteffects intro](#). (The estimators that handle censoring by weighting cannot accommodate a fixed time to censoring because the weights are not well defined with a fixed time to censoring.)

We have fictional data on the time to rearrest among men aged 25–35 who were previously in prison for a felony conviction (`rtime`). The time to censoring is fixed in these data because individuals were followed for a maximum of five years.

Some of the young men chose to enter a vocational training program before release from prison; `train` is 1 for participants and 0 for nonparticipants. The dataset also contains `fail` (which is 1 if the observed time is a failure time and 0 if it is time to censoring), `age` at the time of the first arrest (`age`), an index of the parents' socioeconomic level (`parental`), and the number of years behind in school at the time of the first arrest (`edeficit`).

We estimate the ATET because we wish to allow the gains from the training program to be related to an unobservable characteristic that affects who self-selects into the program; see [Average treatment effect on the treated](#) in [CAUSAL] [stteffects intro](#).

We model the outcome as a function of `age`, `parental`, and `edeficit`.

```
. use https://www.stata-press.com/data/r19/recid2, clear
(Time to rearrest (fictional))

. stteffects ra (age parental edeficit) (train), atet

      Failure _d: fail
      Analysis time _t: rtime
Iteration 0:  EE criterion = 1.875e-23
Iteration 1:  EE criterion = 1.972e-31

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : regression adjustment
Outcome model   : Weibull
Treatment model : none
Censoring model: none
```

_t	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATET						
train						
(Student						
vs						
Nonstudent)	2.440919	.4689057	5.21	0.000	1.52188	3.359957
POmean						
train						
Nonstudent	2.062029	.1231492	16.74	0.000	1.820661	2.303397

When everyone who selected the training got the training, the average time to rearrest is 2.44 years later than the average rearrest time if none of those who selected the training got the training. The average rearrest time if none of those who selected the training got the training is 2.06 years. In other words, the average time to rearrest increases from about 2.06 years to about 4.50 years for the subpopulation of young men who self-selected into the prerelease vocational training program.

Stored results

`stteffects ra` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>stteffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(dead)</code>	<code>_d</code>
<code>e(depvar)</code>	<code>_t</code>
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>ra</code>
<code>e(omodel)</code>	outcome model: <code>weibull</code> , <code>exponential</code> , <code>gamma</code> , or <code>lognormal</code>
<code>e(stat)</code>	statistic estimated: <code>ate</code> , <code>atet</code> , or <code>pomeans</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vce</i> type specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b</code> <code>V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

The methods and formulas for the RA estimators implemented in `stteffects ra` are given in [Methods and formulas](#) of [CAUSAL] `stteffects ipwra`.

References

- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Bai, X., A. A. Tsiatis, and S. M. O'Brien. 2013. Doubly robust estimators of treatment-specific survival distributions in observational studies with stratified sampling. *Biometrics* 69: 830–839. <https://doi.org/10.1111/biom.12076>.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. [Estimation of multivalued treatment effects under conditional independence](#). *Stata Journal* 13: 407–450.
- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: Sage.
- Imbens, G. W. 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. <https://doi.org/10.1162/003465304323023651>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Kalbfleisch, J. D., and R. L. Prentice. 2002. *The Statistical Analysis of Failure Time Data*. 2nd ed. New York: Wiley.
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

- [CAUSAL] [stteffects postestimation](#) — Postestimation tools for stteffects
- [CAUSAL] [stteffects intro](#) — Introduction to treatment effects for observational survival-time data
- [ST] [streg](#) — Parametric survival models
- [ST] [stset](#) — Declare data to be survival-time data
- [U] [20 Estimation and postestimation commands](#)

[Description](#)
[Options](#)
[References](#)

[Quick start](#)
[Remarks and examples](#)
[Also see](#)

[Menu](#)
[Stored results](#)

[Syntax](#)
[Methods and formulas](#)

Description

`stteffects wra` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational survival-time data with random time to censoring. Estimation is by weighted regression adjustment (WRA). WRA estimators use inverse-probability-of-censoring adjusted regression coefficients to compute averages of treatment-level predicted outcomes. Contrasts of these averages estimate the treatment effects. WRA uses estimated weights from a time-to-censoring model to account for censored survival times instead of including a term in the likelihood function. `stteffects wra` offers several choices for the functional forms of the outcome model and the time-to-censoring model. Binary and multivalued treatments are accommodated.

See [\[CAUSAL\]](#) [stteffects intro](#) for an overview of estimating treatment effects from observational survival-time data.

Quick start

Specify time as observed failure time and fail as failure indicator

```
stset time, failure(fail)
```

ATE from a Weibull model for time on x1 and x2 with binary treatment treat2 and a Weibull model on x1 and x2 for censoring

```
stteffects wra (x1 x2) (treat2) (x1 x2)
```

Same as above, but estimate the ATET

```
stteffects wra (x1 x2) (treat2) (x1 x2), atet
```

ATE of treat2 using a gamma model for time and a gamma censoring model

```
stteffects wra (x1 x2, gamma) (treat2) (x1 x2, gamma)
```

ATE for each level of three-valued treatment treat3

```
stteffects wra (x1 x2) (treat3) (x1 x2)
```

Same as above, and specify that treat3 = 3 is the control level using the value label “MyControl” for 3

```
stteffects wra (x1 x2) (treat3) (x1 x2), control("MyControl")
```

Menu

Statistics > Causal inference/treatment effects > Survival outcomes > Weighted regression adjustment

Syntax

```
stteffects wra (omvarlist [ , omoptions ]) (tvar) (cmvarlist [ , cmoptions ])
[if] [in] [ , stat options ]
```

omvarlist specifies the variables that predict the survival-time variable in the outcome model.
tvar must contain integer values representing the treatment levels.
cmvarlist specifies the variables that predict censoring in the censoring model.

<i>omoptions</i>	Description
Model	
<u>weibull</u>	Weibull; the default
<u>exponential</u>	exponential
<u>gamma</u>	two-parameter gamma
<u>lnormal</u>	lognormal
<u>ancillary</u> (<i>avarlist</i> [, <u>noconstant</u>])	specify variables used to model ancillary parameter
<u>noconstant</u>	suppress constant from outcome model
<i>cmoptions</i>	Description
Model	
<u>weibull</u>	Weibull; the default
<u>exponential</u>	exponential
<u>gamma</u>	two-parameter gamma
<u>lnormal</u>	lognormal
<u>ancillary</u> (<i>avarlist</i> [, <u>noconstant</u>])	specify variables used to model ancillary parameter
<u>noconstant</u>	suppress constant from censoring model
<i>stat</i>	Description
Stat	
<u>ate</u>	estimate average treatment effect in population; the default
<u>atet</u>	estimate average treatment effect on the treated
<u>pomeans</u>	estimate potential-outcome means

options	Description
SE/Robust vce(vcetype)	vcetype may be <u>r</u> obust, <u>c</u> luster <i>clustvar</i> , <u>b</u> ootstrap, or <u>j</u> ackknife
Reporting _level(#) aequations noshow display_options	set confidence level; default is level(95) display auxiliary-equation results do not show st setting information control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization maximize_options iterinit(#)	control the maximization process; seldom used specify starting-value iterations; seldom used
Advanced pstolerance(#) osample(newvar) control(# label) tlevel(# label) coeflegend	set tolerance for the overlap assumption identify observations that violate the overlap assumption specify the level of tvar that is the control specify the level of tvar that is the treatment display legend instead of statistics

You must stset your data before using stteffects; see [ST] stset.

omvarlist, cmvarlist, and avarlist may contain factor variables; see [U] 11.4.3 Factor variables.

bootstrap, by, collect, jackknife, and statsby are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the bootstrap prefix; see [R] bootstrap.

fweights, iweights, and pweights may be specified using stset; see Weights under Remarks and examples in [ST] stset.

However, weights may not be specified if you are using the bootstrap prefix.

coeflegend does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

Model

ancillary(avarlist [, noconstant]) specifies the variables used to model the ancillary parameter.

By default, the ancillary parameter does not depend on covariates. Specifying ancillary(avarlist, noconstant) causes the constant to be suppressed in the model for the ancillary parameter.

ancillary() may be specified for the model for survival-time outcome, for the model for the censoring variable, or for both. If ancillary() is specified for both, the varlist used for each model may be different.

noconstant; see [R] Estimation options.

Stat

stat is one of three statistics: *ate*, *atet*, or *pomeans*. *ate* is the default.

ate specifies that the average treatment effect be estimated.

atet specifies that the average treatment effect on the treated be estimated.

pomeans specifies that the potential-outcome means for each treatment level be estimated.

SE/Robust

vce (*vcetype*) specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (*robust*), that allow for intragroup correlation (*cluster clustvar*), and that use bootstrap or jackknife methods (*bootstrap*, *jackknife*); see [R] [vce_option](#).

Reporting

level(#); see [R] [Estimation options](#).

aequations specifies that the results for the outcome-model or treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

noshow prevents *stteffects wra* from showing the key *st* variables. This option is rarely used because most people type *stset*, *show* or *stset*, *noshow* to permanently set whether they want to see these variables mentioned at the top of the output of every *st* command; see [ST] [stset](#).

display_options: *nocl*, *nopvalues*, *noomitted*, *vsquish*, *noemptycells*, *baselevels*, *allbaselevels*, *nofvlabel*, *fvwrap*(#), *fvwrapon*(*style*), *cformat*(%*fmt*), *pformat*(%*fmt*), *sformat*(%*fmt*), and *nolstretch*; see [R] [Estimation options](#).

Maximization

maximize_options: *iterate*(#), *[no]log*, and *from*(*init_specs*); see [R] [Maximize](#). These options are seldom used.

init_specs is one of

matname [, *skip copy*]

[, # ...], *copy*

iterinit(#) specifies the maximum number of iterations used to calculate the starting values. This option is seldom used.

Advanced

pstolerance(#) specifies the tolerance used to check the overlap assumption. The default value is *pstolerance*(1e-5). *stteffects* will exit with an error if an observation has an estimated propensity score smaller than that specified by *pstolerance*().

osample(*newvar*) specifies that indicator variable *newvar* be created to identify observations that violate the overlap assumption.

control(#| *label*) specifies the level of *tvar* that is the control. The default is the first treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. *control*() may not be specified with the statistic *pomeans*. *control*() and *tlevel*() may not specify the same treatment level.

`tlevel(#|label)` specifies the level of *tvar* that is the treatment for the statistic *atet*. The default is the second treatment level. You may specify the numeric level *#* (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic *atet*. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `stteffects` but is not shown in the dialog box:

`coeflegend`; see [\[R\] Estimation options](#).

Remarks and examples

If you are not familiar with the framework for treatment-effects estimation from observational survival-time data, please see [\[CAUSAL\] stteffects intro](#).

Weighted regression-adjustment (WRA) estimators use estimated weights to account for censoring when estimating outcome-regression parameters. The estimated outcome-regression parameters are used to compute averages of treatment-level predicted outcomes. Contrasts of these averages estimate the treatment effects.

WRA estimators use a three-step approach to estimating treatment effects:

1. They estimate the parameters of a time-to-censoring model and compute inverse-probability-of-censoring weights.
2. Using the estimated inverse-probability-of-censoring weights, they use weighted maximum likelihood estimators for the outcome for each treatment level and obtain the treatment-specific predicted mean outcomes for each subject. The inverse-probability-of-censoring weights account for right-censored survival times.
3. They compute the means of the treatment-specific predicted mean outcomes. Contrasts of these averages provide the estimates of the ATEs. By restricting the computations of the means to the subset of treated subjects, we can obtain the ATETs.

WRA estimators differ from RA estimators in that WRA estimators use weights to account for observations lost to censoring while RA estimators use an additional term in the likelihood function. A model for the time to censoring is used to estimate the weights.

WRA estimators require more assumptions than RA estimators. Specifically, they require that the censoring time be random and that the time-to-censoring model be well specified. The implemented WRA estimators also require that the time-to-censoring process not vary by treatment level. The RA estimator and the likelihood-adjusted-censoring version of the inverse-probability-weighted RA estimator do not require these extra assumptions, because they use a likelihood term instead of weights to adjust for the data lost to censoring; see [\[CAUSAL\] stteffects ra](#) and [\[CAUSAL\] stteffects ipwra](#).

Here we note only a few entry points to the vast literature on weighted estimators. [Imbens \(2004\)](#), [Imbens and Wooldridge \(2009\)](#), [Robins and Rotnitzky \(2006\)](#), [Wooldridge \(2002, 2007\)](#), [Cameron and Trivedi \(2005, chap. 25\)](#), [Wooldridge \(2010, chap. 21\)](#), and [Vittinghoff et al. \(2012, chap. 9\)](#) provide excellent general introductions to estimating ATEs and to WRA estimators in particular.

Like `streg` and other survival-time commands, `stteffects vra` uses the outcome variable and the failure indicator computed by, and optionally weights specified with, `stset`. `stteffects vra` is not appropriate for data with time-varying covariates, also known as multiple-record survival-time data, or for delayed-entry data.

► Example 1: Estimating the ATE

Suppose we wish to study the effect of smoking on the time to a second heart attack among women aged 45–55 years. In our fictional sheart dataset, `atime` is the observed time in years to a second heart attack or censoring, and `fail` is the 0/1 indicator that a second heart attack was observed. (When `fail` is 1, `atime` records the time to the second heart attack; when `fail` is 0, `atime` records a censored observation of the time to a second heart attack.) We previously `stset` these data; see [A quick tour of the estimators](#) in [\[CAUSAL\] stteffects intro](#).

The treatment, smoking, is stored in the 0/1 indicator `smoke`. These data also contain age at the time of the first heart attack (`age`), and indices of the level of exercise (`exercise`), diet quality (`diet`), and education (`education`) prior to the first heart attack.

We can use `stteffects wra` to estimate the ATE by WRA. We model the mean survival time using the default Weibull outcome model with `age`, `exercise`, `diet`, and `education` as covariates, and we specify that `smoke` is the treatment variable. We also specify the default Weibull time-to-censoring model and include `age`, square of `age`, `exercise`, and `education`.

```
. use https://www.stata-press.com/data/r19/sheart
(Time to second heart attack (fictional))
. stteffects wra (age exercise diet education)
> (smoke)
> (age c.age#c.age exercise diet education)

      Failure _d: fail
    Analysis time _t: atime

Iteration 0:  EE criterion = 4.096e-18
Iteration 1:  EE criterion = 1.279e-29

Survival treatment-effects estimation      Number of obs      =      2,000
Estimator      : weighted regression adjustment
Outcome model  : Weibull
Treatment model: none
Censoring model: Weibull
```

_t	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
smoke						
(Smoker						
vs						
Nonsmoker)	-2.374174	.6017498	-3.95	0.000	-3.553582	-1.194766
P0mean						
smoke						
Nonsmoker	4.302131	.5528943	7.78	0.000	3.218478	5.385784

When every woman smoked in the population of women aged 45–55 years who have had a heart attack, the average time to a second heart attack is estimated to be 2.37 years less than when no women in the subpopulation of interest smoked. The estimated average time to a second heart attack when no women in the subpopulation of interest smoked is 4.30 years.

Stored results

`stteffects wra` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>stteffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(dead)</code>	<code>_d</code>
<code>e(depvar)</code>	<code>_t</code>
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>wra</code>
<code>e(omodel)</code>	outcome model: weibull, exponential, gamma, or lognormal
<code>e(cmodel)</code>	censoring model: weibull, exponential, gamma, or lognormal
<code>e(stat)</code>	statistic estimated: ate, atet, or pomeans
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

The methods and formulas for the WRA estimators implemented in `stteffects wra` are given in *Methods and formulas* of [CAUSAL] `stteffects ipwra`.

References

- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Bai, X., A. A. Tsiatis, and S. M. O'Brien. 2013. Doubly robust estimators of treatment-specific survival distributions in observational studies with stratified sampling. *Biometrics* 69: 830–839. <https://doi.org/10.1111/biom.12076>.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: Sage.
- Imbens, G. W. 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. <https://doi.org/10.1162/003465304323023651>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Robins, J. M., and A. Rotnitzky. 2006. “Inverse probability weighting in survival analysis”. In *Survival and Event History Analysis*, edited by N. Keiding and P. K. Andersen, 266–271. Chichester, UK: Wiley.
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.
- Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2002. Inverse probability weighted M-estimators for sample selection, attrition, and stratification. *Portuguese Economic Journal* 1: 117–139. <https://doi.org/10.1007/s10258-002-0008-x>.
- . 2007. Inverse probability weighted estimation for general missing data problems. *Journal of Econometrics* 141: 1281–1301. <https://doi.org/10.1016/j.jeconom.2007.02.002>.
- . 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

- [CAUSAL] **stteffects postestimation** — Postestimation tools for stteffects
- [CAUSAL] **stteffects intro** — Introduction to treatment effects for observational survival-time data
- [ST] **streg** — Parametric survival models
- [ST] **stset** — Declare data to be survival-time data
- [U] **20 Estimation and postestimation commands**

Description

The `tebalance` postestimation commands produce diagnostic statistics, test statistics, and diagnostic plots to assess whether a `teffects` or an `stteffects` command balanced the covariates over treatment levels.

Syntax

`tebalance subcommand ... [ , options ]`

<i>subcommand</i>	Description
<code>summarize</code>	compare means and variances in raw and balanced data
<code>overid</code>	overidentification test
<code>density</code>	kernel density plots for raw and balanced data
<code>box</code>	box plots for each treatment level for balanced data

Remarks and examples

This entry provides an overview of the commands in `tebalance`. We recommend that you read this entry before proceeding to [\[CAUSAL\] `tebalance summarize`](#), [\[CAUSAL\] `tebalance overid`](#), [\[CAUSAL\] `tebalance density`](#), or [\[CAUSAL\] `tebalance box`](#) for command-specific syntax and details.

A covariate is said to be balanced when its distribution does not vary over treatment levels.

Covariates are balanced in experimental data because treatment assignment is independent of the covariates because of the study design. In contrast, covariates must be balanced by weighting or matching in observational data because treatment assignment is related to the covariates that also affect the outcome of interest.

The estimators implemented in `teffects` and `stteffects` use a model or matching method to make the outcome conditionally independent of the treatment by conditioning on covariates. If this model or matching method is well specified, it should balance the covariates. Balance diagnostic techniques and tests check the specification of the conditioning method used by a `teffects` or an `stteffects` estimator; see [\[CAUSAL\] `teffects intro advanced`](#) for an introduction to `teffects`, and [\[CAUSAL\] `stteffects intro`](#) for an introduction to `stteffects`.

`tebalance` implements four methods to check for balance after `teffects` and `stteffects`. Which `tebalance` methods are available depends on the `teffects` estimation method, as summarized in the table below.

tebalance		Works after teffects					Works after stteffects	
method	Description	ipw	aipw	ipwra	nnmatch	psmatch	ipw	ipwra
summarize	standardized differences and variance ratios	x	x	x	x	x	x	x
overid	χ^2 test for balance	x	x	x			x	x
density	diagnostic kernel density plots	x	x	x	x	x	x	x
box	diagnostic box plots				x	x		

`tebalance overid` implements a formal test, while the other three methods are exploratory diagnostic techniques. There is no balance check after `teffects ra`, `stteffects ra`, or `stteffects wra`, because they use neither a treatment model nor a matching method.

Austin (2009, 2011) and Guo and Fraser (2015, sec. 5.52) provide introductions to covariate balance. Imai and Ratkovic (2014) derived a test for balance implemented in `tebalance overid`.

The remainder of this entry provides a quick introduction to using `tebalance` to check for balance after `teffects`. See [CAUSAL] [stteffects intro](#) for examples after `stteffects`.

► Example 1: Balance after estimators that use weighting

Inverse-probability-weighted (IPW) estimators use a model for the treatment to make the outcome conditionally independent of the treatment. If this model is well specified, it will also balance the covariates.

Using an extract from Cattaneo (2010), we use `teffects ipw` to estimate the effect of a mother's smoking behavior (`mbsmoke`) on the birthweight of her child (`bweight`), controlling for marital status (`mmarried`), the mother's age (`mage`), whether the mother had a prenatal doctor's visit in the baby's first trimester (`prenatal1`), and whether this baby is the mother's first child (`fbaby`).

```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138–154)
. teffects ipw (bweight) (mbsmoke mmarried mage prenatal1 fbaby)

Iteration 0:  EE criterion = 1.873e-22
Iteration 1:  EE criterion = 3.315e-26

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit
```

bweight	Robust					
	Coefficient	std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (Smoker vs Nonsmoker)	-236.1038	23.86187	-9.89	0.000	-282.8722	-189.3354
POmean						
mbsmoke Nonsmoker	3402.552	9.539555	356.68	0.000	3383.855	3421.249

Rubin (2008) recommends finding a model that balances the covariates before looking at results for the estimated treatment effect. Thus we do not interpret the above results, and we note that we could pay closer heed to Rubin’s recommendation by preceding the `teffects` command with `quietly` to suppress the output.

Imai and Ratkovic (2014) derived a test for balance by viewing the restrictions imposed by balance as overidentifying conditions. This test is implemented in `tebalance overid`, and we use it to test whether the above treatment model balanced the covariates.

```
. tebalance overid
Iteration 0: Criterion = .02146858
Iteration 1: Criterion = .02159149 (backed up)
Iteration 2: Criterion = .02177783
Iteration 3: Criterion = .02260102
Iteration 4: Criterion = .02267955
Iteration 5: Criterion = .02292361
Iteration 6: Criterion = .0243172
Iteration 7: Criterion = .02457057
Iteration 8: Criterion = .02488578
Iteration 9: Criterion = .02529419
Iteration 10: Criterion = .02545882
Iteration 11: Criterion = .02550251
Iteration 12: Criterion = .02552869
Iteration 13: Criterion = .02554463
Iteration 14: Criterion = .02554512
Iteration 15: Criterion = .02554514

Overidentification test for covariate balance
H0: Covariates are balanced
      chi2(5)      = 38.1464
      Prob > chi2   = 0.0000
```

We reject the null hypothesis that the treatment model balanced the covariates.

Let’s use `tebalance summarize` to see where the problem lies. To get an idea of the extent to which the covariates are unbalanced, we begin by summarizing the covariates by group in the raw data by specifying the baseline option.

```
. tebalance summarize, baseline
Covariate balance summary
```

	Raw	Weighted
Number of obs =	4,642	4,642.0
Treated obs =	864	2,315.3
Control obs =	3,778	2,326.7

	Means		Variances	
	Control	Treated	Control	Treated
mmarried	.7514558	.4733796	.1868194	.2495802
mage	26.81048	25.16667	31.87141	28.10429
prenatal1	.8268925	.6898148	.1431792	.2142183
fbaby	.4531498	.3715278	.2478707	.2337654

The output indicates that the covariates may not be balanced in the raw data. For example, the distribution of the mother’s age may differ over the treatment groups. We can investigate the differences further with standardized differences and variance ratios. A perfectly balanced covariate has a standard-

ized difference of zero and variance ratio of one. There are no standard errors on these statistics, so inference is informal. [Austin \(2009\)](#) provides a recent introduction to these diagnostics, although they have been used at least since [Rosenbaum and Rubin \(1985\)](#).

By omitting the baseline option, we obtain these diagnostic statistics for the raw data and the weighted data.

```
. tebalance summarize
```

Covariate balance summary

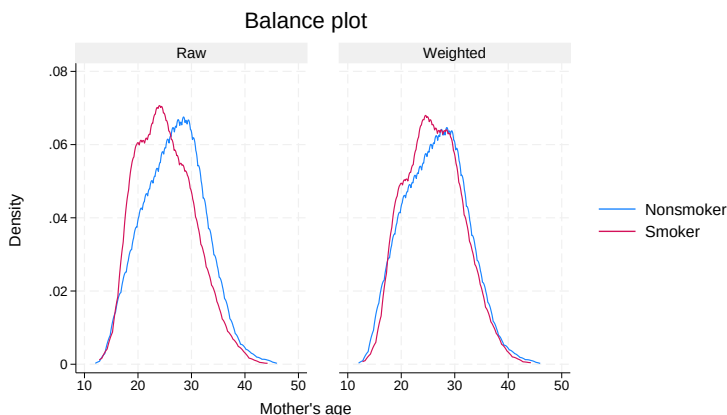
	Raw	Weighted
Number of obs =	4,642	4,642.0
Treated obs =	864	2,315.3
Control obs =	3,778	2,326.7

	Standardized differences		Variance ratio	
	Raw	Weighted	Raw	Weighted
mmarried	-.5953009	-.0105562	1.335944	1.009079
mage	-.300179	-.0672115	.8818025	.8536401
prenatal1	-.3242695	-.0156339	1.496155	1.023424
fbaby	-.1663271	.0257705	.9430944	1.005698

Reviewing the output, we see that for `mmarried`, `prenatal1`, and `fbaby`, our model improved the level of balance. The weighted standardized differences are all close to zero and the variance ratios are all close to one. However, output indicates that `mage` may not be balanced by our model. The weighted standardized difference is close to zero, but the weighted variance ratio still appears to be considerably less than one.

Now, let's use `tebalance density` to look at how the densities of `mage` for treated and control groups differ.

```
. tebalance density mage
```



The plots also indicate a lack of balance in `mage` between the treatment groups.

To try to achieve better balance, we specify a richer model with interactions between mage and the other covariates and look at the resulting standardized differences.

```
. quietly teffects ipw (bweight) (mbsmoke mmarried mage prenatal1 fbaby
> c.mage#(c.mage i.mmarried prenatal1))
. tebalance summarize
```

Covariate balance summary

	Raw	Weighted
Number of obs =	4,642	4,642.0
Treated obs =	864	2,329.1
Control obs =	3,778	2,312.9

	Standardized differences		Variance ratio	
	Raw	Weighted	Raw	Weighted
mmarried	-.5953009	.0053497	1.335944	.9953184
mage	-.300179	.0410889	.8818025	1.076571
prenatal1	-.3242695	.0009807	1.496155	.9985165
fbaby	-.1663271	-.0130638	.9430944	.9965406
mage#				
mage	-.3028275	.0477465	.8274389	1.109134
mmarried#				
mage				
Married	-.6329701	.0197209	1.157026	1.034108
prenatal1#				
mage				
Yes	-.4053969	.0182109	1.226363	1.032561

The standardized difference and variance ratio results for mage look closer to the expected values of zero and one, so we proceed to the formal test.

```
. tebalance overid
Iteration 0: Criterion = .0602349
Iteration 1: Criterion = .06172749 (backed up)
Iteration 2: Criterion = .06428188 (backed up)
Iteration 3: Criterion = .06489132 (backed up)
Iteration 4: Criterion = .06527353 (backed up)
Iteration 5: Criterion = .0664376
Iteration 6: Criterion = .07184528
Iteration 7: Criterion = .0762921
Iteration 8: Criterion = .0767141
Iteration 9: Criterion = .07679677
Iteration 10: Criterion = .07699122
Iteration 11: Criterion = .0776527
Iteration 12: Criterion = .07771774
Iteration 13: Criterion = .07772609
Iteration 14: Criterion = .07773294
Iteration 15: Criterion = .077752
Iteration 16: Criterion = .07775324
Iteration 17: Criterion = .07775325
Iteration 18: Criterion = .07775325

Overidentification test for covariate balance
H0: Covariates are balanced
      chi2(8)      = 11.8612
      Prob > chi2   = 0.1575
```

We do not reject the null hypothesis that the specified treatment model balances the covariates.



► Example 2: Balance after estimators that use matching

Instead of weighting, we might want to use a matching estimator. We can select `teffects nnmatch` or `teffects psmatch` for balance and estimation; in this example, we use `teffects nnmatch`.

```
. teffects nnmatch (bweight mmarried mage prenatal1 fbaby)
> (mbsmoke), bias(mage) ematch(mmarried prenatal1 fbaby)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : nearest-neighbor matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Distance metric: Mahalanobis                      max =     139
```

bweight	AI robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke						
(Smoker						
vs						
Nonsmoker)	-240.4589	28.43008	-8.46	0.000	-296.1808	-184.7369

Again we ignore the estimated effect and first check for balance. We begin by reviewing whether the summary statistics indicate good balance.

```
. tebalance summarize
(refitting the model using the generate() option)
Covariate balance summary
```

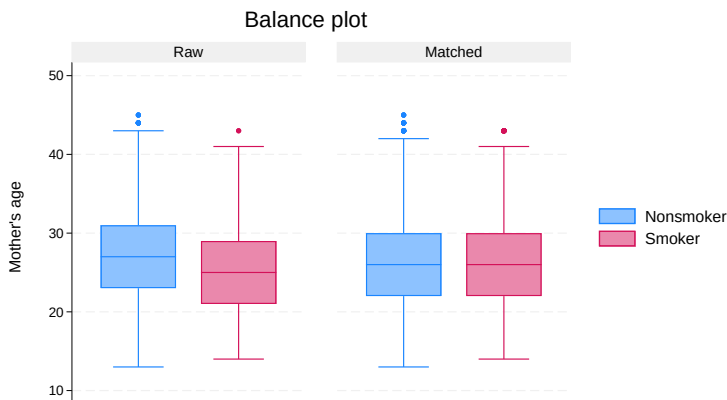
	Raw	Matched
Number of obs =	4,642	9,284
Treated obs =	864	4,642
Control obs =	3,778	4,642

	Standardized differences		Variance ratio	
	Raw	Matched	Raw	Matched
mmarried	-.5953009	-2.42e-16	1.335944	1
mage	-.300179	-.0040826	.8818025	.9815517
prenatal1	-.3242695	-2.78e-16	1.496155	1
fbaby	-.1663271	2.24e-16	.9430944	1

We do not have standard errors on these statistics, so we cannot make any formal conclusions, but the summary statistics appear to indicate much better balance than the IPW results. `tebalance summarize` has to refit the model to recover the matched sample because the `generate()` option was not specified on the `teffects nnmatch` command. The reestimation does not affect the results, although the computation takes longer; see [example 3](#) for details.

Because it is a matching estimator, and not an IPW estimator, we cannot use `tebalance overid` after `teffects nnmatch`. The matching estimators, however, provide diagnostic box plots using `tebalance box` that are not available after the IPW estimators.

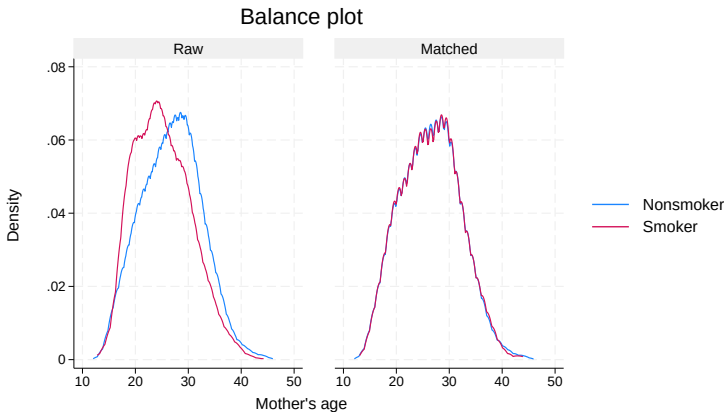
```
. tebalance box mage
(refitting the model using the generate() option)
```



The box plots of the matched data also indicate covariate balance.

Let's look at the kernel density plots using the matched data.

```
. tebalance density mage
(refitting the model using the generate() option)
```



The plots using the matched data appear to be balanced.



□ Technical note

`teffects` implements matching estimators, IPW estimators, regression-adjustment (RA) estimators, and estimators that combine IPW and RA. Matching estimators define a matched sample, and IPW estimators define a weighted sample, each of which can be used to compute covariate balance statistics. RA estimators do not define an adjusted sample that can be used to compute covariate balance statistics, and `tebalance` does not work after `teffects ra`. Only the IPW component of the estimators that combine RA and IPW defines a weighted sample that can be used to compute balance statistics. So, `tebalance` produces the same results after `teffects aipw` or `teffects ipwra` as it does after `teffects ipw`.



▷ Example 3: Faster results after a matching estimator

The `tebalance` commands run in [example 2](#) executed more slowly than necessary. `tebalance` issued the note

```
refitting the model using the generate() option
```

after the commands

```
. tebalance summarize
. tebalance box mage
```

and

```
. tebalance density mage
```

After `teffects nnmatch` or `teffects psmatch`, `tebalance` computes the balance statistics on the matched sample defined by the matching estimator. `teffects nnmatch` and `teffects psmatch` leave behind only variables that identify the matched sample when the `generate()` option is specified. Unless the `generate()` option is specified with `teffects nnmatch` or `teffects psmatch`, each `tebalance` command must rerun the estimation command to recover the matched sample.

Typing

```
. teffects nnmatch (bweight mmarried mage fbaby prenatal1)
> (mbsmoke), bias(mage) ematch(mmarried fbaby prenatal1)
> generate(matchv)
```

would generate variables that identify the matched sample that the `tebalance` commands could use. See *Remarks and examples* in [\[CAUSAL\] tebalance box](#), [\[CAUSAL\] tebalance density](#), and [\[CAUSAL\] tebalance summarize](#) for examples using the option `generate()` on `teffects psmatch` to speed up the postestimation computations.



Methods and formulas

Methods and formulas are presented under the following headings:

[Introduction](#)
[Matched samples](#)
[IPW samples](#)
[Testing the propensity-score model specification](#)

Introduction

For covariate z , we observe values $\{z_1, z_2, \dots, z_N\}$. Define a treatment indicator variable for J treatment levels as $t_i \in \{1, 2, \dots, J\}$, for $i = 1, \dots, N$, and frequency weights as $\{w_1, w_2, \dots, w_N\}$. The sample mean and variance of z for level t are

$$\hat{\mu}_z(t) = \frac{\sum_i^N I(t_i = t) w_i z_i}{N_t} \quad \text{and}$$

$$\hat{\sigma}_z^2(t) = \frac{\sum_i^N I(t_i = t) w_i \{z_i - \hat{\mu}_z(t)\}^2}{N_t - 1}$$

where $N_t = \sum_i^N w_i I(t_i = t)$, and

$$I(t_i = t) = \begin{cases} 1 & \text{if } t_i = t \\ 0 & \text{otherwise} \end{cases}$$

As shown in [Austin \(2011\)](#), the standardized differences for covariate z between level t and the control t_0 are computed as

$$\delta_z(t) = \frac{\hat{\mu}_z(t) - \hat{\mu}_z(t_0)}{\sqrt{\frac{\hat{\sigma}_z^2(t) + \hat{\sigma}_z^2(t_0)}{2}}} \quad (1)$$

The variance ratio is $\rho_z(t) = \{\hat{\sigma}_z^2(t)\} / \{\hat{\sigma}_z^2(t_0)\}$.

Matched samples

We now turn our attention to the matched samples for the potential-outcome mean (POM), average treatment effect (ATE), and average treatment effect on the treated (ATET) estimators. We estimate the covariate for the counter-factual treatment by taking the mean of the matched observations

$$\dot{z}_i = \frac{\sum_{j \in \Omega(i)} w_j z_j}{\sum_{j \in \Omega(i)} w_j}$$

where $\Omega(i) = (k_1, k_2, \dots, k_{m_j})$ is the set of observation indices that are matched to observation i of the opposite treatment condition. The observed covariate and matched covariate pairs, $(z_i, \dot{z}_i)$, $i = 1, \dots, N$, are used in the box plot (see [G-2] **graph box**) and the kernel density plot (see [R] **kdensity**). The ATET sample is limited to those observations from the conditional treatment, $\tilde{t}$, and their matched covariate means.

In *Methods and formulas* of [CAUSAL] **teffects nnmatch**, we define $K_m(i)$ as the number of times observation i is used in a match with observation j of the opposite treatment condition, $i \in \Omega(j)$, weighted by the total number of matches for observation j . Specifically,

$$K_m(i) = \sum_{j=1}^N I\{i \in \Omega(j)\} \frac{w_j}{\sum_{k \in \Omega(j)} w_k}$$

These weights are used in the estimation of the mean and variance for the matched dataset. For the POM and ATE models, the estimated mean and variance are computed as

$$\begin{aligned} \hat{\mu}_{\dot{z}}(t) &= \frac{\sum_i^N I(t_i = t) w_i z_i \{1 + K_m(i)\}}{M_t} \quad \text{and} \\ \hat{\sigma}_{\dot{z}}^2(t) &= \frac{\sum_i^N I(t_i = t) w_i \{1 + K_m(i)\} \{z_i - \hat{\mu}_{\dot{z}}(t)\}^2}{M_t - 1} \end{aligned}$$

where $M_t = \sum_i^N I(t_i = t) w_i \{1 + K_m(i)\}$.

The standardized differences between the control level and all other levels for the matched covariate distribution are computed as in (1), but $\hat{\mu}_{\dot{z}}(t)$ is substituted for $\hat{\mu}_z(t)$ and $\hat{\sigma}_{\dot{z}}^2(t)$ for $\hat{\sigma}_z^2(t)$.

For the ATET model, there is no matched sample for the treatment levels other than the conditional treatment $\tilde{t}$. The covariate mean and variance for the conditional treatment are the same as that of the raw data, $\mu_z(\tilde{t})$ and $\sigma_z^2(\tilde{t})$. However, the covariate mean and variance for the sample matched to the conditional treatment, $t \neq \tilde{t}$, are computed as

$$\begin{aligned} \tilde{\mu}_{\dot{z}}(t) &= \frac{\sum_i^N I(t_i = t) w_i z_i K_m(i)}{M_t} \quad \text{and} \\ \tilde{\sigma}_{\dot{z}}^2(t) &= \frac{\sum_i^N I(t_i = t) w_i K_m(i) \{z_i - \tilde{\mu}_{\dot{z}}(t)\}^2}{M_t - 1} \end{aligned}$$

where $M_t = \sum_i^N I(t_i = t) w_i K_m(i)$.

IPW samples

Computation of the inverse-probability weights is discussed in [Methods and formulas](#) of [\[CAUSAL\] teffects aipw](#) and in [Methods and formulas](#) of [\[CAUSAL\] stteffects ipwra](#). For the POM and ATE estimators, we defined the normalized IPW weights as $\bar{d}_i(t) = N_t d_i(t) / \sum_i^N d_i(t)$, where $d_i(t) = I(t_i = t) / p(\mathbf{z}_i, t, \hat{\gamma})$ for treatment level t and individual i .

For the ATET estimator, we use the normalized weights $\bar{f}_i = N f_i / \sum_i^N f_i$, where $f_i = p(\mathbf{z}_i, \tilde{t}, \hat{\gamma}) / p(\mathbf{z}_i, t_i, \hat{\gamma})$ are the treatment-adjusted inverse-probability weights, and $\tilde{t}$ is the conditional treatment.

We will simplify notation by defining a single weight

$$\bar{w}_i(t) = \begin{cases} \bar{d}_i(t) & \text{if model is ATE or POM} \\ \bar{f}_i(t) & \text{if model is ATET} \end{cases}$$

The covariate mean and variance for treatment level t are

$$\begin{aligned} \tilde{\mu}_{\tilde{z}}(t) &= \frac{\sum_i^N I(t_i = t) w_i \bar{w}_i x_i}{M_t} \quad \text{and} \\ \tilde{\sigma}_{\tilde{z}}^2(t) &= \frac{I(t_i = t) w_i \bar{w}_i \{z_i - \tilde{\mu}_{\tilde{z}}(t)\}^2}{M_t - 1} \end{aligned}$$

where $M_t = \sum_i^N I(t_i = t) w_i \bar{w}_i$.

The kernel density is computed by [kdensity](#) for each covariate conditioned on each treatment level using the raw covariate with `iweights` equal to $w_i \bar{w}_i$.

Testing the propensity-score model specification

We estimate the probability of treatment conditioned on a set of covariates with a propensity-score model. [Imai and Ratkovic \(2014\)](#) derive a test for whether the estimated propensity score balances the covariates. The score equations for parameters of the propensity-score model define an exactly identified generalized method of moments (GMM) estimator. [Imai and Ratkovic \(2014\)](#) use the conditions imposed by mean balance as overidentifying conditions. A standard GMM test for the validity of the overidentifying conditions is then a test for covariate balance. See [\[R\] gmm](#) for a discussion of this overidentifying test, which is known as Hansen's J test in the econometrics literature.

Here are the details about the score equations and the overidentifying balance conditions. Recall from [Methods and formulas](#) of [\[CAUSAL\] teffects aipw](#) and [Methods and formulas](#) of [\[CAUSAL\] stteffects ipwra](#), we have the first-order condition of the treatment model

$$\frac{1}{N} \sum_{i=1}^N \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \hat{\gamma}) = 0$$

For a two-level treatment-effects model with conditional treatment $\tilde{t}$ and control t_0 , the score is

$$\mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \gamma) = \frac{I(t_i = \tilde{t})}{p(\mathbf{z}_i, \tilde{t}, \gamma)} \frac{\partial p(\mathbf{z}_i, t, \gamma)}{\partial \gamma'} - \left\{ \frac{I(t_i = t_0)}{1 - p(\mathbf{z}_i, \tilde{t}, \gamma)} \right\} \frac{\partial p(\mathbf{z}_i, \tilde{t}, \gamma)}{\partial \gamma'} \Bigg|_{\gamma = \hat{\gamma}}$$

The score reduces to

$$\mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \hat{\gamma}) = \left[\frac{I(t_i = \tilde{t}) - p(\mathbf{z}_i, \tilde{t}, \gamma)}{p(\mathbf{z}_i, \tilde{t}, \gamma) \{1 - p(\mathbf{z}_i, \tilde{t}, \gamma)\}} \right] \frac{\partial p(\mathbf{z}_i, \tilde{t}, \gamma)}{\partial \gamma'} \Big|_{\gamma = \hat{\gamma}}$$

The corresponding covariate balancing moment conditions are

$$\mathbf{w}_{\text{tm},i}(\mathbf{z}_i, \gamma) = \left[\frac{I(t_i = \tilde{t}) - p(\mathbf{z}_i, \tilde{t}, \gamma)}{p(\mathbf{z}_i, \tilde{t}, \gamma) \{1 - p(\mathbf{z}_i, \tilde{t}, \gamma)\}} \right] \mathbf{z}_i$$

for the POM and ATE models. For the ATET model with conditional treatment $\tilde{t}$, we multiply by $p(\mathbf{z}_i, \tilde{t}, \gamma)$ and scale by $N/N_{\tilde{t}}$:

$$\mathbf{w}_{\text{tm},i}(\mathbf{z}_i, \gamma) = \frac{N}{N_{\tilde{t}}} \left\{ \frac{I(t_i = \tilde{t}) - p(\mathbf{z}_i, \tilde{t}, \gamma)}{1 - p(\mathbf{z}_i, \tilde{t}, \gamma)} \right\} \mathbf{z}_i$$

We stack the moment conditions

$$\begin{aligned} \mathbf{g}_{\text{tm}}(\mathbf{Z}, \gamma) &= \frac{1}{N} \sum_{i=1}^N \begin{Bmatrix} \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \gamma) \\ \mathbf{w}_{\text{tm},i}(\mathbf{z}_i, \gamma) \end{Bmatrix} \\ &= \frac{1}{N} \sum_{i=1}^N \mathbf{g}_{\text{tm},i}(\mathbf{z}_i, \gamma) \end{aligned}$$

The overidentified GMM estimator is then

$$\tilde{\gamma} = \underset{\gamma}{\operatorname{argmin}} N \mathbf{g}_{\text{tm}}(\mathbf{Z}, \gamma)' \mathbf{W}_{\text{tm}}(\mathbf{Z}, \gamma)^{-1} \mathbf{g}_{\text{tm}}(\mathbf{Z}, \gamma) \quad (2)$$

where

$$\mathbf{W}_{\text{tm}}(\mathbf{Z}, \gamma) = \frac{1}{N} \sum_{i=1}^N E_T \{ \mathbf{g}_{\text{tm},i}(\mathbf{z}, \gamma) \mathbf{g}_{\text{tm},i}(\mathbf{z}, \gamma)' \}$$

and the expectation is taken with respect to treatment distribution. The weight matrix $\mathbf{W}_{\text{tm}}(\mathbf{Z}, \gamma)$ is computed explicitly (Imai and Ratkovic 2014), and (2), written as a maximization problem, is solved using [m1](#).

Finally, Hansen's J statistic is evaluated at its minimum,

$$J = N \mathbf{g}_{\text{tm}}(\mathbf{Z}, \tilde{\gamma})' \mathbf{W}_{\text{tm}}(\mathbf{Z}, \tilde{\gamma})^{-1} \mathbf{g}_{\text{tm}}(\mathbf{Z}, \tilde{\gamma})$$

and is asymptotically distributed as a χ^2 with degrees of freedom d ,

$$d = \operatorname{rank} \{ \mathbf{W}_{\text{tm}}(\mathbf{Z}, \tilde{\gamma}) \} - \operatorname{rank} \left[\frac{1}{N} \sum_{i=1}^N E_T \{ \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \tilde{\gamma}) \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \tilde{\gamma})' \} \right]$$

References

- Austin, P. C. 2009. Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. *Statistics in Medicine* 28: 3083–3107. <https://doi.org/10.1002/sim.3697>.
- . 2011. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate Behavioral Research* 46: 399–424. <https://doi.org/10.1080/00273171.2011.568786>.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Guo, S., and M. W. Fraser. 2015. *Propensity Score Analysis: Statistical Methods and Applications*. 2nd ed. Thousand Oaks, CA: Sage.
- Imai, K., and M. Ratkovic. 2014. Covariate balancing propensity score. *Journal of the Royal Statistical Society, B ser.*, 76: 243–263. <https://doi.org/10.1111/rssb.12027>.
- Jann, B. 2021. [Relative distribution analysis in Stata](#). *Stata Journal* 21: 885–951.
- Rosenbaum, P. R., and D. B. Rubin. 1985. Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *American Statistician* 39: 33–38. <https://doi.org/10.2307/2683903>.
- Rubin, D. B. 2008. For objective causal inference, design trumps analysis. *Annals of Applied Statistics* 2: 808–840. <https://doi.org/10.1214/08-AOAS187>.

Also see

- [CAUSAL] [stteffects intro](#) — Introduction to treatment effects for observational survival-time data
- [CAUSAL] [tebalance box](#) — Covariate balance box
- [CAUSAL] [tebalance density](#) — Covariate balance density
- [CAUSAL] [tebalance overid](#) — Test for covariate balance
- [CAUSAL] [tebalance summarize](#) — Covariate-balance summary statistics
- [CAUSAL] [teffects](#) — Treatment-effects estimation for observational data

[Description](#)
[Options](#)[Quick start](#)
[Remarks and examples](#)[Menu](#)
[Reference](#)[Syntax](#)
[Also see](#)

Description

`tebalance box` produces box plots that are used to check for balance in matched samples after `teffects nnmatch` and `teffects psmatch`.

Quick start

Box plot of the propensity score from the last `teffects psmatch` command

```
tebalance box
```

Box plot of values of `x1` in the treatment and control groups from raw data and the matched sample from the last `teffects nnmatch` or `teffects psmatch` command

```
tebalance box x1
```

Menu

Statistics > Postestimation

Syntax

Box plots for the propensity score

```
tebalance box [ , options ]
```

Box plots for a covariate

```
tebalance box varname [ , options ]
```

options	Description
Main	
<i>boxlook_options</i>	graph box options controlling how the box looks
<i>legending_options</i>	graph box options controlling how the variables are labeled
<i>axis_options</i>	graph box options controlling how numerical <i>y</i> axis is labeled
<i>title_and_other_options</i>	graph box options controlling titles, added text, aspect ratio, etc.
<i>by_options</i>	suboptions inside <i>by()</i> controlling plots by raw and matched samples

Options

Main

boxlook_options are any of the options documented in *boxlook_options* in [G-2] **graph box**.
legending_options are any of the options documented in *legending_options* in [G-2] **graph box**.
axis_options are any of the options documented in *axis_options* in [G-2] **graph box**.
title_and_other_options are any of the options, except *by()*, documented in *title_and_other_options* in [G-2] **graph box**. *tebalance box* uses *by()* to differentiate between raw and matched samples, and some *twoway_options* will be repeated for *by graph* and might be better specified as *byopts()*.
by_options are any of the *byopts* documented in [G-3] **by-option**. *byopts()* generally affects the entire graph, and some *by_options* may be better specified as *twoway_options*; see [G-3] **twoway_options**.

Remarks and examples

When the distribution of a covariate does not vary over the treatment levels, the covariate is said to be balanced. *tebalance box* produces box plots of a covariate over treatment levels for the raw data and for the matched sample produced by *teffects*. If the matched-sample box plots are the same over the treatment levels, the covariate is balanced in the matched sample.

After *teffects nnmatch* and *teffects psmatch*,

```
. tebalance box varname [ , options ]
```

produces box plots to check whether *varname* is balanced.

After `teffects psmatch`,

```
. tebalance box [ , options ]
```

produces box plots to check whether the propensity score estimated by `teffects` is balanced.

We recommend that you read [CAUSAL] **tebalance** before proceeding; it provides an introduction to covariate balance and an overview of the implemented methods.

➤ Example 1: Checking covariate balance after `psmatch`

Using an extract from the data used by Cattaneo (2010), we use `teffects psmatch` to estimate the effect of a mother’s smoking behavior (`mbsmoke`) on the birthweight of her child (`bweight`), controlling for marital status (`mmarried`), the mother’s age (`mage`), whether the mother had a prenatal doctor’s visit in the baby’s first trimester (`prenatal1`), and whether this baby is the mother’s first child (`fbaby`).

```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)

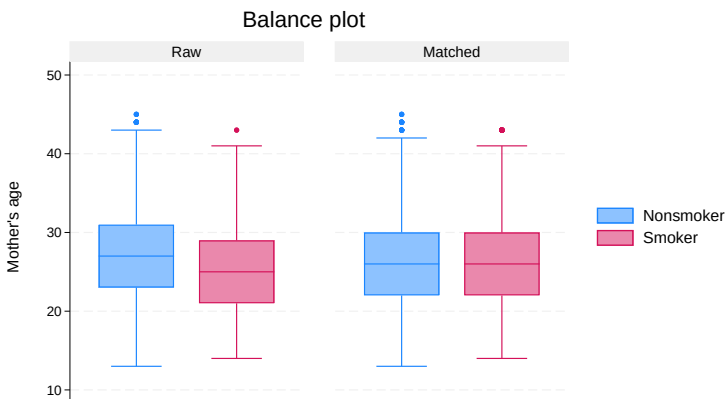
. teffects psmatch (bweight) (mbsmoke mmarried mage prenatal1 fbaby),
> generate(matchv)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : propensity-score matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Treatment model: logit                      max =     139
```

bweight	AI robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke						
(Smoker						
vs						
Nonsmoker)	-235.1714	27.74409	-8.48	0.000	-289.5488	-180.794

We specified the option `generate(matchv)` to speed up the postestimation command that produces density plots, as discussed in example 3 under *Remarks and examples* of [CAUSAL] **tebalance**. We do not interpret the estimated effect produced by this preliminary model but rather check the specification. Now we look at the box plots.

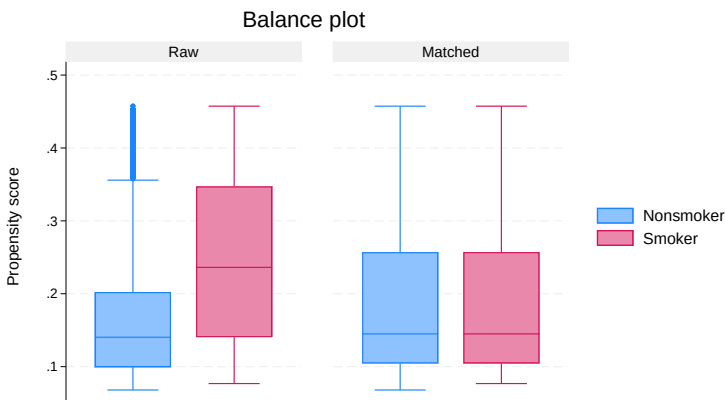
```
. tebalance box mage
```



The box plots for the matched sample are very similar. The medians, the 25th percentiles, and the 75th percentiles appear to be the same, although there may be some differences in the tails, the upper adjacent values, the lower adjacent values, and the outliers. Matching on the estimated propensity score appears to have balanced `mage`, except for the tails.

To get an idea of whether `teffects psmatch` balanced all the covariates, we look at the box plots for the estimated propensity score.

```
. tebalance box
```



The box plots indicate that `teffects psmatch` balanced the estimated propensity scores.

◀

Reference

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.

Also see

[CAUSAL] **tebalance** — Check balance after teffects or stteffects estimation

[CAUSAL] **teffects nnmatch** — Nearest-neighbor matching

[CAUSAL] **teffects psmatch** — Propensity-score matching

[CAUSAL] **teoverlap** — Overlap plots

[Description](#)
[Options](#)
[Also see](#)[Quick start](#)
[Remarks and examples](#)[Menu](#)
[Stored results](#)[Syntax](#)
[Reference](#)

Description

`tebalance density` produces kernel density plots that are used to check for covariate balance after estimation by a [teffects](#) inverse-probability-weighted estimator, a `teffects` matching estimator, or an `stteffects` inverse-probability-weighted estimator.

Quick start

Kernel density plot of the propensity score after `teffects psmatch`

```
tebalance density
```

Kernel density plot of `x1` after a `teffects` command or an `stteffects` command

```
tebalance density x1
```

Same as above, but rescale the kernel bandwidth by a factor of 2

```
tebalance density x1, bwidth(*2)
```

Menu

Statistics > Postestimation

Syntax

Density plots for the propensity score

```
tebalance density [ , options ]
```

Density plots for a covariate

```
tebalance density varname [ , options ]
```

<i>options</i>	Description
Main	
<code>kernel</code> (<i>kernel</i>)	specify the kernel function; default is <code>kernel(epanechnikov)</code>
<code>bwidth</code> (<i>*#</i>)	rescale default bandwidth
<code>line#opts</code> (<i>line_options</i>)	twoway line options for density line number #
<i>twoway_options</i>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>
<code>byopts</code> (<i>by_option</i>)	how subgraphs are combined, labeled, etc.

collect is allowed; see [U] 11.1.10 Prefix commands.

<i>kernel</i>	Description
<code>epanechnikov</code>	Epanechnikov kernel function; the default
<code>triangle</code>	triangle kernel function
<code>epan2</code>	alternative Epanechnikov kernel function
<code>biweight</code>	biweight kernel function
<code>cosine</code>	cosine trace kernel function
<code>gaussian</code>	Gaussian kernel function
<code>parzen</code>	Parzen kernel function
<code>rectangle</code>	rectangle kernel function

Options

Main

`kernel` (*kernel*) specifies the kernel function for use in calculating the kernel density estimates. The default kernel is `kernel(epanechnikov)`.

`bwidth` (**#*) specifies the factor by which the default bandwidths are to be rescaled. A bandwidth is the half-width of the kernel, the width of the density window around each point. Each kernel density plot has its own bandwidth, and by default, each kernel density plot uses its own optimal bandwidth; see [R] `kdensity`. `bwidth()` rescales each plot's optimal bandwidth by the specified amount.

`line#opts` (*line_options*) specifies the line pattern, width, color, and overall style of density line number #. The line numbers are in the same order as the treatment levels specified in `e(tlevels)`.

twoway_options are any of the options documented in [G-3] *twoway_options*, excluding `by()`. These include options for titling the graph (see [G-3] *title_options*) and for saving the graph to disk (see [G-3] *saving_option*). `tebalance density` uses `by()` to differentiate between raw and weighted or matched samples, and some *twoway_options* will be repeated for `by` graph and might be better specified as `byopts()`.

`byopts(by_option)` is as documented in [G-3] [by_option](#). `byopts()` affects how the subgraphs are combined, labeled, etc. `byopts()` generally affects the entire graph, and some *by_options* may be better specified as *twoway_options*; see [G-3] [twoway_options](#).

Remarks and examples

When the distribution of a covariate does not vary over the treatment levels, the covariate is said to be balanced. `tebalance density` produces kernel density plots of a covariate over treatment levels for the raw data and the weighted or matched sample produced by `teffects` or `stteffects`. If the weighted-sample or matched-sample kernel density plots of the covariate are the same over the treatment levels, the covariate is balanced in the weighted or matched sample.

After `telasso` and all `teffects` and `stteffects` commands except `teffects ra`, `stteffects ra`, and `stteffects wra`,

```
. tebalance density varname [ , options ]
```

produces kernel density plots to check whether *varname* is balanced.

After `teffects psmatch`,

```
. tebalance density [ , options ]
```

produces kernel density plots to check whether the propensity score estimated by `teffects psmatch` is balanced. Our discussion of the use of `tebalance density` and interpretation of its results for a covariate below also apply to a propensity score.

We recommend that you read [CAUSAL] [tebalance](#) before proceeding; it provides an introduction to covariate balance and an overview of the implemented methods. See [CAUSAL] [stteffects intro](#) for a discussion of survival-time features.

► Example 1: Checking covariate balance after `psmatch`

Using an extract from the data used by [Cattaneo \(2010\)](#), we use `teffects psmatch` to estimate the effect of a mother's smoking behavior (`mbsmoke`) on the birthweight of her child (`bweight`), controlling for marital status (`mmarried`), the mother's age (`mage`), whether the mother had a prenatal doctor's visit in the baby's first trimester (`prenatal1`), and whether this baby is the mother's first child (`fbaby`).

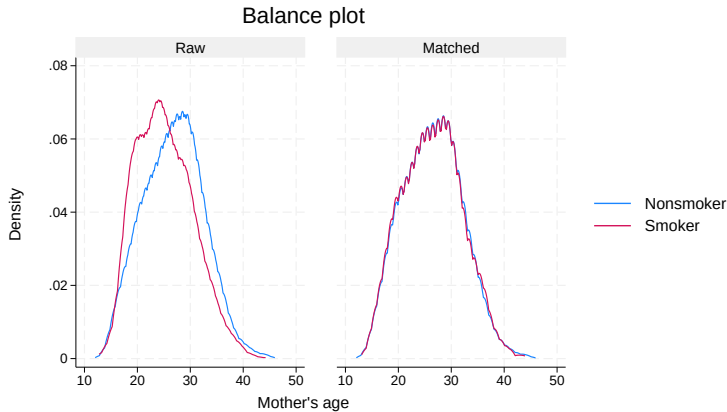
```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects psmatch (bweight) (mbsmoke mmarried mage prenatal1 fbaby),
> generate(matchv)
```

```
Treatment-effects estimation      Number of obs      =      4,642
Estimator      : propensity-score matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Treatment model: logit                        max =     139
```

bweight	AI robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke						
(Smoker						
vs						
Nonsmoker)	-235.1714	27.74409	-8.48	0.000	-289.5488	-180.794

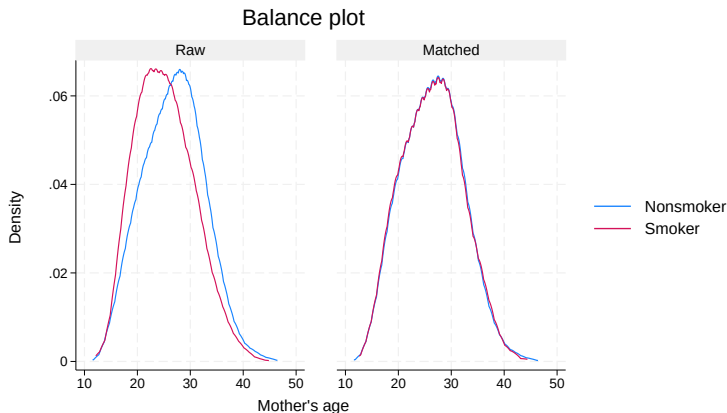
We specified the option `generate(matchv)` to speed up the postestimation command that produces density plots, as discussed in [example 3](#) under *Remarks and examples* in [\[CAUSAL\] tebalance](#) entry. We do not interpret the estimated effect produced by this preliminary model but rather check the specification. We begin by looking at the default density plots.

```
. tebalance density mage
```



The density plots for the matched sample are nearly indistinguishable, implying that matching on the estimated propensity score balanced the covariates. The density plots are too jagged for presentation, so we oversmooth them by scaling up the bandwidth used for each plot.

```
. tebalance density mage, bwidth(*1.5)
```



Option `bwidth()` rescales the default optimal bandwidths by the specified scale factor. Each of the four density plots has its own sample size and optimal bandwidth. Rescaling each of the four bandwidths by 1.5 produces smoother plots.

Stored results

After `teffects` or `stteffects` fits a binary treatment, `tebalance density` stores the following in `r()`:

Scalars

<code>r(bwc_adj)</code>	bandwidth for control in weighted or matched-adjusted sample
<code>r(Nc_adj)</code>	observations on control in weighted or matched-adjusted sample
<code>r(bwt_adj)</code>	bandwidth for treated in weighted or matched-adjusted sample
<code>r(Nt_adj)</code>	observations on treated in weighted or matched-adjusted sample
<code>r(bwc_raw)</code>	bandwidth for control in raw sample
<code>r(Nc_raw)</code>	observations on control in raw sample
<code>r(bwt_raw)</code>	bandwidth for treated in raw sample
<code>r(Nt_raw)</code>	observations on treated in raw sample

Macros

<code>r(kernel)</code>	name of kernel
------------------------	----------------

After `teffects` or `stteffects` fits a multivalued treatment, `tebalance density` stores the following in `r()`:

Scalars

<code>r(bw#_adj)</code>	bandwidth for treatment level # in weighted or matched-adjusted sample
<code>r(N#_adj)</code>	observations on treatment level # in weighted or matched-adjusted sample
<code>r(bw#_raw)</code>	bandwidth for treatment level # in raw sample
<code>r(N#_raw)</code>	observations on treatment level # in raw sample

Macros

<code>r(kernel)</code>	name of kernel
------------------------	----------------

Reference

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.

Also see

- [CAUSAL] [stteffects intro](#) — Introduction to treatment effects for observational survival-time data
- [CAUSAL] [tebalance](#) — Check balance after `teffects` or `stteffects` estimation
- [CAUSAL] [teffects aipw](#) — Augmented inverse-probability weighting
- [CAUSAL] [teffects ipw](#) — Inverse-probability weighting
- [CAUSAL] [teffects ipwra](#) — Inverse-probability-weighted regression adjustment
- [CAUSAL] [teffects nnmatch](#) — Nearest-neighbor matching
- [CAUSAL] [teffects psmatch](#) — Propensity-score matching
- [CAUSAL] [teoverlap](#) — Overlap plots

Description
Options
Also see

Quick start
Remarks and examples

Menu
Stored results

Syntax
References

Description

`tebalance overid` performs a test for covariate balance after estimation by a `teffects` inverse-probability-weighted (IPW) estimator or an `stteffects` IPW estimator.

Quick start

Test for covariate balance after a `teffects` or an `stteffects` IPW estimator

```
tebalance overid
```

Same as above, but test for balance only in base covariates and exclude interaction terms

```
tebalance overid, bonly
```

Menu

Statistics > Postestimation

Syntax

```
tebalance overid [ , bconly nolog iterate(#) ]
```

collect is allowed; see [U] [11.1.10 Prefix commands](#).

Options

Main

`bconly` specifies that only the base covariates be included in the test for balance. By default, the powers and interactions specified by factor-variable notation in the `teffects` or `stteffects` model are also included in the test for balance.

`nolog` suppresses the display of the optimization search log.

`iterate(#)` sets the maximum number of iterations to `#` in the generalized method of moments estimator used to compute the test statistic.

Remarks and examples

When the distribution of a covariate is the same for all treatment levels, the covariate is said to be balanced. `tebalance overid` performs a test to see whether the covariates are balanced after `teffects` or `stteffects`. `tebalance overid` can be executed after `teffects ipw`, `teffects aipw`, `teffects ipwra`, `stteffects ipw`, or `stteffects ipwra`, which use the inverse-probability weights predicted by a treatment model to account for how treatment assignment depends on observed covariates. If the treatment model is well specified, IPW functions of the covariates from the model are balanced.

We recommend that you read [\[CAUSAL\] tebalance](#) before proceeding; it provides an introduction to covariate balance and an overview of the implemented methods. See [\[CAUSAL\] stteffects intro](#) for survival-time discussion and examples.

► Example 1: Base covariates and interactions

This example illustrates the interpretation of the `bconly` option, which excludes powers and interactions when factor variables are included in the propensity-score model.

We frequently use factor variables to include powers of, and interactions between, base covariates in our specification of the propensity-score model. In [example 1](#) under *Remarks and examples* in [\[CAUSAL\] tebalance](#), we rejected the null hypothesis of balance in a model using only base covariates but not in the richer model that included power and interaction terms. By default, `tebalance overid` tests whether the model balances the base covariates and the power-and-interaction covariates. When option `bconly` is specified, `tebalance overid` tests whether the model balances the base covariates only.

Using an extract from the data used by [Cattaneo \(2010\)](#), we use `teffects ipw` to estimate the effect of a mother's smoking behavior (`mbsmoke`) on the birthweight of her child (`bweight`), controlling for marital status (`mmarried`), the mother's age (`mage`), whether the mother had a prenatal doctor's visit in the baby's first trimester (`prenatal1`), and whether this baby is the mother's first child (`fbaby`). In addition to the base covariates, we include the square of `mage`, an interaction between `mage` and `mmarried`, and an interaction between `mage` and `prenatal1` in the model for the propensity score.

```

. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipw (bweight) (mbsmoke mmarried mage prenatal1 fbaby
> c.mage#(c.mage i.mmarried prenatal1)), aequations
Iteration 0: EE criterion = 9.365e-20
Iteration 1: EE criterion = 2.612e-26

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit

```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke (Smoker vs Nonsmoker)	-239.6875	26.43427	-9.07	0.000	-291.4977	-187.8773
POmean						
mbsmoke Nonsmoker	3403.638	9.56792	355.73	0.000	3384.885	3422.39
TME1						
mmarried	.8522468	.462536	1.84	0.065	-.0543072	1.758801
mage	.1742823	.0651039	2.68	0.007	.0466811	.3018836
prenatal1	.4018114	.4341762	0.93	0.355	-.4491584	1.252781
fbaby	-.4824413	.0868982	-5.55	0.000	-.6527587	-.3121239
c.mage# c.mage	-.002515	.0012585	-2.00	0.046	-.0049817	-.0000483
mmarried# c.mage Married	-.0787984	.0175508	-4.49	0.000	-.1131973	-.0443996
prenatal1# c.mage Yes	-.0286228	.0176391	-1.62	0.105	-.0631948	.0059492
_cons	-2.928851	.8409119	-3.48	0.000	-4.577008	-1.280694

We specified option `aequations` to see the parameter estimates for the coefficients in the propensity-score model. There are eight coefficients, five on the base covariates (`mmarried`, `mage`, `fbaby`, `prenatal1`, and `_cons`) and three on the power-and-interaction covariates (`c.mage#c.mage`, `c.mage#1.mmarried`, and `c.mage#1.prenatal1`). Below we test whether the model balances all eight covariates.


```
. tebalance overid
Iteration 0: Criterion = .0602349
Iteration 1: Criterion = .06172749 (backed up)
Iteration 2: Criterion = .06428188 (backed up)
Iteration 3: Criterion = .06489132 (backed up)
Iteration 4: Criterion = .06527353 (backed up)
Iteration 5: Criterion = .0664376
Iteration 6: Criterion = .07184528
Iteration 7: Criterion = .0762921
Iteration 8: Criterion = .0767141
Iteration 9: Criterion = .07679677
Iteration 10: Criterion = .07699122
Iteration 11: Criterion = .0776527
Iteration 12: Criterion = .07771774
Iteration 13: Criterion = .07772609
Iteration 14: Criterion = .07773294
Iteration 15: Criterion = .077752
Iteration 16: Criterion = .07775324
Iteration 17: Criterion = .07775325
Iteration 18: Criterion = .07775325

Overidentification test for covariate balance
H0: Covariates are balanced

      chi2(8)      = 11.8612
Prob > chi2      = 0.1575
```

We cannot reject the null hypothesis that the IPW model balanced all eight covariates.

Below we specify option `bconly` to test whether the IPW model balanced the five base covariates only.

```
. tebalance overid, bconly
Iteration 0: Criterion = .1079977
Iteration 1: Criterion = .10800825 (backed up)
Iteration 2: Criterion = .10844177 (backed up)
Iteration 3: Criterion = .10851215 (backed up)
Iteration 4: Criterion = .10860849 (backed up)
Iteration 5: Criterion = .10907447
Iteration 6: Criterion = .110098
Iteration 7: Criterion = .11163978
Iteration 8: Criterion = .11260697
Iteration 9: Criterion = .11286395
Iteration 10: Criterion = .11331486
Iteration 11: Criterion = .11333976
Iteration 12: Criterion = .11335604
Iteration 13: Criterion = .11335696
Iteration 14: Criterion = .11335696

Overidentification test for covariate balance
H0: Covariates are balanced

      chi2(5)      = 7.82169
Prob > chi2      = 0.1663
```

We cannot reject the null hypothesis that the IPW model balanced the five base covariates.

Each test has a justification.

In a model-based approach, the [Imai and Ratkovic \(2014\)](#) test checks whether the propensity score is correctly specified. We include all eight covariates because they must all be balanced, if the propensity-score model is correctly specified.

A conditional-independence approach can be used to justify only including the base covariates in the test. In this approach, the propensity-score model need only balance the base covariates. Powers and interactions of the base covariates are included to get a propensity-score model that balances the base covariates, but balance of these higher-order terms is more than what needs to be checked.

In large samples, both tests should have nominal coverage under the null hypothesis that the propensity-score model is correctly specified. Under the alternative that the propensity-score model is misspecified, including all the covariates should yield a test with higher power.

The test that includes all the covariates is the default.



Stored results

`tebalance overid` stores the following in `r()`:

Scalars

<code>r(p)</code>	<i>p</i> -value
<code>r(df)</code>	overidentifying constraints, test degrees of freedom
<code>r(chi2)</code>	χ^2 statistic

References

- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Imai, K., and M. Ratkovic. 2014. Covariate balancing propensity score. *Journal of the Royal Statistical Society, B ser.*, 76: 243–263. <https://doi.org/10.1111/rssb.12027>.

