MA4710

REGRESSION ANALYSIS

Regression Analysis Final Project

By: Md Nehal Salik

Submitted to: Dr. Byung-Jun Kim

Table of Contents

1. Introduction

Multiple linear regression analysis is a part of a supervised machine learning algorithm that is used to predict the continuous variable. The algorithm assumes that the relation between the dependent variable and the independent predictor variables is linear and is represented by a line of best fit. In this project, we will use the multiple linear regression in R to fit a model on SENIC data and predict the outcomes. The goal of this project is further mentioned in detail in a separate subheading.

In statistics, the linear regression model is a relationship between the continuous dependent variable and the independent one or more predictor variables. The independent variable can be both categorical and numerical. The case when we have only one independent variable then it is called the simple linear regression and if we have more than one independent variable, then it is called the multiple linear regression. For the scope of this project, we will use multiple or multivariate linear regression with both numerical and categorical variables, since we have multiple predictor variables in the data and some of the predictors are categorical.

1.1 The goal of the project

This project focuses on analyzing a dataset containing characteristics of hospitals participating in the SENIC project. The SENIC data contains 113 samples (rows) and 11 variables. We will fit a multiple linear regression (MLR) model to predict the average length of stay of all patients in a hospital. During the project, we will perform several sets of hypothesis tests associated with this model using R to come to a final most appropriate model. At the end of the project, we aim to achieve the below set of results.

- Model fitting together with justification of the model, assumptions checking, and remediation.
- Computing the point estimates of regression coefficients using a design matrix and normal equations.
- Interpreting adjusted $R2$, denoted by $adj.R2$.
- Conducting a hypothesis test for model adequacy using the overall F-test.
- Conducting multiple hypothesis tests for the regression coefficients based on t-test.
- Obtaining confidence intervals for individual regression coefficients.
- Making a conclusion based on the test and the confidence intervals.
- Model diagnostics and transformation if required.

1.2 Description of each variable in the dataset

The description of the dataset for each column in the dataset is as follows in ascending order:

Length of Stay (1): The average length of stay of all patients in the hospital (days).
Age(2): Average age of patients in years
Infection risk: Estimated probability of acquiring infection in hospital (percent).
Routine Culturing Ratio (4): Ratio of the number of cultures performed to the number of patients without signs or symptoms of hospital-acquired infection, times 100.
Routine X-ray Ratio (5): Ratio of number of X-rays performed to the number of patients without signs or symptoms of pneumonia, times 100.
Number of Beds (6): Number of beds in the hospital during the study period.
Medical School (7): Indicator of whether the hospital is associated with a medical school (1 = Yes, 2 = No).
Region (8): Indicator of the geographic region for hospital (1 = NE, 2 = NC, 3 = S, 4 = W).
Average Census (9): Number of patients per day in hospital during the study period.
Number Nurses (10): Number of full-time equivalents registered and licensed practical nurses during the study period (number of full times plus one-half the number of part-time).
Available Facilities (11): Percent of 35 potential facilities and services that are provided by the hospital.

1.3 Data collection and preparation

First thing First. The first step in any data analysis is getting the required data in a data frame.

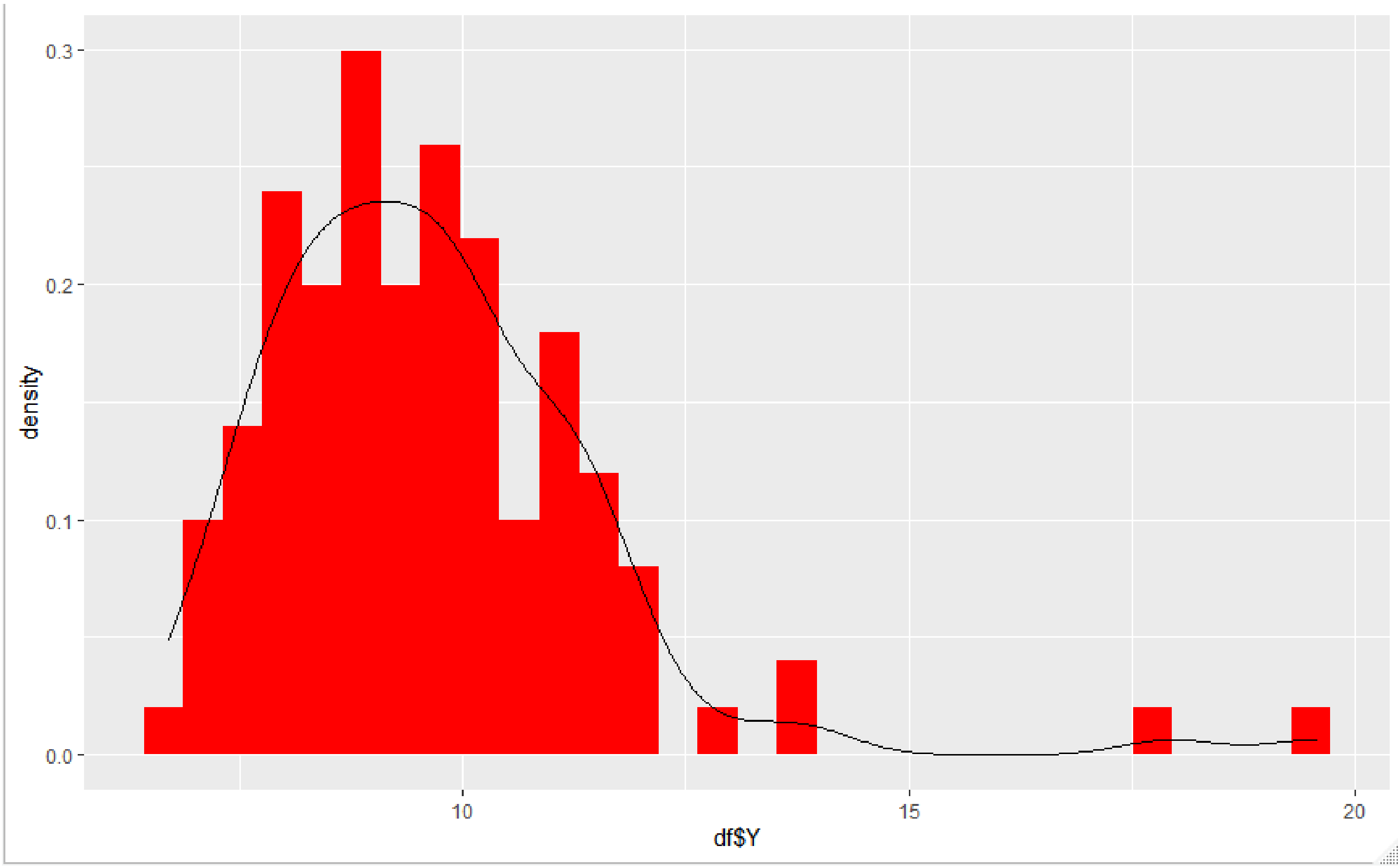
```
# READING THE DATA
senic <- read.csv("C:/Users/nehal/Desktop/MTU/MTU/Courses/MA4710 - RA/MathsFinalProject/SENIC.csv")
head(senic)
install.packages("dplyr")
library(dplyr)
# Getting the required data frame for Analysis
df <- senic
View(df)
# Creating the design matrix of X
Y <- matrix(df[,1], ncol=1)
X <- as.matrix(df[,-1])
X <- cbind(1,X)
colnames(X)[1] <- 'Intercept'
colnames(Y)[1] <- 'Y'
View(X)
View(Y)
```

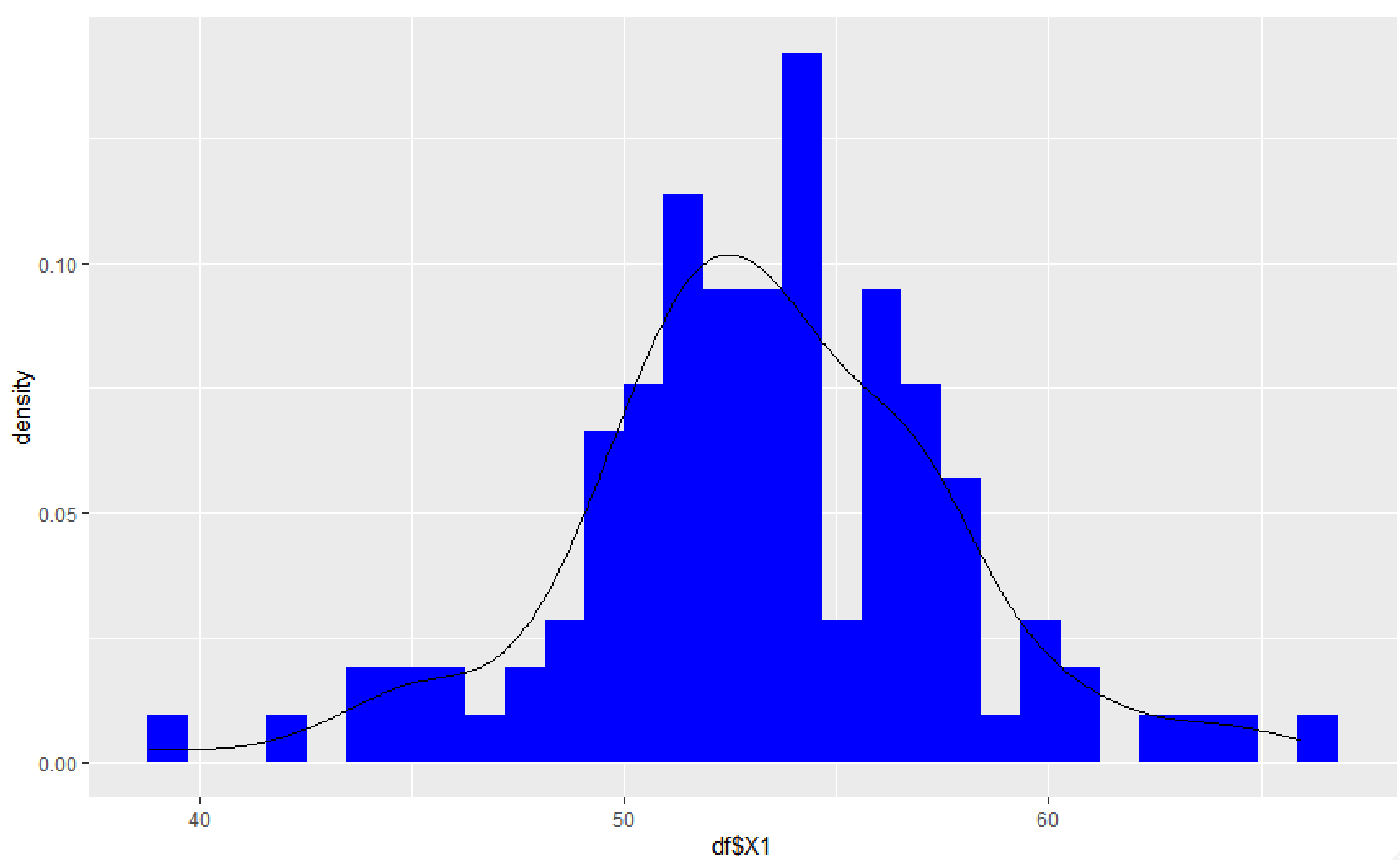
1.4 Exploratory Data Analysis

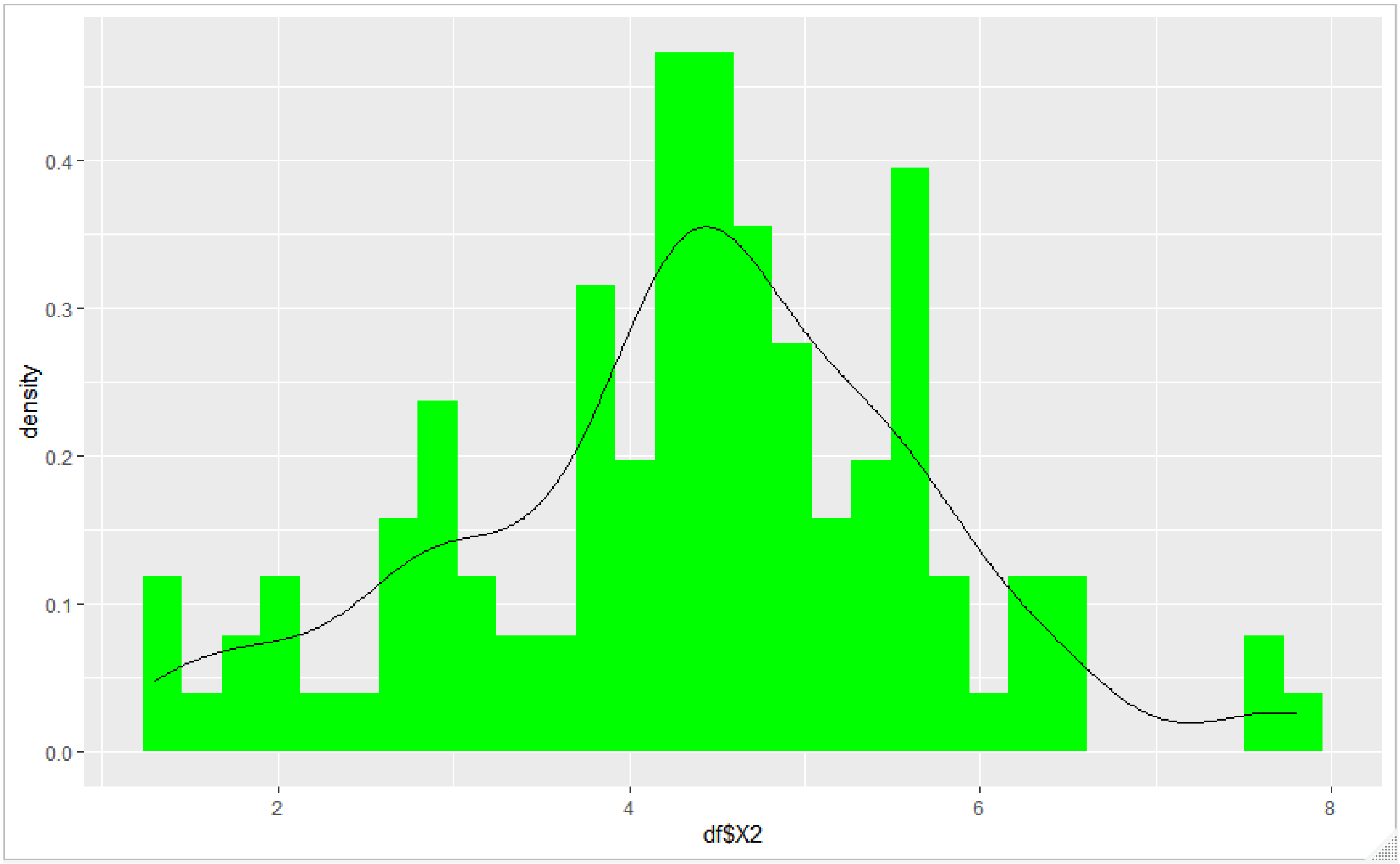
Exploratory Analysis involves performing the below steps on the provided data :

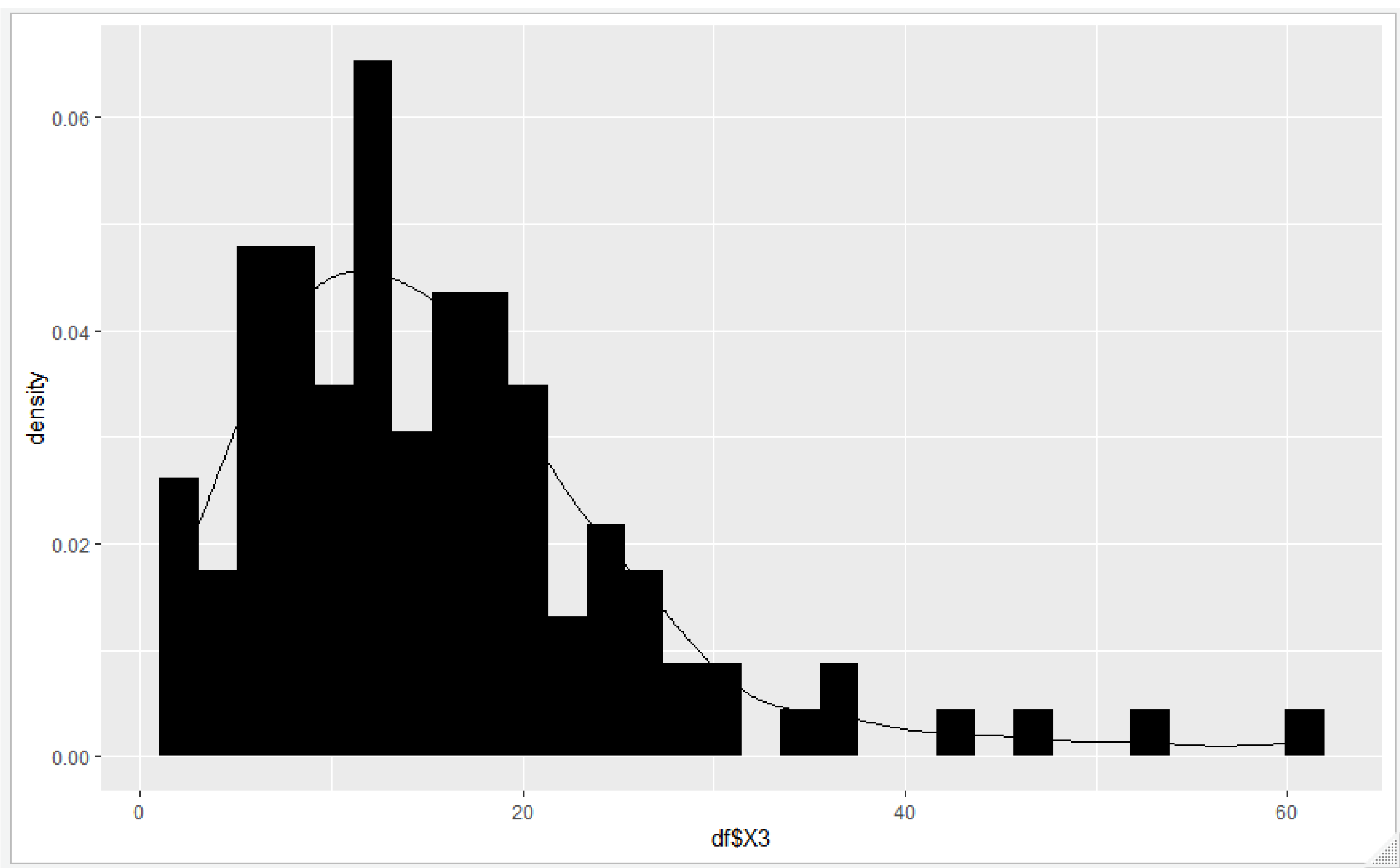
1.4.1 Checking the distribution of the target and response variable.

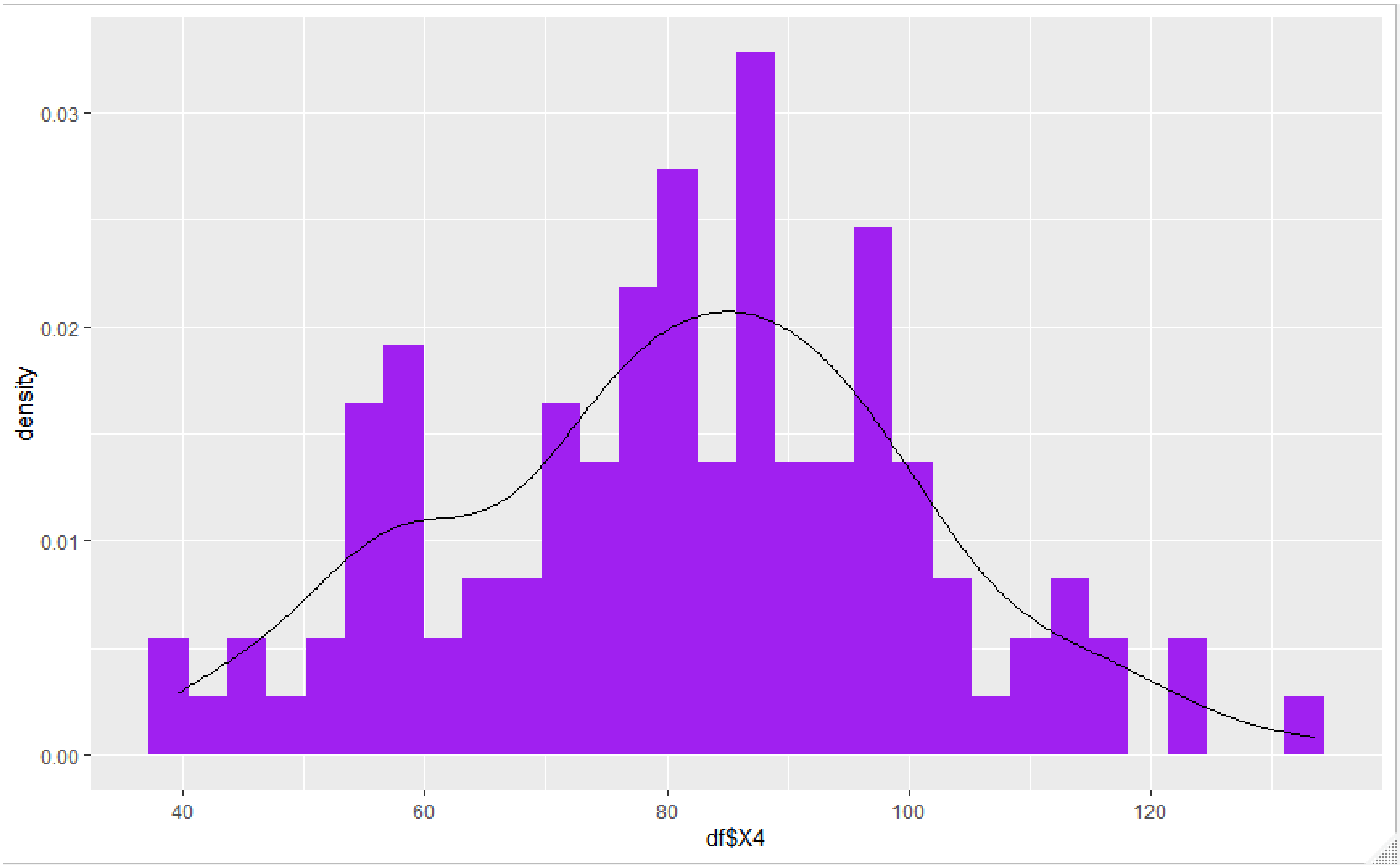
```
## Exploratory Data Analysis
# Creating the histogram of the target and predictor variables
install.packages('ggplot2')
library(ggplot2)
# Building the histogram
ggplot(data=df, aes(df$Y)) +
  geom_histogram(aes(y =..density..), fill = "red") +
  geom_density()
ggplot(data=df, aes(df$X1)) +
  geom_histogram(aes(y =..density..), fill = "blue") +
  geom_density()
ggplot(data=df, aes(df$X2)) +
  geom_histogram(aes(y =..density..), fill = "green") +
  geom_density()
ggplot(data=df, aes(df$X3)) +
  geom_histogram(aes(y =..density..), fill = "black") +
  geom_density()
ggplot(data=df, aes(df$X4)) +
  geom_histogram(aes(y =..density..), fill = "purple") +
  geom_density()
ggplot(data=df, aes(df$X5)) +
  geom_histogram(aes(y =..density..), fill = "blue") +
  geom_density()
ggplot(data=df, aes(df$X6)) +
  geom_histogram(aes(y =..density..), fill = "green") +
  geom_density()
ggplot(data=df, aes(df$X7)) +
  geom_histogram(aes(y =..density..), fill = "black") +
  geom_density()
ggplot(data=df, aes(df$X8)) +
  geom_histogram(aes(y =..density..), fill = "purple") +
  geom_density()
ggplot(data=df, aes(df$X9)) +
  geom_histogram(aes(y =..density..), fill = "blue") +
  geom_density()
ggplot(data=df, aes(df$X10)) +
  geom_histogram(aes(y =..density..), fill = "green") +
  geom_density()
```

