

# **STATA EXTENDED REGRESSION MODELS REFERENCE MANUAL RELEASE 15**



A Stata Press Publication  
StataCorp LLC  
College Station, Texas



® Copyright © 1985–2017 StataCorp LLC  
All rights reserved  
Version 15

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845  
Typeset in  $\text{\TeX}$

ISBN-10: 1-59718-235-4

ISBN-13: 978-1-59718-235-5

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** and NetCourse are registered trademarks of StataCorp LLC.

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

NetCourseNow 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. 2017. *Stata: Release 15*. Statistical Software. College Station, TX: StataCorp LLC.

# Contents

intro	Introduction	1
intro 1	An introduction to the ERM commands	4
intro 2	The models that ERMs fit	10
intro 3	Endogenous covariates features	13
intro 4	Endogenous sample-selection features	19
intro 5	Treatment assignment features	24
intro 6	Model interpretation	31
intro 7	A Rosetta stone for extended regression commands	45
intro 8	Conceptual introduction via worked example	47
eintreg	Extended interval regression	62
eintreg postestimation	Postestimation tools for eintreg	79
eintreg predict	predict after eintreg	82
eoprobit	Extended ordered probit regression	85
eoprobit postestimation	Postestimation tools for eoprobit	101
eoprobit predict	predict after eoprobit	106
eprobit	Extended probit regression	110
eprobit postestimation	Postestimation tools for eprobit	133
eprobit predict	predict after eprobit	140
eregress	Extended linear regression	143
eregress postestimation	Postestimation tools for eregress	157
eregress predict	predict after eregress	162
erm options	Extended regression model options	170
estat teffects	Average treatment effects for extended regression models	176
example 1a	Linear regression with continuous endogenous covariate	179
example 1b	Interval regression with continuous endogenous covariate	185
example 1c	Interval regression with endogenous covariate and sample selection	187
example 2a	Linear regression with binary endogenous covariate	190
example 2b	Linear regression with exogenous treatment	193
example 2c	Linear regression with endogenous treatment	196
example 3a	Probit regression with continuous endogenous covariate	202
example 3b	Probit regression with endogenous covariate and treatment	211
example 4a	Probit regression with endogenous sample selection	216
example 4b	Probit regression with endogenous treatment and sample selection	219
example 5	Probit regression with endogenous ordinal treatment	222
example 6a	Ordered probit regression with endogenous treatment	228
example 6b	Ordered probit regression with endogenous treatment and sample selection	231
predict advanced	predict's advanced features	235
predict treatment	predict for treatment statistics	238
triangularize	How to triangularize a system of equations	242

Glossary ..... 248

Subject and author index ..... 254

# Cross-referencing the documentation

When reading this manual, you will find references to other Stata manuals. For example,

[U] [26 Overview of Stata estimation commands](#)

[XT] [xtabond](#)

[D] [reshape](#)

The first example is a reference to chapter 26, *Overview of Stata estimation commands*, in the *User's Guide*; the second is a reference to the `xtabond` entry in the *Longitudinal-Data/Panel-Data 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>
[BAYES]	<i>Stata Bayesian Analysis Reference Manual</i>
[D]	<i>Stata Data Management Reference Manual</i>
[ERM]	<i>Stata Extended Regression Models 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>
[IRT]	<i>Stata Item Response Theory Reference Manual</i>
[DSGE]	<i>Stata Linearized Dynamic Stochastic General Equilibrium Reference Manual</i>
[XT]	<i>Stata Longitudinal-Data/Panel-Data 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 and Sample-Size Reference Manual</i>
[P]	<i>Stata Programming 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>
[TS]	<i>Stata Time-Series Reference Manual</i>
[TE]	<i>Stata Treatment-Effects Reference Manual: Potential Outcomes/Counterfactual Outcomes</i>
[I]	<i>Stata Glossary and Index</i>
[M]	<i>Mata Reference Manual</i>

# Title

intro — Introduction

Description      Remarks and examples

## Description

ERM stands for extended regression model. The ERMs are linear regression, interval regression, probit, and ordered probit. This manual introduces, explains, and documents ERM features.

## Remarks and examples

The entries in this manual are organized as follows:

*Introductions*  
*Examples*  
*ERM commands*  
*Postestimation*  
*Technical details*  
*Glossary*

## Introductions

Read the introductions first.

We recommend reading [ERM] [intro 1](#)–[ERM] [intro 6](#) in order. In them, you will find introductions to the models that can be fit with the ERM commands, the syntax, the complications—endogenous covariates, sample selection, and treatment assignment—that ERM commands address, and the interpretation of results.

- [ERM] [intro 1](#)    An introduction to the ERM commands
- [ERM] [intro 2](#)    The models that ERMs fit
- [ERM] [intro 3](#)    Endogenous covariates features
- [ERM] [intro 4](#)    Endogenous sample-selection features
- [ERM] [intro 5](#)    Treatment assignment features
- [ERM] [intro 6](#)    Model interpretation

The next introduction is a Rosetta stone for anyone who has used other Stata commands to account for endogenous covariates, sample selection, or nonrandom treatment assignment. It provides a simple mapping of syntax from commands such as `ivregress`, `heckman`, `ivprobit`, `heckoprobit`, and `etregress` to the corresponding ERM command. If you are already familiar with these other commands, this entry may be all you need to get started using the ERM commands.

- [ERM] [intro 7](#)    A Rosetta stone for extended regression commands

Finally, we include an introduction to important concepts in ERMs in the context of a worked example. Here, we discuss endogeneity, sample selection, and nonrandom treatment assignment. We fit models that account for each of these complications, and we show you how to use postestimation commands to interpret the results.

- [ERM] [intro 8](#)    Conceptual introduction via worked example

[ERM] [intro 8](#) can be read either before or after [ERM] [intro 1](#)–[ERM] [intro 6](#).

Examples

The example entries demonstrate how to fit models using `eregress`, `eintreg`, `eprobit`, and `eoprobit`.

We do not recommend selecting the examples you read based only on the type of outcome discussed in the example. The syntax of the ERM commands is interchangeable. Therefore, you can substitute `eintreg`, `eoprobit`, `eprobit`, or `eregress` into any of the examples to fit a model that addresses the same complications. The table below lists the command, the type of outcome variable, and the complications that are addressed in each example to help you locate examples that are of most interest to you.

Example	Command	Outcome	Complications
[ERM] <a href="#">example 1a</a>	<code>eregress</code>	continuous	continuous endogenous covariate
[ERM] <a href="#">example 1b</a>	<code>eintreg</code>	interval	continuous endogenous covariate
[ERM] <a href="#">example 1c</a>	<code>eintreg</code>	interval	continuous endogenous covariate, endogenous sample selection
[ERM] <a href="#">example 2a</a>	<code>eregress</code>	continuous	binary endogenous covariate
[ERM] <a href="#">example 2b</a>	<code>eregress</code>	continuous	exogenous treatment
[ERM] <a href="#">example 2c</a>	<code>eregress</code>	continuous	endogenous treatment
[ERM] <a href="#">example 3a</a>	<code>eprobit</code>	binary	continuous endogenous covariate
[ERM] <a href="#">example 3b</a>	<code>eprobit</code>	binary	continuous endogenous covariate, endogenous treatment
[ERM] <a href="#">example 4a</a>	<code>eprobit</code>	binary	endogenous sample selection
[ERM] <a href="#">example 4b</a>	<code>eprobit</code>	binary	endogenous sample selection, endogenous treatment
[ERM] <a href="#">example 5</a>	<code>eprobit</code>	binary	endogenous ordinal treatment
[ERM] <a href="#">example 6a</a>	<code>eoprobit</code>	ordinal	endogenous treatment
[ERM] <a href="#">example 6b</a>	<code>eoprobit</code>	ordinal	endogenous treatment, endogenous sample selection

The type of outcome does play a role in the way results are interpreted, so examples with the same outcome type will be of interest for interpretation. If your main interest is in interpretation, also see [\[ERM\] intro 6](#) and [\[ERM\] intro 8](#).

ERM commands

The entries for the individual commands provide details on syntax and implementation. The *Methods* and *formulas* sections include full details on the models that can be fit using these commands.

[ERM] <a href="#">eintreg</a>	Extended interval regression
[ERM] <a href="#">eoprobit</a>	Extended ordered probit regression
[ERM] <a href="#">eprobit</a>	Extended probit regression
[ERM] <a href="#">eregress</a>	Extended linear regression
[ERM] <a href="#">erm options</a>	Extended regression model options

## Postestimation

The postestimation commands allow you to estimate treatment effects, obtain predictions, perform tests, and more. They are documented in the entries listed below.

[ERM] <b>eintreg postestimation</b>	Postestimation tools for eintreg
[ERM] <b>eintreg predict</b>	predict after eintreg
[ERM] <b>eoprobit postestimation</b>	Postestimation tools for eoprobit
[ERM] <b>eoprobit predict</b>	predict after eoprobit
[ERM] <b>eprobit postestimation</b>	Postestimation tools for eprobit
[ERM] <b>eprobit predict</b>	predict after eprobit
[ERM] <b>eregress postestimation</b>	Postestimation tools for eregress
[ERM] <b>eregress predict</b>	predict after eregress
[ERM] <b>estat teffects</b>	Average treatment effects for extended regression models
[ERM] <b>predict advanced</b>	predict's advanced features
[ERM] <b>predict treatment</b>	predict for treatment statistics

Examples using postestimation commands are found in [ERM] [intro 8](#) and in the [example entries](#).

## Technical details

ERM commands require that endogenous covariates form a triangular or recursive system. Here, we discuss triangular systems and possible solutions if your model does not have this required form.

[ERM] <b>triangularize</b>	How to triangularize a system of equations
----------------------------	--

## Glossary

Finally, we provide a glossary that can be referred to as needed.

[ERM] <b>Glossary</b>	Glossary of technical terms
-----------------------	-----------------------------



# Title

intro 1 — An introduction to the ERM commands

[Description](#)[Remarks and examples](#)[Also see](#)

## Description

ERM stands for extended regression model. It is our term to designate

- commands for fitting linear regression, interval regression, probit, and ordered probit models that allow
- continuous, binary, and ordinal endogenous covariates, including
- polynomials of endogenous covariates,
- interactions of endogenous covariates,
- interactions of endogenous with exogenous covariates,
- endogenous sample selection, and
- nonrandom exogenous or endogenous treatment assignment.

The features may be used separately or in any combination.

The estimation commands `eregress`, `eintreg`, `eprobit`, and `eoprobit` fit ERMs.

## Remarks and examples

Remarks are presented under the following headings:

- [The problems ERMs solve](#)
- [The simple syntax of ERMs](#)
- [Normality assumption underlying ERMs](#)
- [Learning more about ERMs](#)

## The problems ERMs solve

The ERM commands fit the following models:

Command	Purpose
<code>eregress</code>	linear regression
<code>eintreg</code>	interval regression
<code>eprobit</code>	binary-outcome probit regression
<code>eoprobit</code>	ordinal-outcome probit regression

These models are described in [\[ERM\] intro 2](#).

All the ERM commands provide the following features:

- **Endogenous covariates**

Explanatory variables in the model—covariates—can be exogenous or endogenous.

Endogenous covariates can themselves be continuous (linear), binary (probit), or ordinal (ordered probit).

Endogenous covariates can be interacted with other covariates, whether endogenous or exogenous. They can even be interacted with themselves to form polynomials.

Endogenous covariates can themselves be predicted by other endogenous covariates.

- **Endogenous selection**

Models can be adjusted for situations in which outcomes are unobserved for endogenous reasons.

In a medical trial, patients may skip the final visit, causing the final outcome to be unobserved. They may skip it for reasons correlated with the outcome.

In economic data, wages are observed only for those who have a job. Those who do not have a job may not for reasons correlated with the wage they would have received.

- **Exogenous or endogenous treatment assignment**

The purpose of models is often to measure the effect of a treatment, such as a drug that is administered or a training program that is attended. Ethics often prevent assignment from being random.

In a medical trial, doctors might assign patients most likely to benefit to a trial based on observed characteristics. That is called exogenous treatment assignment.

In another situation, subjects may volunteer, and subjects who perceive larger benefits will be more likely to benefit. If all the determinants of the perceptions are observed, then assignment is exogenous. It can be explained by the observed variables, just as in the previous case.

If the determinants are unobserved, then treatment is endogenous. Errors in the assignment equation will be correlated with errors in the outcome equation.

Stata has other commands that address each of these issues in the case of linear regression, and it has still other commands that can address some of these issues for interval regression, probit, and ordered probit. But Stata has no other commands that can adjust for all the above when they occur together. Even if your problem has only one of the issues, you may still prefer to use the ERM commands because they all have the same simple syntax.

## The simple syntax of ERMs

The basic syntax of the ERM commands is Stata's standard estimation syntax: the command followed by the dependent variable followed by the covariates. Typing

```
. eregress y1 x1 x2
```

fits a linear regression of *y1* on *x1* and *x2*. If you need to use one or more ERM features, you add options to the command.

Option	Purpose
<code>endogenous()</code>	add endogenous covariates
<code>select()</code>	add endogenous sample selection
<code>tobitselect()</code>	add endogenous selection using tobit
<code>extreat()</code>	add exogenous treatment assignment
<code>entreat()</code>	add endogenous treatment assignment

You can type

```
. eregress y x1 x2, endogenous(w = x1 z1 z2)
```

to add endogenous covariate `w` to the right-hand side of the model. The option specifies that `w`'s instruments are variables `x1`, `z1`, and `z2`.

If you did not observe `y` but observed `y0` and `y1`, where  $y_0 \leq y \leq y_1$ , you could fit the equivalent interval regression by typing

```
. eintreg y0 y1 x1 x2, endogenous(w = x1 z1 z2)
```

If you observed `y` but it contained a 0/1 binary outcome, you could fit the equivalent probit model by typing

```
. eprobit y x1 x2, endogenous(w = x1 z1 z2)
```

If `y` contained 1, 2, or 3 for ordered categories, such as not ambulatory, partially ambulatory, and fully ambulatory, you could fit the equivalent ordered probit model by typing

```
. eoprobit y x1 x2, endogenous(w = x1 z1 z2)
```

Syntax is the same regardless of model fit.

Now, let's imagine that the outcome `y` is observed only when variable `selected` is true (that is, not equal to 0). Consider a case where the outcome is observed when

$$\gamma_0 + \gamma_1 x_2 + \gamma_2 w + e.\text{selected} > 0$$

and, just to make the problem more complicated, assume that `w` is endogenous. To fit the model with this added complication, type

```
. eregress y x1 x2, endogenous(w = x1 z1 z2) select(selected = x2 w)
```

You would use the same syntax with the other ERM commands:

```
. eintreg y0 y1 x1 x2, endogenous(w = x1 z1 z2) select(selected = x2 w)
. eprobit y x1 x2, endogenous(w = x1 z1 z2) select(selected = x2 w)
. eoprobit y x1 x2, endogenous(w = x1 z1 z2) select(selected = x2 w)
```

Now, let's complicate the model even more. We also have the variable `treatment`, which records whether the observation was treated. `treatment` also affects `y`. In fact, measuring the effect of `treatment` is the primary reason we are fitting this model. Type

```
. eregress y x1 x2, endogenous(w = x1 z1 z2) select(selected = x2 w) ///
    extreat(treatment)
```

Option `extreat()` handles exogenous treatment. Exogenous treatment is more flexible than you might expect. It handles assignment based on all the covariates used in the model, which in this case are `x1`, `x2`, and `w`.

But let us assume in our data that subjects volunteered. Or perhaps health care professionals assigned subjects to being treated based on information not in the model. That would be reasonable: doctors meet their patients and so know more about them than what is recorded in our data. In any case, we will assume that treatment is a function of observed variables `w`, `z2`, and `z3`, and we will assume that the error in the treatment equation is correlated with the error in the outcome equation. It is that last assumption that handles doctors knowing more about their patients than what is recorded in our data. To fit the model, we type

```
. eregress y x1 x2, endogenous(w = x1 z1 z2) select(selected = x2 w) ///
   entreat(treatment = w z2 z3)
```

We changed from exogenous to endogenous treatment by swapping option `extreat()` for `entreat()`.

Shall we continue? We are just trying to convince you how flexible ERMs are and how simple the syntax is to fit them. We will go one more step. Let's assume that `y` is not continuous but is ordinal. `y` contains 1, 2, and 3, meaning not ambulatory, partially ambulatory, and fully ambulatory. In that case, we type

```
. eoprobit y x1 x2, endogenous(w = x1 z1 z2) select(selected = x2 w) ///
   entreat(treatment = w z2 z3)
```

## Normality assumption underlying ERMs

If you are accustomed to fitting models with `regress` and `ivregress`, you expect that results do not require that the errors be normally distributed. They merely require that they be independent and identically distributed.

The results produced by ERMs share that feature when all the equations are linear. Linear excludes `eintreg`, `eprobit`, and `eoprobit`, as well as endogenous selection and endogenous treatment, both of which depend on a secondary probit model.

The nonlinear models that ERMs fit depend on normality.

## Learning more about ERMs

What follows is a useful footnote. Other Stata commands provide a subset of the features that ERMs provide. We list them below. We will discuss ERMs more in this manual, but ERMs provide so many statistical features that we do not tell you as much about them as you would like. If you would like to know more, read the documentation for the other commands and then use the ERM commands.

`eregress` provides the features of

Feature	Command
linear regression	<code>regress</code>
instrumental variables	<code>ivregress</code>
exogenous treatment assignment	<code>teffects ra</code>
endogenous treatment assignment	<code>eteffects</code> and <code>etregress</code>
endogenous sample selection	<code>heckman</code>

`eintreg` provides the features of

Feature	Command
interval regression	<code>intreg</code>
tobit regression	<code>tobit</code>
instrumental-variables interval regression	—
instrumental-variables tobit regression	<code>ivtobit</code>
exogenous treatment assignment	—
endogenous treatment assignment	—
endogenous sample selection	—

`eprobit` provides the features of

Feature	Command
probit regression	<code>probit</code>
instrumental variables	<code>ivprobit</code>
exogenous treatment assignment	<code>teffects ra</code>
endogenous treatment assignment	—
endogenous sample selection	<code>heckprobit</code>

`eoprobit` provides the features of

Feature	Command
ordered probit regression	<code>oprobit</code>
instrumental variables	—
exogenous treatment assignment	—
endogenous treatment assignment	—
endogenous sample selection	<code>heckoprobit</code>

## Also see

[ERM] [intro 2](#) — The models that ERMs fit

[ERM] [intro 7](#) — A Rosetta stone for extended regression commands

[ERM] [intro 8](#) — Conceptual introduction via worked example

# Title

## intro 2 — The models that ERMs fit

Description

Remarks and examples

Also see

## Description

The ERM commands fit linear regressions, interval regressions, probit regressions, and ordered probit regressions. These models are described below.

## Remarks and examples

Remarks are presented under the following headings:

*Linear regression models*

*Interval regression models*

*Probit regression models*

*Ordered probit regression models*

In what follows, the expression

$$\beta_1 \mathbf{x1}_i + \beta_2 \mathbf{x2}_i + \cdots + \beta_k \mathbf{xk}_i$$

arises so often that we will write it as

$$\mathbf{x}_i \boldsymbol{\beta}$$

$\mathbf{x1}$ ,  $\mathbf{x2}$ , ... are variables in your data. They are the explanatory variables—the covariates—of the models that you fit.  $\mathbf{x1}_i$ ,  $\mathbf{x2}_i$ , ... are the values of the variables in observation  $i$ .

## Linear regression models

Linear regression is for use with continuous dependent variables. To fit a linear regression, type

```
. eregress y x1 x2 ... xk
```

The model fit is

$$y_i = \beta_0 + \mathbf{x}_i \boldsymbol{\beta} + e_i.y$$

where  $e_i.y$  is the error and is assumed to be normally distributed with mean 0 and variance  $\sigma^2$ .

The fitted parameters are  $\beta_0$ ,  $\boldsymbol{\beta}$ , and  $\sigma^2$ .

When you make predictions based on linear regressions, what is predicted is the expected value of  $y$  given  $\mathbf{x}$ .

## Interval regression models

Interval regression is for use with continuous dependent variables. To fit an interval regression, type

```
. eintreg y1 y2 x1 x2 ... xk
```

The model fit is the same as that for linear regression except that  $y$  is not a variable in the dataset:

$$y_i = \beta_0 + \mathbf{x}_i\beta + e_i.y$$

The assumptions are the same as for linear regression too.  $e.y$  is assumed to be normally distributed with mean 0 and variance  $\sigma^2$ .

The fitted parameters are  $\beta_0$ ,  $\beta$ , and  $\sigma^2$ .

When you use `eintreg`, rather than specify  $y$ , the value of the dependent variable, you specify  $y1$  and  $y2$ , where

$$y1_i \leq y_i \leq y2_i$$

Variables  $y1$  and  $y2$  specify the interval in which  $y$  is known to lie. For instance, if subject 1's blood pressure were not precisely recorded but instead a box was checked reporting that the blood pressure was in the range 110 to 139, then  $y1_1$  would equal 110 and  $y2_1$  would equal 139.

If  $y1_i = y2_i$  in all observations, `eintreg` is the same as linear regression. All values are precisely observed.

If  $y1_i = y2_i$  in some observations, those observations are precisely observed.

$y1_i$  may contain a missing value and that means  $y1_i = -\infty$ . In such observations, all that is known is that  $y_i \leq y2_i$ . The observation is left-censored. If the box was checked for subject 2's blood pressure being below 120, then  $y1_2$  would equal . (missing value) and  $y2_2$  would equal 119.

$y2_i$  may contain a missing value and that means  $y2_i = +\infty$ . In such observations, all that is known is that  $y_i \geq y1_i$ . The observations are right-censored. If the box was checked that subject 3's blood pressure was above 160, then  $y1_3$  would equal 161 and  $y2_3$  would equal . (missing value).

If both  $y1_i$  and  $y2_i$  contain missing values, then all that is known is that  $-\infty \leq y_i \leq \infty$ , and the observation is ignored when fitting the model.

`eintreg` can be used to fit tobit models. Assume that you have data in which  $y$  is left-censored at 0. To fit a tobit model, type

```
. generate y1 = cond(y==0, ., y)
. generate y2 = y
. eintreg y1 y2 x1 x2 ... xk
```

When you make predictions based on interval regressions, `predicted` is the expected value of the dependent variable, the unobserved  $y$ , conditioned on the covariates.

## Probit regression models

Probit regression is for use with binary dependent variables. To fit a probit regression, type

```
. eprobit y x1 x2 ... xk
```

Variable  $y$  in theory should contain the values 0 and 1, but `eprobit` does not require that. It treats all nonzero (and nonmissing) values as if they were 1, which means a positive outcome, such as “subject was hired” or “subject tested positive”. The positive result can be a negative event, such as “subject died”.

The model is

$$p_i = \Pr(\text{positive outcome in obs. } i) = \Pr(\beta_0 + \mathbf{x}_i\beta + e_i.y) > 0$$



where  $e_i y$  is assumed to be normally distributed with mean 0 and variance 1. With that assumption, the probability of a positive outcome is

$$p_i = \text{normal}(\beta_0 + \mathbf{x}_i \beta)$$

The fitted parameters are  $\beta_0$  and  $\beta$ .

When you make predictions based on probit regressions, predicted is the probability of a positive outcome conditional on the covariates.

## Ordered probit regression models

Ordered probit regression is for use with ordinal dependent variables. To fit an ordered probit regression, type

```
. eoprobit y x1 x2 ... xk
```

Variable  $y$  is expected to contain 1, 2, ...,  $M$  indicating category number although, just like [oprobit](#), `eoprobit` is less demanding.  $y$  could contain values 2, 3, 5, and 8 to indicate four ordered categories. What is important is that the categories have a natural ordering and that the numbers used to represent them order the categories in the same way. `eoprobit` could be used with the ordered categories 1) not ambulatory, 2) partially ambulatory, and 3) fully ambulatory. Or the order of the categories could be reversed: 1) fully ambulatory, 2) partially ambulatory, and 3) not ambulatory. Reversing the order reverses the signs of the fitted coefficients but does not substantively change the model.

The model fit is

$$\begin{aligned} p_{m,i} &= \Pr(\text{outcome } m \text{ in obs. } i) \\ &= \Pr(c_{m-1} \leq \mathbf{x}_i \beta + e_i y \leq c_m) \end{aligned}$$

where  $e_i y$  is assumed to be normally distributed with mean 0 and variance 1. Thus, the probability that the outcome is  $m$  is

$$p_{m,i} = \text{normal}(c_m - \mathbf{x}_i \beta) - \text{normal}(c_{m-1} - \mathbf{x}_i \beta)$$

where  $c_0$  and  $c_M$  are  $-\infty$  and  $+\infty$ , and  $c_1, \dots, c_{M-1}$  are fit from the data. The  $c$  values play the role of intercepts and are called cutpoints.

The fitted parameters are  $\beta$  and  $c_1, \dots, c_{M-1}$ .

When  $M = 2$ , the ordered probit model reduces to the probit model with  $c_0 = -\beta_0$ .

When you make predictions based on ordered probit regressions, predicted are the probabilities of the dependent variable equaling each category conditional on the covariates.

## Also see

[ERM] [eintreg](#) — Extended interval regression

[ERM] [eoprobit](#) — Extended ordered probit regression

[ERM] [eprobit](#) — Extended probit regression

[ERM] [eregress](#) — Extended linear regression

## Description

Whether you fit linear regressions, interval regressions, probits, or ordered probits, the ERM commands provide the same features. One of those features is endogenous covariates, which are explained below.

## Remarks and examples

Remarks are presented under the following headings:

[What are endogenous and exogenous covariates?](#)

[Solving the problem of endogenous covariates](#)

[Solving the problem of reverse causation](#)

[You can interact endogenous covariates](#)

[You can have continuous, binary, and ordered endogenous covariates](#)

[You can have instruments that are themselves endogenous](#)

## What are endogenous and exogenous covariates?

Consider the model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + e.y$$

In models like this one,  $y$  is called the dependent variable or the outcome variable.  $x_1$  and  $x_2$  are called explanatory variables, exogenous variables, or (exogenous) covariates; we will simply call them covariates.  $e.y$  is called the error.

For ERMs or any regression estimator to meaningfully fit models like the one above, it is required

1. that there be no omitted (confounding) variables that are correlated with  $x_1$  or  $x_2$ .
2. that  $x_1$  and  $x_2$  be measured without error.
3. that there be no reverse causation.  $x_1$  and  $x_2$  affect  $y$ , but  $y$  must not affect  $x_1$  or  $x_2$ .
4. that  $x_1$  and  $x_2$  not be correlated with  $e.y$ .

Any covariate that meets these requirements is called exogenous. Covariates that are not exogenous are endogenous.

## Solving the problem of endogenous covariates

What if  $x_1$  violated some of or all the requirements? What if  $x_1$  was endogenous? Solving the problem of endogenous covariates is straightforward. You find a variable or set of variables that affect  $x_1$  but do not affect  $y$  except through their effect on  $x_1$ . As those variables change, they induce a change in  $x_1$ . That change in turn induces a change in  $y$ , and because that change is known to be caused only by the change in  $x_1$ , the change can be used to disentangle the problem.

The variables that you use to solve the endogenous covariate problem are called instrumental variables.

In this manual, we tend to use the following notation:

Name starts with	Signifies
y	dependent variable
x	exogenous covariate
w	endogenous covariate
z	instrumental variable

Note: The above is notation, not a naming requirement. The software does not require that variables be named this way.

Because we are now assuming that `x1` is an endogenous covariate, let us rename it `w1` and rewrite our model:

$$y = \beta_0 + \beta_1 w1 + \beta_2 x2 + e.y$$

To fit this model, we need one or more variables to serve as instruments for `w1`. Those variables need to be correlated with `w1` and uncorrelated with `y`. Let `z1` and `z2` be two such variables. Finding `z1` and `z2` is more easily said than done, and how you find them is beyond the scope of this manual. Nonetheless, two examples would not be out of order.

1. An economist needed an instrument for income and used spouse's income. Incomes of spouses are correlated, and in the research problem, there was no reason to suspect that spouse's income would affect the outcome other than through the correlation.
2. A health researcher needed an instrument for whether patients were prescribed a new drug. In the research problem, that variable might be endogenous because doctors are more likely to prescribe drugs they expect will be beneficial to patients based on characteristics unobserved in the data. The researcher used whether the drug was on formulary for the patients' insurance as an instrument because it is expected to be correlated with whether the drug was prescribed but not with the outcome.

Anyway, find one or more variables that are correlated with `w1` but not with the dependent variable except through the effect on `w1`. We will assume variables `z1` and `z2` meet the criteria. We can then fit a model with `w1` as a covariate by typing

```
. eregress y x2, endogenous(w1 = z1 z2)
```

The model has two covariates: exogenous covariate `x2` and endogenous covariate `w1`. `w1` was added to the model by the `endogenous()` option. If we wished, we could type `w1` among the covariates, but then we have to specify `endogenous()`'s option `nomain` so that it does not add `w1` for us. We could type

```
. eregress y x2 w1, endogenous(w1 = z1 z2, nomain)
```

Whichever syntax we use, we are using `z1` and `z2` as instruments for `w1`. There is a third instrument we could add to `z1` and `z2`. If we wanted, we could add `x2` by typing

```
. eregress y x2 w1, endogenous(w1 = z1 z2 x2, nomain)
```

We can add `x2` because it is probably correlated with `w1`, and it most certainly affects `y`, and it is exogenous. We at StataCorp would add `x2` almost by reflex. We explain why below.

## Solving the problem of reverse causation

Instrumental variables can solve the four problems we mentioned at the beginning of this section.

1. They can solve the problem of omitted variables that are correlated with  $w_1$ .
2. They can solve the problem of  $w_1$  being measured with error.
3. They can solve the problem of reverse causation, meaning that  $y$  affects  $w_1$ .
4. They can solve the problem of  $x_1$  and  $x_2$  being correlated with  $e.y$ .

We are not saying that we have all those problems, but instrumental variables can solve them if we do.

If we do not include  $x_2$  among the instruments, however, problem 3 is not handled. We must include all the exogenous variables predicting  $y$  to handle reverse causation. In the model above, we have only one exogenous variable. If our model had been

$$y = \beta_0 + \beta_1 w_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + e.y$$

we would have included all of them:

```
. eregress y w1 x2 x3 x4, endogenous(w1 = z1 z2 x2 x3 x4, nomain)
```

This solution to reverse causation works with linear models, meaning `eregress` and `eintreg`. It does not work with `eprobit` and `eoprobit`. There is no solving the reverse-causation problem for those models.

## You can interact endogenous covariates

What we have said so far about endogenous covariates applies not only to ERM commands but also to estimation commands in general.

A feature unique to ERMs is that you can use endogenous covariates in interactions. For instance, `eregress` can fit a model including

```
. eregress y w1 i.x2 i.x2#c.w1, endogenous(w1 = z1 i.x2, nomain)
```

In this model, we are assuming that  $x_2$  is a dummy variable, such as attends school.  $x_2$  is 1 when subjects attend school and is 0 otherwise. Therefore, we use `i.` factor-variable notation when we include  $x_2$  in the model. The right-hand-side variables in this model are

$w_1$	a continuous, endogenous variable
<code>i.x2</code>	attends school
<code>i.x2#c.w1</code>	attends school multiplied by $w_1$

The coefficients on these variables are

$\beta_1$	effect of the endogenous continuous covariate
$\beta_2$	effect of attending school
$\beta_3$	extra effect of $w_1$ when attending school

`eregress` can fit this model. Stata's other instrumental-variable regression command `ivregress` could not. It would complain about the interaction `i.x2#c.w1` because of a limitation on how the usual statistical formulas work. Interactions with endogenous covariates are not allowed.

`eregress` has no difficulty with such models.

Now, we will tell a different backstory about  $y$ ,  $w1$ , and  $x2$ :

$y$	income, job satisfaction, etc.
$w1$	years of schooling after high school
$i.x2$	dummy for schooling, whatever the level, being in a STEM subject

STEM stands for science, technology, engineering, and math. In a model such as

```
. eregress y i.x2 w1 i.x2#c.w1, endogenous(w1 = z1 i.x2, nomain)
```

extra years of schooling increase  $y$  by  $\beta_2$  for non-STEM and by  $\beta_2 + \beta_3$  for STEM.

ERMs not only allow interactions of endogenous with exogenous covariates, but they also allow interactions of endogenous with endogenous covariates and even allow endogenous covariates to be interacted with themselves! Here is an example:

```
. eregress y w1 c.w1#c.w1 i.x2, endogenous(w1 = z1 i.x2, nomain)
```

In this model, the term  $c.w1\#c.w1$  means  $w1^2$ . Years of schooling after high school would increase  $y$  by  $\beta_2 w1 + \beta_3 w1^2$ .

You can also interact endogenous covariates with other endogenous covariates, such as

```
. eregress y w1 w2 c.w1#c.w2 i.x2, endogenous(w1 = z1 i.x2, nomain) ///
                             endogenous(w2 = z2 i.x2, nomain)
```

You can tell your own story about this model.

## You can have continuous, binary, and ordered endogenous covariates

We have discussed continuous endogenous covariates. ERMs also allow binary and ordinal covariates. Consider the model

```
. eregress y w1 i.x2, endogenous(w1 = z1 i.x2, nomain)
```

Obviously,  $w1$  is an endogenous covariate. In the previous section, we speculated that  $w1$  was years of schooling beyond high school, but what if  $w1$  was instead a dummy variable for having a college degree?

If you used the above model as typed, you would be using the linear probability model to handle  $w1$ . Saying that makes the situation sound better than it is. Probabilities are bounded by 0 and 1, and you would be using a linear model to fit them, meaning that some of the predicted probabilities could be below 0 or above 1. You ordinarily would have to live with that. With ERMs, you have a better alternative. You can tell `eregress` to use the probit model to handle  $w1$ ! You type

```
. eregress y i.w1 i.x2, endogenous(w1 = z1 i.x2, probit nomain)
```

In the equation for  $y$ , we now include  $w1$  as a factor variable,  $i.w1$ .

Interactions are allowed with binary endogenous covariates just as they are allowed with continuous endogenous covariates. You could type

```
. eregress y i.x2 i.w1 i.x2#i.w1, endogenous(w1 = z1 i.x2, probit nomain)
```

$w1$  could even be an ordered categorical variable. We have imagined that  $w1$  contains values 0 and 1, with 1 meaning schooling in a STEM subject. Let's imagine that  $w1$  contains the values 1, 2, and 3, with 1 meaning a non-STEM program, 2 meaning a mixed program with some courses from a STEM program, and 3 meaning a STEM program. To fit this model, all we have to do is change `probit` to `oprobit`:

```
. eregress y i.x2 i.w1 i.x2#i.w1, endogenous(w1 = z1 i.x2, oprobit nomain)
```

## You can have instruments that are themselves endogenous

When we type

```
. eregress y w1 x2, endogenous(w1 = z1 z2, nomain)
```

we are specifying a model with an endogenous covariate and handling the problem of its endogeneity with the instruments `z1` and `z2`. The instruments we specified are exogenous in this example, but the ERM commands do not require that. If `z1` had one more of the problems we outlined at the beginning of this manual entry, then it would be endogenous and we might solve the problem that it raises by typing

```
. eregress y w1 x2, endogenous(w1 = z1 z2, nomain) endogenous(z1 = z3, nomain)
```

That could be the end of the story. ERMs can fit the above model.

We would have yet another problem, however, if `z1` also depended on `w1`. ERMs cannot fit models in which one dependent variable depends on another that depends on the first. The following model has that problem:

```
. eregress y w1 x2, endogenous(w1 = z1 z2, nomain) ///
    endogenous(z1 = w1 z3, nomain)
```

If we tried to fit the model, the command would complain:

```
. eregress y w1 x2, endogenous(w1 = z1 z2, nomain)
> endogenous(z1 = w1 z3, nomain)
    endogenous variables do not form a triangular system
    The problem may be fixable. See triangularizing the system.
r(459);
```

The message says that the system needs to be triangular, which is another way of saying the system cannot have simultaneous causation. Do not confuse simultaneous causation with reverse causation, which we previously discussed. Reverse causation concerns one equation, its dependent variable, and a covariate. The covariate affects the dependent variable, and the dependent variable affects the covariate. Simultaneous causation concerns two or more equations. Their dependent variables are mutually dependent.

Nonetheless, the workaround for simultaneous causation is a variation on the workaround for reverse causation. If the equations involved are both linear, take one of them, remove the offending endogenous variable, and substitute the removed variable's exogenous variables.

The two equations involved in this example are

```
endogenous(w1 = z1 z2, nomain)
endogenous(z1 = w1 z3, nomain)
```

We could remove `z1` from the first equation and substitute `z3`. Or we could remove `w1` from the second equation and substitute `z2`. Doing the former results in

```
. eregress y w1 x2, endogenous(w1 = z3 z2, nomain) ///      (1)
    endogenous(z1 = w1 z3, nomain)
```

Doing the latter results in

```
. eregress y w1 x2, endogenous(w1 = z1 z2, nomain) ///      (2)
    endogenous(z1 = z2 z3 nomain)
```

ERMs can fit either model, and results for the main equation will be the same.

The first solution's equation for  $z_1$  has an odd feature. The equation for variable  $z_1$  is irrelevant because  $z_1$  appears nowhere else in the model. We could omit the unnecessary equation and fit the model by typing

```
. eregress y w1 x2, endogenous(w1 = z3 z2, nomain) (3)
```

That will produce the same result too.

Statistically, all the solutions are equally good. Numerically, (3) is sometimes better because it is easier for ERMs to fit models with fewer equations.

In any case, these solutions were available to us because the models involved were linear. Had they been nonlinear, there would have been no solution.

If you want to read more about this problem and its solution, see [\[ERM\] triangularize](#).

## Also see

[\[ERM\] intro 8](#) — Conceptual introduction via worked example

[\[ERM\] triangularize](#) — How to triangularize a system of equations

Description

Endogenous sample-selection problems are handled by the `select()` option. ERMes provide probit and tobit selection. Probit selection is discussed below. Tobit selection is a variation on probit selection that uses censoring of a normal variable as an indicator of selection.

Remarks and examples

Remarks are presented under the following headings:

- Is sample selection a concern in your research problem?*
- The problem and solution of endogenous sample selection*
- Endogenous sample selection handles missing not at random*
- Endogenous sample selection can be used with other features of ERMes*
- Mechanical notes*

Is sample selection a concern in your research problem?

Say that you wish to fit the model

```
. eregress y x1 x2
```

We will tell you two stories about it. In the first, *y* is wage-and-salary income. In the second, *y* is a health outcome for people with a certain malady.

Both of these stories have issues of sample selection. Wages are observed only for people who work. Health outcomes are observed only for people with the malady who seek treatment. Do you care? You might not.

If you are an economist studying the effects of education, you might be perfectly satisfied measuring the return to schooling in terms of increased income of those who work. This would certainly be the situation if you were performing research to determine how schools could be improved.

If you are a medical researcher studying the effect of a treatment, you might be perfectly satisfied measuring the effect of the treatment on those who currently seek it. This would certainly be the situation if you were performing research to determine how the treatment could be improved.

Sample selection is of concern only when changing the selected population—those who work or those who are treated—is under consideration.



## The problem and solution of endogenous sample selection

We wish to fit the model

```
. eregress y x1 x2
```

We observe  $y$  for some of but not all the sample. We observe  $x_1$  and  $x_2$  for the entire sample.

For instance, we might be doing a study of a walking program run by hospitals for patients after heart attacks. Doctors prescribe the program to patients who they believe will benefit. After six months in the program, recorded for each patient is

y	Meaning
1	I feel worse (tired)
2	I feel about like I did when I started the program
3	I feel better

The variable  $y$  will be missing for some of the observations in the data. Those observations correspond to the patients who were not prescribed the program.  $y$  could also be missing if patients were prescribed but dropped out of the program—were lost to follow-up—but we will ignore that right now. We will discuss lost to follow-up in [\[ERM\] intro 5](#).

Variable  $y$  is an ordinal variable, so rather than fitting the model by using `eregress`, we will fit it by using `eoprobit`:

```
. eoprobit y x1 x2
```

Do not type that command yet. If you did, the model would be fit using only the observations on patients who were prescribed the program, because  $y$  is missing otherwise. We are about to discuss those other patients. In fact, let’s create a variable indicating whether patients were selected for inclusion in the program—we will need it later.

```
. generate selected = !missing(y)
```

There are two types of sample selection: exogenous and endogenous. Hardly any issues are created by exogenous sample selection. The real problems are raised by endogenous selection, and to discuss those issues, we need to tell you more about the walking program.

Doctors prescribed the program to their patients based on each patient’s  $x_1$  and  $x_2$  values. Those variables are believed to predict how much a patient would benefit from the program. Indeed, patients in especially poor health might actually be harmed by the program. Say that we are conducting research to evaluate how well  $x_1$  and  $x_2$  predict a benefit and to consider whether the criteria for being prescribed the program should be loosened or tightened. Would extending the program to more patients be beneficial? Or is the program already being used by too many?

That the sample was selected on  $x_1$  and  $x_2$  causes no statistical issues, although it can cause complications. Assume that doctors also based their decisions on  $x_3$  but that was for administrative reasons. That sounds horrible, but it is not necessarily bad; for example, if a patient lives far from the hospital, the doctor might not prescribe the hospital’s walking program as readily. In any case,  $x_3$ , the distance a patient lives from the hospital, affected the decision but is not believed to affect how beneficial the program is for the patient. If we are certain about that, we can ignore  $x_3$ . If we are uncertain, we should add  $x_3$  to the model to verify that the effect really is 0.

The above situation is called exogenous sample selection. It is not a reasonable story, but perhaps you do not yet see why. Anyway, if the only problem is exogenous sample selection, we can ignore it, and the only issue we have is to decide whether to include `x3` in our model. We can fit the model by typing

```
. eoprobit y x1 x2
```

or

```
. eoprobit y x1 x2 x3
```

Typing those commands is equivalent to typing

```
. eoprobit y x1 x2 if selected
```

or

```
. eoprobit y x1 x2 x3 if selected
```

We mention this merely to emphasize that because `y` is missing in the group for which `selected` is 0, all observations for which `selected` is 0 are omitted from the estimation subsample.

The problem with the above story is that doctors know more about their patients than we do. They know more than what is recorded in our database. Doctors meet with their patients and get to know them, and doctors factor everything they know into their decisions. Doctors prescribed the walking program to patients who they believed would benefit. They predicted the benefit on the basis of `x1`, `x2`, and `x3`, as well as on information they know about the patients that is not recorded in the data.

Think of the decision that doctors make as a probit model:

$$\begin{aligned} p &= \text{Pr}(\text{prescribed}) \\ &= \text{Pr}(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + e_i \cdot \text{selected} > 0) \end{aligned}$$

The important part of this model is `e.selected`. The error includes everything doctors know about their patients that is not recorded in the data. Because doctors presumably are making decisions in the patients' best interest, `e.selected` will be correlated positively with `e.y`, which is the error in the model's main equation fit by

```
. eoprobit y x1 x2
```

If we fit the model ignoring this correlation, we would obtain results suitable for predicting outcomes among those who participated in the program but not among those who did not participate.

It is the nonzero correlation of `e.y` and `e.selected` that makes the sample-selection endogenous. `eoprobit` will produce estimates accounting for the correlation if we specify the `select()` option:

```
. eoprobit y x1 x2, select(selected = x1 x2 x3)
```

`eoprobit` will report  $\hat{\rho}$ —the estimate of the correlation between the two errors—and it will report the coefficients in the outcome and selection models. Because we have now accounted for the endogenous sample selection, we can interpret the results in terms of the full population, not just those who were prescribed the treatment.

## Endogenous sample selection handles missing not at random

`select()` can handle cases in which data are missing not at random (MNAR), also known as nonignorable missing data. It can handle them as long as that missingness is modeled in the `select()` equation. It can solve the problem of missing on unobservables.

## Endogenous sample selection can be used with other features of ERMs

You can use `select()` with other features of ERMs, that is, with endogenous covariates and with treatment effects. We have not discussed treatment effects yet. We will show you an example of treatment effects with endogenous sample selection in the next section, [ERM] intro 5.

In the meantime, we will show you one way that `endogenous()` can be used with `select()`. Above, we fit the model

```
. eoprobit y x1 x2, select(selected = x1 x2 x3)
```

In the story we told, `x3` measured an administrative reason we think affected doctors' decisions to prescribe the walking program. Let's imagine that `x3` was endogenous for one reason or another. In the original story, `x3` was the distance a patient lived from the hospital. Perhaps its value is measured with error. Or perhaps `x3` represents some other administrative reason we think is correlated with `y`. Because it is endogenous, we will now refer to this variable as `w3` instead of `x3`. We can address the problem by using the `endogenous()` option:

```
. eoprobit y x1 x2, select(selected = x1 x2 w3) endogenous(w3 = z1 z2, nomain)
```

We included suboption `nomain` because we do not want `w3` to be added to the main equation. `w3` appears only in the selection equation in this model.

Be careful not to omit `nomain` when it is necessary. Endogenous covariates can appear in the main equation, the selection equation, or both. Consider another example in which `x3` is not endogenous but `x2` is. Let's call it `w2` instead of `x2`. We could fit that model by typing

```
. eoprobit y x1, select(selected = x1 w2 x3) endogenous(w2 = z3 z4)
```

`w2` will appear in the main equation because we did not also specify `nomain`. Some users always type `nomain` and explicitly specify all the covariates that appear in the main equation. You could fit the same model by typing

```
. eoprobit y x1 w2, select(selected = x1 w2 x3) endogenous(w2 = z3 z4, nomain)
```

## Mechanical notes

When you specify

```
. eoprobit y ..., select(selected = ...)
```

you can specify variables in just the `y` equation, just the `selected` equation, or both. When the same variables are specified in both equations, it is called functional-form identification. Statistically speaking, the situation would be better if there were some covariates that appeared in the `selected` equation that did not appear in the main equation, but no one is suggesting that you add irrelevant covariates to your model. Still, you should think about whether you have any such variables. We found such a variable (`x3`) in the story above.

## Also see

[ERM] [intro 8](#) — Conceptual introduction via worked example

## Description

ERMs can fit treatment-effect models. Treatment can be binary (not treated or treated) or ordinal (not treated or treated or treated extremely).

Option `extreat()` specifies exogenous treatment effects.

Option `entreat()` specifies endogenous treatment effects.

ERM's treatment-effect features are explained below.

## Remarks and examples

Remarks are presented under the following headings:

[What are treatment-effect models?](#)

[Endogenous and exogenous treatment effects](#)

[Binary and ordinal treatment effects](#)

[Sample versus population standard errors](#)

[Using treatment effects with other ERMs](#)

[Using treatment effects with other features of ERMs](#)

[Using `treat\(\)` and `select\(\)` to handle lost to follow-up](#)

[Treatment statistics reported by `estat teffects`](#)

## What are treatment-effect models?

Let's consider a simple binary treatment-effect problem. A treatment is applied to some patients, and we want to measure its effect. We start by imagining that patients are assigned randomly to the treated group. We observe a continuous outcome  $y$ , such as blood pressure, and we think the treatment affects  $y$ . We think the treatment's effect varies with patients' age,  $x1$ .

Here is one way we could fit the model:

```
. eregress y x1 i.treated i.treated#c.x1
```

Variable `treated` specifies which patients were treated. It contains 1 or 0. The model we just fit is

$$y_i = \beta_0 + \beta_1 x1_i + \beta_2 \text{treated}_i + \beta_3 \text{treated}_i x1_i + e_i.y$$

This model says that the outcome for patients who are not treated is

$$y_i = \beta_0 + \beta_1 x1_i + e_i.y$$

For those treated, the outcome is

$$y_i = (\beta_0 + \beta_2) + (\beta_1 + \beta_3) x1_i + e_i.y$$

$\beta_2$  and  $\beta_3 x1$  measure the effect of being treated. That effect varies observation by observation with the patient's age. Many researchers would stop here, satisfied to have the fitted coefficients.

Researchers who fit treatment-effect models, however, usually want to know the average treatment effect (ATE). We could obtain it.

We would calculate the average outcome when being treated over the entire dataset, the average outcome when not being treated over the entire dataset, and subtract the two results to obtain an ATE for this group of patients. It would not be difficult.

We would calculate new variable `if_not_treated` equal to

$$\text{if\_not\_treated}_i = \hat{\beta}_0 + \hat{\beta}_1 \mathbf{x1}_i$$

We would calculate new variable `if_treated` equal to

$$\text{if\_treated}_i = \text{if\_not\_treated}_i + \hat{\beta}_2 + \hat{\beta}_3 \mathbf{x1}_i$$

Then, we would subtract them:

$$\text{diff}_i = \text{if\_treated}_i - \text{if\_not\_treated}_i$$

We would finally calculate the mean of `diff`:

```
. summarize diff
```

Stata's `margins` command is wonderful at doing things like this, and it even reports the standard error! And we do not even have to calculate `if_not_treated`, `if_treated`, and `diff`. Instead, we just type

```
. margins r.treated
```

The above is what ERMs would do for us if we started by typing

```
. eregress y x1, extreat(treated)
```

and then typed `estat teffects`. `estat teffects` reports the ATE.

```
. estat teffects
```

## Endogenous and exogenous treatment effects

ERMs can fit models far more complicated than the model we just fit. That is good because the story we told above is too simple. For instance, we said that patients were assigned randomly. Ethical considerations sometimes prevent that.

There are two types of treatment effects: exogenous and endogenous. What distinguishes them is the same thing that distinguished them in [ERM] [intro 4](#), where we discussed exogenous and endogenous selection effects. It matters whether the error in the selection (treatment assignment) equation is correlated with the error in the main equation.

Here are four examples of treatment assignment.

1. Assignment is random (as above). In this case, the assignment equation contains only an error, `e.treated`, and it is uncorrelated with `e.y`. Treatment is exogenous.
2. Assignment is determined by hard-and-fast rules. There is no `e.treated`, or if you prefer, it is 0. Either way, it is uncorrelated with `e.y`. Treatment is exogenous.

3. Assignment is determined in part by hard-and-fast rules, but if the patient meets them, a coin is flipped to determine whether the patient is treated or untreated. *e.treated* is the coin flip, and it is uncorrelated with *e.y*. Treatment is exogenous.
4. Assignment is by whatever rules, if any, plus unobserved judgment. Thus, judgment appears in *e.treated*, and we must consider the possibility that it is correlated with *e.y*. Treatment is endogenous.

ERMs can fit models with exogenous or endogenous treatment assignment. You specify the `extreat()` or `entreat()` option. In the four examples above, you would specify

1. `extreat(treated)`
2. `extreat(treated)`
3. `extreat(treated)`
4. `entreat(treated = ...)`

You could fit

```
. eregress y x1 x2 x3, extreat(treated)
```

or

```
. eregress y x1 x2 x3, entreat(treated = x1 z1 z2)
```

Whichever you type, you can obtain the ATE by typing

```
. estat teffects
```

## Binary and ordinal treatment effects

We have been assuming that treatment is binary. ERM's can also fit ordinal treatment models. Think of these models as all being the same treatment but of different intensities. For instance,

1. A rehabilitative exercise program might be attended not at all, once a week, or twice a week.
2. A drug might be administered in different dosages.
3. A jobs program might be attended not at all, once a week, or twice a week.
4. The amount of post-secondary education could be none, some college, graduated, or graduated plus postgraduate.

When treatment is ordinal, variable `treated` contains more than two values. The variable might contain 0, 1, or 2; or 1, 2, or 3; or even 2, 3, or 5. If there are four ordered treatments, the variable contains four different values. The particular values do not matter as long as the numeric values order the treatments in the way they should be ordered.

When the treated variable takes on more than two values, `entreat()` fits the endogenous treatment equation by using ordered probit instead of binary probit.

## Sample versus population standard errors

Researchers who fit treatment models usually want population standard errors.

When we fit the treatment model by hand, not using the `extreat()` or `entreat()` option, we typed

```
. eregress y x1 i.treated i.treated#c.x1
. margins r.treated
```

When we used the `extreat()` option to fit the same model, we used `estat teffects` to obtain the ATE. We typed

```
. eregress y x1, extreat(treated)
. estat teffects
```

In both cases, the standard errors reported for the ATE were the same. The data were treated as fixed and not as a draw from the underlying population.

The standard error would also be that way if we fit a model with endogenous instead of exogenous treatment assignment.

```
. eregress y x1 x2 x3, entreat(treated = x1 z1 z2)
. estat teffects
```

Researchers fitting treatment-effect models often want standard errors for ATEs suitable for predicting to the entire population and not just this particular sample. If you want population-based standard errors, you must fit the model by using the `vce(robust)` option:

```
. eregress y x1, extreat(treated) vce(robust)
. estat teffects
```

Do that and `estat teffects` will report population-based standard errors.

Returning to the `eregress` command, when you do not specify `vce(robust)`, it reports OIM standard errors. OIM stands for observed information matrix. The alternative robust standard errors assume less and are therefore less efficient. While less efficient, robust standard errors still have correct coverage. The standard errors themselves just have more sampling variability. Robust standard errors are absolutely required if `estat teffects` is to report standard errors for the effect in the population.

Requesting the ATE with population standard errors makes sense only if the sample you are using is an unbiased random draw from the population for which you wish to make predictions. If the sample is not, you need to specify your data's probability sampling weights as well. Type

```
. eregress y x1 [pw = weight], extreat(treated) vce(robust)
. estat teffects
```

In this case, you can omit the `vce(robust)` option because it is assumed when probability sampling weights are specified.

Variable `weight` contains inverse probabilities that the observations were sampled from the population. For instance, if some observations were sampled with probability 0.001 and others with 0.0001, then `weight` contains 1,000 and 10,000. For our purposes here, the scale of weights does not matter, so `weight` could just as well contain 1 and 10. Scale of weights matters when you request totals, which `estat teffects` does not produce.

## Using treatment effects with other ERMs

The outcome variable `y` need not be continuous. It can be interval, binary, or ordinal, meaning that you can use the `eintreg`, `eprobit`, or `eoprobit` command to fit the model.

If we had a binary outcome variable, we would type

```
. eprobit y x1 x2 x3, entreat(treated = x1 z1 z2)
```

If we planned on obtaining the ATE with population standard error, we would type

```
. eprobit y x1 x2 x3, entreat(treated = x1 z1 z2) vce(robust)
. estat teffects
```



## Using treatment effects with other features of ERMs

`extreat()` and `entreat()` can be used with `endogenous()` and `select()`. Said differently, treatment models can contain endogenous covariates and be adjusted to handle endogenous sample selection.

By now, you are familiar with the `endogenous()` option. Some examples of `eregress` used with `extreat()` and `entreat()` are

```
. eregress y x1 x2 w1, extreat(treated) endogenous(w1 = x1 z1 z2, nomain)
. eregress y x1 x2 w1, entreat(treated = z3 w1) ///
    endogenous(w1 = x1 z1 z2, nomain)
. eregress y x1 x2, entreat(treated = z3 w1) endogenous(w1 = x1 z1 z2, nomain)
```

We used the `nomain` suboption and explicitly included `w1` in the main equation if we wanted it there. In those cases, we could have omitted the explicit mention and deleted option `nomain`. Equivalent to the first example is

```
. eregress y x1 x2, extreat(treated) endogenous(w1 = x1 z1 z2)
```

Next, we consider use of `entreat()` and `extreat()` with `select()` to account for endogenous and exogenous sample selection.

We wish to fit a treatment-effect model but there is a problem. The treatment-effect model we want to fit is

```
. eregress y x1 x2, entreat(treated = x1 z3)
```

The problem is that the information on `y` was collected at the end of the study, and some patients never showed up—they dropped out along the way. To fit the desired model with the complication, we type

```
. generate selected = !missing(y)
. eregress y x1 x2, entreat(treated = x1 z3) select(selected = x1 z4 z5)
```

To obtain the ATE, we type

```
. estat teffects
```

The model reported by `eregress` and the ATE reported by `estat teffects` will be adjusted for both the endogenous treatment assignment and the endogenous selection effects. The latter adjusts for the censored observations in which the final outcome `y` was not observed.

Reported were sample statistics. If we had wanted population statistics, we would have typed

```
. eregress y x1 x2, entreat(treated = x1 z3) select(selected = x1 z4 z5) ///
    vce(robust)
. estat teffects
```

If treatment assignment had been exogenous, we would have specified `extreat(treated)` instead of `entreat(treated = x1 z3)`.

Note in the above example that treatment can have one arm as in the story we told or it can have multiple arms. In other words, the treatment can be binary or ordinal. Nothing we typed would need to change.

## Using `treat()` and `select()` to handle lost to follow-up

The important feature of the above example is that the censored observations were lost to follow-up. By that, we mean that the patients did not report for the final meeting, and thus `y` was unobserved. Specifying the `select()` equation allowed the error in selection (that is, the unobserved reasons that subjects showed up or did not show up) to be correlated with the error in the main outcome equation (the error in the benefit of the treatment). It also allowed the error in selection to be correlated with the error in the treatment assignment. Said statistically, all endogeneity issues were handled.

This was all possible because the treatment arm was assigned even for the censored observations. Variable `treated` was not missing. It contained a treatment-arm value just as it does in all the other observations.

What if that is not true? What if censoring occurred before the treatment arm was assigned? Then we have an issue we need to discuss. First, here is how you determine whether your data have this issue. Type

```
. assert !missing(treat) if selected==0
```

If `assert` reports that the assertion is false, your data have this issue. You have censored observations for which the treatment arm is unassigned.

ERMs handle this issue differently for exogenous and endogenous treatment assignment. If you are fitting an exogenous treatment model,

```
. eregress y x1 x2, extreat(treated) select(selected = x1 z4 z5)
```

ERMs do not care that the treatment arm is missing. Endogenous selection will be fully handled just as if the treatment arm had been observed. That is, ERM handles the issue as long as the treatment arm does not appear as an explanatory variable in your selection equation.

It would not be unreasonable to fit a model such as

```
. eregress y x1 x2, extreat(treated) select(selected = treated x1 z4 z5)
```

If you are fitting this model, it should be obvious that the treatment arm must be assigned to the censored observations. Your selection equation says that the treatment arm itself will affect whether observations are censored.

Let's put that case aside and return to the usual case of missing treatment with exogenous treatment assignment. The ERM commands fit the model without problem. `estat teffects` will report the ATE. `estat teffects` has options for reporting other statistics, all of which will be fine except ATET—the average treatment effect among the treated. ATET is defined to include all treated observations. Because `treated` is sometimes missing because of selection, the computed ATET will exclude those observations for which treatment assignment is missing.

Now, let's consider endogenous treatment assignment. We want to fit the model

```
. eregress y x1 x2, entreat(treated = x1 z3) select(selected = x1 z4 z5)
```

What makes us hesitate is that some of or all the censored observations have `treated` equal to missing, meaning that treatment was evidently not assigned for them. If we typed the command and fit the model, ERM would fit it omitting those observations. This is equivalent to assuming that the observations were censored completely at random. That could be reasonable. Perhaps most of the censored observations were lost to follow-up—for them, the treatment arm is observed—and only a few were lost before treatment was assigned because of misplaced paperwork.

On the other hand, if all the censored observations were censored before the treatment arm was assigned, the model cannot be fit. Omitting those with missing treatment omits the censored observations, and there is simply no selection equation left to fit. After dropping the observations containing missing values, everyone left in the estimation sample is not censored.

The bottom line is that ERMs cannot fit this model. ERMs place selection after treatment assignment because lost to follow-up is the common case.

You might be able to salvage the situation. Is it just that the treatment-arm values are not in your dataset because the data were not entered? If so, retrieve the data. If that is not the case but the experiment is still ongoing, run the censored observations through the treatment-assignment process.

## Treatment statistics reported by `estat teffects`

`estat teffects` reports ATEs, which are the average effects of the treatment if it had been applied to the entire sample or to the entire underlying population. `estat teffects` reports its value and standard error. The standard error is for the sample if `vce(robust)` was not specified when the model was fit and for the population if `vce(robust)` was specified when the model was fit.

Sometimes ERMs assume `vce(robust)` even when you do not type it. This happens when you specify features that themselves require `vce(robust)`. Option `vce(cluster)` requires `vce(robust)`. Actually, it is a variation on `vce(robust)`, but that is not important for this problem. If you specify `pweights`, then `vce(robust)` is used too.

The types of standard errors reported will be clearly labeled on the output `eregress`, `eintreg`, `eprobit`, or `eoprobit` produces. `estat teffects` indicates clearly in its output, too, whether output is for the sample or the population.

`estat teffects` reports

ATE, the average treatment effect for each treatment for the entire sample/population.

`estat teffects`, `atet` reports

ATET, the average treatment effect for each treatment for the treated sample or population.

`estat teffects`, `pomeans` reports

POMEANS, the potential-outcome means for each treatment arm, meaning means for the untreated and means for each of the treated.

Outcomes here are the values of the dependent variable in the main equation, or  $y$ .

Potential outcomes are the values, observation by observation, of  $y_i$  that would be observed if each was treated and untreated.

Potential-outcome means are the means of treated and, separately, of untreated. The difference between them is the ATE.

All of these statistics can also be reported for subsamples or subpopulations.

## Also see

[\[ERM\] intro 8](#) — Conceptual introduction via worked example

## Description

After you fit a model using one of the ERM commands, you can interpret the coefficients in the usual way. You can also use `margins` to produce counterfactuals, but you must specify one of the options `predict(base())` or `predict(fix())` on the `margins` command.

In this entry, we discuss how to interpret coefficients, how to use `margins`, and how to use `predict`.

## Remarks and examples

Remarks are presented under the following headings:

- [The problem of endogenous covariates](#)
- [How to interpret coefficients](#)
- [How to use margins](#)
- [How to use margins in models without endogenous covariates](#)
- [The two ways to use margins with endogenous covariates](#)
- [Margins with predict\(base\(\)\)](#)
- [Margins with predict\(fix\(\)\)](#)
- [When to use which](#)
- [How to use margins with predict\(base\(\)\)](#)
- [How to use margins with predict\(fix\(\)\)](#)
- [How to use predict](#)

## The problem of endogenous covariates

Endogenous covariates in the main equation cause problems, which means that if your model has no endogenous covariates in the main equation, you have no problems. The following models have no endogenous covariates in the main equation:

```

eregress y x1 x2
eregress y x1 x2 c.x1#c.x2
eregress y x1 x2, select(selected = x1 z1 z2)    ///
               endogenous(z2 = z3 z4, nomain)

eprobit y x1 x2
eprobit y x1 x2 c.x1#c.x2
eprobit y x1 x2, select(selected = x1 z1 z2)    ///
               endogenous(z2 = z3 z4, nomain)

```

We showed examples with `eregress` and `eprobit`. We could just as well have shown examples with `eintreg` and `eoprobit`. Note that the last model for each command we showed has an endogenous covariate, but it is *not* in the main equation.

In any case, if you have no endogenous covariates in the main equation, you interpret coefficients and use `margins` and `predict` just as you usually would.

In the rest of the manual entry, when we write about models with or without endogenous covariates, we mean models with or without endogenous covariates in the main equation.

Models with endogenous covariates in the main equation have issues of interpretation that arise even if you fit a model as simple as

```
eregress y x1, endogenous(x1 = z1, nomain)
```

There are four ways endogenous covariates can end up in the main equation:

1. You specify `endogenous(x1 = ...)` to add variable `x1` to the main equation.
2. You specify `endogenous(x1 = ..., nomain)` and you include `x1` in the main equation.
3. You specify `entreat(treated = ...)` to handle endogenous treatment effects. `entreat()` itself adds endogenous covariate `treated` to the main equation.
4. You specify `select(selected = ...)` to handle endogenous selection and you include `selected` in the main equation. `select()` makes variable `selected` endogenous, but it does not automatically add it to the main equation.

In what follows, we will show examples of endogenous covariates added to the main equation by option `endogenous()`, but we could have added them in any of the above ways.

In this manual entry, we depart from our usual practice of naming exogenous covariates `x1`, `x2`, ... and naming endogenous covariates `w1`, `w2`, .... We depart from this practice because we will introduce a situation and then say, “if `x1` is exogenous, do this; if it is endogenous, do something else”.

## How to interpret coefficients

The rules for interpreting the coefficients from ERM models are as follows:

---

### Rule C1 concerning models fit by `eregress` and `eintreg`.

Interpret the coefficients in the main equation in the usual way. As is usual, it does not matter whether coefficients are for endogenous or exogenous covariates.

### Rule C2 concerning models fit by `eprobit` and `eoprobit` that have no endogenous covariates.

Interpret the coefficients in the main equation in the usual way. They are in standard deviation units just as they usually are. You can calculate probabilities in the usual way, too, by using the unit-normal distribution.

### Rule C3 concerning models fit by `eprobit` and `eoprobit` that include endogenous covariates.

You can interpret the coefficients in the usual way—they are in standard deviation units—but you cannot calculate probabilities because the normal distribution used does not have standard deviation 1. Worse, the value of the standard deviation sometimes depends on the questions being asked. The safest approach is to use `margins`.

---

For rule C1, it does not matter whether you fit simple models such as

```
. eregress y x1 x2
```

or complicated ones such as

```
. eregress y x1 x2, endogenous(x1 = z1, nomain)
. eintreg y1 y2 x1 x2, endogenous(x1 = x2 z1, nomain)   ///
                        endogenous(z1 = x2 z2, nomain)   ///
                        select(selected = x2 z3 z4)
```

The coefficients on `x1` and `x2` in the main equation will have the usual interpretation. That usual interpretation is that they are the expected change in `y` for a one-unit change in `x1` or `x2`.

Rule C2 is not surprising at first glance. It states that `eprobit` and `eoprobit` models without endogenous covariates are ordinary probit and ordered probit models, and so their coefficients are interpreted in the ordinary way. However, the rule says more than that. It says that if you fit

```
. eprobit y x1 x2, select(selected = x2 z1)
```

or

```
. eoprobit y x1 x2, select(selected = x2 z1)
```

then you can interpret the fitted coefficients in the usual way too.

Rule C3 is a disappointment. It says that if you fit `eprobit` or `eoprobit` models with endogenous covariates in the main equation, such as

```
. eprobit y x1 x2, endogenous(x1 = z1, nomain)
```

then you must be careful in interpreting the coefficients. You can interpret them in almost the usual way because they are in standard-deviation units, but they are not in units for which the standard deviation is 1. In the above model, if the estimate of the coefficient on  $x_1$ ,  $\hat{\beta}_1$ , is 2 and  $x_1$  has an average value of 1, then  $x_1$  has a big effect. It is even bigger than usual because the standard deviation of the normal is less than 1. How much bigger? Well, that depends, which we admit is hardly a satisfactory answer.

If the coefficients on  $x_1$  and  $x_2$ ,  $\hat{\beta}_1$  and  $\hat{\beta}_2$ , are almost equal, and if  $x_1$  and  $x_2$  are on roughly the same scale, then the effects of  $x_1$  and  $x_2$  are roughly the same, and that conclusion does not depend on the standard deviation of the normal distribution being used.

More helpfully, rule C3 suggests that you use the `margins` command.

## How to use margins

We warn you that two of the rules for using `margins`—rules M3 and M4—are a mouthful. Think of them as a reminder about the issues rather than an explanation of them. We will explain them. In any case, the rules are as follows:

---

### Rule M1 concerning models with no endogenous covariates.

`margins` can be used just as you ordinarily would use it.

### Rule M2 concerning models with endogenous covariates.

You must specify one of the `margins` options `predict(base())` or `predict(fix())`. Do that, and `margins` produces the results you expect. If you do not specify one of the options, results will be based on reduced-form predictions, which are not what you want.

### Rule M3 concerning models with endogenous covariates.

Often, you will want to specify `predict(base())`. This produces levels and comparisons (averages and differences) conditional on observing all the covariates in your model. These include the covariates outside of the main equation. The averages and differences that `margins` reports will be the best predictions possible for subjects with the same characteristics as the subjects in your data.

### Rule M4 concerning models with endogenous covariates.

Sometimes, you will need to specify option `predict(fix())`. This produces levels and comparisons (averages and differences) conditional on observing only the exogenous covariates in the main equation, and with the endogenous covariates set to the values specified. The averages and differences that `margins` reports will be the best predictions possible for subjects with the same limited set of characteristics as the subjects in your data.

