DATA621_Home_Work_5

Dilip Ganesan

7/10/2018

Home Work Assignment 5.

1.DATA EXPLORATION.(Exploratory Data Analysis EDA)

As first step in our EDA, let us load the train data and do a summary statistics on the loaded dataset.

```
# Let us load the train.csv data
train = read.csv('wine-training-data.csv')
test = read.csv('wine-evaluation-data.csv')

train = within(train, rm('INDEX'))
#test = within(test, rm('INDEX'))

summary(train)
```

```
##      TARGET       FixedAcidity      VolatileAcidity     CitricAcid
##  Min.   :0.000   Min.   :-18.100   Min.   :-2.7900   Min.   :-3.2400
##  1st Qu.:2.000   1st Qu.:  5.200   1st Qu.: 0.1300   1st Qu.: 0.0300
##  Median :3.000   Median :  6.900   Median : 0.2800   Median : 0.3100
##  Mean   :3.029   Mean   :  7.076   Mean   : 0.3241   Mean   : 0.3084
##  3rd Qu.:4.000   3rd Qu.:  9.500   3rd Qu.: 0.6400   3rd Qu.: 0.5800
##  Max.   :8.000   Max.   : 34.400   Max.   : 3.6800   Max.   : 3.8600
##
##  ResidualSugar        Chlorides       FreeSulfurDioxide TotalSulfurDioxide
##  Min.   :-127.800   Min.   :-1.1710   Min.   :-555.00   Min.   :-823.0
##  1st Qu.:  -2.000   1st Qu.:-0.0310   1st Qu.:   0.00   1st Qu.:  27.0
##  Median :   3.900   Median : 0.0460   Median :  30.00   Median : 123.0
##  Mean   :   5.419   Mean   : 0.0548   Mean   :  30.85   Mean   : 120.7
##  3rd Qu.:  15.900   3rd Qu.: 0.1530   3rd Qu.:  70.00   3rd Qu.: 208.0
##  Max.   : 141.150   Max.   : 1.3510   Max.   : 623.00   Max.   :1057.0
##  NA's   :616        NA's   :638       NA's   :647       NA's   :682
##     Density             pH          Sulphates          Alcohol
##  Min.   :0.8881   Min.   :0.480   Min.   :-3.1300   Min.   :-4.70
##  1st Qu.:0.9877   1st Qu.:2.960   1st Qu.: 0.2800   1st Qu.: 9.00
##  Median :0.9945   Median :3.200   Median : 0.5000   Median :10.40
##  Mean   :0.9942   Mean   :3.208   Mean   : 0.5271   Mean   :10.49
##  3rd Qu.:1.0005   3rd Qu.:3.470   3rd Qu.: 0.8600   3rd Qu.:12.40
##  Max.   :1.0992   Max.   :6.130   Max.   : 4.2400   Max.   :26.50
##                   NA's   :395     NA's   :1210      NA's   :653
##   LabelAppeal          AcidIndex          STARS
##  Min.   :-2.000000   Min.   : 4.000   Min.   :1.000
##  1st Qu.:-1.000000   1st Qu.: 7.000   1st Qu.:1.000
##  Median : 0.000000   Median : 8.000   Median :2.000
##  Mean   :-0.009066   Mean   : 7.773   Mean   :2.042
##  3rd Qu.: 1.000000   3rd Qu.: 8.000   3rd Qu.:3.000
##  Max.   : 2.000000   Max.   :17.000   Max.   :4.000
##                                       NA's   :3359
```

Examination of Data Set.

1. There are 12795 observations and 15 variables(excluding the INDEX) in the train data set. Out of 15 variables 0 are discrete, and all 15 are continous variables.
2. TARGET is the predictor variable in the data set.

Statistical Summary of Data Set:

```
summary = describe(train, quant = c(.25,.75))
knitr::kable(summary)
```

	vars	n	mean	sd	median	trimmed	mad	min	
TARGET	1	12795	3.0290739	1.9263682	3.00000	3.0538244	1.4826000	0.00000	
FixedAcidity	2	12795	7.0757171	6.3176435	6.90000	7.0736739	3.2617200	-18.10000	
VolatileAcidity	3	12795	0.3241039	0.7840142	0.28000	0.3243890	0.4299540	-2.79000	
CitricAcid	4	12795	0.3084127	0.8620798	0.31000	0.3102520	0.4151280	-3.24000	
ResidualSugar	5	12179	5.4187331	33.7493790	3.90000	5.5800410	15.7155600	-127.80000	1
Chlorides	6	12157	0.0548225	0.3184673	0.04600	0.0540159	0.1349166	-1.17100	
FreeSulfurDioxide	7	12148	30.8455713	148.7145577	30.00000	30.9334877	56.3388000	-555.00000	6
TotalSulfurDioxide	8	12113	120.7142326	231.9132105	123.00000	120.8895367	134.9166000	-823.00000	10
Density	9	12795	0.9942027	0.0265376	0.99449	0.9942130	0.0093552	0.88809	
pH	10	12400	3.2076282	0.6796871	3.20000	3.2055706	0.3854760	0.48000	
Sulphates	11	11585	0.5271118	0.9321293	0.50000	0.5271453	0.4447800	-3.13000	
Alcohol	12	12142	10.4892363	3.7278190	10.40000	10.5018255	2.3721600	-4.70000	
LabelAppeal	13	12795	-0.0090660	0.8910892	0.00000	-0.0099639	1.4826000	-2.00000	
AcidIndex	14	12795	7.7727237	1.3239264	8.00000	7.6431572	1.4826000	4.00000	
STARS	15	9436	2.0417550	0.9025400	2.00000	1.9711258	1.4826000	1.00000	

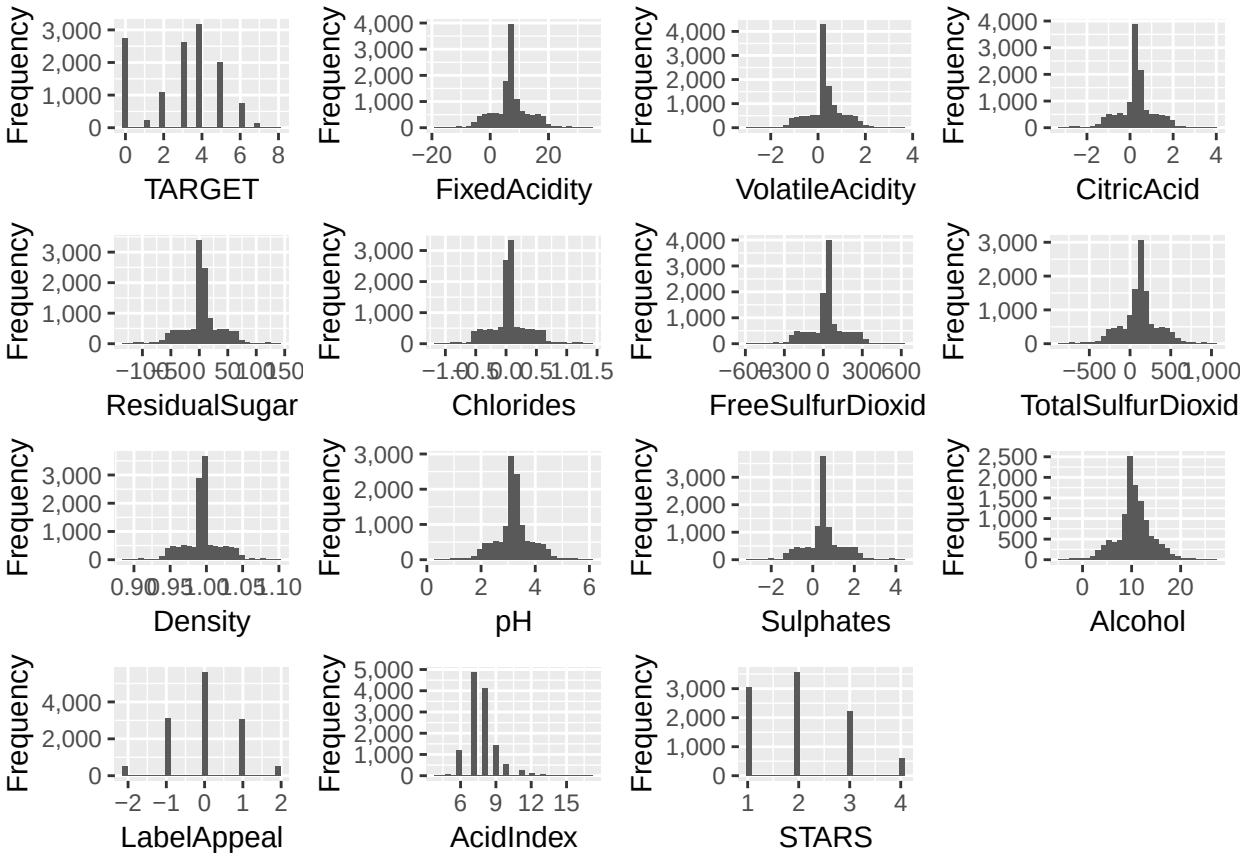
1. There are total of 8200 missing NAs and the max of that are in STARS, we can set the missing STARS NAs to 0.
2. ACIDINDEX, LABELAPPEAL, STARS and TARGET are all discrete variables. So the count regression is the approach we can follow.
3. Many variables has negative values. Need to do some data transformation.

