

# Package ‘RDHonest’

July 12, 2019

**Title** Honest inference in regression discontinuity designs

**Version** 0.3.2

**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

**RoxygenNote** 6.1.1

**URL** <https://github.com/kolesarm/RDHonest>

**VignetteBuilder** knitr

**Language** en-US

**BugReports** <https://github.com/kolesarm/RDHonest/issues>

## R topics documented:

cghs	2
CVb	3
EqKern	4
FRDData	4
FRDHonest	6
FRDOptBW	9
headst	11

IKBW.fit . . . . .	12
kernC . . . . .	13
KernMoment . . . . .	14
lee08 . . . . .	14
LPPData . . . . .	15
LPPHonest . . . . .	16
LPPOptBW . . . . .	19
NPR_MROT.fit . . . . .	21
NPRHonest.fit . . . . .	21
NPROptBW.fit . . . . .	23
NPRPrelimVar.fit . . . . .	25
NPRreg.fit . . . . .	26
plot_RDscatter . . . . .	27
rcp . . . . .	28
RDDData . . . . .	29
RDHonest . . . . .	30
RDHonestBME . . . . .	33
RDOptBW . . . . .	34
RDSmoothnessBound . . . . .	37
RDTEfficiencyBound . . . . .	38
RDOpt.fit . . . . .	39
rebp . . . . .	40
ROTBW.fit . . . . .	41
<b>Index</b>	<b>42</b>

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, <https://doi.org/10.1257/000282806776157641>

## References

Oreopoulos, P. (2006): "Estimating Average and Local Average Treatment Effects When Compulsory Education Schooling Laws Really Matter", *American Economic Review*, 96 (1), 152-175

---

 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  $X \pm cv_{1-\alpha}(B)$  will have 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, $1 - \alpha$ . Vector of values between 0 and 1.

## Value

Data frame with the following columns:

**bias** Value of bias as specified by B

**alpha** Value of  $\alpha$  as specified by alpha

**cv** Critical value

**TeXDescription** LaTeX-friendly description of the current row

## Examples

```
## 90% critical value:
CVb(B = 1, alpha = 0.1)
## Returns data frame with 4 rows
CVb(B = c(0, 0.5, 1), alpha = c(0.05, 0.1))
```

---

EqKern	<i>Equivalent kernel for local linear regression.</i>
--------	---

---

### 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)
```

---

FRDDData	<i>Class Constructor for "FRDDData"</i>
----------	---

---

### Description

Convert data to a standardized format for use with low-level functions. If the cutoff for treatment is non-zero, shift the running variable so that the cutoff is at zero.

### Usage

```
FRDDData(d, cutoff)
```

**Arguments**

- d** list with first element corresponding to the outcome vector, second element to the treatment vector, third element to running variable vector, optionally an element called "sigma2" that is a matrix with four columns corresponding to the [1, 1], [1, 2], [2, 1], and [2, 2] elements of the conditional variance matrix of the outcome and the treatment (or an estimate of the conditional variance matrix), and optionally a column called "weights" if observations are aggregated by cell.
- cutoff** specifies the cutoff for the running variable

**Value**

An object of class "FRDData", which is a list containing the following components:

**Ym** Matrix of outcomes and treatments for observations below cutoff

**Yp** Matrix of outcomes and treatments for observations above cutoff

**Xm** Running variable for observations below cutoff

**Xp** Running variable for observations above cutoff

**wm** weights for observations below cutoff

**wp** weights for observations above cutoff

**sigma2m** Matrix of conditional covariances for the outcome and the treatment for observations below cutoff

**sigma2p** Matrix of conditional covariances for the outcome and the treatment for observations above cutoff

**orig.cutoff** Original cutoff

**var.names** Names of the outcome, the treatment, and the running variable in supplied data frame

**See Also**

[RDDData](#) for sharp RD, and [LPPData](#) for inference at a point

**Examples**

```
## Transform retirement data
d <- FRDData(rcp[, c(6, 3, 2)], cutoff=0)
## Outcome in logs
d <- FRDData(cbind(logcn=log(rcp[, 6]), rcp[, c(3, 2)]), cutoff=0)
```

**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, bw.equal = TRUE, 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 <code>outcome ~ running_variable</code>
data	optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the outcome and running variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , 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 NAs. The default is set by the <code>na.action</code> setting of options (usually <code>na.omit</code> ).
opt.criterion	<p>Optimality criterion that bandwidth is designed to optimize. The options are:</p> <p>"MSE" Finite-sample maximum MSE</p> <p>"FLCI" Length of (fixed-length) two-sided confidence intervals.</p> <p>"OCI" Given quantile of excess length of one-sided confidence intervals</p> <p>The methods use conditional variance given by <code>sigma2</code>, if supplied. Otherwise, for the purpose of estimating the optimal bandwidth, conditional variance is estimated using the method specified by <code>se.initial</code>.</p>
bw.equal	logical specifying whether bandwidths on either side of cutoff should be constrained to equal to each other.

h	bandwidth, a scalar parameter. For fuzzy or sharp RD, it can be a named vector of length two with names "p" and "m", in which case the bandwidth $h["m"]$ is used for observations below the cutoff, and the bandwidth $h["p"]$ is used for observations above the cutoff. If not supplied, optimal bandwidth is computed according to criterion given by <code>opt.criterion</code> .
se.method	<p>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:</p> <p><b>"nn"</b> Nearest neighbor method</p> <p><b>"EHW"</b> Eicker-Huber-White, with residuals from local regression (local polynomial estimators only).</p> <p><b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate <math>\sigma_i^2</math> by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope). Local polynomial estimators only.</p> <p><b>"plugin"</b> Plug-in estimate based on asymptotic variance. Local polynomial estimators in sharp RD only.</p> <p><b>"supplied.var"</b> Use conditional variance supplied by <code>sigma2</code> or <code>d</code> instead of computing residuals</p>
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 <code>se.method</code> .
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	<p>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).</p> <p><b>"EHW"</b> Based on residuals from a local linear regression using a triangular kernel, and and bandwidth given by a rule-of-thumb bandwidth (for inference at a point, see <a href="#">ROTBW.fit</a>), or IK bandwidth (for fuzzy and sharp RD, see <a href="#">IKBW.fit</a>). For fuzzy RD, the IK bandwidth is based on the reduced-form regression.</p> <p><b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate <math>\sigma_i^2</math> by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope).</p> <p><b>"Silverman"</b> 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)</p> <p><b>"SilvermanNN"</b> Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals.</p> <p><b>"nn"</b> Use nearest neighbor estimates, without assuming homoskedasticity</p>
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`. It is calculated using the function [FRDOptBW](#). 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"`

`h1` Half-length of a two-sided CI based on estimate, so that the CI is given by `c(estimate-h1, estimate+h1)`. The CI is optimal if `opt.criterion="FLCI"`

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

`hp, hm` Bandwidths used above and below the cutoff

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

## See Also

[FRDOptBW](#)

## Examples

