Basic Statistical Analysis with R

Tadao Hoshino （星野匡郎）

Econometrics II: ver. 2019 Spring Semester

Reading Data Files

Importing CSV files into **R**

- In order to perform statistical analysis using external datasets, you need to import the data into **R**.
- **R** can read many different data file formats, including
  - csv (Comma-Separated Values) file
  - text file
  - Excel files ($\ast$ You need to install the package "**xlsx**")
- For compatibility with other softwares and ease of editing, csv is the most commonly used format. (csv files can be opened and edited by Excel or any other text editor.)

Importing CSV files into R

- The working directory is the folder where **R** will look for data files and save output files.
- The current working directory can be identified using the `getwd()` command. The default working directory is "My Documents" (∼/Documents).

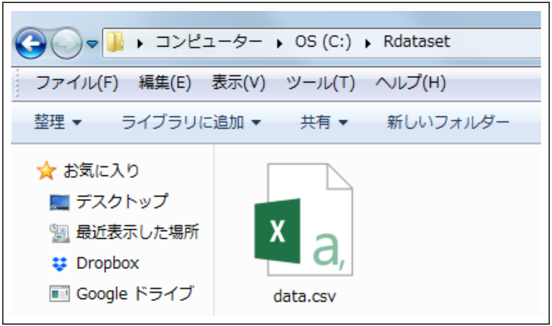
```
Type 'demo()' for some demos, 'help()'
'help.start()' for an HTML browser int
Type 'q()' to quit R.

> getwd()
[1] "C:/Users/hoshino/Documents"
```

- The working directory can be changed using the `setwd()` command:

  `setwd("the location of the new directory")`

Example: importing a csv file into **R**

- Create a new folder (working directory) named, for example, "Rdataset" in the C drive.
- Place the csv file you want to import in the working directory.



∗ A csv file looks quite similar to an Excel file.

Importing CSV files into R

Example: importing a csv file into **R** (cont.)

- Set the working directory to this folder:[1]

```
setwd("C:/Rdataset")
```

- Check whether the working directory is correctly set by `getwd()`.

```
> getwd()
[1] "C:/Users/hoshino/Documents"
> setwd("C:/Rdataset")
> getwd()
[1] "C:/Rdataset"
>
```

[1] Setting working directory can be done manually through the menu bar: [File] $\rightarrow$ [Change dir...] $\rightarrow$ choose your working directory.

Example: importing a csv file into **R** (cont')

- Once the working directory is set, the csv file in the directory can be read by the `read.csv()` command:

```
read.csv("the name of the csv file")
```

- If you type just `read.csv("XXX.csv")` in the console, you can view the data content of the csv file.
    - This is not informative if the data size is big.
- To perform statistical analysis on the imported data, you need to create an **R** object named, for example, "dat" to store the data in **R**.

```
dat <- read.csv("the name of the csv file")
```

Example: importing a csv file into **R** (cont')

```
> dat <- read.csv("data.csv")
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
  cannot open file 'data.csv': No such file or directory
> setwd("C:/Rdataset")
> dat <- read.csv("data.csv")
> |
```

- If the working directory is not correctly specified, the **R** console shows the error message like the above (the texts in blue color).
- You can see all the available files in the working directory using the `list.files()` function.

Descriptive Statistics

Descriptive Statistics of OECD Countries

- A practice data set: **OECD.csv**
  - Data on statistics of some OECD countries.
- The data csv file is available from my website or from **Course Navi**.
- Set your working directory appropriately, and import the csv file by `read.csv()`:

```
setwd("C:/Rdataset")
dat <- read.csv("OECD.csv")
```

Descriptive Statistics of OECD Countries

```
> setwd("C:/Rdataset")
> dat <- read.csv("OECD.csv")
> head(dat)
  Country       POP       GDP     HHEXP EDUEXP MATH
1     AUS 23.126000 1215897.7 656388.3  3.212  494
2     AUT  8.468570  451297.2 217778.4  2.981  497
3     BEL 11.178440  535073.5 256151.6  4.164  507
4     CAN 35.154000 1625347.3 896222.1  3.119  516
5     CZE 10.510720  372257.4 164291.3  2.409  492
6     DNK  5.614932  290376.8 127242.2  4.674  511
> dim(dat)
[1] 35  6
```

- `head()`: displays the first 6 rows of the data.
- `dim()`: returns the dimension of the data (35 observations with 6 variables).

