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

1 Introduction

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

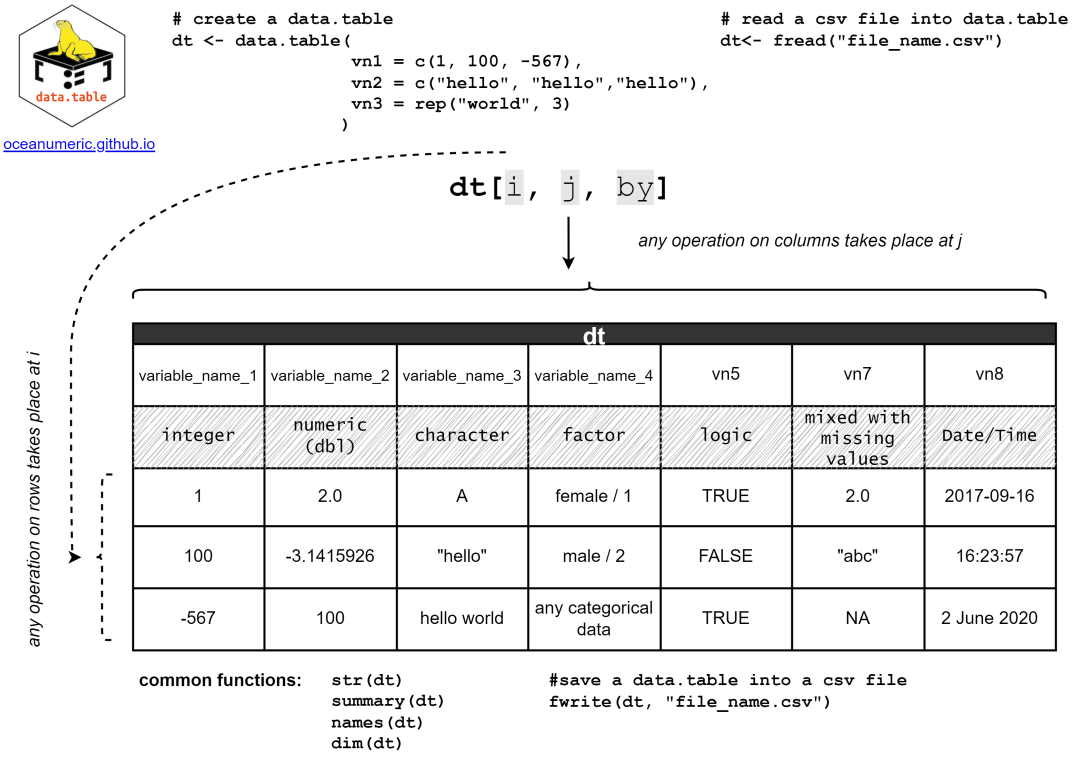
- there are something we know - **data**
- and something we don't know - **error** ε.

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:



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.

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

```r
# install packages
install.packages("stargazer")
# install ISLR if you don't have it
# install.packages("ISLR")
install.packages("corrplot")
# sometimes you need to install other packages
# hopefully you can figure it out by yourself
```

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

```r
# library for data analysis
library(data.table)
library(magrittr)
library(ggplot2)
library(knitr)
library(stargazer)
library(MASS)
library(ISLR)
library(corrplot)
```

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

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

```r
# check the number of missing values in each column
cis %>%
    .[, .SD, .SDcols = is.double] %>%
    # check the number of missing values in each column
    sapply(function(x) sum(is.na(x))) %>%
    # sort the number of missing values in each column
    sort(decreasing = TRUE) %>%
    # convert to data.table and keep rownames as a column
    as.data.table(keep.rownames = TRUE) %>%
    # set variable names
    setnames(c("variables", "Numbe of NAs")) %>%
    # check the first 10
    head(10) %>%
    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.

```r
# four figures in one page
# bar chart, histogram, box plot and box plot compared
# figure size
options(repr.plot.width = 10, repr.plot.height = 10)
par(mfrow = c(2, 2))
cis %>%
    # group by branche and .N calculates the number of frequency
    .[, .N, by = branche] %>%
    # order it in a descending way
    .[order(-N)] %>%
    # get the top 10 branche (industry)
    head(10) %>%
    # plot it
    with(pie(N, labels = branche, main = "Distribution of Industries"))

# histgorma for number of employees
cis %>%
    with(hist(bges, main = "number of employees"))
# boxplot for sales
cis %>%
    with(boxplot(um18,  main = "Boxplot of Sales in 2018"))
# boxplot for log(1+sales)
cis %>%
    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 $\log(x)$.

```
# log transform
options(repr.plot.width = 10, repr.plot.height = 10)
par(mfrow = c(2, 2))
cis %>%
    with(hist(um18, main = "Histgoram of Sales (2018)"))

cis %>%
    with(hist(log(1+um18), main = "Histogram of Log (1+sales)"))

cis %>%
    with(boxplot(um18, main = "Boxplot of Sales (2018)"))
cis %>%
    with(boxplot(log(1+um18), main = "Boxplot of Log Transform "))
```

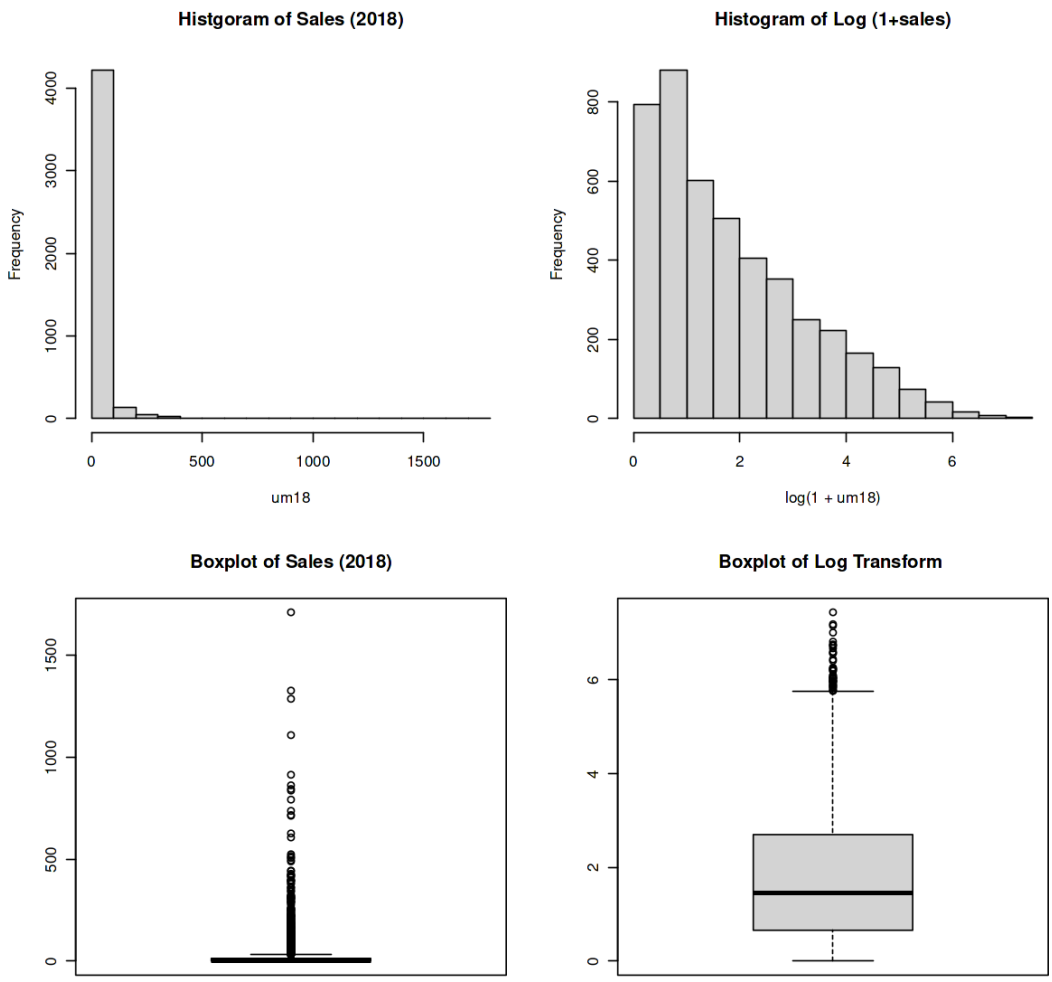
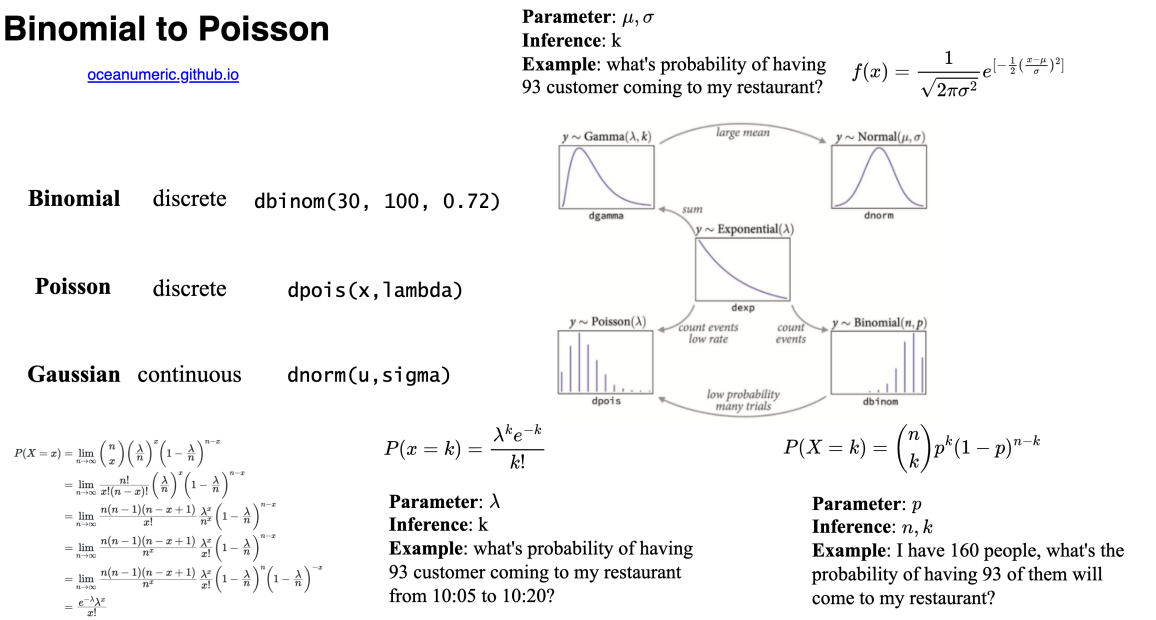


Figure 1: Log transformation of sales

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.



Notes or R-code

3 Simple Linear Regression

In our tutorials, we follow the following analytic frapmework:

	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 nosie 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.

```
# simulate weight and height
# generate 900 random numbers from normal distribution
# mean =  175cm and sd = 10cm
height ← rnorm(900, mean = 175, sd = 10)
# plot histogram
hist(height, breaks = 20, xlab="Height (cm)",
             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:

$$y = b + ax$$

```
# generate weight
# CHANGE THE NUMBER AND PLAY WITH IT :)
weight ← 55 + 0.1 * height
# plot scatter plot
plot(height, weight, main = "Simulated Data Without Noise")
# plot histogram of weight
hist(weight, breaks = 20, xlab="weight (kg)")
# now let's fit a linear regression model
sm1 ← lm(weight ~ height)
stargazer(sm1, type = "text")
```

```
# put the fitted line into the plot
plot(height, weight)
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.

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

```
# add some noise to weight
# weight = 55 + 0.1 * height
# weight2 = weight + noize (rnorm)
weight2 ← weight + rnorm(900, mean = 0, sd = 2)
# plot scatter plot
plot(height, weight2, main = "Simulated Data With Noise")
# now we will fit linear regression with weight2 ~ height
sm2 ← lm(weight2 ~ height)
```

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

Table 1: Regression Results for Two Modles

	Dependent variable:	
	weight	weight2
	(1)	(2)
height	0.100***	0.103***
	(0.000)	(0.007)
Constant	55.000***	54.556***
	(0.000)	(1.155)
Observations	900	900
R^2	1.000	0.212
Adjusted R^2	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

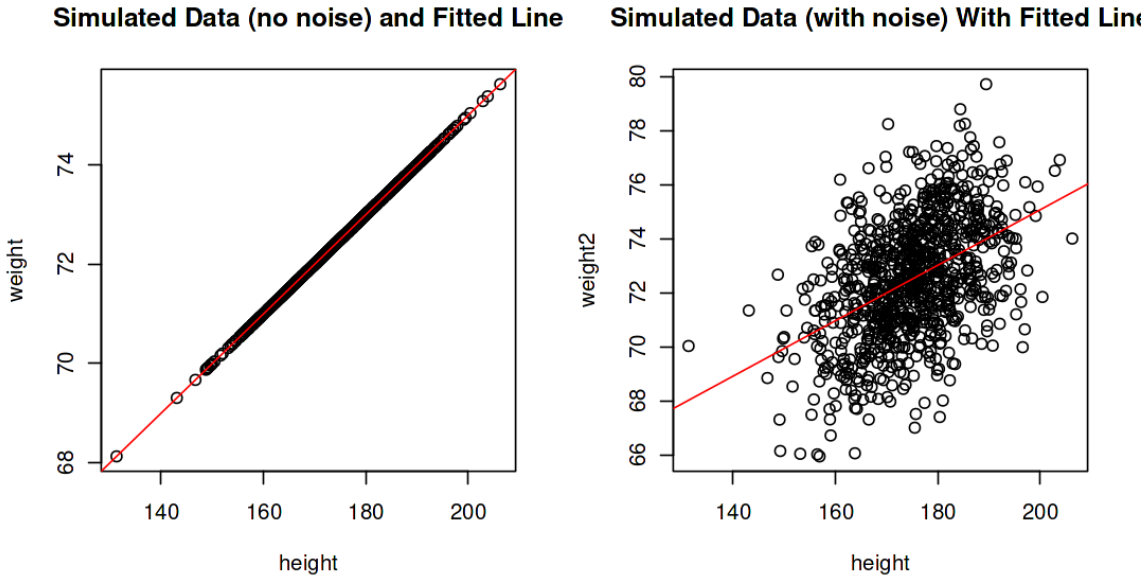


Figure 2: Simulated Data Without or With Noise

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

```
# generate height for female
height_female ← rnorm(450, 170, 5)
female_character ← rep("female", 450)
height_male ← rnorm(450, 175, 10)
male_character ← rep('male', 450)
# put them together as a data.frame and then conver it to the data.
    table
sim_data ← data.frame(height = c(height_female, height_male), gender =
    c(female_character, male_character))
sim_data ← as.data.table(sim_data)
head(sim_data)

# add weight
sim_data %>%
    #[i, j, by]
    .[, weight := ifelse(gender=="female",
                                    50 + 0.09 *height,
                                    55 + 0.1 * height)] -> sim_data2

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.

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

Table 2: Linear Regression Results

	Dependent variable:	
	weight	
	(1)	(2)
height	0.222***	0.098***
	(0.012)	(0.0001)
gendermale		6.713***
		(0.002)
Constant	30.472***	48.603***
	(2.163)	(0.022)
Observations	900	900
R^2	0.261	1.000
Adjusted R^2	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

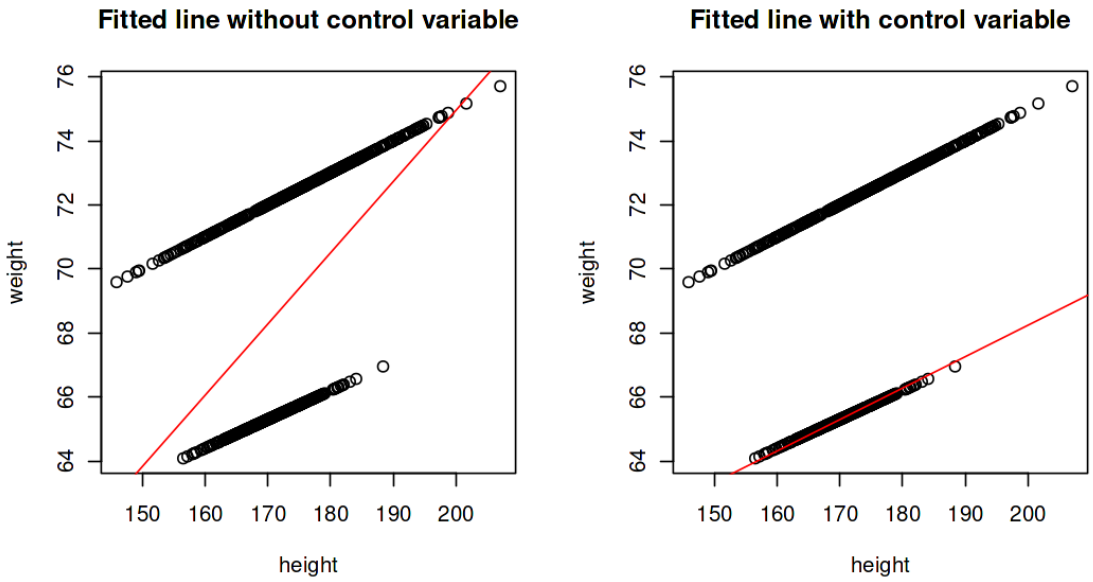


Figure 3: Simulated Data and Fitted Regression Line

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.

```
# install a new package called wooldridge
install.packages("wooldridge")
library(wooldridge)
# load the data
data("wage1")
# convert it to data.table
wage1 <- as.data.table(wage1)
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

```
# univariate anlaysis
wage1 %>%
    with(hist(wage))
# bivariate analysis
wage1 %>%
     with(plot(educ, wage))
# run regression
wage_reg1 <- lm(wage ~ educ, data=wage1)
stargazer(wage_reg1, type="text")
# bivariate: experience and wage
wage1 %>%
    with(plot(exper, wage))
# let's run another regression
wage_reg2 <- lm(wage ~ educ + exper, data=wage1)
stargazer(wage_reg2, type="text")
# let's add non-linear term
wage_reg3 <- lm(wage ~ educ + exper + I(exper^2), data=wage1)
stargazer(wage_reg1, wage_reg2, wage_reg3, type="text")
```

Table 3: Regression Models for Wage

	Dependent variable:		
	wage		
	(1)	(2)	(3)
educ	0.541***	0.644***	0.595***
	(0.053)	(0.054)	(0.053)
exper		0.070***	0.268***
		(0.011)	(0.037)
exper^2			−0.005***
			(0.001)
Constant	−0.905	−3.391***	−3.965***
	(0.685)	(0.767)	(0.752)
Observations	526	526	526
R^2	0.165	0.225	0.269
Adjusted R^2	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<0.1; **p<0.05; ***p<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 an 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 $exper^2$ 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:

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

```
# simulate experience from 0 to 40 years
# seq = sequence generated from 0 to 30 with interval 1
exper_seq ← seq(0, 40, 1)
# ^2 means square
wage_sim ← 0.268 * exper_seq - 0.005 * exper_seq^2
# plot the relationship
plot(exper_seq, wage_sim, type="l")
# add vertical line
abline(v=27, col='red')
# we can put them together
wage1 %>%
    with(plot(exper, wage)) +
    lines(exper_seq, wage_sim, type="l", col='red')
# let's simulate other equation
wage_sim2 ← 10 + 0.268 * exper_seq - 0.005 * exper_seq^2

```

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

Why the red curve above does not fit with the dataset exactly? Wage is real-lfe 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, image let's assume the wage was determined by the following equaion:

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

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.

