ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

XÁC SUẤT THỐNG KÊ (MT2013)

BÁO CÁO BÀI TẬP LỚN SỐ 2

GVHD: Nguyễn Bá Thi
SV thực hiện: Nguyễn Ngọc Phú – 2114417
(Nhóm 20 - Khoa Khoa học và Kỹ thuật Máy tính)
Thành viên nhóm 20: Lã Thị Kiều Ngân – 2114149
Bùi Ngọc Toàn – 1912217
Trương Hoàng Nhật – 2114303
Nguyễn Ngọc Phú – 2114417
Nguyễn Hữu Thông – 2114917

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 11/2020

Mục lục

A PHẦN CHUNG (Đề số 1) 5

1 Đọc dữ liệu (Import data) 5
1.1 Yêu cầu . . . 5
1.2 Lời giải R . . . 5
1.3 Kết quả . . . 5

2 Làm sạch dữ liệu (Data cleaning) 6
2.1 Câu a . . . 6
2.1.1 Yêu cầu . . . 6
2.1.2 Lời giải R . . . 6
2.1.3 Kết quả . . . 6
2.2 Câu b . . . 6
2.2.1 Yêu cầu . . . 6
2.2.2 Lời giải R . . . 6
2.2.3 Kết quả . . . 7

3 Làm rõ dữ liệu (Data visualization) 7
3.1 Câu a . . . 7
3.1.1 Yêu cầu . . . 7
3.1.2 Lời giải R . . . 7
3.2 Câu b . . . 8
3.2.1 Yêu cầu . . . 8
3.2.2 Lời giải R . . . 8
3.2.3 Kết quả . . . 8
3.3 Câu c . . . 8
3.3.1 Yêu cầu . . . 8
3.3.2 Lời giải R . . . 9
3.3.3 Kết quả . . . 9
3.4 Câu d . . . 9
3.4.1 Yêu cầu . . . 9
3.4.2 Lời giải R . . . 10
3.4.3 Kết quả . . . 10
3.5 Câu e . . . 10
3.5.1 Yêu cầu . . . 10
3.5.2 Lời giải R . . . 10
3.5.3 Kết quả . . . 11
3.6 Câu f . . . 12
3.6.1 Yêu cầu . . . 12
3.6.2 Lời giải R . . . 12
3.6.3 Kết quả . . . 12

4 Xây dựng các mô hình hồi quy tuyến tính (Fitting linear regression models) 13
4.1 Câu a . . . 13
4.1.1 Yêu cầu . . . 13
4.1.2 Lời giải R . . . 13
4.1.3 Kết quả . . . 14
4.2 Câu b . . . 14
4.2.1 Yêu cầu . . . 14
4.2.2 Lời giải . . . 14
4.3 Câu c . . . 15
4.3.1 Yêu cầu . . . 15
4.3.2 Lời giải R . . . 15
4.3.3 Kết quả . . . 15
4.4 Câu d . . . 16
4.4.1 Yêu cầu . . . 16
4.4.2 Lời giải . . . 16
4.5 Câu e . . . 17
4.5.1 Yêu cầu . . . 17
4.5.2 Lời giải R . . . 17

5 Dự báo (Predictions) 18

B PHẦN CHUNG (ĐỀ SỐ 4) 19

1 Nhập, làm sạch dữ liệu, thực hiện các thống kê mô tả 19
1.1 Câu a . . . 19
1.1.1 Yêu cầu . . . 19
1.1.2 Lời giải R . . . 19
1.1.3 Kết quả . . . 19
1.2 Câu b . . . 19
1.2.1 Yêu cầu . . . 19
1.2.2 Lời giải R . . . 20
1.2.3 Kết quả . . . 20
1.3 Câu c . . . 20
1.3.1 Yêu cầu . . . 20
1.3.2 Lời giải R . . . 20
1.3.3 Kết quả . . . 21
1.4 Câu d . . . 21
1.4.1 Yêu cầu . . . 21
1.4.2 Lời giải R . . . 22
1.4.3 Kết quả . . . 22
1.5 Câu e . . . 22
1.5.1 Yêu cầu . . . 22
1.5.2 Lời giải R . . . 22
1.5.3 Kết quả . . . 22
1.6 Câu f . . . 23
1.6.1 Yêu cầu . . . 23
1.6.2 Lời giải R . . . 23
1.6.3 Kết quả . . . 24

2 Phân tích phương sai một nhân tố **27**
2.1 Câu a . . . 27
2.1.1 Yêu cầu . . . 27
2.1.2 Lời giải . . . 27
2.2 Câu b . . . 27
2.2.1 Yêu cầu . . . 27
2.2.2 Lời giải . . . 27
2.3 Câu c . . . 28
2.3.1 Yêu cầu . . . 28
2.3.2 Lời giải R . . . 28
2.4 Câu d . . . 30
2.4.1 Yêu cầu . . . 30
2.4.2 Lời giải R . . . 30

C PHẦN RIÊNG **35**

1 Đọc dữ liệu (Import data) **35**
1.1 Lời giải R . . . 35
1.2 Kết quả . . . 35

2 Làm sạch dữ liệu (Data cleaning) **36**
2.1 Lọc dữ liệu . . . 36
2.2 Lời giải R . . . 36
2.3 Kết quả . . . 36
2.4 Kiểm tra dữ liệu bị khuyết . . . 36

3 Làm rõ dữ liệu (Data visualization) **37**
3.1 Tính các giá trị thống kê mô tả của biến liên tục . . . 37
3.1.1 Lời giải R . . . 37
3.1.2 Kết quả . . . 37
3.2 Đồ thị phân phối của biến prp . . . 37
3.2.1 Lời giải R . . . 37
3.2.2 Kết quả . . . 37
3.3 Đồ thị phân phối của biến prp theo các biến liên tục myct, mmin, mmax, cach, chmin, chmax . . . 38
3.3.1 Lời giải R . . . 38
3.3.2 Kết quả . . . 38

4 Xây dựng các mô hình hồi quy tuyến tính **40**
4.1 Mô hình gồm prp là biến phụ thuộc, tất cả các biến còn lại là biến độc lập . . . 40
4.1.1 Lời giải R . . . 40
4.1.2 Kết quả . . . 40
4.2 Đề xuất mô hình hồi quy tuyến tính hợp lý . . . 41
4.2.1 Lời giải R . . . 41
4.2.2 Kết quả . . . 41
4.3 Suy luận sự tác động của các biến đến hiệu năng CPU . . . 42
4.4 Đồ thị biểu diễn sai số hồi quy và giá trị dự báo . . . 42
4.4.1 Lời giải R . . . 42

4.4.2 Kết quả 43
4.4.3 Nhận xét 43

5 Dự báo **43**
5.1 Lời giải R 43
5.2 Nhận xét 44

Phần A
PHẦN CHUNG (Đề số 1)

Tập tin **"gia_nha.csv"** chứa thông tin về giá bán ra thị trường (đơn vị đô la) của 21613 ngôi nhà ở quận King nước Mỹ trong khoảng thời gian từ tháng 5/2014 đến 5/2015. Bên cạnh giá nhà, dữ liệu còn bao gồm các thuộc tính mô tả chất lượng ngôi nhà.
Các biến chính trong bộ dữ liệu:

- **price**: Giá nhà được bán ra.
- **sqft_living15**: Diện tích trung bình của 15 ngôi nhà gần nhất trong khu dân cư.
- **floors**: Số tầng của ngôi nhà được phân loại từ 1-3.5.
- **condition**: Điều kiện kiến trúc của ngôi nhà từ 1 - 5, 1: rất tệ và 5: rất tốt.
- **sqft_above**: Diện tích ngôi nhà.
- **sqft_living**: Diện tích khuôn viên nhà.

1 Đọc dữ liệu (Import data)

1.1 Yêu cầu

Hãy dùng lệnh read.csv() để đọc tệp tin.

1.2 Lời giải R

```
gia_nha <- read.csv("gia_nha.csv")
View(gia_nha)
```

1.3 Kết quả

(row)	X.2	X.1	X	id	date	price	bedrooms	bathrooms	sqft_living
1	1	1	1	7129300520	20141013T000000	221900	3	1	1180
2	2	2	2	6414100192	20141209T000000	538000	3	2.25	2570
3	3	3	3	5631500400	20150225T000000	180000	2	1	770
4	4	4	4	2487200875	20141209T000000	604000	4	3	1960
5	5	5	5	1954400510	20150218T000000	510000	3	2	1680
6	6	6	6	7237550310	20140512T000000	1225000	4	4.5	5420
7	7	7	7	1321400060	20140627T000000	257500	3	2.25	1715
8	8	8	8	2008000270	20150115T000000	291850	3	1.5	1060
9	9	9	9	2414600126	20150415T000000	229500	3	1	1780
10	10	10	10	3793500160	20150312T000000	323000	3	2.5	1890
11	11	11	11	1736800520	20150403T000000	662500	3	2.5	3560
12	12	12	12	9212900260	20140527T000000	468000	2	1	1160
13	13	13	13	114101516	20140528T000000	310000	3	1	1430
14	14	14	14	6054650070	20141007T000000	400000	3	1.75	1370

2 Làm sạch dữ liệu (Data cleaning)

2.1 Câu a

2.1.1 Yêu cầu

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 hỏi này về sau, mọi yêu cầu xử lý đều dựa trên tập dữ liệu con **new_DF** này.

2.1.2 Lời giải R

```
new_DF <- subset(gia_nha, select = c(
"price", "sqft_living15", "floors", "condition", "sqft_above", "sqft_living"))
View(new_DF)
```

2.1.3 Kết quả

(row)	price	sqft_living15	floors	condition	sqft_above	sqft_living
1	221900	1340	1	3	1180	1180
2	538000	1690	2	3	2170	2570
3	180000	2720	1	3	770	770
4	604000	1360	1	5	1050	1960
5	510000	1800	1	3	1680	1680
6	1225000	4760	1	3	3890	5420
7	257500	2238	2	3	1715	1715
8	291850	1650	1	3	1060	1060
9	229500	1780	1	3	1050	1780
10	323000	2390	2	3	1890	1890
11	662500	2210	1	3	1860	3560
12	468000	1330	1	4	860	1160
13	310000	1780	1.5	4	1430	1430
14	400000	1370	1	4	1370	1370
15	530000	1360	1.5	3	1810	1810

2.2 Câu b

2.2.1 Yêu cầu

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 bị khuyết này

2.2.2 Lời giải R

```
apply(is.na(new_DF), 2, sum)
```

```
> apply(is.na(new_DF), 2, sum)
       price sqft_living15        floors     condition    sqft_above
          20             0             0             0             0
 sqft_living
           0
```

2.2.3 Kết quả

Nhận xét:

- Nhận thấy trong bảng new_DF chỉ có 20 dữ liệu bị khuyết, rất bé so với tổng số dữ liệu (21613). Vì thế ta sẽ chọn phương pháp xóa những hàng có dữ liệu bị khuyết bằng lệnh **na.omit()**. Số dữ liệu còn lại trong bảng hiện tại là 21613 - 20 = 21593.
- Lời giải R

```
new_DF <- na.omit(new_DF)
View(new_DF)
```

- Bảng new_DF thu được:

(row)	price	sqft_living15	floors	condition	sqft_above	sqft_living
1	221900	1340	1	3	1180	1180
2	538000	1690	2	3	2170	2570
3	180000	2720	1	3	770	770
4	604000	1360	1	5	1050	1960
5	510000	1800	1	3	1680	1680
6	1225000	4760	1	3	3890	5420
7	257500	2238	2	3	1715	1715
8	291850	1650	1	3	1060	1060
9	229500	1780	1	3	1050	1780
10	323000	2390	2	3	1890	1890
11	662500	2210	1	3	1860	3560
12	468000	1330	1	4	860	1160
13	310000	1780	1.5	4	1430	1430
14	400000	1370	1	4	1370	1370
15	530000	1360	1.5	3	1810	1810

3 Làm rõ dữ liệu (Data visualization)

3.1 Câu a

3.1.1 Yêu cầu

Chuyển đổi các biến price, sqft_living15, sqft_above, sqft_living lần lượt thành log(price), log(sqft_living15), log(sqft_above) và log(sqft_living). Từ đây mọi sự tính toán với các biến trên được hiểu là đã qua đổi biến dạng log.

