# Predicting Wine Quality

*Hamza Mohd Zubair*

*25 October 2016*

# Contents

# Chapter 1

# Introduction

## 1.1 Problem Statement

Testing the quality of wine and tasting it requires manpower and equipment. The aim of the project is to reduce man power which wine companies hire to taste the quality of wine before launching into the market. They are spending huge amount on hiring healthy volunteers and on their retention policies/strategies. We would like to predict the quality of wine based on chemical properties which are already known and easy to calculate using sensors.

## 1.2 Data

Our task is to build classification models which will classify the quality of wine depending on multiple physico-chemical factors. Given below is a sample of the data set that we are using to predict the quality of wine:

Table 1.1: Red Wine Sample Data (Columns: 1-6)

| Fixed Acidity | Volatile Acidity | Citric Acid | Residual Sugar | Chlorides | Free Sulphurdioxide |
|---|---|---|---|---|---|
| 7.4 | 0.70 | 0.00 | 1.9 | 0.076 | 11 |
| 7.8 | 0.88 | 0.00 | 2.6 | 0.098 | 25 |
| 7.8 | 0.76 | 0.04 | 2.3 | 0.092 | 15 |
| 11.2 | 0.28 | 0.56 | 1.9 | 0.075 | 17 |
| 7.4 | 0.70 | 0.00 | 1.9 | 0.076 | 11 |
| 7.4 | 0.66 | 0.00 | 1.8 | 0.075 | 13 |

Table 1.2: Red Wine Sample Data (Columns: 7-12)

| Total Sulfurdioxide | Density | pH | Sulphates | Alcohol | Quality |
|---|---|---|---|---|---|
| 34 | 0.9978 | 3.51 | 0.56 | 9.4 | 5 |
| 67 | 0.9968 | 3.20 | 0.68 | 9.8 | 5 |
| 54 | 0.9970 | 3.26 | 0.65 | 9.8 | 5 |
| 60 | 0.9980 | 3.16 | 0.58 | 9.8 | 6 |
| 34 | 0.9978 | 3.51 | 0.56 | 9.4 | 5 |
| 40 | 0.9978 | 3.51 | 0.56 | 9.4 | 5 |

Table 1.3: White Wine Sample Data (Columns: 1-6)

| Fixed Acidity | Volatile Acidity | Citric Acid | Residual Sugar | Chlorides | Free Sulphurdioxide |
|---|---|---|---|---|---|
| 7.0 | 0.27 | 0.36 | 20.7 | 0.045 | 45 |
| 6.3 | 0.30 | 0.34 | 1.6 | 0.049 | 14 |
| 8.1 | 0.28 | 0.40 | 6.9 | 0.050 | 30 |
| 7.2 | 0.23 | 0.32 | 8.5 | 0.058 | 47 |
| 7.2 | 0.23 | 0.32 | 8.5 | 0.058 | 47 |
| 8.1 | 0.28 | 0.40 | 6.9 | 0.050 | 30 |

Table 1.4: White Wine Sample Data (Columns: 7-12)

| Total Sulfurdioxide | Density | pH | Sulphates | Alcohol | Quality |
|---|---|---|---|---|---|
| 170 | 1.0010 | 3.00 | 0.45 | 8.8 | 6 |
| 132 | 0.9940 | 3.30 | 0.49 | 9.5 | 6 |
| 97 | 0.9951 | 3.26 | 0.44 | 10.1 | 6 |
| 186 | 0.9956 | 3.19 | 0.40 | 9.9 | 6 |
| 186 | 0.9956 | 3.19 | 0.40 | 9.9 | 6 |
| 97 | 0.9951 | 3.26 | 0.44 | 10.1 | 6 |

As you can see in the table below we have the following 11 variables, using which we have to correctly predict the quality of the wines:

Table 1.5: Predictor Variables

| S.No. | Predictor |
|---|---|
| 1 | Fixed Acidity |
| 2 | Volatile Acidity |
| 3 | Citric Acid |
| 4 | Residual Sugar |
| 5 | Chlorides |
| 6 | Free Sulphurdioxide |
| 7 | Total Sulfurdioxide |
| 8 | Density |
| 9 | pH |
| 10 | Sulphates |
| 11 | Alcohol |

# Chapter 2

# Methodology

## 2.1 Pre Processing

Any predictive modeling requires that we look at the data before we start modeling. However, in data mining terms *looking at data* refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as **Exploratory Data Analysis**. To start this process we will first try and look at all the probability distributions of the variables. Most analysis like regression, require the data to be normally distributed. We can visualize that in a glance by looking at the probability distributions or probability density functions of the variable.

In Figure 2.1 we have plotted the probability density functions of all the chemical properties we have available in the data as well as the dependent quality variable. The blue lines indicate Kernel Density Estimations (KDE)[1] of the variable. The red lines represent the normal distribution. So as you can see in the figure most variables either very closely, or somewhat imitate the normal distribution. The distributions for the white wine are very similar and therefore are not shown in this section. Those plots can be viewed in the Appendix: (Fig: 3.1).

### 2.1.1 Outlier Analysis

We can clearly observe from these probability distributions that most of the variables are skewed, for example, *Residual Sugar* , *Chlorides*, *Total Sulphurdioxide* and *Sulphates*. The skew in these distributions can be most likely explained by the presence of outliers and extreme values in the data. We can see the effect of the skew in figures 2.2. The red vertical line represents the mean and we can see that the mean is slightly displaced from the position of the median. This is clearly the effect of outliers and extreme values.

One of the other steps of **pre-processing** apart from checking for normality is the presence of outliers. In this case we use a classic approach of removing outliers, *Tukey's method.*[2] We visualize the outliers using *boxplots*.

In figure 2.3 we have plotted the boxplots of the 11 predictor variables with respect to each quality value ranging from 3 to 8. A lot of useful inferences can be made from these plots. First as you can see, we have a lot of outliers and extreme values in each of the data set. Similar plots for White Wine data set can be seen in the appendix (Fig: 3.2).

[1] In Statistics, Kernel Density Estimations is a non-parametric way to estimate the Probability Density Function. Using KDE inferences about the population are made based on the finite data sample.

[2] In Tukey's Method, outliers have been defined as the data points which are $\pm 1.5SD$, and should be removed from the data. It was given by J. W. Tukey in his famous 1977 book *Exploratory Data Analysis*

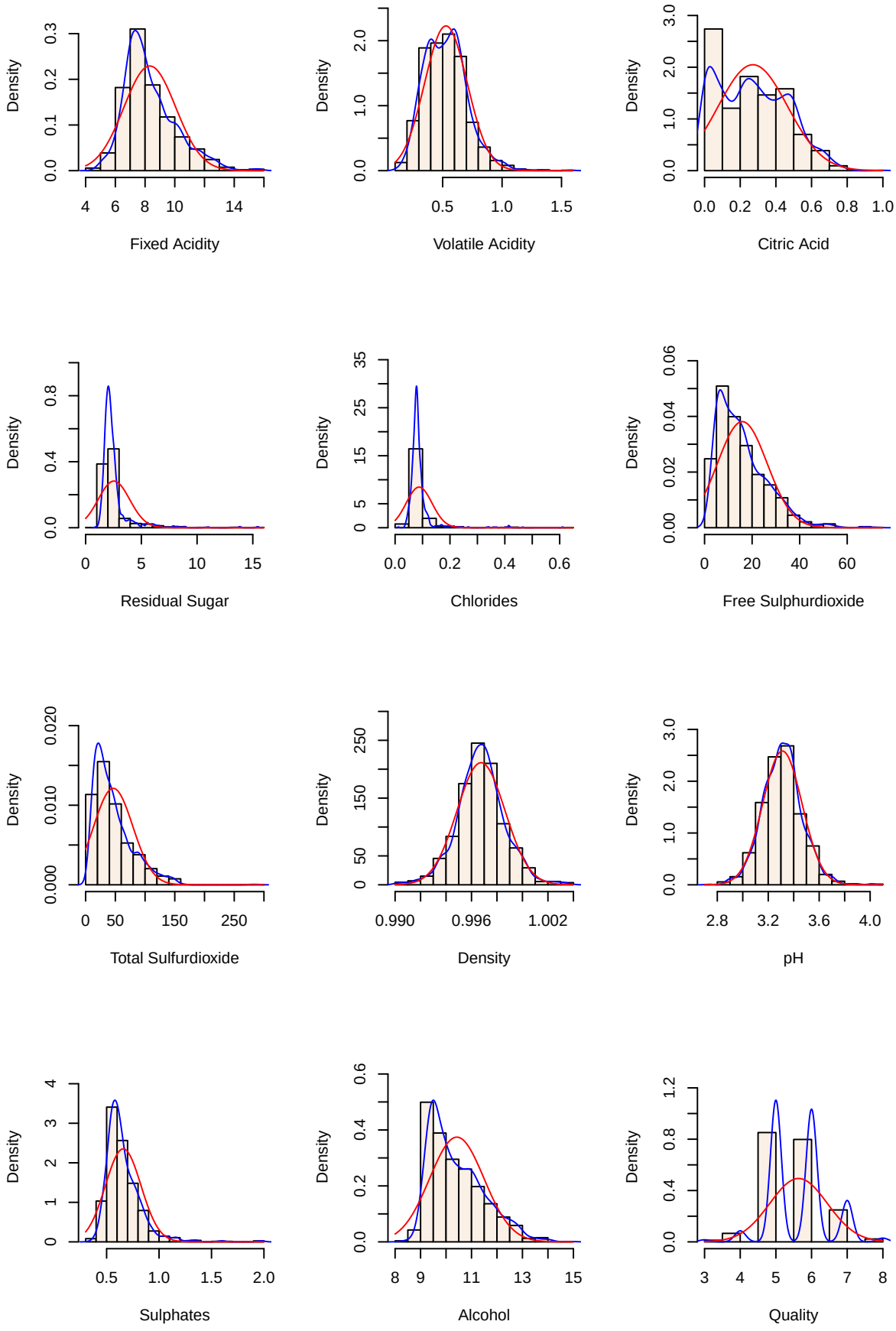


Figure 2.1: Probability Density Functions of Red Wine data (See R code in Appendix)

