Package 'rddapp'

October 22, 2018

Title Regression Discontinuity Design Application

Version 1.1.0

Date 2018-04-25

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)

Suggests rdd (>= 0.57),
rddtools (>= 0.4.0),
foreign (>= 0.8-67),
devtools (>= 1.12.0),
testthat (>= 1.0.2),
roxygen2 (>= 5.0.1),
knitr (>= 1.14),
rmarkdown (>= 1.1.9012),
DT (>= 0.2)

VignetteBuilder knitr

License GPL (>= 2)

LazyData true

RoxygenNote 6.0.1

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'

R topics documented:

`rddapp-package` *Regression Discontinuity Design Application*

Description

rddapp: A package for regression discontinuity designs

Details

The rddapp package provides a set of functions for the analysis of the regression-discontinuity design (RDD). The three main parts are: estimation of effects of interest, power analysis, and assumption checks.

Estimation

A variety of designs can be estimated in various ways. The single-assignment RDD (both sharp and fuzzy) can be analyzed using both a parametric (global) or non-parametric (local) approach. The multiple-assignment RDD (both sharp and fuzzy) can be analyzed using both parametric and non-parametric estimation. The analysis choices are further to use estimate effects based on univariate scaling, the centering approach, or the frontier approach. The frontier approach can currently only be estimated using parametric regression with bootstrapped standard errors.

Power analysis

Statistical power can be be estimated for both the single- and multiple-assignment RDD, (both sharp and fuzzy), including all parametric and non-parametric estimators mentioned in the estimation section. All power analyses are based on a simulation approach, which means that the user has to provide all necessary parameters for a data-generating model.

Assumption checks

An important part of any RDD are checks of underlying assumptions. The package provides users with the option to estimate McCrary's sorting test (to identify violations of assignment rules), checks of discontinuities of other baseline covariates, along with sensitivity checks of the chosen bandwidth parameter for non-parametric models, and so-called placebo tests, that examine the treatment effect at other cut-points along the assignment variable.

Author(s)

Ze Jin <zj58@cornell.edu>, Wang Liao <wl483@cornell.edu>, Irena Papst <ip98@cornell.edu>, Wenyu Zhang <wz258@cornell.edu>, Felix Thoemmes, <fjt36@cornell.edu>

---

attr_check *Attrition Checks*

Description

attr_check reports missing data on treatment variable, assignment variable, and outcome. Currently it only supports the design with one assignment variable. 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, the assignment variable.
y	A numeric object, the outcome variable, with the same dimensionality as x.
t	A numeric object, the treatment variable, with the same dimensionality as x and y.
x2	A numeric object, the secondary assignment variable.

Value

A list with the missing data numbers and percentages 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 numerical vector which is the running variable.
Y	A numerical vector which is the outcome variable.
cutpoint	The cutpoint.
verbose	Logical flag indicating whether to print more information to the terminal. Default is FALSE.
kernel	String indicating which kernel to use. Options are "triangular" (default and recommended), "rectangular", "epanechnikov", "quartic", "triweight", "tricube", and "cosine".

Value

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. http://www.nber.org/papers/w14726.

---

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 the rdd_bw_ik 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 numerical vector which is the running variable.
Y	A numerical vector which is the outcome variable.
cutpoint	The cutpoint.
verbose	Logical flag indicating whether to print more information to the terminal. Default is FALSE.
kernel	String indicating which kernel to use. Options are "triangular" (default and recommended), "rectangular", "epanechnikov", "quartic", "triweight", "tricube", and "cosine".

Value

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.