Visual Exploration of Data set:

Histogram

```
ggtrain = train

plot_histogram(ggtrain)
```

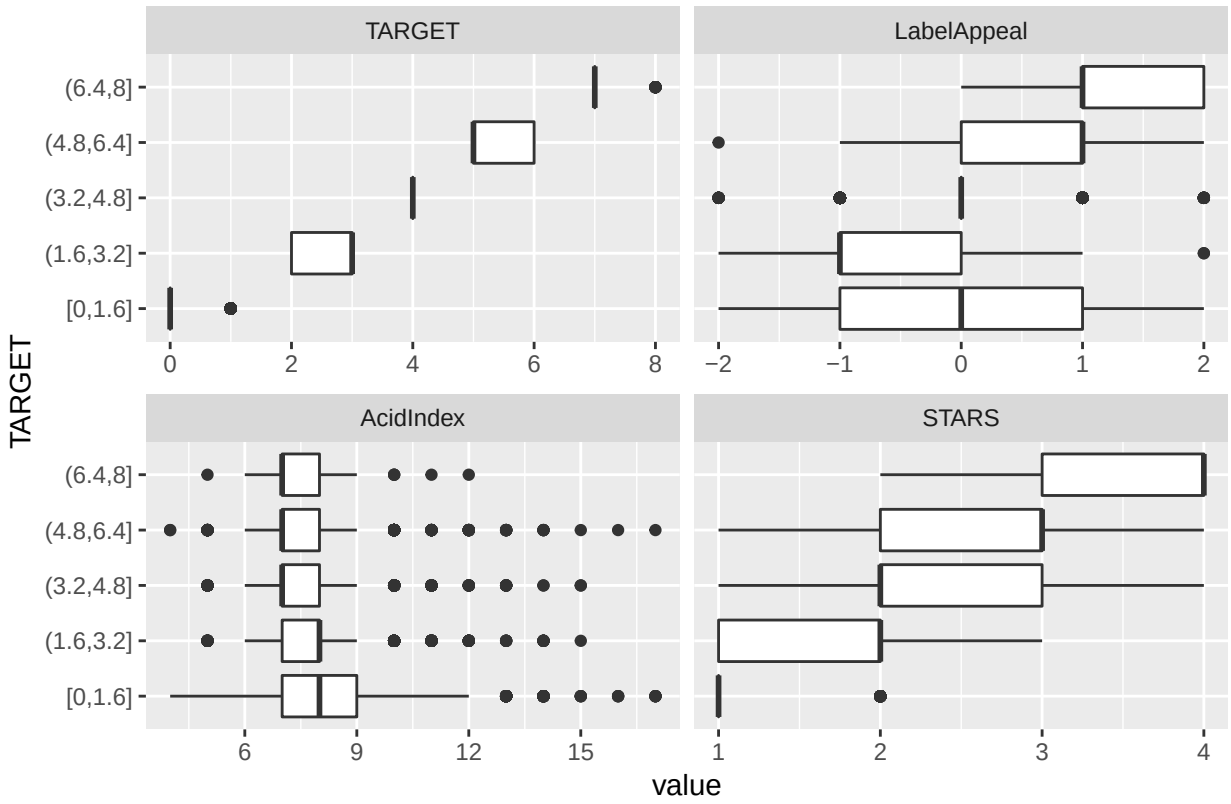


1. From the histogram we can see that variable ACIDINDEX is positively skewed.
2. Most of the continous variables are platykurtic having smaller tails and higher peaks.

BoxPlot

We will do a box plot of the discrete variable and predictor variables.

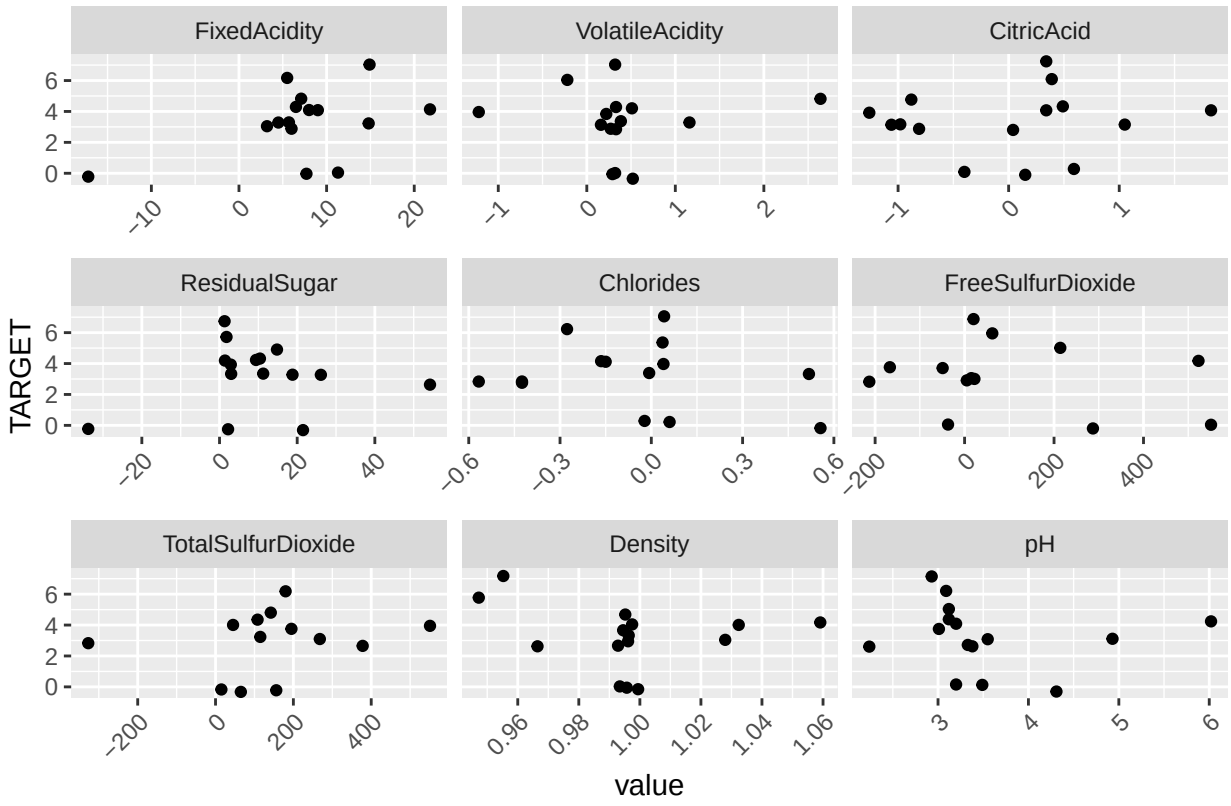
```
boxtrain = subset(train, select = c('TARGET','LabelAppeal','AcidIndex','STARS'))
plot_boxplot(boxtrain,"TARGET")
```



