Hair Product

–Mini Project

By,

Vompolu Sai Tanuj

G1 PGP BABI

Table of Contents

1. Project Objective:

The objective of the project is to do an exploratory data analysis on the given dataset “Factor-Hair-Revised” and **finding a Optimum Regression model while answering the questions asked**. The questions asked are based on Factor Analysis and Regression which will be the primary concern of this project. The solutions and results which we get from the EDA and the questions will help us give us a better understanding of the Dataset and some helpful insights on the variables of the market segmentation and the variables affecting the Customer Satisfaction.

2. Exploration of Dataset (Exploratory Data Analysis):

The dataset provided for this project is stored in a **CSV** (Comma Separated Values) file. The file name is **“Factor-Hair-Revised.csv”** which contains the Variables present in the market segments along with their ratings. Along with the dataset provided are a few **questions based on the Factor Analysis and Regression** related to the segmentation of the markets. There is also details on to what each of the variable **actually means** corresponding to their **short variable names** as given in the dataset.

a. Invoking the required libraries to be used in the R Code.

The required packages must be installed and their respective libraries must be called out in the R code before calling out any function from that library. The libraries if not found in the R directory can be installed by using **install.packages ("package name").** The internet must be connected so that required package gets downloaded and installed. After installation, the library must be called into the R code using the function, **library(packagename).** For this project, the following libraries must be invoked to be used in the R code.

- **readr** – This library is used to **import the dataset** from the **CSV file to the R Studio.**

```
install.packages("readr")
library(readr)
```

- **ggplot2** – This library is used to **create many types of graphs** not found in the in-built library and it has lots of customizations.

```
install.packages("ggplot2")
library(ggplot2)
```

- **hmisc** – This library is useful to **create histograms** for **Uni-Variate Analysis**

```
install.packages("Hmisc")
library(Hmisc)
```

- **reshape2 –** This library has **melt()** function which will be useful for creating boxplot in **Uni-Variate Analysis.**

```
install.packages("reshape2")
library(reshape2)
```

- **psych** – This library is useful for **getting correlation plots**

```
install.packages("psych")
library(psych)
```

- **car –** This library is useful for getting **VIF** values**.**

```
install.packages("car")
library(car)
```

b. Creating the working directory and importing the dataset.

The working directory in the **R Console** must be set to the directory where the CSV file exists. The working directory can be changed using the function **setwd().** To get the current directory of the R Console, the function **getwd()** can be used.

```
> setwd("C:/R programs great lakes/ADV STAT")
> getwd()
[1] "C:/R programs great lakes/ADV STAT"
```

After setting the working directory, we must now import the dataset from the **CSV file** to the R console using the function **read.csv()** with the argument **header = TRUE.** The dataset assigned to this project is assigned to a data frame with the name **Market** and the same will be used to call the dataset further when required in the **R Console.**

```
### Loading up the dataset #####
Market = read.csv("Factor-Hair-Revised.csv",header = TRUE)
```

c. Identification of different variables.

To get an idea on all the variables in the dataset, we must first be able to identify all the variables in the dataset and get an idea about all of them before performing Uni-Variate and Bi-Variate Analysis. The following functions can be used for identification of the variables:

- **summary() –** This function is used to get a **Simple Statistical summary** on each of the variables.

```
> summary(Market)
       ID              ProdQual          Ecom          TechSup     
 Min.   :  1.00   Min.   : 5.000   Min.   :2.200   Min.   :1.300  
 1st Qu.: 25.75   1st Qu.: 6.575   1st Qu.:3.275   1st Qu.:4.250  
 Median : 50.50   Median : 8.000   Median :3.600   Median :5.400  
 Mean   : 50.50   Mean   : 7.810   Mean   :3.672   Mean   :5.365  
 3rd Qu.: 75.25   3rd Qu.: 9.100   3rd Qu.:3.925   3rd Qu.:6.625  
 Max.   :100.00   Max.   :10.000   Max.   :5.700   Max.   :8.500  
    CompRes       Advertising       ProdLine      SalesFImage    
 Min.   :2.600   Min.   :1.900   Min.   :2.300   Min.   :2.900  
 1st Qu.:4.600   1st Qu.:3.175   1st Qu.:4.700   1st Qu.:4.500  
 Median :5.450   Median :4.000   Median :5.750   Median :4.900  
 Mean   :5.442   Mean   :4.010   Mean   :5.805   Mean   :5.123  
 3rd Qu.:6.325   3rd Qu.:4.800   3rd Qu.:6.800   3rd Qu.:5.800  
 Max.   :7.800   Max.   :6.500   Max.   :8.400   Max.   :8.200  
   ComPricing      WartyClaim      OrdBilling       DelSpeed    
 Min.   :3.700   Min.   :4.100   Min.   :2.000   Min.   :1.600  
 1st Qu.:5.875   1st Qu.:5.400   1st Qu.:3.700   1st Qu.:3.400  
 Median :7.100   Median :6.100   Median :4.400   Median :3.900  
 Mean   :6.974   Mean   :6.043   Mean   :4.278   Mean   :3.886  
 3rd Qu.:8.400   3rd Qu.:6.600   3rd Qu.:4.800   3rd Qu.:4.425  
 Max.   :9.900   Max.   :8.100   Max.   :6.700   Max.   :5.500  
  Satisfaction  
 Min.   :4.700  
 1st Qu.:6.000  
 Median :7.050  
 Mean   :6.918  
 3rd Qu.:7.625  
 Max.   :9.900
```

- **dim()** – This function is used to **give out the dimensions** of the Dataset

```
> dim(Market)
[1] 100  13
```

- **colnames()** – This function is used to **give out all the column names** in the Dataset.

```
> colnames(Market)
 [1] "ID"           "ProdQual"     "Ecom"         "TechSup"
 [5] "CompRes"      "Advertising"  "ProdLine"     "SalesFImage"
 [9] "ComPricing"   "WartyClaim"   "OrdBilling"   "DelSpeed"
[13] "Satisfaction"
```

- **str()** – This function is used to give the **class of all the variables** present in the Dataset.

```
> str(Market)
'data.frame':   100 obs. of  13 variables:
 $ ID          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ ProdQual    : num  8.5 8.2 9.2 6.4 9 6.5 6.9 6.2 5.8 6.4 ...
 $ Ecom        : num  3.9 2.7 3.4 3.3 3.4 2.8 3.7 3.3 3.6 4.5
 ...
 $ TechSup     : num  2.5 5.1 5.6 7 5.2 3.1 5 3.9 5.1 5.1 ...
 $ CompRes     : num  5.9 7.2 5.6 3.7 4.6 4.1 2.6 4.8 6.7 6.1
 ...
 $ Advertising : num  4.8 3.4 5.4 4.7 2.2 4 2.1 4.6 3.7 4.7 ...
 $ ProdLine    : num  4.9 7.9 7.4 4.7 6 4.3 2.3 3.6 5.9 5.7 ...
 $ SalesFImage : num  6 3.1 5.8 4.5 4.5 3.7 5.4 5.1 5.8 5.7 ...
 $ ComPricing  : num  6.8 5.3 4.5 8.8 6.8 8.5 8.9 6.9 9.3 8.4
 ...
 $ WartyClaim  : num  4.7 5.5 6.2 7 6.1 5.1 4.8 5.4 5.9 5.4 ...
 $ OrdBilling  : num  5 3.9 5.4 4.3 4.5 3.6 2.1 4.3 4.4 4.1 ...
 $ DelSpeed    : num  3.7 4.9 4.5 3 3.5 3.3 2 3.7 4.6 4.4 ...
 $ Satisfaction: num  8.2 5.7 8.9 4.8 7.1 4.7 5.7 6.3 7 5.5 ...
```

- **head() –** This function is used to give out **the top values** in the dataset. The default is **6.**

```
> head(Market)
  ID ProdQual Ecom TechSup CompRes Advertising ProdLine
1  1      8.5  3.9     2.5     5.9         4.8      4.9
2  2      8.2  2.7     5.1     7.2         3.4      7.9
3  3      9.2  3.4     5.6     5.6         5.4      7.4
4  4      6.4  3.3     7.0     3.7         4.7      4.7
5  5      9.0  3.4     5.2     4.6         2.2      6.0
6  6      6.5  2.8     3.1     4.1         4.0      4.3
  SalesFImage ComPricing WartyClaim OrdBilling DelSpeed
1         6.0        6.8        4.7         5.0      3.7
2         3.1        5.3        5.5         3.9      4.9
3         5.8        4.5        6.2         5.4      4.5
4         4.5        8.8        7.0         4.3      3.0
5         4.5        6.8        6.1         4.5      3.5
6         3.7        8.5        5.1         3.6      3.3
  Satisfaction
1          8.2
2          5.7
3          8.9
4          4.8
5          7.1
6          4.7
```

- **tail() –** This function is used to give out **the bottom values** in the dataset. The default is **6.**

```
> tail(Market)
     ID ProdQual Ecom TechSup CompRes Advertising ProdLine
95   95      9.3  3.8     4.0     4.6         4.7      6.4
96   96      8.6  4.8     5.6     5.3         2.3      6.0
97   97      7.4  3.4     2.6     5.0         4.1      4.4
98   98      8.7  3.2     3.3     3.2         3.1      6.1
99   99      7.8  4.9     5.8     5.3         5.2      5.3
100 100      7.9  3.0     4.4     5.1         5.9      4.2
    SalesFImage ComPricing WartyClaim OrdBilling DelSpeed
95          5.5        7.4        5.3        3.6      3.4
96          5.7        6.7        5.8        4.9      3.6
97          4.8        7.2        4.5        4.2      3.7
98          2.9        5.6        5.0        3.1      2.5
99          7.1        7.9        6.0        4.3      3.9
100         4.8        9.7        5.7        3.4      3.5
    Satisfaction
95           7.7
96           7.3
97           6.3
98           5.4
99           6.4
100          6.4
```

Inferences:

i. According to the summary function, we see that all the variables **lied between 0 – 10 range** except the **ID variable** which had **numbers from 1 to 100**. This indicates that each variable in the dataset has a **rating according to each of the product ID** and the rating scale for each variable goes **from 0 to 10**

ii. The dimensions of the Dataset according to the above results are **100 Rows and 13 Columns.**

iii. The 100 Rows depict that there are **100 product IDs** for which the **ratings in each of the Market segments were taken.**

iv. The 13 Columns depict there are **Thirteen Variables** including the **ID column** that contain ratings.

v. The **colnames** function shows the names of all the columns present in the dataset. Out of the **13 variables,** we can see that the column **Satisfaction,** doesn't come under any of the market segments like the rest of the **12 variables.** Therefore only **12** variables represent the **Market Segments.**

vi. The **str** function shows that all the variables are **numeric** except the **ID** variable which is **Integer** type.

vii. The **head** and **tail** functions show that the order of the variable is in ascending order according to the **ID number.**

d. Dataset Transformation

- The variable names in the given dataset are given in their short forms and must be changed to their full names so that no confusion arises when the Analysis is done. The column names in the dataset can be changed by using the function **colnames().** The full names of each variables are already provided with the dataset.

- Since **Customer ID** is an insignificant column in the dataset which will not be needed for the analysis of the dataset, we must eliminate it by **slicing the column from the original dataset**.

- The variable **Customer Satisfaction** is a dependent variable and not part of the market segmentation variables in the dataset. So another dataset called **Market1,** must be created by slicing the **Customer Satisfaction** column.

The **R Code** for the above transformations are as follows:

```
### Changing the Variable Names according to the given list ##
colnames(Market) = c("Product ID","Product Quality",
                     "E-Commerce","Technical Support",
                     "Complaint Resolution","Advertising",
                     "Product Line","Salesforce Image",
                     "Competitive Pricing","Warranty & Claims",
                     "Order & Billing","Delivery Speed",
                     "Customer Satisfaction")
```

```
> colnames(Market)
 [1] "Product ID"          "Product Quality"     "E-Commerce"          "Technical Support"    "Complaint
Resolution"
 [6] "Advertising"         "Product Line"        "Salesforce Image"    "Competitive Pricing"  "Warranty
& Claims"
[11] "Order & Billing"     "Delivery Speed"      "Customer Satisfaction"
```

```
### Removing the ID variable ###
Market = Market[,-1]
###Creatin a new dataset Market1 without the Satisfaction Variable###
Market1 = Market[,-12]
```

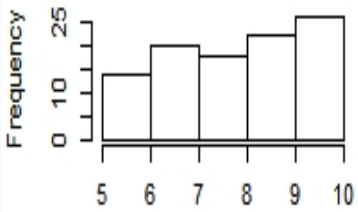
e. Uni-Variate Analysis

There are **11 Independent variables** and **one Dependent variable.** So the Uni-Variate Analysis must be done separately for the Independent and the Dependent variables.

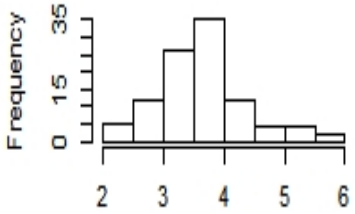
- For the independent variables, the analysis can be done with the help of **Histogram** and **Boxplot.** Histogram helps us in identifying the **distribution of the values** and Boxplot gives us an idea on **outliers**. The histogram for multiple variables can be got by using the **hist.data.frame()** from the library **hmisc.** It gives us the histograms of all the variables at once. The Boxplot for all the variables can be got at once by using **melt()** function in the **ggplot() function** from the library **reshape.**

  Histogram:

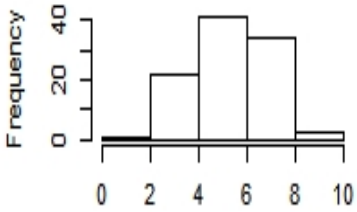
```
### Uni-Variate Analysis ###
hist.data.frame(Market1)
```



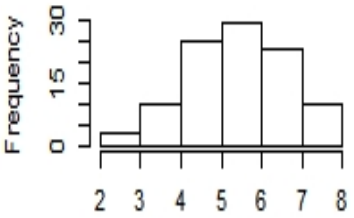
Product Quality
n:100 m:0



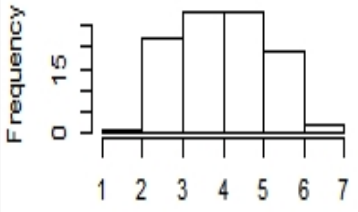
E – Commerce
n:100 m:0



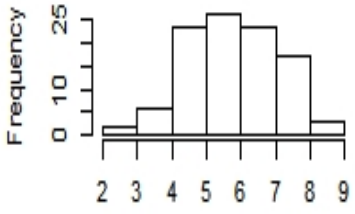
Technical Support
n:100 m:0



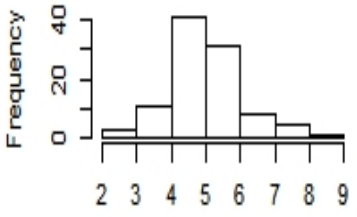
Complaint Resolution
n:100 m:0



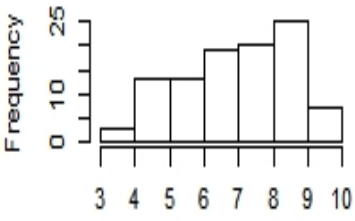
Advertising
n:100 m:0



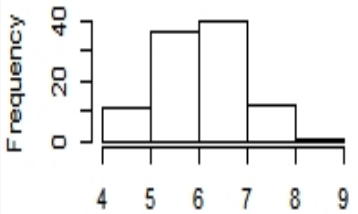
Product Line
n:100 m:0



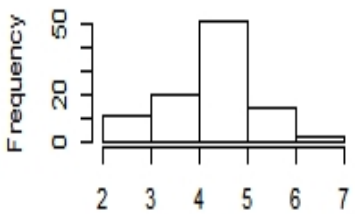
Salesforce Image
n:100 m:0



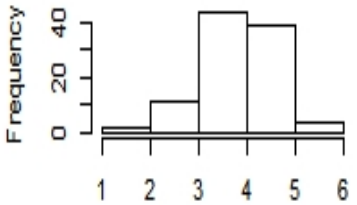
Competitive Pricing
n:100 m:0



Warranty & Claims
n:100 m:0



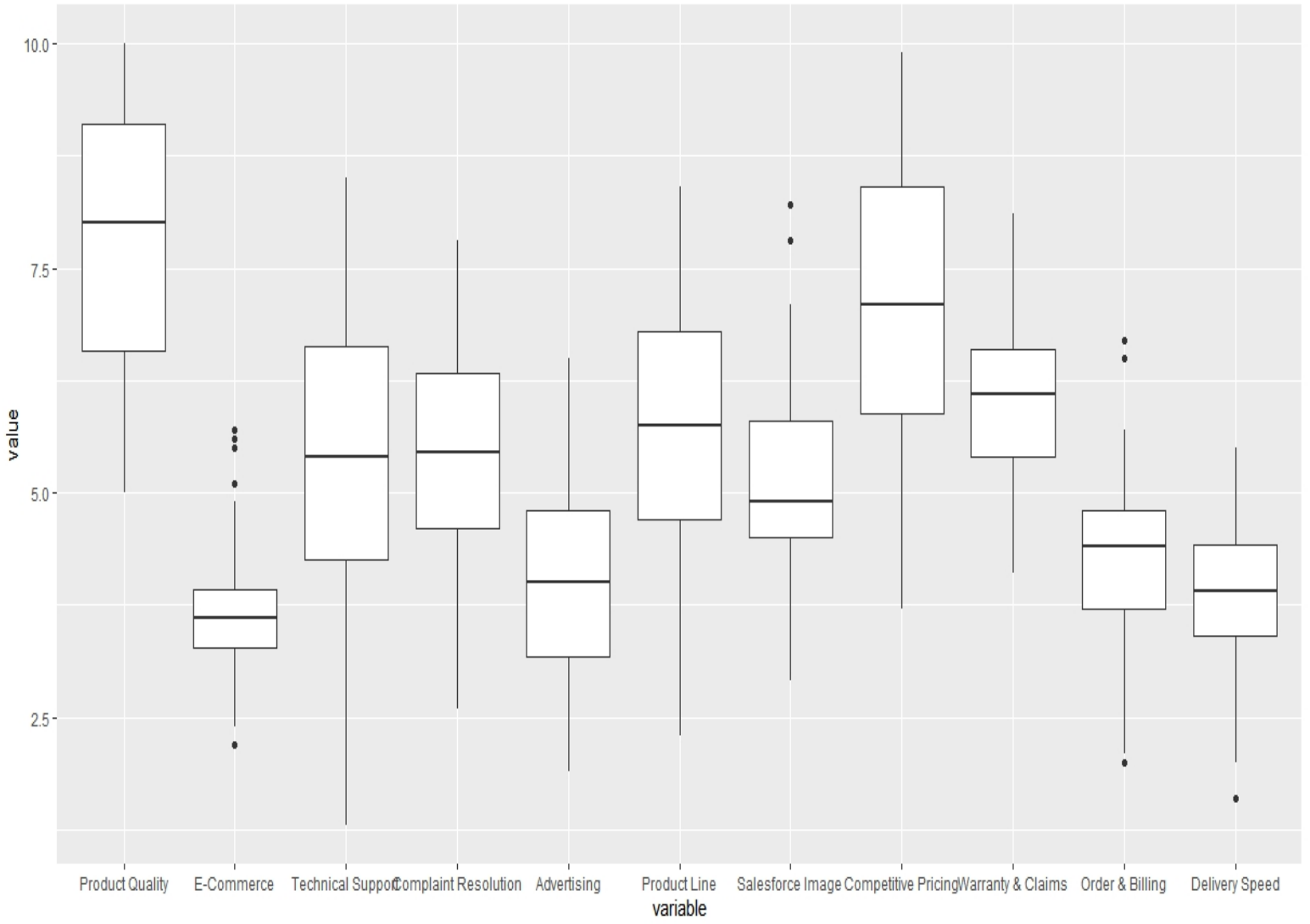
Order & Billing
n:100 m:0



Delivery Speed
n:100 m:0

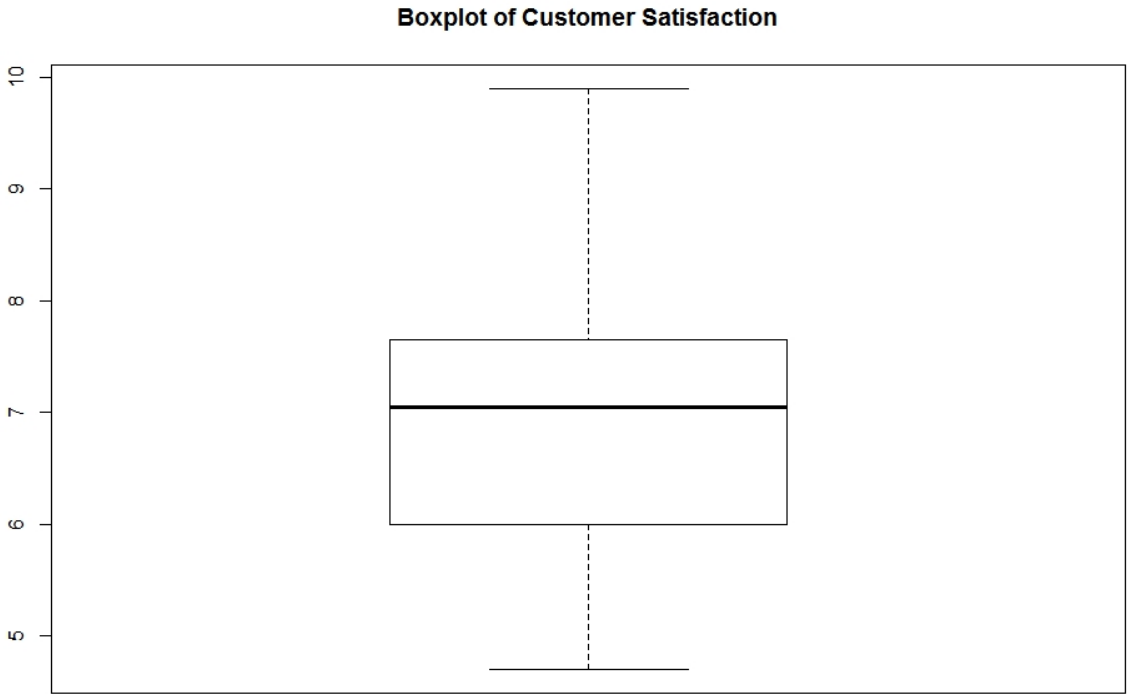
Boxplot:

```
ggplot(melt(Market[,-12]), aes(variable, value)) + geom_boxplot()
```