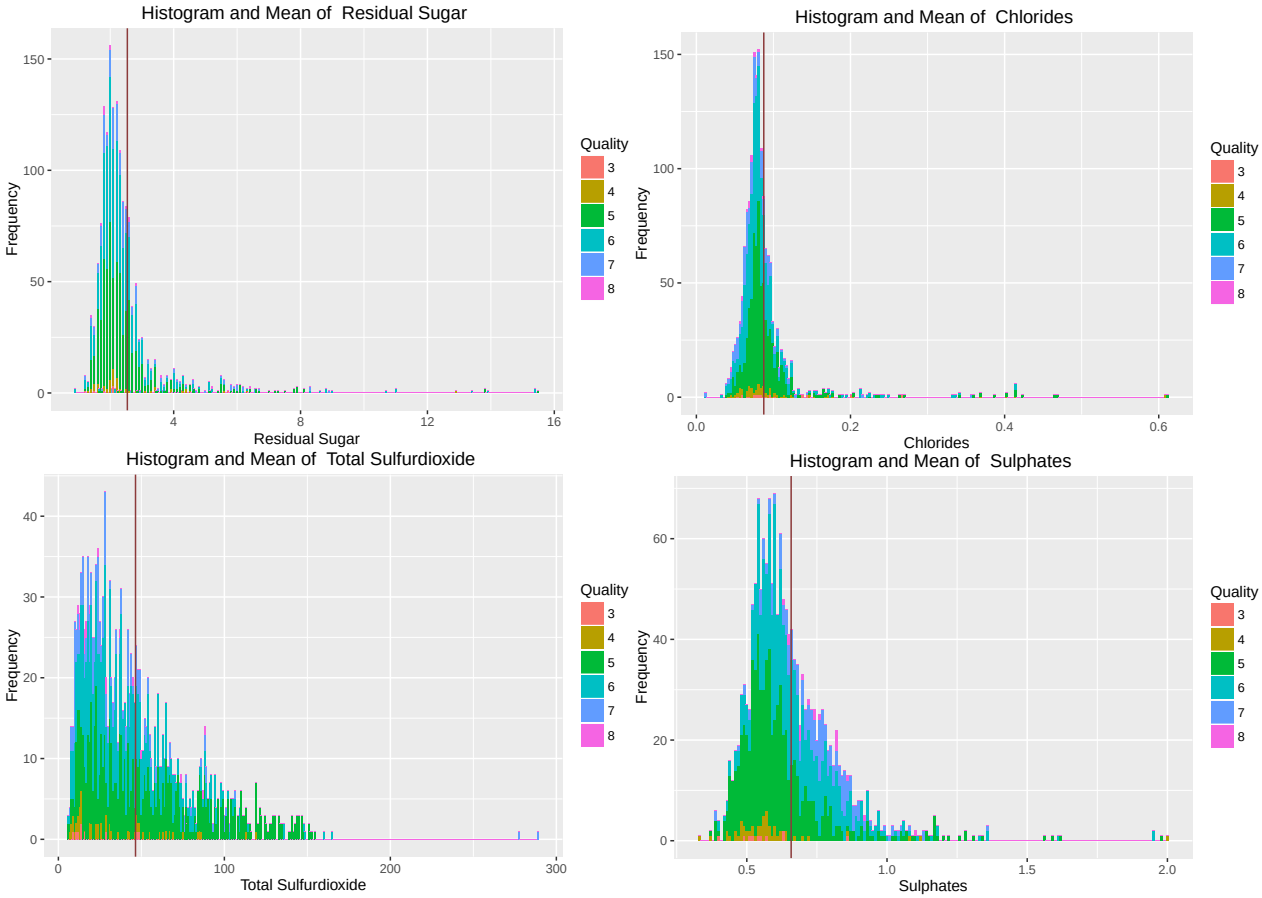


Figure 2.2: Histogram and Mean of Predictor Variables

Other useful inferences can also be drawn from these plots. For example, If you compare the quality boxplots for each of the predictor variables in the Red Wine data, you can see that there is not much difference between the median values of each quality. Even if some difference exists, for example, in *Volatile Acidity* and *Density* where we can see an overall decrease in values with increase in *Quality*, and in *Citric Acid*, *Alcohol* and *Sulphates*, where we can see an overall increase in values with increase in Quality, these differences do not seem to be statistically significant because of extreme overlaps between conditions. Similar and somewhat pronounced patterns can also be seen in the White wine data set.

We will process this data and plot it again after outlier removal in the next sections.

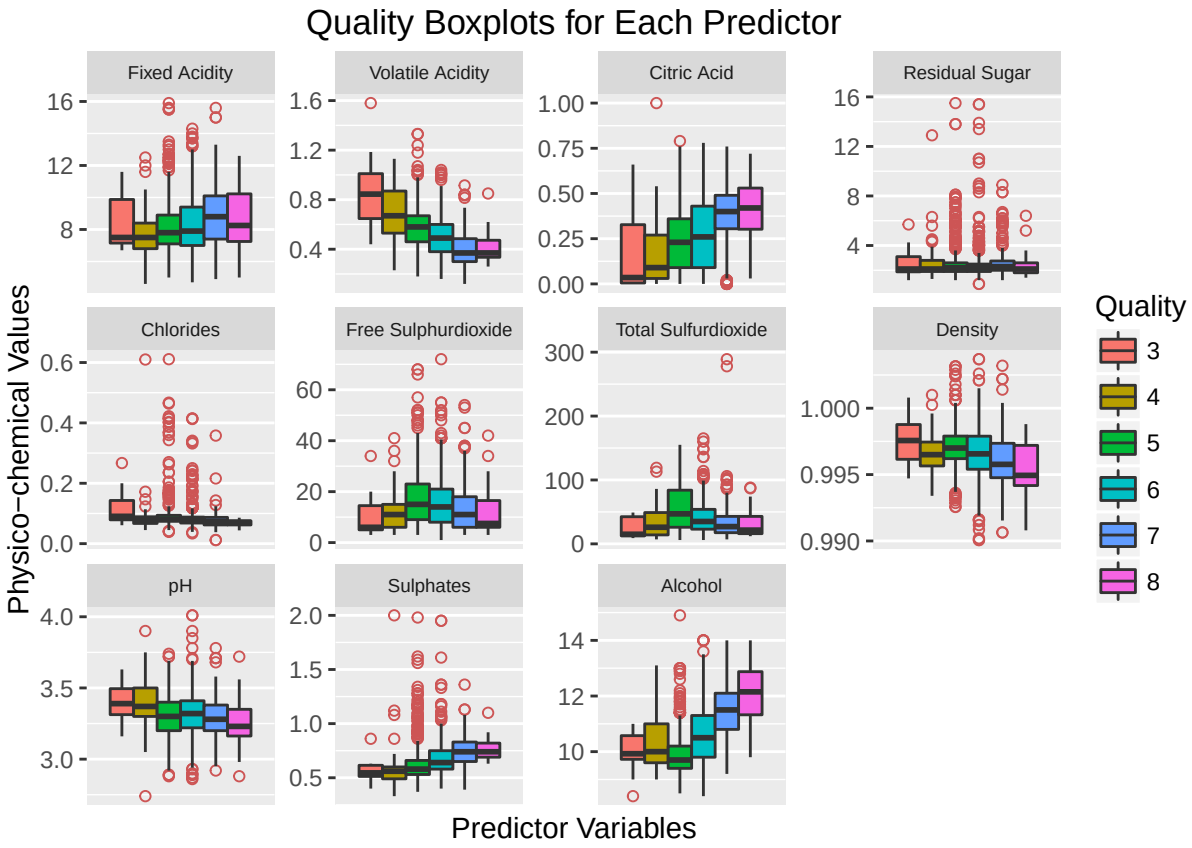


Figure 2.3: Quality vs Predictor Boxplots for **Red Wine** (See R code in Appendix)

You can see in figure 2.4 what effect outliers have on the normalisation of our data. Similar graphs can be seen in the appendix (Figures 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9)

After performing outlier Analysis and removing outliers using *Tukey's Boxplot Method*, We plot the Predictor vs Quality boxplots again. You can compare the differences in Fig:2.5 for Red Wine and Fig:2.6 for White Wine

