

ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA



BÀI TẬP LỚN MÔN XÁC SUẤT THỐNG KÊ

**PHẦN CHUNG – ĐỀ TÀI 2: PHÂN TÍCH ĐIỂM TOÁN CỦA CÁC
EM HỌC SINH TRUNG HỌC**

PHẦN RIÊNG – HIỆU NĂNG CỦA CPU MÁY TÍNH

GVHD: PGS.TS Nguyễn Đình Huy

LỚP L13

Sinh viên thực hiện: Nguyễn Đức An

MSSV: 2010102

Thành phố Hồ Chí Minh - 2021

MỤC LỤC

HOẠT ĐỘNG 1 - PHẦN CHUNG - ĐỀ TÀI 2: PHÂN TÍCH ĐIỂM TOÁN CỦA CÁC EM HỌC SINH TRUNG HỌC	2
1. LỜI MỞ ĐẦU	2
2. ĐỀ BÀI	2
3. THỰC HÀNH VỚI R.....	3
3.1. Đọc dữ liệu.....	3
3.2. Làm sạch dữ liệu (Data cleaning).....	4
3.3. Làm rõ dữ liệu.....	7
3.4. Xây dựng mô hình hồi quy tuyến tính	17
3.5. Dự đoán.....	25
HOẠT ĐỘNG 2 -PHẦN RIÊNG – HIỆU NĂNG CỦA CPU MÁY TÍNH:	29
1. ĐỀ BÀI:	29
2. THỰC HÀNH VỚI R:	29
2.1. Nhập dữ liệu:	29
2.2. Làm sạch dữ liệu:	30
2.3. Làm rõ dữ liệu.....	31
2.4. Anova một nhân tố so sánh về hiệu suất giữa các hãng:.....	36
2.5. Xây dựng các mô hình hồi quy tuyến tính	37
TỔNG KẾT	41
TÀI LIỆU THAM KHẢO.....	42

**PHẦN CHUNG – ĐỀ TÀI 2: PHÂN TÍCH ĐIỂM TOÁN CỦA CÁC EM HỌC SINH
TRUNG HỌC**

THÀNH VIÊN NHÓM 2

Họ và tên	MSSV
Nguyễn Đức An	2010102
Trần Mỹ Trinh	1915647
Nguyễn Thị Thanh Thùy	1915403

PHẦN RIÊNG – HIỆU NĂNG CỦA CPU MÁY TÍNH

THÀNH VIÊN NHÓM (KHOA KH&KT MÁY TÍNH)

Họ và tên	MSSV
Nguyễn Đức An	2010102
Lê Minh Nghĩa	2010445
Huỳnh Tấn Lộc	2010391

HOẠT ĐỘNG 1 - PHẦN CHUNG - ĐỀ TÀI 2: PHÂN TÍCH ĐIỂM TOÁN CỦA CÁC EM HỌC SINH TRUNG HỌC

1. LỜI MỞ ĐẦU

Trong môn Xác suất và Thống kê, chúng em đã được tìm hiểu về các lý thuyết và công thức để tính toán thống kê bằng tay. Tuy nhiên, đối với các tệp dữ liệu lớn, việc tính toán bằng tay dường như là không thể. Thay vào đó, người ta sử dụng các ứng dụng phân tích dữ liệu để xử lý dữ liệu chính xác và nhanh gọn nhất.

Một số công cụ phân tích dữ liệu thường dùng có thể kể đến đó là Microsoft Excel, Python, SPSS hay R. Trong bài báo cáo này, chúng em được yêu cầu sử dụng phần mềm Rstudio để xử lý các số liệu thống kê.

2. ĐỀ BÀI

Tập tin "**diem_so.csv**" chứa thông tin về điểm toán của các em học sinh trung học thuộc hai trường học ở Bồ Đào Nha. Các thuộc tính dữ liệu bao gồm điểm học sinh, nơi cư trú, và một số hoạt động xã hội khác. Dữ liệu được thu thập bằng cách sử dụng báo cáo của các trường và các kết quả khảo sát sinh viên. Dữ liệu gốc được cung cấp tại: <https://archive.ics.uci.edu/ml/datasets/student+performance>.

Các biến chính trong bộ dữ liệu:

- **G1**: Điểm thi học kì 1.
- **G2**: Điểm thi học kì 2.
- **G3**: Điểm cuối khoá.
- **studytime**: Thời gian tự học trên tuần (1 - ít hơn 2 giờ, 2 - từ 2 đến 5 giờ, 3 - từ 5- 10 giờ, or 4 - lớn hơn 10 giờ).
- **failures**: số lần không qua môn (1,2,3, hoặc 4 chỉ nhiều hơn hoặc bằng 4 lần).
- **absences**: số lần nghỉ học.
- **paid** - Có tham gia các lớp học thêm môn Toán ngoài trường (có/không).
- **sex**: Giới tính của học sinh. (Nam/nữ).

Các bước thực hiện:

1. Đọc dữ liệu (Import data): **grade.csv**
2. Làm sạch dữ liệu (Data cleaning): NA (dữ liệu khuyết)

3. Làm rõ dữ liệu: (Data visualization)

(a) Chuyển đổi biến (nếu cần thiết).

(b) Thống kê mô tả: dùng thống kê mẫu và dùng đồ thị.

4. Xây dựng mô hình hồi quy tuyến tính để đánh giá các nhân tố có thể ảnh hưởng đến điểm thi cuối kỳ của sinh viên.

5. Thực hiện dự báo cho điểm Toán của học sinh.

3. THỰC HÀNH VỚI R

3.1. Đọc dữ liệu

Bài làm:

- Dùng lệnh `read.csv()` để đọc tập tin:

Trước khi dùng lệnh `read.csv()`, ta cần lưu ý đặt tệp cần đọc chung 1 thư mục với file R đang mở, sau đó thực hiện câu lệnh trên trong R theo đúng cú pháp:

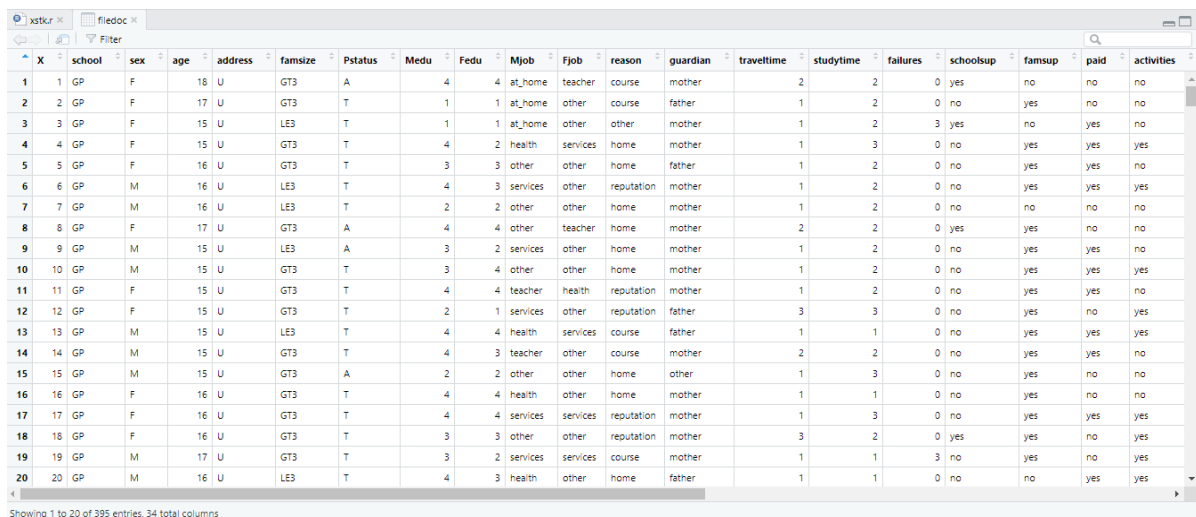
***tên_biến chứa dữ liệu* <- read.csv(file = '*tên dữ liệu cần đọc*')**

Với tên biến chứa dữ liệu có thể chọn tùy ý nhưng phải tuân theo quy tắc tên biến trong R.

```
filedoc <- read.csv(file = 'diem_so.csv')
```

Hình 1. Hình minh họa câu lệnh

Sau khi đọc xong, dữ liệu của chúng ta sẽ được lưu vào 1 biến kiểu data frame tên `filedoc`, ta có thể xem qua biến này bằng cách dùng lệnh `view()` hoặc click đôi vào tên biến ở góc phải màn hình. Lúc này màn hình sẽ hiển thị:



X	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime	studytime	failures	schoolsup	famsup	paid	activities
1	GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother	2	2	0	yes	no	no	no
2	GP	F	17	U	GT3	T	1	1	at_home	other	course	father	1	2	0	no	yes	no	no
3	GP	F	15	U	LE3	T	1	1	at_home	other	other	mother	1	2	3	yes	no	yes	no
4	GP	F	15	U	GT3	T	4	2	health	services	home	mother	1	3	0	no	yes	yes	yes
5	GP	F	16	U	GT3	T	3	3	other	other	home	father	1	2	0	no	yes	yes	no
6	GP	M	16	U	LE3	T	4	3	services	other	reputation	mother	1	2	0	no	yes	yes	yes
7	GP	M	16	U	LE3	T	2	2	other	other	home	mother	1	2	0	no	no	no	no
8	GP	F	17	U	GT3	A	4	4	other	teacher	home	mother	2	2	0	yes	yes	no	no
9	GP	M	15	U	LE3	A	3	2	services	other	home	mother	1	2	0	no	yes	yes	no
10	GP	M	15	U	GT3	T	3	4	other	other	home	mother	1	2	0	no	yes	yes	yes
11	GP	F	15	U	GT3	T	4	4	teacher	health	reputation	mother	1	2	0	no	yes	yes	no
12	GP	F	15	U	GT3	T	2	1	services	other	reputation	father	3	3	0	no	yes	no	yes
13	GP	M	15	U	LE3	T	4	4	health	services	course	father	1	1	0	no	yes	yes	yes
14	GP	M	15	U	GT3	T	4	3	teacher	other	course	mother	2	2	0	no	yes	yes	no
15	GP	M	15	U	GT3	A	2	2	other	other	home	other	1	3	0	no	yes	no	no
16	GP	F	16	U	GT3	T	4	4	health	other	home	mother	1	1	0	no	yes	no	no
17	GP	F	16	U	GT3	T	4	4	services	services	reputation	mother	1	3	0	no	yes	yes	yes
18	GP	F	16	U	GT3	T	3	3	other	other	reputation	mother	3	2	0	yes	yes	no	yes
19	GP	M	17	U	GT3	T	3	2	services	services	course	mother	1	1	3	no	yes	no	yes
20	GP	M	16	U	LE3	T	4	3	health	other	home	father	1	1	0	no	no	yes	yes

Hình 2. Hình minh họa dữ liệu sau khi đọc vào R

3.2. Làm sạch dữ liệu (Data cleaning)

a) Trích ra một dữ liệu con đặt tên là **new_DF** chỉ bao gồm các biến chính mà ta quan tâm như đã trình bày trong phần giới thiệu dữ liệu. Từ câu này về sau, mọi yêu cầu xử lý đều dựa trên tập dữ liệu con **new_DF** này.

Bài làm:

Để trích ra dữ liệu con trong R, ta cần phải dùng đến lệnh `subset()` với cú pháp câu lệnh là:

```
*tên_biến* <- subset(*tên_dữ_liệu_chính*, select = c())
```

Trong đó:

- ***tên_biến*** là tên người dùng chọn để đặt, trong trường hợp này tên biến sẽ là `new_DF` để đúng ý đề bài đặt ra.
- ***tên_dữ_liệu_chính*** là tên của dữ liệu gốc mà mình muốn trích dữ liệu con ra, trong trường hợp này, biến này có tên là `filedoc`.
- **select = c()** là tham số dùng để chọn ra những cột mà mình muốn trích trong dữ liệu con. Tên của những cột này sẽ phải được đặt trong dấu ngoặc kép `" "` và ngăn cách nhau bởi dấu phẩy.

Ngoài ra, lệnh `subset` còn có nhiều tham số khác có thể sử dụng tùy vào mục đích người dùng. Chúng ta có thể tìm hiểu kỹ hơn nếu có nhu cầu.

```
new_DF <- subset(filedoc, select = c("G1", "G2", "G3", "studytime", "failures", "absences", "paid", "sex"))
```

Hình 3. Hình minh họa sử dụng câu lệnh `subset()`

Lúc này, dữ liệu con mới nhận được sẽ có dạng như sau:

	G1	G2	G3	studytime	failures	absences	paid	sex
1	5	6	6	2	0	6	no	F
2	5	NA	6	2	0	4	no	F
3	7	8	10	2	3	10	yes	F
4	15	14	15	3	0	2	yes	F
5	6	10	10	2	0	4	yes	F
6	15	NA	15	2	0	10	yes	M
7	12	12	11	2	0	0	no	M
8	6	5	6	2	0	6	no	F
9	16	NA	19	2	0	0	yes	M
10	14	15	15	2	0	0	yes	M
11	10	8	9	2	0	0	yes	F
12	10	12	12	3	0	4	no	F
13	14	14	14	1	0	2	yes	M
14	10	10	11	2	0	2	yes	M
15	14	16	16	3	0	0	no	M
16	14	14	14	1	0	4	no	F
17	13	14	14	3	0	6	yes	F
18	8	10	10	2	0	4	no	F
19	6	5	5	1	3	16	no	M
20	8	10	10	1	0	4	yes	M
21	13	14	15	2	0	0	no	M

Showing 1 to 21 of 395 entries, 8 total columns

Hình 4. Dữ liệu con nhận được

b) Kiểm tra các dữ liệu bị khuyết trong tập tin. (Các câu lệnh tham khảo: `is.na()`, `which`, `apply()`). Nếu có dữ liệu bị khuyết hãy đề xuất phương pháp thay thế cho những dữ liệu này.