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

Description

A subset of children of the randomized controlled 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 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

http://www.researchconnections.org/childcare/studies/4091/version/1

---

dc_test *McCrary Sorting Test*

Description

`dc_test` implements the McCrary (2008) sorting test. 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	Numerical vector of the running variable.
cutpoint	The cutpoint (defaults to 0).
bin	The binwidth (defaults to 2*sd(runvar)*length(runvar)^(-.5)).
bw	The bandwidth to use (by default uses bandwidth selection calculation from McCrary (2008)).
verbose	Logical flag specifying whether to print diagnostic information to the terminal (defaults to TRUE).
plot	Logical flag indicating whether to plot the histogram and density estimations (defaults to TRUE). The user may wrap this function in additional graphical options to modify the plot.
ext.out	Logical flag indicating whether to return extended output. When FALSE (the default) DCdensity will return only the p-value of the test. When TRUE, DCdensity will return the additional information documented below.
htest	Logical flag indicating whether to return an "htest" object compatible with base R's hypothesis test output.
level	Numerical value between 0 and 1. Confidence level for confidence intervals.
digits	Number of digits to display.
timeout	Numerical value specifying the maximum number of seconds (defaults to 30 seconds) expressions in the function are allowed to run. Specify Inf to run all expressions to completion.
...	Additional arguments affecting the plot.

Value

If ext.out is FALSE, only the p value will be returned. Additional output is enabled when ext.out is TRUE. In this case, a list will be returned with the following elements:

theta	The estimated log difference in heights 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. http://dx.doi.org/10.1016/j.jeconom.2007.05.005.

Examples

```
# 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 in Wong, Steiner and Cook (2013). It is based on the MFRDD code in Stata.

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`	The outcome variable (continuous).
`x1`	The assignment variable 1.
`x2`	The assignment variable 2.
`c1`	The cutoff of assignment variable 1.
`c2`	The cutoff of assignment variable 2.
`t.design`	The treatment option according to design. The 1st entry is for x1: `"g"` means treatment is assigned if x1 is greater than its cutoff, `"geq"` means treatment is assigned if x1 is greater than or equal to its cutoff, `"l"` means treatment is assigned if x1 is less than its cutoff, `"leq"` means treatment is assigned if x1 is less than or equal to its cutoff. The 2nd entry is for x2.
`local`	The range of neighboring points around the cutoff on the standardized scale on each assignment variable, which is a positive number.
`front.bw`	A numeric vector specifying the bandwidths at which to estimate the RD for each of three effects models. If NA, front.bw will be determined by cross validation.
`m`	The number of uniformly-at-random samples to draw as search candidates for front.bw if not given.
`k`	An integer specifying the number of folds for cross validation to determine front.bw if not given.
`kernel`	A string specifying the kernel to be used in the local linear fitting. `"triangular"` kernel is the default. Other options are `"rectangular"`, `"epanechnikov"`, `"quartic"`, `"triweight"`, `"tricube"`, `"gaussian"` and `"cosine"`.

ngrid	The number of non-zero grid points on each assignment variable, which is also the number of zero grid points on each assignment variable. Value used in Wong, Steiner and Cook (2013) is 2500, which may cause long computational time.
margin	The range of grid points beyond the minimum and maximum of sample points on each assignment variable.
boot	The number of bootstrap samples to obtain standard error of estimates.
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).
stop.on.error	Logical. If `TRUE` (the default), removes bootstraps which cause error in the `integrate` function, and resample till the specified number of bootstrap samples are acquired.

Value

`mfrd_est` returns an object of class "`mfrd`".

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 an 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. 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. Fuzzy MRDD may be specified as `y ~ x1 + x2 + z` where `x` is the 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.
subset	An optional vector specifying a subset of observations to be used.
cutpoint	The cutpoint. If omitted, it is assumed to be c(0, 0).
bw	A numeric vector specifying the bandwidths at which to estimate the RD. If omitted or it is `"IK12"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2012 method. If it is `"IK09"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2009 method. Then it 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 numeric vector specifying the bandwidths at which to estimate the RD for each of three effects models in the frontier method. If NA, front.bw will be determined by cross validation.
m	The number of uniformly-at-random samples to draw as search candidates for front.bw if not given.
k	An integer specifying the number of folds for cross validation to determine front.bw if not given.
kernel	A string specifying the kernel to be used in the local linear fitting. `"triangular"` kernel is the default and is the "correct" theoretical kernel to be used for edge estimation as in RDD (Lee and Lemieux, 2010). Other options are `"rectangular"`, `"epanechnikov"`, `"quartic"`, `"triweight"`, `"tricube"`, `"gaussian"` and `"cosine"`.
se.type	This specifies the robust SE calculation method to use. 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).
verbose	Will provide some additional information printed to the terminal.
less	Logical. If `TRUE`, return the estimates of linear and optimal, instead of linear, quadratic, cubic, optimal, half and double.
est.cov	Logical. If `TRUE`, the estimates of covariates will be included. Not applicable if method is `"front"`.
est.itt	Logical. If `TRUE`, the estimates of ITT will be returned. Not applicable if method is `"front"`.
local	The range of neighboring points around the cutoff on the standardized scale on each assignment variable, which is a positive number.
ngrid	The number of non-zero grid points on each assignment variable, which is also the number of zero grid points on each assignment variable. Value used in Wong, Steiner and Cook (2013) is 2500, which may cause long computational time.

margin	The range of grid points beyond the minimum and maximum of sample points on each assignment variable.
boot	The number of bootstrap samples to obtain standard error of estimates.
method	The method to estimate rd effect. Options are "center", "univ", "front".
t.design	The treatment option according to design. The 1st entry is for X1: "g" means treatment is assigned if X1 is greater than its cutoff, "geq" means treatment is assigned if X1 is greater than or equal to its cutoff, "l" means treatment is assigned if X1 is less than its cutoff, "leq" means treatment is assigned if X1 is less than or equal to its cutoff. The 2nd entry is for X2.
stop.on.error	Logical. If TRUE (the default), removes bootstraps which cause error in the integrate function, and resample till the specified number of bootstrap samples are acquired.

Value

mrd_est returns an object of class "mrd".

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. http://journals.sagepub.com/doi/10.3102/1076998611432172.

Examples

