VIETNAM NATIONAL UNIVERSITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
OFFICE FOR INTERNATIONAL STUDY PROGRAMS

PROBABILITY AND STATISTICS (MT2013)

PROJECT REPORT
Class: CC11 —— Group: 2

Under the guidance of: DR. PHAN THI HUONG
Accomplished by: LE GIA HUY – 1952717
PHAM THIEN DANG – 1952653
HOANG THE SON – 2053399
NGUYEN NGOC HUNG – 2053075
TRAN QUANG THIEN – 2053455

Ho Chi Minh City, March 2022

Contents

PROLOUGE

It is the moment for the project. This time, the project is mainly dealt with *multiple linear regression* problems as well as a number of *descriptive statistics* techniques. As we were stated in the previous report, all the outputs of R's computation, rather than captured in the RStudio environment, are showed directly from the command line console; which somewhat eases up our inspection thanks to high contrast and standout texts. Moreover, instead showing the whole R codes at the end of each question, this time the code snippets will be show along with the explanation texts during the demonstration. The structure of the report will also be more specific with a bunch of subsections for each activity. You will find the question, the procedure was carried out to attain the conclusion, and a brief summary for each problem along the way. Again, the assignment table is located at the last section of the document, where you will find the detailed descriptions of the tasks of each member in this project and their according percentage workload.

Now that it is enough for setting up the context, more will be explained when you walk through the document. To get an accomplished report, the team would like to give our instructor (Dr. Phan Thi Huong) a big appreciation for her great effort in helping in all the concepts of this course.

Member list & Workload

No.	Fullname	Student ID	Problems	Work Percentage
1	Le Gia Huy	1952717	- Accomplished Activity 1	100%
2	Pham Thien Dang	1952653	- Accomplished the Latex report	100%
3	Hoang The Son	2053399	- Accomplished Activity 2	100%
4	Nguyen Ngoc Hung	2053075	- Accomplished Activity 2	100%
5	Tran Quang Thien	2053455	- Accomplished Activity 2	100%

1 Activity 1

1.1 Problem

This data approach student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school related features and it was collected by using school reports and questionnaires.

Attribute Information:

- *sex* - student's sex (binary: F - **female** or M - **male**)
- *age* - student's age (numeric: from **15** to **22**)
- *studytime* - weekly study time (1: $<$ 2 hours, 2: 2 to 5 hours, 3: 5 to 10 hours, or 4: $>$ 10 hours)
- *failures* - number of past class failures (numeric: n if $1 \leq n < 3$, else 4).
- *higher* - wants to take higher education (binary: **yes** or **no**)
- *absences* - number of school absences (numeric: from **0** to **93**)
- These grades are related with the course subject, Math or Portuguese:
  - *G1* - first period grade (numeric: from **0** to **20**)
  - *G2* - second period grade (numeric: from **0** to **20**)
  - *G3* - final grade (numeric: from **0** to **20**, output target)

Steps:

1. Import data: **grade.csv**
2. Data cleaning: **NA** (Not available)
3. Data visualization
   (a) Transformation (if it is necessary)
   (b) Descriptive statistics for each of the variables
   (c) Graphs: hist, boxplot, pairs
4. Fitting linear regression models: We want to explore what factors may affect the final grade.
5. Predictions.

1.2 Solution

1.2.1 Import data

At first, installing the libraries for commands and functions is needed to solve the problem in a clear way.

1. Installing the packages:

```
install.packages("dplyr")
install.packages("GGally")
install.packages("broom")
install.packages("ggpubr")
```

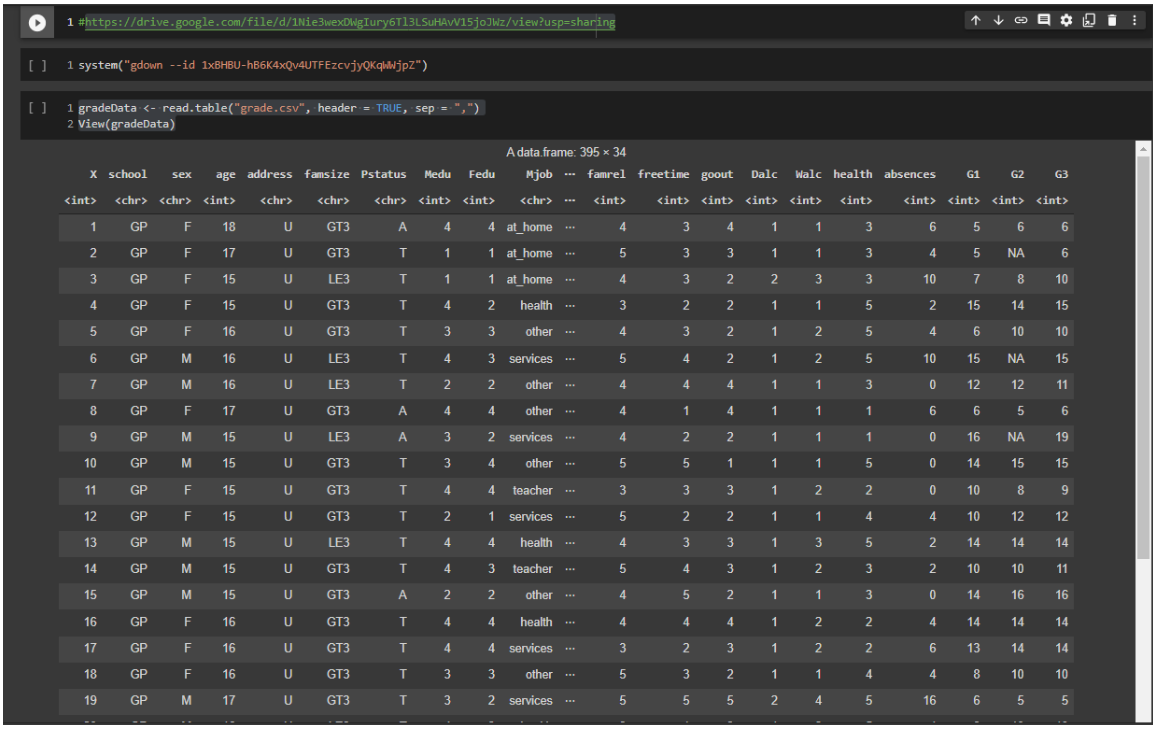
2. Calling the libraries:

```r
library(ggplot2)
library(devtools)
library(GGally)
library(dplyr)
library(broom)
```

After building a group of libraries, inputting the dataset and organizing the variables or factors from the dataset in columns are the following steps.

```r
#https://drive.google.com/file/d/1Nie3wexDWgIury6Tl3LSuHAvV15joJWz/view?usp=sharing
system("gdown --id 1xBHBU-hB6K4xQv4UTFEzcvjyQKqWWjpZ")
gradeData <- read.table("grade.csv", header = TRUE, sep = ",")
View(gradeData)
```

And here for the result via using the *dim(gradeData)* command:



```r
#https://drive.google.com/file/d/1Nie3wexDWgIury6Tl3LSuHAvV15joJWz/view?usp=sharing
```

```r
system("gdown --id 1xBHBU-hB6K4xQv4UTFEzcvjyQKqWWjpZ")
```

```r
gradeData <- read.table("grade.csv", header = TRUE, sep = ",")
View(gradeData)
```

A data.frame: 395 × 34

X	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	⋯	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
<int>	<chr>	<chr>	<int>	<chr>	<chr>	<chr>	<int>	<int>	<chr>	⋯	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
1	GP	F	18	U	GT3	A	4	4	at_home	⋯	4	3	4	1	1	3	6	5	6	6
2	GP	F	17	U	GT3	T	1	1	at_home	⋯	5	3	3	1	1	3	4	5	NA	6
3	GP	F	15	U	LE3	T	1	1	at_home	⋯	4	3	2	2	3	3	10	7	8	10
4	GP	F	15	U	GT3	T	4	2	health	⋯	3	2	2	1	1	5	2	15	14	15
5	GP	F	16	U	GT3	T	3	3	other	⋯	4	3	2	1	2	5	4	6	10	10
6	GP	M	16	U	LE3	T	4	3	services	⋯	5	4	2	1	2	5	10	15	NA	15
7	GP	M	16	U	LE3	T	2	2	other	⋯	4	4	4	1	1	3	0	12	12	11
8	GP	F	17	U	GT3	A	4	4	other	⋯	4	1	4	1	1	1	6	6	5	6
9	GP	M	15	U	LE3	A	3	2	services	⋯	4	2	2	1	1	1	0	16	NA	19
10	GP	M	15	U	GT3	T	3	4	other	⋯	5	5	1	1	1	5	0	14	15	15
11	GP	F	15	U	GT3	T	4	4	teacher	⋯	3	3	3	1	2	2	0	10	8	9
12	GP	F	15	U	GT3	T	2	1	services	⋯	5	2	2	1	1	4	4	10	12	12
13	GP	M	15	U	LE3	T	4	4	health	⋯	4	3	3	1	3	5	2	14	14	14
14	GP	M	15	U	GT3	T	4	3	teacher	⋯	5	4	3	1	2	3	2	10	10	11
15	GP	M	15	U	GT3	A	2	2	other	⋯	4	5	2	1	1	3	0	14	16	16
16	GP	F	16	U	GT3	T	4	4	health	⋯	4	4	4	1	2	2	4	14	14	14
17	GP	F	16	U	GT3	T	4	4	services	⋯	3	2	3	1	2	2	6	13	14	14
18	GP	F	16	U	GT3	T	3	3	other	⋯	5	3	2	1	1	4	4	8	10	10
19	GP	M	17	U	GT3	T	3	2	services	⋯	5	5	5	2	4	5	16	6	5	5

Figure 1: *There are 395 students whose information collected and 34 attributes corresponding to each student*

1.2.2 Data cleaning: NA

Locating the null value in any factors and replacing them is the significant stage in data cleaning. In order to complete this step, by using the *summary(gradeData)* command.

```
      famrel          freetime         goout            Dalc      
 Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000  
 1st Qu.:4.000   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:1.000  
 Median :4.000   Median :3.000   Median :3.000   Median :1.000  
 Mean   :3.944   Mean   :3.235   Mean   :3.109   Mean   :1.481  
 3rd Qu.:5.000   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:2.000  
 Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000  

      Walc           health         absences           G1       
 Min.   :1.000   Min.   :1.000   Min.   : 0.000   Min.   : 3.00  
 1st Qu.:1.000   1st Qu.:3.000   1st Qu.: 0.000   1st Qu.: 8.00  
 Median :2.000   Median :4.000   Median : 4.000   Median :11.00  
 Mean   :2.291   Mean   :3.554   Mean   : 5.709   Mean   :10.91  
 3rd Qu.:3.000   3rd Qu.:5.000   3rd Qu.: 8.000   3rd Qu.:13.00  
 Max.   :5.000   Max.   :5.000   Max.   :75.000   Max.   :19.00  

       G2              G3       
 Min.   : 0.00   Min.   : 0.00  
 1st Qu.: 9.00   1st Qu.: 8.00  
 Median :11.00   Median :11.00  
 Mean   :10.72   Mean   :10.42  
 3rd Qu.:13.00   3rd Qu.:14.00  
 Max.   :19.00   Max.   :20.00  
 NA's   :5                      
```

Figure 2: *There are 5 NA values in G2 column*

So the next step is the change in those values into the median calculated by rest values in this column.

```
gradeData[is.na(gradeData)] = median(gradeData$G2, na.rm = TRUE)
head(gradeData)
```

A data.frame: 6 × 34

	X	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
	<int>	<chr>	<chr>	<int>	<chr>	<chr>	<chr>	<int>	<int>	<chr>	...	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<dbl>	<int>
1	1	GP	F	18	U	GT3	A	4	4	at_home	...	4	3	4	1	1	3	6	5	6	6
2	2	GP	F	17	U	GT3	T	1	1	at_home	...	5	3	3	1	1	3	4	5	11	6
3	3	GP	F	15	U	LE3	T	1	1	at_home	...	4	3	2	2	3	3	10	7	8	10
4	4	GP	F	15	U	GT3	T	4	2	health	...	3	2	2	1	1	5	2	15	14	15
5	5	GP	F	16	U	GT3	T	3	3	other	...	4	3	2	1	2	5	4	6	10	10
6	6	GP	M	16	U	LE3	T	4	3	services	...	5	4	2	1	2	5	10	15	11	15

Figure 3: *There are 5 NA values in G2 column*

1.2.3 Data visualization

1.2.3.1. Transformation

To utilize R program to calculate, all factors or values from the dataset must be transferred to numeric type. Before the transformation process is coded, several implies are established for thorough understanding.

