Inference in Regression Discontinuity Designs under Local Randomization*

Matias D. Cattaneo† Rocio Titiunik‡ Gonzalo Vazquez-Bare§

August 28, 2016

Abstract

We illustrate the implementation of the R functions `rdrandinf`, `rdwinselect`, `rdsensitivity` and `rdrbounds`, used to conduct finite-sample inference in regression discontinuity designs under local randomization. These R functions have the same syntax and capabilities of the Stata commands described in Cattaneo, Titiunik, and Vazquez-Bare [2016b].

Keywords: regression discontinuity designs, quasi-experimental techniques, causal inference, randomization inference, finite-sample methods, Fisher's exact p-values, Neyman's repeated sampling approach.

*We thank Xinwei Ma for helpful comments. Financial support from the National Science Foundation (SES 1357561) is gratefully acknowledged.

†Department of Economics and Department of Statistics, University of Michigan.

‡Department of Political Science, University of Michigan.

§Department of Economics, University of Michigan.

Contents

1 Introduction

This article describes the R functions `rdrandinf`, `rdwinselect`, `rdsensitivity` and `rdrbounds`, used to conduct finite-sample inference in regression discontinuity designs under a local randomization assumption. For brevity, we focus exclusively on software implementation issues. Extensive discussion and details on methodological and practical aspects can be found in Cattaneo et al. [2015], Cattaneo et al. [2016a] and Cattaneo et al. [2016b].

2 `rdrandinf()` syntax

This section describes the syntax of the function `rdrandinf()`, which calculates randomization p-values on a specific window under different randomization mechanisms and potential outcomes models.

2.1 Syntax

```
rdrandinf(Y, R, cutoff = 0, wl = "",wr = "", reps = 1000, statistic = "ttest",
          p = 0, nulltau = 0, evall = "", evalr = "", kernel = "uniform",
          ci, interfci = "", seed = "", fuzzy = "", d = "", dscale = "",
          bernoulli, quietly = FALSE, covariates, obsmin = "", obsstep = "",
          wmin = "", wstep = "", nwindows = 10, rdwstat = "ttest", approx = FALSE,
          rdwreps = 1000, level = .15, plot = FALSE)
```

Arguments

- `Y`: a vector containing the values of the outcome variable.
- `R`: a vector containing the values of the running variable.
- `cutoff = 0`: the RD cutoff (default is 0).
- `wl = ""`: the left limit of the window. The default takes the minimum of the running variable.
- `wr = ""`: the right limit of the window. The default takes the maximum of the running variable.
- `reps = 1000`: the number of replications (default is 1000).
- `statistic = "ttest"`: the statistic to be used in randomization inference. Options are `"ttest"` (difference in means), `"ksmirnov"` (Kolmogorov-Smirnov statistic), `"ranksum"` (Wilcoxon-Mann-Whitney standardized statistic) or `"all"`, which gives all three statistics. Default option is `"ttest"`.
- `p = 0`: the order of the polynomial for outcome transformation model (default is 0).

- `nulltau = 0`: the value of the treatment effect under the null hypothesis (default is 0).
- `evall = ""`: the point at the left of the cutoff at which to evaluate the transformed outcome is evaluated. Default is the cutoff value.
- `evalr = ""`: specifies the point at the right of the cutoff at which the transformed outcome is evaluated. Default is the cutoff value.
- `kernel = "uniform"`: specifies the type of kernel to use as weighting scheme. Allowed kernel types are `"uniform"` (uniform kernel), `"triangular"` (triangular kernel) and `"epan"` (Epanechnikov kernel). Default is `"uniform"`.
- `ci`: calculates a confidence interval for the treatment effect by test inversion. `ci` can be specified as a scalar or a vector, where the first element indicates the level of the confidence interval and the remaining elements, if specified, indicate the grid of treatment effects to be evaluated. This option uses `rdsensitivity` to calculate the confidence interval. See corresponding help for details.
- `interfci = ""`: the level for Rosenbaum's confidence interval under arbitrary interference between units.
- `seed = ""`: the seed to be used for the randomization test.
- `fuzzy = ""`: indicates that the RD design is fuzzy. `fuzzy` can be specified as a vector containing the values of the endogenous treatment variable, or as a list where the first element is the vector of endogenous treatment values and the second element is a string containing the name of the statistic to be used. Allowed statistics are `"ar"` (Anderson-Rubin statistic) and `"tsls"` (2SLS statistic). Default statistic is `"ar"`. The `"tsls"` statistic relies on large-sample approximation.
- `d = ""`: the effect size for asymptotic power calculation. Default is 0.5 * standard deviation of outcome variable for the control group.
- `dscale = ""`: the fraction of the standard deviation of the outcome variable for the control group used as alternative hypothesis for asymptotic power calculation. Default is 0.5.
- `bernoulli`: the probabilities of treatment for each unit when assignment mechanism is a Bernoulli trial. This option should be specified as a vector of length equal to the length of the outcome and running variables.
- `quietly = FALSE`: suppresses the output table.

NOTE: when the window is not specified, `rdrandinf` can choose the window using `rdwinselect`. The following options are available:

- `covariates`: the covariates used by `rdwinselect` to choose the window when `wl` and `wr` are not specified. This should be a matrix of size $n \times k$ where n is the total sample size and k is the number of covariates.

- `obsmin = ""`: the minimum number of observations above and below the cutoff in the smallest window employed by the companion command `rdwinselect`. Default is 10.

- `obsstep = ""`: the minimum number of observations to be added on each side of the cutoff for the sequence of nested windows constructed by the companion command `rdwinselect`. Default is 2.

- `wmin = ""`: the smallest window to be used (if `minobs` is not specified) by the companion command `rdwinselect`. Specifying both `wmin` and `obsmin` returns an error.

- `wstep = ""`: the increment in window length (if `obsstep` is not specified) by the companion command `rdwinselect`. Specifying both `obsstep` and `wstep` returns an error.

- `nwindows = 10`: the number of windows to be used by the companion command `rdwinselect`. Default is 10.

- `rdwstat = "ttest"`: the statistic to be used by the companion command `rdwinselect` (see corresponding help for options). Default option is `"ttest"`.

- `approx = FALSE`: forces the companion command `rdwinselect` to conduct the covariate balance tests using a large-sample approximation instead of finite-sample exact randomization inference methods.

- `rdwreps = 1000`: the number of replications to be used by the companion command `rdwinselect`. Default is 1000.

