# SOSC 5340 Tutorial Four

Instrumental Variable, and Regression Discontinuity

*Yabin YIN*
*HKUST*

*April, 2021*

## Set working directory to the current directory

*Remark:* Need to save current R file before using *getActiveDocumentContext*

## R Packages

**R** packages for IV strategy:

- *ivreg*: https://cran.r-project.org/web/packages/ivreg/index.html
- *lfe*: https://cran.r-project.org/web/packages/lfe/index.html
  - linear models with multiple group fixed effects
  - deals with many levels of "fixed effect"
  - allows for multi-way clustering s.e.
  - can implement IV estimation
- Read the *reference manual* and *vignettes*.

**R** packages for RD strategy:

- *rdrobust*: https://cran.r-project.org/web/packages/rdrobust/index.html
- *rddensity*: https://cran.r-project.org/web/packages/rddensity/index.html
- Read the *reference manual* and *vignettes*.
- Other packages for RD: https://rdpackages.github.io/

## Instrumental Variable

We will use `ivreg`, `lfe`, and `plm` package to fit instrumental-variable regression by two-stage least squares (2SLS).

Let's use the demand for cigarettes as an example, you can find it in @StockWatson2007 (Chapter 12).

```
# library packages
library(AER)
```

```
## Loading required package: car

## Loading required package: carData

## 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
```

```r
library(ivreg)
```

```
## Registered S3 methods overwritten by 'ivreg':
##   method              from
##   anova.ivreg         AER
##   hatvalues.ivreg     AER
##   model.matrix.ivreg  AER
##   predict.ivreg       AER
##   print.ivreg         AER
##   print.summary.ivreg AER
##   summary.ivreg       AER
##   terms.ivreg         AER
##   update.ivreg        AER
##   vcov.ivreg          AER

##
## Attaching package: 'ivreg'

## The following objects are masked from 'package:AER':
##
##     ivreg, ivreg.fit
```

```r
# data processing and transformation
data("CigarettesSW", package = "AER")
?CigarettesSW ## type '?CigarettesSW' to see the introduction of the dataset

## packs: Number of packs per capita
## price: Average price during fiscal year, including sales tax;
## cpi: Consumer price index;
## income: State personal income (total, nominal);
## population: State population;
## tax: Average state, federal and average local excise taxes for fiscal year;
## taxs: Average excise taxes for fiscal year, including sales tax.

CigarettesSW$rprice <- with(CigarettesSW, price/cpi) # real average price
CigarettesSW$rincome <- with(CigarettesSW, income/population/cpi) # real personal income
CigarettesSW$rtax <- with(CigarettesSW, tax/cpi) # real local excise taxes
CigarettesSW$rtdiff <- with(CigarettesSW, (taxs - tax)/cpi) # diff in real local taxes and real taxes

# Estimation
## OLS estimator
ols <- lm(log(packs) ~ log(rprice) + log(rincome),
          data = CigarettesSW, subset = year == 1995)
ols_se <- coeftest(ols, vcov = vcovHC, type = "HC1")

## Equation 12.15
ivreg_12.15 <- ivreg(log(packs) ~ log(rprice) + log(rincome)| ## 2nd stage
                       log(rincome) + rtdiff, ## 1st stage: rtdiff as IV of rprice
                     data = CigarettesSW,
```

```r
                      subset = year == 1995)
ivreg_12.15_se <- coeftest(ivreg_12.15, vcov = vcovHC, type = "HC1")

## Equation 12.16
ivreg_12.16 <- ivreg(log(packs) ~ log(rprice) + log(rincome) |
                       log(rincome) + rtdiff + rtax, ## 1st stage: rtdiff + rtax as IV of rprice
                     data = CigarettesSW,
                     subset = year == 1995)
ivreg_12.16_se <- coeftest(ivreg_12.16, vcov = vcovHC, type = "HC1")

## Show the results
library(texreg)
```

```
## Version:  1.37.5
## Date:     2020-06-17
## Author:   Philip Leifeld (University of Essex)
##
## Consider submitting praise using the praise or praise_interactive functions.
## Please cite the JSS article in your publications -- see citation("texreg").
```

```r
screenreg(list(ols, ivreg_12.15, ivreg_12.16),
          custom.model.names = c('OLS',"IV_rtdiff", "IV_rtdiff+rtax"),
          custom.coef.names = c("Constant", "log price", "log income per capita"),
          override.se = list(ols_se[,2], ivreg_12.15_se[,2], ivreg_12.16_se[,2]),
          override.pvalues = list(ols_se[,4], ivreg_12.15_se[,4], ivreg_12.16_se[,4]),
          #stars = c(0.1, 0.05, 0.01),
          digits = 4)
```

```
##
## ================================================================
##                        OLS          IV_rtdiff    IV_rtdiff+rtax
## ----------------------------------------------------------------
## Constant               10.3420 ***   9.4307 ***   9.8950 ***
##                        (0.9665)     (1.2594)     (0.9592)
## log price              -1.4065 ***  -1.1434 **   -1.2774 ***
##                        (0.2609)     (0.3723)     (0.2496)
## log income per capita   0.3439       0.2145       0.2804
##                        (0.2604)     (0.3117)     (0.2539)
## ----------------------------------------------------------------
## R^2                     0.4327       0.4189       0.4294
## Adj. R^2                0.4075       0.3931       0.4041
## Num. obs.              48           48           48
## ================================================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

