VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF APPLIED SCIENCE

PROBABILITY AND STATISTICS (MT2013)

Assignment

Advisor: Phan Thị Hường
Students: Nguyễn Đình Quân - 2114547
Lê Trọng Đức - 2152523
Văn Tuấn Anh - 2152400
Hoàng Duy Tân - 2053420

HO CHI MINH CITY, NOVEMBER 2022

Contents

1 Introduction

1.1 Content

In this project, in **Activity 1**, we will use a house price dataset to build a model and predict the house price in the future. In **Activity 2**, we will use the StarCraft data set to learn about what factor mainly affect the rank of a player.
Through this project, we can obtain more knowledge about how to build a model, how to predict value of an dependent variable through many independent variables. Moreover, we can leverage our skill of using R-studio and can interpret many plots and graphs.

1.2 Team members & workload

No.	Fullname	Student ID	Problems	Percentage of work
1	Nguyễn Đình Quân	2114547	Part 1 Activity 2	26.67%
2	Lê Trọng Đức	2152523	Part 1 Theory & Activity 1	26.67%
3	Văn Tuấn Anh	2152400	Part 2 Activity 2	26.67%
4	Hoàng Duy Tân	2053420	Part 2 Theory	20%

TOPIC 1

2 Theoretical basis

2.1 Introduction to the Multiple Linear Regression Model

Many applications of regression analysis involve situations that have more than one regressor or predictor variable. A regression model that contains more than one regressor variable is called a **multiple regression model**. In general, the form of multiple linear regression model with k regressor variables is:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \epsilon$$

where

- Y : dependent variable
- x_i : expanatory variables
- β_0 : y-intercept (constant term)
- β_i : regression coefficients, represents the expected change in response Y per unit change in x_j when all the remaining regressors $x_i (i \neq j)$ stay fixed
- ϵ : the model's error term (the residuals)

By using the multiple linear regression, we can understand the relationship between the dependent variable Y and the rest independent variables x_i. It can provide us with important insight which help us to predict the future of Y corresponding to the change x_i.

2.2 Estimation of the parameters

2.2.1 Estimated Multiple Linear Regression equation

Because our data is just a part of lots of more data (the data we may missed or uncollected), we can't really see the bigger picture. We can't calculate an exactly model showing the exactly relationship among the dependent and independent variables.
Either a simple or multiple regression model is initially posed as a hypothesis concerning the relationship among the dependent and independent variables. Therefore, through the sample data, we estimate a model, and the regression function of that model is called **Estimated Multiple Linear Regression equation**.

The multiple linear regression is expressed as follows:

$$\hat{Y} = b_0 + b_1x_1 + b_2x_2 + \cdots + b_px_p$$

where

- $\hat{Y}$: the predicted or expected value of the dependent variable.
- x_i : distinct independent or predictor variables.
- b_0 : y-intercept, the value of $\hat{Y}$ when all of the independent variables x_i are equal to zero.
- b_i :the estimated regression coefficients

There is actually another unknown parameter in our regression model, σ^2 (the variance of the error term ϵ). The residuals $e_i = y_i - \hat{y}_i$ are used to obtain an estimate of σ^2.

2.2.2 Least Squares Estimation of the parameters

The method of least squares may be used to estimate the regression coeffcients in the multiple regression model. Suppose that $n > k$ observations are available, and let x_{ij} denote the ith observation or level of variable x_j. The observations are

$$(x_{i1}, x_{i2}, \ldots, x_{ik}, y_i) \; i\text{=1, 2,}\ldots, n \text{ and } n > k$$

It is customary to present the data for multiple regression in a table such as table below

y	x_1	x_2	$\ldots$	x_k
y_1	x_{11}	x_{12}	$\ldots$	x_{1k}
y_2	x_{21}	x_{22}	$\ldots$	x_{2k}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	x_{n1}	x_{n2}	$\ldots$	x_{nk}

Each observation $(x_{i1}, x_{i2}, \ldots, x_{ik}, y_i)$ satisfies the model equation

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \epsilon_i \\ &= \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \epsilon_i \qquad\qquad i\text{=1, 2,}\ldots, n \end{aligned}$$

The least square function is

$$L = \sum_{i=1}^{n} \epsilon^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij} \right)^2 \tag{1}$$

According to page 481 of Applied Statistics and Probability for engineers ,Douglas C.Montgomery and George C.Runger. We obtain

$$\begin{aligned} n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1} + \hat{\beta}_2 \sum_{i=1}^{n} x_{i2} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{ik} &= \sum_{i=1}^{n} y_i \\ \hat{\beta}_0 \sum_{i=1}^{n} x_{i1} + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1}^2 + \hat{\beta}_2 \sum_{i=1}^{n} x_{i2}x_{i1} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{ik}x_{i1} &= \sum_{i=1}^{n} y_i x_{i1} \\ &\vdots \\ \hat{\beta}_0 \sum_{i=1}^{n} x_{ik} + \hat{\beta}_1 \sum_{i=1}^{n} x_{i1}x_{ik} + \hat{\beta}_2 \sum_{i=1}^{n} x_{i2}x_{ik} + \cdots + \hat{\beta}_k \sum_{i=1}^{n} x_{ik}^2 &= \sum_{i=1}^{n} y_i x_{ik} \end{aligned}$$

2.2.3 R^2 How fit is the model

The adjusted R^2 statistic essentially penalizes the analyst for adding terms to the model. It is an easy way to guard against overfitting, that is, including regressors that are not really useful. Consequently, it is very useful in comparing and evaluating competing regression models.

2.2.4 Hypothesis Tests In Multiple Linear Regression

In multiple linear regression problems, certain tests of hypotheses about the model parameters are useful in measuring model adequacy. In this section, we describe several important hypothesis-testing procedures. As in the simple linear regression case, hypothesis testing requires that the error terms ε_i in the regression model are normally and independently distributed with mean zero and variance σ^2

The test for signifcance of regression is a test to determine whether a linear relationship exists between the response variable y and a subset of the regressor variables $x_1, x_2, x_3, ..., x_k$. The appropriate hypotheses are:

$$\begin{aligned} &H_0 : \beta_0 = \beta_1 = \beta_2 = \beta_3 = ... = \beta_k = 0 \\ &H_1 : \exists j : \beta_j \neq 0 \end{aligned}$$

Rejection of H_0 implies that at least one of the regressor variables $x_1, x_2, ..., x_k$ contributes signifcantly to the model.

The test for signifcance of regression is a generalization of the procedure used in simple linear regression. The total sum of squares SST is partitioned into a sum of squares due to the model or to regression and a sum of squares due to error, say:

$$SS_T = SS_R + SS_E$$

Now if $H_0 : \beta_0 = \beta_1 = \cdots = \beta_k = 0$ is true, SSR/ε^2 is a chi-square random variable with k-degrees of freedom. Note that the number of degrees of freedom for this chi-square random variable is equal to the number of regressor variables in the model. We can also show that the SSR/ε^2 is a chi-square random variable with $n - p$ degrees of freedom, and that SSE and SSR are independent. The test statistic for $H_0 : \beta_0 = \beta_1 = \cdots = \beta_k = 0$ is:

$$F_0 = \frac{SS_R/k}{SS_E/(n-p)} = \frac{MS_R}{MS_E}$$

We should reject H_0 if the computed value of the test statistic F_0 , is greater than $f_{a,k,n-p}$.

2.2.5 Multicollinearity

Multicollinearity happens when independent variables in the regression model are highly correlated to each other. It makes it hard to interpret of model and also creates an overfitting problem. It is a common assumption that people test before selecting the variables into the regression model.
When independent variables are highly correlated, change in one variable would cause change to another and so the model results fluctuate significantly. The model results will be unstable and vary a lot given a small change in the data or model.
Depending on the situation, it may not be a problem for our model if only slight or moderate collinearity issue occurs. However, it is strongly advised to solve the issue if severe collinearity issue exists(e.g. correlation >0.8 between 2 variables or Variance inflation factor(VIF) >20). And in our model, we detect the multicollinearity through calculating the VIF and correlation.

2.2.6 Overfitting

Adding more independent variables to a multiple regression procedure does not mean the regression will be "better" or offer better prediction; in fact it can make thing worse. This is call **Overfitting**.
The ideal is for all of the independent variables to be correlated with the dependent variable but **NOT** with each other.

2.2.7 Variance inflation factor (VIF)

The variance inflation factor (VIF) will helps us identify correlation between independent variables and the strength of that correlation.

- VIF = 1 : No correlation
- VIF from 1 to 5 : Moderate correlation
- VIF > 10 : High correlation

3 How to interpret graphs

In this assignment, we will draw 3 different type of graph: histogram, box plots. Here is a short introduction of these graphs

3.1 Histogram

Histograms show us how frequently every value of the data set appears in a relatively unbiased way.
In a histogram, the type of data being measured is represented on the horizontal axis, and the vertical axis represents how many observations are in each range. These range are divided equally into different intervals. In a histogram, the type of data being measured is represented on the horizontal axis, and the vertical axis represents how many observations are in each bin.panning from the smallest value to the biggest value of the data set. The length of the column show us the number of values in the data set that fall within a particular range. The higher the column,

the more frequent the data is in the data set.

Histograms allow us to see how skewed the data set is and make it easy to compare different data sets.

3.2 Box Plot

Box plot (or a wisker plot) visually show the distribution of numerical data and skewness through displaying the data quartiles (or percentiles) and averages.
A box plot consist of these elements.

- Wisker tail, showing the lower and and upper 25%.
- Box, showing the interqualtile range.
- Median line, showing the median of the data.
- Circles, represent outliars

Box plots are used as visual summary of the data, enabling researchers to quickly identify mean values, the dispersion of the data set, and signs of skewness.

3.3 Pairs

A pair plot displays all scatter plots together in a matrix of panels. The pair plot helps us to visualize the distribution of single variables as well as relationships between two variables. They are a great method to identify trends between variables.

3.4 Residuals vs Fitted plot

Residuals vs Fitted plots the predicted values with the corresponding residual (error) values, used to check the linearity of the data (assumption 1) and the homogeneity of the variances (assumption 3). If the assumption of linearity of the data is NOT satisfied, we will observe that the residuals on the graph will be distributed according to a certain characteristic pattern. (e.g. parabola). If the red line on the scatter plot is a horizontal line other than a curve, then the linearity assumption of the data is satisfied. To test the third assumption (uniform variance) then the residuals must be evenly dispersed around the line $y = 0$.

3.5 Normal Quantile-Quantile plot

The Normal Q-Q graph allows the assumption of the normal distribution of errors to be tested. If the residual points lie on the same line, then the condition for the normal distribution is satisfied. If mostly all the points lie on the 45-degree line, the errors follows the normal distribution.

3.6 Scale - Location plot

The Scale - Location plots the square root of the residuals normalized with the predicted values, which is used to test the third assumption (variance of errors is constant). If the red line on the graph is a horizontal line and the residuals are evenly distributed around this line, then the third assumption is satisfied. If the red line has a slope (or curves) or the residuals are unevenly scattered around this line, then the third assumption is violated.

