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

XÁC SUẤT - THỐNG KÊ

NHÓM: 12 - CHỦ ĐỀ: 06
GIẢNG VIÊN HƯỚNG DẪN: TS. HOÀNG VĂN HÀ

STT	MSSV	Họ	Tên	Ngành học
1	2312749	Nguyễn Đức	Phương	Khoa học máy tính
2	2213698	Nguyễn Hải	Trung	Khoa học máy tính
3	2210714	Phạm Tuấn	Đạt	Khoa học máy tính
4	2213609	Phan Duệ	Triết	Khoa học máy tính
5	2213063	Nguyễn Trung	Tân	Kỹ thuật máy tính

Tp. Hồ Chí Minh, Tháng 08/2024

BẢNG PHÂN CÔNG VIỆC

STT	MSSV	Họ	Tên	Mô tả đóng góp
1	2312749	Nguyễn Đức	Phương	Tổng hợp code R, sửa lỗi code, trình bày
2	2213698	Nguyễn Hải	Trung	HĐ1.1: Code R nhận xét tiền xử lí và thống kê mô tả
3	2210714	Phạm Tuấn	Đạt	HĐ2.1
4	2213609	Phan Duệ	Triết	HĐ1.2: Code R nhận xét tiền mô hình thống kê suy diễn
5	2213063	Nguyễn Trung	Tân	HĐ2.2

Mục lục

Danh sách hình vẽ

I CƠ SỞ LÝ THUYẾT

II HOẠT ĐỘNG 1

1 Đề bài

Dữ liệu được cho trong file **"auto-mpg.csv"** là bộ dữ liệu tiêu thụ nhiên liệu của xe trong thành phố. Dữ liệu được lấy từ UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Auto+MPG).
Bộ dữ liệu gồm 398 quan trắc trên 9 biến sau:

- **mpg**: (continuous) mức tiêu thụ nhiên liệu tính theo dặm trên galon (miles/gallon)
- **cylinders**: (multi-valued discrete) số xy lanh.
- **displacement**: (continuous) kích thước động cơ
- **horsepower**: (continuous) công suất động cơ.
- **weight**: (continuous) khối lượng.
- **acceleration**: (continuous) gia tốc xe.
- **model year**: (multi-valued discrete) năm sản xuất model (2 số cuối).
- **origin**: (multi-valued discrete) nơi sản xuất: 1 - North American, 2 - Europe, 3 - Asia.
- **car name**: (multi-valued discrete) tên xe.

Các bước thực hiện:

1. Nhập và "làm sạch" dữ liệu (lưu ý, biến **"horsepower"** có 6 quan trắc thiếu dữ liệu; xét xem có dữ liệu ngoại lai không?), thực hiện các thống kê mô tả. (Chú ý các cột của file "horsepower"**"horsepower"** được phân tách bởi dấu ";", khi đọc file dữ liệu dùng lệnh **"read.csv"** cần thêm sep = ";")
2. Chia bộ dữ liệu làm 2 phần: mẫu huấn luyện (training dataset) gồm 200 quan trắc đặt tên **"auto_mpg1"** và mẫu kiểm tra (validation dataset) gồm các quan trắc còn lại trong bộ dữ liệu ban đầu đã "làm sạch", đặt tên **"auto_mpg2"**
3. Chọn mô hình tốt nhất giải thích cho biến phụ thuộc **"mpg"** thông qua việc chọn lựa các biến độc lập phụ hợp trong 8 biến độc lập còn lại từ mẫu huấn luyện " **"auto_mpg1"**. Cần trình bày từng bước phương pháp chọn, tiêu chuẩn chọn mô hình, lý do chọn phương pháp đó.
4. Kiểm tra các giả định (giả thiết) của mô hình
5. Nêu ý nghĩa của mô hình đã chọn.
6. Dự báo (Prediction): Sử dụng mẫu kiểm tra (validation dataset) **"auto_mpg2"** và dựa vào mô hình tốt nhất được chọn trên đưa số liệu dự báo cho biến phụ thuộc **"mpg"**. Gọi kết quả dự báo này là biến **"predict_mpg"**.
7. So sánh kết quả dự báo **"ppredict_mpg"** với giá trị thực tế của **"mpg"**. Rút ra nhận xét?

2 Làm sạch dữ liệu và thống kê mô tả

2.1 Nhập và làm sạch dữ liệu

Đọc tệp tin **auto_mpg.csv**

```
> auto_mpgData <- read.csv("C:/HCMUT/233/XSTK/auto_mpg.csv", sep=";")
>
> head(auto_mpgData)
```

	mgp	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	car_name
1	18	8	307.0	130.0	3504	12.0	70	1	chevrolet chevelle malibu
2	15	8	350.0	165.0	3693	11.5	70	1	buick skylark 320
3	18	8	318.0	150.0	3436	11.0	70	1	plymouth satellite
4	16	8	304.0	150.0	3433	12.0	70	1	amc rebel sst
5	17	8	302.0	140.0	3449	10.5	70	1	ford torino
6	15	8	429.0	198.0	4341	10.0	70	1	ford galaxie 500
7	14	8	454.0	220.0	4354	9.0	70	1	chevrolet impala
8	14	8	440.0	215.0	4312	8.5	70	1	plymouth fury iii
9	14	8	455.0	225.0	4425	10.0	70	1	pontiac catalina
10	15	8	390.0	190.0	3850	8.5	70	1	amc ambassador dpl
11	15	8	383.0	170.0	3563	10.0	70	1	dodge challenger se
12	14	8	340.0	160.0	3609	8.0	70	1	plymouth 'cuda 340
13	15	8	400.0	150.0	3761	9.5	70	1	chevrolet monte carlo
14	14	8	455.0	225.0	3086	10.0	70	1	buick estate wagon (sw)

Hình 1: *code R và kết quả khi đọc tệp tin*

Ta chạy lệnh glimpse(auto_mgpData) thấy có 398 quan trắc và 9 cột

```
> glimpse(auto_mpgData)
Rows: 398
Columns: 9
```

Hình 2: *Số quan trắc và cột*

Thay thế giá trị "?" bằng NA và kiểm tra số lượng giá trị NA trong mỗi cột

```
> auto_mpgData[auto_mpgData == "?"] <- NA
> apply(is.na(auto_mpgData), 2, sum)
         mgp    cylinders displacement   horsepower       weight acceleration   model_year
           0            0            0            6            0            0            0
      origin     car_name
           0            0
> apply(is.na(auto_mpgData),2,mean)
         mgp    cylinders displacement   horsepower       weight acceleration   model_year
  0.00000000   0.00000000   0.00000000   0.01507538   0.00000000   0.00000000   0.00000000
      origin     car_name
  0.00000000   0.00000000
>
```

Hình 3: *số lượng giá trị NA trong mỗi cột*

Sau khi chạy lệnh, ta thấy cột horsepower có chứa 6 ô NA, chiếm 1,5%, với tỷ lệ này ta có thể xóa các hàng có chứa ô giá trị NA.

Loại bỏ các hàng chứa giá trị NA và kiểm tra lại số lượng giá trị NA

```
> auto_mpgData <- na.omit(auto_mpgData)
>
> apply(is.na(auto_mpgData), 2, sum)
         mgp    cylinders displacement   horsepower       weight acceleration   model_year
           0            0            0            0            0            0            0
      origin     car_name
           0            0
> glimpse(auto_mpgData)
Rows: 392
Columns: 9
```

Hình 4: *kiểm tra lại số lượng giá trị NA*

Sau khi loại bỏ ta kiểm tra lại thấy còn 392 quan trắc(đã loại bỏ 6 quan trắc)
Kiểm tra kiểu dữ liệu của các cột

```
> flagMpg <- is.numeric(auto_mpgData$mpg)
> flagCylinders <- is.numeric(auto_mpgData$cylinders)
> flagDisplacement <- is.numeric(auto_mpgData$displacement)
> flagHorsePower <- is.numeric(auto_mpgData$horsepower)
> flagWeight <- is.numeric(auto_mpgData$weight)
> flagAcceleration <- is.numeric(auto_mpgData$acceleration)
> flagYear <- is.numeric(auto_mpgData$model_year)
> flagOrigin <- is.numeric(auto_mpgData$origin)
```

Values	
flagAccelera…	TRUE
flagCylinders	TRUE
flagDisplace…	TRUE
flagHorsePow…	FALSE
flagMpg	FALSE
flagOrigin	TRUE
flagWeight	TRUE
flagYear	TRUE

Hình 5: *Kiểm tra dữ liệu*

Chuyển cột 'horsepower' thành kiểu số và kiểm tra lại

```
> auto_mpgData$horsepower <- as.numeric(auto_mpgData$horsepower)
> is.numeric(auto_mpgData$horsepower)
[1] TRUE
```

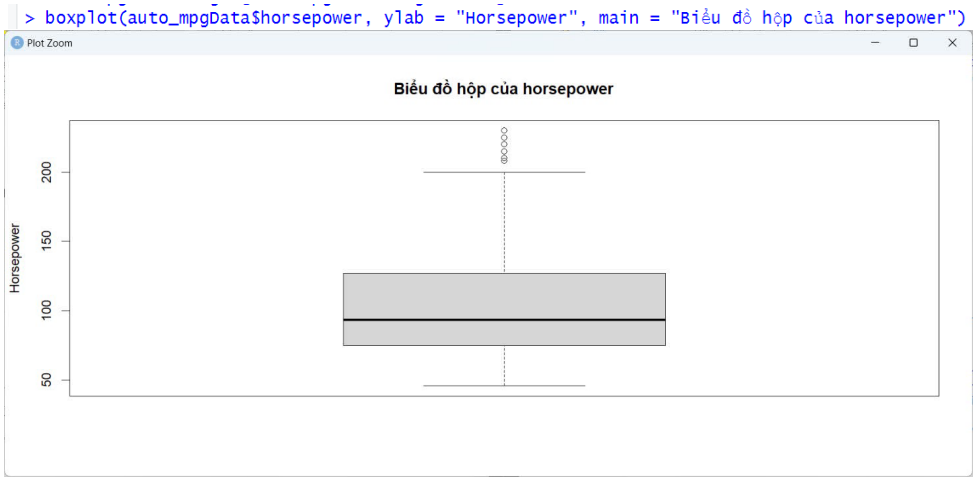
Hình 6: *code R và chuyển đổi dữ liệu 'horsepower'*

Thay thế các giá trị số trong cột 'origin' bằng tên khu vực

```
> auto_mpgData$origin[auto_mpgData$origin == 1] <- "North American"
> auto_mpgData$origin[auto_mpgData$origin == 2] <- "Europe"
> auto_mpgData$origin[auto_mpgData$origin == 3] <- "Asia"
```

Hình 7: *Thay thể giá trị origin*

Vẽ biểu đồ hộp cho cột 'horsepower', tìm và loại bỏ các giá trị ngoại lệ



Hình 8: *Biểu đồ cho cột 'horsepower'*

Xác định các giá trị ngoại lệ

```
> out <- boxplot.stats(auto_mpgData$horsepower)$out
out    num [1:10] 220 215 225 225 215 210 ...
```

Hình 9: *Xác định các giá trị ngoại lệ*

Tìm chỉ số của các giá trị ngoại lệ

```
> out_ind <- which(auto_mpgData$horsepower %in% c(out))
out_ind    int [1:10] 7 8 9 14 26 28 67 94 95 ...
```

Hình 10: *Tìm chỉ số của các giá trị ngoại lệ*

Loại bỏ các hàng chứa giá trị ngoại lệ

```
> clearData <- auto_mpgData[-out_ind, ]
out_ind    int [1:10] 7 8 9 14 26 28 67 94 95 ...
```

Hình 11: *Loại bỏ các hàng chứa giá trị ngoại lệ*

Hiển thị 10 hàng đầu của dữ liệu đã loại bỏ giá trị ngoại lệ

```
> head(clearData, 10)
   mgp cylinders displacement horsepower weight acceleration model_year         origin
1   18         8          307        130   3504         12.0         70 North American
2   15         8          350        165   3693         11.5         70 North American
3   18         8          318        150   3436         11.0         70 North American
4   16         8          304        150   3433         12.0         70 North American
5   17         8          302        140   3449         10.5         70 North American
6   15         8          429        198   4341         10.0         70 North American
10  15         8          390        190   3850          8.5         70 North American
11  15         8          383        170   3563         10.0         70 North American
12  14         8          340        160   3609          8.0         70 North American
13  15         8          400        150   3761          9.5         70 North American
                   car_name
1  chevrolet chevelle malibu
2          buick skylark 320
3         plymouth satellite
4              amc rebel sst
5                ford torino
6           ford galaxie 500
10        amc ambassador dpl
11       dodge challenger se
12        plymouth 'cuda 340
13     chevrolet monte carlo
```

