Package ‘rddapp’

July 23, 2025

Title Regression Discontinuity Design Application

Version 1.3.3

Author Ze Jin [aut],
Wang Liao [aut],
Irena Papst [aut],
Wenyu Zhang [aut],
Kimberly Hochstedler [aut],
Felix Thoemmes [aut, cre]

Maintainer Felix Thoemmes <`fjt36@cornell.edu`>

Description Estimation of both single- and multiple-assignment Regression Discontinuity Designs (RDDs). Provides both parametric (global) and non-parametric (local) estimation choices for both sharp and fuzzy designs, along with power analysis and assumption checks. Introductions to the underlying logic and analysis of RDDs are in Thistlethwaite, D. L., Campbell, D. T. (1960) <doi:10.1037/h0044319> and Lee, D. S., Lemieux, T. (2010) <doi:10.1257/jel.48.2.281>.

Depends R (>= 3.2.3)

Imports AER (>= 1.2-5), sandwich (>= 2.3-4), lmtest (>= 0.9-35), Formula (>= 1.2-1), shiny (>= 0.14), R.utils (>= 2.6.0), plot3D (>= 1.1.1), sp (>= 1.3.1), DT (>= 0.2)

Suggests foreign (>= 0.8-67), devtools (>= 1.12.0), testthat (>= 1.0.2), roxygen2 (>= 5.0.1), knitr (>= 1.14), rmarkdown (>= 1.1.9012)

VignetteBuilder knitr

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Collate 'attr_check.R' 'bw_ik09.R' 'bw_ik12.R' 'data.R' 'wt_kern.R' 'dc_test.R' 'treat_assign.R' 'wt_kern_bivariate.R' 'mfrd_est.R' 'var_center.R' 'rd_est.R' 'mrd_est.R' 'mrd_impute.R' 'mrd_power.R' 'mrd_sens_bw.R' 'mrd_sens_cutoff.R' 'plot.mfrd.R' 'predict.rd.R' 'plot.rd.R' 'print.mfrd.R' 'print.rd.R'

'rd_impute.R' 'rd_power.R' 'rd_sens_bw.R' 'rd_sens_cutoff.R'
'rd_type.R' 'rddapp-package.R' 'sens_plot.R' 'shiny_run.R'
'summary.mfrd.R' 'summary.mrd.R' 'summary.mrdi.R'
'summary.mrdp.R' 'summary.rd.R' 'summary.rdp.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2025-07-23 18:30:02 UTC

Contents

`attr_check` *Attrition Checks*

Description

`attr_check` reports missing data on treatment variable, assignment variable, and outcome. This is an internal function and is typically not directly invoked by the user. It can be accessed using the triple colon, as in rddapp:::attr_check().

Usage

```
attr_check(x1, y, t, x2 = NULL)
```

Arguments

`x1`	A numeric object containing the assignment variable.
`y`	A numeric object containing the outcome variable, with the same dimensionality as `x1`.
`t`	A numeric object containing the treatment variable (coded as 0 for untreated and 1 for treated), with the same dimensionality as `x1` and `y`.
`x2`	A numeric object containing the secondary assignment variable.

Value

`attr_check` returns a list containing the amount and percentage of missing data for all variables and subgroups, by treatment.

`bw_ik09` *Imbens-Kalyanaraman 2009 Optimal Bandwidth Calculation*

Description

`bw_ik09` calculates the Imbens-Kalyanaraman (2009) optimal bandwidth for local linear regression in regression discontinuity designs. It is based on the `IKbandwidth` function in the "rdd" package. This is an internal function and is typically not directly invoked by the user. It can be accessed using the triple colon, as in rddapp:::bw_ik09().

Usage

```
bw_ik09(X, Y, cutpoint = NULL, verbose = FALSE, kernel = "triangular")
```

Arguments

`X`	A numeric vector containing the running variable.
`Y`	A numeric vector containing the outcome variable.
`cutpoint`	A numeric value containing the cutpoint at which assignment to the treatment is determined. The default is 0.
`verbose`	A logical value indicating whether to print more information to the terminal. The default is `FALSE`.
`kernel`	A string indicating which kernel to use. Options are `"triangular"` (default and recommended), `"rectangular"`, `"epanechnikov"`, `"quartic"`, `"triweight"`, `"tricube"`, and `"cosine"`.

Value

`ik_bw09` returns a numeric value specifying the optimal bandwidth.

References

Imbens, G., Kalyanaraman, K. (2009). Optimal bandwidth choice for the regression discontinuity estimator (Working Paper No. 14726). National Bureau of Economic Research. `https://www.nber.org/papers/w14726`.

Drew Dimmery (2016). rdd: Regression Discontinuity Estimation. R package version 0.57. https://CRAN.R-project.org/package=rdd

---

`bw_ik12` *Imbens-Kalyanaraman 2012 Optimal Bandwidth Calculation*

Description

`bw_ik12` calculates the Imbens-Kalyanaraman (2012) optimal bandwidth for local linear regression in regression discontinuity designs. It is based on a function in the "rddtools" package. This is an internal function and is typically not directly invoked by the user. It can be accessed using the triple colon, as in rddapp:::bw_ik12().

Usage

```
bw_ik12(X, Y, cutpoint = NULL, verbose = FALSE, kernel = "triangular")
```

Arguments

`X`	A numeric vector containing the running variable.
`Y`	A numeric vector containing the outcome variable.
`cutpoint`	A numeric value containing the cutpoint at which assignment to the treatment is determined. The default is 0.
`verbose`	A logical value indicating whether to print more information to the terminal. The default is `FALSE`.

`kernel`	A string indicating which kernel to use. Options are `"triangular"` (default and recommended), `"rectangular"`, `"epanechnikov"`, `"quartic"`, `"triweight"`, `"tricube"`, and `"cosine"`.

Value

`ik_bw12` returns a numeric value specifying the optimal bandwidth.

References

Imbens, G., Kalyanaraman, K. (2012). Optimal bandwidth choice for the regression discontinuity estimator. The Review of Economic Studies, 79(3), 933-959. `https://academic.oup.com/restud/article/79/3/933/1533189`.

Stigler, M. and B. Quast, B (2016). rddtools: A toolbox for regression discontinuity in R.

`CARE` *Carolina Abecedarian Project and the Carolina Approach to Responsive Education (CARE), 1972-1992*

Description

A dataset containing a subset of children from the CARE trial on early childhood intervention. The randomized controlled trial was subsetted to mimic a regression-discontinuity design in which treatment was assigned only to mothers whose IQ was smaller than 85.

Usage

```
CARE
```

Format

A data frame with 81 rows and 5 variables:

SUBJECT Unique ID variable

DC_TRT Day Care (Preschool) Treatment Group, 1 = Treatment, 0 = Control

APGAR5 APGAR ("Appearance, Pulse, Grimace, Activity, and Respiration") score at 5 minutes after birth

MOMWAIS0 Biological mother's WAIS (Wechsler Adult Intelligence Scale) full-scale score at subject's birth

SBIQ48 Subject's Stanford Binet IQ score at 48 months

Source

`http://www.icpsr.umich.edu/icpsrweb/ICPSR/studies/4091`

Examples

```
data("CARE")
head(CARE)
```

`dc_test` *McCrary Sorting Test*

Description

`dc_test` implements the McCrary (2008) sorting test to identify violations of assignment rules. It is based on the `DCdensity` function in the "rdd" package.

Usage

```
dc_test(
  runvar,
  cutpoint,
  bin = NULL,
  bw = NULL,
  verbose = TRUE,
  plot = TRUE,
  ext.out = FALSE,
  htest = FALSE,
  level = 0.95,
  digits = max(3, getOption("digits") - 3),
  timeout = 30
)
```

Arguments

`runvar`	A numeric vector containing the running variable.
`cutpoint`	A numeric value containing the cutpoint at which assignment to the treatment is determined. The default is 0.
`bin`	A numeric value containing the binwidth. The default is `2*sd(runvar)*length(runvar)^(-.5)`.
`bw`	A numeric value containing bandwidth to use. If no bandwidth is supplied, the default uses bandwidth selection calculation from McCrary (2008).
`verbose`	A logical value indicating whether to print diagnostic information to the terminal. The default is `TRUE`.
`plot`	A logical value indicating whether to plot the histogram and density estimations The default is `TRUE`. The user may wrap this function in additional graphical options to modify the plot.
`ext.out`	A logical value indicating whether to return extended output. The default is `FALSE`. When `FALSE` `dc_test` will return only the p-value of the test, but will print more information. When `TRUE`, `dc_test` will return and print the additional information documented below.

`htest`	A logical value indicating whether to return an `"htest"` object compatible with base R's hypothesis test output. The default is `FALSE`.
`level`	A numerical value between 0 and 1 specifying the confidence level for confidence intervals. The default is 0.95.
`digits`	A non-negative integer specifying the number of digits to display in all output. The default is `max(3, getOption("digits") - 3)`.
`timeout`	A non-negative numerical value specifying the maximum number of seconds that expressions in the function are allowed to run. The default is 30. Specify `Inf` to run all expressions to completion.

Value

If `ext.out` is `FALSE`, `dc_test` returns a numeric value specifying the p-value of the McCrary (2008) sorting test. Additional output is enabled when `ext.out` is `TRUE`. In this case, `dc_test` returns a list with the following elements:

`theta`	The estimated log difference in heights of the density curve at the cutpoint.
`se`	The standard error of `theta`.
`z`	The z statistic of the test.
`p`	The p-value of the test. A p-value below the significance threshold indicates that the user can reject the null hypothesis of no sorting.
`binsize`	The calculated size of bins for the test.
`bw`	The calculated bandwidth for the test.
`cutpoint`	The cutpoint used.
`data`	A dataframe for the binning of the histogram. Columns are `cellmp` (the midpoints of each cell) and `cellval` (the normalized height of each cell).

References

McCrary, J. (2008). Manipulation of the running variable in the regression discontinuity design: A density test. Journal of Econometrics, 142(2), 698-714. doi:10.1016/j.jeconom.2007.05.005.

Drew Dimmery (2016). rdd: Regression Discontinuity Estimation. R package version 0.57. https://CRAN.R-project.org/package=rdd

Examples

```
set.seed(12345)
# No discontinuity
x <- runif(1000, -1, 1)
dc_test(x, 0)

# Discontinuity
x <- runif(1000, -1, 1)
x <- x + 2 * (runif(1000, -1, 1) > 0 & x < 0)
dc_test(x, 0)
```

`mfrd_est` *Multivariate Frontier Regression Discontinuity Estimation*

Description

`mfrd_est` implements the frontier approach for multivariate regression discontinuity estimation in Wong, Steiner and Cook (2013). It is based on the MFRDD code in Stata from Wong, Steiner, and Cook (2013).

Usage

```
mfrd_est(
  y,
  x1,
  x2,
  c1,
  c2,
  t.design = NULL,
  local = 0.15,
  front.bw = NA,
  m = 10,
  k = 5,
  kernel = "triangular",
  ngrid = 250,
  margin = 0.03,
  boot = NULL,
  cluster = NULL,
  stop.on.error = TRUE
)
```

Arguments

`y`	A numeric object containing outcome variable.
`x1`	A numeric object containing the first assignment variable.
`x2`	A numeric object containing the second assignment variable.
`c1`	A numeric value containing the cutpoint at which assignment to the treatment is determined for `x1`.
`c2`	A numeric value containing the cutpoint at which assignment to the treatment is determined for `x2`.
`t.design`	A character vector of length 2 specifying the treatment option according to design. The first entry is for `x1` and the second entry is for `x2`. Options are `"g"` (treatment is assigned if `x1` is greater than its cutoff), `"geq"` (treatment is assigned if `x1` is greater than or equal to its cutoff), `"l"` (treatment is assigned if `x1` is less than its cutoff), and `"leq"` (treatment is assigned if `x1` is less than or equal to its cutoff). The same options are available for `x2`.