```r
# Note: statistical significance level!
```

### Diagnostics: Weak IV and Over-identification test

A good instrumental variable is highly correlated with one or more of the explanatory variables while remaining uncorrelated with the errors.

If an endogenous regressor is only weakly related to the instrumental variables, then its coefficient will be estimated imprecisely. We hope for a large test statistic and small p-value in the diagnostic test for **weak instruments** (Weak instruments: **F-stat $> 10$**).

Applied to 2SLS regression, the **Wu–Hausman test** is a test of **endogenity**. If all of the regressors are exogenous (*a small test statistics and large p-value*), then both the OLS and 2SLS estimators are consistent, and the OLS estimator is more efficient. But if one or more regressors are endogenous (*a large test statistic and small p-value*), then 2SLS estimator may be better than the OLS estimator.

The **Sargan test** is a test of **overidentification**. When there are more instrumental variables than coefficients to estimate, it's possible that the instrumental variables provide conflicting information about the values of the coefficients. A large test statistic and small p-value for the Sargan test suggest, therefore, that the model is misspecified. The Sargan test is **inapplicable** to a just-identified regression equation, with an equal number of instrumental variables and coefficients.

```r
summary(ivreg_12.15, vcov = vcovHC, diagnostics = TRUE)
```

```
##
## Call:
## ivreg(formula = log(packs) ~ log(rprice) + log(rincome) | log(rincome) +
##     rtdiff, data = CigarettesSW, subset = year == 1995)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.611000 -0.086072  0.009423  0.106912  0.393159
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.4307     1.3538   6.966 1.15e-08 ***
## log(rprice)   -1.1434     0.4032  -2.836  0.00682 **
## log(rincome)   0.2145     0.3319   0.646  0.52137
##
## Diagnostic tests:
##                  df1 df2 statistic  p-value
## Weak instruments   1  45     40.03 1.02e-07 ***
## Wu-Hausman         1  44      0.96    0.332
## Sargan             0  NA        NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1896 on 45 degrees of freedom
## Multiple R-Squared: 0.4189,  Adjusted R-squared: 0.3931
## Wald test: 7.262 on 2 and 45 DF,  p-value: 0.001848
```

```r
summary(ivreg_12.16, vcov = vcovHC, diagnostics = TRUE)
```

```
##
## Call:
## ivreg(formula = log(packs) ~ log(rprice) + log(rincome) | log(rincome) +
##     rtdiff + rtax, data = CigarettesSW, subset = year == 1995)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.6006931 -0.0862222 -0.0009999  0.1164699  0.3734227
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.8950     1.0313   9.595 1.87e-12 ***
## log(rprice)   -1.2774     0.2689  -4.750 2.10e-05 ***
## log(rincome)   0.2804     0.2640   1.062    0.294
```

```
##
## Diagnostic tests:
##                  df1 df2 statistic p-value
## Weak instruments   2  44   167.988  <2e-16 ***
## Wu-Hausman         1  44     3.052  0.0876 .
## Sargan             1  NA     0.333  0.5641
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1879 on 45 degrees of freedom
## Multiple R-Squared: 0.4294,  Adjusted R-squared: 0.4041
## Wald test: 13.88 on 2 and 45 DF,  p-value: 2.019e-05
```

### Panel IV

Now, let's use `lfe` and `plm` package to fit a panel IV regression.

