Loading the Data Set

```r
setwd("D:\\BABI\\Advanced_Statistics\\Dataset")
myfactordata = read.csv("Factor-Hair-Revised.csv", header = TRUE)
```

Let's do some Exploratory Data Analysis

```r
head(myfactordata)

##   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

names(myfactordata)

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

dim(myfactordata)

## [1] 100  13

class(myfactordata)

## [1] "data.frame"

str(myfactordata)

## '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 ...
```

```
summary(myfactordata)
```

```
##        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
```

Let's Remove the First Column - **"ID"** as it's a categorical variable though it is nominal in nature

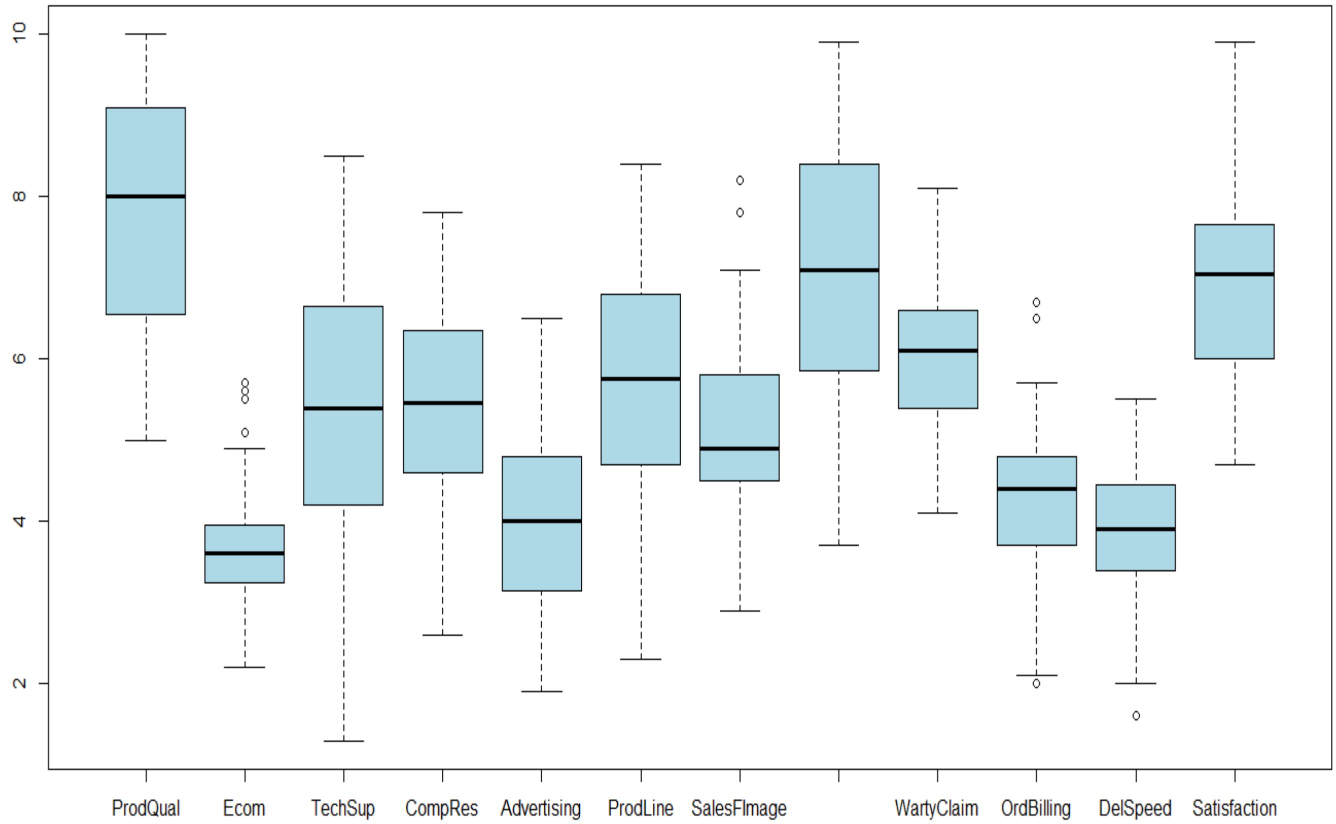
```
myfactordata = myfactordata[,-1]
names(myfactordata)
```

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

```
attach(myfactordata)
```

Let's plot the **Boxplot** for each variable to check whether there are any outliers

```
boxplot(myfactordata[,-1] col = "light blue")
```

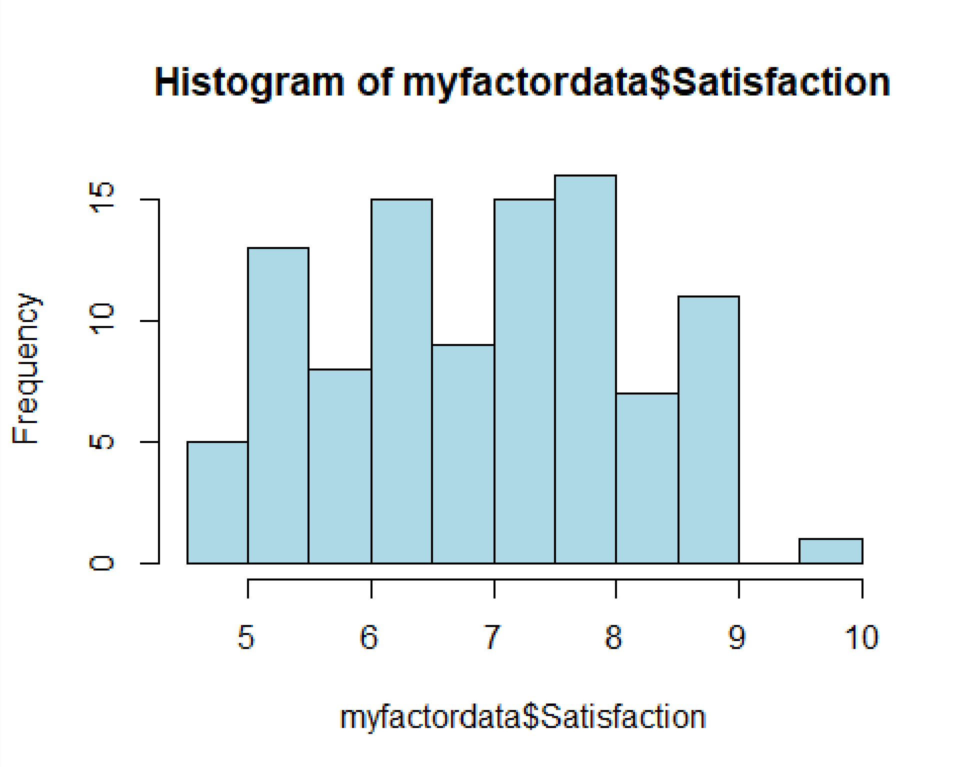


We observe that there are outliers in few of the Variables below:

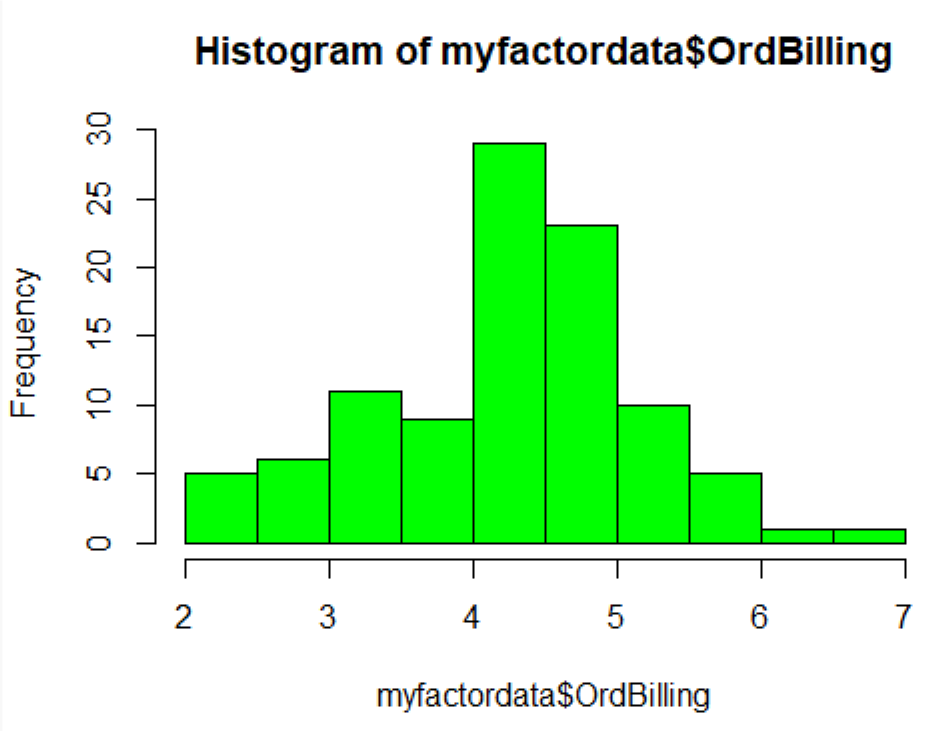
- Ecommerce - Outliers on the Higher side
- SalesforceImage - Outliers on the Higher side
- Order & Billing - Outliers on both Low and Higher sides
- Deliver Speed - Outliers on the Lower side

We could infer that there are instances where Order&Billing and DeliverySpeed have even very low values which is good.

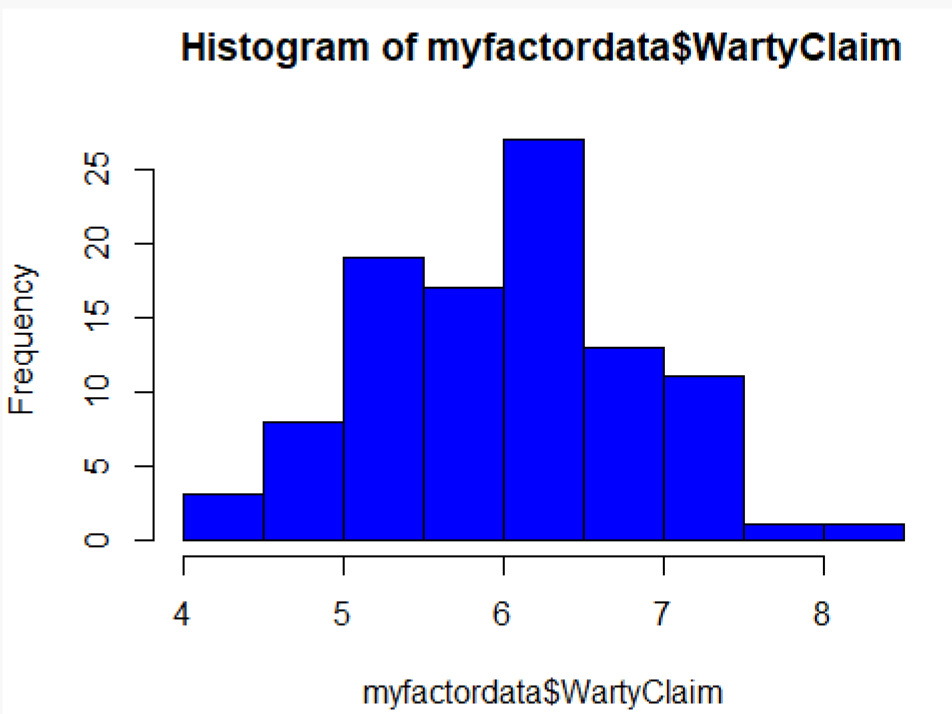
```
hist(myfactordata$Satisfaction, col = "Light Blue")
```