`local`	A non-negative numeric value specifying the range of neighboring points around the cutoff on the standardized scale, for each assignment variable. The default is 0.15.
`front.bw`	A non-negative numeric vector of length 3 specifying the bandwidths at which to estimate the RD for each of three effects models (complete model, heterogeneous treatment model, and treatment only model) detailed in Wong, Steiner, and Cook (2013). If `NA`, `front.bw` will be determined by cross-validation. The default is `NA`.
`m`	A non-negative integer specifying the number of uniformly-at-random samples to draw as search candidates for `front.bw`, if `front.bw` is `NA`. The default is 10.
`k`	A non-negative integer specifying the number of folds for cross-validation to determine `front.bw`, if `front.bw` is `NA`. The default is 5.
`kernel`	A string indicating which kernel to use. Options are `"triangular"` (default and recommended), `"rectangular"`, `"epanechnikov"`, `"quartic"`, `"triweight"`, `"tricube"`, and `"cosine"`.
`ngrid`	A non-negative integer specifying the number of non-zero grid points on each assignment variable, which is also the number of zero grid points on each assignment variable. The default is 250. The value used in Wong, Steiner and Cook (2013) is 2500, which may cause long computational time.
`margin`	A non-negative numeric value specifying the range of grid points beyond the minimum and maximum of sample points on each assignment variable. This grid is used to impute potential outcomes along the frontier, as in Wong, Steiner, and Cook (2013). The default is 0.03.
`boot`	An optional non-negative integer specifying the number of bootstrap samples to obtain standard error of estimates.
`cluster`	An optional vector of length n specifying clusters within which the errors are assumed to be correlated. This will result in reporting cluster robust SEs. It is suggested that data with a discrete running variable be clustered by each unique value of the running variable (Lee and Card, 2008).
`stop.on.error`	A logical value indicating whether to remove bootstraps which cause error in the `integrate` function. If `TRUE`, bootstraps which cause error are removed and resampled until the specified number of bootstrap samples are acquired. If `FALSE`, bootstraps which cause error are not removed. The default is `TRUE`.

Value

`mfrd_est` returns an object of class `"mfrd"`. The functions `summary` and `plot` are used to obtain and print a summary and plot of the estimated regression discontinuity. The object of class `mfrd` is a list containing the following components:

`w`	Numeric vector specifying the weight of frontier 1 and frontier 2, respectively.
`est`	Numeric matrix of the estimate of the discontinuity in the outcome under a complete model (no prefix), heterogeneous treatment (ht) effects model, and treatment (t) only model, for the parametric case and for each corresponding bandwidth. Estimates with suffix "ev1" and "ev2" correspond to expected values for each frontier, under a given model. Estimates with suffix "ate" correspond to average treatment effects across both frontiers, under a given model.

`d`	Numeric matrix of the effect size (Cohen's d) for estimate.
`se`	Numeric matrix of the standard error for each corresponding bandwidth, if applicable.
`m_s`	A list containing estimates for the complete model, under parametric and non-parametric (optimal, half, and double bandwidth) cases. A list of coefficient estimates, residuals, effects, weights (in the non-parametric case), `lm` output (rank of the fitted linear model, fitted values, assignments for the design matrix, qr for linear fit, residual degrees of freedom, levels of the x value, function call, and terms), and output data frame are returned for each model.
`m_h`	A list containing estimates for the heterogeneous treatments model, under parametric and non-parametric (optimal, half, and double bandwidth) cases. A list of coefficient estimates, residuals, effects, weights (in the non-parametric case), `lm` output (rank of the fitted linear model, fitted values, assignments for the design matrix, qr for linear fit, residual degrees of freedom, levels of the x value, function call, and terms), and output data frame are returned for each model.
`m_t`	A list containing estimates for the treatment only model, under parametric and non-parametric (optimal, half, and double bandwidth) cases. A list of coefficient estimates, residuals, effects, weights (in the non-parametric case), `lm` output (rank of the fitted linear model, fitted values, assignments for the design matrix, qr for linear fit, residual degrees of freedom, levels of the x value, function call, and terms), and output data frame are returned for each model.
`dat_h`	A list containing four data frames, one for each case: parametric or non-parametric (optimal, half, and double bandwidth). Each data frame contains functions and densities for each frontier and treatment model.
`dat`	A data frame containing the outcome (`y`) and each input (`x1`, `x2`) for each observation. The data frame also contains indicators of being within the `local` boundary of the `cutpoint` for x1 and x2 (`x1res`, `x2res`), scaled (`zx1`, `zx2`) and centered `x1` and `x2` values (`zcx1`, `zcx2`), and treatment indicators for overall treatment (`tr`) based on treatment assignment from x1 (`tr1`), x2 (`tr2`), and both assignment variables (`trb`).
`obs`	List of the number of observations used in each model.
`impute`	A logical value indicating whether multiple imputation is used or not.
`call`	The matched call.
`front.bw`	Numeric vector of each bandwidth used to estimate the density at the frontier for the three effects models (complete model, heterogeneous treatment model, and treatment only model) detailed in Wong, Steiner, and Cook (2013).

References

Wong, V., Steiner, P, and Cook, T. (2013). Analyzing regression discontinuity designs with multiple assignment variables: A comparative study of four estimation methods. Journal of Educational and Behavioral Statistics, 38(2), 107-141. doi:10.3102/1076998611432172.

Lee, D. and Card, D. (2008). A Regression discontinuity inference with specification error. Journal of Econometrics, 142(2), 655-674. doi:10.1016/j.jeconom.2007.05.003.

Examples

```
set.seed(12345)
x1 <- runif(1000, -1, 1)
x2 <- runif(1000, -1, 1)
cov <- rnorm(1000)
y <- 3 + 2 * (x1 >= 0) + 3 * cov + 10 * (x2 >= 0) + rnorm(1000)
mfrd_est(y = y, x1 = x1, x2 = x2, c1 = 0, c2 = 0, t.design = c("geq", "geq"))
```

`mrd_est` *Multivariate Regression Discontinuity Estimation*

Description

`mrd_est` estimates treatment effects in a multivariate regression discontinuity design (MRDD) with two assignment variables, including the frontier average treatment effect (`tau_MRD`) and frontier-specific effects (`tau_R` and `tau_M`) simultaneously.

Usage

```
mrd_est(
  formula,
  data,
  subset = NULL,
  cutpoint = NULL,
  bw = NULL,
  front.bw = NA,
  m = 10,
  k = 5,
  kernel = "triangular",
  se.type = "HC1",
  cluster = NULL,
  verbose = FALSE,
  less = FALSE,
  est.cov = FALSE,
  est.itt = FALSE,
  local = 0.15,
  ngrid = 250,
  margin = 0.03,
  boot = NULL,
  method = c("center", "univ", "front"),
  t.design = NULL,
  stop.on.error = TRUE
)
```

Arguments

`formula`	The formula of the MRDD; a symbolic description of the model to be fitted. This is supplied in the format of `y ~ x1 + x2` for a simple sharp MRDD or `y ~ x1 + x2 | c1 + c2` for a sharp MRDD with two covariates. A fuzzy MRDD may be specified as `y ~ x1 + x2 + z` where `x1` is the first running variable, `x2` is the second running variable, and `z` is the endogenous treatment variable. Covariates are then included in the same manner as in a sharp MRDD.
`data`	An optional data frame containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`.
`subset`	An optional vector specifying a subset of observations to be used in the fitting process.
`cutpoint`	A numeric vector of length 2 containing the cutpoints at which assignment to the treatment is determined. The default is c(0, 0).
`bw`	A vector specifying the bandwidths at which to estimate the RD for non-parametric models. Possible values are `"IK09"`, `"IK12"`, or a user-specified non-negative numeric vector containing the bandwidths at which to estimate the RD. The default is `"IK12"`. If `bw` is `"IK12"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2012 method. If `bw` is `"IK09"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2009 method. Then, the RD is estimated with that bandwidth, half that bandwidth, and twice that bandwidth. If only a single value is passed into the function, the RD will similarly be estimated at that bandwidth, half that bandwidth, and twice that bandwidth.
`front.bw`	A non-negative numeric vector of length 3 specifying the bandwidths at which to estimate the RD for each of three effects models (complete model, heterogeneous treatment model, and treatment only model) detailed in Wong, Steiner, and Cook (2013). If `NA`, `front.bw` will be determined by cross-validation. The default is `NA`.
`m`	A non-negative integer specifying the number of uniformly-at-random samples to draw as search candidates for `front.bw`, if `front.bw` is `NA`. The default is 10.
`k`	A non-negative integer specifying the number of folds for cross-validation to determine `front.bw`, if `front.bw` is `NA`. The default is 5.
`kernel`	A string indicating which kernel to use. Options are `"triangular"` (default and recommended), `"rectangular"`, `"epanechnikov"`, `"quartic"`, `"triweight"`, `"tricube"`, and `"cosine"`.
`se.type`	This specifies the robust standard error calculation method to use, from the "sandwich" package. Options are, as in `vcovHC`, `"HC3"`, `"const"`, `"HC"`, `"HC0"`, `"HC1"`, `"HC2"`, `"HC4"`, `"HC4m"`, `"HC5"`. The default is `"HC1"`. This option is overridden by `cluster`.
`cluster`	An optional vector of length n specifying clusters within which the errors are assumed to be correlated. This will result in reporting cluster robust SEs. This option overrides anything specified in `se.type`. It is suggested that data with a discrete running variable be clustered by each unique value of the running variable (Lee and Card, 2008).
`verbose`	A logical value indicating whether to print additional information to the terminal, including results of instrumental variable regression, and outputs from background regression models. The default is `FALSE`.

`less`	Logical. If `TRUE`, return the estimates of parametric linear and optimal bandwidth non-parametric models only. If `FALSE` return the estimates of linear, quadratic, and cubic parametric models and optimal, half and double bandwidths in non-parametric models. The default is `FALSE`.
`est.cov`	Logical. If `TRUE`, the estimates of covariates will be included. If `FALSE`, the estimates of covariates will not be included. The default is `FALSE`. This option is not applicable if method is `"front"`.
`est.itt`	Logical. If `TRUE`, the estimates of intent-to-treat (ITT) will be returned. If `FALSE`, the estimates of ITT will not be returned. The default is `FALSE`. This option is not applicable if method is `"front"`.
`local`	A non-negative numeric value specifying the range of neighboring points around the cutoff on the standardized scale, for each assignment variable. The default is 0.15.
`ngrid`	A non-negative integer specifying the number of non-zero grid points on each assignment variable, which is also the number of zero grid points on each assignment variable. The default is 250. The value used in Wong, Steiner and Cook (2013) is 2500, which may cause long computational time.
`margin`	A non-negative numeric value specifying the range of grid points beyond the minimum and maximum of sample points on each assignment variable. The default is 0.03.
`boot`	An optional non-negative integer specifying the number of bootstrap samples to obtain standard error of estimates. This argument is not optional if method is `"front"`.
`method`	A string specifying the method to estimate the RD effect. Options are `"center"`, `"univ"`, `"front"`, based on the centering, univariate, and frontier approaches (respectively) from Wong, Steiner, and Cook (2013).
`t.design`	A character vector of length 2 specifying the treatment option according to design. The first entry is for `x1` and the second entry is for `x2`. Options are `"g"` (treatment is assigned if `x1` is greater than its cutoff), `"geq"` (treatment is assigned if `x1` is greater than or equal to its cutoff), `"l"` (treatment is assigned if `x1` is less than its cutoff), and `"leq"` (treatment is assigned if `x1` is less than or equal to its cutoff). The same options are available for `x2`.
`stop.on.error`	A logical value indicating whether to remove bootstraps which cause error in the `integrate` function. If `TRUE`, bootstraps which cause error are removed and resampled until the specified number of bootstrap samples are acquired. If `FALSE`, bootstraps which cause error are not removed. The default is `TRUE`.

