Package 'binsreg'

March 19, 2019

Type Package

Title Binscatter Estimation and Inference

Date 2019-03-17

Version 0.2.0

Author Matias D. Cattaneo, Richard K. Crump, Max H. Farrell, Yingjie Feng

Maintainer Yingjie Feng <yjfeng@umich.edu>

Description Provides tools for statistical analysis using the binscatter methods developed by Cattaneo, Crump, Farrell and Feng (2019a) <arXiv:1902.09608> and Cattaneo, Crump, Farrell and Feng (2019b) <arXiv:1902.09615>. Binscatter provides a flexible way of describing the mean relationship between two variables based on partitioning/binning of the independent variable of interest. binsreg() implements binscatter estimation and robust (pointwise and uniform) inference of regression functions and derivatives thereof, with particular focus on constructing binned scatter plots. binsregtest() implements hypothesis testing procedures for parametric functional forms of and nonparametric shape restrictions on the regression function. binsregselect() implements data-driven procedures for selecting the number of bins for binscatter estimation. All the commands allow for covariate adjustment, smoothness restrictions and clustering.

Depends R (>= 3.1)

License GPL-2

Encoding UTF-8

LazyData true

Imports ggplot2, sandwich

RoxygenNote 6.1.1

R topics documented:

`binsreg-package` *Binsreg Package Document*

Description

Binscatter provides a flexible, yet parsimonious way of visualizing and summarizing large data sets in regression settings, and has been a popular methodology in applied microeconomics and other social sciences. The binsreg package provides tools for statistical analysis using the binscatter methods developed in Cattaneo, Crump, Farrell and Feng (2019a). `binsreg` implements binscatter estimation with robust inference and plots, including curve estimation, pointwise confidence intervals and uniform confidence band. `binsregtest` implements hypothesis testing procedures for parametric specification of and nonparametric shape restrictions on the unknown regression function. `binsregselect` implements data-driven number of bins selectors for binscatter implementation using either quantile-spaced or evenly-spaced binning/partitioning. All the commands allow for covariate adjustment, smoothness restrictions, and clustering, among other features.

The companion software article, Cattaneo, Crump, Farrell and Feng (2019b), provides further implementation details and empirical illustration. For related Stata and R packages useful for nonparametric data analysis and statistical inference, visit https://sites.google.com/site/nppackages.

Author(s)

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

Richard K. Crump, Federal Reserve Bank 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 (maintainer), University of Michigan, Ann Arbor, MI. <yjfeng@umich.edu>.

References

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2019a: On Binscatter. Working Paper.

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2019b: Binscatter Regressions. Working Paper.

`binsreg` *Data-driven Binscatter Estimation with Robust Inference Procedures and Plots*

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 function 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 function `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 function `binsregtest`.

Usage

```
binsreg(y, x, w = NULL, deriv = 0, dots = c(0, 0), dotsgrid = 0,
  dotsgridmean = T, line = NULL, linegrid = 20, ci = NULL,
  cigrid = 0, cigridmean = T, cb = NULL, cbgrid = 20,
  polyreg = NULL, polyreggrid = 20, polyregcigrid = 0, by = NULL,
  bycolors = NULL, bysymbols = NULL, bylpatterns = NULL,
  legendTitle = NULL, legendoff = F, testmodel = c(3, 3),
  testmodelparfit = NULL, testmodelpoly = NULL, testshape = c(3, 3),
  testshapel = NULL, testshaper = NULL, testshape2 = NULL,
  nbins = NULL, binspos = "qs", binsmethod = "dpi",
  nbinsrot = NULL, samebinsby = F, nsims = 500, simsgrid = 20,
  simsseed = 666, vce = "HC1", cluster = NULL, level = 95,
  noplot = F, dfcheck = c(20, 30), masspoints = "on",
  weights = NULL, subset = NULL)
```

