Analyzing Biochemical Properties to Grade Wine Quality

"A method of procedure that has characterized natural science since the 17th century, consisting in systematic observation, measurement, and experiment, and the formulation, testing, and modification of hypotheses." ~ The Scientific Method

Abstract

Wine is a traditional alcoholic beverage that has been used since the dawn of human civilization. Once viewed as a luxury good, it is increasingly enjoyed by a wider range of consumers. To accommodate this growth, the wine industry is becoming more innovative. Quality evaluation is often considered in the certification process, and it can also be used to improve wine making. This is done by identifying the most influential factors. Having quality standards also helps price setting, because of the ability to distinguish premium brands. Consequently, the goal of this project is to find out critical variables that significantly affect the quality of wine, and later derive the optimal condition of these variables.

The dataset being used is a wine quality data set for red wine and is extracted from the UCI Machine Learning Repository. It contains 1599 cases and 11 variables. The variables are biochemical properties of red wine. At least 3 wine experts rated the quality of each wine, providing a rating between 0 (very bad) and 10 (very excellent).

The analysis will be made on the wine data set to find out which critical variables affect the quality of wine significantly. Their optimal conditions will then be derived. This process will be initiated by a univariate analysis which will include investigating variation of data through distribution plots and standard deviations. This will filter out variables that do not change quality of wine. The larger the variation in data the easier it is to distinguish wines. Secondly, a bivariate analysis will be used to compare relationships between each selected variable and the quality. This is mainly done through correlation computation and narrowing down selected variables to critical factors based on the relationships. Next, a multivariate analysis will be performed on the remaining variables which is focused on correlation of critical factors, leading to their optimal conditions. This analysis will include hypothesis testing such as an analysis of variance. The quality of wine will then be predicted based on the derived condition.

1. Introduction

The primary goal is to assess the factors that influence wine quality the most, and then to predict the wine quality using a model represented by these factors.

The first step of our analysis is to remove the outliers because outliers in the data can cause problems. They may indicate faulty or erroneous data and can implicate fitting in a model. Next, the distribution of each variable is analyzed to verify skewness; left-skewed, right-skewed or normally distributed. If the data does not have a symmetric normal distribution, then it is said to be skewed. A skewed distribution yields improper results for statistical tests. In case of a skewed

distribution, a variable is usually transformed using the log, Box-Cox or the square root transform. The analysis is then progressed with a plot of standard deviation for all variables to find which variables have the most variation and which, the least. This is followed by an exploration of the pairwise relationship between the variables and wine quality. This tells us about the within-variable relationship changes with quality.

After analyzing the distributions, an evaluation is made to see how variables relate to each other using a heat map. Based on these observations, more detailed explorations can be made using scatterplots and 3d plots. The visual explorations help to narrow down the 3 most important variables necessary for modelling.

Upon narrowing down the 3 variables, we can finally build a model that will be used for predictive analysis. Because the model may not necessarily perform best using only the most essential factors affecting wine quality, it is imperative to include a few other underlying factors in order to make the model more robust for predictive analysis, such that it generalizes data at its best.

Table 1: List of variables used

fixed.acidity	chlorides	pH
volatile.acidity	free.sulfur.dioxide	sulphates
citric.acid	total.sulfur.dioxide	alcohol
residual.sugar	density	quality

Table 2: Variable Means

Variables	**Mean**
fixed.acidity	2.86
volatile.acidity	0.72
citric.acid	0.45
residual.sugar	1.48
chlorides	0.28
free.sulfur.dioxide	3.78
total.sulfur.dioxide	6.35
density	0.998
pH	1.82
sulphates	0.79
alcohol	3.21
quality	5.62

2. Methodology

This data set contains 1599 observations, 11 variables, and 1 target variable which is quality. A list of all variables is shown in Table 1.

Before the analysis is begun, it is essential to run a normality test, make necessary corrections, and remove all outliers from the data set. This will contribute to a better evaluation of the data. After removing the outliers, the data set is left with 1282 observations.

```
mvn(Vino)

## $multivariateNormality
##              Test        Statistic p value Result
## 1 Mardia Skewness 24395.9806979268       0     NO
## 2 Mardia Kurtosis 156.018913713532       0     NO
## 3             MVN             <NA>    <NA>     NO
##
## $univariateNormality
##            Test             Variable Statistic   p value Normality
## 1  Shapiro-Wilk    fixed.acidity        0.9420  <0.001      NO
## 2  Shapiro-Wilk   volatile.acidity      0.9743  <0.001      NO
## 3  Shapiro-Wilk     citric.acid         0.9553  <0.001      NO
## 4  Shapiro-Wilk    residual.sugar       0.5661  <0.001      NO
## 5  Shapiro-Wilk      chlorides          0.4842  <0.001      NO
## 6  Shapiro-Wilk free.sulfur.dioxide     0.9018  <0.001      NO
## 7  Shapiro-Wilk total.sulfur.dioxide    0.8732  <0.001      NO
## 8  Shapiro-Wilk       density           0.9909  <0.001      NO
## 9  Shapiro-Wilk          pH             0.9935  <0.001      NO
## 10 Shapiro-Wilk      sulphates          0.8330  <0.001      NO
## 11 Shapiro-Wilk       alcohol           0.9288  <0.001      NO
## 12 Shapiro-Wilk       quality           0.8576  <0.001      NO
##
```

Figure 1: MVN Results

MVN Corrections:

The result of the normality test leads to the decision to transform all of the features (not including the target feature “quality”) with a square root transformation.

Outlier Removal:

Instances beyond the 1.5 Interquartile Ranges (IQR) were considered as outliers and removed. This removal was performed on each feature (not including the target “quality” feature).

3. Results

3.1 Univariate Analysis

"Univariate analyses are used extensively in analyzing quality of a research. Univariate analysis is defined as analysis carried out on only one ("uni") variable ("variate") to summarize or describe the variable (Babbie, 2007; Trochim, 2006). However, another use of the term "univariate analysis" exists and refers to statistical analyses that involve only one dependent variable and which are used to test hypotheses and draw inferences about populations based on samples, also referred to as univariate statistics (Tabachnick & Fidell, 2007). In this work, a distribution plot was used to perform the univariate analysis." (Sandilands, 2006.)

The first analysis work done is investigating the distribution of the data. A distribution plot is made for each variable to see the variation of data within each of the variables.

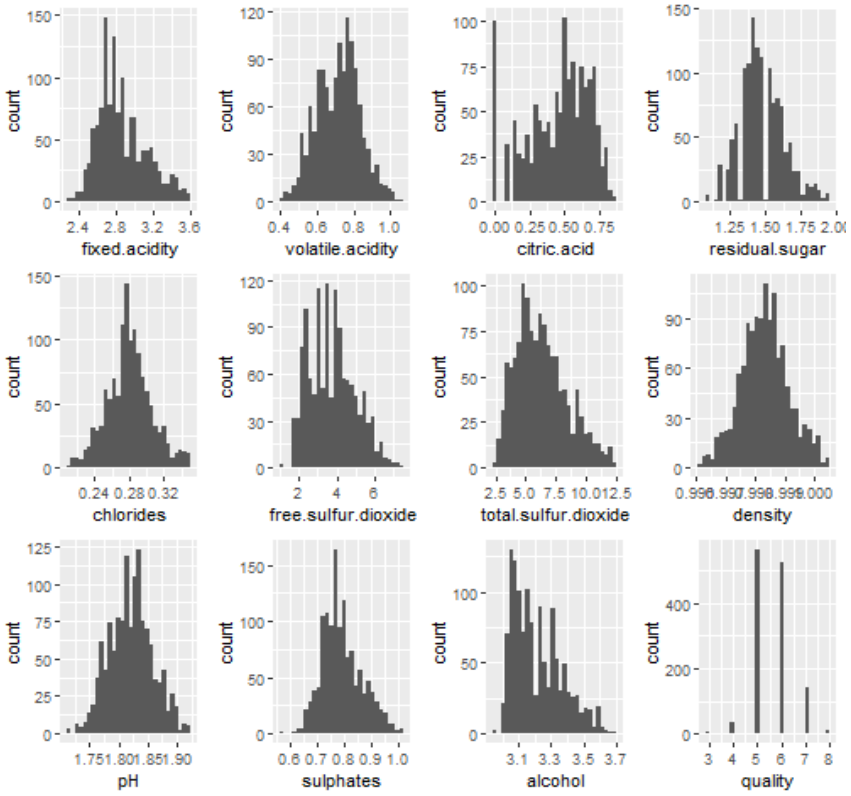


Figure 2: Distribution Plot

After the square root transformation, a few features are left with a slight skewed normal distribution for citric.acid, free.sulfur.oxide, total.sulfur.dioxide, and alcohol. Most of the skewed data is positively skewed (right), meaning their median and modal values are to the left of the mean values.

On looking at the distribution plot, we can tell how much variation is in each of our variables. The larger the spread in the data, the easier it is to distinguish the quality of the wine. If the data does not vary much, it makes it harder to differentiate the quality. The second distribution plots also show in detail how much the values are spread.

A better visual of how the variables vary using quantitative comparison is through their standard deviations. In the figure below, density has the least spread in data, followed by chlorides. This means that the quality of wine is likely not able to be differentiated easily by any change in either of these two variables.

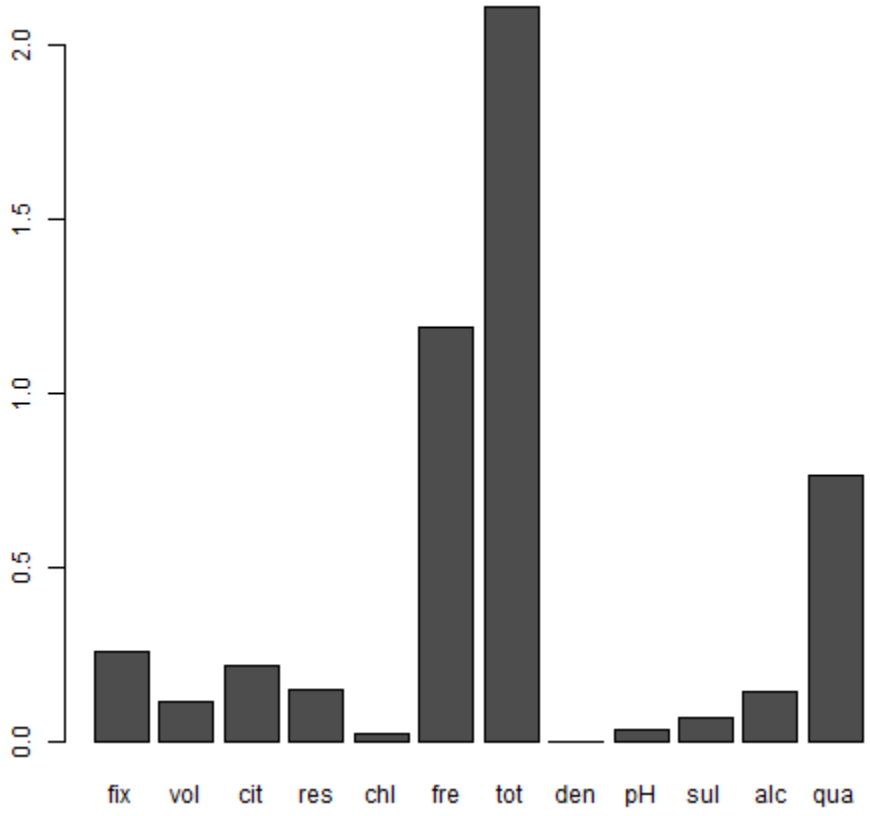


Figure 3: Standard Deviation Plot

3.2 Bivariate Analysis

Bivariate analysis is the simultaneous analysis of two variables (*Beukelman and Brunner, 2016).* It explores the concept of relationship between two variables, whether there exists an association and the strength of this association, or whether there are differences between two variables and the significance of these differences. **Figure 4** shows the correlation plot used to perform the bivariate analysis on the data.

Next, a bivariate analysis is done where the relationship between each variable and quality is analyzed to determine their correlations.

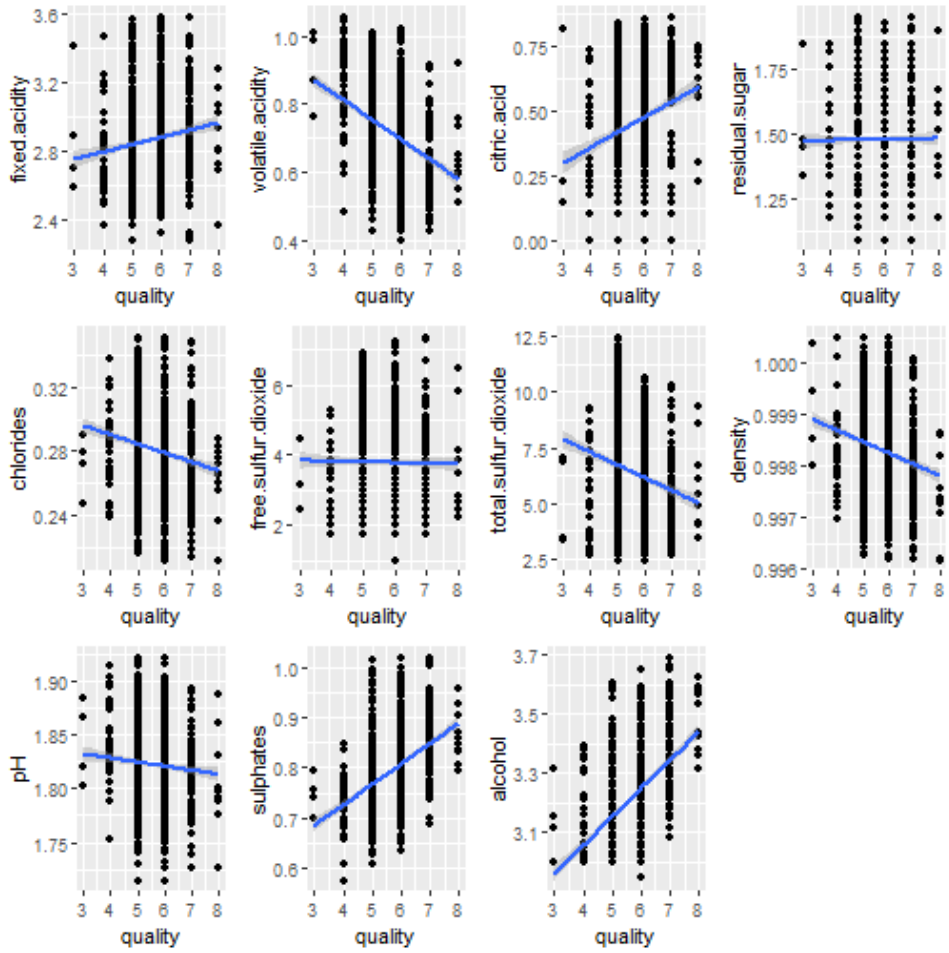


Figure 4: Correlation Plot 1

From the correlation plots, the variables that show the highest correlations with quality are volatile.acidity, sulphates, and alcohol. This can be seen from the slopes of the lines. This may not be the most effective way to view correlation, hence a quantitative method is used for better analysis.

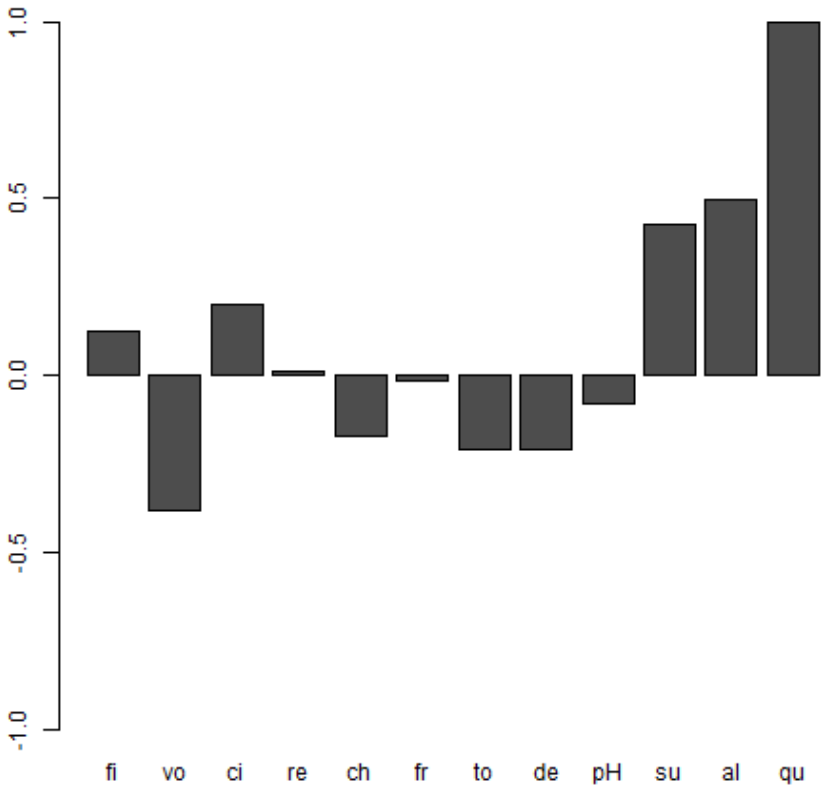


Figure 5: Correlation Plot 2

As previously mentioned, volatile.acidity, sulphates and alcohol show highest correlation on the bar graph. Volatile.acidity shows a negative correlation with quality, which means that as one increases, the other decreases. Sulphates and alcohol show positive correlation.