3.7 Residuals vs Leverage plot

Residuals vs Leverage graph allows to identify points with high influence (influential observations), if they are present in the data set. These high-impact points can be outliers, which are the ones that can have the most impact when analyzing data. If we observe a dashed red line (Cook's distance), and there are some points that cross this distance line, that means those points are high influence points. If we only observe the Cook distance line at the corner of the graph and no point crosses it, then none of the points are really highly influential.

4 Activity 1

4.1 Requirements

The dataset contains house sale prices for King County, which includes Seattle. It includes home sold between May 2014 and May 2015.
Attribute Information:

- *price* - Price of each home sold
- *sqft_living* - Square footage of the apartments interior living space
- *floors* - Number of floors
- *condition* - An index from 1 to 5 on the condition of the apartment
- *sqft_above* - The square footage of the interior housing space that is above ground level
- *sqft_living15* - The square footage of interior housing living space for the nearest 15 neighbors

Step:

1. Import data: **`house_price.csv`**
2. Data cleaning: NA (Not available)
3. Data visualization
   (a) Transformation (if it is necessary)
   (b) Descriptive statistics for each of the variables
   (c) Graphs: hist, boxplot, pairs.
4. Fitting linear regression models: We want to explore what factors may affect home prices in King County.
5. Predictions.

4.2 Implement

4.2.1 Step1. Import the data

To import the data, we use the **"data.table"** package to import the data

```
library(data.table)
library(GGally)
library(car)
data <- fread("D:\\R code\\house_price.csv") #read file
```

Listing 1: Imporing the data

Then we can check if the data already imported by using the `View()` command

```
View(data)
```

4.2.2 Step2. Data cleaning

Before cleaning the data, we need to filtering all the variables that we consider. We save the filtered data `newData`

```
newData <- data[, c("price","sqft_living","floors","condition","sqft_above","sqft_living15")]
```

For this report, in the data cleaning step, all we need to do is to deal with missing values, which is all the values represented by `NA`. First, we need to identify the missing data use `is.na()` which returns a logical vector with `TRUE` in the element location that contain missing values. To identify the location or the number of missing values we can use `which()` and `sum()` functions together with function `apply()`

```
apply(is.na(newData),2, which) # to know which position the value is missing
apply(is.na(newData),2, sum) # to know how many cells not having the values
```

By observing the result in `newData` we conclude that the `price` variable has 20 missing values cells. Now we need to recode those cells.
The method we suggest here is recoding the missing values with the mean values. We have to calculate the mean without those missing values by setting the remove NA function `na.rm` of the mean function to `TRUE`. Then substitute the mean into all the cells with no values.

```
newData$price[is.na(newData$price)] = mean(newData$price, na.rm=TRUE)
apply(is.na(newData),2, which) #check again to make sure no NA values left
```

4.2.3 Step3. Data visualization

4.2.3.a Data transformation

We will create a new data call `newData_2` (which include all the variables in `newData`) and transform all the variables `price`, `sqft_living`, `sqft_above`, `sqft_living15` into `log(price+1)`, `log(sqft_living+1)`, `log(sqft_above+1)` and `log(sqft_living15+1)`.

```
newData_2 <- newData
newData_2[, c("price","sqft_living","sqft_above","sqft_living15")]  <- log(newData_2[, c("
    price","sqft_living","sqft_above","sqft_living15")] + 1)
head(newData_2) #view 6 first lines the new data
```

The reasons why we need to do the log transformation:

- Improve the accuracy of the models: when we establish the linear regression model, the residuals obtained should be normally distributed. Hence, if the residuals are not normally distributed yet, performing log transformation on a variable can make that variable normally distributed.
- Performing log transformation will help us explain better on the relationship between two variables.
- Performing log transformation still allow us to estimate our models by linear regression.

- Moreover, in this report, we prefer log(x+1) to log(x) due to the fact that there are some zero value cells in our data. If we take log of those cells the return values will be negative infinity.

4.2.3.b Descriptive statistics for each of the variables

First,we will make the descriptive statistic for variables `price`, `sqft_living`, `sqft_above`, `sqft_living15`.

```
#mean
mean_1 <- apply(newData[, c("price","sqft_living","sqft_above","sqft_living15")], 2, mean)
#standard variance
sd_1 <- apply(newData[, c("price","sqft_living","sqft_above","sqft_living15")], 2, sd)
#min
min_1 <- apply(newData[, c("price","sqft_living","sqft_above","sqft_living15")], 2, min)
#max
max_1 <- apply(newData[, c("price","sqft_living","sqft_above","sqft_living15")], 2, max)
#median
median_1 <- apply(newData[, c("price","sqft_living","sqft_above","sqft_living15")], 2, median
    )
#plug in the table
data.frame(mean_1, sd_1, min_1, max_1, median_1)
```

```
                 mean_1        sd_1 min_1   max_1 median_1
price         540067.607 366904.3447 75000 7700000   450000
sqft_living     2079.900    918.4409   290   13540     1910
sqft_above      1788.391    828.0910   290    9410     1560
sqft_living15   1986.552    685.3913   399    6210     1840
```

Listing 2: R output

We will make the descriptive statistic for variables `log(price+1)`, `log(sqft_living+1)`, `log(sqft_above+1)` and `log(sqft_living15+1)`.

```
#mean
mean_2 <- apply(newData_2[, c("price","sqft_living","sqft_above","sqft_living15")], 2, mean)
#standard variance
sd_2 <- apply(newData_2[, c("price","sqft_living","sqft_above","sqft_living15")], 2, sd)
#min
min_2 <- apply(newData_2[, c("price","sqft_living","sqft_above","sqft_living15")], 2, min)
#max
max_2 <- apply(newData_2[, c("price","sqft_living","sqft_above","sqft_living15")], 2, max)
#median
median_2 <- apply(newData_2[, c("price","sqft_living","sqft_above","sqft_living15")], 2,
    median)
#plug in the table
data.frame(mean_2, sd_2, min_2, max_2, median_2)
```

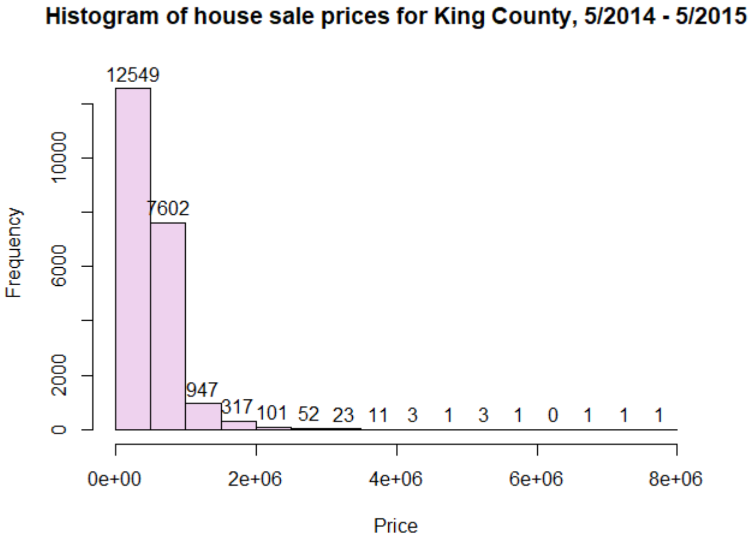
```
                mean_2      sd_2     min_2     max_2  median_2
price        13.047983 0.5263493 11.225257 15.856731 13.017005
sqft_living   7.550910 0.4245612  5.673323  9.513477  7.555382
sqft_above    7.395548 0.4273613  5.673323  9.149634  7.353082
sqft_living15  7.540000 0.3273378  5.991465  8.734077  7.518064
```

Listing 3: R output

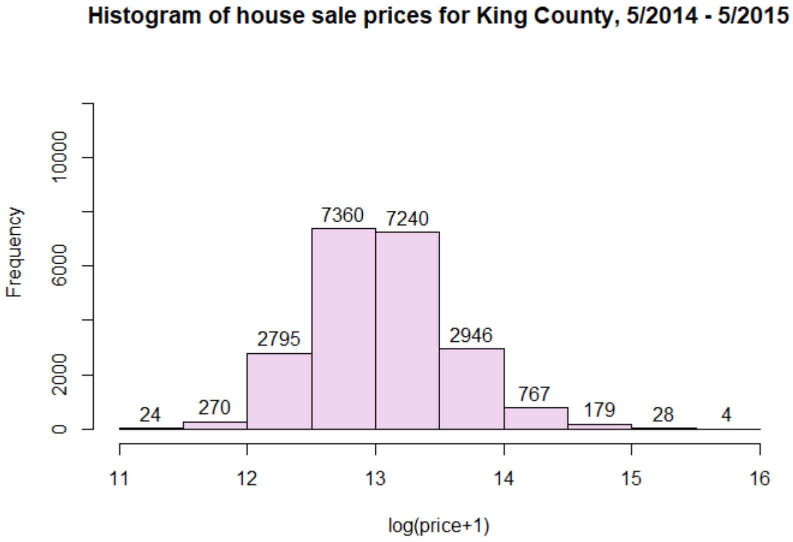
Histogram graph

```
hist(newData$price,
 main = paste("Histogram of house sale prices",
             "for King County, 5/2014 - 5/2015"),
 xlab = "Price",
 col    = "thistle2",  # Color for histogram
 labels = TRUE,
 ylim=c(0,13000)
 )

hist(newData_2$price,
     main = paste("Histogram of house sale prices",
                  "for King County , 5/2014 - 5/2015"),
     xlab = "log(price+1)",
     col    = "thistle2",  # Color for histogram
     labels = TRUE,
     ylim=c(0 ,8000)
     )
```



Hình 1: Histogram of `price`



Hình 2: Histogram of `log(price+1)`

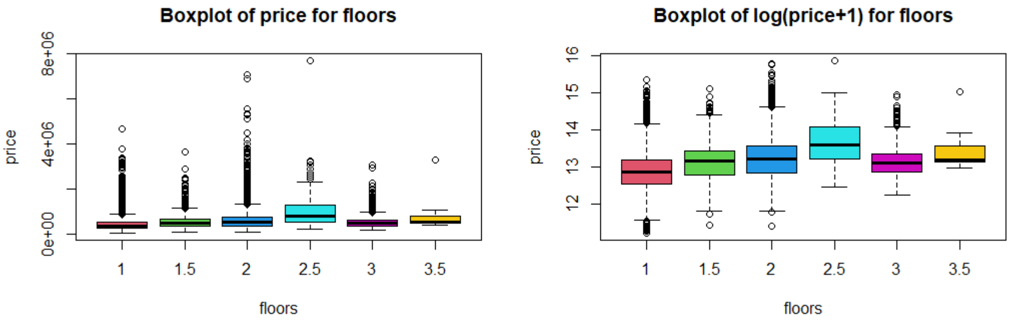
As we can see the histogram of `price`, the distribution of `price` tend to right skewed, showing that most of the house have the same price, and very few houses have the high price. However, the histogram of `log(price+1)` have the bell curve, show that it more follows the normal distribution than the `price` .