---

How to use margins in models without endogenous covariates

Rule M1 was reassuring. It says that if your models include no endogenous covariates in the main equation, you can use margins in the ordinary way. Here is how you would ordinarily use `margins`. The following model has no endogenous covariates:

```
. use http://www.stata-press.com/data/r15/ermexample
(Artificial ERM example data)
. eregress y x1 x2 c.x1#c.x2
(output omitted)
```

The model fit is

$$y_i = \beta_0 + \beta_1x1_i + \beta_2x2_i + \beta_3x1_ix2_i + e_i.y$$

Assume that our interest is in the effect of `x1`. One way to interpret the effect is to interpret the coefficients: A one-unit increase in `x1` increases `y` by  $\beta_1 + \beta_3x2$ . Another way to interpret the effect is by using counterfactuals. In these data, what would be the average change in `y` if `x1` were increased by 1? `margins` will tell us if we type

```
. margins, at(x1=generate(x1)) at(x1=generate(x1+1)) contrast(at(r) nowald)
Contrasts of predictive margins
Model VCE      : OIM
Expression     : mean of y, predict()
1._at          : x1              = x1
2._at          : x1              = x1+1
```

	Delta-method			
	Contrast	Std. Err.	[95% Conf. Interval]	
_at (2 vs 1)	1.109641	.1750625	.7665246	1.452757

You can learn about `margins`, its features, and its syntax in [\[R\] margins](#). We will tell you enough, however, so that everything we say will make sense.

Assume that the data comprise three subgroups in which we have a special interest. For instance, we want to know how an increase in `x1` would affect each subgroup. `margins` can tell us that too.

```

. margins, at(x1=generate(x1)) at(x1=generate(x1+1)) contrast(at(r) nowald)
> over(group)

Contrasts of predictive margins
Model VCE      : OIM
Expression    : mean of y, predict()
over          : group
1._at         : 0.group
                x1              = x1
                1.group
                x1              = x1
                2.group
                x1              = x1
2._at         : 0.group
                x1              = x1+1
                1.group
                x1              = x1+1
                2.group
                x1              = x1+1

```

	Delta-method			
	Contrast	Std. Err.	[95% Conf. Interval]	
_at@group				
(2 vs 1) 0	.5561469	.1960937	.1718103	.9404835
(2 vs 1) 1	1.123401	.1754062	.7796108	1.46719
(2 vs 1) 2	1.641114	.2153742	1.218988	2.063239

`margins` helps us to understand changes that are different in each observation. If we had the simple model `eregress y x1 x2`, we know the effect of incrementing `x1` is to increase `y` by  $\hat{\beta}_1$ , which might be 3. The change would be 3 in every observation. In the model we have, however, the effect of incrementing `x1` is to increase `y` by  $\beta_1 + \beta_3 x_2$ . The average effect depends on the distribution of `x2`.

`margins` helps us to understand how a change affects the average in our data and subgroups of our data. We are using our sample as a proxy for the population and subpopulations, but that is what we usually do in statistics. We assume that our sample is representative. The issues are the same as we discussed in [ERM] [intro 5](#).

If our sample is representative but we want `margins` to report population-based standard errors, we need to specify `vce(robust)` when we fit the model:

```
. eregress y x1 x2 c.x1#c.x2, vce(robust)
```

If our sample is not representative, we can weight it with the inverse probability that its observations were sampled from the underlying population. If we want `margins` to report population-based standard errors, we can type

```
. eregress y x1 x2 c.x1#c.x2 [pw = weight], vce(robust)
```

or type

```
. eregress y x1 x2 c.x1#c.x2 [pw = weight]
```

We can type either because specifying `[pw=weight]` implies `vce(robust)`.

Even when we do specify or imply `vce(robust)`, `margins` will report sample standard errors by default. To obtain population-based standard errors, we must specify or imply `vce(robust)` when we fit the model and, when we use `margins`, we must specify its `vce(unconditional)` option:



```
. margins, at(x1=generate(x1)) at(x1=generate(x1+1)) contrast(at(r)) ///
    vce(unconditional)
```

In the linear regression example we have been discussing, we included an interaction in the model and used `margins` to report averages. We used `margins` because the interaction caused changes to vary observation by observation. Probit and ordered probit models produce predictions that vary observation by observation even in models with no interactions. Consider the following probit model, which is almost the simplest one possible:

```
. eprobit y_p x1
```

The model is

$$\Pr(\text{positive outcome}) = \Pr(\beta_0 + \beta_1 x1_i + e_i.y > 0) = \text{normal}(\beta_0 + \beta_1 x1)$$

Assume that our interest is in `x1` just as it was previously. The effect of a one-unit increase in `x1` is to increase the normal index by  $\hat{\beta}_1$ . Simple, right? No, it is not. The effect in probabilities of that change varies observation by observation. Here is how the results vary if  $\hat{\beta}_1$  were 0.5 and we incremented `x1` by 1. The effect depends on each subject's initial probability of a positive outcome:

Subject's original Pr(pos. outcome)	Increment by	Subject's new Pr(pos. outcome)	Difference
0.01	0.5 s.d.	0.03	0.02
0.10	0.5 s.d.	0.22	0.12
0.20	0.5 s.d.	0.37	0.17
0.40	0.5 s.d.	0.60	0.20
0.50	0.5 s.d.	0.69	0.19
0.60	0.5 s.d.	0.77	0.17
0.90	0.5 s.d.	0.96	0.06
0.99	0.5 s.d.	1.00	0.01

A subject whose original probability was 0.40 experiences an increase of 0.20 when `x1` is incremented by 1. Meanwhile, a subject whose probability was 0.90 experiences a mere 0.06 increase.

Using `margins`, we can obtain the average changes in probabilities in the data due to incrementing `x1` by 1. We type

```
. margins, at(x1=generate(x1)) at(x1=generate(x1+1)) contrast(at(r) nowald)
Contrasts of adjusted predictions
Model VCE      : OIM
Expression    : Pr(y_p==1), predict()
1._at         : x1              = x1
2._at         : x1              = x1+1
```

	Delta-method		
	Contrast	Std. Err.	[95% Conf. Interval]
_at (2 vs 1)	.2961685	.0287644	.2397912 .3525458

We can obtain the changes for each of the three subgroups too:

```
. margins, at(x1=generate(x1)) at(x1=generate(x1+1)) contrast(at(r) nowald)
> over(group)

Contrasts of adjusted predictions
Model VCE      : OIM
Expression     : Pr(y_p==1), predict()
over           : group
1._at          : 0.group
                  x1              = x1
                  1.group
                  x1              = x1
                  2.group
                  x1              = x1
2._at          : 0.group
                  x1              = x1+1
                  1.group
                  x1              = x1+1
                  2.group
                  x1              = x1+1
```

	Delta-method			
	Contrast	Std. Err.	[95% Conf. Interval]	
_at@group				
(2 vs 1) 0	.3857775	.051078	.2856664	.4858885
(2 vs 1) 1	.2944176	.0294406	.2367152	.3521201
(2 vs 1) 2	.2096478	.0202614	.1699363	.2493594

Counterfactuals are useful in complicated linear models—we had an interaction in ours—and in nonlinear models whether simple or complicated.

### The two ways to use margins with endogenous covariates

You may remember that rules M3 and M4 were mouthfuls. These rules said to use `margins` with the `predict(base())` option in one case and `predict(fix())` in another. Moreover, each option was “best”, albeit under different assumptions. We apologize for that. We can make the distinction clear in a reasonably simple model, namely

```
. eregress y x1 x2, endogenous(x1 = z1, nomain)
```

The model is

$$y_i = \beta_0 + \beta_1 x1_i + \beta_2 x2_i + e_i.y$$

$$x1_i = \gamma_0 + \gamma_1 z1_i + e_i.x1$$

where  $\rho = \text{corr}(e.x1, e.y)$  and is nonzero.

Let’s imagine that `y` is a health outcome and `x1` is a 0/1 variable indicating whether a treatment was administered that is expected to improve the outcome. Observations are people, and people choose for themselves whether to have the treatment. Given the story, we *should* fit the model by typing

```
. eregress y i.x1 x2, endogenous(x1 = z1, probit nomain)
```

Nonetheless, we are going to fit the model without the `probit` specification and factor-variable notation for endogenous covariate `x1`:

```
. eregress y x1 x2, endogenous(x1 = z1, nomain)
```

We omit `probit` only because it will be easier for us to explain the difference between `predict(base())` and `predict(fix())`. We need to show you some math, and the math will be simpler in the linear model case.

What is important is that  $\rho$  is likely to be nonzero, no matter how the model is fit.  $\rho$  is the correlation between  $e.y$  and  $e.x1$ .  $e.y$  includes all the unobserved things that affect how well the treatment works.  $e.x1$  includes all the unobserved things that affect whether individuals choose the treatment.  $\rho$  is likely to be nonzero and positive because people who believe that they are more likely to benefit from the treatment ( $e.y > 0$ ) should be more likely to choose the treatment ( $e.x1 > 0$ ).

As a result, the best prediction of  $y$  that we can make for people like person 1 in our data—people who have the same value of  $x1$ ,  $x2$ , and  $z1$ —includes the effect of  $\hat{\rho}$ , albeit indirectly. The best prediction of  $y$  we can make for people like person 1 is that their expected value of  $y$  will be

$$\hat{y}_1 = \hat{\beta}_0 + \hat{\beta}_1 x1_1 + \hat{\beta}_2 x2_1 + \hat{e}_1.y$$

$\hat{e}_1.y$  is our estimate of the expected value of  $e.y$  in the first observation. Expected values of errors are often 0, but not in this case. This one depends on  $\rho$ . Given that we know the values  $x1_1$  and  $z1_1$ , we have an estimate of  $e_1.x1$ , namely

$$\hat{e}_1.x1 = x1_1 - \hat{\gamma}_0 - \hat{\gamma}_1 z1_1$$

Because  $e.x1$  and  $e.y$  are correlated, we can produce an estimate of  $e_1.y$  given  $\hat{e}_1.x1$  and  $\hat{\rho}$ . It is a detail, but the formula is

$$\hat{e}_1.y = \frac{\rho \times \text{s.d.}(e.y)}{\text{s.d.}(e.x1)} \times \hat{e}_1.x1$$

The value of  $\hat{e}_1.y$  can be calculated, and the best prediction we can make for people like person 1 includes it, and is

$$\hat{y}_1 = \hat{\beta}_0 + \hat{\beta}_1 x1_1 + \hat{\beta}_2 x2_1 + \hat{e}_1.y$$

## Margins with `predict(base())`

Here, we temporarily consider  $x1$  to be continuous because we want to consider what happens if we add 1 to  $x1$ .

What is the best prediction we can make for people like person 1 if  $x1$  were incremented by 1? It is

$$\hat{y}_1 = \hat{\beta}_0 + \hat{\beta}_1(x1_1 + 1) + \hat{\beta}_2 x2_1 + \hat{e}_1.y$$

The above is how `margins` with option `predict(base())` makes the calculation for each observation in the data. Observation by observation, it calculates

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1(x1_i + 1) + \hat{\beta}_2 x2_i + \hat{e}_i.y \quad (1)$$

`predict(base())` tells `margins` to include  $\hat{e}_i.y$  in the calculations. This is the best prediction for people like the people in our population conditioned on everything we know about them.

Now, we return to considering  $x1$  to be binary.

## Margins with predict(fix())

`predict(base())` uses (1) and makes its predictions given how the world currently operates. People choose their value of  $x_1$ , and the choice they make is correlated with the outcomes they expect.

`predict(fix())` makes predictions for a world that operates differently. In the alternative world,  $x_1$  is fixed at a value such as 1. This means that the population of people like person 1 is expanded from being all people like person 1 who made the same treatment choice to being all people like person 1 regardless of the treatment choice they made. In the expanded definition of people like person 1, the correlation between  $e.y$  and  $e.x_1$  is broken. The correlation is now 0, and the best prediction for people like person 1 sets  $\hat{e}_1.y$  to 0:

$$\hat{y}_1 = \hat{\beta}_0 + \hat{\beta}_1 x_{11} + \hat{\beta}_2 x_{21} \quad (2)$$

In the jargon of statistics,  $x_1$  is no longer endogenous—it is fixed, and the entire equation for  $x_1$  becomes irrelevant.

When you specify `predict(fix())`, `margins()` makes the calculation for each person by using the approach used for person 1 in (2). It uses

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{2i} + \hat{\beta}_2 x_{2i}$$

These observation-by-observation predictions are called potential outcomes when applied to treatment models. The averages based on them that `margins` reports are called potential-outcome means (POMs). These averages correspond to what would be observed in a world in which  $x_1$  is fixed at a particular value.

We considered  $x_1$  being fixed at a constant value.  $x_1$  can just as well be fixed at different values for different observations.

## When to use which

`margins` can produce counterfactuals in two ways.

When you specify `predict(base())`, `margins` uses

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i} + \hat{e}_i.y$$

for the values of  $x_1$  and  $x_2$  specified. The predictions are a function of  $x_1$  and  $x_2$  and the covariates appearing in the  $x_1$  equation. Those covariates along with  $\hat{\rho}$  go into the calculation of  $\hat{e}_i.y$ . These predictions correspond to how the current world operates.

When you specify `predict(fix())`, `margins` uses

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i}$$

where  $x_1$  is fixed at the value specified. These predictions are based on the exogenous covariates in the main equation ( $x_2$  in this case) and the value to which the fixed variable ( $x_1$ ) is set. These predictions correspond to a different world in which  $x_1$  is no longer endogenous but is fixed to a particular value.

We have shown results for linear models. The formulas are more complicated when models are nonlinear but the assumptions and their implications are the same.

## How to use margins with predict(base())

When we used margins with models in which there were no endogenous covariates, one of the comparisons we ran was

```
. eregress y x1 x2 c.x1#c.x2
. margins, at(x1=generate(x1)) at(x1=generate(x1+1)) contrast(at(r))
```

The `at()` option ran two counterfactuals, although the first was more factual than counterfactual. Margins ran the factual `at(x1=x1)`. It ran the counterfactual `at(x1=generate(x1+1))`. Had we omitted `contrast(at(r))`, margins would have reported the means of `y` under the two scenarios. Because we specified `contrast(at(r))`, margins instead reported the average value of the difference.

To produce counterfactuals based on changing `x1`, you must specify option `predict(base())` in models containing any endogenous covariates in the main equation. You must include the option even if `x1` itself is not endogenous.

Let's imagine that we fit one of the models

```
. eregress y x1 x2 c.x1#c.x2, endogenous(x1 = z1, nomain)
. eregress y x1 x2 c.x1#c.x2, endogenous(x2 = z1, nomain)
. eregress y x1 x2 c.x1#c.x2, endogenous(x1 = z1, nomain) ///
  endogenous(x2 = z2, nomain)
```

and now we want to produce the same counterfactual we produced when the model had no endogenous covariate, that is, when we typed

```
. margins, at(x1=generate(x1)) at(x1=generate(x1+1)) contrast(at(r))
```

To produce the same counterfactual, we type

```
. generate x1orig = x1
. margins, at(x1=generate(x1)) at(x1=generate(x1+1)) contrast(at(r)) ///
  predict(base(x1=x1orig))
```

We did two things differently:

1. We created a new variable containing a copy of `x1`.
2. We added `predict(base(x1=x1orig))` to the margins command, which includes the copied variable.

If we wanted a comparison of `x1+1` with `x1+2`, we would type

```
. generate x1orig = x1
. margins, at(x1=generate(x1+1)) at(x1=generate(x1+2)) contrast(at(r)) ///
  predict(base(x1=x1orig))
```

If we requested counterfactuals that involved changing `x1` and `x2`, we would type

```
. generate x1orig = x1
. generate x2orig = x2
. margins, at(x1=generate(x1) x2=generate(x2)) ///
  at(x1=generate(x1+1) x2=generate(x2+1)) ///
  contrast(at(r)) predict(base(x1=x1orig x2=x2orig))
```

That is,

1. You must copy all variables that will be temporarily changed by margins.
2. You must specify the name of each original variable and the name of each copied variable in the `predict(base(original=copied))` option.

The variables that `margins` changes appear in its `at()` option. They can also appear in *varlist* following the `margins` command, such as

```
. eregress y x1 i.x2, endogenous(x1 = z1)
. generate x2 = x2orig
. margins r.x2, predict(base(x2=x2orig))
. drop x2orig
```

`margins r.x2` compares average values of `y` for each level `x2`. It reports them as differences in average values from the first level.

For examples using `margins` with `predict(base())`, see *Interpreting effects* in [ERM] intro 8 and see [ERM] example 1a.

## How to use margins with predict(fix())

You have fit the model

```
. eregress y i.x1 x2, endogenous(x1 = z1, probit nomain)
```

This is the same example we discussed in *The two ways to use margins with endogenous covariates* except that now we specify the equation for `x1` as a probit equation.

To run the counterfactuals that `x1` is fixed at 0 and fixed at 1, type

```
. margins, predict(fix(x1)) at(x1=0 x1=1)
```

Averages of `y` will be reported based on predictions of `y` given the values of the exogenous covariates in the main equation (`x2` in this case) holding the fixed variable (`x1`) fixed first at 0 and then at 1.

If the model had two endogenous covariates in the main equation,

```
. eregress y x1 x2 x3, endogenous(x1 = z1, probit nomain) ///
                        endogenous(x2 = z2 z3, probit nomain) ///
                        endogenous(z3 = z4, nomain)
```

you could fix both of them by typing

```
. margins, predict(fix(x1 x2)) at(x1=1 x2=0)
```

The average of `y` will be reported given all the values of the exogenous covariates in the main equation (`x3` in this case) holding `x1` and `x2` fixed at the values specified.

You could fix `x1` only:

```
. margins, predict(fix(x1)) at(x1=1)
```

The average of `y` will be reported given

- the values of all the exogenous covariates in the main equation (`x3` in this case), plus
- the values of `x2` and of all the covariates in its equation, whether endogenous or exogenous (`x2`, `z2`, and `z3` in this case), plus
- all the covariates necessary to predict `x2`'s endogenous covariates (`z4` in this case).

Had `z4` been endogenous and had an equation, we would have added that equation's variables, and so on.

In this case, the average of `y` will be reported given `x3`, `x2`, `z2`, `z3`, and `z4`. `x1` will be fixed.

If the model had been

```
. eregress y x1 x2 x3, endogenous(x1 = z1, probit nomain)      ///  
                        endogenous(x2 = x1 z2, probit nomain)
```

then `margins` would refuse to fix just `x1`:

```
. margins, predict(fix(x1)) at(x1=1)  
endogenous x2 depends on fixed x1  
r(498);
```

We tried to fix `x1` and `x1` also affects `x2`. `margins, predict(fix())` cannot do this.

You could, however, fix `x2` because `x2` does not affect `x1`:

```
. margins, predict(fix(x2)) at(x2=1)
```

For examples using `margins` with `predict(fix())`, see [Interpreting effects](#) in [ERM] **intro 8** and see [ERM] **example 1a**.

## How to use `predict`

Regardless of how or why a model was fit, Stata's postestimation `predict` command is used in three ways:

### **In-sample prediction.**

`predict` is used to obtain predicted values from the data used to fit the model.

### **Out-of-sample prediction.**

`predict` is used to obtain predicted values from other data, data not used to fit the model.

### **Counterfactual prediction.**

`predict` is used to obtain what the predicted values would be if the values of a covariate or covariates were changed. Counterfactual prediction can be performed with the data used to fit the model or other data.

The rules for using `predict` after ERM depend on the way `predict` is being used. The rules are the following:

---

### **Rule P1 for models with no endogenous covariates.**

`predict` is used for in-sample, out-of-sample, and counterfactual prediction just as you would ordinarily use it. This rule applies to all models with no endogenous covariates in the main equation.

### **Rule P2 for models with endogenous covariates.**

`predict` is used for in-sample and out-of-sample prediction just as you would ordinarily use it.

### **Rule P3 for models with endogenous covariates.**

You must specify option `base()` or `fix()` when using `predict` for counterfactual prediction.

---

Here is how `predict` is ordinarily used. You have fit the model

```
. eregress y x1 x2
```

The model you fit could just as well be fit by `eintreg`, `eprobit`, or `eoprobit`.

To make in-sample predictions, you type (with the same dataset in memory)

```
. predict yhat
```

New variable `yhat` will contain the predicted values based on the fitted model.

To make out-of-sample predictions, you type

```
. use anotherdataset
. predict yhat
```

New variable `yhat` will contain predicted values based on the fitted model.

You could also use one part of the original dataset to fit the model and make predictions simultaneously both in and outside of the data used to fit the model:

```
. eregress y x1 x2 if subset==1
. predict yhat
```

New variable `yhat` would contain in-sample predictions in observations for which `subset==1` and would contain out-of-sample predictions in the other observations.

You can make counterfactual predictions. You have fit the model

```
. eregress y x1 x2
```

To obtain predicted values of `y` if `x1` were 1 in all observations, you type

```
. eregress y x1 x2
. replace x1 = 1
. predict yhat1
```

New variable `yhat1` will contain predicted values conditional on `x1` being 1.

You use `predict` in models with endogenous covariates in the main equation just as we have shown when making in-sample or out-of-sample predictions. To make counterfactual predictions in models with endogenous covariates in the main equation, such as

```
. eregress y = x1 x2, endogenous(x1 = z1, nomain)
```

you type

```
. generate x1orig = x1
. replace x1 = 1
. predict yhat1, base(x1=x1orig)
. replace x1 = x1orig
. drop x1orig
```

or you type

```
. generate x1orig = x1
. replace x1 = 1
. predict yhat1, fix(x1)
. replace x1 = x1orig
. drop x1orig
```

You must specify option `base()` or `fix()` with `predict` for the same reasons you must specify option `predict(base())` or `predict(fix())` with `margins`. You make the decision about which to specify in the same way you make the decision with `margins`.

Note that `predict` used for making counterfactual predictions works just like `predict` used for making in-sample or out-of-sample predictions in one respect. `predict` uses the values of the variables in memory. To make counterfactual predictions, you must change the contents of those variables.

Using `predict` to predict counterfactuals reproduces results produced by `margins`. Typing

```
. generate x1orig = x1
. margins, predict(base(x1=x1orig)) at(x1=generate(x1+1))
```



is equivalent to typing

```
. generate x1orig = x1
. replace x1 = 1
. predict yhat1, base(x1=x1orig)
. summarize yhat1
```

Both will report the same result for the average of  $y$  if  $x_1$  were incremented by 1.

Typing

```
. margins, predict(fix(x1)) at(x1=1)
```

is equivalent to typing

```
. generate x1orig = x1
. replace x1 = 1
. predict yhat1, fix(x1)
. summarize yhat1
```

For your information, `margins` uses `predict` in making its calculations, and that explains why the options on `margins` are named `predict(base())` and `predict(fix())`. When `margins` uses `predict`, it specifies to predict the options you specified in option `predict()`.

## Also see

[\[ERM\] intro 8](#) — Conceptual introduction via worked example

[\[ERM\] example 1a](#) — Linear regression with continuous endogenous covariate

Description

If you already are familiar with some or all of Stata’s other commands that fit models with endogenous covariates, sample selection, and treatment effects, this entry shows you how to use that knowledge to fit equivalent models using ERMs.

Remarks and examples

Aside from providing a single coherent framework that allows complications to be combined, ERMs use similar syntax and the resulting models have the same interpretation.

In most cases, the estimation method used by the ERM commands and that used by other estimators to fit the same model produce results that are the same. Typically, there are small numerical differences because the optimization is different. Also, ancillary parameters, such as variances of errors, are sometimes parameterized differently. In some cases, a different estimation method is used. In this case, results will be asymptotically equivalent, but in finite samples, results will differ.

The table below provides a basic guide for the correspondence between Stata commands you may already be familiar with and the ERM commands.

Command you know	Equivalent extended regression command
Linear regression with endogenous covariate ivregress liml y1 x (y2 = z)	eregress y1 x, endogenous(y2 = z x)
Probit model with endogenous covariate ivprobit y1 x (y2 = z)	eprobit y1 x, endogenous(y2 = z x)
Tobit model with endogenous covariate ivtobit y1 x (y2 = z), ll(0) ul(20)	generate y1_ll = y1 replace y1_ll = . if y1<=0 generate y1_ul = y1 replace y1_ul = . if y1>=20 & y1<. eintreg y1_ll y1_ul x, endogenous(y2 = z x)
Linear regression with exogenous treatment teffects ra (y x1 x2) (t1)	eregress y x1 x2, extreat(t1) vce(robust) estat teffects
Probit model with exogenous treatment teffects ra (y x1 x2, probit) (t1)	eprobit y x1 x2, extreat(t1) vce(robust) estat teffects

Linear regression with endogenous treatment

```
ereregress y x, treat(t1 = x z)                                ereregress y x, entreat(t1 = x z, nointeract)
```

Linear regression with sample selection

```
heckman y x, select(s1 x z)                                ereregress y x, select(s1 = x z)
```

Probit model with sample selection

```
heckprobit y x, select(s1 x z)                                eprobit y x, select(s1 = x z)
```

Ordered probit model with sample selection

```
heckoprobit y x, select(s1 x z)                                eoprobit y x, select(s1 = x z)
```

---

You can build on the basic syntax of the ERM commands by combining options and suboptions, giving you the flexibility to fit a myriad of models. Here is a short list of what you might try.

Linear regression with a continuous endogenous covariate but where the exogenous variable is not included as an instrument

```
. ereregress y1 x, endogenous(y2 = z1)
```

Linear regression with two continuous endogenous covariates

```
. ereregress y1 x, endogenous(y2 y3 = z1 x)
```

As above, but with different instruments for different endogenous covariates

```
. ereregress y1 x, endogenous(y2 = z1 x) endogenous(y3 = z2 x)
```

As above, but with one endogenous covariate being binary

```
. ereregress y1 x, endogenous(y2 = z1 x) endogenous(y4 = z3 x, probit)
```

Linear regression with a continuous endogenous covariate and an endogenous treatment

```
. ereregress y1 x, endogenous(y2 = z1 x) entreat(t1 = z3 x)
```

As above, but instead include a multivalued treatment

```
. ereregress y1 x, endogenous(y2 = z1 x) entreat(t2 = z3 x, oprobit)
```

As above, and also allow for endogenous selection

```
. ereregress y1 x, endogenous(y2 = z1 x) entreat(t2 = z3 x, oprobit) ///  
    select(s1 = w x)
```

As above, but where censoring of variable s2 indicates selection status

```
. ereregress y1 x, endogenous(y2 = z1 x) entreat(t2 = z3 x, oprobit) ///  
    tobitselect(s2 = w x)
```

`eprobit` or `eoprobit` may be directly substituted for `ereregress` above to fit a probit or ordered probit regression when `y1` is binary or ordinal. To fit a tobit or interval regression, you must use `eintreg` and specify two dependent variables containing the upper and lower bounds of the interval in place of `y1`.

## Also see

[ERM] **intro 1** — An introduction to the ERM commands

[ERM] **intro 8** — Conceptual introduction via worked example

## Description

This entry introduces the concepts of endogenous covariates, nonrandom treatment assignment, and endogenous sample selection through a series of examples. It also provides an overview of how to interpret the results of ERMs.

## Remarks and examples

Remarks are presented under the following headings:

*[Introduction](#)*

*[Complications](#)*

*[Endogenous covariates](#)*

*[Nonrandom treatment assignment](#)*

*[Endogenous sample selection](#)*

*[Interpreting effects](#)*

## Introduction

In a perfect research world, several assumptions we conventionally make about our data and the data-collection process would be true. For example, we could gather data about all the variables that influence the outcome we want to study. These data would be collected on a random sample of the population of interest. Any inferences we made about a relationship between the dependent variable and an independent variable when studying one group would be just as valid if we studied this group again at a different time or even if we conducted the study for a different group.

Often, applied research is complicated when one or more of the classical assumptions are not true. For example, data on key variables of interest may be unavailable. Our interest may lie in a treatment that cannot be randomly assigned or may be endogenous. Or the subjects we have available to study are not representative of the population we want to study.

When any of these things is true, we cannot make accurate inferences using standard regression methods. Stata provides many commands that can be used when one of these complications occurs. The ERM commands allow you to address these problems in isolation and, more importantly, in combination—as they often occur.

Imagine that a large company is considering offering a workplace wellness program to its employees to help them lose weight. They have conducted a pilot study at one location, and all other locations are expected to be similar. In our dataset, the `wellpgm` variable records whether a given employee participated. After one year, the company wants to know whether the program was effective. Our outcome of interest is weight lost in kilograms. We have called this `weightloss0` to distinguish it from the observed `weightloss` later.

In our fictional data, the number of kilograms lost is also determined by the employee's age in years (`age`), the employee's sex (`sex`), and the employee's starting weight in kilograms (`weight`). Because this is an entirely fictitious example, we have a true measure of willingness to engage in healthy behaviors (`health`).

More formally, in our simulated data, the process that determines weight lost is

$$\begin{aligned} \text{weightloss0}_i = & -4 - 0.1 \times \text{age}_i - 1.5 \times \text{sex}_i + 0.14 \times \text{weight}_i + 1.2 \times \text{wellpgm}_i \\ & + 0.5 \times \text{health}_i + u_i \end{aligned}$$

Suppose that we are in the situation described above. We observed complete information for all variables for all employees, and participation in the wellness program was unrelated to any employee attributes that we could not observe. In this case, we could fit our model by typing

```
. use http://www.stata-press.com/data/r15/wellness
(Fictional workplace wellness data)

. regress weightloss0 age i.sex weight i.wellpgm health
```

Source	SS	df	MS	Number of obs	=	545
				F(5, 539)	=	589.61
Model	2417.76071	5	483.552141	Prob > F	=	0.0000
Residual	442.044242	539	.820119187	R-squared	=	0.8454
				Adj R-squared	=	0.8440
Total	2859.80495	544	5.25699439	Root MSE	=	.9056

weightloss0	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
age	-.0991644	.0038045	-26.06	0.000	-.1066378	-.0916909
sex						
male	-1.481883	.0937504	-15.81	0.000	-1.666044	-1.297722
weight	.1359547	.0054405	24.99	0.000	.1252676	.1466419
wellpgm						
yes	1.254928	.1076792	11.65	0.000	1.043406	1.46645
health	.4814308	.0255931	18.81	0.000	.4311564	.5317053
_cons	-3.754726	.4432054	-8.47	0.000	-4.625348	-2.884105

```
. estimates store true
```

From this model, we can estimate the average treatment effect (ATE) of the wellness program by using the coefficient on `wellpgm`. We estimate that the ATE is 1.25 kg. In other words, the average weight lost over the course of the year would be 1.25 kg greater if all the company’s employees participated in the program versus if no employees participated.

Because we simulated these data, we can confirm that all the confidence intervals contain the true values. If we continued to add more observations, our point estimates would become closer and closer to the real values. This is true because the coefficient estimates shown above are consistent. Because they are consistent, we can make inferences about the effects of each variable on the outcome. We `estimates store` these values as `true` for comparison with later models.

Complications

As discussed in [ERM] intro 3, a covariate is endogenous if it is correlated with the error term. Practically, this correlation arises for many reasons. For example, we may have omitted an important variable from our model that is correlated with a variable that we included, as we did here. Or we may not have accurately measured one of the covariates in our model. We could also have the case where a variable in the model and the outcome of interest are partially determined by the same unobserved factors. For concreteness, we focus on the role of a single omitted variable in this conceptual introduction.

Often in observational research, the treatment (participation in the wellness program) was not randomly assigned. As discussed in [ERM] intro 5, we might be able to ignore this issue if we do not suspect that unobserved factors that affect participation also affect the amount of weight loss. However, in this case, we believe participation in the wellness program is also likely to be determined by factors we cannot observe, such as the now-omitted `health` variable.

Further, suppose that the pilot study was structured such that baseline information about all employees was collected at a mandatory benefits meeting at the start of the year. At the end of the year, all employees were asked to go to the company gym during business hours to have their year-end weight recorded, regardless of program participation. Because employees were not required to have their final weight recorded, we observe only the weight of employees who voluntarily went to the gym. We have a selected sample in this case.

Whether an employee is observed in the study could be correlated with unobserved factors that also determine how much weight he or she lost. For example, employees with high values of the now-omitted `health` variable may have generally better diet and exercise habits (independent of the wellness program), leading to higher weight loss. Let's say that for bragging rights, they want to have their superior weight loss recorded, so they are more likely to show up at the end of the year. As discussed in [ERM] intro 4, if selection is related to unobserved factors that are correlated with the outcome, it cannot be ignored.

If we ignore all of these potential complications, we might erroneously fit the model below. In this model, we omit `health`, and `weightloss` records the observed weight loss only for employees who went to the gym at the end of the year.

$$\text{weightloss}_i = \beta_1 \times \text{age}_i + \beta_2 \times \text{sex}_i + \beta_3 \times \text{weight}_i + \beta_4 \times \text{wellpgm}_i + u_i$$

As before, we could fit the model using `regress`.

```
. regress weightloss age i.sex weight i.wellpgm
```

Source	SS	df	MS	Number of obs	=	337
Model	1239.17345	4	309.793362	F(4, 332)	=	219.74
Residual	468.060374	332	1.4098204	Prob > F	=	0.0000
				R-squared	=	0.7258
				Adj R-squared	=	0.7225
Total	1707.23382	336	5.08105304	Root MSE	=	1.1874

weightloss	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
age	-.0800928	.0063184	-12.68	0.000	-.0925219	-.0676637
sex						
male	-1.023886	.146934	-6.97	0.000	-1.312925	-.734847
weight	.0803689	.0074025	10.86	0.000	.0658072	.0949305
wellpgm						
yes	1.913531	.1596906	11.98	0.000	1.599398	2.227664
_cons	-.3699558	.6741312	-0.55	0.584	-1.696063	.9561513

None of the confidence intervals for our coefficient estimates contain the true values. We store the estimates so that we can compare them with estimates from other models later.

```
. estimates store base
```

Endogenous covariates

Continuing with our example, we suspect that `weight` is endogenous now that we cannot observe `health`. Employees who are predisposed to healthy behaviors will likely have a lower starting weight, and this could influence how much weight they lose over the course of the year-long study. If we have a suitable model for how `weight` relates to the unobserved `health`, we can still estimate the parameters consistently.

Let's suppose we believe that the employee's starting weight is a function of the employee's sex and the number of times the employee visits the company gym. We measure gym use as the employee's average number of visits per month to the company gym before the program (`gym`). This will be an instrumental variable for `weight`. Instrumental variables are exogenous covariates that are correlated with the endogenous covariate, not directly related to the outcome, and not correlated with the unobserved error. Because we are using preprogram gym use, we do not expect it to be related to weight loss during the year of the program.

We fit the model using `eregress`, storing the estimates for later comparison.

```
. eregress weightloss age i.sex i.wellpgm, endogenous(weight = i.sex gym)
(output omitted)
. estimates store endog
```

Now, we view and compare the results from each of the commands. We focus on the coefficients here because our interest lies in illustrating how the point estimates change as we address different complications. At the end of the introduction, we show the full output of `eregress` and discuss its interpretation.

```
. estimates table true base endog, stats(N) equations(1) keep(#1:)
```

Variable	true	base	endog
age	-.09916437	-.08009282	-.07964086
sex male	-1.481883	-1.023886	-1.6411717
weight	.13595472	.08036889	.14701973
wellpgm yes	1.2549281	1.9135311	1.9008534
health _cons	.48143082 -3.7547263	-.36995584	-5.5172377
N	545	337	337

Once we account for the endogeneity of `weight`, the coefficients for `sex` and `weight` are close to those of the `true` model and have the correct signs. The estimates for `age` and `wellpgm`, however, are close to each other in the `base` and `endog` models but not close to the `true` values. Our estimates remain inconsistent because we have not yet addressed the endogeneity of the `wellpgm` program indicator.

Endogenous covariates in ERMs need not be continuous. We could instead have an endogenous binary or ordinal covariate. To address the endogeneity of `wellpgm`, we could include an additional model by adding another `endogenous()` option; see [\[ERM\] intro 3](#) for more on specifying models with different types of endogenous covariates. Another way to approach the analysis of binary and ordinal endogenous covariates is in the potential-outcomes framework.

## Nonrandom treatment assignment

Treatment-effect regressions model the effect of a discrete treatment or intervention on the outcome. In observational data, we cannot randomly assign a treatment of interest to individuals. Treatment status may be related to other covariates that we measure. It may even be related to the unobserved factors that affect the outcome and be endogenous. We cannot just take the sample means of the treated and untreated to estimate the ATE. Instead, we can use the potential-outcomes framework to estimate a treatment effect.

In the potential-outcomes framework, the treatment effect is the difference between the outcome that would occur when a given subject receives the treatment and the outcome that would occur when the subject receives the control instead. We only observe the potential outcome associated with that subject's observed treatment value (either treated or control). However, we can estimate both potential outcomes, conditional on covariates, by using information from the model. For more information about the potential-outcomes framework, see [\[TE\] teffects intro advanced](#).

The ERM commands may be used with an exogenous or endogenous treatment where the treatment variable is binary or ordinal.

To address the endogenous selection of participation in the wellness program, we need a model for `wellpgm`. Whether the employee was a smoker at the beginning of the year (`smoke`) is an additional covariate in our treatment model. Because smoking signals a lower willingness to engage in healthy behaviors, it should be correlated with participation in the program, but smoking status measured before the program was offered should not be independently associated with weight loss during the program.

```
. eregress weightloss age i.sex, endogenous(weight = i.sex gym)
> entreat(wellpgm = age i.smoke, nointeract)
(output omitted)
. estimates store entrtr
```

By specifying `nointeract`, we keep the same coefficients for both treatment groups in the main equation. This is not the most common approach. However, we simulated the data this way to keep the `estimates` table results compact and easy to compare across models. We will show you a more interesting model later.

Now, we view and compare the results for the main equation for each of the models.

```
. estimates table true base endog entrtr, stats(N) equations(1) keep(#1:)
```

Variable	true	base	endog	entrtr
age	-.09916437	-.08009282	-.07964086	-.10430319
sex				
male	-1.481883	-1.023886	-1.6411717	-1.5995888
weight	.13595472	.08036889	.14701973	.14151952
wellpgm				
yes	1.2549281	1.9135311	1.9008534	.83556752
health	.48143082			
_cons	-3.7547263	-.36995584	-5.5172377	-3.488841
N	545	337	337	337

In the `entrtr` model, where we have accounted for the endogeneity of starting weight and the endogenous treatment assignment to the wellness program, we estimate that the effect of participating



in the program is 0.84 kg lost. This is closer to the 1.25 kg we estimated in the `true` model than the 1.90 kg we estimated in the `endog` model that did not account for treatment assignment.

## Endogenous sample selection

Sample selection is an ambiguous term because different authors have used it to mean different things. To add more ambiguity, sample selection has been equated with nonresponse bias and selection bias in some disciplines. Much of the ambiguity arises from authors not being precise about when sample selection is ignorable.

Sample selection is like treatment assignment: a process maps each individual into or out of the sample. This process depends on observable covariates and unobservable factors. When unobservable factors that affect who is in the sample are independent of unobservable factors that affect the outcome, then the sample selection is not endogenous. In this case, the sample selection is ignorable—our estimator that ignores sample selection is still consistent.

In contrast, when the unobservable factors that affect who is included in the sample are correlated with the unobservable factors that affect the outcome, the sample selection is endogenous and it is not ignorable, because estimators that ignore endogenous sample selection are not consistent.

The ERM commands may be used with endogenous sample selection with a probit or tobit selection model. A probit selection model is used when we have a binary indicator of selection. A tobit selection model is used when we have a continuous indicator for selection.

We suspect that unobserved factors that influence whether employees came to the gym for the year-end weigh-in also influence the amount of weight lost. In other words, we believe we may have endogenous sample selection. Our `true` model included all information on all 545 employees. In reality, only 337 completed the final weigh-in for our study. However, we still want to know what the potential effect of the program was for all employees. The 0.84 kg that we estimated in *Nonrandom treatment assignment* is not a consistent estimate of the program's ATE in the company if the 337 employees in our study are not representative of the population.

By modeling the sample-selection process, we can include all 545 employees in our estimation sample. The variable `completed` indicates whether the employee completed the final weigh-in. Employees with `completed = 0` have missing values for `weightloss`. However, because all other data were gathered at a mandatory meeting at the start of the year (such as starting weight) or collected from administrative records (such as prior-year visits to the company gym), we have complete information for all other variables.

We include the employee's job classification (`salaried`) and years employed at the company (`experience`) as additional covariates in our selection model that are excluded from the main equation. `salaried` is 1 if the employee is salaried and is 0 if the employee is paid hourly. We anticipate that salaried employees will have more opportunity to visit the gym during the day and that employees who have been with the company longer will be more motivated to help complete the study. Aside from their effect on completing the weigh-in, we do not believe that `salaried` or `experience` have any direct effect on `weightloss`.

We fit our model, accounting for the potentially endogenous selection.

```
. eregress weightloss age i.sex, endogenous(weight = i.sex gym)
> entreat(wellpgm = age i.smoke, nointeract)
> select(completed = i.wellpgm experience i.salaried)
(output omitted)
. estimates store endsel
```

Then, we compare these estimates with those from our previous models.

```
. estimates table true base endog entrtr endsel, stats(N) equations(1) keep(#1:)
```

Variable	true	base	endog	entrtr	endsel
age	-.09916437	-.08009282	-.07964086	-.10430319	-.11149981
sex					
male	-1.481883	-1.023886	-1.6411717	-1.5995888	-1.5607651
weight	.13595472	.08036889	.14701973	.14151952	.14353999
wellpgm					
yes	1.2549281	1.9135311	1.9008534	.83556752	.92462755
health	.48143082				
_cons	-3.7547263	-.36995584	-5.5172377	-3.488841	-3.6798876
N	545	337	337	337	545

After accounting for the potentially endogenous selection that occurs because some employees chose not to complete the final weigh-in, we see that our estimated ATE is 0.925, which is closer to its true value than in the models that did not address selection.

## Interpreting effects

In the previous sections, we showed only the coefficient estimates from the main outcome equation. The full output for `eregress` and the other ERM commands includes estimates of coefficients of covariates in the auxiliary models, error variances, and error correlation terms.

For many models, the coefficient estimates themselves are not directly useful. You will need to use `margins` or `estat teffects` to obtain interpretable effects. However, the correlation estimates always provide relevant information.

The full results for the last `eregress` command that we estimated are as follows:

```
. eregress weightloss age i.sex, endogenous(weight = i.sex gym)
> entreat(wellpgm = age i.smoke, nointeract)
> select(completed = i.wellpgm experience i.salaried)

(iteration log omitted)
```

Extended linear regression	Number of obs	=	545
	Selected	=	337
	Nonselected	=	208
Log likelihood = -2800.8318	Wald chi2(4)	=	749.04
	Prob > chi2	=	0.0000

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
weightloss						
age	-.1114998	.0083531	-13.35	0.000	-.1278715	-.0951281
sex						
male	-1.560765	.2062746	-7.57	0.000	-1.965056	-1.156474
weight	.14354	.0175073	8.20	0.000	.1092263	.1778537
wellpgm						
yes	.9246275	.2750269	3.36	0.001	.3855848	1.46367
_cons	-3.679888	1.464123	-2.51	0.012	-6.549515	-.81026
completed						
wellpgm						
yes	.6553902	.2263862	2.90	0.004	.2116814	1.099099
experience	-.8153984	.0617977	-13.19	0.000	-.9365196	-.6942772
salaried						
yes	.4709859	.1419878	3.32	0.001	.192695	.7492768
_cons	4.902936	.3973849	12.34	0.000	4.124076	5.681796
wellpgm						
age	-.0938617	.0072734	-12.90	0.000	-.1081173	-.079606
smoke						
yes	-1.477078	.1772103	-8.34	0.000	-1.824404	-1.129752
_cons	4.228481	.337379	12.53	0.000	3.56723	4.889732
weight						
sex						
male	9.506396	.6960864	13.66	0.000	8.142091	10.8707
gym	-.8184902	.0779351	-10.50	0.000	-.9712401	-.6657402
_cons	80.10245	.5407952	148.12	0.000	79.04251	81.16239
var(e.weight~s)	2.015328	.263477			1.559777	2.603927
var(e.weight)	65.98395	3.997213			58.59678	74.30241
corr(e.com~d, e.weightloss)	.5434105	.0824836	6.59	0.000	.362338	.6849556
corr(e.wel~m, e.weightloss)	.5878321	.1054098	5.58	0.000	.3440372	.7573749
corr(e.wei~t, e.weightloss)	-.4801763	.089175	-5.38	0.000	-.6353685	-.2877017
corr(e.wel~m, e.completed)	.3753168	.1523364	2.46	0.014	.0470351	.6304273
corr(e.wei~t, e.completed)	-.0643813	.0718768	-0.90	0.370	-.2030702	.0768401
corr(e.wei~t, e.wellpgm)	-.096324	.0691411	-1.39	0.164	-.2292586	.0401382

The `completed`, `wellpgm`, and `weight` equations provide the coefficient estimates for the auxiliary endogenous selection, treatment assignment, and endogenous covariate models.

The correlation estimates tell us about the endogeneity in our model. For example, we speculated that we might have endogenous selection. The error correlation `corr(e.completed, e.weightloss)` is an estimate of the correlation between the error from the selection equation and the error from the outcome equation. The estimate is significant, so we reject the hypothesis that there is no endogenous selection. It is positive, so we conclude that unobserved factors that increase the likelihood of being in the sample tend to occur with unobserved factors that increase the amount of weight lost. Looking at the other correlations, we find that our suspicions of endogenous treatment choice and the endogeneity of initial weight are likewise confirmed.

We estimated an ATE in our running example. In our simple illustration, we were able to use the coefficient on `wellpgm`. If `wellpgm` had been interacted with other covariates in the model, we would have needed to use `estat teffects`. We also could have estimated the effect of the wellness program on just those employees who participated, the *average treatment effect on the treated* (ATET).

Using this regression, if we ask questions about how participating in `wellpgm` affects the expected change in `weightloss`, we will almost always get the same answer: 0.92 kg greater weight loss with the program than without. That is the coefficient on `wellpgm` in the main outcome equation. This model is linear and contains no interactions between the treatment and other covariates. So, whether we ask about the ATE or the ATET, the answer is 0.92. Whether we ask about the expected additional `weightloss` for a person who chose to participate or about all the women who chose to participate, the answer is the same. No matter what, the expected change is always 0.92.

To make this interesting, we will need a more complex model. We could take the `nointeract` suboption off the `entreat()` option. If we did that and refit the model, all the questions above would produce different answers. But, as we said, our data are simulated with no interaction. So let's use another artifice.

Let's assume that the clerk in charge of the final weigh-in overheard management discussing the new program. The managers seemed particularly interested in participants losing at least 4 kg (8.8 pounds). Thinking he was being helpful, our clerk decided to save everyone some effort and did not record actual weights. Instead, he recorded only whether employees were at least 4 kg lighter than they had been at the initial weigh-in.

We can no longer analyze weight loss, but we can analyze the probability of losing at least 4 kg. We fit the same full model but this time use `eprobit`, and our dependent variable becomes `lost4`, which is 0 if the employee lost less than 4 kg and is 1 if the employee lost 4 kg or more.

```
. eprobit lost4 age i.sex, endogenous(weight = i.sex gym)
> entreat(wellpgm = age i.smoke, nointeract)
> select(completed = i.wellpgm experience i.salaried) vce(robust)

(iteration log omitted)
```

Extended probit regression

Number of obs = 545  
Selected = 337  
Nonselected = 208

Wald chi2(4) = 184.27  
Prob > chi2 = 0.0000

Log pseudolikelihood = -2392.5364

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
lost4						
age	-.0461113	.0129744	-3.55	0.000	-.0715406	-.020682
sex						
male	-1.192968	.1806428	-6.60	0.000	-1.547022	-.8389148
weight	.1131467	.0108868	10.39	0.000	.0918089	.1344844
wellpgm						
yes	1.370215	.4048158	3.38	0.001	.5767905	2.163639
_cons	-8.034426	1.199574	-6.70	0.000	-10.38555	-5.683305
completed						
wellpgm						
yes	.6534203	.2310957	2.83	0.005	.200481	1.10636
experience	-.801973	.0676059	-11.86	0.000	-.9344781	-.6694679
salaried						
yes	.3955088	.1549943	2.55	0.011	.0917255	.6992921
_cons	4.862419	.4186367	11.61	0.000	4.041906	5.682932
wellpgm						
age	-.0958611	.0071251	-13.45	0.000	-.109826	-.0818963
smoke						
yes	-1.515911	.1754356	-8.64	0.000	-1.859758	-1.172063
_cons	4.310847	.338842	12.72	0.000	3.646728	4.974965
weight						
sex						
male	9.501602	.6983151	13.61	0.000	8.13293	10.87028
gym	-.8162669	.0765488	-10.66	0.000	-.9662998	-.666234
_cons	80.09771	.5302486	151.06	0.000	79.05844	81.13697
var(e.weight)	65.98399	3.805168			58.93203	73.8798
corr(e.com~d, e.lost4)	.5236573	.1297834	4.03	0.000	.2268709	.7314522
corr(e.wel~m, e.lost4)	.249717	.2438804	1.02	0.306	-.2493086	.6439525
corr(e.wei~t, e.lost4)	-.6846067	.096236	-7.11	0.000	-.8314263	-.448426
corr(e.wel~m, e.completed)	.3678357	.1636913	2.25	0.025	.014886	.6392761
corr(e.wei~t, e.completed)	-.0821217	.074566	-1.10	0.271	-.2255026	.0647412
corr(e.wei~t, e.wellpgm)	-.0888819	.0671873	-1.32	0.186	-.218281	.0435887

These parameter estimates are pretty close to those from running `eregress` on `weightloss`. But unless you like thinking in terms of shifts along a standardized normal distribution, the coefficient of 1.37 on `wellpgm` is difficult to interpret. We still know that the effect of the program is statistically significant, but little more.

Note that we added `vce(robust)`. This will allow us to treat our sample as a draw from a population when using `estat teffects` and `margins`, and thus make inferences about the population. Otherwise, we would be taking the sample as fixed and not as a draw from a population.

If management is thinking about expanding the program, they will want to evaluate its effectiveness. What proportion of employees across all facilities would lose 4 kg or more naturally, either through all employees not participating or through the program simply not being offered? What proportion would lose 4 kg or more if all employees participated? More to the point, what is the difference in those averages? We type

```
. estat teffects
Predictive margins
```

				Number of obs	=	545
	Unconditional Margin	Std. Err.	z	P> z	[95% Conf. Interval]	
ATE wellpgm (yes vs no)	.3857447	.1195805	3.23	0.001	.1513712	.6201182

Only about 40% of employees would be expected to lose 4 kg; that is the ATE.

A related question is, What is the expected average increase in participants losing 4 kg? Let's estimate the expected effect of the wellness program on just those employees who choose to participate, the ATET.

```
. estat teffects, atet
Predictive margins
```

				Number of obs	=	545
				Subpop. no. obs	=	208
	Unconditional Margin	Std. Err.	z	P> z	[95% Conf. Interval]	
ATET wellpgm (yes vs no)	.5335926	.1322879	4.03	0.000	.274313	.7928722

The ATET of 0.53 implies that just over half of those who choose to participate across all facilities would be expected to lose 4 kg. Recall that we believed success in the program would be positively correlated with employees' decision to participate. That is what made the decision endogenous. It is not surprising that we expect better results for participants than we do for all the employees as a whole.

We are going to need `margins` to answer some other questions, so let's introduce it by reestimating the ATET. First though, we generate a copy of the `wellpgm` variable in `wellpgmT`; `margins` will need it.

```
. generate wellpgmT = wellpgm
. margins r(0 1).wellpgm if wellpgm, predict(base(wellpgm=wellpgmT))
> contrast(effects nowald)
```

Contrasts of predictive margins  
Model VCE : Robust  
Expression : Pr(lost4==1), predict(base(wellpgm=wellpgmT))

	Delta-method		z	P> z	[95% Conf. Interval]	
	Contrast	Std. Err.				
wellpgm (yes vs no)	.5335926	.13197	4.04	0.000	.2749361	.7922491

We have reproduced the estimate.

There is a lot happening in that `margins` command.

`r(0 1).wellpgm` tells `margins` to form two counterfactuals—one at `wellpgm=0` and another at `wellpgm=1`—and to then take the reference (`r`) contrast (difference) of those two counterfactuals.

`if wellpgm` restricts the sample to those who participated in the wellness program.

`predict(base(wellpgm=wellpgmT))` specifies that each counterfactual prediction be conditioned on the employee’s actual decision to participate in the program. These values are recorded in `wellpgmT`. Recall that `margins` changes the data to form the counterfactuals, and thus `predict` must be told where to find the employee’s actual choice. The use and meaning of `predict(base())` are discussed more in [\[ERM\] intro 6](#).

`contrast(effects nowald)` tells `margins` to report the  $z$  statistic and probability  $> z$ , which are not shown by default. It also tells `margins` to suppress the overall Wald statistic.

The standard errors are slightly smaller than those from `estat teffects`. If we wanted them to match exactly, we would use the `vce(unconditional)` option with `margins`. That option creates standard errors appropriate to make inferences about the population. The standard errors are so close that we will dispense with `vce(unconditional)` in this section.

Now, let’s ask a series of different questions from a different perspective.

The physical trainer for our fictional company is having lunch with a new employee, Betty. The trainer mentions the wellness program, and Betty asks if it is likely to do her much good. Betty looks to be mid thirties and average weight. She says she goes to the gym a couple of times a month. The trainer recalls people with those characteristics doing well with the program. Betty’s data are already in the company’s database, so the trainer opens Stata on her laptop and types

```
. margins r(0 1).wellpgm if name=="Betty", predict(fix(wellpgm))
> contrast(effects nowald) noesample
```

Warning: prediction constant over observations.

Contrasts of predictive margins  
Model VCE : Robust  
Expression : Pr(lost4==1), predict(fix(wellpgm))

	Delta-method		z	P> z	[95% Conf. Interval]	
	Contrast	Std. Err.				
wellpgm (yes vs no)	.6472455	.1350621	4.79	0.000	.3825287	.9119622

The trainer tells Betty that employees with her characteristics have about a 65% chance of losing 4 kg when they are in the program.

Later, another new employee, Fred, asks whether the program is likely to help him lose that last few kilograms. He is thin, in his upper fifties, and he already goes to the gym about twice a week. Our trainer types

```
. margins r(0 1).wellpgm if name=="Fred", predict(fix(wellpgm))
> contrast(effects nowald) noesample
Warning: prediction constant over observations.
Contrasts of predictive margins
Model VCE      : Robust
Expression     : Pr(lost4==1), predict(fix(wellpgm))
```

	Delta-method		z	P> z	[95% Conf. Interval]	
	Contrast	Std. Err.				
wellpgm (yes vs no)	.0184247	.0225112	0.82	0.413	-.0256965	.0625459

She tells Fred that the program might be good for him but not to expect it to create much weight loss. Fred says he would like to sign up, just so he can meet some other employees.

When Fred leaves, our trainer calls her office mate and makes a wager that Fred will not lose 4 kg on the program. The trainer then realizes that she placed a bet on overall weight loss, not just the loss attributable to the wellness program. To be certain, she checks the potential outcomes of weight loss for Fred being in the program and for Fred being out of the program.

```
. margins i(0 1).wellpgm if name=="Fred", predict(fix(wellpgm)) noesample
Warning: prediction constant over observations.
Predictive margins                                Number of obs      =          1
Model VCE      : Robust
Expression     : Pr(lost4==1), predict(fix(wellpgm))
```

	Delta-method		z	P> z	[95% Conf. Interval]	
	Margin	Std. Err.				
wellpgm no	.0000365	.0000718	0.51	0.612	-.0001043	.0001773
yes	.0184612	.0225357	0.82	0.413	-.025708	.0626304

With a negligible chance of losing 4 kg if Fred chooses not to participate and a slim 2% chance if Fred does participate, our trainer feels pretty good about her wager. Even the upper bound of the confidence intervals makes the trainer confident. Of course, these are the expected results for all employees with Fred's characteristics; Fred might be an overachiever.

Note that our trainer used `predict(fix(wellpgm))` to answer all of these questions. That is both the right and the only thing to do. Neither of these new employees has yet made a choice whether to participate. They have not revealed their unobserved characteristics that cause their weight loss and their decision to participate to be correlated. We called this unobserved characteristic the “willingness to engage in healthy behaviors” when we described the model for our data. Unlike when we computed ATET, we do not yet know Fred's and Betty's treatment choices, so we cannot use `base()` and thus condition our inferences on that additional information. We can make statements only about fixed levels of treatment.

The counterfactuals and contrasts that we computed for Betty and Fred are the expected values from our model conditioned only on the exogenous covariates in the main equation, `age` and `sex`,



and on fixing the values of `wellpgm` first to 0 and then to 1. By “fixing”, we mean setting them to 0 and 1, not letting Betty or Fred choose 0 or 1. These estimates are no better or worse than the ATETs we estimated using `base()`. They are based on less information but use all the information we have about Betty and Fred. The estimates for Betty are what we would expect if we averaged over hundreds of employees who match Betty’s age and sex. The same applies to Fred.

Also note that we typed `r(0 1)`., rather than just `r.` That is because we are operating on a single observation, and `margins` cannot determine the appropriate levels of `wellpgm` for which to form counterfactuals. We had to tell `margins` to use 0 and 1.

It is unlikely that our trainer has Stata on her laptop or has the inclination to type `margins` commands. As analysts, however, we might create a table for her that she can use to assess candidates and help employees form realistic expectations.

Our dataset already has grouping variables for `age`, `gym`, `weight`, and `sex`. We can estimate the expected additional probability of losing more than 4 kg for each combination of these groups by using an `over()` option.

```
. margins r.wellpgm, predict(fix(wellpgm)) contrast(effects nowald)
> over(agegrp gymgrp wtgrp sex)
Contrasts of predictive margins
Model VCE      : Robust
Expression    : Pr(lost4==1), predict(fix(wellpgm))
over          : agegrp gymgrp wtgrp sex
```

	Delta-method		z	P> z	[95% Conf. Interval]	
	Contrast	Std. Err.				
wellpgm@ag~p# gymgrp# wtgrp#sex (yes vs no)						
20-29 0						
60-69						
female	.6430664	.1601075	4.02	0.000	.3292614	.9568714
(yes vs no)						
20-29 0						
60-69 male	0	(omitted)				
(output omitted)						
(yes vs no)						
60 up 11 up						
< 60 female	.0032702	.0059912	0.55	0.585	-.0084723	.0150128
(yes vs no)						
60 up 11 up						
< 60 male	0	(omitted)				

Those rows marked (omitted) represent combinations of characteristics for which we do not have any employees in our sample. We could use our model to extrapolate to those groups, but we are not going to do that. What we do have for each combination of groups is an estimate of the expected increase in the probability of losing 4 kg, a test that the probability is greater than 0, and a 95% confidence interval.

Those results will take a lot of transcription to create something compact for the trainer. And while our hearts are warmed by the tests and confidence intervals, the trainer might not feel the same way. If we wanted to be exceptionally helpful, we could build a table manually showing ATEs for each group.

```
. predict te, te
. table gymgrp wtgrp sex, by(agegrp) contents(mean te) format(%4.2f)
```

Age groups and Gym visit groups	Employee sex; 0=female, 1=male and Weight groups									
	female					male				
	< 60	60-69	70-79	80-89	90 up	< 60	60-69	70-79	80-89	90 up
20-29										
0		0.64	0.58	0.53	0.41			0.63	0.62	0.52
0-5		0.65	0.63	0.57				0.64	0.65	0.59
6-10		0.51	0.64	0.65				0.55	0.58	0.64
11 up		0.40						0.35		
30-39										
0	0.48		0.65	0.62	0.61			0.61	0.65	0.62
0-5		0.48	0.64	0.64					0.56	0.62
6-10	0.21	0.39	0.54	0.60		0.27	0.31	0.31	0.51	0.48
11 up		0.34	0.43					0.24		0.53
40-49										
0		0.45	0.57	0.62	0.65			0.33	0.50	0.62
0-5		0.39	0.48	0.59	0.61			0.27	0.38	0.55
6-10		0.22	0.33	0.38		0.09	0.14	0.14	0.25	0.42
11 up	0.07	0.10		0.28		0.09	0.08	0.08	0.19	
50-59										
0		0.26	0.35	0.46	0.62			0.25	0.29	0.47
0-5	0.06		0.22	0.40	0.62			0.12	0.20	0.32
6-10	0.05	0.04	0.22	0.16	0.39			0.05	0.10	0.13
11 up	0.01	0.01	0.05			0.01	0.01			
60 up										
0			0.17	0.25	0.37			0.07	0.12	0.26
0-5			0.07	0.33	0.28	0.02			0.03	0.11
6-10		0.02	0.04	0.10		0.01	0.02	0.02	0.03	0.08
11 up	0.00		0.02					0.00	0.02	0.03

We first predicted the expected treatment effects for each observation in our sample. Then, we let `table` average those values for each combination of groups. For any combination of groups, these estimates match those from `margins`.

## References

- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- . 2010. *Microeconometrics Using Stata*. Rev. ed. College Station, TX: Stata Press.
- Roodman, D. 2011. Fitting fully observed recursive mixed-process models with `cmp`. *Stata Journal* 11: 159–206.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

- [ERM] [intro 1](#) — An introduction to the ERM commands
- [ERM] [intro 6](#) — Model interpretation
- [ERM] [Glossary](#)

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

**eintreg** fits an interval regression model that accommodates any combination of endogenous covariates, nonrandom treatment assignment, and endogenous sample selection. Continuous, binary, and ordinal endogenous covariates are allowed. Treatment assignment may be endogenous or exogenous. A probit or tobit model may be used to account for endogenous sample selection.

## Quick start

All quick start examples use an interval-measured dependent variable with the interval's lower bound recorded in variable `y_l` and its upper bound recorded in `y_u`.

Regression of  $[y_l, y_u]$  on  $x$  with continuous endogenous covariate  $y_2$  modeled by  $x$  and  $z$

```
eintreg y_l y_u x, endogenous(y2 = x z)
```

As above, but adding continuous endogenous covariate  $y_3$  modeled by  $x$  and  $z_2$

```
eintreg y_l y_u x, endogenous(y2 = x z) endogenous(y3 = x z2)
```

Regression of  $[y_l, y_u]$  on  $x$  with binary endogenous covariate  $d$  modeled by  $x$  and  $z$

```
eintreg y_l y_u x, endogenous(d = x z, probit)
```

Regression of  $[y_l, y_u]$  on  $x$  with endogenous treatment recorded in `trtvar` and modeled by  $x$  and  $z$

```
eintreg y_l y_u x, entreat(trtvar = x z)
```

Regression of  $[y_l, y_u]$  on  $x$  with exogenous treatment recorded in `trtvar`

```
eintreg y_l y_u x, extreat(trtvar)
```

Regression of  $[y_l, y_u]$  on  $x$  with endogenous sample-selection indicator `selvar` modeled by  $x$  and  $z$

```
eintreg y_l y_u x, select(selvar = x z)
```

As above, but adding endogenous covariate  $y_2$  modeled by  $x$  and  $z_2$

```
eintreg y_l y_u x, select(selvar = x z) endogenous(y2 = x z2)
```

As above, but adding endogenous treatment recorded in `trtvar` and modeled by  $x$  and  $z_3$

```
eintreg y_l y_u x, select(selvar = x z) endogenous(y2 = x z2) ///
    entreat(trtvar = x z3)
```

# Menu

Statistics > Endogenous covariates > Models adding selection and treatment > Interval regression

## Syntax

*Basic interval regression with endogenous covariates*

```
eintreg depvar1 depvar2 [indepvars] ,
      endogenous(depvarsen = varlisten) [options]
```

*Basic interval regression with endogenous treatment assignment*

```
eintreg depvar1 depvar2 [indepvars] ,
      entreat(depvartr [= varlisttr]) [options]
```

*Basic interval regression with exogenous treatment assignment*

```
eintreg depvar1 depvar2 [indepvars] ,
      extreat(tvar) [options]
```

*Basic interval regression with sample selection*

```
eintreg depvar1 depvar2 [indepvars] ,
      select(depvars = varlists) [options]
```

*Basic interval regression with tobit sample selection*

```
eintreg depvar1 depvar2 [indepvars] ,
      tobitselect(depvars = varlists) [options]
```

*Interval regression combining endogenous covariates, treatment, and selection*

```
eintreg depvar1 depvar2 [indepvars] [if] [in] [weight] [, extensions options]
```

*depvar*<sub>1</sub> and *depvar*<sub>2</sub> should have the following form:

Type of data		<i>depvar</i> <sub>1</sub>	<i>depvar</i> <sub>2</sub>
point data	$a = [a, a]$	<i>a</i>	<i>a</i>
interval data	$[a, b]$	<i>a</i>	<i>b</i>
left-censored data	$(-\infty, b]$	.	<i>b</i>
right-censored data	$[a, +\infty)$	<i>a</i>	.
missing		.	.

<i>extensions</i>	Description
Model	
<u>endogenous</u> ( <i>enspec</i> )	model for endogenous covariates; may be repeated
<u>entreat</u> ( <i>entrspec</i> )	model for endogenous treatment assignment
<u>extreat</u> ( <i>extrspec</i> )	exogenous treatment
<u>select</u> ( <i>selspec</i> )	probit model for selection
<u>tobitselect</u> ( <i>tselspec</i> )	tobit model for selection
<i>options</i>	Description
Model	
<u>noconstant</u>	suppress constant term
<u>offset</u> ( <i>varname<sub>o</sub></i> )	include <i>varname<sub>o</sub></i> in model with coefficient constrained to 1
<u>constraints</u> ( <i>numlist</i> )	apply specified linear constraints
<u>collinear</u>	keep collinear variables
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>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> (#)	set the number of integration (quadrature) points for integration over four or more dimensions; default is intpoints(128)
<u>triintpoints</u> (#)	set the number of integration (quadrature) points for integration over three dimensions; default is triintpoints(10)
Maximization	
<u>maximize_options</u>	control the maximization process; seldom used
<u>coeflegend</u>	display legend instead of statistics

*enspec* is *depvars<sub>en</sub>* = *varlist<sub>en</sub>* [ , *enopts* ]  
where *depvars<sub>en</sub>* is a list of endogenous covariates. Each variable in *depvars<sub>en</sub>* specifies an endogenous covariate model using the common *varlist<sub>en</sub>* and options.

*entrspec* is *depvar<sub>tr</sub>* [= *varlist<sub>tr</sub>*] [ , *tropts* ]  
where *depvar<sub>tr</sub>* is a variable indicating treatment assignment. *varlist<sub>tr</sub>* is a list of covariates predicting treatment assignment.

*extrspec* is *tvar* [ , nomain nointeract ]  
where *tvar* is a variable indicating treatment assignment.

*selspec* is `depvars = varlists [ , noconstant offset(varnameo) ]`

where *depvar<sub>s</sub>* is a variable indicating selection status. *depvar<sub>s</sub>* must be coded as 0, indicating that the observation was not selected, or 1, indicating that the observation was selected. *varlist<sub>s</sub>* is a list of covariates predicting selection.

*tselspec* is `depvars = varlists [ , tseopts ]`

where *depvar<sub>s</sub>* is a continuous variable. *varlist<sub>s</sub>* is a list of covariates predicting *depvar<sub>s</sub>*. The censoring status of *depvar<sub>s</sub>* indicates selection, where a censored *depvar<sub>s</sub>* indicates that the observation was not selected and a noncensored *depvar<sub>s</sub>* indicates that the observation was selected.

<i>enopts</i>	Description
Model	
<u>probit</u>	treat endogenous covariate as binary
<u>oprobit</u>	treat endogenous covariate as ordinal
<u>nomain</u>	do not add endogenous covariate to main equation
<u>noconstant</u>	suppress constant term

<i>tropts</i>	Description
Model	
<u>nomain</u>	do not add treatment indicator to main equation
<u>nointeract</u>	do not interact treatment with covariates in main equation
<u>noconstant</u>	suppress constant term
<u>offset</u> (varname <sub>o</sub> )	include varname <sub>o</sub> in model with coefficient constrained to 1

<i>tseopts</i>	Description
Model	
<u>ll</u> (varname   #)	left-censoring variable or limit
<u>ul</u> (varname   #)	right-censoring variable or limit
<u>main</u>	add censored selection variable to main equation
<u>noconstant</u>	suppress constant term
<u>offset</u> (varname <sub>o</sub> )	include varname <sub>o</sub> in model with coefficient constrained to 1

*indepvars*, *varlist<sub>en</sub>*, *varlist<sub>tr</sub>*, and *varlist<sub>s</sub>* may contain factor variables; see [U] 11.4.3 Factor variables.

*depvar<sub>1</sub>*, *depvar<sub>2</sub>*, *indepvars*, *depvars<sub>en</sub>*, *varlist<sub>en</sub>*, *depvar<sub>tr</sub>*, *varlist<sub>tr</sub>*, *tvar*, *depvar<sub>s</sub>*, and *varlist<sub>s</sub>* may contain time-series operators; see [U] 11.4.4 Time-series varlists.

*bootstrap*, *by*, *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*.

*vce()* and weights are not allowed with the *svy* prefix; see [SVY] *svy*.

*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

`endogenous(enspec)`, `entreat(entrspec)`, `extreat(extrspec)`, `select(selspec)`, `tobitselect(tselspec)`; see [\[ERM\] erm options](#).

`noconstant`, `offset(varnameo)`, `constraints(numlist)`, `collinear`; see [\[R\] estimation options](#).

### SE/Robust

`vce(vcetype)`; see [\[ERM\] erm options](#).