3.3 Multivariate Analysis

“Multivariate analysis deals with the statistical analysis of data collected on more than one dependent variable (*Olkin and Sampson, 2001*). These variables may be correlated with each other, and their statistical dependence is often taken into account when analyzing such data. This consideration of statistical dependence makes multivariate analysis different in approach and considerably more complex than the corresponding univariate analysis, when there is only one response variable under consideration.” (Nikita, n.d.)

3.3.1 Heat Map

Since prior analysis showed that volatile.acidity, sulphates and alcohols varied most with quality, a multivariate analysis will be done to see how these variables together affect quality. First a heat map is done to check the correlation of all variables, followed by correlation plots of pairs of the significant variables, having a third variable as a hue. This will be our primary approach being used for our multivariate analysis.

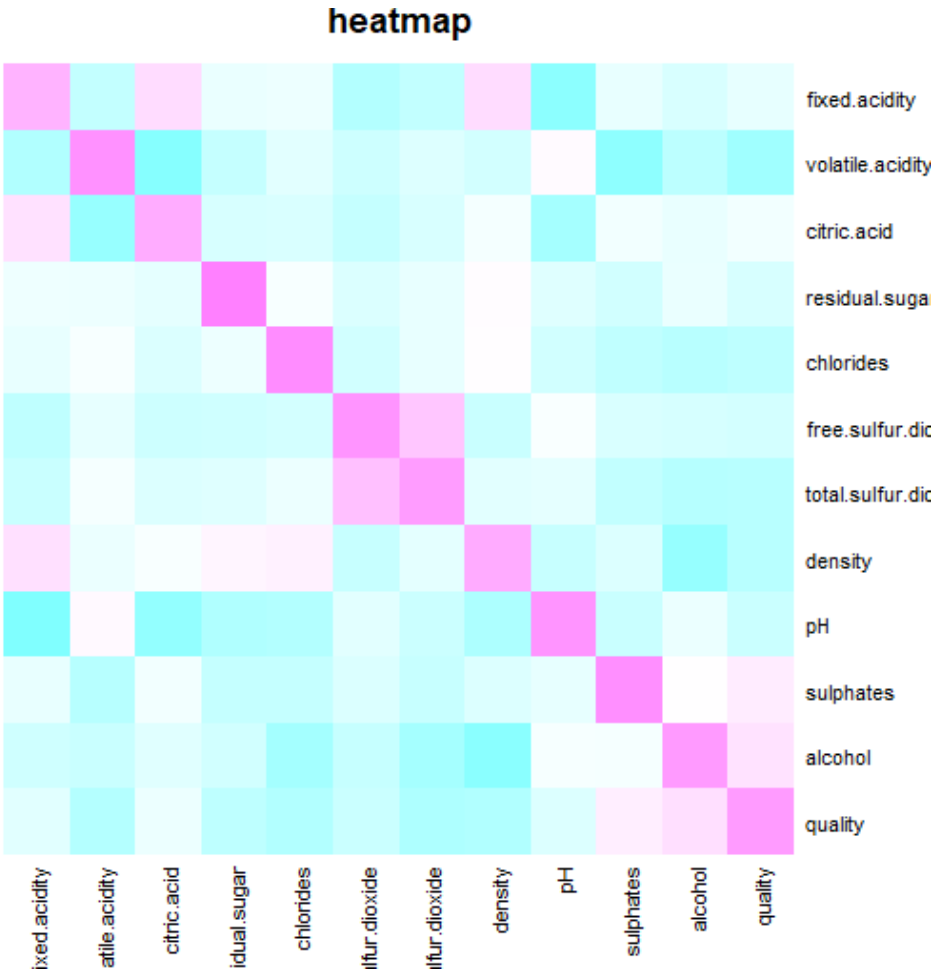


Figure 6: Heat Map

The heat map shows moderate correlations between fixed.acidity and citric acid, fixed.acidity and density, free.sulfur.dioxide and total.sulfur.dioxide, sulphates and quality, and finally alcohol and quality.

*3.3.2 **Scatterplots***

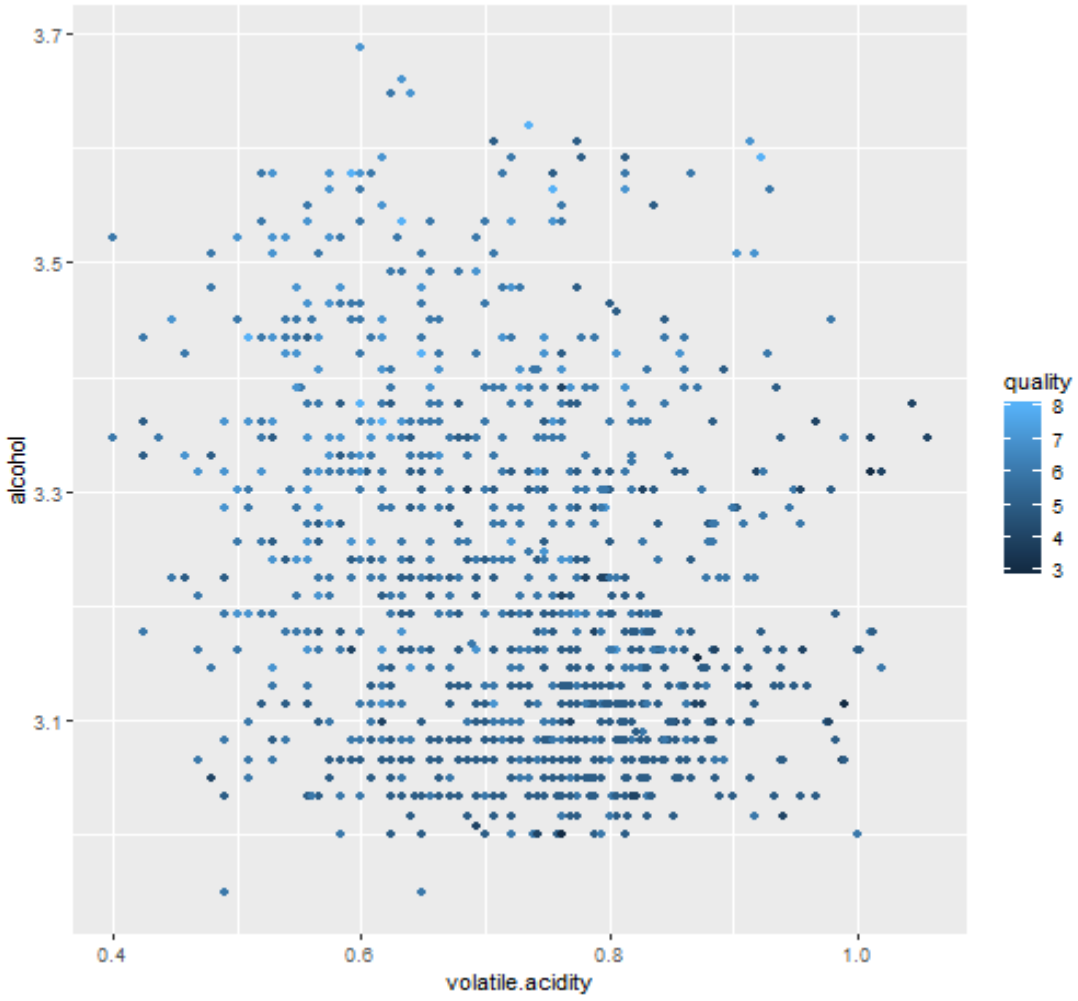


Figure 7: Scatterplot (alcohol, volatile.acidity, quality)

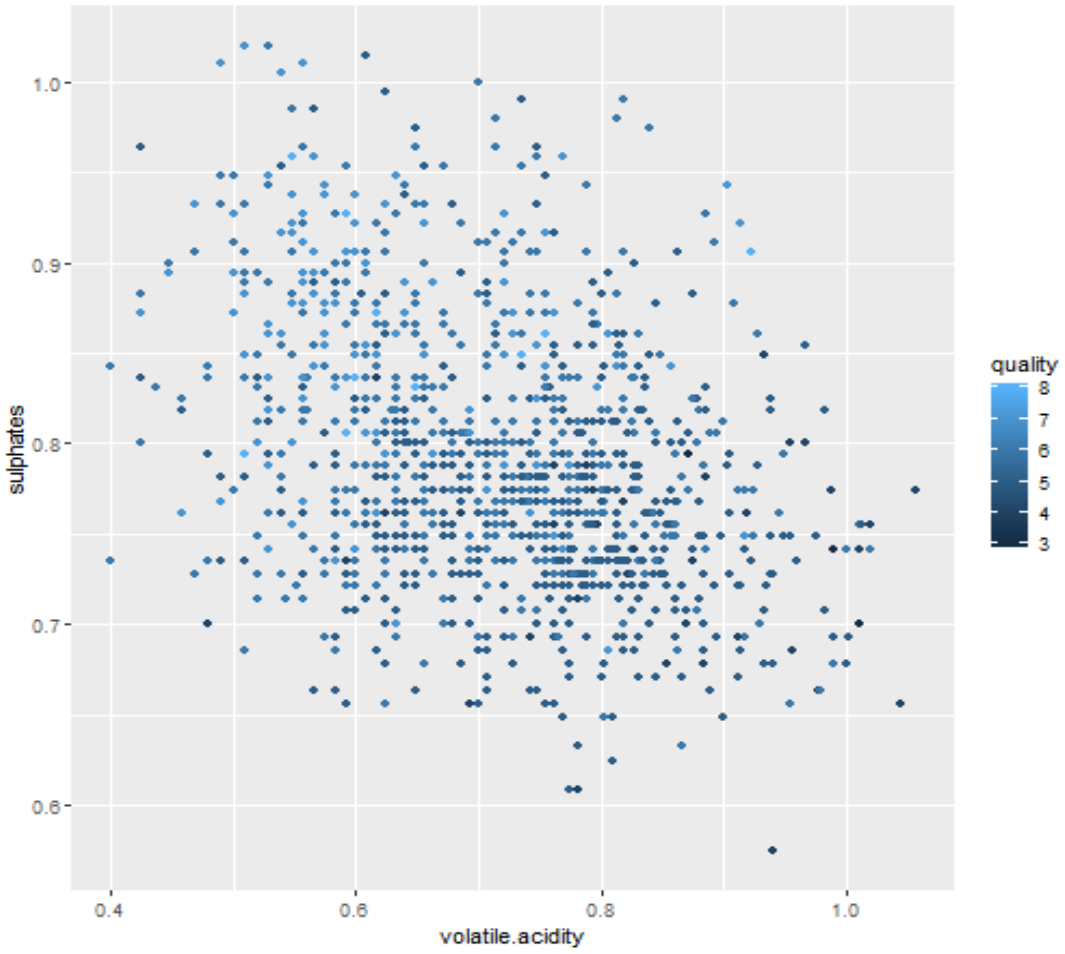


Figure 8: Scatterplot (sulphates, volatile.acidity, quality)

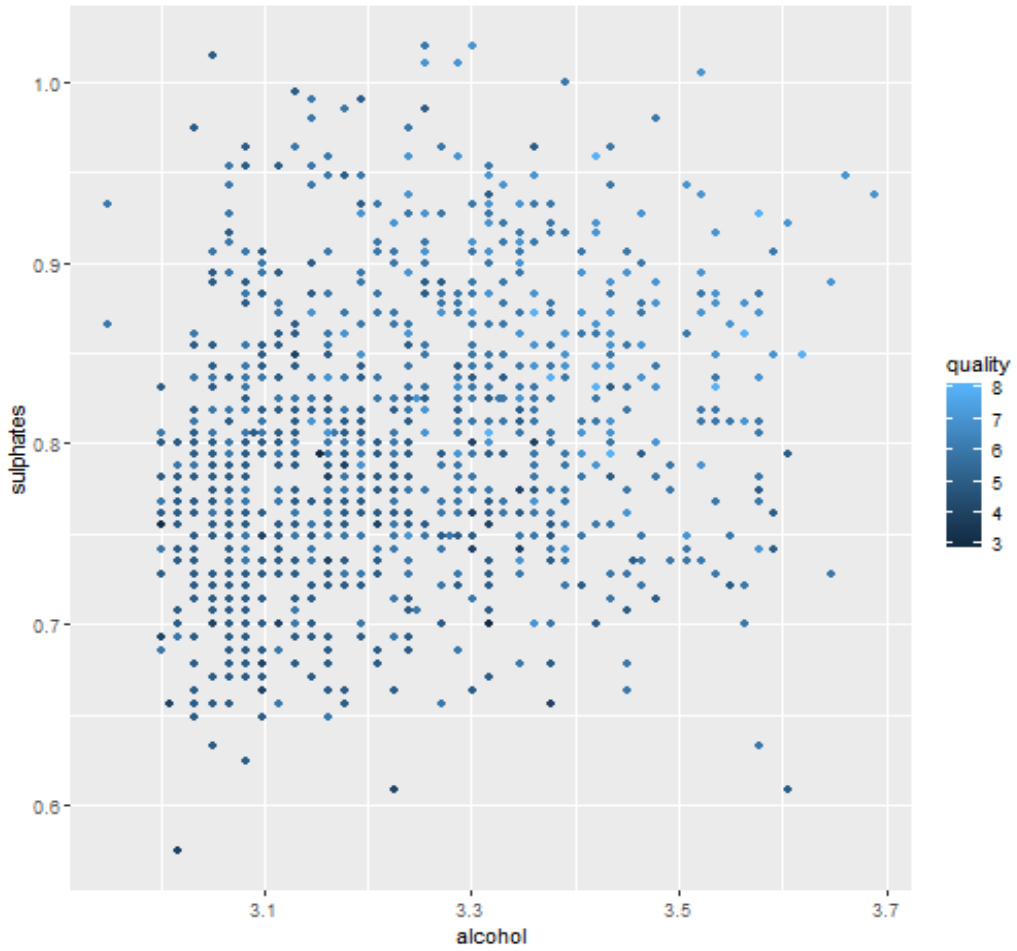


Figure 9: Scatterplot (sulphates, alcohol, quality)

The scatterplots above show how quality changes as the other variables vary. From the scatterplot between alcohol and volatile.acidity, the quality of wine is increasing as alcohol content increases but with volatile acidity mostly below about 0.6. The quality of wine drops when alcohol content is lower and volatile.acidity is mostly above 0.6. This can be seen from the change in color variation, where the light blue color represents higher quality wine, and the deep blue color represents lower quality wine.

The scatterplot between sulphates and volatile acidity shows almost the same relationship as with alcohol and volatile.acidity. The quality of alcohol increases for higher values of sulphates and lower values of volatile.acidity. The blue color darkens as the dots move towards the bottom

right. These two situations agree with our previous analysis which showed positive correlations between sulphates and quality, and alcohol and quality, while volatile.acidity showed a negative correlation with quality.

In the third correlation graph between sulphates and alcohol, the blue color lightens as the dots move to the right and upwards. The quality increases with increasing alcohol content and sulphate concentration.

3.3.3 Variation between the 3 Variables

Figures 10, 11, and 12 show 3D plots of variation between the three variables. The figures show how the three factors vary with each other. The plots vary along the x, y and z axes with respect to the variables. The 3D plots confirm the results from the scatterplots. Thus far, all the analysis has been very useful in narrowing down the variables to produce only the most vital factors needed to predict wine quality. The three factors that have shown to be most useful are *volatile.acidity*, *sulphates* and *alcohol*, and with multiple investigations carried out, it is fairly easy to see that the conclusion is reasonable. Furthermore, it will be a good idea to carry out a performance/predictive analysis using one of the useful algorithms for this purpose, to see how well we can predict the quality of red wine given the 3 mentioned critical factors.