Box plot graph

We will plot the box plot of variable `price` and `log(price+1)` according to the `floors` category.

```r
par(mfrow=c(1,2))
boxplot(price ~ floors,data = newData,
        main = "Boxplot of price for floors",
        col = c(2,3,4,5,6,7))

boxplot(price ~ floors, data = newData_2,
        main = "Boxplot of log(price+1) for floors",
        col = c(2,3,4,5,6,7))
```

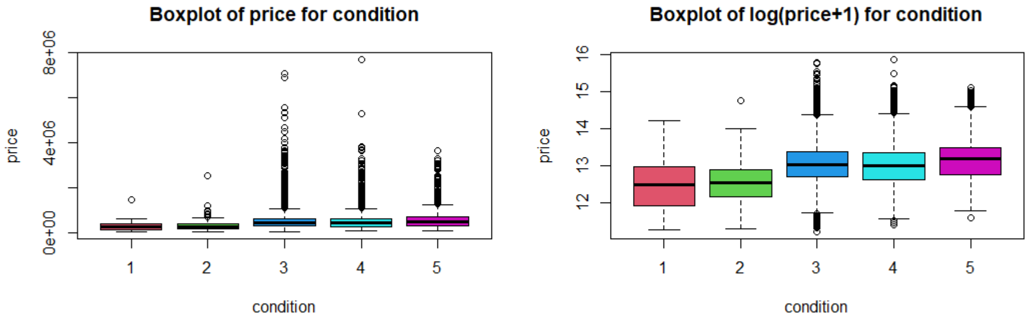


Hình 3: Boxplot for floors

For the graph `log(price+1)` according to `floors`, we see the distribution is different when the `floors` is different. We predict that `floors` is a factor affect the change of `log(price+1)`. Next, we categorized our box plot according to the **condition**.
We will plot the box plot of variable `price` and `log(price+1)` according to the `condition` category.

```r
par(mfrow=c(1,2))
boxplot(price ~ condition, data=newData,
        main = "Boxplot of price for condition",
        col = c(2,3,4,5,6,7))

boxplot(price ~ condition, data=newData_2,
        main = "Boxplot of log(price+1) for condition",
        col = c(2,3,4,5,6,7))
```

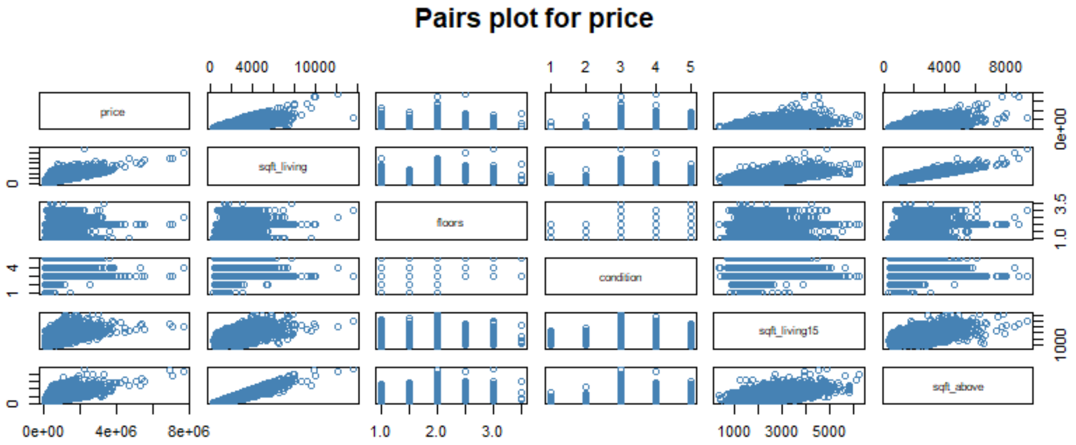


Hình 4: Boxplot for condition

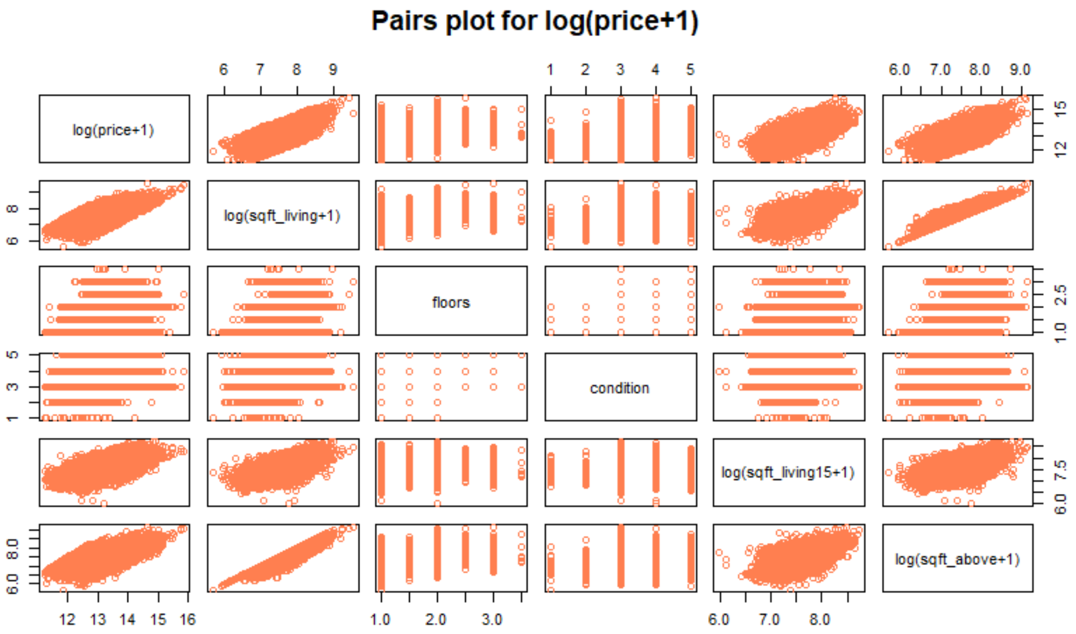
For the graph `log(price+1)` according to `condition`, we see the distribution is different when the `condition` is different. We predict that `condition` is a factor affect the change of `log(price+1)`.
Pairs

```
pairs(~newData$price + newData$sqft_living + newData$floors + newData$condition + newData$
    sqft_living15 +newData$sqft_above,
      data = newData,
      col = "steel blue",
      labels = c("price","sqft_living","floors","condition","sqft_living15","sqft_above"),
      main = "Pairs plot for price"
      )

pairs(~newData_2$price + newData_2$sqft_living + newData_2$floors + newData_2$condition +
    newData_2$sqft_living15 +newData_2$sqft_above,
      data = newData_2,
      col = "coral",
      labels = c("log(price+1)","log(sqft_living+1)","floors","condition","log(sqft_living15
    +1)","log(sqft_above+1)"),
      main = "Pairs plot for log(price+1)"
      )
```



Hình 5: Pairs plot for price



Hình 6: Pairs plot for log(price+1)

Through two pairs plots above we can see that, if base on the pairs plot of `price`, `sqft_living`, `sqft_living15`. `sqft_above`, most of the scatter plots can't show the exactly relationship between variables (for example between the `price` and the `sqft_above`).
However, with the second pairs plot, with the log transformation, the relationship among the variables have been pointed our very clearly. All the correlation at here are positive.
In conclusion, we can see that the log transformation help us explained the relationship among variables more efficiently.

4.2.4 Fitting linear regression models

Before going straight into finding the model, we will plit our dataset into two datasets, a training set `train` and tesing set `test`. The training set will count for 90% of the original data, and testing set will count for 10% rest.

```
sample <- sample(c(TRUE, FALSE), nrow(newData_2), replace=TRUE, prob=c(0.9,0.1))
train  <- newData_2[sample, ]
test   <- newData_2[!sample, ]
```

Listing 4: Spliting data

We will find our model from the training set `train`.
Our linear regression model will include:

- Dependent variable : `log(price+1)`
- Independent variables : `log(sqft_living+1)`, `floors`, `condition`, `log(sqft_above+1)`, `log(sqft_living15+1)`

Our model equation can be expressed as follow:

$$\texttt{log(price+1)} = \beta_0 + \beta_1\texttt{log(sqft\_living)} + \beta_2\texttt{floors} + \beta_3\texttt{condition}+$$
$$+\beta_4\texttt{log(sqft\_above+1)} + \beta_5\texttt{log(sqft\_living15+1)} + \epsilon$$

After estimating all the variables through the `lm()` function. Our result are:

```
lm_model1 <- lm(price ~  sqft_living + floors + condition + sqft_above + sqft_living15,
                 data = train)
summary(lm_model1)
```

Listing 5: Estimating coefficient

- $\hat{\beta}_0 = 5.4554$
- $\hat{\beta}_1 = 0.6881$
- $\hat{\beta}_2 = 0.1380$
- $\hat{\beta}_3 = 0.0869$
- $\hat{\beta}_4 = -0.1851$
- $\hat{\beta}_5 = 0.4328$
- $R^2_{adj} = 0.4969$ (calculate for later use)

Our results seem logical except the β_4, it's negative, means that when the `log(sqft_above+1)` increases, the `log(price+1)` decreases. Therefore, we will have to check for the multicollinearity condition.

<5 : Moderately

```
vif(lm_model1)
```

We will calculate the the `vif()` function. Our result are:

```
  sqft_living         floors      condition     sqft_above  sqft_living15
     4.883326       1.604811       1.093304       5.570194       2.391377
```

5 - 10

Listing 6: R output

We can see that the `log(sqft_above+1)` cause moderate multicollinearity. Therefore, our solution in this case is remove the variable `log(sqft_above+1)`.
After removing `log(sqft_above+1)`, our model equation becomes:

$$\texttt{log(price+1)} = \beta_0 + \beta_1\texttt{log(sqft\_living)} + \beta_2\texttt{floors}+$$
$$+\beta_3\texttt{condition} + \beta_4\texttt{log(sqft\_living15+1)} + \epsilon$$

After estimating all the variables through the `lm()` function. Our result are:

```
lm_model2 <- lm(price ~ sqft_living + floors + condition + sqft_living15 ,
        data = train)
summary(lm_model2)
```

- $\hat{\beta}_0 = 5.3222$
- $\hat{\beta}_1 = 0.5647$
- $\hat{\beta}_2 = 0.1009$
- $\hat{\beta}_3 = 0.09157$
- $\hat{\beta}_4 = 0.3978$

And also just to make sure, we will calculate again the VIF

```
vif(lm_model2)
```

Our result are:

```
  sqft_living        floors     condition sqft_living15
     2.439654      1.244998      1.085430      2.275251
```

Listing 7: R output

Now our result is more beautiful. Our estimated linear regression now is:

$$\begin{aligned}\texttt{log(price+1)} &= 5.3191 + 0.5677 \times \texttt{log(sqft\_living)} + 0.1 \times \texttt{floors} + \\ &+ 0.0897 \times \texttt{condition} + 0.3959 \times \texttt{log(sqft\_living15+1)}\end{aligned}$$

Test for significance of regression

Hypothesis $H_0 : \beta_i = 0 (i = 0, 1, 2, 3, 4)$ (all the regressor variables don't contribute significantly to the model)

Hypothesis $H_0 : \beta_i \neq 0 (i = 0, 1, 2, 3, 4)$ (at least one of the regressor variables contributes significantly to the model).