Hình 12: *Hiển thị 10 hàng đầu của dữ liệu đã loại bỏ giá trị ngoại lệ*

Tính toán các thống kê mô tả cho các biến số
Đối với 5 biến định lượng

```
> continuousVar <- clearData[, c("mgp", "displacement", "horsepower", "weight", "acceleration")]
>
> trung_binh <- apply(continuousVar, 2, mean)  # Tính trung bình
> do_lech_chuan <- apply(continuousVar, 2, sd)  # Tính độ lệch chuẩn
> GTLN <- apply(continuousVar, 2, max)  # Tính giá trị lớn nhất
> GTNN <- apply(continuousVar, 2, min)  # Tính giá trị nhỏ nhất
> trung_vi <- apply(continuousVar, 2, median)  # Tính trung vị
> phan_vi1 <- apply(continuousVar, 2, quantile, probs = 0.25)  # Tính phân vị 25%
> phan_vi3 <- apply(continuousVar, 2, quantile, probs = 0.75)  # Tính phân vị 75%
> t(data.frame(trung_binh, do_lech_chuan, GTLN, GTNN, trung_vi, phan_vi1, phan_vi3))
                    mgp displacement horsepower    weight acceleration
trung_binh    23.721990    188.49084  101.47644 2940.9476    15.666754
do_lech_chuan  7.709764     99.03045   34.16331  825.6676     2.667648
GTLN          46.600000    429.00000  200.00000 5140.0000    24.800000
GTNN           9.000000     68.00000   46.00000 1613.0000     8.000000
trung_vi      23.000000    144.50000   92.00000 2764.5000    15.500000
phan_vi1      17.625000    101.75000   75.00000 2220.0000    14.000000
phan_vi3      29.000000    258.00000  119.00000 3533.7500    17.200000
>
```

Hình 13: *Thống kê mô tả cho 5 biến định lượng*

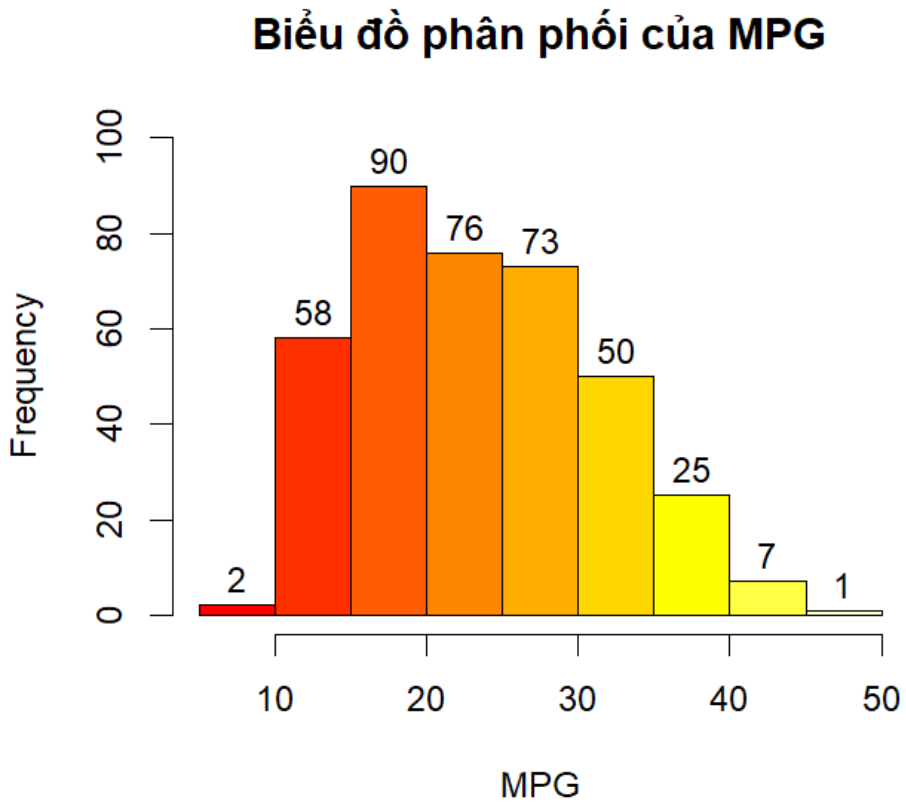
Đối với các biến định tính có thể thống kê còn lại, ta dùng table(<tên biến>):

```
> table(auto_mpgData$origin)

          Asia         Europe North American
            79             68            245
> table(auto_mpgData$cylinders)

  3   4   5   6   8
  4 199   3  83 103
> table(auto_mpgData$model_year)

70 71 72 73 74 75 76 77 78 79 80 81 82
29 27 28 40 26 30 34 28 36 29 27 28 30
>
```

Hình 14: *Thống kê mô tả cho biến định tính*

2.2 Vẽ các biểu đồ phân phối và biểu đồ hộp

```
> hist(clearData$mgp, main = "Biểu đồ phân phối của MPG", xlab = "MPG",
+      col = heat.colors(9), labels = TRUE, ylim = c(0, 100))
```



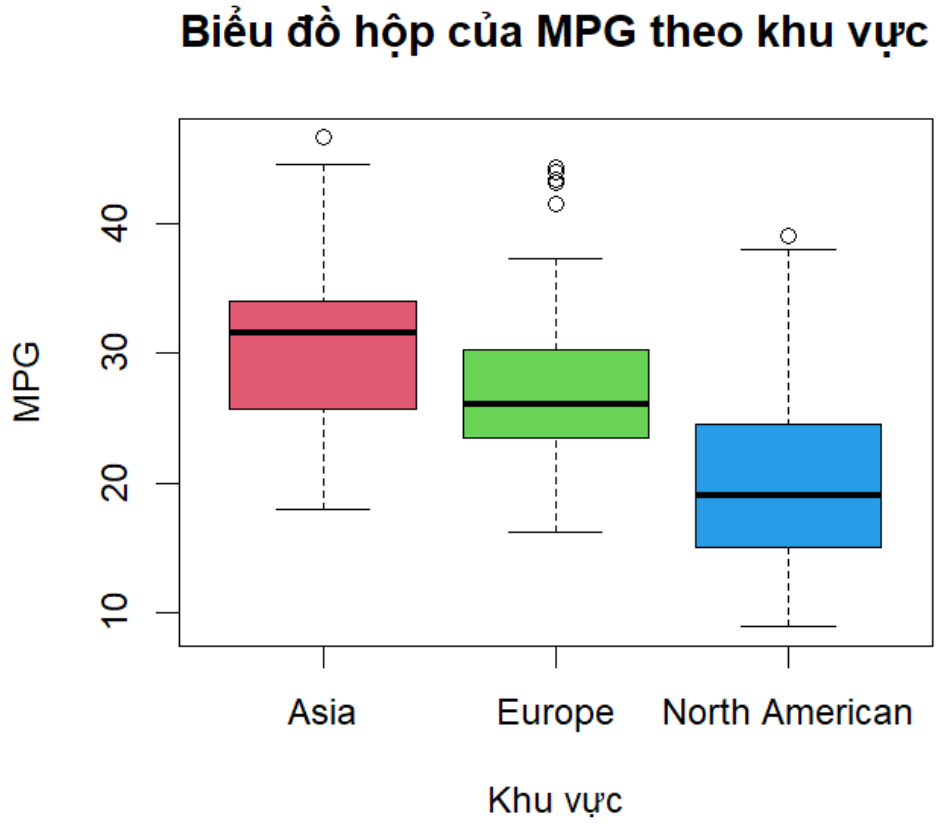
Hình 15: *Biểu đồ phân phối của MGP*

Nhận xét

- Lượng tiêu thụ nhiên liệu cao nhất là 15-20(miles/gallon)
- Lượng tiêu thụ nhiên liệu thấp nhất là ở hai đầu đồ thị 0-10 và 45–50 (miles/gallon)
- Người dùng ưa chuộng phân khúc giá 15-20 và giảm dần ở hai đầu

Biểu đồ hộp cho MPG theo từng khu vực

```
> boxplot(clearData$mgp ~ clearData$origin, main = "Biểu đồ hộp của MPG theo khu vực",
+         ylab = "MPG", xlab = "Khu vực", col = c(2, 3, 4))
```



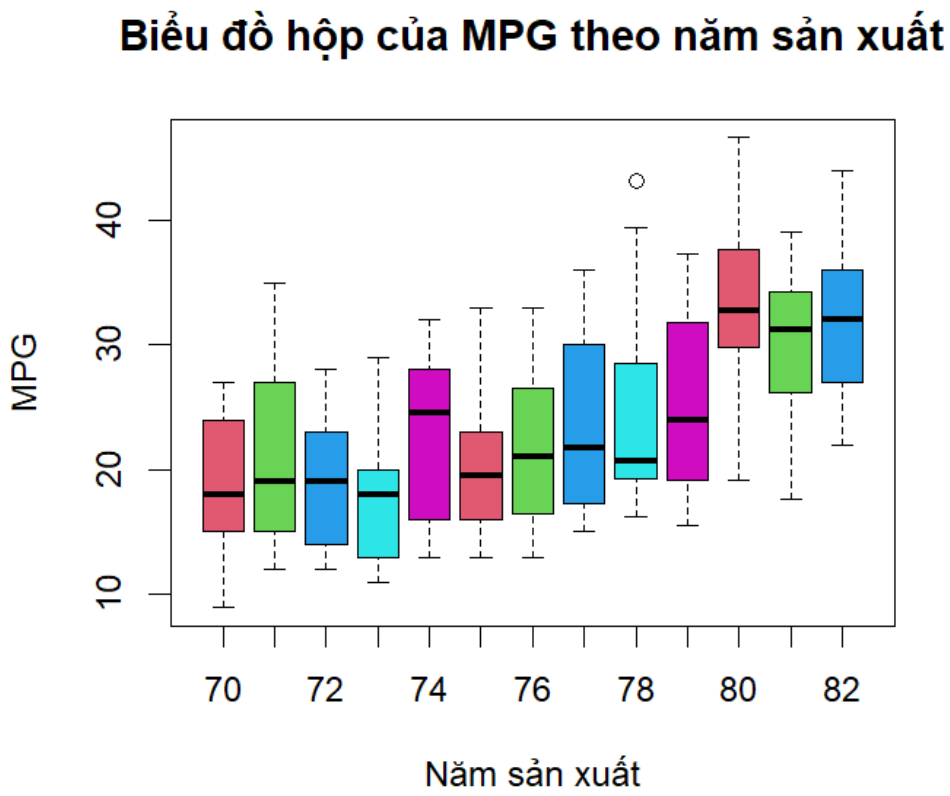
Hình 16: *Biểu đồ hộp cho MPG theo từng khu vực*

Nhận xét

- Nhìn chung mgp giới hạn từ 8 đến 45.
- Ở các mức khu vực tại Asia, Europe và North American đều có dữ liệu nằm ngoài vùng giới hạn, thể hiện rẳng ở đây có khá nhiều biến động.
- Khi origin = Asia, trung vị xấp xỉ 30 miles/gallon. Trực quan ta đếm được có 1 điểm outliers nằm phía trên giá trị lớn nhất, miền phân bố dữ liệu rộng. Gía trị lớn nhất nằm xấp xỉ 40 miles/gallon nhưng cao hơn 40 miles/gallon. Gía trị nhỏ nhất nằm xấp xỉ 20 miles/gallon nhưng thấp hơn 20 miles/gallon.
- Khi origin = Europe, trung vị nằm giữa 20 miles/gallon và 30 miles/gallon. Trực quan ta thấy có 5 điểm outliers, miền phân bố dữ liệu rộng. Gía trị lớn nhất nằm xấp xỉ 40 miles/gallon nhưng lớn hơn giá trị lớn nhất của mgp khi số xy-lanh =1. Gía trị nhỏ nhất xấp xỉ 20 miles/gallon.
- Khi origin = North American, trung vị xấp xỉ 20 miles/gallon. Trực quan ta thấy có 2 điểm outliers, miền phân bố dữ liệu rộng. Gía trị lớn nhất nằm xấp xỉ 40 miles/gallon. Gía trị nhỏ nhất xấp xỉ 10 miles/gallon.

