Package 'RDHonest'

July 5, 2022

Title Honest Inference in Regression Discontinuity Designs

Version 0.4.0.9000

Description Honest and nearly-optimal confidence intervals in fuzzy and sharp regression discontinuity designs and for inference at a point based on local polynomial regression.

Depends R (>= 3.3.0)

License GPL-3

Encoding UTF-8

LazyData true

Imports stats

Suggests spelling,
ggplot2,
testthat,
knitr,
rmarkdown,
Formula,
formatR

RoxygenNote 7.2.0

URL https://github.com/kolesarm/RDHonest

VignetteBuilder knitr

Language en-US

BugReports https://github.com/kolesarm/RDHonest/issues

R topics documented:

`cghs` *Oreopoulos (2006) UK general household survey dataset*

Description

Oreopoulos (2006) UK general household survey dataset

Usage

```
cghs
```

Format

A data frame with 73,954 rows and 2 variables:

earnings Annual earnings in 1998 (UK pounds)

yearat14 Year individual turned 14

Source

American Economic Review data archive, doi:10.1257/000282806776157641

References

Philip Oreopoulos. Estimating average and local average treatment effects when compulsory education schooling laws really matter. American Economic Review, 96(1):152–175, 2006. doi:10.1257/000282806776157641

`CVb` *Critical values for CIs based on a biased Gaussian estimator.*

Description

Computes the critical value $cv_{1-\alpha}(B)$ such that the confidence interval $X \pm cv_{1-\alpha}(B)$ has coverage $1-\alpha$, where X is normally distributed with variance equal to 1 and maximum bias at most B.

Usage

```
CVb(B, alpha = 0.05)
```

Arguments

`B`	Maximum bias, vector of non-negative numbers.
`alpha`	Determines CI level, $1-\alpha$. Scalar between 0 and 1.

Value

Vector of critical values, one for each value of maximum bias supplied by `B`.

Examples

```
## 90% critical value:
CVb(B = 1, alpha = 0.1)
## Usue 95% critical value
CVb(0)
## Returns vector with 3 critical values
CVb(B = c(0, 0.5, 1), alpha = 0.05)
```

`EqKern` *Equivalent kernel for local linear regression.*

Description

Calculates equivalent kernel for local polynomial regression.

Usage

```
EqKern(kernel = "uniform", boundary = TRUE, order = 0)
```

Arguments

kernel	Kernel type. Can be a function supported on $[0, 1]$ (boundary kernel) or $[-1, 1]$ (interior kernel), or else one of "triangular" ($k(u) = (1-\|u\|)_+$), "epanechnikov" ($k(u) = (3/4)(1 - u^2)_+$), or "uniform" ($k(u) = (\|u\| < 1)/2$).
boundary	Logical scalar, specifying whether we are at a boundary.
order	Order of local polynomial: 0 means local constant, 1 local linear, 2 local quadratic etc.

Value

Equivalent kernel function.

Examples

```
EqKern(kernel = "uniform", order = 2)
```

FRDHonest *Honest inference in fuzzy RD*

Description

Calculate estimators and one- and two-sided CIs based on local polynomial estimator in fuzzy RD under second-order Taylor or Hölder smoothness class.

Usage

```
FRDHonest(
  formula,
  data,
  subset,
  weights,
  cutoff = 0,
  M,
  kern = "triangular",
  na.action,
  opt.criterion,
  h,
  se.method = "nn",
  alpha = 0.05,
  beta = 0.8,
  J = 3,
  sclass = "H",
  order = 1,
  se.initial = "EHW",
  T0 = 0
)
```

Arguments

`formula` — object of class `"formula"` (or one that can be coerced to that class) of the form `outcome ~ running_variable`

`data` — optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the outcome and running variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which the function is called.

`subset` — optional vector specifying a subset of observations to be used in the fitting process.

`weights` — Optional vector of weights to weight the observations (useful for aggregated data). Disregarded if optimal kernel is used.

`cutoff` — specifies the RD cutoff in the running variable.

`M` — Bound on second derivative of the conditional mean function.

