

# Hands-On Data Analysis for ININ Using R

Prof. Dr. Cornelia Storz, M.Sc. Fei (Michael) Wang

*Management and Microeconomics, Goethe-Universität Frankfurt*

---

This document was prepared for students who are taking ININ course and planning to take the exam. It is a collection of notes and codes for the course. The notes are based on the tutorials we had in the course. I am trying to make it concise and easy to understand. I hope it can help you to review the course and prepare for the exam. *We are living in a very noisy world, therefore let's keep it simple and clear.* I setup a challenge for myself to deliver a clear and concise review notes within 15 pages. This brings the trade-off, which means some figures and tables are not included in the notes. Therefore, you have to run the codes to see the results.

I hope you enjoy reading it. I also hope you will have this notes with you whenever you want to do some data analysis. If one day, you still refer to this notes and find it still useful, I would be very happy to hear that.

*Keywords:* econometrics, data analysis, regression models, empirical research, innovation, management

---

## Contents

<b>1 Introduction</b>	<b>1</b>
<b>2 Introduction to Data and <code>data.table</code></b>	<b>1</b>
2.1 Data Visualization . . . . .	3
2.2 Log Transformation . . . . .	4
<b>3 Simple Linear Regression</b>	<b>5</b>
3.1 Control Variables . . . . .	7
3.2 Interpretation of Regression Results . . . . .	10
3.3 Regression Diagnostics . . . . .	13
<b>4 Multiple Linear Regression</b>	<b>14</b>
<b>5 Logistic Regression</b>	<b>15</b>
<b>6 Please Read the Following Materials</b>	<b>15</b>

## 1 Introduction

All statistical or econometric or machine learning models are based on the following assumptions:


- there are something we know - **data**
- and something we don't know - **error  $\epsilon$** .

In summary, according to confucius, *to know what we know and what we do not know*, that is called **wisdom**. Or like Plato said, *I know that I know nothing*. To help you to review the course, the notes will be organized as follows:

1. **Data**: using `data.table` to get familiar with the data
2. **Simple linear regression**: how to estimate a simple linear regression model, how to interpret the results
3. **Multiple linear regression**: how to estimate a multiple linear regression model, how to interpret the results, how to test the model
4. **Introduction to logistic regression**: why do we need logistic regression
5. **Data manipulation**: will not be tested in the exam, but it is very useful for your future work or research

## 2 Introduction to Data and `data.table`

Broadly speaking, there are two kinds of data: **structured data** and **unstructured data**. Structured data is data that has a structure, such as a table, whereas unstructured data is data that does not have a structure, such as a text file. In this course, we focus on structured data. This means all the data we will use look like tables, such as the following one:



[oceannumeric.github.io](https://oceannumeric.github.io)

```
# create a data.table
dt <- data.table(
  vn1 = c(1, 100, -567),
  vn2 = c("hello", "hello", "hello"),
  vn3 = rep("world", 3)
)
```

```
# read a csv file into data.table
dt<- fread("file_name.csv")
```

`dt[i, j, by]`

any operation on columns takes place at *j*

variable_name_1	variable_name_2	variable_name_3	variable_name_4	vn5	vn7	vn8
integer	numeric (dbl)	character	factor	logic	mixed with missing values	Date/Time
1	2.0	A	female / 1	TRUE	2.0	2017-09-16
100	-3.1415926	"hello"	male / 2	FALSE	"abc"	16:23:57
-567	100	hello world	any categorical data	TRUE	NA	2 June 2020

any operation on rows takes place at *i*

common functions:

```
str(dt)
summary(dt)
names(dt)
dim(dt)
```

```
#save a data.table into a csv file
fwrite(dt, "file_name.csv")
```

The basic syntax of `data.table` is summarized in the following illustration. **You will not be tested on the syntax of `data.table` in the exam.** However, you will be tested on the underlying concepts of

data.table, such as the type of variables (integer, character, factor, etc.). In the future if you will be working as a data scientist, you can use data.table to do big data analysis. You will need to know the syntax of data.table for practical use not for the exam.

#### Import key packages

```
library(data.table)
library(magrittr)
library(knitr)
library(ggplot2)
```

#### Structure of the dataset

```
str(dt)
head(dt)
summary(dt)
names(dt)
setnames('old', 'new')
setorder(vn5, vn6)
apply(dt, function(x) sum(is.na(x)))
```

#### Check unique or duplicated values

```
dt %>%
  unique(by = c("variables"))
dt %>%
  .[duplicated(variable)]
# print out all duplicates
dt %>%
  .[duplicated(variable) | duplicated(variable, fromLast = TRUE)]
```



#### dt[i, j, by]

```
# common functions in pipe
%>%
with()
kable()
plot()
par(mfrow = c(2, 2))

# ask ChatGPT always
```

dt						
variable_name_1	variable_name_2	variable_name_3	variable_name_4	vn5	vn7	vn8
integer	numeric (dbl)	character	factor	logic	mixed with missing	Date/Time
1	2.0	A	female / 1	TRUE	2.0	2013
100	-3.1415926	hello	male / 2	FALSE	"abc"	2022-09-10
-567	100	hello world	any categorical data	TRUE	NA	09:12:37

```
# class
class(variable)
# is function
is.factor()
is.integer()
is.character()
# as function
as.character()
as.integer()
as.POSIXct()
as.factor()
...
```