Biểu đồ hộp cho MPG theo năm sản xuất

```
> boxplot(clearData$mgp ~ clearData$model_year, main = "Biểu đồ hộp của MPG theo năm sản xuất",
+ .         ylab = "MPG", xlab = "Năm sản xuất", col = c(2, 3, 4, 5, 6))
```



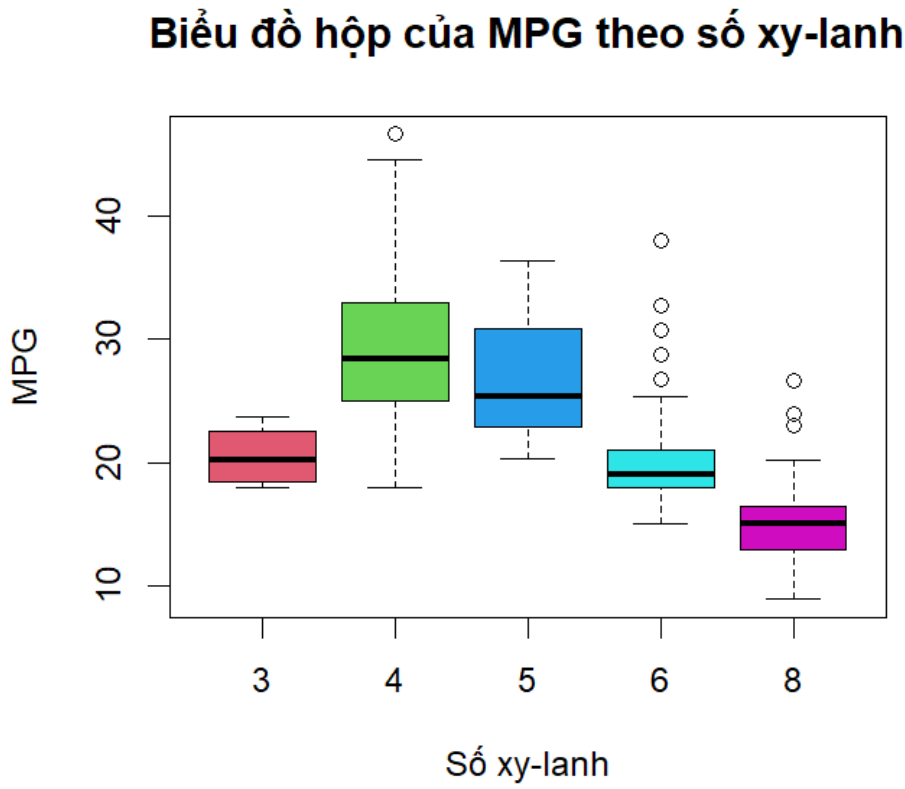
Hình 17: *Biểu đồ hộp cho MPG theo năm sản xuất*

Nhận xét

- Nhìn chung, mgp giới hạn từ 8 đến 45.
- Dựa vào mức trung vị, nhìn chung thể hiện rằng dữ liệu phân bố không đều. Không thể kết luận mgp sẽ tăng phụ thuộc theo năm sản xuất.
- Miền phân bố dữ liệu của biến mgp theo năm sản xuất rộng, giá trị lớn nhất và giá trị nhỏ nhất đều cách nhau hơn 10 đơn vị.
- Trung vị xấp xỉ 20 miles/gallon từ năm sản xuất có giá trị từ 70 đến 79. Trung vị xấp xỉ 30 miles/gallon từ năm sản xuất có giá trị từ 80 đến 82.
- Ở mức năm sản xuất có giá trị 78 có dữ liệu nằm ngoài vùng giới hạn, thể hiện rằng ở đây có khá nhiều biến động, nhìn chung mgp sẽ không ổn định.

Biểu đồ hộp cho MPG theo số xy-lanh

```
> boxplot(clearData$mgp ~ clearData$cylinders, main = "Biểu đồ hộp của MPG theo số xy-lanh",
+         ylab = "MPG", xlab = "Số xy-lanh", col = c(2, 3, 4, 5, 6))
```



Hình 18: *Biểu đồ hộp cho MPG theo số xy-lanh*

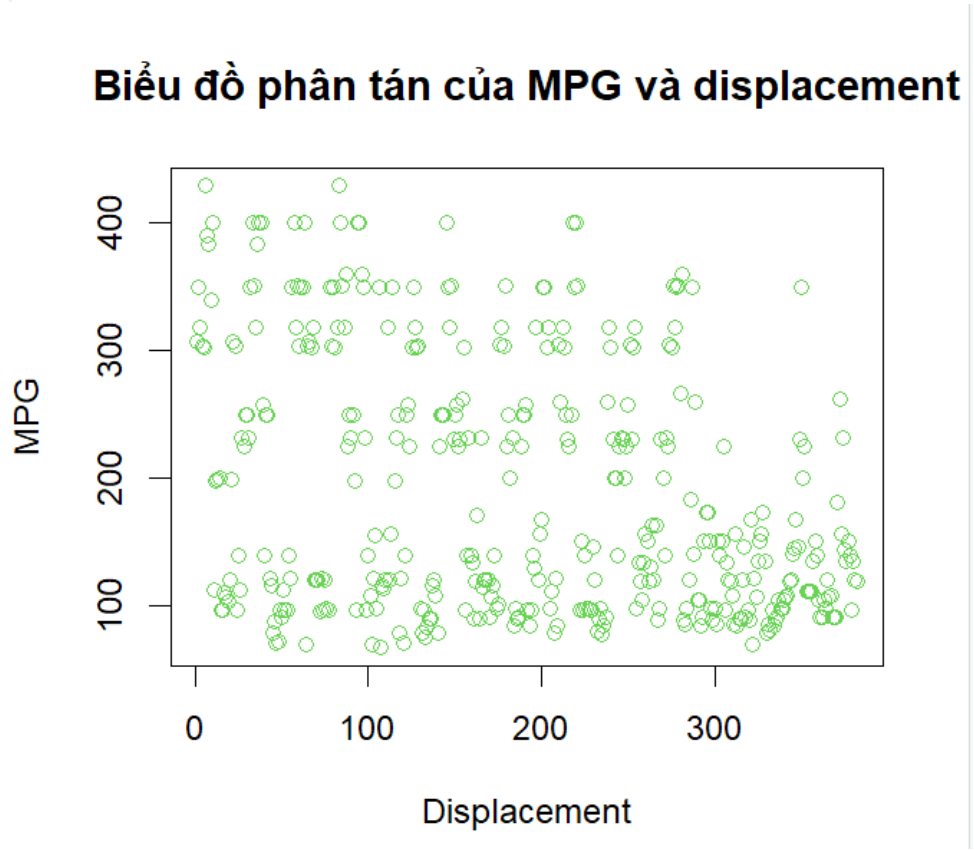
Nhận xét

- Nhìn chung, giới hạn mgp từ 10 đến 45.
- Dựa vào mức trung vị, ta có thể kết luận dữ liệu phân bố không đồng đều. Chưa thể kết luận được mgp và số xy-lanh có quan hệ gì với nhau.
- Khi số xy-lanh = 3, trung vị nằm xấp xỉ 20 miles/gallon. Không có điểm outliers nào, miền phân bố dữ liệu không quá rộng, giá trị lớn nhất và giá trị nhỏ nhất không quá xa nhau mà nằm xung quanh mức 20 miles/gallon.
- Khi số xy-lanh =4, trung vị nằm gần 30 miles/gallon hơn 20 miles/gallon. Có 1 điểm outlier nằm phía trên giá trị cực đại. Miền phân bố dữ liệu rộng khi giá trị lớn nhất cách giá trị nhỏ nhất hơn 20 đơn vị. Gía trị lớn nhất là cao hơn 40 miles/gallon nhưng có thể xấp xỉ 45 miles/gallon vì chưa nằm quá xa 40 miles/gallon. Gía trị nhỏ nhất thấp hơn 20 miles/gallon nhưng nằm không quá xa 20 miles/gallon nên có thể xấp xỉ 27 miles/gallon.
- Khi số xy-lanh = 5, trung vị nằm gần 30 miles/gallon hơn 20 miles/gallon nhưng nhỏ hơn trung vị khi số xy-lanh = 4 vì nằm thấp hơn. Không có điểm outliers nào, miền phân bố dữ liệu rộng. Gía trị lớn nhất nằm trong khoảng 40 miles/gallon đến 30 miles/gallon nhưng gần 40 miles/gallon hơn. Gía trị nhỏ nhất nằm gần 20 miles/gallon.
- Khi số xy-lanh = 6, trung vị nằm gần 20 miles/gallon nhưng nhỏ hơn trung vị khi số xy-lanh = 3 vì nằm thấp hơn. Trực quan ta đếm được có 5 điểm outliers nằm phía trên giá trị lớn nhất, miền phân bố dữ liệu không quá rộng. Gía trị lớn nhất nằm trong khoảng giữa 30 miles/gallon đến 20 miles/gallon. Gía trị nhỏ nhất nằm trong khoảng giữa 20 miles/gallon đến 10 miles/gallon nhưng gần 20 miles/gallon hơn.

- Khi số xy-lanh = 8, trung vị nằm giữa 20 miles/gallon và 10 miles/gallon nhưng xấp xỉ 15 miles/-gallon. Trực quan ta đếm được có 3 điểm outliers nằm phía trên giá trị lớn nhất, miền phân bố dữ liệu không quá rộng. Gía trị lớn nhất nằm xấp xỉ 20 miles/gallon. Gía trị nhỏ nhất nằm xấp xỉ 10 miles/gallon nhưng thấp hơn 10 miles/gallon.

2.3 Vẽ biểu đồ phân tán

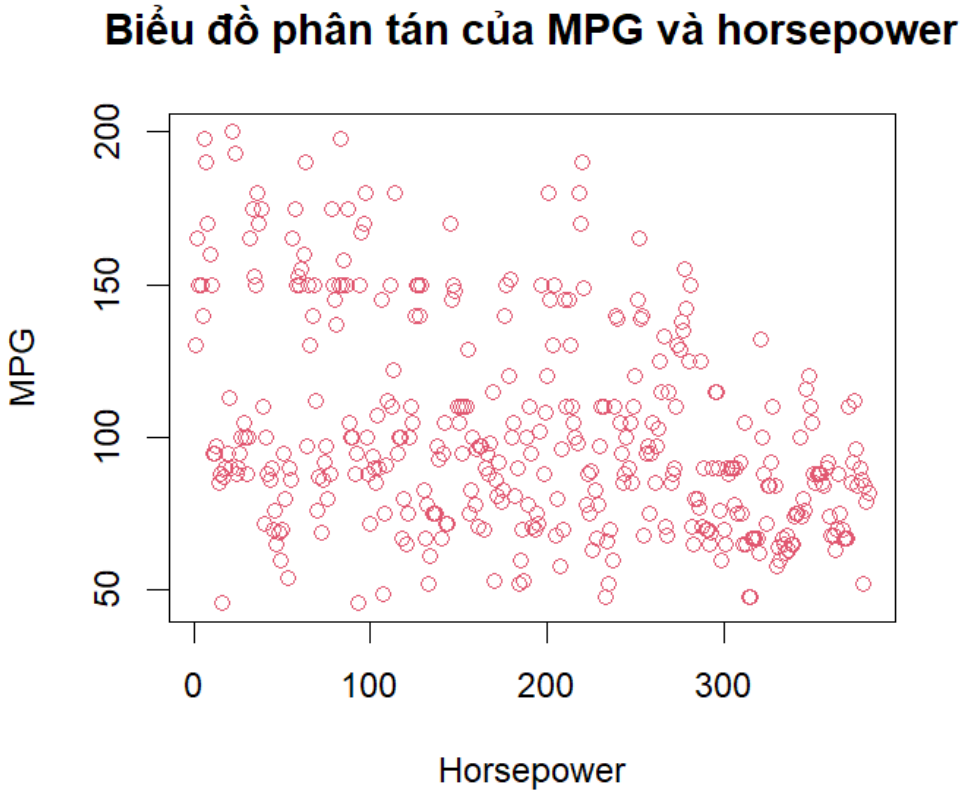
```
> plot(clearData$displacement, clearData$mpg, main = "Biểu đồ phân tán của MPG và displacement",
+      ylab = "MPG", xlab = "Displacement", col = c(3))
```



Hình 19: *MGP-displacement*

Vẽ biểu đồ phân tán thể hiện phân phối của MGP theo biến hourspower

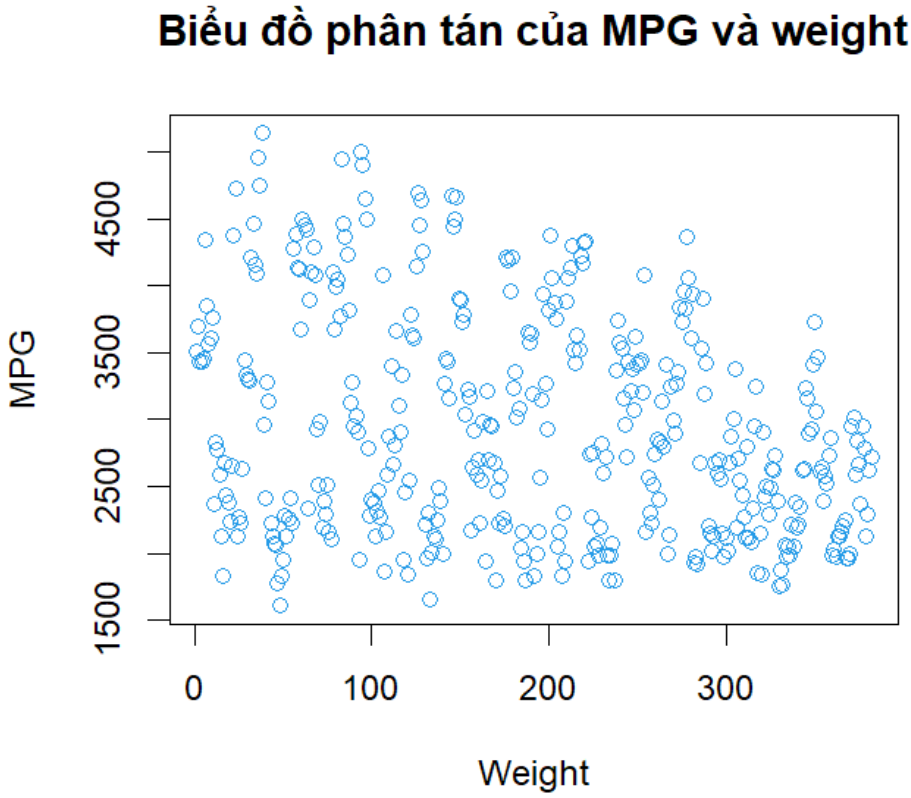
```
> plot(clearData$horsepower, clearData$mpg, main = "Biểu đồ phân tán của MPG và horsepower",
+      ylab = "MPG", xlab = "Horsepower", col = c(2))
```



