wqsreg - A Stata command for weighted quantile sum regression

September 17, 2026

Marta Ponzano[1], Stefano Renzetti[2], Andrea Bellavia[3]

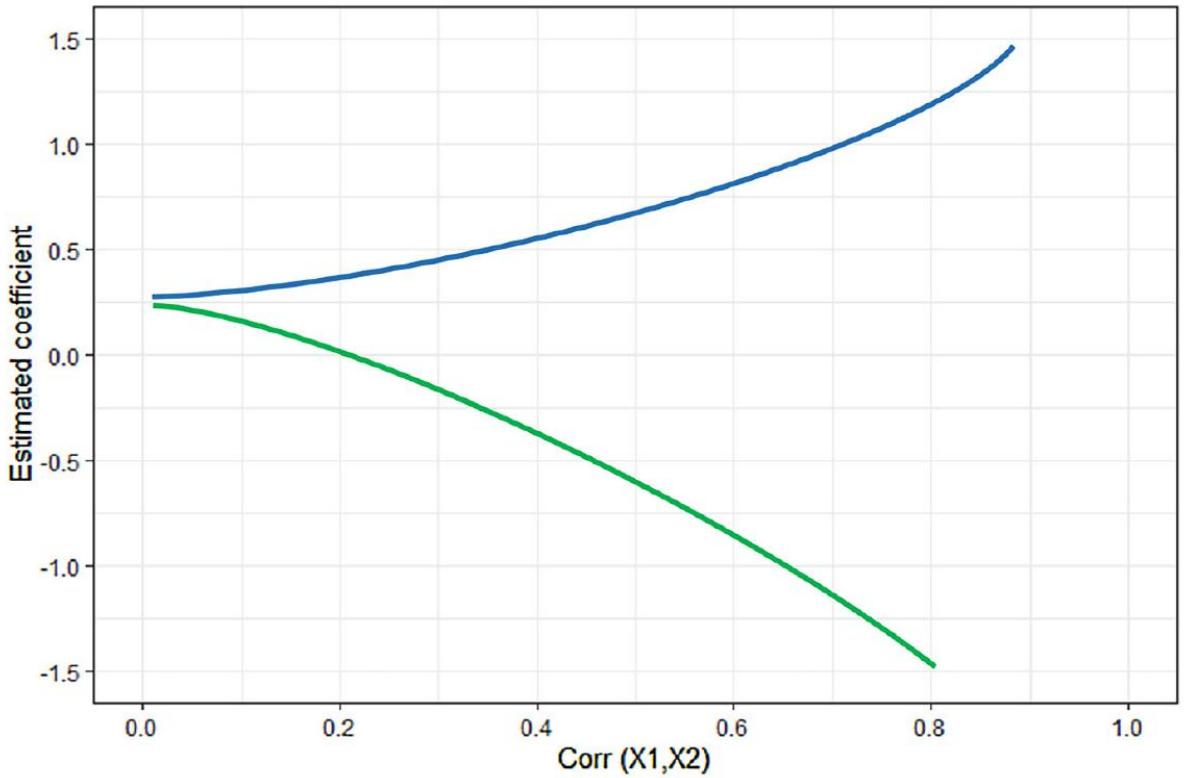
[1] Department of Life Sciences, Health and Health Professions, Link Campus University, Rome, Italy

[2] Department of Medicine and Surgery, University of Parma, Parma, Italy.

[3] Department of Environmental Health, Harvard T.H. Chan School of Public Health, Boston, Massachusetts, USA / TIMI Study Group, Brigham and Women's Hospital, Harvard Medical School, Boston, Massachusetts, USA

Correlated predictors

- The presence of highly correlated predictors is a well-known issue in numerous domains
- Including highly correlated covariates in the same regression model increases the risk of multicollinearity:
  - ↓ accuracy
  - ↑ standard errors



The reversal paradox.