```
FRDHonest(cn~retired | elig_year, data=rcp, cutoff=0, M=c(1, 0.1),
          kern="triangular", opt.criterion="MSE", T0=0)
```



**Description**

Estimate bandwidth for sharp RD based on local polynomial regression that optimizes either maximum mean squared error, or length or quantiles of excess length of a honest CI under second order Hölder or Taylor class.

**Usage**

```
FRDOptBW(formula, data, subset, weights, cutoff = 0, M,
  kern = "triangular", na.action, opt.criterion, bw.equal = TRUE,
  alpha = 0.05, beta = 0.8, 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 NAs. The default is set by the na.action setting of options (usually na.omit).
opt.criterion	<p>Optimality criterion that bandwidth is designed to optimize. The options are:</p> <p>"MSE" Finite-sample maximum MSE</p> <p>"FLCI" Length of (fixed-length) two-sided confidence intervals.</p> <p>"OCI" Given quantile of excess length of one-sided confidence intervals</p> <p>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.</p>

<code>bw.equal</code>	logical specifying whether bandwidths on either side of cutoff should be constrained to equal to each other.
<code>alpha</code>	determines confidence level, $1-\alpha$ for constructing/optimizing confidence intervals.
<code>beta</code>	Determines quantile of excess length to optimize, if bandwidth optimizes given quantile of excess length of one-sided confidence intervals; otherwise ignored.
<code>sclass</code>	Smoothness class, either "T" for Taylor or "H" for Hölder class.
<code>order</code>	Order of local regression 1 for linear, 2 for quadratic.
<code>se.initial</code>	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). <b>"EHW"</b> Based on residuals from a local linear regression using a triangular kernel, and bandwidth given by a rule-of-thumb bandwidth (for inference at a point, see <a href="#">ROTBW.fit</a> ), or IK bandwidth (for fuzzy and sharp RD, see <a href="#">IKBW.fit</a> ). For fuzzy RD, the IK bandwidth is based on the reduced-form regression. <b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate $\sigma_i^2$ by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope). <b>"Silverman"</b> 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) <b>"SilvermanNN"</b> Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals. <b>"nn"</b> Use nearest neighbor estimates, without assuming homoskedasticity
<code>T0</code>	Initial estimate of the treatment effect for calculating the optimal bandwidth. Only relevant for Fuzzy RD.

### Value

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

`hp` bandwidth for observations above cutoff  
`hm` bandwidth for observations below cutoff, equal to `hp` unless `bw.equal==FALSE`  
`sigma2m, sigma2p` estimate of conditional variance just above and just below cutoff,  $\sigma_+^2(0)$  and  $\sigma_-^2(0)$   
`f0` estimate of density of running variable at cutoff, if bandwidth computed using asymptotic method  
`call` the matched call  
`na.action` (where relevant) information on handling of missing data.

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

Armstrong, Timothy B., and Michal Kolesár. 2019. "Simple and Honest Confidence Intervals in Nonparametric Regression", *arXiv*: 1606.01200.

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.

## See Also

[RDHonest](#)

## Examples

```
FRDOptBW(cn~retired | elig_year, data=rcp, cutoff=0, M=c(1, 0.1),
          kern="triangular", opt.criterion="FLCI")
```

---

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 website, <http://faculty.econ.ucdavis.edu/faculty/dlmiller/statafiles/>

### References

Ludwig, J., and D. L. Miller (2007): "Does Head Start improve children's life chances? Evidence from a regression discontinuity design," *Quarterly Journal of Economics*, 122 (1), 159-208.

---

IKBW.fit

*Imbens and Kalyanaraman bandwidth*

---

### Description

Calculate bandwidth for sharp RD based on local linear regression using method by Imbens and Kalyanaraman (2012)

### Usage

```
IKBW.fit(d, kern = "triangular", order = 1, verbose = FALSE)
```

### Arguments

d	object of class "RDDData"
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.
order	Order of local regression 1 for linear, 2 for quadratic.
verbose	Print details of calculation?

### Value

Imbens and Kalyanaraman bandwidth

### References

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

### Examples

```
## Reproduce bandwidth from Section 6.2 in Imbens and Kalyanaraman (2012)
IKBW.fit(RDDData(lee08, cutoff=0))
```

---

kernC	<i>Constants for common kernels.</i>
-------	--------------------------------------

---

## 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	<i>Moments of a kernel.</i>
------------	-----------------------------

---

**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  $\int_X |u^j k(u)|$ , 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	<i>Lee (2008) US House elections dataset</i>
-------	--

---

**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**

Lee, D. S. (2008): "Randomized experiments from non-random selection in U.S. House elections," *Journal of Econometrics*, 142 (2), 675-697.

