**Advanced statistics Group Assignment**

**GROUP NO: 10**

**Group assignment submitted to : Great Lakes Institute of Management**

**Group assignment prepared by: Members: Sanjeev,Subhashree,Nandini,Archana, Swarna**

In total 116 respondents provided 235 observations of the 12 selected brands. How do you characterize the consideration behavior of the 12 selected brands? Analyze and interpret yoru results using factor analysis.

setwd("/Users/anand/Documents/BABI/AdvStats/Group Assignment-AS-May")  
  
#read the dataset & EDA  
cereal\_data <- read.csv("Dataset\_Cereal.csv")  
summary(cereal\_data)

## Cereals Filling Natural Fibre   
## CornFlakes :27 Min. :1.000 Min. :1.000 Min. :1.000   
## Weetabix :27 1st Qu.:3.000 1st Qu.:3.000 1st Qu.:3.000   
## Vitabrit :25 Median :4.000 Median :4.000 Median :4.000   
## NutriGrain :24 Mean :3.881 Mean :3.783 Mean :3.528   
## SpecialK :23 3rd Qu.:4.500 3rd Qu.:4.000 3rd Qu.:4.000   
## RiceBubbles:21 Max. :5.000 Max. :5.000 Max. :5.000   
## (Other) :88   
## Sweet Easy Salt Satisfying  
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :2   
## 1st Qu.:2.000 1st Qu.:4.000 1st Qu.:1.000 1st Qu.:3   
## Median :2.000 Median :5.000 Median :2.000 Median :4   
## Mean :2.506 Mean :4.528 Mean :1.991 Mean :4   
## 3rd Qu.:3.000 3rd Qu.:5.000 3rd Qu.:3.000 3rd Qu.:5   
## Max. :5.000 Max. :5.000 Max. :4.000 Max. :5   
##   
## Energy Fun Kids Soggy   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:3.000 1st Qu.:2.000 1st Qu.:3.000 1st Qu.:1.000   
## Median :4.000 Median :2.000 Median :4.000 Median :2.000   
## Mean :3.643 Mean :2.617 Mean :3.838 Mean :2.255   
## 3rd Qu.:4.000 3rd Qu.:3.000 3rd Qu.:5.000 3rd Qu.:3.000   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
##   
## Economical Health Family Calories   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:3.000 1st Qu.:3.000 1st Qu.:3.000 1st Qu.:2.000   
## Median :3.000 Median :4.000 Median :4.000 Median :3.000   
## Mean :3.217 Mean :3.809 Mean :3.872 Mean :2.702   
## 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:5.000 3rd Qu.:3.000   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
##   
## Plain Crisp Regular Sugar   
## Min. :1.000 Min. :1.0 Min. :1.000 Min. :1.000   
## 1st Qu.:1.000 1st Qu.:2.0 1st Qu.:2.000 1st Qu.:1.000   
## Median :2.000 Median :3.0 Median :3.000 Median :2.000   
## Mean :2.268 Mean :3.2 Mean :3.072 Mean :2.145   
## 3rd Qu.:3.000 3rd Qu.:4.0 3rd Qu.:4.000 3rd Qu.:3.000   
## Max. :5.000 Max. :5.0 Max. :5.000 Max. :5.000   
##   
## Fruit Process Quality Treat   
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:1.000 1st Qu.:2.000 1st Qu.:3.000 1st Qu.:2.000   
## Median :1.000 Median :3.000 Median :4.000 Median :3.000   
## Mean :1.694 Mean :2.932 Mean :3.694 Mean :2.626   
## 3rd Qu.:3.000 3rd Qu.:4.000 3rd Qu.:4.000 3rd Qu.:3.000   
## Max. :5.000 Max. :5.000 Max. :5.000 Max. :5.000   
##   
## Boring Nutritious   
## Min. :1.00 Min. :1.000   
## 1st Qu.:1.00 1st Qu.:3.000   
## Median :2.00 Median :4.000   
## Mean :1.83 Mean :3.664   
## 3rd Qu.:2.00 3rd Qu.:4.000   
## Max. :5.00 Max. :5.000   
##

str(cereal\_data)

## 'data.frame': 235 obs. of 26 variables:  
## $ Cereals : Factor w/ 12 levels "AllBran","CMuesli",..: 12 9 9 2 3 8 9 9 8 3 ...  
## $ Filling : int 5 1 5 5 4 4 4 4 4 4 ...  
## $ Natural : int 5 2 4 5 5 4 4 3 3 3 ...  
## $ Fibre : int 5 2 5 5 3 4 3 3 3 3 ...  
## $ Sweet : int 1 1 5 3 2 2 2 2 2 2 ...  
## $ Easy : int 2 5 5 5 5 5 5 5 5 5 ...  
## $ Salt : int 1 2 3 2 2 2 1 1 1 1 ...  
## $ Satisfying: int 5 5 5 5 5 5 5 5 5 5 ...  
## $ Energy : int 4 1 5 5 4 4 5 4 4 4 ...  
## $ Fun : int 1 1 5 5 5 5 5 4 4 4 ...  
## $ Kids : int 4 5 5 5 5 5 5 5 5 5 ...  
## $ Soggy : int 5 3 3 3 1 1 1 1 1 1 ...  
## $ Economical: int 5 5 3 3 5 5 5 3 3 3 ...  
## $ Health : int 5 2 5 5 5 4 5 4 4 4 ...  
## $ Family : int 5 5 5 5 3 5 5 5 5 5 ...  
## $ Calories : int 1 1 1 1 3 3 3 2 2 2 ...  
## $ Plain : int 3 5 1 1 1 1 1 3 3 3 ...  
## $ Crisp : int 1 5 5 1 5 5 5 4 4 4 ...  
## $ Regular : int 4 1 4 4 3 3 3 4 4 4 ...  
## $ Sugar : int 1 2 3 2 1 2 2 1 1 1 ...  
## $ Fruit : int 1 1 1 5 1 1 1 1 1 1 ...  
## $ Process : int 3 5 2 2 3 3 3 2 2 2 ...  
## $ Quality : int 5 2 5 5 5 5 5 4 4 4 ...  
## $ Treat : int 1 1 4 5 5 5 5 2 2 2 ...  
## $ Boring : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Nutritious: int 5 3 5 5 4 4 4 3 3 3 ...

head(cereal\_data)

## Cereals Filling Natural Fibre Sweet Easy Salt Satisfying Energy Fun  
## 1 Weetabix 5 5 5 1 2 1 5 4 1  
## 2 SpecialK 1 2 2 1 5 2 5 1 1  
## 3 SpecialK 5 4 5 5 5 3 5 5 5  
## 4 CMuesli 5 5 5 3 5 2 5 5 5  
## 5 CornFlakes 4 5 3 2 5 2 5 4 5  
## 6 RiceBubbles 4 4 4 2 5 2 5 4 5  
## Kids Soggy Economical Health Family Calories Plain Crisp Regular Sugar  
## 1 4 5 5 5 5 1 3 1 4 1  
## 2 5 3 5 2 5 1 5 5 1 2  
## 3 5 3 3 5 5 1 1 5 4 3  
## 4 5 3 3 5 5 1 1 1 4 2  
## 5 5 1 5 5 3 3 1 5 3 1  
## 6 5 1 5 4 5 3 1 5 3 2  
## Fruit Process Quality Treat Boring Nutritious  
## 1 1 3 5 1 1 5  
## 2 1 5 2 1 1 3  
## 3 1 2 5 4 1 5  
## 4 5 2 5 5 1 5  
## 5 1 3 5 5 1 4  
## 6 1 3 5 5 1 4

tail(cereal\_data)

## Cereals Filling Natural Fibre Sweet Easy Salt Satisfying Energy Fun  
## 230 PMuesli 4 4 4 3 4 2 4 3 2  
## 231 Weetabix 3 4 4 1 4 2 3 3 2  
## 232 PMuesli 5 4 4 3 4 3 4 4 4  
## 233 Weetabix 4 4 4 1 4 1 4 4 3  
## 234 SpecialK 3 3 3 3 4 2 3 3 2  
## 235 Weetabix 4 4 4 1 4 1 4 3 2  
## Kids Soggy Economical Health Family Calories Plain Crisp Regular Sugar  
## 230 3 2 3 4 3 3 3 2 4 2  
## 231 4 3 4 4 4 3 4 3 4 1  
## 232 4 1 3 4 4 4 1 4 4 3  
## 233 4 2 4 4 3 3 3 3 4 1  
## 234 3 2 3 4 3 2 3 2 3 2  
## 235 4 3 4 4 4 2 2 2 4 1  
## Fruit Process Quality Treat Boring Nutritious  
## 230 3 3 4 2 2 4  
## 231 1 3 4 2 2 4  
## 232 4 2 4 4 1 4  
## 233 1 2 3 3 2 4  
## 234 1 3 3 2 2 3  
## 235 1 2 4 2 3 4

#pre-requisites:  
install.packages("nFactors", repos = "http://cran.us.r-project.org")

##   
## The downloaded binary packages are in  
## /var/folders/zt/tzf46\_r937q1348y\_2qllthw0000gn/T//RtmpjRmOex/downloaded\_packages

library(nFactors)

## Loading required package: MASS

## Loading required package: psych

## Warning: package 'psych' was built under R version 3.4.4

## Loading required package: boot

##   
## Attaching package: 'boot'

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

## Loading required package: lattice

##   
## Attaching package: 'lattice'

## The following object is masked from 'package:boot':  
##   
## melanoma

##   
## Attaching package: 'nFactors'

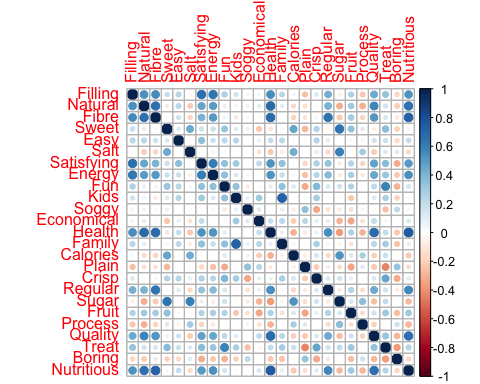
## The following object is masked from 'package:lattice':  
##   
## parallel

library(corrplot)

## Warning: package 'corrplot' was built under R version 3.4.2

## corrplot 0.84 loaded

#use cereal\_data as matrix  
cereal\_matrix <- as.matrix(cereal\_data[,2:26])  
  
  
#check for correlated attributes  
corrplot(cor(cereal\_matrix))