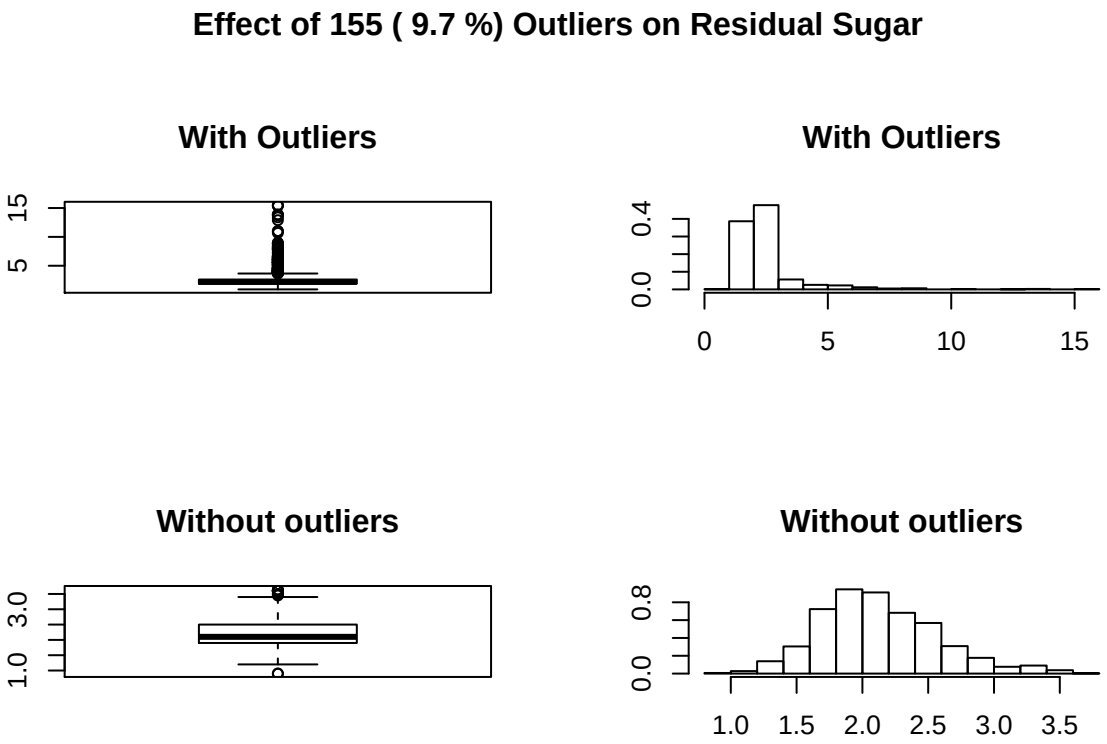


Figure 2.4: Effect of Outlier on a Predictor Variable of Red Wine (See R code in Appendix)

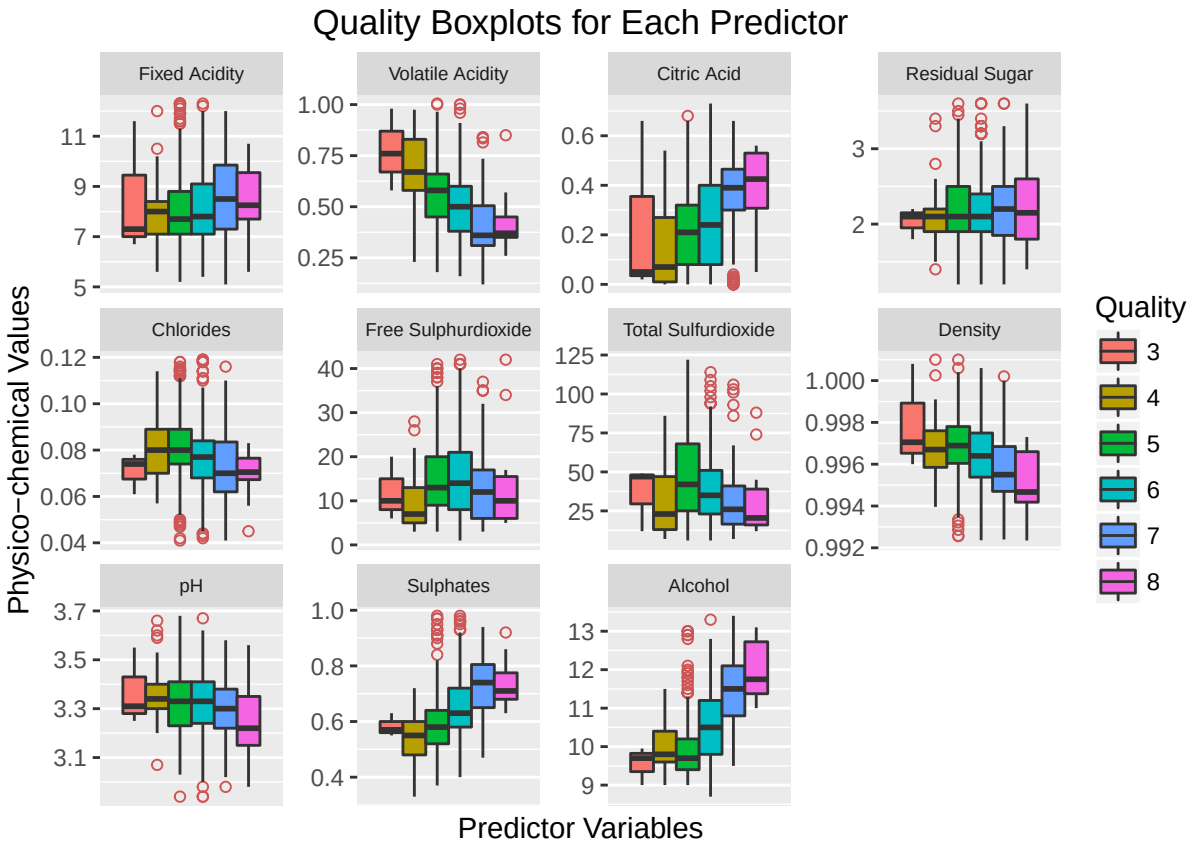


Figure 2.5: Quality vs Predictor Boxplots for **Red Wine** Predictors After Outlier Removal

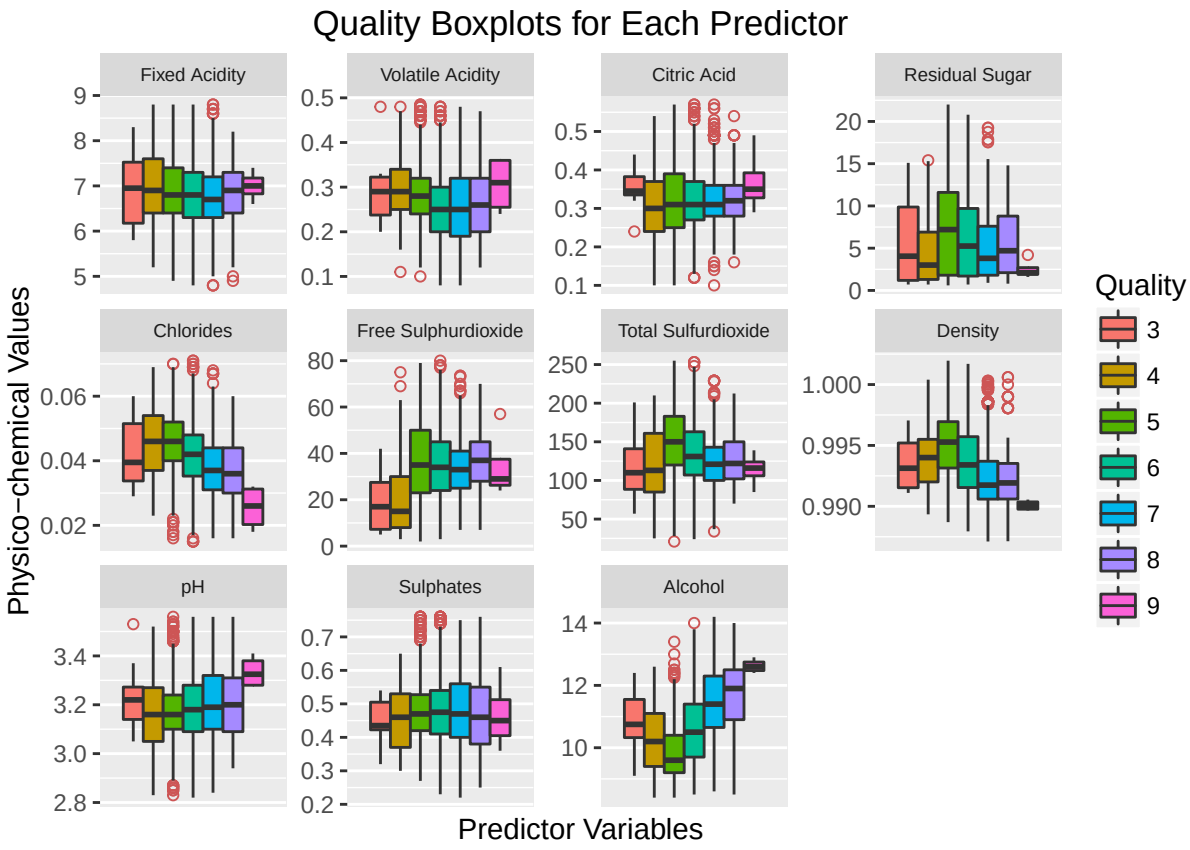


Figure 2.6: Quality vs Predictor Boxplots for **White Wine** Predictors After Outlier Removal

### 2.1.2 Feature Selection

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction. There are several methods of doing that. Below we have used *Random Forests* to perform features selection.

#### 2.1.2.1 Red Wine

```
imppred <- randomForest(quality ~ ., data = d1,
    ntree = 100, keep.forest = FALSE, importance = TRUE)
importance(imppred, type = 1)
##                        %IncMSE
## fixed.acidity        12.62263
## volatile.acidity     19.74335
## citric.acid          12.31410
## residual.sugar       11.20579
## chlorides            13.47935
## free.sulfur.dioxide  11.57610
## total.sulfur.dioxide 16.22204
## density              19.82939
## pH                   12.67484
## sulphates            22.05664
## alcohol              24.57563
```

#### 2.1.2.2 White Wine

```
imppred <- randomForest(quality ~ ., data = d2,
    ntree = 100, keep.forest = FALSE, importance = TRUE)
importance(imppred, type = 1)
##                        %IncMSE
## fixed.acidity        23.56129
## volatile.acidity     44.55030
## citric.acid          30.69948
## residual.sugar       27.37064
## chlorides            25.41836
## free.sulfur.dioxide  46.03742
## total.sulfur.dioxide 26.67965
## density              22.26679
## pH                   29.20979
## sulphates            24.42959
## alcohol              32.10414
```

One thing that becomes clear from above predictor variable importance values is that both the red wine and white wine have different chemical behaviour. Which means both of them need separate models. We can see that *Alcohol* has the highest prediction power for red wine whereas *Volatile Acidity* has the highest prediction power in the case of white wine.

