 Product Service

Management Report

2019

**June 28, 2019**

**Product Service Management Report**

**Authored By: Deepti Lobo**

**CONTENTS**

[Project Objective 3](#_Toc12874008)

[Known Facts 3](#_Toc12874009)

[Exploratory Data Analysis (EDA) 5](#_Toc12874010)

[Descriptive Statistics 7](#_Toc12874011)

[Data Visualization 8](#_Toc12874012)

[Check for correlation among variables 9](#_Toc12874013)

[Performing Simple Linear Regression 13](#_Toc12874014)

[Performing Multiple Linear Regression 27](#_Toc12874015)

[Assumption Checks for MLR 38](#_Toc12874016)

[Assumption of Multi-colinearity 46](#_Toc12874017)

[Performing PCA 47](#_Toc12874018)

[Lets again do MLR with new dataset 56](#_Toc12874019)

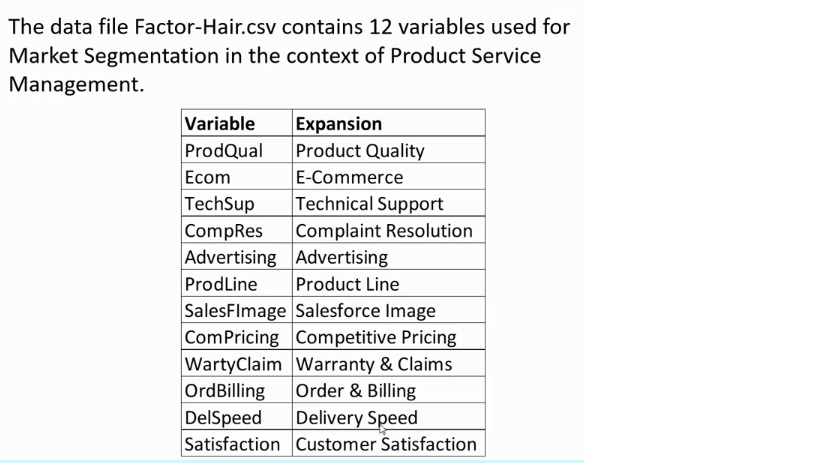
[Conclusion 60](#_Toc12874020)

# 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

1. Perform exploratory data analysis on the dataset. Showcase some charts, graphs. Check for outliers and missing values.
2. Is there evidence of multicollinearity? Showcase your analysis.
3. Perform simple linear regression for the dependent variable with every independent variable.
4. Perform PCA/Factor analysis by extracting 4 factors. Interpret the output and name the Factors.
5. 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.

# Known Facts





#Setup the working directory  
setwd("D:/R-3.5.3/working\_directory")  
getwd()

## [1] "D:/R-3.5.3/working\_directory"

#Import packages  
library(tidyverse)

## -- Attaching packages ---------------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.1 v purrr 0.3.2   
## v tibble 2.1.1 v dplyr 0.8.0.1  
## v tidyr 0.8.3 v stringr 1.4.0   
## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(ggplot2)   
library(car)

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following object is masked from 'package:purrr':  
##   
## some

library(psych)

##   
## Attaching package: 'psych'

## The following object is masked from 'package:car':  
##   
## logit

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library(caTools)   
library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(Metrics)  
library(purrr)  
library(tidyr)  
library(corrplot)

## corrplot 0.84 loaded

#Read the Input File  
mydata = read.csv("Factor-Hair-Revised.csv",header = TRUE)  
#mydata1 = mydata  
attach(mydata)

# Exploratory Data Analysis (EDA)

# Number of rows and columns  
print(paste("Number of rows = ",nrow(mydata),sep=""))

## [1] "Number of rows = 100"

print(paste("Number of columns = ",ncol(mydata),sep=""))

## [1] "Number of columns = 13"

# Names of the columns  
names(mydata)

## [1] "ID" "ProdQual" "Ecom" "TechSup"   
## [5] "CompRes" "Advertising" "ProdLine" "SalesFImage"   
## [9] "ComPricing" "WartyClaim" "OrdBilling" "DelSpeed"   
## [13] "Satisfaction"

# Datatypes of the columns  
str(mydata)

## '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 ...

#Is any missing values ?  
anyNA(mydata)

## [1] FALSE

colSums(is.na(mydata))

## ID ProdQual Ecom TechSup CompRes   
## 0 0 0 0 0   
## Advertising ProdLine SalesFImage ComPricing WartyClaim   
## 0 0 0 0 0   
## OrdBilling DelSpeed Satisfaction   
## 0 0 0

#Top 6 rows of the dataset  
head(mydata)

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

#Bottom 6 rows of the dataset  
tail(mydata)

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

# Descriptive Statistics

#Summary of the dataset  
summary(mydata)

## 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

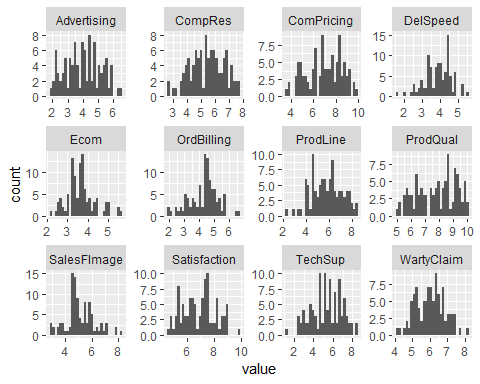
#Ignore Non numeric and unwanted variables such as ID  
mydata = mydata[,-1]  
  
str(mydata)

## 'data.frame': 100 obs. of 12 variables:  
## $ 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 ...

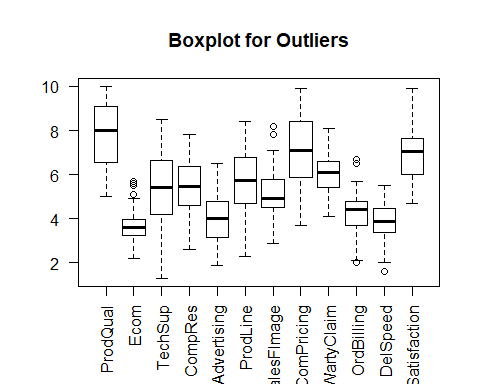
# Data Visualization

## Histogram of the independent Variables  
  
mydata %>% keep(is.numeric) %>% gather() %>%   
 ggplot(aes(value)) + facet\_wrap(~key, scales = "free") + geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



#Box Plot to see the Outliers in Continuous Variables.  
boxplot(mydata,main = "Boxplot for Outliers" , las = 2)



Most of the variables don’t have outliers, except for Ecom, SalesFImage, OrderBilling and DelSpeed.

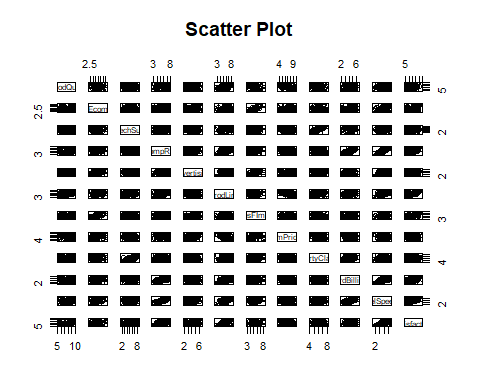
# Check for correlation among variables

If one or more of the variables in a model are correlated, then the model may produce unstable parameter estimates with highly inflated standard errors, resulting in an overall significant model with no significant predictors. Therefore, we need to check, if a correlation exists among the variables.

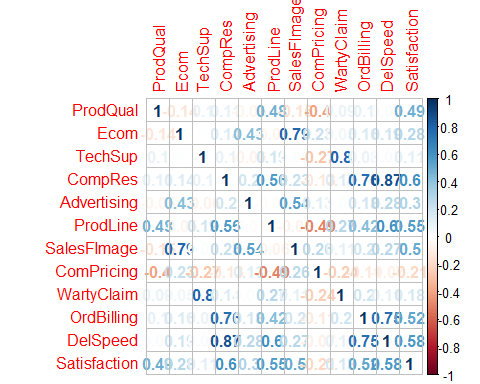
mydataCorr = cor(mydata)  
mydataCorr

## ProdQual Ecom TechSup CompRes  
## ProdQual 1.00000000 -0.1371632174 0.0956004542 0.1063700  
## Ecom -0.13716322 1.0000000000 0.0008667887 0.1401793  
## TechSup 0.09560045 0.0008667887 1.0000000000 0.0966566  
## CompRes 0.10637000 0.1401792611 0.0966565978 1.0000000  
## Advertising -0.05347313 0.4298907110 -0.0628700668 0.1969168  
## ProdLine 0.47749341 -0.0526878383 0.1926254565 0.5614170  
## SalesFImage -0.15181287 0.7915437115 0.0169905395 0.2297518  
## ComPricing -0.40128188 0.2294624014 -0.2707866821 -0.1279543  
## WartyClaim 0.08831231 0.0518981915 0.7971679258 0.1404083  
## OrdBilling 0.10430307 0.1561473316 0.0801018246 0.7568686  
## DelSpeed 0.02771800 0.1916360683 0.0254406935 0.8650917  
## Satisfaction 0.48632500 0.2827450147 0.1125971788 0.6032626  
## Advertising ProdLine SalesFImage ComPricing WartyClaim  
## ProdQual -0.05347313 0.47749341 -0.15181287 -0.40128188 0.08831231  
## Ecom 0.42989071 -0.05268784 0.79154371 0.22946240 0.05189819  
## TechSup -0.06287007 0.19262546 0.01699054 -0.27078668 0.79716793  
## CompRes 0.19691685 0.56141695 0.22975176 -0.12795425 0.14040830  
## Advertising 1.00000000 -0.01155082 0.54220366 0.13421689 0.01079207  
## ProdLine -0.01155082 1.00000000 -0.06131553 -0.49494840 0.27307753  
## SalesFImage 0.54220366 -0.06131553 1.00000000 0.26459655 0.10745534  
## ComPricing 0.13421689 -0.49494840 0.26459655 1.00000000 -0.24498605  
## WartyClaim 0.01079207 0.27307753 0.10745534 -0.24498605 1.00000000  
## OrdBilling 0.18423559 0.42440825 0.19512741 -0.11456703 0.19706512  
## DelSpeed 0.27586308 0.60185021 0.27155126 -0.07287173 0.10939460  
## Satisfaction 0.30466947 0.55054594 0.50020531 -0.20829569 0.17754482  
## OrdBilling DelSpeed Satisfaction  
## ProdQual 0.10430307 0.02771800 0.4863250  
## Ecom 0.15614733 0.19163607 0.2827450  
## TechSup 0.08010182 0.02544069 0.1125972  
## CompRes 0.75686859 0.86509170 0.6032626  
## Advertising 0.18423559 0.27586308 0.3046695  
## ProdLine 0.42440825 0.60185021 0.5505459  
## SalesFImage 0.19512741 0.27155126 0.5002053  
## ComPricing -0.11456703 -0.07287173 -0.2082957  
## WartyClaim 0.19706512 0.10939460 0.1775448  
## OrdBilling 1.00000000 0.75100307 0.5217319  
## DelSpeed 0.75100307 1.00000000 0.5770423  
## Satisfaction 0.52173191 0.57704227 1.0000000

#Scatter Plot  
plot(mydata, main = "Scatter Plot")



#Is there a correlation among the variables   
corrplot(mydataCorr,method = "number")



The correlation matrix ranges between -1 to +1. The data marked in red has negative correlation and the ones in blue have positive correlation. As we can see, a correlation ship exists between the variables. For example, ECom has a correlation of 0.79 with SalesFImage, CompRes has a correlation of 0.87 with DelSpeed, etc.