The $Pr(> |t|)$ of all the variables are less than the significant level $\alpha = 0.05$, thus we can reject the null hypothesis H_0. In conclusion, the coefficient of all the variables are meaningful to our model.

R^2 and adjusted R^2

The R^2 is 49.3% and the R^2_{adj} is 49.29%, both explain the variability of dependent variable response to the independent variables. It implies in 100% the variability of `log(price+1)`, 49.29% mainly cause by these factors `log(sqft_living)`, `floors`,`condition` and `log(sqft_living15+1)`

Moreover, the adjusted R^2_{adj} also help us to guard against **overfitting** in this case. As we remember, the R^2_{adj} of the model before we remove `log(sqft_above+1)` is 49.69%, and after remove our R^2_{adj}, means decreases by 0.4%, which is relatively small, meaning that including this regressor are not really useful.

Check if assumptions of the model are fulfilled

The next step is to check if the model's assumptions are completely fulfilled so that we can determine if the model is appropriate to the task as well as the credibility of the model.

Recall the assumptions of the regression model: $Y_i = \beta_0 + \beta_1 \times X_1 + \beta_2 \times X_2 + \cdots + \beta_i \times X_i + \epsilon_i, i = 1, 2, \ldots, n$

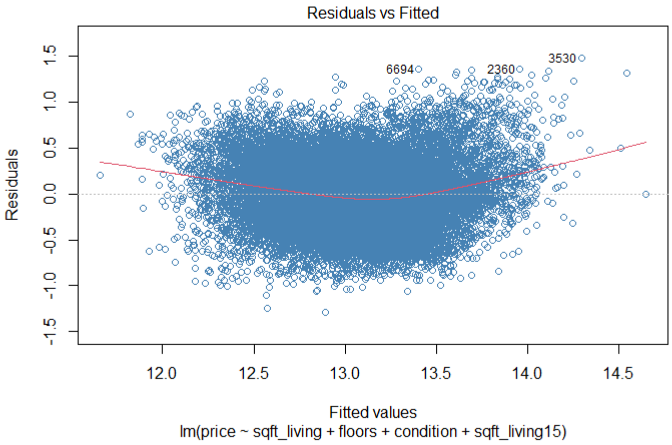
Linearity of data: the relationship between predictor variable X (independent variables) and dependent variable Y is assumed to be linear.

The errors terms are assumed to be:

- Normally distributed.
- The variance is constant. $\epsilon_i \sim N(0, \sigma^2)$
- The errors $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$ are independent of each other.

In this section, we will perform residual analysis to test the model's assumptions:

```r
plot(lm_model2, col = "steel blue", which = 1)
```

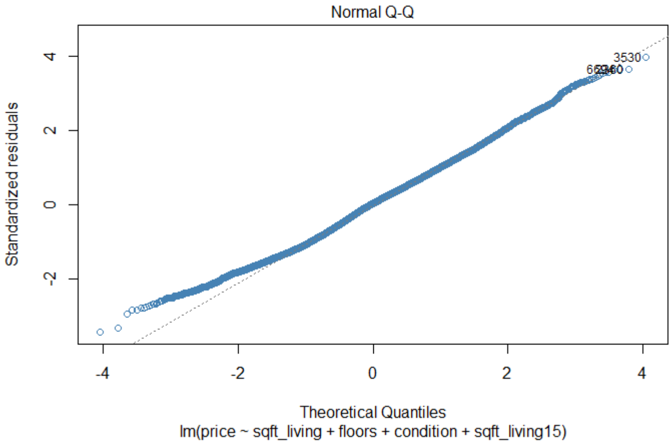


ε constant

Hình 7: Residuals vs Fitted

Look over the plot, we can see that the red line is a curve, here it seems like a parabola curve, means that the assumption of linearity of the data is not satisfied. The variability here is not approximately equal all the way along. Thus, our assumption the variance is uniform is not satisfied either.

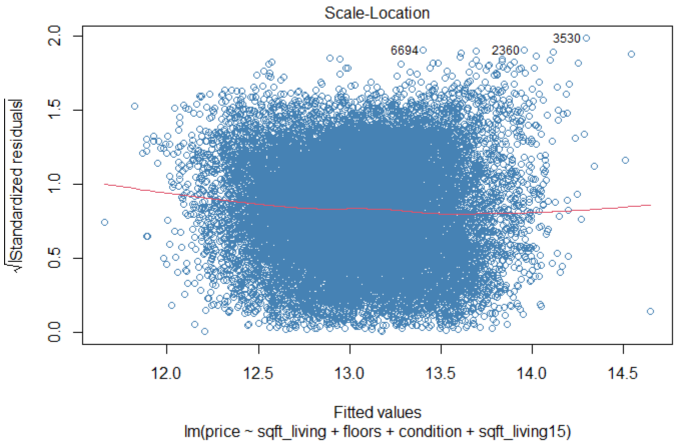
```
plot(lm_model2, col = "steel blue", which = 2)
```



Hình 8: Normal Q-Q plot

In the second plot, We see that mostly all the points lie on the 45-degree line, just some points don't so our assumption about the normal distributed is satisfied.

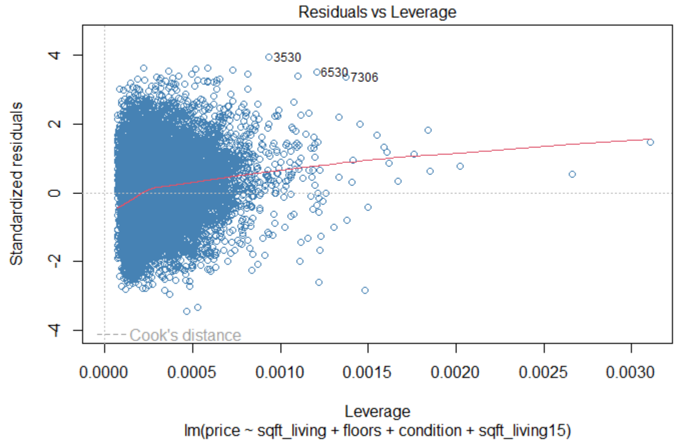
```
plot(lm_model2, col = "steel blue", which = 3)
```



Hình 9: Scale-Location

The third plot (Scale - Location), in our case, the red line is a curve and the residuals are unevenly scattered around this line our assumption about constant variance is not satisfied.

```
plot(lm_model2, col = "steel blue", which = 5)
```



Hình 10: Residuals vs Leverage

The fourth graph (Residuals vs Leverage), here we can't see any res dashed line (Cook's distance), means none of the points are really highly influential.

4.2.5 Prediction

In this section, we will use the `predict()` function to test the data set `test` we have splited before.

```
predicts <- predict(lm_model2, newdata = test)
actuals <- test$price
evaluate <- data.frame(actuals,predicts)
summary(evaluate)
```

```
    actuals          predicts
 Min.   :11.29   Min.   :11.85
 1st Qu.:12.68   1st Qu.:12.79
 Median :13.02   Median :13.04
 Mean   :13.05   Mean   :13.05
 3rd Qu.:13.38   3rd Qu.:13.31
 Max.   :15.74   Max.   :14.34
```

Listing 8: R output

```
m <- lm(actuals~predicts,data = evaluate)
summary(m)
```

```
Call:
lm(formula = actuals ~ predicts, data = evaluate)

Residuals:
     Min       1Q   Median       3Q      Max
-1.10216 -0.26863  0.00372  0.25311  1.38050

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.22739    0.28269  -0.804    0.421
predicts     1.01735    0.02166  46.976   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.3743 on 2119 degrees of freedom
Multiple R-squared:  0.5101,  Adjusted R-squared:  0.5099
F-statistic:  2207 on 1 and 2119 DF,  p-value: < 2.2e-16
```

Listing 9: R output

Conclusion

Based on the given result of function summary the model m have an R^2 value of 51.01% and Adjusted R^2 of 0.5505, this indicates that over 50.1% of the actual `price` can be explained by the predictions of `lm\_model2`. Our model is moderately good.

5 Activity 2

5.1 Data introduction

Dataset **SkillCraft** is used for evaluating the factors affecting on the league index of players. Variables in dataset:

- `GameID`: Unique ID number for each player
- `LeagueIndex`: Bronze, Silver, Gold, Platinum, Diamond, Master, GrandMaster, and Professional leagues coded 1-8 (Ordinal)
- `Age`: Age of each player (ages)
- `HoursPerWeek`: Reported hours spent playing per week (hours)
- `TotalHours`: Reported total hours spent playing (integer)
- `APM`: Action per minute (action/minute)

- `SelectByHotkeys`: Number of unit or building selections made using hotkeys per timestamp (units(buildings))
- `AssignToHotkeys`: Number of units or buildings assigned to hotkeys per timestamp (units(buildings))
- `UniqueHotkeys`: Number of unique hotkeys used per timestamp (hotkeys/timestamp)
- `MinimapAttacks`: Number of attack actions on minimap per timestamp (attack actions/timestamp)
- `MinimapRightClicks`: Number of right-clicks on minimap per timestamp (right clicks/timestamp)
- `NumberOfPACs`: Number of PACs per timestamp (PACs/timestamp)
- `GapBetweenPACs`: Mean duration in milliseconds between PACs
- `ActionLatency`: Mean latency from the onset of a PACs to their first action in milliseconds
- `ActionsInPAC`: Mean number of actions within each PAC (actions/PAC)
- `TotalMapExplored`: The number of 24x24 game coordinate grids viewed by the player per timestamp
- `WorkersMade`: Number of SCVs, drones, and probes trained per timestamp
- `UniqueUnitsMade`: Unique units made per timestamp
- `ComplexUnitsMade`: Number of ghosts, infestors, and high templars trained per timestamp
- `ComplexAbilitiesUsed`: Abilities requiring specific targeting instructions used per timestamp

5.2 Import data

```
SkillCraft <- read.csv("~/Study/221/Probability_and_Statistics/Assignment/Data/SkillCraft.csv
    ")
```

5.3 Data cleaning

We will use `sum()`, `is.na` for counting N/A values:

```
sum(is.na(SkillCraft))
```

We get the result 0, so there is no N/A values in our dataset(SkillCraft). Moreover, It is easy to conclude that GameID do not have any effect on the league index of player. Besides, according to exprerience from playing Age Of Empires (same type as Skill Craft), so we conclude that WorkersMade, UniqueUnitsMade, ComplexUnitsMade, ComplexAbilitiesUsed have a little of influence on LeagueIndex. So we decide to remove these variables by using `subset()` to get a more exactly result.

```
SkillCraftCleaned<-subset(SkillCraft,select=-c(GameID, WorkersMade, UniqueUnitsMade,
    ComplexUnitsMade, ComplexAbilitiesUsed))
```

5.4 Data visualization

All of variables in dataset except for league index are continuous variables.

