



rddtools: tools for Regression Discontinuity Design in R

Bastiaan Quast
The Graduate Institute, Geneva

Matthieu Stigler
UC Davis

Abstract

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

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

1. Introduction

The rddtools package...

2. Design

The package includes the following functions.

3. Application

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

```
library(rddtools)
```

```
Loading required package: AER
```

```
Loading required package: car
```

```
Loading required package: lmtest
```

Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

as.Date, as.Date.numeric

Loading required package: sandwich

Loading required package: survival

Warning: package 'survival' was built under R version 3.2.1

Loading required package: np

Nonparametric Kernel Methods for Mixed Datatypes (version 0.60-2)

[vignette("np_faq",package="np") provides answers to frequently asked questions]

IMPORTANT, this is an ALPHA VERSION

many changes to the API will follow

We can now load the included data set.

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

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

```
\begin{CodeChunk}
```

```
str(indh)
```

```
\begin{CodeOutput} 'data.frame': 729 obs. of 3 variables: $ choice_pg: int 0 1 1 1 1 1 0 1
0 0 ... $ commune : num 30.1 30.1 30.1 30.1 30.1 ... $ poverty : num 30.1 30.1 30.1 30.1
30.1 ... - attr(, "na.action")=Class 'omit' Named int [1:11] 58 289 290 291 292 293 294 295
296 297 ... .. - attr(, "names")= chr [1:11] "58" "289" "290" "291" ... \end{CodeOutput}
\end{CodeChunk}
```

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 contribute 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 municipality 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 threshold 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...

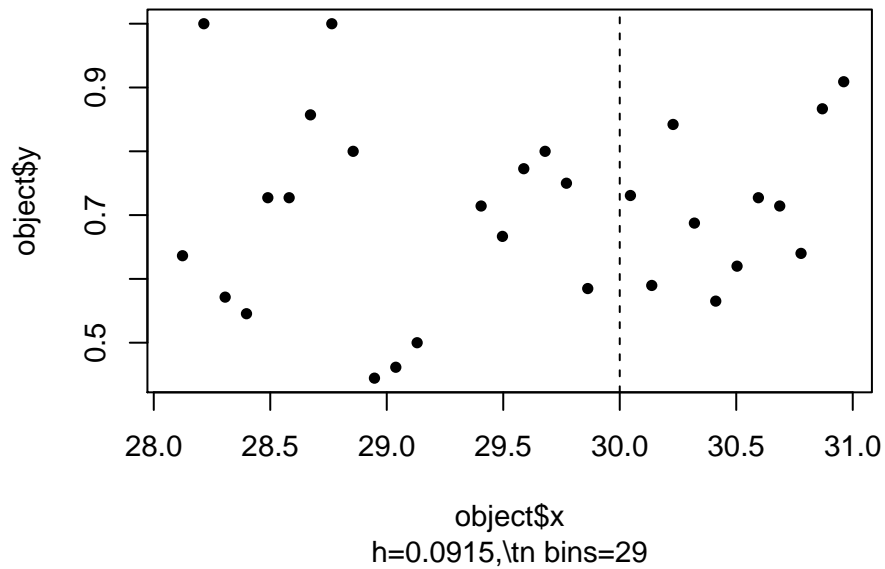
```
summary(rdd_dat_indh)
```

```
### 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.

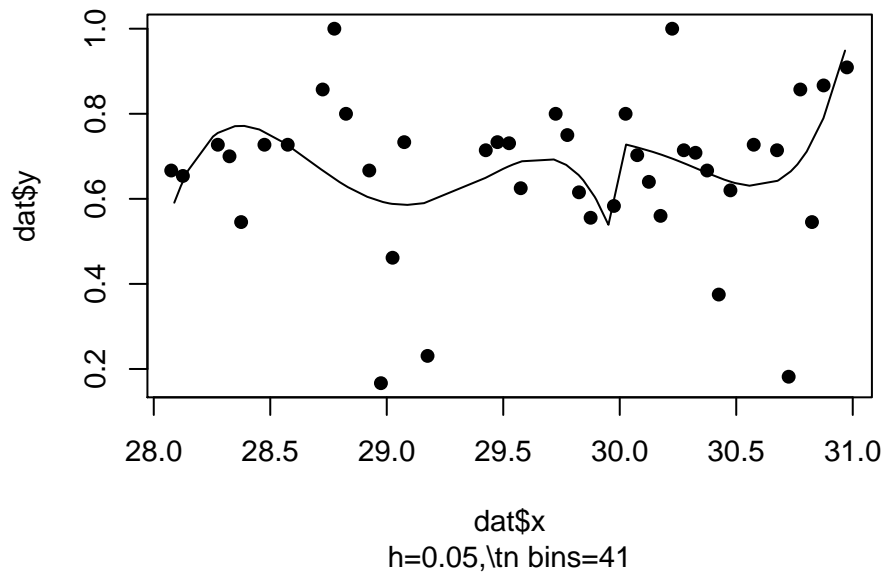
```
(reg_para <- rdd_reg_lm(rdd_dat_indh, order=4))
```

```
### 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.