Bài làm

Ta sử dụng lệnh `is.na()` để tìm kiếm những dữ liệu bị khuyết. Hàm `is.na()` sẽ trả về cho ta 1 vector 2 chiều kiểu logic, ô dữ liệu nào bị khuyết trong tập dữ liệu nhập vào sẽ có giá trị TRUE trong ô tương ứng của vector được trả về theo hàm. Câu lệnh minh họa:

```
x <- is.na(new_DF)
```

Mở biến `x`, ta nhận được bảng sau:

	G1	G2	G3	studytime	failures	absences	paid	sex
1	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
3	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
4	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
5	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
6	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
7	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
8	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
9	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
10	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
11	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
12	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
13	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
14	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
15	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
16	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
17	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
18	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
19	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
20	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
21	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

Showing 1 to 21 of 395 entries, 8 total columns

Hình 5. Bảng giá trị NA nhận được

Vì dữ liệu chúng ta rất nhiều, nên ta không thể dùng mắt thường để tìm kiếm cụ thể những dữ liệu nào sẽ bị khuyết được. Để giải quyết vấn đề này, chúng ta sẽ sử dụng toán tử `%in%` để tìm xem cột nào có giá trị TRUE khi gọi hàm `is.na()`, tương ứng với việc cột đó có dữ liệu bị khuyết.

Câu lệnh sử dụng:

- `g1na <- TRUE %in% x[, "G1"]`
- `g2na <- TRUE %in% x[, "G2"]`
- `g3na <- TRUE %in% x[, "G3"]`
- `sna <- TRUE %in% x[, "studytime"]`
- `fna <- TRUE %in% x[, "failures"]`
- `abna <- TRUE %in% x[, "absences"]`
- `paidna <- TRUE %in% x[, "paid"]`

- `sexna <- TRUE %in% x[, "sex"]`

Ở câu lệnh này, chúng ta quan tâm thêm đến toán tử `[]`, cho phép truy cập vào phần nào của dữ liệu. Cụ thể ở câu lệnh trên, `x[, "G1"]` sẽ giúp chúng ta truy cập vào cột G1 của dữ liệu gốc. Sau đó, ta tham khảo các giá trị nhận được, ta biết rằng chỉ có cột G2 là có dữ liệu bị khuyết.

abna	FALSE
fna	FALSE
g1na	FALSE
g2na	TRUE
g3na	FALSE
paidna	FALSE
sexna	FALSE
sna	FALSE

Hình 6. Các giá trị nhận được sau khi dùng toán tử `%in%`

Để thay thế các giá trị bị khuyết này, nhóm em đã thảo luận với nhau và đưa ra giải pháp là gán những giá trị bị khuyết ấy bằng 0, biểu diễn cho việc những bạn mà cột điểm này bị khuyết có khả năng không đi thi, không tìm được dữ liệu điểm.

`new_DF[is.na(new_DF)] <- 0`

Câu lệnh này sẽ thực hiện đúng như những gì mà em vừa miêu tả ở trên, nó sẽ truy cập vào các ô dữ liệu có `is.na(new_DF) = TRUE` và gán ô đó bằng 0.

3.3. Làm rõ dữ liệu

a. Chuyển đổi biến

Xác định biến liên tục và biến phân loại:

- **Biến liên tục:** G1, G2, G3, absences
- **Biến phân loại:** studytime, failures, paid, sex

Sau đó, ta sử dụng hàm `subset()` để tạo 2 bảng dữ liệu con, chứa thông tin của từng loại biến bằng câu lệnh:

- `bienlientuc <- subset(new_DF, select = c("G1", "G2", "G3", "absences"))`

	G1	G2	G3	absences
1	5	6	6	6
2	5	0	6	4
3	7	8	10	10
4	15	14	15	2
5	6	10	10	4
6	15	0	15	10
7	12	12	11	0
8	6	5	6	6
9	16	0	19	0
10	14	15	15	0
11	10	8	9	0
12	10	12	12	4
13	14	14	14	2
14	10	10	11	2
15	14	16	16	0
16	14	14	14	4
17	13	14	14	6
18	8	10	10	4
19	6	5	5	16
20	8	10	10	4
21	13	14	15	0

Showing 1 to 21 of 395 entries, 4 total columns

Hình 7. Biến liên tục

- `bienphanloai <- subset(new_DF, select = c("studytime", "failures", "paid", "sex"))`

	studytime	failures	paid	sex
1	2	0	no	F
2	2	0	no	F
3	2	3	yes	F
4	3	0	yes	F
5	2	0	yes	F
6	2	0	yes	M
7	2	0	no	M
8	2	0	no	F
9	2	0	yes	M
10	2	0	yes	M
11	2	0	yes	F
12	3	0	no	F
13	1	0	yes	M
14	2	0	yes	M
15	3	0	no	M
16	1	0	no	F
17	3	0	yes	F
18	2	0	no	F
19	1	3	no	M
20	1	0	yes	M
21	2	0	no	M

Showing 1 to 21 of 395 entries, 4 total columns

Hình 8. Biến phân loại

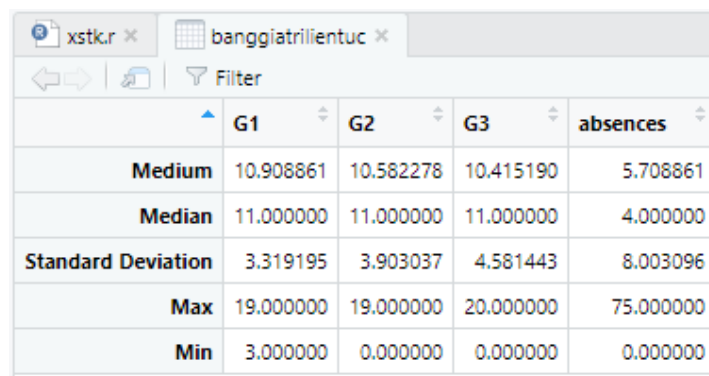
Đối với biến liên tục, ta tạo 1 bảng chứa những giá trị trung bình, trung vị, độ lệch chuẩn, giá trị lớn nhất và giá trị nhỏ nhất:

- `banggiatriliientuc <- matrix(nrow = 5, ncol = 4, dimnames = list(c("Medium", "Median", "Standard Deviation", "Max", "Min"), c("G1", "G2", "G3", "absences")))`

Hàm `matrix` sẽ tạo cho chúng ta 1 ma trận có kích thước $nrow \times ncol$, tên của từng hàng được đặt trong hàm `list` bên trong, tên của từng cột được thêm bằng hàm `c()` như đã giới thiệu ở trên. Để điền các giá trị cần tính vào trong bảng, ta sử dụng vòng lặp `for` trong R.

```
for (i in 1:4){
  banggiatriliientuc[1,i] = mean(bienlientuc[,i])
}
for (i in 1:4){
  banggiatriliientuc[2,i] = median(bienlientuc[,i])
}
for (i in 1:4){
  banggiatriliientuc[3,i] = sd(bienlientuc[,i])
}
for (i in 1:4){
  banggiatriliientuc[4,i] = max(bienlientuc[,i])
}
for (i in 1:4){
  banggiatriliientuc[5,i] = min(bienlientuc[,i])
}
```

Ta sẽ tính theo thứ tự giá trị trung bình, trung vị, độ lệch chuẩn, GTLN và GTNN của từng loại biến, sau đó sử dụng vòng lặp `for` cho chạy từ 1 đến 4 để điền vào bảng số liệu. Sau khi chạy vòng lặp, ta sẽ thu được bảng số liệu sau.



	G1	G2	G3	absences
Medium	10.908861	10.582278	10.415190	5.708861
Median	11.000000	11.000000	11.000000	4.000000
Standard Deviation	3.319195	3.903037	4.581443	8.003096
Max	19.000000	19.000000	20.000000	75.000000
Min	3.000000	0.000000	0.000000	0.000000

Hình 10. Bảng số liệu cho biến liên tục

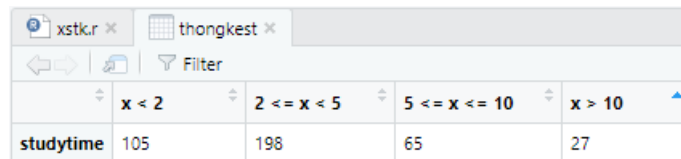
b. Thống kê mô tả: dùng thống kê mẫu và dùng đồ thị

Tạo ma trận chứa dữ liệu thống kê của biến studytime:

- **thongkest <- matrix(nrow = 1, ncol = 4, dimnames = list(c("studytime"), c("x < 2", "2 <= x < 5", "5 <= x <= 10", "x > 10")))**

Sau đó, ta sử dụng hàm sum để đếm xem có bao nhiêu giá trị thỏa mãn 1 điều kiện nào đó cho trước, từ đó thống kê được từng chủng loại của biến.

- **thongkest[,1] = sum(bienphanloai[, "studytime"] == 1)**
- **thongkest[,2] = sum(bienphanloai[, "studytime"] == 2)**
- **thongkest[,3] = sum(bienphanloai[, "studytime"] == 3)**
- **thongkest[,4] = sum(bienphanloai[, "studytime"] == 4)**



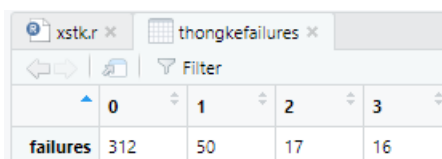
	x < 2	2 <= x < 5	5 <= x <= 10	x > 10
studytime	105	198	65	27

Hình 11. Bảng thống kê cho studytime

Nhận thấy tổng giá trị của cả 4 chủng loại bằng 395, đúng với tổng số học sinh trong dữ liệu gốc, ta có thể yên tâm về câu lệnh mình sử dụng để thống kê.

Tiếp tục thống kê cho các biến phân loại còn lại bằng những câu lệnh tương tự:

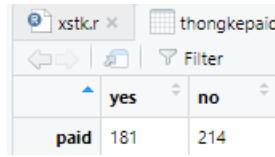
- **m <- table(bienphanloai[, "failures"])**
- **thongkefailures <- matrix(nrow = 1, ncol = 4, dimnames = list(c("failures"), c("0", "1", "2", "3")))**
- **thongkefailures[,1] <- sum(bienphanloai[, "failures"] == 0)**
- **thongkefailures[,2] <- sum(bienphanloai[, "failures"] == 1)**
- **thongkefailures[,3] <- sum(bienphanloai[, "failures"] == 2)**
- **thongkefailures[,4] <- sum(bienphanloai[, "failures"] == 3)**



	0	1	2	3
failures	312	50	17	16

Hình 12. Bảng thống kê cho biến failures

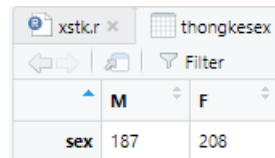
- `thongkepaid <- matrix(nrow = 1, ncol = 2, dimnames = list("paid", c("yes", "no")))`
- `thongkepaid[,1] = sum(bienphanloai[, "paid"] == "yes")`
- `thongkepaid[,2] = sum(bienphanloai[, "paid"] == "no")`



	yes	no
paid	181	214

Hình 13. Bảng thống kê cho biến paid

- `thongkesex <- matrix(nrow = 1, ncol = 2, dimnames = list("sex", c("M", "F")))`
- `thongkesex[,1] = sum(bienphanloai[, "sex"] == "M")`
- `thongkesex[,2] = sum(bienphanloai[, "sex"] == "F")`



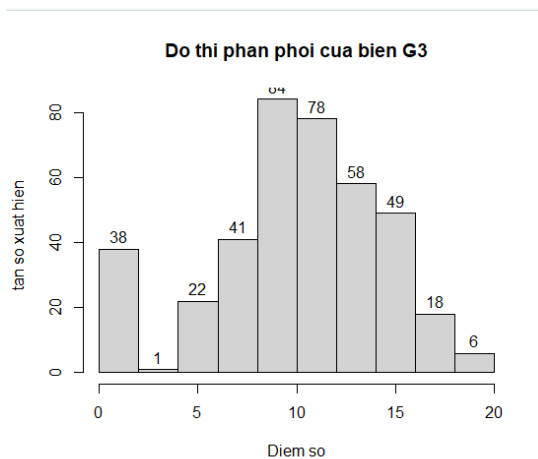
	M	F
sex	187	208

Hình 14. Bảng thống kê cho biến sex

Để vẽ đồ thị, ta sử dụng hàm `hist()` như sau:

- `hist(bienlientuc[, "G3"], main="Đồ thị phân phối của biến G3", xlab="Diem so", ylab="tan so xuất hiện", label = TRUE)`

Trong đó `main` là tên đồ thị, `xlab` và `ylab` lần lượt là tên các cột Ox, Oy mà mình muốn đặt cho. `Label = TRUE` để hiển thị chiều dài cụ thể của từng cột, hay tần suất xuất hiện cụ thể của từng mức điểm. Ta thu được đồ thị phân phối:



Hình 15. Đồ thị phân phối của biến G3

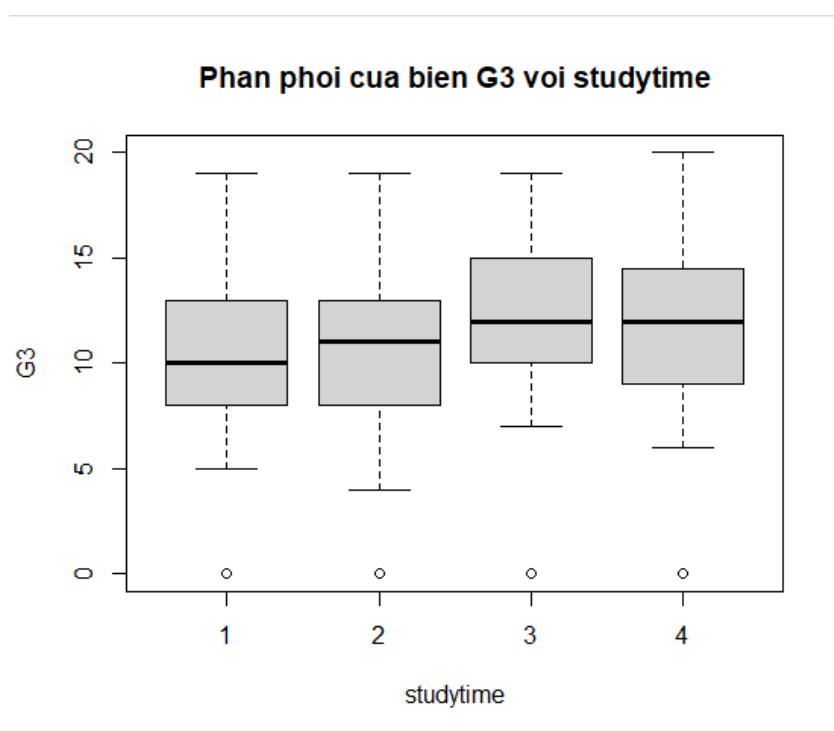
Dựa vào đồ thị, ta thấy được điểm số từ 8 đến 13 có tần suất xuất hiện nhiều hơn các giá trị điểm khác.

