Data 621 Homewor 5

Vanita Thompson, David Moste, Sadia Perveen

May 6, 2021

Contents

Introduction

In this homework assignment, you will explore, analyze and model a dataset containing information on approximately 12,000 commercially available wines.

Discuss the coefficients in the models and decide on the criteria for selecting the best count regression model. For the count regression model, will you use a metric such as AIC, average squared error.

About the Data

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.

Below is a short description of the variables of interest in the data set:

INDEX: Identification Variable (do not use) TARGET: Number of Cases Purchased AcidIndex: Proprietary method of testing total acidity of wine by using a weighted average Alcohol: Alcohol Content Chlorides: Chloride content of wine CitricAcid: Citric Acid Content Density: Density of Wine FixedAcidity: Fixed Acidity of Wine FreeSulfurDioxide: Sulfur Dioxide content of wine LabelAppeal: Marketing Score indicating the appeal of label design for consumers. High numbers suggest customers like the label design. Negative numbers suggest customes don't like the design. ResidualSugar: Residual Sugar of wine STARS: Wine

rating by a team of experts. 4 Stars = Excellent, 1 Star = Poor Sulphates: Sulfate conten of wine TotalSulfurDioxide: Total Sulfur Dioxide of Wine VolatileAcidity: Volatile Acid content of wine pH: pH of wine

Objective

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.

THe 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.

Load Libraries

```
library(ggplot2)
library(ggcorrplot)
library(dplyr)
library(caret)
library(MASS)
library(imputeTS) # Used for imputing missing values
library(nortest) # Test for normality
library(moments) # Skewness and kurtosis
library(glmnet)
library(mltest)
library(car)
library(rpart)
library(rpart.plot)
library(pscl)
library(boot)
library(broom) # glance function
library(WVPlots) # Gini Curve plot
library(modelr) # rsquare function
```

Load the training and evaluation data sets

I will use the training data to train the model and use the evaluation data set to test/evaluate the model.

```
rawwine <- read.csv("https://raw.githubusercontent.com/dmoste/DATA621/master/hw5/wine-training-data.csv
rawwine_evaluation <- read.csv("https://raw.githubusercontent.com/dmoste/DATA621/master/hw5/wine-evalua
```

Data Exploration

Descriptive Statistics

We can start exploring our training data set by looking at basic descriptive statistics. Look at the training dataset structure and drop the Index variable

```
wine = subset(rawwine, select = -c(ï..INDEX))
str(wine)
```

```
## 'data.frame':    12795 obs. of  15 variables:
##  $ TARGET            : int  3 3 5 3 4 0 0 4 3 6 ...
##  $ FixedAcidity      : num  3.2 4.5 7.1 5.7 8 11.3 7.7 6.5 14.8 5.5 ...
##  $ VolatileAcidity   : num  1.16 0.16 2.64 0.385 0.33 0.32 0.29 -1.22 0.27 -0.22 ...
##  $ CitricAcid        : num  -0.98 -0.81 -0.88 0.04 -1.26 0.59 -0.4 0.34 1.05 0.39 ...
##  $ ResidualSugar     : num  54.2 26.1 14.8 18.8 9.4 ...
##  $ Chlorides         : num  -0.567 -0.425 0.037 -0.425 NA 0.556 0.06 0.04 -0.007 -0.277 ...
##  $ FreeSulfurDioxide : num  NA 15 214 22 -167 -37 287 523 -213 62 ...
##  $ TotalSulfurDioxide: num  268 -327 142 115 108 15 156 551 NA 180 ...
##  $ Density           : num  0.993 1.028 0.995 0.996 0.995 ...
##  $ pH                : num  3.33 3.38 3.12 2.24 3.12 3.2 3.49 3.2 4.93 3.09 ...
##  $ Sulphates         : num  -0.59 0.7 0.48 1.83 1.77 1.29 1.21 NA 0.26 0.75 ...
##  $ Alcohol           : num  9.9 NA 22 6.2 13.7 15.4 10.3 11.6 15 12.6 ...
##  $ LabelAppeal       : int  0 -1 -1 -1 0 0 0 1 0 0 ...
##  $ AcidIndex         : int  8 7 8 6 9 11 8 7 6 8 ...
##  $ STARS             : int  2 3 3 1 2 NA NA 3 NA 4 ...
```

The training data set has 12,795 observations with 15 variables. All the variables are numeric/integer.

Look at the evaluation dataset structure and drop the TARGET variables. The TARGET Variable was empty and needed to be predicted at the end.

I do not need it now.

```
wine_evaluation = subset(rawwine_evaluation, select = -c(TARGET))
str(wine_evaluation)
```

```
## 'data.frame':    3335 obs. of  15 variables:
##  $ IN                : int  3 9 10 18 21 30 31 37 39 47 ...
##  $ FixedAcidity      : num  5.4 12.4 7.2 6.2 11.4 17.6 15.5 15.9 11.6 3.8 ...
##  $ VolatileAcidity   : num  -0.86 0.385 1.75 0.1 0.21 0.04 0.53 1.19 0.32 0.22 ...
##  $ CitricAcid        : num  0.27 -0.76 0.17 1.8 0.28 -1.15 -0.53 1.14 0.55 0.31 ...
##  $ ResidualSugar     : num  -10.7 -19.7 -33 1 1.2 1.4 4.6 31.9 -50.9 -7.7 ...
##  $ Chlorides         : num  0.092 1.169 0.065 -0.179 0.038 ...
##  $ FreeSulfurDioxide : num  23 -37 9 104 70 -250 10 115 35 40 ...
##  $ TotalSulfurDioxide: num  398 68 76 89 53 140 17 381 83 129 ...
##  $ Density           : num  0.985 0.99 1.046 0.989 1.029 ...
##  $ pH                : num  5.02 3.37 4.61 3.2 2.54 3.06 3.07 2.99 3.32 4.72 ...
##  $ Sulphates         : num  0.64 1.09 0.68 2.11 -0.07 -0.02 0.75 0.31 2.18 -0.64 ...
##  $ Alcohol           : num  12.3 16 8.55 12.3 4.8 11.4 8.5 11.4 -0.5 10.9 ...
##  $ LabelAppeal       : int  -1 0 0 -1 0 1 0 1 0 0 ...
##  $ AcidIndex         : int  6 6 8 8 10 8 12 7 12 7 ...
##  $ STARS             : int  NA 2 1 1 NA 4 3 NA NA NA ...
```

The evaluation data set has 3,335 observations with 14 variables; all the variables are numerical/integers.

Look at descriptive statistics for both datasets.

```
summary(wine)
```

```
##      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
```

```
summary(wine_evaluation)
```

```
##        IN          FixedAcidity      VolatileAcidity     CitricAcid
##  Min.   :    3   Min.   :-18.200   Min.   :-2.8300   Min.   :-3.1200
##  1st Qu.: 4018   1st Qu.:  5.200   1st Qu.: 0.0800   1st Qu.: 0.0000
##  Median : 7906   Median :  6.900   Median : 0.2800   Median : 0.3100
##  Mean   : 8048   Mean   :  6.864   Mean   : 0.3103   Mean   : 0.3124
##  3rd Qu.:12061   3rd Qu.:  9.000   3rd Qu.: 0.6300   3rd Qu.: 0.6050
##  Max.   :16130   Max.   : 33.500   Max.   : 3.6100   Max.   : 3.7600
##
##  ResidualSugar        Chlorides         FreeSulfurDioxide TotalSulfurDioxide
##  Min.   :-128.300   Min.   :-1.15000   Min.   :-563.00   Min.   :-769.00
##  1st Qu.:  -2.600   1st Qu.: 0.01600   1st Qu.:   3.00   1st Qu.:  27.25
##  Median :   3.600   Median : 0.04700   Median :  30.00   Median : 124.00
##  Mean   :   5.319   Mean   : 0.06143   Mean   :  34.95   Mean   : 123.41
##  3rd Qu.:  17.200   3rd Qu.: 0.17100   3rd Qu.:  79.25   3rd Qu.: 210.00
##  Max.   : 145.400   Max.   : 1.26300   Max.   : 617.00   Max.   :1004.00
##  NA's   :168        NA's   :138        NA's   :152       NA's   :157
##     Density             pH          Sulphates          Alcohol
##  Min.   :0.8898   Min.   :0.600   Min.   :-3.0700   Min.   :-4.20
##  1st Qu.:0.9883   1st Qu.:2.980   1st Qu.: 0.3300   1st Qu.: 9.00
```

```
##  Median :0.9946   Median :3.210   Median : 0.5000   Median :10.40
##  Mean   :0.9947   Mean   :3.237   Mean   : 0.5346   Mean   :10.58
##  3rd Qu.:1.0005   3rd Qu.:3.490   3rd Qu.: 0.8200   3rd Qu.:12.50
##  Max.   :1.0998   Max.   :6.210   Max.   : 4.1800   Max.   :25.60
##                   NA's   :104     NA's   :310       NA's   :185
##   LabelAppeal         AcidIndex          STARS
##  Min.   :-2.00000   Min.   : 5.000   Min.   :1.00
##  1st Qu.:-1.00000   1st Qu.: 7.000   1st Qu.:1.00
##  Median : 0.00000   Median : 8.000   Median :2.00
##  Mean   : 0.01349   Mean   : 7.748   Mean   :2.04
##  3rd Qu.: 1.00000   3rd Qu.: 8.000   3rd Qu.:3.00
##  Max.   : 2.00000   Max.   :17.000   Max.   :4.00
##                                      NA's   :841
```

With the descriptive statistics, we are able to see mean, standard deviation, median, min, max values.

Data Preparation

In this section, we will prepare the dataset for count regression modeling. Logistic regression does not make many of the key assumptions of linear regression and general linear models that are based on ordinary least squares algorithms – particularly regarding linearity, normality, homoscedasticity, and measurement level.

Missing Values

Looking for missing values

```
colSums(is.na(wine))
```

```
##             TARGET       FixedAcidity    VolatileAcidity         CitricAcid
##                  0                  0                  0                  0
##      ResidualSugar          Chlorides  FreeSulfurDioxide TotalSulfurDioxide
##                616                638                647                682
##            Density                 pH          Sulphates            Alcohol
##                  0                395               1210                653
##        LabelAppeal         AcidIndex              STARS
##                  0                  0               3359
```

```
colSums(is.na(wine_evaluation))
```

```
##                 IN       FixedAcidity    VolatileAcidity         CitricAcid
##                  0                  0                  0                  0
##      ResidualSugar          Chlorides  FreeSulfurDioxide TotalSulfurDioxide
##                168                138                152                157
##            Density                 pH          Sulphates            Alcohol
##                  0                104                310                185
##        LabelAppeal         AcidIndex              STARS
##                  0                  0                841
```

The data set shows several missing values.

```
wineclean <- na_mean(wine, option = "mean")
wine_evaluationclean <- na_mean(wine_evaluation, option = "mean")
```

Let's check again!

```
colSums(is.na(wineclean))
```

```
##             TARGET       FixedAcidity    VolatileAcidity         CitricAcid
##                  0                  0                  0                  0
##      ResidualSugar          Chlorides  FreeSulfurDioxide TotalSulfurDioxide
##                  0                  0                  0                  0
##            Density                 pH          Sulphates            Alcohol
##                  0                  0                  0                  0
##        LabelAppeal          AcidIndex              STARS
##                  0                  0                  0
```

```
colSums(is.na(wine_evaluationclean))
```

```
##                 IN       FixedAcidity    VolatileAcidity         CitricAcid
##                  0                  0                  0                  0
##      ResidualSugar          Chlorides  FreeSulfurDioxide TotalSulfurDioxide
##                  0                  0                  0                  0
##            Density                 pH          Sulphates            Alcohol
##                  0                  0                  0                  0
##        LabelAppeal          AcidIndex              STARS
##                  0                  0                  0
```

No more missing values.

Check for Normality

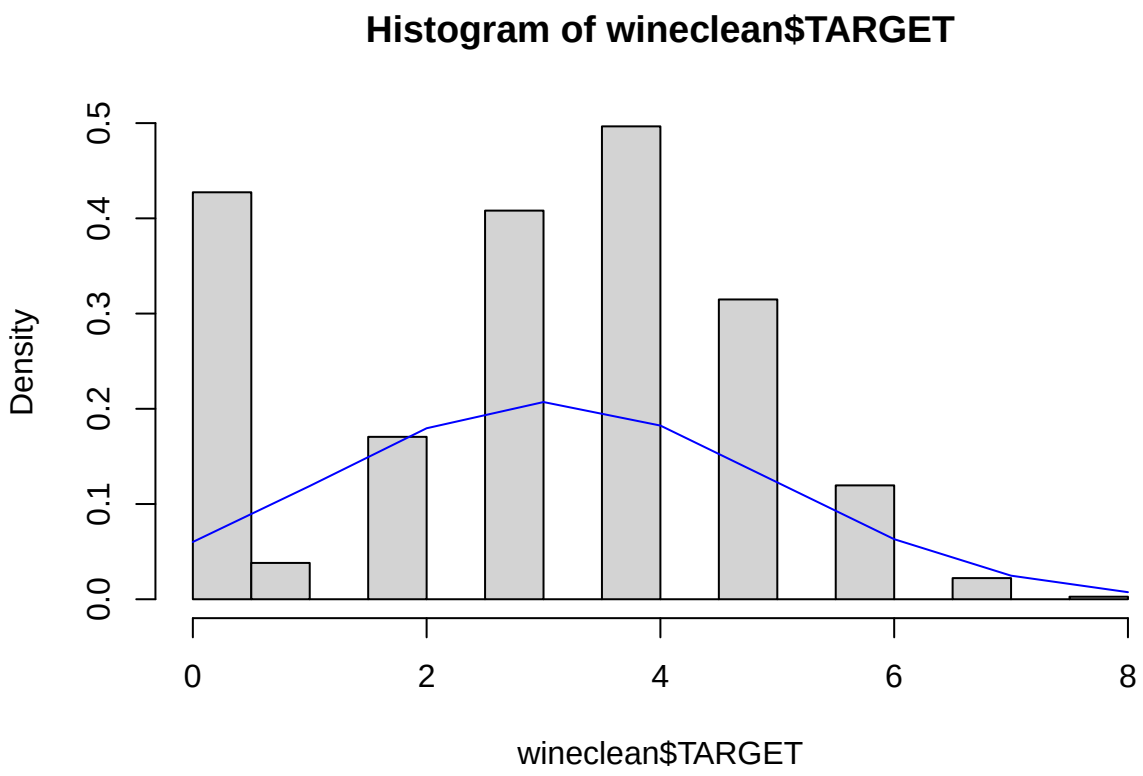
There are specific methods for testing normality but these should be used in conjunction with either a histogram or a Q-Q plot. The Kolmogorov-Smirnov test and the Shapiro-Wilk's test whether the underlying distribution is normal. Both tests are sensitive to outliers and are influenced by sample size.