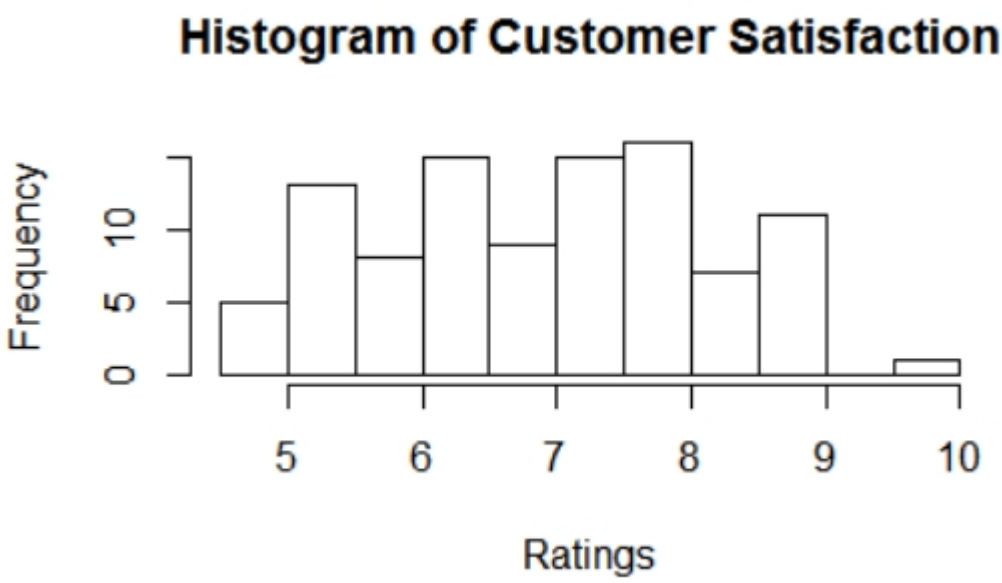


- The **Dependent** variable is the **Customer Satisfaction**. The **Histogram** for this variable can be got through the function **hist()** and the **Boxplot** can be got from the function **boxplot().**

```
boxplot(Market$`Customer Satisfaction`,main = "Boxplot of Customer Satisfaction")
```




```
hist(Market$`Customer Satisfaction`,
     main = "Histogram of Customer Satisfaction",
     xlab = "Ratings")
```



Inferences:

i. From the **Histograms of the Independent variables**, we can see that except the **Product Quality,** rest of all the variables, even though skewed, follow a **Bell-Shaped Curve** or **Normal Distribution.**

ii. The variables **Salesforce Image, Advertising and Order & Billing,** are the only variables that follow almost a perfect **Bell shaped curve** without any skew.

iii. The rest of the variables, **Technical Support, Complaint Resolution, Competitive Pricing, and Delivery Speed** are all **Right skewed**.

iv. **Warranty and Claims and E-Commerce** are all **Left-Skewed.**

v. From the **Boxplots,** we can see that the **Delivery Speed, Order and Billing, Salesforce Image and E-Commerce** are the variables containing outliers.

vi. The **Highest** ratings belong to the Product Quality and the **lowest** ratings belong to the E-Commerce.

vii. From the **Histogram** of the **Customer Satisfaction,** we can see that the distribution is **not Normal distribution** and it has some extreme values but **not outlier.**

viii. From the **Boxplot** of the **Customer Satisfaction,** we can see that the variable contains no **Outliers.**

f.Bi-Variate Analysis

The Bi- Variate Analysis must be done using a Categorical variable such as a Happiness scale. Therefore, we can create a categorical variable by using the **cut()** function. This Categorical variable called **Happiness** which has **3 Categories** namely **Sad, Not Happy and Happy.** This Categorical variable can be used to analyse the variable with high ratings, the variable with lowest ratings and the **Customer Satisfaction**. The **table()** function helps in giving a tabular representation of categorical segmentation. The **fill** argument in the **qplot()** function helps in giving out plots corresponding to the Categorical variable.

Customer Satisfaction:

```
### Bi - Variate Analysis ###
### CUSTOMER SATISFACTION ###
Happiness = cut(Market$`Customer Satisfaction`,3,labels = c("Sad","Not Happy","Happy"))
qplot(Happiness,fill = Happiness)
tab = table(Happiness)
View(prop.table(tab)*100)
```

	CUSTOMER.SATISFACTION	Freq
1	Sad	41
2	Not Happy	40
3	Happy	19

Product Quality:

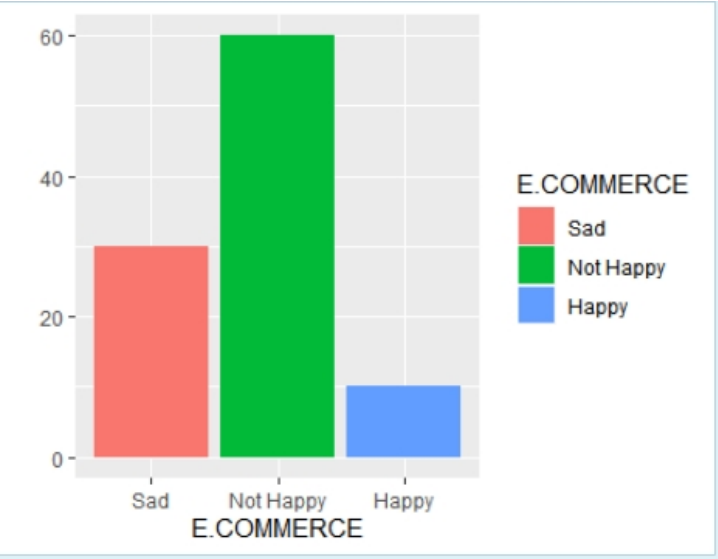
```
### PRODUCT QUALITY ####
Happiness1 = cut(Market$`Product Quality`,3,labels = c("Sad","Not Happy","Happy"))
tab1 = table(Happiness1)
View(prop.table(tab1)*100)
qplot(Happiness1,fill = Happiness1)
```

	PRODUCT.QUALITY	Freq
1	Sad	26
2	Not Happy	31
3	Happy	43

E-Commerce:

```
### E-COMMERCE ###
E-COMMERCE = cut(Market$`E-Commerce`,3,labels = c("Sad","Not Happy","Happy"))
tab2 = table(E-COMMERCE)
prop.table(tab2)*100
qplot(E-COMMERCE,fill = E-COMMERCE)
```

	E.COMMERCE	Freq
1	Sad	30
2	Not Happy	60
3	Happy	10



Inferences:

i. In the variable **Customer Satisfaction,** we can see that 41% are **Sad,** 40% are **Not Happy** and only **19% are Happy.** Which means only 19% of the customers are happy with their products.

ii. In the variable **Product Quality,** we can see that 26% are **Sad,** 31% are **Not Happy** and 43% are **Happy.** We can see that even in the variable with the highest ratings, not the majority of them are **Happy.**

iii. In the variable **E-Commerce,** we can see that 30% are **Sad,** 60% are **Not Happy** and 10% are **Happy.** We can see that majority are **not happy** with this segment. It is due to this variable which contributes to the **lower total ratings of all the customers.**

g. Missing Value Treatment.

The missing values in the dataset can be found by using the using the **function is.na()** and applying the **sum() function** to it to give out the number of Missing Values.

```
> sum(is.na(Market))
[1] 0
```

Inferences:

We can see that there are no missing values in the dataset.

h. Outlier Identification.

We found out in the **Uni-Variate** analysis that the variables namely, **E-Commerce, Salesforce Image, Order and Billing, and Delivery Speed** have outliers. Since there is no function defined for finding out the **Outliers**, we must create a **custom function** using the Outlier rule which states that **all that values that are greater than Q3 + 1.5×IQR and lesser than Q1 – 1.5×IQR.** This function is called **Outlier2** which when given the column number as output gives out the number of Outliers. The Column numbers for the variables **E-Commerce, Salesforce Image, Order and Billing, and Delivery Speed** are **2,7,10 and 11**.

```
### Outlier identification ###
outlier2 = function(i)
{
  IQ = IQR(Market[,i])
  z = quantile(Market[,i])
  z= as.data.frame(z)
  colnames(z) = as.factor(colnames(z))
  q1 = z[2,1]
  q3 = z[4,1]

  Market2 = Market
  subset1 = Market2[Market2[,i] < q1 - 1.5*IQ,]
  lo = nrow(subset1)
  subset2 = Market2[Market2[,i] > q3 + 1.5*IQ,]
  hi = nrow(subset2)
  Outliers = lo + hi
  print(colnames(Market[i]))
  Outliers
}
```

```
> ### outlier identification ###
> outlier2(2)
[1] "E-Commerce"
[1] 7
> outlier2(7)
[1] "Salesforce Image"
[1] 3
> outlier2(10)
[1] "Order & Billing"
[1] 4
> outlier2(11)
[1] "Delivery Speed"
[1] 1
```

Inferences:

i. The variable **E-Commerce** has the highest number of outliers which is **7**

ii. The variable **Salesforce Image** has **3** Outliers.

iii. The variable **Order and Billing** has **4** Outliers.

iv. The variable **Delivery Speed** has the least number of outliers which is only **1.**

3. Solutions to the questions asked regarding the project:

Q1. Perform exploratory data analysis on the dataset. Showcase some charts, graphs. Check for outliers and missing values.

Sol: The Exploratory Data Analysis was done on the dataset as part of the initial steps in the project. The solution contains analysis on the datasets with charts and graphs, treatment of outliers and missing values as asked in the question.

Q2. Is there evidence of multicollinearity? Showcase your analysis.

Sol: **Multicollinearity is a condition where the variables in a regression thought independent, have Correlation between them**. **The Independence of a certain variable** from the rest of the variables depends on the **Correlation** that **exists between them**. The Correlation **may exist** between the **dependent and independent variable** but **not** between the **independent variables.** This poses a **problem in the Regression.**

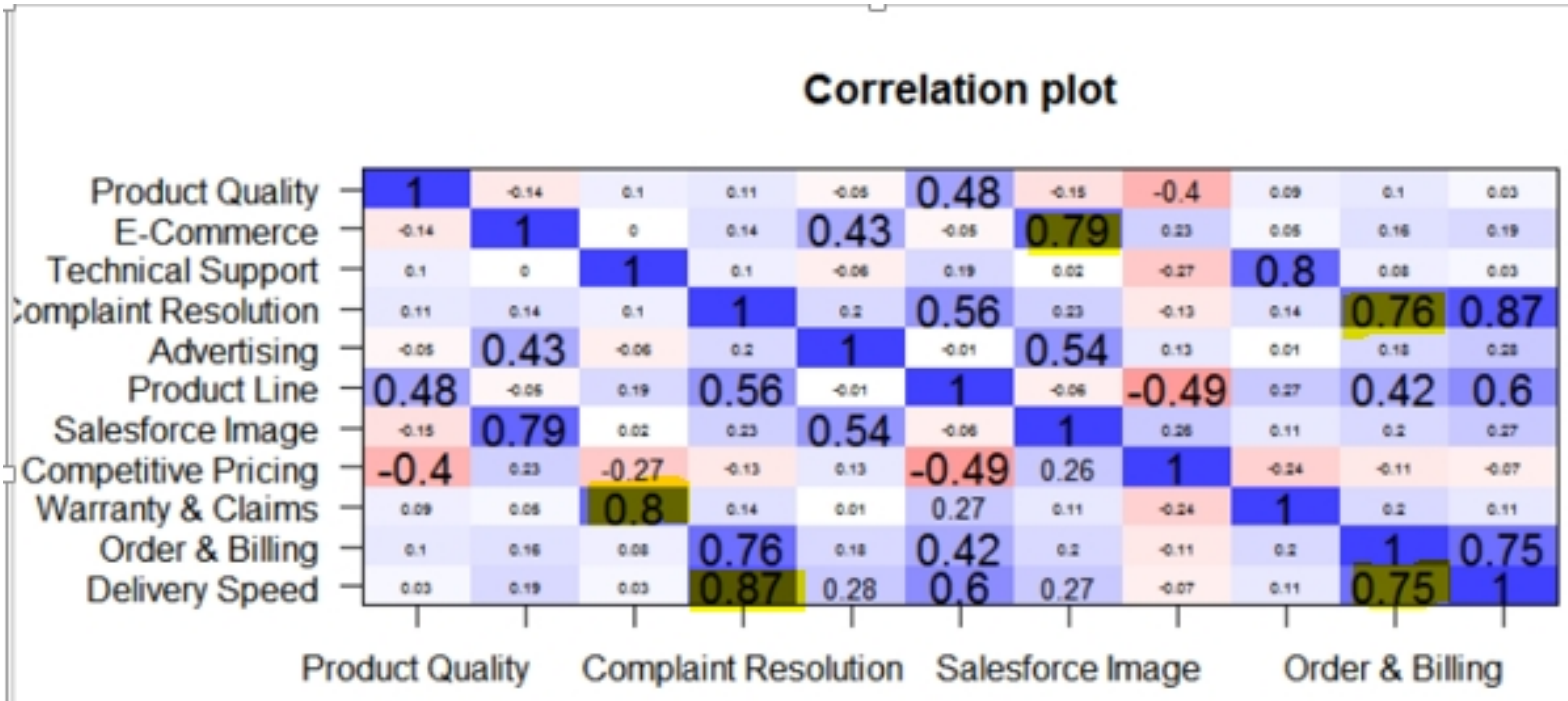
The question on whether **Multicollinearity exists** between the independent variables or not **cannot be determined by just one method**. Just because we have been **able to prove** that Multicollinearity exists or not in one method **doesn't confirm it**. It is purely on the i**nterpretation of the results from the various methods** one can infer and confirm whether the

multicollinearity exists or not. The following methods were applied to the dataset to check for Multicollinearity.

- **<u>Checking the Correlation Graph and Matrix.</u>**

  The basic method for checking Multicollinearity is to interpret the results from the **Correlation Graph and Matrix**. The correlation Matrix can be got by using the function **cor()** on the dataset. The Correlation graph can be got by using the function **cor.plot()** function from the **psych**() library. If the argument **numbers = TRUE,** we can get the values in the same Correlation Graph. We can say that a significant correlation exists if the **value of correlation is more than 0.7 or less than -0.7.**

```
### Question 2 (Collinearity) ####
cor.plot(Market1,numbers = TRUE)
View(cor(Market1))
```



	Product Quality	E-Commerce	Technical Support	Complaint Resolution	Advertising	Product Line	Salesforce Image	Competitive Pricing	Warranty & Claims	Order & Billing	Delivery Speed
Product Quality	1.00000000	-0.1371632174	0.0956004542	0.1063700	-0.05347313	0.47749341	-0.15181287	-0.40128188	0.08831231	0.10430307	0.02771800
E-Commerce	-0.13716322	1.0000000000	0.0008667887	0.1401793	0.42989071	-0.05268784	0.79154371	0.22946240	0.05189819	0.15614733	0.19163607
Technical Support	0.09560045	0.0008667887	1.0000000000	0.0966566	-0.06287007	0.19262546	0.01699054	-0.27078668	0.79716793	0.08010182	0.02544069
Complaint Resolution	0.10637000	0.1401792611	0.0966565978	1.0000000	0.19691685	0.56141695	0.22975176	-0.12795425	0.14040830	0.75686859	0.86509170
Advertising	-0.05347313	0.4298907110	-0.0628700668	0.1969168	1.00000000	-0.01155082	0.54220366	0.13421689	0.01079207	0.18423559	0.27586308
Product Line	0.47749341	-0.0526878383	0.1926254565	0.5614170	-0.01155082	1.00000000	-0.06131553	-0.49494840	0.27307753	0.42440825	0.60185021
Salesforce Image	-0.15181287	0.7915437115	0.0169905395	0.2297518	0.54220366	-0.06131553	1.00000000	0.26459655	0.10745534	0.19512741	0.27155126
Competitive Pricing	-0.40128188	0.2294624014	-0.2707866821	-0.1279543	0.13421689	-0.49494840	0.26459655	1.00000000	-0.24498605	-0.11456703	-0.07287173
Warranty & Claims	0.08831231	0.0518981915	0.7971679258	0.1404083	0.01079207	0.27307753	0.10745534	-0.24498605	1.00000000	0.19706512	0.10939460
Order & Billing	0.10430307	0.1561473316	0.0801018246	0.7568686	0.18423559	0.42440825	0.19512741	-0.11456703	0.19706512	1.00000000	0.75100307
Delivery Speed	0.02771800	0.1916360683	0.0254406935	0.8650917	0.27586308	0.60185021	0.27155126	-0.07287173	0.10939460	0.75100307	1.00000000

Inferences:

From the above chart and matrix, we can see that there are some **Correlation values above 0.7** which indicate the existence of Multicollinearity.

- **Checking the Eigen values.**

  Eigen values are the values corresponding to the each variable when the correlation values of that variable with rest of all the variable are squared and added. The Eigen values can be got by using the function **Eigen** on the dataset. If we find that at least one of the Eigen values is **zero or near to zero,** we can say to some extent that there is an evidence of **Multicollinearity.**

```
> Eigen = eigen(cor(Market1))
> Eigen$values
 [1] 3.42697133 2.55089671 1.69097648 1.08655606 0.60942409 0.55188378 0.40151815 0.24695154 0.20355327
[10] 0.13284158 0.09842702
```

Inferences:

From the above **Eigen values,** we can see that one of the values is close to **zero** indicating Multicollinearity

- **Analysis of Multiple Regression.**

  Analysis of **Multiple Regression model** can also depict the signs of **Multicollinearity**. Searching for inconsistencies, low significance of the variable slope, etc. can be signs of **Multicollinearity**. We can build a model by using the **lm()**

function and use the function **summary()** to give out the information about the model.

```
### Checking the Regression model ###
p = lm(`Customer Satisfaction`~., data = Market)
summary(p)
```

```
Call:
lm(formula = `Customer Satisfaction` ~ ., data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-1.43005 -0.31165  0.07621  0.37190  0.90120

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          -0.66961    0.81233  -0.824  0.41199
`Product Quality`     0.37137    0.05177   7.173 2.18e-10 ***
`E-Commerce`         -0.44056    0.13396  -3.289  0.00145 **
`Technical Support`   0.03299    0.06372   0.518  0.60591
`Complaint Resolution` 0.16703   0.10173   1.642  0.10416
Advertising          -0.02602    0.06161  -0.422  0.67382
`Product Line`        0.14034    0.08025   1.749  0.08384 .
`Salesforce Image`    0.80611    0.09775   8.247 1.45e-12 ***
`Competitive Pricing` -0.03853   0.04677  -0.824  0.41235
`Warranty & Claims`  -0.10298    0.12330  -0.835  0.40587
`Order & Billing`     0.14635    0.10367   1.412  0.16160
`Delivery Speed`      0.16570    0.19644   0.844  0.40124
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5623 on 88 degrees of freedom
Multiple R-squared:  0.8021,    Adjusted R-squared:  0.7774
F-statistic: 32.43 on 11 and 88 DF,  p-value: < 2.2e-16
```

Inferences:

From the above **Model,** we can see that the **Slope Estimate** has negative values which is wrong because increase in any of the variable's ratings will always increase the ratings of **Customer Satisfaction**. The **P-Values** for some of the variables are really **insignificant** which indicate that the **Slope Intercept** found for this value will not help in being a **Good fit** for Regression even though the **R-Squared value is high.** These affected values are the signs of **Multicollinearity.** The values get affected because of the **Multicollinearity between the variables.**

- **Checking the VIF values.**

VIF(Variation Inflation Factors) are the factors which help in comparison of the variance explained by a dependent variable in a Multiple Regression Model by the variance explained by a dependent variable when placed in a Single Regression Model. The VIF values can be obtained by applying the **vif()** function from the **car** library. If most of the values are above 5, it can indicate signs of **Multicollinearity.**

```
> ### Checking the VIF values ###
> vifmatrix = vif(p)
> summary(vifmatrix)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.492   2.642   2.887   3.124   3.363   5.696
```

Inferences:

In the above result, we can see that one of the values has crossed **Five**. Even though it is not major sign of **Multicollinearity,** we can assume that Multicollinearity, even though less in this case, **exists.**

Conclusion:

In all the above methods, we can see results of every method show signs of **Multicollinearity** by which we can conclude that the ***Multicollinearity exists in the dataset and the Independent variables are not truly independent of each other.***

Q3. Perform simple linear regression for the dependent variable with every independent variable.

Sol: **Simple Linear Regression** is the process of creating a **linear equation** between the **Dependent Variable** and a **Single**

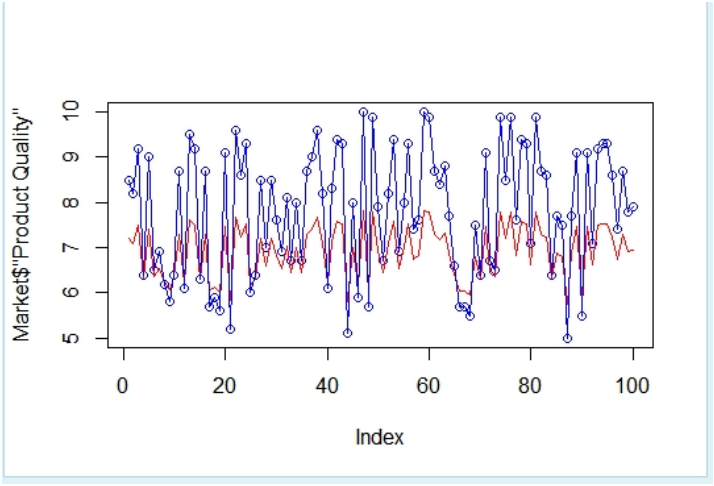
Dependent Variable which establishing a **necessary relationship.**

There is only one Dependent variable which is **Customer Satisfaction** and there are **11 Independent variables.** We can use the **lm()** function to create an equation. The **Regression predicted values** can be got by using the function **predict().** The functions **plot()** and **lines()** give out graphs that show how well the Regression Line fits the actual values. Using the **confint()** to get the **Maximum and Minimum values** that regression can take and this can be represented graphically by using the **qplot()** function.

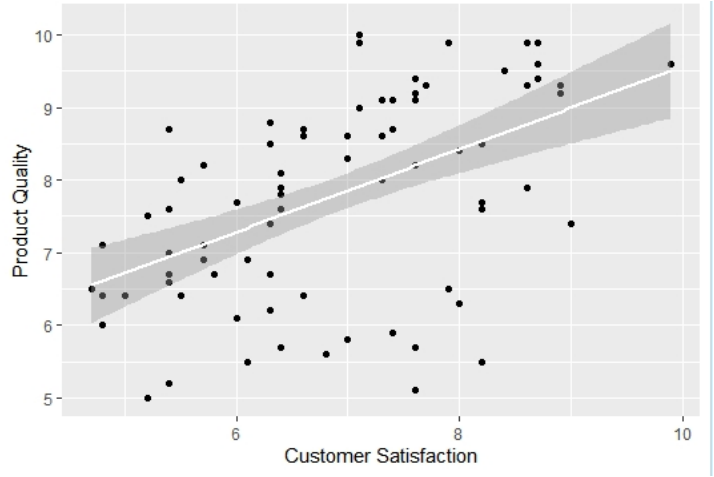
- ## Product Quality:

```
### Question 3 ###

### Product Quality ###
slm1 = lm(`Customer Satisfaction` ~ `Product Quality`,data = Market)
summary(slm1)
p1 = predict(slm1)
p1
plot(p1,col = "Red")
plot(Market$'Product Quality',col = "Blue")
lines(p1,col = "Red")
lines(Market$`Product Quality`,col = "Blue")
confint(slm1)
ggplot(Market,aes(x = `Customer Satisfaction`,y = `Product Quality`))+geom_point(col = "Black")+stat_smooth(method = "lm",col = "White")
```

```
Call:
lm(formula = `Customer Satisfaction` ~ `Product Quality`, data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-1.88746 -0.72711 -0.01577  0.85641  2.25220

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       3.67593    0.59765   6.151 1.68e-08 ***
`Product Quality` 0.41512    0.07534   5.510 2.90e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.047 on 98 degrees of freedom
Multiple R-squared:  0.2365,    Adjusted R-squared:  0.2287
F-statistic: 30.36 on 1 and 98 DF,  p-value: 2.901e-07
```




```
> confint(slm1)
                      2.5 %     97.5 %
(Intercept)       2.4899022 4.8619486
`Product Quality` 0.2656059 0.5646309
```



Inferences:

The equation from the above results can be as follows:

Customer* Satisfaction = *(0.41512)* ×*Product Quality* + *3.67593

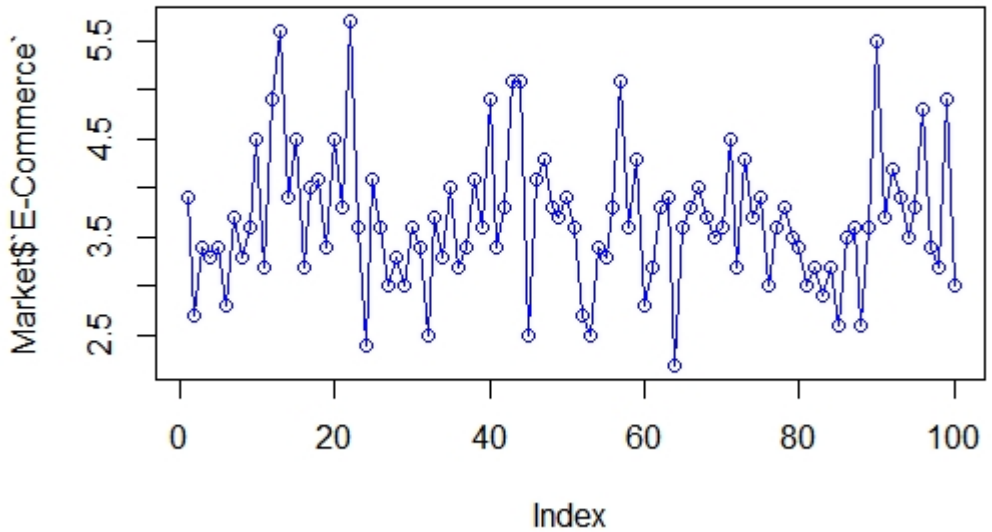
From the Plots, we can see that this Regression Equation is not significant and is **not a fit** equation as it deviates a lot from the actual values. This is also evident in its **low R-Squared value of 0.23.**

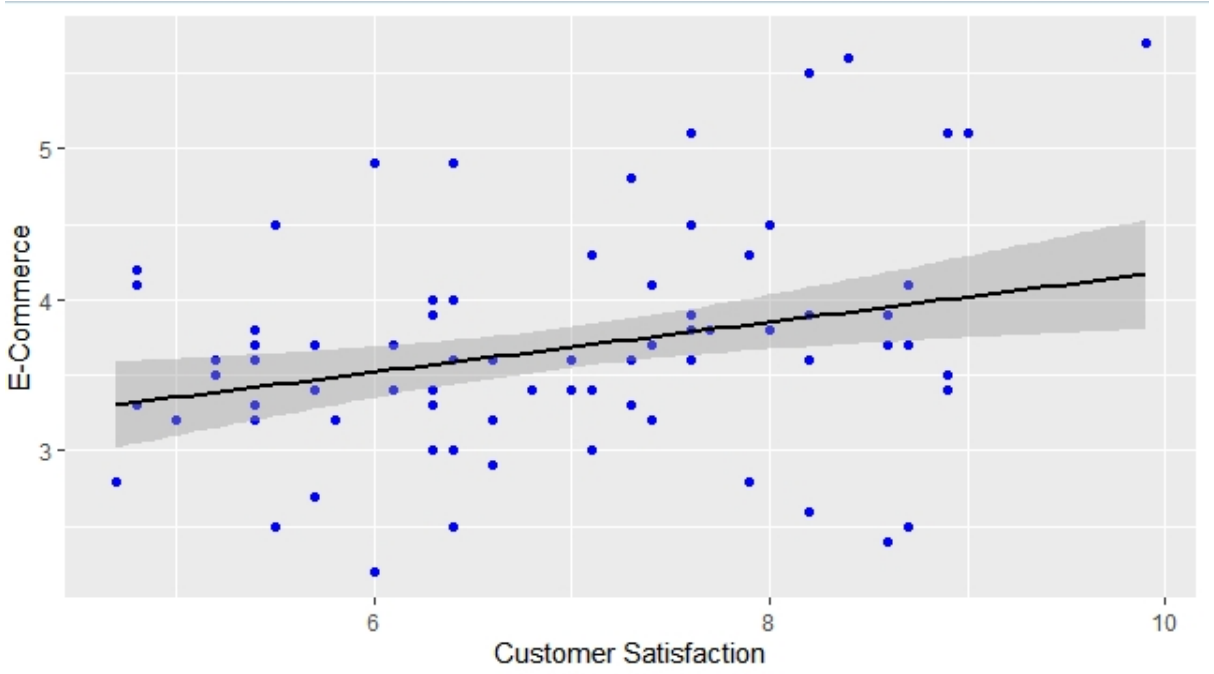
- **E-Commerce:**

```r
### E-Commerce ###
slm2 = lm(`Customer Satisfaction` ~ `E-Commerce`,data = Market)
summary(slm2)
p2 = predict(slm2)
plot(p2,col = "Red")
plot(Market$`E-Commerce`,col = "Blue")
lines(p2,col = "Red")
lines(Market$`E-Commerce`,col = "Blue")
confint(slm2)
ggplot(Market,aes(x = `Customer Satisfaction`,y = `E-Commerce`))+geom_point(col = "Blue")+stat_smooth(method = "lm",col = "Black")
```

```
Call:
lm(formula = `Customer Satisfaction` ~ `E-Commerce`, data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-2.37200 -0.78971  0.04959  0.68085  2.34580

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.1516     0.6161   8.361 4.28e-13 ***
`E-Commerce`   0.4811     0.1649   2.918  0.00437 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.149 on 98 degrees of freedom
Multiple R-squared:  0.07994,   Adjusted R-squared:  0.07056
F-statistic: 8.515 on 1 and 98 DF,  p-value: 0.004368
```

```
                  2.5 %    97.5 %
(Intercept)   3.9288501 6.3742852
`E-Commerce` 0.1539118 0.8081973
```





Inferences:

The equation from the above results can be as follows:

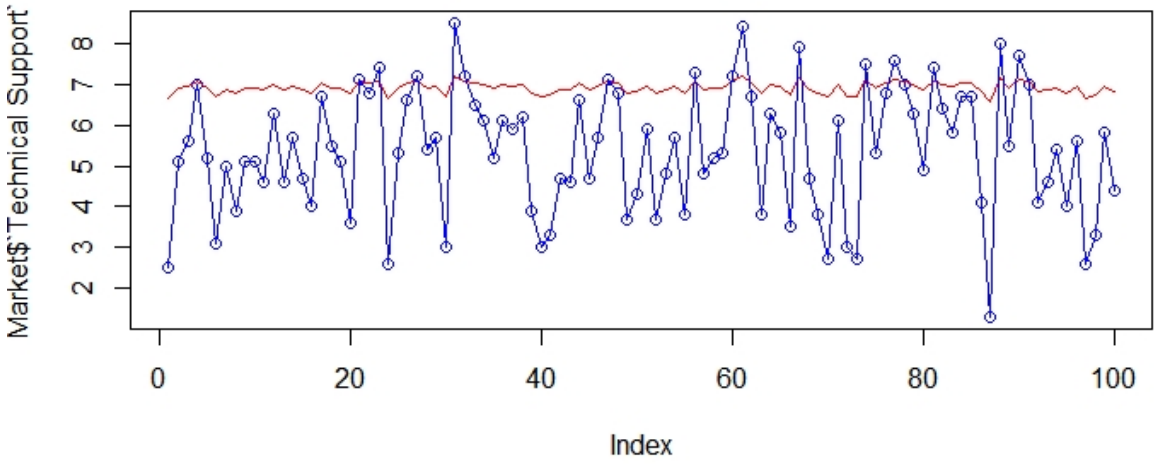
Customer* Satisfaction = *(0.4811)* ×*E-Commerce* + *5.1516

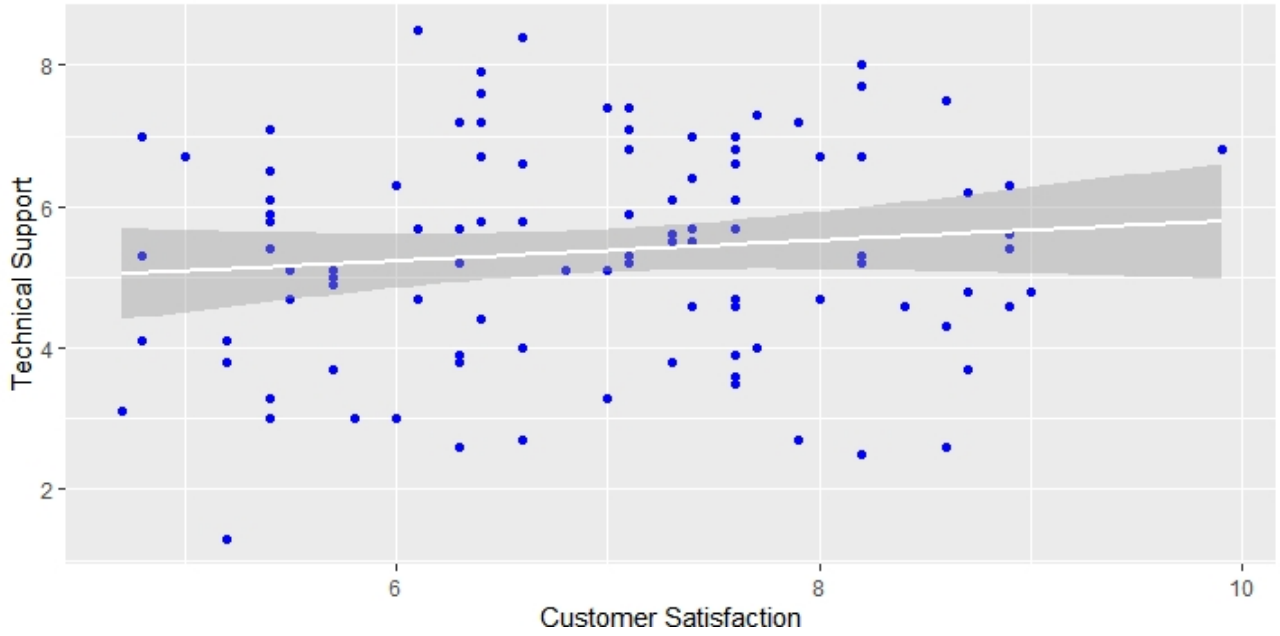
From the Plots, we can see that this Regression Equation is not significant and is **not a fit** equation as it deviates a lot from the actual values. This is also evident in its **low R-Squared value of 0.07**

- ## Technical Support:

```
### Technical Support ###
slm3 = lm(`Customer Satisfaction` ~ `Technical Support`,data = Market)
summary(slm3)
p3 = predict(slm3)
plot(p3,col = "Red")
plot(Market$`Technical Support`,col = "Blue")
lines(p3,col = "Red")
lines(Market$`Technical Support`,col = "Blue")
confint(slm3)
ggplot(Market,aes(x = `Customer Satisfaction`,y = `Technical Support`))+geom_point(col = "Blue")+stat_smooth(method = "lm",col = "White")
```

```
Call:
lm(formula = `Customer Satisfaction` ~ `Technical Support`, data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-2.26136 -0.93297  0.04302  0.82501  2.85617

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         6.44757    0.43592  14.791   <2e-16 ***
`Technical Support`  0.08768    0.07817   1.122    0.265
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.19 on 98 degrees of freedom
Multiple R-squared:  0.01268,   Adjusted R-squared:  0.002603
F-statistic: 1.258 on 1 and 98 DF,  p-value: 0.2647

                           2.5 %    97.5 %
(Intercept)          5.58250018 7.3126423
`Technical Support` -0.06743136 0.2428009
```





Inferences:

The equation from the above results can be as follows:

Customer* Satisfaction = *(0.08768) ×Technical Support + 6.44757

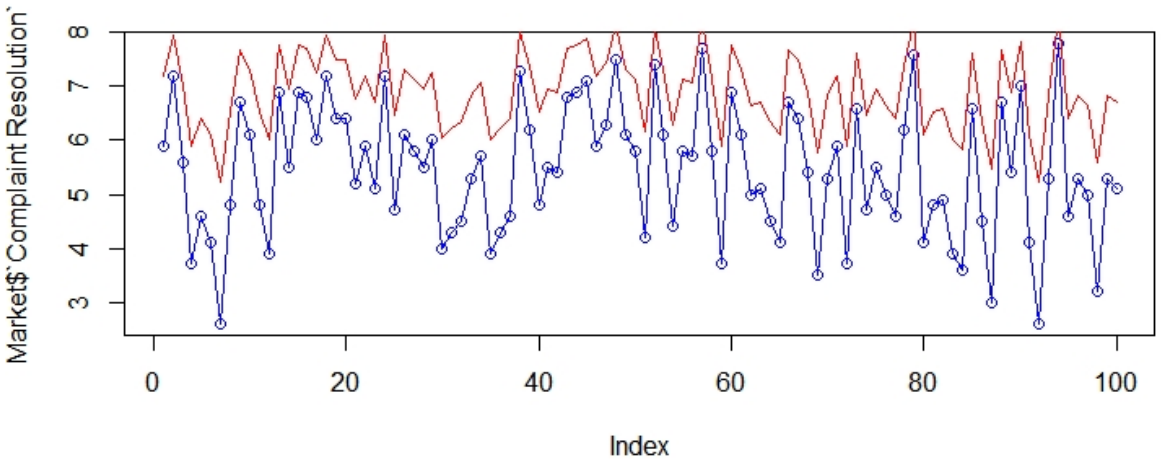
From the Plots, we can see that this Regression Equation is not significant and is **not a fit** equation as it deviates a lot from the actual values. This is also evident in its **low R-Squared value of 0.01**

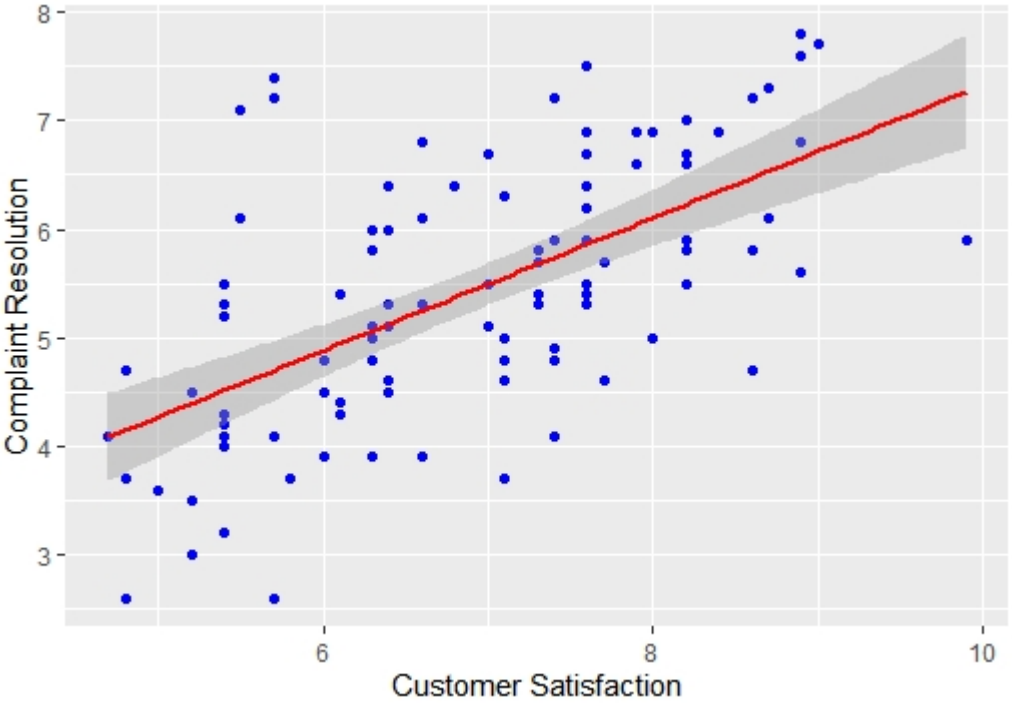
- ## Complaint Resolution:

```
### Complaint Resolution ###
slm4 = lm(`Customer Satisfaction` ~ `Complaint Resolution`,data = Market)
summary(slm4)
p4 = predict(slm4)
plot(p4,col = "Red")
plot(Market$`Complaint Resolution`,col = "Blue")
lines(p4,col = "Red")
lines(Market$`Complaint Resolution`,col = "Blue")
confint(slm4)
ggplot(Market,aes(x = `Customer Satisfaction`,y = `Complaint Resolution`))+geom_point(col = "Blue")+stat_smooth(method = "lm",col = "Red")
```

```
Call:
lm(formula = `Customer Satisfaction` ~ `Complaint Resolution`,
    data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-2.40450 -0.66164  0.04499  0.63037  2.70949

Coefficients:
                       Estimate Std. Error t value
(Intercept)             3.68005    0.44285   8.310
`Complaint Resolution`  0.59499    0.07946   7.488
                       Pr(>|t|)
(Intercept)            5.51e-13 ***
`Complaint Resolution` 3.09e-11 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9554 on 98 degrees of freedom
Multiple R-squared:  0.3639,    Adjusted R-squared:  0.3574
F-statistic: 56.07 on 1 and 98 DF,  p-value: 3.085e-11
```

```
                              2.5 %     97.5 %
(Intercept)             2.8012286 4.5588623
`Complaint Resolution` 0.4373084 0.7526786
```





Inferences:

The equation from the above results can be as follows:

Customer* Satisfaction = *(0.59499)* ×*Complaint Resolution* + *3.68005

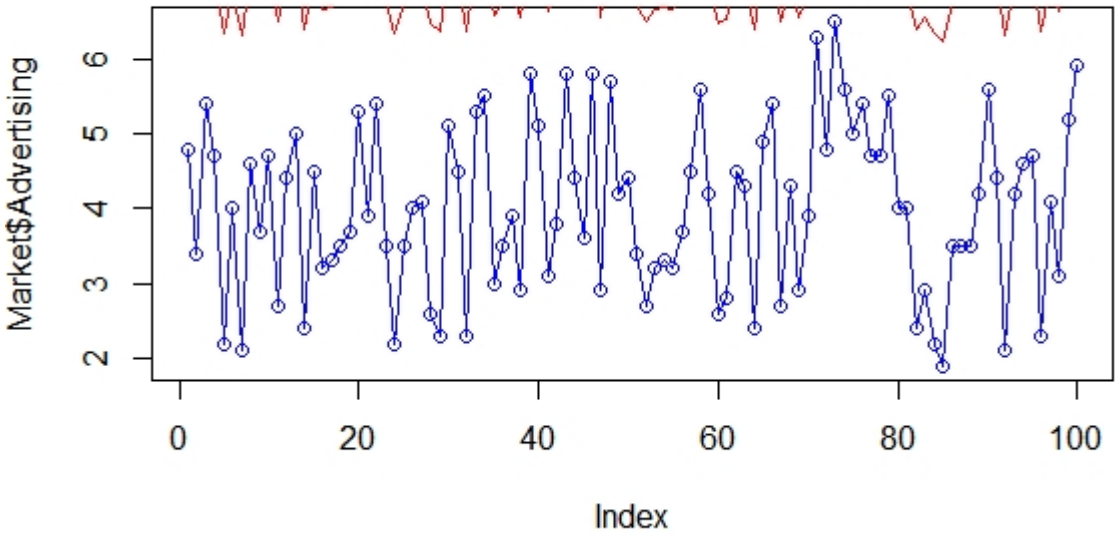
From the Plots, we can see that this Regression Equation is not significant and is **not a fit** equation as it deviates a lot from the actual values. This is also evident in its **low R-Squared value of 0.03**

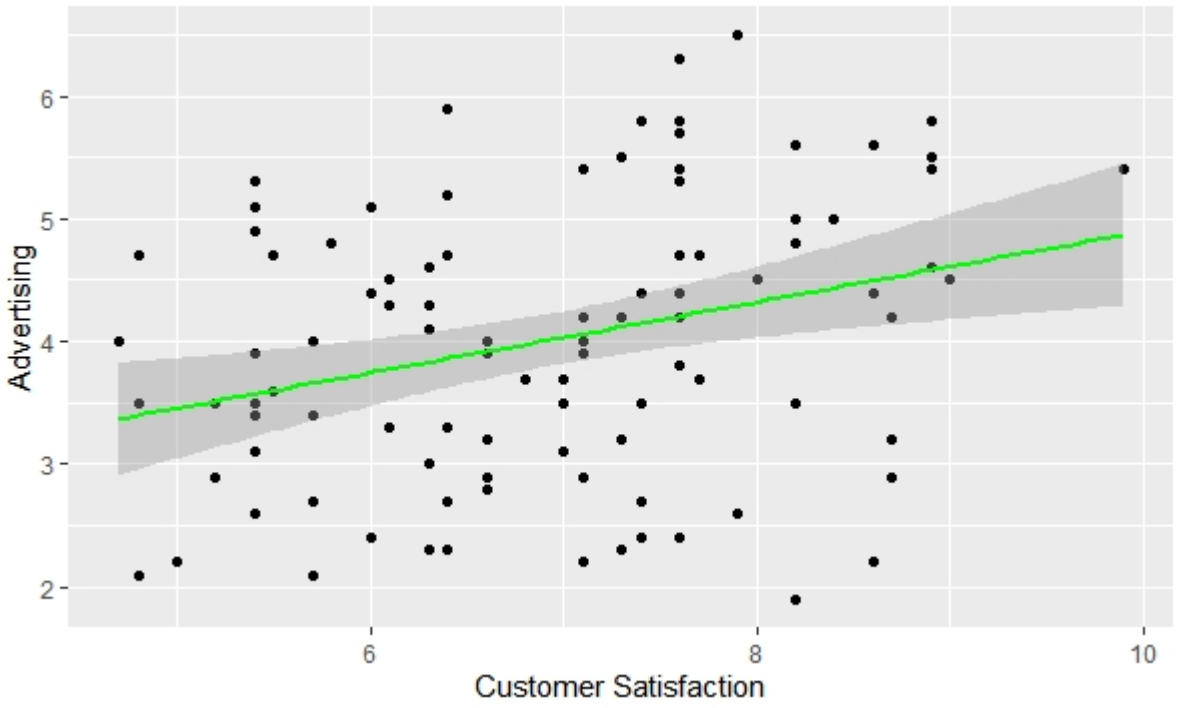
- ## Advertising:

```
### Advertising ###
slm5 = lm(`Customer Satisfaction` ~ `Advertising`,data = Market)
summary(slm5)
p5 = predict(slm5)
plot(p5,col = "Red")
plot(Market$`Advertising`,col = "Blue")
lines(p5,col = "Red")
lines(Market$`Advertising`,col = "Blue")
confint(slm5)
ggplot(Market,aes(x = `Customer Satisfaction`,y = `Advertising`))+geom_point(col = "Black")+stat_smooth(method = "lm",col = "Green")
```

```
Call:
lm(formula = `Customer Satisfaction` ~ Advertising, data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-2.34033 -0.92755  0.05577  0.79773  2.53412

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.6259     0.4237  13.279  < 2e-16 ***
Advertising   0.3222     0.1018   3.167  0.00206 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.141 on 98 degrees of freedom
Multiple R-squared:  0.09282,   Adjusted R-squared:  0.08357
F-statistic: 10.03 on 1 and 98 DF,  p-value: 0.002056
```

```
                2.5 %    97.5 %
(Intercept) 4.7851364 6.4667052
Advertising 0.1202881 0.5241405
```





Inferences:

The equation from the above results can be as follows:

Customer* Satisfaction = *(0.3222)* ×*Advertising* + *5.6259

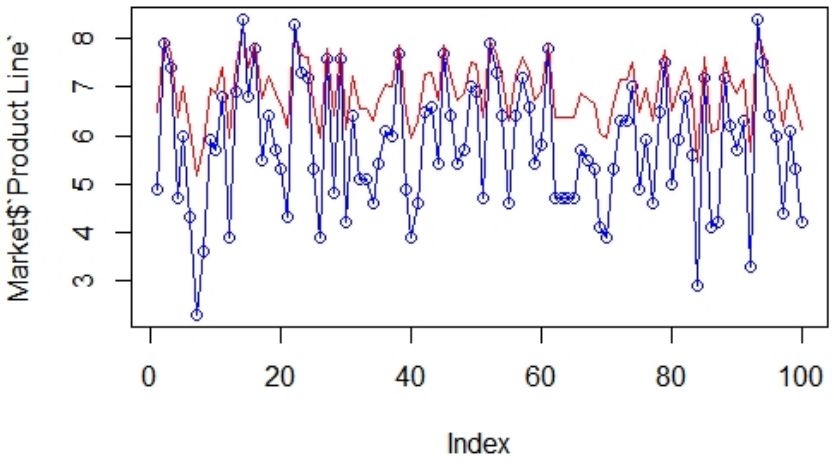
From the Plots, we can see that this Regression Equation is not significant and is **not a fit** equation as it deviates a lot from the actual values. This is also evident in its **low R-Squared value of 0.09**

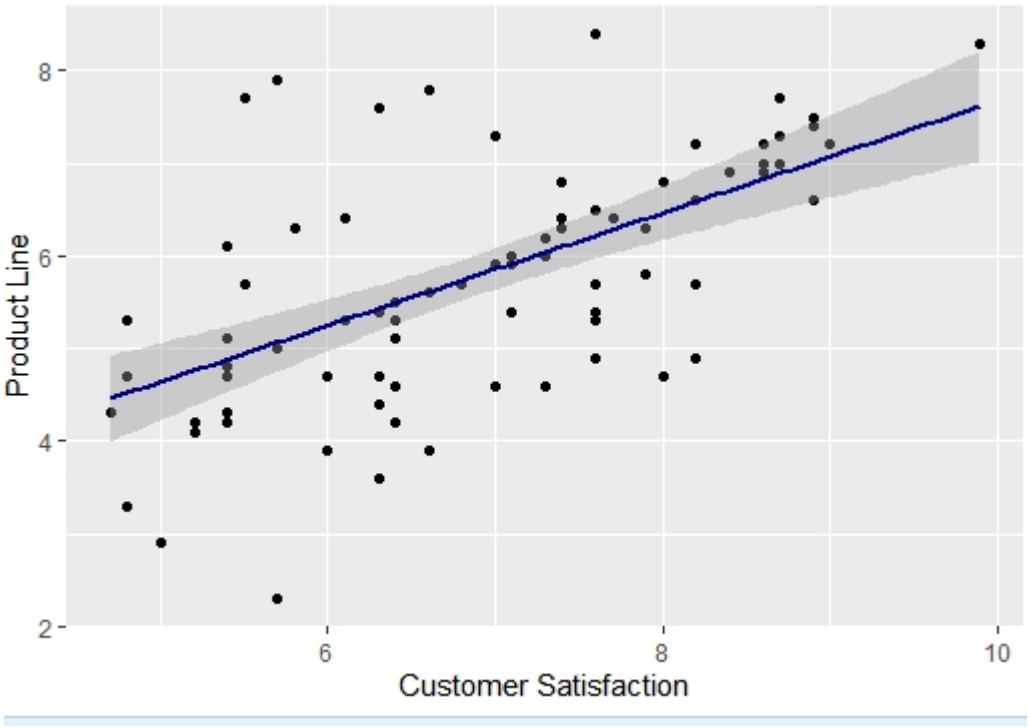
- **Product Line:**

```
### Product Line ###
slm6 = lm(`Customer Satisfaction` ~ `Product Line`,data = Market)
summary(slm6)
p6 = predict(slm6)
plot(p6,col = "Red")
plot(Market$`Product Line`,col = "Blue")
lines(p6,col = "Red")
lines(Market$`Product Line`,col = "Blue")
confint(slm6)
ggplot(Market,aes(x = `Customer Satisfaction`,y = `Product Line`))+geom_point(col = "Black")+stat_smooth(method = "lm",col = "Dark Blue")
```

```
Call:
lm(formula = `Customer Satisfaction` ~ `Product Line`, data = Market)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3634 -0.7795  0.1097  0.7604  1.7373

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.02203    0.45471   8.845 3.87e-14 ***
`Product Line` 0.49887    0.07641   6.529 2.95e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1 on 98 degrees of freedom
Multiple R-squared:  0.3031,    Adjusted R-squared:  0.296
F-statistic: 42.62 on 1 and 98 DF,  p-value: 2.953e-09
```

```
> confint(slm6)
                    2.5 %     97.5 %
(Intercept)     3.1196708 4.9243955
`Product Line` 0.3472346 0.6505145
> |
```





Inferences:

The equation from the above results can be as follows:

Customer* Satisfaction = *(0.49887) ×Product Line+ 4.02203

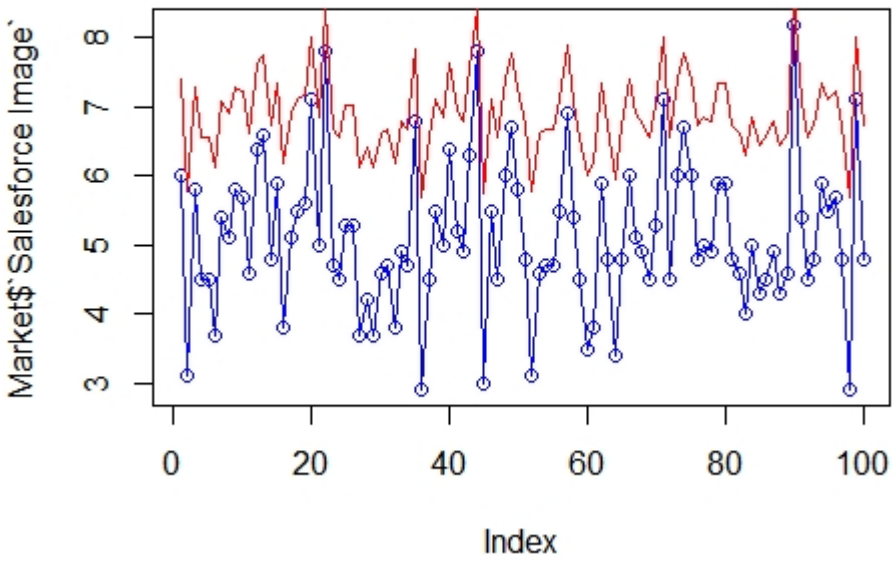
From the Plots, we can see that this Regression Equation is not significant and is **not a fit** equation as it deviates a lot from the actual values. This is also evident in its **low R-Squared value of 0.30**

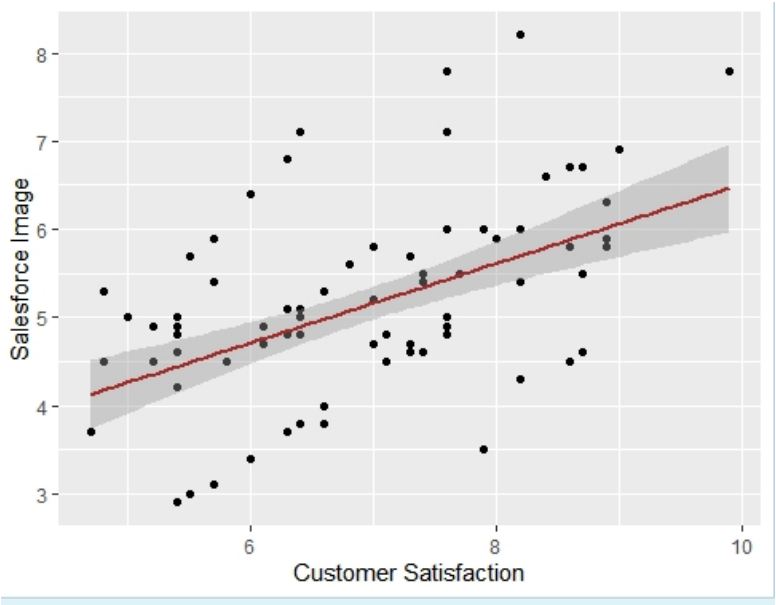
- ## Salesforce Image:

```
### Salesforce Image ###
slm7 = lm(`Customer Satisfaction` ~ `Salesforce Image`,data = Market)
summary(slm7)
p7 = predict(slm7)
plot(p7,col = "Red")
plot(Market$`Salesforce Image`,col = "Blue")
lines(p7,col = "Red")
lines(Market$`Salesforce Image`,col = "Blue")
confint(slm7)
ggplot(Market,aes(x = `Customer Satisfaction`,y = `Salesforce Image`))+geom_point(col = "Black")+stat_smooth(method = "lm",col = "Brown")
```

```
Call:
lm(formula = `Customer Satisfaction` ~ `Salesforce Image`, data = Market

Residuals:
    Min      1Q  Median      3Q     Max
-2.2164 -0.5884  0.1838  0.6922  2.0728

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         4.06983    0.50874   8.000 2.54e-12 ***
`Salesforce Image`  0.55596    0.09722   5.719 1.16e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.037 on 98 degrees of freedom
Multiple R-squared:  0.2502,    Adjusted R-squared:  0.2426
F-statistic:  32.7 on 1 and 98 DF,  p-value: 1.164e-07
```

```
                        2.5 %     97.5 %
(Intercept)         3.0602511 5.0794074
`Salesforce Image` 0.3630295 0.7488857
```





The equation from the above results can be as follows:

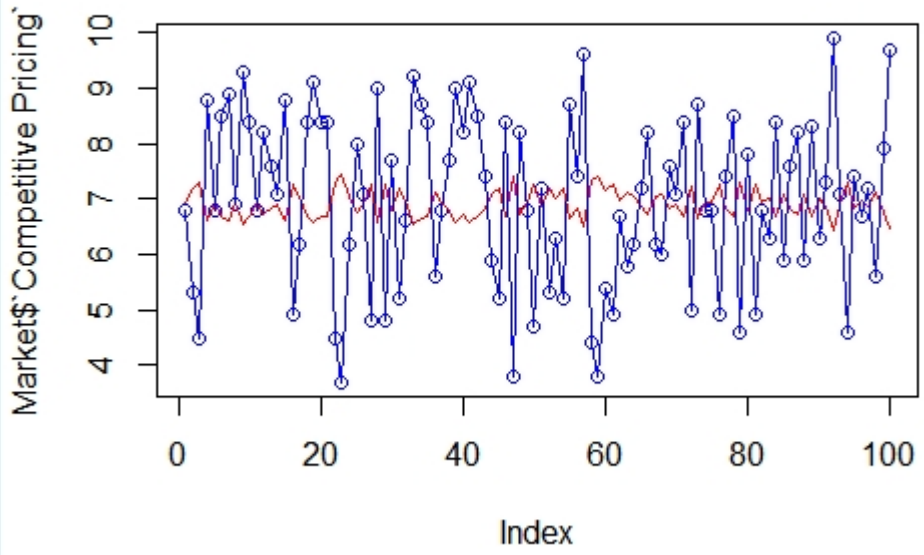
Customer* Satisfaction = *(0.5596) ×Salesforce Image+ 4.06983

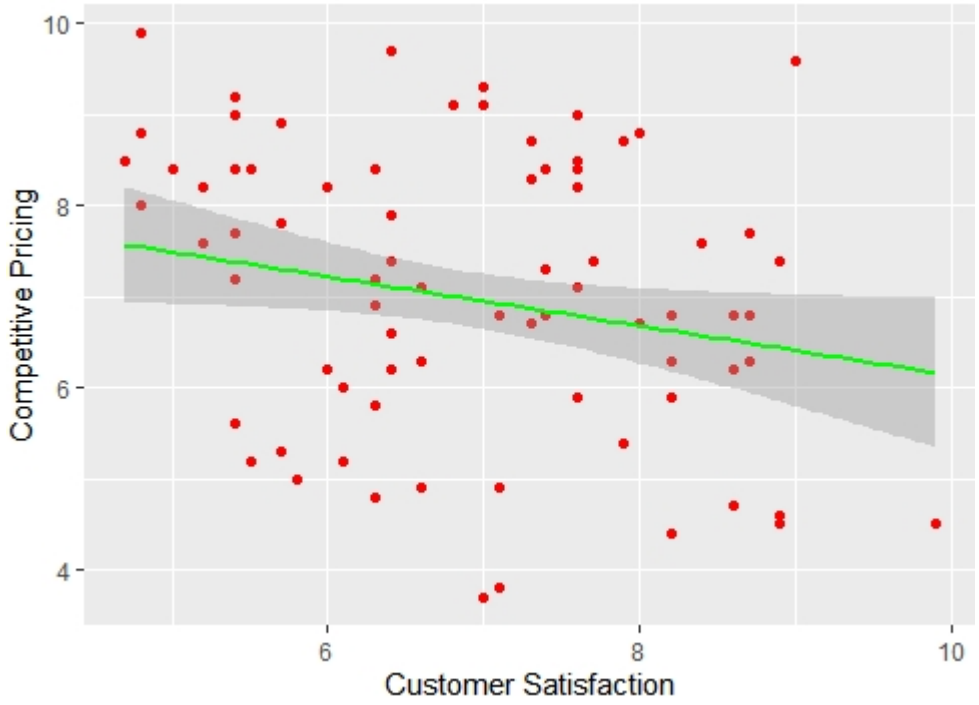
From the Plots, we can see that this Regression Equation is not significant and is **not a fit** equation as it deviates a lot from the actual values. This is also evident in its **low R-Squared value of 0.25**

- **Competitive Pricing:**

```
### Competitive Pricing ###
slm8 = lm(`Customer Satisfaction` ~ `Competitive Pricing`,data = Market)
summary(slm8)
p8 = predict(slm8)
plot(p8,col = "Red")
plot(Market$`Competitive Pricing`,col = "Blue")
lines(p8,col = "Red")
lines(Market$`Competitive Pricing`,col = "Blue")
confint(slm8)
ggplot(Market,aes(x = `Customer Satisfaction`,y = `Competitive Pricing`))+geom_point(col = "Red")+stat_smooth(method = "lm",col = "Green")
```

```
Call:
lm(formula = `Customer Satisfaction` ~ `Competitive Pricing`,
    data = Market)

Residuals:
    Min      1Q  Median      3Q     Max
-1.9728 -0.9915 -0.1156  0.9111  2.5845

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)            8.03856    0.54427  14.769   <2e-16 ***
`Competitive Pricing` -0.16068    0.07621  -2.108   0.0376 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.172 on 98 degrees of freedom
Multiple R-squared:  0.04339,   Adjusted R-squared:  0.03363
F-statistic: 4.445 on 1 and 98 DF,  p-value: 0.03756


                              2.5 %       97.5 %
(Intercept)             6.9584773  9.118646619
`Competitive Pricing` -0.3119192 -0.009434976
```





Inferences:

The equation from the above results can be as follows:

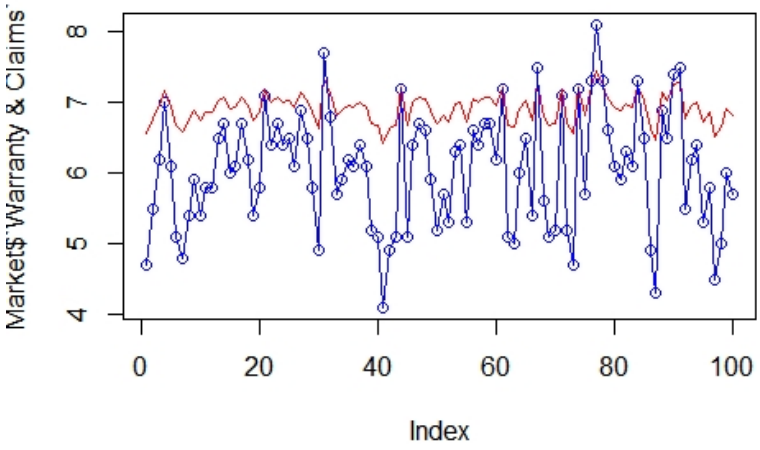
Customer* Satisfaction = *(-0.16068) ×Competitive Pricing+ 8.03856

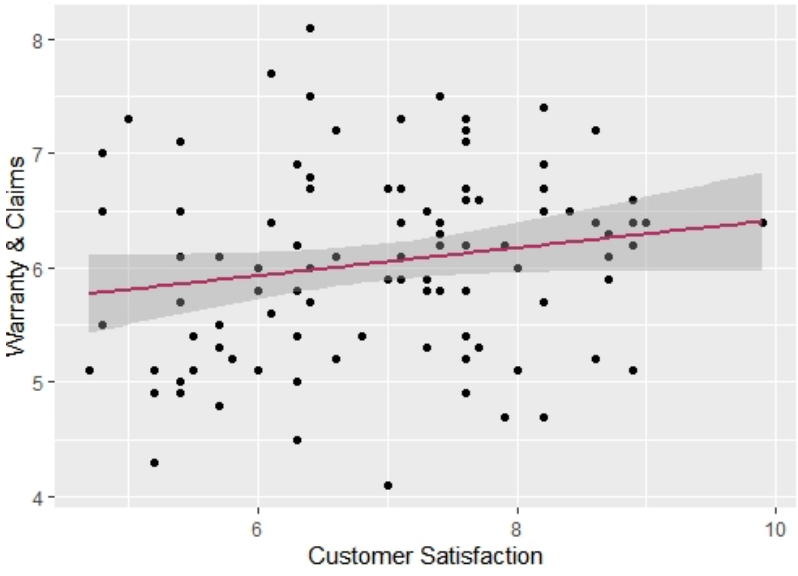
From the Plots, we can see that this Regression Equation is not significant and is **not a fit** equation as it deviates a lot from the actual values. This is also evident in its **low R-Squared value of 0.04**

- ## Warranty & Claims:

```
### Warranty & Claims ###
slm9 = lm(`Customer Satisfaction` ~ `Warranty & Claims`,data = Market)
summary(slm9)
p9 = predict(slm9)
plot(p9,col = "Red")
plot(Market$`Warranty & Claims`,col = "Blue")
lines(p9,col = "Red")
lines(Market$`Warranty & Claims`,col = "Blue")
confint(slm9)
ggplot(Market,aes(x = `Customer Satisfaction`,y = `Warranty & Claims`))+geom_point(col = "Black")+stat_smooth(method = "lm",col = "Maroon")
```

```
Call:
lm(formula = `Customer Satisfaction` ~ `Warranty & Claims`, data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-2.36504 -0.90202  0.03019  0.90763  2.88985

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)          5.3581     0.8813   6.079 2.32e-08 ***
`Warranty & Claims`  0.2581     0.1445   1.786   0.0772 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.179 on 98 degrees of freedom
Multiple R-squared:  0.03152,   Adjusted R-squared:  0.02164
F-statistic:  3.19 on 1 and 98 DF,  p-value: 0.0772


                        2.5 %    97.5 %
(Intercept)          3.6090720 7.107082
`Warranty & Claims` -0.0286887 0.544963
> 0.40887
```





Inferences:

The equation from the above results can be as follows:

Customer* Satisfaction = *(0.2581)* × *Warranty and Claims* + *5.3581

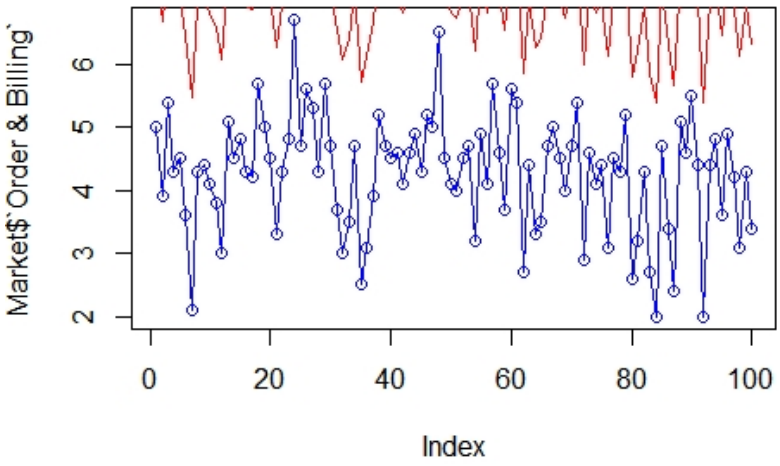
From the Plots, we can see that this Regression Equation is not significant and is **not a fit** equation as it deviates a lot from the actual values. This is also evident in its **low R-Squared value of 0.03**

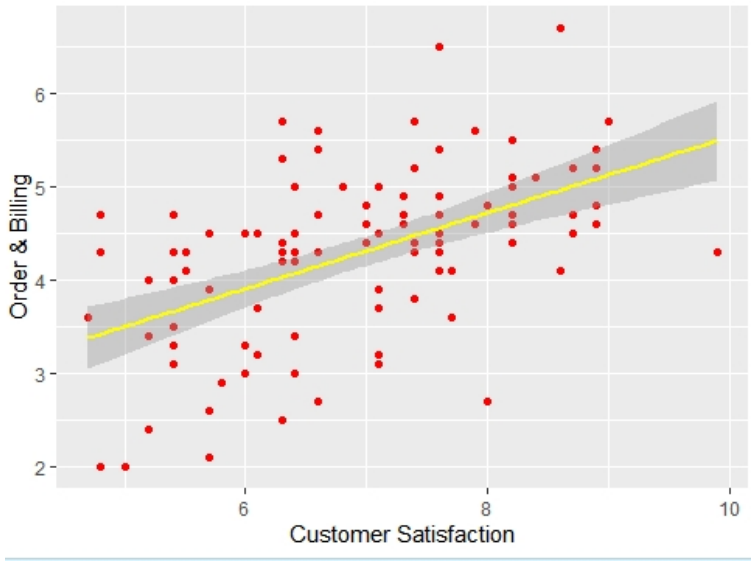
- ## Orders and Billing:

```
### Order & Billing ###
slm10 = lm(`Customer Satisfaction` ~ `Order & Billing`,data = Market)
summary(slm10)
p10 = predict(slm10)
plot(p10,col = "Red")
plot(Market$`Order & Billing`,col = "Blue")
lines(p10,col = "Red")
lines(Market$`Order & Billing`,col = "Blue")
confint(slm10)
ggplot(Market,aes(x = `Customer Satisfaction`,y = `Order & Billing`))+geom_point(col = "Red")+stat_smooth(method = "lm",col = "Yellow")
```

```
Call:
lm(formula = `Customer Satisfaction` ~ `Order & Billing`, data = Market)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4005 -0.7071 -0.0344  0.7340  2.9673

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         4.0541     0.4840   8.377 3.96e-13 ***
`Order & Billing`   0.6695     0.1106   6.054 2.60e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.022 on 98 degrees of freedom
Multiple R-squared:  0.2722,    Adjusted R-squared:  0.2648
F-statistic: 36.65 on 1 and 98 DF,  p-value: 2.602e-08
```

```
                      2.5 %     97.5 %
(Intercept)        3.093639 5.0144660
`Order & Billing` 0.450021 0.8888978
```





Inferences:

The equation from the above results can be as follows:

Customer* Satisfaction = *(0.6695)* × *Order & Billing* + *4.0541

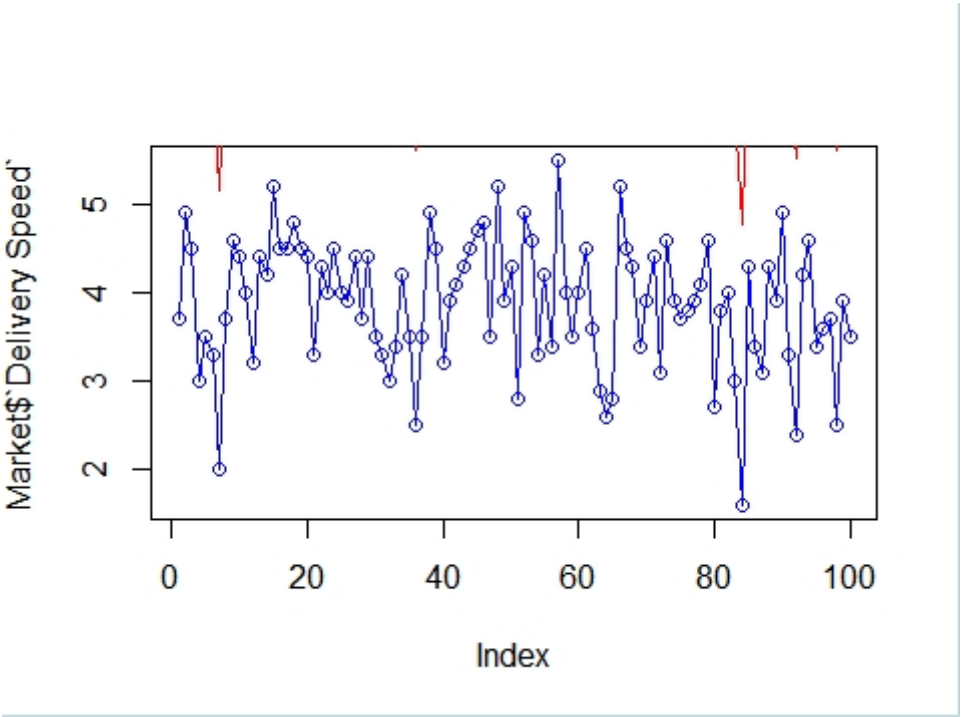
From the Plots, we can see that this Regression Equation is not significant and is **not a fit** equation as it deviates a lot from the actual values. This is also evident in its **low R-Squared value of 0.27**

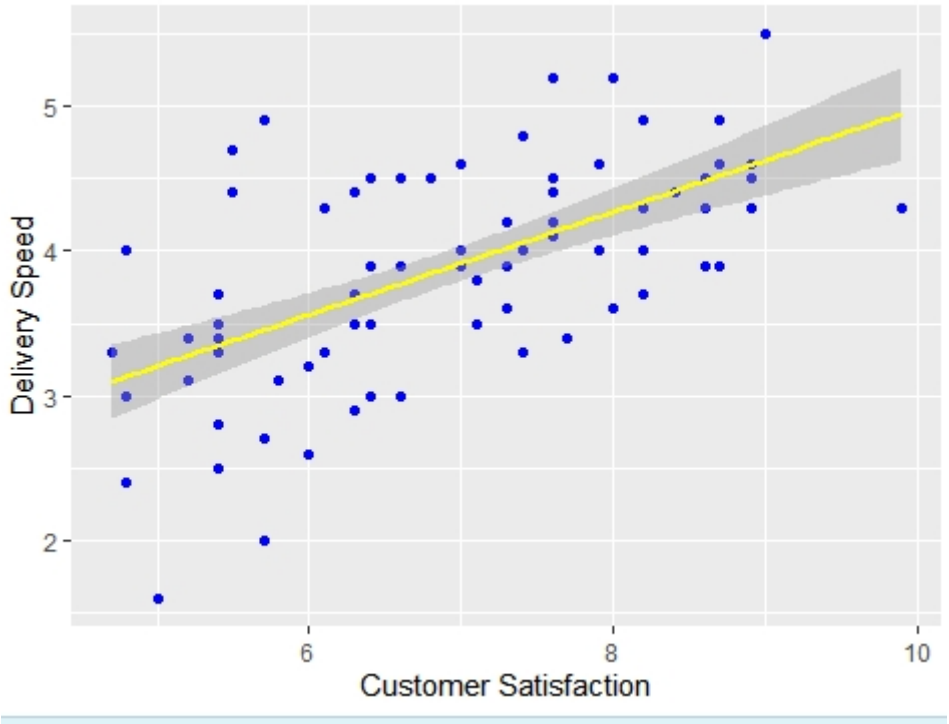
- **Delivery Speed:**

```
### Delivery Speed ###
slm11 = lm(`Customer Satisfaction` ~ `Delivery Speed`,data = Market)
summary(slm11)
p11 = predict(slm11)
plot(p11,col = "Red")
plot(Market$`Delivery Speed`,col = "Blue")
lines(p11,col = "Red")
lines(Market$`Delivery Speed`,col = "Blue")
confint(slm11)
ggplot(Market,aes(x = `Customer Satisfaction`,y = `Delivery Speed`))+geom_point(col = "Blue")+stat_smooth(method = "lm",col = "Yellow")
```

```
Call:
lm(formula = `Customer Satisfaction` ~ `Delivery Speed`, data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-2.22475 -0.54846  0.08796  0.54462  2.59432

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)        3.2791     0.5294   6.194 1.38e-08 ***
`Delivery Speed`   0.9364     0.1339   6.994 3.30e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9783 on 98 degrees of freedom
Multiple R-squared:  0.333,     Adjusted R-squared:  0.3262
F-statistic: 48.92 on 1 and 98 DF,  p-value: 3.3e-10
```

```
> confint(slm11)
                    2.5 %    97.5 %
(Intercept)      2.2285309 4.329613
`Delivery Speed` 0.6707367 1.202103
```





Inferences:

The equation from the above results can be as follows:

Customer* Satisfaction = *(0.9364)* × *Delivery Speed* + *3.2791

From the Plots, we can see that this Regression Equation is not significant and is **not a fit** equation as it deviates a lot from the actual values. This is also evident in its **low R-Squared value of 0.33**

Conclusion:

We can see that all the independent variables do not give a significant and good ***Simple Linear Regression models.*** *This occurs because of the* ***Multicollinearity*** *with other variables which masks the value of* ***Variance Explained*** of that variable in that model.

Q4. Perform PCA/Factor analysis by extracting 4 factors. Interpret the output and name the Factors.

Sol: PCA(Principal Component Analysis) and Factor Analysis are Dimensional Reduction techniques that are used when a dataset contains huge number of dependent variables.

Before performing any of the techniques, we must first find out whether the dataset is suitable for a Dimension Reduction or not. This can be done with Bartlett Test which tests for Homogeneity between the variables. The function **bartlett.test()** can be used to perform the test. If the **P-value** in the test is **less than 0.05**, then we can conclude that Dimension Reduction is necessary. The interpretation of this result and the previous results where we found that **Multicollinearity** exists makes it necessary to perform a Dimension Reduction.

Principal Component Analysis:

PCA technique lessens the number of **Dependent variables** by choosing only the variables that explain the majority of variance occurring in the **Regression Model.** The PCA can be done by calculating the Eigen values, and then using the **Rule of Elbow** and **values above 1** to decide the **number of factors**. Then by doing the **Cumulative sum** of those variance explained by each of those values and then decide the **percentage of variance** we want explained.

```
> ### Principal Component Analysis ###
> EigenPCA = eigen(cor(Market1))
> EigenPCA$values
 [1] 3.42697133 2.55089671 1.69097648 1.08655606 0.60942409 0.55188378 0.40151815 0.24695154
 [9] 0.20355327 0.13284158 0.09842702
> Var.Eigen<-EigenPCA$values/sum(EigenPCA$values)*100
> cumsum(Var.Eigen)
 [1]  31.15428  54.34425  69.71677  79.59455  85.13477  90.15189  93.80206  96.04707  97.89756
[10]  99.10521 100.00000
```

Inferences:

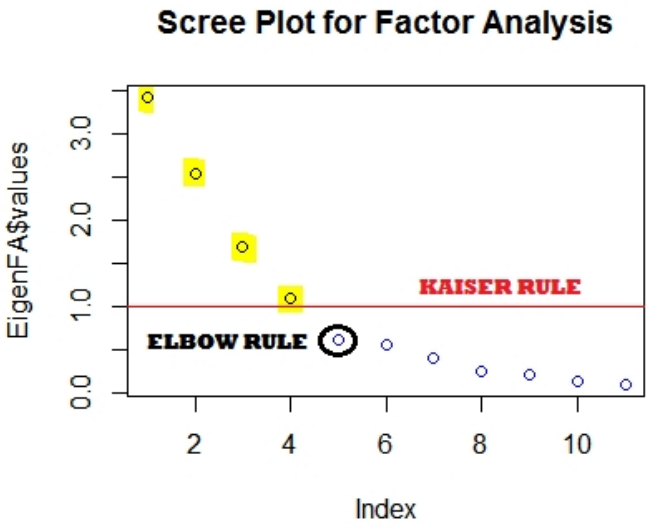
We can see from the above results that only **Four** of the eigen values are above **one.** This means that we can have **four variables** which can explain the maximum variance possible. Then from the **cumulative sum taken,** we can see that **Four of those variables** can explain **79.59455**% of the variance. So choosing the **Right four variables** from the dataset and using the Variance explained by them to create a significant **Regression model** .

Factor Analysis:

The **Factor Analysis** is the technique where the two or more variables which have **Correlation between them** merge into a **single variable or factor** which is **truly Independent** from other **Factors** and hence eliminating **Multicollinearity** and giving us **true Independent Models** for a **significant Multiple Regression Model.**

To perform a Factor Analysis, we need to decide the number of **factors** that need to be extracted from the variables which will be used in making **Regression Model.** This can be done by getting **Eigen Values** and using the **Rule of Elbow** and **the Kaiser rule which states that values above one must be considered** to decide upon how many **significant factors** can be extracted.

```
### Factor Analysis ###
EigenFA = eigen(cor(Market1))
EigenFA$values
plot(EigenFA$values,col = "Blue")
abline(a = 1,b = 0,col = "Red")
```




```
[1] 3.42697133 2.55089671 1.69097648 1.08655606 0.60942409 0.55188378 0.40151815 0.24695154
[9] 0.20355327 0.13284158 0.09842702
```

Inferences:

From the above **Scree Plot**, we can see that according to the **Elbow rule,** we have to consider **Five Factors.** But according to the **Kaiser Rule,** we have to consider only **four factors.** Keeping both the rules while selecting, **we consider four factors that will be used for the Factor Analysis.**

After deciding on the number of variables to **extract,** we need to extract those variables using the function **principal()** from the **psych** library with argument **nfactors = 4**. After extraction of the variables, we must find out which of the variables were used for the extraction for each of the **Four Factors.** Then we can **name the Extracted Factor** based on the variables that were used for **extracting it.** Depending on how much each of the variance the **Factor** explains of the particular variable, we can decide whether that particular variable **was used for extracting the factor or not.** If the values of the variance

explained are not **clear enough to decide** the variable, we perform an **Orthogonal Rotation** giving the argument **rotate = "varimax"** in the **principal()** function. This will help giving us a better picture as all the values are rotated on their respective axes and giving us a clearer values to decide.

```
### Without Rotation ###
UnrotatedFA = principal(Market1,nfactors = 4,rotate = "none")
print(UnrotatedFA)
```

```
Principal Components Analysis
Call: principal(r = Market1, nfactors = 4, rotate = "none")
Standardized loadings (pattern matrix) based upon correlation matrix
                        PC1   PC2   PC3   PC4   h2    u2 com
Product Quality        0.25 -0.50 -0.08  0.67 0.77 0.232 2.2
E-Commerce             0.31  0.71  0.31  0.28 0.78 0.223 2.1
Technical Support      0.29 -0.37  0.79 -0.20 0.89 0.107 1.9
Complaint Resolution   0.87  0.03 -0.27 -0.22 0.88 0.119 1.3
Advertising            0.34  0.58  0.11  0.33 0.58 0.424 2.4
Product Line           0.72 -0.45 -0.15  0.21 0.79 0.213 2.0
Salesforce Image       0.38  0.75  0.31  0.23 0.86 0.141 2.1
Competitive Pricing   -0.28  0.66 -0.07 -0.35 0.64 0.359 1.9
Warranty & Claims      0.39 -0.31  0.78 -0.19 0.89 0.108 2.0
Order & Billing        0.81  0.04 -0.22 -0.25 0.77 0.234 1.3
Delivery Speed         0.88  0.12 -0.30 -0.21 0.91 0.086 1.4

                       PC1  PC2  PC3  PC4
SS loadings           3.43 2.55 1.69 1.09
Proportion Var        0.31 0.23 0.15 0.10
Cumulative Var        0.31 0.54 0.70 0.80
Proportion Explained  0.39 0.29 0.19 0.12
Cumulative Proportion 0.39 0.68 0.88 1.00

Mean item complexity =  1.9
Test of the hypothesis that 4 components are sufficient.

The root mean square of the residuals (RMSR) is  0.06
 with the empirical chi square  39.02  with prob <  0.0018

Fit based upon off diagonal values = 0.97
```

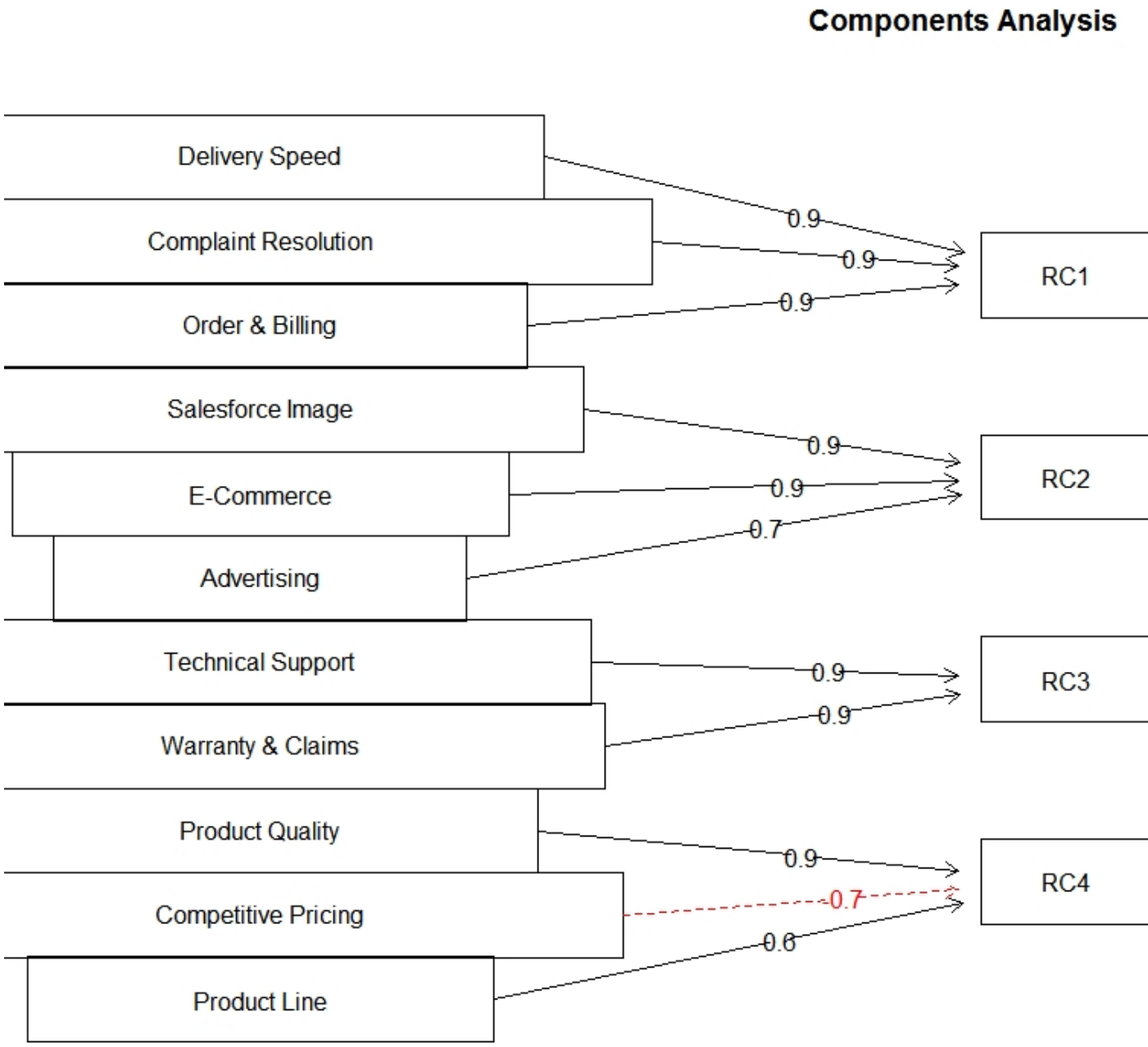
From the above results, we can see that the values are **not significant** enough to differentiate the values. So that's why we use **Orthogonal Rotation** to get significant values.

```
### With Rotation ###
RotatedFA = principal(Market1,nfactors = 4,rotate = "varimax")
print(RotatedFA)
```

```
Principal Components Analysis
Call: principal(r = Market1, nfactors = 4, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix
                        RC1   RC2   RC3   RC4   h2    u2 com
Product Quality        0.00 -0.01 -0.03  0.88 0.77 0.232 1.0
E-Commerce             0.06  0.87  0.05 -0.12 0.78 0.223 1.1
Technical Support      0.02 -0.02  0.94  0.10 0.89 0.107 1.0
Complaint Resolution   0.93  0.12  0.05  0.09 0.88 0.119 1.1
Advertising            0.14  0.74 -0.08  0.01 0.58 0.424 1.1
Product Line           0.59 -0.06  0.15  0.64 0.79 0.213 2.1
Salesforce Image       0.13  0.90  0.08 -0.16 0.86 0.141 1.1
Competitive Pricing   -0.09  0.23 -0.25 -0.72 0.64 0.359 1.5
Warranty & Claims      0.11  0.05  0.93  0.10 0.89 0.108 1.1
Order & Billing        0.86  0.11  0.08  0.04 0.77 0.234 1.1
Delivery Speed         0.94  0.18  0.00  0.05 0.91 0.086 1.1

                       RC1  RC2  RC3  RC4
SS loadings           2.89 2.23 1.86 1.77
Proportion Var        0.26 0.20 0.17 0.16
Cumulative Var        0.26 0.47 0.63 0.80
Proportion Explained  0.33 0.26 0.21 0.20
Cumulative Proportion 0.33 0.59 0.80 1.00

Mean item complexity =  1.2
Test of the hypothesis that 4 components are sufficient.

The root mean square of the residuals (RMSR) is  0.06
 with the empirical chi square  39.02  with prob <  0.0018

Fit based upon off diagonal values = 0.97
```

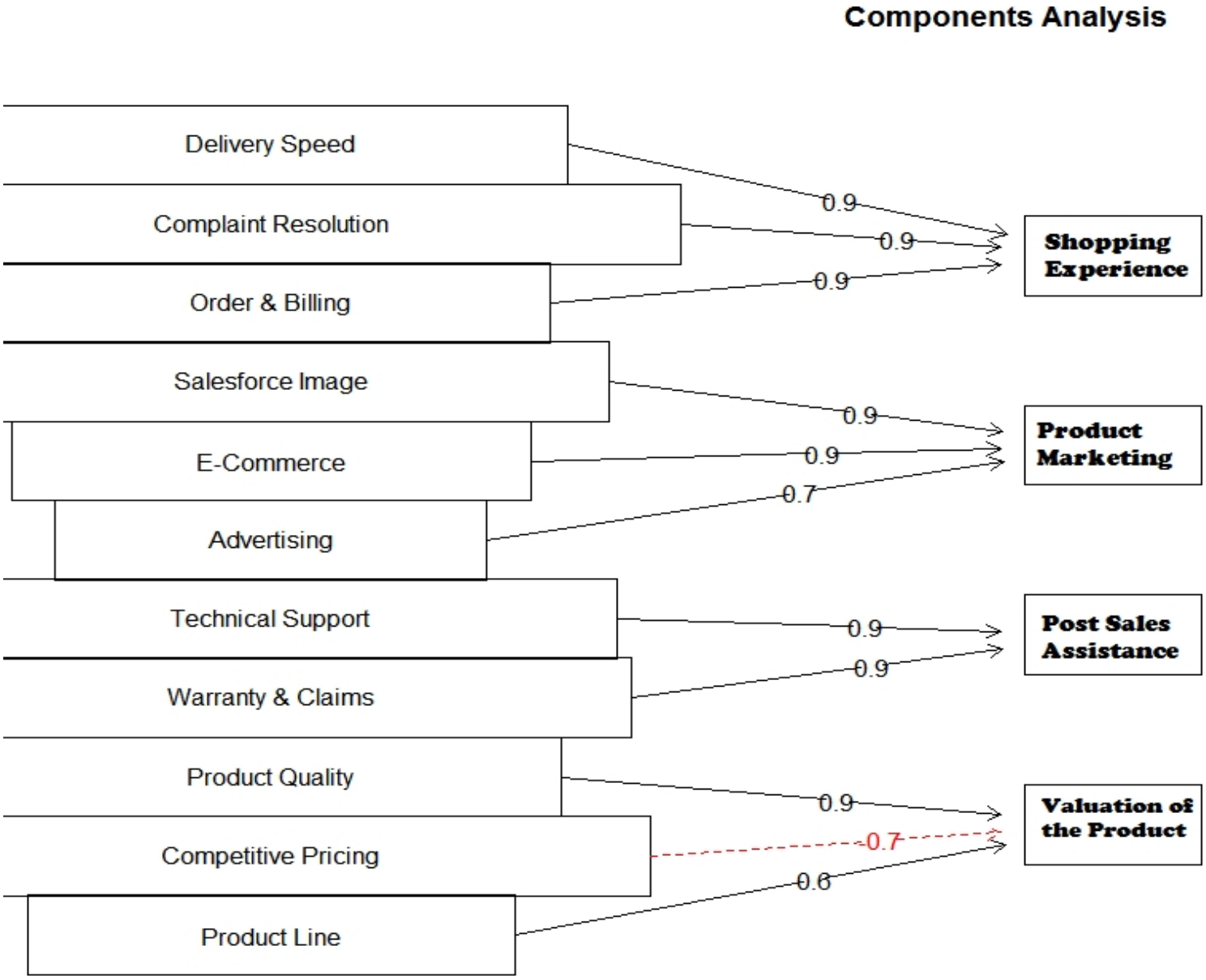
After **Rotation**, we can see that the values are all **significant** and can be **differentiated easily**. By doing a Factor Analysis diagram using the function **fa.diagram()**, we can find out which of the variables correspond to which of the factors.

```
### FA Diagram ####
fa.diagram(RotatedFA)
```



We can see that the factors **RC1, RC2, RC3** and **RC4** can be termed as the following:

Factor name	Variables Included
Shopping Experience(RC1)	Delivery Speed, Complaint Resolution, Order and Billing
Product Marketing(RC2)	Salesforce Image, E-Commerce and Advertising
Post Sales Assistance(RC3)	Technical Support and Warranty & Claims
Valuation of the Prouduct(RC4)	Product Quality, Competitve Pricing and Product Line.



Conclusion:

We can see that from the **11 Variables** given in the dataset, we have reduced the number of variables to **four** variables or segments of the company. We can see that **Factor Analysis** has given us a better picture in the segmentation of the Market. The four new segments thus formed can be explained as below:

1. **Shopping Experience** – It is a segment which describes all round experience of the customer from

the time he spends time selecting the product on the market Place until he receives the product

2. **Product Marketing –** It is a segment which describes how the marketing of the products overall is done in the marketplace to attract more customers
3. **After Sales Assistance –** It is a segment which describes the support provided to the customer after the purchase of the product from the market place if any queries arise.
4. **Valuation of the product –** It is a segment which describes the value and the position of the product in the market as compared to other products.

These four factors, ***Shopping Experience, Product Marketing, After Sales Assistance and Valuation of the product,*** *give proper segmentation in the market and help management understand better and take better decisions.*

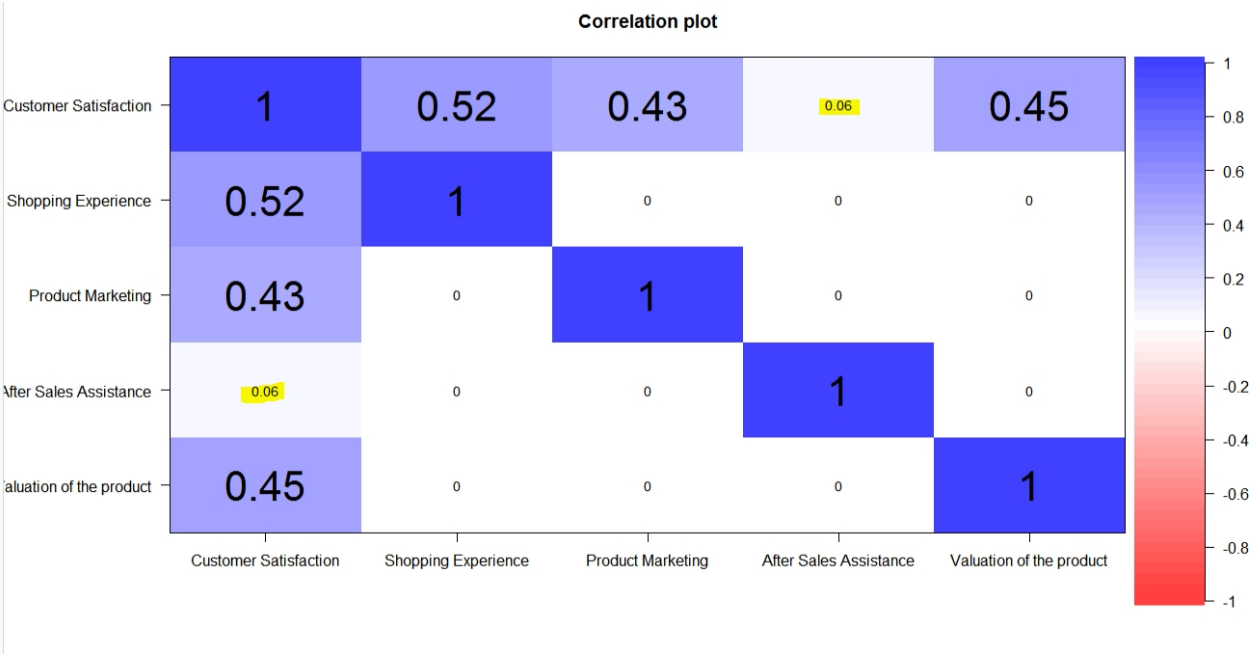
Q5. Perform Multiple linear regression with customer satisfaction as dependent variables and the four factors as independent variables. Comment on the Model output and validity. Your remarks should make it meaningful for everybody

Sol:- From the previous questions, we have performed **Factor Analysis** and extracted **four factors** namely **Shopping Experience, Product Marketing, After Sales Assistance and Valuation of the product.** Taking **Customer Satisfaction** as the **Dependent Variable**, we must create a **Multiple**

Regression Model with the four extracted factors. The factor scores can be extracted by using the vector, **RotatedFA$scores(RotatedFA is the Rotated Factor Analysis model created in the previous question)**. After extracting the factor names, these factors must be used to create a data frame with Customer Satisfaction so that **Multiple Regression Model** can be created. After creating the data frame by using the function **data.frame()**, we must assign names to the columns using the function **colnames().** Then we can check if the data frame has been created as required by using the function **str()**.

```
> ###Question 5 ###
> ### Extracting the Factor scores and Naming them ###
> Market.factors = data.frame(Market$`Customer Satisfaction`,RotatedFA$scores)
> colnames(Market.factors) = c("Customer Satisfaction",
+                              "Shopping Experience",
+                              "Product Marketing",
+                              "After Sales Assistance",
+                              "Valuation of the product")
> str(Market.factors)
'data.frame':   100 obs. of  5 variables:
 $ Customer Satisfaction   : num  8.2 5.7 8.9 4.8 7.1 4.7 5.7 6.3 7 5.5 ...
 $ Shopping Experience     : num  0.127 1.222 0.616 -0.845 -0.32 ...
 $ Product Marketing       : num  0.77 -1.646 0.58 -0.272 -0.834 ...
 $ After Sales Assistance  : num  -1.87845 -0.61403 0.00369 1.26749 -0.0081 ...
 $ Valuation of the product: num  0.366 0.813 1.57 -1.254 0.448 ...
```

After creation of the data frame, we need to check for **Multicollinearity** by plotting a **Correlation Plot** using the function **cor.plot()**.



Inferences:

From the above results, we can see that even though most of the other variables do not have **that high of Correlation with the variable Customer Satisfaction**, the variable **After Sales Assistance** has **very less Correlation** with the Dependent variable **Customer Satisfaction.** This low **Correlation** will be reflected in the Regression Model when its **Slope** becomes insignificant. Even though the **Slopes** of other values become **significant,** due to low correlation values, there are chances that the **Regression Model** might not yield a **Good R-Squared Value.**

We can create the **Multiple Regression Model** by using the function **lm()**.

Using the four variables:

```
### Creating a Multiple Regression Model using the new factors ###
MR = lm(`Customer Satisfaction`~.,data = Market.factors)
summary(MR)

Call:
lm(formula = `Customer Satisfaction` ~ ., data = Market.factors)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6308 -0.4996  0.1372  0.4623  1.5228

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)               6.91800    0.07089  97.589  < 2e-16 ***
`Shopping Experience`     0.61805    0.07125   8.675 1.12e-13 ***
`Product Marketing`       0.50973    0.07125   7.155 1.74e-10 ***
`After Sales Assistance`  0.06714    0.07125   0.942    0.348
`Valuation of the product` 0.54032   0.07125   7.584 2.24e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7089 on 95 degrees of freedom
Multiple R-squared:  0.6605,    Adjusted R-squared:  0.6462
F-statistic: 46.21 on 4 and 95 DF,  p-value: < 2.2e-16
```

Inferences:

From the above results, we can see that **R-Squared value** (**0.6605)** is actually lesser than that **R-Squared value** of the **Multiple Regression model** done with **original values** which is **0.8021**. This lower **R-Squared value** tells us that the **variance explained** by these variables is **lower**. But we can see that the number of insignificant slopes have been reduced to one as opposed to seven. The variable **After Sales Assistance** is an insignificant variable and so it can be removed from the regression model.

After removal of the After Sales Assistance variable:

```
MR1 = lm(`Customer Satisfaction`~`Shopping Experience
         Product Marketing`+`Valuation of the product`,data = Market.factors)
summary(MR1)

Call:
lm(formula = `Customer Satisfaction` ~ `Shopping Experience` +
    `Product Marketing` + `Valuation of the product`, data = Market.factors)

Residuals:
     Min       1Q   Median       3Q      Max
-1.69684 -0.49928  0.09364  0.46420  1.57638

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                6.91800    0.07085  97.646  < 2e-16 ***
`Shopping Experience`      0.61805    0.07120   8.680 1.01e-13 ***
`Product Marketing`        0.50973    0.07120   7.159 1.64e-10 ***
`Valuation of the product` 0.54032    0.07120   7.588 2.09e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7085 on 96 degrees of freedom
Multiple R-squared:  0.6574,    Adjusted R-squared:  0.6466
F-statistic: 61.39 on 3 and 96 DF,  p-value: < 2.2e-16
```

Inferences:

From the above results, we can see that **R-Squared value** has decreased further when the variable **After Sales Assistance** is removed. This may be due to the **interaction effect between the remaining variables.** If the **interaction effect** can be introduced, maybe there might **increase in R-Squared Value**.

