Wine Quality Analysis

Lori Stevens

05/15/2020

Table of Contents

1 Introduction 2
2 Analysis 2
  2.1 Data Creation 2
  2.2 Data Visualization 6
    2.2.1 Distribution 6
    2.2.2 Fixed Acidity 13
    2.2.3 Volatile Acidity 14
    2.2.4 Citric Acid 15
    2.2.5 Residual Sugar 16
    2.2.6 Chlorides 17
    2.2.7 Free Sulfur Dioxide 18
    2.2.8 Total Sulfur Dioxide 19
    2.2.9 Density 20
    2.2.10 pH 21
    2.2.11 Sulphates 22
    2.2.12 Alcohol 23
  2.3 Data Modeling 24
    2.3.1 Logistic Regression 27
    2.3.2 LDA 28
    2.3.3 QDA 28
    2.3.4 Loess 29
    2.3.5 KNN 29
    2.3.6 Random Forest 30
    2.3.7 Ensemble 31
3 Results 32
4 Conclusion 33

1 Introduction

The quality of wine is difficult to define, as many aspects are subjective. However, wine quality is critical in the industry and is monitored in many stages of the product lifecycle by winemakers, distributors, merchants, critics, and sommeliers. Quality assurance teams conduct physiochemical and sensory analysis to ensure quality. In order to do this, it is critical to define what attributes and levels constitute quality. Statistical modeling techniques can be used to understand and predict wine quality based on observed characteristics.

This report details the evaluation of statistical models to predict wine quality using the Wine Quality datasets available in the UCI repository. The datasets are inclusive of "Vinho Verde" red and white wine from Portugal. The objective is to evaluate models to predict wine quality utilizing the Wine Quality datasets. This is a collection of 6,500 ratings released in October 2009. Variables are described in detail in the **Data Creation** section. The full dataset is available at the following location:

https://archive.ics.uci.edu/ml/datasets/Wine+Quality

As the data is limited to Vinho Verde, which is most commonly white, the final model would need to be further evaluated on a broader dataset to ensure the same characteristics and levels can be assumed on other varietals.

Seven statistical models are created, and levels of accuracy compared in this report to determine the best model to predict quality. Most code is referenced directly in this report to allow for easy review of approach, findings, and code.

2 Analysis

This section explains the process and techniques used, including data creation, exploration, visualization, modeling approach, and insights.

2.1 Data Creation

The first step is to create the datasets. The dataset is split into a train, test, and validation dataset. The training dataset is 70% of the dataset to ensure that the model is trained with a large subset of data. The test and validation datasets are equally split with the remainder of the dataset. The test dataset is used to determine the best model, and the validation dataset is used for the final evaluation of the model.

The final validation set is important as the process of selecting the best model may cause overfitting. To get a reliable estimate of how effective the model will be on future data, an incremental dataset is introduced to validate.

```r
# Install Packages
if(!require(dplyr)) install.packages("dplyr")
if(!require(tidyverse)) install.packages("tidyverse")
if(!require(tidyr)) install.packages("tidyr")
if(!require(stringr)) install.packages("stringr")
if(!require(ggplot2)) install.packages("ggplot2")
if(!require(caret)) install.packages("caret")
if(!require(randomForest)) install.packages("randomForest")
if(!require(matrixStats)) install.packages("matrixStats")
if(!require(data.table)) install.packages("data.table")

# Load libraries
library(dplyr)
library(tidyverse)
library(tidyr)
library(stringr)
library(ggplot2)
library(caret)
library(randomForest)
library(matrixStats)
library(data.table)

# Download the wine quality datasets
wine_quality_red <- read.csv("https://archive.ics.uci.edu/ml/machine-
learning-databases/wine-quality/winequality-red.csv", sep = ";") %>%
  mutate(type = "Red")
wine_quality_white <- read.csv("https://archive.ics.uci.edu/ml/machine-
learning-databases/wine-quality/winequality-white.csv", sep = ";") %>%
  mutate(type = "White")

# Combine wine quality datasets
wine_quality <- bind_rows(wine_quality_red, wine_quality_white)
rm(wine_quality_red, wine_quality_white)

# Create test dataset (30% of dataset)
set.seed(1, sample.kind="Rounding")
test_index <- createDataPartition(y = wine_quality$quality, times = 1, p =
0.3, list = FALSE)
test_set <- wine_quality[test_index,]

# Create train dataset (70% of dataset)
train_set <- wine_quality[-test_index,]
rm(test_index)

# Create validation and test datasets (equally split)
set.seed(1, sample.kind="Rounding")
test_index <- createDataPartition(y = test_set$quality, times = 1, p = 0.5,
list = FALSE)
validation <- test_set[test_index,]
```

```r
test_set <- test_set[-test_index,]
rm(test_index)
```

The following variables are included in the dataset:

1. Fixed Acidity(gm/dm^3): the amount of tartaric acid; the primary acidic component in wine grapes; high levels can result in a sour taste.
2. Volatile Acidity(gm/dm^3): the amount of acetic acid; high levels can result in an unpleasant, vinegar taste.
3. Citric Acid(gm/dm^3): the amount of citric acid; often added to wine to increase acidity and add a fresh flavor.
4. Residual Sugar(gm/dm^3): the amount of sugar remaining after fermentation; high levels result in a sweet taste.
5. Chlorides(gm/dm^3): the amount of sodium chloride; used to improve fermentation; high levels can be due to soil and can result in a salty taste.
6. Free Sulfur Dioxide(mg/dm^3): the amount of SO2 that is not bound to other molecules; additive to kill unwanted bacteria and prevent oxidation.
7. Total Sulfur Dioxide(mg/dm^3): the total amount of SO2; high levels can result in a change in taste and color.
8. Density(g/cm^3): the total mass per volume of alcohol, sugar, and water; the more alcohol a wine contains should decrease the overall density.
9. pH: a measurement of acidity; low pH wines can taste crisp, and high pH wines can taste sour and are more susceptible to bacterial growth.
10. Sulphates(g/dm^3): the amount of potassium sulphate; additive which acts as a preservative; high levels can result in a sulfur taste.
11. Alcohol(*% vol*): the percent alcohol content; a higher ABV typically results in a bolder and richer taste.
12. Quality(*1-10*): the rating based on sensory data on a 10-point scale; the average rating given is 5.8, the lowest rating is 3, and the highest rating is 9.
13. Type: classification of wine as red or white; 75% of the dataset is white wine given that Vinho Verde is most commonly white.

Due to privacy issues, important characteristics such as brand, vintage, and cost were not included in the dataset. Hence, these cannot be considered in the analysis.

An example of the data is as follows:

```r
# Generate sample records of dataset
head(wine_quality)
```

```
##   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 type
## 1       5  Red
## 2       5  Red
## 3       5  Red
## 4       6  Red
## 5       5  Red
## 6       5  Red
```

A summary of the dataset validates that there are no missing values. Quality is on a 10-point scale and assigned in whole numbers only.

```
# Generate summary of dataset
summary(wine_quality)
```

```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2500   1st Qu.: 1.800
##  Median : 7.000   Median :0.2900   Median :0.3100   Median : 3.000
##  Mean   : 7.215   Mean   :0.3397   Mean   :0.3186   Mean   : 5.443
##  3rd Qu.: 7.700   3rd Qu.:0.4000   3rd Qu.:0.3900   3rd Qu.: 8.100
##  Max.   :15.900   Max.   :1.5800   Max.   :1.6600   Max.   :65.800
##    chlorides        free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.00900   Min.   :  1.00      Min.   :  6.0        Min.   :0.9871
##  1st Qu.:0.03800   1st Qu.: 17.00      1st Qu.: 77.0        1st Qu.:0.9923
##  Median :0.04700   Median : 29.00      Median :118.0        Median :0.9949
##  Mean   :0.05603   Mean   : 30.53      Mean   :115.7        Mean   :0.9947
##  3rd Qu.:0.06500   3rd Qu.: 41.00      3rd Qu.:156.0        3rd Qu.:0.9970
##  Max.   :0.61100   Max.   :289.00      Max.   :440.0        Max.   :1.0390
##        pH          sulphates          alcohol          quality
##  Min.   :2.720   Min.   :0.2200   Min.   : 8.00   Min.   :3.000
##  1st Qu.:3.110   1st Qu.:0.4300   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.210   Median :0.5100   Median :10.30   Median :6.000
##  Mean   :3.219   Mean   :0.5313   Mean   :10.49   Mean   :5.818
##  3rd Qu.:3.320   3rd Qu.:0.6000   3rd Qu.:11.30   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :9.000
##      type
##  Length:6497
##  Class :character
##  Mode  :character
##
##
##
```

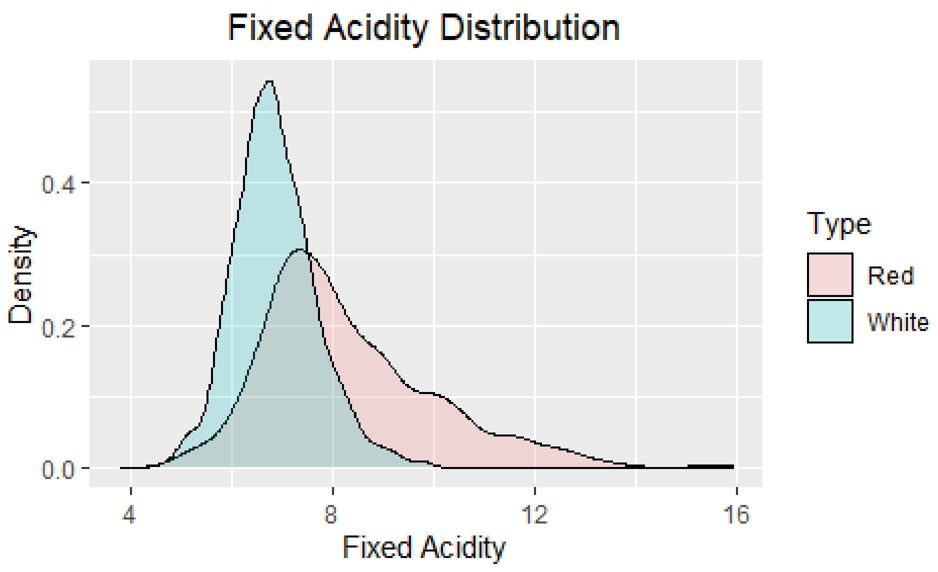
2.2 Data Visualization

In this section, the data is evaluated to determine distribution and any correlation in variables to quality. Data exploration is conducted on the full Wine Quality dataset.

