Package ‘rd2d’

May 28, 2026

Type Package

Title Estimation and Inference for Boundary Discontinuity Designs

Version 1.0.0

URL `https://rdpackages.github.io/`, `https://github.com/rdpackages/rd2d`

BugReports `https://github.com/rdpackages/rd2d/issues`

Description Provides pointwise and uniform estimation and inference methods for boundary discontinuity (BD) designs,
a causal inference design that generalizes univariate regression discontinuity (RD) designs to settings with bivariate scores.
Implements local polynomial methods for location-based and distance-based analyses, including sharp and fuzzy designs,
data-driven bandwidth selection, pointwise confidence intervals, and uniform confidence bands. Methodology is developed in
Cattaneo, Titiunik, and Yu (2026) <doi:10.48550/arXiv.2505.05670> for location-based methods and
Cattaneo, Titiunik, and Yu (2026) <doi:10.48550/arXiv.2510.26051> for distance-based methods.
For an overview and empirical guidance, see Cattaneo, Titiunik, and Yu (2026) <doi:10.48550/arXiv.2511.06474>.
The companion software article is Cattaneo, Titiunik, and Yu (2025) <doi:10.48550/arXiv.2505.07989>.

Imports MASS, expm, ggplot2

Suggests testthat (>= 3.0.0)

License GPL-3

Encoding UTF-8

Maintainer Matias D. Cattaneo <`matias.d.cattaneo@gmail.com`>

NeedsCompilation no

Author Matias D. Cattaneo [aut, cre],
Rocio Titiunik [aut],
Ruiqi Rae Yu [aut]

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Repository CRAN

Date/Publication 2026-05-28 10:50:02 UTC

Contents

`rd2d-package` *rd2d: Estimation and Inference for Boundary Discontinuity Designs*

Description

rd2d implements pointwise and uniform estimation and inference procedures for boundary discontinuity (BD) designs using local polynomial methods. The package includes location-based and distance-based methods, sharp and fuzzy designs, automatic bandwidth selection, pointwise confidence intervals, and uniform confidence bands. Distance-based methods in this package target level effects at two-dimensional boundary points.

Included functions are: rd2d for location-based estimation and inference, rdbw2d for location-based bandwidth selection, rd2d.distance for distance-based estimation and inference, and rdbw2d.distance for distance-based bandwidth selection.

`print()`, `summary()`, and `plot()` methods are available for all four functions. Estimation objects returned by `rd2d` and `rd2d.distance` also support `coef()`, `vcov()`, and `confint()`.

Related Stata, R, and Python packages useful for inference in RD designs are described at:

https://rdpackages.github.io/

For an introduction to regression discontinuity designs, see Cattaneo and Titiunik (2022, doi:10.1146/annureveconomics051520021409) and references therein.

Author(s)

Matias D. Cattaneo, Princeton University. `<matias.d.cattaneo@gmail.com>`. Rocio Titiunik, Princeton University. `<rocio.titiunik@gmail.com>`. Ruiqi Rae Yu, Princeton University. `<raeyuuuu@gmail.com>`.

References

- Cattaneo, M. D., and Titiunik, R. (2022). Regression Discontinuity Designs. doi:10.1146/annureveconomics051520021409.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026). Estimation and Inference in Boundary Discontinuity Designs: Location-Based Methods.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026). Estimation and Inference in Boundary Discontinuity Designs: Distance-Based Methods.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026). Boundary Discontinuity Designs: Theory and Practice.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2025). rd2d: Causal Inference in Boundary Discontinuity Designs.

See Also

Useful links:

- `https://rdpackages.github.io/`
- `https://github.com/rdpackages/rd2d`
- Report bugs at `https://github.com/rdpackages/rd2d/issues`

`coef.rd2d` *Extract rd2d Point Estimates*

Description

Extract rd2d Point Estimates

Usage

```
## S3 method for class 'rd2d'
coef(object, output = "main", order = c("q", "p"), ...)

## S3 method for class 'rd2d.distance'
coef(object, output = "main", order = c("q", "p"), ...)
```

Arguments

`object`	An object returned by `rd2d` or `rd2d.distance`.
`output`	Result table to extract. Default is `"main"`.
`order`	Polynomial order estimate to extract: `"q"` for the inference order or `"p"` for the estimation order.
`...`	Additional arguments ignored.

Value

A named numeric vector of estimates.

`confint.rd2d` *Confidence Intervals for rd2d Estimates*

Description

Confidence Intervals for rd2d Estimates

Usage

```
## S3 method for class 'rd2d'
confint(
  object,
  parm,
  level = 0.95,
  output = "main",
  order = c("q", "p"),
  side = NULL,
  ...
)

## S3 method for class 'rd2d.distance'
confint(
  object,
  parm,
  level = 0.95,
  output = "main",
  order = c("q", "p"),
  side = NULL,
  ...
)
```

Arguments

`object`	An object returned by `rd2d` or `rd2d.distance`.
`parm`	Ignored; included for compatibility with `confint`.
`level`	Confidence level in $(0, 1)$. Default is `0.95`.
`output`	Result table to use. Default is `"main"`.
`order`	Polynomial order estimate to use: `"q"` or `"p"`.
`side`	Confidence interval side. Defaults to the side stored in the fit.
`...`	Additional arguments ignored.

Value

A two-column matrix with lower and upper confidence limits.

`plot.rd2d` *Plot rd2d Estimates and Bandwidths*

Description

Plot rd2d Estimates and Bandwidths

Usage

```
## S3 method for class 'rd2d'
plot(
  x,
  output = "main",
  order = c("q", "p"),
  ci = TRUE,
  level = NULL,
  draw = TRUE,
  ...
)

## S3 method for class 'rd2d.distance'
plot(
  x,
  output = "main",
  order = c("q", "p"),
  ci = TRUE,
  level = NULL,
  draw = TRUE,
  ...
)

## S3 method for class 'rdbw2d'
plot(x, draw = TRUE, ...)

## S3 method for class 'rdbw2d.distance'
plot(x, draw = TRUE, ...)
```

Arguments

`x`	An object returned by `rd2d`, `rd2d.distance`, `rdbw2d`, or `rdbw2d.distance`.
`output`	Result table to plot for estimation objects. Default is `"main"`.
`order`	Polynomial order estimate to plot for estimation objects: `"q"` or `"p"`.
`ci`	Logical. If `TRUE`, include pointwise confidence intervals for estimation objects.
`level`	Confidence level for plotted intervals. Defaults to the level stored in the fit, or `0.95` if unavailable.
`draw`	Logical. If `TRUE`, print the ggplot object.
`...`	Additional arguments ignored.

Value

Invisibly, a `ggplot` object.

`print.rd2d` *Print Method for 2D Local Polynomial RD Estimation*

Description

Prints the results of a 2D local polynomial regression discontinuity (RD) estimation, as obtained from `rd2d`.

Usage

```
## S3 method for class 'rd2d'
print(x, ...)
```

Arguments

`x`	An object of class `rd2d`, returned by `rd2d`.
`...`	Additional arguments passed to the method (currently ignored).

Value

No return value. This function is called for its side effects, which are to print the `rd2d` results.

Author(s)

Matias D. Cattaneo, Princeton University. `<matias.d.cattaneo@gmail.com>`
Rocio Titiunik, Princeton University. `<rocio.titiunik@gmail.com>`
Ruiqi Rae Yu, Princeton University. `<raeyuuuu@gmail.com>`

See Also

`rd2d` for conducting 2D local polynomial RD estimation.

Supported methods: `print.rd2d`, `summary.rd2d`.

`print.rd2d.distance` *Print Method for 2D Local Polynomial RD Estimation (Distance-Based)*

Description

Prints the results of a 2D local polynomial regression discontinuity (RD) estimation using distance-based evaluation, as obtained from `rd2d.distance`.

Usage

```
## S3 method for class 'rd2d.distance'
print(x, ...)
```

Arguments

`x`	An object of class `rd2d.distance`, returned by `rd2d.distance`.
`...`	Additional arguments passed to the method (currently ignored).

Value

No return value. This function is called for its side effects: it prints the `rd2d.distance` results.

Author(s)

Matias D. Cattaneo, Princeton University. <`matias.d.cattaneo@gmail.com`>
Rocio Titiunik, Princeton University. <`rocio.titiunik@gmail.com`>
Ruiqi Rae Yu, Princeton University. <`raeyuuuu@gmail.com`>

See Also

`rd2d.distance` for estimation using distance-based methods in 2D local polynomial RD designs.

