Wine Quality

Introduction

Wine is a part of culture for many people. For me, wine is the only thing I like in weddings. At some point I will get married and I will have to serve out wine for my guests so they can bear through mine.

Problem

Can I determine wine quality based off of quantitative and objective data?

Data Citation:

Paulo Cortez, University of Minho, Guimarães, Portugal, http://www3.dsi.uminho.pt/pcortez A. Cerdeira, F. Almeida, T. Matos and J. Reis, Viticulture Commission of the Vinho Verde Region(CVRVV), Porto, Portugal @2009

Fundamental Setup

```
library(caret)
```

```
df = read.csv("winequality-white.csv",sep=";",header=TRUE)
```

```
for (x in 1:12){
  y = which(is.na(as.numeric(unlist(df[x]))))
  print(y)
}
```

```
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
## integer(0)
```

Data had no missing values so there was nothing to clean.

Partitioning to Training & Testing

```
trainIndex = createDataPartition(df$quality, p = .8, list = FALSE)

train = df[trainIndex,]
test = df[-trainIndex, ]
```

Training & Testing datasets created in 4 to 1 ratio from base data respectively.

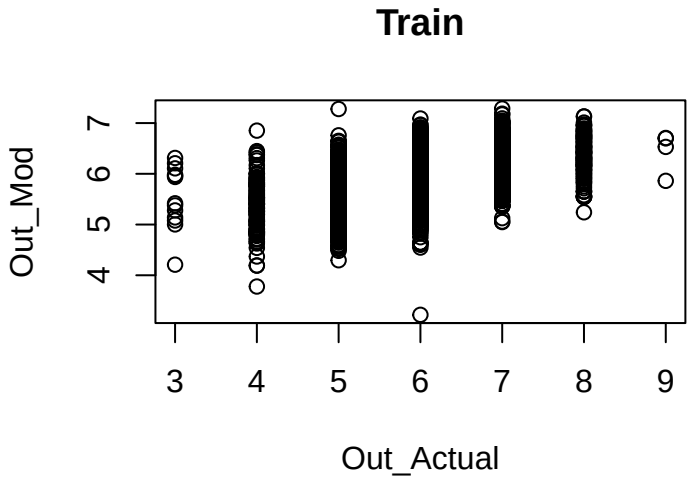
Creating Basic Model

```
mod = train(quality ~., data = train, method = "lm",
             preProcess = c("scale", "center"), trControl = trainControl("none"))
mod_training = predict(mod, train)
mod_testing = predict(mod, test)

trainmodDF = data.frame(train$quality, mod_training)
trainmodDF = subset(trainmodDF, trainmodDF$mod_training>0)
testmodDF = data.frame(test$quality, mod_testing)
testmodDF = subset(testmodDF, testmodDF$mod_testing>0)
```

Basic model used all variables to determine quality output. Data excludes model values less than zero because logically quality values cannot be less than zero on a scale of 0 to 10 and a single outlier can completely alter the results. The occurences of outliers are only one below zero.

Results

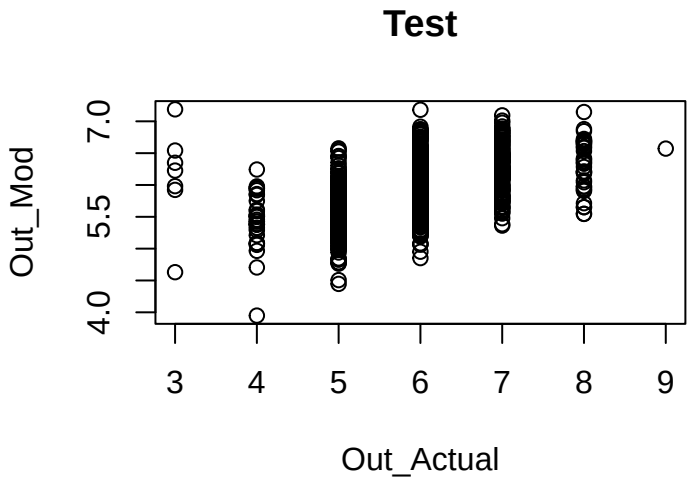
```
plot(trainmodDF$train.quality, trainmodDF$mod_training, xlab = "Out_Actual",
     ylab = "Out_Mod", main = "Train")
```




```
cor(trainmodDF$train.quality, trainmodDF$mod_training)
```

```
## [1] 0.5355109
```

```
plot(testmodDF$test.quality, testmodDF$mod_testing, xlab = "Out_Actual",
     ylab = "Out_Mod", main = "Test")
```




```
cor(testmodDF$test.quality, testmodDF$mod_testing)
```

```
## [1] 0.5096022
```

```r
model = lm(testmodDF$test.quality~testmodDF$mod_testing, data = df)
summary(residuals(model))
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -4.1396 -0.4876 -0.0538  0.0000  0.5056  2.4689
```

```r
summary(trainmodDF$train.quality) #Actual model stats
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.879   6.000   9.000
```

```r
summary(trainmodDF$mod_training) #Basic model stats
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.221   5.552   5.863   5.879   6.228   7.286
```