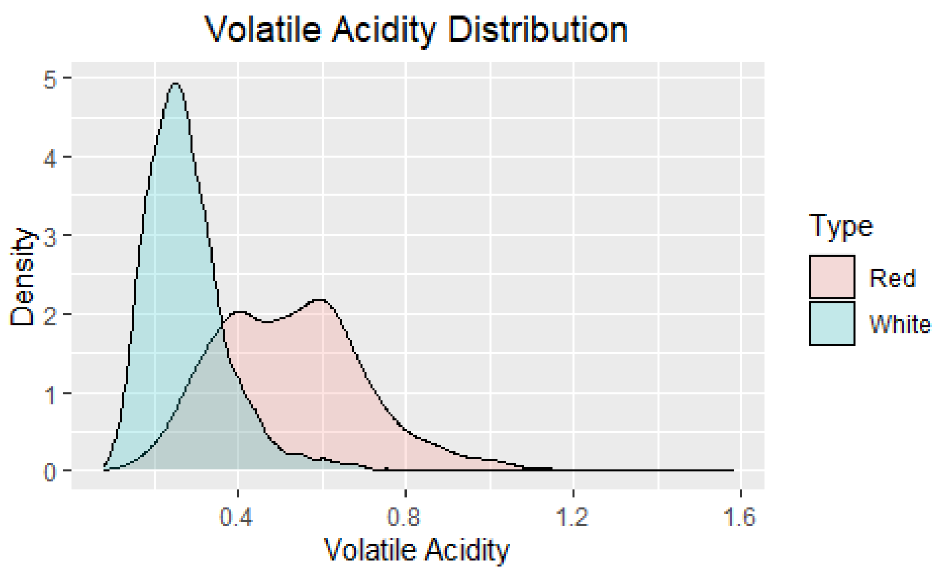
2.2.1 Distribution

Overall, distribution appears normal for many variables. Citric acid, residual sugar, density and ABV have abnormal distributions. In most variables, there is a shift in levels between red and white wine, which is to be expected.

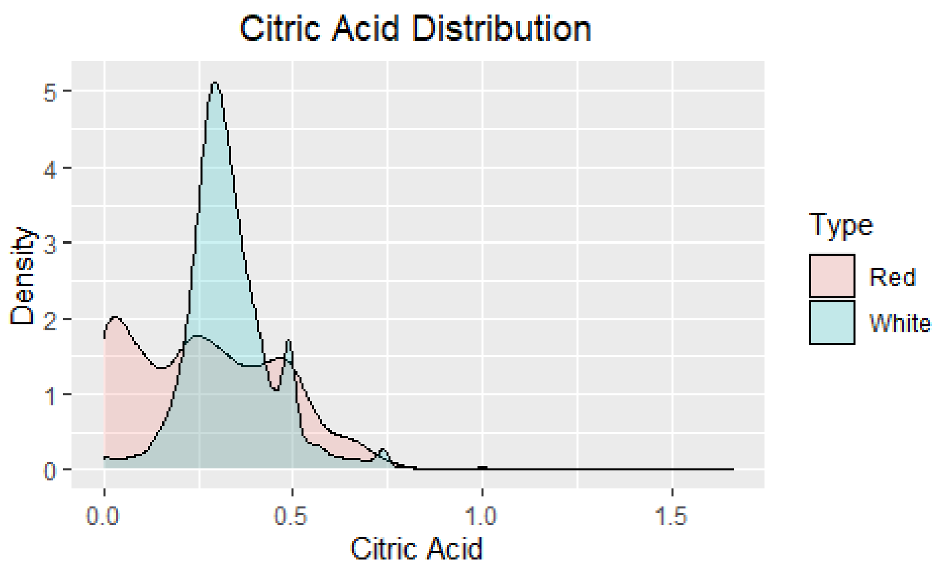
```r
# Create density plot of fixed acidity by wine type
wine_quality %>%
  mutate(Type = type) %>%
  ggplot(aes(fixed.acidity, fill = Type)) +
  geom_density(alpha = 0.2) +
  xlab("Fixed Acidity") +
  ylab("Density") +
  ggtitle("Fixed Acidity Distribution") +
  theme(plot.title = element_text(hjust = 0.5))
```



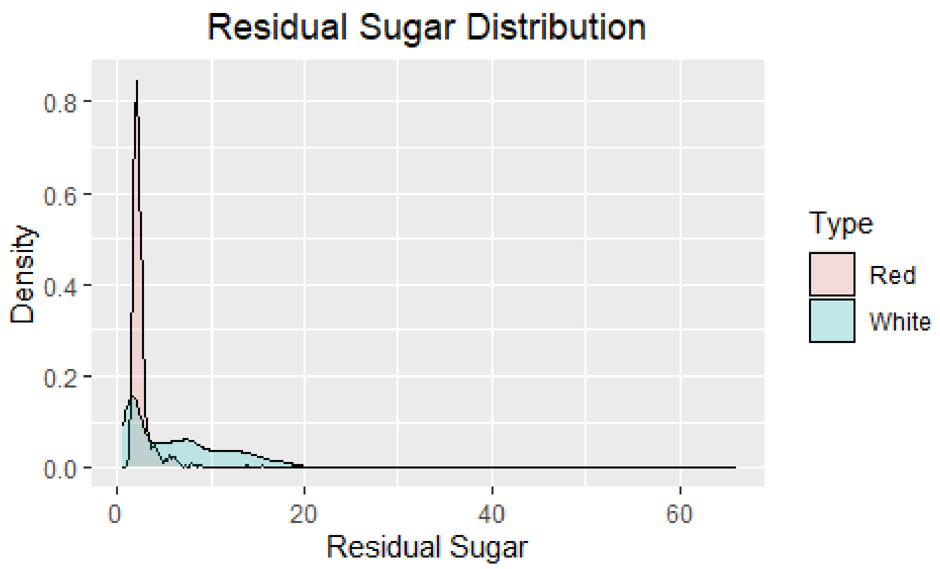
```r
# Create density plot of volatile acidity by wine type
wine_quality %>%
  mutate(Type = type) %>%
  ggplot(aes(volatile.acidity, fill = Type)) +
  geom_density(alpha = 0.2) +
  xlab("Volatile Acidity") +
  ylab("Density") +
  ggtitle("Volatile Acidity Distribution") +
  theme(plot.title = element_text(hjust = 0.5))
```



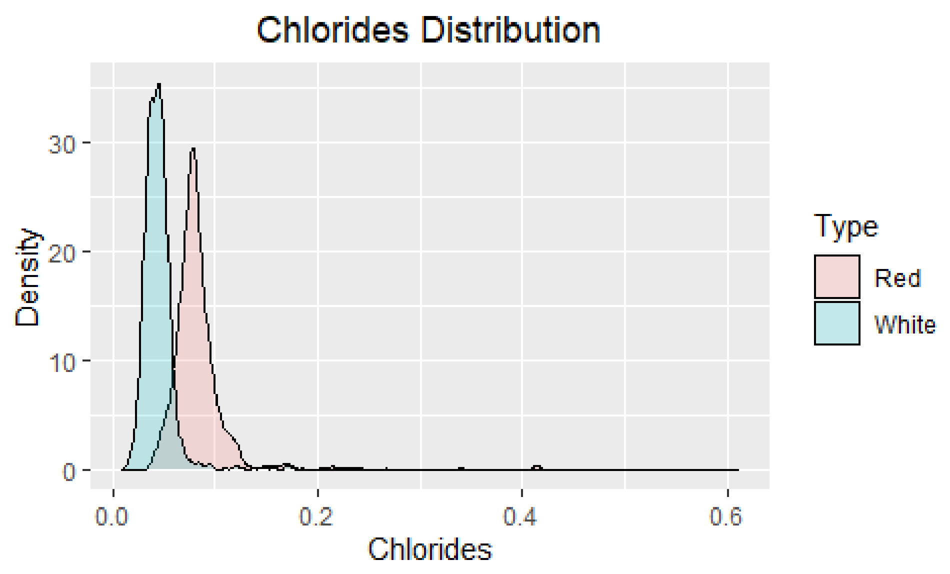
```r
# Create density plot of citric acid by wine type
wine_quality %>%
  mutate(Type = type) %>%
  ggplot(aes(citric.acid, fill = Type)) +
  geom_density(alpha = 0.2) +
  xlab("Citric Acid") +
  ylab("Density") +
  ggtitle("Citric Acid Distribution") +
  theme(plot.title = element_text(hjust = 0.5))
```



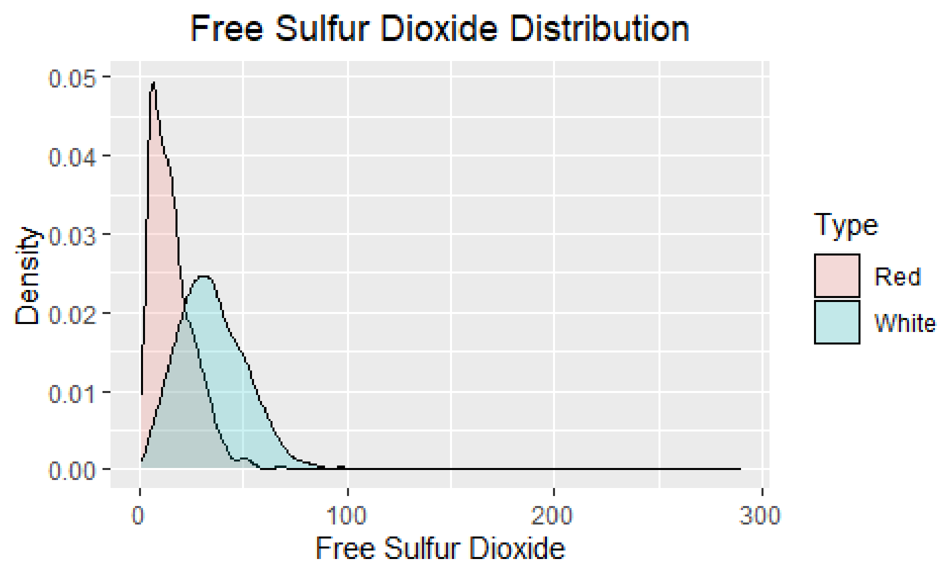
```
# Create density plot of residual sugar by wine type
wine_quality %>%
  mutate(Type = type) %>%
  ggplot(aes(residual.sugar, fill = Type)) +
  geom_density(alpha = 0.2) +
  xlab("Residual Sugar") +
  ylab("Density") +
  ggtitle("Residual Sugar Distribution") +
  theme(plot.title = element_text(hjust = 0.5))
```



