Statistics/Data Analysis

help binsreg

Title

binsreg —— Data-driven Binscatter Estimation with Robust Inference Procedures and Plots.

Syntax

binsreg *depvar indvar* [*covars*] [*if*] [*in*] [*weight*] [, **deriv(***v***)**
dots(*p s***) dotsgrid(***dotsgridoption***) dotsplotopt(***dotsoption***)**
line(*p s***) linegrid(***#***) lineplotopt(***lineoption***)**
ci(*p s***) cigrid(***cigridoption***) ciplotopt(***rcapoption***)**
cb(*p s***) cbgrid(***#***) cbplotopt(***rareaoption***)**
polyreg(*p***) polyreggrid(***#***) polyregcigrid(***#***) polyregplotopt(***lineoption***)**
by(*varname***) bycolors(***colorstylelist***) bysymbols(***symbolstylelist***)**
bylpatterns(*linepatternstylelist***)**
testmodel(*p s***) testmodelparfit(***filename***) testmodelpoly(***p***)**
testshape(*p s***) testshapel(***numlist***) testshaper(***numlist***) testshape2(***numlist***)**
nbins(*#***) binspos(***position***) binsmethod(***method***) nbinsrot(***#***) samebinsby**
nsims(*#***) simsgrid(***#***) simsseed(***seed***)**
dfcheck(*n1 n2***) masspoints(***masspointsoption***)**
vce(*vcetype***) level(***level***) noplot savedata(***filename***) replace** *twoway_options*
]

where *depvar* is the dependent variable, *indvar* is the independent variable for binning, and *covars* are other covariates to be controlled for.

p, s and v are integers satisfying 0 <= s,v <= p, which can take different values in each case.

fweights, **aweight**s and **pweight**s are allowed; see weight.

Description

binsreg implements binscatter estimation with robust inference proposed and plots, following the results in Cattaneo, Crump, Farrell and Feng (2019a). Binscatter provides a flexible way of describing the mean relationship between two variables, after possibly adjusting for other covariates, based on partitioning/binning of the independent variable of interest. The main purpose of this command is to generate binned scatter plots with curve estimation with robust pointwise confidence intervals and uniform confidence band. If the binning scheme is not set by the user, the companion command binsregselect is used to implement binscatter in a data-driven (optimal) way. Hypothesis testing about the regression function can also be conducted via the companion command binsregtest.

A detailed introduction to this command is given in Cattaneo, Crump, Farrell and Feng (2019b). A companion R package with the same capabilities is available (see website below).

Companion commands: binsregtest for hypothesis testing, and binsregselect data-driven (optimal) binning selection.

Related Stata and R packages are available in the following website:

https://sites.google.com/site/nppackages/

Options

Estimand

deriv(*v***)** specifies the derivative order of the regression function for estimation, testing and plotting. The default is **deriv(0)**, which corresponds to the function itself.

Dots

dots(*p s***)** sets a piecewise polynomial of degree *p* with *s* smoothness constraints for point estimation and plotting as "dots". The default is **dots(0 0)**, which corresponds to piecewise constant (canonical binscatter).

dotsgrid(*dotsgridoption***)** specifies the number and location of dots within each bin to be plotted. Two options are available: *mean* and a *numeric* non-negative integer. The option **dotsgrid(***mean***)** adds the sample average of *indvar* within each bin to the grid of evaluation points. The option **dotsgrid(***#***)** adds *#* number of evenly-spaced points to the grid of evaluation points for each bin. Both options can be used simultaneously: for example, **dotsgrid(***mean 5***)** generates six evaluation points within each bin containing the sample mean of *indvar* within each bin and five evenly-spaced points. Given this choice, the dots are point estimates evaluated over the selected grid within each bin. The default is **dotsgrid(***mean***)**, which corresponds to one dot per bin evaluated at the sample average of *indvar* within each bin (canonical binscatter).

dotsplotopt(*dotsoption***)** standard graphs options to be passed on to the twoway command to modify the appearance of the plotted dots.