LPPData

*Class Constructor for "LPPData"***Description**

Convert data to standardized format for use with low-level functions. If the point of interest  $x_0$  is non-zero, shift the independent variable so that it is at zero.

**Usage**

```
LPPData(d, point)
```

**Arguments**

**d** a data frame or a list with first column corresponding to the outcome variable, second column corresponding to the independent variable, optionally a column called "sigma2" that corresponds to the conditional variance of the outcome (or an estimate of the conditional variance), and optionally a column called "weights" if observations are aggregated by cell.

**point** specifies the point  $x_0$  at which to calculate the conditional mean

**Value**

An object of class "LPPData", which is a list containing the following components:

**Y** Outcome vector

**X** Independent variable

**w** Weights

**sigma2** Conditional variance of the outcome

**orig.point** Original value of  $x_0$

**var.names** Names of outcome and independent variable in supplied data frame

**See Also**

[FRDData](#) for fuzzy RD, and [RDData](#) for sharp RD

## Examples

```
## Transform Lee data
d1 <- LPPData(lee08[lee08$margin>=0, ], point=0)
d2 <- LPPData(lee08, point=50)
```

---

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 <code>outcome ~ independent_variable</code>
data	optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the outcome and independent variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , 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 NAs. The default is set by the <code>na.action</code> setting of options (usually <code>na.omit</code> ).
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 <code>sigma2</code> , if supplied. Otherwise, for the purpose of estimating the optimal bandwidth, conditional variance is estimated using the method specified by <code>se.initial</code> .
<code>h</code>	bandwidth, a scalar parameter. For fuzzy or sharp RD, it can be a named vector of length two with names "p" and "m", in which case the bandwidth <code>h["m"]</code> is used for observations below the cutoff, and the bandwidth <code>h["p"]</code> is used for observations above the cutoff. If not supplied, optimal bandwidth is computed according to criterion given by <code>opt.criterion</code> .
<code>se.method</code>	<p>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:</p> <p><b>"nn"</b> Nearest neighbor method</p> <p><b>"EHW"</b> Eicker-Huber-White, with residuals from local regression (local polynomial estimators only).</p> <p><b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate <math>\sigma_i^2</math> by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope). Local polynomial estimators only.</p> <p><b>"plugin"</b> Plug-in estimate based on asymptotic variance. Local polynomial estimators in sharp RD only.</p> <p><b>"supplied.var"</b> Use conditional variance supplied by <code>sigma2</code> or <code>d</code> instead of computing residuals</p>
<code>alpha</code>	determines confidence level, 1-alpha for constructing/optimizing confidence intervals.
<code>beta</code>	Determines quantile of excess length to optimize, if bandwidth optimizes given quantile of excess length of one-sided confidence intervals; otherwise ignored.
<code>J</code>	Number of nearest neighbors, if "nn" is specified in <code>se.method</code> .
<code>sclass</code>	Smoothness class, either "T" for Taylor or "H" for Hölder class.
<code>order</code>	Order of local regression 1 for linear, 2 for quadratic.
<code>se.initial</code>	<p>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).</p> <p><b>"EHW"</b> Based on residuals from a local linear regression using a triangular kernel, and and bandwidth given by a rule-of-thumb bandwidth (for inference at a point, see <a href="#">ROTBW.fit</a>), or IK bandwidth (for fuzzy and sharp RD, see <a href="#">IKBW.fit</a>). For fuzzy RD, the IK bandwidth is based on the reduced-form regression.</p> <p><b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate <math>\sigma_i^2</math> by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope).</p> <p><b>"Silverman"</b> 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)</p> <p><b>"SilvermanNN"</b> Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals.</p> <p><b>"nn"</b> Use nearest neighbor estimates, without assuming homoskedasticity</p>

## Details

The bandwidth is calculated to be optimal for a given performance criterion, as specified by `opt.criterion`. It is calculated using the function [LPPOptBW](#). 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 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

`hp, hm` Bandwidth used, both numbers are equal (i.e. `hp=hm`)

`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. 2019. "Simple and Honest Confidence Intervals in Nonparametric Regression", *arXiv: 1606.01200*.

## See Also

[LPPOptBW](#)

## Examples

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

LPPOptBW

*Optimal Bandwidth Selection for inference at a point***Description**

Estimate bandwidth based on local polynomial regression that optimizes either maximum mean squared error, or length or quantiles of excess length of a honest CI under second order Hölder or Taylor class.