Value

`mrd_est` returns an object of class `"mrd"`. The function `summary` is used to obtain and print a summary of the estimated regression discontinuity. The object of class `mrd` is a list containing the following components for each estimated treatment effect, `tau_MRD` or `tau_R` and `tau_M`:

`type`	A string denoting either `"sharp"` or `"fuzzy"` RDD.
`call`	The matched call.

`est`	Numeric vector of the estimate of the discontinuity in the outcome under a sharp MRDD or the Wald estimator in the fuzzy MRDD, for each corresponding bandwidth, if applicable.
`se`	Numeric vector of the standard error for each corresponding bandwidth, if applicable.
`ci`	The matrix of the 95 for each corresponding bandwidth, if applicable.
`bw`	Numeric vector of each bandwidth used in estimation.
`z`	Numeric vector of the z statistic for each corresponding bandwidth, if applicable.
`p`	Numeric vector of the p-value for each corresponding bandwidth, if applicable.
`obs`	Vector of the number of observations within the corresponding bandwidth, if applicable.
`cov`	The names of covariates.
`model`	For a sharp design, a list of the `lm` objects is returned. For a fuzzy design, a list of lists is returned, each with two elements: `firststage`, the first stage `lm` object, and `iv`, the `ivreg` object. A model is returned for each parametric and non-parametric case and corresponding bandwidth.
`frame`	Returns the model frame used in fitting.
`na.action`	The observations removed from fitting due to missingness.
`impute`	A logical value indicating whether multiple imputation is used or not.
`d`	Numeric vector of the effect size (Cohen's d) for each estimate.

References

Wong, V. C., Steiner, P. M., Cook, T. D. (2013). Analyzing regression-discontinuity designs with multiple assignment variables: A comparative study of four estimation methods. Journal of Educational and Behavioral Statistics, 38(2), 107-141. `https://journals.sagepub.com/doi/10.3102/1076998611432172`.

Imbens, G., Kalyanaraman, K. (2009). Optimal bandwidth choice for the regression discontinuity estimator (Working Paper No. 14726). National Bureau of Economic Research. `https://www.nber.org/papers/w14726`.

Imbens, G., Kalyanaraman, K. (2012). Optimal bandwidth choice for the regression discontinuity estimator. The Review of Economic Studies, 79(3), 933-959. `https://academic.oup.com/restud/article/79/3/933/1533189`.

Lee, D. S., Card, D. (2010). Regression discontinuity inference with specification error. Journal of Econometrics, 142(2), 655-674. doi:10.1016/j.jeconom.2007.05.003.

Lee, D. S., Lemieux, T. (2010). Regression Discontinuity Designs in Economics. Journal of Economic Literature, 48(2), 281-355. doi:10.1257/jel.48.2.281.

Zeileis, A. (2006). Object-oriented computation of sandwich estimators. Journal of Statistical Software, 16(9), 1-16. doi:10.18637/jss.v016.i09

Examples

```
set.seed(12345)
x1 <- runif(1000, -1, 1)
x2 <- runif(1000, -1, 1)
cov <- rnorm(1000)
y <- 3 + 2 * (x1 >= 0) + 3 * cov + 10 * (x2 >= 0) + rnorm(1000)
# centering
mrd_est(y ~ x1 + x2 | cov, method = "center", t.design = c("geq", "geq"))
# univariate
mrd_est(y ~ x1 + x2 | cov, method = "univ", t.design = c("geq", "geq"))
# frontier
mrd_est(y ~ x1 + x2 | cov, method = "front", t.design = c("geq", "geq"))
```

`mrd_impute` *Multiple Imputation of Multivariate Regression Discontinuity Estimation*

Description

`mrd_impute` estimates treatment effects in a multivariate regression discontinuity design (MRDD) with imputed missing values.

Usage

```
mrd_impute(
  formula,
  data,
  subset = NULL,
  cutpoint = NULL,
  bw = NULL,
  front.bw = NA,
  m = 10,
  k = 5,
  kernel = "triangular",
  se.type = "HC1",
  cluster = NULL,
  impute = NULL,
  verbose = FALSE,
  less = FALSE,
  est.cov = FALSE,
  est.itt = FALSE,
  local = 0.15,
  ngrid = 250,
  margin = 0.03,
  boot = NULL,
  method = c("center", "univ", "front"),
  t.design = NULL,
  stop.on.error = TRUE
)
```

Arguments

`formula`	The formula of the MRDD; a symbolic description of the model to be fitted. This is supplied in the format of `y ~ x1 + x2` for a simple sharp MRDD or `y ~ x1 + x2 | c1 + c2` for a sharp MRDD with two covariates. A fuzzy MRDD may be specified as `y ~ x1 + x2 + z` where `x1` is the first running variable, `x2` is the second running variable, and `z` is the endogenous treatment variable. Covariates are then included in the same manner as in a sharp MRDD.
`data`	An optional data frame containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`.
`subset`	An optional vector specifying a subset of observations to be used in the fitting process.
`cutpoint`	A numeric vector of length 2 containing the cutpoints at which assignment to the treatment is determined. The default is `c(0, 0)`.
`bw`	A vector specifying the bandwidths at which to estimate the RD. Possible values are `"IK09"`, `"IK12"`, and a user-specified non-negative numeric vector specifying the bandwidths at which to estimate the RD. The default is `"IK12"`. If `bw` is `"IK12"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2012 method. If `bw` is `"IK09"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2009 method. Then the RD is estimated with that bandwidth, half that bandwidth, and twice that bandwidth. If only a single value is passed into the function, the RD will similarly be estimated at that bandwidth, half that bandwidth, and twice that bandwidth.
`front.bw`	A non-negative numeric vector of length 3 specifying the bandwidths at which to estimate the RD for each of three effects models (complete model, heterogeneous treatment model, and treatment only model) detailed in Wong, Steiner, and Cook (2013). If `NA`, `front.bw` will be determined by cross-validation. The default is `NA`.
`m`	A non-negative integer specifying the number of uniformly-at-random samples to draw as search candidates for `front.bw`, if `front.bw` is `NA`. The default is 10.
`k`	A non-negative integer specifying the number of folds for cross-validation to determine `front.bw`, if `front.bw` is `NA`. The default is 5.
`kernel`	A string indicating which kernel to use. Options are `"triangular"` (default and recommended), `"rectangular"`, `"epanechnikov"`, `"quartic"`, `"triweight"`, `"tricube"`, and `"cosine"`.
`se.type`	This specifies the robust standard error calculation method to use, from the "sandwich" package. Options are, as in `vcovHC`, `"HC3"`, `"const"`, `"HC"`, `"HC0"`, `"HC1"`, `"HC2"`, `"HC4"`, `"HC4m"`, `"HC5"`. The default is `"HC1"`. This option is overridden by `cluster`.
`cluster`	An optional vector of length n specifying clusters within which the errors are assumed to be correlated. This will result in reporting cluster robust SEs. This option overrides anything specified in `se.type`. It is suggested that data with a discrete running variable be clustered by each unique value of the running variable (Lee and Card, 2008).
`impute`	An optional vector of length n containing a grouping variable that specifies the imputed variables with missing values.

`verbose`	A logical value indicating whether to print additional information to the terminal. The default is `FALSE`.
`less`	Logical. If `TRUE`, return the estimates of parametric linear and optimal bandwidth non-parametric models only. If `FALSE` return the estimates of linear, quadratic, and cubic parametric models and optimal, half and double bandwidths in non-parametric models. The default is `FALSE`.
`est.cov`	Logical. If `TRUE`, the estimates of covariates will be included. If `FALSE`, the estimates of covariates will not be included. The default is `FALSE`. This option is not applicable if method is `"front"`.
`est.itt`	Logical. If `TRUE`, the estimates of intent-to-treat (ITT) will be returned. If `FALSE`, the estimates of ITT will not be returned. The default is `FALSE`. This option is not applicable if method is `"front"`.
`local`	A non-negative numeric value specifying the range of neighboring points around the cutoff on the standardized scale, for each assignment variable. The default is 0.15.
`ngrid`	A non-negative integer specifying the number of non-zero grid points on each assignment variable, which is also the number of zero grid points on each assignment variable. The default is 250. The value used in Wong, Steiner and Cook (2013) is 2500, which may cause long computational time.
`margin`	A non-negative numeric value specifying the range of grid points beyond the minimum and maximum of sample points on each assignment variable. The default is 0.03.
`boot`	An optional non-negative integer specifying the number of bootstrap samples to obtain standard error of estimates. This argument is not optional if method is `"front"`.
`method`	A string specifying the method to estimate the RD effect. Options are `"center"`, `"univ"`, `"front"`, based on the centering, univariate, and frontier approaches (respectively) from Wong, Steiner, and Cook (2013).
`t.design`	A character vector of length 2 specifying the treatment option according to design. The first entry is for `x1` and the second entry is for `x2`. Options are `"g"` (treatment is assigned if `x1` is greater than its cutoff), `"geq"` (treatment is assigned if `x1` is greater than or equal to its cutoff), `"l"` (treatment is assigned if `x1` is less than its cutoff), and `"leq"` (treatment is assigned if `x1` is less than or equal to its cutoff). The same options are available for `x2`.
`stop.on.error`	A logical value indicating whether to remove bootstraps which cause error in the `integrate` function. If `TRUE`, bootstraps which cause error are removed and resampled until the specified number of bootstrap samples are acquired. If `FALSE`, bootstraps which cause error are not removed. The default is `TRUE`.

Value

`mrd_impute` returns an object of class "`mrd`" or "`mrdi`" for `"front"` method. The function `summary` is used to obtain and print a summary of the estimated regression discontinuity. The object of class `mrd` is a list containing the following components for each estimated treatment effect, `tau_MRD` or `tau_R` and `tau_M`:

`call`	The matched call.
`type`	A string denoting either `"sharp"` or `"fuzzy"` RDD.
`cov`	The names of covariates.
`bw`	Numeric vector of each bandwidth used in estimation.
`obs`	Vector of the number of observations within the corresponding bandwidth.
`model`	For a sharp design, a list of the `lm` objects is returned. For a fuzzy design, a list of lists is returned, each with two elements: `firststage`, the first stage `lm` object, and `iv`, the `ivreg` object. A model is returned for each parametric and non-parametric case and corresponding bandwidth.
`frame`	Returns the model frame used in fitting.
`na.action`	The observations removed from fitting due to missingness.
`est`	Numeric vector of the estimate of the discontinuity in the outcome under a sharp MRDD or the Wald estimator in the fuzzy MRDD, for each corresponding bandwidth.
`d`	Numeric vector of the effect size (Cohen's d) for each estimate.
`se`	Numeric vector of the standard error for each corresponding bandwidth.
`z`	Numeric vector of the z statistic for each corresponding bandwidth.
`df`	Numeric vector of the degrees of freedom computed using Barnard and Rubin (1999) adjustment for imputation.
`p`	Numeric vector of the p-value for each corresponding bandwidth.
`ci`	The matrix of the 95 for each corresponding bandwidth.
`impute`	A logical value indicating whether multiple imputation is used or not.