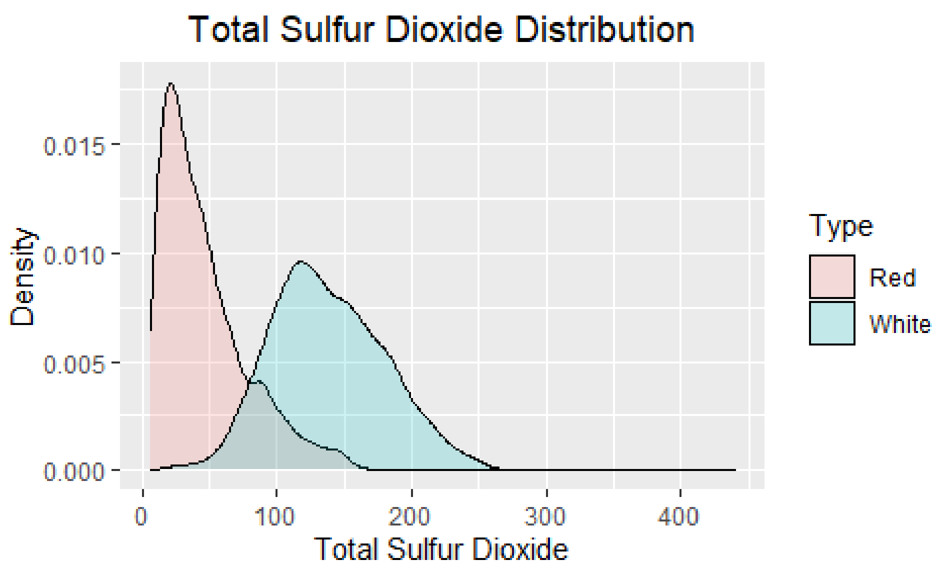
```
# Create density plot of chlorides by wine type
wine_quality %>%
  mutate(Type = type) %>%
  ggplot(aes(chlorides, fill = Type)) +
  geom_density(alpha = 0.2) +
  xlab("Chlorides") +
  ylab("Density") +
  ggtitle("Chlorides Distribution") +
  theme(plot.title = element_text(hjust = 0.5))
```



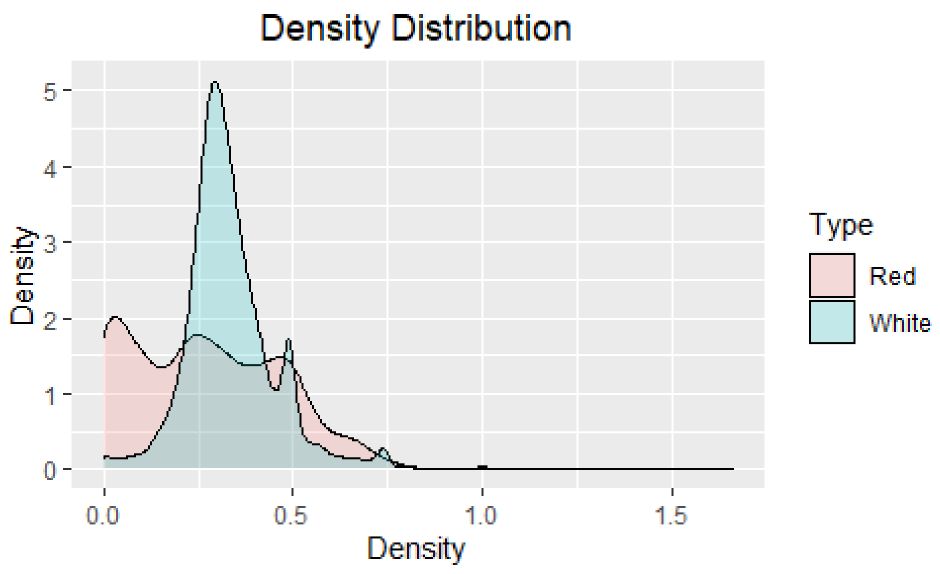
```r
# Create density plot of free sulfur dioxide by wine type
wine_quality %>%
  mutate(Type = type) %>%
  ggplot(aes(free.sulfur.dioxide, fill = Type)) +
  geom_density(alpha = 0.2) +
  xlab("Free Sulfur Dioxide") +
  ylab("Density") +
  ggtitle("Free Sulfur Dioxide Distribution") +
  theme(plot.title = element_text(hjust = 0.5))
```



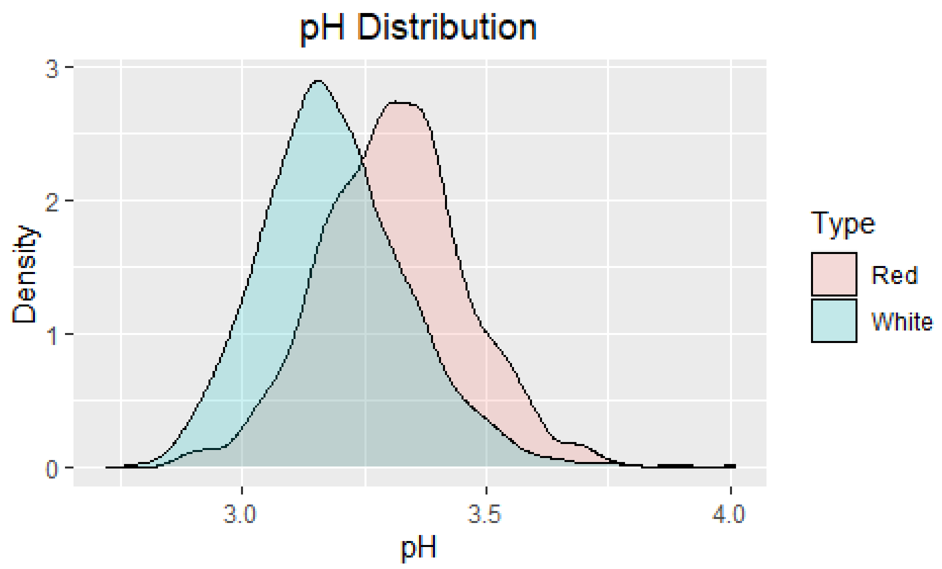
```r
# Create density plot of total sulfur dioxide by wine type
wine_quality %>%
  mutate(Type = type) %>%
  ggplot(aes(total.sulfur.dioxide, fill = Type)) +
  geom_density(alpha = 0.2) +
  xlab("Total Sulfur Dioxide") +
  ylab("Density") +
  ggtitle("Total Sulfur Dioxide Distribution") +
  theme(plot.title = element_text(hjust = 0.5))
```



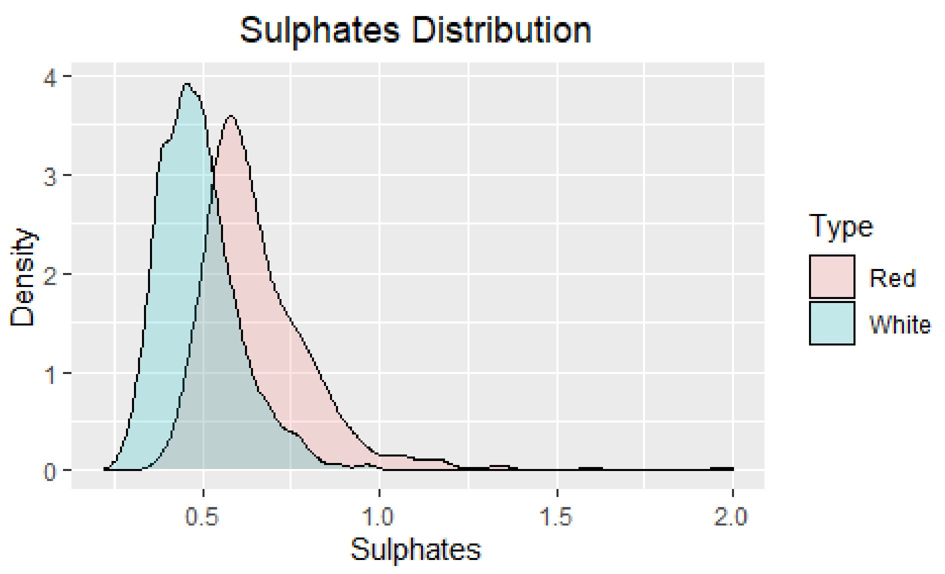
```r
# Create density plot of density by wine type
wine_quality %>%
  mutate(Type = type) %>%
  ggplot(aes(citric.acid, fill = Type)) +
  geom_density(alpha = 0.2) +
  xlab("Density") +
  ylab("Density") +
  ggtitle("Density Distribution") +
  theme(plot.title = element_text(hjust = 0.5))
```



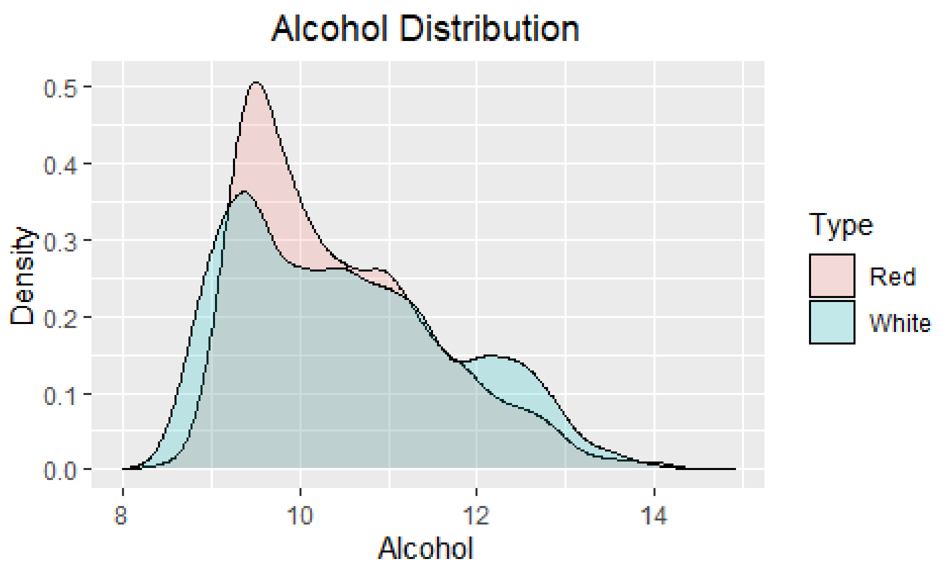
```r
# Create density plot of pH by wine type
wine_quality %>%
  mutate(Type = type) %>%
  ggplot(aes(pH, fill = Type)) +
  geom_density(alpha = 0.2) +
  xlab("pH") +
  ylab("Density") +
  ggtitle("pH Distribution") +
  theme(plot.title = element_text(hjust = 0.5))
```



```r
# Create density plot of sulphates by wine type
wine_quality %>%
  mutate(Type = type) %>%
  ggplot(aes(sulphates, fill = Type)) +
  geom_density(alpha = 0.2) +
  xlab("Sulphates") +
  ylab("Density") +
  ggtitle("Sulphates Distribution") +
  theme(plot.title = element_text(hjust = 0.5))
```