`kern` — specifies kernel function used in the local regression. It can either be a string equal to `"triangular"` ($k(u) = (1-|u|)_+$), `"epanechnikov"` ($k(u) = (3/4)(1-u^2)_+$), or `"uniform"` ($k(u) = (|u| < 1)/2$), or else a kernel function.

`na.action` — function which indicates what should happen when the data contain `NA`s. The default is set by the `na.action` setting of `options` (usually `na.omit`).

`opt.criterion` — Optimality criterion that bandwidth is designed to optimize. The options are:

- `"MSE"` Finite-sample maximum MSE
- `"FLCI"` Length of (fixed-length) two-sided confidence intervals.
- `"OCI"` Given quantile of excess length of one-sided confidence intervals

The methods use conditional variance given by `sigma2`, if supplied. Otherwise, for the purpose of estimating the optimal bandwidth, conditional variance is estimated using the method specified by `se.initial`.

`h` — bandwidth, a scalar parameter. If not supplied, optimal bandwidth is computed according to criterion given by `opt.criterion`.

`se.method` — Vector with methods for estimating standard error of estimate. If `NULL`, standard errors are not computed. The elements of the vector can consist of the following methods:

- **"nn"** Nearest neighbor method
- **"EHW"** Eicker-Huber-White, with residuals from local regression (local polynomial estimators only).
- **"demeaned"** Like EHW, but instead of using the regression residuals, estimate σ_i^2 by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope). Local polynomial estimators only.
- **"supplied.var"** Use conditional variance supplied by `sigma2` or `d` instead of computing residuals

`alpha` — determines confidence level, `1-alpha` for constructing/optimizing confidence intervals.

`beta` — Determines quantile of excess length to optimize, if bandwidth optimizes given quantile of excess length of one-sided confidence intervals; otherwise ignored.

J
: Number of nearest neighbors, if "nn" is specified in `se.method`.

sclass
: Smoothness class, either `"T"` for Taylor or `"H"` for Hölder class.

order
: Order of local regression 1 for linear, 2 for quadratic.

se.initial
: Method for estimating initial variance for computing optimal bandwidth. Except for `"nn"`, all methods assume homoskedasticity on either side of cutoff (for RD), or for all data (for inference at a point).

  "EHW" Based on residuals from a local linear regression using a triangular kernel, and a bandwidth given by a rule-of-thumb bandwidth suggested by Fan and Gijbels (1996) (for inference at a point), or Imbens and Kalyanaraman (2012, IK) bandwidth (for fuzzy and sharp RD). For fuzzy RD, the IK bandwidth is based on the reduced-form regression.

  "demeaned" Like EHW, but instead of using the regression residuals, estimate σ_i^2 by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope).

  "Silverman" Use residuals from local constant regression with uniform kernel and bandwidth selected using Silverman's rule of thumb, as in Equation (14) in Imbens and Kalyanaraman (2012)

  "SilvermanNN" Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals.

  "nn" Use nearest neighbor estimates, without assuming homoskedasticity

T0
: Initial estimate of the treatment effect for calculating the optimal bandwidth. Only relevant for Fuzzy RD.

Details

The bandwidth is calculated to be optimal for a given performance criterion, as specified by `opt.criterion`. Alternatively, the bandwidth can be specified by `h`.

Value

Returns an object of class `"NPRResults"`. The function `print` can be used to obtain and print a summary of the results. An object of class `"NPRResults"` is a list containing the following components

`estimate` Point estimate. This estimate is MSE-optimal if `opt.criterion="MSE"`

`lff` Not relevant for fuzzy RD.

`maxbias` Maximum bias of `estimate`

`sd` Standard deviation of estimate

`lower`, `upper` Lower (upper) end-point of a one-sided CI based on `estimate`. This CI is optimal if `opt.criterion=="OCI"`

`hl` Half-length of a two-sided CI based on `estimate`, so that the CI is given by `c(estimate-hl, estimate+hl)`. The CI is optimal if `opt.criterion="FLCI"`

`eff.obs` Effective number of observations used by `estimate`

`h` Bandwidth used

`naive` Coverage of CI that ignores bias and uses `qnorm(1-alpha/2)` as critical value

`call` the matched call

`fs` Estimate of the first-stage coefficient

Note

subset is evaluated in the same way as variables in formula, that is first in data and then in the environment of formula.

