Sтата®
Statistics/Data Analysis

`help binspwc`

Title

binspwc —— Data-Driven Nonparametric Pairwise Group Comparison using Binscatter.

Syntax

> **binspwc** _depvar_ _indvar_ [_othercovs_] [_if_] [_in_] [_weight_] , **by(**_varname_**)** [
>   **estmethod(**_cmdname_**) deriv(**_v_**) at(**_position_**) nolink**
>   **absorb(**_absvars_**) reghdfeopt(**_reghdfe_option_**)**
>   **pwc(**_p s_**) testtype(**_type_**) lp(**_metric_**)**
>   **bins(**_p s_**) bynbins(**_numlist_**) binspos(**_position_**) binsmethod(**_method_**)**
>   **nbinsrot(**_#_**) samebinsby randcut(**_#_**)**
>   **nsims(**_#_**) simsgrid(**_#_**) simsseed(**_seed_**)**
>   **dfcheck(**_n1 n2_**) masspoints(**_masspointsoption_**)**
>   **vce(**_vcetype_**) asyvar(**_on/off_**) usegtools(**_on/off_**)**]

where _depvar_ is the dependent variable, _indvar_ is the independent variable for binning, and _othercovs_ 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

binspwc implements binscatter-based hypothesis testing procedures for pairwise group comparison of binscatter estimators, following the results in Cattaneo, Crump, Farrell and Feng (2021a). 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 and inference procedures are based on robust bias correction. Binned scatter plots based on different models can be constructed using the companion commands binsreg, binsqreg, binslogit and binsprobit.

A detailed introduction to this command is given in Cattaneo, Crump, Farrell and Feng (2021b). Companion R and Python packages with the same capabilities are available (see website below).

Companion commands: binsreg for binscatter least squares regression with robust inference procedures and plots, binsqreg for binscatter quantile regression with robust inference procedures and plots, binslogit for binscatter logit estimation with robust inference procedures and plots, binsprobit for binscatter probit estimation with robust inference procedures and plots, and binsregselect for data-driven (optimal) binning selection.

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

> https://nppackages.github.io/

Options

    ┌─────────────┐
  ─────┘ Estimand └───

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, **binspwc** implements estimation for each subgroup separately and then conduct _all_ pairwise comparison tests. 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. This option is required.

estmethod(*cmdname***)** specifies the binscatter model. The default is **estmethod(reg)**, which corresponds to the binscatter least squares regression. Other options are: **estmethod(qreg #)** for binscatter quantile regression where # is the quantile to be estimated, **estmethod(logit)** for binscatter logistic regression and **estmethod(probit)** for binscatter probit regression.

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.

at(*position***)** specifies the values of *othercovs* at which the estimated function is evaluated for plotting. The default is **at(mean)**, which corresponds to the mean of *othercovs*. Other options are: **at(median)** for the median of *othercovs*, **at(0)** for zeros, and **at(filename)** for particular values of *othercovs* saved in another file.

Note: when **at(mean)** or **at(median)** is specified, all factor variables in *othercovs* (if specified) are excluded from the evaluation (set as zero).

nolink specifies that the function within the inverse link (logistic) function be reported instead of the conditional probability function. This option is used only if logit or probit model is specified in **estmethod()**.

⌐ 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 the command **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/.

⌐ Pairwise Group Comparison Testing ⌐

pwc(*p s***)** sets a piecewise polynomial of degree *p* with *s* smoothness constraints for pairwise group comparison. The default is **pwc(3 3)**, which corresponds to a cubic B-spline estimate of the function of interest for each group.

testtype(*type***)** specifies the type of pairwise comparison test. The default is **testtype(***2***)**, which corresponds to a two-sided test of the form H0: *mu_1(x)=mu_2(x)*. Other options are: **testtype(***l***)** for the one-sided test of the form H0: *mu_1(x)<=mu_2(x)* and **testtype(***r***)** for the one-sided test of the form H0: *mu_1(x)>=mu_2(x)*.

lp(*metric***)** specifies an Lp metric used for a (two-sided) test for the difference between two groups. The default is **lp(inf)**, which corresponds to the sup-norm. Other options are **Lp(q)** for a positive integer **q**.

⌐ Partitioning/Binning 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(2 2)**, which corresponds to a quadratic spline estimate.

bynbins(*numlist***)** sets a numlist of numbers of bins for partitioning/binning of *indvar*, which is applied to the binscatter estimation for each group. The ordering of the group follows the result of tabulate. If a single number of bins is specified, it applies to the estimation for all groups. 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 <u>numlist</u>
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 <u>binsregselect</u>. 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 <u>binsregselect</u>
and using the full sample.

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.

Simulation

nsims(#**)** specifies the number of random draws for 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 (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
(infimum or Lp metric) 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 (2021b) 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 commands
<u>regress</u>, <u>logit</u>, <u>logit</u>, <u>qreg</u> or **reghdfe**. The default is **vce(robust)**.

asyvar(*on/off***)** specifies the method used to compute standard errors. If
asyvar(on) is specified, the standard error of the nonparametric component is
used and the uncertainty related to other control variables *othercovs* is
omitted. Default is **asyvar(off),** that is, the uncertainty related to
othercovs is taken into account.

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.

Examples

Setup
   . sysuse auto

Generate two groups
   . gen group=price>5000

Test for the difference between two groups
   . binspwc mpg weight foreign, by(group)

Stored results

Scalars
  e(N)   number of observations
  e(p)   degree of polynomial for bin selection
  e(s)   smoothness of polynomial for bin selection
  e(pwc_p)   degree of polynomial for testing
  e(pwc_s)   smoothness of polynomial for testing
Macros
  e(byvalue)   name of groups found in **by()**
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(stat)   test statistics for all pairwise comparisons
  e(pval)   p values for all pairwise comparisons

References

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2021a. On Binscatter.
arXiv:1902.09608.

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2021b. Binscatter
Regressions. *arXiv:1902.09615.*

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, University of Chicago, Chicago, IL. max.farrell@chicagobooth.edu.

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