Also see

- [CAUSAL] [stteffects intro](#) — Introduction to treatment effects for observational survival-time data
- [CAUSAL] [tebalance](#) — Check balance after teffects or stteffects estimation
- [CAUSAL] [teffects aipw](#) — Augmented inverse-probability weighting
- [CAUSAL] [teffects ipw](#) — Inverse-probability weighting
- [CAUSAL] [teffects ipwra](#) — Inverse-probability-weighted regression adjustment
- [CAUSAL] [teoverlap](#) — Overlap plots

Description	Quick start	Menu	Syntax
Option	Remarks and examples	Stored results	Reference
Also see			

Description

`tebalance summarize` reports diagnostic statistics that are used to check for covariate balance over treatment groups after estimation by a `teffects` inverse-probability-weighted (IPW) estimator, a `teffects` matching estimator, or an `stteffects` IPW estimator.

Quick start

Raw and weighted standardized differences and variance ratios of all covariates from the most recently fitted `teffects` model or `stteffects` model

```
tebalance summarize
```

Same as above, but report statistics only for covariates `x1` and `x2`

```
tebalance summarize x1 x2
```

Baseline means and variances for treated and control groups

```
tebalance summarize, baseline
```

Menu

Statistics > Postestimation

Syntax

```
tebalance summarize [ varlist ] [ , baseline ]
```

varlist may contain factor variables; see [U] 11.4.3 Factor variables.

collect is allowed; see [U] 11.1.10 Prefix commands.

Option

Main

baseline specifies that `tebalance summarize` report means and variances by treatment level.

Remarks and examples

When the distribution of a covariate is the same for all treatment levels, the covariate is said to be balanced. `tebalance summarize` reports diagnostic statistics to check for covariate balance after `teffects` or `stteffects`. `tebalance summarize` can be executed after all `teffects` estimators with the exception of `teffects ra` and executed after `stteffects ipw` and `stteffects ipwra`.

We recommend that you read [CAUSAL] [tebalance](#) before proceeding; it provides an introduction to covariate balance and an overview of the implemented methods. See [CAUSAL] [stteffects intro](#) for survival-time discussion and examples.

► Example 1: Checking covariate balance after `psmatch`

Using an extract from the data used by Cattaneo (2010), we use `teffects psmatch` to estimate the effect of a mother's smoking behavior (`mbsmoke`) on the birthweight of her child (`bweight`), controlling for marital status (`mmarried`), the mother's age (`mage`), whether the mother had a prenatal doctor's visit in the baby's first trimester (`prenatal1`), and whether this baby is the mother's first child (`fbaby`).

```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)

. teffects psmatch (bweight) (mbsmoke mmarried mage prenatal1 fbaby),
> generate(matchv)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : propensity-score matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Treatment model: logit                        max =     139
```

bweight	AI robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke						
(Smoker						
vs						
Non smoker)	-235.1714	27.74409	-8.48	0.000	-289.5488	-180.794

We specified the option `generate(matchv)` to speed up the postestimation commands that compute balance statistics, as discussed in [example 3](#) under *Remarks and examples* in [\[CAUSAL\] tebalance](#). We do not interpret the estimated effect produced by this preliminary model but rather check the specification.

We begin by looking at the standardized differences and variance ratios for the raw data and the matched sample.

```
. tebalance summarize
```

```
Covariate balance summary
```

	Raw	Matched
Number of obs =	4,642	9,284
Treated obs =	864	4,642
Control obs =	3,778	4,642

	Standardized differences		Variance ratio	
	Raw	Matched	Raw	Matched
mmarried	-.5953009	.0014107	1.335944	.9987659
mage	-.300179	-.0120277	.8818025	.9952916
prenatal1	-.3242695	.0333609	1.496155	.9491524
fbaby	-.1663271	-.0117326	.9430944	.9969095

The matched sample results indicate that matching on the estimated propensity score balanced the covariates. The standardized differences are all close to zero, and the variance ratios are all close to one. This inference is informal because we do not have standard errors for these statistics.

We may also wish to see the baseline summary statistics.

```
. tebalance summarize, baseline
```

```
Covariate balance summary
```

	Raw	Matched
Number of obs =	4,642	9,284
Treated obs =	864	4,642
Control obs =	3,778	4,642

	Means		Variances	
	Control	Treated	Control	Treated
mmarried	.7514558	.4733796	.1868194	.2495802
mage	26.81048	25.16667	31.87141	28.10429
prenatal1	.8268925	.6898148	.1431792	.2142183
fbaby	.4531498	.3715278	.2478707	.2337654

While we rely on the standardized differences for conclusions about balance in the unmatched sample from this output, the baseline means and variances give us some idea of the scale of the differences.

► Example 2: Multivalued treatments

In the multivalued-treatment case, `tebalance summarize` produces output grouped by treatment level. In the [Cattaneo \(2010\)](#) extract, the variable `msmoke` is an ordered categorical variable specifying the number of cigarettes smoked. We begin by tabulating `msmoke`.

```
. tabulate msmoke
```

Cigarettes smoked during pregnancy	Freq.	Percent	Cum.
0 daily	3,778	81.39	81.39
1-5 daily	200	4.31	85.70
6-10 daily	337	7.26	92.96
11+ daily	327	7.04	100.00
Total	4,642	100.00	

All the treatment groups have significantly smaller numbers of observations than the control group of not smoking. Still, each group has at least 200 observations. We continue by quietly fitting a candidate IPW model and reporting the baseline summaries.

```
. quietly teffects ipw (bweight) (msmoke mmarried mage prenatal1 fbaby)
. tebalance summarize, baseline
Covariate balance summary
```

Treatment		Observations	
		Raw	Weighted
0 daily	=	3,778	1,164.8
1-5 daily	=	200	1,164.4
6-10 daily	=	337	1,157.9
11+ daily	=	327	1,154.9
Total	=	4,642	4,642.0

	Means		Variances	
	Control	Treated	Control	Treated
1-5 daily				
mmarried	.7514558	.455	.1868194	.2492211
mage	26.81048	24.64	31.87141	31.44764
prenatal1	.8268925	.695	.1431792	.2130402
fbaby	.4531498	.48	.2478707	.2508543
6-10 daily				
mmarried	.7514558	.4480712	.1868194	.2480394
mage	26.81048	25.06231	31.87141	27.07051
prenatal1	.8268925	.6795252	.1431792	.2184188
fbaby	.4531498	.3827893	.2478707	.2369648
11+ daily				
mmarried	.7514558	.5107034	.1868194	.250652
mage	26.81048	25.59633	31.87141	26.93471
prenatal1	.8268925	.6972477	.1431792	.2117409
fbaby	.4531498	.293578	.2478707	.2080261

The results for the control level of 0 daily are repeated for the treatment group. These results give a sense of the scale of imbalance in the raw data. Now we compute the balance statistics.

```
. tebalance summarize
```

```
Covariate balance summary
```

Treatment	Observations	
	Raw	Weighted
0 daily =	3,778	1,164.8
1-5 daily =	200	1,164.4
6-10 daily =	337	1,157.9
11+ daily =	327	1,154.9
Total =	4,642	4,642.0

	Standardized differences		Variance ratio	
	Raw	Weighted	Raw	Weighted
1-5 daily				
mmarried	-.634909	-.0016208	1.334021	1.001406
mage	-.3857482	-.0219656	.9867038	.9905584
prenatal1	-.312519	-.0012611	1.487927	1.001898
fbaby	.053769	.0422102	1.012037	1.008631
6-10 daily				
mmarried	-.6506304	-.0108454	1.327696	1.009331
mage	-.3220222	-.0836571	.8493666	.7984901
prenatal1	-.3465797	-.0100232	1.525493	1.015051
fbaby	-.1429048	.0268118	.9560018	1.005899
11+ daily				
mmarried	-.5147672	-.0212969	1.34168	1.018136
mage	-.2239116	-.0636951	.8451058	.8468934
prenatal1	-.3077549	-.0380744	1.478852	1.056645
fbaby	-.3342243	.0155427	.8392526	1.003598

These results indicate that the IPW estimator probably did not fully balance the covariates (the variance ratios for mage at the daily levels of 6–10 cigarettes and 11-plus cigarettes are not close to 1). At this point, we would use a richer model and see whether it balanced the covariates.

Note that we cannot use `tebalance override`, because it has not been implemented for multivalued treatments.

◀

Stored results

`tebalance summarize` stores the following in `r()`:

Matrices

<code>r(size)</code>	number of observations in the raw and matched or weighted samples
<code>r(table)</code>	table of covariate statistics

Reference

Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.

Also see

- [CAUSAL] [stteffects intro](#) — Introduction to treatment effects for observational survival-time data
- [CAUSAL] [stteffects ipw](#) — Survival-time inverse-probability weighting
- [CAUSAL] [stteffects ipwra](#) — Survival-time inverse-probability-weighted regression adjustment
- [CAUSAL] [tebalance](#) — Check balance after teffects or stteffects estimation
- [CAUSAL] [teffects aipw](#) — Augmented inverse-probability weighting
- [CAUSAL] [teffects ipw](#) — Inverse-probability weighting
- [CAUSAL] [teffects ipwra](#) — Inverse-probability-weighted regression adjustment
- [CAUSAL] [teffects nnmatch](#) — Nearest-neighbor matching
- [CAUSAL] [teffects psmatch](#) — Propensity-score matching
- [CAUSAL] [teoverlap](#) — Overlap plots

Description

`teffects` estimates potential-outcome means (POMs), average treatment effects (ATEs), and average treatment effects on the treated (ATETs) using observational data. Regression-adjustment, inverse-probability-weighting, and matching estimators are provided, as are doubly robust methods that combine regression adjustment and inverse-probability weighting.

The outcomes can be continuous, binary, count, fractional, or nonnegative. The treatment model can be binary, or it can be multinomial, allowing for multivalued treatments.

For a brief description and example of each estimator, see *Remarks and examples* in [\[CAUSAL\] teffects intro](#).

Syntax

`teffects subcommand ... [ , options ]`

<i>subcommand</i>	Description
aipw	augmented inverse-probability weighting
ipw	inverse-probability weighting
ipwra	inverse-probability-weighted regression adjustment
nnmatch	nearest-neighbor matching
psmatch	propensity-score matching
ra	regression adjustment

Also see

- [\[CAUSAL\] teffects intro](#) — Introduction to treatment effects for observational data
- [\[CAUSAL\] teffects intro advanced](#) — Advanced introduction to treatment effects for observational data
- [\[CAUSAL\] teffects multivalued](#) — Multivalued treatment effects

Description

This entry provides a nontechnical introduction to treatment-effects estimators and the `teffects` command in Stata. Advanced users may want to instead read [\[CAUSAL\] teffects intro advanced](#) or skip to the individual commands' entries.

The `teffects` command estimates average treatment effects (ATEs), average treatment effects among treated subjects (ATETs), and potential-outcome means (POMs) using observational data.

Treatment effects can be estimated using regression adjustment (RA), inverse-probability weights (IPW), and “doubly robust” methods, including inverse-probability-weighted regression adjustment (IPWRA) and augmented inverse-probability weights (AIPW), and via matching on the propensity score or nearest neighbors.

The outcome can be continuous, binary, count, fractional, or nonnegative. Treatments can be binary or multivalued.

Remarks and examples

This entry presents a nontechnical overview of treatment-effects estimators for those who are new to the subject of treatment-effects estimation or are at least new to Stata's facilities for estimating treatment effects. More advanced users may want to instead read [\[CAUSAL\] teffects intro advanced](#) or skip to the individual commands' entries.

Remarks are presented under the following headings:

- [Introduction](#)
- [Defining treatment effects](#)
- [Estimating treatment effects](#)
 - [Regression adjustment](#)
 - [Inverse-probability weighting](#)
 - [Doubly robust combinations of RA and IPW](#)
 - [Matching](#)
- [Caveats and assumptions](#)
- [A quick tour of the estimators](#)
 - [RA](#)
 - [IPW](#)
 - [IPWRA](#)
 - [AIPW](#)
 - [Nearest-neighbor matching](#)
 - [Propensity-score matching](#)
- [Video examples](#)

Introduction

Suppose we have observed a sample of subjects, some of whom received a treatment and the rest of whom did not. As the name suggests, in most applications, the “subjects” are indeed people. A “treatment” could indeed be a medical treatment such as a new drug regimen or surgical procedure. In social science applications, a treatment could be participation in a job-training program or inclusion in a classroom or school in which a new pedagogical method is being used. However, not all applications use individuals as the subjects. For example, a policy analyst might be interested in examining the impact of an experimental program in which a national agency held a lottery to award only some local governments the resources needed to implement the program. Here the subjects are the local governments, and treatment refers to whether a local government received the resources needed to implement the program.

We would like to know if a treatment has an effect on an outcome Y . The outcome could be the cholesterol level of a patient taking either an existing statin or a new experimental drug, or the outcome could be the wage offered to a person who either did or did not participate in a job-training program. In an ideal world, we would observe Y when a subject is treated (which we denote as Y_1), and we would observe Y when the same subject is not treated (which we denote as Y_0). We would be careful to make both observations under identical conditions so that the only difference is the presence or absence of the treatment. We could then average the difference between Y_1 and Y_0 across all the subjects in our dataset to obtain a measure of the average impact of the treatment.

Unfortunately, this ideal experiment is almost never available in observational data because it is not possible to observe a specific subject having received the treatment and having not received the treatment. When the outcome is the birthweight of a specific baby and the treatment is the mother smoking while pregnant, it is impossible to observe the baby’s birthweight under both treatments of the mother smoking and the mother not smoking.

A classic solution to this problem is to randomize the treatment. High costs or ethical issues rule out this solution in many observational datasets. For example, we could not ask a random selection of pregnant women to smoke.

The defining characteristic of observational data is that treatment status is not randomized. Moreover, that implies that the outcome and treatment are not necessarily independent. The goal of the estimators implemented by `teffects` is to utilize covariates to make treatment and outcome independent once we condition on those covariates.

The treatment-effect estimators implemented by `teffects` allow us to estimate the efficacy of treatments using observational data. The rest of this entry discusses these treatment-effect estimators at an introductory level. For a more technical introduction, see [\[CAUSAL\] teffects intro advanced](#).

Defining treatment effects

We introduce treatment effects more formally by using the potential-outcomes framework, which is also known as the counterfactual framework. What is a potential outcome? Consider a subject that did not receive treatment so that we observe Y_0 . What would Y_1 be for that same subject if it were exposed to treatment? We call Y_1 the potential outcome or counterfactual for that subject. For a subject that did receive treatment, we observe Y_1 , so Y_0 would be the counterfactual outcome for that subject. We can view this as a missing-data problem, and treatment-effect methods can account for that problem.

Treatment-effect estimators allow us to estimate three parameters. The potential-outcome means (POMs) are the means of Y_1 and Y_0 in the population. The average treatment effect (ATE) is the mean of the difference ($Y_1 - Y_0$). Finally, the average treatment effect on the treated (ATET) is the mean of the difference ($Y_1 - Y_0$) among the subjects that actually receive the treatment.

To develop our intuition, suppose we have observed a sample of patients, some of whom received a medication to reduce their blood pressure. Figure 1 plots each of our patient’s systolic blood pressures as a function of weight. We use the color red to indicate patients who did not receive the drug and blue to indicate patients who did receive the drug.

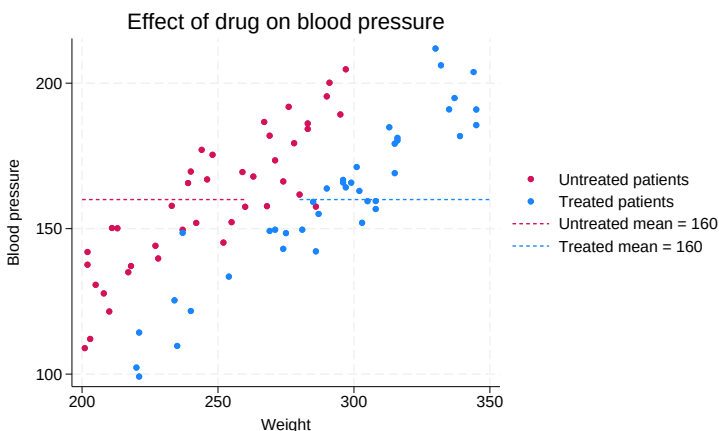


Figure 1.

A remarkable feature of our data is that the average blood pressure of patients not taking the drug is 160, and the average blood pressure of patients taking the drug is also 160. Can we therefore conclude that taking the drug has no impact on blood pressure? The answer is no.

Because this is observational data, we could not randomly assign who would receive the drug and who would not. As a result, treatment status could be related to covariates that also affect blood pressure. Heavier patients were more likely to be prescribed the medication, and blood pressure is correlated with weight. The difference in sample means does not estimate the true average treatment effect, because blood pressure depends on weight and weight is correlated with the treatment.

Suppose that we did in fact observe both potential outcomes for all patients. In figure 2, we continue to use solid dots for our observed data points, and we introduce hollow dots to represent the counterfactual outcomes. That is, the red hollow dots represent the blood pressures we would measure if only our treated patients had not taken the drug, and the blue hollow dots represent the blood pressures we would measure if only our untreated patients had taken the drug. The red and blue dashed lines represent the untreated and treated POMs, respectively. That is, the red line represents the mean of all the red dots, and the blue line represents the mean of all the blue dots.

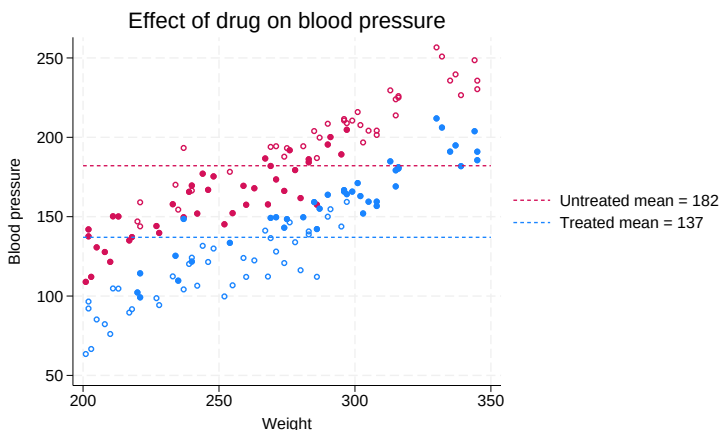


Figure 2.

If we did have the data represented by the hollow dots, then we could say that the average treatment effect is the difference between the mean of all the red dots and the mean of all the blue dots. In this ideal scenario, there are no missing data on the other potential outcome, and we have all the data we need to use the difference in means to estimate the ATE.

Looking at figure 2, we can see why a difference in means using only the solid dots does not estimate the ATE. Using only the solid red dots underestimates the average blood pressure for untreated individuals, and using only the solid blue dots overestimates the average blood pressure for treated individuals.

Estimating an ATE is essentially a missing-data problem. When covariates that affect the potential outcomes are related to treatment, we cannot use a difference in sample means, because the missing data are informative.

The treatment-effect estimators implemented in `teffects` allow for covariates like weight to be related to the potential outcomes and the treatment. Essentially, the estimators implemented by `teffects` utilize covariates to fill in the hollow circles or otherwise account for how the missing data depend on covariates that affect the potential outcomes.

Estimating treatment effects

We cannot estimate the ATE by simply taking the difference between the sample means for the treated and untreated subjects, because there are covariates that are related to the potential outcomes and the treatment. The estimators implemented by `teffects` require us to specify enough of these covariates so that after we condition on these covariates, any remaining influences on the treatment are not related to the potential outcomes. `teffects` implements several different estimators to accomplish this, including regression adjustment (RA), inverse-probability weighting (IPW), “doubly robust” methods that combine elements of RA and IPW, and matching methods. Here we introduce the methods by using intuition and simple examples.

See [\[CAUSAL\] teffects intro advanced](#) for a more technical introduction, and see the individual commands’ entries for estimator-specific details.

Regression adjustment

The RA method extends the idea of using sample means to estimate treatment effects by using a regression model to predict potential outcomes adjusted for covariates. In the examples here, we use linear regression, but the `teffects ra` command provides you with the flexibility to use logistic, probit, and heteroskedastic probit regression models for binary outcomes as well as Poisson regression for nonnegative outcomes; see [\[CAUSAL\] teffects ra](#) for more information.

`bweightex.dta` is a hypothetical dataset based on [Cattaneo \(2010\)](#) that we have created to illustrate treatment-effects estimators using graphs. The subjects in this dataset are women who were pregnant, some of whom smoked during the pregnancy. The outcome variable is the birthweight of the baby, and we want to know whether smoking during pregnancy affects the birthweight. The dataset also contains other demographic variables that we will use later.

Figure 3 illustrates the relationship between birthweight and smoking status as a function of the mother's age:

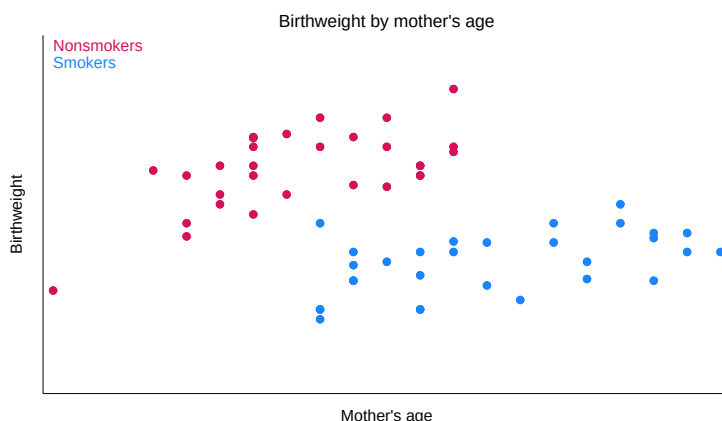


Figure 3.

We see that smokers tend to be older than nonsmokers and that birthweight depends on smoking. Therefore, the difference between the sample means of birthweights of babies born to smokers and nonsmokers will not estimate the true average treatment effect.

We also still have the same problem as in the previous section: we do not observe the counterfactual birthweights of babies. Suppose, however, that we did. In [figure 4](#), we use solid points to represent observed birthweights and the colors red to represent nonsmokers and blue to represent smokers. The hollow points represent the counterfactual birthweights. The hollow blue points represent the birthweights of babies that we would observe if only our young nonsmoking mothers had instead smoked during their pregnancies. Similarly, the hollow red points represent the birthweights of babies that we would observe if only our older smoking mothers had instead not smoked during their pregnancies.

Figure 4 suggests a way to estimate the potential outcomes for each mother:

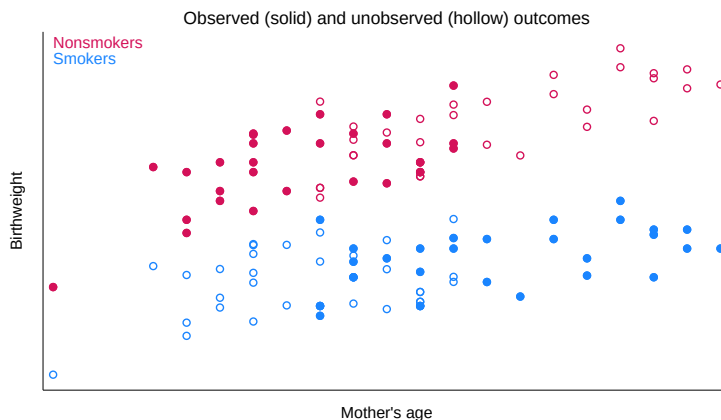


Figure 4.

We could fit a linear regression of birthweight on mother's age by using the observed birthweights for nonsmokers, and we could do likewise for smokers. The following graph includes these two regression lines:

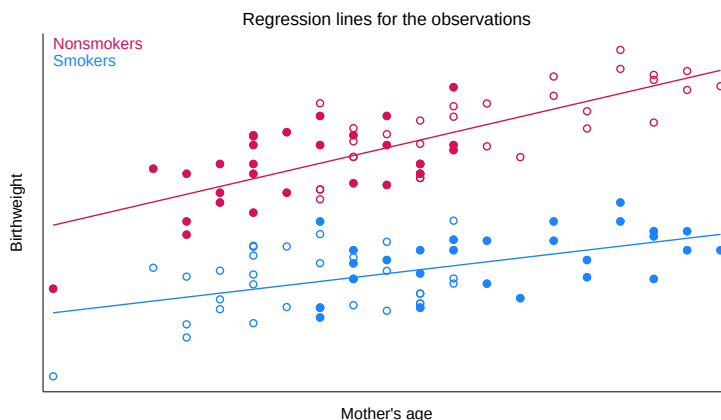


Figure 5.

Figure 5 illustrates the principle behind the RA method. We use the red regression line to predict each baby's birthweight assuming the mother did not smoke, and we use the blue regression line to predict each baby's birthweight assuming the mother did smoke. The treatment effect of smoking for a mother of a particular age is the vertical difference between the red and blue regression lines.

The three parameters we mentioned in the introduction are now easy to estimate. For each mother, we obtain two values, say, bw_0 and bw_1 , representing our predictions of her baby's birthweight assuming the mother did not or did smoke, respectively. The means of these variables represent the untreated and treated POMs. The ATE is the sample mean of the difference ($bw_1 - bw_0$), and the ATET is the sample mean of that difference computed using only the mothers who in fact did smoke during pregnancy.

Adding the circles highlights the fact that the average age is higher for smokers than for nonsmokers. Even though the blue and red lines have different slopes, if the average age was the same for smokers and nonsmokers, a difference in the sample means of birthweights could still estimate the true ATE.

Figure 5 lets us address one more issue. Users who are versed in regression analysis may be inclined to estimate the effect of smoking using a regression model for birthweight as a function of smoking and the mother's age. We clearly see in figure 5 that regression lines for smokers and nonsmokers have different slopes—the effect of age on birthweight is not the same for smokers and nonsmokers. In regression analysis, we would therefore include an interaction term between smoking and age. The RA method fits separate regression lines for smokers and nonsmokers, which also handles these differential effects of age on smoking.

Inverse-probability weighting

As we remarked in our discussion of the RA method, we cannot simply use the sample mean birthweights of babies born to smokers and nonsmokers to estimate the effect of smoking. If we did that, we would conflate the negative effect of smoking with the positive effect of age and the positive relationship between age and smoking. IPW is a treatment-effects estimator that uses weighted means rather than simple unweighted means to disentangle the effects of treatment and other confounders like age.

The concept underlying IPW can be gleaned from figure 2, where, as you will recall, the hollow points represent counterfactual outcomes. As we demonstrated in *Defining treatment effects*, we could estimate the average treatment effect if we knew the means of all the nonsmoking outcomes and the means of all the smoking outcomes. In the context of figure 4, we need the mean of all the red points, both solid and hollow, and the mean of all the blue.

If we could observe all of these points, then the ATE would be the difference between those two means. However, the outcomes illustrated by the hollow circles are unobserved. IPW estimators view the hollow circles as missing data and use weights to correct the estimates of the treated and untreated sample means for the missing data. If we calculate the mean nonsmoking birthweight using just the solid red points, that mean is biased downward because we are ignoring the hollow red points, which correspond to higher birthweights.

In IPW, we apply more weight to the solid red points corresponding to older mothers and less weight to those corresponding to younger mothers. Using this weighting scheme will pull up the estimated mean birthweight of babies born to nonsmoking mothers to estimate the true mean of all nonsmoking outcomes. The method for obtaining the mean smoking birthweight is virtually the same: we need to apply more weight to the younger smoking mothers than to the older smoking mothers to better approximate the true mean of all smoking outcomes.

Where do these weights for the weighted means come from? As the name implies, IPW uses the inverse (reciprocal) of the probability of being in the observed treatment group. These probabilities are obtained by modeling the observed treatment as a function of subject characteristics that determine treatment group. In our exposition of the RA method, we focused solely on the mother's age and smoking status as determinants of each baby's birthweight. To make the results comparable, we will use the same model in this example.

We first fit a logistic model of the mother's smoking status, `mbsmoke`, as a function of the mother's age (`mage`):

```
. use https://www.stata-press.com/data/r19/bweightex
(Hypothetical birthweight data)
. logistic mbsmoke mage
```

Logistic regression

	Number of obs =	60
	LR chi2(1) =	30.45
	Prob > chi2 =	0.0000
	Pseudo R2 =	0.3661

Log likelihood = -26.362201

mbsmoke	Odds ratio	Std. err.	z	P> z	[95% conf. interval]
mage	1.631606	.21316	3.75	0.000	1.263022 2.107754
_cons	7.76e-06	.0000243	-3.76	0.000	1.69e-08 .0035718

Note: `_cons` estimates baseline odds.

Next, we compute the inverse-probability weights, which we will store in a variable called `ps`. In the IPW method, for subjects who did receive treatment, the weight is equal to the reciprocal of the predicted probability of treatment. For subjects who did not receive treatment, the weight is equal to the reciprocal of the predicted probability of not receiving treatment; the probability of not receiving treatment is just one minus the probability of receiving treatment:

```
. predict ps
(option pr assumed; Pr(mbsmoke))
. replace ps = 1/ps if mbsmoke==1
(30 real changes made)
. replace ps = 1/(1-ps) if mbsmoke==0
(30 real changes made)
```

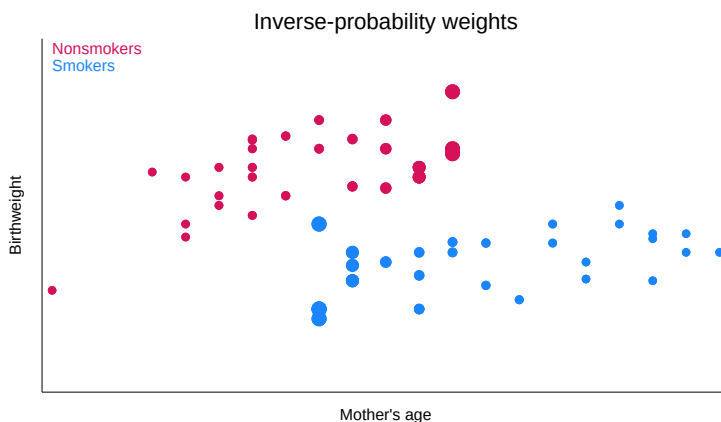


Figure 6.

Figure 6 replicates [figure 3](#) with one twist. Rather than making all the points the same size, we have made the size of the points proportional to the IPW variable `ps`. Notice that the largest blue points correspond to the youngest smoking mothers in our sample, so they will receive the most weight when we compute the weighted mean birthweight of babies born to smoking mothers, just as we explained we wanted to do. Similarly, the red points corresponding to older nonsmoking mothers are larger, representing larger weights.

There is a caveat to the IPW estimator. When we fit our logistic or probit model to obtain the predicted probabilities, it is possible that some of the predictions will be close to zero. Because the IPW is the reciprocal of that probability, the weight becomes arbitrarily large as the probability goes to zero. In those cases, the IPW can become unstable. We can improve the estimated IPW by developing a more accurate treatment model. For example, in our dataset, we have other variables such as marital status and the education level of the baby's father that may also help predict whether the mother smoked during pregnancy. We excluded these variables for simplicity, but in a real analysis, we would want to use all relevant data.

This phenomenon of unstable IPWs is related to the concept of overlap, which means that every subject must have a strictly positive probability of obtaining treatment. We remarked that in our sample, we had few young mothers who smoked. As should be clear from [figure 6](#), the overlap assumption is likely to be violated—young mothers do not appear to have a positive probability of being smokers. We would want to check this assumption before proceeding with an IPW analysis. See [\[CAUSAL\] teoverlap](#) and [\[CAUSAL\] teffects intro advanced](#) for more information about overlap.

Another limitation of the IPW estimator is that we are using weighted means to estimate the POMs and ATE. Thus, unlike the RA estimator, we cannot obtain subject-level predictions of the treatment effects or potential outcomes, because we do not have the two regression lines that we can use to predict outcomes for each subject.

Doubly robust combinations of RA and IPW

You may have noticed a clear distinction between the RA and IPW estimators. In the case of RA, we built linear regression models to predict the outcomes (birthweights) of each subject but said nothing about how treatment (smoking) arises. In the case of IPW, we built a logistic regression model to predict treatment status but did not build a formal model of the outcome. Doubly robust estimators combine the outcome modeling strategy of RA and the treatment modeling strategy of IPW. These estimators have a remarkable property: although they require us to build two models, we only need to specify one of the two models correctly. If we misspecify the treatment model but correctly specify the outcome model, we still obtain correct estimates of the treatment effect. If we correctly specify the treatment model but misspecify the outcome model, we again will obtain correct estimates of the treatment effect.

Stata's `teffects` command implements two doubly robust estimators, the augmented inverse-probability-weighted (AIPW) estimator and the inverse-probability-weighted regression-adjustment (IPWRA) estimator. These estimators combine elements of RA and IPW to be more robust to misspecification.

The AIPW estimator is an IPW estimator that includes an augmentation term that corrects the estimator when the treatment model is misspecified. When the treatment model is correctly specified, the augmentation term vanishes as the sample size becomes large. Like the IPW, the AIPW does not perform well when the predicted treatment probabilities are too close to zero or one.

The IPWRA estimator is an RA estimator that uses estimated inverse-probability weights to correct the estimator when the regression function is misspecified. When the regression function is correctly specified, the weights do not affect the consistency of the estimator.

Matching

Matching estimators are based on the idea of comparing the outcomes of subjects that are as similar as possible with the sole exception of their treatment status. In our birthweight and smoking example, we could select a mother who smokes and select a mother of the same age who does not smoke and compare the birthweights of their infants. The data of each mother serve as the potential outcome for the other mother.

For a single covariate such as age, identifying a pair of comparable mothers is not difficult. If we have a second covariate that is categorical, such as race, we might still be able to identify pairs of mothers who are the same age and of the same race assuming our dataset is large enough. However, once we consider covariates that are measured on continuous scales or allow for more than a few discrete ones, then finding identical matches is a challenge. The solution is to use what is called a similarity measure, which is a statistic that measures how “close” two observations are. `teffects` offers two methods to find comparable observations based on similarity measures: nearest-neighbor matching and propensity-score matching.

Nearest-neighbor matching (NNM) is accomplished by calculating the “distance” between pairs of observations with regard to a set of covariates and then “matching” each subject to comparable observations that are closest to it. For example, suppose we have a variable that records each subject’s annual income to the penny. Say one subject who received treatment had an income of \$69,234.21. The likelihood that our dataset has an untreated subject who also earned \$69,234.21 is nil. However, we can determine the difference between each untreated subject’s income and our treated subject’s income, then match our treated subject with the untreated subjects whose income differences are smallest. Measuring the distance between subjects when we have multiple covariates is no challenge. By default, `teffects` uses what is known as the Mahalanobis distance, which is really nothing more than the Pythagorean theorem adapted to handle the fact that covariates may be correlated and measured on different scales.

NNM does not use a formal model for either the outcome or the treatment status, but this flexibility comes at a price. When matching on more than one continuous covariate, the NNM estimator must be augmented with a bias-correction term. `teffects nmatch` uses a linear function of the covariates specified in the `biasadj()` option to remove the large-sample bias.

Propensity-score matching (PSM) is an alternative to NNM. PSM matches on the estimated predicted probabilities of treatment, known as the propensity scores. PSM does not require bias correction, because it uses a model for the treatment. If the treatment model is reasonably well specified, PSM will perform at least as well as NNM; see [\[CAUSAL\] teffects intro advanced](#).

Caveats and assumptions

To use the estimators implemented in `teffects`, we must make several assumptions about the process that generated our data. Different estimators and statistics may require slightly more or slightly less restrictive assumptions and may exhibit varying degrees of robustness to departures from these assumptions, but in general, all the estimators require some form of the following three assumptions.

The independent and identically distributed (i.i.d.) sampling assumption ensures that the outcome and treatment status of each individual are unrelated to the outcome and treatment status of all the other individuals in the population. Correlated data arising from hierarchical or longitudinal study designs do not meet this assumption.

The conditional-independence (CI) assumption means once we control for all observable variables, the potential outcomes are independent of treatment assignment. The easiest way to understand the CI assumption is to understand when it is violated. In our birthweight example, suppose mothers who did not smoke were more health conscious and consumed better prenatal diets than those who did smoke. Unless we explicitly controlled for health awareness or diet, our model would violate the CI assumption: the mother's decision to smoke or not smoke would not be independent of the baby's birthweight. If we did not control for health awareness, we would overstate the negative impact of smoking on birthweight. Babies born to mothers who smoke weigh less than babies born to nonsmoking mothers not just because of the effects of cigarettes but also because of poorer prenatal diets.

In a study examining the effect of a job-training program, the CI assumption requires that there not be any unobserved factors such as ambition or work ethic that influence both whether a person enrolls in the program and the wage received upon completion. To use the methods implemented by the `teffects` estimators, we must have variables in our dataset that allow us to control for those types of factors.

We mentioned the third assumption, overlap, in our discussions of IPW. More formally, the overlap assumption states that each individual have a positive probability of receiving treatment. In our birthweight example, we noted that there were no observations on young smokers and older nonsmokers. Perhaps we just have an unlucky sample, but to accurately assess the impact of treatment using these methods, we must have overlap to accurately estimate the counterfactual birthweights. In the context of matching estimators, overlap essentially means that we can actually match treated subjects with similar nontreated subjects.

A quick tour of the estimators

The `teffects` command implements six estimators of treatment effects. We introduce each one by showing the basic syntax one would use to apply them to our birthweight example. See each command's entry for more information.

RA

`teffects ra` implements the RA estimator. We estimate the effect of a mother's smoking behavior (`mbsmoke`) on the birthweight of her child (`bweight`), controlling for marital status (`mmarried`), the mother's age (`mage`), whether the mother had a prenatal doctor's visit in the baby's first trimester (`prenatal1`), and whether this baby is the mother's first child (`fbaby`). We use linear regression (the default) to model `bweight`:

```
. use https://www.stata-press.com/data/r19/cattaneo2
. teffects ra (bweight mmarried mage prenatal1 fbaby) (mbsmoke)
```

IPW

`teffects ipw` implements the IPW estimator. Here we estimate the effect of smoking by using a probit model to predict the mother's smoking behavior as a function of marital status, the mother's age, and indicators for first-trimester doctor's visits and firstborn status:

```
. teffects ipw (bweight) (mbsmoke mmarried mage prenatal1 fbaby, probit)
```

IPWRA

`teffects ipwra` implements the IPWRA estimator. We model the outcome, birthweight, as a linear function of marital status, the mother's age, and indicators for first-trimester doctor's visits and firstborn status. We use a logistic model (the default) to predict the mother's smoking behavior, using the same covariates as explanatory variables:

```
. teffects ipwra (bweight mmarried mage prenatal1 fbaby) ///
               (mbsmoke mmarried mage prenatal1 fbaby)
```

AIPW

`teffects aipw` implements the AIPW estimator. Here we use the same outcome- and treatment-model specifications as we did with the IPWRA estimator:

```
. teffects aipw (bweight mmarried mage prenatal1 fbaby) ///
               (mbsmoke mmarried mage prenatal1 fbaby)
```

Nearest-neighbor matching

`teffects nnmatch` implements the NNM estimator. In this example, we match treated and untreated subjects based on marital status, the mother's age, the father's age, and indicators for first-trimester doctor's visits and firstborn status. We use the Mahalanobis distance based on the mother's and father's ages to find matches. We use exact matching on the other three variables to enforce the requirement that treated subjects are matched with untreated subjects who have the same marital status and indicators for first-trimester doctor's visits and firstborn statuses. Because we are matching on two continuous covariates, we request that `teffects nnmatch` include a bias-correction term based on those two covariates:

```
. teffects nnmatch (bweight mage fage) (mbsmoke),          ///
                  ematch(prenatal1 mmarried fbaby) biasadj(mage fage)
```

Propensity-score matching

`teffects psmatch` implements the PSM estimator. Here we model the propensity score using a probit model, incorporating marital status, the mother's age, and indicators for first-trimester doctor's visits and firstborn status as covariates:

```
. teffects psmatch (bweight) (mbsmoke mmarried mage prenatal1 fbaby, probit)
```

Video examples

[Introduction to treatment effects in Stata, part 1](#)

[Introduction to treatment effects in Stata, part 2](#)

References

- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., R. Titiunik, and G. Vazquez-Bare. 2020. Analysis of regression-discontinuity designs with multiple cutoffs or multiple scores. *Stata Journal* 20: 866–891.
- Cerulli, G. 2017. Identification and estimation of treatment effects in the presence of (correlated) neighborhood interactions: Model and Stata implementation via `ntreatreg`. *Stata Journal* 17: 803–833.
- . 2020. Nonparametric synthetic control using the `npsynth` command. *Stata Journal* 20: 844–865.
- Cerulli, G., and M. Ventura. 2019. Estimation of pre- and posttreatment average treatment effects with binary time-varying treatment using Stata. *Stata Journal* 19: 551–565.
- Drukker, D. M. 2016. A generalized regression-adjustment estimator for average treatment effects from panel data. *Stata Journal* 16: 826–836.
- Galiani, S., and B. Quistorff. 2017. The `synth_runner` package: Utilities to automate synthetic control estimation using `synth`. *Stata Journal* 17: 834–849.
- Heß, S. 2017. Randomization inference with Stata: A guide and software. *Stata Journal* 17: 630–651.
- Linden, A., C. Huber, and G. T. Wodtke. 2021. A regression-with-residuals method for analyzing causal mediation: The `rwrmmed` package. *Stata Journal* 21: 559–574.
- Palomba, F. 2024. Getting away from the cutoff in regression discontinuity designs. *Stata Journal* 24: 371–401.
- Pierce, M., and R. Emsley. 2021. Estimating and evaluating personalized treatment recommendations from randomized trials with `ptr`. *Stata Journal* 21: 348–359.
- Tübbicke, S. 2023. `ebct`: Using entropy balancing for continuous treatments to estimate dose–response functions and their derivatives. *Stata Journal* 23: 709–729.

Also see

[CAUSAL] [teffects](#) — Treatment-effects estimation for observational data

[CAUSAL] [teffects intro advanced](#) — Advanced introduction to treatment effects for observational data

[CAUSAL] [teffects multivalued](#) — Multivalued treatment effects

[Description](#)
[Also see](#)[Remarks and examples](#)[Acknowledgments](#)[References](#)

Description

This entry provides a technical overview of treatment-effects estimators and their implementation in Stata. Those who are new to treatment-effects estimation may want to instead see [\[CAUSAL\] teffects intro](#).

The `teffects` command estimates average treatment effects (ATEs), average treatment effects among treated subjects (ATETs), and potential-outcome means (POMs) using observational data.

Treatment effects can be estimated using regression adjustment (RA), inverse-probability weights (IPW), and “doubly robust” methods, including inverse-probability-weighted regression adjustment (IPWRA) and augmented inverse-probability weights (AIPW), and via matching on the propensity score or nearest neighbors.

The outcome can be continuous, binary, count, fractional, or nonnegative. Treatments can be binary or multivalued.

Remarks and examples

This entry presents a technical overview of treatment-effects estimators and their implementation in Stata. Users who are new to treatment-effects estimators for observational data should instead read [\[CAUSAL\] teffects intro](#).

Remarks are presented under the following headings:

- [Introduction](#)
- [Defining treatment effects](#)
- [The potential-outcome model](#)
- [Assumptions needed for estimation](#)
 - [The CI assumption](#)
 - [The overlap assumption](#)
 - [The i.i.d. assumption](#)
- [Comparing the ATE and ATET](#)
- [Overview of treatment-effect estimators](#)
- [RA estimators](#)
- [IPW estimators](#)
- [AIPW estimators](#)
- [IPWRA estimators](#)
- [Nearest-neighbor matching estimators](#)
- [Propensity-score matching estimators](#)
- [Choosing among estimators](#)
- [Model choice](#)

Introduction

The `teffects` commands estimate treatment effects from observed data. A treatment effect is the change in an outcome caused by a subject, often an individual, getting one treatment instead of another. We cannot estimate individual-level treatment effects, because we only observe each individual getting one or another treatment.

Potential-outcome models provide a solution to this missing-data problem and allow us to estimate the distribution of individual-level treatment effects. A potential-outcome model specifies the potential outcomes that each individual would obtain under each treatment level, the treatment assignment process, and the dependence of the potential outcomes on the treatment assignment process.

When the potential outcomes do not depend on the treatment levels, after conditioning on covariates, regression estimators, inverse-probability-weighted estimators, and matching estimators are commonly used.

What we call the potential-outcome model is also known as the Rubin causal model and the counterfactual model. See [Rubin \(1974\)](#); [Holland \(1986\)](#); [Robins \(1986\)](#); [Heckman \(1997\)](#); [Heckman and Navarro-Lozano \(2004\)](#); [Imbens \(2004\)](#); [Cameron and Trivedi \(2005, chap. 2.7\)](#); [Imbens and Wooldridge \(2009\)](#); and [Wooldridge \(2010, chap. 21\)](#) for more detailed discussions.

[Cameron and Trivedi \(2022\)](#) provide an introduction to treatment-effects estimation and to commands within Stata. For a discussion related to health econometrics, see [Deb, Norton, and Manning \(2017\)](#).

Defining treatment effects

Three parameters are often used to measure treatment effects: the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs). In this section, we define each of these terms and introduce the notation and parameters used in the rest of our discussion.

In the binary-treatment case, the two potential outcomes for each individual are y_{0i} and y_{1i} ; y_{0i} is the outcome that would be obtained if i does not get the treatment, and y_{1i} is the outcome that would be obtained if i gets the treatment. y_{0i} and y_{1i} are realizations of the random variables y_0 and y_1 . Throughout this entry, i subscripts denote realizations of the corresponding unsubscripted random variables. We do not discuss multivalued treatments here, because doing so only increases the number of parameters and notation required and detracts from the essential points; see [\[CAUSAL\] teffects multivalued](#) for information about multivalued treatments.

The parameters of interest summarize the distribution of the unobservable individual-level treatment effect $y_1 - y_0$. In defining the parameters, t denotes a random treatment, t_i denotes the treatment received by individual i , $t = 1$ is the treatment level, and $t = 0$ is the control level. Given this notation, we can now define our parameters of interest.

ATE The ATE is the average effect of the treatment in the population:

$$\text{ATE} = E(y_1 - y_0)$$

POM The POM for treatment level t is the average potential outcome for that treatment level:

$$\text{POM}_t = E(y_t)$$

ATET The ATET is the average treatment effect among those that receive the treatment:

$$\text{ATET} = E(y_1 - y_0 | t = 1)$$

For an illustration of these concepts, see [Defining treatment effects](#) in [\[CAUSAL\] teffects intro](#).

The potential-outcome model

Next we specify a potential-outcome model that serves as a touchstone for the rest of our discussion. The model described here generates data in which y_i is the observed outcome variable, t_i is the treatment variable, $\mathbf{x}_i$ is a vector of covariates that affect the outcome, and $\mathbf{w}_i$ is a vector of covariates that affect the treatment assignment. $\mathbf{x}_i$ and $\mathbf{w}_i$ may have elements in common.

This potential-outcome model specifies that the observed outcome variable y is y_0 when $t = 0$ and that y is y_1 when $t = 1$. Algebraically, we say that

$$y = (1 - t)y_0 + ty_1$$

The functional forms for y_0 and y_1 are

$$y_0 = \mathbf{x}'\beta_0 + \epsilon_0 \quad (1)$$

$$y_1 = \mathbf{x}'\beta_1 + \epsilon_1 \quad (2)$$

where β_0 and β_1 are coefficients to be estimated, and ϵ_0 and ϵ_1 are error terms that are not related to $\mathbf{x}$ or $\mathbf{w}$. This potential-outcome model separates each potential outcome into a predictable component, $\mathbf{x}\beta_t$, and an unobservable error term, ϵ_t .

The treatment assignment process is

$$t = \begin{cases} 1 & \text{if } \mathbf{w}'\gamma + \eta > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where γ is a coefficient vector, and η is an unobservable error term that is not related to either $\mathbf{x}$ or $\mathbf{w}$. The treatment assignment process is also separated into a predictable component, $\mathbf{w}'\gamma$, and an unobservable error term, η .

We emphasize six points about this model:

1. The observed data from this model contain y_i , t_i , $\mathbf{w}_i$, and $\mathbf{x}_i$. The data do not reveal both y_{0i} and y_{1i} for any given i .
2. The model for t determines how the data on y_0 and y_1 are missing.
3. The model separates the potential outcomes and treatment assignment into observable and unobservable components.
4. Whether η is independent of the vector (ϵ_0, ϵ_1) is essential in specifying the set of available estimators.
5. The coefficient vectors β_0 , β_1 , and γ are auxiliary parameters. We use estimates of these coefficient vectors to estimate the ATE, the POMs, and the ATET.
6. For notational simplicity, we represented y_0 and y_1 as linear functions. In practice, we can use other functional forms.

In specifying this potential-outcome model, we explicitly showed the functional forms for the potential outcomes and the treatment assignment process. To ease subsequent discussions, we refer to the set of functional forms for the potential outcomes as the “outcome model”, and we refer to the treatment assignment process as the “treatment model”.

Assumptions needed for estimation

As with any type of estimator, we must make some assumptions to use treatment-effects estimators. The particular assumptions we need for each estimator implemented by `teffects` and for each effect parameter vary, but some version of each of the following is required.

CI The conditional-independence CI assumption restricts the dependence between the treatment model and the potential outcomes.

Overlap The overlap assumption ensures that each individual could receive any treatment level.

i.i.d. The independent and identically distributed (i.i.d.) sampling assumption ensures that the potential outcomes and the treatment status of each individual are unrelated to the potential outcomes and treatment statuses of all other individuals in the population.

We now discuss each assumption in detail.

The CI assumption

After conditioning on covariates, when no unobservable variable affects both treatment assignment and the potential outcomes, the potential outcomes are conditionally independent of the treatment. In epidemiological jargon, there are no unmeasured confounders. In econometric jargon, we have selection on observables. If we observe enough covariates, the potential outcomes may indeed be conditionally independent of the treatment.

Intuitively, the CI assumption says that only the covariates $\mathbf{x}$ affect both the treatment and the potential outcomes. Any other factors that affect the treatment must be independent of the potential outcomes, and any other factors that affect the potential outcomes must be independent of the treatment. Formally, the CI assumption states that, conditional on covariates $\mathbf{x}$, the treatment t is independent of the vector of potential outcomes $(y_0, y_1)'$.

The CI assumption allows us to estimate the effects by regression-adjustment (RA) methods, inverse-probability-weighting (IPW) methods, methods that combine RA and IPW concepts, and matching methods. The data only reveal information about $E(y_0|\mathbf{x}, \mathbf{w}, t = 0)$ and $E(y_1|\mathbf{x}, \mathbf{w}, t = 1)$, but we are interested in an average of $E(y_0|\mathbf{x}, \mathbf{w})$ and $E(y_1|\mathbf{x}, \mathbf{w})$, where $\mathbf{x}$ represents the outcome covariates and $\mathbf{w}$ the treatment-assignment covariates. The CI assumption allows us to estimate $E(y_0|\mathbf{x}, \mathbf{w})$ and $E(y_1|\mathbf{x}, \mathbf{w})$ directly from the observations for which $E(y_0|\mathbf{x}, \mathbf{w}, t = 0)$ and $E(y_1|\mathbf{x}, \mathbf{w}, t = 1)$, respectively.

For our potential-outcome model presented in (1) through (3), the CI assumption can be viewed as a set of restrictions on the covariance matrix of the error terms. Suppose that the vector of unobservables $(\epsilon_0, \epsilon_1, \eta)$ is normally distributed

$$\begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \eta \end{pmatrix} \sim N \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_0^2 & \rho_{01}\sigma_0\sigma_1 & \rho_{\eta 0}\sigma_0 \\ \rho_{01}\sigma_0\sigma_1 & \sigma_1^2 & \rho_{\eta 1}\sigma_1 \\ \rho_{\eta 0}\sigma_0 & \rho_{\eta 1}\sigma_1 & 1 \end{pmatrix} \right\} \quad (4)$$

where σ_0 is the standard deviation of ϵ_0 , ρ_{01} is the correlation between ϵ_0 and ϵ_1 , σ_1 is the standard deviation of ϵ_1 , $\rho_{\eta 0}$ is the correlation between η and ϵ_0 , and $\rho_{\eta 1}$ is the correlation between η and ϵ_1 . As is standard in the normally distributed latent-variable specification of a binary-dependent variable, we normalize the variance of η to 1.

CI specifies that $\rho_{\eta 0} = \rho_{\eta 1} = 0$ so that we can write (4) as

$$\begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \eta \end{pmatrix} \sim N \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_0^2 & \rho_{01}\sigma_0\sigma_1 & 0 \\ \rho_{01}\sigma_0\sigma_1 & \sigma_1^2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right\}$$

Writing the covariance matrix this way makes clear what we mean by conditional independence: unobserved shocks that affect whether a subject is treated do not have any effect on the potential outcomes, and unobserved shocks that affect a potential outcome do not affect treatment.

`teffects` implements estimators that require the CI assumption. See [\[CAUSAL\] etregress](#) and [\[CAUSAL\] etpoisson](#) for commands that handle two cases in which the CI assumption is replaced by precise specifications of the joint dependence among the unobservables. [Brown and Mergoupis \(2011\)](#) discuss the `itreatreg` command that extends [\[CAUSAL\] etregress](#).

The CI assumption is also known as unconfoundedness and selection-on-observables in the literature. See [Rosenbaum and Rubin \(1983\)](#); [Heckman \(1997\)](#); [Heckman and Navarro-Lozano \(2004\)](#); [Cameron and Trivedi \(2005, sec. 25.2.1\)](#); [Tsiatis \(2006, sec. 13.3\)](#); [Angrist and Pischke \(2009, chap. 3\)](#); [Imbens and Wooldridge \(2009\)](#); and [Wooldridge \(2010, sec. 21.3\)](#). Some discussions with Stata commands can be found in [Becker and Caliendo \(2007\)](#), [Nichols \(2007\)](#), and [Daniel, De Stavola, and Cousens \(2011\)](#).

□ Technical note

In fact, full CI is stronger than what we need to estimate the ATE, the ATET, or the POMs. For the estimators implemented in `teffects`, we only need a conditional mean independence (CMI) assumption. Intuitively, the CMI assumption says that after accounting for the covariates $\mathbf{x}_i$, the treatment does not affect the conditional mean of each potential outcome. Formally, the CMI requires that $E(y_0|\mathbf{x}, t) = E(y_0|\mathbf{x})$ and that $E(y_1|\mathbf{x}, t) = E(y_1|\mathbf{x})$. The CMI assumption allows the conditional variance to depend on the treatment, while the CI assumption does not.

The CI assumption implies the CMI assumption, but not vice versa.

See [Wooldridge \(2010, sec. 21.2 and 21.3\)](#) for an excellent introduction to this topic, and see [Cattaneo, Drukker, and Holland \(2013\)](#) for some discussion of the multiple treatment case.

□

The overlap assumption

The overlap assumption requires that each individual have a positive probability of receiving each treatment level. Formally, the overlap assumption requires that for each possible $\mathbf{x}$ in the population and each treatment level $\tilde{t}$, $0 < \Pr(t = \tilde{t}|\mathbf{x}) < 1$. [Rosenbaum and Rubin \(1983\)](#) call the combination of the CI and overlap assumptions strong ignorability; see also [Abadie and Imbens \(2006, 237–238\)](#) and [Imbens and Wooldridge \(2009\)](#).

The i.i.d. assumption

The third of the three assumptions listed above is the i.i.d. assumption; it is the standard assumption of having an i.i.d. sample from the population. In potential-outcome models, i.i.d. sampling implies that the potential outcomes and treatment status of each individual are unrelated to the potential outcomes and treatment statuses of all the other individuals in the population. I.i.d. sampling rules out interactions among the individuals. For instance, models of vaccinations in epidemiology and general equilibrium effects in economics violate the independence assumption. This third assumption is a part of what is known as the stable unit treatment value assumption (SUTVA); see [Wooldridge \(2010, 905\)](#) and [Imbens and Wooldridge \(2009\)](#).

Comparing the ATE and ATET

When comparing the ATE and the ATET, two points should be mentioned.

First, the assumptions required to estimate the ATET are less restrictive than the assumptions required to estimate the ATE. Estimating the ATET requires a weaker form of the CI assumption and a weaker version of the overlap assumption.

To estimate the ATE under CI, we require that the unobservables in the treatment model be conditionally independent of the unobservables in both potential outcomes. In contrast, we can estimate the ATET under CI when the unobservables in the treatment model are conditionally independent of just the control-level potential outcome; see [Wooldridge \(2010, 906–912\)](#).

Although the ATE version of overlap requires that all covariate patterns have a positive probability of being allocated to each treatment state, we can estimate the ATET when only the covariate patterns for which someone is treated have a positive probability of being allocated to each treatment state. This weaker overlap assumption can be important in some studies. For example, [Heckman \(1997\)](#) discusses how the ATET makes sense in job-training programs for which many types of individuals have zero chance of signing up. See also [Wooldridge \(2010, 911–913\)](#).

Second, the ATET reduces to the ATE when the mean of the covariates among the treated is the same as the mean of the covariates in the population and when the average contribution from the unobservables for the participants is zero.

Overview of treatment-effect estimators

We can classify the estimators implemented by `teffects` into five categories: 1) estimators based on a model for the outcome variable; 2) estimators based on a model for treatment assignment; 3) estimators based on models for both the outcome variable and the treatment assignment; 4) estimators that match on covariates; and 5) estimators that match on predicted probabilities of treatment. Within each category of estimator, there is a variety of choices about the functional forms for the models.

Because there are several categories of estimators, the user must decide whether to model the outcomes, the probability of treatment, or both. Under correct model specification, using an outcome model and a model for the probability of treatment will produce more efficient estimates. Surprisingly, some of the estimators that use both models only require that one of the two be correctly specified to consistently estimate the effects of interest, a property known as the double-robust property.

With the exception of using a matching estimator with a single continuous covariate, some choice of functional forms is required. There are two aspects one must consider when choosing the functional form for the outcome or treatment assignment. First, one must select the functional form for the conditional mean or conditional probability; depending on the variable being modeled, a linear, a binary choice, or an exponential model may be appropriate. Second, one must determine the appropriate polynomials of the covariates to include in the model. `teffects` offers a wide variety of options to specify different functional form choices for the conditional mean and conditional probability models. The factor-variable notation in Stata allows us to easily specify the desired polynomial in the covariates.

We now provide some intuition behind each type of estimator.

RA estimators

RA estimators use means of predicted outcomes for each treatment level to estimate each POM. ATEs and ATETs are differences in estimated POMs.

The CI assumption implies that we can estimate $E(y_0|\mathbf{x})$ and $E(y_1|\mathbf{x})$ directly from the observations for which $t = 0$ and $t = 1$, respectively. Regression adjustment fits separate regressions for each treatment level and uses averages of the predicted outcomes over all the data to estimate the POMs. The estimated ATEs are differences in the estimated POMs. The estimated ATETs are averages of the predicted outcomes over the treated observations.

RA is a venerable estimator. See [Lane and Nelder \(1982\)](#); [Cameron and Trivedi \(2005, chap. 25\)](#); [Wooldridge \(2010, chap. 21\)](#); and [Vittinghoff et al. \(2012, chap. 9\)](#). The usefulness of RA has been periodically questioned in the literature because it relies on specifying functional forms for the conditional means and because it requires having sufficient observations of each covariate pattern in each treatment level; see [Rubin \(1973\)](#) for an early salvo. Our experience is that RA is an exceptionally useful base-case estimator. We describe its relative advantages and disadvantages in the course of covering other estimators.

IPW estimators

IPW estimators use weighted averages of the observed outcome variable to estimate means of the potential outcomes. The weights account for the missing data inherent in the potential-outcome framework. Each weight is the inverse of the estimated probability that an individual receives a treatment level. Outcomes of individuals who receive a likely treatment get a weight close to one. Outcomes of individuals who receive an unlikely treatment get a weight larger than one, potentially much larger.

IPW estimators model the probability of treatment without any assumptions about the functional form for the outcome model. In contrast, RA estimators model the outcome without any assumptions about the functional form for the probability of treatment model.

IPW estimators become extremely unstable as the overlap assumption gets close to being violated. When the overlap assumption is nearly violated, some of the inverse-probability weights become very large, IPW estimators produce erratic estimates, and the large-sample distribution provides a poor approximation to the finite-sample distribution of IPW estimators. This instability occurs even though the functional form for the treatment model is correctly specified.

In contrast, when the overlap assumption is nearly violated, there are very few observations in a treatment level for some covariate patterns, so RA estimators use the model to predict in regions in which there are very little data. If the model is well specified and there are “enough” observations, an RA estimator will not become unstable as quickly as an IPW estimator, and the large-sample distribution of the RA estimator still provides a good approximation to the finite-sample distribution. However, in real situations in which “all models are approximate”, relying on a correctly specified outcome model with little data is extremely risky.

IPW estimators are a general approach to missing-data problems that obey some CI assumptions. While IPW is an old idea in statistics that dates back to [Horvitz and Thompson \(1952\)](#), biostatisticians and econometricians have been actively working on extending it to handle modern problems and estimation methods. See [Robins and Rotnitzky \(1995\)](#); [Robins, Rotnitzky, and Zhao \(1994, 1995\)](#); and [Wooldridge \(2002, 2007\)](#). IPW has been used extensively in the modern treatment-effect estimation literature. See [Imbens \(2000\)](#); [Hirano, Imbens, and Ridder \(2003\)](#); [Tan \(2010\)](#); [Wooldridge \(2010, chap. 19\)](#); [van der Laan and Robins \(2003\)](#); and [Tsiatis \(2006, chap. 6\)](#).

To see the intuition behind IPW, consider a study with observed outcome variable y , treatment variable $t \in \{0, 1\}$, and potential outcomes y_0 and y_1 . As part of this process, we need to estimate the POM for treatment $t = 1$, $E(y_1)$. Using the observed data, $y_i t_i$ is y_{1i} when $t = 1$, but y_{1i} is unobserved when $t = 0$. An IPW estimator for $E(y_1)$ is $1/N \sum_{i=1}^N y_i t_i / p(\mathbf{x}_i)$, where $p(\mathbf{x}_i)$ is the probability that $t_i = 1$ and is a function of the covariates $\mathbf{x}_i$. If y_{1i} were always observed, the weights would all equal 1. This IPW estimator places a larger weight on those observations for which y_{1i} is observed even though its observation was not likely.

AIPW estimators

Instead of modeling either the outcome, like RA, or the treatment probability, like IPW, augmented inverse-probability-weighted (AIPW) estimators model both the outcome and the treatment probability. A surprising fact is that only one of the two models must be correctly specified to consistently estimate the treatment effects, a property of the AIPW estimators known as being “doubly robust”. Given that two models instead of one are used, it is less surprising that the AIPW estimators can be more efficient than either the RA or the IPW estimators, though deriving this result is rather technical and relies on the theory of semiparametric estimators.

Intuitively, the AIPW estimator is an IPW that includes an augmentation term that corrects the estimator when the treatment model is misspecified. When the treatment is correctly specified, the augmentation term vanishes as the sample size becomes large. Like the IPW, the AIPW does not perform well when the predicted treatment probabilities are too close to zero or one.

AIPW estimators emerge naturally from a technique of producing more efficient estimators from estimators that have a few main parameters of interest and some auxiliary, or nuisance, parameters used in estimating the few main parameters. This method constructs efficient estimating equations for the main parameters that are orthogonal to the auxiliary parameters. The estimators produced by this method are known as efficient-influence function (EIF) estimators.

To gain some intuition, consider finding an EIF estimator from an IPW estimator for two POMs. Note that we only care about the two POM parameters and not about the many auxiliary parameters used to estimate the treatment probabilities. EIF estimators project the equations that yield the POM parameters onto the equations that yield the auxiliary treatment-model parameters and then use the residuals from this projection to estimate the POM parameters.

We refer to these estimators as “AIPW estimators” instead of “EIF estimators” because the former is commonly used in the biostatistical literature for some of the binary-treatment estimators and because the term “augmented inverse-probability-weighted” tells more about how these estimators relate to the other implemented estimators; see [Tsiatis \(2006\)](#) and [Tan \(2010\)](#). The estimators implemented in `teffects aipw` with the `wpls` option are based on those of [Rubin and van der Laan \(2008\)](#), which did well in simulations reported by [Tan \(2010\)](#), and denoted as $\tilde{\alpha}_{RV}(\hat{\pi})$ in [Tan \(2010, 663\)](#).

When either the outcome model or the treatment model is well specified, the AIPW estimators implemented in `teffects aipw` are more robust than either the RA or the IPW estimators because the AIPW estimators are doubly robust but the RA and IPW estimators are not. When both the outcome and the treatment model are misspecified, which estimator is more robust is a matter of debate in the literature; see [Kang and Schafer \(2007\)](#) and [Robins et al. \(2007\)](#) for some debate, and see [Tan \(2010\)](#) for a more recent discussion.

To the best of our knowledge, there is no general solution to the question of which estimator performs best when both the outcome and the treatment models are misspecified. We suspect that the answer depends on the true models, the implemented specifications, and the polynomials in the covariates used. To help users through this process, the estimators implemented in `teffects` offer many functional forms to approximate either the outcome process or the treatment process. In addition, Stata’s factor-variable notation makes it easy to include polynomials in the covariates. Both of these approximation methods rely on having enough data. `teffects` also makes it easy to compare the results produced by different estimators.

The literature on these methods is vast and growing. For double-robust results and explanations, see [Robins and Rotnitzky \(1995\)](#); [Robins, Rotnitzky, and Zhao \(1995\)](#); [van der Laan and Robins \(2003, chap. 6\)](#); [Bang and Robins \(2005\)](#); [Tsiatis \(2006, chap. 13\)](#); [Wooldridge \(2007; 2010, sec. 21.3.4\)](#); and [Tan \(2010\)](#).

IPWRA estimators

Like AIPW estimators, inverse-probability-weighted regression-adjustment (IPWRA) estimators combine models for the outcome and treatment status; also like AIPW estimators, IPWRA estimators are doubly robust. IPWRA estimators emerge naturally from a robust approach to missing-data methods. IPWRA estimators use the inverse of the estimated treatment-probability weights to estimate missing-data-corrected regression coefficients that are subsequently used to compute the POMs.

As far as we know, there is no literature that compares the relative efficiency of AIPW estimators, which emerge from a general approach to creating efficient estimators, and the IPWRA estimators, which emerge from a robust-correction approach to missing-data analysis.

The IPWRA estimators are also known as “Wooldridge’s double-robust” estimators because they were derived in [Wooldridge \(2007\)](#) and discussed at length in [Wooldridge \(2010, section 21.3.4\)](#).

Nearest-neighbor matching estimators

Matching estimators use an average of the outcomes of the nearest individuals to impute the missing potential outcome for each sampled individual. The difference between the observed outcome and the imputed potential outcome is an estimate of the individual-level treatment effect. These estimated individual-level treatment effects are averaged to estimate the ATE or the ATET.

`teffects nnmatch` determines the “nearest” by using a weighted function of the covariates for each observation. This type of matching is known as nearest-neighbor matching (NNM). `teffects psmatch` determines the “nearest” by using the estimated treatment probabilities, which are known as the propensity scores. This second type of matching is known as propensity-score matching (PSM).

NNM is nonparametric in that no explicit functional form for either the outcome model or the treatment model is specified. This flexibility comes at a price; the estimator needs more data to get to the true value than an estimator that imposes a functional form. More formally, the NNM estimator converges to the true value at a rate slower than the parametric rate, which is the square root of the sample size, when matching on more than one continuous covariate. `teffects nnmatch` uses bias correction to fix this problem. PSM provides an alternative to bias correction because it matches on a single continuous covariate, the estimated treatment probabilities.

Abadie and Imbens (2006, 2011) derived the rate of convergence of the NNM estimator and the bias-corrected NNM estimator and the large-sample distributions of the NNM and the bias-corrected NNM estimators. These articles provided the formal results that built on methods suggested in Rubin (1973, 1977).

`teffects nnmatch` is based on the results in Abadie and Imbens (2006, 2011) and a previous implementation in Abadie et al. (2004).

Propensity-score matching estimators

Instead of performing bias correction to handle the case of more than one continuous covariate, a common solution is to combine all the covariate information into estimated treatment probabilities, known as propensity scores, and use this single continuous covariate as the matching variable.

The term “propensity score” is widely used, but we continue to refer to it as the “treatment probability” to be consistent with the other estimators. We call the estimator that matches on the estimated treatment probabilities the “propensity-score matching (PSM) estimator” because the latter term is ubiquitous.

In effect, the PSM estimator parameterizes the bias-correction term in the treatment probability model. One advantage of matching on the estimated treatment probabilities over the bias-correction method is that one can explore the fit of different treatment probability models using standard methods before performing the nonparametric matching. For example, one can select the treatment model by minimizing an information criterion under i.i.d. sampling. We know of no counterpart for selecting the proper order of the bias-correction term for the NNM estimator.

Matching on estimated treatment probability models has been very popular since Rosenbaum and Rubin (1983) showed that if adjusting for covariates $\mathbf{x}_i$ is sufficient to estimate the effects, then one can use the probability of treatment to perform the adjustment. Abadie and Imbens (2012) derived a method to estimate the standard errors of the estimator that matches on estimated treatment probabilities, and this method is implemented in `teffects psmatch`.

Choosing among estimators

There is no definitive way to select one of the estimators implemented in `teffects` over the others. We offer three observations about the tradeoffs among the estimators.

First, if the outcome model is correctly specified, the RA estimator will break down more slowly than the IPW, AIPW, IPWRA, or PSM estimators as the overlap assumption begins to fail. This result depends critically on the ability of the RA estimator to predict into regions in which there are little data.

Second, if the overlap assumption holds, the AIPW and IPWRA estimators have the double-robust property for some functional form combinations. The double-robust property says that if either the outcome model or the treatment model is correctly specified, we can consistently estimate the effects. The properties of double-robust estimators when both models are misspecified are not known, although there is some discussion in the literature about the properties of the AIPW estimators; see [Kang and Schafer \(2007\)](#), [Robins et al. \(2007\)](#), and [Tan \(2010\)](#).

Third, all the estimators require the same assumptions, so if each is correctly specified, they should all produce similar results. Of course, just because they produce similar results does not mean that they are correctly specified; it is possible that they are just behaving similarly in response to some underlying problem.

Model choice

`teffects` offers a broad selection of functional form combinations so that you can choose a combination that fits your data. Picking a functional form that respects the values of the observed outcomes is usually best. Select `linear` for continuous outcomes over the real line; `logit`, `probit`, or `hetprobit` for binary outcomes; and `poisson` for counts or nonnegative outcomes.