Correlated predictors – environmental epidemiology

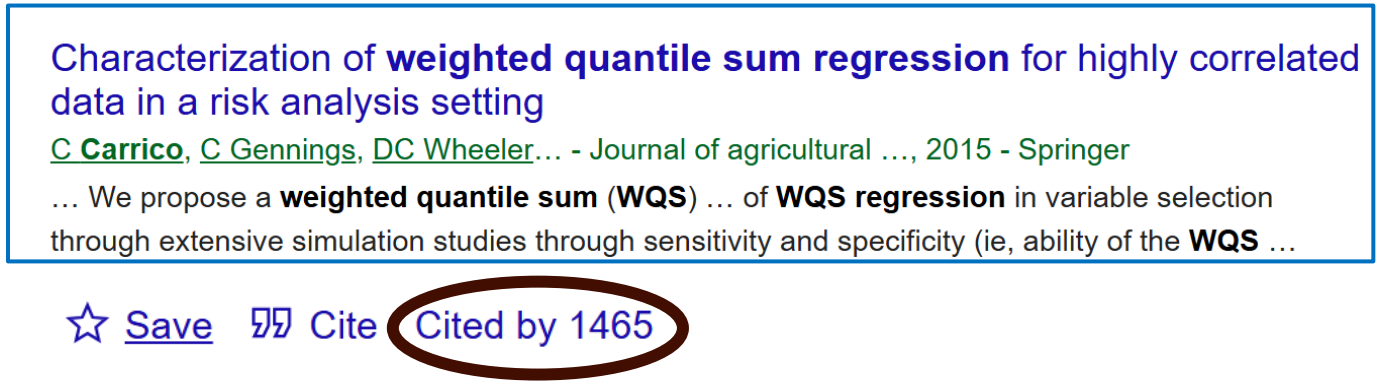
- This issue is particularly common in environmental epidemiology, when studying the health effects of a combined group of multiple factors, known as environmental mixture:

1. Mixture Bad Actors (specific role of each factor)
2. Joint Mixture Effect (overall)

1. 2.

Statistical methods

- To address this complexity, some methodologies have been considered (e.g., penalized regression) and *ad hoc* statistical methods have been developed and then used also in different fields outside of environmental epidemiology.



WQS Regression

- WQS regression is a supervised statistical approach where exposure aggregation via indexing is conducted in two steps:

1. Creating a summary index by weighting each mixture component
2. Including the index in a regression model

1. 2.

1. Creating the summary index - weights estimation

- Split the dataset into training and validation (40/60)
- Score all exposure factors into categories based on quantiles (quartiles)
- Generate *b* bootstrap samples of the training dataset (100)
- For each sample, estimate parameters (weights and βs):

$$f(y) = \beta_0 + \beta_1 \left(\sum_{i=1}^{c} w_i q_i \right) + \boldsymbol{\beta} \cdot \mathbf{C}'$$

$0 \leq w_i \leq 1$; $\sum_{i=1}^{c} w_i = 1$

Assumption of unidirectionality (pre-specified direction): only average results over bootstrap samples with either positive or negative estimates

- Derive $\overline{w}_i$ for each component based on estimates from bootstrap samples
- Derive the final estimate of the WQS as weighted sum of the components

$$WQS = \sum_{i=1}^{c} \bar{w}_i q_i$$

2. Including the index in a regression model

- Fit a regression model in the validation set to estimate the βs

$$f(y) = \beta_0 + \beta_1 \cdot WQS + \boldsymbol{\beta} \cdot \mathbf{C}'$$

WQS Regression in Stata

A Stata command had not yet been implemented. To address this gap, we developed a new command – **wqsreg**

› Eur J Epidemiol. 2026 Jun;41(6):709-713. doi: 10.1007/s10654-026-01423-0. Epub 2026 Jun 25.

Wqsreg: a Stata command for weighted quantile sum regression

Marta Ponzano [1] [2], Stefano Renzetti [3], Chris Gennings [4], Andrea Bellavia [5] [6]

Affiliations + expand

PMID: 42348089 DOI: 10.1007/s10654-026-01423-0

WQS Regression in Stata

-***wqsreg*** fits WQS regression for continuous, binary and count outcomes, while allowing for several flexible components of this framework. It returns estimates from WQS regression, generates plots of the estimated weights, and saves additional related information.

- *https://github.com/PonzanoMarta/wqsreg*
- *net install wqsreg, from("https://raw.githubusercontent.com/PonzanoMarta/wqsreg/master/") replace*

wqsreg -Syntax

outcome variable

list of exposure variables

```
wqsreg yvar, mixture(varlist) ///
[boot(integer 100) validation(integer 60) q(integer 4) b1_neg(integer 0)
cvar(varlist) seed(integer 0)
conv_maxiter(integer 2000) conv_vtol(real 0.000000001) technique(string ‘bfgs’)
model_fam(string 'Linear’)
savingWQSindex(string) figureName(string) savingWeights(string)
id(string) rh_rep(integer 1)]
```

list of confounders

wqsreg -Syntax

Number of bootstrap samples (>0):
- boot = 1: no bootstrapping
- default: boot = 100