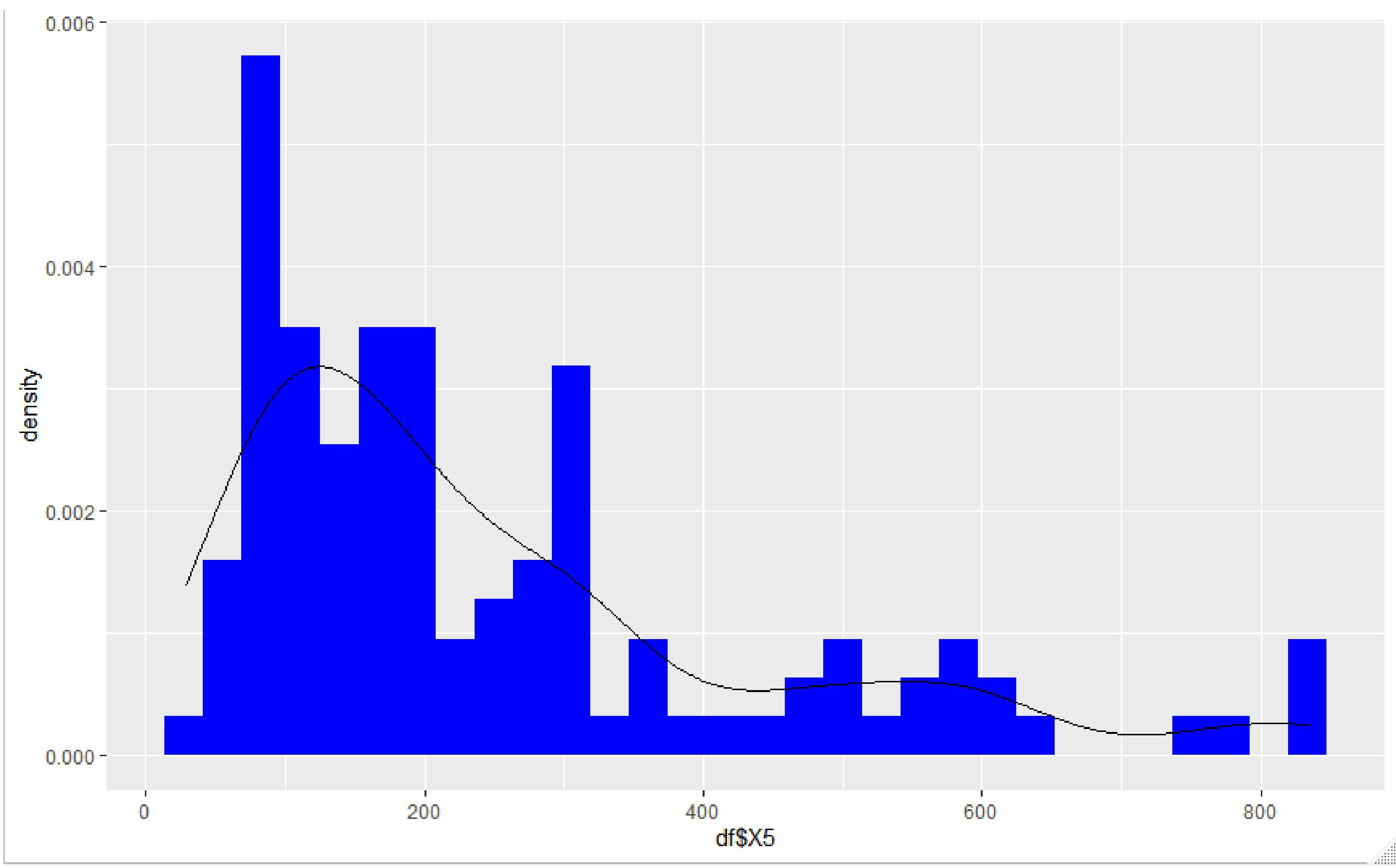

0.3
0.2
0.1
0.0
density
10
15
20
df$Y

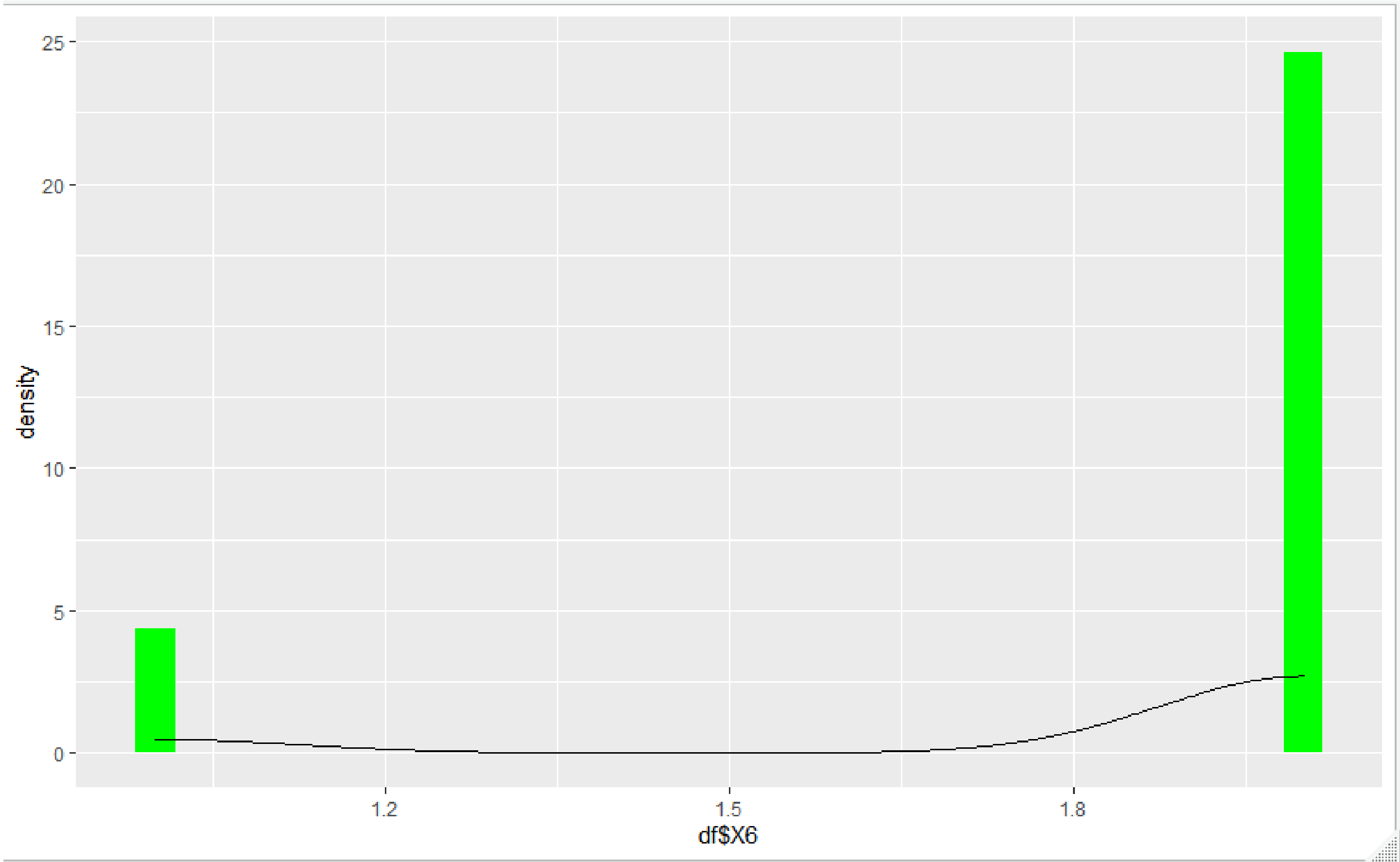

0.10
0.05
0.00
density
40
50
60
df$X1

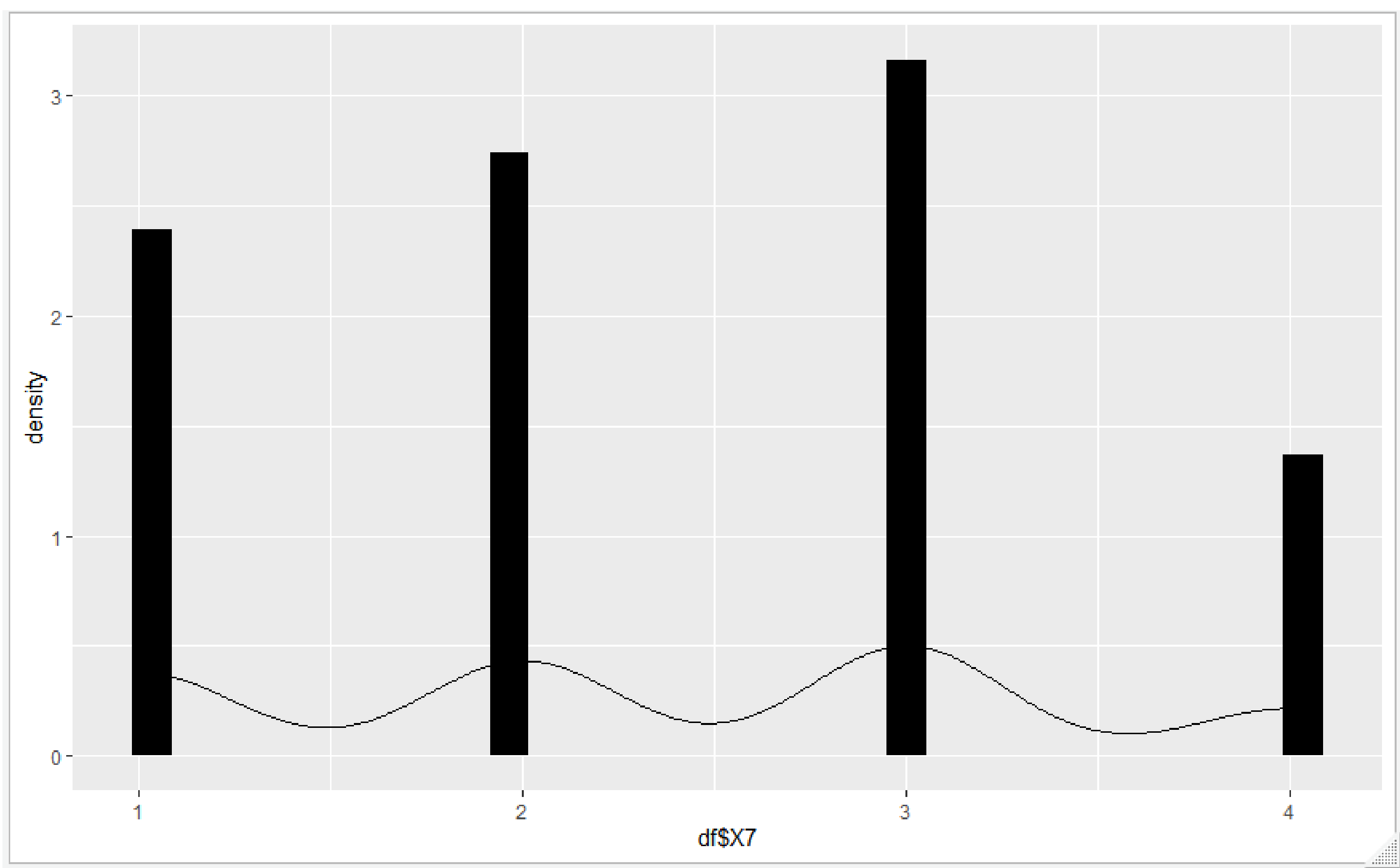

0.4
0.3
0.2
0.1
0.0
density
2
4
6
8
df$X2

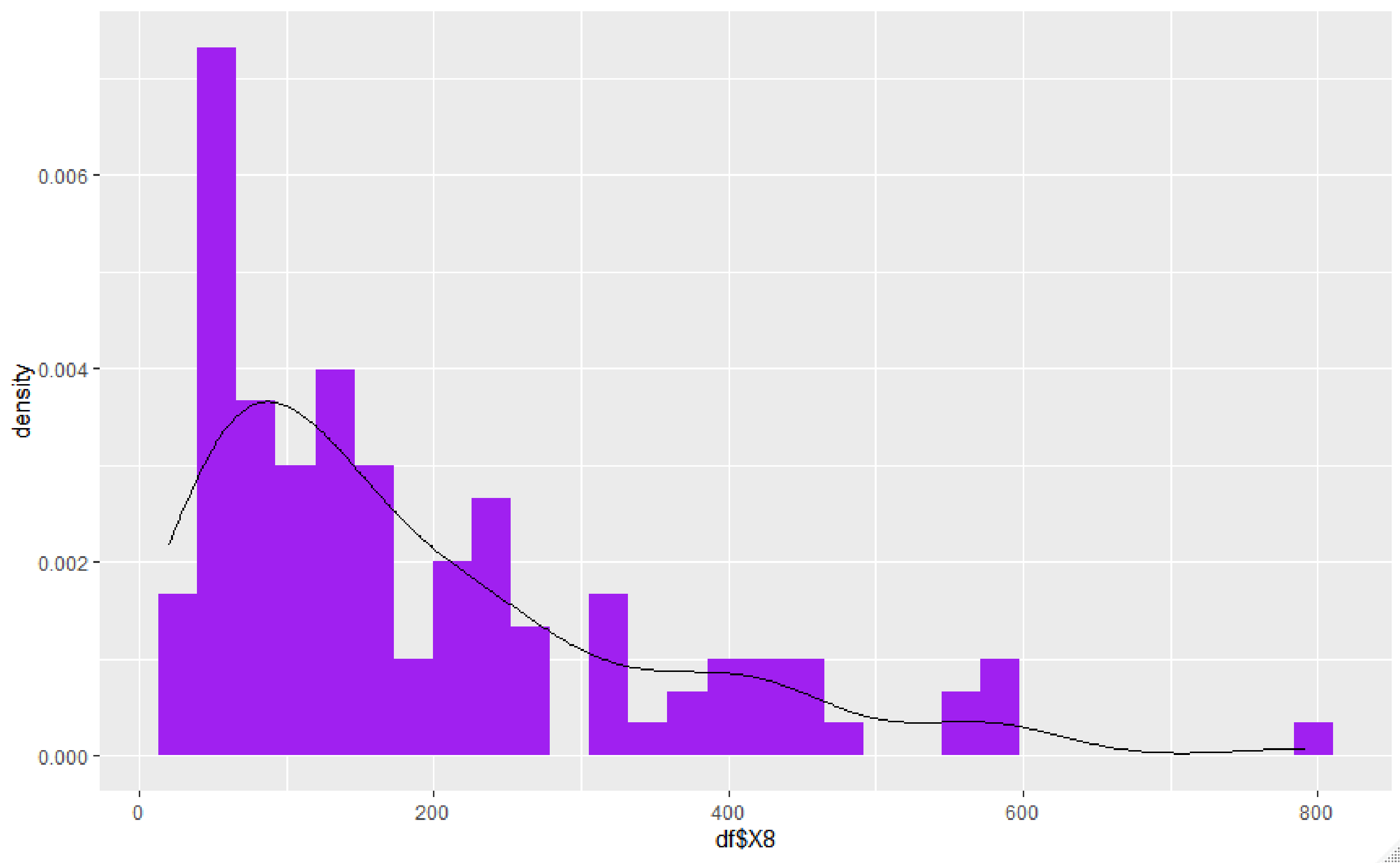

0.06
0.04
0.02
0.00
density
0
20
40
60
df$X3


0.03
0.02
0.01
0.00
density
40
60
80
100
120
df$X4

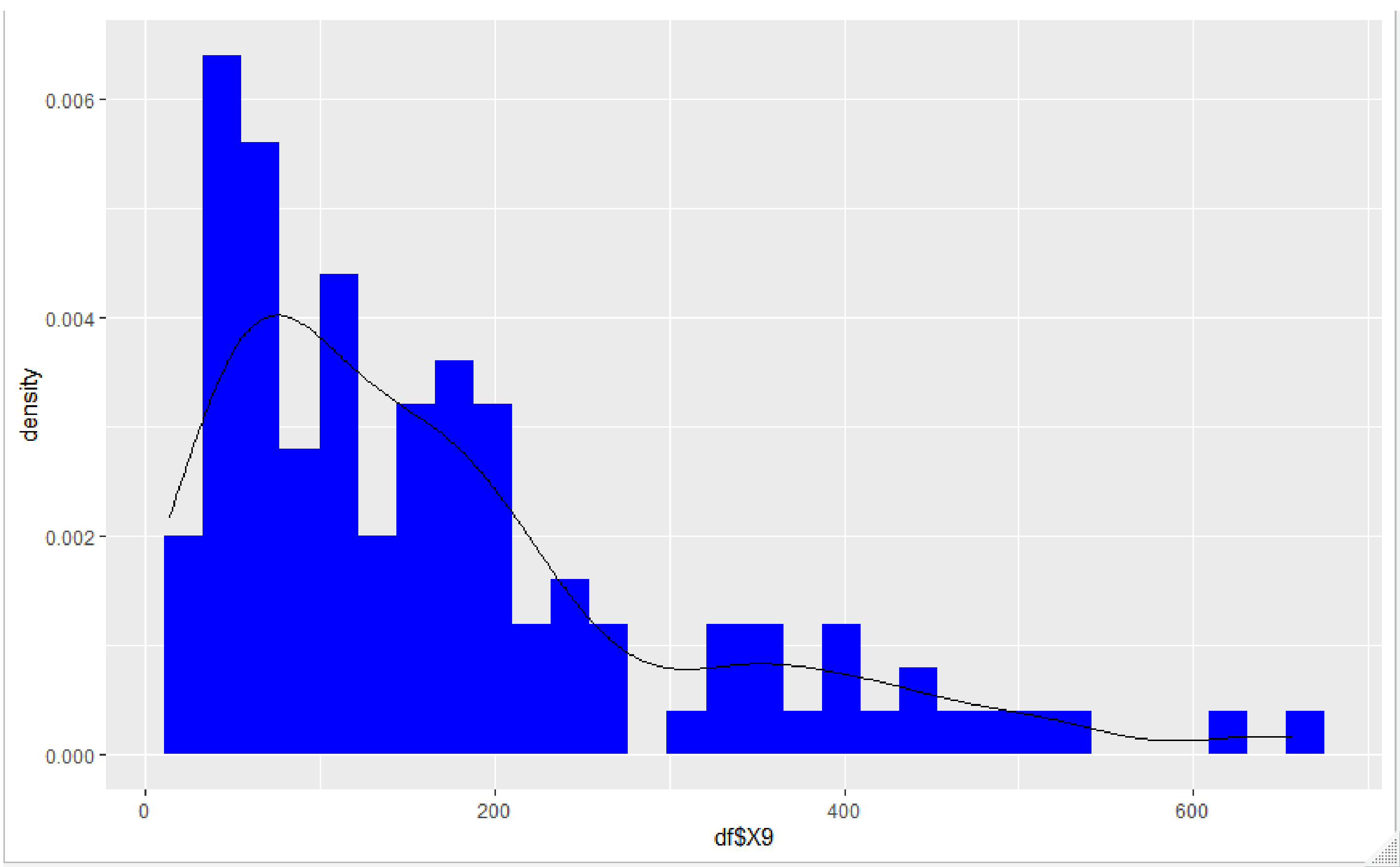

0.006
0.004
0.002
0.000
density
0
200
400
600
800
df$X5


25
20
15
10
5
0
density
1.2
1.5
1.8
df$X6

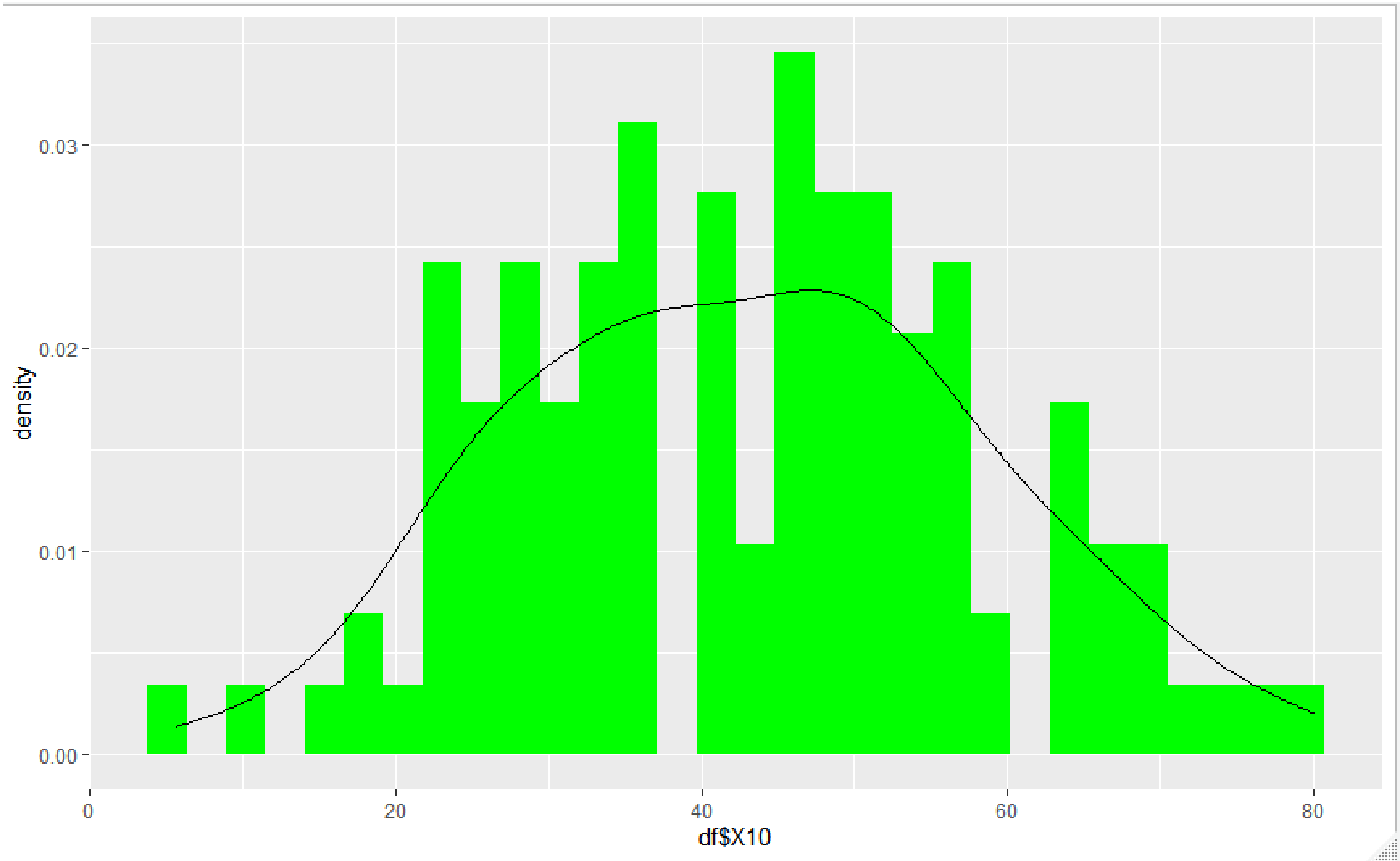

3
2
1
0
density
1
2
3
4
df$X7



0.006
0.004
0.002
0.000
density
0
200
400
600
800
df$X8



0.006
0.004
0.002
0.000
density
0
200
400
600
df$X9



1.4.2 Analysis of the summary statistics.

```
# Getting the Summary Statistics using the psych package
install.packages('psych')
library(psych)
psych::describe(df)
```

```
> psych::describe(df)
    vars   n   mean     sd median trimmed    mad  min    max  range  skew kurtosis    se
Y      1 113   9.65   1.91   9.42    9.47   1.60  6.7  19.56  12.86  2.01     7.48  0.18
X1     2 113  53.23   4.46  53.20   53.28   3.85 38.8  65.90  27.10 -0.10     0.90  0.42
X2     3 113   4.35   1.34   4.40    4.38   1.19  1.3   7.80   6.50 -0.12     0.07  0.13
X3     4 113  15.79  10.23  14.10   14.56   8.60  1.6  60.50  58.90  1.57     3.62  0.96
X4     5 113  81.63  19.36  82.30   81.48  18.98 39.6 133.50  93.90  0.01    -0.33  1.82
X5     6 113 252.17 192.84 186.00  221.59 139.36 29.0 835.00 806.00  1.34     1.10 18.14
X6     7 113   1.85   0.36   2.00    1.93   0.00  1.0   2.00   1.00 -1.93     1.74  0.03
X7     8 113   2.36   1.01   2.00    2.33   1.48  1.0   4.00   3.00  0.06    -1.14  0.09
X8     9 113 191.37 153.76 143.00  168.30 124.54 20.0 791.00 771.00  1.34     1.52 14.46
X9    10 113 173.25 139.27 132.00  151.96 103.78 14.0 656.00 642.00  1.34     1.35 13.10
X10   11 113  43.16  15.20  42.90   42.95  16.90  5.7  80.00  74.30  0.07    -0.50  1.43
```

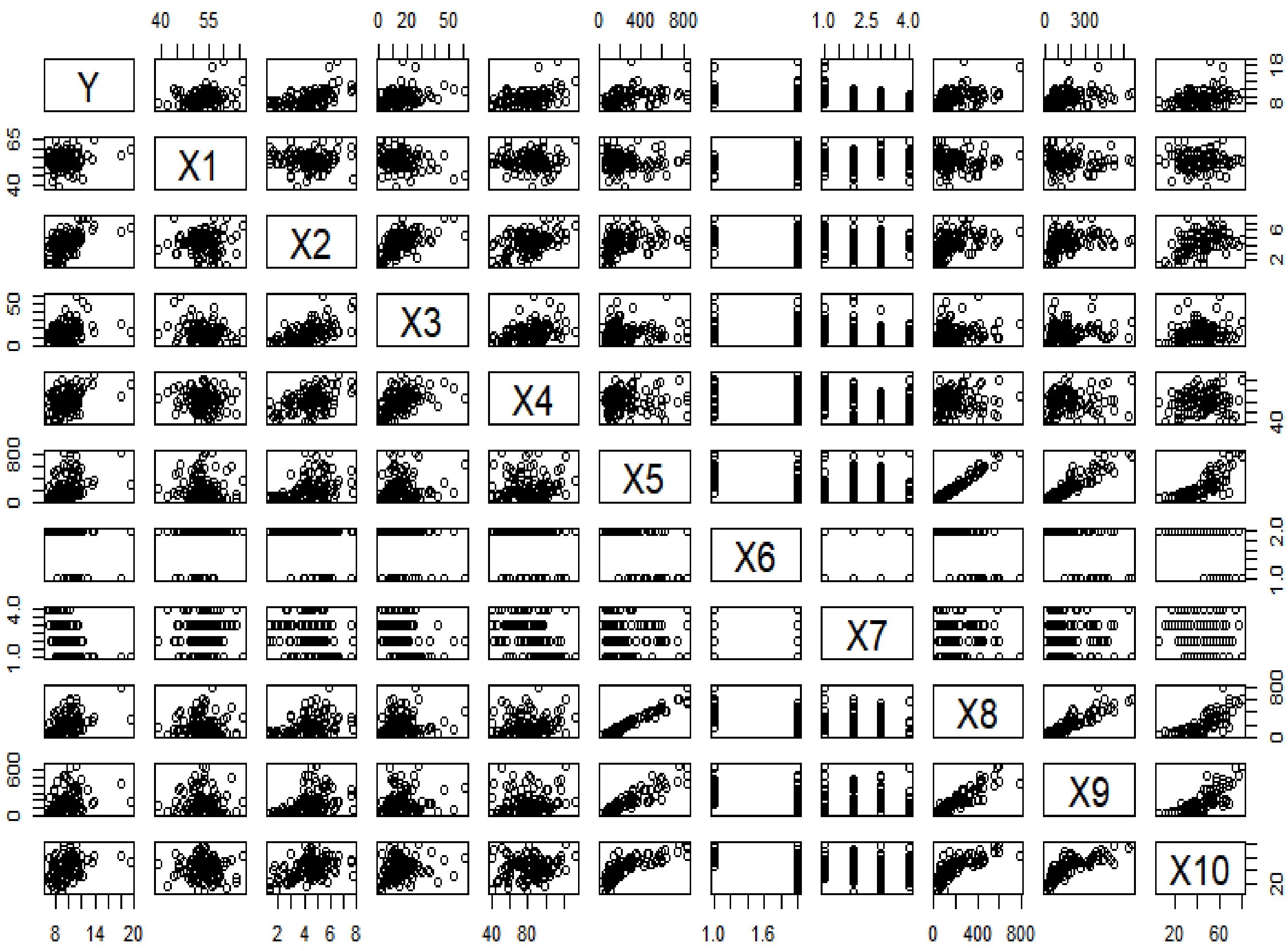
1.4.3 Creating the correlation matrix.

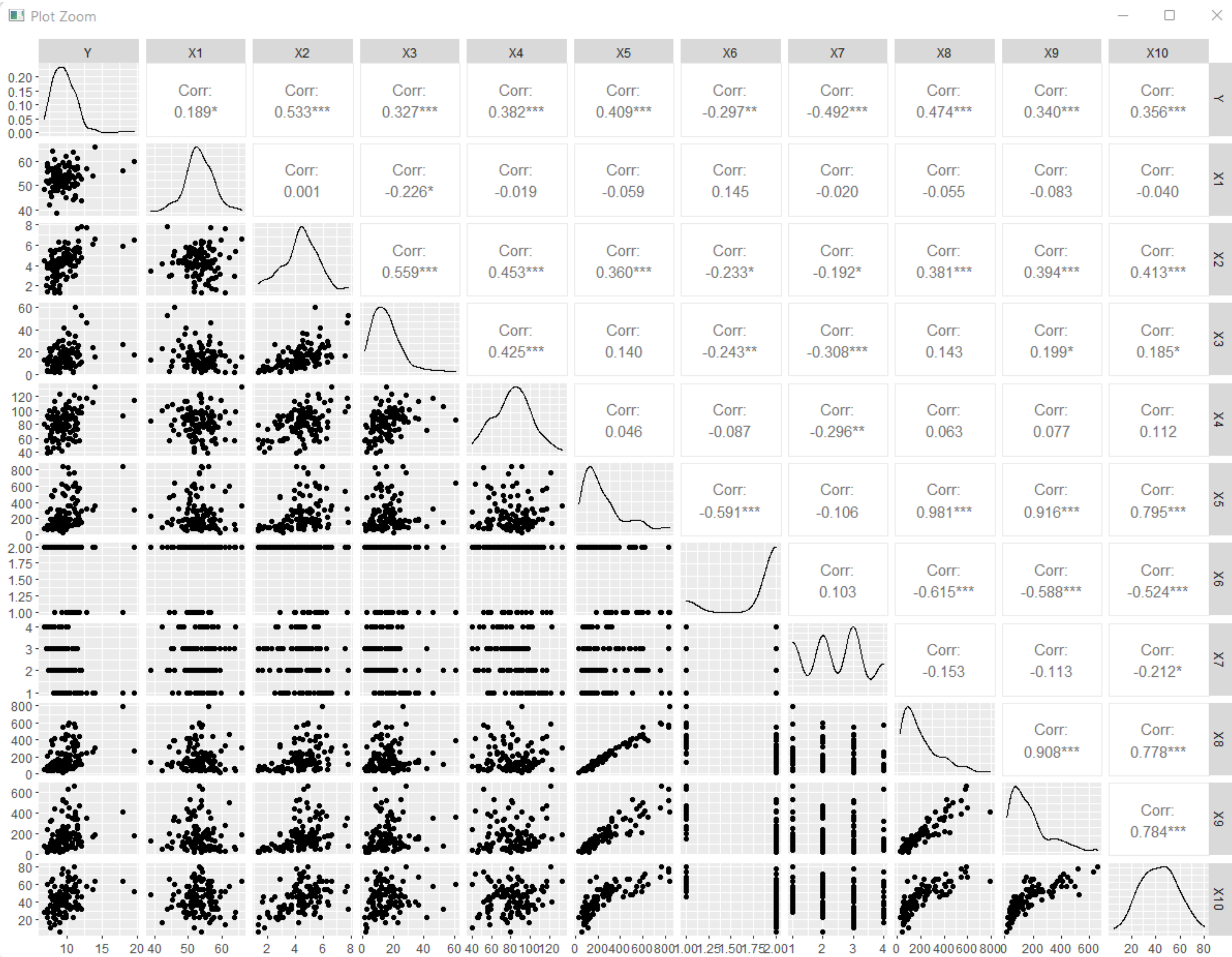
```r
# Creating the correlation matrix
cor_df <- cor(df)
cor_df
```

```
> cor_df
             Y           X1           X2         X3          X4          X5          X6          X7          X8          X9         X10
Y    1.0000000  0.188913972  0.533443831  0.3266838  0.38248193  0.40926525 -0.29695100 -0.49213043  0.47388550  0.34036706  0.35553792
X1   0.1889140  1.000000000  0.001093166 -0.2258468 -0.01885490 -0.05882316  0.14512637 -0.02043194 -0.05477467 -0.08294462 -0.04045138
X2   0.5334438  0.001093166  1.000000000  0.5591589  0.45339156  0.35977000 -0.23302990 -0.19228070  0.38141108  0.39398134  0.41260068