- School: GP = 0 | Sex: Female = 1
- School: MS = 1 | Sex: Male = 0

- Address: U = 0 | Famsize: GT3 = 0
- Address: R = 1 | Famsize: LE3 = 1

- Pstatus: A = 0
- Pstatus: T = 1

- Jobs: at_home = 0 | Reason: course = 0
- Jobs: services = 1 | Reason: home = 1

- Jobs: teacher = 2
- Jobs: health = 3
- Jobs: other = 4

Reason: reputation = 2

Reason: other = 3

- Guardian: father = 0
- Guardian: mother = 1
- Guardian: other = 3

Everything else: no = 0

Everything else: yes = 1

And then, converting these values to numerical values.

```
gradeData[gradeData == "GP"] <- 0
gradeData[gradeData == "MS"] <- 1

gradeData[gradeData == "M"] <- 0
gradeData[gradeData == "F"] <- 1

gradeData[gradeData == "U"] <- 0
gradeData[gradeData == "R"] <- 1

gradeData[gradeData == "GT3"] <- 0
gradeData[gradeData == "LE3"] <- 1

gradeData[gradeData == "A"] <- 0
gradeData[gradeData == "T"] <- 1

gradeData[gradeData == "at_home"] <- 0
gradeData[gradeData == "services"] <- 1
gradeData[gradeData == "teacher"] <- 2
gradeData[gradeData == "health"] <- 3
gradeData$Mjob[gradeData$Mjob == "other"] <- 4
gradeData$Fjob[gradeData$Fjob == "other"] <- 4

gradeData[gradeData == "course"] <- 0
gradeData[gradeData == "home"] <- 1
gradeData[gradeData == "reputation"] <- 2
gradeData$reason[gradeData$reason == "other"] <- 3

gradeData[gradeData == "father"] <- 0
gradeData[gradeData == "mother"] <- 1
gradeData$guardian[gradeData$guardian == "other"] <- 3

gradeData[gradeData == "yes"] <- 0
gradeData[gradeData == "no"] <- 1

head(gradeData)
```

Figure 4: *Converting to numerical values*

Now, our dataframe is now ready for analysing.

A data.frame: 6 × 34

	X	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	⋯	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
	<int>	<chr>	<chr>	<int>	<chr>	<chr>	<chr>	<int>	<int>	<chr>	⋯	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<dbl>	<int>
1	1	0	1	18	0	0	0	4	4	0	⋯	4	3	4	1	1	3	6	5	6	6
2	2	0	1	17	0	0	1	1	1	0	⋯	5	3	3	1	1	3	4	5	11	6
3	3	0	1	15	0	1	1	1	1	0	⋯	4	3	2	2	3	3	10	7	8	10
4	4	0	1	15	0	0	1	4	2	3	⋯	3	2	2	1	1	5	2	15	14	15
5	5	0	1	16	0	0	1	3	3	4	⋯	4	3	2	1	2	5	4	6	10	10
6	6	0	0	16	0	1	1	4	3	1	⋯	5	4	2	1	2	5	10	15	11	15

Figure 5: *Analysing table.*

1.2.3.2. Statistics for each of the variables

After the data cleaning and transformation have been done, class(gradedata and summary command is used to form all the variables into the separate table containing calculating information such as min, 1^{st} Qu., median, mean, 3^{rd} Qu., and max.

```
[ ]
class(gradeData$school) <- "numeric"
class(gradeData$sex) <- "numeric"
class(gradeData$address) <- "numeric"
class(gradeData$famsize) <- "numeric"
class(gradeData$Pstatus) <- "numeric"
class(gradeData$Mjob) <- "numeric"
class(gradeData$Fjob) <- "numeric"
class(gradeData$reason) <- "numeric"
class(gradeData$guardian) <- "numeric"
class(gradeData$schoolsup) <- "numeric"
class(gradeData$famsup) <- "numeric"
class(gradeData$paid) <- "numeric"
class(gradeData$activities) <- "numeric"
class(gradeData$nursery) <- "numeric"
class(gradeData$higher) <- "numeric"
class(gradeData$internet) <- "numeric"
class(gradeData$romantic) <- "numeric"

summary(gradeData)
```

Figure 6: *Example for code.*

For example, as can be seen from the Fig. 7, the description of final score G3:

- The lowest score is 0.00 (Min = 0.00), the highest score is 20.00 (Max = 20.00). The range of G3 will be 20.00 - 0.00 = 20.
- 1^{st} Qu. is 8.00 shows that 25% of students have their final score less than or equal to 8.00.
- Median = 11.00 shows that 50% of students have their final score less than or equal to 11.00.
- 3^{rd} Qu. is 14.00 shows that 75% of students have their final score less than or equal to 14.00.
- Mean = 10.42 shows that the average score of all 395 students is 10.42.

For the dummy variable sex which takes only 2 values 0 or 1, its description shows that:

- Median = 1.0000, meaning that more than 50% of values are 1.
- Mean = 0.5266, meaning that 52.66% of students are female, 47.34% of students are male.

Here is the description the statistics of each variable:

```
       X              school            sex              age       
 Min.   :  1.0   Min.   :0.0000   Min.   :0.0000   Min.   :15.0  
 1st Qu.: 99.5   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:16.0  
 Median :198.0   Median :0.0000   Median :1.0000   Median :17.0  
 Mean   :198.0   Mean   :0.1165   Mean   :0.5266   Mean   :16.7  
 3rd Qu.:296.5   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:18.0  
 Max.   :395.0   Max.   :1.0000   Max.   :1.0000   Max.   :22.0  
    address          famsize          Pstatus            Medu      
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000  
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.:2.000  
 Median :0.0000   Median :0.0000   Median :1.0000   Median :3.000  
 Mean   :0.2228   Mean   :0.2886   Mean   :0.8962   Mean   :2.749  
 3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:4.000  
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :4.000  
      Fedu            Mjob            Fjob           reason     
 Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
 1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.000  
 Median :2.000   Median :2.000   Median :4.000   Median :1.000  
 Mean   :2.522   Mean   :2.241   Mean   :2.762   Mean   :1.081  
 3rd Qu.:3.000   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:2.000  
 Max.   :4.000   Max.   :4.000   Max.   :4.000   Max.   :3.000  
    guardian        traveltime      studytime        failures     
 Min.   :0.0000   Min.   :1.000   Min.   :1.000   Min.   :0.0000  
 1st Qu.:1.0000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000  
 Median :1.0000   Median :1.000   Median :2.000   Median :0.0000  
 Mean   :0.9342   Mean   :1.448   Mean   :2.035   Mean   :0.3342  
 3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:0.0000  
 Max.   :3.0000   Max.   :4.000   Max.   :4.000   Max.   :3.0000  
   schoolsup          famsup            paid          activities    
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000  
 Median :1.0000   Median :0.0000   Median :1.0000   Median :0.0000  
 Mean   :0.8709   Mean   :0.3873   Mean   :0.5418   Mean   :0.4911  
 3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000  
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000  
    nursery           higher           internet          romantic     
 Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000  
 Median :0.0000   Median :0.00000   Median :0.0000   Median :1.0000  
 Mean   :0.2051   Mean   :0.05063   Mean   :0.1671   Mean   :0.6658  
 3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:1.0000  
 Max.   :1.0000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000  
     famrel         freetime         goout            Dalc      
 Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000  
 1st Qu.:4.000   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:1.000  
 Median :4.000   Median :3.000   Median :3.000   Median :1.000  
 Mean   :3.944   Mean   :3.235   Mean   :3.109   Mean   :1.481  
 3rd Qu.:5.000   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:2.000  
 Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000  
      Walc           health         absences            G1       
 Min.   :1.000   Min.   :1.000   Min.   : 0.000   Min.   : 3.00  
 1st Qu.:1.000   1st Qu.:3.000   1st Qu.: 0.000   1st Qu.: 8.00  
 Median :2.000   Median :4.000   Median : 4.000   Median :11.00  
 Mean   :2.291   Mean   :3.554   Mean   : 5.709   Mean   :10.91  
 3rd Qu.:3.000   3rd Qu.:5.000   3rd Qu.: 8.000   3rd Qu.:13.00  
 Max.   :5.000   Max.   :5.000   Max.   :75.000   Max.   :19.00  
       G2              G3       
 Min.   : 0.00   Min.   : 0.00  
 1st Qu.: 9.00   1st Qu.: 8.00  
 Median :11.00   Median :11.00  
 Mean   :10.72   Mean   :10.42  
 3rd Qu.:13.00   3rd Qu.:14.00  
 Max.   :19.00   Max.   :20.00  
```

Figure 7: *The min, max, 1^{st} quartile, median, 3^{rd} quartile and the mean value of all variables are described in the result above.*

1.2.3.3. Graphs: hist, boxplot, pair

1.2.3.3.a. Histogram

A histogram is a bar graph-like representation of data that buckets a range of outcomes into columns along the x-axis. The y-axis represents the number count or percentage of frequencies in the data for each column and can be used to visualize data distributions.

In R , we will call *hist()* function to represent the histogram.

```
options(repr.plot.width=30, repr.plot.height=15)
par(mfrow=c(4,4))
hist(gradeData$G1, main = "G1", col = "green")
hist(gradeData$G2, main = "G2", col = "yellow")
hist(gradeData$G3, main = "G3", col = "orange")
hist(gradeData$age, main = "age")
hist(gradeData$absences, main = "absences")
hist(gradeData$studytime, main = "studytime")
hist(gradeData$health, main = "health")
hist(gradeData$goout, main = "goout")
hist(gradeData$freetime, main = "freetime")
hist(gradeData$Medu, main = "Medu")
hist(gradeData$Fedu, main = "Fedu")
hist(gradeData$famrel,  main = "famrel")
hist(gradeData$Dalc, main = "Dalc")
hist(gradeData$Walc, main = "Walc")
hist(gradeData$traveltime, main = "traveltime")
hist(gradeData$failures, main = "failures")
```

Figure 8: *The lines of code for creating histogram of each variable.*

As the result, we are able to obtain the histogram of each variable.