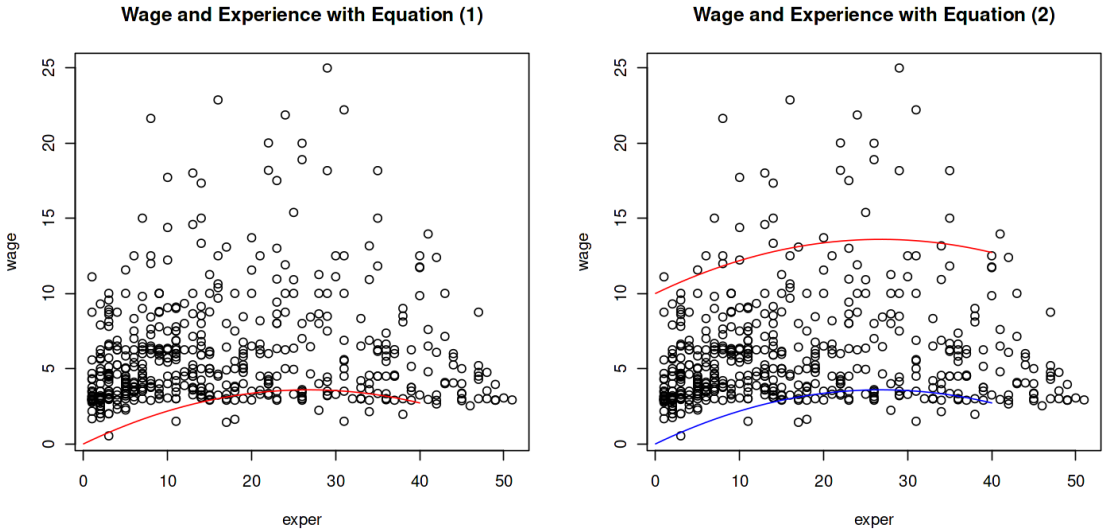


Figure 4: Nolinear relationship illustration

We have controled 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.

```r
# add gender in the regression
wage_reg4 ← lm(wage ~ educ + exper + I(exper^2) + female, data=wage1)
stargazer(wage_reg4, type="text")
```

Table 4

	Dependent variable:		
	wage		
	(1)	(2)	(3)
educ	0.644***	0.595***	0.556***
	(0.054)	(0.053)	(0.050)
exper	0.070***	0.268***	0.255***
	(0.011)	(0.037)	(0.035)
I(exper^2)		−0.005***	−0.004***
		(0.001)	(0.001)
female			−2.114***
			(0.263)
Constant	−3.391***	−3.965***	−2.319***
	(0.767)	(0.752)	(0.739)
Observations	526	526	526
R^2	0.225	0.269	0.350
Adjusted R^2	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.

```
install.packages("performance")
install.packages("see")
install.packages("patchwork")
install.packages("wooldridge")
library(performance)
library(wooldridge)

# high influential points
data("infmrt")
str(infmrt)
infmrt ← as.data.table(infmrt)
infmrt %>%
    .[lphysic ≤ 6.0] %>%
    .[, .(infmort, lpcinc, lphysic, lpopul)] -> infmrt2

head(infmrt2)
infmrt %>%
    with(plot(lphysic, infmort))
infmrt_reg1 ← lm(infmort ~ lpcinc + lphysic + lpopul, data = infmrt)
stargazer(infmrt_reg1, type="text")
check_model(infmrt_reg1)
infmrt_reg2 ← lm(infmort ~ lpcinc + lphysic + lpopul, data = infmrt2)
stargazer(infmrt_reg2, type="text")
check_model(infmrt_reg2)

# multicollinearity
elem_data ← fread("https://shorturl.at/awLNT")
head(elem_data)
str(elem_data)
elem_reg1 ← lm(api00 ~ acs_k3 + avg_ed +  grad_sch + col_grad + some_
    col, data = elem_data)
stargazer(elem_reg1, type='text')
options(repr.plot.width = 11, repr.plot.height = 8)
check_model(elem_reg1)
```

4 Multiple Linear Regression

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

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon \tag{1}$$

where

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

- ε is the error term

We assume 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 ε: $cor(x, \varepsilon) = 0$	serial correlation test

Table 5: A summary of model assumptions and regression diagnostic

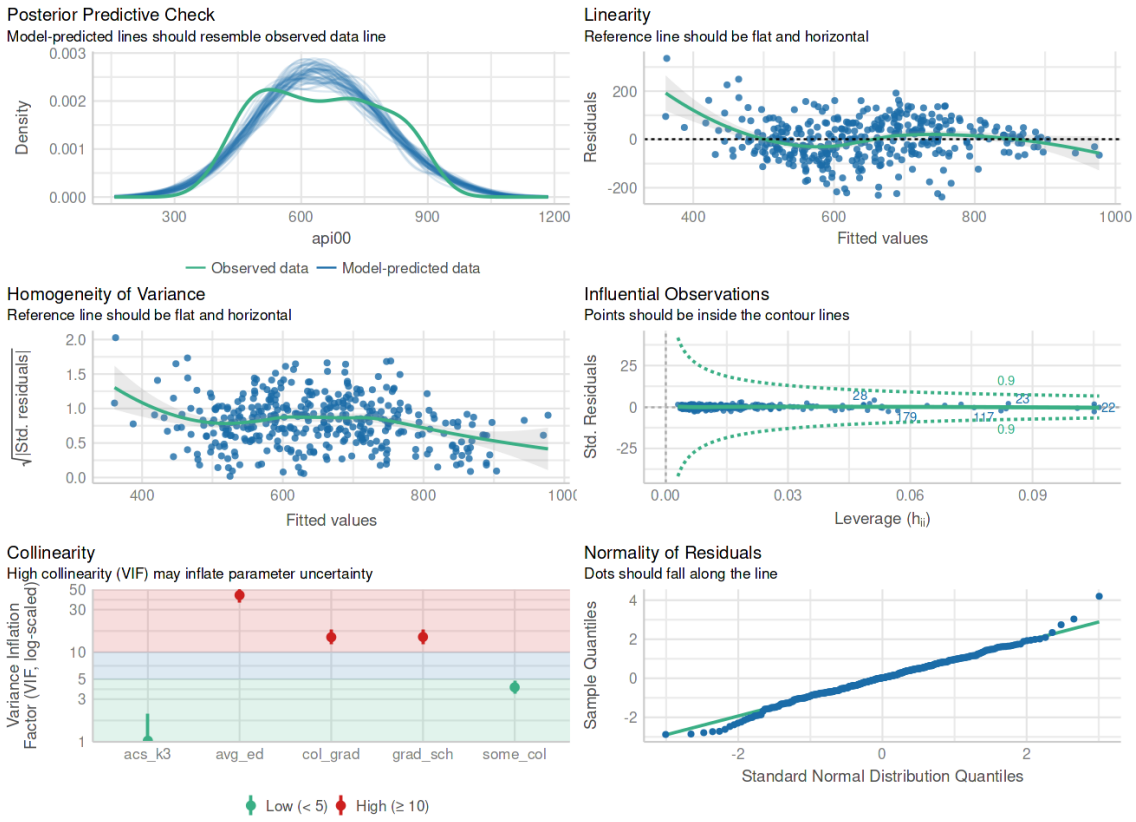


Figure 5: Regression Diagnostics Plots

5 Logistic Regression

You will not 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://daviddalpiaz.github.io/r4sl/logistic-regression.html to learn more about logistic regression.

You see somehow I managed to put everything into a 15 pages document. But, I guess you will now drink ☕ and then finish the rest of the materials.

6 Please Read the Following Materials

All in One for the Exam

Time flies, and the exam is coming. To help you to review the course, we have prepared this notebook, which contains all the materials we have covered in the course. We hope this notebook can help you to review the course and prepare for the exam.

This exam review document **does not cover the materials you studied from the lecture, whereas it only covers the materials from tutorial and lab**. You should also review the lecture materials.

Here is the list of topics we have covered and will be covered in the exam:

1. introduction to data.table (all about structure of data and type of data)
2. using data.table to manipulate data (most of this part **will not be tested** in the exam)
3. basic data visualization
4. introduction to linear regression
5. introduction to logistic regression (maybe a small question)

1. Introduction to 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:

data.table-example

> Small story (will not be tested in the exam): I had a talk with a person who is the principal data scientist and working for the government. He told me that the government is implementing a strategy called "AI in 2030". The goal of this strategy is to make the government to be more data-driven and AI-driven. He told me that every year they have to pay a lot of money to consultancies such as Boston Consulting Group (BCG) to do data analysis for them. He said that the government now is planning to combine data scientist and ChatGPT to do the data analysis. They are hoping that they could reduce 40% of the cost by doing so. The idea is that they will only hire BCG for those very complex data analysis tasks. For those simple tasks, they will use AI to do the data analysis. The main tool that BCG uses is Excel, SQL and Tableau. They are all table-based tools. This means having a good understanding of table-based data analysis is very important. This is why we start from data.table.

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.

data.table-syntax

2. Using data.table to manipulate data

Now, we will use data.table to do some data analysis. We will use the `Community Innovation Survey` (CIS) to do the analysis. The CIS is a survey that is conducted by the European Union (EU) to collect data about innovation activities of firms. The survey we will use is the 2021 CIS from Germany. **You will not be examined on how to manipulate data with data.table**.

```
# install packages
install.packages("stargazer")
# install ISLR if you don't have it
# install.packages("ISLR")
install.packages("corrplot")

# library for data analysis
library(data.table)
library(magrittr)
library(ggplot2)
library(knitr)
library(stargazer)
library(MASS)
library(ISLR)
library(corrplot)

# read data
# original url: https://raw.githubusercontent.com/oceanumeric/data-
science-go-small/main/data/innovation_survey/extmidp21.csv
cis <- fread("https://shorturl.at/wBESZ")

# check dimension, which shows 5083 rows and 284 columns
dim(cis)

# take a look at the first 5 rows
head(cis)
```

From the above table, we can see that we have:

- integer variables: `id`
- character variables: `branche`, `ost`, etc.
- numeric variables (dbl): `bges`, `bges18`, etc.

```
# now we want to select all variables that are type of dbl (double -
numeric like 3.1415926)
# .SD means subset of data in j (select all columns)
# .SDcols means subset columns that are double which is by = is.double
# all things we will do in excel, SQL, or data.table can be summarized
as
```

```r
# I want to do things on rows (i) or columns (j) by some conditions
(by)
# here we want to select all columns that are double
cis %>%
    # using [i, j, by] grammar
    .[, .SD, .SDcols = is.double] %>% head()
```

Before we continue, let's get to know the meaning of some variables (we have 41 numeric variables in total, so we will not cover all of them):

- bges: number of employees (average from 2018 to 2020)
- bges18: number of employees in 2018
- um18: sales in 2018
- lp19: labor productivity in 2019
- softws19: software application intensity in 2019
- wbp: Weiterbildungskostenanteil (share of training costs)
- invs: Investitionsintensitaet (investment intensity)

From the above table, you can see that we have many NA values. This is because the survey is a voluntary survey. This means that firms can choose whether they want to participate in the survey or not. If they choose to participate in the survey, they can choose whether they want to answer all the questions or not. This is why we have many NA values - missing values.

This is very common in real-world data. To deal with this issue, we normally do the following (will not be tested in the exam):

- check the number of missing values
- check the pattern of missing values
- decide whether we want to drop the missing values or impute the missing values

```r
# check the number of missing values in each column
cis %>%
    .[, .SD, .SDcols = is.double] %>%
    # check the number of missing values in each column
    sapply(function(x) sum(is.na(x))) %>%
    # sort the number of missing values in each column
    sort(decreasing = TRUE) %>%
    # convert to data.table and keep rownames as a column
    as.data.table(keep.rownames = TRUE) %>%
    # set variable names
    setnames(c("variables", "Numbe of NAs")) %>%
    # check the first 10
    head(10) %>%
    kable("pipe")

# four figures in one page
# bar chart, histogram, box plot and box plot compared
# figure size
options(repr.plot.width = 10, repr.plot.height = 10)
```

```
par(mfrow = c(2, 2))
cis %>%
    # group by branche and .N calculates the number of frequency
    .[, .N, by = branche] %>%
    # order it in a descending way
    .[order(-N)] %>%
    # get the top 10 branche (industry)
    head(10) %>%
    # plot it
    with(pie(N, labels = branche, main = "Distribution of
Industries"))

# histgorma for number of employees
cis %>%
    with(hist(bges, main = "number of employees"))

# boxplot for sales
cis %>%
    with(boxplot(um18,  main = "Boxplot of Sales in 2018"))
# boxplot for log(1+sales)
cis %>%
    with(boxplot(log(1+um18), main = "Boxplot of Log Transform "))

# log transform
options(repr.plot.width = 10, repr.plot.height = 10)
par(mfrow = c(2, 2))
cis %>%
    with(hist(um18, main = "Histgoram of Sales (2018)"))

cis %>%
    with(hist(log(1+um18), main = "Histogram of Log (1+sales)"))

cis %>%
    with(boxplot(um18, main = "Boxplot of Sales (2018)"))
cis %>%
    with(boxplot(log(1+um18), main = "Boxplot of Log Transform "))
```

Missing Values

- wmenup: Umsatzanteil der Weltmarktneuheiten in 2020 (share of world market novelties in 2020)
- lap2022: Entw. Innovationsaufw. 2022 in % (Development of innovation expenditure 2022 in %)
- fueoefms: Intensitaet öffentliche FuE-Förderung (Intensity of public R&D funding)

We do not know exactly why we have so many missing values. However, we can guess that the reason is that the firms do not want to disclose the information or do not have information. For instance, because of the pandemic in 2020, many firms might not have the information about the share of world market novelties in 2020. This could be the reason why we have so many missing values for `wmenup`.

We have shown that the sample size is 5083, if one variable has more than 50% missing values, we will drop this variable. This is because we can do nothing about this variable. If we impute the missing values, we will introduce bias to the data.

```
dim(cis)
```

```
# get variable names that have more than 50% missing values
cis %>%
    .[, .SD, .SDcols = is.double] %>%
    # check the number of missing values in each column
    sapply(function(x) sum(is.na(x))) %>%
    # sort the number of missing values in each column
    sort(decreasing = TRUE) %>%
    # convert to data.table and keep rownames as a column
    as.data.table(keep.rownames = TRUE) %>%
    head()
# notice that the variable name of second column is a `.`
# we will rename it as `num_missing` (number of missing values)
```

```
cis %>%
    .[, .SD, .SDcols = is.double] %>%
    # check the number of missing values in each column
    sapply(function(x) sum(is.na(x))) %>%
    # sort the number of missing values in each column
    sort(decreasing = TRUE) %>%
    # convert to data.table and keep rownames as a column
    as.data.table(keep.rownames = TRUE) %>%
    # rename the second column
    setnames(., old = ".", new = "num_missing") %>%
    head()
```

```
cis %>%
    .[, .SD, .SDcols = is.double] %>%
    # check the number of missing values in each column
    sapply(function(x) sum(is.na(x))) %>%
    # sort the number of missing values in each column
    sort(decreasing = TRUE) %>%
    # convert to data.table and keep rownames as a column
    as.data.table(keep.rownames = TRUE) %>%
    # rename the second column
    setnames(., old = ".", new = "num_missing") %>%
    # add missing percentage column
    # nrow is the number of rows in the dataset nrow(cis) = dim(cis)
[1] = 5083
    .[, missing_rate := num_missing / nrow(cis)] %>%
    head()
```

> Normally, I will use %>% to do everything in one block. However, here I am trying to show you the steps one by one. So, you can see the process clearly.

```r
cis %>%
    .[, .SD, .SDcols = is.double] %>%
    # check the number of missing values in each column
    sapply(function(x) sum(is.na(x))) %>%
    # sort the number of missing values in each column
    sort(decreasing = TRUE) %>%
    # convert to data.table and keep rownames as a column
    as.data.table(keep.rownames = TRUE) %>%
    # rename the second column
    setnames(., old = ".", new = "num_missing") %>%
    # add missing percentage column
    # nrow is the number of rows in the dataset nrow(cis) = dim(cis)
[1] = 5083
    .[, missing_rate := num_missing / nrow(cis)] %>%
    # filter variables that have missing rate > 0.5
    # we are using [i, j, by] grammar again on rows now
    .[missing_rate >= 0.5]

# as you can see we have 18 variables that have missing rate > 0.5
# we will remove them from the dataset

cis %>%
    .[, .SD, .SDcols = is.double] %>%
    # check the number of missing values in each column
    sapply(function(x) sum(is.na(x))) %>%
    # sort the number of missing values in each column
    sort(decreasing = TRUE) %>%
    # convert to data.table and keep rownames as a column
    as.data.table(keep.rownames = TRUE) %>%
    # rename the second column
    setnames(., old = ".", new = "num_missing") %>%
    # add missing percentage column
    # nrow is the number of rows in the dataset nrow(cis) = dim(cis)
[1] = 5083
    .[, missing_rate := num_missing / nrow(cis)] %>%
    # filter variables that have missing rate > 0.5
    # we are using [i, j, by] grammar again on rows now
    .[missing_rate >= 0.5] %>%
    # select the first column as they are the variable names
    # [i, j, by]
    .[, rn]

cis %>%
    .[, .SD, .SDcols = is.double] %>%
    # check the number of missing values in each column
    sapply(function(x) sum(is.na(x))) %>%
    # sort the number of missing values in each column
    sort(decreasing = TRUE) %>%
    # convert to data.table and keep rownames as a column
    as.data.table(keep.rownames = TRUE) %>%
```

```r
    # rename the second column
    setnames(., old = ".", new = "num_missing") %>%
    # add missing percentage column
    # nrow is the number of rows in the dataset nrow(cis) = dim(cis)
[1] = 5083
    .[, missing_rate := num_missing / nrow(cis)] %>%
    # filter variables that have missing rate > 0.5
    # we are using [i, j, by] grammar again on rows now
    .[missing_rate >= 0.5] %>%
    # select the first column as they are the variable names
    # [i, j, by]
    # save the variable names to a variable called vars_to_remove
    .[, rn] -> vars_to_remove

# in jupyter notebook we can use vars_to_remove to see the variable
names
vars_to_remove

# or you can print out the variable names
print(vars_to_remove)

# now we will remove the variables from the dataset
cis %>%
    # select variables that are numeric (double)
    .[, .SD, .SDcols = is.double] %>%
    # remove variables that are in vars_to_remove
    # with means we are usig column names instead of column indices
    # at by position
    .[, !vars_to_remove, with = FALSE] %>%
    head()

# compare with the original dataset
cis %>%
    .[, .SD, .SDcols = is.double] %>%
    dim()  # 41 columns (variables)

cis %>%
    # select variables that are numeric (double)
    .[, .SD, .SDcols = is.double] %>%
    # remove variables that are in vars_to_remove
    # with means we are using column names instead of column indices
    # with means we are using column names instead of column indices
    .[, !vars_to_remove, with = FALSE] %>%
    dim()  # 23 columns (variables)

# with all numeric variables and not many missing values
# we can check the correlation between variables
cis %>%
    # select variables that are numeric (double)
    .[, .SD, .SDcols = is.double] %>%
    # remove variables that are in vars_to_remove
```

```
    # with means we are using column names instead of column indices
    # with means we are using column names instead of column indices
    .[, !vars_to_remove, with = FALSE] %>%
    # calculate the correlation between variables without missing
values
    cor(use = "pairwise.complete.obs") %>%
    # plot the correlation matrix
    corrplot(method = "color")
```

You might see the correlation plot like the above one in the exam. I will tell you the meaning of variables:

- bges: number of employees (average from 2018 to 2020)
- bges18: number of employees in 2018
- um18: sales in 2018
- lp19: labor productivity in 2019
- softws19: software application intensity in 2019
- wbp: Weiterbildungskostenanteil (share of training costs)
- invs: Investitionsintensitaet (investment intensity)
- exs: Exportintensitaet insgesamt (total export intensity)
- markets: Marketingaufwendungen Intensitaet (marketing expenditure intensity)
- designs: Designaufwendungen Intensitaet (design expenditure intensity)

For instance, I might ask you:

- in the graph, we see the correlated cluster of `markets`, `designs`, and `softws19`. What does this mean?
- in the graph, we see that there is correlation between `exs`, `bges` and `bges18`, what kind of hypothesis can we make based on this correlation?

```
# we can also use circlize to plot the correlation matrix
cis %>%
    # select variables that are numeric (double)
    .[, .SD, .SDcols = is.double] %>%
    # remove variables that are in vars_to_remove
    # with means we are using column names instead of column indices
    # with means we are using column names instead of column indices
    .[, !vars_to_remove, with = FALSE] %>%
    # calculate the correlation between variables without missing
values
    cor(use = "pairwise.complete.obs") %>%
    # plot the correlation matrix with circlize with half matrix
    corrplot(method = "circle", type = "upper")
```

The correlation plot could give us a big picture on what's going on. For instance,

- factors are grouped together based on their correlation
- the bigger the circle, the higher the correlation

- the color of the circle indicates the sign of the correlation (blue: positive, red: negative)

Based on the correlation plot, you can combine your domain knowledge to make hypothesis.

However, **This kind of correlation plot only gives a global picture. It does not tell us the whole story**. For instance, we are not zooming into the industry level analysis. **It is very likely that the correlationb between say `markets` and `designs` is different for different industries**. This is why we need to do industry level analysis.

CIS with Industry Level Analysis

Now, we will do industry level analysis. We will use the `branche` variable to do the industry level analysis.

- `branche`: Einteil. in 21 Wirtschaftszweige (classification into 21 economic sectors)

```
# get to know branche
cis %>%
    # select branche with [i, j, by]
    # .(branchen) is the same as c("branchen")
    .[, .(branche)] %>%
    head()

# we can check how many unique values in branche
cis %>%
    .[, .(branche)] %>%
    # unique() returns the unique values
    unique()  # 21 unique values (industries)

# branche is character
# we can calculate the frequency of each branche
cis %>%
    .[, .(branche)] %>%
    # calculate the frequency of each branche
    # [i, j, by] by = branche means we are grouping by branche
    # .N is calculating the number of rows in each group
    .[, .N, by = branche]

cis %>%
    .[, .(branche)] %>%
    # calculate the frequency of each branche
    # [i, j, by] by = branche means we are grouping by branche
    # .N is calculating the number of rows in each group
    .[, .N, by = branche] %>%
    # sort the frequency in descending order
    # order(-N) means we are sorting N in descending order
    # we put it into i position because we are sorting rows
    .[order(-N)]
```

The industry distribution shows the 'big picutre' of German economy. We can see that the most important industry is `Transport/Post` , `Wasser/Entsorgung/Recycling`, and `Metallerzeugung/-bearbeitung`. **Be careful that we might have selection bias here**. This is because the survey is a voluntary survey. This means that firms can choose whether they want to participate in the survey or not. So, the industry distribution might not be representative.

When we see those industries, we are interested in:

- what are the characteristics of those industries?
- which industries are similar to each other?
- which industries are different from each other?
- what are the characteristics of the firms in those industries?

Since we have many industries, to find out which industries are similar to each other is not easy by analyzing the dataset. **We have to reply on our domain knowledge**. Here, we expect that `Transport/Post` and `Metallerzeugung/-bearbeitung` are similar to each other.