Hình 20: *MGP-hourspower*

Vẽ biểu đồ phân tán thể hiện phân phối của MGP theo biến weight

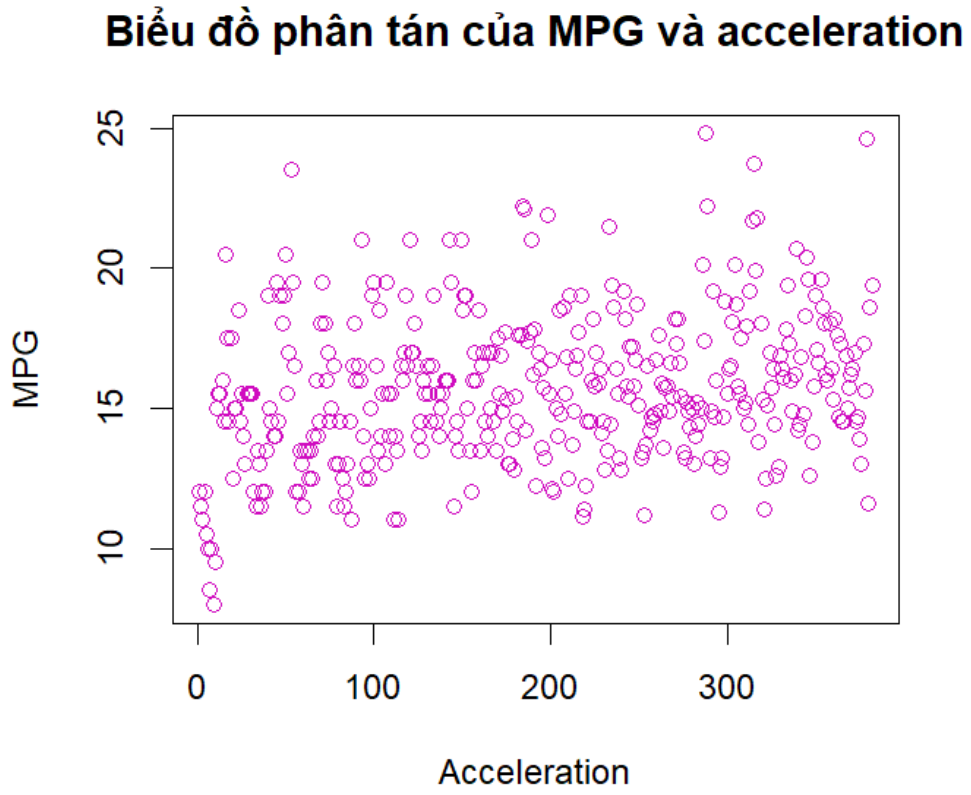
```
> plot(clearData$weight, clearData$mpg, main = "Biểu đồ phân tán của MPG và weight",
+      ylab = "MPG", xlab = "Weight", col = c(4))
```



Hình 21: *MGP-weight*

Vẽ biểu đồ phân tán thể hiện phân phối của MGP theo biến acceleration

```
> plot(clearData$acceleration, clearData$mpg, main = "Biểu đồ phân tán của MPG và acceleration",
       ylab = "MPG", xlab = "Acceleration", col = c(6))
```



Hình 22: *MGP-acceleration*

Nhận xét: Dựa trên các đồ thị phân tán, ta thấy mgp có mối quan hệ tuyến tính với các biến displacement, horsepower, weight (nghịch biến), tuy nhiên lại không quan hệ tuyến tính với biến acceleration. Ta có thể đoán displacement, horsepower, weight là các nhân tố ảnh hưởng đến mức tiêu thụ nhiên liệu, còn acceleration thì không ảnh hưởng đến mức tiêu thụ nhiên liệu.

3 Mô hình hồi quy và giả định thống kê

3.1 Tạo mẫu huấn luyện và mẫu kiểm tra

Tạo mẫu huấn luyện gồm 200 quan trắc, đặt tên là auto_mpg1. Vì trong dữ liệu có biến origin và biến car_name nằm ở dạng định tính nên phải đổi về dạng định lượng

```
auto_mpg1 <- auto_mpgData[1:200, ]
auto_mpg1$origin <- as.numeric(factor(auto_mpg1$origin))
auto_mpg1$car_name <- as.numeric(factor(auto_mpg1$car_name))
```

Tạo mẫu kiểm tra gồm các quan trắc còn lại, đã được làm sạch, đặt tên là auto_mpg2, đổi sang dạng định tính giống như trên.

```
remaining <- setdiff(seq_len(nrow(auto_mpgData)), 1:200)
auto_mgp2 <- clearData[remaining_indices, ]
auto_mgp2$origin <- as.numeric(factor(auto_mgp2$origin))
auto_mgp2$car_name <- as.numeric(factor(auto_mgp2$car_name))
```

3.2 Mô hình hồi quy tuyến tính bội với mẫu huấn luyện

Xét mô hình hồi quy với biến mpg là biến phụ thuộc và 8 biến còn lại là biến độc lập.

```
model_1 <- lm(mgp~., data=auto_mgp1)
summary(model_1)
```

```
Call:
lm(formula = mgp ~ ., data = auto_mgp1)

Residuals:
    Min      1Q  Median      3Q     Max
-9.3922 -1.1756 -0.0259  1.4042  6.5798

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  18.6090769  7.5930144   2.451 0.015153 *
cylinders    -0.2188659  0.3273373  -0.669 0.504542
displacement -0.0020076  0.0068313  -0.294 0.769162
horsepower   -0.0152579  0.0118553  -1.287 0.199650
weight       -0.0039120  0.0005871  -6.664 2.78e-10 ***
acceleration -0.0906468  0.0984641  -0.921 0.358417
model_year    0.2960249  0.0990683   2.988 0.003176 **
origin       -1.2656236  0.3547417  -3.568 0.000455 ***
car_name      0.0016344  0.0042283   0.387 0.699522
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.31 on 191 degrees of freedom
Multiple R-squared:  0.8504,    Adjusted R-squared:  0.8441
F-statistic: 135.7 on 8 and 191 DF,  p-value: < 2.2e-16
```

Nhận xét: Từ kết quả ta thấy rằng p value $< 2.2 \times 10^{-16}$ có ý nghĩa rất cao. Điều này chỉ ra rằng, ít nhất một biến dự báo trong mô hình có ý nghĩa giải thích rất cao cho biến mpg. Để xét ảnh hưởng cụ thể của từng biến độc lập, ta xét trọng số (hệ số βi) và p-value tương ứng:

- Cylinders: βi = -0.2188659 tức là biến này tỉ lệ nghịch với mpg, khi cylinder tăng 1 thì mpg giảm 0.22, p-value $>$ 0.05 nên không có ý nghĩa thống kê
- Displacement: βi = -0.0020076 , khi displacement tăng 1 thì mpg giảm 0.002 .p-value $>$ 0.05 nên không có ý nghĩa thống kê.
- Horsepower: βi = -0.0152579, khi horsepower tăng 1 thì mpg giảm 0.015 .p-value $>$ 0.05 nên không có ý nghĩa thống kê.
- Weight: βi = -0.0039120, khi weight tăng 1 thì mpg giảm 0.004. p-value = $2.78 \times 10^{-10} < 0.05$ nên có ảnh hưởng đến mpg.
- Acceleration: βi = -0.0906468, p-value = 0.358417 $>$ 0.05 nên không có ý nghĩa thống kê
- model_year βi = 0.2960249, p-value = 0.003176 $<$ 0.05 nghĩa là xe mới hơn 1 năm sẽ có mpg cao hơn khoảng 0.296
- origin : βi =-1.2656236, p-value = 0.000455 $<$ 0.05 nên có ảnh hưởng đến mpg, cụ thể là nếu origin tăng 1 thì mpg sẽ giảm đi khoảng -1.27.
- car_name βi =0.0016344, p-value = 0.699522 $>$ 0.05 nên không có ý nghĩa thống kê.

Qua những phân tích trên ta xây dựng mô hình hồi quy mới sau khi đã loại bỏ các biến không có ý nghĩa thống kê:

```
model_2 <- lm(mgp~weight+model_year+origin, data=auto_mgp1)

summary(model_2)
```

```
Call:
lm(formula = mgp ~ weight + model_year + origin, data = auto_mgp1)

Residuals:
   Min     1Q Median     3Q    Max
-9.162 -1.252  0.116  1.482  6.077

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.4211857  6.3574531   1.797    0.074 .
weight      -0.0049263  0.0002275 -21.650  < 2e-16 ***
model_year   0.3779158  0.0862399   4.382 1.91e-05 ***
origin      -1.3840332  0.2965299  -4.667 5.65e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.305 on 196 degrees of freedom
Multiple R-squared:  0.8471,    Adjusted R-squared:  0.8447
F-statistic: 361.8 on 3 and 196 DF,  p-value: < 2.2e-16
```

Như vậy mô hình hồi quy bội về ảnh hưởng của các biến đối với mpg được cho bởi:

$$\widehat{mpg} = 11.4211857 - 0.0049263 \times weight + 0.3779158 \times model\_year - 1.3840332 \times origin$$

$R^2 = 0.8471$ tức là 84.71% sự biến thiên của biến mpg là do weight, model_year và origin.
Khoảng tin cậy:

```
confint(model_2)
```

```
                  2.5 %       97.5 %
(Intercept) -1.116609529 23.958980936
weight      -0.005375019 -0.004477546
model_year   0.207838574  0.547992978
origin      -1.968832044 -0.799234264
```

Khoảng tin cậy 95% cho các hệ số hồi quy cho bởi:

$$-1.116609529 \leq \beta 0 \leq 23.958980936$$

$$0.005375019 \leq \beta 1 \leq -0.004477546$$

$$0.207838574 \leq \beta 2 \leq 0.547992978$$

$$-1.968832044 \leq \beta 3 \leq -0.799234264$$

3.3 Kiểm tra các giả định

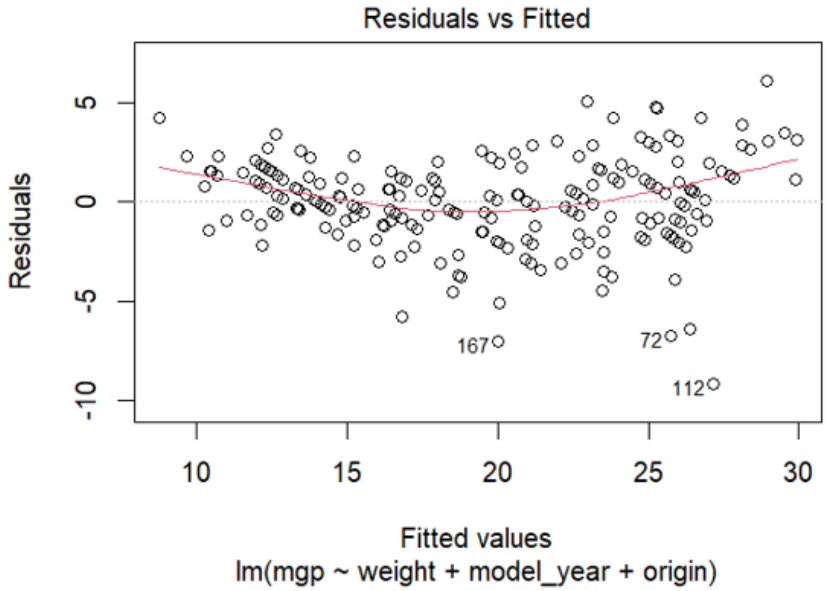
1 Tính tuyến tính của dữ liệu: mối quan hệ giữa biến dự báo X và biến phụ thuộc Y được giả sử là tuyến tính