For binary treatments, you can select among `logit`, `probit`, or `hetprobit` models. For multivalued treatments, only the multinomial logit is available to model the treatment probabilities.

Selecting a functional form of a given set of covariates is a famously difficult problem in statistics. In the treatment-effects context, [Cattaneo, Drukker, and Holland \(2013\)](#) found that model selection by minimizing an information criterion worked well. [Cattaneo, Drukker, and Holland \(2013\)](#) discuss a method and a community-contributed command to facilitate the process.

Acknowledgments

We thank Matias Cattaneo, Guido Imbens, Barbara Sianesi, and Jeffrey Wooldridge for helpful conversations and advice about the literature and methods in this area.

We thank Matias Cattaneo for his collaboration on [Cattaneo, Drukker, and Holland \(2013\)](#) and for many conversations about [Cattaneo \(2010\)](#), a part of which is implemented in `teffects ipw` and `teffects aipw`.

We thank Ashley Holland for her collaboration on [Cattaneo, Drukker, and Holland \(2013\)](#).

We thank Alberto Abadie, Jane Herr, and Guido Imbens for collaboration on [Abadie et al. \(2004\)](#), which is implemented in `teffects nnmatch`.

We thank Matias Cattaneo for sharing an extract of the dataset that he used in [Cattaneo \(2010\)](#) and Barbara Sianesi for sharing an extract of the dataset that she used in [Blundell, Dearden, and Sianesi \(2005\)](#).

We thank Alberto Abadie, Guido Imbens, and László Sándor for sharing their implementation of the standard errors derived in [Abadie and Imbens \(2012\)](#), which are implemented in `teffects psmatch`.

We thank Edwin Leuven and Barbara Sianesi for their inspirational command `psmatch2`, which computes several of the estimators in `teffects` and a few more that we have not implemented.

References

- Abadie, A., D. M. Drukker, J. L. Herr, and G. W. Imbens. 2004. [Implementing matching estimators for average treatment effects in Stata](#). *Stata Journal* 4: 290–311.
- Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267. <https://doi.org/10.1111/j.1468-0262.2006.00655.x>.
- . 2011. Bias-corrected matching estimators for average treatment effects. *Journal of Business and Economic Statistics* 29: 1–11. <https://doi.org/10.1198/jbes.2009.07333>.
- . 2012. Matching on the estimated propensity score. Working Paper 15301, National Bureau of Economic Research. <https://doi.org/10.3386/w15301>.
- Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist’s Companion*. Princeton, NJ: Princeton University Press.
- Bang, H., and J. M. Robins. 2005. Doubly robust estimation in missing data and causal inference models. *Biometrics* 61: 962–972. <https://doi.org/10.1111/j.1541-0420.2005.00377.x>.
- Becker, S. O., and M. Caliendo. 2007. [Sensitivity analysis for average treatment effects](#). *Stata Journal* 7: 71–83.
- Blundell, R. W., L. Dearden, and B. Sianesi. 2005. Evaluating the effect of education on earnings: Models, methods and results from the National Child Development Survey. *Journal of the Royal Statistical Society, A ser.*, 168: 473–512. <https://doi.org/10.1111/j.1467-985X.2004.00360.x>.
- Brown, G. K., and T. Mergoupis. 2011. [Treatment interactions with nonexperimental data in Stata](#). *Stata Journal* 11: 545–555.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- . 2022. *Microeconometrics Using Stata*. 2nd ed. College Station, TX: Stata Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. [Estimation of multivalued treatment effects under conditional independence](#). *Stata Journal* 13: 407–450.
- Cattaneo, M. D., R. Titiunik, and G. Vazquez-Bare. 2020. [Analysis of regression-discontinuity designs with multiple cutoffs or multiple scores](#). *Stata Journal* 20: 866–891.
- Cerulli, G. 2015. [ctreatreg: Command for fitting dose-response models under exogenous and endogenous treatment](#). *Stata Journal* 15: 1019–1045.
- . 2017. [Identification and estimation of treatment effects in the presence of \(correlated\) neighborhood interactions: Model and Stata implementation via ntreatreg](#). *Stata Journal* 17: 803–833.
- . 2022. *Econometric Evaluation of Socio-Economic Programs: Theory and Applications*. 2nd ed. Berlin: Springer.
- Daniel, R. M., B. L. De Stavola, and S. N. Cousens. 2011. [gformula: Estimating causal effects in the presence of time-varying confounding or mediation using the g-computation formula](#). *Stata Journal* 11: 479–517.
- Deb, P., E. C. Norton, and W. G. Manning. 2017. *Health Econometrics Using Stata*. College Station, TX: Stata Press.
- Drukker, D. M. 2016. [A generalized regression-adjustment estimator for average treatment effects from panel data](#). *Stata Journal* 16: 826–836.
- Galiani, S., and B. Quistorff. 2017. [The synth_runner package: Utilities to automate synthetic control estimation using synth](#). *Stata Journal* 17: 834–849.
- Heckman, J. J. 1997. Instrumental variables: A study of implicit behavioral assumptions used in making program evaluations. *Journal of Human Resources* 32: 441–462. <https://doi.org/10.2307/146178>.
- Heckman, J. J., and S. Navarro-Lozano. 2004. Using matching, instrumental variables, and control functions to estimate economic choice models. *Review of Economics and Statistics* 86: 30–57. <https://doi.org/10.1162/003465304323023660>.
- Heß, S. 2017. [Randomization inference with Stata: A guide and software](#). *Stata Journal* 17: 630–651.
- Hirano, K., G. W. Imbens, and G. Ridder. 2003. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica* 71: 1161–1189. <https://doi.org/10.1111/1468-0262.00442>.

- Holland, P. W. 1986. Statistics and causal inference. *Journal of the American Statistical Association* 81: 945–960. <https://doi.org/10.2307/2289064>.
- Horvitz, D. G., and D. J. Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association* 47: 663–685. <https://doi.org/10.1080/01621459.1952.10483446>.
- Imbens, G. W. 2000. The role of the propensity score in estimating dose–response functions. *Biometrika* 87: 706–710. <https://doi.org/10.1093/biomet/87.3.706>.
- . 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29. <https://doi.org/10.1162/003465304323023651>.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86. <https://doi.org/10.1257/jel.47.1.5>.
- Kang, J. D. Y., and J. L. Schafer. 2007. Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data. *Statistical Science* 22: 523–539. <https://doi.org/10.1214/07-STS227>.
- Lane, P. W., and J. A. Nelder. 1982. Analysis of covariance and standardization as instances of prediction. *Biometrics* 38: 613–621. <https://doi.org/10.2307/2530043>.
- Nichols, A. 2007. Causal inference with observational data. *Stata Journal* 7: 507–541.
- Robins, J. M. 1986. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical Modeling* 7: 1393–1512. [https://doi.org/10.1016/0270-0255\(86\)90088-6](https://doi.org/10.1016/0270-0255(86)90088-6).
- Robins, J. M., and A. Rotnitzky. 1995. Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association* 90: 122–129. <https://doi.org/10.2307/2291135>.
- Robins, J. M., A. Rotnitzky, and L. P. Zhao. 1994. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association* 89: 846–866. <https://doi.org/10.2307/2290910>.
- . 1995. Analysis of semiparametric regression models for repeated outcomes in the presence of missing data. *Journal of the American Statistical Association* 90: 106–121. <https://doi.org/10.1080/01621459.1995.10476493>.
- Robins, J. M., M. Sued, Q. Lei-Gomez, and A. Rotnitzky. 2007. Comment: Performance of double-robust estimators when “inverse probability” weights are highly variable. *Statistical Science* 22: 544–559. <https://doi.org/10.1214/07-STS227D>.
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.
- Rubin, D. B. 1973. Matching to remove bias in observational studies. *Biometrics* 29: 159–183. <https://doi.org/10.2307/2529684>.
- . 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. <https://doi.org/10.1037/h0037350>.
- . 1977. Assignment to treatment group on the basis of a covariate. *Journal of Educational Statistics* 2: 1–26. <https://doi.org/10.2307/1164933>.
- Rubin, D. B., and M. J. van der Laan. 2008. Empirical efficiency maximization: Improved locally efficient covariate adjustment in randomized experiments and survival analysis. *International Journal of Biostatistics* 4: art. 5. <https://doi.org/10.2202/1557-4679.1084>.
- Sasaki, Y., and T. Ura. 2022. Average treatment effect estimates robust to the “limited overlap” problem: robustate. *Stata Journal* 22: 344–354.
- Tan, Z. 2010. Bounded, efficient and doubly robust estimation with inverse weighting. *Biometrika* 97: 661–682. <https://doi.org/10.1093/biomet/asq035>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- van der Laan, M. J., and J. M. Robins. 2003. *Unified Methods for Censored Longitudinal Data and Causality*. New York: Springer. <https://doi.org/10.1007/978-0-387-21700-0>.
- Vittinghoff, E., D. V. Glidden, S. C. Shiboski, and C. E. McCulloch. 2012. *Regression Methods in Biostatistics: Linear, Logistic, Survival, and Repeated Measures Models*. 2nd ed. New York: Springer.
- Wooldridge, J. M. 2002. Inverse probability weighted M-estimators for sample selection, attrition, and stratification. *Portuguese Economic Journal* 1: 117–139. <https://doi.org/10.1007/s10258-002-0008-x>.

- . 2007. Inverse probability weighted estimation for general missing data problems. *Journal of Econometrics* 141: 1281–1301. <https://doi.org/10.1016/j.jeconom.2007.02.002>.
- . 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

- [CAUSAL] **teffects** — Treatment-effects estimation for observational data
- [CAUSAL] **teffects intro** — Introduction to treatment effects for observational data
- [CAUSAL] **teffects multivalued** — Multivalued treatment effects

[Description](#)
[Options](#)
[References](#)

[Quick start](#)
[Remarks and examples](#)
[Also see](#)

[Menu](#)
[Stored results](#)

[Syntax](#)
[Methods and formulas](#)

Description

`teffects aipw` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational data by augmented inverse-probability weighting (AIPW). AIPW estimators combine aspects of regression-adjustment and inverse-probability-weighted methods. AIPW estimators have the double-robust property. `teffects aipw` accepts a continuous, binary, count, fractional, or nonnegative outcome and allows a multivalued treatment.

See [\[CAUSAL\] teffects intro](#) or [\[CAUSAL\] teffects intro advanced](#) for more information about estimating treatment effects from observational data.

Quick start

ATE of binary treatment `treat2` by AIPW using a linear model for outcome `y1` on `x1` and `x2` and a logistic model for `treat2` on `x1` and `w`

```
teffects aipw (y1 x1 x2) (treat2 x1 w)
```

Same as above, but use a fractional logistic model for fractional outcome `y2`

```
teffects aipw (y2 x1 x2, flogit) (treat2 x1 w)
```

Same as above, but use a heteroskedastic probit model for binary outcome `y3` and a probit model for `treat2`

```
teffects aipw (y3 x1 x2, hetprobit(x1 x2)) (treat2 x1 w, probit)
```

ATE for each level of three-valued treatment `treat3` on `y1`

```
teffects aipw (y1 x1 x2) (treat3 x1 w)
```

Same as above, and specify that `treat3 = 3` is the control level

```
teffects aipw (y1 x1 x2) (treat3 x1 w), control(3)
```

Same as above, specified using the label “MyControl” corresponding to `treat3 = 3`

```
teffects aipw (y1 x1 x2) (treat3 x1 w), control(MyControl)
```

ATET of binary treatment `treat2` by AIPW using a linear model for outcome `y1` on `x1` and `x2` and a probit model for `treat2` on `x1` and `w`

```
teffects aipw (y1 x1 x2) (treat2 x1 w, probit), atet
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Augmented inverse-probability weighting

Statistics > Causal inference/treatment effects > Binary outcomes > Augmented inverse-probability weighting

Statistics > Causal inference/treatment effects > Count outcomes > Augmented inverse-probability weighting

Statistics > Causal inference/treatment effects > Fractional outcomes > Augmented inverse-probability weighting

Statistics > Causal inference/treatment effects > Nonnegative outcomes > Augmented inverse-probability weighting

Syntax

```
teffects aipw (ovar omvarlist [ , omodel noconstant ])  
              (tvar tmvarlist [ , tmodel noconstant ]) [if] [in] [weight]  
              [ , stat options ]
```

ovar is a binary, count, continuous, fractional, or nonnegative outcome of interest.

omvarlist specifies the covariates in the outcome model.

tvar must contain integer values representing the treatment levels.

tmvarlist specifies the covariates in the treatment-assignment model.

<i>omodel</i>	Description
Model	
linear	linear outcome model; the default
logit	logistic outcome model
probit	probit outcome model
hetprobit(<i>varlist</i>)	heteroskedastic probit outcome model
poisson	exponential outcome model
flogit	fractional logistic outcome model
fprobit	fractional probit outcome model
fhetsprobit(<i>varlist</i>)	fractional heteroskedastic probit outcome model

omodel specifies the model for the outcome variable.

<i>tmodel</i>	Description
Model	
logit	logistic treatment model; the default
probit	probit treatment model
hetprobit(<i>varlist</i>)	heteroskedastic probit treatment model

tmodel specifies the model for the treatment variable.

For multivalued treatments, only `logit` is available and multinomial logit is used.

<i>stat</i>	Description
Stat	
ate	estimate average treatment effect in population; the default
atet	estimate average treatment effect on the treated
pomeans	estimate potential-outcome means

options	Description
Model	
nls	estimate conditional means by nonlinear least squares
wnls	estimate conditional means by weighted nonlinear least squares
SE/Robust	
vce(vcetype)	vcetype may be <u>robust</u> , <u>cluster</u> <i>clustvar</i> , <u>bootstrap</u> , or <u>jackknife</u>
Reporting	
level(#)	set confidence level; default is level(95)
aequations	display auxiliary-equation results
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process; seldom used
Advanced	
pstolerance(#)	set tolerance for overlap assumption
osample(newvar)	newvar identifies observations that violate the overlap assumption
control(# label)	specify the level of tvar that is the control
tlevel(# label)	specify the level of tvar that is the treatment
coeflegend	display legend instead of statistics

omvarlist and tmvarlist may contain factor variables; see [U] 11.4.3 Factor variables.

bayesboot, bootstrap, by, collect, jackknife, and statsby are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the bootstrap prefix; see [R] bootstrap.

fweights, iweights, and pweights are allowed; see [U] 11.1.6 weight.

coeflegend does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

Model
noconstant; see [R] Estimation options.
nls specifies that the parameters of the outcome model be estimated by nonlinear least squares instead of the default maximum likelihood.
wnls specifies that the parameters of the outcome model be estimated by weighted nonlinear least squares instead of the default maximum likelihood. The weights make the estimator of the effect parameters more robust to a misspecified outcome model.
Stat
stat is one of three statistics: ate, atet, or pomeans. ate is the default.
ate specifies that the average treatment effect be estimated.
atet specifies that the average treatment effect on the treated be estimated.
pomeans specifies that the potential-outcome means for each treatment level be estimated.

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [\[R\] vce_option](#).

Reporting

`level(#)`; see [\[R\] Estimation options](#).

`aequations` specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`display_options`: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no!stretch`; see [\[R\] Estimation options](#).

Maximization

`maximize_options`: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [\[R\] Maximize](#). These options are seldom used.

`init_specs` is one of

```
matname [ , skip copy]
# [ , # ...], copy
```

Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `teffects` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption.

`control(#|label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`.

`tlevel(#|label)` specifies the level of `tvar` that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may be specified only with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `teffects aipw` but is not shown in the dialog box:

`coeflegend`; see [\[R\] Estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Video example](#)

Overview

AIPW estimators use inverse-probability weights to correct for the missing-data problem arising from the fact that each subject is observed in only one of the potential outcomes; these estimators use an augmentation term to correct the estimator in case the treatment model is misspecified. If the treatment model is correctly specified, the augmentation term goes to zero in large samples.

AIPW estimators compute averages of the augmented inverse-probability-weighted outcomes for each treatment level. Contrasts of these averages provide estimates of the treatment effects.

AIPW estimators use a model to predict treatment status, and they use another model to predict outcomes. Because of the double-robust property, only one of these two models must be correctly specified for the AIPW estimator to be consistent.

AIPW estimators use a three-step approach to estimating treatment effects:

1. They estimate the parameters of the treatment-assignment model and compute inverse-probability weights.
2. They estimate separate regression models of the outcome for each treatment level and obtain the treatment-specific predicted outcomes for each subject.
3. They compute the weighted means of the treatment-specific predicted outcomes, where the weights include the inverse-probability weights computed in step 1. The contrasts of these weighted averages provide the estimates of the treatment effects.

These steps produce consistent estimates of the effect parameters because treatment assignment is assumed to be independent of the potential outcomes after conditioning on the covariates. The overlap assumption ensures that predicted inverse-probability weights do not get too large. The standard errors reported by `teffects aipw` correct for the three-step process. See [\[CAUSAL\] teffects intro](#) or [\[CAUSAL\] teffects intro advanced](#) for more information about this estimator.

We will illustrate the use of `teffects aipw` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by [Cattaneo \(2010\)](#). This dataset also contains information about each mother's age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), and whether this baby was the mother's first birth (`fbaby`).

► Example 1: Estimating the ATE

We begin by using `teffects aipw` to estimate the average treatment effect of `mbsmoke` on `bweight`. We use a probit model to predict treatment status as a function of `mmarried`, `mage`, and `fbaby`; to maximize the predictive power of this model, we use factor-variable notation to incorporate quadratic effects of the mother’s age, the only continuous covariate in our model. We use linear regression to model birthweight, using `prenatal1`, `mmarried`, `mage`, and `fbaby` as explanatory variables. We type

```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)

. teffects aipw (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)

Iteration 0:  EE criterion = 4.629e-21
Iteration 1:  EE criterion = 1.936e-25

Treatment-effects estimation          Number of obs      =      4,642
Estimator      : augmented IPW
Outcome model  : linear by ML
Treatment model: probit
```

bweight	Robust					
	Coefficient	std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke						
(Smoker						
vs						
Nonsmoker)	-230.9892	26.21056	-8.81	0.000	-282.361	-179.6174
POMean						
mbsmoke						
Nonsmoker	3403.355	9.568472	355.68	0.000	3384.601	3422.109

The average birthweight if all mothers were to smoke would be 231 grams less than the average of 3,403 grams that would occur if none of the mothers had smoked.



By default, `teffects aipw` reports the ATE and the POM for the base (untreated) subjects. The `pomeans` option allows us to view the treated subjects’ POM as well; the `aequations` option displays the regression model coefficients used to predict the POMs as well as the coefficients from the model used to predict treatment.

► Example 2: Displaying the POMs and equations

Here we use the `pomeans` and `aequations` options to obtain estimates of both POMs and view all the fitted equations underlying our estimates:

```
. teffects aipw (bweight prenatal1 mmarried mage fbaby)
> (mb smoke mmarried c.mage##c.mage fbaby medu, probit), pomeans aequations

Iteration 0: EE criterion = 4.629e-21
Iteration 1: EE criterion = 6.876e-26

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : augmented IPW
Outcome model  : linear by ML
Treatment model: probit
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
POMeans						
mb smoke						
Nonsmoker	3403.355	9.568472	355.68	0.000	3384.601	3422.109
Smoker	3172.366	24.42456	129.88	0.000	3124.495	3220.237
OME0						
prenatal1	64.40859	27.52699	2.34	0.019	10.45669	118.3605
mmarried	160.9513	26.6162	6.05	0.000	108.7845	213.1181
mage	2.546828	2.084324	1.22	0.222	-1.538373	6.632028
fbaby	-71.3286	19.64701	-3.63	0.000	-109.836	-32.82117
_cons	3202.746	54.01082	59.30	0.000	3096.886	3308.605
OME1						
prenatal1	25.11133	40.37541	0.62	0.534	-54.02302	104.2457
mmarried	133.6617	40.86443	3.27	0.001	53.5689	213.7545
mage	-7.370881	4.21817	-1.75	0.081	-15.63834	.8965804
fbaby	41.43991	39.70712	1.04	0.297	-36.38461	119.2644
_cons	3227.169	104.4059	30.91	0.000	3022.537	3431.801
TME1						
mmarried	-.6484821	.0554173	-11.70	0.000	-.757098	-.5398663
mage	.1744327	.0363718	4.80	0.000	.1031452	.2457202
c.mage#						
c.mage	-.0032559	.0006678	-4.88	0.000	-.0045647	-.0019471
fbaby	-.2175962	.0495604	-4.39	0.000	-.3147328	-.1204595
medu	-.0863631	.0100148	-8.62	0.000	-.1059917	-.0667345
_cons	-1.558255	.4639691	-3.36	0.001	-2.467618	-.6488926

The coefficient table indicates that the treated POM is 3,172 grams, 231 grams less than the untreated POM. The sections of the table labeled OME0 and OME1 represent the linear regression coefficients for the untreated and treated potential-outcome equations, respectively. The coefficients of the TME1 equation are used in the probit model to predict treatment status.

◀

As is well known, the standard probit model assumes that the error terms in the latent-utility framework are homoskedastic; the model is not robust to departures from this assumption. An alternative is to use the heteroskedastic probit model, which explicitly models the error variance as a function of a set of variables.

► Example 3: Heteroskedastic probit treatment model

Here we refit our model as in the previous examples, but we instead use heteroskedastic probit to model the treatment variable. We posit that the heteroskedasticity is a function of the mother's age. We type

```
. teffects aipw (bweight prenatal1 mmarried fbaby)
> (mb smoke mmarried c.mage##c.mage fbaby medu, hetprobit(c.mage)), aequations

Iteration 0: EE criterion = 1.746e-19
Iteration 1: EE criterion = 4.222e-26

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : augmented IPW
Outcome model  : linear by ML
Treatment model: heteroskedastic probit
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mb smoke (Smoker vs Nonsmoker)	-230.2699	27.35327	-8.42	0.000	-283.8814	-176.6585
POMean						
mb smoke Nonsmoker	3403.657	9.540713	356.75	0.000	3384.957	3422.356
OME0						
prenatal1	69.5048	27.04642	2.57	0.010	16.49479	122.5148
mmarried	173.74	24.63865	7.05	0.000	125.4491	222.0308
fbaby	-79.19473	18.62584	-4.25	0.000	-115.7007	-42.68875
_cons	3260.768	28.29282	115.25	0.000	3205.315	3316.221
OME1						
prenatal1	12.86437	39.83916	0.32	0.747	-65.21894	90.94768
mmarried	113.3491	39.47422	2.87	0.004	35.9811	190.7172
fbaby	64.22326	38.42042	1.67	0.095	-11.07939	139.5259
_cons	3051.268	37.30413	81.79	0.000	2978.153	3124.383
TME1						
mmarried	-.3551755	.1044199	-3.40	0.001	-.5598347	-.1505162
mage	.0831898	.0349088	2.38	0.017	.0147699	.1516097
c.mage#						
c.mage	-.0013458	.0006659	-2.02	0.043	-.002651	-.0000406
fbaby	-.1170697	.044998	-2.60	0.009	-.2052643	-.0288752
medu	-.0435057	.0147852	-2.94	0.003	-.0724842	-.0145272
_cons	-.8757021	.347814	-2.52	0.012	-1.557405	-.1939993
TME1_lnsigma						
mage	-.0236336	.0107134	-2.21	0.027	-.0446315	-.0026357

The equation labeled TME1_lnsigma represents the heteroskedasticity function used to model the logarithm of the variance. Looking at the ATE and POM estimates as well as their standard errors, we can see that they are almost identical to the previous results. Thus, allowing for heteroskedasticity did not change any of our previous conclusions.

Rather than using maximum likelihood to fit the outcome model, you can instruct `teffects aipw` to use a weighted nonlinear least-squares (WNLS) estimator instead. The WNLS estimator may be more robust to outcome model misspecification.

► Example 4: Using the WNLS estimator

Here we use WNLS to fit our outcome model:

```
. teffects aipw (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit), wnls

Iteration 0:  EE criterion = 2.788e-20
Iteration 1:  EE criterion = 9.910e-25

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : augmented IPW
Outcome model  : linear by WNLS
Treatment model: probit
```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE mbsmoke (Smoker vs Nonsmoker)	-227.1956	27.34794	-8.31	0.000	-280.7966	-173.5946
POmean mbsmoke Nonsmoker	3403.251	9.596622	354.63	0.000	3384.442	3422.06

The ATE of -227 is slightly greater than the ATE of -231 estimated in [example 1](#). The estimated POMs are nearly indistinguishable.



➤ Example 5: Estimating ATET

So far, we have discussed estimates of ATE. Here we wish to estimate the ATET. We do so simply by specifying option atet:

```
. teffects aipw (bweight fbaby mage mmarried prenatal1)
> (mb smoke fbaby foreign medu mmarried, probit), atet

Iteration 0: EE criterion = 2.079e-19
Iteration 1: EE criterion = 2.149e-25

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : augmented IPW
Outcome model  : linear by ML
Treatment model: probit
```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATET						
mb smoke						
(Smoker						
vs						
Nonsmoker)	-228.0266	23.18451	-9.84	0.000	-273.4674	-182.5858
POMean						
mb smoke						
Nonsmoker	3365.686	13.4482	250.27	0.000	3339.328	3392.044

The estimated POM is 3,366, which tells us that, if all smoking mothers in the population instead chose not to smoke, the average birthweight of their babies would be 3,366 grams. The observed birthweight among this group of mothers, however, is lower on average by 228 grams, which we learn from the estimated ATET of -228 .



► Example 6: Estimating ATET with multivalued treatment

We can also estimate ATETs with multivalued treatments. Here we use the variable `msmoke` as our treatment variable, which has four levels representing one control group and three treatment groups:

```
. teffects aipw (bweight fbaby mage mmarried prenatal1)
> (msmoke fbaby foreign medu mmarried), atet

Iteration 0: EE criterion = 7.107e-13
Iteration 1: EE criterion = 3.101e-26

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : augmented IPW
Outcome model  : linear by ML
Treatment model: (multinomial) logit
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET						
msmoke						
(1-5 daily						
vs						
0 daily)	-156.7646	36.7927	-4.26	0.000	-228.8769	-84.65219
(6-10 daily						
vs						
0 daily)	-209.2045	35.01555	-5.97	0.000	-277.8337	-140.5753
(11+ daily						
vs						
0 daily)	-220.3197	33.84588	-6.51	0.000	-286.6564	-153.983
POMean						
msmoke						
0 daily	3351.16	14.88082	225.20	0.000	3321.994	3380.325

When estimating ATET with multivalued treatments, we need to specify not only the group to be taken as the control group but also the group to be taken as the treated. By default, `teffects` uses the first level of the treatment variable as the control group (here `msmoke = 0`) and the second level as the group of treated (`msmoke = 1`). To change this, we could specify the `control()` and `tlevel()` options, but here we leave it at the default.

The POM shown at the bottom of the output is an estimate of $E\{Y_i(0)|t = 1\}$. It tells us that, among the population of light-smoking mothers (smoking 1–5 cigarettes per day), the birthweight of their babies would be 3,351 grams on average if none of them chose to smoke. This average birthweight is higher by 157 grams compared with the observed birthweight among women of this group, as indicated by the first ATET in the output table, which is an estimate of $E\{Y_i(1) - Y_i(0)|t = 1\}$. The second ATET in the output is an estimate of $E\{Y_i(2) - Y_i(0)|t = 1\}$. That is, among the group of light-smoking mothers, if they were to smoke more (that is, 6–10 cigarettes per day), the difference in average birthweight would now grow to 209 grams. Likewise, the third ATET in the output (an estimate of $E\{Y_i(3) - Y_i(0)|t = 1\}$) tells us that, if light-smoking mothers were to smoke even more, the average birthweight of their babies would be 220 grams lower than the average birthweight in the counterfactual case where none of these mothers smoked.



Video example

Treatment effects: Augmented inverse-probability weighting

Stored results

teffects aipw stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	teffects
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	aipw
<code>e(tmodel)</code>	logit, probit, or hetprobit
<code>e(omodel)</code>	linear, logit, probit, hetprobit, poisson, flogit, fprobit, or fhetprobit
<code>e(stat)</code>	statistic estimated, ate, atet, or pomeans
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(cme)</code>	ml, nls, or wnls
<code>e(vce)</code>	vcetype specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	b V
<code>e(estat_cmd)</code>	program used to implement estat
<code>e(predict)</code>	program used to implement predict
<code>e(marginsnotok)</code>	predictions disallowed by margins
<code>e(asbalanced)</code>	factor variables fvset as asbalanced
<code>e(asobserved)</code>	factor variables fvset as asobserved

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

The methods and formulas presented here provide the technical details underlying the estimators implemented in `teffects ra`, `teffects ipw`, `teffects aipw`, and `teffects ipwra`. See [Methods and formulas](#) of [\[CAUSAL\] teffects nnmatch](#) for the methods and formulas used by `teffects nnmatch` and `teffects psmatch`.

Methods and formulas are presented under the following headings:

- Parameters and notation
- Overview of EE estimators
- VCE for EE estimators
- TM and OM estimating functions
 - TM estimating functions
 - logit and probit
 - hetprobit
 - mlogit
 - OM estimating functions
 - OM QML
 - linear
 - logit, flogit, probit, and fprobit
 - hetprobit and fheterprobit
 - poisson
 - OM WNL
 - linear
 - logit, flogit, probit, and fprobit
 - hetprobit and fheterprobit
 - poisson
- Effect estimating functions
 - RA estimators
 - RA for POM
 - RA for ATE
 - RA for ATET
 - IPW estimators
 - IPW for POM
 - IPW for ATE
 - IPW for ATET
 - AIPW estimators
 - AIPW for POM
 - AIPW for ATE
 - AIPW for ATET
 - IPWRA estimators

Parameters and notation

We begin by reviewing the effect parameters estimated by `teffects` and some essential notation.

The potential outcome that an individual would obtain if given treatment level $t \in \{0, 1, \dots, q\}$ is y_t . Each y_t is a random variable, the realizations of which are y_{ti} . Throughout this document, i subscripts denote realizations of the corresponding, unsubscripted random variables.

The three parameters of interest are

1. the potential-outcome mean (POM) $\alpha_t = E(y_t)$;
2. the average treatment effect (ATE) $\tau_t = E(y_t - y_0)$; and
3. the average treatment effect on the treated (ATET) $\delta_t = E(y_t - y_0 | t = \tilde{t})$.

The no-treatment level is 0.

The estimators implemented in `teffects` use three assumptions to justify the equations used for estimation and inference about the effect parameters of interest:

1. Conditional mean independence (CMI) allows us to estimate potential-outcome means from the observed outcomes in the sample.
2. Overlap ensures that we have data on each type of individual in each treatment level.

3. Independent observations ensure that the outcome and treatment for one individual has no effect on the outcome or treatment for any other individual.

`teffects ra` implements some regression-adjustment (RA) estimators; `teffects ipw` implements some inverse-probability-weighted (IPW) estimators; `teffects ipwra` implements some inverse-probability-weighted regression-adjustment (IPWRA) estimators; and `teffects aipw` implements some augmented inverse-probability-weighted (AIPW) estimators. All are implemented as estimating-equation (EE) estimators. The estimators are consistent and asymptotically normally distributed under the CMI, overlap, and independence assumptions.

Overview of EE estimators

EE estimators compute estimates by solving sample estimating equations. The sample estimating equations are the sample equivalents of population expectation equations.

Each EE estimator specifies a set of estimating equations for the effect parameters of interest and a set of estimating equations for the auxiliary parameters in the outcome model (OM) or the treatment model (TM). The next few sections provide tremendous detail about the estimating equations that define the RA, IPW, AIPW, and IPWRA estimators.

Ignoring the details for a moment, EE estimators solve systems of equations to compute estimates. A standard robust estimator is consistent for the variance of the estimator (VCE). All the details involve the equations specified by choices of estimator and functional forms for the OM or TM.

When used, the OM is a model for the conditional mean of the outcome variable. We let $\mu(\mathbf{x}, t, \beta_t)$ denote a conditional-mean model for the outcome y conditional on covariates $\mathbf{x}$ and treatment level t . Mathematically, $E(y|\mathbf{x}, t) = \mu(\mathbf{x}, t, \beta_t)$, where β_t are the parameters of the conditional-mean model given treatment level t . The table below provides details about the available functional forms.

Outcome model	Functional form for $\mu(\mathbf{x}, t, \beta_t)$
linear	$\mathbf{x}\beta_t$
logit, flogit	$\exp(\mathbf{x}\beta_t) / \{1 + \exp(\mathbf{x}\beta_t)\}$
probit, fprobit	$\Phi(\mathbf{x}\beta_t)$
poisson	$\exp(\mathbf{x}\beta_t)$
hetprobit, fheterprobit	$\Phi\{\dot{\mathbf{x}}\dot{\beta}_t / \exp(\ddot{\mathbf{x}}\ddot{\beta}_t)\}$

In the cases of `hetprobit` and `fheterprobit`, we use $\dot{\mathbf{x}}$ and $\dot{\beta}_t$ to denote the variables and parameters in the index function, and we use $\ddot{\mathbf{x}}$ and $\ddot{\beta}_t$ to denote the variables and parameters in the variance equation. We define $\mathbf{x}' = (\dot{\mathbf{x}}', \ddot{\mathbf{x}}')$ and $\beta'_t = (\dot{\beta}_t', \ddot{\beta}_t')$.

We write the vector of parameters for the outcome model over all treatment levels as $\beta' = (\beta'_0, \beta'_1, \dots, \beta'_q)$.

Next we provide details about the estimating equations implied by each functional form choice.

When used, the TM is a model for the conditional probability of treatment. We let $p(\mathbf{z}, t, \gamma)$ denote the conditional probability model for the probability that a person receives treatment t , conditional on covariates $\mathbf{z}$. The table below provides details about the functional form options allowed in the case of a binary treatment.

Treatment model	Functional form for $p(\mathbf{z}, t, \gamma)$
logit	$\exp(\mathbf{z}\gamma)/\{1 + \exp(\mathbf{z}\gamma)\}$
probit	$\Phi(\mathbf{z}\gamma)$
hetprobit	$\Phi\{\dot{\mathbf{z}}\dot{\gamma}/\exp(\ddot{\mathbf{z}}\ddot{\gamma})\}$

In the case of `hetprobit`, we use $\dot{\mathbf{z}}$ and $\dot{\gamma}$ to denote the variables and parameters in the index function, and we use $\ddot{\mathbf{z}}$ and $\ddot{\gamma}$ to represent the variables and parameters in the variance equation. We define $\mathbf{z}' = (\dot{\mathbf{z}}', \ddot{\mathbf{z}}')$, and $\gamma' = (\dot{\gamma}', \ddot{\gamma}')$.

In the multivalued-treatment case, $p(\mathbf{z}, t, \gamma)$ is specified as a multinomial logit with $p(\mathbf{z}, t, \gamma) = \exp(\mathbf{z}\gamma_t)/\{1 + \sum_{k=1}^q \exp(\mathbf{z}\gamma_k)\}$ and $\gamma' = (\gamma'_1, \gamma'_2, \dots, \gamma'_q)$. (We present formulas for the case with treatment level 0 as the base with $\gamma'_0 = \mathbf{0}'$; see [R] [mlogit](#) for background.) In `teffects`, the `logit` option in the treatment-model specification means binary logit for the binary-treatment case and multinomial logit for the multivalued-treatment case: this simplifies the use of the command and makes statistical sense.

Below we provide details about the estimating equations implied by each functional form choice. The effect parameters of interest are

1. the POMs denoted by $\alpha' = (\alpha_0, \alpha_1, \dots, \alpha_q)$;
2. the ATEs denoted by $\tau' = (\tau_1, \tau_2, \dots, \tau_q)$; and
3. the ATETs denoted by $\delta' = (\delta_1, \delta_2, \dots, \delta_q)$.

We denote the effect parameters by ϑ and all the parameters in any particular case by θ . More formally, θ is the concatenation of the effect parameters, the OM parameters, and the TM parameters; $\theta' = (\vartheta', \beta', \gamma')$, where ϑ is α , τ , or δ , and β or γ may not be present, depending on the case at hand.

In the subsections below, we discuss estimators for the elements in θ in detail and note how these elements change over the cases defined by effect parameters and estimators. The parameter vector θ denotes all the parameters, no matter which particular case is under consideration.

The EE estimators described in this section are defined by a set of equations,

$$E\{\mathbf{s}(\mathbf{x}, \mathbf{z}, \theta)\} = \mathbf{0}$$

where $\mathbf{s}(\mathbf{x}, \mathbf{z}, \theta)$ is a vector of estimating functions. Note the notation: estimating equations equate the expected value of a vector of estimating functions to zero.

Because each of the estimating functions has mean zero, we can construct estimators that find the estimates $\hat{\theta}$ by solving a system of equations,

$$1/N \sum_i^N \mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta}) = \mathbf{0}$$

where $\mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})$ are the sample realizations of the estimating functions. In other words, the parameter estimates set the average of the realizations of each estimating function to zero. Almost all the details below involve specifying the sample realizations $\mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})$.

Estimators that set the expected value of estimating functions to zero are known as estimating-equations (EE) estimators, M estimators, or Z estimators in the statistics literature and as generalized method of moments (GMM) estimators in the econometrics literature. See [van der Vaart \(1998, 41\)](#), [Stefanski and Boos \(2002\)](#), and [Tsiatis \(2006, sec. 3.2\)](#) for statistics; and see [Wooldridge \(2010, chap. 14\)](#), [Cameron and Trivedi \(2005, chap. 6\)](#), and [Newey and McFadden \(1994\)](#) for econometrics.

We refer to them as EE estimators because this name is closest to being independent of any discipline. The estimators are implemented using `gmm` because they are exactly identified generalized method of moments (GMM) estimators. When weights are specified by the user, they are applied to the estimating equations just as `gmm` applies user-specified weights.

Each estimator has a set of estimating equations for the effect parameters and either an OM or a TM, or both. The OM parameters or the TM parameters are auxiliary parameters used to estimate the effect parameters of interest.

Each set of parameters has its own set of sample estimating equations:

$1/N \sum_i \mathbf{s}_{e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}}) = \mathbf{0}$ are the sample estimating equations for the effect parameters. These equations determine the effect parameter estimates $\hat{\boldsymbol{\theta}}$ as functions of the data and the other estimated parameters.

$1/N \sum_i \mathbf{s}_{\text{om},i}(\mathbf{x}_i, w_i, \hat{\boldsymbol{\beta}}) = \mathbf{0}$ are the sample estimating equations for OM parameters that use the weights w_i , which are functions of the TM parameters.

$1/N \sum_i \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}}) = \mathbf{0}$ are the sample estimating equations for TM parameters.

The whole set of sample estimating functions is $\mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}})$ with

$$\mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}})' = (\mathbf{s}_{e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}})', \mathbf{s}_{\text{om},i}(\mathbf{x}_i, w_i(t), \hat{\boldsymbol{\beta}})', \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})')$$

although not all the estimators have each of three components.

VCE for EE estimators

The Huber/White/robust sandwich estimator is consistent for the variance–covariance of the estimator (VCE). See [van der Vaart \(1998, 41\)](#), [Stefanski and Boos \(2002\)](#), and [Tsiatis \(2006, sec. 3.2\)](#) for statistics; and see [Wooldridge \(2010, chap. 14\)](#), [Cameron and Trivedi \(2005, chap. 6\)](#), and [Newey and McFadden \(1994\)](#) for econometrics.

The formula is

$$\widehat{\mathbf{V}} = (1/N) \overline{\mathbf{G}} \overline{\mathbf{S}} \overline{\mathbf{G}}'$$

where

$$\overline{\mathbf{G}} = \left\{ (1/N) \sum_i \frac{\partial \mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}})}{\partial \hat{\boldsymbol{\theta}}} \right\}^{-1}$$

and

$$\overline{\mathbf{S}} = (1/N) \sum_i \mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}}) \mathbf{s}_i(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}})'$$

The matrix $\overline{\mathbf{G}}$ is not symmetric because our EE estimators come from stacking moment conditions instead of optimizing a single objective function. The implication is that the robust formula should always be used because, even under correct specification, the nonsymmetric $\overline{\mathbf{G}}$ and the symmetric $\overline{\mathbf{S}}$ converge to different matrices.

TM and OM estimating functions

Although the sample estimating functions for the effect parameters, the $\mathbf{s}_{e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\boldsymbol{\theta}})$, are estimator specific, the sample estimating functions for the TM parameters, the $\mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})$, and the sample estimating functions for the OM parameters, the $\mathbf{s}_{\text{om},i}(\mathbf{x}_i, w_i(t), \hat{\boldsymbol{\beta}})'$, are used in multiple estimators. We provide details about the TM and the OM sample estimating functions here.

TM estimating functions

The sample estimating functions used to estimate the parameters of the TM $p(\mathbf{z}, t, \gamma)$ are the sample score equations from the quasimaximum likelihood (QML) estimator.

In the binary-treatment case, $p(\mathbf{z}, t, \gamma)$ may be logit, probit, or heteroskedastic probit. In the multivalued-treatment case, $p(\mathbf{z}, t, \gamma)$ is a multinomial logit. We now give formulas for the $\mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})$ for each case.

logit and probit

In the logit and probit cases,

$$\mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}}) = \left[\frac{g(\mathbf{z}_i \hat{\boldsymbol{\gamma}}') \{t_i - G(\mathbf{z}_i \hat{\boldsymbol{\gamma}}')\}}{G(\mathbf{z}_i \hat{\boldsymbol{\gamma}}') \{1 - G(\mathbf{z}_i \hat{\boldsymbol{\gamma}}')\}} \right] \mathbf{z}_i$$

where $G(z)$ is the logistic cumulative distribution function for the logit, $G(z)$ is the normal cumulative distribution function for the probit, and $g(\cdot) = \{\partial G(z)\}/(\partial z)$ is the corresponding density function.

hetprobit

In the hetprobit case, there are two sets of sample score equations, $\mathbf{s}_{\text{tm},1,i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})$ and $\mathbf{s}_{\text{tm},2,i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})$:

$$\mathbf{s}_{\text{tm},1,i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}}) = \left(\frac{\phi\{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\} [t_i - \Phi\{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\}]}{\Phi\{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\} [1 - \Phi\{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\}] \exp(\ddot{\mathbf{z}}_i \hat{\boldsymbol{\gamma}}')} \right) \dot{\mathbf{z}}_i$$

and

$$\mathbf{s}_{\text{tm},2,i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}}) = \left(\frac{\phi\{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\} \dot{\mathbf{z}}_i \hat{\boldsymbol{\gamma}}' [\Phi\{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\} - t_i]}{\Phi\{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\} [1 - \Phi\{q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})\}] \exp(\ddot{\mathbf{z}}_i \hat{\boldsymbol{\gamma}}')} \right) \ddot{\mathbf{z}}_i$$

where $\phi(\cdot)$ is the standard normal density function, and $q(\mathbf{z}_i, \hat{\boldsymbol{\gamma}}) = (\dot{\mathbf{z}}_i \hat{\boldsymbol{\gamma}}' / \exp(\ddot{\mathbf{z}}_i \hat{\boldsymbol{\gamma}}'))$.

mlogit

In the mlogit case, $p(\mathbf{z}, t, \gamma) = \exp(\mathbf{z}\boldsymbol{\gamma}_t) / \{1 + \sum_{k=1}^q \exp(\mathbf{z}\boldsymbol{\gamma}_k)\}$. We present formulas for the case with treatment level 0 as the base with $\boldsymbol{\gamma}'_0 = \mathbf{0}'$; see [R] **mlogit** for background.

There are q vectors of sample estimating functions for the mlogit case, $\mathbf{s}_{\text{tm},k,i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})$ for each $k \in \{1, \dots, q\}$, 1 for each vector $\hat{\boldsymbol{\gamma}}_k$, $k \in \{1, \dots, q\}$. These sample estimating functions are

$$\mathbf{s}_{\text{tm},k,i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}}) = \begin{cases} \{1 - p(\mathbf{z}_i, k, \hat{\boldsymbol{\gamma}})\} \mathbf{z}'_i & t_i = k \\ -p(\mathbf{z}_i, k, \hat{\boldsymbol{\gamma}}) \mathbf{z}'_i & \text{otherwise} \end{cases}$$

OM estimating functions

The parameters of the OM $\mu(\mathbf{x}, t, \beta_t)$ are estimated either by weighted QML or by weighted nonlinear least squares. The estimating functions used to estimate the parameters of the OM are either the score equations from the weighted QML estimator or the moment conditions for the weighted nonlinear least-squares estimator.

The estimating functions for the OM parameters in $\mu(\mathbf{x}, t, \beta_t)$ vary over the models for the conditional mean because $\mu(\mathbf{x}, t, \beta_t)$ may be linear, logit, probit, heteroskedastic probit, or poisson.

Let N_t be the number of observations in treatment level t , and let $t_i(t) = 1$ if $t_i = t$, with $t_i(t) = 0$ if $t_i \neq t$.

There are two sets of sample estimating functions for the OM parameters with weights $w_i(t)$:

1. $\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\beta}_t\}$ are the sample estimating functions for the weighted QML estimator.
2. $\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\beta}_t\}$ are the sample estimating functions for the weighted nonlinear least-squares estimator.

OM QML

Here are the formulas for the $\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\beta}_t\}$ for each functional form choice.

linear

In the linear case,

$$\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\beta}_t\} = w_i(t)t_i(t)(y_i - \mathbf{x}_i\widehat{\beta}_t')\mathbf{x}_i'$$

logit, flogit, probit, and fprobit

In the logit, flogit, probit, and fprobit cases,

$$\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\beta}_t\} = w_i(t)t_i(t) \left[\frac{g(\mathbf{x}_i\widehat{\beta}_t') \{y_i - G(\mathbf{x}_i\widehat{\beta}_t')\}}{G(\mathbf{x}_i\widehat{\beta}_t') \{1 - G(\mathbf{x}_i\widehat{\beta}_t')\}} \right] \mathbf{x}_i$$

where $G(z)$ is the logistic cumulative distribution function for the logit and flogit, $G(z)$ is the normal cumulative distribution function for the probit and fprobit, and $g(\cdot) = \{\partial G(z)/(\partial z)\}$ is the corresponding density function.

hetprobit and fheterprobit

In the hetprobit and fheterprobit cases, there are two sets of sample score equations, $\mathbf{s}_{\text{ml,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\beta}_t\}$ and $\mathbf{s}_{\text{ml,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\beta}_t\}$:

$$\mathbf{s}_{\text{ml,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\beta}_t\} = w_i(t)t_i(t) \left(\frac{\phi\{q(\mathbf{x}_i, \widehat{\beta}_t)\} [y_i - \Phi\{q(\mathbf{x}_i, \widehat{\beta}_t)\}]}{\Phi\{q(\mathbf{x}_i, \widehat{\beta}_t)\} [1 - \Phi\{q(\mathbf{x}_i, \widehat{\beta}_t)\}] \exp(\ddot{\mathbf{x}}_i\widehat{\beta}_t')} \right) \dot{\mathbf{x}}_i'$$

and

$$\mathbf{s}_{\text{ml,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\beta}_t\} = w_i(t)t_i(t) \left(\frac{\phi\{q(\mathbf{x}_i, \widehat{\beta}_t)\} \dot{\mathbf{x}}_i\widehat{\beta}_t' [\Phi\{q(\mathbf{x}_i, \widehat{\beta}_t)\} - y_i]}{\Phi\{q(\mathbf{x}_i, \widehat{\beta}_t)\} [1 - \Phi\{q(\mathbf{x}_i, \widehat{\beta}_t)\}] \exp(\ddot{\mathbf{x}}_i\widehat{\beta}_t')} \right) \ddot{\mathbf{x}}_i'$$

where $\phi(\cdot)$ is the standard normal density function, $\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}' = [\mathbf{s}_{\text{ml,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}', \mathbf{s}_{\text{ml,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}']$, and $q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t) = \left(\dot{\mathbf{x}}_i \widehat{\boldsymbol{\beta}}_t' / \exp(\ddot{\mathbf{x}}_i \widehat{\boldsymbol{\beta}}_t') \right)$.

poisson

In the poisson case,

$$\mathbf{s}_{\text{ml,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t)\{y_i - \exp(\mathbf{x}_i \widehat{\boldsymbol{\beta}}_t')\} \mathbf{x}_i'$$

OM WNL

Here are the formulas for the $\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}$ for each functional form choice.

linear

In the linear case,

$$\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t)(y_i - \mathbf{x}_i \widehat{\boldsymbol{\beta}}_t') \mathbf{x}_i'$$

logit, flogit, probit, and fprobit

In the logit, flogit, probit, and fprobit cases,

$$\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left[g(\mathbf{x}_i \widehat{\boldsymbol{\beta}}_t') \left\{ y_i - G(\mathbf{x}_i \widehat{\boldsymbol{\beta}}_t') \right\} \right] \mathbf{x}_i$$

where $G(z)$ is the logistic cumulative distribution function for the logit and flogit, $G(z)$ is the normal cumulative distribution function for the probit and fprobit, and $g(\cdot) = \{\partial G(z)/(\partial z)\}$ is the corresponding density function.

hetprobit and fheterprobit

In the hetprobit and fheterprobit cases, there are two sets of sample score equations, $\mathbf{s}_{\text{nls,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}$ and $\mathbf{s}_{\text{nls,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}$:

$$\mathbf{s}_{\text{nls,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left(\frac{\phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\}}{\exp(\ddot{\mathbf{x}}_i \widehat{\boldsymbol{\beta}}_t')} [y_i - \Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\}] \right) \dot{\mathbf{x}}_i'$$

and

$$\mathbf{s}_{\text{nls,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t) \left(\frac{\phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\}}{\exp(\ddot{\mathbf{x}}_i \widehat{\boldsymbol{\beta}}_t')} \dot{\mathbf{x}}_i \widehat{\boldsymbol{\beta}}_t' [\Phi\{q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t)\} - y_i] \right) \dot{\mathbf{x}}_i'$$

where $\phi(\cdot)$ is the standard normal density function, $\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}' = [\mathbf{s}_{\text{nls,om},1,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}', \mathbf{s}_{\text{nls,om},2,i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\}']$, and $q(\mathbf{x}_i, \widehat{\boldsymbol{\beta}}_t) = \left(\dot{\mathbf{x}}_i \widehat{\boldsymbol{\beta}}_t' / \exp(\ddot{\mathbf{x}}_i \widehat{\boldsymbol{\beta}}_t') \right)$.

poisson

In the poisson case,

$$\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, w_i(t), \widehat{\boldsymbol{\beta}}_t\} = w_i(t)t_i(t)\{y_i - \exp(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t')\} \exp(\mathbf{x}_i\widehat{\boldsymbol{\beta}}_t')\mathbf{x}_i'$$

Effect estimating functions

We now describe the sample estimating functions for the effect parameters, which vary over estimator and effect parameter.

RA estimators

RA estimators estimate the effect parameters using means of the observation-level predictions of the conditional means of the outcomes. There is no model for the conditional probability of treatment.

The RA estimators use unweighted QML estimators to estimate the parameters of the conditional mean model. In other words, the RA estimators use the sample estimating functions $\mathbf{s}_{\text{ml,om},i}(\mathbf{x}_i, 1, \widehat{\boldsymbol{\beta}})$, given above.

For the RA estimators, the vector of sample estimating functions is the concatenation of the sample estimating functions for the effect parameters with the sample estimating functions for the OM parameters. Algebraically,

$$\mathbf{s}_{\text{ra},i}(\mathbf{x}_i, \widehat{\boldsymbol{\theta}})' = \mathbf{s}_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\beta}})', \mathbf{s}_{\text{ml,om},i}(\mathbf{x}_i, 1, \widehat{\boldsymbol{\beta}})'$$

The estimating functions $\mathbf{s}_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\beta}})'$ vary over the effect parameter.

RA for POM

The RA estimators for the POM parameters estimate $\boldsymbol{\theta}' = (\boldsymbol{\alpha}', \boldsymbol{\beta}')$ using two types of estimating equations: 1) those for the POM parameters $\boldsymbol{\alpha}$, and 2) those for the conditional-mean model parameters $\boldsymbol{\beta}_t$ in $\mu(\mathbf{x}, t, \boldsymbol{\beta}_t)$.

The sample estimating functions for the $\widehat{\boldsymbol{\beta}}_t$ are given in [OM estimating functions](#) above.

The elements of $\mathbf{s}_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}})$ for the POM parameters $\boldsymbol{\alpha}$ are given by

$$\mu(\mathbf{x}_i, t, \widehat{\boldsymbol{\beta}}_t) - \widehat{\alpha}_t \tag{RAPOM}$$

RA for ATE

The RA estimators for the ATE parameters estimate $\boldsymbol{\theta}' = (\boldsymbol{\tau}', \boldsymbol{\beta}')$ using two types of estimating equations: 1) those for the ATE parameters $\boldsymbol{\tau}$, and 2) those for the OM parameters $\boldsymbol{\beta}_t$ in $\mu(\mathbf{x}, t, \boldsymbol{\beta}_t)$.

The sample estimating functions that determine the $\widehat{\boldsymbol{\beta}}_t$ are given in [OM estimating functions](#) with $w_i(t) = 1$.

The elements of $\mathbf{s}_{\text{ra},e,i}(\mathbf{x}_i, \widehat{\boldsymbol{\tau}}, \widehat{\boldsymbol{\beta}})$ for the ATE parameters $\boldsymbol{\tau}$ are given by

$$\mu(\mathbf{x}_i, t, \widehat{\boldsymbol{\beta}}_t) - \mu(\mathbf{x}_i, 0, \widehat{\boldsymbol{\beta}}_t) - \widehat{\tau}_t \tag{RAATE}$$

RA for ATET

The RA estimators for the ATET parameters estimate $\theta' = (\delta', \beta')$ using two types of estimating equations: 1) those for the ATET parameters δ , and 2) those for the OM parameters β_t in $\mu(\mathbf{x}, t, \beta_t)$.

The sample estimating functions that determine the $\hat{\beta}_t$ are given in *OM estimating functions* above with $w_i(t) = 1$.

The elements of $\mathbf{s}_{\text{ra},e,i}(\mathbf{x}_i, \hat{\delta}, \hat{\beta})$ for the ATET parameters δ are given by

$$Nt_i(\tilde{t})/N_{\tilde{t}} \left\{ \mu(\mathbf{x}_i, t, \hat{\beta}_t) - \mu(\mathbf{x}_i, 0, \hat{\beta}_t) - \hat{\delta}_t \right\} \quad (\text{RAATET})$$

IPW estimators

IPW estimators estimate the effect parameters using means of the observed outcomes weighted by the inverse probability of treatment. There is no outcome model. The IPW estimators use QML estimators to estimate the parameters of the conditional probability model.

The vector of estimating functions is the concatenation of the estimating functions for the effect parameters with the estimating functions for the conditional-probability parameters. The sample estimating functions used by the IPW estimators are

$$\mathbf{s}_{\text{ipw},i}(\mathbf{x}_i, \hat{\theta})' = \mathbf{s}_{\text{ipw},e,i}(\mathbf{x}_i, \hat{\theta}, \hat{\gamma})', \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, 1, \hat{\gamma})'$$

The estimating functions $\mathbf{s}_{\text{ipw},e,i}(\mathbf{z}_i, \hat{\theta}, \hat{\gamma})'$ vary over the effect parameter.

All the IPW estimators use normalized inverse-probability weights. These weights are not related to the weights $w_i(t)$ that were used in the OM equations. The functional form for the normalized inverse-probability weights varies over the effect parameters POM, ATE, and ATET.

The POM and ATE estimators use normalized inverse-probability weights. The unnormalized weights for individual i and treatment level t are $d_i(t) = t_i(t)/p(\mathbf{z}_i, t, \hat{\gamma})$, and the normalized weights are $\bar{d}_i(t) = N_t d_i(t) / \sum_i^N d_i(t)$.

The ATET estimator uses normalized treatment-adjusted inverse-probability weights. The treatment-adjusted inverse-probability weights adjust the inverse-probability weights for the probability of getting the conditional treatment $\tilde{t}$. The unnormalized weights are $f_i = p(\mathbf{z}_i, \tilde{t}, \hat{\gamma})/p(\mathbf{z}_i, t_i, \hat{\gamma})$, and the normalized weights are $\bar{f}_i = N f_i / \sum_i^N f_i$.

The IPW effect estimators are weighted cell averages. The sample estimating functions used in POM estimators are the sample estimating functions from weighted OLS regression on treatment-cell indicators. The POM estimators use a full set of $q + 1$ of treatment indicator variables $\mathbf{a}_i$. (The i th observation on the k th variable in $\mathbf{a}_i$ is 1 if i received treatment $k - 1$ and 0 otherwise, for $k \in \{1, 2, \dots, q + 1\}$.)

The sample estimating functions used in the ATE and the ATET estimators are the sample estimating functions from weighted OLS regression on treatment-cell indicators, excluding the indicator for the control level and including a constant term. The variables $\tilde{\mathbf{a}}_i$ used in the ATE and ATET sample estimating functions include q of treatment indicator variables and a variable that is always 1. (The first q variables in $\tilde{\mathbf{a}}_i$ are treatment indicators: the i th observation on the k th variable in $\tilde{\mathbf{a}}_i$ is 1 if i received treatment k and 0 otherwise, for $k \in \{1, 2, \dots, q\}$. The $(q + 1)$ th variable is always 1.) This definition of $\tilde{\mathbf{a}}_i$ sets the last treatment level to be the control; renaming the treatments handles the more general case allowed for by `teffects`.

Having defined $\mathbf{a}_i$ and $\tilde{\mathbf{a}}_i$, we can give the sample estimating functions that the IPW estimators use for the effects parameters.

IPW for POM

We begin with the IPW estimators for the potential-outcome means. In this case, $\theta' = (\alpha', \gamma')$.

The sample estimating functions for the $\hat{\gamma}$ are given in [TM estimating functions](#) above.

The sample estimating functions for $\hat{\alpha}$ are given by

$$\mathbf{s}_{\text{ipw},e,i,t}(\mathbf{z}_i, \hat{\alpha}, \hat{\gamma})' = \bar{d}_i(t)(y_i - \mathbf{a}_i \hat{\alpha}) \mathbf{a}_i' \quad (\text{IPWPOM})$$

IPW for ATE

The full parameter vector for the IPW estimators for the ATE is $\theta' = (\tau', \gamma')$.

The sample estimating functions for the $\hat{\gamma}$ are given in [TM estimating functions](#) above.

The sample estimating functions for $\hat{\tau}$ are given by

$$\mathbf{s}_{\text{ipw},e,i,t}(\mathbf{z}_i, \hat{\tau}, \hat{\gamma})' = \bar{d}_i(t)(y_i - \tilde{\mathbf{a}}_i \hat{\tau}) \tilde{\mathbf{a}}_i' \quad (\text{IPWATE})$$

IPW for ATET

The full parameter vector for the IPW estimators for the ATET is $\theta' = (\delta', \gamma')$.

The sample estimating functions for the $\hat{\gamma}$ are given in [TM estimating functions](#) above.

The sample estimating functions for $\hat{\delta}$ are given by

$$\mathbf{s}_{\text{ipw},e,i,t}(\mathbf{z}_i, \hat{\delta}, \hat{\gamma})' = \bar{f}_i(t)(y_i - \tilde{\mathbf{a}}_i \hat{\delta}) \tilde{\mathbf{a}}_i' \quad (\text{IPWATET})$$

AIPW estimators

This section documents the sample estimating functions used by the augmented inverse-probability-weighted (AIPW) estimators implemented in `teffects`. These AIPW estimators are efficient-influence-function estimators as discussed in [\[CAUSAL\] teffects intro](#) and [\[CAUSAL\] teffects intro advanced](#). The `teffects` implementation was primarily inspired by [Cattaneo, Drukker, and Holland \(2013\)](#), which was based on [Cattaneo \(2010\)](#). [Tan \(2010\)](#) was influential by identifying the implemented weighted nonlinear least-squares estimator as having relatively good properties when both the conditional mean function and the conditional probability function are misspecified. The ATET implementation follows the moment functions outlined in [Farrell \(2015\)](#).

The AIPW estimating functions for the treatment parameters include terms from a conditional probability model and from a conditional mean model for the outcome.

The sample-estimation-equations vector has three parts for the AIPW estimators:

$$\mathbf{s}_{\text{aipw},i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})' = [\mathbf{s}_{\text{aipw},e,i}(\mathbf{x}_i, \mathbf{z}_i, \hat{\theta})', \mathbf{s}_{\text{aipw},\text{tm},i}(\mathbf{z}_i, \hat{\gamma})', \mathbf{s}_{\text{aipw},\text{om},i}(\mathbf{x}_i, w_i(t), \hat{\beta})']$$

The sample estimating functions for the parameters of the TM are the $\mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \hat{\gamma})$ given in [TM estimating functions](#) above.

`teffects aipw` implements three AIPW estimators: the standard AIPW estimator, the NLS AIPW estimator, and the WNLS AIPW estimator.

The three AIPW estimators differ in how they estimate the parameters of the OM.

By default, `teffects aipw` uses the standard AIPW estimator that estimates the parameters of the OM by the unweighted ML estimator, whose sample estimating functions, $\mathbf{s}_{\text{ml,om},i}(\mathbf{x}_i, 1, \widehat{\beta})$, are given in [OM estimating functions](#) above.

When the `nls` option is specified, `teffects aipw` uses the NLS AIPW estimator that estimates the parameters of the OM by the unweighted NLS estimator, whose sample estimating functions, $\mathbf{s}_{\text{nls,om},i}(\mathbf{x}_i, 1, \widehat{\beta})$, are given in [OM estimating functions](#) above.

When the `wnls` option is specified, `teffects aipw` uses the WNLS AIPW estimator that estimates the parameters of the OM by the WNLS estimator, whose sample estimating functions, $\mathbf{s}_{\text{nls,om},i}\{\mathbf{x}_i, \widehat{w}_i(t), \widehat{\beta}\}$, are given in [OM estimating functions](#) above. The weights for person i in treatment level t are

$$\widehat{w}_i(t) = \frac{t_i(t)}{p(\mathbf{z}_i, t, \widehat{\gamma})} \left\{ \frac{t_i(t)}{p(\mathbf{z}_i, t, \widehat{\gamma})} - 1 \right\} \quad (\text{WNLSW})$$

Now we discuss the sample estimating functions for the effect parameters, the $\mathbf{s}_{e,i}(\mathbf{x}_i, \mathbf{z}_i, \widehat{\theta})$.

AIPW for POM

We begin with the AIPW estimators for the potential-outcome means. In this case, $\theta' = (\alpha', \gamma', \beta')$, and the elements of $\mathbf{s}_{\text{aipw},e,i}(\mathbf{x}_i, \mathbf{z}_i, \widehat{\theta})$ are given by

$$\frac{t_i(t)}{p(\mathbf{z}_i, t, \widehat{\gamma})} y_i - \mu(\mathbf{x}_i, \widehat{\beta}_t) \left\{ \frac{t_i(t)}{p(\mathbf{z}_i, t, \widehat{\gamma})} - 1 \right\} - \alpha_t$$

AIPW for ATE

The AIPW estimators for the ATE estimate $\theta' = (\tau', \gamma', \beta')$, and the elements of $\mathbf{s}_{\text{aipw},e,i}(\mathbf{x}_i, \mathbf{z}_i, \widehat{\theta})$ are given by

$$\begin{aligned} & \frac{t_i(t)}{p(\mathbf{z}_i, t, \widehat{\gamma})} y_i - \mu(\mathbf{x}_i, \widehat{\beta}_t) \left\{ \frac{t_i(t)}{p(\mathbf{z}_i, t, \widehat{\gamma})} - 1 \right\} - \tau_0 \text{ if } t = 0 \\ & \frac{t_i(t)}{p(\mathbf{z}_i, t, \widehat{\gamma})} y_i - \mu(\mathbf{x}_i, \widehat{\beta}_t) \left\{ \frac{t_i(t)}{p(\mathbf{z}_i, t, \widehat{\gamma})} - 1 \right\} - \tau_t - \tau_0 \text{ if } t > 0 \end{aligned}$$

AIPW for ATET

The AIPW estimators for the ATET estimate $\theta' = (\delta', \gamma', \beta')$ and the elements of $\mathbf{s}_{\text{aipw},e,i}(\mathbf{x}_i, \mathbf{z}_i, \widehat{\theta})$ are given by

$$\begin{aligned} & \frac{t_i(\tilde{t})}{\widehat{p}_{\tilde{t}}} \mu(\mathbf{x}_i, \widehat{\beta}_{\tilde{t}}) + \frac{t_i(t)}{\widehat{p}_{\tilde{t}}} \frac{p(\mathbf{z}_i, \tilde{t}, \widehat{\gamma}) \{y_i - \mu(\mathbf{x}_i, \widehat{\beta}_{\tilde{t}})\}}{p(\mathbf{z}_i, t, \widehat{\gamma})} - \frac{t_i(\tilde{t})}{\widehat{p}_{\tilde{t}}} \delta_0 \text{ if } t = 0 \\ & \frac{t_i(\tilde{t})}{\widehat{p}_{\tilde{t}}} \mu(\mathbf{x}_i, \widehat{\beta}_{\tilde{t}}) + \frac{t_i(t)}{\widehat{p}_{\tilde{t}}} \frac{p(\mathbf{z}_i, \tilde{t}, \widehat{\gamma}) \{y_i - \mu(\mathbf{x}_i, \widehat{\beta}_{\tilde{t}})\}}{p(\mathbf{z}_i, t, \widehat{\gamma})} - \frac{t_i(\tilde{t})}{\widehat{p}_{\tilde{t}}} (\delta_t - \delta_0) \text{ if } t > 0 \end{aligned}$$

where $\tilde{t}$ is the conditional treatment, $N_{\tilde{t}}$ is the number of treated observations, and $\widehat{p}_{\tilde{t}} = N_{\tilde{t}}/N$.

IPWRA estimators

The IPWRA estimators combine inverse-probability weighting (IPW) with regression-adjustment (RA) methods. The sample estimating functions for IPWRA include sample estimating functions from both RA and IPW. The vector of sample estimating functions is

$$\mathbf{s}_{\text{ipwra},i}(\mathbf{x}_i, \hat{\boldsymbol{\theta}})' = \mathbf{s}_{\text{ra},e,i}(\mathbf{x}_i, \hat{\boldsymbol{\vartheta}}, \hat{\boldsymbol{\beta}})', \mathbf{s}_{\text{ml},\text{om},i}\{\mathbf{x}_i, w_i(j), \hat{\boldsymbol{\beta}}\}', \mathbf{s}_{\text{tm},i}(\mathbf{z}_i, \hat{\boldsymbol{\gamma}})'$$

where $\hat{\boldsymbol{\theta}}' = (\hat{\boldsymbol{\vartheta}}', \hat{\boldsymbol{\beta}}', \hat{\boldsymbol{\gamma}}')$, $\hat{\boldsymbol{\vartheta}} = \hat{\boldsymbol{\alpha}}$ for POM, $\hat{\boldsymbol{\vartheta}} = \hat{\boldsymbol{\tau}}_t$ for ATE, and $\hat{\boldsymbol{\vartheta}} = \hat{\boldsymbol{\delta}}_t$ for ATET. The sample estimating functions, $\mathbf{s}_{\text{ra},e,i}(\mathbf{x}_i, \hat{\boldsymbol{\vartheta}}, \hat{\boldsymbol{\beta}})$, for POM, ATE, and ATET are given in equations (RAPOM), (RAATE), and (RAATET). For the OM sample estimating functions, $\mathbf{s}_{\text{ml},\text{om},i}(\cdot)$, we replace the RA unity weights, $w_i(t) = 1$, with $d_i(j)$ for POM or ATE and $\bar{f}_i$ for ATET, the normalized inverse-probability weights described in *IPW estimators* above. The specific form of the OM sample estimating functions are given in *OM estimating functions* above. The TM sample estimating functions are given in *TM estimating functions* above.

References

- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. Estimation of multivalued treatment effects under conditional independence. *Stata Journal* 13: 407–450.
- Farrell, M. H. 2015. Robust inference on average treatment effects with possibly more covariates than observations. *Journal of Econometrics* 189: 1–23. <https://doi.org/10.1016/j.jeconom.2015.06.017>.
- Huber, C. 2015. Introduction to treatment effects in Stata: Part 1. *The Stata Blog: Not Elsewhere Classified*. <https://blog.stata.com/2015/07/07/introduction-to-treatment-effects-in-stata-part-1/>.
- Newey, W. K., and D. L. McFadden. 1994. “Large sample estimation and hypothesis testing”. In *Handbook of Econometrics*, edited by R. F. Engle and D. L. McFadden, vol. 4: 2111–2245. Amsterdam: Elsevier. [https://doi.org/10.1016/S1573-4412\(05\)80005-4](https://doi.org/10.1016/S1573-4412(05)80005-4).
- Stefanski, L. A., and D. D. Boos. 2002. The calculus of M-estimation. *American Statistician* 56: 29–38. <https://doi.org/10.1198/000313002753631330>.
- Tan, Z. 2010. Bounded, efficient and doubly robust estimation with inverse weighting. *Biometrika* 97: 661–682. <https://doi.org/10.1093/biomet/asq035>.
- Tsiatis, A. A. 2006. *Semiparametric Theory and Missing Data*. New York: Springer.
- van der Vaart, A. W. 1998. *Asymptotic Statistics*. Cambridge: Cambridge University Press. <https://doi.org/10.1017/CBO9780511802256>.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

- [CAUSAL] [teffects postestimation](#) — Postestimation tools for teffects
- [CAUSAL] [teffects](#) — Treatment-effects estimation for observational data
- [CAUSAL] [teffects ipwra](#) — Inverse-probability-weighted regression adjustment
- [U] [20 Estimation and postestimation commands](#)

Description
Options
References

Quick start
Remarks and examples
Also see

Menu
Stored results

Syntax
Methods and formulas

Description

`teffects ipw` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational data by inverse-probability weighting (IPW). IPW estimators use estimated probability weights to correct for missing data on the potential outcomes. `teffects ipw` accepts a continuous, binary, count, fractional, or nonnegative outcome and allows a multivalued treatment.

See [\[CAUSAL\] teffects intro](#) or [\[CAUSAL\] teffects intro advanced](#) for more information about estimating treatment effects from observational data.