References

Wong, V. C., Steiner, P. M., Cook, T. D. (2013). Analyzing regression-discontinuity designs with multiple assignment variables: A comparative study of four estimation methods. Journal of Educational and Behavioral Statistics, 38(2), 107-141. https://journals.sagepub.com/doi/10.3102/1076998611432172.

Lee, D. S., Lemieux, T. (2010). Regression Discontinuity Designs in Economics. Journal of Economic Literature, 48(2), 281-355. doi:10.1257/jel.48.2.281.

Lee, D. S., Card, D. (2010). Regression discontinuity inference with specification error. Journal of Econometrics, 142(2), 655-674. doi:10.1016/j.jeconom.2007.05.003.

Barnard, J., Rubin, D. (1999). Small-Sample Degrees of Freedom with Multiple Imputation. Biometrika, 86(4), 948-55.

Examples

```
set.seed(12345)
x1 <- runif(300, -1, 1)
x2 <- runif(300, -1, 1)
cov <- rnorm(300)
y <- 3 + 2 * (x1 >= 0) + 3 * cov + 10 * (x2 >= 0) + rnorm(300)
imp <- rep(1:3, each = 100)
```

```
# all examples below have smaller numbers of m to keep run-time low
# centering
mrd_impute(y ~ x1 + x2 | cov, impute = imp, method = "center", t.design = c("geq", "geq"), m = 3)
# univariate
mrd_impute(y ~ x1 + x2 | cov, impute = imp, method = "univ", t.design = c("geq", "geq"), m = 3)
# frontier - don't run due to computation time
## Not run: mrd_impute(y ~ x1 + x2 | cov, impute = imp, method = "front",
                    boot = 1000, t.design = c("geq", "geq"), m = 3)
## End(Not run)
```

---

mrd_power *Power Analysis of Multivariate Regression Discontinuity*

Description

`mrd_power` computes the empirical probability that a resulting parameter estimate of the MRD is significant, i.e. the empirical power (1 - beta).

Usage

```
mrd_power(
  num.rep = 100,
  sample.size = 100,
  x1.dist = "normal",
  x1.para = c(0, 1),
  x2.dist = "normal",
  x2.para = c(0, 1),
  x1.cut = 0,
  x2.cut = 0,
  x1.fuzzy = c(0, 0),
  x2.fuzzy = c(0, 0),
  x1.design = NULL,
  x2.design = NULL,
  coeff = c(0.1, 0.5, 0.5, 1, rep(0.1, 9)),
  eta.sq = 0.5,
  alpha.list = c(0.001, 0.01, 0.05)
)
```

Arguments

`num.rep`	A non-negative integer specifying the number of repetitions used to calculate the empirical power. The default is 100.
`sample.size`	A non-negative integer specifying the number of observations in each sample. The default is 100.
`x1.dist`	A string specifying the distribution of the first assignment variable, `x1`. Options are `"normal"` and `"uniform"`. The default is the `"normal"` distribution.

`x1.para`	A numeric vector of length 2 specifying parameters of the distribution of the first assignment variable, `x1`. If `x1.dist` is `"normal"`, then `x1.para` includes the mean and standard deviation of the normal distribution. If `x1.dist` is `"uniform"`, then `x1.para` includes the upper and lower boundaries of the uniform distribution. The default is `c(0,1)`.
`x2.dist`	A string specifying the distribution of the second assignment variable, `x2`. Options are `"normal"` and `"uniform"`. The default is the `"normal"` distribution.
`x2.para`	A numeric vector of length 2 specifying parameters of the distribution of the second assignment variable, `x2`. If `x2.dist` is `"normal"`, then `x2.para` includes the mean and standard deviation of the normal distribution. If `x2.dist` is `"uniform"`, then `x2.para` includes the upper and lower boundaries of the uniform distribution. The default is `c(0,1)`.
`x1.cut`	A numeric value containing the cutpoint at which assignment to the treatment is determined for the first assignment variable, `x1`. The default is 0.
`x2.cut`	A numeric value containing the cutpoint at which assignment to the treatment is determined for the second assignment variable, `x2`. The default is 0.
`x1.fuzzy`	A numeric vector of length 2 specifying the probabilities to be assigned to the control condition, in terms of the first assignment variable, `x1`, for individuals in the treatment based on the cutoff, and to treatment for individuals in the control condition based on the cutoff. For a sharp design, both entries are 0. For a fuzzy design, the first entry is the probability to be assigned to control for individuals above the cutpoint, and the second entry is the probability to be assigned to treatment for individuals below the cutpoint. The default is `c(0,0)`, indicating a sharp design.
`x2.fuzzy`	A numeric vector of length 2 specifying the probabilities to be assigned to the control, in terms of the second assignment variable, `x2`, for individuals in the treatment based on the cutoff, and to treatment for individuals in the control based on the cutoff. For a sharp design, both entries are 0. For a fuzzy design, the first entry is the probability to be assigned to control for individuals above the cutpoint, and the second entry is the probability to be assigned to treatment for individuals below the cutpoint. The default is `c(0,0)`, indicating a sharp design.
`x1.design`	A string specifying the treatment option according to design for `x1`. Options are `"g"` (treatment is assigned if `x1` is greater than its cutoff), `"geq"` (treatment is assigned if `x1` is greater than or equal to its cutoff), `"l"` (treatment is assigned if `x1` is less than its cutoff), and `"leq"` (treatment is assigned if `x1` is less than or equal to its cutoff).
`x2.design`	A string specifying the treatment option according to design for `x2`. Options are `"g"` (treatment is assigned if `x2` is greater than its cutoff), `"geq"` (treatment is assigned if `x2` is greater than or equal to its cutoff), `"l"` (treatment is assigned if `x2` is less than its cutoff), and `"leq"` (treatment is assigned if `x2` is less than or equal to its cutoff).
`coeff`	A numeric vector specifying coefficients of variables in the linear model to generate data. Coefficients are in the following order: • The 1st entry is the intercept.

- The 2nd entry is the slope of treatment 1, i.e. treatment effect 1.
- The 3rd entry is the slope of treatment 2, i.e. treatment effect 2.
- The 4th entry is the slope of treatment, i.e. treatment effect.
- The 5th entry is the slope of assignment 1.
- The 6th entry is the slope of assignment 2.
- The 7th entry is the slope of interaction between assignment 1 and assignment 2.
- The 8th entry is the slope of interaction between treatment 1 and assignment 1.
- The 9th entry is the slope of interaction between treatment 2 and assignment 1.
- The 10th entry is the slope of interaction between treatment 1 and assignment 2.
- The 11th entry is the slope of interaction between treatment 2 and assignment 2.
- The 12th entry is the slope of interaction between treatment 1, assignment 1 and assignment 2.
- The 13th entry is the slope of interaction between treatment 2, assignment 1 and assignment 2.

The default is `c(0.1, 0.5, 0.5, 1, rep(0.1, 9))`.

`eta.sq`	A numeric value specifying the expected partial eta-squared of the linear model with respect to the treatment itself. It is used to control the variance of noise in the linear model. The default is 0.50.
`alpha.list`	A numeric vector containing significance levels (between 0 and 1) used to calculate the empirical alpha. The default is `c(0.001, 0.01, anad 0.05)`.

Value

`mrd_power` returns an object of class `"mrdp"` containing the number of successful iterations, mean, variance, and power (with `alpha` of 0.001, 0.01, and 0.05) for six estimators. The function `summary` is used to obtain and print a summary of the power analysis. The six estimators are as follows:

- The 1st estimator, `Linear`, provides results of the linear regression estimator of combined RD using the centering approach.
- The 2nd estimator, `Opt`, provides results of the local linear regression estimator of combined RD using the centering approach, with the optimal bandwidth in the Imbens and Kalyanaraman (2012) paper.
- The 3rd estimator, `Linear`, provides results of the linear regression estimator of separate RD in terms of `x1` using the univariate approach.
- The 4th estimator, `Opt`, provides results of the local linear regression estimator of separate RD in terms of `x1` using the univariate approach, with the optimal bandwidth in the Imbens and Kalyanaraman (2012) paper.
- The 5th estimator, `Linear`, provides results of the linear regression estimator of separate RD in terms of `x2` using the univariate approach.
- The 6th estimator, `Opt`, provides results of the local linear regression estimator of separate RD in terms of `x2` using the univariate approach, with the optimal bandwidth in the Imbens and Kalyanaraman (2012) paper.

References

Imbens, G., Kalyanaraman, K. (2012). Optimal bandwidth choice for the regression discontinuity estimator. The Review of Economic Studies, 79(3), 933-959. https://academic.oup.com/restud/article/79/3/933/1533189.

Examples

```
## Not run:
summary(mrd_power(x1.design = "l", x2.design = "l"))
summary(mrd_power(x1.dist = "uniform", x1.cut = 0.5,
                  x1.design = "l", x2.design = "l"))
summary(mrd_power(x1.fuzzy = c(0.1, 0.1), x1.design = "l", x2.design = "l"))

## End(Not run)
```

`mrd_sens_bw` *Bandwidth Sensitivity Simulation for Multivariate Regression Discontinuity*

Description

`mrd_sens_bw` refits the supplied model with varying bandwidths. All other aspects of the model are held constant.

Usage

```
mrd_sens_bw(object, approach = c("center", "univ1", "univ2"), bws)
```

Arguments

`object`	An object returned by `mrd_est` or `mrd_impute`.
`approach`	A string of the approaches to be refitted, choosing from `c("center", "univ1", "univ2")`.
`bws`	A positive numeric vector of the bandwidths for refitting an `mrd` object.

Value

`mrd_sens_bw` returns a dataframe containing the estimate `est` and standard error `se` for each supplied bandwidth and for the Imbens-Kalyanaraman (2012) optimal bandwidth, `bw`, and for each supplied approach, `model`. Approaches are either user specified (`"usr"`) or based on the optimal bandwidth (`"origin"`).

References

Imbens, G., Kalyanaraman, K. (2012). Optimal bandwidth choice for the regression discontinuity estimator. The Review of Economic Studies, 79(3), 933-959. https://academic.oup.com/restud/article/79/3/933/1533189.

Examples

```
set.seed(12345)
x1 <- runif(10000, -1, 1)
x2 <- rnorm(10000, 10, 2)
cov <- rnorm(10000)
y <- 3 + 2 * x1 + 1 * x2 + 3 * cov + 10 * (x1 >= 0) + 5 * (x2 >= 10) + rnorm(10000)
# front.bw arugment was supplied to speed up the example
# users should choose appropriate values for front.bw
mrd <- mrd_est(y ~ x1 + x2 | cov,
               cutpoint = c(0, 10), t.design = c("geq", "geq"), front.bw=c(1,1,1))
mrd_sens_bw(mrd, approach = "univ1", bws = seq(0.1, 1, length.out = 3))
```

`mrd_sens_cutoff` *Cutoff Sensitivity Simulation for Multivariate Regression Discontinuity*

Description

`mrd_sens_cutoff` refits the supplied model with varying cutoff(s). All other aspects of the model, such as the automatically calculated bandwidth, are held constant.