3.1.2 Lời giải R

```
new_DF$price <- log(new_DF$price)
new_DF$sqft_living15 <- log(new_DF$sqft_living15)
new_DF$sqft_above <- log(new_DF$sqft_above)
new_DF$sqft_living <- log(new_DF$sqft_living)
```

3.2 Câu b

3.2.1 Yêu cầu

Đối với các biến liên tục, hãy tính các giá trị thống kê mô tả bao gồm: trung bình, trung vị, độ lệch chuẩn, giá trị lớn nhất và giá trị nhỏ nhất. Xuất kết quả dưới dạng bảng. (Hàm gợi ý: **mean(), median(), sd(), min(), max(), apply(), as.data.frame(), rownames()**).

3.2.2 Lời giải R

Nhận thấy các biến liên tục gồm **price, sqft_living15, sqft_above, sqft_living**, ta có lời giải R sau:

```
Mean <- apply(new_DF[c( "price", "sqft_living15", "sqft_above", "sqft_living" )], 2, mean)
Median <- apply(new_DF[c( "price", "sqft_living15", "sqft_above", "sqft_living" )], 2, median)
Sd <- apply(new_DF[c( "price", "sqft_living15", "sqft_above", "sqft_living" )], 2, sd)
Max <- apply(new_DF[c( "price", "sqft_living15", "sqft_above", "sqft_living" )], 2, max)
Min <- apply(new_DF[c( "price", "sqft_living15", "sqft_above", "sqft_living" )], 2, min)
cont_var <- cbind(Mean, Median, Sd, Max, Min)
cont_var <- as.data.frame(cont_var, stringsAsFactors = FALSE)
View(cont_var)
```

3.2.3 Kết quả

(row)	Mean	Median	Sd	Max	Min
price	13.0478	13.017	0.5266	15.8567	11.2252
sqft_living15	7.5394	7.5175	0.3275	8.7339	5.989
sqft_above	7.3949	7.3524	0.4276	9.1495	5.6699
sqft_living	7.5503	7.5549	0.4248	9.5134	5.6699

3.3 Câu c

3.3.1 Yêu cầu

Đối với các biến phân loại, hãy lập một bảng thống kê số lượng cho từng chủng loại (Hàm gợi ý: **table()**)

3.3.2 Lời giải R

Nhận thấy các biến phân loại gồm **floors, condition**, ta có Lời giải R sau:

```
Floors_tab <- as.data.frame(table(new_DF$floors))
colnames(Floors_tab = c("Floors", "Frequency"))
View(Floors_tab)
Condition_tab <- as.data.frame(table(new_DF$condition))
colnames(Condition_tab = c("Condition", "Frequency"))
View(Condition_tab)
```

3.3.3 Kết quả

Var1	Freq
1	10672
1.5	1909
2	8230
2.5	161
3	613
3.5	8

Hình 1: *Floors*

Var1	Freq
1	30
2	172
3	14016
4	5677
5	1698

Hình 2: *Condition*

3.4 Câu d

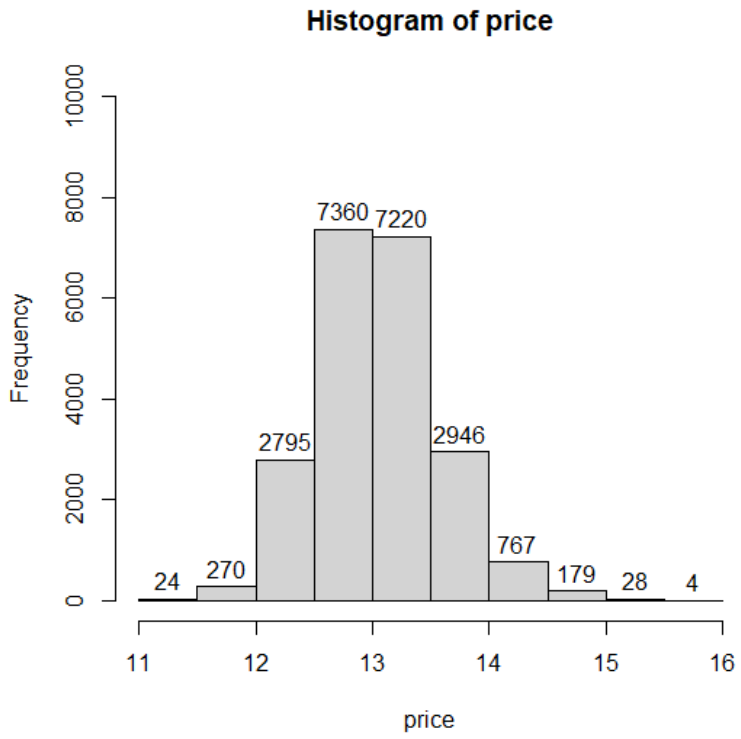
3.4.1 Yêu cầu

Hãy dùng hàm **hist()** để vẽ đồ thị phân phối của biến price.

3.4.2 Lời giải R

```
hist(new_DF$price, xlab = "price", main = "Histogram of price", ylim = c(0, 10000), labels = TRUE )
```

3.4.3 Kết quả



Nhận xét:

- Sau khi chuyển về dạng **log(price)** thì đồ thị phân phối có hình dạng phân phối chuẩn.
- Giá nhà chủ yếu tập trung từ e^{12} đến e^{14} (USD)

3.5 Câu e

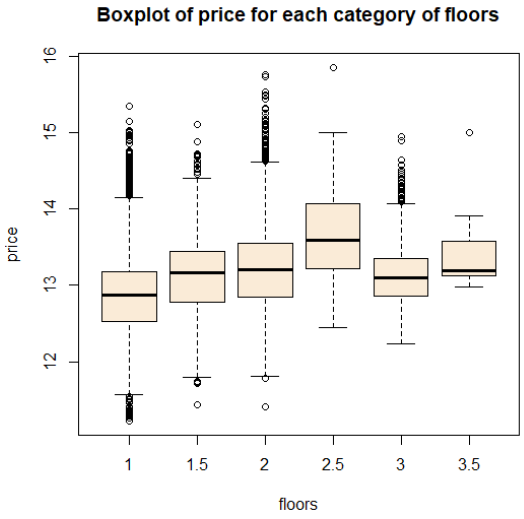
3.5.1 Yêu cầu

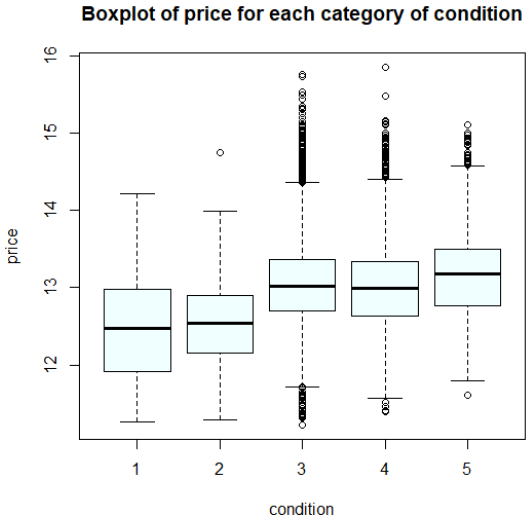
Hãy dùng hàm boxplot() vẽ phân phối của biến price cho từng nhóm phân loại của biến floors và biến condition.

3.5.2 Lời giải R

```
boxplot(price ~ floors, main = "Boxplot of price for each category of floors", data = new_DF, col = "antiquewhite" )
boxplot(price ~ condition, main = "Boxplot of price for each category of condition", data = new_DF, col = "azure1" )
```

3.5.3 Kết quả





Nhận xét: Giá trị trung vị của giá nhà theo số tầng và điều kiện kiến trúc cũng dao động quanh e^{13} (450.000 USD), có khá nhiều điểm ngoại lai, độ phân tán không rộng lắm.

3.6 Câu f

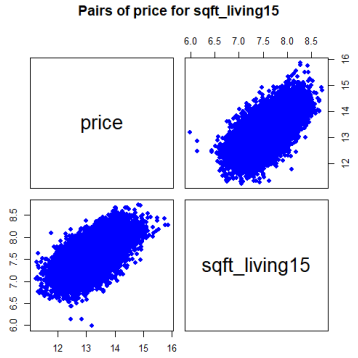
3.6.1 Yêu cầu

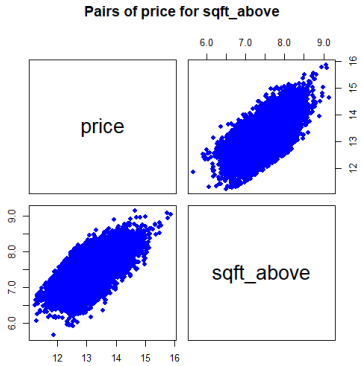
Dùng lệnh pairs() vẽ các phân phối của biến price lần lượt theo các biến sqft_living15, sqft_above và sqft_living.

3.6.2 Lời giải R

```
pairs(price ~ sqft_living15, main = "Pairs of price for sqft_living15", pch = 16, col = "blue", data = new_DF )
pairs(price ~ sqft_above, main = "Pairs of price for sqft_above", pch = 16, col = "blue", data = new_DF )
pairs(price ~ sqft_living, main = "Pairs of price for sqft_living", pch = 16, col = "blue", data = new_DF )
```

3.6.3 Kết quả





Nhận xét:

- Dựa trên đồ thị phân phối của biến price theo các biến sqft_living15, sqft_above và sqft_living ta có thể thấy khi giá trị các biến này tăng thì giá trị biến price cũng tăng.
  $\Rightarrow$ Giữa chúng có thể có tương quan/hồi quy.
- Đồ thị vẫn còn nhiều điểm ngoại lai do biến price chịu ảnh hưởng của nhiều biến khác nhau. $\Rightarrow$ Nếu xét biến price là biến phụ thuộc thì để xác định xem nó phụ thuộc vào những biến nào và phụ thuộc ra sao thì ta cần phải đi xây dựng mô hình hồi quy tuyến tính.

4 Xây dựng các mô hình hồi quy tuyến tính (Fitting linear regression models)

4.1 Câu a

4.1.1 Yêu cầu

Xét mô hình hồi quy tuyến tính bao gồm biến price là một biến phụ thuộc, và tất cả các biến còn lại đều là biến độc lập. Hãy dùng lệnh lm() để thực thi mô hình hồi quy tuyến tính bội.

4.1.2 Lời giải R

Mô hình hồi quy tuyến tính bao gồm:

- Biến phụ thuộc: price.
- Biến độc lập: sqft_living15, sqft_above, sqft_living.
- Biến phân loại: floors, condition.

(Các biến price, sqft_living15, sqft_above, sqft_living đã được chuyển sang dạng log).
Ta sẽ sử dụng hàm lm() để thực thi mô hình hồi quy tuyến tính trên.

```
model1 <- lm(price ~ sqft_living15 + as.factor(floors) + as.factor(condition) + sqft_above +
sqft_living, data = new_DF)
summary(model1)
```

```
Call:
lm(formula = price ~ sqft_living15 + as.factor(floors) + as.factor(condition) +
    sqft_above + sqft_living, data = new_DF)

Residuals:
     Min       1Q   Median       3Q      Max
-1.27088 -0.27135  0.00796  0.24123  1.49961

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)            5.314428   0.093195  57.025  < 2e-16 ***
sqft_living15          0.462153   0.011947  38.684  < 2e-16 ***
as.factor(floors)1.5   0.181243   0.009478  19.122  < 2e-16 ***
as.factor(floors)2     0.067617   0.007348   9.201  < 2e-16 ***
as.factor(floors)2.5   0.373291   0.029843  12.509  < 2e-16 ***
as.factor(floors)3     0.381985   0.015918  23.996  < 2e-16 ***
as.factor(floors)3.5   0.496959   0.130511   3.808 0.000141 ***
as.factor(condition)2  0.018681   0.073005   0.256 0.798040
as.factor(condition)3  0.162216   0.067548   2.401 0.016336 *
as.factor(condition)4  0.208051   0.067621   3.077 0.002095 **
as.factor(condition)5  0.331814   0.068074   4.874  1.1e-06 ***
sqft_above            -0.142451   0.014256  -9.992  < 2e-16 ***
sqft_living            0.670249   0.013105  51.145  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3687 on 21580 degrees of freedom
Multiple R-squared:   0.51,     Adjusted R-squared:  0.5097
F-statistic:  1872 on 12 and 21580 DF,  p-value: < 2.2e-16
```

4.1.3 Kết quả

Nhận xét: Ta nhận thấy hệ số của các biến đều dương, riêng hệ số của biến sqft_above là âm, cho thấy khi diện tích nhà tăng thì giá nhà sẽ giảm, điều này là không hợp lí với thực tế.