Supported methods: `print.rd2d.distance`, `summary.rd2d.distance`.

`print.rdbw2d` *Print Method for Bandwidth Selection for 2D Local Polynomial RD Design*

Description

The print method for bandwidth selection for 2D local polynomial RD design

Usage

```
## S3 method for class 'rdbw2d'
print(x, ...)
```

Arguments

`x`	Class `rdbw2d` objects, obtained by calling `rdbw2d`.
`...`	Additional arguments passed to the method (currently ignored).

Value

No return value, called to print `rdbw2d` results.

Author(s)

Matias D. Cattaneo, Princeton University. `<matias.d.cattaneo@gmail.com>`
Rocio Titiunik, Princeton University. `<rocio.titiunik@gmail.com>`
Ruiqi Rae Yu, Princeton University. `<raeyuuuu@gmail.com>`

See Also

`rdbw2d` for bandwidth selection for 2D local polynomial RD design

Supported methods: `print.rdbw2d`, `summary.rdbw2d`.

`print.rdbw2d.distance` *Print Method for Bandwidth Selection (Distance-Based) in 2D Local Polynomial RD Design*

Description

Print method for displaying summary information from distance-based bandwidth selection in 2D local polynomial regression discontinuity (RD) designs, as produced by `rdbw2d.distance`.

Usage

```
## S3 method for class 'rdbw2d.distance'
print(x, ...)
```

Arguments

`x`	An object of class `rdbw2d.distance`, returned by `rdbw2d.distance`.
`...`	Additional arguments passed to the method (currently ignored).

Value

No return value. This function is called for its side effects: it prints summary information of `rdbw2d.distance`.

Author(s)

Matias D. Cattaneo, Princeton University. `<matias.d.cattaneo@gmail.com>`
Rocio Titiunik, Princeton University. `<rocio.titiunik@gmail.com>`
Ruiqi Rae Yu, Princeton University. `<raeyuuuu@gmail.com>`

See Also

`rdbw2d.distance` for distance-based bandwidth selection in 2D local polynomial RD design.

Supported methods: `print.rdbw2d.distance`, `summary.rdbw2d.distance`.

`rd2d` *Location-Based Methods for Boundary Discontinuity Design*

Description

`rd2d` implements location-based local polynomial boundary discontinuity (BD) point estimators with robust bias-corrected pointwise confidence intervals and uniform confidence bands. See Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026) for methodological background.

Companion commands are: `rdbw2d` for data-driven bandwidth selection.

For other packages of RD designs, visit `https://rdpackages.github.io/`

Usage

```
rd2d(
  Y,
  X,
  assignment,
  b,
  h = NULL,
  deriv = c(0, 0),
  tangvec = NULL,
  p = 1,
  q = NULL,
  kernel = c("tri", "triangular", "epa", "epanechnikov", "uni", "uniform", "gau",
    "gaussian"),
  kernel_type = c("prod", "rad"),
  vce = c("hc1", "hc0", "hc2", "hc3"),
  masspoints = c("check", "adjust", "off"),
  cluster = NULL,
  covs.eff = NULL,
  covs.drop = TRUE,
  covs.tol = 1e-12,
  fitmethod = c("joint", "separate"),
  level = 95,
  params.other = NULL,
  params.cov = NULL,
  side = c("two", "left", "right"),
  bwselect = c("mserd", "cerrd", "imserd", "icerrd", "msetwo", "certwo", "imsetwo",
    "icertwo", "user provided"),
  bwparam = c("main", "itt"),
  method = c("dpi", "rot"),
```

```
  bwcheck = 50 + p + 1,
  scaleregul = 3,
  scalebiascrct = 1,
  stdvars = TRUE,
  fuzzy = NULL
)
```

Arguments

`Y`	Dependent variable; a numeric vector of length N, where N is the sample size.
`X`	Bivariate running variable (a.k.a score variable); a numeric matrix or data frame of dimension $N \times 2$, with each row $\mathbf{X}_i = (X_{1i}, X_{2i})$.
`assignment`	Treatment assignment indicator; a logical or binary vector indicating assignment to the treated side.
`b`	Evaluation points; a matrix or data frame specifying boundary points $\mathbf{b}_j = (b_{1j}, b_{2j})$, of dimension $J \times 2$.
`h`	Bandwidths. Either a positive scalar (same bandwidth for all dimensions and groups), or a matrix/data frame of size $J \times 4$, corresponding to $h_{\text{control},1}$, $h_{\text{control},2}$, $h_{\text{treated},1}$, $h_{\text{treated},2}$ at each evaluation point. If not specified, bandwidth `h` is computed by the companion command `rdbw2d`; for fuzzy designs, `bwparam` determines whether automatic bandwidths target the fuzzy Wald ratio or the reduced-form outcome discontinuity. Default is `h = NULL`.
`deriv`	The order of the derivatives of the regression functions to be estimated; a non-negative integer vector of length 2 specifying the number of derivatives in each coordinate (e.g., $c(1,2)$ corresponds to $\partial_1 \partial_2^2$).
`tangvec`	Tangent vectors; a matrix or data frame of dimension $J \times 2$ specifying directional derivatives. Overrides `deriv` if provided.
`p`	Polynomial order for point estimation ($p = 1$ by default).
`q`	Polynomial order for robust confidence interval construction. Must satisfy $q \geq p$; default is $q = p + 1$.
`kernel`	Kernel function to use. Options are `"tri"` or `"triangular"` (triangular, default), `"epa"` or `"epanechnikov"` (Epanechnikov), `"uni"` or `"uniform"` (uniform), and `"gau"` or `"gaussian"` (Gaussian).
`kernel_type`	Kernel structure. Either `"prod"` for product kernels (default) or `"rad"` for radial kernels.
`vce`	Variance-covariance estimation method. Options are: • `"hc0"`: heteroskedasticity-robust plug-in residual variance estimator without small-sample adjustment. • `"hc1"`: heteroskedasticity-robust plug-in residual variance estimator with HC1 small-sample adjustment (default). • `"hc2"`: heteroskedasticity-robust plug-in residual variance estimator with HC2 adjustment. • `"hc3"`: heteroskedasticity-robust plug-in residual variance estimator with HC3 adjustment.

	Default is `"hc1"`.
`masspoints`	Handling of mass points in the running variable. Options are: • `"check"`: detects presence of mass points and reports the number of unique observations (default). • `"adjust"`: adjusts preliminary bandwidths to ensure a minimum number of unique observations within each side of the cutoff. • `"off"`: ignores presence of mass points.
`cluster`	Cluster ID variable used for cluster-robust variance estimation with degrees-of-freedom weights. Default is `cluster = NULL`.
`covs.eff`	Optional pre-intervention covariates used for efficiency adjustment. Must be a numeric vector, matrix, or data frame with the same number of rows as `Y`. Covariate coefficients are constrained to be common across treatment sides.
`covs.drop`	Logical. If `TRUE` (default), redundant covariate columns are dropped from the common covariate-adjustment solve when the residualized covariate matrix is rank deficient. If `FALSE`, a generalized inverse is used instead.
`covs.tol`	Positive numeric tolerance used to detect rank deficiency in residualized `covs.eff`. Default is `1e-12`.
`fitmethod`	Estimation, inference, and automatic bandwidth-selection method. `"joint"` (default) uses treatment-side interacted local polynomial fits with joint variance corrections, including joint HC1 degrees-of-freedom factors and, when `cluster` is supplied, joint cluster-score aggregation across treatment sides. `"separate"` uses the legacy side-specific fitting and covariance calculations.
`level`	Nominal confidence level for intervals/bands, between 0 and 100 (default is 95).
`params.other`	Optional character vector requesting companion-side result tables. Options are `"main.0"`, `"main.1"` for sharp fits, and `"itt.0"`, `"itt.1"`, `"fs.0"`, `"fs.1"` for fuzzy fits. Unrequested companion tables are returned as `NA`. Default is `NULL`.
`params.cov`	Optional character vector requesting covariance matrices to store for later use by `summary.rd2d`. Options are `"main"`, `"main.0"`, `"main.1"`, `"itt"`, `"itt.0"`, `"itt.1"`, `"fs"`, `"fs.0"`, and `"fs.1"`, subject to the sharp/fuzzy design and `params.other`. Default is `NULL`, which computes no covariance matrices.
`side`	Type of confidence interval. Options: `"two"` (two-sided, default), `"left"` (left tail), or `"right"` (right tail).
`bwselect`	Bandwidth selection strategy. Options: • `"mserd"`. One common MSE-optimal bandwidth selector for the boundary RD treatment effect estimator for each evaluation point (default). • `"cerrd"`. CER-optimal counterpart of `"mserd"`. • `"imserd"`. IMSE-optimal bandwidth selector for the boundary RD treatment effect estimator based on all evaluation points. • `"icerrd"`. CER-optimal counterpart of `"imserd"`. • `"msetwo"`. Two different MSE-optimal bandwidth selectors (control and treatment) for the boundary RD treatment effect estimator for each evaluation point. • `"certwo"`. CER-optimal counterpart of `"msetwo"`.

