Journal of Statistical Software
MMMMMM YYYY, Volume VV, Issue II. http://www.jstatsoft.org/

# rddtools: tools for Regression Discontinuity Design in R

**Matthieu Stigler**
UC Davis

**Bastiaan Quast**
The Graduate Institute, Geneva

**Abstract**

The rddtools package implements functions for handling Regression Discontinuity Design in R.

*Keywords*: RDD, Regression, Discontinuity, Design, R.

## 1. Introduction

The `rddtools` package address

## 2. Design

A unified framework is implemented through the `rdd_data` class which inherits from the `R base data.frame` class. This functionality is made accessible throug hthe associated `rdd_data()` functions and methods.

The package is designed to leveredge of existing implementations of **Regression Discontinuity Design** in R, such as the `rdd` package.

It implements several variants of RDD previously not implemented.

- Simple visualisation of the data using binned-plot: `plot()`
- Bandwidth selection:
- MSE-RDD bandwidth procedure of (G. Imbens and Kalyanaraman 2012): `rdd_bw_ik()`
- MSE global bandwidth procedure of (Ruppert, Sheather, and Wand 1995): `rdd_bw_rsw()`
- Estimation:
- RDD parametric estimation: `rdd_reg_lm()` This includes specifying the polynomial

order, including covariates with various specifications as advocated in (G. W. Imbens and Lemieux 2008).

- RDD local non-parametric estimation: `rdd_reg_np()`. Can also include covariates, and allows different types of inference (fully non-parametric, or parametric approximation).
- RDD generalised estimation: allows to use custom estimating functions to get the RDD coefficient. Could allow for example a probit RDD, or quantile regression.
- Post-Estimation tools:
- Various tools, to obtain predictions at given covariate values ( `rdd_pred()` ), or to convert to other classes, to lm ( `as.lm()` ), or to the package `np` ( `as.npreg()` ).
- Function to do inference with clustered data: `clusterInf()` either using a cluster covariance matrix ( **vcovCluster()** ) or by a degrees of freedom correction (as in (Cameron, Gelbach, and Miller 2008)).
- Regression sensitivity analysis:
- Plot the sensitivity of the coefficient with respect to the bandwith: `plotSensi()`
- **Placebo plot** using different cutpoints: `plotPlacebo()`
- Design sensitivity analysis:
- McCrary test of manipulation of the forcing variable: wrapper `dens_test()` to the function `DCdensity()` from package `rdd`.
- Test of equal means of covariates: `covarTest_mean()`
- Test of equal density of covariates: `covarTest_dens()`
- Datasets
- Contains the seminal dataset of Lee (2008): `house`
- Contains functions to replicate the Monte-Carlo simulations of [Imbens and Kalyanaraman 2012]: `gen_mc_ik()`

# 3. Application

we use the data from the Initiative Nationale du Development Humaine (INDH) a development project in Morocco. The data is included with the `rddtools` package under the name `indh`.

```
data("indh")
```

Now that we have loading the data we can briefly inspect the structure of the data.

```
summary(indh)
```

```
   choice_pg         commune          poverty
 Min.   :0.0000   Min.   :28.09   Min.   :28.09
 1st Qu.:0.0000   1st Qu.:29.01   1st Qu.:29.01
 Median :1.0000   Median :29.95   Median :29.95
 Mean   :0.6722   Mean   :29.73   Mean   :29.73
 3rd Qu.:1.0000   3rd Qu.:30.34   3rd Qu.:30.34
 Max.   :1.0000   Max.   :30.97   Max.   :30.97
```

The `indh` object is a `data.frame` containing 729 observations (representing individuals) of three variables:

- `choice_pg`
- `commune`
- `poverty`

The variable of interest is `choice_pg`, which represent the decision to contibute to a public good or not. The observations are individuals choosing to contribute or not, these individuals are clustered by the variable `commune` which is the municiple structure at which funding was distributed as part of the INDH project. The forcing variable is `poverty` which represents the number of households in a commune living below the poverty threshold. As part of the INDH, commune with a proportion of household below the poverty threshhold greater than 30% were allowed to distribute the funding using a **Community Driven Development** scheme. The cutoff point for our analysis is therefore `30`.

We can now transform the `data.frame` to a special `rdd_data` `data.frame` using the `rdd_data()` function.

```
rdd_dat_indh <- rdd_data(y=choice_pg,
                         x=poverty,
                         data=indh,
                         cutpoint=30 )
```

The structure is similar but contains some additional information.

```
str(rdd_dat_indh)
```

```
Classes 'rdd_data' and 'data.frame':    729 obs. of  2 variables:
 $ x: num  30.1 30.1 30.1 30.1 30.1 ...
 $ y: int  0 1 1 1 1 1 0 1 0 0 ...
 - attr(*, "hasCovar")= logi FALSE
 - attr(*, "labels")= list()
 - attr(*, "cutpoint")= num 30
 - attr(*, "type")= chr "Sharp"
```