Descriptive Statistics of OECD Countries

Definitions of variables

- **POP** Population in 2013 (million persons).
- **GDP** Total gross domestic product (GDP) in 2016 (million USD).
- **HHEXP** Total household consumption expenditure in 2015 (million USD).
- **EDUEXP** Expenditure on education in 2014 (percentage of GDP).
- **MATH** Mathematics performance (PISA, Programme for International Student Assessment) in 2015.

Commands for descriptive statistics.

- Minimum and maximum of x: `min(`x`)` and `max(`x`)`, respectively.
- Measures of central tendency:
  - Mean of x: `mean(`x`)`
  - Median of x: `median(`x`)`
- Measures of dispersion:
  - Standard deviation of x: `sd(`x`)`
  - Variance of x: `var(`x`)`
- Visualizing the data distribution:
  - Scatterplot x vs. y: `plot(`x`, `y`)`
  - Histogram of x: `hist(`x`)`
- Measures of correlation:
  - Correlation coefficient between x and y: `cor(`x`, `y`)`
  - Covariance between x and y: `cov(`x`, `y`)`

Descriptive Statistics of OECD Countries

```
> max(dat$GDP)
[1] 18707189
> dat$Country[which.max(dat$GDP)]
[1] USA
35 Levels: AUS AUT BEL BRA CAN CHE COL CZE
> GDPpc <- dat$GDP/dat$POP # GDP per capita
> max(GDPpc)
[1] 113890.6
> dat$Country[which.max(GDPpc)]
[1] LUX
35 Levels: AUS AUT BEL BRA CAN CHE COL CZE
```

- R uses a dollar sign (`$`) to refer to a specific variable in the data.
- `which.max()` (`which.min()`) is a function that returns the index of the element with the maximum (minimum) value.

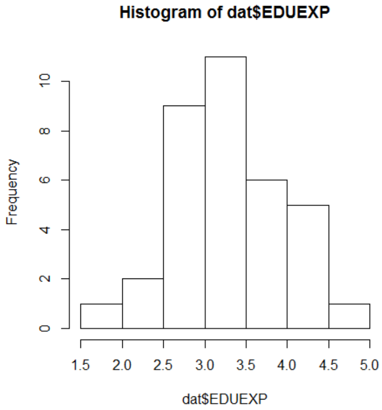
Descriptive Statistics of OECD Countries

```
> mean(dat$HHEXP)
[1] 995890.8
> median(dat$HHEXP)
[1] 273108.5
> sd(dat$HHEXP)
[1] 2038861
> var(dat$HHEXP)
[1] 4.156956e+12
```

- Here, `4.156956e+12` means 4.156956 times 10 to the power 12 (exponential notation).

Histogram of `EDUEXP`:

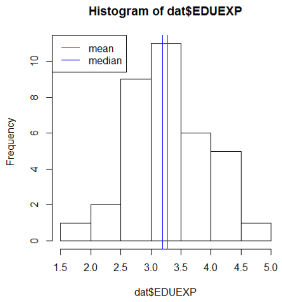
```
hist(dat$EDUEXP)
```



Descriptive Statistics of OECD Countries

- You can add a mean and median line in the histogram.
- Also, a legend can be added using the function `legend()`.

```
abline(v = mean(dat$EDUEXP), col = 2) # col = 2 "red"
abline(v = median(dat$EDUEXP), col = 4) # col = 4 "blue"
legend("topleft", c("mean", "median"), lty = c(1,1), col = c(2, 4))
```



Descriptive Statistics of OECD Countries

```
> cor(dat$EDUEXP, dat$MATH)
[1] 0.01580962
> cov(dat$EDUEXP, dat$MATH)
[1] 0.3720546
> cor(GDPpc, dat$MATH)
[1] 0.4376447
> cov(GDPpc, dat$MATH)
[1] 289654.3
```

- The correlation between the PISA Math score and the amount of educational expenditure is weak.
- The Math score is positively correlated with the GDP per capita.