```r
library(lfe)
```

```
## Loading required package: Matrix

##
## Attaching package: 'lfe'

## The following object is masked from 'package:lmtest':
##
##     waldtest
```

```r
library(plm)
```

```
##
## Attaching package: 'plm'

## The following object is masked from 'package:lfe':
##
##     sargan
```

```r
felm_12.16 <- felm(log(packs) ~ log(rincome) |
                     0 | # Fixed Effects
                     (log(rprice) ~ rtdiff + rtax) | # Instruments
                     0,
                   data = CigarettesSW,
                   subset = year == 1995)

## Stock and Waston 2007: Table 12.1 Column 1
felm_panel <- felm(log(packs) ~ log(rincome) |
                     year | # Fixed Effects
                     (log(rprice) ~ rtdiff + rtax) | # Instruments
                     0,
                   data = CigarettesSW)

## try `plm`, firstly transform our original data to panel data
CigarettesSW_panel <- pdata.frame(CigarettesSW, index = c("state", "year"), drop.index = TRUE)

## Stock and Waston 2007: Table 12.1 Column 1
plm_within <- plm(log(packs) ~ log(rprice) + log(rincome) |
                    log(rincome) + rtdiff + rtax, ## IV
```

```r
             model = c("within"),
             data = CigarettesSW_panel)
plm_within_se <- coeftest(plm_within, vcov. = vcovHC, type='HC1')

screenreg(list(ivreg_12.16, felm_12.16, felm_panel, plm_within),
          override.se = list(ivreg_12.16_se[,2],
                             summary(felm_12.16)$coefficients[,2],
                    summary(felm_panel)$coefficients[,2], plm_within_se[,2]),
          override.pvalues = list(ivreg_12.16_se[,4],
                             summary(felm_12.16)$coefficients[,4],
                    summary(felm_panel)$coefficients[,4], plm_within_se[,4]),
          custom.model.names = c("ivreg 1995",
                                 "felm 1995",
                                 "felm panel",
                                 "plm within"),
          digits = 4)
```

```
##
## =========================================================================
##                          ivreg 1995    felm 1995     felm panel    plm within
## -------------------------------------------------------------------------
## (Intercept)               9.8950 ***    9.8950 ***
##                          (0.9592)      (1.0586)
## log(rprice)              -1.2774 ***                               -1.2675 ***
##                          (0.2496)                                  (0.1604)
## log(rincome)              0.2804        0.2804        0.2808 *      0.2038
##                          (0.2539)      (0.2386)      (0.1392)      (0.2351)
## `log(rprice)(fit)`                     -1.2774 ***   -1.1996 ***
##                                        (0.2632)      (0.1876)
## -------------------------------------------------------------------------
## R^2                       0.4294                                    0.8994
## Adj. R^2                  0.4041                                    0.7922
## Num. obs.                48            48            96            96
## R^2 (full model)                        0.4294        0.5495
## R^2 (proj model)                        0.4294        0.3837
## Adj. R^2 (full model)                   0.4041        0.5348
## Adj. R^2 (proj model)                   0.4041        0.3636
## Num. groups: year                                     2
## =========================================================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```

**Remark**: How to argue the exogeneity of instruments?

If you suspect that IV impacts on Y through A other than the endogenous variable, then regress A on the IV: will be fine if the result is not significant!

# Regression Discontinuity

We will use `rdrobust` and `rddensity` to deal with RD related analysis.

```r
## library packages and load data
library(rdrobust)

## simulated the data
set.seed(3333)
```

```r
s = 10 + 5*qnorm(runif(10000)) # running variable
x = s - 10 # covariate
w = ifelse(s>10, 1, 0) # treatment: threshold = 10
y1 = 600 + 6.5*x - 2*x^2 + 0.001*x^3 + 300*qnorm(runif(10000)) # treated outcome
y0 = 200 + 6.5*x - 0.20*x^2 + 0.01*x^3 + 300*qnorm(runif(10000)) # control outcome
y  = y0 + w*(y1-y0) # Rubin Causal Model

rd_data <- data.frame(s, x, w, y1, y0, y)
head(rd_data)
```

```
##           s          x w       y1       y0        y
## 1 13.144164  3.1441638 1 726.3628 476.6828 726.3628
## 2  9.211231 -0.7887692 0 607.6854 616.9262 616.9262
## 3  7.099556 -2.9004436 0 423.9116 455.4765 455.4765
## 4 14.099379  4.0993793 1 450.1038 346.2623 450.1038
## 5 11.029745  1.0297450 1 694.8643 301.8011 694.8643
## 6  5.619238 -4.3807620 0 245.9212 -15.3554 -15.3554
```