Quick start

ATE of binary `treat2` on `y` by IPW using a logistic model of `treat2` on `x` and `w`

```
teffects ipw (y) (treat2 x w)
```

Same as above, but estimate ATET

```
teffects ipw (y) (treat2 x w), atet
```

Same as above, but estimate potential-outcome means

```
teffects ipw (y) (treat2 x w), pomeans
```

ATE of `treat2` on `y` using heteroskedastic probit for `treat2` as a function of `x` and `w`

```
teffects ipw (y) (treat2 x w, hetprobit(x w))
```

ATE for treatment levels 2 and 3 of three-valued treatment `treat3`

```
teffects ipw (y) (treat3 x w)
```

Same as above, and specify that `treat3 = 3` is the control level

```
teffects ipw (y) (treat3 x w), control(3)
```

Same as above, specified using the label “MyControl” corresponding to `treat3 = 3`

```
teffects ipw (y) (treat3 x w), control(MyControl)
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Inverse-probability weighting (IPW)

Statistics > Causal inference/treatment effects > Binary outcomes > Inverse-probability weighting (IPW)

Statistics > Causal inference/treatment effects > Count outcomes > Inverse-probability weighting (IPW)

Statistics > Causal inference/treatment effects > Fractional outcomes > Inverse-probability weighting (IPW)

Statistics > Causal inference/treatment effects > Nonnegative outcomes > Inverse-probability weighting (IPW)

Syntax

```
teffects ipw (ovar) (tvar tmvarlist [ , tmodel noconstant ]) [ if ] [ in ] [ weight ]  
[ , stat options ]
```

ovar is a binary, count, continuous, fractional, or nonnegative outcome of interest.
tvar must contain integer values representing the treatment levels.
tmvarlist specifies the variables that predict treatment assignment in the treatment model.

<i>tmodel</i>	Description
Model	
<code>logit</code>	logistic treatment model; the default
<code>probit</code>	probit treatment model
<code>hetprobit</code> (<i>varlist</i>)	heteroskedastic probit treatment model
<i>tmodel</i> specifies the model for the treatment variable. For multivalued treatments, only <code>logit</code> is available and multinomial <code>logit</code> is used.	
<i>stat</i>	Description
Stat	
<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated
<code>pomeans</code>	estimate potential-outcome means
<i>options</i>	Description
SE/Robust	
<code>vce</code> (<i>vcetype</i>)	<i>vcetype</i> may be <code>robust</code> , <code>cluster</code> <i>clustvar</i> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level</code> (#)	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
Advanced	
<code>pstolerance</code> (#)	set tolerance for overlap assumption
<code>osample</code> (<i>newvar</i>)	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control</code> (# <i>label</i>)	specify the level of <i>tvar</i> that is the control
<code>tlevel</code> (# <i>label</i>)	specify the level of <i>tvar</i> that is the treatment
<code>coeflegend</code>	display legend instead of statistics

tmvarlist may contain factor variables; see [U] 11.4.3 **Factor variables**.

bayesboot, *bootstrap*, *by*, *collect*, *jackknife*, and *statsby* are allowed; see [U] 11.1.10 **Prefix commands**.

Weights are not allowed with the *bootstrap* prefix; see [R] **bootstrap**.

fweights, *iweights*, and *pweights* are allowed; see [U] 11.1.6 **weight**.

coeflegend does not appear in the dialog box.

See [U] 20 **Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

noconstant; see [R] **Estimation options**.

Stat

stat is one of three statistics: *ate*, *atet*, or *pomeans*. *ate* is the default.

ate specifies that the average treatment effect be estimated.

atet specifies that the average treatment effect on the treated be estimated.

pomeans specifies that the potential-outcome means for each treatment level be estimated.

SE/Robust

vce (*vcetype*) specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (*robust*), that allow for intragroup correlation (*cluster* *clustvar*), and that use bootstrap or jackknife methods (*bootstrap*, *jackknife*); see [R] **vce_option**.

Reporting

level(#); see [R] **Estimation options**.

aequations specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

display_options: *nocl*, *nopvalues*, *noomitted*, *vsquish*, *noemptycells*, *baselevels*, *allbaselevels*, *nofvlabel*, *fvwrap*(#), *fvwrapon*(*style*), *cformat*(%*fmt*), *pformat*(%*fmt*), *sformat*(%*fmt*), and *nolstretch*; see [R] **Estimation options**.

Maximization

maximize_options: *iterate*(#), *[no]log*, and *from*(*init_specs*); see [R] **Maximize**. These options are seldom used.

init_specs is one of

matname [, *skip copy*]

[, # ...], *copy*

Advanced

pstolerance(#) specifies the tolerance used to check the overlap assumption. The default value is *pstolerance*(1e-5). *teffects* will exit with an error if an observation has an estimated propensity score smaller than that specified by *pstolerance*().

osample(*newvar*) specifies that indicator variable *newvar* be created to identify observations that violate the overlap assumption.

`control(#|label)` specifies the level of *tvar* that is the control. The default is the first treatment level.

You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(#|label)` specifies the level of *tvar* that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `teffects ipw` but is not shown in the dialog box:

`coeflegend`; see [\[R\] Estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Video example](#)

Overview

IPW estimators use estimated probability weights to correct for the missing-data problem arising from the fact that each subject is observed in only one of the potential outcomes. IPW estimators use a two-step approach to estimating treatment effects:

1. They estimate the parameters of the treatment model and compute the estimated inverse-probability weights.
2. They use the estimated inverse-probability weights to compute weighted averages of the outcomes for each treatment level. The contrasts of these weighted averages provide the estimates of the ATEs. Using this weighting scheme corrects for the missing potential outcomes.

These steps produce consistent estimates of the effect parameters because the treatment is assumed to be independent of the potential outcomes after conditioning on the covariates. The overlap assumption ensures that predicted inverse-probability weights do not get too large. In fact, `teffects ipw` uses an estimation technique that implements both steps at once so that we do not need to correct the standard errors in the second step to reflect the uncertainty associated with the predicted treatment probabilities.

We will illustrate the use of `teffects ipw` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by [Cattaneo \(2010\)](#). This dataset also contains information about each mother's age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), and whether this baby was the mother's first birth (`fbaby`).

► Example 1: Estimating the ATE

We begin by using `teffects ipw` to estimate the average treatment effect of smoking on birthweight. We will use a probit model to predict treatment status, using `prenatal1`, `mmarried`, `mage`, the square of `mage`, and `fbaby` as explanatory variables:

```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)

. teffects ipw (bweight) (mbsmoke mmarrried c.mage##c.mage fbaby medu, probit)

Iteration 0:  EE criterion = 4.622e-21
Iteration 1:  EE criterion = 8.795e-26

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (Smoker vs Nonsmoker)	-230.6886	25.81524	-8.94	0.000	-281.2856	-180.0917
POmean						
mbsmoke Nonsmoker	3403.463	9.571369	355.59	0.000	3384.703	3422.222

The average birthweight if all mothers were to smoke would be 231 grams less than the average of 3,403 grams that would occur if none of the mothers had smoked.

◀

Sometimes, we are mainly concerned about those subjects that did in fact receive treatment, and we want to know how much the outcome changes as a result of treatment for that subpopulation. The ATET provides us with the answer. Moreover, the ATET can be estimated using weaker assumptions than are required to estimate the ATE; see [\[CAUSAL\] teffects intro advanced](#).

➤ Example 2: Estimating the ATET

```
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit),
> atet

Iteration 0:  EE criterion = 4.633e-21
Iteration 1:  EE criterion = 7.494e-27

Treatment-effects estimation          Number of obs      =      4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

bweight	Robust					
	Coefficient	std. err.	z	P> z	[95% conf. interval]	
ATET						
mbsmoke						
(Smoker						
vs						
Nonsmoker)	-225.1773	23.66458	-9.52	0.000	-271.559	-178.7955
POmean						
mbsmoke						
Nonsmoker	3362.837	14.20149	236.79	0.000	3335.003	3390.671

The average birthweight is 225 grams less when all the mothers who smoke do so than the average of 3,363 grams that would have occurred if none of these mothers had smoked.



We often express statistics as percentages to alleviate scaling issues and aid interpretation. In the present context, we may wish to express an ATE as a percentage of the untreated POM to gain a more intuitive measure of the effect of treatment.

➤ Example 3: Reporting the ATE as a percentage

Here we use the same model as in example 1, but we report the ATE as a percentage of the mean birthweight that would occur if no mothers smoke. First, we use `teffects ipw` to fit the model. We use the `coeflegend` option so that `teffects ipw` reports the names of the parameters. Then we use `nlcom` to obtain the statistic we want along with its delta-method-based standard error. We type

```
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit),
> coeflegend

Iteration 0:  EE criterion = 4.622e-21
Iteration 1:  EE criterion = 8.795e-26

Treatment-effects estimation          Number of obs      =      4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

bweight	Coefficient	Legend
ATE		
mbsmoke		
(Smoker		
vs		
Nonsmoker)	-230.6886	_b[ATE:r1vs0.mbsmoke]
P0mean		
mbsmoke		
Nonsmoker	3403.463	_b[P0mean:0.mbsmoke]

```
. nlcom _b[ATE:r1vs0.mbsmoke] / _b[P0mean:0.mbsmoke]
      _nl_1: _b[ATE:r1vs0.mbsmoke] / _b[P0mean:0.mbsmoke]
```

bweight	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
_nl_1	-.0677806	.0075169	-9.02	0.000	-.0825133	-.0530478

The average birthweight falls by an estimated 6.8% when every mother smokes relative to the case when no mothers smoke. We also obtain a 95% confidence interval of a 5.3% to 8.3% reduction.



Video example

Treatment effects: Inverse-probability weighting

Stored results

`teffects ipw` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>teffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>ipw</code>
<code>e(tmodel)</code>	logit, probit, or hetprobit
<code>e(stat)</code>	statistic estimated, <code>ate</code> , <code>atet</code> , or <code>pomeans</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vce</i> type specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by margins
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

`teffects ipw` implements a [smooth treatment-effects estimator](#). All smooth treatment-effects estimators are documented in [Methods and formulas](#) of [\[CAUSAL\] teffects aipw](#).

References

- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cerulli, G. 2014. `treatrew`: A user-written command for estimating average treatment effects by reweighting on the propensity score. *Stata Journal* 14: 541–561.
- Drukker, D. M. 2014. Using `gmm` to solve two-step estimation problems. *The Stata Blog: Not Elsewhere Classified*. <https://blog.stata.com/2014/12/08/using-gmm-to-solve-two-step-estimation-problems/>.
- Huber, C. 2015. Introduction to treatment effects in Stata: Part 1. *The Stata Blog: Not Elsewhere Classified*. <https://blog.stata.com/2015/07/07/introduction-to-treatment-effects-in-stata-part-1/>.

Also see

[CAUSAL] [teffects postestimation](#) — Postestimation tools for `teffects`

[CAUSAL] [teffects](#) — Treatment-effects estimation for observational data

[U] [20 Estimation and postestimation commands](#)

Description
Options
References

Quick start
Remarks and examples
Also see

Menu
Stored results

Syntax
Methods and formulas

Description

`teffects ipwra` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational data by inverse-probability-weighted regression adjustment (IPWRA). IPWRA estimators use weighted regression coefficients to compute averages of treatment-level predicted outcomes, where the weights are the estimated inverse probabilities of treatment. The contrasts of these averages estimate the treatment effects. IPWRA estimators have the double-robust property. `teffects ipwra` accepts a continuous, binary, count, fractional, or nonnegative outcome and allows a multivalued treatment.

See [\[CAUSAL\] teffects intro](#) or [\[CAUSAL\] teffects intro advanced](#) for more information about estimating treatment effects from observational data.

Quick start

ATE of binary treatment `treat2` estimated by IPWRA using a linear model for outcome `y1` on `x1` and `x2` and a logistic model for `treat2` on `x1` and `w`

```
teffects ipwra (y1 x1 x2) (treat2 x1 w)
```

Same as above, but estimate the ATET

```
teffects ipwra (y1 x1 x2) (treat2 x1 w), atet
```

Probit model for binary outcome `y3`

```
teffects ipwra (y3 x1 x2, probit) (treat2 x1 w)
```

Same as above, but use a heteroskedastic probit model for `y3` and a probit model for `treat2`

```
teffects ipwra (y3 x1 x2, hetprobit(x1 x2)) (treat2 x1 w, probit)
```

Same as above, but use a fractional heteroskedastic probit model for `y4` and a probit model for `treat2`

```
teffects ipwra (y4 x1 x2, fhetprobit(x1 x2)) (treat2 x1 w, probit)
```

ATE for each level of a three-valued treatment `treat3`

```
teffects ipwra (y1 x1 x2) (treat3 x1 w)
```

Same as above, and specify that `treat3 = 3` is the control level using the value label “MyControl” for 3

```
teffects ipwra (y1 x1 x2) (treat3 x1 w), control(MyControl)
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Regression adjustment with IPW

Statistics > Causal inference/treatment effects > Binary outcomes > Regression adjustment with IPW

Statistics > Causal inference/treatment effects > Count outcomes > Regression adjustment with IPW

Statistics > Causal inference/treatment effects > Fractional outcomes > Regression adjustment with IPW

Statistics > Causal inference/treatment effects > Nonnegative outcomes > Regression adjustment with IPW

Syntax

```
teffects ipwra (ovar omvarlist [ , omodel noconstant ])
               (tvar tmvarlist [ , tmodel noconstant ]) [if] [in] [weight]
               [ , stat options ]
```

ovar is a binary, count, continuous, fractional, or nonnegative outcome of interest.

omvarlist specifies the covariates in the outcome model.

tvar must contain integer values representing the treatment levels.

tmvarlist specifies the covariates in the treatment-assignment model.

<i>omodel</i>	Description
---------------	-------------

Model

<code>linear</code>	linear outcome model; the default
<code>logit</code>	logistic outcome model
<code>probit</code>	probit outcome model
<code>hetprobit</code> (<i>varlist</i>)	heteroskedastic probit outcome model
<code>poisson</code>	exponential outcome model
<code>flogit</code>	fractional logistic outcome model
<code>fprobit</code>	fractional probit outcome model
<code>fhetprobit</code> (<i>varlist</i>)	fractional heteroskedastic probit outcome model

omodel specifies the model for the outcome variable.

<i>tmodel</i>	Description
---------------	-------------

Model

<code>logit</code>	logistic treatment model; the default
<code>probit</code>	probit treatment model
<code>hetprobit</code> (<i>varlist</i>)	heteroskedastic probit treatment model

tmodel specifies the model for the treatment variable.

For multivalued treatments, only `logit` is available and multinomial logit is used.

<i>stat</i>	Description
Stat	
<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated
<code>pomeans</code>	estimate potential-outcome means
<i>options</i>	Description
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> , <code>cluster</code> <i>clustvar</i> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>aequations</code>	display auxiliary-equation results
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
Advanced	
<code>pstolerance(#)</code>	set tolerance for overlap assumption
<code>osample(<i>newvar</i>)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control(# <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(# <i>label</i>)</code>	specify the level of <i>tvar</i> that is the treatment
<code>coeflegend</code>	display legend instead of statistics

omvarlist and *tmvarlist* may contain factor variables; see [U] 11.4.3 **Factor variables**.

`bayesboot`, `bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed; see [U] 11.1.10 **Prefix commands**.

Weights are not allowed with the `bootstrap` prefix; see [R] **bootstrap**.

`fweights`, `iweights`, and `pweights` are allowed; see [U] 11.1.6 **weight**.

`coeflegend` does not appear in the dialog box.

See [U] 20 **Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

`noconstant`; see [R] **Estimation options**.

Stat

stat is one of three statistics: `ate`, `atet`, or `pomeans`. `ate` is the default.

`ate` specifies that the average treatment effect be estimated.

`atet` specifies that the average treatment effect on the treated be estimated.

`pomeans` specifies that the potential-outcome means for each treatment level be estimated.

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

Reporting

`level(#)`; see [R] [Estimation options](#).

`aequations` specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

`display_options`: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

Maximization

`maximize_options`: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

`init_specs` is one of

`matname [ , skip copy]`

`# [ , #...], copy`

Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `teffects` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption.

`control(#|label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(#|label)` specifies the level of `tvar` that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `teffects ipwra` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Video example](#)

Overview

IPWRA estimators use probability weights to obtain outcome-regression parameters that account for the missing-data problem arising from the fact that each subject is observed in only one of the potential outcomes. The adjusted outcome-regression parameters are used to compute averages of treatment-level predicted outcomes. The contrasts of these averages provide estimates of the treatment effects.

IPWRA estimators use a model to predict treatment status, and they use another model to predict outcomes. Because IPWRA estimators have the double-robust property, only one of the two models must be correctly specified for the IPWRA estimator to be consistent.

IPWRA estimators use a three-step approach to estimating treatment effects:

1. They estimate the parameters of the treatment model and compute inverse-probability weights.
2. Using the estimated inverse-probability weights, they fit weighted regression models of the outcome for each treatment level and obtain the treatment-specific predicted outcomes for each subject.
3. They compute the means of the treatment-specific predicted outcomes. The contrasts of these averages provide the estimates of the ATEs. By restricting the computations of the means to the subset of treated subjects, we can obtain the ATETs.

These steps produce consistent estimates of the effect parameters because the treatment is assumed to be independent of the potential outcomes after conditioning on the covariates. The overlap assumption ensures that predicted inverse-probability weights do not get too large. The standard errors reported by `teffects ipwra` correct for the three-step process. See [\[CAUSAL\] teffects intro](#) or [\[CAUSAL\] teffects intro advanced](#) for more information about this estimator.

We will illustrate the use of `teffects ipwra` by using data from a study of the effect of a mother's smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by [Cattaneo \(2010\)](#). This dataset also contains information about each mother's age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), and whether this baby was the mother's first birth (`fbaby`).

► Example 1: Estimating the ATE

We begin by using `teffects ipwra` to estimate the average treatment effect of smoking on birthweight. We will use a probit model to predict treatment status as a function of `mmarried`, `mage`, and `fbaby`; to maximize the predictive power of this model, we use factor-variable notation to incorporate quadratic effects of the mother's age, the only continuous covariate in our model. We will use linear regression (the default) to model birthweight, using `prenatal1`, `mmarried`, `mage`, and `fbaby` as explanatory variables. We type

```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipwra (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)

Iteration 0:  EE criterion = 9.261e-21
Iteration 1:  EE criterion = 8.863e-26

Treatment-effects estimation          Number of obs      =      4,642
Estimator      : IPW regression adjustment
Outcome model  : linear
Treatment model: probit
```

bweight	Robust					
	Coefficient	std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (Smoker vs Nonsmoker)	-229.9671	26.62668	-8.64	0.000	-282.1544	-177.7798
POMean						
mbsmoke Nonsmoker	3403.336	9.57126	355.58	0.000	3384.576	3422.095

The average birthweight if all mothers were to smoke would be 230 grams less than the average of 3,403 grams that would occur if none of the mothers had smoked.

◀

By default, `teffects ipwra` displays the ATE and untreated POM. We can specify the `pomeans` option to display both the treated and untreated POMs, and we can use the `aequations` option to display the regression model coefficients used to predict the POMs as well as the coefficients from the model used to predict treatment.

► Example 2: Displaying the POMs and equations

```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipwra (bweight prenatal1 mmarried mage fbaby)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, probit), pomeans aequations

Iteration 0:  EE criterion = 9.261e-21
Iteration 1:  EE criterion = 8.428e-26

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : IPW regression adjustment
Outcome model  : linear
Treatment model: probit
```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
POMeans						
mbsmoke						
Nonsmoker	3403.336	9.57126	355.58	0.000	3384.576	3422.095
Smoker	3173.369	24.86997	127.60	0.000	3124.624	3222.113
OME0						
prenatal1	67.98549	28.78428	2.36	0.018	11.56933	124.4017
mmarried	155.5893	26.46903	5.88	0.000	103.711	207.4677
mage	2.893051	2.134788	1.36	0.175	-1.291056	7.077158
fbaby	-71.9215	20.39317	-3.53	0.000	-111.8914	-31.95162
_cons	3194.808	55.04911	58.04	0.000	3086.913	3302.702
OME1						
prenatal1	34.76923	43.18534	0.81	0.421	-49.87248	119.4109
mmarried	124.0941	40.29775	3.08	0.002	45.11193	203.0762
mage	-5.068833	5.954425	-0.85	0.395	-16.73929	6.601626
fbaby	39.89692	56.82072	0.70	0.483	-71.46966	151.2635
_cons	3175.551	153.8312	20.64	0.000	2874.047	3477.054
TME1						
mmarried	-.6484821	.0554173	-11.70	0.000	-.757098	-.5398663
mage	.1744327	.0363718	4.80	0.000	.1031452	.2457202
c.mage#						
c.mage	-.0032559	.0006678	-4.88	0.000	-.0045647	-.0019471
fbaby	-.2175962	.0495604	-4.39	0.000	-.3147328	-.1204595
medu	-.0863631	.0100148	-8.62	0.000	-.1059917	-.0667345
_cons	-1.558255	.4639691	-3.36	0.001	-2.467618	-.6488926

◀

As is well known, the standard probit model assumes that the error terms in the latent-utility framework are homoskedastic; the model is not robust to departures from this assumption. An alternative is to use the heteroskedastic probit model, which explicitly models the error variance as a function of a set of variables.

► Example 3: Heteroskedastic probit treatment model

Here we use the variables as before, but we use a heteroskedastic probit model to predict treatment status, modeling the heteroskedasticity as a quadratic function of the mother's age:

```
. teffects ipwra (bweight prenatal1 mmarried fbaby c.mage)
> (mbsmoke mmarried c.mage##c.mage fbaby medu, hetprobit(c.mage##c.mage)),
> aequations

Iteration 0: EE criterion = 1.776e-08
Iteration 1: EE criterion = 6.205e-12

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : IPW regression adjustment
Outcome model  : linear
Treatment model: heteroskedastic probit
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (Smoker vs Nonsmoker)	-229.6322	26.33452	-8.72	0.000	-281.2469	-178.0175
POMean						
mbsmoke Nonsmoker	3403.74	9.545798	356.57	0.000	3385.03	3422.449
OME0						
prenatal1	64.95127	28.6216	2.27	0.023	8.853969	121.0486
mmarried	154.2297	26.45867	5.83	0.000	102.3717	206.0878
fbaby	-71.61131	20.33774	-3.52	0.000	-111.4725	-31.75006
mage	3.010148	2.133812	1.41	0.158	-1.172047	7.192343
_cons	3195.355	55.05451	58.04	0.000	3087.45	3303.26
OME1						
prenatal1	38.55272	43.57024	0.88	0.376	-46.84337	123.9488
mmarried	126.3377	40.73979	3.10	0.002	46.48922	206.1863
fbaby	45.43542	56.44831	0.80	0.421	-65.20122	156.0721
mage	-6.06991	5.952512	-1.02	0.308	-17.73662	5.596799
_cons	3195.795	152.3979	20.97	0.000	2897.101	3494.49
TME1						
mmarried	-.0295529	.0238889	-1.24	0.216	-.0763742	.0172684
mage	.0157896	.010546	1.50	0.134	-.0048803	.0364594
c.mage# c.mage	-.0002837	.00019	-1.49	0.135	-.0006562	.0000887
fbaby	-.0093308	.0079984	-1.17	0.243	-.0250074	.0063458
medu	-.0036774	.0030309	-1.21	0.225	-.0096178	.0022631
_cons	-.182223	.1180298	-1.54	0.123	-.4135572	.0491112
TME1_lnsigma						
mage	-.2211477	.0631342	-3.50	0.000	-.3448885	-.0974069
c.mage# c.mage	.0037613	.0012435	3.02	0.002	.0013241	.0061984

The estimated ATE and base-level POM are essentially the same as those produced by the model that used a homoskedastic probit.

Video example

Treatment effects: Inverse-probability-weighted regression adjustment

Stored results

`teffects ipwra` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>teffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>ipwra</code>
<code>e(tmodel)</code>	<code>logit</code> , <code>probit</code> , or <code>hetprobit</code>
<code>e(omodel)</code>	<code>linear</code> , <code>logit</code> , <code>probit</code> , <code>hetprobit</code> , <code>poisson</code> , <code>flogit</code> , <code>fprobit</code> , or <code>fhetsprobit</code>
<code>e(stat)</code>	statistic estimated, <code>ate</code> , <code>atet</code> , or <code>pomeans</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<code>vcetype</code> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b</code> <code>V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

`teffects ipwra` implements a [smooth treatment-effects estimator](#). All smooth treatment-effects estimators are documented in [Methods and formulas](#) of [\[CAUSAL\] teffects aipw](#).

References

- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Huber, C. 2015. Introduction to treatment effects in Stata: Part 1. *The Stata Blog: Not Elsewhere Classified*. <https://blog.stata.com/2015/07/07/introduction-to-treatment-effects-in-stata-part-1/>.

Also see

- [\[CAUSAL\] teffects postestimation](#) — Postestimation tools for `teffects`
- [\[CAUSAL\] teffects](#) — Treatment-effects estimation for observational data
- [\[CAUSAL\] teffects aipw](#) — Augmented inverse-probability weighting
- [\[U\] 20 Estimation and postestimation commands](#)

Description

This entry discusses the use of `teffects` when the treatment is multivalued. This entry presumes you are already familiar with the potential-outcome framework and the use of the `teffects` commands with binary treatments. See [\[CAUSAL\] teffects intro](#) or [\[CAUSAL\] teffects intro advanced](#) for more information.

Remarks and examples

Remarks are presented under the following headings:

[Introduction](#)

[Parameters and notation](#)

[Illustrating multivalued treatments](#)

[Examples](#)

Introduction

When the treatment is binary, each subject could either receive the treatment or not receive the treatment. In contrast, multivalued treatments refer to cases in which each subject could receive one of several different treatments or else not receive treatment at all. For example, in testing the efficacy of a drug, a patient could receive a 10 milligram (mg) dose, a 20 mg dose, a 30 mg dose, or no dose at all. We first want to be able to compare a patient receiving the 10 mg dose with a patient receiving no dose, a patient receiving the 20 mg dose with a patient receiving no dose, and a patient receiving the 30 mg dose with a patient receiving no dose. Once we can make those comparisons, we can then, for example, compare the efficacy of a 30 mg dose with that of a 20 mg dose or a 10 mg dose.

To highlight an example in economics, we consider an unemployed person who could participate in a comprehensive skills training program, attend a one-day workshop that helps job seekers write their resumés, or choose not to participate in either. We want to know how effective each of those programs is relative to not participating; once we know that, we can then compare the effectiveness of the comprehensive program with that of the one-day program.

Multivalued treatments increase the number of parameters that must be estimated and complicate the notation. Fortunately, however, using the `teffects` commands is not much more difficult with multivalued treatments than with binary treatments.

You can use `teffects ra`, `teffects ipw`, `teffects ipwra`, and `teffects aipw` to estimate multivalued treatment effects. However, the theory developed in [Abadie and Imbens \(2006, 2012\)](#) has not been extended to handle multivalued treatments, so you cannot use `teffects nnmatch` or `teffects psmatch` in these cases.

[Cattaneo \(2010\)](#), [Imbens \(2000\)](#), and [Wooldridge \(2010, sec. 21.6.3\)](#) discuss aspects of treatment-effect estimation with multivalued treatments.

Parameters and notation

We denote the potential outcome that subject i would obtain if given treatment-level t as y_{ti} , where y_{ti} is the realization of the random variable y_t . Throughout this entry, i subscripts denote realizations of the corresponding unsubscripted random variables. We again let y_0 denote the potential outcome of a subject who did not receive any treatment. To handle the case of multivalued treatments, we extend the definition of the unobservable, individual-level treatment effects to be $y_t - y_0$ for $t \in \{1, \dots, q\}$.

As in the binary-valued case, we again focus on three parameters of interest: the average treatment effect (ATE), the potential-outcome mean (POM), and the average treatment effect on the treated (ATET).

ATE The ATE is the average effect of giving each individual treatment t instead of treatment 0:

$$\text{ATE}_t = E(y_t - y_0)$$

POM The POM for each treatment level is an average of each potential outcome:

$$\text{POM}_t = E(y_t)$$

ATET The ATET is the average effect among those subjects that receive treatment level $\tilde{t}$ of giving each subject treatment $\tilde{t}$ instead of treatment 0:

$$\text{ATET}_{\tilde{t}, \tilde{t}} = E\{(y_{\tilde{t}} - y_0) | t = \tilde{t}\}$$

The extra notation required to define the ATET in this case indicates the difficulties surrounding this parameter.

Defining the ATET in the multivalued treatment case requires three different treatment levels: $\tilde{t}$ defines the treatment level of the treated potential outcome; 0 is the treatment level of the control potential outcome; and $t = \tilde{t}$ restricts the expectation to include only those individuals who actually receive treatment level $\tilde{t}$.

Illustrating multivalued treatments

To illustrate the concept of a potential outcome and the parameters we would like to estimate, we consider the following table:

y	t	y_0	y_1	y_2
−0.50	0	−0.50	1.06	1.93
2.42	1	2.13	2.42	2.43
3.15	2	1.26	2.57	3.15
−0.39	0	−0.39	−0.18	0.52
2.22	2	−0.24	−0.01	2.22

We observe the outcome y as well as the treatment indicator t . There are three levels of treatment: 0, 1, or 2. Ideally, we would observe y_0 , y_1 , and y_2 , but in fact all we have is y . In the first row, the subject received treatment level 0, so $y = y_0$ for that subject. In the last row, the subject received treatment 2, so $y = y_2$. We reiterate that we do not actually observe y_0 , y_1 , or y_2 .

If we did have data on y_0 , y_1 , and y_2 , then we could define subject-level treatment variables $te_1 = y_1 - y_0$ and $te_2 = y_2 - y_0$. Here we would be following the convention of taking treatment level 0 to be the control level. The following table adds these two variables:

y	t	y_0	y_1	y_2	te_1	te_2
−0.50	0	−0.50	1.06	1.93	1.56	2.43
2.42	1	2.13	2.42	2.43	0.29	0.30
3.15	2	1.26	2.57	3.15	1.31	1.89
−0.39	0	−0.39	−0.18	0.52	0.21	0.91
2.22	2	−0.24	−0.01	2.22	0.23	2.46

Once we have te_1 and te_2 , obtaining the ATEs is straightforward. The ATE of going from treatment 0 to treatment 1 is simply the mean of the five entries in the column labeled te_1 , which here works out to 0.72. Going from treatment level 0 to treatment level 1 causes the outcome to increase an average of 0.72. Similarly, the ATE of going from treatment 0 to treatment 2 is the mean of the entries in the column labeled te_2 , which is 1.60. Exposing all subjects to treatment level 2 would cause the outcome to rise by an average of 1.60 relative to the outcome obtained by exposing them to treatment level 0.

The ATET is the average difference in the potential outcomes among those that get a particular treatment level. To compute this, we must specify two treatment levels: the actual treatment level the subjects we are interested in received as well as the treatment level we want to compare them with. For example, suppose we are interested in the ATET of going from treatment 0 to treatment 1 for those who received treatment 0. This ATET is the average of te_1 for those subjects for which $t = 0$. Here that ATET is just $(1.56 + 0.21)/2 \approx 0.89$. If we exposed the subjects who received treatment 0 to treatment 1 instead, the outcome would increase an average of 0.89.

The ATET of going from treatment 0 to treatment 2 for those subjects who received treatment 2 is the mean of te_2 for those subjects for which $t = 2$, which is $(1.89 + 2.46)/2 \approx 2.18$. Receiving treatment 2 increased the outcome of those who received treatment 2 by an average of 2.18 relative to receiving the control.

Examples

In the remainder of this entry, we provide several examples demonstrating how to estimate multivalued treatments using `teffects`.

► Example 1: Potential outcomes with four treatment levels

`bdsianesi5.dta` contains an extract of data from [Blundell, Dearden, and Sianesi \(2005\)](#). In this dataset on individuals in the United Kingdom, `wages` records hourly wages in pounds; `ed` records the highest educational degree obtained; `paed` records the highest educational level obtained by each individual's father; `math7` records a score obtained on a standardized math test when the individual was seven; `read7` records a score obtained on a standardized reading test when the individual was seven; and `london` and `eastern` are indicators for whether an individual lives in the expensive area of London or the east. We want to know how the level of education obtained affects a person's wage.

We begin by using `mean` to report the estimated means of wages over the four education levels. The value labels on `mean` are coded as `none` for no degree, `0` for an O-level degree, `A` for an A-level degree, or `H` for a higher-education degree.

```
. use https://www.stata-press.com/data/r19/bdsianesi5
(Excerpt from Blundell, Dearden, & Sianesi (2005) JRSSA 168: 473)
. mean wage, over(ed)
Mean estimation                                     Number of obs = 1,693
```

	Mean	Std. err.	[95% conf. interval]	
c.wage@ed				
none	6.057816	.154332	5.755114	6.360518
O	7.501648	.1807359	7.147158	7.856137
A	8.220637	.1540359	7.918516	8.522758
H	10.87703	.2257888	10.43417	11.31988

The output reveals that the estimated mean wage increases as the education level goes from no degree to an O-level degree, to an A-level degree, and to a higher-education degree, as we would expect. Once we control for other characteristics of each individual, do we still observe a positive effect of education on wage?

We use `teffects ra` (see [\[CAUSAL\] teffects ra](#)) to estimate the ATEs of the different education levels by regression adjustment (RA), controlling for each person's location, math score, and father's education level:

```
. teffects ra (wage london eastern paed math7, poisson) (ed)
Iteration 0: EE criterion = 1.865e-18
Iteration 1: EE criterion = 4.077e-30
Treatment-effects estimation               Number of obs   =       1,693
Estimator      : regression adjustment
Outcome model  : Poisson
Treatment model: none
```

wage	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
ed						
(O vs none)	1.181543	.3520371	3.36	0.001	.4915626	1.871523
(A vs none)	1.743079	.3282152	5.31	0.000	1.099789	2.386369
(H vs none)	3.972829	.3840024	10.35	0.000	3.220199	4.72546
POMean						
ed						
none	6.525873	.2931933	22.26	0.000	5.951224	7.100521

Because wages are necessarily positive, we used the `poisson` option inside the outcome-model specification. The estimated POM of the control level of no degree is 6.53 pounds per hour. The estimated ATE of going from no degree to an O-level degree is 1.18 pounds per hour; the estimated ATE of going from no degree to an A-level degree is 1.74 pounds per hour; and the estimated ATE of going from no degree to a higher-education degree is 3.97 pounds per hour. All of these effects are highly significant.

For comparison purposes, we also use `teffects aipw` (see [CAUSAL] `teffects aipw`). We use the same outcome model as before. We use a multinomial logit model to predict education level, using math and reading scores and both the father's and the mother's educational attainment levels as predictors:

```
. teffects aipw (wage london eastern paed math7, poisson)
> (ed math7 read7 maed paed)

Iteration 0:  EE criterion = 1.877e-18
Iteration 1:  EE criterion = 1.029e-30

Treatment-effects estimation          Number of obs      =          1,693
Estimator      : augmented IPW
Outcome model  : Poisson by ML
Treatment model: (multinomial) logit
```

wage	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
ed						
(0 vs none)	1.748197	.3911167	4.47	0.000	.9816221	2.514771
(A vs none)	2.363228	.3741584	6.32	0.000	1.629891	3.096565
(H vs none)	4.359777	.4133059	10.55	0.000	3.549712	5.169842
POMean						
ed						
none	5.946184	.3391531	17.53	0.000	5.281456	6.610912

The results indicate slightly higher treatment effects relative to those indicated by `teffects ra`. That is largely because the AIPW estimator predicts a lower no-higher-education POM than the RA estimator.



► Example 2: Expressing ATEs as percentages

As in the binary-treatment case, expressing the ATEs as percentages of the POM for the control level often aids interpretation. Here we first use the replay facility of `teffects aipw` along with the `coeflegend` option to see how the parameters are named.

```
. teffects, coeflegend

Treatment-effects estimation          Number of obs      =          1,693
Estimator      : augmented IPW
Outcome model  : Poisson by ML
Treatment model: (multinomial) logit
```

wage	Coefficient	Legend
ATE		
ed		
(0 vs none)	1.748197	_b[ATE:r1vs0.ed]
(A vs none)	2.363228	_b[ATE:r2vs0.ed]
(H vs none)	4.359777	_b[ATE:r3vs0.ed]
POMean		
ed		
none	5.946184	_b[POMean:0.ed]

Now that we know the names, we can use `nlcom` to obtain the ATEs relative to the base-level POM:

```
. nlcom (_b[ATE:r1vs0.ed] / _b[POmean:0.ed])
>      (_b[ATE:r2vs0.ed] / _b[POmean:0.ed])
>      (_b[ATE:r3vs0.ed] / _b[POmean:0.ed])

      _nl_1:  _b[ATE:r1vs0.ed] / _b[POmean:0.ed]
      _nl_2:  _b[ATE:r2vs0.ed] / _b[POmean:0.ed]
      _nl_3:  _b[ATE:r3vs0.ed] / _b[POmean:0.ed]
```

wage	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
_nl_1	.2940031	.0808156	3.64	0.000	.1356075	.4523988
_nl_2	.3974361	.0840545	4.73	0.000	.2326923	.5621799
_nl_3	.7332059	.1068848	6.86	0.000	.5237156	.9426962

Wages are 29% higher when everyone receives an O-level degree than if no one receives a degree. Wages are 40% higher when everyone receives an A-level degree than if no one receives a degree. Wages are 73% higher when everyone receives an H-level degree than if no one receives a degree.

Although impressive, these changes are not presented in the way that is most commonly discussed. (There is a large amount of literature on the treatment effect of getting a higher-education degree.) In particular, we might rather want to know the percentage changes in wages relative to a person with an A-level degree. Next we estimate the ATEs treating an A-level degree as the control level; to do that, we use the `control()` option. We also specify `coeflegend` again because we are more interested in how the parameters are named rather than in their standard errors at this point:

```
. teffects aiwp (wage london eastern paed math7, poisson)
> (ed math7 read7 maed paed), control(A) coeflegend

Iteration 0:  EE criterion = 1.870e-18
Iteration 1:  EE criterion = 1.960e-30

Treatment-effects estimation          Number of obs      =          1,693
Estimator      : augmented IPW
Outcome model  : Poisson by ML
Treatment model: (multinomial) logit
```

wage	Coefficient	Legend
ATE		
ed		
(none vs A)	-2.363228	_b[ATE:r0vs2.ed]
(O vs A)	-.6150312	_b[ATE:r1vs2.ed]
(H vs A)	1.996549	_b[ATE:r3vs2.ed]
POmean		
ed		
A	8.309412	_b[POmean:2.ed]

Now we use `nlcom` to obtain the ATE of obtaining a higher-education degree as a percentage of the expected A-level wage:

```
. nlcom _b[ATE:r3vs2.ed] / _b[P0mean:2.ed], noheader
```

wage	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
_nl_1	.2402756	.0355404	6.76	0.000	.1706177	.3099335

The average wage increases by 24% when everyone receives an H-level degree relative to when everyone receives an A-level degree.



► Example 3: Obtaining ATETs

In the previous example, we showed that on average, a higher-education degree increases a person's wage by 24% relative to someone with only an A-level degree. Sometimes, though, we would rather know how much the higher-education degree increases wages among the people who actually have a higher-education degree. To answer that question, we want to examine the ATET rather than the ATE.

Here we use the IPWRA estimator to obtain our answer. We specify the `control(A)` option so that an A-level education is treated as the basis for comparisons. We specify the `atet` option to obtain ATETs rather than ATEs, and we specify the `tlevel(H)` option to indicate that we want the ATETs to be calculated for the subset of people who actually receive higher-education degrees.

```
. teffects ipwra (wage london eastern paed math7, poisson)
> (ed math7 read7 maed paed), atet control(A) tlevel(H)

Iteration 0: EE criterion = 2.731e-18
Iteration 1: EE criterion = 6.253e-31

Treatment-effects estimation          Number of obs      =          1,693
Estimator       : IPW regression adjustment
Outcome model   : Poisson
Treatment model : (multinomial) logit
```

wage	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET						
ed						
(none vs A)	-2.87423	.361093	-7.96	0.000	-3.58196	-2.166501
(O vs A)	-.8246604	.3609131	-2.28	0.022	-1.532037	-.1172837
(H vs A)	1.866757	.3277701	5.70	0.000	1.224339	2.509174
P0mean						
ed						
A	9.010271	.2503971	35.98	0.000	8.519501	9.50104

The point estimates are similar to the ATEs we obtained above, suggesting that the means of the covariates among those with a higher-education degree are similar to the means for the entire population.

In output not shown to save space, we replayed the previous results with the `coeflegend` option to determine how the parameters are named. Armed with that information, we call `nlcom`:

```
. nlcom _b[ATET:r3vs2.ed] / _b[POmean:2.ed], noheader
```

wage	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
_nl_1	.207181	.0407528	5.08	0.000	.127307	.287055

Our estimate of the percentage increase is now noticeably smaller once we restrict ourselves to only those people who actually received a higher-education degree. However, because of the width of the confidence intervals, there is no evidence to suggest that the difference between the estimates is statistically significant.

◀

► Example 4: ATEs comparing adjacent treatments

In the [first example](#), we obtained the three ATEs, and they were all expressed relative to the base level of no degree. Now we show how we can express the gains to an O-level degree relative to no degree, the gains to an A-level degree relative to an O-level degree, and the gains to a higher-education degree relative to an A-level degree.

First, we use an AIPW estimator to obtain all the POMs for our example dataset:

```
. teffects aipw (wage london eastern paed math7, poisson)
> (ed math7 read7 maed paed), pom
Iteration 0: EE criterion = 1.877e-18
Iteration 1: EE criterion = 1.542e-30

Treatment-effects estimation          Number of obs      =      1,693
Estimator       : augmented IPW
Outcome model   : Poisson by ML
Treatment model : (multinomial) logit
```

wage	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
POmeans						
ed						
none	5.946184	.3391531	17.53	0.000	5.281456	6.610912
O	7.694381	.1915192	40.18	0.000	7.31901	8.069752
A	8.309412	.1563348	53.15	0.000	8.003001	8.615823
H	10.30596	.2285837	45.09	0.000	9.857945	10.75398

ATEs are contrasts of POMs, and here we show how to use contrast to obtain the estimated ATEs:

```
. contrast r.ed, nowald
warning: cannot perform check for estimable functions.
Contrasts of marginal linear predictions
Margins: asbalanced
```

	Contrast	Std. err.	[95% conf. interval]	
POMeans				
ed				
(O vs none)	1.748197	.3911167	.9816221	2.514771
(A vs none)	2.363228	.3741584	1.629891	3.096565
(H vs none)	4.359777	.4133059	3.549712	5.169842

These estimated ATEs match those we obtained in [example 2](#).

Now that we know how to use contrast to obtain the ATEs based on the POMs, we can take advantage of contrast’s ability to obtain “reverse adjacent” contrasts, which compare each level with the previous level. We use the `ar.` operator with contrast to accomplish this:

```
. contrast ar.ed, nowald
warning: cannot perform check for estimable functions.
Contrasts of marginal linear predictions
Margins: asbalanced
```

	Contrast	Std. err.	[95% conf. interval]	
POMeans				
ed				
(O vs none)	1.748197	.3911167	.9816221	2.514771
(A vs O)	.6150312	.2432806	.13821	1.091852
(H vs A)	1.996549	.2730712	1.461339	2.531759

These ATEs are for incremental increases. In contrast, the ATEs considered above had a common base.



□ Technical note

The multivalued treatment AIPW estimators implemented in `teffects aipw` are EIF estimators based on the results of [Cattaneo \(2010\)](#). The results in [Cattaneo \(2010\)](#) are for semiparametric estimators, and we implement parametric versions. Of more practical importance, [Cattaneo \(2010\)](#) contains results for quantile treatment effects that are not implemented in `teffects` but implemented in the community-contributed `poparms` command discussed in [Cattaneo, Drukker, and Holland \(2013\)](#). See [Emsley et al. \(2008\)](#) for another implementation of the AIPW estimator, and see [Frölich and Melly \(2010\)](#) for other commands that estimate quantile treatment effects.



References

- Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267. <https://doi.org/10.1111/j.1468-0262.2006.00655.x>.
- . 2012. Matching on the estimated propensity score. Working Paper 15301, National Bureau of Economic Research. <https://doi.org/10.3386/w15301>.
- Blundell, R. W., L. Dearden, and B. Sianesi. 2005. Evaluating the effect of education on earnings: Models, methods and results from the National Child Development Survey. *Journal of the Royal Statistical Society, A ser.*, 168: 473–512. <https://doi.org/10.1111/j.1467-985X.2004.00360.x>.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Cattaneo, M. D., D. M. Drukker, and A. D. Holland. 2013. [Estimation of multivalued treatment effects under conditional independence](#). *Stata Journal* 13: 407–450.
- Emsley, R., M. Lunt, A. Pickles, and G. Dunn. 2008. [Implementing double-robust estimators of causal effects](#). *Stata Journal* 8: 334–353.
- Frölich, M., and B. Melly. 2010. [Estimation of quantile treatment effects with Stata](#). *Stata Journal* 10: 423–457.
- Imbens, G. W. 2000. The role of the propensity score in estimating dose–response functions. *Biometrika* 87: 706–710. <https://doi.org/10.1093/biomet/87.3.706>.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

- [\[CAUSAL\] teffects](#) — Treatment-effects estimation for observational data
- [\[CAUSAL\] teffects intro](#) — Introduction to treatment effects for observational data
- [\[CAUSAL\] teffects intro advanced](#) — Advanced introduction to treatment effects for observational data

[Description](#)
[Options](#)
[References](#)

[Quick start](#)
[Remarks and examples](#)
[Also see](#)

[Menu](#)
[Stored results](#)

[Syntax](#)
[Methods and formulas](#)

Description

`teffects nnmatch` estimates the average treatment effect (ATE) and average treatment effect on the treated (ATET) from observational data by nearest-neighbor matching (NNM). NNM estimators impute the missing potential outcome for each subject by using an average of the outcomes of similar subjects that receive the other treatment level. Similarity between subjects is based on a weighted function of the covariates for each observation. The treatment effect is computed by taking the average of the difference between the observed and imputed potential outcomes for each subject. `teffects nnmatch` accepts a continuous, binary, count, fractional, or nonnegative outcome.

See [\[CAUSAL\] teffects intro](#) or [\[CAUSAL\] teffects intro advanced](#) for more information about estimating treatment effects from observational data.

Quick start

ATE of `treat` on `y` estimated by NNM on `x1` and [indicators](#) for levels of categorical variable `a`

```
teffects nnmatch (y x1 i.a) (treat)
```

Same as above, but estimate the ATET

```
teffects nnmatch (y x1 i.a) (treat), atet
```

Add continuous covariate `x2` and perform bias correction

```
teffects nnmatch (y x1 x2 i.a) (treat), biasadj(x1 x2)
```

Same as above, and match exactly on values of `a`

```
teffects nnmatch (y x1 x2 i.a) (treat), biasadj(x1 x2) ematch(i.a)
```

With robust standard errors

```
teffects nnmatch (y x1 x2 i.a) (treat), vce(robust)
```

With four matches per observation

```
teffects nnmatch (y x1 x2 i.a) (treat), nneighbor(4)
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Nearest-neighbor matching

Statistics > Causal inference/treatment effects > Binary outcomes > Nearest-neighbor matching

Statistics > Causal inference/treatment effects > Count outcomes > Nearest-neighbor matching

Statistics > Causal inference/treatment effects > Fractional outcomes > Nearest-neighbor matching

Statistics > Causal inference/treatment effects > Nonnegative outcomes > Nearest-neighbor matching

Syntax

```
teffects nnmatch (ovar mvarlist) (tvar) [if] [in] [weight]
[ , stat options ]
```

ovar is a binary, count, continuous, fractional, or nonnegative outcome of interest.

mvarlist specifies the matching variables.

tvar must contain integer values representing the treatment levels. Only two treatment levels are allowed.

<i>stat</i>	Description
Stat	
<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated
<i>options</i>	Description
Model	
<code>nnneighbor(#)</code>	specify number of matches per observation; default is <code>nnneighbor(1)</code>
<code>biasadj(<i>varlist</i>)</code>	correct for large-sample bias using specified variables
<code>ematch(<i>varlist</i>)</code>	match exactly on specified variables
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>vce(robust [, nn(#)])</code> ; use robust Abadie–Imbens standard errors with # matches <code>vce(iid)</code> ; use independently and identically distributed Abadie–Imbens standard errors
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>dmvariables</code>	display names of matching variables
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Advanced	
<code>caliper(#)</code>	specify the maximum distance for which two observations are potential neighbors
<code>dtolerance(#)</code>	set maximum distance between individuals considered equal
<code>osample(<i>newvar</i>)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control(# <i>label</i>)</code>	specify the level of <i>tvar</i> that is the control
<code>tlevel(# <i>label</i>)</code>	specify the level of <i>tvar</i> that is the treatment
<code>generate(<i>stub</i>)</code>	generate variables containing the observation numbers of the nearest neighbors
<code>metric(<i>metric</i>)</code>	select distance metric for covariates
<code>coeflegend</code>	display legend instead of statistics

<i>metric</i>	Description
<u>mahalanobis</u>	inverse sample covariate covariance; the default
<u>ivariance</u>	inverse diagonal sample covariate covariance
<u>euclidean</u>	identity
<u>matrix matname</u>	user-supplied scaling matrix

mvarlist may contain factor variables; see [U] 11.4.3 Factor variables.

by, *collect*, and *statsby* are allowed; see [U] 11.1.10 Prefix commands.

fweights are allowed; see [U] 11.1.6 weight.

coeflegend does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

Model

`nneighbor(#)` specifies the number of matches per observation. The default is `nneighbor(1)`. Each observation is matched with at least the specified number of observations from the other treatment level. `nneighbor()` must specify an integer greater than or equal to 1 but no larger than the number of observations in the smallest treatment group.

`biasadj(varlist)` specifies that a linear function of the specified covariates be used to correct for a large-sample bias that exists when matching on more than one continuous covariate. By default, no correction is performed.

Abadie and Imbens (2006, 2011) show that nearest-neighbor matching estimators are not consistent when matching on two or more continuous covariates and propose a bias-corrected estimator that is consistent. The correction term uses a linear function of variables specified in `biasadj()`; see [example 3](#).

`ematch(varlist)` specifies that the variables in *varlist* match exactly. All variables in *varlist* must be numeric and may be specified as factors. `teffects nnmatch` exits with an error if any observations do not have the requested exact match.

Stat

stat is one of two statistics: *ate* or *atet*. *ate* is the default.

ate specifies that the average treatment effect be estimated.

atet specifies that the average treatment effect on the treated be estimated.

SE/Robust

`vce(vcetype)` specifies the standard errors that are reported. By default, `teffects nnmatch` uses robust standard errors estimated using two matches.

`vce(robust [ , nn(#) ])` specifies that robust standard errors be reported and that the requested number of matches be used optionally.

`vce(iid)` specifies that standard errors for independently and identically distributed data be reported.

The standard derivative-based standard-error estimators cannot be used by `teffects nnmatch`, because these matching estimators are not differentiable. The implemented methods were derived by [Abadie and Imbens \(2006, 2011, 2012\)](#); see [Methods and formulas](#).

As discussed in [Abadie and Imbens \(2008\)](#), bootstrap estimators do not provide reliable standard errors for the estimator implemented by `teffects nnmatch`.

Reporting

`level(#)`; see [\[R\] Estimation options](#).

`dmvariables` specifies that the matching variables be displayed.

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [\[R\] Estimation options](#).

Advanced

`caliper(#)` specifies the maximum distance at which two observations are a potential match. By default, all observations are potential matches regardless of how dissimilar they are.

The distance is based on `mvarlist`. If an observation does not have at least `nneighbor(#)` matches, `teffects nnmatch` exits with an error message. Use option `osample(newvar)` to identify all observations that are deficient in matches.

`dtolerance(#)` specifies the tolerance used to determine exact matches. The default value is `dtolerance(sqrt(c(epsdouble)))`.

Integer-valued variables are usually used for exact matching. The `dtolerance()` option is useful when continuous variables are used for exact matching.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption. This variable will identify all observations that do not have at least `nneighbor(#)` matches in the opposite treatment group within `caliper(#)` (for `metric()` distance matching) or `dtolerance(#)` (for `ematch(varlist)` exact matches).

The `vce(robust, nn(#))` option also requires at least `#` matches in the same treatment group within the distance specified by `caliper(#)` or within the exact matches specified by `dtolerance(#)`.

The average treatment effect on the treated, option `atet`, using `vce(iid)` requires only `nneighbor(#)` control group matches for the treated group.

`control(#|label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(#|label)` specifies the level of `tvar` that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

`generate(stub)` specifies that the observation numbers of the nearest neighbors be stored in the new variables `stub1`, `stub2`, This option is required if you wish to perform postestimation based on the matching results. The number of variables generated may be more than `nneighbor(#)` because of tied distances. These variables may not already exist.

`metric(metric)` specifies the distance matrix used as the weight matrix in a quadratic form that transforms the multiple distances into a single distance measure; see [Nearest-neighbor matching estimator](#) in *Methods and formulas* for details.

The following option is available with `teffects nnmatch` but is not shown in the dialog box:

`coeflegend`; see [\[R\] Estimation options](#).

Remarks and examples

The NNM method of treatment-effect estimation imputes the missing potential outcome for each individual by using an average of the outcomes of similar subjects that receive the other treatment level. Similarity between subjects is based on a weighted function of the covariates for each observation. The average treatment effect (ATE) is computed by taking the average of the difference between the observed and potential outcomes for each subject.

`teffects nnmatch` determines the “nearest” by using a weighted function of the covariates for each observation. By default, the Mahalanobis distance is used, in which the weights are based on the inverse of the covariates’ variance–covariance matrix. `teffects nnmatch` also allows you to request exact matching for categorical covariates. For example, you may want to force all matches to be of the same gender or race.

NNM is nonparametric in that no explicit functional form for either the outcome model or the treatment model is specified. This flexibility comes at a price; the estimator needs more data to get to the true value than an estimator that imposes a functional form. More formally, the NNM estimator converges to the true value at a rate slower than the parametric rate, which is the square root of the sample size, when matching on more than one continuous covariate. `teffects nnmatch` uses bias correction to fix this problem. `teffects psmatch` implements an alternative to bias correction; the method matches on a single continuous covariate, the estimated treatment probabilities. See [\[CAUSAL\] teffects intro](#) or [\[CAUSAL\] teffects intro advanced](#) for more information about this estimator.

We will illustrate the use of `teffects nnmatch` by using data from a study of the effect of a mother’s smoking status during pregnancy (`mb smoke`) on infant birthweight (`b weight`) as reported by [Cattaneo \(2010\)](#). This dataset also contains information about each mother’s age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), whether this baby was the mother’s first birth (`fbaby`), and the father’s age (`fage`).

► Example 1: Estimating the ATE

We begin by using `teffects nnmatch` to estimate the average treatment effect of `mbsmoke` on `bweight`. Subjects are matched using the Mahalanobis distance defined by covariates `mage`, `prenatal1`, `mmarried`, and `fbaby`.

```
. use https://www.stata-press.com/data/r19/cattaneo3
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects nnmatch (bweight mage prenatal1 mmarried fbaby) (mbsmoke)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : nearest-neighbor matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Distance metric: Mahalanobis                      max =     139
```

bweight	AI robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke						
(smoker						
vs						
nonsmoker)	-240.3306	28.43006	-8.45	0.000	-296.0525	-184.6087

The average birthweight if all mothers were to smoke would be 240 grams less than the average that would occur if none of the mothers had smoked.



When the list of variables you are matching on includes indicator and categorical variables, you may want to restrict matches to only those subjects who are in the same category. The `ematch()` option of `teffects nnmatch` allows you to specify such variables that must match exactly.

► Example 2: Exact matching

Here we use the `ematch()` option to require exact matches on the binary variables `prenatal1`, `mmarried`, and `fbaby`. We also use Euclidean distance, rather than the default Mahalanobis distance, to match on the continuous variable `mage`.

```
. teffects nnmatch (bweight mage) (mbsmoke),
> ematch(prenatal1 mmarried fbaby) metric(euclidean)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : nearest-neighbor matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Distance metric: Euclidean                      max =     139
```

bweight	AI robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke						
(smoker						
vs						
nonsmoker)	-240.3306	28.43006	-8.45	0.000	-296.0525	-184.6087



Abadie and Imbens (2006, 2011) have shown that nearest-neighbor matching estimators are not consistent when matching on two or more continuous covariates. A bias-corrected estimator that uses a linear function of variables can be specified with `biasadj()`.

► Example 3: Bias adjustment

Here we match on two continuous variables, `mage` and `fage`, and we use the bias-adjusted estimator:

```
. teffects nnmatch (bweight mage fage) (mb smoke),
> ematch(prenatal1 mmarried fbaby) biasadj(mage fage)
```

Treatment-effects estimation	Number of obs	=	4,642
Estimator : nearest-neighbor matching	Matches: requested	=	1
Outcome model : matching	min	=	1
Distance metric: Mahalanobis	max	=	25

bweight	AI robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mb smoke						
(smoker						
vs						
nonsmoker)	-223.8389	26.19973	-8.54	0.000	-275.1894	-172.4883

These results are similar to those reported in [example 1](#).



► Example 4: NNM can reduce to RA

NNM reduces to RA when matching exactly and all the covariates are discrete. We begin our illustration of this point by estimating the ATE by NNM using exact matching on `mmarried` and the mother's age-categories `magecat`.

```
. teffects nnmatch (bweight) (mb smoke), ematch(i.mmarried i.magecat)
```

Treatment-effects estimation	Number of obs	=	4,642
Estimator : nearest-neighbor matching	Matches: requested	=	1
Outcome model : matching	min	=	11
Distance metric: Mahalanobis	max	=	1310

bweight	AI robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mb smoke						
(smoker						
vs						
nonsmoker)	-241.5264	24.39661	-9.90	0.000	-289.3429	-193.71

The RA estimator that includes the interactions among the discrete covariates produces the same point estimate.

```
. teffects ra (bweight i.mmarrried##i.magecat) (mbsmoke)
Iteration 0: EE criterion = 1.523e-23
Iteration 1: EE criterion = 7.899e-26

