Predicting Wine Color & Quality With Discriminant Analysis

Jonathan Jiang

December 15, 2023

Table of Content

1 Introduction

The Wine data set contains information about six hundred randomly sampled Portuguese wines. The wines were tested for eleven continuous variables and two discrete, or grouping, variables. It contains no missing values.

Continuous:

- Fixed acidity (g(tartaric acid)/dm^3)
- Volatile acidity (g(acetic acid)/dm^3)
- Citric acid (g/dm^3)
- Residual sugar (g/dm^3)
- Chlorides (g(sodium chloride)/dm^3)
- Free sulfur dioxide (mg/dm^3)
- Total sulfur dioxide (mg/dm^3)
- Density (g/cm^3)
- pH
- Sulphates (g(potassium sulphate)/dm^3)
- Alcohol (vol.%)

Discrete:

- Quality
- Red (indicator: 1 if red, 0 if white)

Quality is the median grade assigned to each wine by sensory assessors who sampled the wines in a blind taste test. Individual assessors' grades for each wine could range between 0 (very bad) and 10 (excellent).

We will explore the relationships and determine what qualities matter in the creation of wine.

1.1 Questions of Interest and Methodology

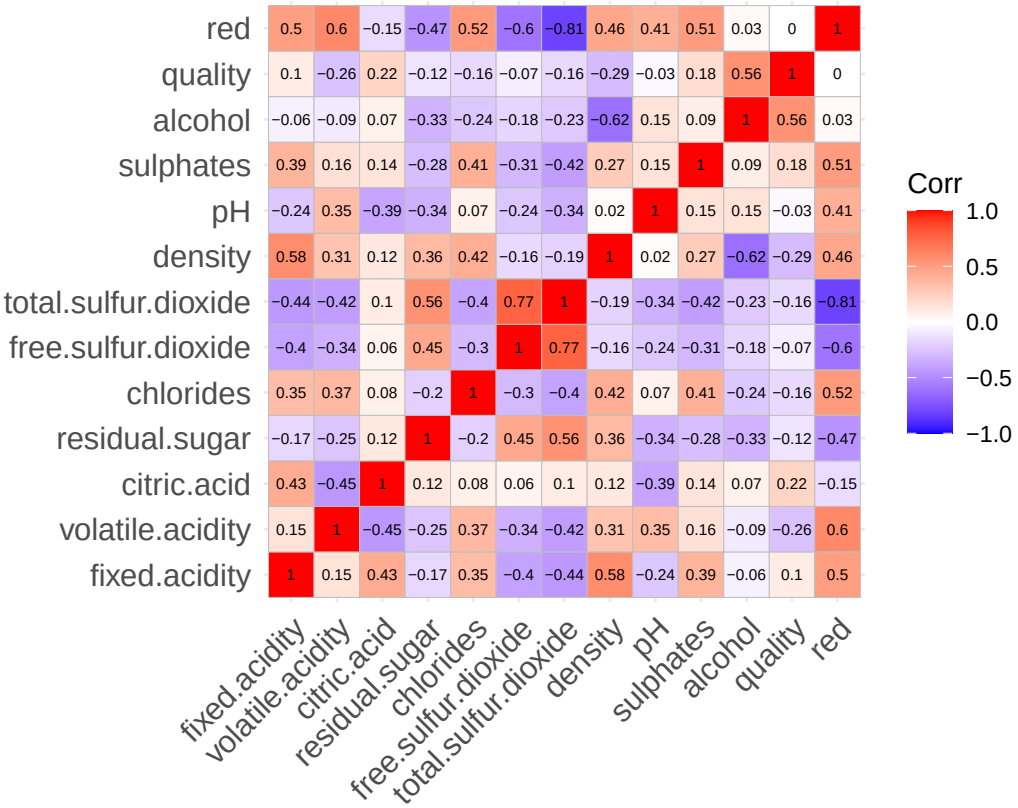
1. Which components of wine affect each other? We will be using correlation graphs to determine patterns/relationships and then expanding with scatter and histogram plots to further explore. We will also use K-means clustering to create clusters of possible relationships for choosing better quality.
2. Is this dataset a normal distribution? We will be using Principle Component Analysis (PCA), Marginal normal and Chi-squared Q-Q plots to determine whether or not this dataset is a normal distribution.
3. Can we predict color or quality of wine? We will be using Linear and Quadratic Discriminant Analysis to see if we can predict the color and quality of wine with the given data.

2 Preliminary Data Exploration

2.1 Summary of Data

To familiarize ourselves with the data and guide our research, we will first examine the correlations between all pairs of variables to see which have the strongest and weakest correlations. Then, we will conduct principal component analysis (PCA) on the continuous variables to determine what combinations of them explain the majority of the variation in the data.

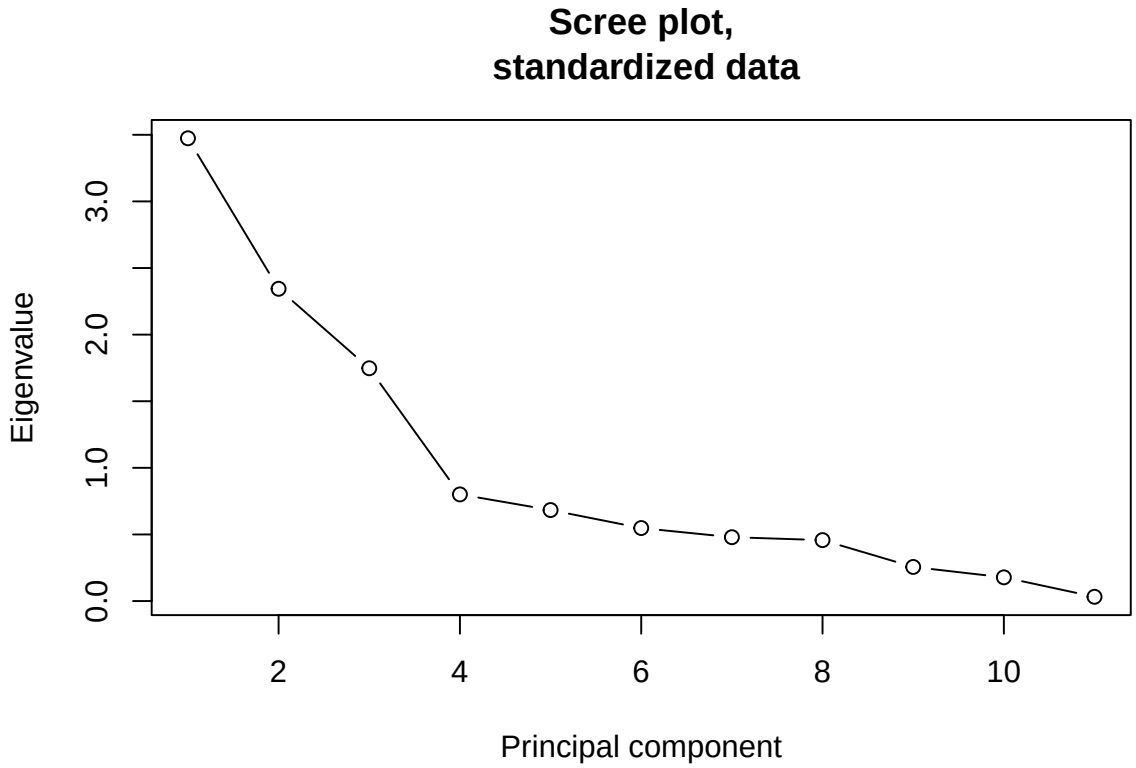
2.1.1 Correlation Plot



As shown with the correlation graph above, we can see that there are some high correlations; however, many of them are not valuable relationships. Free and Total sulfur dioxide are highly correlated; however, we should assume those could be similar as free sulfur dioxide is dependent on total sulfur dioxide ie there can't be more free gas than total. We can also see that density and alcohol have an inverse relationship; however, we know this is true as alcohol is less dense than water. Most variables are unfortunately not related to quality as we wished; however, there is at least one relationship we can explore further in plots.

Relationships that could prove useful to explore further based on high or low correlation: Red (Color) vs Volatile.Acidity, total.sulfur.dioxide vs Red (Color), alcohol vs density, and quality vs alcohol.