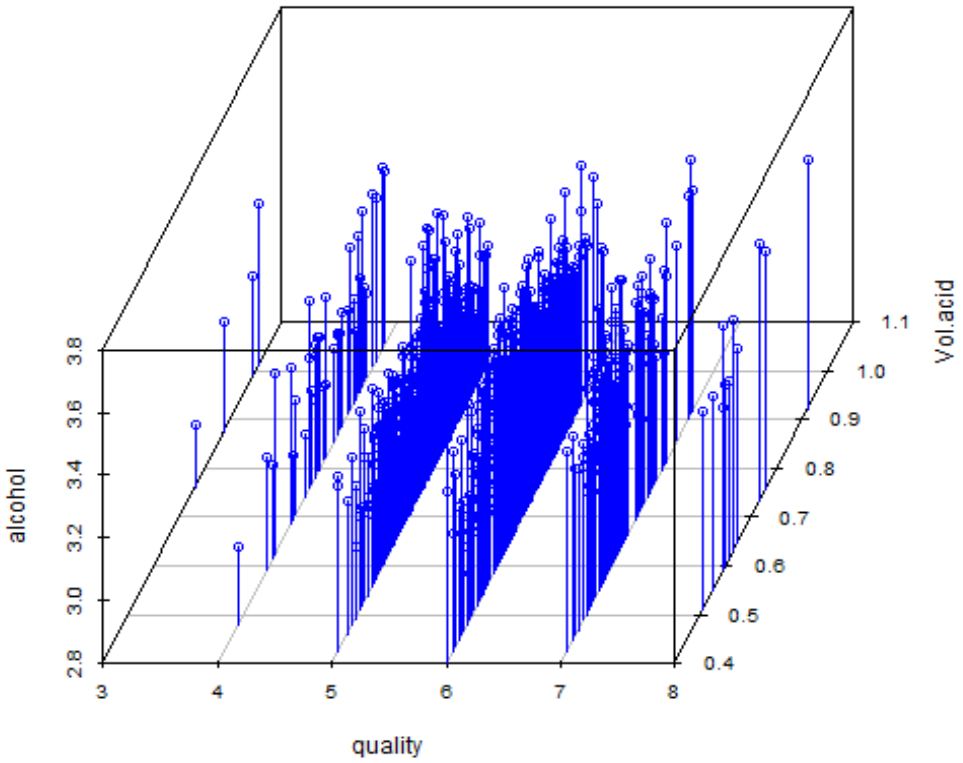


Figure 10: 3D Plot (alcohol, quality, volatile.acidity)

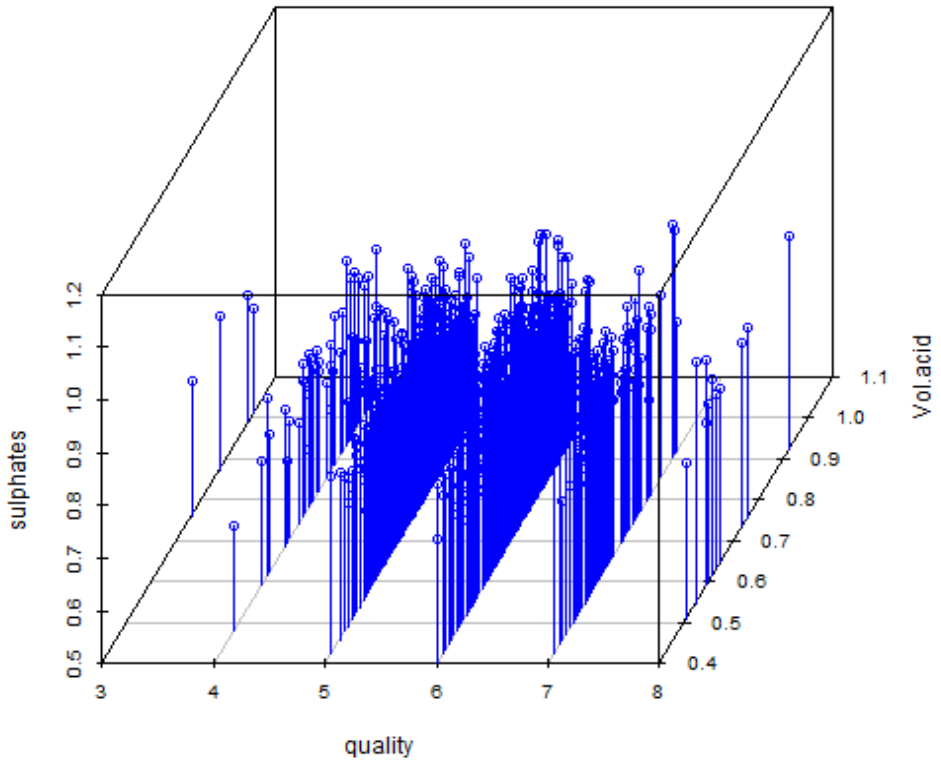


Figure 11: 3D Plot (sulphates, quality, volatile.acidity)

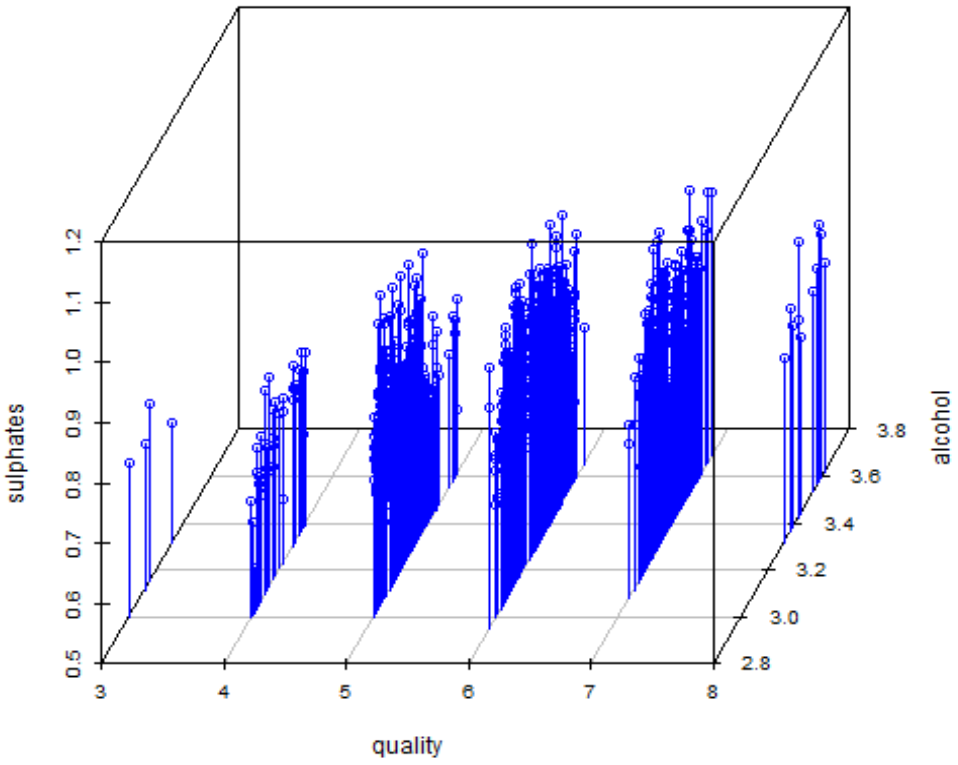


Figure 12: 3D Plot (sulphates, quality, alcohol)

3.4 Linear Model Formulating

To predict the quality of red wine using the above critical factors, data was fitted to a linear model.

3.4.1 Fitting

The fitting process shown below begins with an initial model composed of the 3 most indicative features contributing to quality found in the above analysis. To increase the coefficient of determination and decrease residual standard error, the next step is to add features to the model that had moderate correlation. Adding in the right features works as planned giving the model the complexity it needs to make better predictions. The final step is checking the interactions between features to create an even more robust model with a step algorithm. According to the resulting AIC values, certain interactions do provide a better model such as volatile.acidity and total.sulfur.dioxide. The final model chosen by the step function will be the model used in prediction. This model will also be used in a Binomial Logistic Regression (BLR) later to provide more consistent results about quality.

3.4.2 Assumptions

The data failed normality testing, so the central limit theorem was applied to use the linear model method. After the linear model is created, it is necessary to check the residuals for normality. After checking (plots are shown below) it is confirmed that the residuals sum to about zero and that the tails are fairly even.

3.4.3 Performance

The performance of the linear and BLR models will be calculated based on confusion matrix results. The linear model is made to correctly identify each wine's quality score, while the BLR is made to classify each wine as good or bad. The comparison of train and test scores in both models is what typically is expected, with the train score slightly outperforming the test score. This is an indication that the models are not over fitting what it is fed during training, and can generalize well to new information. With train and test scores of about 63% and 61% respectively, the linear model is not reliable enough. It is important to consider the fact that wine scores likely have biases present as the graded wine qualities come from individuals' personal

opinions about the wine. There is not too much information about how official the wine scores are. The potential problems with the scores and low performance with the model led to the decision to try a Binomial Logistic Regression model. For the BLR model, scores below or equal to 5 were considered bad quality, anything above 5 were good quality. The BLR model did well with train and test at about 76% and 75% respectively. Hence, given the chemical components of red wine this model can predict if it is low or high quality. Fitting, Assumption and Performance details are shown below.

Fitting, Assumption and Performance details:

Linear Model 1: Contains alcohol, sulphates and volatile.acidity.

```
## Call:
## lm(formula = quality ~ alcohol + sulphates + volatile.acidity,
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.30998 -0.36116 -0.07408  0.44724  1.96259
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -2.0584     0.5077  -4.054 5.44e-05 ***
## alcohol            1.9925     0.1363  14.621  < 2e-16 ***
## sulphates          2.8114     0.2896   9.708  < 2e-16 ***
## volatile.acidity  -1.3170     0.1790  -7.359 3.98e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6138 on 957 degrees of freedom
## Multiple R-squared:  0.3835, Adjusted R-squared:  0.3816
## F-statistic: 198.5 on 3 and 957 DF,  p-value: < 2.2e-16
```

R^2 is at a low value of 0.38 and residual standard error is 0.61 which is high. For lower error and higher R^2, add in some of the less correlated items to give more complexity.

Linear Model 2: Contains alcohol, sulphates, volatile.acidity, citric.acid, chlorides, and total.sulfur.dioxide.

```
summary(m1)

##
## Call:
## lm(formula = quality ~ alcohol + sulphates + volatile.acidity +
##     citric.acid + chlorides + total.sulfur.dioxide, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.40608 -0.36511 -0.05915  0.41775  1.92985
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)

## (Intercept)          -1.47285    0.61726  -2.386   0.0172 *
## alcohol               1.88658    0.14459  13.048  < 2e-16 ***
## sulphates             2.86540    0.28982   9.887  < 2e-16 ***
## volatile.acidity     -1.39783    0.22261  -6.279 5.16e-10 ***
## citric.acid          -0.09824    0.11411  -0.861   0.3895
## chlorides            -0.09908    0.80205  -0.124   0.9017
## total.sulfur.dioxide -0.02488    0.01005  -2.476   0.0135 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6122 on 954 degrees of freedom
## Multiple R-squared:  0.3886, Adjusted R-squared:  0.3848
## F-statistic: 101.1 on 6 and 954 DF,  p-value: < 2.2e-16
```

R^2 is 0.38 and residual standard error is 0.61. The R^2 is still considerably low and the standard error is still high. We will then check for interaction between terms with a stepwise algorithm.

Linear Model 3: Contains alcohol, sulphates, volatile.acidity, citric.acid, chlorides, and total.sulfur.dioxide

```
m2= step(m1, scope=list(upper= ~alcohol*sulphates*volatile.acidity*citric.aci
d*chlorides*total.sulfur.dioxide, lower= ~1))
```

```
## Start:  AIC=-936.2
## quality ~ alcohol + sulphates + volatile.acidity + citric.acid +
##     chlorides + total.sulfur.dioxide
##
##                                        Df Sum of Sq    RSS     AIC
## + sulphates:total.sulfur.dioxide        1     7.552 349.97 -954.72
## + volatile.acidity:total.sulfur.dioxide 1     5.311 352.22 -948.59
## + alcohol:sulphates                     1     4.615 352.91 -946.69
## + citric.acid:total.sulfur.dioxide      1     4.121 353.41 -945.34
## - chlorides                             1     0.006 357.53 -938.19
## + volatile.acidity:citric.acid          1     1.365 356.16 -937.88
## - citric.acid                           1     0.278 357.80 -937.46
## <none>                                              357.53 -936.20
## + sulphates:chlorides                   1     0.643 356.88 -935.93
## + citric.acid:chlorides                 1     0.336 357.19 -935.11
## + alcohol:chlorides                     1     0.302 357.22 -935.01
## + chlorides:total.sulfur.dioxide        1     0.212 357.31 -934.77
## + sulphates:volatile.acidity            1     0.158 357.37 -934.63
## + alcohol:total.sulfur.dioxide          1     0.071 357.46 -934.40
## + alcohol:citric.acid                   1     0.051 357.48 -934.34
## + sulphates:citric.acid                 1     0.026 357.50 -934.27
## + alcohol:volatile.acidity              1     0.008 357.52 -934.23
## + volatile.acidity:chlorides            1     0.004 357.52 -934.22
## - total.sulfur.dioxide                  1     2.298 359.82 -932.05
## - volatile.acidity                      1    14.777 372.30 -899.28
## - sulphates                             1    36.633 394.16 -844.46
## - alcohol                               1    63.801 421.33 -780.41
##
## Step:  AIC=-954.72
## quality ~ alcohol + sulphates + volatile.acidity + citric.acid +
##     chlorides + total.sulfur.dioxide + sulphates:total.sulfur.dioxide
```

```
##
##                                        Df Sum of Sq    RSS     AIC
## + alcohol:sulphates                     1     3.207 346.77 -961.57
## + volatile.acidity:total.sulfur.dioxide 1     2.120 347.85 -958.56
## + citric.acid:total.sulfur.dioxide      1     1.618 348.36 -957.17
## - chlorides                             1     0.001 349.98 -956.72
## + volatile.acidity:citric.acid          1     1.178 348.80 -955.96
## - citric.acid                           1     0.410 350.38 -955.59
## <none>                                              349.97 -954.72
## + alcohol:total.sulfur.dioxide          1     0.180 349.79 -953.21
## + alcohol:chlorides                     1     0.178 349.80 -953.21
## + sulphates:volatile.acidity            1     0.133 349.84 -953.08
## + chlorides:total.sulfur.dioxide        1     0.125 349.85 -953.06
## + sulphates:citric.acid                 1     0.081 349.89 -952.94
## + citric.acid:chlorides                 1     0.078 349.90 -952.93
## + alcohol:citric.acid                   1     0.014 349.96 -952.76
## + sulphates:chlorides                   1     0.012 349.96 -952.75
## + volatile.acidity:chlorides            1     0.004 349.97 -952.73
## + alcohol:volatile.acidity              1     0.000 349.97 -952.72
## - sulphates:total.sulfur.dioxide        1     7.552 357.53 -936.20
## - volatile.acidity                      1    14.126 364.10 -918.69
## - alcohol                               1    67.514 417.49 -787.20
##
## Step:  AIC=-961.57
## quality ~ alcohol + sulphates + volatile.acidity + citric.acid +
##     chlorides + total.sulfur.dioxide + sulphates:total.sulfur.dioxide +
##     alcohol:sulphates
##
```

```
##                                        Df Sum of Sq    RSS     AIC
## + volatile.acidity:total.sulfur.dioxide 1    2.5527 344.21 -966.67
## + citric.acid:total.sulfur.dioxide      1    1.6774 345.09 -964.23
## - chlorides                             1    0.0233 346.79 -963.50
## + volatile.acidity:citric.acid          1    1.0619 345.71 -962.51
## - citric.acid                           1    0.6313 347.40 -961.82
## <none>                                             346.77 -961.57
## + sulphates:volatile.acidity            1    0.3919 346.38 -960.65
## + sulphates:chlorides                   1    0.3378 346.43 -960.50
## + alcohol:volatile.acidity              1    0.2092 346.56 -960.15
## + alcohol:citric.acid                   1    0.1461 346.62 -959.97
## + alcohol:chlorides                     1    0.1439 346.62 -959.97
## + volatile.acidity:chlorides            1    0.0540 346.71 -959.72
## + citric.acid:chlorides                 1    0.0224 346.75 -959.63
## + chlorides:total.sulfur.dioxide        1    0.0220 346.75 -959.63
## + sulphates:citric.acid                 1    0.0061 346.76 -959.58
## + alcohol:total.sulfur.dioxide          1    0.0006 346.77 -959.57
## - alcohol:sulphates                     1    3.2072 349.97 -954.72
## - sulphates:total.sulfur.dioxide        1    6.1436 352.91 -946.69
## - volatile.acidity                      1   14.3297 361.10 -924.65
##
## Step:  AIC=-966.67
```

```
## quality ~ alcohol + sulphates + volatile.acidity + citric.acid +
##     chlorides + total.sulfur.dioxide + sulphates:total.sulfur.dioxide +
##     alcohol:sulphates + volatile.acidity:total.sulfur.dioxide
##
##                                        Df Sum of Sq    RSS     AIC
## - chlorides                             1    0.0091 344.22 -968.64
## + volatile.acidity:citric.acid          1    0.7829 343.43 -966.86
## <none>                                               344.21 -966.67
## + alcohol:volatile.acidity              1    0.5582 343.66 -966.23
## + citric.acid:total.sulfur.dioxide      1    0.4180 343.80 -965.83
## + sulphates:chlorides                   1    0.3428 343.87 -965.62
## - citric.acid                           1    1.1368 345.35 -965.50
## + alcohol:chlorides                     1    0.2833 343.93 -965.46
## + alcohol:citric.acid                   1    0.2760 343.94 -965.44
## + sulphates:volatile.acidity            1    0.2625 343.95 -965.40
## + volatile.acidity:chlorides            1    0.2385 343.98 -965.33
## + chlorides:total.sulfur.dioxide        1    0.1631 344.05 -965.12
## + alcohol:total.sulfur.dioxide          1    0.0394 344.18 -964.78
## + citric.acid:chlorides                 1    0.0212 344.19 -964.73
## + sulphates:citric.acid                 1    0.0141 344.20 -964.71
## - volatile.acidity:total.sulfur.dioxide 1    2.5527 346.77 -961.57
## - sulphates:total.sulfur.dioxide        1    3.1324 347.35 -959.96
## - alcohol:sulphates                     1    3.6396 347.85 -958.56
##
```

```
## Step:  AIC=-968.64
## quality ~ alcohol + sulphates + volatile.acidity + citric.acid +
##     total.sulfur.dioxide + sulphates:total.sulfur.dioxide + alcohol:sulpha
tes +
##     volatile.acidity:total.sulfur.dioxide
##
##                                        Df Sum of Sq    RSS     AIC
## + volatile.acidity:citric.acid          1    0.7722 343.45 -968.80
## <none>                                               344.22 -968.64
## + alcohol:volatile.acidity              1    0.5453 343.68 -968.16
## + citric.acid:total.sulfur.dioxide      1    0.4187 343.81 -967.81
## - citric.acid                           1    1.1314 345.36 -967.49
## + sulphates:volatile.acidity            1    0.2705 343.95 -967.40
## + alcohol:citric.acid                   1    0.2605 343.96 -967.37
## + alcohol:total.sulfur.dioxide          1    0.0357 344.19 -966.74
## + sulphates:citric.acid                 1    0.0140 344.21 -966.68
## + chlorides                             1    0.0091 344.21 -966.67
## - volatile.acidity:total.sulfur.dioxide 1    2.5668 346.79 -963.50
## - sulphates:total.sulfur.dioxide        1    3.1239 347.35 -961.96
## - alcohol:sulphates                     1    3.6313 347.86 -960.56
##
## Step:  AIC=-968.8
## quality ~ alcohol + sulphates + volatile.acidity + citric.acid +
##     total.sulfur.dioxide + sulphates:total.sulfur.dioxide + alcohol:sulpha
tes +
##     volatile.acidity:total.sulfur.dioxide + volatile.acidity:citric.acid
```

```
##                                        Df Sum of Sq    RSS     AIC
## <none>                                              343.45 -968.80
## - volatile.acidity:citric.acid          1    0.7722 344.22 -968.64
## + citric.acid:total.sulfur.dioxide      1    0.5352 342.92 -968.30
## + alcohol:volatile.acidity              1    0.4676 342.98 -968.11
## + alcohol:citric.acid                   1    0.1669 343.28 -967.27
## + sulphates:citric.acid                 1    0.1040 343.35 -967.09
## + sulphates:volatile.acidity            1    0.0746 343.38 -967.01
## + alcohol:total.sulfur.dioxide          1    0.0433 343.41 -966.92
## + chlorides                             1    0.0199 343.43 -966.86
## - volatile.acidity:total.sulfur.dioxide 1    2.2945 345.75 -964.40
## - sulphates:total.sulfur.dioxide        1    3.1429 346.59 -962.05
## - alcohol:sulphates                     1    3.4874 346.94 -961.09
```

```
summary(m2)
```

```
##
## Call:
## lm(formula = quality ~ alcohol + sulphates + volatile.acidity +
##     citric.acid + total.sulfur.dioxide + sulphates:total.sulfur.dioxide +
##     alcohol:sulphates + volatile.acidity:total.sulfur.dioxide +
##     volatile.acidity:citric.acid, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.31627 -0.33864 -0.09772  0.41693  1.98625
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|
)
## (Intercept)                           12.16248    4.85068   2.507  0.0123
3 *
## alcohol                               -2.47446    1.42628  -1.735  0.0830
8 .
## sulphates                            -12.43240    5.95505  -2.088  0.0370
9 *
## volatile.acidity                      -3.45172    0.72184  -4.782 2.01e-0
6 ***
## citric.acid                           -1.18985    0.68616  -1.734  0.0832
3 .
## total.sulfur.dioxide                   0.12691    0.14491   0.876  0.3813
7
## sulphates:total.sulfur.dioxide        -0.41035    0.13910  -2.950  0.0032
6 **
## alcohol:sulphates                      5.53370    1.78077   3.107  0.0019
4 **
## volatile.acidity:total.sulfur.dioxide  0.22374    0.08876   2.521  0.0118
8 *
## volatile.acidity:citric.acid           1.31967    0.90250   1.462  0.1440
```

```
1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.601 on 951 degrees of freedom
## Multiple R-squared:  0.4127, Adjusted R-squared:  0.4071
## F-statistic: 74.25 on 9 and 951 DF,  p-value: < 2.2e-16
```