With our domain knowledge, we will focus on three groups of industries:

- indus1: `Transport/Post`, `Metallerzeugung/-bearbeitung`, and `Maschinenbau`
- indus2: `Unternehmensdienste`, `Unternehmensberatung/Werbung`, and `Finanzdienstleistungen`
- indus3: `Elektroindustrie`, and `Mediendienstleistungen`, and `EDV/Telekommunikation`

We will now select the firms in those industries and group them together.

```
# create a variables that will include industries we will focus on
indus1 <- c(
    "Transport/Post", "Metallerzeugung/-bearbeitung", "Maschinenbau"
)
indus2 <- c(
    "Unternehmensdienste", "Unternehmensberatung/Werbung",
"Finanzdienstleistungen"
)
indus3 <- c(
    "Elektroindustrie", "Mediendienstleistungen",
"EDV/Telekommunikation"
)

# now select all the rows that have branche in indus1, indus2, indus3
cis %>%
    # [i, j, by] at i postion
    # filter out branche are in indus1, indus2, indus3
    # | means or in R
    # %in% means is in
    .[branche %in% indus1 | branche %in% indus2 | branche %in% indus3]
```

```
%>%
    dim()

cis %>%
    # [i, j, by] at i postion
    # filter out branche are in indus1, indus2, indus3
    # | means or in R
    # %in% means is in
    .[branche %in% indus1 | branche %in% indus2 | branche %in% indus3]
%>%
    # verify the result
    .[, .(branche)] %>%
    unique()

cis %>%
    # [i, j, by] at i postion
    # filter out branche are in indus1, indus2, indus3
    # | means or in R
    # %in% means is in
    .[branche %in% indus1 | branche %in% indus2 | branche %in% indus3]
%>%
    # verify the result
    .[, .(branche)] %>%
    # calculate the frequency of each branche
    .[, .N, by = branche]

# now we group them
cis %>%
    # [i, j, by] at i postion
    # filter out branche are in indus1, indus2, indus3
    # | means or in R
    # %in% means is in
    .[branche %in% indus1 | branche %in% indus2 | branche %in% indus3]
%>%
    # add a new variable called industry
    # fill indus1, indus2, indus3 based on branche
    .[, industry := ifelse(
        branche %in% indus1, "indus1",
        ifelse(branche %in% indus2, "indus2", "indus3")
    )] %>%
    head()  # variable industry is added to the dataset at the last
column

cis %>%
    # [i, j, by] at i postion
    # filter out branche are in indus1, indus2, indus3
    # | means or in R
    # %in% means is in
    .[branche %in% indus1 | branche %in% indus2 | branche %in% indus3]
%>%
    # add a new variable called industry
```

```
    # fill indus1, indus2, indus3 based on branche
    .[, industry := ifelse(
        branche %in% indus1, "indus1",
        ifelse(branche %in% indus2, "indus2", "indus3")
    )] %>%
    # verify the result by selecting branche and industry
    .[, .(branche, industry)] %>%
    head()
```

After classifying the industry, we can compare the correlation between different industries. We want to see whether the relationship between for instance `export` and `labor productivity` is different for different industries.

```
# we will create a vector that will include all the variables we will
use

# plot correlation matrix for indus1 - machineary industry or related
(transport/post, metal, machine)
cis %>%
    # filter out branche are in indus1, indus2, indus3
    .[branche %in% indus1 | branche %in% indus2 | branche %in% indus3]
%>%
    # add a new variable called industry
    .[, industry := ifelse(
        branche %in% indus1, "indus1",
        ifelse(branche %in% indus2, "indus2", "indus3")
    )] %>%
    # filter out the indust1
    .[industry == "indus1"] %>%
    # select the variables are double
    .[, .SD, .SDcols = is.double] %>%
    # remove variables that are in vars_to_remove
    .[, !vars_to_remove, with = FALSE] %>%
    # calculate the correlation between variables without missing
values
    cor(use = "pairwise.complete.obs") %>%
    # plot the correlation matrix with circlize with half matrix
    corrplot(method = "circle", type = "upper")

# plot correlation matrix for indus2 - business services industry or
related (business services, consulting, finance)
cis %>%
    # filter out branche are in indus1, indus2, indus3
    .[branche %in% indus1 | branche %in% indus2 | branche %in% indus3]
%>%
    # add a new variable called industry
    .[, industry := ifelse(
        branche %in% indus1, "indus1",
        ifelse(branche %in% indus2, "indus2", "indus3")
    )] %>%
    # filter out the indust1
```

```
    .[industry == "indus2"] %>%
    # select the variables are double
    .[, .SD, .SDcols = is.double] %>%
    # remove variables that are in vars_to_remove
    .[, !vars_to_remove, with = FALSE] %>%
    # calculate the correlation between variables without missing
values
    cor(use = "pairwise.complete.obs") %>%
    # plot the correlation matrix with circlize with half matrix
    corrplot(method = "circle", type = "upper")

# plot correlation matrix for indus3 - IT industry or related (IT,
telecom, media)
cis %>%
    # filter out branche are in indus1, indus2, indus3
    .[branche %in% indus1 | branche %in% indus2 | branche %in% indus3]
%>%
    # add a new variable called industry
    .[, industry := ifelse(
        branche %in% indus1, "indus1",
        ifelse(branche %in% indus2, "indus2", "indus3")
    )] %>%
    # filter out the indust1
    .[industry == "indus3"] %>%
    # select the variables are double
    .[, .SD, .SDcols = is.double] %>%
    # remove variables that are in vars_to_remove
    .[, !vars_to_remove, with = FALSE] %>%
    # calculate the correlation between variables without missing
values
    cor(use = "pairwise.complete.obs") %>%
    # plot the correlation matrix with circlize with half matrix
    corrplot(method = "circle", type = "upper")
```

Now, let's put them together and compare them. I will write a function to do this because we do not have to repeat the same code again and again (it will **not** be tested in the exam).

```
plot_correlation <- function(industry_name) {
    cis %>%
    # filter out branche are in indus1, indus2, indus3
    .[branche %in% indus1 | branche %in% indus2 | branche %in% indus3]
%>%
    # add a new variable called industry
    .[, industry := ifelse(
        branche %in% indus1, "indus1",
        ifelse(branche %in% indus2, "indus2", "indus3")
    )] %>%
    # filter out the indust1
    .[industry == industry_name] %>%
    # select the variables are double
    .[, .SD, .SDcols = is.double] %>%
```

```
    # remove variables that are in vars_to_remove
    .[, !vars_to_remove, with = FALSE] %>%
    # calculate the correlation between variables without missing
values
    cor(use = "pairwise.complete.obs") %>%
    # plot the correlation matrix with circlize with half matrix
    corrplot(method = "circle", type = "upper", title=industry_name,
mar=c(0,0,2,0))
}

# let's put them together and compare
cis %>%
    # filter out branche are in indus1, indus2, indus3
    .[branche %in% indus1 | branche %in% indus2 | branche %in% indus3]
%>%
    # add a new variable called industry
    .[, industry := ifelse(
        branche %in% indus1, "indus1",
        ifelse(branche %in% indus2, "indus2", "indus3")
    )] %>%
    # select the variables are double
    .[, .SD, .SDcols = is.double] %>%
    # remove variables that are in vars_to_remove
    .[, !vars_to_remove, with = FALSE] %>%
    # calculate the correlation between variables without missing
values
    cor(use = "pairwise.complete.obs") %>%
    # plot the correlation matrix with circlize with half matrix
    corrplot(method = "circle", type = "upper")  -> corr_overall

# figure size
options(repr.plot.width = 10, repr.plot.height = 10)
plot_correlation("indus1")
plot_correlation("indus2")
plot_correlation("indus3")
```

Now, I put everything together and highlight some interesting phenomena.

corr-plot

As you can see that the correlations between `export` and other variables are very different cross different industries.

Summary: for this part, you need to know that industry level analysis is very important. For the coding part, you will not be tested in the exam. However, you need to know how to do it in real life.

3. Basic data visualization

In this section, we will learn how to use basic data visualization tools to conduct exploratory data analysis by focusing on the following topics:

- univariate analysis (histogram, boxplot, density plot)
- bivariate analysis (scatter plot, line plot, bar plot)

Many concepts in the regression section are based on this section. So, please pay attention to this section.

```
# we will use two datasets
# community innovation survey (cis), religion and innovation survey
(ris)
# read data again
cis <- fread("https://shorturl.at/wBESZ")

str(cis)
```

As you can see that we have many variables. It is not easy to understand the data by looking at the numbers. So, we need to visualize the data. We will select some variables of interest:

- ias: Innovationsintensitaet
- iasx: Stutzung Innovationsintensitaet
- iainvs: Intensitaet investive Inno.
- iainvsx: Stutzung Intens.inv.Inno
- iafues: Intensitaet interne FuE
- iafuesx: Stutzung Intens. interne FuE
- iavfues: Intensitaet externe FuE
- iavfuesx: Stutzung Intens. externe FuE
- iasos: Intensitaet sonst. Innovationsaufw
- iasosx: Stutzung sonst. Innovationsaufw

All those variables are related to innovation, which could be used as dependent variables.

```
# get column index of ias
which(colnames(cis) == "ias")
# get column index of iasosx
which(colnames(cis) == "iasosx")

# now we will select the columns we need from 107 to 116
cis %>%
    # in [i, j, by] you can use index too instead of column name
    .[, 107:116] %>%
    head()

cis %>%
    .[, 107:116] %>%
    # check number of missing values
    sapply(function(x) sum(is.na(x))) %>%
    as.data.table(keep.rownames = TRUE)
```

As you can see that there are so many missing values. We will drop those missing values. We will filter the data based on `ias`.

```
# we will have 2056 firms, which is fine for our analysis
cis %>%
    .[!is.na(ias)] %>%
    dim()

# we will focus on firms that have innovation intensity score
cis %>%
    # remove missing values for ias
    .[!is.na(ias)] %>%
    # check the distribution of industry
    .[, .N, by = branche]

# create a variables that will include industries we will focus on
indus1 <- c(
    "Transport/Post", "Metallerzeugung/-bearbeitung", "Maschinenbau"
)
indus2 <- c(
    "Unternehmensdienste", "Unternehmensberatung/Werbung",
"Finanzdienstleistungen"
)
indus3 <- c(
    "Elektroindustrie", "Mediendienstleistungen",
"EDV/Telekommunikation"
)

# we will filter out those three industries we will focus on
cis %>%
    # remove missing values for ias
    .[!is.na(ias)] %>%
    # filter out branche are in indus1, indus2, indus3
    .[branche %in% indus1 | branche %in% indus2 | branche %in% indus3]
%>%
    # add a new variable called industry
    .[, industry := ifelse(
        branche %in% indus1, "machineary",  # call indus1 as
machineary
        ifelse(branche %in% indus2, "business", "IT")
    )] %>%
    head()

# check distribution of industry
cis %>%
    # remove missing values for ias
    .[!is.na(ias)] %>%
    # filter out branche are in indus1, indus2, indus3
    .[branche %in% indus1 | branche %in% indus2 | branche %in% indus3]
%>%
    # add a new variable called industry
```

```
    .[, industry := ifelse(
        branche %in% indus1, "machineary",  # call indus1 as
machineary
        ifelse(branche %in% indus2, "business", "IT")
    )]%>%
    .[, .N, by = industry]  # quite balanced
```

The above process is about cleaning and organizing the dataset, which is very important. Now, we will create a new dataset based on our clearning process and call it `cis2`.

```
cis %>%
    # remove missing values for ias
    .[!is.na(ias)] %>%
    # filter out branche are in indus1, indus2, indus3
    .[branche %in% indus1 | branche %in% indus2 | branche %in% indus3]
%>%
    # add a new variable called industry
    .[, industry := ifelse(
        branche %in% indus1, "machineary",  # call indus1 as
machineary
        ifelse(branche %in% indus2, "business", "IT")
    )] -> cis2

dim(cis2)  # we only have 1042 firms

# let's check innovation intensity score (ias)
cis2 %>%
    .[, .(ias)] %>% head()

# get summary statistics of ias
cis2 %>%
    .[, .(ias)] %>%
    summary()
```

The above summary is very important. It tells us that the first quartile is 0. This means that 25% of the firms do not have any innovation. The median is 0.001, which means that 50% of the firms have innovation intensity less than 0.001. The third quartile is 0.037, which means that 75% of the firms have innovation intensity less than 0.037. The maximum is 1.156, which means that the maximum innovation intensity is 1.156.

This is **very common** in business world as most of the firms are not very innovative.

```
# plot the distribution of ias as it is a continuous variable
cis2 %>%
    .[, .(ias)] %>%
    with(hist(ias, breaks = 20))

# focus on firms that have ias > 0
cis2 %>%
    .[ias > 0] %>%
    dim()  # we have 536 firms
```

```r
# check balance of industry
cis2 %>%
    .[ias > 0] %>%
    .[, .N, by = industry]  # not very balanced

# focus on firms that have ias > 0
cis2 %>%
    .[ias > 0] %>%
    .[, .(ias)] %>%
    with(hist(ias, breaks = 20))

# boxplot
cis2 %>%
    .[ias > 0] %>%
    .[, .(ias)] %>%
    with(boxplot(ias))

# as it can be seen, there are some outliers
# the distribution is right skewed
# we will use log transformation to make it more normal
cis2 %>%
    .[ias > 0] %>%
    .[, .(ias)] %>%
    # log transformation
    with(boxplot(log(ias)))

cis2 %>%
    .[ias > 0] %>%
    .[, .(ias)] %>%
    # log transformation and histogram
    with(hist(log(ias), breaks = 20))
```

Normal Distribution

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 lecture, there are **two main distributions** you need to know:

- normal distribution (Gaussian distribution)
- binomial distribution

Possion distribution will not be tested in the exam.

normal

```r
# we now want to check whether log(ias) is distributed similarly
across industries
# here we will use ggplot2
options(repr.plot.width = 10, repr.plot.height = 5)
cis2 %>%
```

```
    .[ias > 0] %>%
    .[, .(ias, industry)] %>%
    ggplot(aes(x = log(ias), fill = industry)) +
    geom_histogram(bins = 20, alpha = 0.5) +
    facet_wrap(~industry, nrow = 1) +
    theme_bw()
```

For all industries, the shape is of bell shape. This means that they are more or less normally distributed after we:

- filter out the innovation intensity > 0
- take the log of innovation intensity

From univariate to bivariate analysis

We now have a basic understanding of the distribution of innovation intensity. Now, we want to see how innovation intensity is related to other variables. We will focus on the following variables:

- bges: average number of employees (firm size)
- um: Umsatz in Mio. Euro (turnover in million euro)
- exs: Exportquote (export ratio)
- lp: Laborproduktivitaet (labor productivity)
- invs: Investitionsquote (investment ratio)
- markets: Marketingaufwendungen Intensitaet (marketing intensity)
- designs: Designaufwendungen Intensitaet (design intensity)
- softws: Softwareaufwendungen Intensitaet (software intensity)
- wbp: Weiterbildungskostenanteil (training intensity)
- fues: Intensitaet FuE (R&D intensity)
- iainvs: Intensitaet investive Inno. (investment innovation intensity)

```
# check correlation between ias and other variables
cis2 %>%
    # only select firms that have ias > 0
    .[ias > 0] %>%
    # select variables we need
    .[, .(ias, fues, iainvs, bges, um, exs, lp, invs, markets,
designs, softws, wbp)] %>%
    # check correlation without missing values
    cor(use = "pairwise.complete.obs") %>%
    # plot correlation matrix
    corrplot(method="circle", type="upper", tl.col="black", tl.srt=45)
```

As we can see that `ias` (innovation intensity) and `fues` (R&D intensity) are highly correlated. This is not surprising because R&D is the main source of innovation.

It is interesting that the correlation between `ias` and `lp` is very low and even negative. This means that the more innovative firms are not necessarily more productive. However, to be more accurate, we need to do a regression analysis.

```
# now we plot the scatter plot of ias and other variables
cis2 %>%
    .[, .(ias, markets)] %>%
    head()

options(repr.plot.width = 7, repr.plot.height = 5)
cis2 %>%
    .[ias > 0] %>%
    .[, .(ias, markets)] %>%
    with(plot(markets, ias))
```

Overall, you can see that there is a positive relationship between `ias` and `markets`, however, the relationship is not very strong.

```
# ias and softws
options(repr.plot.width = 7, repr.plot.height = 5)
cis2 %>%
    .[ias > 0] %>%
    .[, .(ias, softws)] %>%
    with(plot(softws, ias))

options(repr.plot.width = 7, repr.plot.height = 5)
cis2 %>%
    .[ias > 0] %>%
    .[, .(ias, invs)] %>%
    with(plot(invs, ias))

options(repr.plot.width = 7, repr.plot.height = 5)
cis2 %>%
    .[ias > 0] %>%
    .[, .(ias, fues)] %>%
    with(plot(fues, ias))
```

`fues` means R&D intensity. As you can see that there is a positive relationship between `ias` and `fues`. The above graph shows those two factors are almost identical.

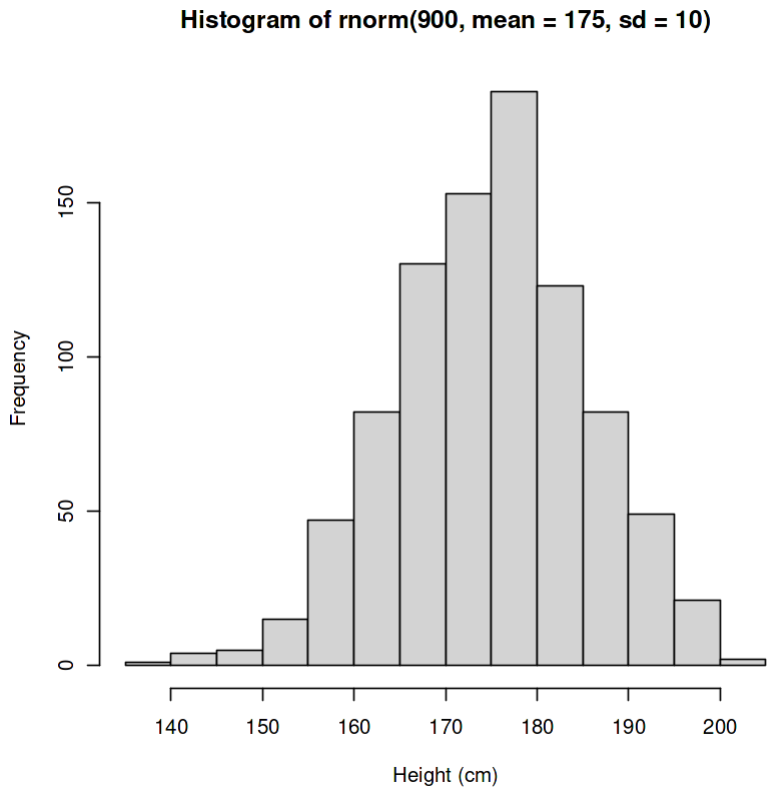
4. Introduction to regression analysis

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

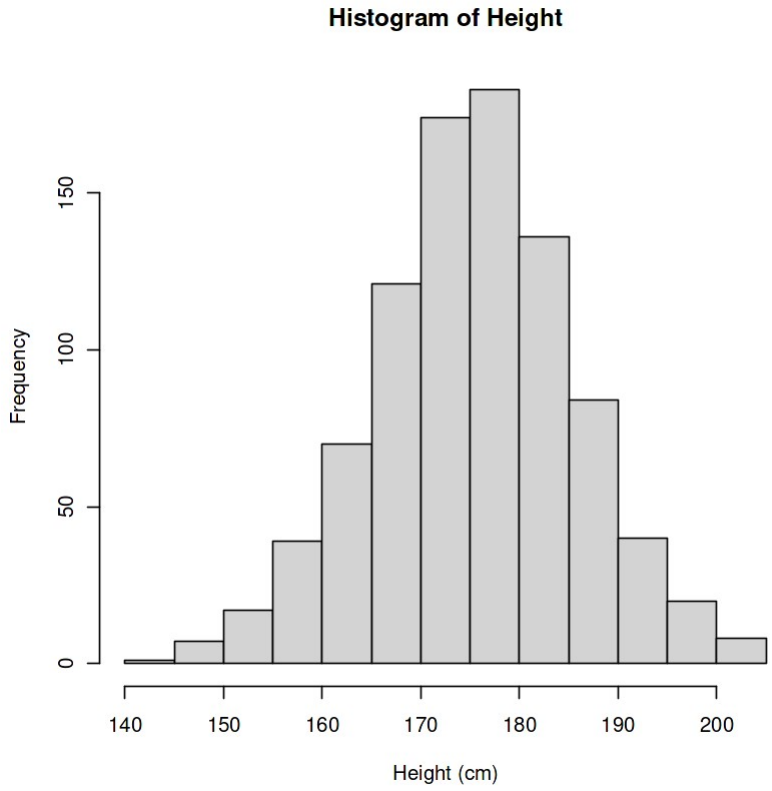
We know that the relationship between `weight` and `height` is roughly linear and positive. We will use this relationship to simulate the data. When we simulate the data, we will add some random nosie to the data as there is no perfect linear relationship between `weight` and `height`. To make you understand 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; TODO Later)

```r
# rnorm = normal distribution function in R
hist(rnorm(900, mean = 175, sd = 10), xlab="Height (cm)")
```




```r
# simulate weight and height
# generate 900 random numbers from normal distribution
# mean =  175cm and sd = 10cm
height <- rnorm(900, mean = 175, sd = 10)
# plot histogram
hist(height, breaks = 20, xlab="Height (cm)", 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:

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

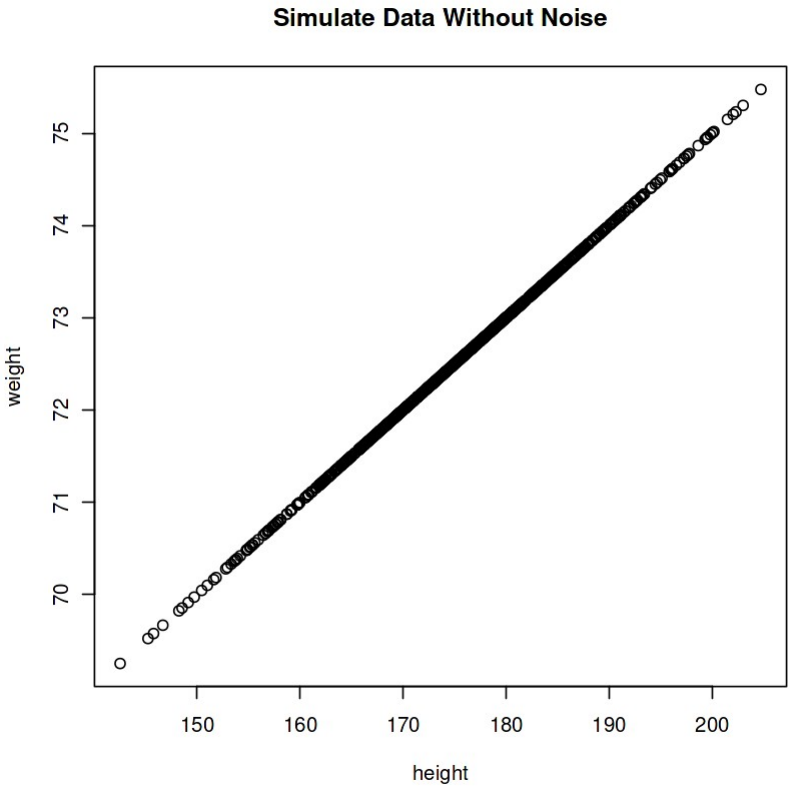
Here we set $\beta_0 = 55$ and $\beta_1 = 0.1$. This means that if the height increases by 1 cm, the weight will increase by 0.1 kg.

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

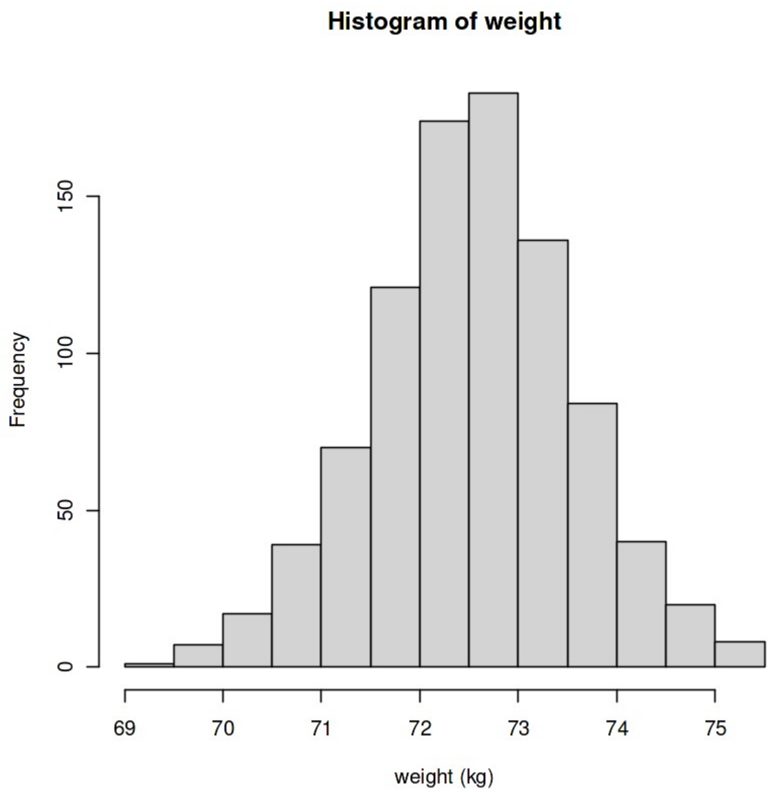
It has the format:

$$y = b + a\,x$$