Line

line(*p s***)** sets a piecewise polynomial of degree *p* with *s* smoothness constraints for plotting as a "line". By default, the line is not included in the plot unless explicitly specified. Recommended specification is **line(3 3)**, which adds a cubic B-spline estimate of the regression function of interest to the binned scatter plot.

linegrid(*#***)** specifies the number of evaluation points of an evenly-spaced grid within each bin used for evaluation of the point estimate set by the **line(p s)** option. The default is **linegrid(20)**, which corresponds to 20 evenly-spaced evaluation points within each bin for fitting/plotting the line.

lineplotopt(*lineoption***)** standard graphs options to be passed on to the twoway command to modify the appearance of the plotted line.

Confidence Intervals

ci(*p s***)** specifies the piecewise polynomial of degree *p* with *s* smoothness constraints used for constructing confidence intervals. By default, the confidence intervals are not included in the plot unless explicitly specified. Recommended specification is **ci(3 3)**, which adds confidence intervals based on a cubic B-spline estimate of the regression function of interest to the binned scatter plot.

cigrid(*cigridoption***)** specifies the number and location of evaluation points in the grid used to construct the confidence intervals set by the **ci(***p s***)** option. Two options are available: *mean* and a *numeric* non-negative integer. The option **cigrid(***mean***)** adds the sample average of *indvar* within each bin to the grid of evaluation points. The option **cigrid(***#***)** adds *#* number of evenly-spaced points to the grid of evaluation points for each bin. Both options can be used simultaneously: for example, **cigrid(***mean 5***)** generates six evaluation points within each bin containing the sample mean of *indvar* within each bin and five evenly-spaced points. The default is **cigrid(***mean***)**, which corresponds to one evaluation point set at the sample average of *indvar* within each bin for confidence interval construction.

ciplotopt(*rcapoption***)** standard graphs options to be passed on to the twoway command to modify the appearance of the confidence intervals.

Confidence Band

cb(*p s***)** specifies the piecewise polynomial of degree *p* with *s* smoothness constraints used for constructing the confidence band. By default, the confidence band is not included in the plot unless explicitly specified. Recommended specification is **cb(3 3)**, which adds a confidence band based on a cubic B-spline estimate of the regression function of interest to the binned scatter plot.

cbgrid(*#***)** specifies the number of evaluation points of an evenly-spaced grid within each bin used for evaluation of the point estimate set by the **cb(p s)** option. The default is **cbgrid(20)**, which corresponds to 20 evenly-spaced evaluation points within each bin for confidence band construction.

cbplotopt(*rareaoption***)** standard graphs options to be passed on to the twoway command to modify the appearance of the confidence band.

Global Polynomial Regression

polyreg(*p***)** sets the degree *p* of a global polynomial regression model for plotting. By default, this fit is not included in the plot unless explicitly specified. Recommended specification is **polyreg(3)**, which adds a fourth order global polynomial fit of the regression function of interest to the binned scatter plot.

polyreggrid(*#***)** specifies the number of evaluation points of an evenly-spaced grid within each bin used for evaluation of the point estimate set by the **polyreg(p)** option. The default is **polyreggrid(20)**, which corresponds to 20 evenly-spaced evaluation points within each bin for confidence interval construction.

polyregcigrid(*#***)** specifies the number of evaluation points of an evenly-spaced grid within each bin used for constructing confidence intervals based on polynomial regression set by the **polyreg(p)** option. The default is **polyregcigrid(0)**, which corresponds to not plotting confidence intervals for the global polynomial regression approximation.

polyregplotopt(*lineoption***)** standard graphs options to be passed on to the twoway command to modify the appearance of the global polynomial regression fit.