X3   0.3266838 -0.225846789  0.559158869  1.0000000  0.42496204  0.13972495 -0.24274409 -0.30827778  0.14294821  0.19889983  0.18513114
X4   0.3824819 -0.018854897  0.453391557  0.4249620  1.00000000  0.04581997 -0.08669664 -0.29634411  0.06291352  0.07738133  0.11192761
X5   0.4092652 -0.058823160  0.359770000  0.1397249  0.04581997  1.00000000 -0.59117997 -0.10562663  0.98099774  0.91550415  0.79452438
X6  -0.2969510  0.145126369 -0.233029901 -0.2427441 -0.08669664 -0.59117997  1.00000000  0.10266758 -0.61475733 -0.58823974 -0.52439032
X7  -0.4921304 -0.020431944 -0.192280702 -0.3082778 -0.29634411 -0.10562663  0.10266758  1.00000000 -0.15274400 -0.11268137 -0.21153192
X8   0.4738855 -0.054774667  0.381411081  0.1429482  0.06291352  0.98099774 -0.61475733 -0.15274400  1.00000000  0.90789698  0.77806330
X9   0.3403671 -0.082944616  0.393981340  0.1988998  0.07738133  0.91550415 -0.58823974 -0.11268137  0.90789698  1.00000000  0.78350550
X10  0.3555379 -0.040451379  0.412600675  0.1851311  0.11192761  0.79452438 -0.52439032 -0.21153192  0.77806330  0.78350550  1.00000000
```

1.4.4 Visualizing the correlation matrix i.e., getting a scatter plot matrix.

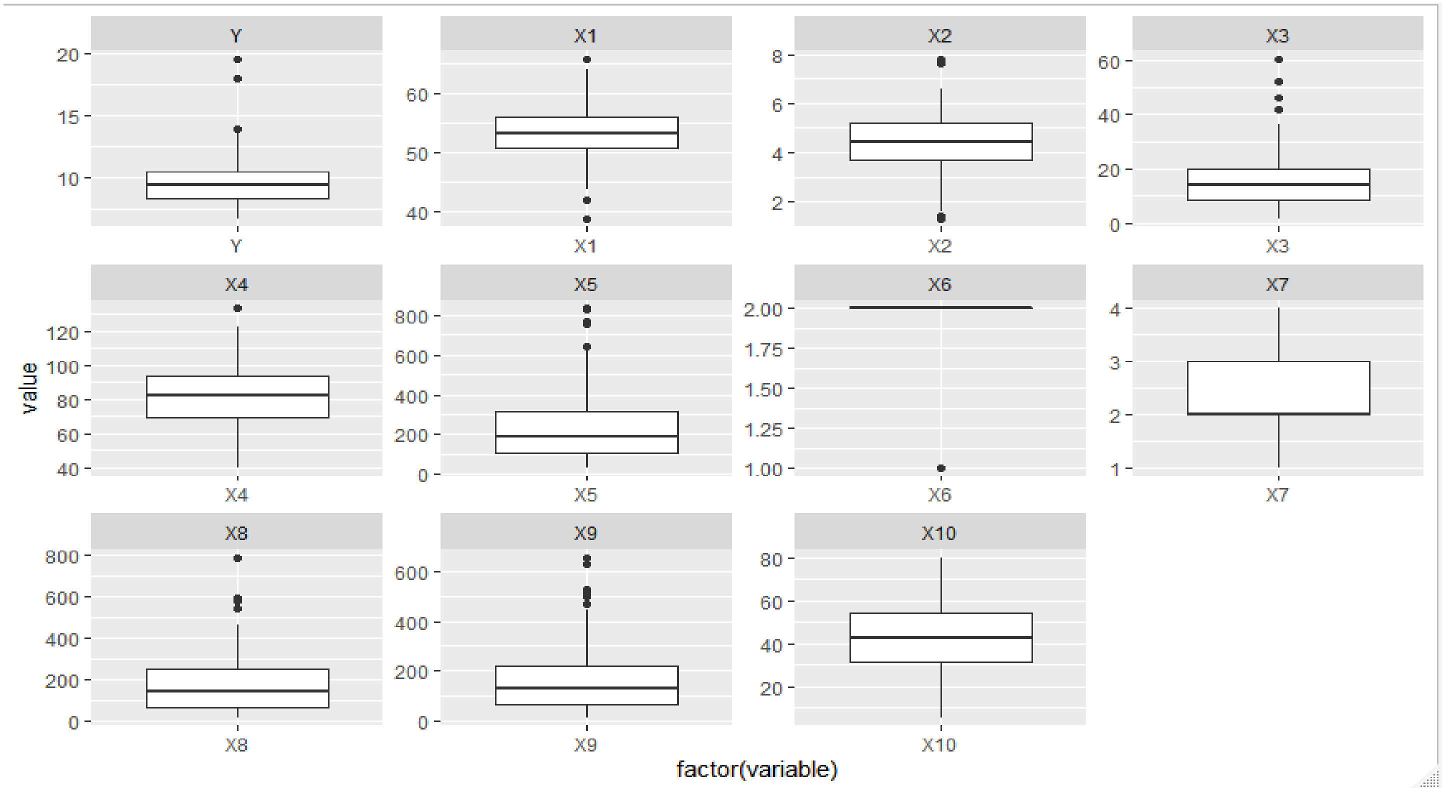
```r
library(GGally)
#generate the pairs plot
ggpairs(df)
# Visualizing the correlation matrix i.e., getting a scatter plot matrix
pairs(df)
```





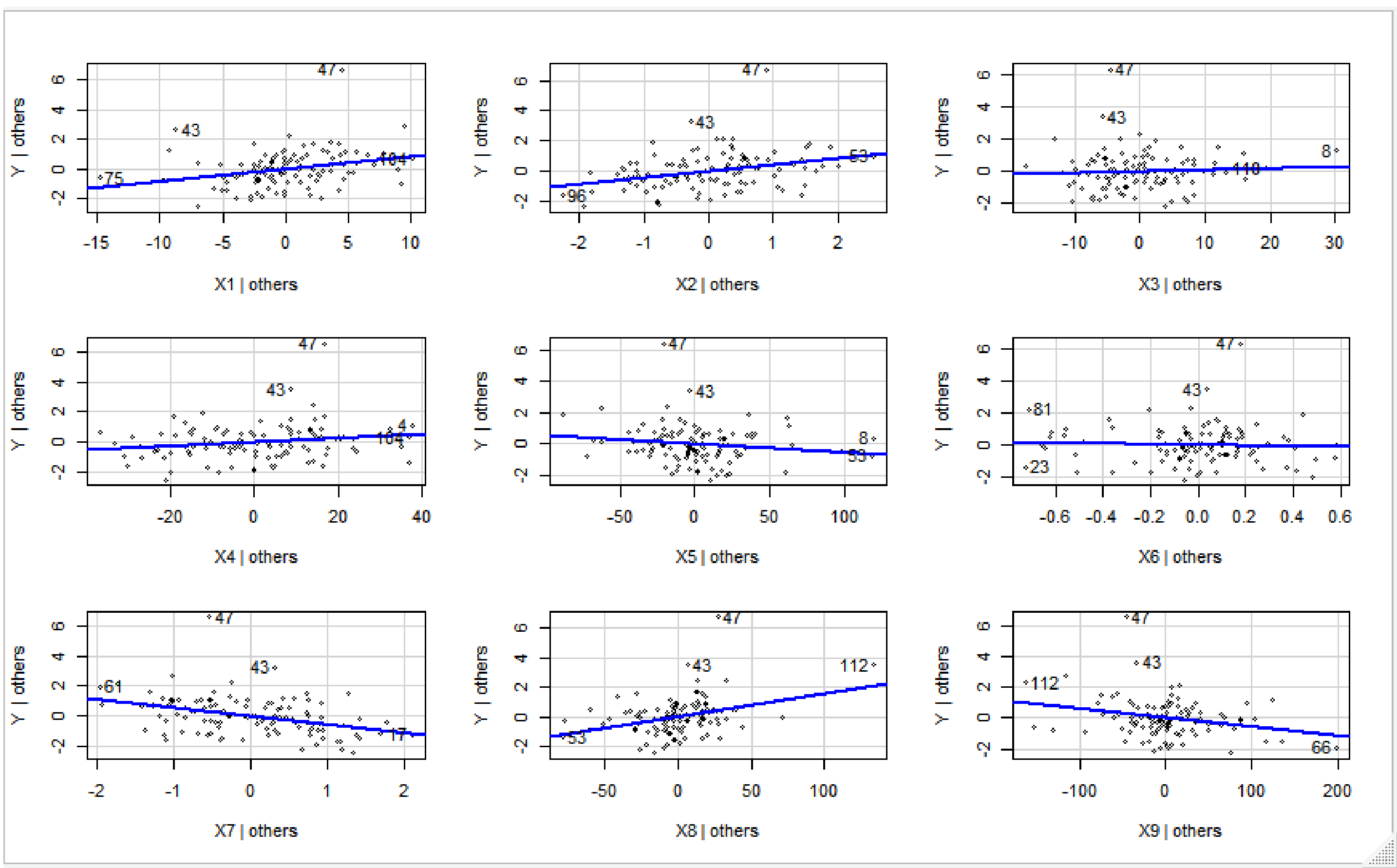
1.4.5 Using box plots to look for outliers.

```
# Getting the box plots to look for the outliers
library(reshape)
senicData <- melt(df)
boxplot <- ggplot(senicData, aes(factor(variable), value))
boxplot + geom_boxplot() + facet_wrap(~variable, scale="free")
```

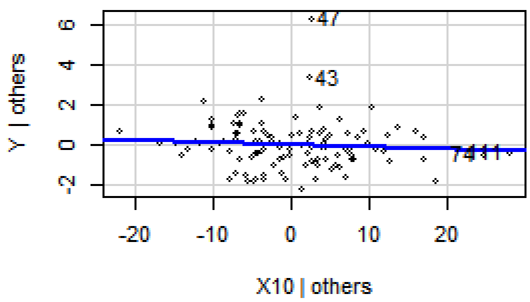


1.4.6 Getting the added variable plot.

```
# Added-Variable Plots
install.packages('car')
library(car)
# Fitting the model
lmfit <- lm(Y ~ . , data = df)
avPlots(lmfit)
```



Added-Variable Plots



1.4.7 Inference of the exploratory data analysis:

From the histogram plots, except for a few outliers' values in the response variable, the response variable is approximately following the normal distribution. Among the predictors, X1, X2, X4, X10 follow the approximately normal distributions. Two of the predictors are categorical and hence can't follow the normal distribution. X3, X5, X8, X9 distribution plots are skewed and show the large no of right-tailed outliers. Summary statistics show the presence of outliers in the X3, X5, X8, and X9. From the correlation matrix, we can easily observe the presence of high multiple collinearities among the multiple variables. From the AV plot, we can find that the partial correlation of Y with all the variables excluding, X3, X5, X6, X9 is almost linearly dependent and is constant for the other cases.