Validation set percentage ([0, 100)):
- validation = 0: no split
- default: validation = 60

```
wqsreg yvar, mixture(varlist) ///
[boot(integer 100) validation(integer 60) q(integer 4) b1_neg(integer 0)
cvar(varlist) seed(integer 0)
conv_maxiter(integer 2000) conv_vtol(real 0.000000001) technique(string ‘bfgs’)
model_fam(string 'Linear’)
savingWQSindex(string) figureName(string) savingWeights(string)
id(string) rh_rep(integer 1)]
```

wqsreg -Syntax

Definition of quantiles:
- default: q = 4

Uni-directionality constraint:
- b1_neg = 1 : negative
- default: b1_neg = 0

```
wqsreg yvar, mixture(varlist) ///
[boot(integer 100) validation(integer 60) q(integer 4) b1_neg(integer 0)
cvar(varlist) seed(integer 0)
conv_maxiter(integer 2000) conv_vtol(real 0.000000001) technique(string ‘bfgs’)
model_fam(string 'Linear’)
savingWQSindex(string) figureName(string) savingWeights(string)
id(string) rh_rep(integer 1)]
```

random seed:
- default: seed = 0

wqsreg -Syntax

Maximum number of iterations to be performed before optimization:

- default: conv_maxiter=2000

Tolerance:

- default: conv_vtol= 0.000000001

```
wqsreg yvar, mixture(varlist) ///
[boot(integer 100) validation(integer 60) q(integer 4) b1_neg(integer 0)
cvar(varlist) seed(integer 0)
conv_maxiter(integer 2000) conv_vtol(real 0.000000001) technique(string ‘bfgs’)
model_fam(string 'Linear’)
savingWQSindex(string) figureName(string) savingWeights(string)
id(string) rh_rep(integer 1)]
```

Optimization method:

- technique = 'nr’: Modified Newton–Raphson
- default: technique = 'bfgs' (Broyden–Fletcher–Goldfarb–Shanno)

wqsreg -Syntax

```
wqsreg yvar, mixture(varlist) ///
[boot(integer 100) validation(integer 60) q(integer 4) b1_neg(integer 0)
cvar(varlist) seed(integer 0)
conv_maxiter(integer 2000) conv_vtol(real 0.000000001) technique(string ‘bfgs’)
model_fam(string 'Linear’)
savingWQSindex(string) figureName(string) savingWeights(string)
id(string) rh_rep(integer 1)]
```

Type of model:

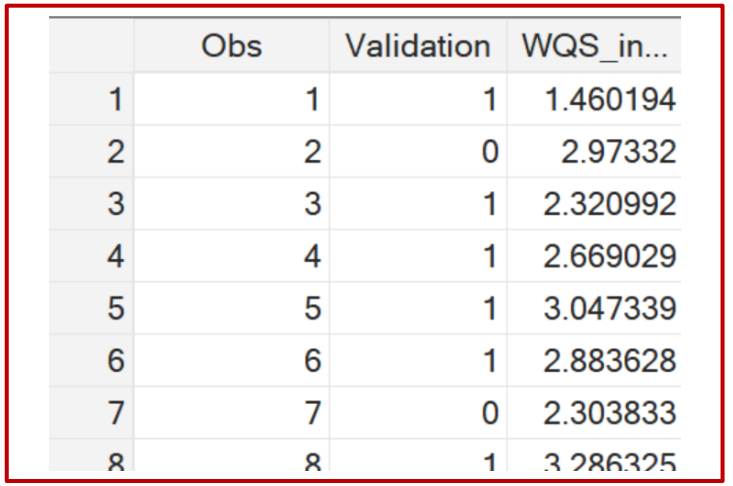
- model_fam='Poisson’
- model_fam ='Logistic’
- default: model_fam='Linear'

wqsreg -Syntax

Name of the dataset that will include the WQS index, only if you want to save it.

```
wqsreg yvar, mixture(varlist) ///
[boot(integer 100) validation(integer 60) q(integer 4) b1_neg(integer 0)
cvar(varlist) seed(integer 0)
conv_maxiter(integer 2000) conv_vtol(real 0.000000001) technique(string ‘bfgs’)
model_fam(string 'Linear’)
savingWQSindex(string) figureName(string) savingWeights(string)
id(string) rh_rep(integer 1)]
```

Variable that identifies the observations



	Obs	Validation	WQS_in...
1	1	1	1.460194
2	2	0	2.97332
3	3	1	2.320992
4	4	1	2.669029
5	5	1	3.047339
6	6	1	2.883628
7	7	0	2.303833
8	8	1	3.286325

wqsreg -Syntax

```
wqsreg yvar, mixture(varlist) ///
[boot(integer 100) validation(integer 60) q(integer 4) b1_neg(integer 0)
cvar(varlist) seed(integer 0)
conv_maxiter(integer 2000) conv_vtol(real 0.000000001) technique(string ‘bfgs’)
model_fam(string 'Linear’)
savingWQSindex(string) figureName(string) savingWeights(string)
id(string) rh_rep(integer 1)]
```