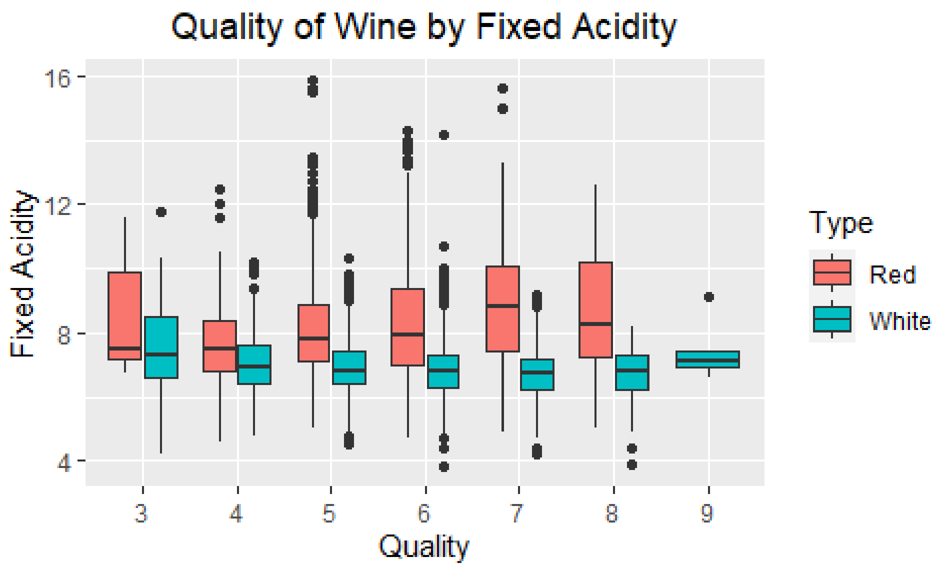
```
# Create density plot of alcohol by wine type
wine_quality %>%
  mutate(Type = type) %>%
  ggplot(aes(alcohol, fill = Type)) +
  geom_density(alpha = 0.2) +
  xlab("Alcohol") +
  ylab("Density") +
  ggtitle("Alcohol Distribution") +
  theme(plot.title = element_text(hjust = 0.5))
```



2.2.2 Fixed Acidity

Fixed acidity is measured in this dataset as the amount of tartaric acid in gm/dm^3. This is the primary acidic component in wine grapes. High levels can result in a sour taste. As can be seen in the boxplot, there are some high outliers, and these tend to have slightly lower quality scores.

```r
# Create boxplot of fixed acidity by wine type
wine_quality %>%
  mutate(Type = factor(type), Quality = factor(quality)) %>%
  ggplot(aes(Quality, fixed.acidity, fill = Type)) +
  geom_boxplot() +
  ylab("Fixed Acidity") +
  ggtitle("Quality of Wine by Fixed Acidity") +
  theme(plot.title = element_text(hjust = 0.5))
```



Overall, the correlation coefficient suggests that there is no correlation between fixed acidity and quality.

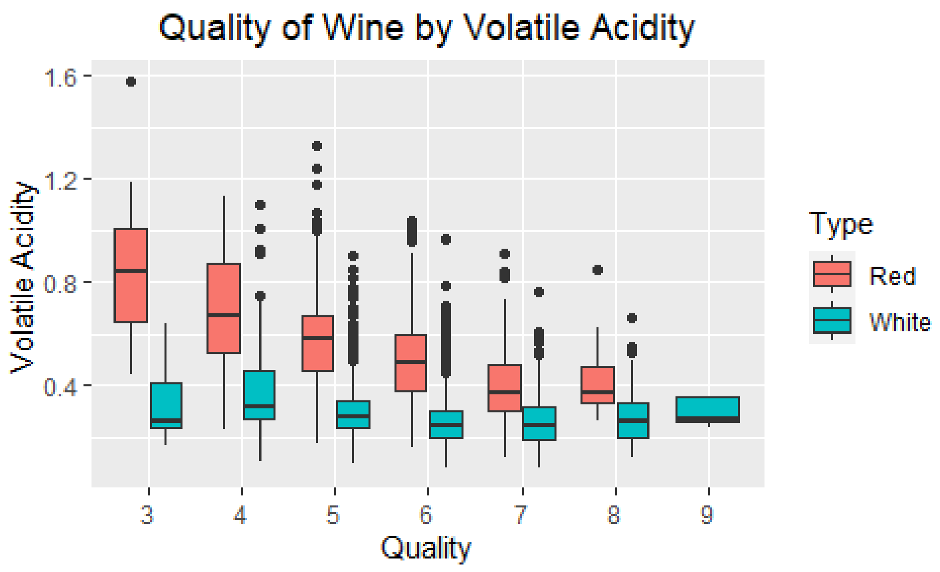
```r
# Calculate correlation of fixed acidity to quality by wine type
cor_fixed <- wine_quality %>%
  group_by(Type = type) %>%
  summarize(Correlation = cor(fixed.acidity, quality))
cor_results <- data.frame(Variable = "Fixed Acidity", Type = cor_fixed$Type,
Correlation = cor_fixed$Correlation)
cor_fixed %>% knitr::kable()
```

Type	Correlation
Red	0.1240516
White	-0.1136628

2.2.3 Volatile Acidity

Volatile acidity is measured in this dataset as the amount of acetic acid in gm/dm^3. High levels can result in an unpleasant, vinegar taste. As can be seen in the boxplot, many white wines measured had low levels. However, lower quality red wines had higher levels.

```
# Create boxplot of volatile acidity by wine type
wine_quality %>%
  mutate(Type = factor(type), Quality = factor(quality)) %>%
  ggplot(aes(Quality, volatile.acidity, fill = Type)) +
  geom_boxplot() +
  ylab("Volatile Acidity") +
  ggtitle("Quality of Wine by Volatile Acidity") +
  theme(plot.title = element_text(hjust = 0.5))
```



Overall, the correlation coefficient suggests that there is some negative correlation between volatile acidity and quality in red wine. The correlation coefficient suggests this is not the case with white wine, as the levels are low.

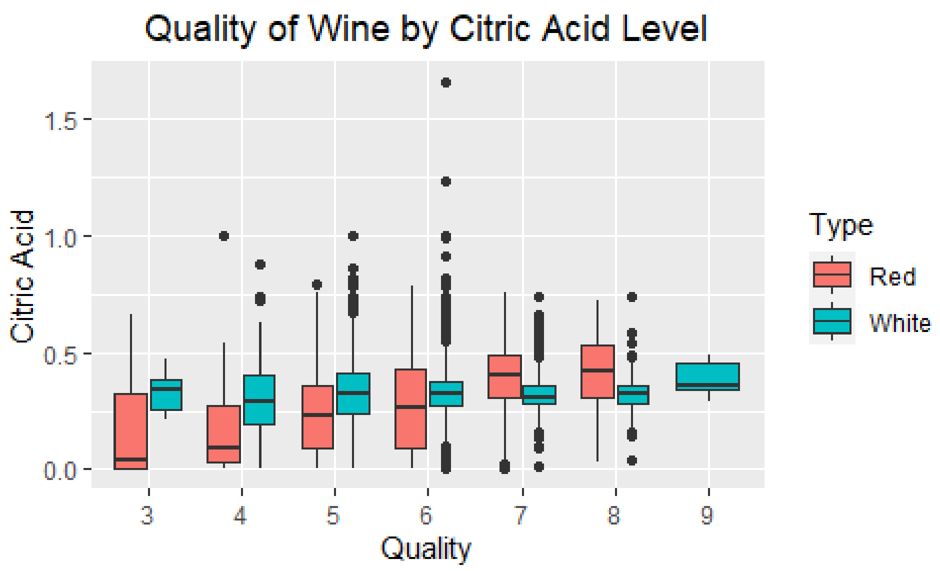
```
# Calculate correlation of volatile acidity to quality by wine type
cor_volat <- wine_quality %>%
  group_by(Type = type) %>%
  summarize(Correlation = cor(volatile.acidity, quality))
cor_results <- bind_rows(cor_results, data_frame(Variable = "Volatile
Acidity", Type = cor_volat$Type, Correlation = cor_volat$Correlation))
cor_volat %>% knitr::kable()
```

Type	Correlation
Red	-0.3905578
White	-0.1947230

2.2.4 Citric Acid

Citric acid is measured in this dataset in gm/dm^3. This is often added to wine to increase acidity and add a fresh flavor. As can be seen in the boxplot, there are some high outliers, and these tend to have higher quality scores. In the case of red wines, the median citric acid level increased with quality.

```r
# Create boxplot of citric acid by wine type
wine_quality %>%
  mutate(Type = factor(type), Quality = factor(quality)) %>%
  ggplot(aes(Quality, citric.acid, fill = Type)) +
  geom_boxplot() +
  ylab("Citric Acid") +
  ggtitle("Quality of Wine by Citric Acid Level") +
  theme(plot.title = element_text(hjust = 0.5))
```



Overall, the correlation coefficient suggests that there is no correlation between citric acid and quality in white wine, and low correlation in red wine.

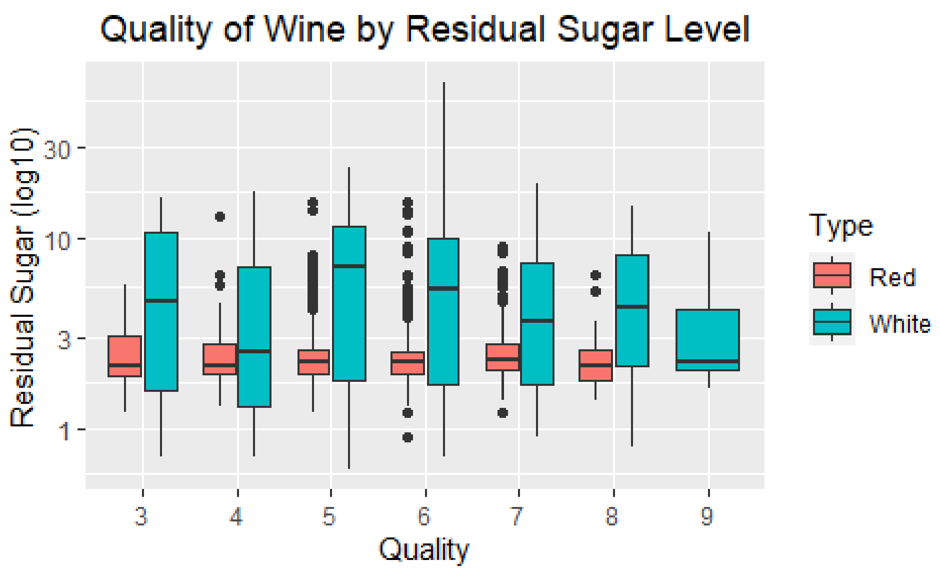
```r
# Calculate correlation of citric acid to quality by wine type
cor_citric <- wine_quality %>%
  group_by(Type = type) %>%
  summarize(Correlation = cor(citric.acid, quality))
cor_results <- bind_rows(cor_results, data_frame(Variable = "Citric Acid",
Type = cor_citric$Type, Correlation = cor_citric$Correlation))
cor_citric %>% knitr::kable()
```

Type	Correlation
Red	0.2263725
White	-0.0092091

2.2.5 Residual Sugar

Residual sugar is the amount of sugar remaining after fermentation, and is measured in gm/dm^3. High levels of residual sugar in wine result in a sweet taste. As there are some extreme outliers in residual sugar, this has been displayed in log scale. Residual sugar overall is lower in the red wine measured.

```r
# Create boxplot of residual sugar by wine type
wine_quality %>%
  mutate(Type = factor(type), Quality = factor(quality)) %>%
  ggplot(aes(Quality, residual.sugar, fill = Type)) +
  geom_boxplot() +
  scale_y_log10() +
  ylab("Residual Sugar (log10)") +
  ggtitle("Quality of Wine by Residual Sugar Level") +
  theme(plot.title = element_text(hjust = 0.5))
```



The correlation coefficient suggests there is no correlation between residual sugar and quality of red or white wine.