1.4.8 Either standardization or centralization of the variables needed?

At this point of the MLR, it doesn't look like, if the standardization or the centralization of the variables are needed. There are some outliers in some of the variables, if we fix them, there won't be any need for standardization.

2. Model/Methods

Model selection for the multiple linear regression involves the below steps.

- Model fitting based on all the predictors.
- Multicollinearity checks.
- Multicollinearity Diagnostics.
- Criteria for Model Selection (Variable Selection):
    1. Adjusted R Square
    2. Mallows' Cp
    3. AIC and BIC
- Sequential variable selection using stepwise regression

2.1 Model Fitting Based on All the Predictors

```
################## Fit a multiple linear regression model ##################

lmfit.full <- lm(Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10, data=df)
summary(lmfit.full)
anova(lmfit.full)
```

```
> lmfit.full <- lm(Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10, data=df)
> summary(lmfit.full)

Call:
lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 +
    X10, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-2.2346 -0.6592 -0.0699  0.6304  6.3389

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.720403   1.888078   1.970 0.051495 .
X1           0.085177   0.027282   3.122 0.002337 **
X2           0.426433   0.124402   3.428 0.000879 ***
X3           0.007916   0.015634   0.506 0.613704
X4           0.012513   0.007092   1.764 0.080670 .
X5          -0.005403   0.003513  -1.538 0.127110
X6          -0.204155   0.430168  -0.475 0.636091
X7          -0.580146   0.132088  -4.392 2.75e-05 ***
X8           0.015991   0.004282   3.734 0.000311 ***
X9          -0.005853   0.002180  -2.685 0.008463 **
X10         -0.012627   0.013594  -0.929 0.355161
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.223 on 102 degrees of freedom
Multiple R-squared:  0.6273,    Adjusted R-squared:  0.5907
F-statistic: 17.16 on 10 and 102 DF,  p-value: < 2.2e-16

> anova(lmfit.full)
Analysis of Variance Table

Response: Y
           Df  Sum Sq Mean Sq F value    Pr(>F)
X1          1  14.604  14.604  9.7660 0.0023154 **
X2          1 116.356 116.356 77.8089 3.284e-14 ***
X3          1   3.248   3.248  2.1720 0.1436244
X4          1   8.606   8.606  5.7549 0.0182590 *
X5          1  31.087  31.087 20.7886 1.430e-05 ***
X6          1   1.514   1.514  1.0124 0.3167176
X7          1  46.675  46.675 31.2122 1.931e-07 ***
X8          1  20.324  20.324 13.5910 0.0003663 ***
X9          1  12.975  12.975  8.6765 0.0039937 **
X10         1   1.290   1.290  0.8628 0.3551614
Residuals 102 152.531   1.495
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

2.2 Multicollinearity checks.

```
summary(lmfit.full)
cor(df)    ###### The signs of some estimates from the output and the correlation matrix are different.
anova(lmfit.full)
```

```
> summary(lmfit.full)

Call:
lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + 
    X10, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.2346 -0.6592 -0.0699  0.6304  6.3389 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.720403   1.888078   1.970 0.051495 .  
X1           0.085177   0.027282   3.122 0.002337 ** 
X2           0.426433   0.124402   3.428 0.000879 ***
X3           0.007916   0.015634   0.506 0.613704    
X4           0.012513   0.007092   1.764 0.080670 .  
X5          -0.005403   0.003513  -1.538 0.127110    
X6          -0.204155   0.430168  -0.475 0.636091    
X7          -0.580146   0.132088  -4.392 2.75e-05 ***
X8           0.015991   0.004282   3.734 0.000311 ***
X9          -0.005853   0.002180  -2.685 0.008463 ** 
X10         -0.012627   0.013594  -0.929 0.355161    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.223 on 102 degrees of freedom
Multiple R-squared:  0.6273,	Adjusted R-squared:  0.5907 
F-statistic: 17.16 on 10 and 102 DF,  p-value: < 2.2e-16

> cor(df)   ###### The signs of some estimates from the output and the correlation matrix are different.
             Y           X1           X2         X3          X4          X5          X6          X7          X8          X9         X10
Y    1.0000000  0.188913972  0.533443831  0.3266838  0.38248193  0.40926525 -0.29695100 -0.49213043  0.47388550  0.34036706  0.35553792
X1   0.1889140  1.000000000  0.001093166 -0.2258468 -0.01885490 -0.05882316  0.14512637 -0.02043194 -0.05477467 -0.08294462 -0.04045138
X2   0.5334438  0.001093166  1.000000000  0.5591589  0.45339156  0.35977000 -0.23302990 -0.19228070  0.38141108  0.39398134  0.41260068
X3   0.3266838 -0.225846789  0.559158869  1.0000000  0.42496204  0.13972495 -0.24274409 -0.30827778  0.14294821  0.19889983  0.18513114
X4   0.3824819 -0.018854897  0.453391557  0.4249620  1.00000000  0.04581997 -0.08669664 -0.29634411  0.06291352  0.07738133  0.11192761
X5   0.4092652 -0.058823160  0.359770000  0.1397249  0.04581997  1.00000000 -0.59117997 -0.10562663  0.98099774  0.91550415  0.79452438
X6  -0.2969510  0.145126369 -0.233029901 -0.2427441 -0.08669664 -0.59117997  1.00000000  0.10266758 -0.61475733 -0.58823974 -0.52439032
X7  -0.4921304 -0.020431944 -0.192280702 -0.3082778 -0.29634411 -0.10562663  0.10266758  1.00000000 -0.15274400 -0.11268137 -0.21153192
X8   0.4738855 -0.054774667  0.381411081  0.1429482  0.06291352  0.98099774 -0.61475733 -0.15274400  1.00000000  0.90789698  0.77806330
X9   0.3403671 -0.082944616  0.393981340  0.1988998  0.07738133  0.91550415 -0.58823974 -0.11268137  0.90789698  1.00000000  0.78350550
X10  0.3555379 -0.040451379  0.412600675  0.1851311  0.11192761  0.79452438 -0.52439032 -0.21153192  0.77806330  0.78350550  1.00000000
> anova(lmfit.full)
Analysis of Variance Table

Response: Y
           Df  Sum Sq Mean Sq F value    Pr(>F)    
X1          1  14.604  14.604  9.7660 0.0023154 ** 
X2          1 116.356 116.356 77.8089 3.284e-14 ***
X3          1   3.248   3.248  2.1720 0.1436244    
X4          1   8.606   8.606  5.7549 0.0182590 *  
X5          1  31.087  31.087 20.7886 1.430e-05 ***
X6          1   1.514   1.514  1.0124 0.3167176    
X7          1  46.675  46.675 31.2122 1.931e-07 ***
X8          1  20.324  20.324 13.5910 0.0003663 ***
X9          1  12.975  12.975  8.6765 0.0039937 ** 
X10         1   1.290   1.290  0.8628 0.3551614    
Residuals 102 152.531   1.495                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> 
```

It is evident from the highlighted terms in the screenshot above that multicollinearity exists and we need to remove them, to check further we will conduct the Variation Inflation factor test to get the predictors having high VIF.

2.3 Multicollinearity Diagnostics.

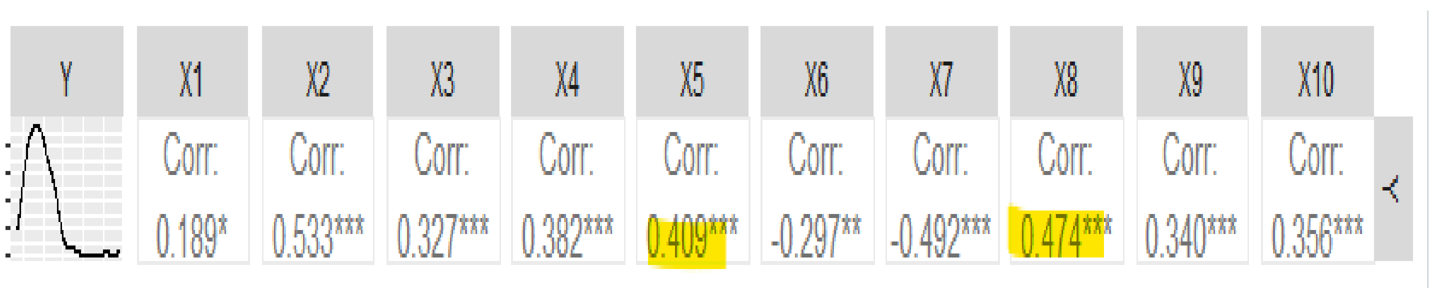
```
# Check Variance Inflation Factor. =

vif(lmfit.full)
```

```
> vif(lmfit.full)
       X1        X2        X3        X4        X5        X6        X7        X8        X9       X10 
 1.109711  2.084059  1.917497  1.412452 34.370407  1.787140  1.331506 32.474034  6.901296  3.198243 
> 
```

As evident from the screenshot above, there is serious multicollinearity with X5 and X8 as the VIF factor is very high, hence we need to diagnose them.

```r
lmfit1 <- lm(Y ~ X1 + X2 + X3 + X4 +  X6 + X7 + X8 + X9 + X10, data=df)
lmfit2 <- lm(Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 +  X9 + X10, data=df)
summary(lmfit1)
summary(lmfit2)
# Comparison between the full model and the reduced models
anova(lmfit1, lmfit.full)
anova(lmfit2, lmfit.full)
# VIF on the reduced models
vif(lmfit1)
vif(lmfit2)
```

```
> lmfit1 <- lm(Y ~ X1 + X2 + X3 + X4 +  X6 + X7 + X8 + X9 + X10, data=df)
> lmfit2 <- lm(Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 +  X9 + X10, data=df)
> summary(lmfit1)

Call:
lm(formula = Y ~ X1 + X2 + X3 + X4 + X6 + X7 + X8 + X9 + X10, 
    data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.2884 -0.6939 -0.0302  0.5947  6.4503 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.126291   1.881902   2.193 0.030585 *  
X1           0.084422   0.027458   3.075 0.002699 ** 
X2           0.460304   0.123246   3.735 0.000308 ***
X3           0.004818   0.015606   0.309 0.758129    
X4           0.012520   0.007139   1.754 0.082431 .  
X6          -0.310418   0.427390  -0.726 0.469295    
X7          -0.634866   0.128048  -4.958 2.81e-06 ***
X8           0.010127   0.001964   5.155 1.23e-06 ***
X9          -0.006632   0.002134  -3.108 0.002438 ** 
X10         -0.018294   0.013172  -1.389 0.167887    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.231 on 103 degrees of freedom
Multiple R-squared:  0.6186,	Adjusted R-squared:  0.5853 
F-statistic: 18.56 on 9 and 103 DF,  p-value: < 2.2e-16
```



```
> summary(lmfit2)

Call:
lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X9 + X10,
    data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4891 -0.7461 -0.1063  0.4994  6.7736

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.988881   1.970502   2.532 0.012859 *
X1           0.084033   0.028944   2.903 0.004518 **
X2           0.527293   0.128838   4.093 8.49e-05 ***
X3          -0.003855   0.016246  -0.237 0.812919
X4           0.011886   0.007522   1.580 0.117157
X5           0.006272   0.001698   3.693 0.000357 ***
X6          -0.618274   0.440966  -1.402 0.163894
X7          -0.726876   0.133796  -5.433 3.73e-07 ***
X9          -0.004891   0.002296  -2.130 0.035565 *
X10         -0.020685   0.014240  -1.453 0.149378
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.297 on 103 degrees of freedom
Multiple R-squared:  0.5763,	Adjusted R-squared:  0.5393
F-statistic: 15.57 on 9 and 103 DF,  p-value: 9.28e-16

> # Comparison between the full model and the reduced models
> anova(lmfit1, lmfit.full)
Analysis of Variance Table

Model 1: Y ~ X1 + X2 + X3 + X4 + X6 + X7 + X8 + X9 + X10
Model 2: Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    103 156.07
2    102 152.53  1     3.538 2.3659 0.1271
> anova(lmfit2, lmfit.full)
Analysis of Variance Table

Model 1: Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X9 + X10
Model 2: Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    103 173.38
2    102 152.53  1     20.85 13.943 0.0003106 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> # VIF on the reduced models
> vif(lmfit1)
      X1       X2       X3       X4       X6       X7       X8       X9      X10
1.109352 2.018760 1.885678 1.412451 1.741045 1.234919 6.744121 6.528684 2.963372
> vif(lmfit2)
      X1       X2       X3       X4       X5       X6       X7       X9      X10
1.109571 1.985806 1.839535 1.411661 7.137955 1.668347 1.213660 6.804914 3.117645
>
```

VIF for normalized data:

```
> vif(lmfit_norm)
       X1        X2        X3        X4        X5        X6        X7        X8        X9       X10
 1.109711  2.084059  1.917497  1.412452 34.370407  1.787140  1.331506 32.474034  6.901296  3.198243
>
```

As evident from the screenshots above and the highlighted terms that there was serious multicollinearity in the model because of the X5 and X8 being both in the model. By removing any one of these from the model we can see that the VIF is not high for any of the predictors. We also tried to scale the data first and then tried to get the VIF, there was not much change and the VIF was still coming high for both the variables. Hence to reduce the impact of multicollinearity, we need to remove either X5 or X8 from the model. Since there is a higher correlation between the output variable Y and X8, I am removing X5 from the model to mitigate the impact of multicollinearity. Also, with X8 the adjusted r squared obtained is almost the same as the full model.

2.4 Criteria For Model Selection and Sequential Variable Selection :

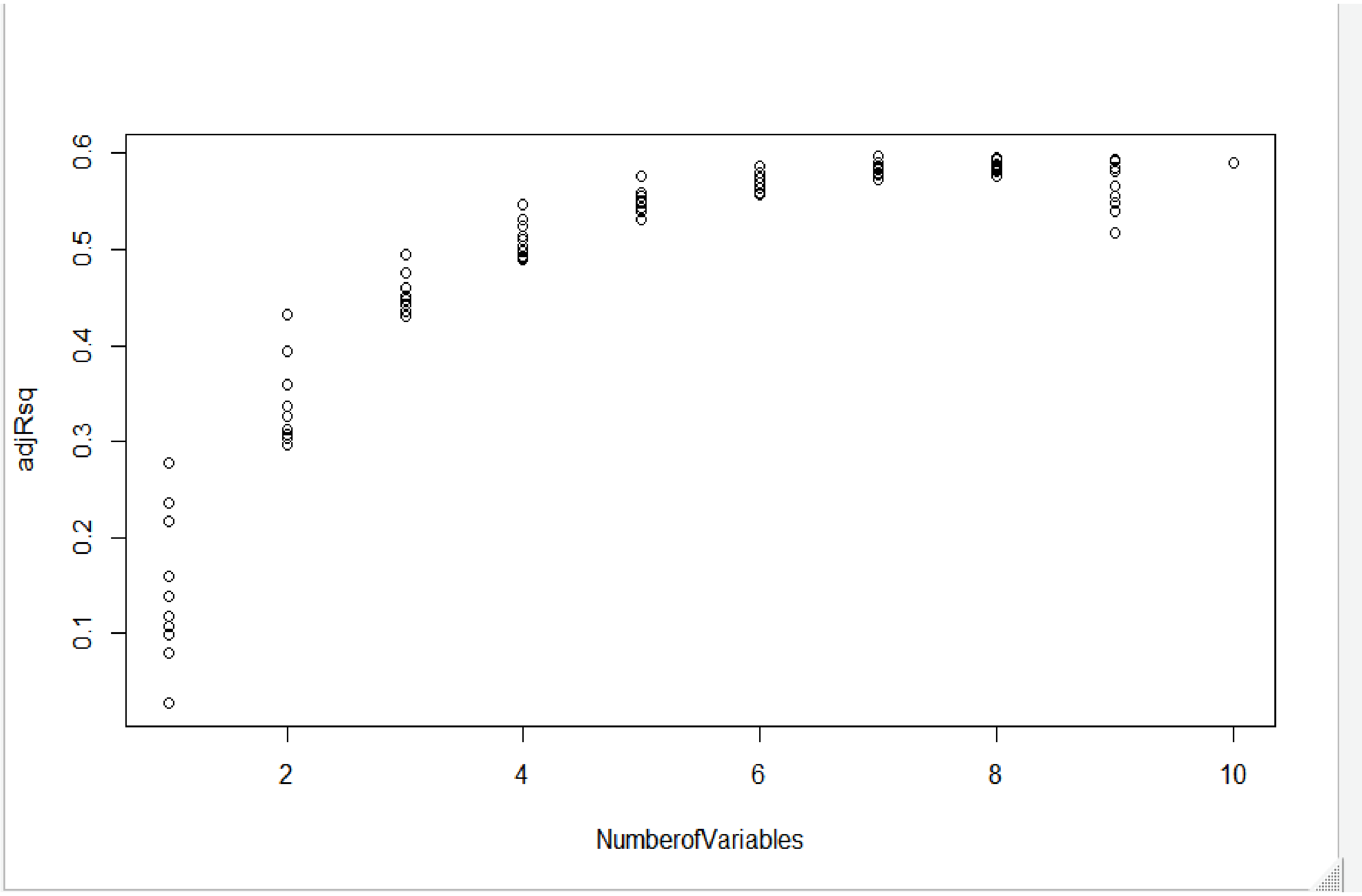
We are going to use the stepwise regression for model selection after considering all the criteria for the model selection.

2.4.1 Adjusted R Square

```
#################### Model Selection (Step wise Regression) #####################

#### Install packages for the model selection ####
install.packages("leaps")
install.packages("HH")
install.packages("StepReg")
#### Load HH, leaps, and StepReg packages ####
library(leaps)
library(HH)
library(StepReg)
par(mfrow=c(1,1))
#### Stepwise Regression ####
#### Adjusted R2 ####
b = bestsubset(data=df,y="Y",select="adjRsq",best=10)
print(b)
stepwise(data=df,y="Y",select="adjRsq")
plot(b[,1:2])
```

```
> b = bestsubset(data=df,y="Y",select="adjRsq",best=10)
> print(b)
   NumberofVariables  adjRsq       VariablesIncludedinModel
1                  1 0.27812                             x2
2                  1 0.23537                             x7
3                  1 0.21758                             x8
4                  1 0.16000                             x5
5                  1 0.13860                             x4
6                  1 0.11854                            x10
7                  1 0.10788                             x9
8                  1 0.09867                             x3
9                  1 0.07997                             x6
10                 1 0.02700                             x1
11                 2 0.43200                          x2 x7
12                 2 0.39415                          x7 x8
13                 2 0.35985                          x5 x7
14                 2 0.35869                          x2 x8
15                 2 0.33761                          x4 x8
16                 2 0.32680                          x2 x5
17                 2 0.31213                          x7 x9
18                 2 0.30767                          x1 x2
19                 2 0.30364                          x2 x6
20                 2 0.29690                          x2 x4
21                 3 0.49560                       x2 x7 x8
22                 3 0.47532                       x2 x5 x7
23                 3 0.46015                       x1 x2 x7
24                 3 0.45108                       x2 x6 x7
25                 3 0.44791                       x4 x7 x8
26                 3 0.44298                       x2 x7 x9
27                 3 0.43530                      x2 x7 x10
28                 3 0.43123                       x2 x3 x7
29                 3 0.43110                       x2 x4 x7
30                 3 0.43088                       x1 x7 x8
31                 4 0.54746                    x2 x7 x8 x9
32                 4 0.53099                    x1 x2 x7 x8
33                 4 0.52400                    x2 x5 x7 x8
34                 4 0.51315                   x2 x7 x8 x10
35                 4 0.50972                    x1 x2 x5 x7
36                 4 0.50290                    x2 x4 x7 x8
37                 4 0.49816                    x2 x5 x7 x9
38                 4 0.49225                    x2 x3 x7 x8
39                 4 0.49095                    x2 x6 x7 x8
40                 4 0.48991                    x1 x2 x6 x7
41                 5 0.57647                 x1 x2 x7 x8 x9
42                 5 0.55861                 x1 x2 x5 x7 x8
43                 5 0.55632                 x2 x5 x7 x8 x9
44                 5 0.55591                 x2 x4 x7 x8 x9
45                 5 0.55033                x2 x7 x8 x9 x10
46                 5 0.54876                x1 x2 x7 x8 x10
47                 5 0.54391                 x2 x6 x7 x8 x9
48                 5 0.54323                 x2 x3 x7 x8 x9
49                 5 0.54044                 x1 x2 x4 x7 x8
50                 5 0.53188                 x2 x4 x5 x7 x8
51                 6 0.58687              x1 x2 x4 x7 x8 x9
52                 6 0.58595              x1 x2 x5 x7 x8 x9
53                 6 0.58027             x1 x2 x7 x8 x9 x10
54                 6 0.57535              x1 x2 x3 x7 x8 x9
55                 6 0.57511              x1 x2 x6 x7 x8 x9
56                 6 0.56863              x1 x2 x4 x5 x7 x8
57                 6 0.56500              x2 x4 x5 x7 x8 x9
58                 6 0.56398             x1 x2 x5 x7 x8 x10
59                 6 0.55836             x2 x4 x7 x8 x9 x10
60                 6 0.55764             x1 x2 x4 x7 x8 x10
61                 7 0.59662           x1 x2 x4 x5 x7 x8 x9
62                 7 0.59024          x1 x2 x4 x7 x8 x9 x10
63                 7 0.58600           x1 x2 x3 x5 x7 x8 x9
64                 7 0.58580          x1 x2 x5 x7 x8 x9 x10
65                 7 0.58487           x1 x2 x4 x6 x7 x8 x9
66                 7 0.58413           x1 x2 x3 x4 x7 x8 x9
67                 7 0.58351           x1 x2 x5 x6 x7 x8 x9
68                 7 0.57968          x1 x2 x6 x7 x8 x9 x10
69                 7 0.57860          x1 x2 x3 x7 x8 x9 x10
70                 7 0.57354          x1 x2 x4 x5 x7 x8 x10
71                 8 0.59615      x1 x2 x4 x5 x7 x8 x9 x10
72                 8 0.59468       x1 x2 x3 x4 x5 x7 x8 x9
73                 8 0.59371       x1 x2 x4 x5 x6 x7 x8 x9
74                 8 0.58889      x1 x2 x4 x6 x7 x8 x9 x10
75                 8 0.58717      x1 x2 x3 x4 x7 x8 x9 x10
76                 8 0.58522      x1 x2 x3 x5 x7 x8 x9 x10
77                 8 0.58391      x1 x2 x5 x6 x7 x8 x9 x10
78                 8 0.58269       x1 x2 x3 x5 x6 x7 x8 x9
```