### Reporting

`level(#)`, `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(#)`, `triintpoints(#)`; see [\[ERM\] erm 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](#).

Setting the optimization type to `technique(bhhh)` resets the default `vcetype` to `vce(opg)`.

The following option is available with `eintreg` but is not shown in the dialog box:

`coeflegend`; see [\[R\] estimation options](#).

## Remarks and examples

`eintreg` fits models that we refer to as “extended interval regression models”, meaning that they accommodate endogenous covariates, nonrandom treatment assignment, and endogenous sample selection. `eintreg` can account for these complications whether they arise individually or in combination.

In this entry, you will find information on the `eintreg` command syntax. You can see [Methods and formulas](#) for a full description of the models that can be fit with `eintreg` and details about how those models are fit.

More information on extended interval regression models is found in the separate introductions and example entries. We recommend reading those entries to learn how to use `eintreg`. Below, we provide a guide to help you locate the ones that will be helpful to you.

For an introduction to `eintreg` and the other extended regression commands (`eregress`, `eprobit`, and `eoprobit`), see [\[ERM\] intro 1](#)–[\[ERM\] intro 8](#).

[\[ERM\] intro 1](#) introduces the ERM commands, the problems they address, and their syntax.

[ERM] [intro 2](#) provides background on the four types of models—linear regression, interval regression, probit regression, and ordered probit regression—that can be fit using ERM commands. This intro also demonstrates how to fit a tobit model using `eintreg` by transforming your dependent variable into the required format.

[ERM] [intro 3](#) considers the problem of endogenous covariates and how to solve it using ERM commands.

[ERM] [intro 4](#) gives an overview of endogenous sample selection and using ERM commands to account for it.

[ERM] [intro 5](#) covers nonrandom treatment assignment and how to account for it using `eintreg` or any of the other ERM commands.

[ERM] [intro 6](#) discusses interpretation of results. You can interpret coefficients from `eintreg` in the usual way, but this introduction goes beyond the interpretation of coefficients. We demonstrate how to find answers to interesting questions by using `margins`. If your model includes an endogenous covariate or an endogenous treatment, the use of `margins` differs from its use after other estimation commands, so we strongly recommend reading this intro if you are fitting these types of models.

[ERM] [intro 7](#) will be helpful if you are familiar with `ivtobit` and other commands that address endogenous covariates, sample selection, or nonrandom treatment assignment. This introduction is a Rosetta stone that maps the syntax of those commands to the syntax of `eintreg`.

[ERM] [intro 8](#) walks you through an example that gives insight into the concepts of endogenous covariates, treatment assignment, and sample selection while fitting models with `eregress` that address these complications. Although the example uses `eregress`, the discussion applies equally to `eintreg`. This intro also demonstrates how to interpret results by using `margins` and `estat teffects`.

Additional examples are presented in [ERM] [example 1a](#)–[ERM] [example 6b](#). For examples using `eintreg`, see

[ERM] <a href="#">example 1b</a>	Interval regression with continuous endogenous covariate
[ERM] <a href="#">example 1c</a>	Interval regression with endogenous covariate and sample selection

See [Examples](#) in [ERM] [intro](#) for an overview of all the examples. These examples demonstrate all four extended regression commands, and all may be interesting because they handle complications in the same way. Examples using `eregress` will be of particular interest because results of models fit by `eintreg` are interpreted in the same way.

You can also find in literature discussion and examples of many models that `eintreg` can fit. For instance, the tobit model was originally conceived in [Tobin \(1958\)](#) as a model of consumption of consumer durables, where purchases were left-censored at 0. [Wooldridge \(2016, sec. 17.4\)](#) introduces censored and truncated regression models. [Cameron and Trivedi \(2010, chap. 16\)](#) discuss the tobit model using Stata examples. `eintreg` can also fit models like the tobit regression model with continuous endogenous regressors ([Newey 1987](#)) and models like the censored regression model with binary endogenous regressors ([Angrist 2001](#)). [Roodman \(2011\)](#) investigated interval regression models with endogenous covariates and endogenous sample selection, and demonstrated how multiple observational data complications could be addressed with a triangular model structure. His work has been used to model processes like the effect of innovation on labor productivity ([Mairesse and Robin 2009](#)) and the effect of insect-resistant crops on pesticide demand ([Fernandez-Cornejo and Wechsler 2012](#)).



## Stored results

**eintreg** stores the following in **e()**:

### Scalars

<b>e(N)</b>	number of observations
<b>e(N_selected)</b>	number of selected observations
<b>e(N_nonselected)</b>	number of nonselected observations
<b>e(N_unc)</b>	number of uncensored observations
<b>e(N_lrc)</b>	number of left-censored observations
<b>e(N_rc)</b>	number of right-censored observations
<b>e(N_int)</b>	number of interval-censored observations
<b>e(k)</b>	number of parameters
<b>e(k_cat#)</b>	number of categories for the <i>#th depvar</i> , ordinal
<b>e(k_eq)</b>	number of equations in <b>e(b)</b>
<b>e(k_eq_model)</b>	number of equations in overall model test
<b>e(k_dv)</b>	number of dependent variables
<b>e(k_aux)</b>	number of auxiliary parameters
<b>e(df_m)</b>	model degrees of freedom
<b>e(ll)</b>	log likelihood
<b>e(N_clust)</b>	number of clusters
<b>e(chi2)</b>	$\chi^2$
<b>e(p)</b>	significance
<b>e(n_quad)</b>	number of integration points for multivariate normal
<b>e(n_quad3)</b>	number of integration points for trivariate normal
<b>e(rank)</b>	rank of <b>e(V)</b>
<b>e(ic)</b>	number of iterations
<b>e(rc)</b>	return code
<b>e(converged)</b>	1 if converged, 0 otherwise

### Macros

<b>e(cmd)</b>	<b>eintreg</b>
<b>e(cmdline)</b>	command as typed
<b>e(depvar)</b>	names of dependent variables
<b>e(wtype)</b>	weight type
<b>e(wexp)</b>	weight expression
<b>e(title)</b>	title in estimation output
<b>e(clustvar)</b>	name of cluster variable
<b>e(offset#)</b>	offset for the <i>#th depvar</i>
<b>e(chi2type)</b>	Wald; type of model $\chi^2$ test
<b>e(vce)</b>	<i>vcetype</i> specified in <b>vce()</b>
<b>e(vcetype)</b>	title used to label Std. Err.
<b>e(opt)</b>	type of optimization
<b>e(which)</b>	max or min; whether optimizer is to perform maximization or minimization
<b>e(ml_method)</b>	type of ml method
<b>e(user)</b>	name of likelihood-evaluator program
<b>e(technique)</b>	maximization technique
<b>e(properties)</b>	<b>b V</b>
<b>e(estat_cmd)</b>	program used to implement <b>estat</b>
<b>e(predict)</b>	program used to implement <b>predict</b>
<b>e(marginsok)</b>	predictions allowed by <b>margins</b>
<b>e(marginsnotok)</b>	predictions disallowed by <b>margins</b>
<b>e(asbalanced)</b>	factor variables <b>fvset</b> as <b>asbalanced</b>
<b>e(asobserved)</b>	factor variables <b>fvset</b> as <b>asobserved</b>

### Matrices

<b>e(b)</b>	coefficient vector
<b>e(cat#)</b>	categories for the <i>#th depvar</i> , ordinal
<b>e(Cns)</b>	constraints matrix
<b>e(ilog)</b>	iteration log (up to 20 iterations)
<b>e(gradient)</b>	gradient vector
<b>e(V)</b>	variance-covariance matrix of the estimators
<b>e(V_modelbased)</b>	model-based variance

### Functions

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

## Methods and formulas

The methods and formulas presented here are for the interval-outcome model. They are a special case of the methods and formulas derived by [Roodman \(2011\)](#).

Methods and formulas are presented under the following headings:

- [Introduction](#)
- [Endogenous covariates](#)
  - [Continuous endogenous covariates](#)
  - [Binary and ordinal endogenous covariates](#)
- [Treatment](#)
- [Endogenous sample selection](#)
  - [Probit endogenous sample selection](#)
  - [Tobit endogenous sample selection](#)
- [Combinations of features](#)
- [Confidence intervals](#)

## Introduction

A regression model of outcome  $y_i$  on covariates  $\mathbf{x}_i$  may be written as

$$y_i = \mathbf{x}_i\beta + \epsilon_i$$

where  $\epsilon_i$  is normal with mean 0 and variance  $\sigma^2$ . Instead of observing  $y_i$ , we observe the endpoints  $y_{li}$  and  $y_{ui}$ .

If  $y_i$  is left-censored, the lower endpoint  $y_{li} = -\infty$  and we know that  $y_i \leq y_{ui}$ . If  $y_i$  is right-censored, the upper endpoint  $y_{ui} = +\infty$  and we know that  $y_i \geq y_{li}$ . If there is no censoring,  $y_{li} = y_{ui} = y_i$ . When  $y_{li}$  and  $y_{ui}$  are real valued and not equal, we know that  $y_{li} \leq y_i \leq y_{ui}$ .

The log likelihood is

$$\begin{aligned} \ln L = & \sum_{i \in U} w_i \ln \phi(y_i - \mathbf{x}_i\beta, \sigma^2) \\ & + \sum_{i \in L} w_i \ln \Phi\left(\frac{y_{ui} - \mathbf{x}_i\beta}{\sigma}\right) \\ & + \sum_{i \in R} w_i \ln \Phi\left(\frac{-y_{li} + \mathbf{x}_i\beta}{\sigma}\right) \\ & + \sum_{i \in I} w_i \ln \left\{ \Phi\left(\frac{y_{ui} - \mathbf{x}_i\beta}{\sigma}\right) - \Phi\left(\frac{y_{li} - \mathbf{x}_i\beta}{\sigma}\right) \right\} \end{aligned}$$

where  $U$  is the set of observations where  $y_i$  is not censored,  $L$  is the set of observations where  $y_i$  is left-censored,  $R$  is the set of observations where  $y_i$  is right-censored,  $I$  is the set of observations where  $y_i$  is interval-censored, and  $w_i$  are the weights.

The conditional mean of  $y_i$  is

$$E(y_i | \mathbf{x}_i) = \mathbf{x}_i\beta$$

If we wished to condition on the censoring, we could calculate an expectation on  $y_i^* = \max\{y_{li}, \min(y_{ij}, y_{ui})\}$  or a constrained mean  $E(y_i | y_{li} < y_i < y_{ui})$ . See [Predictions considering total effects](#) in [\[ERM\] eprobit postestimation](#) for details on how this is done.

If you are willing to take our word for some derivations and notation, the following is complete. Longer explanations and derivations for some terms and functions are provided in [Methods and formulas](#) of [ERM] **eprobit**. For example, we need the two-sided probability function  $\Phi_d^*$  that is discussed in [Introduction](#) in [ERM] **eprobit**.

If you are interested in all the details, we suggest you read [Methods and formulas](#) of [ERM] **eprobit** in its entirety before reading this section. Here, we mainly show how the complications that arise in ERMs are handled in an interval regression framework.

## Endogenous covariates

### Continuous endogenous covariates

An interval regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and  $C$  continuous endogenous covariates  $\mathbf{w}_{ci}$  has the form

$$y_i = \mathbf{x}_i\beta + \mathbf{w}_{ci}\beta_c + \epsilon_i$$

$$\mathbf{w}_{ci} = \mathbf{z}_{ci}\mathbf{A}_c + \epsilon_{ci}$$

As in [Introduction](#), we do not observe  $y_i$  but instead observe the endpoints  $y_{li}$  and  $y_{ui}$ . The vector  $\mathbf{z}_{ci}$  contains variables from  $\mathbf{x}_i$  and other covariates that affect  $\mathbf{w}_{ci}$ . For the model to be identified,  $\mathbf{z}_{ci}$  must contain one extra exogenous covariate not in  $\mathbf{x}_i$  for each of the endogenous regressors in  $\mathbf{w}_{ci}$ . The unobserved errors  $\epsilon_i$  and  $\epsilon_{ci}$  are multivariate normal with mean 0 and covariance

$$\Sigma = \begin{bmatrix} \sigma^2 & \sigma'_{1c} \\ \sigma_{1c} & \Sigma_c \end{bmatrix}$$

Conditional on the endogenous and exogenous covariates,  $\epsilon_i$  has mean and variance

$$\mu_{1|c,i} = E(\epsilon_i | \mathbf{w}_{ci}, \mathbf{x}_i, \mathbf{z}_{ci}) = \sigma'_{1c} \Sigma_c^{-1} (\mathbf{w}_{ci} - \mathbf{z}_{ci} \mathbf{A}_c)'$$

$$\sigma_{1|c}^2 = \text{Var}(\epsilon_i | \mathbf{w}_{ci}, \mathbf{x}_i, \mathbf{z}_{ci}) = \sigma^2 - \sigma'_{1c} \Sigma_c^{-1} \sigma_{1c}$$

Let

$$r_{li} = y_{li} - \mathbf{x}_i\beta - \mathbf{w}_{ci}\beta_c - \mu_{1|c,i}$$

$$r_{ui} = y_{ui} - \mathbf{x}_i\beta - \mathbf{w}_{ci}\beta_c - \mu_{1|c,i}$$

The log likelihood is

$$\begin{aligned} \ln L = & \sum_{i \in U} w_i \ln \phi(r_{li}, \sigma_{1|c}^2) \\ & + \sum_{i \in L} w_i \ln \Phi_1^*(-\infty, r_{ui}, \sigma_{1|c}^2) \\ & + \sum_{i \in R} w_i \ln \Phi_1^*(r_{li}, \infty, \sigma_{1|c}^2) \\ & + \sum_{i \in I} w_i \ln \Phi_1^*(r_{li}, r_{ui}, \sigma_{1|c}^2) \\ & + \sum_{i=1}^N w_i \ln \phi_C(\mathbf{w}_{ci} - \mathbf{z}_{ci} \mathbf{A}_c, \Sigma_c) \end{aligned}$$

where  $U$  is the set of observations where  $y_i$  is not censored,  $L$  is the set of observations where  $y_i$  is left-censored,  $R$  is the set of observations where  $y_i$  is right-censored, and  $I$  is the set of observations where  $y_i$  is interval-censored.

The conditional mean of  $y_i$  is

$$E(y_i | \mathbf{x}_i, \mathbf{w}_{ci}, \mathbf{z}_{ci}) = \mathbf{x}_i \boldsymbol{\beta} + \mathbf{w}_{ci} \boldsymbol{\beta}_c + \boldsymbol{\sigma}'_{1c} \boldsymbol{\Sigma}_c^{-1} (\mathbf{w}_{ci} - \mathbf{z}_{ci} \mathbf{A}_c)'$$

## Binary and ordinal endogenous covariates

Here, we begin by formulating the interval regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and  $B$  binary and ordinal endogenous covariates  $\mathbf{w}_{bi} = [w_{b1i}, \dots, w_{bBi}]$ . Indicator (dummy) variables for the levels of each binary and ordinal covariate are used in the model. You can also interact other covariates with the binary and ordinal endogenous covariates, as in treatment-effect models.

The binary and ordinal endogenous covariates  $\mathbf{w}_{bi}$  are formulated as in [Binary and ordinal endogenous covariates](#) in [ERM] **eprobit**. So we have

$$y_i = \mathbf{x}_i \boldsymbol{\beta} + \mathbf{wind}_{b1i} \boldsymbol{\beta}_{b1} + \dots + \mathbf{wind}_{bBi} \boldsymbol{\beta}_{bB} + \epsilon_i$$

The  $\mathbf{wind}_{bji}$  vectors are defined in [Binary and ordinal endogenous covariates](#) in [ERM] **eprobit**. As in [Introduction](#), we do not observe  $y_i$  but instead observe the endpoints  $y_{li}$  and  $y_{ui}$ . The binary and ordinal endogenous errors  $\epsilon_{b1i}, \dots, \epsilon_{bBi}$  and outcome error  $\epsilon_i$  are multivariate normal with 0 mean and covariance

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_b & \boldsymbol{\sigma}_{1b} \\ \boldsymbol{\sigma}'_{1b} & \sigma^2 \end{bmatrix}$$

From here, we discuss the model with ordinal endogenous covariates. The results for binary endogenous covariates are similar.

As in [Binary and ordinal endogenous covariates](#) in [ERM] **eregress**, for the uncensored observations, we write the joint density of  $y_i$  and  $\mathbf{w}_{bi}$  using the conditional density of  $\epsilon_{b1i}, \dots, \epsilon_{bBi}$  on  $\epsilon_i$ . For the censored observations, we use tools discussed in [Likelihood for multiequation models](#) in [ERM] **eprobit** to formulate the joint density directly.

For  $i \in U$ , the uncensored observations, define

$$r_i = y_i - (\mathbf{x}_i \boldsymbol{\beta} + \mathbf{wind}_{b1i} \boldsymbol{\beta}_{b1} + \dots + \mathbf{wind}_{bBi} \boldsymbol{\beta}_{bB})$$

For the censored observations, define

$$\begin{aligned} r_{li} &= y_{li} - (\mathbf{x}_i \boldsymbol{\beta} + \mathbf{wind}_{b1i} \boldsymbol{\beta}_{b1} + \dots + \mathbf{wind}_{bBi} \boldsymbol{\beta}_{bB}) \\ r_{ui} &= y_{ui} - (\mathbf{x}_i \boldsymbol{\beta} + \mathbf{wind}_{b1i} \boldsymbol{\beta}_{b1} + \dots + \mathbf{wind}_{bBi} \boldsymbol{\beta}_{bB}) \end{aligned}$$

Let

$$\boldsymbol{\Sigma}_{b|1} = \boldsymbol{\Sigma} - \frac{\boldsymbol{\sigma}_{1b} \boldsymbol{\sigma}'_{1b}}{\sigma^2}$$

Now, the log likelihood is

$$\begin{aligned} \ln L &= \sum_{i \in U} w_i \ln \{ \Phi_B^*(\mathbf{l}_i, \mathbf{u}_i, \boldsymbol{\Sigma}_{b|1}) \phi(r_i, \sigma^2) \} \\ &+ \sum_{i \in L} w_i \ln \Phi_{B+1}^*([ \mathbf{l}_{bi} \quad -\infty ], [ \mathbf{u}_{bi} \quad r_{ui} ], \boldsymbol{\Sigma}) \\ &+ \sum_{i \in R} w_i \ln \Phi_{B+1}^*([ \mathbf{l}_{bi} \quad r_{li} ], [ \mathbf{u}_{bi} \quad \infty ], \boldsymbol{\Sigma}) \\ &+ \sum_{i \in I} w_i \ln \Phi_{B+1}^*([ \mathbf{l}_{bi} \quad r_{li} ], [ \mathbf{u}_{bi} \quad r_{ui} ], \boldsymbol{\Sigma}) \end{aligned}$$

where  $U$  is the set of observations where  $y_i$  is not censored,  $L$  is the set of observations where  $y_i$  is left-censored,  $R$  is the set of observations where  $y_i$  is right-censored, and  $I$  is the set of observations where  $y_i$  is interval-censored. The vectors  $\mathbf{l}_{bi}$  and  $\mathbf{u}_{bi}$  are the upper and lower limits for the binary and ordinal endogenous regressors defined in [Binary and ordinal endogenous covariates](#) in [ERM] **eprobit**. The vectors  $\mathbf{l}_i$  and  $\mathbf{u}_i$  are the upper and lower limits for the binary and ordinal endogenous regressors defined in [Binary and ordinal endogenous covariates](#) in [ERM] **eregress**.

The expected value of  $y_i$  conditional on  $\mathbf{w}_{bi}$  can be calculated using the techniques discussed in [Predictions considering total effects](#) in [ERM] **eprobit postestimation**.

## Treatment

In the potential-outcomes framework, the treatment  $t_i$  is a discrete variable taking  $T$  values, indexing the  $T$  potential outcomes of the outcome  $y_i$ :  $y_{1i}, \dots, y_{Ti}$ .

When we observe treatment  $t_i$  with levels  $v_1, \dots, v_T$ , we have

$$y_i = \sum_{j=1}^T 1(t_i = v_j) y_{ji}$$

So for each observation, we only observe the potential outcome associated with that observation's treatment value.

For exogenous treatments, our approach is equivalent to the regression adjustment treatment-effect estimation method. See [TE] [teffects intro advanced](#). We do not model the treatment assignment process. The formulas for the treatment effects and potential-outcome means (POMs) are equivalent to what we provide here for endogenous treatments. The treatment effect on the treated for  $\mathbf{x}_i$  for an exogenous treatment is equivalent to what we provide here for the endogenous treatment when the correlation parameter between the outcome and treatment errors is set to 0. The average treatment effects (ATES) and POMs for exogenous treatments are estimated as predictive margins in an analogous manner to what we describe here for endogenous treatments.

From here, we assume an endogenous treatment  $t_i$ . As in [Treatment](#) in [ERM] **eprobit**, we model the treatment assignment process with a probit or ordered probit model, and we call the treatment assignment error  $\epsilon_{ti}$ . An interval regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and endogenous treatment  $t_i$  taking values  $v_1, \dots, v_T$  has the form

$$\begin{aligned} y_{1i} &= \mathbf{x}_i \beta_1 + \epsilon_{1i} \\ &\vdots \\ y_{Ti} &= \mathbf{x}_i \beta_T + \epsilon_{Ti} \\ y_i &= \sum_{j=1}^T 1(t_i = v_j) y_{ji} \end{aligned}$$

As in [Introduction](#), we do not observe  $y_i$  but instead observe the endpoints  $y_{li}$  and  $y_{ui}$ .

For  $j = 1, \dots, T$ ,  $\epsilon_{ji}$  and  $\epsilon_{ti}$  are bivariate normal with mean 0 and covariance

$$\Sigma = \begin{bmatrix} \sigma^2 & \sigma \rho_{1t} \\ \sigma \rho_{1t} & 1 \end{bmatrix}$$

The treatment is exogenous if  $\rho_{1t} = 0$ . Note that we did not specify the structure of the correlations between the potential-outcome errors. We do not need information about these correlations to estimate POMs and treatment effects because all covariates and the outcome are observed in observations from each group.

From here, we discuss a model with an ordinal endogenous treatment. The results for binary treatment models are similar. The likelihood is derived in a similar manner to [Binary and ordinal endogenous covariates](#).

For  $i \in U$ , the uncensored observations, define

$$r_i = y_i - \mathbf{x}_i \boldsymbol{\beta}_j \quad \text{if } t_i = v_j$$

For the censored observations, define

$$\begin{aligned} r_{li} &= y_{li} - \mathbf{x}_i \boldsymbol{\beta}_j & \text{if } t_i = v_j \\ r_{ui} &= y_{ui} - \mathbf{x}_i \boldsymbol{\beta}_j & \text{if } t_i = v_j \end{aligned}$$

Now, the log likelihood is

$$\begin{aligned} \ln L &= \sum_{i \in U} w_i \ln \left\{ \Phi_1^* \left( l_{ti} - \frac{\rho_{1t}}{\sigma} r_i, u_{ti} - \frac{\rho_{1t}}{\sigma} r_i, 1 - \rho_{1t}^2 \right) \phi(r_i, \sigma^2) \right\} \\ &+ \sum_{i \in L} w_i \ln \Phi_2^*([l_{ti} \quad -\infty], [u_{ti} \quad r_{ui}], \boldsymbol{\Sigma}) \\ &+ \sum_{i \in R} w_i \ln \Phi_2^*([l_{ti} \quad r_{li}], [u_{ti} \quad \infty], \boldsymbol{\Sigma}) \\ &+ \sum_{i \in I} w_i \ln \Phi_2^*([l_{ti} \quad r_{li}], [u_{ti} \quad r_{ui}], \boldsymbol{\Sigma}) \end{aligned}$$

where  $U$  is the set of observations where  $y_i$  is not censored,  $L$  is the set of observations where  $y_i$  is left-censored,  $R$  is the set of observations where  $y_i$  is right-censored, and  $I$  is the set of observations where  $y_i$  is interval-censored.  $l_{ti}$  and  $u_{ti}$  are the limits for the treatment probability given in [Treatment](#) in [\[ERM\] eprobit](#).

The treatment effect  $y_{ji} - y_{1i}$  is the difference in the outcome for individual  $i$  if the individual receives the treatment  $t_i = v_j$  and what the difference would have been if the individual received the control treatment  $t_i = v_1$  instead.

The conditional POM for treatment group  $j$  is

$$\text{POM}_j(\mathbf{x}_i) = E(y_{ji} | \mathbf{x}_i) = \mathbf{x}_i \boldsymbol{\beta}_j$$

For treatment group  $j$ , the treatment effect conditioned on  $\mathbf{x}_i$  is

$$\text{TE}_j(\mathbf{x}_i) = E(y_{ji} - y_{1i} | \mathbf{x}_i) = \text{POM}_j(\mathbf{x}_i) - \text{POM}_1(\mathbf{x}_i)$$

For treatment group  $j$ , the treatment effect on the treated (TET) in group  $h$  is

$$\begin{aligned} \text{TET}_j(\mathbf{x}_i, t_i = v_h) &= E(y_{ji} - y_{1i} | \mathbf{x}_i, t_i = v_h) \\ &= \mathbf{x}_i \boldsymbol{\beta}_j - \mathbf{x}_i \boldsymbol{\beta}_1 + E(\epsilon_{ji} | \mathbf{x}_i, t_i = v_h) - E(\epsilon_{1i} | \mathbf{x}_i, t_i = v_h) \end{aligned}$$

Remembering that the outcome errors and the treatment error  $\epsilon_{ti}$  are multivariate normal, for  $j = 1, \dots, T$  we can decompose  $\epsilon_{ji}$  such that

$$\epsilon_{ji} = \sigma\rho_{1t}\epsilon_{ti} + \psi_{ji}$$

where  $\psi_{ji}$  has mean 0.

It follows that

$$\text{TET}_j(\mathbf{x}_i, t_i = v_h) = \mathbf{x}_i\beta_j - \mathbf{x}_i\beta_1$$

We can take the expectation of these conditional predictions over the covariates to get population average parameters. The `estat teffects` or `margins` commands are used to estimate the expectations as predictive margins once the model is estimated with `eintreg`. The POM for treatment group  $j$  is

$$\text{POM}_j = E(y_{ji}) = E\{\text{POM}_j(\mathbf{x}_i)\}$$

The ATE for treatment group  $j$  is

$$\text{ATE}_j = E(y_{ji} - y_{1i}) = E\{\text{TE}_j(\mathbf{x}_i)\}$$

For treatment group  $j$ , the average treatment effect on the treated (ATET) in treatment group  $h$  is

$$\text{ATET}_{jh} = E(y_{ji} - y_{1i} | t_i = v_h) = E\{\text{TET}_j(\mathbf{x}_i, t_i = v_h) | t_i = v_h\}$$

The conditional mean of  $y_i$  at treatment level  $v_j$  is

$$E(y_i | \mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_j) = \mathbf{x}_i\beta_j + E(\epsilon_i | \mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_j)$$

In *Predictions considering total effects* in [ERM] `eprobit postestimation`, we discuss how the conditional mean of  $\epsilon_i$  is calculated.

## Endogenous sample selection

### Probit endogenous sample selection

The regression for outcome  $y_i$  with selection on  $s_i$  has the form

$$\begin{aligned} y_i &= \mathbf{x}_i\beta + \epsilon_i \\ s_i &= 1 (\mathbf{z}_{si}\alpha_s + \epsilon_{si} > 0) \end{aligned}$$

where  $\mathbf{x}_i$  are covariates that affect the outcome and  $\mathbf{z}_{si}$  are covariates that affect selection. As in the *Introduction* above, we do not observe  $y_i$  but instead observe the endpoints  $y_{li}$  and  $y_{ui}$ . If  $s_i = 1$ , then the observation is selected, and there is an interval regression contribution to the likelihood. If  $s_i = 0$ , then the observation is not selected, and there is no interval regression contribution to the likelihood.

The unobserved errors  $\epsilon_i$  and  $\epsilon_{si}$  are normal with mean 0 and covariance

$$\Sigma = \begin{bmatrix} \sigma^2 & \sigma\rho_{1s} \\ \sigma\rho_{1s} & 1 \end{bmatrix}$$

The likelihood is derived in a similar manner to that in *Treatment*.

For  $i \in U$ , the uncensored and selected observations, define

$$r_i = y_i - \mathbf{x}_i \beta$$

Let

$$\begin{aligned}\mu_{s|1,i} &= \frac{\rho_{1s}}{\sigma} r_i \\ \sigma_{s|1} &= 1 - \rho_{1s}^2\end{aligned}$$

For the selection indicator  $s_i$ , the lower and upper limits on  $\epsilon_{si}$  are

$$l_{si} = \begin{cases} -\infty & s_i = 0 \\ -\mathbf{z}_{si} \alpha_s & s_i = 1 \end{cases} \quad u_{si} = \begin{cases} -\mathbf{z}_{si} \alpha_s & s_i = 0 \\ \infty & s_i = 1 \end{cases}$$

For the censored but selected observations,  $i \notin U$ , define

$$\begin{aligned}r_{li} &= y_{li} - \mathbf{x}_i \beta_j \\ r_{ui} &= y_{ui} - \mathbf{x}_i \beta_j\end{aligned}$$

Now, the log likelihood is

$$\begin{aligned}\ln L &= \sum_{i \in U} w_i \ln \left\{ \Phi_1^*(l_{si} - \mu_{s|1,i}, u_{si} - \mu_{s|1,i}, \sigma_{s|1}^2) \phi(r_i, \sigma^2) \right\} \\ &+ \sum_{i \in L} w_i \ln \Phi_2^*([l_{si} \quad -\infty], [u_{si} \quad r_{ui}], \Sigma) \\ &+ \sum_{i \in R} w_i \ln \Phi_2^*([l_{si} \quad r_{li}], [u_{si} \quad \infty], \Sigma) \\ &+ \sum_{i \in I} w_i \ln \Phi_2^*([l_{si} \quad r_{li}], [u_{si} \quad r_{ui}], \Sigma) \\ &+ \sum_{i \notin S} w_i \ln \Phi_1^*(l_{si}, u_{si}, 1)\end{aligned}$$

where  $U$  is the set of observations where  $y_i$  is not censored,  $L$  is the set of observations where  $y_i$  is left-censored,  $R$  is the set of observations where  $y_i$  is right-censored,  $I$  is the set of observations where  $y_i$  is interval-censored, and  $S$  is the set of selected observations.

The conditional mean of  $y_i$  is

$$E(y_i | \mathbf{x}_i) = \mathbf{x}_i \beta$$

## Tobit endogenous sample selection

Instead of constraining the selection indicator to be binary, tobit endogenous sample selection uses a censored continuous endogenous sample-selection indicator. We allow the selection variable to be left-censored or right-censored.



The underlying regression model for  $y_i$  with tobit selection on  $s_i$  has the form

$$y_i = \mathbf{x}_i\beta + \epsilon_i$$

We observe the selection indicator  $s_i$ , which indicates the censoring status of the latent selection variable  $s_i^*$ ,

$$s_i^* = \mathbf{z}_{si}\alpha_s + \epsilon_{si}$$

$$s_i = \begin{cases} l_i & s_i^* \leq l_i \\ s_i^* & l_i < s_i^* < u_i \\ u_i & s_i^* \geq u_i \end{cases}$$

where  $\mathbf{z}_{si}$  are covariates that affect selection, and  $l_i$  and  $u_i$  are fixed lower and upper limits.

As in [Introduction](#),  $y_i$  is observed via the endpoints  $y_{li}$  and  $y_{ui}$ . If  $s_i^*$  is not censored ( $l_i < s_i^* < u_i$ ), then the observation is selected and there is an interval regression contribution to the likelihood. Otherwise, if  $s_i^*$  is left-censored ( $s_i^* < l_i$ ) or right-censored ( $s_i^* > u_i$ ), then the observation is not selected, and there is no interval regression contribution to the likelihood. The unobserved errors  $\epsilon_i$  and  $\epsilon_{si}$  are normal with mean 0 and covariance

$$\Sigma = \begin{bmatrix} \sigma^2 & \sigma_{1s} \\ \sigma_{1s} & \sigma_s^2 \end{bmatrix}$$

For the selected observations, we can treat  $s_i$  as a continuous endogenous regressor, as in [Continuous endogenous covariates](#). In fact,  $s_i$  may even be used as a regressor for  $y_i$  in [eintreg](#) (specify `tobitselect(..., main)`). On the nonselected observations, we treat  $s_i$  like the probit endogenous sample-selection indicator in [Probit endogenous sample selection](#).

Conditional on  $s_i^*$  and the exogenous covariates,  $\epsilon_i$  has mean and variance

$$\mu_{1|s,i} = E(\epsilon_i | s_i^*, \mathbf{x}_i, \mathbf{z}_{si}) = \sigma_{1s} \sigma_s^{-2} (s_i^* - \mathbf{z}_{si}\alpha_s)$$

$$\sigma_{1|s}^2 = \text{Var}(\epsilon_i | s_i^*, \mathbf{x}_i, \mathbf{z}_{si}) = \sigma^2 - \sigma_{1s} \sigma_s^{-2} \sigma_{1s}$$

Let

$$r_{li} = y_{li} - \mathbf{x}_i\beta - \mu_{1|s,i}$$

$$r_{ui} = y_{ui} - \mathbf{x}_i\beta - \mu_{1|s,i}$$

The log likelihood is

$$\begin{aligned}
 \ln L = & \sum_{i \in U} w_i \ln \phi(r_{li}, \sigma_{1|s}^2) \\
 & + \sum_{i \in L} w_i \ln \Phi_1^*(-\infty, r_{ui}, \sigma_{1|s}^2) \\
 & + \sum_{i \in R} w_i \ln \Phi_1^*(r_{li}, \infty, \sigma_{1|s}^2) \\
 & + \sum_{i \in I} w_i \ln \Phi_1^*(r_{li}, r_{ui}, \sigma_{1|s}^2) \\
 & + \sum_{i \in S} w_i \ln \phi(s_i - \mathbf{z}_{si} \boldsymbol{\alpha}_s, \sigma_s^2) \\
 & + \sum_{i \in L_n} w_i \ln \Phi_1^*(l_{li}, u_{li}, 1) \\
 & + \sum_{i \in R_n} w_i \ln \Phi_1^*(l_{ui}, u_{ui}, 1)
 \end{aligned}$$

where  $S$  is the set of observations for which  $y_{li}$  and  $y_{ui}$  are observed,  $U \subset S$  is the set of observations where  $y_i$  is not censored,  $L \subset S$  is the set of observations where  $y_i$  is left-censored,  $R \subset S$  is the set of observations where  $y_i$  is right-censored,  $I \subset S$  is the set of observations where  $y_i$  is interval-censored,  $L_n$  is the set of observations for which  $s_i^*$  is left-censored, and  $R_n$  is the set of observations for which  $s_i^*$  is right-censored. The lower and upper limits for selection— $l_{li}$ ,  $u_{li}$ ,  $l_{ui}$ , and  $u_{ui}$ —are defined in [Tobit endogenous sample selection](#) in [ERM] [eprobit](#).

When  $s_i$  is not a covariate in  $\mathbf{x}_i$ , we use the standard conditional mean formula,

$$E(y_i | \mathbf{x}_i) = \mathbf{x}_i \boldsymbol{\beta}$$

Otherwise, we use

$$E(y_i | \mathbf{x}_i, s_i, z_{si}) = \mathbf{x}_i \boldsymbol{\beta} + \frac{\sigma_{1s}}{\sigma_s^2} (s_i - z_{si} \boldsymbol{\alpha}_s)$$

## Combinations of features

Extended interval regression models that involve multiple features can be formulated using the techniques discussed in [Likelihood for multiequation models](#) in [ERM] [eprobit](#). Essentially, the density of the observed endogenous covariates can be written in terms of the unobserved normal errors. The observed endogenous and exogenous covariates determine the range of the errors, and the joint density can be evaluated as multivariate normal probabilities and densities.

## Confidence intervals

The estimated variances will always be nonnegative, and the estimated correlations will always fall in  $(-1, 1)$ . To obtain confidence intervals that accommodate these ranges, we must use transformations.

We use the log transformation to obtain the confidence intervals for variance parameters and the atanh transformation to obtain confidence intervals for correlation parameters. For details, see [Confidence intervals](#) in [ERM] [eprobit](#).

## References

- Angrist, J. D. 2001. Estimation of limited dependent variable models with dummy endogenous regressors: Simple strategies for empirical practice. *Journal of Business and Economic Statistics* 19: 2–16.
- Cameron, A. C., and P. K. Trivedi. 2010. *Microeconometrics Using Stata*. Rev. ed. College Station, TX: Stata Press.
- Fernandez-Cornejo, J., and S. Wechsler. 2012. Revisiting the Impact of Bt Corn Adoption by U.S. Farmers. *Agricultural and Resource Economics Review* 41: 377–390.
- Mairesse, J., and S. Robin. 2009. Innovation and productivity: A firm-level analysis for French manufacturing and services using CIS3 and CIS4 data (1998–2000 and 2002–2004). CREST-ENSAE. [congres.afse.fr/docs/2010/543572jmsr\\_ep2009.pdf](https://congres.afse.fr/docs/2010/543572jmsr_ep2009.pdf).
- Newey, W. K. 1987. Efficient estimation of limited dependent variable models with endogenous explanatory variables. *Journal of Econometrics* 36: 231–250.
- Roodman, D. 2011. Fitting fully observed recursive mixed-process models with `cmp`. *Stata Journal* 11: 159–206.
- Tobin, J. 1958. Estimation of relationships for limited dependent variables. *Econometrica* 26: 24–36.
- Wooldridge, J. M. 2016. *Introductory Econometrics: A Modern Approach*. 6th ed. Boston: Cengage.

## Also see

- [ERM] **eintreg postestimation** — Postestimation tools for `eintreg`
- [ERM] **eintreg predict** — `predict` after `eintreg`
- [ERM] **estat teffects** — Average treatment effects for extended regression models
- [ERM] **intro 8** — Conceptual introduction via worked example
- [R] **intreg** — Interval regression
- [R] **ivtobit** — Tobit model with continuous endogenous covariates
- [R] **tobit** — Tobit regression
- [SVY] **svy estimation** — Estimation commands for survey data
- [U] **20 Estimation and postestimation commands**

Postestimation commands  
Methods and formulas

predict  
Also see

margins

Remarks and examples

Postestimation commands

The following postestimation command is of special interest after `eintreg`:

Command	Description
<code>estat teffects</code>	treatment effects and potential-outcome means

The following standard postestimation commands are also available after `eintreg`:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of estimates
<code>estat ic</code>	Akaike’s and Schwarz’s Bayesian information criteria (AIC and BIC)
<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>forecast</code>	dynamic forecasts and simulations
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
* <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 coefficients
<code>predict</code>	predictions, residuals, influence statistics, and other diagnostic measures
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of estimates
<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

\* `forecast` and `lrtest` are not appropriate with `svy` estimation results.

# predict

Predictions after `eintreg` are described in

- [ERM] **eintreg predict**      predict after eintreg
- [ERM] **predict treatment**    predict for treatment statistics
- [ERM] **predict advanced**    predict's advanced features

[ERM] **eintreg predict** describes the most commonly used predictions. If you fit a model with treatment effects, predictions specifically related to these models are detailed in [ERM] **predict treatment**. [ERM] **predict advanced** describes less commonly used predictions, such as predictions of outcomes in auxiliary equations.

# margins

## Description for margins

`margins` estimates margins of response for means, probabilities, potential-outcome means, treatment effects, and linear predictions.

## Menu for margins

Statistics > Postestimation

## Syntax for margins

```
margins [marginlist] [ , options ]
margins [marginlist] , predict(statistic ...) [predict(statistic ...) ...] [options]
```

statistic	Description
Main	
<u>mean</u>	mean; the default
<u>pr</u>	probability for binary or ordinal $y_j$
<u>pomean</u>	potential-outcome mean
<u>te</u>	treatment effect
<u>tet</u>	treatment effect on the treated
<u>xb</u>	linear prediction
<u>pr</u> ( $a,b$ )	$\Pr(a < y_j < b)$ for continuous $y_j$
<u>e</u> ( $a,b$ )	$E(y_j   a < y_j < b)$ for continuous $y_j$
<u>y</u> <u>star</u> ( $a,b$ )	$E(y_j^*), y_j^* = \max\{a, \min(y_j, b)\}$ for continuous $y_j$

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.  
For the full syntax, see [R] **margins**.

## Remarks and examples

See [ERM] [intro 6](#) for an overview of using margins and `predict` after `eintreg`. For examples using `margins`, `predict`, and `estat teffects`, see *Interpreting effects* in [ERM] [intro 8](#) and see [ERM] [example 1a](#).

## Methods and formulas

Counterfactual predictions and inferences for the underlying model in interval regression can be evaluated as in a linear regression model. These predictions and effects are described in *Methods and formulas* of [ERM] [eregress postestimation](#). Methods and formulas for all other predictions are given in *Methods and formulas* of [ERM] [eintreg](#).

## Also see

[ERM] [eintreg](#) — Extended interval regression

[ERM] [eintreg predict](#) — `predict` after `eintreg`

[ERM] [predict treatment](#) — `predict` for treatment statistics

[ERM] [predict advanced](#) — `predict`'s advanced features

[ERM] [eprobit postestimation](#) — Postestimation tools for `eprobit`

[U] [20 Estimation and postestimation commands](#)

Description	Syntax
Options for statistics	Options for how results are calculated
Remarks and examples	Methods and formulas
Also see	

Description

In this entry, we show how to create new variables containing observation-by-observation predictions after fitting a model with `eintreg`.

Syntax

You previously fit the model

```
eintreg y1 yu x1 ..., ...
```

The equation specified immediately after the `eintreg` command is called the main equation. It is

$$y_i = \beta_0 + \beta_1 x1_i + \cdots + e_i.y$$

where  $y1_i \leq y_i \leq yu_i$ .

`predict` calculates predictions for `y` in the main equation. The other equations in the model are called auxiliary equations or complications.

The syntax of `predict` is

```
predict [type] newvar [if] [in] [, stdstatistics howcalculated]
```

<i>stdstatistics</i>	Description
<code>mean</code>	linear prediction; the default
<code>xb</code>	linear prediction excluding all complications
<code>ystar(a,b)</code>	$E(y*_j), y*_j = \max\{a, \min(y_j, b)\}$

*a* and *b* are numeric values, missing (`.`), or variable names.

<i>howcalculated</i>	Description
<code>default</code>	not fixed; base values from data
<code>fix(endogvars)</code>	fix specified endogenous covariates
<code>base(valspecs)</code>	specify base values of any variables
<code>target(valspecs)</code>	more convenient way to specify <code>fix()</code> and <code>base()</code>

Note: The `fix()` and `base()` options affect results only in models with endogenous variables in the main equation. The `target()` option is sometimes a more convenient way to specify the `fix()` and `base()` options.

*endogvars* are names of one or more endogenous variables appearing in the main equation.

*valspecs* specify the values for variables at which predictions are to be evaluated. Each *valspec* is of the form

*varname* = #

*varname* = (*exp*)

*varname* = *othervarname*

For instance, `base(valspecs)` could be `base(w1=0)` or `base(w1=0 w2=1)`.

Notes:

- (1) `predict` can also calculate treatment-effect statistics. See [\[ERM\] predict treatment](#).
- (2) `predict` can also make predictions for the other equations in addition to the main-equation predictions discussed here. See [\[ERM\] predict advanced](#).

## Options for statistics

`mean` specifies that the linear prediction be calculated. In each observation, the linear prediction is the expected value of the dependent variable  $y$  conditioned on the covariates. Results depend on how complications are handled, which is determined by the [howcalculated](#) options.

`xb` specifies that the linear prediction be calculated ignoring all complications. This prediction corresponds to what would be observed in data in which all the covariates in the main equation were exogenous.

`ystar(a, b)` specifies that the linear prediction be censored between  $a$  and  $b$ . If  $a$  is missing (`.`), then  $a$  is treated as  $-\infty$ . If  $b$  is missing (`.`), then  $b$  is treated as  $+\infty$ .  $a$  and  $b$  can be specified as numeric values, missing (`.`), or variable names.

## Options for how results are calculated

By default, predictions are calculated taking into account all complications. This is discussed in [Remarks and examples](#) of [\[ERM\] eregress predict](#).

`fix(varname ...)` specifies a list of endogenous variables from the main equation to be treated as if they were exogenous. This was discussed in [\[ERM\] intro 3](#) and is discussed further in [Remarks and examples](#) of [\[ERM\] eregress predict](#).

`base(varname = ...)` specifies a list of variables from any equation and values for them. Those values will be used in calculating the expected value of  $e_i.y$ . Errors from other equations spill over into the main equation because of correlations between errors. The correlations were estimated when the model was fit. The amount of spillover depends on those correlations and the values of the errors. This issue was discussed in [\[ERM\] intro 3](#) and is discussed further in [Remarks and examples](#) of [\[ERM\] eregress predict](#).

`target(varname = ...)` is sometimes a more convenient way to specify the `fix()` and `base()` options. You specify a list of variables from the main equation and values for them. Those values override the values of the variables calculating  $\beta_0 + \beta_1 x_{1i} + \dots$ . Use of `target()` is discussed in [Remarks and examples](#) of [\[ERM\] eregress predict](#).



## Remarks and examples

Predictions after fitting models with **eintreg** are handled the same as they are after fitting models with **eregress**. The issues are the same. See [\[ERM\] eregress predict](#).

Note that censoring is treated as a nuisance in **eintreg**. Predicted values are not  $y_l$  and  $y_u$ , they are  $y$ .

## Methods and formulas

See *Methods and formulas* in [\[ERM\] eintreg postestimation](#).

## Also see

[\[ERM\] eintreg postestimation](#) — Postestimation tools for eintreg

[\[ERM\] eintreg](#) — Extended interval regression

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`eoprobit` fits an ordered probit regression model that accommodates any combination of endogenous covariates, nonrandom treatment assignment, and endogenous sample selection. Continuous, binary, and ordinal endogenous covariates are allowed. Treatment assignment may be endogenous or exogenous. A probit or tobit model may be used to account for endogenous sample selection.

## Quick start

Ordered probit regression of `y` on `x` with continuous endogenous covariate `y2` modeled by `x` and `z`

```
eoprobit y x, endogenous(y2 = x z)
```

As above, but adding continuous endogenous covariate `y3` modeled by `x` and `z2`

```
eoprobit y x, endogenous(y2 = x z) endogenous(y3 = x z2)
```

Ordered probit regression of `y` on `x` with binary endogenous covariate `d` modeled by `x` and `z`

```
eoprobit y x, endogenous(d = x z, probit)
```

Ordered probit regression of `y` on `x` with endogenous treatment recorded in `trtvar` and modeled by `x` and `z`

```
eoprobit y x, entreat(trtvar = x z)
```

Ordered probit regression of `y` on `x` with exogenous treatment recorded in `trtvar`

```
eoprobit y x, extreat(trtvar)
```

Ordered probit regression of `y` on `x` with endogenous sample-selection indicator `selvar` modeled by `x` and `z`

```
eoprobit y x, select(selvar = x z)
```

As above, but adding endogenous covariate `y2` modeled by `x` and `z2`

```
eoprobit y x, select(selvar = x z) endogenous(y2 = x z2)
```

As above, but adding endogenous treatment recorded in `trtvar` and modeled by `x` and `z3`

```
eoprobit y x, select(selvar = x z) endogenous(y2 = x z2) ///
    entreat(trtvar = x z3)
```

## Menu

Statistics > Endogenous covariates > Models adding selection and treatment > Ordered probit regression

## Syntax

*Basic ordered probit regression with endogenous covariates*

```
eoprobit depvar [indepvars] ,  
      endogenous(depvarsen = varlisten) [options]
```

*Basic ordered probit regression with endogenous treatment assignment*

```
eoprobit depvar [indepvars] ,  
      entreat(depvartr [= varlisttr]) [options]
```

*Basic ordered probit regression with exogenous treatment assignment*

```
eoprobit depvar [indepvars] ,  
      extreat(tvar) [options]
```

*Basic ordered probit regression with sample selection*

```
eoprobit depvar [indepvars] ,  
      select(depvars = varlists) [options]
```

*Basic ordered probit regression with tobit sample selection*

```
eoprobit depvar [indepvars] ,  
      tobitselect(depvars = varlists) [options]
```

*Ordered probit regression combining endogenous covariates, treatment, and selection*

```
eoprobit depvar [indepvars] [if] [in] [weight] [, extensions options]
```

<i>extensions</i>	Description
Model	
<u>endogenous</u> ( <i>enspec</i> )	model for endogenous covariates; may be repeated
<u>entreat</u> ( <i>entrspec</i> )	model for endogenous treatment assignment
<u>extreat</u> ( <i>extrspec</i> )	exogenous treatment
<u>select</u> ( <i>selspec</i> )	probit model for selection
<u>tobitselect</u> ( <i>tselspec</i> )	tobit model for selection
<hr/>	
<i>options</i>	Description
Model	
<u>offset</u> ( <i>varname<sub>o</sub></i> )	include <i>varname<sub>o</sub></i> in model with coefficient constrained to 1
<u>constraints</u> ( <i>numlist</i> )	apply specified linear constraints
<u>collinear</u>	keep collinear variables
SE/Robust	
<u>vce</u> ( <i>vcetype</i> )	<i>vcetype</i> may be <code>oim</code> , <u>robust</u> , <u>cluster</u> <i>clustvar</i> , <code>opg</code> , <u>bootstrap</u> , or <u>jackknife</u>
Reporting	
<u>level</u> (#)	set confidence level; default is <code>level(95)</code>
<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> (#)	set the number of integration (quadrature) points for integration over four or more dimensions; default is <code>intpoints(128)</code>
<u>triintpoints</u> (#)	set the number of integration (quadrature) points for integration over three dimensions; default is <code>triintpoints(10)</code>
Maximization	
<u>maximize_options</u>	control the maximization process; seldom used
<u>coeflegend</u>	display legend instead of statistics

*enspec* is `depvarsen = varlisten [ , enopts ]`

where *depvars<sub>en</sub>* is a list of endogenous covariates. Each variable in *depvars<sub>en</sub>* specifies an endogenous covariate model using the common *varlist<sub>en</sub>* and options.

*entrspec* is `depvartr [= varlisttr] [ , tropts ]`

where *depvar<sub>tr</sub>* is a variable indicating treatment assignment. *varlist<sub>tr</sub>* is a list of covariates predicting treatment assignment.

*extrspec* is `tvar [ , nomain nocutsinteract nointeract ]`

where *tvar* is a variable indicating treatment assignment.

*selspec* is `depvars = varlists [ , noconstant offset(varnameo) ]`

where *depvar<sub>s</sub>* is a variable indicating selection status. *depvar<sub>s</sub>* must be coded as 0, indicating that the observation was not selected, or 1, indicating that the observation was selected. *varlist<sub>s</sub>* is a list of covariates predicting selection.

*tselspec* is `depvars = varlists [ , tseopts ]`

where *depvar<sub>s</sub>* is a continuous variable. *varlist<sub>s</sub>* is a list of covariates predicting *depvar<sub>s</sub>*. The censoring status of *depvar<sub>s</sub>* indicates selection, where a censored *depvar<sub>s</sub>* indicates that the observation was not selected and a noncensored *depvar<sub>s</sub>* indicates that the observation was selected.

<i>enopts</i>	Description
Model	
<u>probit</u>	treat endogenous covariate as binary
<u>oprobit</u>	treat endogenous covariate as ordinal
<u>nomain</u>	do not add endogenous covariate to main equation
<u>noconstant</u>	suppress constant term

<i>tropts</i>	Description
Model	
<u>nomain</u>	do not add treatment indicator to main equation
<u>nocutsinteract</u>	do not interact treatment with cutpoints
<u>nointeract</u>	do not interact treatment with covariates in main equation
<u>noconstant</u>	suppress constant term
<u>offset</u> (varname <sub>o</sub> )	include varname <sub>o</sub> in model with coefficient constrained to 1

<i>tseopts</i>	Description
Model	
<u>ll</u> (varname   #)	left-censoring variable or limit
<u>ul</u> (varname   #)	right-censoring variable or limit
<u>main</u>	add censored selection variable to main equation
<u>noconstant</u>	suppress constant term
<u>offset</u> (varname <sub>o</sub> )	include varname <sub>o</sub> in model with coefficient constrained to 1

*indepvars*, *varlist<sub>en</sub>*, *varlist<sub>tr</sub>*, and *varlist<sub>s</sub>* may contain factor variables; see [U 11.4.3 Factor variables].  
*depvar*, *indepvars*, *depvars<sub>en</sub>*, *varlist<sub>en</sub>*, *depvar<sub>tr</sub>*, *varlist<sub>tr</sub>*, *tvar*, *depvar<sub>s</sub>*, and *varlist<sub>s</sub>* may contain time-series operators; see [U 11.4.4 Time-series varlists].  
*bootstrap*, *by*, *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*.  
*vce()* and weights are not allowed with the *svy* prefix; see [SVY] *svy*.  
*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

`endogenous(enspec)`, `entreat(entrspec)`, `extreat(extrspec)`, `select(selspec)`,  
`tobitselect(tselspec)`; see [\[ERM\] erm options](#).  
`offset(varnameo)`, `constraints(numlist)`, `collinear`; see [\[R\] estimation options](#).

### SE/Robust

`vce(vcetype)`; see [\[ERM\] erm options](#).

### Reporting

`level(#)`, `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(#)`, `triintpoints(#)`; see [\[ERM\] erm 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](#).

Setting the optimization type to `technique(bhhh)` resets the default `vcetype` to `vce(opg)`.

The following option is available with `eoprobit` but is not shown in the dialog box:

`coeflegend`; see [\[R\] estimation options](#).

## Remarks and examples

`eoprobit` fits models that we refer to as “extended ordered probit regression models”, meaning that they accommodate endogenous covariates, nonrandom treatment assignment, and endogenous sample selection. `eoprobit` can account for these complications whether they arise individually or in combination.

In this entry, you will find information on the `eoprobit` command syntax. You can see [Methods and formulas](#) for a full description of the models that can be fit with `eoprobit` and details about how those models are fit.

More information on extended ordered probit models is found in the separate introductions and example entries. We recommend reading those entries to learn how to use `eoprobit`. Below, we provide a guide to help you locate the ones that will be helpful to you.

For an introduction to `eoprobit` and the other extended regression commands (`eregress`, `eintreg`, and `eprobit`), see [ERM] [intro 1](#)–[ERM] [intro 8](#).

[ERM] [intro 1](#) introduces the ERM commands, the problems they address, and their syntax.

[ERM] [intro 2](#) provides background on the four types of models—linear regression, interval regression, probit regression, and ordered probit regression—that can be fit using ERM commands.

[ERM] [intro 3](#) considers the problem of endogenous covariates and how to solve it using ERM commands.

[ERM] [intro 4](#) gives an overview of endogenous sample selection and using ERM commands to account for it when fitting a linear, interval, probit, or ordered probit model.

[ERM] [intro 5](#) covers nonrandom treatment assignment and how to account for it using `eoprobit` or any of the other ERM commands.

[ERM] [intro 6](#) discusses interpretation of results. You can interpret coefficients from `eoprobit` in the usual way, but this introduction goes beyond the interpretation of coefficients. We demonstrate how to find answers to interesting questions by using `margins`. If your model includes an endogenous covariate or an endogenous treatment, the use of `margins` differs from its use after other estimation commands, so we strongly recommend reading this intro if you are fitting these types of models.

[ERM] [intro 7](#) will be particularly helpful if you are familiar with `heckoprobit` and other commands that address endogenous covariates, sample selection, or nonrandom treatment assignment. This introduction is a Rosetta stone that maps the syntax of those commands to the syntax of `eoprobit`.

[ERM] [intro 8](#) walks you through an example that gives insight into the concepts of endogenous covariates, treatment assignment, and sample selection while fitting models with `eregress` that address these complications. Although the example uses `eregress`, the discussion applies equally to `eoprobit`. This intro also demonstrates how to interpret results by using `margins` and `estat teffects`.

Additional examples are presented in [ERM] [example 1a](#)–[ERM] [example 6b](#). For examples using `eoprobit`, see

[ERM] <a href="#">example 6a</a>	Ordered probit regression with endogenous treatment
[ERM] <a href="#">example 6b</a>	Ordered probit regression with endogenous covariate and treatment

See [Examples](#) in [ERM] [intro](#) for an overview of all the examples. These examples demonstrate all four extended regression commands, and all may be interesting because they handle complications in the same way.

You can also find in literature discussion and examples of many models that `eoprobit` can fit. For instance, `eoprobit` can be used to fit models like the ordered probit model with endogenous sample selection discussed in [De Luca and Perotti \(2011\)](#) and the ordered probit models with continuous or binary endogenous covariates discussed in [Wooldridge \(2010, sec. 16.3.3\)](#). [Roodman \(2011\)](#) investigated ordered probit models with endogenous covariates and endogenous sample selection, and demonstrated how multiple observational data complications could be addressed with a triangular model structure. His work has been used to model processes like the effect of living with a child on the happiness of the elderly ([Chyi and Mao 2012](#)) and the effect of parental migration on child education ([Botezat and Pfeiffer 2014](#)).

## Stored results

eoprobit stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(N_selected)</code>	number of uncensored observations
<code>e(N_nonselected)</code>	number of censored observations
<code>e(k)</code>	number of parameters
<code>e(k_cat#)</code>	number of categories for the <i>#th depvar</i> , ordinal
<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_dv)</code>	number of dependent variables
<code>e(k_aux)</code>	number of auxiliary parameters
<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>	$\chi^2$
<code>e(p)</code>	significance
<code>e(n_quad)</code>	number of integration points for multivariate normal
<code>e(n_quad3)</code>	number of integration points for trivariate normal
<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>	eoprobit
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	names of dependent variables
<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(offset#)</code>	offset for the <i>#th depvar</i>
<code>e(chi2type)</code>	Wald; type of model $\chi^2$ test
<code>e(vce)</code>	<i>vcetype</i> 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(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(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(marginsdefault)</code>	default <code>predict()</code> specification for <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(cat#)</code>	categories for the <i>#th depvar</i> , ordinal
<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
------------------------	-------------------------



## Methods and formulas

The methods and formulas presented here are for the ordered probit-outcome model. They are a special case of the methods and formulas derived by [Roodman \(2011\)](#) and extend the methods and formulas for extended probit regressions described in [Methods and formulas](#) of [\[ERM\] eoprobit](#) to extended ordered probit regression models.

Methods and formulas are presented under the following headings:

- [Introduction](#)
- [Endogenous covariates](#)
  - [Continuous endogenous covariates](#)
  - [Binary and ordinal endogenous covariates](#)
- [Treatment](#)
- [Endogenous sample selection](#)
  - [Probit endogenous sample selection](#)
  - [Tobit endogenous sample selection](#)
- [Combinations of features](#)
- [Confidence intervals](#)

## Introduction

An ordered probit regression of outcome  $y_i$  on covariates  $\mathbf{x}_i$  may be written as

$$y_i = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i \leq \kappa_h$$

The values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_H$  is taken as  $+\infty$ . The unobserved error  $\epsilon_i$  is standard normal.

The log likelihood for this model is

$$\ln L = \sum_{i=1}^N w_i \ln \left[ \begin{aligned} &1(y_i = v_1)\Phi(-\mathbf{x}_i\boldsymbol{\beta}) \\ &+ \sum_{h=2}^{H-1} 1(y_i = v_h) \{ \Phi(\kappa_h - \mathbf{x}_i\boldsymbol{\beta}) - \Phi(\kappa_{h-1} - \mathbf{x}_i\boldsymbol{\beta}) \} \\ &+ 1(y_i = v_H)\Phi(\mathbf{x}_i\boldsymbol{\beta}) \end{aligned} \right]$$

where  $w_i$  are the weights.

For  $h = 0, \dots, H$ , define

$$c_{ih} = \begin{cases} -\infty & h = 0 \\ \kappa_h - \mathbf{x}_i\boldsymbol{\beta} & h = 1, \dots, H-1 \\ \infty & h = H \end{cases} \quad (1)$$

This leads to the limits

$$l_{1i} = c_{i(h-1)} \quad \text{if} \quad y_i = v_h \quad (2)$$

and

$$u_{1i} = c_{ih} \quad \text{if} \quad y_i = v_h \quad (3)$$

These are limits on the unobserved  $\epsilon_i$  based on the observed values of  $y_i$  and  $\mathbf{x}_i$ . They let us rewrite the log likelihood concisely as

$$\ln L = \sum_{i=1}^N w_i \ln \Phi_1^*(l_{1i}, u_{1i}, 1)$$

The conditional probabilities of success can be written using similar notation. For  $h = 1, \dots, H$ ,

$$\Pr(y_i = v_h | \mathbf{x}_i) = \Phi_1^*(c_{i(h-1)}, c_{ih}, 1) \quad (4)$$

If you are willing to take our word for some derivations and notation, the following is complete. Longer explanations and derivations for some terms and functions are provided in [Methods and formulas](#) of [ERM] **eoprobit**. For example, we need the two-sided probability function  $\Phi_d^*$  that is discussed in [Introduction](#) in [ERM] **eoprobit**.

If you are interested in all the details, we suggest you read [Methods and formulas](#) of [ERM] **eoprobit** in its entirety before reading this section. Here, we mainly show how the complications that arise in ERMs are handled in an ordered probit framework.

## Endogenous covariates

### Continuous endogenous covariates

An ordered probit regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and  $C$  continuous endogenous covariates  $\mathbf{w}_{ci}$  has the form

$$y_i = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{x}_i \boldsymbol{\beta} + \mathbf{w}_{ci} \boldsymbol{\beta}_c + \epsilon_i \leq \kappa_h$$

$$\mathbf{w}_{ci} = \mathbf{z}_{ci} \mathbf{A}_c + \boldsymbol{\epsilon}_{ci}$$

The values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_H$  is taken as  $+\infty$ . The vector  $\mathbf{z}_{ci}$  contains variables from  $\mathbf{x}_i$  and other covariates that affect  $\mathbf{w}_{ci}$ . The unobserved errors  $\epsilon_i$  and  $\boldsymbol{\epsilon}_{ci}$  are multivariate normal with mean 0 and covariance

$$\begin{bmatrix} 1 & \boldsymbol{\sigma}'_{1c} \\ \boldsymbol{\sigma}_{1c} & \boldsymbol{\Sigma}_c \end{bmatrix}$$

As in [Continuous endogenous covariates](#) in [ERM] **eoprobit**, the likelihood can be written using the conditional density of  $\epsilon_i$  on  $\mathbf{w}_{ci}$ .

Now, for  $h = 0, \dots, H$ , define

$$c_{ih} = \begin{cases} -\infty & h = 0 \\ \kappa_h - \mathbf{x}_i \boldsymbol{\beta} - \boldsymbol{\sigma}'_{1c} \boldsymbol{\Sigma}_c^{-1} (\mathbf{w}_{ci} - \mathbf{z}_{ci} \mathbf{A}_c)' & h = 1, \dots, H-1 \\ \infty & h = H \end{cases}$$

These expressions used the conditional mean of  $\epsilon_i$ . The lower and upper limits for the  $y_i$  probability are

$$l_{1i} = c_{i(h-1)} \quad \text{if} \quad y_i = v_h$$

and

$$u_{1i} = c_{ih} \quad \text{if} \quad y_i = v_h$$

Using these limits, the conditional variance, and the conditional density of  $\mathbf{w}_{ci}$ , we obtain the log likelihood

$$\ln L = \sum_{i=1}^N w_i \left\{ \ln \Phi_1^* (l_{1i}, u_{1i}, 1 - \sigma'_{1c} \Sigma_c^{-1} \sigma_{1c}) + \ln \phi_C(\mathbf{w}_{ci} - \mathbf{z}_{ci} \mathbf{A}_c, \Sigma_c) \right\}$$

The conditional probabilities of success can be written using similar notation. For  $h = 1, \dots, H$ ,

$$\Pr(y_i = v_h | \mathbf{x}_i) = \Phi_1^*(c_{i(h-1)}, c_{ih}, 1 - \sigma'_{1c} \Sigma_c^{-1} \sigma_{1c})$$

## Binary and ordinal endogenous covariates

Here, we begin by formulating the ordered probit regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and  $B$  binary and ordinal endogenous covariates  $\mathbf{w}_{bi} = [w_{b1i}, \dots, w_{bBi}]$ . Indicator (dummy) variables for the levels of each binary and ordinal covariate are used in the model. You can also interact other covariates with the binary and ordinal endogenous covariates, as in treatment-effect models.

The binary and ordinal endogenous covariates  $\mathbf{w}_{bi}$  are formulated as in [Binary and ordinal endogenous covariates](#) in [\[ERM\] eoprobit](#). So we have

$$y_i = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{x}_i \beta + \mathbf{wind}_{b1i} \beta_{b1} + \dots + \mathbf{wind}_{bBi} \beta_{bB} + \epsilon_i \leq \kappa_h$$

where the values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_H$  is taken as  $+\infty$ . The  $\mathbf{wind}_{bj_i}$  vectors are defined in [Binary and ordinal endogenous covariates](#) in [\[ERM\] eoprobit](#). The outcome error  $\epsilon_i$  and binary and ordinal endogenous errors  $\epsilon_{b1i}, \dots, \epsilon_{bBi}$  are multivariate normal with mean 0 and covariance

$$\Sigma = \begin{bmatrix} 1 & \rho'_{1b} \\ \rho_{1b} & \Sigma_b \end{bmatrix}$$

From here, we discuss the model with ordinal endogenous covariates. The results for binary endogenous covariates are similar.

Now, for  $h = 0, \dots, H$ , define

$$c_{ih} = \begin{cases} -\infty & h = 0 \\ \kappa_h - \mathbf{x}_i \beta - \mathbf{wind}_{b1i} \beta_{b1} - \dots - \mathbf{wind}_{bBi} \beta_{bB} & h = 1, \dots, H-1 \\ \infty & h = H \end{cases}$$

The lower and upper limits for the  $y_i$  probability are

$$l_{1i} = c_{i(h-1)} \quad \text{if} \quad y_i = v_h$$

and

$$u_{1i} = c_{ih} \quad \text{if} \quad y_i = v_h$$

Let

$$\mathbf{l}_i = [l_{1i} \quad l_{b1i} \quad \dots \quad l_{bBi}]$$

$$\mathbf{u}_i = [u_{1i} \quad u_{b1i} \quad \dots \quad u_{bBi}]$$

where the  $l_{bji}$  and  $u_{bji}$  are the lower and upper limits for the binary and ordinal endogenous covariate probabilities. They are defined in *Binary and ordinal endogenous covariates* in [ERM] **eoprobit**.

So the log likelihood for this model is

$$\ln L = \sum_{i=1}^N w_i \ln \Phi_{B+1}^*(\mathbf{l}_i, \mathbf{u}_i, \boldsymbol{\Sigma})$$

Now, let

$$\mathbf{l}_{bi} = [l_{b1i} \quad \dots \quad l_{bBi}]$$

$$\mathbf{u}_{bi} = [u_{b1i} \quad \dots \quad u_{bBi}]$$

$$\mathbf{l}_{ih1} = [c_{i(h-1)} \quad \mathbf{l}_{bi}]$$

$$\mathbf{u}_{ih1} = [c_{ih} \quad \mathbf{u}_{bi}]$$

The conditional probabilities are

$$\Pr(y_i = v_h | \mathbf{x}_i, \mathbf{z}_{b1i}, \dots, \mathbf{z}_{bBi}, \mathbf{w}_{bi}) = \frac{\Phi_{B+1}^*(\mathbf{l}_{ih1}, \mathbf{u}_{ih1}, \boldsymbol{\Sigma})}{\Phi_B^*(\mathbf{l}_{bi}, \mathbf{u}_{bi}, \boldsymbol{\Sigma}_b)}$$

## Treatment

In the potential-outcomes framework, the treatment  $t_i$  is a discrete variable taking  $T$  values, indexing the  $T$  potential outcomes of the outcome  $y_i$ :  $y_{1i}, \dots, y_{Ti}$ .

When we observe treatment  $t_i$  with levels  $v_1, \dots, v_T$ , we have

$$y_i = \sum_{j=1}^T 1(t_i = v_{tj}) y_{ji}$$

So for each observation, we only observe the potential outcome associated with that observation's treatment value.

For exogenous treatments, our approach is equivalent to the regression adjustment treatment-effect estimation method. See [TE] **teffects intro advanced**. We do not model the treatment assignment process. The formulas for the treatment effects and potential-outcome means (POMs) are equivalent to what we provide here for endogenous treatments. The treatment effect on the treated for  $\mathbf{x}_i$  for an exogenous treatment is equivalent to what we provide here for the endogenous treatment when the correlation parameter between the outcome and treatment errors is set to 0. The average treatment effects (ATES) and POMs for exogenous treatments are estimated as predictive margins in an analogous manner to what we describe here for endogenous treatments.

