Wine Alcohol Linear Regression Analysis

Department of Mathematics and Statistics

San Jose State University

By

Qian Meng

December, 2020

**INTRODUCTION**

The purpose of this project is to apply the theories and skills learned in Math 261 course to specify a linear regression model based on a multivariate data set.

The data set used in the analysis is related to variants of the Portuguese "Vinho Verde" wine (hereinafter referred to as 'Wine' ). The raw data is collected from the UCI Machine Learning Repository [1]. We combined two 12-variable data sets, one for red wine and one for white wine, to obtain our data set, ‘Wine'. When coding for wine color, we obtained a total of 13 variables (wine color, fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality) and 6,497 observations. The variable ‘wine.color’, was added after we combined the two original data sets. There are two discrete variables: wine color and quality. The remaining 11 variables are continuous.

On the first model fitting trial, 'quality' was set as the response, and the other 12 variables as predictors. Soon we noted that the variation explanatory power of the model was not desirable ($R^2$ was less than 0.3), even though the model produced satisfactory residual plots. This is probably ascribed to the response’s discreteness. Thus, a continuous variable ‘alcohol’ was selected as the new response. The procedure on how we approached the final true model and the performance of the model will be explained later.

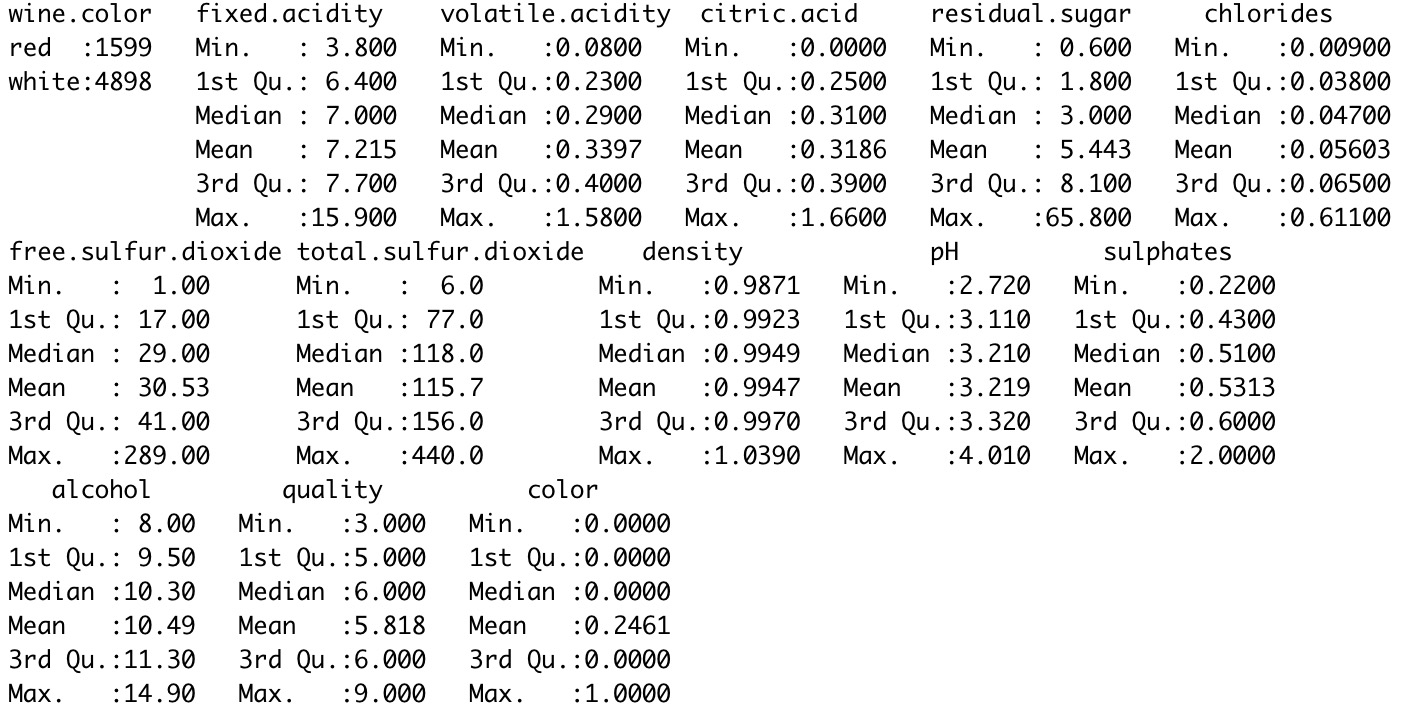
The analysis covered the areas: data exploration and visualization, multicollinearity, variable selection, model fitting, transformations, assumptions checking, and diagnostics for leverage and influence. The detailed explanation will be given in later sections. At the end, we present a final discussion with directions for future work.