```
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 an 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. 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. Fuzzy MRDD may be specified as `y ~ x1 + x2 + z` where `x` is the 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.
subset	An optional vector specifying a subset of observations to be used.
cutpoint	The cutpoint. If omitted, it is assumed to be 0.
bw	A numeric vector specifying the bandwidths at which to estimate the RD. If omitted or it is `"IK12"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2012 method. If it is `"IK09"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2009 method. Then it 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 numeric vector specifying the bandwidths at which to estimate the RD for each of three effects models in the frontier method. If NA, front.bw will be determined by cross validation.
m	The number of uniformly-at-random samples to draw as search candidates for front.bw if not given.
k	An integer specifying the number of folds for cross validation to determine front.bw if not given.
kernel	A string specifying the kernel to be used in the local linear fitting. `"triangular"` kernel is the default and is the "correct" theoretical kernel to be used for edge estimation as in RDD (Lee and Lemieux, 2010). Other options are `"rectangular"`, `"epanechnikov"`, `"quartic"`, `"triweight"`, `"tricube"`, `"gaussian"` and `"cosine"`.
se.type	This specifies the robust SE calculation method to use. 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 specifying the imputed variables with missing values.
verbose	Will provide some additional information printed to the terminal.
less	Logical. If `TRUE`, return the estimates of linear and optimal, instead of linear, quadratic, cubic, optimal, half and double.
est.cov	Logical. If `TRUE`, the estimates of covariates will be included.
est.itt	Logical. If `TRUE`, the estimates of ITT will be returned.
local	The range of neighboring points around the cutoff on the standardized scale on each assignment variable, which is a positive number.
ngrid	The number of non-zero grid points on each assignment variable, which is also the number of zero grid points on each assignment variable. Value used in Wong, Steiner and Cook (2013) is 2500, which may cause long computational time.