From here, we assume an endogenous treatment  $t_i$ . As in *Treatment* in [ERM] **eoprobit**, we model the treatment assignment process with a probit or ordered probit model, and we call the treatment assignment error  $\epsilon_{ti}$ . An ordered probit regression of  $y_i$  on treatment  $t_i$  with levels  $v_{t1}, \dots, v_{tT}$  has the form

$$y_i = \sum_{j=1}^T 1(t_i = v_{tj}) y_{ji}$$

where for  $j = 1, \dots, T$  and exogenous covariates  $\mathbf{x}_i$

$$y_{ji} = v_h \quad \text{iff} \quad \kappa_{(h-1)j} < \mathbf{x}_i \beta_j + \epsilon_{ji} \leq \kappa_{hj}$$

The values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ . For  $j = 1, \dots, T$ ,  $\kappa_{0j}$  is taken as  $-\infty$  and  $\kappa_{Hj}$  is taken as  $+\infty$ . For  $j = 1, \dots, T$ ,  $\epsilon_{ji}$  and  $\epsilon_{ti}$  are bivariate normal with mean 0 and covariance

$$\Sigma = \begin{bmatrix} 1 & \rho_{1t} \\ \rho_{1t} & 1 \end{bmatrix}$$

The treatment is exogenous if  $\rho_{1t} = 0$ . Note that we did not specify the structure of the correlations between the potential-outcome errors. We do not need information about these correlations to estimate POMs and treatment effects because all covariates and the outcome are observed in observations from each group.

From here, we discuss a model with an ordinal endogenous treatment. The results for binary treatment models are similar. Because the unobserved errors are bivariate normal, we can express the log likelihood in terms of the  $\Phi_2^*$  function.

For  $j = 1, \dots, T$  and  $h = 0, \dots, H$ , let

$$c_{1ihj} = \begin{cases} -\infty & h = 0 \\ \kappa_{hj} - \mathbf{x}_i \beta_j & h = 1, \dots, H-1 \\ \infty & h = H \end{cases}$$

The lower and upper limits for the  $y_i$  probability are

$$l_{1i} = c_{i(h-1)j} \quad \text{if} \quad y_i = v_h, t_i = v_{tj}$$

and

$$u_{1i} = c_{ihj} \quad \text{if} \quad y_i = v_h, t_i = v_{tj}$$

The log likelihood for the model is

$$\ln L = \sum_{i=1}^N w_i \ln \Phi_2^*([l_{1i} \quad l_{ti}], [u_{1i} \quad u_{ti}], \Sigma)$$

where the lower and upper limits for the treatment probability,  $l_{ti}$  and  $u_{ti}$ , are defined in [Treatment](#) in [\[ERM\] eoprobit](#).

The conditional probability of obtaining treatment level  $v_{th}$  is

$$\Pr(t_i = v_{th} | \mathbf{z}_{ti}) = \Phi_1^*(c_{ti(h-1)}, c_{tth}, 1)$$

where the cutpoints for the treatment probabilities  $c_{tij}$  are defined in [Treatment](#) in [\[ERM\] eoprobit](#).

For  $h = 1, \dots, H$ , the conditional probabilities for outcome level  $v_h$  at treatment level  $v_{tj}$  are

$$\Pr(y_i = v_h | \mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_{tj}) = \frac{\Phi_2^*([c_{1i(h-1)j} \quad c_{ti(j-1)}], [c_{1ihj} \quad c_{tij}], \Sigma)}{\Phi_1^*(c_{ti(j-1)}, c_{tij}, 1)}$$

The conditional POM for treatment group  $j$  and outcome category  $h$  is

$$\text{POM}_{hj}(\mathbf{x}_i) = E \{1(y_{ji} = v_h) | \mathbf{x}_i\} = \Phi_1^*(c_{1i(h-1)j}, c_{1i(h-1)j}, 1)$$

Conditional on the covariates  $\mathbf{x}_i$  and  $\mathbf{z}_{ti}$  and the treatment  $t_i = v_m$ , the POM for treatment group  $j$  and outcome category  $h$  is

$$\begin{aligned} \text{POM}_{hj}(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_m) &= E \{1(y_{ji} = v_h) | \mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_m\} \\ &= \frac{\Phi_2^*([c_{1i(h-1)j} \quad c_{ti(m-1)}], [c_{1ihj} \quad c_{tim}], \Sigma)}{\Phi_1^*(c_{ti(m-1)}, c_{tim}, 1)} \end{aligned}$$

Without loss of generality,  $t_i = v_{t1}$  corresponds to the control or base level of the treatment. Treatment effects are the differences between the potential outcomes  $y_{2i}, \dots, y_{Ti}$  and the control  $y_{1i}$ . When the potential outcomes are ordered probit, the treatment effect on a particular category is of interest.

The treatment effect of treatment group  $j$  on category  $h$  is  $1(y_{ji} = v_h) - 1(y_{1i} = v_h)$ , the difference in the outcome for individual  $i$  on being in category  $h$  if the individual receives the treatment  $t_i = v_{tj}$  instead of the control  $t_i = v_{t1}$ . Evaluating this treatment effect lets us see how the treatment affects the probability of belonging to outcome category  $h$ .

For treatment group  $j$ , the treatment effect on category  $h$  conditioned on  $\mathbf{x}_i$  is

$$\begin{aligned} \text{TE}_{hj}(\mathbf{x}_i) &= E \{1(y_{ji} = v_h) - 1(y_{1i} = v_h) | \mathbf{x}_i\} \\ &= \text{POM}_{hj}(\mathbf{x}_i) - \text{POM}_{h1}(\mathbf{x}_i) \end{aligned}$$

For treatment group  $j$ , the treatment effect on the treated (TET) on category  $h$  in treatment group  $m$  conditioned on  $\mathbf{x}_i$  and  $\mathbf{z}_{ti}$  is

$$\begin{aligned} \text{TET}_{hj}(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_m) &= E \{1(y_{ji} = v_h) - 1(y_{1i} = v_h) | \mathbf{x}_i, t_i = v_m\} \\ &= \text{POM}_{hj}(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_m) - \text{POM}_{h1}(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_m) \end{aligned}$$

We can take the expectation of these conditional predictions over the covariates to get population average parameters. The `estat teffects` or `margins` command is used to estimate the expectations as predictive margins once the model is fit with `eoprobit`. The POM for treatment group  $j$  and outcome category  $h$  is

$$\text{POM}_{hj} = E \{1(y_{ji} = v_h)\} = E \{\text{POM}_{hj}(\mathbf{x}_i)\}$$

The ATE for treatment group  $j$  and outcome category  $h$  is

$$\text{ATE}_{hj} = E \{1(y_{ji} = v_h) - 1(y_{1i} = v_h)\} = E \{\text{TE}_{hj}(\mathbf{x}_i)\}$$

For treatment group  $j$ , the average treatment effect on the treated (ATET) for outcome category  $h$  in treatment group  $m$  is

$$\begin{aligned} \text{ATET}_{hjm} &= E \{1(y_{ji} = v_h) - 1(y_{1i} = v_h) | t_i = v_m\} \\ &= E \{\text{TET}_{hj}(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_m) | t_i = v_m\} \end{aligned}$$

## Endogenous sample selection

### Probit endogenous sample selection

An ordered probit model for outcome  $y_i$  with selection on  $s_i$  has the form

$$y_i = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{x}_i \boldsymbol{\beta} + \epsilon_i \leq \kappa_h$$

$$s_i = 1 \quad (\mathbf{z}_{si} \boldsymbol{\alpha}_s + \epsilon_{si} > 0)$$

where  $\mathbf{x}_i$  are covariates that affect the outcome and  $\mathbf{z}_{si}$  are covariates that affect selection. The outcome  $y_i$  is observed if  $s_i = 1$  and is not observed if  $s_i = 0$ . The values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_H$  is taken as  $+\infty$ .

The unobserved errors  $\epsilon_i$  and  $\epsilon_{si}$  are normal with mean 0 and covariance

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho_{1s} \\ \rho_{1s} & 1 \end{bmatrix}$$

The lower and upper limits for the  $y_i$  probability,  $l_{1i}$  and  $u_{1i}$ , are as defined in (1)–(3). For the selection indicator, the lower and upper limits  $l_{si}$  and  $u_{si}$  are defined in [Probit endogenous sample selection](#) in [ERM] **eoprobit**.

The log likelihood for the model is

$$\ln L = \sum_{i \in S} w_i \ln \Phi_2^*([l_{1i} \quad l_{si}], [u_{1i} \quad u_{si}], \boldsymbol{\Sigma}) + \sum_{i \notin S} w_i \ln \Phi_1^*(l_{si}, u_{si}, 1)$$

where  $S$  is the set of observations for which  $y_i$  is observed.

In this model, the probability of success is usually predicted conditional on the covariates  $\mathbf{x}_i$  and not on the selection status  $s_i$ . The formulas for the conditional probability are thus the same as in (4).

The conditional probability of selection is

$$\Pr(s_i = 1 | \mathbf{z}_{si}) = \Phi_1^*(-\mathbf{z}_{si} \boldsymbol{\alpha}_s, \infty, 1)$$

### Tobit endogenous sample selection

Instead of constraining the selection indicator to be binary, tobit endogenous sample selection uses a censored continuous sample-selection indicator. We allow the selection variable to be left- or right-censored.

An ordered probit model for outcome  $y_i$  with tobit selection on  $s_i$  has the form

$$y_i = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{x}_i \boldsymbol{\beta} + \epsilon_i \leq \kappa_h$$

where the values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_H$  is taken as  $+\infty$ .

We observe the selection indicator  $s_i$ , which indicates the censoring status of the latent selection variable  $s_i^*$ ,

$$s_i^* = \mathbf{z}_{si}\boldsymbol{\alpha}_s + \epsilon_{si}$$

$$s_i = \begin{cases} l_i & s_i^* \leq l_i \\ s_i^* & l_i < s_i^* < u_i \\ u_i & s_i^* \geq u_i \end{cases}$$

where  $\mathbf{z}_{si}$  are covariates that affect selection, and  $l_i$  and  $u_i$  are fixed lower and upper limits.

The outcome  $y_i$  is observed when  $s_i^*$  is not censored. If  $l_i < s_i^* < u_i$ , then  $y_i$  is observed.  $y_i$  is not observed if  $s_i^* \leq l_i$ , that is, if  $s_i^*$  is left-censored.  $y_i$  is also not observed if  $s_i^*$  is right-censored,  $s_i^* \geq u_i$ . The unobserved errors  $\epsilon_i$  and  $\epsilon_{si}$  are normal with mean 0 and covariance

$$\begin{bmatrix} 1 & \rho_{1s}\sigma_s \\ \rho_{1s}\sigma_s & \sigma_s^2 \end{bmatrix}$$

For the selected observations, we can treat  $s_i$  as a continuous endogenous regressor, as in [Continuous endogenous covariates](#). In fact,  $s_i$  may even be used as a regressor for  $y_i$  in `eoprobit` (specify `tobitselect(..., main)`). On the nonselected observations, we treat  $s_i$  like the probit endogenous sample selection indicator in [Probit endogenous sample selection](#).

The conditional mean of  $\epsilon_i$  is used in the lower and upper limits for the  $y_i$  probability for selected observations. Let

$$c_{i,h} = \begin{cases} -\infty & h = 0 \\ \kappa_h - \mathbf{x}_i\boldsymbol{\beta} - \rho_{1s}\sigma_s^{-1}(s_i - \mathbf{z}_{si}\boldsymbol{\alpha}_s) & h = 1, \dots, H-1 \\ \infty & h = H \end{cases}$$

The limits for the  $y_i$  probability for selected observations are

$$l_{1i} = c_{i(h-1)} \quad \text{if } y_i = v_h$$

and

$$u_{1i} = c_{ih} \quad \text{if } y_i = v_h$$

It follows that the log likelihood is

$$\begin{aligned} \ln L = & \sum_{i \in S} w_i \{ \ln \Phi_1^*(l_{1i}, u_{1i}, 1 - \rho_{1s}^2) + \ln \phi(s_i - \mathbf{z}_{si}\boldsymbol{\alpha}_s, \sigma_s^2) \} \\ & + \sum_{i \in L} w_i \ln \Phi_1^*(l_{1i}, u_{1i}, 1) \\ & + \sum_{i \in U} w_i \ln \Phi_1^*(l_{1i}, u_{1i}, 1) \end{aligned}$$

where  $S$  is the set of observations for which  $y_i$  is observed,  $L$  is the set of observations where  $s_i^*$  is left-censored, and  $U$  is the set of observations where  $s_i^*$  is right-censored. The lower and upper limits for selection— $l_{1i}$ ,  $u_{1i}$ ,  $l_{ui}$ , and  $u_{ui}$ —are defined in [Tobit endogenous sample selection](#) in [\[ERM\] eoprobit](#).



The conditional probabilities on  $s_i = S_i$  are

$$\Pr(y_i = v_h | \mathbf{x}_i) = \Phi_1^*(c_{i(h-1)}, c_{ih}, 1 - \rho_{1s}^2)$$

If we do not include  $s_i$  in the main outcome equation, the probability of success is calculated as (4) again.

## Combinations of features

Extended ordered probit regression models that involve multiple features can be formulated using the techniques discussed in *Likelihood for multiequation models* in [ERM] **eoprobit**. Essentially, the density of the observed endogenous covariates can be written in terms of the unobserved normal errors. The observed endogenous and exogenous covariates determine the range of the errors, and the joint density can be evaluated as multivariate normal probabilities and densities.

## Confidence intervals

The estimated variances will always be nonnegative, and the estimated correlations will always fall in  $(-1, 1)$ . To obtain confidence intervals that accommodate these ranges, we must use transformations.

We use the log transformation to obtain confidence intervals for variance parameters, and we use the atanh transformation to obtain confidence intervals for correlation parameters. For details, see *Confidence intervals* in [ERM] **eoprobit**.

## References

- Botezat, A., and F. Pfeiffer. 2014. The impact of parents' migration on the well-being of children left behind: Initial evidence from Romania. IZA Discussion Paper No. 8225, Institute for the Study of Labor (IZA). <http://ftp.iza.org/dp8225.pdf>
- Chyi, H., and S. Mao. 2012. The determinants of happiness of China's elderly population. *Journal of Happiness Studies* 13: 167–185.
- De Luca, G., and V. Perotti. 2011. Estimation of ordered response models with sample selection. *Stata Journal* 11: 213–239.
- Gregory, C. A. 2015. Estimating treatment effects for ordered outcomes using maximum simulated likelihood. *Stata Journal* 15: 756–774.
- Roodman, D. 2011. Fitting fully observed recursive mixed-process models with **cmp**. *Stata Journal* 11: 159–206.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

- [ERM] **eoprobit postestimation** — Postestimation tools for eoprobit
- [ERM] **eoprobit predict** — predict after eoprobit
- [ERM] **estat teffects** — Average treatment effects for extended regression models
- [ERM] **intro 8** — Conceptual introduction via worked example
- [R] **heckoprobit** — Ordered probit model with sample selection
- [R] **oprobit** — Ordered probit regression
- [SVY] **svy estimation** — Estimation commands for survey data
- [U] **20 Estimation and postestimation commands**

Postestimation commands  
Methods and formulas

[predict](#)  
References

[margins](#)  
Also see

Remarks and examples

Postestimation commands

The following postestimation command is of special interest after `eoprobit`:

Command	Description
<code>estat teffects</code>	treatment effects and potential-outcome means

The following standard postestimation commands are also available after `eoprobit`:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of estimates
<code>estat ic</code>	Akaike’s and Schwarz’s Bayesian information criteria (AIC and BIC)
<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>forecast</code>	dynamic forecasts and simulations
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
* <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 coefficients
<code>predict</code>	predictions, residuals, influence statistics, and other diagnostic measures
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of estimates
<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

\* `forecast` and `lrtest` are not appropriate with `svy` estimation results.

# predict

Predictions after `eoprobit` are described in

- [\[ERM\] eoprobit predict](#) predict after eoprobit
- [\[ERM\] predict treatment](#) predict for treatment statistics
- [\[ERM\] predict advanced](#) predict's advanced features

[\[ERM\] eoprobit predict](#) describes the most commonly used predictions. If you fit a model with treatment effects, predictions specifically related to these models are detailed in [\[ERM\] predict treatment](#). [\[ERM\] predict advanced](#) describes less commonly used predictions, such as predictions of outcomes in auxiliary equations.

# margins

## Description for margins

`margins` estimates margins of response for probabilities, means, potential-outcome means, treatment effects, and linear predictions.

## Menu for margins

Statistics > Postestimation

## Syntax for margins

```
margins [marginlist] [ , options ]  
margins [marginlist] , predict(statistic ...) [predict(statistic ...) ...] [options]
```

statistic	Description
Main	
<u>pr</u>	probability for binary or ordinal $y_j$ ; the default
<u>mean</u>	mean
<u>pomean</u>	potential-outcome mean
<u>te</u>	treatment effect
<u>tet</u>	treatment effect on the treated
<u>xb</u>	linear prediction
<u>pr</u> ( $a,b$ )	$\Pr(a < y_j < b)$ for continuous $y_j$
<u>e</u> ( $a,b$ )	$E(y_j   a < y_j < b)$ for continuous $y_j$
<u>y</u> <u>star</u> ( $a,b$ )	$E(y_j^*), y_j^* = \max\{a, \min(y_j, b)\}$ for continuous $y_j$

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.  
For the full syntax, see [\[R\] margins](#).

## Remarks and examples

See [ERM] [intro 6](#) for an overview of using margins and predict after eoprobit. For examples using margins, predict, and estat teffects, see [Interpreting effects](#) in [ERM] [intro 8](#) and see [ERM] [example 1a](#).

## Methods and formulas

This section contains methods and formulas for counterfactual predictions and inference. Methods and formulas for all other predictions are given in [Methods and formulas](#) of [ERM] [eoprobit](#). In [Methods and formulas](#) of [ERM] [eoprobit](#), we discussed how treatment effects are evaluated in extended ordered probit regression models. Here, we discuss the counterfactual framework used to evaluate the effects of other covariates.

In the extended ordered probit regression model for  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and  $\mathbf{w}_i$ , we partition each set of covariates into two groups. The exogenous covariates  $\mathbf{x}_i$  are partitioned into  $\mathbf{x}_i^c$  and  $\mathbf{x}_i^{nc}$ , where we are interested in the effect of changes in  $\mathbf{x}_i^c$ . Similarly, the endogenous covariates  $\mathbf{w}_i$  are partitioned into  $\mathbf{w}_i^c$  and  $\mathbf{w}_i^{nc}$ , where the effect of changes in  $\mathbf{w}_i^c$  is of interest. The superscripts indicate what is a counterfactual value ( $c$ ) and what is not ( $nc$ ).

If  $\mathbf{x}_i^c = \mathbf{a}_0$  and  $\mathbf{w}_i^c = \mathbf{a}_{20}$ , for covariates  $\mathbf{w}_i^{nc}$  and  $\mathbf{x}_i^{nc}$  we would observe the outcome

$$y_{0i} = v_h \quad \text{iff} \quad \kappa_{h-1} < \beta_{0nc}\mathbf{x}_i^{nc} + \beta_{20nc}\mathbf{w}_i^{nc} + \beta_c\mathbf{a}_0 + \beta_{2c}\mathbf{a}_{20} + \epsilon_{0i} \leq \kappa_h$$

The values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_H$  is taken as  $+\infty$ . Where the unobserved error  $\epsilon_{0i}$  is standard normal. We treat  $\beta_c\mathbf{a}_0 + \beta_{2c}\mathbf{a}_{20} = \beta_{c0}$  as a constant intercept, because it is the same for each value combination of the covariates  $\mathbf{w}_i^{nc}$  and  $\mathbf{x}_i^{nc}$  and the error  $\epsilon_{0i}$ .

Similarly, if  $\mathbf{x}_i^c = \mathbf{a}_1$  and  $\mathbf{w}_i^c = \mathbf{a}_{21}$ , for covariates  $\mathbf{w}_i^{nc}$  and  $\mathbf{x}_i^{nc}$  we would observe the outcome

$$y_{1i} = v_h \quad \text{iff} \quad \kappa_{h-1} < \beta_{1nc}\mathbf{x}_i^{nc} + \beta_{21nc}\mathbf{w}_i^{nc} + \beta_c\mathbf{a}_1 + \beta_{2c}\mathbf{a}_{21} + \epsilon_{1i} \leq \kappa_h$$

To define the effects, for  $j = 0, 1$  and  $h = 1, \dots, H$ , we can examine the variables

$$y_{jhi} = \begin{cases} 1 & \text{if } y_{ji} = v_h \\ 0 & \text{if } y_{ji} \neq v_h \end{cases}$$

The effect of changing  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$  from  $\mathbf{a}_0$  and  $\mathbf{a}_{20}$  to  $\mathbf{a}_1$  and  $\mathbf{a}_{21}$  on the probability that  $y_i = v_h$  is the expected difference between  $y_{1hi}$  and  $y_{0hi}$ .

To obtain this difference, we average the conditional probabilities of  $y_{1hi}$  and  $y_{0hi}$  as a predictive margin.

For  $j = 0, 1$  and  $h = 1, \dots, H$ , we can predict the counterfactual probability for group  $j$  using the tools discussed in [Predictions using the full model](#) in [ERM] [eoprobit postestimation](#):

$$\text{CP}_{jh}(\mathbf{w}_i^{nc}, \mathbf{x}_i^{nc}, \mathbf{z}_i) = \Pr(y_{jhi} = 1 | \mathbf{w}_i^{nc}, \mathbf{x}_i^{nc}, \mathbf{x}_i^c = \mathbf{a}_j, \mathbf{z}_i)$$

where  $\mathbf{z}_i$  are instruments necessary for modeling the endogenous regressors  $\mathbf{w}_i^{nc}$ . By the law of iterated expectations, we have

$$E(y_{1hi} - y_{0hi}) = E\{\text{CP}_{1h}(\mathbf{w}_i^{nc}, \mathbf{x}_i^{nc}, \mathbf{z}_i)\} - E\{\text{CP}_{0h}(\mathbf{w}_i^{nc}, \mathbf{x}_i^{nc}, \mathbf{z}_i)\}$$

So the effect of changing  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$  from  $\mathbf{a}_0$  and  $\mathbf{a}_{20}$  to  $\mathbf{a}_1$  and  $\mathbf{a}_{21}$  can be estimated as a predictive margin on the counterfactual probabilities.

We can use `predict` with the `fix()` and `target()` options to predict the counterfactual probabilities. The `fix()` option is used to indicate the endogenous covariates in  $\mathbf{w}_i^c$ . The `target()` option can be used to set the counterfactual values  $a_j$  and  $a_{2j}$  of  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$ .

When  $\mathbf{w}_i^c$  corresponds to a single ordinal or binary regressor, the difference in counterfactual probabilities corresponds to a treatment effect of  $\mathbf{w}_i^c$ . We can also evaluate the effect of a change in  $\mathbf{w}_i^c$  and  $\mathbf{x}_i^c$ , conditioned on  $\mathbf{w}_i^c$ . This effect is analogous to the treatment effect on the treated discussed in *Methods and formulas* of [ERM] **eoprobit**. We are conditioning the effect on some base value for  $\mathbf{w}_i^c$ ,  $\mathbf{w}_i^c = \mathbf{b}$ .

Now, the counterfactual probabilities are conditioned on  $\mathbf{w}_i^c = \mathbf{b}$ . So for  $j = 0, 1$  and  $h = 1, \dots, H$ , we have

$$\text{CP}_{bjh}(\mathbf{w}_i^{nc}, \mathbf{b}, \mathbf{x}_i^{nc}, \mathbf{z}_i) = \Pr(y_{jhi} = 1 | \mathbf{w}_i^{nc}, \mathbf{w}_i^c = \mathbf{b}, \mathbf{x}_i^{nc}, \mathbf{x}_i^c = \mathbf{a}_j, \mathbf{z}_{bi})$$

where  $\mathbf{z}_{bi}$  are instruments necessary for modeling the endogenous regressors  $\mathbf{w}_i^{nc}$  and  $\mathbf{w}_i^c$ . This counterfactual probability can be evaluated using the tools discussed in *Predictions using the full model* in [ERM] **eoprobit postestimation**.

By the law of iterated expectations, we have

$$\begin{aligned} E(y_{1hi} - y_{0hi} | \mathbf{w}_i^c = \mathbf{b}) &= E \{ \text{CP}_{b1h}(\mathbf{w}_i^{nc}, \mathbf{w}_i^c = \mathbf{b}, \mathbf{x}_i^{nc}, \mathbf{z}_i) | \mathbf{w}_i^c = \mathbf{b} \} - \\ &E \{ \text{CP}_{b0h}(\mathbf{w}_i^{nc}, \mathbf{w}_i^c = \mathbf{b}, \mathbf{x}_i^{nc}, \mathbf{z}_i) | \mathbf{w}_i^c = \mathbf{b} \} \end{aligned}$$

So the effect of changing  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$  from  $\mathbf{a}_0$  and  $\mathbf{a}_{20}$  to  $\mathbf{a}_1$  and  $\mathbf{a}_{21}$  conditioned on  $\mathbf{w}_i^c = \mathbf{b}$  can be estimated as a predictive margin on the counterfactual probabilities.

The base values  $\mathbf{b}$  for  $\mathbf{w}_i^c$  are specified in the `base()` option. As before, `target()` can be used to specify the counterfactual values for  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$ .

When  $\mathbf{x}_i^c = \mathbf{x}_i$  and  $\mathbf{w}_i^c = \mathbf{w}_i$ , the counterfactual probability matches the average structural probability (ASP). Applying the average structural function (ASF) discussed by [Blundell and Powell \(2003\)](#), [Blundell and Powell \(2004\)](#), [Wooldridge \(2005\)](#), and [Wooldridge \(2014\)](#) to a conditional probability on the covariates and unobserved endogenous error produces the ASP.

In the ordered probit model, for exogenous covariates  $\mathbf{x}_i$  and endogenous regressors  $\mathbf{w}_i$ , we have

$$y_i = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{x}_i\boldsymbol{\beta} + \mathbf{w}_i\boldsymbol{\beta}_2 + \epsilon_i \leq \kappa_h$$

The values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_H$  is taken as  $+\infty$ . The error  $\epsilon_i$  is standard normal and correlated with  $\mathbf{w}_i$ .

The ASP provides a structural interpretation of  $\boldsymbol{\beta}$  and  $\boldsymbol{\beta}_2$  when the  $\mathbf{w}_i$  are correlated with  $\epsilon_i$ . Because  $\epsilon_i$  is a normally distributed, mean 0, random variable, we can split it into two mean 0, normally distributed, independent parts,

$$\epsilon_i = u_i + \psi_i$$

where  $u_i = \gamma\epsilon_{2i}$  is the unobserved heterogeneity that gives rise to the endogeneity and  $\psi_i$  is an error term with variance  $\sigma_\psi^2$ .

For  $h = 0, \dots, H$ , define

$$c_{ih} = \begin{cases} -\infty & h = 0 \\ \kappa_h - \mathbf{x}_i\boldsymbol{\beta} - \mathbf{w}_i\boldsymbol{\beta}_2 - u_i & h = 1, \dots, H-1 \\ \infty & h = H \end{cases}$$

Conditional on the covariates and the unobserved heterogeneity, we have

$$\begin{aligned} E\{\mathbf{1}(y_i = v_h) | \mathbf{x}_i, \mathbf{w}_i, u_i\} &= \Pr(y_i = v_h | \mathbf{x}_i, \mathbf{w}_i, u_i) \\ &= \Phi_1^*(c_{i(h-1)}, c_{ih}, \sigma_\psi^2) \end{aligned}$$

Because  $u_i$  is an unobserved random variable, these conditional probabilities are not observable. Integrating out the  $u_i$ , just like we do with random effects in panel-data models, produces the ASP for each category,

$$\text{ASP}_h(\mathbf{x}_i^0, \mathbf{w}_i^0) = \int E\{\mathbf{1}(y_i = v_h) | \mathbf{x}_i^0, \mathbf{w}_i^0, u_i\} f(u_i) du_i$$

where  $f(u_i)$  is the marginal distribution of  $u_i$ , and  $\mathbf{x}_i^0$  and  $\mathbf{w}_i^0$  are given covariate values.

## References

- Blundell, R. W., and J. L. Powell. 2003. Endogeneity in nonparametric and semiparametric regression models. In *Advances in Economics and Econometrics: Theory and Applications, Eighth World Congress*, ed. M. Dewatripont, L. P. Hansen, and S. J. Turnovsky, vol. 2, 312–357. Cambridge: Cambridge University Press.
- . 2004. Endogeneity in semiparametric binary response models. *Review of Economic Studies* 71: 655–679.
- Wooldridge, J. M. 2005. Unobserved heterogeneity and estimation of average partial effects. In *Identification and Inference for Econometric Models: Essays in Honor of Thomas Rothenberg*, ed. D. W. K. Andrews and J. H. Stock, 27–55. New York: Cambridge University Press.
- . 2014. Quasi-maximum likelihood estimation and testing for nonlinear models with endogenous explanatory variables. *Journal of Econometrics* 182: 226–234.

## Also see

- [ERM] [eoprobit](#) — Extended ordered probit regression
- [ERM] [eoprobit predict](#) — predict after eoprobit
- [ERM] [predict treatment](#) — predict for treatment statistics
- [ERM] [predict advanced](#) — predict’s advanced features
- [ERM] [eoprobit postestimation](#) — Postestimation tools for eoprobit
- [U] [20 Estimation and postestimation commands](#)

Description	Syntax
Options for statistics	Options for how results are calculated
Remarks and examples	Methods and formulas
Also see	

Description

In this entry, we show how to create new variables containing observation-by-observation predictions after fitting a model with `eoprobit`.

Syntax

You previously fit the model

```
eoprobit y x1 ..., ...
```

The equation specified immediately after the `eoprobit` command is called the main equation. It is

$$\Pr(y_i = m) = \Pr(c_{m-1} \leq \mathbf{x}_i\beta + e_{i \cdot \mathbf{y}} \leq c_m)$$

Note that the equation produces a probability for each outcome  $m$ ,  $m = 1$  to  $M$ . `predict` calculates predictions for the probabilities in the main equation. The other equations in the model are called auxiliary equations or complications.

The syntax of `predict` is

```
predict [type] { stub*|newvarlist } [if] [in] [, stdstatistics howcalculated]
```

<i>stdstatistics</i>	Description
<code>pr</code>	probability of each outcome; the default
<code>outlevel(#)</code>	calculate probability for $m = \#$ only
<code>xb</code>	linear prediction excluding all complications

<i>howcalculated</i>	Description
<code>default</code>	not fixed; base values from data
<code>fix(<i>endogvars</i>)</code>	fix specified endogenous covariates
<code>base(<i>valspecs</i>)</code>	specify base values of any variables
<code>target(<i>valspecs</i>)</code>	more convenient way to specify <code>fix()</code> and <code>base()</code>

Note: The `fix()` and `base()` options affect results only in models with endogenous variables in the main equation. The `target()` option is sometimes a more convenient way to specify the `fix()` and `base()` options.

*endogvars* are names of one or more endogenous variables appearing in the main equation.

*valspecs* specify the values for variables at which predictions are to be evaluated. Each *valspec* is of the form

*varname* = #

*varname* = (*exp*)

*varname* = *othervarname*

For instance, `base(valspecs)` could be `base(w1=0)` or `base(w1=0 w2=1)`.

Notes:

- (1) `predict` can also calculate treatment-effect statistics. See [\[ERM\] predict treatment](#).
- (2) `predict` can also make predictions for the other equations in addition to the main-equation predictions discussed here. See [\[ERM\] predict advanced](#).

## Options for statistics

`pr` calculates the predicted probability for each outcome. In each observation, the predictions are the probabilities conditioned on the covariates. Results depend on how complications are handled, which is determined by the [howcalculated](#) options.

`outlevel(#)` specifies to calculate only the probability for outcome  $m = \#$  rather than calculating  $M$  probabilities. If you do not specify this option,  $y$  records three outcomes. You type

```
. predict p1 p2 p3
```

to obtain the probabilities for each outcome. If you want only the probability of the third outcome, you can type

```
. predict p3, outlevel(#3)
```

If the third outcome corresponded to  $y==3$ , you could instead type

```
. predict p3, outlevel(3)
```

If the third outcome corresponded to  $y==57$ , you could instead type

```
. predict p3, outlevel(57)
```

Most users number the outcomes 1, 2, and 3. Some users number them 0, 1, and 2. You could even number them 3, 5, and 57. Stata does not care how they are numbered.

`xb` specifies that the linear prediction be calculated ignoring all complications.

## Options for how results are calculated

By default, predictions are calculated taking into account all complications. This is discussed in [Remarks and examples](#) of [\[ERM\] eregress predict](#).

`fix(varname ...)` specifies a list of endogenous variables from the main equation to be treated as if they were exogenous. This was discussed in [\[ERM\] intro 3](#) and is discussed further in [Remarks and examples](#) of [\[ERM\] eregress predict](#).



`base(varname = ...)` specifies a list of variables from any equation and values for them. If `eoprobit` were a linear model, we would tell you those values will be used in calculating the expected value of  $e_i.y$ . That thinking will not mislead you but is not formally correct in the case of `eoprobit`. Linear or nonlinear, errors from other equations spill over into the main equation because of correlations between errors. The correlations were estimated when the model was fit. The amount of spillover depends on those correlations and the values of the errors. This issue was discussed in [ERM] [intro 3](#) and is further discussed in [Remarks and examples](#) of [ERM] [eregress predict](#).

`target(varname = ...)` is sometimes a more convenient way to specify the `fix()` and `base()` options. You specify a list of variables from the main equation and values for them. Those values override the values of the variables calculating  $\beta_0 + \beta_1 x_{1i} + \dots$ . Use of `target()` is discussed in [Remarks and examples](#) of [ERM] [eregress predict](#).

## Remarks and examples

Remarks are presented under the following headings:

*Using predict after eoprobit*  
*How to think about nonlinear models*

## Using predict after eoprobit

`eoprobit` fits ordinal probit models. The outcome variable  $y$  takes on various values such as 1, 2, 3, and 4, and each represents an ordered category, such as cannot walk, walks with difficulty, walks with few problems, and walks well. When you use `predict` after `eoprobit`, remember to specify variables corresponding to each category.

```
. predict p1 p2 p3 p4
```

Alternatively, specify the `outlevel(#)` option.

With this exception, predictions after fitting models with `eoprobit` are handled the same as they are after fitting models with `eregress`. The issues are the same. See [ERM] [eregress predict](#).

## How to think about nonlinear models

What we wrote in [ERM] [eoprobit predict](#) applies equally to the use of `predict` after `eoprobit`. We wrote

Probit is a nonlinear model, and yet we just said that predictions after fitting models with `eoprobit` are handled the same as they are after fitting models with `eregress`. That statement is partly true, not misleading, but false in its details.

The regression-base discussion that we routed you to is framed in terms of expected values. In the nonlinear models, it needs to be framed in terms of distributional assumptions about the errors. For instance, `predict` after `eoprobit` does not predict the expected value (mean) of  $e_i.y$ . It calculates the probability that  $e_i.y$  exceeds  $-x_i\beta$ . These details matter hugely in implementation but can be glossed over for understanding the issues. For a full treatment of the issues, see [Methods and formulas](#) of [ERM] [eoprobit](#).

## Methods and formulas

See *Methods and formulas* of [ERM] [eoprobit postestimation](#).

## Also see

[ERM] [eoprobit postestimation](#) — Postestimation tools for eoprobit

[ERM] [eoprobit](#) — Extended ordered probit regression

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`eprobit` fits a probit regression model that accommodates any combination of endogenous covariates, nonrandom treatment assignment, and endogenous sample selection. Continuous, binary, and ordinal endogenous covariates are allowed. Treatment assignment may be endogenous or exogenous. A probit or tobit model may be used to account for endogenous sample selection.

## Quick start

Probit regression of `y` on `x` with continuous endogenous covariate `y2` modeled by `x` and `z`

```
eprobit y x, endogenous(y2 = x z)
```

As above, but adding continuous endogenous covariate `y3` modeled by `x` and `z2`

```
eprobit y x, endogenous(y2 = x z) endogenous(y3 = x z2)
```

Probit regression of `y` on `x` with binary endogenous covariate `d` modeled by `x` and `z`

```
eprobit y x, endogenous(d = x z, probit)
```

Probit regression of `y` on `x` with endogenous treatment recorded in `trtvar` and modeled by `x` and `z`

```
eprobit y x, entreat(trtvar = x z)
```

Probit regression of `y` on `x` with exogenous treatment recorded in `trtvar`

```
eprobit y x, extreat(trtvar)
```

Probit regression of `y` on `x` with endogenous sample-selection indicator `selvar` modeled by `x` and `z`

```
eprobit y x, select(selvar = x z)
```

As above, but adding endogenous covariate `y2` modeled by `x` and `z2`

```
eprobit y x, select(selvar = x z) endogenous(y2 = x z2)
```

As above, but adding endogenous treatment recorded in `trtvar` and modeled by `x` and `z3`

```
eprobit y x, select(selvar = x z) endogenous(y2 = x z2) ///  
entreat(trtvar = x z3)
```

## Menu

Statistics > Endogenous covariates > Models adding selection and treatment > Probit regression

## Syntax

*Basic probit regression with endogenous covariates*

```
eprobit depvar [indepvars] ,
      endogenous(depvarsen = varlisten) [options]
```

*Basic probit regression with endogenous treatment assignment*

```
eprobit depvar [indepvars] ,
      entreat(depvartr [= varlisttr]) [options]
```

*Basic probit regression with exogenous treatment assignment*

```
eprobit depvar [indepvars] ,
      extreat(tvar) [options]
```

*Basic probit regression with sample selection*

```
eprobit depvar [indepvars] ,
      select(depvars = varlists) [options]
```

*Basic probit regression with tobit sample selection*

```
eprobit depvar [indepvars] ,
      tobitselect(depvars = varlists) [options]
```

*Probit regression combining endogenous covariates, treatment, and selection*

```
eprobit depvar [indepvars] [if] [in] [weight] [, extensions options]
```

<i>extensions</i>	Description
Model	
<u>endogenous</u> ( <i>enspec</i> )	model for endogenous covariates; may be repeated
<u>entreat</u> ( <i>entrspec</i> )	model for endogenous treatment assignment
<u>extreat</u> ( <i>extrspec</i> )	exogenous treatment
<u>select</u> ( <i>selspec</i> )	probit model for selection
<u>tobitselect</u> ( <i>tselspec</i> )	tobit model for selection
<i>options</i>	Description
Model	
<u>noconstant</u>	suppress constant term
<u>offset</u> ( <i>varname<sub>o</sub></i> )	include <i>varname<sub>o</sub></i> in model with coefficient constrained to 1
<u>constraints</u> ( <i>numlist</i> )	apply specified linear constraints
<u>collinear</u>	keep collinear variables
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 <u>level</u> (95)
<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> (#)	set the number of integration (quadrature) points for integration over four or more dimensions; default is <u>intpoints</u> (128)
<u>triintpoints</u> (#)	set the number of integration (quadrature) points for integration over three dimensions; default is <u>triintpoints</u> (10)
Maximization	
<u>maximize_options</u>	control the maximization process; seldom used
<u>coeflegend</u>	display legend instead of statistics

*enspec* is *depvars<sub>en</sub>* = *varlist<sub>en</sub>* [ , *enopts* ]  
where *depvars<sub>en</sub>* is a list of endogenous covariates. Each variable in *depvars<sub>en</sub>* specifies an endogenous covariate model using the common *varlist<sub>en</sub>* and options.

*entrspec* is *depvar<sub>tr</sub>* [= *varlist<sub>tr</sub>*] [ , *tropts* ]  
where *depvar<sub>tr</sub>* is a variable indicating treatment assignment. *varlist<sub>tr</sub>* is a list of covariates predicting treatment assignment.

*extrspec* is *tvar* [ , nomain nointeract ]  
where *tvar* is a variable indicating treatment assignment.

*selspec* is `depvars = varlists [ , noconstant offset(varnameo) ]`

where *depvar<sub>s</sub>* is a variable indicating selection status. *depvar<sub>s</sub>* must be coded as 0, indicating that the observation was not selected, or 1, indicating that the observation was selected. *varlist<sub>s</sub>* is a list of covariates predicting selection.

*tselspec* is `depvars = varlists [ , tselopts ]`

where *depvar<sub>s</sub>* is a continuous variable. *varlist<sub>s</sub>* is a list of covariates predicting *depvar<sub>s</sub>*. The censoring status of *depvar<sub>s</sub>* indicates selection, where a censored *depvar<sub>s</sub>* indicates that the observation was not selected and a noncensored *depvar<sub>s</sub>* indicates that the observation was selected.

<i>enopts</i>	Description
---------------	-------------

Model

<u>probit</u>	treat endogenous covariate as binary
<u>oprobit</u>	treat endogenous covariate as ordinal
<u>nomain</u>	do not add endogenous covariate to main equation
<u>noconstant</u>	suppress constant term

<i>tropts</i>	Description
---------------	-------------

Model

<u>nomain</u>	do not add treatment indicator to main equation
<u>nointeract</u>	do not interact treatment with covariates in main equation
<u>noconstant</u>	suppress constant term
<u>offset</u> (varname <sub>o</sub> )	include varname <sub>o</sub> in model with coefficient constrained to 1

<i>tselopts</i>	Description
-----------------	-------------

Model

<u>ll</u> (varname   #)	left-censoring variable or limit
<u>ul</u> (varname   #)	right-censoring variable or limit
<u>main</u>	add censored selection variable to main equation
<u>noconstant</u>	suppress constant term
<u>offset</u> (varname <sub>o</sub> )	include varname <sub>o</sub> in model with coefficient constrained to 1

*indepvars*, *varlist<sub>en</sub>*, *varlist<sub>tr</sub>*, and *varlist<sub>s</sub>* may contain factor variables; see [U] 11.4.3 Factor variables.

*depvar*, *indepvars*, *depvars<sub>en</sub>*, *varlist<sub>en</sub>*, *depvar<sub>tr</sub>*, *varlist<sub>tr</sub>*, *tvar*, *depvar<sub>s</sub>*, and *varlist<sub>s</sub>* may contain time-series operators; see [U] 11.4.4 Time-series varlists.

*bootstrap*, *by*, *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*.

*vce*() and weights are not allowed with the *svy* prefix; see [SVY] *svy*.

*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

`endogenous(enspec)`, `entreat(entrspec)`, `extreat(extrspec)`, `select(selspec)`, `tobitselect(tselspec)`; see [\[ERM\] erm options](#).

`noconstant`, `offset(varnameo)`, `constraints(numlist)`, `collinear`; see [\[R\] estimation options](#).

### SE/Robust

`vce(vcetype)`; see [\[ERM\] erm options](#).

### Reporting

`level(#)`, `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(#)`, `triintpoints(#)`; see [\[ERM\] erm 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](#).

Setting the optimization type to `technique(bhhh)` resets the default `vcetype` to `vce(opg)`.

The following option is available with `eprobit` but is not shown in the dialog box:

`coeflegend`; see [\[R\] estimation options](#).

## Remarks and examples

`eprobit` fits models that we refer to as “extended probit regression models”, meaning that they accommodate endogenous covariates, nonrandom treatment assignment, and endogenous sample selection. `eprobit` can account for these complications whether they arise individually or in combination.

In this entry, you will find information on the `eprobit` command syntax. You can see [Methods and formulas](#) for a full description of the models that can be fit with `eprobit` and details about how those models are fit.

More information on extended probit regression models is found in the separate introductions and example entries. We recommend reading those entries to learn how to use `eprobit`. Below, we provide a guide to help you locate the ones that will be helpful to you.

For an introduction to `eprobit` and the other extended regression commands (`eregress`, `eintreg`, and `eoprobit`), see [\[ERM\] intro 1](#)–[\[ERM\] intro 8](#).

[\[ERM\] intro 1](#) introduces the ERM commands, the problems they address, and their syntax.

[ERM] **intro 2** provides background on the four types of models—linear regression, interval regression, probit regression, and ordered probit regression—that can be fit using ERM commands.

[ERM] **intro 3** considers the problem of endogenous covariates and how to solve it using ERM commands.

[ERM] **intro 4** gives an overview of endogenous sample selection and using ERM commands to account for it.

[ERM] **intro 5** covers nonrandom treatment assignment and how to account for it using `eprobit` or any of the other ERM commands.

[ERM] **intro 6** discusses interpretation of results. You can interpret coefficients from `eprobit` in the usual way, but this introduction goes beyond the interpretation of coefficients. We demonstrate how to find answers to interesting questions by using `margins`. If your model includes an endogenous covariate or an endogenous treatment, the use of `margins` differs from its use after other estimation commands, so we strongly recommend reading this intro if you are fitting these types of models.

[ERM] **intro 7** will be particularly helpful if you are familiar with `ivprobit`, `heckprobit`, and other commands that address endogenous covariates, sample selection, or nonrandom treatment assignment. This introduction is a Rosetta stone that maps the syntax of those commands to the syntax of `eprobit`.

[ERM] **intro 8** walks you through an example that gives insight into the concepts of endogenous covariates, treatment assignment, and sample selection while fitting models with `eregress` that address these complications. Although the example uses `eregress`, the discussion applies equally to `eprobit`. This intro also demonstrates how to interpret results by using `margins` and `estat teffects`.

Additional examples are presented in [ERM] **example 1a**–[ERM] **example 6b**. For examples using `eprobit`, see

[ERM] <b>example 3a</b>	Probit regression with continuous endogenous covariate
[ERM] <b>example 3b</b>	Probit regression with endogenous covariate and treatment
[ERM] <b>example 4a</b>	Probit regression with endogenous sample selection
[ERM] <b>example 4b</b>	Probit regression with endogenous treatment and sample selection
[ERM] <b>example 5</b>	Probit regression with endogenous ordinal treatment

See *Examples* in [ERM] **intro** for an overview of all the examples. These examples demonstrate all four extended regression commands, and all may be interesting because they handle complications in the same way.

You can also find in literature discussion and examples of many models that `eprobit` can fit. This includes discussion of the probit model with continuous endogenous covariates (Newey 1987), the probit model with multiple endogenous binary covariates (Arendt and Holm 2006), and the probit model with an endogenous treatment (Angrist [2001] and Pindyck and Rubinfeld [1998]). `eprobit` can also be used for probit models with selection, such as that discussed by Van de Ven and Van Pragg (1981), and for the model with a tobit selection equation, discussed in Wooldridge (2010, sec. 19.7). Roodman (2011) investigated probit models with endogenous covariates and endogenous sample selection, and demonstrated how multiple observational data complications could be addressed with a triangular model structure. His work has been used to model processes like the impact of finance on the probability of being an entrepreneur (Karymshakov, Sultakeev, and Sulaimanova 2015) and the impact of foreign direct investment on the probability of creating an innovative product (Vahter 2011).



## Stored results

`eprobit` stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(N_selected)</code>	number of uncensored observations
<code>e(N_nonselected)</code>	number of censored observations
<code>e(k)</code>	number of parameters
<code>e(k_cat#)</code>	number of categories for the <i>#th depvar</i> , ordinal
<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_dv)</code>	number of dependent variables
<code>e(k_aux)</code>	number of auxiliary parameters
<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>	$\chi^2$
<code>e(p)</code>	significance
<code>e(n_quad)</code>	number of integration points for multivariate normal
<code>e(n_quad3)</code>	number of integration points for trivariate normal
<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>	<code>eprobit</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	names of dependent variables
<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(offset#)</code>	offset for the <i>#th depvar</i>
<code>e(chi2type)</code>	Wald; type of model $\chi^2$ test
<code>e(vce)</code>	<i>vcetype</i> 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>b V</b>
<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(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(cat#)</code>	categories for the <i>#th depvar</i> , ordinal
<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
------------------------	-------------------------

## Methods and formulas

The methods and formulas presented here are for the probit-outcome model. They are a special case of the methods and formulas derived by [Roodman \(2011\)](#).

Methods and formulas are presented under the following headings:

- [Introduction](#)
- [Endogenous covariates](#)
  - [Continuous endogenous covariates](#)
  - [Binary and ordinal endogenous covariates](#)
- [Treatment](#)
- [Endogenous sample selection](#)
  - [Probit endogenous sample selection](#)
  - [Tobit endogenous sample selection](#)
- [Combined model](#)
- [Confidence intervals](#)
- [Likelihood for multiequation models](#)

## Introduction

A probit regression of outcome  $y_i$  on covariates  $\mathbf{x}_i$  may be written as

$$y_i = 1 (\mathbf{x}_i\beta + \epsilon_i > 0)$$

where the errors  $\epsilon_i$  are distributed as standard normal. The log likelihood is

$$\ln L = \sum_{i=1}^N w_i \{y_i \ln \Phi(\mathbf{x}_i\beta) + (1 - y_i) \ln \Phi(-\mathbf{x}_i\beta)\}$$

where  $w_i$  are the weights. The conditional probability of success is

$$E(y_i|\mathbf{x}_i) = \Pr(y_i = 1|\mathbf{x}_i) = \Phi(\mathbf{x}_i\beta)$$

The standard normal cumulative distribution function  $\Phi(\cdot)$  used in these expressions is a one-sided probability that the random variable is below a certain point. In the models we describe later, it will be useful to use two-sided probabilities. For two-sided probabilities, we define  $\Phi_d^*$  with three inputs. The first two inputs are  $d$ -dimensional row vectors  $\mathbf{l}$  and  $\mathbf{u}$  that have values in  $\mathbb{R} \cup \{-\infty, \infty\}$ , the extended real line. The final input is a  $d \times d$  real-valued and positive-definite matrix  $\Sigma$ .

$$\Phi_d^*(\mathbf{l}, \mathbf{u}, \Sigma) = \int_{l_1}^{u_1} \dots \int_{l_d}^{u_d} \phi_d(\boldsymbol{\epsilon}, \Sigma) d\epsilon_1 \dots d\epsilon_d$$

where  $\phi_d$  is the density of a mean 0, multivariate normal random variable. For details on the calculation of  $\Phi_d^*$ , see [\[M-5\] mvnormal\(\)](#). The probabilities are approximated using numeric integration. The number of integration or quadrature points can be varied to attain better approximations. For trivariate errors, we use the method of [Drezner \(1994\)](#). For four or more errors, we use the method of [Miwa, Hayter, and Kuriki \(2003\)](#).

The lower and upper limits  $l_{1i}$  and  $u_{1i}$  on the unobserved  $\epsilon_i$  are based on the observed values of  $y_i$  and  $\mathbf{x}_i$  and are defined as

$$l_{1i} = \begin{cases} -\infty & y_i = 0 \\ -\mathbf{x}_i\beta & y_i = 1 \end{cases} \quad u_{1i} = \begin{cases} -\mathbf{x}_i\beta & y_i = 0 \\ \infty & y_i = 1 \end{cases} \quad (1)$$

They let us rewrite the log likelihood concisely as

$$\ln L = \sum_{i=1}^N w_i \ln \Phi_1^*(l_{1i}, u_{1i}, 1)$$

The conditional probability of success can be written using similar notation:

$$\Pr(y_i = 1 | \mathbf{x}_i) = \Phi_1^*(-\mathbf{x}_i \boldsymbol{\beta}, \infty, 1) \quad (2)$$

## Endogenous covariates

### Continuous endogenous covariates

A probit regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and  $C$  continuous endogenous covariates  $\mathbf{w}_{ci}$  has the form

$$y_i = 1 (\mathbf{x}_i \boldsymbol{\beta} + \mathbf{w}_{ci} \boldsymbol{\beta}_c + \epsilon_i > 0)$$

$$\mathbf{w}_{ci} = \mathbf{z}_{ci} \mathbf{A}_c + \boldsymbol{\epsilon}_{ci}$$

The vector  $\mathbf{z}_{ci}$  contains variables from  $\mathbf{x}_i$  and other covariates that affect  $\mathbf{w}_{ci}$ . The unobserved errors  $\epsilon_i$  and  $\boldsymbol{\epsilon}_{ci}$  are multivariate normal with mean 0 and covariance

$$\begin{bmatrix} 1 & \sigma'_{1c} \\ \sigma_{1c} & \boldsymbol{\Sigma}_c \end{bmatrix}$$

We can write the joint density of the dependent variables as a product:

$$f(y_i, \mathbf{w}_{ci} | \mathbf{x}_i, \mathbf{z}_{ci}) = f(y_i | \mathbf{w}_{ci}, \mathbf{x}_i, \mathbf{z}_{ci}) f(\mathbf{w}_{ci} | \mathbf{x}_i, \mathbf{z}_{ci})$$

The conditional density of  $\mathbf{w}_{ci}$  is

$$f(\mathbf{w}_{ci} | \mathbf{x}_i, \mathbf{z}_{ci}) = \phi_C(\mathbf{w}_{ci} - \mathbf{z}_{ci} \mathbf{A}_c, \boldsymbol{\Sigma}_c)$$

Note that

$$\Pr(y_i = 1 | \mathbf{w}_{ci}, \mathbf{x}_i, \mathbf{z}_{ci}) = \Pr(\mathbf{x}_i \boldsymbol{\beta} + \mathbf{w}_{ci} \boldsymbol{\beta}_c + \epsilon_i > 0 | \mathbf{w}_{ci}, \mathbf{x}_i, \mathbf{z}_{ci})$$

So the conditional density of  $y_i$  can be written as a probability for  $\epsilon_i$ . Thus, the conditional distribution of  $\epsilon_i$  can be used to find the conditional density of  $y_i$ . Conditional on the endogenous and exogenous covariates,  $\epsilon_i$  has mean and variance

$$E(\epsilon_i | \mathbf{w}_{ci}, \mathbf{x}_i, \mathbf{z}_{ci}) = \sigma'_{1c} \boldsymbol{\Sigma}_c^{-1} (\mathbf{w}_{ci} - \mathbf{z}_{ci} \mathbf{A}_c)'$$

$$\text{Var}(\epsilon_i | \mathbf{w}_{ci}, \mathbf{x}_i, \mathbf{z}_{ci}) = 1 - \sigma'_{1c} \boldsymbol{\Sigma}_c^{-1} \sigma_{1c}$$

The conditional mean is used in the lower and upper limits for the  $y_i$  probability, which are

$$l_{1i} = \begin{cases} -\infty & y_i = 0 \\ -\mathbf{x}_i\boldsymbol{\beta} - \boldsymbol{\sigma}'_{1c}\boldsymbol{\Sigma}_c^{-1}(\mathbf{w}_{ci} - \mathbf{z}_{ci}\mathbf{A}_c)' & y_i = 1 \end{cases}$$

$$u_{1i} = \begin{cases} -\mathbf{x}_i\boldsymbol{\beta} - \boldsymbol{\sigma}'_{1c}\boldsymbol{\Sigma}_c^{-1}(\mathbf{w}_{ci} - \mathbf{z}_{ci}\mathbf{A}_c)' & y_i = 0 \\ \infty & y_i = 1 \end{cases}$$

Using these limits, the conditional variance, and the conditional density of  $\mathbf{w}_{ci}$ , we obtain the log likelihood

$$\ln L = \sum_{i=1}^N w_i \{ \ln \Phi_1^*(l_{1i}, u_{1i}, 1 - \boldsymbol{\sigma}'_{1c}\boldsymbol{\Sigma}_c^{-1}\boldsymbol{\sigma}_{1c}) + \ln \phi_C(\mathbf{w}_{ci} - \mathbf{z}_{ci}\mathbf{A}_c, \boldsymbol{\Sigma}_c) \}$$

Letting

$$l_{1i1} = -\mathbf{x}_i\boldsymbol{\beta} - \boldsymbol{\sigma}'_{1c}\boldsymbol{\Sigma}_c^{-1}(\mathbf{w}_{ci} - \mathbf{z}_{ci}\mathbf{A}_c)'$$

$$u_{1i1} = \infty$$

the conditional probability of success is

$$\Pr(y_i = 1 | \mathbf{w}_{ci}, \mathbf{x}_i, \mathbf{z}_{ci}) = \Phi_1^*(l_{1i1}, u_{1i1}, 1 - \boldsymbol{\sigma}'_{1c}\boldsymbol{\Sigma}_c^{-1}\boldsymbol{\sigma}_{1c})$$

## Binary and ordinal endogenous covariates

Here, we begin by formulating the probit regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and  $B$  binary and ordinal endogenous covariates  $\mathbf{w}_{bi} = [w_{b1i}, \dots, w_{bBi}]$ . Indicator (dummy) variables for the levels of each binary and ordinal covariate are used in the model. You can also interact other covariates with the binary and ordinal endogenous covariates, as in treatment-effect models.

Let  $j = 1, \dots, B$ . We use a probit model for binary endogenous covariates

$$w_{bji} = 1 (\mathbf{z}_{bji}\boldsymbol{\alpha}_{bj} + \epsilon_{bji} > 0)$$

For ordinal endogenous covariate  $w_{bji}$  that takes values  $v_{bj1}, \dots, v_{bjB_j}$  with covariates  $\mathbf{z}_{bji}$ , we have the ordered probit model

$$w_{bji} = v_{bjh} \quad \text{iff} \quad \kappa_{bj(h-1)} < \mathbf{z}_{bji}\boldsymbol{\alpha}_{bj} + \epsilon_{bji} \leq \kappa_{bjh} \quad (3)$$

The values  $v_{bj1}, \dots, v_{bjB_j}$  are real numbers such that  $v_{bjh} < v_{bjm}$  for  $h < m$ .  $\kappa_{bj0}$  is taken as  $-\infty$  and  $\kappa_{bjB_j}$  is taken as  $+\infty$ . The errors  $\epsilon_{b1i}, \dots, \epsilon_{bBi}$  are multivariate normal with mean 0 and covariance

$$\boldsymbol{\Sigma}_b = \begin{bmatrix} 1 & \rho_{b12} & \cdots & \rho_{b1B} \\ \rho_{b12} & 1 & \cdots & \rho_{b2B} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{b1B} & \rho_{b2B} & \cdots & 1 \end{bmatrix}$$

Because the covariate  $w_{bji}$  is binary or ordinal, the effect of each category in the outcome equation is made with an indicator variable.

$$\mathbf{wind}_{bji} = \begin{bmatrix} 1(w_{bji} = v_{bj1}) \\ \vdots \\ 1(w_{bji} = v_{bjB_j}) \end{bmatrix}' \quad (4)$$

So we have

$$y_i = 1(\mathbf{x}_i\boldsymbol{\beta} + \mathbf{wind}_{b1i}\boldsymbol{\beta}_{b1} + \dots + \mathbf{wind}_{bBi}\boldsymbol{\beta}_{bB} + \epsilon_i > 0)$$

where the outcome error  $\epsilon_i$  and binary and ordinal endogenous errors  $\epsilon_{b1i}, \dots, \epsilon_{bBi}$  are multivariate normal with mean 0 and covariance

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \boldsymbol{\rho}'_{1b} \\ \boldsymbol{\rho}_{1b} & \boldsymbol{\Sigma}_b \end{bmatrix}$$

From here, we discuss the model with ordinal endogenous covariates. The results for binary endogenous covariates are similar.

For  $j = 1, \dots, B$  and  $h = 0, \dots, B_j$ , let

$$c_{bjih} = \begin{cases} -\infty & h = 0 \\ \kappa_{bjh} - \mathbf{z}_{bji}\boldsymbol{\alpha}_{bj} & h = 1, \dots, B_j - 1 \\ \infty & h = B_j \end{cases}$$

The probability for  $w_{bji}$  has lower limit

$$l_{bji} = c_{bjih-1} \quad \text{if } w_{bji} = v_{bjh} \quad (5)$$

and upper limit

$$u_{bji} = c_{bjih} \quad \text{if } w_{bji} = v_{bjh} \quad (6)$$

Letting

$$c_{bi} = -\mathbf{x}_i\boldsymbol{\beta} - \mathbf{wind}_{b1i}\boldsymbol{\beta}_{b1} - \dots - \mathbf{wind}_{bBi}\boldsymbol{\beta}_{bB}$$

the lower and upper limits for the  $y_i$  probability are

$$l_{1i} = \begin{cases} -\infty & y_i = 0 \\ c_{bi} & y_i = 1 \end{cases} \quad u_{1i} = \begin{cases} c_{bi} & y_i = 0 \\ \infty & y_i = 1 \end{cases}$$

and

$$\mathbf{l}_i = [l_{1i} \quad l_{b1i} \quad \dots \quad l_{bBi}] \\ \mathbf{u}_i = [u_{1i} \quad u_{b1i} \quad \dots \quad u_{bBi}]$$

The log likelihood for this model is

$$\ln L = \sum_{i=1}^N w_i \ln \Phi_{B+1}^*(\mathbf{l}_i, \mathbf{u}_i, \boldsymbol{\Sigma})$$

Now, let

$$\begin{aligned}\mathbf{l}_{bi} &= [l_{b1i} \quad \dots \quad l_{bBi}] \\ \mathbf{u}_{bi} &= [u_{b1i} \quad \dots \quad u_{bBi}] \\ \mathbf{l}_{i1} &= [-\infty \quad \mathbf{l}_{bi}] \\ \mathbf{u}_{i1} &= [c_{bi} \quad \mathbf{u}_{bi}]\end{aligned}$$

The conditional probability of success is

$$\Pr(y_i = 1 | \mathbf{x}_i, \mathbf{z}_{b1i}, \dots, \mathbf{z}_{bBi}, \mathbf{w}_{bi}) = \frac{\Phi_{B+1}^*(\mathbf{l}_{i1}, \mathbf{u}_{i1}, \Sigma)}{\Phi_B^*(\mathbf{l}_{bi}, \mathbf{u}_{bi}, \Sigma_b)}$$

## Treatment

In the potential-outcomes framework, the treatment  $t_i$  is a discrete variable taking  $T$  values, indexing the  $T$  potential outcomes of the outcome  $y_i$ :  $y_{1i}, \dots, y_{Ti}$ .

When we observe treatment  $t_i$  with levels  $v_1, \dots, v_T$ , we have

$$y_i = \sum_{j=1}^T 1(t_i = v_j) y_{ji}$$

So for each observation, we only observe the potential outcome associated with that observation's treatment value.

For exogenous treatments, our approach is equivalent to the regression adjustment treatment-effect estimation method. See [TE] [teffects intro advanced](#). We do not model the treatment assignment process. The formulas for the treatment effects and potential-outcome means (POMs) are equivalent to what we provide here for endogenous treatments. The treatment effect on the treated for  $\mathbf{x}_i$  for an exogenous treatment is equivalent to what we provide here for the endogenous treatment when the correlation parameter between the outcome and treatment errors is set to 0. The average treatment effects (ATES) and POMs for exogenous treatments are estimated as predictive margins in an analogous manner to what we describe here for endogenous treatments.

From here, we assume an endogenous treatment  $t_i$ . For ordinal treatment  $t_i$  with covariates  $\mathbf{z}_{ti}$ , we have the ordered probit model

$$t_i = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{z}_{ti}\boldsymbol{\alpha}_t + \epsilon_{ti} \leq \kappa_h \quad (7)$$

The treatment values  $v_1, \dots, v_T$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_T$  is taken as  $+\infty$ . The treatment error  $\epsilon_{ti}$  is standard normal.

We use a probit model for binary treatments that take values in  $\{0, 1\}$ ,

$$t_i = 1 (\mathbf{z}_{ti}\boldsymbol{\alpha}_t + \epsilon_{ti} > 0)$$

A probit regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and endogenous treatment  $t_i$  taking values  $v_1, \dots, v_T$  has the form

$$\begin{aligned}y_{1i} &= 1 (\mathbf{x}_i\boldsymbol{\beta}_1 + \epsilon_{1i} > 0) \\ &\vdots \\ y_{Ti} &= 1 (\mathbf{x}_i\boldsymbol{\beta}_T + \epsilon_{Ti} > 0) \\ y_i &= \sum_{j=1}^T 1(t_i = v_j) y_{ji}\end{aligned}$$

For  $j = 1, \dots, T$ ,  $\epsilon_{ji}$  and  $\epsilon_{ti}$  are bivariate normal with mean 0 and covariance

$$\Sigma = \begin{bmatrix} 1 & \rho_{1t} \\ \rho_{1t} & 1 \end{bmatrix}$$

The treatment is exogenous if  $\rho_{1t} = 0$ . Note that we did not specify the structure of the correlations between the potential-outcome errors. We do not need information about these correlations to estimate POMs and treatment effects because all covariates and the outcome are observed in observations from each group.

From here, we discuss a model with an ordinal endogenous treatment. The results for binary treatment models are similar. Because the unobserved errors are bivariate normal, we can express the log likelihood in terms of the  $\Phi_2^*$  function.

For  $j = 1, \dots, T$ , let

$$c_{1ij} = -\mathbf{x}_i \boldsymbol{\beta}_j$$

The lower and upper limits for the  $y_i$  probability are

$$l_{1i} = \begin{cases} -\infty & y_i = 0 \\ c_{1ij} & y_i = 1, t_i = v_j \end{cases} \quad u_{1i} = \begin{cases} c_{1ij} & y_i = 0, t_i = v_j \\ \infty & y_i = 1 \end{cases}$$

For  $j = 0, \dots, T$ , define

$$c_{tij} = \begin{cases} -\infty & j = 0 \\ \kappa_j - \mathbf{z}_{ti} \boldsymbol{\alpha}_t & j = 1, \dots, T-1 \\ \infty & j = T \end{cases}$$

So for the  $t_i$  probability, we have lower limit

$$l_{ti} = c_{ti(j-1)} \quad \text{if } t_i = v_j \quad (8)$$

and upper limit

$$u_{ti} = c_{tij} \quad \text{if } t_i = v_j \quad (9)$$

The log likelihood for the model is

$$\ln L = \sum_{i=1}^N w_i \ln \Phi_2^*([l_{1i} \quad l_{ti}], [u_{1i} \quad u_{ti}], \Sigma)$$

The conditional probability of obtaining treatment level  $v_h$  is

$$\Pr(t_i = v_h | \mathbf{z}_{ti}) = \Phi_1^*(c_{ti(h-1)}, c_{tih}, 1)$$

The conditional probability of success at treatment level  $v_j$  is

$$\Pr(y_i = 1 | \mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_j) = \frac{\Phi_2^*([c_{1ij} \quad c_{ti(j-1)}], [\infty \quad c_{tij}], \Sigma)}{\Phi_1^*(c_{ti(j-1)}, c_{tij}, 1)}$$

The conditional POM for treatment group  $j$  is

$$\text{POM}_j(\mathbf{x}_i) = E(y_{ji}|\mathbf{x}_i) = \Phi_1^*(c_{1ij}, \infty, 1)$$

Conditional on the covariates  $\mathbf{x}_i$  and  $\mathbf{z}_{ti}$  and the treatment  $t_i = v_h$ , the POM for treatment group  $j$  is

$$\begin{aligned} \text{POM}_j(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_h) &= E(y_{ji}|\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_h) \\ &= \frac{\Phi_2^*([c_{1ij} \quad c_{ti(h-1)}], [\infty \quad c_{tjh}], \Sigma)}{\Phi_1^*(c_{ti(h-1)}, c_{tjh}, 1)} \end{aligned}$$

The treatment effect  $y_{ji} - y_{1i}$  is the difference in the outcome for individual  $i$  if the individual receives the treatment  $t_i = v_j$  instead of the control  $t_i = v_1$  and what the difference would have been if the individual received the control treatment instead.