Arguments

y	outcome variable. A vector.
x	independent variable of interest. A vector.
w	control variables. A matrix or a vector.
deriv	derivative order of the regression function for estimation, testing and plotting. The default is deriv=0, which corresponds to the function itself.
dots	a vector. dots=c(p,s) sets a piecewise polynomial of degree p with s smoothness constraints for point estimation and plotting as "dots". The default is dots=c(0,0), which corresponds to piecewise constant (canonical binscatter)
dotsgrid	number of dots within each bin to be plotted. Given the choice, these dots are point estimates evaluated over an evenly-spaced grid within each bin. The default is dotsgrid=0, and only the point estimates at the mean of x within each bin are presented.
dotsgridmean	If true, the dots corresponding to the point estimates evaluated at the mean of x within each bin are presented. By default, they are presented, i.e., dotsgridmean=T.
line	a vector. line=c(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=c(3,3), which adds a cubic B-spline estimate of the regression function of interest to the binned scatter plot.
linegrid	number of evaluation points of an evenly-spaced grid within each bin used for evaluation of the point estimate set by the line=c(p,s) option. The default is linegrid=20, which corresponds to 20 evenly-spaced evaluation points within each bin for fitting/plotting the line.
ci	a vector. ci=c(p,s) sets a 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=c(3,3), which adds confidence intervals based on cubic B-spline estimate of the regression function of interest to the binned scatter plot.
cigrid	number of evaluation points of an evenly-spaced grid within each bin used for evaluation of the point estimate set by the ci=c(p,s) option. The default is cigrid=1, which corresponds to 1 evenly-spaced evaluation point within each bin for confidence interval construction.

`cigridmean` If true, the confidence intervals corresponding to the point estimates evaluated at the mean of `x` within each bin are presented. The default is `cigridmean=T`.

`cb` a vector. `cb=c(p,s)` sets a 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=c(3,3)`, which adds a confidence band based on cubic B-spline estimate of the regression function of interest to the binned scatter plot.

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

`polyreg` degree 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 cubic (global) polynomial fit of the regression function of interest to the binned scatter plot.

`polyreggrid` 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` 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.

`by` a vector containing the group indicator for subgroup analysis; both numeric and string variables are supported. When `by` 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` an ordered list of colors for plotting each subgroup series defined by the option `by`.

`bysymbols` an ordered list of symbols for plotting each subgroup series defined by the option `by`.

`bylpatterns` an ordered list of line patterns for plotting each subgroup series defined by the option `by`.

`legendTitle` String, title of legend.

`legendoff` If true, no legend is added.

`testmodel` a vector. `testmodel=c(p,s)` sets a piecewise polynomial of degree `p` with `s` smoothness constraints for parametric model specification testing. The default is `testmodel=c(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` a data frame or matrix which contains the evaluation grid and fitted values of the model(s) to be tested against. The first column 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 other columns, which must contain the fitted values at the corresponding evaluation points.

`testmodelpoly` degree of a global polynomial model to be tested against.

`testshape` a vector. `testshape=c(p,s)` sets a piecewise polynomial of degree `p` with `s` smoothness constraints for nonparametric shape restriction testing. The default is `testshape=c(3,3)`, which corresponds to a cubic B-spline estimate of the regression function of interest for one-sided or two-sided testing.

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

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

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

`nbins` number of bins for partitioning/binning of `x`. If not specified, the number of bins is selected via the companion function `binsregselect` in a data-driven, optimal way whenever possible.

`binspos` position of binning knots. The default is `binspos="qs"`, which corresponds to quantile-spaced binning (canonical binscatter). The other options are `"es"` for evenly-spaced binning, or a vector for manual specification of the positions of inner knots (which must be within the range of `x`).

`binsmethod` 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` 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` if true, a common partitioning/binning structure across all subgroups specified by the option `by` is forced. 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.

`nsims` 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` 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` seed for simulation.

`vce` Procedure to compute the variance-covariance matrix estimator. Options are

- `"const"` homoskedastic variance estimator.
- `"HC0"` heteroskedasticity-robust plug-in residuals variance estimator without weights.
- `"HC1"` heteroskedasticity-robust plug-in residuals variance estimator with hc1 weights. Default.
- `"HC2"` heteroskedasticity-robust plug-in residuals variance estimator with hc2 weights.