**METHODS**

Data exploration and visualization

The data set ‘Wine' contains 11 continuous variables and 2 discrete variables (‘‘wine.color’’ and 'quality'). The value for (wine) ‘color’’ is 0-1, 0 is for white wine, and 1 is for red wine. The range of 'quality' is 3 - 9, 3 is for the lowest quality, and 9 is for the highest quality. Among all $6,497$ observations, there are $1,599$ observations of red wine and the other $4,898$ are white wine.

INSERT DATA SUMMARY HERE



By checking the distribution of all the variables, some of the data appear to be a little bit right skewness, which we determine by comparing the mean to the median. (Mean > median implies skewed right, while mean < median implies skewed left.) However, the quantitative assessment we tested in R, suggests none of these skewness is significant or needed to be removed by model transformation.

The boxplot and histogram of all predictors were plotted to explore the data, as well as a scatterplot of the response against each predictor. According to the boxplots, almost all the continuous variables have outliers, and most outliers scatter above the upper boundary of the boxes. This is in line with the previous 'right skewness' finding. We noticed the data point No.4381 as an outlier and an influential point based on our model, which will be discussed later. We created boxplots, histograms, and scatterplots between the response and each predictor in two steps: first, we plotted the entire data set. After plotting the entire data set, we plotted the data set according to wine color. We performed two passes of plotting to determine if our exploratory data analysis could reveal potential interactions with wine color.

Multicollinearity Diagnosis

The correlation scores are computed based on correlation matrix. The highest correlation is between total sulfur dioxide and wine color, which is -0.7. As expected, the Variance Inflation Factors (VIFs) of the variables are low, most are less than 3. The highest VIF is from the discrete variable wine color, which is 6.8. These results indicate that we do not have severe multicollinearity among the variables. Thus, prior to variable selection, we have the model:

$$

\begin{align\*}

\hat{y} &= \beta\_{0} + \beta\_1 x\_{fixed acidity} + \beta\_2 x\_{volatile acidity} + \beta\_3 x\_{citric acid} + \beta\_4 x\_{residual sugar} + \beta\_5 x\_{chlorides} + \beta\_6 x\_{free sulfur dioxide} + \beta\_7 x\_{total sulfur dioxide} + \beta\_8 x\_{density} + \beta\_9 x\_{pH} + \beta\_{10} x\_{sulphates} + \beta\_{11} x\_{quality4} + \beta\_{12} x\_{quality5} + \beta\_{13} x\_{quality6} + \beta\_{14} x\_{quality7} + \beta\_{15} x\_{quality8} + \beta\_{16} x\_{quality9} + \beta\_{17} x\_{red}

\end{align\*}

$$

Variable selection

We were limited by a lack of understanding of the chemical properties of wine, so it was difficult for us to employ subject matter expertise to select the most important variables from the 12 predictors intuitively. Thus, we did variable selection by using ```leaps()``` function in R with the entire scaled wine data set. The leaps function searches the best subsets of the variables in predictors for response in linear regression, using an efficient branch-and-bound algorithm.[2]

The variable Alcohol is set as the response against the other 12 variables as predictors in the full model. Mallow's Cp statistic is selected as the method in the leaps function, and the algorithm seeks for a small value of $C\_p$. The smaller the value of $C\_p$, the more precise the model is, equivalent to the better model involving a subset of given predictors.[3] Note that by using the 'Cp' method, the recommended model is the model that eliminates total sulfur dioxide.

Transformation

In this session, we mainly focused on the multiple linear regression model. Similarly, 'Alcohol' variable is set as the response and all the other 12 variables as predictors. The 'forward selection' and 'backward elimination' model building procedures are executed by using STEP function in R with the whole scaled 'Wine' data set. The step function performs to select formula-based model by its AIC.[2]

We set the 'direction' argument in the function as ‘backward’, ‘forward’, and ‘both’. For the base model, we used a simple linear regression model with alcohol regressed on wine color, due to the requirement of the model to contain at least one categorical variable. The recommended model from all the 3 trials turned to be the same: one that excludes ‘total.sulfur.dioxide’. This result is in agreement with the result from the Mallow's $C\_p$ statistic.

Model fitting

We fit the model based on unscaled data points, and 2 discrete variables are treated as factors. The summary statistics of the linear model indicates the model fitness is not bad, referring to the MSres is 0.25, p value of F-statistic is less than 2.2e-16 (significant), and Adjusted R-squared is 0.82. The high R-squared indicates that the model can explain more than 82% of the responses based on the given predictors.

As expected, t-statistic of predictors 'total.sulfur.dioxide' is not significant. This is consistent with the results from the 'Mallow's Cp statistic' and ‘backward/forward’ transformation. The indicator variable 'quality' has 7 different levels (3-9). The model summary shows only 2 (quality = 7,8) out of the 6 coefficients turns to be significant at level 0.01 and 0.001 correspondingly. Referring to the description of the 'Wine' data set, there are many more normal wines (quality = 5,6) than excellent (quality = 7-9 )or poor ones(quality =3-4). It is sensible to remove predictors 'total.sulfur.dioxide' and 'quality' from the model.

INSERT INIMODEL SUMMARY HERE

After total sulfur dioxide and quality are eliminated, the interaction between the left indicator variables ‘color’ and other predictors are tested. According to the scatterplots (alcohol vs. predictor), the data points of red wine and white wine are distributed considerably differently when ‘fixed.acidity’ and ‘volatile.acidity’. (In the scatter plot, the red pots are for red wine, black pots are for white wine.)

INSERT SCATTER PLOTS HERE

Even if the model doesn’t improve greatly by adding the interaction items (color:fixed.acidity and color:volatile.acidity), the interaction items are left in the final models since we can tell the influence of ‘color’ on ‘fixed.acidity’ and ‘volatile.acidity’. As such, we obtain the final model:

$$

\begin{align\*}

\hat{y} &= \beta\_{0} + \beta\_1 x\_{fixed acidity} + \beta\_2 x\_{volatile acidity} + \beta\_3 x\_{residual sugar} + \beta\_4 x\_{chlorides} + \beta\_5 x\_{free sulfur dioxide} + \beta\_6 x\_{density} + \beta\_7 x\_{pH} + \beta\_8 x\_{sulphates} + \beta\_9 x\_{red} + \beta\_{10} x\_{fixed acidity:red} + \beta\_{11} x\_{volatile acidity:red}

\end{align\*}

$$

**RESULTS**

For the linear regression model concerning wine alcohol content, we require the basic assumptions of linear regression to hold. That is, we require that the relationship between the response variable and the predictor variables is linear, the error term $\vec{\epsilon}}$ has $\text{E}(\vec{\epsilon}}) = 0$, the error term $\vec{\epsilon}}$ has constant variance (homoscedastic), the errors are uncorrelated, and inference additionally requires that the errors are normally and identically distributed. To check our assumptions, we constructed scatterplots to ensure the relationship between the response and each predictor is adequately modeled by a line. We make use of the native ```plot()``` function in R to verify that the residuals are approximately normally distributed and that the variance is approximately constant. We begin with the Raw Residuals versus Fitted Values plot:

INSERT RESIDUAL PLOT HERE

In the above plot, we notice immediately that there is one observation, point 4381, with an abnormally large raw residual and a negative fitted value. We will continue to monitor this point throughout the analysis, particularly when we discuss the leverage and influence of the model, the coefficients, and the fitted values. Regarding the remainder of the points, we see that the expected value remains around zero and that the constant variance assumption is satisfied. We also see that there is no clear curvature left in the residuals, which indicates that we do not require higher order terms or transformations. Based on our observations, we determined that the constant variance assumption is satisfied. Next, we examine the normality assumption of the residuals.

INSERT QQPLOT HERE

Using the qqplot to assess the normality of the residuals, we see that a majority of the points presented in this plot remain near the qqline. As we observed in the raw residuals plot, we observe that the point 4381 is quite a bit farther from the remainder of the points, which serves to obscure the right side of the plot. Upon closer inspection, we see this plot shows heavy tails compared to the normal distribution. As such, we may consider transformations if we are required to perform inference with this model. Next, we will discuss measures of diagnostics regarding leverage and influence of the model.

Regarding the measures of diagnostics for leverage and influence, we will examine the hat values, the R-Student residuals, Cook’s Distance, DFBETAS, and DFFITS for the model. First, we will review the leverage of the points in the model.

