Data 621: Assignment 5

Wine Data

Ritesh Lohiya

July 12, 2018

Overview

In this homework assignment, you will explore, analyze and model a data set containing information on approximately 12,000 commercially available wines. The variables are mostly related to the chemical properties of the wine being sold. The response variable is the number of sample cases of wine that were purchased by wine distribution companies after sampling a wine. These cases would be used to provide tasting samples to restaurants and wine stores around the United States. The more sample cases purchased, the more likely is a wine to be sold at a high end restaurant. A large wine manufacturer is studying the data in order to predict the number of wine cases ordered based upon the wine characteristics. If the wine manufacturer can predict the number of cases, then that manufacturer will be able to adjust their wine offering to maximize sales. Your objective is to build a count regression model to predict the number of cases of wine that will be sold given certain properties of the wine. HINT: Sometimes, the fact that a variable is missing is actually predictive of the target. You can only use the variables given to you (or variables that you derive from the variables provided). Below is a short description of the variables of interest in the data set:

Data Exploration:

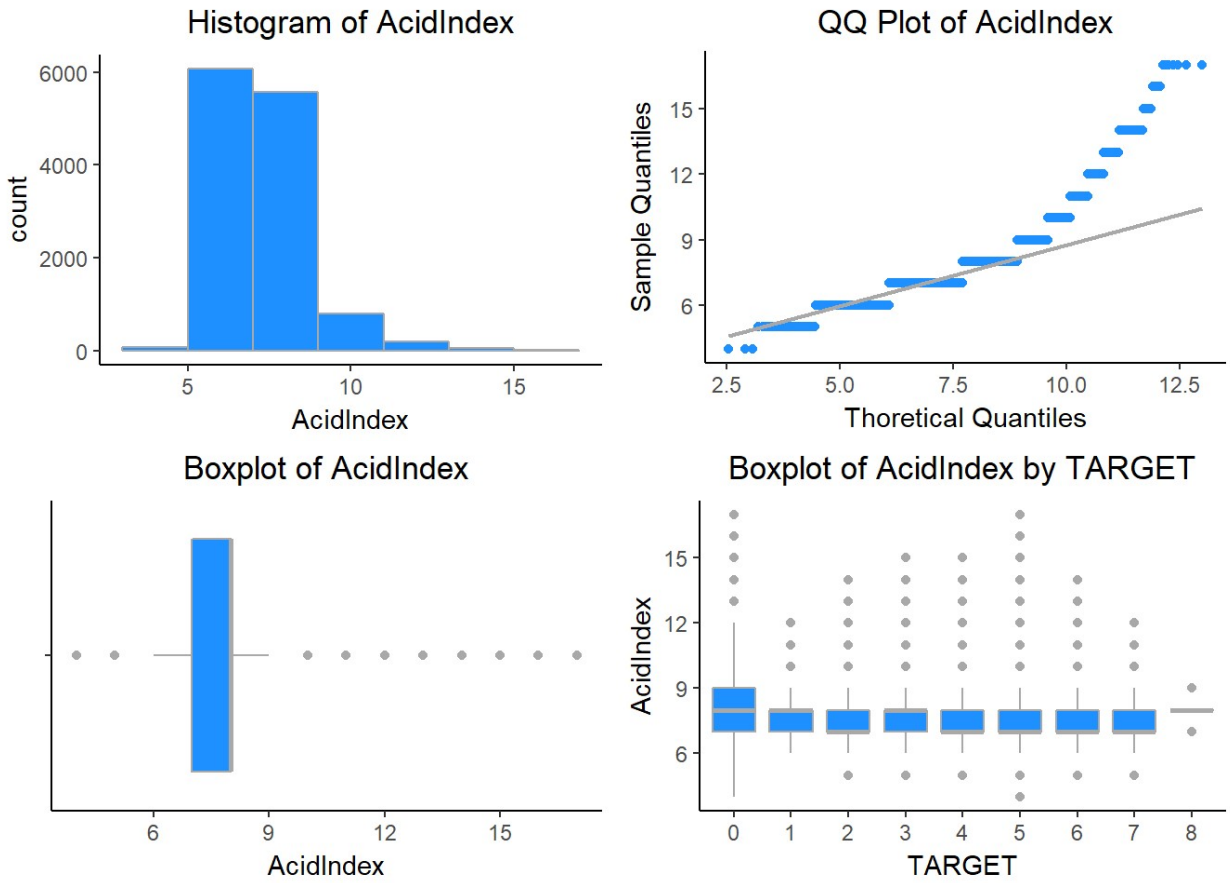
The training data set includes 12795 observations, with 16 variables: 14 predictors, 1 response variables, and one record identifier. Below is a brief description of the included variables:

First we will remove the INDEX column.

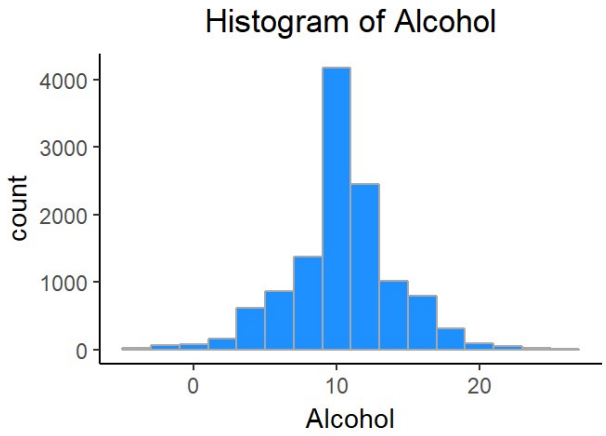
Visual Exploration:

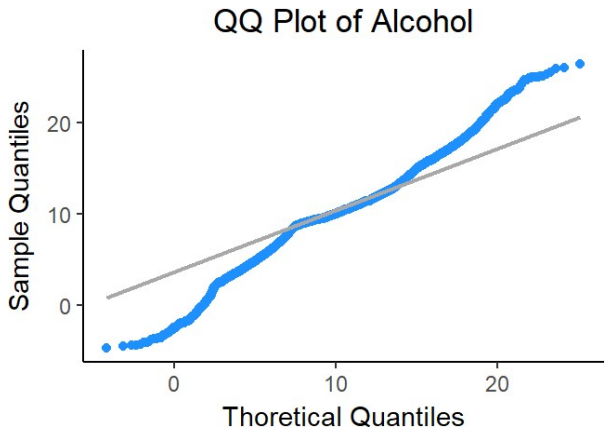
Let's dig into our available variables.

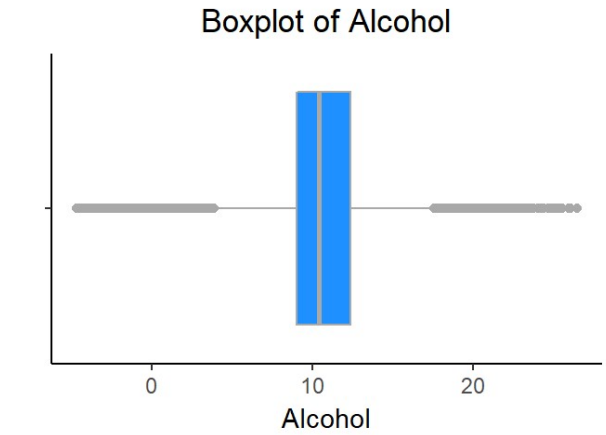
AcidIndex - Proprietary method of testing total acidity of wine by using a weighted average. From the plot below looks like is slightly right skewed. Also correlation with the target is low.

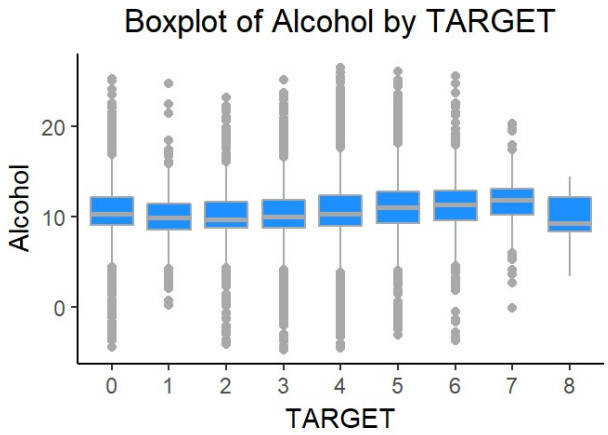


Alcohol - This variable tells us about the Alcohol content. The variable Alcohol is normally distributed.

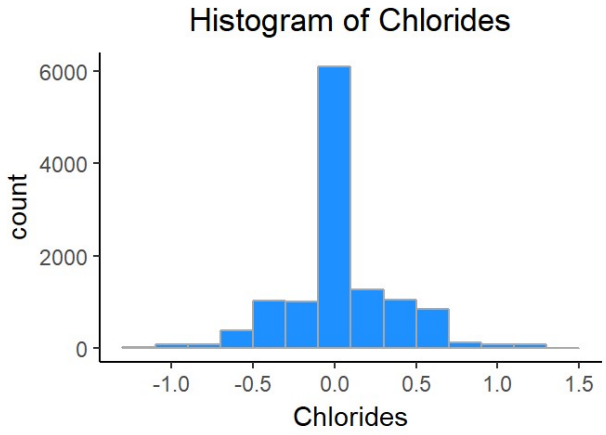

Histogram of Alcohol
count
4000
3000
2000
1000
0
0
10
20
Alcohol

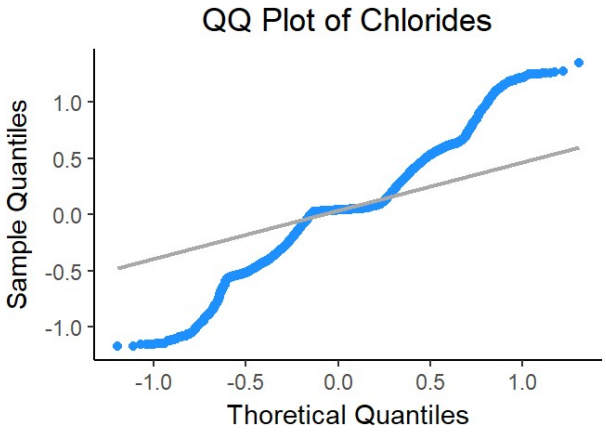

QQ Plot of Alcohol
Sample Quantiles
20
10
0
0
10
20
Thoretical Quantiles

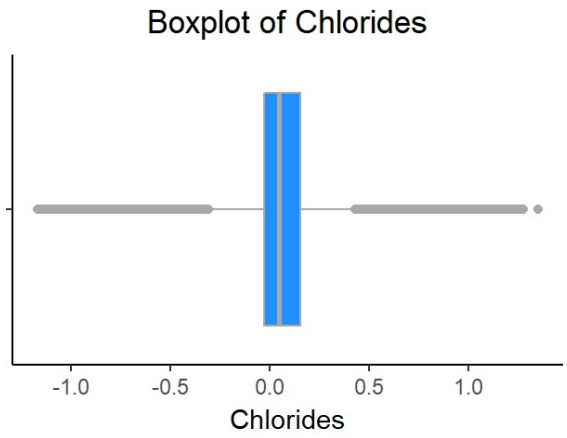

Boxplot of Alcohol
0
10
20
Alcohol

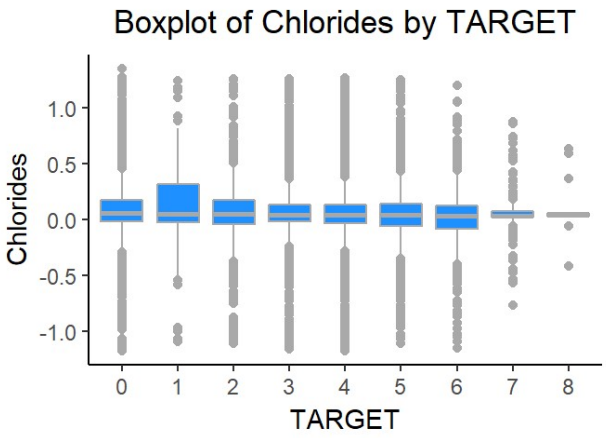

Boxplot of Alcohol by TARGET
Alcohol
20
10
0
0
1
2
3
4
5
6
7
8
TARGET

Chlorides - This variable tells us about the Chloride content of wine. The variable Chlorides is normally distributed.

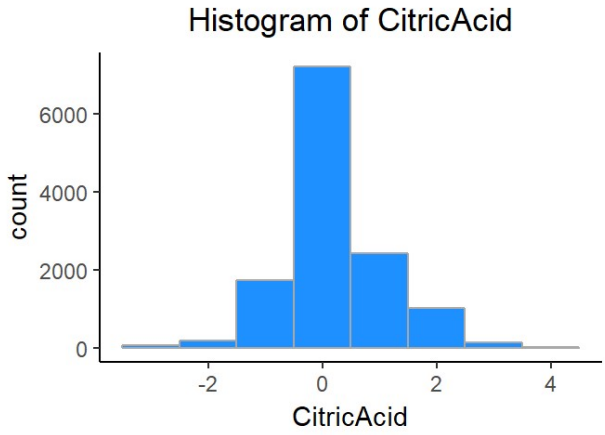


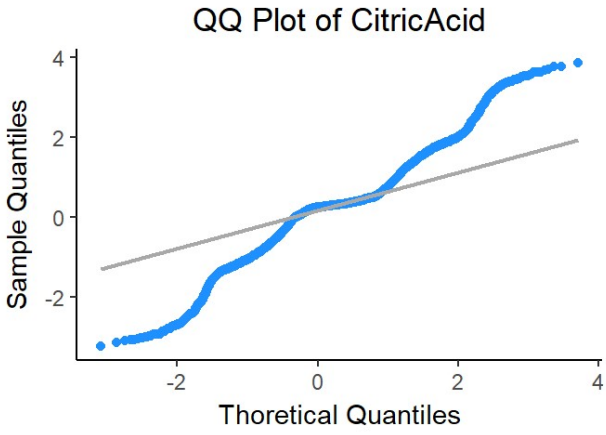


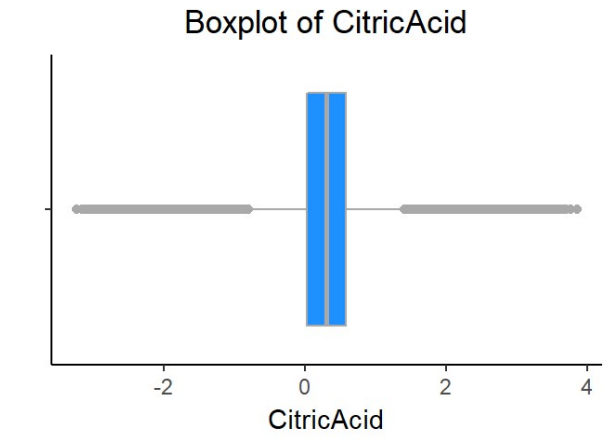


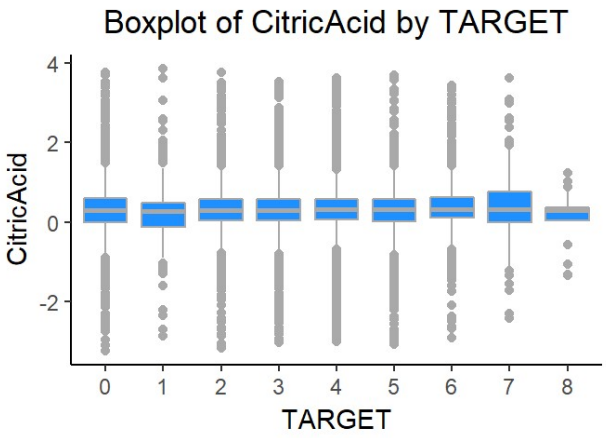


CitricAcid - This variable tells us about the Citric Acid Content of wine. This variable is also normally distributed.