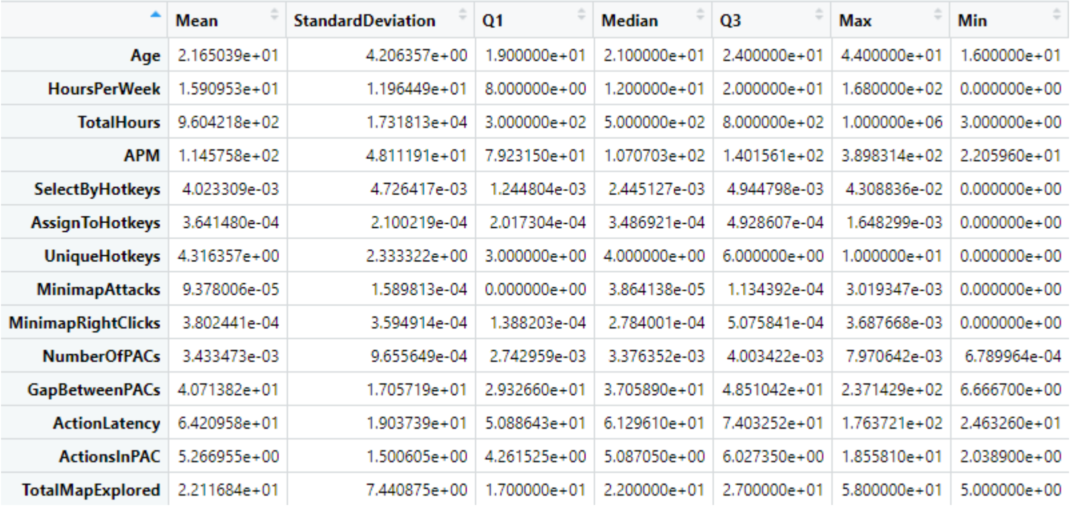
5.4.1 Descriptive statistics

Use apply(), mean, sd, quantile, median,max, min for descriptive statistics for all variables (except forleague index).

```
Mean<-apply(SkillCraftCleaned[,c(2:15)],2,mean)
StandardDeviation<-apply(SkillCraftCleaned[,c(2:15)],2,sd)
Q1<-apply(SkillCraftCleaned[,c(2:15)],2,quantile,probs=0.25)
Median<-apply(SkillCraftCleaned[,c(2:15)],2,quantile,probs=0.5)
Q3<-apply(SkillCraftCleaned[,c(2:15)],2,quantile,probs=0.75)
Max<-apply(SkillCraftCleaned[,c(2:15)],2,max)
Min<-apply(SkillCraftCleaned[,c(2:15)],2,min)
```

Use data.frame() to input all descriptive statistics values into 1 new dataframe.

```
DiscriptiveStatistics<-data.frame(Mean,StandardDeviation,Q1,Median,Q3,Max,Min)
```



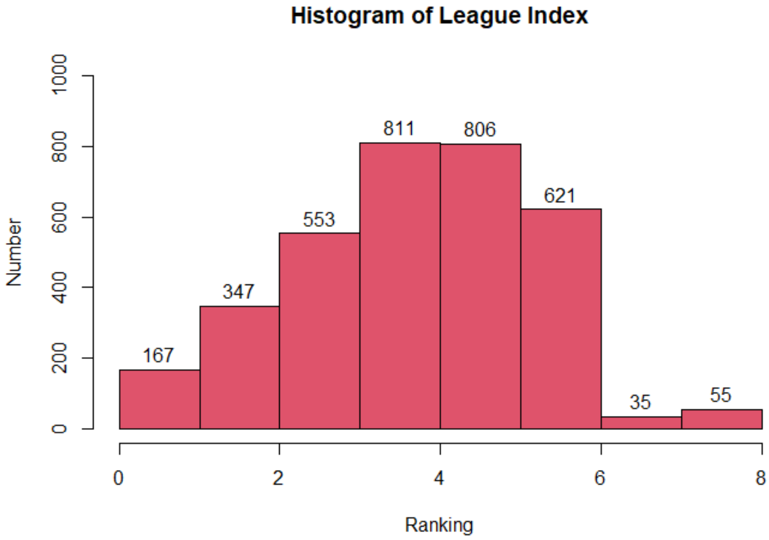
	Mean	StandardDeviation	Q1	Median	Q3	Max	Min
Age	2.165039e+01	4.206357e+00	1.900000e+01	2.100000e+01	2.400000e+01	4.400000e+01	1.600000e+01
HoursPerWeek	1.590953e+01	1.196449e+01	8.000000e+00	1.200000e+01	2.000000e+01	1.680000e+02	0.000000e+00
TotalHours	9.604218e+02	1.731813e+04	3.000000e+02	5.000000e+02	8.000000e+02	1.000000e+06	3.000000e+00
APM	1.145758e+02	4.811191e+01	7.923150e+01	1.070703e+02	1.401561e+02	3.898314e+02	2.205960e+01
SelectByHotkeys	4.023309e-03	4.726417e-03	1.244804e-03	2.445127e-03	4.944798e-03	4.308836e-02	0.000000e+00
AssignToHotkeys	3.641480e-04	2.100219e-04	2.017304e-04	3.486921e-04	4.928607e-04	1.648299e-03	0.000000e+00
UniqueHotkeys	4.316357e+00	2.333322e+00	3.000000e+00	4.000000e+00	6.000000e+00	1.000000e+01	0.000000e+00
MinimapAttacks	9.378006e-05	1.589813e-04	0.000000e+00	3.864138e-05	1.134392e-04	3.019347e-03	0.000000e+00
MinimapRightClicks	3.802441e-04	3.594914e-04	1.388203e-04	2.784001e-04	5.075841e-04	3.687668e-03	0.000000e+00
NumberOfPACs	3.433473e-03	9.655649e-04	2.742959e-03	3.376352e-03	4.003422e-03	7.970642e-03	6.789964e-04
GapBetweenPACs	4.071382e+01	1.705719e+01	2.932660e+01	3.705890e+01	4.851042e+01	2.371429e+02	6.666700e+00
ActionLatency	6.420958e+01	1.903739e+01	5.088643e+01	6.129610e+01	7.403252e+01	1.763721e+02	2.463260e+01
ActionsInPAC	5.266955e+00	1.500605e+00	4.261525e+00	5.087050e+00	6.027350e+00	1.855810e+01	2.038900e+00
TotalMapExplored	2.211684e+01	7.440875e+00	1.700000e+01	2.200000e+01	2.700000e+01	5.800000e+01	5.000000e+00

Hình 11: Discriptive Statistics

5.4.2 Graph Plotting

Use hist() for drawing frequency distribution for league index:

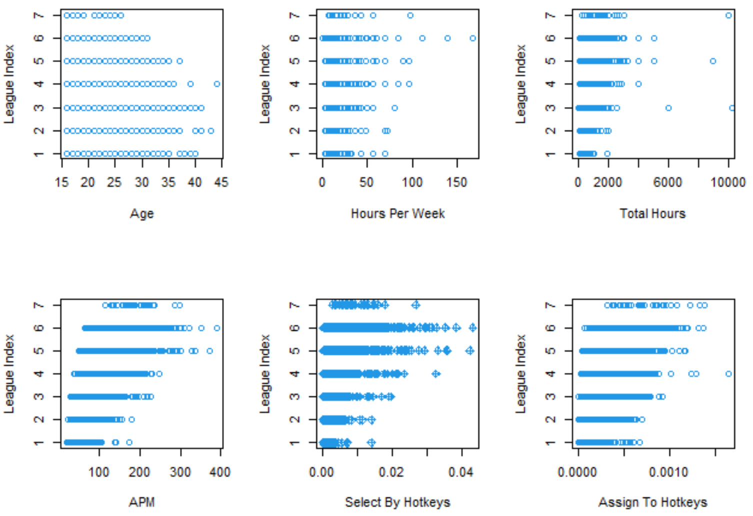
```
hist(SkillCraft$LeagueIndex,xlab="Ranking",ylab="Number",main="Histogram of League Index"
 ,label=T,xlim=c(0,8),ylim= c(0,1000),breaks=c(0,1,2,3,4,5,6,7,8),col=2)
```