Usage

```
mrd_sens_cutoff(object, cutoffs)
```

Arguments

`object`	An object returned by `mrd_est` or `mrd_impute`.
`cutoffs`	A two-column numeric matrix of paired cutoff values to be used for refitting an `mrd` object. The first column corresponds to cutoffs for `x1` and the second column corresponds to cutoffs for `x2`.

Value

`mrd_sens_cutoff` returns a dataframe containing the estimate `est` and standard error `se` for each pair of cutoffs (`A1` and `A2`) and for each `model`. `A1` contains varying cutoffs for assignment 1 and `A2` contains varying cutoffs for assignment 2. The `model` column contains the approach (either centering, univariate 1, or univariate 2) for determining the cutoff and the parametric model (linear, quadratic, or cubic) or non-parametric bandwidth setting (Imbens-Kalyanaraman 2012 optimal, half, or double) used for estimation.

References

Imbens, G., Kalyanaraman, K. (2012). Optimal bandwidth choice for the regression discontinuity estimator. The Review of Economic Studies, 79(3), 933-959. `https://academic.oup.com/restud/article/79/3/933/1533189`.

Examples

```
set.seed(12345)
x1 <- runif(5000, -1, 1)
x2 <- rnorm(5000, 10, 2)
cov <- rnorm(5000)
y <- 3 + 2 * x1 + 1 * x2 + 3 * cov + 10 * (x1 >= 0) + 5 * (x2 >= 10) + rnorm(5000)
# front.bw arugment was supplied to speed up the example
# users should choose appropriate values for front.bw
mrd <- mrd_est(y ~ x1 + x2 | cov,
               cutpoint = c(0, 10), t.design = c("geq", "geq"), front.bw = c(1,1,1))
mrd_sens_cutoff(mrd, expand.grid(A1 = seq(-.5, .5, length.out = 3), A2 = 10))
```

`plot.mfrd` *Plot the Multivariate Frontier Regression Discontinuity*

Description

`plot.mfrd` plots a 3D illustration of the bivariate frontier regression discontinuity design (RDD).

Usage

```
## S3 method for class 'mfrd'
plot(
  x,
  model = c("m_s", "m_h", "m_t"),
  methodname = c("Param", "bw", "Half-bw", "Double-bw"),
  gran = 10,
  raw_data = TRUE,
  color_surface = FALSE,
  ...
)
```

Arguments

`x`	An `mfrd` object returned by `mfrd_est` or contained in the object returned by `mrd_est`.
`model`	A string containing the model specification. Options include one of `c("m_s", "m_h", "m_t")`, which denote the complete model, heterogeneous treatment model, and treatment only model, respectively.
`methodname`	A string containing the method specification. Options include one of `c("Param", "bw", "Half-bw", "Double-bw")`.
`gran`	A non-negative integer specifying the granularity of the surface grid (i.e. the desired number of predicted points before and after the cutoff, along each assignment variable). The default is 10.
`raw_data`	A logical value indicating whether the raw data points are plotted. The default is `TRUE`.

`color_surface`	A logical value indicating whether the treated surface is colored. The default is `FALSE`.
`...`	Additional graphic arguments passed to `persp`.

Examples

```
set.seed(12345)
x1 <- runif(1000, -1, 1)
x2 <- runif(1000, -1, 1)
cov <- rnorm(1000)
y <- 3 + 2 * (x1 >= 0) + 3 * cov + 10 * (x2 >= 0) + rnorm(1000)
model <- mrd_est(y ~ x1 + x2, cutpoint = c(0, 0), t.design = c("geq", "geq"))
plot(model$front$tau_MRD, "m_s", "Param")
```

`plot.rd` *Plot the Regression Discontinuity*

Description

`plot.rd` plots the relationship between the running variable and the outcome. It is based on the `plot.RD` function in the "rdd" package.

Usage

```
## S3 method for class 'rd'
plot(
  x,
  preds = NULL,
  fit_line = c("linear", "quadratic", "cubic", "optimal", "half", "double"),
  fit_ci = c("area", "dot", "hide"),
  fit_ci_level = 0.95,
  bin_n = 20,
  bin_level = 0.95,
  bin_size = c("shade", "size"),
  quant_bin = TRUE,
  xlim = NULL,
  ylim = NULL,
  include_rugs = FALSE,
  ...
)
```

Arguments

`x`	An `rd` object, typically the result of `rd_est`.
`preds`	An optional vector of predictions generated by `predict.rd`. If not supplied, prediction is completed within the `plot.rd` function.

`fit_line`	A character vector specifying models to be shown as fitted lines. Options are `c("linear", "quadratic", "cubic", "optimal", "half", "double")`.
`fit_ci`	A string specifying whether and how to plot prediction confidence intervals around the fitted lines. Options are `c("area", "dot", "hide")`.
`fit_ci_level`	A numeric value between 0 and 1 specifying the confidence level of prediction CIs. The default is 0.95.
`bin_n`	An integer specifying the number of bins for binned data points. If `bin_n` is 0, raw data points are plotted. If `bin_n` is < 0, data points are suppressed. The default is 20.
`bin_level`	A numeric value between 0 and 1 specifying the confidence level for CIs around binned data points. The default is 0.95.
`bin_size`	A string specifying how to plot the number of observations in each bin, by `"size"` or `"shape"`.
`quant_bin`	A logical value indicating whether the data are binned by quantiles. The default is `TRUE`.
`xlim`	An optional numeric vector containing the x-axis limits.
`ylim`	An optional numeric vector containing the y-axis limits.
`include_rugs`	A logical value indicating whether to include the 1d plot for both axes. The default is `FALSE`.
`...`	Additional graphic arguments passed to `plot`.

References

Drew Dimmery (2016). rdd: Regression Discontinuity Estimation. R package version 0.57. https://CRAN.R-project.org/package=rdd

Examples

```
set.seed(12345)
dat <- data.frame(x = runif(1000, -1, 1), cov = rnorm(1000))
dat$tr <- as.integer(dat$x >= 0)
dat$y <- 3 + 2 * dat$x + 3 * dat$cov + 10 * (dat$x >= 0) + rnorm(1000)
rd <- rd_est(y ~ x + tr | cov, data = dat, cutpoint = 0, t.design = "geq")
plot(rd)
```

`predict.rd` *Predict the Regression Discontinuity*

Description

`predict.rd` makes predictions of means and standard deviations of RDs at different cutoffs.

Usage

```
## S3 method for class 'rd'
predict(object, gran = 50, ...)
```

Arguments

`object`	An `rd` object, typically the result of `rd_est`.
`gran`	A non-negative integer specifying the granularity of the data points (i.e. the desired number of predicted points). The default is 50.
`...`	Additional arguments passed to `predict`.

Examples

```
set.seed(12345)
x <- runif(1000, -1, 1)
cov <- rnorm(1000)
y <- 3 + 2 * x + 3 * cov + 10 * (x >= 0) + rnorm(1000)
tr <- as.integer(x >= 0)
rd <- rd_est(y ~ x + tr | cov, cutpoint = 0, t.design = "geq")
predict(rd)
```

`print.mfrd` *Print the Multivariate Frontier Regression Discontinuity*

Description

`print.mfrd` prints a very basic summary of the multivariate frontier regression discontinuity. It is based on the `print.RD` function in the "rdd" package.

Usage

```
## S3 method for class 'mfrd'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`x`	An `mfrd` object, typically the result of `mfrd_est`.
`digits`	A non-negative integer specifying the number of digits to print. The default is `max(3, getOption("digits") - 3)`.
`...`	Additional arguments passed to `print`.

References

Drew Dimmery (2016). rdd: Regression Discontinuity Estimation. R package version 0.57. https://CRAN.R-project.org/package=rdd

---

`print.rd` *Print the Regression Discontinuity*

Description

`print.rd` prints a basic summary of the regression discontinuity. `print.rd` is based on the `print.RD` function in the "rdd" package.

Usage

```
## S3 method for class 'rd'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`x`	An `rd` object, typically the result of `rd_est`.
`digits`	A non-negative integer specifying the number of digits to print. The default is `max(3, getOption("digits") - 3)`.
`...`	Additional arguments passed to `print`.

References

Drew Dimmery (2016). rdd: Regression Discontinuity Estimation. R package version 0.57. https://CRAN.R-project.org/package=rdd

---

`rd_est` *Regression Discontinuity Estimation*

Description

`rd_est` estimates both sharp and fuzzy RDDs using parametric and non-parametric (local linear) models. It is based on the `RDestimate` function in the "rdd" package. Sharp RDDs (both parametric and non-parametric) are estimated using `lm` in the **stats** package. Fuzzy RDDs (both parametric and non-parametric) are estimated using two-stage least-squares `ivreg` in the **AER** package. For non-parametric models, Imbens-Kalyanaraman optimal bandwidths can be used,

Usage

```
rd_est(
  formula,
  data,
  subset = NULL,
  cutpoint = NULL,
  bw = NULL,
  kernel = "triangular",
```

```
  se.type = "HC1",
  cluster = NULL,
  verbose = FALSE,
  less = FALSE,
  est.cov = FALSE,
  est.itt = FALSE,
  t.design = NULL
)
```

Arguments

`formula`	The formula of the RDD; a symbolic description of the model to be fitted. This is supplied in the format of `y ~ x` for a simple sharp RDD or `y ~ x | c1 + c2` for a sharp RDD with two covariates. A fuzzy RDD may be specified as `y ~ x + z` where `x` is the running variable, and `z` is the endogenous treatment variable. Covariates are included in the same manner as in a sharp RDD.
`data`	An optional data frame containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`.
`subset`	An optional vector specifying a subset of observations to be used in the fitting process.
`cutpoint`	A numeric value containing the cutpoint at which assignment to the treatment is determined. The default is 0.
`bw`	A vector specifying the bandwidths at which to estimate the RD. Possible values are `"IK09"`, `"IK12"`, and a user-specified non-negative numeric vector specifying the bandwidths at which to estimate the RD. The default is `"IK12"`. If `bw` is `"IK12"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2012 method. If `bw` is `"IK09"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2009 method. Then the RD is estimated with that bandwidth, half that bandwidth, and twice that bandwidth. If only a single value is passed into the function, the RD will similarly be estimated at that bandwidth, half that bandwidth, and twice that bandwidth.
`kernel`	A string indicating which kernel to use. Options are `"triangular"` (default and recommended), `"rectangular"`, `"epanechnikov"`, `"quartic"`, `"triweight"`, `"tricube"`, and `"cosine"`.
`se.type`	This specifies the robust standard error calculation method to use, from the "sandwich" package. Options are, as in `vcovHC`, `"HC3"`, `"const"`, `"HC"`, `"HC0"`, `"HC1"`, `"HC2"`, `"HC4"`, `"HC4m"`, `"HC5"`. The default is `"HC1"`. This option is overridden by `cluster`.
`cluster`	An optional vector of length n specifying clusters within which the errors are assumed to be correlated. This will result in reporting cluster robust SEs. This option overrides anything specified in `se.type`. It is suggested that data with a discrete running variable be clustered by each unique value of the running variable (Lee and Card, 2008).
`verbose`	A logical value indicating whether to print additional information to the terminal. The default is `FALSE`.

