ĐẠI HỌC QUỐC GIA TP.HCM

TRƯỜNG ĐẠI HỌC BÁCH KHOA

-----o0o-----

BÁO CÁO BÀI TẬP LỚN

MÔN: XÁC SUẤT THỐNG KÊ MT2013

Giáo viên hướng dẫn: Nguyễn Đình Huy
Lớp – Nhóm: L03 – 12
Đề tài: 6

Sinh viên thực hiện	***MSSV***	***Khoa***
Nguyễn Đặng Hà My	2013803	Kỹ thuật Hóa học
Nguyễn Thị Hoài My	2013805	Khoa học và Kỹ thuật Máy tính
Nguyễn Phương Nam	2013828	Cơ khí
Hoàng Bá Nhật	2013993	Cơ khí
Từ Hoàng Phiếm	2014112	Khoa học và Kỹ thuật Máy tính
Tăng Văn Minh	2013787	Khoa học và Kỹ thuật Máy tính

TP. HCM, tháng 04/2022

Bảng phân công công việc

Họ và tên	Nhiệm vụ	Đánh giá
Từ Hoàng Phiếm	R Code câu 1, 2, 3	100%
Nguyễn Thị Hoài My	R Code câu 4, 5	100%
Tăng Văn Minh	R Code câu 6, 7	100%
Hoàng Bá Nhật	Báo cáo câu 4, 5, 6	100%
Nguyễn Phương Nam	Báo cáo cơ sở lí thuyết, câu 7	100%
Nguyễn Đặng Hà My	Báo cáo câu 1, 2, 3 – Tổng hợp báo cáo	100%

Mục lục

1. ĐỘNG CƠ NGHIÊN CỨU & MỤC TIÊU

1.1. Động cơ nghiên cứu

Hiện nay, chúng ta đang sống trong thời đại công nghệ 4.0, các công việc liên quan đến máy tính đang trên đà phát triển vượt bậc. Trong đó, lĩnh vực nghiên cứu và phân tích dữ liệu đang là xu hướng mà nhiều người quan tâm đến. Phân tích và nghiên cứu dữ liệu được hiểu là quá trình thu thập, sàng lọc, chuyển đổi và mô hình hóa dữ liệu nhằm mục đích thu thập được các thông tin cần thiết, các kết quả khảo sát. Từ đó, việc phân tích dữ liệu giúp ta có những dự đoán một cách khoa học. Việc nghiên cứu và phân tích dữ liệu được sử dụng trên nhiều lĩnh vực, chủ yếu là ứng dụng trong kinh tế. Bên cạnh đó, đối với các lĩnh vực liên quan đến khoa học và xã hội thì việc phân tích dữ liệu đóng góp vai trò không hề nhỏ.

1.2. Mục tiêu

Trong bài tập lớn này, chúng ta sẽ bắt đầu với các bài toán thống kê đơn giản từ những dữ liệu được cung cấp sẵn. Qua đó, chúng ta sẽ tìm ra những con số thú vị, có ý nghĩa đối với các dữ liệu thực tế từ việc ứng dụng ngôn ngữ R để phân tích dữ liệu, xây dựng mô hình hồi quy tuyến tính bội, đưa ra các dự đoán hợp lý và kiểm tra tính chính xác của chúng một cách cặn kẽ nhất. Những kết quả mà chúng em tìm ra sẽ là bước khởi đầu cho việc khai phá nguồn dữ liệu của hệ thống sau này, nhằm đạt tới mục tiêu nâng cao kỹ năng giải quyết vấn đề, kỹ năng lập trình, vận dụng kiến thức đã học vào các lĩnh vực liên quan, kỹ năng làm việc nhóm cũng như hướng tới mục tiêu cao hơn là đam mê trong công việc, học tập và nghiên cứu.

2. CƠ SỞ LÝ THUYẾT

2.1. Sơ lược về hồi quy tuyến tính bội

Hồi quy tuyến tính bội là một phần mở rộng của hồi quy tuyến tính đơn. Nó được sử dụng để dự đoán giá trị của một biến phản hồi dựa trên giá trị của hai hay nhiều biến giải thích khác. Biến chúng ta muốn dự đoán được gọi là biến phản hồi (biến phụ thuộc), các biến được sử dụng để dự đoán giá trị của biến phụ thuộc được gọi là biến giải thích.

Hồi quy tuyến tính bội cũng cho phép ta xác định sự phù hợp tổng thể của mô hình và đóng góp tương đối của từng yếu tố dự báo vào tổng phương sai được giải thích. Ví dụ: mức độ thay đổi trong kết quả kỳ thi cuối kỳ môn Văn có thể được giải thích bằng thời gian ôn tập và giới tính "nói chung", nhưng cũng là "đóng góp tương đối" của mỗi biến độc lập trong việc giải phương sai.

2.2. Mô hình hồi quy bội

Giả sử biến Y (biến phản hồi) phụ thuộc vào k biến độc lập X_1, X_2, ..., X_k (các biến giải thích). Mô hình hồi quy tuyến tính bội có dạng:

$$Y = \alpha + X_1\beta_1 + X_2\beta_2 + \ldots + X_k\beta_k + U$$

Trong đó:

- α: hệ số chặn (còn gọi là hệ số tự do), cho ta biết trung bình của Y khi X_1, X_2, ...X_k bằng 0.
- β_j (j = 1, 2, ..., k): các hệ số hồi quy riêng, thể hiện độ biến thiên của Y khi X_j thay đổi, còn các biến khác không đổi.
- U: nhiễu ngẫu nhiên (sai số)

2.3. Phương trình hồi quy bội của mẫu

Gọi các hệ số a, b_1, . . . b_k là ước lượng cho α_1, β_1, . . . β_k được xác định bởi phương pháp bình phương bé nhất:

$$f = \sum_{i=1}^{n} (y_i - a - b_1 x_{1i} - \ldots - b_k x_{ki})^2$$

Từ điều kiện trên, ta có hệ:

$$\begin{cases} \delta f/\delta a = 0 \\ \delta f/\delta b_1 = 0 \\ \ldots \\ \delta f/\delta b_k = 0 \end{cases}$$

Giải hệ phương trình ta sẽ tìm được nghiệm (a, b_1, . . . b_k)

Phương trình $y = a + b_1x_1 + b_2x_2 + ... + b_kx_k$ gọi là phương trình hồi quy bội của mẫu.

Chúng ta cũng có thể tìm được nghiệm (a, b_1, . . . b_k) bằng phương pháp ma trận, tuy nghiên dù phương pháp nào đi nữa thì việc tìm nghiệm bằng phương pháp thủ công là rất phức tạp. Với công nghệ máy tính phát triển, các phần mềm thống kê được phát triển thì việc tìm nghiệm trở nên dễ dàng hơn. Chính vì vậy, chúng ta không nên quá quan tâm đến việc tìm nghiệm bằng phương pháp thử công như thế nào.

Tương tự như đối với hồi quy tuyến tính đơn giản, phương pháp bình phương bé nhất phải thoả mãn những điều kiện:

- Quan hệ giữa Y và X là tuyến tính Các giá trị X_i cho trước và không ngẫu nhiên
- Các sai số U_i có phân phối chuẩn $N(0, \sigma_2)$.
- Các sai số U_i là đại lượng ngẫu nhiên có giá trị trung bình bằng 0.
- Các sai số U_i là đại lượng ngẫu nhiên có phương sai không thay đổi.
- Không có sự tương quan giữa U_i và X_i.
- Các biến X_j độc lập với nhau.

2.4. Khoảng tin cậy của hệ số hồi quy

Mô hình hồi quy bội có dạng:

$$Y = \alpha + X_1\beta_1 + X_2\beta_2 + \ldots + X_k\beta_k + U$$

Tương tự đối với hồi quy đơn giản, ước lượng khoảng của các hệ số như sau:

– Ước lượng khoảng của B_i với độ tin cậy $(1-\alpha) = 100\%$ là:

$$b_i - t_{n-k-1,\alpha/2} S_{bi} < \beta_i < b_i + t_{n-k-1,\alpha/2} S_{bi}$$

– Ước lượng khoảng của α với độ tin cậy $(1-\alpha) = 100\%$ là:

$$a - t_{n-k-1,\alpha/2} S_a < a < a + t_{n-k-1,\alpha/2} S_a$$

2.5. Kiểm định từng tham số hồi quy tổng thể (PI)

Tương tự như đối với kiểm định của hồi quy đơn giản.

Trường hợp $\beta_i = 0$ thì X_i và Y không có mối quan hệ nào, trường hợp $\beta_i > 0$ ($\beta i < 0$) giữa X_i và Y có mỗi quan hệ thuận (nghịch).

Ở mức ý nghĩa α, giả thuyết H_0 kiểm định ở các trường hợp sau:

Giả thuyết	$H_0: B_i \leq 0$ $H_1: B_i > 0$	$H_0: B_i \geq 0$ $H_1: B_i < 0$	$H_0: B_i = 0$ $H_1: B_i \neq 0$
Giá trị kiểm định	$t = \frac{b_i}{S_{bi}}$		
Bác bỏ H_0	$t > t_{n-k-1,\alpha}$	$t < -t_{n-k-1,\alpha}$	$t > t_{n-k-1,\alpha}$ hoặc $t < -t_{n-k-1,\alpha}$

Đây là một phương pháp xây dựng mô hình hồi quy, được gọi là phương pháp loại biến dần. Chúng ta sẽ loại từng biến một dựa vào giá trị p kiểm định lớn cho trước.

2.6. Phân tích phương sai hồi quy

Tương tự đối với hồi quy đơn, ta có:

– Hệ số xác định:

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

Nhưng ở đây hệ số R_2 là nói lên tính chặt chẽ giữa biến phụ thuộc Y và các biến độc lập X_i , tức là nó thể hiện phần trăm biến thiên của Y có thể được giải thích bởi sự biến thiên của tất cả các biến X_i.

Đối với người nghiên cứu thì họ mong muốn hệ số R_2 càng lớn càng tốt, tuy nhiên R_2 là một hàm không giảm theo số lượng biến đưa vào. Điều này có thể dẫn đến một sai lầm về số R_2 bằng cách đưa vào mô hình càng nhiều biến để có hệ số R_2 lớn. Để khắc phục nhược điểm này, người ta đưa ra hệ số xác định điều chỉnh đánh giá mức độ phụ thuộc của Y vào các biển X chính xác hơn.

- Hệ số đã điều chỉnh:

$$\bar{R}^2 = \frac{SSR/(n-k-1)}{SST/(n-1)} = 1 - (1-R^2)\left(\frac{n-1}{n-k-1}\right)$$

Xét về mặt ý nghĩa thì giữa R^2 và $\bar{R}^2$ là như nhau. Thông thường thì hai hệ số này chênh lệch nhau không nhiều. Trong một số trường hợp số lượng biến X tương đối lớn so với n, khi đó ta nên dùng hệ số xác định có điều chỉnh để đo lường mức độ thích hợp của mô hình hồi quy bội.

2.7. Hồi quy tuyến tính bội trong R

Có nhiều cách có thể thực hiện hồi quy tuyến tính nhiều lần nhưng thường được thực hiện thông qua phần mềm thống kê. Một trong những phần mềm được sử dụng nhiều nhất là RStudio, miễn phí, mạnh mẽ và dễ sử dụng.

Các bước để thực hiện hồi quy bội trong R:

- Thu thập dữ liệu: Dữ liệu được thu thập sẽ được sử dụng trong dự đoán
- Thu thập dữ liệu trong R: Thu thập dữ liệu bằng mã và nhập tệp *.csv*
- Làm sạch dữ liệu: Trích ra dữ liệu các biến đề yêu cầu và dùng lệnh ***cbind()***
- Xây dựng mô hình hồi quy tuyến tính bội: Xác định biến phụ thuộc và các biến độc lập, sử dụng lệnh ***lm()*** để thực thi mô hình hồi quy tuyến tính bội

– Dự báo: Thực hiện dự báo và kiểm định thông qua mô hình hồi quy, sử dụng lệnh ***predict()***

2.8. Giới thiệu R

2.8.1. Tính năng

R có chứa nhiều loại kỹ thuật thống kê (mô hình hóa tuyến tính và phi tuyến tính, kiểm thử thống kê cổ điển, phân tích chuỗi thời gian, phân loại, phân nhóm,...) và đồ họa. R cho phép người dùng thêm các tính năng bổ sung bằng cách định nghĩa các hàm mới. R có thể liên kết được với ngôn ngữ C, C++ và Fortran để có thể được gọi trong khi chạy. Người dùng thông thạo có thể viết mã C để xử lý trực tiếp các đối tượng của R.

R cũng có tính mở rộng cao bằng cách sử dụng các gói cho người dùng đưa lên cho một số chức năng và lĩnh vực nghiên cứu cụ thể.

Một điểm mạnh khác của R là nền tảng đồ họa của nó, có thể tạo ra những đồ thị chất lượng cao cùng các biểu tượng toán học. R cũng có đinh dạng văn bản riêng tương tự như LaTeX, dùng để cung cấp tài liệu hướng dẫn toàn diện, có trực tuyến ở các định dạng khác nhau và cả bản in.

2.8.2. Một số thư viện được dùng trong bài báo cáo

```
library(readr)
library(ggplot2)
library(plyr)
library(dplyr)
library(tidyverse)
library(BMA)
library(performance)
```

2.8.3. Một số lệnh được sử dụng trong bài báo cáo

- read.csv: Đọc file .csv trong Rstudio.
- apply: Dùng để tính toán nhanh cả bộ dữ liệu.
- na_omit: Giúp chúng ta loại bỏ các dòng mà có giá trị NA ở bất kỳ cột nào trong dữ liệu.
- mean, median, min, max: tính trung bình, trung vị, giá trị min, max
- table: Tạo bảng.
- hist, pairs, qplot: Vẽ biểu đồ.
- boxplot: Vẽ đồ thị boxplot.
- summary: Liệt kê giá trị tinh toán của mô hình.
- lm: Tính hệ số hồi quy.
- predict: Dự báo các giá trị phản hồi của một tập dữ liệu mới.

3. PHẦN CHUNG

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

3.1. Nhập và "làm sạch" dữ liệu, thực hiện các thống kê mô tả.

3.1.1. Đọc dữ liệu

Đọc dữ liệu và lưu vào biến auto_mpg, vì dữ liệu đầu tiên là dữ liệu thô,nên để chia được cột rõ ràng thì cần thêm thư viện readr dùng lệnh read_delim.

