Regression 2: Implementation in R

Instructor: Yuta Toyama

Last updated: 2021-05-18

Introduction

Acknowledgement

This note is based on "Introduction to Econometrics with R". https://www.econometrics-with-r.org/index.html

Preliminary: packages

- We use the following packages:
  - `AER` :
  - `dplyr` : data manipulation
  - `stargazer` : output of regression results

```
# Install package if you have not done so
# install.packages("AER")
# install.packages("dplyr")
# install.packages("stargazer")
# install.packages("texreg")
# install.packages("estimatr")

# load packages
library("AER")
library("dplyr")
library("stargazer")
library("texreg")
library("estimatr")
```

Empirical setting: Data from California School

- Question: How does the student-teacher ratio affects test scores?
- We use data from California school, which is included in `AER` package.
  - See here for the details: https://www.rdocumentation.org/packages/AER/versions/1.2-6/topics/CASchools

```
# load the the data set in the workspace
data(CASchools)
```

- Use `class()` function to see `CASchools` is `data.frame` object.

```
class(CASchools)
```

```
## [1] "data.frame"
```

- We take 2 steps for the analysis.
  - Step 1: Look at data (descriptive analysis)
  - Step 2: Run regression

Step 1: Descriptive analysis

Descriptive analysis

- It is always important to grasp your data before running regression.
- `head()` function give you a first overview of the data.

```
head(CASchools)
```

```
##   district                          school  county grades students teachers
## 1    75119              Sunol Glen Unified Alameda  KK-08      195    10.90
## 2    61499            Manzanita Elementary   Butte  KK-08      240    11.15
## 3    61549     Thermalito Union Elementary   Butte  KK-08     1550    82.90
## 4    61457 Golden Feather Union Elementary   Butte  KK-08      243    14.00
## 5    61523        Palermo Union Elementary   Butte  KK-08     1335    71.50
## 6    62042         Burrel Union Elementary  Fresno  KK-08      137     6.40
##   calworks   lunch computer expenditure    income   english  read  math
## 1   0.5102  2.0408       67    6384.911 22.690001  0.000000 691.6 690.0
## 2  15.4167 47.9167      101    5099.381  9.824000  4.583333 660.5 661.9
## 3  55.0323 76.3226      169    5501.955  8.978000 30.000002 636.3 650.9
## 4  36.4754 77.0492       85    7101.831  8.978000  0.000000 651.9 643.5
## 5  33.1086 78.4270      171    5235.988  9.080333 13.857677 641.8 639.9
## 6  12.3188 86.9565       25    5580.147 10.415000 12.408759 605.7 605.4
```

- Alternatively, you can use `View()` to see the entire dataset in browser window.

Create variables

- Create several variables that are needed for the analysis.
- We use `dplyr` for this purpose.

```
CASchools %>%
  mutate( STR = students / teachers ) %>%
  mutate( score = (read + math) / 2 ) -> CASchools
```

Descriptive statistics

- There are several ways to show descriptive statistics
- The standard one is to use `summary()` function

```
summary(CASchools)
```

```
##    district           school                  county      grades
##  Length:420         Length:420         Sonoma     : 29   KK-06: 61
##  Class :character   Class :character   Kern       : 27   KK-08:359
##  Mode  :character   Mode  :character   Los Angeles: 27
##                                        Tulare     : 24
##                                        San Diego  : 21
##                                        Santa Clara: 20
##                                        (Other)    :272
##     students          teachers         calworks          lunch
##  Min.   :   81.0   Min.   :   4.85   Min.   : 0.000   Min.   :  0.00
##  1st Qu.:  379.0   1st Qu.:  19.66   1st Qu.: 4.395   1st Qu.: 23.28
##  Median :  950.5   Median :  48.56   Median :10.520   Median : 41.75
##  Mean   : 2628.8   Mean   : 129.07   Mean   :13.246   Mean   : 44.71
##  3rd Qu.: 3008.0   3rd Qu.: 146.35   3rd Qu.:18.981   3rd Qu.: 66.86
##  Max.   :27176.0   Max.   :1429.00   Max.   :78.994   Max.   :100.00
##
##     computer       expenditure       income          english
##  Min.   :   0.0   Min.   :3926   Min.   : 5.335   Min.   : 0.000
##  1st Qu.:  46.0   1st Qu.:4906   1st Qu.:10.639   1st Qu.: 1.941
##  Median : 117.5   Median :5215   Median :13.728   Median : 8.778
##  Mean   : 303.4   Mean   :5312   Mean   :15.317   Mean   :15.768
##  3rd Qu.: 375.2   3rd Qu.:5601   3rd Qu.:17.629   3rd Qu.:22.970
##  Max.   :3324.0   Max.   :7712   Max.   :55.328   Max.   :85.540
##
##       read            math            STR            score
##  Min.   :604.5   Min.   :605.4   Min.   :14.00   Min.   :605.5
##  1st Qu.:640.4   1st Qu.:639.4   1st Qu.:18.58   1st Qu.:640.0
```

