help binsregselect

Title

binsregselect —— Data-driven IMSE-Optimal Partitioning/Binning Selection for Binscatter.

Syntax

binsregselect *depvar indvar* [*othercovs*] [*if*] [*in*] [*weight*] [**, deriv(***v***)**
absorb(*absvars***) reghdfeopt(***reghdfe_option***)**
bins(*p s***) binspos(***position***) binsmethod(***method***) nbinsrot(#)**
nbins(*nbinsopt***)**
pselect(*numlist***) sselect(***numlist***)**
simsgrid(#) savegrid(*filename***) replace**
dfcheck(*n1 n2***) masspoints(***masspointsoption***)**
vce(*vcetype***) usegtools(***on/off***) useeffn(#) randcut(#)**]

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

The degree of the piecewise polynomial p, the number of smoothness constraints s, and the derivative order 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

binsregselect implements data-driven procedures for selecting the number of bins for binscatter estimation. The selected number is optimal in minimizing the (asymptotic) integrated mean squared error (IMSE).

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.

Reghdfe

absorb(*absvars***)** specifies categorical variables (or interactions) representing the fixed effects to be absorbed. This is equivalent to including an indicator/dummy variable for each category of each *absvar*. When **absorb()** is specified, the community-contributed command **reghdfe** instead of the command **regress** is used.

reghdfeopt(*reghdfe_option***)** options to be passed on to **reghdfe**. Important: **absorb()** and **vce()** should not be specified within this option.

For more information about the community-contributed command **reghdfe**, please see http://scorreia.com/software/reghdfe/.

Binning/Degree/Smoothness Selection

bins(*p s***)** sets a piecewise polynomial of degree *p* with *s* smoothness constraints for data-driven (IMSE-optimal) selection of the partitioning/binning scheme. The default is **bins(0 0)**, which corresponds to piecewise constant (canonical binscatter).

binspos(*position***)** specifies the position of binning knots. The default is **binspos(qs)**, which corresponds to quantile-spaced binning (canonical binscatter). Other option is **es** for evenly-spaced binning.

binsmethod(*method***)** specifies the method for data-driven selection of the number of bins. 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.

nbins(*nbinsopt***)** sets the number of bins for degree/smoothness selection. If **nbins(T)** is specified, the command selects the number of bins instead, given the specified degree and smoothness. If a numlist with more than one number is specified, the command selects the number of bins within this list.

pselect(*numlist***)** specifies a list of numbers within which the degree of polynomial *p* for point estimation is selected.

sselect(*numlist***)** specifies a list of numbers within which the number of smoothness constraints *s* for point estimation is selected. If not specified, for each value *p* supplied in the option **pselect()**, only the

piecewise polynomial with the maximum smoothness is considered, i.e., *s=p*.

Note: To implement the degree or smoothness selection, in addition to **pselect()** or **sselect()**, **nbins(#)** must be specified.

Evaluation Points Grid Generation

simsgrid(*#***)** specifies the number of evaluation points of an evenly-spaced grid within each bin used for evaluation of the supremum (infimum or Lp metric) 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.

savegrid(*filename***)** specifies a filename for storing the simulation grid of evaluation points. It contains the following variables: *indvar*, which is a sequence of evaluation points used in approximation; all control variables in *othercovs*, which take values of zero for prediction purpose; *binsreg_isknot*, indicating whether the evaluation point is an inner knot; and *binsreg_bin*, indicating which bin the evaluation point belongs to.

replace overwrites the existing file when saving the grid.

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 (2024c) 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 (or **reghdfe** if **absorb()** is specified). The default is **vce(robust)**.

usegtools(*on/off***)** forces the use of several commands in the community-distributed Stata package **gtools** to speed the computation up, if *on* is specified. Default is **usegtools(off)**.

For more information about the package **gtools**, please see https://gtools.readthedocs.io/en/latest/index.html.

useeffn(#) specifies the effective sample size # to be used when computing the (IMSE-optimal) number of bins. This option is useful for extrapolating the optimal number of bins to larger (or smaller) datasets than the one used to compute it.

randcut(#) specifies the upper bound on a uniformly distributed variable used to draw a subsample for bins selection. Observations for which **runiform()<=#** are used. # must be between 0 and 1.

Examples

Setup

```
. sysuse auto
```

Select IMSE-optimal number of bins using DPI-procedure

```
. binsregselect mpg weight foreign
```

Stored results

Scalars

e(N)	number of observations
e(Ndist)	number of distinct values
e(Nclust)	number of clusters
e(deriv)	order of derivative
e(imse_bsq_rot)	bias constant in IMSE, ROT selection
e(imse_var_rot)	variance constant in IMSE, ROT selection
e(imse_bsq_dpi)	bias constant in IMSE, DPI selection
e(imse_var_dpi)	variance constant in IMSE, DPI selection
e(nbinsrot_poly)	ROT number of bins, unregularized
e(nbinsrot_regul)	ROT number of bins, regularized or user-specified

e(nbinsrot_uknot)	ROT number of bins, unique knots
e(nbinsdpi)	DPI number of bins
e(nbinsdpi_uknot)	DPI number of bins, unique knots
e(prot_poly)	ROT degree of polynomial, unregularized
e(prot_regul)	ROT degree of polynomial, regularized or user-specified
e(prot_uknot)	ROT degree of polynomial, unique knots
e(pdpi)	DPI degree of polynomial
e(pdpi_uknot)	DPI degree of polynomial, unique knots
e(srot_poly)	ROT number of smoothness constraints, unregularized
e(srot_regul)	ROT number of smoothness constraints, regularized or user-specified
e(srot_uknot)	ROT number of smoothness constraints, unique knots
e(sdpi)	DPI number of smoothness constraints
e(sdpi_uknot)	DPI number of smoothness constraints, unique knots

Matrices

e(knot)	numlist of knots
e(m_p)	vector of degrees of polynomial
e(m_s)	vector of number of smoothness constraints
e(m_nbinsrot_poly)	ROT number of bins, unregularized, for each pair of degree and smoothness
e(m_nbinsrot_regul)	ROT number of bins, regularized or user-specified, for each pair of degree and smoothness
e(m_nbinsrot_uknot)	ROT number of bins, unique knots, for each pair of degree and smoothness
e(m_nbinsdpi)	DPI number of bins, for each pair of degree and smoothness
e(m_nbinsdpi_uknot)	DPI number of bins, unique knots, for each pair of degree and smoothness
e(m_imse_bsq_rot)	bias constant in IMSE, ROT selection, for each pair of degree and smoothness
e(m_imse_var_rot)	variance constant in IMSE, ROT selection, for each pair of degree and smoothness
e(m_imse_bsq_dpi)	bias constant in IMSE, DPI selection, for each pair of degree and smoothness
e(m_imse_var_dpi)	variance constant in IMSE, DPI selection, for each pair of degree and smoothness

References

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2024a. On Binscatter. American Economic Review 114(5): 1488–1514.

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2024b. Nonlinear Binscatter Methods. Working Paper.

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2024c.

Binscatter Regressions. Working Paper.

Authors

Matias D. Cattaneo, Princeton University, Princeton, NJ.
cattaneo@princeton.edu.

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

Max H. Farrell, UC Santa Barbara, Santa Barbara, CA. mhfarrell@gmail.com.

Yingjie Feng, Tsinghua University, Beijing, China.
fengyingjiepku@gmail.com.