4.2 Câu b

4.2.1 Yêu cầu

Dựa vào kết quả của mô hình hồi quy tuyến tính trên, những biến nào bạn sẽ loại khỏi mô hình tương ứng với mức tin cậy 5%

4.2.2 Lời giải

Xét mức tin cậy 5%:

- Sig. $>$ 0.05: Chấp nhận giả thiết H_0, tức là hệ số hồi quy ứng với biến phụ thuộc không có ý nghĩa thống kê, ta sẽ loại biến này ra khỏi mô hình.
- Sig. $<$ 0.05: Chấp nhận giả thiết H_1, tức là hệ số hồi quy ứng với biến phụ thuộc có ý nghĩa thống kê, ta sẽ nhận kết quả biến phụ thuộc.

Kết luận: Dựa vào kết quả của mô hình hồi quy M_1, với mức tin cậy 5%, ta thấy biến **condition2** có giá trị Sig. (cột Pr($>$|t|)) là 0.798040 $>$ 0.05, ta chấp nhận giả thiết H_0, tức là biến **condition2** không có ý nghĩa thống kê. Do đó có thể cân nhắc loại bỏ biến **condition** ra khỏi mô hình.

4.3 Câu c

4.3.1 Yêu cầu

Xét 2 mô hình tuyến tính cùng bao gồm biến **price** là biến phụ thuộc nhưng:

1. mô hình M1 chứa tất cả các biến còn lại là biến độc lập.
2. mô hình M2 là loại bỏ biến **condition** từ mô hình M1.

Hãy dùng lệnh **anova()** để đề xuất mô hình hồi quy hợp lý hơn.

4.3.2 Lời giải R

Mô hình M_1 chính là kết quả của câu (a) đã thực hiện ở trên.
Xây dựng mô hình M_2 bao gồm:

- Biến phụ thuộc: **price**.

- Biến độc lập: **sqft_living15, sqft_above, sqft_living**.

- Biến phân loại: **floors**.

Tương tự trên, sử dụng lệnh **as.factor()** để chuyển biến **floors** thành biến nhân tố và dùng lệnh **lm()** để hiện thực mô hình M_2.

```
model2 <- lm(price ~ sqft_living15 + as.factor(floors) + sqft_above + sqft_living, data = new_DF)
summary(model2)
```

4.3.3 Kết quả

```
Call:
lm(formula = price ~ sqft_living15 + as.factor(floors) + sqft_above +
    sqft_living, data = new_DF)

Residuals:
     Min       1Q   Median       3Q      Max
-1.28543 -0.27361  0.00807  0.24556  1.47598

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          5.507119   0.065361  84.257  < 2e-16 ***
sqft_living15        0.451639   0.012023  37.563  < 2e-16 ***
as.factor(floors)1.5 0.194614   0.009523  20.437  < 2e-16 ***
as.factor(floors)2   0.049024   0.007202   6.807 1.02e-11 ***
as.factor(floors)2.5 0.372550   0.030073  12.388  < 2e-16 ***
as.factor(floors)3   0.355923   0.015858  22.444  < 2e-16 ***
as.factor(floors)3.5 0.488617   0.131556   3.714 0.000204 ***
sqft_above          -0.165249   0.014317 -11.543  < 2e-16 ***
sqft_living          0.703102   0.013091  53.709  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3717 on 21584 degrees of freedom
Multiple R-squared:  0.5019,    Adjusted R-squared:  0.5017
F-statistic:  2718 on 8 and 21584 DF,  p-value: < 2.2e-16
```

So sánh hai mô hình M_1 và M_2 bằng lệnh **anova()**:

```
anova(model1, model2)
```

Kết quả:

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

Model 1: price ~ sqft_living15 + as.factor(floors) + as.factor(condition) +
    sqft_above + sqft_living
Model 2: price ~ sqft_living15 + as.factor(floors) + sqft_above + sqft_living
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1  21580 2933.7
2  21584 2982.2 -4     -48.6 89.376 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4.4 Câu d

4.4.1 Yêu cầu

Chọn mô hình hợp lý hơn từ câu (c) hãy suy luận sự tác động của các biến lên giá nhà.

4.4.2 Lời giải

Xét mức ý nghĩa 5%:

- Giả thiết H_0: Hai mô hình M_1 và M_2 giống nhau.
- Giả thiết H_1: Hai mô hình M_1 và M_2 khác nhau.

Dựa vào kết quả so sánh và những phân tích về hai mô hình M_1 và M_2, ta rút ra được kết luận:

- Kết quả so sánh giữa hai mô hình M_1 và M_2 cho thấy hệ số Pr(>F) $\approx 2.2 * 10^{-16} < 0.05$ nên ta bác bỏ giả thiết H_0, chấp nhận giả thiết H_1, hai mô hình trên là khác nhau.
- Thêm vào đó, dựa trên kết quả phân tích về hai mô hình ta thấy tuy mô hình M_1 có biến **condition2** không có ý nghĩa thống kê đối với mô hình, nhưng số lượng biến có ý nghĩa thống kê của mô hình M_1 nhiều hơn so với mô hình M_2. Ngoài ra hệ số R^2 hiệu chỉnh (Adjusted R-squared) của mô hình M_1 (0.5097) lớn hơn so với mô hình M_2 (0.5017) thể hiện độ phù hợp của mô hình M_1 cao hơn so với mô hình M_2.

  $\Rightarrow$ Mô hình M_1 là mô hình hợp lí hơn và được chọn để phân tích.

  Phân tích sự tác động của các biến lên giá nhà:

- Để đánh giá sự tác động của các biến lên giá nhà, ta quan tâm đến các giá trị p-value (cột Pr(>|t|)) tương ứng.
- Với các biến **sqft_living15, floors1.5, floors2, floors2.5, floors3, sqft_above, sqft_living** (có giá trị p-value $< 2 * 10^{-16}$), biến **floors3.5** (có giá trị p-value khoảng 0.000141) và biến **condition5** (có giá trị p-value khoảng $1.1 * 10^{-6}$) có tác động rất lớn đến giá nhà **price**. Ngoài ra các biến **condition3, condition4** cũng có ảnh hưởng đến giá nhà nhưng ít hơn các biến trên.

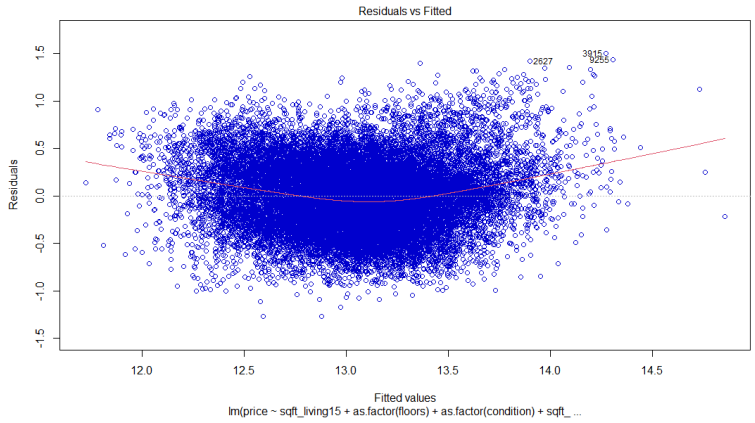
- Biến **condition2** có giá trị p-value khoảng 0.798040 (rất lớn so với 0.05), không có ý nghĩa với mô hình hồi quy nên không có ảnh hưởng nhiều đến giá nhà.
- Các hệ số hồi quy của biến dự báo (cột Estimate) cũng có sự ảnh hưởng kha khá đến biến phụ thuộc khi tăng 1 đơn vị của biến dự báo đó (các biến còn lại không đổi). **Ví dụ:** Hệ số hồi quy của **log(sqft_living)** tương ứng vào khoảng 0.670249, tức là khi log diện tích khuôn viên nhà tăng thêm 1 đơn vị (giả sử các biến còn lại không đổi) ta có thể kì vọng giá trị **log(price)** tăng thêm 0.670249 về mặt đơn vị. Tương tự với các biến còn lại.
- Hệ số R^2 hiệu chỉnh của mô hình có giá trị khoảng 0.5097 nghĩa là khoảng 50,97% sự biến thiên của **log(price)** được giải thích bởi các biến độc lập.

4.5 Câu e

4.5.1 Yêu cầu

Từ mô hình hồi quy mà bạn chọn ở câu (c) hãy dùng lệnh plot() để vẽ đồ thị biểu thị sai số hồi quy (residuals) và giá trị dự báo (fitted values). Nêu ý nghĩa và nhận xét đồ thị.

4.5.2 Lời giải R



Ý nghĩa: Đồ thị này vẽ các giá trị dự báo với các giá trị thặng dư (sai số) tương ứng, dùng để kiểm tra tính tuyến tính của dữ liệu, các sai số có kỳ vọng bằng 0 và tính đồng nhất của các phương sai sai số. Nếu đường màu đỏ trên đồ thị phân tán là đường thẳng nằm ngang mà không phải là đường cong, thì giả định tính tuyến tính của dữ liệu được thỏa mãn. Để kiểm tra giả định phương sai đồng nhất thì các điểm thặng dư phải phân tán đều nhau xung quanh đường thẳng màu đỏ. Để giả định các sai số có kỳ vọng bằng 0 thoả mãn thì đường màu đỏ phải nằm sát đường residuals = 0.

Nhận xét:

+ Đồ thị cho thấy giả định về tính tuyến tính của dữ liệu chưa thực sự thoả mãn.

+ Đồ thị cho ta thấy rằng giả định các sai số có kỳ vọng bằng 0 chưa thoả mãn.

+ Đồ thị cho ta thấy rằng giả định về tính đồng nhất của phương sai chưa thoả mãn.

5 Dự báo (Predictions)

Từ mô hình bạn chọn trong câu (c), hãy dùng lệnh **predict()** để dự báo giá nhà tại 2 thuộc tính như sau:

x1: sqft_living15 = mean(sqft_living15), sqft_above = mean(sqft_above), sqft_living = mean(sqft_living), floor = 2, condition = 3

x2: sqft_living15 = max(sqft_living15), sqft_above = max(sqft_above), sqft_living = max(sqft_living), floor = 2, condition = 3

So sánh khoảng tin cậy cho 2 giá trị dự báo này.