For treatment group  $j$ , the treatment effect conditioned on  $\mathbf{x}_i$  is

$$\text{TE}_j(\mathbf{x}_i) = E(y_{ji} - y_{1i}|\mathbf{x}_i) = \text{POM}_j(\mathbf{x}_i) - \text{POM}_1(\mathbf{x}_i)$$

For treatment group  $j$ , the treatment effect on the treated (TET) in treatment group  $h$  conditioned on  $\mathbf{x}_i$  and  $\mathbf{z}_{ti}$  is

$$\begin{aligned} \text{TET}_j(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_h) &= E(y_{ji} - y_{1i}|\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_h) \\ &= \text{POM}_j(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_h) - \text{POM}_1(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_h) \end{aligned}$$

We can take the expectation of these conditional predictions over the covariates to get population average parameters. The `margins` command is used to estimate the expectations as predictive margins once the model is fit with `eprbit`. The POM for treatment group  $j$  is

$$\text{POM}_j = E(y_{ji}) = E\{\text{POM}_j(\mathbf{x}_i)\}$$

The ATE for treatment group  $j$  is

$$\text{ATE}_j = E(y_{ji} - y_{1i}) = E\{\text{TE}_j(\mathbf{x}_i)\}$$

For treatment group  $j$ , the average treatment effect on the treated (ATET) in treatment group  $h$  is

$$\begin{aligned} \text{ATET}_{jh} &= E(y_{ji} - y_{1i}|t_i = v_h) \\ &= E\{\text{TET}_j(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_h)|t_i = v_h\} \end{aligned}$$



## Endogenous sample selection

### Probit endogenous sample selection

A probit model for outcome  $y_i$  with selection on  $s_i$  has the form

$$\begin{aligned} y_i &= 1 (\mathbf{x}_i \boldsymbol{\beta} + \epsilon_i > 0) \\ s_i &= 1 (\mathbf{z}_{si} \boldsymbol{\alpha}_s + \epsilon_{si} > 0) \end{aligned}$$

where  $\mathbf{x}_i$  are covariates that affect the outcome and  $\mathbf{z}_{si}$  are covariates that affect selection. The outcome  $y_i$  is observed if  $s_i = 1$  and not observed if  $s_i = 0$ . The unobserved errors  $\epsilon_i$  and  $\epsilon_{si}$  are normal with mean 0 and covariance

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho_{1s} \\ \rho_{1s} & 1 \end{bmatrix}$$

The lower and upper limits for the  $y_i$  probability,  $l_{1i}$  and  $u_{1i}$ , are as defined in (1). For the selection indicator, we have lower and upper limits

$$l_{si} = \begin{cases} -\infty & s_i = 0 \\ -\mathbf{z}_{si} \boldsymbol{\alpha}_s & s_i = 1 \end{cases} \quad u_{si} = \begin{cases} -\mathbf{z}_{si} \boldsymbol{\alpha}_s & s_i = 0 \\ \infty & s_i = 1 \end{cases} \quad (10)$$

The log likelihood for the model is

$$\begin{aligned} \ln L = & \sum_{i \in S} w_i \ln \Phi_2^*([l_{1i} \quad l_{si}], [u_{1i} \quad u_{si}], \boldsymbol{\Sigma}) + \\ & \sum_{i \notin S} w_i \ln \Phi_1^*(l_{si}, u_{si}, 1) \end{aligned}$$

where  $S$  is the set of observations for which  $y_i$  is observed.

In this model, the probability of success is usually predicted conditional on the covariates  $\mathbf{x}_i$  and not on the selection status  $s_i$ . The formulas for the conditional probability are thus the same as in (2).

The conditional probability of selection is

$$\Pr(s_i = 1 | \mathbf{z}_{si}) = \Phi_1^*(-\mathbf{z}_{si} \boldsymbol{\alpha}_s, \infty, 1)$$

### Tobit endogenous sample selection

Instead of constraining the selection indicator to be binary, tobit endogenous sample selection uses a censored continuous sample-selection indicator. We allow the selection variable to be left- or right-censored.

A probit model for outcome  $y_i$  with tobit selection on  $s_i$  has the form

$$y_i = 1 (\mathbf{x}_i \boldsymbol{\beta} + \epsilon_i > 0)$$

We observe the selection indicator  $s_i$ , which indicates the censoring status of the latent selection variable  $s_i^*$ ,

$$s_i^* = \mathbf{z}_{si}\boldsymbol{\alpha}_s + \epsilon_{si}$$

$$s_i = \begin{cases} l_i & s_i^* \leq l_i \\ s_i^* & l_i < s_i^* < u_i \\ u_i & s_i^* \geq u_i \end{cases}$$

where  $\mathbf{z}_{si}$  are covariates that affect selection, and  $l_i$  and  $u_i$  are fixed lower and upper limits.

The outcome  $y_i$  is observed when  $s_i^*$  is not censored ( $l_i < s_i^* < u_i$ ). The outcome  $y_i$  is not observed when  $s_i^*$  is left-censored ( $s_i^* \leq l_i$ ) or  $s_i^*$  is right-censored ( $s_i^* \geq u_i$ ). The unobserved errors  $\epsilon_i$  and  $\epsilon_{si}$  are normal with mean 0 and covariance

$$\begin{bmatrix} 1 & \rho_{1s}\sigma_s \\ \rho_{1s}\sigma_s & \sigma_s^2 \end{bmatrix}$$

For the selected observations, we can treat  $s_i$  as a continuous endogenous regressor, as in *Continuous endogenous covariates*. In fact,  $s_i$  may even be used as a regressor for  $y_i$  in `eprobit` (specify `tobitselect(..., main)`). On the nonselected observations, we treat  $s_i$  like the probit endogenous sample-selection indicator in *Probit endogenous sample selection*.

For nonselected observations, we have

$$\begin{aligned} \Pr(s_i^* \leq l_i | \mathbf{z}_{si}, \mathbf{x}_i) &= \Pr(\mathbf{z}_{si}\boldsymbol{\alpha}_s + \epsilon_{si} \leq l_i) \\ &= \Phi\left(\frac{l_i - \mathbf{z}_{si}\boldsymbol{\alpha}_s}{\sigma_s}\right) \end{aligned}$$

and

$$\begin{aligned} \Pr(s_i^* \geq u_i | \mathbf{z}_{si}, \mathbf{x}_i) &= \Pr(\mathbf{z}_{si}\boldsymbol{\alpha}_s + \epsilon_{si} \geq u_i) \\ &= \Phi\left(\frac{\mathbf{z}_{si}\boldsymbol{\alpha}_s - u_i}{\sigma_s}\right) \end{aligned}$$

The lower and upper limits for the  $s_i$  probability for nonselected observations where  $s_i^*$  is left-censored are

$$\begin{aligned} l_{li} &= -\infty \\ u_{li} &= \frac{l_i - \mathbf{z}_{si}\boldsymbol{\alpha}_s}{\sigma_s} \end{aligned}$$

The lower and upper limits for the  $s_i$  probability for nonselected observations where  $s_i^*$  is right-censored are

$$\begin{aligned} l_{ui} &= \frac{u_i - \mathbf{z}_{si}\boldsymbol{\alpha}_s}{\sigma_s} \\ u_{ui} &= \infty \end{aligned}$$

Now, we consider the selected observations. For  $s_i = s_i^* = S_i$ , we can write the joint density of the dependent variables as a product,

$$f(y_i, s_i = S_i | \mathbf{x}_i, \mathbf{z}_{si}) = f(y_i | s_i = S_i, \mathbf{x}_i, \mathbf{z}_{si}) f(s_i = S_i | \mathbf{x}_i, \mathbf{z}_{si})$$

The marginal density of  $s_i = S_i$  is

$$f(s_i = S_i | \mathbf{x}_i, \mathbf{z}_{si}) = \phi(S_i - \mathbf{z}_{si}\boldsymbol{\alpha}_s, \sigma_s^2)$$

The conditional density of  $y_i$  can be written as a probability for  $\epsilon_i$ . Thus, the conditional distribution of  $\epsilon_i$  can be used to find the conditional density of  $y_i$ . Conditional on  $s_i = S_i$ ,  $\epsilon_i$  has mean and variance

$$\begin{aligned} E(\epsilon_i | s_i = S_i, \mathbf{x}_i, \mathbf{z}_{si}) &= \rho_{1s} \sigma_s^{-1} (S_i - \mathbf{z}_{si}\boldsymbol{\alpha}) \\ \text{Var}(\epsilon_i | s_i = S_i, \mathbf{x}_i, \mathbf{z}_{si}) &= 1 - \rho_{1s}^2 \end{aligned}$$

The conditional mean is used in the lower and upper limits for the  $y_i$  probability for selected observations, which are

$$\begin{aligned} l_{1i} &= \begin{cases} -\infty & y_i = 0 \\ -\mathbf{x}_i\boldsymbol{\beta} - \rho_{1s}\sigma_s^{-1}(s_i - \mathbf{z}_{si}\boldsymbol{\alpha}) & y_i = 1 \end{cases} \\ u_{1i} &= \begin{cases} -\mathbf{x}_i\boldsymbol{\beta} - \rho_{1s}\sigma_s^{-1}(s_i - \mathbf{z}_{si}\boldsymbol{\alpha}) & y_i = 0 \\ \infty & y_i = 1 \end{cases} \end{aligned}$$

It follows that the log likelihood is

$$\begin{aligned} \ln L &= \sum_{i \in S} w_i \{ \ln \Phi_1^*(l_{1i}, u_{1i}, 1 - \rho_{1s}^2) + \ln \phi(s_i - \mathbf{z}_{si}\boldsymbol{\alpha}_s, \sigma_s^2) \} \\ &+ \sum_{i \in L} w_i \ln \Phi_1^*(l_{li}, u_{li}, 1) \\ &+ \sum_{i \in U} w_i \ln \Phi_1^*(l_{ui}, u_{ui}, 1) \end{aligned}$$

where  $S$  is the set of observations for which  $y_i$  is observed,  $L$  is the set of observations where  $s_i^*$  is left-censored, and  $U$  is the set of observations where  $s_i^*$  is right-censored.

The probability of success conditional on  $s_i = s_i^* = S_i$  is

$$\Pr(y_i = 1 | \mathbf{x}_i, s_i = s_i^* = S_i) = \Phi_1^*\{-\mathbf{x}_i\boldsymbol{\beta} - \rho_{1s}\sigma_s^{-1}(S_i - \mathbf{z}_{si}\boldsymbol{\alpha}), \infty, 1 - \rho_{1s}^2\}$$

If we do not include  $s_i$  in the main outcome equation, the probability of success is calculated as (2) again.

## Combined model

The probit model with continuous endogenous covariates, ordinal endogenous covariates, an ordinal endogenous treatment, and endogenous sample selection combines all the extensions to the standard probit model that are supported by **eprobit**. The formulation of other combinations of model features can be easily derived from this combined model. In *Likelihood for multiequation models*, we describe the general framework for ERMs with multiple features. Deriving the combined model with tobit rather than probit endogenous sample selection is straightforward. On selected observations, the selection indicator would be treated like a continuous endogenous covariate. On nonselected observations, the model would be identical to the combined model with probit selection.

In this model, the treatment  $t_i$  takes  $T$  values, indexing the potential outcomes of the main outcome  $y_i$ :  $y_{1i}, \dots, y_{Ti}$ . The relationship between the ordinal treatment  $t_i$ , treatment covariates  $\mathbf{z}_{t,i}$ , and error  $\epsilon_{ti}$  is described in (7). For  $j = 1, \dots, B$ , the relationship between the ordinal endogenous covariates  $w_{bji}$ , exogenous covariates  $\mathbf{z}_{bji}$ , and error  $\epsilon_{bji}$  is given in (3). The model also uses the **wind** <sub>$bji$</sub>  terms that are defined in (4).

The probit regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$ ,  $C$  continuous endogenous covariates  $\mathbf{w}_{ci}$ , and  $B$  ordinal endogenous covariates  $\mathbf{w}_{bi} = [w_{b1i}, \dots, w_{bBi}]$  with endogenous treatment  $t_i$  and endogenous sample selection on  $s_i$  has the form

$$\begin{aligned} y_{1i} &= 1 (\mathbf{x}_i \beta_1 + \mathbf{w}_{ci} \beta_{c1} + \mathbf{wind}_{b1i} \beta_{b11} + \dots + \mathbf{wind}_{bBi} \beta_{bB1} + \epsilon_{1i} > 0) \\ &\vdots \\ y_{Ti} &= 1 (\mathbf{x}_i \beta_T + \mathbf{w}_{ci} \beta_{cT} + \mathbf{wind}_{b1i} \beta_{b1T} + \dots + \mathbf{wind}_{bBi} \beta_{bBT} + \epsilon_{Ti} > 0) \\ y_i &= \sum_{j=1}^T 1(t_i = v_j) y_{ji} \\ \mathbf{w}_{ci} &= \mathbf{z}_{ci} \mathbf{A}_c + \epsilon_{ci} \\ s_i &= 1 (\mathbf{z}_{si} \alpha_s + \epsilon_{si} > 0) \end{aligned}$$

where  $\mathbf{z}_{si}$  are covariates that affect selection and  $\mathbf{z}_{ci}$  are covariates that affect the continuous endogenous covariates. The outcome  $y_i$  is observed if  $s_i = 1$  and is not observed if  $s_i = 0$ .

For  $j = 1, \dots, T$ , the unobserved errors  $\epsilon_{ji}, \epsilon_{si}, \epsilon_{ti}, \epsilon_{b1i}, \dots, \epsilon_{bBi}, \epsilon_{ci}$  are multivariate normal with mean 0 and covariance

$$\Sigma = \begin{bmatrix} 1 & \rho_{1s} & \rho_{1t} & \rho'_{1b} & \sigma'_{1c} \\ \rho_{1s} & 1 & \rho_{st} & \rho'_{sb} & \sigma'_{sc} \\ \rho_{1t} & \rho_{st} & 1 & \rho'_{tb} & \sigma'_{tc} \\ \rho_{1b} & \rho_{sb} & \rho_{tb} & \Sigma_b & \Sigma'_{bc} \\ \sigma_{1c} & \sigma_{sc} & \sigma_{tc} & \Sigma_{bc} & \Sigma_c \end{bmatrix}$$

As in *Continuous endogenous covariates*, we can write the joint density of the dependent variables as a product. We have

$$\begin{aligned} f(y_i, s_i, t_i, \mathbf{w}_{bi}, \mathbf{w}_{ci} | \mathbf{x}_i, \mathbf{z}_{si}, \mathbf{z}_{ti}, \mathbf{z}_{b1i}, \dots, \mathbf{z}_{bBi}, \mathbf{z}_{ci}) = \\ f(y_i, s_i, t_i, \mathbf{w}_{bi} | \mathbf{w}_{ci}, \mathbf{x}_i, \mathbf{z}_{si}, \mathbf{z}_{ti}, \mathbf{z}_{b1i}, \dots, \mathbf{z}_{bBi}, \mathbf{z}_{ci}) f(\mathbf{w}_{ci} | \mathbf{z}_{ci}) \end{aligned}$$

We can then use the conditional distribution of  $\epsilon_{ji}, \epsilon_{si}, \epsilon_{ti}, \epsilon_{b1i}, \dots, \epsilon_{bBi}$  to obtain the conditional density of  $y_i, s_i, t_i$ , and  $\mathbf{w}_{bi}$ .

For  $j = 1, \dots, T$ , conditional on  $\mathbf{w}_{ci}$  and the exogenous covariates,  $\epsilon_{ji}$  has mean

$$\begin{aligned} e_{1i} &= E(\epsilon_{ji} | \mathbf{w}_{ci}, \mathbf{x}_i, \mathbf{z}_{si}, \mathbf{z}_{ti}, \mathbf{z}_{b1i}, \dots, \mathbf{z}_{bBi}, \mathbf{z}_{ci}) \\ &= \sigma'_{1,c} \Sigma_c^{-1} (\mathbf{w}_{ci} - \mathbf{z}_{c,i} \mathbf{A}_c)' \end{aligned}$$

Now, for  $j = 1, \dots, T$ , let

$$c_{1ij} = \begin{cases} -\mathbf{x}_i \beta_1 - \mathbf{w}_{ci} \beta_{c,1} - \mathbf{wind}_{b1i} \beta_{b11} - \dots - \mathbf{wind}_{bBi} \beta_{bB1} - e_{1i} & j = 1 \\ \vdots \\ -\mathbf{x}_i \beta_T - \mathbf{w}_{ci} \beta_{cT} - \mathbf{wind}_{b1i} \beta_{b1T} - \dots - \mathbf{wind}_{bBi} \beta_{bBT} - e_{1i} & j = T \end{cases}$$

The lower and upper limits for the  $y_i$  probability are

$$l_{1i} = \begin{cases} -\infty & y_i = 0 \\ c_{1ij} & y_i = 1, t_i = v_j \end{cases} \quad u_{1i} = \begin{cases} c_{1ij} & y_i = 0, t_i = v_j \\ \infty & y_i = 1 \end{cases}$$

The conditional means of the unobserved errors  $\epsilon_{si}, \epsilon_{ti}, \epsilon_{b1i}, \dots, \epsilon_{bBi}$  have similar forms to  $e_{1i}$ . Denote these means by  $e_{si}, e_{ti}, e_{b1i}, \dots, e_{bBi}$ . The lower and upper probability limits for  $s_i, t_i$ , and the ordinal endogenous covariates are obtained by subtracting the means from the limits defined in (10), (8), (9), (5), and (6).

$$\begin{aligned} l_{si}^* &= l_{si} - e_{si} \\ u_{si}^* &= u_{si} - e_{si} \\ l_{ti}^* &= l_{ti} - e_{ti} \\ u_{ti}^* &= u_{ti} - e_{ti} \\ l_{b1i}^* &= l_{b1i} - e_{b1i} \\ u_{b1i}^* &= u_{b1i} - e_{b1i} \\ &\vdots \\ l_{bBi}^* &= l_{bBi} - e_{bBi} \\ u_{bBi}^* &= u_{bBi} - e_{bBi} \end{aligned}$$

We have lower and upper limits; we need a conditional covariance and the conditional density of  $\mathbf{w}_{ci}$  to formulate the likelihood. For  $j = 1, \dots, T$ , conditional on  $\mathbf{w}_{ci}$  and the exogenous covariates,  $\epsilon_{ji}, \epsilon_{si}, \epsilon_{ti}, \epsilon_{b1i}, \dots, \epsilon_{bBi}$  have covariance

$$\Sigma_{o|c} = \begin{bmatrix} 1 & \rho_{1s} & \rho_{1t} & \rho'_{1b} \\ \rho_{1s} & 1 & \rho_{st} & \rho'_{sb} \\ \rho_{1t} & \rho_{st} & 1 & \rho'_{tb} \\ \rho_{1b} & \rho_{sb} & \rho_{tb} & \Sigma_b \end{bmatrix} - \begin{bmatrix} \sigma'_{1c} \\ \sigma'_{sc} \\ \sigma'_{tc} \\ \Sigma'_{bc} \end{bmatrix} \Sigma_c^{-1} \begin{bmatrix} \sigma'_{1c} \\ \sigma'_{sc} \\ \sigma'_{tc} \\ \Sigma'_{bc} \end{bmatrix}'$$

The conditional density of  $\mathbf{w}_{ci}$  is

$$f(\mathbf{w}_{ci} | \mathbf{z}_{ci}) = \phi_C(\mathbf{w}_{ci} - \mathbf{z}_{ci} \mathbf{A}_c, \Sigma_c)$$

Let

$$\begin{aligned} \mathbf{l}_{1i} &= [l_{1i} \quad l_{si}^* \quad l_{ti}^* \quad l_{b1i}^* \quad \dots \quad l_{bBi}^*] \\ \mathbf{u}_{1i} &= [u_{1i} \quad u_{si}^* \quad u_{ti}^* \quad u_{b1i}^* \quad \dots \quad u_{bBi}^*] \\ \mathbf{l}_i &= [l_{si}^* \quad l_{ti}^* \quad l_{b1i}^* \quad \dots \quad l_{bBi}^*] \\ \mathbf{u}_i &= [u_{si}^* \quad u_{ti}^* \quad u_{b1i}^* \quad \dots \quad u_{bBi}^*] \end{aligned}$$

The log likelihood of the model is

$$\begin{aligned} \ln L = & \sum_{i \in S} w_i \ln \Phi_{3+B}^* (\mathbf{l}_{1i}, \mathbf{u}_{1i}, \boldsymbol{\Sigma}_{o|c}) + \\ & \sum_{i \notin S} w_i \ln \Phi_{2+B}^* (\mathbf{l}_i, \mathbf{u}_i, \boldsymbol{\Sigma}_{o|c,-1}) + \\ & \sum_{i=1}^N w_i \ln \phi_C(\mathbf{w}_{ci} - \mathbf{z}_{ci} \mathbf{A}_c, \boldsymbol{\Sigma}_c) \end{aligned}$$

where  $S$  is the set of observations where  $y_i$  is observed, and  $\boldsymbol{\Sigma}_{o|c,-1}$  is  $\boldsymbol{\Sigma}_{o|c}$  with the first row and column removed.

As in previous sections, we use the joint and marginal probabilities to determine conditional probabilities.

For  $j = 1, \dots, T$  and  $i$  such that  $t_i = v_j$ , let

$$\begin{aligned} \mathbf{l}_{i11} &= [c_{1ij} \quad l_{ti}^* \quad l_{b1i}^* \quad \dots \quad l_{bBi}^*] \\ \mathbf{u}_{i11} &= [\infty \quad u_{ti}^* \quad u_{b1i}^* \quad \dots \quad u_{bBi}^*] \\ \mathbf{l}_{i12} &= [l_{ti}^* \quad l_{b1i}^* \quad \dots \quad l_{bBi}^*] \\ \mathbf{u}_{i12} &= [u_{ti}^* \quad u_{b1i}^* \quad \dots \quad u_{bBi}^*] \end{aligned}$$

Let  $\boldsymbol{\Sigma}_{o|c,-s}$  be  $\boldsymbol{\Sigma}_{o|c}$  with the second row and column removed. This is the conditional covariance matrix without the endogenous sample-selection equation components. Let  $\boldsymbol{\Sigma}_{o|c,-s-1}$  be  $\boldsymbol{\Sigma}_{o|c,-s}$  with the first row and column removed.

The conditional probability of success at treatment level  $t_i = v_j$  is

$$\Pr(y_i = 1 | \mathbf{t}_i = v_j, \mathbf{w}_{bi}, \mathbf{w}_{ci}, \mathbf{x}_i, \mathbf{z}_{si}, \mathbf{z}_{ti}, \mathbf{z}_{b1i}, \dots, \mathbf{z}_{bBi}, \mathbf{z}_{ci}) = \frac{\Phi_{2+B}^* (\mathbf{l}_{i11}, \mathbf{u}_{i11}, \boldsymbol{\Sigma}_{o|c,-s})}{\Phi_{1+B}^* (\mathbf{l}_{i12}, \mathbf{u}_{i12}, \boldsymbol{\Sigma}_{o|c,-s-1})}$$

The conditional probabilities of treatment, selection, and the ordinal endogenous covariates are derived in similar ways. We condition on the treatment and the other endogenous covariates together with the exogenous covariates that affect the outcome. POMs and treatment effects are conditioned on the endogenous and exogenous covariates. See [Predictions considering total effects](#) in [ERM] **eprobit postestimation** for more details.

## Confidence intervals

The estimated variances will always be nonnegative, and the estimated correlations will always fall in  $(-1, 1)$ . We use transformations to obtain confidence intervals that accommodate these ranges.

We use the log transformation to obtain the confidence intervals for variance parameters. Let  $\hat{\sigma}^2$  be a point estimate for the variance parameter  $\sigma^2$ , and let  $\widehat{\text{SE}}(\hat{\sigma}^2)$  be its standard error. The  $(1 - \alpha) \times 100\%$  confidence interval for  $\ln(\sigma^2)$  is

$$\ln(\hat{\sigma}^2) \pm z_{\alpha/2} \frac{\widehat{\text{SE}}(\hat{\sigma}^2)}{\hat{\sigma}^2}$$

where  $z_{\alpha/2}$  is the  $1 - \alpha/2$  quantile of the standard normal distribution. Let  $k_u$  be the upper endpoint of this interval, and let  $k_l$  be the lower. The  $(1 - \alpha) \times 100\%$  confidence interval for  $\sigma^2$  is then given by

$$(e^{k_l}, e^{k_u})$$

We use the inverse hyperbolic tangent transformation to obtain confidence intervals for correlation parameters; for details on the hyperbolic functions, see [FN] **Trigonometric functions**. Let  $\hat{\rho}$  be a point estimate for the correlation parameter  $\rho$ , and let  $\widehat{\text{SE}}(\hat{\rho})$  be its standard error. The  $(1 - \alpha) \times 100\%$  confidence interval for  $\text{atanh}(\rho)$  is

$$\text{atanh}(\hat{\rho}) \pm z_{\alpha/2} \widehat{\text{SE}}(\hat{\rho}) \frac{1}{1 - \hat{\rho}^2}$$

where  $z_{\alpha/2}$  is the  $1 - \alpha/2$  quantile of the standard normal distribution. Let  $k_u$  be the upper endpoint of this interval, and let  $k_l$  be the lower. The  $(1 - \alpha) \times 100\%$  confidence interval for  $\rho$  is then given by

$$\{\tanh(k_l), \tanh(k_u)\}$$

## Likelihood for multiequation models

The general framework for ERMs is formulated such that it accommodates multiple features. Binary and ordinal endogenous covariates may occur together with continuous endogenous covariates in ERMs. Endogenous covariates may also occur together with endogenous sample selection or treatments in ERMs.

Here, we show how the log likelihood is formulated when we have multiple auxiliary equations.

Suppose that we have  $H$  auxiliary equations with endogenous outcomes  $y_{1i}, \dots, y_{Hi}$ . We will treat the main outcome  $y_i$  as stage  $J = H + 1$ , so  $y_{Ji} = y_i$ . The ERMs that we fit with **eintreg**, **eoprobit**, **eprobit**, and **eregress** are triangular, so we can order the equations such that the first depends only on exogenous covariates—say,  $\mathbf{w}_{1i} = \mathbf{z}_i$ —and for  $j = 2, \dots, J$ , equation  $j$  depends only on the exogenous covariates  $\mathbf{z}_i$  and the endogenous covariates from equation  $h = j - 1$  and below  $y_{1i}, \dots, y_{hi}$ . These are stored together in  $\mathbf{w}_{ji}$ .

So we have

$$\begin{aligned} y_{1i} &= g_{1i}(\mathbf{w}_{1i}\boldsymbol{\beta}_1 + v_{1i}) \\ &\vdots \\ y_{Hi} &= g_{Hi}(\mathbf{w}_{Hi}\boldsymbol{\beta}_H + v_{Hi}) \\ y_i = y_{Ji} &= g_{Ji}(\mathbf{w}_{Ji}\boldsymbol{\beta}_J + v_{Ji} > 0) \end{aligned}$$

where the form of the functions  $g_{ji}(\cdot)$  is determined by whether the outcome  $y_{ji}$  has a linear, probit, or interval model. The errors  $v_{1i}, \dots, v_{Ji}$  are multivariate normal with mean 0 and covariance  $\boldsymbol{\Sigma}$ .

The covariates  $\mathbf{w}_{ji}$  and the outcome  $y_{ji}$  determine a range for the error  $v_{ji}$ . For example, if  $y_{ji}$  has a linear model, then  $v_{ji} = y_{ji} - \mathbf{w}_{ji}\boldsymbol{\beta}_j$ , the residual. If  $y_{ji} = 1$  and  $y_{ji}$  has a probit model, then  $v_{ji}$  is in the range  $(-\mathbf{w}_{ji}\boldsymbol{\beta}_j, \infty)$ . If  $y_{ij}$  is left-censored at  $l_i$ , then  $v_{ji}$  is in the range  $(-\infty, l_i - \mathbf{w}_{ji}\boldsymbol{\beta}_j)$ .

The density of the endogenous variables can be represented using a multivariate normal density function that is evaluated at the residuals for the continuous outcomes and integrated over the error ranges of the noncontinuous outcomes.

The conditional density of the error  $v_{ji}$  on  $\mathbf{w}_{ji}$  has the form

$$f(v_{ji}|\mathbf{w}_{ji}) = \frac{\int_{\mathbf{V}_{hi}^*} \phi_j(v_{1i}, \dots, v_{ji}, \boldsymbol{\Sigma}_j) d\mathbf{v}_{hi}^*}{\int_{\mathbf{V}_{hi}^*} \phi_h(v_{1i}, \dots, v_{hi}, \boldsymbol{\Sigma}_h) d\mathbf{v}_{hi}^*}$$

where  $\boldsymbol{\Sigma}_j$  is the covariance of  $v_{1i}, \dots, v_{ji}$  and  $\boldsymbol{\Sigma}_h$  is the covariance of  $v_{1i}, \dots, v_{hi}$  where  $h = j - 1$ . The vector  $\mathbf{v}_{hi}^*$  contains the errors that correspond to binary, ordinal, or censored outcomes in  $y_{1i}, \dots, y_{hi}$ . These outcomes induce the error ranges  $\mathbf{V}_{hi}^*$ , which we integrate over. The other errors are determined by the outcomes and covariates as residuals.

If  $y_{ji}$  is continuous, then

$$f(y_{ji}|\mathbf{w}_{ji}) = f(v_{ji}|\mathbf{w}_{ji}) \quad (11)$$

When  $y_{ji}$  is a binary, ordinal, or censored outcome, we have

$$f(y_{ji}|\mathbf{w}_{ji}) = \frac{\int_{\mathbf{V}_{ji}^*} \phi_j(v_{1i}, \dots, v_{ji}, \boldsymbol{\Sigma}_j) d\mathbf{v}_{ji}^*}{\int_{\mathbf{V}_{hi}^*} \phi_h(v_{1i}, \dots, v_{hi}, \boldsymbol{\Sigma}_h) d\mathbf{v}_{hi}^*} \quad (12)$$

So we also integrate over the range of the error  $v_{ji}$  when  $y_{ji}$  is not continuous.

We can express the joint density of the main outcome and the endogenous covariates in terms of the marginal and conditional densities. The denominator in (11) or (12) in the higher stage will cancel out the numerator of (11) or (12) in the lower stage, so we have

$$f(y_{1i}, \dots, y_{ji}|\mathbf{z}_i) = \int_{\mathbf{V}_{ji}^*} \phi_j(v_{1i}, \dots, v_{ji}, \boldsymbol{\Sigma}_j) d\mathbf{v}_{ji}^* \quad (13)$$

If we only have continuous endogenous variables, we have

$$f(y_{1i}, \dots, y_{ji}|\mathbf{z}_i) = \phi_j(v_{1i}, \dots, v_{ji}, \boldsymbol{\Sigma}_j)$$

If  $\mathbf{V}_{ji}^*$  has dimension  $j$ , we can calculate the integral given in (13) by using the  $\Phi_j^*$ . Let  $\mathbf{l}_i$  contain the lower endpoints and  $\mathbf{u}_i$  contain the upper endpoints for  $\mathbf{V}_{ji}^*$ . When we do not have continuous endogenous covariates, we have

$$f(y_{1i}, \dots, y_{ji}|\mathbf{z}_i) = \Phi_j^*(\mathbf{l}_i, \mathbf{u}_i, \boldsymbol{\Sigma}_j)$$

Now, suppose that we have  $C < j$  continuous outcomes in  $y_{1i}, \dots, y_{ji}$ , so the dimension of  $\mathbf{V}_{ji}^*$  is  $j - C$ . Without loss of generality, these  $C$  correspond to the last  $C$  endogenous covariates  $y_{(j-C+1)i}, \dots, y_{ji}$ . The covariates can be reordered as needed.

We partition the covariance

$$\boldsymbol{\Sigma}_j = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}'_{12} \\ \boldsymbol{\Sigma}_{12} & \boldsymbol{\Sigma}_{22} \end{bmatrix}$$

where  $\boldsymbol{\Sigma}_{22}$  is the covariance of the last  $C$  errors.

Conditional on  $v_{(j-C+1)i}, \dots, v_{ji}$ , the errors  $v_{1i}, \dots, v_{(j-C)i}$  have mean and variance

$$\begin{aligned} \boldsymbol{\mu}_{1|2,i} &= \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} \begin{bmatrix} v_{(j-C+1)i} \\ \vdots \\ v_{ji} \end{bmatrix} \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} \boldsymbol{\Sigma}'_{12} \end{aligned}$$



By conditioning on  $v_{(j-C+1)i}, \dots, v_{ji}$ , we can express the density in terms of  $\phi_C$  and  $\Phi_{j-C}^*$ . We can write the joint density in terms of the marginal and conditional densities to obtain

$$f(y_{1i}, \dots, y_{ji} | \mathbf{z}_i) = \phi_C(v_{(j-C+1)i}, \dots, v_{ji}, \Sigma_{22}) \Phi_{j-C}^*(\mathbf{l}_i - \boldsymbol{\mu}_{1|2,i}, \mathbf{u}_i - \boldsymbol{\mu}_{1|2,i}, \Sigma_{1|2})$$

The natural logarithm of the density  $f(y_{1i}, \dots, y_{ji} | \mathbf{z}_i)$  is the log likelihood of the model. We maximize the log likelihood to estimate the model parameters.

## References

- Angrist, J. D. 2001. Estimation of limited dependent variable models with dummy endogenous regressors: Simple strategies for empirical practice. *Journal of Business and Economic Statistics* 19: 2–16.
- Arendt, J. N., and A. Holm. 2006. Probit models with binary endogenous regressors. Discussion Papers on Business and Economics No. 4/2006, Department of Business and Economics, University of Southern Denmark. [static.sdu.dk/mediafiles/Files/Om\\_SDU/Institutter/Ivoe/Disc\\_papers/Disc\\_2006/dpbe4\\_2006.pdf.pdf](http://static.sdu.dk/mediafiles/Files/Om_SDU/Institutter/Ivoe/Disc_papers/Disc_2006/dpbe4_2006.pdf.pdf).
- Drezner, Z. 1994. Computation of the trivariate normal integral. *Mathematics of Computation* 62: 289–294.
- Karymshakov, K., K. Sultakeev, and B. Sulaimanova. 2015. The impact of microfinance on entrepreneurship in Kyrgyzstan. *International Conference on Eurasian Economies*, paper ID 1412. Eurasian Economists Association: Kazan, Russia. <http://www.avekon.org/papers/1412.pdf>.
- Miwa, T., A. J. Hayter, and S. Kuriki. 2003. The evaluation of general non-centred orthant probabilities. *Journal of the Royal Statistical Society, Series B* 65: 223–234.
- Miwa, T., A. J. Hayter, and W. Liu. 2000. Calculations of level probabilities for normal random variables with unequal variances with applications to Bartholomew’s test in unbalanced one-way models. *Computational Statistics & Data Analysis* 34: 17–32.
- Newey, W. K. 1987. Efficient estimation of limited dependent variable models with endogenous explanatory variables. *Journal of Econometrics* 36: 231–250.
- Pindyck, R. S., and D. L. Rubinfeld. 1998. *Econometric Models and Economic Forecasts*. 4th ed. New York: McGraw–Hill.
- Roodman, D. 2011. Fitting fully observed recursive mixed-process models with `cmp`. *Stata Journal* 11: 159–206.
- Vahter, P. 2011. Does FDI spur productivity, knowledge sourcing and innovation by incumbent firms? Evidence from manufacturing industry in Estonia. *World Economy* 34: 1308–1326.
- Van de Ven, W. P. M. M., and B. M. S. Van Praag. 1981. The demand for deductibles in private health insurance: A probit model with sample selection. *Journal of Econometrics* 17: 229–252.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

- [ERM] **eprobit postestimation** — Postestimation tools for `eprobit`
- [ERM] **eprobit predict** — `predict` after `eprobit`
- [ERM] **estat teffects** — Average treatment effects for extended regression models
- [ERM] **intro 8** — Conceptual introduction via worked example
- [R] **biprobit** — Bivariate probit regression
- [R] **heckprobit** — Probit model with sample selection
- [R] **ivprobit** — Probit model with continuous endogenous covariates
- [R] **probit** — Probit regression
- [SVY] **svy estimation** — Estimation commands for survey data
- [U] **20 Estimation and postestimation commands**

Postestimation commands  
Methods and formulas

predict  
References

margins  
Also see

Remarks and examples

Postestimation commands

The following postestimation command is of special interest after `eprobit`:

Command	Description
<code>estat teffects</code>	treatment effects and potential-outcome means

The following standard postestimation commands are also available after `eprobit`:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of estimates
<code>estat ic</code>	Akaike’s and Schwarz’s Bayesian information criteria (AIC and BIC)
<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>forecast</code>	dynamic forecasts and simulations
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
* <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 coefficients
<code>predict</code>	predictions, residuals, influence statistics, and other diagnostic measures
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of estimates
<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

\* `forecast` and `lrtest` are not appropriate with `svy` estimation results.

# predict

Predictions after `eprobit` are described in

- [\[ERM\] eprobit predict](#) predict after eprobit
- [\[ERM\] predict treatment](#) predict for treatment statistics
- [\[ERM\] predict advanced](#) predict's advanced features

[\[ERM\] eprobit predict](#) describes the most commonly used predictions. If you fit a model with treatment effects, predictions specifically related to these models are detailed in [\[ERM\] predict treatment](#). [\[ERM\] predict advanced](#) describes less commonly used predictions, such as predictions of outcomes in auxiliary equations.

# margins

## Description for margins

`margins` estimates margins of response for probabilities, means, potential-outcome means, treatment effects, and linear predictions.

## Menu for margins

Statistics > Postestimation

## Syntax for margins

```
margins [marginlist] [ , options ]
margins [marginlist] , predict(statistic ...) [predict(statistic ...) ...] [options]
```

statistic	Description
Main	
<u>pr</u>	probability for binary or ordinal $y_j$ ; the default
<u>mean</u>	mean
<u>pomean</u>	potential-outcome mean
<u>te</u>	treatment effect
<u>tet</u>	treatment effect on the treated
<u>xb</u>	linear prediction
<u>pr</u> ( $a,b$ )	$\Pr(a < y_j < b)$ for continuous $y_j$
<u>e</u> ( $a,b$ )	$E(y_j   a < y_j < b)$ for continuous $y_j$
<u>ystar</u> ( $a,b$ )	$E(y_j^*), y_j^* = \max\{a, \min(y_j, b)\}$ for continuous $y_j$

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.  
For the full syntax, see [\[R\] margins](#).

## Remarks and examples

See [ERM] [intro 6](#) for an overview of using margins and predict after eprobit. For examples using margins, predict, and estat teffects, see [Interpreting effects](#) in [ERM] [intro 8](#) and see [ERM] [example 1a](#).

## Methods and formulas

These methods build on the discussions in [Methods and formulas](#) of [ERM] [eprobit](#).

Methods and formulas are presented under the following headings:

*Counterfactual predictions and inferences*  
*Predictions using the full model*

## Counterfactual predictions and inferences

In [Methods and formulas](#) of [ERM] [eprobit](#), we discussed how treatment effects are evaluated in extended probit regression models. Here, we discuss the counterfactual framework used to evaluate the effects of other changes to covariates.

In the extended probit regression model for  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and  $\mathbf{w}_i$ , we partition each set of covariates into two groups. The exogenous covariates  $\mathbf{x}_i$  are partitioned into  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^{nc}$ , where we are interested in the effect of changes in  $\mathbf{x}_i^c$ . Similarly, the endogenous covariates  $\mathbf{w}_i$  are partitioned into  $\mathbf{w}_i^c$  and  $\mathbf{w}_i^{nc}$ , where the effect of changes in  $\mathbf{w}_i^c$  is of interest. The superscripts indicate what is a counterfactual value ( $c$ ) and what is not ( $nc$ ).

If  $\mathbf{x}_i^c = \mathbf{a}_0$  and  $\mathbf{w}_i^c = \mathbf{a}_{20}$ , for covariates  $\mathbf{w}_i^{nc}$  and  $\mathbf{x}_i^{nc}$  we would observe the outcome

$$\begin{aligned} y_{0i} &= 1(\beta_{0nc}\mathbf{x}_i^{nc} + \beta_{20nc}\mathbf{w}_i^{nc} + \beta_c\mathbf{a}_0 + \beta_{2c}\mathbf{a}_{20} + \epsilon_{0i} > 0) \\ &= 1(\beta_{0nc}\mathbf{x}_i^{nc} + \beta_{20nc}\mathbf{w}_i^{nc} + \beta_{c0} + \epsilon_{0i} > 0) \end{aligned}$$

where the unobserved error  $\epsilon_{0i}$  is standard normal. We treat  $\beta_c\mathbf{a}_0 + \beta_{2c}\mathbf{a}_{20} = \beta_{c0}$  as a constant intercept, because it is the same for each value combination of the covariates  $\mathbf{w}_i^{nc}$  and  $\mathbf{x}_i^{nc}$  and the error  $\epsilon_{0i}$ .

Similarly, if  $\mathbf{x}_i^c = \mathbf{a}_1$  and  $\mathbf{w}_i^c = \mathbf{a}_{21}$ , for covariates  $\mathbf{w}_i^{nc}$  and  $\mathbf{x}_i^{nc}$  we would observe the outcome

$$\begin{aligned} y_{1i} &= 1(\beta_{1nc}\mathbf{x}_i^{nc} + \beta_{21nc}\mathbf{w}_i^{nc} + \beta_c\mathbf{a}_1 + \beta_{2c}\mathbf{a}_{21} + \epsilon_{1i} > 0) \\ &= 1(\beta_{1nc}\mathbf{x}_i^{nc} + \beta_{21nc}\mathbf{w}_i^{nc} + \beta_{c1} + \epsilon_{1i} > 0) \end{aligned}$$

The effect of changing  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$  from  $\mathbf{a}_0$  and  $\mathbf{a}_{20}$  to  $\mathbf{a}_1$  and  $\mathbf{a}_{21}$  on  $y_i$  is the expected difference between  $y_{1i}$  and  $y_{0i}$ .

To obtain this difference, we average the conditional probabilities of  $y_{1i}$  and  $y_{0i}$  as a predictive margin.

For  $j = 0, 1$ , we can predict the counterfactual probability for group  $j$  by using the tools discussed in [Predictions using the full model](#),

$$\text{CP}_j(\mathbf{w}_i^{nc}, \mathbf{x}_i^{nc}, \mathbf{z}_i) = \Pr(y_{ji} = 1 | \mathbf{w}_i^{nc}, \mathbf{x}_i^{nc}, \mathbf{x}_i^c = \mathbf{a}_j, \mathbf{z}_i)$$

where  $\mathbf{z}_i$  are instruments necessary for modeling the endogenous regressors  $\mathbf{w}_i^{nc}$ . By the law of iterated expectations, we have

$$E(y_{1i} - y_{0i}) = E\{\text{CP}_1(\mathbf{w}_i^{nc}, \mathbf{x}_i^{nc}, \mathbf{z}_i)\} - E\{\text{CP}_0(\mathbf{w}_i^{nc}, \mathbf{x}_i^{nc}, \mathbf{z}_i)\}$$

So the effect of changing  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$  from  $\mathbf{a}_0$  and  $\mathbf{a}_{20}$  to  $\mathbf{a}_1$  and  $\mathbf{a}_{21}$  can be estimated as a predictive margin on the counterfactual probabilities.

We can use `predict` with the `fix()` and `target()` options to predict the counterfactual probabilities. The `fix()` option is used to indicate the endogenous covariates in  $\mathbf{w}_i^c$ . The `target()` option can be used to set the counterfactual values  $a_j$  and  $a_{2j}$  of  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$ .

When  $\mathbf{w}_i^c$  corresponds to a single ordinal or binary regressor, the difference in counterfactual probabilities corresponds to a treatment effect of  $\mathbf{w}_i^c$ . We can also evaluate the structural effect of a change in  $\mathbf{w}_i^c$  and  $\mathbf{x}_i^c$ , conditioned on  $\mathbf{w}_i^c$ . This effect is analogous to the treatment effect on the treated discussed in [Methods and formulas](#) of [ERM] **eprobit**. We are conditioning the effect on some base value for  $\mathbf{w}_i^c$ ,  $\mathbf{w}_i^c = \mathbf{b}$ .

Now, the counterfactual probabilities are conditioned on  $\mathbf{w}_i^c = \mathbf{b}$ . So for  $j = 0, 1$ , we have

$$\text{CP}_{bj}(\mathbf{w}_i^{nc}, \mathbf{b}, \mathbf{x}_i^{nc}, \mathbf{z}_i) = \Pr(y_{ji} = 1 | \mathbf{w}_i^{nc}, \mathbf{w}_i^c = \mathbf{b}, \mathbf{x}_i^{nc}, \mathbf{x}_i^c = \mathbf{a}_j, \mathbf{z}_{bi})$$

where  $\mathbf{z}_{bi}$  are instruments necessary for modeling the endogenous regressors  $\mathbf{w}_i^{nc}$  and  $\mathbf{w}_i^c$ . This counterfactual probability can be evaluated using the tools discussed in [Predictions using the full model](#).

By the law of iterated expectations, we have

$$\begin{aligned} E(y_{1i} - y_{0i} | \mathbf{w}_i^c = \mathbf{b}) &= E\{\text{CP}_{b1}(\mathbf{w}_i^{nc}, \mathbf{w}_i^c = \mathbf{b}, \mathbf{x}_i^{nc}, \mathbf{z}_i) | \mathbf{w}_i^c = \mathbf{b}\} - \\ &E\{\text{CP}_{b0}(\mathbf{w}_i^{nc}, \mathbf{w}_i^c = \mathbf{b}, \mathbf{x}_i^{nc}, \mathbf{z}_i) | \mathbf{w}_i^c = \mathbf{b}\} \end{aligned}$$

So the effect of changing  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$  from  $\mathbf{a}_0$  and  $\mathbf{a}_{20}$  to  $\mathbf{a}_1$  and  $\mathbf{a}_{21}$  conditioned on  $\mathbf{w}_i^c = \mathbf{b}$  can be estimated as a predictive margin on the counterfactual probabilities.

The base values  $\mathbf{b}$  for  $\mathbf{w}_i^c$  are specified in the `base()` option. As before, `target()` can be used to specify the counterfactual values for  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$ .

When  $\mathbf{x}_i^c = \mathbf{x}_i$  and  $\mathbf{w}_i^c = \mathbf{w}_i$ , the counterfactual probability matches the average structural probability (ASP). Applying the average structural function (ASF) discussed by [Blundell and Powell \(2003\)](#), [Blundell and Powell \(2004\)](#), [Wooldridge \(2005\)](#), and [Wooldridge \(2014\)](#) to a conditional probability on the covariates and unobserved endogenous error produces the ASP.

In the probit model, for exogenous covariates  $\mathbf{x}_i$  and endogenous covariates  $\mathbf{w}_i$ , we have

$$y_i = \mathbf{1}(\mathbf{x}_i\boldsymbol{\beta} + \mathbf{w}_i\boldsymbol{\beta}_2 + \epsilon_i > 0)$$

where  $\epsilon_i$  is a standard normal error.

The ASP provides a structural interpretation of  $\boldsymbol{\beta}$  and  $\boldsymbol{\beta}_2$  when the  $\mathbf{w}_i$  are correlated with  $\epsilon_i$ . Because  $\epsilon_i$  is a normally distributed, mean 0, random variable, we can split it into two mean 0, normally distributed, independent parts,

$$\epsilon_i = u_i + \psi_i$$

where  $u_i = \gamma\epsilon_{2i}$  is the unobserved heterogeneity that gives rise to the endogeneity and  $\psi_i$  is an error term with variance  $\sigma_\psi^2$ . Conditional on the covariates and the unobserved heterogeneity, the probability that  $y_i = 1$  is

$$\Pr(y_i = 1 | \mathbf{x}_i, \mathbf{w}_i, u_i) = \Phi\left(\frac{\mathbf{x}_i\boldsymbol{\beta} + \mathbf{w}_i\boldsymbol{\beta}_2 + u_i}{\sigma_\psi}\right)$$

Because  $u_i$  is an unobserved random variable, this conditional probability is not observable. Integrating out the  $u_i$ , just like we do with random effects in panel-data models, produces the ASP,

$$\text{ASP}(\mathbf{x}_i^0, \mathbf{w}_i^0) = \int \Pr(y_i = 1 | \mathbf{x}_i^0, \mathbf{w}_i^0, u_i) f(u_i) du_i$$

where  $f(u_i)$  is the marginal distribution of  $u_i$ , and  $\mathbf{x}_i^0$  and  $\mathbf{w}_i^0$  are given covariate values.

## Predictions using the full model

In this section, we discuss the general framework for predictions made after ERMs with multiple auxiliary equations and conditioned on both the covariates and the instruments. The predictions consider the total effect of all the covariates and instruments on the outcome. See [Counterfactual predictions and inferences](#) for a discussion of predictions that may not involve all the covariates and instruments.

Suppose that we have  $H$  auxiliary equations with endogenous outcomes  $y_{1i}, \dots, y_{Hi}$ . We will treat the main outcome  $y_i$  as stage  $J = H + 1$ , so  $y_{Ji} = y_i$ . The ERMs that we fit with `eintreg`, `eoprobit`, `eprobit`, and `eregress` are triangular, so we can order the equations such that the first depends only on exogenous covariates and instruments—say,  $\mathbf{w}_{1i} = \mathbf{z}_i$ —and for  $j = 2, \dots, J$ , equation  $j$  depends only on the exogenous covariates and instruments  $\mathbf{z}_i$  and the endogenous covariates from equation  $h = j - 1$  and below  $y_{1i}, \dots, y_{hi}$ . These are stored together in  $\mathbf{w}_{ji}$ .

When we predict conditional probabilities for binary and ordinal outcomes, we condition on all the endogenous and exogenous covariates and instruments that affect  $y_{ji}$ . Conditional probabilities are calculated as the ratio of the joint density over the marginal density of the conditioning covariates. For binary or ordinal outcome  $y_{ji}$ , we have

$$\Pr(y_{ji} = Y | y_{1i}, \dots, y_{(j-1)i}, \mathbf{z}_i) = \frac{f(Y, y_{1i}, \dots, y_{(j-1)i} | \mathbf{z}_i)}{f(y_{1i}, \dots, y_{(j-1)i} | \mathbf{z}_i)}$$

where the densities can be computed as described in [\[ERM\] eprobit](#).

Now, suppose instead that  $y_{ji}$  is continuous. We can predict the probability that  $y_{ji}$  lies in the range  $(l_{ji}, u_{ji})$ :

$$\begin{aligned} \Pr(l_{ji}, u_{ji}) &= \Pr(l_{ji} < y_{ji} < u_{ji} | y_{1i}, \dots, y_{(j-1)i}, \mathbf{z}_i) \\ &= \int_{(l_{ji}, u_{ji}) \times \mathbf{V}_{(j-1)i}^*} \phi_j(v_{1i}, \dots, v_{ji}, \boldsymbol{\Sigma}_j) dv_{ji} d\mathbf{v}_{(j-1)i}^* \end{aligned}$$

This integral can be evaluated using the methods discussed in [Likelihood for multiequation models](#) in [\[ERM\] eprobit](#).

The conditional mean of continuous outcome  $y_{ji}$  is

$$E(y_{ji} | \mathbf{w}_{ji}) = \mathbf{w}_{ji} \boldsymbol{\beta}_j + E(v_{ji} | \mathbf{w}_{ji})$$

where  $\mathbf{w}_{ji}$  contains the endogenous covariates  $y_{1i}, \dots, y_{(j-1)i}$  and exogenous covariates  $\mathbf{z}_i$  that affect  $y_{ji}$ .

By conditioning on the binary and ordinal endogenous covariates  $y_{1i}, \dots, y_{(j-1)i}$ , the errors  $v_{hi}, \dots, v_{ji}$  become truncated normal. Together with  $v_{ji}$ , they have a truncated multivariate distribution. So the mean of the continuous endogenous covariate is calculated using the moment formulas for the truncated multivariate normal. The first and second moments of the doubly truncated multivariate normal were derived in [Manjunath and Wilhelm \(2012\)](#). [Tallis \(1961\)](#) derived the first and second moments of the multivariate normal with one-sided truncation.

A key result in [Manjunath and Wilhelm \(2012\)](#) is that

$$\int_{l_1}^{u_1} \dots \int_{l_d}^{u_d} \epsilon_f \phi_d(\epsilon, \Sigma) d\epsilon_1 \dots d\epsilon_d = \sum_{k=1}^d \sigma_{fk} \{F_k(l_k) - F_k(u_k)\} \quad (1)$$

where the functions  $F_k(\cdot)$  are defined as

$$F_k(e) = \int_{l_1}^{u_1} \dots \int_{l_{k-1}}^{u_{k-1}} \int_{l_{k+1}}^{u_{k+1}} \phi_d(e_1, \dots, e_{k-1}, e, e_{k+1}, \dots, e_k, \Sigma) de_1 \dots de_{k-1} de_{k+1} \dots de_d$$

The  $F_k(\cdot)$  functions can be computed like the joint density in [Likelihood for multiequation models](#) in [\[ERM\] eprobit](#). So we have

$$E(v_{ji} | \mathbf{w}_{ji}) = \frac{\sum_{k=j}^J \sigma_{jk} \{F_k(l_{ki}) - F_k(u_{ki})\}}{\Phi_J^*(\mathbf{l}_i, \mathbf{u}_i, \Sigma_j)}$$

where  $l_{ji} = -\infty$  and  $u_{ji} = \infty$ .

If there are continuous endogenous regressors in  $y_{1i}, \dots, y_{ji}$ , we condition on them in calculating (1). As in the calculation of the joint density in [Likelihood for multiequation models](#) in [\[ERM\] eprobit](#), we multiply by the marginal density and adjust the cutpoints and variance.

The constrained mean of continuous outcome  $y_{ji}$ , the mean of  $y_{ji}$  when  $y_{ji}$  falls between  $l_{ji}$  and  $u_{ji}$ , is

$$\begin{aligned} E(l_{ji}, u_{ji}) &= E(y_{ji} | \mathbf{w}_{ji}, l_{ji} < y_{ji} < u_{ji}) \\ &= \mathbf{w}_{ji} \beta_j + E(v_{ji} | \mathbf{w}_{ji}, l_{ji} - \mathbf{w}_{ji} \beta_j < \epsilon_{ji} < u_{ji} - \mathbf{w}_{ji} \beta_j) \end{aligned}$$

We use the same method as for the unconstrained mean, with cutpoints  $l_{ji} - \mathbf{w}_{ji} \beta_j$  and  $u_{ji} - \mathbf{w}_{ji} \beta_j$  instead of  $-\infty$  and  $\infty$ .

Finally, the expected value of continuous  $y_{ji}$  with censoring at  $l_{ji}$  and  $u_{ji}$  is

$$\begin{aligned} E(y_{ji}^* | \mathbf{w}_{ji}) &= l_{ji} \mathbf{1}(\mathbf{w}_{ji} \beta_j + \epsilon_{ji} < l_{ji}) + u_{ji} \mathbf{1}(\mathbf{w}_{ji} \beta_j + \epsilon_{ji} > u_{ji}) \\ &\quad + (\mathbf{w}_{ji} \beta_j + \epsilon_{ji}) \mathbf{1}(l_{ji} \leq \mathbf{w}_{ji} \beta_j + \epsilon_{ji} \leq u_{ji}) \end{aligned}$$

where  $y_{ji}^* = \max\{l_{ji}, \min(y_{ij}, u_{ij})\}$ . This can be calculated using predictions we have already discussed:

$$E(y_{ji}^* | \mathbf{w}_{ji}) = \Pr(-\infty, l_{ji}) l_{ji} + \Pr(l_{ji}, u_{ji}) E(l_{ji}, u_{ji}) + \Pr(u_{ji}, \infty) u_{ji}$$

All the predictions above can be made after estimation by using `predict`. By also specifying either the `pr` or the `pr(lji, uji)` option in `predict`, we can obtain conditional probabilities for a binary or ordinal outcome or the conditional probability that a continuous outcome lies in the specified range  $(l_{ji}, u_{ji})$ .

By also specifying the `mean` option, we obtain the conditional mean of a continuous endogenous covariate. The `e( $l_{ji}, u_{ji}$ )` option is used to obtain the constrained mean, and `ystar( $l_{ji}, u_{ji}$ )` is used to obtain the expected value with censoring.

Prediction of treatment effects and potential-outcome means in models with endogenous covariates use the above formulas for the conditional mean and probabilities applied to the potential outcomes  $y_{1i}, \dots, y_{Ti}$  rather than the observed  $y_i$ . Methods and formulas for other predictions are given in the *Methods and formulas* sections of [ERM] **eoprobit**, [ERM] **eintreg**, and [ERM] **eregress**.

## References

- Blundell, R. W., and J. L. Powell. 2003. Endogeneity in nonparametric and semiparametric regression models. In *Advances in Economics and Econometrics: Theory and Applications, Eighth World Congress*, ed. M. Dewatripont, L. P. Hansen, and S. J. Turnovsky, vol. 2, 312–357. Cambridge: Cambridge University Press.
- . 2004. Endogeneity in semiparametric binary response models. *Review of Economic Studies* 71: 655–679.
- Manjunath, B. G., and S. Wilhelm. 2012. Moments calculation for the doubly truncated multivariate normal density. <https://arxiv.org/pdf/1206.5387.pdf>.
- Tallis, G. M. 1961. The moment generating function of the truncated multi-normal distribution. *Journal of the Royal Statistical Society, Series B* 23: 223–229.
- Wooldridge, J. M. 2005. Unobserved heterogeneity and estimation of average partial effects. In *Identification and Inference for Econometric Models: Essays in Honor of Thomas Rothenberg*, ed. D. W. K. Andrews and J. H. Stock, 27–55. New York: Cambridge University Press.
- . 2014. Quasi-maximum likelihood estimation and testing for nonlinear models with endogenous explanatory variables. *Journal of Econometrics* 182: 226–234.

## Also see

- [ERM] **eprbit** — Extended probit regression
- [ERM] **eprbit predict** — predict after eprbit
- [ERM] **predict treatment** — predict for treatment statistics
- [ERM] **predict advanced** — predict’s advanced features
- [U] **20 Estimation and postestimation commands**



Description	Syntax
Options for statistics	Options for how results are calculated
Remarks and examples	Methods and formulas
Also see	

Description

In this entry, we show how to create new variables containing observation-by-observation predictions after fitting a model with `eprobit`.

Syntax

You previously fit the model

```
eprobit y x1 ..., ...
```

The equation specified immediately after the `eprobit` command is called the main equation. It is

$$\Pr(y_i) = \Pr(\beta_0 + \beta_1x1_i + \cdots + e_i.y > 0)$$

`predict` calculates predictions for  $\Pr(y)$  in the main equation. The other equations in the model are called auxiliary equations or complications.

The syntax of `predict` is

```
predict [type] newvar [if] [in] [, stdstatistics howcalculated]
```

<i>stdstatistics</i>	Description
<code>pr</code>	probability of positive outcome; the default
<code>xb</code>	linear prediction excluding all complications

<i>howcalculated</i>	Description
default	not fixed; base values from data
<code>fix(<i>endogvars</i>)</code>	fix specified endogenous covariates
<code>base(<i>valspecs</i>)</code>	specify base values of any variables
<code>target(<i>valspecs</i>)</code>	more convenient way to specify <code>fix()</code> and <code>base()</code>

Note: The `fix()` and `base()` options affect results only in models with endogenous variables in the main equation. The `target()` option is sometimes a more convenient way to specify the `fix()` and `base()` options.

*endogvars* are names of one or more endogenous variables appearing in the main equation.

*valspecs* specify the values for variables at which predictions are to be evaluated. Each *valspec* is of the form

*varname* = #

*varname* = (*exp*)

*varname* = *othervarname*

For instance, `base(valspecs)` could be `base(w1=0)` or `base(w1=0 w2=1)`.

Notes:

- (1) `predict` can also calculate treatment-effect statistics. See [\[ERM\] predict treatment](#).
- (2) `predict` can also make predictions for the other equations in addition to the main-equation predictions discussed here. See [\[ERM\] predict advanced](#).

## Options for statistics

`pr` calculates the predicted probability of a positive outcome. In each observation, the prediction is the probability conditioned on the covariates. Results depend on how complications are handled, which is determined by the *howcalculated* options.

`xb` specifies that the linear prediction be calculated ignoring all complications.

## Options for how results are calculated

By default, predictions are calculated taking into account all complications. This is discussed in [Remarks and examples](#) of [\[ERM\] eregress predict](#).

`fix(varname ...)` specifies a list of endogenous variables from the main equation to be treated as if they were exogenous. This was discussed in [\[ERM\] intro 3](#) and is discussed further in [Remarks and examples](#) of [\[ERM\] eregress predict](#).

`base(varname = ...)` specifies a list of variables from any equation and values for them. If `eprobit` were a linear model, we would tell you those values will be used in calculating the expected value of  $e_i.y$ . That thinking will not mislead you but is not formally correct in the case of `eprobit`. Linear or nonlinear, errors from other equations spill over into the main equation because of correlations between errors. The correlations were estimated when the model was fit. The amount of spillover depends on those correlations and the values of the errors. This issue was discussed in [\[ERM\] intro 3](#) and is discussed further in [Remarks and examples](#) of [\[ERM\] eregress predict](#).

`target(varname = ...)` is sometimes a more convenient way to specify the `fix()` and `base()` options. You specify a list of variables from the main equation and values for them. Those values override the values of the variables calculating  $\beta_0 + \beta_1 x_1 + \dots$ . Use of `target()` is discussed in [Remarks and examples](#) of [\[ERM\] eregress predict](#).

## Remarks and examples

Remarks are presented under the following headings:

*Using predict after eprobit*  
*How to think about nonlinear models*

## Using predict after eprobit

Predictions after fitting models with `eprobit` are handled the same as they are after fitting models with `eregress`. The issues are the same. See [\[ERM\] eregress predict](#).

## How to think about nonlinear models

Probit is a nonlinear model, and yet we just said that predictions after fitting models with `eprobit` are handled the same as they are after fitting models with `eregress`. That statement is partly true, not misleading, but false in its details.

The regression-base discussion that we routed you to is framed in terms of expected values. In the nonlinear models, it needs to be framed in terms of distributional assumptions about the errors. For instance, `predict` after `eprobit` does not predict the expected value (mean) of  $e_i.y$ . It calculates the probability that  $e_i.y$  exceeds  $-x_i\beta$ . These details matter hugely in implementation but can be glossed over for understanding the issues. For a full treatment of the issues, see [Methods and formulas](#) in [\[ERM\] eprobit](#).

## Methods and formulas

See [Methods and formulas](#) in [\[ERM\] eprobit postestimation](#).

## Also see

[\[ERM\] eprobit postestimation](#) — Postestimation tools for eprobit

[\[ERM\] eprobit](#) — Extended probit regression

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

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

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

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

## Description

**eregress** fits a linear regression model that accommodates any combination of endogenous covariates, nonrandom treatment assignment, and endogenous sample selection. Continuous, binary, and ordinal endogenous covariates are allowed. Treatment assignment may be endogenous or exogenous. A probit or tobit model may be used to account for endogenous sample selection.

## Quick start

Regression of  $y$  on  $x$  with continuous endogenous covariate  $y2$  modeled by  $x$  and  $z$

```
eregress y x, endogenous(y2 = x z)
```

As above, but adding continuous endogenous covariate  $y3$  modeled by  $x$  and  $z2$

```
eregress y x, endogenous(y2 = x z) endogenous(y3 = x z2)
```

Regression of  $y$  on  $x$  with binary endogenous covariate  $d$  modeled by  $x$  and  $z$

```
eregress y x, endogenous(d = x z, probit)
```

Regression of  $y$  on  $x$  with endogenous treatment recorded in `trtvar` and modeled by  $x$  and  $z$

```
eregress y x, entreat(trtvar = x z)
```

Regression of  $y$  on  $x$  with exogenous treatment recorded in `trtvar`

```
eregress y x, extreat(trtvar)
```

Regression of  $y$  on  $x$  with endogenous sample-selection indicator `selvar` modeled by  $x$  and  $z$

```
eregress y x, select(selvar = x z)
```

As above, but adding endogenous covariate  $y2$  modeled by  $x$  and  $z2$

```
eregress y x, select(selvar = x z) endogenous(y2 = x z2)
```

As above, but adding endogenous treatment recorded in `trtvar` and modeled by  $x$  and  $z3$

```
eregress y x, select(selvar = x z) endogenous(y2 = x z2) ///  
entreat(trtvar = x z3)
```

## Menu

Statistics > Endogenous covariates > Models adding selection and treatment > Linear regression

## Syntax

*Basic linear regression with endogenous covariates*

```
eregress depvar [indepvars] ,  
    endogenous(depvarsen = varlisten) [options]
```

*Basic linear regression with endogenous treatment assignment*

```
eregress depvar [indepvars] ,  
    entreat(depvartr [= varlisttr]) [options]
```

*Basic linear regression with exogenous treatment assignment*

```
eregress depvar [indepvars] ,  
    extreat(tvar) [options]
```

*Basic linear regression with sample selection*

```
eregress depvar [indepvars] ,  
    select(depvars = varlists) [options]
```

*Basic linear regression with tobit sample selection*

```
eregress depvar [indepvars] ,  
    tobitselect(depvars = varlists) [options]
```

*Linear regression combining endogenous covariates, treatment, and selection*

```
eregress depvar [indepvars] [if] [in] [weight] [, extensions options]
```

<i>extensions</i>	Description
Model	
<u>endogenous</u> ( <i>enspec</i> )	model for endogenous covariates; may be repeated
<u>entreat</u> ( <i>entrspec</i> )	model for endogenous treatment assignment
<u>extreat</u> ( <i>extrspec</i> )	exogenous treatment
<u>select</u> ( <i>selspec</i> )	probit model for selection
<u>tobitselect</u> ( <i>tselspec</i> )	tobit model for selection
<hr/>	
<i>options</i>	Description
Model	
<u>noconstant</u>	suppress constant term
<u>offset</u> ( <i>varname<sub>o</sub></i> )	include <i>varname<sub>o</sub></i> in model with coefficient constrained to 1
<u>constraints</u> ( <i>numlist</i> )	apply specified linear constraints
<u>collinear</u>	keep collinear variables
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 <code>level(95)</code>
<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> (#)	set the number of integration (quadrature) points for integration over four or more dimensions; default is <code>intpoints(128)</code>
<u>triintpoints</u> (#)	set the number of integration (quadrature) points for integration over three dimensions; default is <code>triintpoints(10)</code>
Maximization	
<u>maximize_options</u>	control the maximization process; seldom used
<u>coeflegend</u>	display legend instead of statistics

*enspec* is `depvarsen = varlisten [ , enopts ]`

where *depvars<sub>en</sub>* is a list of endogenous covariates. Each variable in *depvars<sub>en</sub>* specifies an endogenous covariate model using the common *varlist<sub>en</sub>* and options.

*entrspec* is `depvartr [= varlisttr] [ , tropts ]`

where *depvar<sub>tr</sub>* is a variable indicating treatment assignment. *varlist<sub>tr</sub>* is a list of covariates predicting treatment assignment.

*extrspec* is `tvar [ , nomain nointeract ]`

where *tvar* is a variable indicating treatment assignment.

*selspec* is `depvars = varlists [ , noconstant offset(varnameo) ]`

where *depvar<sub>s</sub>* is a variable indicating selection status. *depvar<sub>s</sub>* must be coded as 0, indicating that the observation was not selected, or 1, indicating that the observation was selected. *varlist<sub>s</sub>* is a list of covariates predicting selection.

*tselspec* is `depvars = varlists [ , tselopts ]`

where *depvar<sub>s</sub>* is a continuous variable. *varlist<sub>s</sub>* is a list of covariates predicting *depvar<sub>s</sub>*. The censoring status of *depvar<sub>s</sub>* indicates selection, where a censored *depvar<sub>s</sub>* indicates that the observation was not selected and a noncensored *depvar<sub>s</sub>* indicates that the observation was selected.

<i>enopts</i>	Description
Model	
<u>probit</u>	treat endogenous covariate as binary
<u>oprobit</u>	treat endogenous covariate as ordinal
<u>nomain</u>	do not add endogenous covariate to main equation
<u>noconstant</u>	suppress constant term

<i>tropts</i>	Description
Model	
<u>nomain</u>	do not add treatment indicator to main equation
<u>nointeract</u>	do not interact treatment with covariates in main equation
<u>noconstant</u>	suppress constant term
<u>offset</u> (varname <sub>o</sub> )	include varname <sub>o</sub> in model with coefficient constrained to 1

<i>tselopts</i>	Description
Model	
<u>ll</u> (varname   #)	left-censoring variable or limit
<u>ul</u> (varname   #)	right-censoring variable or limit
<u>main</u>	add censored selection variable to main equation
<u>noconstant</u>	suppress constant term
<u>offset</u> (varname <sub>o</sub> )	include varname <sub>o</sub> in model with coefficient constrained to 1

*indepvars*, *varlist<sub>en</sub>*, *varlist<sub>tr</sub>*, and *varlist<sub>s</sub>* may contain factor variables; see [U] 11.4.3 Factor variables.

*depvar*, *indepvars*, *depvars<sub>en</sub>*, *varlist<sub>en</sub>*, *depvar<sub>tr</sub>*, *varlist<sub>tr</sub>*, *tvar*, *depvar<sub>s</sub>*, and *varlist<sub>s</sub>* may contain time-series operators; see [U] 11.4.4 Time-series varlists.

*bootstrap*, *by*, *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*.

*vce()* and weights are not allowed with the *svy* prefix; see [SVY] *svy*.

*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

`endogenous(enspec)`, `entreat(entrspec)`, `extreat(extrspec)`, `select(selspec)`, `tobitselect(tselspec)`; see [\[ERM\] erm options](#).  
`noconstant`, `offset(varnameo)`, `constraints(numlist)`, `collinear`; see [\[R\] estimation options](#).

### SE/Robust

`vce(vcetype)`; see [\[ERM\] erm options](#).

### Reporting

`level(#)`, `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(#)`, `triintpoints(#)`; see [\[ERM\] erm 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](#).

Setting the optimization type to `technique(bhhh)` resets the default `vcetype` to `vce(opg)`.

The following option is available with `eregress` but is not shown in the dialog box:

`coeflegend`; see [\[R\] estimation options](#).

## Remarks and examples

`eregress` fits models that we refer to as “extended linear regression models”. We use this term to mean linear regression models that accommodate endogenous covariates, nonrandom treatment assignment, and endogenous sample selection. `eregress` can account for these complications whether they arise individually or in combination.