#### Manipulate rows with i

```
# extract rows based on index
dt[5:17, ] # all columns
dt[1:9, 2:4] # row 1 to 9, column 2 to 4
# subset rows based on conditions
dt %>%
  .[vn2 >= 20]

# logical operators to use in i
> < >= <= is.na() !
& | %in% %like% %between%

# any functions from dplyr
# could be combined within
# the pipe line
```

#### Manipulate columns with j

```
# extract columns
dt %>%
  .[, .(vn5, vn7)]
dt %>%
  .[, c(2:6)] # using column index
# extract columns based on names
dt %>%
  .[, .SD, .SDcols = patterns("^a")]
# extract columns based on type
dt %>%
  .[, .SD, .SDcols = is.integer]
# extract and transform at the same time
dt %>%
  .[, lapply(.SD, tolower), .SDcols = is.character]
# create a new columns on original data
dt %>%
  .[, vn9 := vn2 + 2]
# create or transform columns on original data
# using name vector, .SDcols, and lapply with :=
```

#### Subgroup with by

```
# summarize vn2 by vn4
dt %>%
  .[, .(vn2_mean = mean(vn2)), by = vn4]

# one of the most common way to use by
# is that we need to do some operation on one
# or several variables based on another
# categorical variables, such as
- dt[, .(c=sum(b)), by = a]
- dt[, .(c=mean(b)), by = .(a, d)]

# use keyby if you want to
# sort within the group

# special functions that
# could bring magics
.N, .I .SD
```

Let's start from installing and loading the packages we need for the course.

```
1 # install packages
2 install.packages("stargazer")
3 # install ISLR if you don't have it
4 # install.packages("ISLR")
5 install.packages("corrplot")
6 # sometimes you need to install other packages
7 # hopefully you can figure it out by yourself
```

After you install the packages, you can load them into R.

```
1 # library for data analysis
2 library(data.table)
3 library(magrittr)
4 library(ggplot2)
5 library(knitr)
6 library(stargazer)
7 library(MASS)
8 library(ISLR)
9 library(corrplot)
```

Now, we can load the data into R and manipulate the data.

```
1 # read the dataset
2 cis <- fread("https://shorturl.at/wBESZ")
3 # check structure of the dataset
4 str(cis)
5 # check the first 10 rows of the dataset
6 head(cis, 10)
```

```

7 # check the number of missing values in each column
8 cis %>%
9   .[, .SD, .SDcols = is.double] %>%
10  # check the number of missing values in each column
11  apply(function(x) sum(is.na(x))) %>%
12  # sort the number of missing values in each column
13  sort(decreasing = TRUE) %>%
14  # convert to data.table and keep rownames as a column
15  as.data.table(keep.rownames = TRUE) %>%
16  # set variable names
17  setnames(c("variables", "Numbe of NAs")) %>%
18  # check the first 10
19  head(10) %>%
20  kable("pipe")

```

## 2.1 Data Visualization

It is important to visualize the data before you start to do the analysis. To choose the right figure, you need to know the type of variables. Here is the summary:

- **Categorical variables:** bar chart, pie chart, etc.
- **Continuous variables:** histogram, boxplot, etc.
- **Categorical vs. continuous variables:** boxplot or violin or histogram with different colors

Here is a demo of how to visualize the data. You can try to run the code and see the results.

```

1 # four figures in one page
2 # bar chart, histogram, box plot and box plot compared
3 # figure size
4 options(repr.plot.width = 10, repr.plot.height = 10)
5 par(mfrow = c(2, 2))
6 cis %>%
7   # group by branche and .N calculates the number of frequency
8   .[, .N, by = branche] %>%
9   # order it in a descending way
10  .[order(-N)] %>%
11  # get the top 10 branche (industry)
12  head(10) %>%
13  # plot it
14  with(pie(N, labels = branche, main = "Distribution of Industries"))
15
16 # histgorma for number of employees
17 cis %>%
18   with(hist(bges, main = "number of employees"))
19 # boxplot for sales
20 cis %>%
21   with(boxplot(um18, main = "Boxplot of Sales in 2018"))
22 # boxplot for log(1+sales)
23 cis %>%
24   with(boxplot(log(1+um18), main = "Boxplot of Log Transform "))

```

When you visualize the data, it is important to check two things for continuous variables:

- **shape:** whether the distribution is symmetric or skewed (ideally, we prefer symmetric distribution)
- **outliers:** outliers are extreme values that deviate from other observations on data, they may indicate a variability in a measurement, experimental errors or a novelty.