Câu lệnh sử dụng để vẽ phân phối của biến G3 phân loại theo biến studytime là:

- **boxplot(G3~studytime,data=new_DF, main="Phan phoi cua bien G3 voi studytime", xlab="studytime", ylab="G3")**

Trong đó, tham số data cho biết dữ liệu sử dụng để vẽ đồ thị được lấy từ đâu, toán tử ~ để xác định đồ thị phân phối của mình vẽ dựa trên 2 biến nào.

Đồ thị thu được:



Hình 16. Đồ thị phân phối của biến G3 phân loại theo biến studytime

Một số lưu ý khi đọc đồ thị boxplot:

- Ô chữ nhật màu xanh dương biểu thị phần lớn dữ liệu sẽ phân bố tập trung vào khoảng nào theo từng giá trị của biến dùng để phân loại.
- Gạch đen trong ô chữ nhật đó biểu diễn giá trị trung vị của phân phối dựa theo biến phân loại.
- 2 gạch đen ngắn hơn ở phía ngoài biểu diễn giá trị lớn nhất và nhỏ nhất của phân phối.

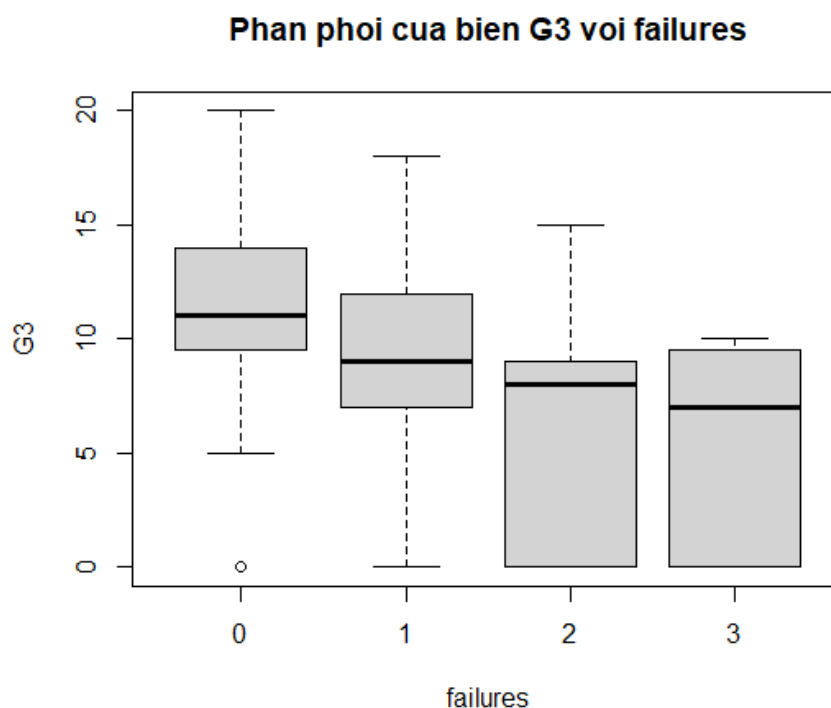
Vậy từ đồ thị trên, ta có thể rút ra được rằng: Giành ra từ 5 đến 10 giờ để học trong 1 tuần (tương ứng với mức 3) sẽ cho ra điểm số tốt nhất, giành ra quá nhiều hoặc

quá ít thời gian để học sẽ cho ra kết quả không tốt bằng (tương ứng các mức 1, 2, 4). Tuy nhiên, sự khác biệt này là không quá nhiều nên ta có thể kết luận, biến studytime không ảnh hưởng nhiều đến kết quả thi cuối kỳ.

Tương tự, ta vẽ đồ thị phân phối của biến G3 theo biến phân loại failures:

- `boxplot(G3~failures,data=new_DF, main="Phan phoi cua bien G3 voi failures", xlab="failures", ylab="G3")`

Đồ thị thu được:



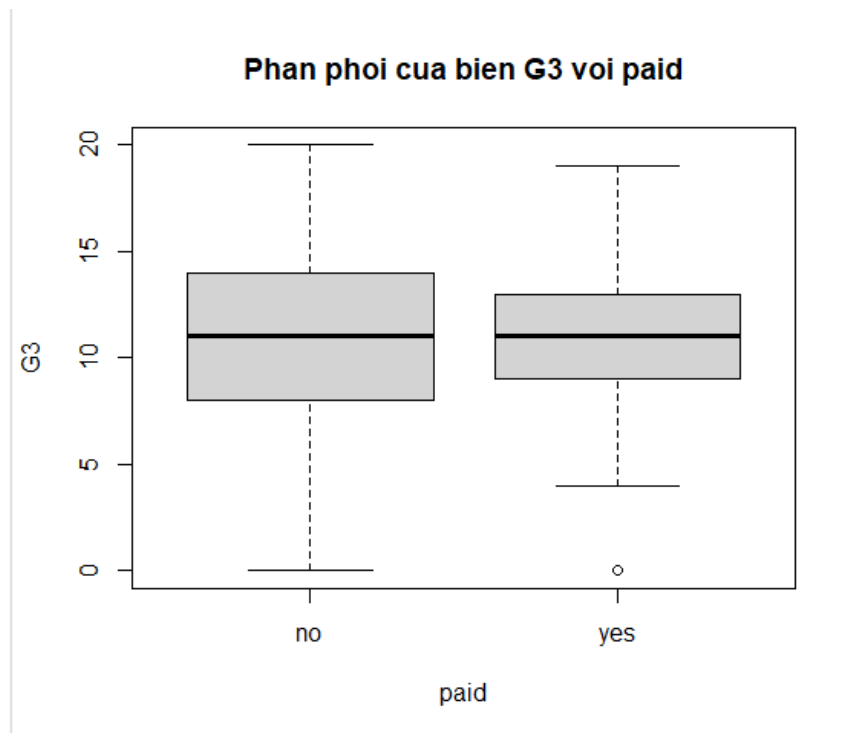
Hình 17. Đồ thị phân phối của biến G3 phân loại theo biến failures

Đồ thị trên cho ta thấy rất rõ rằng, những người chưa từng học lại khi đi thi sẽ cho ra kết quả tốt hơn, người học lại ít sẽ cho ra kết quả tốt hơn người học lại nhiều. Ngoài ra, ta có thể thấy những người đã từng học lại từ 2 lần trở lên khi đi thi cuối kỳ sẽ cho ra kết quả phần lớn nằm dưới mức 10 là mức điểm khá thấp.

Tiếp đến, ta quan sát đồ thị phân phối của biến G3 theo biến phân loại Paid:

- `boxplot(G3~paid,data=new_DF, main="Phan phoi cua bien G3 voi paid", xlab="paid", ylab="G3")`

Đồ thị thu được:



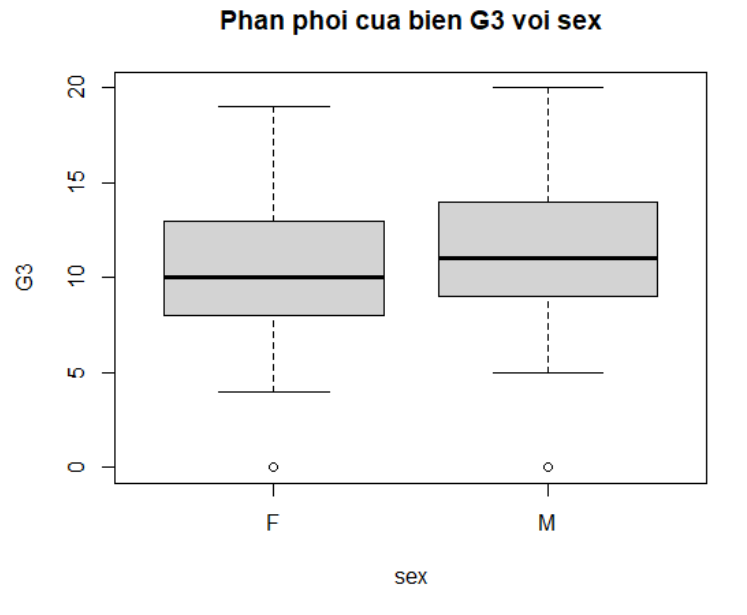
Hình 18. Đồ thị phân phối của biến G3 phân loại theo biến paid

Đồ thị trên giúp ta có được thông tin: Những người có tham gia hay không tham gia lớp học thêm Toán ngoài trường hầu hết đều có điểm số ngang nhau. Tuy nhiên, ở những người không tham gia lớp học thêm toán ngoài trường sẽ có sự khác biệt về điểm lớn hơn so với những người tham gia lớp học thêm ngoài trường.

Cuối cùng, ta quan sát đồ thị phân phối của biến G3 theo biến phân loại Sex:

- `boxplot(G3~sex,data=new_DF, main="Phan phoi cua bien G3 voi sex", xlab="sex", ylab="G3")`

Đồ thị thu được:

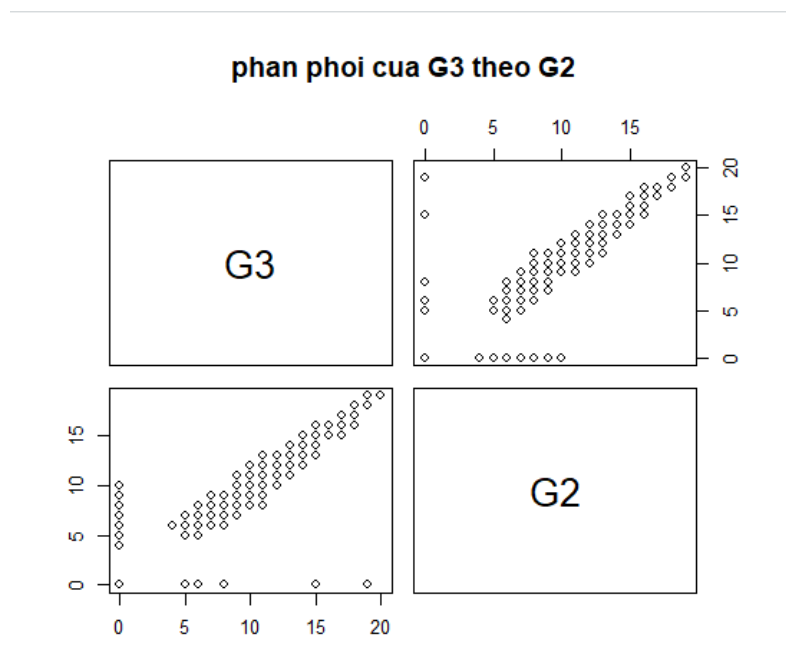


Hình 19. Đồ thị phân phối của biến G3 phân loại theo biến Sex

Ta có thể thấy ở đồ thị này, học sinh nữ có điểm số không tốt bằng học sinh nam. Tuy nhiên, sự khác biệt này là không quá nhiều nên có thể kết luận, biến sex không ảnh hưởng nhiều đến kết quả thi cuối kỳ.

Để vẽ phân phối của biến G3 theo biến G2, ta sử dụng lệnh pairs như sau:

- `pairs(G3 ~ G2, data = new_DF, main = "phan phoi cua G3 theo G2")`



Hình 19. Đồ thị phân phối của biến G3 theo biến G2

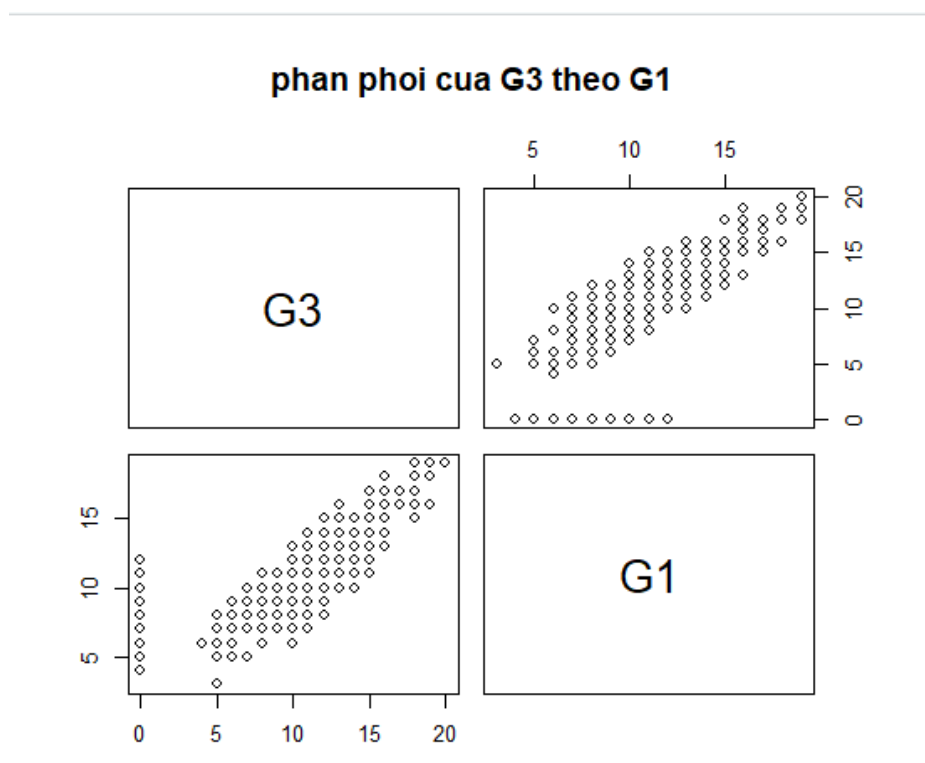
Cũng giống như `boxplot()`, lệnh `pairs()` giúp ta quan sát được phân phối của 1 biến dựa theo giá trị của 1 biến khác. Tuy nhiên, lúc này các giá trị sẽ được biểu diễn dưới dạng điểm, không phải theo khoảng như lệnh `boxplot`. Vì vậy, lệnh `pairs` sẽ phù hợp hơn nếu ta muốn quan sát phân phối của 1 biến dựa theo 1 biến liên tục.

Quan sát đồ thị, ta thấy những người có điểm thi kỳ 2 (G2) cao hơn sẽ cho ra điểm cuối kì (G3) cao hơn và ngược lại.