INSERT LEVERAGE PLOT HERE

With a sample size of $n = 6,497$ and $p = 12$ parameters in the model, the commonly accepted threshold for assessing leverage $h\_{ii} \gt \dfrac{2p}{n}$ is fairly small, with $\dfrac{2p}{n} = `r round( 2 \* p / n, 4)`$. We determined that that $`r lev.cnt`$ points exceed this threshold, or approximately $`r round( lev.cnt / n, 4 ) \* 100`$% of the observations are categorized as leverage points by the definition from class. Next, we will investigate a plot of the R-Student residuals on the vertical axis against the hat values.

INSERT R-STUDENT VS HATVALUES PLOT HERE

In the above plot, we can see there are quite a few points considered leverage points, by being located to the right of the vertical red line. We also observe that point $4381$ contains the most leverage and the largest R-Student residual, $t\_{4381} = `r max(rstudent(wine.final))`$. As for other fitted values, we observe that points with R-Student residuals greater than $3$ possess much smaller hat values, indicating less leverage. Overall, we see that most points fall into an acceptable region in the above plot. We will now examine Cook’s Distance to assess the combination of leverage and influence for each observation in the data set. Cook’s Distance is a measure of the difference in the regression coefficients and fitted values from the model using all data points and the model with the $i^{th}$ response removed:

INSERT COOK’S DISTANCE PLOT HERE

Upon reviewing the Cook’s Distance plot, we readily see point 4381 possesses a very large Cook’s Distance that compresses the vertical space on the graph for the other points. We see that point 4381 has a Cook’s Distance larger than $5$, where a Cook’s Distance greater than $1$ is considered to be a threshold for identifying outliers in a model. We observe that the remainder of the data is well below the threshold for determining outliers. We continue our analysis by examining $DFBETAS$ next.

We include all of the $DFBETAS$ plots in Appendix A. To summarize our findings, we constructed the following table that shows each $DFBETAS\_{j}$ for $j = 0, 1, \cdots, 11$ and the number of points in our data set that are considered to be influential to the calculation of $\beta\_{j}$.

INSERT DFBETAS TABLE HERE

In the above table, we can see that each parameter had a low frequency of points that were considered influential relative to the entire data set. For each point, we recorded the number of parameters it was influential in calculating, to serve for further investigation for potential recording errors. We note that two points, $4381$ and $5501$ were considered to be influential in calculating each model parameter. Next, we calculated the $DFFITS$ for the model.

INSERT DFFITS PLOTS HERE

We collected all points that exceeded the threshold used for testing how influential the $i^{th}$ observation is in the calculation of $\hat{y}\_i$, given by $|DFFITS\_i| > 2\sqrt{\dfrac{p}{n}}$. On the left plot, we notice that point 4381 obscures the view of the remaining points by vertically compressing the graph. On the right, with point 4381 excluded, we can more easily see points the collection of points that exceed the threshold. We note that points with lower index values appear to be more influential in the calculation of the lower indexed fitted values. This is interesting because the red wines are listed at lower indices, and white wines at higher indices. We observe that white wines appear to exceed the $DFFITS$ threshold less often than their red counterparts, but those that do are more pronounced in the distance away from the horizontal line at zero. Next, we will review the $COVRATIO$ for the data set.

INSERT COVRATIO PLOTS HERE

In the above plot on the left, we see that the inclusion of point $4381$ greatly reduces the precision of the model. However, we note that there are three other points that degrade model precision as well: $3253$, $3263$, and $5501$. Removing point $4381$ from the view, we obtain the above plot on the right. What we notice in this plot is that there is a large cluster of points on the left-hand side that fall below the lower threshold. The first $1,599$ points in the data set belong to red wine, while the remaining $4,898$ points belong to white wine. What we observe is that the model is trained largely to fit white wine, and the fitting of red wine appears to affect precision by including it in the same data set as the white wine. In general, we see more spread for the points on the left side of the axis on either $COVRATIO$ plot. Overall, we note that $200$ points in the data set improve model precision by including them in the data set. On the other hand, we also note that $121$ points in the data set degrade model precision by including them in the data set.

Based on these analyses for detecting outliers and influential points, we have collected all points that could be considered outliers. As such, we could review the points to determine whether any of them were true recording errors, and simultaneously improve the fitting of the model. However, while this may improve the fit of the model, any potential tests may be hindered by the refining of the data. Next, we will discuss the interpretation of the model coefficients.