2 Sai số có phân phối chuẩn

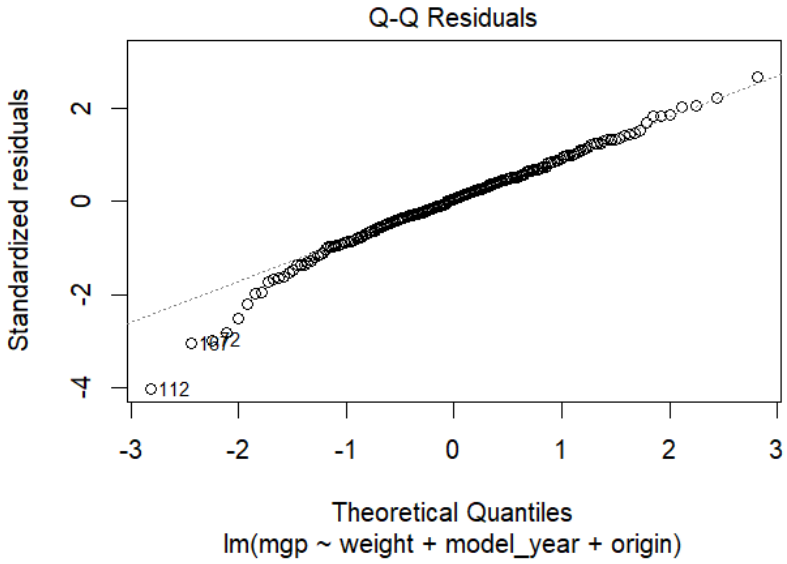
3 Phương sai của các sai số là hằng số: $\varepsilon i \sim N(0, \sigma 2)$

4 Các sai số $\varepsilon 1, \ldots, \varepsilon n$ thì độc lập với nhau

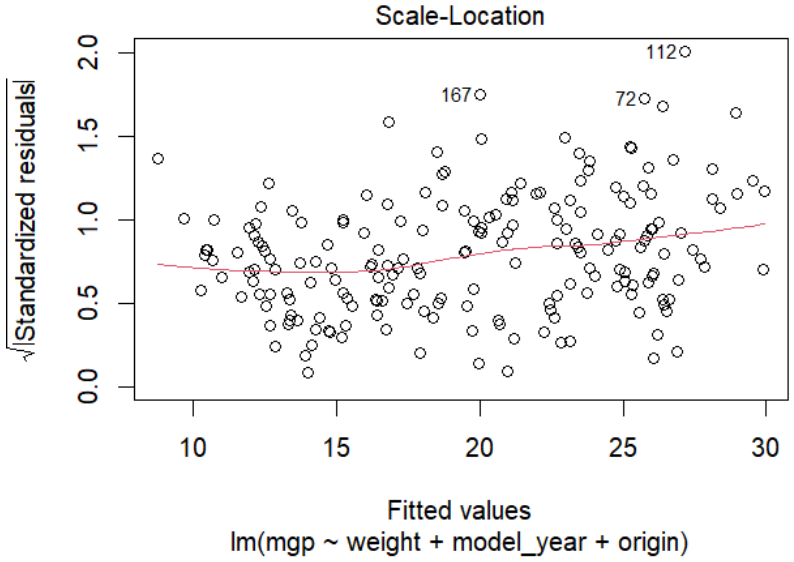
```
plot(model_2,1)
plot(model_2,2)
plot(model_2,3)
plot(model_2,5)
```



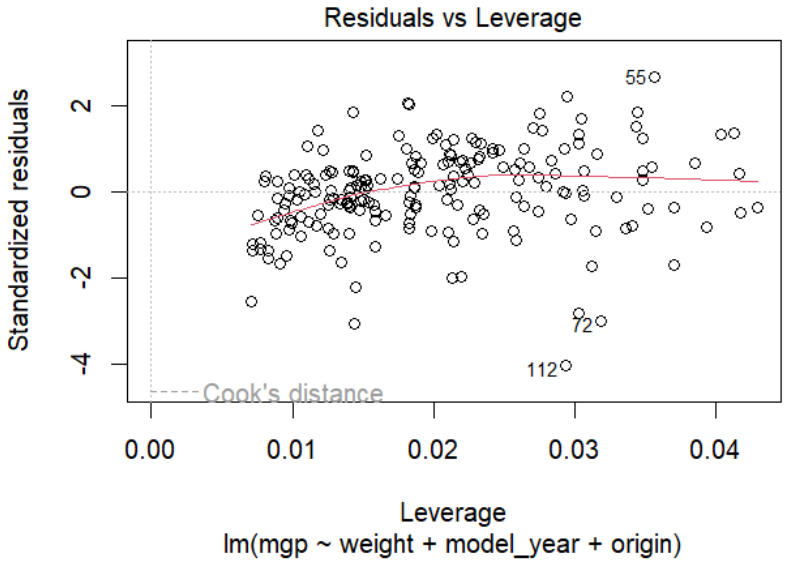
Đồ thị Residuals vs Fitted: Nhìn vào đồ thị ta thấy các điểm thặng dư không phân bố theo một hình mẫu nhất định nên giả định về tính tuyến tính (1) là thỏa. Tuy nhiên đường thẳng màu đỏ là đường cong chứ không phải đường thẳng nên giả định về tính tuyến tính là không thỏa mãn. Các điểm thặng dư phân tán khá đều nhau xung quanh đường thẳng y = 0 nên giả định về phương sai đồng nhất (3) được thỏa mãn.



Đồ thị thứ 2 Normal Q-Q, ta quan sát thấy hầu hết các điểm đều nằm trên đường thẳng nên điều kiện về phân phối chuẩn (2) được thỏa mãn.



Đồ thị thứ 3 Scale – Location. Ta nhận thấy đường màu đỏ tương đối thẳng và các điểm thặng dư phân bố khá đều 2 bên đường thẳng nên giả định phương sai là một hằng số (3) được củng cố.



Đồ thị thứ 4 Residuals vs Leverage. Đường thẳng màu đỏ trong hình được gọi là đường Cook's distance, ta quan sát thấy có các điểm 55, 72, 112 có thể là các điểm có ảnh hưởng cao trong bộ dữ liệu. Tuy nhiên ta có thể thấy các điểm này đều không vượt qua đường Cook nên không cần phải loại bỏ điểm nào cả.

3.4 Xây dựng mô hình hồi quy bội với mẫu kiểm tra

```
predict_mpg <- lm(mgp~weight+model_year+origin, data=auto_mgp2)
summary(predict_mpg)
```

```
confint(predict_mpg)
```

```
Call:
lm(formula = mgp ~ weight + model_year + origin, data = auto_mgp2)

Residuals:
    Min      1Q  Median      3Q     Max
-8.2108 -2.4754 -0.3059  1.8076 12.2146

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.449e+01  1.412e+01  -1.735   0.0844 .
weight      -7.983e-03  5.154e-04 -15.489  < 2e-16 ***
model_year   9.566e-01  1.716e-01   5.574 9.08e-08 ***
origin      -8.065e-01  3.827e-01  -2.107   0.0365 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.787 on 178 degrees of freedom
  (10 observations deleted due to missingness)
Multiple R-squared:  0.7625,    Adjusted R-squared:  0.7585
F-statistic: 190.5 on 3 and 178 DF,  p-value: < 2.2e-16
```

```
confint(predict_mpg)
```

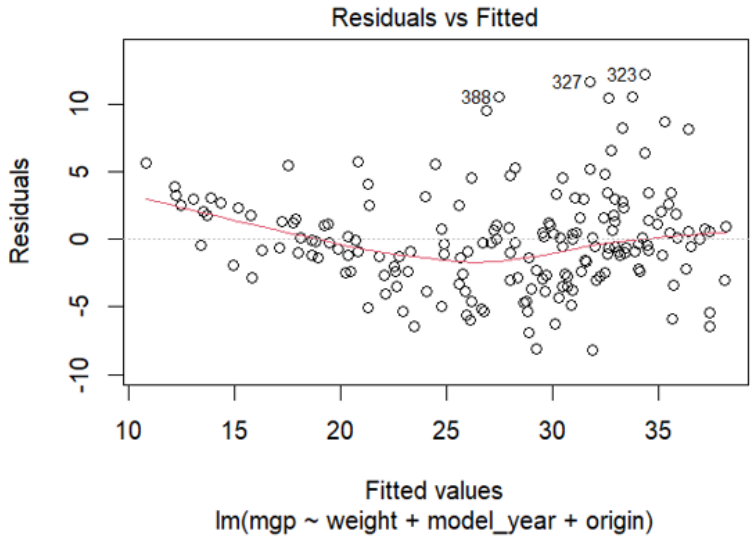
	2.5 %	97.5 %
(Intercept)	-52.349025241	3.360943856
weight	-0.008999905	-0.006965826
model_year	0.617961568	1.295285532
origin	-1.561852911	-0.051246167

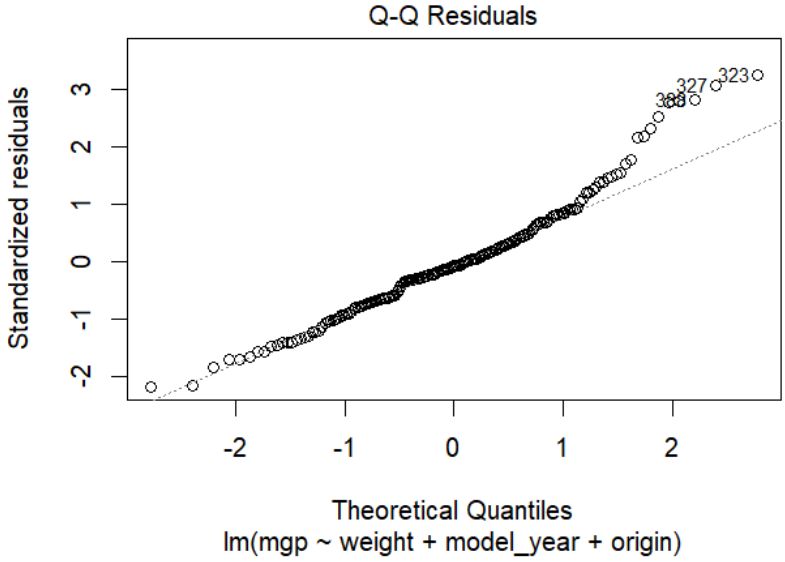
Ta so sánh các thông số, nhận thấy mô hình hoạt động tốt hơn trên dữ liệu huấn luyện so với dữ liệu kiểm tra, với độ chính xác cao hơn và sai số dự đoán thấp hơn.

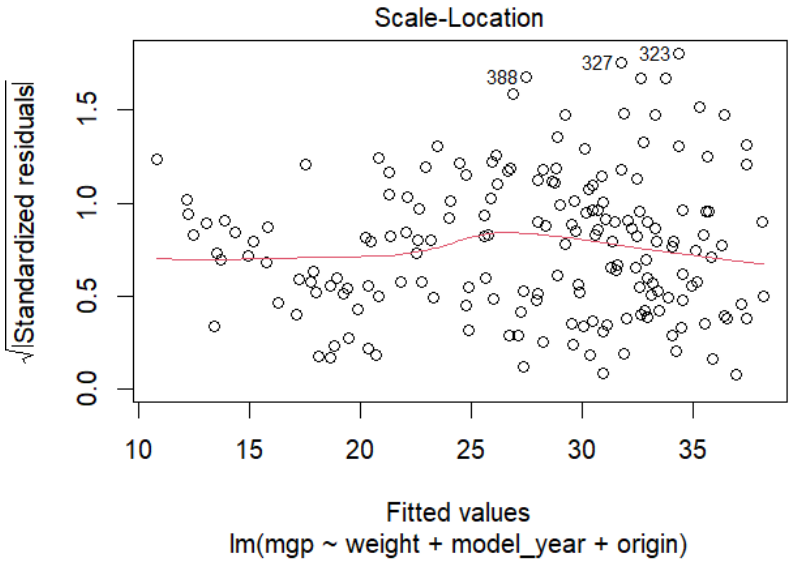
- Hệ số chặn (Intercept) trong mẫu kiểm tra (-24.49) khác biệt khá lớn so với mẫu huấn luyện(11.42).
- Cả hai mẫu đều cho thấy biến weight có hệ số âm và có ý nghĩa thống kê (p-value < 0.05), cho thấy khi trọng lượng tăng, biến mục tiêu giảm.
- Biến model_year có hệ số dương ở cả 2 mẫu và đều có ý nghĩa thống kê (p-value < 0.05)
- Cả hai mẫu đều cho thấy biến origin có hệ số âm và đều có ý nghĩa thống kê (p-value < 0.05
- R^2 của mẫu kiểm tra chỉ có 0.7625 thấp hơn 0.8471 của mẫu huấn luyện. Độ chính xác của mô hình trên mẫu huấn luyện cao hơn đáng kể so với mẫu kiểm tra
- Sai số chuẩn của hồi quy (Residual Standard Error) của mô hình trên mẫu kiểm tra (3.787) cao hơn nhiều so với mẫu huấn luyện (2.305), chỉ ra rằng mô hình dự đoán không chính xác bằng khi áp dụng lên dữ liệu kiểm tra.