- *Code R:*

```
auto_mpg <- read_delim("auto_mpg.csv", delim = ";",
                quote = "'", escape_double = FALSE, na = "?",
                comment = "#", trim_ws = TRUE)
```

- *Kết quả thực nghiệm:*

	mgp	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	car_name
1	18.0	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
2	15.0	8	350.0	165	3693	11.5	70	1	buick skylark 320
3	18.0	8	318.0	150	3436	11.0	70	1	plymouth satellite
4	16.0	8	304.0	150	3433	12.0	70	1	amc rebel sst
5	17.0	8	302.0	140	3449	10.5	70	1	ford torino
6	15.0	8	429.0	198	4341	10.0	70	1	ford galaxie 500
7	14.0	8	454.0	220	4354	9.0	70	1	chevrolet impala
8	14.0	8	440.0	215	4312	8.5	70	1	plymouth fury iii
9	14.0	8	455.0	225	4425	10.0	70	1	pontiac catalina
10	15.0	8	390.0	190	3850	8.5	70	1	amc ambassador dpl
11	15.0	8	383.0	170	3563	10.0	70	1	dodge challenger se
12	14.0	8	340.0	160	3609	8.0	70	1	plymouth 'cuda 340
13	15.0	8	400.0	150	3761	9.5	70	1	chevrolet monte carlo
14	14.0	8	455.0	225	3086	10.0	70	1	buick estate wagon (sw)
15	24.0	4	113.0	95	2372	15.0	70	3	toyota corona mark ii
16	22.0	6	198.0	95	2833	15.5	70	1	plymouth duster
17	18.0	6	199.0	97	2774	15.5	70	1	amc hornet
18	21.0	6	200.0	85	2587	16.0	70	1	ford maverick
19	27.0	4	97.0	88	2130	14.5	70	3	datsun pl510
20	26.0	4	97.0	46	1835	20.5	70	2	volkswagen 1131 deluxe sedan
21	25.0	4	110.0	87	2672	17.5	70	2	peugeot 504
22	24.0	4	107.0	90	2430	14.5	70	2	audi 100 ls
23	25.0	4	104.0	95	2375	17.5	70	2	saab 99e
24	26.0	4	121.0	113	2234	12.5	70	2	bmw 2002
25	21.0	6	199.0	90	2648	15.0	70	1	amc gremlin
26	10.0	8	360.0	215	4615	14.0	70	1	ford f250
27	10.0	8	307.0	200	4376	15.0	70	1	chevy c20
28	11.0	8	318.0	210	4382	13.5	70	1	dodge d200
29	9.0	8	304.0	193	4732	18.5	70	1	hi 1200d

3.1.2. Làm sạch dữ liệu (xoá dữ liệu khuyết và dữ liệu ngoại lai)

3.1.2.1. Xóa dữ liệu khuyết (NA- Not Available)

Thực hiện kiểm tra các dữ liệu bị khuyết và đề suất phương án thay thế cho phần dữ liệu bị khuyết:

- Kiểm tra xem từng biến đó có bao nhiêu dữ liệu khuyết,và xuất các dòng bị khuyết của cột đó
- Phương án: Xoá bỏ những hàng bị khuyết

- *Code R:*

```
apply(is.na(auto_mpg),2,sum) # Dem so luong NA
```

```
apply(is.na(auto_mpg),2,which) # cac cot bi khuyet, xuat dong bi
khuyet

auto_mpg = na.omit(auto_mpg) # xoa du lieu bi khuyet

auto_mpg = na.omit(auto_mpg) # xoa du lieu bi khuyet
```

- Kết quả thực nghiệm:

```
> apply(is.na(auto_mpg),2,sum) #Dem so luong NA
         mgp    cylinders displacement   horsepower       weight acceleration   model_year       origin
           0            0            0            6            0            0            0            0
    car_name
           0
> apply(is.na(auto_mpg),2,which) #cac cot bi khuyet, xuat dong bi khuyet
$mgp
integer(0)

$cylinders
integer(0)

$displacement
integer(0)

$horsepower
[1]  33 127 331 337 355 375

$weight
integer(0)

$acceleration
integer(0)

$model_year
integer(0)

$origin
integer(0)

$car_name
integer(0)

> auto_mpg = na.omit(auto_mpg) #xoa du lieu bi khuyet
```

- Kiểm tra lại xem đã xoá hết các dữ liệu khuyết của các biến hay không bằng cách gọi lại lệnh

- *Code R:*

```
apply(is.na(auto_mpg),2,sum) #Dem so luong NA
```

- *Kết quả thực nghiệm:*

```
> apply(is.na(auto_mpg),2,sum) #Dem so luong NA
        mgp   cylinders displacement  horsepower      weight acceleration  model_year      origin
          0           0            0           0           0            0           0           0
   car_name
          0
```

3.1.2.2. Xóa dữ liệu ngoại lai (Outliers)

Ta nhận thấy duy nhất dữ liệu biến car_name là ko phải dữ liệu thực nên ta tiến hành lưu các biến còn lại có giá trị thực vào biến temp.

- *Code R:*

```
temp <- auto_mpg[,c(1:8)]
```

Ta tiến hành kiểm tra biến nào có giá trị ngoại lai và nếu có thì xuất ra

- *Code R:*

```
temp <- auto_mpg[,c(1:8)]
for(i in 1:ncol(temp)){
    if(length(boxplot.stats( temp[[i]])$out)!=0)
        cat(names(temp[i]),'la bien co gia tri ngoai lai\n')}
```

- *Kết quả thực nghiệm:*

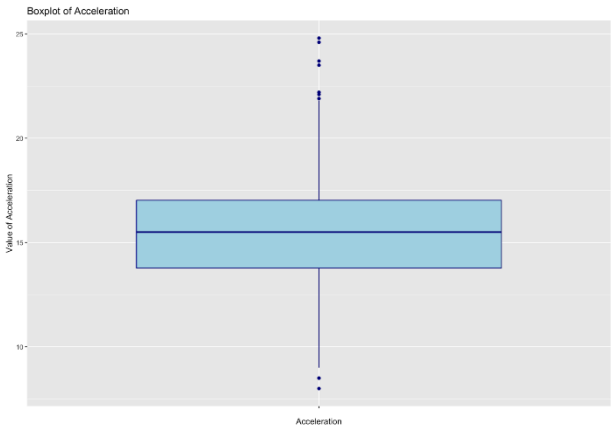
Có 2 biến có giá trị ngoại lai: horsepower và acceleration

Ta tiến hành thêm thư viện tidyverse và gọi lệnh qplot để nhìn rõ hơn dữ liệu ngoại lai.

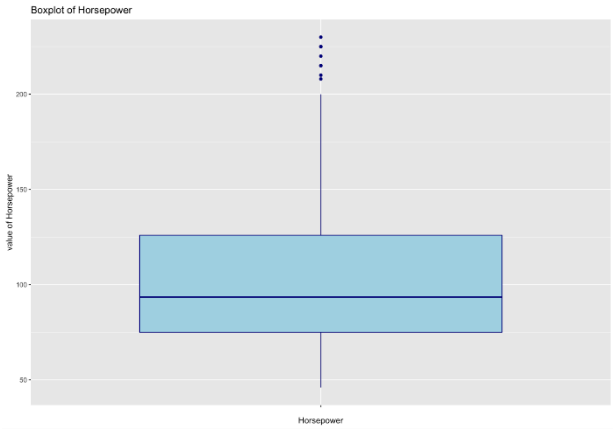
- *Code R:*

```
library(tidyverse)
qplot(y=temp$horsepower,x='',geom='boxplot',
      col=I('darkblue'),fill=I('lightblue'),ylab='value of
Horsepower',
      xlab='Horsepower',main='Boxplot of Horsepower')
qplot(y=temp$acceleration,x='',geom='boxplot',
      col=I('darkblue'),fill=I('lightblue'),ylab='Value of
Acceleration',
      xlab='Acceleration',main='Boxplot of Acceleration')
```

- *Kết quả thực nghiệm:*



Boxplot of Acceleration



Boxplot of Horsepower

Tiến hành xóa outliers:

- *Code R:*

```
for(i in 1:ncol(temp)){
    upper_outliers= quantile(temp[[i]], 0.75) +1.5 *IQR(temp[[i]])
    lower_outliers= quantile(temp[[i]], 0.25) -1.5 *IQR( temp[[i]])
    while(length(boxplot.stats( temp[[i]])$out)!=0){
          temp <- subset(temp, temp[[i]]<=upper_outliers &
temp[[i]]>=lower_outliers)
```

```
    upper_outliers= quantile( temp[[i]], 0.75) +1.5 *IQR( temp[[i]])
    lower_outliers= quantile( temp[[i]], 0.25) -1.5 *IQR( temp[[i]])
    }
}
```

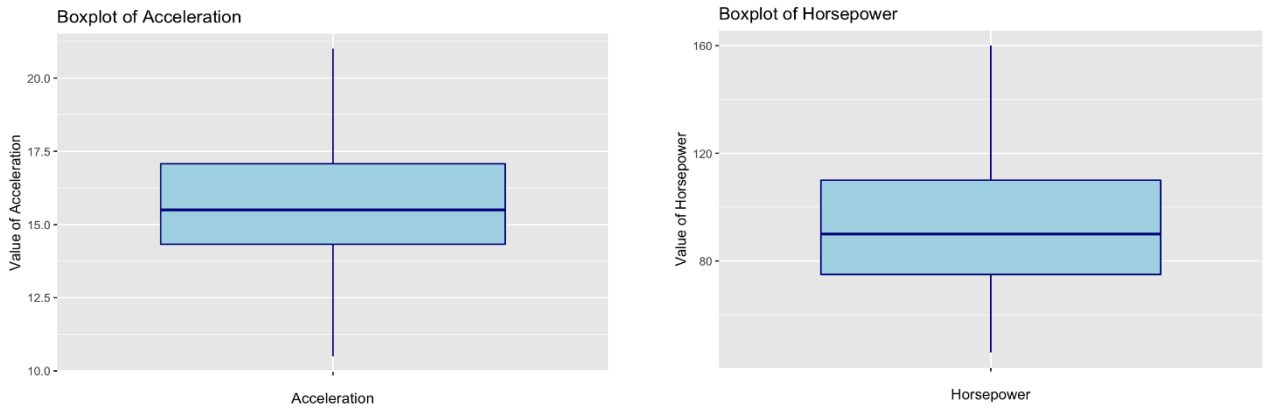
Tiến hành kiểm tra lại các biến, xem có sót outliers không:

- *Code R:*

```
for(i in 1:ncol(temp)){
    print(boxplot.stats( temp[[i]])$out)}

qplot(y=temp$horsepower,x='',geom='boxplot',
   col=I('darkblue'),fill=I('lightblue'),ylab='Value of Horsepower',
   xlab='Horsepower',main='Boxplot of Horsepower')

qplot(y=temp$acceleration,x='',geom='boxplot',
 col=I('darkblue'),fill=I('lightblue'),ylab='Value of Acceleration',
 xlab='Acceleration',main='Boxplot of Acceleration')
```

- *Kết quả thực nghiệm:*

```
numeric(0)
numeric(0)
numeric(0)
numeric(0)
numeric(0)
numeric(0)
numeric(0)
numeric(0)
```



3.1.3. Thực hiện các thống kê mô tả

3.1.3.1. Thống kê dữ liệu

Đầu tiên, ta tiến hành thống kê các biến liên tục.

Dựa vào dữ liệu đã được cung cấp, nhận thấy các biến liên tục bao gồm mgp, displacement, horsepower, weight, acceleration.

- *Code R:*

```
cot <- c("mean", "median", "sd", "min", "max")
hang <- c('mgp','displacement','horsepower','weight','acceleration')
statistic <- c()
for(i in hang){
    cotm <- c(mean(temp[[i]]), median(temp[[i]]), sd(temp[[i]]),
              min(temp[[i]]), max(temp[[i]]))
    statistic <-rbind(statistic,cotm)
}
statistic <-as.data.frame(statistic)
rownames(statistic) <-hang
colnames(statistic) <-cot
statistic
```

- *Kết quả thực nghiệm:*

	mean	median	sd	min	max
mgp	24.30760	24.0	7.226292	11.0	46.6
displacement	175.36111	140.0	87.317949	68.0	400.0
horsepower	96.38889	90.0	26.977197	46.0	160.0
weight	2839.19591	2675.0	751.555825	1613.0	4997.0
acceleration	15.76608	15.5	2.175232	10.5	21.0

Tương tự như trên, nhận thấy các biến phân loại bao gồm: cylinders, model_year, origin, car_name.

- Code R:

```
stat_cylinders = table(temp$cylinders, dnn = "cylinders")
View(stat_cylinders)

stat_model_year = table(temp$model_year, dnn = "model_year")
View(stat_model_year)

stat_origin= table(temp$origin, dnn = "origin")
View(stat_origin)

stat_car_name= table(temp$car_name, dnn = "car_name")
View(stat_car_name)
```

- *Kết quả thực nghiệm:*

Sau khi chạy code R, ta thu được bảng thống kê các biến phân loại như sau:

	origin	Freq
1	1	203
2	2	60
3	3	79

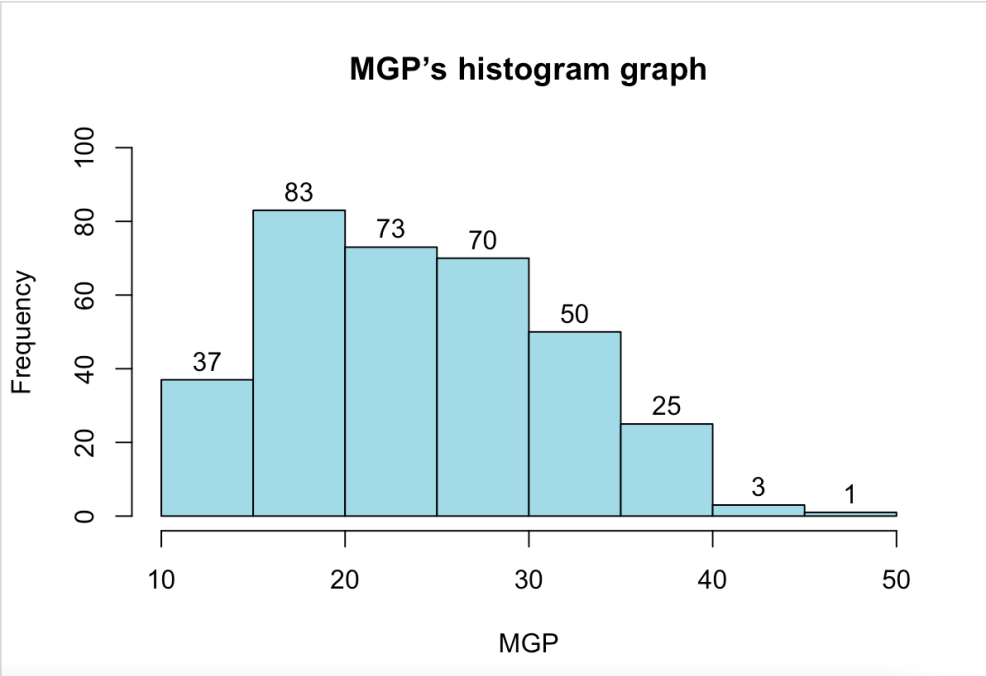
	model_year	Freq
1	70	15
2	71	22
3	72	23
4	73	30
5	74	26
6	75	29
7	76	30
8	77	25
9	78	34
10	79	27
11	80	24
12	81	28
13	82	29

	cylinders	Freq
1	3	4
2	4	189
3	5	3
4	6	82
5	8	64

	car_name	Freq
1	amc ambassador brougham	1
2	amc ambassador dpl	1
3	amc ambassador sst	1
4	amc concord	2
5	amc concord d/l	1
6	amc concord dl 6	1
7	amc gremlin	4
8	amc hornet	4
9	amc hornet sportabout (sw)	1
10	amc matador	5
11	amc matador (sw)	2
12	amc pacer	1
13	amc pacer d/l	1
14	amc rebel sst	1
15	amc spirit dl	1
16	audi 100 ls	1
17	audi 100ls	2
18	audi 4000	1
19	audi 5000	1
20	audi 5000s (diesel)	1

3.1.3.2. Vẽ đồ thị

Để có thể nhận xét rõ hơn về dữ liệu và mức tiêu thụ nhiên liệu, ta sử dụng code R để vẽ biểu đồ nhận xét:



Nhìn vào biểu đồ ta có thể thấy được:

- Mức tiêu thụ nhiên liệu có lượng xe chiếm cao nhất từ gần bằng 15 đến 20.
- Mức tiêu thụ nhiên liệu có lượng xe chiếm thấp nhất từ gần bằng 45 đến gần bằng 50.