**Usage**

```
LPPOptBW(formula, data, subset, weights, point = 0, M,
  kern = "triangular", na.action, opt.criterion, alpha = 0.05,
  beta = 0.8, 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 <code>outcome ~ independent_variable</code>
data	optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the outcome and independent variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , 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 NAs. The default is set by the <code>na.action</code> setting of options (usually <code>na.omit</code> ).
opt.criterion	<p>Optimality criterion that bandwidth is designed to optimize. The options are:</p> <p>"MSE" Finite-sample maximum MSE</p> <p>"FLCI" Length of (fixed-length) two-sided confidence intervals.</p> <p>"OCI" Given quantile of excess length of one-sided confidence intervals</p> <p>The methods use conditional variance given by <code>sigma2</code>, if supplied. Otherwise, for the purpose of estimating the optimal bandwidth, conditional variance is estimated using the method specified by <code>se.initial</code>.</p>
alpha	determines confidence level, $1 - \alpha$ for constructing/optimizing confidence intervals.

<code>beta</code>	Determines quantile of excess length to optimize, if bandwidth optimizes given quantile of excess length of one-sided confidence intervals; otherwise ignored.
<code>sclass</code>	Smoothness class, either "T" for Taylor or "H" for Hölder class.
<code>order</code>	Order of local regression 1 for linear, 2 for quadratic.
<code>se.initial</code>	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).  <b>"EHW"</b> Based on residuals from a local linear regression using a triangular kernel, and and bandwidth given by a rule-of-thumb bandwidth (for inference at a point, see <a href="#">ROTBW.fit</a> ), or IK bandwidth (for fuzzy and sharp RD, see <a href="#">IKBW.fit</a> ). For fuzzy RD, the IK bandwidth is based on the reduced-form regression. <b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate $\sigma_i^2$ by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope). <b>"Silverman"</b> 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) <b>"SilvermanNN"</b> Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals. <b>"nn"</b> Use nearest neighbor estimates, without assuming homoskedasticity

### Value

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

`h` Bandwidth  
`sigma2` estimate of conditional variance at a point  
`call` the matched call  
`na.action` (where relevant) information on handling of missing data.

### 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. 2019. "Simple and Honest Confidence Intervals in Nonparametric Regression", *arXiv: 1606.01200*.

### See Also

[LPPHonest](#)

**Examples**

```
# Lee dataset
LPP0ptBW(voteshare ~ margin, data = lee08, subset=margin>0,
         kern = "uniform", M = 0.1, opt.criterion = "MSE", sclass = "H")
```

NPR\_MROT.fit

*Rule of thumb for choosing M***Description**

Use global quartic regression to estimate a bound on the second derivative for inference under second order Hölder class. For RD, use a separate regression on either side of the cutoff

**Usage**

```
NPR_MROT.fit(d)
```

**Arguments**

`d` object of class "RDDData", "FRDDData", or "LPPData".

**Examples**

```
NPR_MROT.fit(RDDData(lee08, cutoff=0))
NPR_MROT.fit(LPPData(lee08[lee08$margin>0, ], point=0))
d <- FRDDData(cbind(logcn=log(rcp[, 6 ]), rcp[, c(3, 2)]), cutoff=0)
NPR_MROT.fit(d)
```

NPRHonest.fit

*Honest inference in nonparametric regression***Description**

Basic computing engine called by [RDHonest](#), [LPPHonest](#), and [FRDHonest](#) to compute honest confidence intervals for local polynomial estimators.

**Usage**

```
NPRHonest.fit(d, M, kern = "triangular", h, opt.criterion,
             bw.equal = TRUE, alpha = 0.05, beta = 0.8, se.method = "nn",
             J = 3, sclass = "H", order = 1, se.initial = "EHW", T0 = 0,
             T0bias = FALSE)
```

**Arguments**

<code>d</code>	object of class "RDDData", "FRDDData", or "LPPData"
<code>M</code>	Bound on second derivative of the conditional mean function.
<code>kern</code>	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.
<code>h</code>	bandwidth, a scalar parameter. For fuzzy or sharp RD, it can be a named vector of length two with names "p" and "m", in which case the bandwidth $h["m"]$ is used for observations below the cutoff, and the bandwidth $h["p"]$ is used for observations above the cutoff. If not supplied, optimal bandwidth is computed according to criterion given by <code>opt.criterion</code> .
<code>opt.criterion</code>	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 <code>sigma2</code> , if supplied. Otherwise, for the purpose of estimating the optimal bandwidth, conditional variance is estimated using the method specified by <code>se.initial</code> .
<code>bw.equal</code>	logical specifying whether bandwidths on either side of cutoff should be constrained to equal to each other.
<code>alpha</code>	determines confidence level, $1 - \alpha$ for constructing/optimizing confidence intervals.
<code>beta</code>	Determines quantile of excess length to optimize, if bandwidth optimizes given quantile of excess length of one-sided confidence intervals; otherwise ignored.
<code>se.method</code>	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 $\sigma_i^2$ by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope). Local polynomial estimators only. "plugin" Plug-in estimate based on asymptotic variance. Local polynomial estimators in sharp RD only. "supplied.var" Use conditional variance supplied by <code>sigma2</code> or <code>d</code> instead of computing residuals
<code>J</code>	Number of nearest neighbors, if "nn" is specified in <code>se.method</code> .
<code>sclass</code>	Smoothness class, either "T" for Taylor or "H" for Hölder class.
<code>order</code>	Order of local regression 1 for linear, 2 for quadratic.
<code>se.initial</code>	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).

	<p><b>"EHW"</b> Based on residuals from a local linear regression using a triangular kernel, and bandwidth given by a rule-of-thumb bandwidth (for inference at a point, see <a href="#">ROTBW.fit</a>), or IK bandwidth (for fuzzy and sharp RD, see <a href="#">IKBW.fit</a>). For fuzzy RD, the IK bandwidth is based on the reduced-form regression.</p> <p><b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate <math>\sigma_i^2</math> by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope).</p> <p><b>"Silverman"</b> 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)</p> <p><b>"SilvermanNN"</b> Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals.</p> <p><b>"nn"</b> Use nearest neighbor estimates, without assuming homoskedasticity</p>
T0	Initial estimate of the treatment effect for calculating the optimal bandwidth. Only relevant for Fuzzy RD.
T0bias	When evaluating the maximum bias of the estimate, use the estimate itself (if T0bias==FALSE), or use the preliminary estimate T0 (if T0bias==TRUE). Only relevant for Fuzzy RD.

### Value

Returns an object of class "NPRResults", see descriptions in [RDHonest](#), [LPPHonest](#), and [FRDHonest](#).

### Examples