Treatment-effects estimation          Number of obs      =      4,642
Estimator       : regression adjustment
Outcome model   : linear
Treatment model : none
```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE mbsmoke (smoker vs nonsmoker)	-241.5264	24.26233	-9.95	0.000	-289.0797	-193.9732
POmean mbsmoke nonsmoker	3403.651	9.492683	358.56	0.000	3385.046	3422.256

The two estimates of the ATE are the same. The standard errors differ in finite samples because the RA and NNM estimators use different robust estimators of the variance of the estimator.

With exact matching on discrete covariates, the NNM estimator reduces to an average of differences in cell means. With fully interacted discrete covariates, the RA estimator reduces to the same average of difference in cell means.



Video example

[Treatment effects in Stata: Nearest-neighbor matching](#)

Stored results

teffects nnmatch stores the following in e():

Scalars	
e(N)	number of observations
e(nj)	number of observations for treatment level <i>j</i>
e(k_levels)	number of levels in treatment variable
e(treated)	level of treatment variable defined as treated
e(control)	level of treatment variable defined as control
e(k_nnneighbor)	requested number of matches
e(k_nnmin)	minimum number of matches
e(k_nnmax)	maximum number of matches
e(k_robust)	matches for robust VCE
Macros	
e(cmd)	teffects
e(cmdline)	command as typed
e(depvar)	name of outcome variable
e(tvar)	name of treatment variable
e(emvarlist)	exact match variables

e(bavarlist)	variables used in bias adjustment
e(mvarlist)	matching variables
e(subcmd)	nnmatch
e(metric)	mahalanobis, ivariance, euclidean, or matrix <i>matname</i>
e(stat)	statistic estimated, ate or atet
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(tlevels)	levels of treatment variable
e(vce)	<i>vcetype</i> specified in <code>vce()</code>
e(vcetype)	title used to label Std. err.
e(datasignature)	the checksum
e(datasignaturevars)	variables used in calculation of checksum
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables <code>fvset</code> as asbalanced
e(asobserved)	factor variables <code>fvset</code> as asobserved
Matrices	
e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

The methods and formulas presented here provide the technical details underlying the estimators implemented in `teffects nnmatch` and `teffects psmatch`. See [Methods and formulas](#) of [CAUSAL] **teffects aipw** for the methods and formulas used by `teffects aipw`, `teffects ipw`, `teffects ipwra`, and `teffects ra`.

Methods and formulas are presented under the following headings:

Nearest-neighbor matching estimator
Bias-corrected matching estimator
Propensity-score matching estimator
PSM, ATE, and ATET variance adjustment

Nearest-neighbor matching estimator

`teffects nnmatch` implements the nearest-neighbor matching (NNM) estimator for the average treatment effect (ATE) and the average treatment effect on the treated (ATET). This estimator was derived by [Abadie and Imbens \(2006, 2011\)](#) and was previously implemented in Stata as discussed in [Abadie et al. \(2004\)](#).

`teffects psmatch` implements nearest-neighbor matching on an estimated propensity score. A propensity score is a conditional probability of treatment. The standard errors implemented in `teffects psmatch` were derived by [Abadie and Imbens \(2012\)](#).

`teffects nnmatch` and `teffects psmatch` permit two treatment levels: the treatment group with $t = 1$ and a control group with $t = 0$.

Matching estimators are based on the potential-outcome model, in which each individual has a well-defined outcome for each treatment level; see [\[CAUSAL\] teffects intro](#). In the binary-treatment potential-outcome model, y_1 is the potential outcome obtained by an individual if given treatment-level 1 and y_0 is the potential outcome obtained by each individual i if given treatment-level 0. The problem posed by the potential-outcome model is that only y_{1i} or y_{0i} is observed, never both. y_{0i} and y_{1i} are realizations of the random variables y_0 and y_1 . Throughout this document, i subscripts denote realizations of the corresponding, unsubscripted random variables.

Formally, the ATE is

$$\tau_1 = E(y_1 - y_0)$$

and the ATET is

$$\delta_1 = E(y_1 - y_0 | t = 1)$$

These expressions imply that we must have some solution to the missing-data problem that arises because we only observe either y_{1i} or y_{0i} , not both.

For each individual, NNM uses an average of the individuals that are most similar, but get the other treatment level, to predict the unobserved potential outcome. NNM uses the covariates $\{x_1, x_2, \dots, x_p\}$ to find the most similar individuals that get the other treatment level.

More formally, consider the vector of covariates $\mathbf{x}_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,p}\}$ and frequency weight w_i for observation i . The distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is parameterized by the vector norm

$$\|\mathbf{x}_i - \mathbf{x}_j\|_S = \{(\mathbf{x}_i - \mathbf{x}_j)' \mathbf{S}^{-1} (\mathbf{x}_i - \mathbf{x}_j)\}^{1/2}$$

where $\mathbf{S}$ is a given symmetric, positive-definite matrix.

Using this distance definition, we find that the set of nearest-neighbor indices for observation i is

$$\Omega_m^{\mathbf{x}}(i) = \{j_1, j_2, \dots, j_{m_i} \mid t_{j_k} = 1 - t_i, \|\mathbf{x}_i - \mathbf{x}_{j_k}\|_S < \|\mathbf{x}_i - \mathbf{x}_l\|_S, t_l = 1 - t_i, l \neq j_k\}$$

Here m_i is the smallest number such that the number of elements in each set, $m_i = |\Omega_m^{\mathbf{x}}(i)| = \sum_{j \in \Omega_m^{\mathbf{x}}(i)} w_j$, is at least m , the desired number of matches. You set the size of m using the `nneighbor (#)` option. The number of matches for the i th observation may not equal m because of ties or if there are not enough observations with a distance from observation i within the caliper limit, c , $\|\mathbf{x}_i - \mathbf{x}_j\|_S \leq c$. You may set the caliper limit by using the `caliper (#)` option. For ease of notation, we will use the abbreviation $\Omega(i) = \Omega_m^{\mathbf{x}}(i)$.

With the `metric(string)` option, you have three choices for the scaling matrix $\mathbf{S}$: Mahalanobis, inverse variance, or Euclidean.

$$\mathbf{S} = \begin{cases} \frac{(\mathbf{X} - \bar{\mathbf{x}}' \mathbf{1}_n)' \mathbf{W} (\mathbf{X} - \bar{\mathbf{x}}' \mathbf{1}_n)}{\sum_i^n w_i - 1} & \text{if metric = mahalanobis} \\ \text{diag} \left\{ \frac{(\mathbf{X} - \bar{\mathbf{x}}' \mathbf{1}_n)' \mathbf{W} (\mathbf{X} - \bar{\mathbf{x}}' \mathbf{1}_n)}{\sum_i^n w_i - 1} \right\} & \text{if metric = ivariance} \\ \mathbf{I}_p & \text{if metric = euclidean} \end{cases}$$

where $\mathbf{1}_n$ is an $n \times 1$ vector of ones, $\mathbf{I}_p$ is the identity matrix of order p , $\bar{\mathbf{x}} = (\sum_i^n w_i \mathbf{x}_i) / (\sum_i^n w_i)$, and $\mathbf{W}$ is an $n \times n$ diagonal matrix containing frequency weights.

The NNM method predicting the potential outcome for the i th observation as a function of the observed y_i is

$$\hat{y}_{ti} = \begin{cases} y_i & \text{if } t_i = t \\ \frac{\sum_{j \in \Omega(i)} w_j y_j}{\sum_{j \in \Omega(i)} w_j} & \text{otherwise} \end{cases}$$

for $t \in \{0, 1\}$.

We are now set to provide formulas for estimates $\hat{\tau}_1$, the ATE, and $\hat{\delta}_1$, the ATET,

$$\begin{aligned} \hat{\tau}_1 &= \frac{\sum_{i=1}^n w_i (\hat{y}_{1i} - \hat{y}_{0i})}{\sum_{i=1}^n w_i} = \frac{\sum_{i=1}^n w_i (2t_i - 1) \{1 + K_m(i)\} y_i}{\sum_{i=1}^n w_i} \\ \hat{\delta}_1 &= \frac{\sum_{i=1}^n t_i w_i (\hat{y}_{1i} - \hat{y}_{0i})}{\sum_{i=1}^n t_i w_i} = \frac{\sum_{i=1}^n \{t_i - (1 - t_i) K_m(i)\} y_i}{\sum_{i=1}^n t_i w_i} \end{aligned}$$

where

$$K_m(i) = \sum_{j=1}^n I\{i \in \Omega(j)\} \frac{w_j}{\sum_{k \in \Omega(j)} w_k}$$

The estimated variance of $\hat{\tau}_1$ and $\hat{\delta}_1$ are computed as

$$\hat{\sigma}_{\tau}^2 = \frac{\sum_{i=1}^n w_i \left[(\hat{y}_{1i} - \hat{y}_{0i} - \hat{\tau}_1)^2 + \hat{\xi}_i^2 \{K_m^2(i) + 2K_m(i) - K'_m(i)\} \right]}{\left(\sum_{i=1}^n w_i \right)^2}$$

$$\hat{\sigma}_{\delta}^2 = \frac{\sum_{i=1}^n t_i w_i \left[(\hat{y}_{1i} - \hat{y}_{0i} - \hat{\delta}_1)^2 + \hat{\xi}_i^2 \{K_m^2(i) - K'_m(i)\} \right]}{\left(\sum_{i=1}^n t_i w_i \right)^2}$$

where

$$K'_m(i) = \sum_{j=1}^n I\{i \in \Omega(j)\} \frac{w_j}{\left(\sum_{k \in \Omega(j)} w_k \right)^2}$$

and $\xi_i^2 = \text{var}(y_{ti} | \mathbf{x}_i)$ is the conditional outcome variance. If we can assume that ξ_i^2 does not vary with the covariates or treatment (homoskedastic), then we can compute an ATE estimate of ξ_{τ}^2 as

$$\hat{\xi}_{\tau}^2 = \frac{1}{2 \sum_i^n w_i} \sum_{i=1}^n w_i \left[\frac{\sum_{j \in \Omega(i)} w_j \{y_i - y_j(1 - t_i) - \hat{\tau}_1\}^2}{\sum_{j \in \Omega(i)} w_j} \right]$$

and an ATET estimate of ξ_{δ}^2 as

$$\hat{\xi}_{\delta}^2 = \frac{1}{2 \sum_i^n t_i w_i} \sum_{i=1}^n t_i w_i \left[\frac{\sum_{j \in \Omega(i)} t_j w_j \{y_i - y_j(1 - t_i) - \hat{\delta}_1\}^2}{\sum_{j \in \Omega(i)} t_j w_j} \right]$$

If the conditional outcome variance is dependent on the covariates or treatment, we require an estimate for ξ_i^2 at each observation. In this case, we require a second matching procedure, where we match on observations within the same treatment group.

Define the within-treatment matching set

$$\Psi_h^{\mathbf{x}}(i) = \{j_1, j_2, \dots, j_{h_i} \mid t_{j_k} = t_i, \|\mathbf{x}_i - \mathbf{x}_{j_k}\|_S < \|\mathbf{x}_i - \mathbf{x}_l\|_S, t_l = t_i, l \neq j_k\}$$

where h is the desired set size. As before, the number of elements in each set, $h_i = |\Psi_h^{\mathbf{x}}(i)|$, may vary depending on ties and the value of the caliper. You set h using the `vce(robust, nn(#))` option. As before, we will use the abbreviation $\Psi(i) = \Psi_h^{\mathbf{x}}(i)$ where convenient.

We estimate ξ_i^2 by

$$\hat{\xi}_{t_i}^2(\mathbf{x}_i) = \frac{\sum_{j \in \Psi(i)} w_j (y_j - \bar{y}_{\Psi(i)})^2}{\sum_{j \in \Psi(i)} w_j - 1} \quad \text{where} \quad \bar{y}_{\Psi(i)} = \frac{\sum_{j \in \Psi(i)} w_j y_j}{\sum_{j \in \Psi(i)} w_j - 1}$$

Bias-corrected matching estimator

When matching on more than one continuous covariate, the matching estimator described above is biased, even in infinitely large samples; in other words, it is not $\sqrt{n}$ -consistent; see [Abadie and Imbens \(2006, 2011\)](#). Following [Abadie and Imbens \(2011\)](#) and [Abadie et al. \(2004\)](#), `teffects nnmatch` makes an adjustment based on the regression functions $\mu_t(\tilde{\mathbf{x}}_i) = E(y_t \mid \tilde{\mathbf{x}} = \tilde{\mathbf{x}}_i)$, for $t = 0, 1$ and the set of covariates $\tilde{\mathbf{x}}_i = (\tilde{x}_{i,1}, \dots, \tilde{x}_{i,q})$. The bias-correction covariates may be the same as the NNM covariates $\mathbf{x}_i$. We denote the least-squares estimates as $\hat{\mu}_t(\tilde{\mathbf{x}}_i) = \hat{\nu}_t + \hat{\beta}'_t \tilde{\mathbf{x}}_i$, where we regress $\{y_i \mid t_i = t\}$ onto $\{\tilde{\mathbf{x}}_i \mid t_i = t\}$ with weights $w_i K_m(i)$, for $t = 0, 1$.

Given the estimated regression functions, the bias-corrected predictions for the potential outcomes are computed as

$$\hat{y}_{ti} = \begin{cases} y_i & \text{if } t_i = t \\ \frac{\sum_{j \in \Omega_m^*(i)} w_j \{y_j + \hat{\mu}_t(\tilde{\mathbf{x}}_i) - \hat{\mu}_t(\tilde{\mathbf{x}}_j)\}}{\sum_{j \in \Omega_m^*(i)} w_j} & \text{otherwise} \end{cases}$$

The `biasadj(varlist)` option specifies the bias-adjustment covariates $\tilde{\mathbf{x}}_i$.

Propensity-score matching estimator

The propensity-score matching (PSM) estimator uses a treatment model (TM), $p(\mathbf{z}_i, t, \gamma)$, to model the conditional probability that observation i receives treatment t given covariates $\mathbf{z}_i$. The literature calls $p(\mathbf{z}_i, t, \gamma)$ a propensity score, and PSM matches on the estimated propensity scores.

When matching on the estimated propensity score, the set of nearest-neighbor indices for observation i , $i = 1, \dots, n$, is

$$\Omega_m^{\mathbf{p}}(i) = \{j_1, j_2, \dots, j_{m_i} \mid t_{j_k} = 1 - t_i, |\hat{p}_i(t) - \hat{p}_{j_k}(t)| < |\hat{p}_i(t) - \hat{p}_l(t)|, t_l = 1 - t_i, l \neq j_k\}$$

where $\hat{p}_i(t) = p(\mathbf{z}_i, t, \hat{\gamma})$. As was the case with the NNM estimator, m_i is the smallest number such that the number of elements in each set, $m_i = |\Omega_m^{\mathbf{p}}(i)| = \sum_{j \in \Omega_m^{\mathbf{p}}(i)} w_j$, is at least m , the desired number of matches, set by the `nneighbor(#)` option.

We define the within-treatment matching set analogously,

$$\Psi_h^{\mathbf{p}}(i) = \{j_1, j_2, \dots, j_{h_i} \mid t_{j_k} = t_i, |\hat{p}_i(t) - \hat{p}_{j_k}(t)| < |\hat{p}_i(t) - \hat{p}_l(t)|, t_l = t_i, l \neq j_k\}$$

where h is the desired number of within-treatment matches, and $h_i = |\Psi_h^{\mathbf{p}}(i)|$, for $i = 1, \dots, n$, may vary depending on ties and the value of the caliper. The sets $\Psi_h^{\mathbf{p}}(i)$ are required to compute standard errors for $\hat{\tau}_1$ and $\hat{\delta}_1$.

Once a matching set is computed for each observation, the potential-outcome mean, ATE, and ATET computations are identical to those of NNM. The ATE and ATET standard errors, however, must be adjusted because the TM parameters were estimated; see [Abadie and Imbens \(2012\)](#).

PSM, ATE, and ATET variance adjustment

The variances for $\hat{\tau}_1$ and $\hat{\delta}_1$ must be adjusted because we use $\hat{\gamma}$ instead of γ . The adjusted variances for $\hat{\tau}_1$ and $\hat{\delta}_1$ have the following forms, respectively:

$$\begin{aligned}\hat{\sigma}_{\tau,\text{adj}}^2 &= \hat{\sigma}_{\tau}^2 + \hat{\mathbf{c}}_{\tau}' \hat{\mathbf{V}}_{\gamma} \hat{\mathbf{c}}_{\tau} \\ \hat{\sigma}_{\delta,\text{adj}}^2 &= \hat{\sigma}_{\delta}^2 - \hat{\mathbf{c}}_{\delta}' \hat{\mathbf{V}}_{\gamma} \hat{\mathbf{c}}_{\delta} + \frac{\partial \hat{\delta}_1}{\partial \gamma'} \hat{\mathbf{V}}_{\gamma} \frac{\partial \hat{\delta}_1}{\partial \gamma}\end{aligned}$$

In both equations, the matrix $\hat{\mathbf{V}}_{\gamma}$ is the TM coefficient variance–covariance matrix.

The adjustment term for ATE can be expressed as

$$\hat{\mathbf{c}}_{\tau} = \frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i f(\mathbf{z}_i' \hat{\gamma}) \left(\frac{\widehat{\text{cov}}(\mathbf{z}_i, \hat{y}_{i1})}{\hat{p}_i(1)} + \frac{\widehat{\text{cov}}(\mathbf{z}_i, \hat{y}_{i0})}{\hat{p}_i(0)} \right)$$

where

$$f(\mathbf{z}_i' \hat{\gamma}) = \frac{dp(\mathbf{z}_i, 1, \hat{\gamma})}{d(\mathbf{z}_i' \hat{\gamma})}$$

is the derivative of $p(\mathbf{z}_i, 1, \hat{\gamma})$ with respect to $\mathbf{z}_i' \hat{\gamma}$, and

$$\widehat{\text{cov}}(\mathbf{z}_i, \hat{y}_{ti}) = \begin{cases} \frac{\sum_{j \in \Psi_h(i)} w_j (\mathbf{z}_j - \bar{\mathbf{z}}_{\Psi_h(i)}) (y_j - \bar{y}_{\Psi_h(i)})}{\sum_{j \in \Psi_h(i)} w_j - 1} & \text{if } t_i = t \\ \frac{\sum_{j \in \Omega_h(i)} w_j (\mathbf{z}_j - \bar{\mathbf{z}}_{\Omega_h(i)}) (y_j - \bar{y}_{\Omega_h(i)})}{\sum_{j \in \Omega_h(i)} w_j - 1} & \text{otherwise} \end{cases}$$

is a $p \times 1$ vector with

$$\bar{\mathbf{z}}_{\Psi_h(i)} = \frac{\sum_{j \in \Psi_h(i)} w_j \mathbf{z}_j}{\sum_{j \in \Psi_h(i)} w_j} \quad \bar{\mathbf{z}}_{\Omega_h(i)} = \frac{\sum_{j \in \Omega_h(i)} w_j \mathbf{z}_j}{\sum_{j \in \Omega_h(i)} w_j} \quad \text{and} \quad \bar{y}_{\Omega_h(i)} = \frac{\sum_{j \in \Omega_h(i)} w_j y_j}{\sum_{j \in \Omega_h(i)} w_j}$$

Here we have used the notation $\Psi_h(i) = \Psi_h^{\mathbf{p}}(i)$ and $\Omega_h(i) = \Omega_h^{\mathbf{p}}(i)$ to stress that the within-treatment and opposite-treatment clusters used in computing $\hat{\sigma}_{\tau,\text{adj}}^2$ and $\hat{\sigma}_{\delta,\text{adj}}^2$ are based on h instead of the cluster $\Omega_m^{\mathbf{p}}(i)$ based on m used to compute $\hat{\tau}_1$ and $\hat{\delta}_1$, although you may desire to have $h = m$.

The adjustment term $\mathbf{c}_\delta$ for the ATET estimate has two components, $\mathbf{c}_\delta = \mathbf{c}_{\delta,1} + \mathbf{c}_{\delta,2}$, defined as

$$\begin{aligned}\mathbf{c}_{\delta,1} &= \frac{1}{\sum_{i=1}^n t_i w_i} \sum_{i=1}^n w_i \mathbf{z}_i f(\mathbf{z}'_i \hat{\gamma}) (\tilde{y}_{1i} - \tilde{y}_{0i} - \hat{\delta}_1) \\ \mathbf{c}_{\delta,2} &= \frac{1}{\sum_{i=1}^n t_i w_i} \sum_{i=1}^n w_i f(\mathbf{z}'_i \hat{\gamma}) \left\{ \widehat{\text{cov}}(\mathbf{z}_i, \hat{y}_{1i}) + \frac{\hat{p}_i(1)}{\hat{p}_i(0)} \widehat{\text{cov}}(\mathbf{z}_i, \hat{y}_{0i}) \right\}\end{aligned}$$

where

$$\tilde{y}_{ti} = \begin{cases} \frac{\sum_{j \in \Psi_h(-i)} w_j y_j}{\sum_{j \in \Psi_h(-i)} w_j} & \text{if } t = t_i \\ \frac{\sum_{j \in \Omega_h} w_j y_j}{\sum_{j \in \Omega_h} w_j} & \text{otherwise} \end{cases}$$

and the within-treatment matching sets $\Psi_h(-i) = \Psi_h^{\mathbf{p}}(-i)$ are similar to $\Psi_h^{\mathbf{p}}(i)$ but exclude observation i :

$$\Psi_h^{\mathbf{p}}(-i) = \{j_1, j_2, \dots, j_{h_i} \mid j_k \neq i, t_{j_k} = t_i, |\hat{p}_i - \hat{p}_{j_k}| < |\hat{p}_i - \hat{p}_l|, t_l = t_i, l \notin \{i, j_k\}\}$$

Finally, we cover the computation of $\frac{\partial \hat{\delta}_1}{\partial \gamma'}$ in the third term on the right-hand side of $\hat{\sigma}_{\delta, \text{adj}}^2$. Here we require yet another clustering, but we match on the opposite treatment by using the covariates $\mathbf{z}_i = (z_{i,1}, \dots, z_{i,p})'$. We will denote these cluster sets as $\Omega_m^{\mathbf{z}}(i)$, for $i = 1, \dots, n$.

The estimator of the $p \times 1$ vector $(\partial \delta_1) / (\partial \gamma')$ is computed as

$$\frac{\partial \hat{\delta}_1}{\partial \gamma'} = \frac{1}{\sum_{i=1}^n t_i w_i} \sum_{i=1}^n \mathbf{z}_i f(\mathbf{z}'_i \hat{\gamma}) \left\{ (2t_i - 1)(y_i - \bar{y}_{\Omega_m^{\mathbf{z}}(i)}) - \hat{\delta}_1 \right\}$$

where

$$\bar{y}_{\Omega_m^{\mathbf{z}}(i)} = \frac{\sum_{j \in \Omega_m^{\mathbf{z}}(i)} w_j y_j}{\sum_{j \in \Omega_m^{\mathbf{z}}(i)} w_j}$$

References

- Abadie, A., D. M. Drukker, J. L. Herr, and G. W. Imbens. 2004. [Implementing matching estimators for average treatment effects in Stata](#). *Stata Journal* 4: 290–311.
- Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267. <https://doi.org/10.1111/j.1468-0262.2006.00655.x>.
- . 2008. On the failure of the bootstrap for matching estimators. *Econometrica* 76: 1537–1557. <https://doi.org/10.3982/ECTA6474>.
- . 2011. Bias-corrected matching estimators for average treatment effects. *Journal of Business and Economic Statistics* 29: 1–11. <https://doi.org/10.1198/jbes.2009.07333>.
- . 2012. Matching on the estimated propensity score. Working Paper 15301, National Bureau of Economic Research. <https://doi.org/10.3386/w15301>.

- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Díaz, J. D., I. Gutiérrez, and J. Rivera. 2021. Implementing blopmatching in Stata. *Stata Journal* 21: 180–194.
- Drukker, D. M. 2016. Exact matching on discrete covariates is the same as regression adjustment. *The Stata Blog: Not Elsewhere Classified*. <https://blog.stata.com/2016/08/16/exact-matching-on-discrete-covariates-is-the-same-as-regression-adjustment/>.
- Huber, C. 2015. Introduction to treatment effects in Stata: Part 2. *The Stata Blog: Not Elsewhere Classified*. <https://blog.stata.com/2015/08/24/introduction-to-treatment-effects-in-stata-part-2/>.

Also see

- [CAUSAL] [teffects postestimation](#) — Postestimation tools for teffects
- [CAUSAL] [teffects](#) — Treatment-effects estimation for observational data
- [CAUSAL] [teffects psmatch](#) — Propensity-score matching
- [U] [20 Estimation and postestimation commands](#)

Postestimation commands

The following postestimation command is of special interest after `teffects`:

Command	Description
<code>teoverlap</code>	overlap plots
<code>tebalance</code>	check balance of covariates

The following standard postestimation commands are also available:

Command	Description
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>hausman</code>	Hausman’s specification test
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of parameters
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
<code>predict</code>	treatment effects, conditional means at treatment, propensity scores, etc.
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

predict

Description for predict

predict creates a new variable containing predictions such as treatment effects, potential outcomes, conditional means, propensity scores, linear predictions, nearest-neighbor distances, and log square root of latent variances.

Menu for predict

Statistics > Postestimation

Syntaxes for predict

Syntaxes are presented under the following headings:

- Syntax for predict after aipw and ipwra
- Syntax for predict after ipw
- Syntax for predict after nnmatch and psmatch
- Syntax for predict after ra

Syntax for predict after aipw and ipwra

```
predict [ type ] { stub* | newvar | newvars } [ if ] [ in ]  
      [ , statistic tlevel( treat_level ) ]  
  
predict [ type ] stub* [ if ] [ in ], scores
```

statistic	Description
Main	
te	treatment effect; the default
cmean	conditional mean at treatment level
ps	propensity score
xb	linear prediction
psxb	linear prediction for propensity score
lnsigma	log square root of conditional latent variance (for outcome model hetprobit()) at treatment level
pslnsigma	log square root of latent variance (for treatment model hetprobit())

If you do not specify tlevel() and only specify one new variable, te and psxb assume tlevel() specifies the first non-control treatment level.

If you do not specify tlevel() and only specify one new variable, cmean, ps, xb, and lnsigma assume tlevel() specifies the first treatment level.

You specify one or *t* new variables with cmean, ps, xb, and lnsigma, where *t* is the number of treatment levels.

You specify one or *t* − 1 new variables with te, psxb, and pslnsigma.

Syntax for predict after ipw

```
predict [type] { stub* | newvar | newvars } [if] [in]
      [, statistic tlevel(treat_level)]

predict [type] stub* [if] [in], scores
```

statistic	Description
Main	
ps	propensity score; the default
xb	linear prediction for the propensity score
lnsigma	log square root of latent variance (for treatment model hetprobit())

If you do not specify tlevel() and only specify one new variable, ps assumes tlevel() specifies the first treatment level.

If you do not specify tlevel() and only specify one new variable, xb assumes tlevel() specifies the first noncontrol treatment level.

You specify one or t new variables with ps, where t is the number of treatment levels.

You specify one or $t - 1$ new variables with xb and lnsigma.

Syntax for predict after nnmatch and psmatch

```
predict [type] { stub* | newvars } [, statistic tlevel(treat_level)]
```

statistic	Description
Main	
te	treatment effect; the default
po	potential outcome
distance	nearest-neighbor distance
ps	propensity score (psmatch only)
lnsigma	log square root of latent variance (for treatment model hetprobit())

These statistics are available for the estimation sample only and require the estimation option generate(stub). This is because of the nonparametric nature of the matching estimator.

If you do not specify tlevel() and only specify one new variable, po and ps assume tlevel() specifies the first treatment level.

You specify one new variable with te and lnsigma.

You specify one or two new variables with po and ps.

Syntax for predict after ra

```
predict [type] { stub* | newvar | newvars } [if] [in]
      [, statistic tlevel(treat_level)]
```

```
predict [type] stub* [if] [in], scores
```

statistic	Description
Main	
te	treatment effect; the default
cmean	conditional mean at treatment level
xb	linear prediction
lnsigma	log square root of conditional latent variance (for outcome model hetprobit()) at treatment level

If you do not specify tlevel() and only specify one new variable, te assumes tlevel() specifies the first noncontrol treatment level.

If you do not specify tlevel() and only specify one new variable, cmean, xb, and lnsigma assume tlevel() specifies the first treatment level.

You specify one or *t* new variables with cmean, xb, and lnsigma, where *t* is the number of treatment levels.

You specify one or *t* – 1 new variables with te.

Options for predict

Options are presented under the following headings:

- [Options for predict after aipw and ipwra](#)
- [Options for predict after ipw](#)
- [Options for predict after nnmatch and psmatch](#)
- [Options for predict after ra](#)

Options for predict after aipw and ipwra

Main

te, the default, calculates the treatment effect for each noncontrol treatment level or the treatment level specified in tlevel(). If you specify the tlevel() option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

cmean calculates the conditional mean for each treatment level or the treatment level specified in tlevel(). If you specify the tlevel() option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

ps calculates the propensity score of each treatment level or the treatment level specified in tlevel(). If you specify the tlevel() option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

xb calculates the linear prediction at each treatment level or the treatment level specified in tlevel(). If you specify the tlevel() option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`psxb` calculates the linear prediction for the propensity score at each noncontrol level of the treatment or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

`lnsigma` calculates the log square root of the conditional latent variance for each treatment level or the treatment level specified in `tlevel()`. This option is valid when outcome model `hetprobit()` was used. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`pslnsigma` calculates the log square root of the latent variance for the propensity score. This option is only valid when treatment model `hetprobit()` was used. Specify only one new variable.

`tlevel(treat_level)` specifies the treatment level for prediction.

`scores` calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the propensity-score equations.

The j th new variable will contain the scores for the j th parameter in the coefficient table if $j \leq t$, where t is the number of treatment levels. Otherwise, it will contain the scores for fitted equation $j - t$ following the first t parameters in the coefficient table.

Options for predict after ipw

Main

`ps`, the default, calculates the propensity score of each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

`xb` calculates the linear prediction for the propensity score at each noncontrol level of the treatment or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

`lnsigma` calculates the log square root of the latent variance. This option is only valid when treatment model `hetprobit()` was used. Specify only one new variable.

`tlevel(treat_level)` specifies the treatment level for prediction.

`scores` calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the propensity-score equations.

The j th new variable will contain the scores for the j th parameter in the coefficient table if $j \leq t$, where t is the number of treatment levels. Otherwise, it will contain the scores for fitted equation $j - t$ following the first t parameters in the coefficient table.

Options for predict after nnmatch and psmatch

Main

te, the default, calculates the treatment effect.

po calculates the predicted potential outcomes for each observation and treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify new variables for the control and treated groups.

distance calculates the distances of the nearest neighbors for each observation. The number of variables generated is equal to the maximum number of nearest-neighbor matches. This is equal to the number of index variables generated by the estimation option `generate(stub)`. You may use the `stub*` syntax to set the distance variable prefix: `stub1`, `stub2`, ...

ps calculates the propensity score of each treatment level or the propensity score of the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify new variables for the control and treated groups.

lnsigma calculates the log square root of the latent variance. This option is only valid when treatment model `hetprobit()` was used. Specify only one new variable.

tlevel(*treat_level*) restricts potential-outcome estimation to either the treated group or the control group. This option may only be specified with options **po** and **ps**.

Options for predict after ra

Main

te, the default, calculates the treatment effect for each noncontrol treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level (except the control level).

cmean calculates the conditional mean for each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

xb calculates the linear prediction at each treatment level or the treatment level specified in `tlevel()`. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

lnsigma calculates the log square root of the conditional latent variance for each treatment level or the treatment level specified in `tlevel()`. This option is valid when outcome model `hetprobit()` was used. If you specify the `tlevel()` option, you need to specify only one new variable; otherwise, you must specify a new variable for each treatment level.

tlevel(*treat_level*) specifies the treatment level for prediction.

scores calculates the score variables. Parameter-level scores are computed for the treatment mean and average treatment-effect equations. Equation-level scores are computed for the regression equations.

The j th new variable will contain the scores for the j th parameter in the coefficient table if $j \leq t$, where t is the number of treatment levels. Otherwise, it will contain the scores for fitted equation $j - t$ following the first t parameters in the coefficient table.

Remarks and examples

Checking model specification is the most frequent reason for postestimation computation after `teffects`. `teoverlap` provides a graphical method for checking the overlap assumption; see [\[CAUSAL\] teoverlap](#). Summarizing the estimated probabilities provides another check. Recall that the reciprocals of these estimated probabilities are used as weights by some of the estimators. If the estimated probabilities are too small, the weights blow up.

We estimate the ATE of maternal smoking on infant birthweight by inverse-probability weighting; see [example 1](#) of [\[CAUSAL\] teffects ipw](#) for background.

```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)
Iteration 0: EE criterion = 4.622e-21
Iteration 1: EE criterion = 8.795e-26

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke (Smoker vs Nonsmoker)	-230.6886	25.81524	-8.94	0.000	-281.2856	-180.0917
P0mean						
mbsmoke Nonsmoker	3403.463	9.571369	355.59	0.000	3384.703	3422.222

Below we compute and summarize the estimated treatment probabilities.

```
. predict pr1
(option ps assumed; propensity score)
. summarize pr1 if mbsmoke==1, detail
                    propensity score, mbsmoke=Nonsmoker
```

Percentiles		Smallest		
1%	.2991634	.2196947		
5%	.544155	.2258079		
10%	.5973879	.2258079	Obs	864
25%	.63777	.2409025	Sum of wgt.	864
50%	.7601717		Mean	.7456264
		Largest	Std. dev.	.1276102
75%	.8453946	.9533503		
90%	.8943686	.9596144	Variance	.0162844
95%	.9096801	.961022	Skewness	-.7701643
99%	.9367017	.9665684	Kurtosis	3.858214

The smallest values do not imply very large weights.

Below we compute and summarize the estimated probabilities of not getting the treatment.

```
. generate pr0 = 1 -pr1
. summarize pr0 if mbsmoke==0, detail
```

pr0			
Percentiles		Smallest	
1%	.0351884	.0074551	
5%	.0578012	.0079309	
10%	.0674359	.0106305	Obs 3,778
25%	.0950869	.0106305	Sum of wgt. 3,778
50%	.1372589		Mean .1698913
		Largest	Std. dev. .1059434
75%	.2211142	.7547572	
90%	.3242757	.774192	Variance .011224
95%	.3883457	.7803053	Skewness 1.514456
99%	.501537	.7816764	Kurtosis 6.151114

Although there are two small probabilities, overall the small values do not imply large weights.

Also see

- [CAUSAL] [teoverlap](#) — Overlap plots
- [CAUSAL] [teffects aipw](#) — Augmented inverse-probability weighting
- [CAUSAL] [teffects ipw](#) — Inverse-probability weighting
- [CAUSAL] [teffects ipwra](#) — Inverse-probability-weighted regression adjustment
- [CAUSAL] [teffects nnmatch](#) — Nearest-neighbor matching
- [CAUSAL] [teffects psmatch](#) — Propensity-score matching
- [CAUSAL] [teffects ra](#) — Regression adjustment
- [U] [20 Estimation and postestimation commands](#)

[Description](#)
[Options](#)
[References](#)

[Quick start](#)
[Remarks and examples](#)
[Also see](#)

[Menu](#)
[Stored results](#)

[Syntax](#)
[Methods and formulas](#)

Description

`teffects psmatch` estimates the average treatment effect (ATE) and average treatment effect on the treated (ATET) from observational data by propensity-score matching (PSM). PSM estimators impute the missing potential outcome for each subject by using an average of the outcomes of similar subjects that receive the other treatment level. Similarity between subjects is based on estimated treatment probabilities, known as propensity scores. The treatment effect is computed by taking the average of the difference between the observed and potential outcomes for each subject. `teffects psmatch` accepts a continuous, binary, count, fractional, or nonnegative outcome.

See [\[CAUSAL\] teffects intro](#) or [\[CAUSAL\] teffects intro advanced](#) for more information about estimating treatment effects from observational data.

Quick start

ATE of `treat` on `y` estimated by PSM using a logistic model for `treat` on `x` and [indicators](#) for levels of categorical variable `a`

```
teffects psmatch (y) (treat x i.a)
```

Same as above, but estimate the ATET

```
teffects psmatch (y) (treat x i.a), atet
```

ATE of `treat` using a heteroskedastic probit model for treatment

```
teffects psmatch (y) (treat x i.a, hetprobit(x i.a))
```

With 4 matches per observation

```
teffects psmatch (y) (treat x i.a), nneighbor(4)
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Propensity-score matching

Statistics > Causal inference/treatment effects > Binary outcomes > Propensity-score matching

Statistics > Causal inference/treatment effects > Count outcomes > Propensity-score matching

Statistics > Causal inference/treatment effects > Fractional outcomes > Propensity-score matching

Statistics > Causal inference/treatment effects > Nonnegative outcomes > Propensity-score matching

Syntax

```
teffects psmatch (ovar) (tvar tmvarlist [ , tmodel ]) [if] [in] [weight]
[ , stat options ]
```

ovar is a binary, count, continuous, fractional, or nonnegative outcome of interest.

tvar must contain integer values representing the treatment levels.

tmvarlist specifies the variables that predict treatment assignment in the treatment model. Only two treatment levels are allowed.

<i>tmodel</i>	Description
---------------	-------------

Model

logit	logistic treatment model; the default
probit	probit treatment model
hetprobit (<i>varlist</i>)	heteroskedastic probit treatment model

tmodel specifies the model for the treatment variable.

<i>stat</i>	Description
-------------	-------------

Stat

ate	estimate average treatment effect in population; the default
atet	estimate average treatment effect on the treated

<i>options</i>	Description
----------------	-------------

Model

<u>n</u> neighbor(#)	specify number of matches per observation; default is nneighbor(1)
----------------------	--

SE/Robust

vce(<i>vcetype</i>)	<i>vcetype</i> may be vce(<u>r</u> obust [, nn(#)]); use robust Abadie–Imbens standard errors with # matches vce(iid); use independent and identically distributed Abadie–Imbens standard errors
-----------------------	--

Reporting

<u>l</u> evel(#)	set confidence level; default is level(95)
<i>display_options</i>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

Advanced

<u>c</u> aliper(#)	specify the maximum distance for which two observations are potential neighbors
<u>p</u> stolerance(#)	set tolerance for overlap assumption
<u>o</u> sample(<i>newvar</i>)	<i>newvar</i> identifies observations that violate the overlap assumption
<u>c</u> ontrol(# <i>label</i>)	specify the level of <i>tvar</i> that is the control
<u>t</u> level(# <i>label</i>)	specify the level of <i>tvar</i> that is the treatment
<u>g</u> enerate(<i>stub</i>)	generate variables containing the observation numbers of the nearest neighbors
<u>c</u> oef <u>l</u> egend	display legend instead of statistics

tmvarlist may contain factor variables; see [U] 11.4.3 **Factor variables**.

by, *collect*, and *statsby* are allowed; see [U] 11.1.10 **Prefix commands**.

fwweights are allowed; see [U] 11.1.6 **weight**.

coeflegend does not appear in the dialog box.

See [U] 20 **Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

nneighbor(#) specifies the number of matches per observation. The default is *nneighbor*(1). Each individual is matched with at least the specified number of individuals from the other treatment level. *nneighbor*() must specify an integer greater than or equal to 1 but no larger than the number of observations in the smallest group.

Stat

stat is one of two statistics: *ate* or *atet*. *ate* is the default.

ate specifies that the average treatment effect be estimated.

atet specifies that the average treatment effect on the treated be estimated.

SE/Robust

vce(*vcetype*) specifies the standard errors that are reported. By default, *teffects psmatch* uses two matches in estimating the robust standard errors.

vce(*robust* [, *nn*(#)]) specifies that robust standard errors be reported and that the requested number of matches be used optionally.

vce(*iid*) specifies that standard errors for independent and identically distributed data be reported.

The standard derivative-based standard-error estimators cannot be used by *teffects psmatch*, because these matching estimators are not differentiable. The implemented methods were derived by [Abadie and Imbens \(2006, 2011, 2012\)](#); see *Methods and formulas*.

As discussed in [Abadie and Imbens \(2008\)](#), bootstrap estimators do not provide reliable standard errors for the estimator implemented by *teffects psmatch*.

Reporting

level(#); see [R] **Estimation options**.

display_options: *noc*i, *nop*values, *noom*itted, *vsqu*ish, *noem*ptycells, *base*levels, *allbase*levels, *no*fvlabel, *fv*wrap(#), *fv*wrapon(*style*), *c*format(*%fmt*), *p*format(*%fmt*), *s*format(*%fmt*), and *no*lstretch; see [R] **Estimation options**.

Advanced

caliper(#) specifies the maximum distance at which two observations are a potential match. By default, all observations are potential matches regardless of how dissimilar they are.

In *teffects psmatch*, the distance is measured by the estimated propensity score. If an observation has no matches, *teffects psmatch* exits with an error.

pstolerance(#) specifies the tolerance used to check the overlap assumption. The default value is *pstolerance*(1e-5). *teffects* will exit with an error if an observation has an estimated propensity score smaller than that specified by *pstolerance*() .

`osample(newvar)` specifies that indicator variable *newvar* be created to identify observations that violate the overlap assumption. Two checks are made to verify the assumption. The first ensures that the propensity scores are greater than `pstolerance(#)` and less than $1 - \text{pstolerance}(\#)$. The second ensures that each observation has at least `nneighbor(#)` matches in the opposite treatment group within the distance specified by `caliper(#)`.

The `vce(robust, nn(#))` option also requires at least *#* matches in the same treatment group within the distance specified by `caliper(#)`.

The average treatment effect on the treated, option `atet`, using `vce(iid)` requires only `nneighbor(#)` control group matches for the treated group.

`control(#|label)` specifies the level of *tvar* that is the control. The default is the first treatment level.

You may specify the numeric level *#* (a nonnegative integer) or the label associated with the numeric level. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(#|label)` specifies the level of *tvar* that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level *#* (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

`generate(stub)` specifies that the observation numbers of the nearest neighbors be stored in the new variables *stub1*, *stub2*, This option is required if you wish to perform postestimation based on the matching results. The number of variables generated may be more than `nneighbor(#)` because of tied distances. These variables may not already exist.

The following option is available with `teffects psmatch` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

Propensity-score matching uses an average of the outcomes of similar subjects who get the other treatment level to impute the missing potential outcome for each subject. The ATE is computed by taking the average of the difference between the observed and potential outcomes for each subject. `teffects psmatch` determines how near subjects are to each other by using estimated treatment probabilities, known as propensity scores. This type of matching is known as propensity-score matching (PSM).

PSM does not need bias correction, because PSM matches on a single continuous covariate. In contrast, the nearest-neighbor matching estimator implemented in `teffects nnmatch` uses a bias-correction term when matching on more than one continuous covariate. In effect, the PSM estimator parameterizes the bias-correction term in the treatment probability model. See [\[CAUSAL\] teffects intro](#) or [\[CAUSAL\] teffects intro advanced](#) for more information about this estimator.

We will illustrate the use of `teffects psmatch` by using data from a study of the effect of a mother's smoking status during pregnancy (*mb smoke*) on infant birthweight (*bweight*) as reported by [Cattaneo \(2010\)](#). This dataset also contains information about each mother's age (*mage*), education level (*medu*), marital status (*mmarried*), whether the first prenatal exam occurred in the first trimester (*prenatal1*), whether this baby was the mother's first birth (*fbaby*), and the father's age (*face*).

► Example 1: Estimating the ATE

We begin by using `teffects psmatch` to estimate the ATE of `mbsmoke` on `bweight`. We use a logistic model (the default) to predict each subject's propensity score, using covariates `mage`, `medu`, `mmarried`, and `fbaby`. Because the performance of PSM hinges upon how well we can predict the propensity scores, we will use factor-variable notation to include both linear and quadratic terms for `mage`, the only continuous variable in our model:

```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : propensity-score matching      Matches: requested =      1
Outcome model  : matching                      min =      1
Treatment model: logit                      max =      74
```

bweight	AI robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke						
(Smoker						
vs						
Nonsmoker)	-210.9683	32.021	-6.59	0.000	-273.7284	-148.2083

The average birthweight if all mothers were to smoke would be 211 grams less than the average that would occur if none of the mothers had smoked.



By default, `teffects psmatch` estimates the ATE by matching each subject to a single subject with the opposite treatment whose propensity score is closest. Sometimes, however, we may want to ensure that matching occurs only when the propensity scores of a subject and a match differ by less than a specified amount. To do that, we use the `caliper()` option. If a match within the distance specified in the `caliper()` option cannot be found, `teffects psmatch` exits.

► Example 2: Specifying the caliper

Here we reconsider the previous example, first specifying that we only want to consider a pair of observations a match if the absolute difference in the propensity scores is less than 0.03:

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> caliper(0.03)
no propensity-score matches for observation 4504 within caliper 0.03; use option
osample() to identify all observations with deficient matches
r(459);
```

The error arose because there is not a smoking mother whose propensity score is within 0.03 of the propensity score of the nonsmoking mother in observation 4504. If we instead raise the caliper to 0.10, we have matches for all subjects and therefore obtain the same results as in [example 1](#):

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> caliper(0.1)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : propensity-score matching    Matches: requested =      1
Outcome model  : matching                    min =      1
Treatment model: logit                      max =      74
```

bweight	Coefficient	AI robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (Smoker vs Nonsmoker)	-210.9683	32.021	-6.59	0.000	-273.7284	-148.2083



□ Technical note

[Example 2](#) highlights that estimating the ATE requires finding matches for both the treated and control subjects. In contrast, estimating the ATET only requires finding matches for the treated subjects. Because subject 4504 is a control subject, we can estimate the ATET using `caliper(0.03)`. We must also specify `vce(iid)` because the default robust standard errors for the estimated ATET require viable matches for both treated subjects and control subjects. (This requirement comes from the nonparametric method derived by [Abadie and Imbens \[2012\]](#).)

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu),
> atet vce(iid) caliper(0.03)

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : propensity-score matching    Matches: requested =      1
Outcome model  : matching                    min =      1
Treatment model: logit                      max =      74
```

bweight	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
ATET						
mbsmoke (Smoker vs Nonsmoker)	-236.7848	26.11698	-9.07	0.000	-287.9731	-185.5964



In the previous examples, each subject was matched to at least one other subject, which is the default behavior for `teffects psmatch`. However, we can request that `teffects psmatch` match each subject to multiple subjects with the opposite treatment level by specifying the `nneighbor()` option. Matching on more distant neighbors can reduce the variance of the estimator at a cost of an increase in bias.

► Example 3

Now we request that `teffects psmatch` match a mother to four mothers in the opposite treatment group:

```
. teffects psmatch (bweight) (mbsmoke mmarried c.mage#c.mage fbaby medu),
> nneighbor(4)
```

Treatment-effects estimation	Number of obs	=	4,642
Estimator : propensity-score matching	Matches: requested	=	4
Outcome model : matching	min	=	4
Treatment model: logit	max	=	74

bweight	Coefficient	AI robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (Smoker vs Nonsmoker)	-224.006	29.88627	-7.50	0.000	-282.582	-165.43

These results are similar to those reported in [example 1](#).



Video example

[Treatment effects in Stata: Propensity-score matching](#)

Stored results

`teffects psmatch` stores the following in `e()`:

- Scalars
- e(N)

number of observations
- e(nj)

number of observations for treatment level *j*
- e(k_levels)

number of levels in treatment variable
- e(caliper)

maximum distance between matches
- e(treated)

level of treatment variable defined as treated
- e(control)

level of treatment variable defined as control
- e(k_nneighbor)

requested number of matches
- e(k_nnmin)

minimum number of matches
- e(k_nnmax)

maximum number of matches
- e(k_robust)

matches for robust VCE
- Macros
- e(cmd)

teffects
- e(cmdline)

command as typed
- e(depvar)

name of outcome variable
- e(tvar)

name of treatment variable
- e(subcmd)

psmatch
- e(tmodel)

logit, probit, or hetprobit
- e(stat)

statistic estimated, ate or atet
- e(wtype)

weight type
- e(wexp)

weight expression
- e(title)

title in estimation output
- e(tlevels)

levels of treatment variable
- e(psvarlist)

variables in propensity-score model

<code>e(hvarlist)</code>	variables for variance, only if <code>hetprobit</code>
<code>e(vce)</code>	<code>vcetype</code> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(datasignature)</code>	the checksum
<code>e(datasignaturevars)</code>	variables used in calculation of checksum
<code>e(properties)</code>	<code>b</code> <code>V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by margins
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>
Matrices	
<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators
<code>e(bps)</code>	coefficient vector from propensity-score model
<code>e(Vps)</code>	variance–covariance matrix of the estimators from propensity-score model
Functions	
<code>e(sample)</code>	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

The methods and formulas used by `teffects psmatch` are documented in the [Methods and formulas](#) of [\[CAUSAL\]](#) `teffects nnmatch`.

References

- Abadie, A., and G. W. Imbens. 2006. Large sample properties of matching estimators for average treatment effects. *Econometrica* 74: 235–267. <https://doi.org/10.1111/j.1468-0262.2006.00655.x>.
- . 2008. On the failure of the bootstrap for matching estimators. *Econometrica* 76: 1537–1557. <https://doi.org/10.3982/ECTA6474>.
- . 2011. Bias-corrected matching estimators for average treatment effects. *Journal of Business and Economic Statistics* 29: 1–11. <https://doi.org/10.1198/jbes.2009.07333>.
- . 2012. Matching on the estimated propensity score. Working Paper 15301, National Bureau of Economic Research. <https://doi.org/10.3386/w15301>.
- Alejo, J., A. F. Galvao, and G. Montes-Rojas. 2020. A practical generalized propensity-score estimator for quantile continuous treatment effects. *Stata Journal* 20: 276–296.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Díaz, J. D., I. Gutiérrez, and J. Rivera. 2021. Implementing `blopmatching` in Stata. *Stata Journal* 21: 180–194.
- Huber, C. 2015. Introduction to treatment effects in Stata: Part 2. *The Stata Blog: Not Elsewhere Classified*. <https://blog.stata.com/2015/08/24/introduction-to-treatment-effects-in-stata-part-2/>.
- Tazare, J., L. Smeeth, S. J. W. Evans, I. J. Douglas, and E. J. Williamson. 2023. `hdps`: A suite of commands for applying high-dimensional propensity-score approaches. *Stata Journal* 23: 683–708.

Also see

[CAUSAL] [teffects postestimation](#) — Postestimation tools for teffects

[CAUSAL] [teffects](#) — Treatment-effects estimation for observational data

[CAUSAL] [teffects nnmatch](#) — Nearest-neighbor matching

[U] [20 Estimation and postestimation commands](#)

[Description](#)
[Options](#)
[References](#)

[Quick start](#)
[Remarks and examples](#)
[Also see](#)

[Menu](#)
[Stored results](#)

[Syntax](#)
[Methods and formulas](#)

Description

`teffects ra` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational data by regression adjustment (RA). RA estimators use contrasts of averages of treatment-specific predicted outcomes to estimate treatment effects. `teffects ra` accepts a continuous, binary, count, fractional, or nonnegative outcome and allows a multivalued treatment.

See [\[CAUSAL\] teffects intro](#) or [\[CAUSAL\] teffects intro advanced](#) for more information about estimating treatment effects from observational data.

Quick start

ATE from a linear model of `y1` on `x1` and `x2` with binary treatment `treat2`

```
teffects ra (y1 x1 x2) (treat2)
```

Same as above, but estimate the ATET

```
teffects ra (y1 x1 x2) (treat2), atet
```

Same as above, but estimate the potential-outcome means

```
teffects ra (y1 x1 x2) (treat2), pomeans
```

ATE of `treat2` using a heteroskedastic probit model for binary outcome `y2`

```
teffects ra (y2 x1 x2, hetprobit(x1 x2)) (treat2)
```

ATE of `treat2` using a Poisson model for count outcome `y3`

```
teffects ra (y3 x1 x2, poisson) (treat2)
```

ATE for each level of three-valued treatment `treat3`

```
teffects ra (y1 x1 x2) (treat3)
```

Same as above, and specify that `treat3 = 3` is the control level

```
teffects ra (y1 x1 x2) (treat3), control(3)
```

Same as above, specified using the label “MyControl” corresponding to `treat3 = 3`

```
teffects ra (y1 x1 x2) (treat3), control("MyControl")
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Regression adjustment

Statistics > Causal inference/treatment effects > Binary outcomes > Regression adjustment

Statistics > Causal inference/treatment effects > Count outcomes > Regression adjustment

Statistics > Causal inference/treatment effects > Fractional outcomes > Regression adjustment

Statistics > Causal inference/treatment effects > Nonnegative outcomes > Regression adjustment

Syntax

```
teffects ra (ovar omvarlist [ , omodel noconstant ]) (tvar) [ if ] [ in ] [ weight ]
[ , stat options ]
```

ovar is a binary, count, continuous, fractional, or nonnegative outcome of interest.
omvarlist specifies the covariates in the outcome model.
tvar must contain integer values representing the treatment levels.

omodel	Description
Model	
linear	linear outcome model; the default
logit	logistic outcome model
probit	probit outcome model
hetprobit(<i>varlist</i>)	heteroskedastic probit outcome model
poisson	exponential outcome model
flogit	fractional logistic outcome model
fprobit	fractional probit outcome model
fhetprobit(<i>varlist</i>)	fractional heteroskedastic probit outcome model

omodel specifies the model for the outcome variable.

stat	Description
Stat	
ate	estimate average treatment effect in population; the default
atet	estimate average treatment effect on the treated
pomeans	estimate potential-outcome means

options	Description
SE/Robust vce(vcetype)	vcetype may be <u>r</u> obust, <u>c</u> luster <i>clustvar</i> , <u>b</u> ootstrap, or <u>j</u> ackknife
Reporting level(#) aequations display_options	set confidence level; default is level(95) display auxiliary-equation results control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization maximize_options	control the maximization process; seldom used
Advanced control(# label) tlevel(# label) coeflegend	specify the level of <i>tvar</i> that is the control specify the level of <i>tvar</i> that is the treatment display legend instead of statistics

omvarlist may contain factor variables; see [U] 11.4.3 Factor variables.

bayesboot, bootstrap, by, collect, jackknife, and statsby are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the bootstrap prefix; see [R] bootstrap.

fweights, iweights, and pweights are allowed; see [U] 11.1.6 weight.

coeflegend does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

Model	noconstant; see [R] Estimation options.
Stat	stat is one of three statistics: ate, atet, or pomeans. ate is the default. ate specifies that the average treatment effect be estimated. atet specifies that the average treatment effect on the treated be estimated. pomeans specifies that the potential-outcome means for each treatment level be estimated.
SE/Robust	vce(vcetype) specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster <i>clustvar</i>), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [R] vce_option.
Reporting	level(#); see [R] Estimation options. aequations specifies that the results for the outcome-model or the treatment-model parameters be displayed. By default, the results for these auxiliary parameters are not displayed.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

Maximization

maximize_options: `iterate(#)`, `[no]log`, and `from(init_specs)`; see [R] [Maximize](#). These options are seldom used.

init_specs is one of

`matname [ , skip copy]`

`# [ , # ...], copy`

Advanced

`control(#|label)` specifies the level of *tvar* that is the control. The default is the first treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`. `control()` and `tlevel()` may not specify the same treatment level.

`tlevel(#|label)` specifies the level of *tvar* that is the treatment for the statistic `atet`. The default is the second treatment level. You may specify the numeric level # (a nonnegative integer) or the label associated with the numeric level. `tlevel()` may only be specified with statistic `atet`. `tlevel()` and `control()` may not specify the same treatment level.

The following option is available with `teffects ra` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

[Overview](#)

[Video example](#)

Overview

Regression adjustment (RA) estimators use the contrasts of the averages of treatment-specific predicted outcomes to estimate treatment effects. RA estimators use a two-step approach to estimating treatment effects:

1. They fit separate regression models of the outcome on a set of covariates for each treatment level.
2. They compute the averages of the predicted outcomes for each subject and treatment level. These averages reflect the POMs. The contrasts of these averages provide estimates of the ATEs. By restricting the computations of the means to the subset of treated subjects, we obtain the ATETs.

RA estimators are consistent as long as the treatment is independent of the potential outcomes after conditioning on the covariates. In fact, `teffects ra` uses an estimation technique that implements both steps at once so that we do not need to correct the standard errors in the second step to reflect the uncertainty surrounding the predicted outcomes.

We will illustrate the use of `teffects ra` by using data from a study of the effect of a mother’s smoking status during pregnancy (`mbsmoke`) on infant birthweight (`bweight`) as reported by Cattaneo (2010). This dataset also contains information about each mother’s age (`mage`), education level (`medu`), marital status (`mmarried`), whether the first prenatal exam occurred in the first trimester (`prenatal1`), and whether this baby was the mother’s first birth (`fbaby`).

➤ Example 1: Estimating the ATE

We begin by using `teffects ra` to estimate the average treatment effect of smoking, controlling for first-trimester exam status, marital status, mother’s age, and first-birth status. In Stata, we type

```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)

. teffects ra (bweight prenatal1 mmarried mage fbaby) (mbsmoke)

Iteration 0: EE criterion = 7.734e-24
Iteration 1: EE criterion = 1.196e-25

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : regression adjustment
Outcome model   : linear
Treatment model : none
```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
mbsmoke						
(Smoker						
vs						
Nonsmoker)	-239.6392	23.82402	-10.06	0.000	-286.3334	-192.945
POmean						
mbsmoke						
Nonsmoker	3403.242	9.525207	357.29	0.000	3384.573	3421.911

The average birthweight if all mothers were to smoke would be 240 grams less than the average of 3,403 grams that would occur if none of the mothers had smoked.



The previous results showed us the average amount by which infants’ weights are affected by their mothers’ decision to smoke. We may instead be interested in knowing the average amount by which the weight of babies born to smoking mothers was decreased as a result of smoking. The ATET provides us with the answer.

➤ Example 2: Estimating the ATET

To obtain the ATET rather than the ATE, we use the `atet` option:

```
. teffects ra (bweight prenatal1 mmarried mage fbaby) (mb smoke), atet
Iteration 0: EE criterion = 7.629e-24
Iteration 1: EE criterion = 2.697e-26
Treatment-effects estimation      Number of obs      =      4,642
Estimator      : regression adjustment
Outcome model  : linear
Treatment model: none
```

bweight	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATET						
mb smoke						
(Smoker						
vs						
Nonsmoker)	-223.3017	22.7422	-9.82	0.000	-267.8755	-178.7278
POmean						
mb smoke						
Nonsmoker	3360.961	12.75749	263.45	0.000	3335.957	3385.966

The average birthweight is 223 grams less when all the mothers who smoke do so than the average of 3,361 grams that would have occurred if none of these mothers had smoked.

The ATET differs from the ATE because the distribution of the covariates among mothers who smoke differs from the distribution for nonsmoking mothers. For example, in [CAUSAL] [teffects intro](#), we remarked that in our sample, mothers who smoked tended to be older than those who did not. The differing distributions of covariates also affect the estimated POMs.



By default, `teffects ra` reports the ATE, which is the difference between the two POMs in the case of a binary treatment variable. Sometimes, we want to know the estimated POMs themselves. We might also want to see the actual regression equations used to estimate the POMs. Obtaining this information is easy, as the next example illustrates.

► Example 3: Estimating the POMs

Here we use the `pomeans` option to display the POMs and the `aequations` option to display the estimated regression coefficients for the treated and untreated subjects.

```
. teffects ra (bweight prenatal1 mmarried mage fbaby) (mbsmoke),
> pomeans aequations
Iteration 0: EE criterion = 7.734e-24
Iteration 1: EE criterion = 2.850e-26
Treatment-effects estimation      Number of obs      =      4,642
Estimator      : regression adjustment
Outcome model  : linear
Treatment model: none
```

		Robust				
	bweight	Coefficient	std. err.	z	P> z	[95% conf. interval]
POMeans						
	mbsmoke					
	Nonsmoker	3403.242	9.525207	357.29	0.000	3384.573 3421.911
	Smoker	3163.603	21.86351	144.70	0.000	3120.751 3206.455
OME0						
	prenatal1	64.40859	27.52699	2.34	0.019	10.45669 118.3605
	mmarried	160.9513	26.6162	6.05	0.000	108.7845 213.1181
	mage	2.546828	2.084324	1.22	0.222	-1.538373 6.632028
	fbaby	-71.3286	19.64701	-3.63	0.000	-109.836 -32.82117
	_cons	3202.746	54.01082	59.30	0.000	3096.886 3308.605
OME1						
	prenatal1	25.11133	40.37541	0.62	0.534	-54.02302 104.2457
	mmarried	133.6617	40.86443	3.27	0.001	53.5689 213.7545
	mage	-7.370881	4.21817	-1.75	0.081	-15.63834 .8965804
	fbaby	41.43991	39.70712	1.04	0.297	-36.38461 119.2644
	_cons	3227.169	104.4059	30.91	0.000	3022.537 3431.801

The nonsmoker POM for infant birthweight is 3,403 grams; that means that if none of the women in our sample smoked during pregnancy, the expected average birthweight would be 3,403 grams. The POM if all mothers did smoke during pregnancy is 3,164 grams, a difference of 240 grams, as we established in [example 1](#). The coefficients for the equation labeled OME0 represent the linear equation used to estimate the nontreated POM, and the coefficients for the equation labeled OME1 represent the linear equation used to estimate the treated POM. The coefficients are identical to those we would obtain using `regress`, but the standard errors differ slightly because `teffects ra` does not make the small-sample adjustment that `regress` does.

◀

We often express statistics as percentages to alleviate scaling issues and aid interpretation. In the present context, we may wish to express an ATE as a percentage of the untreated POM to gain a more intuitive measure of efficacy.

► Example 4: Reporting the ATE as a percentage

Sometimes, we are interested in reporting the estimated treatment effect as a percentage of the untreated POM. We continue to use the same model as in the previous examples, but we specify the `coeflegend` option so that `teffects ra` reports the names of the parameters. Knowing the correct names to use, we can then use `nlcom` to obtain the percentage change along with its delta-method-based standard error. We type

```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. teffects ra (bweight prenatal1 mmarried mage fbaby) (mbsmoke), coeflegend
Iteration 0: EE criterion = 7.734e-24
Iteration 1: EE criterion = 1.196e-25
Treatment-effects estimation      Number of obs      =      4,642
Estimator      : regression adjustment
Outcome model  : linear
Treatment model: none
```

bweight	Coefficient	Legend
ATE		
mbsmoke		
(Smoker		
vs		
Nonsmoker)	-239.6392	_b[ATE:r1vs0.mbsmoke]
POmean		
mbsmoke		
Nonsmoker	3403.242	_b[POmean:0.mbsmoke]

```
. nlcom _b[ATE:r1vs0.mbsmoke] / _b[POmean:0.mbsmoke]
      _nl_1: _b[ATE:r1vs0.mbsmoke] / _b[POmean:0.mbsmoke]
```

bweight	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
_nl_1	-.070415	.0069245	-10.17	0.000	-.0839867	-.0568433

The average birthweight falls by an estimated 7.0% when every mother smokes relative to the case when no mothers smoke. We also obtain a 95% confidence interval of a 5.7% to 8.4% reduction.



Birthweights cannot be negative, though it is possible for a linear regression model to make negative predictions. A common way to enforce nonnegative predictions is to use an exponential conditional-mean model, which is commonly fitted using the Poisson quasimaximum likelihood estimator, as discussed in [Cameron and Trivedi \(2005, sec. 5.7\)](#), [Wooldridge \(2010, sec. 18.2\)](#), and [Pawitan \(2001, chap. 14\)](#). `teffects ra` provides an option to use this model rather than linear regression for the outcomes.

➤ Example 5: Modeling nonnegative outcomes

Now we refit our model of smoking behavior on birthweight, but we specify the poisson option in the outcome-model equation so that teffects ra uses the Poisson exponential model rather than linear regression:

```
. teffects ra (bweight prenatal1 mmarried mage fbaby, poisson) (mb smoke)
Iteration 0: EE criterion = 3.950e-17
Iteration 1: EE criterion = 1.244e-23
Treatment-effects estimation      Number of obs      =      4,642
Estimator      : regression adjustment
Outcome model  : Poisson
Treatment model: none
```

bweight		Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE	mb smoke						
	(Smoker vs Nonsmoker)	-239.6669	23.83757	-10.05	0.000	-286.3877	-192.9462
POmean	mb smoke						
	Nonsmoker	3403.178	9.526006	357.25	0.000	3384.508	3421.849

In this case, using a model that forces outcomes to be nonnegative did not make any substantive difference. In this dataset, nearly 90% of babies weigh at least 2,700 grams, and even the smallest baby weighs 340 grams. When the dependent variable is so large, the predictions from Poisson and linear regression models are remarkably similar.



We now consider models for fractional outcomes. Fractional responses concern outcomes between 0 and 1. These responses may be averaged 0/1 outcomes such as participation rates, or they may be variables that are naturally on a 0 to 1 scale such as pollution levels, patient oxygen saturation, and Gini coefficients (income inequality measures).

➤ Example 6: Modeling fractional outcomes

We will illustrate the use of `teffects ra` with the `outcome-model` option `fprobit` by using simulated data. The observations are 543 cities at least 200 miles apart. The data contain information about each city’s level of industrialization (`industrial`), average annual rainfall in millimeters (`rainfall`), whether or not the city has a metro or train (`train`), and traffic congestion measured by an index (`traffic`).

Our outcome is the level of pollution (`pollution`) measured on a 0 to 1 scale. Values of `pollution` between 0 and 0.3 have no public health implications, but values greater than 0.7 imply that people with breathing or health problems should remain indoors. We study the effect of a tax on gas-guzzler cars (`guzzler`) on air pollution. A tax that is effective in reducing pollution improves public health.

We estimate the ATE of a gas-guzzler tax on pollution, controlling for average yearly rainfall, traffic congestion, the level of industrialization, and whether the city has a train or a metro by using a fractional probit model.

```
. use https://www.stata-press.com/data/r19/pollution
(Simulated Urban Pollution Data)
. teffects ra (pollution rainfall i.traffic industrial i.train, fprobit) (guzzler)
Iteration 0: EE criterion = 3.023e-16
Iteration 1: EE criterion = 9.917e-32

Treatment-effects estimation      Number of obs      =      534
Estimator      : regression adjustment
Outcome model   : fractional probit
Treatment model : none
```

pollution	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE						
guzzler (tax vs no tax)	-.0960214	.0113896	-8.43	0.000	-.1183447	-.0736981
POMean						
guzzler no tax	.3879346	.0101733	38.13	0.000	.3679952	.407874

The POM if no city were to implement a gas-guzzler tax is an air pollution index of 0.39. If all cities implement a gas-guzzler tax, the air pollution index would decrease by 0.096 relative to a scenario where no city implements the tax.



Video example

Treatment effects: Regression adjustment

Stored results

`teffects ra` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(nj)</code>	number of observations for treatment level j
<code>e(N_clust)</code>	number of clusters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>teffects</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(subcmd)</code>	<code>ra</code>
<code>e(omodel)</code>	linear, logit, probit, <code>hetprobit</code> , <code>poisson</code> , <code>flogit</code> , <code>fprobit</code> , or <code>fhetprobit</code>
<code>e(stat)</code>	statistic estimated, <code>ate</code> , <code>atet</code> , or <code>pomeans</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

`teffects ra` implements a [smooth treatment-effects estimator](#). All smooth treatment-effects estimators are documented in [\[CAUSAL\] teffects aipw](#).

References

- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Cattaneo, M. D. 2010. Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics* 155: 138–154. <https://doi.org/10.1016/j.jeconom.2009.09.023>.
- Drukker, D. M. 2016. Exact matching on discrete covariates is the same as regression adjustment. *The Stata Blog: Not Elsewhere Classified*. <https://blog.stata.com/2016/08/16/exact-matching-on-discrete-covariates-is-the-same-as-regression-adjustment/>.
- Huber, C. 2015. Introduction to treatment effects in Stata: Part 1. *The Stata Blog: Not Elsewhere Classified*. <https://blog.stata.com/2015/07/07/introduction-to-treatment-effects-in-stata-part-1/>.
- Pawitan, Y. 2001. *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. Oxford: Oxford University Press.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

Also see

- [CAUSAL] **teffects postestimation** — Postestimation tools for teffects
- [CAUSAL] **teffects** — Treatment-effects estimation for observational data
- [ERM] **eregress** — Extended linear regression
- [U] **20 Estimation and postestimation commands**

[Description](#)
[Options](#)
[References](#)

[Quick start](#)
[Remarks and examples](#)
[Also see](#)

[Menu](#)
[Stored results](#)

[Syntax](#)
[Methods and formulas](#)

Description

`telasso` estimates the average treatment effect (ATE), the average treatment effect on the treated (ATET), and the potential-outcome means (POMs) from observational data by augmented inverse-probability weighting (AIPW) while using lasso methods to select from potential control variables to be included in the model.

AIPW estimators combine aspects of regression-adjustment and inverse-probability-weighted methods. AIPW estimators have the double-robust property.

`telasso` accepts a continuous, binary, count, or nonnegative outcome.

See [\[CAUSAL\] teffects intro](#) or [\[CAUSAL\] teffects intro advanced](#) for more information about estimating treatment effects from observational data. See [\[LASSO\] Lasso inference intro](#) for more information on estimating coefficients and standard errors for a subset of variables while using lasso methods to select from a set of control variables.

Quick start

ATE of binary treatment `treat` using a linear model for outcome `y1` on `x1–x100` and a logistic model for `treat` on `w1–w100`; use lassos to select variables from `x1–x100` for the outcome model and from `w1–w100` for the treatment model

```
telasso (y1 x1-x100) (treat w1-w100)
```

Same as above, but estimate the ATET

```
telasso (y1 x1-x100) (treat w1-w100), atet
```

Use a Poisson model for count outcome `y2`

```
telasso (y2 x1-x100, poisson) (treat w1-w100)
```

Same as above, but use a probit model for `treat`

```
telasso (y2 x1-x100, poisson) (treat w1-w100, probit)
```

Use BIC instead of a plugin iterative formula to select the optimal λ^* in each lasso

```
telasso (y1 x1-x100) (treat w1-w100), selection(bic)
```

Perform cross-fitting with five folds

```
telasso (y1 x1-x100) (treat w1-w100), xfolds(5)
```

Same as above, but repeat the cross-fitting procedure 15 times and average the results

```
telasso (y1 x1-x100) (treat w1-w100), xfolds(5) resample(15)
```

Use BIC to select covariates in the lasso for `y1` for treatment level 1 only

```
telasso (y1 x1-x100, lasso(1, selection(bic))) (treat w1-w100)
```

Use cross-validation (CV) for the lasso for `treat` only

```
telasso (y1 x1-x100) (treat w1-w100, lasso(selection(cv)))
```

Menu

- Statistics > Causal inference/treatment effects > Continuous outcomes > AIPW with lasso covariate selection
- Statistics > Causal inference/treatment effects > Binary outcomes > AIPW with lasso covariate selection
- Statistics > Causal inference/treatment effects > Count outcomes > AIPW with lasso covariate selection
- Statistics > Causal inference/treatment effects > Nonnegative outcomes > AIPW with lasso covariate selection

Syntax

```
telasso (ovar omvarlist [ , omodel om_options ])  
        (tvar tmvarlist [ , tmodel tm_options ]) [ if ] [ in ] [ weight ] [ , stat options ]
```

ovar is a binary, count, continuous, or nonnegative outcome of interest.
omvarlist specifies the covariates in the outcome model.
tvar must contain a binary value representing the treatment levels.
tmvarlist specifies the covariates in the treatment model.

omodel	Description
Model	
linear	linear outcome model; the default
logit	logistic outcome model
probit	probit outcome model
poisson	exponential-mean outcome model

om_options	Description
Model	
lasso([#,] lasso_options)	specify options for the lassos for <i>ovar</i> at treatment level #; may be repeated
* sqrtlasso	use square-root lassos instead of lassos for <i>ovar</i>
* sqrtlasso([#,] lasso_options)	specify options for the square-root lassos for <i>ovar</i> at treatment level #; may be repeated
noconstant	suppress constant term
ainclude(varlist)	specify variables that should always be included in the outcome model
noselection	suppress model selection for the outcome model
* exposure(varname_e)	include ln(<i>varname_e</i>) in model with coefficient constrained to 1
*sqrtlasso() may be specified only when <i>omodel</i> is linear. exposure() may be specified only when <i>omodel</i> is poisson.	

tmodel	Description
Model	
logit	logistic treatment model; the default
probit	probit treatment model

<i>tm_options</i>	Description
Model	
<code>lasso(<i>lasso_options</i>)</code>	specify options for the lasso for <i>tvar</i>
<code>noconstant</code>	suppress constant term
<code>ainclude(<i>varlist</i>)</code>	specify variables that should always be included in the treatment model
<code>noselection</code>	suppress model selection for the treatment model
<i>stat</i>	Description
Stat	
<code>ate</code>	estimate average treatment effect in population; the default
<code>atet</code>	estimate average treatment effect on the treated
<code>pomeans</code>	estimate potential-outcome means
<i>options</i>	Description
Selection	
<code>selection(plugin)</code>	use a plugin iterative formula to select an optimal value of the lasso penalty parameter λ^* for each lasso; the default
<code>selection(cv)</code>	use CV to select an optimal value of the lasso penalty parameter λ^* for each lasso
<code>selection(adaptive)</code>	use adaptive lasso to select an optimal value of the lasso penalty parameter λ^* for each lasso
<code>selection(bic)</code>	use BIC to select an optimal value of the lasso penalty parameter λ^* for each lasso
<code>xfolds[(#)]</code>	use # folds for cross-fitting
<code>resample[(#)]</code>	repeat sample splitting # times and average results
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>robust</code> or <code>cluster</code> <i>clustvar</i>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
<code>rseed(#)</code>	set random-number seed
<code>verbose</code>	display a verbose iteration log
<code>[no]log</code>	display or suppress an iteration log

Advanced

<code>pstolerance(#)</code>	set tolerance for overlap assumption
<code>osample(newvar)</code>	<i>newvar</i> identifies observations that violate the overlap assumption
<code>control(# label)</code>	specify the level of <i>tvar</i> that is the control
<code>reestimate</code>	refit the model after using <code>lassoselect</code> to select a different λ^*
<code>noheader</code>	do not display the header on the coefficient table
<code>coeflegend</code>	display legend instead of statistics

omvarlist and *tmvarlist* may contain factor variables; see [U] 11.4.3 Factor variables.

`collect` is allowed; see [U] 11.1.10 Prefix commands.

Default weights are not allowed. `iweights` are allowed when `selection(plugin)`, `selection(adaptive)`, `selection(cv)`, or `selection(bic)` is specified. `fweights` are allowed when `selection(plugin)` or `selection(bic)` is specified. See [U] 11.1.6 weight.

`reestimate`, `noheader`, and `coeflegend` do not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

Model

Options specific to the outcome model (*om_options*) are the following:

`lasso([#,] lasso_options)` lets you set different options for different lassos or set advanced options for all lassos for the outcome model. Specify a `#` followed by the options to apply settings to the lasso of *ovar* at treatment level `#` only. This option is repeatable as long as a different treatment level is given in each specification. If `#` is not specified, the lasso options apply to lassos of *ovar* at all treatment levels. *lasso_options* are `selection()`, `grid()`, `stop()`, `cvtolerance()`, `bictolerance()`, `tolerance()`, and `dtolerance()`; see [LASSO] lasso options. When `lasso([#,] selection())` is specified, it overrides any global `selection()` option for the outcome variable at the specified treatment level.

`sqrlasso` specifies that square-root lassos rather than lassos should be used in selecting variables in models for the outcome at both treatment levels. This option is not allowed if *omodel* is `logit`, `probit`, or `poisson`.

`sqrlasso([#,] lasso_options)` works like the option `lasso()`, except square-root lassos for the outcome variable are done rather than regular lassos. This option is not allowed if *omodel* is `logit`, `probit`, or `poisson`. Specify a `#` followed by the options to apply settings to the lasso of *ovar* at treatment level `#` only. This option is repeatable as long as a different treatment level is given in each specification. If `#` is not specified, the lasso options apply to lassos of *ovar* at all treatment levels. *lasso_options* are `selection()`, `grid()`, `stop()`, `cvtolerance()`, `bictolerance()`, `tolerance()`, and `dtolerance()`; see [LASSO] lasso options. When `sqrlasso([#,] selection())` is specified, it overrides any global `selection()` option for the outcome variable at the specified treatment level.

`noconstant` suppresses the constant term in the outcome model.

`ainclude(varlist)` specifies variables that are always included in the outcome model. Lassos will choose to include or exclude other variables in *omvarlist*.

`noselection` specifies not to perform model selection in the outcome model.

`exposure(varnamee)` specifies a variable that reflects the amount of exposure over which the dependent variable events were observed for each observation; $\ln(\text{varname}_e)$ with coefficient constrained to be 1 is entered into the log-link function. This option may be specified only when `omodel` is `poisson`.

Options specific to the treatment model (`tm_options`) are the following:

`lasso(lasso_options)` lets you set advanced options for the lasso for the treatment model. *lasso_options* are `selection()`, `grid()`, `stop()`, `cvtolerance()`, `bictolerance()`, `tolerance()`, and `dtolerance()`; see [LASSO] [lasso options](#). When `lasso(selection())` is specified, it overrides any global `selection()` option for the treatment variable.

`noconstant` suppresses the constant term in the treatment model.

`ainclude(varlist)` specifies variables that are always included in the treatment model. Lassos will choose to include or exclude other variables in *tmvarlist*.

`noselection` specifies not to perform model selection in the treatment model.

Selection

`selection(plugin | cv | adaptive | bic)` specifies the selection method for choosing an optimal value of the lasso penalty parameter λ^* for each lasso or square-root lasso estimation. Separate lassos are estimated for the outcome variable at each treatment level and for the treatment variable. Specifying `selection()` changes the selection method for all these lassos. You can specify different selection methods for different lassos by using `lasso()` or `sqrtlasso()` in the outcome model specification or the option `lasso()` in the treatment model specification. When `lasso()` or `sqrtlasso()` is used to specify a different selection method for the lassos of outcome or treatment variables, they override the global setting made using `selection()` for the specified model.

`selection(plugin)` is the default. It selects λ^* based on a “plugin” iterative formula dependent on the data. See [LASSO] [lasso options](#).

`selection(cv)` selects the λ^* that gives the minimum of the CV function. See [LASSO] [lasso options](#).

`selection(adaptive)` selects λ^* using the adaptive lasso selection method. It cannot be specified when `sqrtlasso()` is specified for the outcome model. See [LASSO] [lasso options](#).

`selection(bic)` selects the λ^* that gives the minimum of the BIC function. See [LASSO] [lasso options](#).

`xfolds[(#)]` specifies the number of folds for cross-fitting. Not specifying `xfolds` or `xfolds(#)` is equivalent to specifying `xfolds(1)`. In other words, by default no cross-fitting is done. Specifying `xfolds` alone is equivalent to specifying `xfolds(10)`; that is, cross-fitting is done by randomly dividing the original data into 10 folds.

`resample[(#)]` specifies that sample splitting be repeated and results averaged. This reduces the effects of the randomness of sample splitting on the estimated coefficients. Not specifying `resample` or `resample(#)` is equivalent to specifying `resample(1)`. In other words, by default no resampling is done. Specifying `resample` alone is equivalent to specifying `resample(10)`; that is, sample splitting is repeated 10 times. For each sample split, lassos are computed. So when this option is not specified, lassos are repeated `xfolds(#)` times. But when `resample(#)` is specified, lassos are repeated `xfolds(#)` $\times$ `resample(#)` times. Thus, while we recommend using `resample` to get final results, note that it can be an extremely time-consuming procedure.

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`) and that allow for intragroup correlation (`cluster clustvar`); see [R] [vce_option](#).

Reporting

`level(#)`; see [R] [Estimation options](#).

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no stretch`; see [R] [Estimation options](#).

Optimization

`rseed(#)` sets the random-number seed. This option can be used to reproduce results. `rseed(#)` is equivalent to typing `set seed #` prior to running `telasso`. Random numbers are used to produce split samples for cross-fitting. So for all `selection()` options, if you want to reproduce your results, you must either use this option or use `set seed`. See [R] [set seed](#).

`verbose` displays a verbose log showing the iterations of each lasso estimation. This option is useful when doing `selection(cv)` or `selection(adaptive)`. It allows monitoring the progress of the lasso estimations for these selection methods, which can be time consuming when there are many variables specified in `omvarlist` or `tmvarlist`.

`[no]log` displays or suppresses a log showing the progress of the estimation. By default, one-line messages indicating when each lasso estimation begins are shown. Specify `verbose` to see a more detailed log.

Advanced

`pstolerance(#)` specifies the tolerance used to check the overlap assumption. The default value is `pstolerance(1e-5)`. `telasso` will exit with an error if an observation has an estimated propensity score smaller than that specified by `pstolerance()`.

`osample(newvar)` specifies that indicator variable `newvar` be created to identify observations that violate the overlap assumption.

`control(#|label)` specifies the level of `tvar` that is the control. The default is the first treatment level. You may specify the numeric level `#` (a nonnegative integer) or the label associated with the numeric level. `control()` may not be specified with statistic `pomeans`.

The following options are available with `telasso` but are not shown in the dialog box:

`reestimate` is an advanced option that refits the `telasso` model based on changes made to the underlying lassos using `lassoselect`. After running `telasso`, you can select a different λ^* for one or more of the lassos estimated by `telasso`. After selecting λ^* , you type `telasso`, `reestimate` to refit the `telasso` model based on the newly selected λ 's.

`reestimate` may be combined only with reporting options.

`noheader` prevents the coefficient table header from being displayed.

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

Remarks are presented under the following headings:

Overview

Estimating the ATE with lassos for covariate selection

Choosing the tuning parameter

Estimating the ATET

High-dimensional semiparametric models

Overview

`telasso` estimates treatment effects while using lasso techniques to select the variables in the model. Like the `lasso for inference` commands, inferences drawn about the ATEs, ATETs, and POMs reported by `telasso` are robust to model selection mistakes made by lasso. In addition, like the `teffects aipw` command, the estimated effects are consistent if the functional form of only one of the outcome or treatment model is correctly specified. Thus, `telasso` is a version of the AIPW estimator that also allows for covariate selection via lasso and is robust to functional form misspecification in one of the models as well as errors in selection of covariates by lasso. It is, however, required that the true covariates are a subset of the specified list of potential covariates. See [\[CAUSAL\] teffects intro](#) or [\[CAUSAL\] teffects intro advanced](#) for an introduction of treatment effects estimation. See [\[CAUSAL\] teffects aipw](#) for more information on the AIPW estimator. See [\[LASSO\] Lasso intro](#) for an introduction to lasso. See [\[LASSO\] Lasso inference intro](#) for an introduction to making inference after using lasso for model selection.

Researchers who have datasets with many variables, sometimes more than observations, face difficult decisions. They cannot use all the variables in their dataset as covariates in a model, and therefore, they may want to use a model selection method such as lasso to select covariates. In addition, rather than simply allowing covariates to enter a model linearly, researchers may want to fit a more flexible and realistic model to their data by including higher-order terms, interactions, spline basis functions, and the like, but this can also produce more terms than can be included in a model and require a method such as lasso for selection. The lasso methods used in `telasso` allow for making valid inference after selecting from among the potential covariates.

In the context of treatment-effect estimation, why do we need model selection? It is because of the intrinsic conflicts between two crucial assumptions used to identify treatment effects: conditional independence (CI) and overlap assumptions. CI means that, dependent on a set of control variables, the potential outcome is independent of the treatment assignment. The more variables there are in the model, the more plausible it is that the CI assumption is satisfied. On the other hand, the overlap assumption implies that there is always a positive probability that any given unit is treated or untreated. The fewer variables there are in the model, the more comfortable we can be that the overlap assumption is satisfied. To summarize, the conflict is that the CI assumption expects many variables in the model, but the overlap assumption expects few variables. By including many potential variables in the model and allowing lasso to select from among them, we can reconcile this conflict. For a more detailed discussion, see [Farrell \(2015\)](#) and [Chernozhukov et al. \(2018\)](#).

Model selection, however, does not come for free. If researchers use model selection but conduct inference ignoring the fact that they did model selection, the inference results are possibly wrong. This is because model selection techniques make mistakes. Making inferences without considering the variability in model selection is a dangerous practice (see [Leeb and Pötscher \[2005\]](#) and [Leeb and Pötscher \[2006\]](#)). Instead, the inference should be robust to model selection mistakes.

Recently, [Farrell \(2015\)](#) and [Chernozhukov et al. \(2018\)](#) proposed that modern model selection techniques, such as lasso methods for inference, be combined with the doubly robust AIPW estimator in [Rosenbaum and Rubin \(1983\)](#) to estimate the ATEs. The intuition behind this method is twofold. On one hand, lasso techniques resolve the conflicts between the CI and overlap assumptions. Although CI assumption expects many variables, we only need the covariates that matter for the outcome. If lasso selects a sparse model, the overlap assumption is more plausible to be satisfied. On the other hand, the proposed estimator should guard against model selection mistakes. The doubly robust AIPW estimator happens to satisfy this property. Double robustness also means that the estimates of treatment effects are consistent if either the outcome model or the treatment model is correctly specified.

`telasso` implements estimators in [Chernozhukov et al. \(2018\)](#) for the ATEs, ATETs, and POMs from observational data.

The `telasso` estimator uses a five-step approach to estimating ATE:

1. It uses lasso techniques to select variables in the outcome model for each treatment level.
2. Based on the selected variables in step 1, it fits separate regression models of the outcome for each treatment level and obtains the treatment-specific predicted outcomes for each subject.
3. It uses lasso techniques to select variables in the treatment model.
4. Based on the selected variables in step 3, it estimates the parameters of the treatment model and computes inverse-probability weights.
5. It computes the weighted means of the treatment-specific predicted outcomes, where the weights are the inverse-probability weights computed in step 4. The contrasts of these weighted averages provide the estimates of the ATE.

Steps 1 and 3 perform the model selection for the outcome and treatment models, respectively. Using the selected variables, steps 2, 4, and 5 construct moment conditions to estimating ATEs. The resulting estimator is consistent under CI, overlap, and independent and identically distributed assumptions. The inference is robust to the mild model selection mistakes that could happen in steps 1 and 3. This estimator is also robust to model misspecification in either the outcome or the treatment model because of the double robust moment condition used in step 5.

`telasso` also implements the double machine learning estimator in [Chernozhukov et al. \(2018\)](#) for ATEs, ATETs, and POMs. Double machine learning relaxes the sparsity assumption needed for lasso methods. The sparsity assumption implies that lasso can only have good properties if it selects a few variables from a potentially large number of candidates. Double machine learning allows `telasso` to select more variables in the models and still be valid.

`telasso` allows the outcome variable to be modeled using a linear, logistic, probit, or Poisson model. For a linear outcome model, square-root lasso can be used instead of lasso in step 1.

Estimating the ATE with lassos for covariate selection

In some cases, we want to estimate a treatment effect when we have several variables in our dataset that, when interacted, create a large set of candidate covariates in our model.

► Example 1: ATE of bilateral lung transplant

We first illustrate `telasso` with an example that compares two types of lung transplants. Bilateral lung transplant (BLT) is usually associated with a higher death rate in the short term after the operation but with a more significant improvement in the quality of life compared with the single lung transplant (SLT).

As a result, for patients who need to decide between these two treatment options, knowing the effect of BLT (versus SLT) on quality of life is essential. We can measure quality of life based on an individual's forced expiratory volume in one second (FEV1).

We have a fictional dataset (`lung.dta`) inspired by [Koch, Vock, and Wolfson \(2018\)](#). The outcome (`fev1p`) is FEV1% measured one year after the operation. FEV1% is the percentage of FEV1 that the patient has relative to a healthy person with similar characteristics. The treatment variable (`transtype`) indicates whether the treatment is BLT or SLT. To open the dataset and describe it, we type

```
. use https://www.stata-press.com/data/r19/lung
(Fictional data on lung transplant)
```

```
. describe *, short
```

Variable name	Storage type	Display format	Value label	Variable label
agep	byte	%10.0g		Patient age (years)
bmip	double	%10.0g		Patient body mass index
diabetesp	byte	%12.0g	lbdiab	Patient diabetes status
heightp	double	%10.0g		Patient height (cm)
o2amt	double	%10.0g		Oxygen delivered
karn	byte	%8.0g	lbyes	Karnofsky score > 60
lungals	double	%10.0g		Lung allocation score
racep	byte	%8.0g	lbrace	Patient race
sexp	byte	%8.0g	lbsex	Patient gender
lifesvent	byte	%8.0g	lbyes	Life support ventilator needed
assisvent	byte	%8.0g	lbyes	Assisted ventilation needed
centervol	double	%10.0g		Center volume
walkdist	double	%10.0g		Walking distance in 6 minutes
o2rest	byte	%8.0g	lbyes	Oxygen needed at rest
aged	byte	%10.0g		Donor age (years)
raced	byte	%8.0g	lbrace	Donor race
bmld	double	%10.0g		Donor body mass index
smoked	byte	%8.0g	lbyes	Donor if has history of smoking
cmv	byte	%8.0g	lbyes	Positive cytomegalovirus test
deathcause	byte	%8.0g	lbyes	Cause of death - traumatic brain injury
diabetesd	byte	%12.0g	lbdiab	Donor diabetes status
expandd	byte	%8.0g	lbyes	Expanded donor needed
heightd	double	%10.0g		Donor height (cm)
sexd	byte	%8.0g	lbsex	Donor gender
distd	int	%10.0g		Donor to treatment center distance
lungpo2	double	%10.0g		Lung P02
lungalloc	byte	%8.0g	lballo	Lung allocation status
hratio	double	%10.0g		Height ratio
ischemict	double	%10.0g		Ischemic time
genderm	byte	%19.0g	lbgm	Matching gender status
racem	byte	%17.0g	lbrm	Matching race status
transtype	byte	%8.0g	lbttau	Lung transplant type
fev1p	double	%10.0g		Percentage of predicted value of FEV1

Thirty-one variables measure some characteristics of the patients and donors. To construct control variables, we want to use these variables and the interactions among them. It would be tedious to type these variable names one by one to distinguish between continuous and categorical variables. `v1` is a suite of commands that simplifies this process.

First, we use `vl set` to automatically partition the variables into continuous and categorical variables. The global macro `$vlcategorical` contains all the categorical variable names, and `$vlcontinuous` contains all the continuous variable names.