Trước tiên ta sẽ tạo một dataframe cho 2 thuộc tính x1, x2 và dùng lệnh **predict()** để dự báo giá nhà.

```
### Predictions ###
x_pred <- data.frame(sqft_living15 = c(mean(new_DF$sqft_living15), max(new_DF$sqft_living15)),
          sqft_above = c(mean(new_DF$sqft_above), max(new_DF$sqft_above)),
          sqft_living = c(mean(new_DF$sqft_living), max(new_DF$sqft_living)),
          floors = c(2,2),
          condition = c(3,3))
pred <- predict(M1, x_pred, interval = "confidence")
pred <- as.data.frame(pred)
pred$range = pred$upr - pred$lwr
pred
```

Hình 3: *Lệnh **predict()** dự báo giá nhà tại hai thuộc tính x1, x2*

Kết quả:

	fit	lwr	upr	range
1	13.03583	13.02619	13.04547	0.01928189
2	14.65366	14.62920	14.67811	0.04891188

Hình 4: *Kết quả dự báo giá nhà tại hai thuộc tính x1, x2*

Giải thích: Cột **fit** thể hiện kết quả dự đoán của từng thuộc tính, khoảng (**lwr, upr**) là độ dài của khoảng ước lượng giá trị (range). Ta sẽ tiến hành so sánh khoảng ước lượng của hai thuộc tính trên bằng cách lấy tỉ số range giữa hai thuộc tính x2/x1.

```
# Compare range
pred$range[2]/pred$range[1]
```

Hình 5: *So sánh khoảng tin cậy giữa hai thuộc tính*

Kết quả cho thấy tỉ số khoảng ước lượng của thuộc tính x2/x1 vào khoảng 2.536675 tức là khoảng ước lượng của thuộc tính x1 nhỏ hơn 2.536675 lần so với thuộc tính x2.
⇒ Khoảng ước lượng cho giá trị dự báo của thuộc tính x1 hợp lí hơn.

Phần B
PHẦN CHUNG (ĐỀ SỐ 4)

Tập tin **flights.rda** cung cấp thông tin về 162049 chuyến bay đã khởi hành từ hai sân bay lớn của vùng Tây bắc Thái Bình Dương của Mỹ, SEA ở Seattle và PDX ở Portland trong năm 2014. Dữ liệu cung cấp bởi Văn phòng Thống kê Vận tải, Mỹ (https://www.transtats.bts.gov/). Dữ liệu này được dùng để phân tích các nguyên nhân gây ra sự khởi hành trễ hoặc hoãn các chuyến bay. Chi tiết về bộ dữ liệu như sau:

- Tổng chuyến bay được thống kê: 162049.
- Tổng số biến: 16.
- Mô tả các biến chính:
  1. *year, month, day*: ngày khởi hành của mỗi chuyến bay.
  2. *carrier*: tên của hãng hàng không, được mã hóa bằng 2 chữ cái in hoa. Ví dụ: UA = United Airlines, AA = American Airlines, DL = Delta Airlines, v.v
  3. *origin và dest*: tên sân bay đi và đến. Đối với sân bay đi, ta chỉ có hai giá trị SEA (Seattle) và PDX (Portland).
  4. *dep_time và arr_time*: thời gian cất cánh và hạ cánh (theo lịch dự kiến).
  5. *dep_delay và air_time*: chênh lệch (phút) giữa thời gian cất cánh/ hạ cánh thực tế với thời gian cất cánh/ hạ cánh in trong vé.
  6. *distance*: khoảng cách giữa hai sân bay (dặm).

1 Nhập, làm sạch dữ liệu, thực hiện các thống kê mô tả

1.1 Câu a

1.1.1 Yêu cầu

Trong R, hãy sử dụng lệnh *read.table* để đọc dữ liệu từ tập tin *flights.rda*. Chú ý rằng hàng đầu tiên dùng để đặt tên biến và dấu ngăn cách giữa các cột là dấu "," thay vì khoảng trắng như mặc định.

1.1.2 Lời giải R

```
load("flights.rda")
View(flights)
```

1.1.3 Kết quả

1.2 Câu b

1.2.1 Yêu cầu

Hãy tạo một data.frame mới, đặt tên là **newFlights**, chỉ chứa các biến chúng ta cần quan tâm là: carrier, origin, dep_time, arr_time, dep_delay và air_time. Từ câu hỏi này về sau, mọi yêu

(row)	year	month	day	dep_time	dep_delay	arr_time	arr_delay	carrier	tailnum	flight
1	2014	1	1	1	96	235	70	AS	N508AS	145
2	2014	1	1	4	-6	738	-23	US	N195UW	1830
3	2014	1	1	8	13	548	-4	UA	N37422	1609
4	2014	1	1	28	-2	800	-23	US	N547UW	466
5	2014	1	1	34	44	325	43	AS	N762AS	121
6	2014	1	1	37	82	747	88	DL	N806DN	1823
7	2014	1	1	346	227	936	219	UA	N14219	1481
8	2014	1	1	526	-4	1148	15	UA	N813UA	229
9	2014	1	1	527	7	917	24	UA	N75433	1576
10	2014	1	1	536	1	1334	-6	UA	N574UA	478
11	2014	1	1	541	1	911	4	UA	N36476	1569
12	2014	1	1	549	24	907	12	US	N548UW	649
13	2014	1	1	550	0	837	-12	DL	N660DL	1634
14	2014	1	1	557	-3	1134	-16	AA	N3JLAA	1094

cầu xử lý đều được thực hiện trên data.frame **newFlights** này.

1.2.2 Lời giải R

```
newFlights <- subset(flights, select = c(
"carrier", "origin", "dep_time", "arr_time", "dep_delay", "air_time")) View(newFlights)
```

1.2.3 Kết quả

(row)	carrier	origin	dep_time	arr_time	dep_delay	air_time
1	AS	PDX	1	235	96	194
2	US	SEA	4	738	-6	252
3	UA	PDX	8	548	13	201
4	US	PDX	28	800	-2	251
5	AS	SEA	34	325	44	201
6	DL	SEA	37	747	82	224
7	UA	SEA	346	936	227	202
8	UA	PDX	526	1148	-4	217
9	UA	SEA	527	917	7	136
10	UA	SEA	536	1334	1	268
11	UA	PDX	541	911	1	130
12	US	PDX	549	907	24	122
13	DL	SEA	550	837	0	82
14	AA	SEA	557	1134	-3	184
15	AS	SEA	557	825	-3	188

1.3 Câu c

1.3.1 Yêu cầu

Trong các biến đang xét, có một số biến chứa nhiều giá trị khuyết (NA). Hãy in bảng thống kê tỷ lệ giá trị khuyết đối với từng biến. Hãy đề xuất một phương pháp để xử lý những giá trị khuyết này.

1.3.2 Lời giải R

```
apply(is.na(newFlights), 2, sum)
sapply(newFlights, function(col) round((sum(length(which(is.na(col)))) / nrow(newFlights)) * 100.00, 8))
```

1.3.3 Kết quả

```
> apply(is.na(newFlights), 2, sum)
  carrier    origin  dep_time  arr_time dep_delay  air_time
        0         0       857       988       857      1301
> sapply(newFlights, function(col) {
+     round((sum(length(which(is.na(col)))) /
+         nrow(newFlights)) * 100.00, 8)
+ })
  carrier    origin  dep_time  arr_time dep_delay  air_time
0.0000000 0.0000000 0.5288524 0.6096921 0.5288524 0.8028436
```

Ta thấy tỉ lệ số lượng giá trị NA không đáng kể. Tuy nhiên số lượng giá trị NA trong mỗi biến khá lớn, có thể ảnh hưởng đến kết quả phân tích. Do đó, thay vì sử dụng phương pháp xóa, ta sẽ thay những giá trị bị khuyết bằng giá trị trung vị.

```
library("magrittr")
library("tidyverse")
newFlights <- newFlights mutate(dep_time = case_when( is.na(dep_time) ~
as.integer(median(dep_time, na.rm = TRUE)), TRUE   dep_time ))
newFlights <- newFlights mutate(arr_time = case_when( is.na(arr_time) ~
median(arr_time, na.rm = TRUE), TRUE   arr_time ))
newFlights <- newFlights mutate(dep_delay = case_when( is.na(dep_delay) ~
median(dep_delay, na.rm = TRUE), TRUE   dep_delay ))
newFlights <- newFlights mutate(air_time = case_when( is.na(air_time) ~
median(air_time, na.rm = TRUE), TRUE   air_time ))
apply(is.na(newFlights), 2, sum)
```

Sau khi đã thay thế, ta kiểm tra lại số lượng giá trị bị khuyết bằng lệnh **apply()**.

```
apply(is.na(newFlights), 2, sum)
carrier    origin  dep_time  arr_time dep_delay  air_time
      0         0         0         0         0         0
```

1.4 Câu d

1.4.1 Yêu cầu

Tính các giá trị thống kê mô tả (cỡ mẫu, trung bình, độ lệch chuẩn, min, max, các điểm tứ phân vị) của thời gian khởi hành trễ (biến dep_delay) của từng hãng hàng không (carrier). Xuất kết quả ra dưới dạng bảng.

1.4.2 Lời giải R

```
columns <- c("mean", "sd", "min", "max", "q25", "q50", "q75")
temp_1d <- cbind(
tapply(newFlights$dep_delay, newFlights$carrier, mean),
tapply(newFlights$dep_delay, newFlights$carrier, sd),
tapply(newFlights$dep_delay, newFlights$carrier, min),
tapply(newFlights$dep_delay, newFlights$carrier, max),
tapply(newFlights$dep_delay, newFlights$carrier, quantile, probs = 0.25),
tapply(newFlights$dep_delay, newFlights$carrier, quantile, probs = 0.5),
tapply(newFlights$dep_delay, newFlights$carrier, quantile, probs = 0.75)
)
colnames(temp_1d) <- columns
View(temp_1d)
```

1.4.3 Kết quả

(row)	mean	median	sd	min	max	q25	q50	q75
AA	7586	10.4847	51.7606	-18	1553	-5	-2	7
AS	62460	2.7763	20.4325	-25	866	-5	-2	2
B6	3540	8.3446	31.4484	-20	365	-6	-2	8
DL	16716	4.8057	29.342	-19	886	-4	-2	4
F9	2698	10.1119	40.9273	-20	815	-6	-2	11
HA	1095	2.5763	47.1759	-17	878	-7	-4	-1
OO	18710	4.3362	28.5554	-37	677	-6	-4	0
UA	16671	9.6713	33.6018	-19	580	-5	-1	8
US	5946	2.6911	25.8819	-26	711	-6	-3	1
VX	3272	7.8371	32.832	-21	358	-5	-2	3
WN	23355	13.2773	30.2354	-11	712	-2	2	17

1.5 Câu e

1.5.1 Yêu cầu

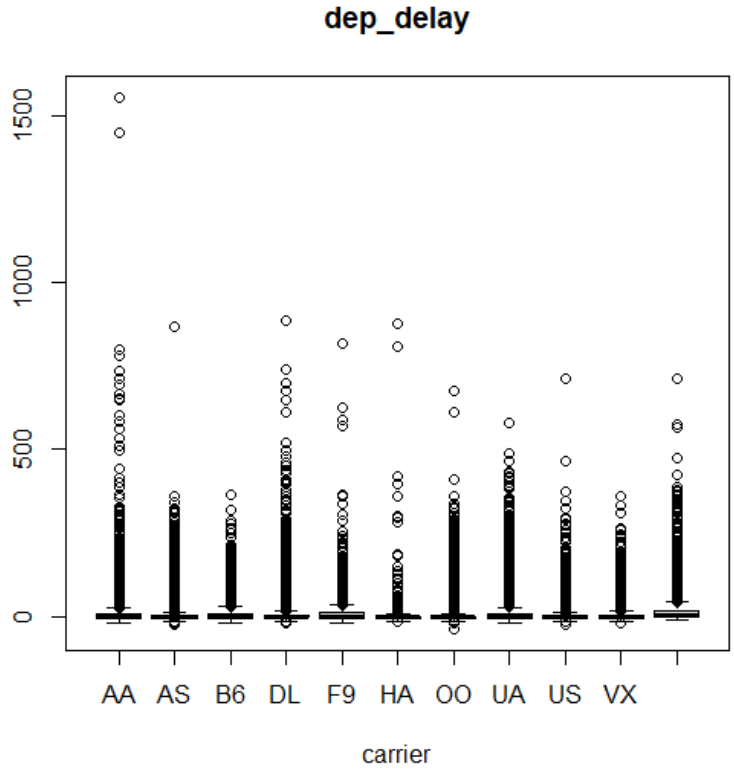
Vẽ đồ thị boxplot cho thời gian khởi hành trễ dep_delay tương ứng với từng hãng hàng không carrier.