- "HC3" heteroskedasticity-robust plug-in residuals variance estimator with hc3 weights.

cluster — cluster ID. Used for compute cluster-robust standard errors.

level — nominal confidence level for confidence interval and confidence band estimation. Default is level=95.

noplot — If true, no plot produced.

dfcheck — adjustments for minimum effective sample size checks, which take into account number of unique values of x (i.e., number of mass points), number of clusters, and degrees of freedom of the different stat models considered. The default is dfcheck=c(20, 30). See Cattaneo, Crump, Farrell and Feng (2019b) for more details.

masspoints — how mass points in x are handled. Available options:

- "on" all mass point and degrees of freedom checks are implemented. Default.
- "noadjust" mass point checks and the corresponding effective sample size adjustments are omitted.
- "nolocalcheck" within-bin mass point and degrees of freedom checks are omitted.
- "off" "noadjust" and "nolocalcheck" are set simultaneously.
- "veryfew" forces the function to proceed as if x has only a few number of mass points (i.e., distinct values). In other words, forces the function to proceed as if the mass point and degrees of freedom checks were failed.

weights — an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. For more details, see lm.

subset — Optional rule specifying a subset of observations to be used.

Value

bins_plot — A ggplot object for binscatter plot.

data.plot — A list containing data for plotting. Each item is a sublist of data frames for each group. Each sublist may contain the following data frames:

- data.dots Data for dots. It contains: x, evaluation points; bin, the indicator of bins; isknot, indicator of inner knots; mid, midpoint of each bin; and fit, fitted values.
- data.line Data for line. It contains: x, evaluation points; bin, the indicator of bins; isknot, indicator of inner knots; mid, midpoint of each bin; and fit, fitted values.
- data.ci Data for CI. It contains: x, evaluation points; bin, the indicator of bins; isknot, indicator of inner knots; mid, midpoint of each bin; ci.l and ci.r, left and right boundaries of each confidence intervals.
- data.cb Data for CB. It contains: x, evaluation points; bin, the indicator of bins; isknot, indicator of inner knots; mid, midpoint of each bin; cb.l and cb.r, left and right boundaries of the confidence band.
- data.poly Data for polynomial regression. It contains: x, evaluation points; bin, the indicator of bins; isknot, indicator of inner knots; mid, midpoint of each bin; and fit, fitted values.
- data.polyci Data for confidence intervals based on polynomial regression. It contains: x, evaluation points; bin, the indicator of bins; isknot, indicator of inner knots; mid, midpoint of each bin; polyci.l and polyci.r, left and right boundaries of each confidence intervals.

cval.by	A vector of critical values for constructing confidence band for each group.
test	Return of binsregtest.
opt	A list containing options passed to the function, as well as N.by (total sample size for each group), Ndist.by (number of distinct values in x for each group), Nclust.by (number of clusters for each group), and nbins.by (number of bins for each group), and byvals (number of distinct values in by).

Author(s)

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

Richard K. Crump, Federal Reserve Bank 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 (maintainer), University of Michigan, Ann Arbor, MI. <yjfeng@umich.edu>.

References

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2019a: On Binscatter. Working Paper.

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2019b: Binscatter Regressions. Working Paper.

See Also

binsregselect, binsregtest.

Examples

```
x <- runif(500); y <- sin(x)+rnorm(500)
## Binned scatterplot
binsreg(y,x)
```

---

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

Description

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

Usage

```
binsregselect(y, x, w = NULL, deriv = 0, bins = c(0, 0),
  binspos = "qs", binsmethod = "dpi", nbinsrot = NULL,
  simsgrid = 20, savegrid = F, vce = "HC1", useeffn = NULL,
  cluster = NULL, dfcheck = c(20, 30), masspoints = "on",
  weights = NULL, subset = NULL, norotnorm = F, numdist = NULL,
  numclust = NULL)
```