- `level = .15`: the minimum accepted value of the p-value from the covariate balance tests to be used by the companion command `rdwinselect`. Default is .15.

- `plot = FALSE`: draws a scatter plot of the minimum p-value from the covariate balance test against window length implemented by the companion command `rdwinselect`.

Value

If the argument `print.screen` is set to `FALSE`, the following list of values will be returned.

- `$sumstats`: summary statistics displayed in the middle panel.

- `$obs.stat`: observed statistic(s).

- `$p.value`: randomization p-value(s).

- `$asy.pvalue`: asymptotic p-value(s).

- `$window`: chosen window.
- `$ci`: confidence interval (only if `ci` option is specified).
- `$interf.ci`: confidence interval under interference (only if `interfci` option is specified).

3 `rdwinselect()` syntax

This section describes the syntax of the command **`rdwinselect`**. This command finds the largest window in which a set of covariates is found to be balanced between treated and control groups.

3.1 Syntax

```
rdwinselect(R, X, cutoff = 0, obsmin = "", obsstep = "", wmin = "", wstep = "",
          nwindows = 10, statistic = "ttest", approx = FALSE,  p = 0,
          evalat = "cutoff", kernel = "uniform", reps = 1000, seed = "",
          level = .15, plot = FALSE, quietly = FALSE)
```

- `R`: a vector containing the values of the running variable.
- `X`: the matrix of covariates to be used in the balancing tests. The matrix is optional but the recommended window is only provided when at least one covariate is specified. This should be a matrix of size $n \times k$ where n is the total sample size and k is the number of covariates.
- `cutoff = 0`: the RD cutoff (default is 0).
- `obsmin = ""`: the minimum number of observations above and below the cutoff in the smallest window. Default is 10.
- `obsstep = ""`: the minimum number of observations to be added on each side of the cutoff for the sequence of nested windows. Default is 2.
- `wmin = ""`: the smallest window to be used (if `minobs` is not specified). Specifying both `wmin` and `obsmin` returns an error.
- `wstep = ""`: the increment in window length (if `obsstep` is not specified). Specifying both `obsstep` and `wstep` returns an error.
- `nwindows = 10`: the number of windows to be used. Default is 10.
- `statistic = "ttest"`: the statistic to be used in the balance tests. Allowed options are `"ttest"` (difference in means statistic), `"ksmirnov"` (Kolmogorov-Smirnov statistic), `"ranksum"` (Wilcoxon-Mann-Whitney standardized statistic) and `"hotelling"` (Hotelling's T-squared statistic). Default option is `"ttest"`.

- `approx = FALSE`: forces the command to conduct the covariate balance tests using a large-sample approximation instead of finite-sample exact randomization inference methods.
- `p = 0`: the order of the polynomial for outcome adjustment model (for covariates). Default is 0.
- `evalat = "cutoff"`: specifies the point at which the adjusted variable is evaluated. Allowed options are `"cutoff"` and `"means"`. Default is `"cutoff"`.
- `kernel = "uniform"`: specifies the type of kernel to use as weighting scheme. Allowed kernel types are `"uniform"` (uniform kernel), `"triangular"` (triangular kernel) and `"epan"` (Epanechnikov kernel). Default is `"uniform"`.
- `reps = 1000`: the number of replications to be used by the companion command `rdwinselect`. Default is 1000.
- `seed = ""`: the seed to be used for the randomization tests.
- `level = .15`: the minimum accepted value of the p-value from the covariate balance tests. Default is .15.
- `plot = FALSE`: draws a scatter plot of the minimum p-value from the covariate balance test against window length.

Value

If the argument `print.screen` is set to `FALSE`, the following list of values will be returned.

- `$window`: recommended window (`NA` if covariates are not specified).
- `$results`: table including window lengths, minimum p-value in each window, corresponding number of the variable with minimum p-value (i.e. column of covariate matrix), Binomial test p-value and sample sizes to the left and right of the cutoff in each window.
- `$summary`: summary statistics.

4 rdsensitivity() syntax

This section describes the syntax of the command `rdsensitivity`, which is used to analyze the sensitivity of randomization p-values and confidence intervals to different window lengths.

4.1 Syntax

```
rdsensitivity(Y, R, cutoff = 0, wlist, tlist, ci, statistic = "ttest",
          nodraw = FALSE, p = 0, evalat = "cutoff", kernel = "uniform",
          reps = 1000, seed = "", fuzzy = "", quietly = FALSE)
```

- `Y`: a vector containing the values of the outcome variable.
- `R`: a vector containing the values of the running variable.
- `cutoff = 0`: the RD cutoff (default is 0).
- `wlist`: the list of window lengths to be evaluated. By default the program constructs 10 windows around the cutoff, the first one including 10 treated and control observations and adding 5 observations to each group in subsequent windows.
- `tlist`: the list of values of the treatment effect under the null to be evaluated. By default the program employs ten evenly spaced points within the asymptotic confidence interval for a constant treatment effect in the smallest window to be used.
- `statistic = "ttest"`: the statistic to be used in the balance tests. Allowed options are `"ttest"` (difference in means statistic), `"ksmirnov"` (Kolmogorov-Smirnov statistic) and `"ranksum"` (Wilcoxon-Mann-Whitney standardized statistic). Default option is `"ttest"`.
- `ci`: returns the confidence interval corresponding to the indicated window length. `ci` has to be a scalar or a two-dimensional vector, where the first value needs to be one of the values in `wlist`. The second value, if specified, indicates the level of the confidence interval. Default level is .05 (95% level CI).
- `nodraw = FALSE`: suppresses contour plot.
- `p = 0`: the order of the polynomial for outcome adjustment model. Default is 0.
- `evalat = "cutoff"`: specifies the point at which the adjusted variable is evaluated. Allowed options are `"cutoff"` and `"means"`. Default is `"cutoff"`.
- `kernel = "uniform"`: specifies the type of kernel to use as weighting scheme. Allowed kernel types are `"uniform"` (uniform kernel), `"triangular"` (triangular kernel) and `"epan"` (Epanechnikov kernel). Default is `"uniform"`.
- `reps = 1000`: the number of replications to be used by the companion command `rdwinselect`. Default is 1000.
- `seed = ""`: the seed to be used for the randomization tests.
- `fuzzy = ""`: indicates that the RD design is fuzzy. `fuzzy` can be specified as a vector containing the values of the endogenous treatment variable, or as a list where the first element is the vector of endogenous treatment values and the second element is a string containing the name of the statistic to be used. Allowed statistics are `"ar"` (Anderson-Rubin statistic) and `"tsls"` (2SLS statistic). Default statistic is `"ar"`. The `"tsls"` statistic relies on large-sample approximation.