```
# generate weight
# CHANGE THE NUMBER AND PLAY WITH IT :)
weight <- 55 + 0.1 * height

# plot scatter plot
plot(height, weight, main = "Simulate Data Without Noise")
```




```r
# plot histogram of weight
hist(weight, breaks = 20, xlab="weight (kg)")
```

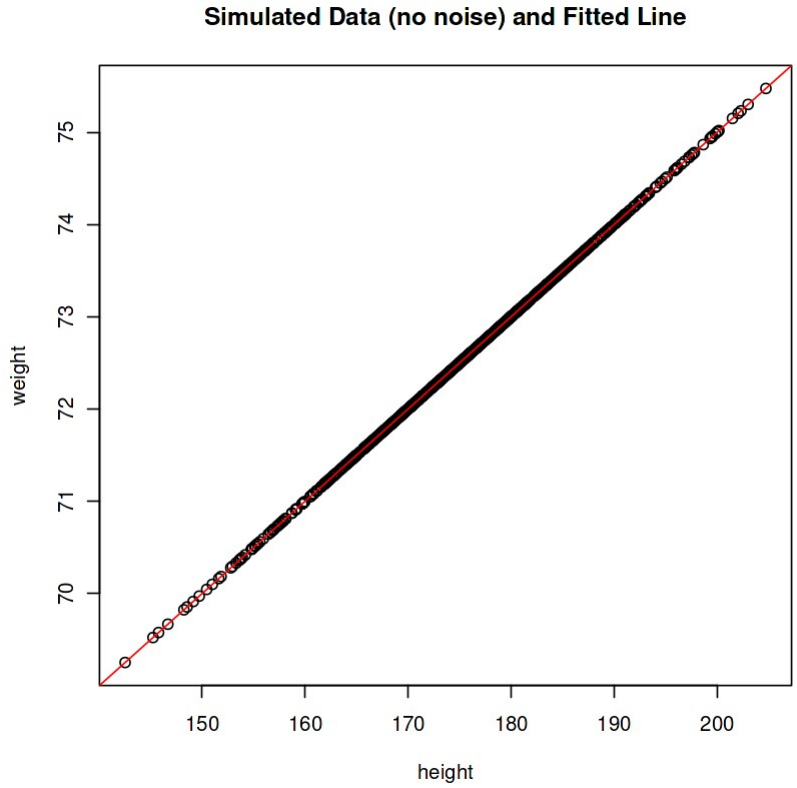



```
# now let's fit a linear regression model
sm1 <- lm(weight ~ height)

stargazer(sm1, type = "text")
```

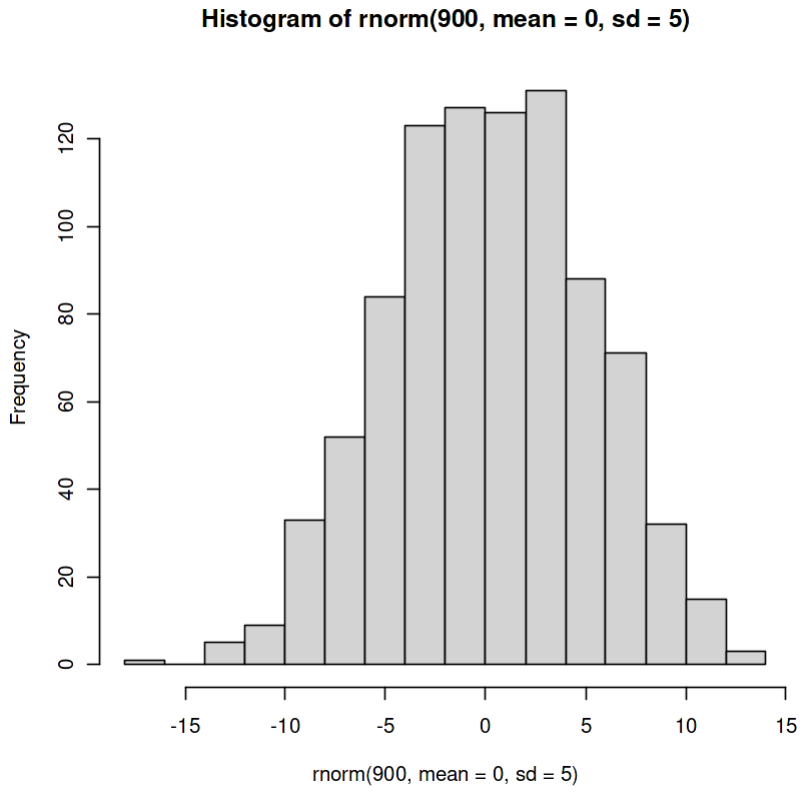
```
======================================================================
=============
                                           Dependent variable:


---------------------------------------------------------------
                                                 weight

----------------------------------------------------------------------
-------------
height                                          0.100***

                                                 (0.000)



Constant                                        55.000***
```

```
                                                          (0.000)

------------------------------------------------------------------------
-------------
Observations                                               900

R2                                                        1.000

Adjusted R2                                               1.000

Residual Std. Error                              0.000 (df = 898)

F Statistic          13,917,167,156,416,598,443,735,038,033,920.000***
(df = 1; 898)
========================================================================
=============
Note:                                                        *p<0.1;
**p<0.05; ***p<0.01
```

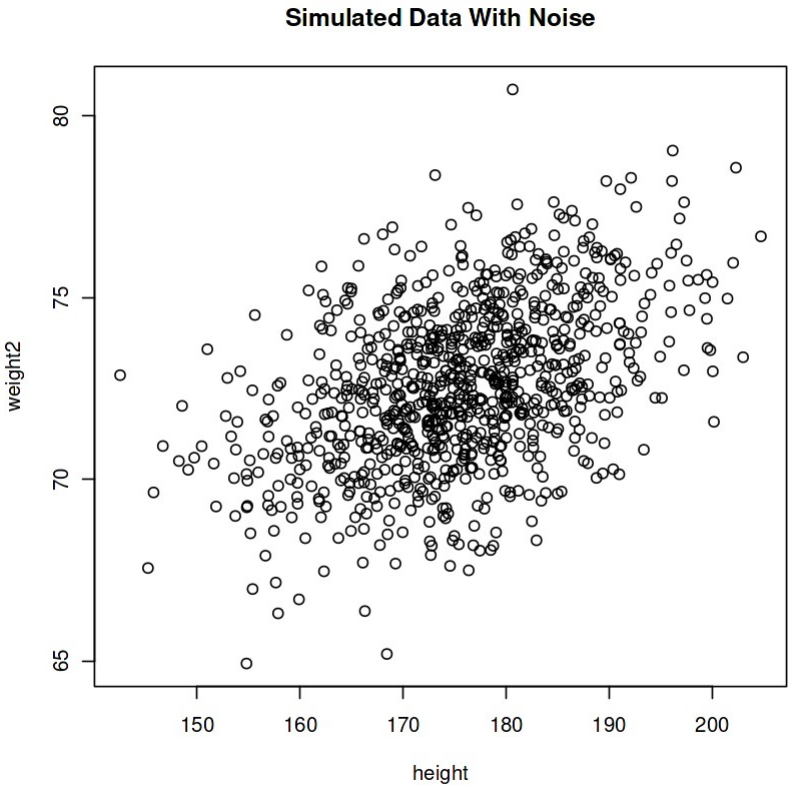
```r
plot(height, weight, main = "Simulated Data (no noise) and Fitted
Line")
abline(sm1, col='red')
```



```r
hist(rnorm(900, mean = 0, sd = 5))
```




```r
# add some noise to weight
# weight = 55 + 0.1 * height
# weight2 = weight + noize (rnorm)
weight2 <- weight + rnorm(900, mean = 0, sd = 2)

# plot scatter plot
plot(height, weight2, main = "Simulated Data With Noise")
```

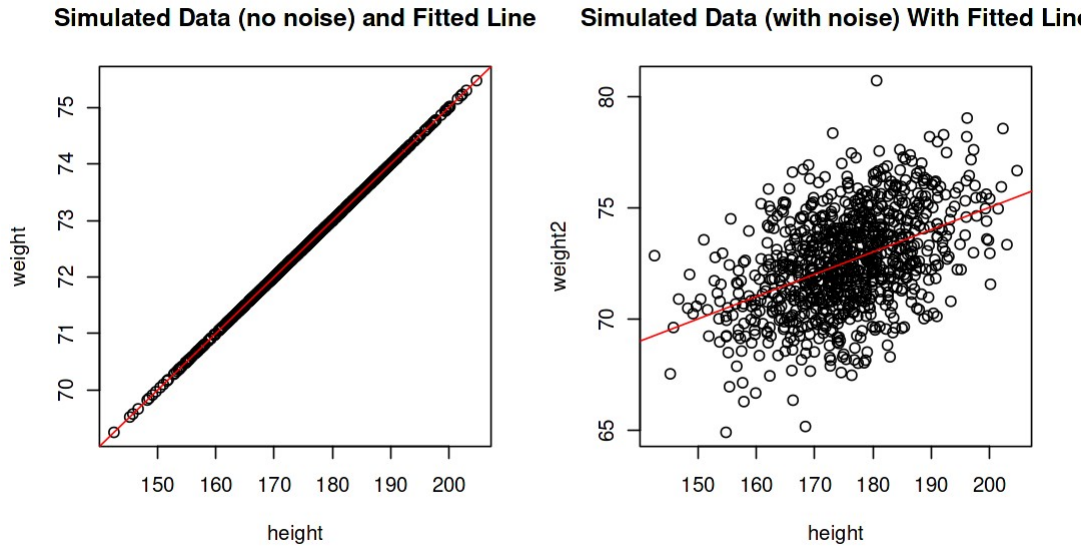



```
# now we will fit linear regression with weight2 ~ height
sm2 <- lm(weight2 ~ height)

# print out the model
stargazer(sm1, sm2, type = "text")
```

```
================================================================================
                                                Dependent variable:

                                    ---------------------------------------------
                                                       weight
weight2
                                                         (1)
(2)
--------------------------------------------------------------------------------
--------------------
height                                                0.100***
0.100***
                                                       (0.000)
(0.007)
```

```
Constant                                                55.000***
55.001***
                                                         (0.000)
(1.151)


------------------------------------------------------------------------
--------------------
Observations                                             900
900
R2                                                      1.000
0.207
Adjusted R2                                             1.000
0.206
Residual Std. Error (df = 898)                          0.000
1.993
F Statistic (df = 1; 898)
13,917,167,156,416,598,443,735,038,033,920.000*** 233.972***
========================================================================
=====================
Note:
*p<0.1; **p<0.05; ***p<0.01
```

```r
# add fitted line
options(repr.plot.width = 9, repr.plot.height = 5)
par(mfrow = c(1, 2))
plot(height, weight, main = "Simulated Data (no noise) and Fitted
Line")
abline(sm1, col='red')
plot(height, weight2, main = "Simulated Data (with noise) With Fitted
Line")
# add model
abline(sm2, col='red')
```



Control variables (from one to two, then to three)

- univariate analysis (one variable)
- bivariate analysis (two variable)
- multivatte anaysis (more than two)

Let's review what we have done:

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

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

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

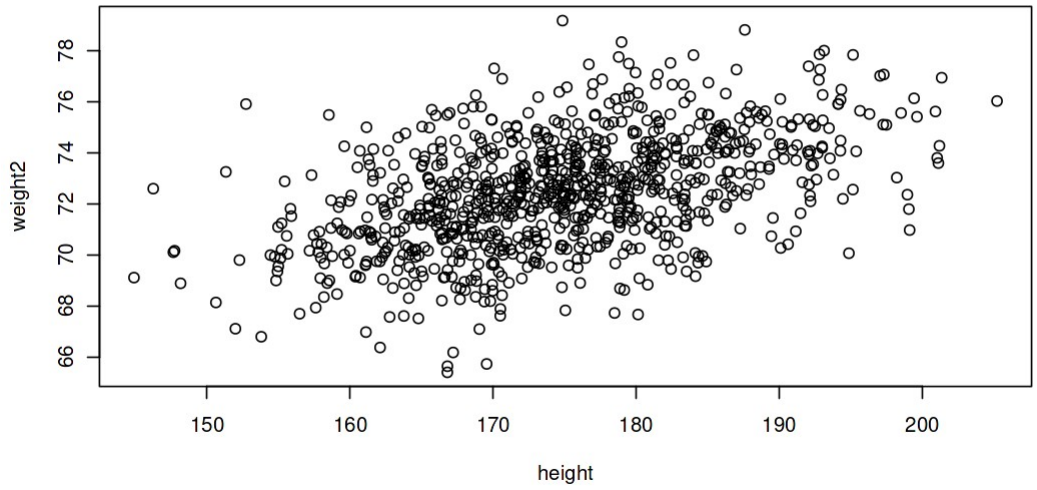
$$weight = 55 + 0.1 * height$$

1. add noise into the data because there is no perfect thing in the real life (except for the GOD)

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

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

```
height <- rnorm(900, mean=175, sd=10)
weight <- 55 + 0.1* height
weight2 <- weight + rnorm(900,0, 2)
plot(height, weight2)
```



Control variables (from one to two, then to three)

- univariate analysis (one variable)
- bivariate analysis (two variable)
- multivatte anaysis (more than two)

Let's review what we have done:

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

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

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

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

1. add noise into the data because there is no perfect thing in the real life (except for the GOD)

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

1. bring one more variable into our analysis, let's say gender (female/male)
2. for female, the distribution of height might be different from male
3. the relationship between height and weight is different for female and male

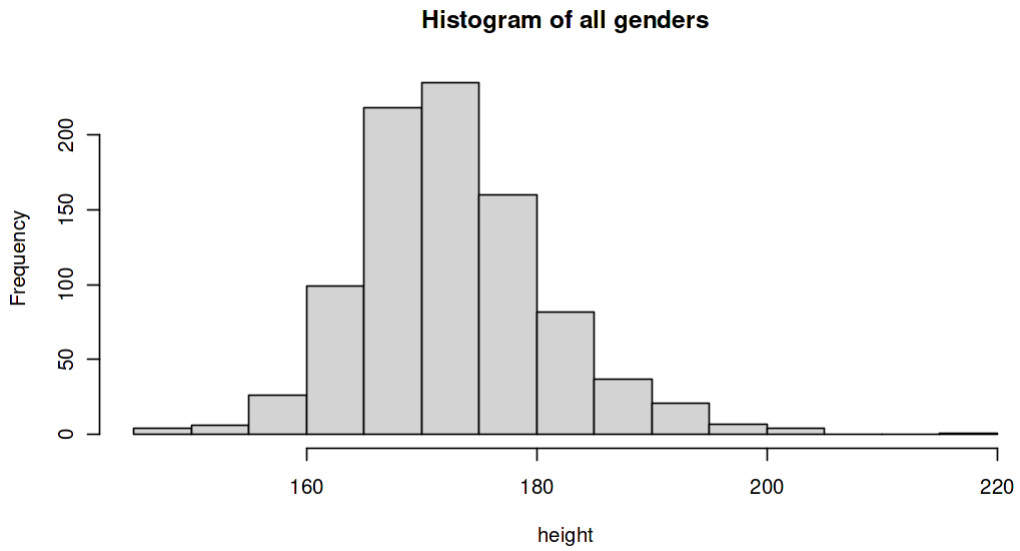
You can see that the complexity has already kicks in even for onlhy three variables.

```
# generate height for female
height_female <- rnorm(450, 170, 5)
female_character <- rep("female", 450)
height_male <- rnorm(450, 175, 10)
male_character <- rep('male', 450)
```

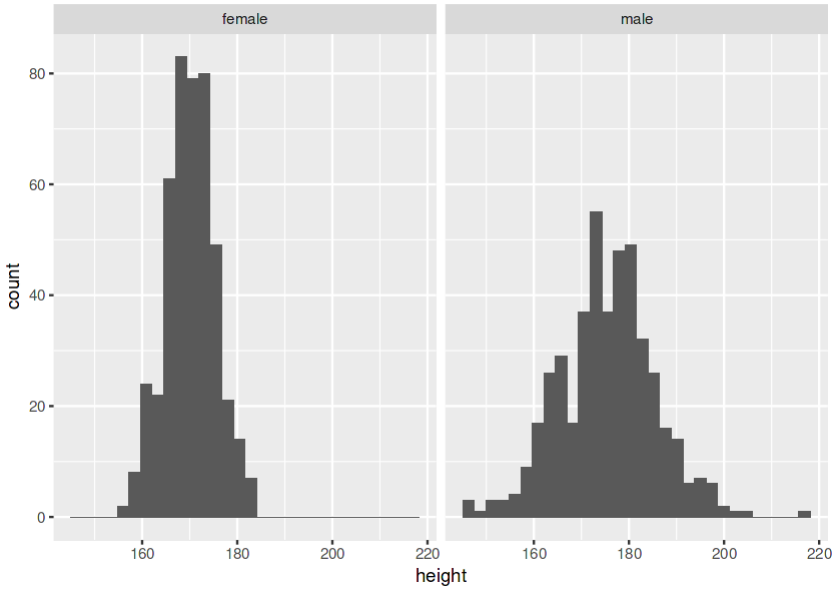
```r
# put them together as a data.frame and then conver it to the
data.table
sim_data <- data.frame(height = c(height_female, height_male), gender
= c(female_character, male_character))
sim_data <- as.data.table(sim_data)
head(sim_data)
```

```
  height   gender
1 162.0804 female
2 169.6521 female
3 173.9911 female
4 175.0827 female
5 169.5079 female
6 166.2232 female
```

```r
sim_data %>%
    with(hist(height, main="Histogram of all genders"))
```




```r
options(repr.plot.width = 7, repr.plot.height = 5)
sim_data %>%
    ggplot(aes(x=height)) +
    geom_histogram(bins=30) +
    facet_wrap(~gender)
```



Control variables (from one to two, then to three)

- univariate analysis (one variable)
- bivariate analysis (two variable)
- multivatte anaysis (more than two)

Let's review what we have done:

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

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

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

$$weight = 55 + 0.1 * height$$

1. add noise into the data because there is no perfect thing in the real life (except for the GOD)

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

1. bring one more variable into our analysis, let's say gender (female/male)
2. for female, the distribution of height might be different from male

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

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

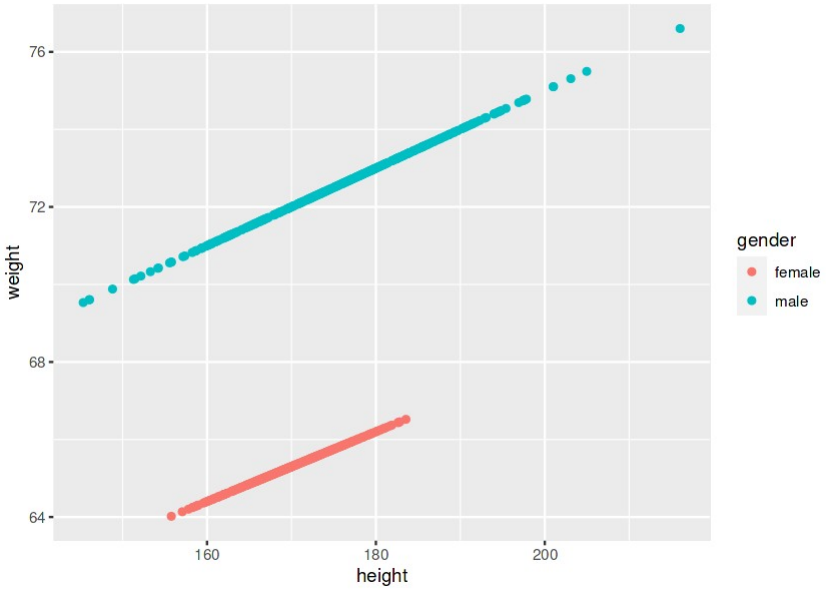
$$weight_f = 50 + 0.09 * height_f; weight_m = 55 + 0.1 * height_m$$

You can see that the complexity has already kicks in even for onlhy three variables.