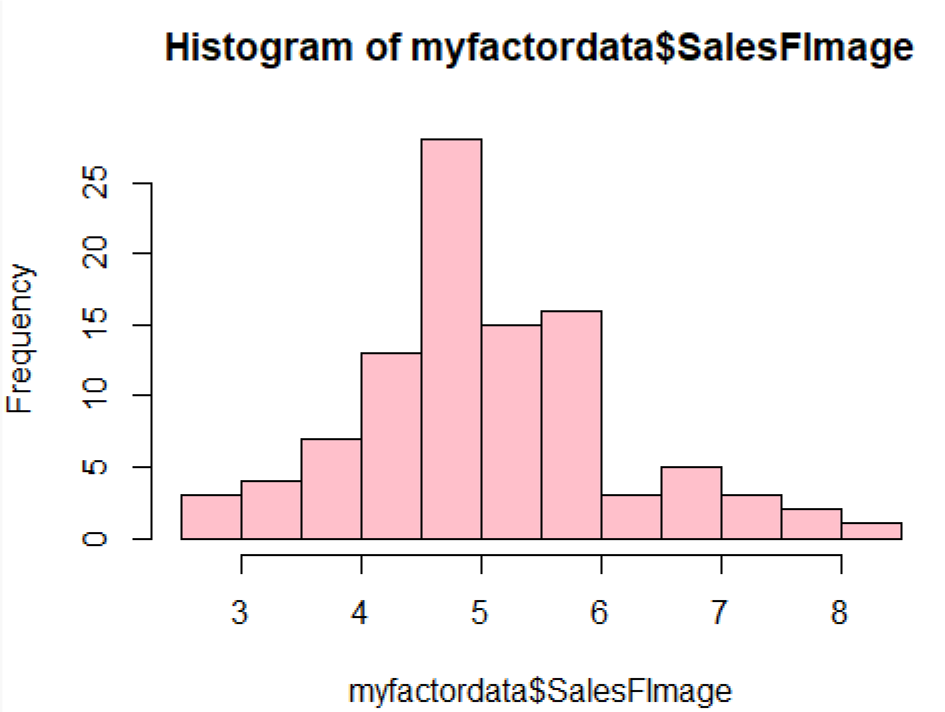

```
hist(myfactordata$OrdBilling, col = "Green")
```



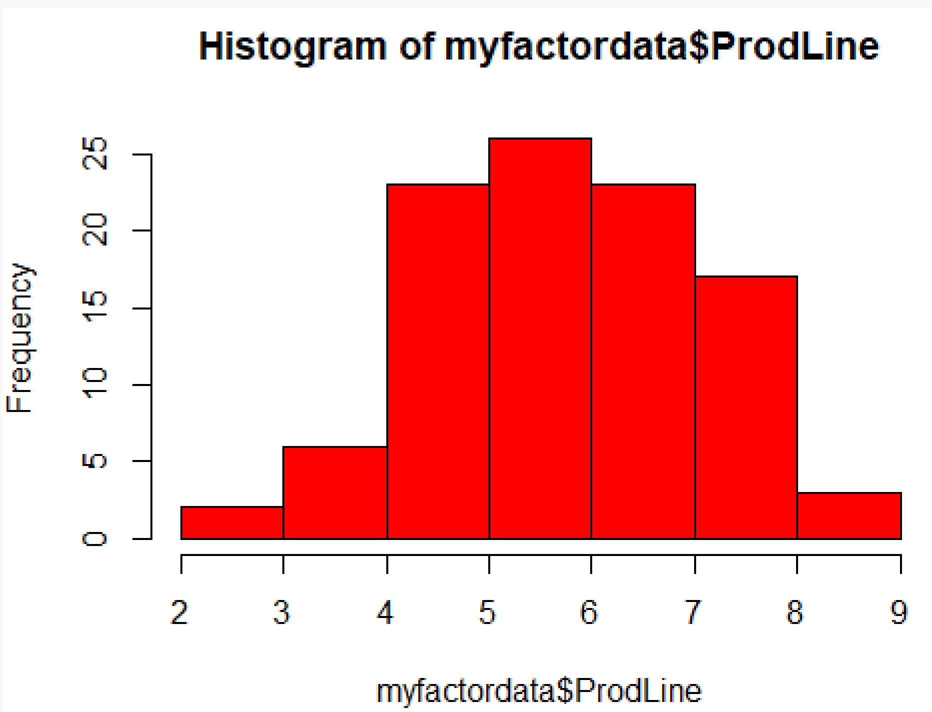

```
hist(myfactordata$WartyClaim, col = "Blue")
```



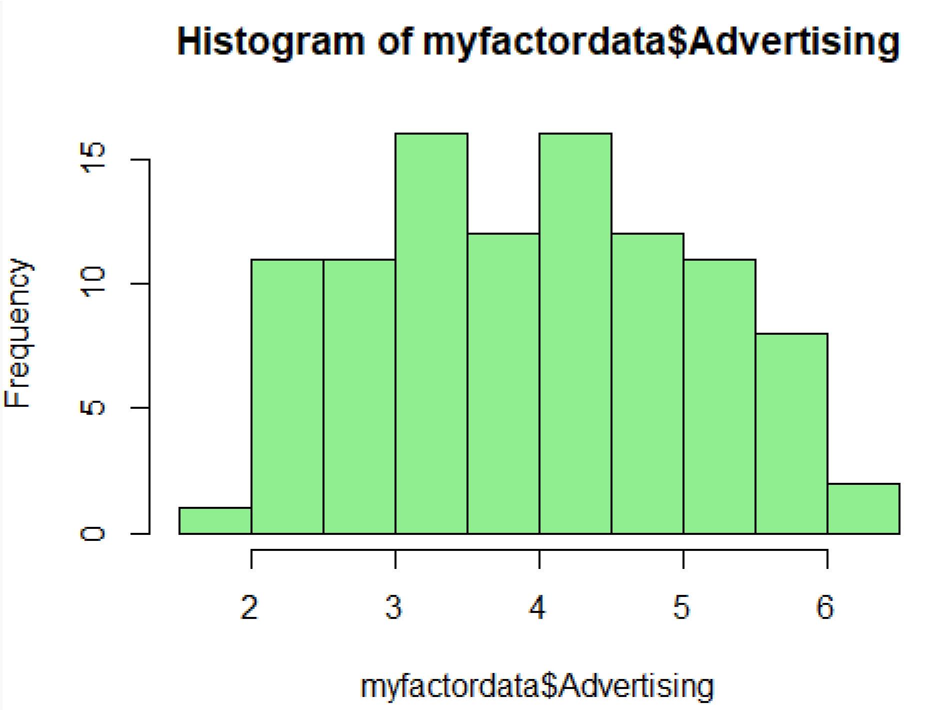

```
hist(myfactordata$SalesFImage,col = "Pink")
```



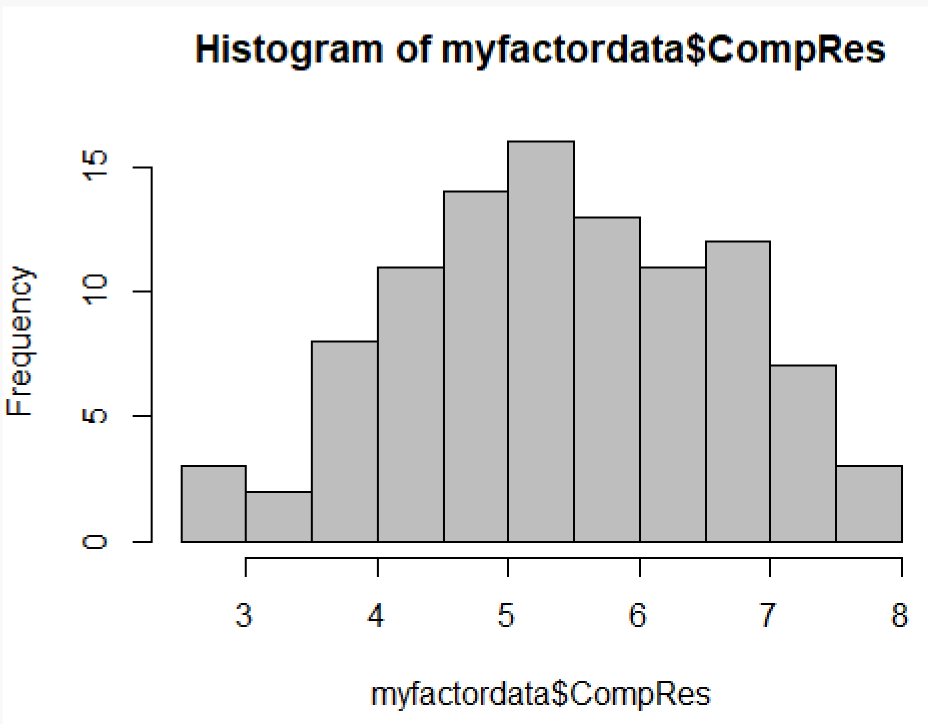
hist(myfactordata$ProdLine, col = "Red")



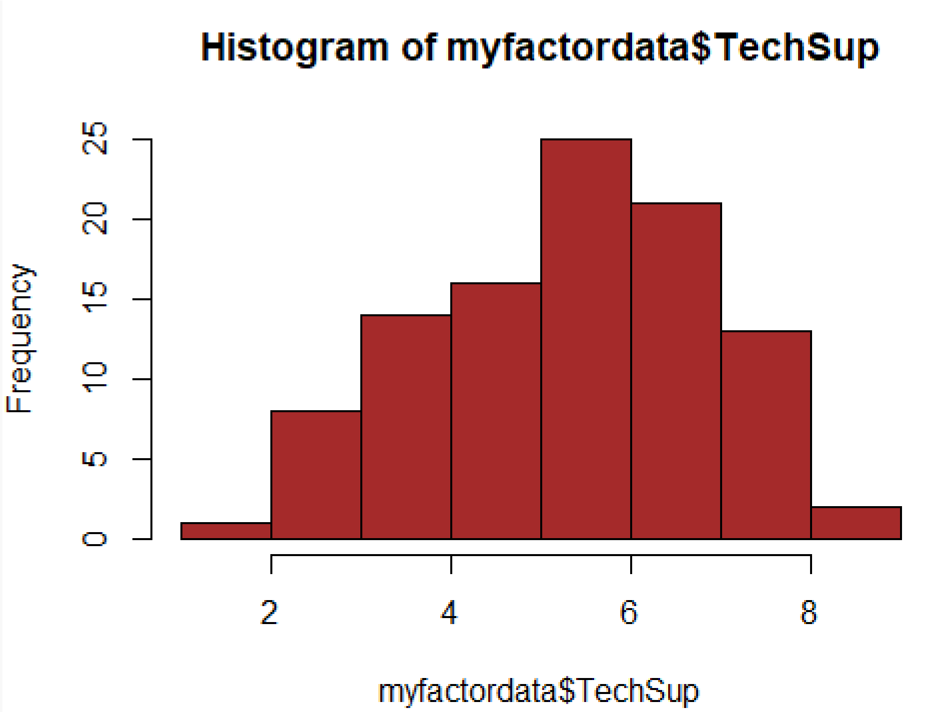
hist(myfactordata$Advertising, col = "Light green")