- `quietly = FALSE`: suppresses the output table.

Value

If the argument `print.screen` is set to `FALSE`, the following list of values will be returned.

- `$tlist`: treatment effects grid.
- `$wlist`: window grid.
- `$results`: table with corresponding p-values for each window and treatment effect pair.
- `$ci`: confidence interval (if `ci` is specified).

5 rdrbounds() syntax

This section describes the syntax of the command **rdrbounds**, which calculates lower and upper bounds for the randomization p-value under different degrees of departure from a local randomized experiment, as suggested by Rosenbaum [2002].

5.1 Syntax

```
rdrbounds(Y, R, cutoff = 0, prob, gamma, expgamma, wlist,
          bound = "both", fmpval = FALSE, statistic = "ttest",
          p = 0, evalat = "cutoff",  kernel = "uniform",
          nulltau = 0, fuzzy = "", reps = 1000, seed = "")
```

- `Y`: a vector containing the values of the outcome variable.
- `R`: a vector containing the values of the running variable.
- `cutoff = 0`: the RD cutoff (default is 0).
- `prob`: the probabilities of treatment for each unit when assignment mechanism is a Bernoulli trial. This option should be specified as a vector of length equal to the length of the outcome and running variables.
- `gamma`: the list of values of gamma to be evaluated.
- `expgamma`: the list of values of exp(gamma) to be evaluated. Default is `expgamma = c(1.5,2,2.5,3)`.
- `wlist`: the list of window lengths to be evaluated. By default the program constructs 10 windows around the cutoff, the first one including 10 treated and control observations and adding 5 observations to each group in subsequent windows.

- `fmpval = FALSE`: reports the p-value under fixed margins randomization, in addition to the p-value under Bernoulli trials.
- `statistic = "ttest"`: the statistic to be used in the balance tests. Allowed options are `"ttest"` (difference in means statistic), `"ksmirnov"` (Kolmogorov-Smirnov statistic) and `"ranksum"` (Wilcoxon-Mann-Whitney standardized statistic). Default option is `"ranksum"`.
- `p = 0`: the order of the polynomial for outcome adjustment model. Default is 0.
- `evalat = "cutoff"`: specifies the point at which the adjusted variable is evaluated. Allowed options are `"cutoff"` and `"means"`. Default is `"cutoff"`.
- `kernel = "uniform"`: specifies the type of kernel to use as weighting scheme. Allowed kernel types are `"uniform"` (uniform kernel), `"triangular"` (triangular kernel) and `"epan"` (Epanechnikov kernel). Default is `"uniform"`.
- `nulltau = 0`: the value of the treatment effect under the null hypothesis (default is 0).
- `fuzzy = ""`: indicates that the RD design is fuzzy. `fuzzy` can be specified as a vector containing the values of the endogenous treatment variable, or as a list where the first element is the vector of endogenous treatment values and the second element is a string containing the name of the statistic to be used. Allowed statistics are `"ar"` (Anderson-Rubin statistic) and `"tsls"` (2SLS statistic). Default statistic is `"ar"`. The `"tsls"` statistic relies on large-sample approximation.
- `reps = 1000`: the number of replications to be used by the companion command `rdwinselect`. Default is 1000.
- `seed = ""`: the seed to be used for the randomization tests.

Value

If the argument `print.screen` is set to `FALSE`, the following list of values will be returned.

- `$gamma`: list of gamma values.
- `$expgamma`: list of exp(gamma) values.
- `$wlist`: window grid.
- `$p.values`: p-values for each window (under gamma = 0).
- `$lower.bound`: list of lower bound p-values for each window and gamma pair.
- `$upper.bound`: list of upper bound p-values for each window and gamma pair.

6 Illustration of Methods

We illustrate how to implement the four functions described above using the dataset from Cattaneo et al. [2015]. This section replicates, as close as possible, section 7 of Cattaneo et al. [2016b].

The `rdrandinf` function requires the `sandwich` R package, so the user should ensure that the package is installed (although the functions will load it automatically).

```
install.packages("sandwich")
```

Additionally, the four functions plus the auxiliary file `rdlocrand_fun` need to be installed in the same directory. Once in the appropriate directory, the four functions can be loaded by adding the following lines to a script file:

```
source("rdwinselect.R")
source("rdrandinf.R")
source("rdsensitivity.R")
source("rdrbounds.R")
```

We start by loading the data:

```
> data = read.csv("rdlocrand_senate.csv")
> dim(data)
[1] 1390   14
> names(data)
 [1] "state"             "year"              "dopen"             "population"        "presdemvoteshlag1" "demmv"
[10] "demvoteshfor2"     "demwinprv1"        "demwinprv2"        "dmidterm"          "dpresdem"
>
> # Select predetermined covariates to be used for window selector
> X  =  cbind(data$presdemvoteshlag1, data$population/1000000, data$demvoteshlag1,
+             data$demvoteshlag2, data$demwinprv1, data$demwinprv2, data$dopen,
+             data$dmidterm, data$dpresdem)
>
> # Assign names to the covariates
> colnames(X) =  c("DemPres Vote", "Population", "DemSen Vote t-1",
+                  "DemSen Vote t-2", "DemSen Win t-1", "DemSen Win t-2",
+                  "Open", "Midterm", "DemPres")
>
> # Running variable and outcome variable
> R = data$demmv
> Y = data$demvoteshfor2
> D = as.numeric(R>=0)
```

The most basic syntax for `rdwinselect` is the following:

```
> tmp = rdwinselect(R,X)

Window selection for RD under local randomization
```

```
Number of obs   =         1390
Order of poly   =         0
Kernel type     =         uniform
Reps            =         1000
Testing method =          rdrandinf
Balance test    =         ttest

Cutoff c = 0              Left of c           Right of c
Number of obs             640                 750
1st percentile            7                   7
5th percentile            32                  37
10th percentile           64                  75
20th percentile           127                 149

Window length / 2        p-value      Var. name       Bin.test     Obs<c  Obs>=c

0.529                     0.183     DemSen Vote t-2     0.327        10     16
0.733                     0.258     Open                0.2          15     24
0.937                     0.145     Open                0.126        16     27
1.141                     0.038     Open                0.161        20     31
1.346                     0.227     Open                0.382        28     36
1.55                      0.101     Midterm             0.728        35     39
1.754                     0.075     Midterm             0.747        41     45
1.958                     0.033     Midterm             0.602        43     49
2.163                     0.057     Midterm             0.48         45     53
2.367                     0.114     Open                0.637        53     59

Recommended window is [-0.733;0.733] with 39 observations (15 below, 24 above).
```

Because in this particular application the cutoff is zero, which is the default value, the `cutoff` option can be omitted. For this reason, this and all the remaining examples will not specify this option. In practice, when the cutoff is not zero, the user can simply specify `cutoff = c`. Alternatively, it may be easier to simply redefine the running variable by recentering it at the cutoff. By default, `rdwinselect` uses the difference-in-means statistic to perform hypothesis tests—but this can be changed with the `statistic` option.

