FACTOR HAIR RUBRICS

Project 3: Multiple Linear Regression & PCA

Submitted by: Usman Tahir
Course: Data Science & Business Analytics
Institute: McCombs – UT Austin and Greatlearning
Tutor: Sambath Margabandhu

CONTENTS

1 Project Objective

The objective of the project is to use the dataset 'Factor-Hair-Revised.csv' to build an optimum regression model to predict satisfaction. You are expected to

- Perform exploratory data analysis on the dataset. Showcase some charts, graphs. Check for outliers and missing values
- Is there evidence of multicollinearity? Showcase your analysis
- Perform simple linear regression for the dependent variable with every independent variable
- Perform PCA/Factor analysis by extracting 4 factors. Interpret the output and name the Factors
- 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.

2 Assumptions

We take following as our project assumptions to conduct descriptive analysis of the dataset:

- Data is representative of the entire population and free from errors.
- Variables included in the data are significant representation to achieve project objective of multiple linear regression.

3 Exploratory Data Analysis

Exploratory journey usually has eight stages, which we are going to follow in this descriptive analysis expedition.

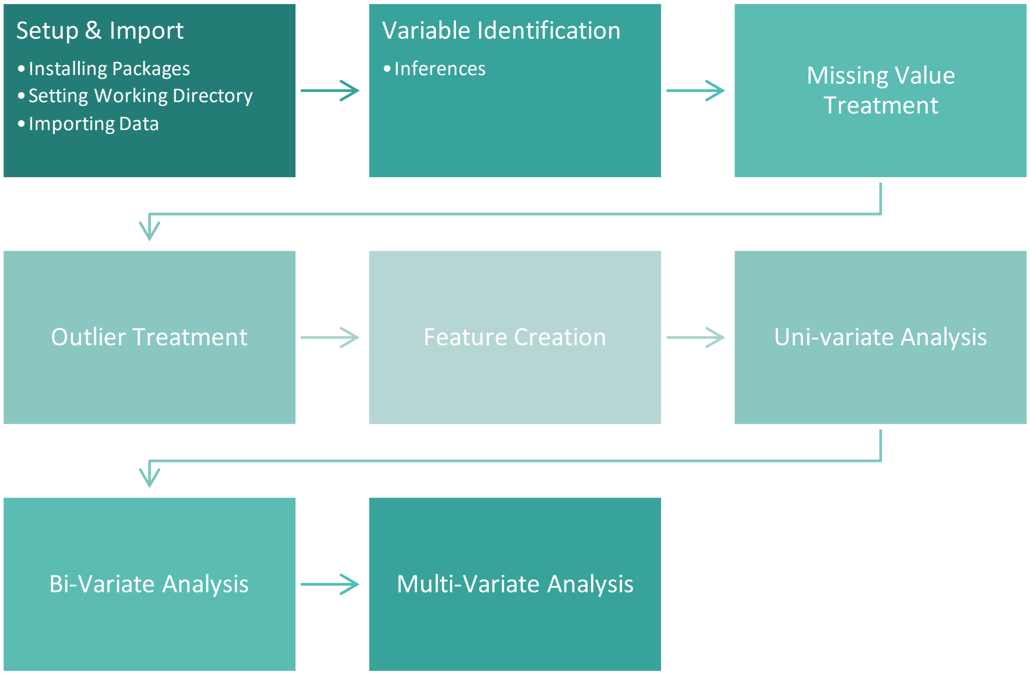


Figure 1: Exploratory Data Analysis Process

Data dictionary of the dataset is mentioned below:

i.	**ProdQual**	– Product Quality
ii.	**Ecom**	– E-commerce
iii.	**TechSup**	– Technical Support
iv.	**CompRes**	– Complaint Resolution
v.	**Advertising**	– Advertising
vi.	**ProdLine**	– Product Line
vii.	**SalesFImage**	– Sales Force Image
viii.	**ComPricing**	– Competitive Pricing
ix.	**WartyClaim**	– Warranty and Claims
x.	**OrdBilling**	– Order and Billing
xi.	**DelSpeed**	– Delivery Speed
xii.	**Satisfaction**	– Customer Satisfaction Score

3.1 Environment Setup and Data Import

This step is used for basically setting the stage for the analysis. Therefore, we will install packages, setting working directory and finally the most important step is to load the dataset:

3.1.1 Installing Packages & Invoking Libraries

We will require below packages for the purpose of analysis, please refer to Step-1 in the Appendix A-Source Code for further details:

LIBRARY	PURPOSE
PACMAN	Package Management – p_load function
READR	Reading CSV files
DATAEXPLORER	Exploring data structure of the dataset
GGPLOT2	Visualization of the insights in data
PYSCH	Correlations through pairs.panels
CORRPLOT	Correlation graphs
NFACTORS	PCA and Factor Analysis
CAR	Variance Inflaction Factor

Table 1: Libraries Usage

3.1.2 Setting up working directory

We are setting the working directory to save all assets of analysis such as plots, tables and other explanatory documents at following path:

```
setwd("C:/DSBA_Course/Proper Learning/Module 3 [Advanced Statistics]/M3 W4 [Project 3]/")
```

Snippet 1: Setting Working Directory

3.1.3 Importing and reading Dataset

Loading the dataset into mydata dataframe through read.csv function:

```
mydata <- read.csv("Factor-Hair-Revised.csv",header = TRUE)
```

Snippet 2: Loading CSV file into variables

3.2 Variable Identification

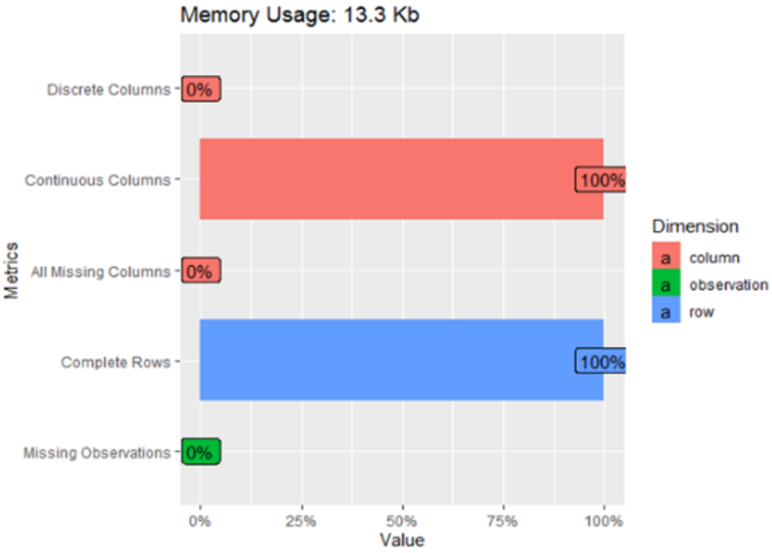
We will be using several R functions to analysis variables in the dataset for below mentioned purpose:

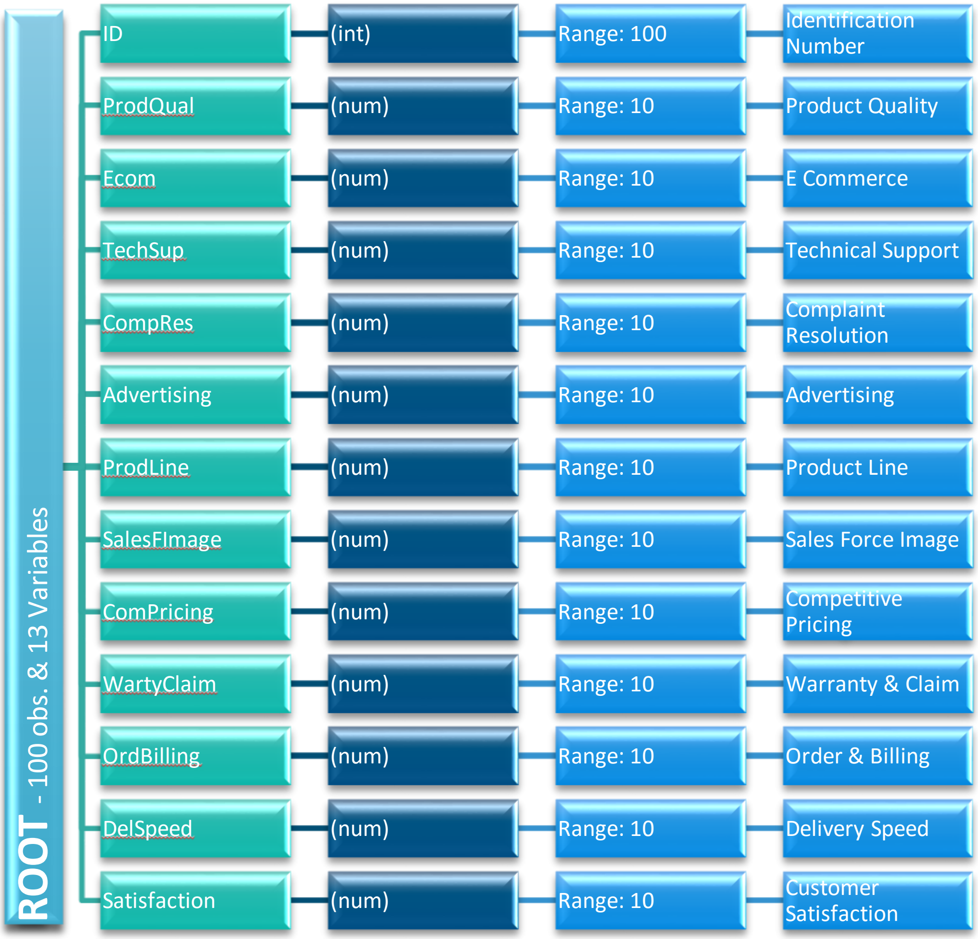
FUNCTION	PURPOSE
DIM()	Identify number of rows and columns in dataset
PLOT_INTRO()	Observations completeness in datasets
NAMES()	Identify if all column names are free from spaces in between
PLOT_STR()	Identify if there are empty values in dataset
HEAD()	Top 6 observations of the dataset
PLOT_MISSING()	Check Missing values
SUMMARY()	5 Number Summaries & Aggregation of variables

Table 2: Functions for Variable Identification

3.2.1 Inferences

- There are 100 rows and 13 columns in the dataset.
- All variables are numeric apart from ID which is integer.
- All variables are continuous with no categorical variable.
- There are no missing values and rows.
- There will be no requirement of data preparation.





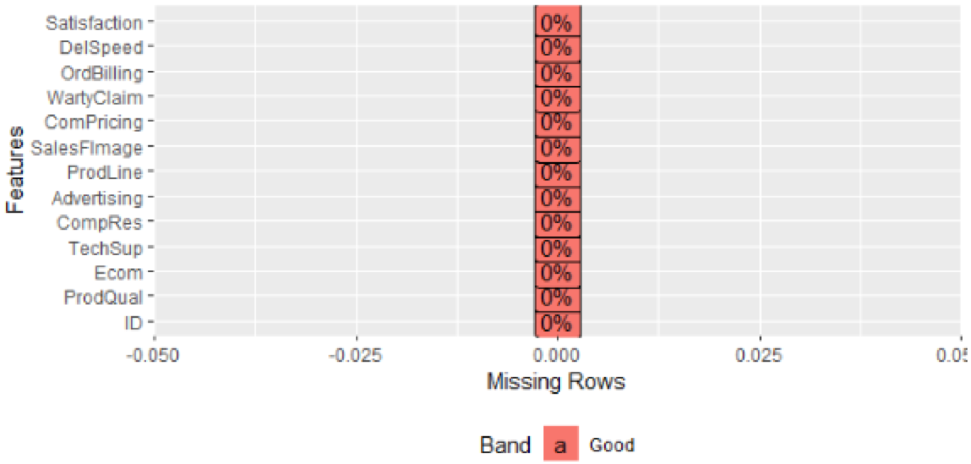
ROOT - 100 obs. & 13 Variables

ID	(int)	Range: 100	Identification Number
ProdQual	(num)	Range: 10	Product Quality
Ecom	(num)	Range: 10	E Commerce
TechSup	(num)	Range: 10	Technical Support
CompRes	(num)	Range: 10	Complaint Resolution
Advertising	(num)	Range: 10	Advertising
ProdLine	(num)	Range: 10	Product Line
SalesFImage	(num)	Range: 10	Sales Force Image
ComPricing	(num)	Range: 10	Competitive Pricing
WartyClaim	(num)	Range: 10	Warranty & Claim
OrdBilling	(num)	Range: 10	Order & Billing
DelSpeed	(num)	Range: 10	Delivery Speed
Satisfaction	(num)	Range: 10	Customer Satisfaction

DATE variable needs to be treated for making it factor rather than integer. Temperature and Season variables seems to be very important for the analysis and may contain interesting insights.

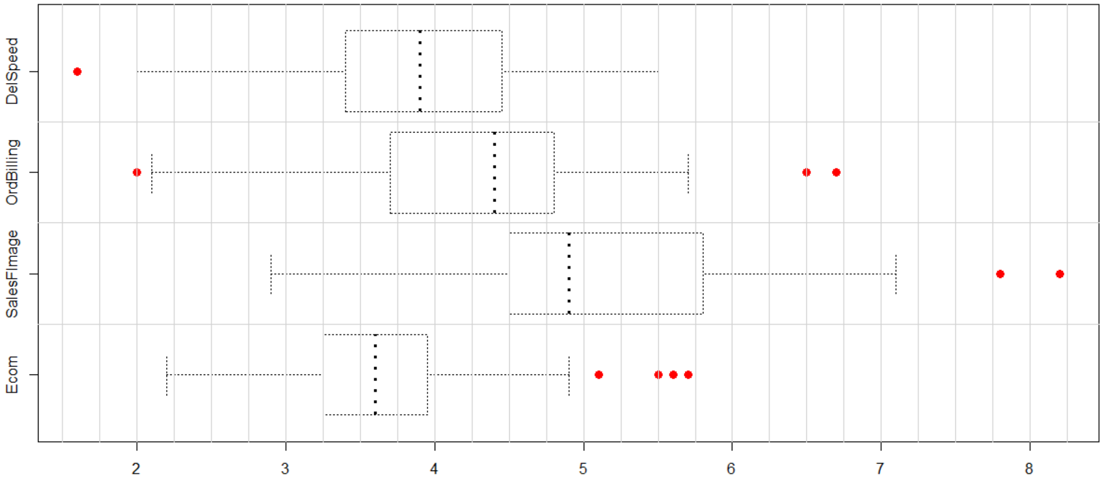
3.3 Missing Value Treatment

There are no missing values in the dataset, therefore there is no requirement of any treatment or data manipulations.



3.4 Outlier Treatment

There are few outliers in four variables of the dataset.

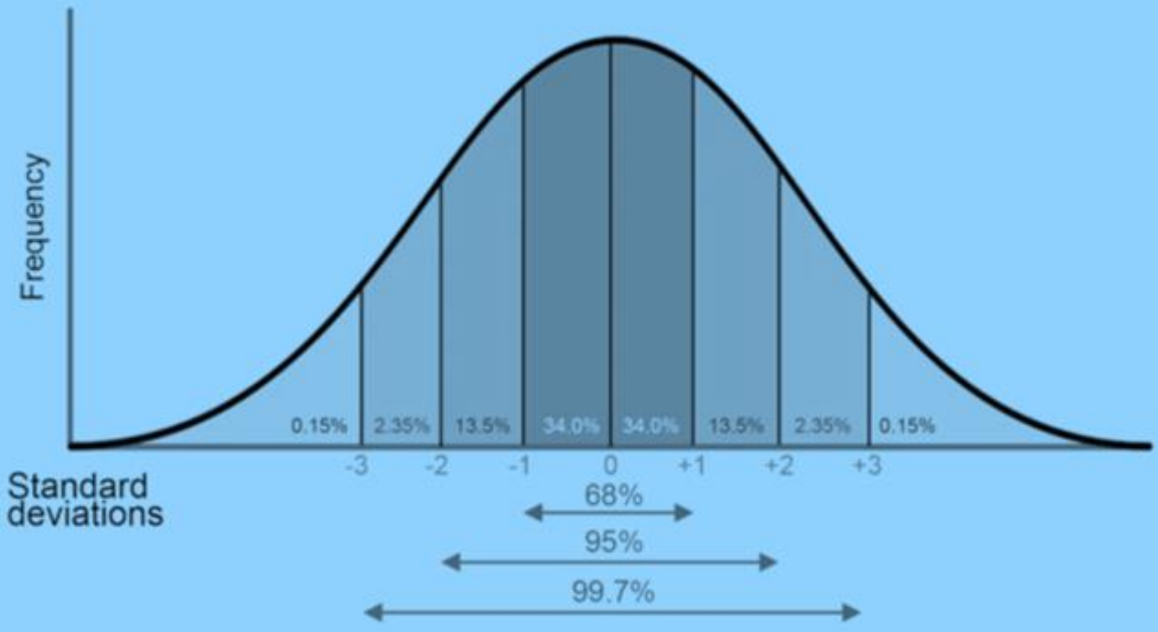


Ecom	SalesFImage	OrdBilling	DelSpeed
5.1	7.8	2.0	1.6
5.1	7.8	2.0	
5.1	8.2	6.5	
5.5		6.7	
5.6			
5.7			

3.5 Variable Transformation/Feature Creation

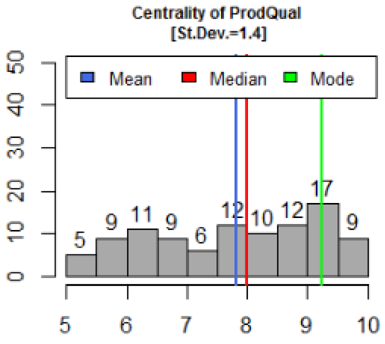
There is no need for creating new variable in the dataset.

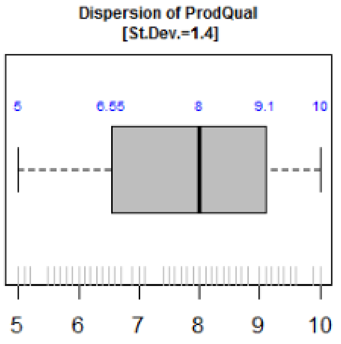
3.6 Univariate Analysis



ProdQual

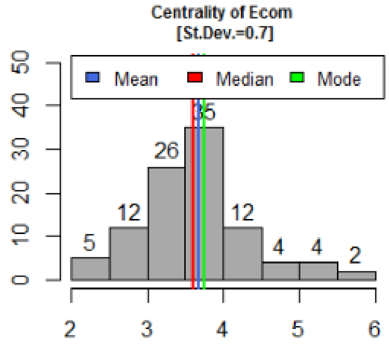
Left skewed with bimodal distribution which indicates there are two different groups of data. It can be due to two different products having different medians. Wide dispersion of about 1.4 standard deviation.

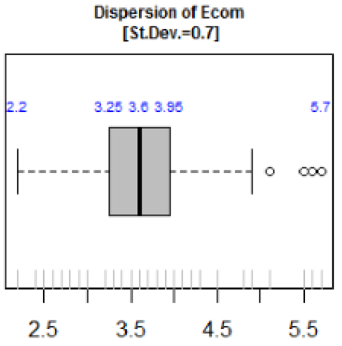

Centrality of ProdQual [St.Dev.=1.4]


Dispersion of ProdQual [St.Dev.=1.4]

Ecom

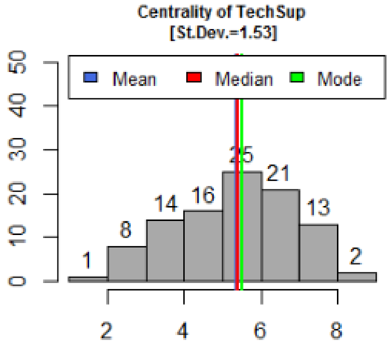
Slightly right skewed with single peak distribution very close to normal. Narrowest dispersion of about 0.7 standard of deviation with some outliers at the upper side.

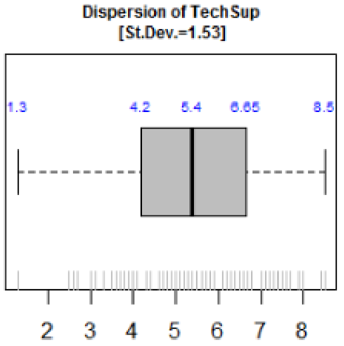

Centrality of Ecom [St.Dev.=0.7]


Dispersion of Ecom [St.Dev.=0.7]

TechSup

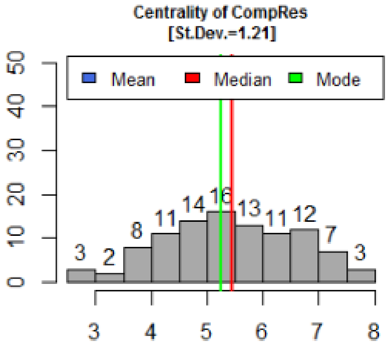
Almost symmetric with single peak distribution very close to normal. widest dispersion of about 1.53 standard of deviation with no outliers.

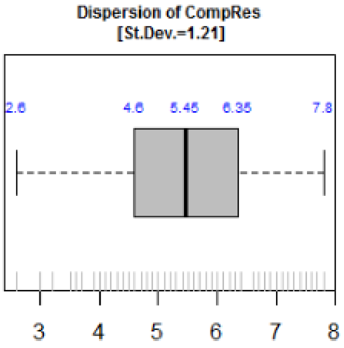

Centrality of TechSup [St.Dev.=1.53]


Dispersion of TechSup [St.Dev.=1.53]

CompRes

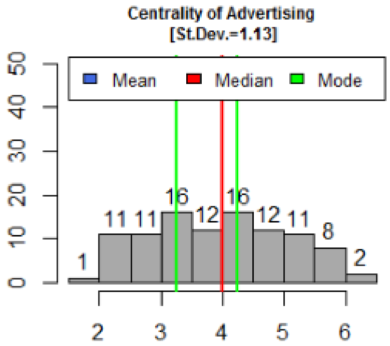
Almost symmetric distribution with moderate dispersion of 1.21 standard deviation. There are no outliers in this variable.

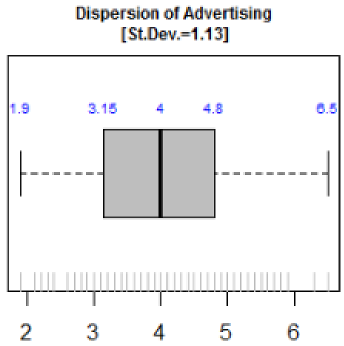

Centrality of CompRes [St.Dev.=1.21]


Dispersion of CompRes [St.Dev.=1.21]

Advertising

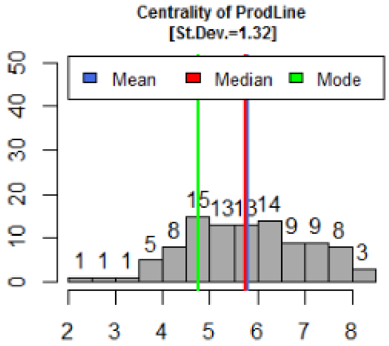
Right skewed with bimodal distribution which indicates there are two different groups of data. It can be due to two different products having different medians. moderately dispersed about 1.4 standard deviation. No outliers.

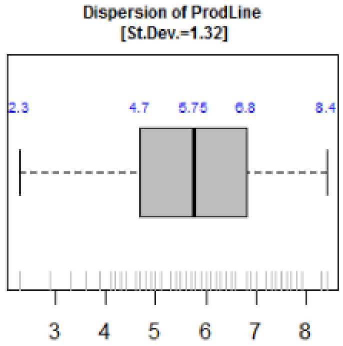




ProdLine

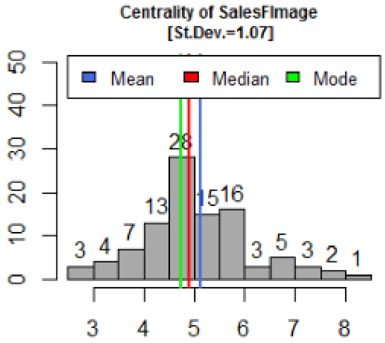
Again twin peak distribution indicating two different groups of products. Widely dispersed distribution with 1.32 standard of deviation. No outliers in the variable.

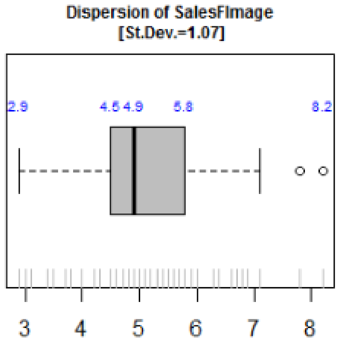




SalesFImage

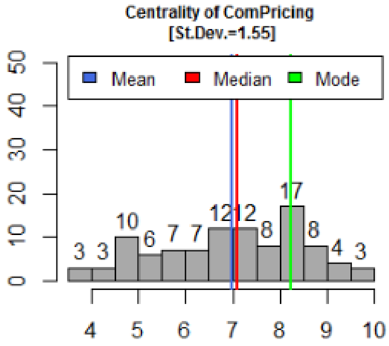
Single peak with almost normal distribution (slightly right skew). Moderately dispersed with 1.07 standard deviation. There are some outliers at the upper end of the distribution.

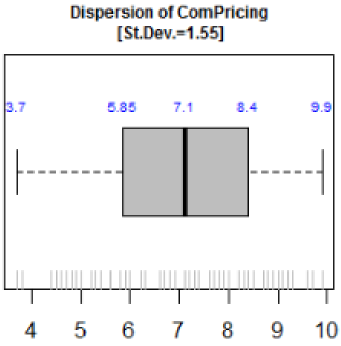




CompPricing

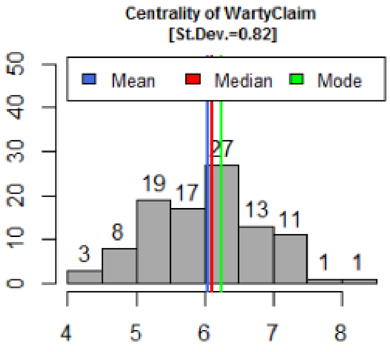
Mutlimodal distribution with three peaks around 5, 7 and 8. Wide dispersion with 1.55 standard deviation. No outliers in the distribution.

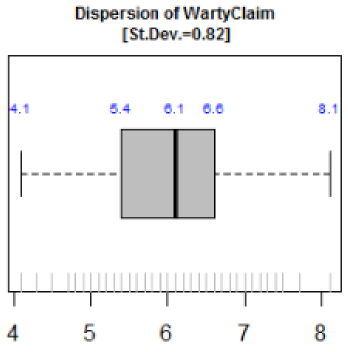




WartyClaim

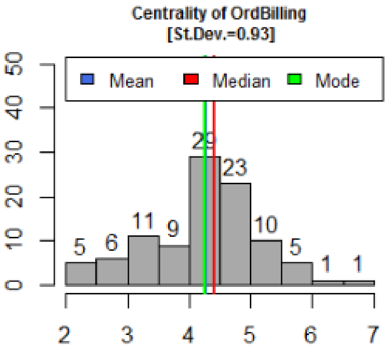
Almost symmetric distribution with couple of peaks showing presence of two types of data. Narrow dispersion with no outliers in the distribution.

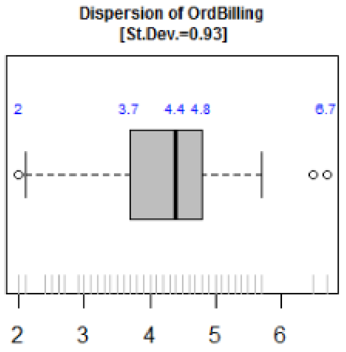




OrdBilling

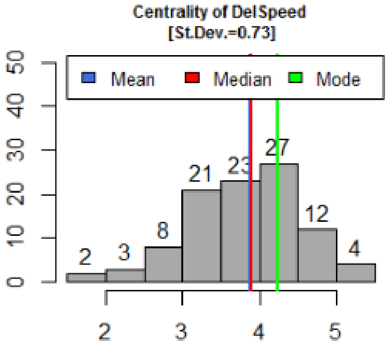
Almost normal distribution with single peak. Narrow dispersion with 0.93 standard of deviation. Outliers at the upper edge of the distribution.

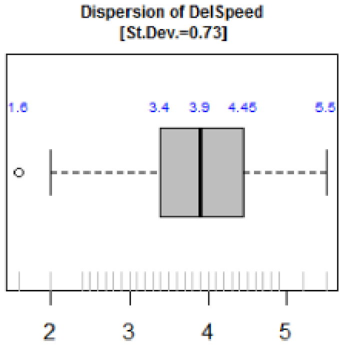




DelSpeed

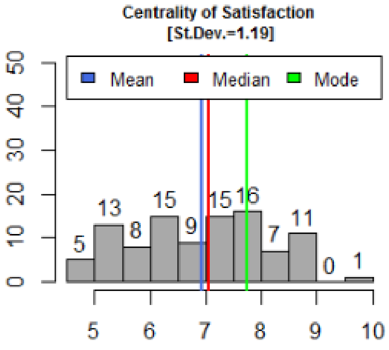
Left Skewed distribution with single peak. Outliers at the lower end of the distribution. Narrow dispersion with 0.73 standard of deviation.

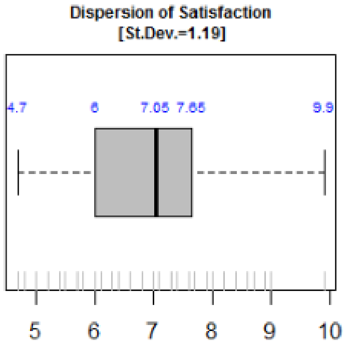




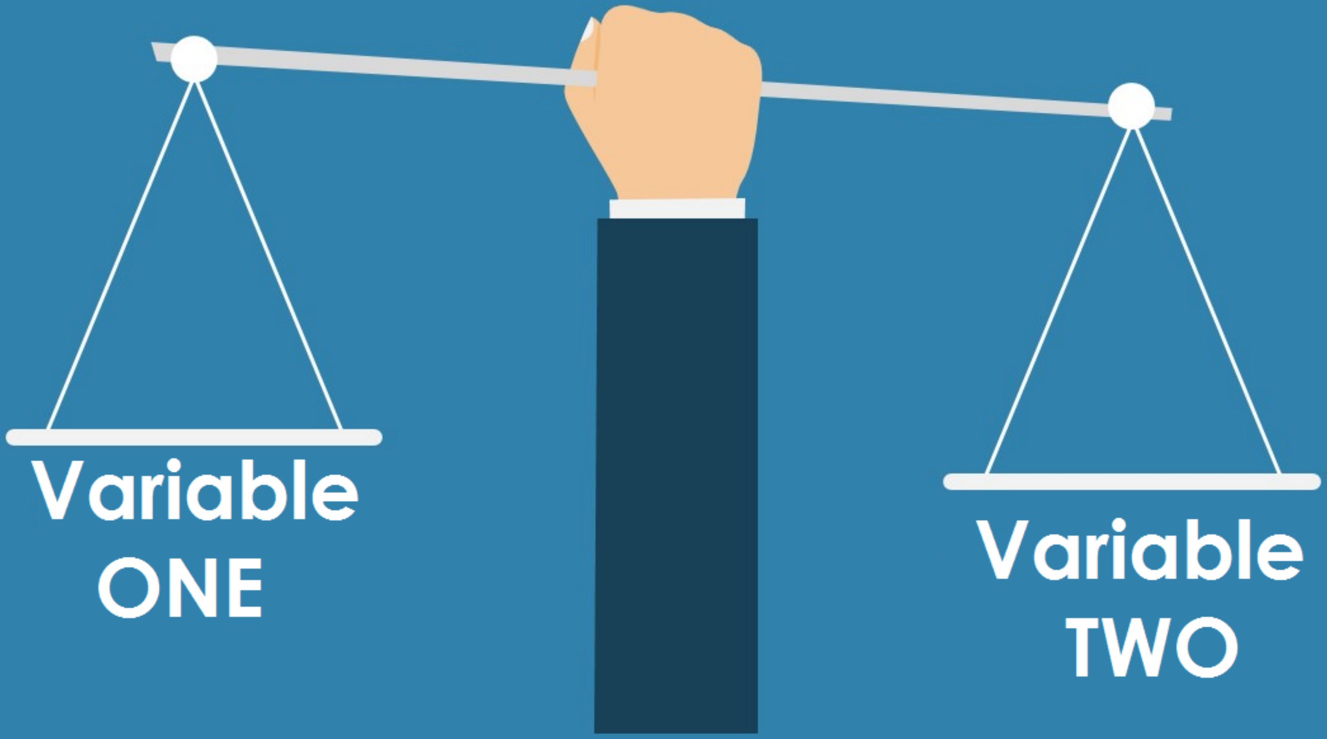
Satisfaction

Almost symmetric distribution with moderate dispersion of 1.19 standard deviation. There are no outliers in this variable.

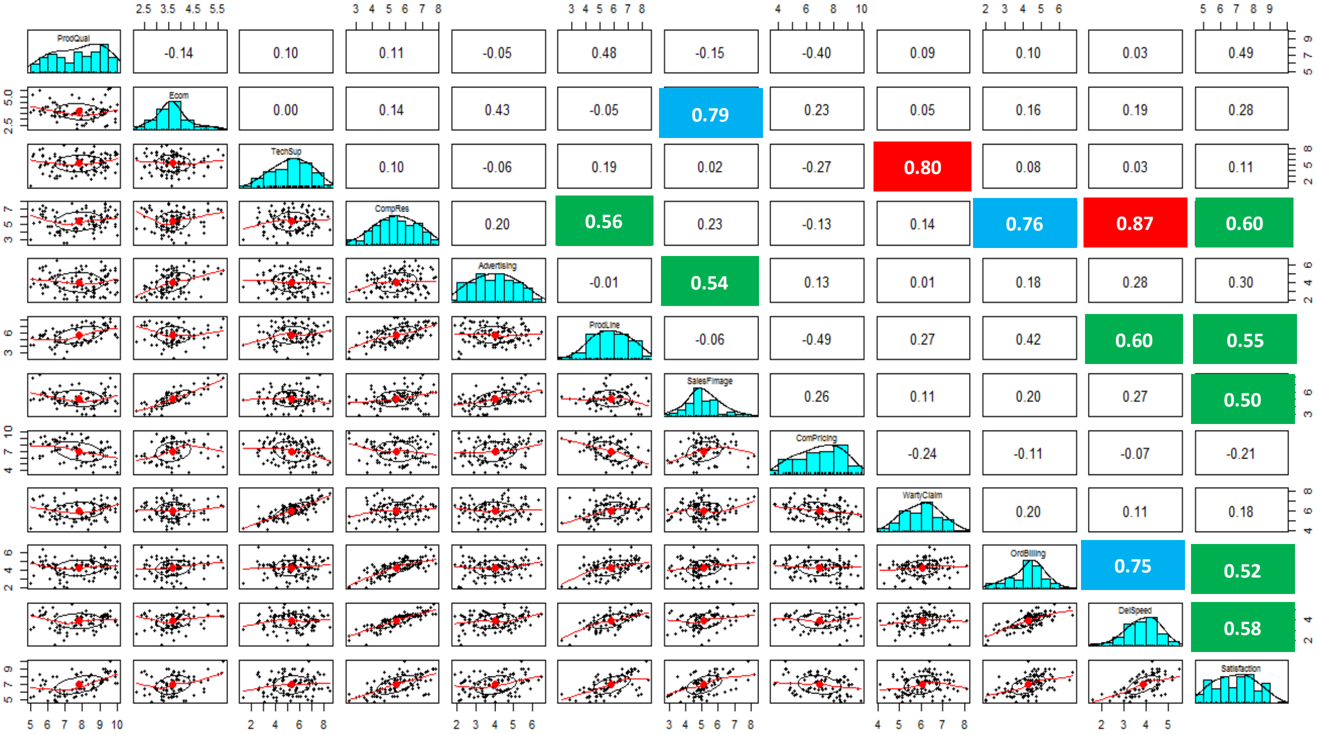




3.7 Bivariate Analysis



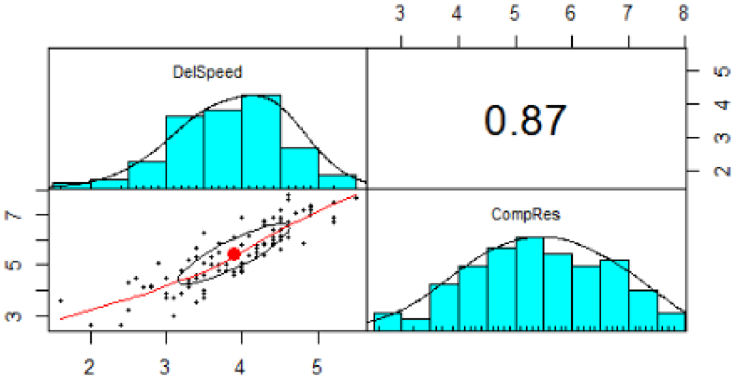
Correlation matrix is showing all the variables of the dataset and their corresponding strengths of correlation:



There are 72 bivariate analysis prospects out of which 13 seems to be promising having more than 50% correlation strength. We will examine them one by one to find out about the insights or clues for further analysis.

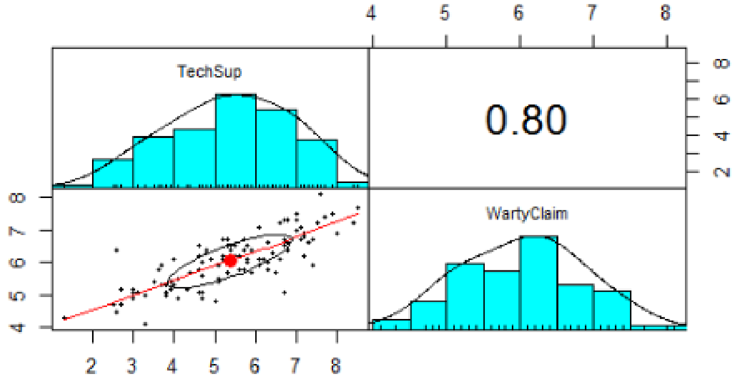
DelSpeed vs CompRes

Very strong correlation of about 87%. Leaner ellipse around the median showing strong cohesion among the variables.



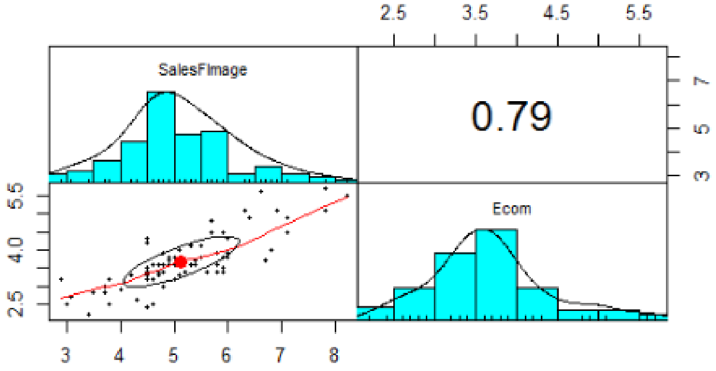
TechSup vs WartyClaim

High correlation of 80%. Intercept is slightly above 4 - WartyClaim.



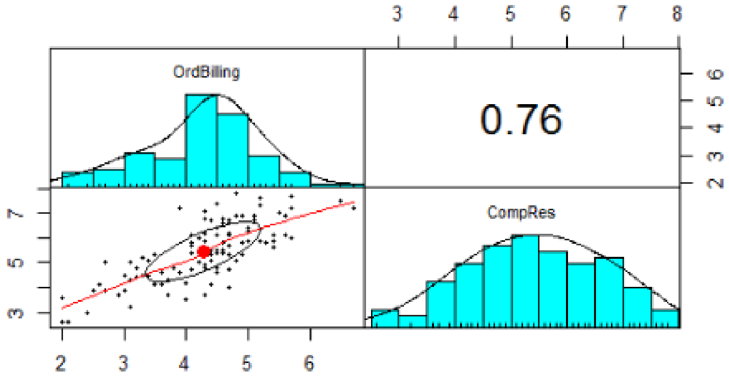
SalesFImage vs Ecom

High correlation with almost 80% strength. Intercept around 2.5 – Ecom.



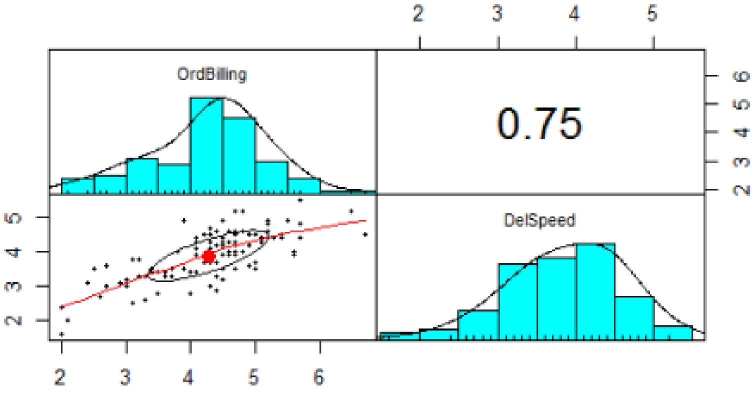
OrdBilling vs CompRes

High correlation with almost 76% strength. Intercept around 3 – ComRes.



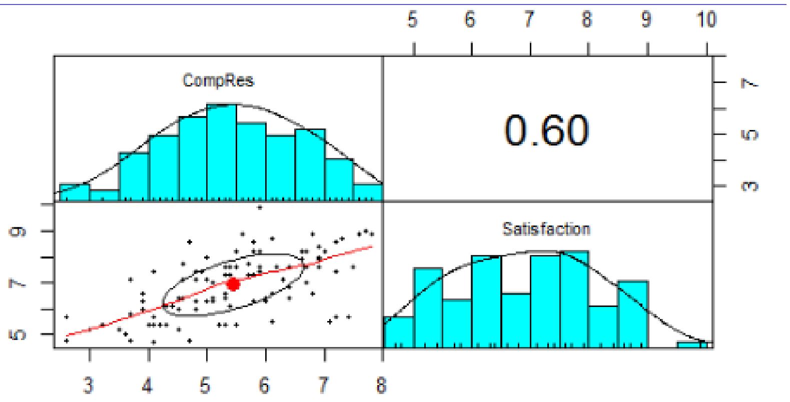
OrdBilling vs DelSpeed

High correlation with almost 75% strength. Intercept around 2.4 – DelSpeed.



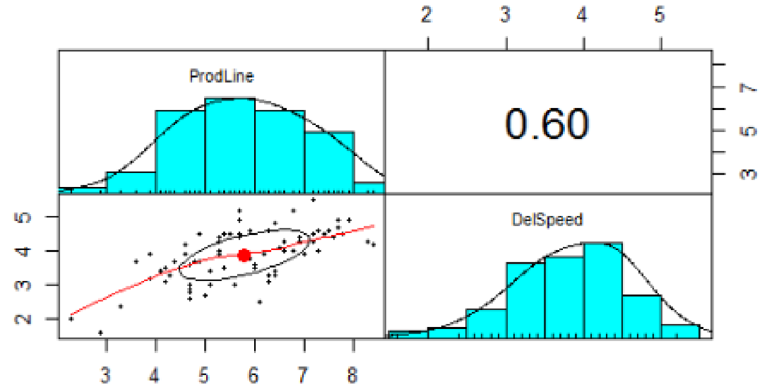
CompRes vs Satisfaction

Correlation of 60%. Intercept is slightly below 5 - Satisfaction.



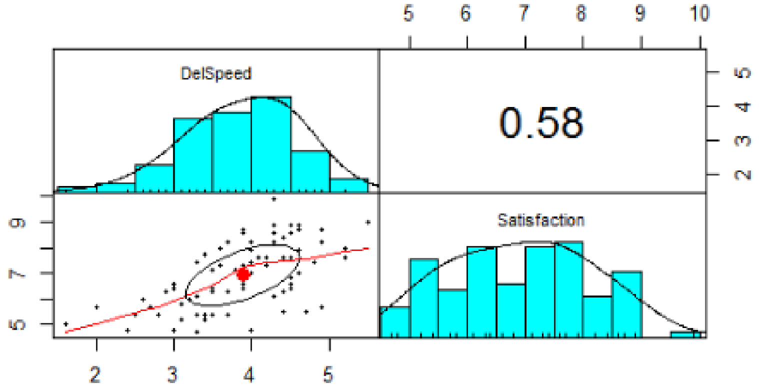
DelSpeed vs ProdLine

Correlation of about 60%. Intercept at around 2 – DelSpeed.



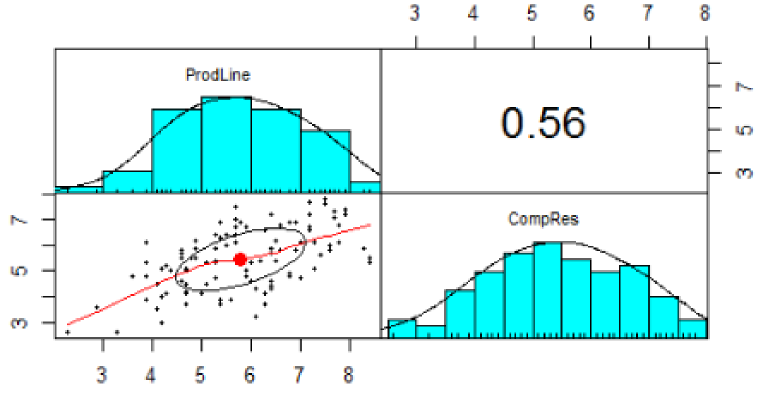
DelSpeed vs Satisfaction

Correlation with almost 58% strength. Intercept around 5 – Satisfaction.



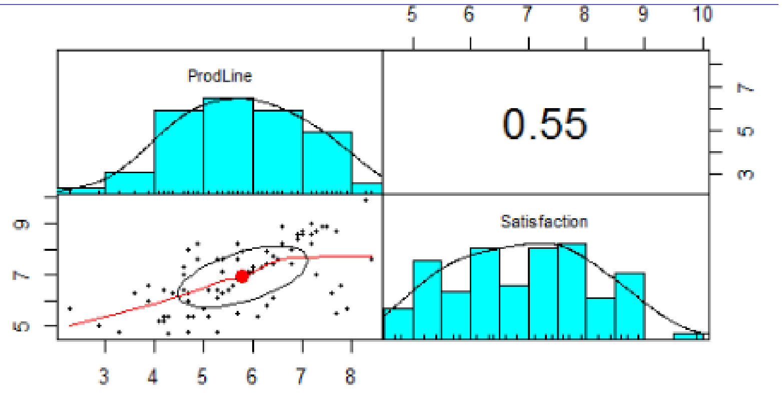
ProdLine vs CompRes

Correlation with almost 56% strength. Intercept around 3 – CompRes.



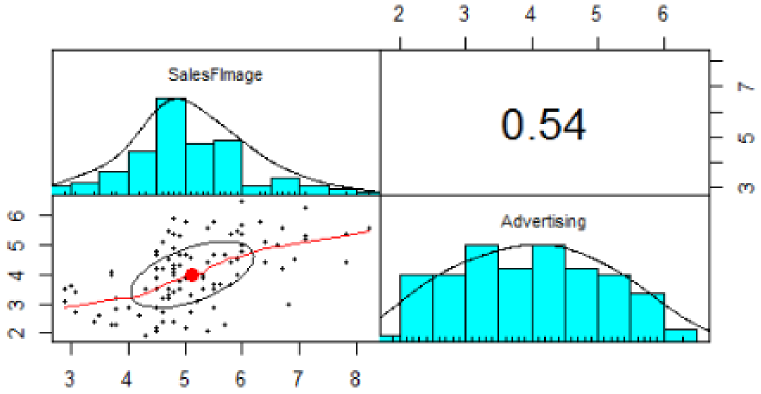
ProdLine vs Satisfaction

Correlation of 55%. Intercept is slightly above 5 - Satisfaction.



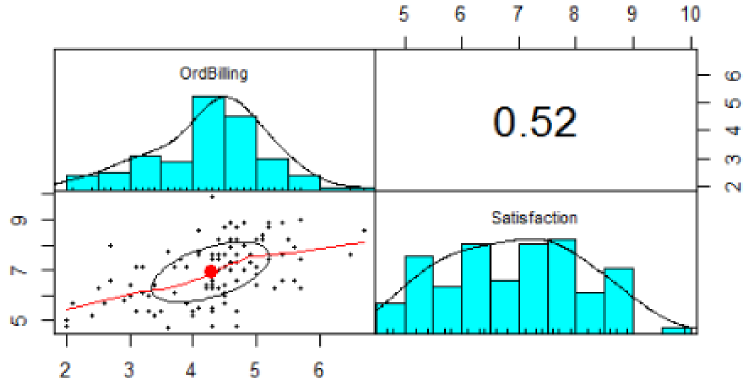
SalesFImage vs Advertising

Correlation of about 54%. Intercept around 3 – Advertising.



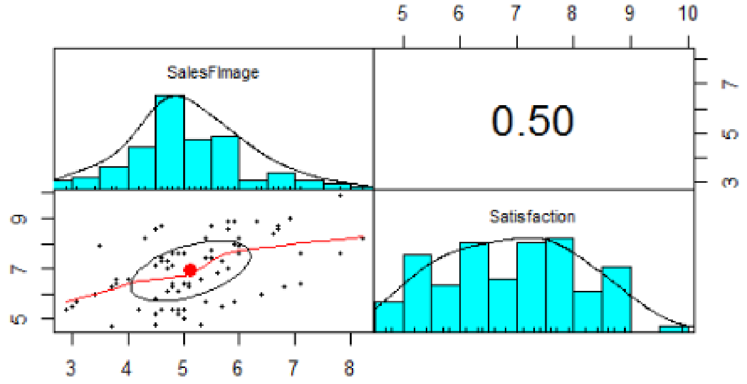
OrdBilling vs Satisfaction

Correlation with almost 52% strength. Intercept above 5 – satisfaction.

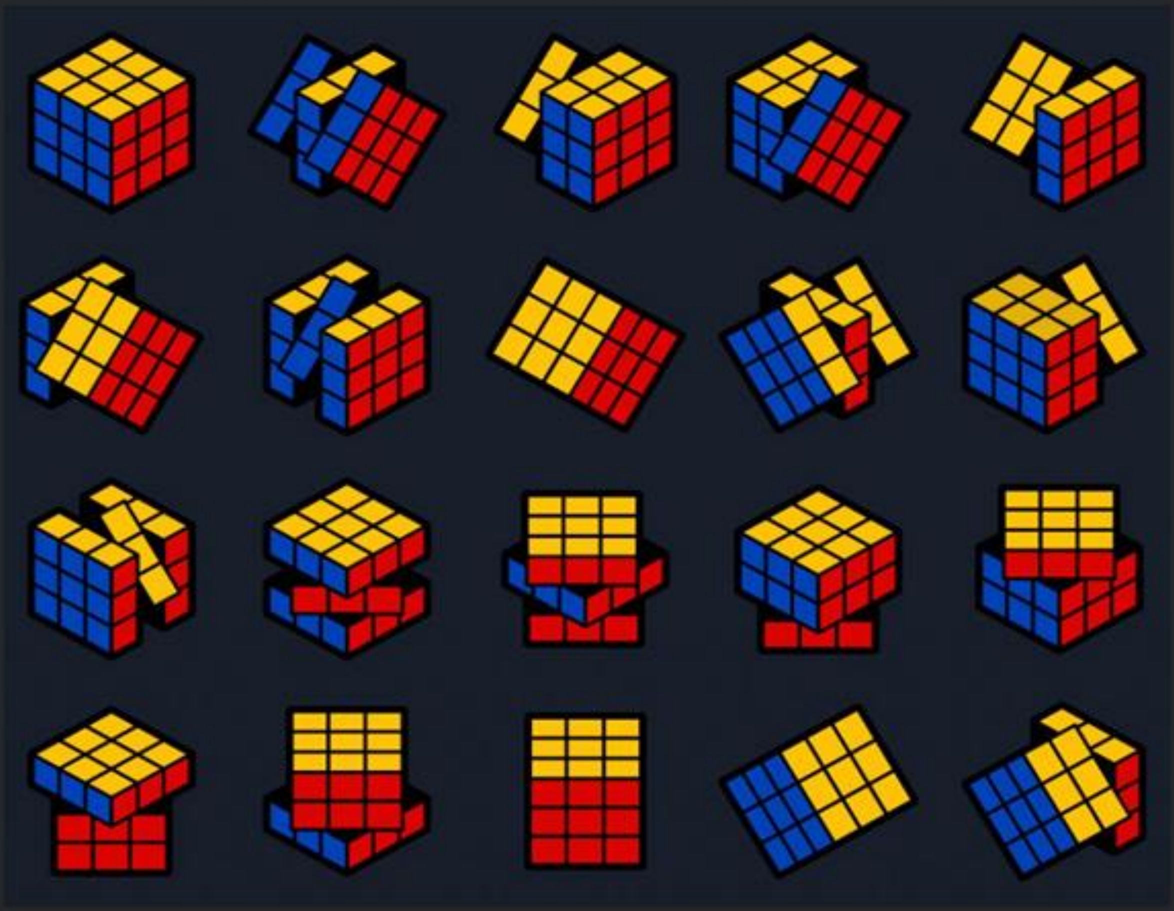


SalesFImage vs Satisfaction

Correlation with almost 50% strength. Intercept around 6 – Satisfaction.



3.8 Simple Linear Regression

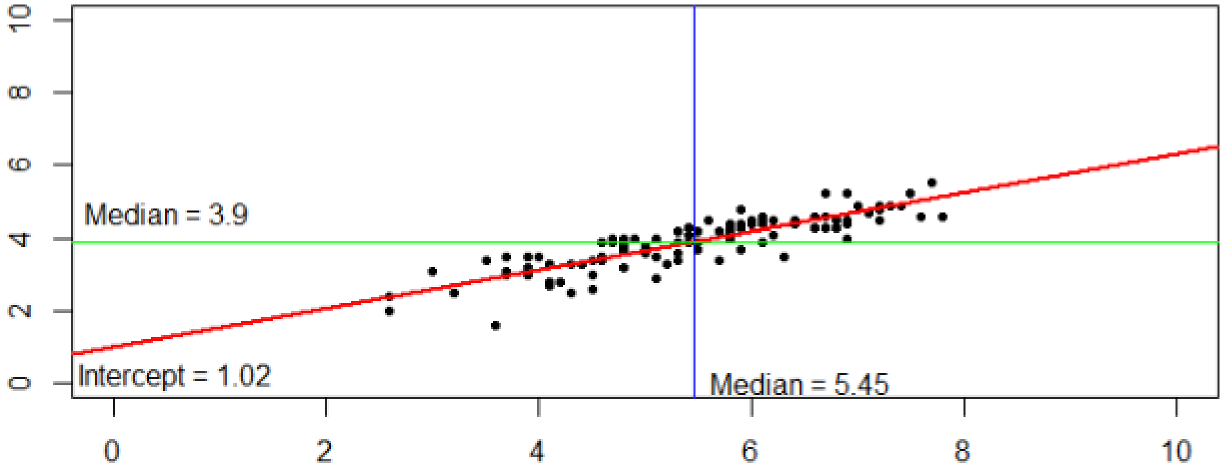


Let’s develop Simple Linear Regression models of all the pairs we identified in previous section with Significant correlation and our minimum threshold is going to be 50% strength. There are total 13 models to be built:

Model: DelSpeed ~ CompRes

Scatter plot showing data with in the range of 2.5 to 8 on CompRes axis. Slope of the linear regression model is 0.52 with Intercept at 1.02.

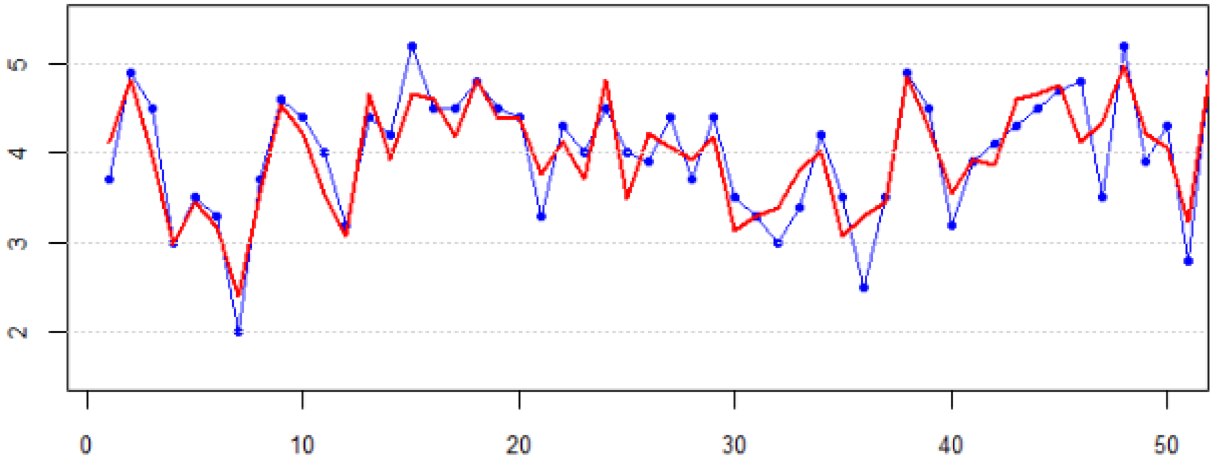
*Model Equation: DelSpeed = **1.02** + **0.52** **.** CompRes*



```
            Estimate      Std. Error     t value      Pr(>|t|)
(Intercept) 1.0247        0.1716         5.97         3.8e-08 ***
CompRes     0.5258        0.0308         17.07        < 2e-16 ***
Multiple R-squared: 0.7484,   Adjusted R-squared: 0.7458
```

Backtracking of the Model:

Backtracking validates the R-Square of 75% as predictions follow the actuals very closely with minimal misses. Model generally fell short on low points whereas predict accurately peaks. Blue is actual and Red is Predicted.



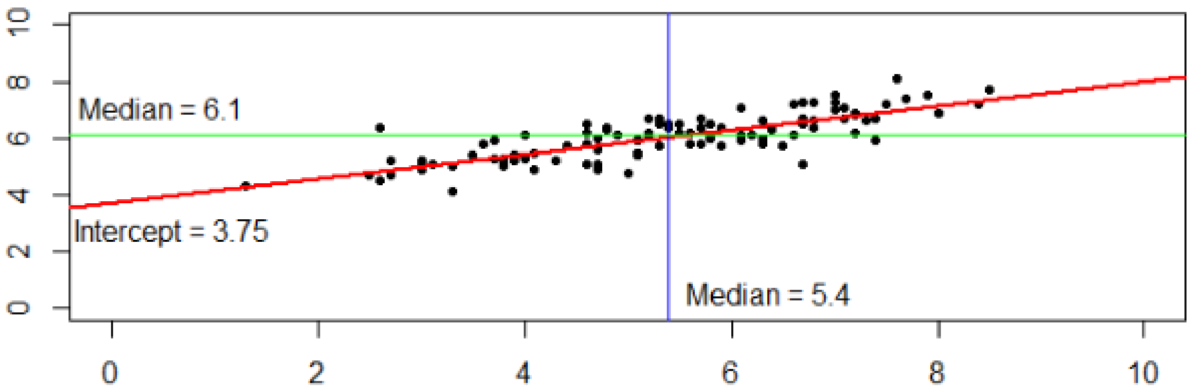
Interpretation of the Model:

- Irrespective of the value of CompRes predictor variable, response variable will have minimum value of 1.02.
- Per unit increase in CompRes variable, cause about 0.52 times increase in the dependent variable.
- Above model captures 74.84% of the variation of Dependant variable “DelSpeed”.
- P-values of Intercept and CompRes variable are highly significant even at 99.9% confidence level.

Model: WartyClaim ~ TechSup

Scatter plot showing data with in the range of 2 to 8.5 on TechSup axis. Slope of the linear regression model is 0.42 with Intercept at 3.75.

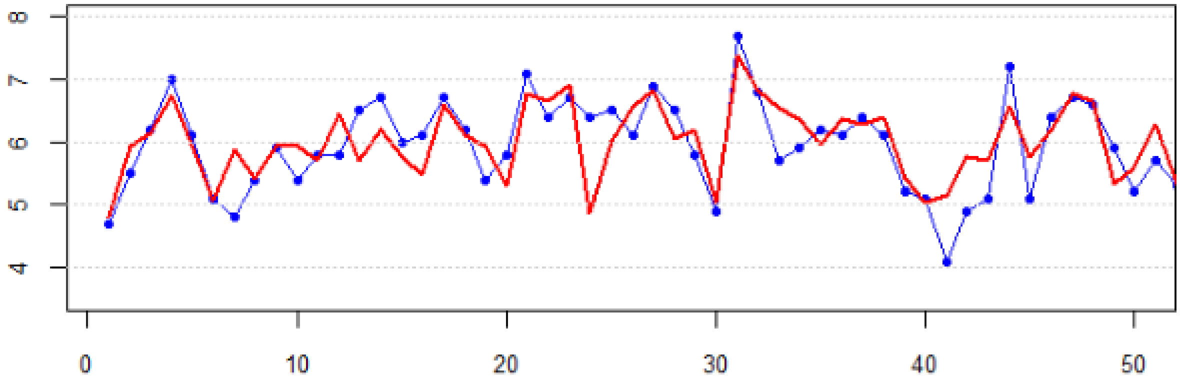
*Model Equation: WartyClaim = **3.75** + **0.42 .** TechSup*



```
             Estimate      Std. Error    t value     Pr(>|t|)
(Intercept)  3.75227       0.18218       20.60       <2e-16 ***
TechSup      0.42698       0.03267       13.07       <2e-16 ***

Multiple R-squared:  0.6355,   Adjusted R-squared:  0.6318
```

Backtracking of the Model:

Backtracking validates the R-Square of 63% as predictions follow the actuals somewhat closely with some misses. Model generally fell short on high points whereas predict lows with some exaggeration. Blue is actual and Red is Predicted.



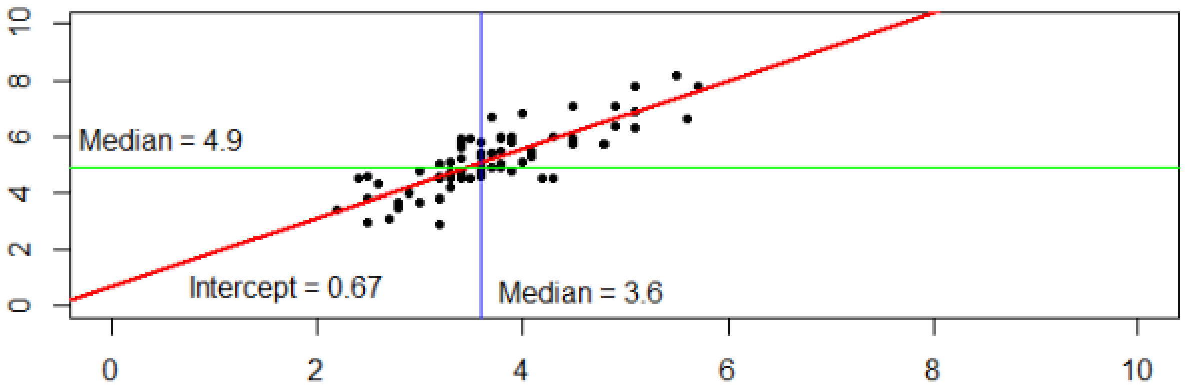
Interpretation of the Model:

- Irrespective of the value of TechSup predictor variable, response variable will have minimum value of 3.75.
- Per unit increase in TechSup variable, cause about 0.42 times increase in the dependent variable “WartyClaim”.
- Above model captures 63.55% of the variation of Dependant variable “WartyClaim” accurately.
- P-values of Intercept and Tech Sup variable are highly significant even at 99.9% confidence level.

Model: SalesFImage ~ Ecom

Scatter plot showing data with in the range of 2 to 8.5 on TechSup axis. Slope of the linear regression model is 0.42 with Intercept at 3.75.

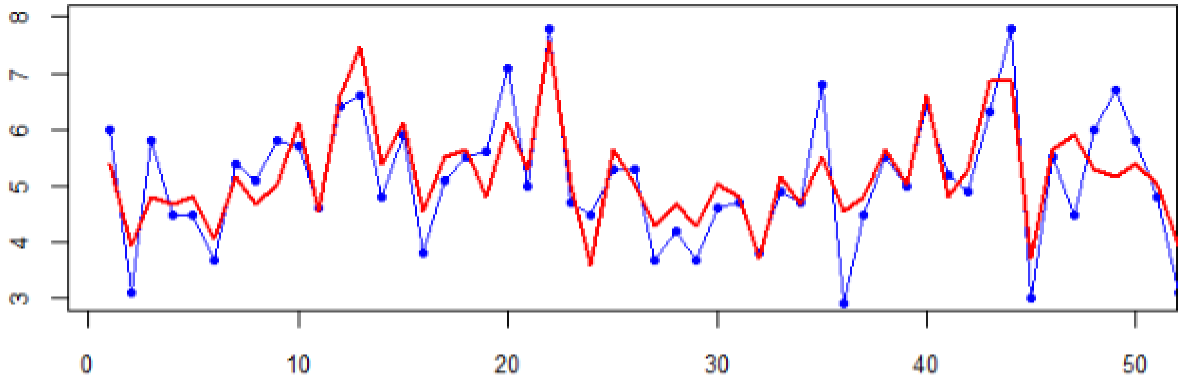
Model Equation: SalesFImage = ***0.67*** *+* ***1.21*** **.** *Ecom*



```
              Estimate      Std. Error     t value       Pr(>|t|)
(Intercept)   0.6738        0.3532         1.908         0.0594 .
Ecom          1.2117        0.0945         12.822        <2e-16 ***

Multiple R-squared:  0.6265,  Adjusted R-squared:  0.6227
```

Backtracking of the Model:

Backtracking validates the R-Square of 62% as predictions follow the actuals somewhat closely with some large misses. Blue is actual and Red is Predicted.



Interpretation of the Model:

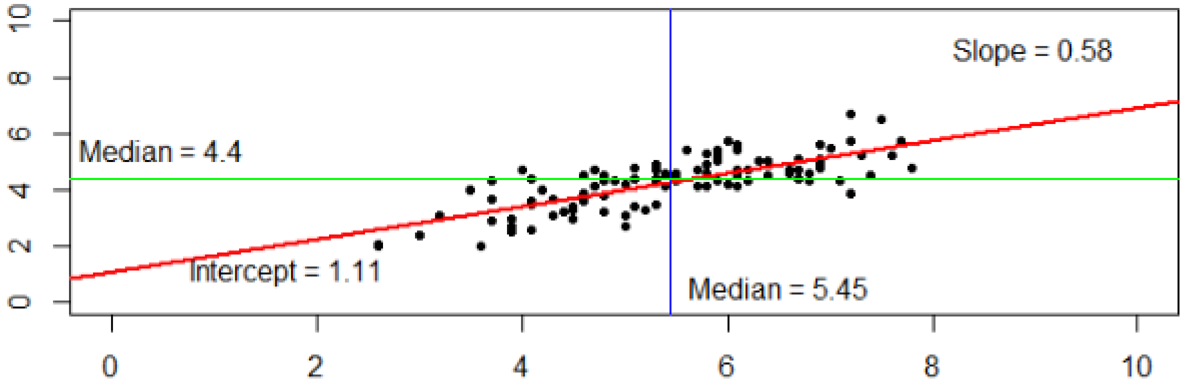
- Irrespective of the value of Ecom predictor variable, response variable will have minimum value of mere 0.67.

- Per unit increase in Ecom variable, cause about 1.21 times increase in the dependent variable “SalesFImage”.
- Above model captures 62.65% of the variations of Dependant variable “SalesFImage” accurately.
- P-values of Intercept is significant at confidence interval of 90% and Ecom variable is highly significant even at 99.9% confidence level.

Model: OrdBilling ~ CompRes

Scatter plot showing data with in the range of 2.5 to 8.0 on CompRes axis. Slope of the linear regression model is 0.58 with Intercept at 1.11.

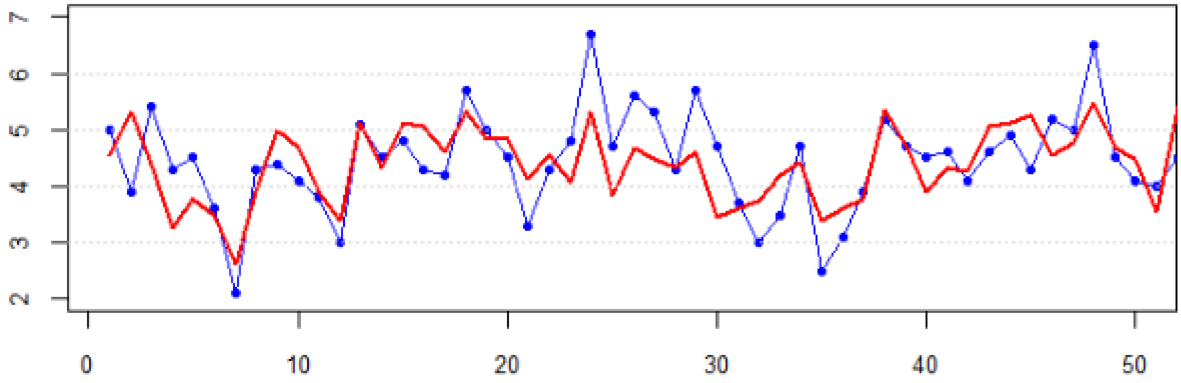
*Model Equation: OrdBilling = **1.11** + **0.58 .** CompRes*



```
            Estimate    Std. Error    t value     Pr(>|t|)
(Intercept) 1.11202     0.28282       3.932       0.000157 ***
CompRes     0.58177     0.05075       11.464      < 2e-16 ***

Multiple R-squared:  0.5729,  Adjusted R-squared:  0.5685
```

Backtracking of the Model:

Backtracking validates the R-Square of 57% as predictions follow the actuals with some large misses. Blue is actual and Red is Predicted.



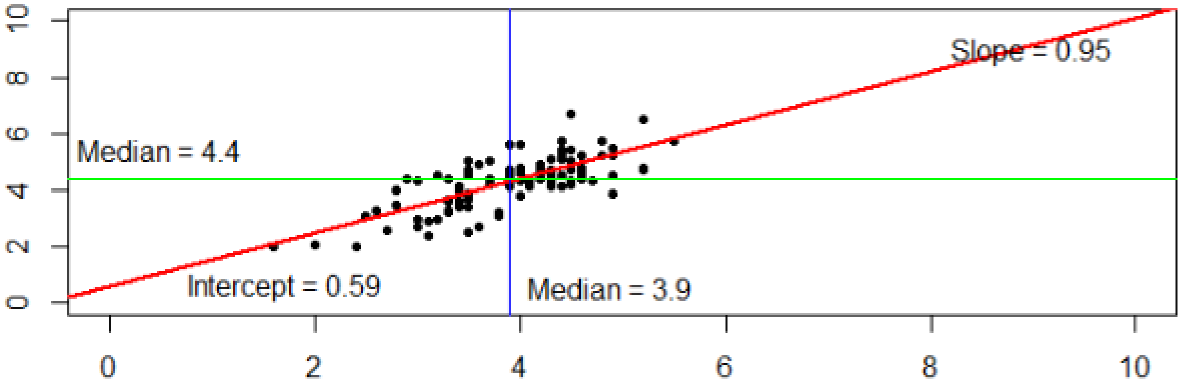
Interpretation of the Model:

- Irrespective of the value of CompRes predictor variable, response variable will have minimum value of 1.11.
- Per unit increase in Ecom variable, cause about 0.58 times increase in the dependent variable "OrdBilling".
- Above model captures 57.29% of the variations of Dependant variable "OrdBilling" accurately.
- P-values of Intercept and CompRes variable is highly significant even at 99.9% confidence level.

Model: OrdBilling ~ DelSpeed

Scatter plot showing data within the range of 1.5 to 6.0 on DelSpeed axis. Slope of the linear regression model is 0.95 with Intercept at 0.59.

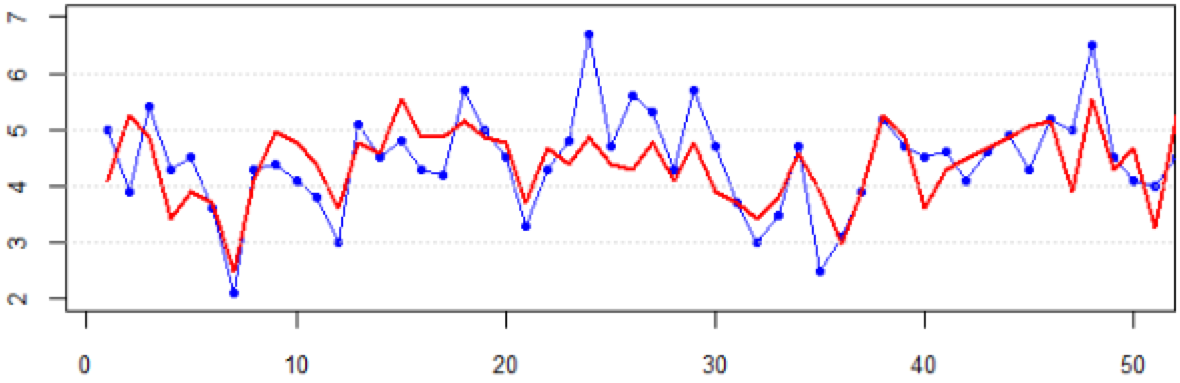
Model Equation: OrdBilling = ***0.59*** *+* ***0.95*** **.** *DelSpeed*



```
            Estimate      Std. Error     t value       Pr(>|t|)
(Intercept) 0.58711       0.33355        1.76          0.0815 .
DelSpeed    0.94979       0.08436        11.26         <2e-16 ***

Multiple R-squared:  0.564,    Adjusted R-squared:  0.5596
```

Backtracking of the Model:

Backtracking validates the R-Square of 56% as predictions follow the actuals with some large misses. Blue is actual and Red is Predicted.



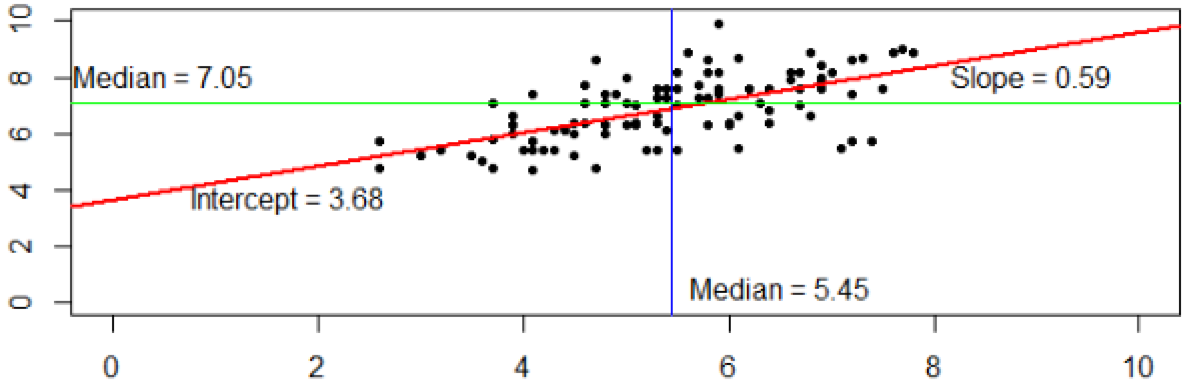
Interpretation of the Model:

- Irrespective of the value of DelSpeed predictor variable, response variable will have minimum value of 0.58.
- Per unit increase in DelSpeed variable, cause about 0.95 times increase in the dependent variable “OrdBilling”.
- Above model captures 56.40% of the variations of Dependant variable “OrdBilling” accurately.
- P-value of Intercept is significant at Confidence interval of 90% while of DelSpeed variable is highly significant even at 99.9% confidence level.

Model: Satisfaction ~ CompRes

Scatter plot showing data within the range of 2.5 to 8.0 on CompRes axis. Slope of the linear regression model is 0.59 with Intercept at 3.68.

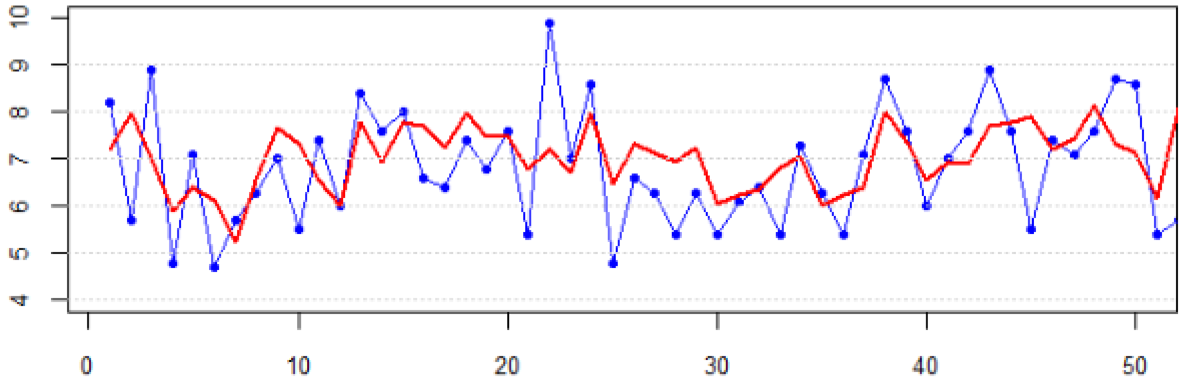
Model Equation: Satisfaction = ***3.68*** *+* ***0.59*** **•** *CompRes*



```
              Estimate        Std. Error      t value        Pr(>|t|)
(Intercept)   3.68005         0.44285         8.310          5.51e-13 ***
CompRes       0.59499         0.07946         7.488          3.09e-11 ***

Multiple R-squared:   0.3639,   Adjusted R-squared:   0.3574
```

Backtracking of the Model:

Backtracking validates the R-Square of 36% as predictions follow the actuals with some large misses. Blue is actual and Red is Predicted.



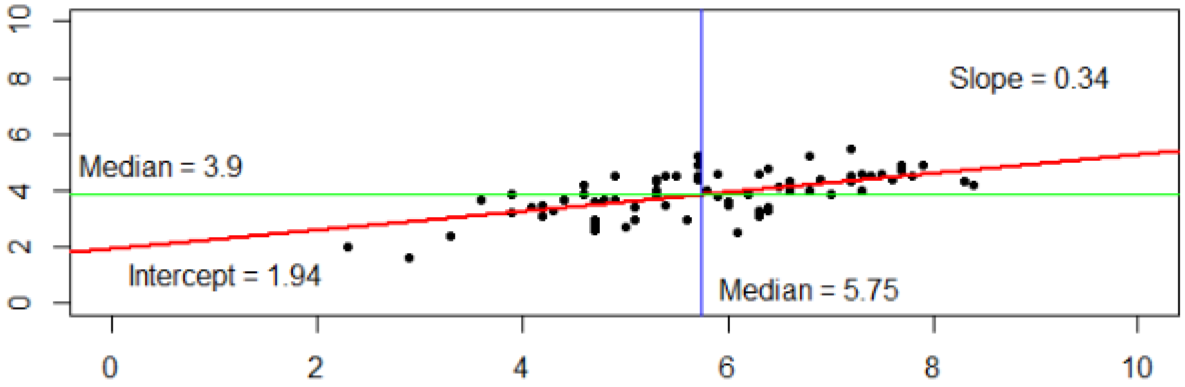
Interpretation of the Model:

- Irrespective of the value of CompRes predictor variable, response variable will have minimum value of 3.68.
- Per unit increase in CompRes variable, cause about 0.59 times increase in the dependent variable “Satisfaction”.
- Above model captures 36.39% of the variations of Dependant variable “Satisfaction” accurately.
- P-values of Intercept and CompRes variable is highly significant even at 99.9% confidence level.

Model: DelSpeed ~ ProdLine

Scatter plot showing data within the range of 2 to 8.5 on ProdLine axis. Slope of the linear regression model is 0.34 with Intercept at 1.93.

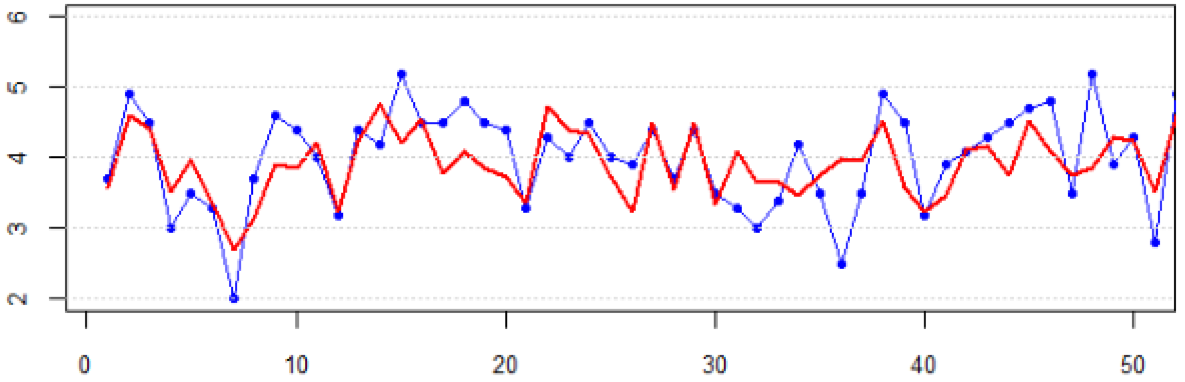
*Model Equation: DelSpeed = **1.93** + **0.33 .** ProdLine*



```
             Estimate    Std. Error   t value     Pr(>|t|)
(Intercept)  1.93514     0.26805      7.219       1.13e-10 ***
ProdLine     0.33606     0.04505      7.460       3.52e-11 ***

Multiple R-squared:  0.3622,  Adjusted R-squared:  0.3557
```

Backtracking of the Model:

Backtracking validates the R-Square of 36% as predictions follow the actuals with some large misses. Blue is actual and Red is Predicted.



Interpretation of the Model:

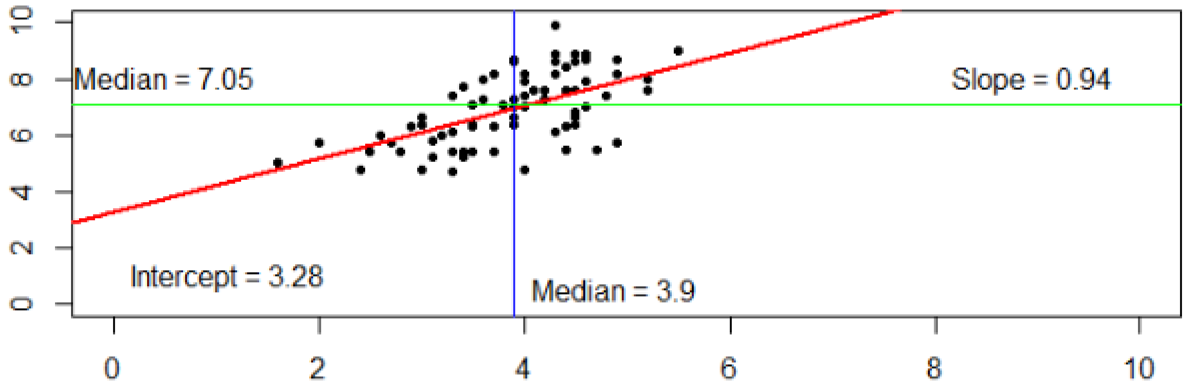
- Irrespective of the value of ProdLine predictor variable, response variable will have minimum value of 1.93.

- Per unit increase in ProdLine variable, cause about 0.33 times increase in the dependent variable "DelSpeed".
- Above model captures 36.22% of the variations of Dependant variable "DelSpeed" accurately.
- P-values of Intercept and ProdLine variable is highly significant even at 99.9% confidence level.

Model: Satisfaction ~ DelSpeed

Scatter plot showing data within the range of 1.5 to 6.0 on DelSpeed axis. Slope of the linear regression model is 0.94 with Intercept at 3.28.

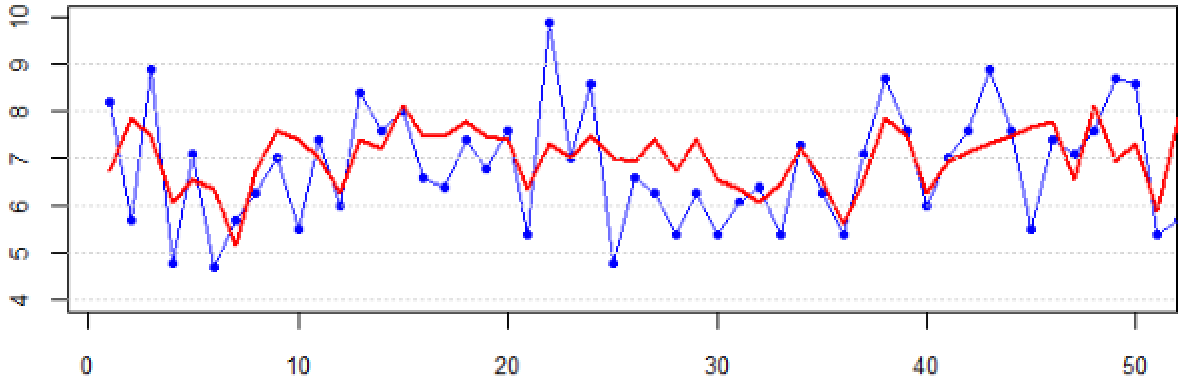
*Model Equation: Satisfaction = **3.28** + **0.93 .** DelSpeed*



```
            Estimate       Std. Error      t value         Pr(>|t|)
(Intercept) 3.2791         0.5294          6.194           1.38e-08 ***
DelSpeed    0.9364         0.1339          6.994           3.30e-10 ***

Multiple R-squared:  0.333,    Adjusted R-squared:  0.3262
```

Backtracking of the Model:

Backtracking validates the R-Square of 33% as predictions follow the actuals with some large misses. Blue is actual and Red is Predicted.



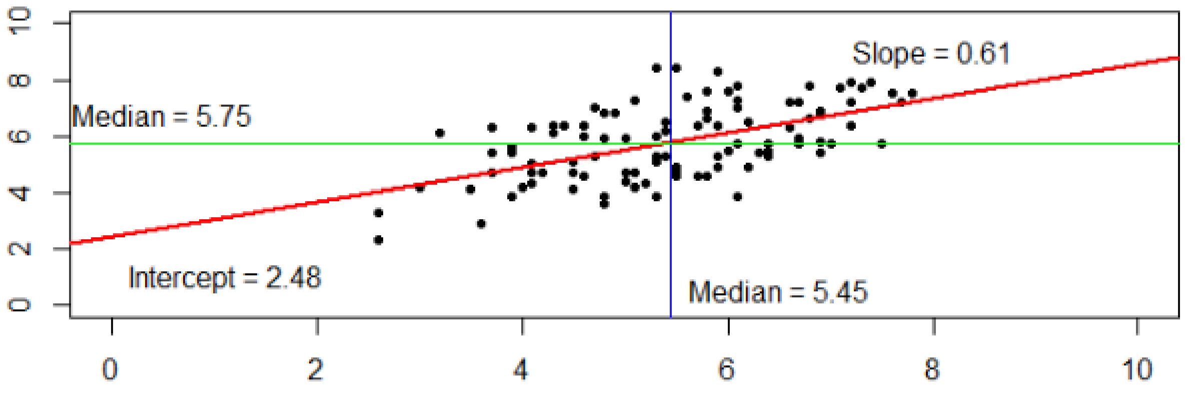
Interpretation of the Model:

- Irrespective of the value of DelSpeed predictor variable, response variable will have minimum value of 3.28.
- Per unit increase in DelSpeed variable, cause about 0.94 times increase in the dependent variable "Satisfaction".
- Above model captures 33.33% of the variations of Dependant variable "Satisfaction" accurately.
- P-values of Intercept and DelSpeed variable is highly significant even at 99.9% confidence level.

Model: ProdLine ~ CompRes

Scatter plot showing data within the range of 2.5 to 8.0 on CompRes axis. Slope of the linear regression model is 0.61 with Intercept at 2.48.

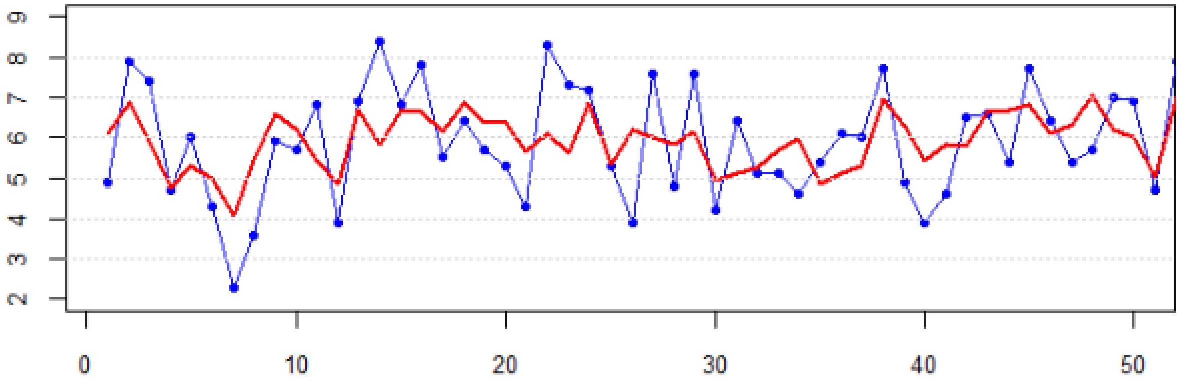
*Model Equation: ProdLine = **2.48** + **0.61** • CompRes*



```
              Estimate      Std. Error     t value      Pr(>|t|)
(Intercept)   2.47954       0.50709        4.890        3.95e-06 ***
CompRes       0.61107       0.09099        6.716        1.23e-09 ***

Multiple R-squared:  0.3152,   Adjusted R-squared:  0.3082
```

Backtracking of the Model:

Backtracking validates the R-Square of 31% as predictions follow the actuals with some large misses. Blue is actual and Red is Predicted.



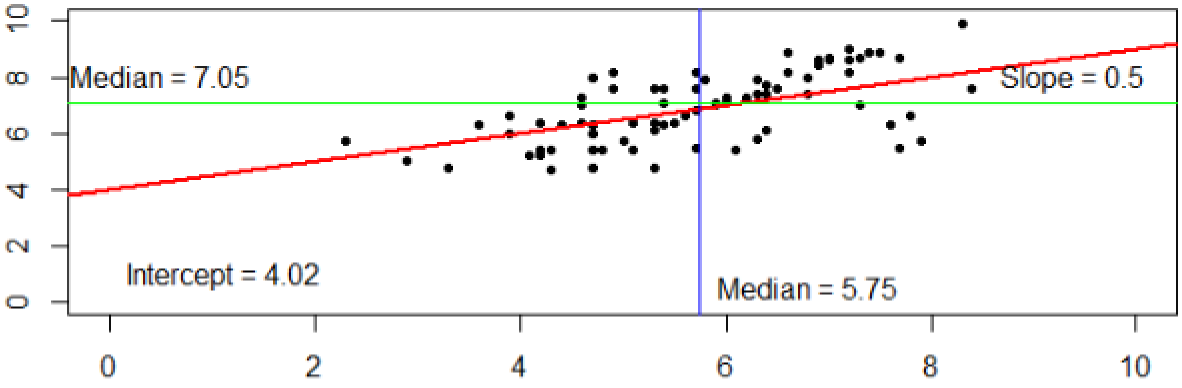
Interpretation of the Model:

- Irrespective of the value of CompRes predictor variable, response variable will have minimum value of 2.48.
- Per unit increase in CompRes variable, cause about 0.61 times increase in the dependent variable “ProdLine”.
- Above model captures 31.52% of the variations of Dependant variable “ProdLine” accurately.
- P-values of Intercept and CompRes variable is highly significant even at 99.9% confidence level.

Model: Satisfaction ~ ProdLine

Scatter plot showing data within the range of 2.0 to 9.0 on ProdLine axis. Slope of the linear regression model is 0.5 with Intercept at 4.02.

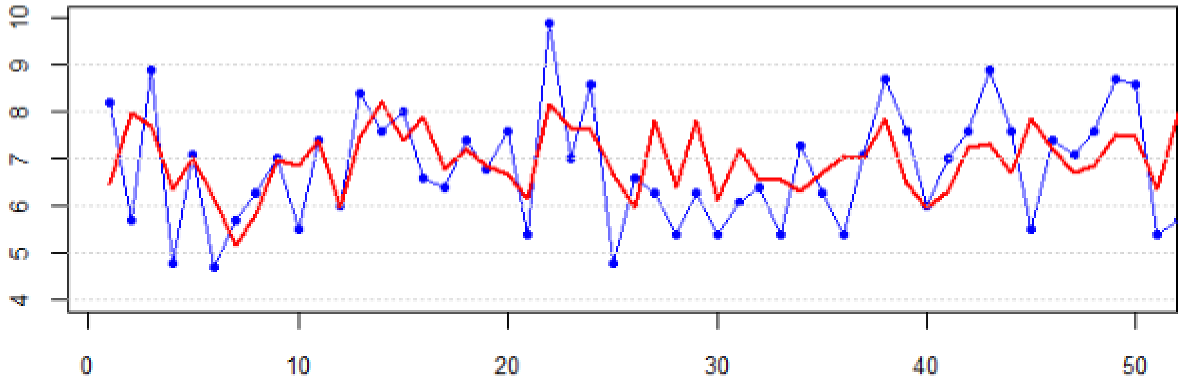
*Model Equation: Satisfaction = **4.02** + **0.5 .** ProdLine*



```
              Estimate      Std. Error    t value      Pr(>|t|)
(Intercept)   4.02203       0.45471       8.845        3.87e-14 ***
ProdLine      0.49887       0.07641       6.529        2.95e-09 ***

Multiple R-squared:  0.3031,  Adjusted R-squared:  0.296
```

Backtracking of the Model:

Backtracking validates the R-Square of 30% as predictions follow the actuals with some large misses. Blue is actual and Red is Predicted.



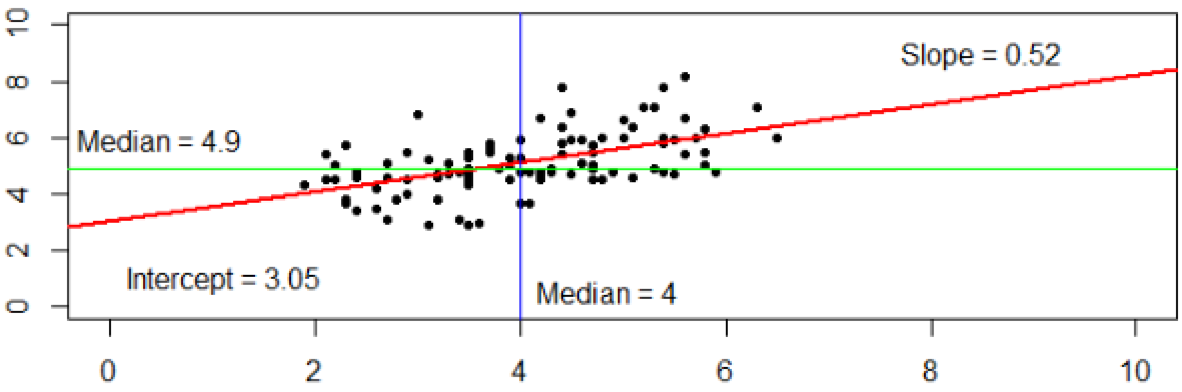
Interpretation of the Model:

- Irrespective of the value of ProdLine predictor variable, response variable will have minimum value of 4.02.
- Per unit increase in ProdLine variable, cause about 0.49 times increase in the dependent variable “Satisfaction”.
- Above model captures 30.31% of the variations of Dependant variable “Satisfaction” accurately.
- P-values of Intercept and ProdLine variable is highly significant even at 99.9% confidence level.

Model: SalesFImage ~ Advertising

Scatter plot showing data within the range of 2.0 to 7.0 on Advertising axis. Slope of the linear regression model is 0.52 with Intercept at 3.05.

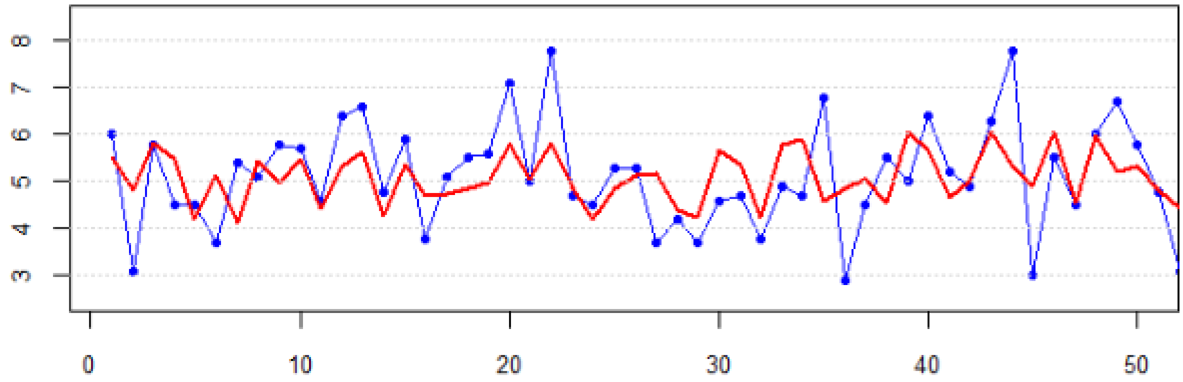
Model Equation: SalesFImage = ***3.05*** *+* ***0.52 .*** *Advertising*



```
              Estimate       Std. Error     t value        Pr(>|t|)
(Intercept)   3.05415        0.33629        9.082          1.19e-14 ***
Advertising   0.51592        0.08076        6.388          5.66e-09 ***

Multiple R-squared:  0.294,    Adjusted R-squared:  0.2868
```

Backtracking of the Model:

Backtracking validates the R-Square of 29% as predictions follow the actuals with some large misses. Blue is actual and Red is Predicted.



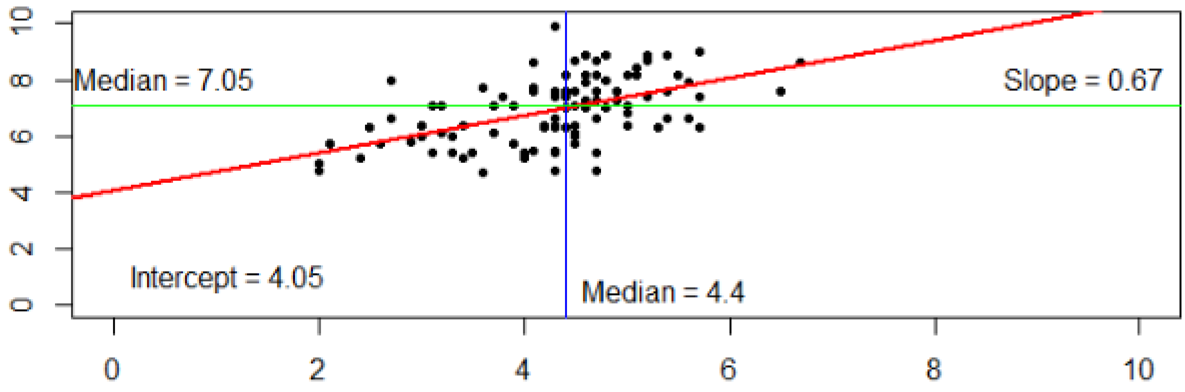
Interpretation of the Model:

- Irrespective of the value of Advertising predictor variable, response variable will have minimum value of 3.05.
- Per unit increase in Advertising variable, cause about 0.51 times increase in the dependent variable “SalesFImage”.
- Above model captures 29.40% of the variations of Dependant variable “SalesFImage” accurately.
- P-values of Intercept and Advertising variable is highly significant even at 99.9% confidence level.

Model: Satisfaction ~ OrdBilling

Scatter plot showing data within the range of 2.0 to 7.0 on Advertising axis. Slope of the linear regression model is 0.67 with Intercept at 4.05.

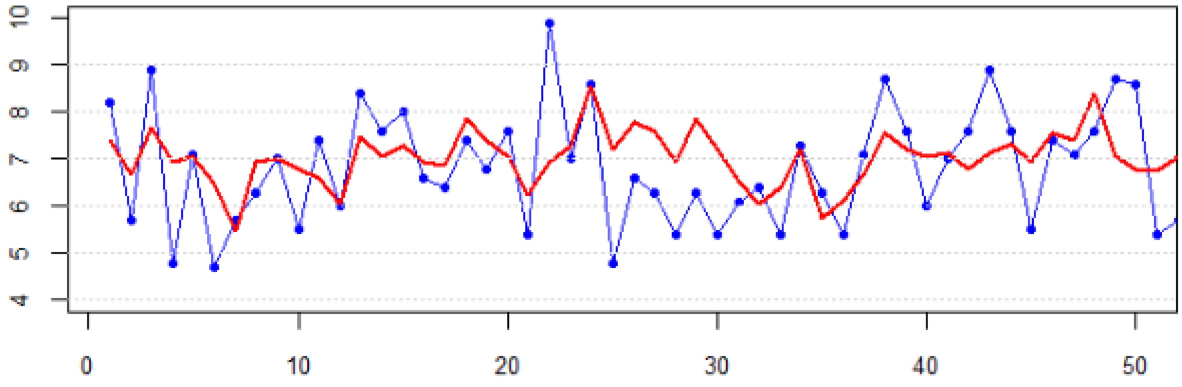
Model Equation: Satisfaction = ***4.05*** *+* ***0.67*** **.** *OrdBilling*



```
              Estimate      Std. Error     t value       Pr(>|t|)
(Intercept)   4.0541        0.4840         8.377         3.96e-13 ***
OrdBilling    0.6695        0.1106         6.054         2.60e-08 ***

Multiple R-squared:  0.2722,  Adjusted R-squared:  0.2648
```

Backtracking of the Model:

Backtracking validates the R-Square of 27% as predictions follow the actuals with some large misses. Blue is actual and Red is Predicted.



Interpretation of the Model:

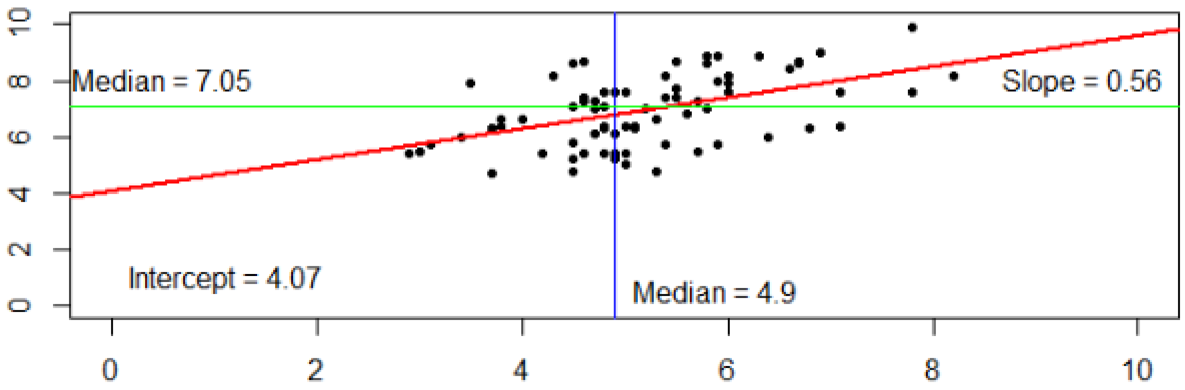
- Irrespective of the value of OrdBilling predictor variable, response variable will have minimum value of 4.05.

- Per unit increase in OrdBilling variable, cause about 0.67 times increase in the dependent variable “Satisfaction”.
- Above model captures 27.22% of the variations of Dependant variable “Satisfaction” accurately.
- P-values of Intercept and OrdBilling variable is highly significant even at 99.9% confidence level.

Model: Satisfaction ~ SalesFImage

Scatter plot showing data within the range of 3.0 to 8.5 on SalesFImage axis. Slope of the linear regression model is 0.67 with Intercept at 4.05.

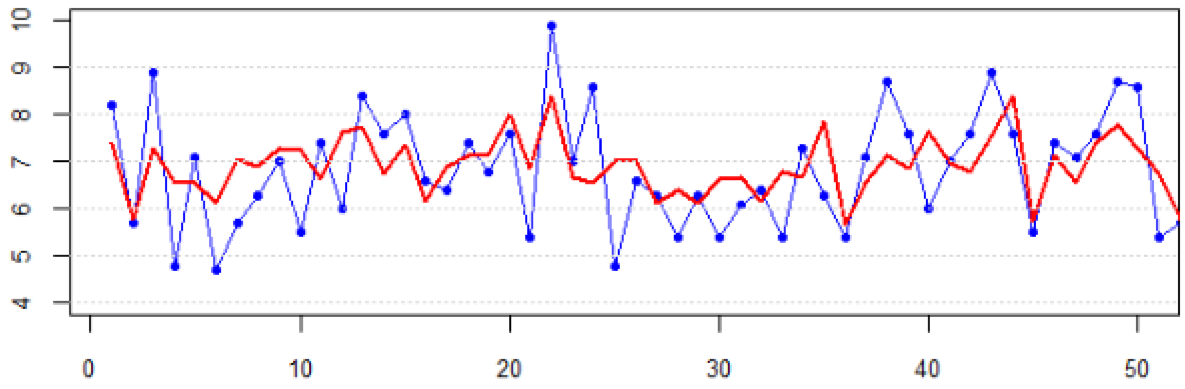
*Model Equation: Satisfaction = **4.07** + **0.56 .** SalesFImage*



```
              Estimate      Std. Error      t value       Pr(>|t|)
(Intercept)   4.06983       0.50874         8.000         2.54e-12 ***
SalesFImage   0.55596       0.09722         5.719         1.16e-07 ***

Multiple R-squared:  0.2502,  Adjusted R-squared:  0.2426
```

Backtracking of the Model:

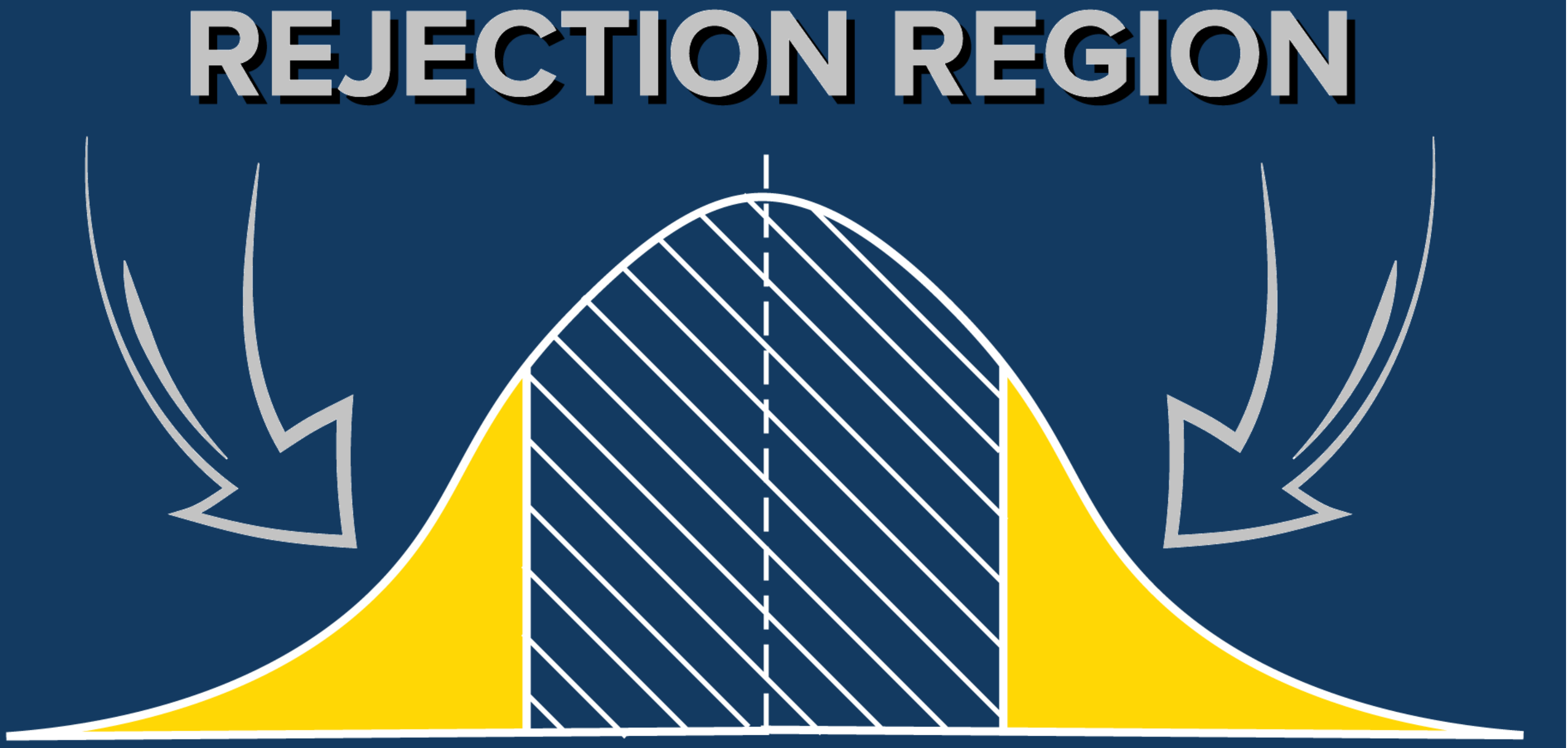
Backtracking validates the R-Square of 25% as predictions follow the actuals with some large misses. Blue is actual and Red is Predicted.



Interpretation of the Model:

- Irrespective of the value of SalesFImage predictor variable, response variable will have minimum value of 4.07.
- Per unit increase in SalesFImage variable, cause about 0.57 times increase in the dependent variable "Satisfaction".
- Above model captures 25.02% of the variations of Dependant variable "Satisfaction" accurately.
- P-values of Intercept and SalesFImage variable is highly significant even at 99.9% confidence level.

4 MultiCollinearity



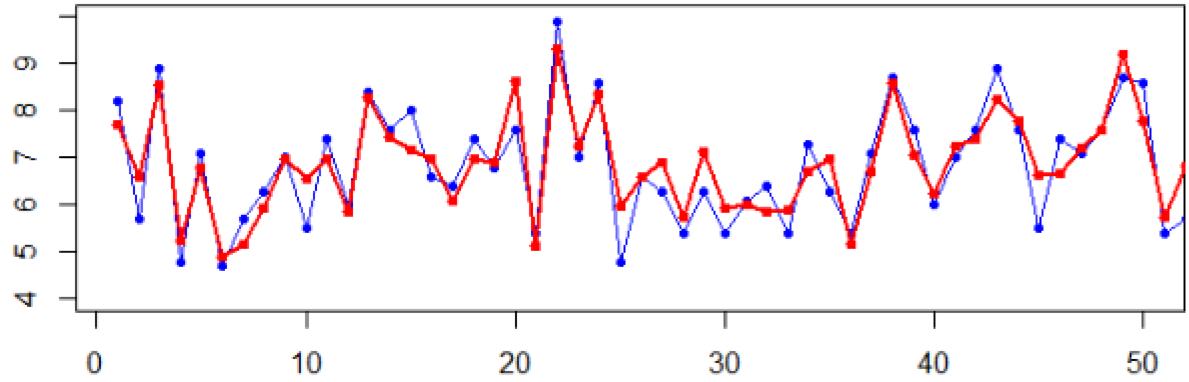
To detect multicollinearity in the dataset we have to propose multiple linear regression model and then conduct Variance Inflation Factor analysis to identify predictors which overlaps and increase ambiguity in explaining the cause of variance. Please note multicollinearity does not adversely effect in prediction rather make ambiguous explanation for change in the response variable.

Our Proposed Model is stated below:

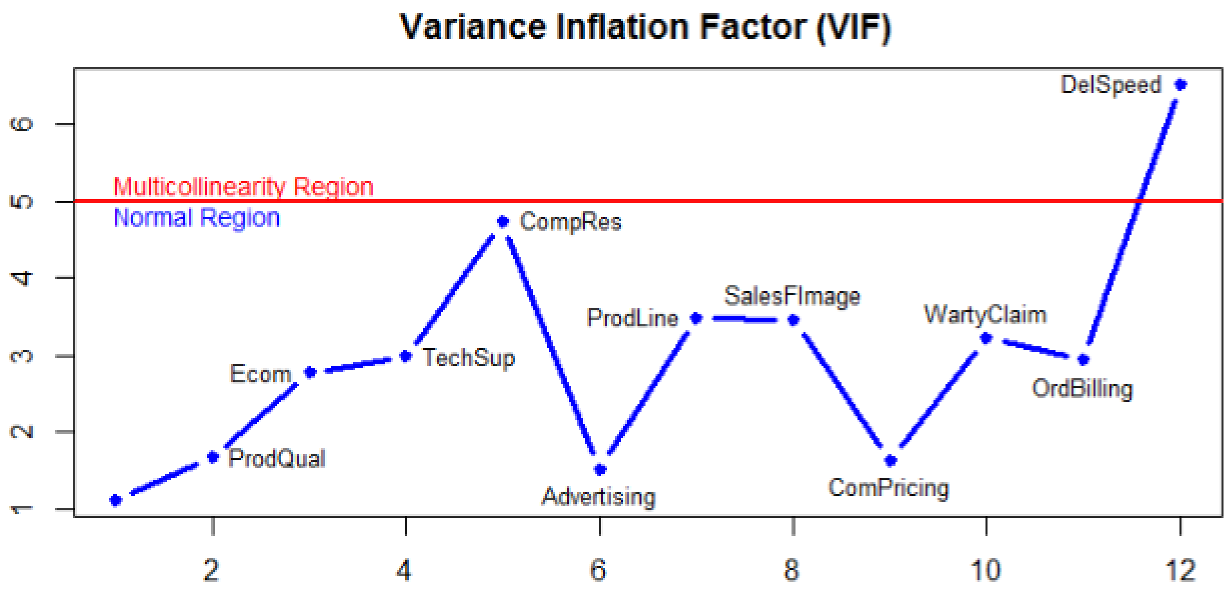
Satisfaction ~ All Other Variables in the dataset

```
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.751368   0.815650  -0.921  0.35950    
ID           0.002145   0.002051   1.046  0.29846    
ProdQual     0.363401   0.052300   6.948 6.41e-10 ***
Ecom        -0.428813   0.134355  -3.192  0.00197 ** 
TechSup      0.036657   0.063777   0.575  0.56693    
CompRes      0.164749   0.101694   1.620  0.10884    
Advertising -0.032713   0.061905  -0.528  0.59854    
ProdLine     0.143919   0.080284   1.793  0.07651 .  
SalesFImage  0.798778   0.097946   8.155 2.39e-12 ***
ComPricing  -0.036226   0.046801  -0.774  0.44100    
WartyClaim  -0.114652   0.123742  -0.927  0.35673    
OrdBilling   0.158780   0.104299   1.522  0.13155    
DelSpeed     0.173458   0.196473   0.883  0.37975    

Multiple R-squared:  0.8046,   Adjusted R-squared:  0.7776
```

With more than 80% coefficient of determination make predictions follow the actuals very closely.



However, if you notice that Satisfaction~OrdBilling was significant in Simple Linear Regression Model while it is insignificant here in this model. Same observations you notice against predictors ProdLine, DelSpeed etc. This ambiguous behavior is due to multicollinearity where predictor variables are strongly correlated with each other. Now lets evaluate them against VIF function.

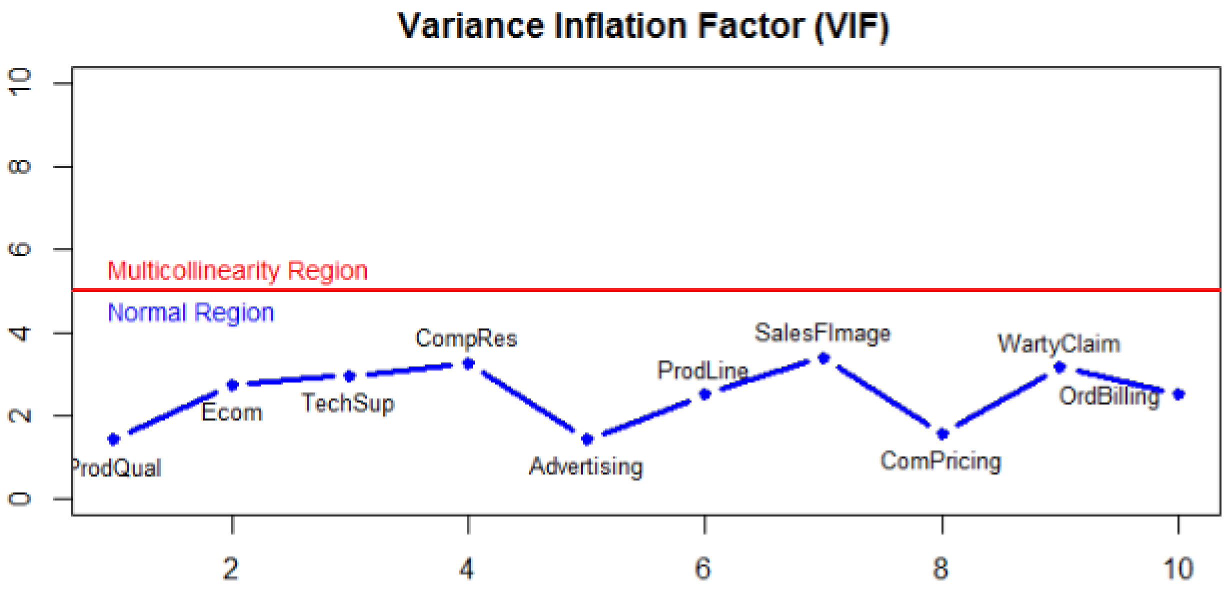


- As per the multiple linear regression model to guage satisfaction among significant correlated independent variables. As per VIF analysis it is validated that there is multicollinearity due to DelSpeed and CompRes in the model. Threshold of 5 has been exceeded by DelSpeed (6.51) in the plot shown here.

- To resolve this we have two options which are Principal Component Analysis and Removing high VIF factor variable from the regression model. We will look into PCA in later section but lets evaluate removing high VIF variable "DelSpeed" from the regression model and see the results, whether there is increase in explanation capacity of the model:

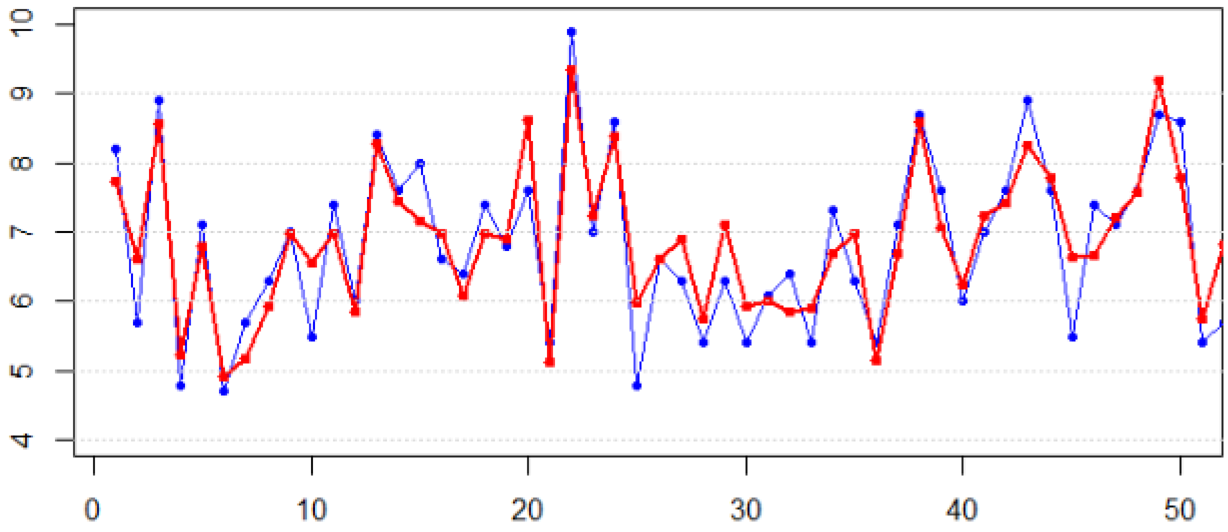
```
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.55463    0.79951  -0.694  0.48967    
ProdQual     0.35588    0.04832   7.364 8.58e-11 ***
Ecom        -0.44513    0.13363  -3.331  0.00126 ** 
TechSup      0.03207    0.06360   0.504  0.61532    
CompRes      0.21464    0.08449   2.540  0.01280 *  
Advertising -0.01554    0.06024  -0.258  0.79705    
ProdLine     0.17617    0.06798   2.592  0.01116 *  
SalesFImage  0.81327    0.09722   8.365 7.70e-13 ***
ComPricing  -0.03127    0.04590  -0.681  0.49754    
WartyClaim  -0.11342    0.12248  -0.926  0.35693    
OrdBilling   0.17865    0.09618   1.857  0.06656 .  

Multiple R-squared:  0.8005,  Adjusted R-squared:  0.7781
```

Notice that Multiple R-Squared is almost the same of 80% even after removing "DelSpeed". However, notice the improvements in Significant values in OrdBilling, SalesFImage, ProdLine, CompRes, Ecom and ProdQual.



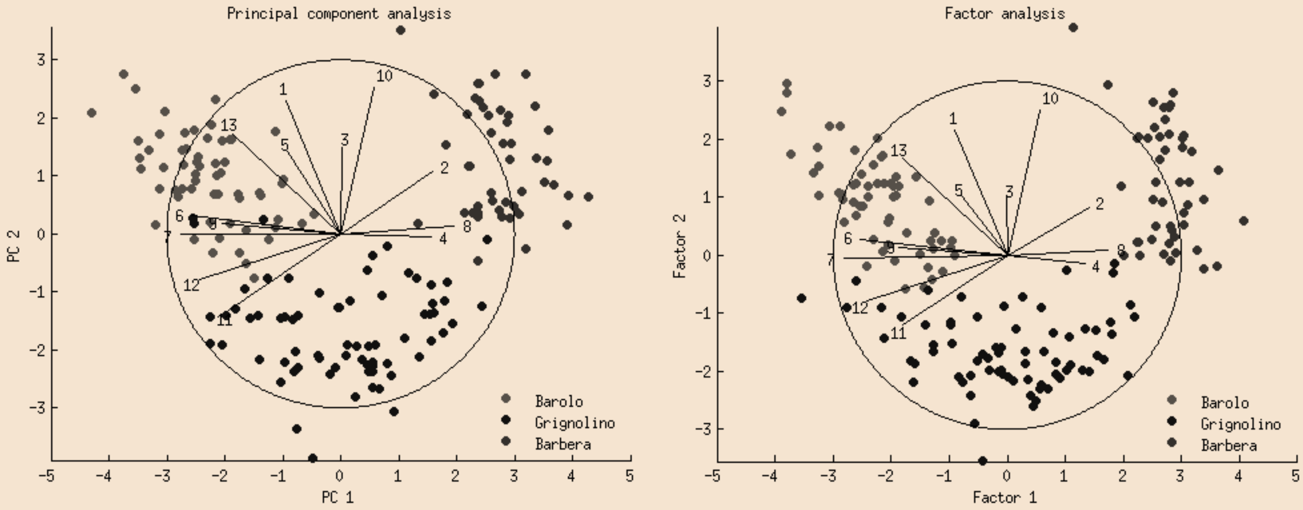
Complete model falls under the normal range having less than 5 VIF which essentially means that multicollinearity is removed from the model and predicitive capability of the model is not dropped significantly, earlier it was 80.46% and after removing DelSpeed from the model it slightly dropped to 80.05%.

Backtracking of the model:

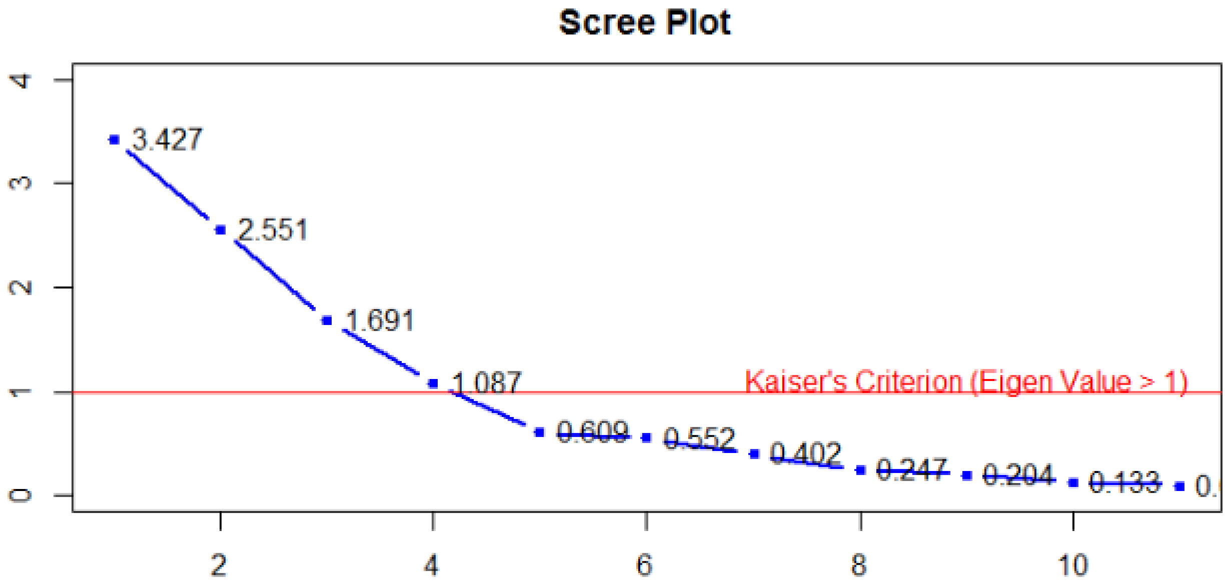


It is evident from the above plot that prediction follows the actuals very closely and thus validates 80% R-Square.

5 Principal Component Analysis



As discussed in previous section to reduce dimensionality in the dataset for removing multicollinearity can be achieved through conducting Principal Component Analysis. First identify how many factors are hidden in the dataset by examining Eigen Values of the variables. Then plotting them in Scree Plot. As per Kaiser's rule eigen values of more than 1.0 worth shortlisting. Therefore, we come to conclusion that there should be FOUR factors/components in our analysis:



Once the number of factors are identified we apply Principal function to calculate the components scores as per the loadings we feed into the dataset. Unrotated values of the factors are mentioned below in the dataset:

	PC1	PC2	PC3	PC4	Communality
ProdQual	0.248	-0.501	-0.081	0.67	0.768
Ecom	0.307	0.713	0.306	0.284	0.777
TechSup	0.292	-0.369	0.794	-0.202	0.893
CompRes	0.871	0.031	-0.274	-0.215	0.881
Advertising	0.34	0.581	0.115	0.331	0.576
ProdLine	0.716	-0.455	-0.151	0.212	0.787
SalesFImage	0.377	0.752	0.314	0.232	0.859
ComPricing	-0.281	0.66	-0.069	-0.348	0.641
WartyClaim	0.394	-0.306	0.778	-0.193	0.892
OrdBilling	0.809	0.042	-0.22	-0.247	0.766
DelSpeed	0.876	0.117	-0.302	-0.206	0.914

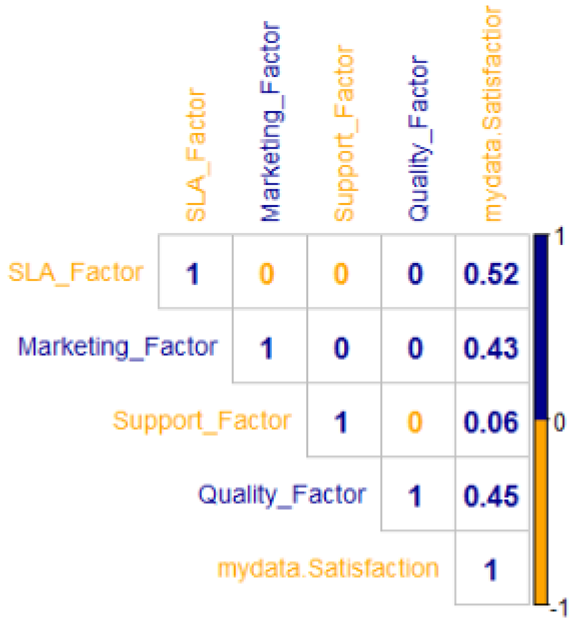
SS Loadings	**3.427**	**2.551**	**1.691**	**1.087**	
Proportion	**0.312**	**0.232**	**0.154**	**0.099**	
Cumulative	**0.312**	**0.543**	**0.697**	**0.796**	
Proportion	**0.391**	**0.291**	**0.193**	**0.124**	
Cumulative	**0.391**	**0.683**	**0.876**	**1**	

Please note that scores are not giving us clear picture that which dimension they are representing therefore, we rotate them through VARIMAX function and get the rotated scores which should represent the data in away that it becomes easier for us to name the factors.

Rotated Values are mentioned below in the table:

	PC1	**PC2**	**PC3**	**PC4**	**Communality**
ProdQual	0.248	-0.501	-0.081	0.670	0.768
Ecom	0.307	0.713	0.306	0.284	0.777
TechSup	0.292	-0.369	0.794	-0.202	0.893
CompRes	0.871	0.031	-0.274	-0.215	0.881
Advertising	0.34	0.581	0.115	0.331	0.576
ProdLine	0.716	-0.455	-0.151	0.212	0.787
SalesFImage	0.377	0.752	0.314	0.232	0.859
ComPricing	-0.281	0.66	-0.069	-0.348	0.641
WartyClaim	0.394	-0.306	0.778	-0.193	0.892
OrdBilling	0.809	0.042	-0.22	-0.247	0.766
DelSpeed	0.876	0.117	-0.302	-0.206	0.914
SS Loadings	**3.427**	**2.551**	**1.691**	**1.087**	
Proportion	**0.312**	**0.232**	**0.154**	**0.099**	
Cumulative	**0.312**	**0.543**	**0.697**	**0.796**	
Proportion	**0.391**	**0.291**	**0.193**	**0.124**	
Cumulative	**0.391**	**0.683**	**0.876**	**1**	

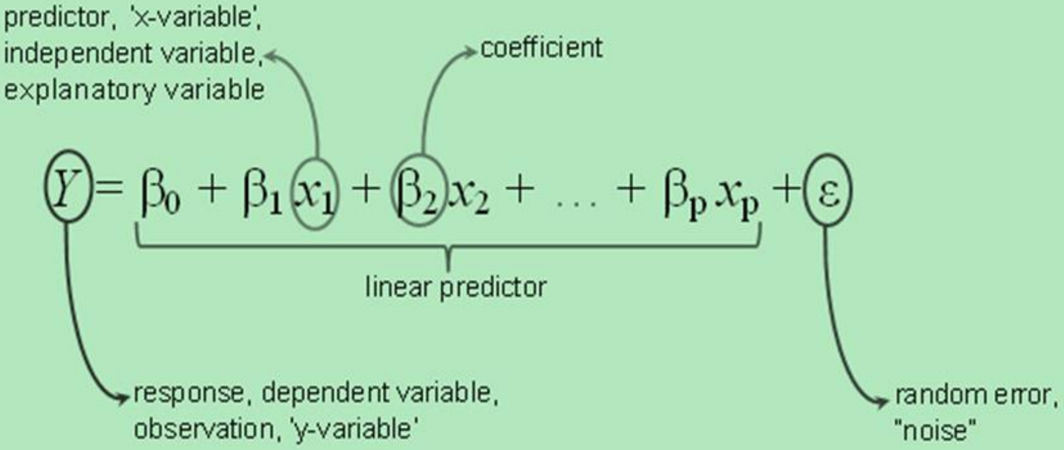
If we look at the scores closely then we find out that PC1 represents Product/Service Level, PC2 represents Marketing, PC3 represents After Sales Support and PC4 represents Quality of the product. Finally lets see what is the correlation matrix of the four components identified along with its correlation with dependent variable "Satisfaction".



	SLA_Factor	Marketing_Factor	Support_Factor	Quality_Factor	mydata.Satisfactior
SLA_Factor	1	0	0	0	0.52
Marketing_Factor		1	0	0	0.43
Support_Factor			1	0	0.06
Quality_Factor				1	0.45
mydata.Satisfaction					1

- Notice that correlation of all the independent variables which are now components or factors is ZERO. All of them have correlation with dependent variable only.
- SLA Factor is the strongest whereas SUPPORT factor is the weakest of all.

6 Multiple Linear Regression


predictor, 'x-variable', independent variable, explanatory variable
coefficient
$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon$
linear predictor
response, dependent variable, observation, 'y-variable'
random error, "noise"

Let’s make regression model based on the components identified in PCA section so that it is free from MultiCollinearity concerns and see its effectiveness:

Satisfaction = *6.918* + (*0.618* **. SLA)** + (*0.510* **. Marketing)** + (*0.540* **. Quality)** + (*0.067* **. Support)**

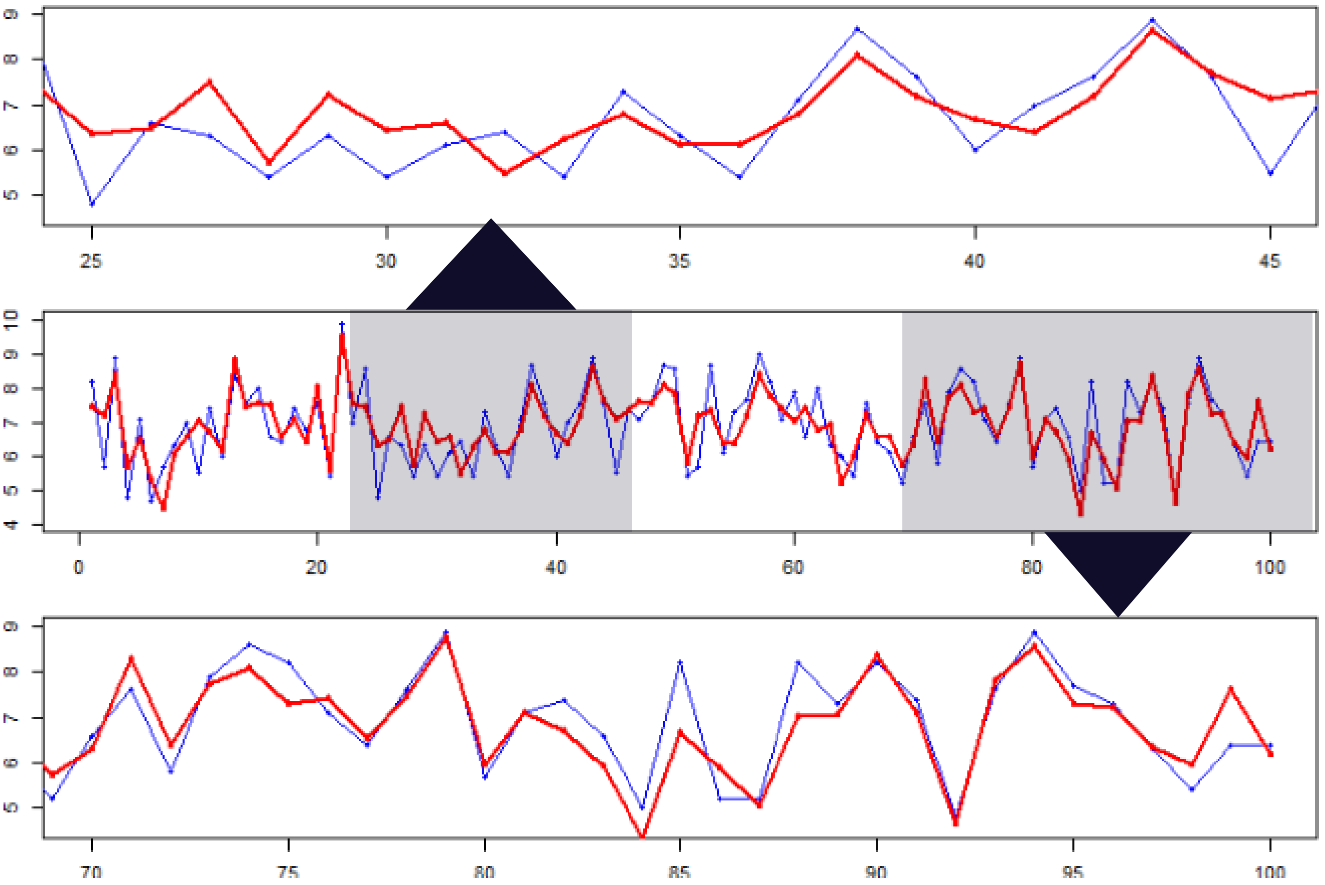
Multiple R Square = **0.6605 ,** Adj. R Square = **0.6462**

SLA, Marketing and Quality has significant p-value, whereas Support has insignificant.