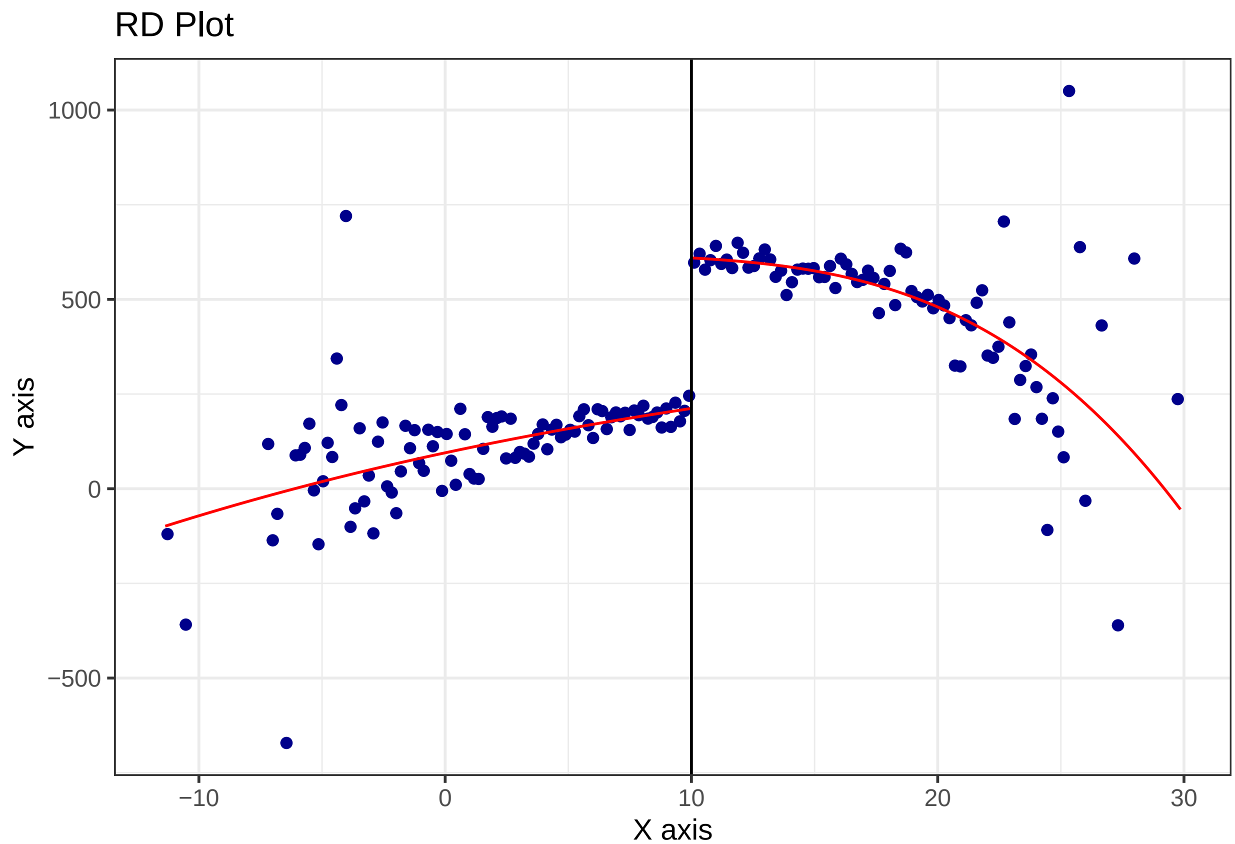
```r
summary(rd_data)
```

```
##        s                  x                    w                y1
##  Min.   :-11.367   Min.   :-21.367491   Min.   :0.0000   Min.   :-605.7
##  1st Qu.:  6.629   1st Qu.: -3.370934   1st Qu.:0.0000   1st Qu.: 345.1
##  Median :  9.991   Median : -0.009284   Median :0.0000   Median : 555.0
##  Mean   :  9.979   Mean   : -0.020945   Mean   :0.4991   Mean   : 550.9
##  3rd Qu.: 13.353   3rd Qu.:  3.352615   3rd Qu.:1.0000   3rd Qu.: 760.1
##  Max.   : 29.857   Max.   : 19.857085   Max.   :1.0000   Max.   :1587.6
##        y0                  y
##  Min.   :-900.243   Min.   :-900.2
##  1st Qu.:  -8.241   1st Qu.: 118.9
##  Median : 194.033   Median : 371.0
##  Mean   : 196.327   Mean   : 370.8
##  3rd Qu.: 401.281   3rd Qu.: 623.3
##  Max.   :1303.145   Max.   :1537.8
```

```r
## Step 1: Visualizing outcome discontinuity
attach(rd_data)
```

```
## The following objects are masked _by_ .GlobalEnv:
##
##     s, w, x, y, y0, y1
```

```r
rdplot(y=y, x=s, c=10, p=3)
```