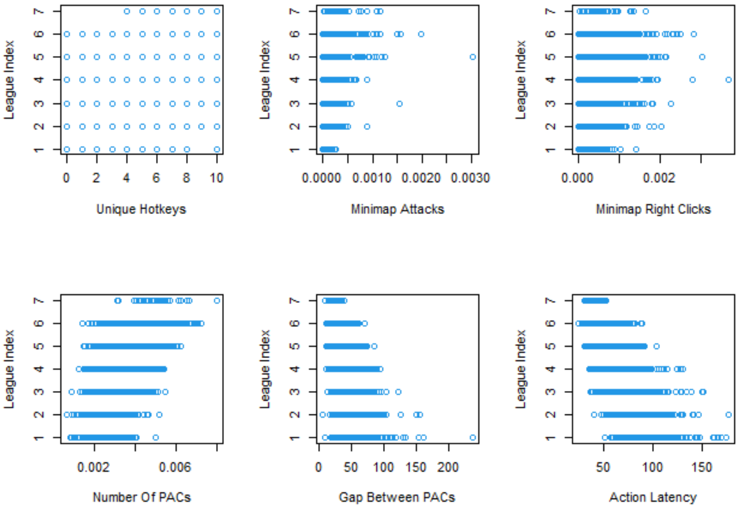


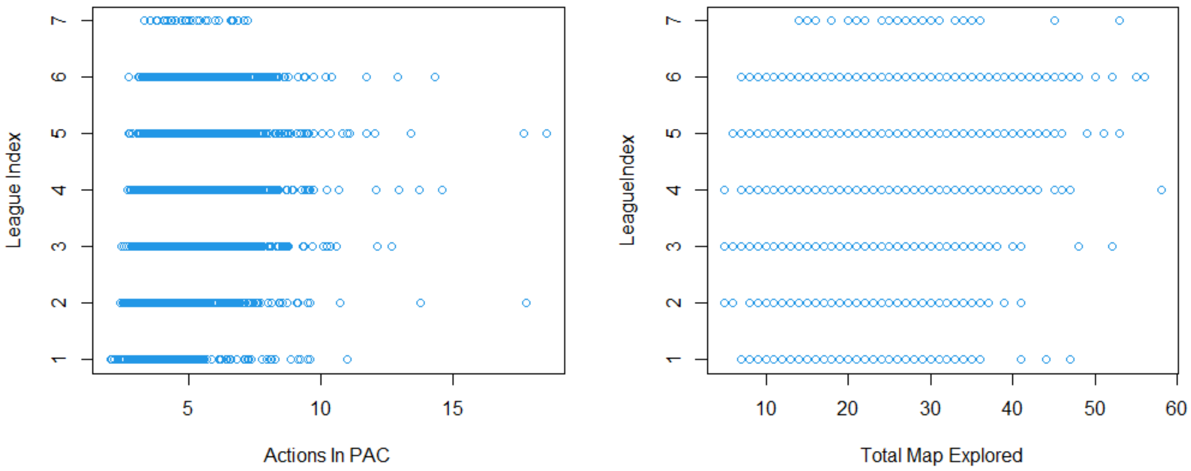
Hình 12: Histogram of League Index

Looking over the plot, we can see that the data is quite normally distributed, the highest column is 811 and the lowest column is 35.
Use `plot()` for drawing distribution graph of "LeagueIndex" according to other variables, `par(mfrow=c())` for plotting graphs in one figure:

```
par(mfrow=c(2,3))
plot(SkillCraftCleaned$Age,SkillCraft$LeagueIndex,xlab='Age',ylab='League Index',col=4)
plot(SkillCraftCleaned$HoursPerWeek,SkillCraftCleaned$LeagueIndex,xlab='Hours Per Week',ylab='League Index',col=4)
plot(SkillCraftCleaned$TotalHours,SkillCraftCleaned$LeagueIndex,xlab='Total Hours',ylab='League Index', xlim=c(0,10000),col=4)
plot(SkillCraftCleaned$APM,SkillCraftCleaned$LeagueIndex,xlab='APM',ylab='League Index',col=4)
plot(SkillCraftCleaned$SelectByHotkeys,SkillCraftCleaned$LeagueIndex,xlab='Select By Hotkeys',ylab='League Index',col=4)
plot(SkillCraftCleaned$AssignToHotkeys,SkillCraftCleaned$LeagueIndex,xlab='Assign To Hotkeys',ylab='League Index',col=4)
plot(SkillCraftCleaned$UniqueHotkeys,SkillCraftCleaned$LeagueIndex,xlab='Unique Hotkeys',ylab='League Index',col=4)
plot(SkillCraftCleaned$MinimapAttacks,SkillCraftCleaned$LeagueIndex,xlab='Minimap Attacks',ylab='League Index',col=4)
plot(SkillCraftCleaned$MinimapRightClicks,SkillCraftCleaned$LeagueIndex,xlab='Minimap Right Clicks',ylab='League Index',col=4)
plot(SkillCraftCleaned$NumberOfPACs,SkillCraftCleaned$LeagueIndex,xlab='Number Of PACs',ylab='League Index',col=4)
plot(SkillCraftCleaned$GapBetweenPACs,SkillCraftCleaned$LeagueIndex,xlab='Gap Between PACs',ylab='League Index',col=4)
plot(SkillCraftCleaned$ActionLatency,SkillCraftCleaned$LeagueIndex,xlab='Action Latency',ylab='League Index',col=4)
plot(SkillCraftCleaned$ActionsInPAC,SkillCraftCleaned$LeagueIndex,xlab='Actions In PAC',ylab='League Index',col=4)
plot(SkillCraftCleaned$TotalMapExplored,SkillCraftCleaned$LeagueIndex,xlab='Total Map Explored',ylab='LeagueIndex',col=4)
```


League Index
1 2 3 4 5 6 7
15 20 25 30 35 40 45
Age
0 50 100 150
Hours Per Week
0 2000 6000 10000
Total Hours
100 200 300 400
APM
0.00 0.02 0.04
Select By Hotkeys
0.0000 0.0010
Assign To Hotkeys


League Index
1 2 3 4 5 6 7
0 2 4 6 8 10
Unique Hotkeys
0.0000 0.0010 0.0020 0.0030
Minimap Attacks
0.000 0.002
Minimap Right Clicks
0.002 0.006
Number Of PACs
0 50 100 150 200
Gap Between PACs
50 100 150
Action Latency



Hình 13: Distribution of `LeagueIndex` according to each independent variables.

Prediction: From above figure, there are linear relations between `LeagueIndex` and `HoursPerWeek`, `APM`, `SelectByHotkeys`, `AssignToHotkeys`, `UniqueHotkeys`, `MinimapAttacks`, `NumberOfPACs`, `GapBetweenPACs`, `ActionLatency`, so these variables will affect to our linear model more than remaining variables that have weak linearity or little of linearity with `LeagueIndex`.

5.5 Building linear regression model

Building a linear regression model will help us determine which elements actually and how they affect the player's league index in the game Star-Craft.

5.5.1 Split data set

To begin with, we will split our data set into 2 parts : `train` and `test`. This will enable us to evaluate the efficiency of the model more persuasively in the final steps.

```
sample <- sample(c(TRUE, FALSE), nrow(SkillCraft), replace=TRUE, prob=c(0.8,0.2))
train  <- SkillCraft[sample, ]
test   <- SkillCraft[!sample, ]
```

5.5.2 Correlation test

After that, we will check the correlation between each pairs of the possible variables for the linear model to see if each of them are independent. **This is because** one important assumption of linear model is that the error terms among observations are not influenced by each other (avoid multicollinearity).

```
correlate<-cor(train[,c(
              'HoursPerWeek'
              ,  'Age'
              ,'TotalHours'
              ,'SelectByHotkeys'
              ,'UniqueHotkeys'
              ,'MinimapRightClicks'
```

```
                ,'ActionsInPAC'
                ,'APM'
                ,'AssignToHotkeys'
                ,'MinimapAttacks'
                ,'NumberOfPACs'
                ,'TotalMapExplored'
                ,'GapBetweenPACs'
                ,'ActionLatency'
              )])
```

The line of codes above will create a table of correlations between each pair of independent variables.
Then from the table, we will take out the cells with value 1 because this is the correlation between the same variables and seek for value that has its absolute value equal or over 0.8.

```
corr <- sort(correlate, decreasing = TRUE)
corr <- corr[which(corr != 1)]
minc <- corr[which(corr <= -0.8, arr.ind = T)] # take out the under - 0.8
maxc <- corr[which(corr >= 0.8, arr.ind = T)] # and over 0.8
which(correlate == minc, arr.ind = T)
```

```
##                 row col
## ActionLatency   14  11
## NumberOfPACs    11  14
which(correlate == maxc, arr.ind = T)
##                   row col
## APM                 8   4
## SelectByHotkeys     4   8
```

Listing 10: R output

Based on the result, we got 2 pairs of variables which have significant correlation values are `ActionLatency` with `NumberOfPACs` and `APM` with `SelectedByHotkeys`. These 2 pairs of variables will be considered to be replaced later when we build our model.

5.5.3 Start to build our model

Let consider our first model (called `model1`) is a linear regression model consist of independent variables: `Age`, `HoursPerWeek`, `TotalHours`, `SelectByHotkeys`, `UniqueHotkeys`, `MinimapRightClicks`, `ActionsInPAC`, `APM`, `AssignToHotkeys`, `MinimapAttacks`, `NumberOfPACs`, `TotalMapExplored`, `GapBetweenPACs`, `ActionLatency`.
The model can be consider based on the formula as follow :

$$\begin{aligned}LeagueIndex = &\beta_0 + \beta_1 \times Age + \beta_2 \times HoursPerWeek + \beta_3 \times TotalHours + \\ &+ \beta_4 \times SelectedByHotkeys + \beta_5 \times UniqueHotkeys + \beta_6 \times MinimapRightClicks + \\ &+ \beta_7 \times ActionsInPAC + \beta_8 \times APM + \beta_9 \times AssignToHotkeys + \\ &+ \beta_{10} \times MiniMapAttacks + \beta_{11} \times NumberOfPACs + \beta_{12} \times TotalMapExplored + \\ &+ \beta_{13} \times GapBetweenPACs + \beta_{14} \times ActionLatency\end{aligned}$$

We start to estimate the variables $\beta_i,\ i = 0, 1, ..., 14$ base on the train dataset:

```
model1<-lm(LeagueIndex~
               Age
              +HoursPerWeek
              +TotalHours
              +SelectByHotkeys
              +UniqueHotkeys
              +MinimapRightClicks
              +ActionsInPAC
              +APM
              +AssignToHotkeys
              +MinimapAttacks
              +NumberOfPACs
              +TotalMapExplored
              +GapBetweenPACs
              +ActionLatency
              ,data=train)
summary(model1)
##
## Call:
## lm(formula = LeagueIndex ~ Age + HoursPerWeek + TotalHours +
##     SelectByHotkeys + UniqueHotkeys + MinimapRightClicks + ActionsInPAC +
##     APM + AssignToHotkeys + MinimapAttacks + NumberOfPACs + TotalMapExplored +
##     GapBetweenPACs + ActionLatency, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2021 -0.6320  0.0462  0.6861  2.7393
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.488e+00  3.709e-01   9.402  < 2e-16 ***
## Age                 9.866e-03  4.824e-03   2.045 0.040945 *
## HoursPerWeek        6.666e-03  1.682e-03   3.964 7.56e-05 ***
## TotalHours         -1.117e-06  9.933e-07  -1.125 0.260764
## SelectByHotkeys     3.626e+01  1.399e+01   2.592 0.009595 **
## UniqueHotkeys       3.342e-02  9.383e-03   3.561 0.000376 ***
## MinimapRightClicks  1.584e+01  6.020e+01   0.263 0.792497
## ActionsInPAC        8.729e-02  3.669e-02   2.379 0.017441 *
## APM                -5.973e-04  2.369e-03  -0.252 0.800996
## AssignToHotkeys     9.357e+02  1.162e+02   8.056 1.18e-15 ***
## MinimapAttacks      1.066e+03  1.340e+02   7.959 2.54e-15 ***
## NumberOfPACs        3.213e+02  7.449e+01   4.313 1.67e-05 ***
## TotalMapExplored   -7.642e-03  3.068e-03  -2.491 0.012803 *
## GapBetweenPACs     -8.770e-03  1.683e-03  -5.211 2.02e-07 ***
## ActionLatency      -2.167e-02  2.339e-03  -9.264  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.9883 on 2649 degrees of freedom
## Multiple R-squared:  0.5366, Adjusted R-squared:  0.5342
## F-statistic: 219.1 on 14 and 2649 DF,  p-value: < 2.2e-16
```

Listing 11: R output

Visualizing the model using function `summary()`, we obtain some analysis result :

$$\begin{aligned}
\widehat{\beta}_0 &= 3.488, & \widehat{\beta}_1 &= 9.866 \times 10^{-3}, \\
\widehat{\beta}_2 &= 6.666 \times 10^{-3}, & \widehat{\beta}_3 &= -1.117 \times 10^{-6}, \\
\widehat{\beta}_4 &= 3.626 \times 10^{1}, & \widehat{\beta}_5 &= 3.342 \times 10^{-2} \\
\widehat{\beta}_6 &= 1.584 \times 10^{1}, & \widehat{\beta}_7 &= 8.729 \times 10^{-2}, \\
\widehat{\beta}_8 &= -5.973 \times 10^{-4}, & \widehat{\beta}_9 &= 9.357 \times 10^{2}, \\
\widehat{\beta}_{10} &= 1.066 \times 10^{3}, & \widehat{\beta}_{11} &= 3.213 \times 10^{2} \\
\widehat{\beta}_{12} &= -7.642 \times 10^{-3}, & \widehat{\beta}_{13} &= -8.770 \times 10^{-3}, \\
\widehat{\beta}_{14} &= -2.167 \times 10^{-2}
\end{aligned}$$

For more information about the above figure, return to our previous section of describing how linear model building function in R works (***lm*** **- function)**

Having the regression estimate coefficients, we will start to test their influence to the model.