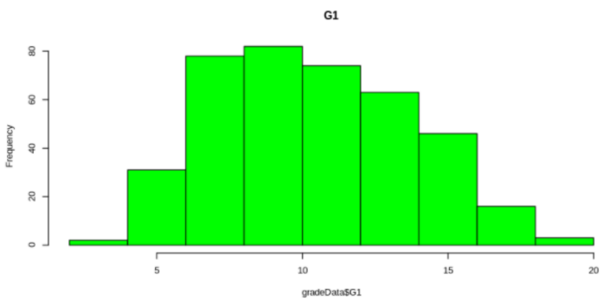


Figure 9: *Histogram for G1.*

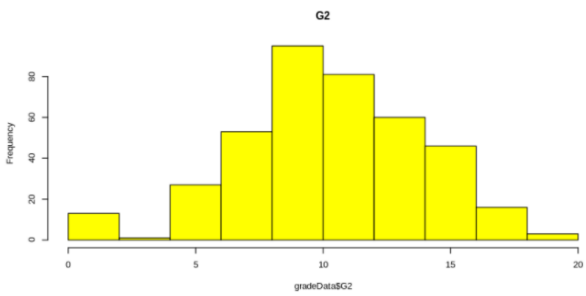


Figure 10: *Histogram for G2.*

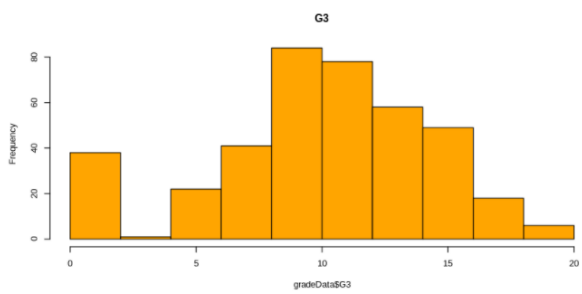


Figure 11: *Histogram for G3.*

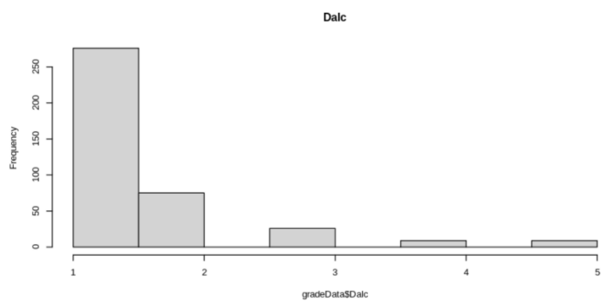


Figure 12: *Histogram for Dalc.*

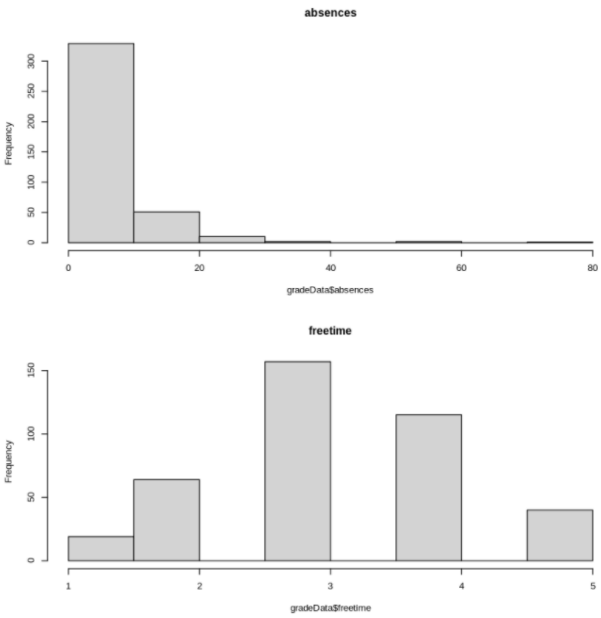


Figure 13: *Histogram for absences and freetime.*

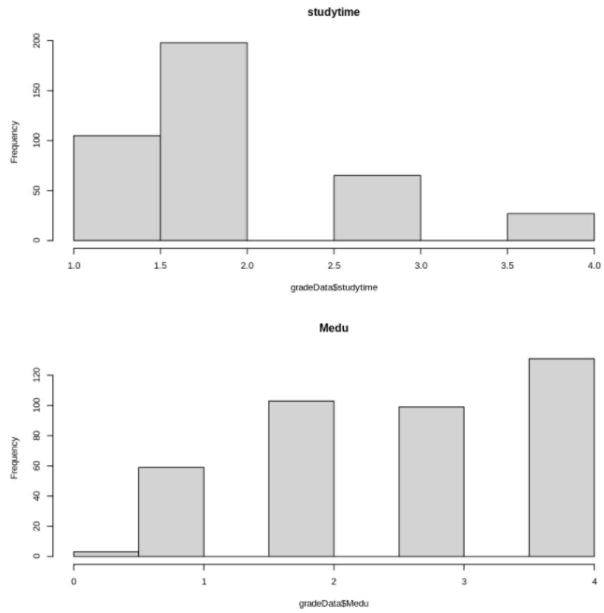


Figure 14: *Histogram for studytime and Medu.*

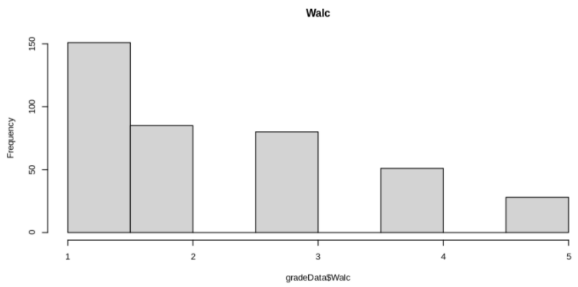


Figure 15: *Histogram for Walc.*

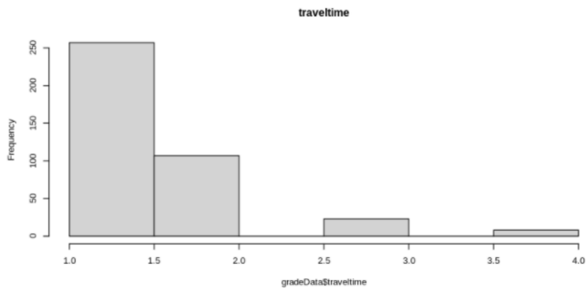


Figure 16: *Histogram for traveltime.*

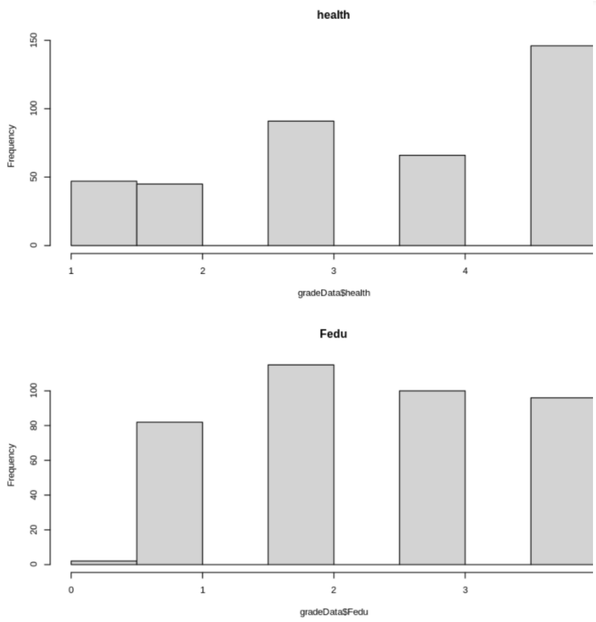


Figure 17: *Histogram for health and Fedu.*

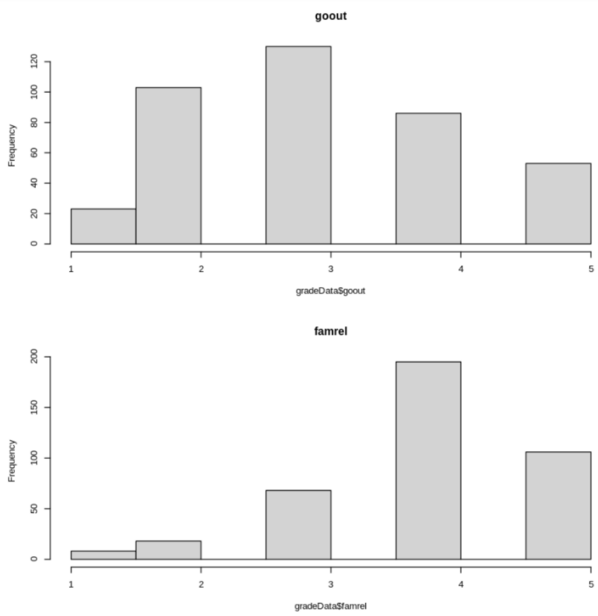


Figure 18: *Histogram for go out and famrel.*

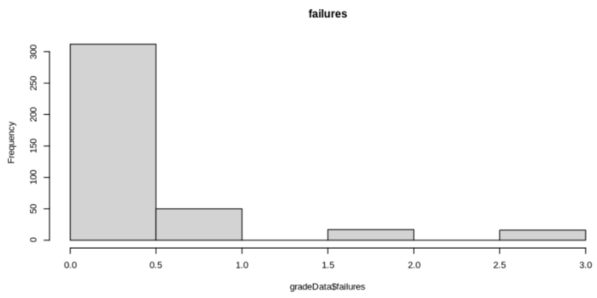


Figure 19: *Histogram for failures.*

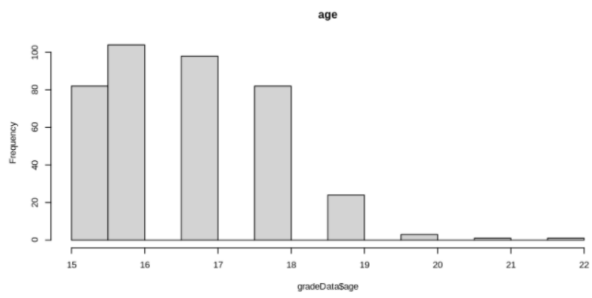


Figure 20: *Histogram for age.*

1.2.3.3.b. Boxplot

Boxplot is a graphical representation of statistical measures like median, upper and lower quartiles, minimum and maximum data values. Thus, we will make 2 situations for comparision among G3 and the others.

In R, we use function *boxplot()* to represent boxplot.