In this entry, you will find information on the `eregress` command syntax. You can see [Methods and formulas](#) for a full description of the models that can be fit with `eregress` and details about how those models are fit.

More information on extended linear regression models is found in the separate introductions and example entries. We recommend reading those entries to learn how to use `eregress`. Below, we provide a guide to help you locate the ones that will be helpful to you.



For an introduction to **eregress** and the other extended regression commands (**eintreg**, **eprobit**, and **eoprobit**), see [ERM] [intro 1](#)–[ERM] [intro 8](#).

[ERM] [intro 1](#) introduces the ERM commands, the problems they address, and their syntax.

[ERM] [intro 2](#) provides background on the four types of models—linear regression, interval regression, probit regression, and ordered probit regression—that can be fit using ERM commands.

[ERM] [intro 3](#) considers the problem of endogenous covariates and how to solve it using ERM commands.

[ERM] [intro 4](#) gives an overview of endogenous sample selection and using ERM commands to account for it.

[ERM] [intro 5](#) covers nonrandom treatment assignment and how to account for it using **eregress** or any of the other ERM commands.

[ERM] [intro 6](#) discusses interpretation of results. You can interpret coefficients from **eregress** in the usual way, but this introduction goes beyond the interpretation of coefficients. We demonstrate how to find answers to interesting questions by using **margins**. If your model includes an endogenous covariate or an endogenous treatment, the use of **margins** differs from its use after other estimation commands, so we strongly recommend reading this intro if you are fitting these types of models.

[ERM] [intro 7](#) will be helpful if you are familiar with **heckman**, **ivregress**, **etregress**, and other commands that address endogenous covariates, sample selection, or nonrandom treatment assignment. This introduction is a Rosetta stone that maps the syntax of those commands to the syntax of **eregress**.

[ERM] [intro 8](#) walks you through an example that gives insight into the concepts of endogenous covariates, treatment assignment, and sample selection while fitting models with **eregress** that address these complications. This intro also demonstrates how to interpret results by using **margins** and **estat teffects**.

Additional examples are presented in [ERM] [example 1a](#)–[ERM] [example 6b](#). For examples using **eregress**, see

[ERM] <a href="#">example 1a</a>	Linear regression with continuous endogenous covariate
[ERM] <a href="#">example 2a</a>	Linear regression with binary endogenous covariate
[ERM] <a href="#">example 2b</a>	Linear regression with exogenous treatment
[ERM] <a href="#">example 2c</a>	Linear regression with endogenous treatment

See [Examples](#) in [ERM] [intro](#) for an overview of all the examples. These examples demonstrate all four extended regression commands, and all may be interesting because they handle complications in the same way.

You can also find in literature discussion and examples of many models that **eregress** can fit. For example, **eregress** can fit the linear regression model with endogenous sample selection (Heckman 1976), the linear regression model with an endogenous treatment (Heckman 1978; Maddala 1983), and the linear regression model with a tobit selection equation (Amemiya 1985; Wooldridge 2010, sec. 19.7). The linear regression model with endogenous regressors and endogenous sample selection discussed in Wooldridge (2010, sec 19.6) is also supported, along with the tobit selection regression with endogenous regressors discussed in Wooldridge (2010, sec 19.7). Roodman (2011) investigated linear regression models with endogenous covariates and endogenous sample selection, and demonstrated how multiple observational data complications could be addressed with a triangular model structure. His work has been used to model processes like the effect of aphid infestations and virus outbreaks on crop yields (Elbakidze, Lu, and Eigenbrode 2011) and the effect of calorie intake per day on food security in poor neighborhoods (Maitra and Rao 2014).

## Stored results

`eregress` stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(N_selected)</code>	number of uncensored observations
<code>e(N_nonselected)</code>	number of censored observations
<code>e(k)</code>	number of parameters
<code>e(k_cat#)</code>	number of categories for the <i>#th depvar</i> , ordinal
<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_dv)</code>	number of dependent variables
<code>e(k_aux)</code>	number of auxiliary parameters
<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>	$\chi^2$
<code>e(p)</code>	significance
<code>e(n_quad)</code>	number of integration points for multivariate normal
<code>e(n_quad3)</code>	number of integration points for trivariate normal
<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>	<code>eregress</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	names of dependent variables
<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(offset#)</code>	offset for the <i>#th depvar</i>
<code>e(chi2type)</code>	Wald; type of model $\chi^2$ test
<code>e(vce)</code>	<i>vcetype</i> 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>	<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(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(cat#)</code>	categories for the <i>#th depvar</i> , ordinal
<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
------------------------	-------------------------

## Methods and formulas

The methods and formulas presented here are for the linear-outcome model. They are a special case of the methods and formulas derived by [Roodman \(2011\)](#).

Methods and formulas are presented under the following headings:

- [Introduction](#)
- [Endogenous covariates](#)
  - [Continuous endogenous covariates](#)
  - [Binary and ordinal endogenous covariates](#)
- [Treatment](#)
- [Endogenous sample selection](#)
  - [Probit endogenous sample selection](#)
  - [Tobit endogenous sample selection](#)
- [Combinations of features](#)
- [Confidence intervals](#)

## Introduction

A linear regression of outcome  $y_i$  on covariates  $\mathbf{x}_i$  may be written as

$$y_i = \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i$$

where the error  $\epsilon_i$  is normal with mean 0 and variance  $\sigma^2$ . The log likelihood is

$$\ln L = \sum_{i=1}^N w_i \ln \phi(y_i - \mathbf{x}_i\boldsymbol{\beta}, \sigma^2)$$

The conditional mean of  $y_i$  is

$$E(y_i|\mathbf{x}_i) = \mathbf{x}_i\boldsymbol{\beta}$$

If you are willing to take our word for some derivations and notation, the following is complete. Longer explanations and derivations for some terms and functions are provided in the [Methods and formulas](#) of [ERM] **eprobit**. For example, we need the two-sided probability function  $\Phi_d^*$  that is discussed in [Introduction](#) in [ERM] **eprobit**.

If you are interested in all the details, we suggest you read [Methods and formulas](#) of [ERM] **eprobit** in its entirety, before reading this section. Here, we mainly show how the complications that arise in ERMs are handled in a linear regression framework.

## Endogenous covariates

### Continuous endogenous covariates

A linear regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and  $C$  continuous endogenous covariates  $\mathbf{w}_{ci}$  has the form

$$\begin{aligned} y_i &= \mathbf{x}_i\boldsymbol{\beta} + \mathbf{w}_{ci}\boldsymbol{\beta}_c + \epsilon_i \\ \mathbf{w}_{ci} &= \mathbf{z}_{ci}\mathbf{A}_c + \boldsymbol{\epsilon}_{ci} \end{aligned}$$

The vector  $\mathbf{z}_{ci}$  contains variables from  $\mathbf{x}_i$  and other covariates that affect  $\mathbf{w}_{ci}$ . For the model to be identified,  $\mathbf{z}_{ci}$  must contain one extra exogenous covariate not in  $\mathbf{x}_i$  for each of the endogenous regressors in  $\mathbf{w}_{ci}$ . The unobserved errors  $\epsilon_i$  and  $\boldsymbol{\epsilon}_{ci}$  are multivariate normal with mean 0 and covariance

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma^2 & \sigma'_{1c} \\ \sigma_{1c} & \boldsymbol{\Sigma}_c \end{bmatrix}$$

The log likelihood is

$$\ln L = \sum_{i=1}^N w_i \ln \phi_{C+1}(\mathbf{r}_i, \Sigma)$$

where

$$\mathbf{r}_i = [y_i - \mathbf{x}_i \quad \mathbf{w}_{ci} - \mathbf{z}_{ci} \mathbf{A}_c]$$

The conditional mean of  $y_i$  is

$$E(y_i | \mathbf{x}_i, \mathbf{w}_{ci}, \mathbf{z}_{ci}) = \mathbf{x}_i \beta + \mathbf{w}_{ci} \beta_c + \sigma'_{1c} \Sigma_c^{-1} (\mathbf{w}_{ci} - \mathbf{z}_{ci} \mathbf{A}_c)'$$

## Binary and ordinal endogenous covariates

Here, we begin by formulating the linear regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and  $B$  binary and ordinal endogenous covariates  $\mathbf{w}_{bi} = [w_{b1i}, \dots, w_{bBi}]$ . Indicator (dummy) variables for the levels of each binary and ordinal covariate are used in the model. You can also interact other covariates with the binary and ordinal endogenous covariates, as in treatment-effect models.

The binary and ordinal endogenous covariates  $\mathbf{w}_{bi}$  are formulated as in [Binary and ordinal endogenous covariates](#) in [ERM] **eprobit**. So we have

$$y_i = \mathbf{x}_i \beta + \mathbf{wind}_{b1i} \beta_{b1} + \dots + \mathbf{wind}_{bBi} \beta_{bB} + \epsilon_i$$

The  $\mathbf{wind}_{bji}$  vectors are defined in [Binary and ordinal endogenous covariates](#) in [ERM] **eprobit**. The binary and ordinal endogenous errors  $\epsilon_{b1i}, \dots, \epsilon_{bBi}$  and outcome error  $\epsilon_i$  are multivariate normal with mean 0 and covariance

$$\Sigma = \begin{bmatrix} \Sigma_b & \sigma_{1b} \\ \sigma'_{1b} & \sigma^2 \end{bmatrix}$$

From here, we discuss the model with ordinal endogenous covariates. The results for binary endogenous covariates are similar.

Using results from [Likelihood for multiequation models](#) in [ERM] **eprobit**, the joint density of  $y_i$  and  $\mathbf{w}_{bi}$  can be written using the conditional density of  $\epsilon_{b1i}, \dots, \epsilon_{bBi}$  on  $\epsilon_i$ .

Define

$$r_i = y_i - (\mathbf{x}_i \beta + \mathbf{wind}_{b1i} \beta_{b1} + \dots + \mathbf{wind}_{bBi} \beta_{bB})$$

Let

$$\boldsymbol{\mu}_{b|1,i} = \frac{\sigma'_{1b}}{\sigma^2} r_i = [e_{b1i} \dots e_{bBi}]$$

$$\Sigma_{b|1} = \Sigma_b - \frac{\sigma_{1b} \sigma'_{1b}}{\sigma^2}$$

For  $j = 1, \dots, B$  and  $h = 0, \dots, B_j$ , let

$$c_{bjih} = \begin{cases} -\infty & h = 0 \\ \kappa_{bjh} - \mathbf{z}_{bji} \boldsymbol{\alpha}_{bj} - e_{bji} & h = 1, \dots, B_j - 1 \\ \infty & h = B_j \end{cases}$$

So, for  $j = 1, \dots, B$ , the probability for  $w_{bji}$  has lower limit

$$l_{bji} = c_{bji(h-1)} \quad \text{if } w_{bji} = v_{bjh}$$

and upper limit

$$u_{bji} = c_{bjih} \quad \text{if } w_{bji} = v_{bjh}$$

Let

$$\mathbf{l}_i = [l_{b1i} \quad \dots \quad l_{bBi}]$$

$$\mathbf{u}_i = [u_{b1i} \quad \dots \quad u_{bBi}]$$

So, the log likelihood for this model is

$$\ln L = \sum_{i=1}^N w_i \ln \{ \Phi_B^*(\mathbf{l}_i, \mathbf{u}_i, \Sigma_{b|1}) \phi(r_i, \sigma^2) \}$$

The expected value of  $y_i$  on  $\mathbf{w}_{bi}$  can be calculated using the techniques discussed in [Predictions using the full model](#) in [ERM] **eprobit postestimation**.

## Treatment

In the potential-outcomes framework, the treatment  $t_i$  is a discrete variable taking  $T$  values, indexing the  $T$  potential outcomes of the outcome  $y_i$ :  $y_{1i}, \dots, y_{Ti}$ .

When we observe treatment  $t_i$  with levels  $v_1, \dots, v_T$ , we have

$$y_i = \sum_{j=1}^T 1(t_i = v_j) y_{ji}$$

So for each observation, we only observe the potential outcome associated with that observation's treatment value.

For exogenous treatments, our approach is equivalent to the regression adjustment treatment-effect estimation method. See [TE] **teffects intro advanced**. We do not model the treatment assignment process. The formulas for the treatment effects and potential-outcome means (POMs) are equivalent to what we provide here for endogenous treatments. The treatment effect on the treated for  $\mathbf{x}_i$  for an exogenous treatment is equivalent to what we provide here for the endogenous treatment when the correlation parameter between the outcome and treatment errors is set to 0. The average treatment effects (ATES) and POMs for exogenous treatments are estimated as predictive margins in an analogous manner to what we describe here for endogenous treatments.

From here, we assume an endogenous treatment  $t_i$ . As in [Treatment](#) in [ERM] **eprobit**, we model the treatment assignment process with a probit or ordered probit model, and we call the treatment assignment error  $\epsilon_{ti}$ . A linear regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and endogenous treatment  $t_i$  taking values  $v_1, \dots, v_T$  has the form

$$y_{1i} = \mathbf{x}_i \beta_1 + \epsilon_{1i}$$

$$\vdots$$

$$y_{Ti} = \mathbf{x}_i \beta_T + \epsilon_{Ti}$$

$$y_i = \sum_{j=1}^T 1(t_i = v_j) y_{ji}$$

For  $j = 1, \dots, T$ ,  $\epsilon_{ji}$  and  $\epsilon_{ti}$  are bivariate normal with mean 0 and covariance

$$\Sigma = \begin{bmatrix} \sigma^2 & \sigma \rho_{1t} \\ \sigma \rho_{1t} & 1 \end{bmatrix}$$

The treatment is exogenous if  $\rho_{1t} = 0$ . Note that we did not specify the structure of the correlations between the potential-outcome errors. We do not need information about these correlations to estimate POMs and treatment effects because all covariates and the outcome are observed in observations from each group.

From here, we discuss a model with an ordinal endogenous treatment. The results for binary treatment models are similar.

As in *Binary and ordinal endogenous covariates*, using the results from *Likelihood for multiequation models* in [ERM] **eprobit**, the joint density of  $y_i$  and  $t_i$  can be written using the conditional density of the treatment error  $\epsilon_{ti}$  on the outcome errors  $\epsilon_{i1}, \dots, \epsilon_{Ti}$ .

Define

$$r_i = y_i - \mathbf{x}_i \boldsymbol{\beta}_j \quad \text{if} \quad t_i = v_j$$

The log likelihood for the model is

$$\ln L = \sum_{i=1}^N w_i \ln \left\{ \Phi_1^* \left( l_{ti} - \frac{\rho_{1t}}{\sigma} r_i, u_{ti} - \frac{\rho_{1t}}{\sigma} r_i, 1 - \rho_{1t}^2 \right) \phi(r_i, \sigma^2) \right\}$$

where  $l_{ti}$  and  $u_{ti}$  are the limits for the treatment probability given in *Treatment* in [ERM] **eprobit**.

The treatment effect  $y_{ji} - y_{1i}$  is the difference in the outcome for individual  $i$  if the individual receives the treatment  $t_i = v_j$  instead of the control  $t_i = v_1$  and what the difference would have been if the individual received the control treatment instead.

The conditional POM for treatment group  $j$  is

$$\text{POM}_j(\mathbf{x}_i) = E(y_{ji} | \mathbf{x}_i) = \mathbf{x}_i \boldsymbol{\beta}_j$$

For treatment group  $j$ , the treatment effect (TE) conditioned on  $\mathbf{x}_i$  is

$$\text{TE}_j(\mathbf{x}_i) = E(y_{ji} - y_{1i} | \mathbf{x}_i) = \text{POM}_j(\mathbf{x}_i) - \text{POM}_1(\mathbf{x}_i)$$

For treatment group  $j$ , the treatment effect on the treated (TET) in group  $h$  for covariates  $\mathbf{x}_i$  is

$$\begin{aligned} \text{TET}_j(\mathbf{x}_i, t_i = v_h) &= E(y_{ji} - y_{1i} | \mathbf{x}_i, t_i = v_h) \\ &= \mathbf{x}_i \boldsymbol{\beta}_j - \mathbf{x}_i \boldsymbol{\beta}_1 \\ &\quad + E(\epsilon_{ji} | \mathbf{x}_i, t_i = v_h) - E(\epsilon_{1i} | \mathbf{x}_i, t_i = v_h) \end{aligned}$$

Remembering that the outcome errors and the treatment error  $\epsilon_{ti}$  are multivariate normal, for  $j = 1, \dots, T$  we can decompose  $\epsilon_{ji}$  such that

$$\epsilon_{ji} = \sigma \rho_{1t} \epsilon_{ti} + \psi_{ji}$$

where  $\psi_{ji}$  has mean 0.

It follows that

$$\text{TET}_j(\mathbf{x}_i, t_i = v_h) = \mathbf{x}_i \boldsymbol{\beta}_j - \mathbf{x}_i \boldsymbol{\beta}_1$$

We can take the expectation of these conditional predictions over the covariates to get population average parameters. The `estat teffects` or `margins` command is used to estimate the expectations as predictive margins once the model is estimated with **eregress**. The POM for treatment group  $j$  is

$$\text{POM}_j = E(y_{ji}) = E\{\text{POM}_j(\mathbf{x}_i)\}$$

The ATE for treatment group  $j$  is

$$\text{ATE}_j = E(y_{ji} - y_{1i}) = E\{\text{TE}_j(\mathbf{x}_i)\}$$

For treatment group  $j$ , the average treatment effect on the treated (ATET) in treatment group  $h$  is

$$\begin{aligned}\text{ATET}_{jh} &= E(y_{ji} - y_{1i} | t_i = v_h) \\ &= E\{\text{TET}_j(\mathbf{x}_i, t_i = v_h) | t_i = v_h\}\end{aligned}$$

The conditional mean of  $y_i$  at treatment level  $v_j$  is

$$E(y_i | \mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_j) = \mathbf{x}_i \boldsymbol{\beta}_j + E(\epsilon_i | \mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_j)$$

In *Predictions using the full model* in [ERM] **eprobbit postestimation**, we discuss how the conditional mean of  $\epsilon_i$  is calculated.

## Endogenous sample selection

### Probit endogenous sample selection

A linear regression for outcome  $y_i$  with selection on  $s_i$  has the form

$$\begin{aligned}y_i &= \mathbf{x}_i \boldsymbol{\beta} + \epsilon_i > 0 \\ s_i &= 1 (\mathbf{z}_{si} \boldsymbol{\alpha}_s + \epsilon_{si} > 0)\end{aligned}$$

where  $\mathbf{x}_i$  are covariates that affect the outcome and  $\mathbf{z}_{si}$  are covariates that affect selection. The outcome  $y_i$  is observed if  $s_i = 1$  and is not observed if  $s_i = 0$ . The unobserved errors  $\epsilon_i$  and  $\epsilon_{si}$  are normal with mean 0 and covariance

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma^2 & \sigma \rho_{1s} \\ \sigma \rho_{1s} & 1 \end{bmatrix}$$

As in the previous section, using the results from *Likelihood for multiequation models* in [ERM] **eprobbit**, the joint density of  $y_i$  and  $s_i$  can be written using the conditional density of the selection error  $\epsilon_{si}$  on the outcome error  $\epsilon_i$ .

For the selection indicator  $s_i$ , we have lower and upper limits

$$l_{si} = \begin{cases} -\infty & s_i = 0 \\ -\mathbf{z}_{si} \boldsymbol{\alpha}_s - \frac{\rho_{1s}}{\sigma} (y_i - \mathbf{x}_i \boldsymbol{\beta}) & s_i = 1 \end{cases} \quad u_{si} = \begin{cases} -\mathbf{z}_{si} \boldsymbol{\alpha}_s & s_i = 0 \\ \infty & s_i = 1 \end{cases}$$

The log likelihood for the model is

$$\ln L = \sum_{i=1}^N w_i \ln \Phi_1^* (l_{si}, u_{si}, 1 - s_i \rho_{1s}^2) + \sum_{i \in S} w_i \ln \phi(y_i - \mathbf{x}_i \boldsymbol{\beta}, \sigma^2)$$

where  $S$  is the set of observations for which  $y_i$  is observed.

The conditional mean of  $y_i$  is

$$E(y_i | \mathbf{x}_i) = \mathbf{x}_i \boldsymbol{\beta}$$

## Tobit endogenous sample selection

Instead of constraining the selection indicator to be binary, tobit endogenous sample selection uses a censored continuous sample-selection indicator. We allow the selection variable to be left-censored or right-censored.

A linear regression model for outcome  $y_i$  with tobit selection on  $s_i$  has the form

$$y_i = \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i > 0$$

We observe the selection indicator  $s_i$ , which indicates the censoring status of the latent selection variable  $s_i^*$ ,

$$s_i^* = \mathbf{z}_{si}\boldsymbol{\alpha}_s + \epsilon_{si}$$

$$s_i = \begin{cases} l_i & s_i^* \leq l_i \\ s_i^* & l_i < s_i^* < u_i \\ u_i & s_i^* \geq u_i \end{cases}$$

where  $\mathbf{z}_{si}$  are covariates that affect selection, and  $l_i$  and  $u_i$  are fixed lower and upper limits.

The outcome  $y_i$  is observed when  $s_i^*$  is not censored ( $l_i < s_i^* < u_i$ ). The outcome  $y_i$  is not observed when  $s_i^*$  is left-censored ( $s_i^* \leq l_i$ ) or  $s_i^*$  is right-censored ( $s_i^* \geq u_i$ ). The unobserved errors  $\epsilon_i$  and  $\epsilon_{si}$  are normal with mean 0 and covariance

$$\begin{bmatrix} \sigma^2 & \sigma_{1s} \\ \sigma_{1s} & \sigma_s^2 \end{bmatrix}$$

For the selected observations, we can treat  $s_i$  as a continuous endogenous regressor, as in [Continuous endogenous covariates](#). In fact,  $s_i$  may even be used as a regressor for  $y_i$  in `eregress` (specify `tobitselect(..., main)`). On the nonselected observations, we treat  $s_i$  like the probit sample-selection indicator in [Probit endogenous sample selection](#).

The log likelihood is

$$\begin{aligned} \ln L = & \sum_{i \in S} w_i \ln \phi_2(y_i - \mathbf{x}_i\boldsymbol{\beta}, s_i - \mathbf{z}_{si}\boldsymbol{\alpha}_s, \boldsymbol{\Sigma}) \\ & + \sum_{i \in L} w_i \ln \Phi_1^*(l_i, u_{li}, 1) \\ & + \sum_{i \in U} w_i \ln \Phi_1^*(l_{ui}, u_{ui}, 1) \end{aligned}$$

where  $S$  is the set of observations for which  $y_i$  is observed,  $L$  is the set of observations where  $s_i^*$  is left-censored, and  $U$  is the set of observations where  $s_i^*$  is right-censored. The lower and upper limits for selection— $l_{li}$ ,  $u_{li}$ ,  $l_{ui}$ , and  $u_{ui}$ —are defined in [Tobit endogenous sample selection](#) in [ERM] `eprobit`.

When  $s_i$  is not a covariate in  $\mathbf{x}_i$ , we use the standard conditional mean formula,

$$E(y_i|\mathbf{x}_i) = \mathbf{x}_i\boldsymbol{\beta}$$

Otherwise, we use

$$E(y_i|\mathbf{x}_i, s_i, z_{si}) = \mathbf{x}_i\boldsymbol{\beta} + \frac{\sigma_{1s}}{\sigma_s^2}(s_i - z_{si}\boldsymbol{\alpha}_s)$$



## Combinations of features

Extended linear regression models that involve multiple features can be formulated using the techniques discussed in *Likelihood for multiequation models* in [ERM] **eprobit**. Essentially, the density of the observed endogenous covariates can be written in terms of the unobserved normal errors. The observed endogenous and exogenous covariates determine the range of the errors, and the joint density can be evaluated as multivariate normal probabilities and densities.

## Confidence intervals

The estimated variances will always be nonnegative, and the estimated correlations will always fall in  $(-1, 1)$ . To obtain confidence intervals that accommodate these ranges, we must use transformations.

We use the log transformation to obtain the confidence intervals for variance parameters, and we use the atanh transformation to obtain confidence intervals for correlation parameters. For details, see *Confidence intervals* in [ERM] **eprobit**.

## References

- Amemiya, T. 1985. *Advanced Econometrics*. Cambridge, MA: Harvard University Press.
- Elbakidze, L., L. Lu, and S. Eigenbrode. 2011. Evaluating vector-virus-yield interactions for peas and lentils under climatic variability: A limited dependent variable analysis. *Journal of Agricultural and Resource Economics* 36: 504–520.
- Heckman, J. 1976. The common structure of statistical models of truncation, sample selection and limited dependent variables and a simple estimator for such models. *Annals of Economic and Social Measurement* 5: 475–492.
- . 1978. Dummy endogenous variables in a simultaneous equation system. *Econometrica* 46: 931–959.
- Keshk, O. M. G. 2003. Simultaneous equations models: What are they and how are they estimated. Program in Statistics and Methodology, Department of Political Science, Ohio State University. [https://polisci.osu.edu/sites/polisci.osu.edu/files/Simultaneous\\_Equations.pdf](https://polisci.osu.edu/sites/polisci.osu.edu/files/Simultaneous_Equations.pdf).
- Maddala, G. S. 1983. *Limited-Dependent and Qualitative Variables in Econometrics*. Cambridge: Cambridge University Press.
- Maitra, C., and P. Rao. 2014. An empirical investigation into measurement and determinants of food security in slums of Kolkata. School of Economics Discussion Paper No. 531, School of Economics, University of Queensland. [espace.library.uq.edu.au/view/UQ:352184](https://espace.library.uq.edu.au/view/UQ:352184)
- Roodman, D. 2011. Fitting fully observed recursive mixed-process models with **cmp**. *Stata Journal* 11: 159–206.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

## Also see

- [ERM] **eregress postestimation** — Postestimation tools for **eregress**
- [ERM] **eregress predict** — predict after **eregress**
- [ERM] **estat teffects** — Average treatment effects for extended regression models
- [ERM] **intro 8** — Conceptual introduction via worked example
- [R] **heckman** — Heckman selection model
- [R] **ivregress** — Single-equation instrumental-variables regression
- [R] **regress** — Linear regression
- [SVY] **svy estimation** — Estimation commands for survey data
- [TE] **etregress** — Linear regression with endogenous treatment effects
- [U] **20 Estimation and postestimation commands**

Postestimation commands  
Methods and formulas

predict  
References

margins  
Also see

Remarks and examples

## Postestimation commands

The following postestimation command is of special interest after **eregress**:

Command	Description
<code>estat teffects</code>	treatment effects and potential-outcome means

The following standard postestimation commands are also available after **eregress**:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of estimates
<code>estat ic</code>	Akaike’s and Schwarz’s Bayesian information criteria (AIC and BIC)
<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>forecast</code>	dynamic forecasts and simulations
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
* <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 coefficients
<code>predict</code>	predictions, residuals, influence statistics, and other diagnostic measures
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of estimates
<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

\* `forecast` and `lrtest` are not appropriate with `svy` estimation results.

# predict

Predictions after `eregress` are described in

- [\[ERM\] eregress predict](#) predict after eregress
- [\[ERM\] predict treatment](#) predict for treatment statistics
- [\[ERM\] predict advanced](#) predict's advanced features

[\[ERM\] eregress predict](#) describes the most commonly used predictions. If you fit a model with treatment effects, predictions specifically related to these models are detailed in [\[ERM\] predict treatment](#). [\[ERM\] predict advanced](#) describes less commonly used predictions, such as predictions of outcomes in auxiliary equations.

# margins

## Description for margins

`margins` estimates margins of response for means, probabilities, potential-outcome means, treatment effects, and linear predictions.

## Menu for margins

Statistics > Postestimation

## Syntax for margins

```
margins [marginlist] [ , options ]
margins [marginlist] , predict(statistic ...) [predict(statistic ...) ...] [options]
```

statistic	Description
Main	
<u>m</u> ean	mean; the default
<u>p</u> r	probability for binary or ordinal $y_j$
<u>p</u> omean	potential-outcome mean
<u>t</u> e	treatment effect
<u>t</u> et	treatment effect on the treated
<u>x</u> b	linear prediction
<u>p</u> r( $a,b$ )	$\Pr(a < y_j < b)$ for continuous $y_j$
<u>e</u> ( $a,b$ )	$E(y_j   a < y_j < b)$ for continuous $y_j$
<u>y</u> star( $a,b$ )	$E(y_j^*), y_j^* = \max\{a, \min(y_j, b)\}$ for continuous $y_j$

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.  
For the full syntax, see [\[R\] margins](#).

## Remarks and examples

See [ERM] [intro 6](#) for an overview of using margins and predict after eregress. For examples using margins, predict, and estat teffects, see [Interpreting effects](#) in [ERM] [intro 8](#) and see [ERM] [example 1a](#).

## Methods and formulas

This section contains methods and formulas for counterfactual predictions and inference. Methods and formulas for all other predictions are given in [Methods and formulas](#) of [ERM] [eregress](#). In [Methods and formulas](#) of [ERM] [eregress](#), we discussed how treatment effects are evaluated in extended linear regression models. Here, we discuss the counterfactual framework used to evaluate the effects of other covariates.

In the extended linear regression model for  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and  $\mathbf{w}_i$ , we partition each set of covariates into two groups. The exogenous covariates  $\mathbf{x}_i$  are partitioned into  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^{nc}$ , where we are interested in the effect of changes in  $\mathbf{x}_i^c$ . Similarly, the endogenous covariates  $\mathbf{w}_i$  are partitioned into  $\mathbf{w}_i^c$  and  $\mathbf{w}_i^{nc}$ , where the effect of changes in  $\mathbf{w}_i^c$  are of interest. The superscripts indicate what is a counterfactual value ( $c$ ) and what is not ( $nc$ ).

If  $\mathbf{x}_i^c = \mathbf{a}_0$  and  $\mathbf{w}_i^c = \mathbf{a}_{20}$ , for covariates  $\mathbf{w}_i^{nc}$  and  $\mathbf{x}_i^{nc}$  we would observe the outcome

$$\begin{aligned} y_{0i} &= \beta_{0nc}\mathbf{x}_i^{nc} + \beta_{20nc}\mathbf{w}_i^{nc} + \beta_c\mathbf{a}_0 + \beta_{2c}\mathbf{a}_{20} + \epsilon_{0i} \\ &= \beta_{0nc}\mathbf{x}_i^{nc} + \beta_{20nc}\mathbf{w}_i^{nc} + \beta_{c0} + \epsilon_{0i} \end{aligned}$$

where the unobserved error  $\epsilon_{0i}$  is normal with mean 0. We treat  $\beta_c\mathbf{a}_0 + \beta_{2c}\mathbf{a}_{20} = \beta_{c0}$  as a constant intercept, because it is the same for each value combination of the covariates  $\mathbf{w}_i^{nc}$  and  $\mathbf{x}_i^{nc}$  and the error  $\epsilon_{0i}$ .

Similarly, if  $\mathbf{x}_i^c = \mathbf{a}_1$  and  $\mathbf{w}_i^c = \mathbf{a}_{21}$ , for covariates  $\mathbf{w}_i^{nc}$  and  $\mathbf{x}_i^{nc}$  we would observe the outcome

$$\begin{aligned} y_{1i} &= \beta_{1nc}\mathbf{x}_i^{nc} + \beta_{21nc}\mathbf{w}_i^{nc} + \beta_c\mathbf{a}_1 + \beta_{2c}\mathbf{a}_{21} + \epsilon_{1i} \\ &= \beta_{1nc}\mathbf{x}_i^{nc} + \beta_{21nc}\mathbf{w}_i^{nc} + \beta_{c1} + \epsilon_{1i} \end{aligned}$$

The effect of changing  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$  from  $\mathbf{a}_0$  and  $\mathbf{a}_{20}$  to  $\mathbf{a}_1$  and  $\mathbf{a}_{21}$  on  $y_i$  is the expected difference between  $y_{1i}$  and  $y_{0i}$ .

To obtain this difference, we average the conditional means of  $y_{1i}$  and  $y_{0i}$  as a predictive margin.

For  $j = 0, 1$ , we can predict the counterfactual mean for group  $j$  by using the tools discussed in [Predictions using the full model](#) in [ERM] [eprob](#) [postestimation](#),

$$\text{CM}_j(\mathbf{w}_i^{nc}, \mathbf{x}_i^{nc}, \mathbf{z}_i) = E(y_{ji} | \mathbf{w}_i^{nc}, \mathbf{x}_i^{nc}, \mathbf{x}_i^c = \mathbf{a}_j, \mathbf{z}_i)$$

where  $\mathbf{z}_i$  are instruments necessary for modeling the endogenous regressors  $\mathbf{w}_i^{nc}$ . By the law of iterated expectations, we have

$$E(y_{1i} - y_{0i}) = E\{\text{CM}_1(\mathbf{w}_i^{nc}, \mathbf{x}_i^{nc}, \mathbf{z}_i)\} - E\{\text{CM}_0(\mathbf{w}_i^{nc}, \mathbf{x}_i^{nc}, \mathbf{z}_i)\}$$

So the effect of changing  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$  from  $\mathbf{a}_0$  and  $\mathbf{a}_{20}$  to  $\mathbf{a}_1$  and  $\mathbf{a}_{21}$  can be estimated as a predictive margin on the counterfactual means.

We can use `predict` with the `fix()` and `target()` options to predict the counterfactual probabilities. The `fix()` option is used to indicate the endogenous covariates in  $\mathbf{w}_i^c$ . The `target()` option can be used to set the counterfactual values  $a_j$  and  $a_{2j}$  of  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$ .

When  $\mathbf{w}_i^c$  corresponds to a single ordinal or binary regressor, the difference in counterfactual probabilities corresponds to a treatment effect of  $\mathbf{w}_i^c$ . We can also evaluate the effect of a change in  $\mathbf{w}_i^c$  and  $\mathbf{x}_i^c$ , conditioned on  $\mathbf{w}_i^c$ . This effect is analogous to the treatment effect on the treated discussed in [Methods and formulas](#) of [ERM] **eregress**. We are conditioning the effect on some base value for  $\mathbf{w}_i^c$ ,  $\mathbf{w}_i^c = \mathbf{b}$ .

Now, the counterfactual means are conditioned on  $\mathbf{w}_i^c = \mathbf{b}$ . So for  $j = 0, 1$ , we have

$$\text{CM}_{bj}(\mathbf{w}_i^{nc}, \mathbf{b}, \mathbf{x}_i^{nc}, \mathbf{z}_i) = E(y_{ji} | \mathbf{w}_i^{nc}, \mathbf{w}_i^c = \mathbf{b}, \mathbf{x}_i^{nc}, \mathbf{x}_i^c = \mathbf{a}_j, \mathbf{z}_{bi})$$

where  $\mathbf{z}_{bi}$  are instruments necessary for modeling the endogenous regressors  $\mathbf{w}_i^{nc}$  and  $\mathbf{w}_i^c$ . This counterfactual mean can be evaluated using the tools discussed in [Predictions using the full model](#) in [ERM] **eprobit postestimation**.

By the law of iterated expectations, we have

$$\begin{aligned} E(y_{1i} - y_{0i} | \mathbf{w}_i^c = \mathbf{b}) &= E\{\text{CM}_{b1}(\mathbf{w}_i^{nc}, \mathbf{w}_i^c = \mathbf{b}, \mathbf{x}_i^{nc}, \mathbf{z}_i) | \mathbf{w}_i^c = \mathbf{b}\} - \\ &\quad E\{\text{CM}_{b0}(\mathbf{w}_i^{nc}, \mathbf{w}_i^c = \mathbf{b}, \mathbf{x}_i^{nc}, \mathbf{z}_i) | \mathbf{w}_i^c = \mathbf{b}\} \end{aligned}$$

So the effect of changing  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$  from  $\mathbf{a}_0$  and  $\mathbf{a}_{20}$  to  $\mathbf{a}_1$  and  $\mathbf{a}_{21}$  conditioned on  $\mathbf{w}_i^c = \mathbf{b}$  can be estimated as a predictive margin on the counterfactual means.

The base values  $\mathbf{b}$  for  $\mathbf{w}_i^c$  are specified in the `base()` option. As before, `target()` can be used to specify the counterfactual values for  $\mathbf{x}_i^c$  and  $\mathbf{w}_i^c$ .

When  $\mathbf{x}_i^c = \mathbf{x}_i$  and  $\mathbf{w}_i^c = \mathbf{w}_i$ , the counterfactual mean matches the average structural mean (ASM). Applying the average structural function (ASF) discussed by [Blundell and Powell \(2003\)](#), [Blundell and Powell \(2004\)](#), [Wooldridge \(2005\)](#), and [Wooldridge \(2014\)](#) to a conditional mean on the covariates and unobserved endogenous error produces the ASM.

In the linear regression model, for exogenous covariates  $\mathbf{x}_i$  and  $C$  endogenous regressors  $\mathbf{w}_i$ , we have

$$y_i = \mathbf{x}_i\beta + \mathbf{w}_i\beta_2 + \epsilon_i$$

where the error  $\epsilon_i$  is normal and correlated with  $\mathbf{w}_i$ .

The ASM provides a useful interpretation of  $\beta$  and  $\beta_2$  when the  $\mathbf{w}_i$  are correlated with  $\epsilon_i$ . Because  $\epsilon_i$  is a normally distributed, mean 0, random variable, we can split it into two mean 0, normally distributed, independent parts,

$$\epsilon_i = u_i + \psi_i$$

where  $u_i = \gamma\epsilon_{2i}$  is the unobserved heterogeneity that gives rise to the endogeneity and  $\psi_i$  is an error term with variance  $\sigma_\psi^2$ .

Conditional on the covariates and the unobserved heterogeneity, the conditional mean of  $y_i$  is

$$E(y_i | \mathbf{x}_i, \mathbf{w}_i, u_i) = \mathbf{x}_i\beta + \mathbf{w}_i\beta_2 + u_i$$

Because  $u_i$  is an unobserved random variable, this conditional expectation is not observable. Integrating out the  $u_i$ , just like we do with random effects in panel-data models, produces the ASM,

$$\text{ASM}(\mathbf{x}_i^0, \mathbf{w}_i^0) = \int E(y_i | \mathbf{x}_i^0, \mathbf{w}_i^0, u_i) f(u_i) du_i$$

where  $f(u_i)$  is the marginal distribution of  $u_i$ , and  $\mathbf{x}_i^0$  and  $\mathbf{w}_i^0$  are given covariate values.

Because  $u_i$  has mean 0, we have

$$\text{ASM}(\mathbf{x}_i^0, \mathbf{w}_i^0) = \mathbf{x}_i^0 \boldsymbol{\beta} + \mathbf{w}_i^0 \boldsymbol{\beta}_2$$

So, the ASM is the linear prediction of the main outcome.

## References

- Blundell, R. W., and J. L. Powell. 2003. Endogeneity in nonparametric and semiparametric regression models. In *Advances in Economics and Econometrics: Theory and Applications, Eighth World Congress*, ed. M. Dewatripont, L. P. Hansen, and S. J. Turnovsky, vol. 2, 312–357. Cambridge: Cambridge University Press.
- . 2004. Endogeneity in semiparametric binary response models. *Review of Economic Studies* 71: 655–679.
- Wooldridge, J. M. 2005. Unobserved heterogeneity and estimation of average partial effects. In *Identification and Inference for Econometric Models: Essays in Honor of Thomas Rothenberg*, ed. D. W. K. Andrews and J. H. Stock, 27–55. New York: Cambridge University Press.
- . 2014. Quasi-maximum likelihood estimation and testing for nonlinear models with endogenous explanatory variables. *Journal of Econometrics* 182: 226–234.

## Also see

- [ERM] [eregress](#) — Extended linear regression
- [ERM] [eregress predict](#) — predict after eregress
- [ERM] [predict treatment](#) — predict for treatment statistics
- [ERM] [predict advanced](#) — predict’s advanced features
- [ERM] [eprobit postestimation](#) — Postestimation tools for eprobit
- [U] [20 Estimation and postestimation commands](#)

eregress predict — predict after eregress

Description	Syntax
Options for statistics	Options for how results are calculated
Remarks and examples	Methods and formulas
Also see	

Description

In this entry, we show how to create new variables containing observation-by-observation predictions after fitting a model with `eregress`.

Syntax

You previously fit the model

```
eregress y x1 ..., ...
```

The equation specified immediately after the `eregress` command is called the main equation. It is

$$y_i = \beta_0 + \beta_1 x1_i + \cdots + e_i.y$$

`predict` calculates predictions for `y` in the main equation. The other equations in the model are called auxiliary equations or complications.

The syntax of `predict` is

```
predict [type] newvar [if] [in] [, stdstatistics howcalculated]
```

<i>stdstatistics</i>	Description
<code>mean</code>	linear prediction; the default
<code>xb</code>	linear prediction excluding all complications

<i>howcalculated</i>	Description
<code>default</code>	not fixed; base values from data
<code>fix(<i>endogvars</i>)</code>	fix specified endogenous covariates
<code>base(<i>valspecs</i>)</code>	specify base values of any variables
<code>target(<i>valspecs</i>)</code>	more convenient way to specify <code>fix()</code> and <code>base()</code>

Note: The `fix()` and `base()` options affect results only in models with endogenous variables in the main equation. The `target()` option is sometimes a more convenient way to specify the `fix()` and `base()` options.

*endogvars* are names of one or more endogenous variables appearing in the main equation.

*valspecs* specify the values for variables at which predictions are to be evaluated. Each *valspec* is of the form

*varname* = #

*varname* = (*exp*)

*varname* = *othervarname*

For instance, `base(valspecs)` could be `base(w1=0)` or `base(w1=0 w2=1)`.

Notes:

- (1) `predict` can also calculate treatment-effect statistics. See [\[ERM\] predict treatment](#).
- (2) `predict` can also make predictions for the other equations in addition to the main-equation predictions discussed here. See [\[ERM\] predict advanced](#).

## Options for statistics

`mean` specifies that the linear prediction be calculated. In each observation, the linear prediction is the expected value of the dependent variable conditioned on the covariates. Results depend on how complications are handled, which is determined by the [howcalculated](#) options.

`xb` specifies that the linear prediction be calculated ignoring all complications. This prediction corresponds to what would be observed in data in which all the covariates in the main equation were exogenous.

## Options for how results are calculated

By default, predictions are calculated taking into account all complications. This is discussed in [Remarks and examples](#).

`fix(varname ...)` specifies a list of endogenous variables from the main equation to be treated as if they were exogenous. This was discussed in [\[ERM\] intro 3](#) and is discussed further in [Remarks and examples](#) below.

`base(varname = ...)` specifies a list of variables from any equation and values for them. Those values will be used in calculating the expected value of  $e_i.y$ . Errors from other equations spill over into the main equation because of correlations between errors. The correlations were estimated when the model was fit. The amount of spillover depends on those correlations and the values of the errors. This issue was discussed in [\[ERM\] intro 3](#) and is discussed further in [Remarks and examples](#) below.

`target(varname = ...)` is sometimes a more convenient way to specify the `fix()` and `base()` options. You specify a list of variables from the main equation and values for them. Those values override the values of the variables calculating  $\beta_0 + \beta_1 x_{1i} + \dots$ . Use of `target()` is discussed in [Remarks and examples](#) below.



## Remarks and examples

Remarks are presented under the following headings:

- [How to think about the model you fit](#)
- [How to think about predictions](#)
- [The default calculation](#)
- [The fix\(\) calculation](#)
- [The base\(\) calculation](#)
- [The alternative target\(\) option for making the fix\(\) and base\(\) calculations](#)

### How to think about the model you fit

You have fit a model, perhaps by typing

```
. eregress y x1 x2 (1)
```

or

```
. eregress y x1 x2, endogenous(x1 = z1 z2, nomain) (2)
```

or

```
. eregress y x1 x2 selected, endogenous(x1 = z1 z2, nomain) (3)
> select(selected = x1 z3 z4)
```

The equation specified immediately after the `eregress` command is called the main equation. In the models above, it is

```
. eregress y x1 x2 (1)
. eregress y x1 x2 (2)
. eregress y x1 x2 selected (3)
```

The equations specified in the options are called the auxiliary equations or complications. In the models above, they are

```
none (1)
. endogenous(x1 = z1 z2, nomain) (2)
. endogenous(x1 = z1 z2, nomain) select(selected = x1 z3 z4) (3)
```

The auxiliary equations arose because of complications in the data you used to fit the model. The focus of ERM is on fitting the main equation correctly in the presence of complications.

### How to think about predictions

`predict` can make different kinds of predictions. The kind is specified by the how-to-calculate options.

Option	Result
<i>none specified</i>	calculate $\hat{y}_i$ for data assuming they were generated just as the data used to fit the model were generated
<code>fix()</code>	calculate $\hat{y}_i$ for data generated with the complication for the specified variable removed
<code>base()</code>	calculate $\hat{y}_i$ just as in the <i>none specified</i> case, but calculate correlation-of-errors effects using the values for the covariates specified

## The default calculation

When you use `predict` without options, you type

```
. predict yhat
```

`predict` calculates the expected values of  $y_i$  that would be observed given the complications present in your data.

Let's consider the three models we mentioned earlier.

```
. eregress y x1 x2 (1)
. eregress y x1 x2, endogenous(x1 = z1 z2, nomain) (2)
. eregress y x1 x2 selected, endogenous(x1 = z1 z2, nomain) (3)
> select(selected = x2 z3 z4)
```

The result from typing `predict yhat` without options will be

1. The expected values of  $y_i$  given  $x1$  and  $x2$ .
2. Same as (1) and taking into account that  $x1$  is endogenous and predicted by  $z1$  and  $x1$ .
3. Same as (2) and taking into account that  $y$  is observed only if the observation is `selected` and that `selected` is endogenous and given by  $x2$ ,  $z3$ , and  $z4$ .

The other calculation options affect how the auxiliary equations are handled. Because model (1) has no auxiliary equations, the default prediction is the only one possible in its case.

`predict` without options can be used to calculate expected values with the data used in fitting the model and with other data that include the same complications. After fitting the model, you can type

```
. use anotherdataset
. predict yhat
```

You will sometimes use `predict` to calculate counterfactuals. If you do that, the default calculation can be used for changes in covariates that are exogenous in the main equation and appear in the main equation only.

Having fit any of the above models, you could type

```
. generate x2orig = x2
. replace x2 = 1000
. predict yhat
. replace x2 = x2orig
```

The predictions obtained would be the expected value of  $y$  given that each subject had  $x2$  set to 1,000.

A safer approach, however, is to specify the `base()` option. We will discuss `base()` in detail below, but the better solution is

```
. generate x2orig = x2
. replace x2 = 1000
. predict yhat, base(x2=x2orig)
. replace x2 = x2orig
```

If `base()` is unnecessary, it will cause no harm to specify it.

## The fix() calculation

The purpose of the other calculation options is to make meaningful counterfactuals when you change the values of endogenous covariates. Option `fix(varname ...)` makes predictions as if the complications associated with *varname* were removed.

Assume you have fit model (3):

```
. eregress y x1 x2 selected, endogenous(x1 = z1 z2, nomain) (3)
> select(selected = x2 z3 z4)
```

Then,

```
. predict yhat1, fix(x1)
```

would produce predictions that correspond to “what would have been observed” if the complication for *x1* had not been present either in the data or in the fitted model. These predicted values would correspond to a world in which the data-generating process was

```
. eregress y x1 x2 selected, select(selected = x2 z3 z4) (3')
```

In this counterfactual world, *x1* is no longer endogenous. This switch from being endogenous to being exogenous is not a technicality. It is full of import. In the real world, *e.x1* is correlated with *e.y*. When we made the default prediction in the previous section, that correlation was taken into account. In this alternative world, there is no correlation. Perhaps *x1* records each subject’s amount of health insurance coverage and *y* is a health outcome. In the world of the data used to fit the model, subjects chose to purchase health insurance, and presumably those who perceived a larger benefit would purchase more. Thus, the correlation between *e.x1* and *e.y* was positive. In the counterfactual world, perhaps purchase of health insurance is mandatory or it is free. Either way, the correlation between *e.x1* and *e.y* becomes 0.

Let’s consider another prediction involving changing an endogenous variable.

```
. predict yhat2, fix(selected)
```

In this counterfactual world, selection is no longer endogenous. The predicted values would correspond to a world in which the data-generating process is

```
. eregress y x1 x2 selected, endogenous(x1 = z1 z2, nomain) (3'')
```

In this counterfactual world, *x1* is back to being endogenous, but selection no longer is. That breaks the correlation between *e.selected* and *e.y* in the same way the previous counterfactual broke the correlation between *e.x1* and *e.y*.

Another possible prediction is

```
. predict yhat2, fix(x1 selected)
```

The predicted values would correspond to a world in which the data-generating process is

```
. eregress y x1 x2 selected (3''')
```

When you use `fix()`, you ordinarily change the values of the variable being fixed. You might type

```
. generate x1orig = x1
. replace x1 = 1 // $1 million
. predict yhat2, fix(x1)
. replace x1 = x1orig
```

or

```
. generate selectedorig = selected
. replace selected = 1          // or 0 as you please
. predict yhat2, fix(x1 selected)
. replace selected = selectedorig
```

or

```
. generate x1orig = x1
. generate selectedorig = selected
. replace x1 = 1                // $1 million
. replace selected = 1          // or 0 as you please
. predict yhat2, fix(x1 selected)
. replace selected = selectedorig
. replace x1 = x1orig
```

## The base() calculation

`fix()` is one way of handling predictions of counterfactuals when an endogenous variable in the main equation is changed. `base()` is the other.

Let's assume you have fit either model (2) or model (3):

```
. eregress y x1 x2, endogenous(x1 = z1 z2, nomain)      (2)
. eregress y x1 x2 selected, endogenous(x1 = z1 z2, nomain) (3)
> select(selected = x2 z3 z4)
```

You cannot haphazardly change the value of an endogenous variable such as `x1` and expect to produce meaningful results. Because of that, you should *not* type

```
. generate x1orig = x1
. replace x1 = x1 + 1
. predict yhat
. replace x1 = x1orig
```

What would happen if you did? In either of the above models, there is an equation for `x1`. It is `endogenous(x1 = z1 z2, nomain)`

which, written mathematically, is

$$x1_i = \gamma_0 + \gamma_1 z1_i + \gamma_2 z2_i + e_i.x1$$

You increased `x1` by 1 but did not change anything else. The equation above still holds, and so incrementing `x1` increased  $e_i.x1$  by 1 too.

What does it mean to increase  $e_i.x1$ ? You are assuming that `x1` increased by 1 because the subjects decided to choose `x1+1` instead of `x1`. The only way that could happen is if they were different subjects.

Here is the thought experiment you just performed. You have data on subjects. What if you had different data on different subjects, each with the same characteristics as the current subjects, but who had chosen a value of `x1` that was one unit larger. Well, if these alternate subjects had chosen a value one unit larger than the current subjects, they would have done so for good reason, and their larger  $e.x1$  would have passed along its effect to the  $e.y$  because of the correlation. The new value of `y` would be the direct effect of `x1` in the `y` equation plus the change in  $e.y$ .

`predict yhat` without options produces the answer to the question that you never wanted to ask. What you wanted to ask was what would be the effect on  $y$  for the current subjects if endogenous variable  $x_1$  was “exogenously” incremented by 1.

`predict, base()` will answer that question.

The subjects in your data are who they are because of their errors. Errors such as  $e.x_1$  are the unobserved things about them that affect their choice of  $x_1$ . You cannot change their errors without changing those unobserved things that make them who they are. If you want to ask about the effects of changes in  $x_1$  holding the subjects constant, you need to ask about changes in  $x_1$  holding  $e_i.x_1$  constant.

`base()` does that and here is how you use it:

```
. generate x1orig = x1
. replace x1 = x1 + 1           // or whatever new values you please
. predict yhat3, base(x1=x1orig)
. replace x1 = x1orig
```

The option says that the unobserved components about the subjects in your data—the unobserved components that make them who they are—are to be calculated ignoring the values stored in  $x_1$  (values that you have changed) and are instead to be calculated at the original values of  $x_1$  (the values that will produce the same endogenously chosen solution). Then, we increase  $x_1$  by 1.

## The alternative `target()` option for making the `fix()` and `base()` calculations

`target()` is sometimes a more convenient way to make predictions using the `fix()` and `base()` calculations.

In the section above, one of the predictions was made by typing

```
. generate x1orig = x1
. replace x1 = x1 + 1           // or whatever new values you please
. predict yhat3, base(x1=x1orig)
. replace x1 = x1orig
```

We could have made the same prediction with `target()` by typing

```
. predict yhat3, target(x1=(x1+1))
```

Using `target()`, we specify the counterfactual calculation and leave variable  $x_1$  unchanged. The unobserved components will be calculated on the basis of the values in variable  $x_1$ .

In the section on `fix()`, one of the predictions was made by typing

```
. generate x1orig = x1
. replace x1 = 1               // $1 million
. predict yhat2, fix(x1)
. replace x1 = x1orig
```

We could have made the same prediction with `target()` by typing

```
. predict yhat, fix(x1) target(x1=1)
```

You can use `target()` by itself as a substitute for `base()`, and you can use `target()` with `fix()`. In both cases, `target()` specifies the counterfactual, and you do not change the data in memory.

## Methods and formulas

See *Methods and formulas* in [\[ERM\] eregress postestimation](#).

## Also see

[\[ERM\] eregress postestimation](#) — Postestimation tools for eregress

[\[ERM\] eregress](#) — Extended linear regression n

# Title

erm options — Extended regression model options

DescriptionSyntaxOptionsAlso see

## Description

This entry describes the options that are common to the extended regression commands; see [\[ERM\] eregress](#), [\[ERM\] eprobit](#), [\[ERM\] eoprobit](#), and [\[ERM\] eintreg](#).

## Syntax

*erm\_cmd* ... [ , *extensions options* ]

*erm\_cmd* is one of [eregress](#), [eprobit](#), [eoprobit](#), or [eintreg](#).

<i>extensions</i>	Description
Model	
<a href="#">endogenous</a> ( <i>enspec</i> )	model for endogenous covariates; may be repeated
<a href="#">entreat</a> ( <i>entrspec</i> )	model for endogenous treatment assignment
<a href="#">extreat</a> ( <i>extrspec</i> )	exogenous treatment
<a href="#">select</a> ( <i>selspec</i> )	probit model for selection
<a href="#">tobitselect</a> ( <i>tselspec</i> )	tobit model for selection

<i>options</i>	Description
Model	
<u>noconstant</u>	suppress constant term
<u>offset</u> ( <i>varname<sub>o</sub></i> )	include <i>varname<sub>o</sub></i> in model with coefficient constrained to 1
<u>constraints</u> ( <i>numlist</i> )	apply specified linear constraints
<u>collinear</u>	keep collinear variables
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 <u>level</u> (95)
<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> (#)	set the number of integration (quadrature) points for integration over four or more dimensions; default is <u>intpoints</u> (128)
<u>triintpoints</u> (#)	set the number of integration (quadrature) points for integration over three dimensions; default is <u>triintpoints</u> (10)
Maximization	
<u>maximize_options</u>	control the maximization process; seldom used
<u>coeflegend</u>	display legend instead of statistics

*enspec* is *depvars<sub>en</sub>* = *varlist<sub>en</sub>* [ , *enopts* ]

where *depvars<sub>en</sub>* is a list of endogenous covariates. Each variable in *depvars<sub>en</sub>* specifies an endogenous covariate model using the common *varlist<sub>en</sub>* and options.

*entrspec* is *depvar<sub>tr</sub>* [= *varlist<sub>tr</sub>*] [ , *tropts* ]

where *depvar<sub>tr</sub>* is a variable indicating treatment assignment. *varlist<sub>tr</sub>* is a list of covariates predicting treatment assignment.

*extrspec* is *tvar* [ , nomain nointeract ]

where *tvar* is a variable indicating treatment assignment.

*selspec* is *depvar<sub>s</sub>* = *varlist<sub>s</sub>* [ , noconstant offset(*varname<sub>o</sub>*) ]

where *depvar<sub>s</sub>* is a variable indicating selection status. *depvar<sub>s</sub>* must be coded as 0, indicating that the observation was not selected, or 1, indicating that the observation was selected. *varlist<sub>s</sub>* is a list of covariates predicting selection.

*tselspec* is *depvar<sub>s</sub>* = *varlist<sub>s</sub>* [ , *tselopts* ]

where *depvar<sub>s</sub>* is a continuous variable. *varlist<sub>s</sub>* is a list of covariates predicting *depvar<sub>s</sub>*. The censoring status of *depvar<sub>s</sub>* indicates selection, where a censored *depvar<sub>s</sub>* indicates that the observation was not selected and a noncensored *depvar<sub>s</sub>* indicates that the observation was selected.



<i>enopts</i>	Description
Model	
<u>probit</u>	treat endogenous covariate as binary
<u>oprobit</u>	treat endogenous covariate as ordinal
<u>nomain</u>	do not add endogenous covariate to main equation
<u>noconstant</u>	suppress constant term
<i>tropts</i>	Description
Model	
<u>nomain</u>	do not add treatment indicator to main equation
<u>nocutsinteract</u>	do not interact treatment with cutpoints
<u>nointeract</u>	do not interact treatment with covariates in main equation
<u>noconstant</u>	suppress constant term
<u>offset</u> ( <i>varname<sub>o</sub></i> )	include <i>varname<sub>o</sub></i> in model with coefficient constrained to 1
nocutsinteract is only available with eoprobit.	
<i>tseopts</i>	Description
Model	
ll( <i>varname</i>   #)	left-censoring variable or limit
ul( <i>varname</i>   #)	right-censoring variable or limit
main	add censored selection variable to main equation
<u>noconstant</u>	suppress constant term
<u>offset</u> ( <i>varname<sub>o</sub></i> )	include <i>varname<sub>o</sub></i> in model with coefficient constrained to 1

## Options

Model
<b>endogenous</b> ( <i>depvars<sub>en</sub></i> = <i>varlist<sub>en</sub></i> [ , <i>enopts</i> ]) specifies the model for endogenous covariates. <i>depvars<sub>en</sub></i> is a list of one or more endogenous covariates modeled with <i>varlist<sub>en</sub></i> . This option may be repeated to allow a different model specification for each endogenous covariate. By default, the endogenous covariates are assumed to be continuous, and a linear Gaussian model is used. Unless the <b>nomain</b> suboption is specified, the variables specified in <i>depvars<sub>en</sub></i> are automatically included in the main equation. The following <i>enopts</i> are available:
<b>probit</b> specifies to use a probit model for the endogenous covariates. <b>probit</b> may not be specified with <b>oprobit</b> ; however, you may specify <b>endogenous</b> (..., <b>probit</b> ) and <b>endogenous</b> (..., <b>oprobit</b> ).
<b>oprobit</b> specifies to use an ordered probit model for the endogenous covariates. <b>oprobit</b> may not be specified with <b>probit</b> ; however, you may specify <b>endogenous</b> (..., <b>probit</b> ) and <b>endogenous</b> (..., <b>oprobit</b> ).
<b>nomain</b> specifies that the endogenous covariate of covariates be excluded from the main model, thus removing the effect. This option is for those who intend to manually construct the effect by adding it to the main model in their own way.
<b>noconstant</b> suppresses the constant term (intercept) in the model for the endogenous covariates.

`entreat()` and `extreat()` specify a model for treatment assignment. You may specify only one.

`entreat(depvartr [= varlisttr] [, trtopts modopts])` specifies a model for endogenous treatment assignment with  $depvar_{tr} = 1$  indicating treatment and  $depvar_{tr} = 0$  indicating no treatment.  $varlist_{tr}$  are the covariates for the treatment model; they are optional.

`extreat(depvartr [, trtopts])` specifies a variable that signals exogenous treatment.  $depvar_{tr} = 1$  indicates treatment and  $depvar_{tr} = 0$  indicates no treatment.

*trtopts* are

`nomain`, `nocutsinteract`, and `nointeract` affect the way the treatment enters the main equation.

`nomain` specifies that the main effect of treatment be excluded from the main equation. Thus, a separate intercept is not estimated for each treatment level. In the case of `eoprobit`, this means separate cutpoints are not added.

`nocutsinteract` specifies that instead of the default of having separate cutpoints for each treatment level, you get one set of cutpoints that are shifted by a constant value for each treatment level. This is implemented by placing a separate constant in the main equation for each treatment level. `nocutsinteract` is available only with `eoprobit`.

`nointeract` specifies that the treatment variable not be interacted with the other covariates in the main equation.

These options allow you to customize how the treatment enters the main equation. When `nomain` and `nointeract` are specified together, they remove the effect entirely, and you will need to explicitly reintroduce the treatment effect.

*modopts* are

`noconstant` suppresses the constant term (intercept) in the treatment model.

`offset(varnameo)` specifies that  $varname_o$  be included in the treatment model with the coefficient constrained to 1.

`select()` and `tobitselect()` specify a model for endogenous sample selection. You may specify only one.

`select(depvars = varlists [, modopts])` specifies a probit model for sample selection with  $varlist_s$  as the covariates for the selection model. When  $depvar_s = 1$ , the model's dependent variable is treated as observed (selected); when  $depvar_s = 0$ , it is treated as unobserved (not selected).

`tobitselect(depvars = varlists [, ll(varname|#) ul(varname|#) main modopts])` specifies a tobit model for sample selection with  $depvar_s$  as a censored selection variable and  $varlist_s$  as the covariates for the selection model.

`ll(arg)` specifies that when  $depvar_s \leq arg$ , the selection variable is treated as censored and the model's dependent variable is unobserved (not selected).

`ul(arg)` specifies that when  $depvar_s \geq arg$ , the selection variable is treated as censored and the model's dependent variable is unobserved (not selected).

`main` specifies that the censored selection variable be included as a covariate in the main equation. By default, it is excluded from the main equation.

Only the uncensored values of the selection variable contribute to the likelihood through the main equation. Thus, the selection variable participates as though it were uncensored.

*modopts* are

`noconstant` suppresses the constant term (intercept) in the selection model.

`offset(varnameo)` specifies that *varname<sub>o</sub>* be included in the selection model with the coefficient constrained to 1.

`noconstant`, `offset(varnameo)`, `constraints(numlist)`, and `collinear`; 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(#)` and `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(#)` and `triintpoints(#)` control the number of integration (quadrature) points used to approximate multivariate normal probabilities in the likelihood and scores.

`intpoints()` sets the number of integration (quadrature) points for integration over four or more dimensions. The number of integration points must be between 3 and 5,000. The default is `intpoints(128)`.

`triintpoints()` sets the number of integration (quadrature) points for integration over three dimensions. The number of integration points must be between 3 and 5,000. The default is `triintpoints(10)`.

When four dimensions of integration are used in the likelihood, three will be used in the scores. The algorithm for integration over four or more dimensions differs from the algorithm for integration over three dimensions.

---

#### 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](#).

Setting the optimization type to `technique(bhhh)` resets the default *vcetype* to `vce(opg)`.

The following option is available with `erm_cmd` but is not shown in the dialog box:  
`coeflegend`; see [R] [estimation options](#).

## Also see

- [ERM] [eintreg](#) — Extended interval regression
- [ERM] [eoprobit](#) — Extended ordered probit regression
- [ERM] [eprobit](#) — Extended probit regression
- [ERM] [eregress](#) — Extended linear regression

Description

Remarks and examples

Menu

Stored results

Syntax

Also see

Options

Description

estat teffects estimates the average treatment effect, average treatment effect on the treated, and potential-outcome mean for ERMs.

Menu

Statistics > Postestimation

Syntax

estat teffects [ , options ]

options	Description
ate	estimate average treatment effect; the default
atet	estimate average treatment effect on the treated
pomean	estimate potential-outcome mean
tlevel(numlist)	calculate treatment effects or potential-outcome means for specified treatment levels
outlevel(numlist)	calculate treatment effects or potential-outcome means for specified levels of ordinal dependent variable
subpop(subspec)	estimate for subpopulation
llevel(#)	set confidence level; default is level(95)
display_options	control columns and column formats, row spacing, line width and factor-variable labeling

Options

- ate estimates the average treatment effect (ATE). This is the default.
- atet estimates the average treatment effect on the treated (ATET). For binary treatments, the ATET is reported for the treated group subpopulation. For ordinal treatments, by default, the ATET is reported for the first noncontrol treatment group subpopulation. You can use the subpop() option to calculate the ATET for a different treatment group.
- pomean estimates the potential-outcome mean (POM).
- tlevel(numlist) specifies the treatment levels for which treatment effects or POMs are calculated. By default, the treatment effects are computed for all noncontrol treatment levels, and the POMs are computed for all treatment levels.

`outlevel(numlist)` specifies the levels of the ordinal dependent variable for which treatment effects or POMs are to be calculated. By default, treatment effects or POMs are computed for all levels of the ordinal dependent variable. This option is only available after `eoprobit`.

`subpop([varname] [if])` specifies the subpopulation for which the ATE, ATET, and POM are calculated. The subpopulation is identified by the indicator variable, by the `if` expression, or by both. A 0 indicates that the observation be excluded, a nonzero indicates that it be included, and a missing value indicates that it be treated as outside of the population (and thus ignored). For instance, for an ordinal treatment `trtvar` with levels 1, 2, and 3, you can specify `subpop(if trtvar==3)` to obtain the ATETs for `trtvar = 3`.

`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.

*display\_options*: `noci`, `nopvalues`, `vsquish`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fnt)`, `pformat(%fnt)`, `sformat(%fnt)`, and `nolstretch`.

`noci` suppresses confidence intervals from being reported in the coefficient table.

`nopvalues` suppresses *p*-values and their test statistics from being reported in the coefficient table.

`vsquish` specifies that the blank space separating factor-variable terms or time-series-operated variables from other variables in the model be suppressed.

`nofvlabel` displays factor-variable level values rather than attached value labels. This option overrides the `fvlabel` setting; see [R] set showbaselevels.

`fvwrap(#)` allows long value labels to wrap the first # lines in the coefficient table. This option overrides the `fvwrap` setting; see [R] set showbaselevels.

`fvwrapon(style)` specifies whether value labels that wrap will break at word boundaries or break based on available space.

`fvwrapon(word)`, the default, specifies that value labels break at word boundaries.

`fvwrapon(width)` specifies that value labels break based on available space.

This option overrides the `fvwrapon` setting; see [R] set showbaselevels.

`cformat(%fnt)` specifies how to format estimates, standard errors, and confidence limits in the estimates table. The maximum format width is 9.

`pformat(%fnt)` specifies how to format *p*-values in the estimates table. The maximum format width is 5.

`sformat(%fnt)` specifies how to format test statistics in the estimates table. The maximum format width is 8.

`nolstretch` specifies that the width of the estimates table not be automatically widened to accommodate longer variable names. The default, `lstretch`, is to automatically widen the estimates table up to the width of the Results window. To change the default, use `set lstretch off`. `nolstretch` is not shown in the dialog box.

## Remarks and examples

`estat teffects` estimates ATEs, ATETs, and POMs after extended regression commands. These are calculated as means of predictions by using `margins` on the predictions from `predict` after the extended regression commands. If the ERM command reported robust standard errors, `estat teffects` reports unconditional standard errors so that inference is for the population effect instead of the sample effect. See *Unconditional standard errors* in [R] margins for more information.

See [\[ERM\] intro 8](#) for an example using `estat teffects`. Methods and formulas for treatment-effect estimation are given in *Methods and formulas* of [\[ERM\] eprobit](#), [\[ERM\] eoprobit](#), [\[ERM\] eregress](#), and [\[ERM\] eintreg](#).

## Stored results

`estat teffects` stores the following in `r()`:

### Macros

<code>r(vce)</code>	<code>vcetype</code> specified in <code>vce()</code>
<code>r(vcetype)</code>	title used to label Std. Err.
<code>r(clustvar)</code>	name of cluster variable

### Matrices

<code>r(b)</code>	estimates
<code>r(V)</code>	variance–covariance matrix of the estimates
<code>r(table)</code>	matrix containing the estimates with their standard errors, test statistics, $p$ -values, and confidence intervals

## Also see

[\[ERM\] eintreg postestimation](#) — Postestimation tools for `eintreg`

[\[ERM\] eoprobit postestimation](#) — Postestimation tools for `eoprobit`

[\[ERM\] eprobit postestimation](#) — Postestimation tools for `eprobit`

[\[ERM\] eregress postestimation](#) — Postestimation tools for `eregress`

example 1a — Linear regression with continuous endogenous covariate

DescriptionRemarks and examplesAlso see

Description

In this example, we show how to estimate and interpret the results of an extended regression model with a continuous outcome and continuous endogenous covariate.

Remarks and examples

The fictional State University is studying the relationship between the high school grade point average (GPA) of the students it admits and their final college GPA. They suspect that unobserved ability affects both high school GPA and college GPA. Thus, high school GPA is an endogenous covariate.