1.5.2 Lời giải R

```
boxplot(newFlights$dep_delay ~ newFlights$carrier, data = newFlights,
main = "dep_delay", xlab = "carrier", ylab = "")
```

1.5.3 Kết quả

Nhận xét: Có rất nhiều điểm outliers trên đồ thị.



1.6 Câu f

1.6.1 Yêu cầu

Ta sẽ quan sát thấy rằng có rất nhiều điểm outliers trên các đồ thị boxplot vừa vẽ (đối với biến dep_delay). Hãy sử dụng khoảng tứ phân vị (interquartile range) để loại bỏ các điểm outlier này và vẽ lại các đồ thị boxplot cho dep_delay. Dựa trên đồ thị boxplot, cho nhận xét về thời gian khởi hành trễ của từng hãng hàng không.

1.6.2 Lời giải R

Đầu tiên ta lọc ra dữ liệu con từ bảng **newFlights** theo từng hãng hàng không:

```
library("sqldf")
d1 <- sqldf("select * from newFlights where carrier = 'AA'")
d2 <- sqldf("select * from newFlights where carrier = 'AS'")
d3 <- sqldf("select * from newFlights where carrier = 'B6'")
d4 <- sqldf("select * from newFlights where carrier = 'DL'")
d5 <- sqldf("select * from newFlights where carrier = 'F9'")
d6 <- sqldf("select * from newFlights where carrier = 'HA'")
d7 <- sqldf("select * from newFlights where carrier = 'OO'")
d8 <- sqldf("select * from newFlights where carrier = 'UA'")
```

```
d9 <- sqldf("select * from newFlights where carrier = 'US'")
d10 <- sqldf("select * from newFlights where carrier = 'VX'")
d11 <- sqldf("select * from newFlights where carrier = 'WN'")
```

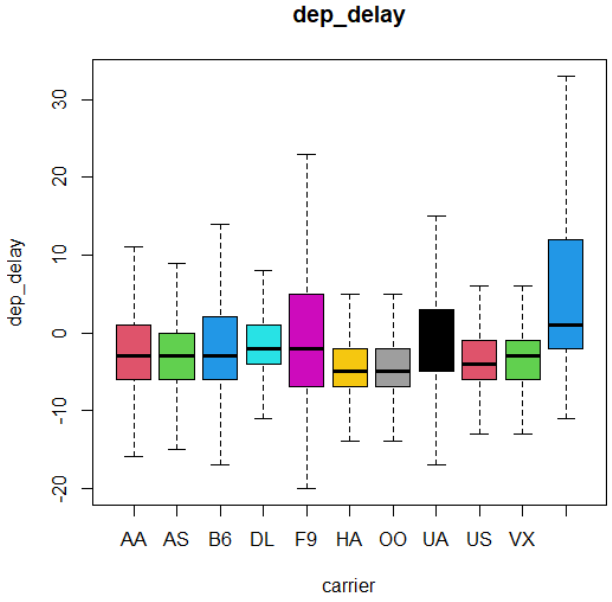
Ở từng phần dữ liệu con này, ta loại bỏ các biến dep_delay ngoại lai đối với từng hãng hàng không, sau đó dùng lệnh rbind để tạo bảng **New_Data** không gồm các giá trị ngoại lai:

```
new_Data <- as.data.frame(rbind(
d1[-which(d1$dep_delay %in% boxplot(d1$dep_delay, plot = FALSE)$out), ],
d2[-which(d2$dep_delay %in% boxplot(d2$dep_delay, plot = FALSE)$out), ],
d3[-which(d3$dep_delay %in% boxplot(d3$dep_delay, plot = FALSE)$out), ],
d4[-which(d4$dep_delay %in% boxplot(d4$dep_delay, plot = FALSE)$out), ],
d5[-which(d5$dep_delay %in% boxplot(d5$dep_delay, plot = FALSE)$out), ],
d6[-which(d6$dep_delay %in% boxplot(d6$dep_delay, plot = FALSE)$out), ],
d7[-which(d7$dep_delay %in% boxplot(d7$dep_delay, plot = FALSE)$out), ],
d8[-which(d8$dep_delay %in% boxplot(d8$dep_delay, plot = FALSE)$out), ],
d9[-which(d9$dep_delay %in% boxplot(d9$dep_delay, plot = FALSE)$out), ],
d10[-which(d10$dep_delay %in% boxplot(d10$dep_delay, plot = FALSE)$out), ],
d11[-which(d11$dep_delay %in% boxplot(d11$dep_delay, plot = FALSE)$out), ]
))
```

Vẽ biểu đồ:

```
boxplot(new_Data$dep_delay ~ new_Data$carrier, xlab = "carrier",
ylab = "dep_delay", main = "dep_delay", outline=FALSE, col=c(2,3,4,5,6,7,8,9,10,11,12))
```

1.6.3 Kết quả



Nhận xét:

- Đối với hãng hàng không AA:
  - Min = - 15 : Thời gian khởi hành sớm nhất là 15 phút.
  - Max = 9 : Thời gian khởi hành trễ nhất là 9 phút.
  - Q1 = -6 : 25% chuyến bay có thời gian khởi hành sớm hơn 6 phút.
  - Q2 = -3 : 50% chuyến bay có thời gian khởi hành sớm hơn 3 phút.
  - Q3 = 0 : 75% chuyến bay có thời gian khởi hành sớm hơn thời điểm dự kiến
- Đối với hãng hàng không AS:
  - Min = -12 : Thời gian khởi hành sớm nhất là 12 phút.
  - Max = 7 : Thời gian khởi hành trễ nhất là 7 phút.
  - Q1 = -5 : 25% chuyến bay có thời gian khởi hành sớm hơn 5 phút.
  - Q2 = -3 : 50% chuyến bay có thời gian khởi hành sớm hơn 3 phút.
  - Q3 = 0 : 75% chuyến bay có thời gian khởi hành sớm hơn thời điểm dự kiến
- Đối với hãng hàng không B6:
  - Min = - 19 : Thời gian khởi hành sớm nhất là 19 phút.
  - Max = 13 : Thời gian khởi hành trễ nhất là 13 phút.
  - Q1 = -7 : 25% chuyến bay có thời gian khởi hành sớm hơn 7 phút.
  - Q2 = -3 : 50% chuyến bay có thời gian khởi hành sớm hơn 3 phút.
  - Q3 = 1 : 75% chuyến bay có thời gian khởi hành trễ hơn 1 phút
- Đối với hãng hàng không DL:
  - Min = - 11 : Thời gian khởi hành sớm nhất là 11 phút.
  - Max = 8 : Thời gian khởi hành trễ nhất là 8 phút.
  - Q1 = -4 : 25% chuyến bay có thời gian khởi hành sớm hơn 4 phút.
  - Q2 = -2 : 50% chuyến bay có thời gian khởi hành sớm hơn 2 phút.
  - Q3 = 1 : 75% chuyến bay có thời gian khởi hành trễ hơn 1 phút
- Đối với hãng hàng không F9:
  - Min = - 19 : Thời gian khởi hành sớm nhất là 19 phút.
  - Max = 15 : Thời gian khởi hành trễ nhất là 15 phút.
  - Q1 = -7 : 25% chuyến bay có thời gian khởi hành sớm hơn 7 phút.
  - Q2 = -3 : 50% chuyến bay có thời gian khởi hành sớm hơn 3 phút.
  - Q3 = 2 : 75% chuyến bay có thời gian khởi hành trễ hơn 2 phút.
- Đối với hãng hàng không HA:
  - Min = - 14 : Thời gian khởi hành sớm nhất là 14 phút.
  - Max = 5 : Thời gian khởi hành trễ nhất là 5 phút.
  - Q1 = -7 : 25% chuyến bay có thời gian khởi hành sớm hơn 7 phút.

  - Q2 = -4 : 50% chuyến bay có thời gian khởi hành sớm hơn 4 phút.
  - Q3 = -2 : 75% chuyến bay có thời gian khởi hành sớm hơn 2 phút.

- Đối với hãng hàng không OO:
  - Min = - 16 : Thời gian khởi hành sớm nhất là 16 phút.
  - Max = 8 : Thời gian khởi hành trễ nhất là 8 phút.
  - Q1 = -7 : 25% chuyến bay có thời gian khởi hành sớm hơn 7 phút.
  - Q2 = -4 : 50% chuyến bay có thời gian khởi hành sớm hơn 4 phút.
  - Q3 = -1 : 75% chuyến bay có thời gian khởi hành sớm hơn 1 phút.
- Đối với hãng hàng không UA:
  - Min = - 15 : Thời gian khởi hành sớm nhất là 15 phút.
  - Max = 13 : Thời gian khởi hành trễ nhất là 13 phút.
  - Q1 = -5 : 25% chuyến bay có thời gian khởi hành sớm hơn 5 phút.
  - Q2 = -2 : 50% chuyến bay có thời gian khởi hành sớm hơn 2 phút.
  - Q3 = 2 : 75% chuyến bay có thời gian khởi hành trễ hơn 2 phút.
- Đối với hãng hàng không US:
  - Min = - 15 : Thời gian khởi hành sớm nhất là 15 phút.
  - Max = 9 : Thời gian khởi hành trễ nhất là 9 phút.
  - Q1 = -6 : 25% chuyến bay có thời gian khởi hành sớm hơn 6 phút.
  - Q2 = -3 : 50% chuyến bay có thời gian khởi hành sớm hơn 3 phút.
  - Q3 = 0 : 75% chuyến bay có thời gian khởi hành sớm hơn thời điểm dự kiến
- Đối với hãng hàng không VX:
  - Min = - 15 : Thời gian khởi hành sớm nhất là 15 phút.
  - Max = 9 : Thời gian khởi hành trễ nhất là 9 phút.
  - Q1 = -6 : 25% chuyến bay có thời gian khởi hành sớm hơn 6 phút.
  - Q2 = -3 : 50% chuyến bay có thời gian khởi hành sớm hơn 3 phút.
  - Q3 = 0 : 75% chuyến bay có thời gian khởi hành sớm hơn thời điểm dự kiến
- Đối với hãng hàng không WN:
  - Min = - 11 : Thời gian khởi hành sớm nhất là 11 phút.
  - Max = 18 : Thời gian khởi hành trễ nhất là 18 phút.
  - Q1 = -2 : 25% chuyến bay có thời gian khởi hành sớm hơn 2 phút.
  - Q2 = 0 : 50% chuyến bay có thời gian khởi hành sớm hơn thời điểm dự kiến.
  - Q3 = 6 : 75% chuyến bay có thời gian khởi hành trễ hơn 6 phút

⇒ Kết luận: Các chuyến bay của hãng hàng không WN có độ trễ lớn nhất

2 Phân tích phương sai một nhân tố

Ta quan tâm đến việc kiểm định rằng liệu có sự khác biệt về thời gian khởi hành trễ trung bình giữa các hãng hàng không đối với các chuyến bay khởi hành từ Portland trong năm 2014 hay không?

2.1 Câu a

2.1.1 Yêu cầu

Hãy giải thích tại sao ta cần dùng phân tích phương sai để trả lời cho câu hỏi trên. Xác định biến phụ thuộc và các nhân tố (hay các biến độc lập).

2.1.2 Lời giải

Chúng ta cần dùng phân tích phương sai để trả lời câu hỏi trên bởi vì mục tiêu của phân tích phương sai là so sánh trung bình của nhiều nhóm. Phân tích phương sai thống kê để kiểm tra sự khác biệt về giá trị trung bình của 3 nhóm trở lên. Biến phụ thuộc là thời gian trễ trung bình của chuyến bay khởi hành từ Portland 2014 (dep_delay). Biến nhân tố là các hãng hàng không khác nhau (carrier).

2.2 Câu b

2.2.1 Yêu cầu

Phát biểu các giả thuyết và đối thuyết bằng lời và công thức toán. Nêu các giả định cần kiểm tra của mô hình.