In this model, error is minimized at 0.6 and R^2 is maximized at 0.4. Although these are not great results, they are slightly better than results in previous models. Future recommendations may entail trial and error with other variables or less variables. In this case, linear model 3 (m2) is our final model.

Our assumptions are checked through plots on the residuals to check normality and constant variance.

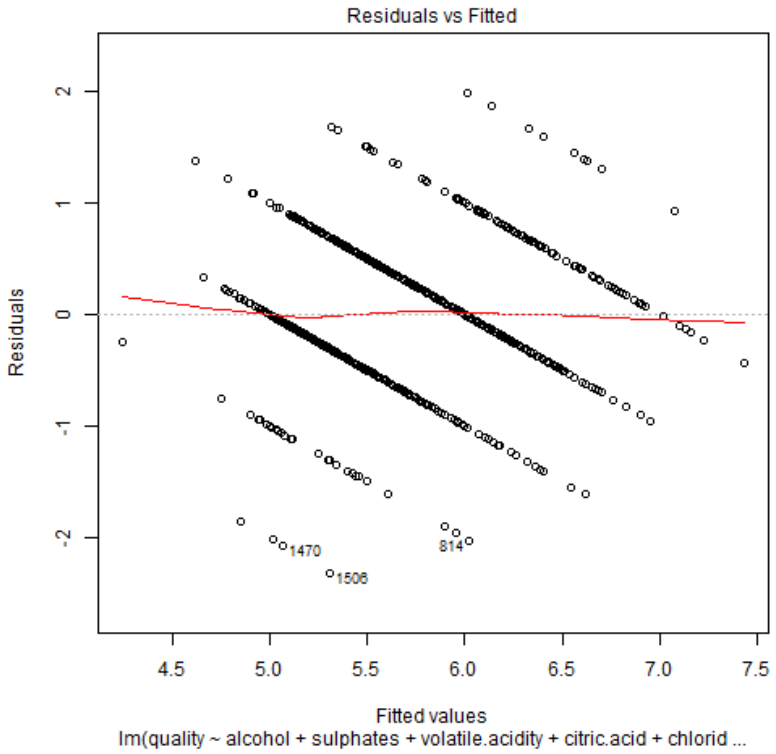


Figure 13: Residuals vs Fitted Plot

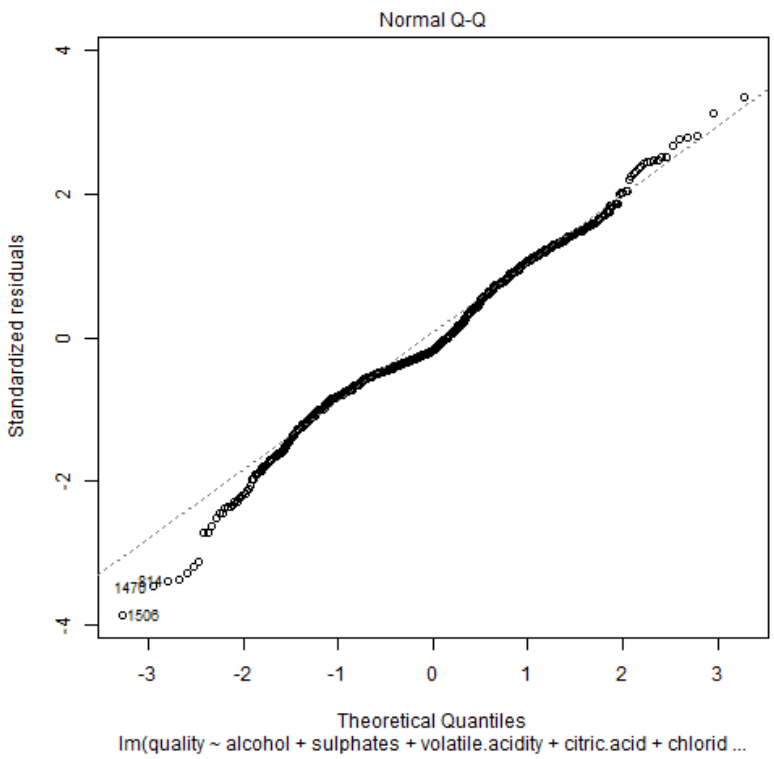


Figure 14: Normal Q-Q Plot

The first plot shows constant variance and the residuals are centered around 0. This does not violate the constant variance assumption. In the second plot, the residuals are aligned along the qq-line and their tails, though slightly deviated, considerably even out on each end. We can say that the normality assumption here is met.

We will now check the performance of the linear model on the classifier. The data is split into training and test sets, to first train the model, then test how well it generalizes for new data not previously seen. The confusion matrices are results from the classifier and represent how often the predicted results are able to mimic the true outputs. The leading diagonal represents the correctly predicted outputs, while everything else shows incorrect predictions.

Linear Model Performance using the training set confusion matrix.

Train confusion matrix:

```
          Reference
Prediction   3   4   5   6   7   8
         3   0   0   0   0   0   0
         4   0   1   0   0   0   0
         5   4  21 314 117   2   0
         6   0   4 103 258  73   6
         7   0   0   2  16  35   5
         8   0   0   0   0   0   0
```

Train accuracy: `0.6326743`

Accuracy result for training set is 0.632 **(63%)**

Linear Model Performance using the test set confusion matrix.

Test confusion matrix:

```
          Reference
Prediction   4   5   6   7   8
         4   0   0   0   0   0
         5   9 100  43   2   0
         6   2  45  82  18   1
         7   0   0   6  13   0
         8   0   0   0   0   0
```

Test Acc: `0.6074766`

Accuracy result for test set is 0.607 **(61%)**

Binomial Logistic Regression

As previously mentioned, a binomial logistic regression will be applied to the linear model by first converting the linear model to a generalized linear model (GLM).

BLR Model: Previous linear model is converted to GLM for BLR analysis.

```
#BLR
model_glm= glm(category ~alcohol + sulphates + volatile.acidity +
    citric.acid + chlorides + total.sulfur.dioxide + sulphates:total.sulfur.d
ioxide +
    volatile.acidity:total.sulfur.dioxide + alcohol:sulphates, data = train,
family=binomial(link = "logit"))
summary(model_glm)

##
## Call:
## glm(formula = category ~ alcohol + sulphates + volatile.acidity +
##     citric.acid + chlorides + total.sulfur.dioxide + sulphates:total.sulfu
r.dioxide +
##     volatile.acidity:total.sulfur.dioxide + alcohol:sulphates,
##     family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5949  -0.7843   0.1719   0.8103   2.2118
##
## Coefficients:
##                                     Estimate Std. Error z value Pr(>|z|
)
## (Intercept)                         13.67547   24.72273   0.553  0.5801
6
## alcohol                             -7.36354    7.46842  -0.986  0.3241
5
## sulphates                          -37.57285   31.24777  -1.202  0.2292
0
## volatile.acidity                    -4.26385    2.55110  -1.671  0.0946
5 .
## citric.acid                         -0.52282    0.45870  -1.140  0.2543
8
## chlorides                           -6.79474    3.37542  -2.013  0.0441
1 *
## total.sulfur.dioxide                 1.21683    0.63334   1.921  0.0547
0 .
## sulphates:total.sulfur.dioxide      -1.80316    0.64925  -2.777  0.0054
8 **
```

```
## volatile.acidity:total.sulfur.dioxide   0.06442    0.37735   0.171  0.8644
5
## alcohol:sulphates                      18.16660    9.59163   1.894  0.0582
2 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1330.58  on 961  degrees of freedom
## Residual deviance:  958.37  on 952  degrees of freedom
## AIC: 978.37
##
## Number of Fisher Scoring iterations: 5
```

Binomial Logistic Regression performance using the training set confusion matrix.

Train CM:

```
           true
predicted     0   1
  Bad Wine  361 137
  Good Wine  93 371
```

Train Acc: `0.7609148`

Accuracy result for training set is **0.76 (76%)**

Binomial Logistic Regression performance using the test set confusion matrix.

Test CM:

```
           true
predicted     0   1
  Bad Wine  116  44
  Good Wine  35 125
```

Test Acc: `0.753125`

Accuracy result for test set is **0.75 (75%)**

4. Summary and Discussion

This project utilized a step-by-step method to present the factors that most contributed to the quality of wine, followed by a predictive analysis to see how well these factors, built into a model, can predict the quality of wine. Initially, a univariate analysis was used to determine how well each variable's data could determine the outcome of the analysis. The use of distribution plots and standard deviation plots showed that some variables such as "***density"*** and "***chlorides"*** had the least ability to distinguish quality due to minimal spread in data. A bivariate analysis followed, where the correlation between each variable and quality was measured. Here it could be seen that "***sulphates"***, "***alcohol"*** and "***volatile.acidity"*** correlated best with quality. Although these were the leading factors, their correlation values were still at a low, ranging from 0.3 to 0.5. This could be because the quality of wine is not necessarily determined by single factors, but a combination of several. A multivariate analysis was then conducted to determine the combination of variables needed to define wine quality. This began with a heat map, where correlation among variables was measured, alongside correlation with quality. Here, we could see how some variables depended on one another, such as ***"fixed.acidity*** and ***citric acid",* "fixed.acidity** and ***density", "free.sulfur.dioxide*** and ***total.sulfur.dioxide", "sulphates*** and ***quality"***, and finally ***"alcohol*** and ***quality".*** Although correlation among variables could be seen, it is not an indication that a combination of them all would well determine the quality of wine. Hence, scatter plots and 3D plots involving pairs of variables and quality were made. The greatest variation was seen when ***"alcohol", "sulphates",*** and ***"volatile.acidity"*** were paired in a three-way with quality. Since these variables showed the most confirmation to the previously done bivariate analysis, a linear model approach was used to validate a construct that involved these three variables and quality. The linear model approach involved 3 different models. The initial model comprised only the 3 leading factors but performed poorly. As stated above, this is probably because a perfect model may not only need the best factors, but underlying factors as well that contribute to variation in data and minimizing error. These metrics were measured using standard error and R^2. It was found that when adding the underlying factors such as "***citric.acid", "chlorides",*** and ***"total.sulfur.dioxide",*** including their interactions, the linear

model showed better results. The assumptions for normality and constant variance was checked to ensure that the model was not violating these assumptions. This linear model was then converted to a generalized linear model and used to perform a binomial logistic regression analysis, where the data was used to build a model needed to predict the quality of wine. The first prediction analysis was done to determine what grade each new case of wine would most likely be predicted as, i.e., either ***4,5,6,7***, or ***8***. This yielded a training set performance of 63% and test performance of 61%. Next, the grades were divided into "***good"*** or "***bad"*** wine, where any grade above 5 was good, while the rest were bad. In this case, training performance was at 76% and test perfomance at 75% at best. This basically entails that with a model that comprises only of the factors, ***"alcohol", "sulphates", "volatile.acidity"*** "***citric.acid", "chlorides",*** and ***"total.sulfur.dioxide",*** we are able to correctly predict approximately 7.5 times out of 10, for every new case of wine variables, that the wine will either be good wine or bad wine. Although this is not a perfect result on performance, it is important to note as mentioned earlier, that the grades of the wine quality could have been biased as these results were only got from mere human judgement. This also tells us that despite how previous analyses were pointing towards 3 main factors to adequately determine wine quality, them alone may not be the best indicators of how good or bad a specific wine could be. This must work with a combination of other factors in order to produce more accurate results.

5. Conclusion

Understanding the quality of wine is important for various reasons such as economics, branding, and awards. With the red wine quality data set, it was proven that wine quality is a result of its chemical component levels. However, statistical analysis such as correlation plots and regression illustrated that not all of wine's chemical components are equally impactful. Volatile Acidity, Sulphates, and Alcohol were the most indicative of the quality of a wine, but the remaining chemical components still play an important role in identifying quality. These findings yielded the ability to construct models that consistently predict wine quality.