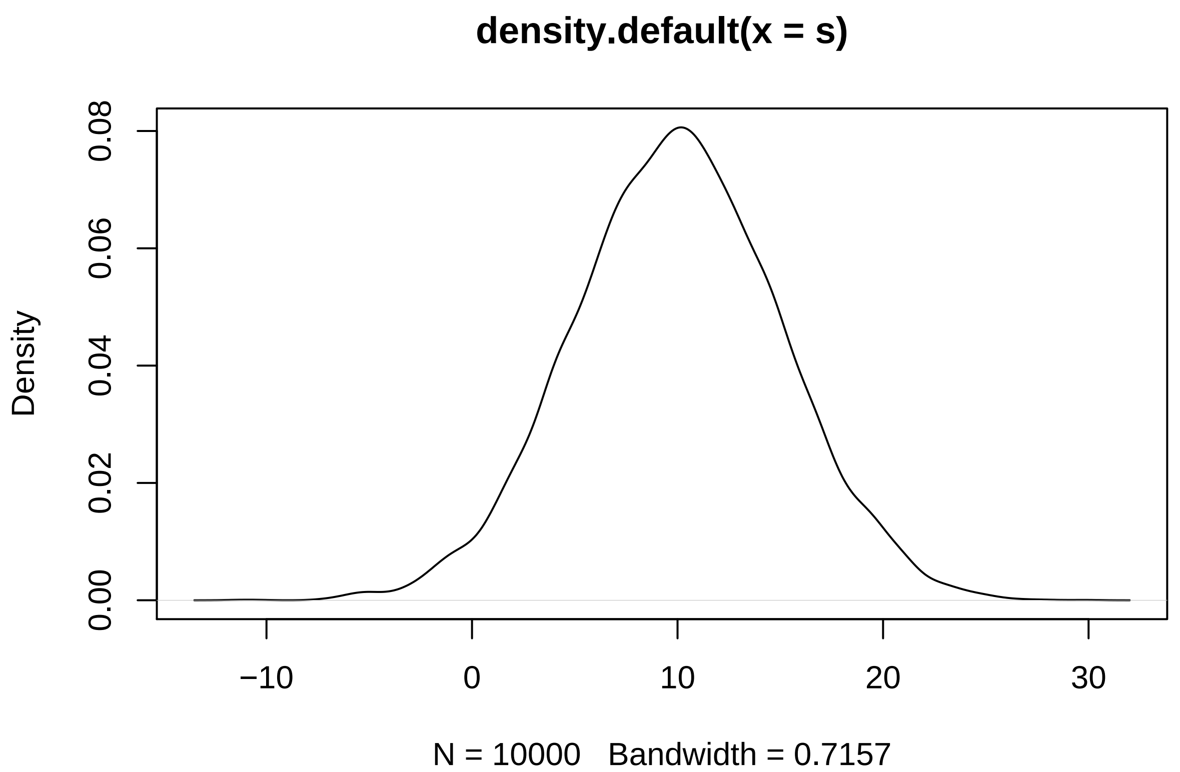

```
# y is the dependent variable, x is the running variable, c is the RD cutoff in x; p specifies the orde
```

```
## Step 2: Testing balancing at the threshold (covariate balance)
rd_data_balance <- subset(rd_data, s>=9 & s<=11)
t.test(x ~ w, data = rd_data_balance)
```