- `"imsetwo"`. Two IMSE-optimal bandwidth selectors (control and treatment) for the boundary RD treatment effect estimator based on all evaluation points.
- `"icertwo"`. CER-optimal counterpart of `"imsetwo"`.
- `"user provided"`. User-provided bandwidths. If `h` is not `NULL`, then `bwselect` is overwritten to `"user provided"`.

`bwparam`	Target parameter used for fuzzy automatic bandwidth selection. Options are `"main"` (default), which selects bandwidths for the linearized fuzzy Wald ratio, and `"itt"`, which selects bandwidths for the reduced-form outcome discontinuity. This option is ignored in sharp designs and when `h` is supplied.
`method`	Bandwidth selection method for bias estimator based on local polynomials. Either `"dpi"` (default) for data-driven plug-in MSE optimal bandwidth selector or `"rot"` for rule-of-thumb bandwidth selector.
`bwcheck`	If a positive integer is provided, the preliminary bandwidth used in the calculations is enlarged so that at least `bwcheck` unique observations are used. Default is `50 + p + 1`.
`scaleregul`	Scaling factor for the regularization term in bandwidth selection. Default is 3.
`scalebiascrct`	Scaling factor used for bias correction based on higher order expansions. Default is 1.
`stdvars`	Logical. If `TRUE`, the running variables X_{1i} and X_{2i} are standardized before computing automatic bandwidths. Default is `TRUE`. Standardization only affects automatic bandwidth selection when bandwidths are not manually provided through `h`.
`fuzzy`	Optional treatment receipt/status variable used for fuzzy RD estimation. The default is `NULL`, which estimates the sharp RD design.

Value

An object of class `"rd2d"`, a list with components:

`main` A data frame with point estimates, standard errors, t-statistics, p-values, confidence intervals, and bandwidths at each evaluation point.

`b1`, `b2` First and second coordinate of evaluation points $\mathbf{b} = (b_1, b_2)$.

`estimate.p` Point estimate $\widehat{\tau}_p(\mathbf{b})$.

`std.err.p` Standard error of $\widehat{\tau}_p(\mathbf{b})$.

`estimate.q` Bias-corrected point estimate $\widehat{\tau}_q(\mathbf{b})$.

`std.err.q` Standard error of the bias-corrected estimate $\widehat{\tau}_q(\mathbf{b})$.

`t.value`, `p.value` t-statistic and p-value based on the bias-corrected estimate.

`ci.lower`, `ci.upper` Pointwise confidence intervals.

`h01`, `h02`, `h11`, `h12` Bandwidths used in each coordinate and group.

`N.Co`, `N.Tr` Effective sample size for the control and treatment sides, respectively.

`main.0`, `main.1` For sharp RD only, companion-side outcome tables requested through `params.other`; otherwise `NA`.

`bw` Bandwidth and effective sample size table.

`itt` For fuzzy RD only, same structure as `main` but for the reduced-form or intention-to-treat outcome discontinuity.

`itt.0`, `itt.1` For fuzzy RD only, outcome summaries for the control and treated sides requested through `params.other`; otherwise `NA`.

`fs` For fuzzy RD only, same structure as `main` but for the first-stage treatment receipt/status discontinuity.

`fs.0`, `fs.1` For fuzzy RD only, first-stage summaries for the control and treated sides requested through `params.other`; otherwise `NA`.

`tau.hat` Point estimates at each evaluation point.

`tau.hat.q` Bias-corrected estimates at each evaluation point.

`se.hat` Standard errors corresponding to `tau.hat`.

`se.hat.q` Standard errors corresponding to `tau.hat.q`.

`params.cov` List of covariance matrices for aggregate inference in `summary.rd2d`. Contains only entries requested through `params.cov`.

`ci` List with pointwise confidence intervals.

`pvalues` Two-sided p-values based on bias-corrected estimates.

`tvalues` t-statistics based on bias-corrected estimates.

`tau.itt`, `tau.itt.q` For fuzzy RD, reduced-form outcome estimates using polynomial orders p and q; otherwise `NA`.

`tau.fs`, `tau.fs.q` For fuzzy RD, first-stage treatment receipt/status estimates using polynomial orders p and q; otherwise `NA`.

`rdmodel` Character label describing the fitted RD model.

`call` Matched function call.

`opt` List of options used in the function call.

Author(s)

Matias D. Cattaneo, Princeton University. <`matias.d.cattaneo@gmail.com`>
Rocio Titiunik, Princeton University. <`rocio.titiunik@gmail.com`>
Ruiqi Rae Yu, Princeton University. <`raeyuuuu@gmail.com`>

References

- Cattaneo, M. D., and Titiunik, R. (2022). Regression Discontinuity Designs. doi:10.1146/annureveconomics051520021409.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026). Estimation and Inference in Boundary Discontinuity Designs: Location-Based Methods.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026). Boundary Discontinuity Designs: Theory and Practice.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2025). rd2d: Causal Inference in Boundary Discontinuity Designs.

See Also

`rdbw2d`, `print.rd2d`, `summary.rd2d`

Examples

```
# Simulated example
set.seed(123)
n <- 800
X1 <- rnorm(n)
X2 <- rnorm(n)
assignment <- as.numeric(X1 > 0)
Y <- 3 + 2 * X1 + 1.5 * X2 + assignment + rnorm(n)
X <- cbind(X1, X2)
b <- matrix(c(0, 0, 0, 1), ncol = 2)

# Estimate treatment effect using rd2d
result <- rd2d(Y, X, assignment, b, params.cov = "main",
               masspoints = "off", bwcheck = 10)
print(result)
summary(result, cbands = "main", repp = 49)

# Fuzzy RD example
fuzzy <- as.numeric(runif(n) < ifelse(assignment == 1, 0.8, 0.2))
Y.fuzzy <- 3 + 2 * X1 + 1.5 * X2 + 1.5 * fuzzy + rnorm(n)
result.fuzzy <- rd2d(Y.fuzzy, X, assignment, b, fuzzy = fuzzy,
                     bwparam = "main", masspoints = "off",
                     bwcheck = 10)
print(result.fuzzy)
summary(result.fuzzy, output = "itt")
```

`rd2d.distance` *Distance-Based Methods for Boundary Discontinuity Design*

Description

`rd2d.distance` implements distance-based local polynomial boundary discontinuity (BD) point estimators with robust bias-corrected pointwise confidence intervals and uniform confidence bands. The command targets level effects at two-dimensional boundary points. See Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026) for methodological background.

Companion commands are: `rdbw2d.distance` for data-driven bandwidth selection.