Tiếp tục, ta vẽ phân phối của biến G3 theo biến G1 bằng câu lệnh:

- `pairs(G3 ~ G1, data = new_DF, main = "phan phoi cua G3 theo G1")`

Đồ thị thu được:



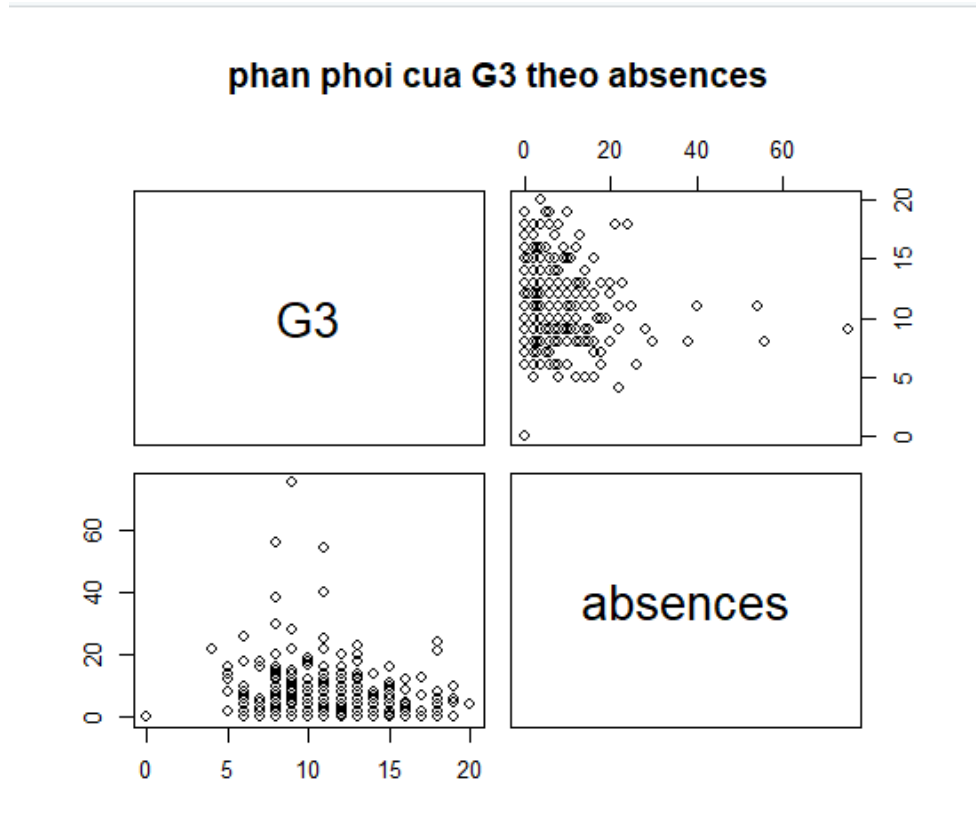
Hình 20. Đồ thị phân phối của biến G3 theo biến G1

Tương tự, đồ thị này cho ta thấy những người có kết quả thi kỳ 1 tốt sẽ cho ra kết quả thi cuối kỳ cũng tốt tương đương.

Ta tiếp tục quan sát đồ thị phân phối của biến G3 theo biến `absences` bằng câu lệnh:

- `pairs(G3 ~ absences, data = new_DF, main = "phan phoi cua G3 theo absences")`

Đồ thị thu được:



Hình 21. Đồ thị phân phối của biến G3 theo biến absences

Đồ thị này có phân phối khá dày đặc trong 1 khoảng, từ đó cho ta thông tin: Không có sự khác biệt đáng kể về điểm số giữa người đi học chăm chỉ và nghỉ tương đối (dưới 10 buổi). Số học sinh đi học chăm chỉ là khá ít so với mặt bằng chung.

Những người nghỉ quá nhiều (từ 18 buổi trở lên) sẽ có kết quả thi tệ hơn.

3.4. Xây dựng mô hình hồi quy tuyến tính

Phân tích hồi quy tuyến tính là một phương pháp phân tích quan hệ giữa biến phụ thuộc Y với một hay nhiều biến độc lập X. Mô hình hóa sử dụng hàm tuyến tính.

Các tham số của mô hình được ước lượng từ dữ liệu. Để xây dựng mô hình hồi quy tuyến tính có G3 là biến phụ thuộc, các biến còn lại là biến độc lập.

Ta sử dụng lệnh `lm()` như sau:

- **M1 <- lm(G3 ~ G1 + G2 + studytime + failures + absences + paid + sex, data =new_DF)**

```
Call:
lm(formula = G3 ~ G1 + G2 + studytime + failures + absences +
    paid + sex, data = new_DF)

Coefficients:
(Intercept)      G1      G2  studytime  failures  absences  paidyes      sexM
   -1.55801    0.43936    0.67071   -0.15589   -0.38662    0.03588    0.38620    0.31118
```

Sau đó, ta gọi hàm `summary()` để nắm được những thông tin cơ bản của mô hình hồi quy mình vừa tạo này bằng lệnh.

- **summary(M1)**

Ta nhận được thông tin:

```
Call:
lm(formula = G3 ~ G1 + G2 + studytime + failures + absences +
    paid + sex, data = new_DF)

Residuals:
    Min       1Q   Median       3Q      Max
-9.1555 -0.6883  0.2171  1.0127 13.1426

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.55801    0.53251   -2.926  0.00364 **
G1           0.43936    0.05861    7.497 4.51e-13 ***
G2           0.67071    0.04894   13.705 < 2e-16 ***
studytime   -0.15589    0.14460   -1.078  0.28167
failures    -0.38662    0.16468   -2.348  0.01939 *
absences     0.03588    0.01404    2.556  0.01098 *
paidyes      0.38620    0.23136    1.669  0.09587 .
sexM         0.31118    0.23891    1.303  0.19351

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.213 on 387 degrees of freedom
Multiple R-squared:  0.7708,    Adjusted R-squared:  0.7667
F-statistic: 185.9 on 7 and 387 DF,  p-value: < 2.2e-16
```

Hình 22. Bảng thông tin mô hình

Trong đó:

- Estimate cho ta biết được hệ số hồi quy của từng biến độc lập.
- Std.Error là sai số chuẩn của từng biến.
- t value là giá trị t sử dụng trong t-test. Dùng trong kiểm định về giả thuyết:

H_0 : Biến độc lập xi không có ảnh hưởng lớn tới biến phụ thuộc.

H_1 : Biến độc lập xi có ảnh hưởng lớn tới biến phụ thuộc.

- $Pr(>|t|)$ là giá trị P-value của kiểm định trên.

Ta chọn mức ý nghĩa 5%, dựa vào mô hình hồi quy tuyến tính trên, những biến mình sẽ loại bỏ khỏi mô hình tương ứng với mức ý nghĩa là 5%.

Ta kiểm định giả thuyết

H_0 : Biến độc lập xi không có ảnh hưởng lớn tới biến phụ thuộc.

=> H_1 : Biến độc lập xi có ảnh hưởng lớn tới biến phụ thuộc.

Ta bắt đầu thực hiện kiểm định và loại khỏi mô hình những biến không có ảnh hưởng lớn tới biến phụ thuộc được chọn.

Bước 1: Tra giá trị ngưỡng (critical value): $c = Z_{\alpha}$
 Miền bác bỏ: $W_{\alpha} = (-\infty, -Z_{\alpha}) \cup (Z_{\alpha}, +\infty)$

Bước 2: Tính giá trị quan sát hay giá trị kiểm định (test value) :

$$Z = U_{qs} = \frac{(f - P_0) \sqrt{n}}{\sqrt{P_0(1 - P_0)}}$$

Bước 3: Kết luận: $|U_{qs}| \leq Z_{\alpha} \Rightarrow H \text{ đúng} \Rightarrow P = P_0$
 $|U_{qs}| > Z_{\alpha} \Rightarrow H \text{ sai} \Rightarrow P \neq P_0$

Chú ý $P \neq P_0 \begin{cases} U_{qs} < -Z_{\alpha} \Rightarrow P < P_0 \\ U_{qs} > Z_{\alpha} \Rightarrow P > P_0 \end{cases}$

$$\begin{array}{ccccccc}
 P < P_0 & -Z_{\alpha} & & P = P_0 & & Z_{\alpha} & P > P_0 \\
 \hline
 \text{Miền bác bỏ} & | & \text{Miền chấp nhận} & | & \text{Miền bác bỏ}
 \end{array}$$

Chú ý: $P_{value} = P(|U| > U_{qs}) < \alpha \Leftrightarrow |U_{qs}| > z_{\alpha/2} = Z_{\alpha}$ 4

Hình 23: Slide hướng dẫn bài toán kiểm định giả thiết về tỉ lệ

Theo lý thuyết ta sẽ loại bỏ khỏi mô hình những biến có $p(>|t|)$ lớn hơn 5% =>

Ta sẽ loại bỏ khỏi mô hình studytime, sex và paid.

Xây dựng mô hình M2, M3:

- Mô hình M2 loại bỏ biến sex từ M1
- Mô hình M3 loại bỏ biến paid từ M2
- **M2 <- lm(G3 ~ G1 + G2 + studytime + failures + absences + paid, data = new_DF)**

- **M3 <- lm(G3 ~ G1 + G2 + studytime + failures + absences, data = new_DF)**

Để so sánh độ hiệu quả của mô hình, ta sử dụng lệnh `anova()`. Hàm `anova()` khi nhận tham số là 2 mô hình hồi quy tuyến tính sẽ trả về kết quả của kiểm định ANOVA, kiểm định giả thuyết sau:

- H_0 : Giữa 2 mô hình, mô hình đơn giản hơn cho ra kết quả tối ưu hơn.
- H_1 : Giữa 2 mô hình, mô hình phức tạp hơn cho ra kết quả tối ưu hơn.

Vậy nên, với mức ý nghĩa là 5%, tùy thuộc vào p-value nhận được mà ta ưu tiên chọn mô hình đơn giản hay phức tạp hơn. Ta bắt đầu sử dụng `anova` để so sánh các mô hình bằng những câu lệnh:

- **`anova(M1,M2)`**

```
Analysis of Variance Table

Model 1: G3 ~ G1 + G2 + studytime + failures + absences + paid + sex
Model 2: G3 ~ G1 + G2 + studytime + failures + absences + paid
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     387 1895.3
2     388 1903.6 -1     -8.309 1.6966 0.1935
```

Hình 24. Anova table giữa M1 và M2

P-value nhận được là $0.1935 > 0.05$. Vậy ta chấp nhận giả thuyết H_0 , ưu tiên chọn mô hình đơn giản hơn là mô hình M2. Ngoài ra từ kết quả trên, ta còn có thể kết luận biến `sex` không có tác động đáng kể tới mô hình M1.

Tiếp tục sử dụng `anova()` để chọn ra mô hình hợp lý hơn giữa M2 và M3 bằng câu lệnh.

- **`anova(M2,M3)`**

```
Analysis of Variance Table

Model 1: G3 ~ G1 + G2 + studytime + failures + absences + paid
Model 2: G3 ~ G1 + G2 + studytime + failures + absences
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     388 1903.6
2     389 1915.8 -1    -12.109 2.468 0.117
```

Hình 25. Anova table giữa M2 và M3

P-value nhận được là $0.117 > 0.05$. Vậy ta chấp nhận giả thuyết H_0 , ưu tiên chọn mô hình đơn giản hơn là mô hình M3. Ngoài ra từ kết quả trên, ta còn có thể kết luận biến `Paid` không có tác động đáng kể tới mô hình M2.

Mô hình tối ưu là mô hình M3, là mô hình với G3 là biến phụ thuộc và G1, G2, absences, studytime, failures là biến độc lập.

Ta gọi hàm summary() để phân tích mô hình M3

Câu lệnh sử dụng:

- **model <- M3**
- **summary(model)**

```
Call:
lm(formula = G3 ~ G1 + G2 + studytime + failures + absences,
    data = new_DF)

Residuals:
    Min       1Q   Median       3Q      Max
-9.1494 -0.6888  0.2234  1.0330 13.5656

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.21410    0.50242  -2.416  0.01613 *
G1           0.43844    0.05835   7.514 3.97e-13 ***
G2           0.67673    0.04895  13.824 < 2e-16 ***
studytime    -0.18323    0.13626  -1.345  0.17952
failures     -0.42179    0.16276  -2.591  0.00992 **
absences      0.03485    0.01402   2.486  0.01333 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.219 on 389 degrees of freedom
Multiple R-squared:  0.7683,    Adjusted R-squared:  0.7654
F-statistic: 258 on 5 and 389 DF, p-value: < 2.2e-16
```

Hình 25. Bảng phân tích mô hình M3

Dựa vào bảng phân tích, ta có phương trình hồi quy tuyến tính đa biến:

$$G3 = -1.2141 + 0.43844 \cdot G1 + 0.67673 \cdot G2 - 0.18323 \cdot \text{studytime} - 0.42179 \cdot \text{failures} + 0.03485 \cdot \text{absences}$$

Vậy là từ dữ liệu đưa vào, ta đã xây dựng được 1 hàm tuyến tính biểu thị mối liên hệ giữa G3 và các biến G1, G2, failures, absences, studytime. Từ hàm hồi quy tuyến tính trên, ta có thể đưa ra được những đánh giá sau dựa vào hệ số đứng trước biến độc lập.

- Những người có điểm thi kỳ 1 và kỳ 2 cao sẽ có xu hướng có điểm thi cuối kỳ cao. Sự tác động này có thể đến từ ý thức học tập.

- Hệ số hồi quy của biến studytime là số âm có thể do phần lớn học sinh dành ra số giờ học trong tuần ở mức 1, 2, 4 cho ra kết quả thấp hơn ở mức 3.
- Những người có số lần không qua môn càng nhiều được thống kê là có điểm thi cuối kỳ có xu hướng thấp hơn so với những người qua môn
- Một điều khá thú vị là hệ số hồi quy của biến số buổi vắng là 1 số dương. Tuy hệ số này là 1 số khá nhỏ nhưng từ đó ta vẫn có thể đưa ra nhận xét rằng, những người có số buổi vắng nhiều lại có xu hướng có điểm thi cuối kì cao hơn đôi chút. Tuy nhiên, điều này chỉ mang tính tương đối và chỉ đúng đối với dữ liệu gốc đang xử lý vì theo dữ liệu, số người đi học chăm chỉ là rất ít, hầu hết đều nghỉ từ 3 đến 15 buổi, mà những học sinh này lại có phân bố điểm số trong khoảng khá rộng (quan sát ở phần trước) nên khi ta xây dựng mô hình, hệ số của biến này có giá trị dương là điều hoàn toàn không quá bất ngờ.