margin	The range of grid points beyond the minimum and maximum of sample points on each assignment variable.
boot	The number of bootstrap samples to obtain standard deviation of estimates.
method	The method to estimate rd effect. Options are "center", "univ", "front".
t.design	The treatment option according to design. The 1st entry is for X1: "g" means treatment is assigned if X1 is greater than its cutoff, "geq" means treatment is assigned if X1 is greater than or equal to its cutoff, "l" means treatment is assigned if X1 is less than its cutoff, "leq" means treatment is assigned if X1 is less than or equal to its cutoff. The 2nd entry is for X2.
stop.on.error	Logical. If TRUE (the default), removes bootstraps which cause error in the integrate function, and resample till the specified number of bootstrap samples are acquired.

Value

mrd_impute returns an object of class "mrd", or "mrdi" for "front" method.

References

Stata: 64 mi estimate - Estimation using multiple imputations

Examples

```
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)
group <- rep(1:10, each = 100)
# centering
mrd_impute(y ~ x1 + x2 | cov, impute = group, method = "center", t.design = c("geq", "geq"))
# univariate
mrd_impute(y ~ x1 + x2 | cov, impute = group, method = "univ", t.design = c("geq", "geq"))
# frontier
mrd_impute(y ~ x1 + x2 | cov, impute = group, method = "front", t.design = c("geq", "geq"))
```

---

mrd_power *Power Analysis of Multivariate Regression Discontinuity*

Description

mrd_power computes the empirical probability that RD is significant, i.e. the empirical alpha of null hypothesis: RD = 0

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	Number of repetitions used to calculate the empirical alpha.
sample.size	Number of observations in each sample.
x1.dist	Distribution of the 1st assignment variable X1. "normal" distribution is the default. "uniform" distribution is the only other option.
x1.para	Parameters of the distribution of the 1st assignment variable X1. If x1.dist is "normal", then x1.para includes the mean and sd of normal distribution. If x1.dist is "uniform", then x1.para includes the upper and lower boundaries of uniform distribution.
x2.dist	Distribution of the 2nd assignment variable X2.
x2.para	Parameters of the distribution of the 2nd assignment variable X2.
x1.cut	Cutpoint of RD with respect to the 1st assignment variable X1.
x2.cut	Cutpoint of RD with respect to the 2nd assignment variable X2.
x1.fuzzy	Probabilities to be assigned to control in terms of the 1st assignment variable X1 for individuals in treatment based on cutoff, and to treatment for individuals in control based on cutoff. For a sharp design, by default, the 1st entry is 0, and the 2nd entry is 0. For a fuzzy design, the 1st entry is the probability to be assigned to control for individuals above the cutpoint, and the 2nd entry is the probability to be assigned to treatment for individuals below the cutpoint.
x2.fuzzy	Probabilities to be assigned to control in terms of the 2nd assignment variable X2 for individuals in treatment based on cutoff, and to treatment for individuals in control based on cutoff.
x1.design	The treatment option according to design. The entry is for X1: "g" means treatment is assigned if X1 is greater than its cutoff, "geq" means treatment is assigned if X1 is greater than or equal to its cutoff, "l" means treatment is assigned if X1 is less than its cutoff, "leq" means treatment is assigned if X1 is less than or equal to its cutoff.
x2.design	The treatment option according to design. The entry is for X2.
coeff	Coefficients of variables in the linear model to generate data 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.
eta.sq	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.
alpha.list	List of significance levels used to calculate the empirical alpha.

Value

mrd_power returns the results of 6 estimators as a table of class "mrdp", including mean, variance, and power of estimate. The function summary is used to obtain and print a summary of the power analysis. The 1st Linear results of the linear regression estimator of combined RD using the centering approach. The 2nd Opt results of the local linear regression estimator of combined RD using the centering approach, with the optimal bandwidth in the IK 2012 paper. The 3rd Linear results of the linear regression estimator of separate RD in terms of x1 using the univariate approach. The 4th Opt results of the local linear regression estimator of separate RD in terms of x1 using the univariate approach, with the optimal bandwidth in the IK 2012 paper. The 5th Linear results of the linear regression estimator of separate RD in terms of x2 using the univariate approach. The 6th Opt results of the local linear regression estimator of separate RD in terms of x2 using the univariate approach, with the optimal bandwidth in the IK 2012 paper.