- This returns the desriptive statistics for all the variables in dataframe.
- You can combine this with `dplyr::select`

```
CASchools %>%
  select(STR, score) %>%
  summary()
```

```
##       STR             score      
##  Min.   :14.00   Min.   :605.5  
##  1st Qu.:18.58   1st Qu.:640.0  
##  Median :19.72   Median :654.5  
##  Mean   :19.64   Mean   :654.2  
##  3rd Qu.:20.87   3rd Qu.:666.7  
##  Max.   :25.80   Max.   :706.8
```

- You can do a bit lengthly thing manually like this.

```
# compute sample averages of STR and score
avg_STR <- mean(CASchools$STR)
avg_score <- mean(CASchools$score)

# compute sample standard deviations of STR and score
sd_STR <- sd(CASchools$STR)
sd_score <- sd(CASchools$score)

# set up a vector of percentiles and compute the quantiles
quantiles <- c(0.10, 0.25, 0.4, 0.5, 0.6, 0.75, 0.9)
quant_STR <- quantile(CASchools$STR, quantiles)
quant_score <- quantile(CASchools$score, quantiles)

# gather everything in a data.frame
DistributionSummary <- data.frame(Average = c(avg_STR, avg_score),
                                  StandardDeviation = c(sd_STR, sd_score),
                                  quantile = rbind(quant_STR, quant_score))
```

```
DistributionSummary
```

```
##               Average StandardDeviation quantile.10. quantile.25. quantile.40.
## quant_STR    19.64043          1.891812      17.3486     18.58236     19.26618
## quant_score 654.15655         19.053347     630.3950    640.05000    649.06999
##             quantile.50. quantile.60. quantile.75. quantile.90.
## quant_STR        19.72321      20.0783     20.87181     21.86741
## quant_score     654.45000     659.4000    666.66249    678.85999
```

- My personal favorite is to use `stargazer` function.

```
stargazer(CASchools, type = "text")
```

```
##
## ===============================================================================
## Statistic    N    Mean    St. Dev.     Min    Pctl(25)  Pctl(75)     Max
## -------------------------------------------------------------------------------
## students    420 2,628.793 3,913.105    81        379      3,008    27,176
## teachers    420  129.067   187.913    4.850    19.662    146.350  1,429.000
## calworks    420  13.246    11.455     0.000     4.395    18.981    78.994
## lunch       420  44.705    27.123     0.000    23.282    66.865    100.000
## computer    420  303.383   441.341      0        46       375.2     3,324
## expenditure 420 5,312.408  633.937  3,926.070 4,906.180 5,601.401 7,711.507
## income      420  15.317     7.226     5.335    10.639    17.629    55.328
## english     420  15.768    18.286       0        1.9      23.0       86
## read        420  654.970   20.108    604.500   640.400   668.725   704.000
## math        420  653.343   18.754      605      639.4     665.8      710
## STR         420  19.640     1.892    14.000    18.582    20.872    25.800
## score       420  654.157   19.053    605.550   640.050   666.662   706.750
## -------------------------------------------------------------------------------
```

- You can choose summary statistics you want to report.

```
CASchools %>%
  stargazer( type = "text", summary.stat = c("n", "p75", "sd") )
```