References

Armstrong, Timothy B., and Michal Kolesár. 2018. "Optimal Inference in a Class of Regression Models." Econometrica 86 (2): 655–83.

Armstrong, Timothy B., and Michal Kolesár. 2020. "Simple and Honest Confidence Intervals in Nonparametric Regression." Quantitative Economics 11 (1): 1–39.

Examples

```
FRDHonest(cn~retired | elig_year, data=rcp, cutoff=0, M=c(5, 0.5),
          kern="triangular", opt.criterion="MSE", T0=0)
```

---

headst *Head Start data from Ludwig and Miller (2007)*

Description

Subset of Ludwig-Miller data. Counties with missing poverty rate, or with both outcomes missing (hs and mortality) were removed. In the original dataset, Yellowstone County, MT (oldcode = 27056) was entered twice, here the duplicate is removed. Yellowstone National Park, MT (oldcode = 27057) is also removed due to it being an outlier for both outcomes. Counties with oldcode equal to (3014, 32032, 47010, 47040, 47074, 47074, 47078, 47079, 47096) matched more than one FIPS entry, so the county labels may not be correct. Mortality data is missing for Alaska.

Usage

```
headst
```

Format

A data frame with 3,127 rows and 9 variables:

statefp State FIPS code

countyfp County FIPS code

oldcode ID in Ludwig-Miller dataset

povrate60 Poverty rate in 1960 relative to 300th poorest county (which had poverty rate 59.1984)

mortHS Average Mortality rate per 100,000 for children aged 5-9 over 1973–83 due to causes addressed as part of Head Start's health services.

mortInj Average Mortality rate per 100,000 for children aged 5-9 over 1973–83 due to injury.

highSchool High school completion rate in 1990 census, ages 18-24

statepc State postal code

county County name

Source

Douglas Miller's former website, http://web.archive.org/web/20190619165949/http://faculty.econ.ucdavis.edu:80/faculty/dlmiller/statafiles/

References

Jens Ludwig and Douglas L. Miller. Does head start improve children's life chances? Evidence from a regression discontinuity design. Quarterly Journal of Economics, 122(1):159–208, February 2007. doi:10.1162/qjec.122.1.159

`kernC` *Constants for common kernels.*

Description

First four moments of uniform, triangular, and Epanechnikov equivalent kernels. Up to numerical integration precision, these moments are matched by `KernMoment()`. See vignette `lpkernels`

Usage

```
kernC
```

Format

A data frame with 18 rows and 19 variables:

kernel Kernel type.

order Order of local polynomial.

boundary Boundary regression?

mu0, mu1, mu2, mu3, mu4 $\int_X u^j k(u)du$, raw moments

nu0, nu1, nu2, nu3, nu4 $\int_X u^j k^2(u)du$, raw moments of kernel squared

pi0, pi1, pi2, pi3, pi4 $\int_X |u^j k(u)|du$, absolute moments

pMSE constant for pointwise MSE optimal bandwidth, $((p+1)!^2\nu_0/(2(p+1)\mu_{p+1}^2))^{1/(2p+3)}$, see page 67 in Fan and Gijbels (1996)

Source

Computed analytically using symbolic math software

References

Fan , J., and I. Gijbels (1996): Local Polynomial Modelling and Its Applications, Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, New York, NY.

KernMoment *Moments of a kernel.*

Description

Computes moments of a kernel over $X = [0, 1]$ (boundary case), or $X = [-1, 1]$ (interior case),

Usage

```
KernMoment(K, moment = 0, boundary = TRUE, type = "raw")
```

Arguments

`K`	kernel function.
`moment`	order j of moment to compute.
`boundary`	Logical scalar, specifying whether we are at a boundary.
`type`	Type of moment. "raw" computes $\int_X u^j k(u)$, "absolute" computes $\int_X \lvert u^j k(u)\rvert$, and "raw2" computes $\int_X u^j k(u)^2$.

Value

Integral value (a scalar).