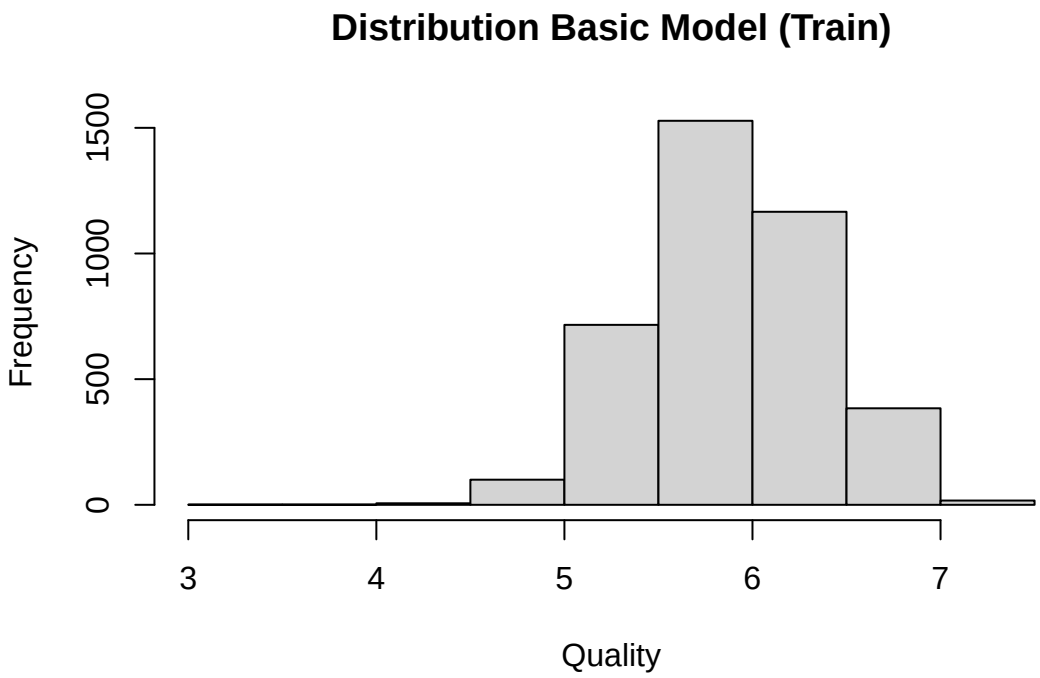
```r
summary(testmodDF$test.quality) #Actual model stats
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.872   6.000   9.000
```

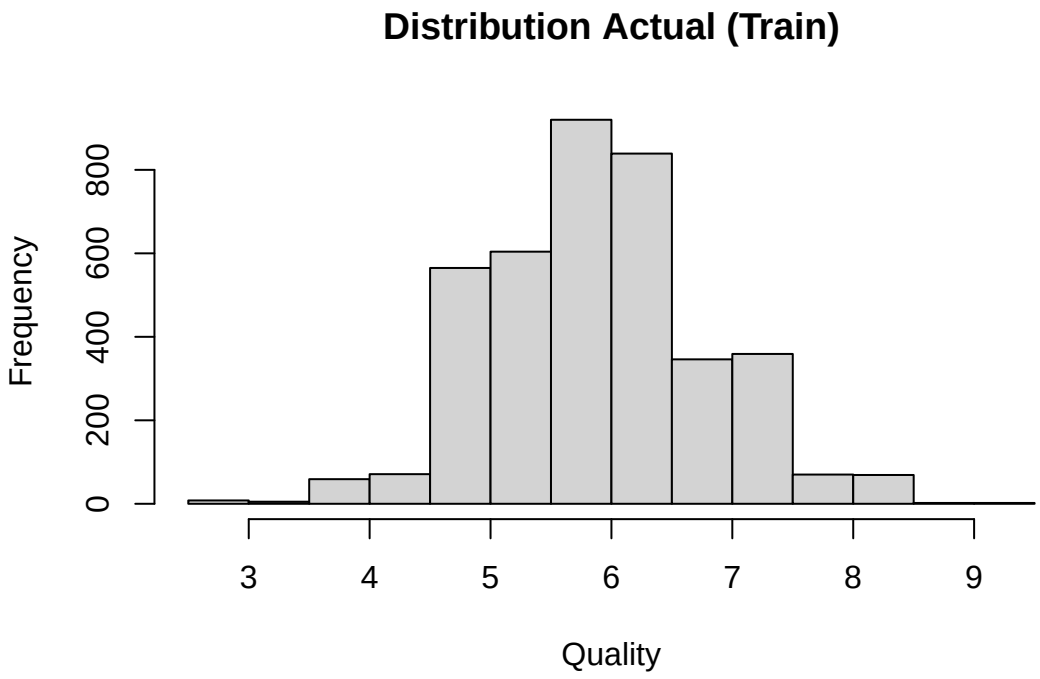
```r
summary(testmodDF$mod_testing) #Basic model stats
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.948   5.545   5.868   5.895   6.238   7.188
```

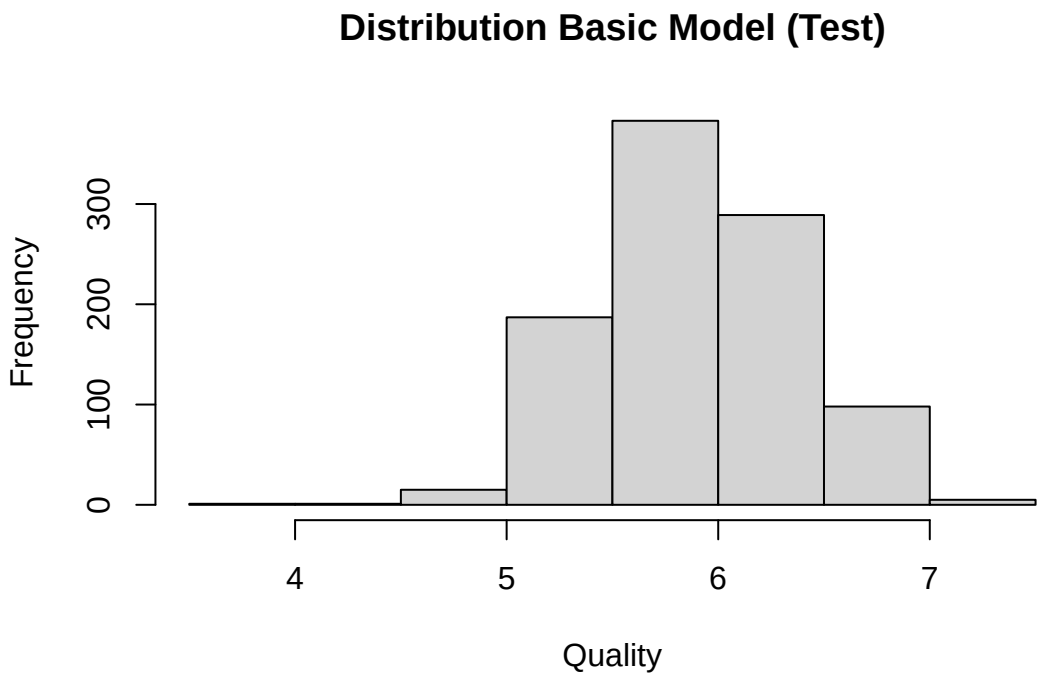
```
hist(trainmodDF$mod_training, xlab = "Quality", main = "Distribution Basic Model (Train)")
```



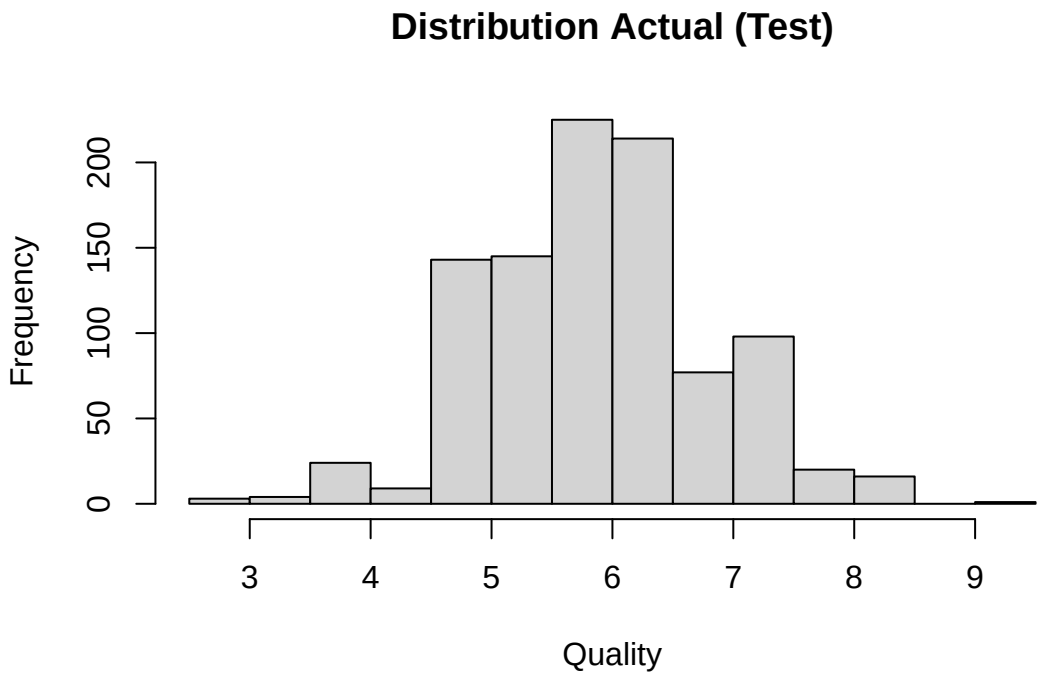

```
hist(jitter(trainmodDF$train.quality), xlab = "Quality", main = "Distribution Actual (Train)")
```



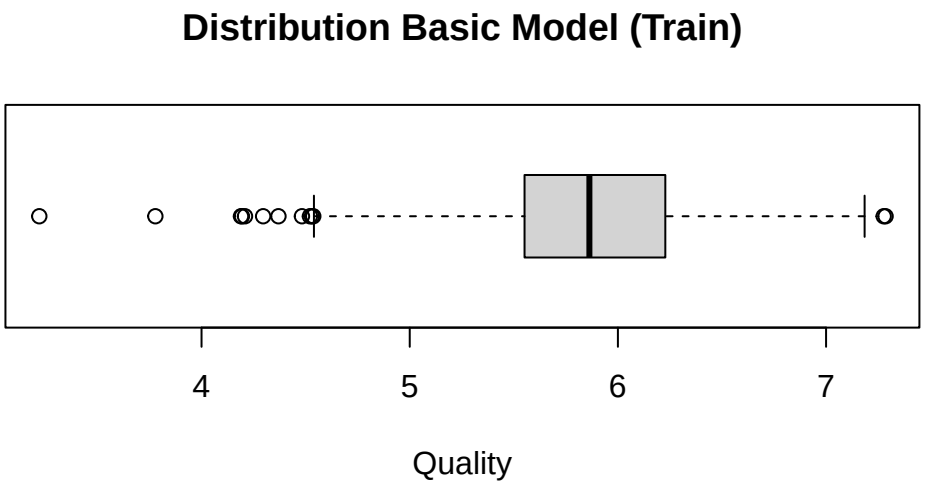
```
hist(testmodDF$mod_testing, xlab = "Quality", main = "Distribution Basic Model (Test)")
```