6. References

1. Charters, S.; Pettigrew, S. The dimensions of wine quality. *Food Qual. Prefer.* 2007, *18*, 997–1007.
2. Verdú Jover, A.J.; Lloréns Montes, F.J.; Fuentes Fuentes, M.D.M. Measuring perceptions of quality in food products: The case of red wine. *Food Qual. Prefer.* 2004, *15*, 453–469.
3. Thach, L. How American Consumers Select Wine. *Wine Bus. Mon.* 2008, *15*, 66–71.
4. Austrian Wine Law, 2015. Available online:http://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen & Gesetzesnumme r=20006757 (accessed on 1 January 2015).
5. Babbie, E. (2007). *The practice of social research*. Belmont, CA: Thomson Wadsworth.
6. Trochim, W. M. (2006). *The research methods knowledge base*. Retrieved from http://www.socialresearchmethods.net/kb/
7. Tabachnick, B. G., & Fidell, L. S. (2007). *Using multivariate statistics*. Boston: Pearson.
8. *Olkin, A.R. Sampson, in International Encyclopedia of the Social & Behavioral Sciences, 2001*
9. https://www.saedsayad.com/bivariate_analysis.htm
10. *Timothy Beukelman, Hermine I. Brunner, in Textbook of Pediatric Rheumatology (Seventh Edition), 2016*

7. Appendix (Code)

Red Wine Analysis

Jamiu Adegbite, Cynthia Atanga, Jasmine Rose Eshun, Stephen Rivera, Mohammad Yawar

5/4/2020

packages

```
library(tidyr)
library(ggplot2)
library(cowplot)

##
## ************************************************************

## Note: As of version 1.0.0, cowplot does not change the

##   default ggplot2 theme anymore. To recover the previous

##   behavior, execute:
##   theme_set(theme_cowplot())

## ************************************************************

library(grid)
library(gridExtra)
library(MVN)

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

## sROC 0.1-2 loaded
```

```
library(scatterplot3d)
library(rgl)

## Warning in rgl.init(initValue, onlyNULL): RGL: unable to open X11 display

## Warning: 'rgl.init' failed, running with 'rgl.useNULL = TRUE'.

library(caTools)
library(lsmeans)

## Loading required package: emmeans

## The 'lsmeans' package is now basically a front end for 'emmeans'.
## Users are encouraged to switch the rest of the way.
## See help('transition') for more information, including how to
## convert old 'lsmeans' objects and scripts to work with 'emmeans'.

library(mda)

## Loading required package: class

## Loaded mda 0.5

library(caret)

## Loading required package: lattice
```

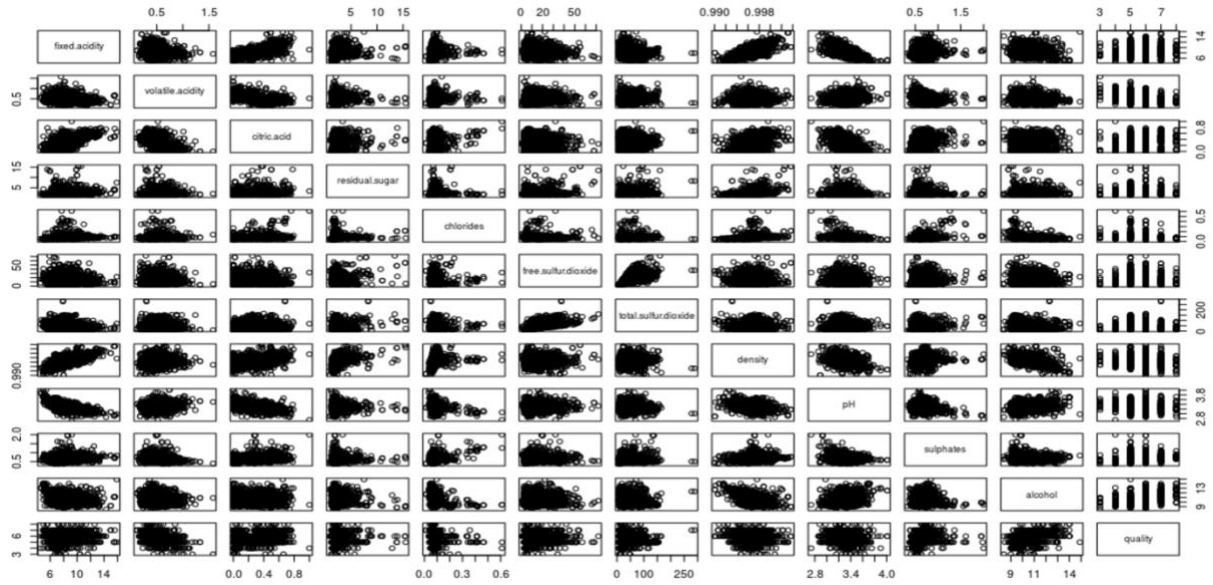
Load in data set from UCI csv file

```
winequality.red=read.csv("winequality-red (3).csv")
Vino=winequality.red
head(Vino)

##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.4             0.70        0.00            1.9     0.076
## 2           7.8             0.88        0.00            2.6     0.098
```

```
## 3           7.8             0.76        0.04            2.3     0.092
## 4          11.2             0.28        0.56            1.9     0.075
## 5           7.4             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  11                  34  0.9978 3.51      0.56     9.4
## 2                  25                  67  0.9968 3.20      0.68     9.8
## 3                  15                  54  0.9970 3.26      0.65     9.8
## 4                  17                  60  0.9980 3.16      0.58     9.8
## 5                  11                  34  0.9978 3.51      0.56     9.4
## 6                  13                  40  0.9978 3.51      0.56     9.4
##   quality
## 1       5
## 2       5
## 3       5
## 4       6
## 5       5
## 6       5
```

Scatterplot Matrix of Data

```
plot(Vino)
```



Check Normality:

```
mvn(Vino)

## $multivariateNormality
##               Test        Statistic p value Result
## 1 Mardia Skewness 24395.9806979268       0     NO
## 2 Mardia Kurtosis 156.018913713532       0     NO
## 3             MVN              <NA>    <NA>     NO
##
## $univariateNormality
##            Test             Variable Statistic   p value Normality
## 1  Shapiro-Wilk    fixed.acidity        0.9420  <0.001      NO
## 2  Shapiro-Wilk   volatile.acidity      0.9743  <0.001      NO
## 3  Shapiro-Wilk     citric.acid         0.9553  <0.001      NO
## 4  Shapiro-Wilk    residual.sugar       0.5661  <0.001      NO
## 5  Shapiro-Wilk      chlorides          0.4842  <0.001      NO
## 6  Shapiro-Wilk free.sulfur.dioxide     0.9018  <0.001      NO
## 7  Shapiro-Wilk total.sulfur.dioxide    0.8732  <0.001      NO
## 8  Shapiro-Wilk       density           0.9909  <0.001      NO
```

```
## 9  Shapiro-Wilk          pH             0.9935  <0.001      NO
## 10 Shapiro-Wilk      sulphates          0.8330  <0.001      NO
## 11 Shapiro-Wilk       alcohol           0.9288  <0.001      NO
## 12 Shapiro-Wilk       quality           0.8576  <0.001      NO
##
## $Descriptives
##                          n        Mean      Std.Dev   Median     Min
Max
## fixed.acidity        1599  8.31963727  1.741096318  7.90000 4.60000
15.90000
## volatile.acidity     1599  0.52782051  0.179059704  0.52000 0.12000
1.58000
## citric.acid          1599  0.27097561  0.194801137  0.26000 0.00000
1.00000
## residual.sugar       1599  2.53880550  1.409928060  2.20000 0.90000
15.50000
## chlorides            1599  0.08746654  0.047065302  0.07900 0.01200
0.61100
## free.sulfur.dioxide  1599 15.87492183 10.460156970 14.00000 1.00000
72.00000
## total.sulfur.dioxide 1599 46.46779237 32.895324478 38.00000 6.00000
289.00000
## density              1599  0.99674668  0.001887334  0.99675 0.99007
1.00369
## pH                   1599  3.31111320  0.154386465  3.31000 2.74000
4.01000
## sulphates            1599  0.65814884  0.169506980  0.62000 0.33000
2.00000
## alcohol              1599 10.42298311  1.065667582 10.20000 8.40000
14.90000
## quality              1599  5.63602251  0.807569440  6.00000 3.00000
8.00000
```

```
##                          25th      75th       Skew   Kurtosis
## fixed.acidity          7.1000  9.200000 0.98090840  1.1196987
## volatile.acidity       0.3900  0.640000 0.67033307  1.2126893
## citric.acid            0.0900  0.420000 0.31774029 -0.7930455
## residual.sugar         1.9000  2.600000 4.53213992 28.4850200
## chlorides              0.0700  0.090000 5.66969370 41.5259635
## free.sulfur.dioxide    7.0000 21.000000 1.24822199  2.0072212
## total.sulfur.dioxide  22.0000 62.000000 1.51268904  3.7856764
## density                0.9956  0.997835 0.07115397  0.9225000
## pH                     3.2100  3.400000 0.19332027  0.7959191
## sulphates              0.5500  0.730000 2.42411764 11.6615285
## alcohol                9.5000 11.100000 0.85921442  0.1916586
## quality                5.0000  6.000000 0.21739311  0.2879148
```

We can conclude that none of the variables are normally distributed. Thus, we will now have to apply CLT.

Applying CLT

```
#fix skew
#Vino[,3]=Vino[,3]+1
Vino[1:11]=sapply(Vino[1:11],sqrt)

##storing info for target variable
ca=vector()
c=vector()
count=0
a=vector()
for (i in Vino$quality){{
  count=count+1}
    if (i==4){
    next}
    else if (i==5){
```

```
    next}
    else if (i==6){
    next}
    else if (i==7){
    next}
    else {
    c=append(c,count)
    a=append(a,i)
    ca=append(ca,c(count,i))}}
```

Remove outliers function

```
# Remove outliers from a column
remove_outliers <- function(x, na.rm = TRUE, ...) {
  qnt <- quantile(x, probs=c(.25, .75), na.rm = na.rm, ...)
  H <- 1.5 * IQR(x, na.rm = na.rm)
  y <- x
  y[x < (qnt[1] - H)] <- NA
  y[x > (qnt[2] + H)] <- NA
  y
}
remove_all_outliers1 <- function(df){
  # We only want the numeric columns
df= sapply(df, remove_outliers)
df=as.data.frame(df)
}
Vino=remove_all_outliers1(Vino)

Vino$quality[is.na(Vino$quality)]= a[1:sum(is.na(Vino$quality))]
Vino=Vino[complete.cases(Vino[-12]),]
```

Attach Clean Data

```
attach(Vino)
search()
```

```
##  [1] ".GlobalEnv"            "Vino"                  "package:mda"
##  [4] "package:class"         "package:lsmeans"       "package:emmeans"
##  [7] "package:caTools"       "package:rgl"
"package:scatterplot3d"
## [10] "package:MVN"           "package:gridExtra"     "package:grid"
## [13] "package:cowplot"       "package:ggplot2"       "package:tidyr"
## [16] "package:stats"         "package:graphics"      "package:grDevices"
## [19] "package:utils"         "package:datasets"      "package:methods"
## [22] "Autoloads"             "package:base"
```

Column names

```
vinonames=names(Vino)
vinonames
```

```
##  [1] "fixed.acidity"        "volatile.acidity"     "citric.acid"
##  [4] "residual.sugar"       "chlorides"            "free.sulfur.dioxide"
##  [7] "total.sulfur.dioxide" "density"              "pH"
## [10] "sulphates"            "alcohol"              "quality"
```

Univariate Exploration

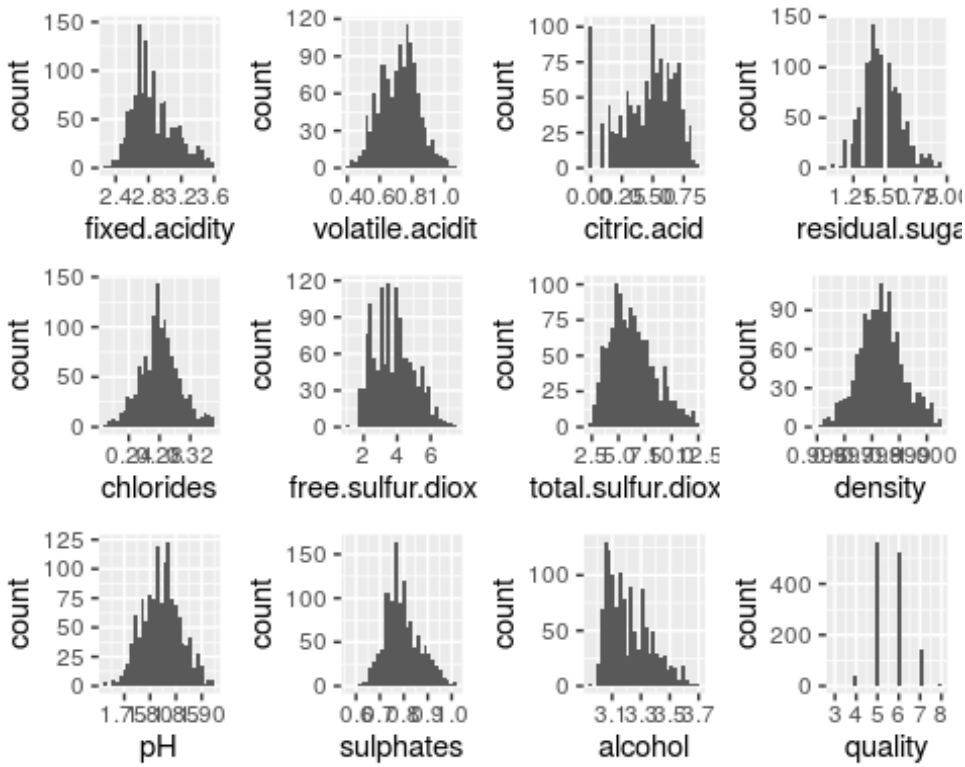
Univariate Plots

```
##histograms
my_plots <- lapply(vinonames, function(var_x){
  p <-
    ggplot(Vino) +
    aes_string(var_x)
```

```
  if(is.numeric(Vino[[var_x]])) {
    p <- p + geom_histogram()

  } else {
    p <- p + geom_bar()
  }

})
```

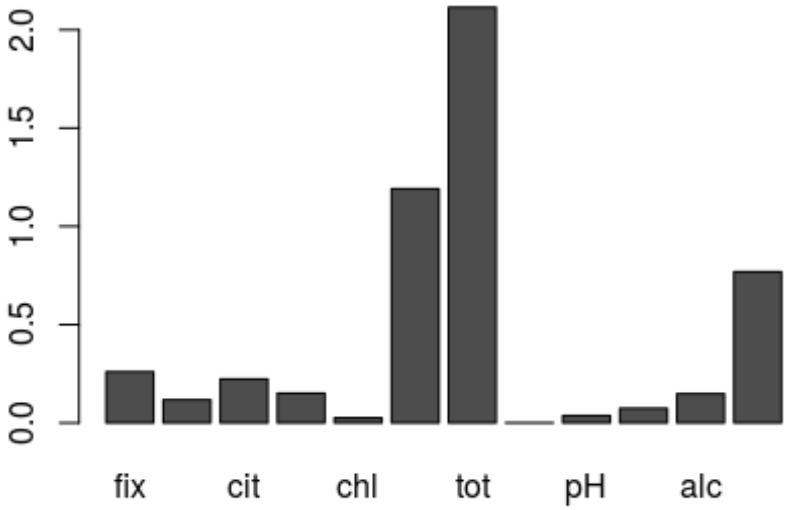
Histogram Plots

```
plot_grid(plotlist = my_plots)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Univariate SD Comparison Plot

```
S=lapply(Vino,sd)
Q=t(matrix(S))
Y=barplot(Q,names.arg=substr(vinonames,1,3))
```



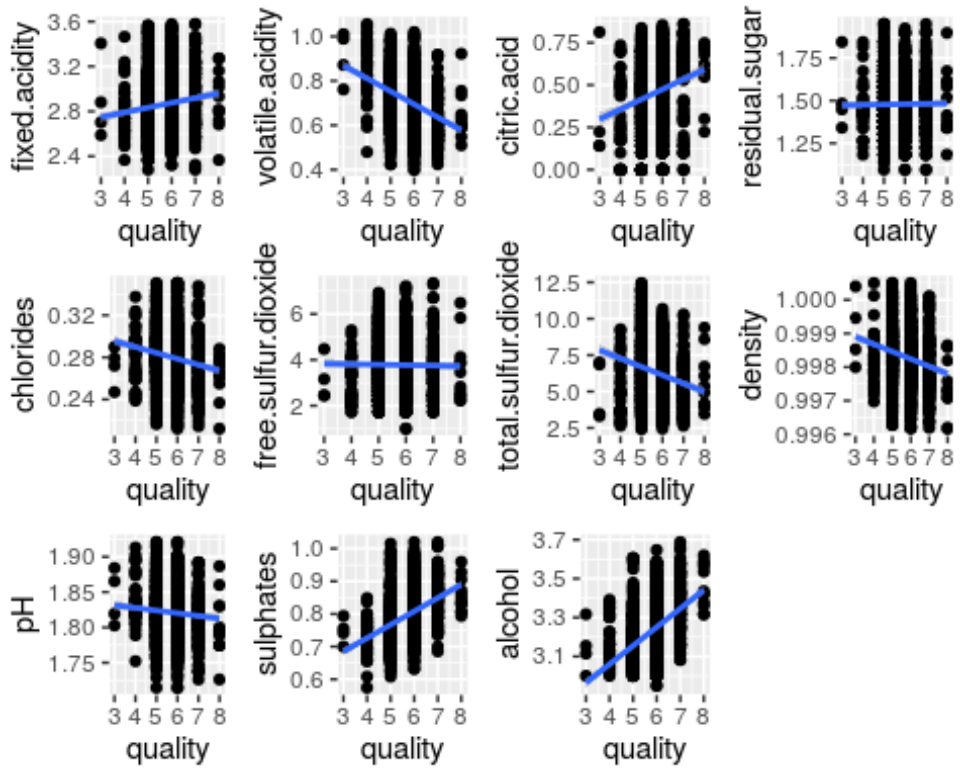
Bivariate Exploration

Bivariate plots with fitted line

```
vinonames=names(Vino)


  p1 <- ggplot(Vino,aes(quality, Vino[,1])) + geom_point()+geom_smooth(method
= "lm")+ylab(vinonames[1])
  p2<-  ggplot(Vino,aes(quality, Vino[,2])) + geom_point()+geom_smooth(method
= "lm")+ylab(vinonames[2])
  p3 <- ggplot(Vino,aes(quality, Vino[,3])) + geom_point()+geom_smooth(method
= "lm")+ylab(vinonames[3])
  p4<-  ggplot(Vino,aes(quality, Vino[,4])) + geom_point()+geom_smooth(method
= "lm")+ylab(vinonames[4])
  p5 <- ggplot(Vino,aes(quality, Vino[,5])) + geom_point()+geom_smooth(method
= "lm")+ylab(vinonames[5])
  p6<-  ggplot(Vino,aes(quality, Vino[,6])) + geom_point()+geom_smooth(method
= "lm")+ylab(vinonames[6])
```

```
  p7 <-  ggplot(Vino,aes(quality, Vino[,7])) +
geom_point()+geom_smooth(method = "lm")+ylab(vinonames[7])
  p8<-  ggplot(Vino,aes(quality, Vino[,8])) + geom_point()+geom_smooth(method
= "lm")+ylab(vinonames[8])
  p9 <-  ggplot(Vino,aes(quality, Vino[,9])) +
geom_point()+geom_smooth(method = "lm")+ylab(vinonames[9])
  p10<-  ggplot(Vino,aes(quality, Vino[,10])) +
geom_point()+geom_smooth(method = "lm")+ylab(vinonames[10])
  p11 <-  ggplot(Vino,aes(quality, Vino[,11])) +
geom_point()+geom_smooth(method = "lm")+ylab(vinonames[11])
  grid.arrange(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11, ncol=4)
```

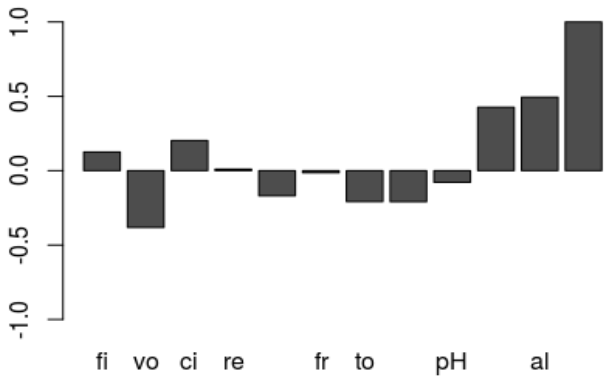
```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```




```
#VA,CA,Density,alcohol
```

Bivariate Correlation Comparison

```
attach(Vino)

## The following objects are masked from Vino (pos = 3):
##
##     alcohol, chlorides, citric.acid, density, fixed.acidity,
##     free.sulfur.dioxide, pH, quality, residual.sugar, sulphates,
##     total.sulfur.dioxide, volatile.acidity

##who is correlated with quality
K=(cor(Vino,quality))
K=t(as.matrix(K))
barplot(K,names.arg = substr(vinonames,1,2),ylim = c(-1,1))
```

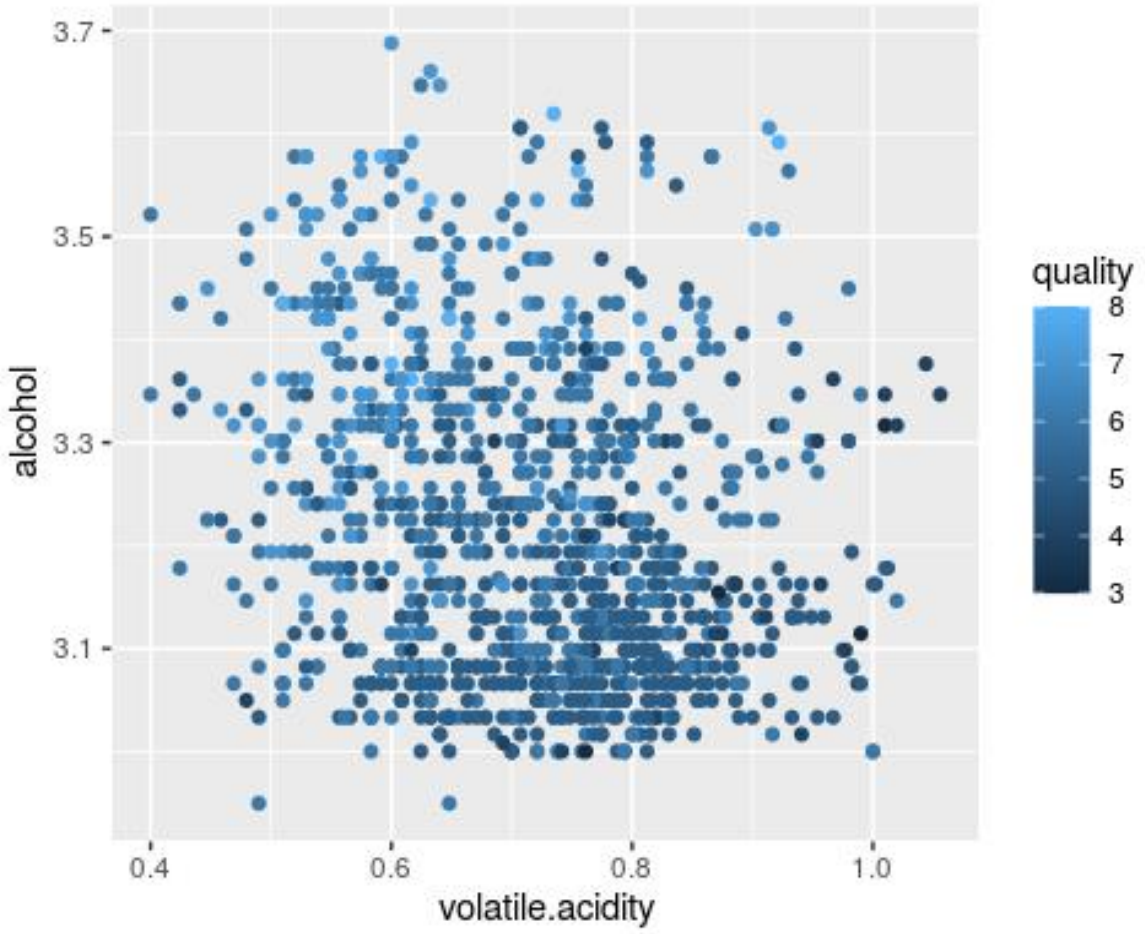



```
##vol,sul,alc higher correlations
```

Multivariate Exploration

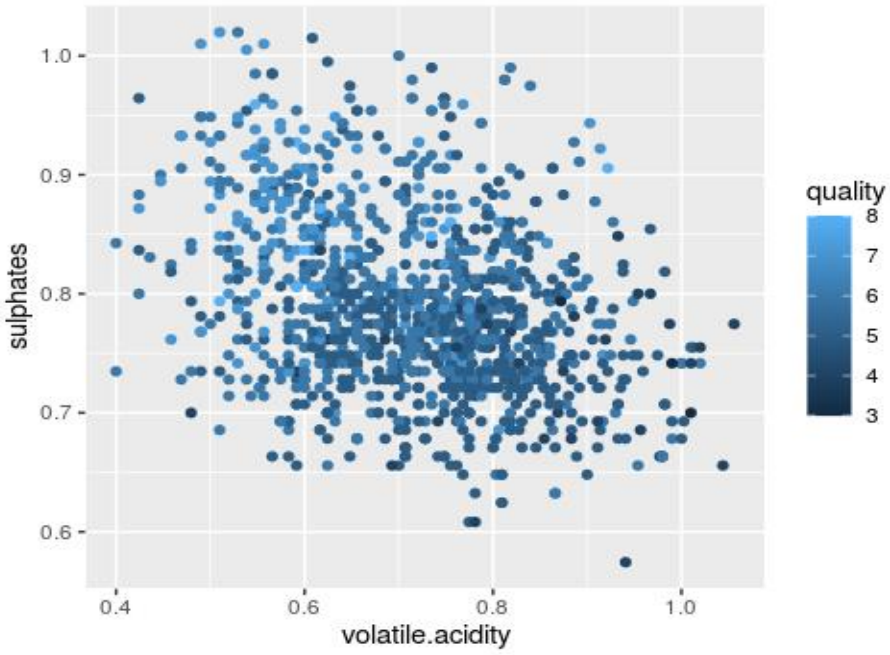
Multivariate Plots in 2-D

```
##VA and alc
require(ggplot2)
qplot(volatile.acidity, alcohol, colour = quality)
```



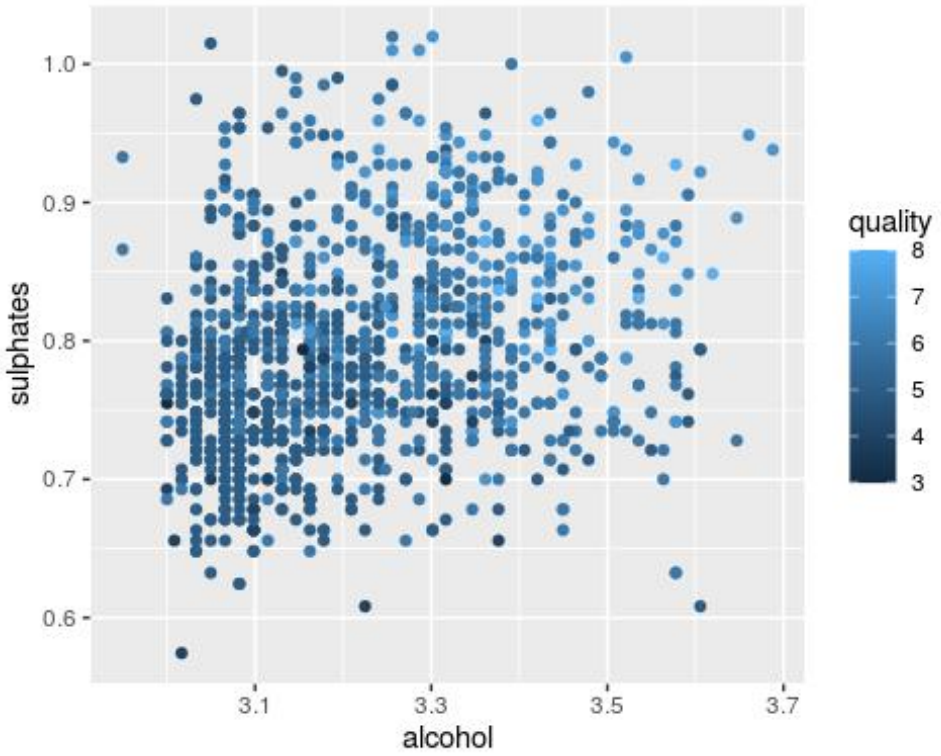
Here, we can see pretty good seperation. The top left has better quality than the bottom right.

```
##VA and sulphates
qplot(volatile.acidity,sulphates, colour = quality)
```



We can see a similar trend as above.

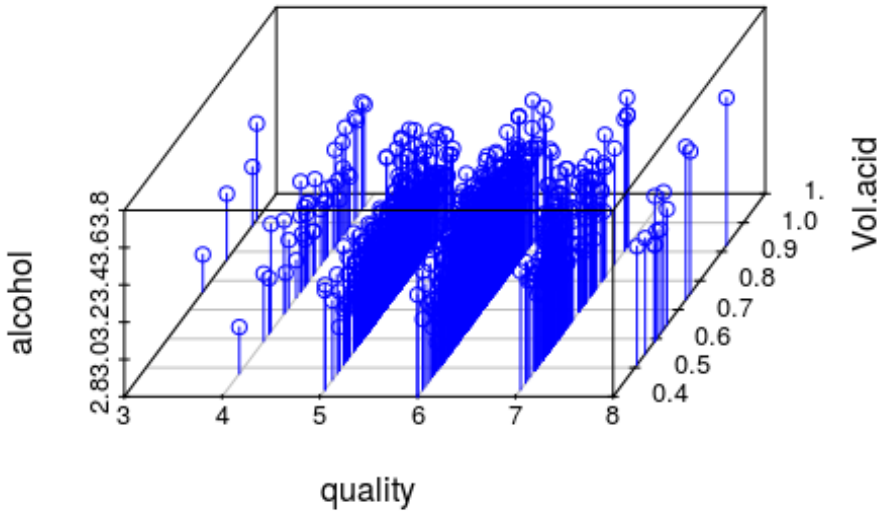
```
##alc and sulphates
qplot(alcohol,sulphates, colour = quality)
```



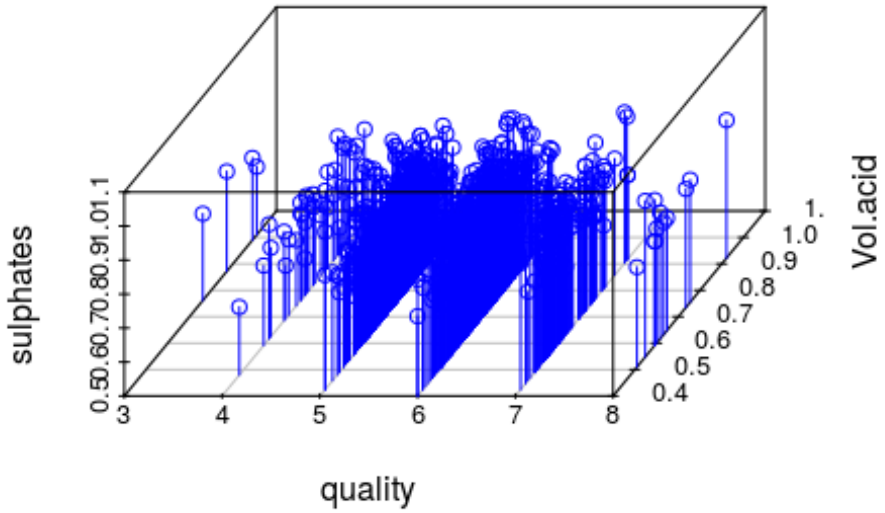
You can see darker to lighter moving from left to right, and top to bottom.

More multivariate plots in 3-D

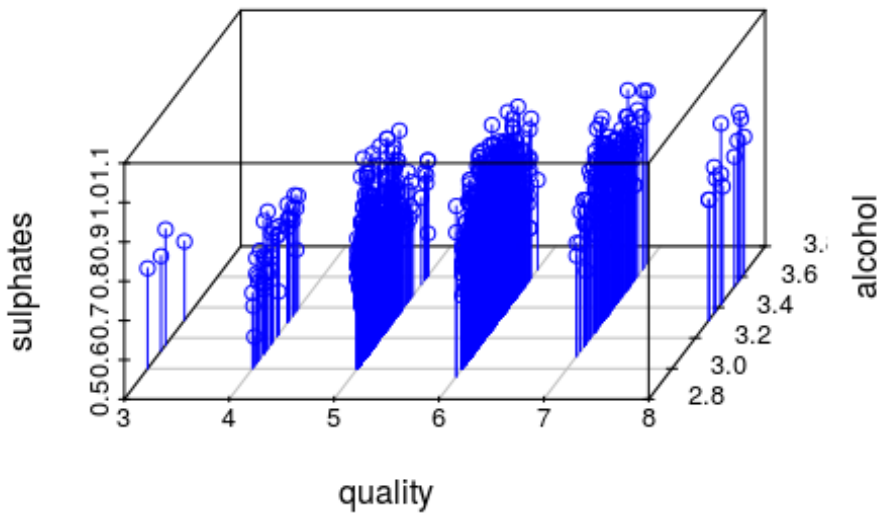
```
##higher quality obs are higher alcohol and lower VA
scatterplot3d(quality,volatile.acidity,alcohol,
xlab = "quality", ylab = "Vol.acid",zlab = "alcohol",
tick.marks = T,angle = 70,color=('blue'),type="h")
```




```
##higher quality obs are higher sulphate and lower VA
scatterplot3d(quality,volatile.acidity,sulphates,
xlab = "quality", ylab = "Vol.acid",zlab = "sulphates",
tick.marks = T,angle = 70,color=('blue'),type="h")
```




```
##higher quality obs are higher alcohol and higher sulphates
scatterplot3d(quality,alcohol,sulphates,
xlab = "quality", ylab = "alcohol",zlab = "sulphates",
tick.marks = T,angle = 70,color=('blue'),type="h")
```



3D plot of top 3 correlations with quality

```
##intereactive plot
A=plot3d(alcohol,volatile.acidity,sulphates,col=rainbow(length(unique(quality
)))),size=5)
legend3d("topright", legend= paste('rating', c(sort(unique(quality)))), pch =
16, col = rainbow(length(unique(quality))), cex=1, inset=c(.02,0.02))
A
```