```
head(sim_data)
```

```
  height   gender
1 162.0804 female
2 169.6521 female
3 173.9911 female
4 175.0827 female
5 169.5079 female
6 166.2232 female
```

```r
# add weight
sim_data %>%
    #[i, j, by]
    .[, weight := ifelse(gender=="female", 50 + 0.09*height, 55 + 0.1
* height)] %>%
    ggplot(aes(x=height, y=weight, color=gender)) +
    geom_point()
```




```r
# generate height for female
height_female <- rnorm(450, 170, 5)
female_character <- rep("female", 450)
height_male <- rnorm(450, 175, 10)
male_character <- rep('male', 450)
# put them together as a data.frame and then conver it to the
data.table
sim_data <- data.frame(height = c(height_female, height_male), gender
= c(female_character, male_character))
sim_data <- as.data.table(sim_data)

sim_data %>%
    #[i, j, by]
    .[, weight := ifelse(gender=="female", 50 + 0.09 *height, 55 + 0.1
* height)] -> sim_data2
```

```
str(sim_data2)
```

```
Classes ‘data.table’ and 'data.frame':900 obs. of  3 variables:
 $ height: num  163 174 171 160 170 ...
 $ gender: chr  "female" "female" "female" "female" ...
 $ weight: num  64.7 65.6 65.4 64.4 65.3 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

```
head(sim_data2)
```

```
  height   gender weight
1 162.7924 female 64.65132
2 173.6455 female 65.62809
3 171.0662 female 65.39596
4 160.3194 female 64.42875
5 169.9002 female 65.29102
6 172.0538 female 65.48484
```

We now assume that relationship between `weight` and `height` following the simple linear one:

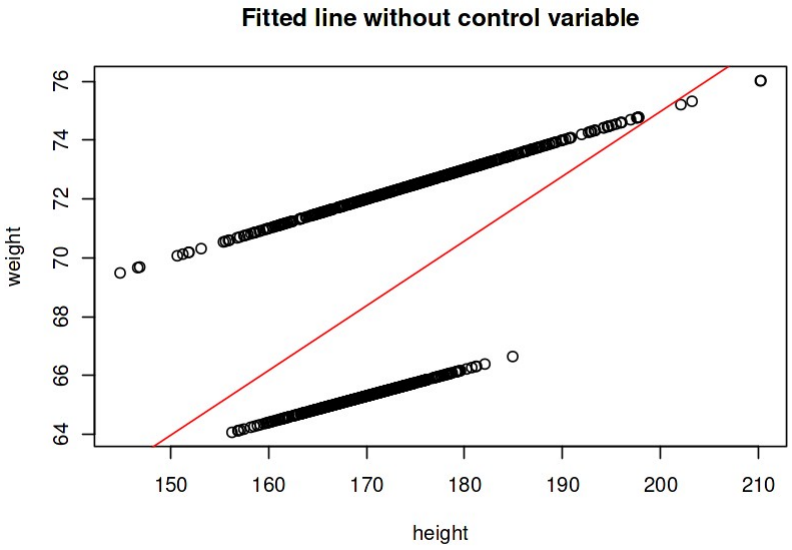
$$weight = \beta_0 + \beta_1 height + \epsilon; \epsilon \sim N(0, sd)$$

```
# sim_data2 has gender property
sm3 <- lm(weight ~ height, data=sim_data2)
stargazer(sm3, type="text")
```

```
================================================
                        Dependent variable:
                    ---------------------------
                              weight
------------------------------------------------
height                       0.220***
                              (0.012)

Constant                     30.933***
                              (2.148)

------------------------------------------------
Observations                    900
R2                             0.259
Adjusted R2                    0.258
Residual Std. Error       3.191 (df = 898)
F Statistic           313.091*** (df = 1; 898)
================================================
Note:                *p<0.1; **p<0.05; ***p<0.01
```

```
sim_data2 %>%
    with(plot(height, weight, main = "Fitted line without control
```

```
variable"))
abline(sm3, col='red')
```



Control variable

Control variable is a variable that is included in a statistical or research analysis to account for potential confounding factors or to assess the relationship between the independent and dependent variables **while holding other factors constant**. In the regression analysis we control different factors by adding control variables:

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

```
head(sim_data2)
```

```
  height   gender weight
1 162.7924 female 64.65132
2 173.6455 female 65.62809
3 171.0662 female 65.39596
4 160.3194 female 64.42875
5 169.9002 female 65.29102
6 172.0538 female 65.48484
```

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

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

$$weight = 48.63 + 0.098 * height + 6.7 * gender_m$$

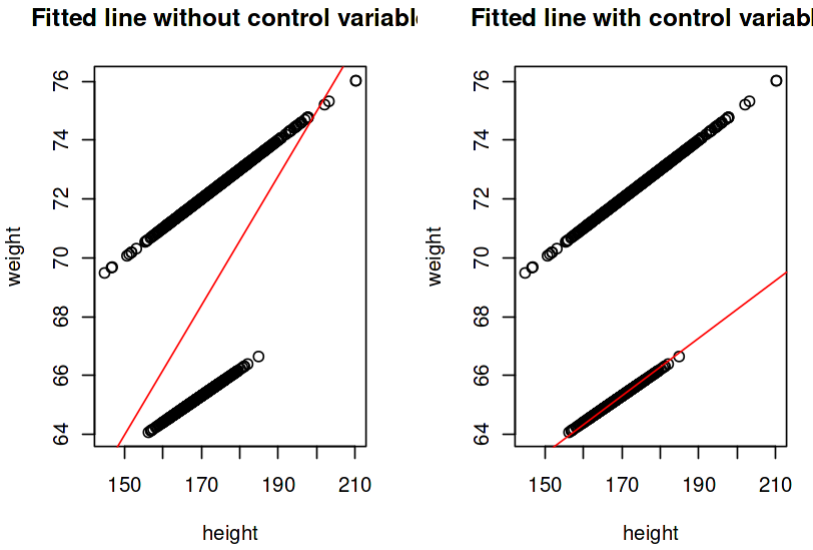
if $gender_m = 1$ we will have

$$weight = 48.63 + 0.098 * height + 6.7 = 55.337 + 0.098 * height$$

```r
# adding gender as control variable
sm4 <- lm(weight ~ height + gender, data=sim_data2)
stargazer(sm3, sm4, type="text")
```

```
==================================================================
=====
                                       Dependent variable:


-------------------------------------------------------
                                             weight

                              (1)                          (2)

------------------------------------------------------------------
-----
height                      0.220***                     0.098***

                            (0.012)                      (0.0001)



gendermale                                               6.707***

                                                         (0.002)



Constant                   30.933***                    48.621***

                            (2.148)                      (0.022)



------------------------------------------------------------------
-----
Observations                  900                          900

R2                           0.259                        1.000

Adjusted R2                  0.258                        1.000

Residual Std. Error     3.191 (df = 898)            0.032 (df = 897)

F Statistic         313.091*** (df = 1; 898) 6,114,846.000*** (df = 2;
897)
==================================================================
=====
```

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

par(mfrow = c(1, 2))
sim_data2 %>%
    with(plot(height, weight, main = "Fitted line without control
variable"))
abline(sm3, col='red')
sim_data2 %>%
    with(plot(height, weight, main = "Fitted line with control
variable"))
abline(sm4, col='red')

Warning message in abline(sm4, col = "red"):
“only using the first two of 3 regression coefficients”
```



Summary

- univariate (one- D)
- bivariate (two: D and A)
- multivariate (more than two, such as three: D - A and B)

The relationship between `weight` (D) and `height` (A) is different for different genders (B). This kind of framework is very common for many regression analysis:

- **relationship between innovation (measured by patent numbers) and export intensity could be different for different industries**

  $$innovation = \beta_0 + \beta_1 exportIntensity + \beta_2 industry + \beta_3 firmSize + ... + \epsilon$$

Interpretation of regression analysis

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 normally **associated** with higher wage.

```
# install a new package called wooldridge
install.packages("wooldridge")
```

```
Installing package into '/home/zou/R/x86_64-pc-linux-gnu-library/4.2'
(as 'lib' is unspecified)
```

```
library(wooldridge)
# load the data
data("wage1")
# convert it to data.table
wage1 <- as.data.table(wage1)
```

```
Attaching package: 'wooldridge'


The following object is masked from 'package:MASS':

    cement
```

```
head(wage1)
```

```
  wage educ exper tenure nonwhite female married numdep smsa northcen
…
1 3.10 11    2     0     0        1      0       2      1    0
…
2 3.24 12   22     2     0        1      1       3      1    0
…
3 3.00 11    2     0     0        0      0       2      0    0
…
4 6.00  8   44    28     0        0      1       0      1    0
…
5 5.30 12    7     2     0        0      1       1      0    0
…
6 8.75 16    9     8     0        0      1       0      1    0
…
  trcommpu trade services profserv profocc clerocc servocc lwage
expersq
```

```
1 0        0     0        0        0       0       0       1.131402
4
2 0        0     1        0        0       0       1       1.175573
484
3 0        1     0        0        0       0       0       1.098612
4
4 0        0     0        0        0       1       0       1.791759
1936
5 0        0     0        0        0       0       0       1.667707
49
6 0        0     0        1        1       0       0       2.169054
81
  tenursq
1   0
2   4
3   0
4 784
5   4
6  64
```

As we can see that we have many variables, 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:

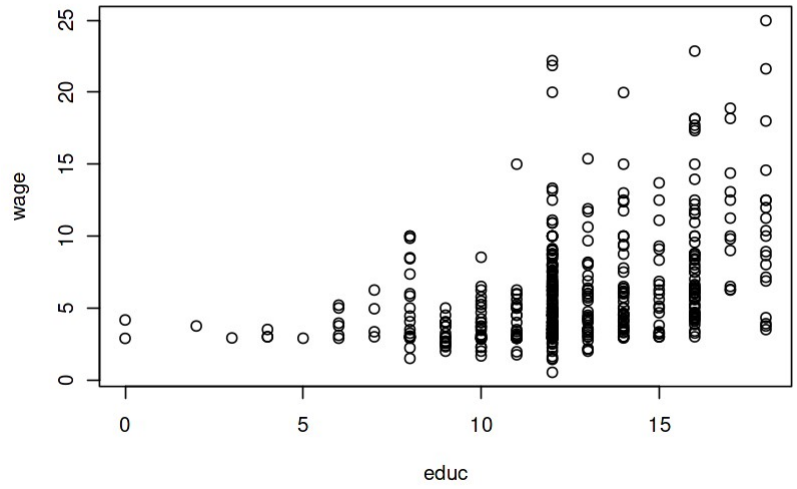
- wage: average hourly earnings
- educ: years of education
- exper: years of experience
- female: =1 if female otherwise =0

```
wage1 %>%
    # lwage = log transformation of wage
    .[, .(wage, educ, exper, female, lwage)] %>%
    head()
```

```
  wage educ exper female lwage
1 3.10 11    2    1      1.131402
2 3.24 12   22    1      1.175573
3 3.00 11    2    0      1.098612
4 6.00  8   44    0      1.791759
5 5.30 12    7    0      1.667707
6 8.75 16    9    0      2.169054
```

```
# univariate anlaysis
wage1 %>%
    with(hist(wage))
```


```r
# bivariate analysis
wage1 %>%
    with(plot(educ, wage))
```



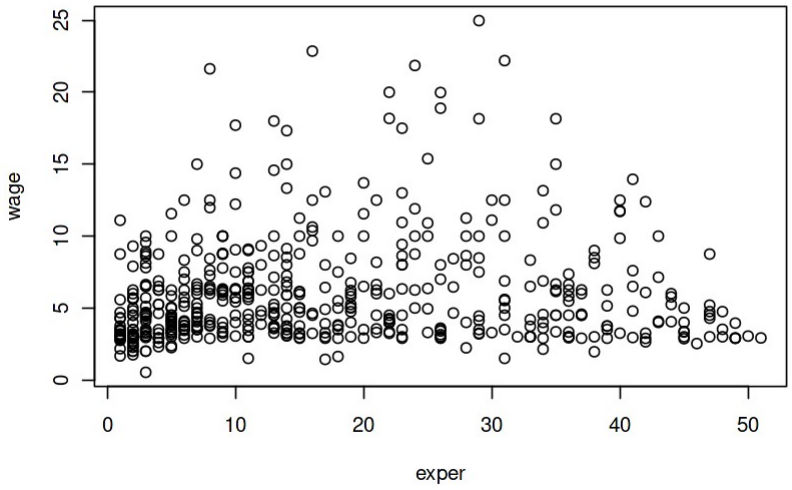

```r
# run regression
wage_reg1 <- lm(wage ~ educ, data=wage1)

stargazer(wage_reg1, type="text")
```

```
================================================
                        Dependent variable:
                    ---------------------------
                               wage
```

```
------------------------------------------------
educ                            0.541***
                                 (0.053)

Constant                         -0.905
                                 (0.685)

------------------------------------------------
Observations                       526
R2                                0.165
Adjusted R2                       0.163
Residual Std. Error         3.378 (df = 524)
F Statistic           103.363*** (df = 1; 524)
================================================
Note:                *p<0.1; **p<0.05; ***p<0.01
```

The model predicts that **an increase in education of 1 year is associated with an increase of 0.5411 dollar an hour in wage**. The intercept of −0.9049 literally means that a person with no education has a predicted hourly wage of −90 cent an hour. **This, of course, is silly. Therefore, we must interpret this equation with caution**.

```
# now we continue to explore relationship between wage and exper
# bivariate: experience and wage
wage1 %>%
    with(plot(exper, wage))
```



Here we notice that there is some **nonlinear relationship** between `wage` and `exper`, which is not surprising. The wage normally increases with experience, but it will stop after reaching a certain level. For instance, most people will not get a higher wage after working for 20 to 30 years (say after 60 years old).

```r
# let's run regression
wage_reg2 <- lm(wage ~ educ + exper, data=wage1)

stargazer(wage_reg2, type="text")
```

```
================================================
                        Dependent variable:
                    ---------------------------
                                wage
------------------------------------------------
educ                          0.644***
                               (0.054)

exper                         0.070***
                               (0.011)

Constant                      -3.391***
                               (0.767)

------------------------------------------------
Observations                     526
R2                              0.225
Adjusted R2                     0.222
Residual Std. Error       3.257 (df = 523)
F Statistic           75.990*** (df = 2; 523)
================================================
Note:                *p<0.1; **p<0.05; ***p<0.01
```

```r
# let's add non-linear term
wage_reg3 <- lm(wage ~ educ + exper + I(exper^2), data=wage1)

stargazer(wage_reg1, wage_reg2, wage_reg3, type="text")
```

```
=============================================================================
======================
                                                Dependent variable:


-----------------------------------------------------------------------------
--
                                                       wage

                              (1)                      (2)
(3)
-----------------------------------------------------------------------------
----------------------
educ                        0.541***                0.644***
0.595***
```

```
                                (0.053)                   (0.054)
(0.053)


exper                                                    0.070***
0.268***
                                                          (0.011)
(0.037)


I(exper2)
-0.005***

(0.001)


Constant                         -0.905                  -3.391***
-3.965***
                                (0.685)                   (0.767)
(0.752)


------------------------------------------------------------------------
----------------------
Observations                      526                       526
526
R2                               0.165                     0.225
0.269
Adjusted R2                      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<0.1; **p<0.05; ***p<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 an nonlinear relationship between experience and wage**. The wage will increase with experience, but it will stop after reaching a certain level.

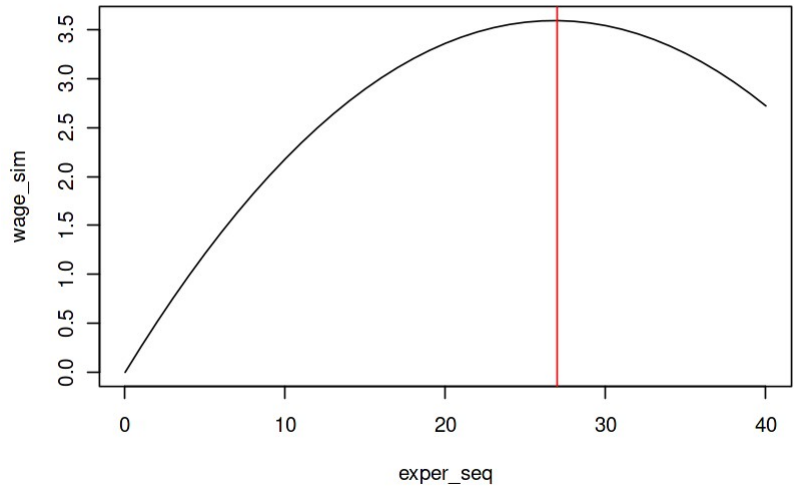
The coefficients for `exper` and `exper2` are 0.268 and − 0.005, now let's plot the relationship between `wage` and `exper` holding other factors constant. This means we can have the following equation:

$$wage = 0.268 \times exper - 0.005 \times exper^2$$

```
seq(0, 10, 1)
```

```
 [1]  0  1  2  3  4  5  6  7  8  9 10
```

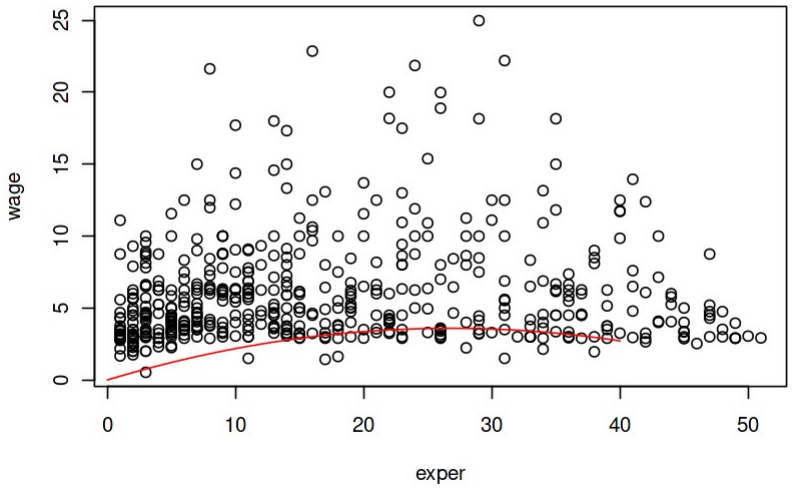
```
# simulate experience from 0 to 40 years
# seq = sequence generated from 0 to 30 with interval 1
exper_seq <- seq(0, 40, 1)
# ^2 means square
wage_sim <- 0.268 * exper_seq - 0.005 * exper_seq^2

# plot the relationship
plot(exper_seq, wage_sim, type="l")

# add vertical line
abline(v=27, col='red')
```



The model we estimated shows that the wage level stops increasing after 27 years of experience. You will not be aksed to do this kind of estimation in the exam. However, it is important to know how to interpret the regression results. **Normally, many variables related to age and experienceare have properities of nonlinear**. For instance, the relationship between `wage` and `age` is nonlinear. The wage will increase with age, but it will stop after reaching a certain level.

```
# we can put them together
wage1 %>%
    with(plot(exper, wage)) +
    lines(exper_seq, wage_sim, type="l", col='red')
```

```
integer(0)
```



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

- educ
- experience
- industry
- networking
- other factors

In the above model, the red curve was generated based on only factor `exper`:

$$wage = 0.268 \times exper - 0.005 \times exper^2$$

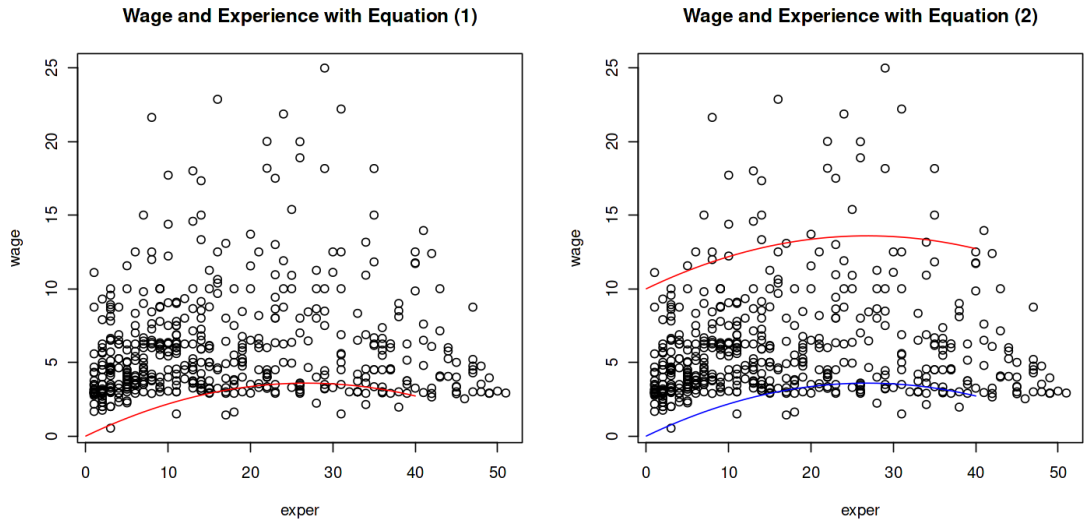
Here we only plot the relationship between `wage` and `exper` without considering other factors. Be aware that `wage` is the average hourly earnings, which is from the real data. `wage` is determined by many factors, such as education and industry. Now, image let's assume **the wage was determined by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

$$wage = 10 + 0.268 \times exper - 0.005 \times exper^2$$

This means that we have nonlinear relationship between `wage` and `exper`. 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 `exper` will be the same and everyone will be added 10 dollars per hour as a constant.

```
options(repr.plot.width = 12, repr.plot.height = 6)
par(mfrow = c(1, 2))
wage1 %>%
    with(plot(exper, wage, main = "Wage and Experience with Equation
(1)")) +
    lines(exper_seq, wage_sim, type="l", col='red')
```

```r
# let's simulate the above equation
wage_sim2 <- 10 + 0.268 * exper_seq - 0.005 * exper_seq^2

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

integer(0)

integer(0)
```



We have controled the experience, here is the short summary of the regression results:

give me a table with four columns and three rows

dependent variable	independent variable of interested	control variable1	control variable2
wage	education: 0.541***		
wage	eductation: 0.644***	exper: 0.070***	
wage	eductation: 0.595***	exper:0.268***	exper2: -0.005***

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.

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

```
================================================
                        Dependent variable:
                    ---------------------------
                               wage
------------------------------------------------
educ                          0.556***
                               (0.050)

exper                         0.255***
                               (0.035)

I(exper2)                     -0.004***
                               (0.001)

female                        -2.114***
                               (0.263)

Constant                      -2.319***
                               (0.739)

------------------------------------------------
Observations                     526
R2                               0.350
Adjusted R2                      0.345
Residual Std. Error        2.989 (df = 521)
F Statistic           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.