Examples

```
KernMoment(function(u) abs(u) < 1, moment = 3, boundary = FALSE)
KernMoment(EqKern(kernel = "triangular", order = 2), moment = 3)
```

lee08 *Lee (2008) US House elections dataset*

Description

Lee (2008) US House elections dataset

Usage

```
lee08
```

Format

A data frame with 6,558 rows and 2 variables:

voteshare Vote share in next election

margin Democratic margin of victory

Source

Mostly Harmless Econometrics data archive, https://economics.mit.edu/faculty/angrist/data1/mhe

References

David S. Lee. Randomized experiments from non-random selection in U.S. House elections. Journal of Econometrics, 142(2):675–697, 2008. doi:10.1016/j.jeconom.2007.05.004

`LPPHonest` *Honest inference at a point*

Description

Calculate estimators and one- and two-sided honest CIs for value of conditional mean at a point based on a local polynomial estimator under second-order Taylor or Hölder smoothness class.

Usage

```
LPPHonest(
  formula,
  data,
  subset,
  weights,
  point = 0,
  M,
  kern = "triangular",
  na.action,
  opt.criterion,
  h,
  se.method = "nn",
  alpha = 0.05,
  beta = 0.8,
  J = 3,
  sclass = "H",
  order = 1,
  se.initial = "EHW"
)
```

Arguments

`formula`	object of class `"formula"` (or one that can be coerced to that class) of the form `outcome ~ independent_variable`
`data`	optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the outcome and independent variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which the function is called.

`subset` optional vector specifying a subset of observations to be used in the fitting process.

`weights` Optional vector of weights to weight the observations (useful for aggregated data).

`point` specifies the point x_0 at which to calculate the conditional mean

`M` Bound on second derivative of the conditional mean function.

`kern` specifies kernel function used in the local regression. It can either be a string equal to `"triangular"` ($k(u) = (1-|u|)_+$), `"epanechnikov"` ($k(u) = (3/4)(1-u^2)_+$), or `"uniform"` ($k(u) = (|u| < 1)/2$), or else a kernel function.

`na.action` function which indicates what should happen when the data contain `NA`s. The default is set by the `na.action` setting of `options` (usually `na.omit`).

`opt.criterion` Optimality criterion that bandwidth is designed to optimize. The options are:

- `"MSE"` Finite-sample maximum MSE
- `"FLCI"` Length of (fixed-length) two-sided confidence intervals.
- `"OCI"` Given quantile of excess length of one-sided confidence intervals

The methods use conditional variance given by `sigma2`, if supplied. Otherwise, for the purpose of estimating the optimal bandwidth, conditional variance is estimated using the method specified by `se.initial`.

`h` bandwidth, a scalar parameter. If not supplied, optimal bandwidth is computed according to criterion given by `opt.criterion`.

`se.method` Vector with methods for estimating standard error of estimate. If `NULL`, standard errors are not computed. The elements of the vector can consist of the following methods:

- **"nn"** Nearest neighbor method
- **"EHW"** Eicker-Huber-White, with residuals from local regression (local polynomial estimators only).
- **"demeaned"** Like EHW, but instead of using the regression residuals, estimate σ_i^2 by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope). Local polynomial estimators only.
- **"supplied.var"** Use conditional variance supplied by `sigma2` or `d` instead of computing residuals

`alpha` determines confidence level, `1-alpha` for constructing/optimizing confidence intervals.

`beta` Determines quantile of excess length to optimize, if bandwidth optimizes given quantile of excess length of one-sided confidence intervals; otherwise ignored.

`J` Number of nearest neighbors, if "nn" is specified in `se.method`.

`sclass` Smoothness class, either `"T"` for Taylor or `"H"` for Hölder class.

`order` Order of local regression 1 for linear, 2 for quadratic.

`se.initial` Method for estimating initial variance for computing optimal bandwidth. Except for `"nn"`, all methods assume homoskedasticity on either side of cutoff (for RD), or for all data (for inference at a point).

"EHW" Based on residuals from a local linear regression using a triangular kernel, and a bandwidth given by a rule-of-thumb bandwidth suggested by Fan and Gijbels (1996) (for inference at a point), or Imbens and Kalyanaraman (2012, IK) bandwidth (for fuzzy and sharp RD). For fuzzy RD, the IK bandwidth is based on the reduced-form regression.