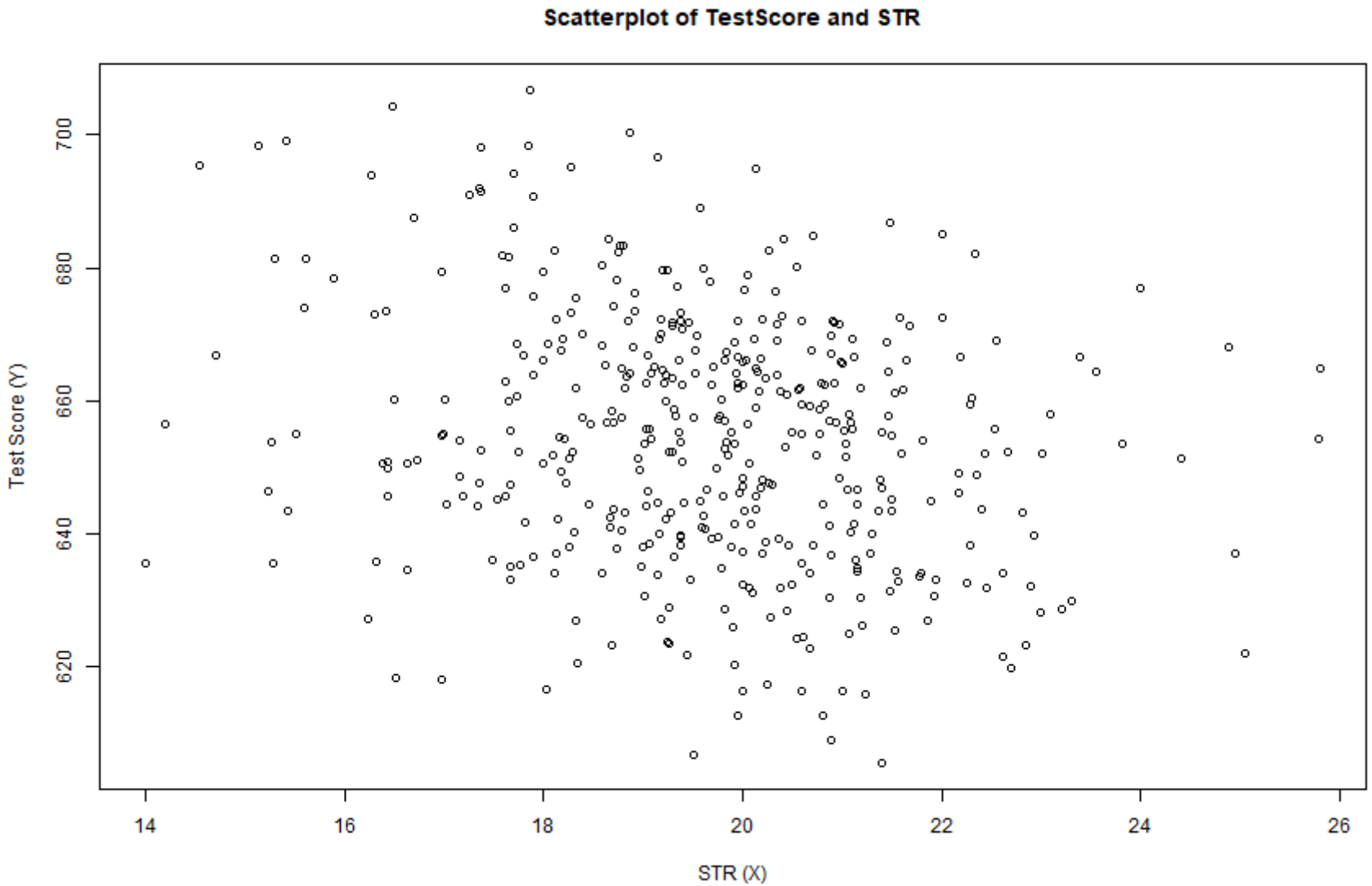
```
##
## ===================================
## Statistic     N  Pctl(75)  St. Dev.
## -----------------------------------
## students    420   3,008   3,913.105
## teachers    420  146.350   187.913
## calworks    420  18.981    11.455
## lunch       420  66.865    27.123
## computer    420   375.2    441.341
## expenditure 420 5,601.401  633.937
## income      420  17.629     7.226
## english     420   23.0     18.286
## read        420  668.725   20.108
## math        420   665.8    18.754
## STR         420  20.872     1.892
## score       420  666.662   19.053
## -----------------------------------
```

- See https://www.jakeruss.com/cheatsheets/stargazer/#the-default-summary-statistics-table for the details.
- `stargazer` can be also used to report regression results.
- But, we will use `texreg` instead.

Scatter plot

- Let's see how test score and student-teacher-ratio is correlated.

```
plot(score ~ STR,
     data = CASchools,
     main = "Scatterplot of TestScore and STR",
     xlab = "STR (X)",
     ylab = "Test Score (Y)")
```


Scatterplot of TestScore and STR
Test Score (Y)
620
640
660
680
700
14
16
18
20
22
24
26
STR (X)

- Use `cor()` to compute the correlation between two numeric vectors.

```
cor(CASchools$STR, CASchools$score)
```

```
## [1] -0.2263627
```

Step 2: Run regression

Simple linear regression

- We use `lm()` function to run linear regression
- First, consider the simple linear regression

$$score_i = \beta_0 + \beta_1 size_i + \epsilon_i$$

  where $size_i$ is the class size (student-teacher-ratio).

  - From now on we call student-teacher-ratio (STR) class size.

- To run this regression, we use `lm`

```
# First, we rename the variable `STR`
CASchools %>%
  dplyr::rename( size = STR) -> CASchools

# Run regression and save results in the varaiable `model1_summary`
model1_summary <- lm( score ~ size, data = CASchools)

# See the results
summary(model1_summary)
```

```
##
## Call:
## lm(formula = score ~ size, data = CASchools)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.727 -14.251   0.483  12.822  48.540
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 698.9329     9.4675  73.825  < 2e-16 ***
## size         -2.2798     0.4798  -4.751 2.78e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Interpretations
  - An increase of one student per teacher leads to 2.2 point decrease in test scores.
  - p value is very small. The effect of the class size on test score is significant.
    - Note: Be careful. These standard errors are NOT heteroskedasiticity robust. We will come back to this point soon.
  - $R^2 = 0.051$, implying that 5.1% of the variance of the dependent variable is explained by the model.
- You can add more variable in the regression (will see this soon)

Robust standard error with `lm_robust`

- We use `lm_robust()` in `estimatr` package to run regression with robust standard error.