For other packages of RD designs, visit `https://rdpackages.github.io/`

Usage

```
rd2d.distance(
  Y,
  distance,
  h = NULL,
```

```
  b = NULL,
  p = 1,
  q = NULL,
  kink.unknown = c(FALSE, FALSE),
  kink.position = NULL,
 kernel = c("tri", "triangular", "epa", "epanechnikov", "uni", "uniform", "gau",
    "gaussian"),
  level = 95,
  cbands = TRUE,
  side = c("two", "left", "right"),
 bwselect = c("mserd", "cerrd", "imserd", "icerrd", "msetwo", "certwo", "imsetwo",
    "icertwo", "user provided"),
  bwparam = c("main", "itt"),
  params.other = NULL,
  params.cov = NULL,
  vce = c("hc1", "hc0", "hc2", "hc3"),
  bwcheck = 50 + p + 1,
  masspoints = c("check", "adjust", "off"),
  cluster = NULL,
  covs.eff = NULL,
  covs.drop = TRUE,
  covs.tol = 1e-12,
  fitmethod = c("joint", "separate"),
  scaleregul = 1,
  cqt = 0.5,
  fuzzy = NULL
)
```

Arguments

`Y`	Dependent variable; a numeric vector of length N, where N is the sample size.
`distance`	Signed distance scores; a numeric matrix or data frame of dimension $N \times J$, where N is the sample size and J is the number of evaluation points. Non-negative values identify observations on the treated side and negative values identify observations on the control side.
`h`	Bandwidths. A positive scalar uses the same bandwidth for both groups and all evaluation points. A matrix/data frame of size $J \times 2$ uses row-specific control and treated bandwidths. If not specified, bandwidths are selected by `rdbw2d.distance`.
`b`	Optional evaluation points; a matrix or data frame specifying boundary points $\mathbf{b}_j = (b_{1j}, b_{2j})$, dimension $J \times 2$.
`p`	Polynomial order for point estimation. Default is `p = 1`.
`q`	Polynomial order for bias-corrected estimation. Must satisfy $q \geq p$. Default is `q = p + 1`, except when `kink.unknown[1] = TRUE`, where the default is `q = p`.
`kink.unknown`	Logical value or vector of length 2 controlling unknown-kink bandwidth adjustments. A scalar `TRUE` is expanded to `c(TRUE, TRUE)`, and a scalar `FALSE` is expanded to `c(FALSE, FALSE)`. With the vector form, the first element controls

	whether the point-estimation bandwidth is shrunk to the unknown-kink rate; the second controls whether the inference bandwidth is further shrunk. Default is `c(FALSE, FALSE)`. The second element can be `TRUE` only when the first element is `TRUE`.
`kink.position`	Optional boundary positions of known kink points. Either a logical vector with one entry per boundary point, where `TRUE` identifies a kink point, or an integer vector with indices between 1 and the number of boundary points. Requires `b` so distances between boundary points can be computed. This option applies only to automatic bandwidth selection.
`kernel`	Kernel function to use. Options are `"tri"` or `"triangular"` (triangular, default), `"epa"` or `"epanechnikov"` (Epanechnikov), `"uni"` or `"uniform"` (uniform), and `"gau"` or `"gaussian"` (Gaussian).
`level`	Nominal confidence level for intervals/bands, between 0 and 100 (default is 95).
`cbands`	Logical. If `TRUE`, stores the covariance matrix needed for uniform confidence bands and aggregate inference computed by `summary(..., cbands = "main")`, `summary(..., WBATE = weights)`, or `summary(..., LBATE = TRUE)`. Default is `TRUE`.
`side`	Type of confidence interval. Options: `"two"` (two-sided, default), `"left"` (left tail), or `"right"` (right tail).
`bwselect`	Bandwidth selection strategy. Options: • `"mserd"`. One common MSE-optimal bandwidth selector for the boundary RD treatment effect estimator for each evaluation point (default). • `"cerrd"`. CER-optimal counterpart of `"mserd"`. • `"imserd"`. IMSE-optimal bandwidth selector for the boundary RD treatment effect estimator based on all evaluation points. • `"icerrd"`. CER-optimal counterpart of `"imserd"`. • `"msetwo"`. Two different MSE-optimal bandwidth selectors (control and treatment) for the boundary RD treatment effect estimator for each evaluation point. • `"certwo"`. CER-optimal counterpart of `"msetwo"`. • `"imsetwo"`. Two IMSE-optimal bandwidth selectors (control and treatment) for the boundary RD treatment effect estimator based on all evaluation points. • `"icertwo"`. CER-optimal counterpart of `"imsetwo"`. • `"user provided"`. User-provided bandwidths. If `h` is not `NULL`, then `bwselect` is overwritten to `"user provided"`.
`bwparam`	Target parameter used for fuzzy automatic bandwidth selection. Options are `"main"` (default), which selects bandwidths for the linearized fuzzy Wald ratio, and `"itt"`, which selects bandwidths for the reduced-form outcome. Ignored in sharp designs.
`params.other`	Optional character vector requesting companion output tables. In sharp designs, available values are `"main.0"` and `"main.1"`. In fuzzy designs, available values are `"itt.0"`, `"itt.1"`, `"fs.0"`, and `"fs.1"`.

`params.cov`	Optional character vector requesting covariance matrices for aggregate inference or confidence bands. Available values are `"main"` for sharp and fuzzy main effects, plus `"main.0"` and `"main.1"` for sharp side-specific effects, and `"itt"`, `"itt.0"`, `"itt.1"`, `"fs"`, `"fs.0"`, and `"fs.1"` for fuzzy reduced-form and first-stage effects. If `cbands = TRUE`, `"main"` is added automatically.
`vce`	Variance-covariance estimator for standard errors. Options: `"hc0"` Heteroskedasticity-robust variance estimator without small sample adjustment (White robust). `"hc1"` Heteroskedasticity-robust variance estimator with degrees-of-freedom correction. (default) `"hc2"` Heteroskedasticity-robust variance estimator using leverage adjustments. `"hc3"` More conservative heteroskedasticity-robust variance estimator (similar to jackknife correction).
`bwcheck`	If a positive integer is provided, the preliminary bandwidth used in the calculations is enlarged so that at least `bwcheck` unique observations are used. Default is `50 + p + 1`.
`masspoints`	Strategy for handling mass points in the running variable. Options: `"check"` (default) Check for repeated values and adjust inference if needed. `"adjust"` Adjust bandwidths to guarantee a sufficient number of unique support points. `"off"` Ignore mass points completely.
`cluster`	Cluster ID variable used for cluster-robust variance estimation with degrees-of-freedom weights. Default is `cluster = NULL`.
`covs.eff`	Optional pre-intervention covariates used for efficiency adjustment. Must be a numeric vector, matrix, or data frame with the same number of rows as `Y`. Covariate coefficients are constrained to be common across treatment sides.
`covs.drop`	Logical. If `TRUE` (default), redundant covariate columns are dropped from the common covariate-adjustment solve when the residualized covariate matrix is rank deficient. If `FALSE`, a generalized inverse is used instead.
`covs.tol`	Positive numeric tolerance used to detect rank deficiency in residualized `covs.eff`. Default is `1e-12`.
`fitmethod`	Estimation, inference, and automatic bandwidth-selection method. `"joint"` (default) uses treatment-side interacted local polynomial fits with joint variance corrections. `"separate"` uses the legacy side-specific fitting and covariance calculations.
`scaleregul`	Scaling factor for the regularization term in bandwidth selection. Default is `1`.
`cqt`	Constant controlling subsample fraction for initial bias estimation. Default is `0.5`.
`fuzzy`	Optional treatment receipt/status variable for fuzzy RD designs. If supplied, `main` reports the fuzzy Wald treatment effect, `itt` reports the reduced-form outcome discontinuity, and `fs` reports the first-stage treatment receipt/status discontinuity.

Value

An object of class `"rd2d.distance"`, a list containing:

- `main` Data frame of point estimates, standard errors, confidence intervals, and bandwidths:
  - `b1` First coordinate of the evaluation point.
  - `b2` Second coordinate of the evaluation point.
  - `estimate.p` Point estimate with polynomial order p.
  - `std.err.p` Standard error for `estimate.p`.
  - `estimate.q` Bias-corrected estimate with polynomial order q.
  - `std.err.q` Standard error for `estimate.q`.
  - `t.value` t-statistic based on $\widehat{\tau}_{\text{distance},q}(\mathbf{b})$.
  - `p.value` Two-sided p-value based on $T_{\text{distance},q}(\mathbf{b})$.
  - `ci.lower` Lower bound of confidence interval.
  - `ci.upper` Upper bound of confidence interval.
  - `h0` Bandwidth used for the control group (negative signed distance).
  - `h1` Bandwidth used for the treatment group (non-negative signed distance).
  - `h0.rbc` Bandwidth used for control-side inference.
  - `h1.rbc` Bandwidth used for treated-side inference.
  - `N.Co` Effective sample size for the control side.
  - `N.Tr` Effective sample size for the treatment side.
- `main.0` Sharp-design summary table for the control side only.
- `main.1` Sharp-design summary table for the treated side only.
- `itt` Fuzzy-design reduced-form outcome summary table.
- `itt.0`, `itt.1` Optional fuzzy-design side-specific reduced-form outcome summary tables requested through `params.other`.
- `fs` Fuzzy-design first-stage treatment receipt/status summary table.
- `fs.0`, `fs.1` Optional fuzzy-design side-specific first-stage treatment receipt/status summary tables requested through `params.other`.
- `bw` Bandwidth and effective-sample-size table.
- `tau.hat` Vector of point estimates $\widehat{\tau}_p(\mathbf{b})$.
- `tau.hat.q` Vector of bias-corrected estimates $\widehat{\tau}_q(\mathbf{b})$.
- `se.hat` Standard errors corresponding to $\widehat{\tau}_p(\mathbf{b})$.
- `se.hat.q` Standard errors corresponding to $\widehat{\tau}_q(\mathbf{b})$.
- `params.cov` List containing covariance matrices requested for aggregate or uniform inference.
- `ci` Pointwise confidence interval endpoints.
- `pvalues` Two-sided p-values based on bias-corrected estimates.
- `tvalues` t-statistics based on bias-corrected estimates.
- `tau.itt`, `tau.itt.q`, `tau.fs`, `tau.fs.q` Fuzzy reduced-form and first-stage point estimates with polynomial orders p and q; these are `NULL` in sharp designs.
- `rdmodel` Character label describing the fitted RD model.
- `call` Matched function call.
- `opt` A list of estimation options (e.g., `p`, `q`, `kernel`, `level`, etc.) and internal variables such as sample size N.