1. Comparing final grade G3 with G1, G2, Medu, Fedu, age, absences, studytime, health and go out.

```
options(repr.plot.width=30, repr.plot.height=15)
par(mfrow=c(3,3))
boxplot(gradeData$G3 ~ gradeData$school, horizontal = TRUE, main = "school-G3")
boxplot(gradeData$G3 ~ gradeData$address, horizontal = TRUE, main = "address-G3")
boxplot(gradeData$G3 ~ gradeData$sex, horizontal = TRUE, main = "sex-G3")
boxplot(gradeData$G3 ~ gradeData$higher, horizontal = TRUE, main = "higher-G3")
boxplot(gradeData$G3 ~ gradeData$failures, horizontal = TRUE, main = "failures-G3")
boxplot(gradeData$G3 ~ gradeData$famrel, horizontal = TRUE, main = "famrel-G3")
boxplot(gradeData$G3 ~ gradeData$reason, horizontal = TRUE, main = "reason-G3")
boxplot(gradeData$G3 ~ gradeData$romantic, horizontal = TRUE, main = "romantic-G3")
boxplot(gradeData$G3 ~ gradeData$nursery, horizontal = TRUE, main = "nursery-G3")
```

Figure 21: *The above codes are used to represent boxplot for case 1.*

As the result, we are able to obtain the boxplot of each variable in case 1.

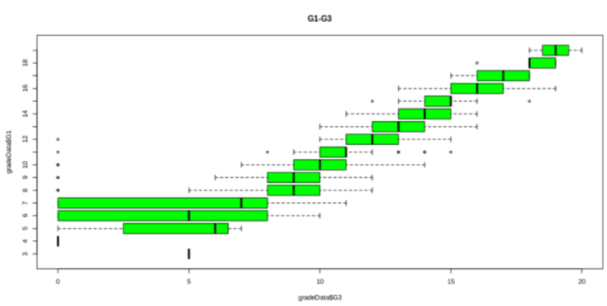


Figure 22: *Boxplot for G1 vs G3.*

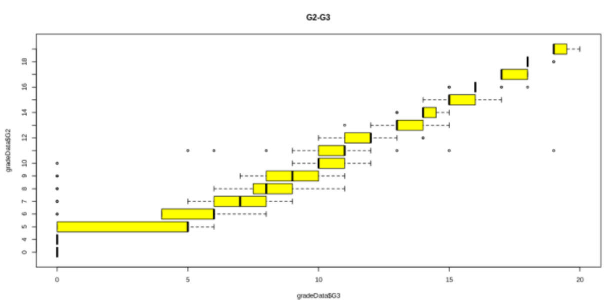


Figure 23: *Boxplot for G2 vs G3.*

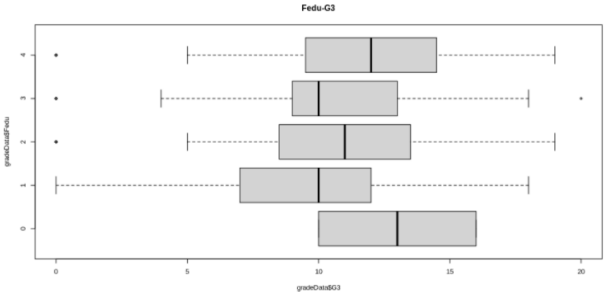


Figure 24: *Boxplot for Fedu vs G3.*

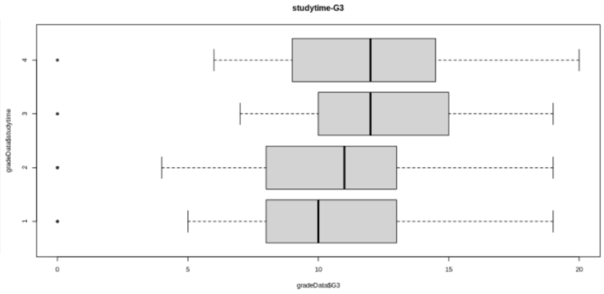


Figure 25: *Boxplot for studytime vs G3.*

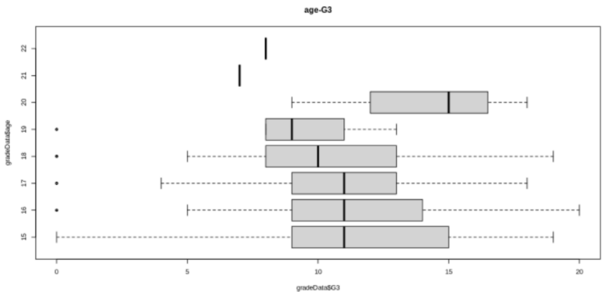


Figure 26: *Boxplot for age vs G3.*

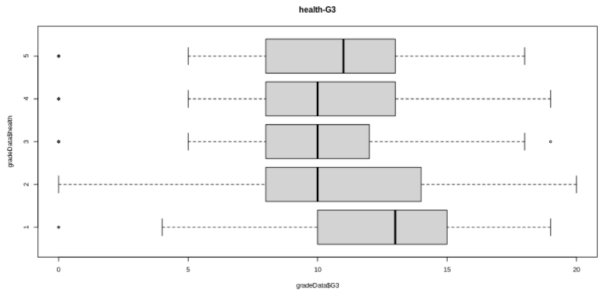


Figure 27: *Boxplot for health vs G3.*

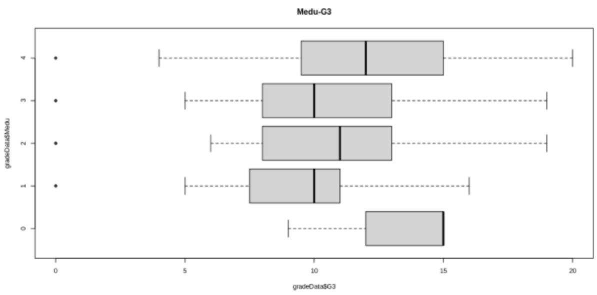


Figure 28: *Boxplot for Medu vs G3.*

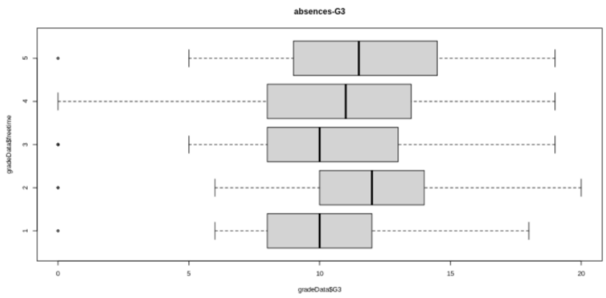


Figure 29: *Boxplot for absences vs G3.*

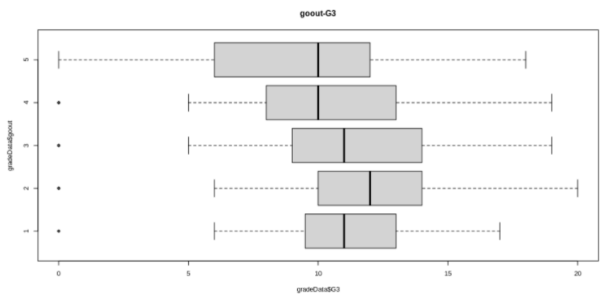


Figure 30: *Boxplot for go out vs G3.*

2. Comparing final grade G3 with school, address, sex, higher, failures, famrel, reason, romantic and nursery.

```r
options(repr.plot.width=30, repr.plot.height=15)
par(mfrow=c(3,3))
boxplot(gradeData$G3 ~ gradeData$school, horizontal = TRUE, main = "school-G3")
boxplot(gradeData$G3 ~ gradeData$address, horizontal = TRUE, main = "address-G3")
boxplot(gradeData$G3 ~ gradeData$sex, horizontal = TRUE, main = "sex-G3")
boxplot(gradeData$G3 ~ gradeData$higher, horizontal = TRUE, main = "higher-G3")
boxplot(gradeData$G3 ~ gradeData$failures, horizontal = TRUE, main = "failures-G3")
boxplot(gradeData$G3 ~ gradeData$famrel, horizontal = TRUE, main = "famrel-G3")
boxplot(gradeData$G3 ~ gradeData$reason, horizontal = TRUE, main = "reason-G3")
boxplot(gradeData$G3 ~ gradeData$romantic, horizontal = TRUE, main = "romantic-G3")
boxplot(gradeData$G3 ~ gradeData$nursery, horizontal = TRUE, main = "nursery-G3")
```

Figure 31: *The above codes are used to represent boxplot for case 2.*

As the result, we are able to obtain the boxplot of each variable in case 2.