`less`	Logical. If `TRUE`, return the estimates of linear and optimal. If `FALSE` return the estimates of linear, quadratic, cubic, optimal, half and double. The default is `FALSE`.
`est.cov`	Logical. If `TRUE`, the estimates of covariates will be included. If `FALSE`, the estimates of covariates will not be included. The default is `FALSE`. This option is not applicable if method is `"front"`.
`est.itt`	Logical. If `TRUE`, the estimates of ITT will be returned. The default is `FALSE`.
`t.design`	A string specifying the treatment option according to design. Options are `"g"` (treatment is assigned if `x` is greater than its cutoff), `"geq"` (treatment is assigned if `x` is greater than or equal to its cutoff), `"l"` (treatment is assigned if `x` is less than its cutoff), and `"leq"` (treatment is assigned if `x` is less than or equal to its cutoff).

Value

`rd_est` returns an object of class `"rd"`. The functions `summary` and `plot` are used to obtain and print a summary and plot of the estimated regression discontinuity. The object of class `rd` is a list containing the following components:

`type`	A string denoting either `"sharp"` or `"fuzzy"` RDD.
`est`	Numeric vector of the estimate of the discontinuity in the outcome under a sharp RDD or the Wald estimator in the fuzzy RDD, for each corresponding bandwidth.
`se`	Numeric vector of the standard error for each corresponding bandwidth.
`z`	Numeric vector of the z statistic for each corresponding bandwidth.
`p`	Numeric vector of the p-value for each corresponding bandwidth.
`ci`	The matrix of the 95 for each corresponding bandwidth.
`d`	Numeric vector of the effect size (Cohen's d) for each estimate.
`cov`	The names of covariates.
`bw`	Numeric vector of each bandwidth used in estimation.
`obs`	Vector of the number of observations within the corresponding bandwidth.
`call`	The matched call.
`na.action`	The number of observations removed from fitting due to missingness.
`impute`	A logical value indicating whether multiple imputation is used or not.
`model`	For a sharp design, a list of the `lm` objects is returned. For a fuzzy design, a list of lists is returned, each with two elements: `firststage`, the first stage `lm` object, and `iv`, the `ivreg` object. A model is returned for each corresponding bandwidth.
`frame`	Returns the dataframe used in fitting the model.

References

Lee, D. S., Lemieux, T. (2010). Regression Discontinuity Designs in Economics. Journal of Economic Literature, 48(2), 281-355. doi:10.1257/jel.48.2.281.

Imbens, G., Lemieux, T. (2008). Regression discontinuity designs: A guide to practice. Journal of Econometrics, 142(2), 615-635. doi:10.1016/j.jeconom.2007.05.001.

Lee, D. S., Card, D. (2010). Regression discontinuity inference with specification error. Journal of Econometrics, 142(2), 655-674. doi:10.1016/j.jeconom.2007.05.003.

Angrist, J. D., Pischke, J.-S. (2009). Mostly harmless econometrics: An empiricist's companion. Princeton, NJ: Princeton University Press.

Drew Dimmery (2016). rdd: Regression Discontinuity Estimation. R package version 0.57. https://CRAN.R-project.org/package=rdd

Imbens, G., Kalyanaraman, K. (2009). Optimal bandwidth choice for the regression discontinuity estimator (Working Paper No. 14726). National Bureau of Economic Research. `https://www.nber.org/papers/w14726`.

Imbens, G., Kalyanaraman, K. (2012). Optimal bandwidth choice for the regression discontinuity estimator. The Review of Economic Studies, 79(3), 933-959. `https://academic.oup.com/restud/article/79/3/933/1533189`.

Examples

```
set.seed(12345)
x <- runif(1000, -1, 1)
cov <- rnorm(1000)
y <- 3 + 2 * x + 3 * cov + 10 * (x >= 0) + rnorm(1000)
rd_est(y ~ x, t.design = "geq")
# Efficiency gains can be made by including covariates (review SEs in "summary" output).
rd_est(y ~ x | cov, t.design = "geq")
```

`rd_impute` *Multiple Imputation of Regression Discontinuity Estimation*

Description

`rd_impute` estimates treatment effects in an RDD with imputed missing values.

Usage

```
rd_impute(
  formula,
  data,
  subset = NULL,
  cutpoint = NULL,
  bw = NULL,
  kernel = "triangular",
  se.type = "HC1",
```

```
  cluster = NULL,
  impute = NULL,
  verbose = FALSE,
  less = FALSE,
  est.cov = FALSE,
  est.itt = FALSE,
  t.design = NULL
)
```

Arguments

`formula`	The formula of the RDD; a symbolic description of the model to be fitted. This is supplied in the format of `y ~ x` for a simple sharp RDD or `y ~ x | c1 + c2` for a sharp RDD with two covariates. A fuzzy RDD may be specified as `y ~ x + z` where `x` is the running variable, and `z` is the endogenous treatment variable. Covariates are included in the same manner as in a sharp RDD.
`data`	An optional data frame containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`.
`subset`	An optional vector specifying a subset of observations to be used in the fitting process.
`cutpoint`	A numeric value containing the cutpoint at which assignment to the treatment is determined. The default is 0.
`bw`	A vector specifying the bandwidths at which to estimate the RD. Possible values are `"IK09"`, `"IK12"`, and a user-specified non-negative numeric vector specifying the bandwidths at which to estimate the RD. The default is `"IK12"`. If `bw` is `"IK12"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2012 method. If `bw` is `"IK09"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2009 method. Then the RD is estimated with that bandwidth, half that bandwidth, and twice that bandwidth. If only a single value is passed into the function, the RD will similarly be estimated at that bandwidth, half that bandwidth, and twice that bandwidth.
`kernel`	A string indicating which kernel to use. Options are `"triangular"` (default and recommended), `"rectangular"`, `"epanechnikov"`, `"quartic"`, `"triweight"`, `"tricube"`, and `"cosine"`.
`se.type`	This specifies the robust standard error calculation method to use, from the "sandwich" package. Options are, as in `vcovHC`, `"HC3"`, `"const"`, `"HC"`, `"HC0"`, `"HC1"`, `"HC2"`, `"HC4"`, `"HC4m"`, `"HC5"`. This option is overridden by `cluster`.
`cluster`	An optional vector specifying clusters within which the errors are assumed to be correlated. This will result in reporting cluster robust SEs. This option overrides anything specified in `se.type`. It is suggested that data with a discrete running variable be clustered by each unique value of the running variable (Lee and Card, 2008).
`impute`	An optional vector of length n, indexing whole imputations.
`verbose`	A logical value indicating whether to print additional information to the terminal. The default is `FALSE`.

`less`	Logical. If `TRUE`, return the estimates of linear and optimal. If `FALSE` return the estimates of linear, quadratic, cubic, optimal, half and double. The default is `FALSE`.
`est.cov`	Logical. If `TRUE`, the estimates of covariates will be included. If `FALSE`, the estimates of covariates will not be included. The default is `FALSE`. This option is not applicable if method is `"front"`.
`est.itt`	Logical. If `TRUE`, the estimates of ITT will be returned. If `FALSE`, the estimates of ITT will not be returned. The default is `FALSE`. This option is not applicable if method is `"front"`.
`t.design`	A string specifying the treatment option according to design. Options are `"g"` (treatment is assigned if `x` is greater than its cutoff), `"geq"` (treatment is assigned if `x` is greater than or equal to its cutoff), `"l"` (treatment is assigned if `x` is less than its cutoff), and `"leq"` (treatment is assigned if `x` is less than or equal to its cutoff).

Value

`rd_impute` returns an object of class `"rd"`. The functions `summary` and `plot` are used to obtain and print a summary and plot of the estimated regression discontinuity. The object of class `rd` is a list containing the following components:

`call`	The matched call.
`impute`	A logical value indicating whether multiple imputation is used or not.
`type`	A string denoting either `"sharp"` or `"fuzzy"` RDD.
`cov`	The names of covariates.
`bw`	Numeric vector of each bandwidth used in estimation.
`obs`	Vector of the number of observations within the corresponding bandwidth.
`model`	For a sharp design, a list of the `lm` objects is returned. For a fuzzy design, a list of lists is returned, each with two elements: `firststage`, the first stage `lm` object, and `iv`, the `ivreg` object. A model is returned for each parametric and non-parametric case and corresponding bandwidth.
`frame`	Returns the model frame used in fitting.
`na.action`	The observations removed from fitting due to missingness.
`est`	Numeric vector of the estimate of the discontinuity in the outcome under a sharp RDD or the Wald estimator in the fuzzy RDD, for each corresponding bandwidth.
`d`	Numeric vector of the effect size (Cohen's d) for each estimate.
`se`	Numeric vector of the standard error for each corresponding bandwidth.
`z`	Numeric vector of the z statistic for each corresponding bandwidth.
`df`	Numeric vector of the degrees of freedom computed using Barnard and Rubin (1999) adjustment for imputation.
`p`	Numeric vector of the p-value for each corresponding bandwidth.
`ci`	The matrix of the 95 for each corresponding bandwidth.

References

Lee, D. S., Card, D. (2010). Regression discontinuity inference with specification error. Journal of Econometrics, 142(2), 655-674. doi:10.1016/j.jeconom.2007.05.003.

Imbens, G., Kalyanaraman, K. (2009). Optimal bandwidth choice for the regression discontinuity estimator (Working Paper No. 14726). National Bureau of Economic Research. `https://www.nber.org/papers/w14726`.

Imbens, G., Kalyanaraman, K. (2012). Optimal bandwidth choice for the regression discontinuity estimator. The Review of Economic Studies, 79(3), 933-959. `https://academic.oup.com/restud/article/79/3/933/1533189`.

Barnard, J., Rubin, D. (1999). Small-Sample Degrees of Freedom with Multiple Imputation. Biometrika, 86(4), 948-55.

Examples

```
set.seed(12345)
x <- runif(1000, -1, 1)
cov <- rnorm(1000)
y <- 3 + 2 * x + 3 * cov + 10 * (x < 0) + rnorm(1000)
group <- rep(1:10, each = 100)
rd_impute(y ~ x, impute = group, t.design = "l")
# Efficiency gains can be made by including covariates (review SEs in "summary" output).
rd_impute(y ~ x | cov, impute = group, t.design = "l")
```

`rd_power` *Power Analysis of Regression Discontinuity*

Description

`rd_power` computes the empirical probability that a resulting parameter estimate of the MRD is significant, i.e. the empirical power (1 - beta).