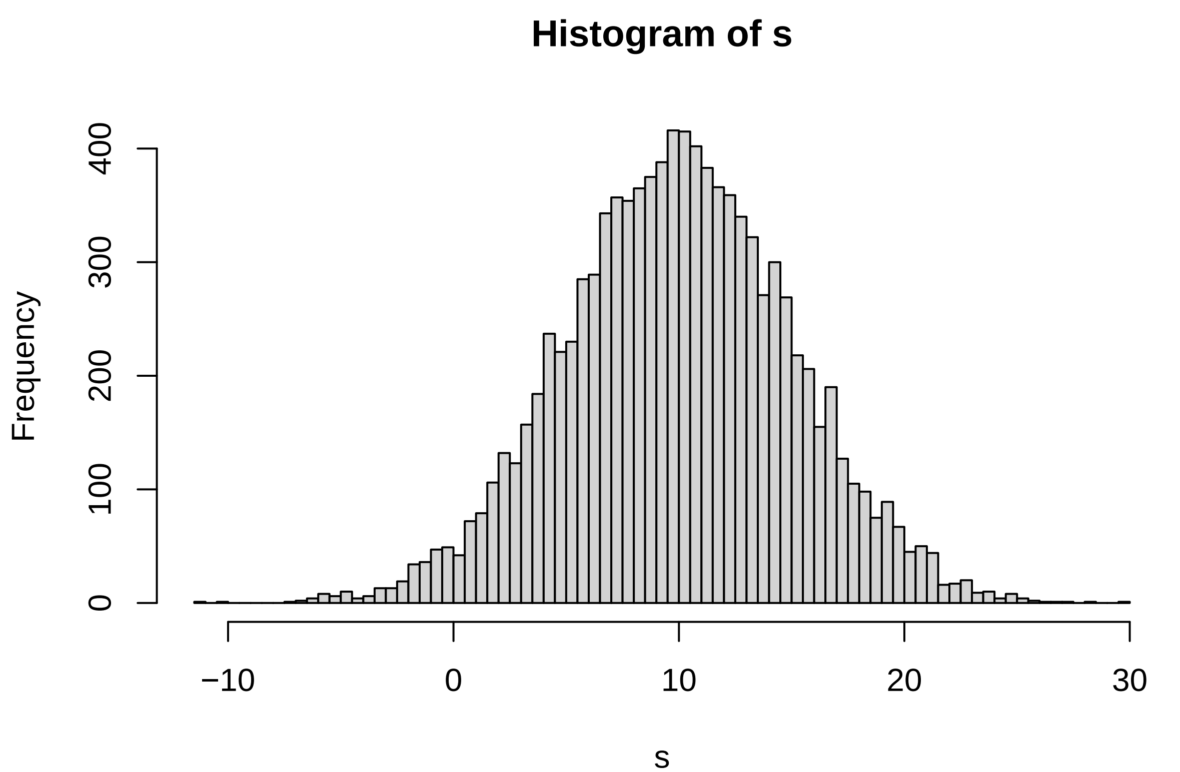
```
##
##  Welch Two Sample t-test
##
## data:  x by w
## t = -68.881, df = 1618.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.0204514 -0.9639448
## sample estimates:
## mean in group 0 mean in group 1
##      -0.4887857       0.5034123
```

```
## we want to test whether covariate x is balanced in the range [9,11]

## Step 3: Testing non-manipulation of the running variable (bunching of running variable)
plot(density(s))
```




```r
hist(s, breaks=100)
```




```r
library(rddensity)
summary(rddensity(s, c=10))
```

```
##
```

```
## Manipulation testing using local polynomial density estimation.
##
## Number of obs =       10000
## Model =               unrestricted
## Kernel =              triangular
## BW method =           estimated
## VCE method =          jackknife
##
## c = 10                Left of c            Right of c
## Number of obs         5009                 4991
## Eff. Number of obs    2174                 2579
## Order est. (p)        2                    2
## Order bias (q)        3                    3
## BW est. (h)           2.877                3.487
##
## Method                T                    P > |T|
## Robust                -0.0809              0.9355
##
##
## P-values of binomial tests (H0: p=0.5).
##
## Window Length / 2          <c      >=c     P>|T|
## 0.012                      14        6     0.1153
## 0.024                      23       18     0.5327
## 0.037                      36       25     0.2000
## 0.049                      44       37     0.5052
## 0.061                      55       48     0.5546
## 0.073                      65       56     0.4672
## 0.085                      74       63     0.3930
## 0.098                      90       74     0.2414
## 0.110                      99       83     0.2661
## 0.122                     111       90     0.1582
```