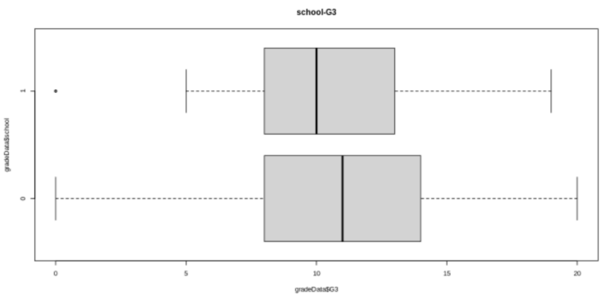


Figure 32: *A Boxplot for school vs G3.*

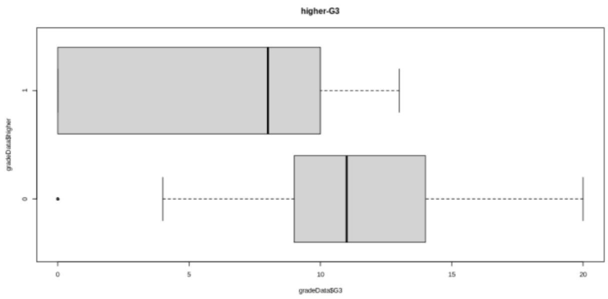


Figure 33: *Boxplot for higher vs G3.*

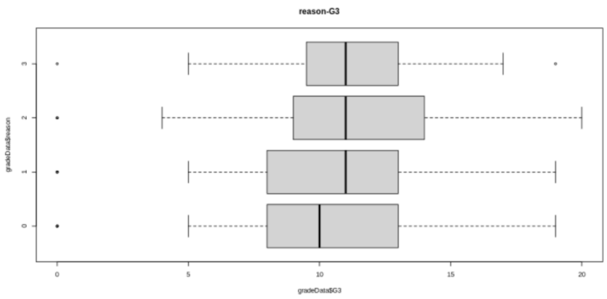


Figure 34: *Boxplot for reason vs G3.*

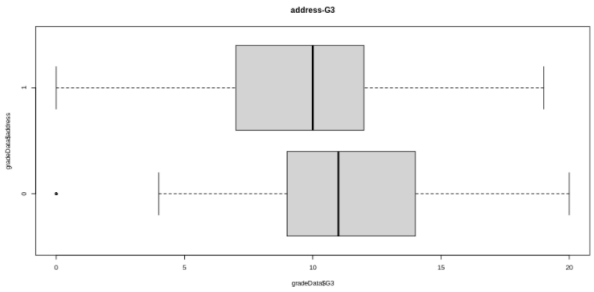


Figure 35: *Boxplot for address vs G3.*

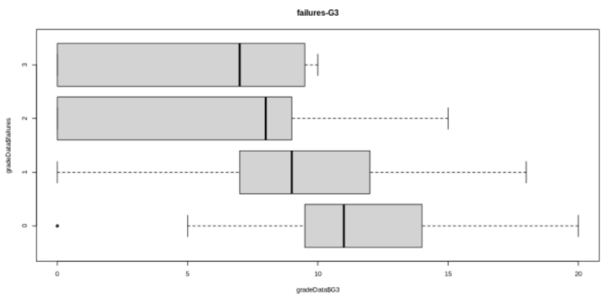


Figure 36: *Boxplot for failures vs G3.*

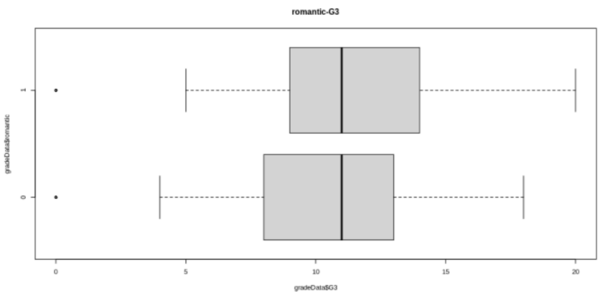


Figure 37: *Boxplot for romantic vs G3.*

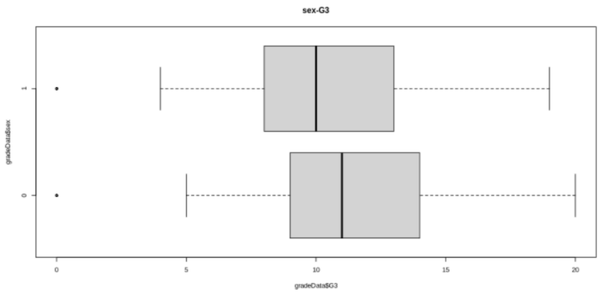


Figure 38: *Boxplot for sex vs G3.*

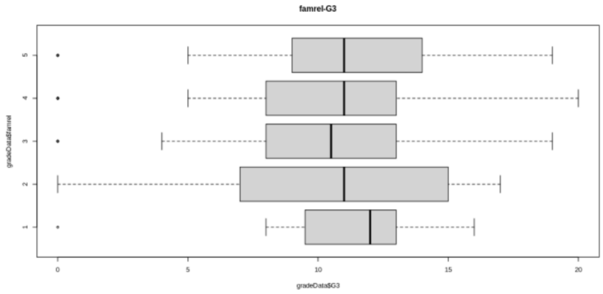


Figure 39: *Boxplot for famrel vs G3.*

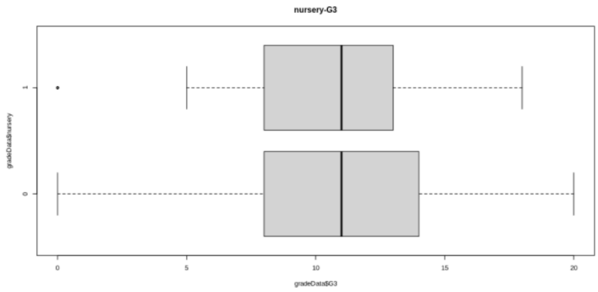


Figure 40: *Boxplot for nursery vs G3.*

1.2.3.3.c. Pairs

The *pairs* command in R function returns a plot matrix, consisting of scatterplots for each variable-combination of a data frame. In other words, using it to show the statistical relationship between variables (failures, age, higher, absences, famrel, Medu, Fedu, G1, G2 and G3).

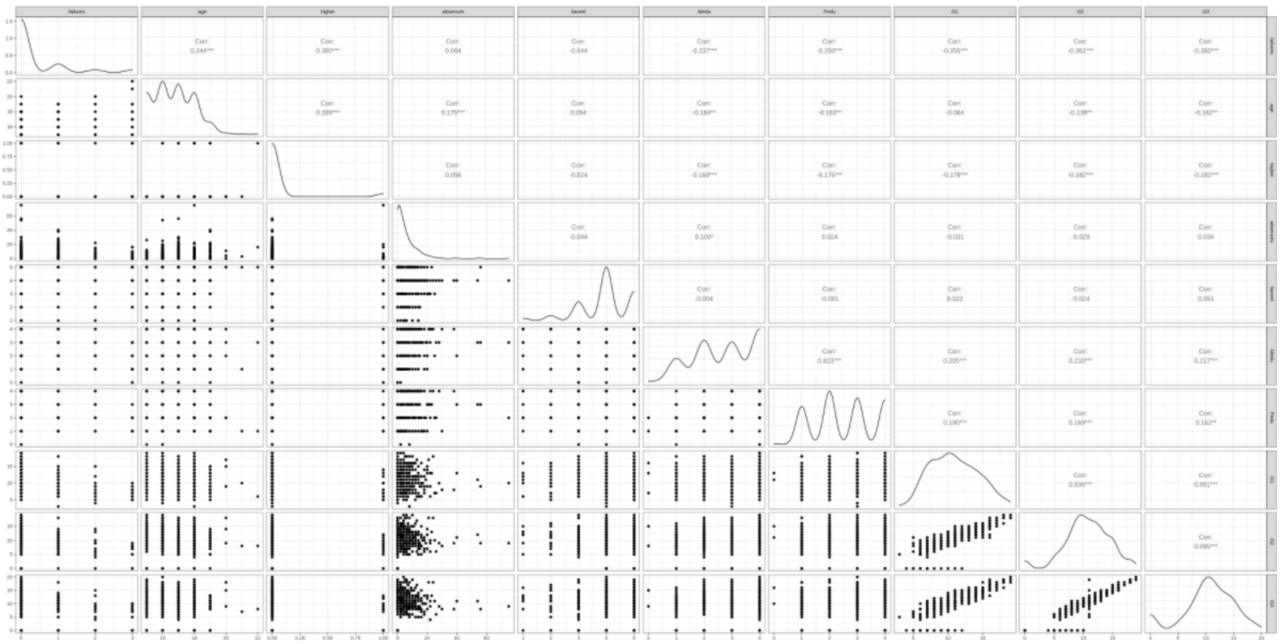


Figure 41: *Some linearity can be seen between pairs of variables, such as G1 and G3, or G2 and G3.*

```
options(repr.plot.width=30, repr.plot.height=15)
ggpairs(subData) + theme_bw()
```

Figure 42: *The basic R syntax for the pairs command.*

1.2.3.4. Fitting linear regression models

First, using below command to confirm that G3 is a function of the other values and *data* = *grade* confirm that R has to compute on dataset called grade.

```
LinearModel <- lm(G3 ~ .,data=gradeData)
summary(LinearModel)
```

Figure 43: *Example for code.*

Here for the result

```
Call:
lm(formula = G3 ~ ., data = gradeData)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.8255 -0.5936  0.2303  1.1035  5.6509 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -3.3579789  2.2313965  -1.505  0.13323    
X           -0.0040667  0.0016418  -2.477  0.01371 *  
school       0.9362638  0.3954825   2.367  0.01844 *  
sex         -0.2058639  0.2329911  -0.884  0.37752    
age          0.0176898  0.1391576   0.127  0.89892    
address      0.0384699  0.2681390   0.143  0.88600    
famsize      0.1233310  0.2251467   0.548  0.58418    
Pstatus     -0.3516909  0.3354882  -1.048  0.29520    
Medu         0.1339457  0.1232768   1.087  0.27796    
Fedu        -0.1784802  0.1214178  -1.470  0.14244    
Mjob         0.0066545  0.0682652   0.097  0.92240    
Fjob         0.0618953  0.0728158   0.850  0.39587    
reason       0.1127894  0.1021186   1.104  0.27011    
guardian     0.0064044  0.1515137   0.042  0.96631    
traveltime   0.0710490  0.1565070   0.454  0.65013    
studytime   -0.0983816  0.1328191  -0.741  0.45935    
failures    -0.2140613  0.1609057  -1.330  0.18424    
schoolsup   -0.4810206  0.3203189  -1.502  0.13405    
famsup      -0.1160770  0.2257009  -0.514  0.60736    
paid        -0.2506935  0.2219728  -1.129  0.25948    
activities   0.3210286  0.2065893   1.554  0.12107    
nursery      0.1883642  0.2542975   0.741  0.45934    
higher      -0.1833291  0.5014175  -0.366  0.71486    
internet     0.0873022  0.2860503   0.305  0.76039    
romantic     0.2312679  0.2205837   1.048  0.29514    
famrel       0.3476824  0.1140148   3.049  0.00246 ** 
freetime     0.0332276  0.1088535   0.305  0.76035    
goout       -0.0005492  0.1044636  -0.005  0.99581    
Dalc        -0.1999261  0.1515949  -1.319  0.18807    
Walc         0.1942372  0.1135938   1.710  0.08814 .  
health       0.0565784  0.0733699   0.771  0.44113    
absences     0.0406511  0.0133409   3.047  0.00248 ** 
G1           0.3077997  0.0582109   5.288 2.15e-07 ***
G2           0.8690375  0.0510109  17.036  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.922 on 361 degrees of freedom
Multiple R-squared:  0.8388,	Adjusted R-squared:  0.824 
F-statistic: 56.91 on 33 and 361 DF,  p-value: < 2.2e-16
```

Figure 44: *Result of the codes.*

Based on p-value, constructing 6 models more by eliminating one by one variable from the low p-value to the worst.

```
LinearModel_1 <- lm(G3 ~ X +school+ famrel + absences + G1 + G2 , data = gradeData)
LinearModel_2 <- lm(G3 ~ school + famrel + absences + G1 + G2, data= gradeData)
LinearModel_3 <- lm(G3 ~ famrel + absences + G1 + G2, data = gradeData)
LinearModel_4 <- lm(G3 ~ absences + G1 + G2, data = gradeData)
LinearModel_5 <- lm(G3 ~ G1 + G2, data = gradeData)
LinearModel_6 <- lm(G3 ~ G2, data = gradeData)
```

Figure 45: *Example for the codes.*

Then, by *anova* command, the comparison between regression models are built.

```
anova(LinearModel_6,LinearModel_5,LinearModel_4,LinearModel_3,LinearModel_2,LinearModel_1,LinearModel)
```

Figure 46: *Example for the codes.*

Now, the result will be taken.

A anova: 7 × 6

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	393	1642.932	NA	NA	NA	NA
2	392	1565.603	1	77.328443	20.9363570	6.538526e-06
3	391	1534.502	1	31.101716	8.4206613	3.937417e-03
4	390	1495.395	1	39.106886	10.5880280	1.245690e-03
5	389	1494.942	1	0.452328	0.1224659	7.265793e-01
6	388	1425.370	1	69.572059	18.8363481	1.851678e-05
7	361	1333.354	27	92.016649	0.9227085	5.792125e-01

Figure 47: *The results of the code.*

Observing the Anova data table from the model 1 to 7, the result has illustrated that the model 2 seems to be the finest model to be built a fitting linear regression model compared to other models because of the p-value (the model 2 has smallest value, p2 $\sim$ 0.019).

```
lm(formula = G3 ~ school + famrel + absences + G1 + G2, data = gradeData)
```

Figure 48: *Model 2.*

Then, having the fitting model below.

```
Residuals:
    Min      1Q  Median      3Q     Max
-9.3242 -0.4523  0.2072  1.0080  7.3526

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.77114    0.56316  -6.696 7.49e-11 ***
school       0.10628    0.30980   0.343  0.73173
famrel       0.35501    0.11080   3.204  0.00147 **
absences     0.03726    0.01241   3.002  0.00285 **
G1           0.23115    0.05443   4.247 2.72e-05 ***
G2           0.93638    0.04870  19.226  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.96 on 389 degrees of freedom
Multiple R-squared:  0.8192,    Adjusted R-squared:  0.8169
F-statistic: 352.6 on 5 and 389 DF,  p-value: < 2.2e-16
```

Figure 49: *The fitting model.*

As the result, we have the formula: **G3** = -3.77114 + 0.93638 × *G2* + 0.23115 × *G1* + 0.35501 × *famrel* + 0.03726 × *absences* + 0.10628 × *school1*.