Arguments

y	outcome variable. A vector.
x	independent variable of interest. A vector.
w	control variables. A matrix or a vector.
deriv	derivative order of the regression function for estimation, testing and plotting. The default is deriv=0, which corresponds to the function itself.
bins	a vector. bins=c(p,s) set 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=c(0, 0), which corresponds to piecewise constant (canonical binscatter).
binspos	position of binning knots. The default is binspos="qs", which corresponds to quantile-spaced binning (canonical binscatter). The other options is "es" for evenly-spaced binning.
binsmethod	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	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.
simsgrid	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.
savegrid	If true, a data frame produced containing grid.
vce	procedure to compute the variance-covariance matrix estimator. Options are • "const" homoskedastic variance estimator. • "HC0" heteroskedasticity-robust plug-in residuals variance estimator without weights. • "HC1" heteroskedasticity-robust plug-in residuals variance estimator with hc1 weights. Default. • "HC2" heteroskedasticity-robust plug-in residuals variance estimator with hc2 weights. • "HC3" heteroskedasticity-robust plug-in residuals variance estimator with hc3 weights.
useeffn	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.
cluster	cluster ID. Used for compute cluster-robust standard errors.
dfcheck	adjustments for minimum effective sample size checks, which take into account number of unique values of x (i.e., number of mass points), number of clusters, and degrees of freedom of the different stat models considered. The default is dfcheck=c(20, 30). See Cattaneo, Crump, Farrell and Feng (2019b) for more details.
masspoints	how mass points in x are handled. Available options: • "on" all mass point and degrees of freedom checks are implemented. Default.

- "noadjust" mass point checks and the corresponding effective sample size adjustments are omitted.
- "nolocalcheck" within-bin mass point and degrees of freedom checks are omitted.
- "off" "noadjust" and "nolocalcheck" are set simultaneously.
- "veryfew" forces the function to proceed as if x has only a few number of mass points (i.e., distinct values). In other words, forces the function to proceed as if the mass point and degrees of freedom checks were failed.

weights	an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. For more details, see lm.
subset	optional rule specifying a subset of observations to be used.
norotnorm	if true, a uniform density rather than normal density used for ROT selection.
numdist	number of distinct for selection. Used to speed up computation.
numclust	number of clusters for selection. Used to speed up computation.

Value

nbinsrot.poly	ROT number of bins, unregularized.
nbinsrot.regul	ROT number of bins, regularized.
nbinsrot.uknot	ROT number of bins, unique knots.
nbinsdpi	DPI number of bins.
nbinsdpi.uknot	DPI number of bins, unique knots.
opt	A list containing options passed to the function, as well as total sample size n, number of distinct values Ndist in x, and number of clusters Nclust.
data.grid	A data frame containing grid.

Author(s)

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

Richard K. Crump, Federal Reserve Bank 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 (maintainer), University of Michigan, Ann Arbor, MI. <yjfeng@umich.edu>.

References

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2019a: On Binscatter. Working Paper.

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2019b: Binscatter Regressions. Working Paper.

See Also

binsreg, binsregtest.

Examples

```
x <- runif(500); y <- sin(x)+rnorm(500)
est <- binsregselect(y,x)
summary(est)
```

`binsregtest` *Data-driven Nonparametric Shape Restriction and Parametric Model Specification Testing using Binscatter*

Description

`binsregtest` implements binscatter-based hypothesis testing procedures for parametric functional forms of and nonparametric shape restrictions on the regression function estimators, following the results in Cattaneo, Crump, Farrell and Feng (2019a). If the binning scheme is not set by the user, the companion function `binsregselect` is used to implement binscatter in a data-driven (optimal) way and inference procedures are based on robust bias correction. Binned scatter plots can be constructed using the companion function `binsreg`.

Usage

```
binsregtest(y, x, w = NULL, deriv = 0, testmodel = c(3, 3),
  testmodelparfit = NULL, testmodelpoly = NULL, testshape = c(3, 3),
  testshapel = NULL, testshaper = NULL, testshape2 = NULL,
  bins = c(0, 0), nbins = NULL, binspos = "qs", binsmethod = "dpi",
  nbinsrot = NULL, nsims = 500, simsgrid = 20, simsseed = 666,
  vce = "HC1", cluster = NULL, dfcheck = c(20, 30),
  masspoints = "on", weights = NULL, subset = NULL, numdist = NULL,
  numclust = NULL)
```