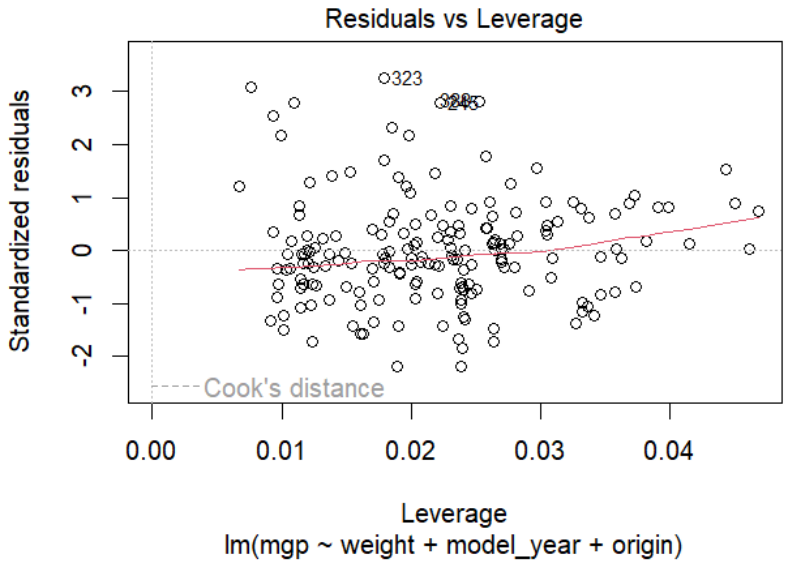
Ta xét các giả định của mẫu kiểm tra:

```
plot(predict_mpg,1)
plot(predict_mpg,2)
plot(predict_mpg,3)
plot(predict_mpg,5)
```









- Đồ thị Residuals vs Fitted cho thấy giả định 1 vi phạm còn giả định 3 có thể được chấp thuận
- Đồ thị Normal Q-Q Hầu hết các điểm đều nằm trên 1 đường thẳng nên giả định 2 thỏa
- Đồ thị Scale – Location cũng giống như mẫu kiểm tra, giả định 3 thỏa
- Đồ thị Residuals vs Leverage có 3 điểm 323, 245, 388 có ảnh hưởng cao, tuy nhiên cũng như mẫu kiểm tra, việc loại bỏ các điểm này là không cần thiết.

4 Kết luận

Bằng cách sử dụng mô hình hồi quy tuyến tính, nhóm đã hiểu thêm về mối quan hệ của các thông số về động cơ của xe trong thành phố như là mức tiêu thụ, kích thước động cơ hay là công suất động cơ, v.v. Từ đó nhóm nhận thấy rằng các yếu tố như là cân nặng, năm sản xuất ảnh hưởng lớn đến mức tiêu thụ

của động cơ.
Qua các biểu đồ thống kê, nhóm đã kiểm tra được các giả định thống kê, tìm ra được các giá trị ảnh hưởng đến kết quả thống kê. Với p-value là khá nhỏ và chỉ số R-squared cho thấy rằng mô hình trên có thể cho nhà sản xuất xem xét và dựa vào những thông tin ấy để có thể điều chỉnh các thông số tối ưu hóa động cơ.

III HOẠT ĐỘNG 2

1 Giới thiệu đề tài

Tập tin "SkillCraft1_Dataset.csv" chứa thông tin về game SkillCraft1, thu thập dữ liệu về nhiều người chơi như độ t
mức Rank, thời gian chơi, cách thức chơi,... và các dữ liệu đặc trưng khác trong game.
Dữ liệu gốc được cung cấp tại: https://archive.ics.uci.edu/dataset/272/skillcraft1+master+table+dataset.

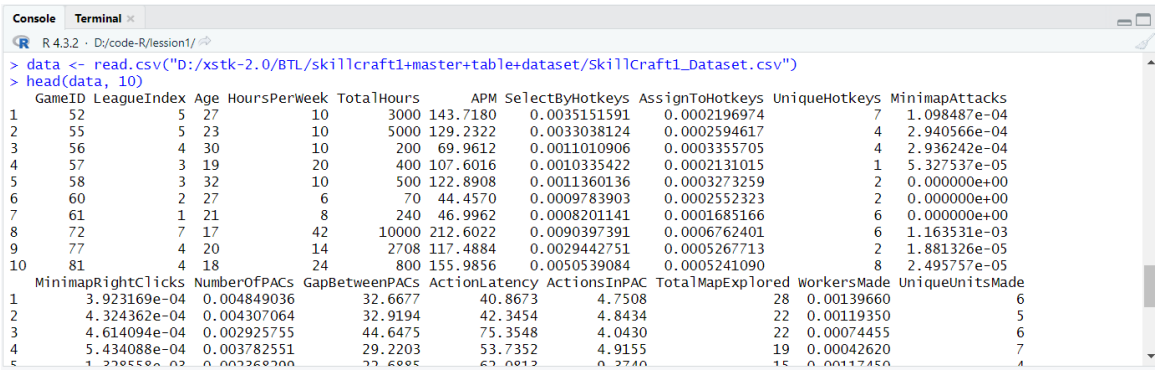
Các biến chính trong dữ liệu:

- LeagueIndex: có giá trị từ 1 đến 8 tương ứng với Bronze, Silver, Gold, Platinum, Diamond, Master, GrandMaster, and Professional
- Age: Tuổi của người chơi
- HoursPerWeek là số giờ chơi mỗi tuần
- TotalHours là Tổng thời gian chơi
- Action per minute là Số hành động mỗi phút
- UniqueHotkeys: là số phím nóng duy nhất sử dụng trong 1 khoảng thời gian
- ActionLatency là độ trễ của hành động
- TotalMapExplored là Tổng khu vực đã khám phá trong game

2 Tiền xử lý số liệu

2.1 Đọc dữ liệu

Dùng lệnh read.csv để đọc dữ liệu và lưu vào biến data và in ra trên console để kiểm tra



```
Console  Terminal
R 4.3.2 · D:/code-R/lession1/
> data <- read.csv("D:/xstk-2.0/BTL/skillcraft1+master+table+dataset/SkillCraft1_Dataset.csv")
> head(data, 10)
   GameID LeagueIndex Age HoursPerWeek TotalHours      APM SelectByHotkeys AssignToHotkeys UniqueHotkeys MinimapAttacks
1      52           5  27           10       3000 143.7180    0.0035151591    0.0002196974             7   1.098487e-04
2      55           5  23           10       5000 129.2322    0.0033038124    0.0002594617             4   2.940566e-04
3      56           4  30           10        200  69.9612    0.0011010906    0.0003355705             4   2.936242e-04
4      57           3  19           20        400 107.6016    0.0010335422    0.0002131015             1   5.327537e-05
5      58           3  32           10        500 122.8908    0.0011360136    0.0003273259             2   0.000000e+00
6      60           2  27            6         70  44.4570    0.0009783903    0.0002552323             2   0.000000e+00
7      61           1  21            8        240  46.9962    0.0008201141    0.0001685166             6   0.000000e+00
8      72           7  17           42      10000 212.6022    0.0090397391    0.0006762401             6   1.163531e-03
9      77           4  20           14       2708 117.4884    0.0029442751    0.0005267713             2   1.881326e-05
10     81           4  18           24        800 155.9856    0.0050539084    0.0005241090             8   2.495757e-05
   MinimapRightClicks NumberOfPACs GapBetweenPACs ActionLatency ActionsInPAC TotalMapExplored WorkersMade UniqueUnitsMade
1        3.923169e-04  0.004849036        32.6677       40.8673       4.7508               28  0.00139660               6
2        4.324362e-04  0.004307064        32.9194       42.3454       4.8434               22  0.00119350               5
3        4.614094e-04  0.002925755        44.6475       75.3548       4.0430               22  0.00074455               6
4        5.434088e-04  0.003782551        29.2203       53.7352       4.9155               19  0.00042620               7
```

Tiếp theo ta tách những cột dữ liệu mà ta quan tâm để thực hiện thống kê
Kiểm ta kiểu dữ liệu của Data:

2.2 Làm rõ dữ liệu

Kiểm tra số lượng dữ liệu bị khuyết của các biến

Dễ dàng quan sát: chỉ có 3 cột dữ liệu có giá trị bị khuyết là: "Age" , "HoursPerWeek", "TotalHours"
Tiếp theo, thay thế giá trị khuyết "?", thành "NA"

2.3 Thay thế giá trị bi khuyết

- Loại bỏ hàng chứa giá trị " NA ":
  Chuyển đổi các cột về dạng số vì 1 số biến mang kiểu dữ liệu "chr" thay vì kiểu "Numeric"
- Thay thế giá trị "NA" thành giá trị trung bình của cột

```
> data <- data[,c(2,3, 4,5,6, 9, 14, 16)]
```

	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	UniqueHotkeys	ActionLatency	TotalMapExplored
1	5	27	10	3000	143.7180	7	40.8673	28
2	5	23	10	5000	129.2322	4	42.3454	22
3	4	30	10	200	69.9612	4	75.3548	22
4	3	19	20	400	107.6016	1	53.7352	19
5	3	32	10	500	122.8908	2	62.0813	15
6	2	27	6	70	44.4570	2	98.7719	16
7	1	21	8	240	46.9962	6	90.5311	15
8	7	17	42	10000	212.6022	6	41.7671	45
9	4	20	14	2708	117.4884	2	46.4321	29
10	4	18	24	800	155.9856	8	52.1538	27
11	3	16	16	6000	153.8010	4	48.0711	24
12	4	26	4	190	79.2948	3	65.5000	19

```
> str(data)
'data.frame':   3395 obs. of  8 variables:
 $ LeagueIndex     : int  5 5 4 3 3 2 1 7 4 4 ...
 $ Age             : chr  "27" "23" "30" "19" ...
 $ HoursPerWeek    : chr  "10" "10" "10" "20" ...
 $ TotalHours      : chr  "3000" "5000" "200" "400" ...
 $ APM             : num  144 129 70 108 123 ...
 $ UniqueHotkeys   : int  7 4 4 1 2 2 6 6 2 8 ...
 $ ActionLatency   : num  40.9 42.3 75.4 53.7 62.1 ...
 $ TotalMapExplored: int  28 22 22 19 15 16 15 45 29 27 ...
>
```

```
> has_question_mark <- sapply(data, function(x) sum(x == "?", na.rm = TRUE))
> print(has_question_mark)
     LeagueIndex              Age     HoursPerWeek       TotalHours              APM    UniqueHotkeys    ActionLatency
               0               55               56               57                0                0                0
TotalMapExplored
               0

>
> print(na_count)
     LeagueIndex              Age     HoursPerWeek       TotalHours              APM    UniqueHotkeys    ActionLatency
               0               55               56                0                0                0                0
TotalMapExplored
               0
> mean_value <- mean(data$TotalHours, na.rm = TRUE)
> data$TotalHours[is.na(data$TotalHours)] <- mean_value
> mean_value <- mean(data$Age, na.rm = TRUE)
> data$Age[is.na(data$Age)] <- mean_value
> mean_value <- mean(data$HoursPerWeek, na.rm = TRUE)
> data$HoursPerWeek[is.na(data$HoursPerWeek)] <- mean_value

>
> print(na_count)
     LeagueIndex              Age     HoursPerWeek       TotalHours              APM    UniqueHotkeys    ActionLatency
               0               55               56                0                0                0                0
TotalMapExplored
               0
> mean_value <- mean(data$TotalHours, na.rm = TRUE)
> data$TotalHours[is.na(data$TotalHours)] <- mean_value
> mean_value <- mean(data$Age, na.rm = TRUE)
> data$Age[is.na(data$Age)] <- mean_value
> mean_value <- mean(data$HoursPerWeek, na.rm = TRUE)
> data$HoursPerWeek[is.na(data$HoursPerWeek)] <- mean_value

> na_count <- sapply(data, function(x) sum(is.na(x)))
> print(na_count)
     LeagueIndex              Age     HoursPerWeek       TotalHours              APM    UniqueHotkeys    ActionLatency
               0                0                0                0                0                0                0
TotalMapExplored
               0
```

- Cuối cùng kiểm tra lại xem còn giá trị khuyết nào không

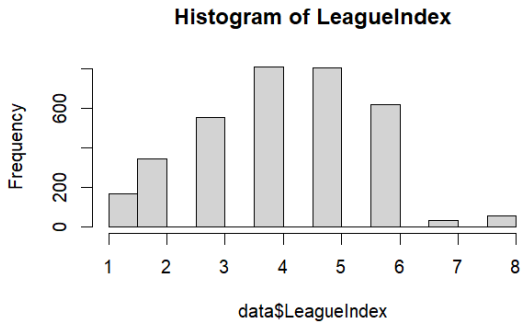
Nhận xét: Như vậy không còn dữ liệu nào trong "DATA" bị khuyết, ta chuyển sang bước tiếp theo