Number of repetitions for repeated holdout validation:

- default: rh_rep= 1 (no repeated holdout validation)

wqsreg -Syntax

Name of the dataset that will include the weights, only if you want to save it

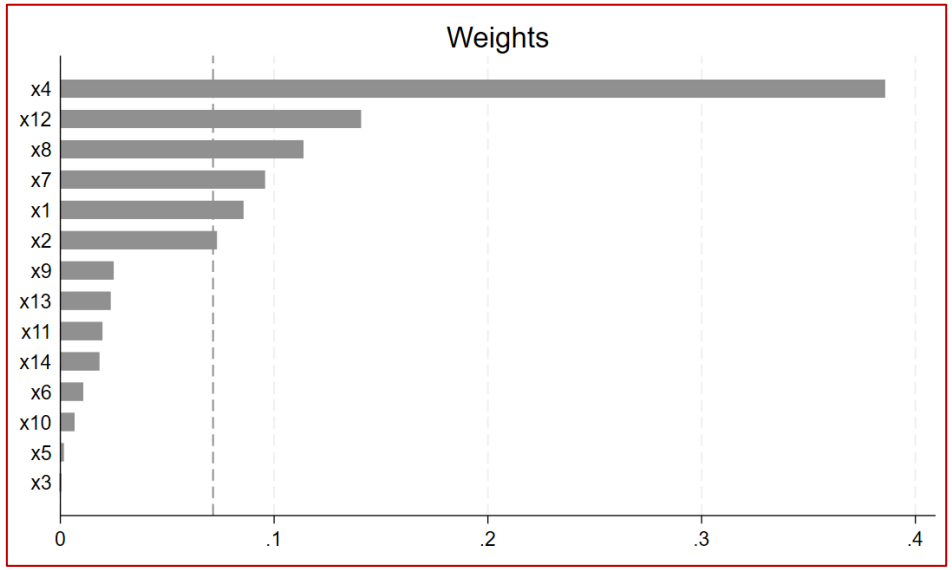
```
wqsreg yvar, mixture(varlist) ///
[boot(integer 100) validation(integer 60) q(integer 4) b1_neg(integer 0)
cvar(varlist) seed(integer 0)
conv_maxiter(integer 2000) conv_vtol(real 0.000000001) technique(string ‘bfgs’)
model_fam(string 'Linear’)
savingWQSindex(string) figureName(string) savingWeights(string)
id(string) rh_rep(integer 1)]
```

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14
1	.0856345	.0732329	.0000635	.3858102	.001562	.010682	.095677	.1137422	.0249356	.0066688	.0196656	.1406262	.0234702	.0182294

wqsreg -Syntax

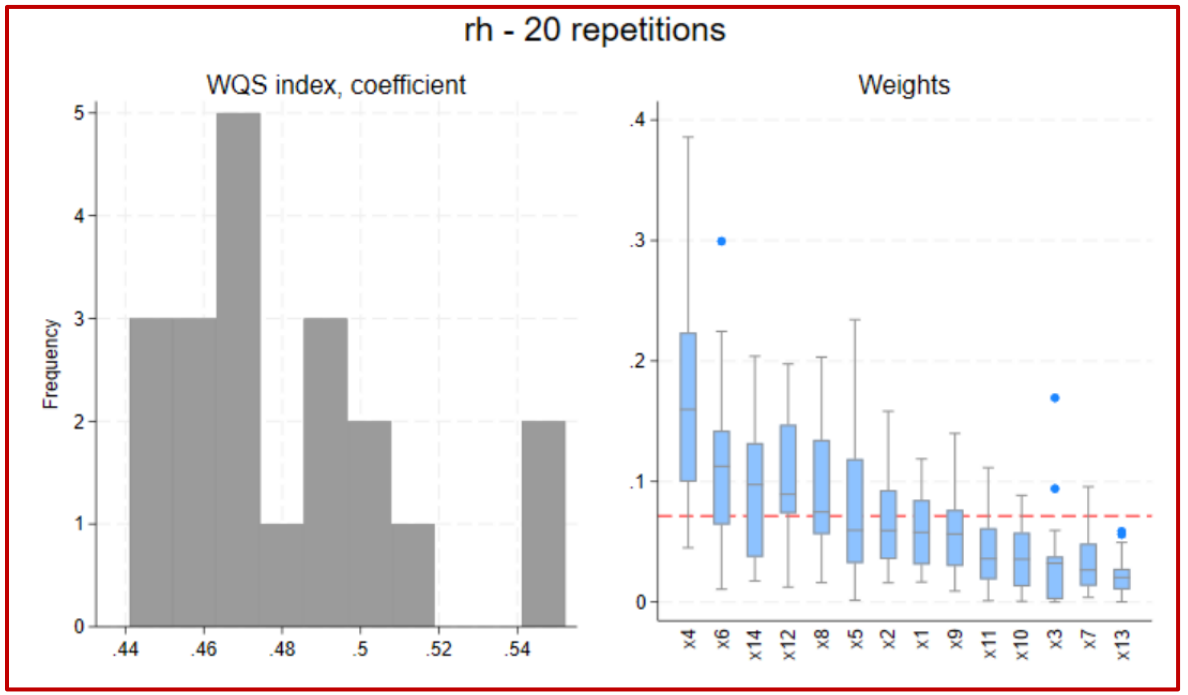
Filename for the plot of weights, only if you want to save it

```
wqsreg yvar, mixture(varlist) ///
[boot(integer 100) validation(integer 60) q(integer 4) b1_neg(integer 0)
cvar(varlist) seed(integer 0)
conv_maxiter(integer 2000) conv_vtol(real 0.000000001) technique(string ‘bfgs’)
model_fam(string 'Linear’)
savingWQSindex(string) figureName(string) savingWeights(string)
id(string) rh_rep(integer 1)]
```