Arguments

`y`	outcome variable. A vector.
`x`	independent variable of interest. A vector.
`w`	control variables. A matrix or a vector.
`deriv`	derivative order of the regression function for estimation, testing and plotting. The default is `deriv=0`, which corresponds to the function itself.
`testmodel`	a vector. `testmodel=c(p,s)` sets a piecewise polynomial of degree `p` with `s` smoothness constraints for parametric model specification testing. The default is `testmodel=c(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`	a data frame or matrix which contains the evaluation grid and fitted values of the model(s) to be tested against. The column 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 other columns, which must contain the fitted values at the corresponding evaluation points.
`testmodelpoly`	degree of a global polynomial model to be tested against.
`testshape`	a vector. `testshape=c(p,s)` sets a piecewise polynomial of degree `p` with `s` smoothness constraints for nonparametric shape restriction testing. The default is `testshape=c(3,3)`, which corresponds to a cubic B-spline estimate of the regression function of interest for one-sided or two-sided testing.

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

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

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

`bins` Degree and smoothness for bin selection.

`nbins` number of bins for partitioning/binning of `x`. If not specified, the number of bins is selected via the companion function `binsregselect` in a data-driven, optimal way whenever possible.

`binspos` position of binning knots. The default is `binspos="qs"`, which corresponds to quantile-spaced binning (canonical binscatter). The other options are `"es"` for evenly-spaced binning, or a vector for manual specification of the positions of inner knots (which must be within the range of `x`).

`binsmethod` 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` 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.

`nsims` 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` 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` seed for simulation.

`vce` Procedure to compute the variance-covariance matrix estimator. Options are

- `"const"` homoskedastic variance estimator.
- `"HC0"` heteroskedasticity-robust plug-in residuals variance estimator without weights.
- `"HC1"` heteroskedasticity-robust plug-in residuals variance estimator with hc1 weights. Default.
- `"HC2"` heteroskedasticity-robust plug-in residuals variance estimator with hc2 weights.
- `"HC3"` heteroskedasticity-robust plug-in residuals variance estimator with hc3 weights.

`cluster` cluster ID. Used for compute cluster-robust standard errors.

`dfcheck` adjustments for minimum effective sample size checks, which take into account number of unique values of `x` (i.e., number of mass points), number of clusters, and degrees of freedom of the different stat models considered. The default is `dfcheck=c(20, 30)`. See Cattaneo, Crump, Farrell and Feng (2019b) for more details.

masspoints	how mass points in x are handled. Available options: • "on" all mass point and degrees of freedom checks are implemented. Default. • "noadjust" mass point checks and the corresponding effective sample size adjustments are omitted. • "nolocalcheck" within-bin mass point and degrees of freedom checks are omitted. • "off" "noadjust" and "nolocalcheck" are set simultaneously. • "veryfew" forces the function to proceed as if x has only a few number of mass points (i.e., distinct values). In other words, forces the function to proceed as if the mass point and degrees of freedom checks were failed.
weights	an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. For more details, see lm.
subset	Optional rule specifying a subset of observations to be used.
numdist	Number of distinct for selection. Used to speed up computation.
numclust	Number of clusters for selection. Used to speed up computation.

Value

testshapeL	Results for testshapel, including: testvalL, null boundary values; stat.shapeL, test statistics; and pval.shapeL, p-value.
testshapeR	Results for testshaper, including: testvalR, null boundary values; stat.shapeR, test statistics; and pval.shapeR, p-value.
testshape2	Results for testshape2, including: testval2, null boundary values; stat.shape2, test statistics; and pval.shape2, p-value.
testpoly	Results for testmodelpoly, including: testpoly, the degree of global polynomial; stat.poly, test statistic; pval.poly, p-value.
testmodel	Results for testmodelparfit, including: stat.model, test statistics; pval.model, p-values.
opt	A list containing options passed to the function, as well as total sample size n, number of distinct values Ndist in x, number of clusters Nclust, and number of bins nbins.

Author(s)

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

Richard K. Crump, Federal Reserve Bank 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 (maintainer), University of Michigan, Ann Arbor, MI. <yjfeng@umich.edu>.

References

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2019a: On Binscatter. Working Paper.

Cattaneo, M. D., R. K. Crump, M. H. Farrell, and Y. Feng. 2019b: Binscatter Regressions. Working Paper.

See Also

binsreg, binsregselect.

Examples

```
x <- runif(500); y <- sin(x)+rnorm(500)
est <- binsregtest(y,x, testmodelpoly=1)
summary(est)
```

Index