Author(s)

Matias D. Cattaneo, Princeton University. <`matias.d.cattaneo@gmail.com`>
Rocio Titiunik, Princeton University. <`rocio.titiunik@gmail.com`>
Ruiqi Rae Yu, Princeton University. <`raeyuuuu@gmail.com`>

References

- Cattaneo, M. D., and Titiunik, R. (2022). Regression Discontinuity Designs. doi:10.1146/annureveconomics051520021409.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026). Estimation and Inference in Boundary Discontinuity Designs: Distance-Based Methods.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026). Boundary Discontinuity Designs: Theory and Practice.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2025). rd2d: Causal Inference in Boundary Discontinuity Designs.

See Also

`rdbw2d.distance`, `rd2d`, `print.rd2d.distance`, `summary.rd2d.distance`

Examples

```
set.seed(123)
n <- 800

# Generate running variables x1 and x2
x1 <- rnorm(n)
x2 <- rnorm(n)

# Define treatment assignment: treated if x1 >= 0
assignment <- as.numeric(x1 >= 0)

# Generate outcome variable Y with some treatment effect
Y <- 3 + 2 * x1 + 1.5 * x2 + 1.5 * assignment + rnorm(n, sd = 0.5)

# Define evaluation points (e.g., at the origin and another point)
eval <- data.frame(x.1 = c(0, 0), x.2 = c(0, 1))

# Compute Euclidean distances to evaluation points
distance.a <- sqrt((x1 - eval$x.1[1])^2 + (x2 - eval$x.2[1])^2)
distance.b <- sqrt((x1 - eval$x.1[2])^2 + (x2 - eval$x.2[2])^2)

# Combine distances into a matrix
distance <- as.data.frame(cbind(distance.a, distance.b))

# Assign positive distances for treatment group, negative for control
assignment_expanded <- matrix(rep(2 * assignment - 1, times = ncol(distance)),
                              nrow = nrow(distance), ncol = ncol(distance))
distance <- distance * assignment_expanded
```

```
# Run the rd2d.distance function
result <- rd2d.distance(Y, distance = distance, b = eval, cbands = FALSE,
                    masspoints = "off", bwcheck = 10)

# View the estimation results
print(result)
summary(result)

# Fuzzy distance-based fit with a user-supplied bandwidth
fuzzy <- as.numeric(runif(n) < ifelse(assignment == 1, 0.8, 0.2))
Y.fuzzy <- 3 + 2 * x1 + 1.5 * x2 + 1.5 * fuzzy + rnorm(n, sd = 0.5)
fuzzy.result <- rd2d.distance(Y.fuzzy, distance = distance, h = 0.8, b = eval, fuzzy = fuzzy,
                          cbands = FALSE, masspoints = "off")
summary(fuzzy.result)
```

`rdbw2d` *Bandwidth Selection for Location-Based Methods for Boundary Discontinuity Design*

Description

`rdbw2d` implements bandwidth selectors for location-based local polynomial boundary discontinuity (BD) estimation and inference. See Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026) for methodological background.

Companion commands are: `rd2d` for point estimation and inference procedures.

For other packages of RD designs, visit `https://rdpackages.github.io/`

Usage

```
rdbw2d(
  Y,
  X,
  assignment,
  b,
  p = 1,
  deriv = c(0, 0),
  tangvec = NULL,
 kernel = c("tri", "triangular", "epa", "epanechnikov", "uni", "uniform", "gau",
    "gaussian"),
  kernel_type = c("prod", "rad"),
 bwselect = c("mserd", "cerrd", "imserd", "icerrd", "msetwo", "certwo", "imsetwo",
    "icertwo"),
  bwparam = c("main", "itt"),
  method = c("dpi", "rot"),
  vce = c("hc1", "hc0", "hc2", "hc3"),
  bwcheck = 20,
  masspoints = c("check", "adjust", "off"),
```

```
  cluster = NULL,
  covs.eff = NULL,
  covs.drop = TRUE,
  covs.tol = 1e-12,
  fitmethod = c("joint", "separate"),
  scaleregul = 1,
  scalebiascrct = 1,
  stdvars = TRUE,
  fuzzy = NULL
)
```

Arguments

`Y`	Dependent variable; a numeric vector of length N, where N is the sample size.
`X`	Bivariate running variable (a.k.a score variable); a numeric matrix or data frame of dimension $N \times 2$, with each row $\mathbf{X}_i = (X_{1i}, X_{2i})$.
`assignment`	Treatment assignment indicator; a logical or binary vector indicating assignment to the treated side.
`b`	Evaluation points; a matrix or data frame specifying boundary points $\mathbf{b}_j = (b_{1j}, b_{2j})$, of dimension $J \times 2$.
`p`	Polynomial order of the local polynomial estimator. Default is `p = 1`.
`deriv`	The order of the derivatives of the regression functions to be estimated; a non-negative integer vector of length 2 specifying the number of derivatives in each coordinate (e.g., $c(1, 2)$ corresponds to $\partial_1 \partial_2^2$).
`tangvec`	Tangent vectors; a matrix or data frame of dimension $J \times 2$ specifying directional derivatives. Overrides `deriv` if provided.
`kernel`	Kernel function to use. Options are `"tri"` or `"triangular"` (triangular, default), `"epa"` or `"epanechnikov"` (Epanechnikov), `"uni"` or `"uniform"` (uniform), and `"gau"` or `"gaussian"` (Gaussian).
`kernel_type`	Kernel structure. Either `"prod"` for product kernels (default) or `"rad"` for radial kernels.
`bwselect`	Bandwidth selection strategy. Options: • `"mserd"`. One common MSE-optimal bandwidth selector for the boundary RD treatment effect estimator for each evaluation point (default). • `"cerrd"`. CER-optimal counterpart of `"mserd"`. • `"imserd"`. IMSE-optimal bandwidth selector for the boundary RD treatment effect estimator based on all evaluation points. • `"icerrd"`. CER-optimal counterpart of `"imserd"`. • `"msetwo"`. Two different MSE-optimal bandwidth selectors (control and treatment) for the boundary RD treatment effect estimator for each evaluation point. • `"certwo"`. CER-optimal counterpart of `"msetwo"`. • `"imsetwo"`. Two IMSE-optimal bandwidth selectors (control and treatment) for the boundary RD treatment effect estimator based on all evaluation points.