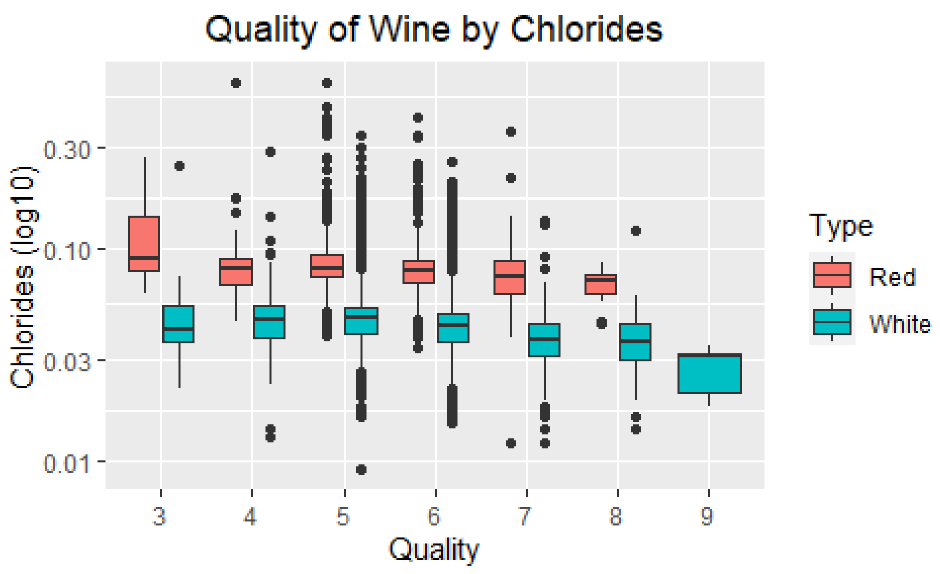
```r
# Calculate correlation of residual sugar to quality by wine type
cor_resid <- wine_quality %>%
  group_by(Type = type) %>%
  summarize(Correlation = cor(residual.sugar, quality))
cor_results <- bind_rows(cor_results, data_frame(Variable = "Residual Sugar",
Type = cor_resid$Type, Correlation = cor_resid$Correlation))
cor_resid %>% knitr::kable()
```

Type	Correlation
Red	0.0137316
White	-0.0975768

2.2.6 Chlorides

Chlorides are measured in this dataset as the amount of sodium chloride in gm/dm^3. Sodium chloride can be used in the winemaking “pre-conditioning” process to improve fermentation. High levels can also be due to lack of rainfall impacting soil, and can result in a salty taste. As there are some extreme outliers, this has been displayed in log scale.

```r
# Create boxplot of chlorides by wine type
wine_quality %>%
  mutate(Type = factor(type), Quality = factor(quality)) %>%
  ggplot(aes(Quality, chlorides, fill = Type)) +
  geom_boxplot() +
  scale_y_log10() +
  ylab("Chlorides (log10)") +
  ggtitle("Quality of Wine by Chlorides") +
  theme(plot.title = element_text(hjust = 0.5))
```



The correlation coefficient suggests there is low correlation between chlorides and quality of red and white wine.

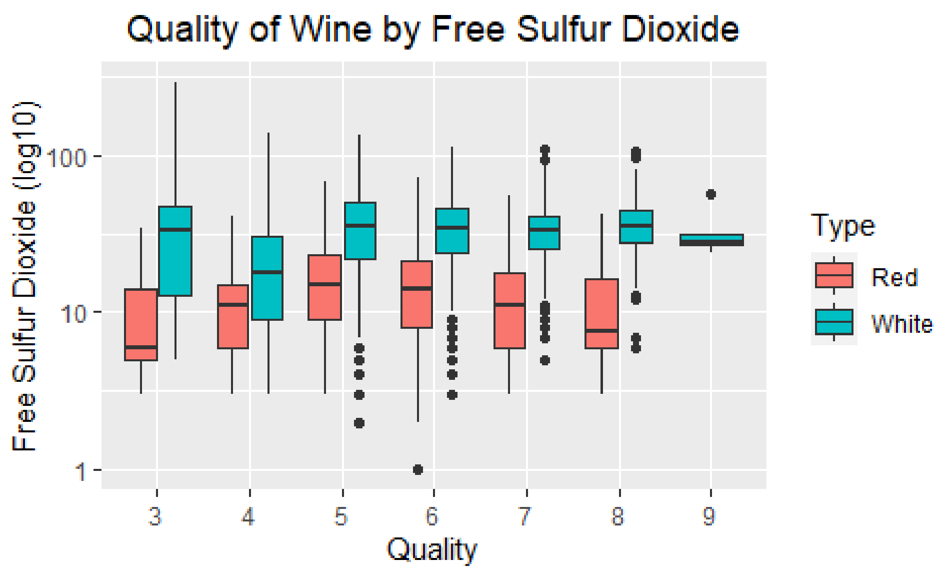
```r
# Calculate correlation of chlorides to quality by wine type
cor_chlor <- wine_quality %>%
  group_by(Type = type) %>%
  summarize(Correlation = cor(chlorides, quality))
cor_results <- bind_rows(cor_results, data_frame(Variable = "Chlorides", Type
= cor_chlor$Type, Correlation = cor_chlor$Correlation))
cor_chlor %>% knitr::kable()
```

Type	Correlation
Red	-0.1289066
White	-0.2099344

2.2.7 Free Sulfur Dioxide

Free sulfur dioxide is measured in this dataset as the amount of SO2 not bound to other molecules in mg/dm^3. Free sulfur dioxide is often used as an additive to kill unwanted bacteria and prevent oxidation. As there are some extreme outliers, this has been displayed in log scale. There does not appear to be a relationship to quality in wines measured.

```r
# Create boxplot of free sulfur dioxide by wine type
wine_quality %>%
  mutate(Type = factor(type), Quality = factor(quality)) %>%
  ggplot(aes(Quality, free.sulfur.dioxide, fill = Type)) +
  geom_boxplot() +
  scale_y_log10() +
  ylab("Free Sulfur Dioxide (log10)") +
  ggtitle("Quality of Wine by Free Sulfur Dioxide") +
  theme(plot.title = element_text(hjust = 0.5))
```



The correlation coefficient suggests there is no correlation between free sulfur dioxide and quality of red and white wine.

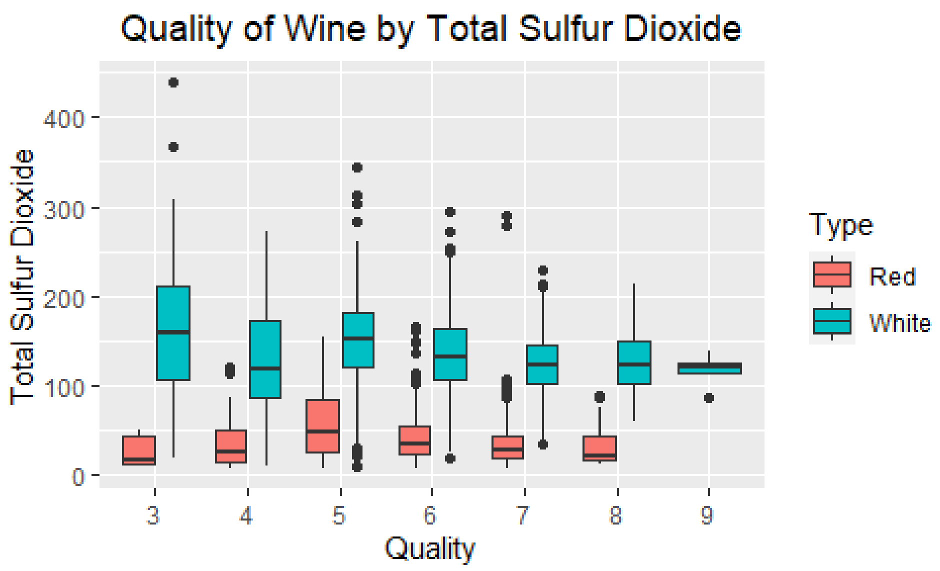
```r
# Calculate correlation of free sulfur dioxide to quality by wine type
cor_fsulf <- wine_quality %>%
  group_by(Type = type) %>%
  summarize(Correlation = cor(free.sulfur.dioxide, quality))
cor_results <- bind_rows(cor_results, data_frame(Variable = "Free Sulfur
Dioxide", Type = cor_fsulf$Type, Correlation = cor_fsulf$Correlation))
cor_fsulf %>% knitr::kable()
```

Type	Correlation
Red	-0.0506561
White	0.0081581

2.2.8 Total Sulfur Dioxide

Total sulfur dioxide is measured in this dataset as the total amount of SO2 in mg/dm^3. Sulfur dioxide is often used as an additive to kill unwanted bacteria and prevent oxidation. High levels can result in a change in taste and color. Outliers are less extreme in total sulfur dioxide levels than free sulfur dioxide levels and does not require log scale.

```
# Create boxplot of total sulfur dioxide by wine type
wine_quality %>%
  mutate(Type = factor(type), Quality = factor(quality)) %>%
  ggplot(aes(Quality, total.sulfur.dioxide, fill = Type)) +
  geom_boxplot() +
  ylab("Total Sulfur Dioxide") +
  ggtitle("Quality of Wine by Total Sulfur Dioxide") +
  theme(plot.title = element_text(hjust = 0.5))
```



The correlation coefficient suggests there is very low correlation between total sulfur dioxide and quality of red and white wine.

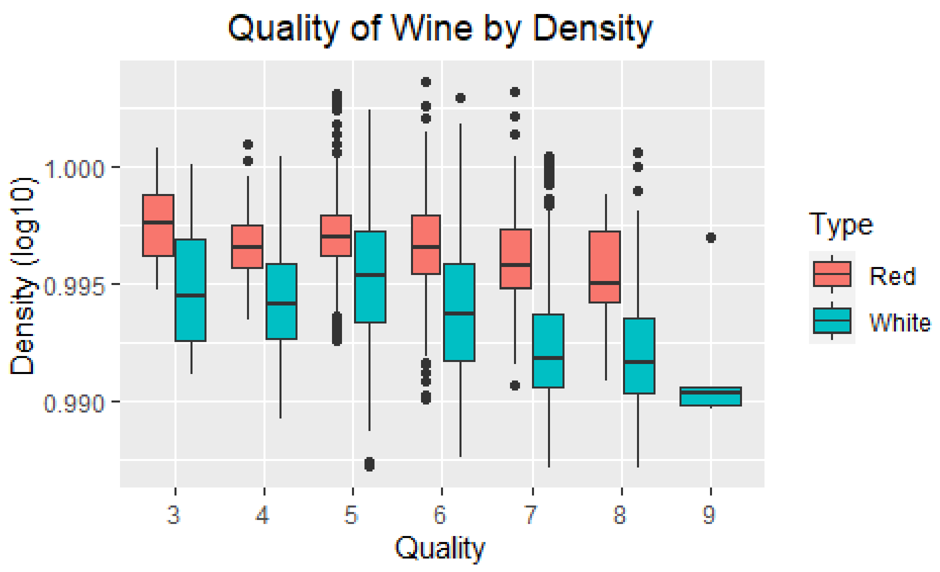
```
# Calculate correlation of total sulfur dioxide to quality by wine type
cor_tsulf <- wine_quality %>%
  group_by(Type = type) %>%
  summarize(Correlation = cor(total.sulfur.dioxide, quality))
cor_results <- bind_rows(cor_results, data_frame(Variable = "Total Sulfur
Dioxide", Type = cor_tsulf$Type, Correlation = cor_tsulf$Correlation))
cor_tsulf %>% knitr::kable()
```

Type	Correlation
Red	-0.1851003
White	-0.1747372

2.2.9 Density

Density is measured as the total mass per volume of alcohol, sugar, and water in g/cm^3. The more alcohol a wine contains should decrease the overall density. As there are some extreme outliers, this has been displayed in log scale and two outliers greater than 1.01 have been excluded. The highest quality wines had a lower density, likely driven by ABV.

```r
# Create boxplot of density by wine type
wine_quality %>%
  filter(density < 1.01) %>%
  mutate(Type = factor(type), Quality = factor(quality)) %>%
  ggplot(aes(Quality, density, fill = Type)) +
  geom_boxplot() +
  scale_y_log10() +
  ylab("Density (log10)") +
  ggtitle("Quality of Wine by Density") +
  theme(plot.title = element_text(hjust = 0.5))
```



The correlation coefficient suggests there is low correlation between density and quality of red and white wine. The correlation is higher in white wine.