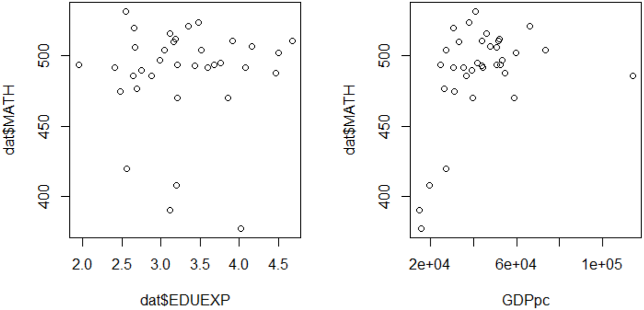
Descriptive Statistics of OECD Countries

Scatterplots of these variables:

```
par(mfrow = c(1,2))
plot(dat$EDUEXP, dat$MATH)
plot(GDPpc, dat$MATH)
```

- In the first line, we split the Graphics window into 1 times 2 sub-windows.
- By default, every time `plot()` is called a new window is created, overwriting the previous plot.

Descriptive Statistics of OECD Countries



Applying the function `summary()` to the data displays the summary of the variables that it contains all at once.

```
> summary(dat[,2:6])
      POP                GDP                HHEXP              EDUEXP           MATH      
 Min.   :  0.3238   Min.   :   17639   Min.   :    7767   Min.   :1.945   Min.   :377.0  
 1st Qu.:  5.5270   1st Qu.:  297352   1st Qu.:  125607   1st Qu.:2.721   1st Qu.:486.0  
 Median : 11.1784   Median :  537701   Median :  273109   Median :3.190   Median :494.0  
 Mean   : 46.2783   Mean   : 1748530   Mean   :  995891   Mean   :3.283   Mean   :487.4  
 3rd Qu.: 61.7313   3rd Qu.: 2201899   3rd Qu.: 1282732   3rd Qu.:3.721   3rd Qu.:508.5  
 Max.   :316.4980   Max.   :18707189   Max.   :11927466   Max.   :4.674   Max.   :532.0  
```

* Note that since the first column of `dat` contains the "name" of each country and is not a variable, it needs to be excluded here.

Linear Regression Analysis

- Outcome variable of interest : dependent variable.
- Variables explaining the variation of the dependent variable : explanatory variables (also referred to as "independent variables" or simply "regressors").

Simple linear regression model

- Linear regression model with a single explanatory variable:

$$Y = \beta_0 + X\beta_1 + \varepsilon$$

- Y: dependent variable, X: explanatory variable, and ε: error term (containing all unobserved determinants of Y).
- β_0: intercept, and β_1: regression coefficient of X. These are the parameters of interest to be estimated.

Multiple linear regression model

- Linear regression model with multiple explanatory variables:

$$\begin{aligned} Y &= \beta_0 + X_1\beta_1 + \cdots + X_k\beta_k + \varepsilon \\ &= \mathbf{X}^\top \beta + \varepsilon, \end{aligned}$$

where $\mathbf{X} = (1, X_1, ..., X_k)^\top$, and $\beta = (\beta_0, \beta_1, ..., \beta_k)^\top$.

Example: Determinants of annual income

- A linear regression model of annual income:

$$\text{Income} = \beta_0 + \text{Experience}\beta_1 + \text{Hours}\beta_2 + \text{Education}\beta_3 + \varepsilon$$

- For example, coefficient β_1 tells us

How much an additional year of working experience affects income,

i.e., $\beta_1 =$ "marginal" effect of Experience variable.

Estimation of β

- Suppose that we have data of n observations $\{(Y_1, \mathbf{X}_1), \ldots, (Y_n, \mathbf{X}_n)\}$.
- The most popular estimator for β is the ordinary least squares (OLS) estimator:

$$\hat{\beta}_n = \underset{b}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (Y_i - \mathbf{X}_i^\top b)^2$$

- The FOC of the minimization problem implies that

$$\underset{(k+1)\times 1}{\mathbf{0}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{X}_i (Y_i - \mathbf{X}_i^\top \hat{\beta}_n)$$

- Rearranging the above equation, we can write the OLS estimator $\hat{\beta}_n$ as

$$\hat{\beta}_n = \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{X}_i \mathbf{X}_i^\top \right)^{-1} \frac{1}{n} \sum_{i=1}^{n} \mathbf{X}_i Y_i.$$

Linear Regression: the price of apartments in Tokyo

- A practice data set: **apartments.csv**
  - Data on individual apartment transactions within Tokyo's 23 wards.
- The data csv file is available from my website or from **Course Navi**.
- Set your working directory appropriately, and import the csv file by `read.csv()`:

```
setwd("C:/Rdataset")
dat <- read.csv("apartments.csv")
```