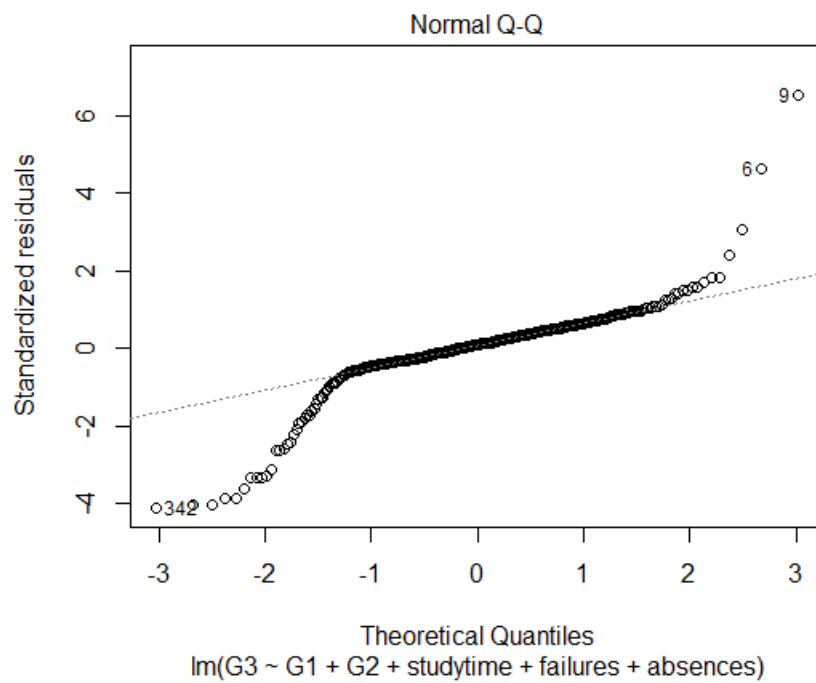
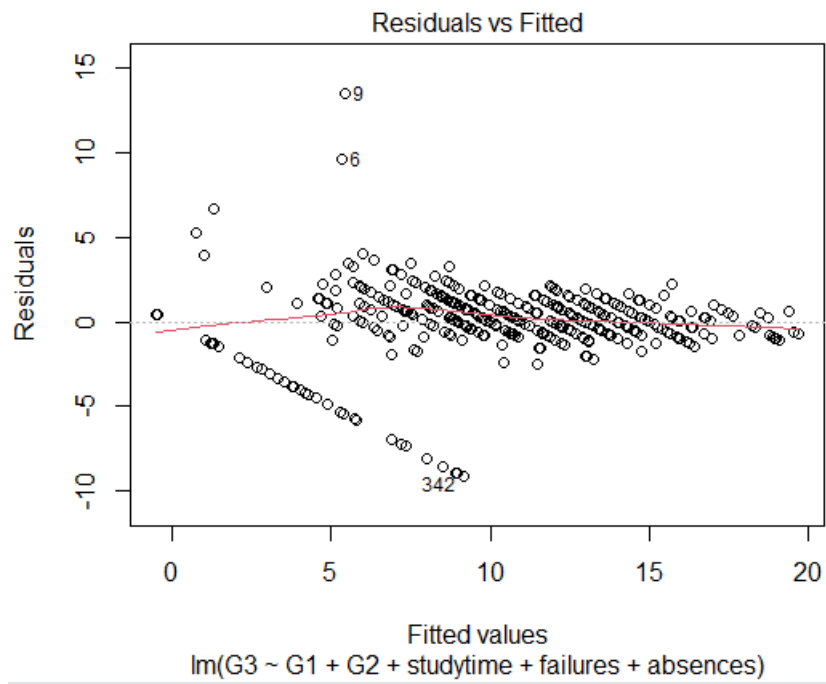
Những đánh giá này đa số trùng khớp khi ta quan sát những đồ thị phân phối đã làm ở trên và ta cũng đã có những suy luận xoay quanh những vấn đề nói trên.

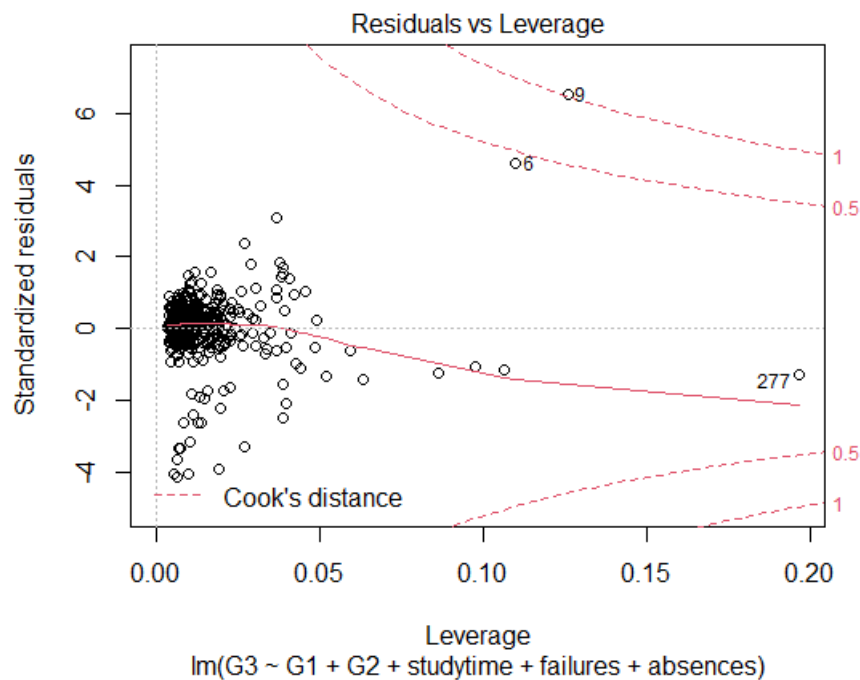
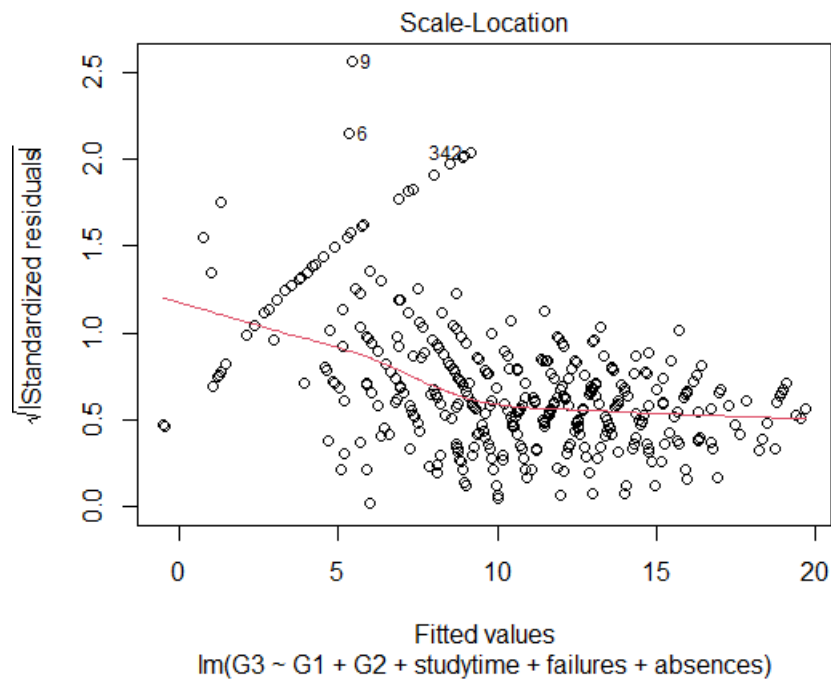
Ta dùng lệnh `plot()` để vẽ đồ thị biểu thị sai số hồi quy và giá trị dự báo.

Ta sử dụng câu lệnh sau:

- `plot(model)` Với model là mô hình tối ưu M3

Sau đó ta nhận return cho tới khi nhận được đồ thị mà ta cần. Đồ thị biểu thị sai số hồi quy và giá trị dự báo sẽ cho ta biết được mối liên hệ giữa giá trị dự báo và sai số hồi quy. Giá trị dự báo G3 này được tính bằng cách thế các giá trị G1, G2, failures, absences, studytime tương ứng vào phương trình hồi quy tuyến tính trên. Sai số hồi quy (residuals) được tính bằng hiệu của giá trị thực tế với giá trị dự đoán. Đồ thị này giúp ta đánh giá được độ chính xác của mô hình.





Hình 26: Đồ thị biểu thị sai số hồi quy và giá trị dự báo

Nhận xét:

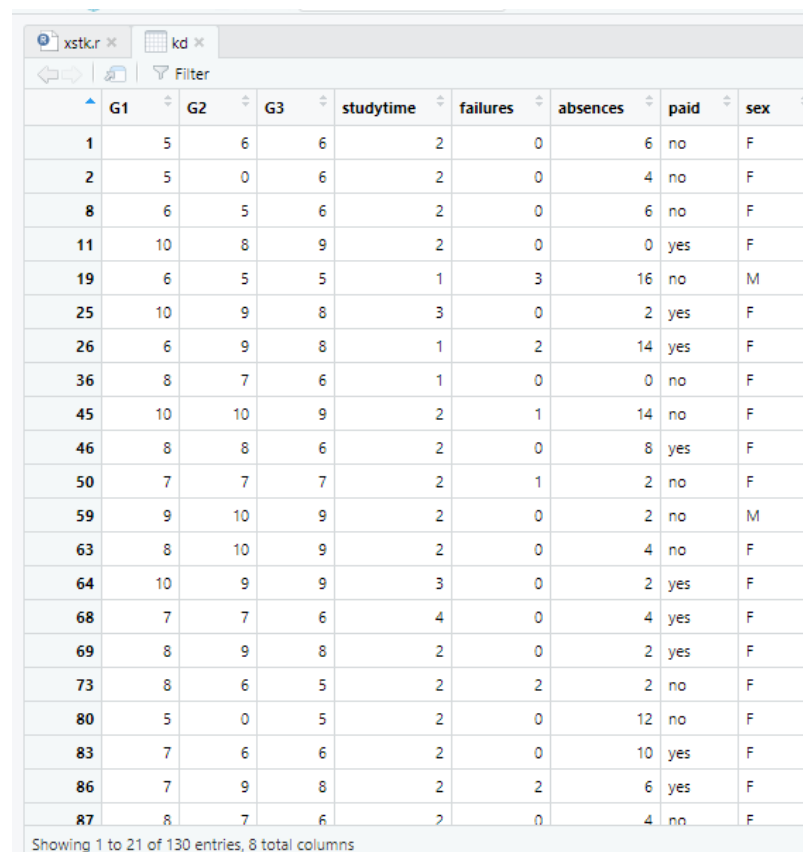
- Những điểm có giá trị residuals dương đại diện cho những bạn học sinh khi đi thi cho ra điểm số cao hơn mô hình dự đoán.

- Những điểm có giá trị residuals âm đại diện cho những bạn học sinh khi đi thi cho ra điểm số thấp hơn mô hình dự đoán.
- Có khá nhiều điểm nằm phía sâu dưới đồ thị, những điểm này đại diện cho những bạn có dữ liệu điểm bị khuyết, bị thay thế bằng điểm 0.
- Mô hình này tương đối chính xác, bởi vì tại những nơi mà dữ liệu phân bố dày đặc, sai số có giá trị khá thấp, đường biểu diễn mối liên hệ trên cũng tương đối thẳng và không lệch quá nhiều so với đường sai số hồi quy $\text{residuals} = 0$.

3.5. Dự đoán

Để đếm tỷ lệ học sinh đạt, không đạt, ta cần lọc ra từ dữ liệu chính 2 dữ liệu con có thông tin của những học sinh trên. Để làm như vậy ta sử dụng câu lệnh `subset()` như sau:

- `kd <- subset(new_DF, G3 < 10)`



	G1	G2	G3	studytime	failures	absences	paid	sex
1	5	6	6	2	0	6	no	F
2	5	0	6	2	0	4	no	F
8	6	5	6	2	0	6	no	F
11	10	8	9	2	0	0	yes	F
19	6	5	5	1	3	16	no	M
25	10	9	8	3	0	2	yes	F
26	6	9	8	1	2	14	yes	F
36	8	7	6	1	0	0	no	F
45	10	10	9	2	1	14	no	F
46	8	8	6	2	0	8	yes	F
50	7	7	7	2	1	2	no	F
59	9	10	9	2	0	2	no	M
63	8	10	9	2	0	4	no	F
64	10	9	9	3	0	2	yes	F
68	7	7	6	4	0	4	yes	F
69	8	9	8	2	0	2	yes	F
73	8	6	5	2	2	2	no	F
80	5	0	5	2	0	12	no	F
83	7	6	6	2	0	10	yes	F
86	7	9	8	2	2	6	yes	F
87	8	7	6	2	0	4	no	F

Showing 1 to 21 of 130 entries, 8 total columns

Hình 27. Bảng dữ liệu chứa thông tin học sinh có điểm thi cuối kỳ không đạt

- `d <- subset(new_DF, G3 >= 10)`

	G1	G2	G3	studytime	failures	absences	paid	sex
3	7	8	10	2	3	10	yes	F
4	15	14	15	3	0	2	yes	F
5	6	10	10	2	0	4	yes	F
6	15	0	15	2	0	10	yes	M
7	12	12	11	2	0	0	no	M
9	16	0	19	2	0	0	yes	M
10	14	15	15	2	0	0	yes	M
12	10	12	12	3	0	4	no	F
13	14	14	14	1	0	2	yes	M
14	10	10	11	2	0	2	yes	M
15	14	16	16	3	0	0	no	M
16	14	14	14	1	0	4	no	F
17	13	14	14	3	0	6	yes	F
18	8	10	10	2	0	4	no	F
20	8	10	10	1	0	4	yes	M
21	13	14	15	2	0	0	no	M
22	12	15	15	1	0	0	yes	M
23	15	15	16	2	0	2	no	M
24	13	13	12	2	0	0	no	M
27	12	12	11	1	0	2	yes	M
28	15	16	15	1	0	4	yes	M

Showing 1 to 21 of 265 entries, 8 total columns

Hình 28: Bảng dữ liệu chứa thông tin học sinh có điểm cuối kì đạt

Sau đó, để biết được số lượng các em học sinh đạt/không đạt, ta chỉ cần đến độ dài của bảng này, để làm như vậy, ta sử dụng hàm `length()` như sau:

- `so_nguoi_dat <- length(kd[, "G3"])`
- `so_nguoi_khong_dat <- length(d[, "G3"])`

Lấy 2 giá trị chia cho 395 (tổng số học sinh) và lưu vào biến `evaluate` bằng lệnh `cbind()`:

- `evaluate <- cbind(so_nguoi_dat/395, so_nguoi_khong_dat/395)`

Biến `evaluate` của ta sau đó sẽ có giá trị như sau:

	V1	V2
1	0.3291139	0.6708861

Hình 29: Biến `evaluate`

Trong đó, V1 là tỷ lệ số học sinh đạt, V2 là tỷ lệ số học sinh không đạt.