## 2.2 Log Transformation

Log transformation is a very useful tool to deal with skewed data. When you have a skewed data, you can try to log transform it. However, it could be tricky. You need to know the underlying theory of log transformation. Generally speaking, log transformation is used to make the data more symmetric. To avoid the negative values, we usually use  $\log(1+x)$  instead of  $\log(x)$ .

```
1 # log transform
2 options(repr.plot.width = 10, repr.plot.height = 10)
3 par(mfrow = c(2, 2))
4 cis %>%
5   with(hist(um18, main = "Histogram of Sales (2018)"))
6
7 cis %>%
8   with(hist(log(1+um18), main = "Histogram of Log (1+sales)"))
9
10 cis %>%
11   with(boxplot(um18, main = "Boxplot of Sales (2018)"))
12 cis %>%
13   with(boxplot(log(1+um18), main = "Boxplot of Log Transform "))
```

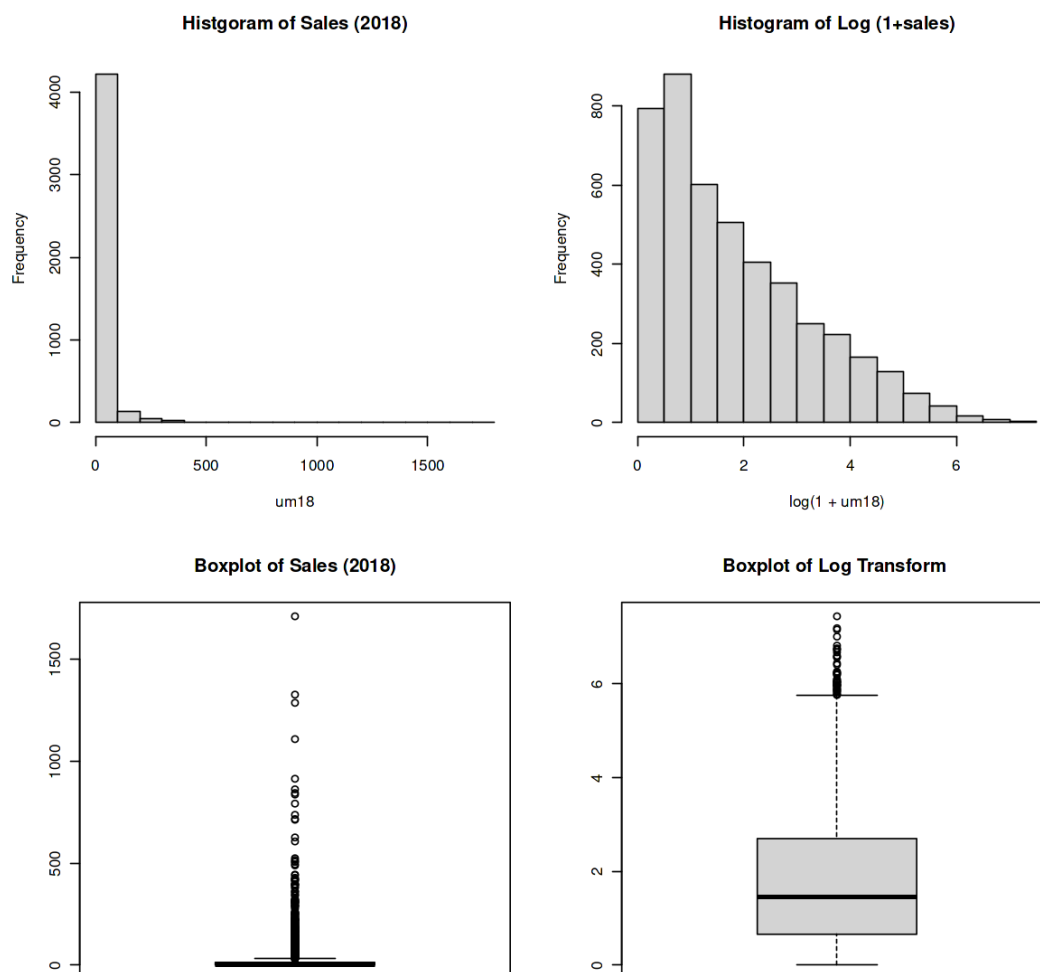


Figure 1: Log transformation of sales

When we fit a linear regression model, if dependent variable is continuous, we prefer the dependent variable to be normally distributed. So, we need to check whether the dependent variable is normally distributed. If not, we could use some transformation to make it normally distributed. As we have covered in the tutorials, there are two main distributions you need to know:

- normal distribution (Gaussian distribution)
- binomial distribution

Possion distribution will not be tested in the exam.

## Binomial to Poisson

[oceanumeric.github.io](https://oceanumeric.github.io)

**Binomial** discrete `dbinom(30, 100, 0.72)`

**Poisson** discrete `dpois(x, lambda)`

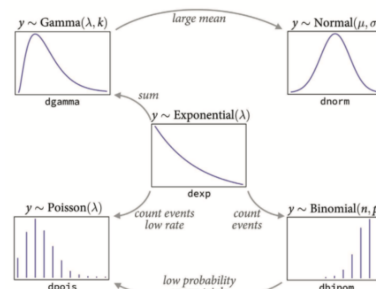
**Gaussian** continuous `dnorm(u, sigma)`