2.2.2 Lời giải

- Giả thuyết H_0: Gọi $u_1, u_2, u_3, ..., u_k$ là các yếu tố ảnh hưởng đến các chuyến bay khởi hành từ Portland trong năm 2014. Ta có giả thuyết :

  $$H_0 : u_1 = u_2 = u_3 = ... = u_k$$

  Việc giờ bay trung bình của giữa các hãng hàng không đối với các chuyến bay khởi hành từ Portland 2014 là bằng nhau.

- Đối thuyết H_1:

  $$H_1 : \exists u_i \neq u_j \ (i \neq j)$$

  Có ít nhất 2 hãng hàng không đối với các chuyến bay khởi hành từ Portland năm 2014 có việc lệch giờ bay trung bình khác nhau.

- Các giả định cần kiểm tra trong ANOVA một nhân tố:
  - Tổng thể có phân phối chuẩn.
  - Tổng thể có phương sai bằng nhau.
  - Mẫu được chọn ngẫu nhiên và độc lập.

Bảng tổng quát phân tích ANOVA

Biến thiên	Tổng độ lệch bình phương	Bậc tự do	Phương sai	Giá trị kiểm định
Giữa các nhóm	SSG	k-1	$MSG = \frac{SSG}{k-1}$	$F = \frac{MSG}{MSW}$
Trong nội bộ nhóm	SSW	n-k	$MSW = \frac{SSW}{n-k}$	
Tổng cộng	SST	n-1		

Hình 6: *Bảng ANOVA 1 nhân tố*

2.3 Câu c

2.3.1 Yêu cầu

Thực hiện kiểm tra các giả định của mô hình (giả định về phân phối chuẩn, tính đồng nhất của các phương sai).

2.3.2 Lời giải R

Trước tiên ta sẽ tạo một data.frame **delay** chứa thông tin về thời gian khởi hành trễ của từng hãng hàng không khởi hành từ Portland trong năm 2014.

```
delay <- subset(newFlights[c(1, 5)], newFlights$origin == "PDX")
delay <- delay[order(delay$carrier), ]
delay <- as.data.frame(c(delay))
View(delay)
```

Kết quả:

(row)	carrier	dep_delay
1	AA	-2
2	AA	-4
3	AA	14
4	AA	21
5	AA	57
6	AA	10
7	AA	-6
8	AA	41
9	AA	29
10	AA	19
11	AA	87
12	AA	-2
13	AA	-3
14	AA	-1
15	AA	-7

- **Kiểm tra giả định về phân phối chuẩn:**
  - Sử dụng kiểm định Anderson - Darling:
    Để hiện thực được kiểm định này ta phải import thư viện *nortest*.

```
library("nortest")
ad.test(delay$dep_delay)
```

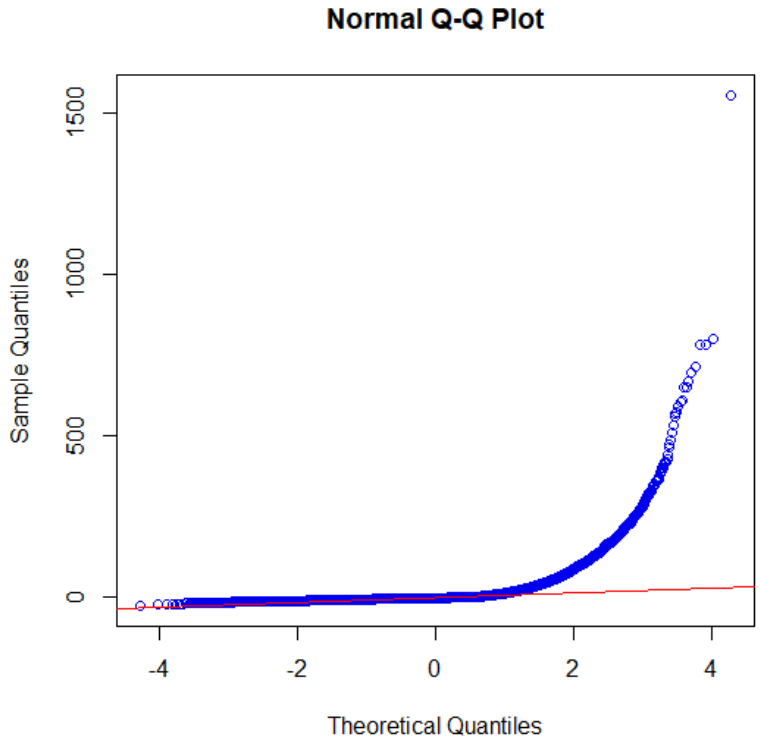
    Kết quả:

```
        Anderson-Darling normality test

data:  delay$dep_delay
A = 8714.8, p-value < 2.2e-16
```

    Nhận xét: Ta có thể thấy p-value $<$ 2.2e - 16 $<$ 0.05. Vậy thời gian khởi hành trễ của các chuyến bay khởi hành từ Portland năm 2014 không có phân phối chuẩn.
  - Kiểm tra lại bằng đồ thị QQ-plot:

```
qqnorm(delay$dep_delay, col = "blue2")
qqline(delay$dep_delay, col = "red")
```

    Kết quả:



Nhận xét: Biểu đồ QQ-plot cho ta thấy có rất nhiều những giá trị quan sát được không nằm trên đường kì vọng của phân phối chuẩn.
$\Rightarrow$ Thời gian khởi hành trễ của các hãng hàng không khởi hành từ Portland không có phân phối chuẩn.

- **Kiểm tra giả định về tính đồng nhất phương sai:**
  Sử dụng phương pháp kiểm định Levene:
  Để hiện thực được kiểm định này ta phải import thư viện *car*

```
library("car")
leveneTest(dep_delay ~ carrier, data = delay)
```

Kết quả:

```
Levene's Test for Homogeneity of Variance (center = median)
        Df F value    Pr(>F)
group   10  52.608 < 2.2e-16 ***
      53324
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Warning message:
In leveneTest.default(y = y, group = group, ...) : group coerced to factor.
```

Nhận xét: Trong kết quả của Levene's test, ta có thể thấy được p-value $<$ 2.2e - 16 $<$ 0.05 nên phương sai không đồng nhất với độ tin cậy 95%.

2.4 Câu d

2.4.1 Yêu cầu

Thực hiện phân tích ANOVA một nhân tố. Trình bày bảng phân tích phương sai trong báo cáo. Cho kết luận.

2.4.2 Lời giải R

```
ano_test <- aov(dep_delay ~ carrier, data = delay)
ano_test
summary(ano_test)
```

Kết quả: Từ kết quả thu được, ta đọc được các giá trị sau:

```
> ano_test
Call:
   aov(formula = dep_delay ~ carrier, data = delay)

Terms:
                carrier Residuals
Sum of Squares   1014194  48922853
Deg. of Freedom      10     53324

Residual standard error: 30.28967
Estimated effects may be unbalanced
> summary(ano_test)
              Df   Sum Sq Mean Sq F value Pr(>F)
carrier       10  1014194  101419   110.5 <2e-16 ***
Residuals   53324 48922853     917
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

1. SSB = 1014194, Bậc tự do k - 1 = 10 (k = 11 nhóm).
2. SSW = 48922853, Bậc tự do N - k = 53324 - 11 = 53313.
3. MSB = SSB/(k-1) = 101419.
4. MSW = SSW/(N-k) = 917.
5. Thống kê kiểm định F = MSB/MSW = 110.5.
6. Giá trị p-value $<$ 2e - 16 rất bé. Do đó, có sự khác biệt về thời gian khởi hành trễ trung bình giữa các hãng hàng không đối với các chuyến bay khởi hành từ Portland

- **So sánh bội sau phân tích phương sai**

```
TukeyHSD(ano_test)
```

Kết quả:

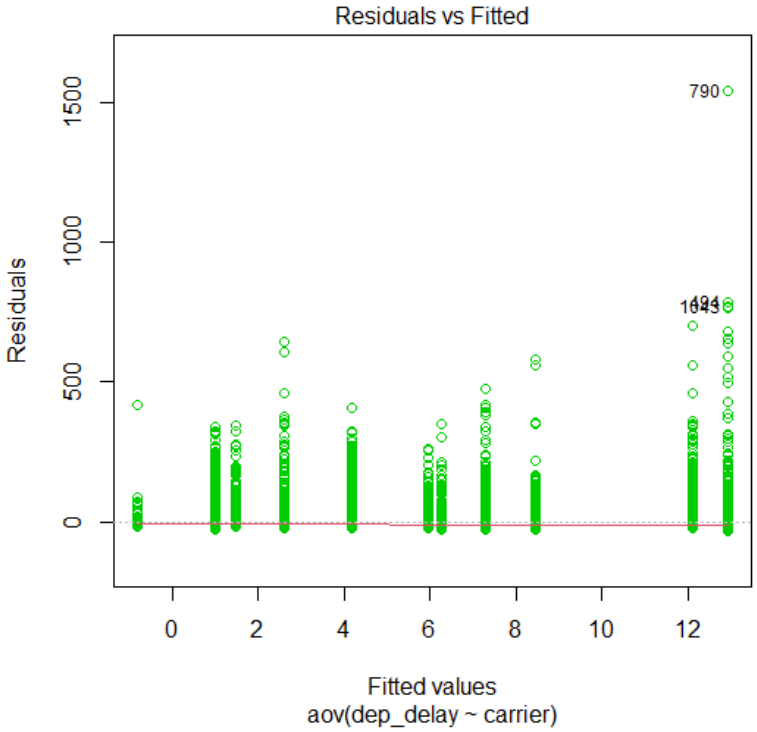
```
> TukeyHSD(ano_test)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = dep_delay ~ carrier, data = delay)

$carrier
             diff           lwr        upr     p adj
AS-AA -11.9415456 -14.196761667 -9.6863295 0.0000000
B6-AA  -6.9669247 -10.391999095 -3.5418503 0.0000000
DL-AA -10.3336444 -12.820636831 -7.8466520 0.0000000
F9-AA  -4.4963138  -7.861497860 -1.1311298 0.0008677
HA-AA -13.7300374 -19.242406339 -8.2176684 0.0000000
OO-AA  -8.7426620 -11.047399461 -6.4379245 0.0000000
UA-AA  -5.6450010  -8.076905754 -3.2130963 0.0000000
US-AA -11.4452985 -14.338686069 -8.5519110 0.0000000
VX-AA  -6.6795499 -10.994324841 -2.3647750 0.0000336
WN-AA  -0.8229467  -3.102234605  1.4563411 0.9863704
B6-AS   4.9746209   2.124155336  7.8250864 0.0000011
DL-AS   1.6079011   0.001927798  3.2138745 0.0494137
F9-AS   7.4452317   4.667015996 10.2234475 0.0000000
HA-AS  -1.7884918  -6.963453713  3.3864701 0.9902503
OO-AS   3.1988836   1.892807773  4.5049594 0.0000000
UA-AS   6.2965445   4.777275058  7.8158140 0.0000000
US-AS   0.4962471  -1.686803721  2.6792978 0.9997162
VX-AS   5.2619956   1.387553524  9.1364378 0.0006339
WN-AS  11.1185988   9.857975045 12.3792226 0.0000000
DL-B6  -3.3667197  -6.403870562 -0.3295689 0.0158595
F9-B6   2.4706109  -1.319325478  6.2605473 0.5785830
HA-B6  -6.7631127 -12.544580096 -0.9816453 0.0077303
OO-B6  -1.7757373  -4.665541567  1.1140671 0.6646173
```

Nhận xét: Ta có thể thấy hãng WN có thời gian trễ lớn nhất trong 11 hãng với thời gian trễ xấp xỉ 18 phút.

- **Kiểm tra các giả định Anova**
  - Kiểm tra tính đồng nhất của giả định phương sai bằng đồ thị thặng dư và phương pháp Levene's test.