**Figure 1 Correlation Plot of 25 attributes**

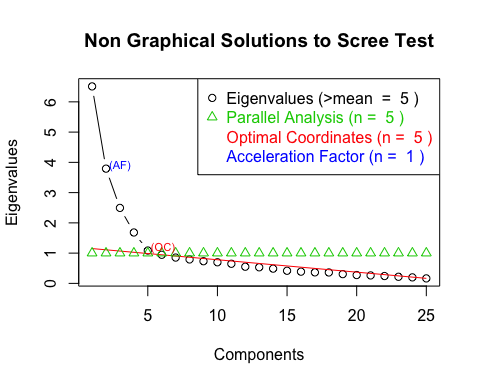
#To know # of factors - get eigen value & plot scree  
ev\_cereal <- eigen(cor(cereal\_matrix))  
ev\_cereal$values

## [1] 6.5104814 3.7921753 2.4942279 1.6821942 1.0856935 0.9450867 0.8532528  
## [8] 0.7910547 0.7326378 0.6977062 0.6481540 0.5507242 0.5314532 0.4874731  
## [15] 0.4168149 0.3869282 0.3640988 0.3608730 0.3061363 0.2755866 0.2628312  
## [22] 0.2428432 0.2183801 0.1986326 0.1645601

ev\_values <- nScree(x=ev\_cereal$values)  
ev\_values

## noc naf nparallel nkaiser  
## 1 5 1 5 5

plotnScree(ev\_values)



**Figure 2 Scree Plot showing 5 Optimal Factors**

#From above nScree, # of optimal factors will be 5  
  
#Perform FA for 5 factors and no rotation  
factor\_cereal <- factanal(~cereal\_matrix, 5, rotation = "none")  
factor\_cereal

##   
## Call:  
## factanal(x = ~cereal\_matrix, factors = 5, rotation = "none")  
##   
## Uniquenesses:  
## cereal\_matrixFilling cereal\_matrixNatural cereal\_matrixFibre   
## 0.286 0.389 0.312   
## cereal\_matrixSweet cereal\_matrixEasy cereal\_matrixSalt   
## 0.360 0.847 0.517   
## cereal\_matrixSatisfying cereal\_matrixEnergy cereal\_matrixFun   
## 0.369 0.431 0.525   
## cereal\_matrixKids cereal\_matrixSoggy cereal\_matrixEconomical   
## 0.239 0.775 0.709   
## cereal\_matrixHealth cereal\_matrixFamily cereal\_matrixCalories   
## 0.214 0.352 0.578   
## cereal\_matrixPlain cereal\_matrixCrisp cereal\_matrixRegular   
## 0.549 0.648 0.550   
## cereal\_matrixSugar cereal\_matrixFruit cereal\_matrixProcess   
## 0.199 0.562 0.764   
## cereal\_matrixQuality cereal\_matrixTreat cereal\_matrixBoring   
## 0.392 0.391 0.671   
## cereal\_matrixNutritious   
## 0.241   
##   
## Loadings:  
## Factor1 Factor2 Factor3 Factor4 Factor5  
## cereal\_matrixFilling 0.719 0.243 0.143 -0.338   
## cereal\_matrixNatural 0.763 -0.116   
## cereal\_matrixFibre 0.757 -0.113 -0.283 0.137   
## cereal\_matrixSweet 0.757 -0.244   
## cereal\_matrixEasy 0.289 0.188 0.168   
## cereal\_matrixSalt -0.258 0.495 -0.215 0.341   
## cereal\_matrixSatisfying 0.690 0.295 0.132 -0.205   
## cereal\_matrixEnergy 0.682 0.237 -0.204   
## cereal\_matrixFun 0.287 0.533 0.191 -0.266   
## cereal\_matrixKids 0.164 0.365 0.762 0.135   
## cereal\_matrixSoggy -0.177 0.143 0.413   
## cereal\_matrixEconomical 0.183 -0.172 0.440 0.143 0.117   
## cereal\_matrixHealth 0.854 -0.177 -0.112 0.110   
## cereal\_matrixFamily 0.260 0.310 0.694   
## cereal\_matrixCalories -0.237 0.536 -0.206 0.131 -0.134   
## cereal\_matrixPlain -0.232 -0.352 0.187 0.480   
## cereal\_matrixCrisp 0.197 0.450 0.172 -0.233 0.164   
## cereal\_matrixRegular 0.605 -0.195 0.196   
## cereal\_matrixSugar -0.356 0.716 -0.325 0.185 0.147   
## cereal\_matrixFruit 0.324 0.257 -0.447 -0.247   
## cereal\_matrixProcess -0.336 0.227 0.198 0.176   
## cereal\_matrixQuality 0.746 0.213   
## cereal\_matrixTreat 0.368 0.589 -0.328 0.124   
## cereal\_matrixBoring -0.314 -0.283 -0.114 0.370   
## cereal\_matrixNutritious 0.834 -0.142 0.170   
##   
## Factor1 Factor2 Factor3 Factor4 Factor5  
## SS loadings 6.049 3.368 2.062 1.165 0.486  
## Proportion Var 0.242 0.135 0.082 0.047 0.019  
## Cumulative Var 0.242 0.377 0.459 0.506 0.525  
##   
## Test of the hypothesis that 5 factors are sufficient.  
## The chi square statistic is 318.9 on 185 degrees of freedom.  
## The p-value is 3.48e-09

#without rotation, there is no clear distinguish between the factors  
  
#try varimax / promax rotation to improve Factor 5 loadings - Promax seems to give better results.  
factor\_cereal\_pro <- factanal(~cereal\_matrix, 5, rotation = "promax")  
factor\_cereal\_pro

##   
## Call:  
## factanal(x = ~cereal\_matrix, factors = 5, rotation = "promax")  
##   
## Uniquenesses:  
## cereal\_matrixFilling cereal\_matrixNatural cereal\_matrixFibre   
## 0.286 0.389 0.312   
## cereal\_matrixSweet cereal\_matrixEasy cereal\_matrixSalt   
## 0.360 0.847 0.517   
## cereal\_matrixSatisfying cereal\_matrixEnergy cereal\_matrixFun   
## 0.369 0.431 0.525   
## cereal\_matrixKids cereal\_matrixSoggy cereal\_matrixEconomical   
## 0.239 0.775 0.709   
## cereal\_matrixHealth cereal\_matrixFamily cereal\_matrixCalories   
## 0.214 0.352 0.578   
## cereal\_matrixPlain cereal\_matrixCrisp cereal\_matrixRegular   
## 0.549 0.648 0.550   
## cereal\_matrixSugar cereal\_matrixFruit cereal\_matrixProcess   
## 0.199 0.562 0.764   
## cereal\_matrixQuality cereal\_matrixTreat cereal\_matrixBoring   
## 0.392 0.391 0.671   
## cereal\_matrixNutritious   
## 0.241   
##   
## Loadings:  
## Factor1 Factor2 Factor3 Factor4 Factor5  
## cereal\_matrixFilling 0.396 0.631   
## cereal\_matrixNatural 0.649 -0.131 0.234   
## cereal\_matrixFibre 0.834 -0.196 0.125   
## cereal\_matrixSweet 0.252 0.629 0.179   
## cereal\_matrixEasy 0.171 0.259 0.123   
## cereal\_matrixSalt -0.159 0.737   
## cereal\_matrixSatisfying 0.355 0.283 0.464   
## cereal\_matrixEnergy 0.411 0.438   
## cereal\_matrixFun 0.556 0.298   
## cereal\_matrixKids -0.129 0.877   
## cereal\_matrixSoggy 0.126 -0.539 0.184 0.124   
## cereal\_matrixEconomical 0.139 -0.182 -0.175 0.419   
## cereal\_matrixHealth 0.832 -0.112   
## cereal\_matrixFamily 0.116 0.779   
## cereal\_matrixCalories -0.187 0.503 0.211   
## cereal\_matrixPlain 0.101 -0.676 0.182   
## cereal\_matrixCrisp 0.509 0.139 0.264 -0.145   
## cereal\_matrixRegular 0.735 0.133 -0.105   
## cereal\_matrixSugar 0.109 0.875 -0.105   
## cereal\_matrixFruit 0.219 0.398 0.113 -0.383 0.133   
## cereal\_matrixProcess -0.108 0.444 -0.197   
## cereal\_matrixQuality 0.681 0.207 -0.100   
## cereal\_matrixTreat 0.105 0.683 0.175 0.185   
## cereal\_matrixBoring -0.563 0.130 -0.122   
## cereal\_matrixNutritious 0.885   
##   
## Factor1 Factor2 Factor3 Factor4 Factor5  
## SS loadings 4.251 2.486 2.349 2.206 1.120  
## Proportion Var 0.170 0.099 0.094 0.088 0.045  
## Cumulative Var 0.170 0.269 0.363 0.452 0.496  
##   
## Factor Correlations:  
## Factor1 Factor2 Factor3 Factor4 Factor5  
## Factor1 1.000 -0.3448 -0.1863 0.320 -0.3572  
## Factor2 -0.345 1.0000 0.0566 0.181 -0.1944  
## Factor3 -0.186 0.0566 1.0000 -0.152 0.0585  
## Factor4 0.320 0.1813 -0.1516 1.000 -0.4897  
## Factor5 -0.357 -0.1944 0.0585 -0.490 1.0000  
##   
## Test of the hypothesis that 5 factors are sufficient.  
## The chi square statistic is 318.9 on 185 degrees of freedom.  
## The p-value is 3.48e-09

#consider only loadings above 0.4 to consolidate the variables  
print(factor\_cereal\_pro, digits = 2, cutoff = 0.4)  
## Call:  
## factanal(x = ~cereal\_matrix, factors = 5, rotation = "promax")  
##   
## Uniquenesses:  
## cereal\_matrixFilling cereal\_matrixNatural cereal\_matrixFibre   
## 0.29 0.39 0.31   
## cereal\_matrixSweet cereal\_matrixEasy cereal\_matrixSalt   
## 0.36 0.85 0.52   
## cereal\_matrixSatisfying cereal\_matrixEnergy cereal\_matrixFun   
## 0.37 0.43 0.53   
## cereal\_matrixKids cereal\_matrixSoggy cereal\_matrixEconomical   
## 0.24 0.77 0.71   
## cereal\_matrixHealth cereal\_matrixFamily cereal\_matrixCalories   
## 0.21 0.35 0.58   
## cereal\_matrixPlain cereal\_matrixCrisp cereal\_matrixRegular   
## 0.55 0.65 0.55   
## cereal\_matrixSugar cereal\_matrixFruit cereal\_matrixProcess   
## 0.20 0.56 0.76   
## cereal\_matrixQuality cereal\_matrixTreat cereal\_matrixBoring   
## 0.39 0.39 0.67   
## cereal\_matrixNutritious   
## 0.24   
##   
## Loadings:  
## Factor1 Factor2 Factor3 Factor4 Factor5  
## cereal\_matrixFilling 0.63   
## cereal\_matrixNatural 0.65   
## cereal\_matrixFibre 0.83   
## cereal\_matrixSweet 0.63   
## cereal\_matrixEasy   
## cereal\_matrixSalt 0.74   
## cereal\_matrixSatisfying 0.46   
## cereal\_matrixEnergy 0.41 0.44   
## cereal\_matrixFun 0.56   
## cereal\_matrixKids 0.88   
## cereal\_matrixSoggy -0.54   
## cereal\_matrixEconomical 0.42   
## cereal\_matrixHealth 0.83   
## cereal\_matrixFamily 0.78   
## cereal\_matrixCalories 0.50   
## cereal\_matrixPlain -0.68   
## cereal\_matrixCrisp 0.51   
## cereal\_matrixRegular 0.74   
## cereal\_matrixSugar 0.87   
## cereal\_matrixFruit   
## cereal\_matrixProcess 0.44   
## cereal\_matrixQuality 0.68   
## cereal\_matrixTreat 0.68   
## cereal\_matrixBoring -0.56   
## cereal\_matrixNutritious 0.89   
##   
## Factor1 Factor2 Factor3 Factor4 Factor5  
## SS loadings 4.25 2.49 2.35 2.21 1.12  
## Proportion Var 0.17 0.10 0.09 0.09 0.04  
## Cumulative Var 0.17 0.27 0.36 0.45 0.50  
##   
## Factor Correlations:  
## Factor1 Factor2 Factor3 Factor4 Factor5  
## Factor1 1.00 -0.345 -0.186 0.32 -0.357  
## Factor2 -0.34 1.000 0.057 0.18 -0.194  
## Factor3 -0.19 0.057 1.000 -0.15 0.059  
## Factor4 0.32 0.181 -0.152 1.00 -0.490  
## Factor5 -0.36 -0.194 0.059 -0.49 1.000  
##   
## Test of the hypothesis that 5 factors are sufficient.  
## The chi square statistic is 318.9 on 185 degrees of freedom.  
## The p-value is 3.48e-09

#loading factors  
load1 <- factor\_cereal\_pro$loadings[,1:5]  
load1

## Factor1 Factor2 Factor3 Factor4  
## cereal\_matrixFilling 0.39564587 -0.056530950 0.003536483 0.09463231  
## cereal\_matrixNatural 0.64907639 -0.056645487 -0.131045410 -0.03188932  
## cereal\_matrixFibre 0.83402789 -0.092748602 0.051952562 -0.19608245  
## cereal\_matrixSweet -0.02248928 0.252156299 0.628549354 0.01350862  
## cereal\_matrixEasy 0.17123388 0.024710576 0.063830507 0.25909725  
## cereal\_matrixSalt 0.07475711 -0.159248635 0.736971724 0.02695003  
## cereal\_matrixSatisfying 0.35543419 0.047206275 0.000609941 0.28301719  
## cereal\_matrixEnergy 0.41089822 0.080148898 0.012115408 0.06549895  
## cereal\_matrixFun -0.03755982 0.556121433 0.084673575 0.29757104  
## cereal\_matrixKids -0.12892164 0.032956420 0.014825904 0.87671907  
## cereal\_matrixSoggy 0.12583292 -0.538514625 0.075514469 0.18353335  
## cereal\_matrixEconomical 0.13896283 -0.181817328 -0.174626498 0.41927935  
## cereal\_matrixHealth 0.83200123 0.006011534 -0.112364552 -0.06941150  
## cereal\_matrixFamily -0.07957528 0.116046697 -0.076646279 0.77881683  
## cereal\_matrixCalories -0.18748288 0.028907048 0.502681459 -0.02142407  
## cereal\_matrixPlain 0.10076528 -0.676324678 0.073995532 0.18158524  
## cereal\_matrixCrisp 0.01896621 0.509441291 0.139291695 0.26449002  
## cereal\_matrixRegular 0.73548918 0.026288467 0.133359052 -0.10490554  
## cereal\_matrixSugar -0.02698528 0.109023700 0.874675134 -0.05414197  
## cereal\_matrixFruit 0.21922797 0.398226253 0.112753132 -0.38269691  
## cereal\_matrixProcess -0.04117556 -0.108171840 0.444012181 0.06390152  
## cereal\_matrixQuality 0.68073436 0.206890591 -0.075304637 0.08236408  
## cereal\_matrixTreat 0.10454054 0.682693631 0.174612704 0.18531981  
## cereal\_matrixBoring 0.01710269 -0.562797648 0.130238024 -0.12207322  
## cereal\_matrixNutritious 0.88515199 0.009875005 0.030908647 -0.05324240  
## Factor5  
## cereal\_matrixFilling 0.630603305  
## cereal\_matrixNatural 0.234246520  
## cereal\_matrixFibre 0.124696011  
## cereal\_matrixSweet 0.178527265  
## cereal\_matrixEasy 0.122798759  
## cereal\_matrixSalt -0.005889574  
## cereal\_matrixSatisfying 0.464081311  
## cereal\_matrixEnergy 0.437820799  
## cereal\_matrixFun 0.031481001  
## cereal\_matrixKids 0.082629734  
## cereal\_matrixSoggy 0.123863128  
## cereal\_matrixEconomical -0.075397880  
## cereal\_matrixHealth 0.052357369  
## cereal\_matrixFamily 0.089212339  
## cereal\_matrixCalories 0.211466134  
## cereal\_matrixPlain -0.071475131  
## cereal\_matrixCrisp -0.144983078  
## cereal\_matrixRegular -0.074115273  
## cereal\_matrixSugar -0.104515511  
## cereal\_matrixFruit 0.132582508  
## cereal\_matrixProcess -0.196824551  
## cereal\_matrixQuality -0.100383994  
## cereal\_matrixTreat -0.066839243  
## cereal\_matrixBoring 0.015382412  
## cereal\_matrixNutritious 0.003124129

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Health factor | Feel factor | Sugar factor | Economy factor | Energy factor |
| Filling |  |  |  |  | 0.63 |
| Natural | 0.65 |  |  |  |  |
| Fibre | 0.83 |  |  |  |  |
| Sweet |  |  | 0.63 |  |  |
| Easy |  |  |  | 0.36 |  |
| Salt |  |  | 0.74 |  |  |
| Satisfying |  |  |  |  | 0.46 |
| Energy | 0.41 |  |  |  | 0.44 |
| Fun |  | 0.56 |  |  |  |
| Kids |  |  |  | 0.88 |  |
| Soggy |  | 0.54 |  |  |  |
| Economical |  |  |  | 0.42 |  |
| Health | 0.83 |  |  |  |  |
| Family |  |  |  | 0.78 |  |
| Calories |  |  | 0.5 |  |  |
| Plain |  | 0.68 |  |  |  |
| Crisp |  | 0.51 |  |  |  |
| Regular | 0.74 |  |  |  |  |
| Sugar |  |  | 0.87 |  |  |
| Fruit |  | 0.4 |  |  |  |
| Process |  |  | 0.44 |  |  |
| Quality | 0.68 |  |  |  |  |
| Treat |  | 0.68 |  |  |  |
| Boring |  | 0.56 |  |  |  |
| Nutritious | 0.89 |  |  |  |  |

**Figure 3 Visual Representation of Factor Loadings**

**Conclusion:**

The provided dataset (Dataset\_Cereal.csv) had 12 Cereal brands evaluated based on 5-point Likert scale on 25 attributes. These 25 attributes were characterized into 5 factors using the ‘Factor Analysis’ dimension reduction technique.

The below list shows the grouping of the 25 attributes in each factor.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Factor 1** | **Factor 2** | **Factor 3** | **Factor 4** | **Factor 5** |
| Natural | Fun | Sweet | Easy | Filling |
| Fibre | Soggy | Salt | Kids | Satisfying |
| Health | Plain | Calories | Economical | Energy |
| Regular | Crisp | Sugar | Family |  |
| Quality | Fruit | Process |  |  |
| Nutritious | Treat |  |  |  |
|  | Boring |  |  |  |

Further, from the grouped attributes, these 5 factors could be named as:

**Health factor, Feel factor, Sugar factor, Economy factor, Energy factor.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Health factor** | **Feel**  **factor** | **Sugar factor** | **Economy factor** | **Energy factor** |
| Natural | Fun | Sweet | Easy | Filling |
| Fibre | Soggy | Salt | Kids | Satisfying |
| Health | Plain | Calories | Economical | Energy |
| Regular | Crisp | Sugar | Family |  |
| Quality | Fruit | Process |  |  |
| Nutritious | Treat |  |  |  |
|  | Boring |  |  |  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. What is the nature of each of the variables? Which variable is dependent variable and what are the independent variables in the model?

2. Check whether the variables require any transformation individually

3. Set up a regression equation, run the model and discuss your results

We begin by importing the data to be evaluated.

base\_data = read.csv("C:/Users/ssripathi/Desktop/EMBA Leslie Salt/basedata.csv")

Once imported, we ensure the data follows the structure we wish to have, especially in terms of factor levels.

base\_data$Flood <- factor(base\_data$Flood,levels=c("0","1"),labels=c("No", "Yes"))

base\_data$County <- factor(base\_data$County,levels=c("0","1"),labels=c("San Mateo", "Santa Clara"))

The data can now be viewed to understand the status. This explains Question 1, on the nature of each variable. Price is our dependent variable, and all the other variables are independent. We have not needed to transform any variables, but we have labeled the factors. This is part of Question 2.

summary(base\_data)

## Price County Size Elevation

## Min. : 1.70 San Mateo :12 Min. : 6.90 Min. : 0.000

## 1st Qu.: 5.35 Santa Clara:19 1st Qu.: 20.35 1st Qu.: 2.000

## Median :11.70 Median : 51.40 Median : 4.000

## Mean :11.95 Mean : 139.97 Mean : 4.645

## 3rd Qu.:16.05 3rd Qu.: 104.10 3rd Qu.: 7.000

## Max. :37.20 Max. :1695.20 Max. :20.000

## Sewer Date Flood Distance

## Min. : 0 Min. :-103.00 No :26 Min. : 0.000

## 1st Qu.: 0 1st Qu.: -63.50 Yes: 5 1st Qu.: 0.850

## Median : 900 Median : -59.00 Median : 4.900

## Mean : 1981 Mean : -58.65 Mean : 5.132

## 3rd Qu.: 3450 3rd Qu.: -51.00 3rd Qu.: 5.500

## Max. :10000 Max. : -4.00 Max. :16.500

Let’s follow this with a simple regression model. This begins Question 3.

reg\_model1 = lm(Price~.,data = base\_data)

summary(reg\_model1)

##

## Call:

## lm(formula = Price ~ ., data = base\_data)

##

## Residuals:

## Min 1Q Median 3Q Max

## -5.169 -2.957 -0.256 2.070 13.031

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) 2.364e+01 3.829e+00 6.174 2.68e-06 \*\*\*

## CountySanta Clara -8.789e+00 3.652e+00 -2.407 0.024532 \*

## Size -6.043e-03 3.501e-03 -1.726 0.097702 .

## Elevation 5.193e-01 2.386e-01 2.177 0.040030 \*

## Sewer -9.573e-04 4.169e-04 -2.296 0.031126 \*

## Date 8.508e-02 4.865e-02 1.749 0.093646 .

## FloodYes -1.202e+01 2.989e+00 -4.020 0.000536 \*\*\*

## Distance 1.858e-01 3.395e-01 0.547 0.589386

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## Residual standard error: 4.431 on 23 degrees of freedom

## Multiple R-squared: 0.747, Adjusted R-squared: 0.67

## F-statistic: 9.703 on 7 and 23 DF, p-value: 1.351e-05

The R squared is not great, but the variables seem reasonably significant in the equation. We run some quick tests to evaluate the model we built. We begin with VIF.

library(car)

## Warning: package 'car' was built under R version 3.5.3

## Loading required package: carData

vif(reg\_model1)

## County Size Elevation Sewer Date Flood Distance

## 4.995597 2.003925 1.649759 1.635122 2.174889 1.907942 3.623612

The VIF here for each variable is below 5, which tells us that there is no signficant multicollinearity.

Unfortunately the simple model above excludes interaction factors entirely. Now that we’ve established independence, let’s try and include some interaction factors.

reg\_model = lm(Price~County+Size+Elevation+Sewer+Date+Flood+Distance+County\*Size+County\*Elevation+County\*Sewer+County\*Date+County\*Flood+County\*Distance+Size\*Elevation+Size\*Sewer+Size\*Date+Size\*Flood+Size\*Distance+Elevation\*Sewer+Elevation\*Date+Elevation\*Flood+Elevation\*Distance+Sewer\*Date+Sewer\*Flood+Sewer\*Distance+Date\*Flood+Date\*Distance+Flood\*Distance,data = base\_data)

summary(reg\_model)

##

## Call:

## lm(formula = Price ~ County + Size + Elevation + Sewer + Date +

## Flood + Distance + County \* Size + County \* Elevation + County \*

## Sewer + County \* Date + County \* Flood + County \* Distance +

## Size \* Elevation + Size \* Sewer + Size \* Date + Size \* Flood +

## Size \* Distance + Elevation \* Sewer + Elevation \* Date +

## Elevation \* Flood + Elevation \* Distance + Sewer \* Date +

## Sewer \* Flood + Sewer \* Distance + Date \* Flood + Date \*

## Distance + Flood \* Distance, data = base\_data)

##

## Residuals:

## 1 2 3 4 5 6

## 1.065e-02 -4.775e-04 -1.952e-16 4.782e-05 -3.960e-16 2.981e-02

## 7 8 9 10 11 12

## -2.051e-01 4.284e-01 1.270e-01 -1.752e+00 1.864e+00 -1.795e-01

## 13 14 15 16 17 18

## -7.204e-02 -2.099e+00 1.058e+00 -2.228e+00 3.080e-02 1.353e+00

## 19 20 21 22 23 24

## 9.813e-02 1.296e+00 -2.060e-03 1.091e+00 -8.505e-01 3.456e-16

## 25 26 27 28 29 30

## -3.318e-16 -6.204e-04 -1.062e-03 4.959e-03 3.253e-04 5.161e-16

## 31

## -1.590e-03

##

## Coefficients: (2 not defined because of singularities)

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) 1.027e+02 1.167e+02 0.880 0.428

## CountySanta Clara -2.674e+02 3.277e+02 -0.816 0.460

## Size 1.787e+00 2.847e+00 0.628 0.564

## Elevation 3.540e+01 4.226e+01 0.838 0.449

## Sewer -6.074e-02 8.199e-02 -0.741 0.500

## Date -2.540e+00 4.171e+00 -0.609 0.575

## FloodYes 1.851e+02 2.830e+02 0.654 0.549

## Distance -3.283e+01 4.719e+01 -0.696 0.525

## CountySanta Clara:Size -3.698e-01 7.070e-01 -0.523 0.629

## CountySanta Clara:Elevation -2.031e+01 3.491e+01 -0.582 0.592

## CountySanta Clara:Sewer 5.537e-02 7.598e-02 0.729 0.507

## CountySanta Clara:Date -1.251e-01 1.471e+00 -0.085 0.936

## CountySanta Clara:FloodYes NA NA NA NA

## CountySanta Clara:Distance 5.064e+01 7.194e+01 0.704 0.520

## Size:Elevation 1.285e-02 1.013e-02 1.268 0.273

## Size:Sewer 2.966e-04 3.660e-04 0.810 0.463

## Size:Date 2.207e-02 3.348e-02 0.659 0.546

## Size:FloodYes 1.117e+00 1.424e+00 0.785 0.476

## Size:Distance -4.356e-02 5.399e-02 -0.807 0.465

## Elevation:Sewer -1.739e-04 3.447e-04 -0.504 0.640

## Elevation:Date 2.487e-01 2.110e-01 1.178 0.304

## Elevation:FloodYes -1.931e+02 2.805e+02 -0.689 0.529

## Elevation:Distance -3.113e-01 1.907e-01 -1.632 0.178

## Sewer:Date 1.019e-05 8.978e-05 0.114 0.915

## Sewer:FloodYes 7.503e-02 1.208e-01 0.621 0.568

## Sewer:Distance 5.980e-03 9.128e-03 0.655 0.548

## Date:FloodYes 2.161e+00 2.157e+00 1.002 0.373

## Date:Distance 1.982e-01 4.810e-01 0.412 0.701

## FloodYes:Distance NA NA NA NA

##

## Residual standard error: 2.385 on 4 degrees of freedom

## Multiple R-squared: 0.9873, Adjusted R-squared: 0.9044

## F-statistic: 11.92 on 26 and 4 DF, p-value: 0.01334

The R squared is much better but almost none of the variables are significant. Clearly we have too many of them and need to strip away the unnecessary elements. Lets run a stepwise regression and eliminate unnecessary elements based on their AIC.

library(MASS)

step.model = stepAIC(reg\_model,direction = "both",trace = FALSE)

summary(step.model)

##

## Call:

## lm(formula = Price ~ County + Size + Elevation + Sewer + Date +

## Flood + Distance + County:Sewer + County:Distance + Size:Elevation +

## Size:Sewer + Size:Date + Size:Flood + Size:Distance + Elevation:Date +

## Elevation:Flood + Elevation:Distance + Sewer:Flood + Sewer:Distance +

## Date:Flood, data = base\_data)

##

## Residuals:

## Min 1Q Median 3Q Max

## -2.44798 -0.09014 0.00000 0.08836 1.86266

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) 3.773e+01 1.242e+01 3.038 0.012516 \*

## CountySanta Clara -8.925e+01 2.811e+01 -3.176 0.009894 \*\*

## Size 1.501e-01 1.473e-01 1.019 0.332009

## Elevation 1.134e+01 2.320e+00 4.885 0.000637 \*\*\*

## Sewer -1.381e-02 4.233e-03 -3.264 0.008523 \*\*

## Date -8.740e-01 2.345e-01 -3.727 0.003927 \*\*

## FloodYes 2.312e+01 1.805e+01 1.281 0.229233

## Distance -5.874e+00 3.227e+00 -1.821 0.098690 .

## CountySanta Clara:Sewer 1.194e-02 3.963e-03 3.013 0.013049 \*

## CountySanta Clara:Distance 1.036e+01 3.951e+00 2.621 0.025538 \*

## Size:Elevation 1.480e-02 3.942e-03 3.756 0.003750 \*\*

## Size:Sewer 1.008e-04 3.408e-05 2.957 0.014367 \*

## Size:Date 2.876e-03 1.769e-03 1.626 0.135053

## Size:FloodYes 3.484e-01 1.142e-01 3.050 0.012256 \*

## Size:Distance -1.824e-02 7.417e-03 -2.459 0.033754 \*

## Elevation:Date 1.786e-01 3.668e-02 4.869 0.000652 \*\*\*

## Elevation:FloodYes -3.268e+01 1.530e+01 -2.135 0.058487 .

## Elevation:Distance -3.871e-01 7.591e-02 -5.100 0.000464 \*\*\*

## Sewer:FloodYes 1.273e-02 4.655e-03 2.734 0.021033 \*

## Sewer:Distance 7.327e-04 4.441e-04 1.650 0.129986

## Date:FloodYes 1.083e+00 2.671e-01 4.054 0.002309 \*\*

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## Residual standard error: 1.576 on 10 degrees of freedom

## Multiple R-squared: 0.9861, Adjusted R-squared: 0.9582

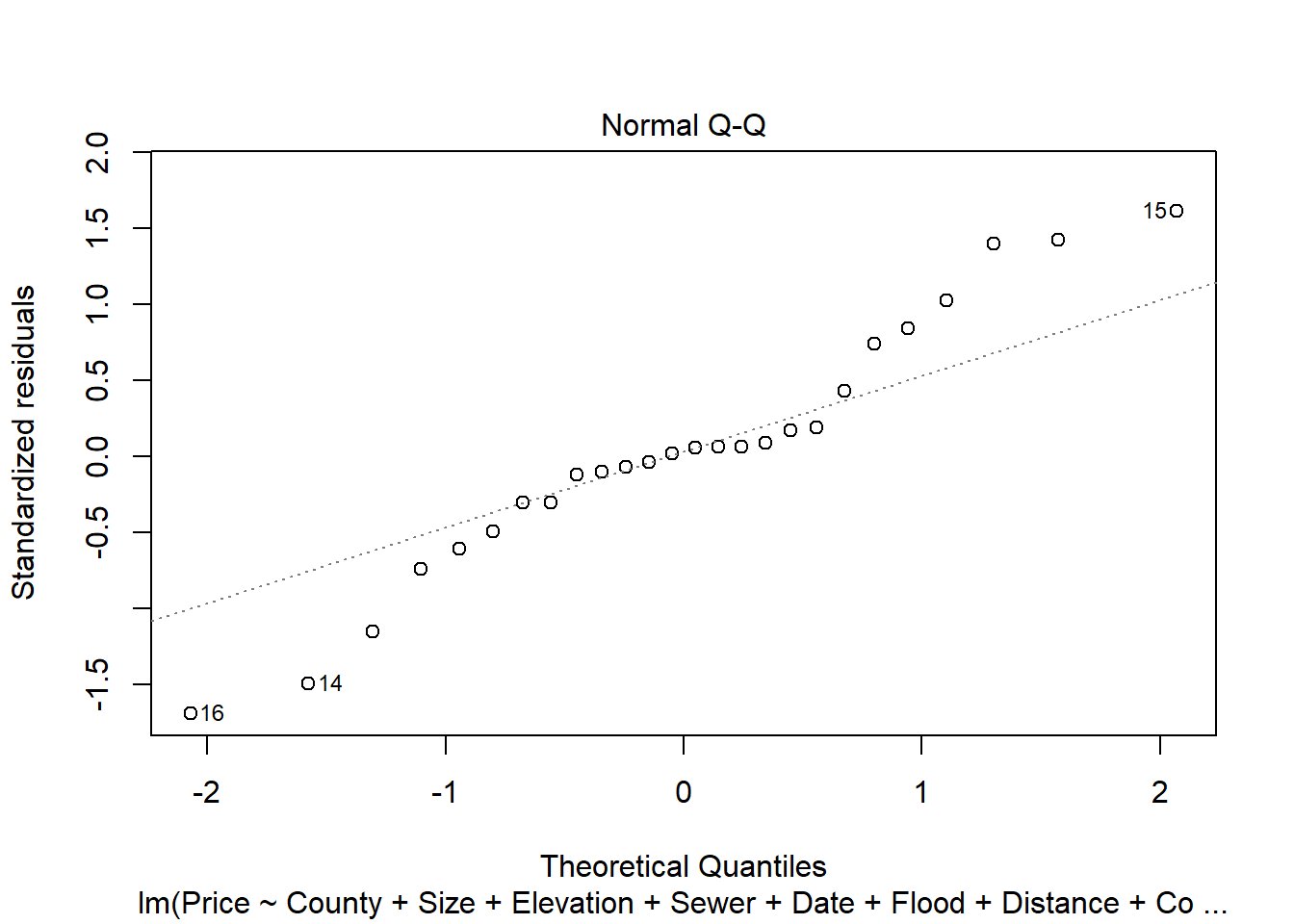
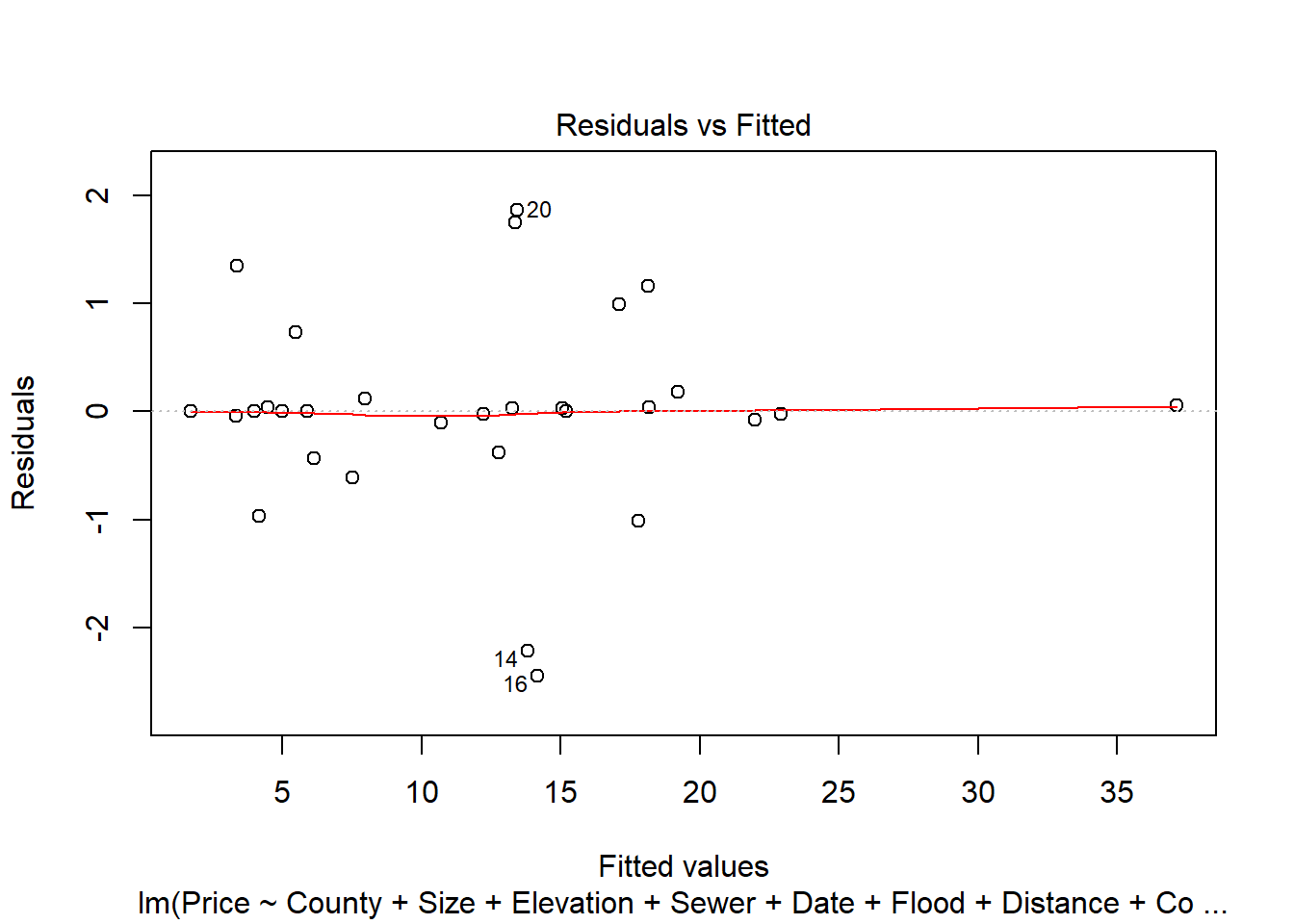
## F-statistic: 35.42 on 20 and 10 DF, p-value: 9.417e-07

The R squared is much better here as is the significance of each variable. Lets use this model to continue our work.

plot(step.model)

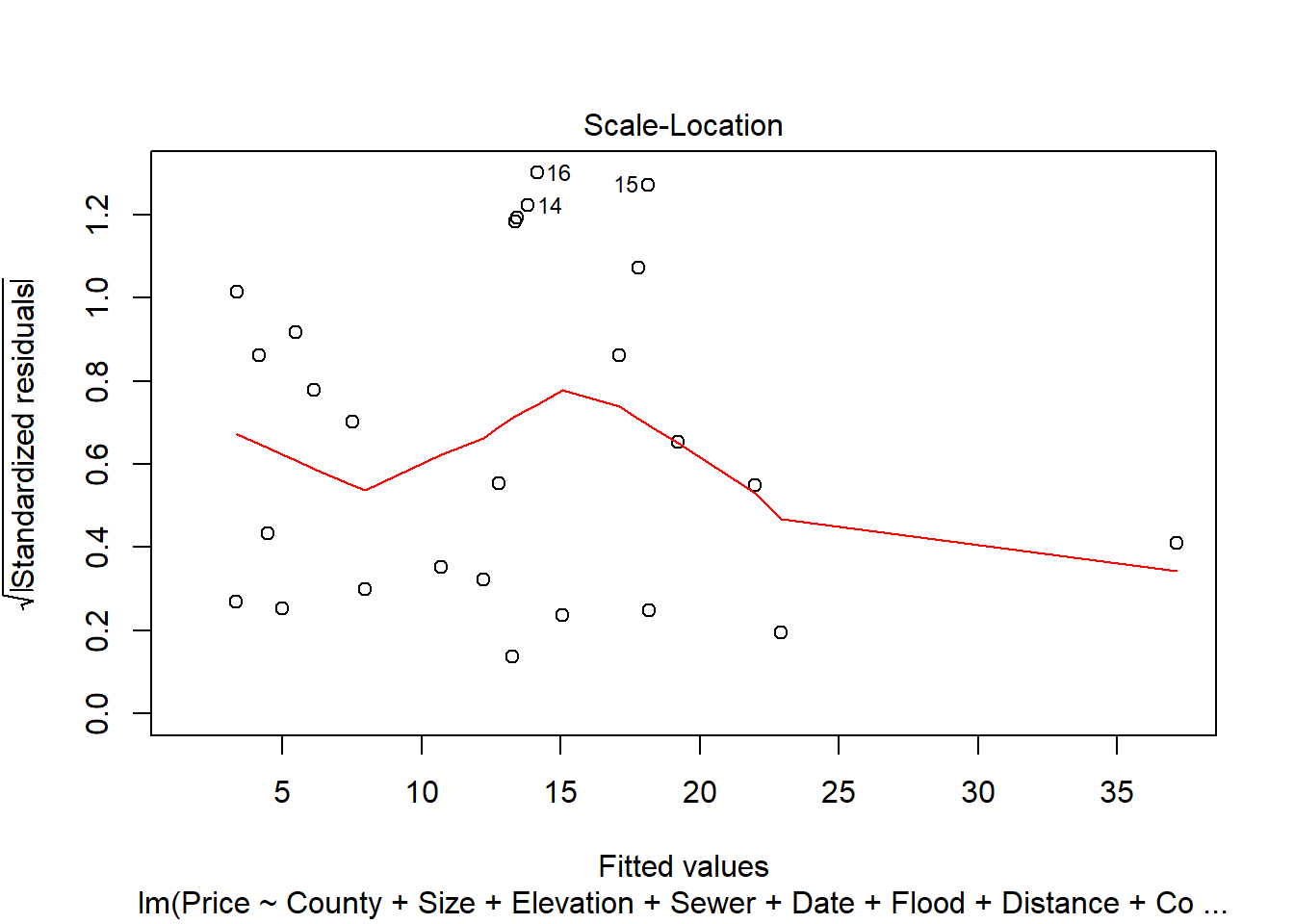
## Warning: not plotting observations with leverage one:

## 3, 5, 24, 25, 30



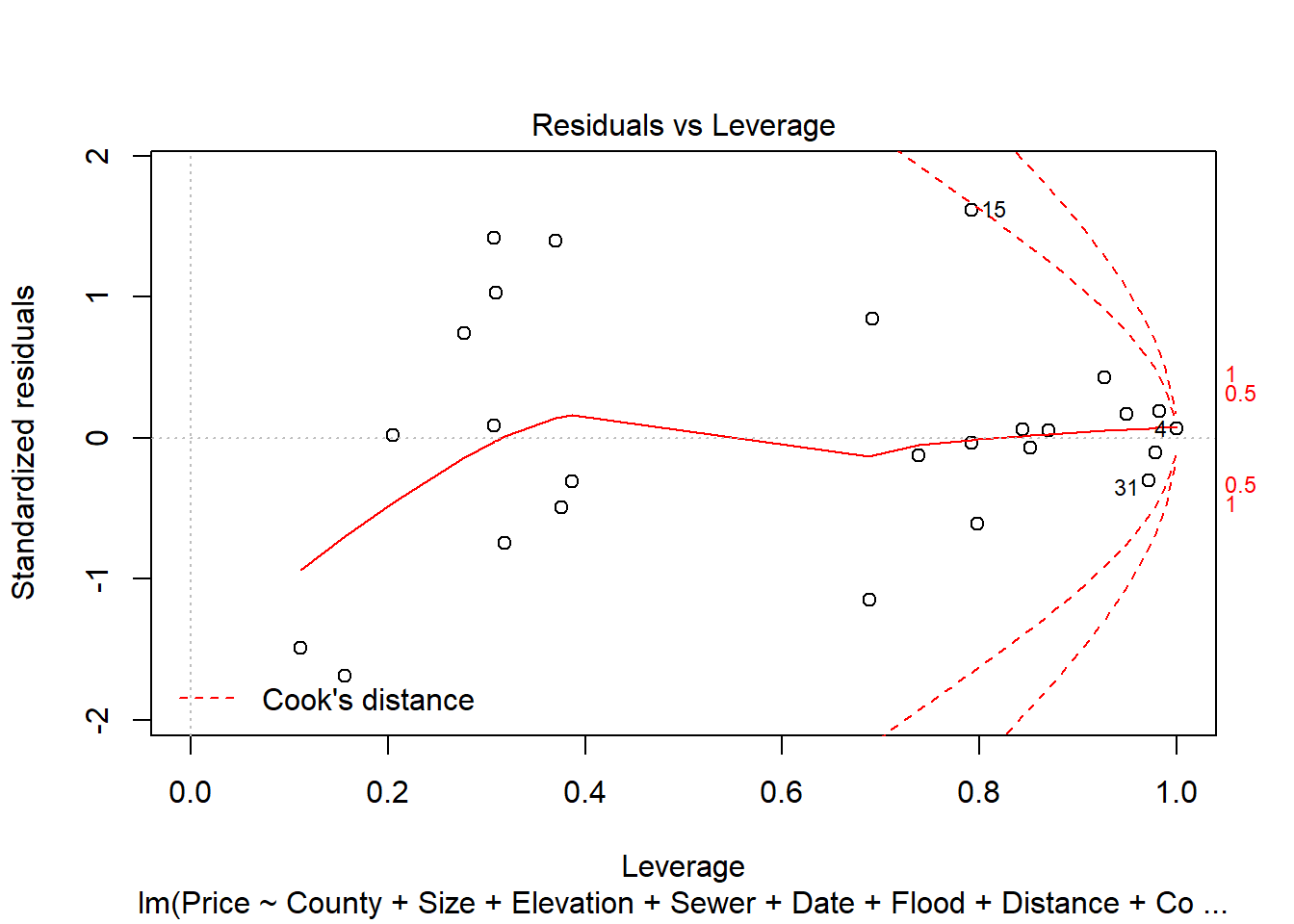
## Warning: not plotting observations with leverage one:

## 3, 5, 24, 25, 30



## Warning in sqrt(crit \* p \* (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit \* p \* (1 - hh)/hh): NaNs produced



Plotting our model does not reveal any major concerns, outside of a slight leftward drift in the residuals.

Lets apply some more evaluations to our model, to rule out additional concerns. First, the Durbin-Watson test.

durbinWatsonTest(step.model)

## lag Autocorrelation D-W Statistic p-value

## 1 -0.4291724 2.858023 0.188

## Alternative hypothesis: rho != 0

While a value of 2 would be ideal, 2.8 is still not too bad. Let us move on and try the Breusch-Pagan test.

library(lmtest)

## Warning: package 'lmtest' was built under R version 3.5.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.5.3

##

## Attaching package: 'zoo'

## The following objects are masked from 'package:base':

##

## as.Date, as.Date.numeric

bptest(step.model)

##

## studentized Breusch-Pagan test

##

## data: step.model

## BP = 14.763, df = 20, p-value = 0.7898

A non-significant p value here indicates a lack of homoscedasticity, which is desirable.

Now that we’ve established the validity of our results, let us examine the predicted values. We predict and list all values considering the unknown factors here, the status of flooding in the area and the time from prediction to actual sale. A time range of 6 months is considered.

predict\_data = data.frame (Price=c(0,0),County=c("Santa Clara","Santa Clara"),Size=c(246.8,246.8),Elevation=c(0,0),Sewer = c(0,0),Date=c(0,0),Flood=c("Yes","No"),Distance = c(0,0))

price\_output = data.frame(Price=double(),Flood\_Estimation = character(),Time\_In\_Months = integer())

for (i in floodlevels){

predict\_data$Flood=i

for (j in 0:6){

predict\_data$Date=j

estimated\_price = predict(step.model,newdata = predict\_data[0,])

price\_output = rbind(price\_output,c(estimated\_price,i,j))

}}

price\_output

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Problem 3 (10 points) All Greens Franchise Explain the importance of X2, X3, X4, X5, X6 on Annual Net Sales, X1. The data (X1, X2, X3, X4, X5, X6) are for each franchise store. X1 = annual net sales/$1000 X2 = number sq. ft./1000 X3 = inventory/$1000 X4 = amount spent on advertising/$1000 X5 = size of sales district/1000 families X6 = number of competing stores in district

**Solution:**

Null hypothesis

Ho: There is no significant linear relationship between the dependent variable (X1 = Annual Net Sales) and independent variables (other X variables) i.e., the slope is equal to zero

Alternative hypothesis

H1: There is significant linear relationship between the dependent variable (Y = Annual Net Sales) and independent variables (other X variables) i.e., the slope is not equal to zero

A simple linear regression model shall be used for this purpose.

|  |  |  |
| --- | --- | --- |
| ANNUAL NET SALES  X1 = annual net sales/$1000 | *Regressed with*  *↔* | LIST OF INDEPENDENT VARIABLES |
| X2 = number sq. ft./1000 |
| X3 = inventory/$1000 |
| X4 = amount spent on advertising/$1000 |
| X5 = size of sales district/1000 families |
| X6 = number of competing stores in district |

Regression equation

[ANS]X1 = A[Intercept]+[SQFT]B2X2+[INV]B3X3+[AMT]B4X4+[SALESIZE]B5X5+[COMP]B6X6

Intercept value: Constant. Indicates value of sales when values of B2 to B6 =0.

Slope Values: B2 to B6 🡪 Change in Annual net sales due to a unit change in that specific independent variable

Summary output – Before checking for multicollinearity

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SUMMARY OUTPUT |  |  | | | |  | | | | | | |  | |  | | |  | | |
|  |  | | |  | | | |  | | |  | | |  | | | | | |  | | |  | | |  | |
| *Regression Statistics* |  | |  | | | |  | | |  | | | | | | |  | |  | | |  | | |
| Multiple R | 0.996584 | | |  | | | |  | | |  | | |  | | | | | |  | | |  | | |  | |
| R Square | 0.993179 | | |  | | | |  | | |  | | |  | | | | | |  | | |  | | |  | |
| Adjusted R Square | 0.991556 | | |  | | | |  | | |  | | |  | | | | | |  | | |  | | |  | |
| Standard Error | 17.64924 | | |  | | | |  | | |  | | |  | | | | | |  | | |  | | |  | |
| Observations | 27 | | |  | | | |  | | |  | | |  | | | | | |  | | |  | | |  | |
|  |  | | |  | | | |  | | |  | | |  | | | | | |  | | |  | | |  | |
| ANOVA |  | | |  | | | |  | | |  | | |  | | | | | |  | | |  | | |  | |
|  | *df* | | | *SS* | | | | *MS* | | | *F* | | | *Significance F* | | | | | |  | | |  | | |  | |
| Regression | 5 | | | 952538.9 | | | | 190507.8 | | | 611.5904 | | | 5.4E-22 | | | | | |  | | |  | | |  | |
| Residual | 21 | | | 6541.41 | | | | 311.4957 | | |  | | |  | | | | | |  | | |  | | |  | |
| Total | 26 | | | 959080.4 | | | |  | | |  | | |  | | | | | |  | | |  | | |  | |
|  |  | | |  | | | |  | | |  | | |  | | | | | |  | | |  | | |  | |
|  | *Coefficients* | | | | *Standard Error* | | | | *t Stat* | | | *P-value* | | | | *Lower 95%* | | | | *Upper 95%* | | | | *Lower 95.0%* | | | *Upper 95.0%* | |
| Intercept | -18.8594 | | | | 30.15023 | | | | -0.62551 | | | 0.538372 | | | | -81.5602 | | | | 43.84142 | | | | -81.5602 | | | 43.84142 | |
| Number sq. ft./1000 | 16.20157 | | | | 3.544437 | | | | 4.570986 | | | 0.000166 | | | | 8.830513 | | | | 23.57263 | | | | 8.830513 | | | 23.57263 | |
| Inventory/$1000 | 0.174635 | | | | 0.057606 | | | | 3.031541 | | | 0.006347 | | | | 0.054837 | | | | 0.294434 | | | | 0.054837 | | | 0.294434 | |
| amount spent on advertising/$1000 | 11.52627 | | | | 2.532103 | | | | 4.552053 | | | 0.000174 | | | | 6.260472 | | | | 16.79207 | | | | 6.260472 | | | 16.79207 | |
| Size of sales district/1000 families | 13.58031 | | | | 1.770457 | | | | 7.670514 | | | 1.61E-07 | | | | 9.898447 | | | | 17.26218 | | | | 9.898447 | | | 17.26218 | |
| Number of competing stores in district | -5.31097 | | | | 1.705427 | | | | -3.11416 | | | 0.005249 | | | | -8.8576 | | | | -1.76434 | | | | -8.8576 | | | -1.76434 | |

*Font made smaller to fit to page*

Interpretation

The equation

[ANS]X1 = A[-18.86]+[SQFT]16.2016X2+[INV]0.17464X3+[AMT]11.5263X4+[SALESIZE]13.5803X5-[COMP]5.31097X6

|  |  |  |  |
| --- | --- | --- | --- |
| Independent variables | Description | Values | Inference |
| [SQFT]B2X2 | X2 = number sq. ft./1000 | 16.2016 | For every square foot increase, annual net sales will increase by $16202 |
| [INV]B3X3 | X3 = inventory/$1000 | 0.17464 | For every one unit increase in stock/inventory, the annual net sales will increase by $1.75 |
| [AMT]B4X4 | X4 = amount spent on advertising/$1000 | 11.5263 | For every one $ increase in amount spent on advertising, the annual net sales will increase by $11,526 |
| [SALESIZE]B5X5 | X5 = size of sales district/1000 families | 13.5803 | For an additional family size of 1, the annual net sales will increase by $13580 |
| [COMP]B6X6 | X6 = number of competing stores in district | -5.31097 | For every 1 unit/1 store added to to competition market, the annual net sales will go down by $5311 |

*The R squared is high at 99&, hence, 99% of change in dependent variable is being explained by independent variables*

Summary output – After checking for multicollinearity

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SUMMARY OUTPUT | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| *Regression Statistics* | |  |  |  |  |  |  |  |
| Multiple R | 0.995085 |  |  |  |  |  |  |  |
| R Square | 0.990195 |  |  |  |  |  |  |  |
| Adjusted R Square | 0.988412 |  |  |  |  |  |  |  |
| Standard Error | 20.67512 |  |  |  |  |  |  |  |
| Observations | 27 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| ANOVA |  |  |  |  |  |  |  |  |
|  | *df* | *SS* | *MS* | *F* | *Significance F* |  |  |  |
| Regression | 4 | 949676.2 | 237419.1 | 555.4175 | 9.58E-22 |  |  |  |
| Residual | 22 | 9404.131 | 427.4605 |  |  |  |  |  |
| Total | 26 | 959080.4 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | *Coefficients* | *Standard Error* | *t Stat* | *P-value* | *Lower 95%* | *Upper 95%* | *Lower 95.0%* | *Upper 95.0%* |
| Intercept | -39.46 | 34.41056 | -1.14674 | 0.263808 | -110.823 | 31.90311 | -110.823 | 31.90311 |
| number sq. ft./1000 | 20.44389 | 3.814801 | 5.359096 | 2.22E-05 | 12.53247 | 28.3553 | 12.53247 | 28.3553 |
| amount spent on advertising/$1000 | 16.96614 | 2.092788 | 8.106959 | 4.73E-08 | 12.62597 | 21.30632 | 12.62597 | 21.30632 |
| size of sales district/1000 families | 15.67296 | 1.909856 | 8.20636 | 3.86E-08 | 11.71216 | 19.63376 | 11.71216 | 19.63376 |
| number of competing stores in district | -4.0433 | 1.936828 | -2.08759 | 0.048629 | -8.06004 | -0.02656 | -8.06004 | -0.02656 |

*Font made smaller to fit to page*

Interpretation

The equation

[ANS]X1 = A[-39.46]+[SQFT]20.444X2+[AMT]16.966X4+[SALESIZE]15.673X5-[COMP]4.043X6

|  |  |  |  |
| --- | --- | --- | --- |
| Independent variables | Description | Values | Inference |
| [SQFT]B2X2 | X2 = number sq. ft./1000 | 20.444 | For every square foot increase, annual net sales will increase by $20,444 |
| [AMT]B4X4 | X4 = amount spent on advertising/$1000 | 16.966 | For every one $ increase in amount spent on advertising, the annual net sales will increase by $16,966 |
| [SALESIZE]B5X5 | X5 = size of sales district/1000 families | 15.673 | For an additional family size of 1, the annual net sales will increase by $15673 |
| [COMP]B6X6 | X6 = number of competing stores in district | -4.043 | For every 1 unit/1 store added to to competition market, the annual net sales will go down by $4043 |

*The R squared is high at 99&, hence, 99% of change in dependent variable is being explained by independent variables*

Checking variance with Variation Inflation Factor test (Check for multicollinearity)

**Purpose:**

Multicollinearity between independent variables could be problematic, like if two or more variables are highly correlated it shows some interdependency between the variables. Hence it is better to check if the independent variables are completely independent of each other. This will help in removing any variables that have high variance due to multicollinearity

The VIF is a test for multicollinearity. The norm is that if variance of the independent variables are greater than 10, then should be removed from the model. That is multicollinearity for that independent variable is high

Reference on this cutoff of 10 is taken from this source: [*https://en.wikipedia.org/wiki/Variance\_inflation\_factor*](https://en.wikipedia.org/wiki/Variance_inflation_factor)

R codes

Checking for correlation in R:

Codes used:

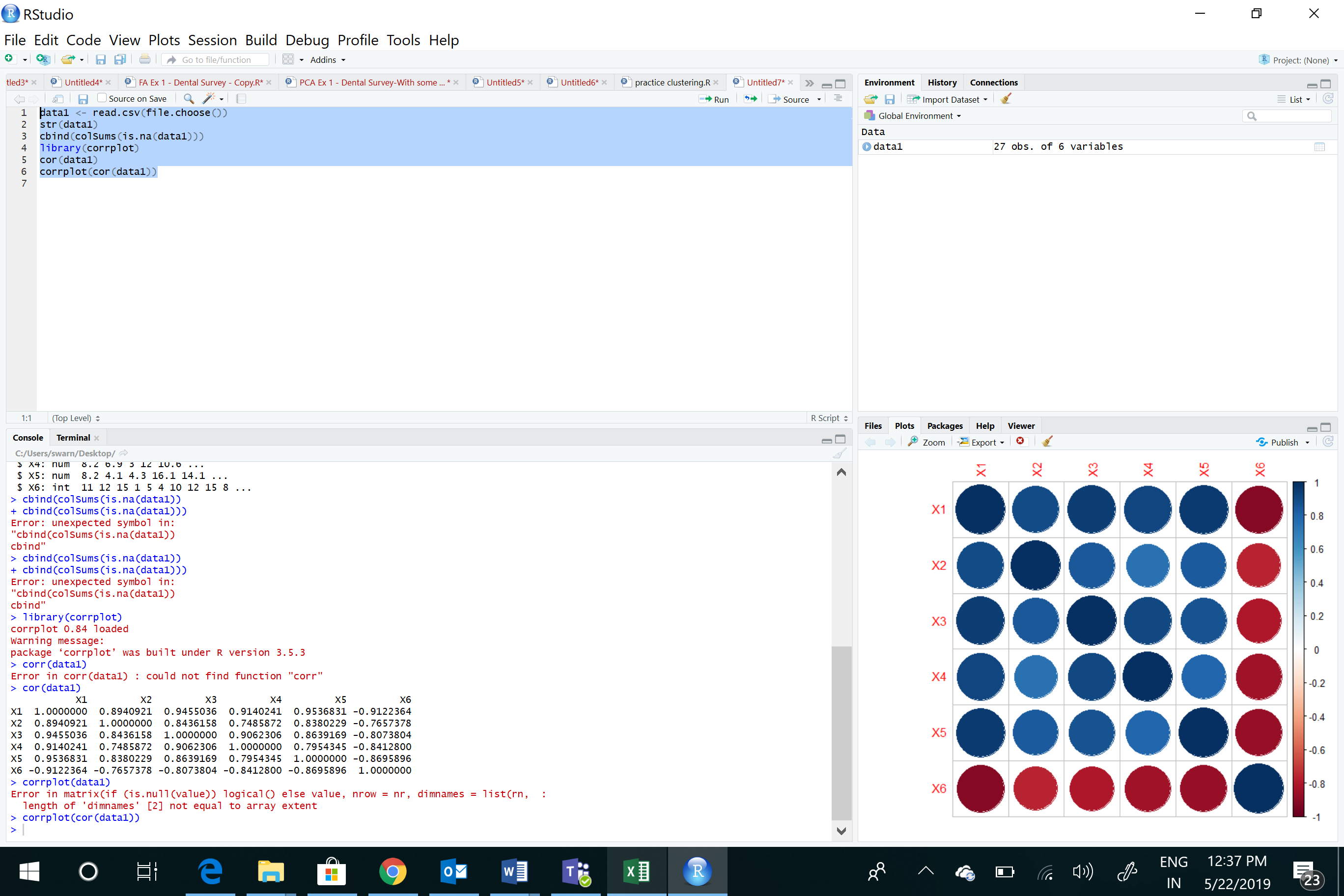
data1 <- read.csv(file.choose())

str(data1)

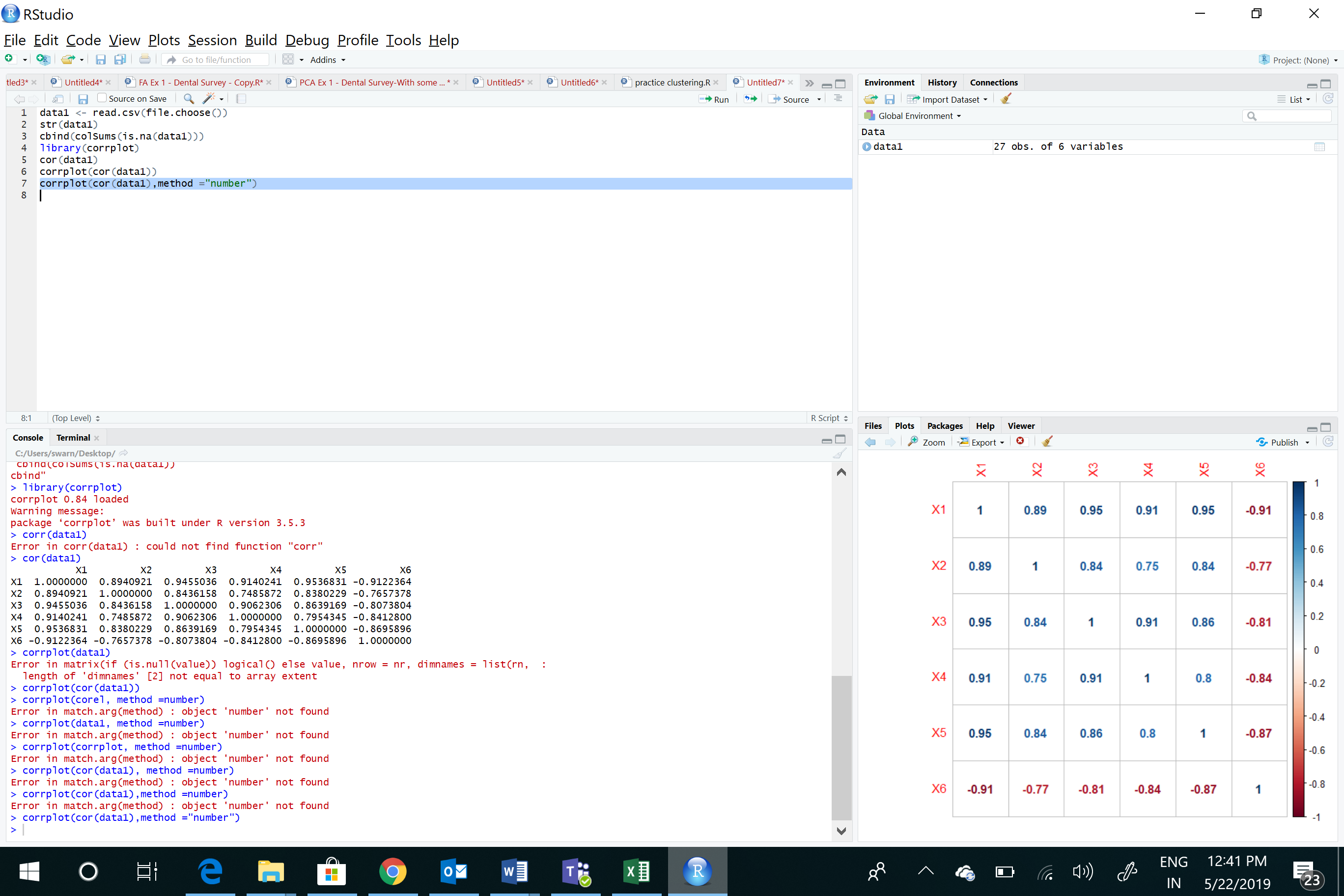
library(corrplot)

cor(data1)

corrplot(cor(data1))



corrplot(cor(data1),method ="number")