```
> stepwise(data=df,y="Y",select="adjRsq")
$process
  Step EffectEntered EffectRemoved EffectNumber    Select
1    0     intercept                          1 0.0000000
2    1            X2                          2 0.2781169
3    2            X7                          3 0.4320022
4    3            X8                          4 0.4956036
5    4            X9                          5 0.5474590
6    5            X1                          6 0.5764681
7    6            X4                          7 0.5868718
8    7            X5                          8 0.5966238

$variate
[1] "intercept" "X2"        "X7"        "X8"        "X9"        "X1"        "X4"        "X5"

> plot(b[,1:2])
```

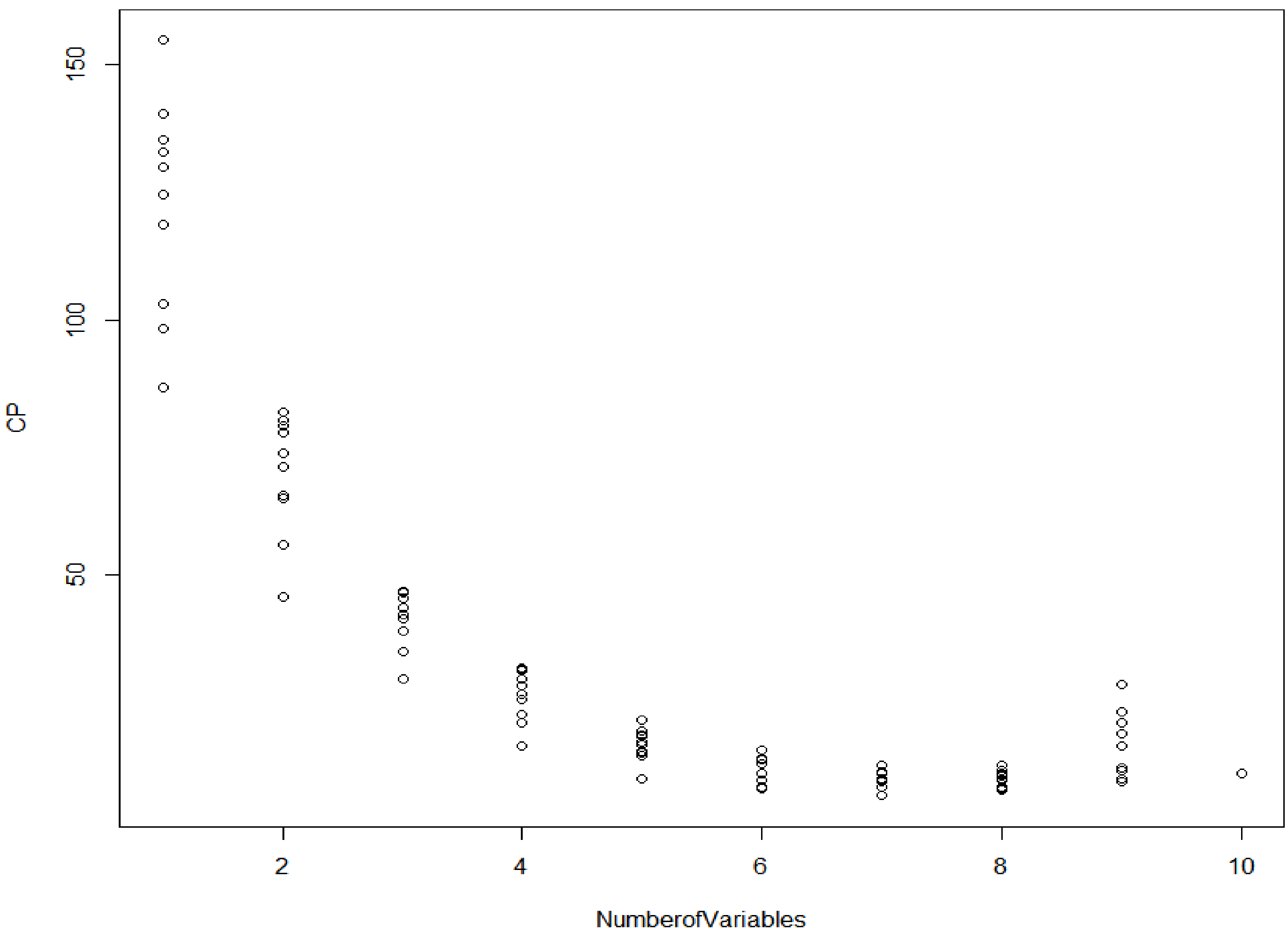


2.4.2 Mallows' *Cp*

```
#### Cp ####
b = bestsubset(data=df,y="Y",select="CP",best=10)
print(b)
stepwise(data=df,y="Y",select="CP")
plot(b[,1:2])
```

21	3	29.32849	x2 x7 x8
22	3	34.73109	x2 x5 x7
23	3	38.76952	x1 x2 x7
24	3	41.18599	x2 x6 x7
25	3	42.03126	x4 x7 x8
26	3	43.34353	x2 x7 x9
27	3	45.38924	x2 x7 x10
28	3	46.47208	x2 x3 x7
29	3	46.50805	x2 x4 x7
30	3	46.56482	x1 x7 x8
31	4	16.41293	x2 x7 x8 x9
32	4	20.75742	x1 x2 x7 x8
33	4	22.60200	x2 x5 x7 x8
34	4	25.46613	x2 x7 x8 x10
35	4	26.37083	x1 x2 x5 x7
36	4	28.17073	x2 x4 x7 x8
37	4	29.42031	x2 x5 x7 x9
38	4	30.98194	x2 x3 x7 x8
39	4	31.32505	x2 x6 x7 x8
40	4	31.59899	x1 x2 x6 x7
41	5	9.72342	x1 x2 x7 x8 x9
42	5	14.39210	x1 x2 x5 x7 x8
43	5	14.99142	x2 x5 x7 x8 x9
44	5	15.09870	x2 x4 x7 x8 x9
45	5	16.55682	x2 x7 x8 x9 x10
46	5	16.96680	x1 x2 x7 x8 x10
47	5	18.23598	x2 x6 x7 x8 x9
48	5	18.41225	x2 x3 x7 x8 x9
49	5	19.14187	x1 x2 x4 x7 x8
50	5	21.38081	x2 x4 x5 x7 x8
51	6	7.99423	x1 x2 x4 x7 x8 x9
52	6	8.23315	x1 x2 x5 x7 x8 x9
53	6	9.70416	x1 x2 x7 x8 x9 x10
54	6	10.97781	x1 x2 x3 x7 x8 x9
55	6	11.03909	x1 x2 x6 x7 x8 x9
56	6	12.71813	x1 x2 x4 x5 x7 x8
57	6	13.65925	x2 x4 x5 x7 x8 x9
58	6	13.92238	x1 x2 x5 x7 x8 x10
59	6	15.37798	x2 x4 x7 x8 x9 x10
60	6	15.56414	x1 x2 x4 x7 x8 x10
61	7	6.48305	x1 x2 x4 x5 x7 x8 x9
62	7	8.12107	x1 x2 x4 x7 x8 x9 x10
63	7	9.20849	x1 x2 x3 x5 x7 x8 x9
64	7	9.26058	x1 x2 x5 x7 x8 x9 x10
65	7	9.49748	x1 x2 x4 x6 x7 x8 x9
66	7	9.68912	x1 x2 x3 x4 x7 x8 x9
67	7	9.84756	x1 x2 x5 x6 x7 x8 x9
68	7	10.82914	x1 x2 x6 x7 x8 x9 x10
69	7	11.10631	x1 x2 x3 x7 x8 x9 x10
70	7	12.40608	x1 x2 x4 x5 x7 x8 x10
71	8	7.61745	x1 x2 x4 x5 x7 x8 x9 x10
72	8	7.99054	x1 x2 x3 x4 x5 x7 x8 x9
73	8	8.23669	x1 x2 x4 x5 x6 x7 x8 x9
74	8	9.46249	x1 x2 x4 x6 x7 x8 x9 x10
75	8	9.90041	x1 x2 x3 x4 x7 x8 x9 x10
76	8	10.39594	x1 x2 x3 x5 x7 x8 x9 x10
77	8	10.72764	x1 x2 x5 x6 x7 x8 x9 x10
78	8	11.03915	x1 x2 x3 x5 x6 x7 x8 x9
79	8	11.32025	x1 x2 x3 x4 x6 x7 x8 x9
80	8	12.48267	x1 x2 x3 x6 x7 x8 x9 x10
81	9	9.22524	x1 x2 x3 x4 x5 x7 x8 x9 x10
82	9	9.25639	x1 x2 x4 x5 x6 x7 x8 x9 x10
83	9	9.86276	x1 x2 x3 x4 x5 x6 x7 x8 x9
84	9	11.36589	x1 x2 x3 x4 x6 x7 x8 x9 x10
85	9	12.11286	x1 x2 x3 x5 x6 x7 x8 x9 x10
86	9	16.21004	x1 x2 x3 x4 x5 x6 x7 x8 x10
87	9	18.74721	x2 x3 x4 x5 x6 x7 x8 x9 x10

```
> stepwise(data=df,y="Y",select="CP")
$process
  Step EffectEntered EffectRemoved EffectNumber      Select
1    0     intercept                          1 162.645091
2    1            X2                          2  86.776009
3    2            X7                          3  45.654273
4    3            X8                          4  29.328487
5    4            X9                          5  16.412934
6    5            X1                          6   9.723423
7    6            X4                          7   7.994230
8    7            X5                          8   6.483045

$variate
[1] "intercept" "X2"        "X7"        "X8"        "X9"        "X1"        "X4"        "X5"

> plot(b[,1:2])
> 
```

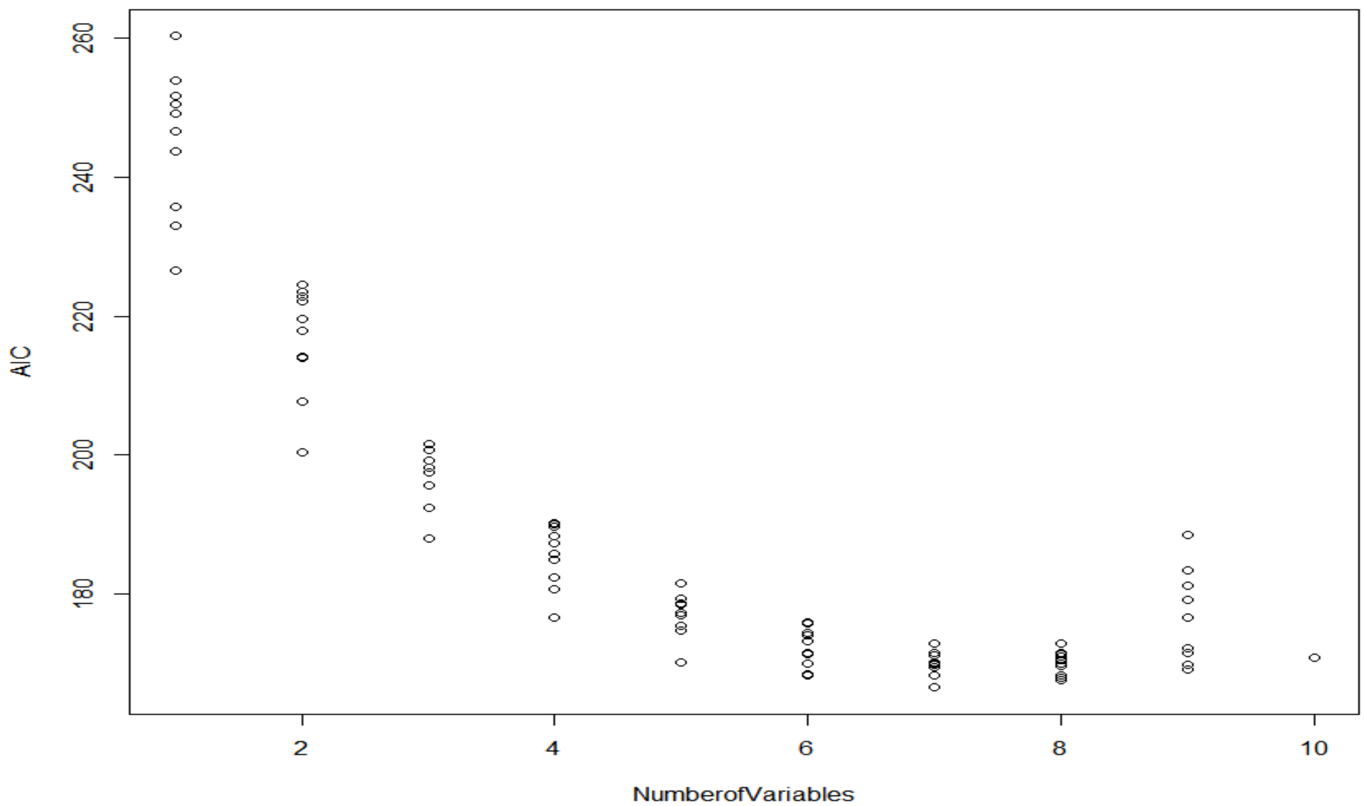


2.4.3 AIC and BIC

```
#### AIC ####
b = bestsubset(data=df,y="Y",select="AIC",best=10)
print(b)
stepwise(data=df,y="Y",select="AIC")
plot(b[,1:2])
#### BIC ####
b = bestsubset(data=df,y="Y",select="BIC",best=10)
print(b)
stepwise(data=df,y="Y",select="BIC")
plot(b[,1:2])
```

AIC Table

12	2	207.7510	x7 x8
13	2	213.9726	x5 x7
14	2	214.1778	x2 x8
15	2	217.8325	x4 x8
16	2	219.6607	x2 x5
17	2	222.0977	x7 x9
18	2	222.8281	x1 x2
19	2	223.4829	x2 x6
20	2	224.5723	x2 x4
21	3	188.0086	x2 x7 x8
22	3	192.4644	x2 x5 x7
23	3	195.6840	x1 x2 x7
24	3	197.5675	x2 x6 x7
25	3	198.2190	x4 x7 x8
26	3	199.2231	x2 x7 x9
27	3	200.7707	x2 x7 x10
28	3	201.5814	x2 x3 x7
29	3	201.6083	x2 x4 x7
30	3	201.6506	x1 x7 x8
31	4	176.7085	x2 x7 x8 x9
32	4	180.7466	x1 x2 x7 x8
33	4	182.4184	x2 x5 x7 x8
34	4	184.9663	x2 x7 x8 x10
35	4	185.7593	x1 x2 x5 x7
36	4	187.3206	x2 x4 x7 x8
37	4	188.3919	x2 x5 x7 x9
38	4	189.7167	x2 x3 x7 x8
39	4	190.0058	x2 x6 x7 x8
40	4	190.2360	x1 x2 x6 x7
41	5	170.1711	x1 x2 x7 x8 x9
42	5	174.8380	x1 x2 x5 x7 x8
43	5	175.4234	x2 x5 x7 x8 x9
44	5	175.5279	x2 x4 x7 x8 x9
45	5	176.9383	x2 x7 x8 x9 x10
46	5	177.3316	x1 x2 x7 x8 x10
47	5	178.5409	x2 x6 x7 x8 x9
48	5	178.7078	x2 x3 x7 x8 x9
49	5	179.3962	x1 x2 x4 x7 x8
50	5	181.4826	x2 x4 x5 x7 x8
51	6	168.2997	x1 x2 x4 x7 x8 x9
52	6	168.5517	x1 x2 x5 x7 x8 x9
53	6	170.0913	x1 x2 x7 x8 x9 x10
54	6	171.4076	x1 x2 x3 x7 x8 x9
55	6	171.4705	x1 x2 x6 x7 x8 x9
56	6	173.1817	x1 x2 x4 x5 x7 x8
57	6	174.1296	x2 x4 x5 x7 x8 x9
58	6	174.3933	x1 x2 x5 x7 x8 x10
59	6	175.8406	x2 x4 x7 x8 x9 x10
60	6	176.0243	x1 x2 x4 x7 x8 x10
61	7	166.5292	x1 x2 x4 x5 x7 x8 x9
62	7	168.3038	x1 x2 x4 x7 x8 x9 x10
63	7	169.4667	x1 x2 x3 x5 x7 x8 x9
64	7	169.5222	x1 x2 x5 x7 x8 x9 x10
65	7	169.7738	x1 x2 x4 x6 x7 x8 x9
66	7	169.9769	x1 x2 x3 x4 x7 x8 x9
67	7	170.1446	x1 x2 x5 x6 x7 x8 x9
68	7	171.1780	x1 x2 x6 x7 x8 x9 x10
69	7	171.4681	x1 x2 x3 x7 x8 x9 x10
70	7	172.8186	x1 x2 x4 x5 x7 x8 x10
71	8	167.5800	x1 x2 x4 x5 x7 x8 x9 x10
72	8	167.9901	x1 x2 x3 x4 x5 x7 x8 x9
73	8	168.2598	x1 x2 x4 x5 x6 x7 x8 x9
74	8	169.5936	x1 x2 x4 x6 x7 x8 x9 x10
75	8	170.0664	x1 x2 x3 x4 x7 x8 x9 x10
76	8	170.5989	x1 x2 x3 x5 x7 x8 x9 x10
77	8	170.9540	x1 x2 x5 x6 x7 x8 x9 x10
78	8	171.2864	x1 x2 x3 x5 x6 x7 x8 x9
79	8	171.5856	x1 x2 x3 x4 x6 x7 x8 x9
80	8	172.8143	x1 x2 x3 x6 x7 x8 x9 x10
81	9	169.1473	x1 x2 x3 x4 x5 x7 x8 x9 x10
82	9	169.1817	x1 x2 x4 x5 x6 x7 x8 x9 x10
83	9	169.8498	x1 x2 x3 x4 x5 x6 x7 x8 x9
84	9	171.4891	x1 x2 x3 x4 x6 x7 x8 x9 x10
85	9	172.2950	x1 x2 x3 x5 x6 x7 x8 x9 x10

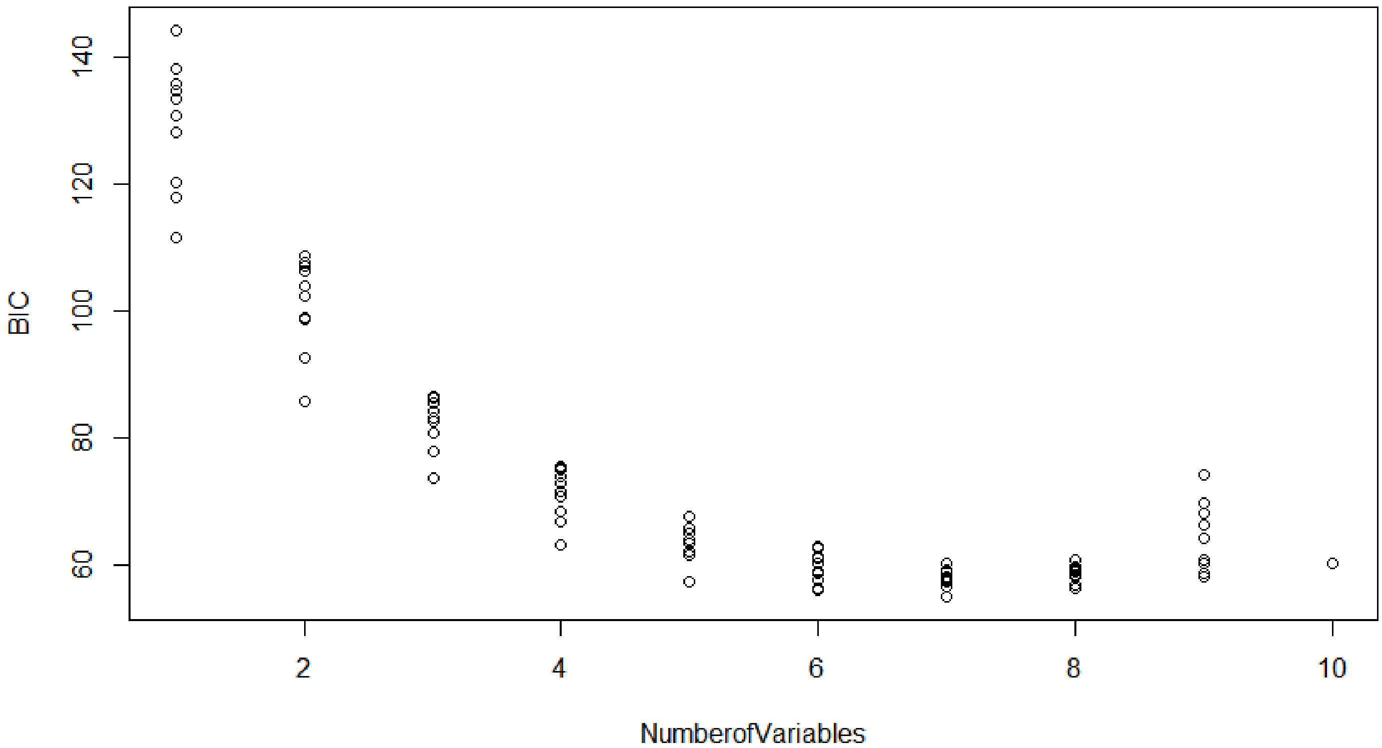