**Parameter:**  $\mu, \sigma$

**Inference:** k

**Example:** what's probability of having 93 customer coming to my restaurant?

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



$$\begin{aligned} P(X=x) &= \lim_{n \rightarrow \infty} \binom{n}{x} \left(\frac{\lambda}{n}\right)^x \left(1 - \frac{\lambda}{n}\right)^{n-x} \\ &= \lim_{n \rightarrow \infty} \frac{n!}{x!(n-x)!} \left(\frac{\lambda}{n}\right)^x \left(1 - \frac{\lambda}{n}\right)^{n-x} \\ &= \lim_{n \rightarrow \infty} \frac{n(n-1)(n-2)\dots(n-x+1)}{x!} \left(\frac{\lambda}{n}\right)^x \left(1 - \frac{\lambda}{n}\right)^{n-x} \\ &= \lim_{n \rightarrow \infty} \frac{n(n-1)(n-2)\dots(n-x+1)}{n^x} \left(\frac{\lambda}{n}\right)^x \left(1 - \frac{\lambda}{n}\right)^{n-x} \\ &= \lim_{n \rightarrow \infty} \frac{n(n-1)(n-2)\dots(n-x+1)}{n^x} \left(\frac{\lambda}{n}\right)^x \left(1 - \frac{\lambda}{n}\right)^n \left(1 - \frac{\lambda}{n}\right)^{-x} \\ &= e^{-\lambda} \frac{\lambda^x}{x!} \end{aligned}$$

$$P(x=k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

**Parameter:**  $\lambda$

**Inference:** k

**Example:** what's probability of having 93 customer coming to my restaurant from 10:05 to 10:20?

$$P(X=k) = \binom{n}{k} p^k (1-p)^{n-k}$$

**Parameter:** p

**Inference:** n, k

**Example:** I have 160 people, what's the probability of having 93 of them will come to my restaurant?

Notes or R-code

## 3 Simple Linear Regression

In our tutorials, we follow the following analytic framework:

	Number of Variables	Focus
Univariate	1	Distribution: Gaussian or Other distributions
Bivariate	2	Correlation or contingency table
Multivariate	3 or more	Regression or classification

Allow me to quote again:

*Dao begets One (such as emptyness, or existence, or the individual), One begets Two (yin and yang, or the binary), Two begets Three (the trinity, or heaven, earth and Huamn beings), and Three begets all things(the world). – Dao De Jing of Lao Tzu*

Many exam questions will be related to regression analysis. So, please pay attention to this section and the next one too. Before we run a regression analysis, let's first understand the basic concepts of regression analysis via simulation.

We know that the relationship between weight and height is roughly linear and positive. We will use this relationship to simulate the data. When we simulate the data, we will add some random noise to the data as there is no perfect linear relationship between weight and height. To make you understand the concept, I will simulate the data step by step:

- scenario 1: no noise (perfect linear relationship)
- scenario 2: add some noise (not perfect linear relationship)
- scenario 3: add some outliers (not perfect linear relationship)

Please run the following code and check the results. You could only learn those concepts by doing it yourself.

```

1 # simulate weight and height
2 # generate 900 random numbers from normal distribution
3 # mean = 175cm and sd = 10cm
4 height <- rnorm(900, mean = 175, sd = 10)
5 # plot histogram
6 hist(height, breaks = 20, xlab="Height (cm)",
7       main="Histogram of Height")

```

Now, we have the height, we assume that there is a linear relationship between height and weight, which has the following form:

$$weight = \beta_0 + \beta_1 \times height$$

Here we set  $\beta_0 = 55$  and  $\beta_1 = 0.1$ . This means that if the height increases by 1 cm, the weight will increase by 0.1 kg.

$$weight = 55 + 0.1 \times height$$

It has the format:

$$y = b + ax$$

```

1 # generate weight
2 # CHANGE THE NUMBER AND PLAY WITH IT :)
3 weight <- 55 + 0.1 * height
4 # plot scatter plot
5 plot(height, weight, main = "Simulated Data Without Noise")
6 # plot histogram of weight
7 hist(weight, breaks = 20, xlab="weight (kg)")
8 # now let's fit a linear regression model
9 sm1 <- lm(weight ~ height)
10 stargazer(sm1, type = "text")

```

```

1 # put the fitted line into the plot
2 plot(height, weight)
3 abline(sm1, col='red')

```

Now, we will add some noise to the data. We will add some random noise to the weight. The random noise is generated from a normal distribution with mean 0 and standard deviation 2.

$$weight = 55 + 0.1 \times height + \varepsilon; \quad \varepsilon \sim N(0,2)$$

```

1 # add some noise to weight
2 # weight = 55 + 0.1 * height
3 # weight2 = weight + noise (rnorm)
4 weight2 <- weight + rnorm(900, mean = 0, sd = 2)
5 # plot scatter plot
6 plot(height, weight2, main = "Simulated Data With Noise")
7 # now we will fit linear regression with weight2 ~ height
8 sm2 <- lm(weight2 ~ height)

```

```

9
10 # print out the model
11 stargazer(m1, sm2, type = "text")

```

Table 1: Regression Results for Two Modles

	<i>Dependent variable:</i>	
	weight (1)	weight2 (2)
height	0.100*** (0.000)	0.103*** (0.007)
Constant	55.000*** (0.000)	54.556*** (1.155)
Observations	900	900
R <sup>2</sup>	1.000	0.212
Adjusted R <sup>2</sup>	1.000	0.212
Residual Std. Error (df = 898)	0.000	1.950
F Statistic (df = 1; 898)	1,887,185,150,343,425,038,960,868,458,496.000***	242.239***

Note:

\*p<0.1; \*\*p<0.05; \*\*\*p<0.01

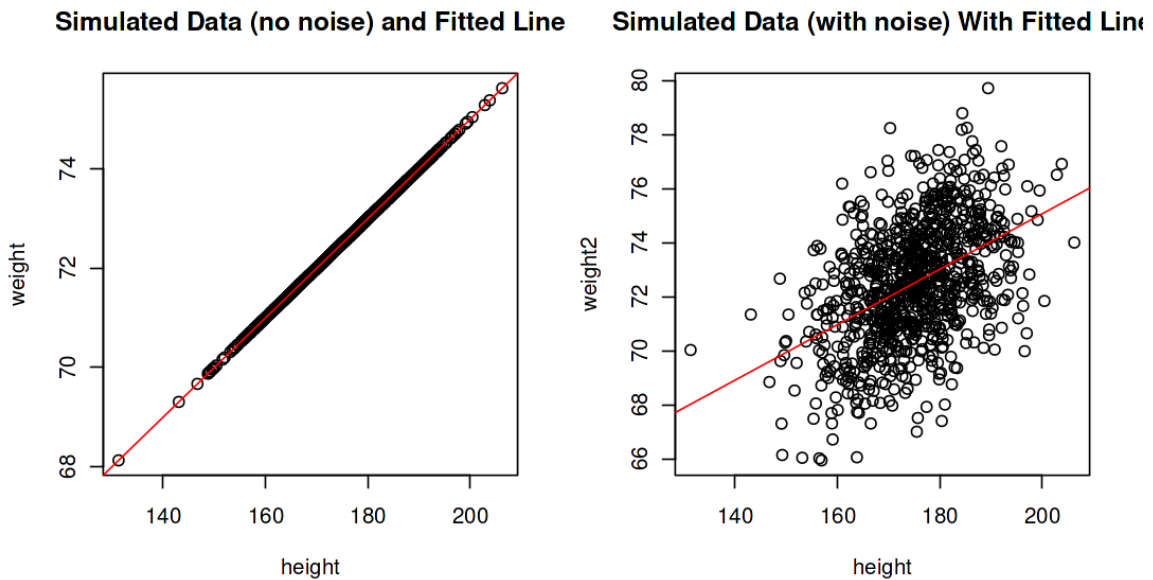


Figure 2: Simulated Data Without or With Noise

### 3.1 Control Variables

Now, let's add some control variables to the model. We will add a categorical variable called gender:

- for female, the distribution of height might be different from male



- the relationship between height and weight is different for female and male

```

1 # generate height for female
2 height_female <- rnorm(450, 170, 5)
3 female_character <- rep("female", 450)
4 height_male <- rnorm(450, 175, 10)
5 male_character <- rep('male', 450)
6 # put them together as a data.frame and then conver it to the data.
  table
7 sim_data <- data.frame(height = c(height_female, height_male), gender =
  c(female_character, male_character))
8 sim_data <- as.data.table(sim_data)
9 head(sim_data)
10
11 # add weight
12 sim_data %>%
13   #[i, j, by]
14   .[, weight := ifelse(gender=="female",
15                        50 + 0.09 *height,
16                        55 + 0.1 * height)] -> sim_data2
17
18 str(sim_data2)

```

Let's review what we have done:

1. simulate one variable (height, follows the normal distribution)

$$height \sim N(175, 10)$$

2. assume there is a perfect linear relationship between weight and height such as

$$weight = 55 + 0.1 * height$$

3. add noise into the data because there is no perfect thing in the real life

$$weight = 55 + 0.1 * height + \varepsilon; \quad \varepsilon \sim N(0, 2)$$

4. bring one more variable into our analysis, let's say gender (female/male)

5. for female, the distribution of height might be different from male

$$height_f \sim N(170, 5); \quad height_m \sim N(175, 10)$$

6. the relationship between height and weight is different for female and male

$$weight_f = 50 + 0.09 * height_f; \quad weight_m = 55 + 0.1 * height_m$$

You can see that the complexity has already kicks in even for onlhy three variables. Please run the following code and take a look at the figure.

```

1 sim_data2 %>%
2   ggplot(aes(x=height, y=weight, color=gender)) +
3   geom_point()

```

Table 2: Linear Regression Results

	<i>Dependent variable:</i>	
	weight	
	(1)	(2)
height	0.222*** (0.012)	0.098*** (0.0001)
gendermale		6.713*** (0.002)
Constant	30.472*** (2.163)	48.603*** (0.022)
Observations	900	900
R <sup>2</sup>	0.261	1.000
Adjusted R <sup>2</sup>	0.260	1.000
Residual Std. Error	3.189 (df = 898)	0.031 (df = 897)
F Statistic	317.177*** (df = 1; 898)	6,389,261.000*** (df = 2; 897)