Now, let's use `rdrobust` function from `rdrobust` package to estimate ATE. Type `?rdrobust` to see the instruction of the function. Basically, you need to input:

- *y*: the dependent variable;
- *x*: the running variable
- *c*: RD cutoff in x;
- *fuzzy*: specifies the treatment status variable used to implement fuzzy RD estimation;
- *p*: specifies the order of the local-polynomial;
- *h*: specifies the main bandwidth used to construct the RD point estimator. If not specified, bandwidth h is computed by the companion command `rdbwselect`;
- *covs*: specifies additional covariates to be used in the polynomial regression.

```r
## Step 4: Estimating ATE by sharp RD
summary(rdrobust(y=y, x=s, c=10, p=3, covs=x))
```

```
## Call: rdrobust
##
## Number of Obs.                 10000
## BW type                        mserd
## Kernel                    Triangular
## VCE method                        NN
##
## Number of Obs.                 5009         4991
```

```
## Eff. Number of Obs.           3770        3780
## Order est. (p)                   3           3
## Order bias  (q)                  4           4
## BW est. (h)                  5.830       5.830
## BW bias (b)                  8.485       8.485
## rho (h/b)                    0.687       0.687
## Unique Obs.                   5009        4991
##
## =============================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =============================================================================
##   Conventional   371.567    25.762    14.423     0.000   [321.075 , 422.060]
##         Robust         -         -    13.290     0.000   [311.289 , 418.988]
## =============================================================================
```

```r
## Step 5: Checking robustness: varying the bandwidth and the polynomial order
## bandwidth selection
summary(rdbwselect(y, s, c=10, p=3, covs=x))
```

```
## Call: rdbwselect
##
## Number of Obs.                 10000
## BW type                        mserd
## Kernel                    Triangular
## VCE method                        NN
##
## Number of Obs.                5009        4991
## Order est. (p)                   3           3
## Order bias  (q)                  4           4
## Unique Obs.                   5009        4991
##
## =======================================================
##                   BW est. (h)    BW bias (b)
##             Left of c Right of c  Left of c Right of c
## =======================================================
##      mserd     5.830      5.830      8.485      8.485
## =======================================================
```

```r
## varying the bandwidth
summary(rdrobust(y, s, c=10, p=3, h=1, covs=x))
```

```
## Call: rdrobust
##
## Number of Obs.                 10000
## BW type                       Manual
## Kernel                    Triangular
## VCE method                        NN
##
## Number of Obs.                5009        4991
## Eff. Number of Obs.            804         817
## Order est. (p)                   3           3
## Order bias  (q)                  4           4
## BW est. (h)                  1.000       1.000
## BW bias (b)                  1.000       1.000
## rho (h/b)                    1.000       1.000
## Unique Obs.                   5009        4991
```

```
##
## =============================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =============================================================================
##   Conventional   323.657    63.082     5.131     0.000   [200.018 , 447.295]
##         Robust         -         -     4.206     0.000   [175.818 , 482.728]
## =============================================================================
```

```r
summary(rdrobust(y, s, c=10, p=3, h=5.950, covs=x))
```

```
## Call: rdrobust
##
## Number of Obs.                10000
## BW type                      Manual
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                5009         4991
## Eff. Number of Obs.           3829         3834
## Order est. (p)                   3            3
## Order bias  (q)                  4            4
## BW est. (h)                  5.950        5.950
## BW bias (b)                  5.950        5.950
## rho (h/b)                    1.000        1.000
## Unique Obs.                   5009         4991
##
## =============================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =============================================================================
##   Conventional   372.006    25.520    14.577     0.000   [321.987 , 422.025]
##         Robust         -         -    11.963     0.000   [312.053 , 434.334]
## =============================================================================
```

```r
summary(rdrobust(y, s, c=10, p=3, h=10, covs=x))
```

```
## Call: rdrobust
##
## Number of Obs.                10000
## BW type                      Manual
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                5009         4991