```
plot(ano_test, 1, col = "green3")
leveneTest(dep_delay ~ carrier, data = delay)
```

Kết quả:




```
Levene's Test for Homogeneity of Variance (center = median)
        Df F value    Pr(>F)
group    10  52.608 < 2.2e-16 ***
      53324
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Warning message:
In leveneTest.default(y = y, group = group, ...) : group coerced to factor.
```

Nhận xét:

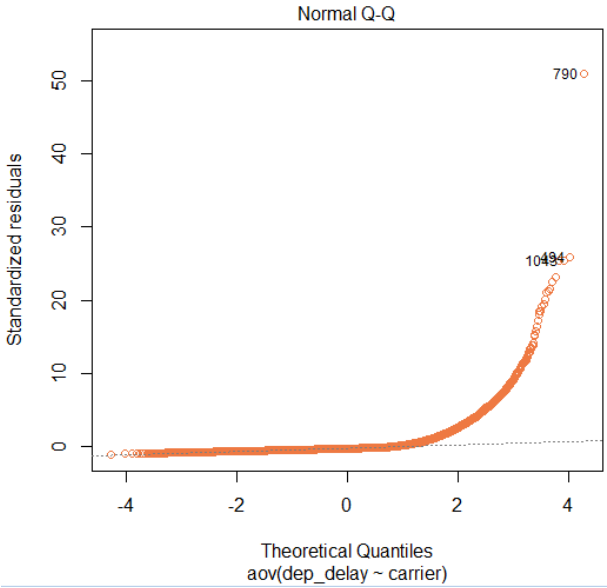
1. Nhìn vào đồ thị thặng dư ta thấy các điểm thặng dư không phân tán đều xung quanh đường thẳng màu đỏ.

2. Ngoài ra, qua kiểm định Levene, ta thấy giá trị p-value $<$ 2.2e-16 $<$ 0.05.
   $\Rightarrow$ Do đó, chúng ta không thể giả định tính đồng nhất của phương sai trong các hãng khác nhau

– Kiểm tra giả định quy tắc.
Sử dụng kiểm định Kruskal test, đồ thị QQ-plot và kiểm định Anderson - Darling. Kiểm định Kruskal-Wallis để so sánh sự khác biệt về giá trị trung bình của 1 biến phụ thuộc theo 2 hay nhiều chiều của biến độc lập nhưng không bắt buộc biến độc lập là phân phối chuẩn.

```
plot(ano_test, 2, col = "sienna2")
aov_res <- residuals(ano_test)
ad.test(aov_res)
kruskal.test(dep_delay ~ carrier, data = delay)
```

Kết quả:




```
> ad.test(aov_res)

        Anderson-Darling normality test

data:  aov_res
A = 8320.1, p-value < 2.2e-16

> kruskal.test(dep_delay ~ carrier, data = delay)

        Kruskal-Wallis rank sum test

data:  dep_delay by carrier
Kruskal-Wallis chi-squared = 6689.1, df = 10, p-value < 2.2e-16
```

Nhận xét:

1. Đồ thị Q-Q cho phép kiểm tra giả định về phân phối chuẩn của các sai số. Nếu các điểm thặng dư nằm trên cùng 1 đường thẳng thì điều kiện về phân phối chuẩn được thỏa.
2. Hơn nữa qua kiểm định Anderson-Darling cho giá trị p-value $<$ 2.2e-16 $<$ 0.05 $\Rightarrow$ Giả định sai số có phân phối chuẩn chưa thực sự thỏa mãn.
3. Qua kiểm định Kruskal-Wallis, ta thấy giá trị p-value $<$ 2.2e-16 $<$ 0.05 nên mô hình có ý nghĩa về mặt thống kê.

Phần C
PHẦN RIÊNG

Tập tin **"machine.data"** chứa các dữ liệu liên quan đến phần cứng máy tính và hiệu năng tương đối của CPU, cả hai được thu thập từ cơ sở dữ liệu trên Web.
Link dữ liệu: https://archive.ics.uci.edu/ml/datasets/Computer+Hardware
Các biến chính trên bộ dữ liệu:

- **myct**: Thời gian 1 chu kỳ tính bằng nano giây.
- **mmin**: Dung lượng tối thiểu của bộ nhớ chính, tính bằng kilobytes.
- **mmax**: Dung lượng tối đa của bộ nhớ chính, tính bằng kilobytes.
- **cach**: Dung lượng bộ nhớ cache tính bằng kilobytes.
- **chmin**: Số lượng kênh tối thiểu.
- **chmax**: Số lượng kênh tối đa.
- **prp**: Hiệu suất tương đối của CPU.

1 Đọc dữ liệu (Import data)

1.1 Lời giải R

```
#import data
data <- read.csv("machine.data", header = FALSE)
#rename column
colnames(data) <- c( "vendor_name", "model_name", "myct", "mmin",
"mmax", "cach", "chmin", "chmax", "prp", "erp" )
View(data)
```

1.2 Kết quả

(row)	vendor_name	model_name	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	132	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
11	apollo	dn320	400	1000	3000	0	1	2	38	23
12	apollo	dn420	400	512	3500	4	1	6	40	24
13	basf	7/65	60	2000	8000	65	1	8	92	70

2 Làm sạch dữ liệu (Data cleaning)

2.1 Lọc dữ liệu

Do bảng dữ liệu chứa nhiều thông tin nên chúng ta cần lọc ra để có một bảng dễ tiếp cận hơn. Ta sẽ chỉ chọn các cột **myct, mmin, mmax, cach, chmin, chmax, prp**.

2.2 Lời giải R

```
new_data <- subset(data, select = c("myct", "mmin", "mmax", "cach", "chmin", "chmax",
"prp" ))
View(new_data)
```

2.3 Kết quả

(row)	myct	mmin	mmax	cach	chmin	chmax	prp
1	125	256	6000	256	16	128	198
2	29	8000	32000	32	8	32	269
3	29	8000	32000	32	8	32	220
4	29	8000	32000	32	8	32	172
5	29	8000	16000	32	8	16	132
6	26	8000	32000	64	8	32	318
7	23	16000	32000	64	16	32	367
8	23	16000	32000	64	16	32	489
9	23	16000	64000	64	16	32	636
10	23	32000	64000	128	32	64	1144
11	400	1000	3000	0	1	2	38
12	400	512	3500	4	1	6	40
13	60	2000	8000	65	1	8	92

2.4 Kiểm tra dữ liệu bị khuyết

Ta kiểm tra dữ liệu bị khuyết bằng lệnh:

```
apply(is.na(new_data), 2, sum)
```

và kết quả:

```
myct  mmin  mmax  cach chmin chmax   prp
   0     0     0     0     0     0     0
```

Điều này chứng tỏ dữ liệu lọc ra không chứa các giá trị NA.

3 Làm rõ dữ liệu (Data visualization)

3.1 Tính các giá trị thống kê mô tả của biến liên tục

3.1.1 Lời giải R

Tất cả các biến chính trong bài toán đều là biến liên tục. Vì vậy ta tính các giá trị trung bình, trung vị, độ lệch chuẩn, giá trị lớn nhất và giá trị nhỏ nhất bằng các lệnh:

```
mean <- apply(new_data, 2, mean)
medium <- apply(new_data, 2, median)
sd <- apply(new_data, 2, sd)
min <- apply(new_data, 2, min)
max <- apply(new_data, 2, max)
data_table <- t(data.frame(mean, medium, sd, min, max))
View(data_table)
```

3.1.2 Kết quả

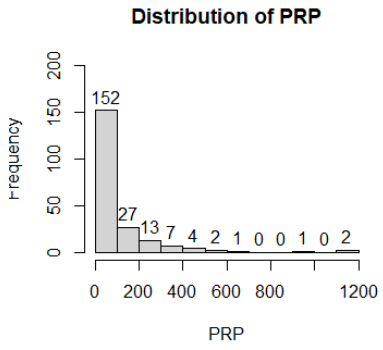
(row)	myct	mmin	mmax	cach	chmin	chmax	prp
mean	203.823	2867.9809	11796.1531	25.2057	4.6986	18.2679	105.622
medium	110	2000	8000	8	2	8	50
sd	260.2629	3878.7428	11726.5644	40.6287	6.8163	25.9973	160.8307
min	17	64	64	0	0	0	6
max	1500	32000	64000	256	52	176	1150

3.2 Đồ thị phân phối của biến prp

3.2.1 Lời giải R

```
hist(new_data$prp, main = "Distribution of PRP", xlab = "PRP", labels = TRUE, ylim = range(0, 200) )
```

3.2.2 Kết quả



Nhận xét:

- Đồ thị không có dạng phân phối chuẩn.
- Hiệu năng CPU phân bố tập trung từ 21-100

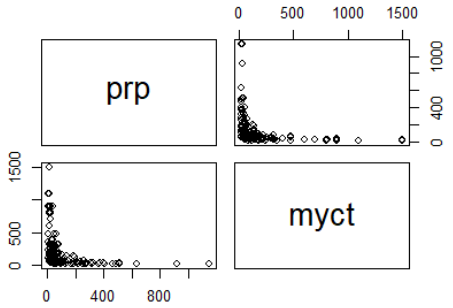
3.3 Đồ thị phân phối của biến prp theo các biến liên tục myct, mmin, mmax, cach, chmin, chmax

3.3.1 Lời giải R

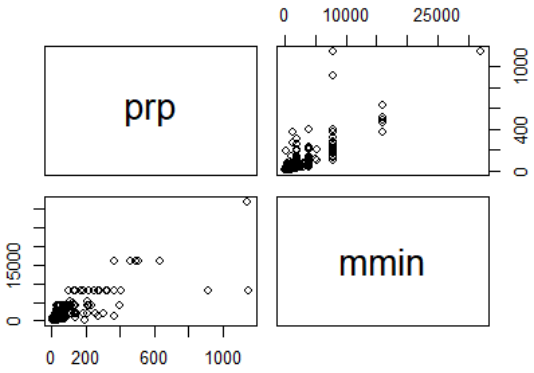
```
pairs(prp ~ myct, new_data)
pairs(prp ~ mmin, new_data)
pairs(prp ~ mmax, new_data)
pairs(prp ~ cach, new_data)
pairs(prp ~ chmin, new_data)
pairs(prp ~ chmax, new_data)
```

3.3.2 Kết quả

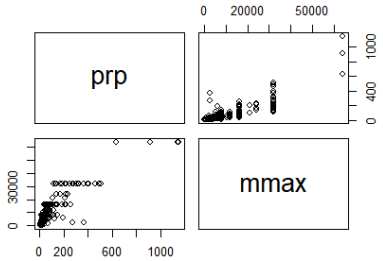
- Đối với biến *myct*:



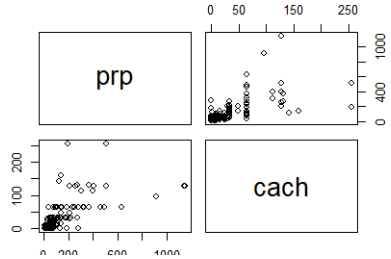
- Đối với biến *mmin*:



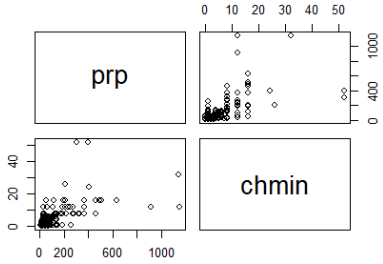
- Đối với biến *mmax*:



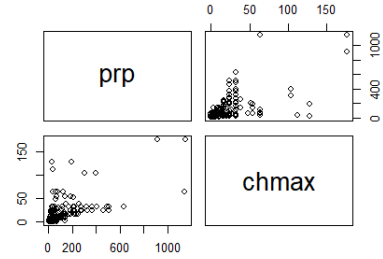
- Đối với biến *cach*:



- Đối với biến *chmin*:



- Đối với biến *chmax*:



Nhận xét:

- Dựa trên đồ thị phân phối của biến prp theo các biến myct, mmin, mmax, cach, chmin và chmax, ta có thể nhận thấy giữa chúng có mối quan hệ tuyến tính (cụ thể ở đây là đồng biến).
- Tuy nhiên vẫn còn nhiều điểm ngoại lai do biến prp chịu ảnh hưởng của nhiều biến khác nhau.
  $\Rightarrow$ Để xác định xem chúng có thật sự có mối quan hệ tuyến tính hay không thì ta phải đi xây dựng mô hình hồi quy tuyến tính giữa các biến độc lập với biến phụ thuộc (prp).

4 Xây dựng các mô hình hồi quy tuyến tính

4.1 Mô hình gồm prp là biến phụ thuộc, tất cả các biến còn lại là biến độc lập

Mô hình hồi quy tuyến tính bao gồm:

- Biến phụ thuộc: prp
- Biến độc lập: myct, mmin, mmax, cach, chmin, chmax

4.1.1 Lời giải R