*Note:* \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

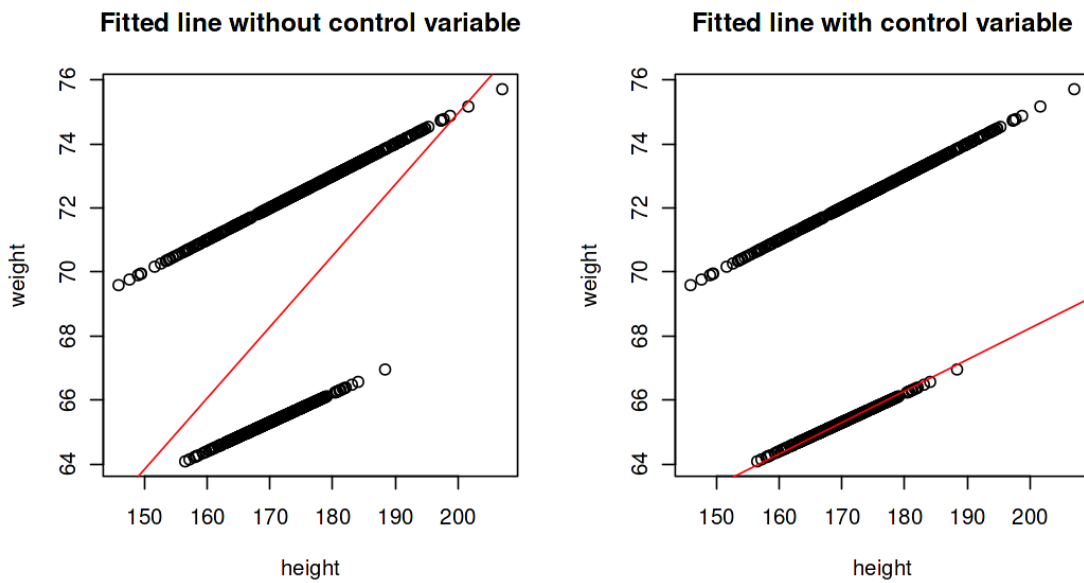


Figure 3: Simulated Data and Fitted Regression Line

Now, we will fit two regression models, one is simple linear regression model:

$$weight = \beta_0 + \beta_1 height + \varepsilon; \quad \varepsilon \sim N(0, sd)$$

and the other one is multiple linear regression model (with a control variable):

$$weight = \beta_0 + \beta_1 height + \beta_2 gender + \varepsilon$$

As you can see, the simple linear regression model is not good enough to capture the relationship between weight and height. The multiple linear regression model is better than the simple linear regression model.

### 3.2 Interpretation of Regression Results

It is very important to know how to interpret the regression analysis results. Again, here we are not talking about the causal relationship, but the association between the dependent variable and independent variables. We will use an example to show you.

The dataset we will explore is about the relationship between wage and education. Based on our common sense, it is likely that the more education is usually **associated** with higher wage.

```
1 # install a new package called wooldridge
2 install.packages("wooldridge")
3 library(wooldridge)
4 # load the data
5 data("wage1")
6 # convert it to data.table
7 wage1 <- as.data.table(wage1)
8 head(wage1)
```

As we can see that we have many variables. However, however we are mainly interested in the relationship between wage and education, so we will only focus on these two variables and other control variables such as:

- wage: average hourly earnings
- educ: years of education
- exper: years of experience
- female: =1 if female otherwise =0

```
1 # univariate analysis
2 wage1 %>%
3   with(hist(wage))
4 # bivariate analysis
5 wage1 %>%
6   with(plot(educ, wage))
7 # run regression
8 wage_reg1 <- lm(wage ~ educ, data=wage1)
9 stargazer(wage_reg1, type="text")
10 # bivariate: experience and wage
11 wage1 %>%
12   with(plot(exper, wage))
13 # let's run another regression
14 wage_reg2 <- lm(wage ~ educ + exper, data=wage1)
15 stargazer(wage_reg2, type="text")
16 # let's add non-linear term
17 wage_reg3 <- lm(wage ~ educ + exper + I(exper^2), data=wage1)
18 stargazer(wage_reg1, wage_reg2, wage_reg3, type="text")
```

Table 3: Regression Models for Wage

	<i>Dependent variable:</i>		
	wage		
	(1)	(2)	(3)
educ	0.541*** (0.053)	0.644*** (0.054)	0.595*** (0.053)
exper		0.070*** (0.011)	0.268*** (0.037)
exper <sup>2</sup>			−0.005*** (0.001)
Constant	−0.905 (0.685)	−3.391*** (0.767)	−3.965*** (0.752)
Observations	526	526	526
R <sup>2</sup>	0.165	0.225	0.269
Adjusted R <sup>2</sup>	0.163	0.222	0.265
Residual Std. Error	3.378 (df = 524)	3.257 (df = 523)	3.166 (df = 522)
F Statistic	103.363*** (df = 1; 524)	75.990*** (df = 2; 523)	64.109*** (df = 3; 522)

Note:

\*p&lt;0.1; \*\*p&lt;0.05; \*\*\*p&lt;0.01

With this model, here is how we will interpret the results: holding other factors constant, an increase in education of 1 year is associated with an increase of 0.595 dollar an hour in wage. The coefficient is significant at 1% level. For experience, we can say that holding other factors constant, there is a nonlinear relationship between experience and wage. The wage will increase with experience, but it will stop after reaching a certain level.