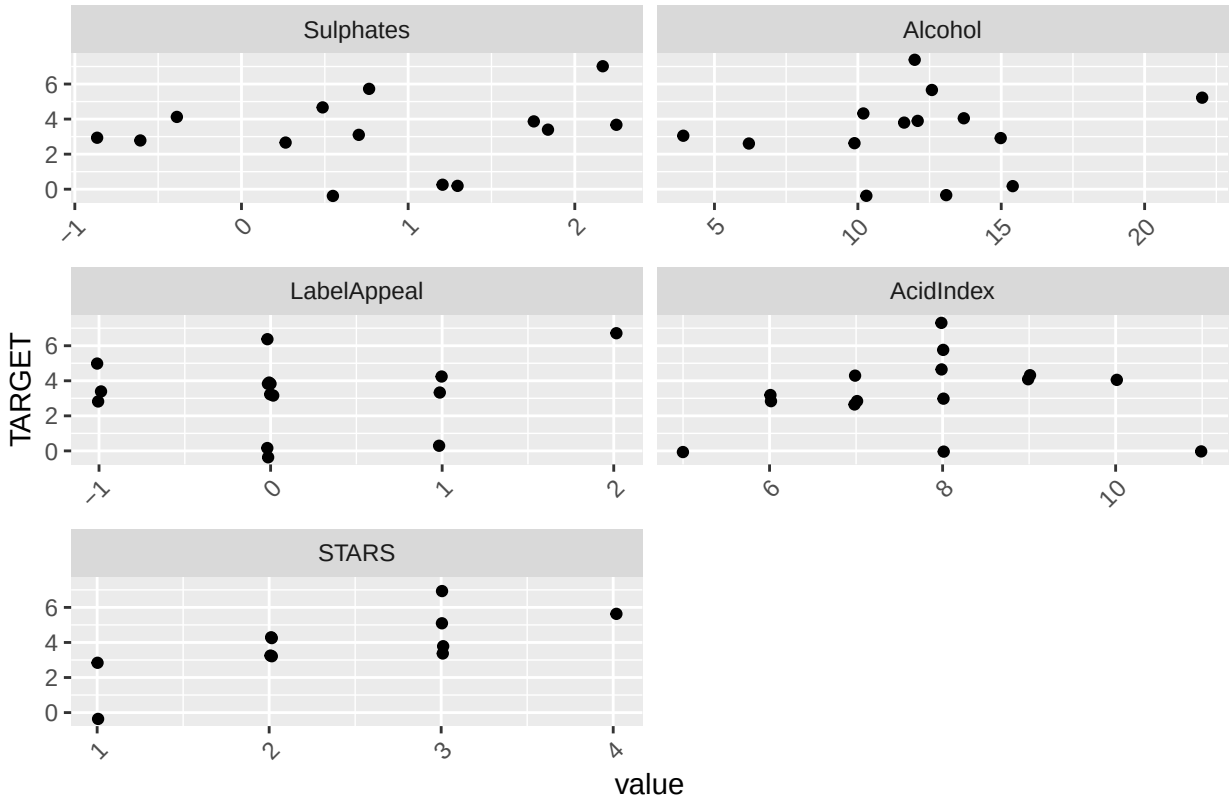
1. From the box plot we can see higher values of STARS and label Appeal the more cases are bought.

ScatterPlot

```
plot_scatterplot(train[1:15,], "TARGET", position = "jitter")
```



Continuous Features: Page 1



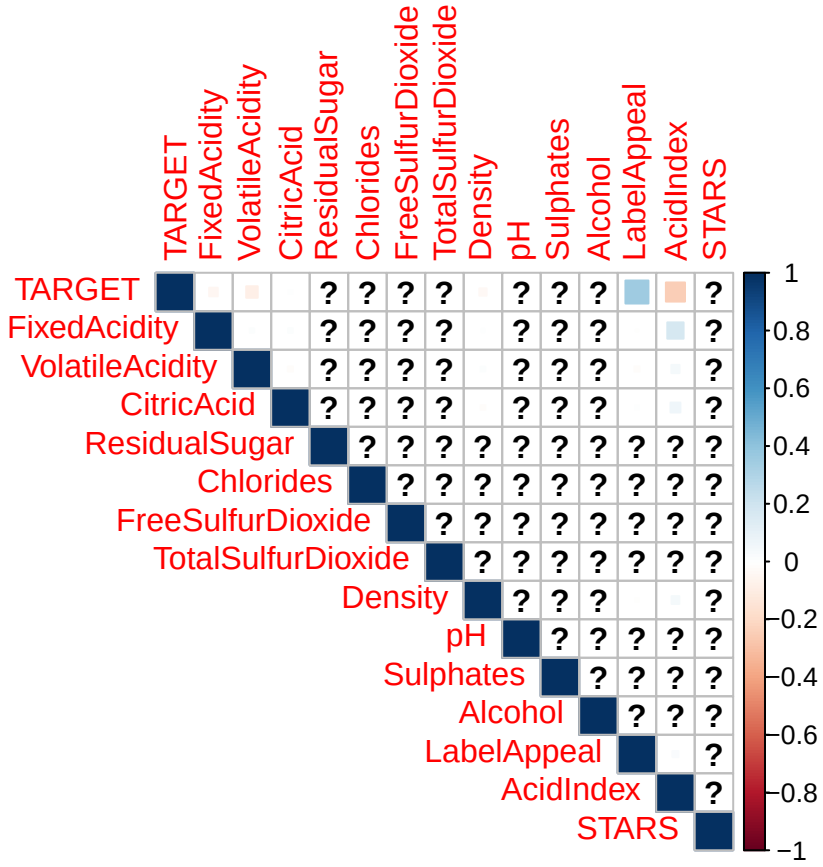
Continuous Features: Page 2

From the scatterplot, we do not see any pronounced positive or negative relationships. We will do a Correlation

next to see in detail.

MultiCollinearity between predictor variables and also with response variables

```
cordata = cor(train)
corrplot(cordata, method = "square", type = "upper")
```

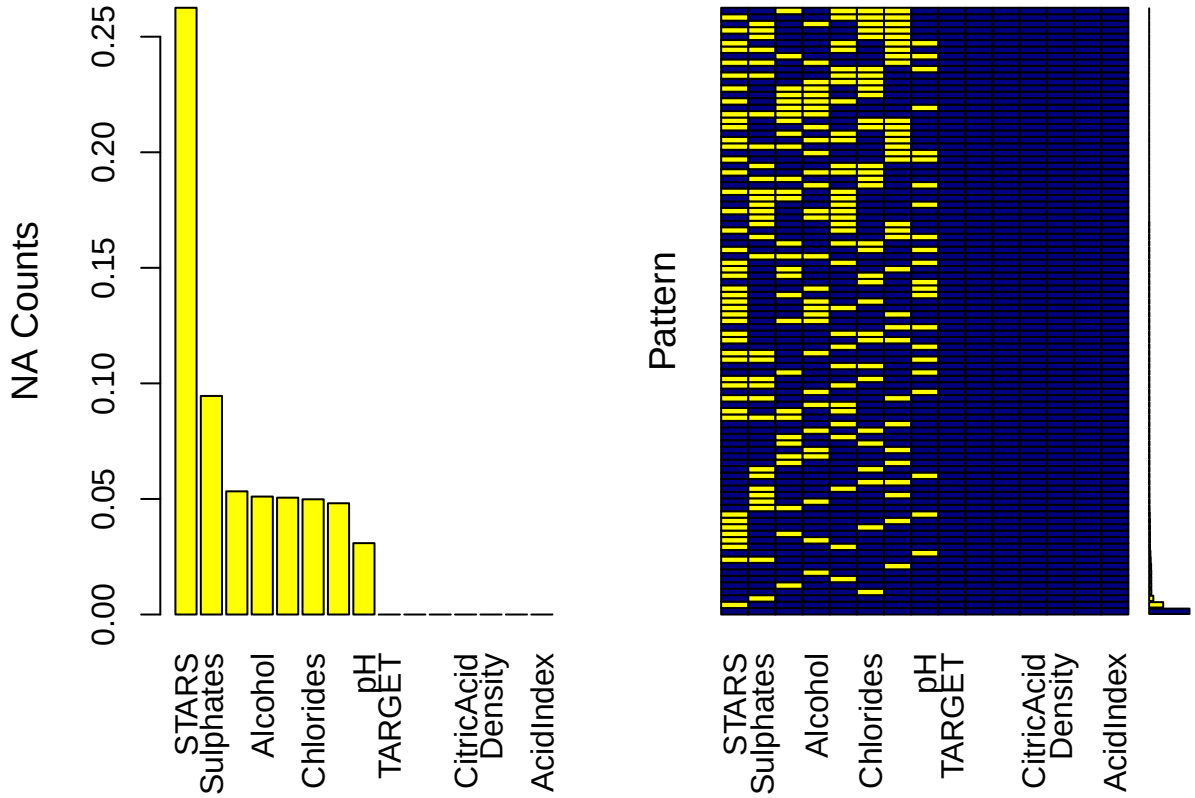


From the corrplot we can see that LabelAppeal and TARGET has a litte positive correlation. AcidIndex and TARGET has a litte negative correlation.

With respect to response and predictor variables, We do not see much of a bigger correlation.