The output of `rdwinselect` is divided in three panels. The upper panel indicates the total sample size, the degree of the polynomial used by `rdrandinf`, the type of kernel used for the weighting scheme (`uniform`, `triangular` or `epan`), the number of replications in the permutation test (whenever this test is performed), the method used to perform the covariate balance tests (`approximate` or `rdrandinf`), the test statistic used (`test`, `ksmirnov` or `ranksum`).

The middle panel provides information on sample sizes. The first row gives the total number of observations to the left and to the right of the cutoff, and also the total sample size. The following four rows provide the same information but around small neighborhoods around the cutoffs defined by the first, fifth, tenth and twentieth percentile of the running variable.

Finally, the main panel gives the result of the two balance tests performed at each of the windows considered. The first column provides the window length of each window considered, divided by two. For example, a value of 0.529 in this column refers to the window $[\bar{r} - 0.529\,;\, \bar{r} + 0.529]$, where $\bar{r}$ is the cutoff (equal to zero in this case) and the window length is $\bar{r} + 0.529 - (\bar{r} - 0.529) = 1.058$. The second column, labeled "`p-value`", provides the minimum p-value of the difference-in-means test, and the name of the corresponding variable associated to this p-value is given in column 3, "`Var. name`". The p-value is obtained by either permutation testing or a t-test, depending on the option specified. The fourth column gives the p-value from a Binomial probability test of the hypothesis that the probability of treatment is 0.5 using the `binom.test` command. Columns 5 and 6 give the number of observations to the left and right of the cutoff inside each window. As indicated in the last line of the output, the largest recommended window (the largest window for which the second column is equal to or above 0.15) in this case is $[-0.733\,;\, 0.733]$ and contains 15 observations below the cutoff and 24 observations above.

By default, `rdwinselect` starts with a window that contains at least 10 observations at each side of the cutoff, and increases the length ensuring that at least two observations are added in each successive window. The user can choose these two values using the `obsmin` and `obsstep` options, respectively, or can define the windows in terms of their length instead of the number of observations. For instance, Cattaneo et al. [2015] start from the window $[-0.5\,;\, 0.5]$ and increase the width by 0.125 using 10,000 replications in the permutation test. To replicate their results, we can type:

```
> tmp = rdwinselect(R,X,wmin=.5,wstep=.125,reps=10000)

Window selection for RD under local randomization

Number of obs  =         1390
Order of poly  =         0
Kernel type    =         uniform
Reps           =         10000
Testing method =         rdrandinf
Balance test   =         ttest

Cutoff c = 0             Left of c           Right of c
Number of obs            640                 750
1st percentile           7                   7
5th percentile           32                  37
10th percentile          64                  75
20th percentile          127                 149

Window length / 2       p-value      Var. name       Bin.test     Obs<c  Obs>=c

0.5                      0.265     DemSen Vote t-2     0.23          9      16
0.625                    0.416     Open                0.377         13     19
0.75                     0.263     Open                0.2           15     24
```

```
0.875                    0.147     Open              0.211        16      25
1                        0.074     Open              0.135        17      28
1.125                    0.038     Open              0.119        19      31
1.25                     0.054     Open              0.105        21      34
1.375                    0.142     Midterm           0.539        30      36
1.5                      0.09      Midterm           0.64         34      39
1.625                    0.111     Midterm           0.734        37      41

Recommended window is [-0.75;0.75] with 39 observations (15 below, 24 above).
```

We see that the command selects the window $[-0.75; 0.75]$, as in Cattaneo et al. [2015]. However, it is important to note that these results can vary slightly because of the randomization process behind the selection procedure. Additionally, observe that the minimum p-value is not necessarily monotonic on the length of the window. The `plot` option allows the user to depict graphically how these values change for different lengths. We will set the number of windows to 80 to have more observations in the plot, and we will specify the `approx` option to speed up the calculations. By specifying this option, the command uses the large-sample approximation instead of randomization inference. It is useful for illustration purposes as it is much faster, but it can be misleading since the approximation may be poor when the sample is small. The output from `rdwinselect` with 80 windows is a long table and will be omitted. The resulting graph is shown in Figure SAF-1.

```
> tmp = rdwinselect(R,X,wmin=.5,wstep=.125,approx=TRUE,nwin=80,quietly=TRUE,plot=TRUE)
```

The figure shows that the p-values vary widely for very short windows, but the sequence stabilizes once the window length is large enough (around the value 3 in this case).

Once the window has been selected, randomization inference to test the sharp null hypothesis of no treatment effect can be performed using `rdrandinf`. For example, take the window $[-0.75\,;\,0.75]$, which is the one selected by Cattaneo et al. [2015] and replicated above. The basic syntax for `rdrandinf` is:

```
> tmp = rdrandinf(Y,R,wl=-.75,wr=.75)

Selected window = [-0.75;0.75]

Running randomization-based test...

Randomization-based test complete.


Number of obs =       1297
Order of poly =       0
Kernel type   =       uniform
Reps          =       1000
Window        =       set by user
H0:       tau =       0
Randomization =       fixed margins
```

```
Cutoff c = 0          Left of c       Right of c
Number of obs         595             702
Eff. number of obs    15              22
Mean of outcome       42.808          52.497
S.d. of outcome       7.042           7.742
Window                -0.75           0.75

                               Finite sample       Large sample

Statistic             T              P>|T|          P>|T|       Power vs d = 3.521

Diff. in means        9.689          0              0           0.3
```

Like `rdwinselect`, the output of `rdrandinf` is divided in three panels. The upper panel gives the total sample size, the order of the polynomial, the type of kernel used for the weighting scheme (`uniform`, `triangular` or `epan`), the number of replications in the randomization test, and whether the window was specified by the user by setting `wl` and `wr` or calculated using `rdwinselect` as will be illustrated shortly. The middle panel provides the number of observations at each side of the cutoff, sample size below and above the cutoff inside the specified window, some descriptive statistics for the outcome inside the window, and the selected window. Note that the first line in this panel displays the number of observations with non-missing values of the outcome and running variable, so the sample sizes shown can differ from the total sample size.