- Intercept is at 6.9 score or 70% showing that customers are significantly at satisfied level even without rating against predictor variables.
- SLA factor has the most significant effect on satisfaction. On each unit increase in SLA while keeping all other variable constant, there is a net increase in Satisfaction of about 0.61 (6%).
- Support factor has the minimum effect on satisfaction. On each unit increase in Support while keeping all other variables constant, triggers increase of 0.067 (½ %).
- Similarly, Marketing & Quality factors has about 5% effect on Response Variable.

Visual representation of the regression can be seen in the plots on the right of the slide. With 66% R-Square, model depicts reasonable performance.

Upper plot has the window of 20 predictions and Lower plot has the window of 30 predictions. We can closely monitor the performance in these two graphs point by point.

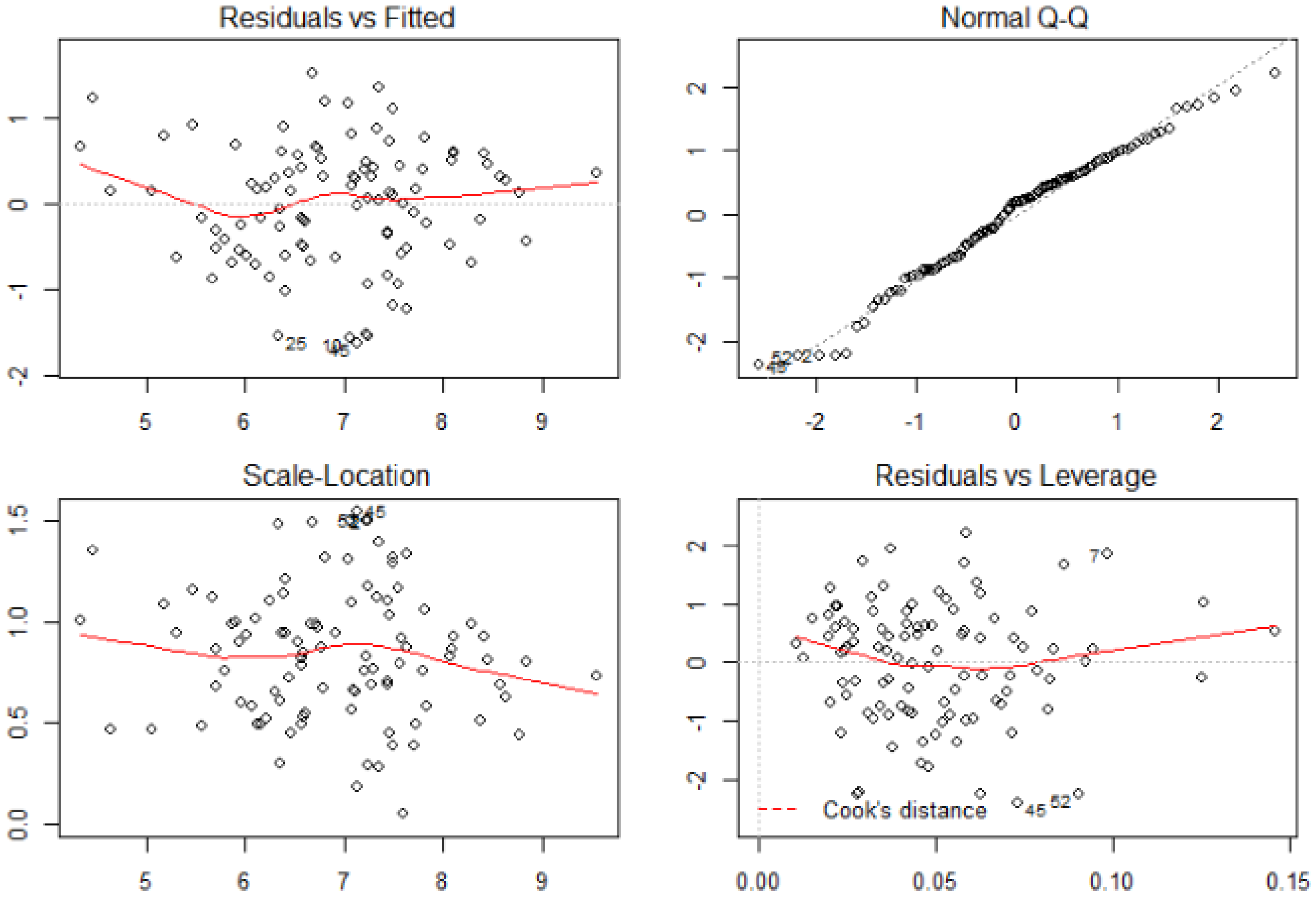


- **Residual Vs Fitted:** We cannot detect any non-linear relations in the plot.

- **Normal Q-Q:** Residuals are almost normally distributed with minor exceptions at the end points which can be ignored.

- **Scale-Location:** The residuals appear with random spread in the plot.

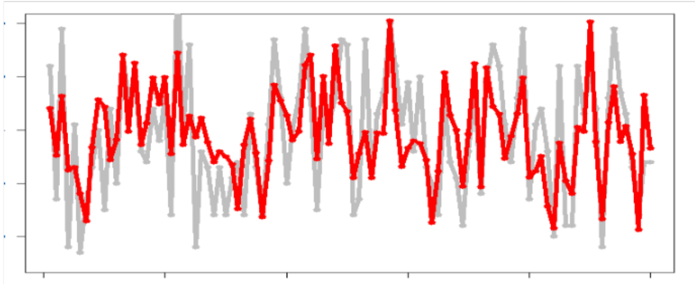
- **Residuals Vs Leverage:** We barely identify cook’s distance line in the plot which indicates there are no influential cases which could impact regression performance.


Residuals vs Fitted
Normal Q-Q
Scale-Location
Residuals vs Leverage
Cook's distance
25
10
45
52
7
5
6
7
8
9
-2
-1
0
1
2
0.0
0.5
1.0
1.5
0.00
0.05
0.10
0.15

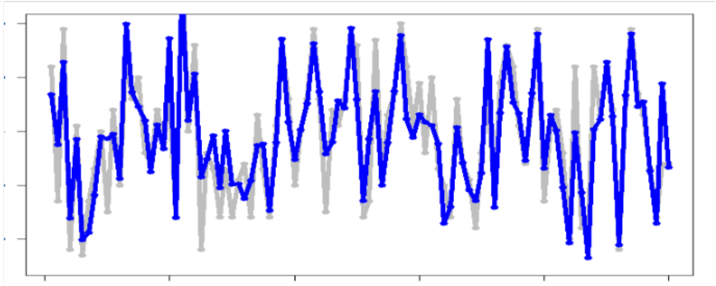
7

In this analysis it has been noticed that after removing DelSpeed variable from the Multiple Linear Regression, (NORMAL) than model worked better with (Multiple R Square = 80%) than the model of factors PCA Based (Multiple R Square = 66%) . Consequently variations in the predictions are not captured well in PCA Based regression model.

Backtracking of both models are shown on here “Gray” is actual whereas “Blue & Red” are predicted