In order to best understand our data, we start with an exploratory data analysis using tables…

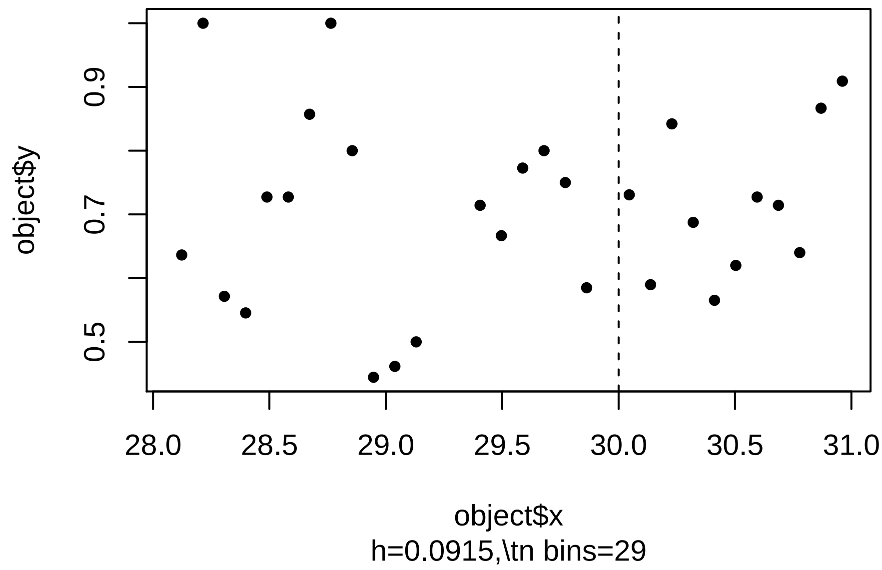
```
summary(rdd_dat_indh)
```

```
### rdd_data object ###

Cutpoint: 30
Sample size:
    -Full : 729
    -Left : 371
    -Right: 358
Covariates: no
```

...and plots.

```
plot(rdd_dat_indh[1:715,])
```




We can now continue with a standard Regression Discontinuity Design (RDD) estimation.

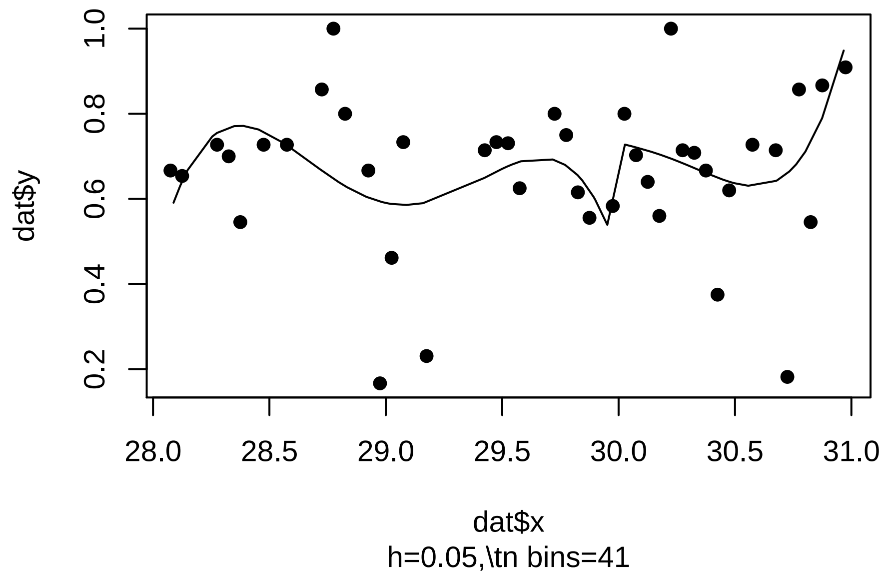
```
(reg_para <- rdd_reg_lm(rdd_dat_indh, order=4))
```

```
### RDD regression: parametric ###
    Polynomial order:  4
    Slopes:  separate
    Number of obs: 729 (left: 371, right: 358)

    Coefficient:
  Estimate Std. Error t value Pr(>|t|)
D  0.26428    0.16590   1.593   0.1116
```

and visualising this estimation.

```
plot(reg_para)
```




In addition to the parametric estimation, we can also perform a non-parametric estimation.