5.5.4 Hypothesis test using $P_{value}(Pr(> |t|))$

Set

- Hypothesis H_0 : $\beta_i = 0$ with $i = 0, 1, 2, \ldots, 14$ (Regression coefficient is not statistically significant)
- Hypothesis H_1: $\beta_i \neq 0$ where $i = 0, 1, 2, \ldots, 14$ (Regression coefficient is statistically significant)

Method of testing by $P_{value}(Pr(> |t|))$:

- Observe in the Coefficients section, the values (Pr(>|t|)) are the P_{values}.
- Consider the 5% significance level

We have Pr(>|t|) of `TotalHours`, `MinimapRightClicks`, `ActionsInPAC`, `APM` variables are greater than 5% significance level, so there are no sufficient evidence to reject hypothesis H_0 corresponding to these variables. Therefore, we will eliminate the coefficient corresponding to these variables to avoid suffering from over-fitting.

Rebuild the model and name it as `model2`

```
model2<-lm(LeagueIndex~
              +HoursPerWeek
              +SelectByHotkeys
              +UniqueHotkeys
              +ActionsInPAC
              +AssignToHotkeys
              +MinimapAttacks
              +NumberOfPACs
              +TotalMapExplored
              +GapBetweenPACs
              +ActionLatency
           ,data=train)
summary(model2)
##
## Call:
## lm(formula = LeagueIndex ~ +HoursPerWeek + SelectByHotkeys +
##     UniqueHotkeys + ActionsInPAC + AssignToHotkeys + MinimapAttacks +
##     NumberOfPACs + TotalMapExplored + GapBetweenPACs + ActionLatency,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2024 -0.6387  0.0575  0.6838  2.7159
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.706e+00  3.366e-01  11.012  < 2e-16 ***
## HoursPerWeek      6.195e-03  1.666e-03   3.719 0.000204 ***
## SelectByHotkeys   3.205e+01  5.048e+00   6.350 2.53e-10 ***
## UniqueHotkeys     3.539e-02  9.335e-03   3.791 0.000153 ***
## ActionsInPAC      7.910e-02  1.746e-02   4.529 6.18e-06 ***
## AssignToHotkeys   9.319e+02  1.161e+02   8.029 1.46e-15 ***
## MinimapAttacks    1.095e+03  1.320e+02   8.296  < 2e-16 ***
## NumberOfPACs      3.049e+02  4.663e+01   6.539 7.40e-11 ***
## TotalMapExplored -7.169e-03  3.034e-03  -2.363 0.018195 *
## GapBetweenPACs   -9.103e-03  1.675e-03  -5.434 6.02e-08 ***
## ActionLatency    -2.092e-02  2.287e-03  -9.150  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9886 on 2653 degrees of freedom
## Multiple R-squared:  0.5357, Adjusted R-squared:  0.5339
## F-statistic:   306 on 10 and 2653 DF,  p-value: < 2.2e-16
```

Listing 12: R output

5.5.5 Remove more variables

The new model is built and there are no variables with their coefficients have uncertainty about its statistical meaning to the model. Therefore, we will start to consider the 2 pairs of variables mentioned earlier that have high correlation which are `ActionLatency` with `NumberOfPACs` and `APM with SelectedByHotkeys`.

Now that variable `APM` is already removed, we only have to decide which one to be removed between `ActionLatency` and `NumberOfPACs`.

Because the P_{value} of `ActionLatency` ($< 2 \times 10^{-16}$) is significantly smaller than that of `NumberOfPACs` (5.96×10^{-13}) (indicates that its statistical influence to the model is greater), we will remove variable `NumberOfPACs` from the model.

```
model3<-lm(LeagueIndex~
              +HoursPerWeek
              +SelectByHotkeys
              +UniqueHotkeys
              +ActionsInPAC
              +AssignToHotkeys
              +MinimapAttacks
              +TotalMapExplored
              +GapBetweenPACs
              +ActionLatency
           ,data=train)
summary(model3)
##
## Call:
## lm(formula = LeagueIndex ~ +HoursPerWeek + SelectByHotkeys +
##     UniqueHotkeys + ActionsInPAC + AssignToHotkeys + MinimapAttacks +
##     TotalMapExplored + GapBetweenPACs + ActionLatency, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2531 -0.6483  0.0814  0.6973  3.0267
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.628e+00  1.654e-01  34.037  < 2e-16 ***
## HoursPerWeek      7.048e-03  1.674e-03   4.211 2.63e-05 ***
## SelectByHotkeys   3.686e+01  5.033e+00   7.322 3.22e-13 ***
## UniqueHotkeys     4.056e-02  9.374e-03   4.327 1.57e-05 ***
## ActionsInPAC      1.301e-02  1.435e-02   0.907    0.365
## AssignToHotkeys   1.016e+03  1.163e+02   8.740  < 2e-16 ***
## MinimapAttacks    1.072e+03  1.330e+02   8.063 1.12e-15 ***
## TotalMapExplored -1.633e-03  2.936e-03  -0.556    0.578
## GapBetweenPACs   -1.018e-02  1.680e-03  -6.056 1.60e-09 ***
## ActionLatency    -3.166e-02  1.604e-03 -19.739  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9964 on 2654 degrees of freedom
## Multiple R-squared:  0.5282, Adjusted R-squared:  0.5266
## F-statistic: 330.1 on 9 and 2654 DF,  p-value: < 2.2e-16
```

Listing 13: R output

Having rebuilt the model, we will apply again hypothesis test to remove some variables that recently appeared to have their P_{value} over 5% and form our final linear regression model.

- Hypothesis $H_0 : \beta_i = 0$ with $i = 0; 2; 4; 5; 7; 9; 10; 12; 13; 14$ (Regression coefficient is not statistically significant)
- Hypothesis H_1: $\beta_i \neq 0$ where $i = 0; 2; 4; 5; 7; 9; 10; 12; 13; 14$ (Regression coefficient is statistically significant)

Method of testing by P_{value} (Pr(>|t|)):

- Observe in the coefficients section, the values (Pr(>|t|)) are the P_{values}.
- Consider the 5% significance level

We have Pr(>|t|) of `ActionsInPAC`, `TotalMapExplored` variable are greater than 5% significance level, so there are no sufficient evidence to reject hypothesis H_0 corresponding to these variables. Therefore, we will eliminate the coefficient corresponding to these variables to avoid suffering from over-fitting.

```
model3<-lm(LeagueIndex~
              +HoursPerWeek
              +SelectByHotkeys
              +UniqueHotkeys
              +AssignToHotkeys
              +MinimapAttacks
              +GapBetweenPACs
              +ActionLatency
           ,data=train)
summary(model3)
##
## Call:
## lm(formula = LeagueIndex ~ +HoursPerWeek + SelectByHotkeys +
##     UniqueHotkeys + AssignToHotkeys + MinimapAttacks + GapBetweenPACs +
##     ActionLatency, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2554 -0.6544  0.0799  0.6972  3.0368
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.660e+00  1.124e-01  50.346  < 2e-16 ***
## HoursPerWeek     7.129e-03  1.671e-03   4.267 2.05e-05 ***
## SelectByHotkeys  3.777e+01  4.968e+00   7.603 4.00e-14 ***
## UniqueHotkeys    3.882e-02  9.201e-03   4.219 2.54e-05 ***
## AssignToHotkeys  1.009e+03  1.161e+02   8.691  < 2e-16 ***
## MinimapAttacks   1.072e+03  1.315e+02   8.155 5.30e-16 ***
## GapBetweenPACs  -1.089e-02  1.560e-03  -6.979 3.74e-12 ***
## ActionLatency   -3.112e-02  1.492e-03 -20.856  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9963 on 2656 degrees of freedom
## Multiple R-squared:  0.5279, Adjusted R-squared:  0.5267
## F-statistic: 424.3 on 7 and 2656 DF,  p-value: < 2.2e-16
```

Listing 14: R output

5.5.6 Compare now and then

Now that we have no other adjustments to add to our model, let's run some tests to see if our model does actually get better.
In our case, we chose to use **variance analysis for two linear models** using ANOVA.