Another step of Exploratory Data Analysis is to look for highly correlated variables in the data. A very simple way of looking at correlations in the data is shown below. Without much detail and at a glance you can see that Red Wine data does not any variables with higher correlation than 0.8, but *Residual Sugar* and *Density* in White wine data are highly correlated (0.8 - 0.9).

**Red Wine**

```
symnum(cor(d1.r))
##                      FA VA CA RS C FS TS D p S A Q
## Fixed Acidity        1
## Volatile Acidity        1
## Citric Acid          ,  .  1
## Residual Sugar             1
## Chlorides                     1
## Free Sulphurdioxide              1
## Total Sulfurdioxide              ,  1
## Density              ,     .  .          1
## pH                   ,     .             . 1
## Sulphates                  .     .           1
## Alcohol                                  .     1
## Quality                 .                      . 1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

**White Wine**

```
symnum(cor(d2.r))
##                      FA VA CA RS C FS TS D p S A Q
## Fixed Acidity        1
## Volatile Acidity        1
## Citric Acid                1
## Residual Sugar             1
## Chlorides                     1
## Free Sulphurdioxide              1
## Total Sulfurdioxide              .    ,  1
## Density                          +       .  1
## pH                   .                      1
## Sulphates                                     1
## Alcohol                          .  .    .  ,     1
## Quality                                  .     . 1
## attr(,"legend")
## [1] 0 ' ' 0.3 '.' 0.6 ',' 0.8 '+' 0.9 '*' 0.95 'B' 1
```

## 2.2 Modeling

### 2.2.1 Model Selection

In our early stages of analysis during pre-processing we have come to understand that red wine and white wine have completely different chemical behaviours. Therefore, we can neither combine the data sets nor use a single model for predicting both variables. Hence, we need to analyse the data sets separately and generate separate models for both types of data sets.

The dependent variable can fall in either of the four categories:

1. Nominal
2. Ordinal
3. Interval
4. Ratio

If the dependent variable, in our case *Quality*, is Nominal the only predictive analysis that we can perform is **Classification**, and if the dependent variable is Interval or Ratio the normal method is to do a **Regression** analysis, or classification after binning. But the dependent variable we are dealing with is *Ordinal*, for which both classification and regression can be done, because even though the Quality variable has categories, these categories have an order associated with them, which is ranks.

You always start your model building from the most simplest to more complex. Therefore we use Multiple Linear Regression.

### 2.2.2 Multiple Linear Regression

```
lrmodel.red <- lm(Quality ~ ., data = d1.r)
summary(lrmodel.red)
##
## Call:
## lm(formula = Quality ~ ., data = d1.r)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.68911 -0.36652 -0.04699  0.45202  2.02498
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.197e+01  2.119e+01   1.036   0.3002
## `Fixed Acidity`        2.499e-02  2.595e-02   0.963   0.3357
## `Volatile Acidity`    -1.084e+00  1.211e-01  -8.948  < 2e-16 ***
## `Citric Acid`         -1.826e-01  1.472e-01  -1.240   0.2150
## `Residual Sugar`       1.633e-02  1.500e-02   1.089   0.2765
## Chlorides             -1.874e+00  4.193e-01  -4.470 8.37e-06 ***
## `Free Sulphurdioxide`  4.361e-03  2.171e-03   2.009   0.0447 *
## `Total Sulfurdioxide` -3.265e-03  7.287e-04  -4.480 8.00e-06 ***
## Density               -1.788e+01  2.163e+01  -0.827   0.4086
## pH                    -4.137e-01  1.916e-01  -2.159   0.0310 *
## Sulphates              9.163e-01  1.143e-01   8.014 2.13e-15 ***
## Alcohol                2.762e-01  2.648e-02  10.429  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.648 on 1587 degrees of freedom
## Multiple R-squared:  0.3606, Adjusted R-squared:  0.3561
## F-statistic: 81.35 on 11 and 1587 DF,  p-value: < 2.2e-16
```

As you can see the *Adjusted R-squared* value, we can explain only about 36% of the data using our multiple linear regression model. This is not very impressive, but at least looking at the *F-statistic* and combined p-value we can reject the null hypothesis that target variable does not depend on any of the predictor variables.

Looking at the significance values of some of the predictor we can see that there is some scope of improvement in this model. We can improve this multiple linear regression model using *ANOVA*.

```
kable(anova(lrmodel.red), booktabs = T)
```

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) |
|---|---|---|---|---|---|
| 'Fixed Acidity' | 1 | 16.0376825 | 16.0376825 | 38.1923671 | 0.0000000 |
| 'Volatile Acidity' | 1 | 143.5727497 | 143.5727497 | 341.9062054 | 0.0000000 |
| 'Citric Acid' | 1 | 0.0244028 | 0.0244028 | 0.0581132 | 0.8095345 |
| 'Residual Sugar' | 1 | 0.1580759 | 0.1580759 | 0.3764443 | 0.5396003 |
| Chlorides | 1 | 13.0618667 | 13.0618667 | 31.1057168 | 0.0000000 |
| 'Free Sulphurdioxide' | 1 | 2.9742159 | 2.9742159 | 7.0828404 | 0.0078611 |
| 'Total Sulfurdioxide' | 1 | 30.0926593 | 30.0926593 | 71.6630904 | 0.0000000 |
| Density | 1 | 61.3103726 | 61.3103726 | 146.0054007 | 0.0000000 |
| pH | 1 | 7.1536526 | 7.1536526 | 17.0358108 | 0.0000386 |
| Sulphates | 1 | 55.6965637 | 55.6965637 | 132.6365956 | 0.0000000 |
| Alcohol | 1 | 45.6721611 | 45.6721611 | 108.7643394 | 0.0000000 |
| Residuals | 1587 | 666.4107004 | 0.4199185 | NA | NA |

```
lrmodel.red2 <- update(lrmodel.red, . ~ . - `Citric Acid` -
    `Residual Sugar`)
summary(lrmodel.red2)
##
## Call:
## lm(formula = Quality ~ `Fixed Acidity` + `Volatile Acidity` +
##     Chlorides + `Free Sulphurdioxide` + `Total Sulfurdioxide` +
##     Density + pH + Sulphates + Alcohol, data = d1.r)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.69252 -0.36640 -0.04809  0.46154  2.02199
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            9.2835190 17.0048169   0.546   0.5852
## `Fixed Acidity`        0.0032712  0.0219874   0.149   0.8817
## `Volatile Acidity`    -1.0096872  0.1020966  -9.890  < 2e-16 ***
## Chlorides             -2.0099867  0.4045139  -4.969 7.46e-07 ***
## `Free Sulphurdioxide`  0.0050613  0.0021275   2.379   0.0175 *
## `Total Sulfurdioxide` -0.0034664  0.0007018  -4.939 8.68e-07 ***
## Density               -4.9004699 17.3651139  -0.282   0.7778
## pH                    -0.4705130  0.1806878  -2.604   0.0093 **
## Sulphates              0.8892038  0.1120163   7.938 3.85e-15 ***
## Alcohol                0.2851370  0.0224255  12.715  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6481 on 1589 degrees of freedom
## Multiple R-squared:  0.3595, Adjusted R-squared:  0.3559
## F-statistic:  99.1 on 9 and 1589 DF,  p-value: < 2.2e-16
```

Using ANOVA we saw that *Citric Acid* and *Residual Sugar* contribute the least for the reduction of the fitting error of the model. However removing these variables also did not change the predictive power of our regression model. Therefore, this is the maximum accuracy that we can get from this model.

### 2.2.3 Regression Trees

Now we will try and use a different regression model to predict our *Quality* target variable. We will use a regression tree to predict the values of our target variable.

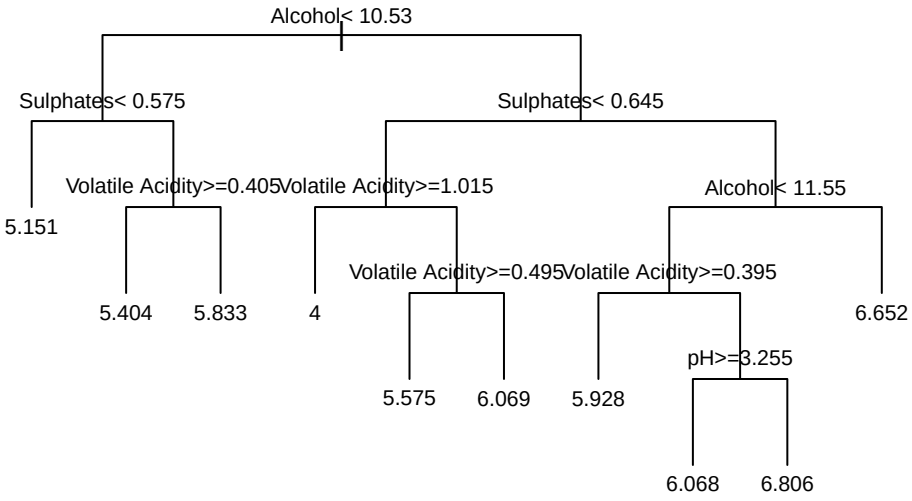


Figure 2.7: Regression Tree for Red Wine

### 2.2.4 Classification

Using Classification for prediction analysis in this case is not normal, though it can be done. The reason is, the values of the target variable, *quality* are ranks. And Rank is not a purely categorical variable, like male, female categories because there is an order in rank. For example rank 1 is better than rank 2. Such variables are called ordinal. Though Classification predictions have been done on ordinal variables it is not a recommended approach, because the information stored in terms of the order is lost.

# Chapter 3

# Conclusion

## 3.1 Model Evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:

1. Predictive Performance
2. Interpretability
3. Computational Efficiency

In our case of Wine Data, the latter two, *Interpretability* and *Computation Efficiency*, do not hold much significance. Therefore we will use *Predictive performance* as the criteria to compare and evaluate models.

Predictive performance can be measured by comparing Predictions of the models with real values of the taret variables, and calculating some average error measure.

### 3.1.1 Mean Absolute Error (MAE)

MAE is one of the error measures used to calculate the predictive performance of the model. We will apply this measure to our models that we have generated in the previous section.

```
lrm.pred.red <- predict(lrmodel.red, d1.r)
rt.pred.red <- predict(rtmodel.red, d1.r)
mae.lrm.red <- mean(abs(lrm.pred.red - d1.r[,
    12]))
mae.rt.red <- mean(abs(rt.pred.red - d1.r[, 12]))
mae.lrm.red
## [1] 0.50049
mae.rt.red
## [1] 0.5023972
```

### 3.1.2 Mean Squared Error (MSE)

MSE can be obtained as follows

```
(mse.lrm.red <- mean((lrm.pred.red - d1.r[, 12])^2))
## [1] 0.4167672
```

```
(mse.rt.red <- mean((rt.pred.red - d1.r[, 12])^2))
## [1] 0.4122482
```

## 3.2 Model Selection

We can see that both models perform comparatively on average and therefore we can select either of the two models without any loss of information.

# Appendix A - Extra Figures

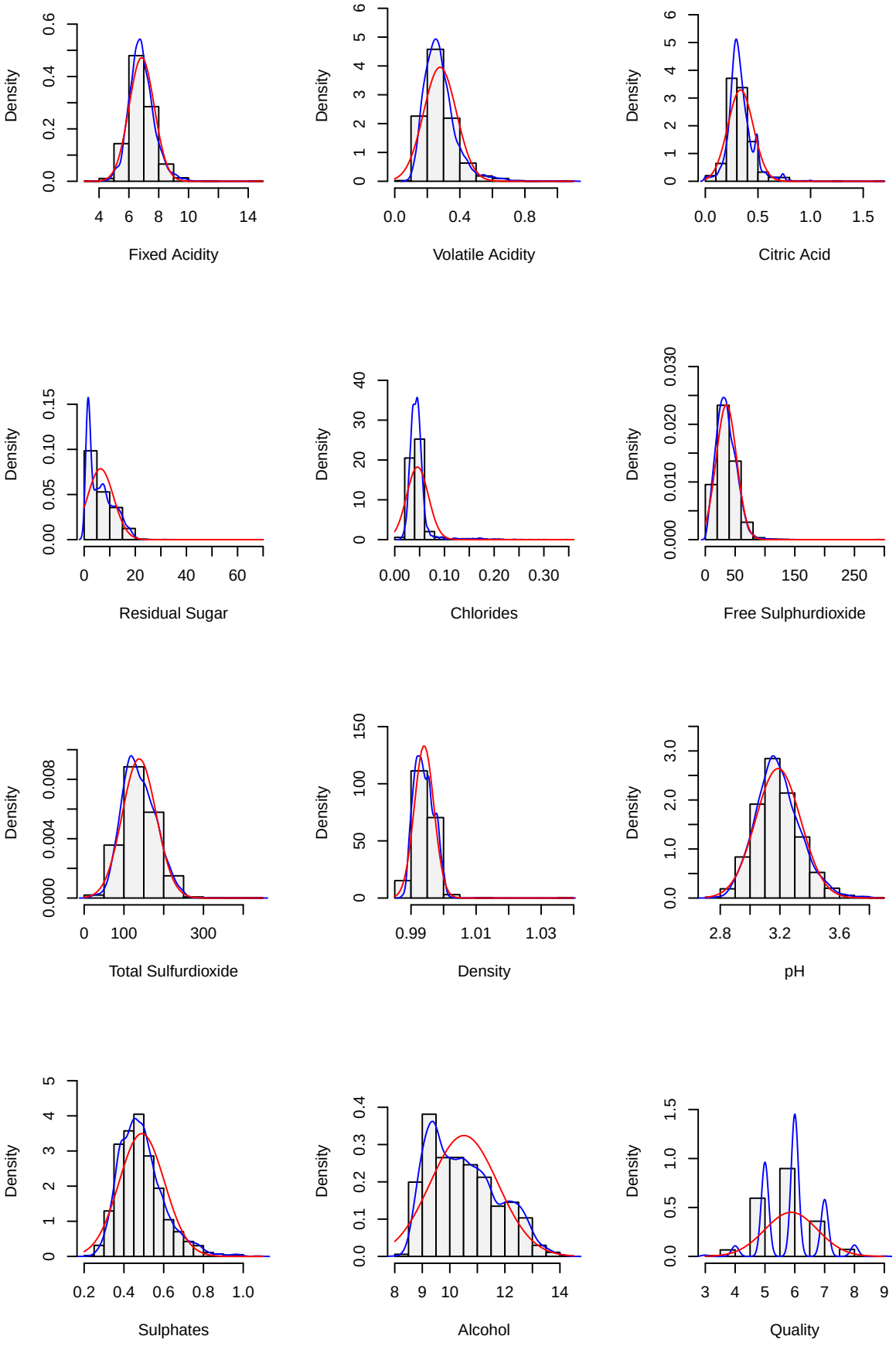


Figure 3.1: Probability Density Functions of White Wine data (See R code in Appendix)

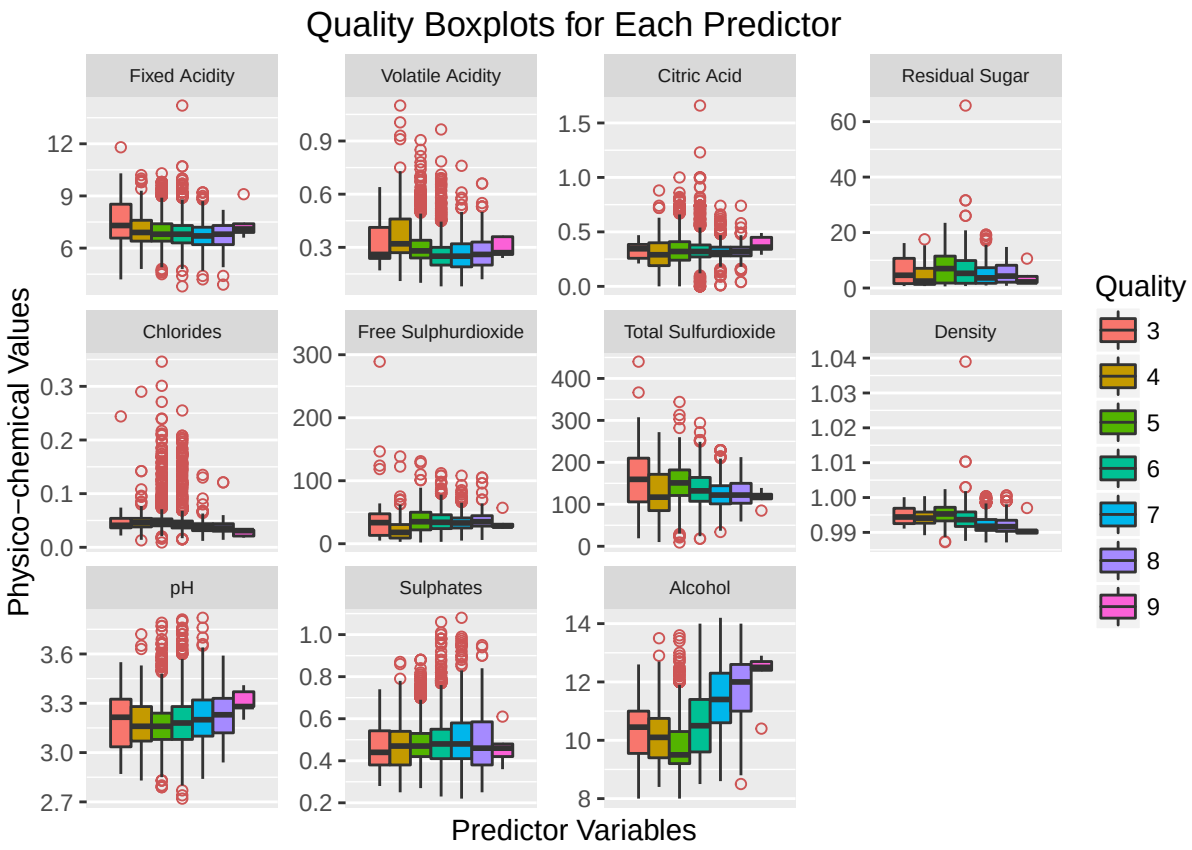