```
model1 <- lm(prp ~ myct + mmin + mmax + cach + chmin + chmax, data = new_data)
summary(model1)
```

4.1.2 Kết quả

```
Call:
lm(formula = prp ~ myct + mmin + mmax + cach + chmin + chmax,
    data = new_data)

Residuals:
    Min      1Q  Median      3Q     Max
-195.82  -25.17    5.40   26.52  385.75

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.589e+01  8.045e+00  -6.948 5.00e-11 ***
myct         4.885e-02  1.752e-02   2.789   0.0058 **
mmin         1.529e-02  1.827e-03   8.371 9.42e-15 ***
mmax         5.571e-03  6.418e-04   8.681 1.32e-15 ***
cach         6.414e-01  1.396e-01   4.596 7.59e-06 ***
chmin       -2.704e-01  8.557e-01  -0.316   0.7524
chmax        1.482e+00  2.200e-01   6.737 1.65e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 59.99 on 202 degrees of freedom
Multiple R-squared:  0.8649,    Adjusted R-squared:  0.8609
F-statistic: 215.5 on 6 and 202 DF,  p-value: < 2.2e-16
```

Nhận xét 1:
Từ cột **Estimate**, ta có phương trình hồi quy:
prp = -5.589e+01 + (4.885e-02)*myct + (1.529e-02)*mmin + (5.571e-03)*mmax + (6.414e-01)*cach + (-2.704e-01)*chmin + (1.482e+00)*chmax
Xét mức ý nghĩa 5%
GIẢ THUYẾT

- H0: Hệ số hồi quy không có ý nghĩa thống kê.
- H1: Hệ số hồi quy có ý nghĩa thống kê.

KIỂM ĐỊNH SỰ PHÙ HỢP CỦA HỆ SỐ HỒI QUY
Phương pháp kiểm định bằng p - value (Pr(>|t|))

- Pr(>|t|) > mức ý nghĩa α - chấp nhận giả thiết H0, tức hệ số hồi quy ứng với biến phụ thuộc không có ý nghĩa thống kê, ta sẽ loại biến này ra khỏi mô hình.
- Pr(>|t|) < mức ý nghĩa α - bác bỏ giả thiết H0, chấp nhận giả thiết H1, tức hệ số hồi quy ứng với biến phụ thuộc có ý nghĩa thống kê, ta sẽ nhận kết quả biến này.

Nhận xét 2:
Dựa vào kết quả, xét mức ý nghĩa $\alpha = 5\%$, ta sẽ loại bỏ biến **chmin**.

4.2 Đề xuất mô hình hồi quy tuyến tính hợp lý

Xét mô hình tuyến tính cùng bao gồm biến **prp** là biến phụ thuộc nhưng:
Mô hình **model2** là mô hình loại bỏ biến **chmin** từ **model1**.

- Biến phụ thuộc: prp
- Biến độc lập: myct, mmin, mmax, cach, chmax

Ta dùng lệnh **anova()** để đề xuất mô hình hồi quy hợp lý hơn như sau:

4.2.1 Lời giải R

```
model1 <- lm(prp ~ myct + mmin + mmax + cach + chmin + chmax, data = new_data)
summary(model1)
model2 <- lm(prp ~ myct + mmin + mmax + cach + chmax, data = new_data)
summary(model2)
anova(model1, model2)
```

4.2.2 Kết quả

- Mô hình 2:

```
Call:
lm(formula = prp ~ myct + mmin + mmax + cach + chmax, data = new_data)

Residuals:
    Min      1Q  Median      3Q     Max
-193.37  -24.95    5.76   26.64  389.66

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.608e+01  8.007e+00  -7.003 3.59e-11 ***
myct         4.911e-02  1.746e-02   2.813   0.0054 **
mmin         1.518e-02  1.788e-03   8.490 4.34e-15 ***
mmax         5.562e-03  6.396e-04   8.695 1.18e-15 ***
cach         6.298e-01  1.344e-01   4.687 5.07e-06 ***
chmax        1.460e+00  2.076e-01   7.031 3.06e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 59.86 on 203 degrees of freedom
Multiple R-squared:  0.8648,    Adjusted R-squared:  0.8615
F-statistic: 259.7 on 5 and 203 DF,  p-value: < 2.2e-16
```

- So sánh 2 mô hình

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

Model 1: prp ~ myct + mmin + mmax + cach + chmin + chmax
Model 2: prp ~ myct + mmin + mmax + cach + chmax
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    202 726920
2    203 727279 -1   -359.25 0.0998 0.7524
>
```

Xét mức ý nghĩa 5%
GIẢ THUYẾT

- H0: Hai mô hình model1, model2 giống nhau.
- H1: Hai mô hình model1, model2 khác nhau.

Nhận xét:
Theo kiểm định p - value với mức ý nghĩa $\alpha = 5\%$ ở mô hình 1 và 2, ta thấy p - value > 0.05 nên không thể bác bỏ được H0. Để chọn mô hình hiệu quả hơn, ta dựa vào hệ số Adjusted R-squared (hệ số xác định hiệu chỉnh) từ kết quả của câu lệnh *summary()*. Nhận thấy Adjusted R-squared ở mô hình 2 cao hơn (0.8615), do đó ta chọn mô hình 2 thay thế mô hình 1.

4.3 Suy luận sự tác động của các biến đến hiệu năng CPU

GIẢI THÍCH

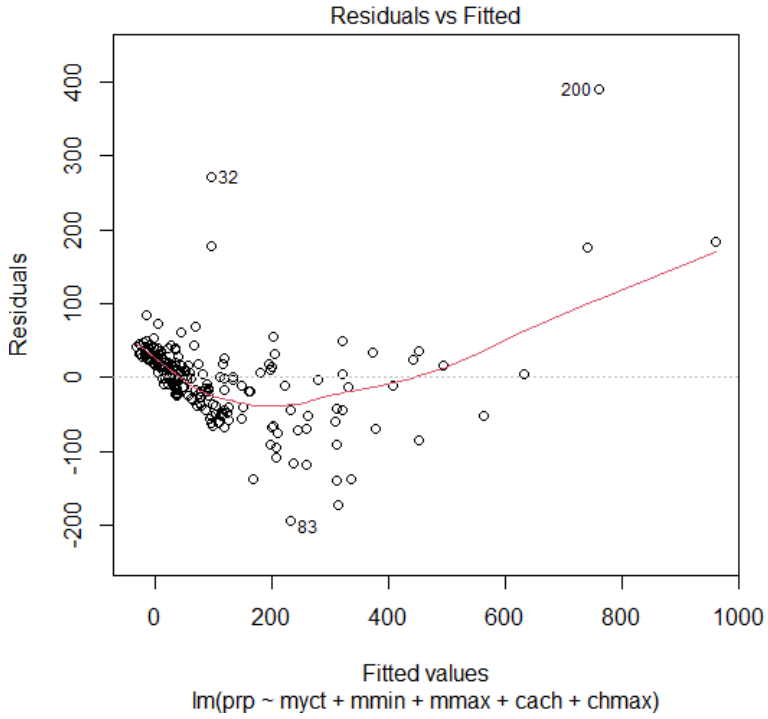
- Để đánh giá sự tác động của các biến lên hiệu năng CPU, ta quan tâm các hệ số hồi quy p - value tương ứng. Với các biến **myct, mmin, mmax, cach, chmax**, p - value < 0.01. Điều đó chứng tỏ ảnh hưởng của tất cả các biến trong mô hình hồi quy đều có ý nghĩa rất lớn đến hiệu năng CPU.
- Hệ số hồi quy của 1 biến dự báo cũng được xem như ảnh hưởng trung bình lên biến phụ thuộc là hiệu năng CPU khi tăng 1 đơn vị của biến dự báo đó (giả sử khi các biến dự báo khác không đổi). Ví dụ, hệ số hồi quy ứng với mmin = 0.01518 khi mmin tăng 1 kilobyte thì ta có thể kỳ vọng hiệu năng CPU có thể tăng 0.01518(giả sử rằng các biến dự báo còn lại không đổi). Tương tự cũng như hệ số hồi quy ứng với chmax = 1.46 thì ứng với chmax tăng 1 thì ta có thể kỳ vọng hiệu năng CPU tăng 1.46(giả sử rằng các biến dự báo còn lại không đổi), tương tự cho các biến còn lại.
- Hệ số R^2 hiệu chỉnh của mô hình có giá trị khoảng 0.8615 nghĩa là khoảng $86,15.\%$ sự biến thiên của prp được giải thích bởi các biến độc lập.

4.4 Đồ thị biểu diễn sai số hồi quy và giá trị dự báo

4.4.1 Lời giải R

```
plot(model2, which = 1)
```

4.4.2 Kết quả



4.4.3 Nhận xét

- Giả định về tính tuyến tính của dữ liệu chưa thực sự thoả mãn.
- Giả định các sai số có kỳ vọng bằng 0 chưa thoả mãn.
- Giả định về tính đồng nhất của phương sai chưa thoả mãn

5 Dự báo

Dùng mô hình **model2** để dự báo hiệu năng CPU tại 2 thuộc tính như sau:

- **x1**: myct = mean(myct), mmin = mean(mmin), mmax = mean(mmax), cach = mean(cach), chmax = mean(chmax)
- **x2**: myct = max(myct), mmin = max(mmin), mmax = max(mmax), cach = max(cach), chmax = max(chmax)

5.1 Lời giải R

Tạo một dataframe data_test gồm 5 cột tương ứng với 5 biến (myct, mmin, mmax, cach, chmax) 2 hàng tương ứng với 2 thuộc tính dự đoán. Đặt lại tên 2 hàng là test1, test2.

```
data_test <- data.frame(
myct <- c(mean(new_data$myct), max(new_data$myct),
mmin <- c(mean(new_data$mmin), max(new_data$mmin)),
mmax <- c(mean(new_data$mmax), max(new_data$mmax)),
cach <- c(mean(new_data$cach), max(new_data$cach)),
chmax <- c(mean(new_data$chmax), max(new_data$chmax))
)
rownames(data_test) <- c("test1", "test2")
colnames(data_test) <- c("myct", "mmin", "mmax", "cach", "chmax")
View(data_test)
```

(row)	myct	mmin	mmax	cach	chmax
test1	203.823	2867.9809	11796.1531	25.2057	18.2679
test2	1500	32000	64000	256	176

Dùng lệnh **predict()** để dự báo hiệu năng CPU tại 2 thuộc tính **x1, x2**

```
pred = data.frame(predict(model2,data_test,interval="confidence"))
pred$range = pred$upr - pred$lwr
```

(row)	fit	lwr	upr	range
test1	105.622	97.4585	113.7855	16.3269
test2	1277.4822	1171.2865	1383.6778	212.3913

5.2 Nhận xét

Cột fit thể hiện kết quả dự đoán của từng thuộc tính, khoảng (lwr, upr) là độ dài của khoảng ước lượng giá trị (range). Ta sẽ tiến hành so sánh khoảng ước lượng của hai thuộc tính trên bằng cách lấy tỉ số range giữa hai thuộc tính x2/x1

```
> pred[2, 4] / pred[1, 4]
[1] 13.00865
```

Kết quả cho thấy tỉ số khoảng ước lượng của thuộc tính x2/x1 vào khoảng 13.00865 tức là khoảng ước lượng của thuộc tính x1 nhỏ hơn 13.00865 lần so với thuộc tính x2.
⇒ Khoảng ước lượng cho giá trị dự báo của thuộc tính x1 hợp lí hơn.