Examples

```
## Not run:
mrd_power(x1.design = "l", x2.design = "l")
mrd_power(x1.dist = "uniform", x1.cut = 0.5, x1.design = "l", x2.design = "l")
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 supplemented model with varying bandwidth. Other estimation parameters 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 bandwidth for refitting an mrd object.

Value

A dataframe which contains the estimate est and standard error se for each supplemented bandwidth.

Examples

```
x1 <- runif(1000, -1, 1)
x2 <- rnorm(1000, 10, 2)
cov <- rnorm(1000)
y <- 3 + 2 * x1 + 1 * x2 + 3 * cov + 10 * (x1 >= 0) + 5 * (x2 >= 10) + rnorm(1000)
mrd <- mrd_est(y ~ x1 + x2 | cov, cutpoint = c(0, 10), t.design = c("geq", "geq"))
mrd_sens_bw(mrd, approach = "univ1", bws = seq(0.1, 1, length.out = 5))
```

---

mrd_sens_cutoff *Cutoff Sensitivity Simulation for Multivariate Regression Discontinuity*

Description

`mrd_sens_cutoff` refits the supplemented model with varying cutoff(s). Other estimation parameters, 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.

Value

A dataframe which contains the estimate `est` and standard error `se` for each pairs of cutoffs (`A1` and `A2`). `A1` contains varying cutoffs on assignment 1, and `A2` assignment 2.

Examples

```
x1 <- runif(1000, -1, 1)
x2 <- rnorm(1000, 10, 2)
cov <- rnorm(1000)
y <- 3 + 2 * x1 + 1 * x2 + 3 * cov + 10 * (x1 >= 0) + 5 * (x2 >= 10) + rnorm(1000)
mrd <- mrd_est(y ~ x1 + x2 | cov, cutpoint = c(0, 10), t.design = c("geq", "geq"))
mrd_sens_cutoff(mrd, expand.grid(A1 = seq(-.5, .5, length.out = 5), A2 = 10))
```

plot.mfrd *Plot the Multivariate Frontier Regression Discontinuity*

Description

plot.mfrd plots the 3D illustration of the bivariate frontier RDD.

Usage

```
## S3 method for class 'mfrd'
plot(x, model = c("m_s", "m_h", "m_t"), gran = 2,
  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	Option for the model specification, one of c("m_s", "m_h", "m_t"), which denotes the complete model, heterogeneous treatment model and treatment only model respectively.
gran	Granularity of the surface grid i.e. desired number of predicted points before and after the cutoff along each assignment variable.
raw_data	Whether the raw data points are plotted.
color_surface	Whether the treated surface is colored.
...	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 <- mfrd_est(y = y, x1 = x1, x2 = x2, c1 = 0, c2 = 0, t.design = c("geq", "geq"))
plot(model, "m_s")
```

plot.rd *Plot the Regression Discontinuity* plot.rd *plots the relationship between the running variable and the outcome. It is based on the* plot.RD *function in the* **rdd** *package.*

Description

Plot the Regression Discontinuity

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	Predictions generated by predict.rd.
fit_line	Models to be shown as fitted lines.
fit_ci	Whether and how to plot prediction CIs around the fitted lines.
fit_ci_level	Confidence level of prediction CIs.
bin_n	Number of bins for binned data points (plot raw data points if = 0; suppress data points if < 0).
bin_level	Confidence level for CIs around binned data points.
bin_size	How to plot the number of observations in each bin.
quant_bin	Whether the data are binned per quantiles.
xlim	x-axis limits.
ylim	y-axis limits.
include_rugs	Whether to include 1d plot for both axes.
...	Additional arguments affecting the plots produced.

Examples

```
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 *Prediction the Regression Discontinuity*

Description

predict.rd makes predictions of mean and standard deviation 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	Granularity of the data points i.e. desired number of predicted points.
...	Additional arguments affecting the predictions produced.