From this it can be said that there can be a problem of multicollinearity. Since the correlation values are very high

While regressing the data set in R, below are the codes used

reg1 <- lm(X1 ~ ., data= data1)

summary(reg1)

Call:

lm(formula = X1 ~ ., data = data1)

Residuals:

Min 1Q Median 3Q Max

-26.338 -9.699 -4.496 4.040 41.139

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -18.85941 30.15023 -0.626 0.538372

X2 16.20157 3.54444 4.571 0.000166 \*\*\*

X3 0.17464 0.05761 3.032 0.006347 \*\*

X4 11.52627 2.53210 4.552 0.000174 \*\*\*

X5 13.58031 1.77046 7.671 1.61e-07 \*\*\*

X6 -5.31097 1.70543 -3.114 0.005249 \*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 17.65 on 21 degrees of freedom

Multiple R-squared: 0.9932, Adjusted R-squared: 0.9916

F-statistic: 611.6 on 5 and 21 DF, p-value: < 2.2e-16

As mentioned earlier all parameters are significant with a high R-squared value. However the VIF test will help remove multicollinearity

Running the VIF test in R, the below values are obtained:

library(car)

vif(reg1)

X2 X3 X4 X5 X6

4.240914 10.122480 7.624391 6.912318 5.818768

As explained, VIF>10 will be removed indicating high multicollinearity. In this case it is X3 or inventory /$1000 that will be removed

Re-running regression without inventory, using the below R codes:

reg2 <- lm(X1 ~ .-X3, data= data1)

summary(reg2)

Call:

lm(formula = X1 ~ . - X3, data = data1)

Residuals:

Min 1Q Median 3Q Max

-30.422 -12.858 -6.477 16.160 45.255

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -39.460 34.411 -1.147 0.2638

X2 20.444 3.815 5.359 2.22e-05 \*\*\*

X4 16.966 2.093 8.107 4.73e-08 \*\*\*

X5 15.673 1.910 8.206 3.86e-08 \*\*\*

X6 -4.043 1.937 -2.088 0.0486 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.68 on 22 degrees of freedom

Multiple R-squared: 0.9902, Adjusted R-squared: 0.9884

F-statistic: 555.4 on 4 and 22 DF, p-value: < 2.2e-16

From this we can see all variables have statistical significance. Checking vif for this model:

vif(reg2)

X2 X4 X5 X6

3.579850 3.795323 5.861520 5.468943

All values are below 10, so the problem of multicollinearity is resolved

Graphical representation – R codes

install.packages("GGally")

library(GGally)

ggpairs(data1)

Below chart gives correlation values. From this we can see X3 and X5 have maximum correlation with dependent variable X1, after the VIF test X3 was removed and reg2 was considered a better model than reg1

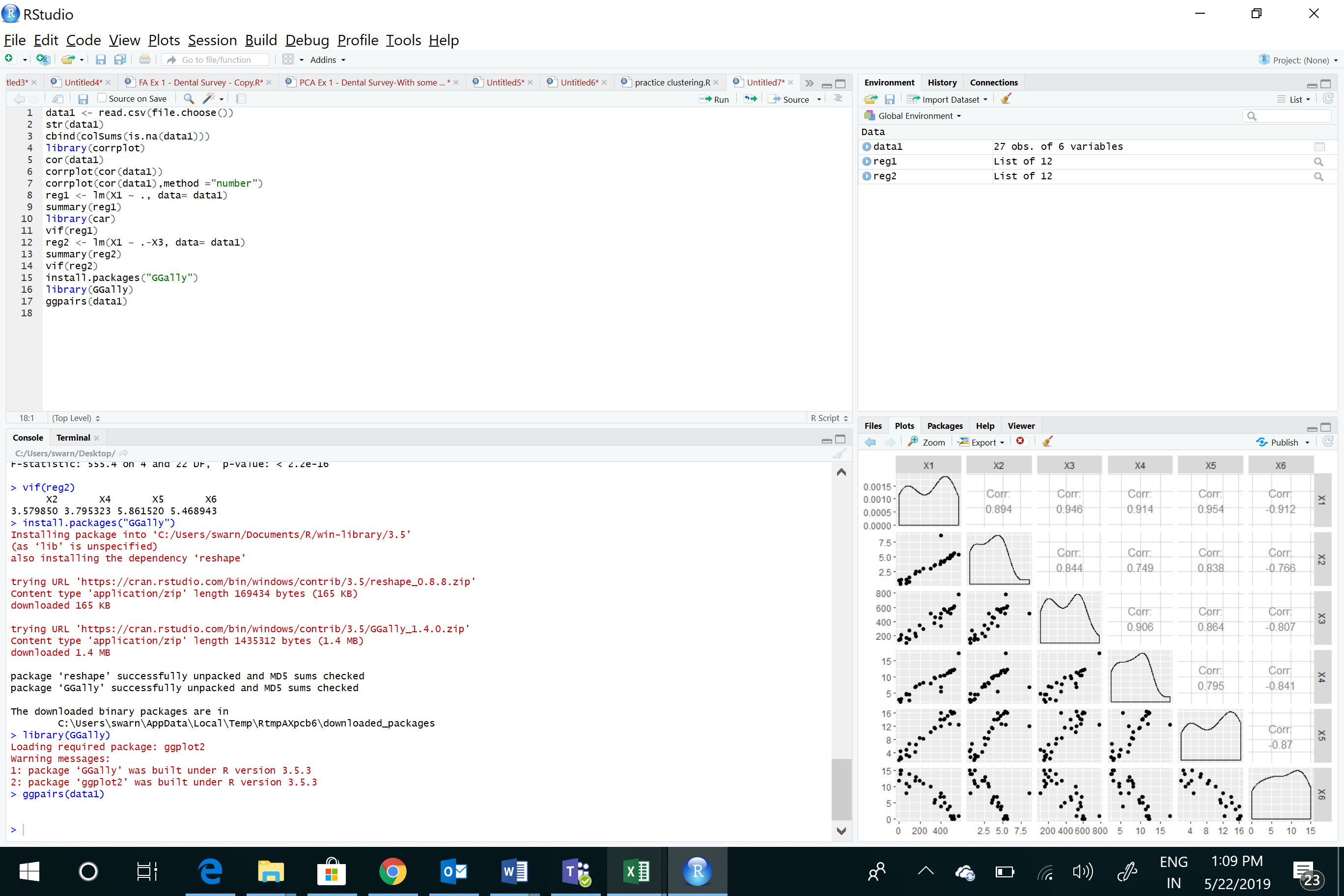
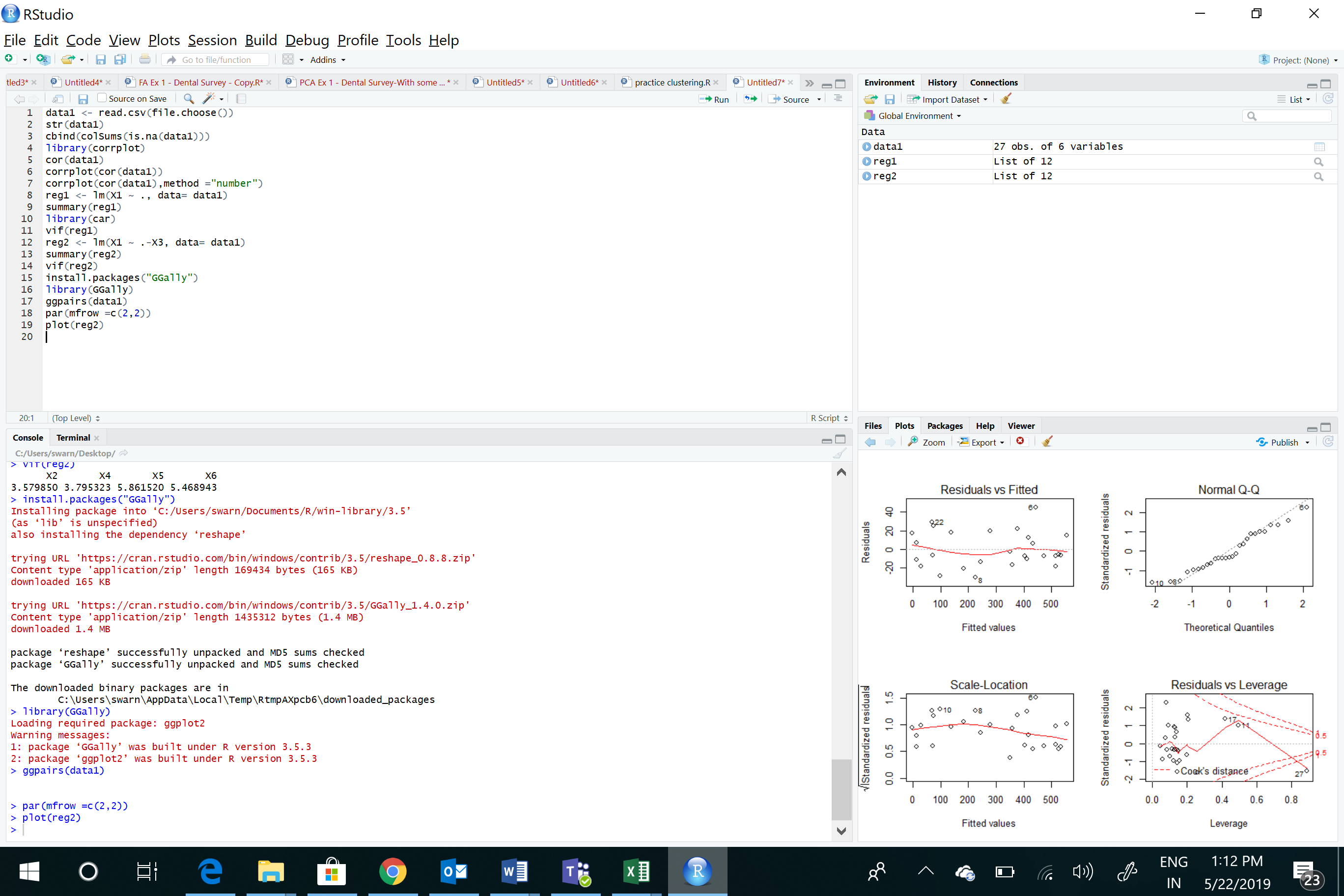


Chart depicting model fit:

par(mfrow =c(2,2))

plot(reg2)

The second model is used. Checked and resolved the issue of multicollinearity



**THANK** **YOU**