Using data on the 2,500 students in the cohort expected to graduate in 2010, the researchers at State U model college GPA (`gpa`) as a function of high school GPA (`hsgpa`). In both cases, GPA is measured in 0.01 increments, and we ignore complications due to the boundary points. We also ignore that, unfortunately, State U has a high dropout rate and college GPA is missing for these students, leaving the researchers with a sample of about 1,500 students.

The State U researchers expect that the effect of high school competitiveness on college GPA is negligible once high school GPA is controlled for. So they include a ranking of the high school (`hscomp`) as an instrumental covariate for high school GPA. They include parental income measured in \$10,000s, which they believe may also influence student performance, in the main model and in the model for high school GPA.



```
. use http://www.stata-press.com/data/r15/class10
(Class of 2010 profile)

. eregress gpa income, endogenous(hsgpa = income i.hscomp)
Iteration 0:   log likelihood = -638.58598
Iteration 1:   log likelihood = -638.58194
Iteration 2:   log likelihood = -638.58194

Extended linear regression               Number of obs   =       1,528
                                         Wald chi2(2)    =       1167.79
Log likelihood = -638.58194              Prob > chi2      =       0.0000
```

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
gpa						
income	.0575145	.0055174	10.42	0.000	.0467007	.0683284
hsgpa	1.235868	.133686	9.24	0.000	.9738484	1.497888
_cons	-1.217141	.3828614	-3.18	0.001	-1.967535	-.4667464
hsgpa						
income	.0356403	.0019553	18.23	0.000	.0318079	.0394726
hscomp						
moderate	-.1310549	.0136503	-9.60	0.000	-.1578091	-.1043008
high	-.2331173	.0232712	-10.02	0.000	-.278728	-.1875067
_cons	2.951233	.0164548	179.35	0.000	2.918982	2.983483
var(e.gpa)	.1436991	.0083339			.1282592	.1609977
var(e.hsgpa)	.0591597	.0021403			.05511	.063507
corr(e.hsgpa, e.gpa)						
	.2642138	.0832669	3.17	0.002	.0948986	.4186724

The estimate of the correlation between the errors from the main and auxiliary equations is 0.26. The  $z$  statistic may be used for a Wald test of the null hypothesis that there is no endogeneity. The researchers reject this hypothesis. Because the estimate is positive, they conclude that unobservable factors that increase high school GPA tend to also increase college GPA.

Having satisfied themselves that it is appropriate to account for endogeneity of high school GPA, they examine the coefficient estimates. The estimates for the main equation are interpreted just like those from `regress`; see [R] [regress](#). For example, the researchers expect the difference in college GPA is about 1.24 points for students with a difference of 1 point in high school GPA.

As we discussed in [ERM] [intro 8](#), the coefficients on `hsgpa` and `income` in this regression pretty much say everything there is to say about how college GPA changes when either high school GPA or parents' income changes. This is true because our model is linear and we have no interactions. We could make this the end of our story. But it is not the end if we want to ask questions about expected levels of college GPA.

If we want to ask questions about the eventual level of college GPA, we must be specific about how we arrived at our values for `hsgpa`. Let's look at a single observation; we will pretend it is for Billy.

```
. gen str name = "Billy" in 537
(2,499 missing values generated)
. list income if name=="Billy"
```

	income
537.	2

What if we don't have records from Billy's high school and all we know about Billy is his parents' income? We could form counterfactuals about Billy. We could fix Billy's high school GPA at 2.00, and we could fix his high school GPA at 3.00. These are values we are choosing, not the value that Billy arrived at through his own actions. We'll let `margins` give us the expected values for college GPA under these two counterfactuals.

```
. margins if name=="Billy", at(hsgpa=(2 3)) predict(fix(hsgpa))
Warning: prediction constant over observations.

Predictive margins                                Number of obs      =           1
Model VCE    : OIM

Expression   : mean of gpa, predict(fix(hsgpa))
1._at        : hsgpa                =           2
2._at        : hsgpa                =           3
```

	Delta-method		z	P> z	[95% Conf. Interval]	
	Margin	Std. Err.				
_at						
1	1.369625	.1251674	10.94	0.000	1.124301	1.614948
2	2.605493	.0190405	136.84	0.000	2.568174	2.642811

When we set Billy's high school GPA to 2.00 and consider his parents' income of \$20,000, Billy's expected college GPA is 1.37. More correctly, this is the expected GPA for anyone whose parents' income is \$20,000 and whose high school GPA is fixed at 2.00. Keeping his parents' income constant and fixing his high school GPA at 3.00, we see that Billy's expected college GPA rises to 2.61.

But in reality, we know more about Billy.

```
. list gpa hsgpa income hscomp if name=="Billy"
```

	gpa	hsgpa	income	hscomp
537.	1.03	2	2	high

And with this, we can ask a slightly different question. What is Billy's expected GPA given all that we know about him, including the competitiveness of his high school and the unobserved thing or things that drive the correlation between high school and college GPAs? What if we further ask how that expectation would change if we granted Billy one additional unit of high school GPA, taking him from 2.00 to 3.00. These are the same two counterfactuals for the value of high school GPA, but a different assumption about how Billy arrived at a 2.00. To obtain these counterfactuals, we run the same `margins` command, changing the `fix()` option to `base()`.

```
. generate hsgpaT = hsgpa                                // Observed ("True") H.S. GPA
. margins if name=="Billy", at(hsgpa=(2 3)) predict(base(hsgpa=hsgpaT))
Warning: prediction constant over observations.
Predictive margins                                         Number of obs      =          1
Model VCE      : OIM
Expression    : mean of gpa, predict(base(hsgpa=hsgpaT))
1._at         : hsgpa              =          2
2._at         : hsgpa              =          3
```

	Delta-method					
	Margin	Std. Err.	z	P> z	[95% Conf. Interval]	
_at						
1	1.044564	.1242365	8.41	0.000	.8010648	1.288063
2	2.280432	.0207685	109.80	0.000	2.239726	2.321138

The numbers are not the same. The expected GPA of 1.04 is closer to Billy’s true value of 1.03 than was the estimate using only `income`. That need not be the case for any individual, but given that we used more information, we would expect it to be true if we averaged over others with the same characteristics.

As discussed in [ERM] intro 8, we needed to save Billy’s true value of `hsgpa` because `margins` manipulates the data to obtain its results. We did not need to do this with the `fix()` option because predictions using `fix()` do not care what Billy’s true value of `hsgpa` is or how he arrived at that value. Predictions using `base()`, on the other hand, use Billy’s true value of `hsgpa` and all information from the model about how Billy arrived at that GPA. The `base()` option instructs `margins` to use true `hsgpaT` when it formed both of its counterfactuals. Thus, both counterfactuals include information about his high school’s competitiveness and information about the unobserved factor or factors creating the correlation between GPAs. The same values for this information are used when `margins` creates each counterfactual. We could say that, compared with the counterfactuals computed under `fix()`, these counterfactuals include more of what makes Billy, Billy. They are still the expected value for anyone with the same covariates, but they incorporate the fact that the GPA of 2.0 was arrived at through Billy’s own actions and include the competitiveness of his high school.

In the parlance of treatment effects, our first set of estimates could be called the potential outcomes given the fixed treatment levels: 2.00 and 3.00. If that doesn’t help your understanding, then skip this paragraph. The second set of values would be the counterfactuals required to estimate the treatment effect on the untreated (TEU). Why are we being so cagey with the language—“could be” instead of “are” and “counterfactual” instead of “potential outcome” in the second case? Experts in treatment effects don’t like applying the term “potential outcome” when the treatment is continuous. That implies an infinite number of potential outcomes. They are even protective of the term when used to create the pieces needed for the TEU. Regardless, the computation is exactly what would be done to form these potential outcomes for a binary or ordinal treatment, and the interpretation conveys the same meaning.

Neither the `fix()` nor the `base()` counterfactuals can be said to be better. They simply answer different questions. When we consider exogenous changes to variables like high school GPA, the counterfactuals from `base()` will often be more relevant to answering many questions. Whether a guidance counselor or a policy maker is asking the question, both are likely to face the existing GPAs of individual students or those in the population.

Let’s take the next step and estimate the resulting changes in expected college GPA for our two situations. We just need to add `contrast(at(r))` to each of our two `margins` commands.

```
. margins if name=="Billy", at(hsgpa=(2 3)) predict(fix(hsgpa))
> contrast(at(r) effects nowald)
Warning: prediction constant over observations.

Contrasts of predictive margins
Model VCE      : OIM
Expression     : mean of gpa, predict(fix(hsgpa))
1._at          : hsgpa          =          2
2._at          : hsgpa          =          3
```

	Delta-method		z	P> z	[95% Conf. Interval]	
	Contrast	Std. Err.				
_at (2 vs 1)	1.235868	.133686	9.24	0.000	.9738484	1.497888

```
. margins if name=="Billy", at(hsgpa=(2 3)) predict(base(hsgpa=hsgpaT))
> contrast(at(r) effects nowald)
Warning: prediction constant over observations.

Contrasts of predictive margins
Model VCE      : OIM
Expression     : mean of gpa, predict(base(hsgpa=hsgpaT))
1._at          : hsgpa          =          2
2._at          : hsgpa          =          3
```

	Delta-method		z	P> z	[95% Conf. Interval]	
	Contrast	Std. Err.				
_at (2 vs 1)	1.235868	.133686	9.24	0.000	.9738484	1.497888

As we have said repeatedly, the estimates of the effects are the same. It does not matter how Billy arrived at his 2.00. What's more, the standard errors are the same, and they are the same as the standard error of the regression coefficient from our `eregress` output. In this case, the additional information that was so important in getting the right GPA estimates is subtracted out when we compute the differences. That is a direct result of the model being linear and having additive errors. Stretching the parlance of treatment effects again, we could call our first contrast an estimate of the treatment effect and the second a treatment effect on the untreated. For linear models without interactions, these are always the same value.

Would we see anything different if we averaged the effects over the sample to get estimates of the effects in the population? Just remove Billy from the commands.

```
. margins, at(hsgpa=(2 3)) predict(fix(hsgpa)) contrast(at(r) effects nowald)
Contrasts of predictive margins
Model VCE      : OIM
Expression    : mean of gpa, predict(fix(hsgpa))
1._at         : hsgpa          =          2
2._at         : hsgpa          =          3
```

	Delta-method				
	Contrast	Std. Err.	z	P> z	[95% Conf. Interval]
_at (2 vs 1)	1.235868	.133686	9.24	0.000	.9738484    1.497888

```
. margins, at(hsgpa=(2 3)) predict(base(hsgpa=hsgpaT)) contrast(at(r) effects nowald)
(output omitted)
```

Not surprisingly, the estimated effect is still 1.24—the same value we have gotten every time, the same value as the coefficient on `hsgpa`. Perhaps more surprisingly, the standard error of the population-average estimate is also the same as the standard error of the coefficient. We don’t gain or lose any information when we take an average over an estimate that is constant for all the observations.

We leave it to you to run the last command and see that `fix()` and `base()` produce the same results.

In linear models without interactions, we have just seen that the effects are the same for many questions, but the levels are often different. In nonlinear models, these differences in the levels will lead to differences in the effects.

The models in the remaining two examples in this series, [\[ERM\] example 1b](#) and [\[ERM\] example 1c](#), have exactly the same interpretation we gave to the model in this entry. Adding interval censoring and endogenous sample selection do not affect either the relevant questions or how they are answered.

Also see

- [\[ERM\] eregress](#) — Extended linear regression
- [\[ERM\] eregress postestimation](#) — Postestimation tools for `eregress`
- [\[ERM\] intro 3](#) — Endogenous covariates features
- [\[ERM\] intro 8](#) — Conceptual introduction via worked example

example 1b — Interval regression with continuous endogenous covariate

DescriptionRemarks and examplesAlso see

Description

Continuing from [ERM] [example 1a](#), we now consider the case where the dependent variable is interval-censored. We fit this model using `eintreg`.

Remarks and examples

We now assume that, for reasons of confidentiality, the researchers conducting the study do not observe the actual college GPA for those with a GPA below 2.0. For the rest, they are given college GPA only in increments of 0.5 points. So the outcome has both left- and interval-censored observations. The model remains the same.

The lower and upper endpoints for college GPA are stored in `gpal` and `gpau`. Both variables contain a missing value for students who dropped out of college. Other than the change in command name and specification of the dependent variable, the command to fit the model is exactly the same.

```
. eintreg gpal gpau income, endogenous(hsgpa = income i.hscomp)
Iteration 0:  log likelihood = -1716.9969
Iteration 1:  log likelihood = -1716.9968

Extended interval regression                                Number of obs      =          1,528
                                                           Uncensored          =              0
                                                           Left-censored       =             150
                                                           Right-censored      =              0
                                                           Interval-cens.      =           1,378

                                                           Wald chi2(2)        =           912.68
                                                           Prob > chi2          =           0.0000

Log likelihood = -1716.9968
```

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
income	.0551638	.0057859	9.53	0.000	.0438236	.066504
hsgpa	1.111672	.1407083	7.90	0.000	.8358891	1.387456
_cons	-.8180699	.4032468	-2.03	0.042	-1.608419	-.0277207
hsgpa						
income	.0356351	.0019553	18.22	0.000	.0318027	.0394675
hscomp						
moderate	-.1317151	.0136277	-9.67	0.000	-.1584249	-.1050052
high	-.2320803	.0233633	-9.93	0.000	-.2778715	-.186289
_cons	2.951568	.0164465	179.46	0.000	2.919333	2.983802
var(e.gpal)						
var(e.hsgpa)	.1354248	.0090267			.1188397	.1543245
	.0591594	.0021403			.0551097	.0635066
corr(e.hsgpa, e.gpal)						
	.2700108	.0897936	3.01	0.003	.0868241	.4355353

We again find that unobservable factors that increase high school GPA tend to increase college GPA. The parameter estimates here are interpreted just as we did in [\[ERM\] example 1a](#). In that example, the estimated coefficient on `hsgpa` was 1.24; here it is 1.11. Like the relationship between `regress` and `intreg`, the 1.24 and 1.11 estimate the same parameter, the relationship between `hsgpa` and the uncensored outcome.

We will not further interpret this model here. Instead we refer you to the interpretation in [\[ERM\] example 1a](#). The interval censoring of the dependent variable demonstrated here makes no difference in what commands you would type to answer questions or in how you would interpret the results of those commands. In fact, we encourage you to run the commands discussed in [\[ERM\] example 1a](#) on this model and compare the results.

Because interval regression is a generalization of tobit regression, you can also use `eintreg` to fit a tobit model with endogenous selection. However, you must convert your dependent variable into interval form. We illustrate how to do this in [\[ERM\] intro 7](#).

## Also see

[\[ERM\] eintreg](#) — Extended interval regression

[\[ERM\] eintreg postestimation](#) — Postestimation tools for `eintreg`

[\[ERM\] intro 3](#) — Endogenous covariates features

[\[ERM\] intro 8](#) — Conceptual introduction via worked example

# Title

example 1c — Interval regression with endogenous covariate and sample selection

DescriptionRemarks and examplesAlso see

## Description

In [ERM] [example 1a](#) and [ERM] [example 1b](#), we ignored the observations that were dropped because of missing data on GPA. In this example, we show you how to fit a model that includes a continuous endogenous covariate, a censored outcome, and endogenous sample selection.

## Remarks and examples

In the previous two examples, the researchers excluded students who dropped out of college because they are missing college GPA data on these students. So they were estimating parameters for the population of students who graduate from college. Let’s suppose they are interested in expected college GPA for all students who enroll, even those who drop out. They suspect that unobserved ability affects both the decision to stay in school and college GPA and thus that they have an endogenously selected sample.

To model the selection, they need a covariate that affects the probability that they observe a student’s GPA but does not affect the level of the student’s GPA. They include an indicator for whether the student participated in a retention program and whether the student had a roommate who also went to State U. They expect that students with a roommate who went to the same college were more likely to remain in school because they felt more included in the college environment.



```
. eintreg gpal gpau income, endogenous(hsgpa = income i.hscomp)
> select(graduate = hsgpa income i.roommate i.program)
(iteration log omitted)

Extended interval regression                                Number of obs    =      2,500
                                                           Selected       =      1,528
                                                           Nonselected      =       972
                                                           Uncensored       =        0
                                                           Left-censored    =       150
                                                           Right-censored   =        0
                                                           Interval-cens.   =     1,378
                                                           Wald chi2(2)     =     734.96
                                                           Prob > chi2      =     0.0000

Log likelihood = -2851.3222
```

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
income	.0338548	.0075484	4.49	0.000	.0190602	.0486495
hsgpa	1.19378	.1443563	8.27	0.000	.9108467	1.476713
_cons	-.7895643	.3908796	-2.02	0.043	-1.555674	-.0234543
graduate						
hsgpa	2.215481	.4411331	5.02	0.000	1.350876	3.080086
income	.1920393	.0162334	11.83	0.000	.1602224	.2238563
roommate						
yes	.1547087	.0455906	3.39	0.001	.0653528	.2440645
1.program	.4858749	.0523443	9.28	0.000	.383282	.5884678
_cons	-7.524521	1.237529	-6.08	0.000	-9.950034	-5.099008
hsgpa						
income	.047866	.0016981	28.19	0.000	.0445377	.0511942
hscomp						
moderate	-.1337635	.0115749	-11.56	0.000	-.1564499	-.1110771
high	-.2284481	.0190089	-12.02	0.000	-.2657049	-.1911914
_cons	2.793802	.0132125	211.45	0.000	2.767906	2.819698
var(e.gpal)	.1753568	.0085604			.1593564	.1929636
var(e.hsgpa)	.0685863	.0019399			.0648876	.0724958
corr(e.gra~e, e.gpal)	-.9124422	.0327448	-27.87	0.000	-.9583429	-.8205981
corr(e.hsgpa, e.gpal)	.0534114	.0937195	0.57	0.569	-.1300101	.2332982
corr(e.hsgpa, e.graduate)	.2747613	.0955172	2.88	0.004	.079342	.4498437

The coefficients from the main equation for `hsgpa` continue to be interpreted as in [\[ERM\] example 1b](#). Now, however, they are estimates for the population of all admitted students, not the population of all graduates. The estimated effect of high school GPA for this population is slightly higher, 1.19 compared with 1.11.

As with [\[ERM\] example 1b](#), we will not further interpret this model here. Instead we refer you to the interpretation performed in [\[ERM\] example 1a](#). The addition of endogenous sample selection makes no difference in what commands you would type to answer questions or to how you would interpret the results of those commands. In fact, we encourage you to run the commands discussed in [\[ERM\] example 1a](#) on this model and compare the results. The only thing to keep in mind is that now the population we are making inferences about is all students admitted to school.

## Also see

[ERM] [eintreg](#) — Extended interval regression

[ERM] [eintreg postestimation](#) — Postestimation tools for eintreg

[ERM] [intro 3](#) — Endogenous covariates features

[ERM] [intro 4](#) — Endogenous sample-selection features

[ERM] [intro 8](#) — Conceptual introduction via worked example

example 2a — Linear regression with binary endogenous covariate

DescriptionRemarks and examplesAlso see

## Description

In this example, we show how to estimate and interpret the results of an extended regression model with a continuous outcome and endogenous binary covariate.

## Remarks and examples

Suppose that we want to study the effect of having a college degree on wages. One way to approach the problem is to look at the coefficient on an indicator for whether an individual has a college degree. This gives us an idea of how different the average wage is for individuals with a college degree compared with those without one. However, as in [ERM] [example 1a](#), we suspect that unobserved factors such as ability affect both the probability of graduating from college and wage level. Thus, we need to account for the potential endogeneity of the indicator for having a college degree.

In our fictional study, we collect data on the hourly wages (`wage`) and educational attainment (`college`) of 6,000 adults. We believe that differences in job tenure (`tenure`) and age (`age`) may also affect wages. We can control for these covariates by specifying them in the main equation. We specify `college` in the `endogenous()` option, but this time we also include the `probit` suboption to indicate that the variable is binary. We model graduation as a function of the level of parental education (`peduc`), which we assume does not have a direct effect on wage.

```
. use http://www.stata-press.com/data/r15/wageed
(Wages for 20 to 74 year olds, 2015)
. eregress wage c.age#c.age tenure, endogenous(college = i.peduc, probit)
> vce(robust)

Iteration 0:   log pseudolikelihood = -18063.148
Iteration 1:   log pseudolikelihood =  -18060.2
Iteration 2:   log pseudolikelihood = -18060.164
Iteration 3:   log pseudolikelihood = -18060.164

Extended linear regression              Number of obs      =       6,000
                                         Wald chi2(4)        =      7584.74
Log pseudolikelihood = -18060.164      Prob > chi2         =       0.0000
```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
wage						
age	.4200372	.0163312	25.72	0.000	.3880286	.4520457
c.age#c.age	-.0033523	.0001759	-19.06	0.000	-.003697	-.0030075
tenure	.4921838	.0182788	26.93	0.000	.4563581	.5280095
college						
yes	5.238087	.1721006	30.44	0.000	4.900776	5.575398
_cons	5.524288	.3428735	16.11	0.000	4.852268	6.196307
college						
peduc						
college	.8605996	.0361723	23.79	0.000	.7897032	.9314959
graduate	1.361257	.0490862	27.73	0.000	1.26505	1.457465
doctorate	1.583818	.119513	13.25	0.000	1.349577	1.818059
_cons	-.9731264	.0294779	-33.01	0.000	-1.030902	-.9153508
var(e.wage)	8.99487	.2465919			8.524314	9.491402
corr(e.col~e, e.wage)	.5464027	.0286061	19.10	0.000	.4879055	.600014

The estimated correlation between the errors from the main and auxiliary equations is 0.55 and is significantly different from 0. We conclude that having a college degree is endogenous and that unobservable factors that increase the probability of graduating from college tend to also increase wages.

We find that graduating from college increases the expected wage by \$5.24 given a person's age and employment tenure. This estimate is different than comparing the average wages for college graduates and noncollege graduates.

```
. tabulate college, summarize(wage)
```

indicator for college degree	Summary of hourly wage		Freq.
	Mean	Std. Dev.	
no	17.768516	3.0674174	3,766
yes	25.520703	5.045888	2,234
Total	20.654913	5.4248886	6,000

The difference in the average wages is \$7.75, but unlike our regression coefficient, that value does not adjust for the different distribution of ages and tenures among college graduates and noncollege graduates.

Another approach to this problem is the potential-outcomes framework. With this approach, we consider the expected wage for each individual without a college degree versus the expected wage for each individual with a college degree. Specifically, we might like to know the average expected change in wages for those who complete college. This is called the average treatment effect on the treated. We consider this approach in [\[ERM\] example 2b](#) and [\[ERM\] example 2c](#).

[\[ERM\] example 2c](#) also includes an interpretation of how the expected level of income varies by age, tenure, and whether one graduates from college. That analysis could also be applied to this model.

## Also see

[\[ERM\] eregress](#) — Extended linear regression

[\[ERM\] eregress postestimation](#) — Postestimation tools for eregress

[\[ERM\] estat teffects](#) — Average treatment effects for extended regression models

[\[ERM\] intro 8](#) — Conceptual introduction via worked example

example 2b — Linear regression with exogenous treatment

DescriptionRemarks and examplesAlso see

Description

In this example, we show how to estimate and interpret the results of an extended regression model with a continuous outcome and exogenous binary treatment.

Remarks and examples

In [ERM] [example 2a](#), we analyzed the effect of having a college degree on wages as a binary endogenous covariate. Now suppose that we approach our research question instead in the potential-outcomes framework. With this approach, we consider the expected wage for each individual without a college degree versus the expected wage for each individual with a college degree. Specifically, we might like to know the average expected change in wages for those who complete college, the average treatment effect on the treated (ATET).

As before, we use `wageed.dta` with educational attainment data on 6,000 adults. We control for differences in job tenure (`tenure`) and age (`age`) by specifying them in the main equation. For the time being, we consider the treatment (`college`) to be exogenous. We want to make inferences about the average effect of a college degree on the wages of all individuals who complete college, not just the subjects in our study sample, so we specify `vce(robust)`. This will allow us to estimate the standard errors of the ATET accounting for the fact the variables in our sample represent just one draw from the population.

```
. use http://www.stata-press.com/data/r15/wageed
(Wages for 20 to 74 year olds, 2015)

. eregress wage c.age#c.age tenure, extreat(college) vce(robust)
Iteration 0:   log pseudolikelihood = -13989.589
Iteration 1:   log pseudolikelihood = -13989.586
Iteration 2:   log pseudolikelihood = -13989.586

Extended linear regression                                Number of obs      =      6,000
                                                         Wald chi2(8)       =    439363.91
Log pseudolikelihood = -13989.586                      Prob > chi2        =      0.0000
```

wage	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
college#c.age						
no	.2454534	.0180052	13.63	0.000	.2101638	.280743
yes	.7042756	.0225386	31.25	0.000	.6601007	.7484505
college#c.age#c.age						
no	-.0018998	.0001935	-9.82	0.000	-.002279	-.0015206
yes	-.0054223	.000243	-22.31	0.000	-.0058986	-.0049459
college#c.tenure						
no	.3206065	.0207164	15.48	0.000	.2800031	.36121
yes	.4935213	.0257599	19.16	0.000	.4430329	.5440097
college						
no	9.851871	.3701276	26.62	0.000	9.126435	10.57731
yes	4.384709	.4654545	9.42	0.000	3.472435	5.296983
var(e.wage)	6.20477	.1152627			5.982922	6.434843

Because we specified the command as a treatment-effects model, `eregress` automatically interacts the `college` variable with all other covariates in the model, thus essentially creating separate models for those who graduate from college and those who do not. There is nothing wrong with interpreting the coefficients. This is, after all, just a regression. The coefficients labeled `no` are the estimates of the parameters of the wage model for those who are not college graduates. The coefficients labeled `yes` are the estimates of the parameters for the model of those who are college graduates. Tenure in the company has a larger effect for college graduates than nongraduates. It is 49 cents an hour per tenure year for college graduates and 32 cents for nongraduates. The effect of age is more difficult to interpret because of the quadratic term. The effect of age is clearly different between the groups, but the pattern of that difference is not obvious. See [ERM] [example 2c](#) for some tools you could apply to this model that would make that pattern obvious. The effect of college graduation is harder still to see. For any person, it would be the difference of the values predicted by the two models. Again, see [ERM] [example 2c](#) for ways to visualize the effect.

If we are interested only in the average effect, we can estimate that using the `estat teffects` command.

```
. estat teffects, atet
Predictive margins                                Number of obs    =      6,000
                                                Subpop. no. obs  =      2,234
```

	Unconditional Margin	Std. Err.	z	P> z	[95% Conf. Interval]	
ATET college (yes vs no)	7.62719	.0863465	88.33	0.000	7.457954	7.796426

The average wage is estimated to be \$7.63 higher per hour for the population of college graduates than the wage would have been if those same individuals had not completed college.

We have ignored several potential complications in this example. One of which is that unobserved factors such as ability that influence whether individuals complete college could also influence their wage. In that case, the treatment assignment (obtaining a college degree) would be endogenous. If the treatment were endogenous, we would model its coefficients and the correlation between the treatment assignment errors and the outcome errors. See [ERM] [example 2c](#) for an example with an endogenous treatment.

Also see

- [ERM] [eregress](#) — Extended linear regression
- [ERM] [eregress postestimation](#) — Postestimation tools for eregress
- [ERM] [estat teffects](#) — Average treatment effects for extended regression models
- [ERM] [intro 8](#) — Conceptual introduction via worked example



# Title

example 2c — Linear regression with endogenous treatment

DescriptionRemarks and examplesAlso see

## Description

Continuing from [\[ERM\] example 2b](#), we now consider the case where the treatment is endogenous.

## Remarks and examples

In [\[ERM\] example 2b](#), we assumed that graduating from college was an exogenous treatment. However, unobserved factors such as ability may affect whether individuals graduate from college and also affect their wage. Thus, it may be more appropriate for us to treat having a college degree as an endogenous treatment. We found endogeneity in [\[ERM\] example 2a](#), which analyzes the treatment instead as a binary endogenous covariate. You may want to compare the result of this example with the results from [\[ERM\] example 2b](#).

Because college graduation is now assumed to be endogenous, we must specify a model for college. We model graduation as a function of the level of parental education (`peduc`), which we further assume does not have a direct effect on wage. The endogenous treatment equation is specified in option `entreat()`.

```
. eregress wage c.age##c.age tenure, entreat(college = i.peduc) vce(robust)
Iteration 0:   log pseudolikelihood = -17382.446
Iteration 1:   log pseudolikelihood = -17381.922
Iteration 2:   log pseudolikelihood = -17381.92

Extended linear regression               Number of obs   =       6,000
                                         Wald chi2(8)     =    348743.60
Log pseudolikelihood = -17381.92        Prob > chi2      =       0.0000
```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
wage						
college#						
c.age						
no	.2338084	.0176633	13.24	0.000	.199189	.2684279
yes	.6777385	.0219827	30.83	0.000	.6346531	.7208239
college#						
c.age#c.age						
no	-.0018611	.00019	-9.79	0.000	-.0022335	-.0014887
yes	-.0052533	.0002372	-22.14	0.000	-.0057183	-.0047883
college#						
c.tenure						
no	.3948863	.0207452	19.04	0.000	.3542263	.4355462
yes	.5883544	.0257213	22.87	0.000	.5379415	.6387673
college						
no	10.86301	.3675208	29.56	0.000	10.14268	11.58333
yes	3.184255	.4612019	6.90	0.000	2.280316	4.088194
college						
peduc						
college	.849575	.0356419	23.84	0.000	.7797181	.9194318
graduate	1.347272	.0491996	27.38	0.000	1.250843	1.443701
doctorate	1.541025	.1174797	13.12	0.000	1.310769	1.771281
_cons	-.973061	.0292791	-33.23	0.000	-1.030447	-.9156749
var(e.wage)	7.629807	.2245651			7.202122	8.082889
corr(e.col~e, e.wage)	.623109	.0267317	23.31	0.000	.5679046	.6727326

As in [ERM] [example 2b](#), most of the coefficients are difficult to directly interpret. The estimated correlation between the errors from the main and auxiliary equations is 0.62. The  $z$  statistic may be used for a Wald test of the null hypothesis that there is no endogenous treatment. We reject this hypothesis and conclude that having a college degree is an endogenous treatment. Because the estimate is positive, we conclude that unobserved factors that increase the chance of having a college degree also tend to increase wage.

We can use `estat teffects` to estimate the average effect of a college degree on wage. We use the `atet` option to estimate the ATET.

```
. estat teffects, atet
Predictive margins                                Number of obs    =      6,000
                                                Subpop. no. obs  =      2,234
```

	Unconditional					
	Margin	Std. Err.	z	P> z	[95% Conf. Interval]	
ATET college (yes vs no)	5.144136	.1656339	31.06	0.000	4.819499	5.468772

We estimate that the average wage for those who graduated from college is \$5.14 higher than it would have been had those same individuals not graduated from college. This is \$2.49 less than the result from our model in [\[ERM\] example 2b](#) that did not account for the endogeneity of college graduation. We said “same individuals” to emphasize that \$5.14 is a treatment effect on those who chose to attend college and graduated. More formally, it is our estimate of what the average increase in wage is in the whole population for everyone who chose to attend college and graduated.

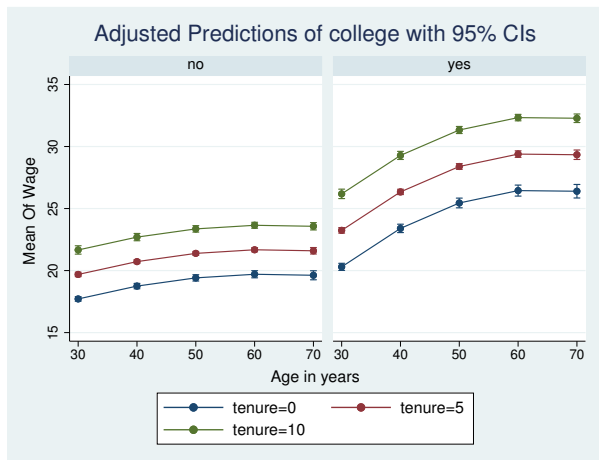
Is this effect constant for everyone? Let’s approach that question by first profiling expected wages for some representative values of age and tenure. We can ask `margins` to do that by typing

```
. margins college, predict(base(college=1)) vce(unconditional)
> at(age=(30(10)70) tenure=(0 5 10) peduc=2)
(output omitted)
```

We used the `at()` option to request values of age from 30 to 70 in units of 10 years and, for each of those ages, tenures of 0, 5, and 10. We also requested `college = 0` and `college = 1`, but we did that by typing `college` right after the `margins` command. We could have instead typed `college=(0 1)` in our `at()` option, but this is better. You will see that in a minute. We still want estimates for those who chose to go to college and graduated, so we specify `predict(base(college=1))`. That means we are further conditioning on the unobservable factors that increased the probability of graduating from college.

If you run the `margins` command, you will see that it takes a few seconds and that it produces a lot of output. Let's graph the results,

```
. marginsplot, by(college)
```



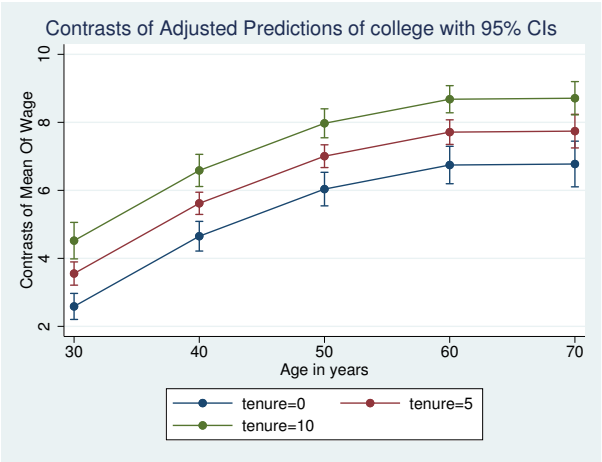
The first thing we notice is that these results are far too regular, and we should review our data collection process. That aside, the age–earnings profiles on the left, where we have taken the degrees away from our college graduates, are distinctly different from those on the right, where they get to retain their degrees. We see that tenure does have an effect, and if we look closely, it has a larger effect on college graduates: the profiles are further apart on the right. What do the points on this graph represent? Each point in the panel on the right is the expected wage for someone who graduated from college, whose parents graduated from college, and who has the age and tenure shown on the graph. Each point on the left is a counterfactual where we assume those same people did not graduate from college but where we continue to condition on the fact that their endogenous choice was to attend and complete college.

Seeing that, we have to ask, what are the profiles of the effect of college? To find those, we just add an `r.` to `college` on our previous `margins` command. Now you know why we specified `college` the way we did.

```
. margins r.college, predict(base(college=1)) vce(unconditional)
> at(age=(30(10)70) tenure=(0 5 10) peduc=2)
(output omitted)
```

Again, the output is long, so we graph the results.

```
. marginsplot
```



College affects wages the least when people are young and have no tenure. The largest effects are seen for those older than 50 and even more so when they also have long tenure. Each point represents the expected increase in wages due to graduating from college among those who chose to attend college and graduated. So each is an average treatment effect on the treated (ATET). Unlike overall average ATETs, these are conditioned on being at a specific age and having a specific tenure. Each point is bracketed by a pointwise 95% confidence interval. The confidence intervals reveal that we have pretty tight estimates for each of the ATETs. Note that the previous graph also displayed 95% confidence intervals. They were just so narrow that they are difficult to see.

Some might quibble with the “A” we just used in ATET because we have specified values for every covariate. Even so, taking the expectation over the errors in the model is a form of averaging. If you prefer call them the expected TETs (treatment effects on the treated).

We have focused on treatment effects on the treated, those who graduated from college. We could have just as easily asked about treatment effects on the untreated, those who did not graduate from college. What would we expect wages to do if they did graduate from college? Maybe we could reduce the cost of admission or otherwise affect their decision or institute mandatory college attendance. It is a minor change to what we have already typed. In each case, just change

```
predict(base(college=1))
```

to

```
predict(base(college=0))
```

If you do that, you will be conditioning on a decision not to attend college or a failure to complete college. If you make this change and reproduce the first graph, you will find that even after one graduates from college, wages are expected to be a little lower for this group. Recall that the unobserved factors that affected choosing to attend college were positively correlated with wages. If you run the contrasts to obtain the ATEU (average treatment effects on the untreated), you will find that those effects are identical to the ATETs! That is because our model is linear and because we are not averaging over the observations. The effect of the unobserved factors is different for college graduates and nongraduates. However, that effect is subtracted out when the counterfactuals are differenced to estimate the effect of college.

We gave parents' education short shrift in our analysis, locking it at the single value representing undergraduate degree. You can easily explore how differing levels of parents' education affect the results. Try typing

```
margins college, predict(base(college=1)) vce(unconditional)    ///
      at(age=36 tenure=10 peduc=(1 2 3 4))
```

You will find that parents' education does affect expected wages through the correlation between our two equations.

As is often the case with models having complications, estimation is just the first step.

See *Treatment* under *Methods and formulas* in [ERM] [eregress](#) and *Estimating treatment effects with margins* in [R] [margins](#), [contrast](#) for additional information about calculating the ATET.

## Also see

[ERM] [eregress](#) — Extended linear regression

[ERM] [eregress postestimation](#) — Postestimation tools for eregress

[ERM] [estat teffects](#) — Average treatment effects for extended regression models

[ERM] [intro 8](#) — Conceptual introduction via worked example

example 3a — Probit regression with continuous endogenous covariate

DescriptionRemarks and examplesAlso see

## Description

In this example, we show how to estimate and interpret the results of an extended regression model with a binary outcome and continuous endogenous covariate.

## Remarks and examples

In [ERM] [example 1a](#) through [ERM] [example 1c](#), we showed how researchers at the fictional State University might approach an investigation of the relationship between the high school grade point average (GPA) of the students the university admits and their final college GPA. Suppose instead that they would like to know how the probability of college graduation is related to high school grade point average (GPA). They again suspect that high school GPA is endogenous in a model of the probability of college graduation.

Their model for graduation includes parental income in \$10,000s and whether the student had a roommate who also went to State U. The State U researchers expect that the effect of high school competitiveness on the probability of graduating from college is negligible once the other covariates are controlled for. So they use the ranking of the high school (`hscomp`) as the instrumental variable for high school GPA. They also include parental income in the auxiliary model for high school GPA.

We want to make inferences about how our covariates affect graduation rates in the population, not just in our sample. We add `vce(robust)` so that subsequent calls to `estat teffects` and `margins` will be able to consider our sample as a draw from the population.

```
. use http://www.stata-press.com/data/r15/class10
(Class of 2010 profile)
. eprobit graduate income i.roommate, endogenous(hsgpa = income i.hscomp)
> vce(robust)

Iteration 0:  log pseudolikelihood = -1418.5008
Iteration 1:  log pseudolikelihood = -1418.4414
Iteration 2:  log pseudolikelihood = -1418.4414

Extended probit regression               Number of obs   =      2,500
                                         Wald chi2(3)    =      326.79
Log pseudolikelihood = -1418.4414       Prob > chi2     =      0.0000
```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
<b>graduate</b>						
income	.1597677	.0158826	10.06	0.000	.1286384	.1908969
roommate						
yes	.2636312	.0563563	4.68	0.000	.1531748	.3740876
hsgpa	1.01877	.4324788	2.36	0.018	.1711273	1.866413
_cons	-3.647166	1.204728	-3.03	0.002	-6.008389	-1.285943
<b>hsgpa</b>						
income	.047859	.0016461	29.07	0.000	.0446327	.0510853
hscomp						
moderate	-.135734	.0114717	-11.83	0.000	-.158218	-.1132499
high	-.225314	.0195055	-11.55	0.000	-.2635441	-.1870838
_cons	2.794711	.0127943	218.43	0.000	2.769634	2.819787
var(e.hsgpa)	.0685893	.0019597			.064854	.0725398
corr(e.hsgpa, e.graduate)	.3687006	.0919048	4.01	0.000	.1765785	.5337596

The estimate of the correlation between the errors of our two equations is 0.37 and is significantly different from zero, so we have endogeneity. Because the correlation is positive, we conclude that the unobservable factors that increase high school GPA also increase the probability of graduation.

The results for the main equation are interpreted as you would those from `probit`. We can obtain directions but not effect sizes from the coefficients in the main equation. For example, we see that family income and high school GPA are positively associated with the probability that a student graduates.

Let's ask something more interesting. What if we could increase each student's high school GPA by one point, moving a 2.0 to a 3.0, a 2.5 to a 3.5, and so on? We obviously cannot increase anyone's GPA by one point if he or she is already above a 3.0; so we restrict our population of interest to students with a GPA at or below 3.0. `margins` will give us the population-average expected graduation rate given each student's current GPA if we specify `at(hsgpa=generate(hsgpa))`. It will also give us the population-average expected graduation rate with an additional point in each student's GPA if we specify `at(hsgpa=generate(hsgpa+1))`. We want to hold each student's unobservable characteristics to be those that are implied by their current data, so we also create a variable holding the true values of `hsgpa` and specify `predict(base(hsgpa=hsgpaT))`.



```
. generate hsgpaT = hsgpa // True value of GPA for margins
. margins, at(hsgpa=generate(hsgpa)) at(hsgpa=generate(hsgpa+1))
> predict(base(hsgpa=hsgpaT)) subpop(if hsgpa <= 3) vce(unconditional)
Predictive margins                                Number of obs    =      2,500
                                                Subpop. no. obs    =      1,430

Expression   : Pr(graduate==yes), predict(base(hsgpa=hsgpaT))
1._at        : hsgpa                = hsgpa
2._at        : hsgpa                = hsgpa+1
```

	Unconditional					
	Margin	Std. Err.	z	P> z	[95% Conf. Interval]	
_at						
1	.4315243	.0214675	20.10	0.000	.3894487	.4735998
2	.7737483	.0953191	8.12	0.000	.5869264	.9605702

For students with a high school GPA at or below 3.0, the expected graduation rate is 43%. If those same students are given an additional point in their GPA, the graduation rate rises to 77%.

By adding `contrast(at(r))` to our `margins` command, we can difference those two counterfactuals and estimate the average effect of giving an additional point of GPA. We also added `effects` to add test statistics and `nowald` to clean up the output.

```
. margins, at(hsgpa=generate(hsgpa)) at(hsgpa=generate(hsgpa+1))
> subpop(if hsgpa <= 3) predict(base(hsgpa=hsgpaT))
> contrast(at(r) nowald effects) vce(unconditional)
Contrasts of predictive margins
Expression   : Pr(graduate==yes), predict(base(hsgpa=hsgpaT))
1._at        : hsgpa                = hsgpa
2._at        : hsgpa                = hsgpa+1
```

	Unconditional					
	Contrast	Std. Err.	z	P> z	[95% Conf. Interval]	
_at (2 vs 1)	.342224	.113214	3.02	0.003	.1203287	.5641194

Giving students an additional point in their GPA increased graduation rates by just over 34%, with a 95% confidence interval from 12% to 56%.

Does this effect differ across any of our other covariates? Our dataset has a grouping variable for family income `incomegrp`, so let's estimate the effect within each income grouping. We just add `over(incomegrp)` to our prior `margins` command.

```
. margins, at(hsgpa=generate(hsgpa)) at(hsgpa=generate(hsgpa+1))
> subpop(if hsgpa <= 3) predict(base(hsgpa=hsgpaT))
> contrast(at(r) nowald effects) noatlegend vce(unconditional) over(incomegrp)

Contrasts of predictive margins

Expression   : Pr(graduate==yes), predict(base(hsgpa=hsgpaT))
over         : incomegrp
```

	Unconditional		z	P> z	[95% Conf. Interval]	
	Contrast	Std. Err.				
_at@						
incomegrp						
(2 vs 1)						
< 20K	.3690987	.1359989	2.71	0.007	.1025457	.6356516
(2 vs 1)						
20-39K	.3698609	.1273853	2.90	0.004	.1201903	.6195316
(2 vs 1)						
40-59K	.3516159	.1103376	3.19	0.001	.1353581	.5678737
(2 vs 1)						
60-79K	.3094611	.0927492	3.34	0.001	.1276761	.4912461
(2 vs 1)						
80-99K	.255203	.0748521	3.41	0.001	.1084956	.4019105
(2 vs 1)						
100-119K	.1829494	.0552683	3.31	0.001	.0746256	.2912732
(2 vs 1)						
120-139K	.1238028	.0459416	2.69	0.007	.0337588	.2138467
(2 vs 1)						
140K up	.0485429	.0207233	2.34	0.019	.0079259	.0891598

The effect is largest for the low-income groups and declines as income goes up. It becomes almost negligible for students from households whose income is above \$140,000.

We can see this relationship more clearly if we graph the results.

```
. marginsplot
```

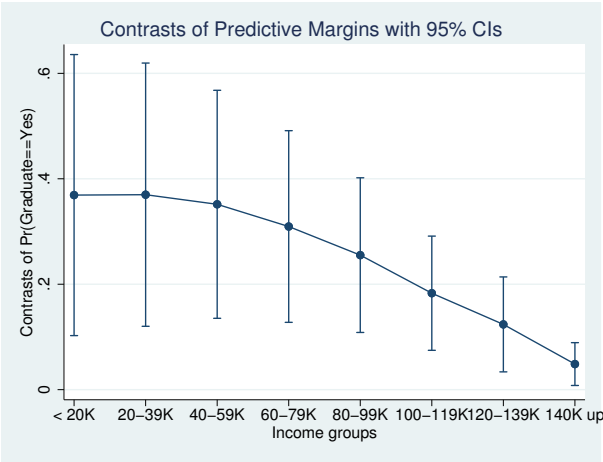


Figure 1.

Our point estimates of the effect on the probability of graduating are near 0.4 for the lowest-income groups and fall below 0.2 for incomes over \$100,000.

So we can examine subpopulation averages and effects and make inferences about their values.

We can also examine averages and effects at specified values of the covariates in our model. Let's consider students who do not have roommates and evaluate them at 5 levels of high school GPA (2.0, 2.5, 3.0, 3.5, and 4.0) and at two levels of income (\$30,000 and \$110,000).

```
. margins, at(roommate=0 hsgpa=(2 2.5 3 3.5 4) income=(3 11)) noatlegend
Predictive margins                                Number of obs      =      2,500
Model VCE      : Robust
Expression     : Pr(graduate==yes), predict()
```

	Delta-method		z	P> z	[95% Conf. Interval]	
	Margin	Std. Err.				
_at						
1	.0068488	.0076828	0.89	0.373	-.0082092	.0219068
2	.1215437	.0353464	3.44	0.001	.052266	.1908213
3	.5517785	.0320675	17.21	0.000	.4889272	.6146297
4	.9232607	.043002	21.47	0.000	.8389784	1.007543
5	.9967789	.0051452	193.73	0.000	.9866944	1.006863
6	.0470211	.0496759	0.95	0.344	-.0503419	.144384
7	.3531365	.1042001	3.39	0.001	.1489081	.5573649
8	.8213242	.023535	34.90	0.000	.7751964	.867452
9	.9867056	.0071801	137.42	0.000	.9726328	1.000778
10	.9997797	.0003587	2787.22	0.000	.9990767	1.000483

Looking at all combinations of GPA and income, we see that graduation probabilities range from 0.0068 to 0.9998 for these values of the covariates.

We have suppressed the long legend that explains the \_at levels in the table, so let's explain the lines. All results are for students without roommates. Lines 1–5 are for students with family incomes of \$30,000 with the first line representing a GPA of 2, the second a GPA of 2.5, and so on. Lines 6–10

represent the same levels of GPA for students with a family income of \$110,000. Because our model has only three covariates in the main equation and because we have specified values for each of the covariates, these can be considered fully conditional estimates. Even so, they are averages in the sense that they are expected values. Each probability represents what we would expect if hundreds of students were sampled who had the same values of the covariates as those on the corresponding line.

The patterns in these results are easier to see on a graph.

```
. marginsplot
```

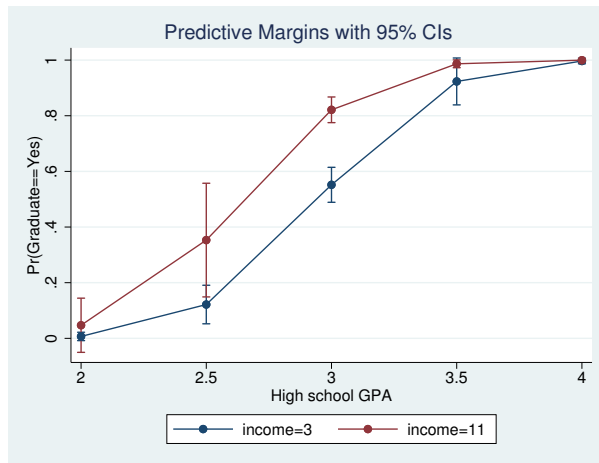


Figure 2.

Students with a GPA of 2.0 have nearly no chance of graduating, regardless of income. For those with a GPA between 2.5 and 3.0, the graduation rates differ sharply depending on income level. At GPAs of 3.5 and above, graduation rates are so high that there is again little difference due to income. These results aren't surprising; it's easier to struggle through school when you do not also have to worry over money issues.

What if we could grant the lower-income students a higher income? We would want to hold their unobservables at their initial level while moving them to the higher income. Perhaps they are adopted. Perhaps we are using this increase in income as a proxy for providing financial aid to lower-income students. Regardless, we use `predict(base(income=3))` to hold their unobservable characteristics to their initial level as we move income from \$30,000 to \$110,000.

```
. margins, at(roommate=0 hsgpa=(2 2.5 3 3.5 4) income=(3 11))
> noatlegend predict(base(income=3))
(output omitted)
```

We dispense with showing you the output and go straight to the graph. You can run the `margins` command if you wish.

```
. marginsplot
```

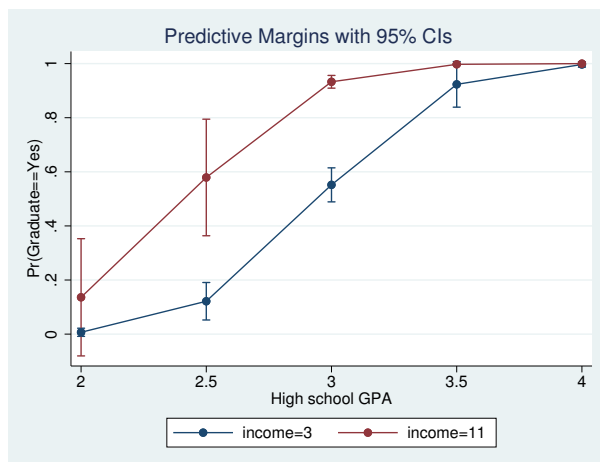


Figure 3.

The separation between graduation probabilities for incomes of \$30,000 and \$110,000 is even larger for those who obtain their high school GPA while in a family with \$30,000 income and are then moved to \$110,000.

Let's explore that a bit more, not because made-up data are interesting but because we have yet more tools to show you. `margins` will compute contrasts (differences) between our `at()` groupings but is an all-or-none proposition. It is either all levels or all differences. We want to see the differences in the lines we have been drawing while keeping our levels of GPA. We are going to estimate and graph the differences between the lines on the graph we just drew and also on the graph we drew before that. So we are going to compare the effects for those born with higher incomes and the effects with those granted higher incomes at entry to college. The latter is a proper effect due to an exogenous change. The former is just a comparison of two groups. We type

```
. margins, at(roommate=0 hsgpa=(2 2.5 3 3.5 4) income=3)
> predict(target(income=3)) predict(target(income=11) base(income=11))
> predict(target(income=11)) contrast(predict(r) nowald effects) noatlegend
(output omitted)
```

Let's focus first on the syntax. The `predict(target())`s are new; see [ERM] [eoprobit predict](#) for a detailed explanation. Briefly, `target()` specifies a counterfactual value directly. So `predict(target(income=3))` specifies an income of \$30,000. Because that is the same value `margins` is specifying, that is more of a factual than a counterfactual. Well, it is a factual for low-income students and is shown as the blue line in [figure 2](#) and [figure 3](#).

`predict(target(income=11) base(income=11))` specifies that both the counterfactual income and the `base()` income from which the student's unobservable characteristics are obtained are \$110,000. So it too is a factual. It is a factual for high-income students and is shown as the red line in [figure 2](#). `predict(target(income=11))` specifies that our counterfactual income is 11, but because `margins` is setting the income to 3, the unobservable characteristics will be for a student whose parents earn \$30,000. This is the red line in [figure 3](#).

The results are

```
. margins, at(roommate=0 hsgpa=(2 2.5 3 3.5 4) income=3)
> predict(target(income=3)) predict(target(income=11) base(income=11))
> predict(target(income=11)) contrast(predict(r) nowald effects) noatlegend

Contrasts of predictive margins
Model VCE      : Robust
1._predict      : Pr(graduate==yes), predict(target(income=3))
2._predict      : Pr(graduate==yes), predict(target(income=11) base(income=11))
3._predict      : Pr(graduate==yes), predict(target(income=11))
```

	Delta-method					[95% Conf. Interval]
	Contrast	Std. Err.	z	P> z		
_predict@_at						
(2 vs 1) 1	.0401723	.0421568	0.95	0.341	-.0424536	.1227981
(2 vs 1) 2	.2315929	.0725988	3.19	0.001	.0893018	.3738839
(2 vs 1) 3	.2695457	.0440405	6.12	0.000	.1832279	.3558636
(2 vs 1) 4	.0634449	.036527	1.74	0.082	-.0081467	.1350365
(2 vs 1) 5	.0030008	.0047942	0.63	0.531	-.0063957	.0123973
(3 vs 1) 1	.1292645	.1030033	1.25	0.209	-.0726182	.3311473
(3 vs 1) 2	.4575187	.078341	5.84	0.000	.3039732	.6110642
(3 vs 1) 3	.3809832	.0367368	10.37	0.000	.3089803	.4529861
(3 vs 1) 4	.0741338	.0414526	1.79	0.074	-.0071118	.1553795
(3 vs 1) 5	.0031996	.0051059	0.63	0.531	-.0068078	.0132071

And their graph is

```
. marginsplot
```

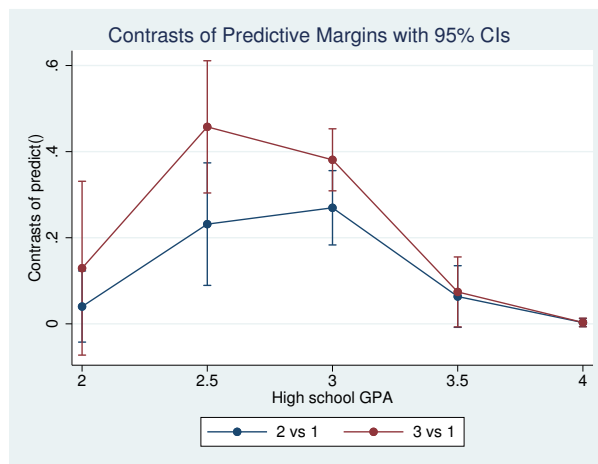


Figure 4.

The blue points and line represent the difference between a student from a family earning \$30,000 and a student from a family earning \$110,000. The red points and line represents the difference between the same student who started in a family earning \$30,000 but was granted \$110,000 family earnings on entry into college. The higher income means much more to those who achieved their GPA while in a lower-income family. This is particularly true for those with GPAs between 2.5 and 3.0.

Recall that our estimation results indicated a positive correlation between unobservable factors that increase a student's GPA and those that increase the probability that the student graduates. The

`margins` results above are driven by lower-income students having higher levels of these unobservable factors for any given level of high school GPA. In fact, the only thing that makes the two lines different is that the students who started with incomes of \$30,000 have different unobservable characteristics from those who started with incomes of \$110,000. All other covariates are the same. How important are those unobserved factors? We assess that directly by comparing our two counterfactuals that set income at \$110,000.

We delete the line `predict(target(income=3))` so that we are comparing the two counterfactuals against each other, rather than each against the counterfactual of \$30,000 family income.

```
. margins, at(roommate=0 hsgpa=(2 2.5 3 3.5 4) income=3)
> predict(target(income=11) base(income=11)) predict(target(income=11))
> contrast(predict(r) nowald effects) noatlegend
(output omitted)
. marginsplot
```

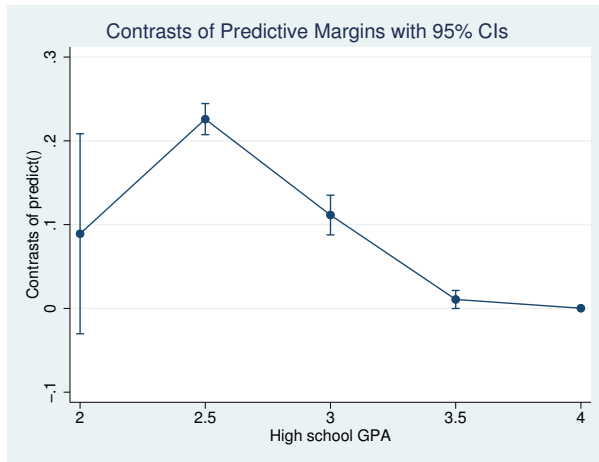


Figure 5.

These results directly measure the contribution of the student’s unobservable characteristics to graduation rates. At a GPA of 2.0, a student from a family earning \$30,000 and then being moved to a family income of \$110,000 would be 10 percentage points more likely to graduate than a student from a family who always earned \$110,000 .

That effect rises to over 20 percentage points if the student’s GPA is 2.5.

So we can also analyze fully conditional counterfactuals and make complex inferences.

## Also see

[ERM] [eprobit](#) — Extended probit regression

[ERM] [eprobit postestimation](#) — Postestimation tools for `eprobit`

[ERM] [intro 3](#) — Endogenous covariates features

[ERM] [intro 8](#) — Conceptual introduction via worked example

# Title

example 3b — Probit regression with endogenous covariate and treatment

DescriptionRemarks and examplesAlso see

## Description

We model a binary outcome that depends on a continuous endogenous covariate and has an endogenous treatment by using `eprobit` with the `endogenous()` and `entreat()` options.

## Remarks and examples

Continuing from [\[ERM\] example 3a](#), State U administrators have implemented a voluntary program to increase retention freshman year. Whether a student chose to participate is stored in the indicator variable `program`. They are concerned that unobservable factors that influence a student’s decision to participate in the college retention program also influence the probability of graduation. For example, students who have higher self-motivation may be more likely to join and also more likely to graduate without the program. Thus, they are concerned that participation in the program may be an endogenously chosen treatment.

The researchers believe the program was easier to access for students who lived on campus freshman year. They also think students who had scholarships may have been more motivated to attend the program. However, they do not believe either of these variables independently affects the probability of graduation after controlling for other covariates in the model. They use an indicator for on-campus residence during the freshman year (`campus`), having a scholarship of any kind (`scholar`), and parents’ income in the treatment assignment model.



```
. eprobit graduate income i.roommate, endogenous(hsgpa = income i.hscomp)
> entreat(program = i.campus i.scholar income) vce(robust)

Iteration 0:   log pseudolikelihood = -2793.4696
Iteration 1:   log pseudolikelihood = -2792.9298
Iteration 2:   log pseudolikelihood = -2792.9017
Iteration 3:   log pseudolikelihood = -2792.9016

Extended probit regression                                Number of obs      =      2,500
                                                         Wald chi2(8)       =      404.26
                                                         Prob > chi2        =      0.0000

Log pseudolikelihood = -2792.9016
```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
graduate						
program#						
c.income						
0	.1760785	.0201833	8.72	0.000	.13652	.2156371
1	.1925761	.021294	9.04	0.000	.1508405	.2343116
roommate#						
program						
yes#0	.3110885	.0814304	3.82	0.000	.1514878	.4706892
yes#1	.2475942	.0756877	3.27	0.001	.0992491	.3959394
program#						
c.hsgpa						
0	1.160053	.4590276	2.53	0.011	.2603759	2.059731
1	.9379774	.4450455	2.11	0.035	.0657043	1.81025
program						
0	-4.350156	1.312558	-3.31	0.001	-6.922721	-1.77759
1	-3.393398	1.242536	-2.73	0.006	-5.828725	-.9580717
program						
campus						
yes	.7433155	.0735249	10.11	0.000	.5992092	.8874217
scholar						
yes	.8970451	.0585469	15.32	0.000	.7822952	1.011795
income	-.0799274	.0088987	-8.98	0.000	-.0973686	-.0624862
_cons	-.3810042	.0860131	-4.43	0.000	-.5495867	-.2124217
hsgpa						
income	.0478626	.0016461	29.08	0.000	.0446363	.0510889
hscomp						
moderate	-.1350116	.0115013	-11.74	0.000	-.1575538	-.1124694
high	-.2269435	.019326	-11.74	0.000	-.2648218	-.1890652
_cons	2.79442	.0128088	218.16	0.000	2.769315	2.819525
var(e.hsgpa)	.0685874	.0019597			.064852	.0725379
corr(e.pro~m,						
e.graduate)	.3791651	.1035775	3.66	0.000	.1605878	.5622919
corr(e.hsgpa,						
e.graduate)	.4001679	.089854	4.45	0.000	.2109447	.5604834
corr(e.hsgpa,						
e.program)	-.0201748	.02637	-0.77	0.444	-.0717594	.0315174

The main equation output is slightly different from that in [ERM] example 3a. Because program was specified as a treatment, it was automatically interacted with each of the other covariates in the graduate equation.

The correlation between the errors from the graduation equation and those from the program participation equation is estimated to be 0.38 and is significantly different from zero. The researchers conclude that unobservable factors that increase the chance of participating in the program also increase the chance of graduating.

Now, we use `estat teffects` to estimate the ATE of program participation on college graduation. We specified `vce(robust)` when we fit the model, so `estat teffects` reports standard errors and tests for the population ATE.

. estat teffects						
Predictive margins			Number of obs		=	2,500
	Unconditional					
	Margin	Std. Err.	z	P> z	[95% Conf. Interval]	
ATE						
program						
(1 vs 0)	.1017846	.0488905	2.08	0.037	.005961	.1976081

We estimate that the ATE is 0.10. In other words, the average probability of graduating increases by 0.10 when all students participate in the program versus when no students participate in the program.

We might be interested if those students who self-selected into the program increased their graduation probability by more than 0.10. We estimate the average treatment effect on the treated (ATET).

```
. estat teffects, atet
```

Predictive margins						Number of obs	=	2,500
						Subpop. no. obs	=	1,352

---

	Unconditional					[95% Conf. Interval]	
	Margin	Std. Err.	z	P> z			
ATET program (1 vs 0)	.1015967	.0493961	2.06	0.040	.0047821	.1984113	

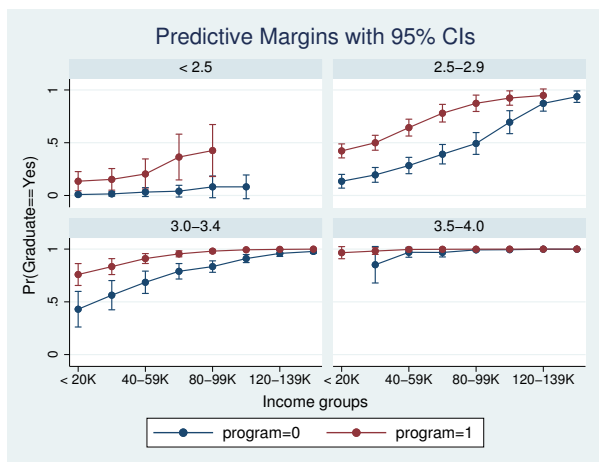
In this case, the students who chose the program did no better on average than choosing by flipping a coin. Both the ATE and ATET are 0.10.

Those are the overall averages. Do graduation rates for participants and nonparticipants differ by high school GPA and parents' income? Our dataset has grouping variables, so we can let `margins` estimate graduation rates subpopulations defined by all three covariates.

```
. margins, over(program incomegrp hsgpagrp) vce(unconditional)
```

The output is copious. You can type the command and see it if you like. The patterns are easier to see on a `marginsplot`.

```
. marginsplot, plot(program) xlabel(0 4 8 12)
```



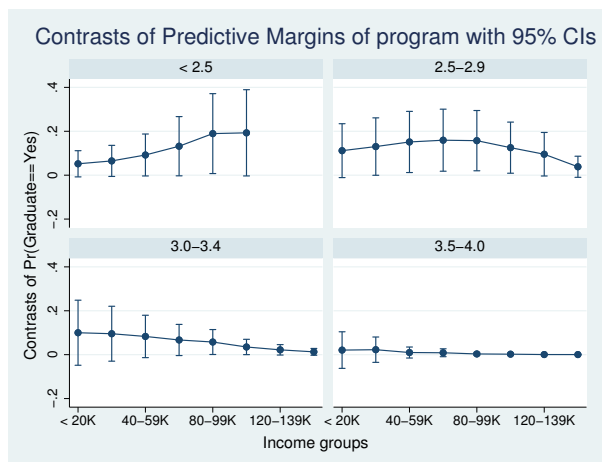
The red line shows expected graduation rates for those who participated in the program. The blue line shows rates for nonparticipants. Clearly, the differences between the groups in the program and those out of the program differ dramatically across GPA and family income. For GPAs at or above 3.5, the graduation rates are so high that there was no room for differences. For most other groups, the graduation rates are estimated to be substantially higher among those who participated. The only exceptions are extremely low-income students with GPAs below 2.5 and extremely high-income students with GPAs at or above 3.5.

We were careful not to call the comparisons above effects or attribute them directly to the program. They are indeed expected rates for the groups, but the students self-selected into program participation groups. If we want to compare graduation rates assuming all students don't participate and then assuming all students do participate, we need to instruct `margins` to `fix()` the values for program participation and also add the `r.` to `program`.

```
. margins r.program, over(incomegrp hsgpagrp) vce(unconditional)
> predict(fix(program)) contrast(nowald)
(output omitted)
```

The output is again long, so we leave you to see it for yourself. The graphs reveal the patterns across groups.

```
. marginsplot, by(hsgrp) xlabel(0 4 8 12)
```



These differences are close to what we would have seen had we differenced the red and blue lines of the first graph. In this graph, each point is an estimate of the average treatment effect for a subpopulation defined by a range of GPAs and a range of family income. We note that the confidence intervals, as represented by the capped lines, are fairly wide.

## Also see

[ERM] [eprobit](#) — Extended probit regression

[ERM] [eprobit postestimation](#) — Postestimation tools for `eprobit`

[ERM] [estat teffects](#) — Average treatment effects for extended regression models

[ERM] [intro 3](#) — Endogenous covariates features

[ERM] [intro 5](#) — Treatment assignment features

[ERM] [intro 8](#) — Conceptual introduction via worked example

example 4a — Probit regression with endogenous sample selection

DescriptionRemarks and examplesAlso see

Description

In this example, we show how to estimate and interpret the results of an extended regression model with a binary outcome and endogenous sample selection.

Remarks and examples

We are interested in whether regular exercise and body mass index (BMI) influence the chance of having a subsequent heart attack. In our fictional study, we collected data on 625 men who had a heart attack when they were between the ages of 50 and 55. Some men withdrew from the study before it completed, and we believe their reasons for leaving are related to unobserved factors that also affect their chances of having a second heart attack. We did, however, observe all cases where a second heart attack was fatal.

To account for the endogenous sample selection, we specify an auxiliary model for selection using a covariate that belongs in the auxiliary model and is excluded from the main equation. We expect that the direct effect of whether a man had regular checkups before the study is negligible after we condition on other covariates.

The outcome of interest is whether the man had another heart attack within five years of his first heart attack (`attack`). We believe that the man’s current age is also an important exogenous covariate along with BMI. We model the indicator for whether the man was observed for the full five years of the study (`full`) as a function of an indicator for having regular checkups along with the covariates from the main equation.

```
. use http://www.stata-press.com/data/r15/heartsm
(Heart attacks)

. eprobit attack age bmi i.exercise, select(full = age bmi i.checkup) vce(robust)

Iteration 0:   log pseudolikelihood = -409.23137
Iteration 1:   log pseudolikelihood = -408.78569
Iteration 2:   log pseudolikelihood = -408.78452
Iteration 3:   log pseudolikelihood = -408.78452

Extended probit regression               Number of obs   =       625
                                         Selected        =       458
                                         Nonselected     =       167

                                         Wald chi2(3)    =      142.85
Log pseudolikelihood = -408.78452       Prob > chi2      =      0.0000
```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
<b>attack</b>						
age	.2237091	.0351334	6.37	0.000	.1548489	.2925693
bmi	.1760896	.0298853	5.89	0.000	.1175155	.2346636
exercise						
yes	-1.438937	.1515198	-9.50	0.000	-1.735911	-1.141964
_cons	-15.78445	2.105945	-7.50	0.000	-19.91202	-11.65687
<b>full</b>						
age	-.1599347	.032953	-4.85	0.000	-.2245214	-.095348
bmi	-.1146582	.0208896	-5.49	0.000	-.1556011	-.0737152
checkup						
yes	2.306638	.1660248	13.89	0.000	1.981236	2.632041
_cons	11.66488	1.942686	6.00	0.000	7.857284	15.47247
<b>corr(e.full, e.attack)</b>	-.4537026	.1636665	-2.77	0.006	-.71301	-.0852183

We estimate that the correlation between the errors from the outcome equation and the errors from the selection equation is  $-0.45$ . This is significantly different from zero, so selection into the study is endogenous. Because the correlation is negative, we conclude that unobserved factors that increase the chance of staying in the study tend to occur with unobserved factors that decrease the chance of having a subsequent heart attack.