Examples

```
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	The number of digits to print.
...	Additional arguments.

---

print.rd *Print the Regression Discontinuity*

Description

print.rd prints a very basic summary of the regression discontinuity. It 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	The number of digits to print.
...	Additional arguments.

rd_est *Regression Discontinuity Estimation*

Description

rd_est estimates both sharp and fuzzy RDD, 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. 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. Fuzzy RDD may be specified as `y ~ x + z` where x is the running variable, and z is the endogenous treatment variable. Covariates are then included in the same manner as in a sharp RDD.
data	An optional data frame.
subset	An optional vector specifying a subset of observations to be used
cutpoint	The cutpoint. If omitted, it is assumed to be 0.
bw	A numeric vector specifying the bandwidths at which to estimate the RD. If omitted or it is "IK12", the bandwidth is calculated using the Imbens-Kalyanaraman 2012 method. If it is "IK09", the bandwidth is calculated using the Imbens-Kalyanaraman 2009 method. Then it 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 specifying the kernel to be used in the local linear fitting. "triangular" kernel is the default and is the "correct" theoretical kernel to be used for edge estimation as in RDD (Lee and Lemieux, 2010). Other options are "rectangular", "epanechnikov", "quartic", "triweight", "tricube", "gaussian" and "cosine".
se.type	This specifies the robust SE calculation method to use. 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).
verbose	Will provide some additional information printed to the terminal.

less	Logical. If TRUE, return the estimates of linear and optimal, instead of linear, quadratic, cubic, optimal, half and double.
est.cov	Logical. If TRUE, the estimates of covariates will be included.
est.itt	Logical. If TRUE, the estimates of ITT will be returned.
t.design	The treatment option according to design. The entry is for X: "g" means treatment is assigned if X is greater than its cutoff, "geq" means treatment is assigned if X is greater than or equal to its cutoff, "l" means treatment is assigned if X is less than its cutoff, "leq" means 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 design, or the Wald estimator in the fuzzy design 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 effective 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 observations removed from fitting due to missingness.
impute	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 model frame used in fitting.

References

Lee, D. S., Lemieux, T. (2010). Regression Discontinuity Designs in Economics. Journal of Economic Literature, 48(2), 281-355. http://www.aeaweb.org/articles.php?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. http://dx.doi.org/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. http://dx.doi.org/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.

Examples

```
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
rd_est(y ~ x | cov, t.design = "geq")
```

`rd_impute` *Multiple Imputation of Regression Discontinuity Estimation*

Description

`rd_impute` estimates treatment effects in a 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. 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. Fuzzy RDD may be specified as `y ~ x + z` where `x` is the running variable, and `z` is the endogenous treatment variable. Covariates are then included in the same manner as in a sharp RDD.
`data`	An optional data frame.
`subset`	An optional vector specifying a subset of observations to be used
`cutpoint`	The cutpoint. If omitted, it is assumed to be 0.
`bw`	A numeric vector specifying the bandwidths at which to estimate the RD. If omitted or it is `"IK12"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2012 method. If it is `"IK09"`, the bandwidth is calculated using the Imbens-Kalyanaraman 2009 method. Then it 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 specifying the kernel to be used in the local linear fitting. `"triangular"` kernel is the default and is the "correct" theoretical kernel to be used for edge estimation as in RDD (Lee and Lemieux, 2010). Other options are `"rectangular"`, `"epanechnikov"`, `"quartic"`, `"triweight"`, `"tricube"`, `"gaussian"` and `"cosine"`.
`se.type`	This specifies the robust SE calculation method to use. 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 specifying the imputed variables with missing values.
verbose	Will provide some additional information printed to the terminal.
less	Logical. If TRUE, return the estimates of linear and optimal, instead of linear, quadratic, cubic, optimal, half and double.
est.cov	Logical. If TRUE, the estimates of covariates will be included.
est.itt	Logical. If TRUE, the estimates of ITT will be returned.
t.design	The treatment option according to design. The entry is for X: "g" means treatment is assigned if X is greater than its cutoff, "geq" means treatment is assigned if X is greater than or equal to its cutoff, "l" means treatment is assigned if X is less than its cutoff, "leq" means treatment is assigned if X is less than or equal to its cutoff.

Value

rd_impute returns an object of class "rd".

References

Stata: 64 mi estimate - Estimation using multiple imputations

Examples