```
> stepwise(data=df,y="Y",select="AIC")
$process
  Step EffectEntered EffectRemoved EffectNumber   Select
1    0     intercept                          1 262.4131
2    1            X2                          2 226.5738
3    2            X7                          3 200.4599
4    3            X8                          4 188.0087
5    4            X9                          5 176.7085
6    5            X1                          6 170.1711
7    6            X4                          7 168.2996
8    7            X5                          8 166.5292

$variate
[1] "intercept" "X2"        "X7"        "X8"        "X9"        "X1"        "X4"        "X5"
```

BIC Table

```
32          4   66.86228                 x1 x2 x7 x8
33          4   68.39498                 x2 x5 x7 x8
34          4   70.73337                x2 x7 x8 x10
35          4   71.46181                 x1 x2 x5 x7
36          4   72.89690                 x2 x4 x7 x8
37          4   73.88236                 x2 x5 x7 x9
38          4   75.10167                 x2 x3 x7 x8
39          4   75.36778                 x2 x6 x7 x8
40          4   75.57978                 x1 x2 x6 x7
41          5   57.41697              x1 x2 x7 x8 x9
42          5   61.58842              x1 x2 x5 x7 x8
43          5   62.11261              x2 x5 x7 x8 x9
44          5   62.20618              x2 x4 x7 x8 x9
45          5   63.47009             x2 x7 x8 x9 x10
46          5   63.82286             x1 x2 x7 x8 x10
47          5   64.90783              x2 x6 x7 x8 x9
48          5   65.05768              x2 x3 x7 x8 x9
49          5   65.67575              x1 x2 x4 x7 x8
50          5   67.55105              x2 x4 x5 x7 x8
51          6   56.07919           x1 x2 x4 x7 x8 x9
52          6   56.29881           x1 x2 x5 x7 x8 x9
53          6   57.64142          x1 x2 x7 x8 x9 x10
54          6   58.79078           x1 x2 x3 x7 x8 x9
55          6   58.84577           x1 x2 x6 x7 x8 x9
56          6   60.34209           x1 x2 x4 x5 x7 x8
57          6   61.17197           x2 x4 x5 x7 x8 x9
58          6   61.40288          x1 x2 x5 x7 x8 x10
59          6   62.67160          x2 x4 x7 x8 x9 x10
60          6   62.83281          x1 x2 x4 x7 x8 x10
61          7   54.98371        x1 x2 x4 x5 x7 x8 x9
62          7   56.49181       x1 x2 x4 x7 x8 x9 x10
63          7   57.48169        x1 x2 x3 x5 x7 x8 x9
64          7   57.52888       x1 x2 x5 x7 x8 x9 x10
65          7   57.74326        x1 x2 x4 x6 x7 x8 x9
66          7   57.91639        x1 x2 x3 x4 x7 x8 x9
67          7   58.05931        x1 x2 x5 x6 x7 x8 x9
68          7   58.94067       x1 x2 x6 x7 x8 x9 x10
69          7   59.18827       x1 x2 x3 x7 x8 x9 x10
70          7   60.34203       x1 x2 x4 x5 x7 x8 x10
71          8   56.38071    x1 x2 x4 x5 x7 x8 x9 x10
72          8   56.72058     x1 x2 x3 x4 x5 x7 x8 x9
73          8   56.94425     x1 x2 x4 x5 x6 x7 x8 x9
74          8   58.05140    x1 x2 x4 x6 x7 x8 x9 x10
75          8   58.44428    x1 x2 x3 x4 x7 x8 x9 x10
76          8   58.88716    x1 x2 x3 x5 x7 x8 x9 x10
77          8   59.18263    x1 x2 x5 x6 x7 x8 x9 x10
78          8   59.45941     x1 x2 x3 x5 x6 x7 x8 x9
79          8   59.70857     x1 x2 x3 x4 x6 x7 x8 x9
80          8   60.73304    x1 x2 x3 x6 x7 x8 x9 x10
81          9   58.23309 x1 x2 x3 x4 x5 x7 x8 x9 x10
82          9   58.26093 x1 x2 x4 x5 x6 x7 x8 x9 x10
83          9   58.80139  x1 x2 x3 x4 x5 x6 x7 x8 x9
84          9   60.13001 x1 x2 x3 x4 x6 x7 x8 x9 x10
85          9   60.78444 x1 x2 x3 x5 x6 x7 x8 x9 x10
86          9   64.30759 x1 x2 x3 x4 x5 x6 x7 x8 x10
```

```
> stepwise(data=df,y="Y",select="BIC")
$process
  Step EffectEntered EffectRemoved EffectNumber    Select
1    0     intercept                          1 147.54972
2    1            x2                          2 111.52505
3    2            x7                          3  85.76639
4    3            x8                          4  73.68800
5    4            x9                          5  63.16567
6    5            x1                          6  57.41697
7    6            x4                          7  56.07919
8    7            x5                          8  54.98371

$variate
[1] "intercept" "x2"        "x7"        "x8"        "x9"        "x1"        "x4"        "x5"

> plot(b[,1:2])
```



2.4.4 Inferences :

From the criterion of adjusted r square, we can conclude that the best-fitted model should have 7 predictors. Among the 7 predictors observation, no 62 suits the best, since it does not have both X5 and X8.

Similarly for other criteria of model selection as well, we can observe from the graphs above that the best fit is suited with 7 predictors, and observation no 62 suits the best. Since we won't have to account for the multicollinearity anymore.

Hence, the final model will have X1, X2, X4, X7, X8, X9, X10 as the predictors.

```
#### Final Model? ####
reduced.lmfit <- lm(Y ~ X1 + X2 + X4 + X7 + X8 + X9 + X10, data=df)
summary(reduced.lmfit)
```

```
> summary(reduced.lmfit)

Call:
lm(formula = Y ~ X1 + X2 + X4 + X7 + X8 + X9 + X10, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.2412 -0.6533 -0.0504  0.5832  6.3562 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.706844   1.612363   2.299  0.02348 *  
X1           0.079294   0.026077   3.041  0.00298 ** 
X2           0.474196   0.107338   4.418 2.43e-05 ***
X4           0.013236   0.006997   1.892  0.06127 .  
X7          -0.644265   0.123078  -5.235 8.53e-07 ***
X8           0.010360   0.001867   5.549 2.17e-07 ***
X9          -0.006448   0.002099  -3.071  0.00272 ** 
X10         -0.017803   0.013015  -1.368  0.17428    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.224 on 105 degrees of freedom
Multiple R-squared:  0.6158,	Adjusted R-squared:  0.5902 
F-statistic: 24.05 on 7 and 105 DF,  p-value: < 2.2e-16
```

3. Assumption Checking

3.1 Model Assumptions

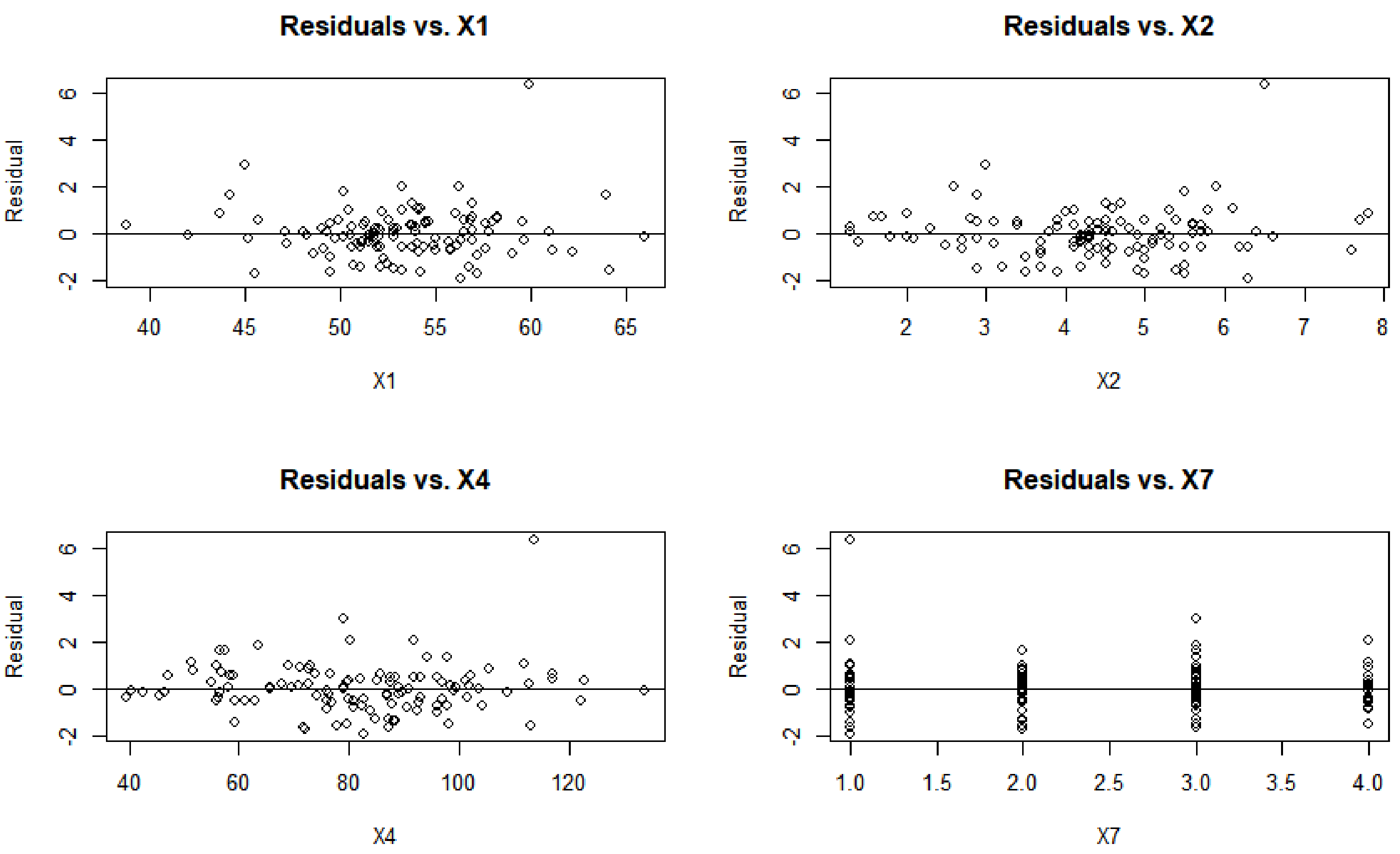
Below are the four very important assumptions required to perform the multiple linear regression.

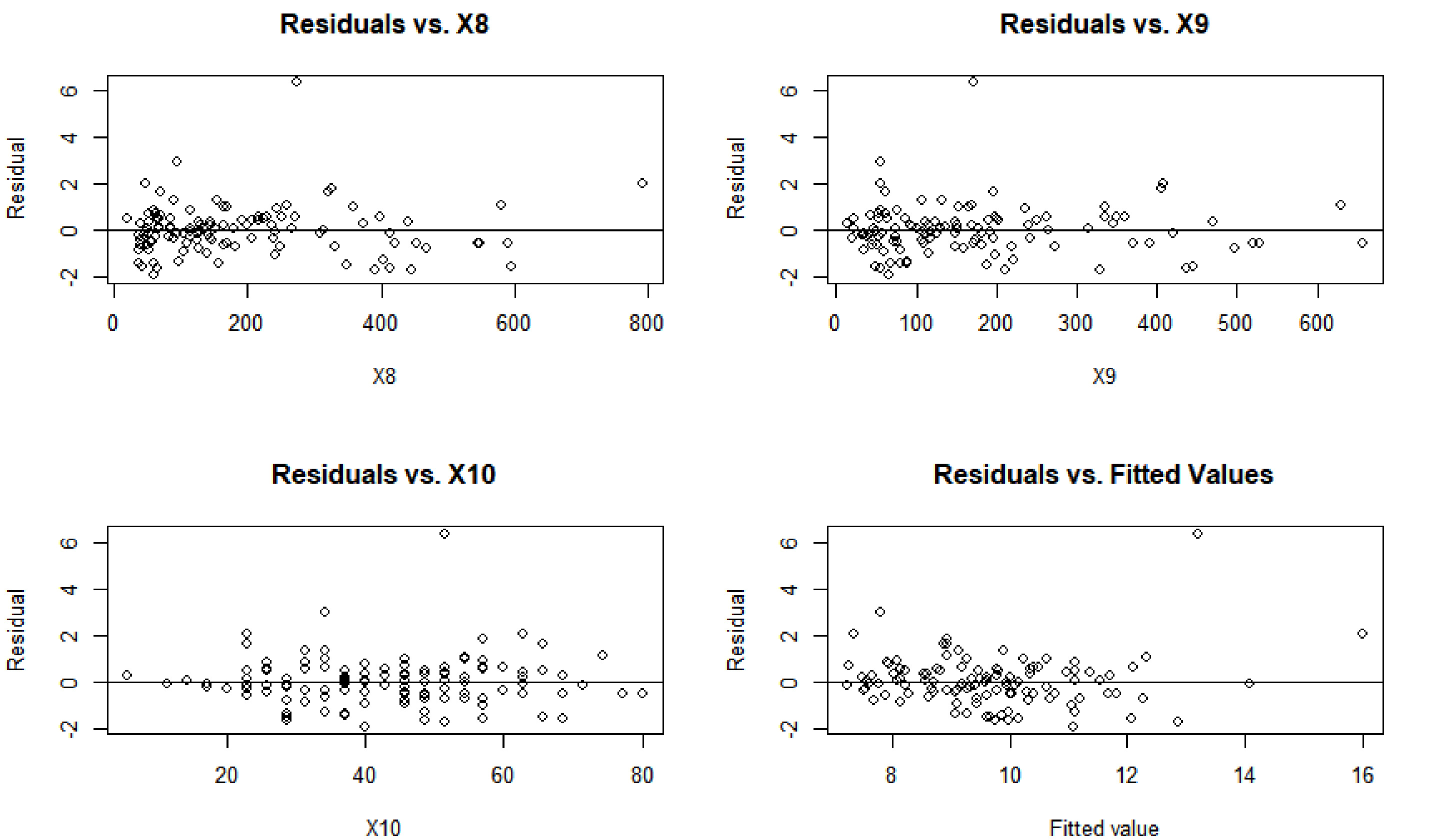
1. Linearity
2. Homogenous or constant variance
3. Independence
4. Normality

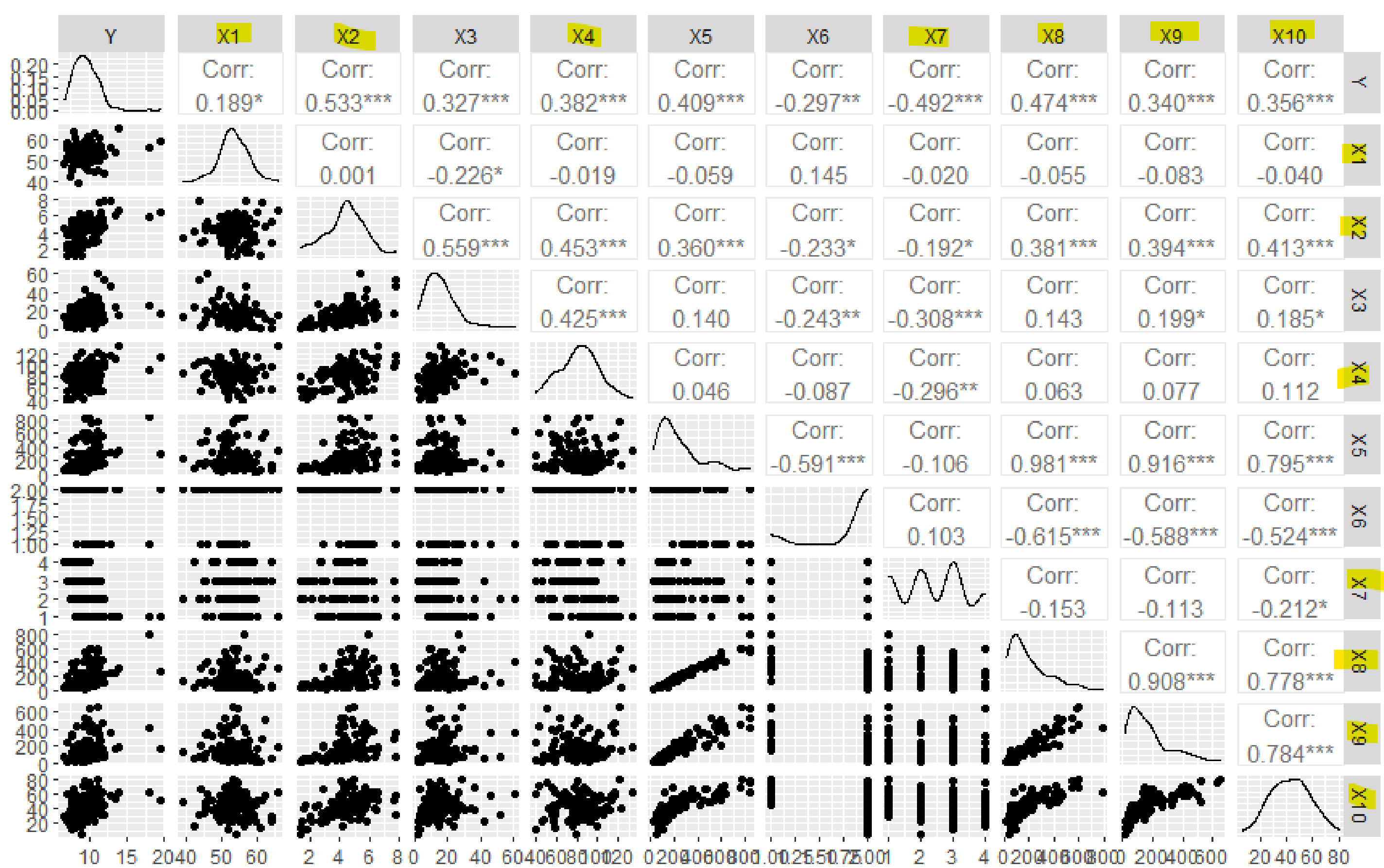
In the next section, we will diagnose any broken assumptions.

3.2 Model Diagnostics using residual plot

```
######### Assumption Checking #################

# Regression Diagnostics #

res <- rstudent(reduced.lmfit)
fitted.y <- fitted(reduced.lmfit)

######## Residual Plots ##########

par(mfrow=c(2,2))
plot(res ~ df$X1, xlab="X1", ylab="Residual", main="Residuals vs. X1")
abline(h=0)
plot(res ~ df$X2, xlab="X2", ylab="Residual", main="Residuals vs. X2")
abline(h=0)
plot(res ~ df$X4, xlab="X4", ylab="Residual", main="Residuals vs. X4")
abline(h=0)
plot(res ~ df$X7, xlab="X7", ylab="Residual", main="Residuals vs. X7")
abline(h=0)
plot(res ~ df$X8, xlab="X8", ylab="Residual", main="Residuals vs. X8")
abline(h=0)
plot(res ~ df$X9, xlab="X9", ylab="Residual", main="Residuals vs. X9")
abline(h=0)
plot(res ~ df$X10, xlab="X10", ylab="Residual", main="Residuals vs. X10")
abline(h=0)

plot(res ~ fitted.y, xlab="Fitted value", ylab="Residual", main="Residuals vs. Fitted values")
abline(h=0)
```



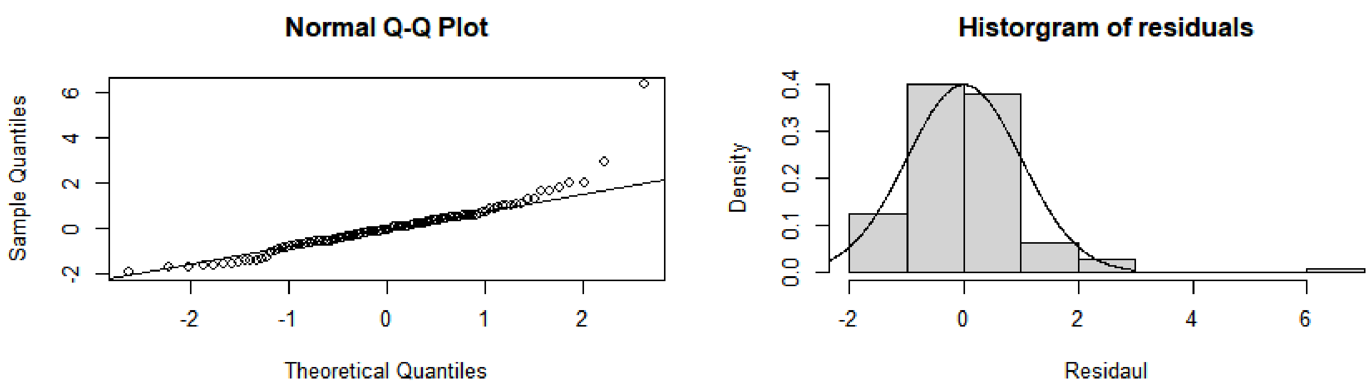

Residuals vs. X8
Residual
X8
Residuals vs. X9
Residual
X9
Residuals vs. X10
Residual
X10
Residuals vs. Fitted Values
Residual
Fitted value


Y
X1
X2
X3
X4
X5
X6
X7
X8
X9
X10
Corr: 0.189*
Corr: 0.533***
Corr: 0.327***
Corr: 0.382***
Corr: 0.409***
Corr: -0.297**
Corr: -0.492***
Corr: 0.474***
Corr: 0.340***
Corr: 0.356***
Corr: 0.001
Corr: -0.226*
Corr: -0.019
Corr: -0.059
Corr: 0.145
Corr: -0.020
Corr: -0.055
Corr: -0.083
Corr: -0.040
Corr: 0.559***
Corr: 0.453***
Corr: 0.360***
Corr: -0.233*
Corr: -0.192*
Corr: 0.381***
Corr: 0.394***
Corr: 0.413***
Corr: 0.425***
Corr: 0.140
Corr: -0.243**
Corr: -0.308***
Corr: 0.143
Corr: 0.199*
Corr: 0.185*
Corr: 0.046
Corr: -0.087
Corr: -0.296**
Corr: 0.063
Corr: 0.077
Corr: 0.112
Corr: -0.591***
Corr: -0.106
Corr: 0.981***
Corr: 0.916***
Corr: 0.795***
Corr: 0.103
Corr: -0.615***
Corr: -0.588***
Corr: -0.524***
Corr: -0.153
Corr: -0.113
Corr: -0.212*
Corr: 0.908***
Corr: 0.778***
Corr: 0.784***

These plots provide the same information about the linearity and independence of the error terms assumptions. Residual vs predictor plot is more useful in multiple linear regression, when we look for, whether the addition of a new predictor to the model is beneficial or not.
This plot is a classic example of well-behaved residuals vs. fitted value plot except for the presence of a few outliers. Below are the characteristics of a well-behaved residual vs. fitted value plot and what they suggest about the appropriateness of the simple linear regression model:

- The residuals "bounce randomly" around the 0 lines. This suggests that the assumption that the relationship is linear is reasonable.
- The residuals roughly form a "horizontal band" around the 0 lines. This suggests that the variances of the error terms are equal.
- Few residual "stand out" from the basic random pattern of residuals. This suggests that there are few outliers.
- The residuals vs the predictor also explain the linear relationship between the response variable and the predictors, since there is no forming pattern in the plot and also, the same can be verified from the scatter plot matrix. However, the linearity is weak with some of the predictors.

Hence from the above plots, the assumption of linearity, independence, constant variance comes out to be true. To diagnose the assumptions of normality, we need to draw the q-q plot.



We can see from the plot above that the Q-Q plot is approximately normal with the presence of few outliers in the end.
Also, the histograms of residuals have a normal distribution with some outliers. Hence from the graphical diagnostics, none of the assumptions are getting violated.

In the next section, we will perform various numerical tests to make sure that the diagnostics obtained due to the graphical interpretation confirms the test results.

3.3 Assumption checking using different tests.

1. First, we are checking for the Normality assumption