Subgroup Analysis

by(*varname***)** specifies the variable containing the group indicator to perform subgroup analysis; both numeric and string variables are supported. When **by(***varname***)** is specified, **binsreg** implements estimation and inference by each subgroup separately, but produces a common binned scatter plot. By default, the binning structure is selected for each subgroup separately, but see the option **samebinsby** below for imposing a common binning structure across subgroups.

bycolors(*colorstylelist***)** specifies an ordered list of colors for plotting each subgroup series defined by the option **by()**.

bysymbols(*symbolstylelist***)** specifies an ordered list of symbols for plotting each subgroup series defined by the option **by()**.

bylpatterns(*linepatternstylelist***)** specifies an ordered list of line patterns for plotting each subgroup series defined by the option **by()**.

Parametric Model Specification Testing

testmodel(*p s***)** sets a piecewise polynomial of degree *p* with *s* smoothness constraints for parametric model specification testing, implemented via the companion command binsregtest. The default is **testmodel(3 3)**, which corresponds to a cubic B-spline estimate of the regression function of interest for testing against the fitting from a parametric model specification.

testmodelparfit(*filename***)** specifies a dataset which contains the evaluation grid and fitted values of the model(s) to be tested against. The file must have a variable with the same name as *indvar*, which contains a series of evaluation points at which the binscatter model and the parametric model of interest are compared with each other. Each parametric model is represented by a variable named as *binsreg_fit**, which must contain the fitted values at the corresponding evaluation points.

testmodelpoly(*p***)** specifies the degree of a global polynomial model to be tested against.

Nonparametric Shape Restriction Testing

testshape(*p s***)** sets a piecewise polynomial of degree *p* with *s* smoothness constraints for nonparametric shape restriction testing, implemented via the companion command binsregtest. The default is **testmodel(3 3)**, which corresponds to a cubic B-spline estimate of the regression function of interest for one-sided or two-sided testing.

testshapel(*numlist***)** specifies a numlist of null boundary values for hypothesis testing. Each number *a* in the *numlist* corresponds to one boundary of a one-sided hypothesis test to the left of the form H0: *sup_x mu(x)<=a*.

testshaper(*numlist***)** specifies a numlist of null boundary values for hypothesis testing. Each number *a* in the *numlist* corresponds to one boundary of a one-sided hypothesis test to the right of the form H0: *inf_x mu(x)>=a*.

testshape2(*numlist***)** specifies a numlist of null boundary values for hypothesis testing. Each number *a* in the *numlist* corresponds to one boundary of a two-sided hypothesis test of the form H0: *sup_x |mu(x)-a|=0*.

Partitioning/Binning Selection

nbins(*#***)** sets the number of bins for partitioning/binning of *indvar*. If not specified, the number of bins is selected via the companion command binsregselect in a data-driven, optimal way whenever possible.

binspos(*position***)** specifies the position of binning knots. The default is **binspos(qs)**, which corresponds to quantile-spaced binning (canonical binscatter). Other options are: **es** for evenly-spaced binning, or a numlist for manual specification of the positions of inner knots (which must be within the range of *indvar*).

binsmethod(*method***)** specifies the method for data-driven selection of the number of bins via the companion command binsregselect. The default is **binsmethod(dpi)**, which corresponds to the IMSE-optimal direct plug-in rule. The other option is: **rot** for rule of thumb implementation.

nbinsrot(*#***)** specifies an initial number of bins value used to construct the DPI number of bins selector. If not specified, the data-driven ROT selector is used instead.

samebinsby forces a common partitioning/binning structure across all subgroups specified by the option **by()**. The knots positions are selected according to the option **binspos()** and using the full sample. If **nbins()** is not specified, then the number of bins is selected via the companion command binsregselect and using the full sample.

Simulation

nsims(*#***)** specifies the number of random draws for constructing confidence bands and hypothesis testing. The default is **nsims(500)**, which corresponds to 500 draws from a standard Gaussian random vector of size [(p+1)*J - (J-1)*s].

simsgrid(#) specifies the number of evaluation points of an evenly-spaced grid within each bin used for evaluation of the supremum (or infimum) operation needed to construct confidence bands and hypothesis testing procedures. The default is **simsgrid(20)**, which corresponds to 20 evenly-spaced evaluation points within each bin for approximating the supremum (or infimum) operator.

simsseed(#) sets the seed for simulations.