## Eff. Number of Obs.           4755         4757
## Order est. (p)                   3            3
## Order bias  (q)                  4            4
## BW est. (h)                 10.000       10.000
## BW bias (b)                 10.000       10.000
## rho (h/b)                    1.000        1.000
## Unique Obs.                   5009         4991
##
## =============================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =============================================================================
##   Conventional   397.226    20.718    19.173     0.000   [356.619 , 437.833]
```

```
##          Robust          -           -    14.895      0.000    [324.653 , 423.039]
## =============================================================================
```

```r
summary(rdrobust(y, s, c=10, p=3, h=50, covs=x))
```

```
## Call: rdrobust
##
## Number of Obs.                  10000
## BW type                        Manual
## Kernel                     Triangular
## VCE method                         NN
##
## Number of Obs.                  5009         4991
## Eff. Number of Obs.             5009         4991
## Order est. (p)                     3            3
## Order bias  (q)                    4            4
## BW est. (h)                   50.000       50.000
## BW bias (b)                   50.000       50.000
## rho (h/b)                      1.000        1.000
## Unique Obs.                     5009         4991
##
## =============================================================================
##          Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =============================================================================
##    Conventional   399.370    16.532    24.157     0.000    [366.967 , 431.773]
##          Robust          -           -    21.172      0.000    [373.133 , 449.265]
## =============================================================================
```

```r
## varying the polynomial order
summary(rdrobust(y, s, c=10, p=1, h=5.950, covs=x))
```

```
## Call: rdrobust
##
## Number of Obs.                  10000
## BW type                        Manual
## Kernel                     Triangular
## VCE method                         NN
##
## Number of Obs.                  5009         4991
## Eff. Number of Obs.             3829         3834
## Order est. (p)                     1            1
## Order bias  (q)                    2            2
## BW est. (h)                    5.950        5.950
## BW bias (b)                    5.950        5.950
## rho (h/b)                      1.000        1.000
## Unique Obs.                     5009         4991
##
## =============================================================================
##          Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =============================================================================
##    Conventional   405.872    13.937    29.123     0.000    [378.557 , 433.187]
##          Robust          -           -    19.792      0.000    [354.110 , 431.955]
## =============================================================================
```

```r
summary(rdrobust(y, s, c=10, p=3, h=5.950, covs=x))
```

```
## Call: rdrobust
##
## Number of Obs.                 10000
## BW type                       Manual
## Kernel                    Triangular
## VCE method                        NN
##
## Number of Obs.                 5009         4991
## Eff. Number of Obs.            3829         3834
## Order est. (p)                    3            3
## Order bias  (q)                   4            4
## BW est. (h)                   5.950        5.950
## BW bias (b)                   5.950        5.950
## rho (h/b)                     1.000        1.000
## Unique Obs.                    5009         4991
##
## =============================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =============================================================================
##   Conventional   372.006    25.520    14.577     0.000   [321.987 , 422.025]
##         Robust         -         -    11.963     0.000   [312.053 , 434.334]
## =============================================================================
```

```r
summary(rdrobust(y, s, c=10, p=5, h=5.950, covs=x))
```

```
## Call: rdrobust
##
## Number of Obs.                 10000
## BW type                       Manual
## Kernel                    Triangular
## VCE method                        NN
##
## Number of Obs.                 5009         4991
## Eff. Number of Obs.            3829         3834
## Order est. (p)                    5            5
## Order bias  (q)                   6            6
## BW est. (h)                   5.950        5.950
## BW bias (b)                   5.950        5.950
## rho (h/b)                     1.000        1.000
## Unique Obs.                    5009         4991
##
## =============================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =============================================================================
##   Conventional   372.619    37.154    10.029     0.000   [299.798 , 445.439]
##         Robust         -         -     8.127     0.000   [268.006 , 438.369]
## =============================================================================
```