The coefficients for *exper* and *exper*<sup>2</sup> are 0.268 and 0.005, now let's plot the relationship between wage and *exper* holding other factors constant. This means we can have the following equation:

$$\text{wage} = 0.268 \times \text{exper} - 0.005 \times \text{exper}^2 \quad (\text{wage-equation (1)})$$

```

1 # simulate experience from 0 to 40 years
2 # seq = sequence generated from 0 to 30 with interval 1
3 exper_seq <- seq(0, 40, 1)
4 # ^2 means square
5 wage_sim <- 0.268 * exper_seq - 0.005 * exper_seq^2
6 # plot the relationship
7 plot(exper_seq, wage_sim, type="l")
8 # add vertical line
9 abline(v=27, col='red')
10 # we can put them together
11 wage1 %>%
12   with(plot(exper, wage)) +
13     lines(exper_seq, wage_sim, type="l", col='red')
14 # let's simulate other equation
15 wage_sim2 <- 10 + 0.268 * exper_seq - 0.005 * exper_seq^2
16

```

```

17 # then we plot it again
18 wage1 %>%
19   with(plot(exper, wage, main = "Wage and Experience with Equation
      (2)")) +
20   lines(exper_seq, wage_sim2, type="l", col='red') +
21   lines(exper_seq, wage_sim, type="l", col='blue')

```

Why the red curve above does not fit with the dataset exactly? Wage is real-life data, it is determined:

- educ
- experience
- industry
- networking
- etc.

Here we only plot the relationship between wage and exper without considering other factors. Be aware that wage is the average hourly earnings, which is from the real data. wage is determined by many factors, such as education and industry. Now, imagine let's assume the wage was determined by the following equation:

$$wage = 10 + 0.268 \times exper - 0.005 \times exper^2 \quad (\text{wage-equation (2)})$$

This means that we have nonlinear relationship between wage and experience. The wage will increase with experience, but it will stop after reaching a certain level. Then, no matter what kind of industry you are in, or degree you have, the relationship between wage and experience will be the same and everyone will be added 10 dollars per hour as a constant.



Figure 4: Nolinear relationship illustration

We have controlled the experience, here is the short summary of the regression results.

The regression results are not causal, but they are useful for us to understand the relationship between dependent variable and independent variables. Here we can be very confident to say that holding other factors constant, there is a very strong positive association between education and wage. The reason is that the coefficient of education did not change much when we add more control variables (such as experience). This means whether for people who have more experience or not, the education is still positively associated with wage.

Now, how about the gender? Does the relationship still hold for different genders? Let's run another regression analysis.

```

1 # add gender in the regression
2 wage_reg4 <- lm(wage ~ educ + exper + I(exper^2) + female, data=wage1)
3 stargazer(wage_reg4, type="text")

```

Table 4

	<i>Dependent variable:</i>		
	wage		
	(1)	(2)	(3)
educ	0.644*** (0.054)	0.595*** (0.053)	0.556*** (0.050)
exper	0.070*** (0.011)	0.268*** (0.037)	0.255*** (0.035)
I(exper^2)		−0.005*** (0.001)	−0.004*** (0.001)
female			−2.114*** (0.263)
Constant	−3.391*** (0.767)	−3.965*** (0.752)	−2.319*** (0.739)
Observations	526	526	526
R <sup>2</sup>	0.225	0.269	0.350
Adjusted R <sup>2</sup>	0.222	0.265	0.345
Residual Std. Error	3.257 (df = 523)	3.166 (df = 522)	2.989 (df = 521)
F Statistic	75.990*** (df = 2; 523)	64.109*** (df = 3; 522)	70.170*** (df = 4; 521)

Note:

\*p<0.1; \*\*p<0.05; \*\*\*p<0.01

Here we notice that the coefficient for female is  $-2.114$ , which means holding other factors constant, women is associated with a decrease of 2.114 dollar an hour in wage comparing to men. This means even with same education, experience, there is still negative association between being female and wage. Therefore, we can say this might be due to the gender discrimination in the labor market.

### 3.3 Regression Diagnostics

Robustness check is a very important concept in regression analysis. It is very important to check whether the results are robust to different specifications. For instance, we can run the regression analysis with different control variables. If the results are robust, then we can be more confident about the results.

After running the regression analysis, we need to check whether the results are reliable. There are many ways to check the reliability of the results. Here we will introduce three ways to check the reliability of the results:

- residual plot: the residual plot is used to check whether the residuals are randomly distributed. If the residuals are randomly distributed, then we can say the results are reliable. Otherwise, we need to check the model specification. For instance, we might need to add more control variables to the model.

- VIF: VIF is used to check whether there is multicollinearity in the model. If the VIF is larger than 10, then we need to check whether there is multicollinearity in the model. If there is multicollinearity, then we need to remove some variables from the model.
- influential points: influential points are extreme values that deviate from other observations on data, they may indicate a variability in a measurement, experimental errors or a novelty. If there are some influential points, then we need to check whether they are correct. If they are correct, then we need to run the regression analysis again without those influential points.