Lập một bảng số liệu mới đặt tên là new_X bao gồm toàn bộ các biến độc lập trong mô hình này.

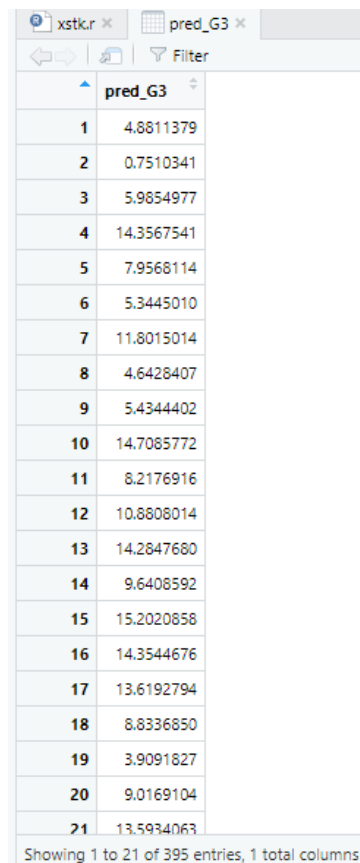
- **new_X <- cbind(G1 = new_DF\$G1, G2 = new_DF\$G2, absences = new_DF\$absences, studytime = new_DF\$studytime, failures = new_DF\$failures)**

Sau đó, ta dùng hàm as.data.frame() để biến new_X thành 1 bảng số liệu để quan sát và dự đoán hơn bằng câu lệnh:

- **new_X <- as.data.frame(new_X)**

Tiếp đến, dùng lệnh predict() để đưa ra số liệu dự báo cho biến G3 phụ thuộc vào new_X. Gọi kết quả dự báo này là biến pred_G3.

- **pred_G3 <- predict(model, new_X)**
- **pred_G3 <- as.data.frame(pred_G3)**



	pred_G3
1	4.8811379
2	0.7510341
3	5.9854977
4	14.3567541
5	7.9568114
6	5.3445010
7	11.8015014
8	4.6428407
9	5.4344402
10	14.7085772
11	8.2176916
12	10.8808014
13	14.2847680
14	9.6408592
15	15.2020858
16	14.3544676
17	13.6192794
18	8.8336850
19	3.9091827
20	9.0169104
21	13.5934063

Hình 31: Biến pred_X chứa giá trị dự đoán G3

So sánh kết quả dự báo `pred_G3` với kết quả thực tế của biến `G3`.

Ta tính toán tỷ lệ đạt / không đạt của biến `G3` dự đoán bằng cách tương tự như biến `G3` thực tế như sau:

- `pkd <- subset(pred_G3, pred_G3 < 10)`
- `pd <- subset(pred_G3, pred_G3 >= 10)`
- `ty_le_du_doan_k_dat <- length(pkd[, "pred_G3"])/395`
- `ty_le_du_doan_dat <- length(pd[, "pred_G3"])/395`

Tiếp đến, ta dùng lệnh `matrix` để tạo bảng so sánh:

- `bang_so_sanh_thuc_te_du_doan <- matrix(nrow = 2, ncol = 2, dimnames = list(c("Quan sat", "Du Bao"), c("Dat", "Khong Dat")))`
- `bang_so_sanh_thuc_te_du_doan[1,1] <- so_nguoi_dat/395`
- `bang_so_sanh_thuc_te_du_doan[1,2] <- so_nguoi_khong_dat/395`
- `bang_so_sanh_thuc_te_du_doan[2,1] <- ty_le_du_doan_dat`
- `bang_so_sanh_thuc_te_du_doan[2,2] <- ty_le_du_doan_k_dat`

Kết quả thu được:

	Dat	Khong Dat
Quan sat	0.3291139	0.6708861
Du Bao	0.5341772	0.4658228

Hình 34: Bảng so sánh tỷ lệ đạt/không đạt giữa thực tế và dự đoán

Qua kết quả, ta có thể thấy kết quả dự đoán khá sát so với thực tế. Ta không thể yêu cầu kết quả dự đoán bằng đúng với thực tế trong trường hợp này được. Vì mô hình hồi quy tuyến tính chúng ta xây dựng đã loại đi 1 số biến có ảnh hưởng ít tới biến cần dự đoán, trong khi dữ liệu mà chúng ta nhập vào để dự đoán lại chính là dữ liệu gốc ta dùng để xây dựng mô hình.

HOẠT ĐỘNG 2 -PHẦN RIÊNG – HIỆU NĂNG CỦA CPU MÁY TÍNH:

1. ĐỀ BÀI:

Tập tin machine.data cung cấp thông tin về tập hợp 209 bộ xử lý trung tâm máy tính (CPU) được 30 nhà cung cấp bán ra thị trường được thu thập và đăng trên trang web <https://archive.ics.uci.edu/> vào năm 1987. Dữ liệu trong tập tin này dùng để phân tích các thuộc tính ngay bên trong CPU có tác động hiệu năng của CPU đó:

- Tổng số CPU được nghiên cứu: 209
- Tổng số biến: 10
- Mô tả các biến chính:
 1. **vendor name:** tên 30 nhà phân phối (adviser, amdahl,apollo, basf, bti, burroughs, c.r.d, cambex, cdc, dec, dg, formation, four-phase, gould, honeywell, hp, ibm, ipl, magnuson, microdata, nas, ncr, nixdorf, perkin-elmer, prime, siemens, sperry, sratus, wang)
 2. **Model Name:** các chuỗi có chứa kí tự đặc biệt thể hiện tên model
 3. **MYCT:** chu kì máy trên nano giây (integer)
 4. **MMIN:** dung lượng tối thiểu của bộ nhớ chính (kilobytes) (integer)
 5. **MMAX:** dung lượng tối đa của bộ nhớ chính (kilobytes) (integer)
 6. **CACH:** dung lượng bộ nhớ đệm (kilobytes) (integer)
 7. **CHMIN:** số đường truyền tối thiểu (đơn vị) (integer)
 8. **CHMAX:** số đường truyền tối đa (đơn vị) (integer)
 9. **PRP:** Hiệu năng được các nhà phát hành công bố (integer)
 10. **ERP:** Hiệu năng qua mô hình các nhà khoa học (integer)

2. THỰC HÀNH VỚI R:

2.1. Nhập dữ liệu:

a. Đọc dữ liệu từ tập tin machine.data

i. Hiện thực lệnh:

```
my_data<-read.csv("machine.csv")
machine<-data.frame(my_data[,c(1,3,4,5,6,7,8,9)])
names(machine) <- c("Vendor
```

```
name","MYCT","MMIN","MMAX","CACH","CHMIN","CHMAX",
"PRP")
```

ii. Kết quả từ lệnh:

	Vendor	Model	MYCT	MMIN	MMAX	CACH	CHMIN	CHMAX	PRP	ERP
1	adviser	32/60	125	256	6000	256	16	128	198	199
2	amdahl	470v/7	29	8000	32000	32	8	32	269	253
3	amdahl	470v/7a	29	8000	32000	32	8	32	220	253
4	amdahl	470v/7b	29	8000	32000	32	8	32	172	253
5	amdahl	470v/7c	29	8000	16000	32	8	16	172	132
6	amdahl	470v/b	26	8000	32000	64	8	32	318	290
7	amdahl	580-5840	23	16000	32000	64	16	32	367	381
8	amdahl	580-5850	23	16000	32000	64	16	32	489	381
9	amdahl	580-5860	23	16000	64000	64	16	32	636	749
10	amdahl	580-5880	23	32000	64000	128	32	64	1144	1238

Data	
machine	209 obs. of 8 variables
my_data	209 obs. of 10 variables

	Vendor name	MYCT	MMIN	MMAX	CACH	CHMIN	CHMAX	PRP
1	adviser	125	256	6000	256	16	128	198
2	amdahl	29	8000	32000	32	8	32	269
3	amdahl	29	8000	32000	32	8	32	220
4	amdahl	29	8000	32000	32	8	32	172
5	amdahl	29	8000	16000	32	8	16	132
6	amdahl	26	8000	32000	64	8	32	318
7	amdahl	23	16000	32000	64	16	32	367
8	amdahl	23	16000	32000	64	16	32	489
9	amdahl	23	16000	64000	64	16	32	636
10	amdahl	23	32000	64000	128	32	64	1144

iii. Ý tưởng: Đọc dữ liệu tập tin machine.data vào data.frame(my_data), đưa các biến cần quan tâm bao gồm Vendor name, MYCT, MMIN, MMAX, CACH, CHMIN, CHMAX, PRP tạo thành một data.frame mới tên machine. Từ đây bắt đầu làm việc trên data.frame này.

2.2. Làm sạch dữ liệu:

a. Làm sạch dữ liệu bị khuyết thiếu:

i. Hiện thực lệnh:

```
which(is.na(machine))
```


ii. Kết quả từ lệnh:

```
> which(is.na(machine))  
integer(0)
```

iii. Kết luận: Vây tập tin machine.data không chứa giá trị khuyết nào.

2.3. Làm rõ dữ liệu

a. Tính các giá trị thống kê của các biến liên tục

i. Hiện thực lệnh:

```
contVariable<-machine  
contVariable$`Vendor name`<-NULL  
summary(contVariable)
```

ii. Kết quả từ lệnh:

```
> contVariable<-machine  
> contVariable$`Vendor name`<-NULL  
> summary(contVariable)  
      MYCT      MMIN      MMAX      CACH      CHMIN      CHMAX      PRP  
Min.   : 17.0   Min.   : 64    Min.   : 64    Min.   : 0.00   Min.   : 0.000   Min.   : 0.00   Min.   : 6.0  
1st Qu.: 50.0   1st Qu.: 768   1st Qu.: 4000  1st Qu.: 0.00   1st Qu.: 1.000   1st Qu.: 5.00   1st Qu.: 27.0  
Median : 110.0  Median : 2000  Median : 8000  Median : 8.00   Median : 2.000   Median : 8.00   Median : 50.0  
Mean   : 203.8  Mean   : 2868  Mean   :11796  Mean   : 25.21  Mean   : 4.699   Mean   : 18.27  Mean   : 105.6  
3rd Qu.: 225.0  3rd Qu.: 4000  3rd Qu.:16000  3rd Qu.: 32.00  3rd Qu.: 6.000   3rd Qu.: 24.00  3rd Qu.: 113.0  
Max.   :1500.0  Max.   :32000  Max.   :64000  Max.   :256.00  Max.   :52.000   Max.   :176.00  Max.   :1150.0  
> view(contVariable)
```

iii. Giải thích lệnh:

- Lệnh summary(): tổng hợp các dữ liệu thống kê có trong data.frame

b. Lập bảng thống kê với các biến phân loại (ở đây là biến Vendor name)

i. Hiện thực lệnh:

```
vendors<-machine$`Vendor name`  
vendorData<-data.frame(table(machine$`Vendor name`))  
names(vendorData)<-c("Vendor name", "Freq")  
Percentage <-  
round(vendorData$Freq/sum(vendorData$Freq)*100,digits=4)  
vendorData$Percentage<-Percentage
```

ii. Kết quả của lệnh:

	Vendor name	Freq	Percentage
1	adviser	1	0.4785
2	amdahl	9	4.3062
3	apollo	2	0.9569
4	basf	2	0.9569
5	bti	2	0.9569
6	burroughs	8	3.8278
7	c.r.d	6	2.8708
8	cambex	5	2.3923
9	cdc	9	4.3062
10	dec	6	2.8708
11	dg	7	3.3493
12	formation	5	2.3923
13	four-phase	1	0.4785
14	gould	3	1.4354
15	harris	7	3.3493
16	honeywell	13	6.2201
17	hp	7	3.3493
18	ibm	32	15.3110
19	ipl	6	2.8708
20	magnuson	6	2.8708
21	microdata	1	0.4785
22	nas	19	9.0909
23	ncr	13	6.2201
24	nixdorf	3	1.4354
25	perkin-elmer	3	1.4354
26	prime	5	2.3923
27	siemens	12	5.7416
28	sperry	13	6.2201
29	sratus	1	0.4785
30	wang	2	0.9569

c. Tính các giá trị thống kê mô tả của biến PRP đối với từng nhà phân phối (Vendor name)

i. Hiện thực lệnh:

```
f <-function(x){c(sample_size=length(x),mean =
round(mean(x),2), sd = round(sd(x),2),
min=min(x),max=max(x),fst_quant=quantile(x)[2],
snd_quant=quantile(x)[3],thrd_quant=quantile(x)[4])
vendorDataStatistic<-do.call(data.frame,aggregate(~`Vendor
name`,data=machine[,c(1,8)],f))
statistic("PRP")
}
```

ii. Kết quả của lệnh:

	Vendor.name	PRP.sample_size	PRP.mean	PRP.sd	PRP.min	PRP.max	PRP.fst_quant.25.	PRP.snd_quant.50.	PRP.thrd_quant.75.
1	adviser	1	198.00	NA	198	198	198.00	198.0	198.00
2	amdahl	9	416.33	315.42	132	1144	220.00	318.0	489.00
3	apollo	2	39.00	1.41	38	40	38.50	39.0	39.50
4	basf	2	115.00	32.53	92	138	103.50	115.0	126.50
5	bti	2	22.50	17.68	10	35	16.25	22.5	28.75
6	burroughs	8	49.75	34.21	19	120	29.50	32.0	64.75
7	c.r.d	6	42.67	23.85	23	77	27.00	30.0	60.00
8	cambex	5	42.80	13.39	26	60	36.00	40.0	52.00
9	cdc	9	130.11	124.46	20	368	32.00	71.0	208.00
10	dec	6	47.33	24.67	18	72	25.00	51.0	69.50
11	dg	7	54.14	41.41	24	138	25.00	36.0	65.50
12	formation	5	16.80	4.15	12	22	14.00	16.0	20.00
13	four-phase	1	36.00	NA	36	36	36.00	36.0	36.00
14	gould	3	182.33	66.40	144	259	144.00	144.0	201.50
15	harris	7	49.71	16.42	36	84	40.00	45.0	51.50
16	honeywell	13	60.46	56.59	16	189	22.00	38.0	66.00
17	hp	7	36.43	18.92	17	64	24.00	32.0	47.00
18	ibm	32	85.09	119.70	6	465	18.00	35.5	82.00
19	ipl	6	69.00	37.74	27	136	47.75	63.0	77.50
20	magnuson	6	39.67	18.34	16	65	27.50	38.5	51.75
21	microdata	1	30.00	NA	30	30	30.00	30.0	30.00
22	nas	19	176.89	152.91	40	510	64.00	105.0	245.50
23	ncr	13	63.31	61.27	8	212	21.00	42.0	100.00
24	nixdorf	3	32.00	8.19	25	41	27.50	30.0	35.50
25	perkin-elmer	3	41.67	14.43	25	50	37.50	50.0	50.00
26	prime	5	53.80	33.08	30	109	32.00	38.0	60.00
27	siemens	12	121.83	117.96	6	405	30.25	94.0	176.75
28	sperry	13	254.92	368.88	12	1150	21.00	70.0	307.00
29	sratus	1	52.00	NA	52	52	52.00	52.0	52.00
30	wang	2	56.00	15.56	45	67	50.50	56.0	61.50

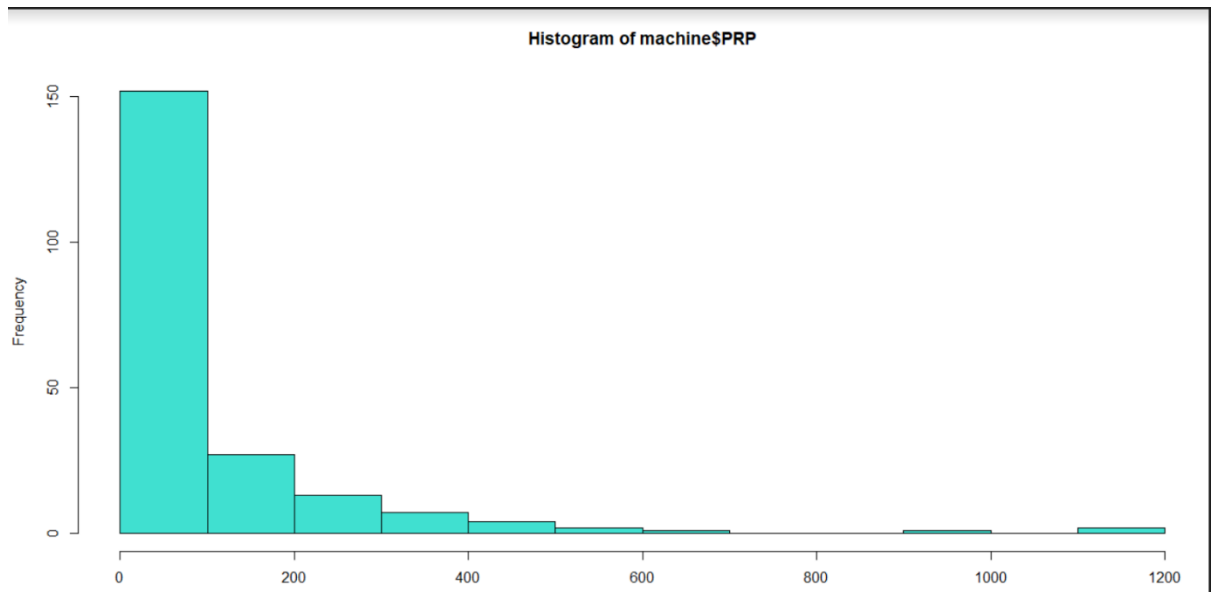
iii. Nhận xét: Do có một số nhà phân phối (Vendor name) chỉ cung cấp duy nhất 1 sản phẩm nên khi đó độ lệch chuẩn của biến PRP theo nhà phân phối đó sẽ là giá trị NA – Not Available.

d. Vẽ đồ thị phân phối cho biến PRP

i. Hiện thực lệnh:

`hist(machine$PRP, col="#55DDE0", border="brown")`

ii. Kết quả của lệnh:

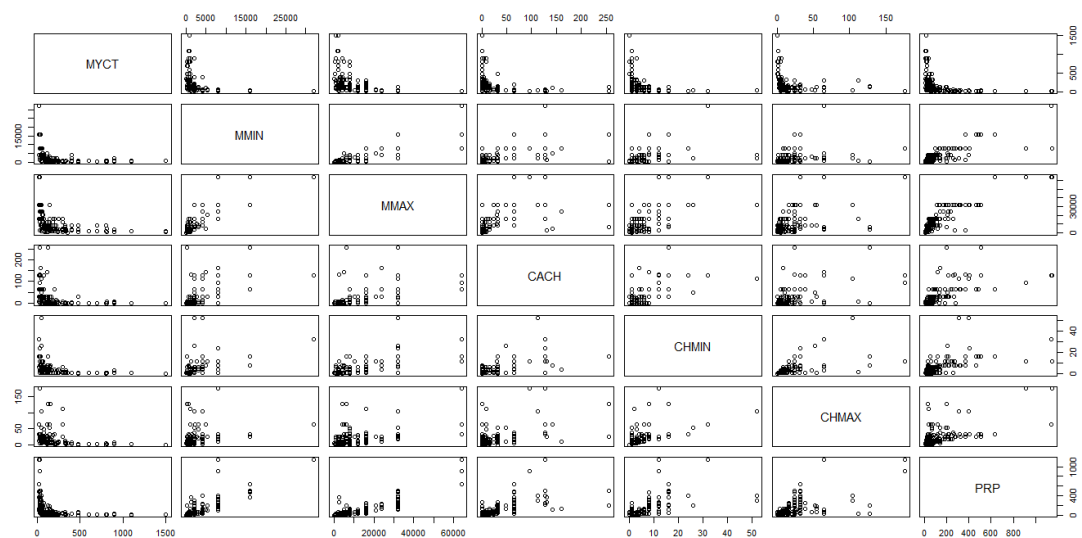


e. Vẽ đồ thị pair biểu thị sự phân bố của biến PRP theo các biến liên tục

i. Thực hiện lệnh:

```
pairs(machine[,2:8])
```

ii. Kết quả của lệnh:

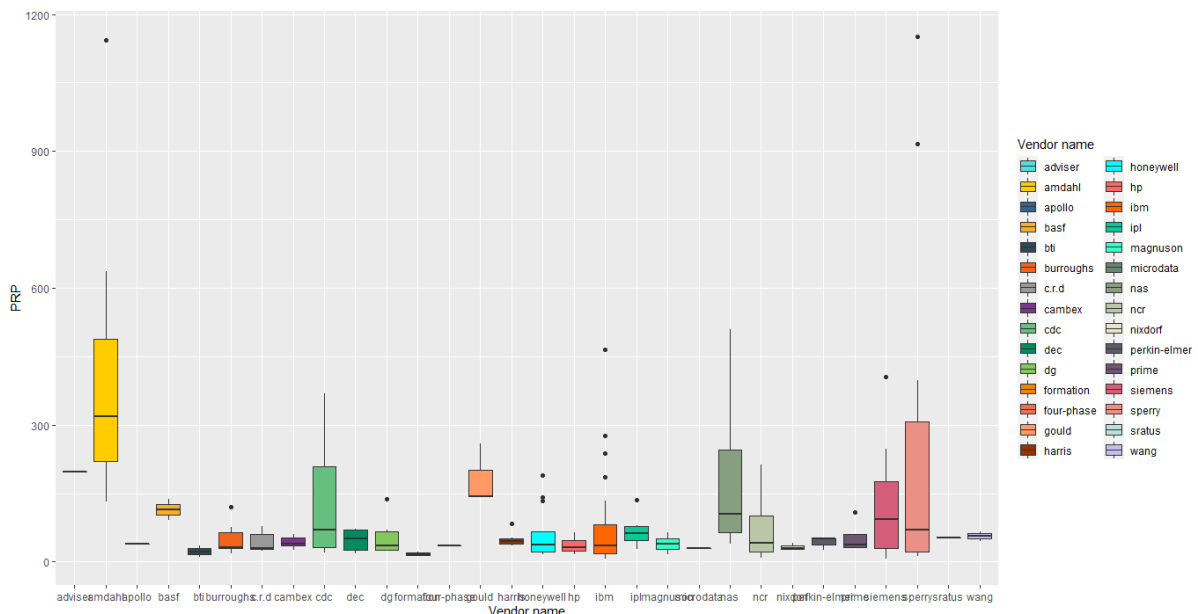


f. Vẽ đồ thị boxplot biểu thị sự phân phối của biến PRP theo các nhà phân phối (Vendor name)

i. Thực hiện lệnh:

```
p<-ggplot(machine, aes(x=`Vendor name`,y=PRP, fill=`Vendor name`))
+geom_boxplot()
p<-p+scale_fill_manual(values=c("#55DDE0", "#FFCC00", "#33658A",
"#F6AE2D", "#2F4858", "#F26419", "#999999", "#79378B",
"#67BF7F", "#008C5E", "#83C75D", "#EC870E", "#EB7153",
"#FF9966", "#993300", "#00FFFF", "#FF6666", "#FF6600",
"#00CC99", "#33FFCC",
"#698474", "#889e81", "#bac7a7", "#e5e4cc", "#5d5b6a",
"#6e5773", "#d45d79", "#ea9085", "#badfdb", "#c3bef0"))
```

ii. Kết quả của lệnh:

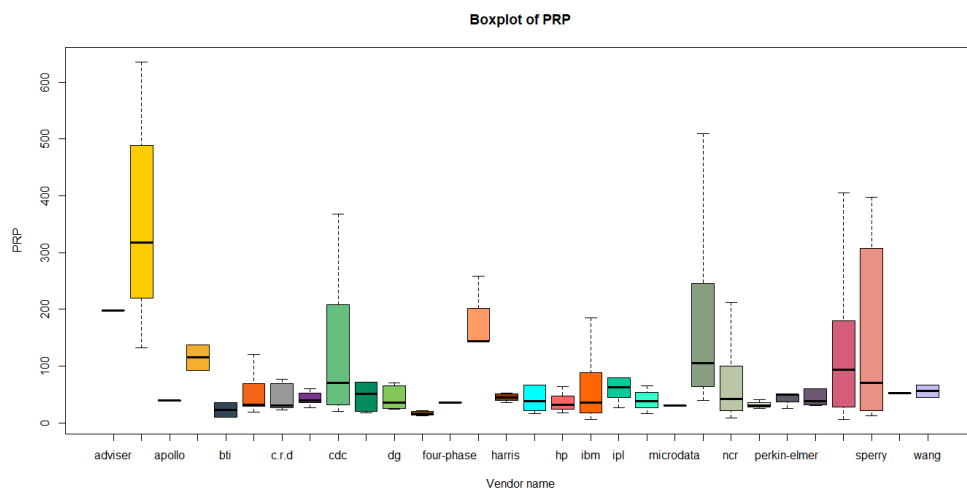


- iii. Nhận xét: Do đồ thị hiển thị các điểm outliers nên khiến đồ thị khó hiển thị rõ hơn. Ta nên loại các điểm này khỏi đồ thị để có thể quan sát rõ hơn.
- g. Vẽ đồ thị boxplot biểu thị sự phân phối của biến PRP theo các nhà phân phối (Vendor name) không có các điểm outliers

i. Thực hiện lệnh:

```
boxplot(PRP~Vendor name`,
data=machine,
main="Boxplot of PRP",
col=c("#55DDE0", "#FFCC00", "#33658A", "#F6AE2D",
"#2F4858", "#F26419", "#999999", "#79378B", "#67BF7F", "#008C5E",
"#83C75D", "#EC870E", "#EB7153", "#FF9966", "#993300",
"#00FFFF", "#FF6666", "#FF6600", "#00CC99", "#33FFCC",
"#698474", "#889e81", "#bac7a7", "#e5e4cc", "#5d5b6a",
"#6e5773", "#d45d79", "#ea9085", "#badfdb", "#c3bef0"),
outline=FALSE)
```

ii. Kết quả của lệnh:



- iii. Nhận xét: Ta thấy từ các đồ thị và dữ liệu thống kê trên, kết hợp với tính chất của bộ dữ liệu thì mô hình dữ liệu phù hợp cho bộ dữ liệu này là mô hình hồi quy tuyến tính.

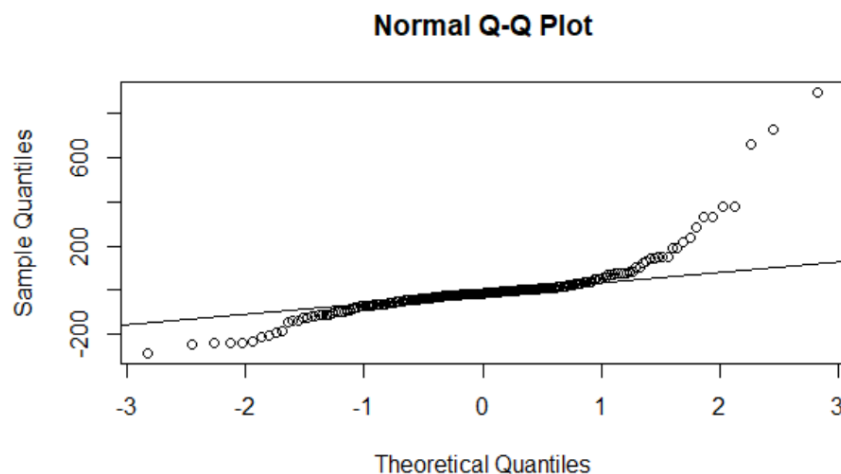
2.4. Anova một nhân tố so sánh về hiệu suất giữa các hãng:

a. Vẽ đồ thị sai số để xác định phân phối chuẩn:

- i. Thực thi lệnh:

```
data<-machine[,c(1,8)]
tbn<- tapply(data$PRP, data$`Vendor name`, mean)
s<-data$PRP - as.numeric(tbn[data$`Vendor name`])
data$s<- s
boxplot(data$s~ data$`Vendor name`)
qqnorm(data$s)
qqline(data$s)
```

- ii. Kết quả từ lệnh:



iii. Nhận xét:

Đồ thị này có các điểm giá trị bám khá sát theo đường thẳng phân vị chuẩn. => Dữ liệu có tính phân phối chuẩn, có thể áp dụng phương pháp anova một nhân tố.