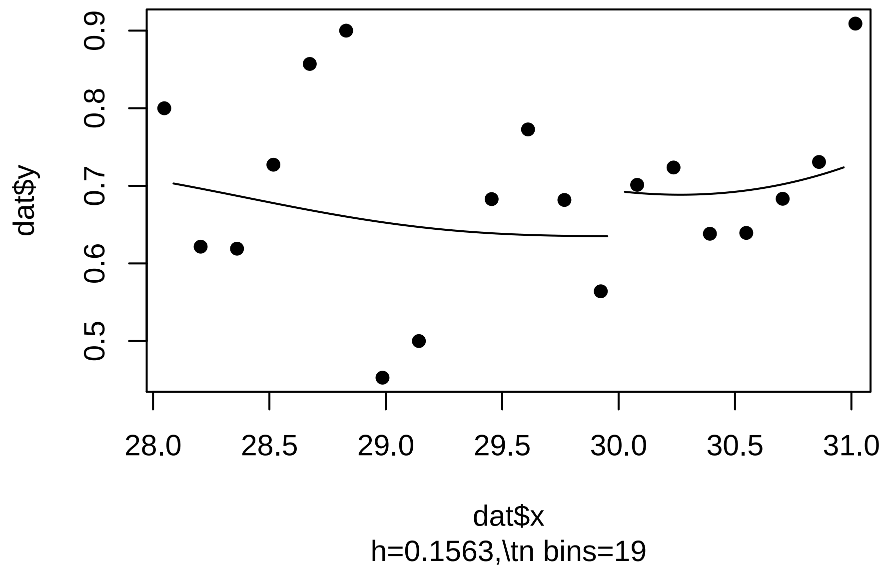
```
bw_ik <- rdd_bw_ik(rdd_dat_indh)
(reg_nonpara <- rdd_reg_np(rdd_object=rdd_dat_indh, bw=bw_ik))
```

```
### RDD regression: nonparametric local linear###
    Bandwidth:  0.7812904
    Number of obs: 467 (left: 146, right: 321)

    Coefficient:
  Estimate Std. Error z value Pr(>|z|)
D 0.178174   0.095319  1.8692  0.06159 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

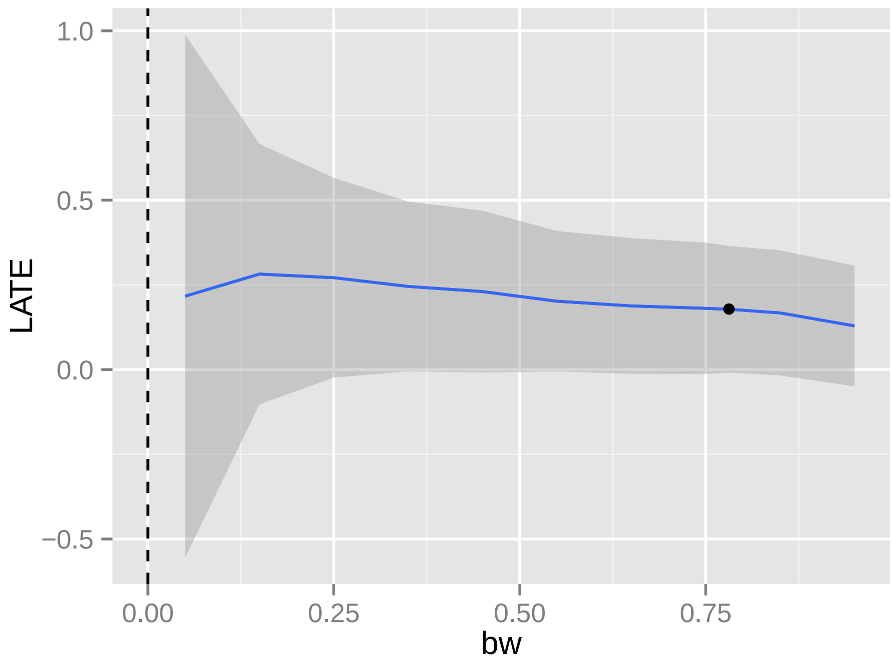
and visualising the non-parametric estimation.

```
plot(reg_nonpara)
```




Sensitity tests.

```
plotSensi(reg_nonpara, from=0.05, to=1, by=0.1)
```



## References

Cameron, A Colin, Jonah B Gelbach, and Douglas L Miller. 2008. "Bootstrap-Based Improvements for Inference with Clustered Errors." *The Review of Economics and Statistics* 90 (3). MIT Press: 414–27.

Imbens, Guido W, and Thomas Lemieux. 2008. "Regression Discontinuity Designs: A Guide to Practice." *Journal of Econometrics* 142 (2). Elsevier: 615–35.

Imbens, Guido, and Karthik Kalyanaraman. 2012. "Optimal Bandwidth Choice for the Regression."

Lee, David S. 2008. "Randomized Experiments from Non-Random Selection in US House Elections." *Journal of Econometrics* 142 (2). Elsevier: 675–97.

Ruppert, David, Simon J Sheather, and Matthew P Wand. 1995. "An Effective Bandwidth Selector for Local Least Squares Regression." *Journal of the American Statistical Association* 90 (432). Taylor & Francis: 1257–70.

**Affiliation:**

Matthieu Stigler
UC Davis
California
E-mail: matthieu.stigler@gmail.com
URL: https://matthieustigler.github.io/ Bastiaan Quast
The Graduate Institute, Geneva
Maison de la paix Geneva, Switzerland
E-mail: bquast@gmail.com
URL: http://qua.st/

*Journal of Statistical Software* http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd