Starbucks Regression Analysis
Phase II

Loading libraries & checking for missing values or null values:

```r
library(readr)
library(tidyr)
library(stringr)
library(plyr)
library(dplyr)
library(RVAideMemoire)
library(DescTools)
library(corrplot)
library(glmnet)
library(ggfortify)

#Setup Working Directory
setwd("/Users/manpreetkaurgurtatta/Library/CloudStorage/Box-Box/Sem 1/Data Stats/Assignments/Assignment 4- Regressions")

#Loading and Reading the Dataset
Starbucks <- read.csv("Starbucks_data-3.csv")
View(Starbucks)
head(Starbucks)

##Checking Dimensions of Dataset
dim(Starbucks)
#122 rows  23 columns

#Checking Variable Names
names(Starbucks)

#Checking Data structure
str(Starbucks)

#Renaming files to simplify data
names(Starbucks) <- c('Timestamp', 'Gender', 'Age', 'Status', 'Income', 'Visits', 'ServiceMode', 'Timespend',
                      'Nearestlocation', 'Membershipcard', 'Frequentpurchase', 'SpendPurchase', 'QualityRate',
                      'PriceRate', 'PromoRate', 'Ambiance', 'WiFiRate', 'ServiceRate', 'chooseRate',
                       'PromoMethod', 'loyalty', 'BeforePromoSatRate', 'AfterPromoSatRate')
```

```r
Starbucks_Updated <- Starbucks
str(Starbucks_Updated)
View(Starbucks_Updated)

#Checking for Missing Data
colnames(Starbucks)
sum(is.na(Starbucks))

#For Missing data, using listwise deletion.
Starbucks_Updated <- na.omit(Starbucks)
sum(is.na(Starbucks_Updated))
#No missing data
View(Starbucks_Updated)

#Converting into numeric values

#Gender
table(Starbucks_Updated$Gender)
Starbucks_Updated$Gender_num <- revalue(Starbucks_Updated$Gender, c("Male"="1", "Female"="0"))
Starbucks_Updated$Gender_num <- as.numeric(Starbucks_Updated$Gender_num)

#Membershipcard
Starbucks_Updated$Membershipcard_num <- revalue(Starbucks_Updated$Membershipcard, c("Yes"="1", "No"="0"))
Starbucks_Updated$Membershipcard_num <- as.numeric(Starbucks_Updated$Membershipcard_num)

#loyalty
Starbucks_Updated$loyalty_num <- revalue(Starbucks_Updated$loyalty, c("Yes"="1", "No"="0"))
Starbucks_Updated$loyalty_num <- as.numeric(Starbucks_Updated$loyalty_num)

#Nearestlocation
Starbucks_Updated$Nearestlocation_num <- revalue(Starbucks_Updated$Nearestlocation, c("within 1km"="0", "1km - 3km"="1",
                                                                                       "more than 3km"="2"))
Starbucks_Updated$Nearestlocation_num <- as.numeric(Starbucks_Updated$Nearestlocation_num)

#..........................................................................................................
```

1. Create a **multiple linear regression**. How would you rate the quality of Starbucks compared to other brands (Coffee Bean, Old Town White Coffee..) to be: (use as a numerical variable).

a. Are we able to use all independent variables? If not, why?

Independent variables (IVs) are the ones that you include in the model to explain or predict changes in the dependent variable which indicates that they stand alone and other variables in the model do not influence them. In the given dataset, we are not able to use all the independent variables since a lot of the variables are categorical which only increases the complexity of the regression model as it requires the data to be numeric.

b. What assumptions do we need to check for? Are there any violations of these assumptions with this equation? If so, how would we correct them?

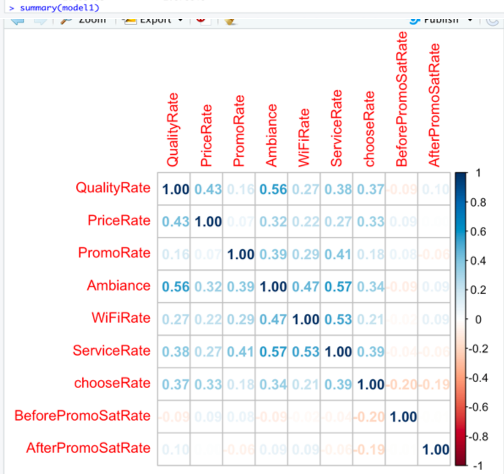
Following assumptions need to be checked for:

- There shouldn't be any Multicollinearity in the data. By identifying the variables causing multicollinearity issues, for example - correlations or VIF values and removing those variables from the regression, we can center the data.
- There shouldn't be any autocorrelation in the data. This can be corrected by removing the variables having strong correlations.
- The relationship between dependent and independent should be linear. The violation to this assumption can be corrected by simply applying a nonlinear transformation to the independent variable such as taking the log or the square root.
- The residuals of the multiple linear regression model should be normally distributed.

c. Run a **manual regression**, and check for **VIF**, what do we learn about VIF? Are there any additional variables we need to remove from the model?

```r
#.........................................................................................................................
#Q1(c):
#First we will create a new subset containing numeric variables only
names(Starbucks_Updated)
DataViz <- Starbucks_Updated[, c(13,14,15,16,17,18,19,22,23)]
str(DataViz)

#Checking Spearman Correlations
library(corrplot)
corrplot(cor(DataViz, method = "spearman"))
corrplot(cor(DataViz, method = "spearman"), method="number")

#Creating Linear Regression Model
model1 <- lm(QualityRate ~ ., data=DataViz)
model1

#Checking for VIF
library(DescTools)
VIF(model1)
summary(model1)

#Creating new subset and model since there are variables causing multicollinearity
DataViz_new <- Starbucks_Updated[, c(13,14,15,16,17,18,19,22,23,24,25,26,27)]
View(DataViz_new)

#Creating a manual regression model
model2 <- lm(QualityRate~ ., data=DataViz_new)
model2

#Checking  VIF
library(DescTools)
VIF(model2)
summary(model2)


#.........................................................................................................................
```

```
> #Q1(c):
> #First we will create a new subset containing numeric variables only
> names(Starbucks_Updated)
 [1] "Timestamp"          "Gender"             "Age"                "Status"             "Income"
 [6] "Visits"             "ServiceMode"        "Timespend"          "Nearestlocation"    "Membershipcard"
[11] "Frequentpurchase"   "SpendPurchase"      "QualityRate"        "PriceRate"          "PromoRate"
[16] "Ambiance"           "WiFiRate"           "ServiceRate"        "chooseRate"         "PromoMethod"
[21] "loyalty"            "BeforePromoSatRate" "AfterPromoSatRate"  "Gender_num"         "Membershipcard_num"
[26] "loyalty_num"        "Nearestlocation_num"
> DataViz <- Starbucks_Updated[, c(13,14,15,16,17,18,19,22,23)]
> str(DataViz)
'data.frame':	112 obs. of  9 variables:
 $ QualityRate       : int  4 4 4 2 3 4 5 4 5 4 ...
 $ PriceRate         : int  3 3 3 1 3 3 5 2 4 3 ...
 $ PromoRate         : int  5 4 4 4 4 5 5 3 4 3 ...
 $ Ambiance          : int  5 4 4 3 2 5 5 3 4 4 ...
 $ WiFiRate          : int  4 4 4 3 2 4 3 3 4 3 ...
 $ ServiceRate       : int  4 5 4 3 3 5 5 3 4 3 ...
 $ chooseRate        : int  3 2 3 3 3 4 5 3 4 4 ...
 $ BeforePromoSatRate: int  5 4 4 1 4 2 4 4 1 1 ...
 $ AfterPromoSatRate : int  5 3 5 3 2 1 5 4 4 5 ...
> #Checking Spearman Correlations
> library(corrplot)
> corrplot(cor(DataViz, method = "spearman"))
> corrplot(cor(DataViz, method = "spearman"), method="number")
> #Creating Linear Regression Model
> model1 <- lm(QualityRate ~ ., data=DataViz)
> model1

Call:
lm(formula = QualityRate ~ ., data = DataViz)

Coefficients:
       (Intercept)           PriceRate           PromoRate            Ambiance            WiFiRate         ServiceRate
           1.09067             0.22944            -0.02698             0.41364            -0.06198             0.07392
        chooseRate  BeforePromoSatRate   AfterPromoSatRate
           0.11401            -0.03537             0.05440

> #Checking for VIF
> library(DescTools)
> VIF(model1)
         PriceRate          PromoRate           Ambiance           WiFiRate        ServiceRate         chooseRate
          1.320663           1.253813           1.778244           1.564381           2.105013           1.391633
BeforePromoSatRate  AfterPromoSatRate
          1.099468           1.076646
> summary(model1)
```



```
Call:
lm(formula = QualityRate ~ ., data = DataViz)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.59388 -0.33101 -0.04901  0.33616  2.00394 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)        1.09067    0.41944   2.600  0.01068 *  
PriceRate          0.22944    0.06994   3.281  0.00141 ** 
PromoRate         -0.02698    0.06838  -0.394  0.69403    
Ambiance           0.41364    0.09131   4.530 1.59e-05 ***
WiFiRate          -0.06198    0.08498  -0.729  0.46739    
ServiceRate        0.07392    0.11035   0.670  0.50440    
chooseRate         0.11401    0.07248   1.573  0.11876    
BeforePromoSatRate -0.03537    0.05151  -0.687  0.49383    
AfterPromoSatRate  0.05440    0.04724   1.151  0.25222    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6674 on 103 degrees of freedom
Multiple R-squared:  0.4594,	Adjusted R-squared:  0.4174 
F-statistic: 10.94 on 8 and 103 DF,  p-value: 4.643e-11

> #Creating new subset and model since there are variables causing multicollinearity
> DataViz_new <- Starbucks_Updated[, c(13,14,15,16,17,18,19,22,23,24,25,26,27)]
> View(DataViz_new)
> #Creating a manual regression model
> model2 <- lm(QualityRate~ ., data=DataViz_new)
> model2

Call:
lm(formula = QualityRate ~ ., data = DataViz_new)

Coefficients:
       (Intercept)           PriceRate           PromoRate            Ambiance            WiFiRate  
           1.16958             0.21470            -0.01756             0.39809            -0.05499  
       ServiceRate          chooseRate  BeforePromoSatRate   AfterPromoSatRate          Gender_num  
           0.06627             0.08561            -0.05100             0.05249             0.14463  
 Membershipcard_num         loyalty_num  Nearestlocation_num  
           0.13789             0.08935            -0.04742  

> #Checking  VIF
> library(DescTools)
> VIF(model2)
```


```
> #Checking  VIF
> library(DescTools)
> VIF(model2)
```


```
          PriceRate           PromoRate            Ambiance            WiFiRate         ServiceRate          chooseRate 
           1.456313            1.272986            1.905759            1.601744            2.250098            1.476012 
 BeforePromoSatRate   AfterPromoSatRate          Gender_num  Membershipcard_num         loyalty_num Nearestlocation_num 
           1.150799            1.101165            1.086031            1.339601            1.398289            1.206390 
> summary(model2)

Call:
lm(formula = QualityRate ~ ., data = DataViz_new)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.41169 -0.38543 -0.07132  0.29971  1.75068 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)          1.16958    0.48219   2.426  0.01710 *  
PriceRate            0.21470    0.07357   2.918  0.00436 ** 
PromoRate           -0.01756    0.06902  -0.254  0.79967    
Ambiance             0.39809    0.09469   4.204 5.75e-05 ***
WiFiRate            -0.05499    0.08613  -0.638  0.52465    
ServiceRate          0.06627    0.11428   0.580  0.56331    
chooseRate           0.08561    0.07477   1.145  0.25497    
BeforePromoSatRate  -0.05100    0.05279  -0.966  0.33630    
AfterPromoSatRate    0.05249    0.04786   1.097  0.27540    
Gender_num           0.14463    0.13270   1.090  0.27842    
Membershipcard_num   0.13789    0.14632   0.942  0.34828    
loyalty_num          0.08935    0.18204   0.491  0.62463    
Nearestlocation_num -0.04742    0.08538  -0.555  0.57984    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6685 on 99 degrees of freedom
Multiple R-squared:  0.4786,	Adjusted R-squared:  0.4154 
F-statistic: 7.573 on 12 and 99 DF,  p-value: 9.314e-10

> #..........................................................................................................
```

In model1, we found some correlations between variables causing multicollinearity. Logically, VIF measures the extent of correlation between two independent variables. The higher the VIF the difficult it is assessing the contribution of independent variable to a model. Hence, we removed the variables having VIF more than 5 and created a new manual regression model, model 2 that removed the multicollinearity. This way we got the value of VIF for all variables well within its range. (between 1 and 2).

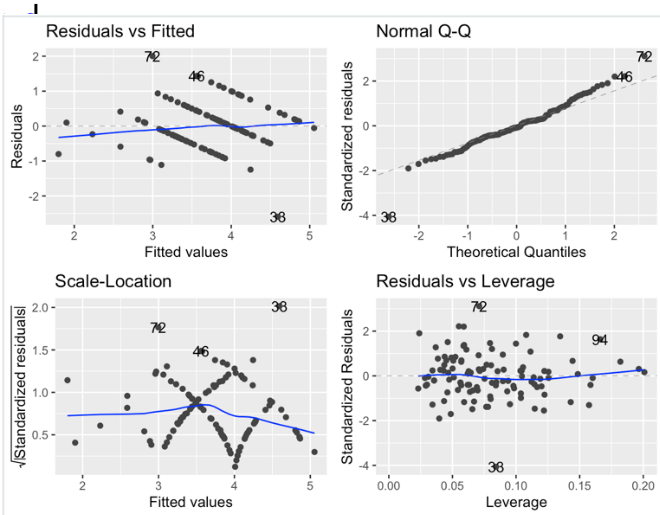
d. Create a plot of the **model diagnostic plots**. What do we learn from these plots? Are there any issues with this model? Do you trust the reliability of these model?

Input-

```
#...........................................................................................................

#Q1(d):
#To create Diagnostic Plots for Model Fit
library(ggfortify)
par(mfrow = c(2, 2))
plot(model1)

autoplot(model1)
par(mfrow = c(1, 1))
#...........................................................................................................
```

Output-

```
> #.........................................................................................................
>
> #Q1(d):
> #To create Diagnostic Plots for Model Fit
> library(ggfortify)
> par(mfrow = c(2, 2))
> plot(model1)
>
> autoplot(model1)
> par(mfrow = c(1, 1))
> #.........................................................................................................
```



There are 4 model diagnostic plots:

1. The **Residuals Vs. Fitted plot** is used to detect **non-linearity, unequal error variances, and outliers**. In the given scenario, the blue line in the plot being horizontal portrays that the given model does not have any linearity problems.

2. The **Normal Q-Q plot** shows the distribution of the data against the expected normal distribution. In our data, since the points are diagonally aligned, we can say that the model is normally distributed.
3. The **Scale-Location plot** is used to check the assumption of equal variance (homoscedasticity). In our data, the blue line is inclined downwards towards the end, which is indicative of the model having homoscedasticity problems.
4. The **Residuals Vs. Leverage plot** is used to detect **outliers** in a linear regression model. The plot clearly indicates the presence of outliers suggesting a significant impact on the model.

e. Using a **stepwise equation**, explain the output of the regressions. Comment on the overall significance of the regression fit. Which predictors have coefficients that are significantly different from zero at the .05 level?

```
#..............................................................................................................

#Q1(e):
#Using Stepwise Multiple Linear Regression
null = lm(QualityRate ~ 1-QualityRate, data=DataViz_new)
null

full = lm(QualityRate ~ .-QualityRate, data=DataViz_new)
full

#Stepwise Regression
train_Stepwise = step(null, scope = list(upper=full), direction="both")
summary(train_Stepwise)

#..............................................................................................................
```

```
> #.........................................................................................................
>
> #Q1(e):
> #Using Stepwise Multiple Linear Regression
> null = lm(QualityRate ~ 1-QualityRate, data=DataViz_new)
> null

Call:
lm(formula = QualityRate ~ 1 - QualityRate, data = DataViz_new)

Coefficients:
(Intercept)
      3.714

>
> full = lm(QualityRate ~ .-QualityRate, data=DataViz_new)
> full

Call:
lm(formula = QualityRate ~ . - QualityRate, data = DataViz_new)

Coefficients:
        (Intercept)            PriceRate            PromoRate             Ambiance
            1.16958              0.21470             -0.01756              0.39809
           WiFiRate          ServiceRate           chooseRate    BeforePromoSatRate
           -0.05499              0.06627              0.08561              -0.05100
  AfterPromoSatRate           Gender_num    Membershipcard_num          loyalty_num
            0.05249              0.14463              0.13789              0.08935
Nearestlocation_num
           -0.04742

>
> #Stepwise Regression
> train_Stepwise = step(null, scope = list(upper=full), direction="both")
Start:  AIC=-29.08
QualityRate ~ 1 - QualityRate

                      Df Sum of Sq    RSS     AIC
+ Ambiance             1   29.5684 55.289 -75.064
+ PriceRate            1   19.1601 65.697 -55.746
+ ServiceRate          1   15.0130 69.844 -48.890
+ chooseRate           1   12.2373 72.620 -44.525
+ loyalty_num          1    9.1602 75.697 -39.877
```

```
+ loyalty_num          1    2.3976 52.891 -78.030
+ Membershipcard_num   1    1.6316 53.657 -76.419
+ Gender_num           1    0.9966 54.292 -75.101
<none>                             55.289 -75.064
+ Nearestlocation_num  1    0.8280 54.461 -74.754
+ ServiceRate          1    0.6202 54.669 -74.328
+ AfterPromoSatRate    1    0.1780 55.111 -73.425
+ PromoRate            1    0.1692 55.120 -73.407
+ BeforePromoSatRate   1    0.1162 55.173 -73.300
+ WiFiRate             1    0.0002 55.288 -73.065
- Ambiance             1   29.5684 84.857 -29.083

Step:  AIC=-88.12
QualityRate ~ Ambiance + PriceRate

                      Df Sum of Sq    RSS     AIC
+ chooseRate           1    1.2888 47.046 -89.146
+ Membershipcard_num   1    1.2131 47.122 -88.966
<none>                             48.335 -88.119
+ Gender_num           1    0.7767 47.558 -87.933
+ BeforePromoSatRate   1    0.5634 47.771 -87.432
+ loyalty_num          1    0.5034 47.831 -87.291
+ AfterPromoSatRate    1    0.2379 48.097 -86.671
+ Nearestlocation_num  1    0.2274 48.107 -86.647
+ ServiceRate          1    0.1712 48.164 -86.516
+ PromoRate            1    0.0963 48.238 -86.342
+ WiFiRate             1    0.0702 48.265 -86.282
- PriceRate            1    6.9539 55.289 -75.064
- Ambiance             1   17.3622 65.697 -55.746

Step:  AIC=-89.15
QualityRate ~ Ambiance + PriceRate + chooseRate

                      Df Sum of Sq    RSS     AIC
+ Membershipcard_num   1    0.9135 46.133 -89.342
<none>                             47.046 -89.146
+ Gender_num           1    0.6041 46.442 -88.593
+ AfterPromoSatRate    1    0.5250 46.521 -88.403
- chooseRate           1    1.2888 48.335 -88.119
+ BeforePromoSatRate   1    0.2892 46.757 -87.836
+ loyalty_num          1    0.2468 46.799 -87.735
+ Nearestlocation_num  1    0.2220 46.824 -87.675
+ PromoRate            1    0.1362 46.910 -87.471


+ WiFiRate             1    0.1165 46.930 -87.423
+ ServiceRate          1    0.0199 47.026 -87.193
- PriceRate            1    4.8203 51.866 -80.221
- Ambiance             1   14.5524 61.598 -60.961

Step:  AIC=-89.34
QualityRate ~ Ambiance + PriceRate + chooseRate + Membershipcard_num

                      Df Sum of Sq    RSS     AIC
<none>                             46.133 -89.342
- Membershipcard_num   1    0.9135 47.046 -89.146
- chooseRate           1    0.9891 47.122 -88.966
+ AfterPromoSatRate    1    0.5281 45.604 -88.631
+ Gender_num           1    0.4472 45.685 -88.433
+ BeforePromoSatRate   1    0.4450 45.688 -88.427
+ PromoRate            1    0.1357 45.997 -87.672
+ loyalty_num          1    0.0771 46.056 -87.529
+ WiFiRate             1    0.0742 46.058 -87.522
+ Nearestlocation_num  1    0.0532 46.079 -87.471
+ ServiceRate          1    0.0026 46.130 -87.348
- PriceRate            1    4.7097 50.842 -80.454
- Ambiance             1   12.4249 58.557 -64.631
> summary(train_Stepwise)

Call:
lm(formula = QualityRate ~ Ambiance + PriceRate + chooseRate +
    Membershipcard_num, data = DataViz_new)

Residuals:
     Min       1Q   Median       3Q      Max
-2.30257 -0.28839 -0.03345  0.31405  1.96279

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         1.10452    0.30026   3.679 0.000369 ***
Ambiance            0.40514    0.07547   5.368 4.65e-07 ***
PriceRate           0.22086    0.06683   3.305 0.001292 **
chooseRate          0.10195    0.06731   1.515 0.132806
Membershipcard_num  0.19055    0.13091   1.456 0.148441
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6566 on 107 degrees of freedom
Multiple R-squared:  0.4564,	Adjusted R-squared:  0.436
F-statistic: 22.45 on 4 and 107 DF,  p-value: 1.757e-13
```