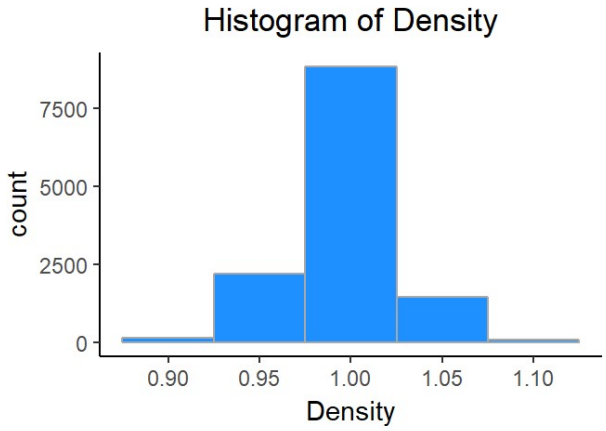


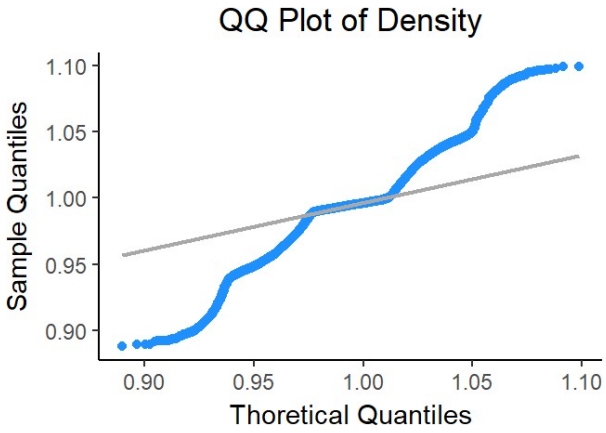


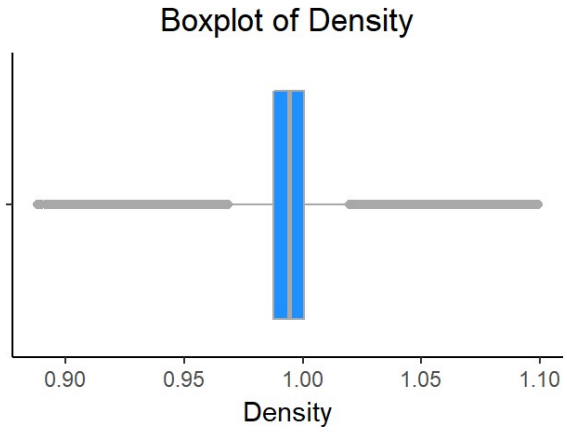


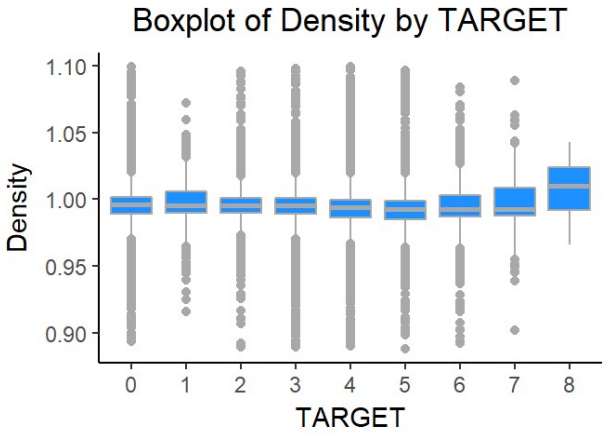


Density - This variable tells us about the Density of wine. Density is also normally distributed.

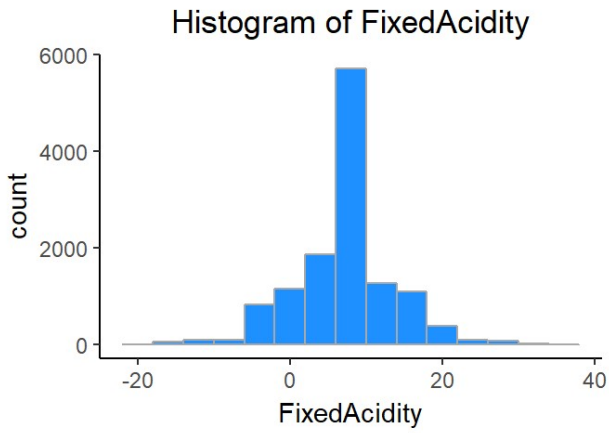


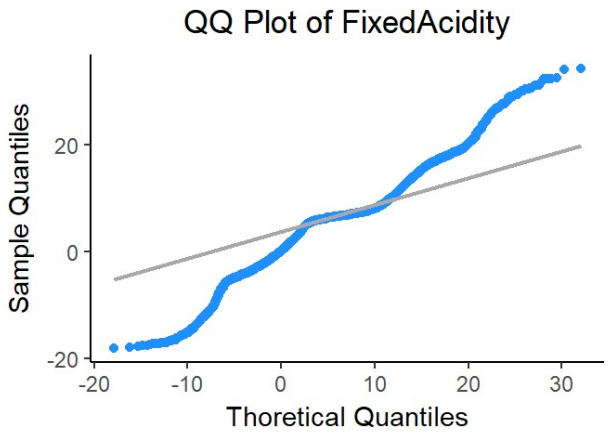


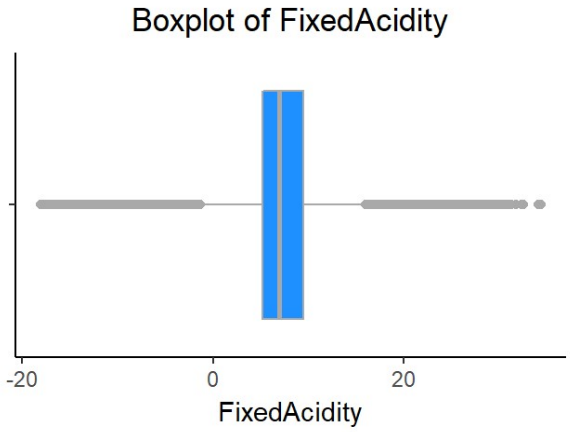


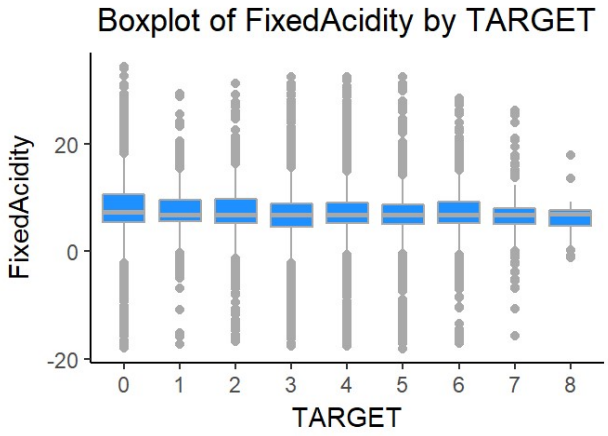


FixedAcidity - This variable tells us about the FixedAcidity of wine. Its also normally distributed.

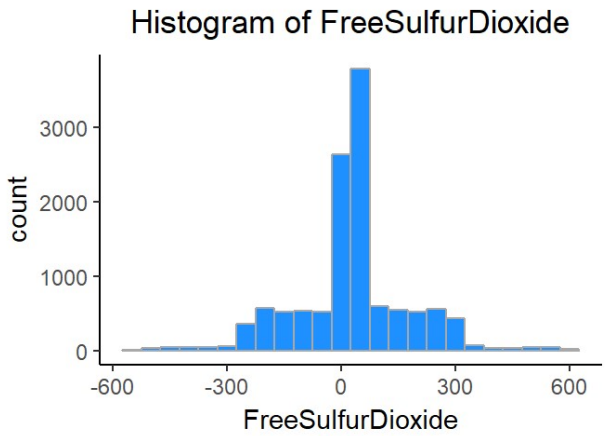


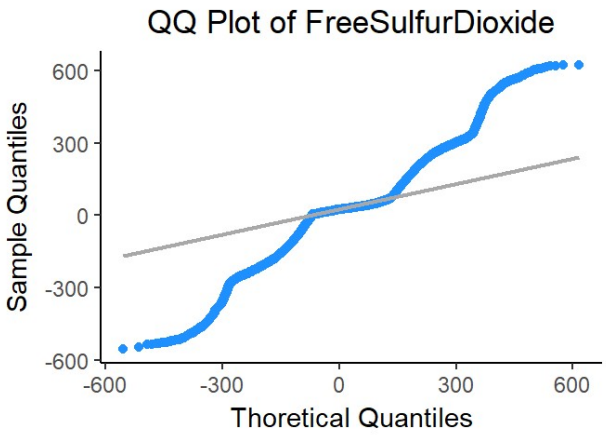


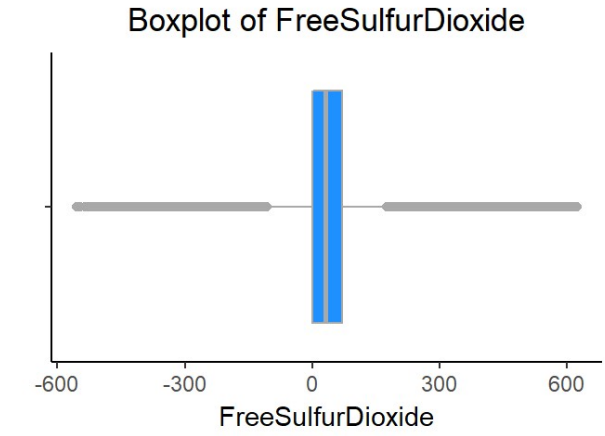


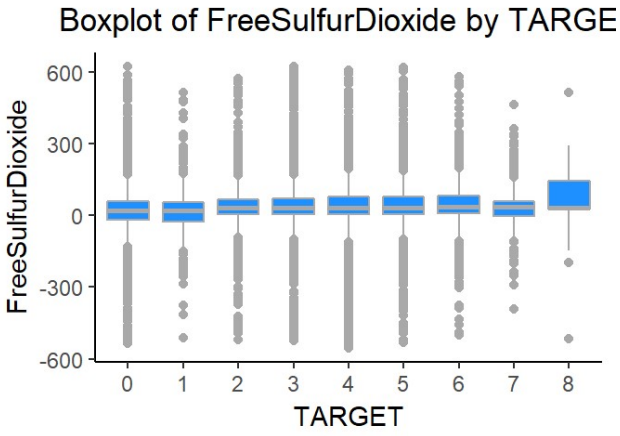


FreeSulfurDioxide - This variable tells us about the Sulfur Dioxide content of wine. It is slightly right skewed.

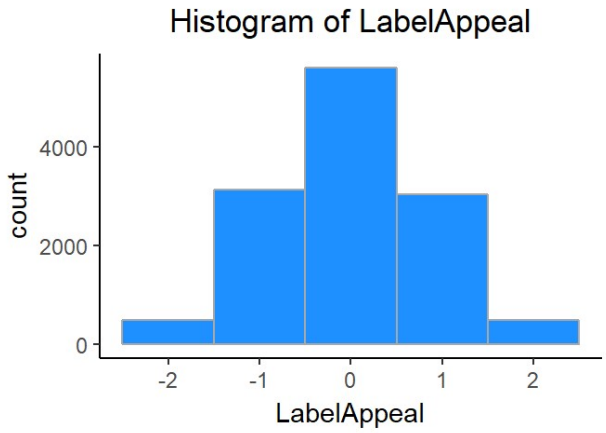


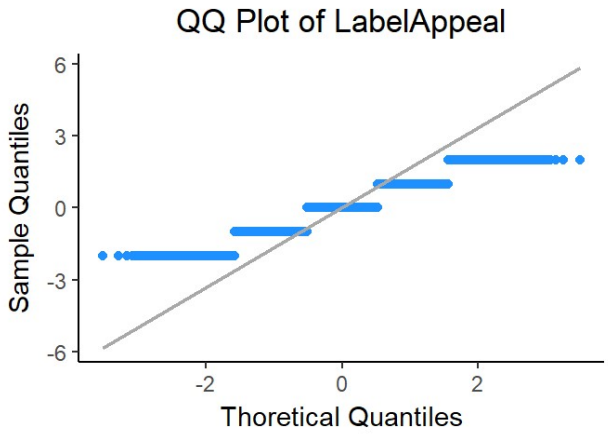


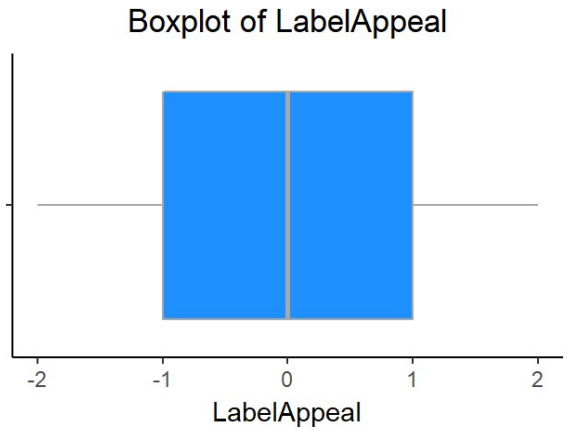


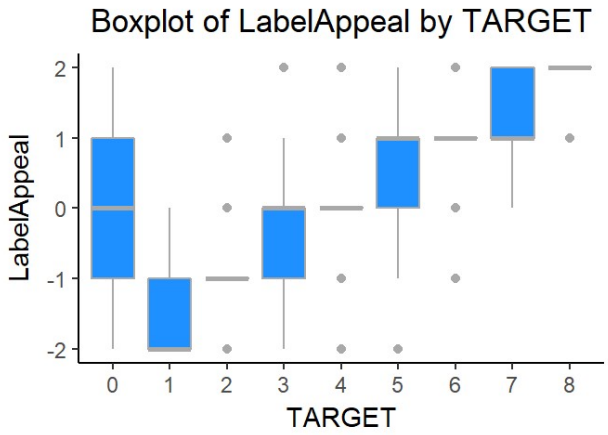


LabelAppeal - Marketing Score indicating the appeal of label design for consumers. High numbers suggest customers like the label design. Negative numbers suggest customers don't like the design. Many consumers purchase based on the visual appeal of the wine label design. Higher numbers suggest better sales.

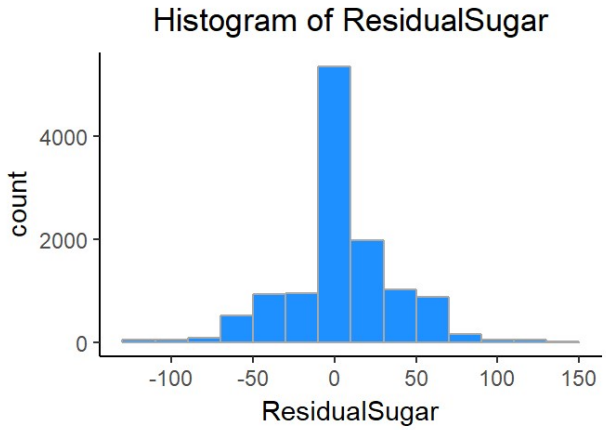


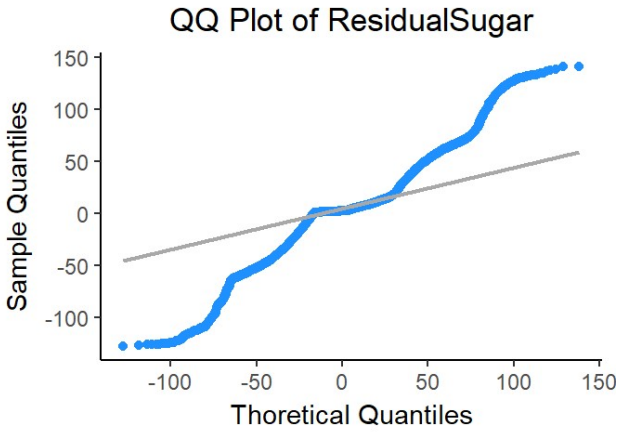


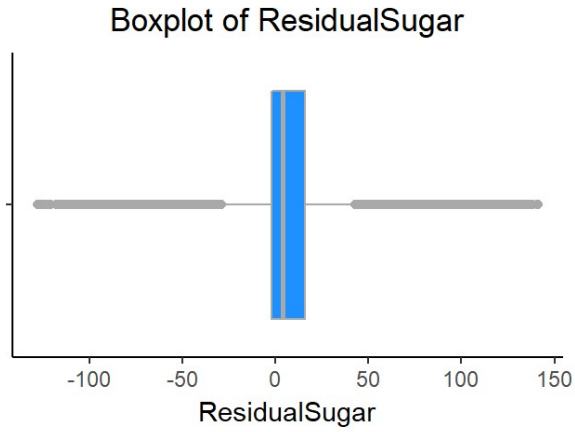


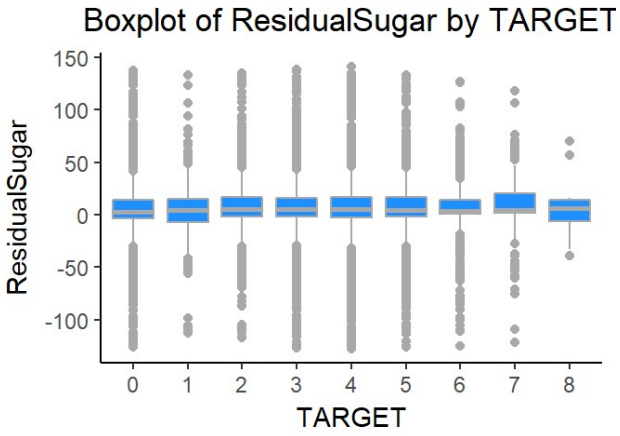


ResidualSugar - This variable tells us about the ResidualSugar of wine. ResidualSugar is also normally distributed.

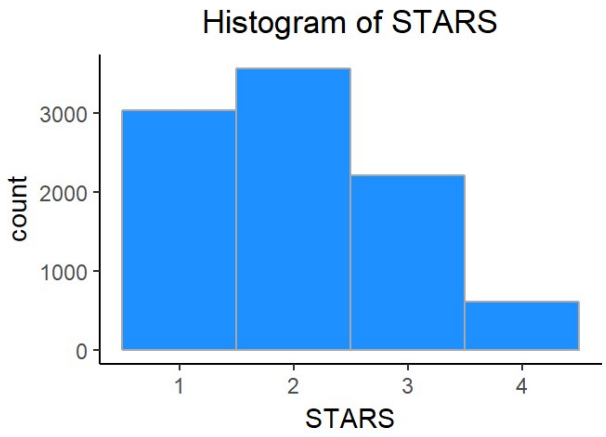


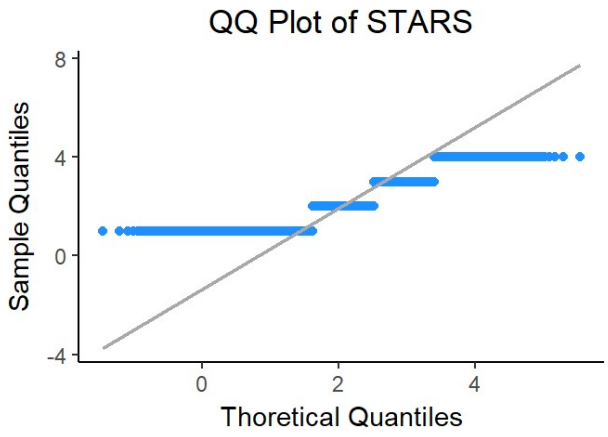


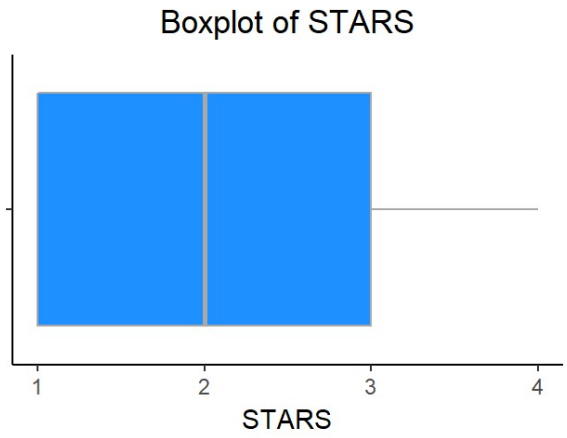


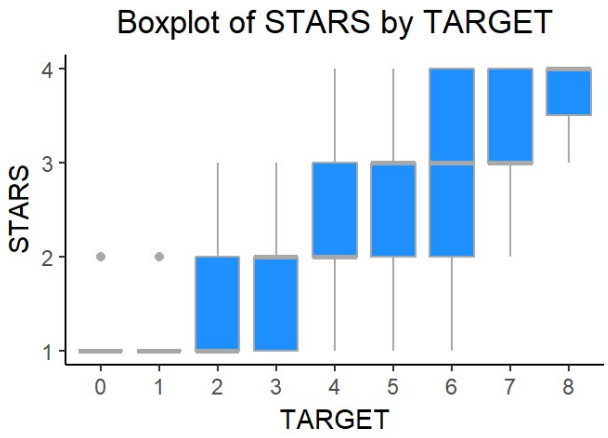


STARS - Wine rating by a team of experts. 4 Stars = Excellent, 1 Star = Poor. A high number of stars suggests high sales.

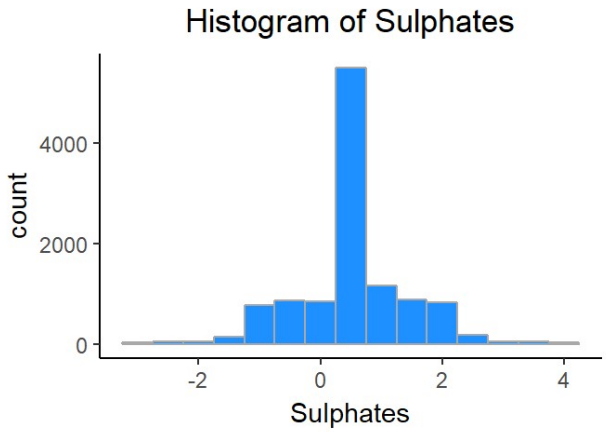


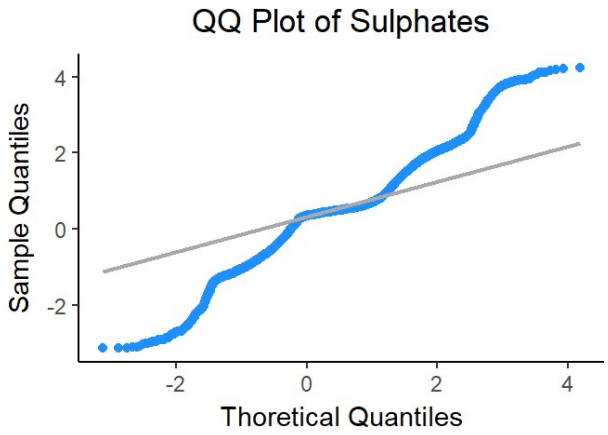


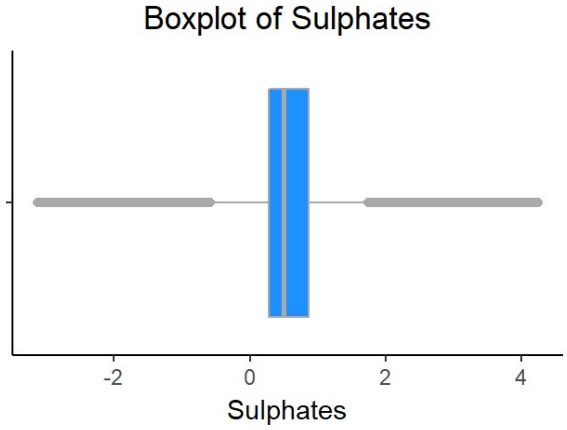


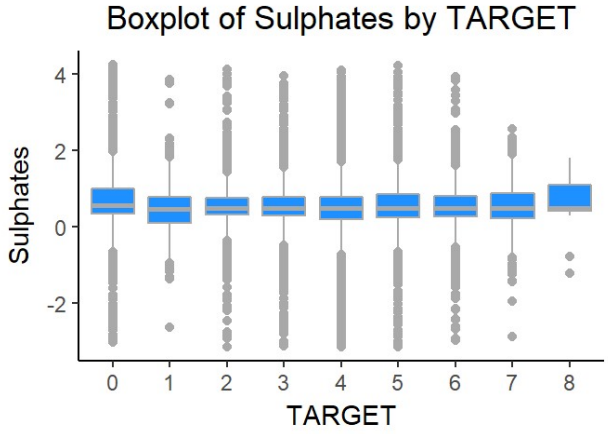


Sulphates - This variable tells us about the Sulphates content of wine.

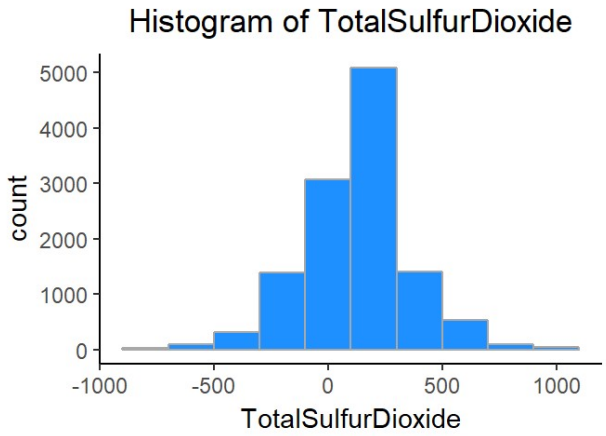


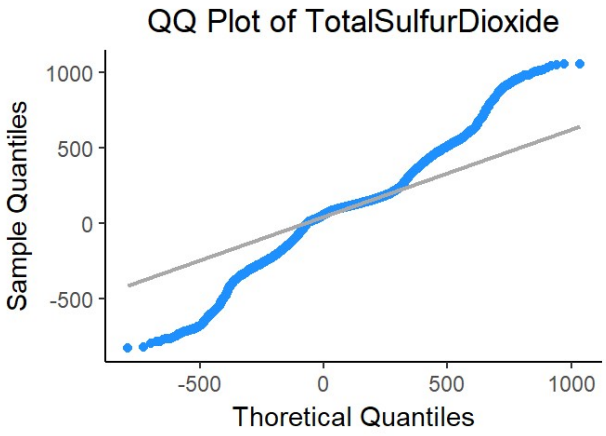


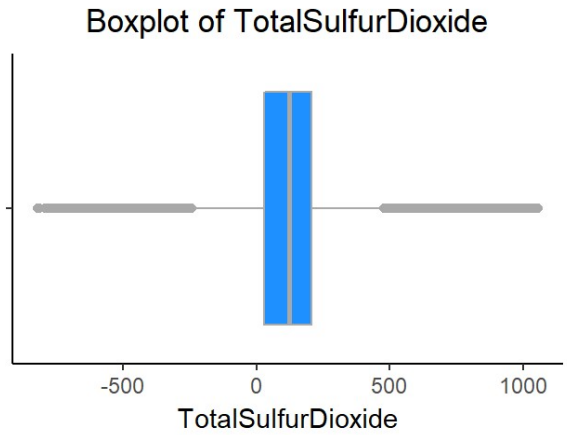


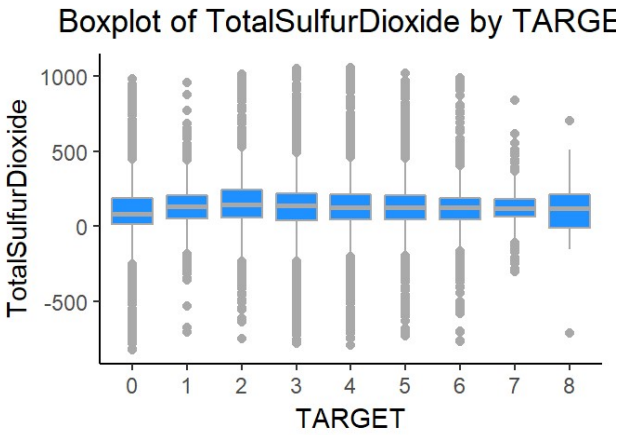


TotalSulfurDioxide - This variable tells us about the Total Sulfur Dioxide of Wine.

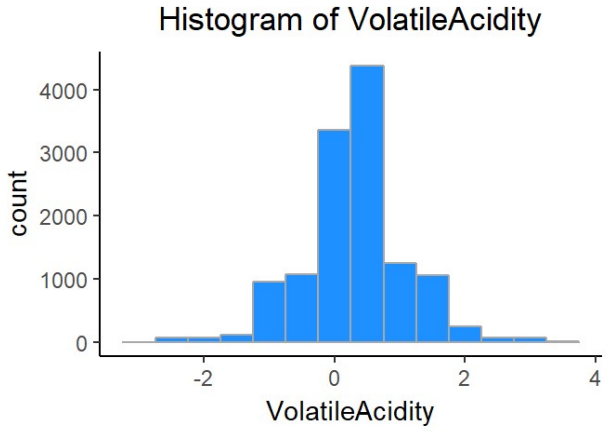


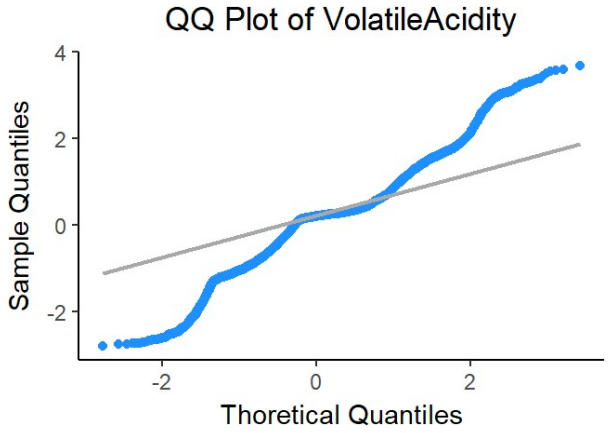


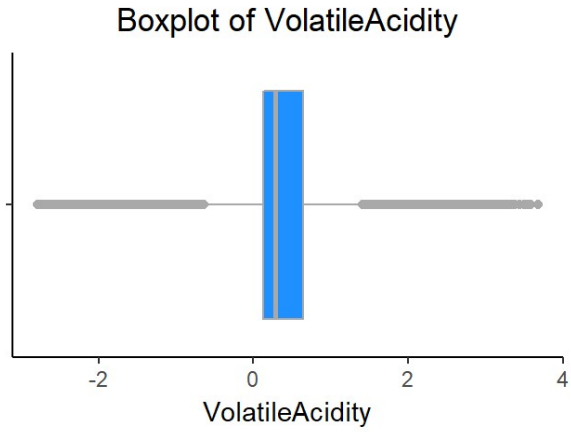


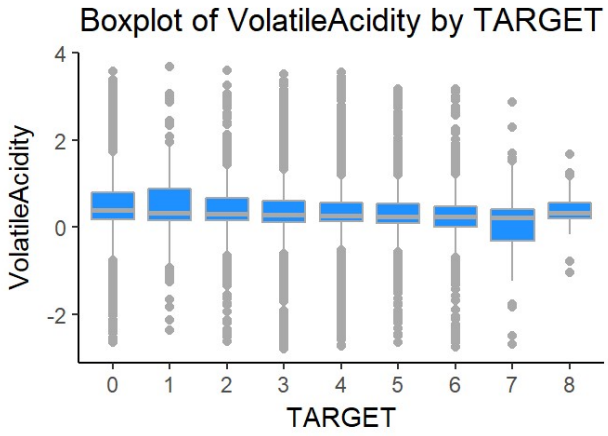


VolatileAcidity - This variable tells us about the VolatileAcidity content of Wine.

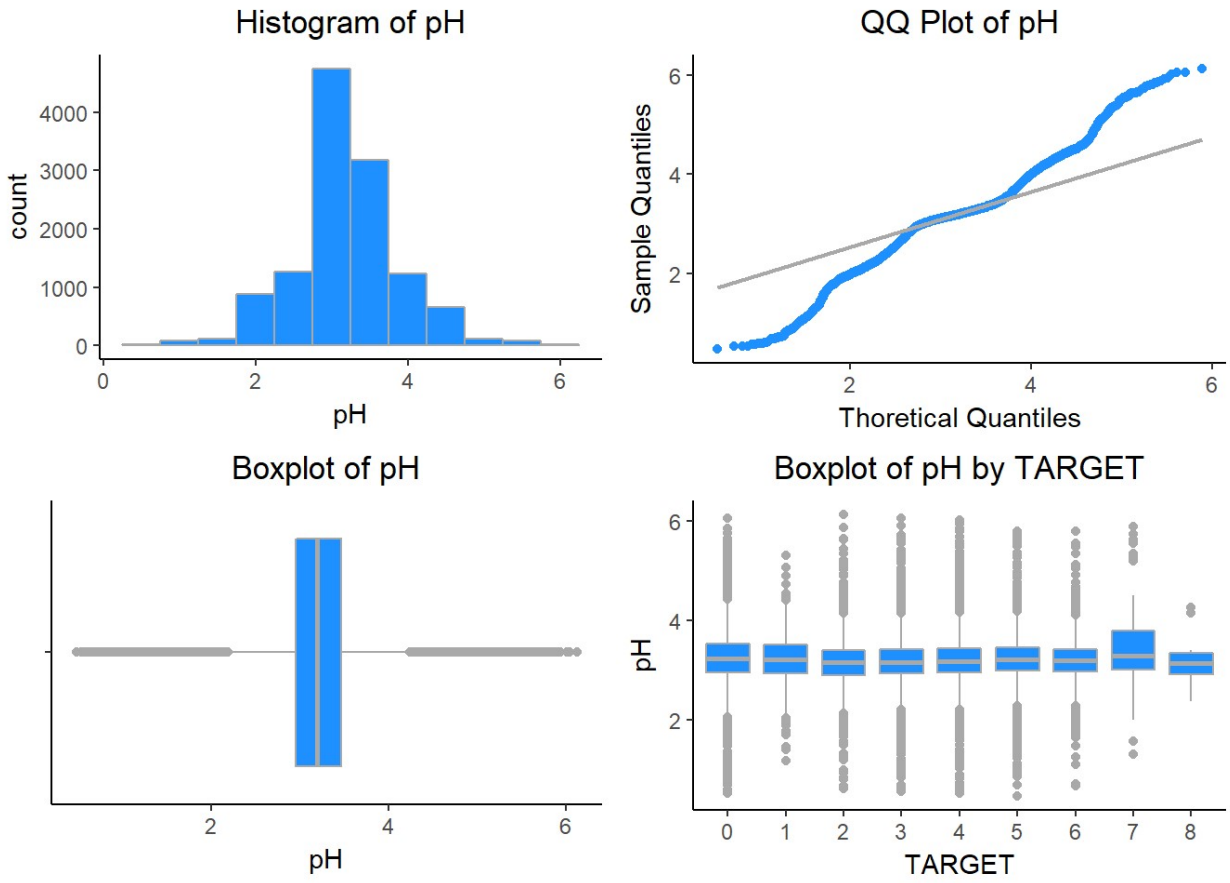






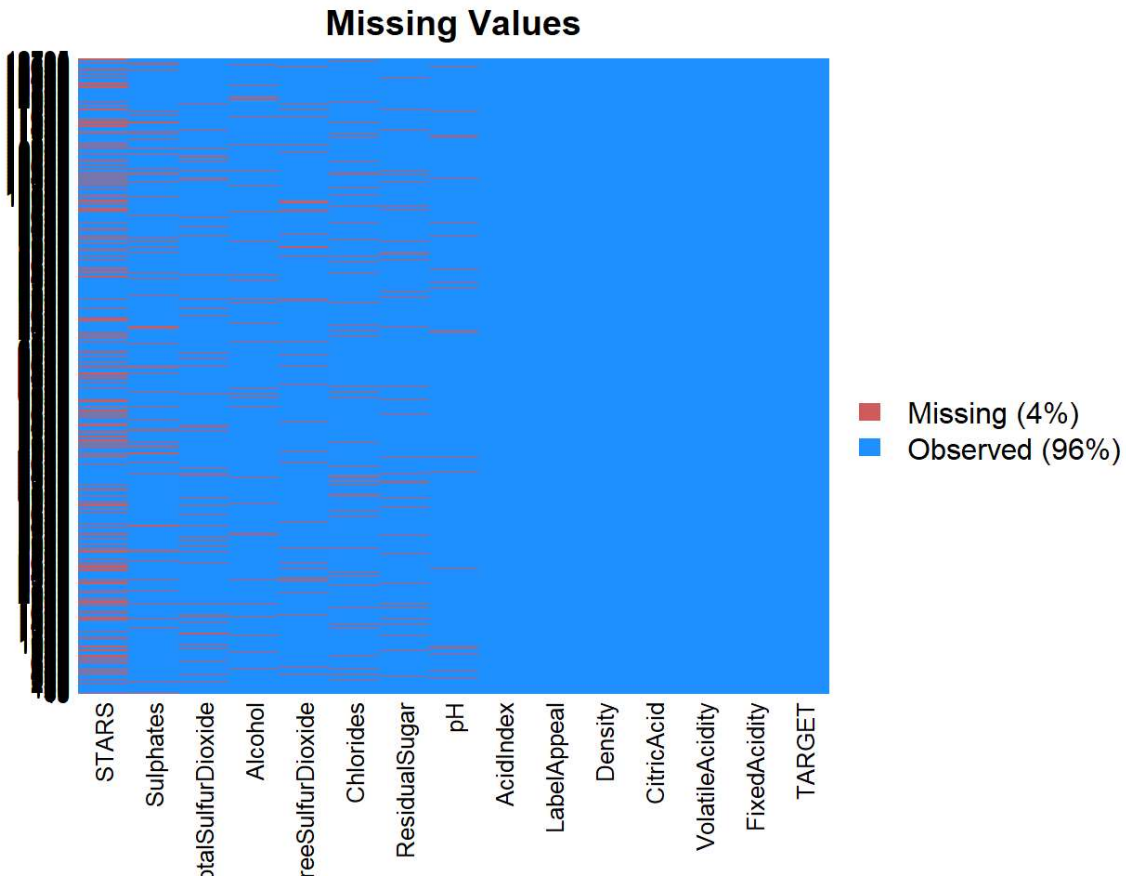


pH - This variable tells us about the pH of Wine.



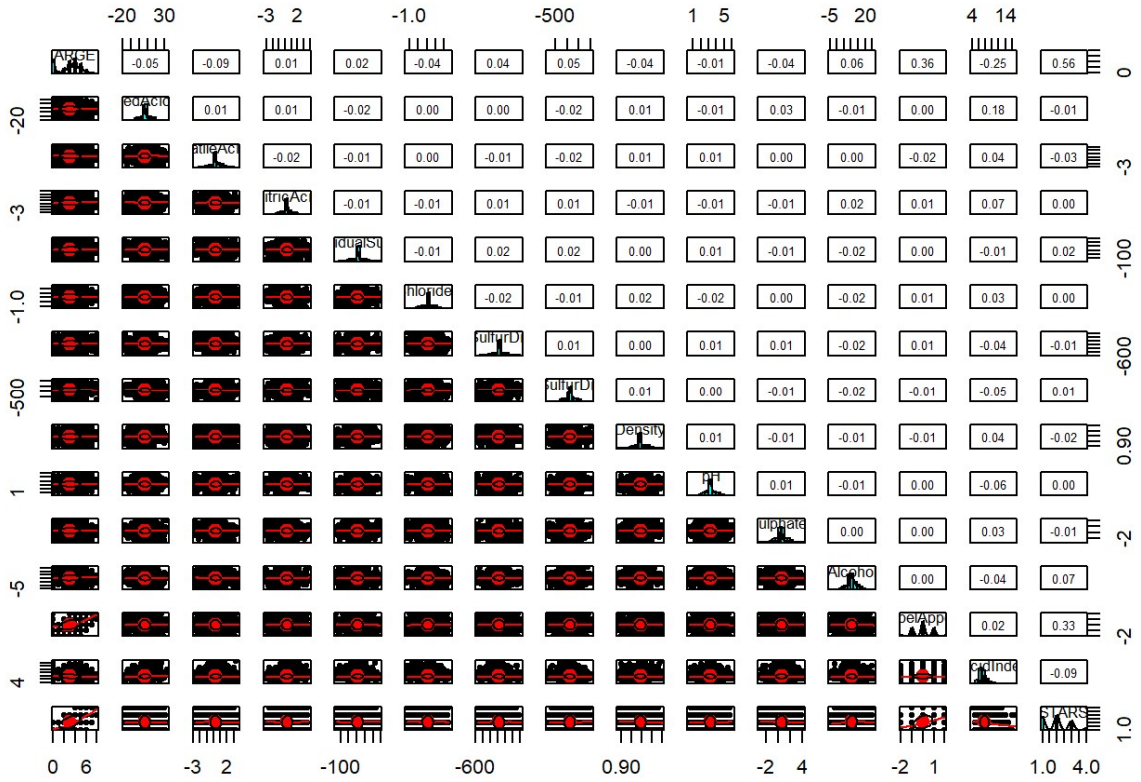
Missing Values

Now we will see the missing values in the dataset. For this i have used Amelia package. The below table shows a summary of the NA values in the data. Only STARS had an NA frequency higher than 10%, so this was a concern. There needs to be taken care while we do data preparation.



Correlations:

Finding correlations: The correlation plot below shows how variables in the dataset are related to each other. Looking at the plot, we don't see much correlations.



Now we will see the TARGET Variable.

TARGET - Number of Cases Purchased

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    StdD    Skew    Kurt
##    0.00    2.00    3.00    3.03    4.00    8.00    1.93   -0.33    2.12
```

Data Preparation:

Let's first split the data into training and test. We split the data into 80:20.

```
set.seed(999)
sampl = sample.split(wine_train$TARGET, SplitRatio = .80)
wine_train1 <- subset(wine_train, sampl == TRUE)
wine_test1 <- subset(wine_train, sampl == FALSE)
```

Also used the mice package to impute missing values. There is very low correlation between AcidIndex and TARGET, so I applied log transformation on AcidIndex.

```
wine_train2$AcidIndex <- log(wine_train2$AcidIndex)
wine_test2$AcidIndex <- log(wine_test2$AcidIndex)
```

Rest of the data looks good and don't think we need any transformation.

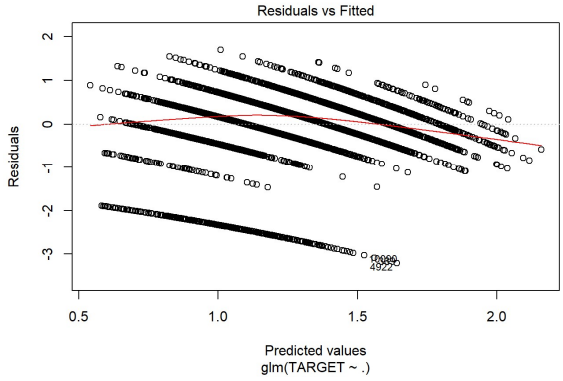
Build Models:

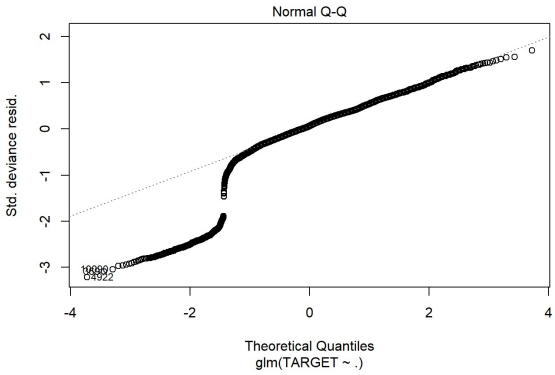
We will build a variety of models using both the imputed and non-imputed data.

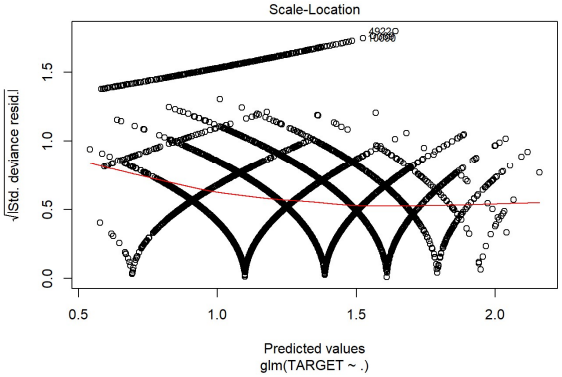
1. Poisson model without imputations.

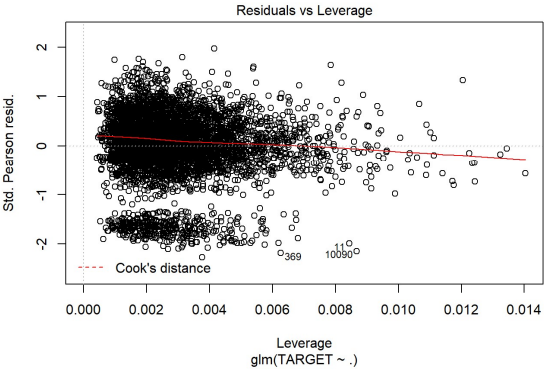
```
model1 = glm(TARGET ~  ., data=wine_train1, family=poisson)
summary(model1)
```

```
## Call:
## glm(formula = TARGET ~ ., family = poisson, data = wine_train1)
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2128  -0.2757   0.0647   0.3766   1.6981
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.608e+00  2.796e-01   5.750 8.90e-09 ***
## FixedAcidity         6.705e-04  1.177e-03   0.570  0.56901
## VolatileAcidity     -2.750e-02  9.283e-03  -2.963  0.00305 **
## CitricAcid          -3.835e-03  8.519e-03  -0.450  0.65259
## ResidualSugar        1.828e-05  2.152e-04   0.085  0.93232
## Chlorides           -3.764e-02  2.314e-02  -1.627  0.10377
## FreeSulfurDioxide    5.671e-05  4.892e-05   1.159  0.24630
## TotalSulfurDioxide   2.230e-05  3.177e-05   0.702  0.48274
## Density             -4.025e-01  2.749e-01  -1.464  0.14326
## pH                   2.307e-04  1.085e-02   0.021  0.98303
## Sulphates           -5.984e-03  7.973e-03  -0.751  0.45293
## Alcohol              3.262e-03  2.004e-03   1.628  0.10360
## LabelAppeal          1.730e-01  8.858e-03  19.530  < 2e-16 ***
## AcidIndex           -4.967e-02  6.666e-03  -7.451 9.28e-14 ***
## STARS                1.929e-01  8.328e-03  23.160  < 2e-16 ***
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Dispersion parameter for poisson family taken to be 1)
##     Null deviance: 4720.5  on 5143  degrees of freedom
## Residual deviance: 3242.8  on 5129  degrees of freedom
##   (5093 observations deleted due to missingness)
## AIC: 18545
## Number of Fisher Scoring iterations: 5
```


Residuals vs Fitted
Residuals
Predicted values
glm(TARGET ~ .)


Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
glm(TARGET ~ .)


Scale-Location
√|Std. deviance resid.|
Predicted values
glm(TARGET ~ .)

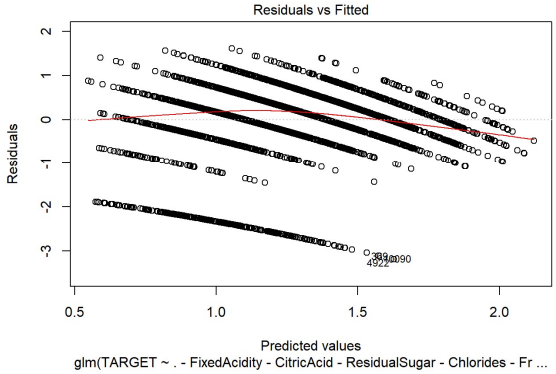

Residuals vs Leverage
Std. Pearson resid.
Cook's distance
Leverage
glm(TARGET ~ .)

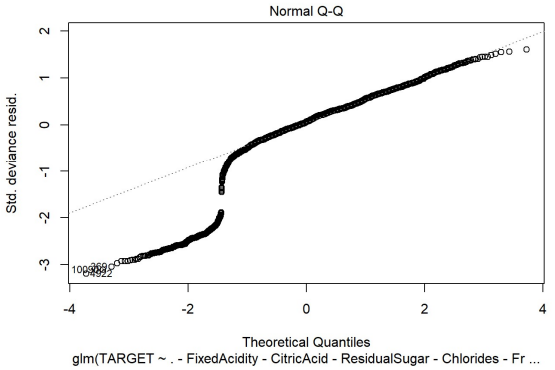
2. Poisson model without imputations and only significant variables.

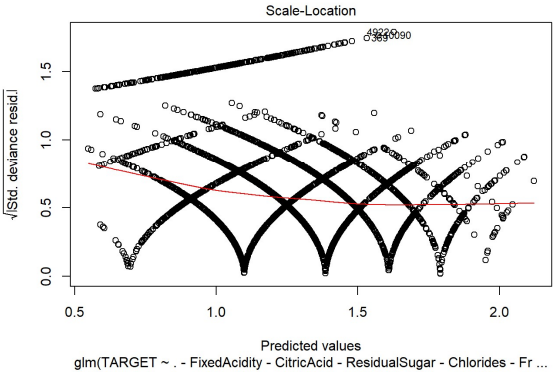
```
model2 = glm(TARGET ~  .-FixedAcidity-CitricAcid-ResidualSugar-Chlorides-FreeSulfurDioxide-TotalS
ulfurDioxide-Density-pH-Sulphates-Alcohol, data=wine_train1, family=poisson)

summary(model2)
```

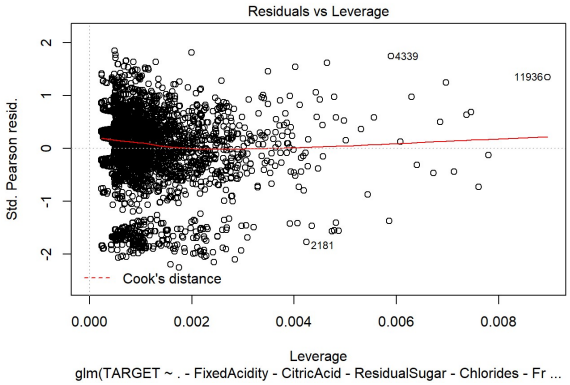
```
## Call:

## glm(formula = TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar -

##     Chlorides - FreeSulfurDioxide - TotalSulfurDioxide - Density -

##     pH - Sulphates - Alcohol, family = poisson, data = wine_train1)

##  Deviance Residuals:

##     Min       1Q   Median       3Q      Max

## -3.1898  -0.2777   0.0622   0.3764   1.6086

## Coefficients:

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

## (Intercept)       1.251442   0.054724  22.868  < 2e-16 ***

## VolatileAcidity -0.027581   0.009278  -2.973  0.00295 **

## LabelAppeal       0.173177   0.008853  19.562  < 2e-16 ***

## AcidIndex        -0.050616   0.006553  -7.724 1.13e-14 ***

## STARS             0.194208   0.008292  23.421  < 2e-16 ***

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

## (Dispersion parameter for poisson family taken to be 1)

##

##     Null deviance: 4720.5  on 5143  degrees of freedom

## Residual deviance: 3253.1  on 5139  degrees of freedom

##   (5093 observations deleted due to missingness)

## AIC: 18535

##

## Number of Fisher Scoring iterations: 5
```

```
plot(model2)
```


Residuals vs Fitted
Residuals
Predicted values
glm(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar - Chlorides - Fr ...


Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
glm(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar - Chlorides - Fr ...


Scale-Location
√|Std. deviance resid.|
Predicted values
glm(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar - Chlorides - Fr ...

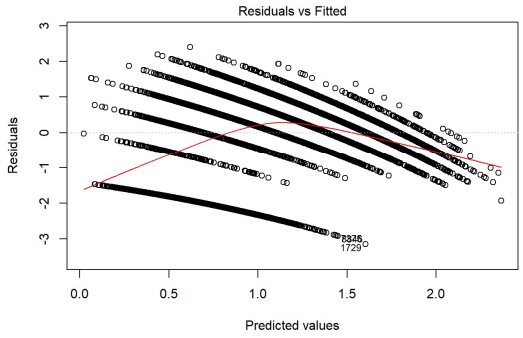

Residuals vs Leverage
Std. Pearson resid.
Cook's distance
Leverage
glm(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar - Chlorides - Fr ...

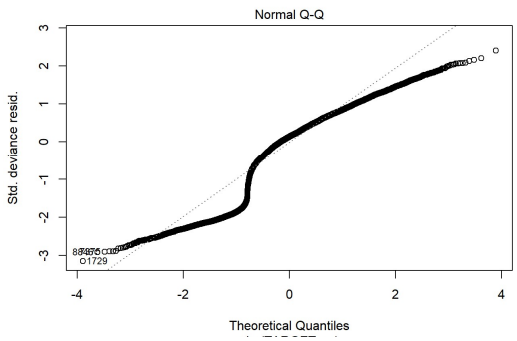
3. Poisson model with Imputation.

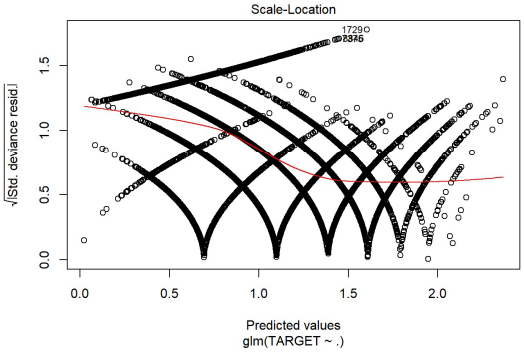
```
model3 = glm(TARGET ~  ., data=wine_train2, family=poisson)
summary(model3)
```

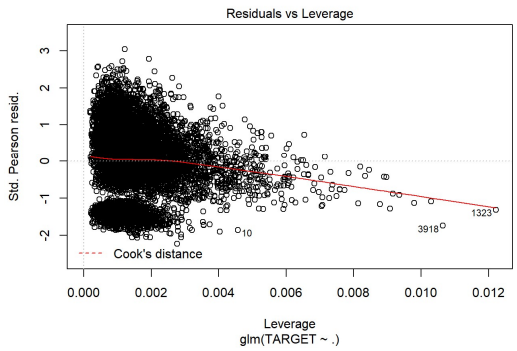
```
## Call:
## glm(formula = TARGET ~ ., family = poisson, data = wine_train2)
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1516  -0.6809   0.1304   0.6390   2.4033
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          2.382e+00  2.277e-01  10.463  < 2e-16 ***
## FixedAcidity        -1.332e-04  9.197e-04  -0.145  0.88487
## VolatileAcidity     -4.351e-02  7.275e-03  -5.982 2.21e-09 ***
## CitricAcid           8.883e-03  6.576e-03   1.351  0.17679
## ResidualSugar        1.508e-04  1.675e-04   0.900  0.36797
## Chlorides           -6.506e-02  1.791e-02  -3.633  0.00028 ***
## FreeSulfurDioxide    1.143e-04  3.804e-05   3.005  0.00266 **
## TotalSulfurDioxide   8.709e-05  2.446e-05   3.560  0.00037 ***
## Density             -4.047e-01  2.141e-01  -1.890  0.05876 .
## pH                  -1.788e-02  8.407e-03  -2.126  0.03347 *
## Sulphates           -1.327e-02  6.163e-03  -2.153  0.03129 *
## Alcohol              2.690e-03  1.546e-03   1.740  0.08187 .
## LabelAppeal          1.432e-01  6.783e-03  21.107  < 2e-16 ***
## AcidIndex           -7.622e-01  4.005e-02 -19.029  < 2e-16 ***
## STARS                3.401e-01  6.252e-03  54.395  < 2e-16 ***
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Dispersion parameter for poisson family taken to be 1)
##     Null deviance: 18291  on 10236  degrees of freedom
## Residual deviance: 12830  on 10222  degrees of freedom
## AIC: 38418
## Number of Fisher Scoring iterations: 5
```

```
plot(model3)
```


Residuals vs Fitted
Residuals
Predicted values
2375
1729


Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
1729


Scale-Location
√|Std. deviance resid.|
1729
2375
Predicted values
glm(TARGET ~ .)

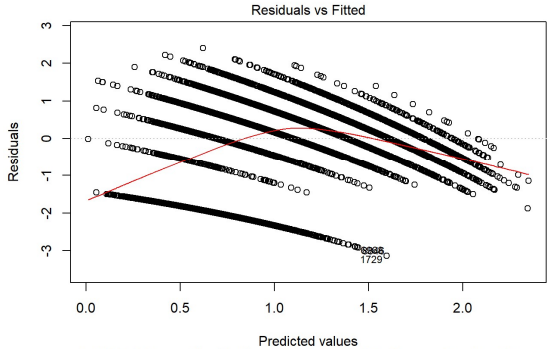

Residuals vs Leverage
Std. Pearson resid.
Cook's distance
10
3918
1323
Leverage
glm(TARGET ~ .)

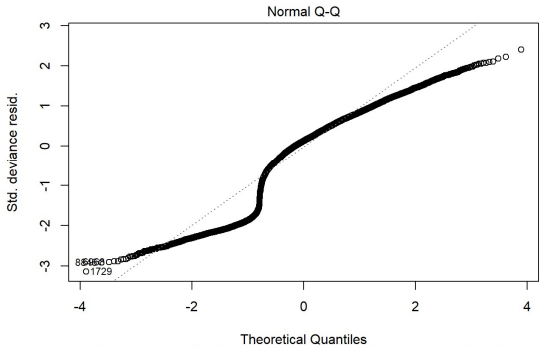
4. Poisson model with imputations and only significant variables.

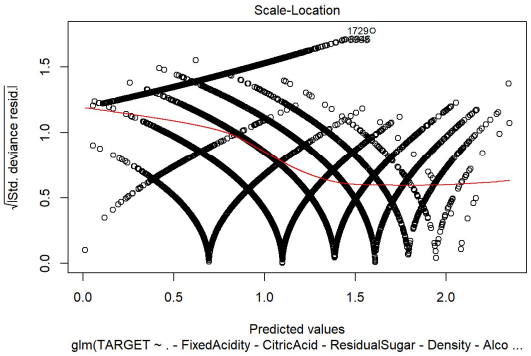
```
model4 = glm(TARGET ~  .-FixedAcidity-CitricAcid-ResidualSugar-Density-Alcohol, data=wine_train2,
family=poisson)

summary(model4)
```

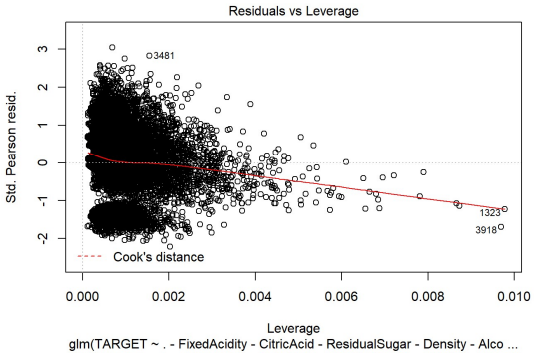
```
## Call:

## glm(formula = TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar -

##     Density - Alcohol, family = poisson, data = wine_train2)

## Deviance Residuals:

##     Min       1Q   Median       3Q      Max

## -3.1405  -0.6852   0.1288   0.6412   2.4039

## Coefficients:

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

## (Intercept)          2.019e+00  8.848e-02  22.820  < 2e-16 ***

## VolatileAcidity     -4.388e-02  7.273e-03  -6.033 1.61e-09 ***

## Chlorides           -6.711e-02  1.790e-02  -3.750 0.000177 ***

## FreeSulfurDioxide    1.119e-04  3.802e-05   2.943 0.003256 **

## TotalSulfurDioxide   8.560e-05  2.442e-05   3.505 0.000457 ***

## pH                  -1.818e-02  8.404e-03  -2.164 0.030488 *

## Sulphates           -1.327e-02  6.157e-03  -2.155 0.031143 *

## LabelAppeal          1.433e-01  6.783e-03  21.120  < 2e-16 ***

## AcidIndex           -7.665e-01  3.941e-02 -19.448  < 2e-16 ***

## STARS                3.410e-01  6.237e-03  54.673  < 2e-16 ***

## ---

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

##

## (Dispersion parameter for poisson family taken to be 1)

##

##     Null deviance: 18291  on 10236  degrees of freedom

## Residual deviance: 12839  on 10227  degrees of freedom

## AIC: 38417

##

## Number of Fisher Scoring iterations: 5
```

```
plot(model4)
```


Residuals vs Fitted
Residuals
Predicted values
1729

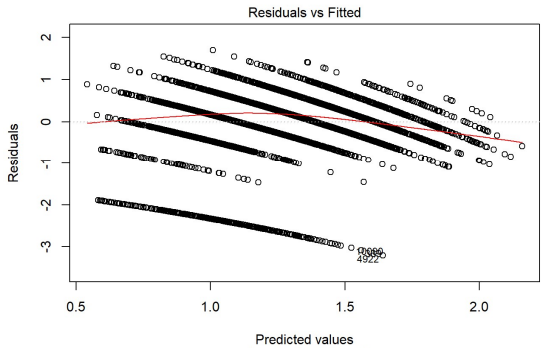

Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
1729

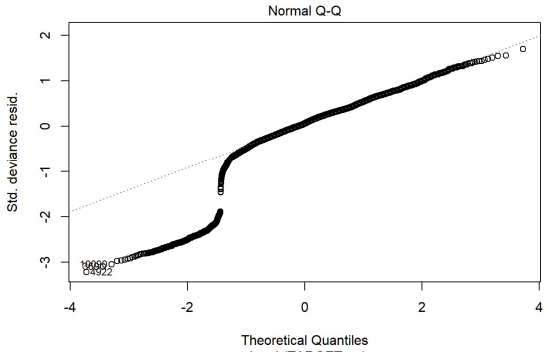

Scale-Location
√|Std. deviance resid.|
1729
Predicted values
glm(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar - Density - Alco ...

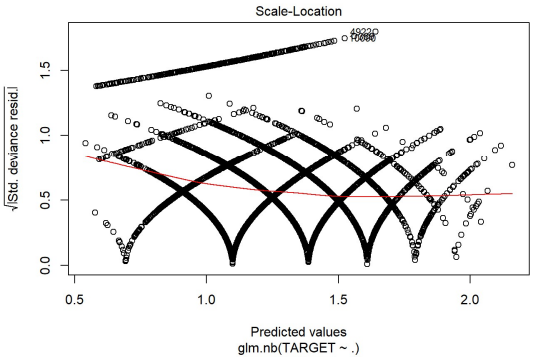

Residuals vs Leverage
Std. Pearson resid.
3481
1323
3918
Cook's distance
Leverage
glm(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar - Density - Alco ...

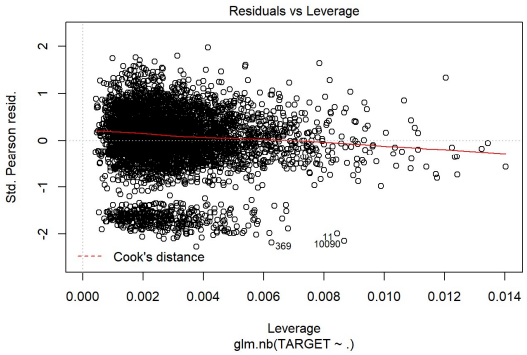
5. Negative Binomial without imputations:

```
model5 <- glm.nb(TARGET ~ ., data = wine_train1)
summary(model5)
## Call:
## glm.nb(formula = TARGET ~ ., data = wine_train1, init.theta = 138898.9107, link = log)
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2127  -0.2757   0.0647   0.3766   1.6981
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.608e+00  2.796e-01   5.750 8.91e-09 ***
## FixedAcidity         6.705e-04  1.177e-03   0.570  0.56900
## VolatileAcidity     -2.750e-02  9.283e-03  -2.963  0.00305 **
## CitricAcid          -3.835e-03  8.519e-03  -0.450  0.65259
## ResidualSugar        1.828e-05  2.152e-04   0.085  0.93231
## Chlorides           -3.764e-02  2.314e-02  -1.627  0.10378
## FreeSulfurDioxide    5.671e-05  4.892e-05   1.159  0.24630
## TotalSulfurDioxide   2.230e-05  3.177e-05   0.702  0.48275
## Density             -4.025e-01  2.750e-01  -1.464  0.14326
## pH                   2.307e-04  1.085e-02   0.021  0.98303
## Sulphates           -5.984e-03  7.973e-03  -0.751  0.45293
## Alcohol              3.262e-03  2.004e-03   1.628  0.10360
## LabelAppeal          1.730e-01  8.858e-03  19.529  < 2e-16 ***
## AcidIndex           -4.967e-02  6.666e-03  -7.451 9.28e-14 ***
## STARS                1.929e-01  8.328e-03  23.160  < 2e-16 ***
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Dispersion parameter for Negative Binomial(138898.9) family taken to be 1
##     Null deviance: 4720.4  on 5143  degrees of freedom
## Residual deviance: 3242.7  on 5129  degrees of freedom
##   (5093 observations deleted due to missingness)
## AIC: 18547
## Number of Fisher Scoring iterations: 1
##               Theta:  138899
##           Std. Err.:  259921
## Warning while fitting theta: iteration limit reached
##  2 x log-likelihood:  -18515.07
plot(model5)
```


Residuals vs Fitted
Residuals
Predicted values
10999
4922


Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
10999
4922


Scale-Location
√|Std. deviance resid.|
Predicted values
glm.nb(TARGET ~ .)
4922
10999

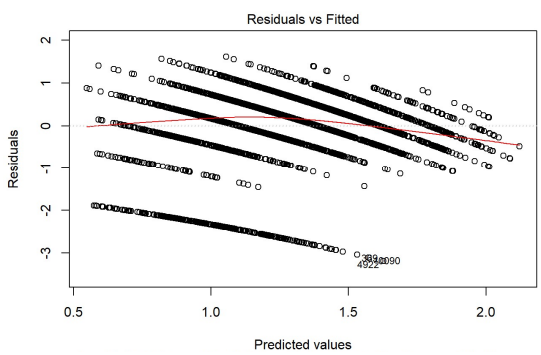

Residuals vs Leverage
Std. Pearson resid.
Cook's distance
Leverage
glm.nb(TARGET ~ .)
369
11
10990

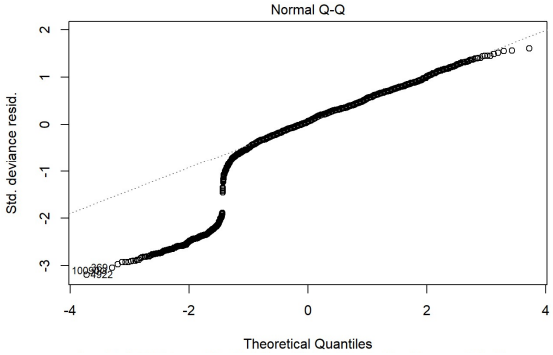
6. Negative Binomial without imputations and only significant variables:

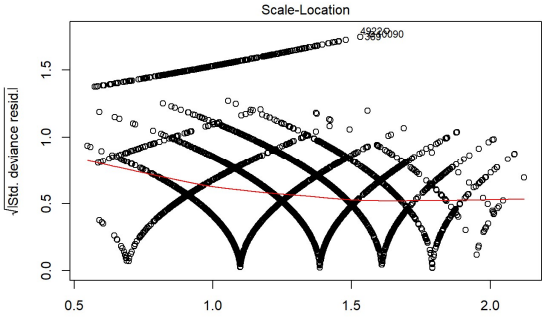
```
model6 <- glm.nb(TARGET ~ .-FixedAcidity-CitricAcid-ResidualSugar-Chlorides-FreeSulfurDioxide-Tot
alSulfurDioxide-Density-pH-Sulphates-Alcohol, data = wine_train1)

summary(model6)
```

```
## Call:
## glm.nb(formula = TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar -
##     Chlorides - FreeSulfurDioxide - TotalSulfurDioxide - Density -
##     pH - Sulphates - Alcohol, data = wine_train1, init.theta = 138402.5261,
##     link = log)
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1898  -0.2777   0.0622   0.3764   1.6086
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.251443   0.054725  22.868  < 2e-16 ***
## VolatileAcidity -0.027581   0.009279  -2.973  0.00295 **
## LabelAppeal      0.173177   0.008853  19.562  < 2e-16 ***
## AcidIndex       -0.050616   0.006553  -7.724 1.13e-14 ***
## STARS            0.194209   0.008292  23.421  < 2e-16 ***
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Dispersion parameter for Negative Binomial(138402.5) family taken to be 1)
##     Null deviance: 4720.4  on 5143  degrees of freedom
## Residual deviance: 3253.0  on 5139  degrees of freedom
##   (5093 observations deleted due to missingness)
## AIC: 18537
## Number of Fisher Scoring iterations: 1
##               Theta:  138403
##           Std. Err.:  258834
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -18525.37
```

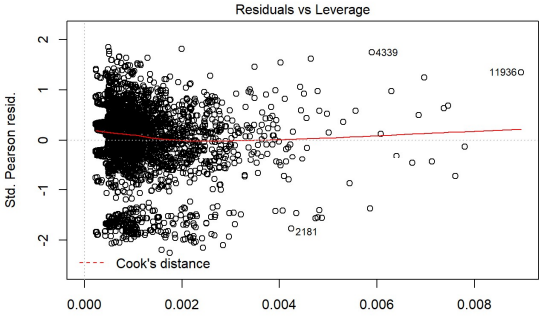
```
plot(model6)
```







glm.nb(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar - Chlorides - ...



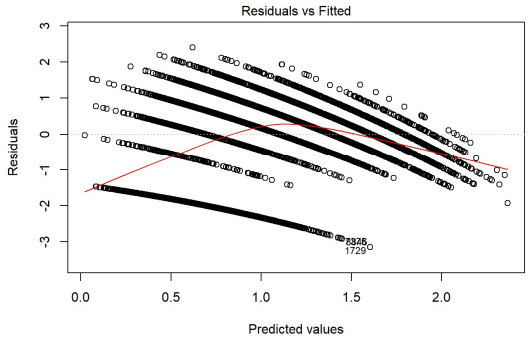
glm.nb(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar - Chlorides - ...

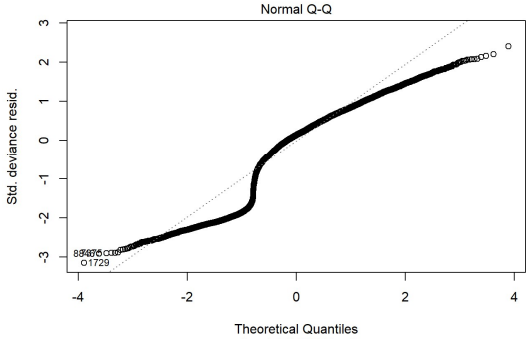
7. Negative Binomial with imputations:

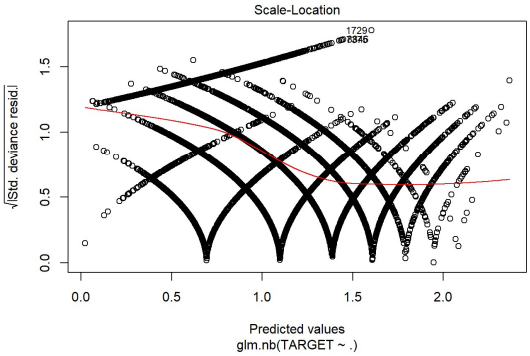
```
model7 <- glm.nb(TARGET ~ ., data = wine_train2)
summary(model7)
```

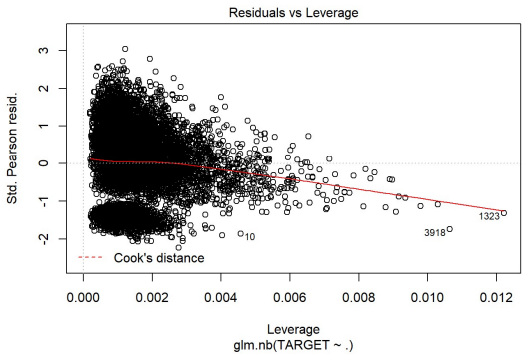
```
## Call:
## glm.nb(formula = TARGET ~ ., data = wine_train2, init.theta = 48897.24324, link = log)
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1515  -0.6808   0.1304   0.6390   2.4032
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          2.382e+00  2.277e-01  10.463  < 2e-16 ***
## FixedAcidity        -1.332e-04  9.197e-04  -0.145 0.884879
## VolatileAcidity     -4.351e-02  7.275e-03  -5.981 2.21e-09 ***
## CitricAcid           8.883e-03  6.577e-03   1.351 0.176804
## ResidualSugar        1.508e-04  1.675e-04   0.900 0.367960
## Chlorides           -6.506e-02  1.791e-02  -3.633 0.000280 ***
## FreeSulfurDioxide    1.143e-04  3.804e-05   3.005 0.002657 **
## TotalSulfurDioxide   8.709e-05  2.446e-05   3.560 0.000371 ***
## Density             -4.047e-01  2.141e-01  -1.890 0.058762 .
## pH                  -1.788e-02  8.407e-03  -2.126 0.033466 *
## Sulphates           -1.327e-02  6.164e-03  -2.153 0.031286 *
## Alcohol              2.690e-03  1.546e-03   1.740 0.081887 .
## LabelAppeal          1.432e-01  6.783e-03  21.106  < 2e-16 ***
## AcidIndex           -7.622e-01  4.005e-02 -19.029  < 2e-16 ***
## STARS                3.401e-01  6.252e-03  54.393  < 2e-16 ***
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Dispersion parameter for Negative Binomial(48897.24) family taken to be 1)
##     Null deviance: 18290  on 10236  degrees of freedom
## Residual deviance: 12830  on 10222  degrees of freedom
## AIC: 38420
## Number of Fisher Scoring iterations: 1
##               Theta:  48897
##           Std. Err.:  63448
## Warning while fitting theta: iteration limit reached
##  2 x log-likelihood:  -38388.3
```

```
plot(model7)
```


Residuals vs Fitted
Residuals
Predicted values
8325
1729


Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
1729


Scale-Location
√|Std. deviance resid.|
1729
8346
Predicted values
glm.nb(TARGET ~ .)

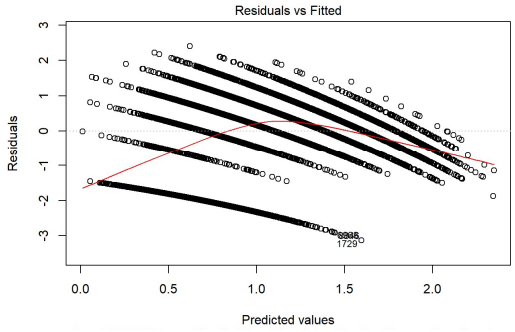

Residuals vs Leverage
Std. Pearson resid.
1323
3918
10
Cook's distance
Leverage
glm.nb(TARGET ~ .)

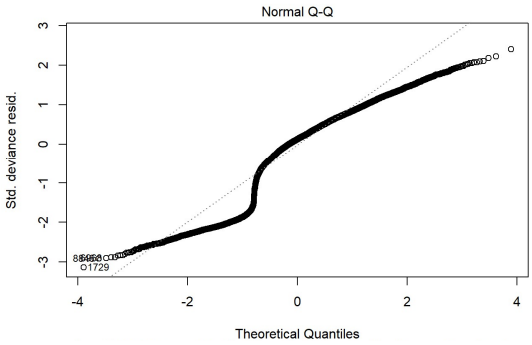
8. Negative Binomial with imputations and only significant variables:

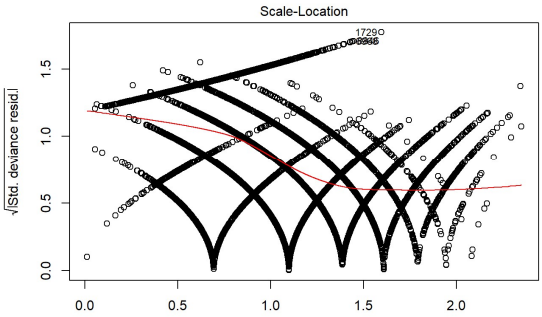
```
model8 <- glm.nb(TARGET ~ .-FixedAcidity-CitricAcid-ResidualSugar-Density-Alcohol, data = wine_tr
ain2)

summary(model8)
```

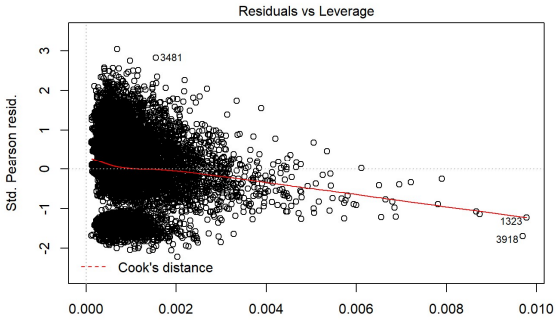
```
## Call:

## glm.nb(formula = TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar -

##     Density - Alcohol, data = wine_train2, init.theta = 48805.90033,

##     link = log)

## Deviance Residuals:

##     Min       1Q   Median       3Q      Max

## -3.1405  -0.6852   0.1288   0.6412   2.4038

## Coefficients:

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

## (Intercept)          2.019e+00  8.849e-02  22.820  < 2e-16 ***

## VolatileAcidity     -4.388e-02  7.273e-03  -6.033 1.61e-09 ***

## Chlorides           -6.711e-02  1.790e-02  -3.750 0.000177 ***

## FreeSulfurDioxide    1.119e-04  3.802e-05   2.942 0.003257 **

## TotalSulfurDioxide   8.561e-05  2.443e-05   3.505 0.000457 ***

## pH                  -1.818e-02  8.404e-03  -2.164 0.030489 *

## Sulphates           -1.327e-02  6.157e-03  -2.155 0.031144 *

## LabelAppeal          1.433e-01  6.783e-03  21.119  < 2e-16 ***

## AcidIndex           -7.665e-01  3.941e-02 -19.447  < 2e-16 ***

## STARS                3.410e-01  6.237e-03  54.671  < 2e-16 ***

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

## (Dispersion parameter for Negative Binomial(48805.9) family taken to be 1)

##     Null deviance: 18290  on 10236  degrees of freedom

## Residual deviance: 12839  on 10227  degrees of freedom

## AIC: 38420

## Number of Fisher Scoring iterations: 1

##               Theta:  48806

##           Std. Err.:  63368

## Warning while fitting theta: iteration limit reached

##

##  2 x log-likelihood:  -38397.65
```

```
plot(model8)
```


Residuals vs Fitted
Residuals
Predicted values
1729


Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
1729


Scale-Location
√|Std. deviance resid.|
1729
Predicted values
glm.nb(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar - Density - A ...

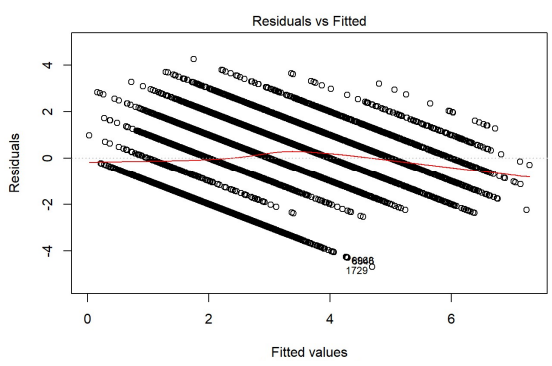

Residuals vs Leverage
Std. Pearson resid.
3481
1323
3918
Cook's distance
Leverage
glm.nb(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar - Density - A ...

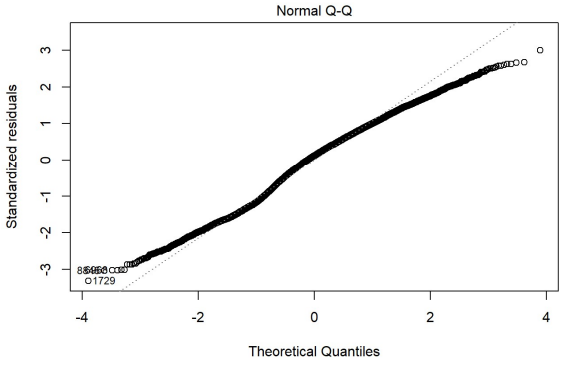
9. Linear Model with imputations.

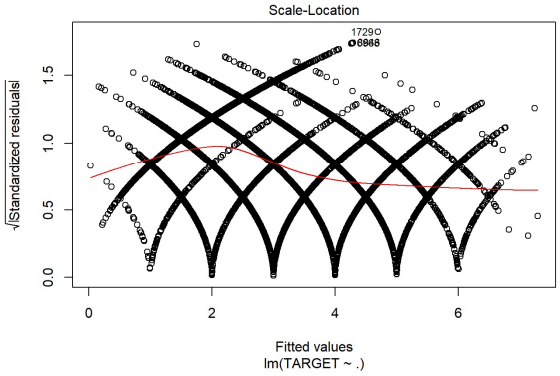
```
model9 <- lm(TARGET ~ ., data = wine_train2)
summary(model9)
```

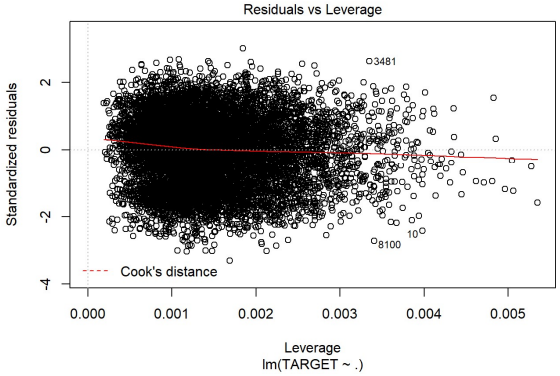
```
##
## Call:
## lm(formula = TARGET ~ ., data = wine_train2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6944 -1.0191  0.1692  1.0335  4.2502
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.149e+00  5.564e-01  11.052  < 2e-16 ***
## FixedAcidity        -1.428e-04  2.255e-03  -0.063  0.94952
## VolatileAcidity     -1.265e-01  1.792e-02  -7.056 1.82e-12 ***
## CitricAcid           2.771e-02  1.630e-02   1.699  0.08927 .
## ResidualSugar        4.479e-04  4.138e-04   1.083  0.27904
## Chlorides           -1.956e-01  4.398e-02  -4.448 8.77e-06 ***
## FreeSulfurDioxide    2.930e-04  9.398e-05   3.117  0.00183 **
## TotalSulfurDioxide   2.365e-04  6.006e-05   3.938 8.28e-05 ***
## Density             -1.099e+00  5.263e-01  -2.088  0.03678 *
## pH                  -4.064e-02  2.071e-02  -1.962  0.04978 *
## Sulphates           -3.621e-02  1.519e-02  -2.384  0.01713 *
## Alcohol              1.131e-02  3.782e-03   2.991  0.00279 **
## LabelAppeal          4.379e-01  1.644e-02  26.633  < 2e-16 ***
## AcidIndex           -2.041e+00  9.250e-02 -22.067  < 2e-16 ***
## STARS                1.162e+00  1.665e-02  69.754  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.417 on 10222 degrees of freedom
## Multiple R-squared:  0.4598, Adjusted R-squared:  0.4591
## F-statistic: 621.5 on 14 and 10222 DF,  p-value: < 2.2e-16
```

```
plot(model9)
```


Residuals vs Fitted
Residuals
Fitted values
6988
1729


Normal Q-Q
Standardized residuals
Theoretical Quantiles
1729


Scale-Location
√|Standardized residuals|
Fitted values
lm(TARGET ~ .)
1729
6988

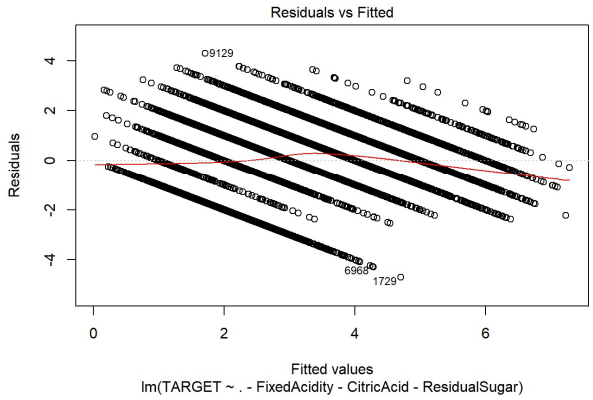

Residuals vs Leverage
Standardized residuals
Leverage
lm(TARGET ~ .)
Cook's distance
3481
8100
10

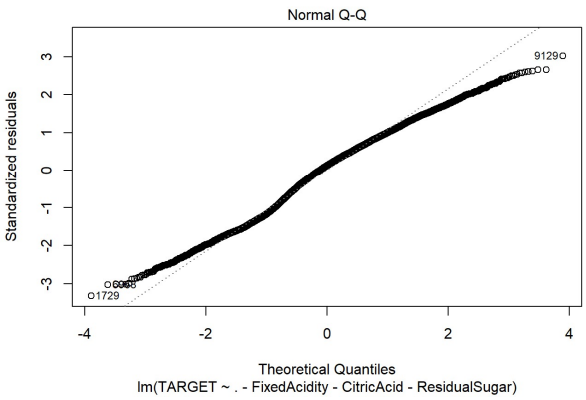
10. Linear Model with imputations and only significant variables.

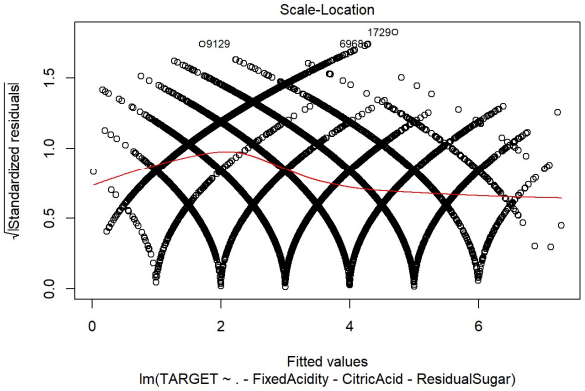
```
model10 <- lm(TARGET ~ .-FixedAcidity-CitricAcid-ResidualSugar, data = wine_train2)
summary(model10)
```

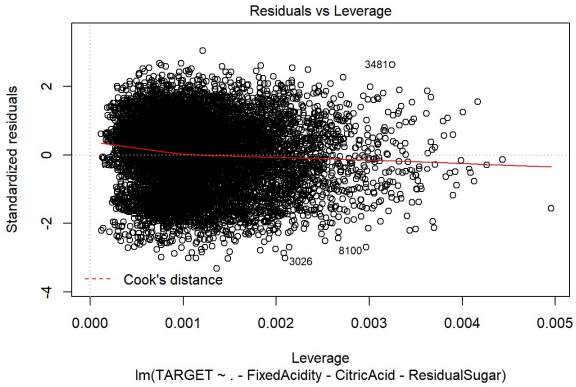
```
##
## Call:
## lm(formula = TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar,
##     data = wine_train2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7075 -1.0195  0.1718  1.0343  4.2907
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.139e+00  5.563e-01  11.036  < 2e-16 ***
## VolatileAcidity     -1.273e-01  1.792e-02  -7.104 1.30e-12 ***
## Chlorides           -1.970e-01  4.397e-02  -4.479 7.57e-06 ***
## FreeSulfurDioxide    2.939e-04  9.397e-05   3.128  0.00177 **
## TotalSulfurDioxide   2.389e-04  6.003e-05   3.980 6.94e-05 ***
## Density             -1.101e+00  5.263e-01  -2.093  0.03638 *
## pH                  -4.059e-02  2.071e-02  -1.960  0.05008 .
## Sulphates           -3.699e-02  1.517e-02  -2.437  0.01481 *
## Alcohol              1.136e-02  3.781e-03   3.005  0.00266 **
## LabelAppeal          4.379e-01  1.644e-02  26.632  < 2e-16 ***
## AcidIndex           -2.031e+00  9.085e-02 -22.351  < 2e-16 ***
## STARS                1.162e+00  1.665e-02  69.794  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.417 on 10225 degrees of freedom
## Multiple R-squared:  0.4596, Adjusted R-squared:  0.459
## F-statistic: 790.6 on 11 and 10225 DF,  p-value: < 2.2e-16
```

```
plot(model10)
```


Residuals vs Fitted
Residuals
Fitted values
lm(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar)


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar)


Scale-Location
√|Standardized residuals|
Fitted values
lm(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar)


Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage
lm(TARGET ~ . - FixedAcidity - CitricAcid - ResidualSugar)

Now let's see the output of the Models using test data:

We will use the squared loss to validate the model.

```
modelValidation <- function(mod, test){
  preds = predict(mod, test)
  diffMat = as.numeric(preds) - as.numeric(test$TARGET)
  diffMat = diffMat^2
  loss <- mean(diffMat)
  return(loss)
}
```

Poisson model with imputations.

```
modelValidation(model3, wine_test2)
```

```
## [1] 6.852209
```

Poisson model with imputations and only significant variables.

```
modelValidation(model4, wine_test2)
```

```
## [1] 6.854547
```

Negative Binomial with imputations:.

```
modelValidation(model7, wine_test2)
```

```
## [1] 6.852205
```

Negative Binomial with imputations and only significant variables.

```
modelValidation(model8, wine_test2)
```

```
## [1] 6.854543
```

Linear Model with imputations.

```
modelValidation(model9, wine_test2)
```

```
## [1] 2.029061
```

Linear Model with imputations and only significant variables.

```
modelValidation(model10, wine_test2)
```

```
## [1] 2.030002
```

Model Selection:

From the above models, i would like to go with Model10 - Linear Model with imputations and only significant variables as it uses less variables and is parsimonious. Also the R2 looks fine. The squared loss is also fine.

Prediction:

We will use the same method to impute and use log transformation for AcidIndex.

```
wine_eval <- read.csv("https://raw.githubusercontent.com/Riteshlohiya/Data621-Assignment-5/master
/wine_evaluation_data.csv", stringsAsFactors = FALSE)


do_factors <- function(wine_instance){
  wine_instance <- within(wine_instance, {
      LabelAppeal <- factor(LabelAppeal)
      AcidIndex <- factor(AcidIndex)
      STARS <- factor(STARS)
  })
  return (wine_instance)
}
summary(wine_eval)
```

```
wine_eval <- mice(wine_eval, m=1, maxit = 5, seed = 42)
wine_eval <- complete(wine_eval)
wine_eval <- as.data.frame(wine_eval)
wine_eval$AcidIndex <- log(wine_eval$AcidIndex)
wine_eval$TARGET1 <- predict(model10, newdata=wine_eval)
write.csv(wine_eval,"Evaluation_Full_Data.csv", row.names=FALSE)
```

The output is present at the below given location.

Link for output: https://github.com/Riteshlohiya/Data621-Assignment-5/blob/master/Evaluation_Full_Data.csv

Appendix

---

title: "Data621 Assignment 5"

author: "Ritesh Lohiya"

date: "July 12, 2018"

output: html_document

#Overview

In this homework assignment, you will explore, analyze and model a data set containing information on approximately 12,000 commercially available wines. The variables are mostly related to the chemical properties of the wine being sold. The response variable is the number of sample cases of wine that were purchased by wine distribution companies after sampling a wine. These cases would be used to provide tasting samples to restaurants and wine stores around the United States. The more sample cases purchased, the more likely is a wine to be sold at a high end restaurant. A large wine manufacturer is studying the data in order to predict the number of wine cases ordered based upon the wine characteristics. If the wine manufacturer can predict the number of cases, then that manufacturer will be able to adjust their wine offering to maximize sales. Your objective is to build a count regression model to predict the number of cases of wine that will be sold given certain properties of the wine. HINT: Sometimes, the fact that a variable is missing is actually predictive of the target. You can only use the variables given to you (or variables that you derive from the variables provided).

```{r}

library(readr)

library(kableExtra)

library(tidyverse)

library(knitr)

library(psych)

library(gridExtra)
```

```
library(usdm)

library(mice)

library(ggiraph)

library(cowplot)

library(reshape2)

library(corrgram)

library(caTools)

library(caret)

library(ROCR)

library(pROC)

library(reshape2)

library(Amelia)

library(qqplotr)

library(moments)

library(car)

library(MASS)

library(geoR)

library(pander)
```

#DATA EXPLORATION:

```{r}

wine_train <- read.csv("https://raw.githubusercontent.com/Riteshlohiya/Data621-Assignment-5/master/wine_training_data.csv", stringsAsFactors = FALSE)
```

```
do_factors <- function(wine_instance){
  wine_instance <- within(wine_instance, {
     LabelAppeal <- factor(LabelAppeal)
     AcidIndex <- factor(AcidIndex)
     STARS <- factor(STARS)
  })
  return (wine_instance)
}
summary(wine_train)
```

Removing the Index column:

```{r}
wine_train <- wine_train[,-c(1)]
```

There are 12795 observations and 16 variables. Each wine has 14 potential predictor variables, and 1 response variable. The response variable is "TARGET", which is the number of cases purchased.

Visual Exploration:

Let's dig into our available variables.

AcidIndex - Proprietary method of testing total acidity of wine by using a weighted average.

```{r}
with(wine_train, c(summary(AcidIndex), SD=sd(AcidIndex), Skew=skewness(AcidIndex), Kurt=kurtosis(AcidIndex)))

hist <- ggplot(wine_train, aes(AcidIndex)) + geom_histogram(fill = 'dodgerblue', binwidth = 2, color = 'darkgray' ) +
 theme_classic() + labs(title = 'Histogram of AcidIndex') + theme(plot.title = element_text(hjust = 0.5))

qq_plot <- ggplot(wine_train, aes(sample=AcidIndex)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +
  labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of AcidIndex") + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

box_plot <- ggplot(wine_train, aes(x="", AcidIndex)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +
  labs(title = 'Boxplot of AcidIndex', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()

box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), AcidIndex)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +
  labs(x='TARGET', title = 'Boxplot of AcidIndex by TARGET') + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```

Alcohol - This variable tells us about the Alcohol content.

```{r}

with(wine_train, c(summary(Alcohol), SD=sd(Alcohol), Skew=skewness(Alcohol), Kurt=kurtosis(Alcohol)))

hist <- ggplot(wine_train, aes(Alcohol)) + geom_histogram(fill = 'dodgerblue', binwidth = 2, color = 'darkgray' ) +
 theme_classic() + labs(title = 'Histogram of Alcohol') + theme(plot.title = element_text(hjust = 0.5))

qq_plot <- ggplot(wine_train, aes(sample=Alcohol)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +
  labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of Alcohol") + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

box_plot <- ggplot(wine_train, aes(x="", Alcohol)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +
  labs(title = 'Boxplot of Alcohol', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()

box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), Alcohol)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +
  labs(x='TARGET', title = 'Boxplot of Alcohol by TARGET') + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```

Chlorides - This variable tells us about the Chloride content of wine.

```{r}

with(wine_train, c(summary(Chlorides), SD=sd(Chlorides), Skew=skewness(Chlorides), Kurt=kurtosis(Chlorides)))
```

```
hist <- ggplot(wine_train, aes(Chlorides)) + geom_histogram(fill = 'dodgerblue', binwidth = .2, color = 'darkgray' ) +
 theme_classic() + labs(title = 'Histogram of Chlorides') + theme(plot.title = element_text(hjust = 0.5))

qq_plot <- ggplot(wine_train, aes(sample=Chlorides)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +
  labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of Chlorides") + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

box_plot <- ggplot(wine_train, aes(x="", Chlorides)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +
  labs(title = 'Boxplot of Chlorides', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()

box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), Chlorides)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +
  labs(x='TARGET', title = 'Boxplot of Chlorides by TARGET') + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```

CitricAcid - This variable tells us about the Citric Acid Content of wine.

```{r}
with(wine_train, c(summary(CitricAcid), SD=sd(CitricAcid), Skew=skewness(CitricAcid), Kurt=kurtosis(CitricAcid)))

hist <- ggplot(wine_train, aes(CitricAcid)) + geom_histogram(fill = 'dodgerblue', binwidth = 1, color = 'darkgray' ) +
```

```
 theme_classic() + labs(title = 'Histogram of CitricAcid') + theme(plot.title = element_text(hjust = 0.5))

qq_plot <- ggplot(wine_train, aes(sample=CitricAcid)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +
  labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of CitricAcid") + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

box_plot <- ggplot(wine_train, aes(x="", CitricAcid)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +
  labs(title = 'Boxplot of CitricAcid', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()

box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), CitricAcid)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +
  labs(x='TARGET', title = 'Boxplot of CitricAcid by TARGET') + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```

Density - This variable tells us about the Density of wine.