```
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
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 RD is significant, i.e. the empirical alpha of null hypothesis: RD = 0

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	Number of repetitions used to calculate the empirical alpha.
sample.size	Number of observations in each sample.
x.dist	Distribution of the assignment variable X. "normal" distribution is the default. "uniform" distribution is the only other option.
x.para	Parameters of the distribution of the assignment variable X. If x.dist is "normal", then x.para includes the mean and sd of normal distribution. If x.dist is "uniform", then x.para includes the upper and lower boundaries of uniform distribution.
x.cut	Cutpoint of RD with respect to the assignment variable X.
x.fuzzy	Probabilities to be assigned to control for individuals in treatment based on cutoff, and to treatment for individuals in control based on cutoff. For a sharp design, by default, the 1st entry is 0, and the 2nd entry is 0. For a fuzzy design, the 1st entry is the probability to be assigned to control for individuals above the cutpoint, and the 2nd entry is the probability to be assigned to treatment for individuals below the cutpoint.
x.design	The treatment option according to design. The entry is for X: "g" means treatment is assigned if X is greater than its cutoff, "geq" means treatment is assigned if X is greater than or equal to its cutoff, "l" means treatment is assigned if X is less than its cutoff, "leq" means treatment is assigned if X is less than or equal to its cutoff.
coeff	Coefficients of variables in the linear model to generate data 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.
eta.sq	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.
alpha.list	List of significance levels used to calculate the empirical alpha.

Value

rd_power returns the results of 2 estimators as a table of class "rdp", including mean, variance, and power of estimate. The function summary is used to obtain and print a summary of the power analysis. The 1st Linear results of the linear regression estimator The 2nd Opt results of the local linear regression estimator of RD, with the optimal bandwidth in the IK 2012 paper.

Examples

```
## Not run:
rd_power(x.design = "l")
rd_power(x.dist = "uniform", x.cut = 0.5, x.design = "l")
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 supplemented model with varying bandwidth. Other estimation parameters 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 bandwidth for refitting an rd object.

Value

A dataframe which contains the estimate est and standard error se for each supplemented bandwidth.

Examples

```
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 supplemented model with varying cutoff(s). Other estimation parameters, 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 in the refitting of an rd object.

Value

A dataframe contains the estimate est and standard error se for each cutoff values (A1). Column A1 contains varying cutoffs on the assignment variable.

Examples

```
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	The name of a numeric variable (treated = positive values).
assign_1	The variable name of the primary assignment.
cutoff_1	The cutoff value of the primary assignment.
operator_1	The operator for the primary assignment.
assign_2	The variable name of the secondary assignment.
cutoff_2	The cutoff value of the secondary assignment.
operator_2	The operator for the secondary assignment.

Value

A list of two elements:

crosstab	The cross-table as a data.frame.
type	The type of design as a string.

Examples

```
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))
rddapp:::rd_type(df, 'y', 'x', 0, 'geq')
```

sens_plot *Plot the Simulated Estimates for Sensitivity Analyses*

Description

sens_plot plots the sensitivity analysis for cutpoint or bandwidth.

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	The confidence level for CIs (assuming a normal sampling distribution).
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 means the varying cutoffs are for assignment 1, and A2 assignment 2.
plot_models	A character vector specifying the models (i.e., models estimated with different approaches) to be plotted. Possible values are unique(sim_results$model)).
yrange	A numeric vector of the range of y-axis

Examples

```
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 Shiny App for Regression Discontinuity*

Description

shiny_run looks for the Shiny app for rdd and launch it.