	- `"icertwo"`. CER-optimal counterpart of `"imsetwo"`.
`bwparam`	Target parameter used for fuzzy bandwidth selection. Options are `"main"` (default), which selects bandwidths for the linearized fuzzy Wald ratio, and `"itt"`, which selects bandwidths for the reduced-form outcome discontinuity. This option is ignored in sharp designs, where only `"main"` is used.
`method`	Bandwidth selection method for bias estimator based on local polynomials. Either `"dpi"` (default) for data-driven plug-in MSE optimal bandwidth selector or `"rot"` for rule-of-thumb bandwidth selector.
`vce`	Variance-covariance estimation method. Options are: - `"hc0"`: heteroskedasticity-robust plug-in residual variance estimator without small-sample adjustment. - `"hc1"`: heteroskedasticity-robust plug-in residual variance estimator with HC1 small-sample adjustment (default). - `"hc2"`: heteroskedasticity-robust plug-in residual variance estimator with HC2 adjustment. - `"hc3"`: heteroskedasticity-robust plug-in residual variance estimator with HC3 adjustment. Default is `"hc1"`.
`bwcheck`	If a positive integer is provided, the preliminary bandwidth used in the calculations is enlarged so that at least `bwcheck` unique observations are used. Default is `20`.
`masspoints`	Handling of mass points in the running variable. Options are: - `"check"`: detects presence of mass points and reports the number of unique observations (default). - `"adjust"`: adjusts preliminary bandwidths to ensure a minimum number of unique observations within each side of the cutoff. - `"off"`: ignores presence of mass points.
`cluster`	Cluster ID variable used for cluster-robust variance estimation with degrees-of-freedom weights. Default is `cluster = NULL`.
`covs.eff`	Optional pre-intervention covariates used for efficiency adjustment. Must be a numeric vector, matrix, or data frame with the same number of rows as `Y`. Covariate coefficients are constrained to be common across treatment sides.
`covs.drop`	Logical. If `TRUE` (default), redundant covariate columns are dropped from the common covariate-adjustment solve when the residualized covariate matrix is rank deficient. If `FALSE`, a generalized inverse is used instead.
`covs.tol`	Positive numeric tolerance used to detect rank deficiency in residualized `covs.eff`. Default is `1e-12`.
`fitmethod`	Estimation method used for the variance constants entering bandwidth selection. `"joint"` (default) uses treatment-side interacted local polynomial fits with joint variance corrections, matching the default in `rd2d`. `"separate"` uses the legacy side-specific variance calculations.
`scaleregul`	Scaling factor for the regularization term in bandwidth selection. Default is 1.
`scalebiascrct`	Scaling factor used for bias correction based on higher order expansions. Default is 1.

`stdvars`	Logical. If `TRUE`, the running variables X_{1i} and X_{2i} are standardized before computing the bandwidths. Default is `TRUE`.
`fuzzy`	Optional treatment receipt/status variable used to construct bandwidth selectors for fuzzy RD estimation. If supplied, `bwparam` controls whether the selector targets the fuzzy Wald ratio or the reduced-form outcome discontinuity.

Value

A list of class `"rdbw2d"` containing:

- `bws` Data frame of estimated bandwidths for each evaluation point:
  - `b1` First coordinate of the evaluation point.
  - `b2` Second coordinate of the evaluation point.
  - `h01` Estimated bandwidth for X_{1i} in the control group ($\mathcal{A}_0$).
  - `h02` Estimated bandwidth for X_{2i} in the control group ($\mathcal{A}_0$).
  - `h11` Estimated bandwidth for X_{1i} in the treatment group ($\mathcal{A}_1$).
  - `h12` Estimated bandwidth for X_{2i} in the treatment group ($\mathcal{A}_1$).
- `mseconsts` Data frame of intermediate quantities used in bandwidth calculation:
  - `N.Co` Effective sample size for the control group $\mathcal{A}_0$.
  - `N.Tr` Effective sample size for the treatment group $\mathcal{A}_1$.
  - `bias.0` Bias constant estimate for the control group.
  - `bias.1` Bias constant estimate for the treatment group.
  - `var.0` Variance constant estimate for the control group.
  - `var.1` Variance constant estimate for the treatment group.
  - `var.01` Covariance constant between the control and treatment variance scores, used by joint clustered common-bandwidth selectors.
  - `reg.bias.0` Bias correction adjustment for the control group.
  - `reg.bias.1` Bias correction adjustment for the treatment group.
  - `reg.var.0` Variance of the bias estimate for the control group.
  - `reg.var.1` Variance of the bias estimate for the treatment group.
  - `reg.var.01` Covariance counterpart of `reg.var.0` and `reg.var.1`, used by joint clustered common-bandwidth selectors.
- `opt` List containing:
  - `p` Polynomial order used for estimation.
  - `kernel` Kernel function used.
  - `kernel_type` Type of kernel (product or radial).
  - `stdvars` Logical indicating if standardization was applied.
  - `bwselect` Bandwidth selection strategy used.
  - `bwparam` Target parameter for fuzzy bandwidth selection.
  - `method` Bandwidth estimation method.
  - `vce` Variance estimation method.
  - `fitmethod` Variance-fitting method used by the selector.
  - `scaleregul` Scaling factor for regularization.
  - `scalebiascrct` Scaling factor for bias correction.
  - `N` Total sample size N.
- `call` Matched function call.

Author(s)

Matias D. Cattaneo, Princeton University. `<matias.d.cattaneo@gmail.com>`
Rocio Titiunik, Princeton University. `<rocio.titiunik@gmail.com>`
Ruiqi Rae Yu, Princeton University. `<raeyuuuu@gmail.com>`

References

- Cattaneo, M. D., and Titiunik, R. (2022). Regression Discontinuity Designs. doi:10.1146/annureveconomics051520021409.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026). Estimation and Inference in Boundary Discontinuity Designs: Location-Based Methods.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026). Boundary Discontinuity Designs: Theory and Practice.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2025). rd2d: Causal Inference in Boundary Discontinuity Designs.

See Also

`rd2d`, `print.rdbw2d`, `summary.rdbw2d`

Examples

```
# Simulated example
set.seed(123)
n <- 800
X1 <- rnorm(n)
X2 <- rnorm(n)
assignment <- as.numeric(X1 > 0)
Y <- 3 + 2 * X1 + 1.5 * X2 + assignment + rnorm(n)
X <- cbind(X1, X2)
b <- matrix(c(0, 0, 0, 1), ncol = 2)

# MSE optimal bandwidth for rd2d
bws <- rdbw2d(Y, X, assignment, b, masspoints = "off", bwcheck = 10)

# View the bandwidth selection results
print(bws)
summary(bws)

# Fuzzy bandwidth selection can target the Wald ratio or the reduced form
fuzzy <- as.numeric(runif(n) < ifelse(assignment == 1, 0.8, 0.2))
Y.fuzzy <- 3 + 2 * X1 + 1.5 * X2 + 1.5 * fuzzy + rnorm(n)
bws.fuzzy <- rdbw2d(Y.fuzzy, X, assignment, b, fuzzy = fuzzy, bwparam = "main",
                    masspoints = "off", bwcheck = 10)
print(bws.fuzzy)
```

`rdbw2d.distance` *Bandwidth Selection for Distance-Based Methods for Boundary Discontinuity Design*

Description

`rdbw2d.distance` implements bandwidth selectors for distance-based local polynomial boundary discontinuity (BD) estimation and inference. The command targets level effects at two-dimensional boundary points. See Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026) for methodological background.

Usage

```
rdbw2d.distance(
  Y,
  distance,
  b = NULL,
  p = 1,
  kink.unknown = c(FALSE, FALSE),
  kink.position = NULL,
 kernel = c("tri", "triangular", "epa", "epanechnikov", "uni", "uniform", "gau",
    "gaussian"),
 bwselect = c("mserd", "cerrd", "imserd", "icerrd", "msetwo", "certwo", "imsetwo",
    "icertwo"),
  bwparam = c("main", "itt"),
  vce = c("hc1", "hc0", "hc2", "hc3"),
  bwcheck = 20 + p + 1,
  masspoints = c("check", "adjust", "off"),
  cluster = NULL,
  covs.eff = NULL,
  covs.drop = TRUE,
  covs.tol = 1e-12,
  fitmethod = c("joint", "separate"),
  scaleregul = 1,
  cqt = 0.5,
  fuzzy = NULL
)
```

Arguments

`Y`	Dependent variable; a numeric vector of length N, where N is the sample size.
`distance`	Signed distance scores; a numeric matrix or data frame of dimension $N \times J$, where N is the sample size and J is the number of evaluation points. Non-negative values identify observations on the treated side and negative values identify observations on the control side.
`b`	Optional evaluation points; a matrix or data frame specifying boundary points $\mathbf{b}_j = (b_{1j}, b_{2j})$, dimension $J \times 2$.