Mass Points and Degrees of Freedom

dfcheck(*n1 n2***)** sets cutoff values for minimum effective sample size checks, which take into account the number of unique values of *indvar* (i.e., adjusting for the number of mass points), number of clusters, and degrees of freedom of the different statistical models considered. The default is **dfcheck(20 30)**. See Cattaneo, Crump, Farrell and Feng (2019b) for more details.

masspoints(*masspointsoption***)** specifies how mass points in *indvar* are handled. By default, all mass point and degrees of freedom checks are implemented. Available options:
masspoints(*noadjust***)** omits mass point checks and the corresponding effective sample size adjustments.
masspoints(*nolocalcheck***)** omits within-bin mass point and degrees of freedom checks.
masspoints(*off***)** sets **masspoints(***noadjust***)** and **masspoints(***nolocalcheck***)** simultaneously.
masspoints(*veryfew***)** forces the command to proceed as if *indvar* has only a few number of mass points (i.e., distinct values). In other words, forces the command to proceed as if the mass point and degrees of freedom checks were failed.

Other Options

vce(*vcetype***)** specifies the *vcetype* for variance estimation used by the command regress. The default is **vce(robust)**.

level(#) sets the nominal confidence level for confidence interval and confidence band estimation.

noplot omits binscatter plotting.

savedata(*filename***)** specifies a filename for saving all data underlying the binscatter plot (and more).

replace overwrites the existing file when saving the graph data.

twoway options any unrecognized options are appended to the end of the twoway command generating the binned scatter plot.

Examples

Run a binscatter regression and report the plot
```
. binsreg y x w
```

Stored results

Scalars

e(N)	number of observations
e(level)	confidence level
e(dots_p)	degree of polynomial for dots
e(dots_s)	smoothness of polynomial for dots
e(line_p)	degree of polynomial for line
e(line_s)	smoothness of polynomial for line
e(ci_p)	degree of polynomial for confidence interval
e(ci_s)	smoothness of polynomial for confidence interval
e(cb_p)	degree of polynomial for confidence band
e(cb_s)	smoothness of polynomial for confidence band
e(testshape_p)	degree of polynomial for testing shape
e(testshape_s)	smoothness of polynomial for testing shape

e(testmodel_p)	degree of polynomial for testing models
e(testmodel_s)	smoothness of polynomial for testing models
e(testpolyp)	degree of polynomial regression model
e(stat_poly)	statistic for testing global polynomial model
e(pval_poly)	p value for testing global polynomial model
Locals	
e(testvalueL)	values in **testshapel()**
e(testvalueR)	values in **testshaper()**
e(testvalue2)	values in **testshape2()**
e(testvarlist)	varlist found in **testmodel()**
Matrices	
e(N_by)	number of observations for each group
e(Ndist_by)	number of distinct values for each group
e(Nclust_by)	number of clusters for each group
e(nbins_by)	number of bins for each group
e(cval_by)	critical value for each group, used for confidence bands
e(stat_shapeL)	statistics for **testshapel()**
e(pval_shapeL)	p values for **testshapel()**
e(stat_shapeR)	statistics for **testshaper()**
e(pval_shapeR)	p values for **testshaper()**
e(stat_shape2)	statistics for **testshape2()**
e(pval_shape2)	p values for **testshape2()**
e(stat_model)	statistics for **testmodel()**
e(pval_model)	p values for **testmodel()**

References

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2019a. On Binscatter. *arXiv:1902.09608*.

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2019b. Binscatter Regressions. *arXiv:1902.09615*.

Authors

Matias D. Cattaneo, University of Michigan, Ann Arbor, MI. cattaneo@umich.edu.

Richard K. Crump, Federal Reserve Band of New York, New York, NY. richard.crump@ny.frb.org.

Max H. Farrell, University of Chicago, Chicago, IL. max.farrell@chicagobooth.edu.

Yingjie Feng, University of Michigan, Ann Arbor, MI. yjfeng@umich.edu.