

**ĐẠI HỌC QUỐC GIA**

**ĐẠI HỌC BÁCH KHOA TP HỒ CHÍ MINH**

🙞∙∙∙☼∙∙∙🙜



**BÀI TẬP LỚN MÔN XÁC SUẤT THỐNG KÊ**

**LỚP L14 --- NHÓM 11 --- HK 201**

**Giảng viên hướng dẫn: Nguyễn Kiều Dung**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Sinh viên thực hiện** | **Mã số sinh viên** | **Lớp** | **Ngành học** | **Ký tên tham dự** |
| 1 | Vũ Đình Phú | 1914674 | L06 | Khoa học Máy Tính |  |
| 2 | Lâm Thiện Toàn | 1915540 | L06 | Khoa học Máy Tính |  |
| 3 | Ngô Đức Trí | 1915656 | L06 | Khoa học Máy Tính |  |
| 4 | Lý Thanh Bách | 1910038 | L14 | Khoa học Máy Tính |  |
| 5 | Ngô Thị Hà Bắc | 1912700 | L14 | Khoa học Máy Tính |  |
| 6 | Hoàng Đình Thành | 1915130 | L14 | Khoa học Máy Tính |  |
| 7 | **Huỳnh Đức Thịnh** | 1910563 | L14 | Khoa học Máy Tính |  |
| 8 | Nguyễn Văn Xuân Vũ | 1915982 | L14 | Khoa học Máy Tính |  |

*Thành phố Hồ Chí Minh – 2020*

1. **PHẦN CHUNG:**
2. **Hồi quy tuyến tính bội:**

**Bài tập 1.** Tập tin "gia\_nha.csv" chứa thông tin về giá bán ra thị trường (đơn vị đô la) của 21613 ngôi nhà ở quân King nước Mỹ trong khoảng thời gian từ tháng 5/2014 đến 5/2015. Bên cạnh giá nhà, dữ liệu còn bao gồm các thuộc tính mô tả chất lượng ngôi nhà. Dữ liệu gốc được cung cấp tại: <https://www.kaggle.com/harlfoxem/housesalesprediction>.

Các biến chính trong bộ dữ liệu:

• **price**: Giá nhà được bán ra.

• **sqft\_living15**: Diện tích trung bình của 15 ngôi nhà gần nhất trong khu dân cư.

• **floors**: Số tầng của ngôi nhà được phân loại từ 1-3.5.

• **condition**: Điều kiện kiến trúc của ngôi nhà từ 1 − 5, 1: rất tệ và 5: rất tốt.

• **sqft\_above**: Diện tích ngôi nhà.

• **sqft\_living**: Diện tích khuôn viên nhà.

**Bài làm**

Thiết lập thư viện dùng cho cả bài:

library(stats)

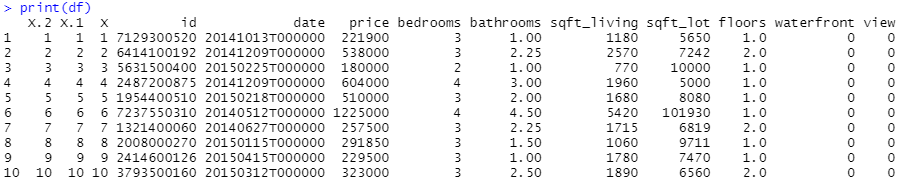
library(dplyr)

library(magrittr)

1. **Đọc dữ liệu (Import Data):**

Code: df = read.csv('gia\_nha.csv');

Output: In dataframe **df** được kết quả như sau:



Lưu ý: Tổng cộng dataframe **df** có 21572 dòng. Dataframe **df** còn các cột condition, grade, sqft\_above không đủ vị trí để hiển thị.

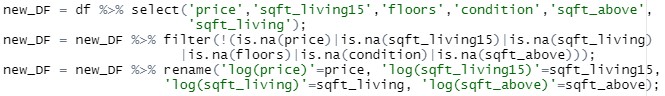
1. **Làm sạch dữ liệu (Data cleaning):**
2. Trích ra một dữ liệu con đặt tên là **new\_DF** chỉ bao gồm các biến chính mà ta quan

tâm như đã trình bày trong phần giới thiệu dữ liệu.

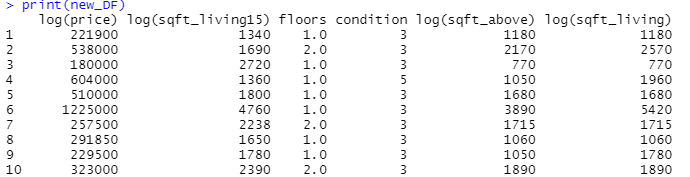
1. Kiểm tra các dữ liệu bị khuyết trong tập tin. Nếu có dữ liệu bị khuyết, đề xuất phương

pháp thay thế cho những dữ liệu bị khuyết này.

Code:



Output: In dataframe **new\_DF** nhận được kết quả như sau



Lưu ý: Tổng cộng dataframe **new\_DF** có 21427 dòng.

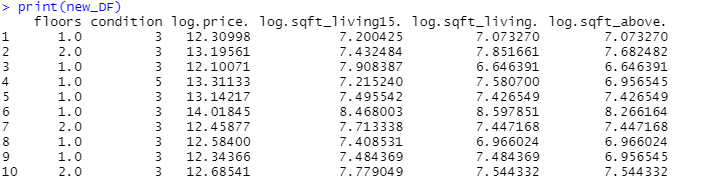
1. **Làm rõ dữ liệu (Data visualization):**
2. Chuyển đổi các biến **price**, **sqft\_living15**, **sqft\_above**, **sqft\_living** lần lượt thành

log(*price*), log(*sqft\_living15*), log(*sqft\_above*), và log(*sqft\_living*).

Code:



Output: In dataframe **new\_DF** nhận được kết quả như sau:



1. Đối với các biến liên tục, tính các giá trị thống kê mô tả bao gồm: trung bình, trung vị

độ lệch chuẩn, giá trị lớn nhất và giá trị nhỏ nhất. Xuất kết quả dưới dạng bảng.

Code:

table = data.frame(Name = c('mean','median','sd','min','max'),

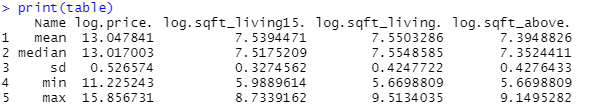
log.price.= c(mean(new\_DF[,'log.price.']), median(new\_DF[,'log.price.']),sd(new\_DF[,'log.price.']),min(new\_DF[,'log.price.']),max(new\_DF[,'log.price.'])),

log.sqft\_living15.= c(mean(new\_DF[,'log.sqft\_living15.']), median(new\_DF[,'log.sqft\_living15.']),sd(new\_DF[,'log.sqft\_living15.']),min(new\_DF[,'log.sqft\_living15.']),max(new\_DF[,'log.sqft\_living15.'])),

log.sqft\_living.= c(mean(new\_DF[,'log.sqft\_living.']), median(new\_DF[,'log.sqft\_living.']),sd(new\_DF[,'log.sqft\_living.']),min(new\_DF[,'log.sqft\_living.']),max(new\_DF[,'log.sqft\_living.'])),

log.sqft\_above.= c(mean(new\_DF[,'log.sqft\_above.']), median(new\_DF[,'log.sqft\_above.']),sd(new\_DF[,'log.sqft\_above.']),min(new\_DF[,'log.sqft\_above.']),max(new\_DF[,'log.sqft\_above.']))

Output: In dataframe **table** nhận được kết quả như sau

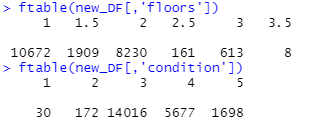


1. Lập bảng tần số đối với các biến phân loại:

Code:

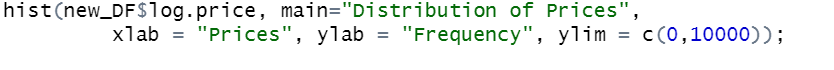


Output:

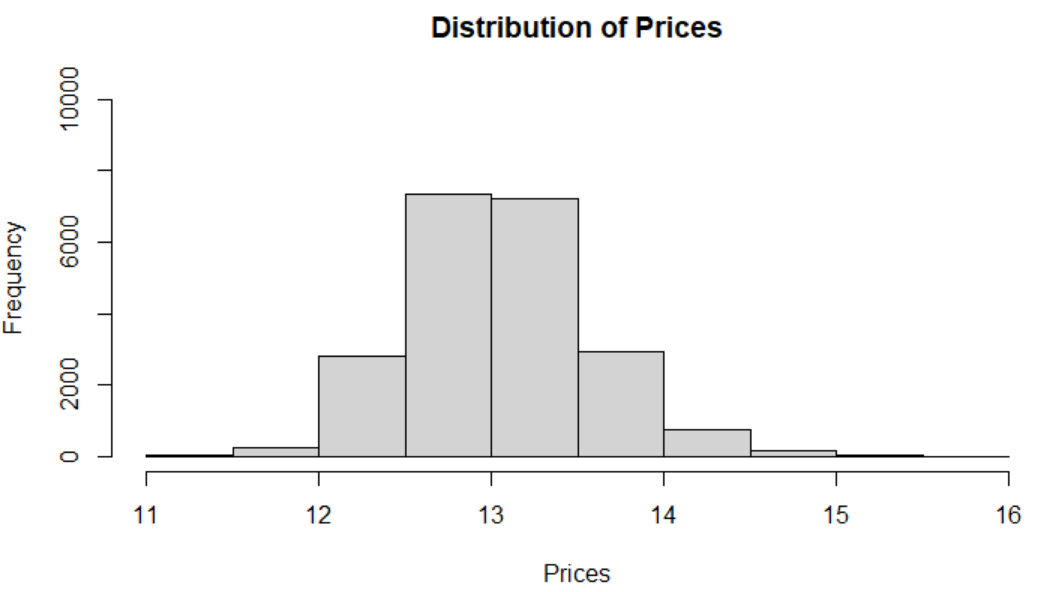


1. Dùng hàm **hist()** để vẽ đồ thị phân phối của biến **price**.

Code:



Output:



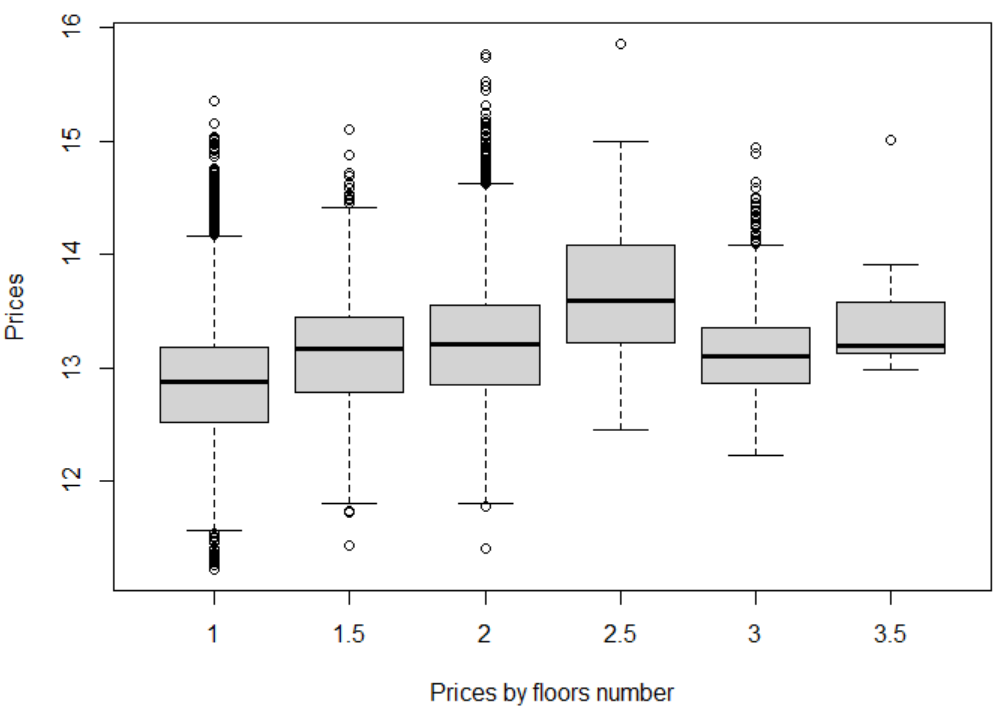
1. Dùng hàm **boxplot()** vẽ phân phối của biến **price** cho từng nhóm phân loại của biến

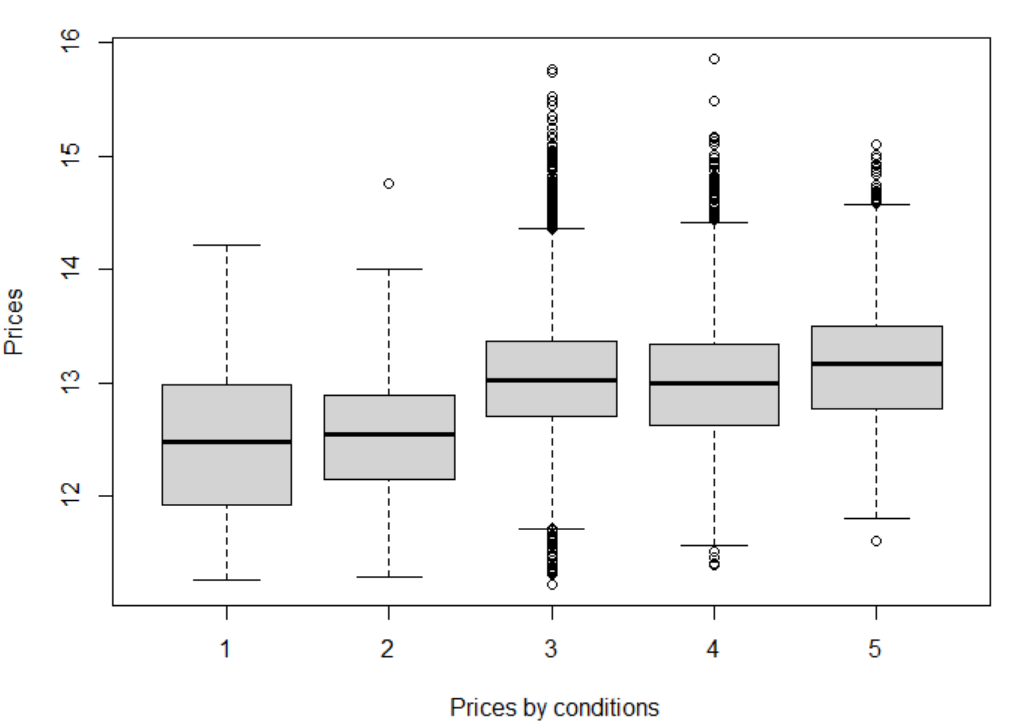
**floors** và biến **condition**.

Code:



Output:





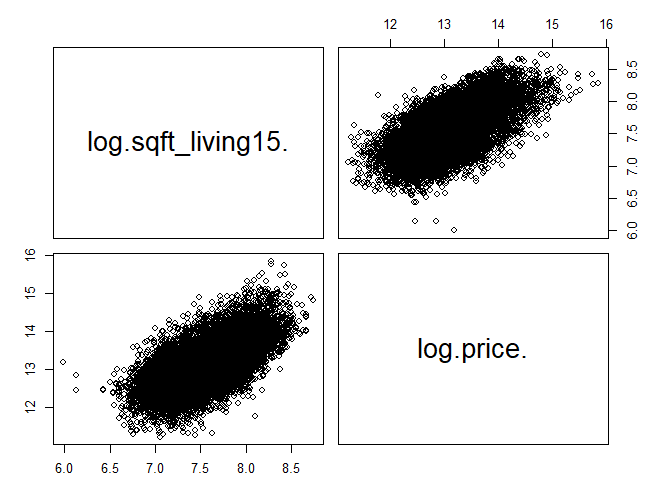
1. Dùng lệnh **pairs()** vẽ các phân phối của biến **price** lần lượt theo các biến **sqft\_living15**,

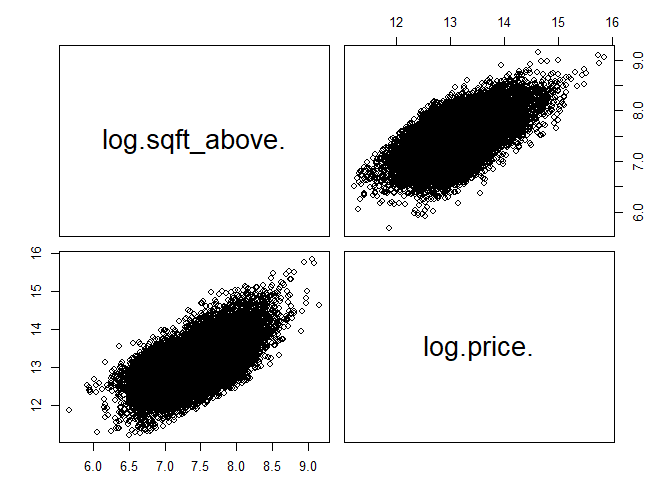
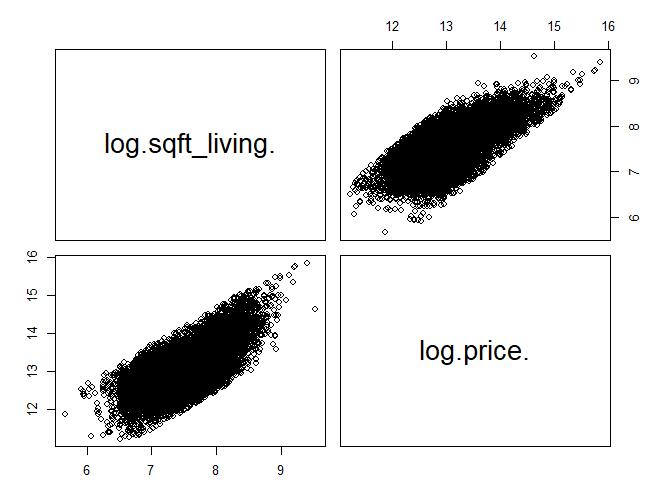
**sqft\_above**, và **sqft\_living**.

Code:



Output: (tượng trưng cho biến **log.sqft\_living15**)





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

Chúng ta muốn khám phá rằng có những nhân tố nào và tác động như thế nào đến giá nhà ở quận King.

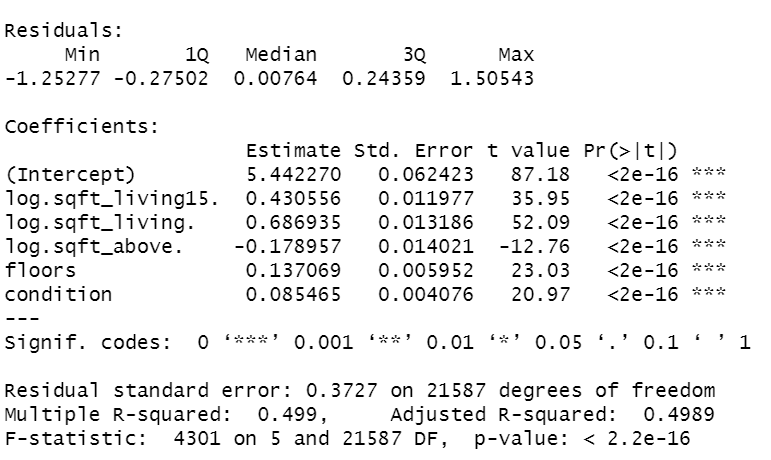
1. Xét mô hình hồi quy tuyến tính bao gồm biến **price** là một biến phụ thuộc, và tất cả

các biến còn lại đều là biến độc lập. Hãy dùng lệnh **lm()** để thực thi mô hình hồi quy tuyến tính bội.

Code:



Thu được kết quả **lmPrice** chứa thông tin, các tham số cho chương trình hồi quy, gọi hàm **summary():**



Các thông tin:

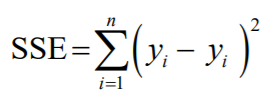
- **Residual**: thống kê thông tin về độ lệch so sánh giữa hàm giả thiết và dữ liệu cho trước

- **Coffecients**:

+ **Estimate**: các hệ số của phương trình hồi quy

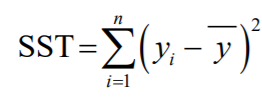
Price = 5.4423 + 0.4306 \* sqft\_living15 + 0.6869 \* sqft\_living – 0.1789 \* sqft\_above + 0.1371 \* floors + 0.085465 \* condition

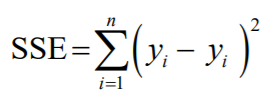
+ **Residual Standard Error**: Sai số chuẩn ước lượng

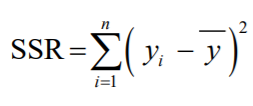


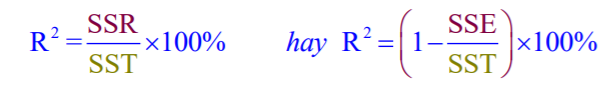
n = 21587; k = 5

+ **Multiple R-squared**: Hệ số tương quan tuyến tính thể hiện mức độ phù hợp của mô hình









Multiple R-sq = 0.499 Adjusted Multiple R-sq = 0.4989

+ **F-Statistic**: Tiêu chuẩn kiểm định

t-stat = 4301

Giả thiết:

H0: R2= 0

H1: R2 0

Miền bác bỏ:

Tiêu chuẩn thuộc miền bác bỏ, bác bỏ H0 thừa nhận H1, biến **price** có mối quan hệ tuyến tính với tập giá trị input.

b. Dựa vào kết quả P-value (Pr > |t|) của 5 feature đều rất nhỏ, vậy rõ ràng khả năng bác bỏ H0 của 5 biến trên đều rất cao (2.2 \* << 0.05), vậy không bác bỏ bất kỳ feature nào ở mức tin cậy 0.05.

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

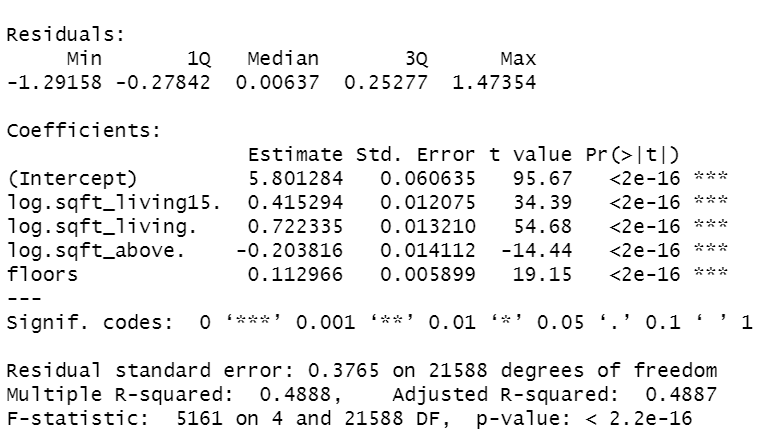
• Mô hình M1 chứa tất cả các biến còn lại là biến độc lập

• Mô hình M2 là loại bỏ biến **condition** từ mô hình M1.

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

Mô hình M1 đã được thực hiện ở câu a.

Mô hình M2: loại bỏ **‘condition’**



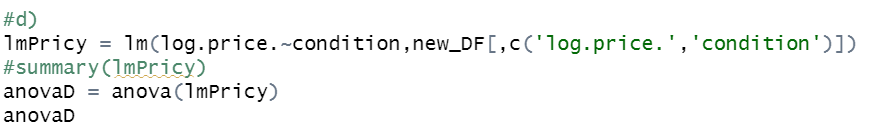
Giả thiết:

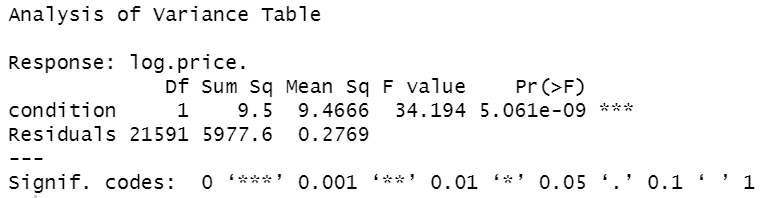
H0: R2= 0

H1: R2 0

Sau khi loại bỏ **‘condition’**, thông số R-sq giảm xuống, tuy vẫn bác bỏ mạnh H0 thừa nhận tương quan tuyến tính của mô hình, mức bác bỏ vẫn rất cao (P-value = 2.2 \* << 0.05) nhưng không tốt bằng mô hình ban đầu.

Sử dụng lệnh **anova()** khi xét mô hình **price** theo **condition:**





Giả thiết:

H0: Price không phụ thuộc vài sự biến thiên của condition

H1: Price phụ thuộc vài sự biến thiên của condition

P-value = 5.061 \* 10-9 << . Vậy bác bỏ H0 tốt, thừa nhận sự phụ thuộc vào sự biến thiên của condition, vậy rõ rang mô hình M1 là mô hình chuẩn xác hơn.

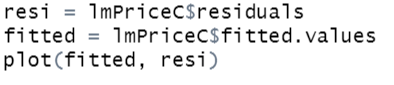
1. Mô hình hợp lý hơn ở câu c là mô hình M1.

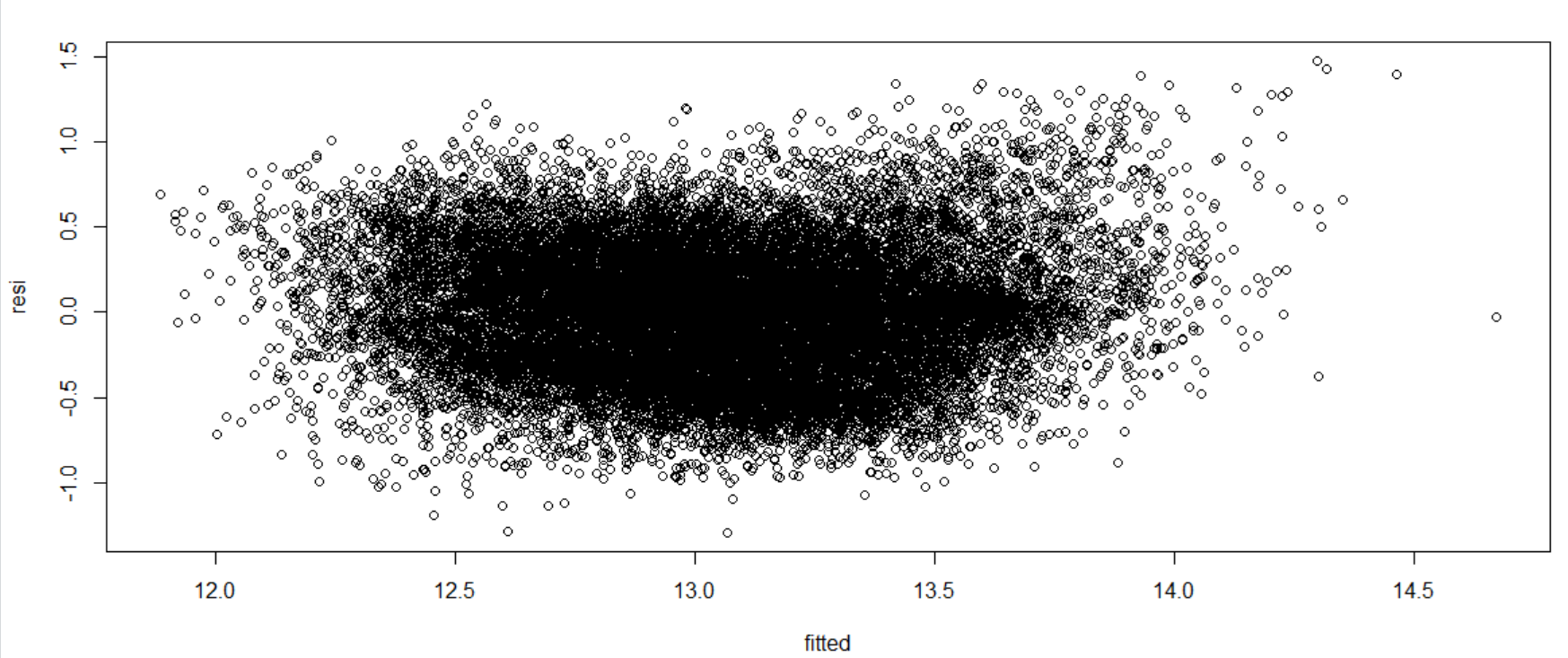
Các biến của M1 tác động vào giá trị **price** theo quan hệ tuyến tính xác định bằng hệ số **estimated coefficient**.

3 giá trị **sqft\_living**, **sqft\_living15**, **floors** tác động đáng kể vào giá nhà theo tỷ lệ thuận, **conditions** cũng tác động theo xu hướng tăng dần nhưng mức tác động thấp hơn.

Giá trị **sqft\_above** có tác động theo chiều tỉ lệ nghịch với giá nhà do hệ số âm

e) Đồ thị biểu diễn sai số hồi quy so với giá trị dự báo:



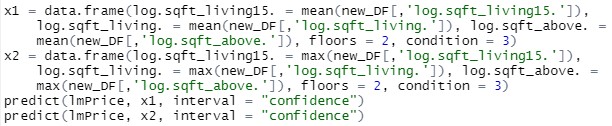


Giá nhà dao động phần lớn xung quanh giá trị 1012 – 1014, log (sai số dự báo dao động) từ -1 đến 1 tức dự báo chênh lệch lên đến 10 lần, sự chênh lệch do dự báo độc lập với giá trị dự báo được bởi mô hình

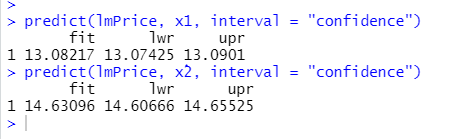
1. **Dự báo (Predictions):**

Để tiến hành dự đoán giá nhà tại 2 thuộc tính x1 và x2, trước tiên ta cần phải tạo 2 data frame x1 chứa các biến sqft\_living15 = mean(sqft\_living15), sqft\_living = mean(sqft\_living), sqft\_above = mean(sqft\_above), floors = 2, condition = 3 và x2 chứa các biến sqft\_living15 = max(sqft\_living15), sqft\_living = max(sqft\_living), sqft\_above = max(sqft\_above), floors = 2, condition = 3. Sau đó sử dụng lệnh predict với mô hình hồi quy tuyến tính thu được ở câu 4c và độ tin cậy 95%.

Code:



Ta thu được kết quả như sau:



Khoảng tin cậy cho giá trị **price** dự đoán được ở x2 lớn hơn x1.**II. PHẦN RIÊNG:**

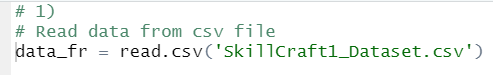
Tập tin “**SkillCraft1\_Dataset.csv**” chứa thông tin nghiên cứu Phép đo từ xa trò chơi điện tử như một công cụ quan trọng trong nghiên cứu học kỹ năng phức tạp của Thompson, Blair, Chen, & Henrey (2013). Các thuộc tính dữ liệu bao gồm số ID, tuổi người chơi, số giờ chơi mỗi tuần, tổng số giờ chơi được báo cáo và các số lượng đơn vị được thực hiện bằng phím trên mỗi dấu thời gian. Dữ liệu được đo từ xa của nhiều người chơi ở một số cấp độ kỹ năng. Dữ liệu gốc được cung cấp tại: <https://archive.ics.uci.edu/ml/datasets/SkillCraft1%2BMaster%2BTable%2BDataset>

Các biến chính trong bộ dữ liệu :

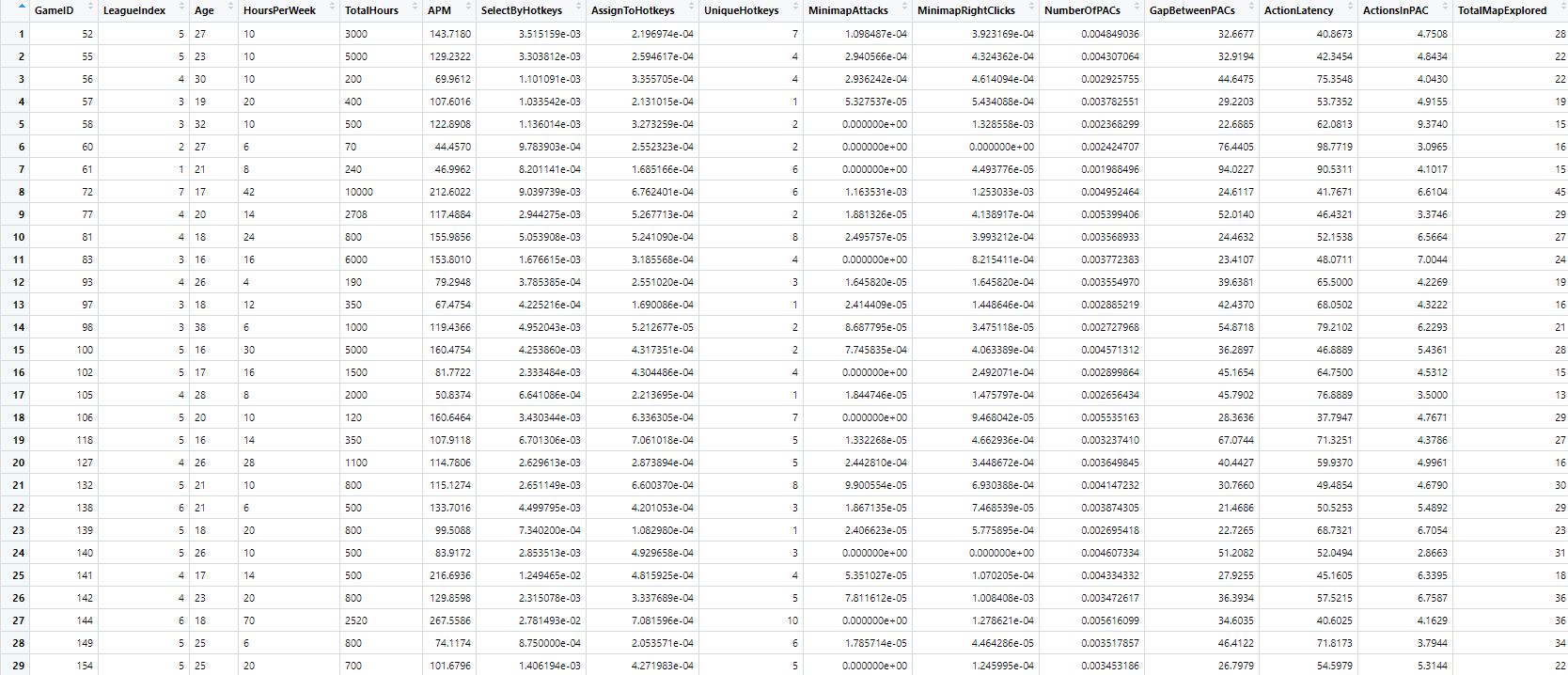
* **Age**: Tuổi của người chơi.
* **HoursPerWeek**: Số giờ chơi được báo cáo mỗi tuần.
* **TotalHours**: Tổng số giờ chơi được báo cáo.
* **APM**: Hành động mỗi phút .
* **SelectByHotkeys**: Số lượng đơn vị hoặc lựa chọn tòa nhà được thực hiện bằng phím nóng trên mỗi dấu thời gian.
* **UniqueHotkeys**: Số lượng phím nóng duy nhất được sử dụng trên mỗi dấu thời gian.
* **MinimapAttacks**: Số hành động tấn công trên bản đồ nhỏ trên mỗi dấu thời gian .
* **MinimapRightClicks**: Số lần nhấp chuột phải vào bản đồ nhỏ trên mỗi dấu thời gian .

1. **Đọc dữ liệu: (Import Data)**

Ta dùng lệnh **read.csv()** như sau:

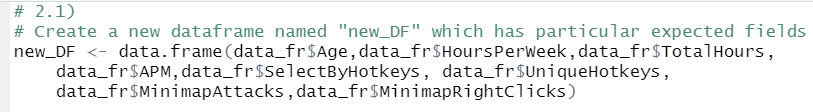


Kết quả ta thu được một data frame với rất nhiều fields như sau :

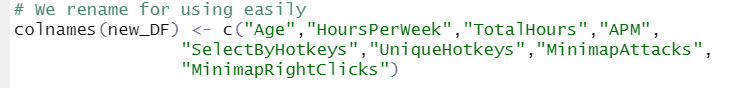


1. **Làm sạch dữ liệu (Data cleaning):**

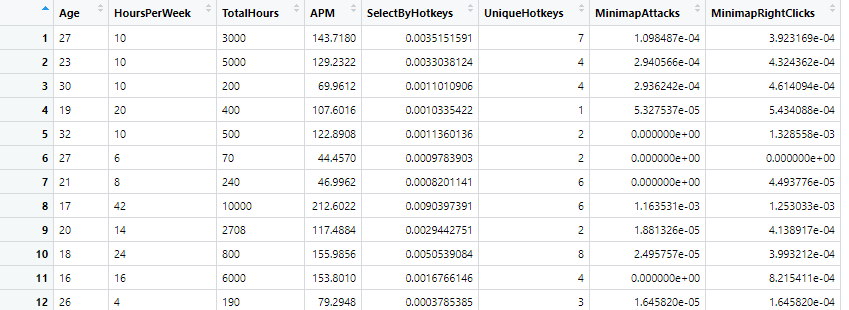
Ta sẽ thực hiện trích ra một dữ liệu con đặt tên là **new\_DF** chỉ bao gồm các biến mà ta quan tâm .



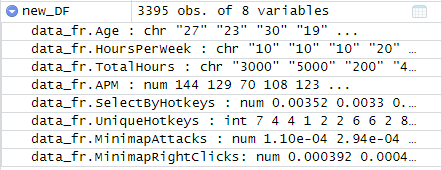
Sau đó ta chạy thêm lệnh sau để đặt lại tên các fields của **new\_DF**:



Kết quả ta thu được **new\_DF** như sau:



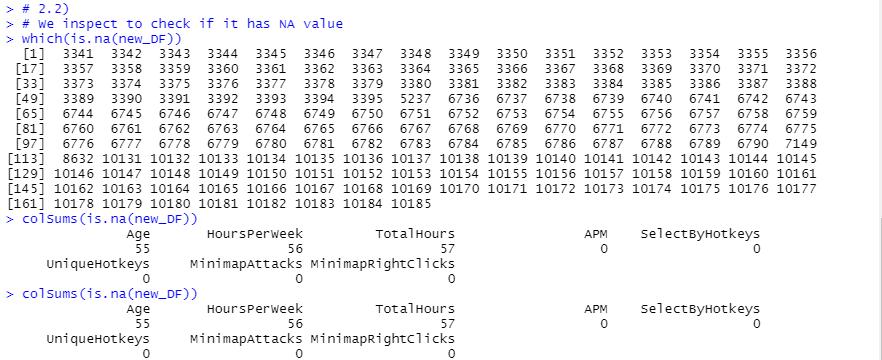
Ta thấy có 3 cột có kiểu char



Ta chạy lệnh ép kiểu về num (các kí tự “?” sẽ được chuyển về N.A)

C:\Users\MyPC\Downloads\Untitled.png

Ta chạy lệnh sau để xem thử có những điểm dữ liệu khuyết nào:



Như vậy, ta thấy một số lượng nhỏ dữ liệu bị khuyết.

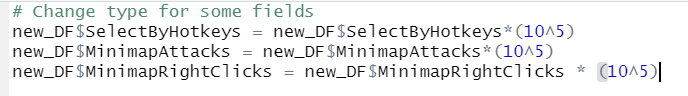
Ở đây ta có một số cách xử lí dữ liệu bị khuyết như là thay thế bằng một giá trị trung bình. Tuy nhiên, số lượng dữ liệu bị khuyết rất nhỏ so với tổng dữ liệu nên nhóm quyết định xóa bỏ những hàng chứa các dữ liệu khuyết này.

Ta dùng lệnh sau:

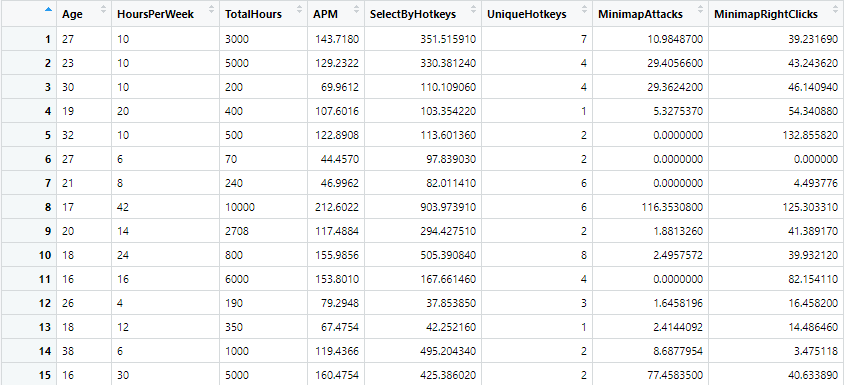
C:\Users\MyPC\Downloads\Untitled.png

1. **Làm rõ dữ liệu (Data visualization) :**

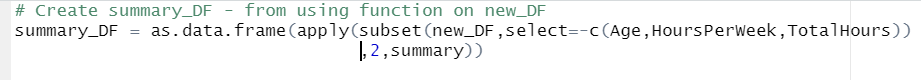
Chuyển đổi các biến **SelectByHotkeys**, **MinimapAttacks**, **MinimapRightClicks** lần lượt thành **SelectByHotkeys \* 10^5**, **MinimapAttacks\* 10^5** , **MinimapRightClicks\*10^5**. Từ đây mọi sự tính toán với các biến trên được hiểu là đã qua đổi biến dạng mũ.



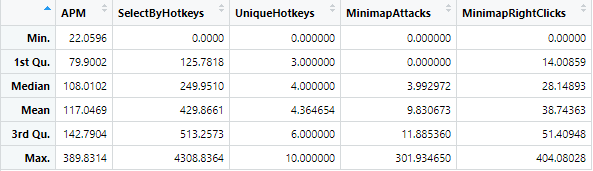
Dữ liệu lúc này trở thành:



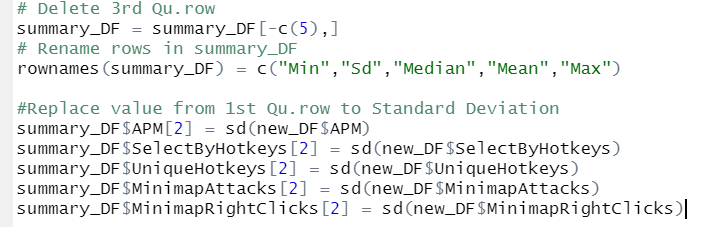
Ta chạy lệnh **summary()** cho **new\_DF**, kết quả thu về lưu vào **summary\_DF**:



Giá trị của **summary\_DF** lúc này cho ta:

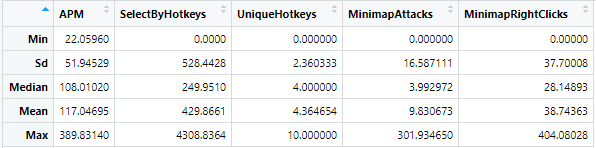


Thông tin từ lệnh **summary()** đã cho ta đủ thông tin về min, max, median, mean, 1st & 3rd quartile. Ở đây ta cần tính thêm độ lệch chuẩn và không cần sử dụng thông tin 1st & 3rd quartile nên ta thay giá trị 1st quartile bằng thông tin độ lệnh chuẩn và xóa đi hàng 3rd quartile. Ta dùng lệnh như sau:

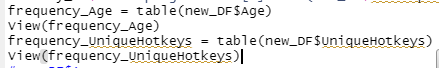


Ta chỉ summary các biến liên tục

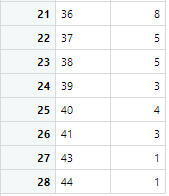
Kết quả cuối cùng ta thu được **summary\_DF** chứa các thông tin cần như sau:



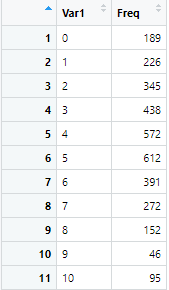
Bây giờ ta sẽ tạo bảng cho các phân loại: ta dùng lệnh



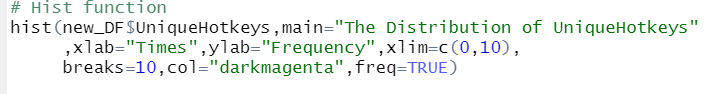
* Phân loại theo độ tuổi:



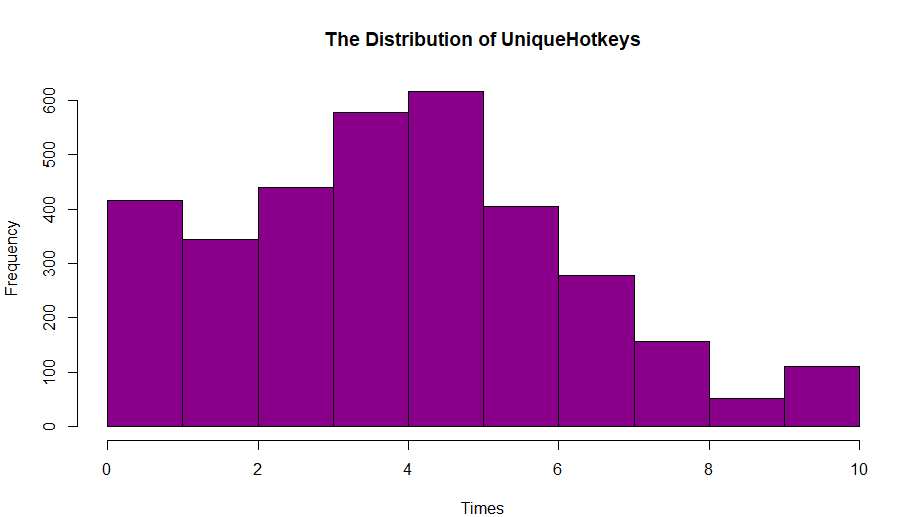
* Phân loại theo UniqueHotkeys:



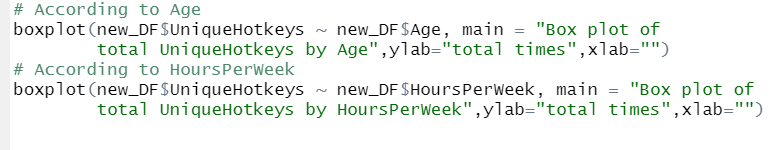
Ta dùng lệnh hist() để vẽ đồ thị phân phối của biến như sau:



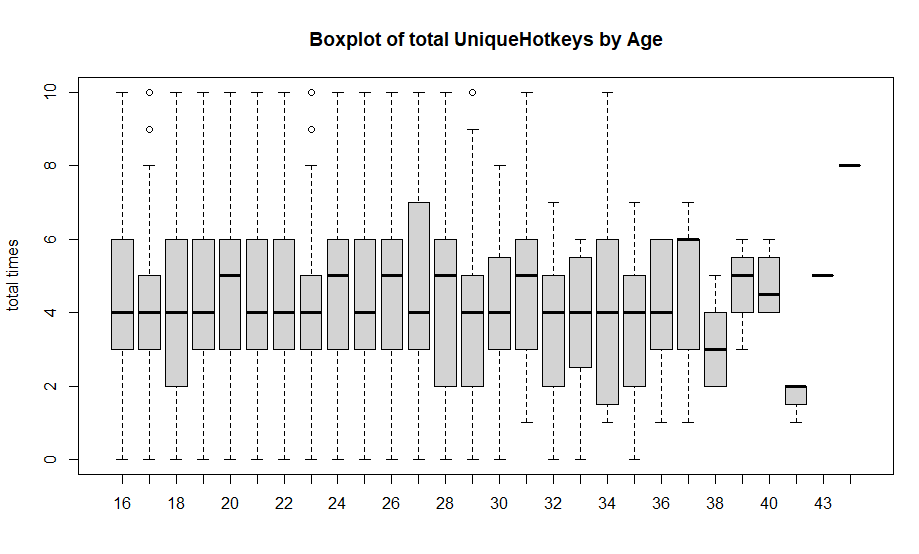
Kết quả ta thu được đồ thị như sau:

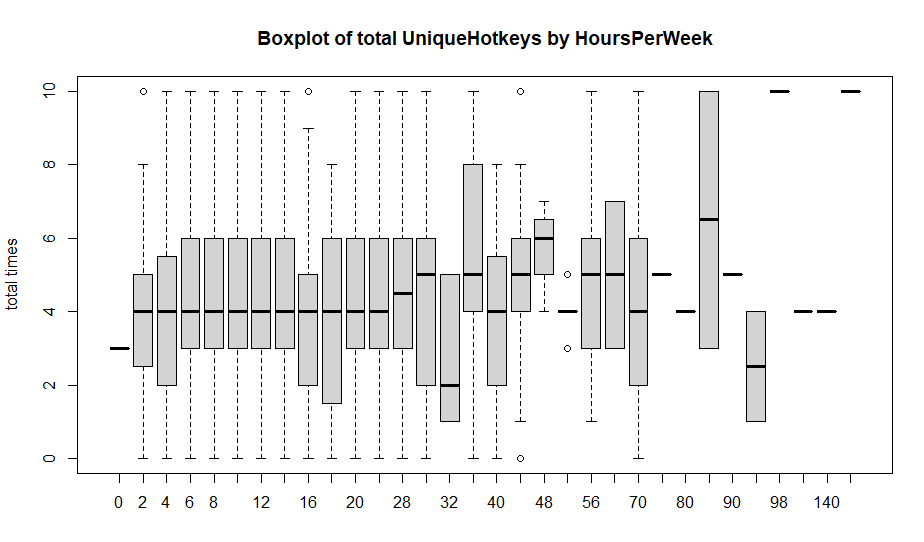


Dùng hàm **boxplot()** ta vẽ phân phối biến **UniqueHotkeys** cho từng nhóm phân loại của biến **Age**, **HoursPerWeek** như sau:

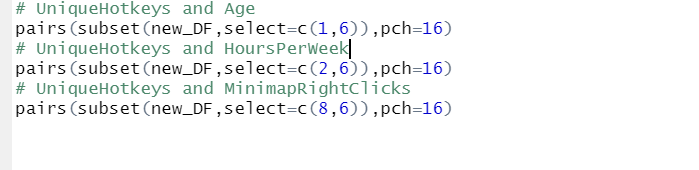


Kết quả lần lượt ta thu được 2 đồ thị sau:

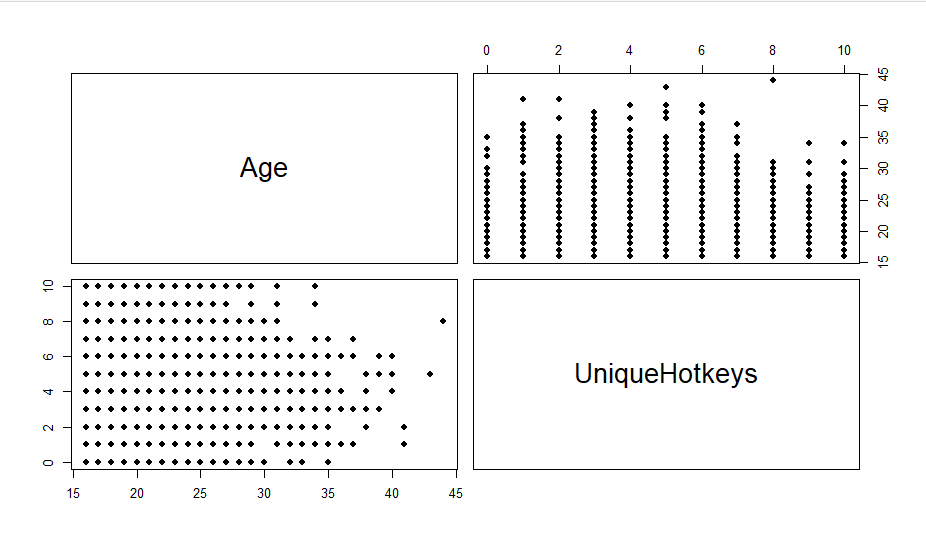


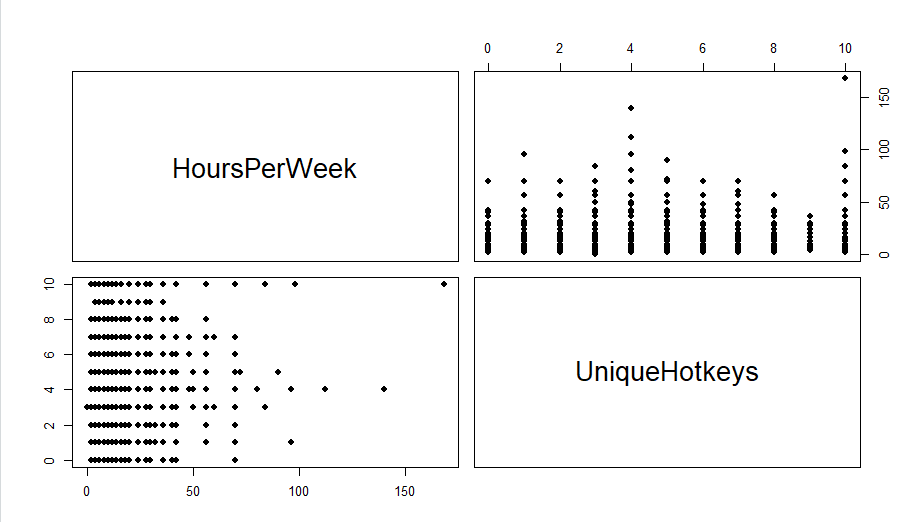


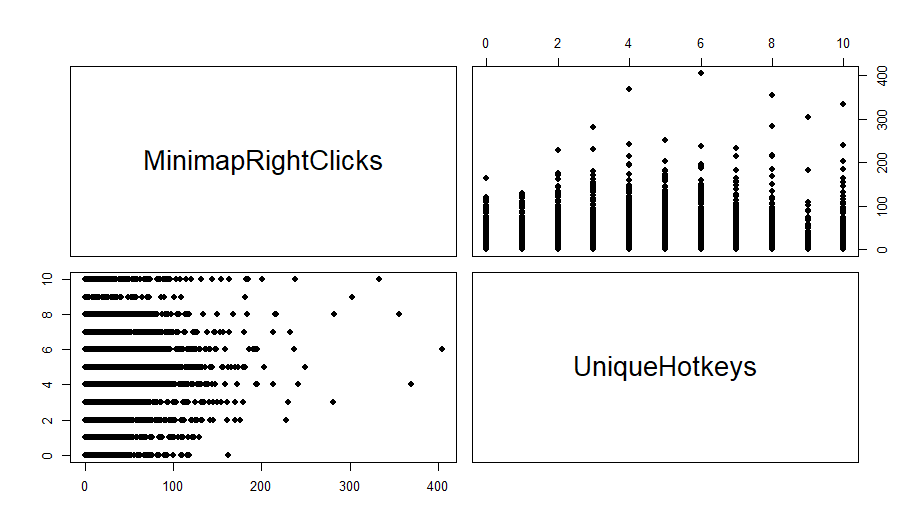
Dùng lệnh **pairs()** ta vẽ các phân phối của biến **UniqueHotkeys** lần lượt theo các biến **Age**, **HoursPerWeek**, **MinimapRightClicks** như sau:



Kết quả thu được 3 đồ thị dưới :







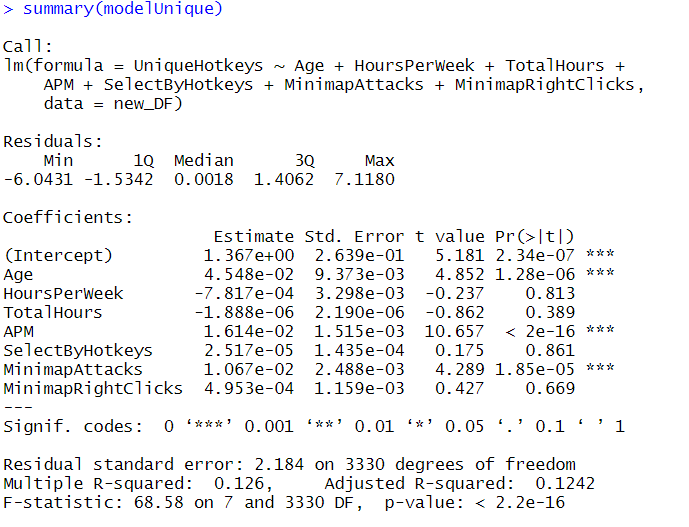
**4. Xây dựng các mô hình hồi quy tuyến tính (Fitting linear regression models):**

Chúng ta muốn khám phá rằng có những nhân tố nào và tác động như thế nào đến số lượng phím nóng duy nhất được sử dụng trên mỗi dấu thời gian (UniqueHotkeys).

1. Xét mô hình hồi quy tuyến tính với **UniqueHotkeys** là biến phụ thuộc, các biến còn lại là biến độc lập. Ta dùng lệnh **lm()** như sau:

https://lh5.googleusercontent.com/-FOQmRti_aVHPAHzYMoscTMEI6IWKHt-Gs_b70qyPAveTFkeuhj-EAoxMtIy0RishqAfhNSaFgxe5uQFGJDP751DO4cBvgss6M6p4AQB5088Ovb-JAUTV0KDk7lGYKYWZzosOKcF

Chạy lệnh **summary()** để thu được các thông số từ **modelUnique** như sau:



1. Quan sát giá trị P-value, chính là giá trị Pr (>| t |) trong phần Coefficients.

Với mức tin cậy 5%: Ta sẽ loại bỏ biến HoursPerWeek, TotalHours,    SelectByHotkeys, MinimapRightClicks vì P-value > 0.05.

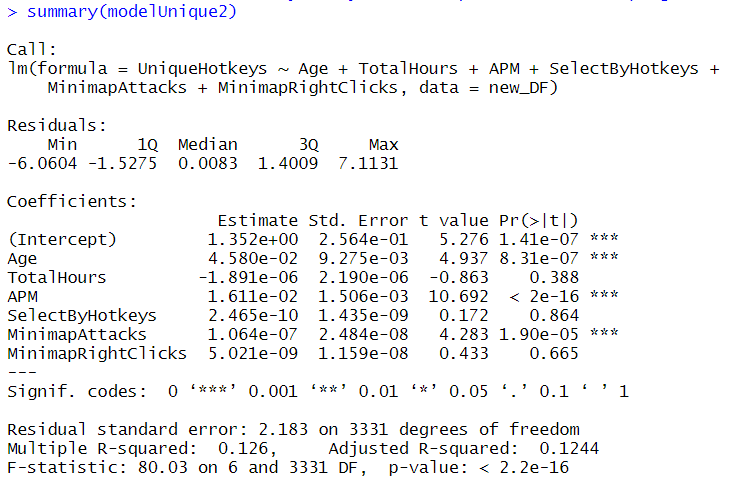
1. Xét mô hình ban đầu (M1) và mô hình tuyến tính M2 là loại bỏ biến **HoursPerWeek**

từ M1. Dùng lệnh **anova()** để đề xuất mô hình hồi quy tuyến tính hợp lý hơn.

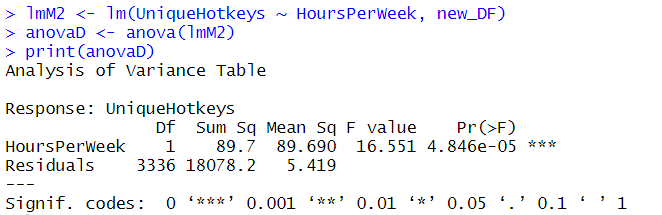
Mô hình M1 đã được thực hiện ở trên.

Mô hình M2:

https://lh5.googleusercontent.com/Ei6CRC9AiNh8AWSoB0Bx4IkTgE3NtomQvgzJSA0FshpVnEAT2zbVdBvBs0T3OFULQbnbohs-KjGculhMXlGNEiIyQoGYVuVpJipWFQC8f57WO23jnuhtgTjauyvM5orWCSnZZBth



Sử dụng lệnh **anova()** khi xét mô hình của biến **UniqueHotkeys** theo biến **HoursPerWeek**:



Giả thiết:

            H0:  UniqueHotkeys không phụ thuộc vào sự biến thiên của HoursPerWeek

H1: UniqueHotkeys phụ thuộc vào sự biến thiên của HoursPerWeek

 P-value  = 4.846 \* 10-9 <<  = 0.05. Vậy bác bỏ H0, thừa nhận sự phụ thuộc vào sự biến thiên của HoursPerWeek, vậy rõ ràng mô hình M1 là mô hình chuẩn xác hơn.

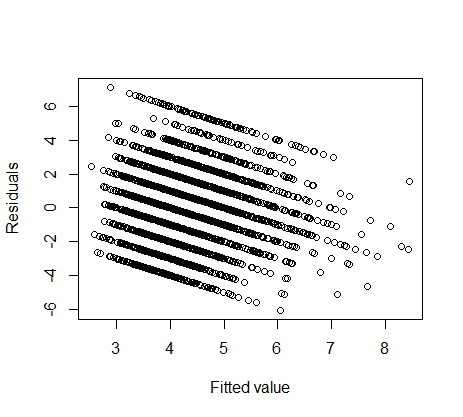
1. Chọn mô hình hợp lí hơn từ câu (c), hãy suy luận sự tác động của các biến lên  **UniqueHotkeys**.

Các biến của M1 tác động vào giá trị **UniqueHotkeys** theo quan hệ tuyến tính xác định bằng hệ số **estimated coefficient**.

Các giá trị **Age**, **APM**, **MinimapAttacks** tác động vào giá trị **UniqueHotkeys** theo tỷ lệ thuận. Trong đó giá trị **Age** có tác động lớn nhất và giá trị **MinimapAttacks** có tác động nhỏ nhất.

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

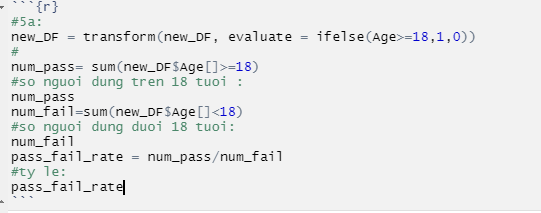
https://lh4.googleusercontent.com/dxXFY1uEno_BmvXP4mjwXiChXvAvNTd6Vmb47BwpQV-9wdnaTPfb2lk5iCQPHU8POKEzHEiBc60T7gBQGev_09DCAQixOt9CwF6tZWwJOKVzf0WLAlZ39y57dDR8tCIs6O4ooeUa

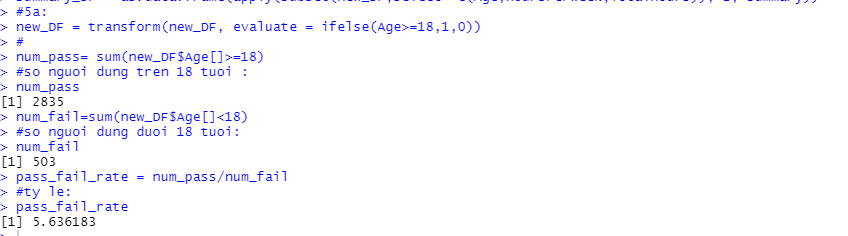


**5. Dự báo (Prediction):**

a. Giả sử đây là game dành cho người lớn trên 18 tuổi, chúng ta cần thống kê số người dùng có độ tuổi dưới 18 tuổi đã tham gia để đưa ra các biện pháp xử lý. Thống kê tỷ lệ đạt (Age >= 18) hoặc không đạt ( Age < 18) của người dùng. Tính tỷ lệ đạt/không đạt.

Trong **new\_DF** ta tạo thêm cột mới **“evaluate”** và thêm các giá trị 0 hoặc 1 vào cột này tùy vào giá trị tuổi **Age** ( lớn hơn hoặc bằng 18 tuổi tương ứng với giá trị 1, nhỏ hơn 18 tuổi tương ứng với giá trị 0) .





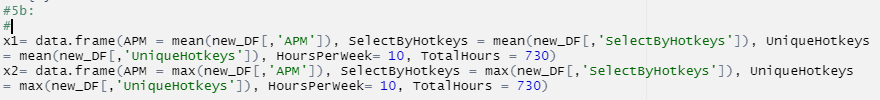
b. Xét mô hình tuyến tính cùng bao gồm biến **Age** là biến phụ thuộc, còn lại là biến độc lập. Sử dụng lệnh **predict()** để dự báo tuổi của người dùng tại 2 thuộc tính như sau:

x1: APM = mean(APM), SelectByHotkeys = mean(SelectByHotkeys), UniqueHotkeys = mean(UniqueHotkeys), HoursPerWeek= 10, TotalHours = 730.

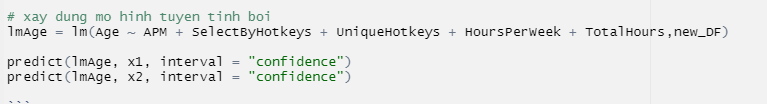
x2: APM = max(APM), SelectByHotkeys = max(SelectByHotkeys), UniqueHotkeys= max(UniqueHotkeys), HoursPerWeek= 10, TotalHours = 730.

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

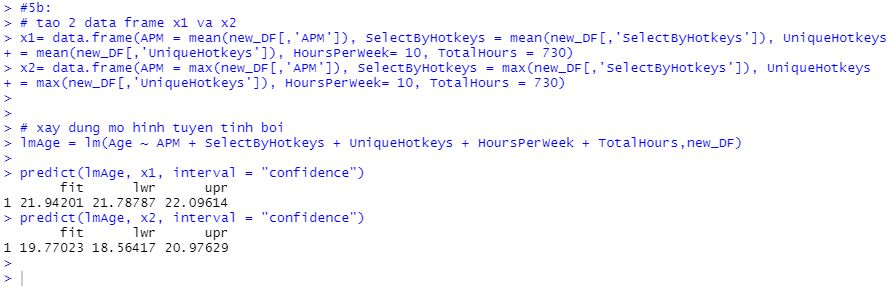
Tạo 2 data frame x1 và x2 :



Sử dụng lệnh **predict()** với mô hình hồi quy tuyến tính trên với độ tin cậy 95%.



Kết quả thu được:



Như vậy, khoảng tin cậy cho giá trị **Age** (tuổi) dự đoán ở x1 lớn hơn x2