Usage

```
rd_power(
  num.rep = 100,
  sample.size = 100,
  x.dist = "normal",
  x.para = c(0, 1),
  x.cut = 0,
  x.fuzzy = c(0, 0),
  x.design = NULL,
  coeff = c(0.3, 1, 0.2, 0.3),
  eta.sq = 0.5,
  alpha.list = c(0.001, 0.01, 0.05)
)
```

Arguments

`num.rep`	A non-negative integer specifying the number of repetitions used to calculate the empirical power. The default is 100.
`sample.size`	A non-negative integer specifying the number of observations in each sample. The default is 100.
`x.dist`	A string specifying the distribution of the assignment variable, `x`. Options are `"normal"` and `"uniform"`. The default is the `"normal"` distribution.
`x.para`	A numeric vector of length 2 specifying parameters of the distribution of the first assignment variable, `x1`. If `x.dist` is `"normal"`, then `x.para` includes the mean and standard deviation of the normal distribution. If `x.dist` is `"uniform"`, then `x.para` includes the upper and lower boundaries of the uniform distribution. The default is `c(0,1)`.
`x.cut`	A numeric value containing the cutpoint at which assignment to the treatment is determined. The default is 0.
`x.fuzzy`	A numeric vector of length 2 specifying the probabilities to be assigned to the control, in terms of the assignment variable, `x`, for individuals in the treatment based on the cutoff, and to treatment for individuals in the control based on the cutoff. For a sharp design, both entries are 0. For a fuzzy design, the first entry is the probability to be assigned to control for individuals above the cutpoint, and the second entry is the probability to be assigned to treatment for individuals below the cutpoint. The default is c(0,0), indicating a sharp design.
`x.design`	A string specifying the treatment option according to design. Options are `"g"` (treatment is assigned if `x` is greater than its cutoff), `"geq"` (treatment is assigned if `x` is greater than or equal to its cutoff), `"l"` (treatment is assigned if `x` is less than its cutoff), and `"leq"` (treatment is assigned if `x` is less than or equal to its cutoff).
`coeff`	A numeric vector specifying coefficients of variables in the linear model to generate data. Coefficients are in the following order: • The 1st entry is the intercept. • The 2nd entry is the slope of treatment, i.e. treatment effect. • The 3rd entry is the slope of assignment. • The 4th entry is the slope of interaction between treatment and assignment. The default is `c(0.3, 1, 0.2, 0.3)`.
`eta.sq`	A numeric value specifying the expected partial eta-squared of the linear model with respect to the treatment itself. It is used to control the variance of noise in the linear model. The default is 0.50.
`alpha.list`	A numeric vector containing significance levels (between 0 and 1) used to calculate the empirical alpha. The default is `c(0.001, 0.01, 0.05)`.

Value

`rd_power` returns an object of class `"rdp"`, including containing the mean, variance, and power (with `alpha` of 0.001, 0.01, and 0.05) for two estimators. The function `summary` is used to obtain and print a summary of the power analysis. The two estimators are:

- The 1st estimator, `Linear`, provides results of the linear regression estimator.
- The 2nd estimator, `Opt`, provides results of the local linear regression estimator of RD, with the optimal bandwidth in the Imbens and Kalyanaraman (2012) paper.

References

Imbens, G., Kalyanaraman, K. (2012). Optimal bandwidth choice for the regression discontinuity estimator. The Review of Economic Studies, 79(3), 933-959. `https://academic.oup.com/restud/article/79/3/933/1533189`.

Examples

```
## Not run:
summary(rd_power(x.design = "l"))
summary(rd_power(x.dist = "uniform", x.cut = 0.5, x.design = "l"))
summary(rd_power(x.fuzzy = c(0.1, 0.1), x.design = "l"))

## End(Not run)
```

`rd_sens_bw` *Bandwidth Sensitivity Simulation for Regression Discontinuity*

Description

`rd_sens_bw` refits the supplied model with varying bandwidths. All other aspects of the model are held constant.

Usage

```
rd_sens_bw(object, bws)
```

Arguments

`object`	An object returned by `rd_est` or `rd_impute`.
`bws`	A positive numeric vector of the bandwidths for refitting an `rd` object.

Value

`rd_sens_bw` returns a dataframe containing the estimate `est` and standard error `se` for each supplied bandwidth and for the Imbens-Kalyanaraman (2012) optimal bandwidth, `bw`, and for each supplied approach, `model`. Approaches are either user specified (`"usr"`) or based on the optimal bandwidth (`"origin"`).

References

Imbens, G., Kalyanaraman, K. (2012). Optimal bandwidth choice for the regression discontinuity estimator. The Review of Economic Studies, 79(3), 933-959. `https://academic.oup.com/restud/article/79/3/933/1533189`.

Examples

```
set.seed(12345)
x <- runif(1000, -1, 1)
cov <- rnorm(1000)
y <- 3 + 2 * x + 3 * cov + 10 * (x >= 0) + rnorm(1000)
rd <- rd_est(y ~ x | cov, t.design = "geq")
rd_sens_bw(rd, bws = seq(.1, 1, length.out = 5))
```

`rd_sens_cutoff` *Cutoff Sensitivity Simulation for Regression Discontinuity*

Description

`rd_sens_cutoff` refits the supplied model with varying cutoff(s). All other aspects of the model, such as the automatically calculated bandwidth, are held constant.

Usage

```
rd_sens_cutoff(object, cutoffs)
```

Arguments

`object`	An object returned by `rd_est` or `rd_impute`.
`cutoffs`	A numeric vector of cutoff values to be used for refitting an `rd` object.

Value

`rd_sens_cutoff` returns a dataframe containing the estimate `est` and standard error `se` for each cutoff value (`A1`). Column `A1` contains varying cutoffs on the assignment variable. The `model` column contains the parametric model (linear, quadratic, or cubic) or non-parametric bandwidth setting (Imbens-Kalyanaraman 2012 optimal, half, or double) used for estimation.

References

Imbens, G., Kalyanaraman, K. (2012). Optimal bandwidth choice for the regression discontinuity estimator. The Review of Economic Studies, 79(3), 933-959. https://academic.oup.com/restud/article/79/3/933/1533189.

Examples

```
set.seed(12345)
x <- runif(1000, -1, 1)
cov <- rnorm(1000)
y <- 3 + 2 * x + 3 * cov + 10 * (x >= 0) + rnorm(1000)
rd <- rd_est(y ~ x | cov, t.design = "geq")
rd_sens_cutoff(rd, seq(-.5, .5, length.out = 10))
```

`rd_type` *Determine Type of Regression Discontinuity Design*

Description

`rd_type` cross-tabulates observations based on (1) a binary treatment and (2) one or two assignments and their cutoff values. This is an internal function and is typically not directly invoked by the user. It can be accessed using the triple colon, as in rddapp:::rd_type().

Usage

```
rd_type(
  data,
  treat,
  assign_1,
  cutoff_1,
  operator_1 = NULL,
  assign_2 = NULL,
  cutoff_2 = NULL,
  operator_2 = NULL
)
```

Arguments

`data`	A `data.frame`, with each row representing an observation.
`treat`	A string specifying the name of the numeric treatment variable (treated = positive values).
`assign_1`	A string specifying the variable name of the primary assignment.
`cutoff_1`	A numeric value containing the cutpoint at which assignment to the treatment is determined, for the primary assignment.
`operator_1`	The operator specifying the treatment option according to design for the primary assignment. Options are `"g"` (treatment is assigned if `x1` is greater than its cutoff), `"geq"` (treatment is assigned if `x1` is greater than or equal to its cutoff), `"l"` (treatment is assigned if `x1` is less than its cutoff), and `"leq"` (treatment is assigned if `x1` is less than or equal to its cutoff).
`assign_2`	An optional string specifying the variable name of the secondary assignment.
`cutoff_2`	An optional numeric value containing the cutpoint at which assignment to the treatment is determined, for the secondary assignment.
`operator_2`	The operator specifying the treatment option according to design for the secondary assignment. Options are `"g"` (treatment is assigned if `x2` is greater than its cutoff), `"geq"` (treatment is assigned if `x2` is greater than or equal to its cutoff), `"l"` (treatment is assigned if `x2` is less than its cutoff), and `"leq"` (treatment is assigned if `x2` is less than or equal to its cutoff).

Value

`rd_type` returns a list of two elements:

`crosstab`	The cross-table as a data.frame. Columns in the dataframe include treatment rules, number of observations in the control condition, number of observations in the treatment condition, and the probability of an observation being in treatment or control.
`type`	A string specifying the type of design used, either `"SHARP"` or `"FUZZY"`.

Examples

```
set.seed(12345)
x <- runif(1000, -1, 1)
cov <- rnorm(1000)
y <- 3 + 2 * x + 3 * cov + 10 * (x >= 0) + rnorm(1000)
df <- data.frame(cbind(y, x, t = x>=0))
rddapp:::rd_type(df, 't', 'x', 0, 'geq')
```

`sens_plot` *Plot the Simulated Estimates for Sensitivity Analyses*

Description

`sens_plot` plots the sensitivity analysis for cutpoints or bandwidths.

Usage

```
sens_plot(
  sim_results,
  level = 0.95,
  x = c("A1", "A2", "bw"),
  plot_models = unique(sim_results$model),
  yrange = NULL
)
```

Arguments

`sim_results`	A `data.frame` returned by `rd_sens_cutoff`, `rd_sens_bw`, `mrd_sens_cutoff`, or `mrd_sens_bw`.
`level`	A numeric value between 0 and 1 specifying the confidence level for CIs (assuming a normal sampling distribution). The default is 0.95.
`x`	A string of the column name of the varying parameter in `sim_results`. This will be used as the x-axis in the plot. Possible values are `c("A1", "A2", "bw")`, which are column names in `sim_results`. `A1` specifies that the varying cutoffs are for assignment 1 and `A2` specifies assignment 2. `bw` indicates that the varying parameter is bandwidth.

`plot_models`	A character vector specifying the models to be plotted (i.e. models estimated with different approaches). Possible values are `unique(sim_results$model)`.
`yrange`	An optional numeric vector specifying the range of the y-axis.

Examples

```
set.seed(12345)
x <- runif(1000, -1, 1)
cov <- rnorm(1000)
y <- 3 + 2 * x + 3 * cov + 10 * (x >= 0) + rnorm(1000)
m <- rd_est(y ~ x | cov, t.design = "geq")
sim_cutoff <- rd_sens_cutoff(m, seq(-.5, .5, length.out = 10))
sens_plot(sim_cutoff, x = "A1", plot_models = c("linear", "optimal"))
sim_bw <- rd_sens_bw(m, seq(.1, 1, length.out = 10))
sens_plot(sim_bw, x = "bw")
```

`shiny_run` *Launch the R Shiny App for "rddapp"*

Description

`shiny_run` launches the R Shiny application for "rddapp".

Usage

```
shiny_run(app_name = "shinyrdd")
```

Arguments

`app_name`	A string specifying the name of the R Shiny app. The default is `"shinyrdd"`.

Examples

```
## Not run:
shiny_run()
shiny_run("shinyrdd")

## End(Not run)
```

`summary.mfrd` *Summarize the Multivariate Frontier Regression Discontinuity*

Description

`summary.mfrd` is a `summary` method for class `"mfrd"`. It is based on the `summary.RD` function in the "rdd" package.