Following that, plotting that model.

```
plot(LinearModel_2)
```

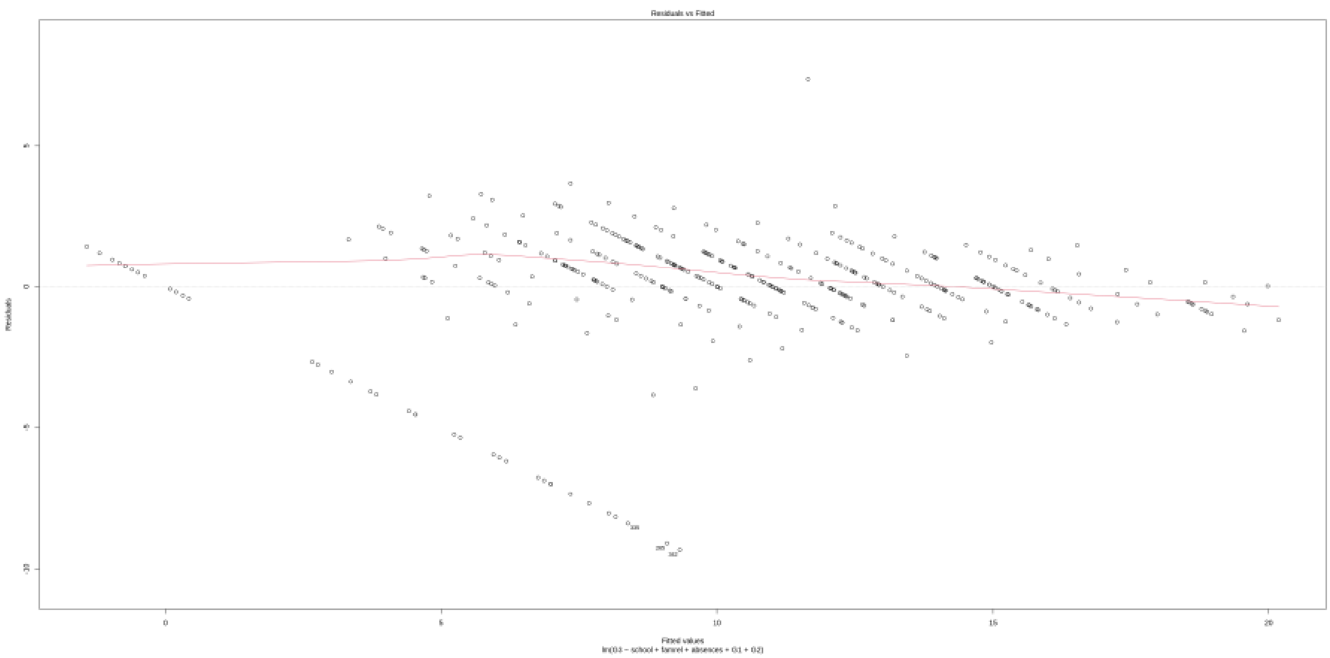


Figure 50: *Residuals vs Fitted.*

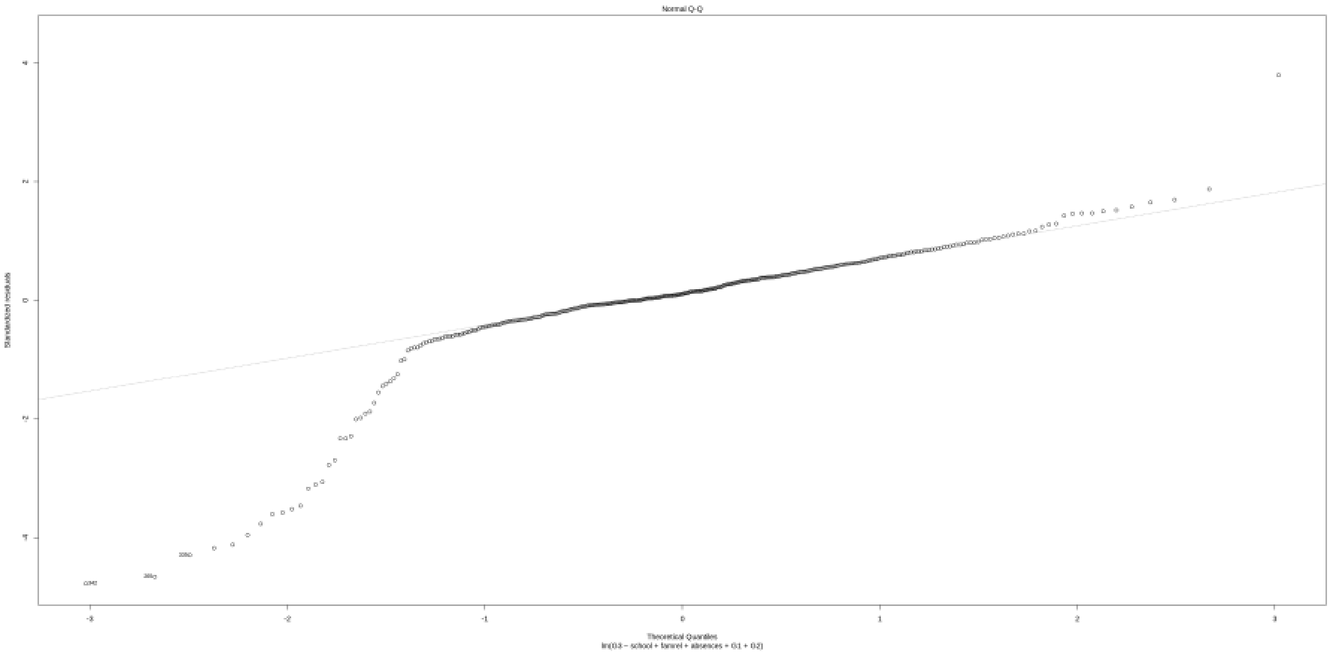


Figure 51: *Normal Q-Q.*

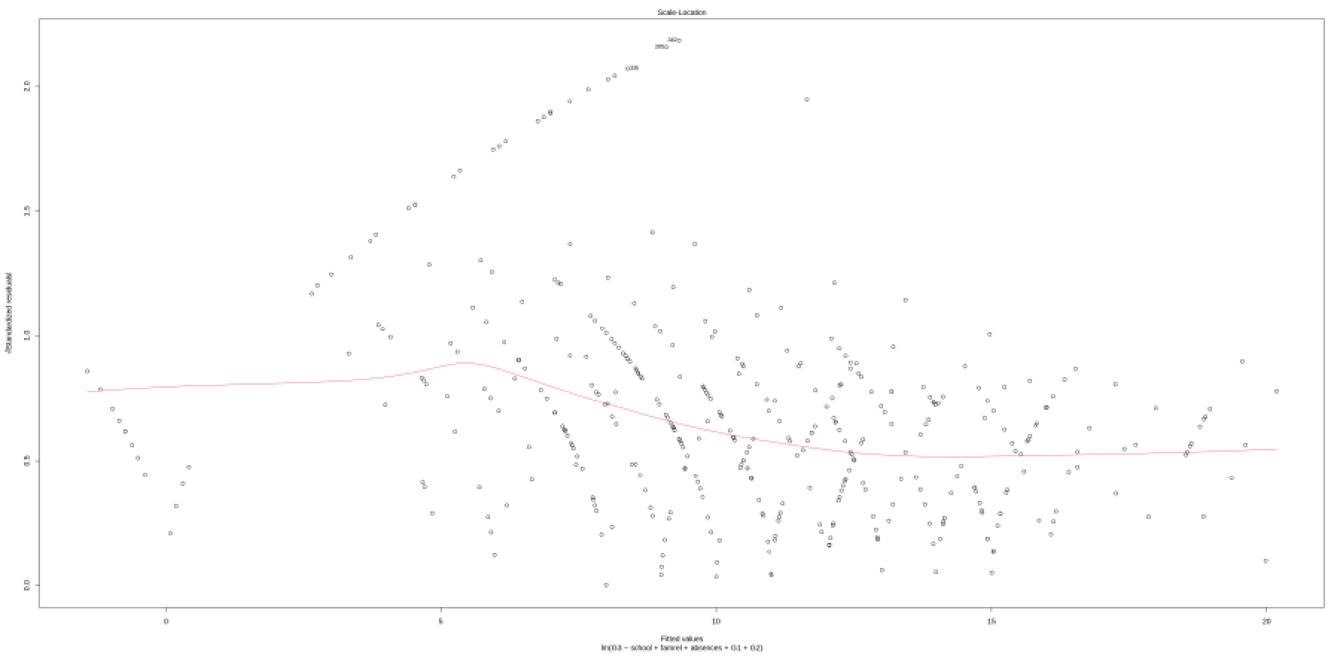


Figure 52: *Scale-Location.*

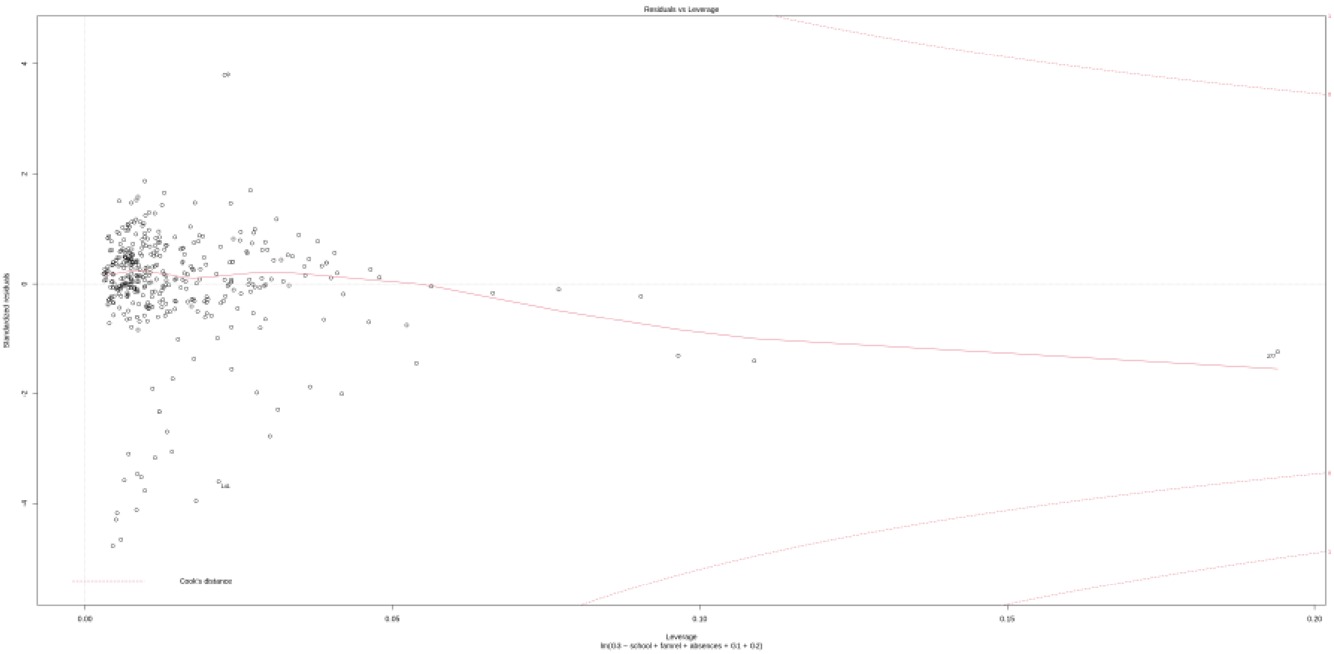


Figure 53: *Residuals vs Leverage.*

1.2.4 Predictions

1.2.4.1. Evaluation

First, in order to evaluate whether those students passed or failed based on final grade, the condition order: *if their final grade is not less than 10, they are passed*; which is used to *evaluate*. After that step, the prediction data also is built as the same function above but predict_G3.

```
evaluate = gradeData$G3
evaluate = ifelse(evaluate >=10,"pass","fail")
observe = table(evaluate)
View (observe)
```