```
d <- RDData(lee08, cutoff=0)
NPRHonest.fit(d, opt.criterion="MSE", M=NPR_MROT.fit(d))
```

---

NPROptBW.fit

*Optimal bandwidth selection in nonparametric regression*


---

### Description

Basic computing engine called by [RDOptBW](#), [FRDOptBW](#), and [LPPOptBW](#) to compute the optimal bandwidth

### Usage

```
NPROptBW.fit(d, M, kern = "triangular", opt.criterion, bw.equal = TRUE,
  alpha = 0.05, beta = 0.8, sclass = "H", order = 1,
  se.initial = "EHW", T0 = 0)
```

**Arguments**

<code>d</code>	object of class "RDData", "FRDData", or "LPPData"
<code>M</code>	Bound on second derivative of the conditional mean function.
<code>kern</code>	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.
<code>opt.criterion</code>	<p>Optimality criterion that bandwidth is designed to optimize. The options are:</p> <p>"MSE" Finite-sample maximum MSE</p> <p>"FLCI" Length of (fixed-length) two-sided confidence intervals.</p> <p>"OCI" Given quantile of excess length of one-sided confidence intervals</p> <p>The methods use conditional variance given by <code>sigma2</code>, if supplied. Otherwise, for the purpose of estimating the optimal bandwidth, conditional variance is estimated using the method specified by <code>se.initial</code>.</p>
<code>bw.equal</code>	logical specifying whether bandwidths on either side of cutoff should be constrained to equal to each other.
<code>alpha</code>	determines confidence level, $1 - \alpha$ for constructing/optimizing confidence intervals.
<code>beta</code>	Determines quantile of excess length to optimize, if bandwidth optimizes given quantile of excess length of one-sided confidence intervals; otherwise ignored.
<code>sclass</code>	Smoothness class, either "T" for Taylor or "H" for Hölder class.
<code>order</code>	Order of local regression 1 for linear, 2 for quadratic.
<code>se.initial</code>	<p>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).</p> <p><b>"EHW"</b> Based on residuals from a local linear regression using a triangular kernel, and bandwidth given by a rule-of-thumb bandwidth (for inference at a point, see <a href="#">ROTBW.fit</a>), or IK bandwidth (for fuzzy and sharp RD, see <a href="#">IKBW.fit</a>). For fuzzy RD, the IK bandwidth is based on the reduced-form regression.</p> <p><b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate <math>\sigma_i^2</math> by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope).</p> <p><b>"Silverman"</b> 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)</p> <p><b>"SilvermanNN"</b> Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals.</p> <p><b>"nn"</b> Use nearest neighbor estimates, without assuming homoskedasticity</p>
<code>T0</code>	Initial estimate of the treatment effect for calculating the optimal bandwidth. Only relevant for Fuzzy RD.

**Value**

Returns an object of class "NPRBW", see descriptions in [RDOptBW](#), [FRDOptBW](#), and [LPPOptBW](#).



## References

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

## Examples

```
## Lee data
d <- RDDData(lee08, cutoff=0)
NPROptBW.fit(d, M=0.1, opt.criterion="MSE")[c("hp", "hm")]
```

---

NPRPrelimVar.fit	<i>Compute preliminary estimate of variance</i>
------------------	---

---

## Description

Compute estimate of variance, which can then be used in optimal bandwidth calculations. Except for `se.initial="nn"`, these estimates are unweighted.

## Usage

```
NPRPrelimVar.fit(d, se.initial = "EHW")
```

## Arguments

- |                         |   |
|-------------------------|---|
| <code>d</code>          | object of class "RDDData", "FRDDData", or "LPPData"   |
| <code>se.initial</code> | 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).<br><br><b>"EHW"</b> Based on residuals from a local linear regression using a triangular kernel, and and bandwidth given by a rule-of-thumb bandwidth (for inference at a point, see <a href="#">ROTBW.fit</a> ), or IK bandwidth (for fuzzy and sharp RD, see <a href="#">IKBW.fit</a> ). For fuzzy RD, the IK bandwidth is based on the reduced-form regression.<br><b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate $\sigma_i^2$ by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope).<br><b>"Silverman"</b> 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)<br><b>"SilvermanNN"</b> Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals.<br><b>"nn"</b> Use nearest neighbor estimates, without assuming homoskedasticity |

## Value

object of the same class as `d` containing estimated variances.

NPRreg.fit

*Nonparametric Regression***Description**

Calculate fuzzy or sharp RD estimate, or estimate of a conditional mean at a point (depending on the class of d), and its variance using local polynomial regression of order order.

**Usage**

```
NPRreg.fit(d, h, kern = "triangular", order = 1, se.method = "nn",
  no.warning = FALSE, J = 3)
```

**Arguments**

d	object of class "LPPData", "RDDData", or "FRDDData"
h	bandwidth, a scalar parameter. For fuzzy or sharp RD, it can be a named vector of length two with names "p" and "m", in which case the bandwidth h["m"] is used for observations below the cutoff, and the bandwidth h["p"] is used for observations above the cutoff. If not supplied, optimal bandwidth is computed according to criterion given by opt.criterion.
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.
order	Order of local regression 1 for linear, 2 for quadratic.
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:  <b>"nn"</b> Nearest neighbor method <b>"EHW"</b> Eicker-Huber-White, with residuals from local regression (local polynomial estimators only). <b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate $\sigma_i^2$ by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope). Local polynomial estimators only. <b>"plugin"</b> Plug-in estimate based on asymptotic variance. Local polynomial estimators in sharp RD only. <b>"supplied.var"</b> Use conditional variance supplied by sigma2 or d instead of computing residuals
no.warning	Don't warn about too few observations
J	Number of nearest neighbors, if "nn" is specified in se.method.

**Value**

list with elements:

**estimate** point estimate

**se** Named vector of standard error estimates, as specified by `se.method`.

**w** Implicit weight function used

**sigma2** Estimate of  $\sigma^2(X)$  for values of  $X$  receiving positive kernel weight. By default, estimates are based on squared regression residuals, as used in "EHW". If `se.method="demeaned"` or `se.method="nn"` is specified, estimates are based on that method, with "nn" method used if both are specified.

**eff.obs** Number of effective observations

**Examples**