```
########
 # Checking normality using the Shapiro test
shapiro.test(res)
```

```
        Shapiro-Wilk normality test

data:  res
W = 0.86966, p-value = 1.559e-08
```

In the Shapiro-Wilk test, the null hypothesis states that the error terms are normally distributed while the alternative hypothesis states that the error terms are not normally distributed. Our decision rule states that if the test statistic is small and the p-value is less than the significance level (alpha = 0.05), then we must reject the null hypothesis. If the test statistic is large and the p-value is greater than the significance level, we must accept the null hypothesis. We calculated a test statistic of 0.86966 and a p-value less than 0.05. Thus, we must reject the null hypothesis and conclude that the error terms are not normally distributed.

Thus from the test above, the normality assumption has failed.

2. Test for the constancy of Variance.

We will conduct the Breush-Pagan test to conduct the constancy of variance.

```
### Checking the constancy of variance using Breush Pagan test
library(lmtest)
bptest(reduced.lmfit)
```

```
> bptest(reduced.lmfit)

        studentized Breusch-Pagan test

data:  reduced.lmfit
BP = 8.8676, df = 7, p-value = 0.2623
```

In the Breusch-Pagan test, the null hypothesis states that there is constant error variance and the alternative hypothesis states that there is no constant variance. The decision rule is that if the p-value is less than the significance level of 0.05, we will reject the null hypothesis and conclude that the error variance is not constant. If the p-value is greater than the significance level of 0.05, we will accept the null hypothesis and conclude that the error variance is constant. We calculated a test statistic of 8.8676 and a p-value of 0.2623, so we fail to reject the null hypothesis and conclude that the error variance is constant.

3. Darwin Watson test for checking independence.

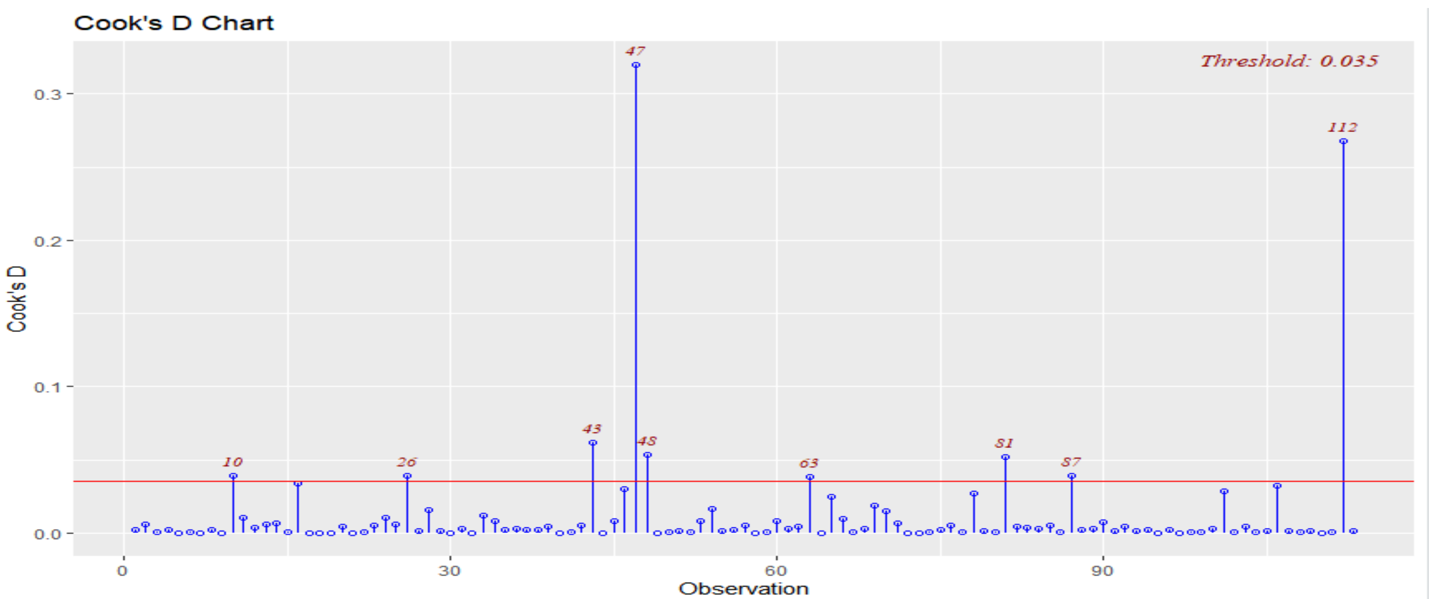
```
### Checking the independence using the darwin- watson test
library(lmtest)
dwtest(reduced.lmfit)
```

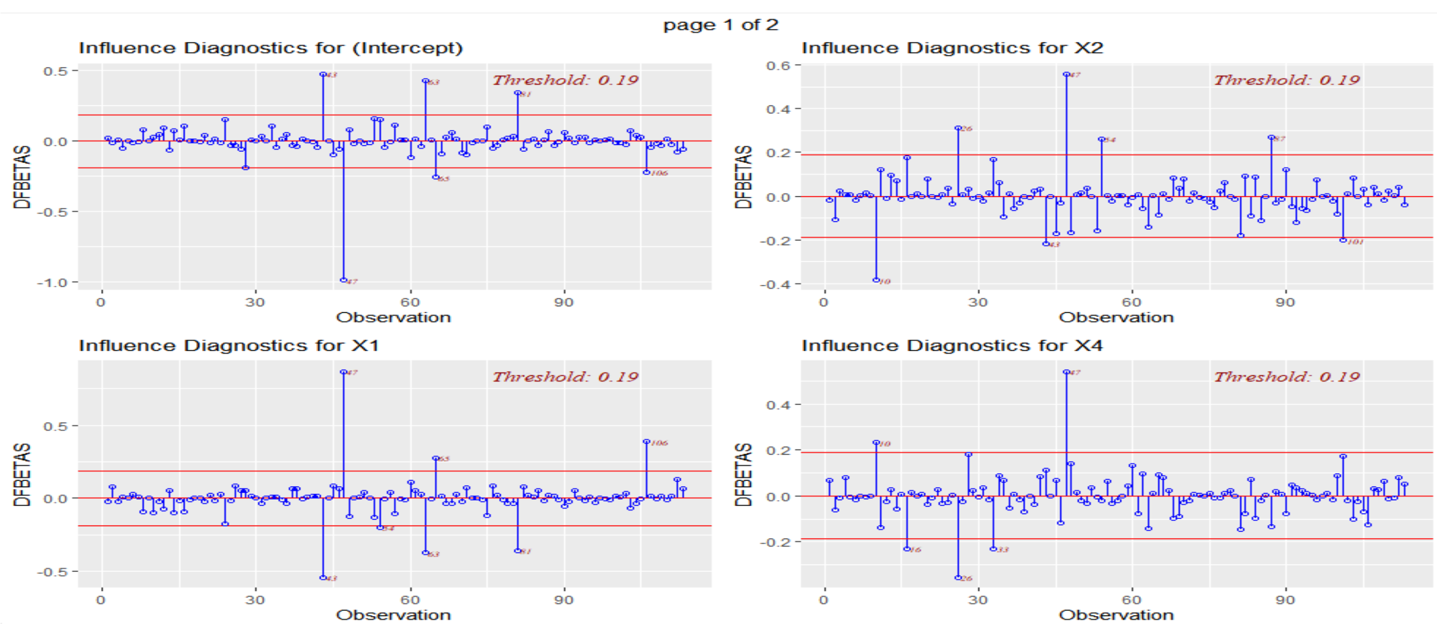
```
> dwtest(reduced.lmfit)

        Durbin-Watson test

data:  reduced.lmfit
DW = 1.9691, p-value = 0.4288
alternative hypothesis: true autocorrelation is greater than 0
```

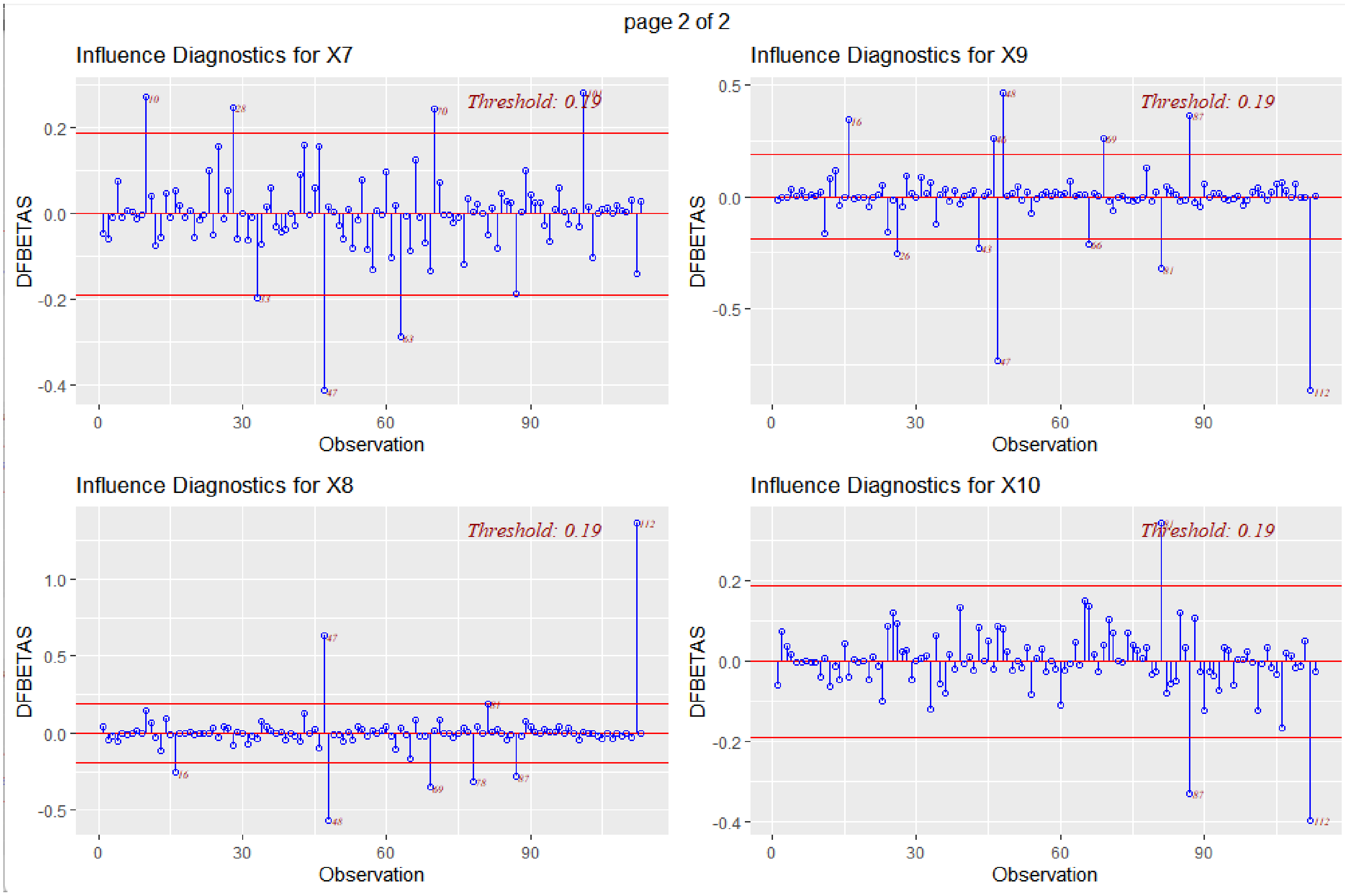
Since the p-value is greater than 0.05, we need to accept the null hypothesis and conclude that the null hypothesis is not rejected thus the assumption of independence is not violated.

4. Detecting outliers

```
############ Detecting outliers ##########


# Detection methods of influential points
# Install a package, called "olsrr"
install.packages("olsrr")
library(olsrr)
# DFFITS
ols_plot_dffits(reduced.lmfit)
# Cook's D
ols_plot_cooksd_chart(reduced.lmfit)
# DFBETAS
ols_plot_dfbetas(reduced.lmfit)
cooks.distance(reduced.lmfit)
dffits(reduced.lmfit)
dfbetas(reduced.lmfit)
```







The Cook's D chart and the dffits and dfbetas suggest that our model is infested with the presence of outliers and it can have an effect on the final model. We need to fix the issue of outliers in the remedial action.

5. Testing Multicollinearity

```
## Testing Multicollinearity
# Check Variance Inflation Factor
vif(reduced.lmfit)
```

```
> vif(reduced.lmfit)
      X1       X2       X4       X7       X8       X9      X10
1.012654 1.549749 1.373126 1.154718 6.165637 6.394902 2.928264
>
```

None of the values obtained is greater than 10, hence there is not much effect of multicollinearity.

4. The necessity of remedial actions

Since the model has been shown to show linearity, constant variance (homoscedasticity), independence, except for the normality, a box-cox transformation is needed to make the model normal. There was no multicollinearity found in the reduced model. Hence, we just need to perform the box-cox transformation to correct the model normality issue.

5. Transformation methods

Using the box- cox transformation, we have transformed the model, to fix its normality assumptions.

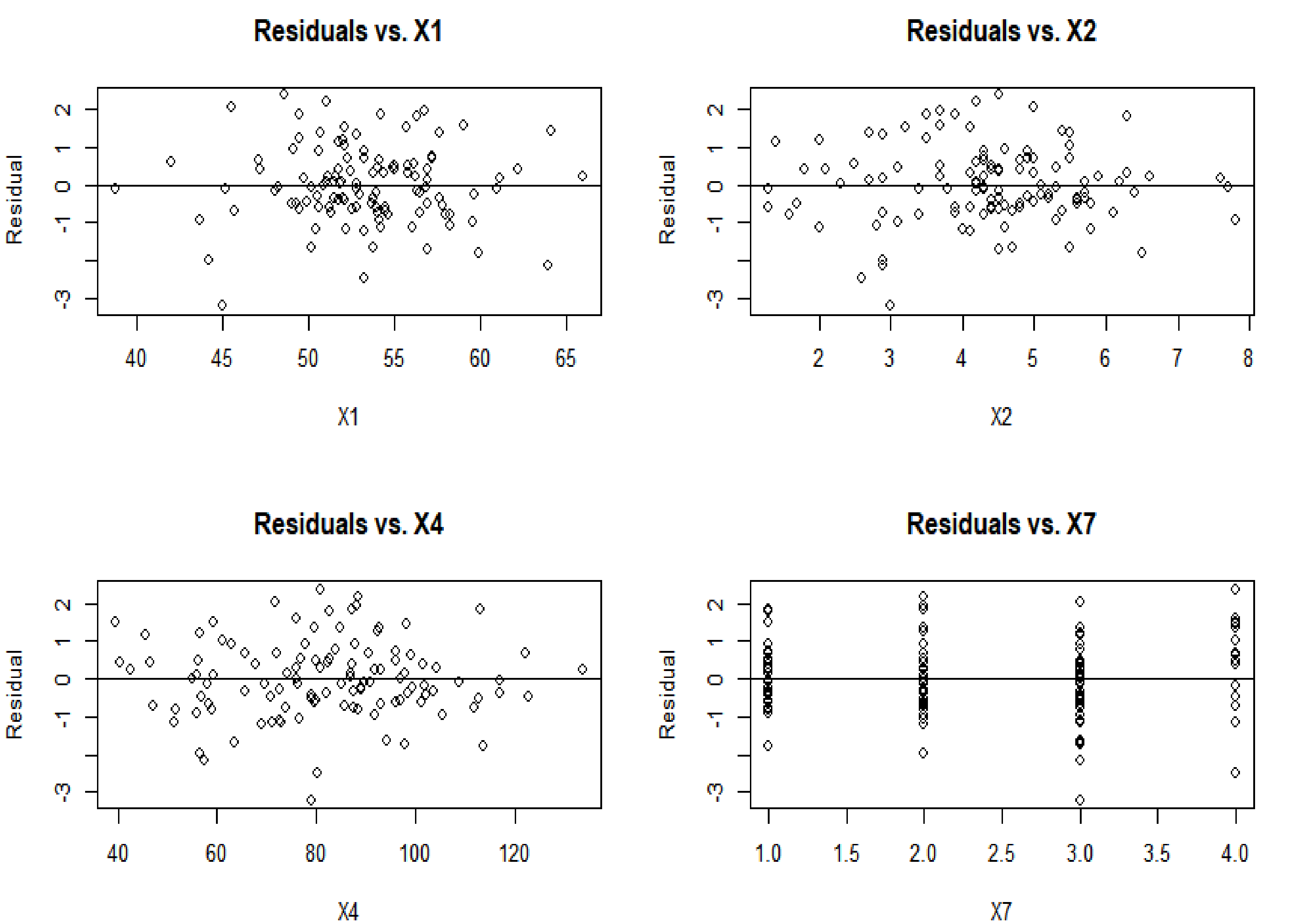
```
######### Transformation #########
library(EnvStats)
boxcox.summary <- boxcox(reduced.lmfit, optimize=TRUE)
lambda <- boxcox.summary$lambda
trans.Y <- df$Y^lambda
df <- cbind(df,trans.Y)