The stepwise equation regression is a method in which the choice of predictive variables is carried out by an automatic procedure. It is evident from the results that the difference between Multiple R-squared error and adjusted R-Squared error is almost negligible which means there’s no overfitting. Further, the Global p-test value is less than 0.05 showing the result is significant.

2. Create a linear regression. How would you rate the quality of Starbucks compared to other brands (Coffee Bean, Old Town White Coffee..) to be: (use as a numerical variable) using **Lasso Regression.**

a) How is Lasso regression different than manual/automatic regression?

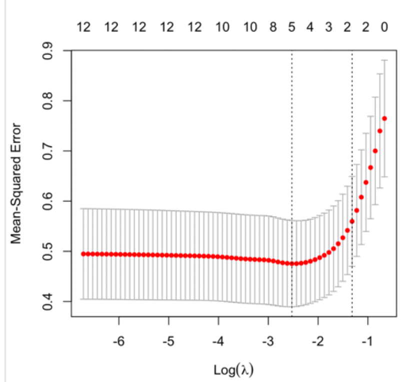
The main advantage of Lasso Regression over other manual/ automatic regression methods is that it has the ability to set the coefficients for features it does not consider interesting to zero. It produces simpler and more interpretable models that incorporate only a reduced set of the predictors , has reduced overfitting and does some automatic feature selection to decide which features should and should not be included on its own unlike other regression methods.