Figure 54: *The code for evaluate.*

```
evaluate
fail pass
 130  265
```

Figure 55: *The result of evaluate.*

```
Predict_G3 = predict(LinearModel_2,gradeData)
Predict_G3 = ifelse(Predict_G3>=10, "pass", "fail")
observe = table(Predict_G3)
View (observe)
```

Figure 56: *The code for Predict_G3.*

```
Predict_G3
fail pass
 185  210
```

Figure 57: *The result of Predict_G3.*

The percent error for students who failed is $\frac{185-130}{130}$ x $100\% = 42.31\%$.

The percent error for students who passed is $\frac{265-210}{265}$ x $100\% = 20.75\%$.

1.2.4.2. Prediction a new data

First, creating a data frame to predict the final grade. As below, the new data frame is given as an example

```
newd = data.frame(school = 1,famrel =5,absences =20, G1 =10, G2 =11)
```

Then, using *predict* command to compute G3 (final grade) from the others factor in the data frame.

```
G3_predict = predict(LinearModel_2,newd)
```

And using *round* command to round the result.

```
round(G3_predict, digits = 4)
```

Then, we will have the result.

```
1: 11.4671
```

Finally, the final result computed by R is **11.4671**.

2 Activity 2

2.1 Problem

For this activity, we use a dataset that approaches the influence of parents educational level in guiding children to prepare homework to take the exam. The data contains several factors that are considered to influence the average score of student.

There are 3 attributes that will be focused on in this activity:

- *ParentLevel*: The education level of each student (binary: 0 - ***bachelor's degree***, ***master's degree***, ***associate's degree*** or 1 - ***high school***, ***some college***, ***some highe school***).
- *TestPreparation*: The preparation before having a test (binary: 0 - ***completed*** or 1 - ***none***).
- *AverageScore*: Student's average score (numeric: **0 - 100**)

We want to know whether the education level of parents and the preparation before having a test affects the average score of student or not.

2.2 Solution

2.2.1 Import Data

First, we will install and calling necessary library. After that, reading dataset and choosing needed elements will be the next step.

```r
install.packages("car")
library(car)
```

Figure 58: *Installing and calling.*

```r
#choose 3 variables Parent, Preparation, Average score
df <- df[,c('ParentLevel', 'TestPreparation','AverageScore')]
head(df)
dim(df)
```

Figure 59: *Read and choose elements.*

Select necessary variables, which are "ParentLevel", "TestPreparation" and "AverageScore".

```
> #choose 3 variables Parent, Preparation, Average score
> df <- df[,c('ParentLevel', 'TestPreparation','AverageScore')]

> head(df)
         ParentLevel TestPreparation AverageScore
1  bachelor's degree            none           73
2       some college       completed           83
3    master's degree            none           93
4 associate's degree            none           50
5       some college            none           77
6 associate's degree            none           78

> dim(df)
[1] 1000    3
```

Figure 60: *There are 1000 students that the experiment be conducted on*

2.2.2 Data Visualizaion

2.2.2.1. Transformation

To utilize R program to calculate, all factors or values from the dataset must be transferred to numeric type. Before the transformation process is coded, several implies are established for thorough understanding in order to convert these values to numerical values.

```
#data transformation
df[df == "completed"] <- 0
df[df == "none"] <- 1
df[df == "bachelor's degree"] <- 0
df[df == "master's degree"] <- 0
df[df == "associate's degree"] <- 0
df[df == "high school"] <- 1
df[df == "some college"] <- 1
df[df == "some high school"] <- 1
```

Figure 61: *Converting to numerical values*

And then, converting to specific value to plot the paragraph

```
> df
   ParentLevel TestPreparation AverageScore
1         High    Not-Prepared           73
2          Low        Prepared           83
3         High    Not-Prepared           93
4         High    Not-Prepared           50
5          Low    Not-Prepared           77
6         High    Not-Prepared           78
7          Low        Prepared           92
8          Low    Not-Prepared           41
9          Low        Prepared           65
10         Low    Not-Prepared           50
11        High    Not-Prepared           55
12        High    Not-Prepared           45
13         Low    Not-Prepared           73
14         Low        Prepared           74
15        High    Not-Prepared           54
16         Low    Not-Prepared           74
17         Low    Not-Prepared           88
```

Figure 62: *Converting to specific value.*

```
df$ParentLevel[df$ParentLevel == 0] <- "High"
df$ParentLevel[df$ParentLevel == 1] <- "Low"
df$TestPreparation[df$TestPreparation == 0] <- "Prepared"
df$TestPreparation[df$TestPreparation == 1] <- "Not-Prepared"
```

Figure 63: *Example for code.*

2.2.2.2. Visualization

The frequency of each parent level line type is plotted as followed:

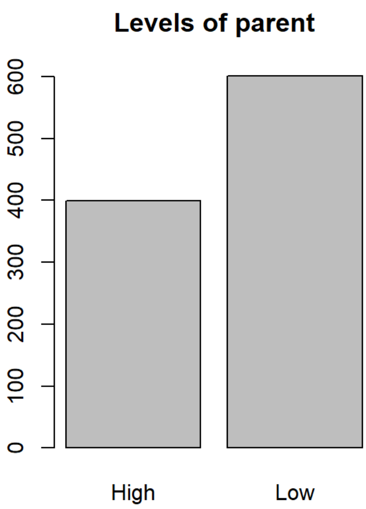


Figure 64: *Levels of parents.*

```
#parent level
barplot(table(df$ParentLevel), main="Levels of parent", names.arg = c("High","Low"))
```

Figure 65: *Example for code.*

And, the same for status of preparation

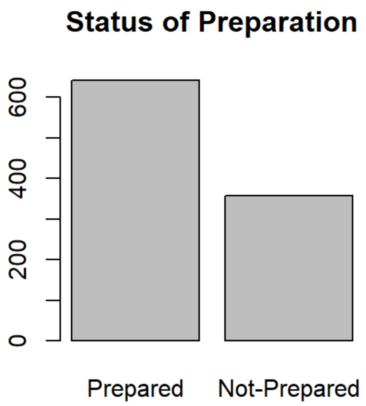


Figure 66: *Status of Preparation.*

```
#preparation
barplot(table(df$TestPreparation), main="Status of Preparation", names.arg = c("Prepared","Not-Prepared"))
```

Figure 67: *Example for code.*

After that, we will retransform data to integer to plot against different combinations.

```
#re-transform data to binary value
df$ParentLevel[df$ParentLevel == "High"] <- 0
df$ParentLevel[df$ParentLevel == "Low"] <- 1
df$TestPreparation[df$TestPreparation == "Prepared"] <- 0
df$TestPreparation[df$TestPreparation == "Not-Prepared"] <- 1
```

Figure 68: *Example for codes.*

Then, receptivity rating is plotted separately against different combinations of ParentLevel and Preparation.

```
#plot
boxplot(df$AverageScore[df$ParentLevel == 0][df$TestPrepara
ation == 0], df$AverageScore[df$ParentLevel == 1][df$TestPre
paration == 0], df$AverageScore[df$ParentLevel == 0][df$Tes
tPreparation == 1], df$AverageScore[df$ParentLevel == 1][df$
TestPreparation == 1], ylab = "Average Score", main="Averag
ore for each parent's level and student's preparation", name
n", names = c("High-Prepared", "Low-Prepared", "High-NotPrepared", "Low-NotPrepared"))
```

Figure 69: *Example for code.*

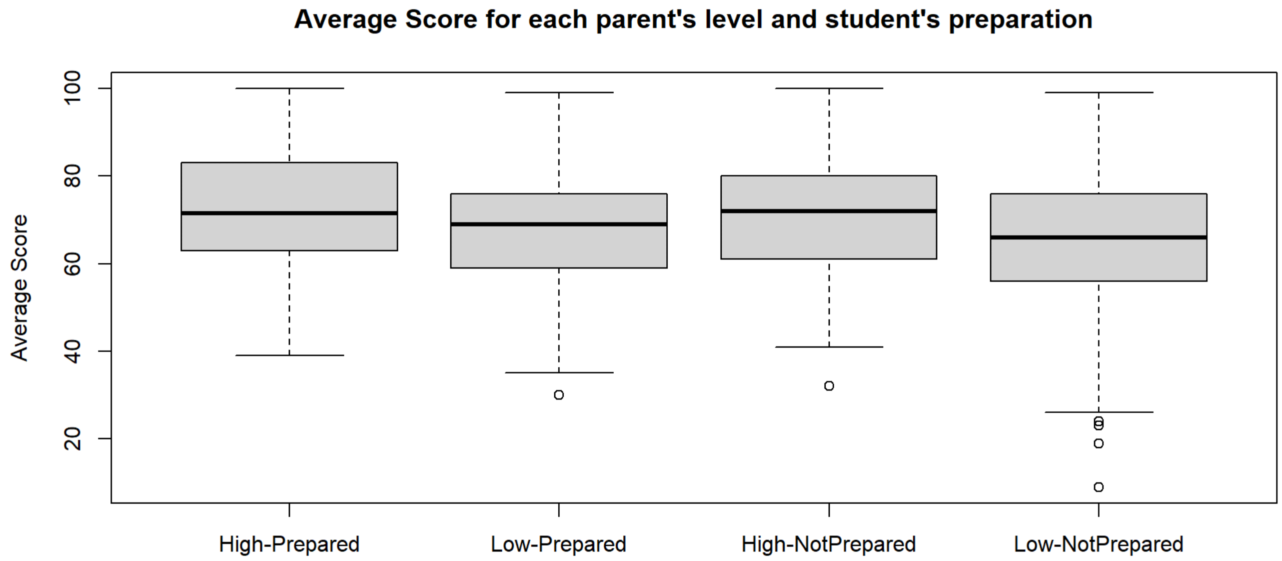


Figure 70: *Different combinations of ParentLevel and Preparation.*

2.2.3 Model of Variances Analysis

At the significance level $\alpha = 5\%$, we test the 3 following hypotheses

- H_{0a}: Different types of ParentLevel lines do not affect the rating of average student's score (Main effect for ParentLevel Line).
- H_{0b}: The preparation of test does not affect the rating of average student's score (Main effect for TestPreparation).
- H_{0c}: There is no interaction between types of ParentLevel lines and TestPreparation on the average student's score (Interaction effect).

