Package ‘rdbounds’

June 12, 2019

Title Manipulation Robust Regression Discontinuity Bounds Estimation

Version 1.01

Authors Francois Gerard <fgerard@columbia.edu>, Leonard Goff <leonard.goff@columbia.edu>, Miikka Rokkanen <miikka.rokkanen@gmail.com>, Christoph Rothe <rothe@vwl.uni-mannheim.de>

Maintainer Leonard Goff <leonard.goff@columbia.edu>

Description This package implements the estimation procedure in Gerard, Rokkanen, and Rothe (2018) to estimate bounds on treatment effects under potential manipulation of the running varible. This is a preliminary version and is still undergoing testing. We appreciate hearing any comments or issues noted.

URL http://www.nber.org/papers/w22892

Imports data.table,
formattable,
parallel

License GPL-3

Encoding UTF-8

RoxygenNote 6.1.1

R topics documented:

`rdbounds` *Manipulation Robust Regression Discontinuity Bounds Estimation*

Description

This function implements the estimation procedure in Gerard, Rokkanen, and Rothe (2018) to estimate bounds on treatment effects under potential manipulation of the running varible. Returns an rdbounds object, which can then be passed to rdbounds_summary and rdbounds_export. Note on refinements: "Refinement A" calculates bounds based on the assumption that always-assigned units are at least as likely to receive treatment than potentially-assigned units (Theorem 3 in paper). "Refinement B" calculates bounds based on the assumption that always-assigned units $always$ receive treatment (Theorem 4 in paper).

Usage

```
rdbounds(y, x, covs = NULL, treatment = NULL, c = 0,
  discrete_x = FALSE, discrete_y = FALSE, bwsx, bwy = NULL,
  kernel = "triangular", orders = array(1, dim = c(2)),
  evaluation_ys = NULL, ymin = NULL, ymax = NULL, type = "ate",
  percentiles = NULL, num_tau_pairs = 50, refinement_A = FALSE,
  refinement_B = FALSE, right_effects = FALSE, yextremes = NULL,
  num_lambdas = 50, num_bootstraps = c(100, 20), Kn = NULL,
  alpha = 0.05, potential_taus = NULL, parallelize = TRUE,
  progressFile = NULL, warningsFile = NULL, kernel_y = NULL,
  bwsxcov = NULL, bwycov = NULL, CDFinputs = list(original = NULL,
  CIsetup = NULL))
```

Arguments

y
specifies the outcome/dependent variable. Required.

x
specifies the running variable that determines eligibilty for treatment. Required.

covs
specifies covariates to implement the covariate-based refinement. Expected as a single factor variable.

treatment
specifies the treatment status variable if implementing a Fuzzy RDD. Defaults to computation of Sharp RDD results only.

c
specifies the threshold for assignment to treatment (assigned iff x >= c). Defaults to 0.

discrete_x
Boolean. If TRUE, treat each value of x as a mass-point for density estimation. Defaults to FALSE.

discrete_y
Boolean. If TRUE, treat each value of y as a mass-point for density estimation. Defaults to FALSE.

bwsx
is a vector of bandwidths in x, respectively for 1) estimation of the discontinuity in the density of x at the cutoff; and 2)local polynomial estimation of conditional means. Expects either a single bandwidth to be used for both or a vector of two. Required.

bwy
is a bandwidth for density estimation of y, implemented if discrete_y=FALSE. Required if discrete_y=FALSE.

kernel
specificies a kernel function to be used throughout estimation for x. Choices are triangular, rectangular, gaussian and epanechnikov. Defaults to triangular.

orders
specifies the order of polynomial regression, for: 1) estimation of the discontinuity in the density at the cutoff (τ in paper), and 2) local polynomial regressions. Expects either a single integer to be used for both or a vector of two values. Defaults to 1 (local linear regression) for all. Estimation of τ can only be implemented up to quadratic order if discrete_X=FALSE.

`evaluation_ys` an explicit vector of y-values to evaluate CDF's at (and PDF's if `discrete_y = FALSE`). If `evaluation_ys` is not set, the set of unique values of y in the sample will be used. Caution is required if `discrete_y=TRUE`, because computation will assume a probability mass function can be estimated from differences in estimated CDF's at subsequent values of `evaluation_ys`. This can bias FRD estimates if `evaluation_ys` does not contain all values in the support of y.