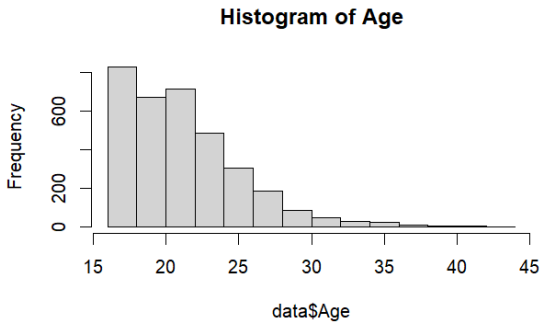
3 Thống kê tả

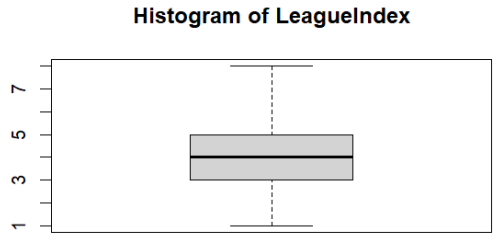
Sau khi làm sạch dữ liệu ta dùng lệnh "Summary" để xem tổng quan dữ liệu, kết quả thu được trả về giá trị nhỏ nhất (Min), điểm phân vị thứ nhất (1st Qu), trung vị (Median), giá trị trung bình (Mean), điểm phân vị thứ ba (3rd Qu) và giá trị lớn nhất (Max) của từng biến

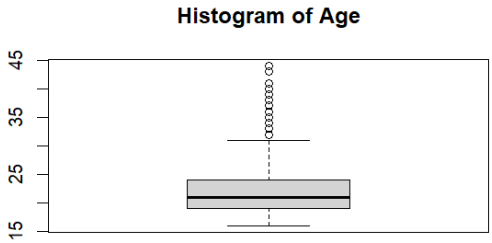
```
> summary(data)
  LeagueIndex         Age          HoursPerWeek      TotalHours           APM          UniqueHotkeys    ActionLatency   
 Min.   :1.000   Min.   :16.00   Min.   :  0.00   Min.   :     3.0   Min.   : 22.06   Min.   : 0.000   Min.   : 24.09  
 1st Qu.:3.000   1st Qu.:19.00   1st Qu.:  8.00   1st Qu.:   300.0   1st Qu.: 79.90   1st Qu.: 3.000   1st Qu.: 50.45  
 Median :4.000   Median :21.00   Median : 12.00   Median :   500.0   Median :108.01   Median : 4.000   Median : 60.93  
 Mean   :4.184   Mean   :21.65   Mean   : 15.91   Mean   :   960.4   Mean   :117.05   Mean   : 4.365   Mean   : 63.74  
 3rd Qu.:5.000   3rd Qu.:24.00   3rd Qu.: 20.00   3rd Qu.:   800.0   3rd Qu.:142.79   3rd Qu.: 6.000   3rd Qu.: 73.68  
 Max.   :8.000   Max.   :44.00   Max.   :168.00   Max.   :1000000.0   Max.   :389.83   Max.   :10.000   Max.   :176.37  
 TotalMapExplored
 Min.   : 5.00   
 1st Qu.:17.00   
 Median :22.00   
 Mean   :22.13   
 3rd Qu.:27.00   
 Max.   :58.00   
```

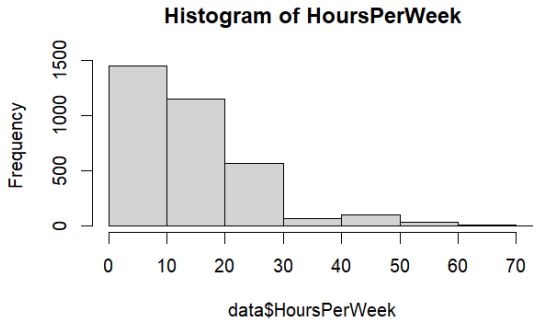
Tiếp theo, nhóm phân tích dữ liệu của biến bằng cách trực quan hóa dữ liệu qua hai dạng biểu đồ là biểu đồ tần suất (histogram) và biểu đồ hộp (boxplot) cho các biến liên tục. Biểu đồ tần số sẽ cho thấy cái nhìn tổng quan về phân phối của biến. Biểu đồ hộp giúp biểu diễn rõ ràng các đại lượng quan trọng của biến như giá trị lớn nhất, nhỏ nhất, điểm phân vị,... (Vì có nhiều biến nên nhóm lựa chon những biến hợp lí nhất)

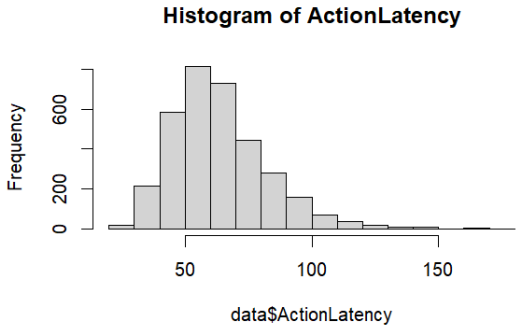


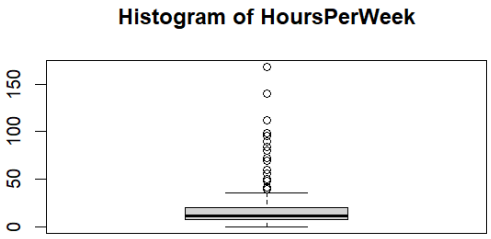


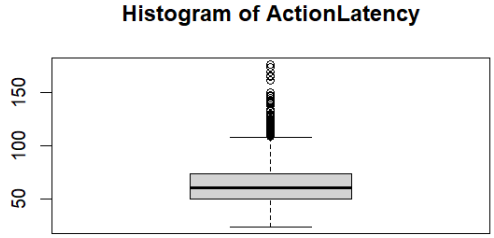


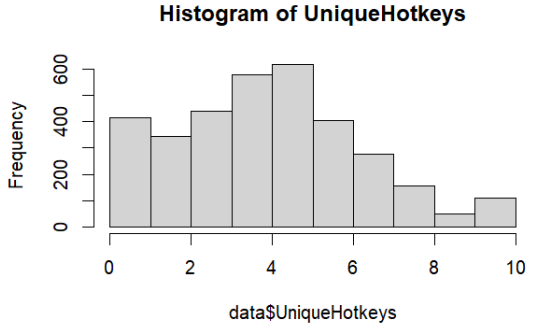


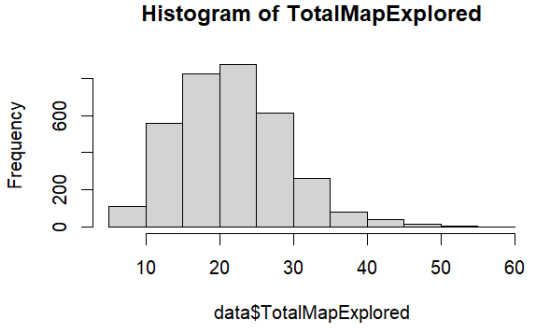


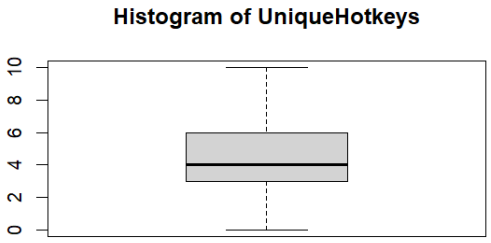


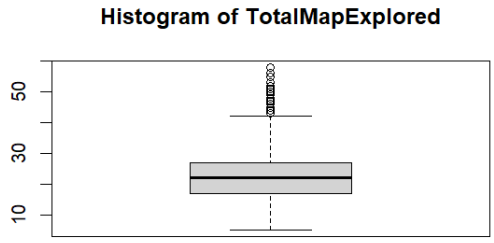












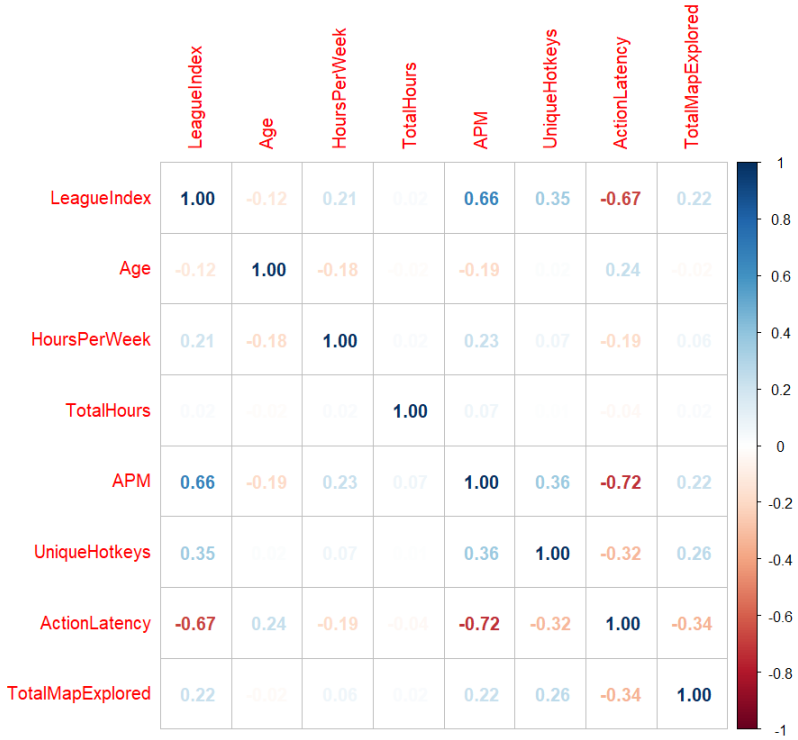
Qua dữ liệu thu được và quan sát từ biểu đồ ta có thể đưa ra các nhận xét như sau:

- LeagueIndex: có giá trị từ 1 đến 8 tương ứng với Bronze, Silver, Gold, Platinum, Diamond, Master, GrandMaster, and Professional, trong đó giá trị 4 có số lượng lớn nhất với 811 lần, với trung bình là 4.184 chứng tỏ biến "LeagueIndex" có hình dáng phân phối đối xứng.
- Age: (Tuổi) có giá trị từ 16 đến 44, trong đó giá trị 16 có số lượng lớn nhất với 256 lần xuất hiện, có giá trị trung bình là 21.65 chứng tỏ biến "Age" có hình dáng phân phối đối lệch đối xứng.
- HoursPerWeek:là số giờ chơi mỗi tuần có giá trị từ 0 đến 168 trong đó giá trị 6 có số lần xuất hiện

với 323 lần, có giá trị trung bình là 15,91 chứng tỏ biến "HoursPerWeek" có hình dáng phân phối đối lệch đối xứng.

- TotalHours: Tổng thời gian chơi, có giá trị từ 3 - 25000 giờ có giá trị trung bình là 960.4 vậy có thể kết luận "TotalHours" có hình dáng phân phối đối lệch đối xứng.
- Action per minute: Số hành động mỗi phút có giá trị từ 22 đến 390, với giá trị trung bình là 117.05 chứng tỏ biến " Action per minute" có hình dáng phân phối đối xứng.
- UniqueHotkeys: là số phím nóng duy nhất sử dụng trong 1 khoảng thời gian có giá trị từ 0 - 10, có mốt là 5 với số lần xuất hiện là 617, giá trị trung bình là 4.365 chứng tỏ biến " UniqueHotkeys" có hình dáng phân phối đối xứng.
- ActionLatency: là độ trễ của hành động, có giá trị từ 24 đến 176, giá trị trung bình là 63,74 chứng tỏ biến " ActionLatency" có hình dáng phân phối đối xứng.
- TotalMapExplored: là Tổng khu vực đã khám phá, có giá trị từ 5 đến 58, giá trị trung bình là 22.13 với giá trị 23 xuất hiện nhiều nhất là 191 lần chứng tỏ biến "TotalMapExplored" có hình dáng phân phối đối xứng.