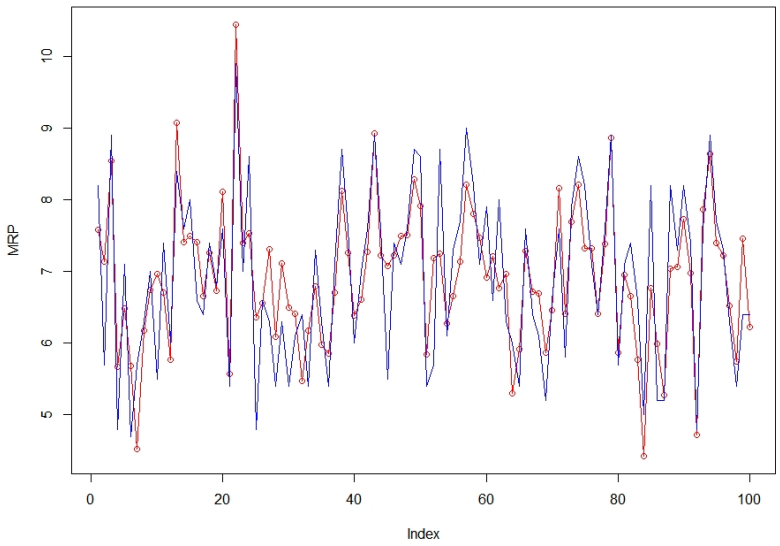
Considering the practically possible interactions between the variables:

The interaction was chosen keeping the **Business intuition in mind** as to which of these interactions may occur in real life situation, we can try to consider the below **interaction effect**:

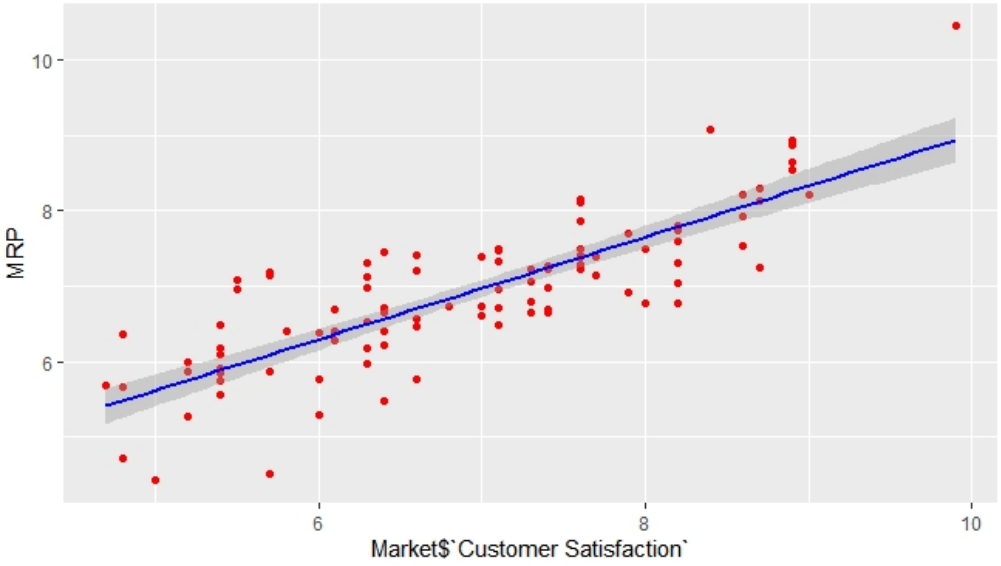
Product Marketing and Valuation of the product – If the marketing of the product is done right, it might increase sales of the product, which in turn decides the valuation of the product. The product with high valuation in the market can be marketed extensively than the product with lower valuation. The new variable formed from the interaction can be named

as **“Product Appeal”.** Then we can form the regression model with the new variable as follows:

```
###Creation of new interaction effect ###
Product.Appeal = Market.factors$`Product Marketing`*Market.factors$`Valuation of the product`
MR2 = lm(`Customer Satisfaction`~`Shopping Experience`+`Product Marketing`+
           `Valuation of the product`+Product.Appeal,data = Market.factors)
summary(MR2)

Call:
lm(formula = `Customer Satisfaction` ~ `Shopping Experience` +
    `Product Marketing` + `Valuation of the product` + Product.Appeal,
    data = Market.factors)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5780 -0.4711  0.1079  0.4134  1.4461

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                 6.91800    0.06900 100.257  < 2e-16 ***
`Shopping Experience`       0.65504    0.07092   9.236 7.10e-15 ***
`Product Marketing`         0.45163    0.07317   6.172 1.65e-08 ***
`Valuation of the product`  0.50659    0.07066   7.169 1.62e-10 ***
Product.Appeal              0.18653    0.07490   2.490   0.0145 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.69 on 95 degrees of freedom
Multiple R-squared:  0.6784,    Adjusted R-squared:  0.6648
F-statistic: 50.09 on 4 and 95 DF,  p-value: < 2.2e-16


###Plotting against the predicted values ###
MRP = predict(MR2)
plot(MRP,col = "Red")
lines(MRP,col = "Red")
lines(Market$`Customer Satisfaction`,col = "Blue")
### Finding Upper and Lower limits ###
confint(MR2)
ggplot(Market,aes(x = Market$`Customer Satisfaction`,y = MRP))
+geom_point(col = "red1")+stat_smooth(method = "lm",col = "blue")
```



```
> confint(MR2)
                                2.5 %     97.5 %
(Intercept)                6.78101294 7.0549871
`Shopping Experience`      0.51423853 0.7958375
`Product Marketing`        0.30637568 0.5968924
`Valuation of the product` 0.36630918 0.6468659
Product.Appeal             0.03783845 0.3352131
```



Inferences:

From the above results, we can see that we have new interaction variable between **Product Marketing** and **Valuation of the Product** namely **Product Appeal** gave a **significant slope** in the Regression Model. And due to the new interaction variable, we were also able to increase the **R-Squared value** while having only significant variables in the model (p-value below 5).

Conclusion:

The regression model from using the results can be formulated as follows:

Customer* Satisfaction = *(0.65504) × Shopping Experience + (0.45163) × Product Marketing + (0.50659) × Valuation of the product + (0.18653) × Product Appeal + 6.91800

This equation can be considered good fit as proved by the ***R-Squared value and line graph between the predicted values and the original values.*** *The Regression model may not* ***explain much of the variance*** *as compared to the* ***model with original variables,*** *but it does give us all* ***significant variables*** *which in turn makes the overall* ***Regression Model*** *significant.*

4. Project Conclusion:

The basic idea behind the project using the dataset "**Hair-Product-Revised.csv"** was to understand proper segmentation of the market and creation of an **Optimum Regression Model** so that it helps **management** take decisions accordingly. The main objective to create a **Regression Model for Customer Satisfaction** was achieved through the **techniques** of **Factor Analysis and Regression Analysis.** During the analysis, few questions asked related to the project were also answered. The inferences that were made during the analysis of the project were as follows:

- As part of initial **Exploration of the Dataset**, we have seen that by analysing the **Customer Satisfaction** variable, **most** of the **Customers** were not satisfied and their most of the ratings showed that they were either **Sad** or **Not Happy.** The **E-Commerce** section contributed more to the **highest number of negative ratings** compared to all other sections. The **Product Quality** section contributed to the **highest number of positive ratings** when compared to all other sections. From this, it can be concluded that the

whenever there are positive Customer Satisfaction ratings, it is only due to the **Quality of the products** available in the Market Place rather than **E-Commerce** experience that Customers have.

- The **original variables** in the dataset had **Multicollinearity** which resulted in **Non- Significant Regression Models.** So more emphasis must be made on the selection of the variables in the dataset especially when those variables are supposed to be used for **Regression Analysis**.
- From **Factor Analysis and Principal Component Analysis,** we understood that only **four variables,** were enough to extract a model and explain almost 70% of the variance. This suggests that before selecting variables for analysis, more care must be taken on how much they actually contribute to the dataset.
- Even though we were able to create a **Multiple Regression Model**, the **goodness of fit** for the model was considerable. This happened due to the fact that model, even though **significant**, could not explain majority of the variance. Even to though we were able to explain one of the interaction, there are still many **unexplained interactions between the extracted variables** which can be considered to make the **model better.**
- As we saw from the **Correlation Plot** of the extracted Four Factors, we saw that the variable **After Sales Assistance** had very low **Correlation** with the **Customer Satisfaction**. That includes the original variables that were used for extraction like **Warranty & Claims** and

Technical Support. So we can say that effect of variable **After Sales Assistance** is negligible when compared to the other variables and could have been avoided while making the dataset.

- And also the fact that we did not get a **Good Regression Model** helms from the fact that even though we had lot number of variables to create the **Regression Model**, none of those variables never had any correlation that would be deemed as good to begin with. None of the variables were **more than 0.60** or **less than -0.60**. And the variables like **Warranty & Claims** and **Technical Support** had very negligible values of 0.16 and 0.11. So maybe better emphasis must be made while making the dataset like the increase in Sample Size, proper selection of the sample,etc.