Finally, the main panel gives the results from the randomization test. The first column, labeled "Statistic", indicates the statistic used in the randomization test. The second column gives the observed value of the selected statistic and the third column shows its finite-sample p-value obtained from the randomization test. The fourth column gives the asymptotic p-value, that is, the p-value obtained from the corresponding asymptotic distribution of the chosen statistic. Finally the fifth column gives the asymptotic power against an alternative value that can be specified using the options `d()` or `dscale()`. The default is `dscale(.5)`, that is, an effect size equal to half the standard deviation of the outcome for the control group inside the window (the critical value for the power calculation is set to 1.96).

As mentioned above, `rdrandinf` uses the difference in means as the default statistic, but it can also use the Kolmogorov-Smirnov and the rank sum statistics. By adding `statistic(all)` as an option we can obtain the result for all three statistics. The output is:

```
> tmp = rdrandinf(Y,R,wl=-.75,wr=.75,statistic='all')

Selected window = [-0.75;0.75]

Running randomization-based test...

Randomization-based test complete.
```

```
Number of obs =       1297
Order of poly =       0
Kernel type   =       uniform
Reps          =       1000
Window        =       set by user
H0:       tau =       0
Randomization =       fixed margins

Cutoff c = 0          Left of c       Right of c
Number of obs         595             702
Eff. number of obs    15              22
Mean of outcome       42.808          52.497
S.d. of outcome       7.042           7.742
Window                -0.75           0.75

                                  Finite sample        Large sample

Statistic            T              P>|T|          P>|T|      Power vs d = 3.521

Diff. in means       9.689          0.001          0          0.3
Kolmogorov-Smirnov   0.552          0.005          0.005      NA
Rank sum z-stat      -3.217         0.001          0.001      0.209
```

We can see that the three statistics provide basically the same result in terms of inference; the randomization test rejects the sharp null of no treatment effect at one percent significance level in all three cases. Also note that the `rdrandinf` command does not provide the asymptotic power for the Kolmogorov-Smirnov statistic.

The window in which to perform the randomization-based tests can be set manually using `wl` and `wr`. This options specify the lower and upper limits of the chosen window. Importantly, these are window limits and not lengths, so for instance, if the cutoff is 100 and the user wants a window of ±5, the correct syntax is `wl` = 95 and `wr` = 105. We advise the user to always normalize the cutoff to zero by centering the running variable to avoid confusion.

Alternatively, the user can specify the list of covariates to have `rdrandinf` select the window automatically using `rdwinselect`. All the options allowed in `rdwinselect` can be passed through `rdrandinf`. For example:

```
> tmp = rdrandinf(Y,R,statistic='all',covariates=X,wmin=.5,wstep=.125,rdwreps=10000)

Running rdwinselect...

rdwinselect complete.

Selected window = [-0.75;0.75]

Running randomization-based test...
```

```
Randomization-based test complete.


Number of obs =        1297
Order of poly =        0
Kernel type   =        uniform
Reps          =        1000
Window        =        rdwinselect
H0:       tau =        0
Randomization =        fixed margins

Cutoff c = 0           Left of c       Right of c
Number of obs          595             702
Eff. number of obs     15              22
Mean of outcome        42.808          52.497
S.d. of outcome        7.042           7.742
Window                 -0.75           0.75

                                   Finite sample        Large sample

Statistic              T              P>|T|          P>|T|      Power vs d = 3.521

Diff. in means         9.689          0              0          0.3
Kolmogorov-Smirnov     0.552          0.004          0.005      NA
Rank sum z-stat        -3.217         0.001          0.001      0.209
```

Note that the reported p-values are slightly different. As explained above, the reason is that the two commands are performing the randomization test starting from different seeds. The user can obtain the exact same results for the two syntaxes by setting the same seed—for example, `seed=9876`—in both commands.

The `rdrandinf` command allows the user to specify a polynomial transformation model for the outcomes using the option `p`. By default, the command sets `p=0`, which means no transformation. When the `p` is set to an integer larger than zero, the slopes (and possibly higher order terms) are subtracted from the outcomes, leaving a residualized version of the outcome that only differs above and below the cutoff in the intercept. For instance, to perform a linear transformation, the syntax is:

```
> tmp = rdrandinf(Y,R,wl=-.75,wr=.75,statistic='all',p=1)

Selected window = [-0.75;0.75]

Running randomization-based test...

Randomization-based test complete.
```

```
Number of obs =       1297
Order of poly =       1
Kernel type   =       uniform
Reps          =       1000
Window        =       set by user
H0:       tau =       0
Randomization =       fixed margins

Cutoff c = 0          Left of c       Right of c
Number of obs         595             702
Eff. number of obs    15              22
Mean of outcome       42.808          52.497
S.d. of outcome       7.042           7.742
Window                -0.75           0.75

                                 Finite sample        Large sample

Statistic            T              P>|T|          P>|T|      Power vs d = 3.521

Diff. in means       15.297         0              0.066      0.071
Kolmogorov-Smirnov   0.797          0              NA         NA
Rank sum z-stat      -4.455         0              NA         NA
```

When a model for the outcomes is specified—that is, when `p` is set to a number greater than zero—with the option `statistic="ttest"` fits a regression of the outcome on the treatment dummy interacted with a polynomial of the running variable, and uses the difference in intercepts as the test-statistic. The other test-statistics use as outcomes the residuals described above. Note that the command does not provide the asymptotic p-value nor the asymptotic power of the Kolmogorov-Smirnov and rank sum statistics, as the asymptotic distribution does not account for the model transformation and hence can be misleading.