2.1.2 Principal Component Analysis

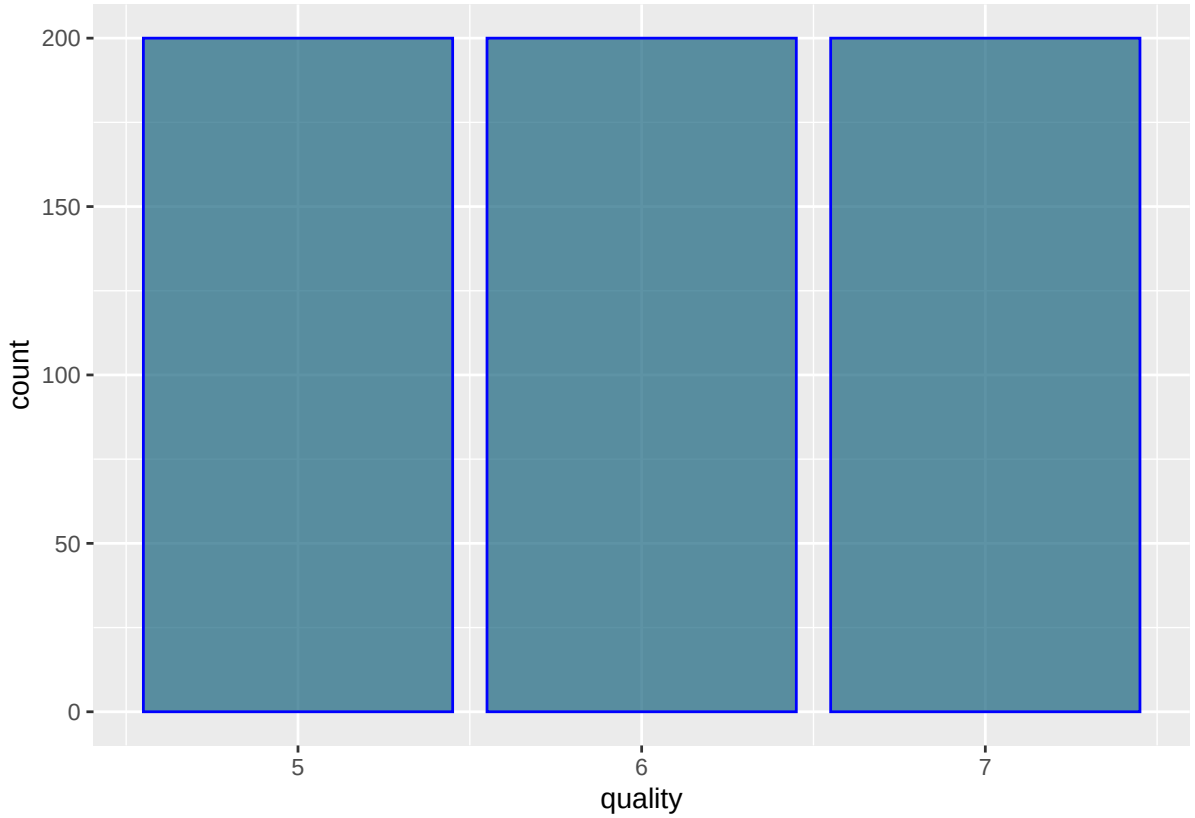


In the Scree plot above, we can see that the plot starts to form a straight line after the fourth principal component. This means that the first four principal components explain the majority of the variation. These four principal components explain about 76.05% of the variation in the data. As we look at the principal component analysis, we see that the first principal component does not have large association with any variable. The second principal component has a large positive association with density and a large negative association with alcohol. This suggests that as density increases, alcohol decreases. This corresponds to the correlation graph we created earlier as the relationship between density and alcohol was -0.62. The third principal component has a large negative association with citric acid. This component could be viewed as the measure of how citric acid negatively affects wine. Lastly, the fourth principal component has a large negative association with sulfates. This component also could be viewed as the measure of how sulfates negatively affects wine.

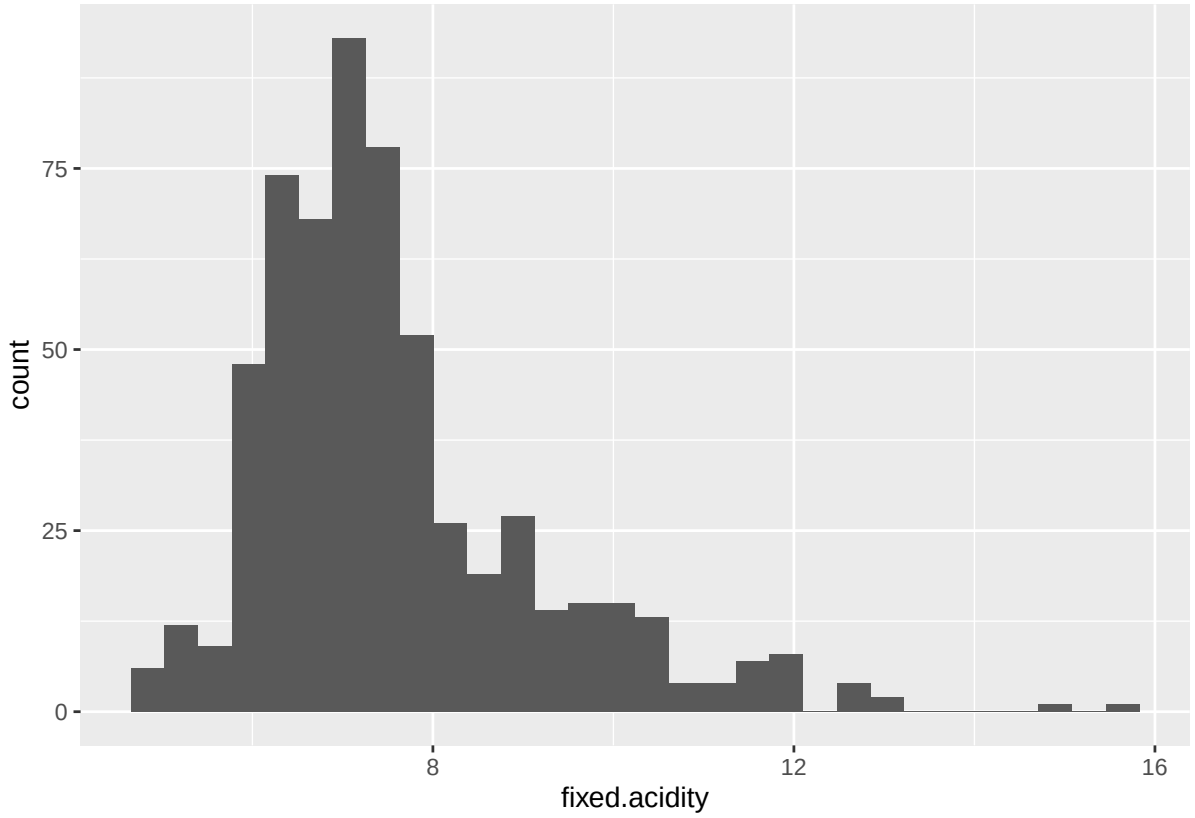
	PC1	PC2	PC3	PC4
fixed.acidity	-0.3148198	0.3573809	-0.2849295	0.2775527
volatile.acidity	-0.3223341	-0.0727480	0.4201880	0.1756126
citric.acid	0.0396185	0.3271767	-0.5458551	-0.1258108
residual.sugar	0.3042613	0.3581461	0.1942990	0.0482146
chlorides	-0.3379684	0.2206808	0.0925372	-0.3507942
free.sulfur.dioxide	0.4156757	0.1036857	0.1113214	-0.3992248
total.sulfur.dioxide	0.4681582	0.1276707	0.0919911	-0.1986879
density	-0.2173891	0.5068506	0.2681863	0.0488469
pH	-0.2004334	-0.3363338	0.3038913	-0.3466478
sulphates	-0.3290751	0.0950897	-0.1709109	-0.6553383

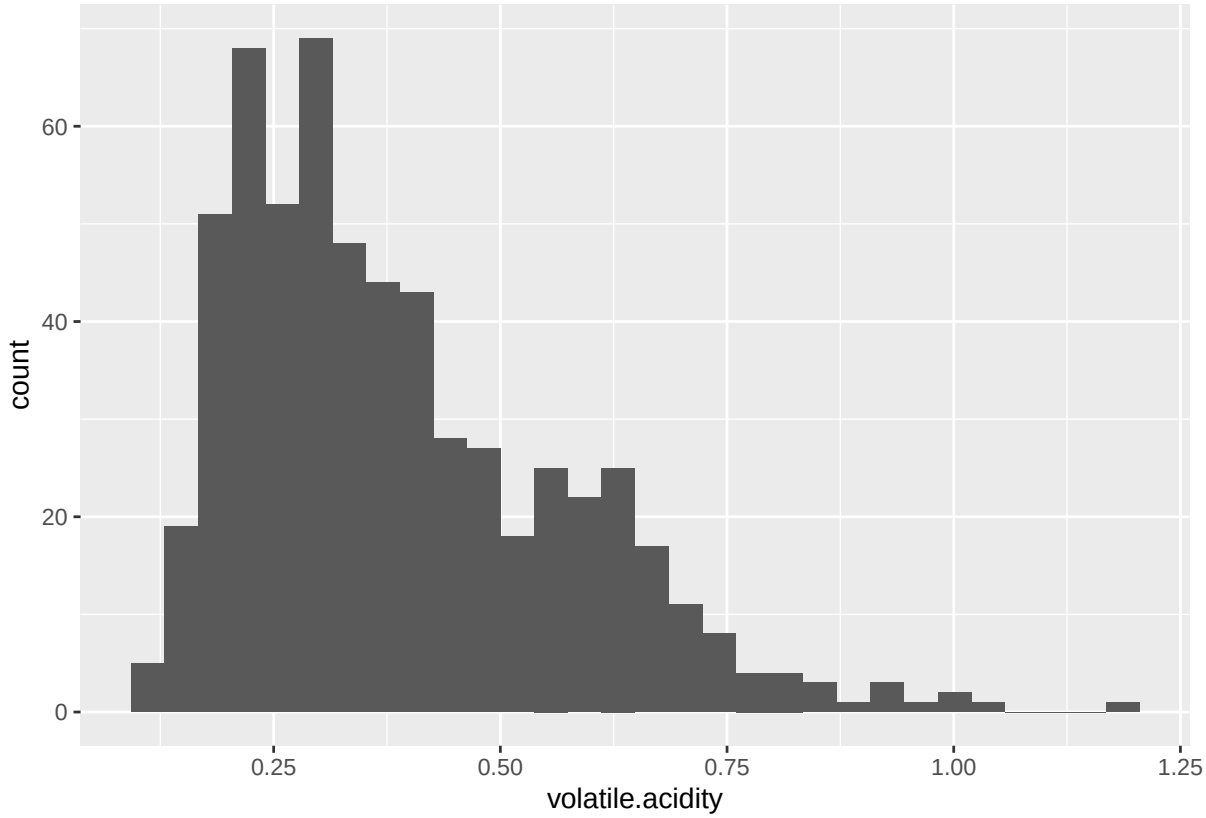
	PC1	PC2	PC3	PC4
alcohol	-0.0306873	-0.4205506	-0.4285383	-0.0062309

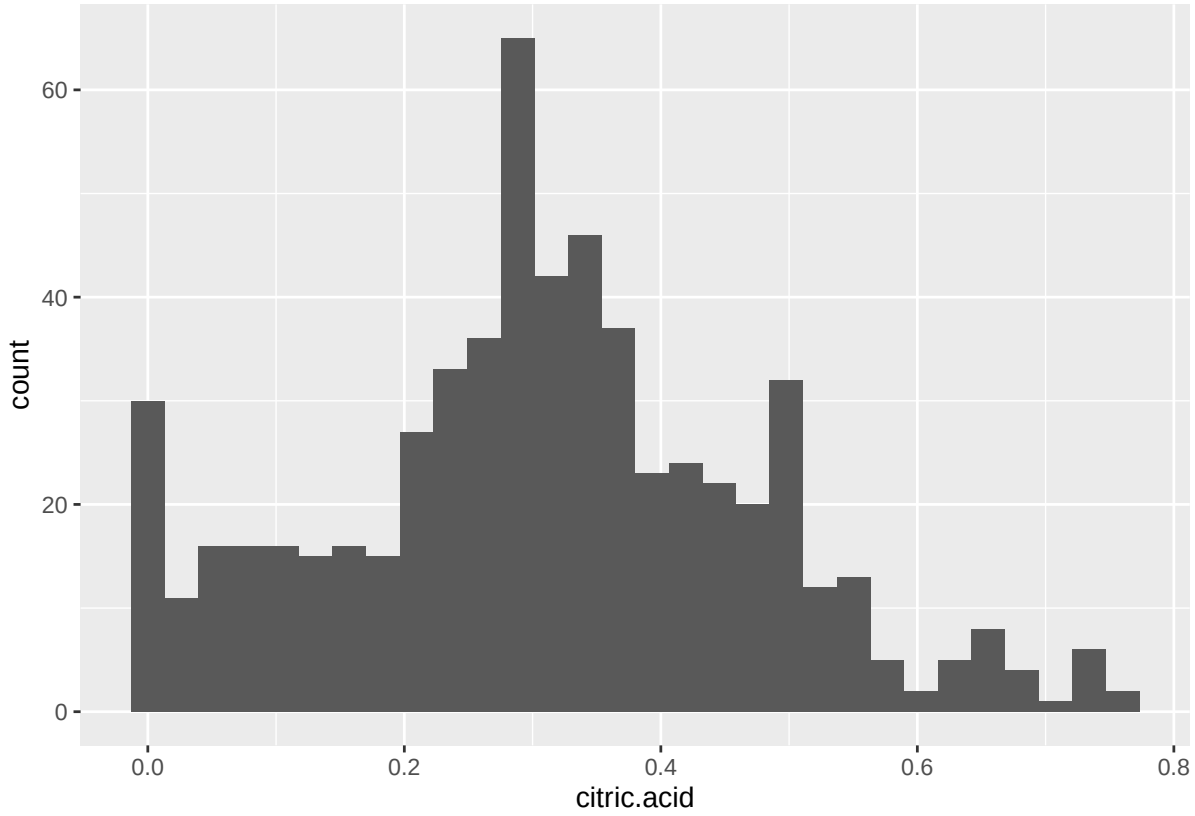
2.1.3 Additional Data Visualizations



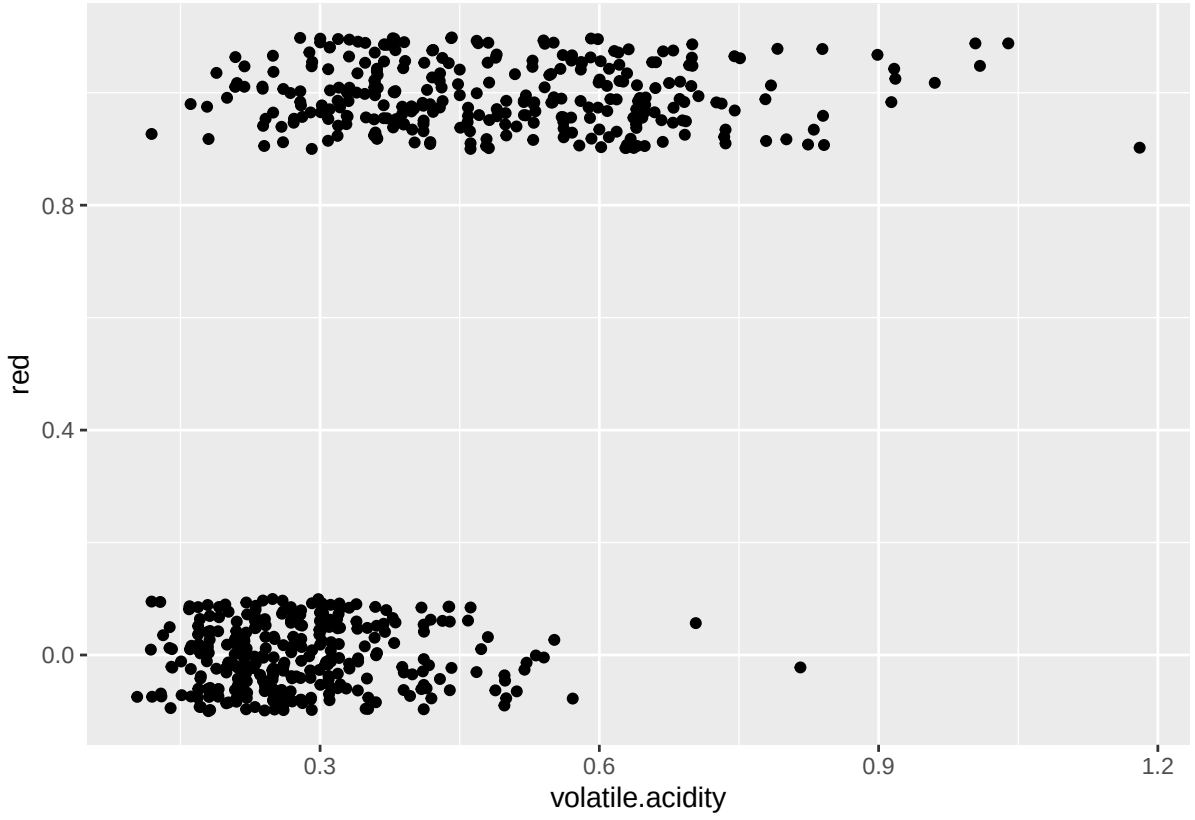
We can do a preliminary check on our data by graphing the first couple variables and quality in histograms. We can see that quality is 200 of 5-7. This is not ideal as 5 would be considered average quality and 7 would be higher quality wine. We don't have a reference to a "bad" wine; however, we should still be able determine what components are important in wine.


count
75
50
25
0
8
12
16
fixed.acidity

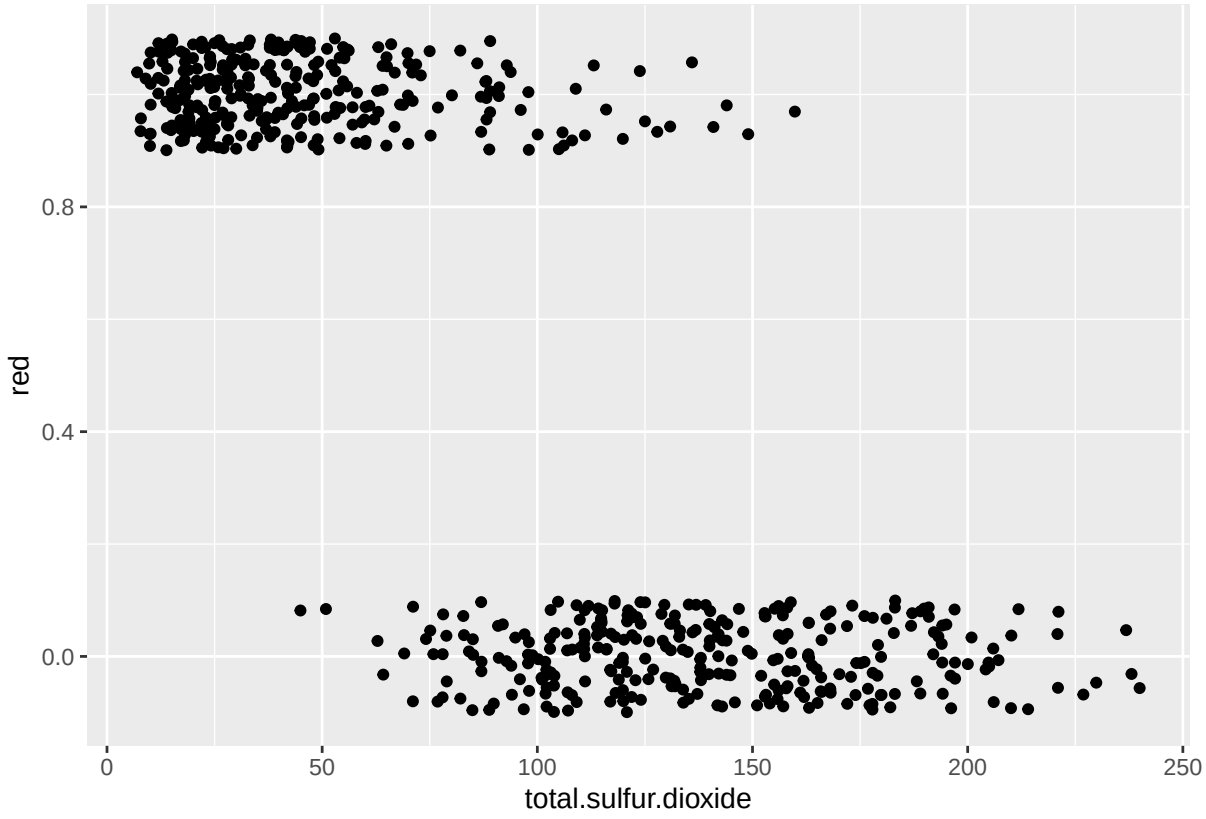

count
60
40
20
0
0.25
0.50
0.75
1.00
1.25
volatile.acidity



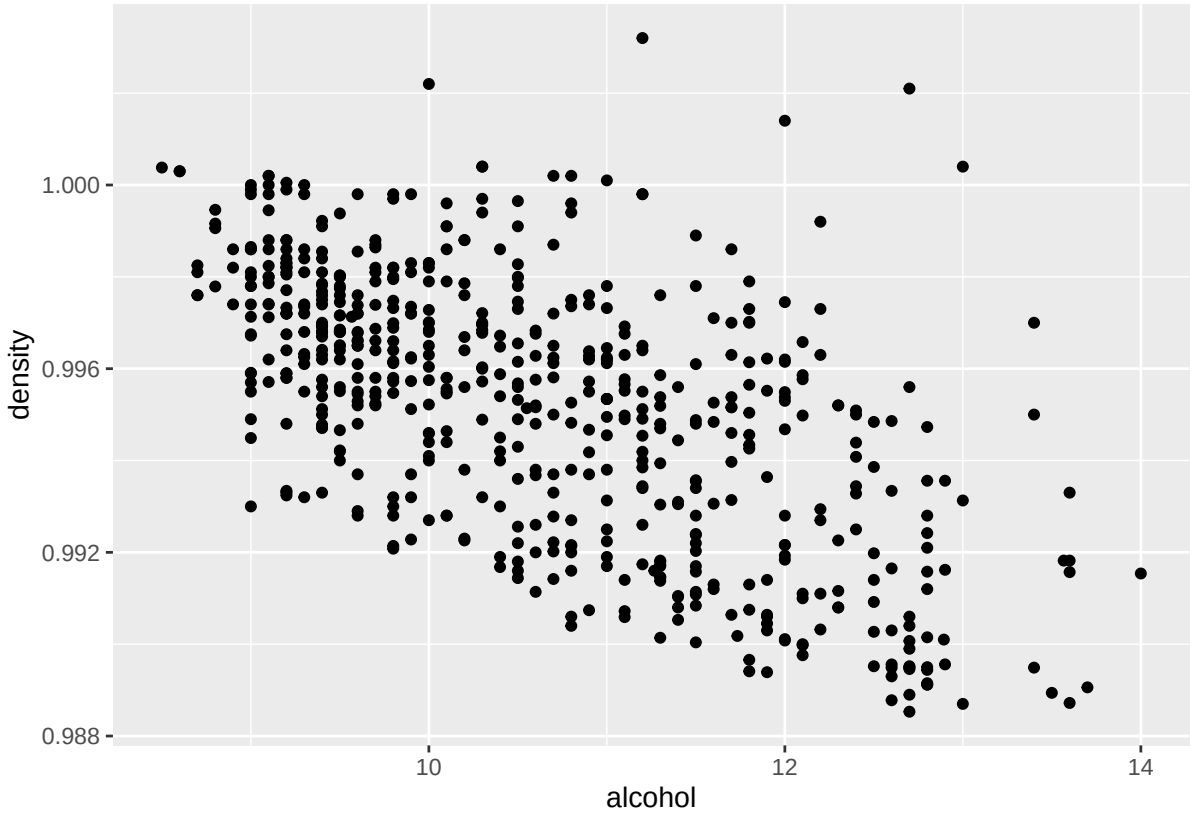
With the histograms above, we can see that Fixed acidity and Volatile acidity are skewed right with Citric Acid as a normal distribution. Nothing stands out so far; however, we will check for outliers and multi-variate normality later in the report with a Chi-squared test.



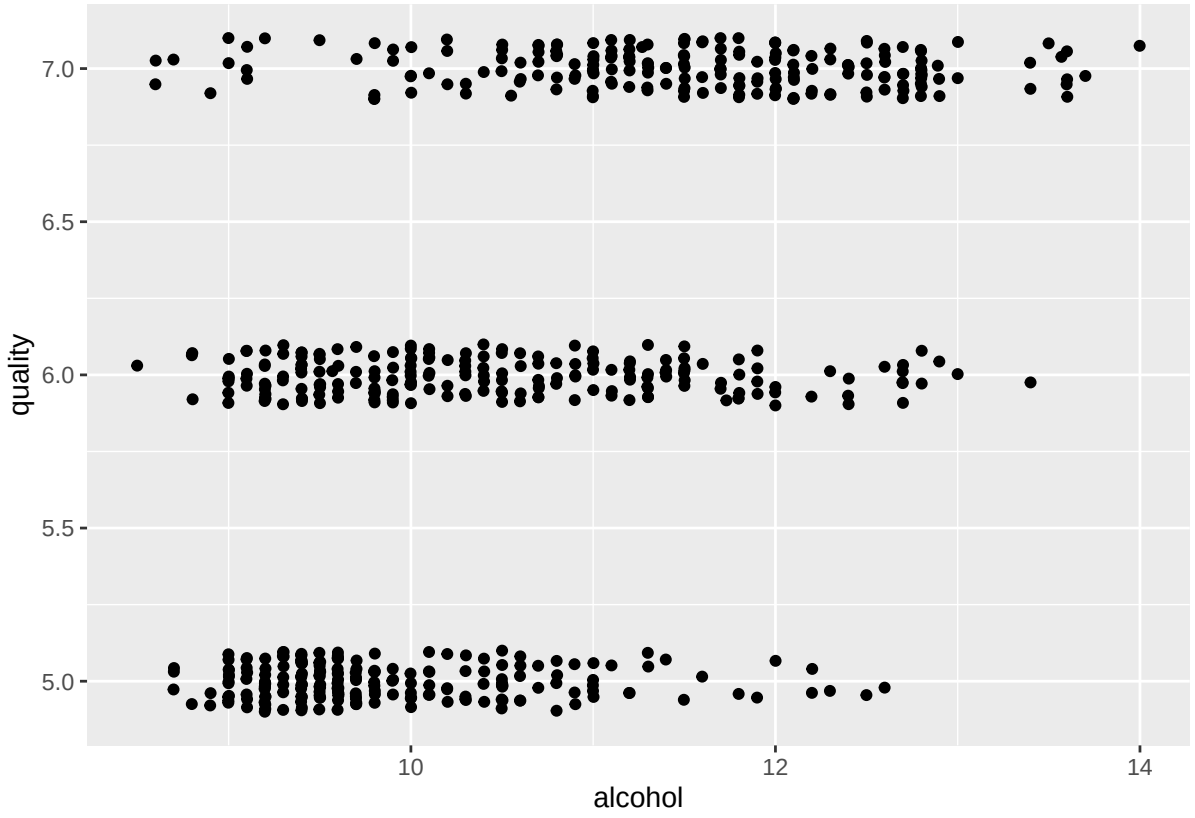
In the plot above, we can see that volatile.acidity can explain the color to some extent. We can see that a majority of white wine has low volatile.acidity while red wine's volatile acidity can vary greatly. We can also see that almost any wine above a 0.6 volatile.acidity has to be the color red. Volatile acidity refers to the steam-distillable acids present in wine, primarily acetic acid. Acetic acid is a byproduct of fermentation, normally associated with vinegar. This acid results in the production of other aroma compounds such as ethyl acetate and acetaldehyde, which can be unpleasant to smell (**?**). As volatile.acidity is related to the aftermath of fermentation, it is hard to come to a conclusion without knowing more about fermentation methods/process.



In the plot above, we can see that total.sulfur.dioxide can explain the color to a large extent. A majority of red wine has low total.sulfur.dioxide and a majority of white wine has a high total.sulfur.dioxide. Sulfur Dioxide is produced naturally in low amounts during fermentation; however, most sulfur dioxide is added by the winemaker. This compound is important as it's antimicrobial, preserving the wine beyond a couple years, and antioxidant as it protects fruit from browning. Furthermore, Sulfur dioxide can bind to the compounds that make red wine red, anthocyanins. Once bound, the compound gets "bleached" and turns into a much lighter color (**?**). This explains the pattern we see above.



In the plot above, we can see that alcohol can explain the density to some extent. We see that the points seem to follow a negative sloping line. Alcohol is less dense than water, so it would make sense that higher alcohol content means a lower density than water in wine; however, there are many other factors and ingredients in wine, so it would be foolish to assume that alcohol is the only reason.

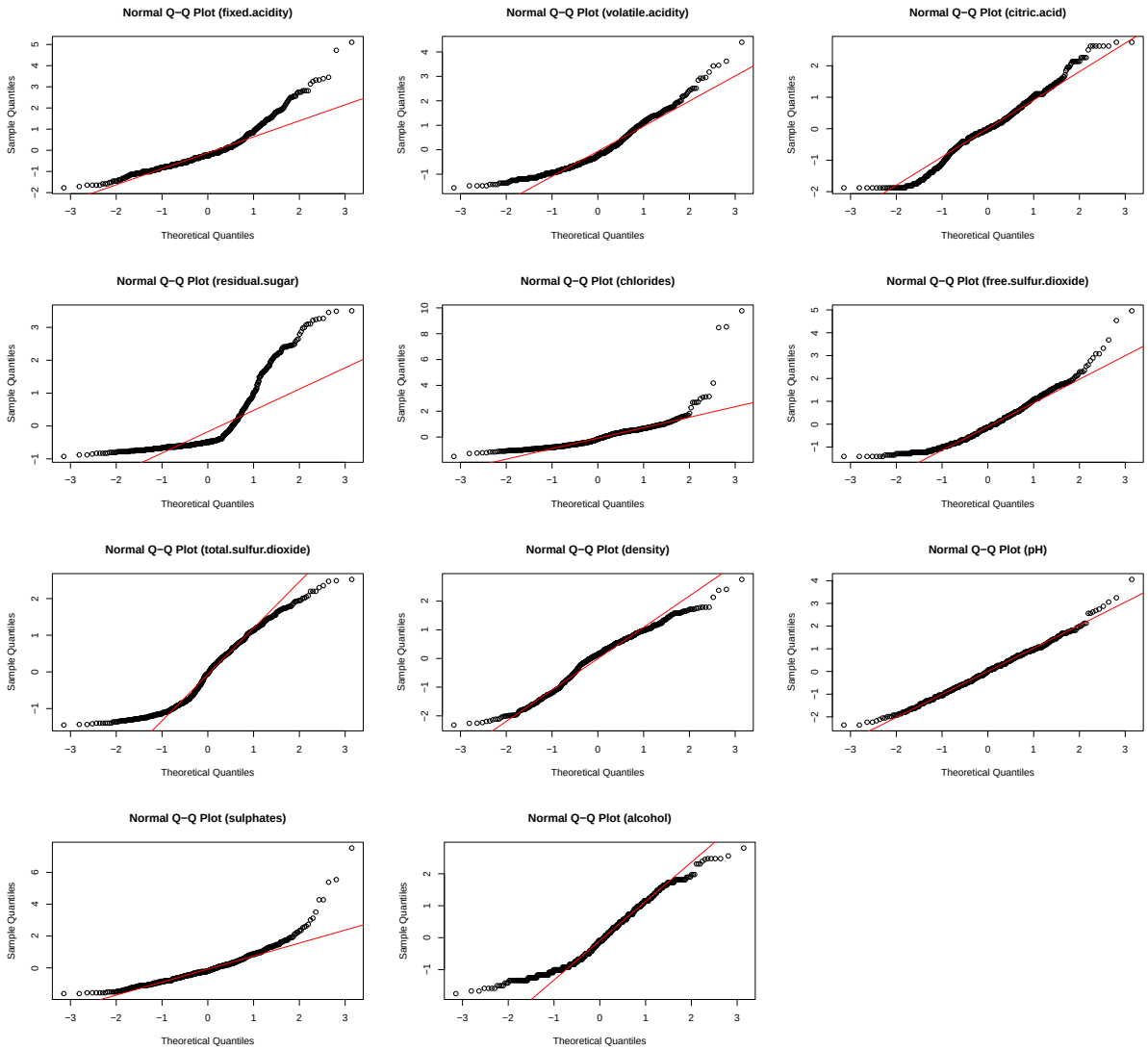


In the above graph, we can see a general trend of higher quality with higher % of alcohol. However, the variable of alcohol encompasses too large of a spectrum. Ethanol comprises the majority of the alcohol % in wine; however, higher alcohols, also known as fusel oils, are present as well and are created in small amounts by yeast during the fermentation process. Higher alcohols may have an aromatic effect and some can be considered positive, but others may become negative (**?**). We cannot come to a conclusion on whether or not alcohol % affects quality.

3 Analysis of Multi-Variate Normality

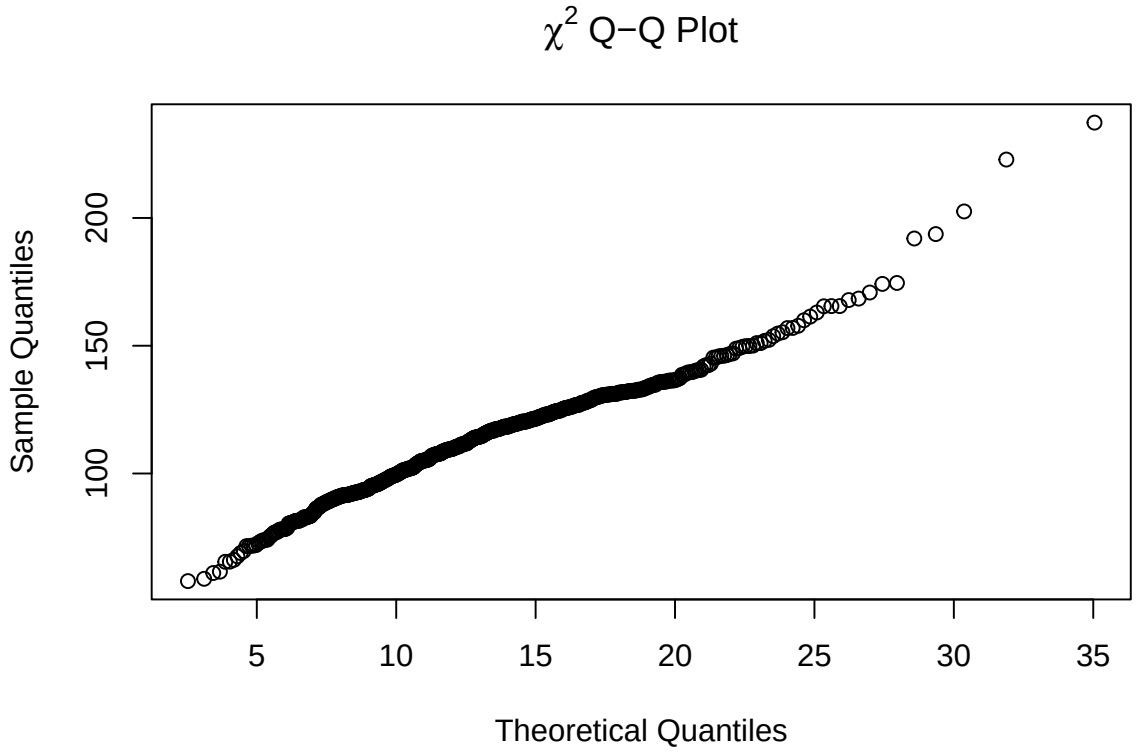
Before proceeding to any statistical tests or modelling, it's necessary that we check that the data follows a multivariate normal distribution. Because the sample size is large (600 observations total), the central limit theorem suggests we may already be able to assume normality, but we will still inspect marginal normal Q-Q plots and a chi-squared Q-Q plot.

3.1 Marginal Normal Q-Q Plots



While marginal Q-Q plots don't suggest whether a joint distribution will have a normal distribution or not, they're still beneficial for identifying potential outliers that could skew the joint distribution. Most of the marginal Q-Q plots suggest their associated variables do not follow a normal distribution, as indicated by how far the points deviate from a straight line. This is particularly evident for residual sugar and volatile acidity. On the other hand, the normal Q-Q plots for density and pH, and alcohol content appear approximately normal. Most of the points follow the line in the chlorides Q-Q plot, but it still contains a few significant outliers in the top right portion of the plot.

3.2 Chi-Squared Q-Q Plot



For the most part, points in the chi-squared Q-Q plot appear to follow a straight line, suggesting the data follows a multivariate normal distribution. Still, it should be noted that there are around five potential outliers in the top right portion of the plot, indicating that the associated observations have particularly large Mahalanobis distances relative to the chi-square quantile values.

4 Further Exploration of Grouping Variables

4.1 Hotelling's T^2 Hypothesis Test

Because the data includes categorical variables of color and quality, we are curious as to whether there is truly a difference between the populations of red and white wines, as well as wines of the highest quality (from the Wine data set, this is wine that received a median grade of a 7 out of 10) versus wines of lower quality.

The observations in the data set are already known to be independent and we have already concluded that the data follows a multivariate normal distribution, but we will also assume that, for each test, the data from both populations have the same variance-covariance matrix.

With all necessary assumptions out of the way, we completed two Hotelling's T^2 Hypothesis Tests: first for a difference in red and white wine populations, and then for a difference in high quality and lower quality wine populations. Both tests resulted in p-values of zero, meaning that if, in reality, the populations were the same, we would observe our data with a 0% chance. Therefore, red and white wines are distinct populations. Additionally, high quality wines and lower quality wines are distinct populations.

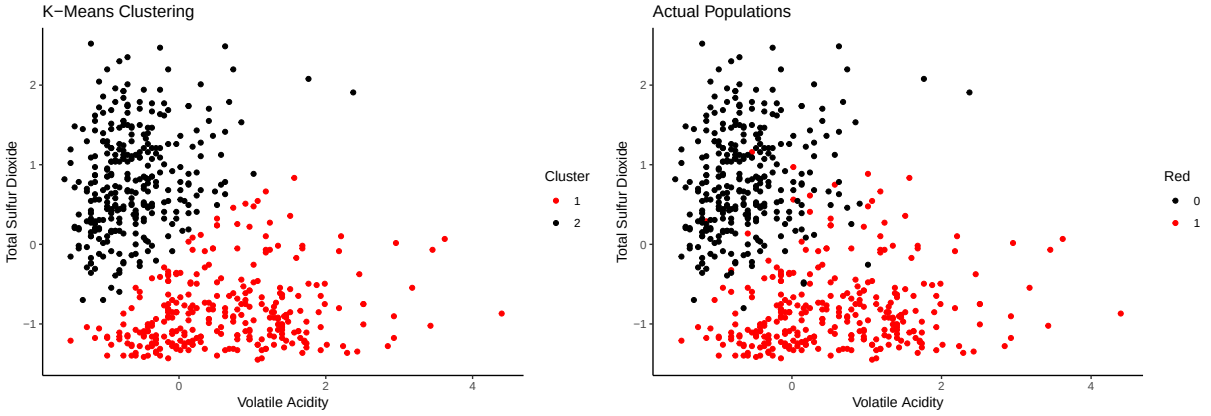
Because these tests are independent of each other, there is no need to apply a Bonferroni correction. We know that they are independent because the previously attained correlation plot showed that there is no correlation between red (the indicator variable for color) and quality.

Knowing that there are distinct populations of both different wine colors and quantities, we will now explore applications of machine learning algorithms.

4.2 K-Means Clustering

The Wine data set supplies labeled data, meaning that it can be used for supervised machine learning. We will use this to attempt to build models that can predict wine color and quality, but first, we are curious as to how the results of k-means clustering of the data based on the two continuous variables most correlated with each grouping variable compare to the actual populations. If the clusters closely match the true populations, the two continuous variables the clusters were built off of may be strong predictors of the color and quality of new wines.

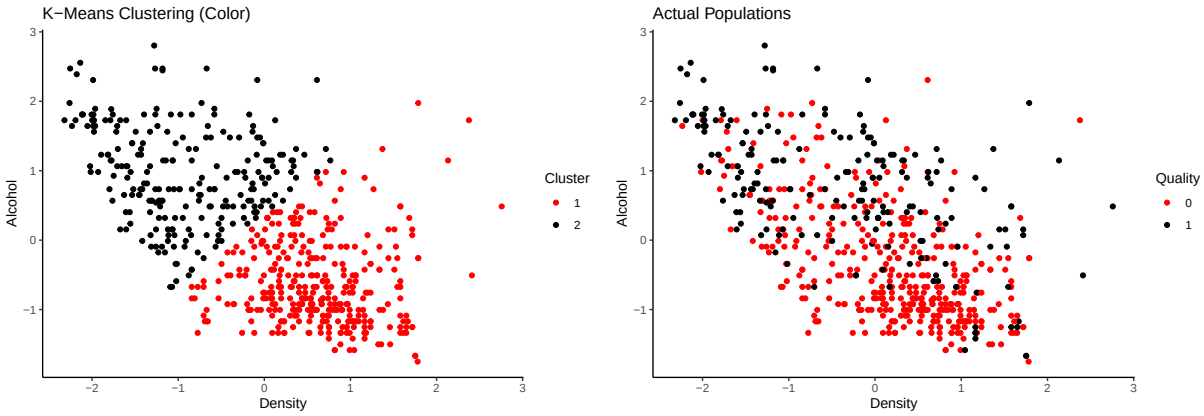
4.2.1 Wine Color



The earlier correlation plot showed that the indicator variable for color had the strongest correlation

with total sulfur dioxide (r = 0.81), followed by volatile acidity (r = -0.6) and free sulfur dioxide (r = 0.6). We opted to use volatile acidity rather than free sulfur dioxide despite their correlations with color being equal in magnitude. This is because free sulfur dioxide had a strong correlation with total sulfur dioxide (r = 0.77), meaning that pair would likely produce less meaningful results. From the similarity between the scatter plots above, it's clear that the k-means clustering method separated the populations of red and white wines quite well, with just a small number of incorrect assignments. We expect that similar success will come of our prediction models for wine color using discriminant analyses.

4.2.2 Wine Quality



The earlier correlation plot showed that the indicator variable for quality had the strongest correlation with total sulfur dioxide (r = 0.56), followed by volatile acidity (r = -0.29). Therefore, these variables were selected for use in the k-means cluster formation for color. Because the coloring of points on the scatter plots above differ greatly, we can conclude that the k-means clustering method does not distinguish between wines of the highest quality and others very well when relying on total sulfur dioxide and volatile acidity alone. The plot with the actual populations colored by quality shows that there is a significant amount of overlap in the populations, so it may also be difficult to accurately predict wine quality using discriminant analyses.

5 Predictive Modelling

5.1 Predicting Color

5.1.1 QDA Using All Continous Variables

With the results from the Chi-Squared plot, we are able to fulfill one of the two major assumptions required to use QDA, assumption of normal distribution. The next major assumption of equal variance-covariance matrices was fulfilled when we standardized the data, changing all the variances to the same. After dividing the dataset into two equal training and testing sets, we ran QDA to obtain an accuracy of about 98%. Instead of using the entire dataset, we will try using a different variable combination to see if accuracy can be increased.

5.1.2 QDA Using Two Continuous Variables

As shown in the graphs above, we tried to predict color in a different way. We divided the dataset based on continuous variables that have strongest correlation with color, and obtain an accuracy of about 95.6666667%. We see that the previous accuracy was about the same due to variance in accuracy calculation. This accuracy is good and there is no need to try Quadratic Discriminant Analysis; however, the more important grouping variable comes next. We'll predict quality next.

5.1.3 Model Comparison

Average accuracy over 5 iterations

- QDA using all continuous variables: 98%
- QDA using total sulfur dioxide & volatile acidity: 95.6666667%

5.2 Predicting Wine Quality

5.2.1 QDA Using All Continuous Variables

Here, we used Quadratic Discriminant Analysis to predict quality. After dividing the dataset into two equal training and testing sets, we ran QDA to obtain an accuracy of about 75.4%. This is much lower than the accuracy when predicting color, so we will try Linear Discriminant Analysis to obtain a better accuracy.

5.2.2 LDA Using All Continuous Variables

After dividing the dataset into two equal training and testing sets, we ran LDA to obtain an accuracy of about 78.4%. The accuracy is better; however, it is still not ideal. Instead of two equal training and testing sets, we will choose the continuous variables that have strongest correlation with quality to try to obtain a better accuracy.

5.2.3 LDA Using Three Continuous Variables

In the graphs above, we tried to predict quality. After dividing the dataset based on the continuous variables that have strongest correlation with quality, we obtained an accuracy of about

77.2666667% as well. Unfortunately, the accuracy is still the same.

5.2.4 Model Comparison

Average accuracy over 5 iterations:

- QDA using all continuous variables: 75.4%
- LDA using all continuous variables: 78.4%
- LDA using volatile acidity, density, and alcohol: 77.2666667%

6 Conclusion

Throughout this report, we have been trying to answer three main questions. Which components of wine affect each other?: From our correlation graph, we were able to find 4 possible relationships but in the end we found 2 relationships of potentially predicting color: Volatile Acidity and Total Sulfur Dioxide vs Color. The other relationships did not show any useful patterns.

Is this dataset a normal distribution?: Through Marginal Normal Q-Q Plots, we saw that the large majority had normal distributions but were slightly affected by outliers. With the Chi-Squared Q-Q Plot following, we were able to show that distribution of the dataset was normal.

Can we predict color or quality of wine?: With proving a normal distribution, we were allowed to use Hotelling T^2 Hypothesis Test, LDA, and QDA. With our prediction of the color of wine, we managed to achieve about a 96% accuracy with LDA and QDA. Our prediction of quality did not go as well as we managed to only get about 78% with LDA. Were we successful? Yes, but some other questions remain. We wanted to predict quality; however, the dataset only has 5,6,7. In a scale of 1-10, we can only judge mediocre wines against themselves. Without a larger variety of qualities, we are unable to predict wine quality accurately, only getting about 78%. Thankfully, there's only 2 possible colors and we were able to predict them extremely accurately at about 98% with prediction models.

7 Code Appendix

```
knitr::opts_chunk$set(echo = FALSE, message = FALSE,
  warning = FALSE, error = FALSE, cache = TRUE,
  fig.path='figs/', cache.path = 'cache/')
options(scipen=999)

# packages
library(tidyverse)
library(readxl)
library(caret)
library(viridis)
library(kableExtra)
library(ggeasy)
library(gridExtra)
library(ggcorrplot)
library(dplyr)
library(ggplot2)
library(MASS)
library(klaR)
library(Hotelling)


# importing wine data set
wine <- read.csv("wine.csv")


# sample correlation matrix
wine_corr = round(cor(wine),2)

ggcorrplot(wine_corr, lab = TRUE, lab_size = 2)


# standardizing the wine data
wine_standardized <- cbind(scale(wine[, 1:11]),
                           quality = wine[, 12],
                           red = wine[, 13])

# PCA
PCA <- prcomp(wine_standardized[, 1:11], retx = TRUE)

# PCs
PCs <- PCA$x

# scree plot for PCs
plot(PCA$sdev^2, type = "b", xlab = "Principal component",
     ylab = "Eigenvalue",
     main = "Scree plot,\n standardized data")
```

```r
# proportion of variance explained by first 4 PCs
prop_var_4PCs <- cumsum(PCA$sdev^2/sum(PCA$sdev^2))[4]


first_4_pcs_df = as.data.frame(PCA$rotation[, 1:4])
kable_table <- kable(first_4_pcs_df, "simple") %>%
  kable_styling()

kable_table


ggplot(wine, aes(x=quality)) + geom_bar(color="blue", fill=rgb(0.1,0.4,0.5,0.7))


ggplot(wine, aes(x=fixed.acidity)) + geom_histogram()

ggplot(wine, aes(x=volatile.acidity)) + geom_histogram()

ggplot(wine, aes(x=citric.acid)) + geom_histogram()

##Fixed acidity and Volatile acidity are skewed right
##Citric Acid is normal dist


#Volatile Acidity vs Red
ggplot(wine, aes(x = volatile.acidity, y= red)) + geom_jitter(height = 0.1)


#Total Sulfur Dioxide vs Red
ggplot(wine, aes(x = total.sulfur.dioxide, y = red)) + geom_jitter(height = 0.1)


#Alcohol vs Density
ggplot(wine, aes(x = alcohol, y = density)) + geom_point()


#Alcohol vs Quality
ggplot(wine, aes(x = alcohol, y = quality)) + geom_jitter(height = 0.1)


#par(mfrow = c(1, 2))
for (i in 1:11) {
  title = paste("Normal Q-Q Plot (", colnames(wine_standardized)[i], ")", sep = "")
  qqnorm(wine_standardized[, i], main = title)
  qqline(wine_standardized[, i], col = "red")
}


## chi-squared Q-Q plot
n <- dim(wine_standardized)[1]
```

```r
p <- dim(wine_standardized)[2]
S <- cov(wine_standardized)

par(mfrow = c(1, 1))
theoQ <- qchisq(((1:n) - 0.5)/n, p)
sampQ <- sort(diag(wine_standardized %*% solve(S) %*% t(wine_standardized)))
plot(theoQ, sampQ, xlab = "Theoretical Quantiles", ylab = "Sample Quantiles",
main = expression(paste(chi^2, " Q-Q Plot")))


wine_for_color = as.data.frame(wine_standardized[, -12])

red <- wine_for_color[wine_for_color$red == 1, -12]
white <- wine_for_color[wine_for_color$red == 0, -12]

H_col <- hotelling.test(red, white)

wine_for_quality = as.data.frame(wine_standardized) %>%
  mutate(max_quality = if_else(quality == max(quality), 1, 0)) %>%
  dplyr::select(-red, -quality)

high_quality <- wine_for_quality[wine_for_quality$max_quality == 1, -12]
other_quality <- wine_for_quality[wine_for_quality$max_quality == 0, -12]

H_qual <- hotelling.test(high_quality, other_quality)


# convert to data frame
wine_standardized = as.data.frame(wine_standardized)

# into two clusters:
X <- wine_standardized[, c(2,7)] # highest corrs w/ color

cls <- kmeans(X, 2)

y <- as.numeric(wine_standardized[,13]==1)

clustered_data <- data.frame(X, Cluster = as.factor(cls$cluster))
ggplot(clustered_data, aes(x = X[, 1], y = X[, 2], color = Cluster)) +
  geom_point() +
  scale_color_manual(values = c("red", "black")) +  # Set consistent colors
  ggtitle("K-Means Clustering") +
  labs(x = "Volatile Acidity",
       y = "Total Sulfur Dioxide") +
  theme_classic()


ggplot(wine_standardized, aes(x = volatile.acidity, y = total.sulfur.dioxide,
                              color = as.factor(red))) +
  geom_point() +
```

```
  scale_color_manual(values = c("black", "red")) +
  ggtitle("Actual Populations") +
  labs(color = "Red",
       x = "Volatile Acidity",
       y = "Total Sulfur Dioxide") +
  theme_classic()


# create new df with indicator column
# max_quality = 1 if observation is of highest quality
# max_quality = 0 else
wine_standardized1 = wine_standardized %>% mutate(max_quality =
                                                    if_else(quality ==max(quality),
                                                            1, 0))

X <- wine_standardized1[,c(8,11)] # highest corrs w/ quality

# into two:
cls <- kmeans(X, 2)


y <- as.numeric(wine_standardized1[,12]==max(wine_standardized1[,12]))

clustered_data <- data.frame(X, Cluster = as.factor(cls$cluster))
ggplot(clustered_data, aes(x = X[, 1], y = X[, 2], color = Cluster)) +
  geom_point() +
  scale_color_manual(values = c("red", "black")) +  # Set consistent colors
  ggtitle("K-Means Clustering (Color)") +
  labs(x = "Density",
       y = "Alcohol") +
  theme_classic()


ggplot(wine_standardized1, aes(x = density, y = alcohol, color = as.factor(max_quality))) +
  geom_point() +
  scale_color_manual(values = c("red", "black")) +
  ggtitle("Actual Populations") +
  labs(color = "Quality",
       x = "Density",
       y = "Alcohol") +
  theme_classic()


# initialize vector to store accuracy of each iteration
c_qda_allvar_accuracies <- numeric(5)

for (i in 1:5) {
  set.seed(i)

  # index for training set
```

```r
  # note: 1/2 of data is being used for each set
  ind = sample(dim(wine_standardized)[1], round(dim(wine_standardized)[1]/2))

  # partitioning data into the sets
  training <- wine_standardized[ind, ]
  testing <- wine_standardized[-ind, ]

  # training model
  qda_model <- qda(red ~ ., data = training)

  # testing model
  prediction = predict(qda_model, newdata = testing)

  # confusion matrix
  tab <- table(Predicted = prediction$class, testing$red)

  misclass_rt = 1-sum(diag(tab))/sum(tab)
  accuracy = sum(diag(tab))/sum(tab)
  c_qda_allvar_accuracies[i] = accuracy
}

# Calculate overall average accuracy
average_accuracy <- mean(c_qda_allvar_accuracies)*100

# partitionplot = partimat(wine_standardized[,6:9], as.factor(wine_standardized$red), method


X <- wine_standardized[, c(2,7)] # continuous vars that have strongest corre w/ color
y <- as.numeric(wine_standardized[,13]=="1")
## y = 0 => white
## y = 1 => red

# initialize vector to store accuracy of each iteration
c_qda_2var_accuracies <- numeric(5)

for (i in 1:5) {
  set.seed(i)

  # index for training set
  # note: 1/2 of data is being used for each set
  ind = sample(dim(wine_standardized)[1], round(dim(wine_standardized)[1]/2))
  Xtraining <- X[ind,]
  Xtesting <- X[-ind,]
  ytraining <- y[ind]
  ytesting <- y[-ind]

  qdaFit <- qda(Xtraining, ytraining)
  qdaFit

  prd <- predict(qdaFit, Xtesting)
```

```
  # confusion matrix
  tab <- table(Predicted = prd$class, Actual = ytesting)

  misclass_rt = 1-sum(diag(tab))/sum(tab)
  accuracy = sum(diag(tab))/sum(tab)

  c_qda_2var_accuracies[i] = accuracy
}

# Calculate overall average accuracy
average_accuracy <- mean(c_qda_2var_accuracies)*100

# partitionplot = partimat(X, as.factor(y), method = "qda")


# convert quality to binary and remove irrelevant columns
wine_standardized1 = wine_standardized %>% mutate(max_quality =
                                                    if_else(quality == max(quality),
                                                            1, 0)) %>%
  dplyr::select(-red, -quality)

# initialize vector to store accuracy of each iteration
q_qda_allvar_accuracies <- numeric(5)

for (i in 1:5) {
  set.seed(i)

  # index for training set
  # note: 1/2 of data is being used for each set
  ind = sample(dim(wine_standardized1)[1], round(dim(wine_standardized1)[1]/2))

  # partitioning data into the sets
  training <- wine_standardized1[ind, ]
  testing <- wine_standardized1[-ind, ]

  # training model
  qda_model <- qda(max_quality ~ ., data = training)

  # testing model
  prediction = predict(qda_model, newdata = testing)

  # confusion matrix
  tab <- table(Predicted = prediction$class, testing$max_quality)

  misclass_rt = 1-sum(diag(tab))/sum(tab)
  accuracy = sum(diag(tab))/sum(tab)

  q_qda_allvar_accuracies[i] = accuracy
}
```

```r
# Calculate overall average accuracy
average_accuracy <- mean(q_qda_allvar_accuracies)*100

# partitionplot = partimat(wine_standardized[,6:9], as.factor(wine_standardized1$max_quality


# initialize vector to store accuracy of each iteration
q_lda_allvar_accuracies <- numeric(5)

for (i in 1:5) {
  set.seed(i)

  # index for training set
  # note: 1/2 of data is being used for each set
  ind = sample(dim(wine_standardized1)[1], round(dim(wine_standardized1)[1]/2))

  # partitioning data into the sets
  training <- wine_standardized1[ind, ]
  testing <- wine_standardized1[-ind, ]

  # training model
  lda_model <- lda(max_quality ~ ., data = training)

  # testing model
  prediction = predict(lda_model, newdata = testing)

  # confusion matrix
  tab <- table(Predicted = prediction$class, testing$max_quality)

  misclass_rt = 1-sum(diag(tab))/sum(tab)
  accuracy = sum(diag(tab))/sum(tab)

  q_lda_allvar_accuracies[i] = accuracy
}

# Calculate overall average accuracy
average_accuracy <- mean(q_lda_allvar_accuracies)*100

# partitionplot = partimat(wine_standardized[,6:9], as.factor(wine_standardized1$max_quality


X <- wine_standardized[,c(2,8,11)] # continuous vars that have strongest corr w/ quality
y <- as.numeric(wine_standardized[,12]==max(wine_standardized[,12])) # highest quality when

# initialize vector to store accuracy of each iteration
q_lda_3var_accuracies <- numeric(5)

for (i in 1:5) {
  set.seed(i)
```

```
  # index for training set
  # note: 1/2 of data is being used for each set
  ind = sample(dim(wine_standardized)[1], round(dim(wine_standardized)[1]/2))
  Xtraining <- X[ind,]
  Xtesting <- X[-ind,]
  ytraining <- y[ind]
  ytesting <- y[-ind]

  ldaFit <- lda(Xtraining, ytraining)
  ldaFit

  prd <- predict(ldaFit, Xtesting)

  # confusion matrix
  tab <- table(Predicted = prd$class, Actual = ytesting)

  misclass_rt = 1-sum(diag(tab))/sum(tab)
  accuracy = sum(diag(tab))/sum(tab)

  q_lda_3var_accuracies[i] = accuracy
}

average_accuracy <- mean(q_lda_3var_accuracies)*100

# partitionplot = partimat(X, as.factor(y), method = "lda")
```