5. Appendix - A(Source Code):

Hair Factor Analysis

```
### Hair Factor Project ###

> ### Setting up working directory
> setwd("C:/R programs great lakes/ADV STAT")
> getwd()
[1] "C:/R programs great lakes/ADV STAT"

### Invoking The Requiered Libraries ###
> install.packages("readr")
> library(readr)
> install.packages("ggplot2")
> library(ggplot2)
> install.packages("psych")
> library(psych)
> install.packages("reshape2")
> library(reshape2)
> install.packages("car")
> library(car)

> ### Loading up the dataset #####
> Market = read.csv("Factor-Hair-Revised.csv",header = TRUE)
> View(Market)
> ### Intial exploration of the dataset ###
> summary(Market)
       ID            ProdQual          Ecom          TechSup         CompR
es
```

```
 Min.   :  1.00   Min.   : 5.000   Min.   :2.200   Min.   :1.300   Min.   :
2.600
 1st Qu.: 25.75   1st Qu.: 6.575   1st Qu.:3.275   1st Qu.:4.250   1st Qu.:
4.600
 Median : 50.50   Median : 8.000   Median :3.600   Median :5.400   Median :
5.450
 Mean   : 50.50   Mean   : 7.810   Mean   :3.672   Mean   :5.365   Mean   :
5.442
 3rd Qu.: 75.25   3rd Qu.: 9.100   3rd Qu.:3.925   3rd Qu.:6.625   3rd Qu.:
6.325
 Max.   :100.00   Max.   :10.000   Max.   :5.700   Max.   :8.500   Max.   :
7.800
  Advertising       ProdLine      SalesFImage      ComPricing      WartyCla
im
 Min.   :1.900   Min.   :2.300   Min.   :2.900   Min.   :3.700   Min.   :4.
100
 1st Qu.:3.175   1st Qu.:4.700   1st Qu.:4.500   1st Qu.:5.875   1st Qu.:5.
400
 Median :4.000   Median :5.750   Median :4.900   Median :7.100   Median :6.
100
 Mean   :4.010   Mean   :5.805   Mean   :5.123   Mean   :6.974   Mean   :6.
043
 3rd Qu.:4.800   3rd Qu.:6.800   3rd Qu.:5.800   3rd Qu.:8.400   3rd Qu.:6.
600
 Max.   :6.500   Max.   :8.400   Max.   :8.200   Max.   :9.900   Max.   :8.
100
   OrdBilling       DelSpeed      Satisfaction
 Min.   :2.000   Min.   :1.600   Min.   :4.700
 1st Qu.:3.700   1st Qu.:3.400   1st Qu.:6.000
 Median :4.400   Median :3.900   Median :7.050
 Mean   :4.278   Mean   :3.886   Mean   :6.918
 3rd Qu.:4.800   3rd Qu.:4.425   3rd Qu.:7.625
 Max.   :6.700   Max.   :5.500   Max.   :9.900
> dim(Market)
[1] 100  13
> colnames(Market)
 [1] "ID"           "ProdQual"     "Ecom"         "TechSup"      "CompRes"
 [6] "Advertising"  "ProdLine"     "SalesFImage"  "ComPricing"   "WartyClai
m"
[11] "OrdBilling"   "DelSpeed"     "Satisfaction"
> str(Market)
'data.frame':     100 obs. of  13 variables:
 $ ID          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ ProdQual    : num  8.5 8.2 9.2 6.4 9 6.5 6.9 6.2 5.8 6.4 ...
 $ Ecom        : num  3.9 2.7 3.4 3.3 3.4 2.8 3.7 3.3 3.6 4.5 ...
 $ TechSup     : num  2.5 5.1 5.6 7 5.2 3.1 5 3.9 5.1 5.1 ...
 $ CompRes     : num  5.9 7.2 5.6 3.7 4.6 4.1 2.6 4.8 6.7 6.1 ...
 $ Advertising : num  4.8 3.4 5.4 4.7 2.2 4 2.1 4.6 3.7 4.7 ...
 $ ProdLine    : num  4.9 7.9 7.4 4.7 6 4.3 2.3 3.6 5.9 5.7 ...
 $ SalesFImage : num  6 3.1 5.8 4.5 4.5 3.7 5.4 5.1 5.8 5.7 ...
 $ ComPricing  : num  6.8 5.3 4.5 8.8 6.8 8.5 8.9 6.9 9.3 8.4 ...
 $ WartyClaim  : num  4.7 5.5 6.2 7 6.1 5.1 4.8 5.4 5.9 5.4 ...
 $ OrdBilling  : num  5 3.9 5.4 4.3 4.5 3.6 2.1 4.3 4.4 4.1 ...
 $ DelSpeed    : num  3.7 4.9 4.5 3 3.5 3.3 2 3.7 4.6 4.4 ...
 $ Satisfaction: num  8.2 5.7 8.9 4.8 7.1 4.7 5.7 6.3 7 5.5 ...
> head(Market)
  ID ProdQual Ecom TechSup CompRes Advertising ProdLine SalesFImage ComPric
ing WartyClaim
1  1      8.5  3.9     2.5     5.9         4.8      4.9         6.0
6.8        4.7
```

```
2  2       8.2  2.7      5.1      7.2          3.4       7.9          3.1
5.3         5.5
3  3       9.2  3.4      5.6      5.6          5.4       7.4          5.8
4.5         6.2
4  4       6.4  3.3      7.0      3.7          4.7       4.7          4.5
8.8         7.0
5  5       9.0  3.4      5.2      4.6          2.2       6.0          4.5
6.8         6.1
6  6       6.5  2.8      3.1      4.1          4.0       4.3          3.7
8.5         5.1
  OrdBilling DelSpeed Satisfaction
1        5.0      3.7          8.2
2        3.9      4.9          5.7
3        5.4      4.5          8.9
4        4.3      3.0          4.8
5        4.5      3.5          7.1
6        3.6      3.3          4.7
> tail(Market)
     ID ProdQual Ecom TechSup CompRes Advertising ProdLine SalesFImage ComP
ricing WartyClaim
95   95       9.3  3.8      4.0      4.6          4.7       6.4          5.5
7.4         5.3
96   96       8.6  4.8      5.6      5.3          2.3       6.0          5.7
6.7         5.8
97   97       7.4  3.4      2.6      5.0          4.1       4.4          4.8
7.2         4.5
98   98       8.7  3.2      3.3      3.2          3.1       6.1          2.9
5.6         5.0
99   99       7.8  4.9      5.8      5.3          5.2       5.3          7.1
7.9         6.0
100 100       7.9  3.0      4.4      5.1          5.9       4.2          4.8
9.7         5.7
    OrdBilling DelSpeed Satisfaction
95         3.6      3.4          7.7
96         4.9      3.6          7.3
97         4.2      3.7          6.3
98         3.1      2.5          5.4
99         4.3      3.9          6.4
100        3.4      3.5          6.4
> #### Dataset transformation ###
> ### Changing the Variable Names according to the given list ##
> colnames(Market) = c("Product ID","Product Quality",
+                      "E-Commerce", "Technical Support",
+                      "Complaint Resolution","Advertising",
+                      "Product Line","Salesforce Image",
+                      "Competitive Pricing","Warranty & Claims",
+                      "Order & Billing","Delivery Speed",
+                      "Customer Satisfaction")
> colnames(Market)
 [1] "Product ID"            "Product Quality"       "E-Commerce"
 [4] "Technical Support"     "Complaint Resolution"  "Advertising"
 [7] "Product Line"          "Salesforce Image"      "Competitive Pricing"
[10] "Warranty & Claims"     "Order & Billing"       "Delivery Speed"
[13] "Customer Satisfaction"
> ### Removing the ID variable ###
> Market = Market[,-1]
> ###Creatin a new dataset Market1 without the Satisfaction Variable###
> Market1 = Market[,-12]
> colnames(Market1)
 [1] "Product Quality"      "E-Commerce"           "Technical Support"
 [4] "Complaint Resolution" "Advertising"          "Product Line"
```

```
 [7] "Salesforce Image"     "Competitive Pricing"  "Warranty & Claims"
[10] "Order & Billing"      "Delivery Speed"
> ### Uni-Variate Analysis ###
> hist.data.frame(Market1)
> ggplot(melt(Market1), aes(variable, value)) + geom_boxplot()
No id variables; using all as measure variables
> hist(Market$`Customer Satisfaction`,
+      main = "Histogram of Customer Satisfaction",
+      xlab = "Ratings")
> boxplot(Market$`Customer Satisfaction`,
+         main = "Boxplot of Customer Satisfaction")
> ### Bi - Variate Analysis ###
> ### CUSTOMER SATISFACTION ###
> CUSTOMER.SATISFACTION= cut(Market$`Customer Satisfaction`,3,labels = c("S
ad","Not Happy","Happy"))
> qplot(CUSTOMER.SATISFACTION,fill = CUSTOMER.SATISFACTION)
> tab = table(CUSTOMER.SATISFACTION)
> View(prop.table(tab)*100)
> ### PRODUCT QUALITY ####
> PRODUCT.QUALITY = cut(Market$`Product Quality`,3,labels = c("Sad","Not Ha
ppy","Happy"))
> tab1 = table(PRODUCT.QUALITY)
> View(prop.table(tab1)*100)
> qplot(PRODUCT.QUALITY,fill = PRODUCT.QUALITY)
> ### E-COMMERCE ###
> E.COMMERCE = cut(Market$`E-Commerce`,3,labels = c("Sad","Not Happy","Happ
y"))
> tab2 = table(E.COMMERCE)
> View(prop.table(tab2)*100)
> qplot(E.COMMERCE,fill = E.COMMERCE)
> ### Missing Value Treatment ####
> sum(is.na(Market))
[1] 0
> ### Outlier identification ###
> outlier2 = function(i)
+ {
+   IQ = IQR(Market[,i])
+   z = quantile(Market[,i])
+   z= as.data.frame(z)
+   colnames(z) = as.factor(colnames(z))
+   q1 = z[2,1]
+   q3 = z[4,1]
+
+   Market2 = Market
+   subset1 = Market2[Market2[,i] < q1 - 1.5*IQ,]
+   lo = nrow(subset1)
+   subset2 = Market2[Market2[,i] > q3 + 1.5*IQ,]
+   hi = nrow(subset2)
+   Outliers = lo + hi
+   print(colnames(Market[i]))
+   Outliers
+ }
> ### outlier identification ###
> outlier2(2)
[1] "E-Commerce"
[1] 7
> outlier2(7)
[1] "Salesforce Image"
[1] 3
> outlier2(10)
[1] "Order & Billing"
```

```
[1] 4
> outlier2(10)
[1] "Order & Billing"
[1] 4
> ### Correlation Matrix and Correlation Plot ###
> cor.plot(Market1,numbers = TRUE)
> View(cor(Market1))
> ### Checking the Eigen Values ####
> Eigen = eigen(cor(Market1))
> Eigen$values
 [1] 3.42697133 2.55089671 1.69097648 1.08655606 0.60942409 0.55188378 0.40
151815 0.24695154
 [9] 0.20355327 0.13284158 0.09842702
> ### Checking the Regression model ###
> p = lm(`Customer Satisfaction`~., data = Market)
> summary(p)

Call:
lm(formula = `Customer Satisfaction` ~ ., data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-1.43005 -0.31165  0.07621  0.37190  0.90120

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           -0.66961    0.81233  -0.824  0.41199
`Product Quality`      0.37137    0.05177   7.173 2.18e-10 ***
`E-Commerce`          -0.44056    0.13396  -3.289  0.00145 **
`Technical Support`    0.03299    0.06372   0.518  0.60591
`Complaint Resolution` 0.16703    0.10173   1.642  0.10416
Advertising           -0.02602    0.06161  -0.422  0.67382
`Product Line`         0.14034    0.08025   1.749  0.08384 .
`Salesforce Image`     0.80611    0.09775   8.247 1.45e-12 ***
`Competitive Pricing` -0.03853    0.04677  -0.824  0.41235
`Warranty & Claims`   -0.10298    0.12330  -0.835  0.40587
`Order & Billing`      0.14635    0.10367   1.412  0.16160
`Delivery Speed`       0.16570    0.19644   0.844  0.40124
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5623 on 88 degrees of freedom
Multiple R-squared:  0.8021,  Adjusted R-squared:  0.7774
F-statistic: 32.43 on 11 and 88 DF,  p-value: < 2.2e-16

> ### Checking the VIF values ###
> vifmatrix = vif(p)
> summary(vifmatrix)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.509   2.196   2.977   3.163   3.464   6.516
> colnames(Market)
 [1] "Product Quality"       "E-Commerce"            "Technical Support"
 [4] "Complaint Resolution"  "Advertising"           "Product Line"
 [7] "Salesforce Image"      "Competitive Pricing"   "Warranty & Claims"
[10] "Order & Billing"       "Delivery Speed"        "Customer Satisfaction
"
> ### Product Quality ###
> slm1 = lm(`Customer Satisfaction` ~ `Product Quality`,data = Market)
> summary(slm1)

Call:
```

```
lm(formula = `Customer Satisfaction` ~ `Product Quality`, data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-1.88746 -0.72711 -0.01577  0.85641  2.25220

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       3.67593    0.59765   6.151 1.68e-08 ***
`Product Quality`  0.41512    0.07534   5.510 2.90e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.047 on 98 degrees of freedom
Multiple R-squared:  0.2365,  Adjusted R-squared:  0.2287
F-statistic: 30.36 on 1 and 98 DF,  p-value: 2.901e-07

> p1 = predict(slm1)
> plot(p1,col = "Red")
> plot(Market$'Product Quality',col = "Blue")
> lines(p1,col = "Red")
> lines(Market$`Product Quality`,col = "Blue")
> confint(slm1)
                     2.5 %    97.5 %
(Intercept)       2.4899022 4.8619486
`Product Quality` 0.2656059 0.5646309
> ggplot(Market,aes(x = `Customer Satisfaction`,y = `Product Quality`))+geo
m_point(col = "Black")+stat_smooth(method = "lm",col = "White")
> ### E-Commerce ###
> slm2 = lm(`Customer Satisfaction` ~ `E-Commerce`,data = Market)
> summary(slm2)

Call:
lm(formula = `Customer Satisfaction` ~ `E-Commerce`, data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-2.37200 -0.78971  0.04959  0.68085  2.34580

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.1516     0.6161   8.361 4.28e-13 ***
`E-Commerce`   0.4811     0.1649   2.918  0.00437 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.149 on 98 degrees of freedom
Multiple R-squared:  0.07994, Adjusted R-squared:  0.07056
F-statistic: 8.515 on 1 and 98 DF,  p-value: 0.004368

> p2 = predict(slm2)
> plot(p2,col = "Red")
> plot(Market$`E-Commerce`,col = "Blue")
> lines(p2,col = "Red")
  invalid graphics state
> lines(Market$`E-Commerce`,col = "Blue")
> confint(slm2)
                 2.5 %    97.5 %
(Intercept)  3.9288501 6.3742852
`E-Commerce` 0.1539118 0.8081973
```

```
> ggplot(Market,aes(x = `Customer Satisfaction`,y = `E-Commerce`))+geom_poi
nt(col = "Blue")+stat_smooth(method = "lm",col = "Black")
> ### Technical Support ###
> slm3 = lm(`Customer Satisfaction` ~ `Technical Support`,data = Market)
> summary(slm3)

Call:
lm(formula = `Customer Satisfaction` ~ `Technical Support`, data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-2.26136 -0.93297  0.04302  0.82501  2.85617

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         6.44757    0.43592  14.791   <2e-16 ***
`Technical Support`  0.08768    0.07817   1.122    0.265
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.19 on 98 degrees of freedom
Multiple R-squared:  0.01268, Adjusted R-squared:  0.002603
F-statistic: 1.258 on 1 and 98 DF,  p-value: 0.2647

> p3 = predict(slm3)
> plot(p3,col = "Red")
> plot(Market$`Technical Support`,col = "Blue")
> lines(p3,col = "Red")
> lines(Market$`Technical Support`,col = "Blue")
> confint(slm3)
                         2.5 %    97.5 %
(Intercept)          5.58250018 7.3126423
`Technical Support` -0.06743136 0.2428009
> ggplot(Market,aes(x = `Customer Satisfaction`,y = `Technical Support`))+g
eom_point(col = "Blue")+stat_smooth(method = "lm",col = "White")
> ### Complaint Resolution ###
> slm4 = lm(`Customer Satisfaction` ~ `Complaint Resolution`,data = Market)
> summary(slm4)

Call:
lm(formula = `Customer Satisfaction` ~ `Complaint Resolution`,
    data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-2.40450 -0.66164  0.04499  0.63037  2.70949

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           3.68005    0.44285   8.310 5.51e-13 ***
`Complaint Resolution`  0.59499    0.07946   7.488 3.09e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9554 on 98 degrees of freedom
Multiple R-squared:  0.3639,  Adjusted R-squared:  0.3574
F-statistic: 56.07 on 1 and 98 DF,  p-value: 3.085e-11

> p4 = predict(slm4)
> plot(p4,col = "Red")
> plot(Market$`Complaint Resolution`,col = "Blue")
```

```
> lines(p4,col = "Red")
> lines(Market$`Complaint Resolution`,col = "Blue")
> confint(slm4)
                            2.5 %    97.5 %
(Intercept)            2.8012286 4.5588623
`Complaint Resolution` 0.4373084 0.7526786
> ggplot(Market,aes(x = `Customer Satisfaction`,y = `Complaint Resolution`)
)+geom_point(col = "Blue")+stat_smooth(method = "lm",col = "Red")
> ### Advertising ###
> slm5 = lm(`Customer Satisfaction` ~ `Advertising`,data = Market)
> summary(slm5)

Call:
lm(formula = `Customer Satisfaction` ~ Advertising, data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-2.34033 -0.92755  0.05577  0.79773  2.53412

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.6259     0.4237  13.279  < 2e-16 ***
Advertising   0.3222     0.1018   3.167  0.00206 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.141 on 98 degrees of freedom
Multiple R-squared:  0.09282, Adjusted R-squared:  0.08357
F-statistic: 10.03 on 1 and 98 DF,  p-value: 0.002056

> p5 = predict(slm5)
> plot(p5,col = "Red")
> plot(Market$`Advertising`,col = "Blue")

> lines(p5,col = "Red")
> lines(Market$`Advertising`,col = "Blue")
> confint(slm5)
                2.5 %    97.5 %
(Intercept) 4.7851364 6.4667052
Advertising 0.1202881 0.5241405
> ggplot(Market,aes(x = `Customer Satisfaction`,y = `Advertising`))+geom_po
int(col = "Black")+stat_smooth(method = "lm",col = "Green")
> ### Product Line ###
> slm6 = lm(`Customer Satisfaction` ~ `Product Line`,data = Market)
> summary(slm6)

Call:
lm(formula = `Customer Satisfaction` ~ `Product Line`, data = Market)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3634 -0.7795  0.1097  0.7604  1.7373

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.02203    0.45471   8.845 3.87e-14 ***
`Product Line` 0.49887    0.07641   6.529 2.95e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1 on 98 degrees of freedom
```

```
Multiple R-squared:  0.3031,  Adjusted R-squared:  0.296
F-statistic: 42.62 on 1 and 98 DF,  p-value: 2.953e-09

> p6 = predict(slm6)
> plot(p6,col = "Red")
> plot(Market$`Product Line`,col = "Blue")
> lines(p6,col = "Red")
> lines(Market$`Product Line`,col = "Blue")
> confint(slm6)
                   2.5 %    97.5 %
(Intercept)    3.1196708 4.9243955
`Product Line` 0.3472346 0.6505145
> ggplot(Market,aes(x = `Customer Satisfaction`,y = `Product Line`))+geom p
oint(col = "Black")+stat_smooth(method = "lm",col = "Dark Blue")
> ### Salesforce Image ###
> slm7 = lm(`Customer Satisfaction` ~ `Salesforce Image`,data = Market)
> summary(slm7)

Call:
lm(formula = `Customer Satisfaction` ~ `Salesforce Image`, data = Market)

Residuals:
    Min      1Q  Median      3Q     Max
-2.2164 -0.5884  0.1838  0.6922  2.0728

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        4.06983    0.50874   8.000 2.54e-12 ***
`Salesforce Image`  0.55596    0.09722   5.719 1.16e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.037 on 98 degrees of freedom
Multiple R-squared:  0.2502,  Adjusted R-squared:  0.2426
F-statistic:  32.7 on 1 and 98 DF,  p-value: 1.164e-07

> p7 = predict(slm7)
> plot(p7,col = "Red")
> plot(Market$`Salesforce Image`,col = "Blue")
> lines(p7,col = "Red")
> lines(Market$`Salesforce Image`,col = "Blue")
> confint(slm7)
                      2.5 %    97.5 %
(Intercept)        3.0602511 5.0794074
`Salesforce Image` 0.3630295 0.7488857
> ggplot(Market,aes(x = `Customer Satisfaction`,y = `Salesforce Image`))+ge
om_point(col = "Black")+stat_smooth(method = "lm",col = "Brown")
> ### Competitive Pricing ###
> slm8 = lm(`Customer Satisfaction` ~ `Competitive Pricing`,data = Market)
> summary(slm8)

Call:
lm(formula = `Customer Satisfaction` ~ `Competitive Pricing`,
    data = Market)

Residuals:
    Min      1Q  Median      3Q     Max
-1.9728 -0.9915 -0.1156  0.9111  2.5845

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)            8.03856    0.54427  14.769   <2e-16 ***
`Competitive Pricing` -0.16068    0.07621  -2.108   0.0376 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.172 on 98 degrees of freedom
Multiple R-squared:  0.04339, Adjusted R-squared:  0.03363
F-statistic: 4.445 on 1 and 98 DF,  p-value: 0.03756

> p8 = predict(slm8)
> plot(p8,col = "Red")
> plot(Market$`Competitive Pricing`,col = "Blue")
> lines(p8,col = "Red")
> lines(Market$`Competitive Pricing`,col = "Blue")
> confint(slm8)
                           2.5 %       97.5 %
(Intercept)            6.9584773  9.118646619
`Competitive Pricing` -0.3119192 -0.009434976
> ggplot(Market,aes(x = `Customer Satisfaction`,y = `Competitive Pricing`))
+geom_point(col = "Red")+stat_smooth(method = "lm",col = "Green")
> ### Warranty & Claims ###
> slm9 = lm(`Customer Satisfaction` ~ `Warranty & Claims`,data = Market)
> summary(slm9)

Call:
lm(formula = `Customer Satisfaction` ~ `Warranty & Claims`, data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-2.36504 -0.90202  0.03019  0.90763  2.88985

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)          5.3581     0.8813   6.079 2.32e-08 ***
`Warranty & Claims`  0.2581     0.1445   1.786   0.0772 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.179 on 98 degrees of freedom
Multiple R-squared:  0.03152, Adjusted R-squared:  0.02164
F-statistic:  3.19 on 1 and 98 DF, p-value: 0.0772

> p9 = predict(slm9)
> plot(p9,col = "Red")
> plot(Market$`Warranty & Claims`,col = "Blue")
> lines(p9,col = "Red")
> lines(Market$`Warranty & Claims`,col = "Blue")
> confint(slm9)
                         2.5 %   97.5 %
(Intercept)          3.6090720 7.107082
`Warranty & Claims` -0.0286887 0.544963
> ggplot(Market,aes(x = `Customer Satisfaction`,y = `Warranty & Claims`))+g
eom_point(col = "Black")+stat_smooth(method = "lm",col = "Maroon")
> ### Order & Billing ###
> slm10 = lm(`Customer Satisfaction` ~ `Order & Billing`,data = Market)
> summary(slm10)

Call:
lm(formula = `Customer Satisfaction` ~ `Order & Billing`, data = Market)

Residuals:
```

```
    Min      1Q  Median      3Q     Max
-2.4005 -0.7071 -0.0344  0.7340  2.9673

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         4.0541     0.4840   8.377 3.96e-13 ***
`Order & Billing`   0.6695     0.1106   6.054 2.60e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.022 on 98 degrees of freedom
Multiple R-squared:  0.2722,  Adjusted R-squared:  0.2648
F-statistic: 36.65 on 1 and 98 DF,  p-value: 2.602e-08

> p10 = predict(slm10)
> plot(p10,col = "Red")
> plot(Market$`Order & Billing`,col = "Blue")
> lines(p10,col = "Red")
> lines(Market$`Order & Billing`,col = "Blue")
> confint(slm10)
                     2.5 %    97.5 %
(Intercept)       3.093639 5.0144660
`Order & Billing` 0.450021 0.8888978
> ggplot(Market,aes(x = `Customer Satisfaction`,y = `Order & Billing`))+geo
m_point(col = "Red")+stat_smooth(method = "lm",col = "Yellow")
> ### Delivery Speed ###
> slm11 = lm(`Customer Satisfaction` ~ `Delivery Speed`,data = Market)
> summary(slm11)

Call:
lm(formula = `Customer Satisfaction` ~ `Delivery Speed`, data = Market)

Residuals:
     Min       1Q   Median       3Q      Max
-2.22475 -0.54846  0.08796  0.54462  2.59432

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)        3.2791     0.5294   6.194 1.38e-08 ***
`Delivery Speed`   0.9364     0.1339   6.994 3.30e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9783 on 98 degrees of freedom
Multiple R-squared:  0.333,   Adjusted R-squared:  0.3262
F-statistic: 48.92 on 1 and 98 DF,  p-value: 3.3e-10

> p11 = predict(slm11)
> plot(p11,col = "Red")
> plot(Market$`Delivery Speed`,col = "Blue")
> lines(p11,col = "Red")
  invalid graphics state
> lines(Market$`Delivery Speed`,col = "Blue")
> confint(slm11)
                     2.5 %   97.5 %
(Intercept)      2.2285309 4.329613
`Delivery Speed` 0.6707367 1.202103
> ggplot(Market,aes(x = `Customer Satisfaction`,y = `Delivery Speed`))+geom
_point(col = "Blue")+stat_smooth(method = "lm",col = "Yellow")
> ### Question 4 ###
> ### Barlett Test ####
```

```
> bartlett.test(Market)

        Bartlett test of homogeneity of variances

data:  Market
Bartlett's K-squared = 146.52, df = 11, p-value < 2.2e-16

> ### Principal Component Analysis ###
> EigenPCA = eigen(cor(Market1))
> EigenPCA$values
 [1] 3.42697133 2.55089671 1.69097648 1.08655606 0.60942409 0.55188378 0.40
151815 0.24695154
 [9] 0.20355327 0.13284158 0.09842702
> Var.Eigen<-EigenPCA$values/sum(EigenPCA$values)*100
> cumsum(Var.Eigen)
 [1]  31.15428  54.34425  69.71677  79.59455  85.13477  90.15189  93.80206
96.04707
 [9]  97.89756  99.10521 100.00000
> ### Factor Analysis ###
> EigenFA = eigen(cor(Market1))
> EigenFA$values
 [1] 3.42697133 2.55089671 1.69097648 1.08655606 0.60942409 0.55188378 0.40
151815 0.24695154
 [9] 0.20355327 0.13284158 0.09842702
> plot(EigenFA$values,col = "Blue",main = "Scree Plot for Factor Analysis")
> abline(a = 1,b = 0,col = "Red")
> ### Without Rotation ###
> UnrotatedFA = principal(Market1,nfactors = 4,rotate = "none")
> print(UnrotatedFA)
Principal Components Analysis
Call: principal(r = Market1, nfactors = 4, rotate = "none")
Standardized loadings (pattern matrix) based upon correlation matrix
                       PC1   PC2   PC3   PC4   h2    u2 com
Product Quality       0.25 -0.50 -0.08  0.67 0.77 0.232 2.2
E-Commerce            0.31  0.71  0.31  0.28 0.78 0.223 2.1
Technical Support     0.29 -0.37  0.79 -0.20 0.89 0.107 1.9
Complaint Resolution  0.87  0.03 -0.27 -0.22 0.88 0.119 1.3
Advertising           0.34  0.58  0.11  0.33 0.58 0.424 2.4
Product Line          0.72 -0.45 -0.15  0.21 0.79 0.213 2.0
Salesforce Image      0.38  0.75  0.31  0.23 0.86 0.141 2.1
Competitive Pricing  -0.28  0.66 -0.07 -0.35 0.64 0.359 1.9
Warranty & Claims     0.39 -0.31  0.78 -0.19 0.89 0.108 2.0
Order & Billing       0.81  0.04 -0.22 -0.25 0.77 0.234 1.3
Delivery Speed        0.88  0.12 -0.30 -0.21 0.91 0.086 1.4

                       PC1  PC2  PC3  PC4
SS loadings           3.43 2.55 1.69 1.09
Proportion Var        0.31 0.23 0.15 0.10
Cumulative Var        0.31 0.54 0.70 0.80
Proportion Explained  0.39 0.29 0.19 0.12
Cumulative Proportion 0.39 0.68 0.88 1.00

Mean item complexity =  1.9
Test of the hypothesis that 4 components are sufficient.

The root mean square of the residuals (RMSR) is  0.06
 with the empirical chi square  39.02  with prob <  0.0018

Fit based upon off diagonal values = 0.97
> print(UnrotatedFA$loadings,cutoff = 0.4)
```

```
Loadings:
                     PC1    PC2    PC3    PC4
Product Quality             -0.501         0.670
E-Commerce                   0.713
Technical Support                    0.794
Complaint Resolution  0.871
Advertising                  0.581
Product Line          0.716 -0.455
Salesforce Image             0.752
Competitive Pricing          0.660
Warranty & Claims                    0.778
Order & Billing       0.809
Delivery Speed        0.876

                 PC1   PC2   PC3   PC4
SS loadings    3.427 2.551 1.691 1.087
Proportion Var 0.312 0.232 0.154 0.099
Cumulative Var 0.312 0.543 0.697 0.796
> ### With Rotation ###
> RotatedFA = principal(Market1, nfactors = 4 , rotate = "varimax")
> print(RotatedFA)
Principal Components Analysis
Call: principal(r = Market1, nfactors = 4, rotate = "varimax")
Standardized loadings (pattern matrix) based upon correlation matrix
                      RC1   RC2   RC3   RC4   h2    u2 com
Product Quality      0.00 -0.01 -0.03  0.88 0.77 0.232 1.0
E-Commerce           0.06  0.87  0.05 -0.12 0.78 0.223 1.1
Technical Support    0.02 -0.02  0.94  0.10 0.89 0.107 1.0
Complaint Resolution 0.93  0.12  0.05  0.09 0.88 0.119 1.1
Advertising          0.14  0.74 -0.08  0.01 0.58 0.424 1.1
Product Line         0.59 -0.06  0.15  0.64 0.79 0.213 2.1
Salesforce Image     0.13  0.90  0.08 -0.16 0.86 0.141 1.1
Competitive Pricing -0.09  0.23 -0.25 -0.72 0.64 0.359 1.5
Warranty & Claims    0.11  0.05  0.93  0.10 0.89 0.108 1.1
Order & Billing      0.86  0.11  0.08  0.04 0.77 0.234 1.1
Delivery Speed       0.94  0.18  0.00  0.05 0.91 0.086 1.1

                       RC1  RC2  RC3  RC4
SS loadings           2.89 2.23 1.86 1.77
Proportion Var        0.26 0.20 0.17 0.16
Cumulative Var        0.26 0.47 0.63 0.80
Proportion Explained  0.33 0.26 0.21 0.20
Cumulative Proportion 0.33 0.59 0.80 1.00

Mean item complexity =  1.2
Test of the hypothesis that 4 components are sufficient.

The root mean square of the residuals (RMSR) is  0.06
 with the empirical chi square  39.02  with prob <  0.0018

Fit based upon off diagonal values = 0.97


> ### FA Diagram ####
> fa.diagram(RotatedFA)
> ###Using Factor Analysiss ###
> ### Extracting the Factor scores and Naming them ###
> Market.factors = data.frame(Market$`Customer Satisfaction`,RotatedFA$scor
es)
> colnames(Market.factors) = c("Customer Satisfaction",
+                              "Shopping Experience",
```

```
+                               "Product Marketing",
+                               "After Sales Assistance",
+                               "Valuation of the product")
> str(Market.factors)
'data.frame':  100 obs. of  5 variables:
 $ Customer Satisfaction   : num  8.2 5.7 8.9 4.8 7.1 4.7 5.7 6.3 7 5.5 ...
 $ Shopping Experience     : num  -0.139 1.636 0.356 -1.221 -0.486 ...
 $ Product Marketing       : num  0.94 -2.04 0.876 -0.557 -0.421 ...
 $ After Sales Assistance  : num  -1.724 -0.591 0.016 1.25 -0.031 ...
 $ Valuation of the product: num  0.0968 0.6504 1.3821 -0.6456 0.4755 ...
> ### Checking the Factors for Correlation with Customer Satisfaction and M
ultiCollinearity ###
> cor.plot(Market.factors,numbers = TRUE)
> ### Creating a Multiple Regression Model using the new factors ###
> MR = lm(`Customer Satisfaction`~.,data = Market.factors)
> summary(MR)

Call:
lm(formula = `Customer Satisfaction` ~ ., data = Market.factors)

Residuals:
     Min       1Q   Median       3Q      Max
-1.70781 -0.47233  0.08959  0.41930  1.35539

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                 6.91800    0.06659 103.889  < 2e-16 ***
`Shopping Experience`       0.57991    0.06816   8.508 2.53e-13 ***
`Product Marketing`         0.61811    0.06734   9.180 9.37e-15 ***
`After Sales Assistance`    0.05779    0.07135   0.810     0.42
`Valuation of the product`  0.61117    0.07609   8.033 2.57e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6659 on 95 degrees of freedom
Multiple R-squared:  0.7004,   Adjusted R-squared:  0.6878
F-statistic: 55.54 on 4 and 95 DF,  p-value: < 2.2e-16

> MR1 = lm(`Customer Satisfaction`~`Valuation of the product`+
+            `Product Marketing`+`Shopping Experience`,data = Market.factor
s)
> summary(MR1)

Call:
lm(formula = `Customer Satisfaction` ~ `Valuation of the product` +
    `Product Marketing` + `Shopping Experience`, data = Market.factors)

Residuals:
     Min       1Q   Median       3Q      Max
-1.68562 -0.46163  0.09543  0.40168  1.39498

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                 6.91800    0.06647 104.076  < 2e-16 ***
`Valuation of the product`  0.61448    0.07584   8.102 1.72e-12 ***
`Product Marketing`         0.61903    0.06721   9.211 7.37e-15 ***
`Shopping Experience`       0.57967    0.06804   8.520 2.23e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6647 on 96 degrees of freedom
```

```
Multiple R-squared:  0.6984,   Adjusted R-squared:  0.689
F-statistic: 74.09 on 3 and 96 DF,  p-value: < 2.2e-16

> ###Creation of new interaction effect ###
> Product.Appeal = Market.factors$`Product Marketing`*Market.factors$`Valua
tion of the product`
> MR2 = lm(`Customer Satisfaction`~`Shopping Experience`+`Product Marketing
`+
+             `Valuation of the product`+Product.Appeal,data = Market.facto
rs)
> summary(MR2)

Call:
lm(formula = `Customer Satisfaction` ~ `Shopping Experience` +
    `Product Marketing` + `Valuation of the product` + Product.Appeal,
    data = Market.factors)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6934 -0.4263  0.1142  0.4046  1.3394

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                 6.92506    0.06449 107.374  < 2e-16 ***
`Shopping Experience`       0.60653    0.06672   9.091 1.45e-14 ***
`Product Marketing`         0.55695    0.06917   8.052 2.34e-12 ***
`Valuation of the product`  0.58755    0.07421   7.917 4.50e-12 ***
Product.Appeal              0.19046    0.07126   2.673  0.00885 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6444 on 95 degrees of freedom
Multiple R-squared:  0.7195,   Adjusted R-squared:  0.7077
F-statistic: 60.91 on 4 and 95 DF,  p-value: < 2.2e-16

> ###Plotting against the predicted values ###
> MRP = predict(MR2)
> plot(MRP,col = "Red")
> lines(MRP,col = "Red")
> lines(Market$`Customer Satisfaction`,col = "Blue")
> ### Finding Upper and Lower limits ###
> confint(MR2)
                               2.5 %    97.5 %
(Intercept)                6.79702099 7.0530969
`Shopping Experience`      0.47407918 0.7389864
`Product Marketing`        0.41962818 0.6942659
`Valuation of the product` 0.44022160 0.7348761
Product.Appeal             0.04899422 0.3319226
> ggplot(Market,aes(x = Market$`Customer Satisfaction`,y = MRP))+
+   geom_point(col = "red1")+stat_smooth(method = "lm",col = "blue")
```