Figure 3.2: Quality Boxplots for all the Predictor Variables for **White Wine** (See R code in Appendix)

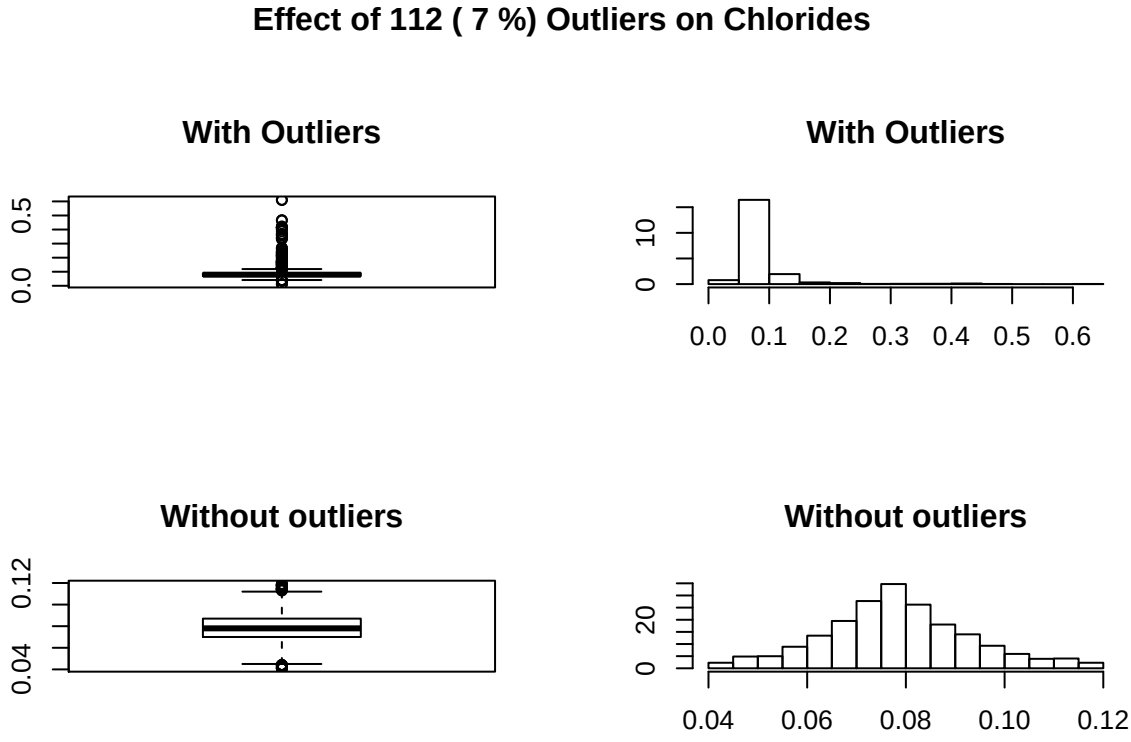


Figure 3.3: Effect of Outliers on Predictor Variables of **Red Wine**

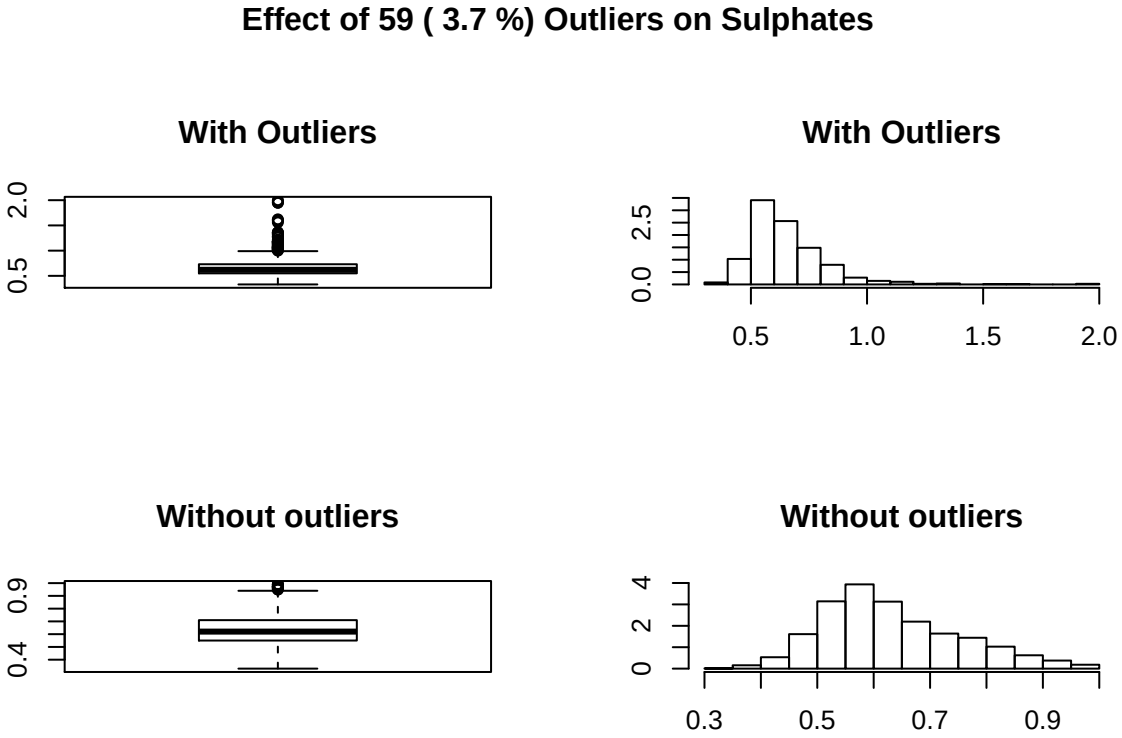


Figure 3.4: Effect of Outliers on Predictor Variables of Red Wine

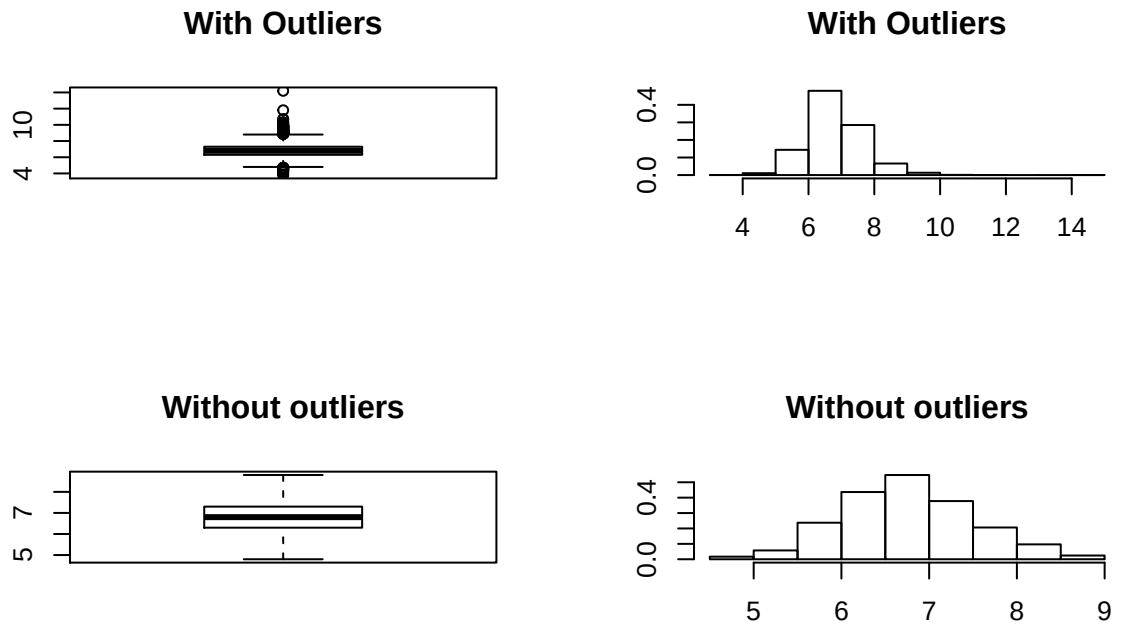


Figure 3.5: Effect of Outliers on Predictor Variables of White Wine

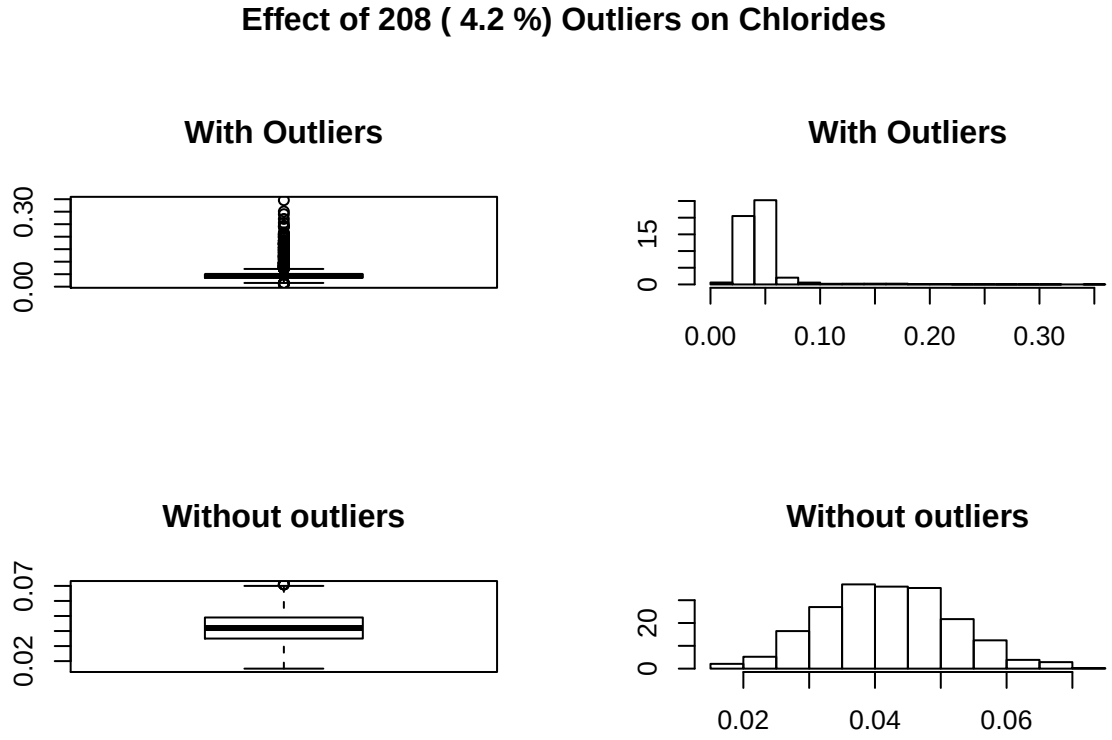


Figure 3.6: Effect of Outliers on Predictor Variables of White Wine

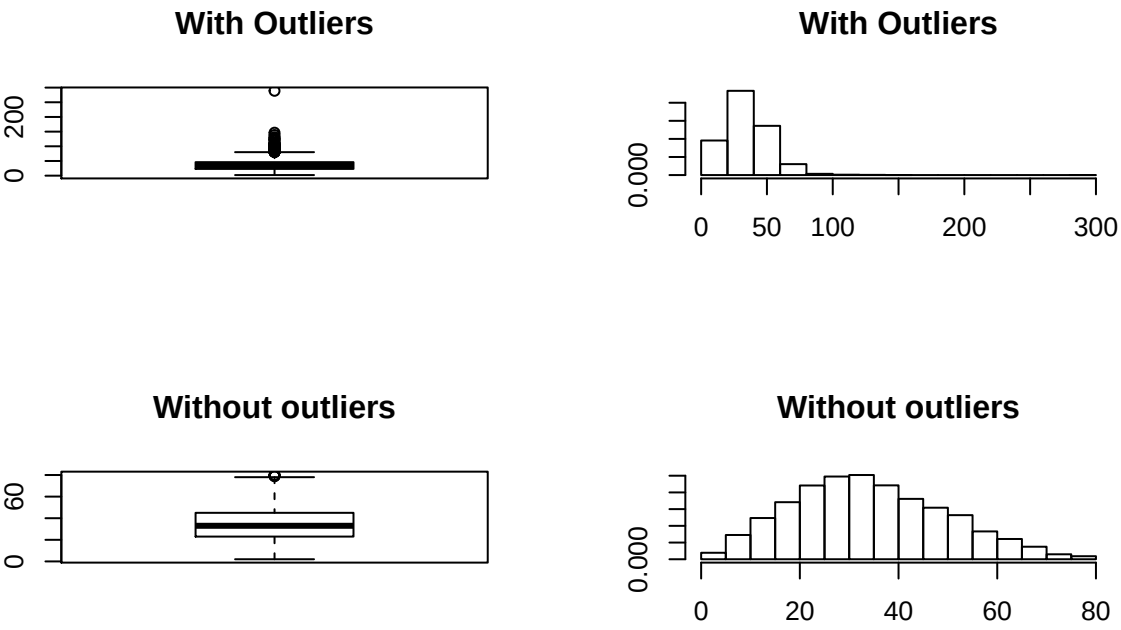


Figure 3.7: Effect of Outliers on Predictor Variables of White Wine

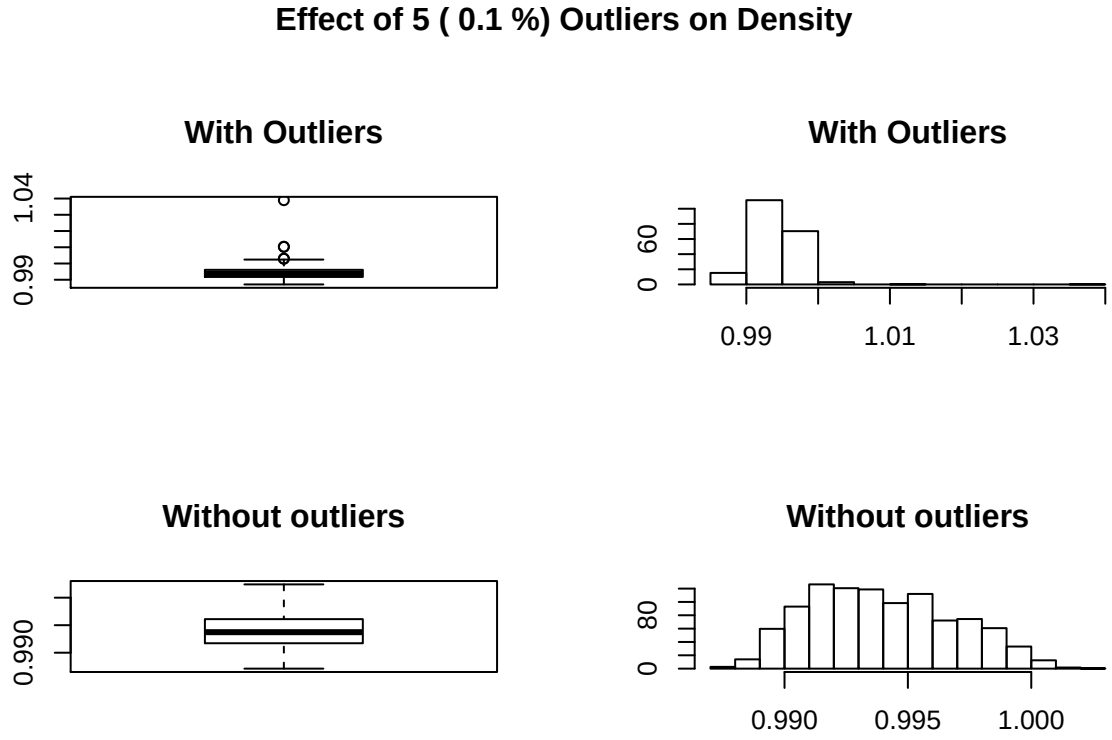


Figure 3.8: Effect of Outliers on Predictor Variables of White Wine

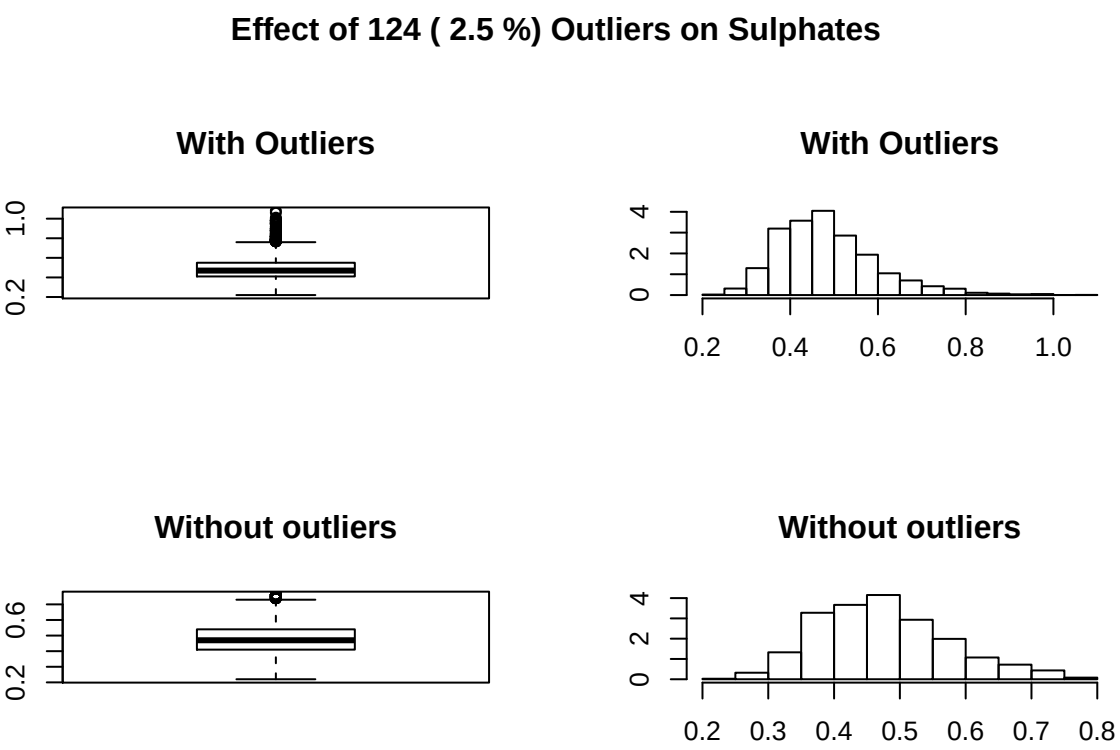


Figure 3.9: Effect of Outliers on Predictor Variables of White Wine

# Appendix B - R Code

### Red Wine Muliple Histograms (Fig: 2.1)

```
multi.hist(d1.r, main = NA, dcol = c("blue", "red"),
    dlty = c("solid", "solid"), bcol = "linen")
```

### Red Wine Histograms with Means (Fig: 2.2)

```
d1.b <- d1.r
d1.b[, 12] <- as.factor(d1.b[, 12])
lh <- c(4, 5, 7, 10)
bw <- c(0.05, 0.003, 0.6, 0.01)
for (i in 1:4) {
    base <- ggplot(d1.b)
    layering.h <- geom_histogram(aes(x = d1.b[,
        lh[i]], fill = Quality), binwidth = bw[i])
    layering.l <- geom_vline(aes(xintercept = mean(d1.b[,
        lh[i]])), color = "indianred4")
    theming.x <- xlab(colnames(d1.b)[lh[i]])
    theming.y <- ylab("Frequency")
    theming.t <- ggtitle(paste("Histogram and Mean of ",
        colnames(d1.b)[lh[i]], ""))
    themed <- base + layering.h + layering.l +
        theming.x + theming.y + theming.t
    print(themed)
}
```

### White Wine Muliple Histograms (Fig: 3.1)

```
multi.hist(d2.r, main = NA, dcol = c("blue", "red"),
    dlty = c("solid", "solid"), bcol = "grey95")
```

### Red Wine Boxplots (Fig: 2.3)

```
qboxp <- function(df) {
    d1.b <- df
    d1.b[, 12] <- as.factor(d1.b[, 12])
```