`p`	Polynomial order for point estimation. Default is `p = 1`.
`kink.unknown`	Logical value or vector of length 2 controlling unknown-kink bandwidth adjustments. A scalar `TRUE` is expanded to `c(TRUE, TRUE)`, and a scalar `FALSE` is expanded to `c(FALSE, FALSE)`. With the vector form, the first element controls whether the point-estimation bandwidth is shrunk to the unknown-kink rate; the second controls whether the inference bandwidth is further shrunk in `rd2d.distance`. For bandwidth selection, only the first element is used. Default is `c(FALSE, FALSE)`.
`kink.position`	Optional boundary positions of known kink points. Either a logical vector with one entry per boundary point, where `TRUE` identifies a kink point, or an integer vector with indices between 1 and the number of boundary points. Requires `b` so distances between boundary points can be computed.
`kernel`	Kernel function to use. Options are `"tri"` or `"triangular"` (triangular, default), `"epa"` or `"epanechnikov"` (Epanechnikov), `"uni"` or `"uniform"` (uniform), and `"gau"` or `"gaussian"` (Gaussian).
`bwselect`	Bandwidth selection strategy. Options: • `"mserd"`. One common MSE-optimal bandwidth selector for the boundary RD treatment effect estimator for each evaluation point (default). • `"cerrd"`. CER-optimal counterpart of `"mserd"`. • `"imserd"`. IMSE-optimal bandwidth selector for the boundary RD treatment effect estimator based on all evaluation points. • `"icerrd"`. CER-optimal counterpart of `"imserd"`. • `"msetwo"`. Two different MSE-optimal bandwidth selectors (control and treatment) for the boundary RD treatment effect estimator for each evaluation point. • `"certwo"`. CER-optimal counterpart of `"msetwo"`. • `"imsetwo"`. Two IMSE-optimal bandwidth selectors (control and treatment) for the boundary RD treatment effect estimator based on all evaluation points. • `"icertwo"`. CER-optimal counterpart of `"imsetwo"`.
`bwparam`	Target parameter used for fuzzy automatic bandwidth selection. Options are `"main"`, which selects bandwidths for the linearized fuzzy Wald ratio, and `"itt"`, which selects bandwidths for the reduced-form outcome. Ignored in sharp designs.
`vce`	Variance-covariance estimator for standard errors. Options: `"hc0"` Heteroskedasticity-robust variance estimator without small sample adjustment (White robust). `"hc1"` Heteroskedasticity-robust variance estimator with degrees-of-freedom correction. (default) `"hc2"` Heteroskedasticity-robust variance estimator using leverage adjustments. `"hc3"` More conservative heteroskedasticity-robust variance estimator (similar to jackknife correction).
`bwcheck`	If a positive integer is provided, the preliminary bandwidth used in the calculations is enlarged so that at least `bwcheck` unique observations are used. Default is `20 + p + 1`.

`masspoints`	Strategy for handling mass points in the running variable. Options: `"check"` (default) Check for repeated values and adjust inference if needed. `"adjust"` Adjust bandwidths to guarantee a sufficient number of unique support points. `"off"` Ignore mass points completely.
`cluster`	Cluster ID variable used for cluster-robust variance estimation with degrees-of-freedom weights. Default is `cluster = NULL`.
`covs.eff`	Optional pre-intervention covariates used for efficiency adjustment. Must be a numeric vector, matrix, or data frame with the same number of rows as `Y`. Covariate coefficients are constrained to be common across treatment sides.
`covs.drop`	Logical. If `TRUE` (default), redundant covariate columns are dropped from the common covariate-adjustment solve when the residualized covariate matrix is rank deficient. If `FALSE`, a generalized inverse is used instead.
`covs.tol`	Positive numeric tolerance used to detect rank deficiency in residualized `covs.eff`. Default is `1e-12`.
`fitmethod`	Estimation method used for the variance constants entering bandwidth selection. `"joint"` (default) uses treatment-side interacted local polynomial fits with joint variance corrections, matching the default in `rd2d.distance`. `"separate"` uses the legacy side-specific variance calculations.
`scaleregul`	Scaling factor for the regularization term in bandwidth selection. Default is `1`.
`cqt`	Constant controlling subsample fraction for initial bias estimation. Default is `0.5`.
`fuzzy`	Optional treatment receipt/status variable used for fuzzy RD bandwidth selection. If supplied, `bwparam` controls whether the selector targets the fuzzy Wald ratio or the reduced-form outcome.

Value

An object of class `"rdbw2d.distance"`, containing:

- `bws` Data frame of optimal bandwidths for each evaluation point:
  - `b1` First coordinate of the evaluation point $b1$.
  - `b2` Second coordinate of the evaluation point $b2$.
  - `h0` Bandwidth for observations with negative signed distance.
  - `h1` Bandwidth for observations with non-negative signed distance.
- `mseconsts` Data frame of intermediate bias and variance constants used for MSE/IMSE calculations, including `v.01`, the cross-side covariance constant used by joint clustered bandwidths, and `N.Co` and `N.Tr`, the effective sample sizes for observations with negative and non-negative signed distances, respectively.
- `opt` A list of options and settings used in estimation, including `p`, `kernel`, sample size N, and user-specified choices.
- `call` Matched function call.

Author(s)

Matias D. Cattaneo, Princeton University. <matias.d.cattaneo@gmail.com>
Rocio Titiunik, Princeton University. <rocio.titiunik@gmail.com>
Ruiqi Rae Yu, Princeton University. <raeyuuuu@gmail.com>

References

- Cattaneo, M. D., and Titiunik, R. (2022). Regression Discontinuity Designs. doi:10.1146/annureveconomics051520021409.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026). Estimation and Inference in Boundary Discontinuity Designs: Distance-Based Methods.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2026). Boundary Discontinuity Designs: Theory and Practice.
- Cattaneo, M. D., Titiunik, R., and Yu, R. R. (2025). rd2d: Causal Inference in Boundary Discontinuity Designs.

See Also

`rd2d.distance`, `rd2d`, `summary.rdbw2d.distance`, `print.rdbw2d.distance`

Examples

```
set.seed(123)
n <- 800

# Generate running variables x1 and x2
x1 <- rnorm(n)
x2 <- rnorm(n)

# Define treatment assignment: treated if x1 >= 0
assignment <- as.numeric(x1 >= 0)

# Generate outcome variable Y with some treatment effect
Y <- 3 + 2 * x1 + 1.5 * x2 + 1.5 * assignment + rnorm(n, sd = 0.5)

# Define evaluation points (e.g., at the origin and another point)
eval <- data.frame(x.1 = c(0, 0), x.2 = c(0, 1))

# Compute Euclidean distances to evaluation points
distance.a <- sqrt((x1 - eval$x.1[1])^2 + (x2 - eval$x.2[1])^2)
distance.b <- sqrt((x1 - eval$x.1[2])^2 + (x2 - eval$x.2[2])^2)

# Combine distances into a matrix
distance <- as.data.frame(cbind(distance.a, distance.b))

# Assign positive distances for treatment group, negative for control
assignment_expanded <- matrix(rep(2 * assignment - 1, times = ncol(distance)),
                              nrow = nrow(distance), ncol = ncol(distance))
distance <- distance * assignment_expanded
```

```
# Run the rdbw2d.distance function
bws <- rdbw2d.distance(Y, distance = distance, b = eval, masspoints = "off", bwcheck = 10)

# View the estimation results
print(bws)
summary(bws)

# Fuzzy distance-based bandwidths
fuzzy <- as.numeric(runif(n) < ifelse(assignment == 1, 0.8, 0.2))
Y.fuzzy <- 3 + 2 * x1 + 1.5 * x2 + 1.5 * fuzzy + rnorm(n, sd = 0.5)
bws.fuzzy <- rdbw2d.distance(Y.fuzzy, distance = distance, b = eval, fuzzy = fuzzy,
                         bwparam = "main", masspoints = "off",
                         bwcheck = 10)
print(bws.fuzzy)
```

`summary.rd2d` *Summary Method for 2D Local Polynomial RD Estimation*

Description

Summarizes estimation and bandwidth results from a 2D local polynomial regression discontinuity (RD) design, as produced by `rd2d`.

Usage

```
## S3 method for class 'rd2d'
summary(object, ...)
```

Arguments

`object`	An object of class `rd2d`, typically returned by `rd2d`.
`...`	Optional named arguments. Unsupported option names produce an error. Supported options include: • cbands: Optional character vector requesting uniform confidence bands for displayed outputs. Options are `"main"`, `"main.0"`, `"main.1"`, `"itt"`, `"itt.0"`, `"itt.1"`, `"fs"`, `"fs.0"`, and `"fs.1"`, subject to the sharp/fuzzy design and covariance matrices stored by `rd2d`. • `repp`: Positive integer. Number of Gaussian simulation repetitions used for uniform confidence band and LBATE critical values. Default is `1000`. • `WBATE`: Optional numeric weights for a weighted boundary average treatment effect row. The weights must match the full set of evaluation points for the selected output. The selected output must have a matching covariance matrix stored by `rd2d`. • `LBATE`: Logical. If `TRUE`, prints a largest boundary average treatment effect row. The selected output must have a matching covariance matrix stored by `rd2d`.