b) Explain the output of the Lasso regression.

```
#..........................................................................................................

#Q2(b):
library(glmnet)
x=model.matrix(QualityRate ~ ., data=DataViz_new)
y=DataViz_new$QualityRate
cv.lasso <- cv.glmnet(x,y, typemeasure="mse", alpha=1)
cv.lasso

ls(cv.lasso)
plot(cv.lasso)
Lambda.best <- cv.lasso$lambda.min

predict(cv.lasso, s = Lambda.best, type = "coefficients")

fit<-cv.lasso$glmnet.fit
fit

#Ordinary Least Squares Regression
model3 <- lm(QualityRate ~ PriceRate+PromoRate+Ambiance+WiFiRate+ServiceRate+chooseRate+BeforePromoSatRate+
               AfterPromoSatRate , data=DataViz_new)
summary(model3)

#..........................................................................................................
```

```
> #Q2(b):
> library(glmnet)
> x=model.matrix(QualityRate ~ ., data=DataViz_new)
> y=DataViz_new$QualityRate
> cv.lasso <- cv.glmnet(x,y, typemeasure="mse", alpha=1)
> cv.lasso

Call:  cv.glmnet(x = x, y = y, typemeasure = "mse", alpha = 1)

Measure: Mean-Squared Error

     Lambda Index Measure      SE Nonzero
min 0.07993    21  0.4754 0.08601       5
1se 0.26790     8  0.5596 0.08958       2
>
> ls(cv.lasso)
 [1] "call"       "cvlo"       "cvm"        "cvsd"       "cvup"       "glmnet.fit" "index"      "lambda"     "lambda.1se" "lambda.min"
[11] "name"       "nzero"
> plot(cv.lasso)
> Lambda.best <- cv.lasso$lambda.min
>
> predict(cv.lasso, s = Lambda.best, type = "coefficients")
14 x 1 sparse Matrix of class "dgCMatrix"
                            s1
(Intercept)         1.56681900
(Intercept)         .
PriceRate           0.17811745
PromoRate           .
Ambiance            0.36512392
WiFiRate            .
ServiceRate         .
chooseRate          0.06282393
BeforePromoSatRate  .
AfterPromoSatRate   .
Gender_num          .
Membershipcard_num  0.08244070
loyalty_num         0.00456664
Nearestlocation_num .
>
> fit<-cv.lasso$glmnet.fit
> fit

Call:  glmnet(x = x, y = y, typemeasure = "mse", alpha = 1)

   Df  %Dev  Lambda
1   0  0.00 0.51380
2   1  5.92 0.46820
3   1 10.83 0.42660
4   1 14.91 0.38870
5   2 18.53 0.35420
6   2 22.69 0.32270
7   2 26.15 0.29400
8   2 29.01 0.26790
9   2 31.40 0.24410
10  2 33.37 0.22240
11  2 35.01 0.20270
12  3 36.59 0.18470
13  3 37.94 0.16830
14  3 39.06 0.15330
15  4 40.03 0.13970
16  4 40.98 0.12730
17  4 41.77 0.11600
18  4 42.43 0.10570
19  4 42.97 0.09628
20  4 43.42 0.08773
21  5 43.81 0.07993
22  5 44.14 0.07283
23  5 44.41 0.06636
24  8 44.79 0.06047
25  8 45.23 0.05509
26  8 45.59 0.05020
27  9 45.89 0.04574
28  9 46.16 0.04168
29 10 46.38 0.03797
30 10 46.59 0.03460
31 10 46.76 0.03153
32 10 46.90 0.02873
33 10 47.01 0.02617
34 10 47.11 0.02385
35 10 47.19 0.02173
36 10 47.26 0.01980
37 11 47.35 0.01804
38 11 47.43 0.01644
39 11 47.50 0.01498
40 12 47.56 0.01365
41 12 47.61 0.01243
```

```
42 12 47.65 0.01133
43 12 47.69 0.01032
44 12 47.72 0.00941
45 12 47.74 0.00857
46 12 47.76 0.00781
47 12 47.78 0.00712
48 12 47.79 0.00648
49 12 47.80 0.00591
50 12 47.81 0.00538
51 12 47.82 0.00490
52 12 47.83 0.00447
53 12 47.83 0.00407
54 12 47.84 0.00371
55 12 47.84 0.00338
56 12 47.84 0.00308
57 12 47.85 0.00281
58 12 47.85 0.00256
59 12 47.85 0.00233
60 12 47.85 0.00212
61 12 47.85 0.00193
62 12 47.85 0.00176
63 12 47.86 0.00161
64 12 47.86 0.00146
65 12 47.86 0.00133
66 12 47.86 0.00122
>
> #Ordinary Least Squares Regression
> model3 <- lm(QualityRate ~ PriceRate+PromoRate+Ambiance+WiFiRate+ServiceRate+chooseRate+BeforePromoSatRate+
+                  AfterPromoSatRate , data=DataViz_new)
> summary(model3)

Call:
lm(formula = QualityRate ~ PriceRate + PromoRate + Ambiance +
    WiFiRate + ServiceRate + chooseRate + BeforePromoSatRate +
    AfterPromoSatRate, data = DataViz_new)

Residuals:
     Min       1Q   Median       3Q      Max
-2.59388 -0.33101 -0.04901  0.33616  2.00394

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         1.09067    0.41944   2.600  0.01068 *
PriceRate           0.22944    0.06994   3.281  0.00141 **
PromoRate          -0.02698    0.06838  -0.394  0.69403
Ambiance            0.41364    0.09131   4.530 1.59e-05 ***
WiFiRate           -0.06198    0.08498  -0.729  0.46739
ServiceRate         0.07392    0.11035   0.670  0.50440
chooseRate          0.11401    0.07248   1.573  0.11876
BeforePromoSatRate -0.03537    0.05151  -0.687  0.49383
AfterPromoSatRate   0.05440    0.04724   1.151  0.25222
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6674 on 103 degrees of freedom
Multiple R-squared:  0.4594,	Adjusted R-squared:  0.4174
F-statistic: 10.94 on 8 and 103 DF,  p-value: 4.643e-11

>
> #.....................................................................................................
```



The lasso regression model concludes a minimal correlation between Multiple R-Squared error and the Adjusted R-squared error. We used the following variables for this regression; 'PriceRate', 'PromoRate', 'Ambiance', 'WiFiRate', 'ServiceRate', 'chooseRate', 'PromoMethod', 'BeforePromoSatRate' and 'AfterPromoSatRate'

c) What do you learn about student grades based upon this regression?

In the context of the question and the dataset, 'PriceRate', 'PromoRate', 'Ambiance', 'WiFiRate', 'ServiceRate', 'chooseRate', 'PromoMethod', 'BeforePromoSatRate' 'AfterPromoSatRate', are the variables significant to QualityRate. In case we want to improve the student grades based upon this regression, we’d need to improve these variables as well.