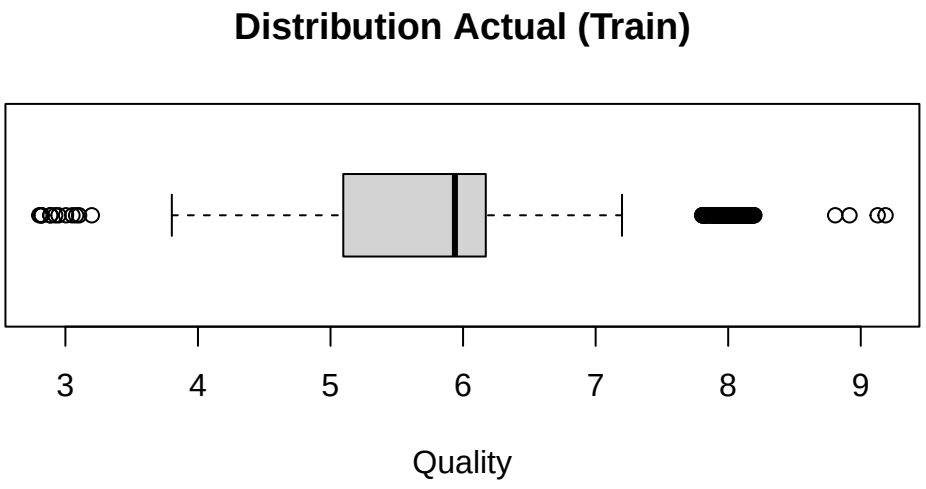

```
hist(jitter(testmodDF$test.quality), xlab = "Quality", main = "Distribution Actual (Test)")
```



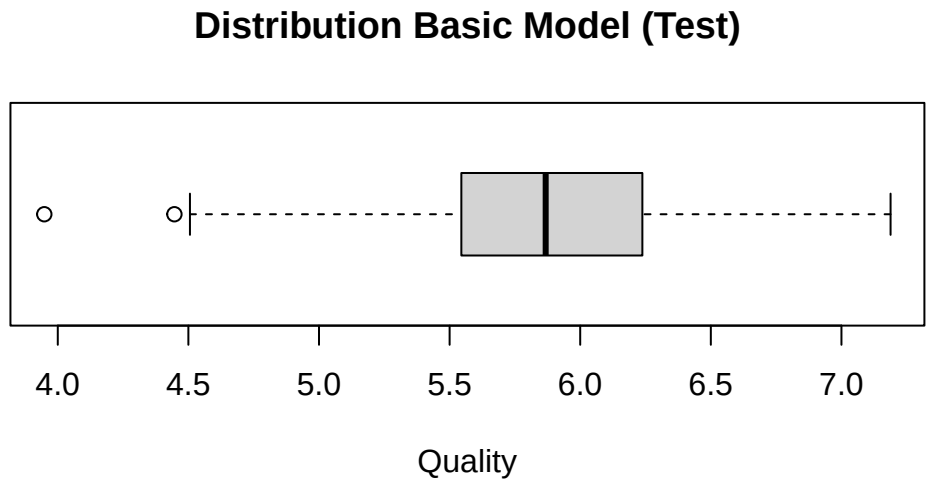
```
boxplot(trainmodDF$mod_training, xlab = "Quality",
        main = "Distribution Basic Model (Train)", horizontal = TRUE)
```



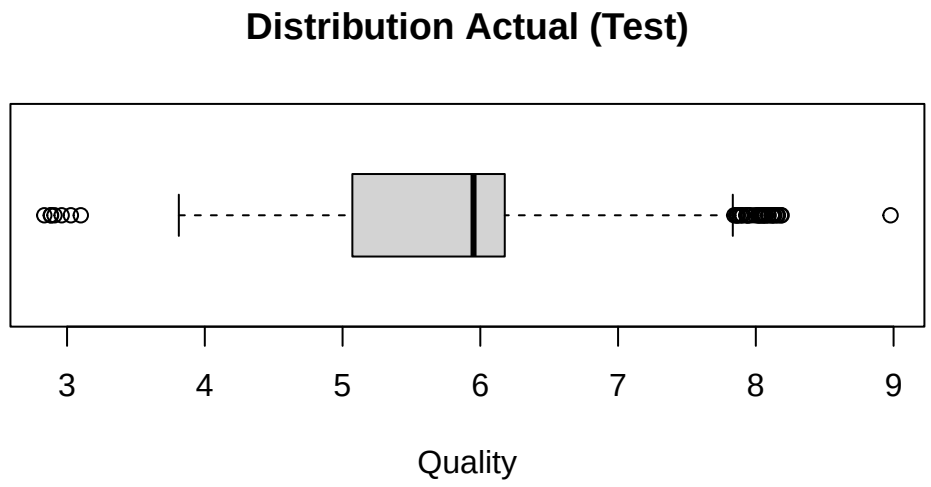

```
boxplot(jitter(trainmodDF$train.quality), xlab = "Quality",
        main = "Distribution Actual (Train)", horizontal = TRUE)
```



```
boxplot(testmodDF$mod_testing, xlab = "Quality",
        main = "Distribution Basic Model (Test)", horizontal = TRUE)
```




```
boxplot(jitter(testmodDF$test.quality), xlab = "Quality",
        main = "Distribution Actual (Test)", horizontal = TRUE)
```



Note: Observations made based off multiple iterations of model and I cannot give out a single number due to nature of RMarkdown.

Model has a correlation in the .5 are basically meaning about 40%-50% of data points in the plot can't be explained by model. The distribution of the train and test models model are mostly in line with the actual model. The models however skew more to the left, have less outliers, and seldom reach even proximity of the max value of 9. Sometimes the models have values that reach below 3.