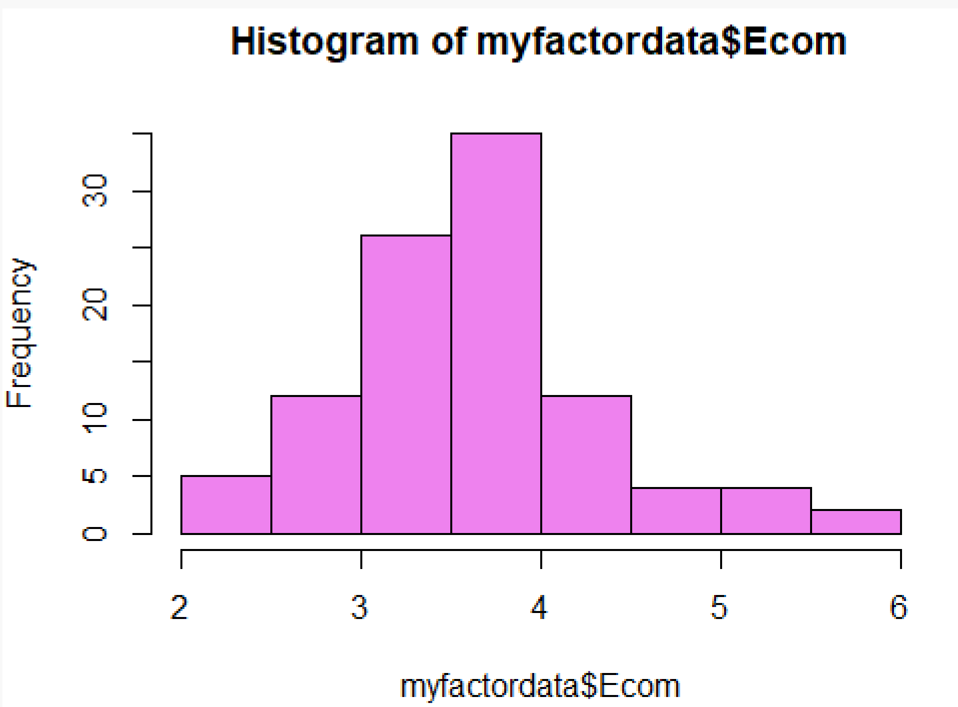
hist(myfactordata$CompRes, col = "Grey")



hist(myfactordata$TechSup, col = "Brown")



hist(myfactordata$Ecom, col = "Violet")



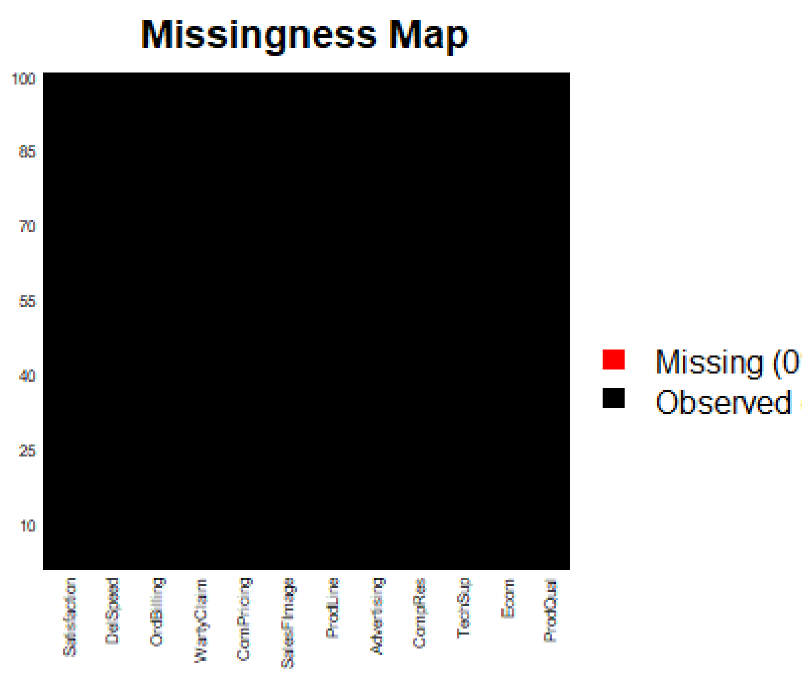
Let’s Load the required libraries before we proceed further …

```
library(psych)
library(Amelia)
```

```
## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.5, built: 2018-05-07)
## ## Copyright (C) 2005-2020 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##

library(corrplot)

## Warning: package 'corrplot' was built under R version 3.6.2

## corrplot 0.84 loaded

library(ppcor)

## Warning: package 'ppcor' was built under R version 3.6.2

## Loading required package: MASS
```

Let's check if there are any missing Variables?

```
missmap(myfactordata, col = c('red','black'),y.cex=0.5,x.cex=0.5)
```



As per the missing map plotted it is clear that there is no missing data

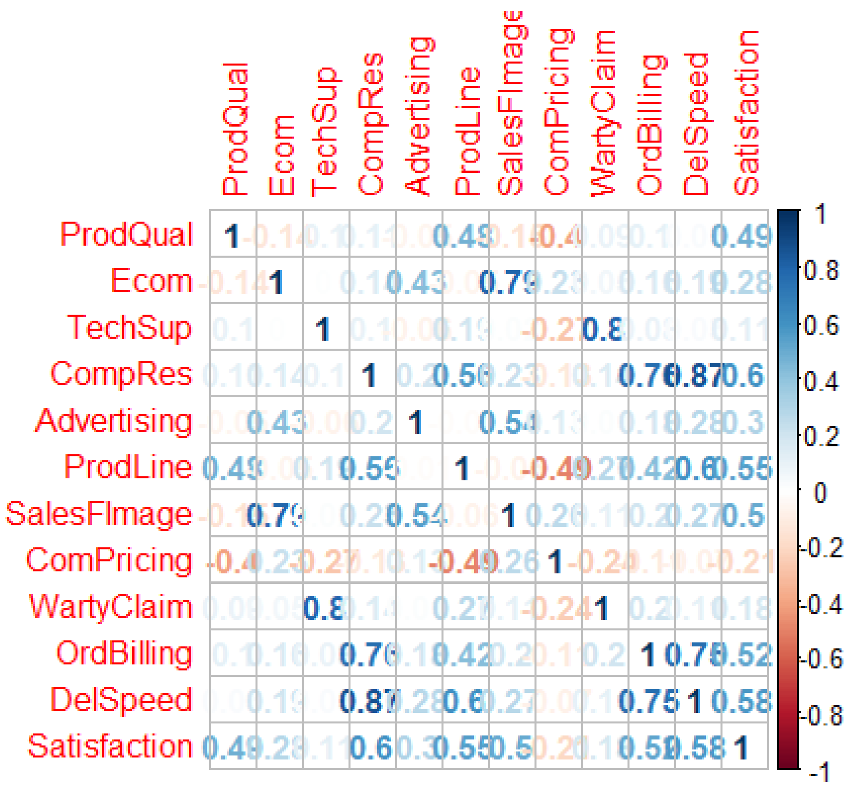
Our end goal is to build a good Multiple Linear Regression model. However, for any Linear regression model to predict with high accuracy, it is important that the Independent variables do not have correlation between each other and have high correlation with the Dependent Variable only.

In our dataset, Satisfaction is Dependent Variable or Response Variable and rest of the variables are Independent Variable.

So, before we proceed, we need to confirm if there are any multicollinearity issue between the Independent Variables. For example, in case Salary is dependent variable explained by 2 independent variable Age and experience, both Age and experience by itself have very high correlation, so they fight with each other in trying to explain the dependent variable salary and eventually both have lesser explanatory power.

We shall run cor and corrplot to check the correlation between the independent variables.

```
factordatamatrix = cor(myfactordata)
corrplot(factordatamatrix, method = "number")
```



It’s clear from the plot that, there are high correlation between few of the Independent variables and it would impact the overall significance of the model built.

1. SalesForceImage and Ecommerce are highly correlated
2. WartyClaim and TechSupport are highly correlated
3. DelSpeed and Complaint Resolution are highly correlated
4. At the same time even OrderBilling and Complain Resolution also are highly correlated
5. On top of it OrdBilling and Delivery Speed are highly correlated.

As there are correlation between one Independent Variable with more than one independent Variables, there is a **multicollinearity** issue with the dataset.

Just to double confirm the multicollinearity, lets even check the p-value of the correlation to know how significant the correlations are.

```
pcor(myfactordata,method = "pearson")

## $estimate
##                  ProdQual          Ecom      TechSup      CompRes
## ProdQual      1.000000000  0.1549597037  0.002424222 -0.056354142
## Ecom          0.154959704  1.0000000000  0.082359000 -0.033513777
## TechSup       0.002424222  0.0823590001  1.000000000  0.143603415
## CompRes      -0.056354142 -0.0335137768  0.143603415  1.000000000
## Advertising   0.112376746 -0.0002972504 -0.059300254 -0.064806093
## ProdLine      0.281144724  0.1538660545 -0.125349050  0.020305650
## SalesFImage  -0.376228551  0.7321011890 -0.093310796 -0.022150933
## ComPricing   -0.014021386  0.0149131857 -0.132972485 -0.004487151
## WartyClaim   -0.042752416 -0.1232553822  0.787729606 -0.109686211
## OrdBilling    0.054523215  0.1546990521 -0.165914724  0.288344382
## DelSpeed     -0.335084210 -0.0083917930 -0.021914916  0.528932328
## Satisfaction  0.607438787 -0.3308440834  0.055111867  0.172416347
##                Advertising    ProdLine  SalesFImage   ComPricing
## ProdQual      0.1123767461  0.28114472 -0.376228551 -0.014021386
## Ecom         -0.0002972504  0.15386605  0.732101189  0.014913186
## TechSup      -0.0593002540 -0.12534905 -0.093310796 -0.132972485
## CompRes      -0.0648060931  0.02030565 -0.022150933 -0.004487151
## Advertising   1.0000000000 -0.13192319  0.262879343 -0.063298038
## ProdLine     -0.1319231866  1.00000000 -0.230170570 -0.361757904
## SalesFImage   0.2628793429 -0.23017057  1.000000000  0.126612489
## ComPricing   -0.0632980381 -0.36175790  0.126612489  1.000000000
## WartyClaim    0.0275909587  0.25718360  0.189300271  0.020351554
## OrdBilling   -0.0326580430 -0.28098068 -0.182359301 -0.086500869
## DelSpeed      0.2046544064  0.50189505  0.005889925  0.190437993
## Satisfaction -0.0449735411  0.18325156  0.660251850 -0.087467711
##               WartyClaim  OrdBilling     DelSpeed Satisfaction
## ProdQual     -0.04275242  0.05452322 -0.335084210   0.60743879
## Ecom         -0.12325538  0.15469905 -0.008391793  -0.33084408
## TechSup       0.78772961 -0.16591472 -0.021914916   0.05511187
## CompRes      -0.10968621  0.28834438  0.528932328   0.17241635
## Advertising   0.02759096 -0.03265804  0.204654406  -0.04497354
## ProdLine      0.25718360 -0.28098068  0.501895051   0.18325156
## SalesFImage   0.18930027 -0.18235930  0.005889925   0.66025185
## ComPricing    0.02035155 -0.08650087  0.190437993  -0.08746771
## WartyClaim    1.00000000  0.25984451 -0.091649002  -0.08868071
## OrdBilling    0.25984451  1.00000000  0.350530212   0.14880055
## DelSpeed     -0.09164900  0.35053021  1.000000000   0.08955614
## Satisfaction -0.08868071  0.14880055  0.089556143   1.00000000
##
## $p.value
```

```
##                 ProdQual         Ecom      TechSup      CompRes
## ProdQual     0.000000e+00 1.447435e-01 9.819081e-01 5.977987e-01
## Ecom         1.447435e-01 0.000000e+00 4.402823e-01 7.538375e-01
## TechSup      9.819081e-01 4.402823e-01 0.000000e+00 1.769171e-01
## CompRes      5.977987e-01 7.538375e-01 1.769171e-01 0.000000e+00
## Advertising  2.916350e-01 9.977814e-01 5.787601e-01 5.439511e-01
## ProdLine     7.269030e-03 1.476353e-01 2.391187e-01 8.493380e-01
## SalesFImage  2.576212e-04 2.442012e-16 3.817042e-01 8.358301e-01
## ComPricing   8.956443e-01 8.890480e-01 2.115149e-01 9.665194e-01
## WartyClaim   6.890839e-01 2.471182e-01 3.262868e-20 3.034147e-01
## OrdBilling   6.097697e-01 1.454288e-01 1.180864e-01 5.850710e-03
## DelSpeed     1.245192e-03 9.374303e-01 8.375553e-01 8.359069e-08
## Satisfaction 2.182238e-10 1.447603e-03 6.059096e-01 1.041593e-01
##              Advertising     ProdLine  SalesFImage   ComPricing
## ProdQual      0.29163501 7.269030e-03 2.576212e-04 0.8956443272
## Ecom          0.99778145 1.476353e-01 2.442012e-16 0.8890479591
## TechSup       0.57876015 2.391187e-01 3.817042e-01 0.2115148546
## CompRes       0.54395113 8.493380e-01 8.358301e-01 0.9665194173
## Advertising   0.00000000 2.151737e-01 1.230672e-02 0.5533828069
## ProdLine      0.21517368 0.000000e+00 2.907563e-02 0.0004593443
## SalesFImage   0.01230672 2.907563e-02 0.000000e+00 0.2343791374
## ComPricing    0.55338281 4.593443e-04 2.343791e-01 0.0000000000
## WartyClaim    0.79629803 1.440249e-02 7.394541e-02 0.8490014635
## OrdBilling    0.75993047 7.304606e-03 8.538047e-02 0.4175555780
## DelSpeed      0.05300070 4.664278e-07 9.560619e-01 0.0721942454
## Satisfaction  0.67382405 8.383625e-02 1.449719e-12 0.4123500152
##                WartyClaim   OrdBilling     DelSpeed Satisfaction
## ProdQual     6.890839e-01 0.6097697424 1.245192e-03 2.182238e-10
## Ecom         2.471182e-01 0.1454287628 9.374303e-01 1.447603e-03
## TechSup      3.262868e-20 0.1180864174 8.375553e-01 6.059096e-01
## CompRes      3.034147e-01 0.0058507099 8.359069e-08 1.041593e-01
## Advertising  7.962980e-01 0.7599304715 5.300070e-02 6.738241e-01
## ProdLine     1.440249e-02 0.0073046065 4.664278e-07 8.383625e-02
## SalesFImage  7.394541e-02 0.0853804743 9.560619e-01 1.449719e-12
## ComPricing   8.490015e-01 0.4175555780 7.219425e-02 4.123500e-01
## WartyClaim   0.000000e+00 0.0133877361 3.902770e-01 4.058729e-01
## OrdBilling   1.338774e-02 0.0000000000 7.064114e-04 1.615975e-01
## DelSpeed     3.902770e-01 0.0007064114 0.000000e+00 4.012356e-01
## Satisfaction 4.058729e-01 0.1615974575 4.012356e-01 0.000000e+00
## 
## $statistic
##                ProdQual         Ecom     TechSup     CompRes  Advertising
## ProdQual      0.00000000  1.471424516  0.02274128 -0.52949015  1.060907458
## Ecom          1.47142452  0.000000000  0.77522957 -0.31456380 -0.002788456
## TechSup       0.02274128  0.775229571  0.00000000  1.36122814 -0.557266374
## CompRes      -0.52949015 -0.314563798  1.36122814  0.00000000 -0.609215688
## Advertising   1.06090746 -0.002788456 -0.55726637 -0.60921569  0.000000000
## ProdLine      2.74821968  1.460787002 -1.18522655  0.19052316 -1.248460807
## SalesFImage  -3.80921125 10.081854496 -0.87916864 -0.20784517  2.555921881
## ComPricing   -0.13154519  0.139913642 -1.25856891 -0.04209363 -0.594981366
```

```
## WartyClaim   -0.40142023 -1.165122048 11.99562484 -1.03519395  0.258924708
## OrdBilling    0.51223505  1.468888754 -1.57829305  2.82489237 -0.306523103
## DelSpeed     -3.33624278 -0.078724768 -0.20562952  5.84663073  1.961341689
## Satisfaction  7.17336519 -3.288799958  0.51778207  1.64199901 -0.422316520
##                ProdLine SalesFImage  ComPricing WartyClaim OrdBilling
## ProdQual      2.7482197 -3.80921125 -0.13154519 -0.4014202  0.5122350
## Ecom          1.4607870 10.08185450  0.13991364 -1.1651220  1.4688888
## TechSup      -1.1852266 -0.87916864 -1.25856891 11.9956248 -1.5782930
## CompRes       0.1905232 -0.20784517 -0.04209363 -1.0351939  2.8248924
## Advertising  -1.2484608  2.55592188 -0.59498137  0.2589247 -0.3065231
## ProdLine      0.0000000 -2.21876450 -3.64012829  2.4965744 -2.7464786
## SalesFImage  -2.2187645  0.00000000  1.19736653  1.8084929 -1.7398559
## ComPricing   -3.6401283  1.19736653  0.00000000  0.1909540 -0.8145030
## WartyClaim    2.4965744  1.80849286  0.19095404  0.0000000  2.5242649
## OrdBilling   -2.7464786 -1.73985585 -0.81450302  2.5242649  0.0000000
## DelSpeed      5.4434474  0.05525335  1.81976992 -0.8633775  3.5110350
## Satisfaction  1.7486638  8.24679932 -0.82367672 -0.8351894  1.4115877
##                 DelSpeed Satisfaction
## ProdQual     -3.33624278    7.1733652
## Ecom         -0.07872477   -3.2888000
## TechSup      -0.20562952    0.5177821
## CompRes       5.84663073    1.6419990
## Advertising   1.96134169   -0.4223165
## ProdLine      5.44344736    1.7486638
## SalesFImage   0.05525335    8.2467993
## ComPricing    1.81976992   -0.8236767
## WartyClaim   -0.86337748   -0.8351894
## OrdBilling    3.51103503    1.4115877
## DelSpeed      0.00000000    0.8435005
## Satisfaction  0.84350046    0.0000000
## 
## $n
## [1] 100
## 
## $gp
## [1] 10
## 
## $method
## [1] "pearson"
```

It is evident from the **P-Values of the Correlated Variable** that there is significant Correlation, and hence it is sure we have **multicollinearity** issue.

Before we run an MLR, we shall run Linear Regression on each of the Independent Variable to see which has highest explanatory power over the dependent variable - **Satisfaction**

```
LinearModel_ProdQual = lm(Satisfaction~ProdQual)
print(summary(LinearModel_ProdQual), digits = 10)
```

```
##
## Call:
## lm(formula = Satisfaction ~ ProdQual)
##
## Residuals:
##            Min             1Q         Median             3Q            Max
## -1.88745536501 -0.72710926895 -0.01577047821  0.85641075592  2.25219853894
##
## Coefficients:
##                   Estimate    Std. Error t value   Pr(>|t|)
## (Intercept) 3.67592539247 0.59765313111 6.15060 1.6807e-08 ***
## ProdQual    0.41511838765 0.07534135835 5.50983 2.9010e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.046703 on 98 degrees of freedom
## Multiple R-squared:  0.2365120037,   Adjusted R-squared:  0.2287213099
## F-statistic: 30.35827214 on 1 and 98 DF,  p-value: 2.900993371e-07
```

```r
LinearModel_Ecom = lm(Satisfaction~Ecom)
print(summary(LinearModel_Ecom), digits = 10)
```

```
##
## Call:
## lm(formula = Satisfaction ~ Ecom)
##
## Residuals:
##            Min             1Q         Median             3Q            Max
## -2.37199680537 -0.78970680258  0.04958502808  0.68084501128  2.34579593920
##
## Coefficients:
##                  Estimate   Std. Error t value   Pr(>|t|)
## (Intercept) 5.1515676717 0.6161439235 8.36098 4.2799e-13 ***
## Ecom        0.4810545556 0.1648516480 2.91811  0.0043677 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.149024 on 98 degrees of freedom
## Multiple R-squared:  0.07994474333,   Adjusted R-squared:  0.07055642438
## F-statistic: 8.515341649 on 1 and 98 DF,  p-value: 0.00436771189
```

```r
LinearModel_TechSup = lm(Satisfaction~TechSup)
print(summary(LinearModel_TechSup), digits = 10)
```

```
##
## Call:
## lm(formula = Satisfaction ~ TechSup)
##
## Residuals:
##            Min             1Q         Median             3Q            Max
## -2.26136458412 -0.93296546817  0.04301965393  0.82501474853  2.85617236807
```

```
##
## Coefficients:
##                  Estimate    Std. Error  t value Pr(>|t|)
## (Intercept) 6.44757125761 0.43592102077 14.79069  < 2e-16 ***
## TechSup     0.08768476093 0.07816511212  1.12179  0.26469
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.190287 on 98 degrees of freedom
## Multiple R-squared:  0.01267812466,  Adjusted R-squared:  0.00260341165
## F-statistic: 1.258410502 on 1 and 98 DF,  p-value: 0.2646932926
```

```r
LinearModel_CompRes = lm(Satisfaction~CompRes)
print(summary(LinearModel_CompRes), digits = 10)
```

```
##
## Call:
## lm(formula = Satisfaction ~ CompRes)
##
## Residuals:
##           Min             1Q         Median             3Q            Max
## -2.40449919620 -0.66163999790  0.04498711986  0.63036912473  2.70949298440
##
## Coefficients:
##                Estimate   Std. Error t value   Pr(>|t|)
## (Intercept) 3.6800454610 0.4428476912 8.30996 5.5081e-13 ***
## CompRes     0.5949934838 0.0794596537 7.48799 3.0853e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.95538 on 98 degrees of freedom
## Multiple R-squared:  0.3639257693,   Adjusted R-squared:  0.3574352159
## F-statistic: 56.07006804 on 1 and 98 DF,  p-value: 3.085348525e-11
```

```r
LinearModel_Advertising = lm(Satisfaction~Advertising)
print(summary(LinearModel_Advertising), digits = 10)
```

```
##
## Call:
## lm(formula = Satisfaction ~ Advertising)
##
## Residuals:
##           Min             1Q         Median             3Q            Max
## -2.34032784538 -0.92755285135  0.05576855166  0.79773109043  2.53412216655
##
## Coefficients:
##                Estimate   Std. Error  t value   Pr(>|t|)
## (Intercept) 5.6259207826 0.4236826403 13.27862 < 2.22e-16 ***
## Advertising 0.3222142687 0.1017533427  3.16662  0.0020561 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.140954 on 98 degrees of freedom
## Multiple R-squared:  0.09282348879,  Adjusted R-squared:  0.08356658562
## F-statistic: 10.02748835 on 1 and 98 DF,  p-value: 0.002056064961
```

```
LinearModel_ProdLine = lm(Satisfaction~ProdLine)
print(summary(LinearModel_ProdLine), digits = 10)
```

```
##
## Call:
## lm(formula = Satisfaction ~ ProdLine)
##
## Residuals:
##           Min            1Q        Median            3Q           Max
## -2.3633673055 -0.7794968251  0.1096631877  0.7604509904  1.7373079539
##
## Coefficients:
##                   Estimate    Std. Error t value   Pr(>|t|)
## (Intercept) 4.02203313530 0.45471260765 8.84522 3.8680e-14 ***
## ProdLine    0.49887456756 0.07641343799 6.52862 2.9531e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.000017 on 98 degrees of freedom
## Multiple R-squared:  0.3031008275,   Adjusted R-squared:  0.2959896114
## F-statistic:  42.6229249 on 1 and 98 DF,  p-value: 2.953079537e-09
```

```
LinearModel_SalesFImage = lm(Satisfaction~SalesFImage)
print(summary(LinearModel_SalesFImage), digits = 10)
```

```
##
## Call:
## lm(formula = Satisfaction ~ SalesFImage)
##
## Residuals:
##           Min            1Q        Median            3Q           Max
## -2.2164044938 -0.5884256978  0.1837764226  0.6921700614  2.0727658206
##
## Coefficients:
##                   Estimate    Std. Error t value   Pr(>|t|)
## (Intercept) 4.06982925602 0.50874011068 7.99982 2.5392e-12 ***
## SalesFImage 0.55595759203 0.09721907888 5.71861 1.1643e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.037274 on 98 degrees of freedom
## Multiple R-squared:  0.2502053484,   Adjusted R-squared:  0.2425543826
## F-statistic: 32.70245273 on 1 and 98 DF,  p-value: 1.16431356e-07
```

```
LinearModel_ComPricing = lm(Satisfaction~ComPricing)
print(summary(LinearModel_ComPricing), digits = 10)
```

```
##
## Call:
## lm(formula = Satisfaction ~ ComPricing)
##
## Residuals:
##           Min            1Q        Median            3Q           Max
## -1.9728067755 -0.9914656222 -0.1156192718  0.9111255164  2.5844849035
##
## Coefficients:
##                  Estimate    Std. Error  t value Pr(>|t|)
## (Intercept)  8.03856195765  0.54426927455 14.76946  < 2e-16 ***
## ComPricing  -0.16067708025  0.07621294255 -2.10827 0.037559 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.17163 on 98 degrees of freedom
## Multiple R-squared:  0.04338709402,  Adjusted R-squared:  0.03362573784
## F-statistic: 4.444781361 on 1 and 98 DF,  p-value: 0.03755876831
```

```
LinearModel_WartyClaim = lm(Satisfaction~WartyClaim)
print(summary(LinearModel_WartyClaim), digits = 10)
```

```
##
## Call:
## lm(formula = Satisfaction ~ WartyClaim)
##
## Residuals:
##            Min             1Q         Median             3Q            Max
## -2.36503725661 -0.90201780982  0.03019303992  0.90763418619  2.88984503593
##
## Coefficients:
##                 Estimate   Std. Error t value   Pr(>|t|)
## (Intercept) 5.3580771769 0.8813473789 6.07942 2.3214e-08 ***
## WartyClaim  0.2581371542 0.1445354304 1.78598   0.077196 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.178873 on 98 degrees of freedom
## Multiple R-squared:  0.03152216275,  Adjusted R-squared:  0.02163973584
## F-statistic: 3.189718785 on 1 and 98 DF,  p-value: 0.07719560357
```

```
LinearModel_OrdBilling = lm(Satisfaction~OrdBilling)
print(summary(LinearModel_OrdBilling), digits = 10)
```

```
##
## Call:
## lm(formula = Satisfaction ~ OrdBilling)
##
## Residuals:
##           Min            1Q        Median            3Q           Max
## -2.4005118743 -0.7070997382 -0.0344037578  0.7340083782  2.9672718928
```

```
##
## Coefficients:
##                   Estimate   Std. Error t value   Pr(>|t|)
## (Intercept) 4.0540526111 0.4839652782 8.37674 3.9589e-13 ***
## OrdBilling  0.6694594177 0.1105779647 6.05418 2.6020e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.021944 on 98 degrees of freedom
## Multiple R-squared:  0.2722041884,   Adjusted R-squared:  0.2647777005
## F-statistic: 36.65315193 on 1 and 98 DF,  p-value: 2.601982011e-08
```

```
LinearModel_DelSpeed = lm(Satisfaction~DelSpeed)
print(summary(LinearModel_DelSpeed), digits = 10)
```

```
##
## Call:
## lm(formula = Satisfaction ~ DelSpeed)
##
## Residuals:
##           Min            1Q        Median            3Q           Max
## -2.2247518745 -0.5484558168  0.0879641351  0.5446155834  2.5943221399
##
## Coefficients:
##                   Estimate   Std. Error t value   Pr(>|t|)
## (Intercept) 3.2790720669 0.5293818910 6.19415 1.3784e-08 ***
## DelSpeed    0.9364199519 0.1338813670 6.99440 3.3005e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9783458 on 98 degrees of freedom
## Multiple R-squared:  0.3329777866,   Adjusted R-squared:  0.3261714375
## F-statistic: 48.92164373 on 1 and 98 DF,  p-value: 3.300472407e-10
```

We can observe that only 3 of the independent variables have at least 30% of the explanatory power over the dependent variable Satisfaction. The 3 independent variables are - **Delivery Speed, warranty Claim and Complaint Resolution**

Now let's try to run MLR:

```
MLRModel_Factor = lm(Satisfaction~., data = myfactordata[,-12])
summary(MLRModel_Factor)
```

```
##
## Call:
## lm(formula = Satisfaction ~ ., data = myfactordata[, -12])
##
## 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
```

Adjusted R-squared is 0.7774, so the model seems have the capacity to explain ~78% variations in the Dependent variable based on the Variation in the Independent variable. Also, we can see that only 3 of the independent variables have high confidence interval than others. F-statistic of 32.43 at p-value: < 2.2e-16 is good, and implied that our Model is good predictor.

Linear Equation of the Model:

Satisfaction = - 0.669 + 0.371*ProdQual - 0.440*Ecom + 0.032*TechSup + 0.167*CompRes - 0.026*Advertising + 0.140ProdLine + 0.806*SalesFImage - 0.038*CompPricing - 0.102*WartyClaim + 0.146*OrdBilling + 0.165*DelSpeed

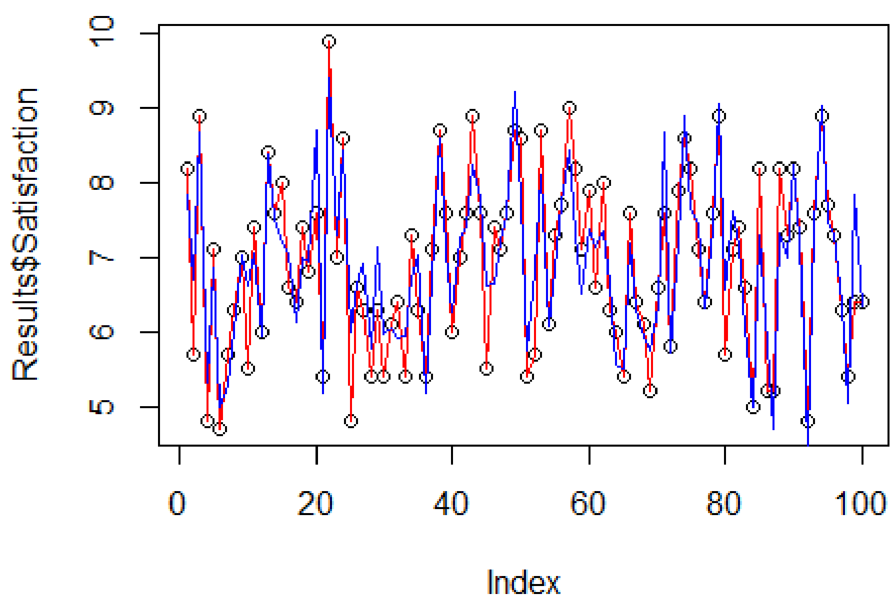
Let's also try plotting the actual and predicted satisfaction by the Model

```
Results = data.frame(myfactordata, fitted.value=fitted(MLRModel_Factor), resi
dual = resid(MLRModel_Factor))
head(Results)
```

```
##   ProdQual Ecom TechSup CompRes Advertising ProdLine SalesFImage
## 1      8.5  3.9     2.5     5.9         4.8      4.9         6.0
## 2      8.2  2.7     5.1     7.2         3.4      7.9         3.1
## 3      9.2  3.4     5.6     5.6         5.4      7.4         5.8
## 4      6.4  3.3     7.0     3.7         4.7      4.7         4.5
## 5      9.0  3.4     5.2     4.6         2.2      6.0         4.5
## 6      6.5  2.8     3.1     4.1         4.0      4.3         3.7
##   ComPricing WartyClaim OrdBilling DelSpeed Satisfaction fitted.value
## 1        6.8        4.7        5.0      3.7          8.2     7.835026
## 2        5.3        5.5        3.9      4.9          5.7     6.688189
## 3        4.5        6.2        5.4      4.5          8.9     8.666668
## 4        8.8        7.0        4.3      3.0          4.8     5.333509
```

```
## 5         6.8         6.1         4.5        3.5            7.1       6.875291
## 6         8.5         5.1         3.6        3.3            4.7       5.000756
##      residual
## 1  0.3649742
## 2 -0.9881894
## 3  0.2333317
## 4 -0.5335090
## 5  0.2247092
## 6 -0.3007562
```

```
plot(Results$Satisfaction)
lines(Results$Satisfaction, col = "red")
lines(Results$fitted.value, col = "Blue")
```



Plot shows, that the Predicted values are almost overlapping with the actual values. So the Model is generating prediction close enough.

We assume in regression that the independent variables are not correlated and they are all independent, but we saw earlier that there is correlation between few independent variables. Let's check the **Variable Inflation Factor (VIF)** values of the independent variables to identify the variable causing multicollinearity issue. Higher the VIF value, higher the issue.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.6.2

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.6.2

##
## Attaching package: 'ggplot2'

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

library(car)

## Warning: package 'car' was built under R version 3.6.2

## Loading required package: carData

##
## Attaching package: 'car'

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

vif(MLRModel_Factor)

##    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
```

It is apparent that **Delivery Speed** and **Complaint Resolution** are creating more multicollinearity issue.

As it is evident that there is multicollinearity issue with the independent variables, let's try to perform **Factor Analysis with PCA** and group the independent variables which are closely correlated. And use the Newly created Factors to build another model and check its validity. To start with PCA, lets calculate the **Eigen Value** to determine number of factors to consider.

Calculating Eigen Value

```
ev = eigen(cor(myfactordata[,-12]))
ev

## eigen() decomposition
## $values
##  [1] 3.42697133 2.55089671 1.69097648 1.08655606 0.60942409 0.55188378
##  [7] 0.40151815 0.24695154 0.20355327 0.13284158 0.09842702
```

```
##
## $vectors
##             [,1]        [,2]        [,3]       [,4]        [,5]
##  [1,] -0.1337896  0.31349802  0.06227164  0.6431362  0.23166620
##  [2,] -0.1659528 -0.44650918 -0.23524791  0.2723803  0.42228844
##  [3,] -0.1576926  0.23096734 -0.61095105 -0.1933931 -0.02395667
##  [4,] -0.4706836 -0.01944394  0.21035078 -0.2063204  0.02865743
##  [5,] -0.1837350 -0.36366471 -0.08809705  0.3178945 -0.80387024
##  [6,] -0.3867652  0.28478056  0.11627864  0.2029023  0.11667416
##  [7,] -0.2036696 -0.47069599 -0.24134210  0.2221772  0.20437283
##  [8,]  0.1516886 -0.41345650  0.05304529 -0.3335435  0.24892601
##  [9,] -0.2129336  0.19167191 -0.59856398 -0.1853020 -0.03292706
## [10,] -0.4372177 -0.02639905  0.16892981 -0.2368536  0.02675377
## [11,] -0.4730891 -0.07305172  0.23262477 -0.1973299 -0.03543294
##              [,6]         [,7]        [,8]        [,9]       [,10]
##  [1,]  0.56456996 -0.191641317  0.13547311  0.03132810 -0.06659717
##  [2,] -0.26325703 -0.059626208 -0.12202642 -0.54251104 -0.28155772
##  [3,]  0.10876896  0.017199915  0.46470964 -0.35929961  0.38817090
##  [4,]  0.02815231  0.008499596  0.51339754  0.09324751 -0.53467243
##  [5,]  0.20056937  0.063069619 -0.05347713 -0.15468169 -0.03715799
##  [6,] -0.09819533  0.608147555 -0.33320710 -0.08415534  0.23479794
##  [7,] -0.10497225 -0.001437351  0.16910665  0.64489911  0.35341191
##  [8,]  0.70973595  0.308248871 -0.09883227 -0.09414389  0.04518224
##  [9,]  0.13983966  0.030640243 -0.44354040  0.31756604 -0.43534752
## [10,]  0.11947974 -0.659319893 -0.36601754 -0.09907265  0.30386545
## [11,] -0.02979992  0.234239274  0.06539059 -0.02188514  0.12010386
##             [,11]
##  [1,] -0.18279209
##  [2,] -0.06233863
##  [3,]  0.05192956
##  [4,]  0.36253352
##  [5,]  0.08118684
##  [6,]  0.38507778
##  [7,]  0.08469869
##  [8,]  0.10295751
##  [9,] -0.12893245
## [10,]  0.19415064
## [11,] -0.77563222
```

```
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
```

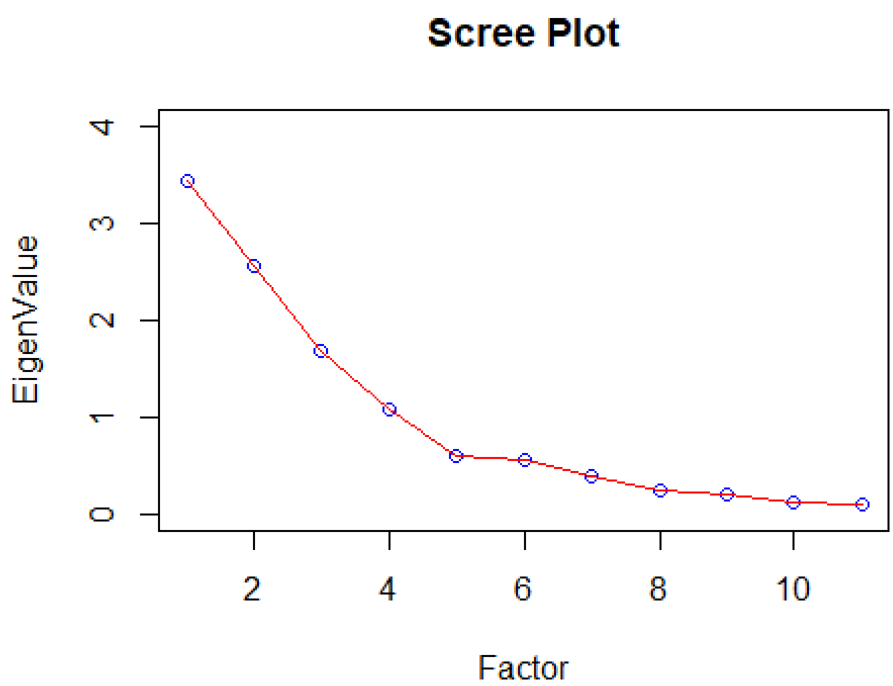
Let’s plot the **Scree plot** and apply Kaiser rule to choose the number of factors.

Scree Plot

```
Factor=c(1:11)
Factor
```

```
## [1]  1  2  3  4  5  6  7  8  9 10 11
```

```
Scree=data.frame(Factor, Eigenvalue)
plot(Scree,main="Scree Plot", col="Blue",ylim=c(0,4))
lines(Scree,col="Red")
```



There are 4 Eigen values above 1 and others are flattened and are below 1, so as per **Kaiser rule** lets go with 4 Factors for PCA.

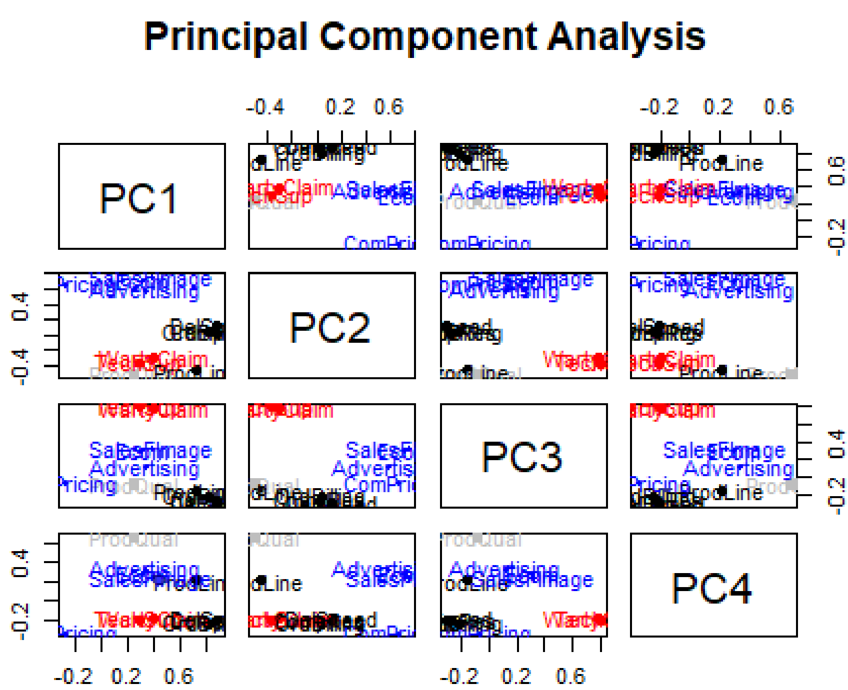
Running UnRotated PCA

```
library(psych)
Unrotate=principal(myfactordata[,-12], nfactors=4, rotate="none")
print(Unrotate,digits=3)
```

```
## Principal Components Analysis
## Call: principal(r = myfactordata[, -12], nfactors = 4, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                 PC1    PC2    PC3    PC4    h2     u2  com
## ProdQual      0.248 -0.501 -0.081  0.670 0.768 0.2320 2.20
## Ecom          0.307  0.713  0.306  0.284 0.777 0.2229 2.14
## TechSup       0.292 -0.369  0.794 -0.202 0.893 0.1069 1.87
## CompRes       0.871  0.031 -0.274 -0.215 0.881 0.1187 1.33
## Advertising   0.340  0.581  0.115  0.331 0.576 0.4240 2.38
## ProdLine      0.716 -0.455 -0.151  0.212 0.787 0.2129 2.01
## SalesFImage   0.377  0.752  0.314  0.232 0.859 0.1406 2.10
## ComPricing   -0.281  0.660 -0.069 -0.348 0.641 0.3594 1.94
## WartyClaim    0.394 -0.306  0.778 -0.193 0.892 0.1078 1.98
```

```
## OrdBilling   0.809  0.042 -0.220 -0.247 0.766 0.2339 1.35
## DelSpeed     0.876  0.117 -0.302 -0.206 0.914 0.0856 1.40
##
##                        PC1   PC2   PC3   PC4
## SS loadings           3.427 2.551 1.691 1.087
## Proportion Var        0.312 0.232 0.154 0.099
## Cumulative Var        0.312 0.543 0.697 0.796
## Proportion Explained  0.391 0.291 0.193 0.124
## Cumulative Proportion 0.391 0.683 0.876 1.000
##
## 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  39.023  with prob <  0.00177
##
## Fit based upon off diagonal values = 0.968
```

```r
UnrotatedProfile=plot(Unrotate,row.names(Unrotate$loadings))
```

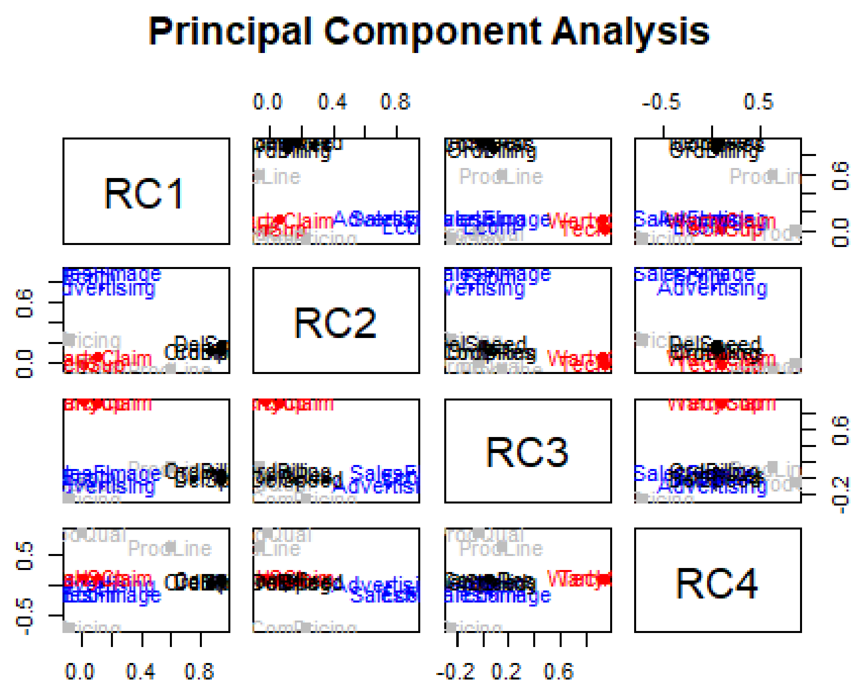


Unrotated output of PCA does not help us group the independent variables correctly, so let's get the rotated PCA loadings to determine the grouping

Running UnRotated PCA

```r
Rotate=principal(myfactordata[,-12],nfactors=4,rotate="varimax")
print(Rotate,digits=3)
```

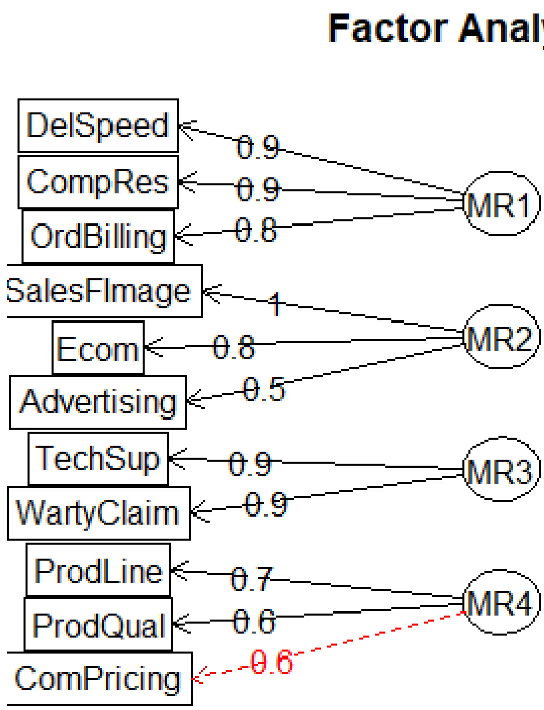
```
## Principal Components Analysis
## Call: principal(r = myfactordata[, -12], nfactors = 4, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                RC1    RC2    RC3    RC4    h2     u2  com
## ProdQual     0.002 -0.013 -0.033  0.876 0.768 0.2320 1.00
## Ecom         0.057  0.871  0.047 -0.117 0.777 0.2229 1.05
## TechSup      0.018 -0.024  0.939  0.101 0.893 0.1069 1.03
## CompRes      0.926  0.116  0.049  0.091 0.881 0.1187 1.06
## Advertising  0.139  0.742 -0.082  0.015 0.576 0.4240 1.10
## ProdLine     0.591 -0.064  0.146  0.642 0.787 0.2129 2.12
## SalesFImage  0.133  0.900  0.076 -0.159 0.859 0.1406 1.12
## ComPricing  -0.085  0.226 -0.246 -0.723 0.641 0.3594 1.47
## WartyClaim   0.110  0.055  0.931  0.102 0.892 0.1078 1.06
## OrdBilling   0.864  0.107  0.084  0.039 0.766 0.2339 1.05
## DelSpeed     0.938  0.177 -0.005  0.052 0.914 0.0856 1.08
##
##                         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
## Proportion Explained  0.330 0.255 0.212 0.203
## Cumulative Proportion 0.330 0.586 0.797 1.000
##
## 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  39.023  with prob <  0.00177
##
## Fit based upon off diagonal values = 0.968

RotatedProfile=plot(Rotate,row.names(Rotate$loadings),cex=1.0)
```



Let’s use fa function to perform PCA again just to leverage its capability to draw the grouping of independent variables in each PCA

Grouping of Independent Variables

```
Rotate1=fa(myfactordata[,-12],nfactors=4,rotate="varimax")
fa.diagram(Rotate1)
```



Let's check if the new factors are significant enough?

```
RC1_MLR_Data = cbind(myfactordata[,c(4,10,11)],Rotate$scores[,1])
head(RC1_MLR_Data)

##   CompRes OrdBilling DelSpeed Rotate$scores[, 1]
## 1     5.9        5.0      3.7          0.1274910
## 2     7.2        3.9      4.9          1.2216666
## 3     5.6        5.4      4.5          0.6158214
## 4     3.7        4.3      3.0         -0.8446267
## 5     4.6        4.5      3.5         -0.3197943
## 6     4.1        3.6      3.3         -0.6470292

RC1_MLR_Data_lm = lm(Rotate$scores[, 1]~., data = RC1_MLR_Data[,-4])
summary(RC1_MLR_Data_lm)

##
## Call:
## lm(formula = Rotate$scores[, 1] ~ ., data = RC1_MLR_Data[, -4])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49232 -0.10225  0.02771  0.11943  0.36076
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.11854    0.10727 -47.715  < 2e-16 ***
```

```
## CompRes      0.27572    0.03407   8.092 1.81e-12 ***
## OrdBilling   0.30486    0.03367   9.053 1.61e-14 ***
## DelSpeed     0.59544    0.05549  10.731  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1944 on 96 degrees of freedom
## Multiple R-squared:  0.9633, Adjusted R-squared:  0.9622
## F-statistic: 840.9 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
RC2_MLR_Data = cbind(myfactordata[,c(2,5,7)],Rotate$scores[,2])
head(RC2_MLR_Data)
```

```
##   Ecom Advertising SalesFImage Rotate$scores[, 2]
## 1  3.9         4.8         6.0          0.7698686
## 2  2.7         3.4         3.1         -1.6458617
## 3  3.4         5.4         5.8          0.5800037
## 4  3.3         4.7         4.5         -0.2719218
## 5  3.4         2.2         4.5         -0.8340650
## 6  2.8         4.0         3.7         -1.0672683
```

```
RC2_MLR_Data_lm = lm(Rotate$scores[, 2]~., data = RC2_MLR_Data[,-4])
summary(RC2_MLR_Data_lm)
```

```
##
## Call:
## lm(formula = Rotate$scores[, 2] ~ ., data = RC2_MLR_Data[, -4])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45663 -0.13507  0.01335  0.12344  0.33067
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.26482    0.10114  -52.05   <2e-16 ***
## Ecom         0.60226    0.04254   14.16   <2e-16 ***
## Advertising  0.31797    0.01923   16.53   <2e-16 ***
## SalesFImage  0.34711    0.02986   11.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1812 on 96 degrees of freedom
## Multiple R-squared:  0.9682, Adjusted R-squared:  0.9672
## F-statistic:   973 on 3 and 96 DF,  p-value: < 2.2e-16
```

```
RC3_MLR_Data = cbind(myfactordata[,c(3,9)],Rotate$scores[,3])
head(RC3_MLR_Data)
```

```
##   TechSup WartyClaim Rotate$scores[, 3]
## 1     2.5        4.7       -1.878446273
## 2     5.1        5.5       -0.614030010
```

```
## 3     5.6        6.2        0.003689252
## 4     7.0        7.0        1.267493254
## 5     5.2        6.1       -0.008096627
## 6     3.1        5.1       -1.303198892
```

```r
RC3_MLR_Data_lm = lm(Rotate$scores[, 3]~., data = RC3_MLR_Data[,-3])
summary(RC3_MLR_Data_lm)
```

```
##
## Call:
## lm(formula = Rotate$scores[, 3] ~ ., data = RC3_MLR_Data[, -3])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27104 -0.13002 -0.01434  0.11596  0.39519
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.58146    0.13971  -39.95   <2e-16 ***
## TechSup      0.35318    0.01797   19.65   <2e-16 ***
## WartyClaim   0.61007    0.03356   18.18   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1653 on 97 degrees of freedom
## Multiple R-squared:  0.9732, Adjusted R-squared:  0.9727
## F-statistic:  1764 on 2 and 97 DF,  p-value: < 2.2e-16
```

```r
RC4_MLR_Data = cbind(myfactordata[,c(1,6,8)],Rotate$scores[,4])
head(RC4_MLR_Data)
```

```
##   ProdQual ProdLine ComPricing Rotate$scores[, 4]
## 1      8.5      4.9        6.8          0.3664848
## 2      8.2      7.9        5.3          0.8130648
## 3      9.2      7.4        4.5          1.5699769
## 4      6.4      4.7        8.8         -1.2541645
## 5      9.0      6.0        6.8          0.4475377
## 6      6.5      4.3        8.5         -1.0527792
```

```r
RC4_MLR_Data_lm = lm(Rotate$scores[, 2]~., data = RC4_MLR_Data[,-4])
summary(RC4_MLR_Data_lm)
```

```
##
## Call:
## lm(formula = Rotate$scores[, 2] ~ ., data = RC4_MLR_Data[, -4])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.77900 -0.68128 -0.04165  0.52498  3.10641
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.84317    1.05335  -1.750   0.0833 .
## ProdQual     0.05856    0.08266   0.708   0.4804
## ProdLine     0.02507    0.09250   0.271   0.7869
## ComPricing   0.17784    0.07554   2.354   0.0206 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9852 on 96 degrees of freedom
## Multiple R-squared:  0.05885,    Adjusted R-squared:  0.02943
## F-statistic: 2.001 on 3 and 96 DF,  p-value: 0.119
```

Except RC4, all other groupings are typically have more than 95% R-Squared value and P-Values are also highly significant. So, the Groupings have come out really well.

Next let's build the new dataset with the factors and the DV - Satisfaction.

```
Rotate$scores
```

```
##                RC1         RC2          RC3         RC4
##   [1,]  0.12749104  0.76986860 -1.878446273  0.36648477
##   [2,]  1.22166663 -1.64586166 -0.614030010  0.81306481
##   [3,]  0.61582140  0.58000368  0.003689252  1.56997685
##   [4,] -0.84462665 -0.27192183  1.267493254 -1.25416452
##   [5,] -0.31979430 -0.83406501 -0.008096627  0.44753766
##   [6,] -0.64702925 -1.06726829 -1.303198892 -1.05277921
##   [7,] -2.62679851 -0.24588272 -0.555423494 -1.22601470
##   [8,] -0.27936394 -0.15732039 -0.749311481 -1.01464175
##   [9,]  1.05151341 -0.17228834 -0.092252815 -1.65809634
##  [10,]  0.42875382  0.76353272 -0.450377116 -0.89116595
##  [11,] -0.13580761 -0.76759698 -0.463706767  0.60634140
##  [12,] -1.45030579  1.35959912  0.437785016 -1.06981053
##  [13,]  0.62461823  2.11311565 -0.168284409  0.87466736
##  [14,]  0.42724294 -0.40405102  0.432245882  0.90236591
##  [15,]  1.43869881  0.66394839 -0.268050576 -1.04431806
##  [16,]  0.91969055 -1.05791159 -0.556847385  1.16667179
##  [17,]  0.52182175 -0.31959634  1.106009732 -1.03228845
##  [18,]  1.71349224 -0.16356534  0.254874808 -1.47834954
##  [19,]  1.16101062 -0.41943765 -0.375574495 -1.76167798
##  [20,]  0.29327394  1.77627892 -0.950139113  0.24112808
##  [21,] -0.61501848 -0.17897273  1.525943540 -1.83178487
##  [22,] -0.11282553  2.83382456  0.634265462  2.24434088
##  [23,]  0.08062000 -0.35141218  1.141318858  1.33498913
##  [24,]  1.94944755 -1.67141336 -0.859208476  0.50283683
##  [25,]  0.11534004 -0.01629685  0.471841920 -1.25041487
##  [26,]  0.57499258 -0.24490397  0.624292860 -1.35435360
##  [27,]  0.82896381 -0.98564797  1.042612499  0.92163700
##  [28,]  0.11695051 -1.10728007  0.379702318 -1.35959873
##  [29,]  1.15812632 -1.60628019 -0.055788125  0.79531052
##  [30,] -0.50739097  0.16192496 -1.551322987 -0.30617006
##  [31,] -0.81074131 -0.17909238  2.256638942  0.21624964
```

```
## [32,] -1.07438259 -1.60132074  1.186706049 -0.07026025
## [33,] -0.49992323  0.30576561  0.157100923 -0.97020760
## [34,]  0.27885747  0.07142401 -0.032941868 -0.65628441
## [35,] -1.21092268  0.61247373  0.275773660 -0.68907425
## [36,] -1.37569442 -1.05901060  0.277541003  1.02901615
## [37,] -0.62476762 -0.24359504  0.310901127  0.66051905
## [38,]  1.36407521  0.03533514  0.111220579  0.58229289
## [39,]  0.60127495  0.47053204 -1.291508459 -0.44567425
## [40,] -0.58595295  1.48246242 -1.184474889 -1.03900017
## [41,]  0.19167763 -0.38987441 -1.981705114 -0.59621998
## [42,]  0.04337736  0.09038218 -1.165712378  0.53711635
## [43,]  0.40978439  1.95821980 -1.094672035  0.98888677
## [44,]  0.77547735  1.61343935  1.512055016 -1.14923990
## [45,]  1.26977129 -1.77421869 -0.982794252  0.73741113
## [46,]  1.06006213  0.67869812  0.324241314 -1.10289754
## [47,] -0.12283972 -0.09120895  0.996132311  1.41658476
## [48,]  2.09832312  0.46224836  0.840138645 -1.68134357
## [49,]  0.15604110  0.88202250 -0.835276700  1.29848126
## [50,]  0.22982346  0.50302016 -0.877037378  1.03687279
## [51,] -0.94183170 -0.37565064  0.194174450 -0.65267018
## [52,]  1.56112818 -1.90837771 -1.176496580  0.72135781
## [53,]  0.86011758 -1.08934973 -0.241431240  0.87182584
## [54,] -0.81818435 -0.52905894  0.539901007  0.33090833
## [55,]  0.54057306 -0.67964718 -1.060702696 -0.81493134
## [56,] -0.36862437  0.28299033  0.917529711  0.60437604
## [57,]  1.97865621  1.43218345 -0.085319811 -0.83928511
## [58,]  0.20552648  0.51721871  0.347543516  0.85780222
## [59,] -1.34118399  0.55279292  0.326579529  1.94033636
## [60,]  0.85269365 -1.57772836  0.565957142  0.74035745
## [61,]  0.99335190 -1.26473291  1.700148685  0.79107349
## [62,] -1.10480994  0.70911509 -0.152796271  0.39572776
## [63,] -0.75921278  0.26001089 -1.188441475  0.78014681
## [64,] -1.09474826 -1.95079477  0.427161087 -0.14850194
## [65,] -1.20922892  0.15287985  0.577570622 -0.51556079
## [66,]  1.34313803  0.53659415 -1.039141561 -1.24941075
## [67,]  0.90215965 -0.58791187  2.062390350 -1.31875384
## [68,]  0.42318247 -0.24798003 -0.301264201 -0.84662237
## [69,] -0.87487795 -0.60376193 -0.997620068 -0.52944051
## [70,]  0.14372369 -0.15149397 -1.275988102 -1.00015303
## [71,]  0.34387385  2.05641521  0.686346140  0.09426189
## [72,] -1.16028876 -0.18463387 -1.205197353  0.71392258
## [73,]  0.92620350  1.31556747 -1.869872622 -0.55887325
## [74,] -0.56659595  1.40049678  1.226627789  1.34965616
## [75,] -0.29927186  0.87194345 -0.294625640  0.30300903
## [76,] -0.89076271  0.23334622  1.037887857  1.61337977
## [77,] -0.35535699  0.14354788  2.057316893 -0.63270298
## [78,]  0.21054781  0.34218260  1.073262401  0.30917078
## [79,]  1.12960563  0.64023318  0.441396478  1.46536309
## [80,] -1.53178615  0.28775431  0.032504303 -0.31110748
## [81,] -0.84995072 -0.24812793  0.452562850  1.53107516
```

```
## [82,]  0.02821132 -0.91638751  0.493585747  0.40440014
## [83,] -1.39215814 -0.98489128  0.207609940  0.62550901
## [84,] -2.48589153 -0.73564594  1.633547463 -1.44488070
## [85,]  1.00347560 -1.78211709  0.797684019 -0.01141758
## [86,] -0.82905678 -0.41939997 -1.080457442 -0.45156381
## [87,] -1.42542804 -0.29820535 -2.155317026 -1.27019948
## [88,]  1.07076650 -1.29822928  1.400760179  0.04006707
## [89,]  0.08823132 -0.05909838  0.134228700  0.23513720
## [90,]  1.07621515  2.37671168  1.892951438 -1.01341980
## [91,] -0.78483349  0.46274897  1.391773475  0.61318828
## [92,] -2.34793070 -0.26426141 -0.534487111 -1.18940207
## [93,]  0.29898878  0.20636519 -0.371416070  1.20810631
## [94,]  1.10722906  0.37021414  0.053771549  1.44542651
## [95,] -0.79676401  0.71175008 -1.087719898  1.06131961
## [96,] -0.11270919  0.39627233  0.048312077  0.34767120
## [97,] -0.20833274 -0.25264090 -1.880921516 -0.32081680
## [98,] -1.58596201 -1.12347151 -1.337515839  1.23670188
## [99,] -0.32827278  1.90243479  0.140227444 -0.12061112
## [100,] -0.62744070  0.21100398 -0.748923176 -0.69590553

PCA_MLR_Data = cbind(myfactordata[12], Rotate$scores)
head(PCA_MLR_Data)

##   Satisfaction        RC1        RC2          RC3        RC4
## 1          8.2  0.1274910  0.7698686 -1.878446273  0.3664848
## 2          5.7  1.2216666 -1.6458617 -0.614030010  0.8130648
## 3          8.9  0.6158214  0.5800037  0.003689252  1.5699769
## 4          4.8 -0.8446267 -0.2719218  1.267493254 -1.2541645
## 5          7.1 -0.3197943 -0.8340650 -0.008096627  0.4475377
## 6          4.7 -0.6470292 -1.0672683 -1.303198892 -1.0527792
```

Naming the Factored Groups

Factors	Variables	Group Label	Short Description about Grouping
RC1	DelSpeed,CompRes,OrdBilling	Sales	These IVs mostly explains about the underlying Sales/Purchase factor.
RC2	SalesFImage,Ecom,Advertising	Brand Name	These IVs mostly explains about the underlying factors they impact the Product Brand Name.

RC3	WartyClaim,TechSup	Support	These IVs mostly explains about the underlying factors that contributes to Product Support.
RC4	ProdLine,ProdQual,CompPricing	Prod Segment	These IVs mostly explains about the underlying factors that segment the product.

```
names(PCA_MLR_Data) <- c("Satisfaction", "Sales", "Brand_Name",
                         "Support", "Prod_Segment")
head(PCA_MLR_Data)

##   Satisfaction      Sales Brand_Name      Support Prod_Segment
## 1          8.2  0.1274910  0.7698686 -1.878446273    0.3664848
## 2          5.7  1.2216666 -1.6458617 -0.614030010    0.8130648
## 3          8.9  0.6158214  0.5800037  0.003689252    1.5699769
## 4          4.8 -0.8446267 -0.2719218  1.267493254   -1.2541645
## 5          7.1 -0.3197943 -0.8340650 -0.008096627    0.4475377
## 6          4.7 -0.6470292 -1.0672683 -1.303198892   -1.0527792

PCA_MLR_Data_lm = lm(Satisfaction ~ ., data = PCA_MLR_Data[,-1])
summary(PCA_MLR_Data_lm)

##
## Call:
## lm(formula = Satisfaction ~ ., data = PCA_MLR_Data[, -1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6308 -0.4996  0.1372  0.4623  1.5228
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.91800    0.07089  97.589  < 2e-16 ***
## Sales         0.61805    0.07125   8.675 1.12e-13 ***
## Brand_Name    0.50973    0.07125   7.155 1.74e-10 ***
## Support       0.06714    0.07125   0.942    0.348
## Prod_Segment  0.54032    0.07125   7.584 2.24e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7089 on 95 degrees of freedom
## Multiple R-squared:  0.6605, Adjusted R-squared:  0.6462
## F-statistic: 46.21 on 4 and 95 DF,  p-value: < 2.2e-16
```

R-Squared is lower than our initial Model…

'Support' Factor is having low significance when compared to other Factors, so let's try to remove it and check if the model improves.

```
PCA_MLR_Data_lm1 = lm(Satisfaction ~ Sales+Brand_Name+Prod_Segment, data = PC
A_MLR_Data[,-1])
summary(PCA_MLR_Data_lm1)

##
## Call:
## lm(formula = Satisfaction ~ Sales + Brand_Name + Prod_Segment,
##     data = PCA_MLR_Data[, -1])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.69684 -0.49928  0.09364  0.46420  1.57638
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.91800    0.07085  97.646  < 2e-16 ***
## Sales         0.61805    0.07120   8.680 1.01e-13 ***
## Brand_Name    0.50973    0.07120   7.159 1.64e-10 ***
## Prod_Segment  0.54032    0.07120   7.588 2.09e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7085 on 96 degrees of freedom
## Multiple R-squared:  0.6574, Adjusted R-squared:  0.6466
## F-statistic: 61.39 on 3 and 96 DF,  p-value: < 2.2e-16
```

Unfortunately, we don't see any improvement in the model.

Conclusion: For the Factored model R-Squared value is only 66% and it is lesser than the initial Model, so it is fair to use the Original Model which has ~78% Adj R-Squared for Regression/Prediction rather than the Factored Model.

However, for knowing the explanatory power of each variable and to identify the underlying factor of independent variables it is required to run PCA.