Missing Value

```
VIM::aggr(train, col=c('navyblue','yellow'),
                  numbers=TRUE, sortVars=TRUE,
                   labels=names(train),
                   ylab=c("NA Counts","Pattern"))
```




```
##
##  Variables sorted by number of missings:
##            Variable      Count
##               STARS 0.26252442
##           Sulphates 0.09456819
##  TotalSulfurDioxide 0.05330207
##             Alcohol 0.05103556
##   FreeSulfurDioxide 0.05056663
##           Chlorides 0.04986323
##       ResidualSugar 0.04814381
##                  pH 0.03087143
##              TARGET 0.00000000
##        FixedAcidity 0.00000000
##     VolatileAcidity 0.00000000
##          CitricAcid 0.00000000
##             Density 0.00000000
##         LabelAppeal 0.00000000
##           AcidIndex 0.00000000
```

2.DATA PREPARATION

From our visual exploration we have identified 8 variables are having missing values.

1. We will set Zero for NA values for variable STARS.
2. 10 out of 15 variables has negative values. This shows the data set might not be accurate. We will take an absolute value for these variables.

For rest of the variable which are missing values we can try Random Forrest Algorithm for imputation. In the

last excercise we tried executing imputing these random missing values using MICE package. But running the package lead to crashing of R server multiple times. So for that project dropped using mice.

Now for this project we can try using missForest and see the performance.

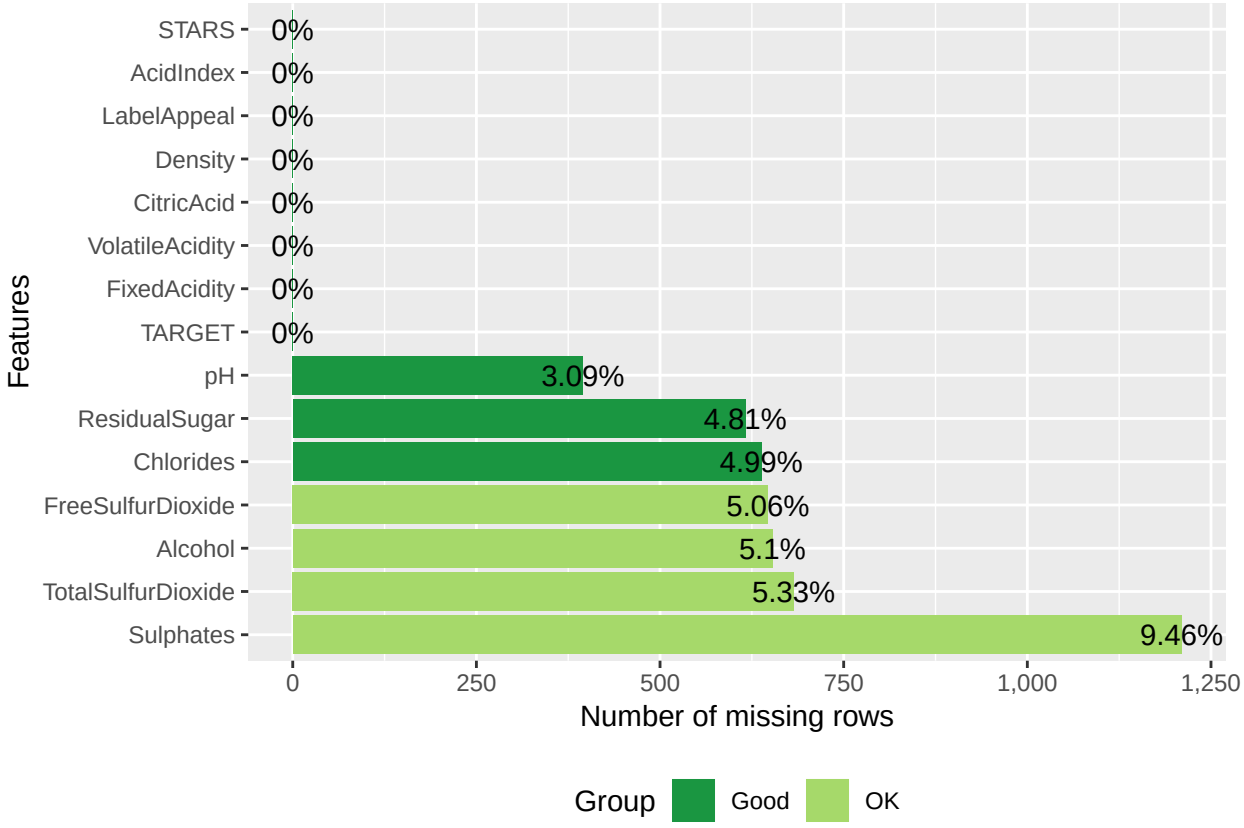
```
train$STARS[is.na(train$STARS)] = 0
test$STARS[is.na(test$STARS)] = 0

train =
  train %>%
  mutate(
    FixedAcidity = abs(FixedAcidity),
    VolatileAcidity = abs(VolatileAcidity),
    CitricAcid = abs(CitricAcid),
    ResidualSugar = abs(ResidualSugar),
    Chlorides = abs(Chlorides),
    FreeSulfurDioxide = abs(FreeSulfurDioxide),
    TotalSulfurDioxide = abs(TotalSulfurDioxide),
    Sulphates = abs(Sulphates),
    Alcohol = abs(Alcohol))


test =
  test %>%
  mutate(
    FixedAcidity = abs(FixedAcidity),
    VolatileAcidity = abs(VolatileAcidity),
    CitricAcid = abs(CitricAcid),
    ResidualSugar = abs(ResidualSugar),
    Chlorides = abs(Chlorides),
    FreeSulfurDioxide = abs(FreeSulfurDioxide),
    TotalSulfurDioxide = abs(TotalSulfurDioxide),
    Sulphates = abs(Sulphates),
    Alcohol = abs(Alcohol))
```

We will plot and see the missing elements. This is after filling the missed values.

```
plot_missing(train)
```



If we see below, we are seeing following variables has missing values.

Sulphates TotalSulfurDioxide Alcohol FreeSulfurDioxide Chlorides ResidualSugar pH

For the above we will use the missForest Package and do the imputation.

```
imputed_train = missForest(train, variablewise = T)
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
##   missForest iteration 4 in progress...done!
##   missForest iteration 5 in progress...done!
```

```
#imputed_train$ximp
train_imputed = imputed_train$ximp

#imputed_train$OOBerror

imputed_eval = missForest(test, variablewise = T)
```

```
##   removed variable(s) 2 due to the missingness of all entries
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
```

3. BUILDING MODELS

Classification:

As approach we are going to build the following models.

For the following problem we are going to use the Linear Model.

Base Model and Transformed Variables

1. The first model will be Base Model. It will contain all transformed imputed data. It contains all the 15 variables.

```
regbaseplustransform = lm(TARGET ~ .  , data = train_imputed)

summary(regbaseplustransform)
```

```
##
## Call:
## lm(formula = TARGET ~ ., data = train_imputed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4944 -0.9565  0.0600  0.9127  6.0318
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         4.048e+00  4.504e-01   8.986  < 2e-16 ***
## FixedAcidity        5.294e-06  2.386e-03   0.002 0.998230
## VolatileAcidity    -1.150e-01  2.115e-02  -5.435 5.56e-08 ***
## CitricAcid          3.457e-02  1.937e-02   1.784 0.074375 .
## ResidualSugar      -5.898e-05  4.815e-04  -0.122 0.902508
## Chlorides          -9.692e-02  5.135e-02  -1.887 0.059141 .
## FreeSulfurDioxide   1.967e-04  1.113e-04   1.768 0.077154 .
## TotalSulfurDioxide  2.732e-04  7.400e-05   3.691 0.000224 ***
## Density            -8.399e-01  4.425e-01  -1.898 0.057681 .
## pH                 -3.557e-02  1.756e-02  -2.026 0.042792 *
## Sulphates          -4.523e-02  1.879e-02  -2.407 0.016087 *
## Alcohol             1.221e-02  3.324e-03   3.673 0.000241 ***
## LabelAppeal         4.316e-01  1.369e-02  31.533  < 2e-16 ***
## AcidIndex          -2.103e-01  9.208e-03 -22.836  < 2e-16 ***
## STARS               9.795e-01  1.046e-02  93.657  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.326 on 12780 degrees of freedom
## Multiple R-squared:  0.5266, Adjusted R-squared:  0.5261
## F-statistic:  1016 on 14 and 12780 DF,  p-value: < 2.2e-16
```

Observations

From the above model, we see 5 out of the 15 variables has (stat-sig) p-values at a significance level greater than 0.05. These variable can be dropped in our next model to see how our model performs. The below are the variables which can be dropped in our next model.

FixedAcidity ResidualSugar FreeSulfurDioxide CitricAcid Density

The p-values of LableAppeal, AcidIndex and STARS are very less than 0.05. They have significance impact on the model.

Base Model Transformation pls Backward Elimination

2. For this model, we are going to use the Base Model(Transformation) and we are going to remove the variables which has higher p-value(>0.05).

The variables discussed above have been removed from base_model variables.

```
regbasebackward=
lm(formula = TARGET ~ VolatileAcidity + Chlorides + TotalSulfurDioxide +
    pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS,
   data = train_imputed)

summary(regbasebackward)
```

```
##
## Call:
## lm(formula = TARGET ~ VolatileAcidity + Chlorides + TotalSulfurDioxide +
##     pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS,
##     data = train_imputed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5158 -0.9617  0.0608  0.9083  5.9855
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3.260e+00  1.071e-01  30.442  < 2e-16 ***
## VolatileAcidity    -1.152e-01  2.116e-02  -5.445 5.29e-08 ***
## Chlorides          -9.908e-02  5.136e-02  -1.929 0.053720 .
## TotalSulfurDioxide  2.742e-04  7.398e-05   3.706 0.000212 ***
## pH                 -3.591e-02  1.756e-02  -2.045 0.040901 *
## Sulphates          -4.502e-02  1.879e-02  -2.396 0.016575 *
## Alcohol             1.215e-02  3.324e-03   3.654 0.000259 ***
## LabelAppeal         4.323e-01  1.369e-02  31.583  < 2e-16 ***
## AcidIndex          -2.106e-01  9.053e-03 -23.265  < 2e-16 ***
## STARS               9.801e-01  1.046e-02  93.727  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.326 on 12785 degrees of freedom
## Multiple R-squared:  0.5263, Adjusted R-squared:  0.5259
## F-statistic:  1578 on 9 and 12785 DF,  p-value: < 2.2e-16
```

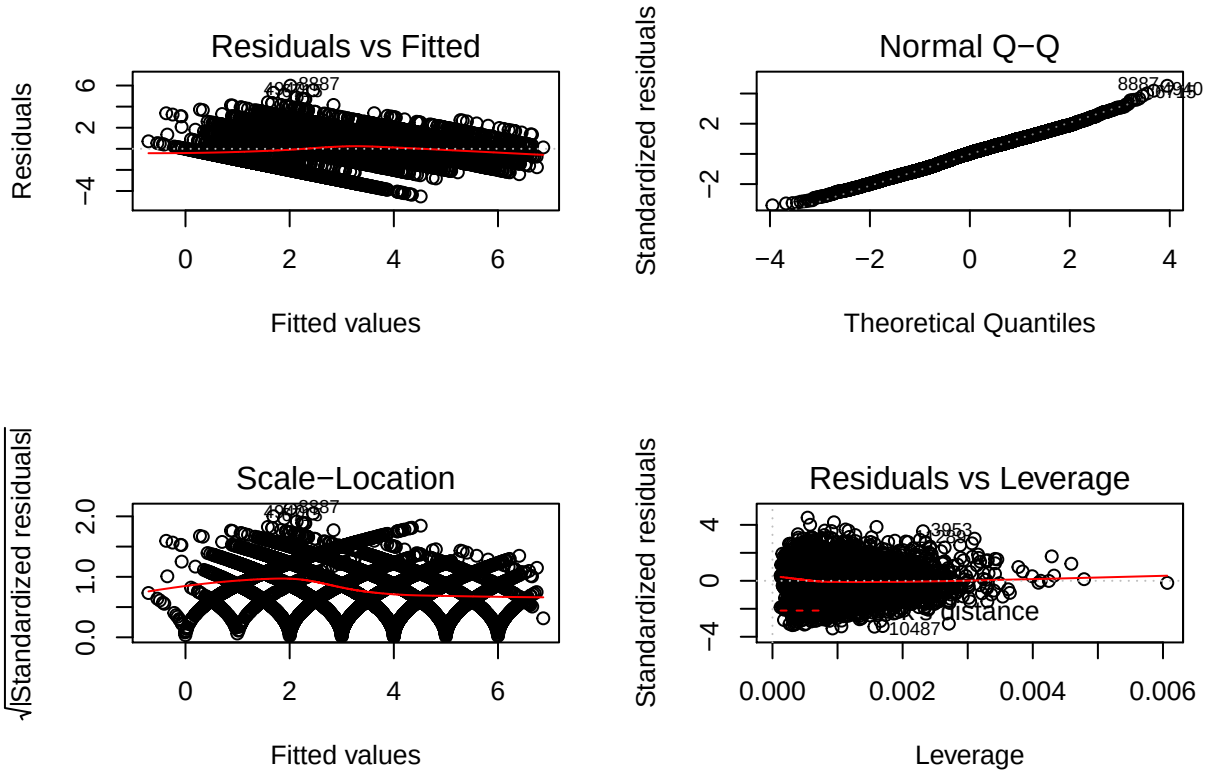
Observations

From the above model, we see all of our variables has p-values at a significance level lesser than 0.05. This shows the model is very good.

Also this model is parsimonous compared to the Base model.

Let us do some plots and discuss about them.

```
par(mfrow=c(2,2))
plot(regbasebackward)
```



The Q-Q Plot which shows the normality of residual is near normal plot.

Residual vs Fitted plot displays almost constant variance.

```
knitr::kable(vif(regbasebackward))
```

	x
VolatileAcidity	1.004826
Chlorides	1.001993
TotalSulfurDioxide	1.004684
pH	1.004874
Sulphates	1.002834
Alcohol	1.006528
LabelAppeal	1.081945
AcidIndex	1.044634
STARS	1.119719

None of the variables has variance inflation factor > 4. Which indicates it is signficant model compared to base model.

Step Model

3. For our next model, we will use the step function on the base model and transformation variables.

```
base_step_model = step(regbaseplustransform)
```

```
## Start:  AIC=7237.29
## TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid + ResidualSugar +
##     Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
##     pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS
```

```
##
##                       Df Sum of Sq   RSS     AIC
## - FixedAcidity         1       0.0 22474  7235.3
## - ResidualSugar        1       0.0 22474  7235.3
## <none>                             22474  7237.3
## - FreeSulfurDioxide    1       5.5 22479  7238.4
## - CitricAcid           1       5.6 22479  7238.5
## - Chlorides            1       6.3 22480  7238.9
## - Density              1       6.3 22480  7238.9
## - pH                   1       7.2 22481  7239.4
## - Sulphates            1      10.2 22484  7241.1
## - Alcohol              1      23.7 22497  7248.8
## - TotalSulfurDioxide   1      24.0 22498  7248.9
## - VolatileAcidity      1      52.0 22526  7264.8
## - AcidIndex            1     917.0 23391  7747.0
## - LabelAppeal          1    1748.5 24222  8194.0
## - STARS                1   15424.8 37898 13921.5
##
## Step:  AIC=7235.29
## TARGET ~ VolatileAcidity + CitricAcid + ResidualSugar + Chlorides +
##     FreeSulfurDioxide + TotalSulfurDioxide + Density + pH + Sulphates +
##     Alcohol + LabelAppeal + AcidIndex + STARS
##
##                       Df Sum of Sq   RSS     AIC
## - ResidualSugar        1       0.0 22474  7233.3
## <none>                             22474  7235.3
## - FreeSulfurDioxide    1       5.5 22479  7236.4
## - CitricAcid           1       5.6 22479  7236.5
## - Chlorides            1       6.3 22480  7236.9
## - Density              1       6.3 22480  7236.9
## - pH                   1       7.2 22481  7237.4
## - Sulphates            1      10.2 22484  7239.1
## - Alcohol              1      23.7 22497  7246.8
## - TotalSulfurDioxide   1      24.0 22498  7246.9
## - VolatileAcidity      1      52.0 22526  7262.8
## - AcidIndex            1     945.9 23420  7760.8
## - LabelAppeal          1    1748.6 24222  8192.0
## - STARS                1   15426.2 37900 13920.0
##
## Step:  AIC=7233.3
## TARGET ~ VolatileAcidity + CitricAcid + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + Density + pH + Sulphates + Alcohol +
##     LabelAppeal + AcidIndex + STARS
##
##                       Df Sum of Sq   RSS     AIC
## <none>                             22474  7233.3
## - FreeSulfurDioxide    1       5.5 22479  7234.4
## - CitricAcid           1       5.6 22479  7234.5
## - Chlorides            1       6.3 22480  7234.9
## - Density              1       6.3 22480  7234.9
## - pH                   1       7.2 22481  7235.4
## - Sulphates            1      10.2 22484  7237.1
## - Alcohol              1      23.7 22497  7244.8
## - TotalSulfurDioxide   1      23.9 22498  7244.9
```

```
## - VolatileAcidity     1      52.0 22526  7260.9
## - AcidIndex           1     945.9 23420  7758.8
## - LabelAppeal         1    1748.6 24222  8190.0
## - STARS               1   15426.3 37900 13918.0
```

```
summary(base_step_model)
```

```
##
## Call:
## lm(formula = TARGET ~ VolatileAcidity + CitricAcid + Chlorides +
##     FreeSulfurDioxide + TotalSulfurDioxide + Density + pH + Sulphates +
##     Alcohol + LabelAppeal + AcidIndex + STARS, data = train_imputed)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4931 -0.9562  0.0607  0.9126  6.0326
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         4.046e+00  4.501e-01   8.989  < 2e-16 ***
## VolatileAcidity    -1.150e-01  2.115e-02  -5.437 5.53e-08 ***
## CitricAcid          3.460e-02  1.937e-02   1.786 0.074060 .
## Chlorides          -9.691e-02  5.135e-02  -1.887 0.059147 .
## FreeSulfurDioxide   1.968e-04  1.113e-04   1.769 0.076974 .
## TotalSulfurDioxide  2.730e-04  7.399e-05   3.690 0.000225 ***
## Density            -8.398e-01  4.424e-01  -1.898 0.057695 .
## pH                 -3.557e-02  1.756e-02  -2.026 0.042760 *
## Sulphates          -4.522e-02  1.879e-02  -2.407 0.016100 *
## Alcohol             1.221e-02  3.323e-03   3.674 0.000240 ***
## LabelAppeal         4.316e-01  1.369e-02  31.536  < 2e-16 ***
## AcidIndex          -2.103e-01  9.065e-03 -23.195  < 2e-16 ***
## STARS               9.795e-01  1.046e-02  93.668  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.326 on 12782 degrees of freedom
## Multiple R-squared:  0.5266, Adjusted R-squared:  0.5262
## F-statistic:  1185 on 12 and 12782 DF,  p-value: < 2.2e-16
```

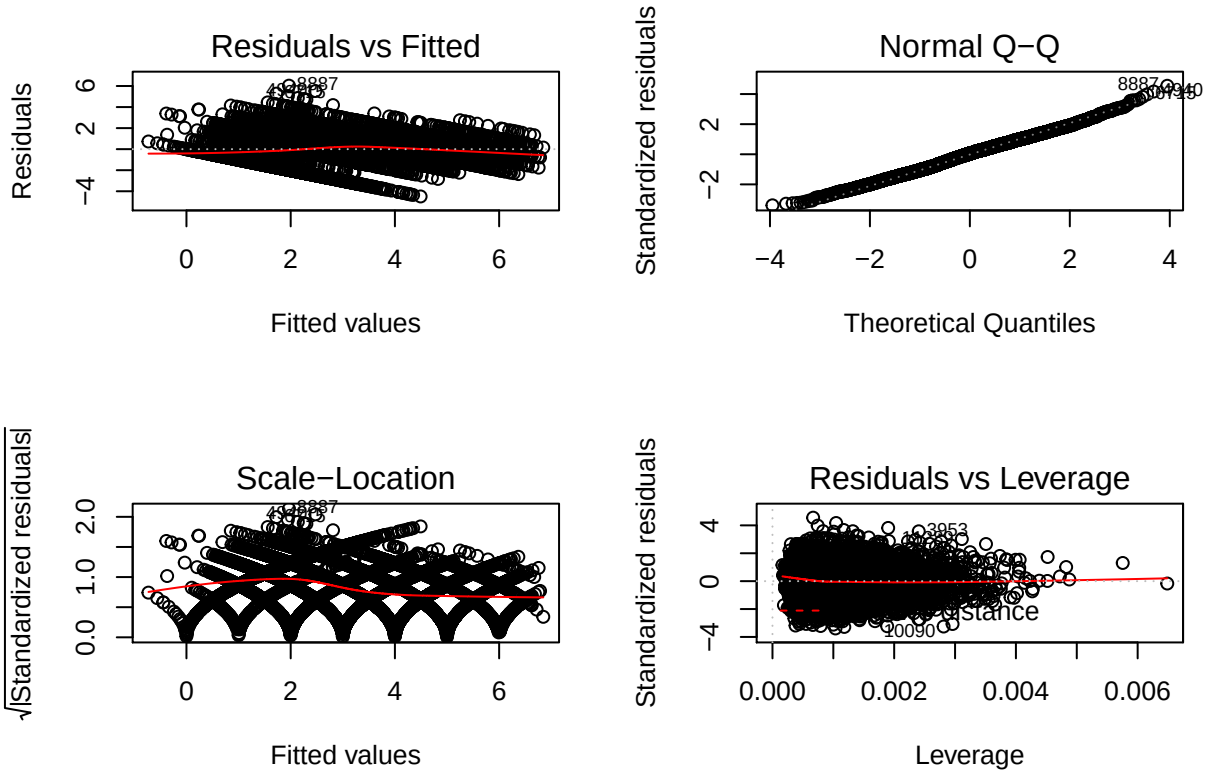
Observations

From the above model, we see 3 variables dropped from 15.

Also this model is parsimonous compared to the Base model but the backward elimination model was better with 9 variables.

Let us do some plots and discuss about them.

```
par(mfrow=c(2,2))
plot(base_step_model)
```



The Q-Q Plot which shows the normality of residual is near normal plot.

Residual vs Fitted plot displays almost constant variance. But as the backward elimination the standardized residual displays nonconstant variance.

```
knitr::kable(vif(base_step_model))
```

	x
VolatileAcidity	1.004949
CitricAcid	1.002401
Chlorides	1.002233
FreeSulfurDioxide	1.001298
TotalSulfurDioxide	1.005310
Density	1.003008
pH	1.005027
Sulphates	1.003273
Alcohol	1.006715
LabelAppeal	1.082268
AcidIndex	1.048009
STARS	1.120224

None of the variables has variance inflation factor > 4. Which indicates it is signficant model compared to base model.

Comparison of Linear Regression Models.

```
results = NULL
modellist = list(linearBase = regbaseplustransform, linearBackWard = regbasebackward, LinearBICStep= bas

for(i in names(modellist)){
```

```
    s = summary(modellist[[i]])
    name = i
    mse <- mean(s$residuals^2)
    r2 <- s$r.squared
    f <- s$fstatistic[1]
    k <- s$fstatistic[2]
    n <- s$fstatistic[3]
    results = rbind(results, data.frame(
        name = name, rsquared = r2, mse = mse, f = f,
        k = k, n = n))
}
rownames(results) = NULL
results
```

```
##             name  rsquared      mse        f  k     n
## 1     linearBase 0.5266431 1.756440 1015.618 14 12780
## 2 linearBackWard 0.5262726 1.757815 1578.122  9 12785
## 3  LinearBICStep 0.5266425 1.756442 1185.071 12 12782
```

Poisson Regression Analysis.

In our experiment the response variable is Count, so the Poisson Regression is the way to go to solve this problem.

As base model we will use the base entire set of variables.

```
posmod = glm(TARGET ~ ., family = "poisson", data = train_imputed)
summary(posmod)
```

```
##
## Call:
## glm(formula = TARGET ~ ., family = "poisson", data = train_imputed)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9413  -0.7207   0.0674   0.5798   3.2595
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.553e+00  1.960e-01   7.921 2.35e-15 ***
## FixedAcidity        -4.407e-04  1.046e-03  -0.421  0.67357
## VolatileAcidity     -4.155e-02  9.394e-03  -4.424 9.71e-06 ***
## CitricAcid           1.201e-02  8.321e-03   1.443  0.14893
## ResidualSugar        3.300e-05  2.087e-04   0.158  0.87436
## Chlorides           -3.686e-02  2.241e-02  -1.645  0.09994 .
## FreeSulfurDioxide    7.405e-05  4.817e-05   1.537  0.12423
## TotalSulfurDioxide   1.029e-04  3.193e-05   3.223  0.00127 **
## Density             -2.975e-01  1.921e-01  -1.549  0.12145
## pH                  -1.619e-02  7.632e-03  -2.121  0.03393 *
## Sulphates           -1.857e-02  8.272e-03  -2.245  0.02476 *
## Alcohol              2.596e-03  1.445e-03   1.797  0.07230 .
## LabelAppeal          1.330e-01  6.064e-03  21.938  < 2e-16 ***
## AcidIndex           -8.762e-02  4.542e-03 -19.290  < 2e-16 ***
## STARS                3.121e-01  4.527e-03  68.950  < 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: 22861  on 12794  degrees of freedom
## Residual deviance: 14750  on 12780  degrees of freedom
## AIC: 46722
##
## Number of Fisher Scoring iterations: 5
```

```
#knitr::kable(vif(posmod))
```

Observations

Deviance is approximately normally distributed since the Median is almost zero.

None of the variables has VIF greater than 4.

Since the above coefficients are in exponentional terms will do an exponential transformation. From the below we can say that for every 1 point in increase in STARS rating the number of cases purchased will increase by 1.366.

With p-value of chisq test almost zero, this shows that the deviance is not small enough for a good fit.

```
cov.m1 = vcovHC(posmod, type="HC0")
std.err = sqrt(diag(cov.m1))
r.est = cbind(Estimate= exp(coef(posmod)), "Robust SE" = std.err,
LL = exp(coef(posmod)) - 1.96 * std.err,
UL = exp(coef(posmod)) + 1.96 * std.err)

r.est
```

```
##                     Estimate    Robust SE        LL        UL
## (Intercept)        4.7235070 1.611492e-01 4.4076545 5.0393596
## FixedAcidity       0.9995594 8.570141e-04 0.9978796 1.0012391
## VolatileAcidity    0.9592969 7.796614e-03 0.9440156 0.9745783
## CitricAcid         1.0120815 6.510783e-03 0.9993204 1.0248427
## ResidualSugar      1.0000330 1.648062e-04 0.9997100 1.0003560
## Chlorides          0.9638094 1.802079e-02 0.9284887 0.9991302
## FreeSulfurDioxide  1.0000741 3.890326e-05 0.9999978 1.0001503
## TotalSulfurDioxide 1.0001029 2.563404e-05 1.0000527 1.0001532
## Density            0.7427008 1.566654e-01 0.4356366 1.0497651
## pH                 0.9839442 6.226114e-03 0.9717410 0.9961473
## Sulphates          0.9816002 6.820051e-03 0.9682329 0.9949675
## Alcohol            1.0025996 1.160229e-03 1.0003256 1.0048737
## LabelAppeal        1.1422956 5.333082e-03 1.1318428 1.1527485
## AcidIndex          0.9161089 4.519742e-03 0.9072502 0.9249676
## STARS              1.3663039 4.766981e-03 1.3569607 1.3756472
```

```
with(posmod, cbind(res.deviance = deviance, df = df.residual,
  p = pchisq(deviance, df.residual, lower.tail=FALSE)))
```

```
##      res.deviance    df            p
## [1,]     14750.13 12780 3.731438e-32
```

Poisson Base Step

We will use the base model and do a step function on it.

```r
posmod = glm(TARGET ~ ., family = "poisson", data = train_imputed)
pos_step_model = step(posmod)
```

```
## Start:  AIC=46722.15
## TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid + ResidualSugar +
##     Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
##     pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS
##
##                       Df Deviance   AIC
## - ResidualSugar        1    14750 46720
## - FixedAcidity         1    14750 46720
## <none>                      14750 46722
## - CitricAcid           1    14752 46722
## - FreeSulfurDioxide    1    14752 46723
## - Density              1    14752 46723
## - Chlorides            1    14753 46723
## - Alcohol              1    14753 46723
## - pH                   1    14755 46725
## - Sulphates            1    14755 46725
## - TotalSulfurDioxide   1    14760 46730
## - VolatileAcidity      1    14770 46740
## - AcidIndex            1    15140 47110
## - LabelAppeal          1    15231 47201
## - STARS                1    19613 51583
##
## Step:  AIC=46720.18
## TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid + Chlorides +
##     FreeSulfurDioxide + TotalSulfurDioxide + Density + pH + Sulphates +
##     Alcohol + LabelAppeal + AcidIndex + STARS
##
##                       Df Deviance   AIC
## - FixedAcidity         1    14750 46718
## <none>                      14750 46720
## - CitricAcid           1    14752 46720
## - FreeSulfurDioxide    1    14752 46721
## - Density              1    14753 46721
## - Chlorides            1    14753 46721
## - Alcohol              1    14753 46721
## - pH                   1    14755 46723
## - Sulphates            1    14755 46723
## - TotalSulfurDioxide   1    14760 46728
## - VolatileAcidity      1    14770 46738
## - AcidIndex            1    15140 47108
## - LabelAppeal          1    15231 47199
## - STARS                1    19613 51581
##
## Step:  AIC=46718.36
## TARGET ~ VolatileAcidity + CitricAcid + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + Density + pH + Sulphates + Alcohol +
##     LabelAppeal + AcidIndex + STARS
##
##                       Df Deviance   AIC
## <none>                      14750 46718
## - CitricAcid           1    14752 46718
```

```
## - FreeSulfurDioxide   1    14753 46719
## - Density             1    14753 46719
## - Chlorides           1    14753 46719
## - Alcohol             1    14754 46720
## - pH                  1    14755 46721
## - Sulphates           1    14755 46721
## - TotalSulfurDioxide  1    14761 46727
## - VolatileAcidity     1    14770 46736
## - AcidIndex           1    15152 47118
## - LabelAppeal         1    15231 47197
## - STARS               1    19614 51580
```

```
summary(pos_step_model)
```

```
##
## Call:
## glm(formula = TARGET ~ VolatileAcidity + CitricAcid + Chlorides +
##     FreeSulfurDioxide + TotalSulfurDioxide + Density + pH + Sulphates +
##     Alcohol + LabelAppeal + AcidIndex + STARS, family = "poisson",
##     data = train_imputed)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9482  -0.7190   0.0672   0.5787   3.2644
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.552e+00  1.959e-01   7.922 2.33e-15 ***
## VolatileAcidity     -4.159e-02  9.394e-03  -4.427 9.53e-06 ***
## CitricAcid           1.204e-02  8.319e-03   1.448  0.14774
## Chlorides           -3.686e-02  2.241e-02  -1.645  0.09998 .
## FreeSulfurDioxide    7.405e-05  4.817e-05   1.537  0.12424
## TotalSulfurDioxide   1.029e-04  3.193e-05   3.224  0.00127 **
## Density             -2.972e-01  1.921e-01  -1.547  0.12177
## pH                  -1.621e-02  7.631e-03  -2.125  0.03361 *
## Sulphates           -1.862e-02  8.271e-03  -2.252  0.02435 *
## Alcohol              2.596e-03  1.445e-03   1.797  0.07234 .
## LabelAppeal          1.330e-01  6.064e-03  21.939  < 2e-16 ***
## AcidIndex           -8.790e-02  4.494e-03 -19.561  < 2e-16 ***
## STARS                3.121e-01  4.527e-03  68.954  < 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: 22861  on 12794  degrees of freedom
## Residual deviance: 14750  on 12782  degrees of freedom
## AIC: 46718
##
## Number of Fisher Scoring iterations: 5
```

```
cov.m1 = vcovHC(pos_step_model, type="HC0")
std.err = sqrt(diag(cov.m1))
r.est = cbind(Estimate= exp(coef(pos_step_model)), "Robust SE" = std.err,
LL = exp(coef(pos_step_model)) - 1.96 * std.err,
```

```
UL = exp(coef(pos_step_model)) + 1.96 * std.err)

r.est
```

```
##                       Estimate    Robust SE        LL        UL
## (Intercept)          4.7199901 1.610807e-01 4.4042720 5.0357083
## VolatileAcidity      0.9592625 7.797325e-03 0.9439798 0.9745453
## CitricAcid           1.0121155 6.513242e-03 0.9993495 1.0248815
## Chlorides            0.9638149 1.802058e-02 0.9284946 0.9991353
## FreeSulfurDioxide    1.0000740 3.890101e-05 0.9999978 1.0001503
## TotalSulfurDioxide   1.0001029 2.564251e-05 1.0000527 1.0001532
## Density              0.7428946 1.566692e-01 0.4358231 1.0499662
## pH                   0.9839175 6.225774e-03 0.9717150 0.9961200
## Sulphates            0.9815493 6.818663e-03 0.9681848 0.9949139
## Alcohol              1.0025993 1.160238e-03 1.0003252 1.0048734
## LabelAppeal          1.1422887 5.332456e-03 1.1318371 1.1527403
## AcidIndex            0.9158528 4.486931e-03 0.9070584 0.9246471
## STARS                1.3663223 4.766979e-03 1.3569791 1.3756656
```

```
with(pos_step_model, cbind(res.deviance = deviance, df = df.residual,
  p = pchisq(deviance, df.residual, lower.tail=FALSE)))
```

```
##      res.deviance    df            p
## [1,]     14750.33 12782 4.253026e-32
```

Compared to the base model the BIC Model is more parsimonous, but the rest of output values are almost similar to base model.