In the presence of arbitrary interference, a confidence interval for a particular measure of the effects of the program can be obtained with the `interfci` option. For example, to obtain a 95 percent confidence interval, we type:

```
> tmp = rdrandinf(Y,R,wl=-.75,wr=.75,interfci=.05)

Selected window = [-0.75;0.75]

Running randomization-based test...

Randomization-based test complete.


Number of obs =       1297
Order of poly =       0
Kernel type   =       uniform
Reps          =       1000
```

```
Window        =       set by user
H0:       tau =       0
Randomization =       fixed margins

Cutoff c = 0          Left of c       Right of c
Number of obs         595             702
Eff. number of obs    15              22
Mean of outcome       42.808          52.497
S.d. of outcome       7.042           7.742
Window                -0.75           0.75

                                  Finite sample        Large sample

Statistic             T               P>|T|          P>|T|      Power vs d = 3.521

Diff. in means        9.689           0              0          0.3


0.95% confidence interval under interference: [3.981;15.283]
```

In terms of interpretation, it is important to keep in mind that the confidence interval under interference is not a confidence interval for the point estimate (and in fact, it may even not contain the point estimate). The interference confidence interval is constructed based on the difference between the observed statistic and the statistic that would be observed if the treatment was withheld from all units. In our application, allowing for arbitrary interference we can say with 95 percent confidence that the "excess" benefit of the treated group compared to the control group is roughly between 3.98 and 15. Again, in this particular example the point estimate under SUTVA happens to be contained in the confidence interval under interference, but this need not be the case and has no clear interpretation.

The `rdlocrand` package provides two types of sensitivity analyses to assess how p-values change with window length. The first one, `rdsensitivity`, calculates and plots a matrix of p-values over a range of values for the treatment effect under the null hypothesis (rows) and window lengths (columns). For instance, we can see how the p-values change by starting from the selected window, increasing the window length by 0.25 and over a range of treatment effects that is roughly the point estimate plus and minus 10:

```
> rdsensitivity(Y,R,wlist=seq(.75,2,by=.25),tlist=seq(0,20,by=1))

Running sensitivity analysis...

Sensitivity analysis complete.
$tlist
 [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20

$wlist
[1] 0.75 1.00 1.25 1.50 1.75 2.00
```

```
$results
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
 [1,] 0.000 0.000 0.000 0.000 0.000 0.000
 [2,] 0.001 0.000 0.000 0.000 0.000 0.000
 [3,] 0.004 0.003 0.000 0.000 0.000 0.000
 [4,] 0.005 0.003 0.002 0.001 0.001 0.000
 [5,] 0.030 0.007 0.002 0.002 0.001 0.002
 [6,] 0.072 0.017 0.011 0.017 0.006 0.009
 [7,] 0.146 0.071 0.035 0.040 0.025 0.028
 [8,] 0.285 0.147 0.074 0.109 0.082 0.105
 [9,] 0.509 0.292 0.158 0.246 0.225 0.276
[10,] 0.778 0.561 0.281 0.447 0.448 0.566
[11,] 0.925 0.835 0.533 0.721 0.806 0.928
[12,] 0.597 0.817 0.746 0.923 0.800 0.662
[13,] 0.364 0.527 0.970 0.591 0.478 0.352
[14,] 0.207 0.255 0.667 0.337 0.231 0.150
[15,] 0.098 0.130 0.434 0.182 0.084 0.034
[16,] 0.043 0.041 0.288 0.065 0.034 0.017
[17,] 0.022 0.019 0.140 0.022 0.009 0.005
[18,] 0.006 0.011 0.077 0.006 0.001 0.000
[19,] 0.003 0.000 0.025 0.000 0.001 0.000
[20,] 0.001 0.000 0.016 0.000 0.000 0.000
[21,] 0.000 0.000 0.006 0.000 0.000 0.000
```

Note that the `rdsensitivity` command does not display any output, so we show the return values to illustrate the results. In addition to the p-values, the `rdsensitivity` command returns the plot shown in Figure SAF-2. The plot depicts the grid of window lengths in the horizontal axis and the grid of treatment effects under the null. The color represents the p-value for each pair of window length and treatment effect, where white corresponds to zero and black corresponds to one. This is simply a graphical display of the results given by `rdsensitivity`. The plot can be replicated (or modified) with the following code:

```
> tmp = rdsensitivity(Y,R,wlist=seq(.75,2,by=.25),tlist=seq(0,20,by=1))

Running sensitivity analysis...

Sensitivity analysis complete.
> xaxis = tmp$wlist
> yaxis = tmp$tlist
> zvalues = tmp$results
> filled.contour(xaxis,yaxis,t(zvalues),
+                xlab='window',ylab='treatment effect',
+                key.title=title(main = 'p-value',cex.main=.8),
+                levels=seq(0,1,by=.01),col=gray.colors(100,1,0))
```

One way to interpret these results is to see the range of values for which the p-value is above, say, .05, as a 95 percent confidence interval for the point estimate (assuming a constant additive treatment effect). Thus, the above table shows how the confidence interval for the treatment effect changes as the window length increases. For instance, the 95 percent confidence interval for the window $[-.75\,;\,.75]$ is roughly $[5\,;\,14]$, whereas for the window $[-2\,;\,2]$ it becomes $[7\,;\,14]$. In this case, the point estimate seems to be relatively stable over the range of windows considered. The confidence interval for the window $[-.75\,;\,.75]$ can be obtained using the `ci` option:

```
> tmp = rdsensitivity(Y,R,wlist=seq(.75,2,by=.25),tlist=seq(0,20,by=1),ci=0.75)

Running sensitivity analysis...

Sensitivity analysis complete.
> tmp$ci
[1]  5 14
```

Additionally, `rdsensitivity` can be called from within `rdrandinf` to obtain confidence intervals for the point estimates obtained using the `ci` option. The syntax is the following:

```
> tmp = rdrandinf(Y,R,wl=-.75,wr=.75,ci=c(.05,seq(3,20,by=1)))

Selected window = [-0.75;0.75]

Running randomization-based test...

Randomization-based test complete.

Running sensitivity analysis...

Sensitivity analysis complete.


Number of obs =       1297
Order of poly =       0
Kernel type   =       uniform
Reps          =       1000
Window        =       set by user
H0:       tau =       0
Randomization =       fixed margins

Cutoff c = 0          Left of c       Right of c
Number of obs         595             702
Eff. number of obs    15              22
Mean of outcome       42.808          52.497
S.d. of outcome       7.042           7.742
Window                -0.75           0.75

                                 Finite sample        Large sample

Statistic            T              P>|T|          P>|T|     Power vs d = 3.521
```

```
Diff. in means       9.689          0               0         0.3


0.95% confidence interval: [5,14]
```