Please run the following code and check the results.

```

1 install.packages("performance")
2 install.packages("see")
3 install.packages("patchwork")
4 install.packages("wooldridge")
5 library(performance)
6 library(wooldridge)
7
8 # high influential points
9 data("infmrt")
10 str(infmrt)
11 infmrt <- as.data.table(infmrt)
12 infmrt %>%
13   .[lphysic ≤ 6.0] %>%
14   .[, .(infmort, lpcinc, lphysic, lpopul)] -> infmrt2
15
16 head(infmrt2)
17 infmrt %>%
18   with(plot(lphysic, infmort))
19 infmrt_reg1 <- lm(infmort ~ lpcinc + lphysic + lpopul, data = infmrt)
20 stargazer(infmrt_reg1, type="text")
21 check_model(infmrt_reg1)
22 infmrt_reg2 <- lm(infmort ~ lpcinc + lphysic + lpopul, data = infmrt2)
23 stargazer(infmrt_reg2, type="text")
24 check_model(infmrt_reg2)
25
26 # multicollinearity
27 elem_data <- fread("https://shorturl.at/awLNT")
28 head(elem_data)
29 str(elem_data)
30 elem_reg1 <- lm(api00 ~ acs_k3 + avg_ed + grad_sch + col_grad + some_
31   col, data = elem_data)
32 stargazer(elem_reg1, type='text')
33 options(repr.plot.width = 11, repr.plot.height = 8)
34 check_model(elem_reg1)

```

## 4 Multiple Linear Regression

To summarize what we have learned, we have the following regression models:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon \quad (1)$$

where

- $y$  is the dependent variable
- $x_1, x_2, \dots, x_k$  are independent variables
- $\beta_0, \beta_1, \dots, \beta_k$  are coefficients

- $\varepsilon$  is the error term

We assume the following assumptions.

Assumptions	Diagnostic check
A1: linear relationships between $y$ and each $x$	check via plots
A2: Independence of observations	check via plots
A3: $E(\varepsilon x) = 0$	check via plots
A4: $Var(\varepsilon x) = \sigma^2$	check via plots
A5: Normality of the error $\varepsilon \sim N(0, \sigma^2)$	check via plots
A6: No correlation between $x$ and $\varepsilon$ : $cor(x, \varepsilon) = 0$	serial correlation test

Table 5: A summary of model assumptions and regression diagnostic

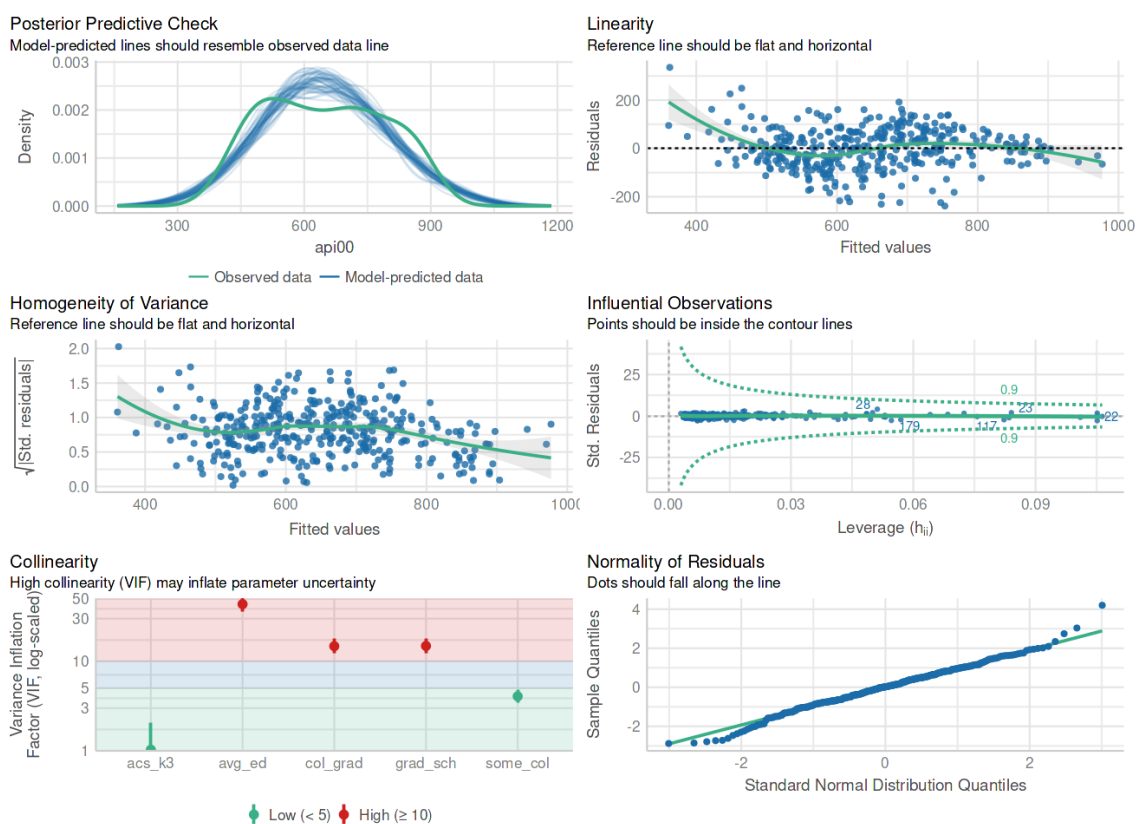


Figure 5: Regression Diagnostics Plots

## 5 Logistic Regression

You will not be tested on logistic regression in the exam. However, it is very important to know when the dependent variable is not continuous. For instance, if the dependent variable is binary, then we can use logistic regression.

You can follow this link <https://davidalpiaz.github.io/r4sl/logistic-regression.html> to learn more about logistic regression.

## 6 Please Read the Following Materials