Exploration for Improvement

```
model = glm(quality~., data=train)
summary(model)
```

```
##
## Call:
## glm(formula = quality ~ ., data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3125  -0.4978  -0.0357   0.4534   3.1384
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.394e+02  2.015e+01   6.918 5.31e-12 ***
## fixed.acidity         5.799e-02  2.272e-02   2.552   0.0107 *
## volatile.acidity     -1.870e+00  1.264e-01 -14.795  < 2e-16 ***
## citric.acid           1.232e-02  1.073e-01   0.115   0.9087
## residual.sugar        7.516e-02  8.184e-03   9.184  < 2e-16 ***
## chlorides             1.064e-01  6.217e-01   0.171   0.8641
## free.sulfur.dioxide   4.903e-03  9.555e-04   5.131 3.02e-07 ***
## total.sulfur.dioxide -1.387e-04  4.146e-04  -0.335   0.7379
## density              -1.395e+02  2.044e+01  -6.823 1.03e-11 ***
## pH                    6.557e-01  1.152e-01   5.693 1.34e-08 ***
## sulphates             6.054e-01  1.098e-01   5.514 3.72e-08 ***
## alcohol               2.111e-01  2.607e-02   8.097 7.47e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.5560353)
##
##     Null deviance: 3045.9  on 3918  degrees of freedom
## Residual deviance: 2172.4  on 3907  degrees of freedom
## AIC: 8835.5
##
## Number of Fisher Scoring iterations: 2
```

Summary of model between quality and variables. values we're looking at in this are the P values on the far right under coefficients. We can see that citric acid, chlorides, and total sulfur dioxide have extremely high P values and therefore should be discarded in final model.

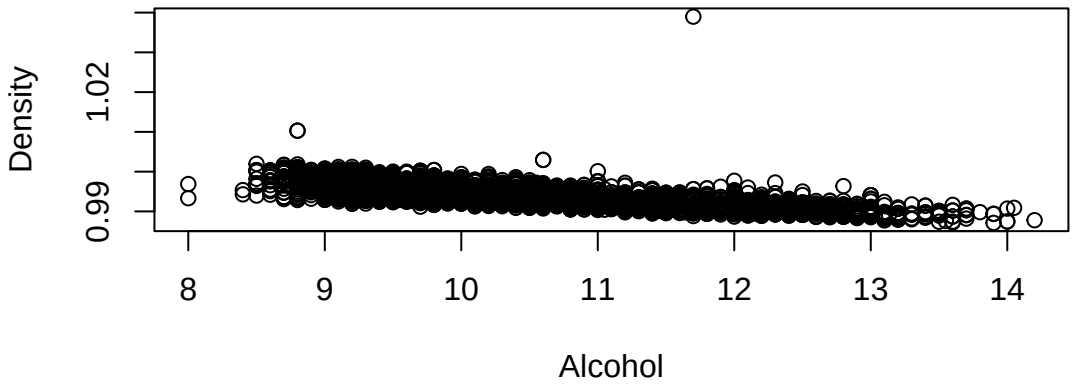
```
cor(train)
```

```
##                       fixed.acidity volatile.acidity  citric.acid residual.sugar
## fixed.acidity           1.00000000      -0.02183647  0.283121787     0.08213545
## volatile.acidity       -0.02183647       1.00000000 -0.160869742     0.06262040
## citric.acid             0.28312179      -0.16086974  1.000000000     0.09007521
## residual.sugar          0.08213545       0.06262040  0.090075209     1.00000000
## chlorides               0.01386524       0.05858220  0.117964531     0.08085148
## free.sulfur.dioxide    -0.06634639      -0.11226925  0.091150461     0.29901076
## total.sulfur.dioxide    0.08078027       0.08010449  0.107766437     0.39650190
## density                 0.25525899       0.02670778  0.147426986     0.83804826
## pH                     -0.41844990      -0.03184732 -0.161033336    -0.19423776
## sulphates              -0.02287961      -0.04429385  0.070512105    -0.03124943
## alcohol                -0.11261594       0.07269003 -0.078662065    -0.43913297
## quality                -0.11341391      -0.19768344 -0.006567824    -0.09976821
##                        chlorides free.sulfur.dioxide total.sulfur.dioxide
## fixed.acidity         0.01386524        -0.066346392          0.080780275
## volatile.acidity      0.05858220        -0.112269252          0.080104494
## citric.acid           0.11796453         0.091150461          0.107766437
## residual.sugar        0.08085148         0.299010762          0.396501895
## chlorides             1.00000000         0.101789587          0.194240142
## free.sulfur.dioxide   0.10178959         1.000000000          0.604874641
## total.sulfur.dioxide  0.19424014         0.604874641          1.000000000
## density               0.25289753         0.286511891          0.523657715
## pH                   -0.08012672         0.003336403          0.007450529
## sulphates             0.01491980         0.055240381          0.127572613
## alcohol              -0.36259534        -0.241173178         -0.442628508
## quality              -0.19971391         0.037078881         -0.157670924
##                           density           pH    sulphates      alcohol
## fixed.acidity         0.25525899 -0.418449895 -0.022879613 -0.112615940
## volatile.acidity      0.02670778 -0.031847319 -0.044293854  0.072690031
## citric.acid           0.14742699 -0.161033336  0.070512105 -0.078662065
## residual.sugar        0.83804826 -0.194237757 -0.031249431 -0.439132967
## chlorides             0.25289753 -0.080126722  0.014919795 -0.362595336
## free.sulfur.dioxide   0.28651189  0.003336403  0.055240381 -0.241173178
## total.sulfur.dioxide  0.52365771  0.007450529  0.127572613 -0.442628508
## density               1.00000000 -0.086196251  0.067168321 -0.771324003
## pH                   -0.08619625  1.000000000  0.151990460  0.111748039
## sulphates             0.06716832  0.151990460  1.000000000 -0.009991385
## alcohol              -0.77132400  0.111748039 -0.009991385  1.000000000
## quality              -0.30830587  0.097394404  0.060840757  0.437519650
##                           quality
## fixed.acidity        -0.113413910
## volatile.acidity     -0.197683440
## citric.acid          -0.006567824
## residual.sugar       -0.099768210
## chlorides            -0.199713908
## free.sulfur.dioxide   0.037078881
## total.sulfur.dioxide -0.157670924
## density              -0.308305874
## pH                    0.097394404
## sulphates             0.060840757
## alcohol               0.437519650
```