PCA BASED



NORMAL

Appendix A – Source Code

```
### Loading Libraries ----
require(pacman) # package management p_load function
p_load(readr,install=TRUE,update=getOption("par_update")) # reading csv files
p_load(DataExplorer,install=TRUE,update=getOption("par_update")) # for exploratory data analysis
p_load(ggplot2,install=TRUE,update=getOption("par_update")) # Visualization
p_load(psych,install=TRUE,update=getOption("par_update")) #panel.pairs for bivariate analysis
p_load(corrplot,install=TRUE,update=getOption("par_update")) # correlations of variables
p_load(nFactors,install=TRUE,update=getOption("par_update")) #for PCA and Factor analysis
p_load(car,install=TRUE,update=getOption("par_update")) #VIF function for multicollinearity

### Working Directory Setup ----
setwd("C:/DSBA_Course/Proper Learning/Module 3 [Advanced Statistics]/M3 W4 [Project 3]/")

### Loading Dataset ----
mydata <- read.csv("Factor-Hair-Revised.csv",header = TRUE)

### Explore Dataset ----
dim(mydata) #100 rows and 13 columns
plot_intro(mydata) # all are continuous variables
names(mydata) # standard format names of variables
plot_str(mydata, max_level = 1) # all numeric variables
head(mydata)
tail(mydata)
plot_missing(mydata) #no missing values
summary(mydata) #same scale of variables apart from ID
boxplot(mydata[2:13],pch=16,outcol="red",cex=1.25,lty=3)

Outliers = data.frame(Ecom=mydata[,3],SalesFImage=mydata[,8],OrdBilling=mydata[,11],DelSpeed=mydata[,12])
boxplot(Outliers ,pch=16,outcol="red",cex=1.25,lty=3, horizontal = TRUE,labels=TRUE)
abline(v=seq(1,9,0.25),col="lightgray")
abline(h=seq(1.5,4,1),col="lightgray")

boxplot.stats(mydata[,3])$out
boxplot.stats(mydata[,8])$out
boxplot.stats(mydata[,11])$out
boxplot.stats(mydata[,12])$out

### UNIVARIATE ANALYSIS ----
par(mfrow=c(1,2),mai = c(0.5, 0.5, 0.5, 0.5))
```

```
for(indexI in 2:13){

h <- hist(mydata[,indexI], main = paste0("Centrality of ",names(mydata[indexI]),"\n [St.Dev.=",round(sd(mydata[,indexI]),2),"]"),
ylim = c(0,50), labels = TRUE, col = "dark grey",xlab = NULL, cex.main=0.75,cex=0.5,cex.labels=0.5,breaks = 10)
mode <- h$mids[h$counts == max(h$counts)]
abline(v = mean(mydata[,indexI]),col = "royalblue",lwd = 2)
abline(v = median(mydata[,indexI]),col = "red",lwd = 2)
abline(v = mode,col = "green",lwd = 2)
legend("topright", inset=.01,c("Mean","Median","Mode"), fill=c("royalblue","red","green"), horiz=TRUE, cex=0.8)

boxplot(mydata[,indexI], horizontal = TRUE, main = paste0("Dispersion of ",names(mydata[indexI]),"\n
[St.Dev.=",round(sd(mydata[,indexI]),2),"]"), col="grey",xlab=NULL,cex.main=0.75)
text(fivenum(mydata[,indexI]), labels =fivenum(mydata[,indexI]), y=1.3, col = "blue", cex = 0.6)
rug(mydata[,indexI], side = 1, ticksize = 0.08, lwd = 0.9, col = "grey");
}



### BIVARIATE ANALYSIS ----
par(mfrow=c(1,1),mai = c(0.5, 0.5, 0.5, 0.5))
pairs.panels(mydata[2:13])
# Strong Correlations CompRes vs DelSpeed, TechSupport vs WartyClaim strong correlation
# Moderate Correlation in DelSpeed Vs OrdBilling, CompRes vs OrdBilling, Ecom vs SalesFImage

#DelSpeed vs CompRes
temp1 <- data.frame(DelSpeed=mydata$DelSpeed,CompRes=mydata$CompRes)
pairs.panels(temp1,gap=0,cex.cor = 0.5,cex=0.75,cex.labels=0.85)

#TechSup vs WartyClaim
temp2 <- data.frame(TechSup=mydata$TechSup,WartyClaim=mydata$WartyClaim)
pairs.panels(temp2,gap=0,cex.cor = 0.5,cex=0.75,cex.labels=0.85)

#SalesFImage vs Ecom
temp3 <- data.frame(SalesFImage=mydata$SalesFImage,Ecom=mydata$Ecom)
pairs.panels(temp3,gap=0,cex.cor = 0.5,cex=0.75,cex.labels=0.85)

#OrdBilling vs CompRes
temp4 <- data.frame(OrdBilling=mydata$OrdBilling,CompRes=mydata$CompRes)
pairs.panels(temp4,gap=0,cex.cor = 0.5,cex=0.75,cex.labels=0.85)
```

```
#OrdBilling vs DelSpeed
temp5 <- data.frame(OrdBilling=mydata$OrdBilling,DelSpeed=mydata$DelSpeed)
pairs.panels(temp5,gap=0,cex.cor = 0.5,cex=0.75,cex.labels=0.85)

#CompRes vs Satisfaction
temp6 <- data.frame(CompRes=mydata$CompRes,Satisfaction=mydata$Satisfaction)
pairs.panels(temp6,gap=0,cex.cor = 0.5,cex=0.75,cex.labels=0.85)

#ProdLine vs DelSpeed
temp7 <- data.frame(ProdLine=mydata$ProdLine,DelSpeed=mydata$DelSpeed)
pairs.panels(temp7,gap=0,cex.cor = 0.5,cex=0.75,cex.labels=0.85)

#DelSpeed vs Satisfaction
temp8 <- data.frame(DelSpeed=mydata$DelSpeed,Satisfaction=mydata$Satisfaction)
pairs.panels(temp8,gap=0,cex.cor = 0.5,cex=0.75,cex.labels=0.85)

#ProdLine vs CompRes
temp9 <- data.frame(ProdLine=mydata$ProdLine,CompRes=mydata$CompRes)
pairs.panels(temp9,gap=0,cex.cor = 0.5,cex=0.75,cex.labels=0.85)

#ProdLine vs Satisfaction
temp10 <- data.frame(ProdLine=mydata$ProdLine,Satisfaction=mydata$Satisfaction)
pairs.panels(temp10,gap=0,cex.cor = 0.5,cex=0.75,cex.labels=0.85)

#SalesFImage vs Advertising
temp11 <- data.frame(SalesFImage=mydata$SalesFImage,Advertising=mydata$Advertising)
pairs.panels(temp11,gap=0,cex.cor = 0.5,cex=0.75,cex.labels=0.85)

#OrdBilling vs Satisfaction
temp12 <- data.frame(OrdBilling=mydata$OrdBilling,Satisfaction=mydata$Satisfaction)
pairs.panels(temp12,gap=0,cex.cor = 0.5,cex=0.75,cex.labels=0.85)

#SalesFImage vs Satisfaction
temp13 <- data.frame(SalesFImage=mydata$SalesFImage,Satisfaction=mydata$Satisfaction)
pairs.panels(temp13,gap=0,cex.cor = 0.5,cex=0.75,cex.labels=0.85)

### SIMPLE LINEAR REGRESSION ----
## Lets make SLM models of all the significant correlated pairs identified above

### DelSpeed Vs CompRes SLM Model
```

```
SLM_Model_One <- lm(temp1$DelSpeed~temp1$CompRes)
summary(SLM_Model_One)

plot(temp1$DelSpeed~temp1$CompRes,pch=20,xlim=c(0,10),ylim=c(0,10))
abline(SLM_Model_One,col="red",lwd=2)
abline(v=median(temp1$CompRes),col="blue")
abline(h=median(temp1$DelSpeed),col="green")
text(median(temp1$CompRes),0,paste0("Median = ",median(temp1$CompRes)),pos=4)
text(0.5,median(temp1$DelSpeed),paste0("Median = ",median(temp1$DelSpeed)),pos=3)
text(0.6,SLM_Model_One$coefficients[1],paste0("Intercept = ",round(SLM_Model_One$coefficients[1],digits=2)),pos=1)

Prediction <- predict(SLM_Model_One,interval = "confidence", level = 0.99)
Actual <- temp1$DelSpeed
BackTrack <- data.frame(Actual,Prediction)

plot(BackTrack$Actual, col = "blue",
    pch=20, xlab="Number of Predictions", ylab="DelSpeed",
    ylim = c(1.5,5.5), xlim=c(1,50), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

lines(BackTrack$Actual, col = "blue", lwd=1)
lines(BackTrack$fit, col = "red", lwd=2)
abline(h=1,col="lightgray",lty="dotted")
abline(h=2,col="lightgray",lty="dotted")
abline(h=3,col="lightgray",lty="dotted")
abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")


### TechSup Vs WartyClaim SLM Model
Dependant_Variable = temp2$WartyClaim
Independant_Variable = temp2$TechSup

SLM_Model <- lm(Dependant_Variable~Independant_Variable)
summary(SLM_Model)

plot(Dependant_Variable~Independant_Variable,pch=20,xlim=c(0,10),ylim=c(0,10))
abline(SLM_Model,col="red",lwd=2)
abline(v=median(Independant_Variable),col="blue")
abline(h=median(Dependant_Variable),col="green")
```

```
text(median(Independant_Variable),0.5,paste0("Median = ",median(Independant_Variable)),pos=4)
text(0.5,median(Dependant_Variable),paste0("Median = ",median(Dependant_Variable)),pos=3)
text(0.6,SLM_Model$coefficients[1],paste0("Intercept = ",round(SLM_Model$coefficients[1],digits=2)),pos=1)

Prediction <- predict(SLM_Model,interval = "confidence", level = 0.99)
Actual <- Dependant_Variable
BackTrack <- data.frame(Actual,Prediction)

plot(BackTrack$Actual, col = "blue",
    pch=20, xlab="Number of Predictions", ylab="DelSpeed",
    ylim = c(3.5,8.0), xlim=c(1,50), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

lines(BackTrack$Actual, col = "blue", lwd=1)
lines(BackTrack$fit, col = "red", lwd=2)
abline(h=3,col="lightgray",lty="dotted")
abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")
abline(h=7,col="lightgray",lty="dotted")
abline(h=8,col="lightgray",lty="dotted")


### SalesFImage Vs Ecom SLM Model
Dependant_Variable = temp3$SalesFImage
Independant_Variable = temp3$Ecom

SLM_Model <- lm(Dependant_Variable~Independant_Variable)
summary(SLM_Model)

plot(Dependant_Variable~Independant_Variable,pch=20,xlim=c(0,10),ylim=c(0,10))
abline(SLM_Model,col="red",lwd=2)
abline(v=median(Independant_Variable),col="blue")
abline(h=median(Dependant_Variable),col="green")
text(median(Independant_Variable),0.5,paste0("Median = ",median(Independant_Variable)),pos=4)
text(0.5,median(Dependant_Variable),paste0("Median = ",median(Dependant_Variable)),pos=3)
text(0.6,SLM_Model$coefficients[1],paste0("Intercept = ",round(SLM_Model$coefficients[1],digits=2)),pos=4)

Prediction <- predict(SLM_Model,interval = "confidence", level = 0.99)
Actual <- Dependant_Variable
BackTrack <- data.frame(Actual,Prediction)
```

```
plot(BackTrack$Actual, col = "blue",
     pch=20, xlab="Number of Predictions", ylab="DelSpeed",
     ylim = c(3.0,8.0), xlim=c(1,50), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

lines(BackTrack$Actual, col = "blue", lwd=1)
lines(BackTrack$fit, col = "red", lwd=2)
abline(h=3,col="lightgray",lty="dotted")
abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")
abline(h=7,col="lightgray",lty="dotted")
abline(h=8,col="lightgray",lty="dotted")


### OrdBilling Vs CompRes SLM Model
Dependant_Variable = temp4$OrdBilling
Independant_Variable = temp4$CompRes

SLM_Model <- lm(Dependant_Variable~Independant_Variable)
summary(SLM_Model)

plot(Dependant_Variable~Independant_Variable,pch=20,xlim=c(0,10),ylim=c(0,10))
abline(SLM_Model,col="red",lwd=2)
abline(v=median(Independant_Variable),col="blue")
abline(h=median(Dependant_Variable),col="green")
text(median(Independant_Variable),0.5,paste0("Median = ",median(Independant_Variable)),pos=4)
text(0.5,median(Dependant_Variable),paste0("Median = ",median(Dependant_Variable)),pos=3)
text(0.6,SLM_Model$coefficients[1],paste0("Intercept = ",round(SLM_Model$coefficients[1],digits=2)),pos=4)
text(9,8,paste0("Slope = ",round(SLM_Model$coefficients[2],digits=2)),pos=3)

Prediction <- predict(SLM_Model,interval = "confidence", level = 0.99)
Actual <- Dependant_Variable
BackTrack <- data.frame(Actual,Prediction)

plot(BackTrack$Actual, col = "blue",
     pch=20, xlab="Number of Predictions", ylab="DelSpeed",
     ylim = c(2.0,7.0), xlim=c(1,50), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

lines(BackTrack$Actual, col = "blue", lwd=1)
lines(BackTrack$fit, col = "red", lwd=2)
abline(h=3,col="lightgray",lty="dotted")
abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")
abline(h=7,col="lightgray",lty="dotted")
abline(h=8,col="lightgray",lty="dotted")


### OrdBilling Vs DelSpeed SLM Model
Dependant_Variable = temp5$OrdBilling
Independant_Variable = temp5$DelSpeed

SLM_Model <- lm(Dependant_Variable~Independant_Variable)
summary(SLM_Model)

plot(Dependant_Variable~Independant_Variable,pch=20,xlim=c(0,10),ylim=c(0,10))
abline(SLM_Model,col="red",lwd=2)
abline(v=median(Independant_Variable),col="blue")
abline(h=median(Dependant_Variable),col="green")
text(median(Independant_Variable),0.5,paste0("Median = ",median(Independant_Variable)),pos=4)
text(0.5,median(Dependant_Variable),paste0("Median = ",median(Dependant_Variable)),pos=3)
text(0.6,SLM_Model$coefficients[1],paste0("Intercept = ",round(SLM_Model$coefficients[1],digits=2)),pos =4)
text(9,8,paste0("Slope = ",round(SLM_Model$coefficients[2],digits=2)),pos=3)

Prediction <- predict(SLM_Model,interval = "confidence", level = 0.99)
Actual <- Dependant_Variable
BackTrack <- data.frame(Actual,Prediction)

plot(BackTrack$Actual, col = "blue",
    pch=20, xlab="Number of Predictions", ylab="DelSpeed",
    ylim = c(2.0,7.0), xlim=c(1,50), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

lines(BackTrack$Actual, col = "blue", lwd=1)
lines(BackTrack$fit, col = "red", lwd=2)
abline(h=3,col="lightgray",lty="dotted")
abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")
abline(h=7,col="lightgray",lty="dotted")
abline(h=8,col="lightgray",lty="dotted")

### Satisfaction Vs CompRes SLM Model
Dependant_Variable = temp6$Satisfaction
Independant_Variable = temp6$CompRes

SLM_Model <- lm(Dependant_Variable~Independant_Variable)
summary(SLM_Model)

plot(Dependant_Variable~Independant_Variable,pch=20,xlim=c(0,10),ylim=c(0,10))
abline(SLM_Model,col="red",lwd=2)
abline(v=median(Independant_Variable),col="blue")
abline(h=median(Dependant_Variable),col="green")
text(median(Independant_Variable),0.5,paste0("Median = ",median(Independant_Variable)),pos=4)
text(0.5,median(Dependant_Variable),paste0("Median = ",median(Dependant_Variable)),pos=3)
text(0.6,SLM_Model$coefficients[1],paste0("Intercept = ",round(SLM_Model$coefficients[1],digits=2)),pos=4)
text(8,8,paste0("Slope = ",round(SLM_Model$coefficients[2],digits=2)),pos=4)

Prediction <- predict(SLM_Model,interval = "confidence", level = 0.99)
Actual <- Dependant_Variable
BackTrack <- data.frame(Actual,Prediction)

plot(BackTrack$Actual, col = "blue",
    pch=20, xlab="Number of Predictions", ylab="DelSpeed",
    ylim = c(4.0,10), xlim=c(1,50), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

lines(BackTrack$Actual, col = "blue", lwd=1)
lines(BackTrack$fit, col = "red", lwd=2)
abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")
abline(h=7,col="lightgray",lty="dotted")
abline(h=8,col="lightgray",lty="dotted")
abline(h=9,col="lightgray",lty="dotted")



### DelSpeed Vs ProdLine SLM Model
Dependant_Variable = temp7$DelSpeed
Independant_Variable = temp7$ProdLine
```

```
SLM_Model <- lm(Dependant_Variable~Independant_Variable)
summary(SLM_Model)

plot(Dependant_Variable~Independant_Variable,pch=20,xlim=c(0,10),ylim=c(0,10))
abline(SLM_Model,col="red",lwd=2)
abline(v=median(Independant_Variable),col="blue")
abline(h=median(Dependant_Variable),col="green")
text(median(Independant_Variable),0.5,paste0("Median = ",median(Independant_Variable)),pos=4)
text(0.5,median(Dependant_Variable),paste0("Median = ",median(Dependant_Variable)),pos=3)
text(0,1,paste0("Intercept = ",round(SLM_Model$coefficients[1],digits=2)),pos=4)
text(8,8,paste0("Slope = ",round(SLM_Model$coefficients[2],digits=2)),pos=4)

Prediction <- predict(SLM_Model,interval = "confidence", level = 0.99)
Actual <- Dependant_Variable
BackTrack <- data.frame(Actual,Prediction)

plot(BackTrack$Actual, col = "blue",
     pch=20, xlab="Number of Predictions", ylab="DelSpeed",
     ylim = c(2.0,6), xlim=c(1,50), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

lines(BackTrack$Actual, col = "blue", lwd=1)
lines(BackTrack$fit, col = "red", lwd=2)
abline(h=2,col="lightgray",lty="dotted")
abline(h=3,col="lightgray",lty="dotted")
abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")
abline(h=7,col="lightgray",lty="dotted")

### Satisfaction Vs DelSpeed SLM Model
Dependant_Variable = temp8$Satisfaction
Independant_Variable = temp8$DelSpeed

SLM_Model <- lm(Dependant_Variable~Independant_Variable)
summary(SLM_Model)

plot(Dependant_Variable~Independant_Variable,pch=20,xlim=c(0,10),ylim=c(0,10))
abline(SLM_Model,col="red",lwd=2)
abline(v=median(Independant_Variable),col="blue")
abline(h=median(Dependant_Variable),col="green")
text(median(Independant_Variable),0.5,paste0("Median = ",median(Independant_Variable)),pos=4)
text(0.5,median(Dependant_Variable),paste0("Median = ",median(Dependant_Variable)),pos=3)
text(0,1,paste0("Intercept = ",round(SLM_Model$coefficients[1],digits=2)),pos=4)
text(8,8,paste0("Slope = ",round(SLM_Model$coefficients[2],digits=2)),pos=4)

Prediction <- predict(SLM_Model,interval = "confidence", level = 0.99)
Actual <- Dependant_Variable
BackTrack <- data.frame(Actual,Prediction)

plot(BackTrack$Actual, col = "blue",
    pch=20, xlab="Number of Predictions", ylab="DelSpeed",
    ylim = c(4.0,10), xlim=c(1,50), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

lines(BackTrack$Actual, col = "blue", lwd=1)
lines(BackTrack$fit, col = "red", lwd=2)
abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")
abline(h=7,col="lightgray",lty="dotted")
abline(h=8,col="lightgray",lty="dotted")
abline(h=9,col="lightgray",lty="dotted")


### ProdLine Vs CompRes SLM Model
Dependant_Variable = temp9$ProdLine
Independant_Variable = temp9$CompRes

SLM_Model <- lm(Dependant_Variable~Independant_Variable)
summary(SLM_Model)

plot(Dependant_Variable~Independant_Variable,pch=20,xlim=c(0,10),ylim=c(0,10))
abline(SLM_Model,col="red",lwd=2)
abline(v=median(Independant_Variable),col="blue")
abline(h=median(Dependant_Variable),col="green")
text(median(Independant_Variable),0.5,paste0("Median = ",median(Independant_Variable)),pos=4)
text(0.5,median(Dependant_Variable),paste0("Median = ",median(Dependant_Variable)),pos=3)
text(0,1,paste0("Intercept = ",round(SLM_Model$coefficients[1],digits=2)),pos=4)
text(8,8,paste0("Slope = ",round(SLM_Model$coefficients[2],digits=2)),pos=3)
```

```
Prediction <- predict(SLM_Model,interval = "confidence", level = 0.99)
Actual <- Dependant_Variable
BackTrack <- data.frame(Actual,Prediction)

plot(BackTrack$Actual, col = "blue",
    pch=20, xlab="Number of Predictions", ylab="DelSpeed",
    ylim = c(2.0,9), xlim=c(1,50), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

lines(BackTrack$Actual, col = "blue", lwd=1)
lines(BackTrack$fit, col = "red", lwd=2)
abline(h=3,col="lightgray",lty="dotted")
abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")
abline(h=7,col="lightgray",lty="dotted")
abline(h=8,col="lightgray",lty="dotted")


### Satisfaction Vs ProdLine SLM Model
Dependant_Variable = temp10$Satisfaction
Independant_Variable = temp10$ProdLine

SLM_Model <- lm(Dependant_Variable~Independant_Variable)
summary(SLM_Model)

plot(Dependant_Variable~Independant_Variable,pch=20,xlim=c(0,10),ylim=c(0,10))
abline(SLM_Model,col="red",lwd=2)
abline(v=median(Independant_Variable),col="blue")
abline(h=median(Dependant_Variable),col="green")
text(median(Independant_Variable),0.5,paste0("Median = ",median(Independant_Variable)),pos=4)
text(0.5,median(Dependant_Variable),paste0("Median = ",median(Dependant_Variable)),pos=3)
text(0,1,paste0("Intercept = ",round(SLM_Model$coefficients[1],digits=2)),pos=4)
text(8.5,8,paste0("Slope = ",round(SLM_Model$coefficients[2],digits=2)),pos=4)

Prediction <- predict(SLM_Model,interval = "confidence", level = 0.99)
Actual <- Dependant_Variable
BackTrack <- data.frame(Actual,Prediction)

plot(BackTrack$Actual, col = "blue",
    pch=20, xlab="Number of Predictions", ylab="DelSpeed",
    ylim = c(4.0,10), xlim=c(1,50), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

lines(BackTrack$Actual, col = "blue", lwd=1)
lines(BackTrack$fit, col = "red", lwd=2)
abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")
abline(h=7,col="lightgray",lty="dotted")
abline(h=8,col="lightgray",lty="dotted")
abline(h=9,col="lightgray",lty="dotted")


### SalesFImage Vs Advertising SLM Model
Dependant_Variable = temp11$SalesFImage
Independant_Variable = temp11$Advertising

SLM_Model <- lm(Dependant_Variable~Independant_Variable)
summary(SLM_Model)

plot(Dependant_Variable~Independant_Variable,pch=20,xlim=c(0,10),ylim=c(0,10))
abline(SLM_Model,col="red",lwd=2)
abline(v=median(Independant_Variable),col="blue")
abline(h=median(Dependant_Variable),col="green")
text(median(Independant_Variable),0.5,paste0("Median = ",median(Independant_Variable)),pos=4)
text(0.5,median(Dependant_Variable),paste0("Median = ",median(Dependant_Variable)),pos=3)
text(0,1,paste0("Intercept = ",round(SLM_Model$coefficients[1],digits=2)),pos=4)
text(8.5,8,paste0("Slope = ",round(SLM_Model$coefficients[2],digits=2)),pos=3)

Prediction <- predict(SLM_Model,interval = "confidence", level = 0.99)
Actual <- Dependant_Variable
BackTrack <- data.frame(Actual,Prediction)

plot(BackTrack$Actual, col = "blue",
    pch=20, xlab="Number of Predictions", ylab="DelSpeed",
    ylim = c(2.5,8.5), xlim=c(1,50), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

lines(BackTrack$Actual, col = "blue", lwd=1)
lines(BackTrack$fit, col = "red", lwd=2)
abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")
abline(h=7,col="lightgray",lty="dotted")
abline(h=8,col="lightgray",lty="dotted")
abline(h=3,col="lightgray",lty="dotted")



### Satisfaction Vs OrdBilling SLM Model
Dependant_Variable = temp12$Satisfaction
Independant_Variable = temp12$OrdBilling

SLM_Model <- lm(Dependant_Variable~Independant_Variable)
summary(SLM_Model)

plot(Dependant_Variable~Independant_Variable,pch=20,xlim=c(0,10),ylim=c(0,10))
abline(SLM_Model,col="red",lwd=2)
abline(v=median(Independant_Variable),col="blue")
abline(h=median(Dependant_Variable),col="green")
text(median(Independant_Variable),0.5,paste0("Median = ",median(Independant_Variable)),pos=4)
text(0.5,median(Dependant_Variable),paste0("Median = ",median(Dependant_Variable)),pos=3)
text(0,1,paste0("Intercept = ",round(SLM_Model$coefficients[1],digits=2)),pos=4)
text(8.5,8,paste0("Slope = ",round(SLM_Model$coefficients[2],digits=2)),pos=4)

Prediction <- predict(SLM_Model,interval = "confidence", level = 0.99)
Actual <- Dependant_Variable
BackTrack <- data.frame(Actual,Prediction)

plot(BackTrack$Actual, col = "blue",
     pch=20, xlab="Number of Predictions", ylab="DelSpeed",
     ylim = c(4,10.0), xlim=c(1,50), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

lines(BackTrack$Actual, col = "blue", lwd=1)
lines(BackTrack$fit, col = "red", lwd=2)
abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")
abline(h=7,col="lightgray",lty="dotted")
abline(h=8,col="lightgray",lty="dotted")
abline(h=9,col="lightgray",lty="dotted")
```

```
### Satisfaction Vs SalesFImage SLM Model
Dependant_Variable = temp13$Satisfaction
Independant_Variable = temp13$SalesFImage

SLM_Model <- lm(Dependant_Variable~Independant_Variable)
summary(SLM_Model)

plot(Dependant_Variable~Independant_Variable,pch=20,xlim=c(0,10),ylim=c(0,10))
abline(SLM_Model,col="red",lwd=2)
abline(v=median(Independant_Variable),col="blue")
abline(h=median(Dependant_Variable),col="green")
text(median(Independant_Variable),0.5,paste0("Median = ",median(Independant_Variable)),pos=4)
text(0.5,median(Dependant_Variable),paste0("Median = ",median(Dependant_Variable)),pos=3)
text(0,1,paste0("Intercept = ",round(SLM_Model$coefficients[1],digits=2)),pos=4)
text(8.5,8,paste0("Slope = ",round(SLM_Model$coefficients[2],digits=2)),pos=4)

Prediction <- predict(SLM_Model,interval = "confidence", level = 0.99)
Actual <- Dependant_Variable
BackTrack <- data.frame(Actual,Prediction)

plot(BackTrack$Actual, col = "blue",
    pch=20, xlab="Number of Predictions", ylab="DelSpeed",
    ylim = c(4,10.0), xlim=c(1,50), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

lines(BackTrack$Actual, col = "blue", lwd=1)
lines(BackTrack$fit, col = "red", lwd=2)
abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")
abline(h=7,col="lightgray",lty="dotted")
abline(h=8,col="lightgray",lty="dotted")
abline(h=9,col="lightgray",lty="dotted")

## specialized the insignificant value according to the significant level
#Satisfaction based dataset with about 50% correlated
names(mydata)
Satisfaction_Pairs = mydata
par(mfrow=c(1,1),mai = c(0.5, 0.5, 0.5, 0.5))
Satisfaction_Pairs = Satisfaction_Pairs[,-1]
Satisfaction_Pairs = Satisfaction_Pairs[,-2]
Satisfaction_Pairs = Satisfaction_Pairs[,-2]
Satisfaction_Pairs = Satisfaction_Pairs[,-3]
Satisfaction_Pairs = Satisfaction_Pairs[,-5]
Satisfaction_Pairs = Satisfaction_Pairs[,-5]
pairs.panels(Satisfaction_Pairs)

Satisfaction_Pairs[7]

par(mfrow=c(1,3),mai = c(0.3, 0.25, 0.8, 0.25))

for(indexLoop in 1:6){

  Sub1 = 7
  Sub2 = indexLoop

  SLM_Model <- lm(Satisfaction_Pairs[,Sub1]~Satisfaction_Pairs[,Sub2])
  Summary_Model <- summary(SLM_Model)
  Equation <-
    paste0("Linear Model: Y =",round(SLM_Model$coefficients[1],digits=3),
           "+",round(SLM_Model$coefficients[2],digits=3),
           "X     R-Square=",round(Summary_Model$r.squared,digits=3),"       \nP-Value
=",round(Summary_Model$coefficients[2,4],digits = 11))

  Prediction <- predict(SLM_Model,interval = "confidence", level = 0.99)
  Actual <- Satisfaction_Pairs[,Sub1]
  BackTrack <- data.frame(Actual,Prediction)

  plot(BackTrack$Actual, col = "blue",
       pch=20, xlab="Number of Predictions", ylab="Satisfaction",
       main=paste("",names(Satisfaction_Pairs[Sub1])," ~ ",
                  names(Satisfaction_Pairs[Sub2])," ","\n",Equation)  ,
       ylim = c(4,10), xlim=c(1,25), cex.main=1.0, cex.axis=0.8,cex.lab=0.8)

  lines(BackTrack$Actual, col = "blue", lwd=1)
  lines(BackTrack$fit, col = "red", lwd=2)
  abline(h=4,col="lightgray",lty="dotted")
  abline(h=5,col="lightgray",lty="dotted")
  abline(h=6,col="lightgray",lty="dotted")
  abline(h=7,col="lightgray",lty="dotted")
  abline(h=8,col="lightgray",lty="dotted")
  abline(h=9,col="lightgray",lty="dotted")
  abline(h=10,col="lightgray",lty="dotted")

}

### MultiCollinearity ----

#making model to employ VIF to check multicollinearity
Multi_Colinear_Model <- lm(Satisfaction~.,data = mydata)
summary(Multi_Colinear_Model)

Prediction <- predict(Multi_Colinear_Model,interval = "confidence", level = 0.95)
Actual <- mydata$Satisfaction
BackTrack2 <- data.frame(Actual,Prediction)
BackTrack2

plot(BackTrack2$Actual, col = "blue", pch=20, xlab="Number of Predictions", ylab="Revenue", main = NULL, ylim = c(4,10),
xlim=c(1,50))
lines(BackTrack2$Actual, col = "blue", lwd=1)
points(BackTrack2$fit, col = "red",pch=20,lwd=2)
lines(BackTrack2$fit, col = "red", lwd=2)

# Variance Inflation factor to check multicollinearity if more than 5 threshold
my_VIF <- vif(Multi_Colinear_Model)
par(mfrow=c(1,1),mai = c(0.5, 0.5, 0.5, 0.5))
plot(my_VIF,type="b",pch=20,col="blue", main="Variance Inflation Factor (VIF)",lwd=3)
text(2,unname(my_VIF[2]),names(my_VIF[2]),pos=4,cex=0.8,col="black")
text(3,unname(my_VIF[3]),names(my_VIF[3]),pos=2,cex=0.8,col="black")
text(4,unname(my_VIF[4]),names(my_VIF[4]),pos=4,cex=0.8,col="black")
text(5,unname(my_VIF[5]),names(my_VIF[5]),pos=4,cex=0.8,col="black")
text(6,unname(my_VIF[6]),names(my_VIF[6]),pos=1,cex=0.8,col="black")
text(7,unname(my_VIF[7]),names(my_VIF[7]),pos=2,cex=0.8,col="black")
text(8,unname(my_VIF[8]),names(my_VIF[8]),pos=3,cex=0.8,col="black")
text(9,unname(my_VIF[9]),names(my_VIF[9]),pos=1,cex=0.8,col="black")
text(10,unname(my_VIF[10]),names(my_VIF[10]),pos=3,cex=0.8,col="black")
text(11,unname(my_VIF[11]),names(my_VIF[11]),pos=1,cex=0.8,col="black")
text(12,unname(my_VIF[12]),names(my_VIF[12]),pos=2,cex=0.8,col="black")

abline(h=5,col="red",lwd=2)
```

```
text(0.8,5.2,"Multicollinearity Region",pos=4,cex=0.9,col="red")
text(0.8,4.8,"Normal Region",pos=4,cex=0.9,col="blue")

Temp_Data_Df <- mydata
Temp_Data_Df <- Temp_Data_Df[2:13]
Temp_Data_Df <- Temp_Data_Df[-11]

Non_Multi_Colinear_Model <- lm(Satisfaction~.,data = Temp_Data_Df)

str(Temp_Data_Df)

Prediction <- predict(Non_Multi_Colinear_Model,interval = "confidence", level = 0.95)
Actual <- Temp_Data_Df$Satisfaction
BackTrack2 <- data.frame(Actual,Prediction)
BackTrack2

plot(BackTrack2$Actual, col = "blue", pch=20, xlab="Number of Predictions", ylab="Revenue", main = NULL, ylim = c(4,10),
xlim=c(1,50))
lines(BackTrack2$Actual, col = "blue", lwd=1)
points(BackTrack2$fit, col = "red",pch=20,lwd=2)
lines(BackTrack2$fit, col = "red", lwd=2)

abline(h=4,col="lightgray",lty="dotted")
abline(h=5,col="lightgray",lty="dotted")
abline(h=6,col="lightgray",lty="dotted")
abline(h=7,col="lightgray",lty="dotted")
abline(h=8,col="lightgray",lty="dotted")
abline(h=9,col="lightgray",lty="dotted")
abline(h=10,col="lightgray",lty="dotted")

# Variance Inflation factor to check multicollinearity if more than 5 threshold
# after deleting DELSPEED from the regression model
my_VIF <- vif(Non_Multi_Colinear_Model)
par(mfrow=c(1,1),mai = c(0.5, 0.5, 0.5, 0.5))
plot(my_VIF,type="b",pch=20,col="blue", main="Variance Inflation Factor (VIF)",lwd=3, ylim=c(0,10))
text(1,unname(my_VIF[1]),names(my_VIF[1]),pos=1,cex=0.8,col="black")
text(2,unname(my_VIF[2]),names(my_VIF[2]),pos=1,cex=0.8,col="black")
text(3,unname(my_VIF[3]),names(my_VIF[3]),pos=1,cex=0.8,col="black")
text(4,unname(my_VIF[4]),names(my_VIF[4]),pos=3,cex=0.8,col="black")
text(5,unname(my_VIF[5]),names(my_VIF[5]),pos=1,cex=0.8,col="black")
text(6,unname(my_VIF[6]),names(my_VIF[6]),pos=3,cex=0.8,col="black")
text(7,unname(my_VIF[7]),names(my_VIF[7]),pos=3,cex=0.8,col="black")
text(8,unname(my_VIF[8]),names(my_VIF[8]),pos=1,cex=0.8,col="black")
text(9,unname(my_VIF[9]),names(my_VIF[9]),pos=3,cex=0.8,col="black")
text(10,unname(my_VIF[10]),names(my_VIF[10]),pos=2,cex=0.8,col="black")
abline(h=5,col="red",lwd=2)
text(0.8,5.5,"Multicollinearity Region",pos=4,cex=0.9,col="red")
text(0.8,4.5,"Normal Region",pos=4,cex=0.9,col="blue")



### Principal Component Factor ----

# Eigen Values apart from ID and Satisfaction variables
ev <- eigen(cor(mydata[2:12]))
ev
EigenValues <- ev$values
Factor <- c(1,2,3,4,5,6,7,8,9,10,11)

#Number of factors for PCA
Scree <- data.frame(Factor,EigenValues)
plot(Scree,main="Scree Plot", col="Blue",ylim=c(0,4),ylab="Eigen Values", xlab="Variables", pch=20, type="b", lwd=2)
abline(h=1,col="red")
text(Scree, labels = round(EigenValues,3), pos=4)
text(9,1.1,labels = "Kaiser's Criterion (Eigen Value > 1)", cex=1, col="red")

Unrotate=principal(mydata[2:12], nfactors=4, rotate="none")
print(Unrotate,digits=3)
UnrotatedProfile=plot(Unrotate,row.names(Unrotate$loadings))

#verimax rotation to help identifying factors and naming them
Rotate=principal(mydata[2:12],nfactors=4,rotate="varimax")
print(Rotate,digits=3)
RotatedProfile=plot(Rotate,cex=1.0)

temp = mydata[2:12]
RFA <- factanal(temp,factors=4,rotation = "varimax")
print(RFA,digits = 3)
FAVars <- RFA$rotmat
FA_One <- FAVars[,1]
```

```
FA_Two <- FAVars[,2]

fData <- Rotate$scores
SLA_Factor <- fData[,1]
Marketing_Factor <- fData[,2]
Support_Factor <- fData[,3]
Quality_Factor <- fData[,4]

CorrDS <- data.frame(SLA_Factor,Marketing_Factor,Support_Factor,Quality_Factor,mydata$Satisfaction)
corrplot(cor(CorrDS),col=c("orange","darkblue"),tl.col = c("orange","darkblue"),tl.cex = 0.9,method = "number",type="upper")

### MULTIPLE LINEAR REGRESSION ----

# Preparing dataset

MLR_Model <- lm(mydata.Satisfaction~.,data = CorrDS)
summary(MLR_Model)

# Satisfaction = 6.918 + 0.618*SLA_Factor + 0.509*Marketing_factor + 0.067*Support_Factor + 0.540*Product_Factor

# backtracking to see effectiveness visually

Prediction <- predict(MLR_Model,interval = "confidence", level = 0.95)
Actual <- mydata$Satisfaction
BackTrack <- data.frame(Actual,Prediction)
BackTrack

par(mfrow=c(1,1),mai = c(0.25, 0.25, 0.25, 0.25))
  plot(BackTrack$Actual, col = "gray", pch=20, xlab="Number of Predictions", ylab="Revenue", main = NULL, ylim = c(4.5,9.0),
xlim=c(1,100),lwd=5)
  lines(BackTrack$Actual, col = "gray", lwd=5)
  points(BackTrack$fit, col = "red",pch=20,lwd=5)
  lines(BackTrack$fit, col = "red", lwd=5)

  plot(BackTrack$Actual, col = "gray", pch=20, xlab="Number of Predictions", ylab="Revenue", main = NULL, ylim = c(4.5,9.0),
xlim=c(1,100),lwd=5)
  lines(BackTrack$Actual, col = "gray", lwd=5)
  points(BackTrack2$fit, col = "blue",pch=20,lwd=5)
  lines(BackTrack2$fit, col = "blue", lwd=5)
  grid(20,col="lightgray")

#Validity of the model
par(mfrow=c(2,2),mai = c(0.35, 0.35, 0.35, 0.35))
plot(MLR_Model)

### CONCLUSION ----
plot(Multi_Colinear_Model)
#Comparing PCA based MLR with NORMAL MultiColinear Model
```