- b. Sử dụng anova một nhân tố:

- i. Thực thi lệnh:

```
M<- aov(data$PRP~ data$`Vendor name`)
anova(M)
```

- ii. Kết quả từ lệnh:

```
Analysis of Variance Table

Response: data$PRP
          Df Sum Sq Mean Sq F value    Pr(>F)    
data$`vendor name`  29 1672791    57682   2.785 1.908e-05 ***
Residuals        179 3707446    20712                     
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> |
```

iii. Nhận xét:

$Pr(>F)$ có giá trị nhỏ hơn 0.5 nên từ đó ta có thể kết luận rằng có sự khác biệt về hiệu suất của CPU đến từ các hãng

2.5. Xây dựng các mô hình hồi quy tuyến tính (Fitting linear regression models):

Chúng ta muốn xác định xem có những nhân tố nào và tác động như thế nào đến hiệu năng của bộ xử lý trung tâm (CPU).

- a. Xét mô hình hồi quy tuyến tính bao gồm biến PRP là một biến phụ thuộc, còn tất cả các biến liên tục còn lại đều là biến độc lập. Thực thi mô hình tuyến tính bội.

i. Thực thi lệnh:

```
machine$MYCT <- log(machine$MYCT)
machine$MMIN <- log(machine$MMIN)
machine$MMAX <- log(machine$MMAX)
model1 <-
lm(PRP~MYCT+MMAX+MMIN+CACH+CHMIN+CHMAX,data
=machine)
summary(model1)
```

ii. Kết quả từ lệnh:

```
> model1 <- lm(PRP~MYCT+MMAX+MMIN+CACH+CHMIN+CHMAX,data=machine)
> summary(model1)
```

```
Call:
lm(formula = PRP ~ MYCT + MMAX + MMIN + CACH + CHMIN + CHMAX,
    data = machine)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-305.43  -37.04   -2.89   33.27  606.44
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -471.6372    116.2434  -4.057 7.09e-05 ***
MYCT          5.7945      9.6811    0.599  0.5502
MMAX         17.8533      9.9031    1.803  0.0729 .
MMIN         42.9775     10.3235    4.163 4.65e-05 ***
CACH          1.0578      0.2175    4.863 2.32e-06 ***
CHMIN         2.9539      1.3263    2.227  0.0270 *
CHMAX         1.8391      0.3189    5.767 2.98e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 94.7 on 202 degrees of freedom
Multiple R-squared:  0.6633,    Adjusted R-squared:  0.6533
F-statistic: 66.32 on 6 and 202 DF,  p-value: < 2.2e-16
```

```
> |
```

- Dựa trên mô hình hồi quy tuyến tính trên, những biến sẽ bị loại khỏi mô hình với mức tin cậy 5% sẽ là biến MYCT, MMAX.
- b. Xét hai mô hình tuyến tính cùng bao gồm biến PRD là biến phụ thuộc nhưng:
 - Mô hình model1 chứa các tất cả các biến liên tục còn lại là biến độc lập.

- Mô hình model2 loại bỏ biến MYCT và MMAX khỏi mô hình model1.

Hãy đề xuất mô hình hợp lý hơn.

i. Thực hiện lệnh:

```
model2<-lm(PRP~MMIN+CACH+CHMIN+CHMAX,data=machine)
anova(model1,model2)
```

ii. Kết quả từ lệnh:

```
> anova(model1,model2)
Analysis of Variance Table

Model 1: PRP ~ MYCT + MMAX + MMIN + CACH + CHMIN + CHMAX
Model 2: PRP ~ MMIN + CACH + CHMIN + CHMAX
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     202 1811601
2     204 1841495 -2     -29893 1.6666 0.1915
> |
```

iii. Nhận xét:

- + Giá trị p-value ($\text{Pr}(>F)$) trong hình trên là khá lớn, vậy mô hình phù hợp sẽ là mô hình ít biến độc lập hơn, và đó là mô hình model2.
- + Các nhân tố MMIN, MYCT, CACH, MMAX, CHMAX tác động đến PRP

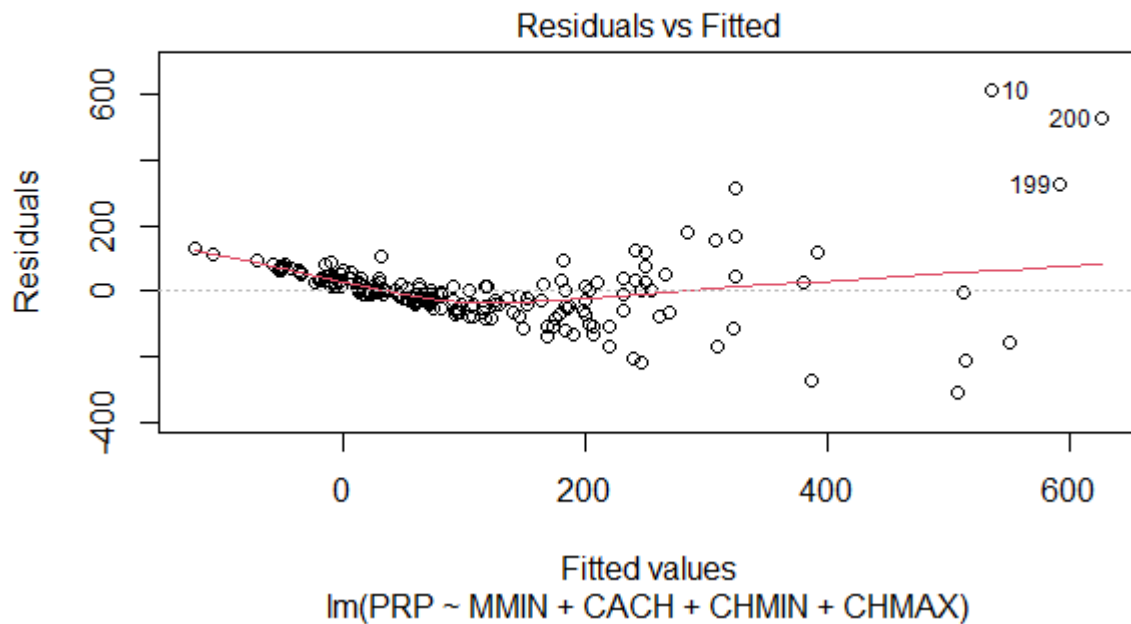
c. Vẽ đồ thị biểu thị sai số hồi quy và giá trị dự báo, đồ thị phân vị chuẩn

i. Thực hiện lệnh:

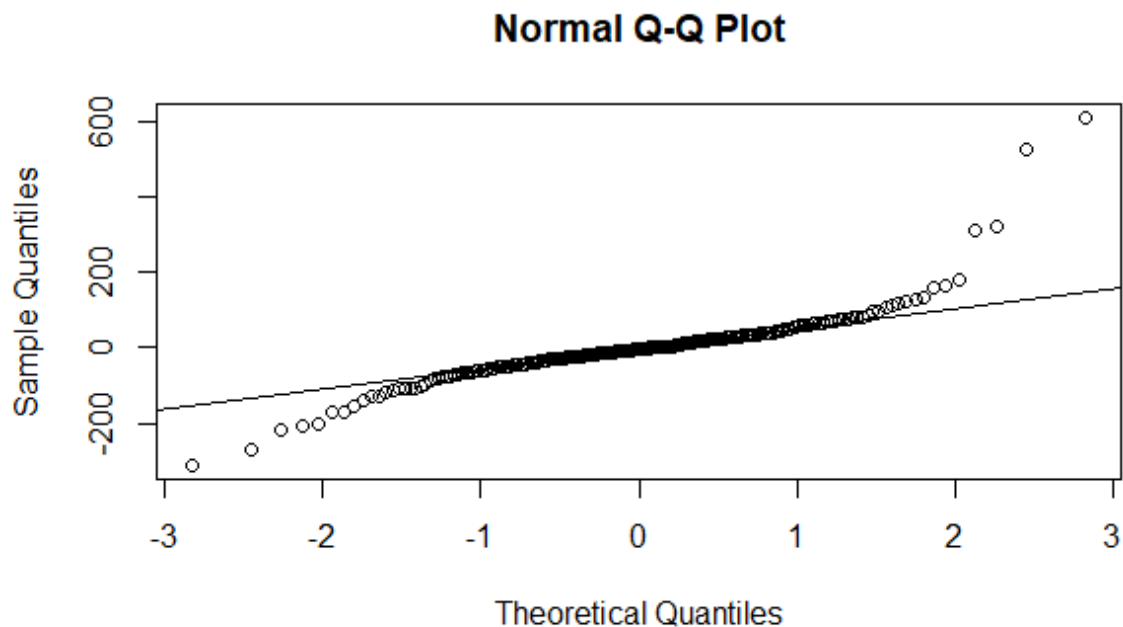
```
plot(model2,which=1)
machine$PRED_PRP<-predict(model2,newdata = machine)
residual<-resid(model2)
qqnorm(residual)
qqline(residual)
```

ii. Kết quả từ lệnh:

- Đồ thị sai số hồi quy và giá trị dự báo:



- Đồ thị phân vị chuẩn



- iii. Nhận xét: Đồ thị sai số có hình dạng khá thô, phân bố ở một số chỗ khá đều, tuy nhiên phần không đều là nhiều, **có dạng hình loa**, và khá không đối xứng qua trục Ox, không bị ảnh hưởng nhiều bởi các điểm rìa. Về đồ thị phân vị chuẩn thì đồ thị này có các điểm giá trị bám khá sát theo đường thẳng phân vị chuẩn.
- iv. Kết luận:
Các kiểm định của mô hình hồi quy tuyến tính chúng ta đưa ra khá không thỏa mãn

d. Kiểm tra lại dựa vào các dự đoán cũng như giá trị thật:

i. Thực thi lệnh:

```
machine$PRED_PRP<-predict(model2,newdata = machine)
compare<-matrix(c(sum(machine$PRP<=100),sum(machine$PRP<=500
&
machine$PRP>100),sum(machine$PRP>500),sum(machine$PRED_PRP
<=100),sum(machine$PRED_PRP<=500 &
machine$PRED_PRP>100),sum(machine$PRED_PRP>500)),ncol=3,by
row=TRUE)
colnames(compare)<-c("Low","Average","High")
rownames(compare)<-c("Observe","Prediction")
compare<-as.table(compare)
compare
```

ii. Kết quả lệnh:

	Vendor name	MYCT	MMIN	MMAX	CACH	CHMIN	CHMAX	PRP	PRED_PRP
1	adviser	4.828314	5.545177	8.699515	256	16	128	198	507.4040990
2	amdahl	3.367296	8.987197	10.373491	32	8	32	269	231.4915372
3	amdahl	3.367296	8.987197	10.373491	32	8	32	220	231.4915372
4	amdahl	3.367296	8.987197	10.373491	32	8	32	172	231.4915372
5	amdahl	3.367296	8.987197	9.680344	32	8	16	132	200.2104903
6	amdahl	3.258097	8.987197	10.373491	64	8	32	318	265.6277552
7	amdahl	3.135494	9.680344	10.373491	64	16	32	367	323.8380763
8	amdahl	3.135494	9.680344	10.373491	64	16	32	489	323.8380763
9	amdahl	3.135494	9.680344	11.066638	64	16	32	636	323.8380763
10	amdahl	3.135494	10.373491	11.066638	128	32	64	1144	536.0705996
11	apollo	5.991465	6.907755	8.006368	0	1	2	38	13.3461971
12	apollo	5.991465	6.238325	8.160518	4	1	6	40	-8.3908376

```

              Low Average High
observe      152      51    6
Prediction  128      74    7
> |
```

iii. Nhận xét:

Có thể thấy được rằng các dự đoán của mô hình 2 vẫn chưa được chính xác, đặc biệt là trong khoảng bé hơn 100 và từ 100 đến 500.

TỔNG KẾT

Với đề tài *“Phân tích điểm toán của các học sinh trung học ở Bồ Đào Nha”* sử dụng ngôn ngữ lập trình R để xử lý dữ liệu thống kê về các yếu tố ảnh hưởng đến kết quả học tập của các nhóm đối tượng, nhóm chúng em đã có cái nhìn trực quan hơn về cách trích xuất dữ liệu từ file excel vào RStudio, cách xử lý từ dữ liệu thô, chất lọc được những dữ liệu có giá trị, thậm chí là có thể khái quát hóa tình hình chung và đưa ra những dự đoán về tập dữ liệu.

Ngoài ra, việc tìm hiểu về ngôn ngữ lập trình R và sử dụng RStudio để ứng dụng vào các bước tính toán phân tích và vẽ đồ thị đã giúp cho chúng em có thêm kỹ năng về lập trình, R là một trong những ngôn ngữ lập trình có nhiều vai trò trong việc hỗ trợ tính toán và giải quyết những vấn đề phức tạp từ tập dữ liệu nhờ có sự trợ giúp của máy tính.

Chắc chắn quá trình thực hiện đề tài sẽ không thể tránh khỏi những thiếu sót. Vì vậy, nhóm mong muốn sẽ nhận được sự góp ý của các thầy cô và bạn bè để đề tài hoàn thiện hơn.

TÀI LIỆU THAM KHẢO

1. Nguyễn Đình Huy (2016). Giáo trình Xác suất và Thống kê. TP Hồ Chí Minh: NXB Đại học Quốc gia TP Hồ Chí Minh.
2. Nguyễn Văn Tuấn. Phân tích số liệu và biểu đồ bằng R. Truy cập từ https://cran.rproject.org/doc/contrib/Intro_to_R_Vietnamese.pdf
3. CPU Performance Data Set, Truy cập từ: <https://archive.ics.uci.edu/ml/index.php>
4. Student Performance Data Set, Truy cập từ: <https://archive.ics.uci.edu/ml/datasets/student+performance>