```
NPRreg.fit(RDData(lee08, cutoff=0), h=5, order=2, se.method=c("nn", "plugin", "EHW"))
NPRreg.fit(LPPData(lee08[lee08$margin>=0, ], point=0), h=5, order=1)
d <- FRDData(cbind(logcn=log(rcp[, 6]), rcp[, c(3, 2)]), cutoff=0)
r <- NPRreg.fit(d, h=10, order=1)
```

---

plot\_RDscatter

*Scatterplot of binned raw observations*

---

**Description**

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

**Usage**

```
plot_RDscatter(d, avg = 10, xlab = NULL, ylab = NULL,
  window = NULL, vert = TRUE, propdotsize = FALSE)
```

**Arguments**

<code>d</code>	Object of class "RDdata"
<code>avg</code>	Number of observations to average over. If set to <code>Inf</code> , then take averages for each possible value of the running variable (convenient when the running variable is discrete).
<code>xlab, ylab</code>	x- and y-axis labels
<code>window</code>	Width of a window around cutoff to which the graph should be restricted. If not specified, full data range will be plotted
<code>vert</code>	Draw a vertical line at cutoff?
<code>propdotsize</code>	If <code>TRUE</code> , then size of points is proportional to number of observations that the point averages over (useful when <code>avg=Inf</code> ). Otherwise the size of points is constant.

## Examples

```
plot_RDscatter(RDData(lee08, cutoff=0), avg=20)
plot_RDscatter(RDData(data.frame(y=log(cghs$earnings), x=cghs$yearat14),
                             cutoff=1947), avg=Inf, proddotsize=TRUE)
```

rcp

*Battistin et al. (2009) retirement consumption puzzle dataset*

## Description

Battistin et al. (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, <https://doi.org/10.1257/aer.99.5.2209>

## References

Battistin, Erich, Agar Brugiavini, Enrico Rettore, and Guglielmo Weber. 2009. "The Retirement Consumption Puzzle: Evidence from a Regression Discontinuity Approach." *American Economic Review* 99 (5): 2209–26.

---

RDData	<i>Class Constructor for "RDData"</i>
--------	---------------------------------------

---

**Description**

Convert data to a standardized format for use with low-level functions. If the cutoff for treatment is non-zero, shift the running variable so that the cutoff is at zero.

**Usage**

```
RDData(d, cutoff)
```

**Arguments**

<b>d</b>	a data frame or a list with first column corresponding to the outcome variable, second column corresponding to the running variable, optionally a column called "sigma2" that corresponds to the conditional variance of the outcome (or an estimate of the conditional variance), and optionally a column called "weights" if observations are aggregated by cell.
<b>cutoff</b>	specifies the cutoff for the running variable

**Value**

An object of class "RDData", which is a list containing the following components:

**Ym** Outcome vector for observations below cutoff  
**Yp** Outcome vector for observations above cutoff  
**Xm** Running variable for observations below cutoff  
**Xp** Running variable for observations above cutoff  
**wm** weights for observations below cutoff  
**wp** weights for observations above cutoff  
**sigma2m** Conditional variance of the outcome for observations below cutoff  
**sigma2p** Conditional variance of the outcome for observations above cutoff  
**orig.cutoff** Original cutoff  
**var.names** Names of the outcome and the running variable in supplied data frame

**See Also**

[FRDData](#) for fuzzy RD, and [LPPData](#) for inference at a point

**Examples**

```
## Transform Lee data
d <- RDData(lee08, cutoff=0)
```

RDHonest

*Honest inference in RD***Description**

Calculate estimators and one- and two-sided CIs based on local polynomial estimator in RD under second-order Taylor or Hölder smoothness class. If kern="optimal", calculate optimal estimators under second-order Taylor smoothness class.

**Usage**

```
RDHonest(formula, data, subset, weights, cutoff = 0, M,
  kern = "triangular", na.action, opt.criterion, bw.equal = TRUE, 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 NAs. The default is set by the na.action setting of options (usually na.omit).
opt.criterion	<p>Optimality criterion that bandwidth is designed to optimize. The options are:</p> <p>"MSE" Finite-sample maximum MSE</p> <p>"FLCI" Length of (fixed-length) two-sided confidence intervals.</p> <p>"OCI" Given quantile of excess length of one-sided confidence intervals</p> <p>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.</p>