The second type of sensitivity analysis is performed with `rdrbounds`. As explained above, this command calculates upper and lower bounds for the randomization p-value under Bernoulli trials for a range of values of a parameter $\Gamma \equiv \exp(\gamma)$ that captures the strength with which an unobservable binary variable U_i affects the probability of selection into treatment. The basic syntax is:

```
> rdrbounds(Y,R,expgamma=c(1.5,2,3),wlist=c(.5,.75,1),reps=1000)

Calculating randomization p-value...

Bernoulli p-value (w = 0.5) = 0.006
Bernoulli p-value (w = 0.75) = 0.002
Bernoulli p-value (w = 1) = 0

Running sensitivity analysis...

Sensitivity analysis complete.
$gamma
[1] 0.4054651 0.6931472 1.0986123

$expgamma
[1] 1.5 2.0 3.0

$wlist
[1] 0.50 0.75 1.00

$p.values
[1] 0.006 0.002 0.000

$lower.bound
            [,1] [,2] [,3]
[1,] 0.006000000    0    0
[2,] 0.008000000    0    0
[3,] 0.006018054    0    0

$upper.bound
      [,1]  [,2]  [,3]
[1,] 0.039 0.015 0.007
[2,] 0.105 0.064 0.029
[3,] 0.278 0.269 0.179
```

The output from `rdrbounds` is divided in several parts. The first one shows the randomization p-value based on Bernoulli trials for each window. The matrices `$lower.bound` and `$upper.bound` present the lower and upper bounds for the p-values for different values of γ and windows. The wider the distance between the lower and upper bounds, the more sensitive the inference to deviations from a randomized experiment. The remaining elements give the list of values used to calculate the bounds.

The `fmpval` option adds the fixed margins randomization p-value to the first panel of the output. This allows the user to compare the p-values obtained using each method.

```
> tmp = rdrbounds(Y,R,expgamma=c(1.5,2,3),wlist=c(.5,.75,1),reps=1000,fmpval=TRUE)

Calculating randomization p-value...

Bernoulli p-value (w = 0.5) = 0.006
Fixed margins p-value (w = 0.5) = 0.009
Bernoulli p-value (w = 0.75) = 0.002
Fixed margins p-value (w = 0.75) = 0.003
Bernoulli p-value (w = 1) = 0
Fixed margins p-value (w = 1) = 0.001

Running sensitivity analysis...

Sensitivity analysis complete.
```

We can see that the p-values obtained by both methods are very similar, which we found to be usually true in applications as long as the number of replications is large enough.

Finally, when using outcome transformation, the `rdrandinf` command allows the user to choose in which point to evaluate the transformed outcomes. By default, the evaluation point is the cutoff, which emulates the idea used in the local polynomial approach of estimating the effect at the cutoff. However, whenever the local randomization assumption is plausible, the cutoff need not be the point of interest. For instance, to set the evaluation points at the means of the running variable inside the window below and above the cutoff, we can type:

```
> ii = (R>=-.75) & (R<=.75) & !is.na(Y) & !is.na(R)
> m0 = mean(R[ii & D==0],na.rm=TRUE)
> m1 = mean(R[ii & D==1],na.rm=TRUE)
> tmp = rdrandinf(Y,R,wl=-.75,wr=.75,p=1,evall=m0,evalr=m1)

Selected window = [-0.75;0.75]

Running randomization-based test...

Randomization-based test complete.


Number of obs =       1297
```

```
Order of poly =       1
Kernel type   =       uniform
Reps          =       1000
Window        =       set by user
H0:       tau =       0
Randomization =       fixed margins

Cutoff c = 0          Left of c       Right of c
Number of obs         595             702
Eff. number of obs    15              22
Mean of outcome       42.808          52.497
S.d. of outcome       7.042           7.742
Window                -0.75           0.75

                                Finite sample       Large sample

Statistic             T              P>|T|          P>|T|      Power vs d = 3.521

Diff. in means        9.689          0              0          0.283
```

The user can verify that the point estimate in this case is the same as when no transformation is used, which is due to the fact that the transformation comes from a linear regression which by construction passes through the means. The p-values, however, can differ. Incidentally, note that the means are taken over the sample inside the window with non-missing values for the outcome and the running variable. The reason is that `rdrandinf` drops the observations inside the window with missing outcomes of running variable. Similarly, `rdwinselect` drops, at each evaluated window, the observations with missing values of the covariates and running variable.

7 Auxiliary Functions

This section introduces the auxiliary functions used by the four functions described above. All the auxiliary functions are included in the file `rdlocrand_fun`. We do not recommend to use these functions directly, as they typically do not contain error handling and incorrectly using them may lead to unexpected mistakes.

7.1 Statistics for randomization inference

```
rdrandinf.model(Y,D,statistic,pvalue=FALSE,kweights,endogtr,delta="")
```

This function uses the outcome and treatment variable $D = \mathbf{1}(R \geq c)$ to calculate the observed statistics, asymptotic p-values and asymptotic power for all the statistics used by `rdrandinf`, namely, `"ttest"`, `"ksmirnov"`, `"ranksum"`, `"all"`, `"ar"` and `"wald"`.

7.2 Hotelling's T^2 statistic

`hotelT2(X,D)`

This function computes Hotelling's T^2 statistic and its asymptotic p-value for the matrix of covariates `X` for `rdwinselect`.

7.3 Default window length

`wlength(R,D,num)`

This function finds the window containing at least `num` observations at each side of the cutoff. It is used to find the default initial window in `rdwinselect`.

7.4 Default window increment

`findstep(R,D,obsmin,obsstep,times)`

This function finds a list of increments of length `times`, starting from window with `obsmin` observations and adding `obsstep` observations at each side in each step. It is used to find the default window increments in `rdwinselect`.

References

Matias D. Cattaneo, Brigham Frandsen, and Rocio Titiunik. Randomization inference in the regression discontinuity design: An application to party advantages in the u.s. senate. *Journal of Causal Inference*, 3(1):1–24, 2015.

Matias D. Cattaneo, Rocio Titiunik, and Gonzalo Vazquez-Bare. Comparing inference approaches for rd designs: A reexamination of the effect of head start on child mortality. working paper, University of Michigan, 2016a.

Matias D. Cattaneo, Rocio Titiunik, and Gonzalo Vazquez-Bare. Inference in regression discontinuity designs under local randomization. *Stata Journal*, 16(2):331–367, 2016b.

Paul R. Rosenbaum. *Observational Studies*. Springer, New York, 2002.