Visual Inspection

Let's look at some interesting part of the data by exploring the TARGET, pH, Alcohol and STARS variables

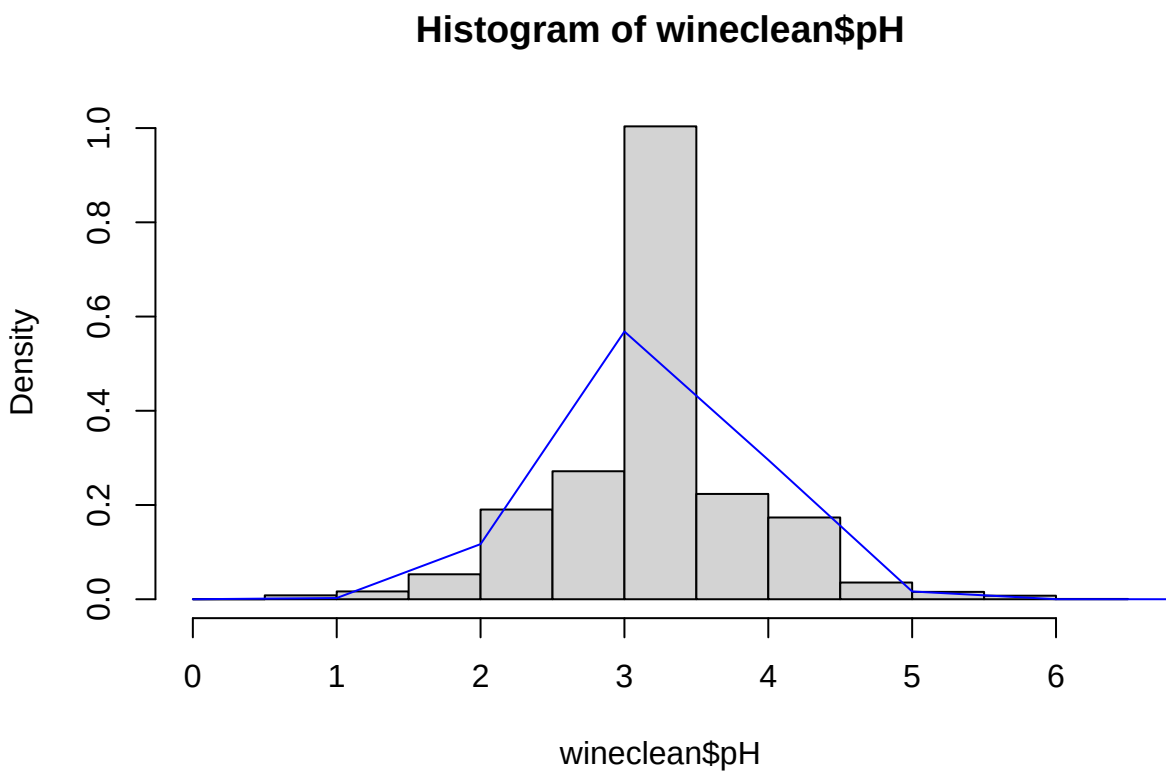
Wine - Target

```
mean = mean(wineclean$TARGET)
  sd = sd(wineclean$TARGET)
hist(wineclean$TARGET, probability = TRUE)
x <- 0:8
y <- dnorm(x = x, mean = mean, sd = sd)
lines(x = x, y = y, col = "blue") #The Target distribution is hardly normal!
```



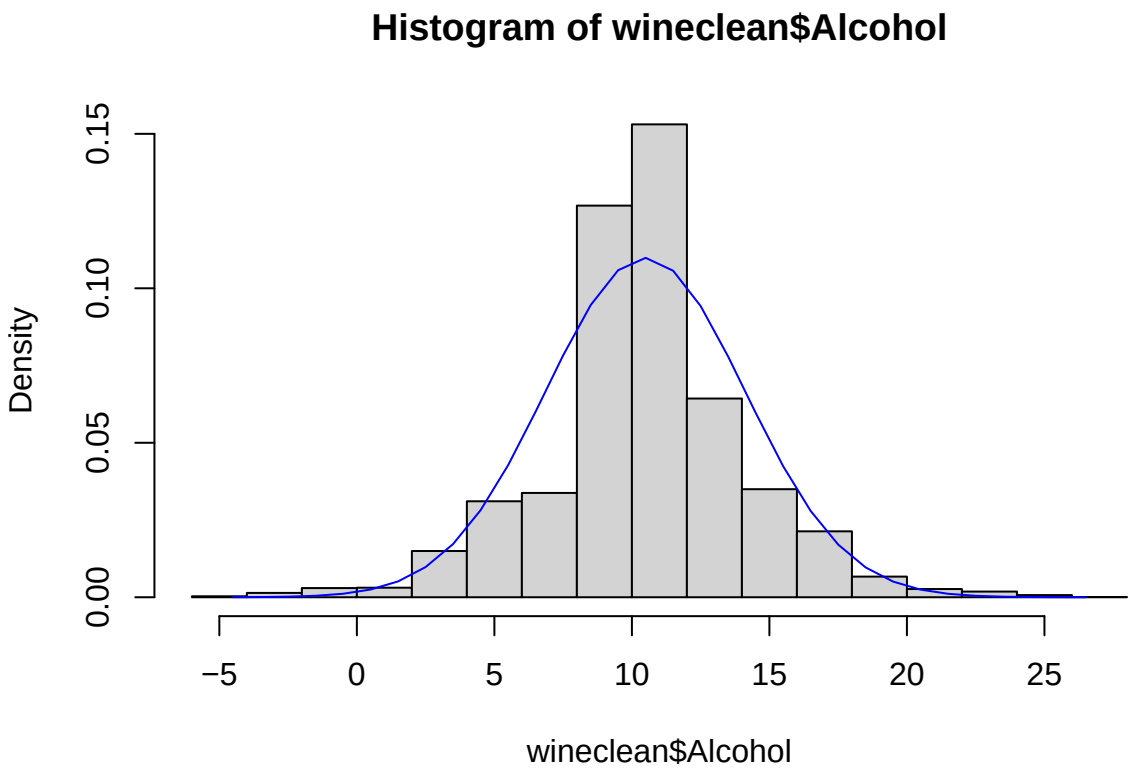
Wine - pH

```
mean = mean(wineclean$pH)
  sd = sd(wineclean$pH)
hist(wineclean$pH, probability = TRUE)
x <- 0:7
y <- dnorm(x = x, mean = mean, sd = sd)
lines(x = x, y = y, col = "blue") #The pH distribution is not normal!
```



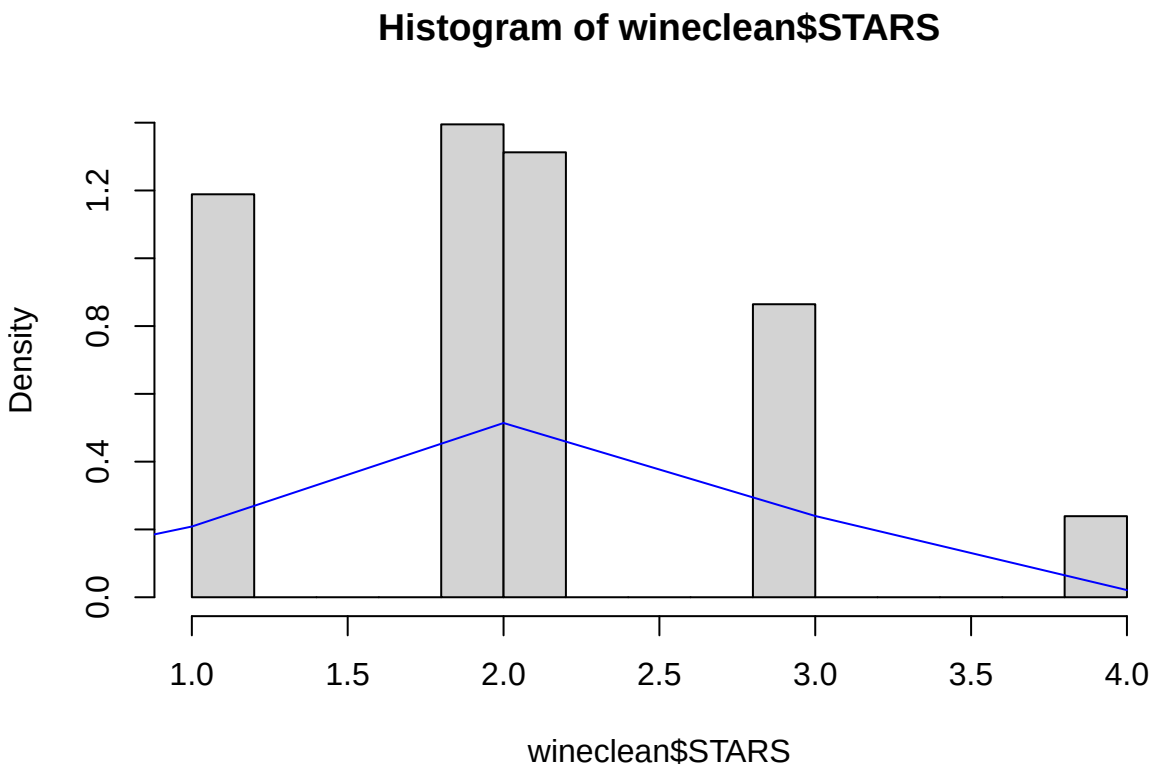
Wine - Alcohol

```
mean = mean(wineclean$Alcohol)
  sd = sd(wineclean$Alcohol)
hist(wineclean$Alcohol, probability = TRUE)
x <- -4.5:27
y <- dnorm(x = x, mean = mean, sd = sd)
lines(x = x, y = y, col = "blue") #The alcohol distribution seems normal!
```

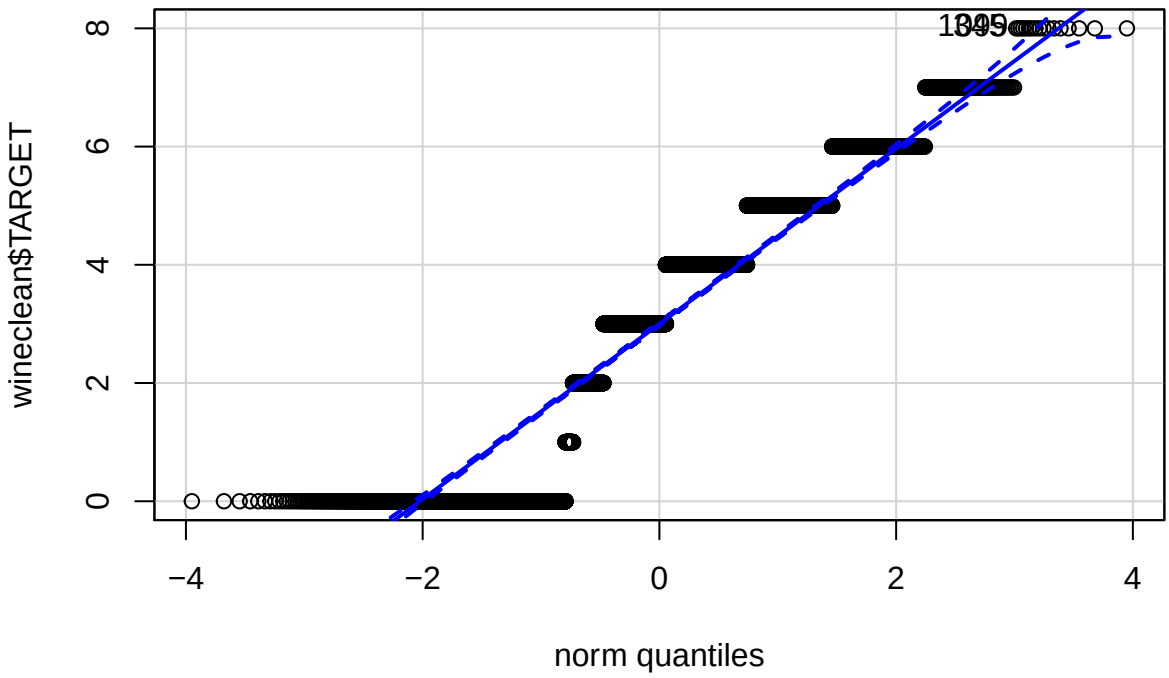


Wine - STARS

```
mean = mean(wineclean$STARS)
  sd = sd(wineclean$STARS)
hist(wineclean$STARS, probability = TRUE)
x <- 0:4
y <- dnorm(x = x, mean = mean, sd = sd)
lines(x = x, y = y, col = "blue") # Doesn't look  normal here!
```

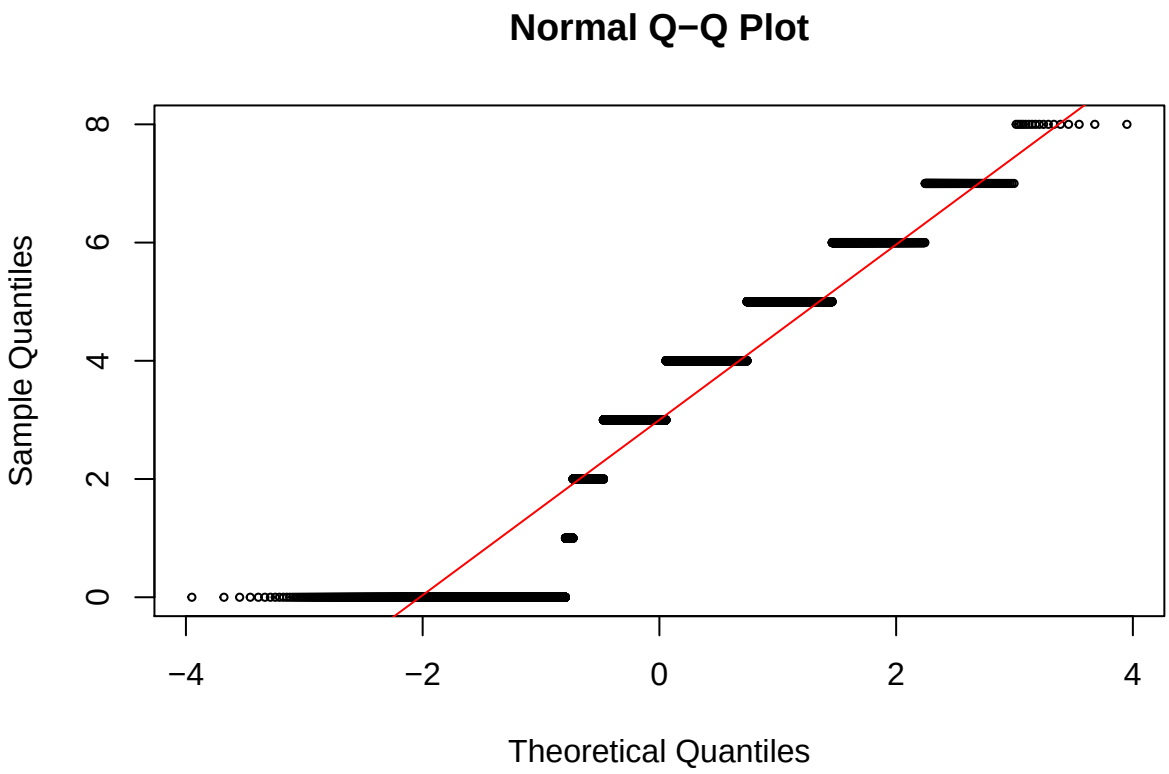



```
qqPlot(wineclean$TARGET)
```

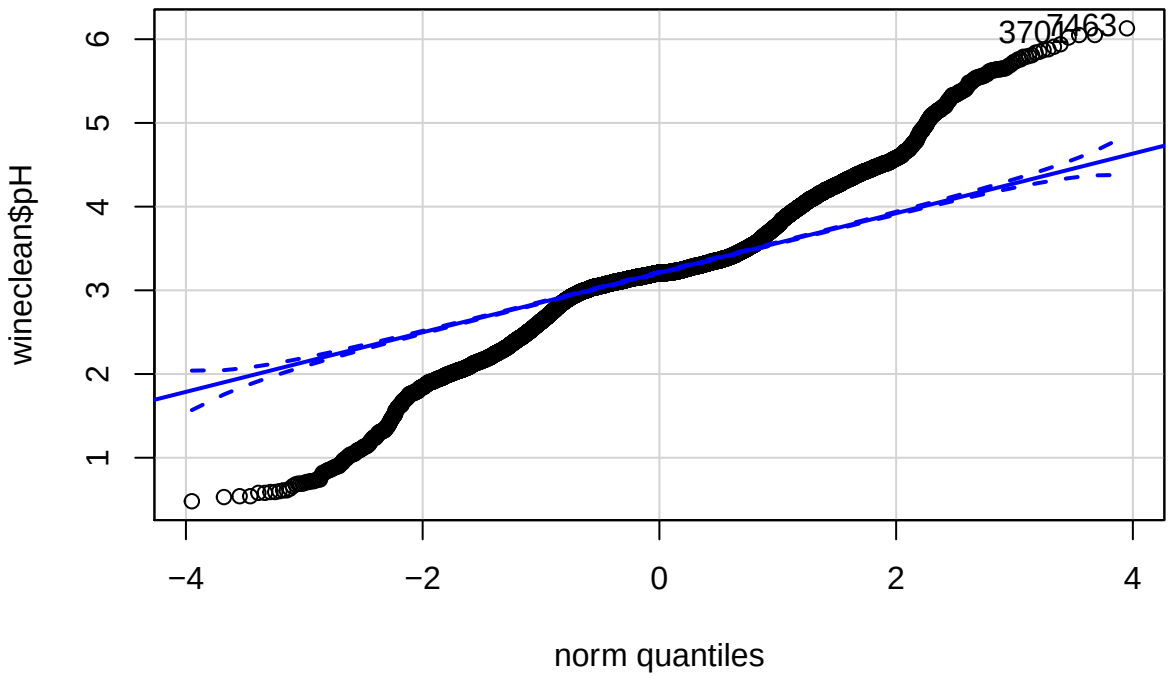



```
## [1]  345 1099
```

```
qqnorm(wineclean$TARGET,pch = 1, cex = 0.5)
qqline(wineclean$TARGET, col = "red", lwd = 1)
```

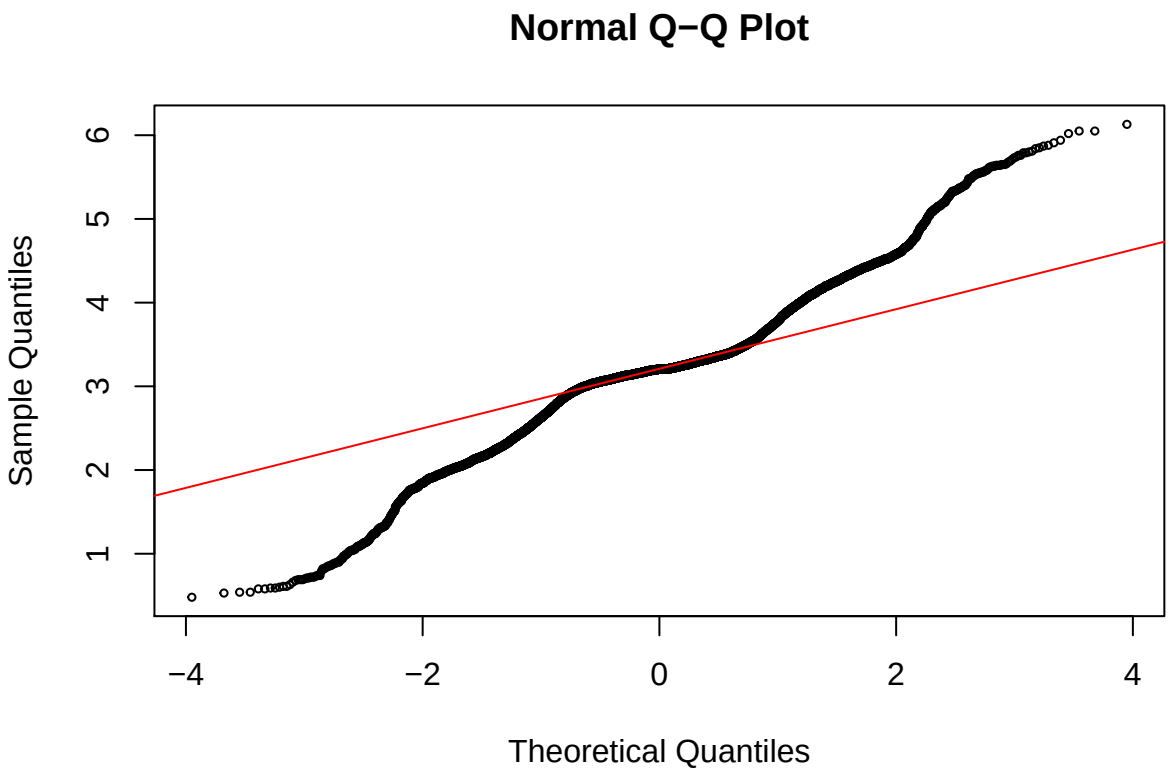



```
qqPlot(wineclean$pH)
```

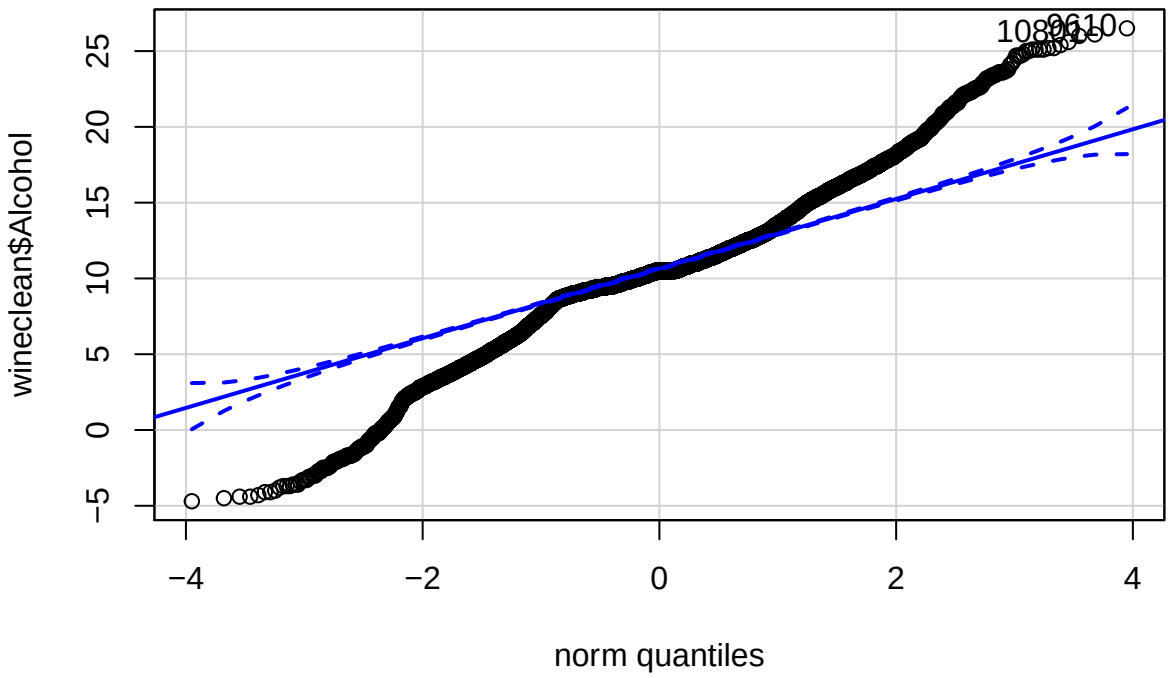



```
## [1] 7463 3701
```

```
qqnorm(wineclean$pH,pch = 1, cex = 0.5)
qqline(wineclean$pH, col = "red", lwd = 1)
```

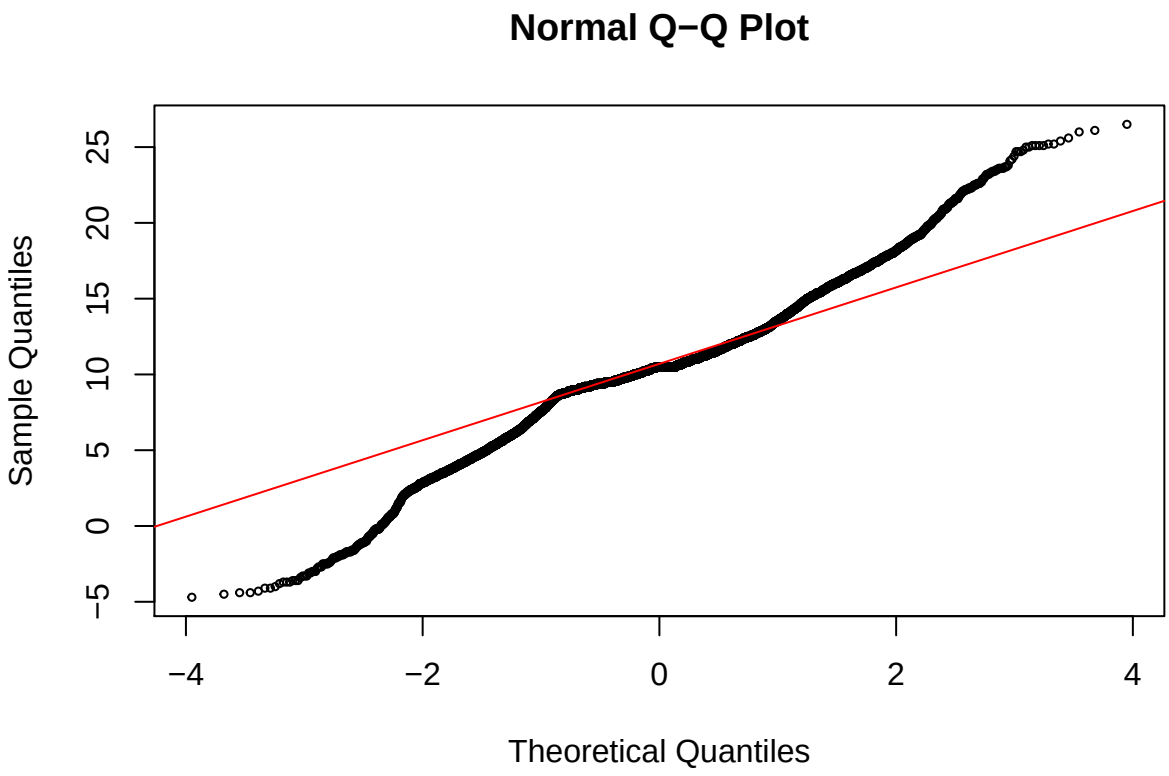



```
qqPlot(wineclean$Alcohol)
```

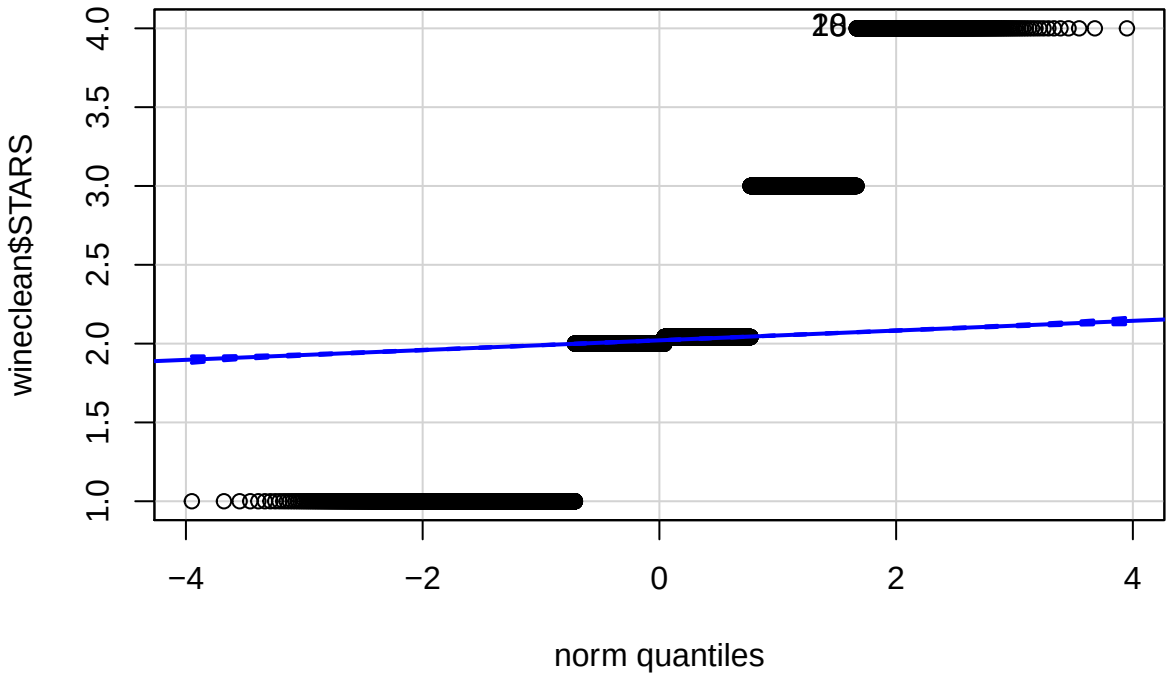



```
## [1]  9610 10801
```

```
qqnorm(wineclean$Alcohol,pch = 1, cex = 0.5)
qqline(wine$Alcohol, col = "red", lwd = 1)
```

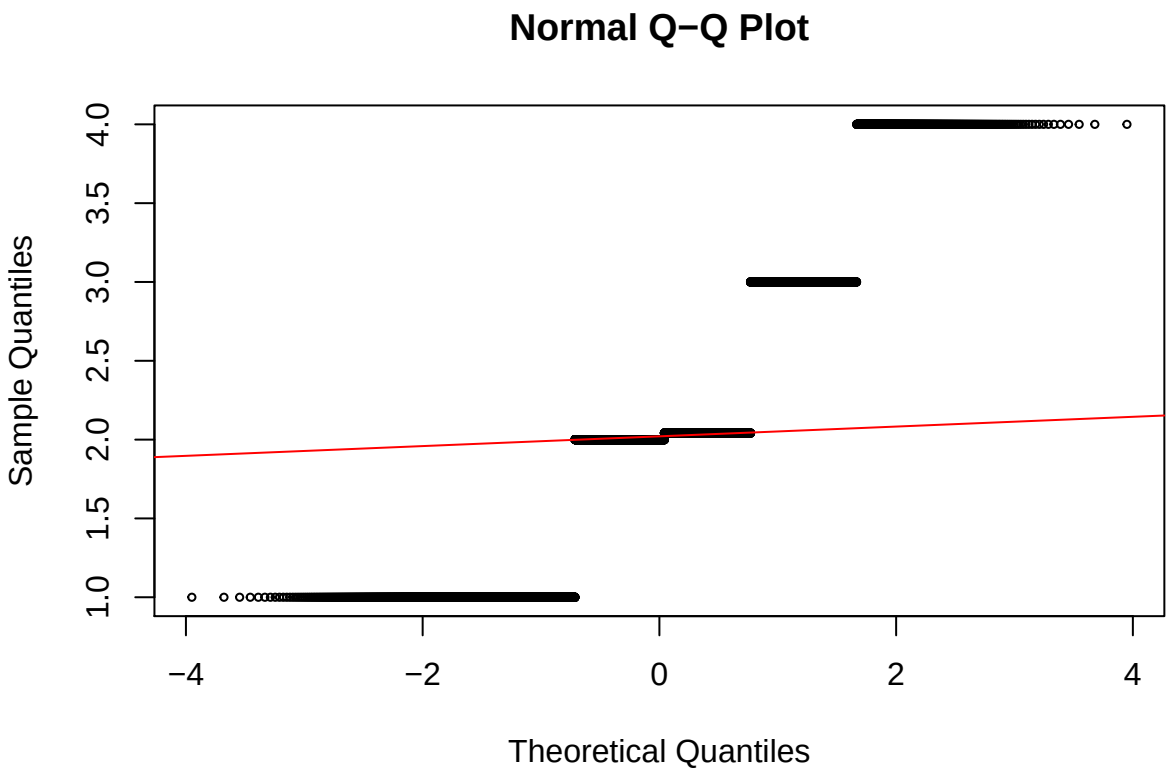



```
qqPlot(wineclean$STARS)
```




```
## [1] 10 28
```

```
qqnorm(wineclean$STARS,pch = 1, cex = 0.5)
qqline(wineclean$STARS, col = "red", lwd = 1)
```



Normality Significance Test

The central limit theorem tells us that no matter what distribution things have, the sampling distribution tends to be normal if the sample is large enough (n > 30).

The Anderson-Darling Test

```
ad.test(wineclean$TARGET)
```

```
##
##  Anderson-Darling normality test
##
## data:  wineclean$TARGET
## A = 482.92, p-value < 2.2e-16
```

```
ad.test(wineclean$pH)
```

```
##
##  Anderson-Darling normality test
##
## data:  wineclean$pH
## A = 252.67, p-value < 2.2e-16
```

```
ad.test(wineclean$Alcohol)
```

```
##
##  Anderson-Darling normality test
##
## data:  wineclean$Alcohol
## A = 176.7, p-value < 2.2e-16
```

```
ad.test(wineclean$STARS)
```

```
##
##  Anderson-Darling normality test
##
## data:  wineclean$STARS
## A = 907.58, p-value < 2.2e-16
```

Since the p-value is less than $\alpha = 0.05$, there is a rare chance that the data came from a normal distribution. The Anderson-Darling test, while having excellent theoretical properties, has a serious flaw when applied to real world data.

The Anderson-Darling test is severely affected by ties in the data due to poor precision. When a significant number of ties exist, the Anderson-Darling will frequently reject the data as non-normal, regardless of how well the data fits the normal distribution.

The Shapiro-Wilks Test

The Shapiro-Wilks test is also affected by ties and appropriate for sample size between 3 to 5,000 which will not be suitable due to our data size.

The Skewness-Kurtosis All Test

The Skewness-Kurtosis All test is not affected by ties and thus the preferred test for our Modeling purpose.

```
skewness(wineclean)
```

```
##             TARGET       FixedAcidity    VolatileAcidity         CitricAcid
##       -0.326339296       -0.022588609        0.020382355       -0.050312939
##      ResidualSugar          Chlorides  FreeSulfurDioxide TotalSulfurDioxide
##       -0.054456489        0.031219228        0.006561857       -0.007379607
##            Density                 pH          Sulphates            Alcohol
##       -0.018695956        0.044993320        0.006213769       -0.031534869
##        LabelAppeal          AcidIndex              STARS
##        0.008430445        1.648689223        0.520872330
```

The skewness here shows that the distribution of the data for each variable are either slightly skewed to the left or negatively skewed. It is skewed to the left because the computed value is negative, and is slightly because the value is close to zero.

```
kurtosis(wineclean)
```

```
##             TARGET       FixedAcidity    VolatileAcidity         CitricAcid
##           2.123086           4.675730           4.832966           4.838696
##      ResidualSugar          Chlorides  FreeSulfurDioxide TotalSulfurDioxide
##           5.132597           5.040740           5.094926           4.938680
##            Density                 pH          Sulphates            Alcohol
##           4.900725           4.795048           5.249855           4.784418
##        LabelAppeal          AcidIndex              STARS
##           2.738136           8.191373           3.129534
```

There are three types of kurtosis: mesokurtic, leptokurtic, and platykurtic. A positive value shows heavy-tails (i.e. a lot of data in the tails). A negative value shows light-tails (i.e. little data in the tails).

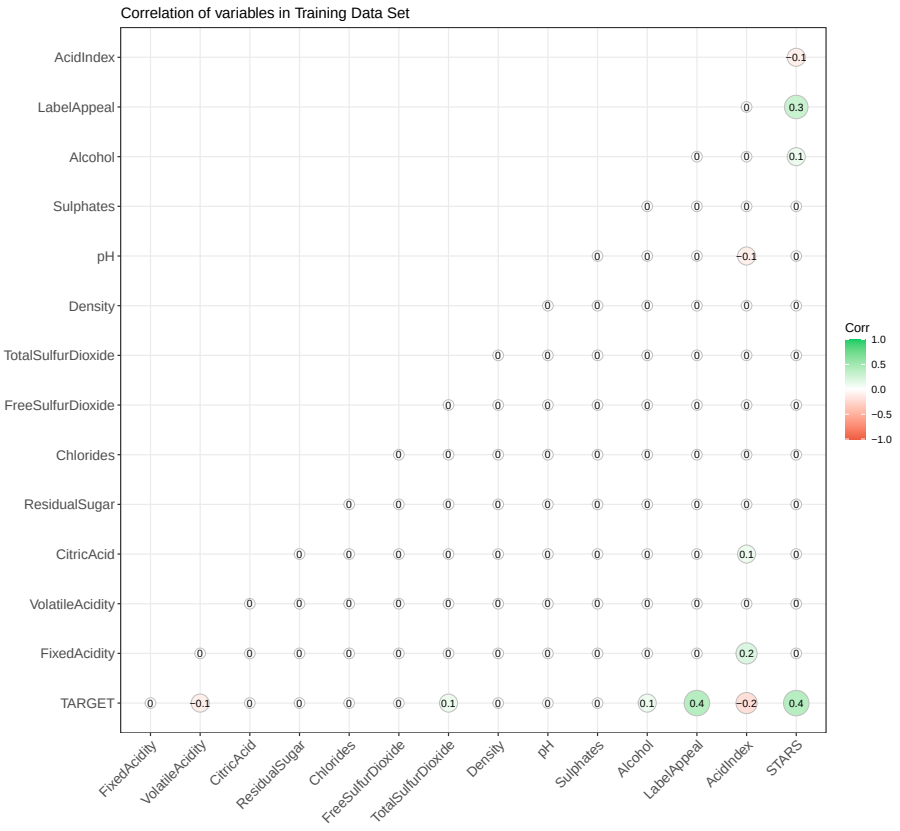
The standard normal distribution has a kurtosis of 3, so if the values are close to 3 then our graph's tails are nearly normal which is known as mesokurtic.

A leptokurtic distribution has excess positive kurtosis, where the kurtosis is greater than 3. The tails are fatter than the normal distribution. These is mostly evident in our data except for LabelAppeal which is 2.7 hence, close to 3.

Correlation and Distribution

The approach below gives the following correlation for these variables

```
# Look at correlation between variables
corr <- round(cor(wineclean), 1)
ggcorrplot(corr,
           type="lower",
           lab=TRUE,
           lab_size=3,
           method="circle",
           colors=c("tomato2", "white", "springgreen3"),
           title="Correlation of variables in Training Data Set",
           ggtheme=theme_bw)
```



There is no strong correlation between the variables to worry about.

Spliting the wine dataset

Use 80% for training and 20% for testing the model

```
set.seed(1234)
train <- createDataPartition(y = wineclean$TARGET, p = 0.80, list = FALSE)
wine_train <- na.omit(wine[train,])
wine_test  <- na.omit(wine[-train,])
```

The Training dataset now has 10,238 observations with 15 variables and the testing has 2557 observations with 15 variables.

Build Models

Using the training data set, build at least twodifferent poissonregression models, at least two different negative binomial regression models, and at least two multiple linear regression models, using different variables (or the same variables with different transformations). Sometimes poisson and negative binomial regression models give the same results.

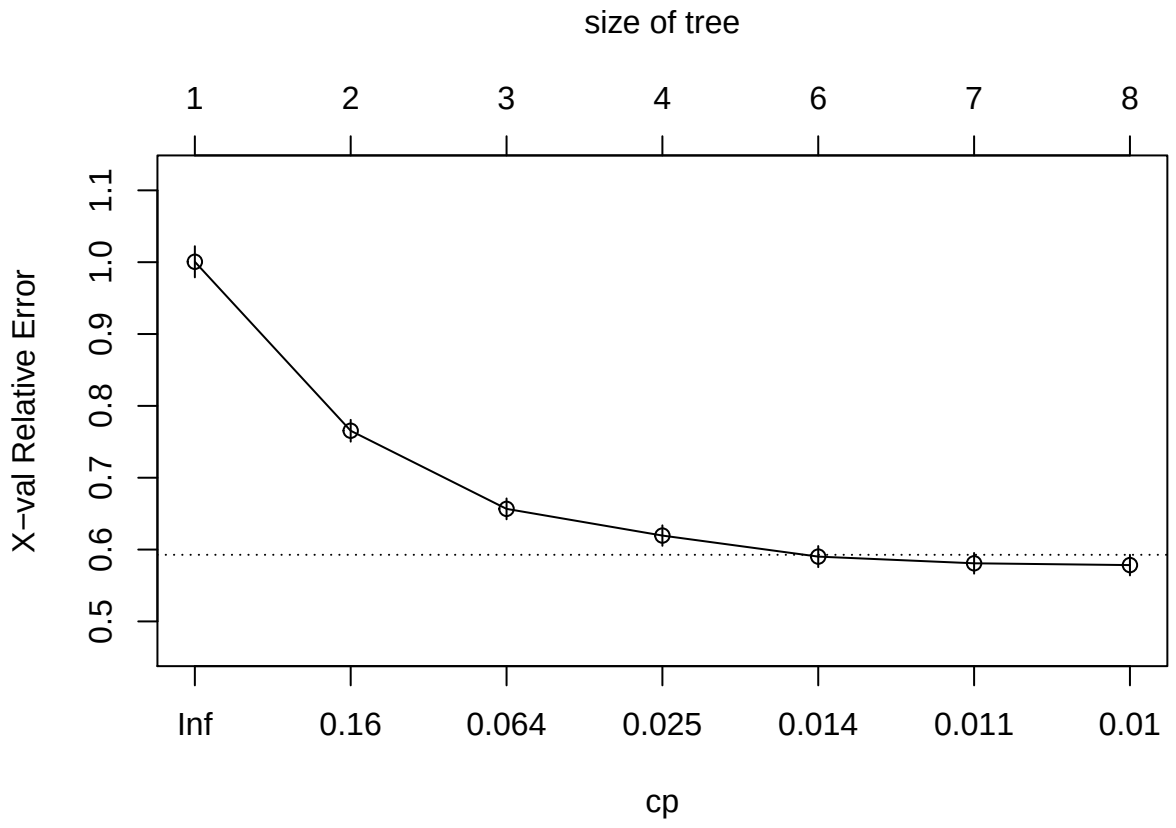
You may want to consider building zero-inflated poisson and negative binomial regression models. You may select the variables manually, use an approach such as Forward or Stepwise, use a different approach such as trees, or use a combination of techniques.

Modeling Method - Regression Tree

```
winetree <- rpart(TARGET ~., method="anova", data=wine_train)
printcp(winetree) # display the results
```

```
##
## Regression tree:
## rpart(formula = TARGET ~ ., data = wine_train, method = "anova")
##
## Variables actually used in tree construction:
## [1] AcidIndex   LabelAppeal STARS
##
## Root node error: 12473/5161 = 2.4167
##
## n= 5161
##
##         CP nsplit rel error  xerror     xstd
## 1 0.235374      0   1.00000 1.00058 0.021679
## 2 0.108577      1   0.76463 0.76540 0.015170
## 3 0.037243      2   0.65605 0.65665 0.014523
## 4 0.016619      3   0.61881 0.61948 0.014238
## 5 0.011551      5   0.58557 0.59022 0.014679
## 6 0.010306      6   0.57402 0.58089 0.014384
## 7 0.010000      7   0.56371 0.57841 0.014344
```

```
plotcp(winetree) # visualize cross-validation results
```




```
summary(winetree) # detailed summary of splits
```

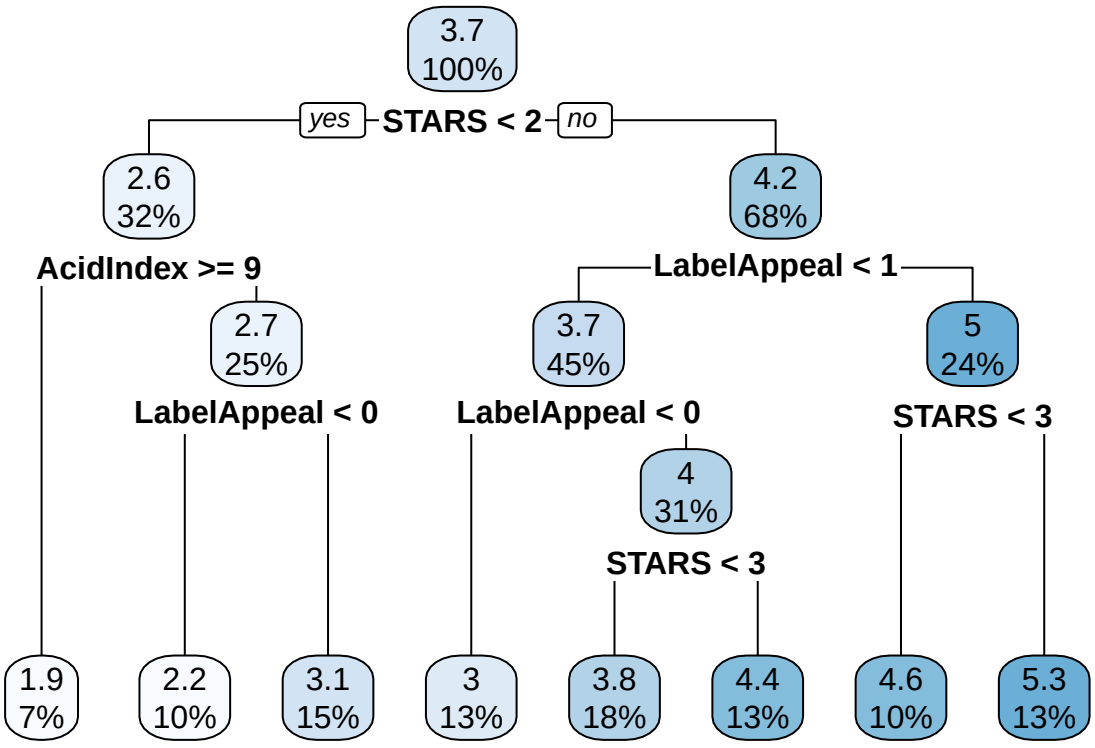
```
## Call:
## rpart(formula = TARGET ~ ., data = wine_train, method = "anova")
##   n= 5161
##
##           CP nsplit rel error    xerror       xstd
## 1 0.23537389      0 1.0000000 1.0005768 0.02167916
## 2 0.10857717      1 0.7646261 0.7653955 0.01517014
## 3 0.03724285      2 0.6560489 0.6566482 0.01452326
## 4 0.01661855      3 0.6188061 0.6194781 0.01423781
## 5 0.01155136      5 0.5855690 0.5902192 0.01467937
## 6 0.01030559      6 0.5740176 0.5808911 0.01438352
## 7 0.01000000      7 0.5637120 0.5784057 0.01434364
##
## Variable importance
##       STARS LabelAppeal   AcidIndex
##          57          37           4
##
## Node number 1: 5161 observations,    complexity param=0.2353739
##   mean=3.669056, MSE=2.416731
##   left son=2 (1643 obs) right son=3 (3518 obs)
##   Primary splits:
##       STARS       < 1.5      to the left,  improve=0.23537390, (0 missing)
##       LabelAppeal < 0.5      to the left,  improve=0.16091720, (0 missing)
```

```
##       AcidIndex   < 9.5      to the right, improve=0.02283996, (0 missing)
##       Alcohol     < 10.01667 to the left,  improve=0.01256173, (0 missing)
##       Density     < 0.992845 to the right, improve=0.01240742, (0 missing)
##   Surrogate splits:
##       LabelAppeal       < -1.5     to the left,  agree=0.693, adj=0.037, (0 split)
##       AcidIndex         < 13.5     to the right, agree=0.684, adj=0.006, (0 split)
##       FreeSulfurDioxide < -506.5   to the left,  agree=0.682, adj=0.002, (0 split)
##       Density           < 1.09023  to the right, agree=0.682, adj=0.002, (0 split)
##       FixedAcidity      < -17.6    to the left,  agree=0.682, adj=0.001, (0 split)
##
## Node number 2: 1643 observations,    complexity param=0.01661855
##   mean=2.565429, MSE=2.623078
##   left son=4 (336 obs) right son=5 (1307 obs)
##   Primary splits:
##       AcidIndex          < 8.5      to the right, improve=0.04671870, (0 missing)
##       LabelAppeal        < -0.5     to the left,  improve=0.04155929, (0 missing)
##       VolatileAcidity    < 0.3675   to the right, improve=0.02036090, (0 missing)
##       Chlorides          < 0.0635   to the right, improve=0.01882068, (0 missing)
##       TotalSulfurDioxide < 97.5     to the left,  improve=0.01391331, (0 missing)
##   Surrogate splits:
##       Chlorides   < 1.2065   to the right, agree=0.797, adj=0.006, (0 split)
##       LabelAppeal < 1.5      to the right, agree=0.797, adj=0.006, (0 split)
##       CitricAcid  < 3.395    to the right, agree=0.796, adj=0.003, (0 split)
##
## Node number 3: 3518 observations,    complexity param=0.1085772
##   mean=4.18448, MSE=1.485865
##   left son=6 (2303 obs) right son=7 (1215 obs)
##   Primary splits:
##       LabelAppeal < 0.5      to the left,  improve=0.25907500, (0 missing)
##       STARS       < 2.5      to the left,  improve=0.13707280, (0 missing)
##       Alcohol     < 10.11667 to the left,  improve=0.02661829, (0 missing)
##       Chlorides   < 0.0395   to the right, improve=0.01544906, (0 missing)
##       Density     < 0.992815 to the right, improve=0.01517274, (0 missing)
##   Surrogate splits:
##       STARS             < 3.5      to the left,  agree=0.673, adj=0.053, (0 split)
##       CitricAcid        < 3.355    to the left,  agree=0.657, adj=0.006, (0 split)
##       VolatileAcidity   < -2.555   to the right, agree=0.656, adj=0.004, (0 split)
##       FreeSulfurDioxide < -509     to the right, agree=0.655, adj=0.002, (0 split)
##       ResidualSugar     < -119.4   to the right, agree=0.655, adj=0.002, (0 split)
##
## Node number 4: 336 observations
##   mean=1.875, MSE=2.901042
##
## Node number 5: 1307 observations,    complexity param=0.01661855
##   mean=2.742923, MSE=2.397569
##   left son=10 (514 obs) right son=11 (793 obs)
##   Primary splits:
##       LabelAppeal        < -0.5     to the left,  improve=0.068040630, (0 missing)
##       VolatileAcidity    < 0.3425   to the right, improve=0.018925450, (0 missing)
##       FreeSulfurDioxide  < 18.5     to the left,  improve=0.017633050, (0 missing)
##       TotalSulfurDioxide < 102.5    to the left,  improve=0.008879143, (0 missing)
##       Chlorides          < 0.0585   to the right, improve=0.007139648, (0 missing)
##   Surrogate splits:
##       TotalSulfurDioxide < 616.5    to the right, agree=0.610, adj=0.008, (0 split)
```

```
##       Sulphates          < 2.33     to the right, agree=0.610, adj=0.008, (0 split)
##       ResidualSugar      < 74.7     to the right, agree=0.609, adj=0.006, (0 split)
##       Alcohol            < 18.95    to the right, agree=0.608, adj=0.004, (0 split)
##       FixedAcidity       < 15.95    to the right, agree=0.607, adj=0.002, (0 split)
##
## Node number 6: 2303 observations,    complexity param=0.03724285
##   mean=3.733825, MSE=1.09589
##   left son=12 (689 obs) right son=13 (1614 obs)
##   Primary splits:
##       LabelAppeal < -0.5     to the left,  improve=0.18405350, (0 missing)
##       STARS       < 2.5      to the left,  improve=0.10884050, (0 missing)
##       Alcohol     < 10.65    to the left,  improve=0.03126687, (0 missing)
##       Density     < 0.992605 to the right, improve=0.03038305, (0 missing)
##       Chlorides   < 0.0395   to the right, improve=0.02360232, (0 missing)
##   Surrogate splits:
##       VolatileAcidity   < 2.63     to the right, agree=0.703, adj=0.006, (0 split)
##       Density           < 0.89588  to the left,  agree=0.702, adj=0.004, (0 split)
##       ResidualSugar     < 123.05   to the right, agree=0.702, adj=0.003, (0 split)
##       FreeSulfurDioxide < 586.5    to the right, agree=0.702, adj=0.003, (0 split)
##       FixedAcidity      < 29.3     to the right, agree=0.701, adj=0.001, (0 split)
##
## Node number 7: 1215 observations,    complexity param=0.01155136
##   mean=5.038683, MSE=1.110438
##   left son=14 (533 obs) right son=15 (682 obs)
##   Primary splits:
##       STARS           < 2.5      to the left,  improve=0.10678850, (0 missing)
##       LabelAppeal     < 1.5      to the left,  improve=0.10330040, (0 missing)
##       Alcohol         < 10.85    to the left,  improve=0.04642333, (0 missing)
##       Chlorides       < 0.0425   to the right, improve=0.02506496, (0 missing)
##       VolatileAcidity < 0.5775   to the right, improve=0.01019747, (0 missing)
##   Surrogate splits:
##       VolatileAcidity < 0.425    to the right, agree=0.580, adj=0.043, (0 split)
##       Alcohol         < 5.45     to the left,  agree=0.577, adj=0.036, (0 split)
##       AcidIndex       < 8.5      to the right, agree=0.573, adj=0.026, (0 split)
##       Sulphates       < 2.265    to the right, agree=0.570, adj=0.019, (0 split)
##       ResidualSugar   < 48.25    to the right, agree=0.569, adj=0.017, (0 split)
##
## Node number 10: 514 observations
##   mean=2.241245, MSE=1.311451
##
## Node number 11: 793 observations
##   mean=3.068096, MSE=2.83269
##
## Node number 12: 689 observations
##   mean=3.046444, MSE=0.9180171
##
## Node number 13: 1614 observations,    complexity param=0.01030559
##   mean=4.027261, MSE=0.8840152
##   left son=26 (940 obs) right son=27 (674 obs)
##   Primary splits:
##       STARS      < 2.5      to the left,  improve=0.090088990, (0 missing)
##       Alcohol    < 10.93333 to the left,  improve=0.042069700, (0 missing)
##       Density    < 0.992815 to the right, improve=0.038968650, (0 missing)
##       Chlorides  < 0.0395   to the right, improve=0.018321190, (0 missing)
```

```
##       CitricAcid < 0.265    to the left,  improve=0.009464323, (0 missing)
##   Surrogate splits:
##       pH                 < 4.375    to the left,  agree=0.588, adj=0.013, (0 split)
##       FixedAcidity       < 18.95    to the left,  agree=0.586, adj=0.007, (0 split)
##       TotalSulfurDioxide < -606     to the right, agree=0.586, adj=0.007, (0 split)
##       ResidualSugar      < 72.9     to the left,  agree=0.585, adj=0.006, (0 split)
##       Sulphates          < -2.725   to the right, agree=0.585, adj=0.006, (0 split)
##
## Node number 14: 533 observations
##   mean=4.649156, MSE=1.147077
##
## Node number 15: 682 observations
##   mean=5.343109, MSE=0.8705463
##
## Node number 26: 940 observations
##   mean=3.788298, MSE=0.8796503
##
## Node number 27: 674 observations
##   mean=4.360534, MSE=0.699392
```

```
rpart.plot(winetree, extra = "auto",fallen.leaves = TRUE, box.palette = "auto")
```



The Regression Tree shows that 3 variables STARS, LabelAppeal and AcidIndex are important for prdicting the Target variable which is the Number of Cases Purchased. We will not dig much into the Regression Tree as the goal was to help us identify relevant variables for the Count and Multilinear Regression Models.

Model Method - Poisson

Poisson Model 1 - All Variables

Using all the fifteen (15) variables for the Poisson Model provided an AIC value of 18608.

We will consider these variables in the next model.

```
wine_poisson1 <- glm(TARGET ~., data = wine_train, family = poisson(link="log"))
summary(wine_poisson1)
```

```
##
## Call:
## glm(formula = TARGET ~ ., family = poisson(link = "log"), data = wine_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.11509  -0.27575   0.06379   0.37590   1.69260
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.616e+00  2.799e-01   5.775 7.72e-09 ***
## FixedAcidity         6.457e-04  1.189e-03   0.543  0.58699
## VolatileAcidity     -2.688e-02  9.353e-03  -2.874  0.00405 **
## CitricAcid           4.657e-04  8.472e-03   0.055  0.95616
## ResidualSugar       -7.046e-05  2.159e-04  -0.326  0.74418
## Chlorides           -3.107e-02  2.300e-02  -1.351  0.17676
## FreeSulfurDioxide    7.730e-05  4.889e-05   1.581  0.11386
## TotalSulfurDioxide   2.742e-05  3.158e-05   0.868  0.38514
## Density             -3.768e-01  2.740e-01  -1.375  0.16898
## pH                  -6.337e-03  1.074e-02  -0.590  0.55501
## Sulphates           -7.269e-03  7.880e-03  -0.922  0.35627
## Alcohol              3.452e-03  1.958e-03   1.763  0.07788 .
## LabelAppeal          1.799e-01  8.805e-03  20.427  < 2e-16 ***
## AcidIndex           -5.054e-02  6.662e-03  -7.587 3.28e-14 ***
## STARS                1.888e-01  8.360e-03  22.583  < 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: 4742.1  on 5160  degrees of freedom
## Residual deviance: 3249.6  on 5146  degrees of freedom
## AIC: 18608
##
## Number of Fisher Scoring iterations: 5
```

The AIC value is high, the Model also indicated that four (4) variables are significant in predicting the number of wine cases purchased. The three variables have p-values less than 0.05, the variabes are VolatileAcidity, LabelAppeal, AcidIndex, VolatileAcidity and STARS.

Deviance - wine_poisson1

```
deviance1 <- glance(wine_poisson1)
deviance1
```

```
## # A tibble: 1 x 8
##   null.deviance df.null logLik    AIC    BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl>  <dbl>  <dbl>    <dbl>       <int> <int>
## 1         4742.    5160 -9289. 18608. 18706.    3250.        5146  5161
```

The deviance of the wine_poisson1 model (All Variables) is 3249.562 and the BIC is 18706.19.

Pseudo R Square - wine_poisson1

The Null deviance shows how well the response variable is predicted by a model that includes only the intercept (grand mean) whereas residual with the inclusion of independent variables. Above, we can see that the addition of 14 (5160-5146 = 14) independent variables decreased the deviance from 4742.1 to 3249.6.

Greater difference in values means a bad fit.

Null Deviance - Residual Deviance / Null Deviance

```
((wine_poisson1$null.deviance-wine_poisson1$deviance)/wine_poisson1$null.deviance)*100
```

```
## [1] 31.47414
```

31.5% of Total variability in the data was explained by this Model

Overdispersion - wine_poisson1

In Poisson regression, it is very important to check for overdispersion since the variance and means are equal.

The potential problem with Poisson GLMs is overdispersion which means that the variance is larger than the mean. Overdispersion occurs when the observed variance of the response variable is larger than would be predicted by the Poisson distribution.

Method - Residual Deviance / Degrees of freedom.

A model is overdispersed when the value is greater than 1.

```
wine_poisson1$deviance/wine_poisson1$df.residual
```

```
## [1] 0.6314733
```

This Model is not over-dispersed at 0.6314733.

Poisson Model 2 - Four Variables

```
wine_poisson2 <- glm(TARGET ~ VolatileAcidity + LabelAppeal + AcidIndex + STARS, data = wine_train, fami
summary(wine_poisson2)
```

```
##
## Call:
## glm(formula = TARGET ~ VolatileAcidity + LabelAppeal + AcidIndex +
##     STARS, family = poisson(link = "log"), data = wine_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
```

```
## -3.13313  -0.27938   0.06009   0.37871   1.65392
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.264052   0.054995  22.985  < 2e-16 ***
## VolatileAcidity -0.027164   0.009350  -2.905  0.00367 **
## LabelAppeal       0.179694   0.008802  20.415  < 2e-16 ***
## AcidIndex        -0.051053   0.006550  -7.794 6.48e-15 ***
## STARS             0.190259   0.008333  22.833  < 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: 4742.1  on 5160  degrees of freedom
## Residual deviance: 3261.4  on 5156  degrees of freedom
## AIC: 18600
##
## Number of Fisher Scoring iterations: 5
```

The AIC is at 18600 and converged after the fifth iteration.

Deviance - wine_poisson2

```
deviance2 <- glance(wine_poisson2)
deviance2
```

```
## # A tibble: 1 x 8
##   null.deviance df.null logLik    AIC    BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl>  <dbl>  <dbl>    <dbl>       <int> <int>
## 1         4742.    5160 -9295. 18600. 18633.    3261.        5156  5161
```

The deviance of the wine_poisson2 model (All Variables) is 3249.562 and the BIC is 18632.5.

Pseudo R Square - wine_poisson2

Null Deviance - residual Deviance / Null Deviance

```
((wine_poisson2$null.deviance-wine_poisson2$deviance)/wine_poisson2$null.deviance)*100
```

```
## [1] 31.22522
```

31.2% of Total variability in the data was explained by this Model

Overdispersion - wine_poisson2

```
wine_poisson2$deviance/wine_poisson2$df.residual
```

```
## [1] 0.6325379
```

This Model is not over-dispersed at 0.6325379.

Poisson Model 3 - Three Variables

```
wine_poisson3 <- glm(TARGET ~ LabelAppeal + AcidIndex + STARS, data = wine_train, family = poisson(link=
summary(wine_poisson3)
```

```
##
## Call:
## glm(formula = TARGET ~ LabelAppeal + AcidIndex + STARS, family = poisson(link = "log"),
##     data = wine_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.12824  -0.27612   0.03793   0.39077   1.65607
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.256539   0.054903  22.887  < 2e-16 ***
## LabelAppeal  0.180044   0.008796  20.469  < 2e-16 ***
## AcidIndex   -0.051326   0.006544  -7.843 4.39e-15 ***
## STARS        0.190889   0.008329  22.920  < 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: 4742.1  on 5160  degrees of freedom
## Residual deviance: 3269.8  on 5157  degrees of freedom
## AIC: 18606
##
## Number of Fisher Scoring iterations: 5
```

The AIC is at 18606 and converged after the fifth iteration.

Deviance - wine_poisson3

```
deviance3 <- glance(wine_poisson3)
deviance3
```

```
## # A tibble: 1 x 8
##   null.deviance df.null logLik    AIC    BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl>  <dbl>  <dbl>    <dbl>       <int> <int>
## 1         4742.    5160 -9299. 18606. 18632.    3270.        5157  5161
```

The deviance of the wine_poisson3 model (All Variables) is 3269.808 and the BIC is 18632.39.

Pseudo R Square - wine_poisson1

Null Deviance - residual Deviance / Null Deviance

```
((wine_poisson3$null.deviance-wine_poisson3$deviance)/wine_poisson3$null.deviance)*100
```

```
## [1] 31.04719
```

31.04% of Total variability in the data was explained by this Model

Overdispersion - wine_poisson3

```
wine_poisson3$deviance/wine_poisson3$df.residual
```

```
## [1] 0.6340523
```

This Model is not overdispersed at 0.6340523.

Model Method - Negative Binomial

Negative Binomial - All Variables

```
wine_NB1 <- glm.nb(TARGET ~., data = wine_train)
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
summary(wine_NB1)
```

```
##
## Call:
## glm.nb(formula = TARGET ~ ., data = wine_train, init.theta = 137917.6811,
##     link = log)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.11507  -0.27574   0.06379   0.37590   1.69258
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.616e+00  2.799e-01   5.774 7.72e-09 ***
## FixedAcidity         6.457e-04  1.189e-03   0.543  0.58699
## VolatileAcidity     -2.688e-02  9.353e-03  -2.874  0.00405 **
## CitricAcid           4.657e-04  8.472e-03   0.055  0.95617
## ResidualSugar       -7.045e-05  2.159e-04  -0.326  0.74419
## Chlorides           -3.107e-02  2.300e-02  -1.351  0.17676
## FreeSulfurDioxide    7.730e-05  4.889e-05   1.581  0.11386
## TotalSulfurDioxide   2.742e-05  3.158e-05   0.868  0.38514
## Density             -3.768e-01  2.740e-01  -1.375  0.16899
## pH                  -6.337e-03  1.074e-02  -0.590  0.55501
## Sulphates           -7.269e-03  7.880e-03  -0.922  0.35628
## Alcohol              3.452e-03  1.958e-03   1.763  0.07788 .
## LabelAppeal          1.799e-01  8.805e-03  20.427  < 2e-16 ***
## AcidIndex           -5.054e-02  6.662e-03  -7.587 3.28e-14 ***
## STARS                1.888e-01  8.360e-03  22.582  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(137917.7) family taken to be 1)
##
```

```
##     Null deviance: 4742.0  on 5160  degrees of freedom
## Residual deviance: 3249.5  on 5146  degrees of freedom
## AIC: 18610
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  137918
##           Std. Err.:  256832
## Warning while fitting theta: iteration limit reached
##
##  2 x log-likelihood:  -18578.04
```

The Negative Binomial Model did not fully converge to Theta as can be seen above, this is mostly due to the data undispersed relative to the Negative Binomial distribution.

It appears the distribution of our data is not very suitable for this type of Model.

We will investigate further with a Zero Inflated Negative Binomial Model to determine if it makes sense to consider this Modeling for our analysis.

Alternative to Negative Binomial - Three variables

The function below will develop a Negative Binomial Model using the three significant variables, if it fails to converge, the Function will automatically build a Poisson Regression Model using the Three variables then calculate the coefficients.

```
ModelGLM <- function() {
  wineGlm <- glm.nb(TARGET ~ LabelAppeal + AcidIndex + STARS, data = wine_train)
  if (wineGlm$th.warn == "iteration limit reached") {
    wineGlm <- glm(TARGET ~ LabelAppeal + AcidIndex + STARS, data = wine_train, family = poisson)
  }
  wineGlm
}
ModelPoisson <- ModelGLM()
```

```
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached

## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =
## control$trace > : iteration limit reached
```

```
summary(ModelPoisson)$coefficients
```

```
##                Estimate  Std. Error   z value      Pr(>|z|)
## (Intercept)  1.25653882 0.054902594 22.886693 6.304520e-116
## LabelAppeal  0.18004370 0.008795916 20.469010  4.067963e-93
## AcidIndex   -0.05132631 0.006544122 -7.843117  4.394981e-15
## STARS        0.19088891 0.008328531 22.919878 2.944038e-116
```

The ModelPoisson produced using this function is the same as the Poisson Model 3 known as wine_poisson3 above which used the three significant variables previously identified.

Model Method - Zero Inflated Model

Zero Inflated Negative Binomial Model - Three variables

```
wineZeroInfNB <- zeroinfl(TARGET ~ LabelAppeal + AcidIndex + STARS, data = wine_train, dist = "negbin")
summary(wineZeroInfNB)
```

```
##
## Call:
## zeroinfl(formula = TARGET ~ LabelAppeal + AcidIndex + STARS, data = wine_train,
##     dist = "negbin")
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -2.26486 -0.30888  0.02207  0.35595  5.13505
##
## Count model coefficients (negbin with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.210469   0.056958  21.252   <2e-16 ***
## LabelAppeal  0.213804   0.009011  23.728   <2e-16 ***
## AcidIndex   -0.017232   0.006890  -2.501   0.0124 *
## STARS        0.114322   0.008813  12.972   <2e-16 ***
## Log(theta)  16.156801   6.724217   2.403   0.0163 *
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.4424     0.7141  -2.020   0.0434 *
## LabelAppeal   0.7390     0.1014   7.284 3.23e-13 ***
## AcidIndex     0.4981     0.0555   8.974  < 2e-16 ***
## STARS        -4.1158     0.5404  -7.616 2.61e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 10394636.8566
## Number of iterations in BFGS optimization: 22
## Log-likelihood: -8946 on 9 Df
```

The Zero Inflated Negative Binomial Model converged with Theta = 10394636.8566 at the 22nd iteration. Theta controls the excess variability compared to Poisson, although this is better than the ordinary Negative Binomial Regression Model above, the variability is too high as can be seen from the output above.

We can also see that p-value for produced for the Log Theta is only slightly significant at 0.016. Therefore, we will continue to focus on the Poisson Regression and compare it to the Multilinear Regression for this task.

Zero Inflated Poisson Regression Model - Three variables

Zero-inflated poisson regression is used to model count data that has an excess of zero counts. Further, theory suggests that the excess zeros are generated by a separate process from the count values and that the excess zeros can be modeled independently.

```
ZeroInfPoisson <- zeroinfl(TARGET ~ LabelAppeal + AcidIndex + STARS, data = wine_train)
summary(ZeroInfPoisson)
```

```
##
## Call:
## zeroinfl(formula = TARGET ~ LabelAppeal + AcidIndex + STARS, data = wine_train)
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -2.26502 -0.30892  0.02206  0.35597  5.14594
##
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.210340   0.056958  21.250   <2e-16 ***
## LabelAppeal  0.213802   0.009011  23.728   <2e-16 ***
## AcidIndex   -0.017218   0.006890  -2.499   0.0125 *
## STARS        0.114330   0.008813  12.972   <2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.44638    0.71480  -2.023    0.043 *
## LabelAppeal  0.73939    0.10149   7.285 3.22e-13 ***
## AcidIndex    0.49880    0.05553   8.982  < 2e-16 ***
## STARS       -4.11776    0.54128  -7.607 2.80e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 15
## Log-likelihood: -8946 on 8 Df
```

All of the predictors in both the count and inflation portions of the model are statistically significant except AcidIndex which is only slightly significant at 0.012.

This model fits the data but the model output above does not indicate if our zero-inflated model is an improvement over the ordinary Poisson Regression Model.

Let's investigate and compare the Models!

Comparing the Poisson Model to the Zero Inflated

Using the Vuong Non-Nested Hypothesis Test-Statistic to compare models.

```
vuong(ModelPoisson, ZeroInfPoisson)
```

```
## Vuong Non-Nested Hypothesis Test-Statistic:
## (test-statistic is asymptotically distributed N(0,1) under the
##  null that the models are indistinguishible)
## -------------------------------------------------------------
##                Vuong z-statistic             H_A    p-value
## Raw                   -11.53631 model2 > model1 < 2.22e-16
## AIC-corrected         -11.40545 model2 > model1 < 2.22e-16
## BIC-corrected         -10.97695 model2 > model1 < 2.22e-16
```

The Vuong test compares the zero-inflated model with the Poisson Regression Model. We can see that the three test statistic are significant, indicating that the zero-inflated model is superior to the Poisson Model.

Model Method - Multilinear Regression

Multilinear Regression Model - Stepwise All variables

```
wine_OLS1 <-step(lm(TARGET ~., data = wine_train), direction = "both")
```

```
## Start:  AIC=1537.33
## TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid + ResidualSugar +
##     Chlorides + FreeSulfurDioxide + TotalSulfurDioxide + Density +
##     pH + Sulphates + Alcohol + LabelAppeal + AcidIndex + STARS
##
##                       Df Sum of Sq    RSS    AIC
## - CitricAcid           1      0.07 6911.6 1535.4
## - ResidualSugar        1      0.48 6912.0 1535.7
## - pH                   1      0.72 6912.3 1535.9
## - FixedAcidity         1      1.49 6913.0 1536.4
## - Sulphates            1      2.60 6914.1 1537.3
## <none>                             6911.5 1537.3
## - TotalSulfurDioxide   1      3.20 6914.7 1537.7
## - Density              1      5.94 6917.5 1539.8
## - FreeSulfurDioxide    1      7.54 6919.1 1541.0
## - Chlorides            1      7.97 6919.5 1541.3
## - Alcohol              1     15.57 6927.1 1546.9
## - VolatileAcidity      1     31.60 6943.1 1558.9
## - AcidIndex            1    196.70 7108.2 1680.2
## - LabelAppeal          1   1525.97 8437.5 2564.9
## - STARS                1   1960.58 8872.1 2824.1
##
## Step:  AIC=1535.39
## TARGET ~ FixedAcidity + VolatileAcidity + ResidualSugar + Chlorides +
##     FreeSulfurDioxide + TotalSulfurDioxide + Density + pH + Sulphates +
##     Alcohol + LabelAppeal + AcidIndex + STARS
##
##                       Df Sum of Sq    RSS    AIC
## - ResidualSugar        1      0.47 6912.1 1533.7
## - pH                   1      0.72 6912.3 1533.9
## - FixedAcidity         1      1.48 6913.1 1534.5
## - Sulphates            1      2.59 6914.2 1535.3
## <none>                             6911.6 1535.4
## - TotalSulfurDioxide   1      3.20 6914.8 1535.8
## + CitricAcid           1      0.07 6911.5 1537.3
## - Density              1      5.91 6917.5 1537.8
## - FreeSulfurDioxide    1      7.52 6919.1 1539.0
## - Chlorides            1      7.93 6919.5 1539.3
## - Alcohol              1     15.54 6927.2 1545.0
## - VolatileAcidity      1     31.56 6943.2 1556.9
## - AcidIndex            1    197.54 7109.2 1678.8
## - LabelAppeal          1   1525.91 8437.5 2562.9
## - STARS                1   1960.51 8872.1 2822.2
##
## Step:  AIC=1533.74
## TARGET ~ FixedAcidity + VolatileAcidity + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + Density + pH + Sulphates + Alcohol +
##     LabelAppeal + AcidIndex + STARS
```

```
##
##                       Df Sum of Sq    RSS    AIC
## - pH                   1      0.74 6912.8 1532.3
## - FixedAcidity         1      1.48 6913.6 1532.8
## - Sulphates            1      2.58 6914.7 1533.7
## <none>                             6912.1 1533.7
## - TotalSulfurDioxide   1      3.16 6915.2 1534.1
## + ResidualSugar        1      0.47 6911.6 1535.4
## + CitricAcid           1      0.07 6912.0 1535.7
## - Density              1      5.87 6918.0 1536.1
## - FreeSulfurDioxide    1      7.45 6919.5 1537.3
## - Chlorides            1      7.96 6920.0 1537.7
## - Alcohol              1     15.65 6927.7 1543.4
## - VolatileAcidity      1     31.57 6943.7 1555.3
## - AcidIndex            1    197.25 7109.3 1677.0
## - LabelAppeal          1   1526.68 8438.8 2561.7
## - STARS                1   1960.06 8872.1 2820.2
##
## Step:  AIC=1532.29
## TARGET ~ FixedAcidity + VolatileAcidity + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + Density + Sulphates + Alcohol + LabelAppeal +
##     AcidIndex + STARS
##
##                       Df Sum of Sq    RSS    AIC
## - FixedAcidity         1      1.47 6914.3 1531.4
## - Sulphates            1      2.62 6915.4 1532.2
## <none>                             6912.8 1532.3
## - TotalSulfurDioxide   1      3.20 6916.0 1532.7
## + pH                   1      0.74 6912.1 1533.7
## + ResidualSugar        1      0.49 6912.3 1533.9
## + CitricAcid           1      0.07 6912.8 1534.2
## - Density              1      5.86 6918.7 1534.7
## - FreeSulfurDioxide    1      7.50 6920.3 1535.9
## - Chlorides            1      7.87 6920.7 1536.2
## - Alcohol              1     15.82 6928.6 1542.1
## - VolatileAcidity      1     31.58 6944.4 1553.8
## - AcidIndex            1    196.52 7109.3 1675.0
## - LabelAppeal          1   1526.34 8439.2 2559.9
## - STARS                1   1961.75 8874.6 2819.6
##
## Step:  AIC=1531.39
## TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +
##     Density + Sulphates + Alcohol + LabelAppeal + AcidIndex +
##     STARS
##
##                       Df Sum of Sq    RSS    AIC
## - Sulphates            1      2.49 6916.8 1531.2
## <none>                             6914.3 1531.4
## - TotalSulfurDioxide   1      3.13 6917.4 1531.7
## + FixedAcidity         1      1.47 6912.8 1532.3
## + pH                   1      0.73 6913.6 1532.8
## + ResidualSugar        1      0.49 6913.8 1533.0
## + CitricAcid           1      0.07 6914.2 1533.3
## - Density              1      5.87 6920.2 1533.8
```

```
## - FreeSulfurDioxide   1      7.61 6921.9 1535.1
## - Chlorides           1      7.86 6922.2 1535.2
## - Alcohol             1     15.80 6930.1 1541.2
## - VolatileAcidity     1     31.38 6945.7 1552.8
## - AcidIndex           1    195.95 7110.3 1673.6
## - LabelAppeal         1   1527.29 8441.6 2559.4
## - STARS               1   1962.20 8876.5 2818.7
##
## Step:  AIC=1531.25
## TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +
##     Density + Alcohol + LabelAppeal + AcidIndex + STARS
##
##                       Df Sum of Sq    RSS    AIC
## <none>                             6916.8 1531.2
## + Sulphates           1      2.49 6914.3 1531.4
## - TotalSulfurDioxide  1      3.10 6919.9 1531.6
## + FixedAcidity        1      1.34 6915.4 1532.2
## + pH                  1      0.76 6916.0 1532.7
## + ResidualSugar       1      0.49 6916.3 1532.9
## + CitricAcid          1      0.05 6916.7 1533.2
## - Density             1      5.81 6922.6 1533.6
## - FreeSulfurDioxide   1      7.37 6924.2 1534.8
## - Chlorides           1      7.95 6924.7 1535.2
## - Alcohol             1     15.63 6932.4 1540.9
## - VolatileAcidity     1     31.36 6948.1 1552.6
## - AcidIndex           1    198.43 7115.2 1675.2
## - LabelAppeal         1   1526.51 8443.3 2558.5
## - STARS               1   1965.99 8882.8 2820.3
```

```
summary(wine_OLS1)
```

```
##
## Call:
## lm(formula = TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide +
##     TotalSulfurDioxide + Density + Alcohol + LabelAppeal + AcidIndex +
##     STARS, data = wine_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8202 -0.5183  0.1224  0.7232  3.2976
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         4.533e+00  6.125e-01   7.402 1.56e-13 ***
## VolatileAcidity    -1.002e-01  2.074e-02  -4.832 1.39e-06 ***
## Chlorides          -1.242e-01  5.106e-02  -2.433 0.014992 *
## FreeSulfurDioxide   2.538e-04  1.083e-04   2.343 0.019142 *
## TotalSulfurDioxide  1.062e-04  6.996e-05   1.518 0.128975
## Density            -1.261e+00  6.061e-01  -2.081 0.037494 *
## Alcohol             1.473e-02  4.319e-03   3.411 0.000652 ***
## LabelAppeal         6.546e-01  1.941e-02  33.717  < 2e-16 ***
## AcidIndex          -1.669e-01  1.373e-02 -12.156  < 2e-16 ***
## STARS               7.337e-01  1.918e-02  38.263  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.159 on 5151 degrees of freedom
## Multiple R-squared:  0.4454, Adjusted R-squared:  0.4445
## F-statistic: 459.7 on 9 and 5151 DF,  p-value: < 2.2e-16
```

```
Metrics1 <- data.frame(
  R2 = rsquare(wine_OLS1, data = wine_train),
  RMSE = rmse(wine_OLS1, data = wine_train),
  MAE = mae(wine_OLS1, data = wine_train)
)
print(Metrics1)
```

```
##          R2     RMSE      MAE
## 1 0.4454482 1.157671 0.855548
```

The Stepwise variable selection process identified five variables significant which are VolatileAcidity, Alcohol, LabelAppeal, AcidIndex and STARS with an R^2 of 45%, Residual Error of 1.159 and F-Statistic of 459.7.

Notice that some of the coefficients are negative which means these variables will negatively impact the number of wine cases purchased.

We will explore these coefficient a little further in this analysis.

Multilinear Regression Model - Five variables

```
wine_OLS2 <-lm(TARGET ~ VolatileAcidity + Alcohol + LabelAppeal + AcidIndex + STARS, data = wine_train)
summary(wine_OLS2)
```

```
##
## Call:
## lm(formula = TARGET ~ VolatileAcidity + Alcohol + LabelAppeal +
##     AcidIndex + STARS, data = wine_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8491 -0.5212  0.1209  0.7302  3.1772
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.309222   0.126742  26.110  < 2e-16 ***
## VolatileAcidity -0.101673   0.020770  -4.895 1.01e-06 ***
## Alcohol          0.014585   0.004322   3.375 0.000744 ***
## LabelAppeal      0.655134   0.019436  33.707  < 2e-16 ***
## AcidIndex       -0.168766   0.013738 -12.285  < 2e-16 ***
## STARS            0.734810   0.019193  38.286  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.16 on 5155 degrees of freedom
## Multiple R-squared:  0.4434, Adjusted R-squared:  0.4429
## F-statistic: 821.5 on 5 and 5155 DF,  p-value: < 2.2e-16
```

```
Metrics2 <- data.frame(
  R2 = rsquare(wine_OLS2, data = wine_train),
  RMSE = rmse(wine_OLS2, data = wine_train),
  MAE = mae(wine_OLS2, data = wine_train)
)
print(Metrics2)
```

```
##          R2     RMSE       MAE
## 1 0.4434464 1.159759 0.8575683
```

Although the F-Statistic improved to 821.5, the R^2 did not show any significant improvement.

Multilinear Regression Model - Two variables

Using only the two variables with positive coefficient yield the folling model outcome.

```
wine_OLS3 <-lm(TARGET ~ LabelAppeal + STARS, data = wine_train)
summary(wine_OLS3)
```

```
##
## Call:
## lm(formula = TARGET ~ LabelAppeal + STARS, data = wine_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8984 -0.5533  0.1572  0.7463  3.1572
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.07658    0.04260   48.74   <2e-16 ***
## LabelAppeal  0.64472    0.01977   32.62   <2e-16 ***
## STARS        0.76621    0.01940   39.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.182 on 5158 degrees of freedom
## Multiple R-squared:  0.4225, Adjusted R-squared:  0.4223
## F-statistic:  1887 on 2 and 5158 DF,  p-value: < 2.2e-16
```

```
Metrics3 <- data.frame(
  R2 = rsquare(wine_OLS3, data = wine_train),
  RMSE = rmse(wine_OLS3, data = wine_train),
  MAE = mae(wine_OLS3, data = wine_train)
)
print(Metrics3)
```

```
##          R2     RMSE       MAE
## 1 0.4225459 1.181334 0.8669629
```

Although the F-Statistic improved to 1887 the R^2 did not show slight decrease from the previuos OLS Models.

Model Performance

Evaluating the Selected Model

Comparing the Poisson Model Fit

```
ModelName <- c("wine_poisson1", "wine_poisson2","wine_poisson3")
Model_PseudRSq <- c("31.50%", "31.20% ", "31.04%")
Model_Overdisp <- c("0.6314", "0.6325", "0.6341")
Model_Deviance <- c("3249.56", "3261.36", "3269.81")
Model_AIC <- c("18,608", "18,600", "18,606")
Model_BIC <- c("18,706", "18,633", "18,632")
Model_Performance <- data.frame(ModelName,Model_PseudRSq,Model_Overdisp,Model_Deviance,Model_AIC,Model_
Model_Performance
```

```
##       ModelName Model_PseudRSq Model_Overdisp Model_Deviance Model_AIC
## 1 wine_poisson1         31.50%         0.6314        3249.56    18,608
## 2 wine_poisson2        31.20%          0.6325        3261.36    18,600
## 3 wine_poisson3         31.04%         0.6341        3269.81    18,606
##   Model_BIC
## 1    18,706
## 2    18,633
## 3    18,632
```

Comparing the OLS Model Fit

```
ModelName <- c("wine_OLS1", "wine_OLS2","wine_OLS3")
Model_RSquared <- c("45%", "44% ", "42%")
Model_RMSE <- c("1.157", "1.159", "1.181")
Model_FStatistic <- c("459.7", "821.5", "1887.0")
Model_Performance <- data.frame(ModelName,Model_RSquared,Model_RMSE,Model_FStatistic)
Model_Performance
```

```
##   ModelName Model_RSquared Model_RMSE Model_FStatistic
## 1 wine_OLS1            45%      1.157            459.7
## 2 wine_OLS2           44%       1.159            821.5
## 3 wine_OLS3            42%      1.181           1887.0
```

The performance Metrics above indicates that none of the models are performing at optimal level. Although the OLS Models have better R^2, we will not be deploying any of the OLS model since this is a count problem wher we're trying the predict the of wine cases purchased based on certain properties of the wine.

The Regression tree from the begining indicated that only three (3) variables are significant for our prediction and several other Models we have analyzed also shows that these three variables continue to be significant and consistent.

Therefore, we will select the count Model with these three variables which is wine_poisson3 as our selected Model for this task.

Coefficients Analysis

```
exp(coef(wine_poisson3))
```

```
## (Intercept) LabelAppeal   AcidIndex       STARS
##   3.5132405   1.1972697   0.9499686   1.2103250
```

From the above, we can say that one unit increase in LabelAppeal will likely increase the chance of the number of wine cases purchased by 1.197.

One unit increase in AcidIndex will likely increase the chance of the number of wine cases purchased by 0.949.

One unit increase in STARS will likely increase the chance of the number of wine cases purchased by 1.210.

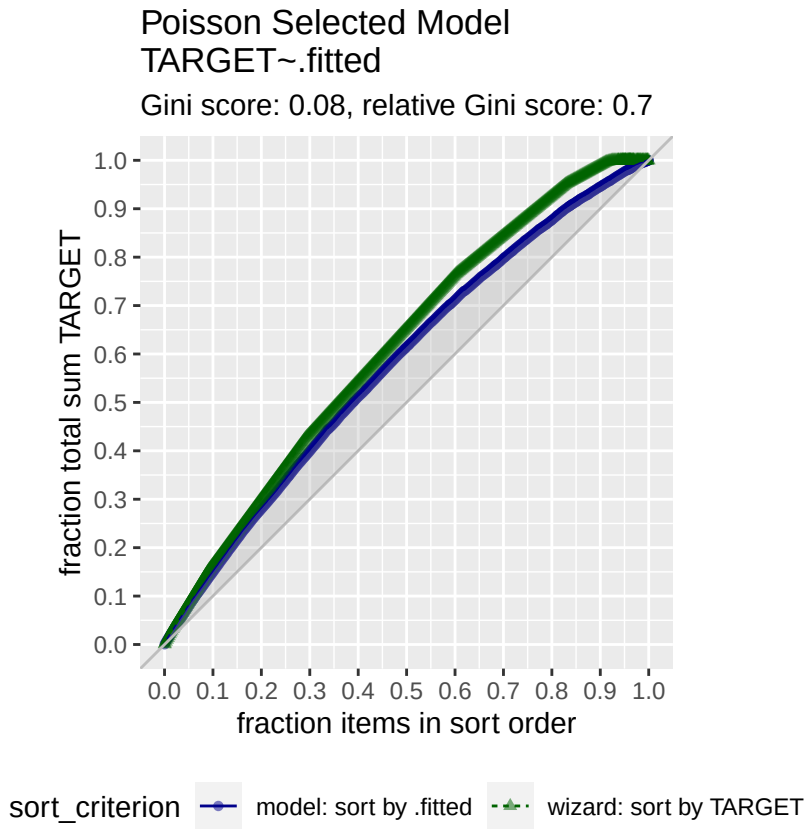
The most important aspect of Poisson regression is that exponentiated parameters have a multiplicative rather than an additive effect on the response variable.

Model Cross Validation

The use case for Gain Curve is to compare a predictive model score to an actual outcome (either binary (0/1) or continuous). In this case the gain curve plot measures how well the model score sorts the data compared to the true outcome value.

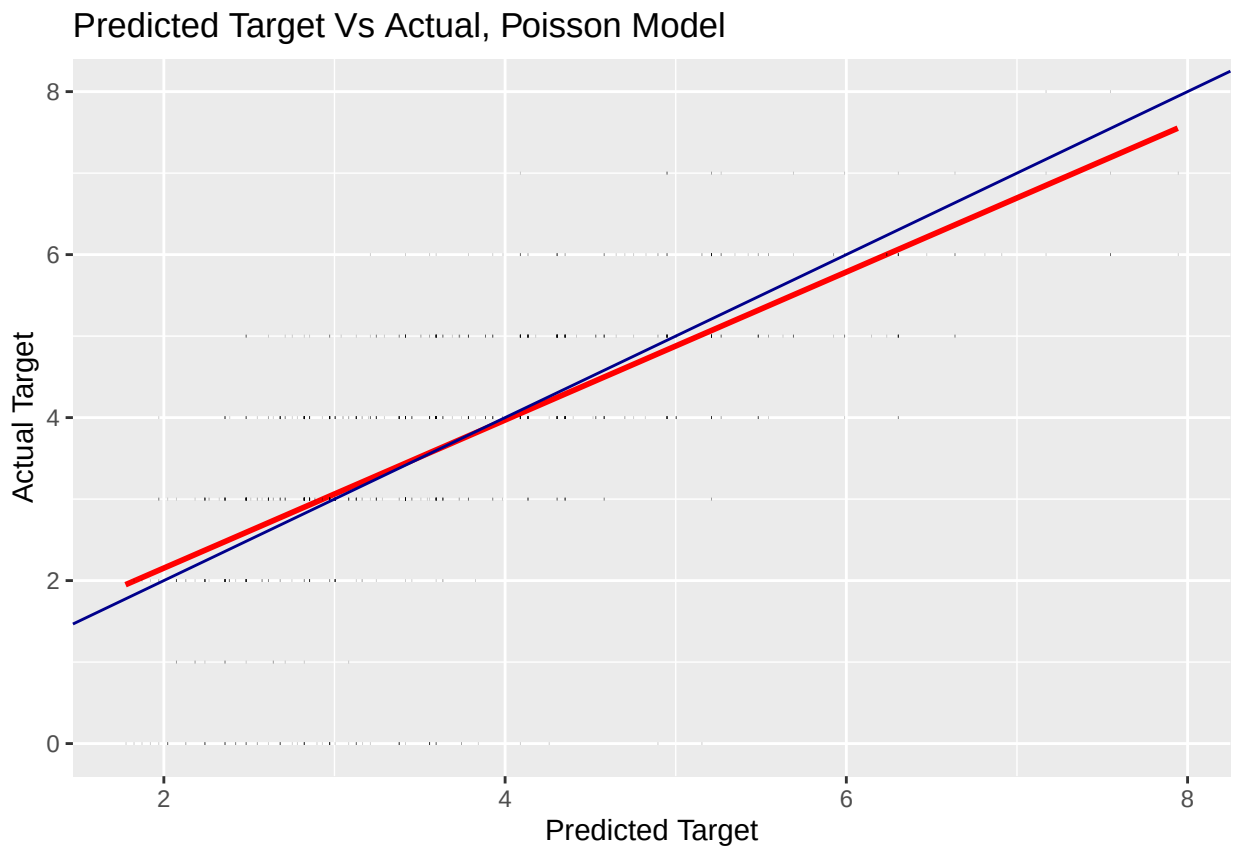
Cross Validating the Model using Gain Curve on training dataset yields the following chart and a relative Gini score of 70%.

```
wineGini <- augment(wine_poisson3) %>% data.frame(wine_train) %>%
          GainCurvePlot(xvar = ".fitted", truthVar = "TARGET", title = "Poisson Selected Model")
wineGini
```



Using Test Dataset

```
prediction <- predict(wine_poisson3,newdata = wine_test,type="response")
ggplot(wine_test, aes(x = prediction, y = wine_test$TARGET)) +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  geom_segment(aes(xend = prediction, yend = wine_test$TARGET), alpha = .2) +
  geom_abline(color = "darkblue")+
  labs(y = "Actual Target", x = "Predicted Target") +
  ggtitle("Predicted Target Vs Actual, Poisson Model")
```



The Model prediction is close to Actual as indicated by evaluating the Model using the testing data.

Model Prediction

Making predictions using the evaluation Dataset

```
newEvaluation <- na.omit(wine_evaluation)
prediction <- predict(wine_poisson3,newdata = newEvaluation,type="response")
newEvaluation$TARGET <- round(prediction, digits = 0)
newEvaluation <- newEvaluation[,c(16,2:15)]
View(newEvaluation)
write.csv(newEvaluation, "Poisson_Prediction.csv")
```

Conclusion

The data was not a good fit for a Negative Binomial Model based on the expectation of a Negative Binomial distribution. The Theta could not converge at a reason value after multiple iteration.

The selected Poisson Model was based on the predictive ability obtained from the Regression Tree and the conconrence of the subsequent models concerning the significant variables. The selected Model was also based on the peformance from the AIC, Pseudo R^2, Deviance value, Overdispersion and the significant p-value from the Vuong Non-Nested Hypothesis Test-Statistic.

Cross Validating the selected Model shows relative Gini score of 70% and evaluating the selected model using the testing data shows predictive values close to the actuals as can be seen from the Target Vs Actual plot above.

From the coefficient analysis, we saw that one unit increase in LabelAppeal will likely increase the chance of the number of wine cases purchased by 1.197. One unit increase in AcidIndex will likely increase the chance of the number of wine cases purchased by 0.949 and one unit increase in STARS will likely increase the chance of the number of wine cases purchased by 1.210.

Using the Evaluation data provided, the Model was able to predict new Target as the number of wine cases to be purchased using the chemical properties from the significant variables identified above. The result of the new target or Model prediction is saved as a new variable called newEvaluation and can be viewed directly or saved as csv file.