The results for the main outcome equation (`attack`) and auxiliary selection equation (`full`) are interpreted just as you would those from `heckprobit`. Which is to also say that the results for the main equation can be interpreted as you would those from a probit regression using `probit` on uncensored data. The goal of including a selection model is to estimate the parameters of the main equation as though there were no selection.

Age and BMI have increased the chances of having another heart attack, while regular exercise decreases the chances. However, the magnitude of the effect on the probability of another heart attack cannot be determined from the coefficient estimates themselves. We can use `margins` to examine the effect of different covariates on the probability of having a second heart attack. But first we want to investigate a possible further complication in our data: regular exercise may be an endogenous treatment. We explore this in [\[ERM\] example 4b](#).

## Also see

[ERM] [eprobit](#) — Extended probit regression

[ERM] [eprobit postestimation](#) — Postestimation tools for eprobit

[ERM] [intro 4](#) — Endogenous sample-selection features

[ERM] [intro 8](#) — Conceptual introduction via worked example

example 4b — Probit regression with endogenous treatment and sample selection

DescriptionRemarks and examplesAlso see

Description

Continuing from [ERM] example 4a, we show you how to estimate and interpret the results of a model for a binary outcome when the model includes an endogenous treatment and the data are subject to endogenous sample selection.

Remarks and examples

In [ERM] example 4a, we ignored the possibility that regular exercise was an endogenous treatment. However, we suspect that unobserved factors that influence the choice to exercise may be correlated with the unobserved factors that affect the chance of having another heart attack.

We would like to know the average expected change in probability of having a subsequent heart attack for those who exercise. That is, we are interested in estimating the average treatment effect on the treated (ATET). We continue to include BMI and age in our outcome model, and to account for endogenous sample selection, we specify the same auxiliary model for selection we did in [ERM] example 4a. We add a third equation to account for endogenous treatment assignment. Whether a man ever joined a gym is an instrumental variable predicting exercise that we do not expect to otherwise affect `attack`, so we include it in our model for regular exercise.



```
. eprobit attack age bmi, select(full = age bmi i.checkup)
> entreat(exercise = bmi i.gym) vce(robust)
(iteration log omitted)
Extended probit regression                                Number of obs    =      625
                                                         Selected       =      458
                                                         Nonselected    =      167
                                                         Wald chi2(6)   =     111.78
                                                         Prob > chi2    =      0.0000
Log pseudolikelihood = -711.90507
```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
attack						
exercise#						
c.age						
no	.2156634	.0550909	3.91	0.000	.1076872	.3236397
yes	.221641	.0423742	5.23	0.000	.1385891	.3046928
exercise#						
c.bmi						
no	.1925833	.04278	4.50	0.000	.108736	.2764306
yes	.2134441	.038381	5.56	0.000	.1382186	.2886696
exercise						
no	-16.07086	3.282712	-4.90	0.000	-22.50486	-9.636863
yes	-17.84655	2.61864	-6.82	0.000	-22.97899	-12.71411
full						
age	-.1650386	.0321825	-5.13	0.000	-.228115	-.1019621
bmi	-.1143184	.0206726	-5.53	0.000	-.154836	-.0738008
checkup						
yes	2.315167	.1639928	14.12	0.000	1.993747	2.636587
_cons	11.92957	1.898426	6.28	0.000	8.208727	15.65042
exercise						
bmi	-.1815549	.0211349	-8.59	0.000	-.2229786	-.1401313
gym						
yes	1.517225	.1248316	12.15	0.000	1.27256	1.761891
_cons	3.941703	.5728064	6.88	0.000	2.819023	5.064383
corr(e.full, e.attack)	-.5338178	.1584217	-3.37	0.001	-.7737932	-.1598432
corr(e.exe~e, e.attack)	-.435728	.1467897	-2.97	0.003	-.676196	-.1113554
corr(e.exe~e, e.full)	.3212358	.0928654	3.46	0.001	.1293396	.4899396

The correlation between the errors that affect having a subsequent heart attack and the errors that affect staying in the study is estimated to be  $-0.53$  and is significant. So we do have endogenous selection and conclude that unobservable factors that increase the chance of staying in the study also tend to decrease the chance of having a subsequent heart attack.

Increases in age and BMI increase the chance of having another heart attack. This is true both for those who exercise, coefficients marked *yes*, and for those who do not, coefficients marked *no*.

We use `estat teffects` to estimate the ATET of regular exercise on having a subsequent heart attack. We specified `vce(robust)` when we fit the model so that `estat teffects` will report unconditional standard errors for the population ATET rather than the sample ATET.

```
. estat teffects, atet
Predictive margins                Number of obs   =       625
                                Subpop. no. obs  =       291
```

	Unconditional		z	P> z	[95% Conf. Interval]	
	Margin	Std. Err.				
ATET exercise (yes vs no)	-.2993399	.0840334	-3.56	0.000	-.4640424	-.1346374

The estimated ATET is  $-0.30$ . Thus, for those who exercise regularly, the average probability of having a subsequent heart attack is 0.30 lower than it would be if they did not exercise regularly.

Also see

- [ERM] [eprobit](#) — Extended probit regression
- [ERM] [eprobit postestimation](#) — Postestimation tools for `eprobit`
- [ERM] [estat teffects](#) — Average treatment effects for extended regression models
- [ERM] [intro 4](#) — Endogenous sample-selection features
- [ERM] [intro 5](#) — Treatment assignment features
- [ERM] [intro 8](#) — Conceptual introduction via worked example

# Title

example 5 — Probit regression with endogenous ordinal treatment

DescriptionRemarks and examplesAlso see

## Description

We model a binary outcome that depends on an endogenous ordinal treatment by using `eprobit` with the `entreat()` option.

## Remarks and examples

We are interested in estimating the average treatment effects (ATEs) of different levels of exercise intensity on the chance of having a subsequent heart attack. In our fictional study, we collected data on 625 men who had a heart attack when they were between the ages of 50 and 55. The outcome of interest is whether the man had another heart attack within five years of his first heart attack (`attack`). We believe that body mass index (BMI) and age are important covariates.

The `exintensity` variable records the intensity of exercise using the scale of 0 (no exercise), 1 (moderate), and 2 (heavy). We suspect that unobserved factors that influence the choice to exercise at a certain intensity level also affect the chance of having another heart attack, so we specify `exintensity` as an endogenous treatment. Whether an individual ever joined a gym is included as an instrumental covariate in the treatment model that we specify in `entreat()`.

```
. use http://www.stata-press.com/data/r15/heartsm
(Heart attacks)
. eprobit attack age bmi, entreat(exintensity = bmi i.gym) vce(robust)
(iteration log omitted)
Extended probit regression               Number of obs      =       625
                                         Wald chi2(9)        =      152.33
Log pseudolikelihood = -728.6686        Prob > chi2          =       0.0000
```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
attack						
exintensity#						
c.age						
none	.2118759	.0514612	4.12	0.000	.1110138	.312738
moderate	.2338466	.0425341	5.50	0.000	.1504813	.3172119
heavy	.2346887	.0805152	2.91	0.004	.0768818	.3924957
exintensity#						
c.bmi						
none	.1948171	.0386314	5.04	0.000	.119101	.2705332
moderate	.2062276	.0405785	5.08	0.000	.1266952	.2857599
heavy	.2155222	.0765592	2.82	0.005	.0654689	.3655755
exintensity						
none	-15.90911	3.043587	-5.23	0.000	-21.87444	-9.943793
moderate	-18.2922	2.499325	-7.32	0.000	-23.19079	-13.39362
heavy	-18.61821	5.395246	-3.45	0.001	-29.1927	-8.043721
exintensity						
bmi	-.1720462	.0204172	-8.43	0.000	-.2120632	-.1320292
gym						
yes	1.518834	.1192361	12.74	0.000	1.285136	1.752532
/exintensity						
cut1	-3.677846	.5537938			-4.763262	-2.59243
cut2	-2.386538	.5372719			-3.439572	-1.333505
corr(e.exi~y, e.attack)	-.4722803	.1091789	-4.33	0.000	-.6575129	-.2332112

The estimated correlation between the errors in the main outcome and auxiliary treatment equations is  $-0.47$ . This is significantly different from zero, so we confirm that the choice of exercise intensity level is endogenous. Because it is negative, we conclude that unobservable factors that increase the intensity of exercising tend to decrease the chance of having a subsequent heart attack. The cutpoints for the ordered probit model for the endogenous treatment are shown just beneath the treatment model.

The coefficients for `exintensity` in the main equation indicate that both moderate and heavy exercise have a negative effect because they are smaller, more negative, than the coefficient for no exercise. BMI has a positive effect on the chance of having another heart attack, regardless of exercise level. In fact, the values of the three coefficients for `bmi` are so close that we might not need separate parameters for the three levels of exercise. The same could be said of the three coefficients on `age`.

The coefficients for the intercepts of heavy and moderate exercise are close in magnitude. To test whether these two coefficients are equal, we can use `test`.

```
. test 1.exintensity == 2.exintensity
( 1)  [attack]1.exintensity - [attack]2.exintensity = 0
      chi2( 1) =      0.00
      Prob > chi2 =      0.9557
```

We cannot reject that the coefficients are equal.

We also have separate coefficients on `age` and `bmi` for heavy and moderate exercise. To jointly test the equality of each coefficient associated with heavy exercise with the corresponding coefficient associated with moderate exercise, we type

```
. test (1.exintensity == 2.exintensity)
>      (1.exintensity#c.bmi == 2.exintensity#c.bmi)
>      (1.exintensity#c.age == 2.exintensity#c.age)
( 1)  [attack]1.exintensity - [attack]2.exintensity = 0
( 2)  [attack]1.exintensity#c.bmi - [attack]2.exintensity#c.bmi = 0
( 3)  [attack]1.exintensity#c.age - [attack]2.exintensity#c.age = 0
      chi2( 3) =      0.04
      Prob > chi2 =      0.9983
```

We do not have any evidence that heavy and moderate exercise have a different effect on the probability of a second heart attack.

That was some pretty tricky coefficient referencing in our `test` command. We suggest you type

```
. erprobit, coeflegend
```

to see how to reference coefficients in `test`, `nlcom`, and other postestimation commands.

What if every man in the population did not exercise? What if they all exercised moderately? What if they all exercised heavily? `estat teffects` can estimate the average probability of a second heart attack over the five years for each of those counterfactuals.

. estat teffects, pomean

Predictive margins					Number of obs	=	625
	Unconditional						
	Margin	Std. Err.	z	P> z	[95% Conf. Interval]		
P0mean							
exintensity							
none	.7918941	.0329342	24.04	0.000	.7273443	.856444	
moderate	.5419335	.0326336	16.61	0.000	.4779728	.6058942	
heavy	.5336232	.0767752	6.95	0.000	.3831466	.6840998	

When no one in the population exercises, we estimate that 79% will have subsequent heart attacks. We are pretty confident in that number: the 95% confidence interval begins at 73% and ends at 86%. It does not matter much whether every man exercises moderately or heavily. Either intensity drops the expected rate of subsequent heart attacks to about 54%. These are the average potential-outcome means (POMs) under the three exercise-intensity regimes.

The difference between these POMs gives us estimates of the average treatment effects (ATEs) in the population. `estat teffects` will estimate those too.

```
. estat teffects
```

Predictive margins			Number of obs		=	625
	Margin	Unconditional Std. Err.	z	P> z	[95% Conf. Interval]	
ATE exintensity (moderate vs none) (heavy vs none)	-.2499606	.0507776	-4.92	0.000	-.349483	-.1504383
	-.2582709	.0965797	-2.67	0.007	-.4475637	-.0689781

We estimate that the ATE for heavy intensity compared with no exercise is  $-0.26$ . So the average probability of a subsequent heart attack is 26 percentage points lower when all men in the population exercise with heavy intensity versus when none of them exercise at all. The estimated ATE for moderate intensity versus none is  $-0.25$ . We again see no substantive difference between moderate and heavy exercise.

We used `vce(robust)` at estimation so that `estat teffects` would report standard errors that account for sampling variability in our covariates and are therefore valid for inference about the POMs, ATEs, and ATETs in the population from which our sample was drawn.

We have established that men who choose to exercise have unobserved attributes that tend to decrease their chance of another heart attack beyond the direct effect of exercising and beyond the effect of the other covariates. We can include the effect of these attributes for men who exercise by estimating the average treatment effect on the treated (ATET).

```
. estat teffects, atet
(subpopulation of first non-control treatment level assumed)
```

Predictive margins			Number of obs		=	625
			Subpop. no. obs		=	201
	Margin	Unconditional Std. Err.	z	P> z	[95% Conf. Interval]	
ATET exintensity (moderate vs none) (heavy vs none)	-.2992132	.0592607	-5.05	0.000	-.4153619	-.1830644
	-.309572	.1077129	-2.87	0.004	-.5206854	-.0984586

The ATETs are both about 0.30, making them about 5 percentage points higher than the ATEs. We cannot, however, directly attribute that difference to the unobserved attributes. The ATETs are also averaged over subsamples and are therefore affected by any differences in the distribution of `age` or `bmi` in treated subsamples. The effect of those distributions could be either positive or negative.

With some care, we can extract just the effect of the unobserved attributes. It is a little tricky, both conceptually and syntactically. So continue reading only if you are truly interested.

Let’s consider only the moderate exercisers. When we type

```
. margins r(0 1).exintensity, subpop(if exintensity == 1)
```

margins will produce the average difference for exintensity levels 0 and 1 (none and moderate). subpop(if exintensity == 1) restricts the average to men who exercised moderately. If we were to add

```
. margins r(0 1).exintensity, subpop(if exintensity == 1) ///  
      predict(base(exintensity=1))
```

margins would use the unobserved attributes associated with moderate exercise for both of the counterfactuals it requires to compute the contrast. Which is to say, it would use the true value of exercise intensity in the subpopulation we are averaging over. If you were to guess that this difference will be the ATET, you would be correct. For each man who chose moderate exercise, the ATET computation compares the man’s expected probability of another attack using all the information on the man with that same man’s expected probability if he instead did not choose to exercise. When we say “same man”, we mean that he retains his original unobserved attributes when evaluating the counterfactual that he does not exercise. The ATET is then the average of that comparison over all those who exercise moderately.

If we pretend that same man did not exercise, then we could obtain the unobserved attributes for someone just like him who does not exercise. We tell margins to do that for each man by adding

```
. margins r(0 1).exintensity, subpop(if exintensity == 1)      ///  
      predict(base(exintensity=1)) predict(base(exintensity=0))
```

That last predict() says to base both counterfactuals on each man’s observed covariates but assume their decision had been not to exercise. Thus, each man obtains the unobserved attributes of a man with his characteristics who chose not to exercise. When we take the contrast of those two counterfactuals, we have the effect on the probability of an attack for someone who chose not to exercise. We can average those effects too. Adding the obligatory vce() option to get population standard errors, we have

```
. margins r(0 1).exintensity, subpop(if exintensity == 1)  
> predict(base(exintensity=1)) predict(base(exintensity=0))  
> contrast(effects nowald) vce(unconditional)  
Contrasts of predictive margins  
1._predict : Pr(attack==yes), predict(base(exintensity=1))  
2._predict : Pr(attack==yes), predict(base(exintensity=0))
```

	Unconditional					
	Contrast	Std. Err.	z	P> z	[95% Conf. Interval]	
exintensity@ _predict (moderate vs none) 1	-.2992132	.0592607	-5.05	0.000	-.4153619	-.1830644
(moderate vs none) 2	-.2352377	.0477143	-4.93	0.000	-.3287561	-.1417193

As we surmised, the first line is the ATET for moderate exercise and exactly matches the line from estat teffects. The second line is the average effect of treatment if the men who exercise moderately are instead given the unobserved attributes of men with exactly their observed characteristics

but who choose not to exercise. The difference in the effects is about 0.06. That makes the average effects of the unobserved attributes on those who exercise moderately about 25% greater than the effect would be for the same men had they had the attributes of nonexercisers:  $0.06/0.24 = 0.25$ .

We can use `margins` to test whether the ATE for heavy exercise and the ATE for moderate exercise are equal. We specify two `predict()` options. On the first, we request treatment effects (`te`) for heavy exercisers (`tlevel(heavy)`). On the second, we request the treatment effects for moderate exercisers (`tlevel(moderate)`). We add `contrast(predict(r))` to request the difference between the predictions (their contrast). Finally, we use `vce(unconditional)` to request standard errors that account for sampling variability in the covariates and thus allow us to make inferences about the population.

```
. margins, predict(te tlevel(heavy)) predict(te tlevel(moderate))
> contrast(predict(r)) vce(unconditional)
Contrasts of predictive margins
1._predict    : treatment effect Pr(attack==yes), exintensity: heavy vs. none,
                predict(te tlevel(heavy))
2._predict    : treatment effect Pr(attack==yes), exintensity: moderate vs.
                none, predict(te tlevel(moderate))
```

	df	chi2	P>chi2
_predict	1	0.01	0.9085

	Unconditional			
	Contrast	Std. Err.	[95% Conf. Interval]	
_predict (2 vs 1)	.0083103	.0722814	-.1333587	.1499793

We cannot reject that the ATE for heavy exercise is equal to the ATE for moderate exercise. This result agrees with what we saw when we tested the coefficients for heavy and moderate exercise.

As we have seen repeatedly in the examples in the manual, most of the interesting questions are answered by `estat teffects` and `margins` and not by the parameter estimates themselves. This is particularly true of models estimated using `eprobit` and `eoprobit`.

Also see

- [ERM] [eprobit](#) — Extended probit regression
- [ERM] [eprobit postestimation](#) — Postestimation tools for `eprobit`
- [ERM] [estat teffects](#) — Average treatment effects for extended regression models
- [ERM] [intro 5](#) — Treatment assignment features
- [ERM] [intro 8](#) — Conceptual introduction via worked example



# Title

example 6a — Ordered probit regression with endogenous treatment

DescriptionRemarks and examplesAlso see

## Description

In this example, we show how to estimate and interpret the results of an extended regression model with an ordinal outcome and endogenous treatment.

## Remarks and examples

We are studying the effect of having health insurance on women’s health status, which we measure with a health score from 1 (poor) to 5 (excellent). We want to estimate the average treatment effect (ATE) of insurance on the probability of having each of the five statuses. We suspect that our model needs to account for the health insurance being an endogenous treatment.

In our fictional study, we collect data on a sample of 6,000 women between the ages of 25 and 30. In addition to the insurance indicator, we include an indicator for whether the woman exercises regularly and the number of years of schooling she completed (`grade`) as exogenous covariates. For our treatment model, we use `grade` and an indicator for whether the woman is currently working or attending school (`workschool`), which is excluded from the outcome model.

```
. use http://www.stata-press.com/data/r15/womenhlth
(Women's health status)
. eoprobit health i.exercise grade, entreat(insured = grade i.workschool)
> vce(robust)
(iteration log omitted)
```

```
Extended ordered probit regression          Number of obs    =      6,000
                                           Wald chi2(4)      =      516.93
Log pseudolikelihood = -9105.4376          Prob > chi2       =      0.0000
```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
health						
exercise#						
insured						
yes#no	.5296149	.0619049	8.56	0.000	.4082835	.6509463
yes#yes	.5190249	.033872	15.32	0.000	.4526371	.5854127
insured#						
c.grade						
no	.1079014	.0250326	4.31	0.000	.0588383	.1569645
yes	.1296456	.0107428	12.07	0.000	.10859	.1507012
insured						
grade	.3060024	.0100506	30.45	0.000	.2863036	.3257012
workschool						
yes	.5387767	.0446794	12.06	0.000	.4512067	.6263466
_cons	-3.592452	.1348431	-26.64	0.000	-3.85674	-3.328165
/health						
insured#						
c.cut1						
no	.6282326	.2393499			.1591154	1.09735
yes	-.7255086	.2470598			-1.209737	-.2412803
insured#						
c.cut2						
no	1.594089	.2300159			1.143266	2.044912
yes	.4404531	.1986825			.0510426	.8298636
insured#						
c.cut3						
no	2.526424	.2241048			2.087186	2.965661
yes	1.332514	.1845713			.9707608	1.694267
insured#						
c.cut4						
no	3.41748	.2356708			2.955574	3.879386
yes	2.292828	.1760594			1.947758	2.637899
corr(e.ins~d, e.health)	.3414241	.0940374	3.63	0.000	.1460223	.5111858

The estimated correlation between the errors from the health status equation and the errors from the health insurance equation is 0.34. This is significantly different from zero, so the treatment choice of being insured is endogenous. Because it is positive, we conclude that unobserved factors that increase the chance of having health insurance tend to also increase the chance of being in a high health status.

We see estimates of both the coefficients and the cutpoints for two equations, one for insured women (`yes`) and one for uninsured (`no`). For both insured and uninsured, exercise and education have positive effects on health status.

We could use `estat teffects` to estimate the ATE of insurance on the probabilities of each health category.

```
. estat teffects
```

Feel free to run that command and see the results. We estimate and interpret other estimates of these ATEs in [\[ERM\] example 6b](#) after adjusting for endogenous sample selection that is introduced in that example. The ATE estimates there are slightly different, but they estimate the same thing. Given a sufficiently large sample, the two sets of estimates would converge to the same values.

## Also see

[\[ERM\] eoprobit](#) — Extended ordered probit regression

[\[ERM\] eoprobit postestimation](#) — Postestimation tools for `eoprobit`

[\[ERM\] estat teffects](#) — Average treatment effects for extended regression models

[\[ERM\] intro 5](#) — Treatment assignment features

[\[ERM\] intro 8](#) — Conceptual introduction via worked example

Description

Continuing from [ERM] [example 6a](#), we show you how to estimate and interpret the results of a model for an ordinal outcome when the model includes an endogenous treatment and the data are subject to endogenous sample selection.

Remarks and examples

Suppose that we collected our data at doctors’ offices and thus observe health score information only from women who visited their doctor in the study time frame (`drvisit = 1`). We suspect that unobserved factors that affect whether a woman visited the doctor are related to those that affect whether she has insurance and to those that affect her health status. Thus, we have an endogenously selected sample and an endogenously chosen treatment.

For our selection model, we use the endogenous treatment indicator for insurance status and regular checkups before the study (`regcheck`), which is excluded from the outcome model. Our command is otherwise exactly the same as specified in [ERM] [example 6a](#).

```
. eoprobit health i.exercise c.grade, entreat(insured = grade i.workschool)
> select(select = i.insured i.regcheck) vce(robust)
(iteration log omitted)

Extended ordered probit regression          Number of obs      =      6,000
                                           Selected           =      4,693
                                           Nonselected        =      1,307

                                           Wald chi2(4)       =     367.30
                                           Prob > chi2        =      0.0000

Log pseudolikelihood = -9806.1189
```

	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
health						
exercise#						
insured						
yes#no	.4169984	.0851131	4.90	0.000	.2501798	.583817
yes#yes	.5399986	.037546	14.38	0.000	.4664098	.6135874
insured#						
c.grade						
no	.1317866	.0342405	3.85	0.000	.0646765	.1988967
yes	.1343324	.0129342	10.39	0.000	.1089818	.159683

select						
insured						
yes						
	1.01669	.092325	11.01	0.000	.8357364	1.197644
regcheck						
yes						
	.5374105	.0397297	13.53	0.000	.4595417	.6152793
_cons						
	-.1690644	.0743716	-2.27	0.023	-.3148301	-.0232987
insured						
grade						
	.3057852	.0100116	30.54	0.000	.2861628	.3254076
workschool						
yes						
	.5314797	.0452607	11.74	0.000	.4427703	.6201891
_cons						
	-3.584315	.1348183	-26.59	0.000	-3.848554	-3.320077
/health						
insured#						
c.cut1						
no						
	.7262958	.3313472			.0768673	1.375724
yes						
	-.5450451	.3181876			-1.168681	.0785912
insured#						
c.cut2						
no						
	1.719809	.3129056			1.106526	2.333093
yes						
	.5683456	.2464686			.085276	1.051415
insured#						
c.cut3						
no						
	2.620793	.3056038			2.021821	3.219766
yes						
	1.442022	.2227768			1.005387	1.878656
insured#						
c.cut4						
no						
	3.48945	.3158536			2.870389	4.108512
yes						
	2.391497	.2090187			1.981828	2.801166
corr(e.sel~t,						
e.health)						
	.496699	.0990366	5.02	0.000	.2795869	.665485
corr(e.ins~d,						
e.health)						
	.4032487	.121518	3.32	0.001	.1421331	.6118937
corr(e.ins~d,						
e.select)						
	.2661948	.0555596	4.79	0.000	.1543216	.3713287

At both levels of the treatment, exercise and education still have positive effects on health status.

The correlation between the errors from the selection equation and the errors from the main equation is 0.497. This is significantly different from zero, so we confirm our suspicion of endogeneity. Because it is positive, we conclude that unobservable factors that increase the chance of being in the study also tend to increase the chance of being in a higher health status category.

What are the expected average probabilities of being in each health status if every woman had insurance? If every woman did not have insurance? We can answer those questions using `estat teffects`.

```
. estat teffects, pomean
Predictive margins                                Number of obs      =      6,000
P0mean_Pr1   : Pr(health=1=poor)
P0mean_Pr2   : Pr(health=2=not good)
P0mean_Pr3   : Pr(health=3=fair)
P0mean_Pr4   : Pr(health=4=good)
P0mean_Pr5   : Pr(health=5=excellent)
```

	Margin	Unconditional Std. Err.	z	P> z	[95% Conf. Interval]	
P0mean_Pr1						
insured						
no	.1028382	.0327177	3.14	0.002	.0387126	.1669637
yes	.0058955	.0033611	1.75	0.079	-.0006921	.0124831
P0mean_Pr2						
insured						
no	.2621517	.0479497	5.47	0.000	.1681719	.3561314
yes	.0618234	.0116191	5.32	0.000	.0390504	.0845965
P0mean_Pr3						
insured						
no	.3216819	.0259933	12.38	0.000	.270736	.3726278
yes	.1759926	.0100741	17.47	0.000	.1562478	.1957374
P0mean_Pr4						
insured						
no	.2144017	.0402798	5.32	0.000	.1354547	.2933488
yes	.3237595	.009282	34.88	0.000	.3055672	.3419519
P0mean_Pr5						
insured						
no	.0989265	.0521147	1.90	0.058	-.0032163	.2010694
yes	.4325289	.0165829	26.08	0.000	.400027	.4650309

These are the estimates of the average [potential-outcome means](#) for the population. We can consider the values in this table to be either the average proportion of women in the status category (what is reported) or the average probabilities for all women being in a status category (if we multiply by 100). The first pair of rows shows the probabilities of being in the first health status, poor. If all women are uninsured, the probability of having a poor health status is 0.10. If all women are insured, that probability falls to 0.01. At the other end of the spectrum, only 9.9% of women are expected to have excellent health if no women are insured. That number rises to 43.3% if all women are insured.

If we sum all the proportions labeled no, that sum is 1.0. The same is true of the yeses. The sum of the proportions must be 1.0 because each woman can be in only one health status.

In any health status, if we subtract the potential-outcome mean when assuming all women are uninsured from the mean when assuming all women to be insured, we estimate the average treatment effect (ATE). This is the ATE that being insured has on the probability of being in the health status category. Let's do that.

```
. estat teffects
Predictive margins                                Number of obs      =      6,000
ATE_Pr1      : Pr(health=1=poor)
ATE_Pr2      : Pr(health=2=not good)
ATE_Pr3      : Pr(health=3=fair)
ATE_Pr4      : Pr(health=4=good)
ATE_Pr5      : Pr(health=5=excellent)
```

	Unconditional					
	Margin	Std. Err.	z	P> z	[95% Conf. Interval]	
ATE_Pr1 insured (yes vs no)	-.0969427	.0333853	-2.90	0.004	-.1623767	-.0315086
ATE_Pr2 insured (yes vs no)	-.2003283	.0552089	-3.63	0.000	-.3085358	-.0921207
ATE_Pr3 insured (yes vs no)	-.1456893	.0322109	-4.52	0.000	-.2088216	-.082557
ATE_Pr4 insured (yes vs no)	.1093578	.0437353	2.50	0.012	.0236382	.1950774
ATE_Pr5 insured (yes vs no)	.3336024	.0637745	5.23	0.000	.2086066	.4585982

Looking at the last line, we see that the average proportion of being in excellent health in the population of women aged 25 to 30 is 0.33 greater when all women have health insurance versus when no women have health insurance.

Because we specified `vce(robust)` at estimation, all of our estimates from `estat teffects` reported standard errors for the population ATE rather than standard errors that are conditional on the sample ATE.

Also see

- [ERM] [eoprobit](#) — Extended ordered probit regression
- [ERM] [eoprobit postestimation](#) — Postestimation tools for eoprobit
- [ERM] [estat teffects](#) — Average treatment effects for extended regression models
- [ERM] [intro 4](#) — Endogenous sample-selection features
- [ERM] [intro 5](#) — Treatment assignment features
- [ERM] [intro 8](#) — Conceptual introduction via worked example

Description

predict’s features are documented in

- [ERM] [eregress predict](#)
- [ERM] [eintreg predict](#)
- [ERM] [eprobit predict](#)
- [ERM] [eoprobit predict](#)
- [ERM] [predict treatment](#)

Here, we document predict’s advanced features.

Syntax

```
predict [type] newvar [if] [in] [, treatstatistic howcalculated treatmodifier  
oprobitmodifier advanced]
```

In some cases, more than one new variable needs to be specified:

```
predict [type] { stub*|newvarlist } [if] [in] [, treatstatistic howcalculated  
treatmodifier oprobitmodifier advanced]
```

With the exception of *advanced*, you have seen this syntax in the other `predict` manual entries. We will not cover old ground.

<i>advanced</i>	Description
<code>equation(depvar)</code>	calculate results for specified dependent variable
<code>nooffset</code>	ignore option <code>offset()</code> specified when model was fit in making calculation
<code>pr(a, b)</code>	calculate $\Pr(a < \mathbf{x}_i\boldsymbol{\beta} + e_i.depvar < b)$ ; <i>a</i> and <i>b</i> are numbers or variable names
<code>e(a, b)</code>	calculate $E(y_i a < y_i < b)$ , where $y_i = \mathbf{x}_i\boldsymbol{\beta} + e_i.depvar$ ; <i>a</i> and <i>b</i> are numbers or variable names
<code>scores</code>	calculate equation-level score variables

Also note that even though option `mean` was not included in *treatstatistic* for `eprobit` and `eoprobit`, it is allowed with them. `mean` returns the probability of a positive outcome after `eprobit` and returns the expected value of the outcome after `eoprobit`.



## Remarks and examples

The most important of the advanced features is the `equation()` option. Previously, we documented that `predict` calculates results for the main equation only. That was not true. The `equation()` option can be used to target the other equations. The `equation()` option is important because it can apply so many of `predict`'s features to them.

ERMs provide three types of equations. The `endogenous()` option names two of them and leaves the other unnamed:

```
endogenous(..., none specified ...)  
endogenous(..., probit ...)  
endogenous(..., oprobit ...)
```

*none specified* should have been called `linear`. Meanwhile, `entreat()` adds `probit` or `oprobit` equations, `select()` adds `probit` equations, and `tobitselect()` adds `linear` equations. Thus, there are three types of equations in total: `linear`, `probit`, and `oprobit`.

`equation()` can be used to provide the following `predict` features with the other equations in the model:

Option	Description
Linear equations	
<code>mean</code>	linear prediction
<code>xb</code>	linear prediction excluding complications
<code>ystar()</code>	censored prediction
<code>e()</code>	constrained expected value
<code>pr()</code>	probability in range
Probit equations	
<code>xb</code>	linear prediction excluding complications
<code>pr</code>	probability of positive outcome
<code>mean</code>	synonym for <code>pr</code>
Ordered probit equations	
<code>xb</code>	linear prediction excluding complications
<code>pr</code>	probability of each outcome
<code>mean</code>	expected value of outcome

Note 1: Option `outlevel(#)` is used with `pr` in `oprobit` equations to restrict the calculation to the specified outcome.

Note 2: When `equation(devar)` is the main equation, you can use any of `predict`'s options.

Note 3: For the main equation, options `e()` and `pr()` can be used with *howcalculated* options `fix()`, `base()`, and `target()`.

Options not allowed with `equation()` are disallowed. The disallowed options include `predict`'s treatment options as well as `fix()`, `base()`, and `target()`.

## Methods and formulas

See [Methods and formulas](#) of [ERM] [eprobit postestimation](#).

## Also see

- [ERM] **eintreg postestimation** — Postestimation tools for eintreg
- [ERM] **eintreg predict** — predict after eintreg
- [ERM] **eoprobit postestimation** — Postestimation tools for eoprobit
- [ERM] **eoprobit predict** — predict after eoprobit
- [ERM] **eprobit postestimation** — Postestimation tools for eprobit
- [ERM] **eprobit predict** — predict after eprobit
- [ERM] **eregress postestimation** — Postestimation tools for eregress
- [ERM] **eregress predict** — predict after eregress

Description	Syntax	Options
Remarks and examples	Methods and formulas	Also see

## Description

`predict` has options to predict potential-outcome means, treatment effects, and treatment effects on the treated after models fit using the `entreat()` or `extreat()` option. The `predict` options are described below.

For standard use of `predict`, see

[ERM] [eregress predict](#)

[ERM] [eintreg predict](#)

[ERM] [eprobit predict](#)

[ERM] [eoprobit predict](#)

For advanced use of `predict`, see

[ERM] [predict advanced](#)

Also see [ERM] [estat teffects](#) for reports of average treatment statistics.

## Syntax

You previously fit a model by using the `entreat()` or `extreat()` option,

```

eregress   y      x1 ... , ... entreat(treated = ...) ...
eintreg    y1 yu  x1 ... , ... entreat(treated = ...) ...
eprobit    y      x1 ... , ... entreat(treated = ...) ...
eoprobit   y      x1 ... , ... entreat(treated = ...) ...
eregress   y      x1 ... , ... extreat(treated) ...
eintreg    y1 yu  x1 ... , ... extreat(treated) ...
eprobit    y      x1 ... , ... extreat(treated) ...
eoprobit   y      x1 ... , ... extreat(treated) ...
    
```

In these cases, `predict` has extra features. `predict`'s extra syntax for these features is

```

predict [type] newvar [if] [in], treatstatistic [treatmodifier oprobitmodifier]
    
```

In some cases, more than one new variable needs to be specified:

```

predict [type] { stub* | newvarlist } [if] [in], treatstatistic [treatmodifier
    oprobitmodifier]
    
```

<i>treatstatistic</i>	Description
<code>pomean</code>	potential-outcome mean (POM)
<code>te</code>	treatment effect (TE)
<code>tet</code>	treatment effect on the treated (TET)

<i>treatmodifier</i>	Description
<code>tlevel(#)</code>	treatment level for which <i>treatstatistic</i> is calculated
# may be specified as a value recorded in variable <code>treated</code> , such as 1, 2, ... or such as 1, 5, ..., depending on the values recorded.	
# may also be specified as #1, #2, ..., meaning the first, second, ... values recorded in <code>treated</code> .	

<i>oprobitmodifier</i>	Description
<code>outlevel(#)</code>	ordered outcome for which <i>treatstatistic</i> is calculated
When used after models fit with <code>eoprobit</code> , <i>treatstatistic</i> is calculated for the specified outcome, or for the first outcome if you do not specify otherwise.	
<code>outlevel(#)</code> specifies the outcome for which statistics are to be calculated. # is specified in the same way as with <code>tlevel()</code> , but the meaning is different. In the case of <code>outlevel()</code> , you are specifying the outcome, not the treatment level.	

## Options

The options for the statistic to be calculated—`pomean`, `te`, and `tet`—are mutually exclusive. You calculate one treatment statistic per `predict` command.

`pomean` calculates the POMs for each treatment level. The POMs are the expected value of *y* that would have been observed if everyone was assigned to each of the treatment levels.

If there were two treatment levels (a control and a treatment), you would type

```
. predict pom1 pom2, pomean
```

If there were three levels, you would type

```
. predict pom1 pom2 pom3, pomean
```

`pomean` can alternatively be used with `tlevel()` to produce individual POMs:

```
. predict pom1, pomean tlevel(#1)
. predict pom2, pomean tlevel(#2)
```

If you have fit the model using `eoprobit`, the POMs calculated for the examples above would be for *y*'s first outcome. You can change that. See [Predicting treatment effects after eoprobit](#) in [Remarks and examples](#) below.

`te` calculates the TEs for each treatment level. The TEs are the differences in the POMs. For instance, if there were two treatment levels—a control and a treatment—there would be one treatment effect and it would be `pom2-pom1`. If there were three levels, there would be two treatment effects, `pom2-pom1` and `pom3-pom1`.

If there were two treatment levels—a control and a treatment—you would type

```
. predict te2, te
```

If there were three levels, you would type

```
. predict te2 te3, te
```

`te` can alternatively be used with `tlevel()` to produce individual TES:

```
. predict te2, te tlevel(#2)
. predict te3, te tlevel(#3)
```

If you have fit the model using `eoprobit`, the TES calculated for the examples above would be for  $y$ 's first outcome. You can change that. See [Predicting treatment effects after eoprobit](#) in [Remarks and examples](#) below.

`tet` calculates the TETs. The TETs are the differences in the POMs conditioned on treatment level.

If there were two treatment levels—a control and a treatment—you would type

```
. predict tet2, tet
```

If there were three levels, you would type

```
. predict tet2 tet3, tet
```

`tet` can alternatively be used with `tlevel()` to produce individual TETs:

```
. predict tet2, tet tlevel(#2)
. predict tet3, tet tlevel(#3)
```

If you have fit the model using `eoprobit`, the TETs calculated for the examples above would be for  $y$ 's first outcome. You can change that. See [Predicting treatment effects after eoprobit](#) in [Remarks and examples](#) below.

`tlevel(#)` is optionally used with `pomean`, `te`, or `tet`. Its use is illustrated above.

`outlevel(#)` is optionally used with `pomean`, `te`, or `tet` with models fit by `eoprobit`. See [Predicting treatment effects after eoprobit](#) in [Remarks and examples](#) below.

## Remarks and examples

Remarks are presented under the following headings:

[Predicting treatment effects after eregress and eintreg](#)

[Predicting treatment effects after eprobit](#)

[Predicting treatment effects after eoprobit](#)

### Predicting treatment effects after eregress and eintreg

`eregress` and `eintreg` concern models with a continuous outcome variable. In `eregress` models,  $y_i$  is observed. In `eintreg` models,  $y_i$  is not observed directly, but it is known that  $y_l_i \leq y_i \leq y_u_i$ .

Thus, the treatment statistics are expressed in the units of  $y$ . If  $y$  is blood pressure, the units are presumably mmHG. POMs are in mmHG. TES and TETs are differences in blood pressure expressed in mmHG.

### Predicting treatment effects after eprobit

`eprobit` concerns models with binary outcomes, and predictions are in terms of the probability of a positive outcome. Thus, POMs are probabilities. TES and TETs are differences in probabilities.

## Predicting treatment effects after eoprobit

`eoprobit` concerns models with ordinal outcome variables, and predictions are in terms of the probabilities—the probability of each outcome.

Treatment statistics are calculated on the basis of probabilities of outcomes. Thus, POMs are probabilities. TES and TETs are differences in probabilities.

We want probabilities and differences in probabilities, but you need to specify which probability. The probability for the first outcome? The second?

If you do not specify which and simply type

```
. predict pom1 pom2 pom3, pomean
```

then the POMs are calculated for the first outcome, what `eoprobit` calls `outlevel(#1)`. If you wanted to obtain the POMs for `outlevel(#2)`, you would type

```
. predict pom1 pom2 pom3, pomean outlevel(#2)
```

If you wanted them for `outlevel(#3)`, you would type

```
. predict pom1 pom2 pom3, pomean outlevel(#3)
```

The same logic applies to calculating TE and TET with the `te` and `tet` options. `outlevel(#1)` is used unless you specify otherwise.

## Methods and formulas

See *Methods and formulas* in [\[ERM\] eintreg](#), [\[ERM\] eoprobit](#), [\[ERM\] eprobit](#), and [\[ERM\] eregress](#).

## Also see

[\[ERM\] eintreg postestimation](#) — Postestimation tools for `eintreg`

[\[ERM\] eintreg predict](#) — predict after `eintreg`

[\[ERM\] eoprobit postestimation](#) — Postestimation tools for `eoprobit`

[\[ERM\] eoprobit predict](#) — predict after `eoprobit`

[\[ERM\] eprobit postestimation](#) — Postestimation tools for `eprobit`

[\[ERM\] eprobit predict](#) — predict after `eprobit`

[\[ERM\] eregress postestimation](#) — Postestimation tools for `eregress`

[\[ERM\] eregress predict](#) — predict after `eregress`

## Description

ERMs allow endogenous covariates, but they must form a triangular system, also known as a recursive system. Said differently, ERM's do not allow simultaneous causation. This was explained for simple cases in [ERM] [intro 3](#). How to triangularize complicated systems is described below.

## Remarks and examples

The day will come when you try to fit a model and the ERM command responds with the following error:

```
. eregress y w1 w2 w3 x1 x2,
>     endogenous(w1 = w2   z1 z2 x1 x2 x5, nomain)
>     endogenous(w2 = w1   z1 z3 x1 x2 x5, nomain)
>     endogenous(w3 = w1   z4 z5 x1 x2 x5, nomain)
>     endogenous(z1 =     z5   x1 x2 x4, nomain)
endogenous variables do not form a triangular system
The problem may be fixable. See triangularizing the system.
r(459);
```

The error can even occur in simple models:

```
. eregress y w1 x2 x3, endogenous(w1 = y z1 x2, nomain)
endogenous variables do not form a triangular system
The problem may be fixable. See triangularizing the system.
r(459);
```

The error message says the problem may be fixable. We explain below how to find the problem, how to determine whether it is fixable, and how to fix it when it is.

Remarks are presented under the following headings:

- What is a triangular system?*
- Triangularizing nontriangular systems*
- You can only triangularize linear equations*
- Options `entreat()`, `select()`, and `tobitselect()` also add endogenous variables*
- Workarounds involving the main equation*
- Why the above is a workaround and not a fix*

## What is a triangular system?

ERMs require that the endogenous covariates in the model being fit form a triangular system. The endogenous covariates are the dependent variable in the main equation plus the dependent variables in the `endogenous()` options, and in the `entreat()`, `select()`, and `tobitselect()` options, but we will cover those options later.

The endogenous variables are  $y$ ,  $w1$ ,  $w2$ ,  $w3$ , and  $z1$  in the model

```
. eregress y w1 w2 w3 x1 x2 x5,
      endogenous(w1 =      z1 z2 x1 x2 x5, nomain)
      endogenous(w2 = w1   z1 z3 x1 x2 x5, nomain)
      endogenous(w3 = w1   z4 z5 x1 x2 x5, nomain)
      endogenous(z1 =      z5      x1 x2 x4, nomain)
      ///
```

The system that needs to be triangular is  $y$ ,  $w1$ ,  $w2$ ,  $w3$ , and  $z1$ . That system is

Endogenous variable	which depends on the endogenous variable(s)
$y$	$w1$ $w2$ $w3$
$w1$	$z1$
$w2$	$w1$ $z1$
$w3$	$w1$
$z1$	(none)

A system is triangular when the dependencies can be ordered such that each endogenous variable is already defined before it is used as an explanatory variable. The system, in order, is

Endogenous variable	which depends on the endogenous variable(s)
$z1$	(none)
$w1$	$z1$
$w3$	$w1$
$w2$	$w1$ $z1$
$y$	$w1$ $w2$ $w3$

The system is in order and triangular because

1. Endogenous variable  $z1$  depends on no other endogenous variables.
2. Endogenous variable  $w1$  depends on  $z1$ , and  $z1$ 's definition has already been listed.
3. Endogenous variable  $w3$  depends on  $w1$ , and  $w1$ 's definition has already been listed.
4. Endogenous variable  $w2$  depends on  $w1$  and  $z1$ , and their definitions have already been listed.
5. Endogenous variable  $y$  depends on  $w1$ ,  $w2$ , and  $w3$ , and their definitions have already been listed.

When the system is triangular, ERMs can fit the model.

## Triangularizing nontriangular systems

Consider the model

```
. eregress y w1 w2 w3 x1 x2 x5,
>      endogenous(w1 = w2   z1 z2 x1 x2 x5, nomain)
>      endogenous(w2 = w1   z1 z3 x1 x2 x5, nomain)
>      endogenous(w3 = w1   z4 z5 x1 x2 x5, nomain)
>      endogenous(z1 =      z5      x1 x2 x4, nomain)
endogenous variables do not form a triangular system
The problem may be fixable. See triangularizing the system.
r(459);
```



The ERM command has already told us that the system defined by this model is not triangular. Thus, if we try to order the definitions as we did above, we will not be successful. Where we run into difficulties, however, will tell us where the problem is.

The endogenous variables in this model are `y`, `w1`, `w2`, `w3`, and `z1`. Their definitions in the order in which they appear in the command are

Endogenous variable	which depends on the endogenous variable(s)
<code>y</code>	<code>w1 w2 w3</code>
<code>w1</code>	<code>w2 z1</code>
<code>w2</code>	<code>w1 z1</code>
<code>w3</code>	<code>w1</code>
<code>z1</code>	<i>(none)</i>

The definitions in as near to the correct order as we can get them are

Endogenous variable	which depends on the endogenous variable(s)
<code>z1</code>	<i>(none)</i>
<code>w1</code>	<code>z1 w2</code> ← problem here
<code>w2</code>	<code>w1 z1</code>
<code>w3</code>	<code>w1</code>
<code>y</code>	<code>w1 w2 w3</code>

The problem appears in the second line where `w1` is defined in terms of `z1` and `w2`: `w2` has not yet been defined. Obviously, we need to put its definition above that for `w1`. However, if we move the definition of `w2` above that of `w1`, we still have a problem: `w2` depends on `z1` and `w1`, and now `w1` has not yet been defined!

You might notice that there are three endogenous variables involved in the problem—`w1`, `w2`, and `w3`—but just focus on the first pair of definitions that cause the problem. It does not matter which two of the three they are. In our case, they are

```
endogenous(w1 = w2 z1 z2 x1 x2 x5, nomain)
endogenous(w2 = w1 z1 z3 x1 x2 x5, nomain)
```

As we said in [\[ERM\] intro 3](#), there is a workaround for the problem when both equations are linear, as they are in this case. The workaround is

When the simultaneous-causation problem occurs in linear equations defined by `endogenous()` options, remove the endogenous variable from one equation and substitute for it all the variables from the removed variable’s equation except, of course, the variable you just removed.

The workaround in this case either

1. Removes `w2` from the first equation and substitutes “`z1 z3 x1 x2 x5`” for it.
2. Removes `w1` from the second equation and substitutes “`z1 z2 x1 x2 x5`” for it.

It does not matter which we do.

To remind you, we are fixing the first equation:

```
endogenous(w1 = w2 z1 z2 x1 x2 x5, nomain)
```

When we remove `w2` and substitute “`z1 z3 x1 x2 x5`”, we obtain

```
z1 z3 x1 x2 x5 z1 z2 x1 x2 x5
```

Now, we need to remove the duplicates. Removing them, we have

```
z3 z1 z2 x1 x2 x5
```

Thus, the first equation becomes

```
endogenous(w1 = z3 z1 z2 x1 x2 x5, nomain)
```

We can now try fitting the model again:

```
. eregress y w1 w2 w3 x1 x2 x5, ///
      endogenous(w1 = z3 z1 z2 x1 x2 x5, nomain) ///
      endogenous(w2 = w1 z1 z3 x1 x2 x5, nomain) ///
      endogenous(w3 = w1 z4 z5 x1 x2 x5, nomain) ///
      endogenous(z1 = z5 x1 x2 x4, nomain)
```

When we try to fit the model, it will be successful or it will repeat the same error we saw earlier:

```
endogenous variables do not form a triangular system
The problem may be fixable. See triangularizing the system.
r(459);
```

In this case, the model will be successfully fit. If you do get the error, repeat the process. Remove the problems one at a time.

## You can only triangularize linear equations

The rule is

When the simultaneous-causation problem occurs in *linear* equations defined by `endogenous()` options, remove the endogenous variable from one equation and substitute for it all the variables from the removed variable’s equation except, of course, the variable you just removed.

Triangularization involves a pair of equations that must both be linear. In the example above, both were linear:

```
endogenous(w1 = w2 z1 z2 x1 x2 x5, nomain)
endogenous(w2 = w1 z1 z3 x1 x2 x5, nomain)
```

They would not have both been linear if either had been fit by `probit` or `oprobit`. If one or both of the equations had been

```
endogenous(w1 = w2 z1 z2 x1 x2 x5, nomain probit)
endogenous(w2 = w1 z1 z3 x1 x2 x5)
```

or

```
endogenous(w1 = w2 z1 z2 x1 x2 x5, nomain)
endogenous(w2 = w1 z1 z3 x1 x2 x5, nomain oprobit)
```

there would have been no solving the simultaneous-causation problem.

This linearity requirement applies only to the two equations directly involved. Other equations can be nonlinear and there will be no issue. The workaround we outlined would have worked just as well had the model been

```
. eregress y w1 w2 w3 x1 x2 x5,          ///
      endogenous(w1 = w2  z1 z2 x1 x2 x5, nomain)      ///
      endogenous(w2 = w1  z1 z3 x1 x2 x5, nomain)      ///
      endogenous(w3 = w1  z4 z5 x1 x2 x5, nomain probit)  ///
      endogenous(z1 =    z5   x1 x2 x4, nomain probit)
```

or even

```
. eprobit y w1 w2 w3 x1 x2 x5,          ///
      endogenous(w1 = w2  z1 z2 x1 x2 x5, nomain)      ///
      endogenous(w2 = w1  z1 z3 x1 x2 x5, nomain)      ///
      endogenous(w3 = w1  z4 z5 x1 x2 x5, nomain probit)  ///
      endogenous(z1 =    z5   x1 x2 x4, nomain probit)
```

## Options `entreat()`, `select()`, and `tobitselect()` also add endogenous variables

The example above contained repeated uses of the `endogenous()` option. When you make the list of endogenous variables, you must also include the dependent variables *treated* and *selected* from the options

```
entreat(treated= ...)
select(selected= ...)
tobitselect(selected= ...)
```

The above options make *treated* and *selected* endogenous. Unlike with the `endogenous()` option, however, the variables are not automatically added to the main equation even if you do not specify `nomain`.

These three options are nonlinear. If the simultaneous-causation problem involves equations created by these options, then there is no workaround for the simultaneous-causation problem.

## Workarounds involving the main equation

The example of the simultaneous-causation problem involved two equations defined by `endogenous()` options. The problem could also occur when one of the equations is the main equation. In [ERM] intro 3, we discussed problems involving the main equation as if they were different from simultaneous causation, but they are not. It is the same problem that has the same workaround, but with an important difference.

In workarounds involving equations defined by `endogenous()` equations, the workaround may be applied to either equation.

In workarounds involving the main equation and an `endogenous()` equation, the workaround must be applied to the `endogenous()` equation.

When the simultaneous-causation problem involves the main equation fit by `eregress` and an `endogenous()` linear equation, remove the dependent variable from the `endogenous()` equation and substitute for it all the variables from the main equation except, of course, the variable you just removed.

Also notice that this rule applies only to main equations fit by `eregress`. What about `eintreg`, `eprobit`, and `eoprobit`?

The simultaneous-causation problem does not arise in models fit by `eintreg`. There is no way you could include `eintreg`'s dependent variables as explanatory variables in another equation.

The simultaneous-causation problem can arise in models fit by `eprobit` and `eoprobit`, but those are nonlinear equations, and that means you cannot apply the workaround. The workaround requires that both equations be linear.

The main equation must be linear if it is one of the two equations involved in the simultaneous-causation problem. Otherwise, the main equation is not required to be linear.

## Why the above is a workaround and not a fix

It is a detail, but you may have noticed that we provided a workaround and not a fix. The purpose of ERMs is to obtain valid estimates of the coefficients of the main equation—its structural parameters—in light of lots of complications. It so happens that ERMs produce estimates of structural parameters for all the other equations if the system is truly triangular. That is not important, but it is true.

When you triangularize a nontriangular system, ERMs no longer produce estimates of the structural parameters for the equations that you modify. They produce estimates of the reduced-form equation, and that is sufficient. Valid estimates of the reduced-form equation ensures that estimates of the coefficients in the main equation are estimates of its structural parameters.

Thus, what we provided is a workaround, not a fix. If you use the workaround, do not interpret any equations modified as estimates of their structural parameters.

## Also see

[\[ERM\] intro 3](#) — Endogenous covariates features

# Glossary

**average structural function.** The average structural function (ASF) is used to calculate predicted values of ERMs.

The ASF averages out the heterogeneity caused by the endogeneity from a conditional mean or a conditional probability in a model with endogenous covariates. Applying the ASF to a conditional mean produces an average structural mean (ASM). Applying the ASF to a conditional probability produces an average structural probability (ASP). Contrasts of ASMs or ASPs produced by a covariate change define a causal structural effect. [Blundell and Powell \(2003, 2004\)](#) and [Wooldridge \(2005, 2014\)](#) are seminal papers that define and extend the ASF. See [Wooldridge \(2010, 22–24\)](#) for a textbook introduction.

**average structural mean.** The average structural mean (ASM) is the result of applying the [average structural function](#) to a conditional mean.

**average structural probability.** The average structural probability (ASP) is the result of applying the [average structural function](#) to a conditional probability.

**average treatment effect.** See [treatment effects](#).

**average treatment effect on the treated.** See [treatment effects](#).

**average treatment effect on the untreated.** See [treatment effects](#).

**binary variable.** A binary variable is any variable that records two values, the two values representing false and true, such as whether a person is sick. We usually speak of the two values as being 0 and 1 with 1 meaning true, but Stata requires merely that 0 means false and nonzero and nonmissing mean true. Also see [continuous variable](#), [categorical variable](#), and [interval variable](#).

**categorical variable.** A categorical variable is a variable that records the category number for, say, lives in the United States, lives in Europe, and lives in Asia. Categorical variables play no special role in this manual, but [ordered categorical variables](#) do. The example given is unordered. The categories United States, Europe, and Asia have no natural ordering. We listed the United States first only because the author of this manual happens to live in the United States.

The way we use the term, categorical variables usually record two or more categories, and the term binary variable is used for categorical variables having two categories.

We usually speak of categorical variables as if they take on the values 1, 2, . . . . Stata does not require that. However, the values do need to be integers.

**censoring, censored, left-censored, right-censored, interval-censored.** Censoring involves not observing something but knowing when and where you do not observe it.

For instance, sometimes patients/subjects/units being studied—observations in your dataset—have values equal to missing. Such observations are said to be censored when there is a reason they are missing. A variable is missing because a potential worker chooses not to work, because a potential patient chooses not to be a patient, because a potential subject was not prescribed the treatment, etc. Such censored outcomes cause difficulty when there is an unobserved component to the reason they are censored that is correlated with the outcome being studied. ERM option `select()` addresses these issues.

Another type of censoring—interval-censoring—involves not observing a value precisely but knowing its range. You do not observe blood pressure, but you know it is in the range 120 to 140. Or you know it is less than 120 or greater than 160. ERM command `eintreg` fits models in which the dependent variable is interval-censored.

Left-censoring is open-ended interval-censoring in which measurements below a certain value are unobserved. Blood pressure is less than 120.

Right-censoring is open-ended interval-censoring in which measurements above a certain value are unobserved. Blood pressure is above 160.

**conditional mean.** The conditional mean of a variable is the expected value based on a function of other variables. If  $y$  is a linear function of  $x_1$  and  $x_2$ — $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \text{noise}$ —then the conditional mean of  $y$  for  $x_1 = 2$  and  $x_2 = 4$  is  $\beta_0 + 2\beta_1 + 4\beta_2$ .

**confounding variable, confounder.** A confounding variable is an omitted explanatory variable in a model that is correlated with variables included in the model. The fitted coefficients on the observed variables will include the effect of the variables, as intended, plus the effect of being correlated with the omitted variable.

Confounders are often omitted from the model because they are unobserved. See [ERM] intro 3.

**continuous variable.** A continuous variable is a variable taking on any value on the number line. In this manual, however, we use the term to mean the variable is not a [binary variable](#), not a [categorical variable](#), and not an [interval variable](#).

**counterfactual.** The result that would be expected from a thought experiment that assumes things counter to what are currently true. What would be the average income if everyone had one more year of schooling? What would be the effect of an experimental medical treatment if the treatment were made widely available? Stata's `margins` command produces statistical answers to these kinds of thought experiments and reports standard errors as well.

**counterfactual predictions.** Counterfactual predictions are used when you have endogenous covariates in your main equation and you wish to estimate either counterfactuals or the effect on the outcome of changing the values of covariates. They are obtained using `predict` options `base()` and `fix()`.

**covariate.** A covariate is a variable appearing on the right-hand side (RHS) of a model. Covariates can be exogenous or endogenous, but when the term is used without qualification, it usually means exogenous covariate. Covariates are also known as explanatory variables. Also see [endogenous covariate](#) and [exogenous covariate](#).

**dependent variable.** A dependent variable is a variable appearing on the left-hand side of an equation in a model. It is the variable to be explained. Every equation of a model has a dependent variable. The term “the dependent variable” is often used in this manual to refer to the dependent variable of the [main equation](#). Also see [ERM] intro 3.

**endogenous covariate.** An endogenous covariate is a [covariate](#) appearing in a model 1) that is correlated with omitted variables that also affect the outcome; 2) that is measured with error; 3) that is affected by the dependent variable; or 4) that is correlated with the model's error. See [ERM] intro 3.

**endogenous sample selection.** Endogenous sample selection refers to situations in which the subset of the data used to fit a model has been selected in a way correlated with the model's outcome.

Mechanically, the subset used is the subset containing nonmissing values of variables used by the model. A variable is unobserved—contains missing values—because a potential worker chooses not to work, because a potential patient chooses not to be a patient, because a potential subject was not prescribed the treatment, etc. Such censored outcomes cause difficulty when there is an unobserved component to the reason they are censored that is correlated with the outcome being studied.

ERM option `select()` can address these issues when the dataset contains observations for which the dependent variable was missing.

**endogenous and exogenous treatment assignment.** See [treatment assignment](#).

**error.** Error is the random component (residual) appearing at the end of the equations in a model. These errors account for the unobserved information explaining the outcome variable. Errors in this manual are written as *e.depvarname*, such as  $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + e.y$ .

**exogenous covariate.** An exogenous covariate is a [covariate](#) that is uncorrelated with the error term in the model. See [\[ERM\] intro 3](#).

**explanatory variable.** Explanatory variable is another word for [covariate](#).

**extended regression models.** Extended regression models (ERMs) are generalized structural equation models that allow identity and probit links and Gaussian, binomial, and ordinal families for the main outcome. They extend interval regression, ordered probit, probit, and linear regression models by accommodating endogenous covariates, nonrandom and endogenous treatment assignment, and endogenous sample selection.

**individual-level treatment effect.** An individual-level [treatment effect](#) is the difference in the individuals outcome that would occur when given one treatment instead of another. It is the difference between two potential outcomes for the individual. The blood pressure after taking a pill minus the blood pressure were the pill not taken is the individual-level treatment effect of the pill on blood pressure.

**informative missingness.** See [missingness](#).

**instrument.** Instrument is an informal word for [instrumental variable](#).

**instrumental variable.** An instrumental variable is a variable that affects an [endogenous covariate](#) but does not affect the [dependent variable](#). See [\[ERM\] intro 3](#).

**interval measurement.** Interval measurement is a synonym for interval-censoring. See [censoring](#).

**interval variable.** An interval variable is actually a pair of variables that record the lower and upper bounds for a variable whose precise values are unobserved. `y1b` and `yub` might record such values for a variable *y*. Then it is known that, for each observation *i*,  $y1b_i \leq y \leq yub_i$ . ERM estimation command `eintreg` fits such models. Also see [censoring](#).

**interval-censoring.** See [censoring](#).

**left-hand-side (LHS) variable.** A left-hand-side variable is another word for [dependent variable](#).

**lost due to follow up.** Lost due to follow up refers to patients who are actively participating in a clinical trial but have been lost later. The statistical fear is that they are lost for reasons correlated with how well the experimental treatment was working for them.

**main equation.** The main equation in an ERM is the first equation specified, the equation appearing directly after the `eregress`, `eintreg`, `eprobit`, or `eoprobit` command. The purpose of ERMs is to produce valid estimates of the coefficients in the main equation, meaning the structural coefficients, in the presence of complications such as endogeneity, selection, or treatment assignment.

**measurement error, measured with error.** A variable measured with error has recorded value equal to  $x + \epsilon$ , where *x* is the true value. The error is presumably uncorrelated with all other errors in the model. In that case, fitted coefficients will be biased toward zero. See [\[ERM\] intro 3](#).

**missing at random (MAR).** See [missingness](#).

**missing completely at random (MCAR).** See [missingness](#).

**missing not at random (MNAR).** See [missingness](#).

**missingness.** Missingness refers to how missing observations in data occur. The categories are 1) missing not at random (MNAR), 2) missing at random (MAR), and 3) missing completely at random (MCAR).

In what follows we will refer to missing observations to mean not only observations entirely missing from a dataset but also the omitted observations because of missing values when fitting models.

MNAR observations refer to cases in which the missingness depends on the outcome under study. The solution in this case is to model that dependency. When observations are missing because of missing values, ERM option `select()` can be used to model the missingness.

MAR observation refer to cases in which the missingness does not depend on the outcome under study but does depend on other variables correlated with the outcome. The solution for some of the problems raised is to include those other variables as covariates in your model. Importantly, you do not need to model the reason for missingness.

MCAR observations are just that and obviously not a problem other than to cause loss of efficiency.

The MNAR and MAR cases are known jointly as informative missingness.

**multivalued treatment.** A multivalued treatment is a treatment with more than two arms. See [treatment arms](#).

**observational data.** Observational data are data collected over which the researcher had no control. The opposite of observational data is experimental data. Use of observational data often introduces statistical issues that experimental data would not. For instance, in a treatment study based on observational data, researchers had no control over treatment assignment; thus the treatment assignment needs to be modeled.

**omitted variables.** Omitted variables is an informal term for [covariates](#) that should appear in the model but do not. They do not because they are unmeasured, because of ignorance or other reasons. Problems arise when the variables that are not omitted are correlated with the omitted variables.

**ordered categorical variable.** An ordered categorical variable is a [categorical variable](#) in which the categories can be ordered, such as healthy, sick, and very sick. Actually recorded in the variable are integers such as 1, 2, and 3. The integers need not be sequential, but they must reflect the ordering. Also see [binary variable](#) and [continuous variable](#).

**outcome variable.** See [dependent variable](#).

**potential outcome.** Potential outcome is a term used in the treatment-effects literature. It is the outcome an individual would have had if given a specific treatment. Individual in this case means conditional on the individual's covariates, which are in the main equation in models fit by ERMs. It is the outcome that would have been observed for that individual. For instance, each patient in a study has one potential blood pressure after taking a pill and another had he or she not taken it. Also see [treatment effects](#).

**potential-outcome means.** Potential-outcome means (POMs) is a term used in the treatment-effects literature. They are the means (averages) of [potential outcomes](#). The average treatment effect (see [treatment effects](#)) is the difference between the potential-outcome mean for treated and untreated over the population.

**recursive (structural) model.** ERMs fit recursive models. A model is not recursive when one endogenous variable depends (includes its equation) on another endogenous variable that depends on the first. Said in symbols, when  $A$  depends on  $B$ , which depends on  $A$ . A model is also not recursive when  $A$  depends on  $B$  depends on  $C$ , which depends on  $A$ , and so on. See [\[ERM\] triangularize](#).



**reverse causation and simultaneous causation.** We use the term reverse causation in this manual when the [dependent variable](#) in the main equation of an ERM affects a [covariate](#) as well as when the covariate affects the dependent variable. Stressed persons may be physically unhealthy because they are stressed and further stressed because they are unhealthy. When a covariate suffers from reverse causation, the solution is to make it endogenous and find [instruments](#) for it.

Our use of the term reverse causation is typical of how it is used elsewhere. Reverse causation is a reason to make a variable endogenous. Reverse causation is discussed in [\[ERM\] intro 3](#).

The term simultaneous causation is sometimes used as a synonym for reverse causation elsewhere, but we draw a distinction. We use the term when two already endogenous variables affect each other. Simultaneous causation is discussed in [\[ERM\] triangularize](#).

**right-hand-side (RHS) variable.** A right-hand-side variable is another word for [covariate](#).

**sample selection.** Sample selection is another term for [endogenous sample selection](#).

**selection.** Selection is another term for [endogenous sample selection](#).

**selection on unobservables.** Selection on unobservables is another term for [endogenous sample selection](#).

**simultaneous causation.** See [recursive \(structural\) model](#).

**simultaneous system.** A simultaneous system is a multiple-equation model in which dependent variables can affect each other freely. The equation for  $y_1$  could include  $y_2$ , and the equation for  $y_2$  include  $y_1$ . ERMs cannot fit simultaneous systems. Because the focus of ERMs is on one equation in particular—the main equation—you can substitute the covariates for  $y_1$  into the  $y_2$  equation to form the reduced-form result and still obtain estimates of the structural parameters of the  $y_1$  equation. In this manual, we discuss this issue using the terms reverse causation and [recursive \(structural\) model](#). In the manual, it is discussed in [\[ERM\] triangularize](#).

**TE.** See [treatment effect](#).

**tobit estimator.** Tobit is an estimation technique for dealing with dependent variables that are censored. The classic tobit model dealt with left-censoring, in which the outcome variable was recorded as zero if it would have been zero or below. The estimator has since been generalized to dealing with models in which observations can be left-censored, right-censored, or interval-censored. See [censoring](#).

**treatment.** A treatment is a drug, government program, or anything else administered to a patient, job seeker, etc., in hopes of improving an outcome.

**treatment arms.** Sometimes, experiments are run on more than one [treatment](#) simultaneously. Each different treatment is called an arm of the treatment. The controls (those not treated) are also an arm of the treatment.

**treatment assignment.** Treatment assignment is the process by which subjects are assigned to a [treatment arm](#). That process can be endogenous or exogenous, meaning that the random component (error) in the assignment is correlated or is not correlated with the outcomes of the treatments. It is often endogenous because doctors assign subjects or subjects choose based in part on unobserved factors correlated with the treatment's outcome.

**treatment effects.** A treatment effect (TE) is the effect of a treatment in terms of a measured outcome such as blood pressure, ability to walk, likelihood of finding employment, etc. The statistical problem is to measure the effect of a treatment in the presence of complications such as censoring, treatment assignment, and so on.

ERMs fit treatment-effect models when one of the options `entreat()` or `extreat()` is specified for endogenous or exogenous treatment assignment. Meanwhile, the outcome model is specified in the main equation.

The TE is, for each person, the difference in the predicted outcomes based on the covariates in the main equation given that treatment is locked at treated or untreated.

The treatment effect on the treated (TET) is, for each person who was treated, the difference in the predicted outcomes based on the covariates in the main equation and the fact that they were assigned to or choose to be treated.

The treatment effect on the untreated (TEU) is, for each person who was not treated, the difference in predicted outcomes based on the covariates in the main equation and the fact that they were assigned to or choose not to be treated.

The average treatment effect (ATE) is an estimate of the average effect in a population after accounting for statistical issues.

The average effect on the treated (ATET) is an estimate of the average effect that would have been observed for those who were in fact treated in the data.

The average effect on the untreated (ATEU) is an estimate of the average effect that would have been observed for those who were in fact not treated in the data.

**triangular system.** See [recursive \(structural\) model](#).

## References

- Blundell, R. W., and J. L. Powell. 2003. Endogeneity in nonparametric and semiparametric regression models. In *Advances in Economics and Econometrics: Theory and Applications, Eighth World Congress*, ed. M. Dewatripont, L. P. Hansen, and S. J. Turnovsky, vol. 2, 312–357. Cambridge: Cambridge University Press.
- . 2004. Endogeneity in semiparametric binary response models. *Review of Economic Studies* 71: 655–679.
- Wooldridge, J. M. 2005. Unobserved heterogeneity and estimation of average partial effects. In *Identification and Inference for Econometric Models: Essays in Honor of Thomas Rothenberg*, ed. D. W. K. Andrews and J. H. Stock, 27–55. New York: Cambridge University Press.
- . 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.

# Subject and author index

See the [combined subject index](#) and the [combined author index](#) in the *Glossary and Index*.