```
. vl set
```

Macro	Macro's contents	
	# Vars	Description
System		
<code>\$vlcategorical</code>	18	categorical variables
<code>\$vlcontinuous</code>	13	continuous variables
<code>\$vluncertain</code>	2	perhaps continuous, perhaps categorical variables
<code>\$vlother</code>	0	all missing or constant variables

Notes

1. Review contents of `vlcategorical` and `vlcontinuous` to ensure they are correct. Type `vl list vlcategorical` and type `vl list vlcontinuous`.
2. If there are any variables in `vluncertain`, you can reallocate them to `vlcategorical`, `vlcontinuous`, or `vlother`. Type `vl list vluncertain`.
3. Use `vl move` to move variables among classifications. For example, type `vl move (x50 x80) vlcontinuous` to move variables `x50` and `x80` to the continuous classification.
4. `vl`names are global macros. Type the `vl`name without the leading dollar sign (\$) when using `vl` commands. Example: `vlcategorical not $vlcategorical`. Type the dollar sign with other Stata commands to get a varlist.

```
. display "$vlcontinuous"
bmip heightp o2amt lungals centervol walkdist bmid heightd distd lungpo2
> hratio ischemict fev1p
. display "$vlcategorical"
diabetesp karn racep sexp lifesvent assisvent o2rest raced smoked cmv
> deathcause diabetesd expandd sexd lungalloc genderm racem transtype
```

Second, we use `vl create` to create customized variable lists. Specifically, `$cvars` contains all the continuous variables except the outcome (`fev1p`), and `$fvars` consists of all the categorical variables except the treatment (`transtype`). Finally, `vl sub` substitutes the global macro `$allvars` with the full second-order interaction between the continuous variables in `$cvars` and categorical variables in `$fvars`. We will use `$allvars` as the control variables for both outcome model and treatment model.

```
. vl create cvars = vlcontinuous - (fev1p)
note: $cvars initialized with 12 variables.
. vl create fvars = vlcategorical - (transtype)
note: $fvars initialized with 17 variables.
. vl sub allvars = c.cvars i.fvars c.cvars#i.fvars
```


Now we are ready to use `telasso` to estimate the ATEs. We assume a linear outcome model and a logit treatment model, the defaults for `telasso`. We type

```
. telasso (fevip $allvars) (transtype $allvars)
Estimating lasso for outcome fevip if transtype = 0 using plugin method ...
Estimating lasso for outcome fevip if transtype = 1 using plugin method ...
Estimating lasso for treatment transtype using plugin method ...
Estimating ATE ...

Treatment-effects lasso estimation      Number of observations      =      937
Outcome model: linear                  Number of controls          =      454
Treatment model: logit                 Number of selected controls =       8
```

fevip	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE transtype (BLT vs SLT)	37.51841	.1606703	233.51	0.000	37.20351	37.83332
POmean transtype SLT	46.4938	.2021582	229.99	0.000	46.09757	46.89002

The FEV1% if all the patients were to choose BLT is expected to be 38 percentage points higher than the 46% average expected if all patients were to choose a SLT. Among the 454 control variables, `telasso` selects only 8 of them. To summarize the model selection for both the outcome and the treatment models, we can use `lassoinfo`.

```
. lassoinfo
      Estimate: active
      Command: telasso
```

Variable	Model	Selection		No. of selected variables
		Method	lambda	
fevip				
transt~e ~0	linear	plugin	.2239121	5
transt~e ~1	linear	plugin	.1986153	6
transtype	logit	plugin	.0748279	3

If we want to see which variables are selected by each lasso, we can use `lassocoeff`. Notice that there are two lassos associated with the outcome of `fev1p`. One is for outcome `fev1p` when the treatment `transtype` is 0, and the other is for `fev1p` when `transtype` is 1. So, in `lassocoeff`, we need to use the options `tlevel()` and `for()` to refer to the lasso for the outcome variable at a specified treatment level. In contrast, for the treatment of `transtype`, there is only one lasso; therefore, we only need to use option `for()` to specify the lasso for the treatment variable.

```
. lassocoef (., for(fev1p) tlevel(0)) (., for(fev1p) tlevel(1))
> (., for(transtype))
```

	fev1p(0)	fev1p(1)	transtype
heightp	x	x	
centervol	x	x	
walkdist	x	x	x
lungpo2	x	x	x
diabetesd#c.lungpo2 0	x		
diabetesp#c.walkdist 0		x	
assisvent#c.walkdist 0		x	
ischemict _cons	x	x	x

Legend:

```
b - base level
e - empty cell
o - omitted
x - estimated
```

The lassos selected `walkdist` and `lungpo2` in all three models, while `heightp`, `centervol`, `0.diabetesd#c.lungpo2`, `0.diabetesp#c.walkdist`, `0.assisvent#c.walkdist`, and `ischemict` were each selected as covariates in one or two models.

Choosing the tuning parameter

By default, `telasso` uses a plugin method to choose the tuning parameter λ in the lasso steps. We can also use BIC, cross-validation, or adaptive lasso to select the optimal λ .

► Example 2: Choosing λ via BIC

Here we use the `selection(bic)` option to select λ by minimizing BIC.

```
. telasso (fevip $allvars) (transtype $allvars), selection(bic)
Estimating lasso for outcome fevip if transtype = 0 using BIC ...
Estimating lasso for outcome fevip if transtype = 1 using BIC ...
Estimating lasso for treatment transtype using BIC ...
Estimating ATE ...

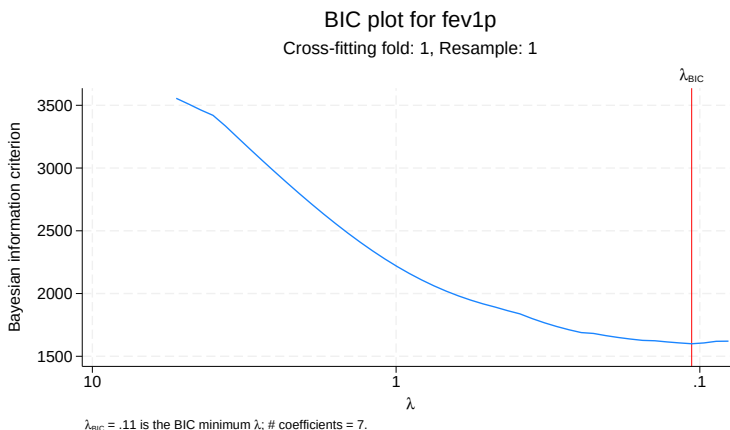
Treatment-effects lasso estimation      Number of observations      =      937
Outcome model: linear                   Number of controls          =      454
Treatment model: logit                  Number of selected controls =      18
```

fevip	Robust		z	P> z	[95% conf. interval]	
	Coefficient	std. err.				
ATE transtype (BLT vs SLT)	37.54872	.2222001	168.99	0.000	37.11322	37.98423
POmean transtype SLT	46.44739	.2282797	203.47	0.000	45.99997	46.89481

We can interpret the estimation results in a similar way as in [example 1](#). The FEV1% if all the patients were to choose BLT is expected to be 38 percentage points higher than the average of 46% that would be expected if all patients were to choose a SLT. This result is similar to example 1, where we used the default plugin method to select the tuning parameter. However, among 454 controls, `telasso` with BIC selects 18 of them, which is more than the plugin method selected.

We can use `bicplot` for a global view about how the BIC function changes as the λ 's change. Here we show the BIC plot for the outcome `fev1p` when treatment `transtype` is 1. In `bicplot`, we use options `for()` and `tlevel()` to refer to the lasso for the outcome variable at a specified treatment level.

```
. bicplot, for(fev1p) tlevel(1)
```



From this plot, we can see that the optimal $\lambda = 0.11$ is chosen by the minimum BIC. If we want to investigate this further, we can use `lassoknots` to see which variables are selected or dropped for each λ .

```
. lassoknots, for(fev1p) tlevel(1) display(nonzero var bic)
```

ID	lambda	No. of nonzero coef.	BIC	Variables (A)dded, (R)emoved, or left (U)nchanged
2	4.823808	1	3510.118	A walkdist
5	3.649034	2	3335.87	A lungpo2
28	.4294227	3	1863.595	A heightp
29	.391274	4	1838.266	A centervol
35	.2239014	6	1682.065	A 0.diabetesp#c.walkdist 0.assisvent#c.walkdist
40	.1406166	7	1623.74	A heightd
* 43	.1063713	7	1599.977	U
44	.0969216	9	1606.419	A 0.karn#c.walkdist 0.raced#c.lungpo2
45	.0883113	12	1619.473	A 0.sexd#c.centervol 1.racep#c.centervol 0.deathcause#c.centervol
46	.080466	13	1620.666	A 1.lungalloc#c.lungpo2

* lambda selected by Bayesian information criterion.

We see that the 43rd λ , with value 0.1064, minimizes the BIC function, and there are seven selected variables at this λ .

Estimating the ATET

Sometimes, we want to estimate the ATETs to determine the effect on those who actually received the treatment.

► Example 3: ATET for BLT

We use the `atet` option to estimate the ATETs. We type

```
. telasso (fev1p $allvars) (transtype $allvars), atet
Estimating lasso for outcome fev1p if transtype = 0 using plugin method ...
Estimating lasso for outcome fev1p if transtype = 1 using plugin method ...
Estimating lasso for treatment transtype using plugin method ...
Estimating ATET ...

Treatment-effects lasso estimation      Number of observations      =      937
Outcome model: linear                  Number of controls          =      454
Treatment model: logit                 Number of selected controls =       8
```

	fev1p	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET transtype (BLT vs SLT)		35.78157	.1831478	195.37	0.000	35.42261	36.14053
POmean transtype SLT		43.35214	.2190783	197.88	0.000	42.92275	43.78153

For the patients who have a BLT, we expect the average FEV1% to be 36 percentage points higher than if all of them choose an SLT.

◀

High-dimensional semiparametric models

Sometimes, the theory suggests that some variables are essential to control the confounders, but it is silent on the functional form of the model. In the next example, we will illustrate how to use `telasso` to estimate the ATE in a high-dimensional semiparametric model.

► Example 4: ATE of 401(k) eligibility

We want to estimate the effect of 401(k) eligibility (`e401`) on net financial assets (`asset`) using data reported by [Chernozhukov and Hansen \(2004\)](#). These data are from a sample of households in the 1990 Survey of Income and Program Participation (SIPP). The data contain information on the head of the household: income (`income`), age (`age`), years of education (`educ`), whether to receive pension benefit (`pension`), marital status (`married`), and whether to participate in the IRA (`ira`).

One concern when determining the effect of 401(k) eligibility on financial assets is that choosing to work for a company that offers a 401(k) plan is not randomly assigned. To overcome this issue, [Poterba, Venti, and Wise \(1994, 1995\)](#) propose that after conditioning on income, we can take working for a company that offers a 401(k) as exogenous.

We want to include income and age as covariates in the model, but we do not want to assume that they enter the model linearly. Because we want a more flexible model, we use `makespline` to create B-splines of order 3 with 3 knots at percentiles of variables income and age. The generated variables will be used to form a semiparametric model. These variables have the common stub `_bs` for easy reference later.

```
. use https://www.stata-press.com/data/r19/assets, clear
(Excerpt from Chernozhukov and Hansen (2004))
. makespline bspline income age, basis(_bs) knots(3)
```

We describe the generated B-spline variables. Seven terms are generated for each of the original variables.

```
. describe _bs*
```

Variable name	Storage type	Display format	Value label	Variable label
_bs_1_1	double	%10.0g		B-spline basis term 1 for income
_bs_1_2	double	%10.0g		B-spline basis term 2 for income
_bs_1_3	double	%10.0g		B-spline basis term 3 for income
_bs_1_4	double	%10.0g		B-spline basis term 4 for income
_bs_1_5	double	%10.0g		B-spline basis term 5 for income
_bs_1_6	double	%10.0g		B-spline basis term 6 for income
_bs_1_7	double	%10.0g		B-spline basis term 7 for income
_bs_2_1	double	%10.0g		B-spline basis term 1 for age
_bs_2_2	double	%10.0g		B-spline basis term 2 for age
_bs_2_3	double	%10.0g		B-spline basis term 3 for age
_bs_2_4	double	%10.0g		B-spline basis term 4 for age
_bs_2_5	double	%10.0g		B-spline basis term 5 for age
_bs_2_6	double	%10.0g		B-spline basis term 6 for age
_bs_2_7	double	%10.0g		B-spline basis term 7 for age

Next, we define the control variables as the categorical variables (pension, married, and ira), the generated spline variables (`_bs*`), and all interactions among these variables. The global macro `$controls` contains the defined control variables, and we will use it in both the outcome and the treatment model.

```
. global vars c.(_bs*) i.(pension married ira)
. global controls $vars ($vars)#($vars)
```

Now, we are ready to use `telasso` to estimate the ATEs of 401(k) eligibility on net financial assets. We use a linear nonparametric series to approximate the outcome model. Moreover, we use a logit nonparametric series to approximate the treatment model. The global macro `$controls` defines terms in the nonparametric series.

```
. telasso (assets $controls) (e401 $controls), rseed(111)
Estimating lasso for outcome assets if e401k = 0 using plugin method ...
Estimating lasso for outcome assets if e401k = 1 using plugin method ...
Estimating lasso for treatment e401k using plugin method ...
Estimating ATE ...

Treatment-effects lasso estimation      Number of observations      =      9,913
Outcome model:  linear                  Number of controls          =       221
Treatment model: logit                  Number of selected controls =        47
```

assets	Robust					
	Coefficient	std. err.	z	P> z	[95% conf. interval]	
ATE						
e401k						
(Eligible						
vs						
Not elig..)	8395.301	1159.093	7.24	0.000	6123.521	10667.08
P0mean						
e401k						
Not eligi..	13849.61	813.9608	17.02	0.000	12254.28	15444.94

The net financial assets if all the workers work for the companies with a 401(k) plan is expected to be \$8,395 more than the average of \$13,850 that is expected if all the workers work for the companies without a 401(k) plan.

◀

Stored results

telasso stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(NO)</code>	number of observations for treatment level 0
<code>e(N1)</code>	number of observations for treatment level 1
<code>e(N_clust)</code>	number of clusters
<code>e(k_omvars)</code>	number of potential control variables in the outcome model
<code>e(k_omvars_sel)</code>	number of selected control variables in the outcome model
<code>e(k_tmvars)</code>	number of potential control variables in the treatment model
<code>e(k_tmvars_sel)</code>	number of selected control variables in the treatment model
<code>e(k_controls)</code>	number of potential control variables in the outcome and treatment models
<code>e(k_controls_sel)</code>	number of selected control variables in the outcome and treatment models
<code>e(k_levels)</code>	number of levels in treatment variable
<code>e(n_xfolds)</code>	number of folds for cross-fitting
<code>e(n_resample)</code>	number of resamples
<code>e(treated)</code>	level of treatment variable defined as treated
<code>e(control)</code>	level of treatment variable defined as control
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	telasso
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(tvar)</code>	name of treatment variable
<code>e(tmodel)</code>	logit or probit
<code>e(omodel)</code>	linear, logit, probit, or poisson
<code>e(omvars)</code>	potential control variables in the outcome model
<code>e(omvars_sel)</code>	selected control variables in the outcome model

<code>e(tmvars)</code>	potential control variables in the treatment model
<code>e(tmvars_sel)</code>	selected control variables in the treatment model
<code>e(stat)</code>	statistic estimated: ate, atet, or pomeans
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(tlevels)</code>	levels of treatment variable
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(rngstate)</code>	random-number state used
<code>e(properties)</code>	b V
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(select_cmd)</code>	program used to implement <code>lassoselect</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>
Matrices	
<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators
Functions	
<code>e(sample)</code>	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

Methods and formulas are presented under the following headings:

The model
Neyman orthogonal moments
Double machine learning
Resampling the partitions

The model

We consider estimating the ATEs, ATETs, and POMs when the treatment effects are heterogeneous and the treatment is binary. For notational simplicity, we drop the subscript i indicating the i th observation.

The outcome model is

$$y = g_0(\tau, \mathbf{x}) + u \quad E(u|\tau, \mathbf{x}) = 0$$

where y is the outcome variable, τ is the binary treatment variable, $\mathbf{x}$ are potentially high-dimensional control variables in the outcome model, and $g_0(\tau, \mathbf{x})$ is the expected potential outcome given a level of treatment and covariates $\mathbf{x}$. Because each unit can only be treated or not treated, the observed outcome y can only be one of $g_0(1, \mathbf{x}) + u$ or $g_0(0, \mathbf{x}) + u$.

The treatment model is

$$\tau = m_0(\mathbf{z}) + v \quad E(v|\mathbf{z}) = 0$$

where $\mathbf{z}$ are potentially high-dimensional covariates in the treatment model. $m_0(\mathbf{z})$ is the expected value of τ given $\mathbf{z}$. In other words, $m_0(\mathbf{z})$ is the probability of a unit getting treated given $\mathbf{z}$.

The parameter of interest θ_0 is ATE, ATET, or POMs.

ATE is

$$\theta_0 = E\{g_0(1, \mathbf{x}) - g_0(0, \mathbf{x})\}$$

ATET is

$$\theta_0 = E\{g_0(1, \mathbf{x}) - g_0(0, \mathbf{x})|\tau = 1\}$$

POMs when $\tau = 1$ is

$$\theta_0 = E\{g_0(1, \mathbf{x})\}$$

POMs when $\tau = 0$ is

$$\theta_0 = E\{g_0(0, \mathbf{x})\}$$

We have several remarks about the above model.

1. The outcome model can be one of `linear`, `logit`, `probit`, or `poisson`. Let β_t be the outcome model parameters when treatment $\tau = t$. The table below provides details about the available functional form of $g_0(\tau, \mathbf{x})$.

Outcome model	Functional form for $g_0(\tau = t, \mathbf{x})$
<code>linear</code>	$\mathbf{x}'\beta_t$
<code>logit</code>	$\exp(\mathbf{x}'\beta_t)/\{1 + \exp(\mathbf{x}'\beta_t)\}$
<code>probit</code>	$\Phi(\mathbf{x}'\beta_t)$
<code>poisson</code>	$\exp(\mathbf{x}'\beta_t)$

2. The treatment model can be either `logit` or `probit`. Let γ be the parameters in the treatment model. The table below provides details about the available functional form of $m_0(\mathbf{z})$.

Treatment model	Functional form for $m_0(\mathbf{z})$
<code>logit</code>	$\exp(\mathbf{z}'\gamma)/\{1 + \exp(\mathbf{z}'\gamma)\}$
<code>probit</code>	$\Phi(\mathbf{z}'\gamma)$

3. Both $\mathbf{x}$ and $\mathbf{z}$ can be high dimensional, and there may be more variables than the number of observations. However, the outcome model parameter β_t and the treatment model parameter γ are assumed to be sparse. This means that there are only a few nonzero elements in both β_t and γ .
4. Although we assume that the functional forms of $g_0(\cdot)$ and $m_0(\cdot)$ are known, we do not know which variables should enter the model. We use lasso techniques to select variables from the potential high-dimensional controls $\mathbf{x}$ and $\mathbf{z}$. The resulting estimators should guard against possible model selection errors made by lassos.

Next we discuss the methods that `telasso` uses to estimate ATEs, ATETs, and POMs.

Neyman orthogonal moments

Denote θ_0 as the parameter of interest, which can be ATE, ATET, or POM. Following Chernozhukov et al. (2018), we use the following moment conditions to estimate θ_0 .

The moment condition for estimating ATE is

$$\psi(\mathbf{w}; \theta, \boldsymbol{\eta}) = \{g(1, \mathbf{x}) - g(0, \mathbf{x})\} + \frac{\tau \{y - g(1, \mathbf{x})\}}{m(\mathbf{z})} - \frac{(1 - \tau) \{y - g(0, \mathbf{x})\}}{1 - m(\mathbf{z})} - \theta \quad (1)$$

where $\mathbf{w} = (y, \mathbf{x}', \mathbf{z}', \tau)'$ and $\boldsymbol{\eta}$ is the nuisance parameter consisting of $g(\tau, \mathbf{x})$ and $m(\mathbf{z})$. Using the variables selected by lassos, $g(1, \mathbf{x})$, $g(0, \mathbf{x})$, and $m(\mathbf{z})$ are approximations to the true functions $g_0(1, \mathbf{x})$, $g_0(0, \mathbf{x})$, and $m_0(\mathbf{z})$, respectively.

The moment condition for estimating ATET is

$$\psi(\mathbf{w}; \theta; \boldsymbol{\eta}) = \frac{\tau \{y - g(0, \mathbf{x})\}}{p} - \frac{m(\mathbf{z})(1 - \tau) \{y - g(0, \mathbf{x})\}}{p \{1 - m(\mathbf{z})\}} - \frac{\tau \theta}{p} \quad (2)$$

where $p = E(\tau)$.

The moment condition for estimating POM for $\tau = 1$ is

$$\psi(\mathbf{w}; \theta, \boldsymbol{\eta}) = g(1, \mathbf{x}) + \frac{\tau \{y - g(1, \mathbf{x})\}}{m(\mathbf{z})} - \theta \quad (3)$$

The moment condition for estimating POM for $\tau = 0$ is

$$\psi(\mathbf{w}; \theta, \boldsymbol{\eta}) = g(0, \mathbf{x}) + \frac{(1 - \tau) \{y - g(0, \mathbf{x})\}}{1 - m(\mathbf{z})} - \theta \quad (4)$$

To estimate θ_0 , we first use lasso techniques to select variables in the outcome and treatment models. Based on the selected variables, we can estimate the post-lasso predictions for $g(\tau, \mathbf{x})$ and $m(\mathbf{z})$. Denote $\tilde{\boldsymbol{\eta}}$ as the estimates for $g(\tau, \mathbf{x})$ and $m(\mathbf{z})$.

The estimator $\hat{\theta}$ for θ_0 is the solution to

$$\frac{1}{n} \sum_{i=1}^n [\psi_i(\mathbf{w}; \theta, \tilde{\boldsymbol{\eta}})] = 0 \quad (5)$$

The variance estimator for $\hat{\theta}$ is

$$\frac{1}{n^2} \sum_{i=1}^n [\psi_i(\mathbf{w}; \hat{\theta}, \tilde{\boldsymbol{\eta}})^2] = 0$$

To identify θ_0 , we need to assume the conditional mean independence, overlap, and independent treatment assignment. For a detailed discussion of these assumptions, see [CAUSAL] [teffects intro](#) or [CAUSAL] [teffects intro advanced](#).

The moment conditions defined in (1) to (4) are Neyman orthogonal. Intuitively, Neyman orthogonal moments mean that the estimator or parameter of interest is still consistent even if model selection makes some mild mistakes. For a formal introduction on Neyman orthogonality, see section 2.1 in Chernozhukov et al. (2018).

The moment conditions in (1) to (4) also imply the AIPW estimator for treatment effects. The AIPW estimator is doubly robust in the sense that only one of the outcome model or the treatment model is required to be correctly specified.

Double machine learning

`telasso` also implements the double machine learning estimator in Chernozhukov et al. (2018) for ATEs, ATETs, and POMs. One advantage of double machine learning is that it allows `telasso` to select more variables in the models. The sparsity assumption is crucial for the validity of the estimators implemented in `telasso`. In other words, the sparsity assumption allows only a few variables to be selected. Double machine learning techniques relax this sparsity requirement to some extent such that more variables can be selected and the estimation results are still valid.

`telasso` with the `xfold()` option implements cross-fit estimation for ATEs, ATETs, and POMs. Cross-fit estimation fits the nuisance parameters and parameter of interest in different samples. It has the following structure.

Let K be the specified number of cross-fitting folds.

1. Randomly partition the sample into K subsamples called folds.
2. Define I_k to be the observations in fold k , and define IC_k to be the sample observations not in fold k .
3. For each $k = 1, \dots, K$, fill in the observations of $i \in I_k$ for the moment condition $\psi(\cdot)$ in (1), (2), (3), or (4) depending on the specified statistic.
 - a. Using observations $i \in IC_k$, perform model selection, and estimate the parameters in the outcome and treatment models.
 - b. Based on the estimates in step 3a, predict the nuisance parameter $\tilde{\eta}_k$ for observations $i \in I_k$.
 - c. Based on $\tilde{\eta}_k$, fill in the moment conditions $\psi(\cdot)$ for observations $i \in I_k$.
4. The estimator $\hat{\theta}$ is the solution in (5).

Resampling the partitions

The K folds are chosen once by default. Specify the `resample(#)` option to have the K folds randomly selected $\#$ times. This resampling removes the dependence of the estimator on any specifically selected folds, at the cost of more computer time.

Let S be the specified number of resamples.

1. For each random partition $s = 1, \dots, S$, use a cross-fit estimator to obtain point estimate $\hat{\theta}_s$ and the estimated VCE $\widehat{\text{Var}}(\hat{\theta}_s)$.
2. The mean resampling-corrected point estimates are

$$\tilde{\theta} = \frac{1}{S} \sum_{s=1}^S \hat{\theta}_s$$

3. The mean resampling-corrected estimate of the VCE is

$$\widetilde{\text{Var}}(\tilde{\theta}) = \frac{1}{S} \sum_{s=1}^S \left\{ \widehat{\text{Var}}(\hat{\theta}_s) + (\hat{\theta}_s - \tilde{\theta})(\hat{\theta}_s - \tilde{\theta})' \right\}$$

References

- Ahrens, A., C. B. Hansen, M. E. Schaffer, and T. Wiemann. 2024. [ddml: Double/debiased machine learning in Stata](#). *Stata Journal* 24: 3–45.
- Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. B. Hansen, W. K. Newey, and J. M. Robins. 2018. Double/debiased machine learning for treatment and structural parameters. *Econometrics Journal* 21: C1–C68. <https://doi.org/10.1111/ectj.12097>.
- Chernozhukov, V., and C. B. Hansen. 2004. The effects of 401(k) participation on the wealth distribution: An instrumental quantile regression analysis. *Review of Economics and Statistics* 86: 735–751. <https://doi.org/10.1162/0034653041811734>.
- Drukker, D. M., and D. Liu. 2025. [posis: Command for the sure-independence-screening Neyman orthogonal estimator](#). *Stata Journal* 25: 561–586.
- Farrell, M. H. 2015. Robust inference on average treatment effects with possibly more covariates than observations. *Journal of Econometrics* 189: 1–23. <https://doi.org/10.1016/j.jeconom.2015.06.017>.
- Koch, B., D. M. Vock, and J. Wolfson. 2018. Covariate selection with group lasso and doubly robust estimation of causal effects. *Biometrics* 74: 8–17. <https://doi.org/10.1111/biom.12736>.
- Leeb, H., and B. M. Pötscher. 2005. Model selection and inference: Facts and fiction. *Econometric Theory* 21: 21–59. <https://doi.org/10.1017/S0266466605050036>.
- . 2006. Can one estimate the conditional distribution of post-model-selection estimators? *Annals of Statistics* 34: 2554–2591. <https://doi.org/10.1214/009053606000000821>.
- Poterba, J. M., S. F. Venti, and D. A. Wise. 1994. “401(k) plans and tax-deferred saving”. In *Studies in the Economics of Aging*, edited by D. A. Wise, 105–142. New York: National Bureau of Economic Research.
- . 1995. Do 401(k) contributions crowd out other personal saving? *Journal of Public Economics* 58: 1–32. [https://doi.org/10.1016/0047-2727\(94\)01462-W](https://doi.org/10.1016/0047-2727(94)01462-W).
- Rosenbaum, P. R., and D. B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70: 41–55. <https://doi.org/10.2307/2335942>.

Also see

[CAUSAL] [telasso postestimation](#) — Postestimation tools for telasso

[CAUSAL] [teffects aipw](#) — Augmented inverse-probability weighting

[LASSO] [Lasso inference intro](#) — Introduction to inferential lasso models

[U] [20 Estimation and postestimation commands](#)

Postestimation commands

The following postestimation commands are of special interest after `telasso`:

Command	Description
<code>teoverlap</code>	overlap plots
<code>tebalance</code>	check balance of covariates
<code>bicplot</code>	plot Bayesian information criterion function
* <code>coefpath</code>	plot path of coefficients
* <code>cvplot</code>	plot cross-validation function
<code>lassocoeff</code>	display selected coefficients
<code>lassoinfo</code>	information about lasso estimation results
<code>lassoknots</code>	knot table of coefficient selection and measures of fit
* <code>lassoselect</code>	select alternative λ^*

*`coefpath`, `cvplot`, and `lassoselect` require the selection method of the lasso to be `selection(cv)`, `selection(adaptive)`, or `selection(bic)`. See [\[LASSO\] lasso options](#).

The following standard postestimation commands are also available:

Command	Description
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of parameters
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
<code>predict</code>	treatment effects, conditional means at treatment, propensity scores, etc.
<code>predictnl</code>	point estimates for generalized predictions
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

predict

Description for predict

predict creates a new variable containing predictions such as treatment effects, potential outcomes, conditional means, propensity scores, and linear predictions.

Menu for predict

Statistics > Postestimation

Syntax for predict

```
predict [type] { stub* | newvar | newvars } [if] [in]
      [, statistic tlevel(treat_level)]
```

statistic	Description
Main	
te	treatment effect; the default
cmean	conditional mean at treatment level
ps	propensity score
xb	linear prediction
psxb	linear prediction for propensity score

Option tlevel() may not be combined with te or psxb.

If you do not specify tlevel() and only specify one new variable, then cmean, ps, and xb assume tlevel() specifies the control.

You specify one or two new variables with cmean, ps, and xb.

You specify one new variable with te and psxb.

Options for predict

Main

te, the default, calculates the treatment effect for the noncontrol treatment level. You need to specify only one new variable.

cmean calculates the conditional mean for each treatment level or the treatment level specified in tlevel(). If you specify the tlevel() option, you need to specify only one new variable; otherwise, you must specify two new variables corresponding to the control and noncontrol treatment levels.

ps calculates the propensity score of each treatment level or the treatment level specified in tlevel(). If you specify the tlevel() option, you need to specify only one new variable; otherwise, you must specify two new variables corresponding to the control and noncontrol treatment levels.

xb calculates the linear prediction at each treatment level or the treatment level specified in tlevel(). If you specify the tlevel() option, you need to specify only one new variable; otherwise, you must specify two new variables corresponding to the control and noncontrol treatment levels.

`psxb` calculates the linear prediction for the propensity score at the noncontrol level of the treatment.

You need to specify only one new variable.

`tlevel(treat_level)` specifies the treatment level for prediction.

Remarks and examples

Some of the `telasso` postestimation commands explore the lasso results computed within `telasso`. Here is a list of such commands: `coefpath`, `cvplot`, `bicplot`, `lassoknots`, `lassoselect`, and `lassocoef`.

When referring to a lasso result computed by `telasso`, there is a distinction between the outcome model and the treatment model. To refer to the lasso result for the treatment model, we need to specify the treatment variable with the `for(tvar)` option. In contrast, to refer to the lasso result for the outcome model, we need to specify the outcome variable at a specific treatment level with the `for(ovar)` and `tlevel(#)` options. In summary, for the treatment model, the `for(tvar)` option is required; for the outcome model, both the `for(ovar)` and the `tlevel(#)` options are required.

Examples that demonstrate how to use the `telasso` command and explore the lasso results using the postestimation tools can be found in [Remarks and examples](#) in [\[CAUSAL\] telasso](#).

Also see

[\[CAUSAL\] telasso](#) — Treatment-effects estimation using lasso

[\[U\] 20 Estimation and postestimation commands](#)

[Description](#)
[Options](#)
[Also see](#)[Quick start](#)
[Remarks and examples](#)[Menu](#)
[Stored results](#)[Syntax](#)
[References](#)

Description

One of the assumptions required to use the `teffects` and `stteffects` estimators is the overlap assumption, which states that each individual has a positive probability of receiving each treatment level. `teoverlap`, a postestimation command, plots the estimated densities of the probability of getting each treatment level. These plots can be used to check whether the overlap assumption is violated.

Quick start

Visually check whether the overlap assumption is violated

```
teoverlap
```

Same as above, but use the Epanechnikov kernel function

```
teoverlap, kernel(epanechnikov)
```

Plot probability of getting treatment 3 for subjects receiving treatments 2 or 3 of a multivalued treatment

```
teoverlap, ptlevel(3) tlevels(2 3)
```

Same as above, and change legend labels to “Treated 2” and “Treated 3”

```
teoverlap, ptlevel(3) tlevels(2 3) ///  
    legend(label(1 "Treated 2") label(2 "Treated 3"))
```

Menu

Statistics > Postestimation

Syntax

```
teoverlap [ , treat_options kden_options ]
```

<i>treat_options</i>	Description
Main	
<code>ptlevel(<i>treat_level</i>)</code>	calculate predicted probabilities for treatment level <i>treat_level</i> ; by default, <code>ptlevel()</code> corresponds to the first treatment level
<code>tlevels(<i>treatments</i>)</code>	specify conditioning treatment levels; default is all treatment levels
<code>nolabel</code>	use treatment level values and not value labels in legend and axis titles

<i>kden_options</i>	Description
Main	
<code>kernel(<i>kernel</i>)</code>	specify kernel function; default is <code>kernel(triangle)</code>
<code>n(#)</code>	estimate densities using # points; default is <code>e(N)</code> , the number of observations in the estimation sample
<code>bwidth(#)</code>	half-width of kernel
<code>at(<i>var_x</i>)</code>	estimate densities using the values specified by <i>var_x</i>
Kernel plots	
<code>line#opts(<i>cline_options</i>)</code>	affect rendition of density for conditioning treatment #
Add plots	
<code>addplot(<i>plot</i>)</code>	add other plots to the generated graph
Y axis, X axis, Titles, Legend, Overall	
<code>twoway_options</code>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>

<i>kernel</i>	Description
<code>triangle</code>	triangle kernel function; the default
<code>epanechnikov</code>	Epanechnikov kernel function
<code>epan2</code>	alternative Epanechnikov kernel function
<code>biweight</code>	biweight kernel function
<code>cosine</code>	cosine trace kernel function
<code>gaussian</code>	Gaussian kernel function
<code>parzen</code>	Parzen kernel function
<code>rectangle</code>	rectangle kernel function

collect is allowed; see [U] 11.1.10 Prefix commands.

Options

Main

`ptlevel(treat_level)` specifies that predicted probabilities be calculated for treatment level *treat_level*. The default is `ptlevel(first)`, where *first* is the first treatment level.

`tlevels(treatments)` specifies the observations for which to obtain predicted probabilities. By default, all treatment levels are used. Specify *treatments* as a space-delimited list.

For instance,

```
. teoverlap, ptlevel(1) tlevels(1 2)
```

says to predict the probability of getting treatment level 1 for those subjects who actually obtained treatment levels 1 or 2.

`no label` specifies that treatment level values and not value labels be used in legend and axis titles.

`kernel(kernel)` specifies the kernel function for use in calculating the kernel density estimates. The default kernel is the triangle kernel (`triangle`).

`n(#)` specifies the number of points at which the density estimate is to be evaluated. The default is $e(N)$, the estimation sample size.

`bwidth(#)` specifies the half-width of the kernel, the width of the density window around each point. If `bwidth()` is not specified, the “optimal” width is calculated and used; see [R] [kdensity](#). The optimal width is the width that would minimize the mean integrated squared error if the data were Gaussian and a Gaussian kernel were used, so it is not optimal in any global sense. In fact, for multimodal and highly skewed densities, this width is usually too wide and oversmooths the density ([Silverman 1986](#)).

`at(var_x)` specifies a variable that contains the values at which the density should be estimated. This option allows you to more easily obtain density estimates for different variables or different subsamples of a variable and then overlay the estimated densities for comparison.

Kernel plots

`line#opts(cline_options)` affect the rendition of the plotted kernel density estimates. See [G-3] [cline_options](#).

Add plots

`addplot(plot)` provides a way to add other plots to the generated graph. See [G-3] [addplot_option](#).

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] [twoway_options](#), excluding `by()`. These include options for titling the graph (see [G-3] [title_options](#)) and for saving the graph to disk (see [G-3] [saving_option](#)).

Remarks and examples

`teoverlap` plots the estimated densities of the probability of getting each treatment level after `teffects`.

These plots can be used to check whether the overlap assumption is violated. The overlap assumption is satisfied when there is a chance of seeing observations in both the control and the treatment groups at each combination of covariate values; see [CAUSAL] [teffects intro](#) or [CAUSAL] [teffects intro advanced](#).

The overlap assumption is required by the estimators implemented in `teffects`. Intuitively, when the overlap assumption is violated, we cannot predict, or otherwise account for, the unobserved outcomes for some individuals.

There is evidence that the overlap assumption is violated when an estimated density has too much mass around 0 or 1; see [Busso, DiNardo, and McCrary \(2014\)](#). An implication of this point is that when the overlap assumption is violated, the estimated densities will have relatively little mass in the regions in which they overlap.

► Example 1: Assumption not violated

Continuing with [example 1](#) of [\[CAUSAL\] teffects ipw](#), we estimate the average treatment effect of smoking on birthweight and then draw the overlap plot:

```
. use https://www.stata-press.com/data/r19/cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)

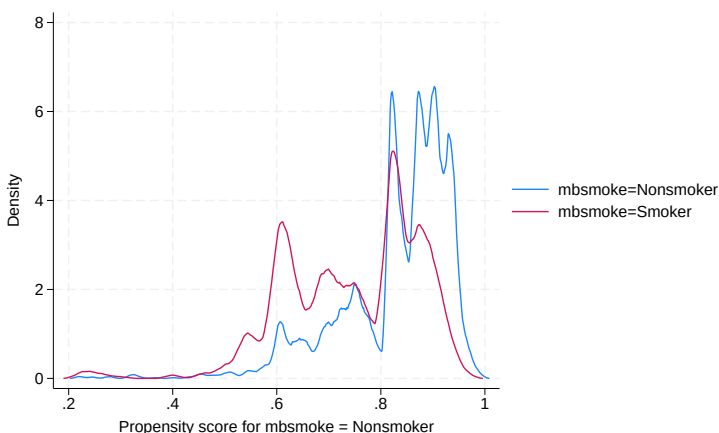
. teffects ipw (bweight) (mbsmoke mmarried c.mage##c.mage fbaby medu, probit)

Iteration 0: EE criterion = 4.622e-21
Iteration 1: EE criterion = 8.795e-26

Treatment-effects estimation      Number of obs      =      4,642
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: probit
```

bweight	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
mbsmoke (Smoker vs Nonsmoker)	-230.6886	25.81524	-8.94	0.000	-281.2856	-180.0917
P0mean						
mbsmoke Nonsmoker	3403.463	9.571369	355.59	0.000	3384.703	3422.222

```
. teoverlap
```



The graph displays the estimated density of the predicted probabilities that a nonsmoking mother is a nonsmoker and the estimated density of the predicted probabilities that a smoking mother is a nonsmoker.

Neither plot indicates too much probability mass near 0 or 1, and the two estimated densities have most of their respective masses in regions in which they overlap each other. Thus there is no evidence that the overlap assumption is violated.

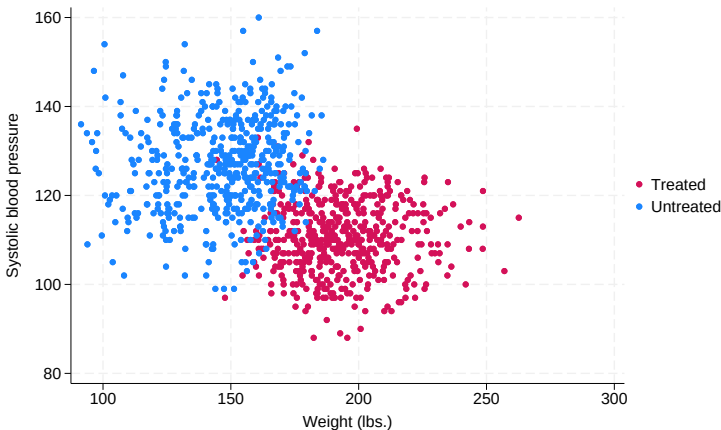


► Example 2: Assumption violated

This example produces an overlap plot that indicates a failure of the overlap assumption. We will use simulated data, so we know that the assumption is not true.

In our simulated dataset, some of the 1,000 adult males were given drug XY1 for high blood pressure and others were not. A scatterplot of systolic blood pressure (`systolic`) and weight (`weight`) reveals that heavier men were given the treatment. (The scatterplots corresponding to the treatment group are colored red, while the scatterplots corresponding to the control group are colored blue.)

```
. use https://www.stata-press.com/data/r19/systolic2
(Systolic blood pressure)
. twoway (scatter systolic weight if xy1==1, mcolor(stred))
> (scatter systolic weight if xy1==0, mcolor(stblue)),
> legend(label(1 "Treated") label(2 "Untreated"))
```



There are no observations in the treated group for small weights, and there are no observations in the control group for large weights. There is clear evidence that the overlap assumption is violated.

Drawing an overlaid scatterplot is a straightforward way to check the overlap assumption in this example because there is only one covariate. This method is not available when there is more than one covariate. The predicted probability is a one-dimensional measure that captures the relevant multivariate information.

Below we estimate the parameters needed to calculate the predicted probabilities. The `pstolerance(1e-8)` option is specified to ensure that estimation is performed as long as the predicted probabilities are at least as large as $1e-8$.

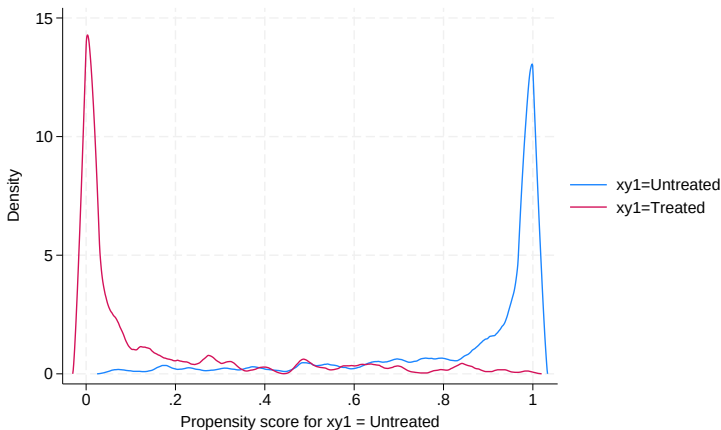
```
. teffects ipw (systolic) (xy1 weight), pstolerance(1e-8)
Iteration 0: EE criterion = 9.523e-18
Iteration 1: EE criterion = 3.489e-28

Treatment-effects estimation      Number of obs      =      1,000
Estimator      : inverse-probability weights
Outcome model  : weighted mean
Treatment model: logit
```

systolic	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE						
xy1 (Treated vs Untreated)	-16.23679	2.191703	-7.41	0.000	-20.53245	-11.94114
POMean						
xy1 Untreated	127.9094	.7004533	182.61	0.000	126.5365	129.2822

Now we can obtain the overlap plot.

```
. teoverlap
```



The estimated density of the predicted probabilities that a treated individual is not assigned to XY1 treatment has most of its mass near 0. The estimated density of the predicted probabilities that an untreated individual is not assigned to XY1 treatment has most of its mass near 1. Note that the two have very little mass in the region in which they overlap. There is clear evidence that the overlap assumption is violated.

Stored results

`teoverlap` stores the following in `r()`:

Scalars

<code>r(bwidthj)</code>	kernel bandwidth for treatment level j
<code>r(nj)</code>	number of points at which the estimate was evaluated for treatment level j
<code>r(scalej)</code>	density bin width for treatment level j

Macros

<code>r(kernel)</code>	name of kernel
------------------------	----------------

References

- Busso, M., J. DiNardo, and J. McCrary. 2014. New evidence on the finite sample properties of propensity score reweighting and matching estimators. *Review of Economics and Statistics* 96: 885–897. https://doi.org/10.1162/REST_a_00431.
- Silverman, B. W. 1986. *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.

Also see

- [CAUSAL] [stteffects](#) — Treatment-effects estimation for observational survival-time data
- [CAUSAL] [stteffects ipw](#) — Survival-time inverse-probability weighting
- [CAUSAL] [stteffects ipwra](#) — Survival-time inverse-probability-weighted regression adjustment
- [CAUSAL] [teffects](#) — Treatment-effects estimation for observational data
- [CAUSAL] [teffects aipw](#) — Augmented inverse-probability weighting
- [CAUSAL] [teffects ipw](#) — Inverse-probability weighting
- [CAUSAL] [teffects ipwra](#) — Inverse-probability-weighted regression adjustment
- [CAUSAL] [teffects nnmatch](#) — Nearest-neighbor matching
- [CAUSAL] [teffects psmatch](#) — Propensity-score matching
- [CAUSAL] [teffects ra](#) — Regression adjustment
- [CAUSAL] [telasso](#) — Treatment-effects estimation using lasso

⁺This command includes features that are part of [StataNow](#).[Description](#)[Options](#)[Acknowledgments](#)[Quick start](#)[Remarks and examples](#)[References](#)[Menu](#)[Stored results](#)[Also see](#)[Syntax](#)[Methods and formulas](#)

Description

`xthdidregress` estimates average treatment effects on the treated (ATETs) that may vary over time and over treatment cohorts. Treatment cohorts are groups subject to treatment at different points in time. `xthdidregress` provides four estimators: extended two-way fixed effects (TWFE), regression adjustment (RA), inverse-probability weighting (IPW), and augmented inverse-probability weighting (AIPW). See [\[CAUSAL\] teffects intro](#) for a discussion of RA, IPW, and AIPW estimators.

`xthdidregress` is for panel data. For repeated cross-sectional data, see [\[CAUSAL\] hdidregress](#).

Quick start

Estimate ATETs of treatment `treat` on outcome `y` with group `grpvar`; use the RA estimator and model `y` with covariate `x` on `xtset` data

```
xthdidregress ra (y x) (treat), group(grpvar)
```

Same as above, but use the TWFE estimator

```
xthdidregress twfe (y x) (treat), group(grpvar)
```

Use the IPW estimator and model `treat` using covariate `z`

```
xthdidregress ipw (y) (treat z), group(grpvar)
```

Use the AIPW estimator, model `y` using covariate `x`, and model `treat` using covariate `z`

```
xthdidregress aipw (y x) (treat z), group(grpvar)
```

Same as above, but use the not-yet-treated group as the control group

```
xthdidregress aipw (y x) (treat z), group(grpvar) ///
    controlgroup(notyet)
```

Same as above, but cluster at the county level

```
xthdidregress aipw (y x) (treat z), group(grpvar) ///
    controlgroup(notyet) vce(cluster county)
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Heterogeneous DID > Panel-data heterogeneous DID (TWFE)

Statistics > Causal inference/treatment effects > Continuous outcomes > Heterogeneous DID > Panel-data heterogeneous DID (RA)

Statistics > Causal inference/treatment effects > Continuous outcomes > Heterogeneous DID > Panel-data heterogeneous DID (IPW)

Statistics > Causal inference/treatment effects > Continuous outcomes > Heterogeneous DID > Panel-data heterogeneous DID (AIPW)

Syntax

Two-way fixed effects

```
xthdidregress twfe (ovar [omvarlist]) (tvar) [if] [in] [weight],
    group(groupvar) [options]
```

Regression adjustment

```
xthdidregress ra (ovar [omvarlist]) (tvar) [if] [in] [weight],
    group(groupvar) [options]
```

Inverse-probability weighting

```
xthdidregress ipw (ovar) (tvar [tmvarlist]) [if] [in] [weight],
    group(groupvar) [options]
```

Augmented inverse-probability weighting

```
xthdidregress aipw (ovar [omvarlist]) (tvar [tmvarlist]) [if] [in] [weight],
    group(groupvar) [options]
```

ovar is a continuous outcome of interest.

omvarlist specifies the covariates in the outcome model and may contain factor variables; see [\[U\] 11.4.3 Factor variables](#).

tvar must be a binary variable indicating observations subject to treatment.

tmvarlist specifies the covariates in the treatment model and may contain factor variables; see [\[U\] 11.4.3 Factor variables](#).

groupvar is a categorical variable that indicates the group level at which the treatment occurs.

<i>options</i>	Description
Model	
* <u>group</u> (<i>groupvar</i>)	specify group variable
<u>controlgroup</u> (<i>cgtype</i>)	specify the type of control group; default is <code>controlgroup(never)</code>
<u>cohortvar</u> (<i>cvar</i> [, <code>replace</code>])	specify the variable name for the generated cohort
<u>usercohort</u> (<i>varname</i>)	specify name of cohort variable to be used for estimation
† <u>basetime</u> (<i>btspec</i>)	specify the type of base time for pretreatment periods; default is <code>basetime(adaptive)</code>
‡ <u>hettype</u> (<i>hetspec</i>)	specify the type of heterogeneity; default is <code>hettype(timecohort)</code>
+‡ <u>over</u> (<i>overspec</i>)	specify the subpopulation over which each ATET is computed; default is <code>over(cohort)</code>
+‡ <u>noxinteract</u>	omit covariate interactions
SE/Robust	
<u>vce</u> (<i>vcetype</i>)	<i>vcetype</i> may be <code>cluster</code> <i>clustvar</i> , <code>robust</code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<u>level</u> (#)	set confidence level; default is <code>level(95)</code>
[<code>no</code>] <u>log</u>	suppress iteration log
<u>nodots</u>	suppress replication dots
<i>display_options</i>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
<u>coeflegend</u>	display legend instead of statistics
<i>cgtype</i>	Description
<code>never</code>	use the never-treated group as the control group; the default
<code>notyet</code>	use the not-yet-treated group as the control group
<i>btspec</i>	Description
<code>adaptive</code>	specify the adaptive base time for pretreatment ATETs; the default
<code>common</code>	specify a common base time for all pretreatment ATETs
<i>hetspec</i>	Description
<code>timecohort</code>	heterogeneous treatment effects over time and cohort; the default
<code>time</code>	heterogeneous treatment effects over time
<code>cohort</code>	heterogeneous treatment effects over cohort

<i>overspec</i>	Description
cohort	use cohort subpopulations when controlling for covariate heterogeneity; the default
timecohort	use time and cohort subpopulations when controlling for covariate heterogeneity

⁺These features are part of [StataNow](#).

*group() is required.

[†]basetime() may be specified only when ra, ipw, or aipw is specified.

[‡]hettype() and noxinteract may be specified only when twfe is also specified. over() may be specified only when twfe and hettype(timecohort) are also specified.

A panel variable and a time variable must be specified using xtset; see [\[XT\] xtset](#).

by, collect, and statsby are allowed; see [\[U\] 11.1.10 Prefix commands](#).

fweights, aweights, and pweights are allowed; see [\[U\] 11.1.6 weight](#). Weights must be constant within panel.

coeflegend does not appear in the dialog box.

See [\[U\] 20 Estimation and postestimation commands](#) for more capabilities of estimation commands.

Options

Model

group(*groupvar*) specifies a group variable that indicates the group level at which the treatment occurs. *groupvar* may be, for example, states, counties, or hospitals. group() also defines the clusters for the default cluster-robust standard errors. group() is required. You may specify only one group variable.

controlgroup(*cgtype*) specifies the type of control group. A control group can be either a never-treated group or a not-yet-treated group. A never-treated group refers to the units that are untreated from the first to the last period. A not-yet-treated group refers to the units that are untreated up to a specific period. *cgtype* can be one of never, referring to the never-treated group, or notyet, referring to the not-yet-treated group. By default, *cgtype* is never.

cohortvar(*cvar* [, replace]) specifies the variable name *cvar* for the generated cohort variable. The cohort variable is a categorical variable indicating the period when the unit is first treated. By default, `_did_cohort` is used as the name of the cohort variable. If `_did_cohort` already exists in the dataset, it is replaced if option cohortvar() is not specified.

If suboption replace is specified, *cvar* is replaced.

usercohort(*varname*) specifies a variable to be used as a cohort indicator during estimation. By default, a cohort variable is generated using the information in the estimation sample to indicate the period when a unit is first treated. usercohort() overrides this default and allows you to provide a cohort indicator. This is useful, for instance, when there are gaps in the estimation sample, but you know a group was treated at the time when the gap is present in the data.

basetime(*btspec*) specifies how the base time is chosen when computing the pretreatment ATETs with the ra, ipw, or aipw estimator. *btspec* is one of adaptive (the default) or common.

adaptive specifies that the base time for pretreatment ATETs be chosen adaptively. The base time for each pretreatment period t for cohort g is the previous period, $t - 1$.

common specifies that a common base time of $g - 1$ be used for all pretreatment ATETs for cohort g . A long-run violation of the parallel trends assumption is easier to identify when using this common base time.

The base time for posttreatment periods is $g - 1$, regardless of whether the adaptive or common base time is used for pretreatment periods.

`hettype(hetspec)` specifies time or cohort heterogeneity for the `twfe` estimator. By default, treatment is interacted with time and cohort. You may choose to keep one of time or cohort interactions using `hetspec`.

`hetspec` may be one of `timecohort` for heterogeneous treatment effects over both time and cohort, `time` for heterogeneous treatment effects over time only, or `cohort` for heterogeneous treatment effects over cohort only. By default, `hetspec` is `timecohort`.

`over(overspec)` is part of StataNow. It specifies the subpopulation over which treatment effects are averaged for the `twfe` estimator when the heterogeneity type is specified as `timecohort` (`hettype(timecohort)`). When covariates are specified, we need to control for their variation when computing treatment effects. `overspec` may be one of `cohort`, for effects averaged over each cohort, or `timecohort`, for effects averaged over each time period and cohort. The default is `over(cohort)`, meaning effects are averaged over each cohort and adjusted for the time period effect. If there are no covariates, this option has no effect.

`over()` is not allowed when `hettype()` is specified as `time` or `cohort`.

`noxinteract` is part of StataNow. It specifies that the covariates in `omvarlist` not be interacted with the difference-in-difference regressors in the TWFE model. Covariates are still included in the model as levels.

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that allow for intra-group correlation (`cluster clustvar`), that are robust to intragroup correlation among group variable (`robust`), and that use bootstrap or jackknife sampling done at the panel level (`bootstrap`, `jackknife`); see [R] [vce_option](#).

`vce(cluster clustvar)`, the default, uses the variable specified in `group(groupvar)`.

Specifying `vce(robust)` is equivalent to specifying `vce(cluster clustvar)`, where `clustvar` is the variable specified in the `group(groupvar)` option.

Reporting

`level(#)`; see [R] [Estimation options](#).

`log` and `nolog` specify whether to display the iteration log. The iteration log is displayed by default unless you used `set iterlog off` to suppress it; see `set iterlog` in [R] [set iter](#).

`nodots` suppresses display of the replication dots.

`display_options:` `nocl`, `nopvalues`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

The following option is available with `xthdidregress` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

It is common to study the effects of a treatment, for example, a policy or intervention, on a group. `xthdidregress` is for data where the treated groups are subject to the treatment at different points in time and they remain exposed to the treatment. For example, a health policy such as an increase in the age to purchase cigarettes is implemented in a given region, and over time, other regions decide to imitate the initiative. Another example is change in work policies across industries. Perhaps airlines implement a minimum number of hours between shifts for safety reasons. The policy is subsequently adopted by other similar industries. Some similar industries may never adopt the policy, remaining untreated, or it might be that all similar industries eventually adopt the policy.

`xthdidregress` estimates ATET parameters that change over time and treatment cohorts (groups treated at different points in time). Each one of these ATETs has the same interpretation that the parameters of a two-time two-group difference-in-differences (DID) parameter would have. Because there are multiple DID parameters, we refer to them as heterogeneous treatment effects or as heterogeneous DID. This is in contrast to estimating only one ATET, which assumes there is no variation across time or cohort. If you assume no variation across time or cohort, you may use `xtddidregress`; see [\[CAUSAL\] didregress](#).

`xthdidregress` provides four estimators: TWFE, outlined in [Wooldridge \(2021\)](#); RA, IPW, and AIPW, outlined in [Callaway and Sant’Anna \(2021\)](#). Each one of these estimators fits a model for the outcome of interest, a model for the treatment, or a model for both. For example, RA and TWFE model the outcome; IPW models the treatment; and AIPW models both. If the model for the outcome is correctly specified, RA and TWFE are best, with TWFE being more efficient. If the treatment model is correctly specified, IPW should be best. AIPW models both treatment and outcome. If at least one of the models is correctly specified, it provides consistent estimates. Thus, it allows us to misspecify one of the models and still get consistent estimates, a property called double robustness. See [\[CAUSAL\] teffects intro](#) for a discussion of RA, IPW, and AIPW estimators.

`xthdidregress` is for panel data. For repeated cross-sectional data, see [\[CAUSAL\] hdidregress](#). Below, we illustrate how to use `xthdidregress`. For more information about the methods used below, see [\[CAUSAL\] DID intro](#). For general discussions about the methods, see [Roth et al. \(2022\)](#) and [de Chaisemartin and D’Haultfœuille \(2023\)](#) and the references therein.

▷ Example 1: Fitting a heterogeneous DID model

We are interested in how the number of registrations of a dog breed with the American Kennel Club (AKC), `registered`, is affected by dogs being the protagonists in a movie, `movie`. We conjecture that the number of registrations increases if the dog breed appears as the protagonist in a movie. We also conjecture that registrations increase if the dog has won the Best in Show award from the Westminster Kennel Club, `best`, in the 10 years before 2034. We use simulated data, but there is some evidence of the effect of movies on dog breed registrations. See, for example, [Ghirlanda, Acerbi, and Herzog \(2014\)](#).

There are 141 dog breeds in our sample, which extends between the years 2031 and 2040. At the beginning of the sample, none of the breeds are featured in a movie. This changes in 2034, when four breeds are featured in movies. The next year in which we see an increase of breeds featured in movies is 2036, when 7 more breeds are featured. In 2037, there is a substantial increase, with 22 more breeds featured. There is no increase in breeds in movies thereafter. The table below illustrates this.

```
. use https://www.stata-press.com/data/r19/akc
(Fictional dog breed and AKC registration data)
. tabulate year movie
```

Year	Was a movie protagonist		Total
	0	1	
2031	141	0	141
2032	141	0	141
2033	141	0	141
2034	137	4	141
2035	137	4	141
2036	134	7	141
2037	119	22	141
2038	119	22	141
2039	119	22	141
2040	119	22	141
Total	1,307	103	1,410

To obtain the effect of movie on registered controlling for best, we first xtset our panel dataset:

```
. xtset breed year
Panel variable: breed (strongly balanced)
Time variable: year, 2031 to 2040
Delta: 1 unit
```

We chose regression adjustment, `ra`, to fit our model (see [\[CAUSAL\] teffects ra](#) for details on this model). In the first set of parentheses, we define the outcome, `registered`, and any covariates that affect the outcome directly. In the second set of parentheses, we define the observation-level treatment variable, `movie`. Other models allow us to define covariates that affect the treatment, but this is not the case of the `ra` model. After the comma, we need to define the group variable in `group()`; this is a required option. The group variable defines at which level the treatment occurs and also identifies the clustering variable, which in this case is `breed`.

```
. xthdidregress ra (registered best) (movie), group(breed)
note: variable _did_cohort, containing cohort indicators formed by treatment
      variable movie and group variable breed, was added to the dataset using
      the estimation sample.
```

Computing ATET for each cohort and time:

Cohort 2034 (9): done

Cohort 2036 (9): done

Cohort 2037 (9): done

Treatment and time information

Time variable: year

Time interval: 2031 to 2040

Control: `_did_cohort = 0`

Treatment: `_did_cohort > 0`

	<code>_did_cohort</code>
Number of cohorts	4
Number of obs	
Never treated	1190
2034	40
2036	30
2037	150

```

Heterogeneous-treatment-effects regression          Number of obs   = 1,410
                                                    Number of panels =   141

Estimator:      Regression adjustment
Panel variable:  breed
Treatment level: breed
Control group:   Never treated

```

(Std. err. adjusted for 141 clusters in breed)

Cohort	ATET	Robust std. err.	z	P> z	[95% conf. interval]	
2034						
year						
2032	-254.8927	266.1024	-0.96	0.338	-776.4439	266.6584
2033	-257.5329	217.9389	-1.18	0.237	-684.6852	169.6194
2034	701.1318	127.0935	5.52	0.000	452.0331	950.2304
2035	1099.044	282.0704	3.90	0.000	546.196	1651.892
2036	1367.632	225.8702	6.05	0.000	924.9343	1810.329
2037	2008.294	237.2396	8.47	0.000	1543.313	2473.275
2038	2472.624	278.2949	8.88	0.000	1927.176	3018.072
2039	2689.615	504.3324	5.33	0.000	1701.142	3678.088
2040	3110.97	568.916	5.47	0.000	1995.915	4226.025
2036						
year						
2032	216.0259	122.9107	1.76	0.079	-24.87472	456.9265
2033	-172.5154	372.0776	-0.46	0.643	-901.7741	556.7433
2034	-218.0495	504.5267	-0.43	0.666	-1206.904	770.8045
2035	621.033	156.1306	3.98	0.000	315.0227	927.0434
2036	999.0781	180.1055	5.55	0.000	646.0779	1352.078
2037	1003.333	250.5916	4.00	0.000	512.1829	1494.484
2038	1556.669	451.6914	3.45	0.001	671.3697	2441.967
2039	2590.674	662.6979	3.91	0.000	1291.81	3889.538
2040	2225.712	486.9917	4.57	0.000	1271.225	3180.198
2037						
year						
2032	-114.582	160.0972	-0.72	0.474	-428.3668	199.2028
2033	-127.9856	183.3941	-0.70	0.485	-487.4315	231.4603
2034	33.40901	168.0312	0.20	0.842	-295.9262	362.7442
2035	130.3495	166.2261	0.78	0.433	-195.4477	456.1468
2036	-10.48288	167.5059	-0.06	0.950	-338.7884	317.8226
2037	1717.016	268.5592	6.39	0.000	1190.65	2243.383
2038	2086.798	278.0215	7.51	0.000	1541.886	2631.71
2039	2473.611	268.186	9.22	0.000	1947.976	2999.246
2040	2835.117	378.6699	7.49	0.000	2092.938	3577.296

Note: ATET computed using covariates.