"demeaned" Like EHW, but instead of using the regression residuals, estimate σ_i^2 by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope).

"Silverman" Use residuals from local constant regression with uniform kernel and bandwidth selected using Silverman's rule of thumb, as in Equation (14) in Imbens and Kalyanaraman (2012)

"SilvermanNN" Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals.

"nn" Use nearest neighbor estimates, without assuming homoskedasticity

Details

The bandwidth is calculated to be optimal for a given performance criterion, as specified by `opt.criterion`. Alternatively, the bandwidth can be specified by `h`.

Value

Returns an object of class `"NPResults"`. The function `print` can be used to obtain and print a summary of the results. An object of class `"NPRResults"` is a list containing the following components

`estimate` Point estimate. This estimate is MSE-optimal if `opt.criterion="MSE"`

`lff` Not relevant for inference at a point

`maxbias` Maximum bias of `estimate`

`sd` Standard deviation of estimate

`lower`, `upper` Lower (upper) end-point of a one-sided CI based on `estimate`. This CI is optimal if `opt.criterion="OCI"`

`hl` Half-length of a two-sided CI based on `estimate`, so the CI is `c(estimate-hl, estimate+hl)`. The CI is optimal if `opt.criterion="FLCI"`

`eff.obs` Effective number of observations used by `estimate`

`h` Bandwidth used

`naive` Coverage of CI that ignores bias and uses `qnorm(1-alpha/2)` as critical value

`call` The matched call

`fs` Not relevant for inference at a point

Note

`subset` is evaluated in the same way as variables in `formula`, that is first in `data` and then in the environment of `formula`.

References

Armstrong, Timothy B., and Michal Kolesár. 2020. "Simple and Honest Confidence Intervals in Nonparametric Regression." Quantitative Economics 11 (1): 1–39.

Examples

```
# Lee dataset
LPPHonest(voteshare ~ margin, data = lee08, subset = margin>0,
          kern = "uniform", M = 0.1, h = 10, sclass = "T")
```

---

plot_RDscatter — *Scatterplot of binned raw observations*

Description

Scatterplot of raw observations in which each point corresponds to an binned average.

Usage

```
plot_RDscatter(
  formula,
  data,
  subset,
  cutoff = 0,
  avg = 10,
  xlab = NULL,
  ylab = NULL,
  vert = TRUE,
  propdotsize = FALSE
)
```

Arguments

`formula`	object of class `"formula"` (or one that can be coerced to that class) of the form `y ~ x`
`data`	optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the outcome and running variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which the function is called.
`subset`	optional vector specifying a subset of observations to be used
`cutoff`	specifies the RD cutoff in the running variable.
`avg`	Number of observations to average over. If set to `Inf`, then take averages for each possible value of the running variable (convenient when the running variable is discrete).
`xlab, ylab`	x- and y-axis labels

vert
: Draw a vertical line at cutoff?

propdotsize
: If TRUE, then size of points is proportional to number of observations that the point averages over (useful when avg=Inf). Otherwise the size of points is constant.

Value

A "ggplot" object.

Examples

```
plot_RDscatter(I(log(earnings))~yearat14, data=cghs, cutoff=1947,
               avg=Inf, propdotsize=TRUE)
```

---

rcp — *Battistin, Brugiavini, Rettore, and Weber (2009) retirement consumption puzzle dataset*

Description

Battistin, Brugiavini, Rettore, and Weber (2009) retirement consumption puzzle dataset

Usage

```
rcp
```

Format

A data frame with 30,006 rows and 6 variables:

survey_year Survey year

elig_year Years to/from eligibility (males)

retired Retirement status (males)

food Total household food expenditure

c Total household consumption

cn Total household expenditure on non-durable goods

Source

American Economic Review data archive, doi:10.1257/aer.99.5.2209

References

Erich Battistin, Agar Brugiavini, Enrico Rettore, and Guglielmo Weber. The retirement consumption puzzle: Evidence from a regression discontinuity approach. American Economic Review, 99(5):2209–2226, 2009. doi:10.1257/aer.99.5.2209

RDHonest *Honest inference in RD*

Description

Calculate estimators and bias-aware one- and two-sided CIs for the sharp RD parameter.