`ymin` left/lower bound on `y` at which to implement a boundary kernel correction if `discrete_y=FALSE` and `y` is a variable with bounded support (e.g. after censoring). Defaults to `NULL`, meaning no boundary kernel correction is implemented on the left side of the support of `y`.

`ymax` right/upper bound on `y` at which to implement a boundary kernel correction if `discrete_y=FALSE` and `y` is a variable with bounded support (e.g. after censoring). Defaults to `NULL`, meaning no boundary kernel correction is implemented on the left side of the support of `y`.

`type` "ate" for average treatment effects (default) or "qte" for quantile treatment effects at the percentiles given by parameter `percentiles`. Defaults to `ate`.

`percentiles` vector of percentiles at which to asses quantile treatment effects. Defaults to median (.5). User may add -1 as a percentile, in order to estimate average treatment effects along with QTE's. For example, `percentiles=c(-1,.3,.5)` will compute ATEs as well as the 30 percent and 50 percent QTEs

`num_tau_pairs` integer number of points to search over in the set of possible values for (τ_0, τ_1) in notation of paper, for fuzzy RD estimation. Defaults to 50. If set to 1, the single tau is set to the "rightmost" (t=1) extreme of the set T, such that user can enforce the assumption that always-assigned units always receive treatment (see below), if this is consistent with data.

`refinement_A` Boolean. If `TRUE`, additionally calculate refined bounds with the restriction that always assigned units are at least as likely to be treated as potentially assigned units (i.e. $\tau_1 \geq \tau$; see Corollary 1 in paper) Defaults to FALSE.

`refinement_B` Boolean. If `TRUE`, additionally calculate refined bounds for `right_effects` with the restriction that always assigned units on the right side of the cutoff are always treated (i.e. $\tau_0 = 0$; see Corollary 2 in paper) Defaults to FALSE.

`right_effects` boolean. If set to `TRUE`, additionally estimate causal effects for units just to the right of the cutoff. Defaults to `FALSE`.

`yextremes` extreme values Y_L and Y_U to assume if `right_effects=TRUE`, e.g. `yextremes=c(0,100)`. Defaults to the sample range of `y`.

`num_lambdas` integer number of points to search over for the causal effect of units just to the right of the cutoff (lambda in paper). Defaults to 50.

`num_bootstraps` A vector of the number of bootstrap resamples desired, where the first component is the number of bootstrap samples for estimating confidence intervals, and the second is the number of samples for diagnostic testing of the estimated discontinuity in the density at the cutoff. If a scalar is given, the same number is used for both. Defaults to `num_bootstraps = c(100,20)`. To avoid bootstrap testing altogether, set `num_bootstraps=NULL` or `num_bootstraps=c(0,0)`.

`Kn` a hardcoded constant for κ_n (see Section 5.2 on inference in paper). Defaults to $log(n)^{1/2}$, where n is the number of observations.

`alpha` sets the level for confidence intervals. Defaults to alpha=.05 for 95 percent confidence intervals.

`potential_taus` vector of different values of τ to use for the confidence intervals estimating the potential impact of manipulation, e.g. `potential_taus=c(.025, .05, .1, .2)`.

`parallelize` indicates whether to parallelize bootstrap computations across the available number of cores on machine, minus one. Defaults to `TRUE`.

`progressFile` a file to output progress to (useful if `parallelize=TRUE` and the individual cores can't write to screen). File will be appended to.

`warningsFile` a file to output full warning messages to from bootstrap estimation if `parallelize=TRUE`). File will be appended to.

`kernel_y` allows a separate kernel for density estimation of `y`. Same choices as kernel for `x`. Defaults to kernel specified for use with `x`.

`bwsxcov` an optional separate `bwsx` to use for quantities that are computed on a subsample conditioned on a value of `covs` (e.g. covariate-conditional CDFs).

`bwycov` an optional separate `bwy` to use for quantities that are computed on a subsample conditioned on a value of `covs` (e.g. covariate-conditional CDFs).

`CDFinputs` optional, the `rdbounds$CDFinputs` object from a previous run of `rdbounds` on the same dataset. This can be used to speed up processing by allowing CDF and PDF estimation to be skipped on a second run.