Respectively, we have 3 alternative hypotheses:

- H_{1a}: Different types of ParentLevel lines affect the rating of average student's score.
- H_{1b}: THe preparation of test affects the rating of average student's score.
- H_{1c}: There is an interaction between types of ParentLevel lines and TestPreparation on the average student's score.

Since we are analyzing the effects of 2 independent variables: ParentLevel Lines and TestPreparation on 1 dependent variable, which is the average score of student, Two-Way ANOVA is applied for the model of variances.

To test Two-Way ANOVA with both main effects and interaction effect, we used aov() function with command Receptivity $\sim$ ParentLevel $*$ TestPreparation where "$*$" indicates interaction.

Here for the codes:

```
#model of variance analysis
summary(aov(AverageScore ~ ParentLevel*TestPreparation, data = df))
```

Figure 71: *Example for code.*

Following is the result:

```
> #model of variance analysis
> summary(aov(AverageScore ~ ParentLevel*TestPreparation, data = df))
                             Df Sum Sq Mean Sq F value   Pr(>F)    
ParentLevel                   1   6331    6331  34.299 6.42e-09 ***
TestPreparation               1  12922   12922  70.007  < 2e-16 ***
ParentLevel:TestPreparation   1      1       1   0.008     0.93    
Residuals                   996 183836     185                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 72: *The result of code.*

As can be seen from the image, the Sum Squares, Mean Squares, F values and p values for 3 hypothesis tests were shown in the first 3 rows respectively:

- The first row tests the effect of ParentLevel Line types on the average score of student. Because p-value (6.42e-09) is much smaller than α (0.05), H_{0a} is rejected.
- The second row tests the effect of TestPrepration on the average score of stuent. As p-value (¡ 2e-16) is significantly smaller than α (0.05), H_{0b} is rejected.
- The third row testes the interaction effect between ParentLevel Line types and TestPreparation. Since value (0.93) is greater than α (0.05), H_{0c} is not rejected.

Conclusion: With significance level $\alpha = 0.05$, we have evidence to confirm that different ParentLevel types affects the average student's score and there does not have an interaction between types of ParentLevel lines and TestPreparation on the average student's score.

2.2.4 Model adequacy checking

ANOVA assumes that observations are independent normally distributed and variances between groups are homogeneous. The assumption of independence can be guaranteed, as the experiments are conducted randomly from students. Now we need to check for the homogeneity of variance and the normality assumptions to see whether our model is valid or not.

2.2.4.1. Homogeneity of variances assumption

There are 2 levels of "TestPreparation", 2 levels of "ParentLevel", in total there are 4 groups of combination.

Here for the codes:

```
#1. homogeneity of variances assumption
ANOVA <- aov(AverageScore ~ ParentLevel*TestPreparation, data = df)
plot(ANOVA,1)
leveneTest(AverageScore ~ as.factor(ParentLevel)*as.factor(TestPreparation), data = df)
```

Figure 73: *The example of code.*

The residual plots for each group:

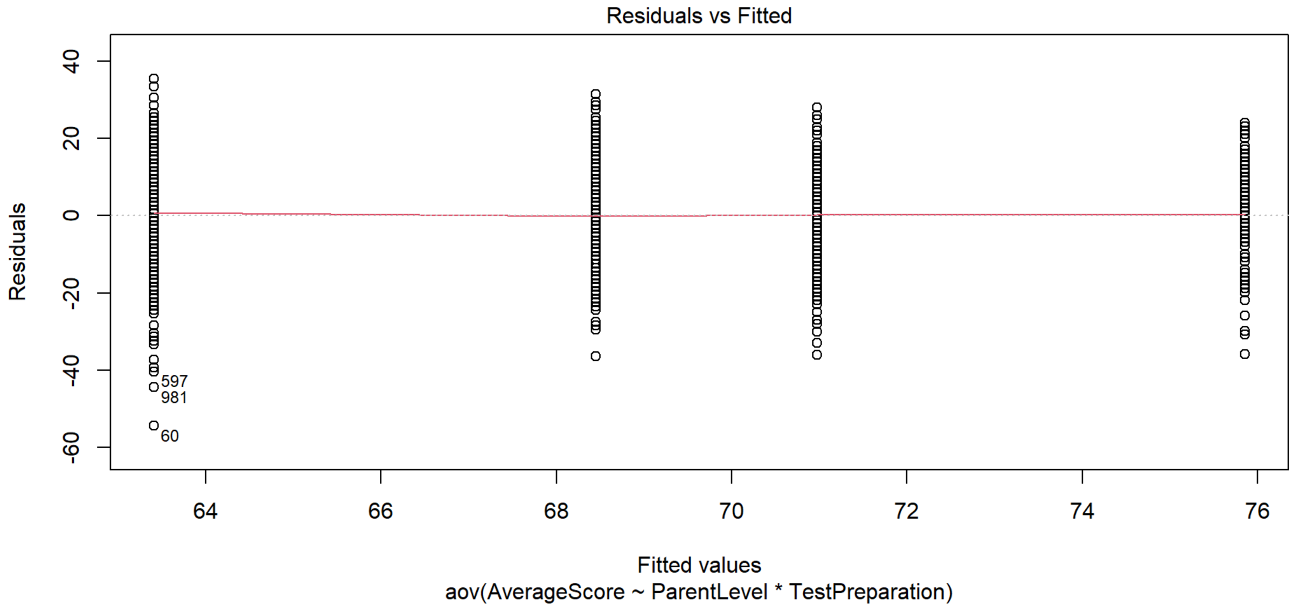


Figure 74: *The result of codes.*

Although there are some outliers such as point 60, the variances seem to be the same between groups. The data variance of 2 middle groups may be slightly smaller but it is acceptable. No strange patterns found in the residual plots, indicating the homogeneity of variance.

Levene's test can also be used to check the assumption of constant variances, by using function leveneTest from the package *Car*:

```
Levene's Test for Homogeneity of Variance (center = median)
       Df F value Pr(>F)
group   3  0.9142 0.4334
      996
```

Figure 75: *The example of codes.*

From the output above we can see that p-value is much larger than the significance level of 0.05. This means that we do not have enough evidence to suggest that variance across groups is statistically significant different. Therefore, we can assume the homogeneity of variances.

2.2.4.2. Normality assumption

We use the normality plot of residuals (Q-Q plot), in which the quantiles of the residuals are plotted against the quantiles of the normal distribution. If the normality assumption is correct, the plot of residuals should approximately follows a straight line.

Standardized residuals plot are used instead of residual plot, the result must be the same:

```
> #2. normality assumption
> plot(ANOVA,2)
```

Figure 76: *The example of codes.*

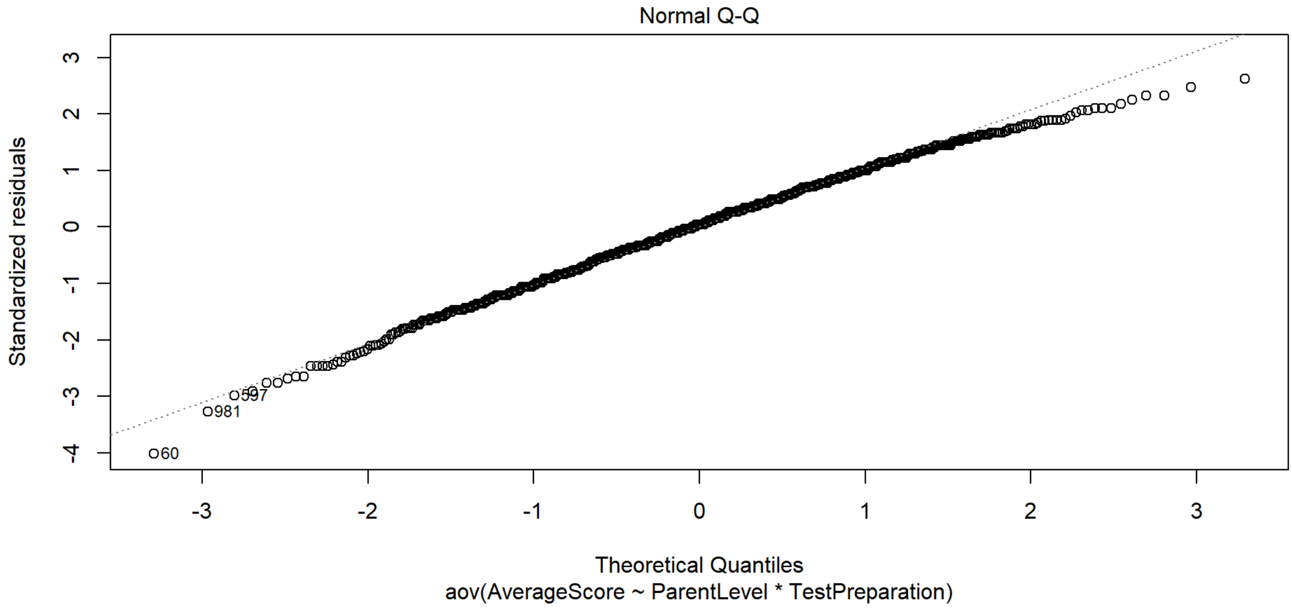


Figure 77: *The result of codes.*

As can be seen from the image, all the points fall approximately along the reference line, so we can assume normality of our data.

With the normality and variances homogeneity assumptions validated, the results from our two factor ANOVA test are more reliable.

3 Bibliography

References

[1] Source code Activity 1 - we run directly on the google collab and then coverting to the R file.
Available here.

[2] Source code entire the assignement - we put they on the github .
Available here.

[3] R-tutor.com. 2021. *Estimated Multiple Regression Equation — R Tutorial.*
Available here [Accessed 1 March 2021].

[4] Advstats.psychstat.org. 2021. *Relative Importance of Predictors – Advanced Statistics using R.*
Available here [Accessed 1 March 2021].

[5] Youtube.com. 2021. *R Stats: Multiple Regression - Variable Selection.*
Available here [Accessed 1 March 2021].

[6] Phillips, N., 2021. *YaRrr! The Pirate's Guide to R.*
Available here [Accessed 3 March 2021].

[7] Nguyen Van, T., 2006. *PHAN TICH SO LIEU VA TAO BIEU DO BANG R.* Ho Chi Minh City: Nha xuat ban Dai hoc Bach Khoa TP. Ho Chi Minh.

[8] Archive.ics.uci.edu. 2021. *Wine quality dataset.*
Available here [Accessed 10 March 2021].