Negative Binomial Regression Analysis.

As part of Negative Binom, we are going to do BIC on base model.

```
base_neg_model = glm.nb(TARGET ~ ., data = train_imputed)

step_neg_base_model = step(base_neg_model, k = log(n), trace = 0)
summary(step_neg_base_model)
```

```
##
## Call:
## glm.nb(formula = TARGET ~ VolatileAcidity + TotalSulfurDioxide +
##     LabelAppeal + AcidIndex + STARS, data = train_imputed, init.theta = 48864.73642,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0071  -0.7166   0.0618   0.5746   3.2583
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.2248603  0.0377606  32.438  < 2e-16 ***
## VolatileAcidity     -0.0417893  0.0093922  -4.449 8.61e-06 ***
## TotalSulfurDioxide   0.0001020  0.0000319   3.197  0.00139 **
## LabelAppeal          0.1329614  0.0060619  21.934  < 2e-16 ***
## AcidIndex           -0.0881735  0.0044692 -19.729  < 2e-16 ***
## STARS                0.3132517  0.0045129  69.413  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(48864.74) family taken to be 1)
##
##     Null deviance: 22860  on 12794  degrees of freedom
## Residual deviance: 14773  on 12789  degrees of freedom
## AIC: 46729
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  48865
##           Std. Err.:  50635
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -46715.48
```

```
#knitr::kable(vif(step_neg_base_model))
```

p-value of all the variables are less than 0.05. This makes the model ideal for prediction.

p-Value from Chi Sq test is almost near zero, which says deviance is not small enough for a good fit.

None of the variables has VIF > 4. No multicollinearity issue.

```
cov.m1 = vcovHC(step_neg_base_model, type="HC0")
std.err = sqrt(diag(cov.m1))
r.est = cbind(Estimate= exp(coef(step_neg_base_model)), "Robust SE" = std.err,
LL = exp(coef(step_neg_base_model)) - 1.96 * std.err,
UL = exp(coef(step_neg_base_model)) + 1.96 * std.err)

r.est
```

```
##                      Estimate    Robust SE        LL        UL
## (Intercept)        3.4036906 3.729208e-02 3.3305982 3.4767831
## VolatileAcidity    0.9590718 7.801612e-03 0.9437807 0.9743630
## TotalSulfurDioxide 1.0001020 2.563691e-05 1.0000517 1.0001522
## LabelAppeal        1.1422059 5.329529e-03 1.1317600 1.1526518
## AcidIndex          0.9156020 4.460576e-03 0.9068592 0.9243447
## STARS              1.3678658 4.752512e-03 1.3585508 1.3771807
```

```
with(step_neg_base_model, cbind(res.deviance = deviance, df = df.residual,
  p = pchisq(deviance, df.residual, lower.tail=FALSE)))
```

```
##      res.deviance    df            p
## [1,]     14772.67 12789 1.565761e-32
```

4. MODEL SELECTION:

Coefficient Comparison

Comparison of the coefficient between Linear Model and Generalized Linear Models. We can see the for LM models the value is between 3 and 4. For GLM models the value is hovering around (1.5). For both Poisson and Negative Binomial, there is no much difference. The values are almost identical.

```
knitr::kable(regbaseplustransform$coefficients)
```

	x
(Intercept)	4.0475426
FixedAcidity	0.0000053
VolatileAcidity	-0.1149779
CitricAcid	0.0345684
ResidualSugar	-0.0000590
Chlorides	-0.0969182
FreeSulfurDioxide	0.0001967
TotalSulfurDioxide	0.0002732
Density	-0.8399127
pH	-0.0355722
Sulphates	-0.0452343
Alcohol	0.0122082
LabelAppeal	0.4316034
AcidIndex	-0.2102681
STARS	0.9794917

```
knitr::kable(regbasebackward$coefficients)
```

	x
(Intercept)	3.2595186
VolatileAcidity	-0.1151829
Chlorides	-0.0990797
TotalSulfurDioxide	0.0002742
pH	-0.0359055
Sulphates	-0.0450213
Alcohol	0.0121474
LabelAppeal	0.4323019
AcidIndex	-0.2106073
STARS	0.9801435

```
knitr::kable(base_step_model$coefficients)
```

	x
(Intercept)	4.0459292
VolatileAcidity	-0.1149917
CitricAcid	0.0345983
Chlorides	-0.0969077
FreeSulfurDioxide	0.0001968
TotalSulfurDioxide	0.0002730
Density	-0.8397707
pH	-0.0355737
Sulphates	-0.0452190
Alcohol	0.0122110
LabelAppeal	0.4316046
AcidIndex	-0.2102544
STARS	0.9794858

```
knitr::kable(posmod$coefficients)
```

	x
(Intercept)	1.5525515
FixedAcidity	-0.0004407
VolatileAcidity	-0.0415546
CitricAcid	0.0120091
ResidualSugar	0.0000330
Chlorides	-0.0368617
FreeSulfurDioxide	0.0000741
TotalSulfurDioxide	0.0001029
Density	-0.2974620
pH	-0.0161861
Sulphates	-0.0185712
Alcohol	0.0025963
LabelAppeal	0.1330400
AcidIndex	-0.0876200
STARS	0.3121092

```
knitr::kable(pos_step_model$coefficients)
```

	x
(Intercept)	1.5518067
VolatileAcidity	-0.0415905
CitricAcid	0.0120427
Chlorides	-0.0368560
FreeSulfurDioxide	0.0000740
TotalSulfurDioxide	0.0001029
Density	-0.2972010
pH	-0.0162132
Sulphates	-0.0186230
Alcohol	0.0025959
LabelAppeal	0.1330339
AcidIndex	-0.0878997
STARS	0.3121227

```
knitr::kable(step_neg_base_model$coefficients)
```

	x
(Intercept)	1.2248603
VolatileAcidity	-0.0417893
TotalSulfurDioxide	0.0001020
LabelAppeal	0.1329614
AcidIndex	-0.0881735
STARS	0.3132517

RMSE

Comparison of Root Mean Square Error between LM models and GLM Models. The value for LM Models

are less compared to GLM models.

There is no difference between poisson and negative binomial RMSE values. They are identical. The Chi-Sq test value of negative binomial was better compared to the rest also the model has parsimonious variable compared to other models. So we can pick Negative Binom Model for our final prediction analysis.

```
rmslmbase = qpcR::RMSE(regbaseplustransform)
rmslmBack = qpcR::RMSE(regbasebackward)
rmslmStep = qpcR::RMSE(base_step_model)
rmsposStep = sqrt(boot::cv.glm(pos_step_model$model, pos_step_model, K = 50)$delta[1])
rmsposBase = sqrt(boot::cv.glm(posmod$model, posmod, K = 50)$delta[1])
rmsNb = sqrt(boot::cv.glm(step_neg_base_model$model, step_neg_base_model, K = 50)$delta[1])
RMSEResults = data.frame("Base Model" = rmslmbase,
                         "Base Backward Elimination" = rmslmBack,
                         "Base Step" = rmslmStep,
                         "Pos Base" = rmsposBase,
                         "Pos Step Base" = rmsposStep,
                         "Negative Binom" = rmsNb)

knitr::kable(RMSEResults)
```

Base.Model	Base.Backward.Elimination	Base.Step	Pos.Base	Pos.Step.Base	Negative.Binom
1.325308	1.325826	1.325308	1.40612	1.405995	1.406186

Evalution

For our final model we picked the negative binomial with varying dispersion for prediction.

```
summary(predict(step_neg_base_model, newdata = imputed_eval$ximp, type = "response"))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.6937  1.9241  2.7004  3.0573  3.7850  9.9201
```

References

https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/

https://stats.idre.ucla.edu/r/dae/poisson-regression/