```
# now, let's put all tables together
stargazer(wage_reg2, wage_reg3, wage_reg4, type="text")
```

```
================================================================================
```

```
=====================
                                                Dependent variable:

-------------------------------------------------------------------------
-
                                                      wage

                              (1)                     (2)
(3)
-------------------------------------------------------------------------
---------------------
educ                        0.644***                0.595***
0.556***
                             (0.054)                 (0.053)
(0.050)


exper                       0.070***                0.268***
0.255***
                             (0.011)                 (0.037)
(0.035)


I(exper2)                                           -0.005***
-0.004***
                                                     (0.001)
(0.001)


female
-2.114***

(0.263)


Constant                    -3.391***               -3.965***
-2.319***
                             (0.767)                 (0.752)
(0.739)


-------------------------------------------------------------------------
---------------------
Observations                  526                     526
526
R2                           0.225                   0.269
0.350
Adjusted R2                  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
```

Robustness check

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.

Regression diagnostics

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 two ways:

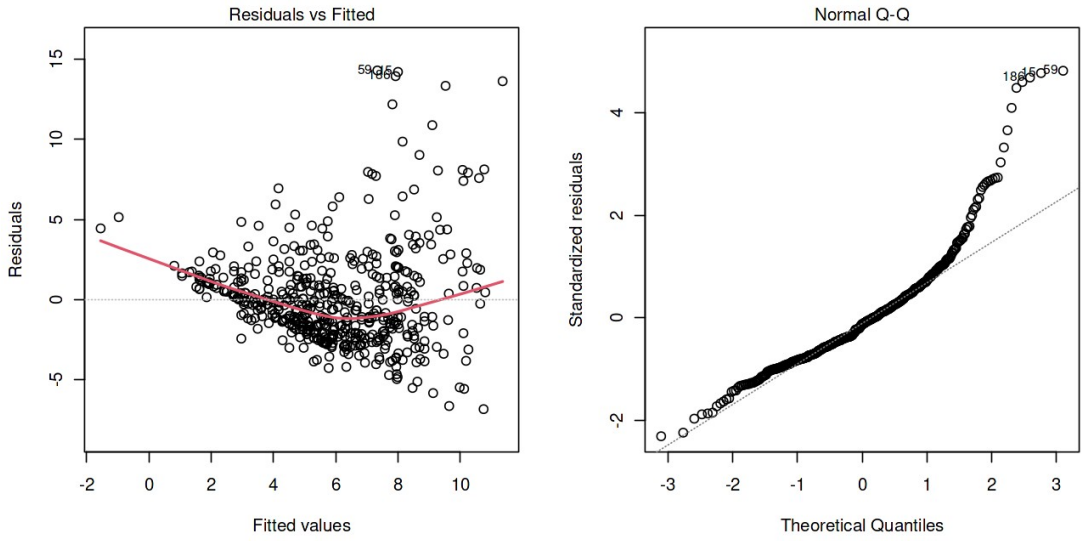
- **residual plot**: the residual plot is used to check whether the residuals are randomly distributed. If the residuals are randomly distributed, then we can say the results are reliable. Otherwise, we need to check the model specification. For instance, we might need to add more control variables to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

The residuals are just nosize we assumed in the model:

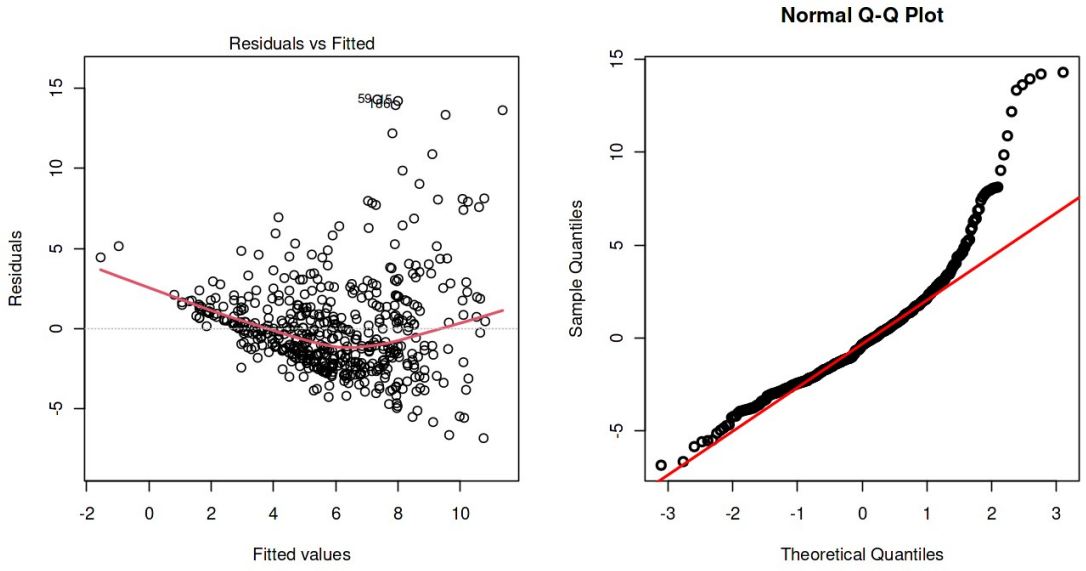
$$wage \quad ¿\beta_0+\beta_1 educ+\beta_2 exper+\beta_3 exper^2+\beta_4 femal$$

$$wage-\left(\beta_0+\beta_1 educ+\beta_2 exper+\beta_3 exper^2+\beta_4 female\right) \quad ¿\epsilon \sim N\left(0,\sigma^2\right)$$

```
# residual plot
options(repr.plot.width = 11, repr.plot.height = 6)
par(mfrow=c(1,2))
plot(wage_reg4, which=1, lwd=2)
plot(wage_reg4, which=2)
```




```
# residual plot
options(repr.plot.width = 11, repr.plot.height = 6)
par(mfrow=c(1,2))
plot(wage_reg4, which=1, lwd=2)
qqnorm(resid(wage_reg4),lwd=2)
qqline(resid(wage_reg4), col='red', lwd=2)
```



Log transformation

This only applies to continuous variables (like wage, R&D, etc. for variables like yes/no, female/male)

As we have discussed before, sometimes the dependent variable is not normally distributed. For instance, the wage is not normally distributed. In this case, we can use log transformation to make the dependent variable normally distributed.

The original model is:

$$wage = \beta_0 + \beta_1 \times educ + \beta_2 \times exper + \beta_3 \times exper^2 + \beta_4 \times female + \epsilon ; \epsilon \sim N(0, \sigma^2)$$

```
stargazer(wage_reg1, wage_reg2, wage_reg3, wage_reg4, type="text")
```

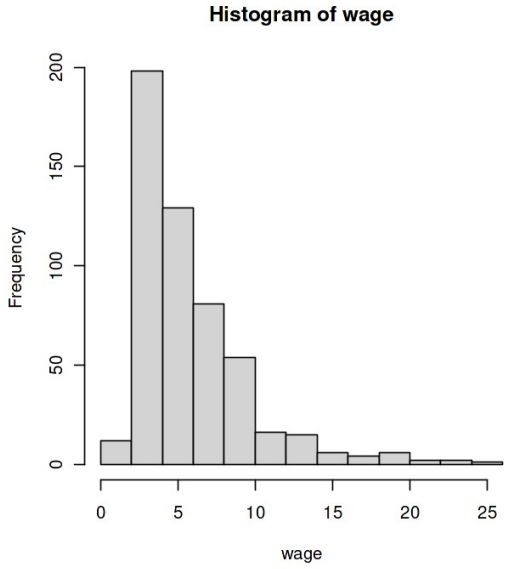
```
================================================================================
================================================
                                                                Dependent
variable:

--------------------------------------------------------------------------------
-------------------------
                                                                      wage

                              (1)                     (2)
(3)                     (4)
--------------------------------------------------------------------------------
-----------------------------------------------
educ                          0.541***                0.644***
0.595***                0.556***
                              (0.053)                 (0.054)
(0.053)                 (0.050)


exper                                                 0.070***
0.268***                0.255***
                                                      (0.011)
(0.037)                 (0.035)


I(exper2)
-0.005***               -0.004***

(0.001)                 (0.001)


female
-2.114***

(0.263)


Constant                      -0.905                  -3.391***
-3.965***               -2.319***
```

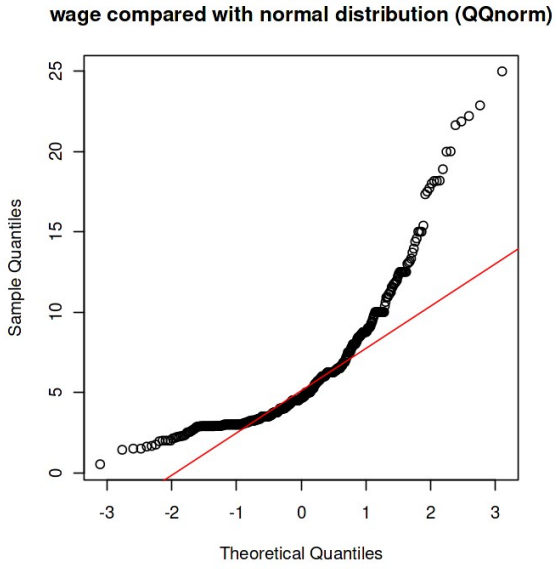
```
                                (0.685)                  (0.767)
(0.752)                  (0.739)


------------------------------------------------------------------------
-----------------------------------------
Observations                     526                      526
526                      526
R2                              0.165                    0.225
0.269                    0.350
Adjusted R2                     0.163                    0.222
0.265                    0.345
Residual Std. Error     3.378 (df = 524)        3.257 (df = 523)
3.166 (df = 522)        2.989 (df = 521)
F Statistic         103.363*** (df = 1; 524) 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
```

Now, let's use log transformation on the dependent variable:

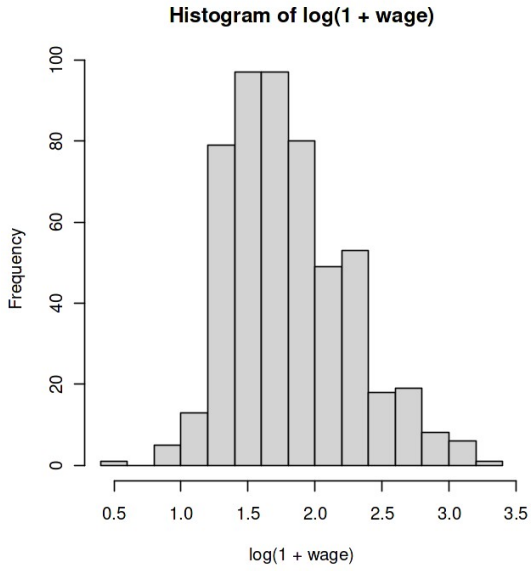
$$wage \quad ¿\beta_0+\beta_1\times educ+\beta_2\times exper+\beta_3\times exper^2+\beta_4\times female+\epsilon;\epsilon\sim N\left(0,\sigma^2\right)$$
$$\ln\left(wage\right) \quad ¿\beta_0+\beta_1\times educ+\beta_2\times exper+\beta_3\times exper^2+\beta_4\times female+\epsilon;\epsilon\sim N\left(0,\sigma^2\right)$$

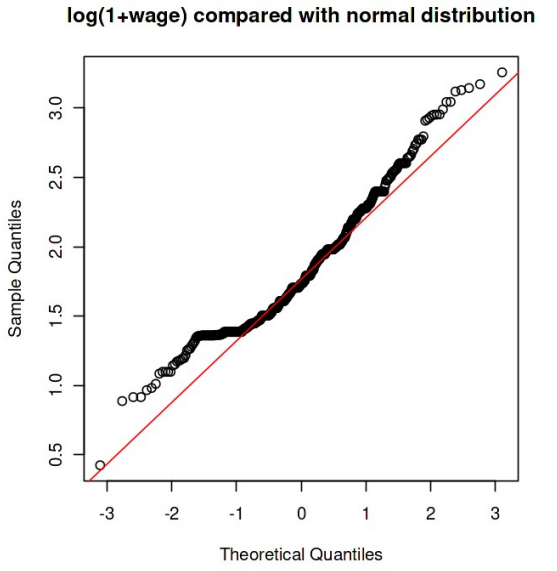
```
# univariate analysis
options(repr.plot.width = 11, repr.plot.height = 6)
par(mfrow=c(1,2))
wage1 %>%
    with(hist(wage))
qqnorm(wage1$wage, main="wage compared with normal distribution
(QQnorm)")
qqline(wage1$wage, col='red')
```






```
# log transformation for wage (depedent variable)
par(mfrow=c(1,2))
wage1 %>%
    with(hist(log(1+wage)))
qqnorm(log(1+wage1$wage), main="log(1+wage) compared with normal
distribution")
qqline(log(1+wage1$wage), col='red')
```





Now, we will fit with

$$\ln(wage) = \beta_0 + \beta_1 \times educ + \beta_2 \times exper + \beta_3 \times exper^2 + \beta_4 \times female + \epsilon; \epsilon \sim N(0, \sigma^2)$$

```r
# let's fit the model
wage_reg5 <- lm(log(1+wage) ~ educ + exper + I(exper^2) + female,
data=wage1)

stargazer(wage_reg5, type="text")
```

```
================================================
                        Dependent variable:
                    ---------------------------
                           log(1 + wage)
------------------------------------------------
educ                         0.071***
                              (0.006)

exper                        0.033***
                              (0.004)

I(exper2)                    -0.001***
                             (0.0001)

female                       -0.283***
                              (0.030)

Constant                     0.784***
                              (0.086)

------------------------------------------------
Observations                    526
R2                             0.402
Adjusted R2                    0.397
Residual Std. Error      0.346 (df = 521)
F Statistic           87.415*** (df = 4; 521)
================================================
Note:                *p<0.1; **p<0.05; ***p<0.01
```

Since we are dealing with log transformation now. The interpretation will be different. The following table gives the summary

	X	$\ln(X)$
Y	linear: $Y = \beta_0 + \beta_1 X$	linear-log: $Y = \beta_0 + \beta_1 \ln(X)$
	one unit change in X is associated with β_1 change in Y	one unit change in X is associated with $\beta_1/100$ change in Y
$\ln(Y)$	log-linear $\ln(Y) = \beta_0 + \beta_1 X$	log-log: $\ln(Y) = \beta_0 + \beta_1 \ln(X)$
	one unit change in X is	one percentage change in X

	X	$\ln(X)$
	associated with $100 \cdot \beta_1$	is associated with β_1
	percentage change in Y	percent change in Y

```
wage1 %>%
    .[, .(wage, educ, exper, lwage)] %>%
    head()
```

```
  wage educ exper lwage
1 3.10 11    2    1.131402
2 3.24 12   22    1.175573
3 3.00 11    2    1.098612
4 6.00  8   44    1.791759
5 5.30 12    7    1.667707
6 8.75 16    9    2.169054
```

```
wage1 %>%
    with(summary(wage))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.530   3.330   4.650   5.896   6.880  24.980
```

```
# suppose my wage is 5.896 dollar per hour
# extra year of education -> 5.896 + 0.556
print(5.896 + 0.556)
# 0.071 * 100 = 7.1 percent
# extra year of education -> 5.896 * (1 + 0.071)
print(5.896 * (1 + 0.071))
```

```
[1] 6.452
[1] 6.314616
```

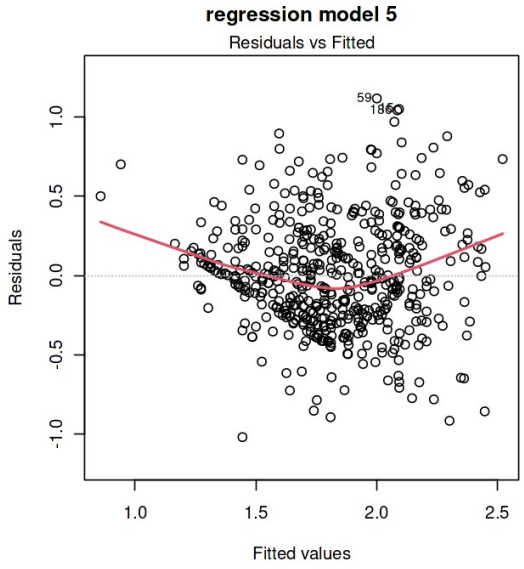
```
# put everything together
stargazer(wage_reg2, wage_reg3, wage_reg4, wage_reg5, type="text")
```

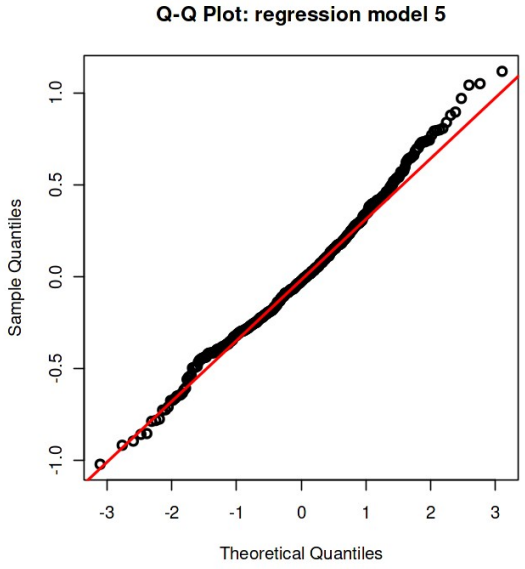
```
===============================================================================
================================================
                                                                Dependent
variable:

-------------------------------------------------------------------------------
-------------------------
                                                              wage
log(1 + wage)
                                (1)                           (2)
(3)                        (4)
-------------------------------------------------------------------------------
-----------------------------------------
educ                          0.644***                      0.595***
0.556***                   0.071***
```

```
                            (0.054)                  (0.053)
(0.050)                  (0.006)


exper                       0.070***                 0.268***
0.255***                 0.033***
                            (0.011)                  (0.037)
(0.035)                  (0.004)


I(exper2)                                            -0.005***
-0.004***                -0.001***
                                                      (0.001)
(0.001)                 (0.0001)


female
-2.114***                -0.283***

(0.263)                  (0.030)


Constant                    -3.391***                -3.965***
-2.319***                0.784***
                            (0.767)                  (0.752)
(0.739)                  (0.086)


------------------------------------------------------------------------
-----------------------------------------
Observations                  526                      526
526                      526
R2                           0.225                    0.269
0.350                    0.402
Adjusted R2                  0.222                    0.265
0.345                    0.397
Residual Std. Error    3.257 (df = 523)        3.166 (df = 522)
2.989 (df = 521)         0.346 (df = 521)
F Statistic         75.990*** (df = 2; 523) 64.109*** (df = 3; 522)
70.170*** (df = 4; 521) 87.415*** (df = 4; 521)
========================================================================
=========================================
Note:
*p<0.1; **p<0.05; ***p<0.01
```

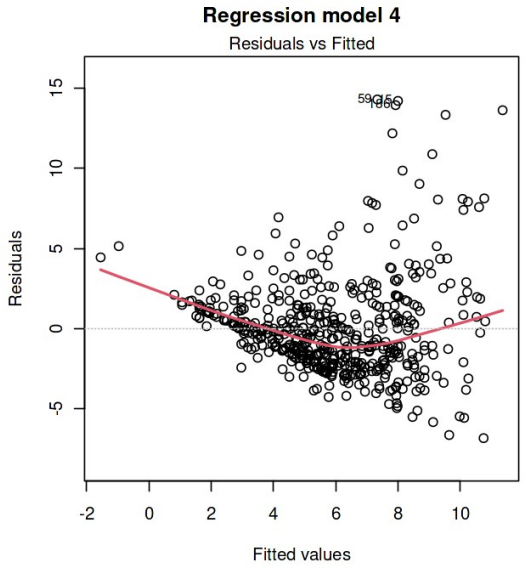
```r
# residual plot for log transformation
options(repr.plot.width = 11, repr.plot.height = 6)
par(mfrow=c(1,2))
plot(wage_reg5, which=1, lwd=2, main="regression model 5")
```

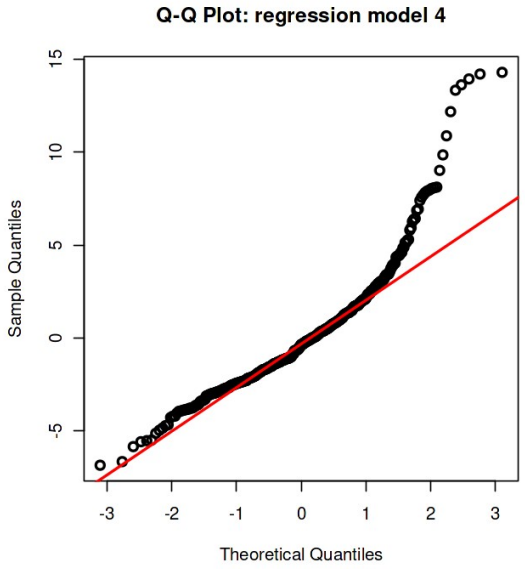
```
qqnorm(resid(wage_reg5),lwd=2, main="Q-Q Plot: regression model 5")
qqline(resid(wage_reg5), col='red', lwd=2)
```






```
# residual plot without log transformation
options(repr.plot.width = 11, repr.plot.height = 6)
par(mfrow=c(1,2))
plot(wage_reg4, which=1, lwd=2, main="Regression model 4")
qqnorm(resid(wage_reg4),lwd=2, main="Q-Q Plot: regression model 4")
qqline(resid(wage_reg4), col='red', lwd=2)
```





Summary

- why we need control variables

- check whether the coefficient is consistent or not by adding different control variables
- using dummy variable - gender discrimination
- why we need log transformation
- how to interpret the log transformation

The another example

Now, let's use another example to illustrate the regression analysis. The data is about the relationship between research and development expenditure and the sales.

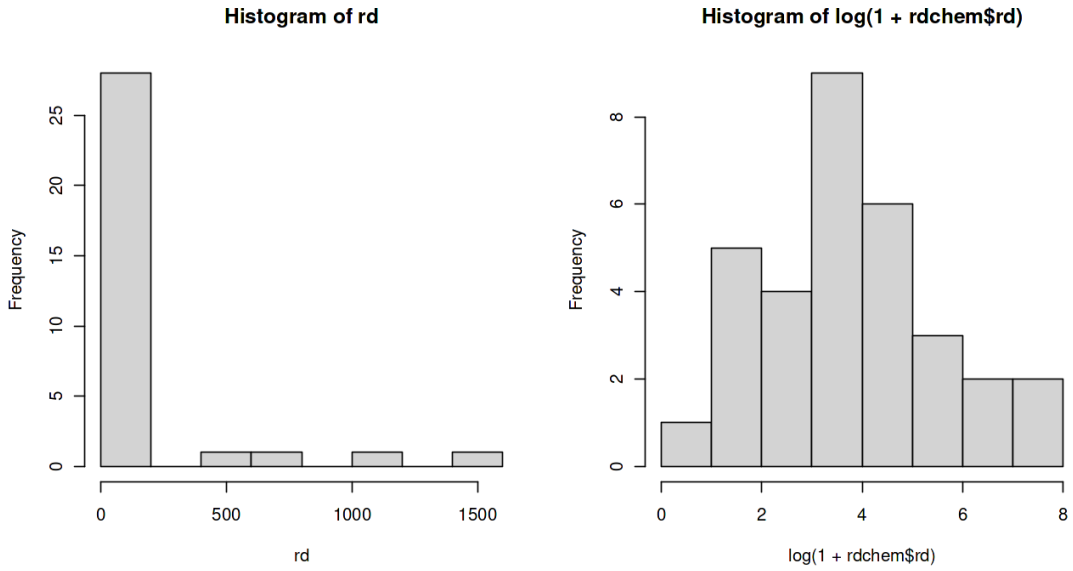
```
data(rdchem)
# convert it to data.table
rdchem <- as.data.table(rdchem)
str(rdchem)
head(rdchem)
```