######### Re-fitting a model using the transformed response variable. ##########

boxcox.lmfit <- lm(trans.Y ~ X1 + X2 + X4 + X7 + X8 + X9 + X10 , data=df)
summary(boxcox.lmfit)
boxcox.res <- rstudent(boxcox.lmfit)
boxcox.fitted.y <- fitted(boxcox.lmfit)
```

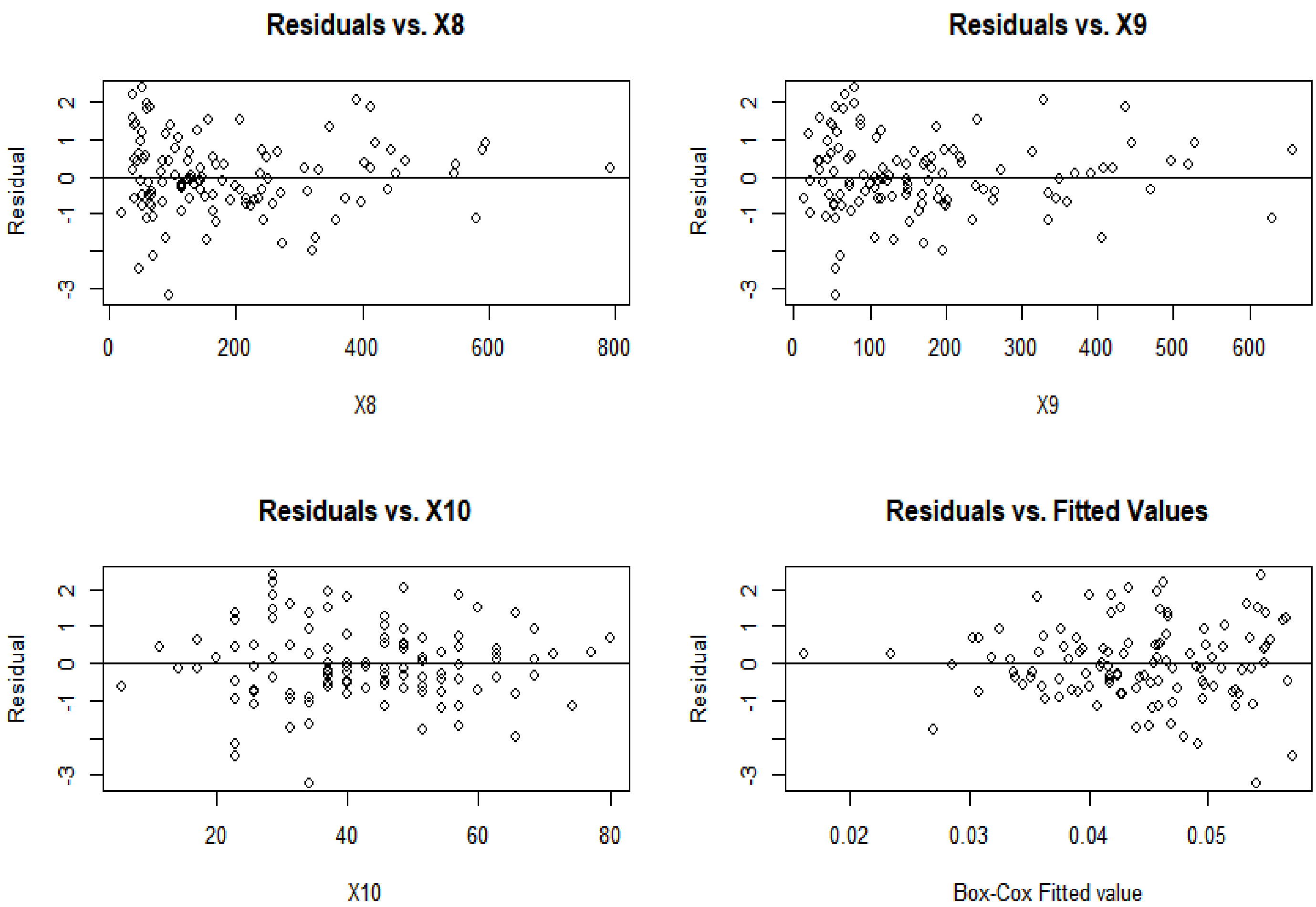
The model is summarized below after the box cox transformation.

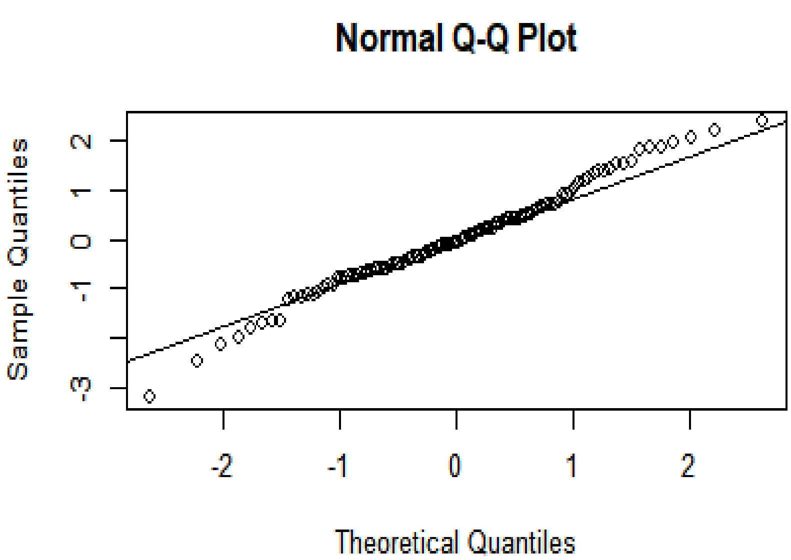
```
> summary(boxcox.lmfit)

Call:
lm(formula = trans.Y ~ X1 + X2 + X4 + X7 + X8 + X9 + X10, data = df)

Residuals:
       Min         1Q     Median         3Q        Max 
-0.0200863 -0.0040058 -0.0005342  0.0036149  0.0152468 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  6.878e-02  8.867e-03   7.757 5.94e-12 ***
X1          -3.337e-04  1.434e-04  -2.327 0.021896 *  
X2          -2.509e-03  5.903e-04  -4.250 4.64e-05 ***
X4          -5.215e-05  3.848e-05  -1.355 0.178212    
X7           3.965e-03  6.768e-04   5.858 5.43e-08 ***
X8          -3.643e-05  1.027e-05  -3.548 0.000582 ***
X9           1.424e-05  1.155e-05   1.233 0.220320    
X10          7.669e-05  7.157e-05   1.072 0.286389    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.006729 on 105 degrees of freedom
Multiple R-squared:  0.5865,	Adjusted R-squared:  0.559 
F-statistic: 21.28 on 7 and 105 DF,  p-value: < 2.2e-16
```

6. Model diagnostics of the transformed model

```
######### Multicollinearity ##########
vif(boxcox.lmfit) ### OK ###
######## Residual Plots ##########
par(mfrow=c(2,2))
plot(boxcox.res ~ df$X1, xlab="X1", ylab="Residual", main="Residuals vs. X1")
abline(h=0)
plot(boxcox.res ~ df$X2, xlab="X2", ylab="Residual", main="Residuals vs. X2")
abline(h=0)
plot(boxcox.res ~ df$X4, xlab="X4", ylab="Residual", main="Residuals vs. X4")
abline(h=0)
plot(boxcox.res ~ df$X7, xlab="X7", ylab="Residual", main="Residuals vs. X7")
abline(h=0)
plot(boxcox.res ~ df$X8, xlab="X8", ylab="Residual", main="Residuals vs. X8")
abline(h=0)
plot(boxcox.res ~ df$X9, xlab="X9", ylab="Residual", main="Residuals vs. X9")
abline(h=0)
plot(boxcox.res ~ df$X10, xlab="X10", ylab="Residual", main="Residuals vs. X10")
abline(h=0)
plot(boxcox.res ~ boxcox.fitted.y, xlab="Box-Cox Fitted value", ylab="Residual", main="Residuals vs. Fitted values")
abline(h=0)
library(GGally)
#generate the pairs plot
ggpairs(df)
######### Constancy of Error Variances #########
library(lmtest)
bptest(boxcox.lmfit)
######### Normality ###########
qqnorm(boxcox.res);qqline(boxcox.res)
shapiro.test(boxcox.res)
```




Residuals vs. X8
Residual
X8
Residuals vs. X9
Residual
X9
Residuals vs. X10
Residual
X10
Residuals vs. Fitted Values
Residual
Box-Cox Fitted value


Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles

```
> ######### Multicollinearity ##########
> vif(boxcox.lmfit) ### OK ###
      X1       X2       X4       X7       X8       X9      X10 
1.012654 1.549749 1.373126 1.154718 6.165637 6.394902 2.928264 
> ######## Residual Plots ##########
> par(mfrow=c(2,2))
> plot(boxcox.res ~ df$X1, xlab="X1", ylab="Residual", main="Residuals vs. X1")
> abline(h=0)
> plot(boxcox.res ~ df$X2, xlab="X2", ylab="Residual", main="Residuals vs. X2")
> abline(h=0)
> plot(boxcox.res ~ df$X4, xlab="X4", ylab="Residual", main="Residuals vs. X4")
> abline(h=0)
> plot(boxcox.res ~ df$X7, xlab="X7", ylab="Residual", main="Residuals vs. X7")
> abline(h=0)
> plot(boxcox.res ~ df$X8, xlab="X8", ylab="Residual", main="Residuals vs. X8")
> abline(h=0)
> plot(boxcox.res ~ df$X9, xlab="X9", ylab="Residual", main="Residuals vs. X9")
> abline(h=0)
> plot(boxcox.res ~ df$X10, xlab="X10", ylab="Residual", main="Residuals vs. X10")
> abline(h=0)
> plot(boxcox.res ~ boxcox.fitted.y, xlab="Box-Cox Fitted value", ylab="Residual", main="Residuals vs. Fitted Values")
> abline(h=0)
> library(GGally)
> #generate the pairs plot
> ggpairs(df)
 plot: [1,1] [>-----------------------------------------------------------------------------------------------------------------
ata` must be uniquely named but has duplicate columns
Run `rlang::last_error()` to see where the error occurred.
> ######### Constancy of Error Variances #########
> library(lmtest)
> bptest(boxcox.lmfit)

	studentized Breusch-Pagan test

data:  boxcox.lmfit
BP = 7.4581, df = 7, p-value = 0.3828

> ######### Normality ###########
> qqnorm(boxcox.res);qqline(boxcox.res)
> shapiro.test(boxcox.res)

	Shapiro-Wilk normality test

data:  boxcox.res
W = 0.98797, p-value = 0.4153
```

As can be seen from the results obtained above, post the box cox transformation, all the assumptions mandatory for the Multiple linear regression have passed. We are good to go with the final model. The final model passed all the necessary required conditions required for fitting the multiple linear regression model.

The transformation method did work on fixing the normality issue of the model.

7. Result

The final model that we decided best fit our data is shown below in the summary output of the final model. This model contains the predictors X1, X2, X4, X7, X8, X9, X10. This model produced a relatively moderate adjusted R^2 value of 0.559, meaning that about 55. 9% of the data can be explained by this model.

We tried to further improve the model accuracy using the interaction terms, however, they seem to violate the other assumptions when put into the model.
Mentioned below are the final summary of the model and the ANOVA table.

```
> summary(final.lmfit)

Call:
lm(formula = trans.Y ~ X1 + X2 + X4 + X7 + X8 + X9 + X10, data = df)

Residuals:
       Min         1Q     Median         3Q        Max 
-0.0200863 -0.0040058 -0.0005342  0.0036149  0.0152468 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  6.878e-02  8.867e-03   7.757 5.94e-12 ***
X1          -3.337e-04  1.434e-04  -2.327 0.021896 *  
X2          -2.509e-03  5.903e-04  -4.250 4.64e-05 ***
X4          -5.215e-05  3.848e-05  -1.355 0.178212    
X7           3.965e-03  6.768e-04   5.858 5.43e-08 ***
X8          -3.643e-05  1.027e-05  -3.548 0.000582 ***
X9           1.424e-05  1.155e-05   1.233 0.220320    
X10          7.669e-05  7.157e-05   1.072 0.286389    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.006729 on 105 degrees of freedom
Multiple R-squared:  0.5865,	Adjusted R-squared:  0.559 
F-statistic: 21.28 on 7 and 105 DF,  p-value: < 2.2e-16

> anova(final.lmfit)
Analysis of Variance Table

Response: trans.Y
           Df    Sum Sq   Mean Sq F value    Pr(>F)    
X1          1 0.0002387 0.0002387  5.2724   0.02365 *  
X2          1 0.0033460 0.0033460 73.9031 8.447e-14 ***
X4          1 0.0002423 0.0002423  5.3526   0.02264 *  
X7          1 0.0019142 0.0019142 42.2789 2.734e-09 ***
X8          1 0.0008323 0.0008323 18.3823 4.024e-05 ***
X9          1 0.0001178 0.0001178  2.6027   0.10969    
X10         1 0.0000520 0.0000520  1.1482   0.28639    
Residuals 105 0.0047539 0.0000453                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

Some of the predictors fail the individual t-test, however overall the model passes the global f-test. This might be due to the smaller amount of multicollinearity present in the model.

8. Conclusion

We started the model fitting by first importing the dataset and then performing the exploratory data analysis. Exploratory data analysis provided some insights into how the dataset looks like. The data set with 19 observations is too small observation to make a good multiple linear regression model. Latter we fit a full model based on all the predictors in the model and found that man assumptions were getting violated. Then we looked for the multicollinearity in the model and removed the predictor responsible for multicollinearity based on the performed analysis. We did stepwise regression and looked into all the criteria required to select the final model. Once the final model was selected, we performed the assumption diagnostics and assumption checking. Based on the assumption checking, the normality assumption of the model was violated. Latter we performed the box cox transformation to fix the normality issue of the model and finally got our transformed model.

The transformed model had an efficiency of 55 percent, which is a moderate efficiency. We further tried to bring the effect of the interaction terms into the model and we got an efficiency of 64 percent, however, it seems to violate the other assumptions which were not possible to fix.

I also tried to fit the scale of the data to check an impact on its effective r square, however, there was not much impact. Overall the result obtained is satisfactory given the size of the dataset.

There could be some possibility of improvements such as removing any possible inflection points, scaling the data further, and inserting interaction terms, and performing hit and trail to improve the model efficiency.

9. Appendix

```
# READING THE DATA

senic <- read.csv("C:/Users/nehal/Desktop/MTU/MTU/Courses/MA4710 - RA/MathsFinalProject/SENIC.csv")

head(senic)

install.packages("dplyr")

library(dplyr)

# Getting the required data frame for Analysis

df <- senic

View(df)

# Creating the design matrix of X

Y <- matrix(df[,1], ncol=1)

X <- as.matrix(df[,-1])

X <- cbind(1,X)

colnames(X)[1] <- 'Intercept'

colnames(Y)[1] <- 'Y'

View(X)

View(Y)

## Exploratory Data Analysis

# Creating the histogram of the target and predictor variables

install.packages('ggplot2')

library(ggplot2)

# Building the histogram

ggplot(data=df, aes(df$Y)) +

  geom_histogram(aes(y =..density..), fill = "red") +

  geom_density()

ggplot(data=df, aes(df$X1)) +

  geom_histogram(aes(y =..density..), fill = "blue") +

  geom_density()

ggplot(data=df, aes(df$X2)) +

  geom_histogram(aes(y =..density..), fill = "green") +

  geom_density()

ggplot(data=df, aes(df$X3)) +

  geom_histogram(aes(y =..density..), fill = "black") +

  geom_density()

ggplot(data=df, aes(df$X4)) +

  geom_histogram(aes(y =..density..), fill = "purple") +

  geom_density()

ggplot(data=df, aes(df$X5)) +

  geom_histogram(aes(y =..density..), fill = "blue") +

  geom_density()
```

```
ggplot(data=df, aes(df$X6)) +
  geom_histogram(aes(y =..density..), fill = "green") +
  geom_density()
ggplot(data=df, aes(df$X7)) +
  geom_histogram(aes(y =..density..), fill = "black") +
  geom_density()
ggplot(data=df, aes(df$X8)) +
  geom_histogram(aes(y =..density..), fill = "purple") +
  geom_density()
ggplot(data=df, aes(df$X9)) +
  geom_histogram(aes(y =..density..), fill = "blue") +
  geom_density()
ggplot(data=df, aes(df$X10)) +
  geom_histogram(aes(y =..density..), fill = "green") +
  geom_density()
# Getting the Summary Statistics using the psych package
install.packages('psych')
library(psych)
psych::describe(df)
# Creating the correlation matrix
cor_df <- cor(df)
cor_df
library(GGally)
#generate the pairs plot
ggpairs(df)
# Visualizing the correlation matrix i.e., getting a scatter plot matrix
pairs(df)
# Getting the box plots to look for the outliers
library(reshape)
senicData <- melt(df)
boxplot <- ggplot(senicData, aes(factor(variable), value))
boxplot + geom_boxplot() + facet_wrap(~variable, scale="free")
# Added-Variable Plots
install.packages('car')
library(car)
# Fitting the model
lmfit <- lm(Y ~ . , data = df)
avPlots(lmfit)
##################################################################
# For scaling, however no change was visible
```

```
summary(lmfit)
library(plyr)
library(readr)
library(ggplot2)
library(GGally)
library(dplyr)
library(mlbench)
library(caret)
preproc2 <- preProcess(df, method=c("range"))
norm2 <- predict(preproc2, df)
summary(norm2)
lmfit_norm <- lm(Y ~ . , data = norm2)
vif(lmfit_norm)
############################################################################################
# Model fitting
# Fit a multiple linear regression model
lmfit.full <- lm(Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10, data=df)
summary(lmfit.full)
anova(lmfit.full)
# Multicollinearity check
summary(lmfit.full)
cor(df)   ###### The signs of some estimates from the output and the correlation matrix are different.
anova(lmfit.full)
# Check Variance Inflation Factor
vif(lmfit.full)
#  Consider other regression models
lmfit1 <- lm(Y ~ X1 + X2 + X3 + X4 +  X6 + X7 + X8 + X9 + X10, data=df)
lmfit2 <- lm(Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 +  X9 + X10, data=df)
summary(lmfit1)
summary(lmfit2)
# Comparison between the full model and the reduced models
anova(lmfit1, lmfit.full)
anova(lmfit2, lmfit.full)
# VIF on the reduced models
vif(lmfit1)
vif(lmfit2)
#################### Model Selection (Step wise Regression) #####################
#### Install packages for the model selection ####
install.packages("leaps")
install.packages("HH")
```

```
install.packages("StepReg")
#### Load HH, leaps, and StepReg packages ####
library(leaps)
library(HH)
library(StepReg)
par(mfrow=c(1,1))
#### Stepwise Regression ####
#### Adjusted R2 ####
b = bestsubset(data=df,y="Y",select="adjRsq",best=10)
print(b)
stepwise(data=df,y="Y",select="adjRsq")
plot(b[,1:2])

#### Cp ####
b = bestsubset(data=df,y="Y",select="CP",best=10)
print(b)
stepwise(data=df,y="Y",select="CP")
plot(b[,1:2])
#### AIC ####
b = bestsubset(data=df,y="Y",select="AIC",best=10)
print(b)
stepwise(data=df,y="Y",select="AIC")
plot(b[,1:2])
#### BIC ####
b = bestsubset(data=df,y="Y",select="BIC",best=10)
print(b)
stepwise(data=df,y="Y",select="BIC")
plot(b[,1:2])
#### Final Model? ####
lmfit_reduced_final <- lm(Y ~ X1 + X2 + X4 + X7 + X8 + X9 + X10 + X5 + (X1 + X2 + X4  + X8 + X9 + X10 + X5)* X7, data=df)
reduced.lmfit <- lm(Y ~ X1 + X2 + X4 + X7 + X8 + X9 + X10, data=df)
summary(reduced.lmfit)
######### Assumption Checking #################
# Regression Diagnostics #
res <- rstudent(reduced.lmfit)
fitted.y <- fitted(reduced.lmfit)
######## Residual Plots ##########
par(mfrow=c(2,2))
plot(res ~ df$X1, xlab="X1", ylab="Residual", main="Residuals vs. X1")
abline(h=0)
```

```
plot(res ~ df$X2, xlab="X2", ylab="Residual", main="Residuals vs. X2")
abline(h=0)
plot(res ~ df$X4, xlab="X4", ylab="Residual", main="Residuals vs. X4")
abline(h=0)
plot(res ~ df$X7, xlab="X7", ylab="Residual", main="Residuals vs. X7")
abline(h=0)
plot(res ~ df$X8, xlab="X8", ylab="Residual", main="Residuals vs. X8")
abline(h=0)
plot(res ~ df$X9, xlab="X9", ylab="Residual", main="Residuals vs. X9")
abline(h=0)
plot(res ~ df$X10, xlab="X10", ylab="Residual", main="Residuals vs. X10")
abline(h=0)

plot(res ~ fitted.y, xlab="Fitted value", ylab="Residual", main="Residuals vs. Fitted Values")
abline(h=0)

library(GGally)
#generate the pairs plot
ggpairs(df)
######### Normality ###########
qqnorm(res);qqline(res)
shapiro.test(res)
hist(res, main="Historgram of residuals",
     xlab="Residaul",probability = TRUE)
lines(seq(-3,3,length.out = 1000),dnorm(seq(-3,3,length.out = 1000)))
########
 # Checking normality using the Shapiro test
shapiro.test(res)
### Checking the constancy of variance using Breush Pagan test
library(lmtest)
bptest(reduced.lmfit)
### Checking the independence using the Darwin- Watson test
library(lmtest)
dwtest(reduced.lmfit)
############# Detecting outliers ##########
# Detection methods of influential points
# Install a package, called "olsrr"
install.packages("olsrr")
library(olsrr)
# DFFITS
```

```
ols_plot_dffits(reduced.lmfit)

# Cook's D

ols_plot_cooksd_chart(reduced.lmfit)

# DFBETAS

ols_plot_dfbetas(reduced.lmfit)

cooks.distance(reduced.lmfit)

dffits(reduced.lmfit)

dfbetas(reduced.lmfit)

## Testing Multicollinearity

# Check Variance Inflation Factor

vif(reduced.lmfit)

######### Transformation #########

library(EnvStats)

boxcox.summary <- boxcox(reduced.lmfit, optimize=TRUE)

lambda <- boxcox.summary$lambda

trans.Y <- df$Y^lambda

df <- cbind(df,trans.Y)

######### Re-fitting a model using the transformed response variable. ##########

boxcox.lmfit <- lm(trans.Y ~ X1 + X2 + X4 + X7 + X8 + X9 + X10 , data=df)

summary(boxcox.lmfit)

boxcox.res <- rstudent(boxcox.lmfit)

boxcox.fitted.y <- fitted(boxcox.lmfit)

######### Multicollinearity ##########

vif(boxcox.lmfit) ### OK ###

######## Residual Plots ##########

par(mfrow=c(2,2))

plot(boxcox.res ~ df$X1, xlab="X1", ylab="Residual", main="Residuals vs. X1")

abline(h=0)

plot(boxcox.res ~ df$X2, xlab="X2", ylab="Residual", main="Residuals vs. X2")

abline(h=0)

plot(boxcox.res ~ df$X4, xlab="X4", ylab="Residual", main="Residuals vs. X4")

abline(h=0)

plot(boxcox.res ~ df$X7, xlab="X7", ylab="Residual", main="Residuals vs. X7")

abline(h=0)

plot(boxcox.res ~ df$X8, xlab="X8", ylab="Residual", main="Residuals vs. X8")

abline(h=0)

plot(boxcox.res ~ df$X9, xlab="X9", ylab="Residual", main="Residuals vs. X9")

abline(h=0)

plot(boxcox.res ~ df$X10, xlab="X10", ylab="Residual", main="Residuals vs. X10")

abline(h=0)
```

```
plot(boxcox.res ~ boxcox.fitted.y, xlab="Box-Cox Fitted value", ylab="Residual", main="Residuals vs. Fitted Values")

abline(h=0)

library(GGally)

#generate the pairs plot

ggpairs(df)

######### Constancy of Error Variances #########

library(lmtest)

bptest(boxcox.lmfit)

######### Normality ###########

qqnorm(boxcox.res);

qqline(boxcox.res)

shapiro.test(boxcox.res)

######### Final Model ##########

final.lmfit <- boxcox.lmfit

summary(final.lmfit)

anova(final.lmfit)
```