```
model1_robust <- lm_robust( score ~ size, data = CASchools, se_type = "HC1")

summary(model1_robust)
```

```
##
## Call:
## lm_robust(formula = score ~ size, data = CASchools, se_type = "HC1")
##
## Standard error type:  HC1
##
## Coefficients:
##             Estimate Std. Error t value   Pr(>|t|) CI Lower CI Upper  DF
## (Intercept)   698.93    10.3644  67.436 9.487e-227  678.560  719.306 418
## size           -2.28     0.5195  -4.389  1.447e-05   -3.301   -1.259 418
##
## Multiple R-squared:  0.05124 ,    Adjusted R-squared:  0.04897
## F-statistic: 19.26 on 1 and 418 DF,  p-value: 1.447e-05
```

- Notice that robust standard errors are larger than the one we obtained from `lm`!

Report by texreg

- `texreg` is useful to show the regression result.
  - `screenreg` function shows the table on R markdown.
  - You can use `htmlreg` (`texreg`) to get html (latex) format.
- `stargazer` is also used to show regression results, however it does not follow `lm_robust`.

```
# Create output by `screenreg` function.
screenreg(l=list(model1_summary, model1_robust),
          digits = 3,
          # caption = 'title',
          custom.model.names = c("model1", "model1 Robust"),
          custom.coef.names = NULL,  # add a class, if you want to change the names of variables.
          include.ci = F,
          include.rsquared = FALSE, include.adjrs = TRUE, include.nobs = TRUE,
          include.pvalues = FALSE, include.df = FALSE, include.rmse = FALSE,
          custom.header = list("score" = 1:2), # you can add header especially to indicate dependen
          stars = numeric(0) # to delete star expression
          )
```

```
##
## ====================================
##                        score
##              ----------------------
##              model1    model1 Robust
## ------------------------------------
## (Intercept)  698.933   698.933
##               (9.467)  (10.364)
## size          -2.280    -2.280
##               (0.480)   (0.519)
## ------------------------------------
## Adj. R^2       0.049     0.049
## Num. obs.    420       420
## ====================================
```

Full results

Taken from https://www.econometrics-with-r.org/7-6-analysis-of-the-test-score-data-set.html

```
# estimate different model specifications
spec1 <- lm_robust(score ~ size, data = CASchools, se_type = "HC1")
spec2 <- lm_robust(score ~ size + english, data = CASchools, se_type = "HC1")
spec3 <- lm_robust(score ~ size + english + lunch, data = CASchools, se_type = "HC1")
spec4 <- lm_robust(score ~ size + english + calworks, data = CASchools, se_type = "HC1")
spec5 <- lm_robust(score ~ size + english + lunch + calworks, data = CASchools, se_type = "HC1")

# generate a table using texreg
screenreg(l = list(spec1, spec2, spec3, spec4, spec5),
          digits = 3,
          # caption = 'title',
          custom.model.names = c("(I)", "(II)", "(III)", "(IV)", "(V)"),
          custom.coef.names = NULL,  # add a class, if you want to change the names of variables.
          include.ci = F,
          include.rsquared = FALSE, include.adjrs = TRUE, include.nobs = TRUE,
          include.pvalues = FALSE, include.df = FALSE, include.rmse = FALSE,
          custom.header = list("score" = 1:2), # you can add header especially to indicate dependen
          stars = numeric(0) # to delete star expression
          )
```

```
##
## ================================================================
##                        score
##                -----------------
##                (I)       (II)      (III)     (IV)      (V)
## ----------------------------------------------------------------
## (Intercept)  698.933   686.032   700.150   697.999   700.392
##              (10.364)   (8.728)   (5.568)   (6.920)   (5.537)
## size          -2.280    -1.101    -0.998    -1.308    -1.014
##               (0.519)   (0.433)   (0.270)   (0.339)   (0.269)
## english                 -0.650    -0.122    -0.488    -0.130
##                         (0.031)   (0.033)   (0.030)   (0.036)
## lunch                             -0.547              -0.529
##                                   (0.024)             (0.038)
## calworks                                    -0.790    -0.048
##                                             (0.068)   (0.059)
## ----------------------------------------------------------------
## Adj. R^2       0.049     0.424     0.773     0.626     0.773
## Num. obs.    420       420       420       420       420
## ================================================================
```

- The coefficient on the class size decreases as we add more explantory variables. Can you explain why? (Hint: omitted variable bias)