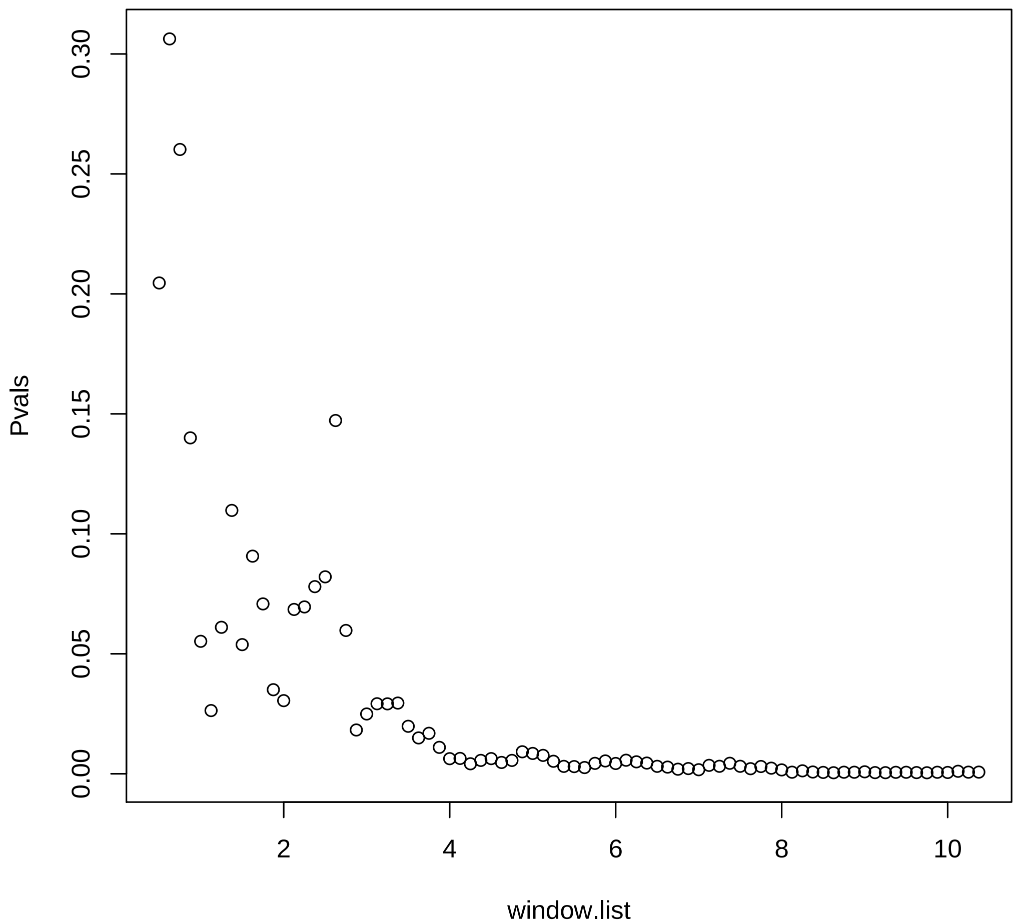

Figure SAF-1. Plot of p-values.

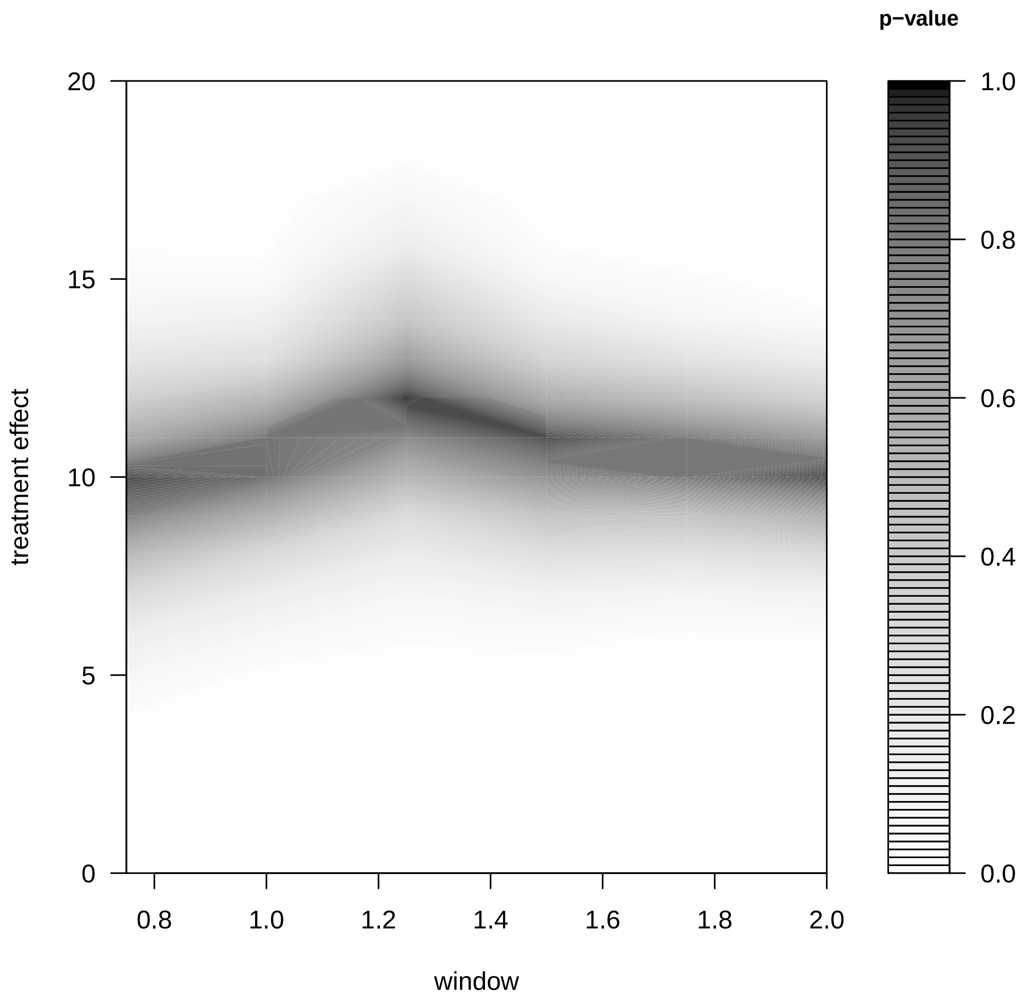

Figure SAF-2. Sensitivity analysis.