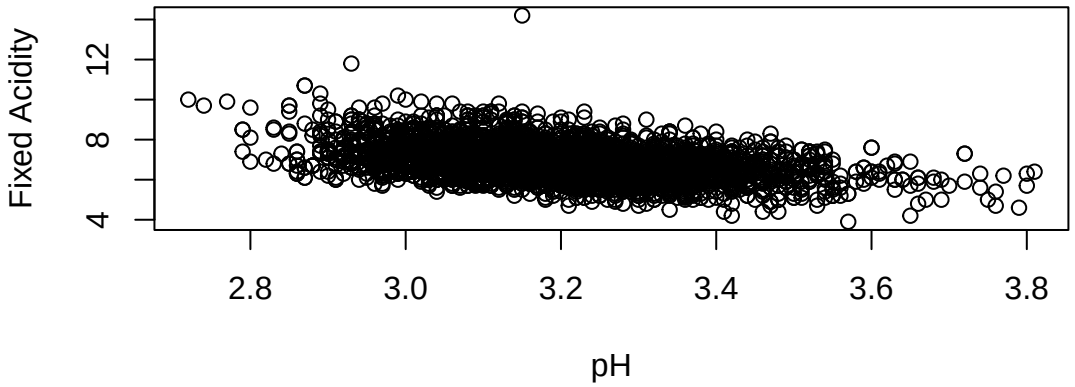
```
## quality               1.000000000
```

Correlation between variables. I will be including all corelations with absolute values consistently at or above .2 into the model.

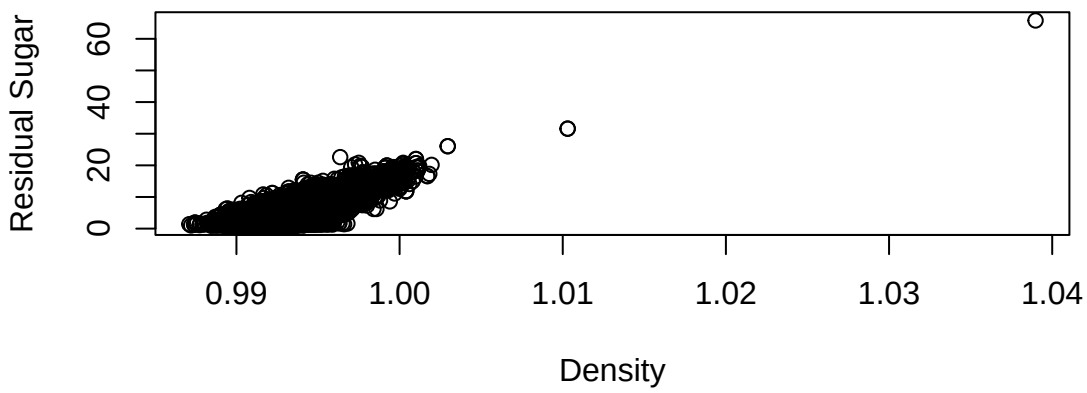
```
plot(train$alcohol, train$density, xlab = "Alcohol", ylab = "Density")
```




```
plot(train$pH, train$fixed.acidity, xlab = "pH", ylab = "Fixed Acidity")
```

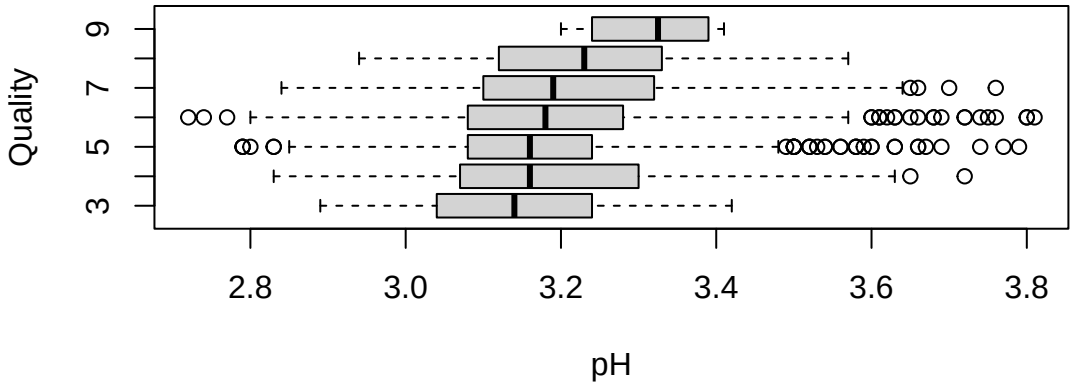


```
plot(train$density, train$residual.sugar, xlab = "Density", ylab = "Residual Sugar")
```