```
Classes ‘data.table’ and 'data.frame':32 obs. of  8 variables:
 $ rd      : num  430.6 59 23.5 3.5 1.7 ...
 $ sales   : num  4570 2830 597 134 42 ...
 $ profits : num  186.9 467 107.4 -4.3 8 ...
 $ rdintens: num  9.42 2.08 3.94 2.62 4.05 ...
 $ profmarg: num  4.09 16.5 18 -3.22 19.05 ...
 $ salessq : num  20886730 8008900 356170 17849 1764 ...
 $ lsales  : num  8.43 7.95 6.39 4.89 3.74 ...
 $ lrd     : num  6.065 4.078 3.157 1.253 0.531 ...
 - attr(*, "time.stamp")= chr "25 Jun 2011 23:03"
 - attr(*, ".internal.selfref")=<externalptr>
```

```
  rd    sales  profits rdintens profmarg  salessq     lsales   lrd

1 430.6 4570.2 186.9   9.421906  4.089536 20886730.00 8.427312
6.0651798
2  59.0 2830.0 467.0   2.084806 16.501766  8008900.00 7.948032
4.0775375
3  23.5  596.8 107.4   3.937668 17.995979   356170.22 6.391582
3.1570003
4   3.5  133.6  -4.3   2.619760 -3.218563    17848.96 4.894850
1.2527629
5   1.7   42.0   8.0   4.047619 19.047619     1764.00 3.737670
0.5306283
6   8.4  390.0  47.3   2.153846 12.128205   152100.00 5.966147
2.1282318
```

```
# univariate analysis

par(mfrow=c(1,2))
rdchem %>%
    with(hist(rd))
```

```
hist(log(1+rdchem$rd))
```



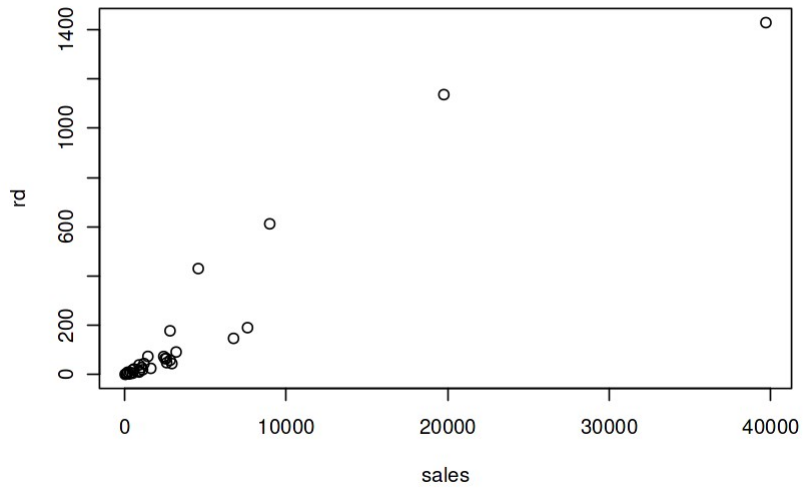
We will fit two mdels:

$$R \wedge D = \beta_0 + \beta_1 \times sales + \beta_2 \times profitMargin + \epsilon; \epsilon \sim N(0, \sigma^2)$$

and

$$\ln(R \wedge D) = \beta_0 + \beta_1 \times sales + \beta_2 \times profitMargin + \epsilon; \epsilon \sim N(0, \sigma^2)$$

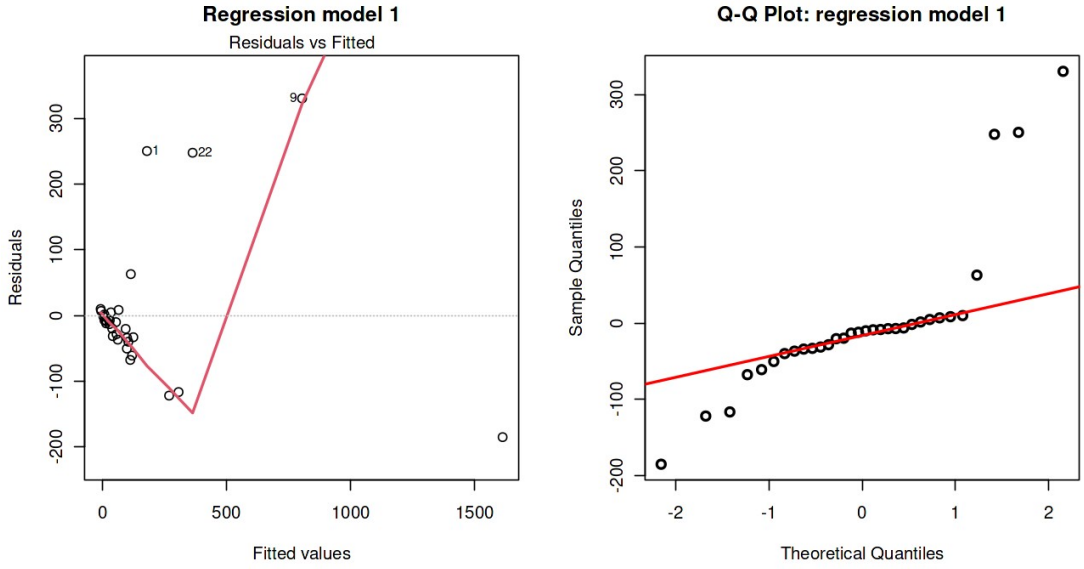
```
options(repr.plot.width = 7, repr.plot.height = 5)
rdchem %>%
    with(plot(sales, rd))
```



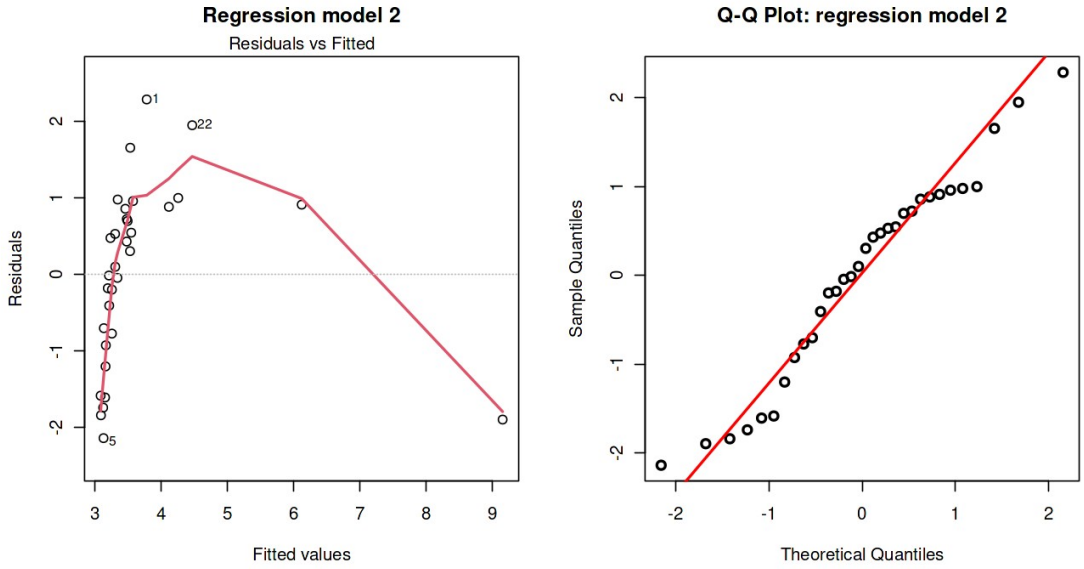
```r
rd_reg1 <- lm(rd ~ sales + profmarg, data=rdchem)
rd_reg2 <- lm(log(1+rd) ~ sales + profmarg, data=rdchem)

stargazer(rd_reg1, rd_reg2, type="text")
```

```
===============================================================
                                   Dependent variable:
                               ----------------------------
                                     rd        log(1 + rd)
                                    (1)            (2)
---------------------------------------------------------------
sales                             0.041***      0.0002***
                                  (0.002)       (0.00003)

profmarg                           0.857          0.003
                                  (2.605)        (0.030)

Constant                          -9.013        3.076***
                                  (33.025)       (0.380)

---------------------------------------------------------------
Observations                        32             32
R2                                 0.902          0.495
Adjusted R2                        0.895          0.460
Residual Std. Error (df = 29)     105.024         1.208
F Statistic (df = 2; 29)         133.599***     14.224***
===============================================================
Note:                           *p<0.1; **p<0.05; ***p<0.01
```

```r
# check residual plot
options(repr.plot.width = 11, repr.plot.height = 6)
par(mfrow=c(1,2))
plot(rd_reg1, which=1, lwd=2, main="Regression model 1")
qqnorm(resid(rd_reg1),lwd=2, main="Q-Q Plot: regression model 1")
qqline(resid(rd_reg1), col='red', lwd=2)
```




```
options(repr.plot.width = 11, repr.plot.height = 6)
par(mfrow=c(1,2))
plot(rd_reg2, which=1, lwd=2, main="Regression model 2")
qqnorm(resid(rd_reg2),lwd=2, main="Q-Q Plot: regression model 2")
qqline(resid(rd_reg2), col='red', lwd=2)
```



Multicollienarity

Multicollienarity is a very important concept in regression analysis. It means that there is a strong correlation between two or more independent variables. For instance, if we have two independent variables, X_1 and X_2, and they are strongly correlated, then we say there is a multicollinearity in the model.

This will cause the following problems:

- it increases the variance of the coefficients and makes the coefficients less reliable.
- it makes the interpretation of the coefficients difficult.

We will simulate a dataset to illustrate the problem of multicollinearity.

We have simulated the relationship between `weight` and `height` as follows:

$$weight = \beta_0 + \beta_1 \times height + \epsilon; \epsilon \sim N(0, \sigma^2)$$

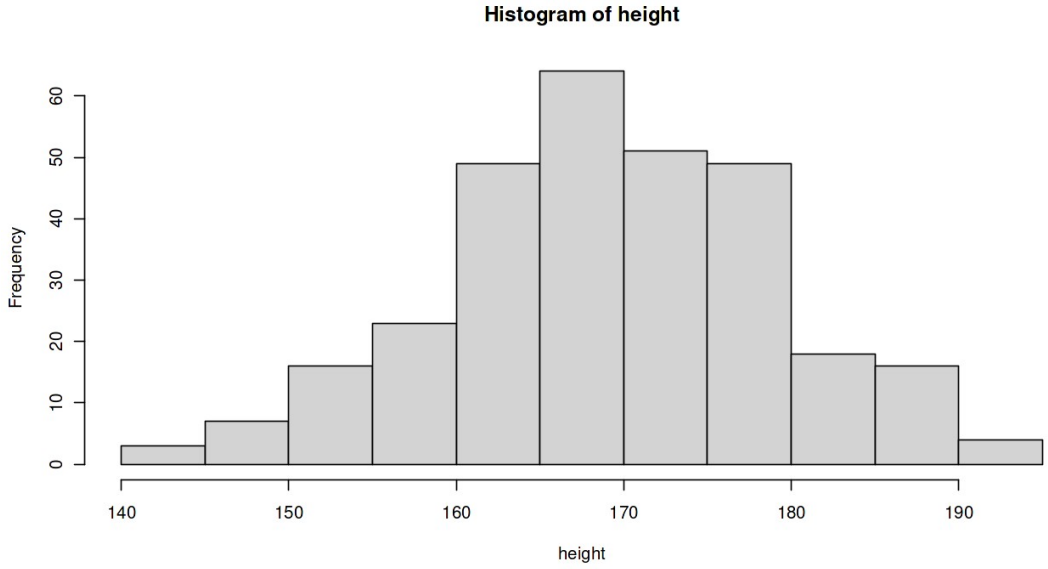
Now, let's add another variable called `handspan` to the model. The `handspan` is strongly correlated with `height`. On average, human's handspan is about 5% to 6% of the height. For instance, if the height is 170cm, then the handspan is about 9 cm. Therefore, we can simulate the `handspan` as follows:

$$handspan = 0.3 + 0.05 \times height + \epsilon; \epsilon \sim N(0, \sigma^2)$$

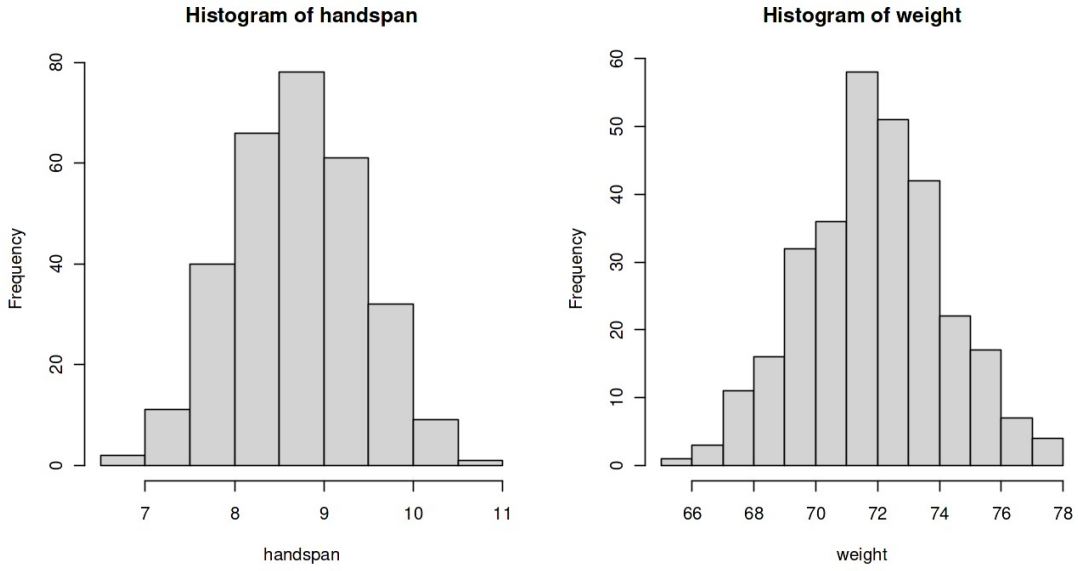
At the same time, we assume that the `weight` is related to `height` as follows:

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

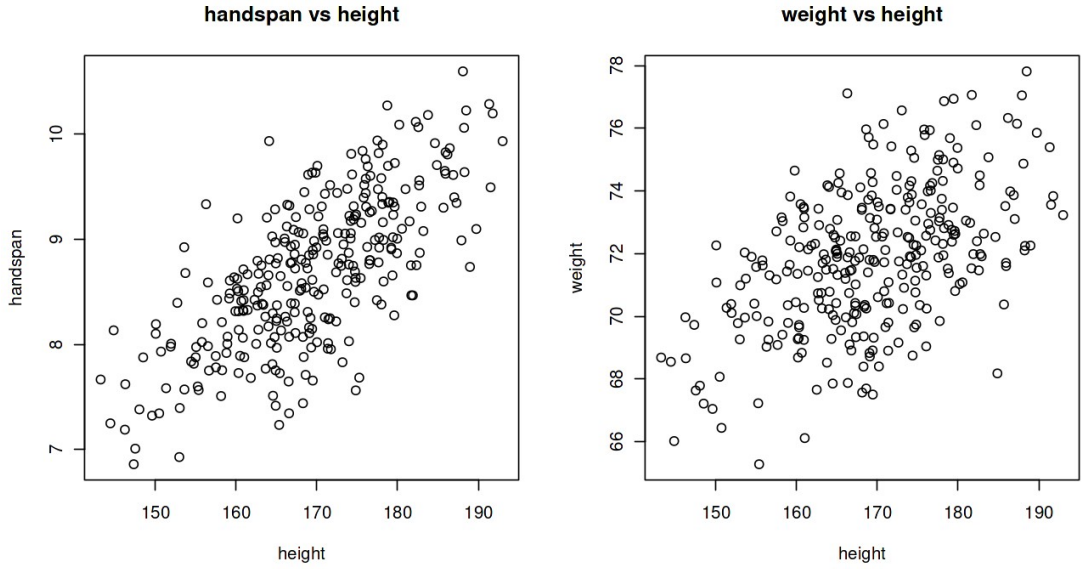
```
# generate height data with 300 observations mean = 170, sd = 10
height <- rnorm(300, 170, 10)
hist(height)
```




```
# generate handspan data
handspan <- 0.3 + 0.05 * height + rnorm(300, 0, 0.5)
# generate weight data
weight <- 55 + 0.1 * height + rnorm(300, 0, 2)

options(repr.plot.width = 11, repr.plot.height = 6)
par(mfrow=c(1,2))
hist(handspan)
hist(weight)
```




```
par(mfrow=c(1,2))
plot(height, handspan, main="handspan vs height")
plot(height, weight, main="weight vs height")
```




```
# put everything together
hhw_data <- data.table(height, handspan, weight)
head(hhw_data)
```

```
  height   handspan weight
1 169.6650 8.983400 71.73677
2 168.6164 7.710540 75.96892
3 188.9704 8.737125 72.25947
4 176.3088 9.693411 72.95468
5 156.6497 7.751262 71.31004
6 179.4688 9.232054 72.60444
```

```
# check correlation
hhw_data %>%
    with(cor(height, handspan))
```

```
[1] 0.7133502
```

Now, let's fit with two models:

$$weight = \beta_0 + \beta_1 \cdot height + \epsilon ; \epsilon \sim N\left(0, \sigma^2\right)$$

and

$$weight = \beta_0 + \beta_1 \cdot height + \beta_2 \cdot handspan + \epsilon ; \epsilon \sim N\left(0, \sigma^2\right)$$

```
hhw_reg1 <- lm(weight ~ height, data=hhw_data)
hhw_reg2 <- lm(weight ~ height + handspan, data=hhw_data)
hhw_reg3 <- lm(weight ~ handspan, data=hhw_data)

stargazer(hhw_reg1, hhw_reg2, hhw_reg3, type="text")
```

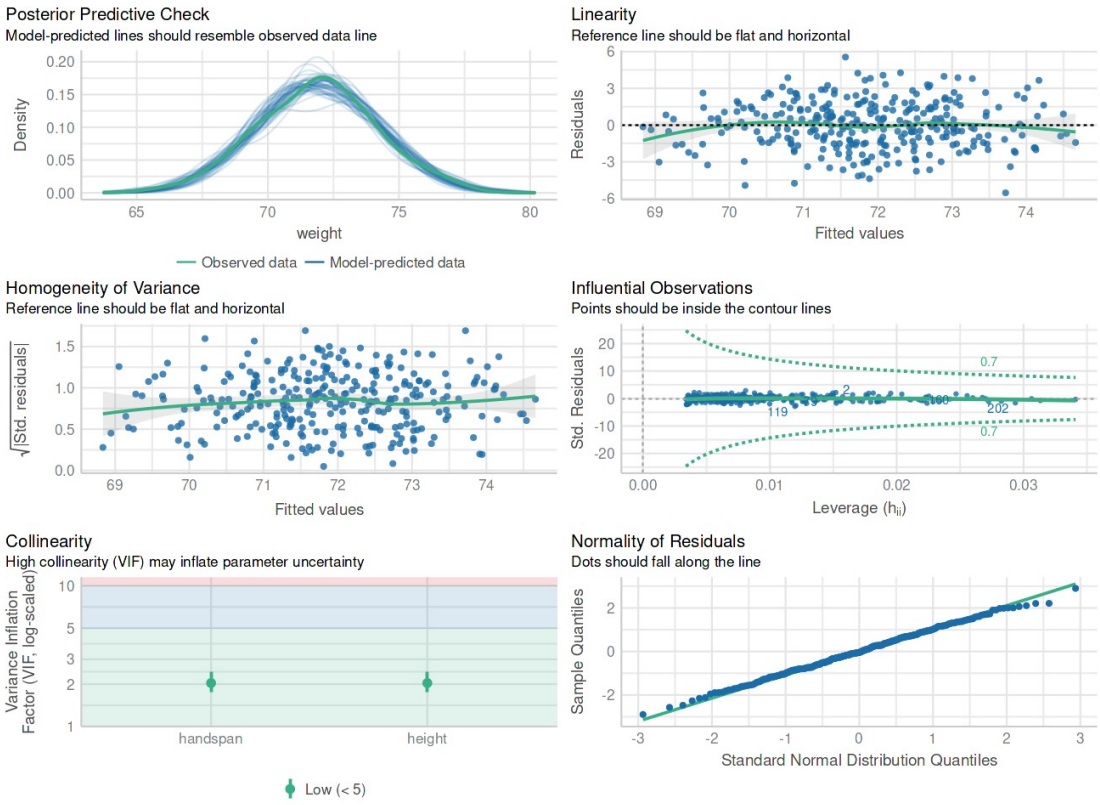
```
================================================================================
                                                Dependent variable:

--------------------------------------------------------------------------------
                                                       weight

                              (1)                      (2)                      (3)
--------------------------------------------------------------------------------
height                      0.117***                 0.114***                 0.070
                            (0.011)                  (0.016)                  (0.224)

handspan                                                                      1.206***
                                                                              (0.169)

Constant                    51.997***                51.993***                61.390***
                            (1.896)                  (1.899)                  (1.476)

--------------------------------------------------------------------------------
Observations                  300                      300                      300
R2                           0.270                    0.270                    0.146
Adjusted R2                  0.268                    0.265                    0.143
Residual Std. Error     1.945 (df = 298)         1.947 (df = 297)         2.104 (df = 298)
F Statistic         110.276*** (df = 1; 298) 55.020*** (df = 2; 297) 50.801*** (df = 1; 298)
================================================================================
Note:
*p<0.1; **p<0.05; ***p<0.01
```

```
install.packages("performance")
install.packages("see")
install.packages("patchwork")
```

```
Installing package into ‘/home/zou/R/x86_64-pc-linux-gnu-library/4.2’
(as ‘lib’ is unspecified)

also installing the dependencies ‘insight’, ‘datawizard’


Installing package into ‘/home/zou/R/x86_64-pc-linux-gnu-library/4.2’
(as ‘lib’ is unspecified)

also installing the dependencies ‘bayestestR’, ‘correlation’,
‘effectsize’, ‘modelbased’, ‘parameters’


Installing package into ‘/home/zou/R/x86_64-pc-linux-gnu-library/4.2’
(as ‘lib’ is unspecified)
```

```
library("performance")
```

```
# test for multicollinearity
options(repr.plot.width = 11, repr.plot.height = 8)
check_model(hhw_reg2)
```




```
# another example
elem_data <- fread("https://shorturl.at/awLNT")
head(elem_data)
```