```
    df.m <- melt(d1.b, id.var = "Quality")
    base <- ggplot(data = df.m, aes(x = variable,
        y = value))
    layering <- geom_boxplot(aes(fill = Quality),
        outlier.shape = 1, outlier.color = "indianred3")
    scaling <- scale_x_discrete(breaks = NULL)
    faceting <- facet_wrap(~variable, scales = "free")
    theming.x <- xlab("Predictor Variables")
    theming.y <- ylab("Physico-chemical Values")
    theming.title <- ggtitle("Quality Boxplots for Each Predictor")
    theming.facet <- theme(strip.text = element_text(size = 7),
        axis.text.x = element_blank(), axis.ticks.x = element_blank())
    themed <- base + layering + scaling + faceting +
        theming.x + theming.y + theming.title +
        theming.facet
    themed
}
qboxp(d1.r)
```

### Effect of Outliers (Fig: 2.4)

```
outtona <- function(vl, df) {
    for (i in vl) {
        outlier <- boxplot.stats(df[, i])$out
        df[, i] <- ifelse(df[, i] %in% outlier,
            NA, df[, i])
    }
    return(df)
}

outliereff <- function(varlist, df) {
    for (i in varlist) {
        total = length(df[, i])
        par(mfrow = c(2, 2), oma = c(0, 0, 3,
            0))
        boxplot(df[, i], main = "With Outliers")
        hist(df[, i], main = "With Outliers",
            xlab = NA, ylab = NA, prob = TRUE)
        df <- outtona(i, df)
        boxplot(df[, i], main = "Without outliers")
        hist(df[, i], main = "Without outliers",
            xlab = NA, ylab = NA, prob = TRUE)
        out <- sum(is.na(df[, i]))
        per <- round((out)/total * 100, 1)
        title(paste("Effect of", out, "(", per,
            "%)", "Outliers on", colnames(df)[i],
            sep = " "), outer = TRUE)
    }
}
outliereff(4, d1.r)
d1.on <- outtona(1:11, d1.r)
d1.f <- d1.on[complete.cases(d1.on), ]
```

```
d2.on <- outtona(1:11, d2.r)
d2.f <- d2.on[complete.cases(d2.on), ]
```

### White Wine Boxplots (Fig: 3.2)

```
qboxp(d2.r)
```

### Complete R File

```
knitr::opts_chunk$set(echo = FALSE, tidy.opts = list(width.cutoff = 45),
    tidy = TRUE, comment = "##")
library(knitr)
library(tidyverse)
library(xtable)
library(MASS)
library(psych)
library(reshape2)
library(randomForest)
library(rpart)
rm(list = ls())
d1 <- read.csv("./casestudy/winequality-red.csv",
    sep = ";")
d2 <- read.csv("./casestudy/winequality-white.csv",
    sep = ";")
d1.r <- dplyr::rename(d1, `Fixed Acidity` = fixed.acidity,
    `Volatile Acidity` = volatile.acidity, `Citric Acid` = citric.acid,
    `Residual Sugar` = residual.sugar, Chlorides = chlorides,
    `Free Sulphurdioxide` = free.sulfur.dioxide,
    `Total Sulfurdioxide` = total.sulfur.dioxide,
    Density = density, Sulphates = sulphates,
    Alcohol = alcohol, Quality = quality)

d2.r <- dplyr::rename(d2, `Fixed Acidity` = fixed.acidity,
    `Volatile Acidity` = volatile.acidity, `Citric Acid` = citric.acid,
    `Residual Sugar` = residual.sugar, Chlorides = chlorides,
    `Free Sulphurdioxide` = free.sulfur.dioxide,
    `Total Sulfurdioxide` = total.sulfur.dioxide,
    Density = density, Sulphates = sulphates,
    Alcohol = alcohol, Quality = quality)

head(d1.r[, 1:6]) %>% kable(caption = "Red Wine Sample Data (Columns: 1-6)",
    booktabs = TRUE, longtable = TRUE)
head(d1.r[, 7:12]) %>% kable(caption = "Red Wine Sample Data (Columns: 7-12)",
    booktabs = TRUE, longtable = TRUE)
head(d2.r[, 1:6]) %>% kable(caption = "White Wine Sample Data (Columns: 1-6)",
    booktabs = TRUE, longtable = TRUE)
head(d2.r[, 7:12]) %>% kable(caption = "White Wine Sample Data (Columns: 7-12)",
    booktabs = TRUE, longtable = TRUE)
var <- colnames(d1.r)[-ncol(d1.r)]
num <- 1:length(var)
```

```
df <- data.frame(S.No. = num, Predictor = var)

kable(df, caption = "Predictor Variables", booktabs = TRUE,
    longtable = TRUE)

multi.hist(d1.r, main = NA, dcol = c("blue", "red"),
    dlty = c("solid", "solid"), bcol = "linen")
d1.b <- d1.r
d1.b[, 12] <- as.factor(d1.b[, 12])
lh <- c(4, 5, 7, 10)
bw <- c(0.05, 0.003, 0.6, 0.01)
for (i in 1:4) {
    base <- ggplot(d1.b)
    layering.h <- geom_histogram(aes(x = d1.b[,
        lh[i]], fill = Quality), binwidth = bw[i])
    layering.l <- geom_vline(aes(xintercept = mean(d1.b[,
        lh[i]])), color = "indianred4")
    theming.x <- xlab(colnames(d1.b)[lh[i]])
    theming.y <- ylab("Frequency")
    theming.t <- ggtitle(paste("Histogram and Mean of ",
        colnames(d1.b)[lh[i]], ""))
    themed <- base + layering.h + layering.l +
        theming.x + theming.y + theming.t
    print(themed)
}

qboxp <- function(df) {
    d1.b <- df
    d1.b[, 12] <- as.factor(d1.b[, 12])
    df.m <- melt(d1.b, id.var = "Quality")
    base <- ggplot(data = df.m, aes(x = variable,
        y = value))
    layering <- geom_boxplot(aes(fill = Quality),
        outlier.shape = 1, outlier.color = "indianred3")
    scaling <- scale_x_discrete(breaks = NULL)
    faceting <- facet_wrap(~variable, scales = "free")
    theming.x <- xlab("Predictor Variables")
    theming.y <- ylab("Physico-chemical Values")
    theming.title <- ggtitle("Quality Boxplots for Each Predictor")
    theming.facet <- theme(strip.text = element_text(size = 7),
        axis.text.x = element_blank(), axis.ticks.x = element_blank())
    themed <- base + layering + scaling + faceting +
        theming.x + theming.y + theming.title +
        theming.facet
    themed
}
qboxp(d1.r)

outtona <- function(vl, df) {
    for (i in vl) {
        outlier <- boxplot.stats(df[, i])$out
        df[, i] <- ifelse(df[, i] %in% outlier,
            NA, df[, i])
```

```
    }
    return(df)
}

outliereff <- function(varlist, df) {
    for (i in varlist) {
        total = length(df[, i])
        par(mfrow = c(2, 2), oma = c(0, 0, 3,
            0))
        boxplot(df[, i], main = "With Outliers")
        hist(df[, i], main = "With Outliers",
            xlab = NA, ylab = NA, prob = TRUE)
        df <- outtona(i, df)
        boxplot(df[, i], main = "Without outliers")
        hist(df[, i], main = "Without outliers",
            xlab = NA, ylab = NA, prob = TRUE)
        out <- sum(is.na(df[, i]))
        per <- round((out)/total * 100, 1)
        title(paste("Effect of", out, "(", per,
            "%)", "Outliers on", colnames(df)[i],
            sep = " "), outer = TRUE)
    }
}
outliereff(4, d1.r)
d1.on <- outtona(1:11, d1.r)
d1.f <- d1.on[complete.cases(d1.on), ]
d2.on <- outtona(1:11, d2.r)
d2.f <- d2.on[complete.cases(d2.on), ]
qboxp(d1.f)
qboxp(d2.f)
imppred <- randomForest(quality ~ ., data = d1,
    ntree = 100, keep.forest = FALSE, importance = TRUE)
importance(imppred, type = 1)
imppred <- randomForest(quality ~ ., data = d2,
    ntree = 100, keep.forest = FALSE, importance = TRUE)
importance(imppred, type = 1)
symnum(cor(d1.r))
symnum(cor(d2.r))
lrmodel.red <- lm(Quality ~ ., data = d1.r)
summary(lrmodel.red)
kable(anova(lrmodel.red), booktabs = T)
lrmodel.red2 <- update(lrmodel.red, . ~ . - `Citric Acid` -
    `Residual Sugar`)
summary(lrmodel.red2)
rtmodel.red <- rpart(Quality ~ ., data = d1.r)
plot(rtmodel.red, uniform = T, branch = 1, margin = 0.05,
    cex = 0.9)
text(rtmodel.red, cex = 0.7)
lrm.pred.red <- predict(lrmodel.red, d1.r)
rt.pred.red <- predict(rtmodel.red, d1.r)
mae.lrm.red <- mean(abs(lrm.pred.red - d1.r[,
    12]))
mae.rt.red <- mean(abs(rt.pred.red - d1.r[, 12]))
```

```
mae.lrm.red
mae.rt.red
(mse.lrm.red <- mean((lrm.pred.red - d1.r[, 12])^2))
(mse.rt.red <- mean((rt.pred.red - d1.r[, 12])^2))

multi.hist(d2.r, main = NA, dcol = c("blue", "red"),
    dlty = c("solid", "solid"), bcol = "grey95")
qboxp(d2.r)
outliereff(5, d1.r)
outliereff(10, d1.r)
outliereff(1, d2.r)
outliereff(5, d2.r)
outliereff(6, d2.r)
outliereff(8, d2.r)
outliereff(10, d2.r)
```

# References

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 6. Springer.

Wickham, Hadley. 2009. *Ggplot2: Elegant Graphics for Data Analysis*. Springer Science & Business Media.