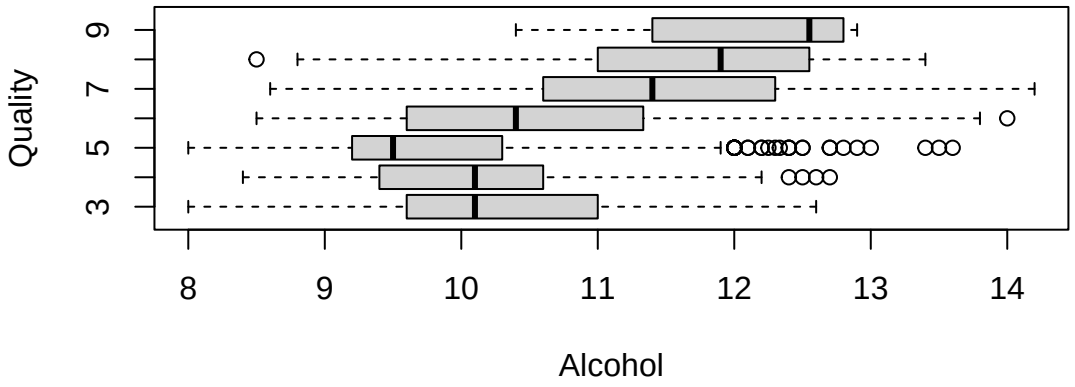


Some plots of functions with some corelation.

```
boxplot(train$pH~train$quality, horizontal = TRUE, ylab = "Quality", xlab = "pH")
```




```
boxplot(train$alcohol~train$quality, horizontal = TRUE, ylab = "Quality", xlab = "Alcohol")
```



Boxplots for pH and alcohol. Both appear to have median trend closer sqare root shape.

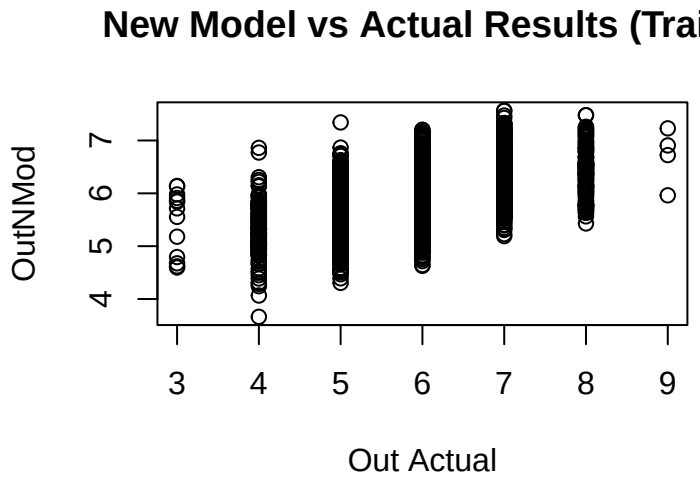
Building New Model

```
newmod = train(quality ~ fixed.acidity + volatile.acidity + residual.sugar +
               free.sulfur.dioxide + density + sqrt(pH) + sulphates +
               sqrt(alcohol) +

               alcohol*density +
               density*residual.sugar +
               density*total.sulfur.dioxide +
               total.sulfur.dioxide*free.sulfur.dioxide +
               total.sulfur.dioxide*residual.sugar +
               pH*fixed.acidity +
               alcohol*total.sulfur.dioxide +
               alcohol*residual.sugar +
               alcohol*chlorides

               + alcohol*total.sulfur.dioxide +
               alcohol*free.sulfur.dioxide +
               density*fixed.acidity +
               density*chlorides +
               density*free.sulfur.dioxide +
               citric.acid * fixed.acidity
               , data = train,
               method = "lm", preProcess = c("scale", "center"),
               trControl = trainControl("none"))

newmod_training = predict(newmod, train)
newmod_testing = predict(newmod, test)

newtrainmodDF = data.frame(train$quality, newmod_training)
newtrainmodDF = subset(newtrainmodDF, newtrainmodDF$newmod_training>0)
newtestmodDF = data.frame(test$quality, newmod_testing)
newtestmodDF = subset(newtestmodDF, newtestmodDF$newmod_testing>0)
```

New model adjustments include discarding some variables with quality, relations between variables, and square rooting pH and alcohol.

Results New Model

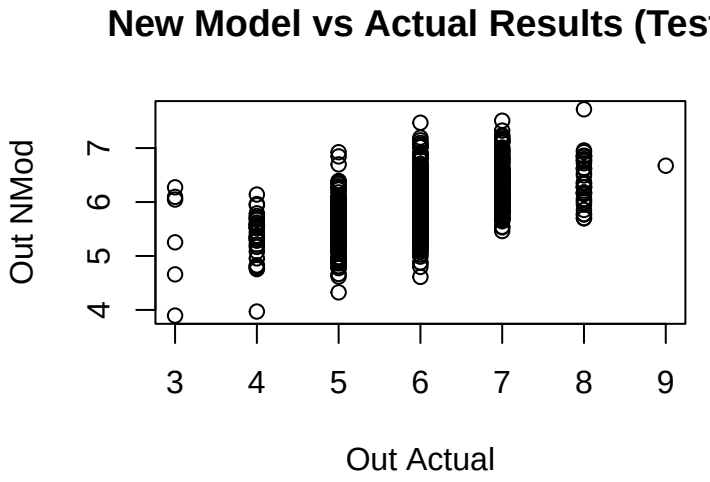
```
plot(newtrainmodDF$train.quality, newtrainmodDF$newmod_training, xlab = "Out Actual",
     ylab = "OutNMod", main = "New Model vs Actual Results (Train)")
```




```
cor(newtrainmodDF$train.quality, newtrainmodDF$newmod_training)
```

```
## [1] 0.5708859
```

```
plot(newtestmodDF$test.quality, newtestmodDF$newmod_testing, xlab = "Out Actual",
     ylab = "Out NMod", main = "New Model vs Actual Results (Test)")
```




```
cor(newtestmodDF$test.quality, newtestmodDF$newmod_testing)
```

```
## [1] 0.5687948
```

```
model = lm(newtrainmodDF$train.quality~newtrainmodDF$newmod_training, data = df)
summary(residuals(model))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -3.13907 -0.49914 -0.01119  0.00000  0.43465  3.03779
```

```
summary(newtrainmodDF$train.quality) #Actual model stats
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.879   6.000   9.000
```

```
summary(newtrainmodDF$newmod_training) #Basic model stats
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.663   5.550   5.836   5.879   6.200   7.566
```

```
summary(newtestmodDF$test.quality) #Actual model stats
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.875   6.000   9.000
```