Note: Base time for pretreatment ATETs is the previous period.

Notice the note below the command. A variable with the name `_did_cohort` has been generated. Using the group variable and the observation-level treatment, `xthdidregress` generated treatment-time cohorts. The new variable creates treatment groups based on the time when a group was first treated. For instance, if a Boxer and a Rottweiler are featured in movies in 2034, they are grouped in the 2034 cohort. The variable also contains a category for a control group. In this case, the control group is formed by the breeds that are not featured in a movie. Cohorts are an important input for estimation and for postestimation commands. You do not need to adhere to the default name, `_did_cohort`, and may provide your own name using the `cohortvar()` option.

Next appears a table that gives you a sense of the treatment groups and time. You see the time variable, year, and its range, 2031 to 2040. Then we see what defines a treated or a control group. The table after provides group-level information about the cohort-time groups. The first row tells you the number of cohorts. Following the number of cohorts is a tabulation showing how many observations are in each cohort. For instance, 1,190 observations are never treated in our data. The table gives you a sense of the amount of information available in each cohort and might hint at the variability of cohort-level estimates.

The next table presents the ATET estimates. The first panel shows the ATETs for the 2034 cohort. We first have the 2032 ATET of -255 , with a confidence interval that includes 0. This is as expected; before treatment, the effect should be 0. We should interpret the ATET to mean that among the breeds about which a movie was made in 2034, the expected AKC registrations in 2032 are 255 lower than if the breed had never been in a movie. At treatment onset, in 2034, we observe a treatment effect for the number of registrations of 701, and on the last year of our data, for the 2034 cohort, we see a treatment effect of 3,111 registrations. We interpret the results for the remaining cohorts similarly.



► Example 2: Visualizing estimation results

In the example above, in the output table, we had three cohorts and nine time periods. There is a lot of information to process, and it can get even more daunting if we had more cohorts and time periods. To better visualize the results, we can use `estat atetplot`:

```
. estat atetplot
```

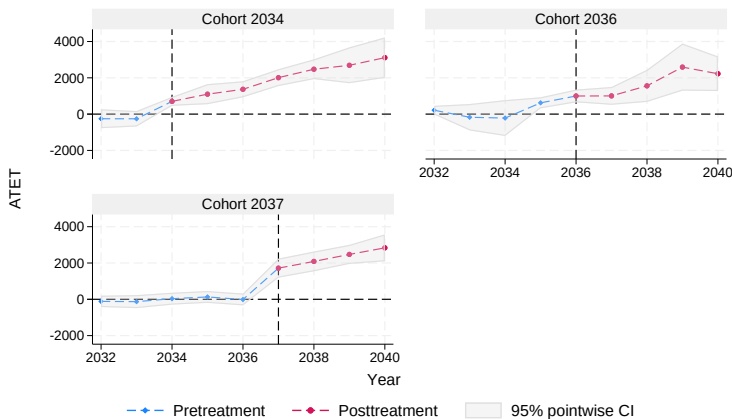


Figure 1. ATETs by cohort over time

The graph shows the pretreatment and the posttreatment ATETs for each cohort and their pointwise confidence intervals. ATETs increase over time after treatment and are flat before treatment.



► Example 3: Less heterogeneity; aggregating and summarizing treatment effects

So far, we have allowed treatment effects to change over cohort and over time. But we might want to obtain only one treatment effect for each cohort, abstracting away from time variation within cohorts. You would get this using the postestimation command `estat aggregation`.

```
. estat aggregation, cohort
ATET over cohort                                     Number of obs = 1,410
                                                    (Std. err. adjusted for 141 clusters in breed)
```

Cohort	ATET	Robust std. err.	z	P> z	[95% conf. interval]	
2034	1921.33	187.2787	10.26	0.000	1554.271	2288.389
2036	1675.093	130.4929	12.84	0.000	1419.332	1930.855
2037	2278.136	166.5283	13.68	0.000	1951.746	2604.525

Note: Aggregation weights vary across times and cohorts.

We now have 3 treatment effects to analyze instead of 27. For the 2034 cohort, we have a treatment effect of 1,921 registrations for the periods after treatment. For the 2036 cohort, the effect is 1,675 registrations, and for the 2037 cohort, it is 2,278. We cannot see how the treatment evolves over time for each cohort, but we have a sense of the average effect over time for each of them.

We could instead want to see the treatment effect at each point in time, abstracting from cohort-level variation.

```
. estat aggregation, time
ATET over time                                     Number of obs = 1,410
                                                    (Std. err. adjusted for 141 clusters in breed)
```

Time	ATET	Robust std. err.	z	P> z	[95% conf. interval]	
2034	701.1318	127.0935	5.52	0.000	452.0331	950.2304
2035	1099.044	282.0704	3.90	0.000	546.196	1651.892
2036	1209.68	170.2043	7.11	0.000	876.0858	1543.275
2037	1672.655	202.1854	8.27	0.000	1276.379	2068.932
2038	2084.658	214.5072	9.72	0.000	1664.232	2505.084
2039	2528.847	225.8763	11.20	0.000	2086.138	2971.557
2040	2802.171	291.8412	9.60	0.000	2230.173	3374.17

Note: Aggregation weights vary across times and cohorts.

We see the treatment effects for each one of the posttreatment periods. As before, we have the option to look at the effects graphically. We just need to use the graph option.

```
. estat aggregation, time graph
```

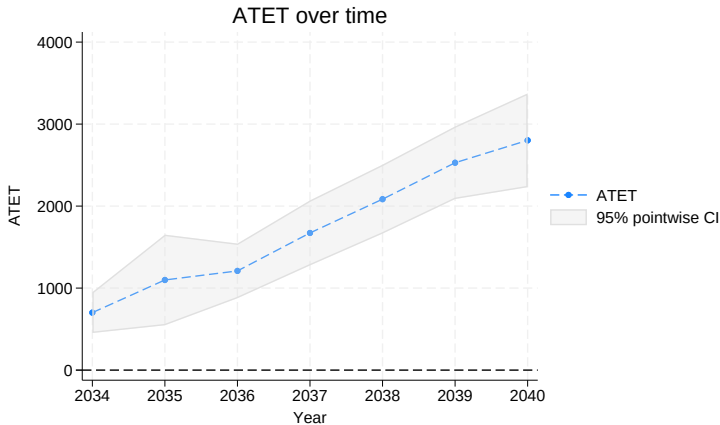


Figure 2. ATETs over time

► Example 4: Dynamic treatment effects

We could also ask what the evolution of the treatment effect is after treatment. For instance, we might want to know what happens one period after the onset of treatment, two periods after treatment, and so forth. It might be the case that treatment effects vanish over time or even change patterns. We might also want to see whether, before treatment, we observe a treatment effect or a pattern that might suggest that there is anticipation of treatment. `estat aggregation` allows us to answer these questions by using the `dynamic` option.

```
. estat aggregation, dynamic graph
```

Duration of exposure ATET

Number of obs = 1,410

(Std. err. adjusted for 141 clusters in breed)

Exposure	ATET	Robust std. err.	z	P> z	[95% conf. interval]	
-5	-114.582	160.0972	-0.72	0.474	-428.3668	199.2028
-4	-70.65034	156.3185	-0.45	0.651	-377.029	235.7283
-3	-.9117242	153.0999	-0.01	0.995	-300.982	299.1585
-2	12.79653	144.8216	0.09	0.930	-271.0486	296.6417
-1	30.71473	132.8508	0.23	0.817	-229.668	291.0975
0	1434.409	206.3277	6.95	0.000	1030.014	1838.804
1	1759.461	224.0229	7.85	0.000	1320.385	2198.538
2	2147.486	221.903	9.68	0.000	1712.564	2582.408
3	2651.452	284.8928	9.31	0.000	2093.073	3209.832
4	2366.805	267.4253	8.85	0.000	1842.661	2890.949
5	2689.615	504.3324	5.33	0.000	1701.142	3678.088
6	3110.97	568.916	5.47	0.000	1995.915	4226.025

Note: Base time for pretreatment ATETs is the previous period.

Note: Exposure is the number of periods since the first treatment time.

Note: Aggregation weights vary across times and cohorts.

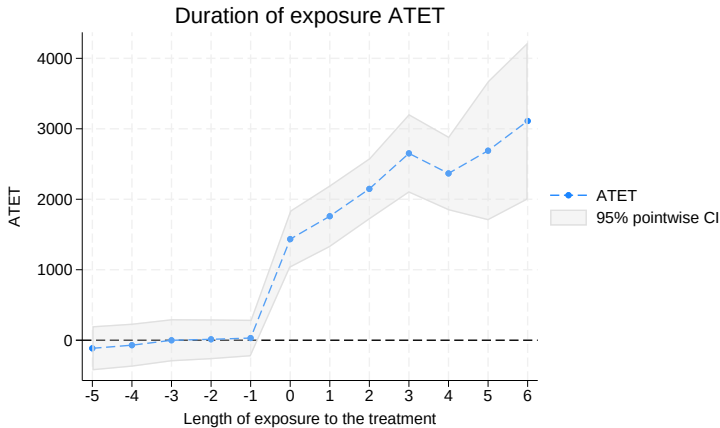


Figure 3. ATET dynamics

From the output table, we see that 5 periods before treatment onset, there is no significant treatment effect or, for that matter, no significant effect at any time of exposure before treatment. This suggests there is no anticipation of treatment. At treatment onset, 0, there is a positive effect, which seems to increase as exposure time to treatment augments.



► Example 5: TWFE estimation

The literature on heterogeneous DID started by pointing out the problems that arise when one assumes erroneously that the treatment effects are homogeneous. It suggested that TWFE estimation was inadequate. [Wooldridge \(2021\)](#) suggests that fixed-effects estimation can be used if we extend it to include interactions between treatment-time cohorts and time.

Another important insight of [Wooldridge \(2021\)](#) is that you can use pooled ordinary least squares and add panel-level averages of covariates and obtain the same point estimates as one would get with fixed-effects estimation in the context of DID estimation. This is an extension of the intuition by [Mundlak \(1978\)](#). `xthdidregress` and `hdidregress` fit pooled ordinary least-squares models using these ideas. Below, we present the results we obtain using the `twfe` estimator.

```
. xthdidregress twfe (registered best) (movie), group(breed)
note: variable _did_cohort, containing cohort indicators formed by treatment
      variable movie and group variable breed, was added to the dataset using
      the estimation sample.
```

Treatment and time information

Time variable: year

Time interval: 2031 to 2040

Control: _did_cohort = 0

Treatment: _did_cohort > 0

	_did_cohort
Number of cohorts	4
Number of obs	
Never treated	1190
2034	40
2036	30
2037	150

Heterogeneous-treatment-effects regression

Number of obs = 1,410

Number of panels = 141

Estimator: Two-way fixed effects

Panel variable: breed

Treatment level: breed

Control group: Never treated

Heterogeneity: Cohort and time

Over: Cohort

(Std. err. adjusted for 141 clusters in breed)

Cohort		ATET	Robust std. err.	t	P> t	[95% conf. interval]	
2034							
	year						
	2034	469.2023	148.8998	3.15	0.002	174.8195	763.5852
	2035	823.8532	211.7491	3.89	0.000	405.2138	1242.493
	2036	1108.669	179.8404	6.16	0.000	753.1144	1464.223
	2037	1752.287	283.487	6.18	0.000	1191.818	2312.756
	2038	2216.617	173.4446	12.78	0.000	1873.708	2559.526
	2039	2433.608	521.4074	4.67	0.000	1402.757	3464.458
	2040	2854.963	494.1892	5.78	0.000	1877.924	3832.001
2036							
	year						
	2036	1336.121	96.75296	13.81	0.000	1144.835	1527.406
	2037	1343.004	297.7487	4.51	0.000	754.3383	1931.669
	2038	1896.339	399.2574	4.75	0.000	1106.985	2685.692
	2039	2930.344	591.0712	4.96	0.000	1761.765	4098.924
	2040	2565.382	578.9303	4.43	0.000	1420.806	3709.958
2037							
	year						
	2037	1750.126	216.4288	8.09	0.000	1322.234	2178.017
	2038	2119.908	217.1685	9.76	0.000	1690.554	2549.262
	2039	2506.72	303.9392	8.25	0.000	1905.816	3107.624
	2040	2868.227	313.1679	9.16	0.000	2249.077	3487.377

Note: ATET computed using covariates.

The output is almost the same as the one for the `ra` estimator in [example 1](#). There are a couple of noteworthy differences. First, the estimator fits an extended TWFE regression. Second, the ATET parameters are shown for each cohort only at the time of treatment exposure and for the periods thereafter but not for the pretreatment periods. As discussed in [Wooldridge \(2021\)](#), these are the parameters identified using the parallel-trends assumption he derives.

As we did before, we could use `estat` aggregation to explore different ways of looking at our treatment effects and `estat atetplot` to visualize the estimated ATETs.



► Example 6: Reducing model complexity

When we fit the `ra` model, we had to estimate ATET parameters for each cohort over time. The complexity of the model grows with the number of cohorts and the number of time periods. As is described in [Methods and formulas](#), the `ra` estimator uses a different subset of the data to obtain each parameter. To get a reliable estimator of each parameter, you need sufficient data for each subsample. Sometimes, there are few observations for a given cohort in a given set of time periods.

We can ameliorate this problem by reducing the amount of heterogeneity we assume. For the `twfe` estimator, the complexity of the model comes from the interactions between the observation-level treatment with cohort and time and the interactions between the observation-level treatment, cohort, time, and covariates. This allows us to decide which interactions to include in our model. We could, for instance, allow for heterogeneity at the cohort level instead of at the cohort and time level. We use the `hettype()` option with the argument `cohort` to do this:

```
. xthdidregress twfe (registered best) (movie), group(breed) hettype(cohort)
note: variable _did_cohort, containing cohort indicators formed by treatment
      variable movie and group variable breed, was added to the dataset using
      the estimation sample.
```

Treatment and time information

Time variable: year

Time interval: 2031 to 2040

Control: `_did_cohort = 0`

Treatment: `_did_cohort > 0`

	<code>_did_cohort</code>
Number of cohorts	4
Number of obs	
Never treated	1190
2034	40
2036	30
2037	150

```
Heterogeneous-treatment-effects regression      Number of obs   = 1,410
                                                Number of panels =  141

Estimator:      Two-way fixed effects
Panel variable:  breed
Treatment level: breed
Control group:   Never treated
Heterogeneity:   Cohort
```

(Std. err. adjusted for 141 clusters in breed)

Cohort	ATET	Robust std. err.	t	P> t	[95% conf. interval]	
2034	1662.492	108.002	15.39	0.000	1448.966	1876.017
2036	1978.645	54.21043	36.50	0.000	1871.468	2085.822
2037	2276.223	70.63244	32.23	0.000	2136.579	2415.867

Note: ATET computed using covariates.

You fit a regression model with fewer terms and obtain treatment effects only at the cohort level. You could also have the treatment effect change over time but not over cohort by typing `hettype(time)`.

For the estimators proposed by [Callaway and Sant’Anna \(2021\)](#), heterogeneity is built in, so we need to estimate all the ATET parameters.



► Example 7: Defining your own cohort

By default, `xthdidregress` creates a cohort variable based on the estimation sample. Yet this might be inadequate if a researcher has more information than is provided in the dataset. Suppose that our dog breed dataset looked something like this for the Boxer breed:

```
. list year breed movie if breed==29, noobs sepby(breed)
```

year	breed	movie
2031	Boxer	0
2032	Boxer	0
2033	Boxer	0
2035	Boxer	1
2036	Boxer	1
2037	Boxer	1
2038	Boxer	1
2039	Boxer	1
2040	Boxer	1

There is no information for the year 2034. If a Boxer appeared in a movie in 2034, the Boxer breed should belong to the 2034 cohort. However, `xthdidregress` has no information about the year 2034 in the estimation sample and will classify the Boxer breed as belonging to the 2035 cohort. `xthdidregress`’s inability to determine the proper cohort is not exclusive to situations with gaps in your panel data. In fact, Stata excludes observations in your sample if any of the variables used during estimation are missing. If all observations for the time period in which a group is first treated are omitted because of missing values, `xthdidregress` cannot assign the group to the appropriate cohort.

If you have information about the cohort values, instead of letting the command create a cohort variable, you can provide the cohort variable with the `usercohort()` option. Suppose you had a cohort variable, `mycohort`; then you would type

```
. xthdidregress twfe (registered best) (movie), group(breed) usercohort(mycohort)
```

Another possibility is to generate the cohort variable `mycohort` yourself using the `gencohort` command; this is helpful when you have missing information on covariates or the outcome but have enough information about the treatment. Suppose you had missing information about the control variable `best` but had full information about the treatment variable. Below, we drop information about the 2034 cohort to illustrate the point.

```
. replace best = . if year==2034 & breed==29|breed==55
(11 real changes made, 11 to missing)
```

These observations for year 2034 would not be used during estimation, but we have enough information in them to create our own cohort variable.

```
. gencohort mycohort, treat(movie) time(year) group(breed)
. list year breed movie best mycohort if breed==29, noobs sepby(breed)
```

year	breed	movie	best	mycohort
2031	Boxer	0	0	2034
2032	Boxer	0	0	2034
2033	Boxer	0	0	2034
2034	Boxer	1	.	2034
2035	Boxer	1	0	2034
2036	Boxer	1	0	2034
2037	Boxer	1	0	2034
2038	Boxer	1	0	2034
2039	Boxer	1	0	2034
2040	Boxer	1	0	2034

The `mycohort` variable can now be specified in the `usercohort()` option of `xthdidregress` to properly treat the Boxer breed as belonging to cohort 2034.



Stored results

`xthdidregress` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_clust)</code>	number of clusters
<code>e(N_panels)</code>	number of panels
<code>e(tmin)</code>	first time period
<code>e(tmax)</code>	last time period
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(xinteract)</code>	1 if covariate interactions are included, 0 otherwise

Macros

<code>e(cmd)</code>	<code>xthdidregress</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of outcome variable
<code>e(clustvar)</code>	name of cluster variable
<code>e(control_group)</code>	control group
<code>e(het_type)</code>	heterogeneity type for <code>twfe</code> estimator

e(over_type)	subpopulation over which each ATET is computed for twfe estimator
e(cohortvar)	name of cohort variable
e(usercohort)	name of user-specified cohort variable
e(ovar)	name of outcome variable
e(wtype)	weight type
e(wexp)	weight expression
e(marginsnotok)	predictions disallowed by margins
e(ivar)	panel identifier variable
e(timevar)	time variable
e(treatname)	name of treatment variable
e(basetime)	type of pretreatment base time
e(estat_cmd)	program used to implement estat
e(vce)	vctype specified in vce()
e(vcetype)	title used to label Std. err.
e(method)	estimator method
e(properties)	b V

Matrices

e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
e(cohort_count)	matrix with cohort count information

Functions

e(sample)	marks estimation sample
-----------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

Methods and formulas are presented under the following headings:

The model

The RA, IPW, and AIPW estimators

Panel data

The TWFE estimator

The model

`xthdidregress` estimates the treatment-effects parameters using the DID approach 1) with multiple periods, 2) with different timing when a unit is first treated, and 3) when the treatment effects can be heterogeneous across time and cohort. In particular, `xthdidregress` provides the RA, IPW, and AIPW estimators, outlined in [Callaway and Sant’Anna \(2021\)](#), and the TWFE estimator, outlined in [Wooldridge \(2021\)](#).

We observe data $\{y_{it}, \mathbf{x}_{it}, d_{it}, \mathbf{z}_{it}\}$ for unit i at time t with unit $i, i = 1, \dots, N$, and time t , with $t = 1, \dots, T$, where

1. y_{it} is the observed outcome;
2. $\mathbf{x}_{it}$ are covariates for the outcome model;
3. d_{it} is an indicator that equals one if an observation is treated or zero otherwise; and
4. $\mathbf{z}_{it}$ are the covariates for the treatment-assignment model.

Denote $y_{it}(g)$ as the potential outcome for unit i at time t if it was first treated at time g . Denote $y_{it}(0)$ as the potential outcome for unit i and t if it is never treated. All the units can be grouped into cohorts, and each cohort starts the treatment simultaneously. For example, cohort g means all the units start treatment at time g . Let G_{ig} be an indicator that equals one if unit i is first treated at time g . Then the units in cohort g can be denoted by $G_{ig} = 1$. When a unit i is never treated, we denote $G_{i0} = 1$. Thus, cohort 0 indicates all the units that are never treated. We assume that once a unit is treated, it will remain treated.

To reveal how the heterogeneous treatment effects evolve across cohorts and time, we are interested in estimating the ATET for each combination of cohort and time. Let $\theta(g, t)$ be the ATET for cohort g at time t , which is defined as

$$\theta(g, t) = \mathbf{E} \{y_t(g) - y_t(0) | G_g = 1\} \quad (\text{ATET})$$

where G_g equals 1 if a unit belongs to cohort g . All the four estimators provided in `xthdidregress` and `hdidregress` estimate $\theta(g, t)$ in equation (ATET). We cannot directly estimate $\theta(g, t)$ using equation (ATET) because the potential outcomes $y_t(g)$ and $y_t(0)$ are not observable.

Next, we will describe the RA, IPW, and AIPW estimators.

The RA, IPW, and AIPW estimators

To estimate the ATET for cohort g at time t , the RA, IPW, and AIPW estimators transform the estimation into a classical two groups and two periods difference-in-differences setup. Thus, we need to restrict the data to an estimation sample with only two groups and only two periods based on the values of g and t . For the two groups, one group comprises all observations in cohort g ; the other group comprises untreated observations not in cohort g , also known as a control group. For the two periods, one period is the data in time t ; the other period is a period when cohort g is not treated, also known as base time.

There are two ways to define the control group. One way is to use the units that are never treated as the control group. Let C^{NEV} be an indicator that equals one if a unit belongs to the never-treated group. In particular, $C^{\text{NEV}} = G_0$. Another way is to use the units not in cohort g and not yet treated at time t as the control group. Let $C^{\text{NY}}_{g,t}$ be an indicator that equals one if a unit belongs to the not-yet-treated group by time t . In particular, $C^{\text{NY}}_{g,t} = (1 - G_g)(1 - d_t)$. To simplify, we indicate control, in both cases, as $C^*_{g,t}$.

The definitions of the RA, IPW, and AIPW estimators depend on the definition of $C^*_{g,t}$, which can either be C^{NEV} or $C^{\text{NY}}_{g,t}$. However, regardless of the control group's choice, the estimators' definitions can always be written using the general notation $C^*_{g,t}$.

There are also two ways to define the base time. One way is to adaptively choose the base time for the pretreatment periods. When the adaptive method is used to compute the ATET for cohort g at time t , for the pretreatment periods, the base time is $t - 1$; for the posttreatment periods, the base time is $g - 1$. Another way is to use a common base time $g - 1$ for both pretreatment and posttreatment periods. The common base time is useful for identifying a violation of the parallel trends assumption in event studies as discussed in Roth (2024). To simplify the notation, we indicate the base time in both cases as t_0 .

Panel data

The estimands for RA, IPW, and AIPW are defined as follows.

The RA estimand is

$$\theta_{\text{RA}}(g, t) = \mathbf{E} \left[\frac{G_g}{\mathbf{E}(G_g)} \{y_t - y_{g-1} - m_{g,t}(\mathbf{x})\} \right] \quad (\text{RA})$$

where $m_{g,t}(\mathbf{x}) = \mathbf{E}(y_t - y_{g-1} | \mathbf{x}, C_{g,t}^* = 1)$.

The IPW estimand is

$$\theta_{\text{IPW}}(g, t) = \mathbf{E} \left(\left[\frac{G_g}{\mathbf{E}(G_g)} - \frac{\frac{p_{g,t}(\mathbf{z})C_{g,t}^*}{1-p_{g,t}(\mathbf{z})}}{\mathbf{E} \left\{ \frac{p_{g,t}(\mathbf{z})C_{g,t}^*}{1-p_{g,t}(\mathbf{z})} \right\}} \right] (y_t - y_{g-1}) \right) \quad (\text{IPW})$$

where

$$p_{g,t}(\mathbf{z}) = \text{Pr}(G_g = 1 | \mathbf{z}, G_g + C_{g,t}^* = 1) \quad (\text{Pz})$$

The AIPW estimand is

$$\theta_{\text{AIPW}}(g, t) = \mathbf{E} \left(\left[\frac{G_g}{\mathbf{E}(G_g)} - \frac{\frac{p_{g,t}(\mathbf{z})C_{g,t}^*}{1-p_{g,t}(\mathbf{z})}}{\mathbf{E} \left\{ \frac{p_{g,t}(\mathbf{z})C_{g,t}^*}{1-p_{g,t}(\mathbf{z})} \right\}} \right] \{y_t - y_{g-1} - m_{g,t}(\mathbf{x})\} \right) \quad (\text{AIPW})$$

Callaway and Sant’Anna (2021) show that the estimand for RA, IPW, and AIPW is the same as $\theta(g, t)$ in equation (ATET). In other words,

$$\theta(g, t) = \theta_{\text{RA}}(g, t) = \theta_{\text{IPW}}(g, t) = \theta_{\text{AIPW}}(g, t)$$

However, unlike the nonestimable $\theta(g, t)$ in equation (ATET), the estimands in equation (RA), (IPW), and (AIPW) are estimable because only the observed random variables such as $\mathbf{x}$, y , $\mathbf{z}$, G_g , and $C_{g,t}^*$ are used.

The identification of the estimators sheds light on how to estimate $\theta(g, t)$. The estimator can be generally divided into three steps:

1. Restrict the sample to time t and t_0 , and keep only the units in cohort g or in control group $C_{g,t}^*$. When option `basetime(adaptive)` is specified, $t_0 = g - 1$ if $t \geq g$ or $t_0 = t - 1$ if $t < g$. When option `basetime(common)` is specified, $t_0 = g - 1$. Only the paired observations are used. In other words, unit i will be used if it is observed at both times t and t_0 .
2. Use a parametric model to estimate the nuisance function for $p_{g,t}(\mathbf{z})$ and $m_{g,t}(\mathbf{x})$. In practice, we use a linear regression model to fit $m_{g,t}(\mathbf{x})$ and a logit regression model to fit $p_{g,t}(\mathbf{z})$. In particular, do the following:

- a. Run a linear regression of $y_t - y_{t_0}$ on $\mathbf{x}$ when $C_{g,t}^* = 1$.
 - b. Denote the linear predictor as $\widehat{m}_{g,t}(\mathbf{x})$.
 - c. Run a logit regression of G_g on $\mathbf{z}$.
 - d. Denote the prediction of the probability of a positive outcome as $\widehat{p}_{g,t}(\mathbf{z})$.
3. Plug in the nuisance function estimates $\widehat{m}_{g,t}(\mathbf{x})$, $\widehat{p}_{g,t}(\mathbf{z})$, or both into the estimating equation in equation (RA), (IPW), or (AIPW). Notice that the expectation operator $\mathbf{E}(\cdot)$ is replaced by the sample average.

The variance–covariance matrix for $\widehat{\theta}(g, t)$ for each cohort g and time t is computed using the influence function approach proposed in theorem 2 in Callaway and Sant’Anna (2021). The influence function approach is numerically equivalent to the generalized method-of-moments approach. However, it is much faster because it avoids computing the covariance matrix for the parameters in the nuisance functions $m_{g,t}(\mathbf{x})$ and $p_{g,t}(\mathbf{z})$. For more discussions on influence functions, see Hampel et al. (1986), Newey and McFadden (1994), and Jann (2020).

The TWFE estimator

A TWFE estimator fits

$$y_{it} = \alpha_i + \gamma_t + \mathbf{x}_{it}\boldsymbol{\beta} + d_{it}\tau + \epsilon_{it}$$

Wooldridge (2021) extends this model to incorporate interactions between the treatment, d_{it} , cohort, G_{ig} , posttreatment periods, and covariates. We define indicators for posttreatment periods as f_s with s going from q to T , where q is the first time period we observe treatment. For instance, f_q equals 1 if we are in time period q and 0 otherwise. If `noxinteract` has not been specified, the extended fixed-effects model is given by

$$\begin{aligned} y_{it} = & \eta + \sum_{g=q}^T G_{ig}\alpha_g + \sum_{s=q}^T f_s\gamma_s + \sum_{g=q}^T \sum_{s=g}^T d_{it}G_{ig}f_s\tau_{gs} + \sum_{g=q}^T G_{ig}\mathbf{x}_i\boldsymbol{\alpha}_g + \sum_{s=q}^T f_s\mathbf{x}_i\boldsymbol{\gamma}_s \\ & + \sum_{g=q}^T \sum_{s=g}^T d_{it}G_{ig}f_s\mathbf{x}_i\boldsymbol{\tau}_{gs} + \mathbf{x}_{it}\boldsymbol{\beta} + \epsilon_{it} \end{aligned} \quad (\text{TWFE})$$

We can fit equation (TWFE) using pooled ordinary least squares or a within estimator. We are going to use the estimator proposed by Mundlak (1978). This gives the same point estimates as using the within estimator of `xtreg ... , fe` for the parameters in equation (TWFE) but has different degrees of freedom because of the additional terms added by the Mundlak approach. Unlike within estimation, the Mundlak approach works for both repeated cross-sectional data and panel data. Also, it has good properties to obtain partial effects under various data-generating processes, as pointed out in Wooldridge (2019).

Given this model, the cohort-time treatment effects are a function of τ_{gs} , $\boldsymbol{\tau}_{gs}^{\mathbf{x}}$, and the covariates. We can compute these effects using `margins` by typing

```
. margins, dydx(d) at(year=q ... year=T) over(cohort) vce(unconditional)
```

where `d` is the treatment indicator, `year` indicates treatment times at which treatment will be evaluated using `at()`, and `cohort` is the treatment-time cohorts.

If `over(timecohort)` was specified with `xthdidregress`, we can obtain effects by typing

```
. margins, dydx(d) over(year cohort) vce(unconditional)
```

We use `vce(unconditional)` to account for the variation in the covariates. When there are no covariates specified or when `noxinteract` is specified, `vce(unconditional)` has no effect. Furthermore, if we include covariates but specify `noxinteract`, our equation reduces to

$$y_{it} = \eta + \sum_{g=q}^T G_{ig} \alpha_g + \sum_{s=q}^T f_s \gamma_s + \sum_{g=q}^T \sum_{s=q}^T d_{it} G_{ig} f_s \tau_{gs} + \mathbf{x}_{it} \beta + \epsilon_{it}$$

In this case, τ_{gs} are the treatment effects.

In practice, `xthdidregress` computes the treatment effects analytically rather than by use of `margins`. Specifically, a modified Mundlak regression model is fit. The modified regression interacts treatment indicators with demeaned covariates instead of the covariates themselves. By default, the covariates are demeaned by cohort-specific covariate means. If `over(timecohort)` is specified, the covariates are demeaned by cohort-time-specific means. Treatment-effect parameters can be estimated as coefficients of this regression rather than as linear combinations of regression coefficients.

The modified Mundlak regression is treated as being fit following a set of first-stage regressions of each covariate on cohort indicators. GMM-style standard errors account for variation in these first-stage regressions and are equivalent to the standard errors produced by `margins` with the `vce(unconditional)` option.

With the `hettype()` option, we reduce the complexity of equation (TWFE). In particular, if we ask for `hettype(time)`, we have

$$y_{it} = \eta + \sum_{g=q}^T G_{ig} \alpha_g + \sum_{s=q}^T f_s \gamma_s + \sum_{s=q}^T d_{it} f_s \tau_s + \sum_{g=q}^T G_{ig} \mathbf{x}_i \alpha_g^x + \sum_{s=q}^T f_s \mathbf{x}_i \gamma_s^x + \sum_{s=q}^T d_{it} f_s \mathbf{x}_i \tau_s^x + \mathbf{x}_{it} \beta + \epsilon_{it}$$

Now treatment varies over time but not over cohort. If we use the `hettype(cohort)` option, we have

$$y_{it} = \eta + \sum_{g=q}^T G_{ig} \alpha_g + \sum_{s=q}^T f_s \gamma_s + \sum_{g=q}^T d_{it} G_{ig} \tau_g + \sum_{g=q}^T G_{ig} \mathbf{x}_i \alpha_g^x + \sum_{s=q}^T f_s \mathbf{x}_i \gamma_s^x + \sum_{g=q}^T d_{it} G_{ig} \mathbf{x}_i \tau_g^x + \mathbf{x}_{it} \beta + \epsilon_{it}$$

Now treatment varies over cohort but not over time.

When the `controlgroup(notyet)` option is specified, the G_{ig} indicator excludes the last treated cohort. As discussed in Wooldridge (2021), when every group is eventually treated, we cannot identify the treatment effect for this cohort. It is therefore sensible to use the last treated cohort as a control group. When some of the units in our sample are never treated, we can always identify all cohorts, and the `twfe` estimator will always revert to using `controlgroup(never)`.

Acknowledgments

We thank Fernando Rios-Avila of the Levy Economics Institute of Bard College for helpful conversations and advice and for his contributions in developing heterogeneous treatment-effects software and accompanying material. His community-contributed commands `csdid` and `drdid` implement some of the methods in this entry and a few others we have yet to include.

We thank Pedro H. C. Sant’Anna of Emory University for helpful conversations and advice and for his continuing contributions to the heterogeneous treatment-effects literature.

We thank Jeff Wooldridge of Michigan State University for helpful conversations about extended two-way fixed-effects estimation.

We also thank Austin Nichols of Amazon for his advice, for his insights, and for organizing the WHYDC causal inference conference.

References

- Callaway, B., and P. H. C. Sant’Anna. 2021. Difference-in-differences with multiple time periods. *Journal of Econometrics* 225: 200–230. <https://doi.org/10.1016/j.jeconom.2020.12.001>.
- de Chaisemartin, C., and X. D’Haultfœuille. 2023. Two-way fixed effects and differences-in-differences with heterogeneous treatment effects: A survey. *Econometrics Journal* 26: C1–C30. <https://doi.org/10.1093/ectj/utac017>.
- Freyaldenhoven, S., C. B. Hansen, J. Pérez Pérez, J. M. Shapiro, and C. Carreto. 2025. *xtevent: Estimation and visualization in the linear panel event-study design*. *Stata Journal* 25: 97–135.
- Ghirlanda, S., A. Acerbi, and H. Herzog. 2014. Dog movie stars and dog breed popularity: A case study in media influence on choice. *PLOS ONE* 9: e106565. <https://doi.org/10.1371/journal.pone.0106565>.
- Hampel, F. R., E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. 1986. *Robust Statistics: The Approach Based on Influence Functions*. New York: Wiley. <https://doi.org/10.1002/9781118186435>.
- Jann, B. 2020. Influence functions continued. A framework for estimating standard errors in reweighting, matching, and regression adjustment. Working Papers 35, University of Bern Social Sciences. <https://ideas.repec.org/p/bss/wpaper/35.html>.
- Mundlak, Y. 1978. On the pooling of time series and cross section data. *Econometrica* 46: 69–85. <https://doi.org/10.2307/1913646>.
- Newey, W. K., and D. L. McFadden. 1994. “Large sample estimation and hypothesis testing”. In *Handbook of Econometrics*, edited by R. F. Engle and D. L. McFadden, vol. 4: 2111–2245. Amsterdam: Elsevier. [https://doi.org/10.1016/S1573-4412\(05\)80005-4](https://doi.org/10.1016/S1573-4412(05)80005-4).
- Roth, J. 2024. Interpreting event-studies from recent difference-in-differences methods. ArXiv Working Paper No. arXiv:2401.12309, <https://doi.org/10.48550/arXiv.2401.12309>.
- Roth, J., P. H. C. Sant’Anna, A. Bilinski, and J. Poe. 2022. What’s trending in difference-in-differences? A synthesis of the recent econometrics literature. arXiv:2201.01194 [econ.EM], <https://doi.org/10.48550/arXiv.2201.01194>.
- Wooldridge, J. M. 2019. Correlated random effects models with unbalanced panels. *Journal of Econometrics* 211: 137–150. <https://doi.org/10.1016/j.jeconom.2018.12.010>.
- . 2021. Two-way fixed effects, the two-way Mundlak regression, and difference-in-differences estimators. Working paper, Department of Economics, Michigan State University, East Lansing, MI. <https://doi.org/10.2139/ssrn.3906345>.

Also see

- [CAUSAL] **hdidregress postestimation** — Postestimation tools for hdidregress and xthdidregress⁺
- [CAUSAL] **DID intro** — Introduction to difference-in-differences estimation
- [CAUSAL] **didregress** — Difference-in-differences estimation⁺
- [CAUSAL] **gencohort** — Create a cohort variable
- [CAUSAL] **hdidregress** — Heterogeneous difference in differences⁺
- [CAUSAL] **teffects intro** — Introduction to treatment effects for observational data
- [CAUSAL] **teffects intro advanced** — Advanced introduction to treatment effects for observational data
- [CAUSAL] **xtswitchdid** — Difference in differences with switching treatments for panel data⁺
- [U] **20 Estimation and postestimation commands**

⁺This command includes features that are part of [StataNow](#).

Description

`hdidregress` and `xthdidregress` allow the same postestimation commands; see [\[CAUSAL\] hdidregress postestimation](#).

⁺This command is part of [StataNow](#).[Description](#)[Options](#)[Acknowledgments](#)[Quick start](#)[Remarks and examples](#)[References](#)[Menu](#)[Stored results](#)[Also see](#)[Syntax](#)[Methods and formulas](#)

Description

`xtswitchdid` estimates event-study effects for panel data using the heterogeneous difference-in-differences (DID) estimator of [de Chaisemartin and D'Haultfœuille \(2026\)](#). These effects are dynamic treatment effects that are of interest in event studies, which explore how treatment effects change over time. Treatments can be binary or multivalued. Switching (nonabsorbing) treatments are also allowed—subjects may switch in and out of treatment or to different levels of treatment.

Quick start

Estimate event-study effects of treatment `treat` on outcome `y` with group `grpvar` on `xtset` data

```
xtswitchdid (y) (treat), group(grpvar)
```

Same as above, but model `y` using covariate `x`

```
xtswitchdid (y x) (treat), group(grpvar)
```

Same as above, but estimate effects for at most three periods of exposure

```
xtswitchdid (y x) (treat), group(grpvar) neffects(3)
```

Same as above, but request raw (nonnormalized) effects

```
xtswitchdid (y x) (treat), group(grpvar) neffects(3) raw
```

Same as above, but request effects only for switchers-in, and require the switchers to be common across all three effects

```
xtswitchdid (y x) (treat), group(grpvar) neffects(3) raw ///
    switchgroup(in) commonswitchers
```

Estimate path-specific treatment effects for a treatment path selected by indicator variable `pathvar`

```
xtswitchdid (y) (treat), group(grpvar) path(pathvar)
```

Menu

Statistics > Causal inference/treatment effects > Continuous outcomes > Heterogeneous DID > Panel-data DID with switching treatments

Syntax

```
xtswitchdid (ovar [omvarlist]) (tvar) [if] [in] [weight], group(groupvar) [options]
```

ovar is the outcome of interest.

omvarlist specifies the covariates in the outcome model and may contain factor variables; see [U] 11.4.3 Factor variables.

tvar indicates the treatment level for each observation and may be binary or multivalued.

groupvar is a categorical variable that indicates the group level at which the treatment occurs.

options	Description
Model	
* <u>group</u> (<i>groupvar</i>)	specify group variable
<u>super</u> group(<i>supergroupvar</i>)	specify supergroup variable
<u>switch</u> group(<i>sgtype</i>)	specify the type of switcher group; default is switchgroup(all)
<u>control</u> group(<i>cgtype</i>)	specify the type of control group; default is controlgroup(all)
<u>path</u> (<i>pathspec</i>)	request path-specific treatment effects for the specified treatment path
<u>neff</u> ects(#)	specify the number of event-study effects to estimate; default is all possible effects
<u>common</u> switchers	request that a common set of switchers be used for all event-study effects
<u>raw</u>	request event-study effects that have not been normalized
<u>no</u> listwise	do not automatically drop observations with missing values in some variables
SE/Robust	
<u>vce</u> (<i>vcetype</i>)	<i>vcetype</i> may be <u>cluster</u> <i>clustvar</i> , <u>robust</u> , <u>bootstrap</u> , or <u>jackknife</u>
Reporting	
<u>level</u> (#)	set confidence level; default is level(95)
<u>show</u> supergroups	show supergroups in the initial treatment table
<u>display</u> _options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
<u>coef</u> legend	display legend instead of statistics

*group() is required.

If the variable specified by group() is coarser than the panel variable, then *omvarlist* may not contain factor variables.

A panel variable and a time variable must be specified using xtset; see [XT] xtset.

by, collect, and statsby are allowed; see [U] 11.1.10 Prefix commands.

weights are allowed; see [U] 11.1.6 weight. Weights must be constant within panels.

coeflegend does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

<i>sgtype</i>	Description
all	include both switchers-in and switchers-out; the default
in	include only switchers-in
out	include only switchers-out

<i>cgtype</i>	Description
all	use both never-switchers and not-yet-switchers as controls; the default
never	use only never-switchers as controls

Options

Model

`group(groupvar)` specifies a group variable that indicates the group level at which treatment occurs. *groupvar* may be, for example, states, counties, or hospitals. `group()` also defines the clusters for the default cluster-robust standard errors. `group()` is required. *groupvar* may be the panel variable defined in `xtset`, which must in any case be nested within the levels defined by *groupvar*.

`supergroup(supergroupvar)` specifies a supergroup variable within levels of which groups can be compared with each other. *groupvar* must be nested in *supergroupvar*. For example, if *groupvar* specifies counties, *supergroupvar* may specify states. If no supergroup variable is specified, switchers are compared with all available control groups with the same initial treatment value. If a supergroup variable is specified, however, switchers are compared with only control groups that are also in the same supergroup.

`switchgroup(sgtype)` specifies the groups to use as switchers. Switchers are either switchers-in, meaning their first treatment different from their initial treatment is greater than the initial treatment, or switchers-out, meaning their first treatment different from their initial treatment is less than the initial treatment. *sgtype* can be one of `all` (referring to all switchers), `in` (referring to switchers-in), or `out` (referring to switchers-out). The default is `switchgroup(all)`.

`controlgroup(cgtype)` specifies the groups to use as controls. Control groups are either never-switchers, meaning that they keep the same value of the treatment variable until the last period, or not-yet-switchers, meaning that they keep their initial treatment value up to a specific period. *cgtype* can be `all` (referring to all possible control groups) or `never` (referring only to the never-switchers). The default is `controlgroup(all)`.

`path(pathspec)` specifies a treatment path for which treatment effects will be estimated. Only those switchers whose path matches the indicated path will be included in the estimation.

pathspec may be specified as *numlist*, indicating treatment values starting at the period before treatment. For example, `0 2 2 2` indicates a treatment path where groups have an initial treatment value of 0 and then stay at the treated value of 2 for at least 3 periods.

pathspec may alternatively be an indicator variable, where observations with the desired path have the value 1 (or any positive number) and other observations have the value 0. Path variables must indicate only one distinct path, which will be truncated at the length specified by `neffects()` if needed. If multiple groups have the same treatment path, the path variable need not indicate the path in every such group; an indicator for one such group is sufficient to define the path. Path variables can be generated using `estat paths` or constructed manually.

`neffects(#)` specifies the maximum number of event-study effects to estimate, starting from exposure $\ell = 1$. By default, `xtswitchdid` estimates and reports as many effects as are available in the data.

`commonswitchers` requests that each event-study effect be estimated using the same set of switchers. If `commonswitchers` is not specified, all available switchers are used for each effect, which typically means that shorter exposures are estimated using much more data than longer exposures. Specifying `commonswitchers` ensures that the same set of data is used for all estimated effects. Note that if `commonswitchers` and `neffects()` are specified together, the common set of switchers will be common up to the maximum number of effects specified in `neffects()`.

`raw` disables the normalization of event-study effects. The default when this option is not specified is to normalize the event-study effect for exposure ℓ by the average incremental treatment dose experienced by switching groups, relative to their baseline treatment, over the ℓ periods starting from when they first switch.

`nolistwise` requests that the default behavior of dropping observations with missing values in any variables used for estimation be disabled. Instead, observations where the outcome or covariates are missing may still be used to infer treatment paths.

SE/Robust

`vce(vctype)` specifies the type of standard error reported, which includes types that allow for intra-group correlation (`cluster clustvar`), that are robust to intragroup correlation among group variables (`robust`), and that use bootstrap or jackknife sampling done at the panel level (`bootstrap`, `jackknife`); see [R] [vce_option](#).

`vce(cluster clustvar)` with the variable specified in `group(groupvar)` is the default. If `clustvar` is specified as another variable, it must nest the variable specified in `group(groupvar)`.

Specifying `vce(robust)` is equivalent to specifying `vce(cluster clustvar)`, where `clustvar` is the variable specified in the `group(groupvar)` option.

Reporting

`level(#)`; see [R] [Estimation options](#).

`showsupergroups` requests that the initial treatment table, printed above the table of estimates, tabulate groups by initial treatment and supergroup, rather than only by initial treatment. If no variable is specified in `supergroup()`, this option has no effect.

`display_options`: `noci`, `nopvalues`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

The following option is available with `xtswitchdid` but is not shown in the dialog box:

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

Researchers often want to study the effects of a treatment, such as a policy or intervention, on an outcome for a group by exploiting variation in the timing or presence of treatment. In some cases, these treatments are binary, meaning that each group is either treated or untreated at any point in time, and absorbing, meaning that a group that is treated remains treated in subsequent periods.

In many applications, however, the treatments that are of interest are multivalued. For example, a treatment variable may count the number of active interventions in a given jurisdiction and thus take any nonnegative integer value. In the examples below, we take the number of new safety rules in a minor hockey league, ranging from 0 to 3, as a treatment.

Treatments can also be switching rather than absorbing. A treated group, after entering treatment, may transition to a higher level of treatment or lower level of treatment or exit treatment altogether. A hockey league may introduce three new safety rules in one year but repeal one of them two years later.

Furthermore, groups need not be first observed as untreated. A hockey league may have always had one safety rule but within the period of our study decide to transition to two safety rules—or vice versa.

`xtswitchdid` estimates event-study effects for panel data using the heterogeneous DID estimator of [de Chaisemartin and D’Haultfœuille \(2026\)](#), which allows for multivalued and switching treatment variables and does not require that groups be first observed at treatment level 0.

These event-study effects are estimated at each of multiple exposures, where the ℓ th effect is associated with having experienced a first treatment change ℓ periods ago. The effects can be compared with the dynamic treatment effects that are estimated by `estat aggregation`, dynamic after `xthdiddregress`. They cannot, however, be given the same interpretation as a simple aggregation of average treatment effects on the treated (ATETs). In the setting of `xthdiddregress`, treatments are binary and absorbing, so groups are distinguished from each other only by the timing of their treatment. In the more general setting of `xtswitchdid`, we must also consider the magnitude of treatments, the timing of switches between multiple treatment levels, and the initial level of treatment.

With `xtswitchdid`, the potential outcomes in the [de Chaisemartin and D’Haultfœuille \(2026\)](#) framework are defined for paths of treatment for groups. In contrast, `xtdidregress` estimates homogeneous treatment effects defined in terms of the potential outcome for group g being treated $Y_g(1)$ and the potential outcome for group g being untreated $Y_g(0)$. Also different is `xthdiddregress`, which estimates heterogeneous treatment effects defined in terms of potential outcomes that are functions of treatment cohort, where treatment cohort refers to the point in time when a group is first exposed to treatment.

To illustrate, suppose treatment is binary and there are three time periods. We could consider the following potential outcomes: $Y_g(0, 0, 0)$, $Y_g(0, 1, 0)$, and $Y_g(0, 1, 1)$. Here $Y_g(0, 0, 0)$ is the potential outcome of having been exposed to treatment level 0 in all three periods. $Y_g(0, 1, 0)$ is the outcome of having been exposed to treatment level 0 in the first and third period and to treatment level 1 in the second period. Similarly, $Y_g(0, 1, 1)$ is the outcome of having been exposed to treatment level 0 in the first period and to treatment level 1 in the second and third periods. In settings like this, with only a few distinct treatment paths, we could use the `path()` option in `xtswitchdid` to isolate, say, the 0 1 1 treatment path, and estimate the effect $E\{Y_g(0, 1, 1) - Y_g(0, 0, 0)\}$. In this simple case, we can interpret the effect as an aggregation of ATETs, which is the same way we would interpret the results from `estat aggregation`, dynamic after `xthdiddregress`.

In general, however, there can be too many paths and too little data for each path for this to be feasible. By default, `xtswitchdid` aggregates over groups that have the same initial treatment value, even if they have different treatment paths. For example, in the case above, we would aggregate the effects for the groups exposed to treatment path 0 1 0 together with those for the groups exposed to treatment path 0 1 1, even though 0 1 0 is only “weakly stronger” as a treatment than the initial treatment value of 0.

In this way, we concern ourselves mainly with each group’s “first switch” away from its initial treatment, while allowing the timing of the first switch to be staggered across groups. Accordingly, we can think of the treatment designs where `xtswitchdid` is applicable as staggered, first-switch designs.

For each group, we consider the effect of its observed treatment path relative to the counterfactual path where it had remained at its initial treatment value. In the simple case described above, the counterfactual path is 0 0 0. Thus, we aggregate over the effects $E\{Y_g(0, 1, 0) - Y_g(0, 0, 0)\}$ and $E\{Y_g(0, 1, 1) - Y_g(0, 0, 0)\}$. Because we do not observe the counterfactual path for any group g , we rely on the presence of control groups—groups that start at the same initial value as the group of interest but that have experienced no treatment switch over the relevant period of time.

The ℓ th event-study effect estimated by `xtswitchdid` can then be thought of as the average effect of being exposed to a treatment weakly stronger than a group’s initial treatment state after ℓ periods, where the average is conditional on the data. By default, the estimated effects are normalized by the incremental treatment dose across the ℓ periods starting with the period where the treatment is first switched, facilitating comparisons across exposures. This normalization also ensures that the ℓ th effect is a weighted average of the effects of the current treatment and the first through $\ell - 1$ th treatment lags on the outcome. For example, consider a group with a baseline treatment of 1, which experiences the path 1 3 2. Here the first event-study effect, for $\ell = 1$, would be normalized by a constant of $(3 - 1) = 2$, because treatment in the first switched period, 3, is higher than the baseline treatment by 2. The second event-study effect would be normalized by a constant of $(3 - 1) + (2 - 1) = 3$ and would consist of a weighted average of the effect of the current treatment, weighted by $(2 - 1)/3 = 1/3$, and the effect of the first lagged treatment, weighted by $(3 - 1)/3 = 2/3$.

Because some groups switch “in”, meaning their first treatment switch is to a level higher than their initial treatment level, and some groups switch “out”, meaning their first treatment switch is to a level lower than their initial treatment level, the group-specific effect of a treatment switch is multiplied by -1 for switchers-out before aggregating across groups. In other words, we assume that a unit increase in the treatment variable has the opposite effect of a unit decrease in the treatment variable.

When aggregating to estimate effects, `xtswitchdid` excludes observations for groups starting from time periods where the group has experienced treatment levels both higher and lower than its initial treatment levels. In other words, a group’s treatment paths can freely vary while remaining weakly higher than the initial treatment level (switchers-in) or freely vary at levels weakly lower than the initial treatment level (switchers-out), but it must not do both.

Data must be `xtset` before using `xtswitchdid`. When treatments and outcomes are observed periodically (such as every four years), the `delta()` option in `xtset` can be used to specify the relevant period.

For panel-data applications where treatments are binary and absorbing, `xthdidregress` can be used; see [CAUSAL] [xthdidregress](#). When, in addition, treatment effects are thought not to vary across time or treatment cohort, `xtddidregress` can be used; see [CAUSAL] [didregress](#).

▷ Example 1: Estimating event-study effects

We have fictional data on hockey teams in 30 minor hockey leagues in North America and are interested in the effect of the number of new safety rules, `safety`, on the number of player injury days, `injdays`, per team each season, normalized by season length. We hypothesize that exposure to these safety rules reduces `injdays`. Each safety rule is a rule change made at the league level. For example, one rule may require mouthguards, and another rule may set a minimum age for body checking.

Our panel has 462 teams across the 30 leagues, each of which is observed for some of or all the years between 2010 and 2025, with the exception of 2020, a pandemic year where outcomes and covariates are missing for all teams. We also observe the region in which each league is located, `region`, the team win percentages each season transformed by the logit function, `lwinpct`, and a measure of the intensity of

each team’s schedule, `schedule`. `schedule` is a function of both the number of games played per season and a subjective measure of the competitiveness of each team’s schedule and takes values between 1 and 6.

We can load our data and tabulate the treatment values by year:

```
. use https://www.stata-press.com/data/r19/hockey
(Fictional minor hockey team injury data)
. tabulate year safety
```

Year	Number of safety measures				Total
	0	1	2	3	
2010	213	121	41	0	375
2011	215	122	42	0	379
2012	43	30	306	0	379
2013	96	48	314	0	458
2014	100	34	328	0	462
2015	100	34	328	0	462
2016	100	34	328	0	462
2017	156	34	261	11	462
2018	156	34	261	11	462
2019	156	34	261	11	462
2020	141	34	233	11	419
2021	141	34	233	11	419
2022	123	34	85	177	419
2023	123	34	85	177	419
2024	123	34	85	177	419
2025	123	34	85	177	419
Total	2,109	729	3,276	763	6,877

`safety` takes four distinct values in the data: 0, 1, 2, and 3. In 2010, the first year of our panel, we observed 375 hockey teams, of which 213 had implemented no safety rules, 121 had implemented 1 safety rule, and 41 had implemented 2 safety rules. In subsequent years, leagues generally moved to higher levels of the treatment. Not easily observable here is that some groups in fact moved to lower levels of the treatment in some years. For example, we see that between 2016 and 2017, the number of teams implementing 0 safety rules rose from 100 to 156. We will use other tools to examine treatment paths in more detail.

Before obtaining event-study effects, we first `xtset` our data, with `team` as the panel variable and `year` as the time variable:

```
. xtset team year
Panel variable: team (unbalanced)
Time variable: year, 2010 to 2025
Delta: 1 unit
```

To obtain estimates of event-study effects using `xtswitchdid`, we use syntax similar to that of `xthdidregress` and other DID commands. Our outcome variable, `injdays`, is specified in the first set of parentheses, along with our covariates, `lwinpct` and `schedule`. The second set of parentheses contains the treatment variable, `safety`. Our panel and time variables are already set, but we must specify the group variable using the `group()` option. Here the group variable is the same as the panel variable, which in this case is `team`.

```
. xtswitchdid (injdays lwinpct schedule) (safety), group(team)

Treatment and time information
Panel variable:  team
Time variable:  year
Time interval:  2010 thru 2025
Treatment level: team
Control:        safety maintaining baseline
Treatment:      safety changing from baseline
```

Initial treatment	Groups	Obs	Switch out	Switch in	Absorbing
0	272	3,749		1 2 3	No
1	140	1,956		2 3	No
2	50	723		3	Yes

DID with switching treatments

Number of obs = 6,428
Number of panels = 462
Number of groups = 462
(Std. err. adjusted for 462 clusters in team)

	Normalized effect	Robust std. err.	z	P> z	[95% conf. interval]	
Exposure						
1	-6.871766	2.679854	-2.56	0.010	-12.12418	-1.619348
2	-2.75066	1.380428	-1.99	0.046	-5.45625	-.0450695
3	-3.061524	.8162018	-3.75	0.000	-4.66125	-1.461798
4	-1.038777	.7065305	-1.47	0.141	-2.423551	.3459975
5	-.9033563	.5854862	-1.54	0.123	-2.050888	.2441756
6	-1.140814	.4573033	-2.49	0.013	-2.037112	-.2445157
7	-.9005077	.4021147	-2.24	0.025	-1.688638	-.1123773
8	-1.239846	.4123365	-3.01	0.003	-2.048011	-.4316814
9	.7121286	.9774432	0.73	0.466	-1.203625	2.627882
10	-1.615785	.3346074	-4.83	0.000	-2.271603	-.9599661
11	-1.405509	.457877	-3.07	0.002	-2.302931	-.5080862
12	-1.613272	.430911	-3.74	0.000	-2.457842	-.7687025
13	-1.213585	.3641668	-3.33	0.001	-1.927339	-.4998312
14	-.3144064	.3203691	-0.98	0.326	-.9423183	.3135055

Note: Effects computed using covariates.

The header gives information about the model and the treatment variable. The initial treatment table shows that there are three values of the treatment variable for which we observe teams with that value as their initial treatment. This is important because our estimator computes differences only between groups that share an initial treatment value.

Most teams start with zero safety rules implemented, as reflected in the Groups column. The number of observations reflects the fact that each team is observed at multiple time periods. Note that both switchers and groups that never switch, but are used as control groups, are counted in these summary statistics.

In this dataset, all treatment switches away from the initial treatment value are switches “in” to treatment, meaning to a higher level of treatment as opposed to a lower. The Switch in column lists the distinct treatment values that groups in each initial-treatment-value group are observed to eventually take. Among the groups whose initial treatment value is 0, there are teams whose first “switch” is to 1, others whose first switch is to 2, and others whose first switch is to 3.

The Absorbing column indicates whether all teams with a particular initial treatment have a treatment path that is absorbing. The Yes in the third row indicates that all teams with initial treatment 2 have absorbing treatment paths, while the Nos in the first and second rows indicate that there are at least some groups with initial treatment levels 0 and 1 whose treatment paths are not absorbing. Here an absorbing treatment path is one such that when the group has switched from its initial treatment, it does not again change its treatment to another value.

The table of results gives normalized event-study effects as defined in [Methods and formulas](#). The effect at exposure 1 is the average effect on injdays of increasing the number of safety rules by one, as measured in the first treated period, relative to a counterfactual where the treatment had remained at the initial number of safety rules. Any team that changed its number of safety rules, whether to a treatment level higher or lower than its initial treatment level, experienced it for at least one period and was thus used to compute this effect.

The effect at exposure 5 is the average effect of additional safety rules five periods after the most recent preswitch period, relative to a counterfactual where the treatment never changed from the initial value. Any team that was observed for five periods starting with its treatment switch was included in the estimation of this effect, provided we have at least one control group with the same initial treatment value. For example, a team that began at zero safety rules and increased to two safety rules in 2012 and was observed until at least 2016 would be included in the computation of the effect for exposure 5.

We may be interested in a summary measure of the effect of safety on injdays, rather than in the effect of a certain exposure length. The event-study effects can be aggregated over exposures to produce an average total effect that we can interpret as the average effect of being exposed to a unit of treatment, conditional on the distribution of exposures we observe. We can compute an average total effect using the estat total postestimation command.

```
. estat total
Average total effect per unit of treatment      Number of obs   = 6,428
                                                Number of panels = 462
                                                Number of groups = 462
                                                (Std. err. adjusted for 462 clusters in team)
```

injdays	Robust					
	Effect	std. err.	z	P> z	[95% conf. interval]	
safety	-8.65764	2.281467	-3.79	0.000	-13.12923	-4.186046

We see evidence for a negative effect of `safety` on `injdays`, where an additional unit of treatment corresponds to an average of 8.7 fewer player injury days per team in the seasons following the treatment change.



► Example 2: Visualizing event-study effects

We can use the `estat eventplot` postestimation command to visualize normalized event-study effects by exposure:

```
. estat eventplot
```

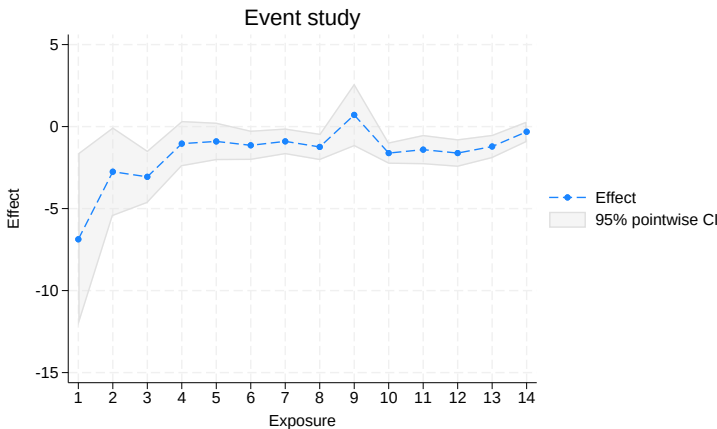


Figure 1. Event-study effects by exposure

In the first year of exposure to a treatment switch, we see an effect that is negative but imprecisely estimated. In exposures 2 through 8, we see an effect that declines in magnitude as teams are exposed to treatment for a longer period—maybe enforcement slips over time, or maybe teams gradually adjust their style of play in response to rule changes.

Note the imprecision in our estimate for the effect at exposure 9. This reflects the fact that, for most groups, the ninth period of exposure to switching from initial treatment is 2020, when no data points were observed.



► Example 3: Treatment paths

When treatments are multivalued or switching, interpretation becomes more challenging than in the standard case of binary and absorbing treatments. We may wish to isolate a single treatment path to measure its treatment effect, which can be interpreted directly. The postestimation utility estat paths lists the treatment paths observed in the data:

```
. estat paths
note: pathlen() not specified; using path length 14.
note: variable _did_paths, which encodes a unique value for each treatment
      path, was added to the dataset.
```

Distribution of treatment paths

Path length: 14, not counting initial treatment
Treatment: team changing initial treatment level

Paths	Freq.	Percent	Cum.
0 1 1 2 2 2 2 2 . 2 2 2 2 2	14	4.56	4.56
0 2 . 2 2 2 2 2	1	0.33	4.89
0 2 2 2 . 2 2 2 2 . 2 3 3 3 3	3	0.98	5.86
0 2 2 2 2 2 0 0 . . 0 0 0 0 0	1	0.33	6.19
0 2 2 2 2 2 0 0 0 . 0 0 0 0 0	55	17.92	24.10
0 2 2 2 2 2 2 2 . . 2 3 3 3 3	1	0.33	24.43
0 2 2 2 2 2 2 2 . 2 3 3 3 3 .	1	0.33	24.76
0 2 2 2 2 2 2 2	11	3.58	28.34
0 2 2 2 2 2 2 2 . . 3 3 3 3 3	2	0.65	28.99
0 2 2 2 2 2 2 2 2 . 2 3 3 3 .	1	0.33	29.32
0 2 2 2 2 2 2 2 2 . 2 3 3 3 3	82	26.71	56.03
0 3 3 3 3	18	5.86	61.89
1 2 . 2 2 2 2 2 2 . 2 2 2 2 2	1	0.33	62.21
1 2 2 2 2 2 2 . 2 . 2 2 2 2 2	1	0.33	62.54
1 2 2 2 2 2 2 2 2	16	5.21	67.75
1 2 2 2 2 2 2 2 2 . 2 . 3 3 3	1	0.33	68.08
1 2 2 2 2 2 2 2 2 . 2 2 2 2 2	30	9.77	77.85
1 2 2 2 2 2 2 2 2 . 2 3 3 3 3	57	18.57	96.42
2 3 3 3 . 3 3 3 3 3	11	3.58	100.00
Total	307	100.00	

Below the command, a note informs us that a default path length has been set because we have not chosen one using the pathlen() option. In cases of long or numerous treatment paths, it may be useful to distinguish paths only up to a smaller number of periods. A second note tells us that a new variable has been created that records the treatment path for each observation.

The output shows us the treatment paths, starting from the last period before the treatment changed. Note specifically that the *j*th element of the path does not necessarily correspond to the *j*th period in the data or to any fixed period. It is possible for groups to have the same treatment path despite starting treatment at different times, though that does not occur in this dataset.

Missing values reflect time periods along the path where no treatment is observed or where observations have been dropped because of missing data. Every treatment path in this dataset happens to have at least one missing value corresponding to the year 2020, when no treatments were observed.

We saw earlier that there was at least one treatment path that was not absorbing, and now we can see them in the output. The fourth- and fifth-listed paths show there are 56 teams who switch into treatment (2 safety rules) and then, after 5 periods, switch back out of it.

We may be interested in event-study effects for these treatment paths. We can run `estat paths` with the `generate()` option to obtain path indicators that we can use in the `xtswitchdid` command with the `path()` option. (Alternatively, we could type the path as *numlist*.)

```
. quietly estat paths, generate(safetypath)
. describe safetypath*
```

Variable name	Storage type	Display format	Value label	Variable label
safetypath1	byte	%8.0g		_did_paths==0 1 1 2 2 2 2 2 2 . 2 2 2 2 2
safetypath2	byte	%8.0g		_did_paths==0 2 . 2 2 2 2 2 2
safetypath3	byte	%8.0g		_did_paths==0 2 2 2 . 2 2 2 2 . 2 3 3 3 3
safetypath4	byte	%8.0g		_did_paths==0 2 2 2 2 2 0 0 . . 0 0 0 0 0
safetypath5	byte	%8.0g		_did_paths==0 2 2 2 2 2 0 0 0 . 0 0 0 0 0
safetypath6	byte	%8.0g		_did_paths==0 2 2 2 2 2 2 2 . . 2 3 3 3 3
safetypath7	byte	%8.0g		_did_paths==0 2 2 2 2 2 2 2 . 2 3 3 3 3 .
safetypath8	byte	%8.0g		_did_paths==0 2 2 2 2 2 2 2 2
safetypath9	byte	%8.0g		_did_paths==0 2 2 2 2 2 2 2 2 . . 3 3 3 3
safetypath10	byte	%8.0g		_did_paths==0 2 2 2 2 2 2 2 2 . 2 3 3 3 .
safetypath11	byte	%8.0g		_did_paths==0 2 2 2 2 2 2 2 2 . 2 3 3 3 3
safetypath12	byte	%8.0g		_did_paths==0 3 3 3 3
safetypath13	byte	%8.0g		_did_paths==1 2 . 2 2 2 2 2 2 . 2 2 2 2 2
safetypath14	byte	%8.0g		_did_paths==1 2 2 2 2 2 2 2 . 2 . 2 2 2 2 2
safetypath15	byte	%8.0g		_did_paths==1 2 2 2 2 2 2 2 2
safetypath16	byte	%8.0g		_did_paths==1 2 2 2 2 2 2 2 2 . 2 . 3 3 3
safetypath17	byte	%8.0g		_did_paths==1 2 2 2 2 2 2 2 2 . 2 2 2 2 2
safetypath18	byte	%8.0g		_did_paths==1 2 2 2 2 2 2 2 2 . 2 3 3 3 3
safetypath19	byte	%8.0g		_did_paths==2 3 3 3 . 3 3 3 3 3

The fifth-listed path is generated as safetypath5 and corresponds to the path starting with 0 2 2 2 2 2 0 0 0. We can use this variable to estimate path-specific effects:

```
. xtswitchdid (injdays lwinpct schedule) (safety), group(team) path(safetypath5)
note: 4 groups with only a single available time period were dropped.
note: effect of exposure 9 cannot be estimated because of missing data.

DID with switching treatments                                Number of obs   = 2,222
                                                            Number of panels =  268
                                                            Number of groups =  268

Treatment path: (0 2 2 2 2 2 0 0 0 . 0 0 0 0 0)
Treatment type: Switcher-in, nonabsorbing

                               (Std. err. adjusted for 268 clusters in team)
```

	Normalized effect	Robust std. err.	z	P> z	[95% conf. interval]	
Exposure						
1	-8.803413	3.036455	-2.90	0.004	-14.75476	-2.85207
2	-3.423294	1.351155	-2.53	0.011	-6.071509	-.7750789
3	-2.574106	.9883007	-2.60	0.009	-4.51114	-.6370724
4	-1.090292	.8124545	-1.34	0.180	-2.682673	.50209
5	-.3275366	.6412151	-0.51	0.609	-1.584295	.9292218
6	-1.138047	.5870595	-1.94	0.053	-2.288663	.0125683
7	-1.17035	.5815419	-2.01	0.044	-2.310152	-.0305493
8	-1.320975	.5834758	-2.26	0.024	-2.464567	-.1773839
9	0	(omitted)				
10	-.9877028	.5803887	-1.70	0.089	-2.125244	.1498381
11	-.9415612	.927531	-1.02	0.310	-2.759489	.8763662
12	-1.390374	.9659295	-1.44	0.150	-3.283561	.5028133
13	-.3080592	.9232056	-0.33	0.739	-2.117509	1.501391
14	-.8733448	.8459017	-1.03	0.302	-2.531282	.7845921

Note: Effects computed using covariates.

The output now omits the first table about initial treatment groups because we are looking only at one path. The header displays the path, informing us that groups with this treatment path are switchers-in and that the treatment is nonabsorbing.

For teams that follow this path, switching from 0 to 2 and then back to 0 after 5 periods, we expect an effect of 8.8 fewer injury days in the first period after the initial switch, 3.4 fewer injury days in the second period, and so on. The event-study effect for 9 periods of exposure is missing. This is again because all groups with this treatment path would have experienced that level of exposure in 2020, which we did not observe.



► Example 4: Placebo estimators

We can use placebo estimators, as proposed in [de Chaisemartin and D’Haultfœuille \(2026\)](#), to test the assumptions of parallel trends and no anticipation. In our example, parallel trends is the assumption that, in the absence of safety rules, injuries for teams who implemented safety rules would have followed the same trend (growth, decline, or no change) as injuries for teams who did not implement safety rules. No anticipation is the assumption that safety rules have no effect on injuries in the years prior to their introduction; that is, teams do not change their injury rates (that is, their behavior) in anticipation of safety rules that will soon be introduced. See [de Chaisemartin and D’Haultfœuille \(2026\)](#) for details about the assumptions.

In this context, a placebo estimator is one that takes the same form as the estimator of interest (in this case, a normalized event-study effect estimator) but that measures the effect over a period when no treatment has occurred. Where the event-study effect estimator for exposure ℓ measures the difference between a period t and a later period $t + \ell$, the corresponding placebo estimator measures the difference between the same period t and an earlier period $t - \ell$.