wqsreg -Syntax

```
wqsreg yvar, mixture(varlist) ///
[boot(integer 100) validation(integer 60) q(integer 4) b1_neg(integer 0)
cvar(varlist) seed(integer 0)
conv_maxiter(integer 2000) conv_vtol(real 0.000000001) technique(string ‘bfgs’)
model_fam(string 'Linear’)
savingWQSindex(string) figureName(string) savingWeights(string)
id(string) rh_rep(integer 1)]
```



Possible errors generated by ***wqsreg*** include:

```
. wqsreg y   , mixture(x*) cvar(z1) b1_neg(1)
Note: for a more robust estimation the use of repeated holdouts is recommended
Error: There are no negative b1
```

```
. wqsreg y   , mixture(x*) cvar(z1) b1_neg(2)
Note: for a more robust estimation the use of repeated holdouts is recommended
Error, b1_neg can be 0 (positive b1, the default) or 1 (negative b1)

. wqsreg y   , mixture(x*) cvar(z1) validation(120)
Note: for a more robust estimation the use of repeated holdouts is recommended
Error, Validation must be numeric in [0; 100)

. wqsreg y   , mixture(x*) cvar(z1) boot(0)
Note: for a more robust estimation the use of repeated holdouts is recommended
Error, please insert a postive number of bootstrap samples. Note that boot=1 means no bootstrapping

. wqsreg y   , mixture(x*) cvar(z1) boot(100) savingWQSindex(a)
Note: for a more robust estimation the use of repeated holdouts is recommended
Error, id is required when savingWQSindex is reported

. wqsreg y   , mixture(x*) cvar(z1) id(Obs) boot(100) savingWQSindex(a) savingWeights(a)
Note: for a more robust estimation the use of repeated holdouts is recommended
Error, please provide different names for savingWQSindex and savingWeights
```

Discussion

- The importance of appropriately exploring complex multidimensional exposures, such as environmental mixtures, is increasingly recognized.
- **`wqsreg`** is the first command to apply WQS regression in Stata.

Selected references

- Bellavia A (2025), Statistical Methods for Environmental Mixtures, Springer;
- Carrico C, Gennings C, Wheeler DC, Factor-Litvak P (2015). Characterization of weighted quantile sum regression for highly correlated data in a risk analysis setting. Journal of agricultural, biological, and environmental statistics;
- Czarnota J, Gennings C, Wheeler DC (2015). Assessment of weighted quantile sum regression for modeling chemical mixtures and cancer risk. Cancer informatics;

Selected references

- Ponzano M, Renzetti S, Gennings C, Bellavia A (2026). Wqsreg: a Stata command for weighted quantile sum regression. European Journal of Epidemiology;
- Renzetti S, Curtin P, Just AC, Bello G, Gennings C (2023). Package 'gWQS'.
- Tanner EM, Bornehag C-G, Gennings C (2019). Repeated holdout validation for weighted quantile sum regression. MethodsX.

ponzano.marta@gmail.com
m.ponzano@unilink.it