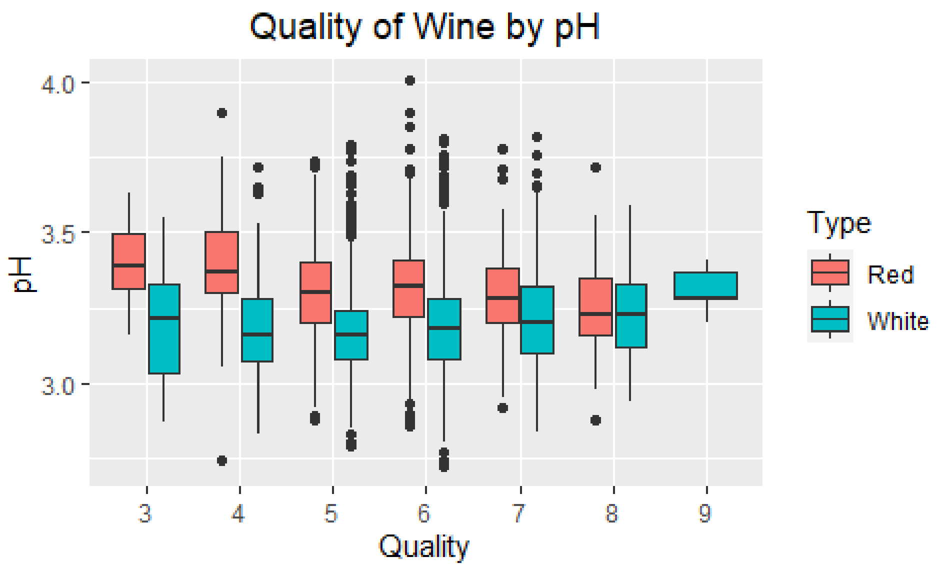
```r
# Calculate correlation of density to quality by wine type
cor_densi <- wine_quality %>%
  group_by(Type = type) %>%
  summarize(Correlation = cor(density, quality))
cor_results <- bind_rows(cor_results, data_frame(Variable = "Density", Type =
cor_densi$Type, Correlation = cor_densi$Correlation))
cor_densi %>% knitr::kable()
```

Type	Correlation
Red	-0.1749192
White	-0.3071233

2.2.10 pH

pH is a measurement of acidity, with basic wines having the lowest pH and acidic wines having the highest. Low pH wines can taste crisp, and high pH wines can taste sour and are more susceptible to bacterial growth. However, there does not appear to be a strong relationship based on the wines measured in the dataset.

```r
# Create boxplot of pH by wine type
wine_quality %>%
  mutate(Type = factor(type), Quality = factor(quality)) %>%
  ggplot(aes(Quality, pH, fill = Type)) +
  geom_boxplot() +
  ylab("pH") +
  ggtitle("Quality of Wine by pH") +
  theme(plot.title = element_text(hjust = 0.5))
```



The correlation coefficient suggests there is no correlation between density and quality of red and white wine.

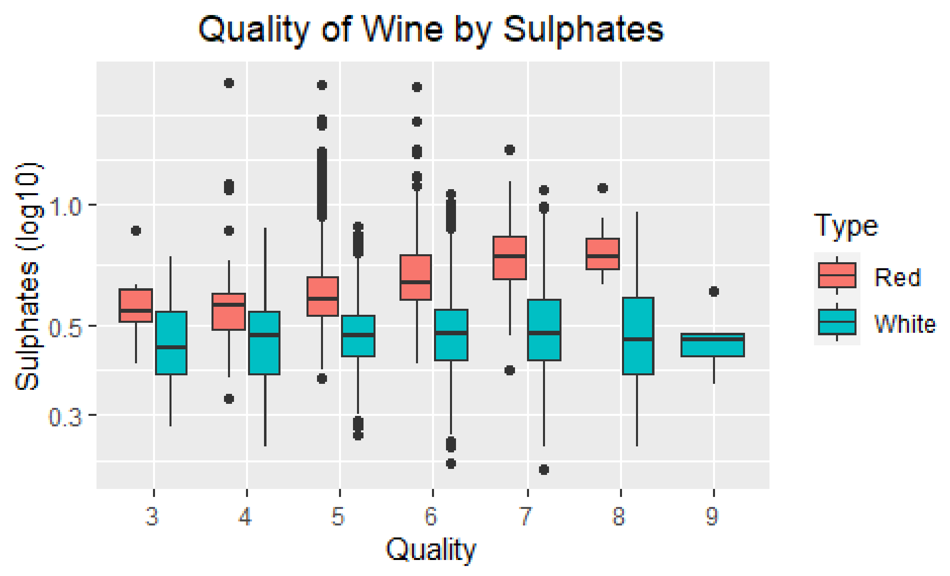
```r
# Calculate correlation of pH to quality by wine type
cor_ph <- wine_quality %>%
  group_by(Type = type) %>%
  summarize(Correlation = cor(pH, quality))
cor_results <- bind_rows(cor_results, data_frame(Variable = "pH", Type =
cor_ph$Type, Correlation = cor_ph$Correlation))
cor_ph %>% knitr::kable()
```

Type	Correlation
Red	-0.0577314
White	0.0994272

2.2.11 Sulphates

Sulphates are measured in this dataset as the total amount of potassium sulphate in g/dm^3. Sulphates are often used as an additive which acts as a preservative. Extremely high levels can result in a sulfur taste. As there are some extreme outliers, this has been displayed in log scale. Higher quality red wines had higher median levels of sulphates.

```r
# Create boxplot of sulphates by wine type
wine_quality %>%
  mutate(Type = factor(type), Quality = factor(quality)) %>%
  ggplot(aes(Quality, sulphates, fill = Type)) +
  geom_boxplot() +
  scale_y_log10() +
  ylab("Sulphates (log10)") +
  ggtitle("Quality of Wine by Sulphates") +
  theme(plot.title = element_text(hjust = 0.5))
```



The correlation coefficient suggests there is no correlation between sulphates and quality of white wine, and low correlation in red wine.

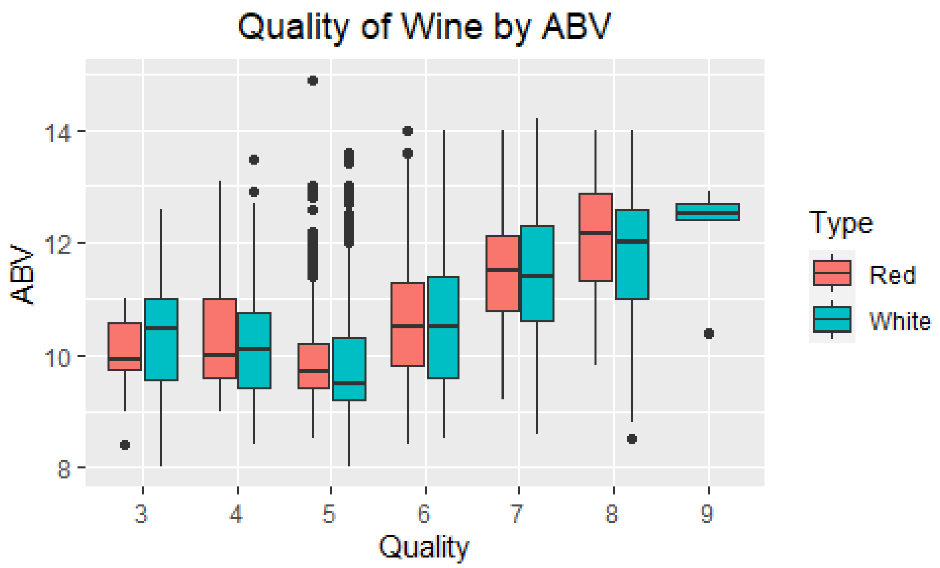
```r
# Calculate correlation of sulphates to quality by wine type
cor_sulph <- wine_quality %>%
  group_by(Type = type) %>%
  summarize(Correlation = cor(sulphates, quality))
cor_results <- bind_rows(cor_results, data_frame(Variable = "Sulphates", Type
= cor_sulph$Type, Correlation = cor_sulph$Correlation))
cor_sulph %>% knitr::kable()
```

Type	Correlation
Red	0.2513971
White	0.0536779

2.2.12 Alcohol

Alcohol in this dataset is a measurement of the percent alcohol content by volume (ABV) as (*% vol*). A higher ABV wine typically results in a bolder and richer taste. As can be seen in the boxplot, wines with a higher ABV were rated as higher quality of those measured.

```
# Create boxplot of alcohol by wine type
wine_quality %>%
  mutate(Type = factor(type), Quality = factor(quality)) %>%
  ggplot(aes(Quality, alcohol, fill = Type)) +
  geom_boxplot() +
  ylab("ABV") +
  ggtitle("Quality of Wine by ABV") +
  theme(plot.title = element_text(hjust = 0.5))
```



The correlation coefficient suggests there is moderate correlation between ABV and quality of white and red wine.

```
# Calculate correlation of alcohol to quality by wine type
cor_abv <- wine_quality %>%
  group_by(Type = type) %>%
  summarize(Correlation = cor(alcohol, quality))
cor_results <- bind_rows(cor_results, data_frame(Variable = "Alcohol", Type =
cor_abv$Type, Correlation = cor_abv$Correlation))
cor_abv %>% knitr::kable()
```