```
  snum dnum api00 api99 growth meals ell yr_rnd mobility acs_k3 … hsg
some_col
1  906 41   693   600   93     67     9  0      11       16     …  0
0
2  889 41   570   501   69     92    21  0      33       15     …  0
0
3  887 41   546   472   74     97    29  0      36       17     …  0
0
4  876 41   571   487   84     90    27  0      27       20     … 45
9
5  888 41   478   425   53     89    30  0      44       18     … 50
0
6 4284 98   858   844   14     10     3  0      10       20     …  8
24
  col_grad grad_sch avg_ed full emer enroll mealcat collcat
1  0         0          NA    76  24   247    2       1
2  0         0          NA    79  19   463    3       1
3  0         0          NA    68  29   395    3       1
4  9         0        1.91    87  11   418    3       1
5  0         0        1.50    87  13   520    3       1
6 36        31        3.89   100   0   343    1       2
```

```
str(elem_data)
```

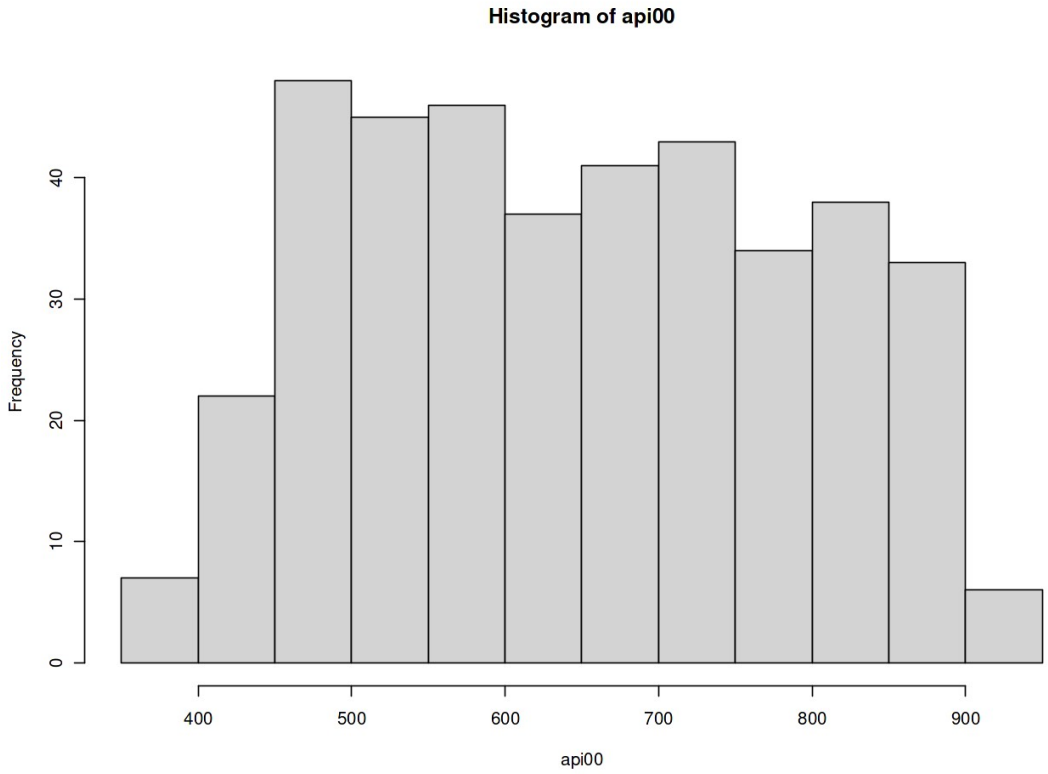
```
Classes ‘data.table’ and 'data.frame':400 obs. of  22 variables:
 $ snum    : int  906 889 887 876 888 4284 4271 2910 2899 2887 ...
 $ dnum    : int  41 41 41 41 41 98 98 108 108 108 ...
 $ api00   : int  693 570 546 571 478 858 918 831 860 737 ...
 $ api99   : int  600 501 472 487 425 844 864 791 838 703 ...
 $ growth  : int  93 69 74 84 53 14 54 40 22 34 ...
 $ meals   : int  67 92 97 90 89 10 5 2 5 29 ...
 $ ell     : int  9 21 29 27 30 3 2 3 6 15 ...
 $ yr_rnd  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ mobility: int  11 33 36 27 44 10 16 44 10 17 ...
 $ acs_k3  : int  16 15 17 20 18 20 19 20 20 21 ...
 $ acs_46  : int  22 32 25 30 31 33 28 31 30 29 ...
 $ not_hsg : int  0 0 0 36 50 1 1 0 2 8 ...
 $ hsg     : int  0 0 0 45 50 8 4 4 9 25 ...
 $ some_col: int  0 0 0 9 0 24 18 16 15 34 ...
 $ col_grad: int  0 0 0 9 0 36 34 50 42 27 ...
 $ grad_sch: int  0 0 0 0 0 31 43 30 33 7 ...
 $ avg_ed  : num  NA NA NA 1.91 1.5 ...
 $ full    : int  76 79 68 87 87 100 100 96 100 96 ...
 $ emer    : int  24 19 29 11 13 0 0 2 0 7 ...
 $ enroll  : int  247 463 395 418 520 343 303 1513 660 362 ...
 $ mealcat : int  2 3 3 3 3 1 1 1 1 1 ...
 $ collcat : int  1 1 1 1 1 2 2 2 2 3 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

- `api00` : grade API score in 2000
- `acs_k3`: average class size
- `avg_ed`: average education degree of parents
- `grad_sch`: parent graduation school
- `col_grad`: parent colledge degree
- `some_col`: parent some college

```
elem_data %>%
    .[, .(api00, acs_k3, avg_ed, grad_sch, col_grad, some_col)] %>%
    head()
```

```
  api00 acs_k3 avg_ed grad_sch col_grad some_col
1 693   16       NA    0        0        0
2 570   15       NA    0        0        0
3 546   17       NA    0        0        0
4 571   20     1.91    0        9        9
5 478   18     1.50    0        0        0
6 858   20     3.89   31       36       24
```

```
elem_data %>%
    .[, .(api00, acs_k3, avg_ed, grad_sch, col_grad, some_col)] %>%
    with(hist(api00))
```



Now I want to exam:

$$grade_{api00} = \beta_0 + \beta_1 acsK3 + \beta_2 avgEd + \beta_3 gradSch + \beta_4 colGrad + \beta_5 someCol + \epsilon ; \epsilon \sim N\left(0, \sigma^2\right)$$

```
elem_data <- fread("https://shorturl.at/awLNT")
elem_reg1 <- lm(api00 ~ acs_k3 + avg_ed +  grad_sch + col_grad +
some_col, data = elem_data)
stargazer(elem_reg1, type='text')
```

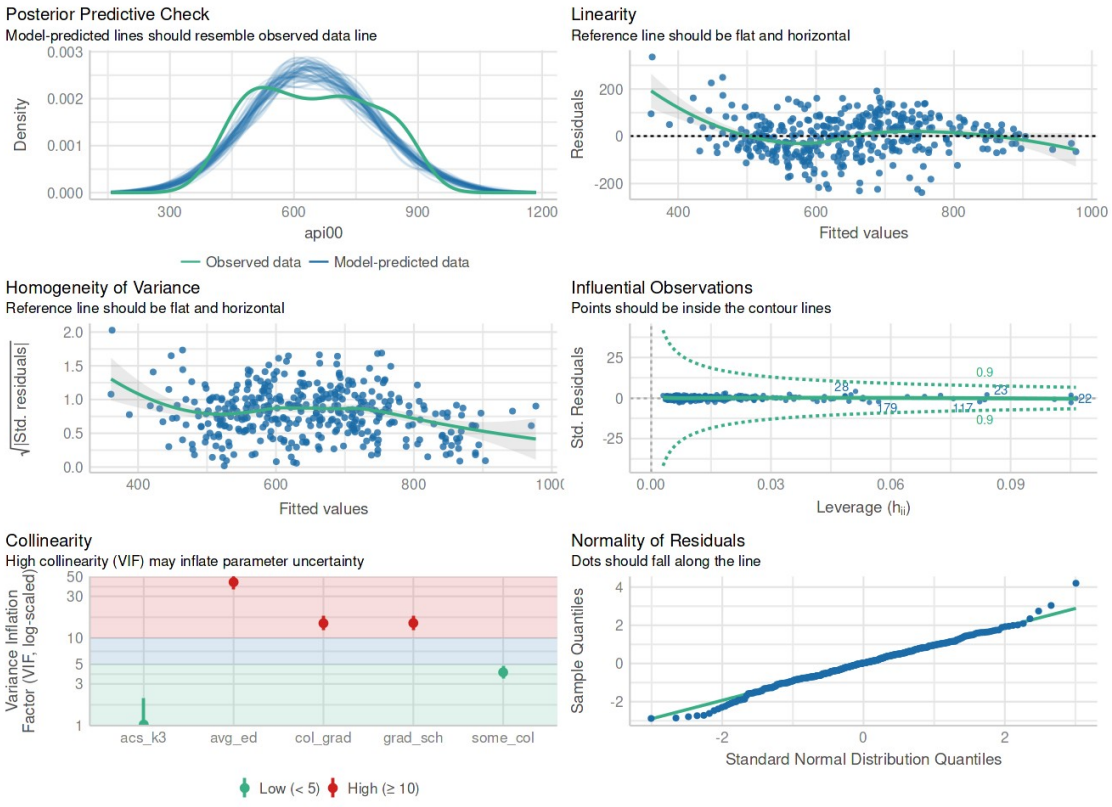
```
================================================
                        Dependent variable:
                    ---------------------------
                               api00
------------------------------------------------
acs_k3                        11.457***
                               (3.275)

avg_ed                       227.264***
                              (37.220)

grad_sch                      -2.091
                              (1.352)

col_grad                     -2.968***
                              (1.018)
```

```
some_col                        -0.760
                                (0.811)

Constant                        -82.609
                               (81.846)

------------------------------------------------
Observations                      379
R2                               0.658
Adjusted R2                      0.654
Residual Std. Error       83.861 (df = 373)
F Statistic           143.793*** (df = 5; 373)
================================================
Note:                *p<0.1; **p<0.05; ***p<0.01
```

```
install.packages("performance")
install.packages("see")
install.packages("patchwork")
```

```
library(performance)
```

```
options(repr.plot.width = 11, repr.plot.height = 8)
check_model(elem_reg1)
```



Summary

Assumptions	Diagnostic check
A1: linear relationship between y and x	check via plots
A2: independence of observations	check via plots
A3: $E(\epsilon \vee x) = 0$	check via plots
A4: $Var(\epsilon	x) = \sigma^2 $
A5: normal distribution of $\epsilon \sim N(0, \sigma^2)$	check via plots
A6: No correlation between x and ϵ	check via plots

High influential points

```
install.packages("wooldridge")
```

```
Installing package into '/home/zou/R/x86_64-pc-linux-gnu-library/4.2'
(as 'lib' is unspecified)
```

```
library(wooldridge)
data("infmrt")
str(infmrt)
```

```
'data.frame':   102 obs. of  12 variables:
 $ year   : int  1987 1990 1987 1990 1987 1990 1987 1990 1987 1990 ...
 $ infmort: num  8.3 6.2 7.8 7.1 8.5 ...
 $ afdcprt: int  52 62 11 21 20 25 234 282 42 52 ...
 $ popul  : int  1186 1228 1056 1109 547 563 5856 6016 986 1003 ...
 $ pcinc  : int  13996 17125 18083 21051 14267 17630 19131 22558 15683
18771 ...
 $ physic : int  173 178 186 200 244 253 322 337 244 254 ...
 $ afdcper: num  4.38 5.05 1.04 1.89 3.66 ...
 $ d90    : int  0 1 0 1 0 1 0 1 0 1 ...
 $ lpcinc : num  9.55 9.75 9.8 9.95 9.57 ...
 $ lphysic: num  5.15 5.18 5.23 5.3 5.5 ...
 $ DC     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ lpopul : num  7.08 7.11 6.96 7.01 6.3 ...
 - attr(*, "time.stamp")= chr "25 Jun 2011 23:03"
```

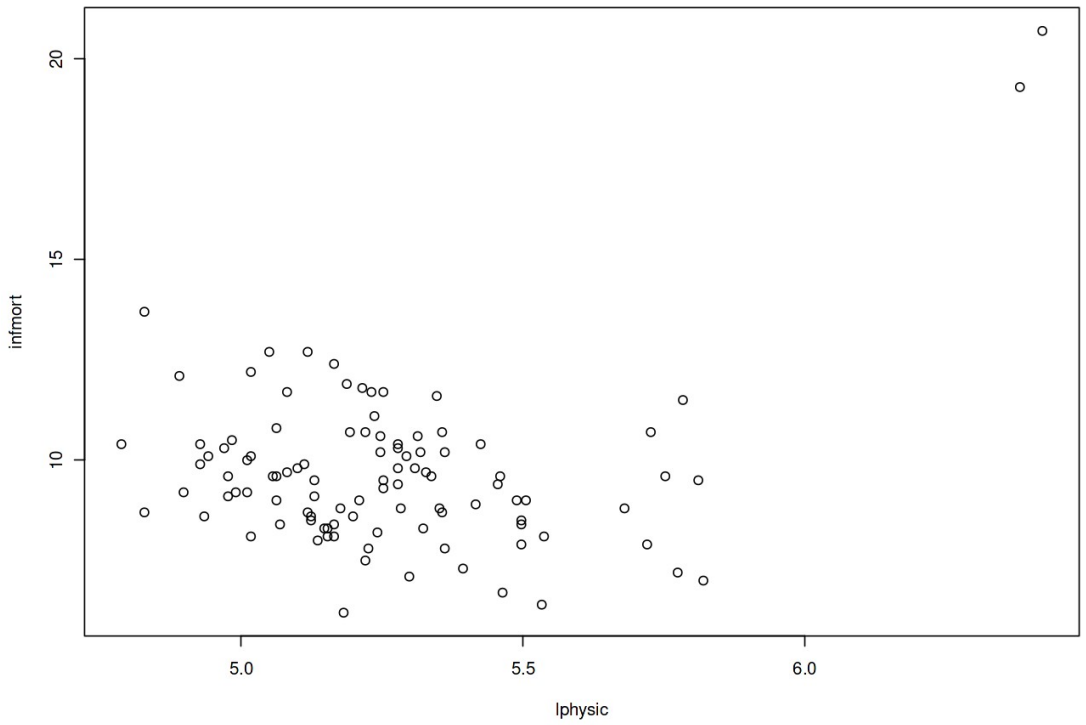
```
infmrt <- as.data.table(infmrt)
infmrt %>%
    .[lphysic <= 6.0] %>%
    .[, .(infmort, lpcinc, lphysic, lpopul)] -> infmrt2
```

```
head(infmrt2)
```

```
  infmort lpcinc   lphysic  lpopul
1 8.3     9.546527 5.153292 7.078341
2 6.2     9.748295 5.181784 7.113142
```

```
3 7.8     9.802728 5.225747 6.962244
4 7.1     9.954703 5.298317 7.011214
5 8.5     9.565704 5.497168 6.304449
6 6.4     9.777357 5.533390 6.333280
```

```
infmrt %>%
    with(plot(lphysic, infmort))
```

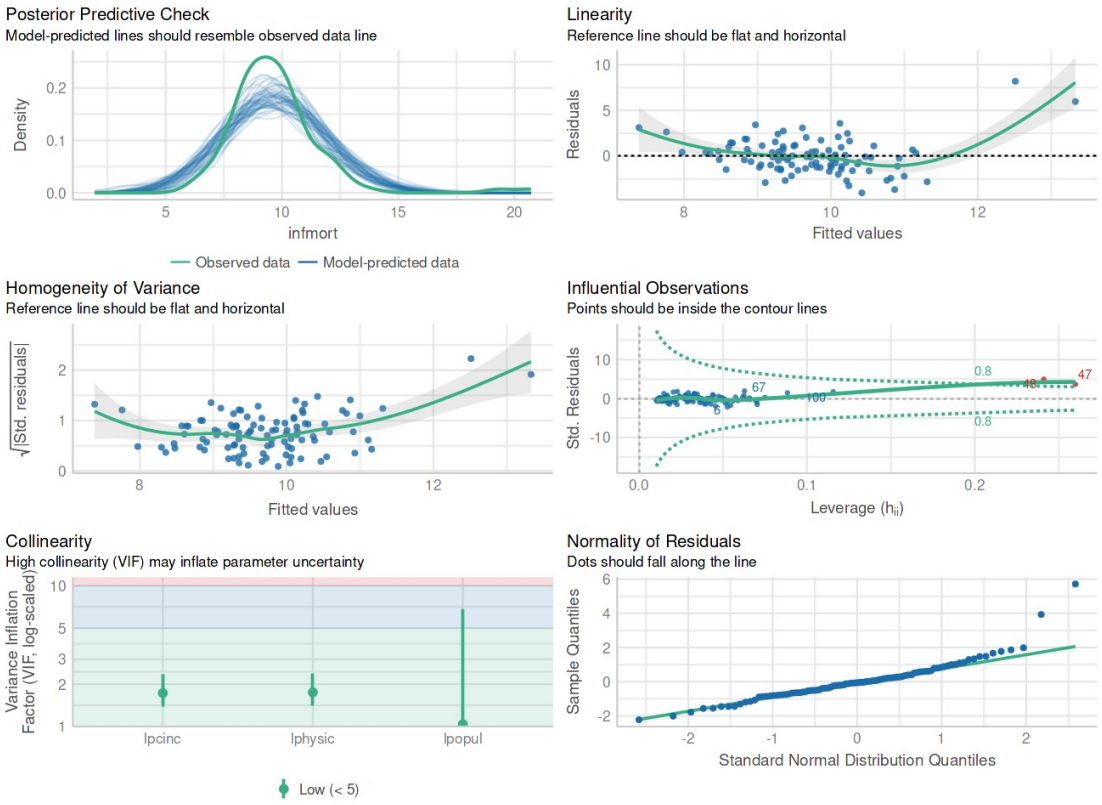



```
infmrt_reg1 <- lm(infmort ~ lpcinc + lphysic + lpopul, data = infmrt)
stargazer(infmrt_reg1, type="text")
```

```
================================================
                        Dependent variable:
                    ---------------------------
                              infmort
------------------------------------------------
lpcinc                       -4.884***
                              (1.293)

lphysic                      4.028***
                              (0.891)

lpopul                        -0.054
                              (0.187)
```

```
Constant                        36.226***
                                (10.135)

-----------------------------------------------
Observations                      102
R2                               0.182
Adjusted R2                      0.157
Residual Std. Error         1.891 (df = 98)
F Statistic             7.260*** (df = 3; 98)
===============================================
Note:                 *p<0.1; **p<0.05; ***p<0.01
```

```
check_model(infmrt_reg1)
```

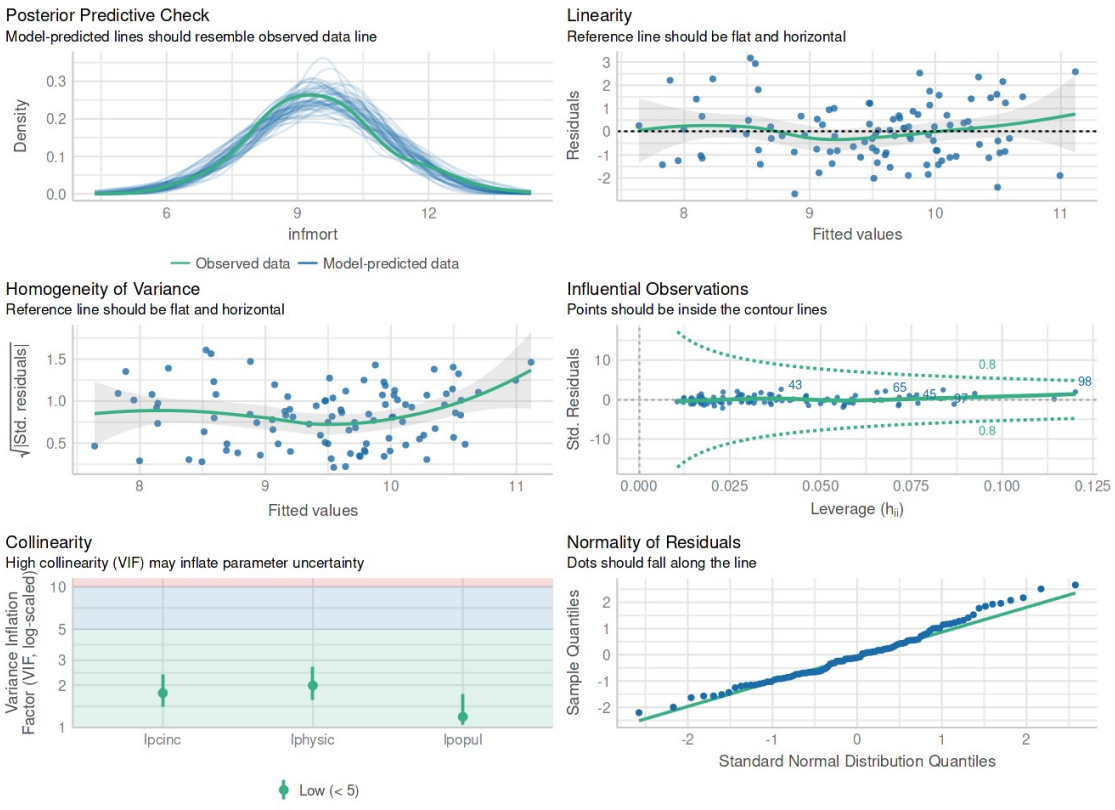



```
infmrt_reg2 <- lm(infmort ~ lpcinc + lphysic + lpopul, data = infmrt2)
stargazer(infmrt_reg2, type="text")
```

```
===============================================
                        Dependent variable:
                    ---------------------------
                              infmort
-----------------------------------------------
lpcinc                        -2.484***
                              (0.889)
```

```
lphysic                          -1.516*
                                 (0.773)

lpopul                           0.576***
                                 (0.136)

Constant                         36.854***
                                 (6.744)

-----------------------------------------------
Observations                       100
R2                                0.286
Adjusted R2                       0.264
Residual Std. Error         1.254 (df = 96)
F Statistic            12.832*** (df = 3; 96)
===============================================
Note:                  *p<0.1; **p<0.05; ***p<0.01
```

```
check_model(infmrt_reg2)
```



Correlation does not imply causation

It is important to note that correlation does not imply causation. Even the coefficient is significant, we cannot say that the independent variable causes the dependent variable. For instance, here the coefficient of `markets` is significant. However, we cannot say that firms

become more innovative because they spend more on marketing. It could be the case that firms become more innovative, so they spend more on marketing.

Our regression model only tells us that there is a relationship between the dependent variable and the independent variable. The relationship is significant enough to reject the null hypothesis that the coefficient is zero. However, it does not tell us the direction of the relationship. It does not tell us whether the independent variable causes the dependent variable or the dependent variable causes the independent variable.