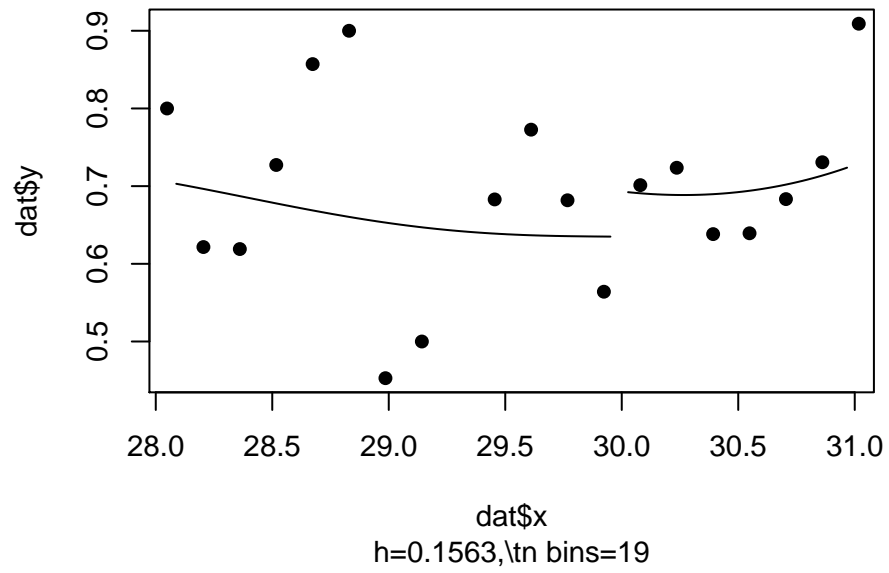
```
bw_ik <- rdd_bw_ik(rdd_dat_indh)
(reg_nonpara <- rdd_reg_np(rdd_object=rdd_dat_indh, bw=bw_ik))
```

```
### 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
```

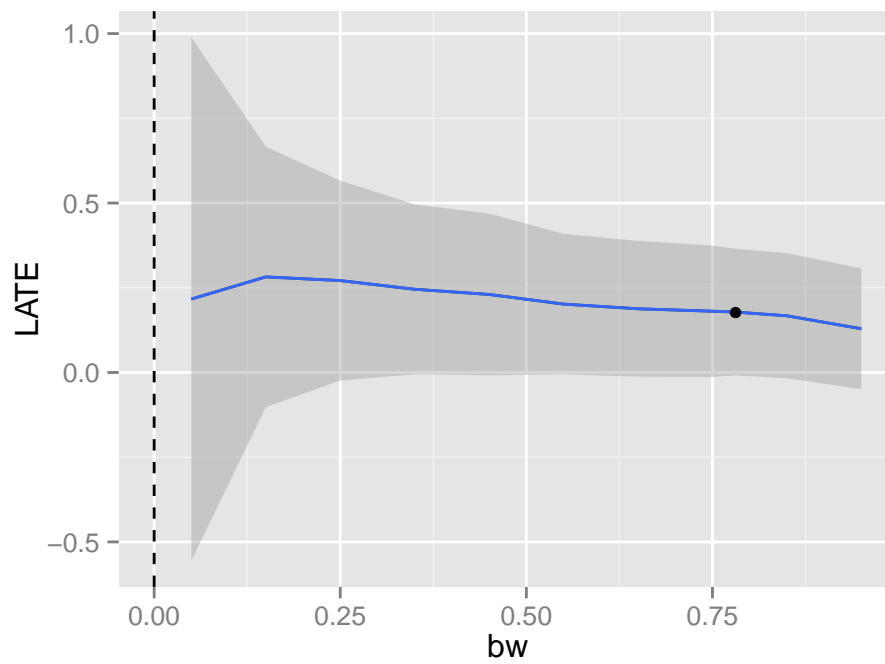
and visualising the non-parametric estimation.

```
plot(reg_nonpara)
```



Sensitivity tests.

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



4. References

Affiliation:

Bastiaan Quast

The Graduate Institute, Geneva

Maison de la paix Geneva, Switzerland

E-mail: bquast@gmail.com

URL: <http://qua.st/> Matthieu Stigler

UC Davis

California

E-mail: matthieustigler@gmail.com

URL: <https://matthieustigler.github.io/>