Multivariate Comparisons

```
##multivariate corr
##sample mean vector
xbar=sapply(Vino,mean)
xbar=as.matrix(xbar)
xbar

##                          [,1]
## fixed.acidity        2.8601576
## volatile.acidity     0.7165244
## citric.acid          0.4525418
## residual.sugar       1.4791486
## chlorides            0.2807291
```

```
## free.sulfur.dioxide  3.7805769
## total.sulfur.dioxide 6.3452260
## density              0.9983246
## pH                   1.8216080
## sulphates            0.7925424
## alcohol              3.2145917
## quality              5.6232449
```

```
##sample variance-covariance matrix
co=cov(Vino)
co
```

```
##                      fixed.acidity volatile.acidity   citric.acid
## fixed.acidity         0.0675228722    -8.288476e-03  3.589591e-02
## volatile.acidity     -0.0082884763     1.379550e-02 -1.570088e-02
## citric.acid           0.0358959145    -1.570088e-02  4.985481e-02
## residual.sugar        0.0092818436     8.503039e-04  4.820992e-03
## chlorides             0.0012683155     3.779848e-04  4.084090e-04
## free.sulfur.dioxide  -0.0487156715     7.873991e-04 -1.179594e-02
## total.sulfur.dioxide -0.0393171027     2.756209e-02  3.549552e-02
## density               0.0001324938     3.417944e-06  5.329759e-05
## pH                   -0.0066657270     9.561850e-04 -4.161491e-03
## sulphates             0.0036138626    -3.077133e-03  4.107490e-03
## alcohol              -0.0008069748    -3.945883e-03  3.479006e-03
## quality               0.0251310313    -3.446182e-02  3.479612e-02
##                      residual.sugar     chlorides free.sulfur.dioxide
## fixed.acidity          9.281844e-03  1.268316e-03       -4.871567e-02
## volatile.acidity       8.503039e-04  3.779848e-04        7.873991e-04
## citric.acid            4.820992e-03  4.084090e-04       -1.179594e-02
## residual.sugar         2.286257e-02  9.801352e-04        1.753041e-02
## chlorides              9.801352e-04  6.620527e-04        1.064640e-03
## free.sulfur.dioxide    1.753041e-02  1.064640e-03        1.418173e+00
## total.sulfur.dioxide   5.867106e-02  1.012216e-02        1.811020e+00
```

```
## density                4.772219e-05  7.963144e-06       -2.787912e-05
## pH                    -3.307367e-04 -1.544386e-04        6.136735e-03
## sulphates              5.335397e-04 -1.039377e-04        8.695091e-03
## alcohol                2.449208e-03 -9.843051e-04       -6.310354e-03
## quality                1.196307e-03 -3.336785e-03       -1.268292e-02
##                      total.sulfur.dioxide       density            pH
## fixed.acidity              -0.0393171027  1.324938e-04 -6.665727e-03
## volatile.acidity            0.0275620906  3.417944e-06  9.561850e-04
## citric.acid                 0.0354955211  5.329759e-05 -4.161491e-03
## residual.sugar              0.0586710579  4.772219e-05 -3.307367e-04
## chlorides                   0.0101221606  7.963144e-06 -1.544386e-04
## free.sulfur.dioxide         1.8110197505 -2.787912e-05  6.136735e-03
## total.sulfur.dioxide        4.4701429587  2.726358e-04 -7.179856e-04
## density                     0.0002726358  6.537879e-07 -7.231497e-06
## pH                         -0.0007179856 -7.231497e-06  1.370806e-03
## sulphates                  -0.0065699219  6.877240e-06  8.454135e-06
## alcohol                    -0.0850218737 -6.064791e-05  6.414472e-04
## quality                    -0.3382800010 -1.298637e-04 -2.228537e-03
##                          sulphates       alcohol       quality
## fixed.acidity         3.613863e-03 -8.069748e-04  0.0251310313
## volatile.acidity     -3.077133e-03 -3.945883e-03 -0.0344618225
## citric.acid           4.107490e-03  3.479006e-03  0.0347961163
## residual.sugar        5.335397e-04  2.449208e-03  0.0011963068
## chlorides            -1.039377e-04 -9.843051e-04 -0.0033367846
## free.sulfur.dioxide   8.695091e-03 -6.310354e-03 -0.0126829176
## total.sulfur.dioxide -6.569922e-03 -8.502187e-02 -0.3382800010
## density               6.877240e-06 -6.064791e-05 -0.0001298637
## pH                    8.454135e-06  6.414472e-04 -0.0022285368
## sulphates             5.445466e-03  2.917371e-03  0.0241949062
## alcohol               2.917371e-03  2.198153e-02  0.0563496448
## quality               2.419491e-02  5.634964e-02  0.5909658869
```

```r
##sample correlations matrix
r=cor(Vino)
r
```

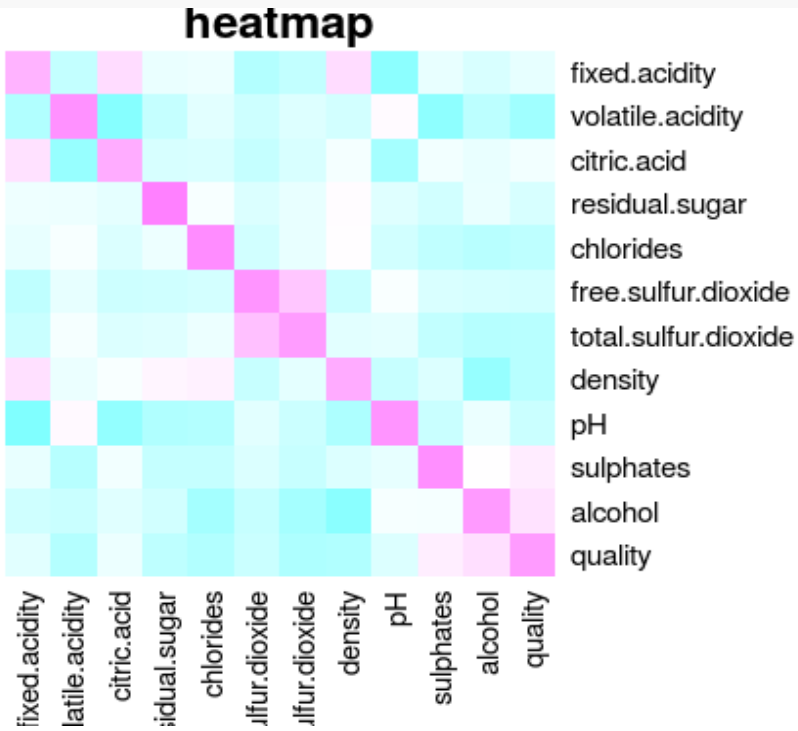
```
##                      fixed.acidity volatile.acidity citric.acid
residual.sugar
## fixed.acidity           1.00000000     -0.271569236  0.61867996
0.23623608
## volatile.acidity       -0.27156924      1.000000000 -0.59868979
0.04787876
## citric.acid             0.61867996     -0.598689794  1.00000000
0.14279742
## residual.sugar          0.23623608      0.047878761  0.14279742
1.00000000
## chlorides               0.18969487      0.125071787  0.07108787
0.25192841
## free.sulfur.dioxide    -0.15742681      0.005629394 -0.04436234
0.09735652
## total.sulfur.dioxide   -0.07156412      0.110989724  0.07518987
0.18352729
## density                 0.63059683      0.035989666  0.29521307
0.39033665
## pH                     -0.69284183      0.219879768 -0.50339330    -
0.05907882
## sulphates               0.18846411     -0.355025876  0.24929035
0.04781748
## alcohol                -0.02094622     -0.226593047  0.10509274
0.10925319
## quality                 0.12580667     -0.381670291  0.20271974
0.01029198
##                       chlorides free.sulfur.dioxide total.sulfur.dioxide
## fixed.acidity         0.18969487        -0.157426806         -0.071564122
## volatile.acidity      0.12507179         0.005629394          0.110989724
```

```
## citric.acid           0.07108787        -0.044362337          0.075189870
## residual.sugar        0.25192841         0.097356525          0.183527290
## chlorides             1.00000000         0.034744971          0.186065665
## free.sulfur.dioxide   0.03474497         1.000000000          0.719279904
## total.sulfur.dioxide  0.18606567         0.719279904          1.000000000
## density               0.38275411        -0.028953168          0.159479115
## pH                   -0.16211438         0.139182563         -0.009172063
## sulphates            -0.05474055         0.098944591         -0.042109683
## alcohol              -0.25802069        -0.035740467         -0.271232072
## quality              -0.16869441        -0.013853943         -0.208129869
##                         density           pH    sulphates     alcohol
## fixed.acidity         0.63059683 -0.692841835  0.188464107 -0.02094622
## volatile.acidity      0.03598967  0.219879768 -0.355025876 -0.22659305
## citric.acid           0.29521307 -0.503393298  0.249290350  0.10509274
## residual.sugar        0.39033665 -0.059078820  0.047817475  0.10925319
## chlorides             0.38275411 -0.162114379 -0.054740545 -0.25802069
## free.sulfur.dioxide  -0.02895317  0.139182563  0.098944591 -0.03574047
## total.sulfur.dioxide  0.15947911 -0.009172063 -0.042109683 -0.27123207
## density               1.00000000 -0.241558204  0.115259887 -0.50590446
## pH                   -0.24155820  1.000000000  0.003094309  0.11685413
## sulphates             0.11525989  0.003094309  1.000000000  0.26665222
## alcohol              -0.50590446  0.116854127  0.266652220  1.00000000
## quality              -0.20892392 -0.078297993  0.426506057  0.49440295
##                         quality
## fixed.acidity         0.12580667
## volatile.acidity     -0.38167029
## citric.acid           0.20271974
## residual.sugar        0.01029198
## chlorides            -0.16869441
## free.sulfur.dioxide  -0.01385394
## total.sulfur.dioxide -0.20812987
## density              -0.20892392
```

```
## pH                   -0.07829799
## sulphates             0.42650606
## alcohol               0.49440295
## quality               1.00000000
```

Visual for multivariate correlation comparison

```
M=as.matrix(Vino)
heatmap(r, Colv = NA, Rowv = NA, scale="column", main="heatmap",revC=TRUE,
col=cm.colors(256))
```



Linear Model Formulating

```
#Train and Test Split
set.seed(100)
trainRowIndex = sample(1:nrow(Vino), 0.75*nrow(Vino))
train= Vino[trainRowIndex, ]
test=Vino[-trainRowIndex, ]
```

```r
m0=lm(quality~alcohol+sulphates+volatile.acidity,data = train)
summary(m0)
```

```
##
## Call:
## lm(formula = quality ~ alcohol + sulphates + volatile.acidity,
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.30998 -0.36116 -0.07408  0.44724  1.96259
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -2.0584     0.5077  -4.054 5.44e-05 ***
## alcohol            1.9925     0.1363  14.621  < 2e-16 ***
## sulphates          2.8114     0.2896   9.708  < 2e-16 ***
## volatile.acidity  -1.3170     0.1790  -7.359 3.98e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6138 on 957 degrees of freedom
## Multiple R-squared:  0.3835, Adjusted R-squared:  0.3816
## F-statistic: 198.5 on 3 and 957 DF,  p-value: < 2.2e-16
```

```r
##for lower error higher R^2 add in some of the less correlated items to give more complexity

m1=lm(quality~alcohol+sulphates+volatile.acidity+citric.acid+chlorides+total.sulfur.dioxide,data=train)
summary(m1)
```

```
##
## Call:
```

```
## lm(formula = quality ~ alcohol + sulphates + volatile.acidity +
##     citric.acid + chlorides + total.sulfur.dioxide, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.40608 -0.36511 -0.05915  0.41775  1.92985
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -1.47285    0.61726  -2.386   0.0172 *
## alcohol               1.88658    0.14459  13.048  < 2e-16 ***
## sulphates             2.86540    0.28982   9.887  < 2e-16 ***
## volatile.acidity     -1.39783    0.22261  -6.279 5.16e-10 ***
## citric.acid          -0.09824    0.11411  -0.861   0.3895
## chlorides            -0.09908    0.80205  -0.124   0.9017
## total.sulfur.dioxide -0.02488    0.01005  -2.476   0.0135 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6122 on 954 degrees of freedom
## Multiple R-squared:  0.3886, Adjusted R-squared:  0.3848
## F-statistic: 101.1 on 6 and 954 DF,  p-value: < 2.2e-16

## next check for interaction between terms with stepwise algo

m2= step(m1, scope=list(upper=
~alcohol*sulphates*volatile.acidity*citric.acid*chlorides*total.sulfur.dioxid
e, lower= ~1))

## Start:  AIC=-936.2
## quality ~ alcohol + sulphates + volatile.acidity + citric.acid +
##     chlorides + total.sulfur.dioxide
##
##                                       Df Sum of Sq    RSS     AIC
```

```
## + sulphates:total.sulfur.dioxide         1     7.552 349.97 -954.72
## + volatile.acidity:total.sulfur.dioxide  1     5.311 352.22 -948.59
## + alcohol:sulphates                      1     4.615 352.91 -946.69
## + citric.acid:total.sulfur.dioxide       1     4.121 353.41 -945.34
## - chlorides                              1     0.006 357.53 -938.19
## + volatile.acidity:citric.acid           1     1.365 356.16 -937.88
## - citric.acid                            1     0.278 357.80 -937.46
## <none>                                               357.53 -936.20
## + sulphates:chlorides                    1     0.643 356.88 -935.93
## + citric.acid:chlorides                  1     0.336 357.19 -935.11
## + alcohol:chlorides                      1     0.302 357.22 -935.01
## + chlorides:total.sulfur.dioxide         1     0.212 357.31 -934.77
## + sulphates:volatile.acidity             1     0.158 357.37 -934.63
## + alcohol:total.sulfur.dioxide           1     0.071 357.46 -934.40
## + alcohol:citric.acid                    1     0.051 357.48 -934.34
## + sulphates:citric.acid                  1     0.026 357.50 -934.27
## + alcohol:volatile.acidity               1     0.008 357.52 -934.23
## + volatile.acidity:chlorides             1     0.004 357.52 -934.22
## - total.sulfur.dioxide                   1     2.298 359.82 -932.05
## - volatile.acidity                       1    14.777 372.30 -899.28
## - sulphates                              1    36.633 394.16 -844.46
## - alcohol                                1    63.801 421.33 -780.41
##
## Step:  AIC=-954.72
## quality ~ alcohol + sulphates + volatile.acidity + citric.acid +
##     chlorides + total.sulfur.dioxide + sulphates:total.sulfur.dioxide
##
##                                         Df Sum of Sq    RSS     AIC
## + alcohol:sulphates                      1     3.207 346.77 -961.57
## + volatile.acidity:total.sulfur.dioxide  1     2.120 347.85 -958.56
## + citric.acid:total.sulfur.dioxide       1     1.618 348.36 -957.17
## - chlorides                              1     0.001 349.98 -956.72
```

```
## + volatile.acidity:citric.acid           1     1.178 348.80 -955.96
## - citric.acid                            1     0.410 350.38 -955.59
## <none>                                               349.97 -954.72
## + alcohol:total.sulfur.dioxide           1     0.180 349.79 -953.21
## + alcohol:chlorides                      1     0.178 349.80 -953.21
## + sulphates:volatile.acidity             1     0.133 349.84 -953.08
## + chlorides:total.sulfur.dioxide         1     0.125 349.85 -953.06
## + sulphates:citric.acid                  1     0.081 349.89 -952.94
## + citric.acid:chlorides                  1     0.078 349.90 -952.93
## + alcohol:citric.acid                    1     0.014 349.96 -952.76
## + sulphates:chlorides                    1     0.012 349.96 -952.75
## + volatile.acidity:chlorides             1     0.004 349.97 -952.73
## + alcohol:volatile.acidity               1     0.000 349.97 -952.72
## - sulphates:total.sulfur.dioxide         1     7.552 357.53 -936.20
## - volatile.acidity                       1    14.126 364.10 -918.69
## - alcohol                                1    67.514 417.49 -787.20
##
## Step:  AIC=-961.57
## quality ~ alcohol + sulphates + volatile.acidity + citric.acid +
##     chlorides + total.sulfur.dioxide + sulphates:total.sulfur.dioxide +
##     alcohol:sulphates
##
##                                         Df Sum of Sq    RSS     AIC
## + volatile.acidity:total.sulfur.dioxide  1    2.5527 344.21 -966.67
## + citric.acid:total.sulfur.dioxide       1    1.6774 345.09 -964.23
## - chlorides                              1    0.0233 346.79 -963.50
## + volatile.acidity:citric.acid           1    1.0619 345.71 -962.51
## - citric.acid                            1    0.6313 347.40 -961.82
## <none>                                               346.77 -961.57
## + sulphates:volatile.acidity             1    0.3919 346.38 -960.65
## + sulphates:chlorides                    1    0.3378 346.43 -960.50
## + alcohol:volatile.acidity               1    0.2092 346.56 -960.15
```

```
## + alcohol:citric.acid                    1    0.1461 346.62 -959.97
## + alcohol:chlorides                      1    0.1439 346.62 -959.97
## + volatile.acidity:chlorides             1    0.0540 346.71 -959.72
## + citric.acid:chlorides                  1    0.0224 346.75 -959.63
## + chlorides:total.sulfur.dioxide         1    0.0220 346.75 -959.63
## + sulphates:citric.acid                  1    0.0061 346.76 -959.58
## + alcohol:total.sulfur.dioxide           1    0.0006 346.77 -959.57
## - alcohol:sulphates                      1    3.2072 349.97 -954.72
## - sulphates:total.sulfur.dioxide         1    6.1436 352.91 -946.69
## - volatile.acidity                       1   14.3297 361.10 -924.65
##
## Step:  AIC=-966.67
## quality ~ alcohol + sulphates + volatile.acidity + citric.acid +
##     chlorides + total.sulfur.dioxide + sulphates:total.sulfur.dioxide +
##     alcohol:sulphates + volatile.acidity:total.sulfur.dioxide
##
##                                        Df Sum of Sq    RSS     AIC
## - chlorides                             1    0.0091 344.22 -968.64
## + volatile.acidity:citric.acid          1    0.7829 343.43 -966.86
## <none>                                             344.21 -966.67
## + alcohol:volatile.acidity              1    0.5582 343.66 -966.23
## + citric.acid:total.sulfur.dioxide      1    0.4180 343.80 -965.83
## + sulphates:chlorides                   1    0.3428 343.87 -965.62
## - citric.acid                           1    1.1368 345.35 -965.50
## + alcohol:chlorides                     1    0.2833 343.93 -965.46
## + alcohol:citric.acid                   1    0.2760 343.94 -965.44
## + sulphates:volatile.acidity            1    0.2625 343.95 -965.40
## + volatile.acidity:chlorides            1    0.2385 343.98 -965.33
## + chlorides:total.sulfur.dioxide        1    0.1631 344.05 -965.12
## + alcohol:total.sulfur.dioxide          1    0.0394 344.18 -964.78
## + citric.acid:chlorides                 1    0.0212 344.19 -964.73
## + sulphates:citric.acid                 1    0.0141 344.20 -964.71
```

```
## - volatile.acidity:total.sulfur.dioxide  1    2.5527 346.77 -961.57
## - sulphates:total.sulfur.dioxide         1    3.1324 347.35 -959.96
## - alcohol:sulphates                      1    3.6396 347.85 -958.56
##
## Step:  AIC=-968.64
## quality ~ alcohol + sulphates + volatile.acidity + citric.acid +
##     total.sulfur.dioxide + sulphates:total.sulfur.dioxide +
alcohol:sulphates +
##     volatile.acidity:total.sulfur.dioxide
##
##                                         Df Sum of Sq    RSS     AIC
## + volatile.acidity:citric.acid           1    0.7722 343.45 -968.80
## <none>                                             344.22 -968.64
## + alcohol:volatile.acidity               1    0.5453 343.68 -968.16
## + citric.acid:total.sulfur.dioxide       1    0.4187 343.81 -967.81
## - citric.acid                            1    1.1314 345.36 -967.49
## + sulphates:volatile.acidity             1    0.2705 343.95 -967.40
## + alcohol:citric.acid                    1    0.2605 343.96 -967.37
## + alcohol:total.sulfur.dioxide           1    0.0357 344.19 -966.74
## + sulphates:citric.acid                  1    0.0140 344.21 -966.68
## + chlorides                              1    0.0091 344.21 -966.67
## - volatile.acidity:total.sulfur.dioxide  1    2.5668 346.79 -963.50
## - sulphates:total.sulfur.dioxide         1    3.1239 347.35 -961.96
## - alcohol:sulphates                      1    3.6313 347.86 -960.56
##
## Step:  AIC=-968.8
## quality ~ alcohol + sulphates + volatile.acidity + citric.acid +
##     total.sulfur.dioxide + sulphates:total.sulfur.dioxide +
alcohol:sulphates +
##     volatile.acidity:total.sulfur.dioxide + volatile.acidity:citric.acid
##
##                                         Df Sum of Sq    RSS     AIC
```

```
## <none>                                                343.45 -968.80
## - volatile.acidity:citric.acid           1    0.7722 344.22 -968.64
## + citric.acid:total.sulfur.dioxide       1    0.5352 342.92 -968.30
## + alcohol:volatile.acidity               1    0.4676 342.98 -968.11
## + alcohol:citric.acid                    1    0.1669 343.28 -967.27
## + sulphates:citric.acid                  1    0.1040 343.35 -967.09
## + sulphates:volatile.acidity             1    0.0746 343.38 -967.01
## + alcohol:total.sulfur.dioxide           1    0.0433 343.41 -966.92
## + chlorides                              1    0.0199 343.43 -966.86
## - volatile.acidity:total.sulfur.dioxide  1    2.2945 345.75 -964.40
## - sulphates:total.sulfur.dioxide         1    3.1429 346.59 -962.05
## - alcohol:sulphates                      1    3.4874 346.94 -961.09
```

```
summary(m2)
```