```{r}
with(wine_train, c(summary(Density), SD=sd(Density), Skew=skewness(Density), Kurt=kurtosis(Density)))

hist <- ggplot(wine_train, aes(Density)) + geom_histogram(fill = 'dodgerblue', binwidth = .05, color = 'darkgray' ) +
 theme_classic() + labs(title = 'Histogram of Density') + theme(plot.title = element_text(hjust = 0.5))
qq_plot <- ggplot(wine_train, aes(sample=Density)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +
```

```
 labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of Density") + theme_classic() +
 theme(plot.title = element_text(hjust = 0.5))

box_plot <- ggplot(wine_train, aes(x="", Density)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +
 labs(title = 'Boxplot of Density', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()

box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), Density)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +
 labs(x='TARGET', title = 'Boxplot of Density by TARGET') + theme_classic() +
 theme(plot.title = element_text(hjust = 0.5))

grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```

FixedAcidity - This variable tells us about the FixedAcidity of wine.

```{r}
with(wine_train, c(summary(FixedAcidity), SD=sd(FixedAcidity), Skew=skewness(FixedAcidity), Kurt=kurtosis(FixedAcidity)))

hist <- ggplot(wine_train, aes(FixedAcidity)) + geom_histogram(fill = 'dodgerblue', binwidth = 4, color = 'darkgray' ) +
 theme_classic() + labs(title = 'Histogram of FixedAcidity') + theme(plot.title = element_text(hjust = 0.5))

qq_plot <- ggplot(wine_train, aes(sample=FixedAcidity)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +
 labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of FixedAcidity") + theme_classic() +
 theme(plot.title = element_text(hjust = 0.5))
```

box_plot <- ggplot(wine_train, aes(x="", FixedAcidity)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +

 labs(title = 'Boxplot of FixedAcidity', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()

box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), FixedAcidity)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +

 labs(x='TARGET', title = 'Boxplot of FixedAcidity by TARGET') + theme_classic() +

 theme(plot.title = element_text(hjust = 0.5))

grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)

```

FreeSulfurDioxide - This variable tells us about the Sulfur Dioxide content of wine.

```{r}

with(wine_train, c(summary(FreeSulfurDioxide), SD=sd(FreeSulfurDioxide), Skew=skewness(FreeSulfurDioxide),
Kurt=kurtosis(FreeSulfurDioxide)))

hist <- ggplot(wine_train, aes(FreeSulfurDioxide)) + geom_histogram(fill = 'dodgerblue', binwidth = 50, color = 'darkgray' ) +

 theme_classic() + labs(title = 'Histogram of FreeSulfurDioxide') + theme(plot.title = element_text(hjust = 0.5))

qq_plot <- ggplot(wine_train, aes(sample=FreeSulfurDioxide)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +

 labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of FreeSulfurDioxide") + theme_classic() +

 theme(plot.title = element_text(hjust = 0.5))

box_plot <- ggplot(wine_train, aes(x="", FreeSulfurDioxide)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +

```
 labs(title = 'Boxplot of FreeSulfurDioxide', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()

box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), FreeSulfurDioxide)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +
 labs(x='TARGET', title = 'Boxplot of FreeSulfurDioxide by TARGET') + theme_classic() +
 theme(plot.title = element_text(hjust = 0.5))

grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```

LabelAppeal - Marketing Score indicating the appeal of label design for consumers. High numbers suggest customers like the label design. Negative numbers suggest customers don't like the design. Many consumers purchase based on the visual appeal of the wine label design. Higher numbers suggest better sales.

```{r}
with(wine_train, c(summary(LabelAppeal), SD=sd(LabelAppeal), Skew=skewness(LabelAppeal), Kurt=kurtosis(LabelAppeal)))


hist <- ggplot(wine_train, aes(LabelAppeal)) + geom_histogram(fill = 'dodgerblue', binwidth = 1, color = 'darkgray' ) +
 theme_classic() + labs(title = 'Histogram of LabelAppeal') + theme(plot.title = element_text(hjust = 0.5))


qq_plot <- ggplot(wine_train, aes(sample=LabelAppeal)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +
 labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of LabelAppeal") + theme_classic() +
 theme(plot.title = element_text(hjust = 0.5))


box_plot <- ggplot(wine_train, aes(x="", LabelAppeal)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +
 labs(title = 'Boxplot of LabelAppeal', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()
```

```
box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), LabelAppeal)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +
  labs(x='TARGET', title = 'Boxplot of LabelAppeal by TARGET') + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```

ResidualSugar - This variable tells us about the ResidualSugar of wine.

```{r}
with(wine_train, c(summary(ResidualSugar), SD=sd(ResidualSugar), Skew=skewness(ResidualSugar), Kurt=kurtosis(ResidualSugar)))

hist <- ggplot(wine_train, aes(ResidualSugar)) + geom_histogram(fill = 'dodgerblue', binwidth = 20, color = 'darkgray' ) +
 theme_classic() + labs(title = 'Histogram of ResidualSugar') + theme(plot.title = element_text(hjust = 0.5))

qq_plot <- ggplot(wine_train, aes(sample=ResidualSugar)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +
  labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of ResidualSugar") + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

box_plot <- ggplot(wine_train, aes(x="", ResidualSugar)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +
  labs(title = 'Boxplot of ResidualSugar', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()

box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), ResidualSugar)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +
  labs(x='TARGET', title = 'Boxplot of ResidualSugar by TARGET') + theme_classic() +
```

```
 theme(plot.title = element_text(hjust = 0.5))

grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```

STARS - Wine rating by a team of experts. 4 Stars = Excellent, 1 Star = Poor. A high number of stars suggests high sales.

```{r}
with(wine_train, c(summary(STARS), SD=sd(STARS), Skew=skewness(STARS), Kurt=kurtosis(STARS)))

hist <- ggplot(wine_train, aes(STARS)) + geom_histogram(fill = 'dodgerblue', binwidth = 1, color = 'darkgray' ) +
 theme_classic() + labs(title = 'Histogram of STARS') + theme(plot.title = element_text(hjust = 0.5))

qq_plot <- ggplot(wine_train, aes(sample=STARS)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +
  labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of STARS") + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

box_plot <- ggplot(wine_train, aes(x="", STARS)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +
  labs(title = 'Boxplot of STARS', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()

box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), STARS)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +
  labs(x='TARGET', title = 'Boxplot of STARS by TARGET') + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
```

```
grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```

Sulphates - This variable tells us about the Sulphates content of wine.

```{r}
with(wine_train, c(summary(Sulphates), SD=sd(Sulphates), Skew=skewness(Sulphates), Kurt=kurtosis(Sulphates)))

hist <- ggplot(wine_train, aes(Sulphates)) + geom_histogram(fill = 'dodgerblue', binwidth = .5, color = 'darkgray' ) +
 theme_classic() + labs(title = 'Histogram of Sulphates') + theme(plot.title = element_text(hjust = 0.5))

qq_plot <- ggplot(wine_train, aes(sample=Sulphates)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +
  labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of Sulphates") + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

box_plot <- ggplot(wine_train, aes(x="", Sulphates)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +
  labs(title = 'Boxplot of Sulphates', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()

box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), Sulphates)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +
  labs(x='TARGET', title = 'Boxplot of Sulphates by TARGET') + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```

TotalSulfurDioxide - This variable tells us about the Total Sulfur Dioxide of Wine.

```{r}
with(wine_train, c(summary(TotalSulfurDioxide), SD=sd(TotalSulfurDioxide), Skew=skewness(TotalSulfurDioxide), Kurt=kurtosis(TotalSulfurDioxide)))

hist <- ggplot(wine_train, aes(TotalSulfurDioxide)) + geom_histogram(fill = 'dodgerblue', binwidth = 200, color = 'darkgray' ) +
 theme_classic() + labs(title = 'Histogram of TotalSulfurDioxide') + theme(plot.title = element_text(hjust = 0.5))

qq_plot <- ggplot(wine_train, aes(sample=TotalSulfurDioxide)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +
  labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of TotalSulfurDioxide") + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

box_plot <- ggplot(wine_train, aes(x="", TotalSulfurDioxide)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +
  labs(title = 'Boxplot of TotalSulfurDioxide', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()

box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), TotalSulfurDioxide)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +
  labs(x='TARGET', title = 'Boxplot of TotalSulfurDioxide by TARGET') + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```

VolatileAcidity - This variable tells us about the VolatileAcidity content of Wine.

```{r}

with(wine_train, c(summary(VolatileAcidity), SD=sd(VolatileAcidity), Skew=skewness(VolatileAcidity), Kurt=kurtosis(VolatileAcidity)))

hist <- ggplot(wine_train, aes(VolatileAcidity)) + geom_histogram(fill = 'dodgerblue', binwidth = .5, color = 'darkgray' ) +
 theme_classic() + labs(title = 'Histogram of VolatileAcidity') + theme(plot.title = element_text(hjust = 0.5))

qq_plot <- ggplot(wine_train, aes(sample=VolatileAcidity)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +
  labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of VolatileAcidity") + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

box_plot <- ggplot(wine_train, aes(x="", VolatileAcidity)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +
  labs(title = 'Boxplot of VolatileAcidity', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()

box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), VolatileAcidity)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +
  labs(x='TARGET', title = 'Boxplot of VolatileAcidity by TARGET') + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```

pH - This variable tells us about the pH of Wine.

```{r}
```

```
with(wine_train, c(summary(pH), SD=sd(pH), Skew=skewness(pH), Kurt=kurtosis(pH)))
```

```
hist <- ggplot(wine_train, aes(pH)) + geom_histogram(fill = 'dodgerblue', binwidth = .5, color = 'darkgray' ) +
 theme_classic() + labs(title = 'Histogram of pH') + theme(plot.title = element_text(hjust = 0.5))
```

```
qq_plot <- ggplot(wine_train, aes(sample=pH)) + stat_qq_point(color='dodgerblue') + stat_qq_line(color='darkgray') +
  labs(x="Thoretical Quantiles", y="Sample Quantiles", title = "QQ Plot of pH") + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
```

```
box_plot <- ggplot(wine_train, aes(x="", pH)) + geom_boxplot(fill='dodgerblue', color='darkgray')+ theme_classic() +
  labs(title = 'Boxplot of pH', x="") + theme(plot.title = element_text(hjust = 0.5)) + coord_flip()
```

```
box_TARGET <- ggplot(wine_train, aes(x=factor(TARGET), pH)) + geom_boxplot(fill='dodgerblue', color='darkgrey') +
  labs(x='TARGET', title = 'Boxplot of pH by TARGET') + theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
```

```
grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```
```

Now lets see NA's for all the variables other than STARS.STARS has NAs that is more than 10%.

```{r}
Non_NAs <- sapply(wine_train, function(y) sum(length(which(!is.na(y)))))
```
```

NAs <- sapply(wine_train, function(y) sum(length(which(is.na(y)))))

NA_Percent <- NAs / (NAs + Non_NAs)

NA_SUMMARY <- data.frame(Non_NAs,NAs,NA_Percent)

missmap(wine_train, main = "Missing Values")

kable(NA_SUMMARY)

```

Finding correlations: The correlation plot below shows how variables in the dataset are related to each other. Looking at the plot, we dont see much correlations.

```{r}

names(wine_train)

cor(drop_na(wine_train))

```

```{r}

pairs.panels(wine_train[1:15])

```

Now we will see the TARGET Variable.

TARGET - Number of Cases Purchased
```

```{r}

options(width=100)

round(with(wine_train, c(summary(TARGET), StdD=sd(TARGET), Skew=skewness(TARGET), Kurt=kurtosis(TARGET))),2)

```

#DATA PREPARATION:

Lets first split the data into training and test.

```{r}

set.seed(999)

sampl = sample.split(wine_train$TARGET, SplitRatio = .80)

wine_train1 <- subset(wine_train, sampl == TRUE)

wine_test1 <- subset(wine_train, sampl == FALSE)

```

We will now use the mice package to impute missing values.

```{r}

wine_train2 <- mice(wine_train1, m=1, maxit = 5, seed = 42)

wine_train2 <- complete(wine_train2)

wine_train2 <- as.data.frame(wine_train2)
```

```
wine_test2 <- test <- mice(wine_test1, m=1, maxit = 5, seed = 42)

wine_test2 <- complete(wine_test2)

wine_test2 <- as.data.frame(wine_test2)
```

There is very low correlation between AcidIndex and TARGET, lets do log transformation on AcidIndex.

```{r}
wine_train2$AcidIndex <- log(wine_train2$AcidIndex)

wine_test2$AcidIndex <- log(wine_test2$AcidIndex)
```

#BUILD MODELS:

1. Poisson model without imputations.

```{r}
model1 = glm(TARGET ~  ., data=wine_train1, family=poisson)

summary(model1)

grid.arrange(hist, qq_plot, box_plot, box_TARGET, ncol=2)
```

2. Poisson model without imputations and only significant variables.

```{r}

model2 = glm(TARGET ~ .-FixedAcidity-CitricAcid-ResidualSugar-Chlorides-FreeSulfurDioxide-TotalSulfurDioxide-Density-pH-Sulphates-Alcohol, data=wine_train1, family=poisson)

summary(model2)

plot(model2)

```

3. Poisson model with Imputation.

```{r}

model3 = glm(TARGET ~ ., data=wine_train2, family=poisson)

summary(model3)

plot(model3)

```

4. Poisson model with imputations and only significant variables.

```{r}

model4 = glm(TARGET ~ .-FixedAcidity-CitricAcid-ResidualSugar-Density-Alcohol, data=wine_train2, family=poisson)

summary(model4)

plot(model4)

```

5. Negative Binomial without imputations:

```{r}

model5 <- glm.nb(TARGET ~ ., data = wine_train1)

summary(model5)

plot(model5)

```

6. Negative Binomial without imputations and only significant variables:

```{r}

model6 <- glm.nb(TARGET ~ .-FixedAcidity-CitricAcid-ResidualSugar-Chlorides-FreeSulfurDioxide-TotalSulfurDioxide-Density-pH-Sulphates-Alcohol, data = wine_train1)

summary(model6)

plot(model6)

```

7. Negative Binomial with imputations:

```{r}

model7 <- glm.nb(TARGET ~ ., data = wine_train2)

summary(model7)

plot(model7)

```

8. Negative Binomial with imputations and only significant variables:

```{r}

model8 <- glm.nb(TARGET ~ .-FixedAcidity-CitricAcid-ResidualSugar-Density-Alcohol, data = wine_train2)

summary(model8)

plot(model8)

```

9. Linear Model with imputations.

```{r}

model9 <- lm(TARGET ~ ., data = wine_train2)

summary(model9)

plot(model9)

```

10. Linear Model with imputations and only significant variables.

```{r}

model10 <- lm(TARGET ~ .-FixedAcidity-CitricAcid-ResidualSugar, data = wine_train2)

summary(model10)

plot(model10)

```

Now lets see the output of the Models using test data:

We will use the squared loss to validate the model.

```{r}
modelValidation <- function(mod, test){
  preds = predict(mod, test)
  diffMat = as.numeric(preds) - as.numeric(test$TARGET)
  diffMat = diffMat^2
  loss <- mean(diffMat)
  return(loss)
}
```

Poisson model with imputations.

```{r}
modelValidation(model3, wine_test2)
```

Poisson model with imputations and only significant variables.

```{r}
modelValidation(model4, wine_test2)
```

Negative Binomial with imputations:.

```{r}
modelValidation(model7, wine_test2)
```

Negative Binomial with imputations and only significant variables.

```{r}
modelValidation(model8, wine_test2)
```

Linear Model with imputations.

```{r}
modelValidation(model9, wine_test2)
```

Linear Model with imputations and only significant variables.

```{r}
modelValidation(model10, wine_test2)
```

#MODEL SELECTION:

From the above models, i would like to go with Model10 - Linear Model with imputations and only significant variables as it uses less variables and is parsimonious. Also the R2 looks fine. The squared loss is also fine.

#Prediction:

We will use the same method to impute and use log transformation for AcidIndex.

```{r}
wine_eval <- read.csv("https://raw.githubusercontent.com/Riteshlohiya/Data621-Assignment-5/master/wine_evaluation_data.csv", stringsAsFactors = FALSE)

do_factors <- function(wine_instance){
  wine_instance <- within(wine_instance, {
    LabelAppeal <- factor(LabelAppeal)
    AcidIndex <- factor(AcidIndex)
    STARS <- factor(STARS)
  })
  return (wine_instance)
}
summary(wine_eval)
```

```{r}

wine_eval <- mice(wine_eval, m=1, maxit = 5, seed = 42)

wine_eval <- complete(wine_eval)

wine_eval <- as.data.frame(wine_eval)

```



```{r}

wine_eval$AcidIndex <- log(wine_eval$AcidIndex)

```



```{r}

wine_eval$TARGET1 <- predict(model10, newdata=wine_eval)

write.csv(wine_eval,"Evaluation_Full_Data.csv", row.names=FALSE)

```