- `subset`: Integer vector of indices of evaluation points to display. Aggregates and uniform-band critical values are still computed using all evaluation points. Defaults to all evaluation points.
- `output`: Character. Use `"main"` to display sharp or fuzzy main results; `"main.0"` or `"main.1"` to display sharp control- or treatment-side main results; `"itt"`, `"itt.0"`, or `"itt.1"` to display fuzzy reduced-form or intention-to-treat results; `"fs"`, `"fs.0"`, or `"fs.1"` to display fuzzy first-stage treatment receipt/status results; and `"bw"` to display bandwidth information. If omitted, `summary()` prints all default and requested estimate tables and does not print `"bw"`. Use `output = "bw"` to print bandwidth information.
- `sep`: Integer vector of length three. Controls spacing in the output. `sep[1]` controls spacing for the columns of bandwidths, estimation, t-statistic, and p-value in the `"main"` table. `sep[2]` controls spacing for confidence intervals and confidence bands in the `"main"` table. `sep[3]` controls spacing for the columns in the `"bw"` table. Default is `c(7, 17, 8)`.

Value

Invisibly returns a list with displayed tables and uniform confidence bands requested through `cbands`. Each returned estimate table has the same columns as the corresponding `rd2d` output, with pointwise confidence intervals always retained and `cb.lower` and `cb.upper` added only when uniform bands are requested. Requested WBATE and LBATE rows are appended to the corresponding returned table.

Author(s)

Matias D. Cattaneo, Princeton University. `<matias.d.cattaneo@gmail.com>`
Rocio Titiunik, Princeton University. `<rocio.titiunik@gmail.com>`
Ruiqi Rae Yu, Princeton University. `<raeyuuuu@gmail.com>`

See Also

`rd2d` for estimation using 2D local polynomial RD design.

Supported methods: `print.rd2d`, `summary.rd2d`.

`summary.rd2d.distance` *Summary Method for 2D Local Polynomial RD Estimation (Distance-Based)*

Description

Summarizes estimation and bandwidth results from a 2D local polynomial regression discontinuity (RD) design using distance-based methods, as returned by `rd2d.distance`.

Usage

```
## S3 method for class 'rd2d.distance'
summary(object, ...)
```

Arguments

`object`	An object of class `rd2d.distance`, returned by `rd2d.distance`.
`...`	Optional arguments. Supported options include: • `cbands`: Character vector. Use `cbands = "main"` to display uniform confidence bands for the main distance-based estimates. Other stored outputs can be requested when their covariance matrices were stored through `params.cov`. The default displays pointwise confidence intervals. • `repp`: Positive integer. Number of Gaussian simulation repetitions used for uniform confidence band and LBATE critical values. Default is `1000`. • `WBATE`: Optional numeric weights for a weighted boundary average treatment effect row. The weights must match the full set of evaluation points and are normalized internally. The fitted object must contain the covariance matrix stored by `rd2d.distance(cbands = TRUE)` or requested through `params.cov`. • `LBATE`: Logical. If `TRUE`, prints a largest boundary average treatment effect row. The fitted object must contain the covariance matrix stored by `rd2d.distance(cbands = TRUE)` or requested through `params.cov`. • `subset`: Integer vector of indices of evaluation points to display. Defaults to all evaluation points. • `output`: Character vector. Use `"main"` to display treatment effect estimates or `"bw"` to display bandwidth information. In sharp designs, `"main.0"` and `"main.1"` display side-specific estimates. In fuzzy designs, `"itt"` and `"fs"` display reduced-form and first-stage estimates, and `"itt.0"`, `"itt.1"`, `"fs.0"`, and `"fs.1"` display requested side-specific companion estimates. • `sep`: Integer vector of length three. Controls spacing in the output. `sep[1]` controls spacing for the columns of boundary points, estimation, t-value, and p-value in the `"main"` table. `sep[2]` controls spacing for confidence intervals and confidence bands in the `"main"` table. `sep[3]` controls spacing for the columns in the `"bw"` table. Default is `c(7, 17, 8)`.

Value

Invisibly returns an object of class `"summary.rd2d.distance"`, a list with elements:

- `tables`: named list of returned summary tables. Estimate tables retain pointwise confidence intervals and add `cb.lower` and `cb.upper` only when uniform bands are requested.
- `cbands`: named list of confidence-band endpoints for outputs requested through `cbands`.
- `outputs`: character vector of summarized outputs.
- `call`: matched summary call.

Requested WBATE and LBATE rows are appended to the returned estimation table(s), except for `output = "bw"`. The function is also called for its side effect of printing a formatted summary.

Author(s)

Matias D. Cattaneo, Princeton University. <`matias.d.cattaneo@gmail.com`>
Rocio Titiunik, Princeton University. <`rocio.titiunik@gmail.com`>
Ruiqi Rae Yu, Princeton University. <`raeyuuuu@gmail.com`>

See Also

`rd2d.distance` for estimation using distance-based 2D local polynomial RD design.

Supported methods: `print.rd2d.distance`, `summary.rd2d.distance`.

`summary.rdbw2d` *Summary Method for Bandwidth Selection for 2D Local Polynomial RD Design*

Description

Summary method for objects of class `rdbw2d`, displaying bandwidth selection results for 2D local polynomial regression discontinuity designs.

Usage

```
## S3 method for class 'rdbw2d'
summary(object, ...)
```

Arguments

`object`	An object of class `rdbw2d`, typically returned by `rdbw2d`.
`...`	Optional arguments. Supported options include: • `subset`: Integer vector of indices of evaluation points to display. Defaults to all evaluation points. • sep: Integer. Controls spacing in the output. Default is `8`.

Value

No return value. Called for its side effects of printing a formatted summary of `rdbw2d` results.

Author(s)

Matias D. Cattaneo, Princeton University. <`matias.d.cattaneo@gmail.com`>
Rocio Titiunik, Princeton University. <`rocio.titiunik@gmail.com`>
Ruiqi Rae Yu, Princeton University. <`raeyuuuu@gmail.com`>

See Also

`rdbw2d` for bandwidth selection in 2D local polynomial RD design.

Supported methods: `print.rdbw2d`, `summary.rdbw2d`.

`summary.rdbw2d.distance`

Summary Method for Bandwidth Selection in 2D Local Polynomial RD Design (Distance-Based)

Description

Summarizes bandwidth selection results from a 2D local polynomial regression discontinuity (RD) design using distance-based methods, as returned by `rdbw2d.distance`.

Usage

```
## S3 method for class 'rdbw2d.distance'
summary(object, ...)
```

Arguments

`object`	An object of class `rdbw2d.distance`, returned by `rdbw2d.distance`.
`...`	Optional arguments. Supported options include: • `subset`: Integer vector of indices of evaluation points to display. Defaults to all evaluation points. • `sep`: Integer vector of length two. Controls spacing in the output. `sep[1]` controls spacing for the columns of evaluation points in the table. `sep[2]` controls spacing for the columns of bandwidths in the table. Default is `c(8, 14)`.

Value

No return value. This function is called for its side effects: it prints a formatted summary of `rdbw2d.distance` results.

Author(s)

Matias D. Cattaneo, Princeton University. <`matias.d.cattaneo@gmail.com`>
Rocio Titiunik, Princeton University. <`rocio.titiunik@gmail.com`>
Ruiqi Rae Yu, Princeton University. <`raeyuuuu@gmail.com`>

See Also

`rdbw2d.distance` for bandwidth selection using 2D local polynomial RD design with distance-based methods.

Supported methods: `print.rdbw2d.distance`, `summary.rdbw2d.distance`.

`vcov.rd2d` *Extract rd2d Covariance Matrices*

Description

Extract rd2d Covariance Matrices

Usage

```
## S3 method for class 'rd2d'
vcov(object, output = "main", ...)

## S3 method for class 'rd2d.distance'
vcov(object, output = "main", ...)
```

Arguments

`object`	An object returned by `rd2d` or `rd2d.distance`.
`output`	Covariance output to extract. Default is `"main"`.
`...`	Additional arguments ignored.

Value

A covariance matrix.

Index