Usage

```
## S3 method for class 'mfrd'
summary(object, level = 0.95, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`object`	An object of class `"mfrd"`, usually a result of a call to `mfrd_est`.
`level`	A numeric value between 0 and 1 specifying the confidence level for confidence intervals. The default is 0.95.
`digits`	A non-negative integer specifying the number of digits to display. The default is `max(3, getOption("digits") - 3)`.
`...`	Additional arguments passed to `summary`.

Value

`summary.mfrd` returns a list containing the following components:

`coefficients`	A matrix containing estimates and confidence intervals (if applicable) for the complete model.
`ht_coefficients`	A matrix containing estimates and confidence intervals (if applicable) for the heterogeneous treatment model.
`t_coefficients`	A matrix containing estimates and confidence intervals (if applicable) for the treatment only model.

References

Drew Dimmery (2016). rdd: Regression Discontinuity Estimation. R package version 0.57. https://CRAN.R-project.org/package=rdd

---

`summary.mrd` *Summarize the Multivariate Regression Discontinuity*

Description

`summary.mrd` is a `summary` method for class `"mrd"`. It is based on `summary.RD` function in the "rdd" package.

Usage

```
## S3 method for class 'mrd'
summary(object, level = 0.95, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`object`	An object of class `"mrd"`, usually a result of a call to `mrd_est`.
`level`	A numeric value between 0 and 1 specifying the confidence level for confidence intervals. The default is 0.95.
`digits`	A non-negative integer specifying the number of digits to display. The default is `max(3, getOption("digits") - 3)`.
`...`	Additional arguments passed to `summary`.

Value

`summary.mrd` returns a list which has the following components depending on methods implemented in the `"mrd"` object:

`center_coefficients`	A matrix containing bandwidths, number of observations, estimates, SEs, confidence intervals, z-values and p-values for each estimated bandwidth and/or parametric model.
`univR_coefficients`	A matrix containing bandwidths, number of observations, estimates, SEs, confidence intervals, z-values and p-values for each estimated bandwidth and/or parametric model.
`univM_coefficients`	A matrix containing bandwidths, number of observations, estimates, SEs, confidence intervals, z-values and p-values for each estimated bandwidth and/or parametric model.
`front_coefficients`	A matrix containing estimates and confidence intervals (if applicable) for the complete model.
`front_ht_coefficients`	A matrix containing estimates and confidence intervals (if applicable) for the heterogeneous treatment model.

`front_t_coefficients`	A matrix containing estimates and confidence intervals (if applicable) for the treatment only model.

References

Drew Dimmery (2016). rdd: Regression Discontinuity Estimation. R package version 0.57. https://CRAN.R-project.org/package=rdd

---

`summary.mrdi` *Summarize the Multiple Imputation of Multivariate Regression Discontinuity*

Description

`summary.mrdi` is a `summary` method for class `"mrdi"`. It is based on `summary.RD` function in the "rdd" package.

Usage

```
## S3 method for class 'mrdi'
summary(object, level = 0.95, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`object`	An object of class `"mrdi"`, usually a result of a call to `mrd_impute` with `"front"` method.
`level`	A numeric value between 0 and 1 specifying the confidence level for confidence intervals. The default is 0.95.
`digits`	A non-negative integer specifying the number of digits to display. The default is `max(3, getOption("digits") - 3)`.
`...`	Additional arguments passed to `summary`.

Value

`summary.mrdi` returns a list which has the following components:

`coefficients`	A matrix containing estimates and confidence intervals (if applicable) for the complete model.
`ht_coefficients`	A matrix containing estimates and confidence intervals (if applicable) for the heterogeneous treatment model.
`t_coefficients`	A matrix containing estimates and confidence intervals (if applicable) for the treatment only model.

References

Drew Dimmery (2016). rdd: Regression Discontinuity Estimation. R package version 0.57. https://CRAN.R-project.org/package=rdd

---

`summary.mrdp` *Summarize the Power Analysis of Regression Discontinuity*

Description

`summary.mrdp` is a `summary` method for class `"mrdp"`. It is based on `summary.RD` function in the "rdd" package.

Usage

```
## S3 method for class 'mrdp'
summary(object, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`object`	An object of class `"mrdp"`, usually a result of a call to `mrd_power`.
`digits`	A non-negative integer specifying the number of digits to display. The default is `max(3, getOption("digits") - 3)`.
`...`	Additional arguments passed to `summary`.

Value

`summary.mrdp` returns a list which has the following components:

`coefficients`	A matrix containing the mean, variance, and empirical alpha of each estimator.

References

Drew Dimmery (2016). rdd: Regression Discontinuity Estimation. R package version 0.57. https://CRAN.R-project.org/package=rdd

---

`summary.rd` *Summarize the Regression Discontinuity*

Description

`summary.rd` is a `summary` method for class `"rd"` It is based on `summary.RD` function in the "rdd" package.

Usage

```
## S3 method for class 'rd'
summary(object, level = 0.95, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`object`	An object of class `"rd"`, usually a result of a call to `rd_est`.
`level`	A numeric value between 0 and 1 specifying the confidence level for confidence intervals. The default is 0.95.
`digits`	A non-negative integer specifying the number of digits to display. The default is `max(3, getOption("digits") - 3)`.
`...`	Additional arguments passed to `summary`..

Value

`summary.rd` returns a list which has the following components:

`coefficients`	A matrix containing bandwidths, number of observations, estimates, SEs, confidence intervals, z-values and p-values for each estimated bandwidth.

References

Drew Dimmery (2016). rdd: Regression Discontinuity Estimation. R package version 0.57. https://CRAN.R-project.org/package=rdd

---

`summary.rdp` *Summarize the Power Analysis of Regression Discontinuity*

Description

`summary.rdp` is a `summary` method for class `"rdp"`. It is based on `summary.RD` function in the "rdd" package.

Usage

```
## S3 method for class 'rdp'
summary(object, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`object`	An object of class `"rdp"`, usually a result of a call to `rd_power`.
`digits`	A non-negative integer specifying the number of digits to display. The default is `max(3, getOption("digits") - 3)`.
`...`	Additional arguments passed to `summary`.

Value

`summary.rdp` returns a list which has the following components:

`coefficients`	A matrix containing the mean, variance, and empirical alpha of each estimator.

References

Drew Dimmery (2016). rdd: Regression Discontinuity Estimation. R package version 0.57. https://CRAN.R-project.org/package=rdd

---

`treat_assign` *Treatment Assignment for Regression Discontinuity*

Description

`treat_assign` computes the treatment variable, `t`, based on the cutoff of assignment variable, `x`. This is an internal function and is typically not directly invoked by the user. It can be accessed using the triple colon, as in rddapp:::treat_assign().

Usage

```
treat_assign(x, cut = 0, t.design = "l")
```

Arguments

`x`	A numeric vector containing the assignment variable, `x`.
`cut`	A numeric value containing the cutpoint at which assignment to the treatment is determined. The default is 0.
`t.design`	A string specifying the treatment option according to design. Options are `"g"` (treatment is assigned if `x` is greater than its cutoff), `"geq"` (treatment is assigned if `x` is greater than or equal to its cutoff), `"l"` (treatment is assigned if `x` is less than its cutoff), and `"leq"` (treatment is assigned if `x` is less than or equal to its cutoff). The default is `"l"`.

Value

`treat_assign` returns the treatment variable as a vector according to the design, where 1 means the treated group and 0 means the control group.

`var_center` *Assignment Centering for Multivariate Frontier Regression Discontinuity*

Description

`var_center` computes the univariate assignment variable, `x` based on the cutoffs of two assignment variables: `x1` and `x2`. This is an internal function and is typically not directly invoked by the user. It can be accessed using the triple colon, as in rddapp:::var_center().

Usage

```
var_center(x, cut = c(0, 0), t.design = NULL, t.plot = FALSE)
```

Arguments

`x`	Data frame or matrix of two assignment variables, where the first column is `x1` and the second column is `x2`.
`cut`	A numeric vector of length 2 containing the cutpoints at which assignment to the treatment is determined. The default is `c(0, 0)`.
`t.design`	A character vector of length 2 specifying the treatment option according to design. The first entry is for `x1` and the second entry is for `x2`. Options are `"g"` (treatment is assigned if `x1` is greater than its cutoff), `"geq"` (treatment is assigned if `x1` is greater than or equal to its cutoff), `"l"` (treatment is assigned if `x1` is less than its cutoff), and `"leq"` (treatment is assigned if `x1` is less than or equal to its cutoff). The same options are available for `x2`.
`t.plot`	A logical value indicating whether to calculate the univariate treatment variable, `t`, and make a plot. The default is `FALSE`.

Value

`var_center` returns the univariate assignment variable as a vector according to the design.

`wt_kern` *Kernel Weight Calculation*

Description

`wt_kern` calculates the appropriate kernel weights for a vector. This is useful when, for instance, one wishes to perform local regression. It is based on the `kernelwts` function in the "rdd" package. This is an internal function and is typically not directly invoked by the user. It can be accessed using the triple colon, as in rddapp:::wt_kern().

Usage

```
wt_kern(X, center, bw, kernel = "triangular")
```

Arguments

`X`	A numeric vector containing the the input `X` values. This variable represents the axis along which kernel weighting should be performed.
`center`	A numeric value specifying the point from which distances should be calculated.
`bw`	A numeric value specifying the bandwidth.
`kernel`	A string indicating which kernel to use. Options are `"triangular"` (default and recommended), `"rectangular"`, `"epanechnikov"`, `"quartic"`, `"triweight"`, `"tricube"`, and `"cosine"`.

Value

`wt_kern` returns a vector of weights with length equal to that of the `X` input (one weight per element of `X`).

References

Drew Dimmery (2016). rdd: Regression Discontinuity Estimation. R package version 0.57. https://CRAN.R-project.org/package=rdd

---

`wt_kern_bivariate` *Bivariate Kernel Weight Calculation*

Description

`wt_kern_bivariate` calculates the appropriate weights for two variables for Multivariate Frontier Regression Discontinuity Estimation with nonparametric implementation. Kernel weights are calculated based on the L1 distance of the two variables from the frontiers. This is an internal function and is typically not directly invoked by the user. It can be accessed using the triple colon, as in rddapp:::wt_kern_bivariate().

Usage

```
wt_kern_bivariate(
  X1,
  X2,
  center1,
  center2,
  bw,
  kernel = "triangular",
  t.design = NULL
)
```

Arguments

`X1`	The input x1 values for the first vector. This variable represents the axis along which kernel weighting should be performed; the first assignment variable in an MRDD.
`X2`	The input x2 values for the second vector. `X2` has the same length as `X1`. This variable represents the axis along which kernel weighting should be performed.; the second assignment variable in an MRDD.
`center1`	A numeric value specifying the point from which distances should be calculated for the first vector, `X1`.
`center2`	A numeric value specifying the point from which distances should be calculated for the second vector, `X2`.

`bw`	A numeric vector specifying the bandwidths for each of three effects models (complete model, heterogeneous treatment model, and treatment only model) detailed in Wong, Steiner, and Cook (2013).
`kernel`	A string indicating which kernel to use. Options are `"triangular"` (default and recommended), `"rectangular"`, `"epanechnikov"`, `"quartic"`, `"triweight"`, `"tricube"`, and `"cosine"`.
`t.design`	A character vector of length 2 specifying the treatment option according to design. The first entry is for `x1` and the second entry is for `x2`. Options are `"g"` (treatment is assigned if `x1` is greater than its cutoff), `"geq"` (treatment is assigned if `x1` is greater than or equal to its cutoff), `"l"` (treatment is assigned if `x1` is less than its cutoff), and `"leq"` (treatment is assigned if `x1` is less than or equal to its cutoff). The same options are available for `x2`.

Value

`wt_bivariate_kern` returns a matrix of weights and distances with length equal to that of the `X1` and `X2` input. The first and second weights and distances are calculated with respect to all frontiers of different treatments. The third weight and distance are calculated with respect to the overall frontier of treatment versus non-treatment.

Index