Linear Regression: the price of apartments in Tokyo

```
> setwd("C:/Rdataset")
> dat <- read.csv("apartments.csv")
> head(dat)
   price  area floor renov    stdist commercial industrial
1 20.038 19.70     3     1 0.3123682          1          0
2 96.300 91.24    23     0 0.3116436          0          1
3 39.300 42.08    13     0 0.2460939          1          0
4 85.600 74.36    15     0 0.4952629          0          0
5  5.700 17.89     6     0 0.8047969          0          0
6 25.200 32.37     8     0 0.5117592          1          0
> dim(dat)
[1] 500   7
> 
```

Linear Regression: the price of apartments in Tokyo

Definitions of variables

Dependent variable (1st column)

- **price** Price of the property (1 mil. JPY)

Explanatory variables (2nd - 7th columns)

- **area** Area of the property (m^2)
- **floor** Floor level of the property.
- **renov** Dummy variable: 1 when the property has a history of renovations; 0 otherwise.
- **stdist** Distance (km) to the nearest railway station.
- **commercial** Dummy variable: 1 when the property is located in a commercially zoned area; 0 otherwise.
- **industrial** Dummy variable: 1 when the property is located in an industrially zoned area; 0 otherwise.

- We estimate a linear regression model defined as follows:

$$\text{price} = \beta_0 + \beta_1\text{area} + \beta_2\text{floor} + \beta_3\text{renov} + \beta_4\text{stdist} + \beta_5\text{commercial} + \beta_6\text{industrial} + error.$$

- To perform a linear regression in **R**, we can use the `lm()` function. The result is saved into an object named, for example, "lm_result".

```
lm_result <- lm(price ~ area + floor + renov + stdist
                + commercial + industrial, dat)
```

or equivalently,

```
lm_result <- lm(price ~ ., dat)
```

- The summary of the estimation results can be displayed by the `summary()` function:

```
summary(lm_result)
```

Linear Regression: the price of apartments in Tokyo

```
> lm_result <- lm(price ~ area + floor + renov + stdist
+ + commercial + industrial, dat)
> summary(lm_result)

Call:
lm(formula = price ~ area + floor + renov + stdist + commercial +
    industrial, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-52.310  -7.245   0.239   6.737  94.330

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.23499    1.78733   2.929  0.00356 **
area         0.58407    0.02755  21.200  < 2e-16 ***
floor        1.09543    0.08516  12.863  < 2e-16 ***
renov       -6.06346    1.46946  -4.126 4.33e-05 ***
stdist      -9.65552    2.26933  -4.255 2.50e-05 ***
commercial  -2.41797    1.38548  -1.745  0.08157 .
industrial  -3.99950    1.57769  -2.535  0.01155 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.98 on 493 degrees of freedom
Multiple R-squared:  0.6929,    Adjusted R-squared:  0.6892
F-statistic: 185.4 on 6 and 493 DF,  p-value: < 2.2e-16

>
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.23499    1.78733   2.929  0.00356 **
area          0.58407    0.02755  21.200  < 2e-16 ***
floor         1.09543    0.08516  12.863  < 2e-16 ***
renov        -6.06346    1.46946  -4.126 4.33e-05 ***
stdist       -9.65552    2.26933  -4.255 2.50e-05 ***
commercial   -2.41797    1.38548  -1.745  0.08157 .
industrial   -3.99950    1.57769  -2.535  0.01155 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- All explanatory variables except **commercial** (which has a t statistic of -1.745) are statistically significant at less than 5% level.
- The variable **commercial** is significant at the 10% level.

Linear Regression: the price of apartments in Tokyo

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.23499    1.78733   2.929  0.00356 **
area         0.58407    0.02755  21.200  < 2e-16 ***
floor        1.09543    0.08516  12.863  < 2e-16 ***
renov       -6.06346    1.46946  -4.126 4.33e-05 ***
stdist      -9.65552    2.26933  -4.255 2.50e-05 ***
commercial  -2.41797    1.38548  -1.745  0.08157 .
industrial  -3.99950    1.57769  -2.535  0.01155 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results imply, for example, that

- 1 m^2 increase in area size increases the property price by about 600,000 JPY.
- One-storey increase in floor level has a positive effect just about 1 mil. JPY.
- 1 km increase in distance to railway station decreases the property price by about 10 mil. JPY.