- Hypothesis H_0: The two models `model1` and `model3` are equally effective.
- Hypothesis H_1: The two models `model1` and `model3` perform differently.

```
anova(model1,model3)

## Analysis of Variance Table
##
## Model 1: LeagueIndex ~ Age + HoursPerWeek + TotalHours + SelectByHotkeys +
##     UniqueHotkeys + MinimapRightClicks + ActionsInPAC + APM +
##     AssignToHotkeys + MinimapAttacks + NumberOfPACs + TotalMapExplored +
##     GapBetweenPACs + ActionLatency
## Model 2: LeagueIndex ~ +HoursPerWeek + SelectByHotkeys + UniqueHotkeys +
##     AssignToHotkeys + MinimapAttacks + GapBetweenPACs + ActionLatency
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   2614 2567.8
## 2   2621 2610.7 -7   -42.904 6.2394 2.847e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Listing 15: R output

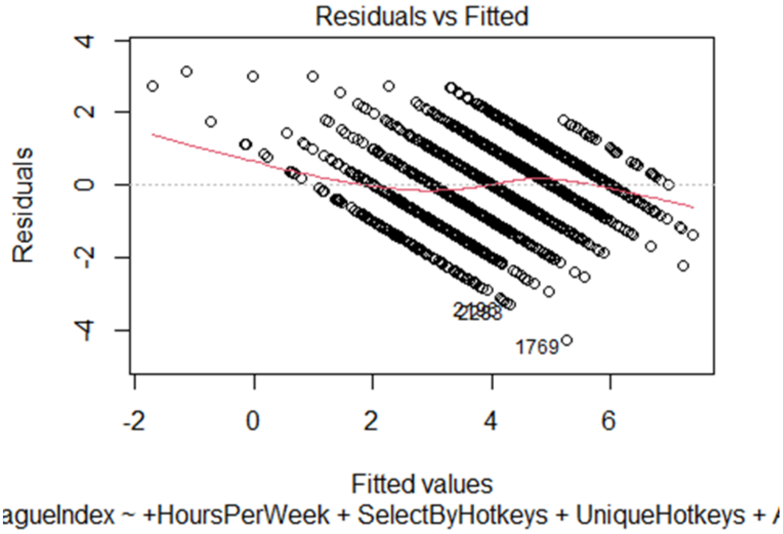
Since $P_{value} = 2.847 \times 10^{-7}$ is smaller than the 5% significance level, hypothesis H0 is rejected. And we can conclude that `model3` must have some improvements compared to `model1`.

5.5.7 Check if assumptions of the model are fulfilled

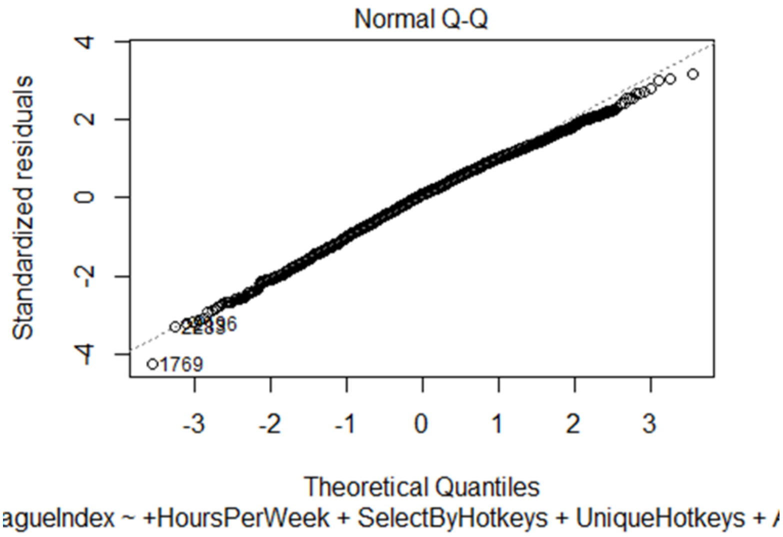
The next step is to check if the model's assumptions are completely fulfilled so that we can determine if the model is appropriate to the task as well as the credibility of the model. Recall the assumptions of the regression model:

$Y_i = \beta_0 + \beta_1 \cdot X_1 + \ldots \beta_i \cdot X_i + \epsilon_i, i = 1, \ldots n$

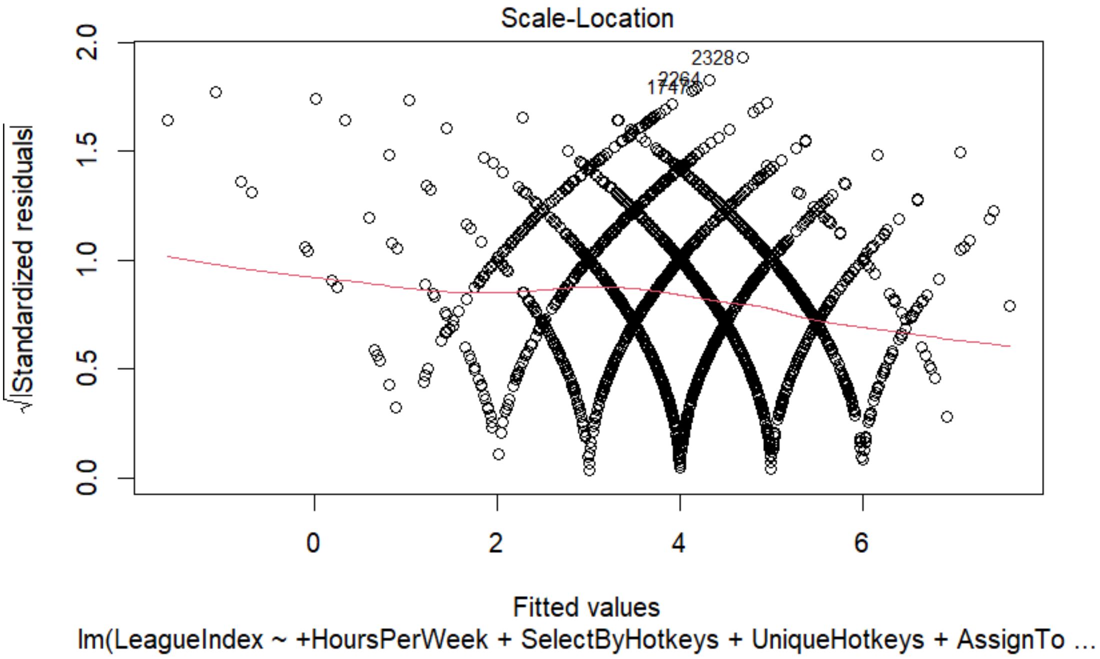
- Linearity of data: the relationship between predictor variable X (independent variables) and dependent variable Y is assumed to be linear.
- Normally distributed error.
- The variance of the errors is constant: $\epsilon_i \sim N\left(0, \sigma^2\right)$
- The errors $\epsilon_1, \ldots, \epsilon_n$ are independent of each other.



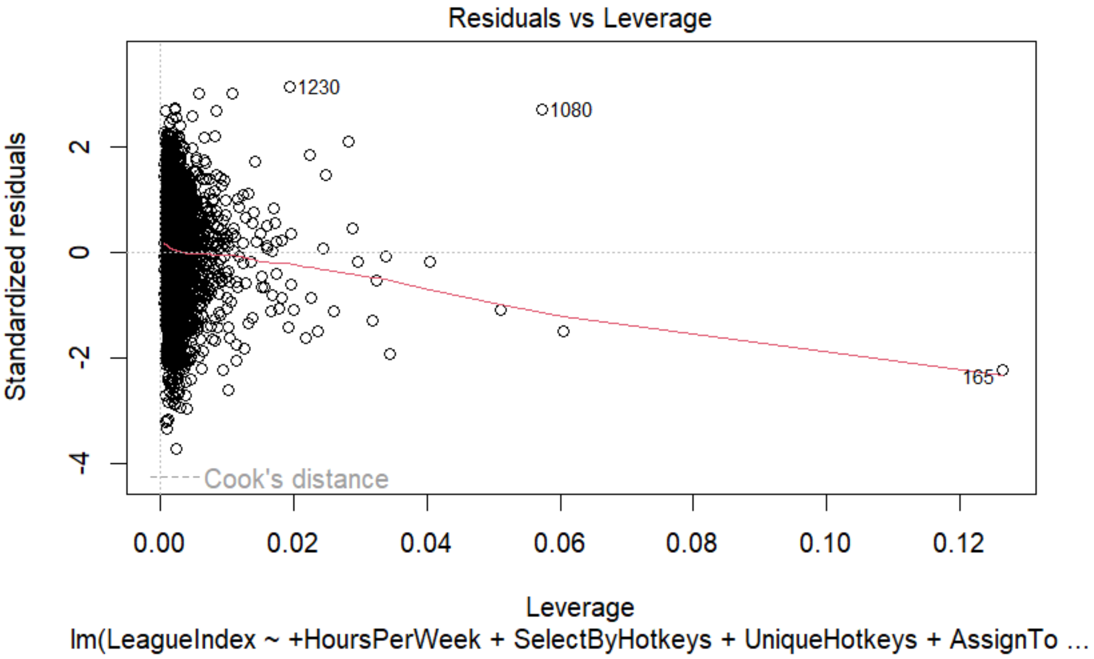
Hình 14: Residuals vs Fitted



Hình 15: Normal Q-Q



Hình 16: Scale-Location



Hình 17: Residuals vs Leverage

Conclusion:

- The Normal Q-Q graph shows that except for a few falling of on the head and tail, most of the residuals lie on the line. Therefore, the condition of normal distribution is satisfied so as the assumption of standard error.
- In the Residuals vs Fitted graph, the red line is quite curved and unstable but in general it is still very much horizontal and does not differ much from the line y = 0. So we shall conclude that the linearity of the data is relatively satisfied.
- In the Residuals vs Fitted graph, though the residuals concentrate mostly on the tail of line y = 0, they can be seen evenly spread through out this area. Also, in the Scale - Location plot, having the red line is fairly having no slopes and the residuals are distributed evenly, we shall conclude that the third assumption of constant variance of errors is not violated.
- In the last graph of Residuals vs Leverage, it is shown that there are observations 1239, 1096 and 166 which can be highly influential points in the data set. However, non of them seem to get any near to Cook's distance meaning that there is no need to remove any Outliers.

  To sum up, the linear model seem to be inappropriate for the task. However, since most of the assumptions are still satisfied, we can still expect that the model will have better accuracy for predicting larger value of League Index.

5.5.8 Prediction (on test data set)

In this final section, we will apply the model on the test data set to generate its prediction using function *predict*. As expected, the linear model produces decimal number other than integer number so we apply function *round* to get the exact result.

After that we will create linear model m out from the equation $actuals = \beta_0 + \beta_1 \times predicts$ .The data frame of this model will consist of 2 column of the predict and actual value of $LeagueIndex$ in data set $test$.

```
predicts <- predict(model3, newdata = test)
predicts <- round(predicts)
actuals <- test$LeagueIndex
evaluate <- data.frame(actuals,predicts,predicts-actuals)
summary(evaluate)
##     actuals          predicts       predicts...actuals
## Min.   :1.000   Min.   :-1.000   Min.   :-3.000000
## 1st Qu.:3.000   1st Qu.: 3.000   1st Qu.:-1.000000
## Median :4.000   Median : 4.000   Median : 0.000000
## Mean   :4.135   Mean   : 4.137   Mean   : 0.001471
## 3rd Qu.:5.000   3rd Qu.: 5.000   3rd Qu.: 1.000000
## Max.   :7.000   Max.   : 8.000   Max.   : 3.000000
##
## Call:
## lm(formula = actuals ~ predicts, data = evaluate)
##
## Residuals:
##     Min       1Q   Median       3Q      Max
## -3.00361 -0.96648 -0.00361  0.99639  2.95926
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.15213    0.15389   0.989    0.323
## predicts     0.96287    0.03598  26.762   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.02 on 678 degrees of freedom
## Multiple R-squared:  0.5137, Adjusted R-squared:  0.513
## F-statistic: 716.2 on 1 and 678 DF,  p-value: < 2.2e-16
```

Listing 16: Prediction output

Conclusion :

Based on the given result of function `summary()` the model `m` have an R^2 value of 0.5137 and R^2_{adj} of 0.513, this indicates that over 51% of the actual `LeagueIndex` can be explained by the predictions of `model3`. That will leave an amount of error of 1 in the predictions which is not very good due to the fact that `LeagueIndex` has only 7 values.

6 References

The Intuition behind the Assumptions of Linear Regression Algorithm
Multicollinearity in Regression
Statistics 101: Multiple Linear Regression, The Very Basics
How to detect and deal with Multicollinearity
Applied statistics and probability for engineers - 6th Edition by Douglas C. Motgomery, George C. Runger.
Introductory Statistics with R - 2nd edition by Peter Dalgaard.
A Beginner's Guide to R by Alain F.Zuur, Elena N.Ieno, Erik H.W.G.Messters.