Usage

```
RDHonest(
  formula,
  data,
  subset,
  weights,
  cutoff = 0,
  M,
  kern = "triangular",
  na.action,
  opt.criterion = "MSE",
  h,
  se.method = "nn",
  alpha = 0.05,
  beta = 0.8,
  J = 3,
  sclass = "H",
  order = 1,
  se.initial = "EHW"
)
```

Arguments

`formula`	object of class `"formula"` (or one that can be coerced to that class) of the form `outcome ~ running_variable`
`data`	optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the outcome and running variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which the function is called.
`subset`	optional vector specifying a subset of observations to be used in the fitting process.
`weights`	Optional vector of weights to weight the observations (useful for aggregated data). Disregarded if optimal kernel is used.
`cutoff`	specifies the RD cutoff in the running variable.
`M`	Bound on second derivative of the conditional mean function.
`kern`	specifies kernel function used in the local regression. It can either be a string equal to `"triangular"` ($k(u) = (1-\|u\|)_+$), `"epanechnikov"` ($k(u) = (3/4)(1-u^2)_+$), or `"uniform"` ($k(u) = (\|u\| < 1)/2$), or else a kernel function.

`na.action` function which indicates what should happen when the data contain `NA`s. The default is set by the `na.action` setting of `options` (usually `na.omit`).

`opt.criterion` Optimality criterion that bandwidth is designed to optimize. The options are:

- `"MSE"` Finite-sample maximum MSE
- `"FLCI"` Length of (fixed-length) two-sided confidence intervals.
- `"OCI"` Given quantile of excess length of one-sided confidence intervals

The methods use conditional variance given by `sigma2`, if supplied. Otherwise, for the purpose of estimating the optimal bandwidth, conditional variance is estimated using the method specified by `se.initial`.

`h` bandwidth, a scalar parameter. If not supplied, optimal bandwidth is computed according to criterion given by `opt.criterion`.

`se.method` Vector with methods for estimating standard error of estimate. If `NULL`, standard errors are not computed. The elements of the vector can consist of the following methods:

- **"nn"** Nearest neighbor method
- **"EHW"** Eicker-Huber-White, with residuals from local regression (local polynomial estimators only).
- **"demeaned"** Like EHW, but instead of using the regression residuals, estimate σ_i^2 by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope). Local polynomial estimators only.
- **"supplied.var"** Use conditional variance supplied by `sigma2` or `d` instead of computing residuals

`alpha` determines confidence level, `1-alpha` for constructing/optimizing confidence intervals.

`beta` Determines quantile of excess length to optimize, if bandwidth optimizes given quantile of excess length of one-sided confidence intervals; otherwise ignored.

`J` Number of nearest neighbors, if "nn" is specified in `se.method`.

`sclass` Smoothness class, either `"T"` for Taylor or `"H"` for Hölder class.

`order` Order of local regression 1 for linear, 2 for quadratic.

`se.initial` Method for estimating initial variance for computing optimal bandwidth. Except for `"nn"`, all methods assume homoskedasticity on either side of cutoff (for RD), or for all data (for inference at a point).

- **"EHW"** Based on residuals from a local linear regression using a triangular kernel, and a bandwidth given by a rule-of-thumb bandwidth suggested by Fan and Gijbels (1996) (for inference at a point), or Imbens and Kalyanaraman (2012, IK) bandwidth (for fuzzy and sharp RD). For fuzzy RD, the IK bandwidth is based on the reduced-form regression.
- **"demeaned"** Like EHW, but instead of using the regression residuals, estimate σ_i^2 by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope).
- **"Silverman"** Use residuals from local constant regression with uniform kernel and bandwidth selected using Silverman's rule of thumb, as in Equation (14) in Imbens and Kalyanaraman (2012)

"SilvermanNN" Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals.

"nn" Use nearest neighbor estimates, without assuming homoskedasticity

Details

The bandwidth is calculated to be optimal for a given performance criterion, as specified by `opt.criterion`. Alternatively, for local polynomial estimators, the bandwidth can be specified by `h`. If `kern="optimal"`, calculate optimal estimators under second-order Taylor smoothness class.