References

Francois Gerard, Miikka Rokkanen, and Christoph Rothe (2016)."Bounds on Treatment Effects in Regression Discontinuity Designs under Manipulation of the Running Variable, with an Application to Unemployment Insurance in Brazil". NBER Working Paper 22892.

Examples

```
df<-rdbounds_sampledata(50000, covs=TRUE)
rdbounds_est<-rdbounds(y=df$y,x=df$x, covs=as.factor(df$cov), treatment=df$treatment, c=0,
                       discrete_x=FALSE, discrete_y=FALSE,
                       bwsx=c(.2,.5), bwy = .1, kernel="epanechnikov", orders=1,
                       evaluation_ys = seq(from = 0, to=23, by=.2),
                       refinement_A=TRUE, refinement_B=TRUE,
                       right_effects=TRUE, yextremes = c(0,23),
                       num_bootstraps=0)
rdbounds_summary(rdbounds_est, title_prefix="Sample Data Results")
```

`rdbounds_export` *Export Results from Manipulation Robust RD Estimation*

Description

This function exports tables from manipulation robust RD estimation

Usage

```
rdbounds_export(rdbounds, file_name = NULL, view_it = FALSE)
```

Arguments

`rdbounds` An rdbounds object resulting from `rdbounds`. Required.

`file_name` base filename to output tables to. Expects a string of the form "path/filename", where filename has no extension and will be the root filename for a series of different files containing different tables. If omitted no files will be produced.

`view_it` Boolean. View main results table in Rstudio viewer. Defaults to `FALSE`.

Examples

```
df<-rdbounds_sampledata(50000, covs=TRUE)
rdbounds_est<-rdbounds(y=df$y,x=df$x, covs=as.factor(df$cov), treatment=df$treatment, c=0,
                       discrete_x=FALSE, discrete_y=FALSE,
                       bwsx=c(.2,.5), bwy = .1, kernel="epanechnikov", orders=1,
                       evaluation_ys = seq(from = 0, to=23, by=.2),
                       refinement_A=TRUE, refinement_B=TRUE,
                       right_effects=TRUE, yextremes = c(0,23),
                       num_bootstraps=0)
rdbounds_summary(rdbounds_est, title_prefix="Sample Data Results")
```

rdbounds_sampledata *Generate a simulated dataset for testing estimation*

Description

This function generates a simulated dataset with which to test rdbounds. The x-values of potentially-assigned units (95 Specifically:

$$y = (x+10)/2*treatment*(alwaysassigned = 0)+5*treatment*(alwaysassigned = 1)+normal(0, 1))$$

and y is censored at 0 and 23.

Usage

```
rdbounds_sampledata(sample_size = 50000, covs = FALSE)
```

Arguments

sample_size	Sample size for the dataset.
covs	If set to TRUE, generates a sample in which half of the units have one of two covariate values, where the proportion of always-assigned units is slightly different for each.

Examples

```
df<-rdbounds_sampledata(50000);
```

rdbounds_summary *Summarize Results from Manipulation Robust RD Estimation*

Description

This function reports main estimands from rdbounds, as a formatted table, and optionally as text output.

Usage

```
rdbounds_summary(rdbounds, title_prefix = "", text = TRUE)
```

Arguments

`rdbounds` an rdbounds object resulting from the function rdbounds(). Required.

`title_prefix` Optional prefix before "Average Treatment EFfects" or "Quantile Treatment Effects" in table.

`text` if set to `TRUE`, display results as text as well as formatted table. Defaults to `TRUE`.

Examples

```
df<-rdbounds_sampledata(50000, covs=TRUE)
rdbounds_est<-rdbounds(y=df$y,x=df$x, covs=as.factor(df$cov), treatment=df$treatment, c=0,
                       discrete_x=FALSE, discrete_y=FALSE,
                       bwsx=c(.2,.5), bwy = .1, kernel="epanechnikov", orders=1,
                       evaluation_ys = seq(from = 0, to=23, by=.2),
                       refinement_A=TRUE, refinement_B=TRUE,
                       right_effects=TRUE, yextremes = c(0,23),
                       num_bootstraps=0)
rdbounds_summary(rdbounds_est, title_prefix="Sample Data Results")
```

Index