To check for multicollinearity in independent variables, we use VIF - Variable Inflation Factor. This measures the impact of collinearity among the variables in a regression model. The Variance Inflation Factor (VIF) is 1/Tolerance, it is always greater than or equal to 1.

#We are considering the VIF values greater than 2.5 as having high multicollinearlity.  
  
#Full model by taking satisfaction as the dependent variable.  
  
full\_model = lm(Satisfaction~., data = mydata)  
vif(full\_model)

## ProdQual Ecom TechSup CompRes Advertising ProdLine   
## 1.635797 2.756694 2.976796 4.730448 1.508933 3.488185   
## SalesFImage ComPricing WartyClaim OrdBilling DelSpeed   
## 3.439420 1.635000 3.198337 2.902999 6.516014

Observation

The correlation matrix, along with the graphs, showed that correlation existed among the variables. The VIF test shows, that multiple variables have the value greater than 2.5. With DelSpeed having the highest value of 6.516014. We may conclude that this model is not good for regression as multi-collinearity exists.

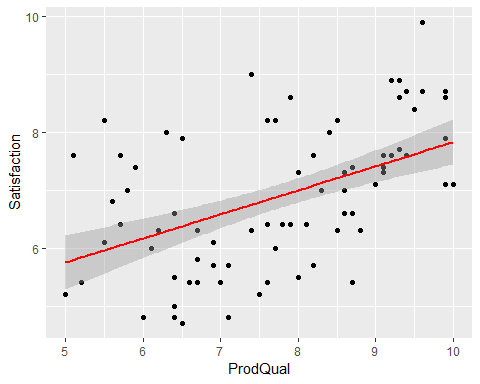
# Performing Simple Linear Regression

Linear regression models are used to show or predict the relationship between two variables or factors. The factor that is being predicted (the factor that the equation solves for) is called the dependent variable. The factors that are used to predict the value of the dependent variable are called the independent variables. Here Customer Satisfaction is being considered as the dependent variable for testing SLR. The equation for calculation is Y = b0 + b1\*X.

#SLR for ProdQual  
ProdQual\_lm = lm(Satisfaction~ProdQual, data = mydata)  
summary(ProdQual\_lm)

##   
## Call:  
## lm(formula = Satisfaction ~ ProdQual, data = mydata)  
##   
## 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 \*\*\*  
## ProdQual 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

#PLot the linear model (line of best fit)  
qplot(ProdQual,Satisfaction,data= mydata) + stat\_smooth(method="lm", col="red")



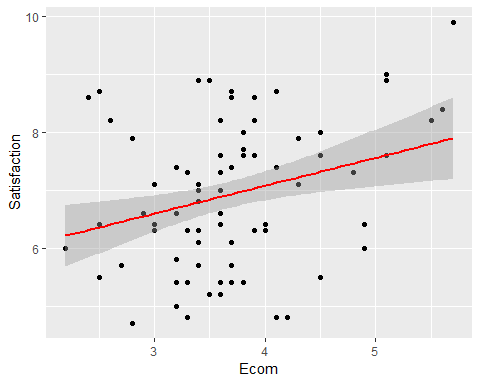
Observation

The estimated regression line equation, Satisfation = 3.67593 + 0.41512 \* ProdQual. This means, an additional rating of 1 in Product Quality will raise the Satisfaction by 4.09. (i.e) Satifaction = 3.67593 + 0.41512 \* 1 = 4.09105. The p-value of 2.901e-07 < 0.05 is highly significant. The R2 is 0.2365 which is around 23.65%. It implies that 23.65% of the variation in the satisfaction is explained by the Product Quality.

#SLR for Ecom  
Ecom\_lm = lm(Satisfaction~Ecom, data = mydata)  
summary(Ecom\_lm)

##   
## Call:  
## lm(formula = Satisfaction ~ Ecom, data = mydata)  
##   
## 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 \*\*\*  
## Ecom 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

#PLot the linear model (line of best fit)  
qplot(Ecom,Satisfaction,data= mydata) + stat\_smooth(method="lm", col="red")



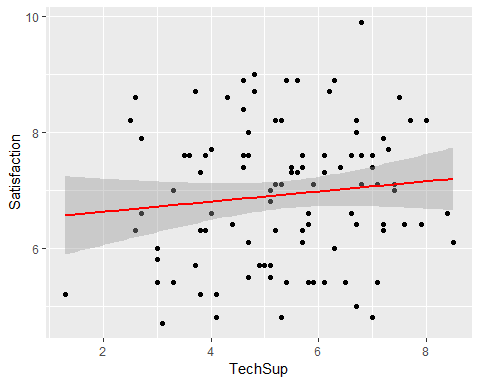
Observation

The estimated regression line equation, Satisfation = 5.1516 + 0.4811 \* Ecom. This means, an additional rating of 1 in E-Commerce will raise the Satisfaction by 5.63. (i.e) Satifaction = 5.1516 + 0.4811 \* 1 = 5.6327. The p-value of 0.004368 < 0.05 is highly significant. The R2 is 0.07994 which is around 7.99%. It implies that 7.99% of the variation in the satisfaction is explained by the E-Commerce.

#SLR for TechSup  
TechSup\_lm = lm(Satisfaction~TechSup, data = mydata)  
summary(TechSup\_lm)

##   
## Call:  
## lm(formula = Satisfaction ~ TechSup, data = mydata)  
##   
## 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 \*\*\*  
## TechSup 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

#PLot the linear model (line of best fit)  
qplot(TechSup,Satisfaction,data= mydata) + stat\_smooth(method="lm", col="red")



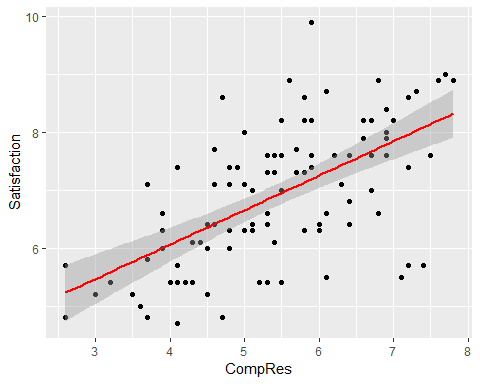
Observation

The estimated regression line equation, Satisfation = 6.44757 + 0.08768 \* TechSup. This means, an additional rating of 1 in Technical Support will raise the Satisfaction by 6.54. (i.e) Satifaction = 6.44757 + 0.08768 \* 1 = 6.53525. The p-value of 0.2647 > 0.05 is not highly significant. The R2 is 0.01268 which is around 1.27%. It implies that 1.27% of the variation in the satisfaction is explained by the Technical Support.

#SLR for CompRes  
CompRes\_lm = lm(Satisfaction~CompRes, data = mydata)  
summary(CompRes\_lm)

##   
## Call:  
## lm(formula = Satisfaction ~ CompRes, data = mydata)  
##   
## 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 \*\*\*  
## CompRes 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

#PLot the linear model (line of best fit)  
qplot(CompRes,Satisfaction,data= mydata) + stat\_smooth(method="lm", col="red")



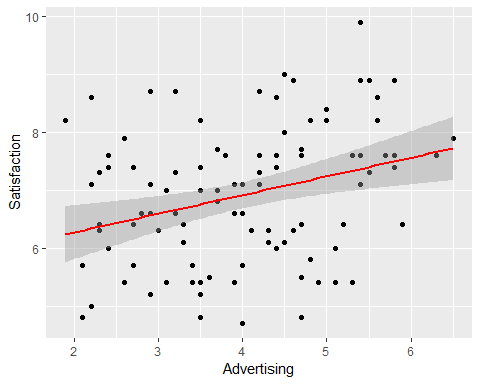
Observation

The estimated regression line equation, Satisfation = 3.68005 + 0.59499 \* CompRes. This means, an additional rating of 1 in Complaint Resolution will raise the Satisfaction by 4.28. (i.e) Satifaction = 3.68005 + 0.59499 \* 1 = 4.27504. The p-value of 3.085e-11 < 0.05 is highly significant. The R2 is 0.3639 which is around 36.39%. It implies that 36.39% of the variation in the satisfaction is explained by the Complaint Resolution.

#SLR for Advertising  
Advertising\_lm = lm(Satisfaction~Advertising, data = mydata)  
summary(Advertising\_lm)

##   
## Call:  
## lm(formula = Satisfaction ~ Advertising, data = mydata)  
##   
## 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

#PLot the linear model (line of best fit)  
qplot(Advertising,Satisfaction,data= mydata) + stat\_smooth(method="lm", col="red")



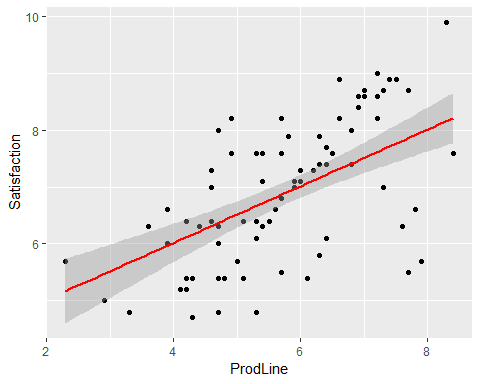
Observation

The estimated regression line equation, Satisfation = 5.6259 + 0.3222 \* Advertising. This means, an additional rating of 1 in Advertising will raise the Satisfaction by 5.95. (i.e) Satifaction = 5.6259 + 0.3222 \* 1 = 5.9481. The p-value of 0.002056 < 0.05 is highly significant. The R2 is 0.09282 which is around 9.28%. It implies that 9.28% of the variation in the satisfaction is explained by the Advertising.

#SLR for ProdLine  
ProdLine\_lm = lm(Satisfaction~ProdLine, data = mydata)  
summary(ProdLine\_lm)

##   
## Call:  
## lm(formula = Satisfaction ~ ProdLine, data = mydata)  
##   
## 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 \*\*\*  
## ProdLine 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

#PLot the linear model (line of best fit)  
qplot(ProdLine,Satisfaction,data= mydata) + stat\_smooth(method="lm", col="red")



Observation

The estimated regression line equation, Satisfation = 4.02203 + 0.49887 \* ProdLine. This means, an additional rating of 1 in Product Line will raise the Satisfaction by 4.52. (i.e) Satifaction = 4.02203 + 0.49887 \* 1 = 4.5209. The p-value of 2.953e-09 < 0.05 is highly significant. The R2 is 0.3031 which is around 30.31%. It implies that 30.31% of the variation in the satisfaction is explained by the Product Line.

#SLR for SalesFImage  
SalesFImage\_lm = lm(Satisfaction~SalesFImage, data = mydata)  
summary(SalesFImage\_lm)

##   
## Call:  
## lm(formula = Satisfaction ~ SalesFImage, data = mydata)  
##   
## 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 \*\*\*  
## SalesFImage 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

#PLot the linear model (line of best fit)  
qplot(SalesFImage,Satisfaction,data= mydata) + stat\_smooth(method="lm", col="red")



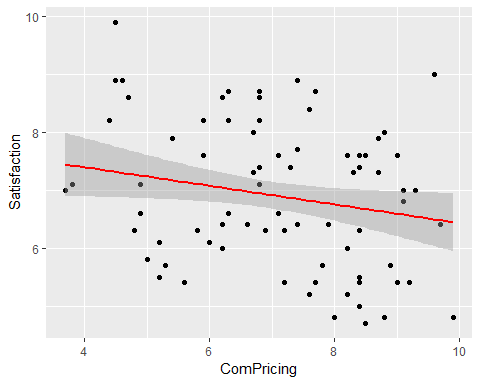
Observation

The estimated regression line equation, Satisfation = 4.06983 + 0.55596 \* SalesFImage. This means, an additional rating of 1 in SalesForce Image will raise the Satisfaction by 4.63. (i.e) Satifaction = 4.06983 + 0.55596 \* 1 = 4.62579. The p-value of 1.164e-07 < 0.05 is highly significant. The R2 is 0.2502 which is around 25.02%. It implies that 25.02% of the variation in the satisfaction is explained by the SalesForce Image.

#SLR for ComPricing  
ComPricing\_lm = lm(Satisfaction~ComPricing, data = mydata)  
summary(ComPricing\_lm)

##   
## Call:  
## lm(formula = Satisfaction ~ ComPricing, data = mydata)  
##   
## 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 \*\*\*  
## ComPricing -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

#PLot the linear model (line of best fit)  
qplot(ComPricing,Satisfaction,data= mydata) + stat\_smooth(method="lm", col="red")



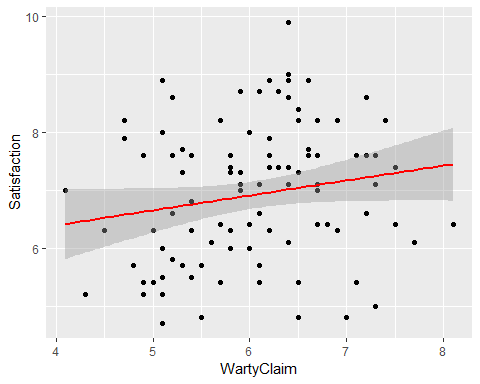
Observation

The estimated regression line equation, Satisfation = 8.03856 + (-0.16068) \* ComPricing. This means, an additional rating of 1 in Competitive Pricing will raise the Satisfaction by 7.88. (i.e) Satifaction = 8.03856 + (-0.16068) \* 1 = 7.87788. The p-value of 0.03756 < 0.05 is highly significant. The R2 is 0.04339 which is around 4.34%. It implies that 4.34% of the variation in the satisfaction is explained by the Competitive Pricing.

#SLR for WartyClaim  
WartyClaim\_lm = lm(Satisfaction~WartyClaim, data = mydata)  
summary(WartyClaim\_lm)

##   
## Call:  
## lm(formula = Satisfaction ~ WartyClaim, data = mydata)  
##   
## 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 \*\*\*  
## WartyClaim 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

#PLot the linear model (line of best fit)  
qplot(WartyClaim,Satisfaction,data= mydata) + stat\_smooth(method="lm", col="red")



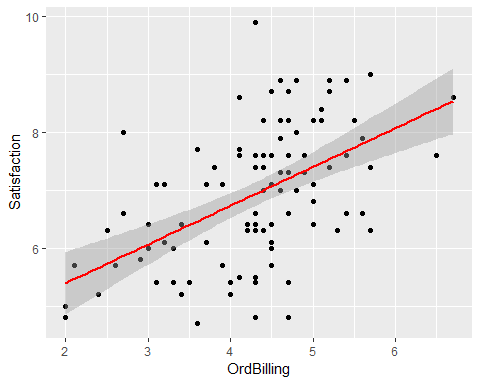
Observation

The estimated regression line equation, Satisfation = 5.3581 + 0.2581 \* WartyClaim This means, an additional rating of 1 in Warranty & Claims will raise the Satisfaction by 5.62. (i.e) Satifaction = 5.3581 + 0.2581 \* 1 = 5.6162. The p-value of 0.0772 > 0.05 is not highly significant. The R2 is 0.03152 which is around 3.15%. It implies that 3.15% of the variation in the satisfaction is explained by the Warranty & Claims.

#SLR for OrdBilling  
OrdBilling\_lm = lm(Satisfaction~OrdBilling, data = mydata)  
summary(OrdBilling\_lm)

##   
## Call:  
## lm(formula = Satisfaction ~ OrdBilling, data = mydata)  
##   
## 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 \*\*\*  
## OrdBilling 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

#PLot the linear model (line of best fit)  
qplot(OrdBilling,Satisfaction,data= mydata) + stat\_smooth(method="lm", col="red")



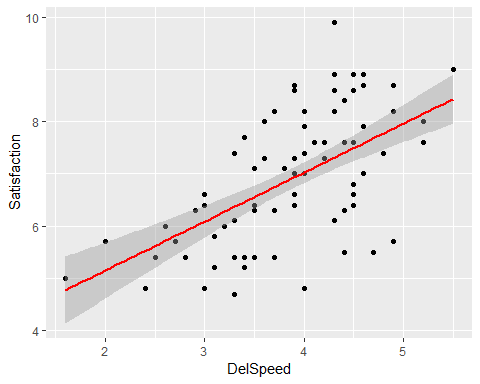
Observation

The estimated regression line equation, Satisfation = 4.0541 + 0.6695 \* OrdBilling This means, an additional rating of 1 in Order & Billing will raise the Satisfaction by 5.62. (i.e) Satifaction = 4.0541 + 0.6695 \* 1 = 5.6162. The p-value of 2.602e-08 < 0.05 is highly significant. The R2 is 0.2722 which is around 27.22%. It implies that 27.22% of the variation in the satisfaction is explained by the Order & Billing.

#SLR for DelSpeed  
DelSpeed\_lm = lm(Satisfaction~DelSpeed, data = mydata)  
summary(DelSpeed\_lm)

##   
## Call:  
## lm(formula = Satisfaction ~ DelSpeed, data = mydata)  
##   
## 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 \*\*\*  
## DelSpeed 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

#PLot the linear model (line of best fit)  
qplot(DelSpeed,Satisfaction,data= mydata) + stat\_smooth(method="lm", col="red")



Observation

The estimated regression line equation, Satisfation = 3.2791 + 0.9364 \* DelSpeed. This means, an additional rating of 1 in Delivery Speed will raise the Satisfaction by 4.22. (i.e) Satifaction = 3.2791 + 0.9364 \* 1 = 4.2155,,, The p-value of 3.3e-10 < 0.05 is highly significant. The R2 is 0.333 which is around 33.3%. It implies that 33.3% of the variation in the satisfaction is explained by the Delivery Speed.

# Performing Multiple Linear Regression

full\_model = lm(Satisfaction~., data = mydata)  
summary(full\_model)

##   
## Call:  
## lm(formula = Satisfaction ~ ., data = mydata)  
##   
## 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   
## ProdQual 0.37137 0.05177 7.173 2.18e-10 \*\*\*  
## Ecom -0.44056 0.13396 -3.289 0.00145 \*\*   
## TechSup 0.03299 0.06372 0.518 0.60591   
## CompRes 0.16703 0.10173 1.642 0.10416   
## Advertising -0.02602 0.06161 -0.422 0.67382   
## ProdLine 0.14034 0.08025 1.749 0.08384 .   
## SalesFImage 0.80611 0.09775 8.247 1.45e-12 \*\*\*  
## ComPricing -0.03853 0.04677 -0.824 0.41235   
## WartyClaim -0.10298 0.12330 -0.835 0.40587   
## OrdBilling 0.14635 0.10367 1.412 0.16160   
## DelSpeed 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

#Variable selection method  
forward\_model = stepAIC(lm(Satisfaction~., data = mydata),direction = "forward")

## Start: AIC=-103.91  
## Satisfaction ~ ProdQual + Ecom + TechSup + CompRes + Advertising +   
## ProdLine + SalesFImage + ComPricing + WartyClaim + OrdBilling +   
## DelSpeed

backward\_model = stepAIC(lm(Satisfaction~., data = mydata),direction = "backward")

## Start: AIC=-103.91  
## Satisfaction ~ ProdQual + Ecom + TechSup + CompRes + Advertising +   
## ProdLine + SalesFImage + ComPricing + WartyClaim + OrdBilling +   
## DelSpeed  
##   
## Df Sum of Sq RSS AIC  
## - Advertising 1 0.0564 27.885 -105.707  
## - TechSup 1 0.0848 27.914 -105.606  
## - ComPricing 1 0.2145 28.043 -105.142  
## - WartyClaim 1 0.2206 28.049 -105.120  
## - DelSpeed 1 0.2250 28.054 -105.104  
## <none> 27.829 -103.910  
## - OrdBilling 1 0.6301 28.459 -103.671  
## - CompRes 1 0.8526 28.681 -102.892  
## - ProdLine 1 0.9670 28.796 -102.494  
## - Ecom 1 3.4205 31.249 -94.317  
## - ProdQual 1 16.2727 44.101 -59.868  
## - SalesFImage 1 21.5072 49.336 -48.652  
##   
## Step: AIC=-105.71  
## Satisfaction ~ ProdQual + Ecom + TechSup + CompRes + ProdLine +   
## SalesFImage + ComPricing + WartyClaim + OrdBilling + DelSpeed  
##   
## Df Sum of Sq RSS AIC  
## - TechSup 1 0.0939 27.979 -107.371  
## - DelSpeed 1 0.1896 28.075 -107.030  
## - ComPricing 1 0.2023 28.088 -106.984  
## - WartyClaim 1 0.2280 28.113 -106.893  
## <none> 27.885 -105.707  
## - OrdBilling 1 0.6463 28.532 -105.416  
## - CompRes 1 0.8901 28.775 -104.565  
## - ProdLine 1 1.0563 28.942 -103.989  
## - Ecom 1 3.4349 31.320 -96.091  
## - ProdQual 1 16.2547 44.140 -61.781  
## - SalesFImage 1 23.0550 50.940 -47.452  
##   
## Step: AIC=-107.37  
## Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine + SalesFImage +   
## ComPricing + WartyClaim + OrdBilling + DelSpeed  
##   
## Df Sum of Sq RSS AIC  
## - WartyClaim 1 0.1467 28.126 -108.848  
## - DelSpeed 1 0.1818 28.161 -108.724  
## - ComPricing 1 0.2459 28.225 -108.496  
## <none> 27.979 -107.371  
## - OrdBilling 1 0.5855 28.565 -107.300  
## - ProdLine 1 1.0003 28.979 -105.858  
## - CompRes 1 1.0119 28.991 -105.818  
## - Ecom 1 3.3742 31.353 -97.985  
## - ProdQual 1 16.3758 44.355 -63.295  
## - SalesFImage 1 22.9921 50.971 -49.391  
##   
## Step: AIC=-108.85  
## Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine + SalesFImage +   
## ComPricing + OrdBilling + DelSpeed  
##   
## Df Sum of Sq RSS AIC  
## - ComPricing 1 0.2021 28.328 -110.132  
## - DelSpeed 1 0.2606 28.386 -109.926  
## - OrdBilling 1 0.4919 28.618 -109.114  
## <none> 28.126 -108.848  
## - ProdLine 1 0.8706 28.996 -107.800  
## - CompRes 1 1.0162 29.142 -107.299  
## - Ecom 1 3.2874 31.413 -99.794  
## - ProdQual 1 17.1823 45.308 -63.168  
## - SalesFImage 1 23.1243 51.250 -50.845  
##   
## Step: AIC=-110.13  
## Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine + SalesFImage +   
## OrdBilling + DelSpeed  
##   
## Df Sum of Sq RSS AIC  
## - DelSpeed 1 0.1801 28.508 -111.498  
## - OrdBilling 1 0.5702 28.898 -110.139  
## <none> 28.328 -110.132  
## - CompRes 1 1.0523 29.380 -108.485  
## - ProdLine 1 1.5397 29.868 -106.840  
## - Ecom 1 3.3731 31.701 -100.882  
## - ProdQual 1 17.6001 45.928 -63.809  
## - SalesFImage 1 22.9355 51.263 -52.819  
##   
## Step: AIC=-111.5  
## Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine + SalesFImage +   
## OrdBilling  
##   
## Df Sum of Sq RSS AIC  
## <none> 28.508 -111.498  
## - OrdBilling 1 0.9140 29.422 -110.342  
## - CompRes 1 2.3970 30.905 -105.425  
## - ProdLine 1 2.6247 31.133 -104.691  
## - Ecom 1 3.3996 31.908 -102.232  
## - ProdQual 1 18.6802 47.188 -63.102  
## - SalesFImage 1 24.1101 52.618 -52.211

both\_model = stepAIC(lm(Satisfaction~., data = mydata),direction = "both")

## Start: AIC=-103.91  
## Satisfaction ~ ProdQual + Ecom + TechSup + CompRes + Advertising +   
## ProdLine + SalesFImage + ComPricing + WartyClaim + OrdBilling +   
## DelSpeed  
##   
## Df Sum of Sq RSS AIC  
## - Advertising 1 0.0564 27.885 -105.707  
## - TechSup 1 0.0848 27.914 -105.606  
## - ComPricing 1 0.2145 28.043 -105.142  
## - WartyClaim 1 0.2206 28.049 -105.120  
## - DelSpeed 1 0.2250 28.054 -105.104  
## <none> 27.829 -103.910  
## - OrdBilling 1 0.6301 28.459 -103.671  
## - CompRes 1 0.8526 28.681 -102.892  
## - ProdLine 1 0.9670 28.796 -102.494  
## - Ecom 1 3.4205 31.249 -94.317  
## - ProdQual 1 16.2727 44.101 -59.868  
## - SalesFImage 1 21.5072 49.336 -48.652  
##   
## Step: AIC=-105.71  
## Satisfaction ~ ProdQual + Ecom + TechSup + CompRes + ProdLine +   
## SalesFImage + ComPricing + WartyClaim + OrdBilling + DelSpeed  
##   
## Df Sum of Sq RSS AIC  
## - TechSup 1 0.0939 27.979 -107.371  
## - DelSpeed 1 0.1896 28.075 -107.030  
## - ComPricing 1 0.2023 28.088 -106.984  
## - WartyClaim 1 0.2280 28.113 -106.893  
## <none> 27.885 -105.707  
## - OrdBilling 1 0.6463 28.532 -105.416  
## - CompRes 1 0.8901 28.775 -104.565  
## - ProdLine 1 1.0563 28.942 -103.989  
## + Advertising 1 0.0564 27.829 -103.910  
## - Ecom 1 3.4349 31.320 -96.091  
## - ProdQual 1 16.2547 44.140 -61.781  
## - SalesFImage 1 23.0550 50.940 -47.452  
##   
## Step: AIC=-107.37  
## Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine + SalesFImage +   
## ComPricing + WartyClaim + OrdBilling + DelSpeed  
##   
## Df Sum of Sq RSS AIC  
## - WartyClaim 1 0.1467 28.126 -108.848  
## - DelSpeed 1 0.1818 28.161 -108.724  
## - ComPricing 1 0.2459 28.225 -108.496  
## <none> 27.979 -107.371  
## - OrdBilling 1 0.5855 28.565 -107.300  
## - ProdLine 1 1.0003 28.979 -105.858  
## - CompRes 1 1.0119 28.991 -105.818  
## + TechSup 1 0.0939 27.885 -105.707  
## + Advertising 1 0.0655 27.914 -105.606  
## - Ecom 1 3.3742 31.353 -97.985  
## - ProdQual 1 16.3758 44.355 -63.295  
## - SalesFImage 1 22.9921 50.971 -49.391  
##   
## Step: AIC=-108.85  
## Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine + SalesFImage +   
## ComPricing + OrdBilling + DelSpeed  
##   
## Df Sum of Sq RSS AIC  
## - ComPricing 1 0.2021 28.328 -110.132  
## - DelSpeed 1 0.2606 28.386 -109.926  
## - OrdBilling 1 0.4919 28.618 -109.114  
## <none> 28.126 -108.848  
## - ProdLine 1 0.8706 28.996 -107.800  
## + WartyClaim 1 0.1467 27.979 -107.371  
## - CompRes 1 1.0162 29.142 -107.299  
## + Advertising 1 0.0602 28.066 -107.062  
## + TechSup 1 0.0126 28.113 -106.893  
## - Ecom 1 3.2874 31.413 -99.794  
## - ProdQual 1 17.1823 45.308 -63.168  
## - SalesFImage 1 23.1243 51.250 -50.845  
##   
## Step: AIC=-110.13  
## Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine + SalesFImage +   
## OrdBilling + DelSpeed  
##   
## Df Sum of Sq RSS AIC  
## - DelSpeed 1 0.1801 28.508 -111.498  
## - OrdBilling 1 0.5702 28.898 -110.139  
## <none> 28.328 -110.132  
## + ComPricing 1 0.2021 28.126 -108.848  
## + WartyClaim 1 0.1029 28.225 -108.496  
## - CompRes 1 1.0523 29.380 -108.485  
## + Advertising 1 0.0498 28.278 -108.308  
## + TechSup 1 0.0007 28.327 -108.135  
## - ProdLine 1 1.5397 29.868 -106.840  
## - Ecom 1 3.3731 31.701 -100.882  
## - ProdQual 1 17.6001 45.928 -63.809  
## - SalesFImage 1 22.9355 51.263 -52.819  
##   
## Step: AIC=-111.5  
## Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine + SalesFImage +   
## OrdBilling  
##   
## Df Sum of Sq RSS AIC  
## <none> 28.508 -111.498  
## - OrdBilling 1 0.9140 29.422 -110.342  
## + DelSpeed 1 0.1801 28.328 -110.132  
## + WartyClaim 1 0.1662 28.342 -110.083  
## + ComPricing 1 0.1217 28.386 -109.926  
## + Advertising 1 0.0171 28.491 -109.558  
## + TechSup 1 0.0135 28.495 -109.546  
## - CompRes 1 2.3970 30.905 -105.425  
## - ProdLine 1 2.6247 31.133 -104.691  
## - Ecom 1 3.3996 31.908 -102.232  
## - ProdQual 1 18.6802 47.188 -63.102  
## - SalesFImage 1 24.1101 52.618 -52.211

summary(forward\_model) # Same as full model

##   
## Call:  
## lm(formula = Satisfaction ~ ProdQual + Ecom + TechSup + CompRes +   
## Advertising + ProdLine + SalesFImage + ComPricing + WartyClaim +   
## OrdBilling + DelSpeed, data = mydata)  
##   
## 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   
## ProdQual 0.37137 0.05177 7.173 2.18e-10 \*\*\*  
## Ecom -0.44056 0.13396 -3.289 0.00145 \*\*   
## TechSup 0.03299 0.06372 0.518 0.60591   
## CompRes 0.16703 0.10173 1.642 0.10416   
## Advertising -0.02602 0.06161 -0.422 0.67382   
## ProdLine 0.14034 0.08025 1.749 0.08384 .   
## SalesFImage 0.80611 0.09775 8.247 1.45e-12 \*\*\*  
## ComPricing -0.03853 0.04677 -0.824 0.41235   
## WartyClaim -0.10298 0.12330 -0.835 0.40587   
## OrdBilling 0.14635 0.10367 1.412 0.16160   
## DelSpeed 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

forward\_model$anova # Lowest AIC = -103.9097

## Stepwise Model Path   
## Analysis of Deviance Table  
##   
## Initial Model:  
## Satisfaction ~ ProdQual + Ecom + TechSup + CompRes + Advertising +   
## ProdLine + SalesFImage + ComPricing + WartyClaim + OrdBilling +   
## DelSpeed  
##   
## Final Model:  
## Satisfaction ~ ProdQual + Ecom + TechSup + CompRes + Advertising +   
## ProdLine + SalesFImage + ComPricing + WartyClaim + OrdBilling +   
## DelSpeed  
##   
##   
## Step Df Deviance Resid. Df Resid. Dev AIC  
## 1 88 27.82884 -103.9097

summary(backward\_model) # ProdQual + Ecom + CompRes + ProdLine + SalesFImage + OrdBilling

##   
## Call:  
## lm(formula = Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine +   
## SalesFImage + OrdBilling, data = mydata)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.40616 -0.32428 0.03067 0.38672 0.97362   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.26902 0.49935 -2.541 0.01270 \*   
## ProdQual 0.36499 0.04676 7.806 8.59e-12 \*\*\*  
## Ecom -0.43635 0.13103 -3.330 0.00125 \*\*   
## CompRes 0.22577 0.08074 2.796 0.00628 \*\*   
## ProdLine 0.17655 0.06034 2.926 0.00431 \*\*   
## SalesFImage 0.78167 0.08814 8.869 5.06e-14 \*\*\*  
## OrdBilling 0.15911 0.09215 1.727 0.08753 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5537 on 93 degrees of freedom  
## Multiple R-squared: 0.7973, Adjusted R-squared: 0.7842   
## F-statistic: 60.96 on 6 and 93 DF, p-value: < 2.2e-16

backward\_model$anova # Lowest AIC = -111.4983

## Stepwise Model Path   
## Analysis of Deviance Table  
##   
## Initial Model:  
## Satisfaction ~ ProdQual + Ecom + TechSup + CompRes + Advertising +   
## ProdLine + SalesFImage + ComPricing + WartyClaim + OrdBilling +   
## DelSpeed  
##   
## Final Model:  
## Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine + SalesFImage +   
## OrdBilling  
##   
##   
## Step Df Deviance Resid. Df Resid. Dev AIC  
## 1 88 27.82884 -103.9097  
## 2 - Advertising 1 0.05640123 89 27.88524 -105.7073  
## 3 - TechSup 1 0.09392498 90 27.97917 -107.3710  
## 4 - WartyClaim 1 0.14665583 91 28.12582 -108.8482  
## 5 - ComPricing 1 0.20213586 92 28.32796 -110.1321  
## 6 - DelSpeed 1 0.18012574 93 28.50808 -111.4983

summary(both\_model) #ProdQual + Ecom + CompRes + ProdLine + SalesFImage + OrdBilling

##   
## Call:  
## lm(formula = Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine +   
## SalesFImage + OrdBilling, data = mydata)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.40616 -0.32428 0.03067 0.38672 0.97362   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.26902 0.49935 -2.541 0.01270 \*   
## ProdQual 0.36499 0.04676 7.806 8.59e-12 \*\*\*  
## Ecom -0.43635 0.13103 -3.330 0.00125 \*\*   
## CompRes 0.22577 0.08074 2.796 0.00628 \*\*   
## ProdLine 0.17655 0.06034 2.926 0.00431 \*\*   
## SalesFImage 0.78167 0.08814 8.869 5.06e-14 \*\*\*  
## OrdBilling 0.15911 0.09215 1.727 0.08753 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5537 on 93 degrees of freedom  
## Multiple R-squared: 0.7973, Adjusted R-squared: 0.7842   
## F-statistic: 60.96 on 6 and 93 DF, p-value: < 2.2e-16

both\_model$anova # Lowest AIC = -111.4983

## Stepwise Model Path   
## Analysis of Deviance Table  
##   
## Initial Model:  
## Satisfaction ~ ProdQual + Ecom + TechSup + CompRes + Advertising +   
## ProdLine + SalesFImage + ComPricing + WartyClaim + OrdBilling +   
## DelSpeed  
##   
## Final Model:  
## Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine + SalesFImage +   
## OrdBilling  
##   
##   
## Step Df Deviance Resid. Df Resid. Dev AIC  
## 1 88 27.82884 -103.9097  
## 2 - Advertising 1 0.05640123 89 27.88524 -105.7073  
## 3 - TechSup 1 0.09392498 90 27.97917 -107.3710  
## 4 - WartyClaim 1 0.14665583 91 28.12582 -108.8482  
## 5 - ComPricing 1 0.20213586 92 28.32796 -110.1321  
## 6 - DelSpeed 1 0.18012574 93 28.50808 -111.4983

Observation

The best variable selection method is the backward model and we will take the backward\_model and predict with it. The STEPWISE BACKWARD MODEL evaluates stepwise the variables that do not contribute significantly to the analysis and eliminates them. The process starts with the full model, the dependent variable (Satisfaction) will be compared with all the other variables using the function (lm), the variable which presents the highest p-value score should be eliminated, this process is repeated until no further improvement is possible.

This model has given the lowest AIC of -111.4983.

Predicting using best model and measure performance

# Split the observations into Training DataSet and Test Dataset of (70:30)  
set.seed(1)  
indices = sample(1:nrow(mydata), 0.7\*nrow(mydata))   
trainDS = mydata[indices,]   
testDS = mydata[-indices,]  
nrow(trainDS)

## [1] 70

nrow(testDS)

## [1] 30

#Train the best regression model(backward variables) with trained Dataset  
trained\_model = lm(Satisfaction~  
 ProdQual + Ecom + CompRes + ProdLine + SalesFImage + OrdBilling, trainDS)  
  
summary(trained\_model)

##   
## Call:  
## lm(formula = Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine +   
## SalesFImage + OrdBilling, data = trainDS)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.60683 -0.32849 0.07617 0.41057 0.83523   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.68357 0.62327 -2.701 0.008865 \*\*   
## ProdQual 0.41817 0.05849 7.149 1.10e-09 \*\*\*  
## Ecom -0.69583 0.17017 -4.089 0.000125 \*\*\*  
## CompRes 0.24254 0.10179 2.383 0.020211 \*   
## ProdLine 0.18724 0.07526 2.488 0.015512 \*   
## SalesFImage 1.01687 0.12334 8.244 1.33e-11 \*\*\*  
## OrdBilling 0.07922 0.11208 0.707 0.482310   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5665 on 63 degrees of freedom  
## Multiple R-squared: 0.8048, Adjusted R-squared: 0.7862   
## F-statistic: 43.28 on 6 and 63 DF, p-value: < 2.2e-16

#Rsquared for Train  
train\_R2\_Pref = summary(trained\_model)$r.squared  
  
#Prediction : Trained  
pred\_train\_pca = predict(trained\_model,trainDS)  
  
# PREDICT for TRAINS DATASET  
trainedPredSat = predict(trained\_model, trainDS, type="response")  
  
#TRAIN : Model Performance  
mse\_trained\_perf = mse(trainDS$Satisfaction,trainedPredSat)  
rmse\_trained\_perf = sqrt(mse\_trained\_perf)  
  
# PREDICT for TEST DATASET : Predict the Customer Satisfaction based on trained model for test dataset  
predSat = predict(trained\_model, testDS,type="response")  
  
#TEST : Model performance  
mse\_test\_perf = mse(testDS$Satisfaction,predSat)  
rmse\_test\_perf = sqrt(mse\_test\_perf)  
  
#Rsquared for Test  
cor(testDS$Satisfaction,predSat)^2

## [1] 0.8088671

Observation

|  |  |  |
| --- | --- | --- |
|  | Train | Test |
| R-SQRD | 0.8047717 | 0.8088671 |
| RMSE | 0.5373889 | 0.6215275 |

The R2 for Train and Test is 0.8. This shows that there is no improvement in the R2 value. The RMSE for Train is 0.54, whereas for Test is 0.62. This shows that the RMSE has increased for test. Hence, we can say that we need to improve the model furthur.

# Assumption Checks for MLR

1.Linearity:

Independent variables should not be correlated with each other i.e. No multi-Collinear. We need to understand if model has multi-collinearity, use VIF - Variable Inflation Factor

#VIF is greater than 2.5, remove those variable  
#else take all varibale within 2  
  
#VIF = 1/(1-r) for every variable becomes dependent  
#Full model  
vif(full\_model)

## ProdQual Ecom TechSup CompRes Advertising ProdLine   
## 1.635797 2.756694 2.976796 4.730448 1.508933 3.488185   
## SalesFImage ComPricing WartyClaim OrdBilling DelSpeed   
## 3.439420 1.635000 3.198337 2.902999 6.516014

Since multiple variables have a VIF score greater than 2.5, we can assume multicollinearity exists.

2. Independence of Error or Auto-Correlation: Durbin Watson Test

The Durbin Watson (DW) statistic is a test for autocorrelation in the residuals from a statistical regression analysis.

#If DW-Stat approach 0 : Residuals have POsitive Auto-corelation  
#If DW-Stat approch 2 : Residual have no auto-corelation  
#if DW-Stat greater than 2 : Negatively auto-corelated  
  
durbinWatsonTest(full\_model)

## lag Autocorrelation D-W Statistic p-value  
## 1 -0.170307 2.335774 0.096  
## Alternative hypothesis: rho != 0

#Null Hypothsis : Residuals are auto-corelated  
#Alternate : Residuals are not NOT auto-corelated

As we can see it is negatively correlated. It is having a p-value of 0.096 > 0.05.

3. Normality in Errors/Residuals: Shapiro Wilk test: BoxCox: Transformation

#Shapiro Wilk test: Before transformation: p > 0.05 is normal  
#ProdQual + Ecom + CompRes + ProdLine + SalesFImage + OrdBilling  
  
shapiro.test(Satisfaction) # normal

##   
## Shapiro-Wilk normality test  
##   
## data: Satisfaction  
## W = 0.97516, p-value = 0.05556

shapiro.test(ProdQual) # not normal

##   
## Shapiro-Wilk normality test  
##   
## data: ProdQual  
## W = 0.94972, p-value = 0.0007953

shapiro.test(Ecom) # not normal

##   
## Shapiro-Wilk normality test  
##   
## data: Ecom  
## W = 0.95852, p-value = 0.003157

shapiro.test(CompRes) # normal

##   
## Shapiro-Wilk normality test  
##   
## data: CompRes  
## W = 0.98646, p-value = 0.4023

shapiro.test(ProdLine) # normal

##   
## Shapiro-Wilk normality test  
##   
## data: ProdLine  
## W = 0.98692, p-value = 0.4324

shapiro.test(SalesFImage) # not normal

##   
## Shapiro-Wilk normality test  
##   
## data: SalesFImage  
## W = 0.97403, p-value = 0.04534

shapiro.test(OrdBilling) # not normal

##   
## Shapiro-Wilk normality test  
##   
## data: OrdBilling  
## W = 0.97405, p-value = 0.04549

As most of the variables are not normal, it is violating the normality.

# Common Box-Cox Transformations  
# Lambda value () Transformed data (Y')  
# -3.5 to -2.5 Y-3 = 1/Y3  
# -2.5 to -1.5 Y-2 = 1/Y2  
# -1.5 to 0.75 Y-1 = 1/Y1 (INverse)  
# -0.75 to -0.25 inverse sqrt =1/sqrt(y)  
# -0.25 to 0.25 natural log  
# 0.25 to 0.75 sqrt(Y)  
# 0.75 to 1.5 NOne (Y)  
# 1.5 to 2.5 Y2 = Y square  
# 2.5 > Y3  
  
install.packages("forecast")

## Warning: unable to access index for repository https://cloud.r-project.org/src/contrib:  
## cannot open URL 'https://cloud.r-project.org/src/contrib/PACKAGES'

## Warning: package 'forecast' is not available (for R version 3.5.3)

## Warning: unable to access index for repository https://cloud.r-project.org/bin/windows/contrib/3.5:  
## cannot open URL 'https://cloud.r-project.org/bin/windows/contrib/3.5/PACKAGES'

library(moments)  
library(forecast)

##   
## Attaching package: 'forecast'

## The following object is masked from 'package:Metrics':  
##   
## accuracy

#ProdQual + Ecom + CompRes + ProdLine + SalesFImage + OrdBilling  
lambda\_Sat = BoxCox.lambda(Satisfaction) #0.2587586 == sqrt(Y)  
lambda\_ProdQual=BoxCox.lambda(ProdQual) #1.999924 == Y square   
lambda\_Ecom=BoxCox.lambda(Ecom) #-0.02918471 == natural log  
lambda\_CompRes=BoxCox.lambda(CompRes) #0.8856065 == None (Y)  
lambda\_ProdLine=BoxCox.lambda(ProdLine) #0.6460062 == sqrt(Y)   
lambda\_SalesFImage=BoxCox.lambda(SalesFImage) #0.4146074 == sqrt(Y)  
lambda\_OrdBilling=BoxCox.lambda(OrdBilling) # 1.666167== Y square  
  
#model based on transformation model\_tr  
model\_tr = lm(sqrt(Satisfaction)~  
 (ProdQual)^2+  
 log(Ecom)+  
 CompRes +  
 sqrt(ProdLine)+  
 sqrt(SalesFImage)+  
 OrdBilling ^2, data = mydata)  
summary(model\_tr) ## increaed R2

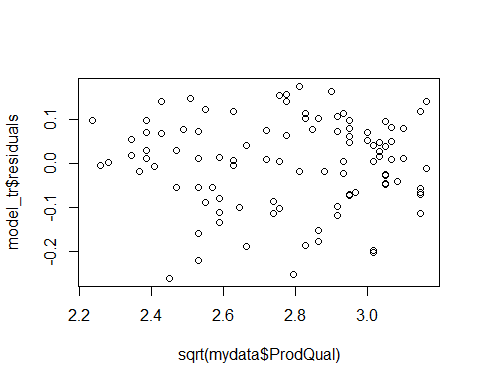
##   
## Call:  
## lm(formula = sqrt(Satisfaction) ~ (ProdQual)^2 + log(Ecom) +   
## CompRes + sqrt(ProdLine) + sqrt(SalesFImage) + OrdBilling^2,   
## data = mydata)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.26248 -0.06562 0.01016 0.07551 0.17541   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.202807 0.147389 1.376 0.172125   
## ProdQual 0.069700 0.008744 7.971 3.89e-12 \*\*\*  
## log(Ecom) -0.318690 0.089661 -3.554 0.000598 \*\*\*  
## CompRes 0.043866 0.015122 2.901 0.004644 \*\*   
## sqrt(ProdLine) 0.161505 0.053295 3.030 0.003163 \*\*   
## sqrt(SalesFImage) 0.680636 0.073280 9.288 6.56e-15 \*\*\*  
## OrdBilling 0.029172 0.017287 1.687 0.094865 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1038 on 93 degrees of freedom  
## Multiple R-squared: 0.8052, Adjusted R-squared: 0.7927   
## F-statistic: 64.08 on 6 and 93 DF, p-value: < 2.2e-16

summary(trained\_model) ## based on best variable selection (backward)

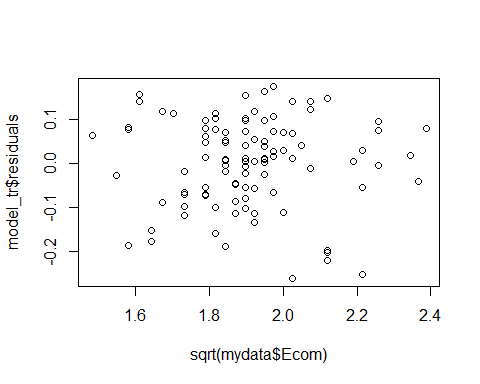
##   
## Call:  
## lm(formula = Satisfaction ~ ProdQual + Ecom + CompRes + ProdLine +   
## SalesFImage + OrdBilling, data = trainDS)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.60683 -0.32849 0.07617 0.41057 0.83523   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.68357 0.62327 -2.701 0.008865 \*\*   
## ProdQual 0.41817 0.05849 7.149 1.10e-09 \*\*\*  
## Ecom -0.69583 0.17017 -4.089 0.000125 \*\*\*  
## CompRes 0.24254 0.10179 2.383 0.020211 \*   
## ProdLine 0.18724 0.07526 2.488 0.015512 \*   
## SalesFImage 1.01687 0.12334 8.244 1.33e-11 \*\*\*  
## OrdBilling 0.07922 0.11208 0.707 0.482310   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5665 on 63 degrees of freedom  
## Multiple R-squared: 0.8048, Adjusted R-squared: 0.7862   
## F-statistic: 43.28 on 6 and 63 DF, p-value: < 2.2e-16

4. Equal Variance: Homoscadisticity

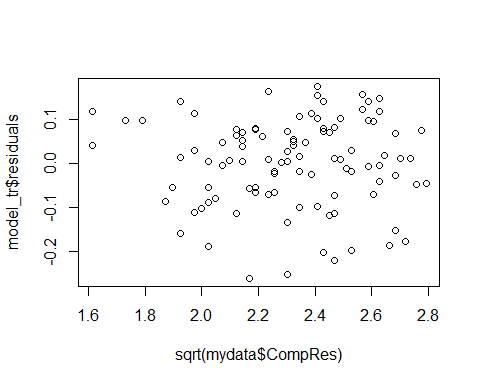
#Residual plot can without pattern suggest Equal Variance  
plot(sqrt(mydata$ProdQual),model\_tr$residuals)



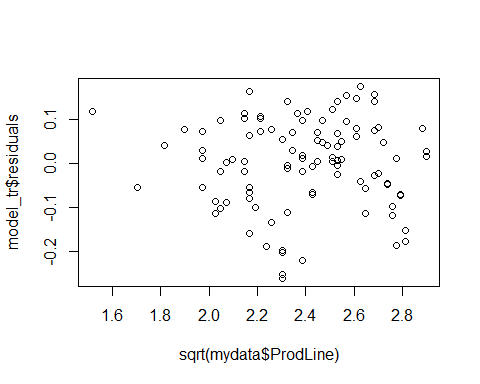
plot(sqrt(mydata$Ecom),model\_tr$residuals)



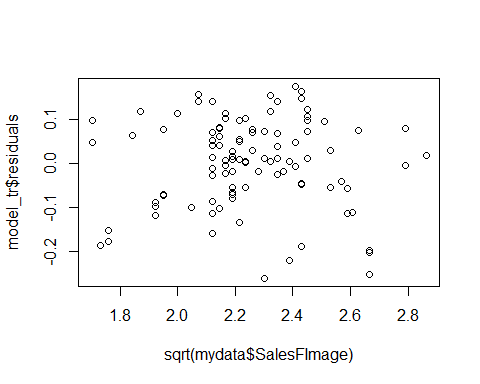
plot(sqrt(mydata$CompRes),model\_tr$residuals)



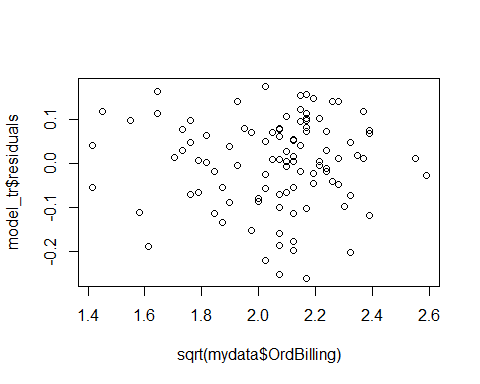
plot(sqrt(mydata$ProdLine),model\_tr$residuals)



plot(sqrt(mydata$SalesFImage),model\_tr$residuals)



plot(sqrt(mydata$OrdBilling),model\_tr$residuals)



# Assumption of Multi-colinearity

As the data shows signs of multi-colinearity, I have checked whether FA or PCA is required using KMO & bartlet test.

# Kaiser-Meyer-Olkin (KMO) Test is a measure of how suited your data is for Factor Analysis. KMO returns values between 0 and 1.  
  
#KMO Test :   
# 0.00 to 0.49 unacceptable.  
# 0.50 to 0.59 miserable.  
# 0.60 to 0.69 mediocre.  
# 0.70 to 0.79 middling.  
# 0.80 to 0.89 meritorious.  
# 0.90 to 1.00 marvelous.  
  
KMO(mydataCorr)

## Kaiser-Meyer-Olkin factor adequacy  
## Call: KMO(r = mydataCorr)  
## Overall MSA = 0.66  
## MSA for each item =   
## ProdQual Ecom TechSup CompRes Advertising   
## 0.49 0.59 0.52 0.83 0.83   
## ProdLine SalesFImage ComPricing WartyClaim OrdBilling   
## 0.70 0.52 0.77 0.52 0.79   
## DelSpeed Satisfaction   
## 0.72 0.66

Observation

The Overall MSA = 0.66, the degree of common variance in our dataset is rather “mediocre”. We would need to do reduction.

#Bartlett’s test for Sphericity compares the observed correlation matrix to the identity matrix. In other words, it checks if there is a redundancy between the variables that can be summarized with a few number of factors.  
  
cortest.bartlett(mydataCorr,n = nrow(mydata))

## $chisq  
## [1] 769.6422  
##   
## $p.value  
## [1] 1.65971e-120  
##   
## $df  
## [1] 66

Observation

We reject the null hypothesis at the 5% level (p-value = 1.65971e-120 < 0.05). We can perform efficiently a PCA on our dataset.

Removing the dependent Variable Customer Satisfaction.

#mydata$Satisfaction = NULL

# Performing PCA

The eigenvalue is a measure of how much of the variance of the observed variables a factor explains. Any factor with an eigenvalue ≥1 explains more variance than a single observed variable.

#Eigen value computation  
PCACorr = cor(mydata[,-12])  
ev = eigen(PCACorr) #PCACorr has the correlation value for mydata  
print(ev,digits=5)

## eigen() decomposition  
## $values  
## [1] 3.426971 2.550897 1.690976 1.086556 0.609424 0.551884 0.401518  
## [8] 0.246952 0.203553 0.132842 0.098427  
##   
## $vectors  
## [,1] [,2] [,3] [,4] [,5] [,6] [,7]  
## [1,] -0.13379 0.313498 0.062272 0.64314 0.231666 0.564570 -0.1916413  
## [2,] -0.16595 -0.446509 -0.235248 0.27238 0.422288 -0.263257 -0.0596262  
## [3,] -0.15769 0.230967 -0.610951 -0.19339 -0.023957 0.108769 0.0171999  
## [4,] -0.47068 -0.019444 0.210351 -0.20632 0.028657 0.028152 0.0084996  
## [5,] -0.18373 -0.363665 -0.088097 0.31789 -0.803870 0.200569 0.0630696  
## [6,] -0.38677 0.284781 0.116279 0.20290 0.116674 -0.098195 0.6081476  
## [7,] -0.20367 -0.470696 -0.241342 0.22218 0.204373 -0.104972 -0.0014374  
## [8,] 0.15169 -0.413457 0.053045 -0.33354 0.248926 0.709736 0.3082489  
## [9,] -0.21293 0.191672 -0.598564 -0.18530 -0.032927 0.139840 0.0306402  
## [10,] -0.43722 -0.026399 0.168930 -0.23685 0.026754 0.119480 -0.6593199  
## [11,] -0.47309 -0.073052 0.232625 -0.19733 -0.035433 -0.029800 0.2342393  
## [,8] [,9] [,10] [,11]  
## [1,] 0.135473 0.031328 -0.066597 -0.182792  
## [2,] -0.122026 -0.542511 -0.281558 -0.062339  
## [3,] 0.464710 -0.359300 0.388171 0.051930  
## [4,] 0.513398 0.093248 -0.534672 0.362534  
## [5,] -0.053477 -0.154682 -0.037158 0.081187  
## [6,] -0.333207 -0.084155 0.234798 0.385078  
## [7,] 0.169107 0.644899 0.353412 0.084699  
## [8,] -0.098832 -0.094144 0.045182 0.102958  
## [9,] -0.443540 0.317566 -0.435348 -0.128932  
## [10,] -0.366018 -0.099073 0.303865 0.194151  
## [11,] 0.065391 -0.021885 0.120104 -0.775632

EigenValue = ev$values  
EigenValue

## [1] 3.42697133 2.55089671 1.69097648 1.08655606 0.60942409 0.55188378  
## [7] 0.40151815 0.24695154 0.20355327 0.13284158 0.09842702

#Observaton : Based on Kaizer principal Eigen values > 1 can be considered as number of factors to consider. Here we are getting 4 factors.  
  
#Relative variance explained  
round((EigenValue/sum(EigenValue))\*100,2)

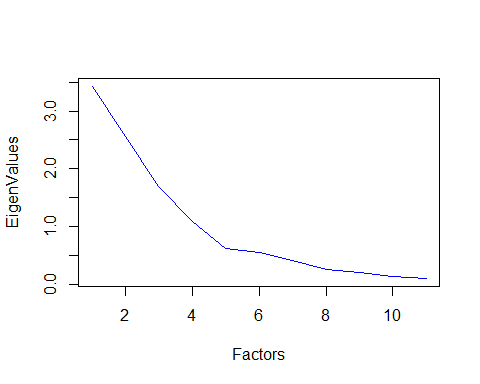
## [1] 31.15 23.19 15.37 9.88 5.54 5.02 3.65 2.25 1.85 1.21 0.89

#Cummulative variance explained  
cumsum(round((EigenValue/sum(EigenValue))\*100,2))

## [1] 31.15 54.34 69.71 79.59 85.13 90.15 93.80 96.05 97.90 99.11  
## [11] 100.00

#Scree plot : Elbow Method  
plot(EigenValue,col="blue",type="line",xlab = "Factors",ylab="EigenValues") #suggest 4 factors

## Warning in plot.xy(xy, type, ...): plot type 'line' will be truncated to  
## first character



Finding number of Factors to consider

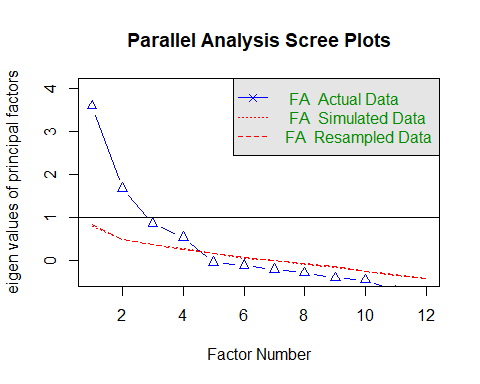
The blue line shows eigenvalues of actual data and the two red lines (placed on top of each other) show simulated and resampled data. Here we look at the large drops in the actual data and spot the point where it levels off to the right.

parallel = fa.parallel(mydata, fm = 'minres', fa = 'fa')

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =  
## rotate, : A loading greater than abs(1) was detected. Examine the loadings  
## carefully.

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs  
## = np.obs, : The estimated weights for the factor scores are probably  
## incorrect. Try a different factor extraction method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =  
## rotate, : An ultra-Heywood case was detected. Examine the results carefully



## Parallel analysis suggests that the number of factors = 4 and the number of components = NA

library(psych)  
str(mydata)

## 'data.frame': 100 obs. of 12 variables:  
## $ 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 ...

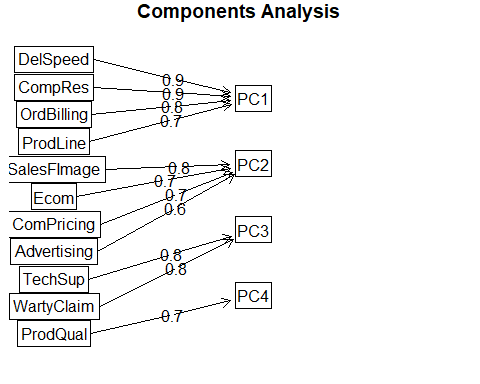
unrotated = principal(mydata[,-12], nfactors = 4, rotate = "none")   
unrotated

## Principal Components Analysis  
## Call: principal(r = mydata[, -12], nfactors = 4, rotate = "none")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## PC1 PC2 PC3 PC4 h2 u2 com  
## ProdQual 0.25 -0.50 -0.08 0.67 0.77 0.232 2.2  
## Ecom 0.31 0.71 0.31 0.28 0.78 0.223 2.1  
## TechSup 0.29 -0.37 0.79 -0.20 0.89 0.107 1.9  
## CompRes 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  
## ProdLine 0.72 -0.45 -0.15 0.21 0.79 0.213 2.0  
## SalesFImage 0.38 0.75 0.31 0.23 0.86 0.141 2.1  
## ComPricing -0.28 0.66 -0.07 -0.35 0.64 0.359 1.9  
## WartyClaim 0.39 -0.31 0.78 -0.19 0.89 0.108 2.0  
## OrdBilling 0.81 0.04 -0.22 -0.25 0.77 0.234 1.3  
## DelSpeed 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

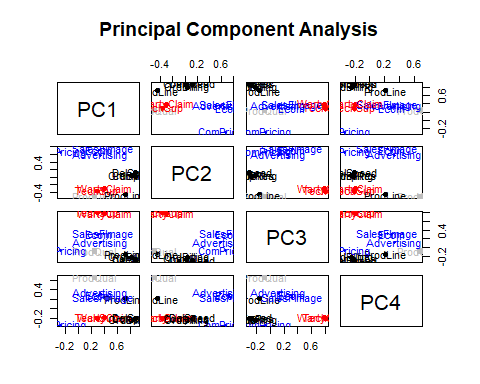
Observation

There are 4 factors, PC1, PC2, PC3 and PC4. The “SS loadings” row is the sum of squared loadings. This is used to determine the value of a particular factor. A factor is worth keeping if the SS loading is greater than 1. In this dataset, all are greater than 1. As we can see PC1 has the highest of 3.43 and the value reduces with every column. With PC4 having the lowest value of 1.09. The SS loading divided by the no of variables, gives us the proportion variation. PC1 has a variation of 31% and so on. The cumulatie variance adds the proportion variance, to give a total of 80%. In other words, 80% of the variation in the data is explained together by PC1, PC2, PC3 and PC4.

fa.diagram(unrotated)



unrotated\_profile = plot(unrotated,row.names(unrotated$loadings))



Orthogonal 90 degree

A VARIMAX rotation is a change of coordinates used in principal component analysis (PCA) that maximizes the sum of the variances of the squared loadings. Thus, all the coefficients (squared correlation with factors) will be either large or near zero, with few intermediate values.

rotated = principal(mydata[,-12], nfactors = 4, rotate = "varimax")   
  
rotated

## Principal Components Analysis  
## Call: principal(r = mydata[, -12], nfactors = 4, rotate = "varimax")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## RC1 RC2 RC3 RC4 h2 u2 com  
## ProdQual 0.00 -0.01 -0.03 0.88 0.77 0.232 1.0  
## Ecom 0.06 0.87 0.05 -0.12 0.78 0.223 1.1  
## TechSup 0.02 -0.02 0.94 0.10 0.89 0.107 1.0  
## CompRes 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  
## ProdLine 0.59 -0.06 0.15 0.64 0.79 0.213 2.1  
## SalesFImage 0.13 0.90 0.08 -0.16 0.86 0.141 1.1  
## ComPricing -0.09 0.23 -0.25 -0.72 0.64 0.359 1.5  
## WartyClaim 0.11 0.05 0.93 0.10 0.89 0.108 1.1  
## OrdBilling 0.86 0.11 0.08 0.04 0.77 0.234 1.1  
## DelSpeed 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

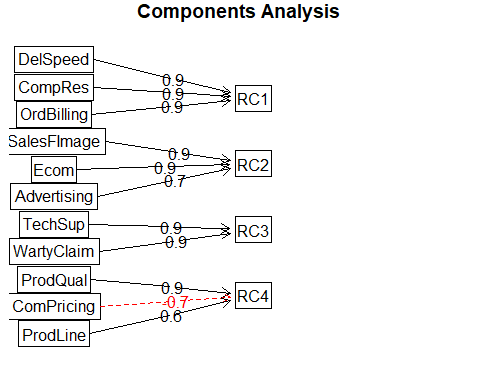
rotated$loadings

##   
## Loadings:  
## RC1 RC2 RC3 RC4   
## ProdQual 0.876  
## Ecom 0.871 -0.117  
## TechSup 0.939 0.101  
## CompRes 0.926 0.116   
## Advertising 0.139 0.742   
## ProdLine 0.591 0.146 0.642  
## SalesFImage 0.133 0.900 -0.159  
## ComPricing 0.226 -0.246 -0.723  
## WartyClaim 0.110 0.931 0.102  
## OrdBilling 0.864 0.107   
## DelSpeed 0.938 0.177   
##   
## RC1 RC2 RC3 RC4  
## SS loadings 2.893 2.234 1.856 1.774  
## Proportion Var 0.263 0.203 0.169 0.161  
## Cumulative Var 0.263 0.466 0.635 0.796

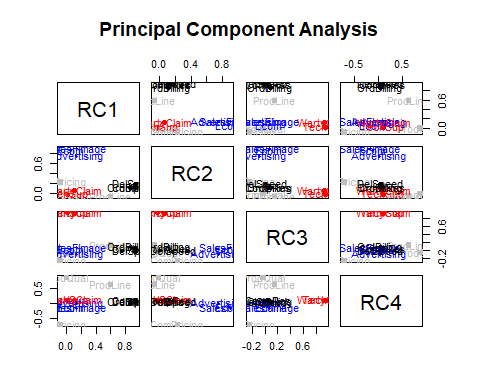
Observation

There are 4 factors, PC1, PC2, PC3 and PC4. In this dataset, all are greater than 1. As we can see PC1 has the highest of 2.893 and the value reduces with every column. With PC4 having the lowest value of 1.774. PC1 has a variation of 26.3% and so on. The cumulatie variance adds the proportion variance, to give a total of 79.6%. In other words, 80% of the variation in the data is explained together by PC1, PC2, PC3 and PC4.

# EIgen values has changed, more clear loading balancing, communality remains same  
fa.diagram(rotated)



rotated\_profile = plot(rotated,row.names(rotated$loadings),cex = 1.0)



# Giving profiled names to the factors  
# RC1 = Sales\_Distribution

# RC2 = Marketing

# RC3 = After\_Sales\_Service

# RC4 = Value\_For\_Money  
  
# Scores are the new dataset for regression again  
dim(rotated$scores)

## [1] 100 4

dim(mydata)

## [1] 100 12

## Create a new data.structure using scores for four factors and Target variable  
mydata\_PCA = cbind(mydata[,12],rotated$scores)  
mydata\_PCA = as.data.frame(mydata\_PCA)  
View(mydata\_PCA)  
names(mydata\_PCA) = c("Satisfaction","Sales\_Distribution","Marketing","After\_Sales\_Service","Value\_For\_Money")  
str(mydata\_PCA)

## 'data.frame': 100 obs. of 5 variables:  
## $ Satisfaction : num 8.2 5.7 8.9 4.8 7.1 4.7 5.7 6.3 7 5.5 ...  
## $ Sales\_Distribution : num 0.127 1.222 0.616 -0.845 -0.32 ...  
## $ Marketing : num 0.77 -1.646 0.58 -0.272 -0.834 ...  
## $ After\_Sales\_Service: num -1.87845 -0.61403 0.00369 1.26749 -0.0081 ...  
## $ Value\_For\_Money : num 0.366 0.813 1.57 -1.254 0.448 ...

# Lets again do MLR with new dataset

#Model buiding  
set.seed(1)  
indices = sample(1:nrow(mydata\_PCA), 0.7\*nrow(mydata\_PCA))   
trainDS\_PCA = mydata\_PCA[indices,]   
testDS\_PCA = mydata\_PCA[-indices,]  
  
nrow(trainDS\_PCA)

## [1] 70

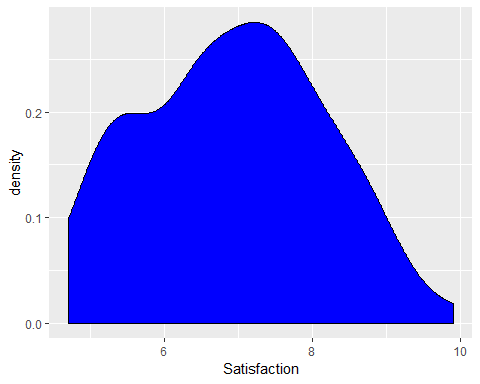
nrow(testDS\_PCA)

## [1] 30

names(trainDS\_PCA)

## [1] "Satisfaction" "Sales\_Distribution" "Marketing"   
## [4] "After\_Sales\_Service" "Value\_For\_Money"

##Explore the data  
ggplot(trainDS\_PCA, aes(Satisfaction)) + geom\_density(fill = "blue")



#mLet us make the model  
final\_model = lm(Satisfaction~.,trainDS\_PCA)  
summary(final\_model)

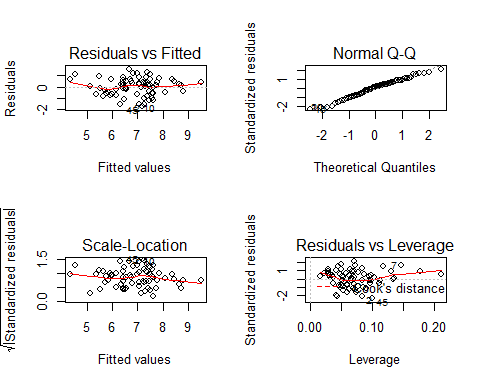
##   
## Call:  
## lm(formula = Satisfaction ~ ., data = trainDS\_PCA)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.6787 -0.5618 0.1229 0.5309 1.5261   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.92883 0.09319 74.351 < 2e-16 \*\*\*  
## Sales\_Distribution 0.60051 0.09789 6.134 5.68e-08 \*\*\*  
## Marketing 0.49472 0.09286 5.328 1.33e-06 \*\*\*  
## After\_Sales\_Service 0.03800 0.09133 0.416 0.679   
## Value\_For\_Money 0.54582 0.09367 5.827 1.92e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.776 on 65 degrees of freedom  
## Multiple R-squared: 0.622, Adjusted R-squared: 0.5987   
## F-statistic: 26.74 on 4 and 65 DF, p-value: 3.941e-13

Here F stat has a value of 26.74, which is greater than 1. Showing that there is a relationship between the predictors and response variable.

The p-value 3.941e-13 < 0.05, showing that the model is highly significant. From the summary, we can say that the After\_Sales\_Service is less significant, as the p-value is large for it. It may have to be removed, to get a better model.

The R2 closer to 1 indicates that the model explains the large value of the variance of the model and hence a good fit. In this case, the value is 0.622 (closer to 1) and hence the model is a good fit.

par(mfrow=c(2,2))  
plot(final\_model)



Fitted vs Residual graph

Residuals plots should be random in nature and there should not be any pattern in the graph. The average of the residual plot should be close to zero. From the above plot, we can see that the red trend line is almost near zero.

Normal Q-Q

Plot Q-Q plot shows whether the residuals are normally distributed. Ideally, the plot should be on the dotted line. If the Q-Q plot is not on the line, then models need to be reworked to make the residual normal. In the above plot, we see that most of the plots are on the line except towards the end.

Scale-Location

This shows how the residuals are spread and whether the residuals have an equal variance or not.

Residuals vs Leverage

The plot helps to find influential observations. Here we need to check for points that are outside the dashed line. A point outside the dashed line will be influential point and removal of that will affect the regression coefficients.

#Multicollinearity check  
vif(final\_model)

## Sales\_Distribution Marketing After\_Sales\_Service   
## 1.013825 1.026036 1.016930   
## Value\_For\_Money   
## 1.013152

Observation

As we can observe, all the variables have a value less than 2.5. We can safely assume that thereis no muticollinearity present in this model.

#Rsquared for Train  
train\_R2 = summary(final\_model)$r.squared  
  
#Prediction : Trained  
pred\_train\_pca = predict(final\_model,trainDS\_PCA)  
  
#trained performance  
mse\_per\_train\_pca = mse(trainDS\_PCA$Satisfaction,pred\_train\_pca)  
rmse\_per\_train\_pca = sqrt(mse\_per\_train\_pca)  
  
#Prediction : Test  
pred\_test\_pca = predict(final\_model,testDS\_PCA)  
  
#Test performance  
mse\_per\_test\_pca = mse(testDS\_PCA$Satisfaction,pred\_test\_pca)  
rmse\_per\_test\_pca = sqrt(mse\_per\_test\_pca)  
  
#Rsquared for Test  
cor(testDS\_PCA$Satisfaction,pred\_test\_pca)^2

## [1] 0.7679074

Observation

|  |  |  |
| --- | --- | --- |
|  | Trained | Tested |
| R-SQRD | 0.6219858 | 0.7679074 |
| RMSE | 0.747775 | 0.5400406 |

The R2 for Train is 0.62, whereas for Test is 0.77. This shows that there is an improvement in the R2 value. As it is closer to 1, we can say this is a best fit model. The RMSE for Train is 0.75, whereas for Test is 0.54. This shows that the RMSE has reduced for test. Hence, we can say this is again an indication of a best fir model. The VIF scores are below 2.5 for all the independent variables.

We can safely conclude that this is a best fit model.

# Conclusion

The product service market dataset had 13 variables. As customer satisfaction is the most important factor for any product. This was taken as a dependent variable. The data was tested to find the best fitting model. As it showed signs of multi-collinearity, PCA was performed on the dataset for data reduction.

This created a simplified dataset having five variables. They are “Satisfaction”, “Sales Distribution”, “Marketing”, “After Sales Service” and “Value for Money”. This was tested for dependency among the independent variables. It never showed any signs of multi-collinearity.

As we can see the data from various tests, the results have improved positively. Hence, we can conclude that this is the best fit model.