Value

Returns an object of class `"NPRResults"`. The function `print` can be used to obtain and print a summary of the results. An object of class `"NPRResults"` is a list containing the following components

`estimate` Point estimate. This estimate is MSE-optimal if `opt.criterion="MSE"`

`lff` Least favorable function, only relevant for optimal estimator under Taylor class.

`maxbias` Maximum bias of `estimate`

`sd` Standard deviation of estimate

`lower`, `upper` Lower (upper) end-point of a one-sided CI based on `estimate`. This CI is optimal if `opt.criterion=="OCI"`

`hl` Half-length of a two-sided CI based on `estimate`, so that the CI is given by `c(estimate-hl, estimate+hl)`. The CI is optimal if `opt.criterion="FLCI"`

`eff.obs` Effective number of observations used by `estimate`

`h` Bandwidth used

`naive` Coverage of CI that ignores bias and uses `qnorm(1-alpha/2)` as critical value

`call` the matched call

`fs` Not relevant for sharp RD

Note

`subset` is evaluated in the same way as variables in `formula`, that is first in `data` and then in the environment of `formula`.

References

Armstrong, Timothy B., and Michal Kolesár. 2018. "Optimal Inference in a Class of Regression Models." Econometrica 86 (2): 655–83.

Armstrong, Timothy B., and Michal Kolesár. 2020. "Simple and Honest Confidence Intervals in Nonparametric Regression." Quantitative Economics 11 (1): 1–39.

Imbens, Guido, and Kalyanaraman, Karthik, "Optimal bandwidth choice for the regression discontinuity estimator." The Review of Economic Studies 79 (3): 933-959.

Kolesár, Michal, and Christoph Rothe. 2018. "Inference in Regression Discontinuity Designs with a Discrete Running Variable." American Economic Review 108 (8): 2277–2304.

Examples

```
# Lee dataset
RDHonest(voteshare ~ margin, data = lee08, kern = "uniform", M = 0.1, h = 10)
```

RDHonestBME — *CIs in sharp RD with discrete regressors under bounded misspecification error class*

Description

Computes honest CIs for local linear regression with uniform kernel under the bounded misspecification error class of functions, as considered in Kolesár and Rothe (2018)

Usage

```
RDHonestBME(
  formula,
  data,
  subset,
  weights,
  cutoff = 0,
  na.action,
  h = Inf,
  alpha = 0.05,
  order = 0,
  regformula
)
```

Arguments

formula	object of class "formula" (or one that can be coerced to that class) of the form outcome ~ running_variable
data	optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the outcome and running variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which the function is called.
subset	optional vector specifying a subset of observations to be used in the fitting process.
weights	Optional vector of weights to weight the observations (useful for aggregated data). Disregarded if optimal kernel is used.
cutoff	specifies the RD cutoff in the running variable.
na.action	function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options (usually na.omit).
h	bandwidth, a scalar parameter. If not supplied, optimal bandwidth is computed according to criterion given by opt.criterion.

`alpha`	determines confidence level, $1 - \alpha$
`order`	Order of local regression `1` for linear, `2` for quadratic.
`regformula`	Explicitly specify regression formula to use instead of running a local linear regression, with `y` and `x` denoting the outcome and the running variable, and cutoff is normalized to `0`. Local linear regression (`order = 1`) is equivalent to `regformula = "y~x*I(x>0)"`. Inference is done on the `order+2`th element of the design matrix

Details

The parameter `weights` is ignored, it is only included to keep a unified interface with `RDHonest`.

Note

`subset` is evaluated in the same way as variables in `formula`, that is first in `data` and then in the environment of `formula`.

References

Kolesár, Michal, and Christoph Rothe. 2018. "Inference in Regression Discontinuity Designs with a Discrete Running Variable." American Economic Review 108 (8): 2277–2304.