Usage

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

Arguments

app_name	The name of Shiny app.

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 `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`	Numerical value between 0 and 1. Confidence level for confidence intervals.
`digits`	Number of digits to display.
`...`	Additional arguments.

Value

`summary.mfrd` 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.

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	Numerical value between 0 and 1. Confidence level for confidence intervals.
digits	Number of digits to display.
...	Additional arguments.

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.

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

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

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.

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

Description

summary.mrdi is a summary method for class "mrdi"

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	Numerical value between 0 and 1. Confidence level for confidence intervals.
digits	Number of digits to display.
...	Additional arguments.

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.

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

Description

summary.mrdp is a summary method for class "mrdp"

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	Number of digits to display.
...	Additional arguments.

Value

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

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

---

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	Numerical value between 0 and 1. Confidence level for confidence intervals.
digits	Number of digits to display.
...	Additional arguments.

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.

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

Description

summary.rdp is a summary method for class "rdp"

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	Number of digits to display.
...	Additional arguments.

Value

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

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

treat_assign *Treatment Assignment for Regression Discontinuity*

Description

treat_assign computes the treatment variable T based on the cutoff of assignment variables 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	The vector of assignment variable X.
cut	The cutoff of assignment variables X.
t.design	The treatment option according to design. The entry is for X: "g" means treatment is assigned if X is greater than its cutoff, "geq" means treatment is assigned if X is greater than or equal to its cutoff, "l" means treatment is assigned if X is less than its cutoff, "leq" means treatment is assigned if X is less than or equal to its cutoff.

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, 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 1st column is X1, the 2nd column is X2
cut	Cutoffs of two assignment variables X1, X2.
t.design	The treatment option according to design. The 1st entry is for x1: "g" means treatment is assigned if x1 is greater than its cutoff, "geq" means treatment is assigned if x1 is greater than or equal to its cutoff, "l" means treatment is assigned if x1 is less than its cutoff, "leq" means treatment is assigned if x1 is less than or equal to its cutoff. The 2nd entry is for x2.
t.plot	Whether calculate the univariate treatment variable T and make a plot

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	The input x values. This variable represents the axis along which kernel weighting should be performed.
center	The point from which distances should be calculated.
bw	The bandwidth.
kernel	A string indicating the kernel to use. Options are "triangular" (the default), "epanechnikov", "quartic", "triweight", "tricube", "gaussian", and "cosine".

Value

A vector of weights with length equal to that of the X input (one weight per element of X).

---

wt_kern_bivariate *Bivariate Kernel Weight Calculation*

Description

wt_kern_bivariate calculates the appropriate weights for two variables for nonparametric implementation of Multivariate Frontier Regression Discontinuity Estimation. Kernel weights are calculated based on the L1 distance of the two variables from the frontiers.

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.
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.
center1	The point from which distances should be calculated for the first vector.
center2	The point from which distances should be calculated for the second vector.
bw	A numeric vector specifying the bandwidths for each of three effects models
kernel	A string indicating the kernel to use. Options are "triangular" (the default), "epanechnikov", "quartic", "triweight", "tricube", "gaussian", and "cosine".
t.design	The treatment option according to design. The 1st entry is for X1: "g" means treatment is assigned if X1 is greater than its cutoff, "geq" means treatment is assigned if X1 is greater than or equal to its cutoff, "l" means treatment is assigned if X1 is less than its cutoff, "leq" means treatment is assigned if X1 is less than or equal to its cutoff. The 2nd entry is for X2.

Value

Three vectors 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 fontiers of different treatments. The third weight and distance are calculated with respect to the overall frontier of treatment versus non-treatment.

Index