<code>bw.equal</code>	logical specifying whether bandwidths on either side of cutoff should be constrained to equal to each other.
<code>h</code>	bandwidth, a scalar parameter. For fuzzy or sharp RD, it can be a named vector of length two with names "p" and "m", in which case the bandwidth $h["m"]$ is used for observations below the cutoff, and the bandwidth $h["p"]$ is used for observations above the cutoff. If not supplied, optimal bandwidth is computed according to criterion given by <code>opt.criterion</code> .
<code>se.method</code>	<p>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:</p> <p><b>"nn"</b> Nearest neighbor method</p> <p><b>"EHW"</b> Eicker-Huber-White, with residuals from local regression (local polynomial estimators only).</p> <p><b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate <math>\sigma_i^2</math> by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope). Local polynomial estimators only.</p> <p><b>"plugin"</b> Plug-in estimate based on asymptotic variance. Local polynomial estimators in sharp RD only.</p> <p><b>"supplied.var"</b> Use conditional variance supplied by <code>sigma2</code> or <code>d</code> instead of computing residuals</p>
<code>alpha</code>	determines confidence level, $1-\alpha$ for constructing/optimizing confidence intervals.
<code>beta</code>	Determines quantile of excess length to optimize, if bandwidth optimizes given quantile of excess length of one-sided confidence intervals; otherwise ignored.
<code>J</code>	Number of nearest neighbors, if "nn" is specified in <code>se.method</code> .
<code>sclass</code>	Smoothness class, either "T" for Taylor or "H" for Hölder class.
<code>order</code>	Order of local regression 1 for linear, 2 for quadratic.
<code>se.initial</code>	<p>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).</p> <p><b>"EHW"</b> Based on residuals from a local linear regression using a triangular kernel, and bandwidth given by a rule-of-thumb bandwidth (for inference at a point, see <a href="#">ROTBW.fit</a>), or IK bandwidth (for fuzzy and sharp RD, see <a href="#">IKBW.fit</a>). For fuzzy RD, the IK bandwidth is based on the reduced-form regression.</p> <p><b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate <math>\sigma_i^2</math> by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope).</p> <p><b>"Silverman"</b> 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)</p> <p><b>"SilvermanNN"</b> Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals.</p> <p><b>"nn"</b> Use nearest neighbor estimates, without assuming homoskedasticity</p>

## Details

The bandwidth is calculated to be optimal for a given performance criterion, as specified by `opt.criterion`. For local polynomial estimators, this optimal bandwidth is calculated using the function [RDOptBW](#). Alternatively, for local polynomial estimators, the bandwidths above and below the cutoff 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` 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"`

`h1` Half-length of a two-sided CI based on estimate, so that the CI is given by `c(estimate-h1, estimate+h1)`. The CI is optimal if `opt.criterion="FLCI"`

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

`hp, hm` Bandwidths used above and below the cutoff

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

Armstrong, Timothy B., and Michal Kolesár. 2019. "Simple and Honest Confidence Intervals in Nonparametric Regression", *arXiv: 1606.01200*.

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.

## See Also

[RDOptBW](#)



## Examples

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

---

RDHonestBME	<i>CIs in sharp RD with discrete regressors under bounded misspecification error class</i>
-------------	--

---

## 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. For fuzzy or sharp RD, it can be a named vector of length two with names "p" and "m", in which case the bandwidth h["m"] is used for observations below the cutoff, and the bandwidth h["p"] is used for observations above the cutoff. 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+2th 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)")
```

---

RDOptBW

---

*Optimal Bandwidth Selection in Regression Discontinuity*


---

## Description

Estimate bandwidth for sharp RD based on local polynomial regression that optimizes either maximum mean squared error, or length or quantiles of excess length of a honest CI under second order Hölder or Taylor class.

## Usage

```
RDOptBW(formula, data, subset, weights, cutoff = 0, M,
         kern = "triangular", na.action, opt.criterion, bw.equal = TRUE,
         alpha = 0.05, beta = 0.8, 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 NAs. The default is set by the na.action setting of options (usually na.omit).
opt.criterion	<p>Optimality criterion that bandwidth is designed to optimize. The options are:</p> <p>"MSE" Finite-sample maximum MSE</p> <p>"FLCI" Length of (fixed-length) two-sided confidence intervals.</p> <p>"OCI" Given quantile of excess length of one-sided confidence intervals</p> <p>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.</p>
bw.equal	logical specifying whether bandwidths on either side of cutoff should be constrained to equal to each other.
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.
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	<p>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).</p> <p><b>"EHW"</b> Based on residuals from a local linear regression using a triangular kernel, and and bandwidth given by a rule-of-thumb bandwidth (for inference at a point, see <a href="#">ROTBW.fit</a>), or IK bandwidth (for fuzzy and sharp RD, see <a href="#">IKBW.fit</a>). For fuzzy RD, the IK bandwidth is based on the reduced-form regression.</p> <p><b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate <math>\sigma_i^2</math> by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope).</p> <p><b>"Silverman"</b> 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)</p> <p><b>"SilvermanNN"</b> Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals.</p> <p><b>"nn"</b> Use nearest neighbor estimates, without assuming homoskedasticity</p>

**Value**

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

`hp` bandwidth for observations above cutoff

`hm` bandwidth for observations below cutoff, equal to `hp` unless `bw.equal==FALSE`

`sigma2m, sigma2p` estimate of conditional variance just above and just below cutoff,  $\sigma_+^2(0)$  and  $\sigma_-^2(0)$

`f0` estimate of density of running variable at cutoff, if bandwidth computed using asymptotic method

`call` the matched call

`na.action` (where relevant) information on handling of missing data.

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

Armstrong, Timothy B., and Michal Kolesár. 2019. "Simple and Honest Confidence Intervals in Nonparametric Regression", *arXiv: 1606.01200*.

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.