Examples

```
RDHonestBME(log(cghs$earnings)~yearat14, data=cghs, h=3,
            order=1, cutoff=1947)
## Equivalent to
RDHonestBME(log(cghs$earnings)~yearat14, data=cghs, h=3,
            cutoff=1947, order=1, regformula="y~x*I(x>=0)")
```

`RDSmoothnessBound` *Lower bound on smoothness constant M in RD designs*

Description

Estimate a lower bound on smoothness constant M and provide a lower confidence interval.

Usage

```
RDSmoothnessBound(
  d,
  s,
  separate = TRUE,
  multiple = TRUE,
  alpha = 0.05,
  sclass = "T"
)
```

Arguments

d	object of class "RDData"
s	Number of support points that curvature estimates should average over
separate	If TRUE, report estimates separately for data above and below cutoff. If FALSE, report pooled estimates
multiple	If TRUE, use multiple curvature estimates. If FALSE, use a single estimate using only observations closest to the cutoff.
alpha	determines confidence level 1-alpha.
sclass	Smoothness class, either "T" for Taylor or "H" for Hölder class.

Value

Returns a list with the following elements

mu+,mu- Lower bound of CI for observations above and below cutoff

Z+,Z- Point estimate used for lower bound

sd+,sd- Standard deviations of point estimates

References

Armstrong, Timothy B., and Michal Kolesár. 2018. "Optimal Inference in a Class of Regression Models." Econometrica 86 (2): 655–83.

Kolesár, Michal, and Christoph Rothe. 2018. "Inference in Regression Discontinuity Designs with a Discrete Running Variable." American Economic Review 108 (8): 2277–2304.

RDTEfficiencyBound *Finite-sample efficiency bounds for minimax CIs*

Description

Compute efficiency of minimax one-sided CIs at constant functions, or efficiency of two-sided fixed-length CIs at constant functions under second-order Taylor smoothness class.

Usage

```
RDTEfficiencyBound(
  d,
  M,
  opt.criterion = "FLCI",
  alpha = 0.05,
  beta = 0.5,
  se.initial = "EHW"
)
```

Arguments

- `d` — object of class `"RDData"`
- `M` — Bound on second derivative of the conditional mean function.
- `opt.criterion` — `"FLCI"` for computing efficiency of two-sided CIs, and `"OCI"` for minimax one-sided CIs.
- `alpha` — determines confidence level, `1-alpha` for constructing/optimizing confidence intervals.
- `beta` — Determines quantile of excess length to optimize, if bandwidth optimizes given quantile of excess length of one-sided confidence intervals; otherwise ignored.
- `se.initial` — Method for estimating initial variance for computing optimal bandwidth. Except for `"nn"`, all methods assume homoskedasticity on either side of cutoff (for RD), or for all data (for inference at a point).

  "EHW" Based on residuals from a local linear regression using a triangular kernel, and a bandwidth given by a rule-of-thumb bandwidth suggested by Fan and Gijbels (1996) (for inference at a point), or Imbens and Kalyanaraman (2012, IK) bandwidth (for fuzzy and sharp RD). For fuzzy RD, the IK bandwidth is based on the reduced-form regression.

  "demeaned" Like EHW, but instead of using the regression residuals, estimate σ_i^2 by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope).

  "Silverman" Use residuals from local constant regression with uniform kernel and bandwidth selected using Silverman's rule of thumb, as in Equation (14) in Imbens and Kalyanaraman (2012)

  "SilvermanNN" Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals.

  "nn" Use nearest neighbor estimates, without assuming homoskedasticity

References

Armstrong, Timothy B., and Michal Kolesár. 2018. "Optimal Inference in a Class of Regression Models." Econometrica 86 (2): 655–83.

`rebp` *Austrian unemployment duration data from Lalive (2008)*

Description

Subset of Lalive data for individuals in the regions affected by the REBP program

Usage

```
rebp
```

Format

A data frame with 29,371 rows and 4 variables:

age Age in years, at monthly accuracy

period Indicator for whether REBP is in place

female Indicator for female

duration unemployment duration in weeks

Source

Rafael Lalive's website, https://sites.google.com/site/rafaellalive/

References

Rafael Lalive. How do extended benefits affect unemployment duration? A regression discontinuity approach. Journal of Econometrics, 142(2):785–806, February 2008. doi:10.1016/j.jeconom.2007.05.013

Index