Tiếp theo, nhóm thực hiện vẽ biểu đồ tương quan và lập bảng hệ số tương quan giữa các biến được chọn từ data nhằm trực quan hoá sự phụ thuộc tuyến tình giữa các biến



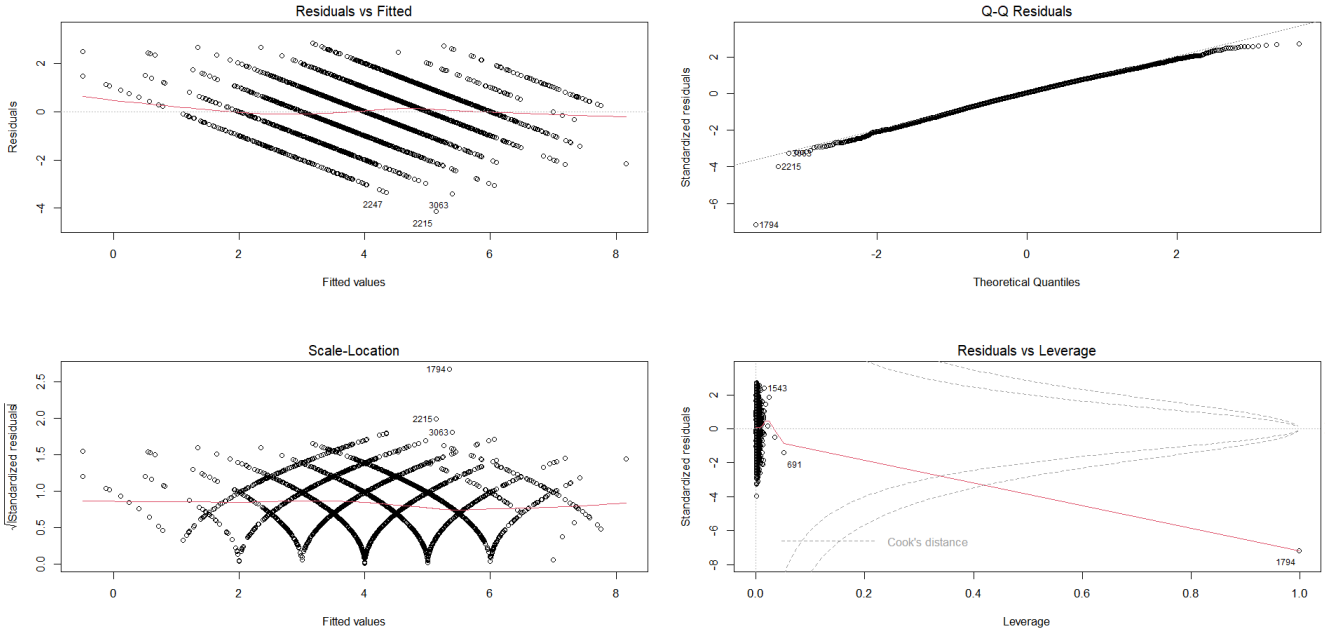
	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	UniqueHotkeys	ActionLatency	TotalMapExplored
LeagueIndex	1.00	-0.12	0.21	0.02	0.66	0.35	-0.67	0.22
Age	-0.12	1.00	-0.18	-0.02	-0.19	0.02	0.24	-0.02
HoursPerWeek	0.21	-0.18	1.00	0.02	0.23	0.07	-0.19	0.06
TotalHours	0.02	-0.02	0.02	1.00	0.07	0.03	-0.04	0.02
APM	0.66	-0.19	0.23	0.07	1.00	0.36	-0.72	0.22
UniqueHotkeys	0.35	0.02	0.07	0.03	0.36	1.00	-0.32	0.26
ActionLatency	-0.67	0.24	-0.19	-0.04	-0.72	-0.32	1.00	-0.34
TotalMapExplored	0.22	-0.02	0.06	0.02	0.22	0.26	-0.34	1.00

Ta có thể thấy các biến có tương quan mạnh như là: LeagueIndex, Action Per Minute, UniqueHotkeys, Action Latency. Tiếp theo các biến có tương quan trung bình như Hours Per Week, Total Map Explore. Cuối cùng là các biến có tương quan yếu: Age, TotalHours. Qua những thông tin trên, ta thấy biến LeagueIndex có tương quan mạnh với các nhóm còn lại, chính vì thế sự phụ thuộc tuyến tính của "LeagueIndex" với các thành phần còn lại sẽ được khảo sát.

4 Mô hình hồi quy và dự đoán

4.1 Kiểm định

Ta tiến hành kiểm định phần dư của mô hình:



Kết luận: Từ 4 biểu đồ trên, ta suy ra được phần dư trong mô hình có dạng phân phối chuẩn.Ta chuyển qua bước tiếp theo.

4.2 Quan hệ tuyến tính

```
> summary(model)

Call:
lm(formula = LeagueIndex ~ Age + HoursPerWeek + TotalHours +
    APM + UniqueHotkeys + ActionLatency + TotalMapExplored, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-4.1403 -0.6772  0.0516  0.7333  2.8274

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       4.439e+00  1.804e-01  24.606  < 2e-16 ***
Age               1.767e-02  4.476e-03   3.949 8.01e-05 ***
HoursPerWeek      7.542e-03  1.560e-03   4.834 1.40e-06 ***
TotalHours       -1.329e-06  1.041e-06  -1.276   0.2019
APM               9.671e-03  5.138e-04  18.823  < 2e-16 ***
UniqueHotkeys     6.418e-02  8.323e-03   7.711 1.63e-14 ***
ActionLatency    -3.243e-02  1.412e-03 -22.970  < 2e-16 ***
TotalMapExplored -4.559e-03  2.604e-03  -1.751   0.0801 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.039 on 3387 degrees of freedom
Multiple R-squared:  0.5325,    Adjusted R-squared:  0.5315
F-statistic: 551.2 on 7 and 3387 DF,  p-value: < 2.2e-16
```

Từ kết quả trên, ta có thể thấy R-squared = 0.5325. Đồng nghĩa với việc phần lớn sự phụ thuộc của biến LeagueIndex có thể được giải thích bởi đa số các biến độc lập trong mô hình

4.3 Mô hình hồi quy tuyến tính

Trong phần này, nhóm sẽ xây dựng mô hình hồi quy tuyến tính bội với LeagueIndex là biến phụ thuộc và các biến còn lại là biến độc lập. Mục tiêu của nhóm là điều tra mối quan hệ giữa LeagueIndex và các độc lập này các biến và để phát triển một mô hình dự đoán ước tính chính xác Cache dựa trên các biến này các nhân tố. Phân tích này sẽ cung cấp những hiểu biết có giá trị về các yếu tố tác động đáng kể hiệu suất hệ thống máy tính và tạo điều kiện tối ưu hóa cấu hình hệ thống trong tương lai. Xây dựng mô hình thống kê bằng cách sử dụng hàm Lm trong R. Hàm lm cho phép ước tính mối quan hệ giữa một biến phụ thuộc và một hoặc nhiều biến độc lập.
Kiểm định giả thuyết thống kê

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

Với mức ý nghĩa (significance level) là 5%, nhân tố nào có Pr(>t) lớn hơn 0.05 sẽ không đạt mức ý nghĩa thống kê và không được coi là có ảnh hưởng đáng kể đến giá nhà. Từ bảng kết quả của mô hình hồi quy tuyến tính, Pr(>t) của condition2 lớn hơn 0.05, vì vậy ta chấp nhận giả thuyết H cho nhân tố condition2. Tức là tất cả các nhân tố trừ nhân tố condition2 được giữ lại trong mô hình và được coi là có ảnh hưởng đáng kể đến biến LeagueIndex.

```
> summary(model)

Call:
lm(formula = LeagueIndex ~ Age + HoursPerWeek + TotalHours +
    APM + UniqueHotkeys + ActionLatency + TotalMapExplored, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-4.1403 -0.6772  0.0516  0.7333  2.8274

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       4.439e+00  1.804e-01  24.606  < 2e-16 ***
Age               1.767e-02  4.476e-03   3.949 8.01e-05 ***
HoursPerWeek      7.542e-03  1.560e-03   4.834 1.40e-06 ***
TotalHours       -1.329e-06  1.041e-06  -1.276   0.2019
APM               9.671e-03  5.138e-04  18.823  < 2e-16 ***
UniqueHotkeys     6.418e-02  8.323e-03   7.711 1.63e-14 ***
ActionLatency    -3.243e-02  1.412e-03 -22.970  < 2e-16 ***
TotalMapExplored -4.559e-03  2.604e-03  -1.751   0.0801 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.039 on 3387 degrees of freedom
Multiple R-squared:  0.5325,    Adjusted R-squared:  0.5315
F-statistic: 551.2 on 7 and 3387 DF,  p-value: < 2.2e-16
```

Ta có thể nhận định về mô hình hồi quy như sau:

- Residuals: Các giá trị dư cho thấy sự phân bố của sai số dự đoán. Các giá trị dư dao động từ -4.1403 đến 2.8274, với giá trị trung vị gần bằng 0 (0.0516), cho thấy mô hình có thể dự đoán khá tốt.
- Intercept: Hệ số chặn (Intercept) là 4.394 với giá trị p rất nhỏ (<2e-16), cho thấy nó có ý nghĩa thống kê cao.
- Age: Hệ số của Age là -0.06451 với giá trị p <2e-16, cho thấy tuổi có ảnh hưởng tiêu cực và có ý nghĩa thống kê đến LeagueIndex.
- HoursPerWeek: Hệ số là 0.01916 với giá trị p <2e-16, cho thấy số giờ chơi mỗi tuần có ảnh hưởng tích cực và có ý nghĩa thống kê.
- TotalHours: Hệ số là -7.676e-05 với giá trị p <2e-16, cho thấy tổng số giờ chơi có ảnh hưởng tiêu cực và có ý nghĩa thống kê.
- APM: Hệ số của APM (Actions Per Minute) bị cắt ngắn trong hình, nhưng có vẻ như nó cũng có ý nghĩa thống kê cao.

Kết luân:

- Residual Standard Error: Sai số chuẩn của phần dư là 1.039, cho thấy mức độ sai số của mô hình.
- Multiple R-squared: Giá trị R-squared là 0.5325, cho thấy khoảng 53.25% biến thiên của LeagueIndex được giải thích bởi các biến độc lập trong mô hình.
- Adjusted R-squared: Giá trị R-squared điều chỉnh là 0.5315, điều chỉnh cho số lượng biến trong mô hình
- F-statistic: Giá trị F-statistic là 551.2 với giá trị p <2.2e-16, cho thấy mô hình tổng thể có ý nghĩa thống kê.

4.4 Dùng mô hình để dự đoán

Để dự đoán dữ liệu trong tương lai, nhóm sử dụng hàm predict()

```
> # Dự đoán giá trị LeagueIndex dựa trên dữ liệu hiện tại
> predictions <- predict(model, newdata = data)
> head(predictions)
       1        2        3        4        5        6
5.373538 4.946962 3.433246 4.200330 4.314126 2.243077
```

Giả sử ta có thêm các giá trị mới (new data frame) chứa giá trị của các biến độc lập

```
> # Ví dụ về dữ liệu mới
> new_data <- data.frame(
+   Age = c(25, 30, 22),
+   HoursPerWeek = c(15, 20, 35),
+   TotalHours = c(2000, 5000, 300),
+   APM = c(100, 120, 85),
+   UniqueHotkeys = c(5, 7, 4),
+   ActionLatency = c(50, 45, 60),
+   TotalMapExplored = c(20, 25, 18)
+ )
>
> # Dự đoán giá trị LeagueIndex cho dữ liệu mới
> new_predictions <- predict(model, newdata = new_data)
>
> # Hiển thị các dự đoán
> new_predictions
       1        2        3
4.566337 5.149584 4.141950
>
```

5 Kết luận

Trong bài này, nhóm đã sử dụng mô hình hồi quy tuyến tính để nghiên cứu mối quan hệ giữa các yếu tố như tuổi, số giờ chơi mỗi tuần, tổng số giờ chơi, và một số chỉ số liên quan đến kỹ năng chơi game với thứ hạng trong trò chơi (LeagueIndex).

Mô hình hồi quy tuyến tính cung cấp tầm quan trọng và ảnh hưởng của từng biến độc lập đối với thứ hạng của người chơi. Các kết quả cho thấy rằng một số yếu tố như APM (Actions Per Minute) và UniqueHotkeys có tác động đáng kể đến LeagueIndex, trong khi các yếu tố khác cũng có mức độ ảnh hưởng khác nhau.

Qua các biểu đồ chẩn đoán, chúng ta kiểm tra được tính hợp lý của mô hình, phát hiện các giá trị ngoại lai và kiểm tra giả thuyết về phân phối của residuals. Kết quả phân tích cho thấy mô hình có ý nghĩa thống kê với giá trị p-value rất nhỏ, tuy nhiên, chỉ số R-squared chỉ ra rằng mô hình chỉ giải thích được một phần biến thiên của thứ hạng, gợi ý rằng còn nhiều yếu tố khác chưa được xem xét trong phân tích này. Mô hình hồi quy tuyến tính đã giúp ta hiểu rõ hơn về các yếu tố ảnh hưởng đến thứ hạng của người chơi trong trò chơi.

IV Tài liệu tham khảo

[1] Douglas C. Montgomery, George C. Runger, *Applied Statistics and Probability for Engineers, 6th Edition*, Năm xuất bản 2014.

[2] Link tổng hợp code : https://github.com/Duckphuong/XSTK/blob/main/12_6.R