**See Also**

[RDHonest](#)

**Examples**

```
## Use Lee dataset
RDOptBW(voteshare ~ margin, data = lee08, kern = "uniform",
        M = 0.1, opt.criterion = "MSE", sclass = "H")
```

---

RDSmoothnessBound	<i>Lower bound on smoothness constant <math>M</math> in RD designs</i>
-------------------	--

---

### 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

<code>d</code>	object of class "RDDData"
<code>s</code>	Number of support points that curvature estimates should average over
<code>separate</code>	If TRUE, report estimates separately for data above and below cutoff. If FALSE, report pooled estimates
<code>multiple</code>	If TRUE, use multiple curvature estimates. If FALSE, use a single estimate using only observations closest to the cutoff.
<code>alpha</code>	determines confidence level $1-\alpha$ .
<code>sclass</code>	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, Tim, 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 "RDDData"   |
| 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).<br><b>"EHW"</b> Based on residuals from a local linear regression using a triangular kernel, and and bandwidth given by a rule-of-thumb bandwidth (for inference at a point, see <a href="#">ROTBW.fit</a> ), or IK bandwidth (for fuzzy and sharp RD, see <a href="#">IKBW.fit</a> ). For fuzzy RD, the IK bandwidth is based on the reduced-form regression.<br><b>"demeaned"</b> Like EHW, but instead of using the regression residuals, estimate $\sigma_i^2$ by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope).<br><b>"Silverman"</b> 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)<br><b>"SilvermanNN"</b> Use Silverman's rule of thumb to pick the bandwidth, but use nearest neighbor estimates, rather than the residuals.<br><b>"nn"</b> Use nearest neighbor estimates, without assuming homoskedasticity |

### References

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

RDTOpt.fit

*Optimal inference in RD under Taylor class***Description**

Basic computing engine called by [RDHonest](#) to compute honest confidence intervals for optimal estimators in RD under second-order Taylor class.

**Usage**

```
RDTOpt.fit(d, M, opt.criterion, alpha = 0.05, beta = 0.5,
  se.method = "supplied.var", J = 3, se.initial = "IKEHW")
```

**Arguments**

d	object of class "RDDData"
M	Bound on second derivative of the conditional mean function.
opt.criterion	<p>Optimality criterion that bandwidth is designed to optimize. The options are:</p> <p>"MSE" Finite-sample maximum MSE</p> <p>"FLCI" Length of (fixed-length) two-sided confidence intervals.</p> <p>"OCI" Given quantile of excess length of one-sided confidence intervals</p> <p>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.</p>
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.method	<p>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:</p> <p>"nn" Nearest neighbor method</p> <p>"EHW" Eicker-Huber-White, with residuals from local regression (local polynomial estimators only).</p> <p>"demeaned" Like EHW, but instead of using the regression residuals, estimate <math>\sigma_i^2</math> by subtracting the estimated intercept from the outcome (and not subtracting the estimated slope). Local polynomial estimators only.</p> <p>"plugin" Plug-in estimate based on asymptotic variance. Local polynomial estimators in sharp RD only.</p> <p>"supplied.var" Use conditional variance supplied by sigma2 or d instead of computing residuals</p>
J	Number of nearest neighbors, if "nn" is specified in se.method.

`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 and bandwidth given by a rule-of-thumb bandwidth (for inference at a point, see [ROTBW.fit](#)), or IK bandwidth (for fuzzy and sharp RD, see [IKBW.fit](#)). For fuzzy RD, the IK bandwidth is based on the reduced-form regression.

**"demeaned"** Like EHW, but instead of using the regression residuals, estimate  $\sigma_i^2$  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

### Value

Returns an object of class "RDResults", see description in [RDHonest](#)

---

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

Lalive, R. (2008): "How Do Extended Benefits Affect Unemployment Duration? A Regression Discontinuity Approach." *Journal of Econometrics*, 142 (2): 785-806.

---

ROTBW.fit

*Rule of thumb bandwidth for inference at a point*

---

## Description

Calculate bandwidth for inference at a point with local linear regression using method in Fan and Gijbels (1996, Chapter 4.2).

## Usage

```
ROTBW.fit(d, kern = "triangular", order = 1, boundary = NULL)
```

## Arguments

d	object of class "LPPData"
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.
order	Order of local regression 1 for linear, 2 for quadratic.
boundary	Is point at a boundary?

## Value

ROT bandwidth

## 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.

## Examples

```
dp <- LPPData(lee08[lee08$margin>0, ], point=0)
bp1 <- ROTBW.fit(dp, kern="uniform", order=1)
```

# Index

## \*Topic **datasets**

- cghs, [2](#)
- headst, [11](#)
- kernC, [13](#)
- lee08, [14](#)
- rcp, [28](#)
- rebp, [40](#)
- RDHonestBME, [33](#)
- RDOptBW, [23](#), [24](#), [32](#), [34](#)
- RDSmoothnessBound, [37](#)
- RDTEfficiencyBound, [38](#)
- RDOpt.fit, [39](#)
- rebp, [40](#)
- ROTBW.fit, [7](#), [10](#), [17](#), [20](#), [23–25](#), [31](#), [35](#), [38](#),  
[40](#), [41](#)
- cghs, [2](#)
- CVb, [3](#)
- EqKern, [4](#)
- FRDData, [4](#), [15](#), [29](#)
- FRDHonest, [6](#), [21](#), [23](#)
- FRDOptBW, [8](#), [9](#), [23](#), [24](#)
- headst, [11](#)
- IKBW.fit, [7](#), [10](#), [12](#), [17](#), [20](#), [23–25](#), [31](#), [35](#), [38](#),  
[40](#)
- kernC, [13](#)
- KernMoment, [14](#)
- lee08, [14](#)
- LPPData, [5](#), [15](#), [29](#)
- LPPHonest, [16](#), [20](#), [21](#), [23](#)
- LPPOptBW, [18](#), [19](#), [23](#), [24](#)
- NPR\_MROT.fit, [21](#)
- NPRHonest.fit, [21](#)
- NPROptBW.fit, [23](#)
- NPRPrelimVar.fit, [25](#)
- NPRreg.fit, [26](#)
- plot\_RDscatter, [27](#)
- rcp, [28](#)
- RDDData, [5](#), [15](#), [29](#)
- RDHonest, [11](#), [21](#), [23](#), [30](#), [34](#), [36](#), [39](#), [40](#)