Để nhận xét mối tương quan giữa mgp với các yếu tố khác, ta dùng code R để vẽ các biểu đồ boxplot tương ứng:

Boxplot: Biểu đồ hộp (Boxplot) hay còn gọi là biểu đồ hộp và râu (Box and whisker plot) là biểu đồ diễn tả 5 vị trí phân bố của dữ liệu, đó là: giá trị nhỏ nhất (min), tứ phân vị thứ nhất (Q1), trung vị (median), tứ phân vị thứ 3 (Q3) và giá trị lớn nhất (max).

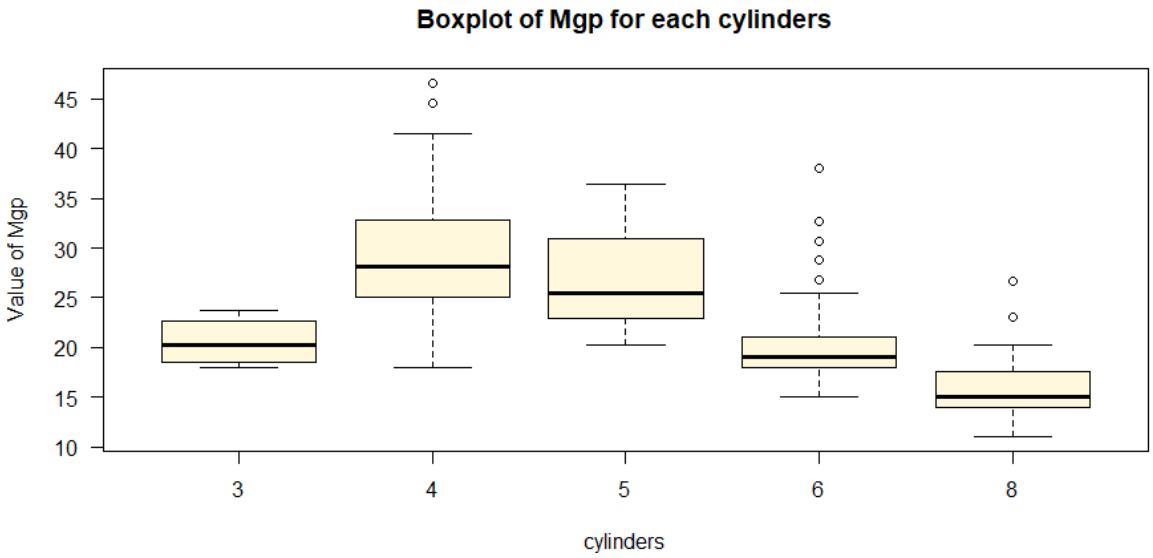
- Code R:

```
boxplot(data = temp, mgp ~ cylinders, col = "cornsilk",
        main = "Boxplot of Mgp for each cylinders",
        xlab = "cylinders", ylab = "Value of Mgp")
```

```
boxplot(data = temp, mgp ~ model_year, col = "darkgoldenrod1",
        main = "Boxplot of Mgp for each model_year",
        xlab = "model_year", ylab = "Value of Mgp")

boxplot(data = temp, mgp ~ origin, col = "darkolivegreen1",
        main = "Boxplot of Mgp for each origin",
        xlab = "origin", ylab = "Value of Mgp")

boxplot(data = auto_mpg, mgp ~ car_name, col = "darksalmon",
        main = "Boxplot of Mgp for each car_name",
        xlab = "car_name", ylab = "Value of Mgp")
```

- *Kết quả thực nghiệm:*

+ Biểu đồ tương quan giữa mgp với biến cylinders:

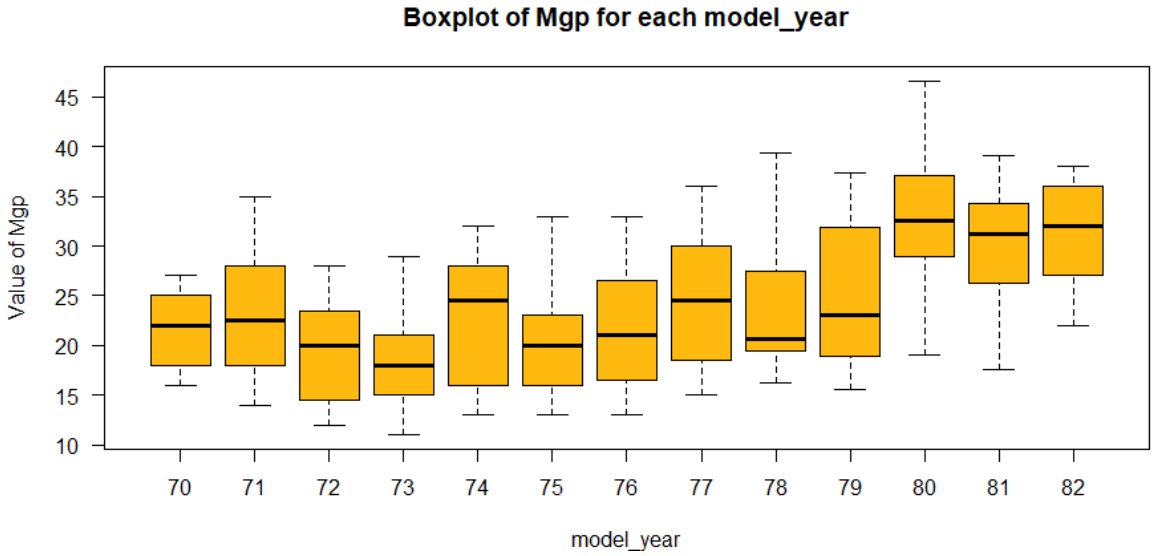


Nhận xét:

Đối với **cylinders = 3**: ta thấy giá trị nhỏ nhất (min) của mgp khoảng 17, giá trị lớn nhất (max) của mgp là khoảng 24, phân vị Q1 gần bằng 18 và phân vị Q3 là khoảng 23, trung vị bằng 20 và các giá trị ngoại lai.

Tương tự như vậy đối với cylinders = 4, cylinders = 5, cylinders = 6, cylinders = 8.

+ Biểu đồ tương quan giữa mgp với biến model_year:

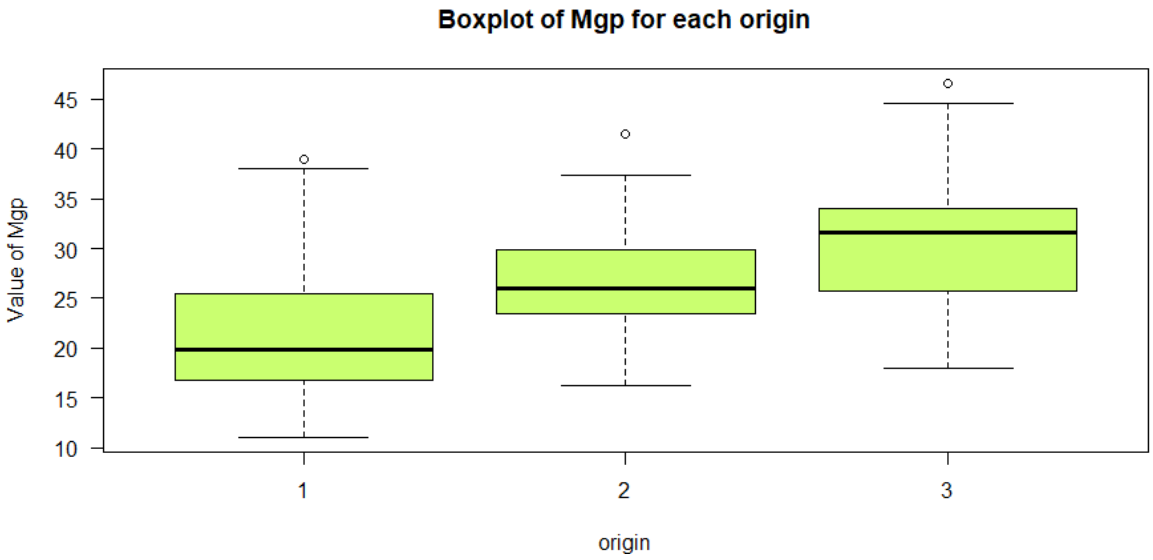


Nhận xét:

Đối với **model_year = 70**: ta thấy giá trị nhỏ nhất (min) của mgp khoảng 16, giá trị lớn nhất (max) của mgp là khoảng 27, phân vị Q1 gần bằng 18 và phân vị Q3 là 25, trung vị gần bằng 22 và các giá trị ngoại lai.

Tương tự như vậy đối với các giá trị model_year khác.

+ Biểu đồ tương quan giữa mgp với biến origin:

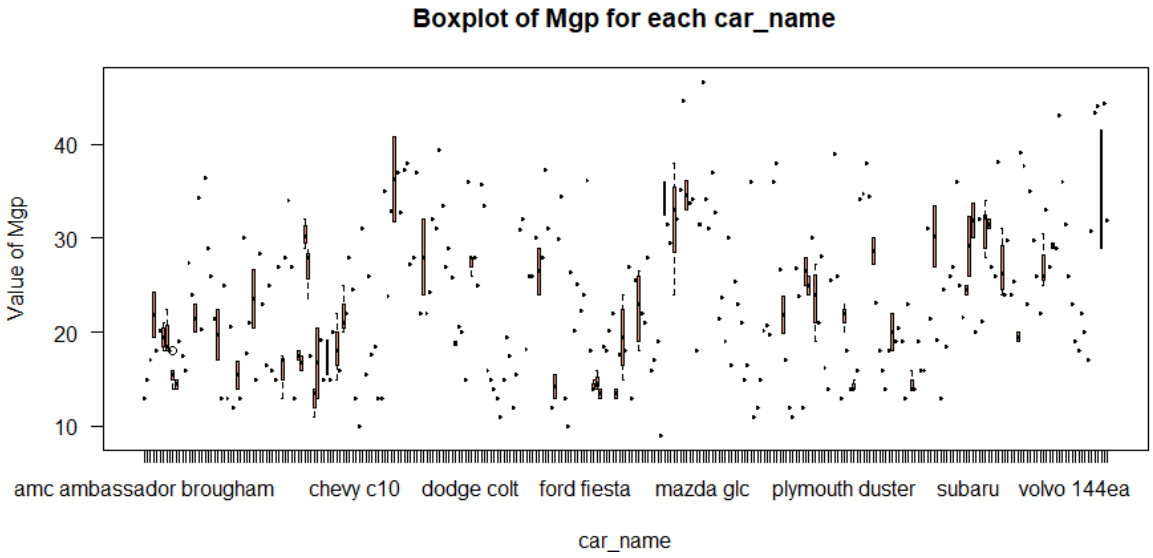


Nhận xét:

Đối với **origin = 1**: ta thấy giá trị nhỏ nhất (min) của mgp khoảng 11, giá trị lớn nhất (max) của mgp là khoảng 39, phân vị Q1 gần bằng 17 và phân vị Q3 là 25, trung vị bằng 20 và các giá trị ngoại lai.

Tương tự như vậy đối với origin = 2 và origin = 3.

+ Biểu đồ tương quan giữa mgp với biến car_name:

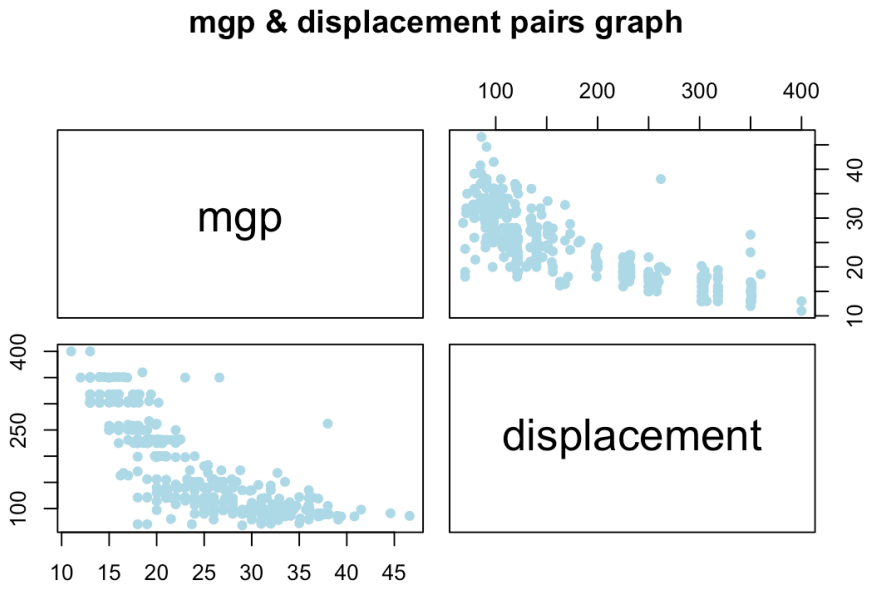


Để xét phân phối của biến Gross theo các biến Budget, Views và Screens, ta có thể dùng lệnh pairs() để vẽ các biểu đồ:

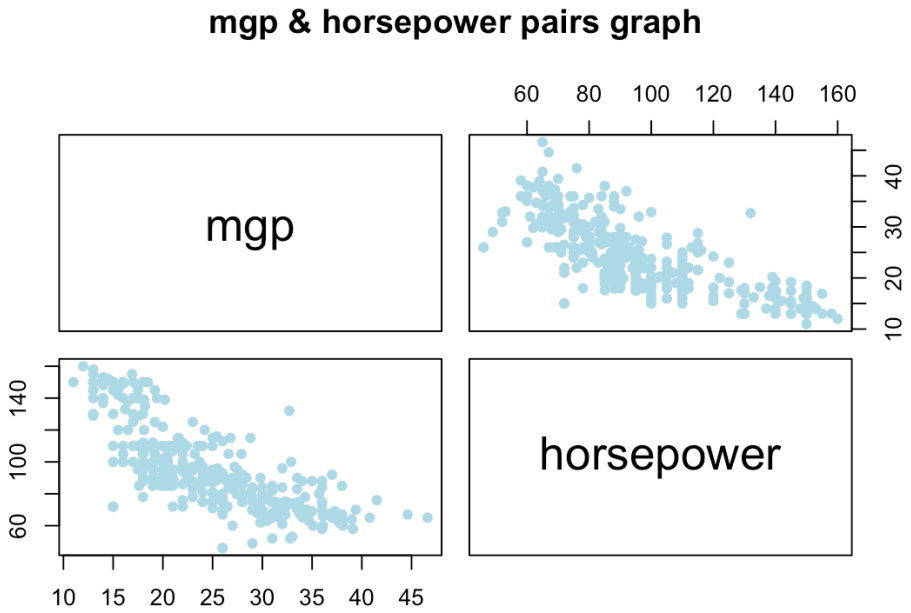
- Code R

```
pairs(data = temp, mgp ~ displacement, main = "mgp & displacement
            pairs graph", pch = 16,col = "lightblue")

pairs(data = temp, mgp ~ horsepower, main = "mgp & horsepower pairs
                graph", pch = 16,col = "lightblue")

pairs(data = temp, mgp ~ weight, main = "mgp & weight pairs graph",
                          pch = 16, col = "lightblue")

 pairs(data = temp, mgp ~ acceleration, main = "mgp & acceleration
             pairs graph",pch = 16, col = "lightblue")
```

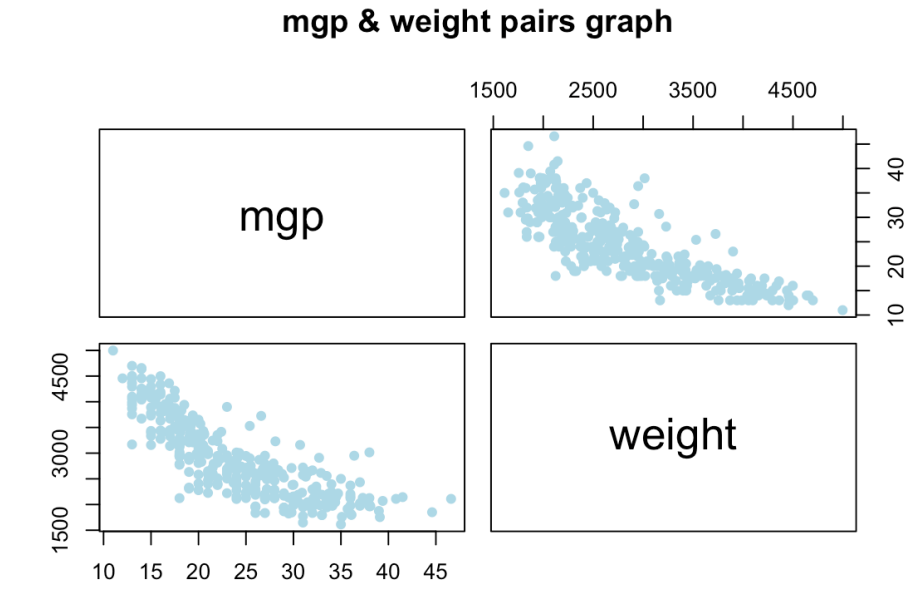
- Kết quả thực nghiệm
  - Biểu đồ phân phối của biến mgp theo biến **displacement**:



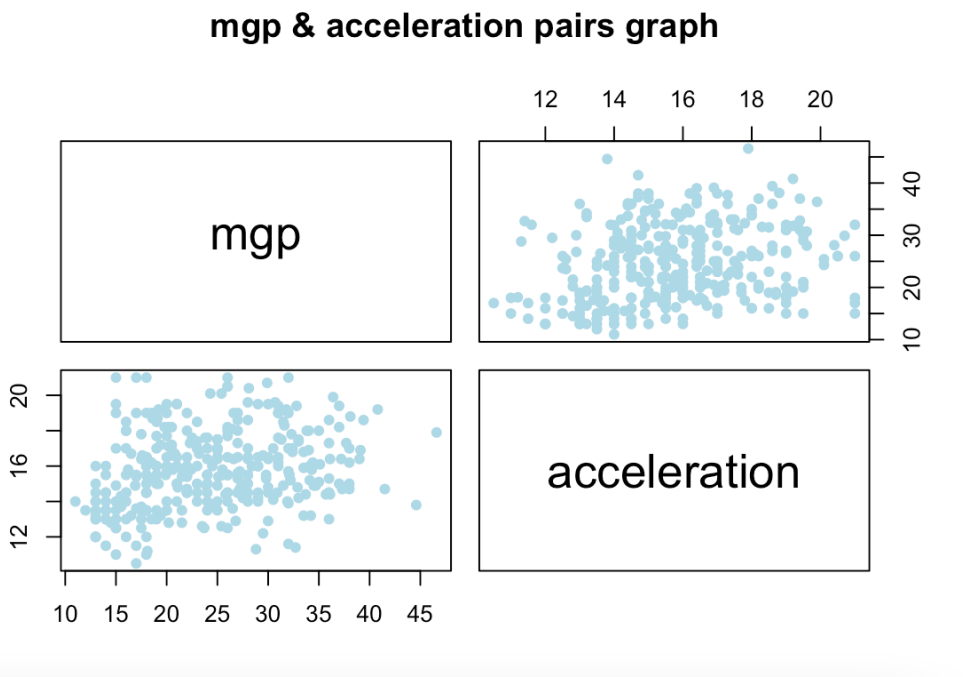
- Biểu đồ phân phối của biến mgp theo biến **horsepower**:



- Biểu đồ phân phối của biến mgp theo biến **weight**:



- Biểu đồ phân phối của biến mgp theo biến **acceleration**:



3.2. Chia bộ dữ liệu làm 2 phần: mẫu huấn luyện, và mẫu kiểm tra

- mẫu huấn luyện (training dataset) gồm 200 quan trắc, đặt tên "auto_mpg1"
- mẫu kiểm tra (validation dataset) gồm các quan trắc còn lại trong bộ dữ liệu ban đầu đã "làm sạch", đặt tên "auto_mpg2".

Code R:

```
auto_mpg1 <- temp[1:200,]
auto_mpg2 <- temp[201:342,]
```

Kết quả thực nghiệm:

- auto_mpg1:

	mgp	cylinders	displacement	horsepower	weight	acceleration	model_year	origin
1	18	8	307.0	130	3504	12.0	70	1
2	18	8	318.0	150	3436	11.0	70	1
3	16	8	304.0	150	3433	12.0	70	1
4	17	8	302.0	140	3449	10.5	70	1
5	24	4	113.0	95	2372	15.0	70	3
6	22	6	198.0	95	2833	15.5	70	1
7	18	6	199.0	97	2774	15.5	70	1
8	21	6	200.0	85	2587	16.0	70	1
9	27	4	97.0	88	2130	14.5	70	3
10	26	4	97.0	46	1835	20.5	70	2
11	25	4	110.0	87	2672	17.5	70	2
12	24	4	107.0	90	2430	14.5	70	2
13	25	4	104.0	95	2375	17.5	70	2

- auto_mpg2:

	mgp	cylinders	displacement	horsepower	weight	acceleration	model_year	origin
1	36.1	4	98	66	1800	14.4	78	1
2	32.8	4	78	52	1985	19.4	78	3
3	39.4	4	85	70	2070	18.6	78	3
4	36.1	4	91	60	1800	16.4	78	3
5	19.9	8	260	110	3365	15.5	78	1
6	19.4	8	318	140	3735	13.2	78	1
7	20.2	8	302	139	3570	12.8	78	1
8	19.2	6	231	105	3535	19.2	78	1
9	20.5	6	200	95	3155	18.2	78	1
10	20.2	6	200	85	2965	15.8	78	1
11	25.1	4	140	88	2720	15.4	78	1
12	20.5	6	225	100	3430	17.2	78	1
13	19.4	6	232	90	3210	17.2	78	1

3.3. 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".

3.3.1. Phương pháp chọn mô hình tối ưu BMA

Mô hình tối ưu được hiểu là mô hình sử dụng ít tham số nhưng giải thích dữ liệu nhiều, một trong những phương pháp chọn mô hình tối ưu hiệu quả và được nhiều người sử dụng là phương pháp BMA (Bayesian model average).

- X: biến tiên lượng cho biến phụ thuộc Y.
- Chọn ra 5 mô hình tối ưu nhất
- p!=0: xác suất hồi quy khác 0,và khác 0 thì nó sẽ ảnh hưởng đến biến phụ thuộc
- EV: giá trị kì vọng của các hệ số hồi quy
- SD: độ lệch chuẩn
- Biến nào không có trong mô hình nào thì giá trị tương ứng trong mô hình đó là '.'
- nVar: là số biến xuất hiện trong mô hình.
- BIC: giá trị BIC càng thấp thì mô hình càng tốt.
- r2 : giải thích phần trăm số dao động.
- post prob: xác suất xuất hiện của mô hình.

- *Code R:*

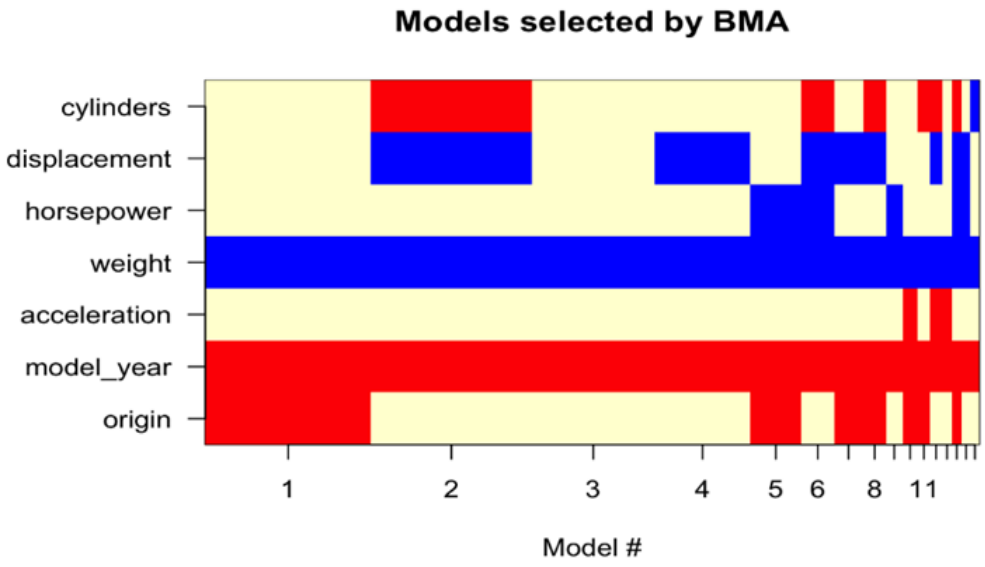
Để dùng được phương pháp bayes ta cần cài đặt thư viện (BMA)

```
(pValue>0.05 la khong co y nghia ve mat thong ke)
library(BMA)
X = auto_mpg1[, c('cylinders','displacement','horsepower','weight'
                  ,'acceleration','model_year','origin')]
Y = auto_mpg1$mgp
result <- bicreg(X, Y, strict = FALSE, OR = 20)
summary(result)
imageplot.bma(result)
```

- *Kết quả thực nghiệm:*

```
Call:
bicreg(x = X, y = Y, strict = FALSE, OR = 20)


  16  models were selected
 Best  5  models (cumulative posterior probability =  0.7703 ):

               p!=0    EV        SD        model 1     model 2     model 3     model 4     model 5
Intercept      100.0   5.879237  6.0669662  4.470e+00   5.586e+00   5.815e+00   7.596e+00   7.428e+00
cylinders       33.4   0.296855  0.4954312        .      9.524e-01         .           .           .
displacement    48.0  -0.010602  0.0136087        .     -2.883e-02         .     -1.202e-02        .
horsepower      15.3  -0.003006  0.0089164        .           .             .           .     -2.235e-02
weight         100.0  -0.005526  0.0007625 -5.970e-03  -5.015e-03  -6.369e-03  -5.076e-03  -5.220e-03
acceleration     4.8   0.002197  0.0211118        .           .             .           .           .
model_year     100.0   0.434698  0.0844057  4.538e-01   4.182e-01   4.660e-01   4.209e-01   4.120e-01
origin          39.6   0.260089  0.3753739  6.807e-01         .             .           .     7.893e-01

nVar                                             3           4           2           3           4
r2                                           0.828       0.833       0.823       0.827       0.831
BIC                                      -3.367e+02  -3.366e+02  -3.361e+02  -3.356e+02  -3.343e+02
post prob                                    0.214       0.207       0.159       0.123       0.066
```

Model 1, model 2, model 3, model 4, model 5 là những mô hình tối ưu nhất theo phương pháp BMA. Mô hình 1 có xác suất xuất hiện 21,4% (post prob = 0.214), cao nhất trong tất cả các mô hình, BIC thấp nhất (-3.367*10^-2), cùng với đó mô hình 1 có 3 biến xuất hiện trong mô hình (nVar = 3) nhưng giải thích được 82,8% (r2 = 0,828) mức ý nghĩa, do đó nhóm chọn mô hình tối ưu là mô hình 1.



Xác suất xuất hiện của các biến trong các mô hình

Màu xanh thể hiện biến có hệ số hồi quy < 0, màu đỏ thể hiện biến có hệ số hồi quy > 0, màu vàng nhạt thể hiện biến không xuất hiện trong mô hình.

3.3.2. Mô hình hồi quy bội

- *Code R:*

```
# mo hinh toi uu se loai bo cac bien cylinders, displacement,
horsepower, acceleration
optimal_model <-lm(mgp~weight + model_year + origin, data =
auto_mpg1)
summary(optimal_model)
```

- *Kết quả thực nghiệm*

```
Call:
lm(formula = mgp ~ weight + model_year + origin, data = auto_mpg1)

Residuals:
    Min      1Q  Median      3Q     Max
-8.9582 -1.5132  0.2134  1.5863  6.1221

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.470083   5.873416   0.761   0.4475
weight      -0.005970   0.000266 -22.449  < 2e-16 ***
model_year   0.453798   0.079638   5.698 4.39e-08 ***
origin       0.680675   0.281238   2.420   0.0164 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.41 on 196 degrees of freedom
Multiple R-squared:  0.8284,    Adjusted R-squared:  0.8258
F-statistic: 315.5 on 3 and 196 DF,  p-value: < 2.2e-16
```

Nhận xét:

- Mô hình không còn những biến độc lập không có ý nghĩa thống kê theo kiểm định.

- Phần dư của mô hình (Residuals) có giá trị trung vị là 0.2134 gần với giá trị 0 và phân vị thứ nhất, phân vị thứ ba có vị trí khá cân đối, vì vậy mô hình có thể chấp nhận được.
- Giá trị phương sai phần dư (Residual standard error) của mô hình là 2.41. Giá trị này càng nhỏ thì phương trình tuyến tính thu được càng hiệu quả.
- Ngoài ra ta còn có thể sử dụng trị số r^2 (Multiple R- squared) hay hệ số xác định bộI (coefficient of determination). Ở mô hình trên, giá trị này là 0.828 tức có nghĩa là phương trình tuyến tính thu được giải thích khoảng 82,8% các khác biệt về APM giữa các cá nhân. Trị số này có giá trị trong khoảng từ 0 đến 1 và trị số này càng cao thì là một dấu hiệu cho thấy mối liên hệ giữa biến phụ thuộc và các biến độc lập trong mô hình càng chặt chẽ.
- Các giá trị Pr(>F) của các biến độc lập nhỏ hơn mức ý nghĩa. Điều này mang ý nghĩa mô hình có thể chấp nhận được.

3.4. Kiểm tra các giả định (giả thiết) của mô hình.

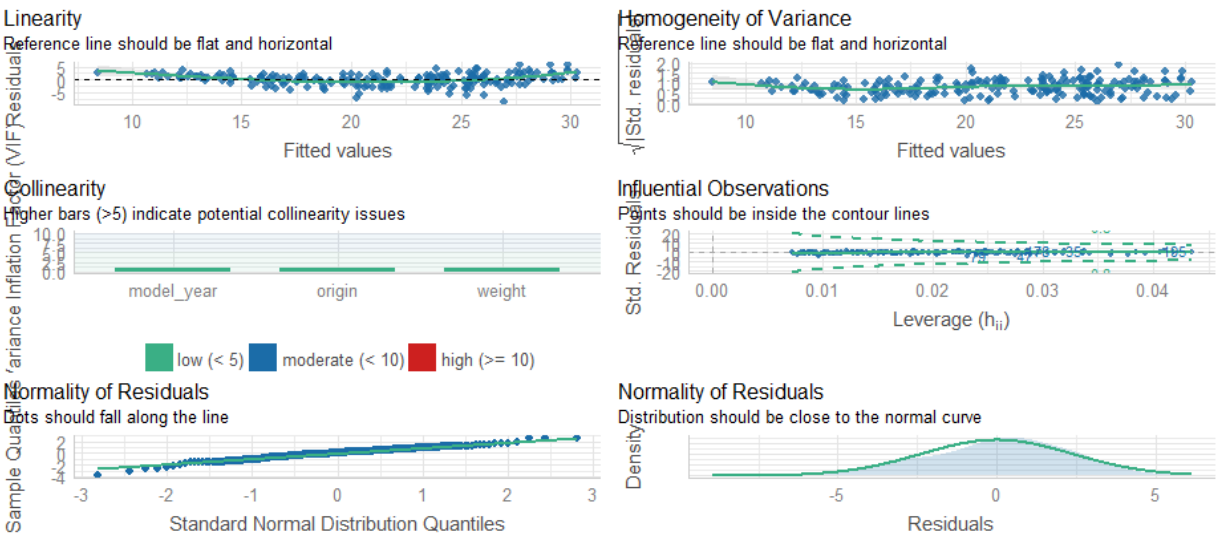
- *Các gói cần cài đặt*

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

- *Code R*

```
library(performance)
check_model(optimal_model)
```

- *Kết quả thực nghiệm*



Giả định: Liên hệ tuyến tính giữa biến phụ thuộc với biến độc lập

Biểu đồ phân tán Scatter Plot giữa các phần dư chuẩn hóa và giá trị dự đoán chuẩn hóa giúp chúng ta dò tìm xem dữ liệu hiện tại có vi phạm giả định liên hệ tuyến tính hay không. Nếu phần dư chuẩn hóa phân bổ ngẫu nhiên xung quanh đường tung độ 0 và hình dạng tạo thành một đường thẳng, chúng ta có thể kết luận giả định quan hệ tuyến tính không bị vi phạm.

Trên *biểu đồ Linearity* (biểu đồ trên cùng bên trái), phần dư chuẩn hóa phân bổ ngẫu nhiên tập trung xung quanh đường tung độ 0 tạo thành dạng đường thẳng, do vậy giả định quan hệ tuyến tính giữa biến phụ thuộc với các biến độc lập không bị vi phạm.

Giả định: Phương sai phần dư không thay đổi

Giả định phương sai của phần dư không thay đổi (hay còn gọi là phương sai đồng nhất). Nếu xảy ra hiện tượng phương sai thay đổi, kết quả của phương trình hồi quy sẽ không chính xác, làm sai lệch kết quả so với thực tế, từ đó khiến người nghiên cứu đánh giá nhầm chất lượng của phương trình hồi quy tuyến tính. Để đánh giá mô hình hồi quy có vi phạm giả định này hay không, chúng ta sẽ dựa vào biểu đồ Scatter Plot giữa các phần dư chuẩn hóa và giá trị dự đoán chuẩn hóa như ở giả định liên hệ tuyến

tính. Nếu các điểm phân vị phân bố khá đồng đều trên và dưới trục tung độ 0 dù X tăng hay giảm thì giả định phương sai phần dư không thay đổi không bị vi phạm.

Kết quả từ *biểu đồ Homogeneity of variance* (biểu đồ trên cùng bên phải) cho thấy, các điểm phân vị dao động khá đồng đều trên dưới trục tung độ 0. Các điểm phân vị hầu như dọc theo trục tung độ 0. Do đó, giả định phương sai phần dư đồng nhất không bị vi phạm.

Giả định: Không có đa cộng tuyến

Đa cộng tuyến phát sinh khi có mối tương quan tuyến tính mạnh giữa các biến độc lập, có điều kiện với các biến khác trong mô hình. Điều quan trọng là phải kiểm tra nó vì nó có thể dẫn đến sự không chính xác hoặc sự không ổn định của các tham số ước tính khi một biến thay đổi. Nó có thể được đánh giá bằng cách tính toán hệ số lạm phát phương sai (VIF). VIF đo lường mức độ tăng của phương sai của hệ số hồi quy ước tính so với tình huống trong đó các biến giải thích hoàn toàn độc lập. Giá trị VIF cao là dấu hiệu của đa cộng tuyến (ngưỡng thường được chấp nhận là 5 hoặc 10 tùy thuộc vào miền). Cách dễ nhất để giảm VIF là loại bỏ một số biến độc lập có tương quan, hoặc cuối cùng là chuẩn hóa dữ liệu.

Biểu đồ Collinearity (biểu đồ ở giữa bên trái), với các biến độc lập có giá trị VIF nhỏ hơn 1, các giá trị VIF này ở mức thấp, do đó không có hiện tượng đa cộng tuyến.

Giả định: Không có điểm ảnh hưởng

Nếu dữ liệu chứa các giá trị ngoại lai, điều cần thiết là phải xác định chúng để chúng không ảnh hưởng đến kết quả của hồi quy. Một điểm nằm gần thẳng hàng của đường hồi quy và không ảnh hưởng đến đường hồi quy thì không phải là điểm ngoại lai.

Kết quả từ *biểu đồ Influential Observations* (biểu đồ ở giữa bên phải) cho thấy, các điểm phân vị dao động xung quanh đường hồi quy, không có điểm ngoại lai. Do đó, giả định không có điểm ảnh hưởng không bị vi phạm.

Giả định: Phân phối chuẩn của phần dư

Biểu đồ Normality of Residuals (biểu đồ dưới cùng bên trái) có các điểm phân vị trong phân phối của phần dư tập trung thành 1 đường chéo, nghĩa là phần dư có phân phối chuẩn. Như vậy, giả định phân phối chuẩn của phần dư không bị vi phạm.

Biểu đồ Normality of Residuals (biểu đồ dưới cùng bên phải) cũng cho kết luận tương tự, giá trị trung bình Mean gần bằng 0, với độ lệch chuẩn gần bằng 1, đường cong phân phối có dạng hình chuông nên ta có thể khẳng định phân phối là xấp xỉ chuẩn, giả định phân phối chuẩn của phần dư không bị vi phạm.

3.5. Nêu ý nghĩa của mô hình đã chọn.

Mô hình tối ưu nhất được chọn là model 1, loại bỏ 4 biến độc lập cylinders, displacement, horsepower và acceleration. (kết quả thực nghiệm mục 3.2)

```
  16  models were selected
 Best  5  models (cumulative posterior probability =  0.7703 ):

              p!=0    EV        SD         model 1
Intercept     100.0   5.879237  6.0669662   4.470e+00
cylinders      33.4   0.296855  0.4954312       .
displacement   48.0  -0.010602  0.0136087       .
horsepower     15.3  -0.003006  0.0089164       .
weight        100.0  -0.005526  0.0007625  -5.970e-03
acceleration    4.8   0.002197  0.0211118       .
model_year    100.0   0.434698  0.0844057   4.538e-01
origin         39.6   0.260089  0.3753739   6.807e-01

nVar                                          3
r2                                           0.828
BIC                                         -3.367e+02
post prob                                    0.214
```

Giải thích:

R2 = 0.828: hệ số xác định R2, thể hiện các biến độc lập trong mô hình đang giải thích được khoảng 82.8% sự biến thiên của biến phụ thuộc mpg trong mô hình.

Cột dọc model 1: là hệ số hồi quy của mỗi biến độc lập trong mô hình hồi quy (Còn gọi là Beta), những biến có dấu "." là những biến không có nhiều ý nghĩa trong mô hình.

Ta có phương trình hồi quy tuyến tính như sau:

mpg = α + β1 ∗ cylinders + β2 ∗ displacement + β3 ∗ horsepower + β4 ∗ weight+ β5 ∗ acceleration + β6 ∗ model_year + β7 ∗ origin

Các hệ số chính của mô hình:

α = 4.47; β4 = **-** 0.00597; β6 = + 0.4538; β7 = 0.6807

Vậy:

mpg = 4.47 - 0.00597*weight + 0.4538* model_year + 0.6807* origin

Kết luận:

Biến weight có tương quan âm với biến phụ thuộc mpg (Vì hệ số hồi quy của biến này < 0).

Biến model_year và origin có tương quan dương với biến phụ thuộc mpg (Vì hệ số hồi quy của 2 biến này > 0).

Nhận xét:

Khi weight giảm đi 1 thì mpg giảm đi 0.00597 lần trong điều kiện các yếu tố khác không đổi.

Khi giá trị model_year tăng 1 thì lượng mpg sẽ tăng lên 0.4538 lần trong điều kiện các yếu tố khác không đổi (Tương tự với origin).

3.6. 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".

Dùng hàm ***predict()*** để đưa ra dự báo.

- *Code R*

```
# dung ham predict de dua ra du bao
predict_mpg = predict(optimal_model, auto_mpg2,interval =
"prediction")
```

- *Kết quả thực nghiệm*

Filter

	fit	lwr	upr
1	29.80024	24.90589	34.69460
2	30.05707	25.20555	34.90859
3	29.54959	24.69867	34.40050
4	31.16159	26.30735	36.01584
5	20.45656	15.64025	25.27288
6	18.24751	13.42935	23.06568
7	19.23263	14.41625	24.04901
8	19.44159	14.62540	24.25779
9	21.71035	16.89160	26.52909
10	22.84472	18.02161	27.66783
11	24.30747	19.47570	29.13924
12	20.06849	15.25241	24.88456
13	21.38197	16.56411	26.19984
14	20.36701	15.55077	25.18325
15	22.21783	17.39739	27.03828
16	18.93411	14.11736	23.75086
17	20.18789	15.37177	25.00402
18	20.09834	15.28225	24.91442

3.7. So sánh kết quả dự báo "predict_mpg" với giá trị thực tế của "mpg". Rút ra nhận xét?

- Code R

```
# them cot gia tri thuc te de so sanh
predict_mgp = data.frame(predict_mgp, auto_mpg2["mgp"])
```

- Kết quả thực nghiệm

	fit	lwr	upr	mgp
1	29.80024	24.90589	34.69460	36.1
2	30.05707	25.20555	34.90859	32.8
3	29.54959	24.69867	34.40050	39.4
4	31.16159	26.30735	36.01584	36.1
5	20.45656	15.64025	25.27288	19.9
6	18.24751	13.42935	23.06568	19.4
7	19.23263	14.41625	24.04901	20.2
8	19.44159	14.62540	24.25779	19.2
9	21.71035	16.89160	26.52909	20.5
10	22.84472	18.02161	27.66783	20.2
11	24.30747	19.47570	29.13924	25.1
12	20.06849	15.25241	24.88456	20.5
13	21.38197	16.56411	26.19984	19.4

Nhận xét:

- Cột fit đại diện cho giá trị biến phụ thuộc ước lượng từ mô hình
- Cột lwr và upr là những giá trị giới hạn khoảng tin cậy 95%
- Cột mgp là giá trị thực tế thu thập được

Kiểm tra và so sánh kết quả dự báo với kết quả thực tế của biến mgp

- Code R

```
suitable = predict_mpg["lwr"] & predict_mpg["mgp"] <= predict_mpg["upr"]
sum(suitable)
sum(!suitable)
```

- Kết quả thực nghiệm

```
> sum(suitable)
[1] 102
> sum(!suitable)
[1] 40
```

- Số giá trị phù hợp: 102
- Số giá trị không phù hợp: 40

Vậy có 72% giá trị phù hợp với mức tin cậy 95%

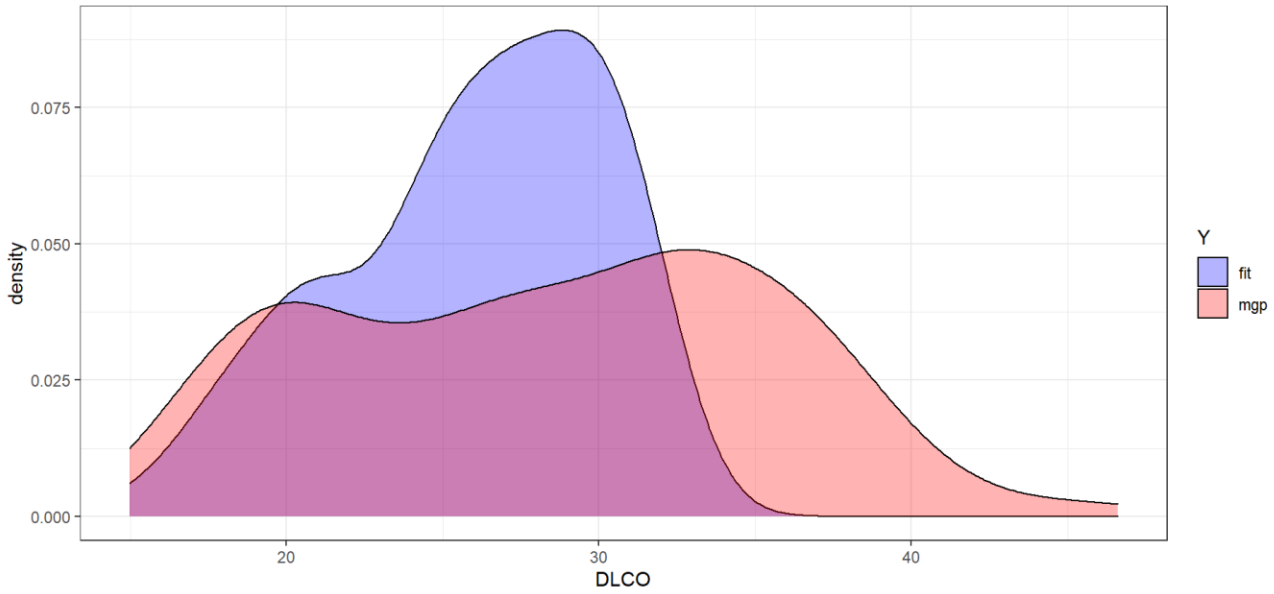
Hình ảnh trực quan dữ liệu:

- Code R

```
predict_mpg%>%gather(mgp,fit,key="Y",value="DLCO")%>%
  ggplot(aes(x=DLCO, fill=Y))+
  geom_density(alpha=0.3)+
  scale_fill_manual(values=c("blue","red"))+
  theme_bw()
```

- Kết quả thực nghiệm

Sự khác biệt giữa dự báo (đồ thị xanh) và thực tế (đồ thị đỏ)


0.075
0.050
0.025
0.000
density
20
30
40
DLCO
Y
fit
mgp

4. PHẦN RIÊNG

Sinh viên tự tìm một bộ dữ liệu thuộc về chuyên ngành của mình. Khuyến khích sinh viên sử dụng dữ liệu thực tế sẵn có từ các thí nghiệm, khảo sát, dự án,... trong chuyên ngành của mình. Ngoài ra sinh viên có thể tự tìm kiếm dữ liệu từ những nguồn khác hoặc tham khảo trong kho dữ liệu cung cấp trong tập tin "kho_du_lieu_BTL_xstk.xlsx".

Sinh viên được tự do chọn phương pháp lý thuyết phù hợp để áp dụng phân tích dữ liệu của mình, nhưng phải đảm bảo 2 phần: Làm rõ dữ liệu (data visualization) và mô hình dữ liệu (model fitting).

Bài tập lớn sử dụng tập tin "2014 and 2015 CSM dataset.xlsx" chứa thông tin về các bộ phim ra mắt trong 2 năm 2014 và năm 2015, có thể truy cập tại https://archive.ics.uci.edu/ml/datasets/CSM+%28Conventional+and+Social+Media+Movies%29+Dataset+2014+and+2015.

Các biến chính trong bộ dữ liệu:

- Genre: Thể loại.
- Gross: Tổng danh thu ($).
- Budget: Chi phí sản xuất ($).
- Screens - Số rạp chiếu phim trong tuần đầu công chiếu.
- Sequel: Số phần.
- Views: Số lượt xem.

4.1. Đọc dữ liệu (Import data)

Ta thực hiện đọc dữ liệu từ file "2014 and 2015 CSM dataset.xlsx":

- Code R

```
new_data <- read_excel("2014 and 2015 CSM dataset.xlsx")
View(new_data)
```

- Kết quả thực nghiệm

	Movie	Genre	Gross	Budget	Screens	Sequel	Views
1	13 Sins	8	9.13e+03	4000000	45	1	3280543
2	22 Jump Street	1	1.92e+08	50000000	3306	2	583289
3	3 Days to Kill	1	3.07e+07	28000000	2872	1	304861
4	300: Rise of an Empire	1	1.06e+08	110000000	3470	2	452917
5	A Haunted House 2	8	1.73e+07	3500000	2310	2	3145573
6	A Long Way Off	3	2.90e+04	500000	*NA*	1	91137
7	A Million Ways to Die in the West	8	4.26e+07	40000000	3158	1	3013011
8	A Most Violent Year	1	5.75e+06	20000000	818	1	1854103
9	A Walk Among the Tombstones	10	2.60e+07	28000000	2714	1	2213659
10	About Last Night	8	4.86e+07	12500000	2253	1	5218079
11	American Sniper	1	3.50e+08	58800000	3555	1	3927600
12	And So It Goes	8	1.52e+07	30000000	1762	1	519327
13	Annabelle	15	8.43e+07	6500000	3185	1	19032902

Showing 1 to 13 of 231 entries, 7 total columns

4.2. Làm sạch dữ liệu (Data cleaning): NA (dữ liệu khuyết)

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 Genre, Sequel, Budget, Screens, Views và Gross.

- Code R

```
data_use <- new_data %>% as_tibble() %>%
select(Genre, Sequel, Budget, Screens, Views, Gross)
```

```
View(data_use)
```

- Kết quả thực nghiệm

Sau khi chạy câu lệnh R để đọc dữ liệu, ta thu được bảng "data_use" như sau:

	Genre	Sequel	Budget	Screens	Views	Gross
1	8	1	4000000	45	3280543	9.13e+03
2	1	2	50000000	3306	583289	1.92e+08
3	1	1	28000000	2872	304861	3.07e+07
4	1	2	110000000	3470	452917	1.06e+08
5	8	2	3500000	2310	3145573	1.73e+07
6	3	1	500000	*NA*	91137	2.90e+04
7	8	1	40000000	3158	3013011	4.26e+07
8	1	1	20000000	818	1854103	5.75e+06
9	10	1	28000000	2714	2213659	2.60e+07
10	8	1	12500000	2253	5218079	4.86e+07
11	1	1	58800000	3555	3927600	3.50e+08
12	8	1	30000000	1762	519327	1.52e+07
13	15	1	6500000	3185	19032902	8.43e+07

Showing 1 to 13 of 231 entries, 6 total columns

Sau đó ta tiến hành tìm các dữ liệu khuyết (N/A):

- Code R

```
apply(is.na(data_use), 2, which)
```

- Kết quả thực nghiệm

```
$Genre
integer(0)

$Sequel
integer(0)

$Budget
[1] 121

$Screens
 [1]   6  25  33  39  68  85  96 116 129 230

$Views
integer(0)

$Gross
integer(0)
```

Cuối cùng, ta xử lý các dữ liệu khuyết để tạo thành bảng dữ liệu hoàn chỉnh.

Có 3 cách xử lý giá trị khuyết:

- Xóa bỏ giá trị N/A đó khỏi tập dữ liệu.
- Thay giá trị N/A đó bằng mean, median, mode của các số liệu khác cùng tập dữ liệu có liên quan tới dữ liệu đó.
- Sử dụng mô hình dự đoán cho dữ liệu khuyết.

Ở đây, ta thấy có 10/217 dữ liệu bị khuyết, chiếm 4,6% tổng số dữ liệu, nên ta sẽ xử lý như sau:

- Đối với các biến liên tục, thay dữ liệu khuyết bằng giá trị mean.
- Đối với các biến phân loại, thay dữ liệu khuyết bằng giá trị median.

- Code R

```
for(i in c(1:2)) {
   data_use[is.na(data_use[,i]), i] <-
   median(as.numeric(unlist(data_use[,i])), na.rm = TRUE)
}
```

```
for(i in c(3:6)) {
   data_use[is.na(data_use[,i]), i] <-
   mean(as.numeric(unlist(data_use[,i])), na.rm = TRUE)
}
View(data_use)
```

- Kết quả thực nghiệm

	Genre	Sequel	Budget	Screens	Views	Gross
1	8	1	4000000	45.000	3280543	9.13e+03
2	1	2	50000000	3306.000	583289	1.92e+08
3	1	1	28000000	2872.000	304861	3.07e+07
4	1	2	110000000	3470.000	452917	1.06e+08
5	8	2	3500000	2310.000	3145573	1.73e+07
6	3	1	500000	2209.244	91137	2.90e+04
7	8	1	40000000	3158.000	3013011	4.26e+07
8	1	1	20000000	818.000	1854103	5.75e+06
9	10	1	28000000	2714.000	2213659	2.60e+07
10	8	1	12500000	2253.000	5218079	4.86e+07
11	1	1	58800000	3555.000	3927600	3.50e+08
12	8	1	30000000	1762.000	519327	1.52e+07
13	15	1	6500000	3185.000	19032902	8.43e+07

Showing 1 to 13 of 231 entries, 6 total columns

4.3. Làm rõ dữ liệu: (Data visualization)

4.3.1 Thống kê dữ liệu

Đầu tiên, ta tiến hành thống kê các biến liên tục.

Dựa vào dữ liệu đã được cung cấp, nhận thấy các biến liên tục gồm Budget, Screens, Views và Gross, ta có lời giải R sau:

- Code R

```
cot <- c("mean", "median", "sd", "min", "max")
hang <- c("Budget", "Screens", "Views", "Gross")
tp <- c()
for (i in hang) {
cotm <- c(mean(data_use[[i]]), median(data_use[[i]]), sd(data_use[[i]]),
min(data_use[[i]]), max(data_use[[i]]))
tp <- rbind(tp,cotm)
}

tp <- as.data.frame(tp)
colnames(tp) <- cot
rownames(tp) <- hang
data_analysis <- tp
View(data_analysis)
```

- Kết quả thực nghiệm

Sau khi chạy câu lệnh R để thống kê dữ liệu, ta thu được bảng "data_analysis":

	mean	median	sd	min	max
Budget	47921730.047	28000000	54170099.586	70000	250000000
Screens	2209.244	2757	1431.593	2	4324
Views	3712851.290	2409338	4511104.243	698	32626778
Gross	68066033.203	37400000	88902891.222	2470	643000000

Tương tự như trên, nhận thấy các biến phân loại bao gồm Genre và Sequel, ta dùng R để thống kê các biến phân loại:

- Code R

```
b1_3.1.2 = table(data_use$Genre, dnn = "Genre")
View(b1_3.1.2)
```

```
b2_3.1.2 = table(data_use$Sequel, dnn = "Sequel")
View(b2_3.1.2)
```

- Kết quả thực nghiệm

Sau khi chạy code R, ta thu được bảng thống kê các biến phân loại như sau:

	Genre	Freq
1	1	65
2	2	12
3	3	46
4	4	1
5	6	3
6	7	2
7	8	54
8	9	13
9	10	12
10	12	13
11	15	10

	Sequel	Freq
1	1	188
2	2	25
3	3	8
4	4	3
5	5	4
6	6	1
7	7	2

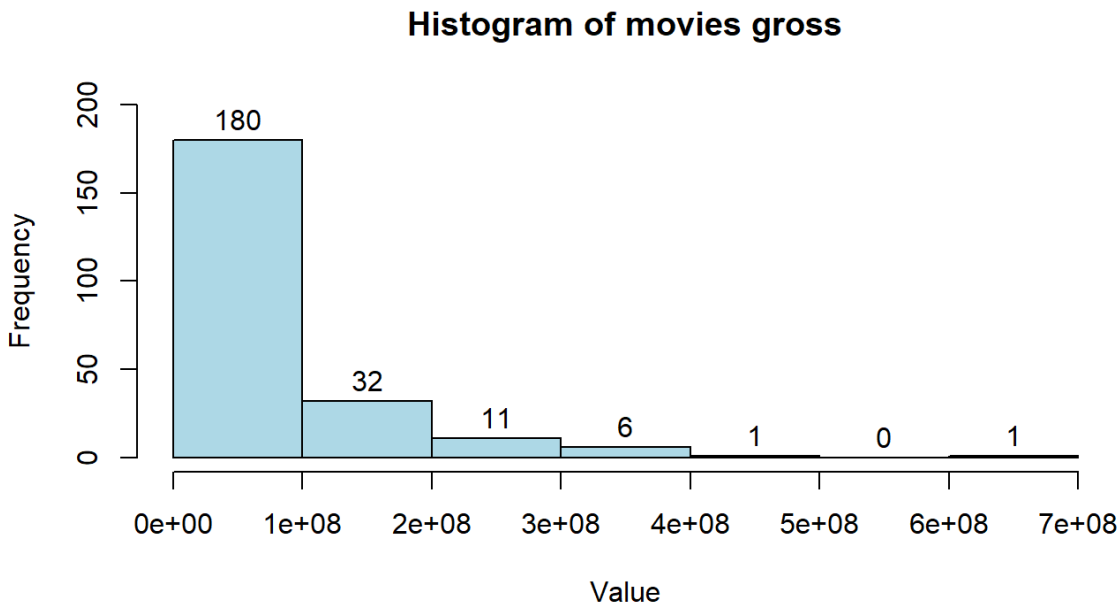
4.3.2. Vẽ đồ thị

Để có thể nhận xét rõ hơn về doanh thu của các bộ phim, ta sử dụng code R để vẽ biểu đồ nhận xét:

- Code R

```
hist(data_use$Gross, xlab = "Value", main = "Histogram of movies gross",
    ylim=c(0,200), breaks = 7, labels = T, col = "lightblue")
```

- Kết quả thực nghiệm



Nhìn vào biểu đồ ta có thể thấy được:

- Từ 0 đến 100 triệu đô có đến 180 bộ phim, chiếm tỉ trọng lớn nhất trong biểu đồ.
- Bên cạnh đó, không có bộ phim nào có mức doanh thu từ 500 đến 600 triệu đô.

Để nhận xét mối tương quan giữa Gross với các yếu tố khác, ta dùng code R để vẽ các biểu đồ boxplot tương ứng:

Boxplot: Biểu đồ hộp (Boxplot) hay còn gọi là biểu đồ hộp và râu (Box and whisker plot) là biểu đồ diễn tả 5 vị trí phân bố của dữ liệu, đó là: giá trị nhỏ nhất (min), tứ phân vị thứ nhất (Q1), trung vị (median), tứ phân vị thứ 3 (Q3) và giá trị lớn nhất (max).

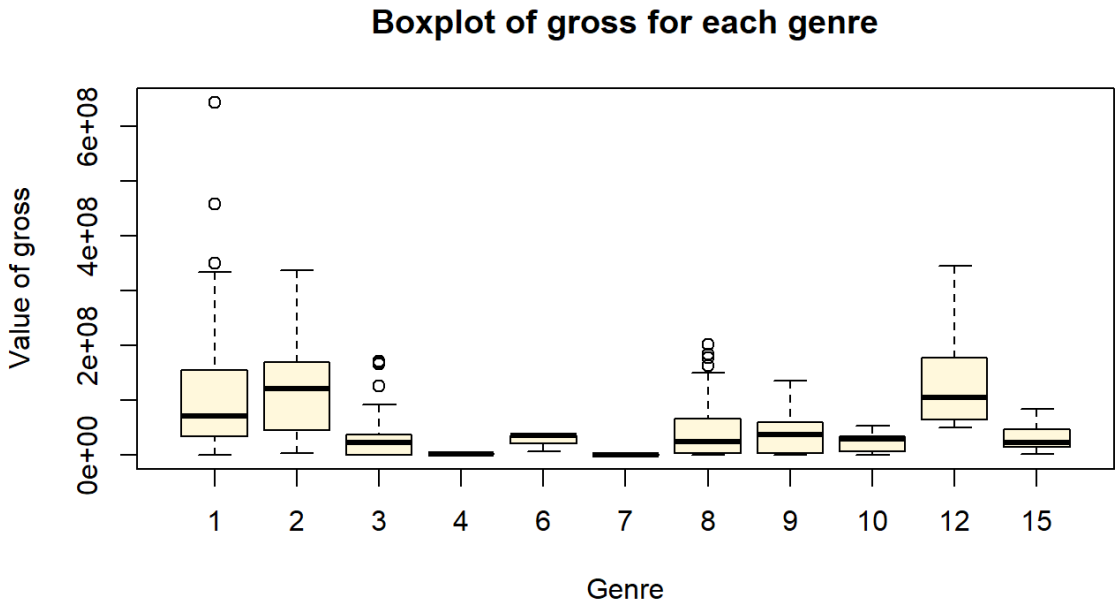
- Code R

```
boxplot(data = data_use, Gross ~ Genre, col = "cornsilk",
        main = "Boxplot of gross for each genre",
        xlab = "Genre", ylab = "Value of gross")
boxplot(data = data_use, Gross ~ Sequel, col = "darkgoldenrod1",
```

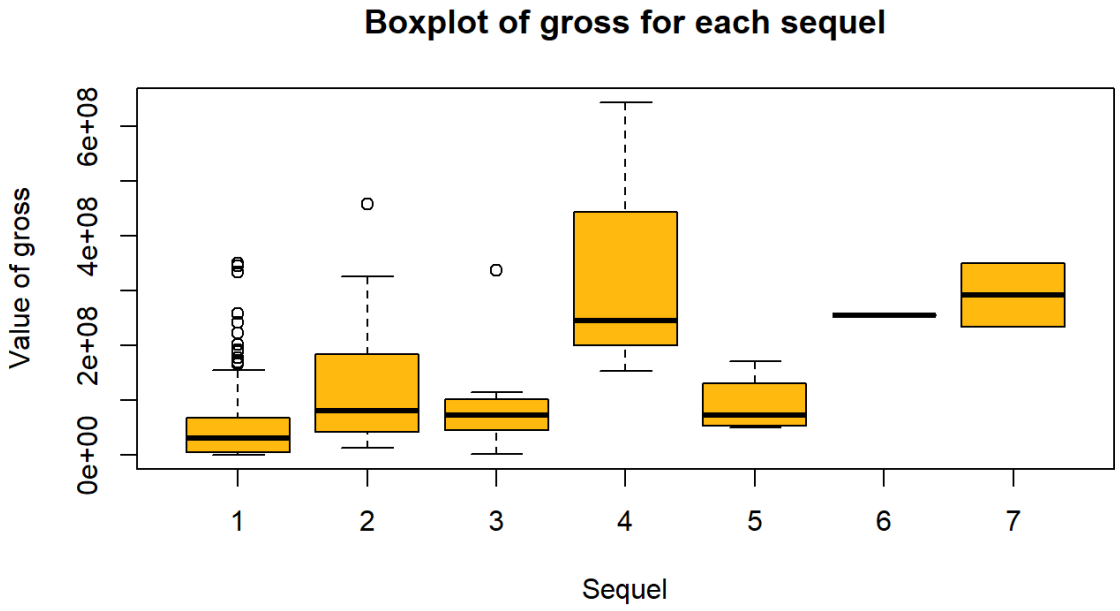
```
        main = "Boxplot of gross for each sequel",
        xlab = "Sequel", ylab = "Value of gross")
```

- Kết quả thực nghiệm

+ Biểu đồ tương quan giữa Gross với biến Genre:



+ Biểu đồ tương quan giữa Gross với biến Sequel:



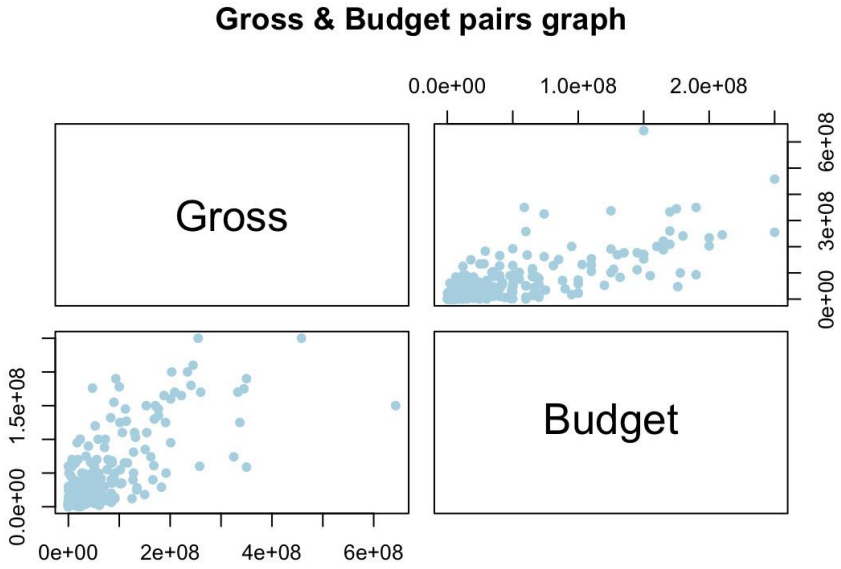
Để xét phân phối của biến Gross theo các biến Budget, Views và Screens, ta có thể dùng lệnh pairs() để vẽ các biểu đồ:

- Code R

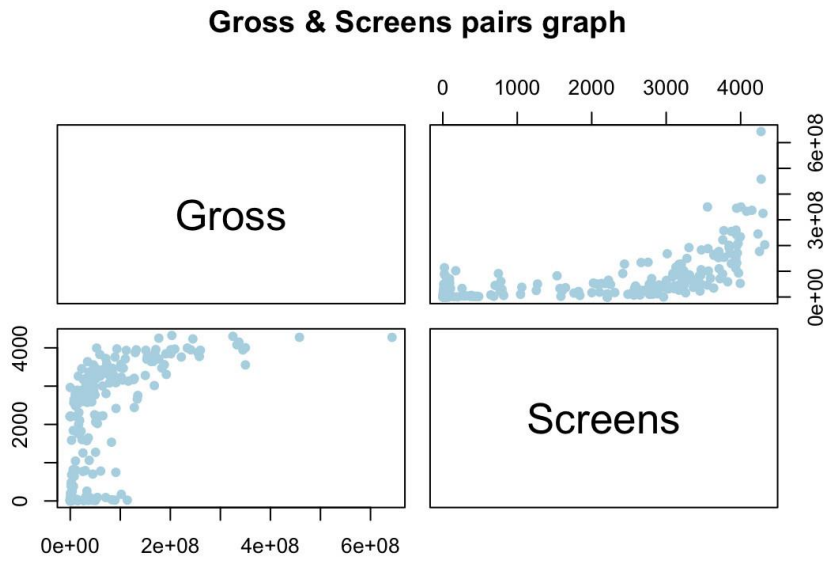
```
boxplot(data = data_use, Gross ~ Genre, col = "cornsilk",
        main = "Boxplot of gross for each genre",
        xlab = "Genre", ylab = "Value of gross")
boxplot(data = data_use, Gross ~ Sequel, col = "darkgoldenrod1",
        main = "Boxplot of gross for each sequel",
        xlab = "Sequel", ylab = "Value of gross")
```

- Kết quả thực nghiệm

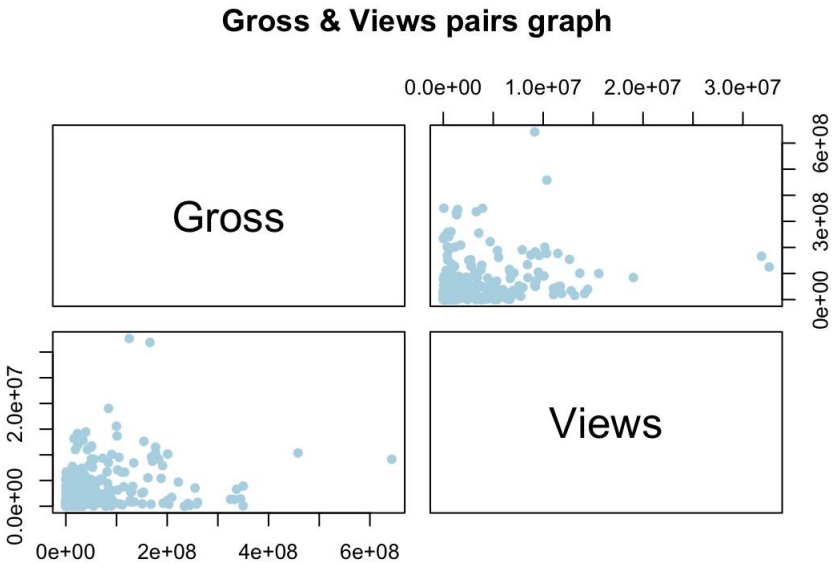
+ Biểu đồ phân phối của biến Gross theo biến Budget:



+ Biểu đồ phân phối của biến Gross theo biến Screens:



+ Biểu đồ phân phối của biến Gross theo biến Views:



4.4. Chọn mô hình hồi quy bội

4.4.1. Chọn mô hình tối ưu

- X: biến tiên lượng cho biến phụ thuộc Y.
- Chọn ra 5 mô hình tối ưu nhất

- p!=0: xác suất hồi quy khác 0,và khác 0 thì nó sẽ ảnh hưởng đến biến phụ thuộc
- EV: giá trị kì vọng của các hệ số hồi quy
- SD: độ lệch chuẩn
- Biến nào không có trong mô hình nào thì giá trị tương ứng trong mô hình đó là '.'
- nVar: là số biến xuất hiện trong mô hình.
- BIC: giá trị BIC càng thấp thì mô hình càng tốt.
- r2 : giải thích phần trăm số dao động.
- post prob: xác suất xuất hiện của mô hình.

- *Code R:*

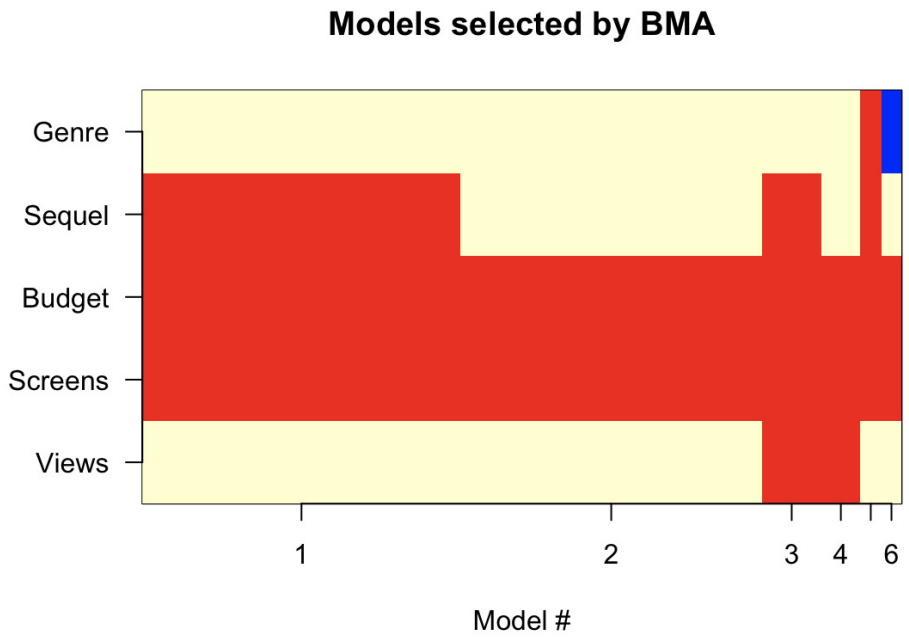
Để dùng được phương pháp bayes ta cần cài đặt thư viện (BMA)

```
library(BMA)
X = data_use[, c('Genre', 'Sequel', 'Budget', 'Screens', 'Views')]
Y = data_use$Gross

result <- bicreg(X, Y, strict = FALSE, OR = 20)
summary(result)
par(mar=c(3,3,3,3))
imageplot.bma(result)
```

- *Kết quả thực nghiệm:*

```
  6  models were selected
 Best  5  models (cumulative posterior probability =  0.9731 ):

           p!=0    EV          SD          model 1     model 2     model 3
Intercept  100.0  -1.624e+07  9.695e+06  -2.083e+07  -1.049e+07  -2.403e+07
Genre        5.5  -4.691e+03  2.327e+05        .           .           .
Sequel      52.5   5.658e+06  6.311e+06   1.067e+07        .       1.139e+07
Budget     100.0   9.011e-01  1.034e-01   8.585e-01   9.479e-01   8.580e-01
Screens    100.0   1.489e+04  3.397e+03   1.505e+04   1.499e+04   1.391e+04
Views       13.0   1.534e-01  5.141e-01        .           .       1.282e+00

nVar                                          3           2           4
r2                                         0.566       0.555       0.570
BIC                                     -1.764e+02  -1.763e+02  -1.730e+02
post prob                                  0.419       0.397       0.078
```

Model 1, model 2, model 3, model 4, model 5 là những mô hình tối ưu nhất theo phương pháp BMA. Mô hình 1 có xác suất xuất hiện 41.9% (post prob = 0.419), cao nhất trong tất cả các mô hình, BIC thấp nhất (-1.764*10^-2), cùng với đó mô hình 1 có 3 biến xuất hiện trong mô hình (nVar = 3) nhưng giải thích được 56.6 % (r2 = 0,566) mức ý nghĩa, do đó nhóm chọn mô hình tối ưu là mô hình 1.



Xác suất xuất hiện của các biến trong các mô hình

Màu xanh thể hiện biến có hệ số hồi quy < 0, màu đỏ thể hiện biến có hệ số hồi quy > 0, màu vàng nhạt thể hiện biến không xuất hiện trong mô hình.

4.4.2. Mô hình hồi quy bội

- *Code R:*

```
# mo hinh toi uu se loai bo cac bien Genre, Views
optimal_model <-lm(Gross ~ Sequel + Budget + Screens, data =
data_use)
summary(optimal_model)
```

- *Kết quả thực nghiệm*

```
Residuals:
       Min         1Q     Median         3Q        Max
-141421941  -28037744   -2978866   15440139  428048066

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.083e+07  8.394e+06  -2.482   0.0138 *
Sequel       1.067e+07  4.540e+06   2.350   0.0197 *
Budget       8.585e-01  9.638e-02   8.908  < 2e-16 ***
Screens      1.505e+04  3.351e+03   4.493 1.12e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 58970000 on 227 degrees of freedom
Multiple R-squared:  0.5658,    Adjusted R-squared:  0.5601
F-statistic: 98.61 on 3 and 227 DF,  p-value: < 2.2e-16
```

Nhận xét:

- Mô hình không còn những biến độc lập không có ý nghĩa thống kê theo kiểm định.
- Các giá trị Pr(>F) của các biến độc lập nhỏ hơn mức ý nghĩa. Điều này mang ý nghĩa mô hình có thể chấp nhận được.
- Ngoài ra ta còn có thể sử dụng trị số r^2 (Multiple R- squared) hay hệ số xác định bộI (coefficient of determination). Ở mô hình trên, giá trị này là 0.5658 tức có

nghĩa là phương trình tuyến tính thu được giải thích khoảng 56,58% các khác biệt về APM giữa các cá nhân. Trị số này có giá trị trong khoảng từ 0 đến 1 và trị số này càng cao thì là một dấu hiệu cho thấy mối liên hệ giữa biến phụ thuộc và các biến độc lập trong mô hình càng chặt chẽ.

Ta có phương trình hồi quy tuyến tính như sau:

$$Gross = \alpha + \beta1 * Genre + \beta2 * Sequel + \beta3 * Budget + \beta4 * Screens + \beta5 * Views$$

Các hệ số chính của mô hình:

α = -2.083e+07; $\beta2$ = 1.067e+07; $\beta3$ = 8.585e-01; $\beta4$ = 1.505e+04

Vậy:

Gross =-2.083e+07 + 1.067e+07*Genre + 8.585e-01* Budget + 1.505e+04* Screens

Kết luận:

- Biến Sequel, Budget và Screens có tương quan dương với biến phụ thuộc Gross (Vì hệ số hồi quy của 3 biến này > 0).

Nhận xét:

- Khi giá trị Sequel tăng 1 thì lượng Gross sẽ tăng lên 1.067e+07 lần trong điều kiện các yếu tố khác không đổi (Tương tự với 2 biến còn lại).

4.5. Kiểm tra các giả định (giả thiết) của mô hình.

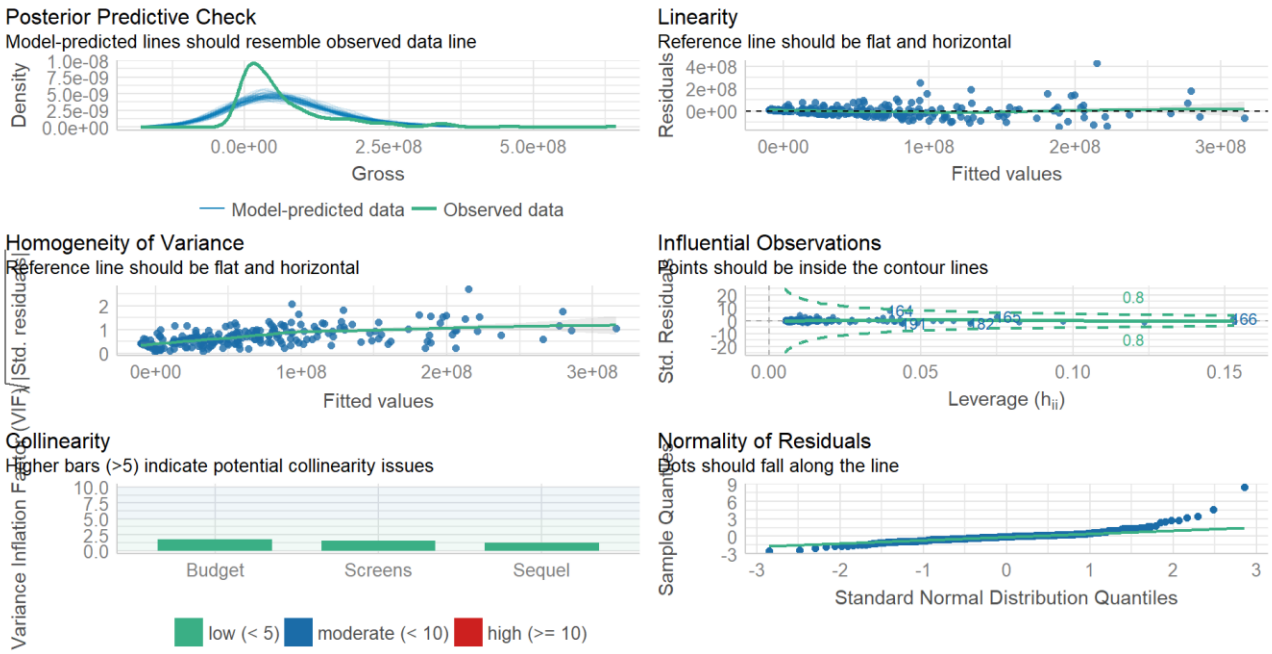
- *Các gói cần cài đặt*

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

- *Code R*

```
library(performance)
check_model(optimal_model)
```

- *Kết quả thực nghiệm*



Giả định: Liên hệ tuyến tính giữa biến phụ thuộc với biến độc lập

Trên *biểu đồ Linearity* (biểu đồ trên cùng bên phải), phần dư chuẩn hóa phân bố ngẫu nhiên tập trung xung quanh đường tung độ 0 tạo thành dạng đường thẳng, do vậy giả định quan hệ tuyến tính giữa biến phụ thuộc với các biến độc lập không bị vi phạm.

Giả định: Phương sai phần dư không thay đổi

Kết quả từ *biểu đồ Homogeneity of variance* (biểu đồ trên ở giữa bên trái) cho thấy, các điểm phân vị dao động khá đồng đều trên dưới trục tung độ 0. Các điểm phân vị hầu như dọc theo trục tung độ 0. Do đó, giả định phương sai phần dư đồng nhất không bị vi phạm.

Giả định: Không có đa cộng tuyến

Biểu đồ Collinearity (biểu đồ ở dưới cùng bên trái), với các biến độc lập có giá trị VIF nhỏ hơn 2.5, các giá trị VIF này ở mức thấp, do đó không có hiện tượng đa cộng tuyến.

Giả định: Không có điểm ảnh hưởng

Kết quả từ *biểu đồ Influential Observations* (biểu đồ ở giữa bên phải) cho thấy, các điểm phân vị dao động xung quanh đường hồi quy, không có điểm ngoại lai. Do đó, giả định không có điểm ảnh hưởng không bị vi phạm.

Giả định: Phân phối chuẩn của phần dư

Biểu đồ Normality of Residuals (biểu đồ dưới cùng bên phải) có các điểm phân vị trong phân phối của phần dư tập trung thành 1 đường chéo, nghĩa là phần dư có phân phối chuẩn. Như vậy, giả định phân phối chuẩn của phần dư không bị vi phạm.

4.5. Dự đoán

4.5.1. Dự đoán tỉ lệ

Dự đoán tỉ lệ đạt doanh thu cao hoặc không cao.

- *Code R*

```
High_gross <- function(x){
     if(x >= 100000000) return("High gross")
     else return("Not high gross")
}
ket_qua <- c(apply(data_use["Gross"], MARGIN = 1, FUN = High_gross))
data_use <- cbind (data_use, ket_qua)
view(data_use)
```

- *Kết quả thực nghiệm*

	Genre	Sequel	Budget	Screens	Views	Gross	ket_qua
1	8	1	4000000	45.000	3280543	9.13e+03	Not high gross
2	1	2	50000000	3306.000	583289	1.92e+08	High gross
3	1	1	28000000	2872.000	304861	3.07e+07	Not high gross
4	1	2	110000000	3470.000	452917	1.06e+08	High gross
5	8	2	3500000	2310.000	3145573	1.73e+07	Not high gross
6	3	1	500000	2209.244	91137	2.90e+04	Not high gross
7	8	1	40000000	3158.000	3013011	4.26e+07	Not high gross
8	1	1	20000000	818.000	1854103	5.75e+06	Not high gross
9	10	1	28000000	2714.000	2213659	2.60e+07	Not high gross
10	8	1	12500000	2253.000	5218079	4.86e+07	Not high gross
11	1	1	58800000	3555.000	3927600	3.50e+08	High gross
12	8	1	30000000	1762.000	519327	1.52e+07	Not high gross
13	15	1	6500000	3185.000	19032902	8.43e+07	Not high gross

Showing 1 to 13 of 231 entries, 7 total columns

4.5.2. Thống kê tỉ lệ

- *Code R*

```
ti_le = prop.table(table(data_use$Gross >= 100000000))
ti_le
```

- *Kết quả thực nghiệm*

```
    FALSE      TRUE
0.7748918 0.2251082
```

Vậy tỉ lệ phim có doanh thu cao (> 10000000$) là 22.5.% và tỉ lệ phim có doanh thu không cao (<= 10000000$) là 77.5%

4.5.3. Kiểm định kết quả dự đoán thông qua mô hình hồi quy

- *Code R*

```
newtab <- data_use %>% as_tibble() %>% select(Gross, Sequel, Budget,
Screens, Views)

pred_Gross <- predict(optimal_model)
newtab <- cbind(newtab, pred_Gross)
view(newtab)
```

- *Kết quả thực nghiệm*

	Gross	Sequel	Budget	Screens	Views	pred_Gross
1	9.13e+03	1	4000000	45.000	3280543	-6055275.6
2	1.92e+08	2	50000000	3306.000	583289	93194526.4
3	3.07e+07	1	28000000	2872.000	304861	57107434.1
4	1.06e+08	2	110000000	3470.000	452917	147175340.5
5	1.73e+07	2	3500000	2310.000	3145573	38278890.4
6	2.90e+04	1	500000	2209.244	91137	23520605.8
7	4.26e+07	1	40000000	3158.000	3013011	71715294.3
8	5.75e+06	1	20000000	818.000	1854103	19318059.5
9	2.60e+07	1	28000000	2714.000	2213659	54728886.7
10	4.86e+07	1	12500000	2253.000	5218079	34481696.2
11	3.50e+08	1	58800000	3555.000	3927600	93832179.4
12	1.52e+07	1	30000000	1762.000	519327	42114450.9
13	8.43e+07	1	6500000	3185.000	19032902	43360920.5

Showing 1 to 13 of 231 entries, 6 total columns

4.5.4. So sánh kết quả

- *Code R*

```
ti_le1 = prop.table(table(newtab$pred_Gross >= 100000000))
So_sanh <- data.frame(cbind(ti_le, ti_le1))
```

```
colnames(So_sanh) <- c("Thuc te", "Du doan")
rownames(So_sanh) <- c("Not high gross", "High gross")
View(So_sanh)
```

- *Kết quả thực nghiệm*

	Thuc te	Du doan
Not high gross	0.7748918	0.7705628
High gross	0.2251082	0.2294372

Nhận xét:

Nhìn vào bảng kết quả so sánh giữa thực tế và dự đoán dựa trên mô hình hồi quy tuyến tính, ta có thể thấy 2 giá trị không có sự chênh lệch đáng kể. Sự chênh lệch nhỏ này có thể do sự ảnh hưởng của các giá trị ngoại lai làm ảnh hưởng đến kết quả dự đoán.

KẾT LUẬN

Bài tập lớn sử dụng các kiến thức cơ bản đã học trên lớp kèm theo các tài liệu sinh viên tự tìm hiểu thêm về hồi quy tuyến tính bội và ngôn ngữ R. Trong bài sử dụng ngôn ngữ R để phân tích dữ liệu, xây dựng mô hình hồi quy, đưa ra các dự đoán liên quan và kiểm định độ chính xác của chúng.

Thông qua bài tập lớn này, chúng ta đã có thể phát huy được khả năng lĩnh hội kiến thức môn Xác suất và thống kê của mình trong suốt 1 học kỳ vừa qua, nâng cao tinh thần tự giác, tự tìm tòi và học hỏi của sinh viên. Trong suốt quá trình làm việc, các thành viên trong nhóm đoàn kết, giúp đỡ lẫn nhau, không xảy ra mâu thuẫn hay tranh chấp, nâng cao kĩ năng làm việc nhóm của mỗi người.

Bên cạnh đó, chúng ta còn học được cách sử dụng ngôn ngữ R để phân tích và kiểm định dữ liệu, thống kê và đưa ra các dự đoán hợp lý.

TÀI LIỆU THAM KHẢO

[1] Nguyễn Đình Huy, Giáo trình xác suất thống kê, lần 9, 2018. NXB Đại học Quốc gia TP.HCM.

[2] Nguyễn Đình Huy, Slide bài giảng trên lớp.

[3] XuLyDinhLuong.com, (5/1/2020), Kiểm tra vi phạm các giả định hồi quy trong SPSS, Truy cập từ: https://xulydinhluong.com/kiem-tra-vi-pham-cac-gia-dinh-hoi-quy-trong-spss/

[4] R-bloggers, (3/10/2021), Multiple linear regression made simple, Truy cập từ: https://www.r-bloggers.com/2021/10/multiple-linear-regression-made-simple/

[5] MOSL, Hồi quy tuyến tính | Mô hình OLS – Cách đọc kết quả Stata, Truy cập từ: https://mosl.vn/mo-hinh-hoi-quy-tuyen-tinh/