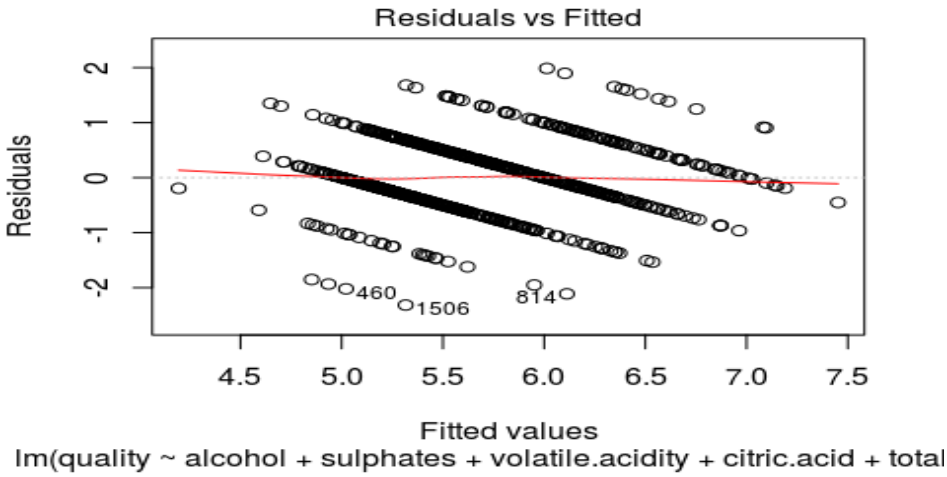
```
##
## Call:
## lm(formula = quality ~ alcohol + sulphates + volatile.acidity +
##     citric.acid + total.sulfur.dioxide + sulphates:total.sulfur.dioxide +
##     alcohol:sulphates + volatile.acidity:total.sulfur.dioxide +
##     volatile.acidity:citric.acid, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.31627 -0.33864 -0.09772  0.41693  1.98625
##
## Coefficients:
##                                      Estimate Std. Error t value
Pr(>|t|)
## (Intercept)                          12.16248    4.85068   2.507
0.01233 *
## alcohol                              -2.47446    1.42628  -1.735
0.08308 .
```

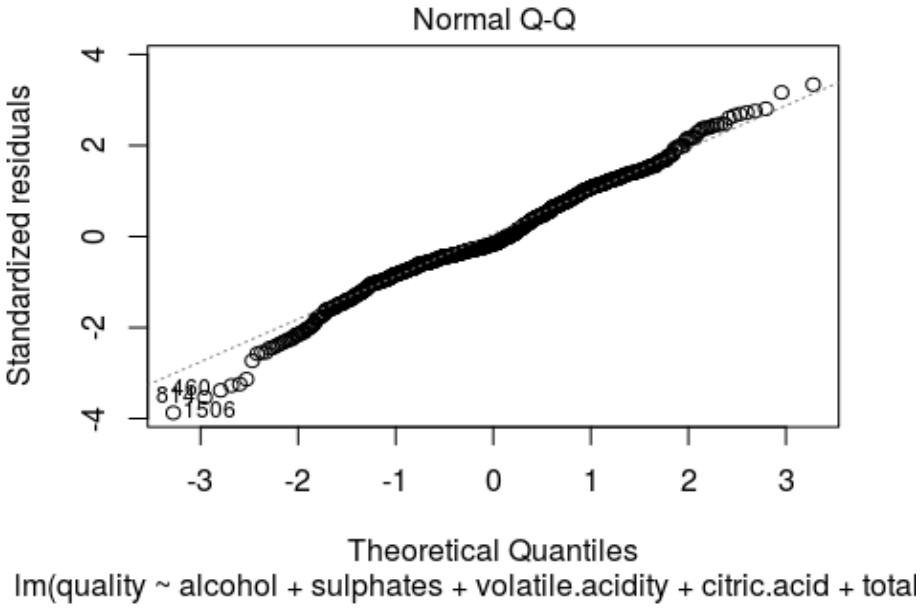
```
## sulphates                                -12.43240    5.95505  -2.088
0.03709 *
## volatile.acidity                          -3.45172    0.72184  -4.782 2.01e-
06 ***
## citric.acid                               -1.18985    0.68616  -1.734
0.08323 .
## total.sulfur.dioxide                       0.12691    0.14491   0.876
0.38137
## sulphates:total.sulfur.dioxide            -0.41035    0.13910  -2.950
0.00326 **
## alcohol:sulphates                          5.53370    1.78077   3.107
0.00194 **
## volatile.acidity:total.sulfur.dioxide      0.22374    0.08876   2.521
0.01188 *
## volatile.acidity:citric.acid               1.31967    0.90250   1.462
0.14401
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.601 on 951 degrees of freedom
## Multiple R-squared:  0.4127, Adjusted R-squared:  0.4071
## F-statistic: 74.25 on 9 and 951 DF,  p-value: < 2.2e-16
```

Error minimized R^2, maximized m2 is final model.

Check Assumptions

```
plot(m2,which = c(1:2))
```





Residuals sum to zero. QQ plot shows fairly even tails. Overall, we can say the assumptions are met.

Linear Model Performance

```
#Train and Test Split
set.seed(100)
trainRowIndex = sample(1:nrow(Vino), 0.75*nrow(Vino))
train= Vino[trainRowIndex, ]
test=Vino[-trainRowIndex, ]
```

```r
#train preds
distPred=predict(m2,train)
head(distPred)
```

```
##      655     1243     1265     1161      613     1036
## 5.120413 6.137987 6.315824 6.252214 5.436460 6.003223
```

```r
#train preds rounded
distPred1=round(distPred)
head(distPred1)
```

```
##  655 1243 1265 1161  613 1036
##    5    6    6    6    5    6
```

```r
#trainCM
train_quality=train$quality
distPred1=as.factor(distPred1)
traintab= confusionMatrix(distPred1, as.factor(train_quality))
```

```
## Warning in levels(reference) != levels(data): longer object length is not
a
## multiple of shorter object length
```

```
## Warning in confusionMatrix.default(distPred1, as.factor(train_quality)):
Levels
## are not in the same order for reference and data. Refactoring data to
match.
```

```r
conf_mat=traintab$table
conf_mat
```

```
##           Reference
## Prediction   3   4   5   6   7   8
##          3   0   0   0   0   0   0
##          4   0   1   0   0   0   0
##          5   4  21 314 117   2   0
```

```
##          6   0   4 103 258  73   6
##          7   0   0   2  16  35   5
##          8   0   0   0   0   0   0
```

```
options(digits=3)
conf_mat_norm=(rbind(conf_mat[1,]/(sum(conf_mat[1,])),conf_mat[2,]/(sum(con
f_mat[2,])),conf_mat[3,]/(sum(conf_mat[3,])),conf_mat[4,]/(sum(conf_mat[4,]
)),conf_mat[5,]/(sum(conf_mat[5,])),conf_mat[6,]/(sum(conf_mat[6,]))))
row.names(conf_mat_norm)=c(3,4,5,6,7,8)
conf_mat_norm
```

```
##         3       4      5     6       7      8
## 3     NaN     NaN    NaN   NaN     NaN    NaN
## 4 0.00000 1.00000 0.0000 0.000 0.00000 0.0000
## 5 0.00873 0.04585 0.6856 0.255 0.00437 0.0000
## 6 0.00000 0.00901 0.2320 0.581 0.16441 0.0135
## 7 0.00000 0.00000 0.0345 0.276 0.60345 0.0862
## 8     NaN     NaN    NaN   NaN     NaN    NaN
```

```
##train acc
sum(diag(traintab))/length(train$quality)
```

```
## [1] 0.633
```

```
##test preds
distPred=predict(m2,test)
head(distPred)
```

```
##        5        7        8       23       30       33
## 5.121632 5.104281 5.147273 5.817507 5.366394 5.283952
```

```
##rounded test preds
distPred1=round(distPred)
head(distPred1)
```

```
##  5  7  8 23 30 33
##  5  5  5  6  5  5
```

```
##testCM
test_quality=test$quality
distPred1=as.factor(distPred1)
testtab= confusionMatrix(distPred1, as.factor(test_quality))
```

```
## Warning in levels(reference) != levels(data): longer object length is not
a
## multiple of shorter object length
```

```
## Warning in confusionMatrix.default(distPred1, as.factor(test_quality)):
Levels
## are not in the same order for reference and data. Refactoring data to
match.
```

```
conf_mat_test=testtab$table
conf_mat_test
```

```
##           Reference
## Prediction   4   5   6   7   8
##          4   0   0   0   0   0
##          5   9 100  43   2   0
##          6   2  45  82  18   1
##          7   0   0   6  13   0
##          8   0   0   0   0   0
```

```
options(digits=3)
conf_mat_norm_test=(rbind(conf_mat_test[1,]/(sum(conf_mat_test[1,])),conf_ma
t_test[2,]/(sum(conf_mat_test[2,])),conf_mat_test[3,]/(sum(conf_mat_test[3,
])),conf_mat_test[4,]/(sum(conf_mat_test[4,])),conf_mat_test[5,]/(sum(conf_
mat_test[5,]))))
row.names(conf_mat_norm_test)=c(4,5,6,7,8)
conf_mat_norm_test
```

```
##        4     5     6     7       8
## 4    NaN   NaN   NaN   NaN     NaN
## 5 0.0584 0.649 0.279 0.013 0.00000
## 6 0.0135 0.304 0.554 0.122 0.00676
## 7 0.0000 0.000 0.316 0.684 0.00000
## 8    NaN   NaN   NaN   NaN     NaN
```

```
#test acc
sum(diag(testtab))/length(test$quality)
```

```
## [1] 0.607
```

These results are not great even with our best models. We know due to the statstical analysis that these features do highly contribute to quality, but we are having a dificult time predicting ranking of quality. This could be due to judges/biases so next we group rankings into good or bad and make binomial regression instead of linear regression.

Binomial LR

```
##covert to good bad qualities
Vino$category[Vino$quality <= 5] = 0
Vino$category[Vino$quality > 5] = 1


Vino$category=as.factor(Vino$category)


Vino$category=as.factor(Vino$category)
head(Vino)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1      2.720294        0.8366600   0.0000000       1.378405 0.2756810
## 2      2.792848        0.9380832   0.0000000       1.612452 0.3130495
## 3      2.792848        0.8717798   0.2000000       1.516575 0.3033150
## 4      3.346640        0.5291503   0.7483315       1.378405 0.2738613
## 5      2.720294        0.8366600   0.0000000       1.378405 0.2756810
```

```
## 6      2.720294        0.8124038   0.0000000       1.341641 0.2738613
##   free.sulfur.dioxide total.sulfur.dioxide   density       pH sulphates
## 1            3.316625             5.830952 0.9988994 1.873499 0.7483315
## 2            5.000000             8.185353 0.9983987 1.788854 0.8246211
## 3            3.872983             7.348469 0.9984989 1.805547 0.8062258
## 4            4.123106             7.745967 0.9989995 1.777639 0.7615773
## 5            3.316625             5.830952 0.9988994 1.873499 0.7483315
## 6            3.605551             6.324555 0.9988994 1.873499 0.7483315
##    alcohol quality category
## 1 3.065942       5        0
## 2 3.130495       5        0
## 3 3.130495       5        0
## 4 3.130495       6        1
## 5 3.065942       5        0
## 6 3.065942       5        0
```

```r
#Train/Test
set.seed(25)
spl = sample.split(Vino$category, SplitRatio = 0.75)
train = subset(Vino, spl==TRUE)
test = subset(Vino, spl==FALSE)
```

BLR Model: convert previous LM to GLM for BLR

```r
#BLR
model_glm= glm(category ~alcohol + sulphates + volatile.acidity +
    citric.acid + chlorides + total.sulfur.dioxide +
sulphates:total.sulfur.dioxide +
    volatile.acidity:total.sulfur.dioxide + alcohol:sulphates, data = train,
family=binomial(link = "logit"))
summary(model_glm)
```

```
##
## Call:
```

```
## glm(formula = category ~ alcohol + sulphates + volatile.acidity +
##     citric.acid + chlorides + total.sulfur.dioxide +
sulphates:total.sulfur.dioxide +
##     volatile.acidity:total.sulfur.dioxide + alcohol:sulphates,
##     family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5949  -0.7843   0.1719   0.8103   2.2118
##
## Coefficients:
##                                      Estimate Std. Error z value
Pr(>|z|)
## (Intercept)                          13.67547   24.72273   0.553
0.58016
## alcohol                              -7.36354    7.46842  -0.986
0.32415
## sulphates                           -37.57285   31.24777  -1.202
0.22920
## volatile.acidity                     -4.26385    2.55110  -1.671
0.09465 .
## citric.acid                          -0.52282    0.45870  -1.140
0.25438
## chlorides                            -6.79474    3.37542  -2.013
0.04411 *
## total.sulfur.dioxide                  1.21683    0.63334   1.921
0.05470 .
## sulphates:total.sulfur.dioxide       -1.80316    0.64925  -2.777
0.00548 **
## volatile.acidity:total.sulfur.dioxide  0.06442    0.37735   0.171
0.86445
## alcohol:sulphates                    18.16660    9.59163   1.894
```

```
0.05822 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1330.58  on 961  degrees of freedom
## Residual deviance:  958.37  on 952  degrees of freedom
## AIC: 978.37
##
## Number of Fisher Scoring iterations: 5

##BLR predictions
head(fitted(model_glm))

##         1         2         3         6         8         9
## 0.2250694 0.2131683 0.2633153 0.2369868 0.2679365 0.3372768

head(predict(model_glm, type = "response"))

##         1         2         3         6         8         9
## 0.2250694 0.2131683 0.2633153 0.2369868 0.2679365 0.3372768

##BLR converted predictions
trainpred = ifelse(predict(model_glm, type = "response") > 0.5,"Good Wine",
"Bad Wine")
head(trainpred)

##          1          2          3          6          8          9
## "Bad Wine" "Bad Wine" "Bad Wine" "Bad Wine" "Bad Wine" "Bad Wine"

##BLR Train CM & acc
traintab = confusion(trainpred,train$category)
traintab
```

```
##            true
## predicted     0   1
##   Bad Wine  361 137
##   Good Wine  93 371
```

```
sum(diag(traintab))/length(train$category)
```

```
## [1] 0.761
```

```
#BLR converted test pred
testpred = ifelse(predict(model_glm, newdata = test, type = "response") >
0.5, "Good Wine", "Bad Wine")
```

```
##BLR Test CM & acc
testtab = confusion(testpred, test$category)
testtab
```

```
##            true
## predicted     0   1
##   Bad Wine  116  44
##   Good Wine  35 125
```

```
sum(diag(testtab))/length(test$category)
```

```
## [1] 0.753
```

Conclusion

The chemical components of wine contribute to the quality of wine. Alcohol,Sulphates,and Volatile Acidity contribute significantly to the quality. It is difficult to predict wine quality rating, due to inherent limitations of a wine rating system. However, we were able to successfully classify wine into good and bad quality at about a 75% rate.