We repeat the `xtswitchdid` command from [example 1](#) and test the assumptions of parallel trends and no anticipation by using `estat ptrends`.

```
. quietly xtswitchdid (injdays lwinpct schedule) (safety), group(team)
. estat ptrends
Placebo estimates                                     Number of obs   = 6,428
                                                         Number of panels =  462
                                                         Number of groups =  462
                                                         (Std. err. adjusted for 462 clusters in team)
```

Placebo	Normalized effect	Robust std. err.	z	P> z	[95% conf. interval]	
-6	1.181488	2.081433	0.57	0.570	-2.898045	5.261022
-5	.4904605	2.581445	0.19	0.849	-4.569079	5.55
-4	-.4558745	.6090937	-0.75	0.454	-1.649676	.7379272
-3	1.016594	.7035227	1.45	0.148	-.3622855	2.395473
-2	-.3029398	1.40519	-0.22	0.829	-3.057062	2.451182
-1	-4.016654	3.185429	-1.26	0.207	-10.25998	2.226673

```
Test of parallel-trends and no-anticipation assumptions
H0: Placebos are zero for all exposures
      chi2(6) =    6.55
Prob > chi2 = 0.3647
```

The placebo estimate labeled -6 is the difference between periods t and $t - 6$. Similarly, the estimate labeled -5 is the difference between periods t and $t - 5$. The remaining placebo effects are interpreted similarly.

Only six placebo estimates are computed, reflecting the fact that most groups have only a small number of pretreatment periods available to observe. Note that placebo estimates are computed using only groups that actually experience switches of the corresponding exposure. For this reason, `estat ptrends` will not display more placebo estimates than there are event-study estimates in the main output.

We can plot the placebo estimates by exposure using the `graph` option. We run the command `quietly` to suppress the tabular output shown above.

```
. quietly estat ptrends, graph
```

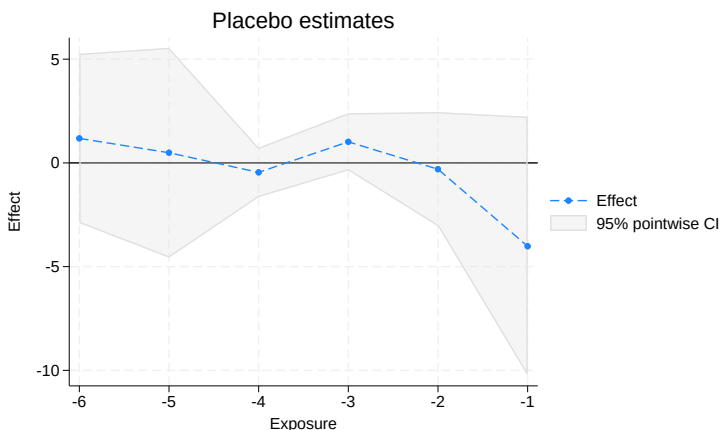


Figure 2. Placebo estimates by exposure

For estimation to be valid, we rely on a parallel-trends assumption, as described in [de Chaisemartin and D'Haultfœuille \(2026\)](#). All confidence intervals include zero. This plot gives evidence that, prior to treatment switches, there was no systematic difference between teams that experienced switches and those that did not. In other words, we have evidence that treated and controls were following the same trend prior to the intervention.

We can plot the main effect estimates on the same plot as our placebo estimates by adding the `alleffects` option:

```
. quietly estat ptrends, graph alleffects
```

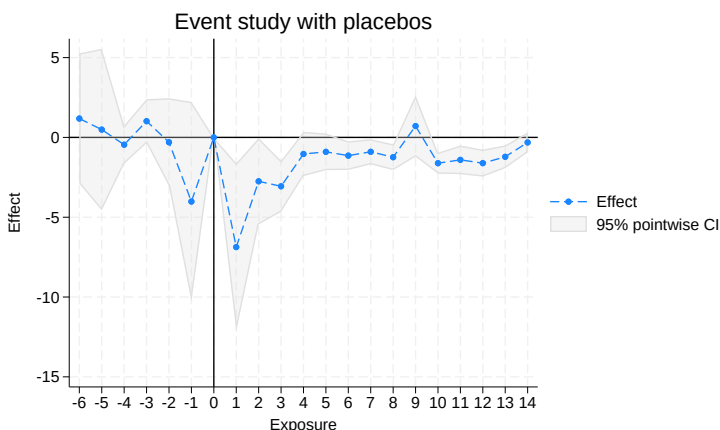


Figure 1. Event-study effect and placebo estimates by exposure

➤ Example 5: Working with coarser treatment groups

While we have team-level data, treatment in our dataset takes place at the league level. We have a relatively small number of leagues (30), but we may want to cluster our standard errors by league:

```
. xtswitchdid (injdays lwinpct schedule) (safety), group(team)
> vce(cluster league)

Treatment and time information
Panel variable:  team
Time variable:  year
Time interval:  2010 thru 2025
Treatment level: team
Control:        safety maintaining baseline
Treatment:      safety changing from baseline
```

Initial treatment	Groups	Obs	Switch out	Switch in	Absorbing
0	272	3,749		1 2 3	No
1	140	1,956		2 3	No
2	50	723		3	Yes

DID with switching treatments

Number of obs = 6,428
Number of panels = 462
Number of groups = 462
(Std. err. adjusted for 30 clusters in league)

	Normalized effect	Robust std. err.	z	P> z	[95% conf. interval]	
Exposure						
1	-6.871766	2.731872	-2.52	0.012	-12.22614	-1.517395
2	-2.75066	.9620966	-2.86	0.004	-4.636334	-.864985
3	-3.061524	1.264431	-2.42	0.015	-5.539762	-.5832855
4	-1.038777	.6633062	-1.57	0.117	-2.338833	.2612794
5	-.9033563	.7267021	-1.24	0.214	-2.327666	.5209537
6	-1.140814	.3921218	-2.91	0.004	-1.909358	-.3722692
7	-.9005077	.213569	-4.22	0.000	-1.319095	-.4819201
8	-1.239846	.2984216	-4.15	0.000	-1.824742	-.6549505
9	.7121286	.8880673	0.80	0.423	-1.028451	2.452708
10	-1.615785	.6452509	-2.50	0.012	-2.880453	-.3511161
11	-1.405509	1.287507	-1.09	0.275	-3.928976	1.117959
12	-1.613272	1.357707	-1.19	0.235	-4.27433	1.047785
13	-1.213585	.9524794	-1.27	0.203	-3.08041	.6532404
14	-.3144064	.3302621	-0.95	0.341	-.9617082	.3328954

Note: Effects computed using covariates.

Compared with the results from clustering at the team level in [example 1](#), some of our event-study effects are now estimated more precisely, while others show larger standard errors.

Our panel variable is team, and until now, we have also taken team as our group variable. However, xtswitchdid also allows for treatment groups that are coarser than panels, provided treatment is fixed within each group. In such cases, xtswitchdid first transforms outcomes and covariates to means within each group before estimation and weights each group by the number of panels within it. We specify group(league) to indicate that treatment was applied at the league level.

```
. xtswitchdid (injdays lwinpct schedule) (safety), group(league)

Treatment and time information

Panel variable:  team
Time variable:   year
Time interval:   2010 thru 2025

Treatment level: league
Control:         safety maintaining baseline
Treatment:       safety changing from baseline
```

Initial treatment	Groups	Obs	Switch out	Switch in	Absorbing
0	17	236		1 2 3	No
1	9	127		2 3	No
2	4	57		3	Yes

DID with switching treatments

Number of obs = 6,428
Number of panels = 462
Number of groups = 30
(Std. err. adjusted for 30 clusters in league)

	Normalized effect	Robust std. err.	z	P> z	[95% conf. interval]	
Exposure						
1	-5.272442	2.456315	-2.15	0.032	-10.08673	-.4581535
2	-2.330813	.8055745	-2.89	0.004	-3.90971	-.7519158
3	-2.911389	1.066263	-2.73	0.006	-5.001226	-.8215528
4	-1.00979	.6159314	-1.64	0.101	-2.216994	.1974131
5	-.7906042	.6420756	-1.23	0.218	-2.049049	.4678408
6	-.9883763	.4369299	-2.26	0.024	-1.844743	-.1320093
7	-.7840238	.3195645	-2.45	0.014	-1.410359	-.157689
8	-.8602774	.4150936	-2.07	0.038	-1.673846	-.046709
9	.775594	.848628	0.91	0.361	-.8876864	2.438874
10	-1.339494	.4164269	-3.22	0.001	-2.155676	-.5233121
11	-1.249448	1.324925	-0.94	0.346	-3.846253	1.347358
12	-1.611414	1.386344	-1.16	0.245	-4.328599	1.105771
13	-1.067826	.9374129	-1.14	0.255	-2.905122	.7694693
14	-.2887542	.383987	-0.75	0.452	-1.041355	.4638464

Note: Effects computed using covariates.

Most of the resulting estimated effects are smaller in absolute value than those estimated with group(team). In general, effects are not equivalent across different specifications of the treatment group variable. Specifying a treatment group coarser than the panel variable may be appropriate when the parallel-trends assumption applied at the panel level is thought to be too strong.



► Example 6: Supergroups

By default, switchers are compared with control groups that share an initial treatment value when estimating event-study effects by DID. For example, switchers who start at treatment value 0 are compared only with control groups that also started at treatment value 0 (and either never switched or switched later than the switcher in question).

We may wish to further restrict the set of control groups. In our hockey dataset, we have a variable, `region`, that can be used for this purpose. By using the `supergroup()` option, we can require that switchers be compared with only control groups that are in the same region as the switcher, in addition to sharing the initial treatment value.

```
. xtswitchdid (injdays lwinpct schedule) (safety), group(team)
> supergroup(region)
note: 4 initial treatment and supergroup combinations dropped because of no
      switchers.
note: 3 initial treatment and supergroup combinations dropped because of no
      controls.
note: 1 initial treatment and supergroup combination dropped because of
      insufficient data.
note: effect of exposure 9 cannot be estimated because of missing data.
Treatment and time information
Panel variable:  team
Time variable:   year
Time interval:   2010 thru 2025
Supergroup:      region
Treatment level: team
Control:          safety maintaining baseline
Treatment:        safety changing from baseline
```

Initial treatment	Groups	Obs	Switch out	Switch in	Absorbing
0	204	2,198		1 2	No

DID with switching treatments

Number of obs

= 2,198

Number of panels

= 204

Number of groups

= 204

Number of supergroups

= 3

(Std. err. adjusted for 204 clusters in team)

	Normalized effect	Robust std. err.	z	P> z	[95% conf. interval]	
Exposure						
1	-6.155398	2.743241	-2.24	0.025	-11.53205	-.7787444
2	-2.284206	1.505154	-1.52	0.129	-5.234254	.6658426
3	-2.285472	1.047505	-2.18	0.029	-4.338544	-.2323989
4	-.8643861	.8303966	-1.04	0.298	-2.491934	.7631612
5	-1.110083	.6256107	-1.77	0.076	-2.336258	.1160909
6	-1.896084	.5303533	-3.58	0.000	-2.935557	-.8566108
7	-1.172279	.5187632	-2.26	0.024	-2.189036	-.1555216
8	-1.248924	.4757709	-2.63	0.009	-2.181418	-.3164301
9	0 (omitted)					
10	-1.792667	.4404417	-4.07	0.000	-2.655917	-.9294172

Note: Effects computed using covariates.

Because we are segmenting the data more finely, some combinations have been dropped, because there were no switchers, no controls, or both in the given subset of the data (see the notes at the top of the output). Furthermore, we can now estimate event-study effects only up to 10 years of exposure.

The `supergroup()` option is useful because it allows for weakening the standard parallel-trends assumption to hold only within specified supergroups, as discussed in [de Chaisemartin et al. \(2025\)](#). Here the event-study effects we find using region as a supergroup are slightly smaller in absolute value than in previous examples, which may suggest that the default assumption was too restrictive.

To see detailed information about supergroups in the first table, we use the `showsupergroups` option.

```
. xtswitchdid (injdays lwinpct schedule) (safety), group(team)
> supergroup(region) showsupergroups
note: 4 initial treatment and supergroup combinations dropped because of no
      switchers.
note: 3 initial treatment and supergroup combinations dropped because of no
      controls.
note: 1 initial treatment and supergroup combination dropped because of
      insufficient data.
note: effect of exposure 9 cannot be estimated because of missing data.
Treatment and time information
Panel variable:  team
Time variable:   year
Time interval:   2010 thru 2025
Supergroup:      region
Treatment level: team
Control:          safety maintaining baseline
Treatment:        safety changing from baseline
```

Initial treatment / supergroup	Groups	Obs	Switch out	Switch in	Absorbing
0 / 2	59	643		1 2	No
0 / 3	123	1,287		2	No
0 / 4	22	268		2	Yes

DID with switching treatments

Number of obs = 2,198
Number of panels = 204
Number of groups = 204
Number of supergroups = 3

(Std. err. adjusted for 204 clusters in team)

	Normalized effect	Robust std. err.	z	P> z	[95% conf. interval]	
Exposure						
1	-6.155398	2.743241	-2.24	0.025	-11.53205	-.7787444
2	-2.284206	1.505154	-1.52	0.129	-5.234254	.6658426
3	-2.285472	1.047505	-2.18	0.029	-4.338544	-.2323989
4	-.8643861	.8303966	-1.04	0.298	-2.491934	.7631612
5	-1.110083	.6256107	-1.77	0.076	-2.336258	.1160909
6	-1.896084	.5303533	-3.58	0.000	-2.935557	-.8566108
7	-1.172279	.5187632	-2.26	0.024	-2.189036	-.1555216
8	-1.248924	.4757709	-2.63	0.009	-2.181418	-.3164301
9	0	(omitted)				
10	-1.792667	.4404417	-4.07	0.000	-2.655917	-.9294172

Note: Effects computed using covariates.

Notice that the header table now gives information on initial treatment and supergroup combinations rather than only on initial treatment values.



► Example 7: Common set of switchers

By default, `xtswitchdid` uses all switchers with available differences at each exposure to compute event-study effects. For example, if a given group has an observed treatment path of 0 0 1 and thus is observed for only one year starting from its first treatment switch, that team will be used to estimate the effect of exposure 1, even though it cannot be used for the effects of exposures 2 or higher.

If the `commonswitchers` option is specified, however, `xtswitchdid` will instead require that a common set of switchers be used for all exposures. If there are four exposures to be computed, `xtswitchdid` with `commonswitchers` will include only those switchers for which there are at least four periods since exposure to a switch. The group with the path 0 0 1 will be excluded from all effect estimation, as will any group observed for fewer than 4 periods starting with the first treatment switch. This may improve interpretability across exposures, but it will also tend to give less precise results as fewer groups are used to compute event-study effects, especially at smaller exposures for which there would otherwise be many groups available.

Note that while the `commonswitchers` option restricts the groups that are used as switchers for each exposure, it does not restrict which groups can be used as controls and in particular does not enforce a common set of control groups.

Here we estimate event-study effects of safety on injdays using the commonswitchers option. There are effects for 14 exposures, which means that only switchers observed for 14 periods after a switch are included in estimation.

```
. xtswitchdid (injdays lwinpct schedule) (safety), group(team)
> commonswitchers
note: 1 initial treatment value dropped because of no switchers in common set.
note: effect of exposure 9 cannot be estimated because of missing data.

Treatment and time information
Panel variable:  team
Time variable:   year
Time interval:   2010 thru 2025

Treatment level: team
Control:         safety maintaining baseline
Treatment:       safety changing from baseline
```

Initial treatment	Groups	Obs	Switch out	Switch in	Absorbing
0	272	3,749		1 2 3	No
1	140	1,956		2 3	No

DID with switching treatments

Number of obs = 5,705
Number of panels = 412
Number of groups = 412
(Std. err. adjusted for 412 clusters in team)

	Normalized effect	Robust std. err.	z	P> z	[95% conf. interval]	
Exposure						
1	-6.379932	3.026874	-2.11	0.035	-12.3125	-.4473692
2	-2.725799	1.54051	-1.77	0.077	-5.745143	.2935439
3	-3.271019	.9280748	-3.52	0.000	-5.090012	-1.452026
4	-1.152499	.775865	-1.49	0.137	-2.673166	.3681689
5	-.9635046	.5899073	-1.63	0.102	-2.119702	.1926925
6	-1.187439	.4627569	-2.57	0.010	-2.094426	-.2804524
7	-.9359829	.4124152	-2.27	0.023	-1.744302	-.1276639
8	-1.275036	.4203548	-3.03	0.002	-2.098917	-.4511563
9	0	(omitted)				
10	-1.604533	.3351186	-4.79	0.000	-2.261353	-.9477123
11	-1.386951	.4582902	-3.03	0.002	-2.285183	-.4887183
12	-1.623364	.4316211	-3.76	0.000	-2.469326	-.7774026
13	-1.217649	.3648081	-3.34	0.001	-1.932659	-.502638
14	-.3144064	.3203692	-0.98	0.326	-.9423185	.3135057

Notes: Effects computed using covariates.
All effects estimated using a common set of switchers.

In the notes above the output, we are informed that an initial treatment group has been dropped. Groups for which the initial treatment level is 2 are not observed for 14 periods since switching, as would be required to be included in the common set of switchers. Specifically, the path 2 3 3 3 3 . 3 3 3 3 3, which is observed for 10 periods starting with the treatment switch, is excluded. The absence of groups with initial treatment 2 is reflected in the initial treatment table.

xtswitchdid also has an neffects() option, which specifies a maximum number of event-study effects to compute. Typically, this option is only cosmetic, because each effect is computed separately. But when commonswitchers is specified, specifying neffects() determines for which ex-

posures the common set of switchers must be valid. This is an important consideration because imposing commonswitchers in combination with a large value of neffects() can amount to discarding most of the switchers in the dataset. In this example, we may want to impose a maximum of five effects:

```
. xtswitchdid (injdays lwinpct schedule) (safety), group(team)
> commonswitchers neffects(5)
```

Treatment and time information
Panel variable: team
Time variable: year
Time interval: 2010 thru 2025
Treatment level: team
Control: safety maintaining baseline
Treatment: safety changing from baseline

Initial treatment	Groups	Obs	Switch out	Switch in	Absorbing
0	272	3,749		1 2 3	No
1	140	1,956		2 3	No
2	50	723		3	Yes

DID with switching treatments

Number of obs = 6,428
Number of panels = 462
Number of groups = 462
(Std. err. adjusted for 462 clusters in team)

	Normalized effect	Robust std. err.	z	P> z	[95% conf. interval]	
Exposure						
1	-6.291381	3.01167	-2.09	0.037	-12.19414	-.3886173
2	-2.375	1.540421	-1.54	0.123	-5.394169	.6441702
3	-2.954174	.9135166	-3.23	0.001	-4.744633	-1.163714
4	-1.064917	.787923	-1.35	0.177	-2.609217	.479384
5	-.9033563	.5854862	-1.54	0.123	-2.050888	.2441756

Notes: Effects computed using covariates.
All effects estimated using a common set of switchers.

Note that initial treatment level 2 is no longer dropped and results are different from the case where neffects() was not specified.



Stored results

xtswitchdid stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_clust)</code>	number of clusters
<code>e(N_g)</code>	number of groups
<code>e(N_exposure)</code>	number of periods of exposure
<code>e(N_panels)</code>	number of panels
<code>e(N_supergroups)</code>	number of supergroups
<code>e(neffects)</code>	number of effects
<code>e(tmin)</code>	first time period
<code>e(tmax)</code>	last time period

Macros

<code>e(cmd)</code>	xtswitchdid
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(clustvar)</code>	name of cluster variable
<code>e(controlgroup)</code>	control group
<code>e(switchgroup)</code>	switcher group
<code>e(path)</code>	name of path variable or <i>numlist</i> as specified
<code>e(path_num)</code>	treatment path
<code>e(path_switch)</code>	path switcher group
<code>e(path_absorb)</code>	path absorbing or nonabsorbing
<code>e(supergroupvar)</code>	name of supergroup variable
<code>e(showsupergroups)</code>	showsupergroups, if specified
<code>e(raw)</code>	raw, if specified
<code>e(commonswitchers)</code>	commonswitchers, if specified
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(marginsnotok)</code>	predictions disallowed by margins
<code>e(listwise)</code>	listwise or nolistwise
<code>e(ivar)</code>	name of panel variable
<code>e(timevar)</code>	name of time variable
<code>e(groupvar)</code>	name of group variable
<code>e(treatment)</code>	name of treatment variable
<code>e(estat_cmd)</code>	program used to implement estat
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(properties)</code>	b V

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals
-----------------------	--

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

Methods and formulas are presented under the following headings:

[Overview](#)
[Standard errors](#)
[Covariates](#)
[Missingness](#)

Overview

We observe data $\{y_{gt}, d_{gt}\}$ for groups $g = 1, \dots, G$, at periods $t = 1, \dots, T$, where y_{gt} is the observed outcome and d_{gt} is the treatment variable. We let n_{gt} be the number of panels within each group. If $n_{gt} > 1$ so that the dataset is not aggregated to the group level, outcomes are transformed to group-level means. Treatments are required to be constant within group-time cells and are also collapsed to the group level. There may also be covariates $\{\mathbf{x}_{gt}\}$, but, for now, we will deal with the case in which there are no covariates.

Denote $y_{gt}(\mathbf{d})$ as the potential outcome for group g at time t if it experienced the treatment path given by the T -length vector $\mathbf{d}$. Let $\mathbf{d}_g = (d_{g1}, \dots, d_{gT})$, and let $\mathbf{D} = (\mathbf{d}_1, \dots, \mathbf{d}_G)$ be the vector of observed treatment paths across all groups.

We do not require initial treatment values d_{g1} to be zero or any other particular or common value. Groups for which treatment changes at least once are called switchers. If the first treatment change moves the treatment to a value higher than the initial treatment value, we call the group a switcher-in, and we call it a switcher-out if the first treatment change moves it to a lower value. s_g takes the value 1 if group g is a switcher-in, -1 if it is a switcher-out, and undefined otherwise. By default, all switchers are included when computing event-study effects. When the `switchgroup(in)` option is specified, only switchers with $s_g = 1$ are included, and only switchers with $s_g = -1$ are included when `switchgroup(out)` is specified.

In a group for which treatment both rises above and falls below the group's initial treatment value at some point in the observed treatment path for that group, s_g is undefined. Such groups are excluded from the estimation sample.

We let t_g denote the last period for which group g 's treatment has not yet changed from d_{g1} , its initial treatment value. We let $T_g \geq t_g$ denote the last time period for which there exists some group g' for which $d_{g'1} = d_{g1}$ and $t_g < t_{g'} = T_{g'}$. In other words, there exists a group that can serve as a control for group g if and only if $t_g < T_g$.

We let $\ell = 1, \dots, L$ index the periods of exposure to treatment, relative to the last untreated period $t_g - 1$, where L is equal to the maximum value of $T_g - t_g$ across groups. n_ℓ is the number of groups for which $t_g + \ell \leq T_g$, meaning that control groups are available to estimate the ℓ th event-study effect.

Our main estimand of interest is the following normalized event-study effect,

$$\delta_\ell^n = \frac{1}{\bar{\delta}_\ell^d} \sum_{g: t_g + \ell \leq T_g} s_g n_{g, t_g + \ell} E\{y_{g, t_g + \ell} - y_{g, t_g + \ell}(d_{g1}, \dots, d_{g1}) | \mathbf{D}\}$$

where $\delta_{g, \ell}^d = \sum_{k=0}^{\ell-1} (d_{g, t_g + k} - d_{g1})$ is the total “dose” of treatment received by a group g from t_g until $t_g + \ell$ and $\bar{\delta}_\ell^d = \sum_{g: t_g + \ell \leq T_g} |\delta_{g, \ell}^d|$ is the total of the absolute values of such doses over each group for which the ℓ th event-study effect can be estimated.

This event-study effect aggregates effects across groups conditional on the observed treatment data. For each group, we are concerned with the difference between the actual treatment path the group experienced up to period $t_g - 1 + \ell$ and a counterfactual path where the group remained at their initial treatment level.

We cannot observe the counterfactual object $y_{g,t_g+\ell}(d_{g1}, \dots, d_{g1})$ for switchers and thus estimate these effects by the principle of difference in differences, where the differences in outcomes for switchers before and after a treatment change and the differences in outcomes that occurred at the same time for comparable never-switchers or not-yet-switchers are themselves differenced to generate a plausible estimate of the difference of interest. The estimator is given by

$$\hat{\delta}_\ell^n = \frac{1}{\delta_\ell^d} \sum_{g:t_g+\ell \leq T_g} s_g \left\{ n_{g,t_g+\ell}(y_{g,t_g+\ell} - y_{g,t_g}) - \frac{1}{n_{t_g+\ell}^g} \sum_{\substack{g': d_{g'1}=d_{g1} \\ t_{g'} > t_g+\ell}} n_{g',t_g+\ell}(y_{g',t_g+\ell} - y_{g',t_g}) \right\}$$

where for any period t , $n_t^g = \sum_{g'} \mathbf{1}(d_{g'1} = d_{g1}, t_{g'} + 1 > t)$.

When the `controlgroup(never)` option is specified, the control groups for group g are defined the same way, except that groups whose treatments ever change from their initial treatments are excluded as controls.

The raw event-study effect δ_ℓ and its estimator $\hat{\delta}_\ell$ are identical to their normalized equivalents with δ_ℓ^d replaced by n_ℓ .

Standard errors

By default, the analytic standard errors given in [de Chaisemartin and D'Haultfœuille \(2026\)](#) and its companion technical documentation ([de Chaisemartin et al. 2025](#)) are computed. Note that these standard errors can be conservative, particularly when there are few groups associated with each initial treatment level.

For the purpose of computing standard errors, writing the event-study estimator as an aggregation over group-specific quantities, without secondary sums over all other groups g' , is convenient. We can write $\hat{\delta}_\ell^n = (1/\delta_\ell^d) \sum_{g=1}^G \mathbf{1}(\ell \leq T_g - 1) \hat{\Delta}_{g,\ell}$, where

$$\hat{\Delta}_{g,\ell} = \frac{1}{n_\ell} \sum_{t=\ell+1}^{T_g} \left\{ \frac{n_{gt\ell}^{\text{in}} - n_{gt\ell}^{\text{out}}}{n_t^g} \mathbf{1}(t_g + 1 > t) + s_g \mathbf{1}(t_g + \ell = t) \right\} n_{gt}(y_{gt} - y_{gt-\ell})$$

Here

$$n_{gt\ell}^{\text{in}} = \sum_{\substack{s_{g'1}=1 \\ t_{g'}+\ell=t \\ g': d_{g'1}=d_{g1} \\ n_t^{g'}>0}} n_{g't}$$

and $n_{gt\ell}^{\text{out}}$ is defined equivalently with the condition $s_{g'} = -1$. An estimator of the variance of the estimator is constructed similarly using the sum of squared deviations from estimated means. We first define

$$\widehat{\Delta}_{g,l}^{V,\text{in}} = \frac{1}{n_\ell} \sum_{t=\ell+1}^{T_g} \left\{ \mathbf{1}(t_g + \ell = t) - \frac{n_{gt\ell}^{\text{in}}}{n_t^g} \mathbf{1}(t_g + 1 > t, S_g = 1) \right\} n_{gt} r_{gt\ell}^{\text{in}} (y_{gt} - y_{gt-\ell} - \widehat{m}_{gt\ell}^{\text{in}})$$

$$\widehat{\Delta}_{g,l}^{V,\text{out}} = \frac{1}{n_\ell} \sum_{t=\ell+1}^{T_g} \left\{ -\mathbf{1}(t_g + \ell = t) + \frac{n_{gt\ell}^{\text{out}}}{n_t^g} \mathbf{1}(t_g + 1 > t, S_g = -1) \right\} n_{gt} r_{gt\ell}^{\text{out}} (y_{gt} - y_{gt-\ell} - \widehat{m}_{gt\ell}^{\text{out}})$$

where $\widehat{m}_{gt\ell}^{\text{in}}$ and $\widehat{m}_{gt\ell}^{\text{out}}$ are sample means of ℓ -length differences among a particular cohort and $r_{gt\ell}^{\text{in}}$ and $r_{gt\ell}^{\text{out}}$ are adjustment factors. Here we define $\widehat{m}_{gt\ell}^{\text{in}}$ and $r_{gt\ell}^{\text{in}}$, but $\widehat{m}_{gt\ell}^{\text{out}}$ and $r_{gt\ell}^{\text{out}}$ are similarly defined.

For periods t , where group g has not yet experienced a switch ($t \leq t_g$), $\widehat{m}_{gt\ell}^{\text{in}}$ is computed in terms of not-yet-switched group cohorts defined by initial treatment d_{g1} ,

$$\widehat{m}_{gt\ell}^{\text{in}} = \frac{\sum_{g': d_{g'1}=d_{g1}} n_{g't} (y_{g't} - y_{g',t-\ell})}{\sum_{g': d_{g'1}=d_{g1}} n_{g't}}$$

provided there is at least one other group not yet switched sharing that value of the initial treatment d_{g1} . Otherwise, $\widehat{m}_{gt\ell}$ is computed in terms of group cohorts defined by initial treatment d_{g1} , including both not-yet-switched groups and switching-in groups for which $t = t_g + \ell$.

For periods t in which group g experienced its first switch from its baseline treatment value in period $t - \ell + 1$ ($t = t_g + \ell$), $\widehat{m}_{gt\ell}^{\text{in}}$ is computed in terms of cohorts defined by initial treatment d_{g1} , latest period where no change had occurred t_g , and treatment value at first treatment change d_{g,t_g+1} , which we can call path cohorts. We have

$$\widehat{m}_{gt\ell}^{\text{in}} = \frac{\sum_{\substack{g': d_{g'1}=d_{g1} \\ t'_g=t_g \\ d_{g',t'_g+1}=d_{g,t_g+1}}} n_{g't} (y_{g't} - y_{g',t-\ell})}{\sum_{\substack{g': d_{g'1}=d_{g1} \\ t'_g=t_g \\ d_{g',t'_g+1}=d_{g,t_g+1}}} n_{g't}}$$

provided there is at least one other group in the path cohort. Otherwise, $\widehat{m}_{gt\ell}^{\text{in}}$ is computed in terms of a cohort made up of any nonswitchers or switchers-in sharing the value of the initial treatment D_{g1} .

$r_{gt\ell}^{\text{in}}$ takes the value $\sqrt{n_{\text{coh}}/(n_{\text{coh}} - 1)}$, where n_{coh} is the number of groups in the cohort used to compute $\widehat{m}_{gt\ell}^{\text{in}}$.

The variance estimator for $\widehat{\delta}_\ell^n$ can then be written as

$$\widehat{\sigma}_{\ell,n}^2 = \frac{n_\ell^2}{(\delta_\ell^d)^2} \sum_{g=1}^G \mathbf{1}(\ell \leq T_g - 1) (\widehat{\Delta}_{g,l}^V)^2$$

where $\widehat{\Delta}_{g,l}^V = \widehat{\Delta}_{g,l}^{V,\text{in}} + \widehat{\Delta}_{g,l}^{V,\text{out}}$. The estimated covariance between $\widehat{\delta}_\ell^n$ and $\widehat{\delta}_{\ell'}^n$ is

$$\widehat{\sigma}_{\ell,\ell'}^n = \frac{n_\ell n_{\ell'}}{\delta_\ell^d \delta_{\ell'}^d} \sum_{g=1}^G \mathbf{1}(\ell \leq T_g - 1) \mathbf{1}(\ell' \leq T_g - 1) \widehat{\Delta}_{g,l}^V \widehat{\Delta}_{g,l'}^V$$

For both estimators, the raw equivalent is obtained by replacing δ_ℓ^d by n_ℓ for each exposure time ℓ .

If clustered variance–covariance estimators are requested, the size of the cohorts used to construct $\widehat{m}_{gt\ell}^{\text{in}}$, $\widehat{m}_{gt\ell}^{\text{out}}$, $r_{gt\ell}^{\text{in}}$, and $r_{gt\ell}^{\text{out}}$ is determined by the number of clusters represented, rather than the number of groups in each cohort. The clustered variance estimator can then be written

$$\widehat{\sigma}_{\ell,n}^2 = \frac{n_\ell^2}{(\delta_\ell^d)^2} \sum_{c=1}^C \left\{ \sum_{g \in g_c} \mathbf{1}(\ell \leq T_g - 1) \widehat{\Delta}_{c,\ell}^V \right\}^2$$

where $c = 1, \dots, C$ indexes clusters and g_c is the set of groups in cluster c .

Covariates

Covariates $\mathbf{x}_{gt}$ may be included in the model. As with the other variables, if covariates are not aggregated up to the group level, means are taken so that there is one observation for each (g, t) cell.

$\widehat{\delta}_{\ell,n}^{X,n}$ is unmodified relative to $\widehat{\delta}_\ell^n$, except that differences $y_{g,t_g+\ell} - y_{g,t_g}$ are replaced by the differences $y_{g,t_g+\ell} - y_{g,t_g} - (\mathbf{x}_{g,t_g+\ell} - \mathbf{x}_{g,t_g})' \widehat{\pi}_{d_{g1}}$ for each group g .

$\widehat{\pi}_{d_{g1}}$ is a vector of coefficients from regression of the first differences of y_{gt} on first differences of $\mathbf{x}_{gt}$, entering both linearly and interacted with indicator variables corresponding to each distinct value of d_{g1} , after partialing out the full set of time fixed effects (also included both linearly and as interactions with indicators for distinct values of d_{g1}), fit on the subset of group-time cells, where $t_g \geq t$, and with importance weights given by n_{gt} .

To write down a variance estimator, let $\tilde{\mathbf{x}}_{gt}$ be the vector of covariates projected off the time fixed effects, or in other words, the residuals from the partialing-out regression. Also let $S_{XX,d}$ be the mean sum-of-squares matrix of the projected covariates $\tilde{\mathbf{x}}_{gt}$ for a given subset of group-time cells (g, t) , where $d_{g1} = d$ and $t_g \geq t$, and let $\widehat{\Delta}_{y_{gt}}$ be the fitted values (linear predictions) for the first differences of y_{gt} . We can then write

$$\widehat{\sigma}_{\ell,n,X}^2 = \frac{n_\ell^2}{(\delta_\ell^d)^2} \sum_{g=1}^G (\widehat{\Delta}_{g,\ell}^{V,X} + e_{g,\ell})^2$$

Here $\widehat{\Delta}_{g,\ell}^{V,X}$ is equivalent to $\widehat{\Delta}_{g,\ell}^V$, except that differences $y_{g,t_g+\ell} - y_{g,t_g}$ are replaced by the differences $y_{g,t_g+\ell} - y_{g,t_g} - (\mathbf{x}_{g,t_g+\ell} - \mathbf{x}_{g,t_g})' \widehat{\pi}_{d_{g1}}$ for each group g , including in the definitions of $\widehat{m}_{gt\ell}^{\text{in}}$ and $\widehat{m}_{gt\ell}^{\text{out}}$. The term $e_{g\ell}$ corrects for the fact that the differences were constructed using the estimated quantities $\widehat{\pi}_{d_{g1}}$ and is given as

$$e_{g,\ell} = \sum_{d \in \mathcal{D}_1} M_{d\ell} \left\{ S_{XX,d}^{-1} \frac{1}{n_d} \sum_{t=2}^{t_g} n_{gt} \tilde{\mathbf{x}}_{gt} r_d^X (\Delta y_{gt} - \widehat{\Delta}_{y_{gt}}) \mathbf{1}(d_{g1} = d, t_g \geq 2) - \widehat{\pi}_d \right\}$$

where

$$M_{d\ell} = \sum_{\substack{d_{g1}=d \\ g: t_g+\ell \leq T_g}} n_{g,t_g+\ell} \left\{ (\mathbf{x}_{g,t_g+\ell} - \mathbf{x}_{g,t_g})' - \sum_{\substack{d_{g'1}=d_{g1} \\ g': t_{g'}+1 > t_g+\ell}} \frac{n_{g',t_g+\ell}}{n_{t_g+\ell}} (\mathbf{x}_{g',t_g+\ell} - \mathbf{x}_{g',t_g})' \right\}$$

Further, $r_d^X = \sqrt{n_{\text{coh}}/(n_{\text{coh}} - 1)}$, where n_{coh} is the number of group-time cells for which $d_{g1} = d$.

In some cases, there may be insufficient data to compute $\hat{\pi}_{d_{g1}}$ and adjust for covariates for certain values of the initial treatment d_{g1} , even if there is enough information to compute DID estimators. In these cases, a note is printed and estimation proceeds without covariate adjustment for those values of d_{g1} .

Missingness

By default, `xtswitchdid` handles missing values the way most other estimation commands, including `hddidregress` and `xtddidregress`, handle them. Observations for which one of the variables specified by the user is missing are dropped at the outset, and estimation proceeds using the remaining sample. Time of first treatment switch for a group is determined by the first observed treatment value different from the observed baseline, even if there are intervening gaps in the time series that may have arisen from missing covariates or other values.

The `nolistwise` option still drops observations with missing group, panel, or time variables but otherwise handles missingness on the fly throughout the estimation procedure. Observations with missing outcomes or covariates but observed treatments are used to infer time of treatment switch but are excluded from effect estimation.

As a general principle, `xtswitchdid` with the `nolistwise` option attempts to maximize what can be estimated given the nonmissing data without imposing strong assumptions. If missing values of the treatment variable prevent inferring when a group had its first treatment switch, observations after those missing values are discarded. This corresponds to the “conservative” option for dealing with missing treatments discussed in [de Chaisemartin et al. \(2025\)](#). Any missing treatment values in periods prior to $t_g + \ell$ for a given group g and exposure ℓ are imputed as equal to the value of the treatment at the time of the first treatment switch, as in the same article. This means that these missing treatment values are assumed not to violate the requirement that treatments cannot both rise above and fall below the initial treatment level. It also allows computing $\hat{\delta}_\ell^d$ for the normalized estimators and average total effect estimator. This imputation does not affect inference of an individual event-study effect being nonzero but may affect interpretation; the `raw` option may be more appropriate with the `nolistwise` option.

Note that whether `nolistwise` is specified or not, `xtswitchdid` drops data that cannot be used to compute differences. In particular, groups that are singletons, where only one time period is observed, are excluded from the estimation sample. Initial treatment groups for which there are no switchers or no available controls are also dropped.

Acknowledgments

We thank Clément de Chaisemartin of Sciences Po and Xavier D’Haultfœuille of CREST-ENSAE for their significant contributions to the literature on DID methods and for helpful conversations and advice. Their community-contributed command `did_multipligt_dyn` implements a full-featured version of the method in this entry. We also thank their capable research assistants—with special thanks to David Arboleda, Diego Ciccía, Felix Knau, and Anzony Quispe for their feedback.

References

- de Chaisemartin, C., D. Ciccia, F. Knau, M. Malézieux, D. Sow, D. Arboleda, R. Angotti, X. D'Haultfœuille, B. Li, H. Fabre, and A. Quispe. 2025. Using `did_multiplengt_dyn` to estimate event-study effects in complex designs: Overview, and four examples based on real datasets. Unpublished manuscript. <https://doi.org/10.2139/ssrn.5337463>.
- de Chaisemartin, C., and X. D'Haultfœuille. 2026. Difference-in-differences estimators of intertemporal treatment effects. *Review of Economics and Statistics* 108: 863–880. https://doi.org/10.1162/rest_a_01414.

Also see

- [CAUSAL] **xtswitchdid postestimation** — Postestimation tools for `xtswitchdid`⁺
- [CAUSAL] **DID intro** — Introduction to difference-in-differences estimation
- [CAUSAL] **didregress** — Difference-in-differences estimation⁺
- [CAUSAL] **hdidregress** — Heterogeneous difference in differences⁺
- [CAUSAL] **xthdidregress** — Heterogeneous difference in differences for panel data⁺
- [U] **20 Estimation and postestimation commands**

⁺Postestimation features after xtswitchdid are part of [StataNow](#).

Postestimation commands
Methods and formulas

estat
References

Remarks and examples
Also see

Stored results

Postestimation commands

The following postestimation commands are of special interest after xtswitchdid:

Command	Description
<code>estat ptrends</code>	parallel-trends and no-anticipation test using placebos
<code>estat eventplot</code>	plot the event-study effects at each exposure
<code>estat total</code>	estimate average total effect
<code>estat paths</code>	tabulate treatment paths observed in the estimation sample

The following postestimation commands are also available:

Command	Description
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of parameters
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

estat

Description for estat

- `estat ptrends` tests the parallel-trends and no-anticipation assumptions by testing whether all available placebos are equal to zero and optionally displays the placebo estimates in a graph.
- `estat eventplot` plots the event-study effects for each exposure.
- `estat total` aggregates the event-study effects across exposures to give an average total effect.
- `estat paths` tabulates the treatment paths observed in the estimation sample and generates a variable encoding the relevant treatment path for each group. It optionally produces indicator variables for each path, which can be used by the main command to produce path-specific estimates.

Menu for estat

Statistics > Postestimation

Syntax for estat

Test parallel-trends and no-anticipation assumptions using placebos

```
estat ptrends [ , ptrends_options ]
```

Plot event-study effects

```
estat eventplot [ , eventplot_options ]
```

Compute average total effect

```
estat total [ , level(#)]
```

Tabulate treatment paths

```
estat paths [ , paths_options ]
```

ptrends_options	Description
level(#)	set confidence level; default is level(95)
graph	display a plot of placebo estimates
graph(graph_opts)	affect rendition of the aggregation plot
alleffects	plot main effects with placebo estimates

eventplot_options	Description
level(#)	set confidence level
graph_opts	affect rendition of the event-study effects plot

paths_options	Description
pathlen(#)	set path length
freqsort	request paths be listed in descending frequency order
generate(pathstub)	generate path indicator variables with names pathstub#
pathvar(pvar[, replace])	specify the name for the generated path variable

<i>graph_opts</i>	Description
Main	
<i>noci</i>	do not plot the confidence intervals
Marker options	
<i>marker_options</i>	change the look of markers (color, size, etc.)
Line options	
<i>connect_options</i>	change the look of lines or connecting method
CI plot	
<i>ciopts (area_options)</i>	affect rendition of the confidence interval
Y axis, X axis, Titles, Legend, Overall	
<i>twoway_options</i>	any options other than by() documented in [G-3] <i>twoway_options</i>

Options for estat

Options for estat are presented under the following headings:

- [Options for estat ptrends](#)
- [Options for estat eventplot](#)
- [Option for estat total](#)
- [Options for estat paths](#)

Options for estat ptrends

`level(#)` specifies the confidence level, as a percentage, for CIs. The default is `level(95)` or as set by `set level`; see [U] 20.8 Specifying the width of confidence intervals.

`graph` requests that a graph of the placebo estimates be displayed.

`graph(graph_opts)` affects the rendition of the placebo estimates plot. *graph_opts* may be the following:

- `noci` removes plots of the CIs. The default is to plot the CIs.
- marker_options* affect the rendition of markers drawn at the plotted points, including their shape, size, color, and outline; see [G-3] *marker_options*.
- connect_options* specify how points on a graph are to be connected; see [G-3] *connect_options*.
- ciopts(area_options)* affects the rendition of the CIs; see [G-3] *area_options*.
- twoway_options* are any of the options documented in [G-3] *twoway_options*, excluding `by()`. These include options for titling the graph (see [G-3] *title_options*) and for saving the graph to disk (see [G-3] *saving_option*).

`alleffects` specifies that the graph, if requested, plot the main event-study effects along with the placebo estimates.

Options for estat eventplot

`level(#)` specifies the confidence level, as a percentage, for CIs. The default is `level(95)` or as set by `set level`; see [U] 20.8 Specifying the width of confidence intervals.

Main

`noci` removes plots of the CIs. The default is to plot the CIs.

Marker options

`marker_options` affect the rendition of markers drawn at the plotted points, including their shape, size, color, and outline; see [G-3] *marker_options*.

Line options

`connect_options` specify how points on a graph are to be connected; see [G-3] *connect_options*.

CI plot

`ciopts(area_options)` affects the rendition of the CIs; see [G-3] *area_options*.

Y axis, X axis, Titles, Legend, Overall

`twoway_options` are any of the options documented in [G-3] *twoway_options*, excluding `by()`. These include options for titling the graph (see [G-3] *title_options*) and for saving the graph to disk (see [G-3] *saving_option*).

Option for estat total

`level(#)` specifies the confidence level, as a percentage, for CIs. The default is `level(95)` or as set by `set level`; see [U] 20.8 Specifying the width of confidence intervals.

Options for estat paths

`pathlen(#)` specifies the length of the treatment paths to tabulate. By default, `estat paths` uses the number of event-study effects computed in the main command.

`freqsort` requests that the treatment paths be listed in descending frequency order. By default, they are listed in lexicographical order.

`generate(pathstub)` requests that indicator variables be generated for each treatment path. The new variables will be named `pathstub#`.

`pathvar(pvar[ , replace])` specifies `pvar` as the name for the generated treatment path variable, which encodes each treatment path as a different number and assigns value labels for each path. By default, `_did_paths` is used as the name of this path variable. If `_did_paths` already exists in the dataset, it is replaced if `pathvar()` is not specified.

`replace` specifies that `pvar` be overwritten if it exists.

Remarks and examples

For examples of the `estat` commands above, see [CAUSAL] *xtswitchdid*.

Stored results

`estat ptrends` stores the following in `r()`:

Scalars

<code>r(chi2)</code>	χ^2
<code>r(df)</code>	test constraints degrees of freedom
<code>r(p)</code>	two-sided p -value
<code>r(drop)</code>	1 if constraints were dropped, 0 otherwise
<code>r(dropped_i)</code>	index of i th constraint dropped

Matrices

<code>r(b)</code>	coefficient vector
<code>r(V)</code>	variance–covariance matrix of the estimators
<code>r(table)</code>	matrix containing test statistics and critical values

`estat eventplot` stores the following in `r()`:

Matrices

<code>r(table)</code>	matrix containing test statistics and critical values
-----------------------	---

`estat total` stores the following in `r()`:

Scalars

<code>r(b)</code>	coefficient
<code>r(V)</code>	variance of the estimator

Matrices

<code>r(table)</code>	matrix containing test statistics and critical values
-----------------------	---

`estat paths` stores the following in `r()`:

Scalars

<code>r(n_paths)</code>	number of treatment paths
<code>r(pathlen)</code>	path length

Macros

<code>r(pathvar)</code>	name of path variable
<code>r(pathvar)</code>	stub name for path indicator variables

Methods and formulas

Methods and formulas are presented under the following headings:

Test for all placebos being zero
Measure of average total effect

Test for all placebos being zero

`estat ptrends` tests implications of the parallel-trends and no-anticipation assumptions by testing whether all observed placebos are zero, as described in [de Chaisemartin, D’Haultfœuille, and Vazquez-Bare \(2024\)](#).

Placebo estimates for the ℓ th exposure are computed in the same way that event-study effect estimates are computed, but the observed outcome $y_{g,t_g+\ell}$ is replaced by the observed outcome $y_{g,t_g-\ell}$. In other words, we still take the last unswitched period as a baseline, but the difference is taken using the outcome ℓ periods before, rather than ℓ periods after, provided that outcome is available in the data. If the parallel-trends and no-anticipation assumptions hold, we expect these differences to be zero on average.

The normalized placebo estimator without covariates for the ℓ th exposure is written, using notation similar to that in [CAUSAL] **xtswitchdid**, as

$$\hat{\delta}_{\ell}^{n,p} = \frac{1}{\delta_{\ell}^{d,p}} \sum_{g: t_g - \ell \geq 1, t_g + \ell \leq T_g} s_g \left\{ n_{g, t_g + \ell} (y_{g, t_g - \ell} - y_{g, t_g}) - \frac{1}{n_{t_g + \ell}^g} \sum_{g': d_{g', 1} = d_{g, 1} \atop g': t_{g'} > t_g + \ell} n_{g', t_g + \ell} (y_{g', t_g - \ell} - y_{g', t_g}) \right\}$$

where $\delta_{g, \ell}^{d,p} = \sum_{k=0}^{\ell-1} (d_{g, t_g - k} - d_{g, 1})$ is the total “dose” of treatment received by a group g from t_g “until” $t_g - \ell$ and $\delta_{\ell}^{d,p}$ is the total of the absolute values of such doses over each group for which the ℓ th placebo estimate can be computed.

Note that, as written, the ℓ th placebo estimate can be computed only if the ℓ th event-study effect estimate is computed. Only groups for which the ℓ th event-study effect is available are included in the ℓ th placebo estimator. This means that if L event-study effects have been computed, there can be at most L placebo estimates computed.

The raw placebo estimator $\hat{\delta}_{\ell}^p$ is identical to its normalized equivalents with δ_{ℓ}^d replaced by n_{ℓ} . Estimators with covariates are computed identically as for the main effects, with the same modifications as above.

The covariance estimator for the placebos also corresponds to the covariance estimator for the main effects, with future outcomes at a time period $t_g + \ell$ replaced by past outcomes at time period $t_g - \ell$.

The placebos are tested against the null of all estimates being zero using `test`. When the variance–covariance matrix is rank deficient, such as when there are more placebo estimates than clusters, constraints are dropped and a note is printed.

Measure of average total effect

For the sake of brevity, we restrict attention to when there are no switchers-out. See [de Chaisemartin et al. \(2025\)](#) for the computation of $\hat{\delta}$ in the general case, which involves constructing separate quantities for switchers-out and switchers-in and combining use of appropriate weights. `estat total` aggregates over exposures to compute the following effect,

$$\hat{\delta} = \frac{\sum_{g: t_g < T_g} \sum_{\ell=1}^{T_g - t_g} \hat{\delta}_{g, \ell}}{\sum_{g: t_g < T_g} \sum_{\ell=1}^{T_g - t_g} (d_{g, t_g + \ell} - d_{g, 1})}$$

where

$$\hat{\delta}_{g, \ell} = n_{g, t_g + \ell} (y_{g, t_g + \ell} - y_{g, t_g}) - \frac{1}{n_{t_g + \ell}^g} \sum_{g': d_{g', 1} = d_{g, 1} \atop g': t_{g'} > t_g + \ell} n_{g', t_g + \ell} (y_{g', t_g + \ell} - y_{g', t_g})$$

When the treatment is absorbing, this effect can be interpreted as an average effect of the total treatment dose over each period.

In the case with no switchers-out we consider here, $\hat{\delta}$ can be written as a weighted average of the raw estimators $\hat{\delta}_{\ell}$ over exposures. The variance estimator for $\hat{\delta}$ can be computed accordingly. See [de Chaisemartin et al. \(2025\)](#) for the general case.

References

- de Chaisemartin, C., D. Ciccia, F. Knau, M. Malézieux, D. Sow, D. Arboleda, R. Angotti, X. D’Haultfœuille, B. Li, H. Fabre, and A. Quispe. 2025. Using `did_multipligt_dyn` to estimate event-study effects in complex designs: Overview, and four examples based on real datasets. Unpublished manuscript. <https://doi.org/10.2139/ssrn.5337463>.
- de Chaisemartin, C., X. D’Haultfœuille, and G. Vazquez-Bare. 2024. Difference-in-difference estimators with continuous treatments and no stayers. *American Economic Association Papers and Proceedings* 114: 610–613. <https://doi.org/10.1257/pandp.20241049>.

Also see

[CAUSAL] **xtswitchdid** — Difference in differences with switching treatments for panel data⁺

[CAUSAL] **DID intro** — Introduction to difference-in-differences estimation

[U] **20 Estimation and postestimation commands**

Glossary

absorbing treatment. A binary treatment is said to be absorbing when units that receive the treatment remain treated in subsequent periods. When a treatment variable is multivalued, absorbing treatments are those for which units, once having *switched* away from their initial treatment level, do not switch their treatment level again.

ADTE. See *average direct treatment effect*.

ADTET. See *average direct treatment effect with respect to the treated*.

AIPW estimator. See *augmented inverse-probability-weighted estimator*.

AITE. See *average indirect treatment effect*.

AITEC. See *average indirect treatment effect with respect to controls*.

analysis time. Analysis time is like time, except that 0 has a special meaning: $t = 0$ is the time of onset of risk, the time when failure first became possible.

Analysis time is usually not what is recorded in a dataset. A dataset of patients might record calendar time. Calendar time must then be mapped to analysis time.

The letter t is reserved for time in analysis-time units. The term *time* is used for time measured in other units.

The *origin* is the *time* corresponding to $t = 0$, which can vary subject to subject. Thus $t = \text{time} - \text{origin}$.

anticipatory effect. In an *event study*, anticipatory effects (or anticipation effects) are any observed effects of an event on an outcome at time periods before the event actually happens. Anticipatory effects occur when units in the study change their behavior in anticipation of the event.

ATE. See *average treatment effect*.

ATET. See *average treatment effect on the treated*.

augmented inverse-probability-weighted estimator. An augmented inverse-probability-weighted (AIPW) estimator is an inverse-probability-weighted estimator that includes an augmentation term that corrects the estimator when the treatment model is misspecified. When the treatment is correctly specified, the augmentation term vanishes as the sample size becomes large. An AIPW estimator uses both an outcome model and a treatment model and is a doubly robust estimator.

average direct treatment effect. The average direct treatment effect (ADTE) is the average direct effect of the treatment on the outcome when the mediator is held at its value associated with being untreated.

For example, suppose that exercise has a direct effect on well-being as well as an indirect effect via the production of endorphins. The average direct treatment effect measures the average difference in potential outcomes for well-being when everyone in the population exercised but with endorphins set to levels that would be observed if no one exercised compared with when no one in the population exercised.

average direct treatment effect with respect to the treated. The average direct treatment effect with respect to the treated (ADTET) is the average direct treatment effect when the mediator is held at its value associated with being treated.

For example, suppose that exercise has a direct effect on well-being as well as an indirect effect via the production of endorphins. The average direct treatment effect with respect to the treated measures the average difference in the potential outcomes for well-being when everyone in the population exercised compared with when no one exercised but with endorphins set to levels that would be observed if everyone exercised.

average indirect treatment effect. The average indirect treatment effect (AITE) is the average indirect effect through a mediator.

For example, suppose that exercise has a direct effect on well-being as well as an indirect effect via the production of endorphins. The average indirect treatment effect measures the average difference in the potential outcomes for well-being when everyone in the population exercised compared with when everyone exercised but with endorphins set to levels that would be observed if no one exercised.

average indirect treatment effect with respect to controls. The average indirect treatment effect with respect to controls (AITEC) is the average indirect effect of a mediator under the control condition.

For example, suppose that exercise has a direct effect on well-being as well as an indirect effect via the production of endorphins. The average indirect treatment effect with respect to controls measures the average difference in the potential outcomes for well-being when no one in the population exercised but with endorphins set to levels that would be observed if everyone exercised compared with the case where no one in the population exercised.

average treatment effect. The average treatment effect is the average effect of the treatment among all individuals in a population.

average treatment effect on the treated. The average treatment effect on the treated is the average effect of the treatment among those individuals who actually get the treatment.

CATE. See *conditional average treatment effect*.

causal mediation analysis. Causal mediation analysis is a method that disentangles the causal mechanisms that operate between a cause and an effect by decomposing causal effects of a treatment, or exposure, on an outcome variable into direct and indirect effects. The indirect effects are due to the treatment affecting the outcome via intermediary variables, also known as *mediators*. Direct effects are causal effects that are due to mechanisms other than those captured by the mediators.

censored, left-censored, and right-censored. An observation is left-censored when the exact time of failure is not known; it is merely known that the failure occurred before t_l . Suppose that the event of interest is becoming employed. If a subject is already employed when first interviewed, his outcome is left-censored.

An observation is right-censored when the time of failure is not known; it is merely known that the failure occurred after t_r . If a patient survives until the end of a study, the patient's time of death is right-censored.

In common usage, *censored* without a modifier means right-censored.

Also see *truncation, left-truncation, and right-truncation*.

CI assumption. See *conditional-independence assumption*.

complier. In an experimental setting, compliers are units that adhere to the treatment status they were assigned. In an observational-data setting, an instrumental variable may motivate units to self-select into a certain treatment status; compliers are then units that opt for the treatment status they were motivated into.

conditional average treatment effect. The conditional average treatment effect (CATE) is the average of the treatment effects conditional on a set of variables.

conditional mean. The conditional mean expresses the average of one variable as a function of some other variables. More formally, the mean of y conditional on $\mathbf{x}$ is the mean of y for given values of $\mathbf{x}$; in other words, it is $E(y|\mathbf{x})$.

A conditional mean is also known as a regression or as a conditional expectation.

conditional-independence assumption. The conditional-independence assumption requires that the common variables that affect treatment assignment and treatment-specific outcomes be observable. The dependence between treatment assignment and treatment-specific outcomes can be removed by conditioning on these observable variables.

This assumption is also known as a selection-on-observables assumption because its central tenet is the observability of the common variables that generate the dependence.

counterfactual. A counterfactual is an outcome a subject would have obtained had that subject received a different level of treatment. In the binary-treatment case, the counterfactual outcome for a person who received treatment is the outcome that person would have obtained had the person instead not received treatment; similarly, the counterfactual outcome for a person who did not receive treatment is the outcome that person would have obtained had the person received treatment.

Also see *potential outcome*.

doubly robust estimator. A doubly robust estimator only needs one of two auxiliary models to be correctly specified to estimate a parameter of interest.

Doubly robust estimators for treatment effects are consistent when either the outcome model or the treatment model is correctly specified.

EE estimator. See *estimating-equation estimator*.

estimating-equation estimator. An estimating-equation (EE) estimator calculates parameters estimates by solving a system of equations. Each equation in this system is the sample average of a function that has mean zero.

These estimators are also known as M estimators or Z estimators in the statistics literature and as generalized method of moments (GMM) estimators in the econometrics literature.

event study. Event studies evaluate the impact of an event, such as a sudden policy change or an intervention, on an outcome. Estimates of the impact are called event-study effects.

failure event. Survival analysis is really time-to-failure analysis, and the failure event is the event under analysis. The failure event can be death, heart attack, myopia, or finding employment. Many authors—including Stata—write as if the failure event can occur only once per subject, but when we do, we are being sloppy. Survival analysis encompasses repeated failures, and all of Stata’s survival analysis features can be used with repeated-failure data.

GATE. See *group average treatment effect*.

GATES. See *sorted group average treatment effect*.

group average treatment effect. The group average treatment effect (GATE) is the average of the treatment effects conditional on being in one of the prespecified groups. There is a GATE for each group.

hazard, cumulative hazard, and hazard ratio. The hazard or hazard rate at time t , $h(t)$, is the instantaneous rate of failure at time t conditional on survival until time t . Hazard rates can exceed 1. Say that the hazard rate were 3. If an individual faced a constant hazard of 3 over a unit interval and if the failure event could be repeated, the individual would be expected to experience three failures during the time span.

The cumulative hazard, $H(t)$, is the integral of the hazard function $h(t)$, from 0 (the onset of risk) to t . It is the total number of failures that would be expected to occur up until time t , if the failure event could be repeated. The relationship between the cumulative hazard function, $H(t)$, and the survivor function, $S(t)$, is

$$S(t) = \exp\{-H(t)\}$$

$$H(t) = -\ln\{S(t)\}$$

The hazard ratio is the ratio of the hazard function evaluated at two different values of the covariates: $h(t | \mathbf{x})/h(t | \mathbf{x}_0)$. The hazard ratio is often called the relative hazard, especially when $h(t | \mathbf{x}_0)$ is the baseline hazard function.

IATE. See *individualized average treatment effect*.

i.i.d. sampling assumption. See *independent and identically distributed sampling assumption*.

independent and identically distributed sampling assumption. The independent and identically distributed (i.i.d.) sampling assumption specifies that each observation is unrelated to (independent of) all the other observations and that each observation is a draw from the same (identical) distribution.

individualized average treatment effect. The individualized average treatment effect (IATE) is the average of the treatment effects conditional on observation-level characteristics. There is one IATE for each observation in the data.

individual-level treatment effect. An individual-level treatment effect is the difference in an individual's outcome that would occur because this individual is given one treatment instead of another. In other words, an individual-level treatment effect is the difference between two potential outcomes for an individual.

For example, the blood pressure an individual would obtain after taking a pill minus the blood pressure an individual would obtain had that person not taken the pill is the individual-level treatment effect of the pill on blood pressure.

inverse-probability-weighted estimators. Inverse-probability-weighted (IPW) estimators use weighted averages of the observed outcome variable to estimate the potential-outcome means. The weights are the reciprocals of the treatment probabilities estimated by a treatment model.

inverse-probability-weighted regression-adjustment estimators.

Inverse-probability-weighted regression-adjustment (IPWRA) estimators use the reciprocals of the estimated treatment probability as weights to estimate missing-data-corrected regression coefficients that are subsequently used to compute the potential-outcome means.

IPW estimators. See *inverse-probability-weighted estimators*.

IPWRA estimators. See *inverse-probability-weighted regression-adjustment estimators*.

LATE. See *local average treatment effect*.

left-censored. See *censored, left-censored, and right-censored*.

left-truncation. See *truncation, left-truncation, and right-truncation*.

local average treatment effect. The local average treatment effect (LATE) is the average of the treatment effect for the subpopulation of individuals who comply with their assigned treatments.

matching estimator. An estimator that compares differences between the outcomes of similar—that is, matched—individuals. Each individual that receives a treatment is matched to a similar individual that does not get the treatment, and the difference in their outcomes is used to estimate the individual-level treatment effect. Likewise, each individual that does not receive a treatment is matched to a similar individual that does get the treatment, and the difference in their outcomes is used to estimate the individual-level treatment effect.

mediator. A mediator is an intermediary variable on the causal pathway between a treatment, or exposure, and an outcome variable. For example, endorphin production may be a mediator in the relationship between exercise (the treatment) and well-being (the outcome). In this case, exercise affects endorphin production, which, in turn, affects well-being.

mediator-specific natural effect. A mediator-specific natural effect is the direct or indirect effect of a treatment on an outcome in causal mediation models with multiple mediators that are causally ordered and where a focus is put on the causal pathway related to one of the mediators while paths related to the remaining mediators are combined. In a causal mediation model with two mediators that are causally ordered, type-1 mediator-specific effects refer to direct and indirect natural effects related to the mediator that appears first in the causal sequence. Type-2 mediator-specific effects are direct and indirect natural effects related to the mediator that is caused by the first mediator.

multiple-record st data. See [st data](#).

multivalued treatment effect. A multivalued treatment refers to a treatment that has more than two values. For example, a person could have taken a 20 mg dose of a drug, a 40 mg dose of the drug, or not taken the drug at all.

natural direct effect. The natural direct effect (NDE) is the average direct effect of the treatment on the outcome when the mediator is held at its value associated with being untreated.

For example, suppose that exercise has a direct effect on well-being as well as an indirect effect via the production of endorphins. The natural direct effect measures the average difference in the potential outcomes for well-being when everyone in the population exercised but with endorphins set to levels that would be observed if no one exercised compared with when no one in the population exercised. In the causal mediation literature, the natural direct effect is also referred to as the pure natural direct effect; it is synonymous with the [average direct treatment effect](#).

natural indirect effect. The natural indirect effect (NIE) is the average indirect effect through a mediator.

For example, suppose that exercise has a direct effect on well-being as well as an indirect effect via the production of endorphins. The natural indirect effect measures the average difference in the potential outcomes for well-being when everyone in the population exercised compared with when everyone exercised but with endorphins set to levels that would be observed if no one exercised. In the causal mediation literature, the natural indirect effect is also referred to as the total natural indirect effect; it is synonymous with the [average indirect treatment effect](#).

NDE. See [natural direct effect](#).

nearest-neighbor matching. Nearest-neighbor matching uses the distance between observed variables to find similar individuals.

NIE. See [natural indirect effect](#).

nonabsorbing treatment. See [absorbing treatment](#).

observational data. In observational data, treatment assignment is not controlled by those who collected the data; thus some common variables affect treatment assignment and treatment-specific outcomes.

outcome model. An outcome model is a model used to predict the outcome as a function of covariates and parameters.

overlap assumption. The overlap assumption requires that each individual have a positive probability of each possible treatment level.

placebo estimator. A placebo estimator is one that takes the same form as the estimator of interest (such as an event-study effect estimator) but measures the effect over a period where no change in treatment has occurred. A placebo estimator can be used to estimate or test for the presence of [anticipatory effects](#). The name “placebo” in this context stems from the fact that anticipatory effects arise before a treatment has been applied; that is, they arise in the absence of treatment.

PNIE. See [pure natural indirect effect](#).

POMs. See [potential-outcome means](#).

potential outcome. The potential outcome is the outcome an individual would obtain if given a specific treatment.

For example, an individual has one potential blood pressure after taking a pill and another potential blood pressure had that person not taken the pill.

potential-outcome means. The potential-outcome means refers to the means of the potential outcomes for a specific treatment level.

The mean blood pressure if everyone takes a pill and the mean blood pressure if no one takes a pill are two examples.

The average treatment effect is the difference between potential-outcome mean for the treated and the potential-outcome mean for the not treated.

propensity score. The propensity score is the probability that an individual receives a treatment.

propensity-score matching. Propensity-score matching uses the distance between estimated propensity scores to find similar individuals.

pure natural indirect effect. The pure natural indirect effect (PNIE) is the average indirect effect of a mediator under the control condition.

For example, suppose that exercise has a direct effect on well-being as well as an indirect effect via the production of endorphins. The pure natural indirect effect measures the average difference in the potential outcomes for well-being when no one in the population exercised but with endorphins set to levels that would be observed if everyone exercised compared with the case when no one in the population exercised. The pure natural indirect effect is synonymous with the [average indirect treatment effect with respect to controls](#).

regression-adjustment estimators. Regression-adjustment estimators use means of predicted outcomes for each treatment level to estimate each potential-outcome mean.

right-censored. See [censored](#), [left-censored](#), and [right-censored](#).

right-truncation. See [truncation](#), [left-truncation](#), and [right-truncation](#).

selection-on-observables. See [conditional-independence assumption](#).

shape parameter. A shape parameter governs the shape of a probability distribution. One example is the parameter p of the Weibull model.

single-record st data. See *st data*.

smooth treatment-effects estimator. A smooth treatment-effects estimator is a smooth function of the data so that standard methods approximate the distribution of the estimator. The RA, IPW, AIPW, and IPWRA estimators are all smooth treatment-effects estimators while the nearest-neighbor matching estimator and the propensity-score matching estimator are not.

sorted group average treatment effect. The sorted group average treatment effect (GATES) computes the average treatment effect conditional on being in one of the groups when using a prespecified number of groups. The groups are determined by quantiles of individual-level treatment-effects values.

st data. *st* stands for survival time. In survival-time data, each observation represents a span of survival, recorded in variables $t0$ and t . For instance, if in an observation $t0$ were 3 and t were 5, the span would be $(t0, t]$, meaning from just after $t0$ up to and including t .

Sometimes variable $t0$ is not recorded; $t0$ is then assumed to be 0. In such a dataset, an observation that had $t = 5$ would record the span $(0, 5]$.

Each observation also includes a variable d , called the failure variable, which contains 0 or nonzero (typically, 1). The failure variable records what happened at the end of the span: 0, the subject was still alive (had not yet failed) or 1, the subject died (failed).

Sometimes variable d is not recorded; d is then assumed to be 1. In such a dataset, all time-span observations would be assumed to end in failure.

Finally, each observation in an *st* dataset can record the entire history of a subject or each can record a part of the history. In the latter case, groups of observations record the full history. One observation might record the period $(0, 5]$ and the next, $(5, 8]$. In such cases, there is a variable ID that records the subject for which the observation records a time span. Such data are called multiple-record *st* data. When each observation records the entire history of a subject, the data are called single-record *st* data. In the single-record case, the ID variable is optional.

See [ST] *stset*.

survival-time data. See *st data*.

survivor function. Also known as the survivorship function and the survival function, the survivor function, $S(t)$, is 1) the probability of surviving beyond time t , or equivalently, 2) the probability that there is no failure event prior to t , 3) the proportion of the population surviving to time t , or equivalently, 4) the reverse cumulative distribution function of T , the time to the failure event: $S(t) = \Pr(T > t)$. Also see *hazard*, *cumulative hazard*, and *hazard ratio*.

switcher. In the context of event studies, switchers are units or groups that transition across levels of treatment over time or exit treatment altogether rather than remaining at their initial treatment level. Switchers-in are those whose first switch is to a treatment level higher than their initial treatment level. Switchers-out are those whose first switch is to a treatment level lower than their initial treatment level. See also *absorbing treatment*.

TE. See *total effect*.

TNDE. See *total natural direct effect*.

total effect. The total effect (TE) is the average difference in outcomes that we expect when everyone receives the treatment versus when no one receives the treatment. In the causal mediation literature, the TE is also referred to as the total average treatment effect and the marginal total effect; it is synonymous with the *average treatment effect*.

total natural direct effect. The total natural direct effect (TNDE) is the average direct treatment effect when the mediator is held at its value associated with being treated.

For example, suppose that exercise has a direct effect on well-being as well as an indirect effect via the production of endorphins. The total natural direct effect measures the average difference in the potential outcomes for well-being when everyone in the population exercised compared with when no one exercised but with endorphins set to levels that would be observed if everyone exercised. The total natural direct effect is synonymous with the [average direct treatment effect with respect to the treated](#).

treatment model. A treatment model is a model used to predict treatment-assignment probabilities as a function of covariates and parameters.

treatment path. In the context of an event study, the treatment path of a unit or group is the sequence of treatment levels the unit or group experiences during the observed time periods. See also [absorbing treatment](#).

truncation, left-truncation, and right-truncation. In survival analysis, truncation occurs when subjects are observed only if their failure times fall within a certain observational period of a study. Censoring, on the other hand, occurs when subjects are observed for the whole duration of a study, but the exact times of their failures are not known; it is known only that their failures occurred within a certain time span.

Left-truncation occurs when subjects come under observation only if their failure times exceed some time t_l . It is only because they did not fail before t_l that we even knew about their existence. Left-truncation differs from left-censoring in that, in the censored case, we know that the subject failed before time t_l , but we just do not know exactly when.

Imagine a study of patient survival after surgery, where patients cannot enter the sample until they have had a post-surgical test. The patients' survival times will be left-truncated. This is a “delayed entry” problem, one common type of left-truncation.

Right-truncation occurs when subjects come under observation only if their failure times do not exceed some time t_r . Right-truncated data typically occur in registries. For example, a cancer registry includes only subjects who developed a cancer by a certain time, and thus survival data from this registry will be right-truncated.

unconfoundedness. See [conditional-independence assumption](#).

weighted-regression-adjustment estimator. Weighted-regression-adjustment estimators use means of predicted outcomes for each treatment level to estimate each potential-outcome mean. The weights are used to estimate censoring-adjusted regression coefficients.

Subject and author index

See the [combined subject index](#) and the [combined author index](#) in the *Stata Index*.