Type	Correlation
Red	0.4761663
White	0.4355747

2.3 Data Modeling

According to the correlation coefficients calculated on the dataset, the only variables that have a relationship to quality are alcohol content, density, volatile acidity, and chlorides. The correlation coefficient is an informative measure of how two variables move together. The number is positive if they move in parallel, and negative if they move in opposite directions. Perfect correlation is equal to 1, and moderate is 0.5.

```r
# Create summary of red and white wine correlation results
cor_summary <- cor_results %>%
  filter(Type=="Red") %>%
  mutate(Red = Correlation) %>%
  select(Variable, Red)
cor_white <- cor_results %>%
  filter(Type=="White") %>%
  mutate(White = Correlation) %>%
  select(Variable, White)
cor_summary <- data.frame(inner_join(cor_summary, cor_white))

# Calculate total correlation for variables
cor_fixed2 <- wine_quality %>%
    summarize(Correlation = cor(fixed.acidity, quality))
cor_results2 <- data.frame(Variable = "Fixed Acidity", Total =
cor_fixed2$Correlation)

cor_volat2 <- wine_quality %>%
  summarize(Correlation = cor(volatile.acidity, quality))
cor_results2 <- bind_rows(cor_results2, data_frame(Variable = "Volatile
Acidity", Total = cor_volat2$Correlation))

cor_citric2 <- wine_quality %>%
  summarize(Correlation = cor(citric.acid, quality))
cor_results2 <- bind_rows(cor_results2, data_frame(Variable = "Citric Acid",
Total = cor_citric2$Correlation))

cor_resid2 <- wine_quality %>%
  summarize(Correlation = cor(residual.sugar, quality))
cor_results2 <- bind_rows(cor_results2, data_frame(Variable = "Residual
Sugar", Total = cor_resid2$Correlation))

cor_chlor2 <- wine_quality %>%
  summarize(Correlation = cor(chlorides, quality))
cor_results2 <- bind_rows(cor_results2, data_frame(Variable = "Chlorides",
Total = cor_chlor2$Correlation))

cor_fsulf2 <- wine_quality %>%
  summarize(Correlation = cor(free.sulfur.dioxide, quality))
cor_results2 <- bind_rows(cor_results2, data_frame(Variable = "Free Sulfur
Dioxide", Total = cor_fsulf2$Correlation))
```

```r
cor_tsulf2 <- wine_quality %>%
  summarize(Correlation = cor(total.sulfur.dioxide, quality))
cor_results2 <- bind_rows(cor_results2, data_frame(Variable = "Total Sulfur
Dioxide", Total = cor_tsulf2$Correlation))

cor_densi2 <- wine_quality %>%
  summarize(Correlation = cor(density, quality))
cor_results2 <- bind_rows(cor_results2, data_frame(Variable = "Density",
Total = cor_densi2$Correlation))

cor_ph2 <- wine_quality %>%
  summarize(Correlation = cor(pH, quality))
cor_results2 <- bind_rows(cor_results2, data_frame(Variable = "pH", Total =
cor_ph2$Correlation))

cor_sulph2 <- wine_quality %>%
  summarize(Correlation = cor(sulphates, quality))
cor_results2 <- bind_rows(cor_results2, data_frame(Variable = "Sulphates",
Total = cor_sulph2$Correlation))

cor_abv2 <- wine_quality %>%
  summarize(Correlation = cor(alcohol, quality))
cor_results2 <- bind_rows(cor_results2, data_frame(Variable = "Alcohol",
Total = cor_abv2$Correlation))

# Add total correlation to summary of red and white wine correlation results
cor_summary <- data.frame(inner_join(cor_summary, cor_results2))
cor_summary %>%
  arrange(desc(abs(Total))) %>%
  knitr::kable()
```

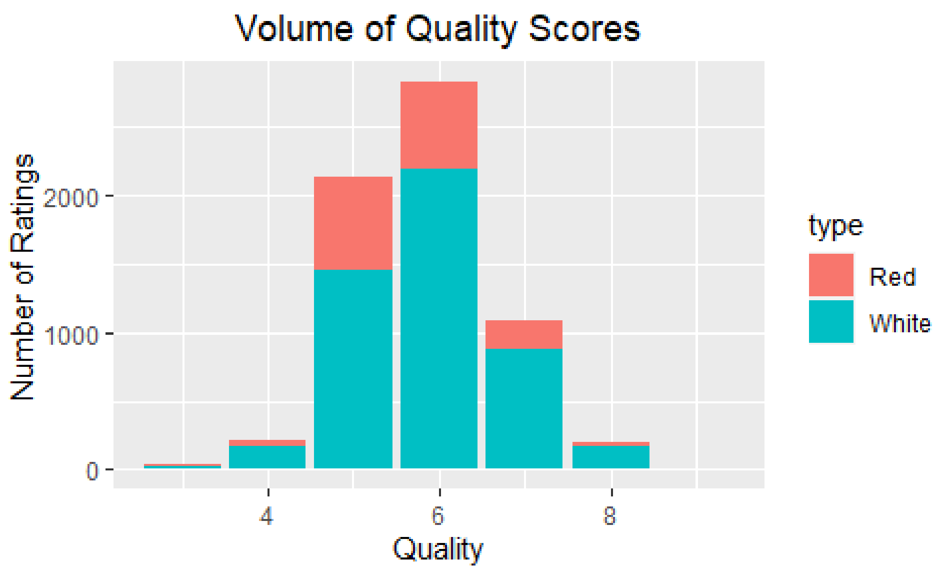
Variable	Red	White	Total
Alcohol	0.4761663	0.4355747	0.4443185
Density	-0.1749192	-0.3071233	-0.3058579
Volatile Acidity	-0.3905578	-0.1947230	-0.2656995
Chlorides	-0.1289066	-0.2099344	-0.2006655
Citric Acid	0.2263725	-0.0092091	0.0855317
Fixed Acidity	0.1240516	-0.1136628	-0.0767432
Free Sulfur Dioxide	-0.0506561	0.0081581	0.0554631
Total Sulfur Dioxide	-0.1851003	-0.1747372	-0.0413855
Sulphates	0.2513971	0.0536779	0.0384854
Residual Sugar	0.0137316	-0.0975768	-0.0369805
pH	-0.0577314	0.0994272	0.0195057

Correlation is not always a good indication of a relationship between variables, as random variables are found to have correlation at times when there is no logical relationship. In this case, there is a logical relationship between these variables and quality. These will be the key variables that drive the various models to be evaluated in this section to determine the optimal model for predicting wine quality based on physiochemical analysis.

In each of the models, the training set, test set and validation set are separated into a file of predictors (x) and outcomes (y). The predictors are the four variables identified in the exploration and correlation analysis as having relationships to wine quality: alcohol content, density, volatile acidity, and chlorides. The outcomes to be predicted are defined as lower quality and higher quality wine.

Lower quality wines are defined as a quality rating of 5 or less, which accounts for a third of the dataset. These will require additional sensory testing. Higher quality wines are defined as a quality rating of 6 or more. Ideally, lower quality wines would be defined as a lower quality score accounting for closer to 20% of the dataset. However, given that so few wines are accounted for in ratings of 4 or less, a higher cutoff has been used.

```r
# Create bar chart by quality
wine_quality %>%
  group_by(quality, type) %>%
  summarize(count = n()) %>%
  ggplot(aes(quality, count, fill = type)) +
  geom_bar(stat="identity", show.legend = TRUE, position = "stack") +
  xlab("Quality") +
  ylab("Number of Ratings") +
  ggtitle("Volume of Quality Scores") +
  theme(plot.title = element_text(hjust = 0.5))
```



2.3.1 Logistic Regression

The first model is a simple **binary logistic regression** model. The dataset is fit into a linear regression model, which then can be acted upon by a logistic function predicting the target categorical variable. In this case, this will be if the wine is of lower or higher quality.

The logistic regression model uses the **maximum likelihood estimate (MLE)**. This is a method of estimating the parameters of a statistical model by finding the parameters that maximize the likelihood of accurate prediction. The accuracy of this model is **74.97%.** Sensitivity (true positives) and precision (proportion of predicted positives that were predicted correctly) are higher, but specificity (true negatives) is lower.

```r
# Create predictor train, test, and validation datasets
train_x <- train_set %>%
  select(alcohol, density, volatile.acidity, chlorides)
test_x <- test_set %>%
  select(alcohol, density, volatile.acidity, chlorides)
validation_x <- validation %>%
  select(alcohol, density, volatile.acidity, chlorides)

# Create outcome train, test, and validation datasets
train_y <- train_set %>%
  mutate(Quality = factor(ifelse(quality > 5, "H", "L"))) %>%
  select(Quality)
test_y <- test_set %>%
  mutate(Quality = factor(ifelse(quality > 5, "H", "L"))) %>%
  select(Quality)
validation_y <- validation %>%
  mutate(Quality = factor(ifelse(quality > 5, "H", "L"))) %>%
  select(Quality)

# Fit logistic regression model on the training dataset
train_glm <- train(train_x, train_y$Quality, method = "glm")

# Make predictions on the test dataset
glm_preds <- predict(train_glm, test_x)

# Calculate accuracy, sensitivity, specificity, and precision
accuracy <- mean(glm_preds == test_y$Quality)
sensitivity <- sensitivity(glm_preds, test_y$Quality)
specificity <- specificity(glm_preds, test_y$Quality)
precision <- precision(glm_preds, test_y$Quality)
results <- data_frame(Model = "Logistic Regression", Accuracy = accuracy,
Sensitivity = sensitivity, Specificity = specificity, Precision = precision)
results %>% knitr::kable()
```

Model	Accuracy	Sensitivity	Specificity	Precision
Logistic Regression	0.7497436	0.8427877	0.5893855	0.7796102

2.3.2 LDA

Linear discriminant analysis (LDA) is similar to logistic regression. LDA assumes all predictors share the same standard deviations and correlations. Hence, the boundary will be a line. Accuracy improves slightly with this model to **75.38%**.

```r
# Fit LDA model on the training dataset
train_lda <- train(train_x, train_y$Quality, method = "lda")

# Make predictions on the test dataset
lda_preds <- predict(train_lda, test_x)

# Calculate accuracy, sensitivity, specificity, and precision
accuracy <- mean(lda_preds == test_y$Quality)
sensitivity <- sensitivity(lda_preds, test_y$Quality)
specificity <- specificity(lda_preds, test_y$Quality)
precision <- precision(lda_preds, test_y$Quality)
results <- bind_rows(results,data_frame(Model = "LDA", Accuracy = accuracy,
Sensitivity = sensitivity, Specificity = specificity, Precision = precision))
results %>% knitr::kable()
```

Model	Accuracy	Sensitivity	Specificity	Precision
Logistic Regression	0.7497436	0.8427877	0.5893855	0.7796102
LDA	0.7538462	0.8508914	0.5865922	0.7800892

2.3.3 QDA

Quadratic discriminant analysis (QDA) is a version of Naive Bayes in which distributions are assumed to be multivariate normal. QDA is ideal when there are a minimal number of predictors, but results in overfitting if there are too many. Accuracy declines in this model to **71.79%**. Sensitivity is higher than the logistic regression and LDA model, however, specificity is extremely low.

```r
# Fit QDA model on the training dataset
train_qda <- train(train_x, train_y$Quality, method = "qda")

# Make predictions on the test dataset
qda_preds <- predict(train_qda, test_x)

# Calculate accuracy, sensitivity, specificity, and precision
accuracy <- mean(qda_preds == test_y$Quality)
sensitivity <- sensitivity(qda_preds, test_y$Quality)
specificity <- specificity(qda_preds, test_y$Quality)
precision <- precision(qda_preds, test_y$Quality)
results <- bind_rows(results,data_frame(Model = "QDA", Accuracy = accuracy,
Sensitivity = sensitivity, Specificity = specificity, Precision = precision))
results %>% knitr::kable()
```

Model	Accuracy	Sensitivity	Specificity	Precision
Logistic Regression	0.7497436	0.8427877	0.5893855	0.7796102
LDA	0.7538462	0.8508914	0.5865922	0.7800892
QDA	0.7179487	0.9027553	0.3994413	0.7215026

2.3.4 Loess

Local weighted regression (loess) permits the consideration of larger window sizes. The result is typically more optimal as larger sample sizes are used to estimate local parameters. Overall accuracy improves with local weighted regression to **75.89%**.

```
# Fit loess model on the training dataset
train_gam <- train(train_x, train_y$Quality, method = "gamLoess")

# Make predictions on the test dataset
gam_preds <- predict(train_gam, test_x)

# Calculate accuracy, sensitivity, specificity, and precision
accuracy <- mean(gam_preds == test_y$Quality)
sensitivity <- sensitivity(gam_preds, test_y$Quality)
specificity <- specificity(gam_preds, test_y$Quality)
precision <- precision(gam_preds, test_y$Quality)
results <- bind_rows(results,data_frame(Model = "Loess", Accuracy = accuracy,
Sensitivity = sensitivity, Specificity = specificity, Precision = precision))
results %>% knitr::kable()
```

Model	Accuracy	Sensitivity	Specificity	Precision
Logistic Regression	0.7497436	0.8427877	0.5893855	0.7796102
LDA	0.7538462	0.8508914	0.5865922	0.7800892
QDA	0.7179487	0.9027553	0.3994413	0.7215026
Loess	0.7589744	0.8363047	0.6256983	0.7938462

2.3.5 KNN

K-nearest neighbors (kNN) estimates the conditional probabilities in a similar way to bin smoothing. However, kNN is easier to adapt to multiple dimensions. kNN evaluates data points for the k nearest points, and averages the results to determine a prediction. The set of points used to compute the average are known as the neighborhood.

The first step is to determine the value of **k** that is used to tune the model. Typically, larger k values will result in smoother estimates, while smaller k values will result in more flexible estimates. kNN accuracy is **74.87**, similar to logistic regression.

```
# Generate sequence of tuning options
set.seed(1, sample.kind = "Rounding")
tuning <- data.frame(k = seq(3, 21, 2))
train_knn <- train(train_x, train_y$Quality, method = "knn", tuneGrid =
tuning)
tuning <- train_knn$bestTune

# Fit KNN model on the training dataset
train_knn <- train(train_x, train_y$Quality, method = "knn", tuneGrid =
tuning)

# Make predictions on the test dataset
knn_preds <- predict(train_knn, test_x)

# Calculate accuracy, sensitivity, specificity, and precision
accuracy <- mean(knn_preds == test_y$Quality)
sensitivity <- sensitivity(knn_preds, test_y$Quality)
specificity <- specificity(knn_preds, test_y$Quality)
precision <- precision(knn_preds, test_y$Quality)
results <- bind_rows(results,data_frame(Model = "KNN", Accuracy = accuracy,
Sensitivity = sensitivity, Specificity = specificity, Precision = precision))
results %>% knitr::kable()
```

Model	Accuracy	Sensitivity	Specificity	Precision
Logistic Regression	0.7497436	0.8427877	0.5893855	0.7796102
LDA	0.7538462	0.8508914	0.5865922	0.7800892
QDA	0.7179487	0.9027553	0.3994413	0.7215026
Loess	0.7589744	0.8363047	0.6256983	0.7938462
KNN	0.7487179	0.8282010	0.6117318	0.7861538

2.3.6 Random Forest

Random forest improves prediction and reduces instability by averaging multiple decision trees. Random forest generates many predictors, each using regression or classification trees, and forms a prediction based on the average prediction of all trees.

The first step in the process is to determine the value of **mtry** that is used to consider a subset of predictors. This reduces correlation between trees increasing accuracy. Random forest results with an accuracy of **77.23%**, which is above the prior models.

```
# Generate sequence of tuning options
set.seed(1, sample.kind = "Rounding")
tuning <- data.frame(mtry = seq(3, 9, 2))
train_rf <- train(train_x, train_y$Quality, method = "rf", nodesize = 1,
tuneGrid = tuning)
tuning <- train_rf$bestTune

# Fit random forest model on the training dataset
train_rf <- train(train_x, train_y$Quality, method = "rf", nodesize = 1,
tuneGrid = tuning)

# Make predictions on the test dataset
```

```r
rf_preds <- predict(train_rf, test_x)

# Calculate accuracy, sensitivity, specificity, and precision
accuracy <- mean(rf_preds == test_y$Quality)
sensitivity <- sensitivity(rf_preds, test_y$Quality)
specificity <- specificity(rf_preds, test_y$Quality)
precision <- precision(rf_preds, test_y$Quality)
results <- bind_rows(results,data_frame(Model = "Random Forest", Accuracy =
accuracy, Sensitivity = sensitivity, Specificity = specificity, Precision =
precision))
results %>% knitr::kable()
```

Model	Accuracy	Sensitivity	Specificity	Precision
Logistic Regression	0.7497436	0.8427877	0.5893855	0.7796102
LDA	0.7538462	0.8508914	0.5865922	0.7800892
QDA	0.7179487	0.9027553	0.3994413	0.7215026
Loess	0.7589744	0.8363047	0.6256983	0.7938462
KNN	0.7487179	0.8282010	0.6117318	0.7861538
Random Forest	0.7723077	0.8282010	0.6759777	0.8149920

The variable importance validates that the most important variables in the model are alcohol and density. Volatile acidity is of lower importance, and chlorides appear as no importance.

```r
# Generate variable importance ranking
imp <- varImp(train_rf)
imp

## rf variable importance
##
##                   Overall
## density            100.00
## alcohol             90.40
## volatile.acidity    38.02
## chlorides            0.00
```

2.3.7 Ensemble

Ensembles attempt to further improve predictions by combining multiple models into a single model. In this model, an ensemble is created using the predictions from the six prior models: logistic regression, LDA, QDA, loess, k-nearest neighbors, and random forest. Accuracy of the ensemble model is **76.20%**, which is just below the random forest model.

```r
# Fit ensemble model on the predictions
ensemble <- cbind(glm = glm_preds == "H", lda = lda_preds == "H", qda =
qda_preds == "H", loess = gam_preds == "H", rf = rf_preds == "H", knn =
knn_preds == "H")
```

```
# Make predictions on the ensemble
ensemble_preds <- factor(ifelse(rowMeans(ensemble) > 0.5, "H", "L"))

# Calculate accuracy, sensitivity, specificity, and precision
accuracy <- mean(ensemble_preds == test_y$Quality)
sensitivity <- sensitivity(ensemble_preds, test_y$Quality)
specificity <- specificity(ensemble_preds, test_y$Quality)
precision <- precision(ensemble_preds, test_y$Quality)
results <- bind_rows(results,data_frame(Model = "Ensemble", Accuracy =
accuracy, Sensitivity = sensitivity, Specificity = specificity, Precision =
precision))
results %>% knitr::kable()
```

Model	Accuracy	Sensitivity	Specificity	Precision
Logistic Regression	0.7497436	0.8427877	0.5893855	0.7796102
LDA	0.7538462	0.8508914	0.5865922	0.7800892
QDA	0.7179487	0.9027553	0.3994413	0.7215026
Loess	0.7589744	0.8363047	0.6256983	0.7938462
KNN	0.7487179	0.8282010	0.6117318	0.7861538
Random Forest	0.7723077	0.8282010	0.6759777	0.8149920
Ensemble	0.7620513	0.8444084	0.6201117	0.7929985

3 Results

The random forest model has the highest accuracy. The accuracy is still relatively low overall, which is a result of irregular distributions, moderate to low correlations, and differences between red and white wine. The model could be further improved by running separate models for red and white wine respectively. In addition, the results could be improved by increasing the dataset to include additional variables such as region, varietal, brand, vintage, and type of rating (critic, quality assurance, etc.).

Model	Accuracy	Sensitivity	Specificity	Precision
Logistic Regression	0.7497436	0.8427877	0.5893855	0.7796102
LDA	0.7538462	0.8508914	0.5865922	0.7800892
QDA	0.7179487	0.9027553	0.3994413	0.7215026
Loess	0.7589744	0.8363047	0.6256983	0.7938462
KNN	0.7487179	0.8282010	0.6117318	0.7861538
Random Forest	0.7723077	0.8282010	0.6759777	0.8149920
Ensemble	0.7620513	0.8444084	0.6201117	0.7929985

When running the model for white wine only, the accuracy improves across all models. Random forest remains most effective at **77.56%**. As the improvement was not substantial, this was not run for red wine considering that Vinho Verde is most commonly white.

Model	Accuracy	Sensitivity	Specificity	Precision
Logistic Regression	0.7606019	0.8770833	0.5378486	0.7839851
LDA	0.7674419	0.8875000	0.5378486	0.7859779
QDA	0.7318741	0.9145833	0.3824701	0.7390572
Loess	0.7701778	0.8791667	0.5617530	0.7932331
KNN	0.7510260	0.8562500	0.5498008	0.7843511
Random Forest	0.7756498	0.8500000	0.6334661	0.8160000
Ensemble	0.7729138	0.8854167	0.5577689	0.7929104

When running the final model for white wine against the validation set, the accuracy is comparable. Random forest remains the most effective at **79.69%**. This increased from the prior evaluation of the model on the test set.

Model	Accuracy	Sensitivity	Specificity	Precision
Logistic Regression	0.7555249	0.8577406	0.5569106	0.7899807
LDA	0.7541436	0.8577406	0.5528455	0.7884615
QDA	0.7251381	0.9037657	0.3780488	0.7384615
Loess	0.7638122	0.8598326	0.5772358	0.7980583
KNN	0.7527624	0.8493724	0.5650407	0.7914230
Random Forest	0.7969613	0.8702929	0.6544715	0.8303393
Ensemble	0.7651934	0.8577406	0.5853659	0.8007812

4 Conclusion

In conclusion, the random forest model is the optimal model to predict wine quality based on the current defined variables and dataset. The model could be further improved by increasing the dataset and including additional variables such as region, varietal, brand, vintage, and type of rating (critic, quality assurance, etc.).

As the current model is limited, and missing information that could drive quality and scores overall, prior to moving forward on a broader dataset it would be critical to repeat this exercise to evaluate variable relationships and model accuracy.

The findings of this analysis will be valuable to build a future, more sophisticated model to support the quality assurance process in all stages of the product lifecycle. Improving quality assurance processes will effectively reduce risks associated with poor quality product such as increased cost, poor product ratings impacting revenue potential, decreased customer satisfaction, and in worst case scenarios, product recall. Early detection of quality issues can mitigate downstream risk.

References

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.