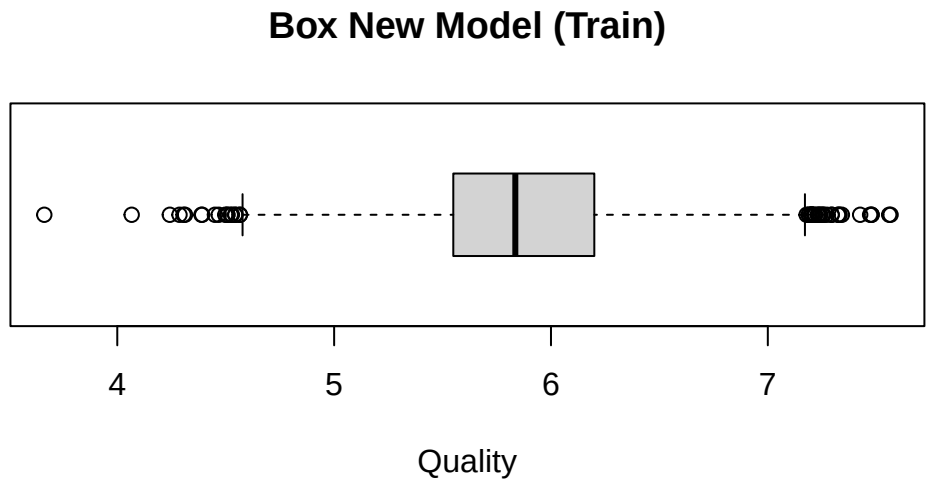
```
summary(newtestmodDF$newmod_testing) #Basic model stats
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.895   5.539   5.841   5.887   6.214   7.718
```

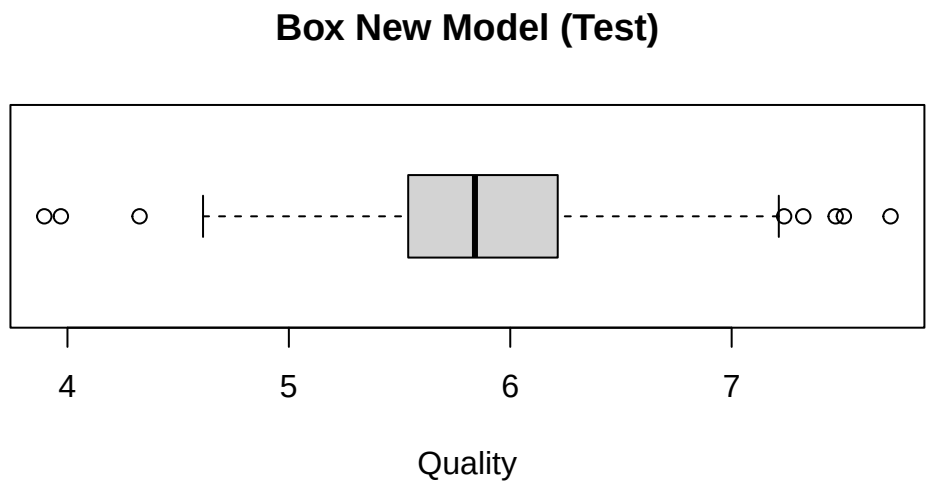
```
hist(newtrainmodDF$newmod_training, xlab = "New Mod Dist", main = "New Model Distribution (Train)")
```


```
hist(newtestmodDF$newmod_testing, xlab = "New Mod Dist", main = "New Model Distribution (Test)")
```

```
boxplot(newtrainmodDF$newmod_training, xlab = "Quality", main = "Box New Model (Train)",
        horizontal = TRUE)
```




```
boxplot(newtestmodDF$newmod_testing, xlab = "Quality", main = "Box New Model (Test)",
        horizontal = TRUE)
```



Results of New Model

Largely the same types of observations from the first model. Most significant differences are training boxplot has more outliers than 1st model training and correlation improves consistently by 4-5 percent.

Conclusions

Overall I was able somewhat reliably predict the quality of the wine data, with an average correlation of around .5-.6. However I was hoping I would be able to predict at a much more reliable rate than what I got.

The reason why it is hard to predict the quality rating is because we're trying to measure subjectivity and human judgement, which are both wildly varying and unpredictable. People have various preferences for which wine is better, being it a sweet one or a dry one, or whether more alcohol is better. The quality ratings are also at all times in the mercy of one's mood when they're doing the testing. They could give a wine a lower rating because they're in a bad mood or tired of the same-ish wine taste, or better because they're craving wine or heard a funny joke.

However, despite my shortcoming, I was able to create a more reliable model than the base one with throughout multiple iterations an improvement in predictions by 4-5 percent. Another success I was able to achieve is determine the single biggest variable in determining quality; alcohol content. When you think about it, it makes sense; who doesn't like getting drunk?

Data Citation:
Paulo Cortez, University of Minho, Guimarães, Portugal, http://www3.dsi.uminho.pt/pcortez A. Cerdeira, F. Almeida, T. Matos and J. Reis, Viticulture Commission of the Vinho Verde Region(CVRVV), Porto, Portugal @2009