Below, we provide the model summaries for both the unscaled and the scaled models. Following the summaries, we interpret the model coefficients.

INSERT SUMMARIES HERE

For the unscaled model, we have the following coefficient interpretations:

Fixed.acidity: When fixed acidity increases by one unit, alcohol content increases by approximately $.5461$%, on average, when all other variables are held constant.

Volatile.acidity: When volatile acidity increases by one unit, alcohol content increases by approximately $.8376$%, on average, when all other variables are held constant.

Residual.sugar: When residual sugar increases by one unit, alcohol content increases by approximately $.2407$%, on average, when all other variables are held constant.

Chlorides: When chlorides increase by one unit, alcohol content decreases by approximately $.6795$%, on average, when all other variables are held constant.

Free.sulfur.dioxide: When free sulfur dioxide increases by one unit, alcohol content decreases by approximately $.0029$%, on average, when all other variables are held constant.

Density: When density increases by one unit, alcohol content decreases by approximately $683.4$%, on average, when all other variables are held constant.

pH: When pH increases by one unit, alcohol content increases by approximately $2.707$%, on average, when all other variables are held constant.

Sulphates: When sulphates increase by one unit, alcohol content increases by approximately $1.087$%, on average, when all other variables are held constant.

Color: Alcohol content for red wine is approximately $1.219$% higher than alcohol content for white wine, on average, when all other variables are equal and the only difference is wine color.

Fixed.acidity:color: Among red wines, when fixed acidity increases by one unit, alcohol content increases by approximately $.0356$%, on average, when all other variables are held constant.

Volatile.acidity:color: Among red wines, when volatile acidity increases by one unit, alcohol content decreases by approximately $.7081$%, on average, when all other variables are held constant.

Most of the coefficients in the model have a reasonable interpretation. The exception is the coefficient for density, which is too high to realistically have the effect that our model states. This is due to there being a low variance for most observations of the density predictor. We note that the observation that appeared on every influence diagnostic plot, 4381, has a density that was considerably different from the other $6,496$ observations. In the below figure, we show how much point 4381 varied in density from the other points.

INSERT DENSITY SCATTERPLOT HERE

Since the above plot shows that the value for density can create a large residual that shows up in each diagnostic plot for leverage and influence, we created a scaled data set to test whether the same issue occurred. In our scaled data set, we performed unit normal scaling for the response and each predictor. For the scaled model, we have the following coefficient interpretations:

Fixed.acidity: When fixed acidity increases by one standard deviation, the mean standardized response increases by approximately $.5936$ when all other variables are held constant.

Volatile.acidity: When volatile acidity increases by one standard deviation, the mean standardized response increases by approximately $.1156$ when all other variables are held constant.

Residual.sugar: When residual sugar increases by one standard deviation, the mean standardized response increases by approximately $.9602$ when all other variables are held constant.

Chlorides: When chlorides increase by one standard deviation, the mean standardized response decreases by approximately $.0200$ when all other variables are held constant.

Free.sulfur.dioxide: When free sulfur dioxide increases by one standard deviation, the mean standardized response decreases by approximately $.0436$ when all other variables are held constant.

Density: When density increases by one standard deviation, the mean standardized response decreases by approximately $1.7182$ when all other variables are held constant.

pH: When pH increases by one standard deviation, the mean standardized response increases by approximately $.3649$ when all other variables are held constant.

Sulphates: When sulphates increase by one standard deviation, the mean standardized response increases by approximately $.1357$ when all other variables are held constant.

Color: The standardized mean response for red wine is approximately $1.0356$ higher than for white wine, when all other variables are held constant and only differ by wine color.

Fixed.acidity:color: Among red wines, when fixed acidity increases by one standard deviation, the mean standardized response increases by approximately $.0387$ when all other variables are held constant.

Volatile.acidity:color: Among red wines, when volatile acidity increases by one standard deviation, the mean standardized response decreases by approximately $.0977$ when all other variables are held constant.

Comparing the magnitude of the coefficient estimates between the scaled and unscaled models, we notice that the scaled model can serve to inform future data collection methods and research, in particular as it relates to the density and residual sugar content of the wines. It appears that residual sugar, density, color, and both fixed and volatile acidity play a crucial role in determining alcohol content. We also note that these variables will likely differ between red and white wines, illustrated in the scatterplots color-coded by wine color, located in Appendix A.

Overall, we regard our preference for the unscaled model, as its units are more readily understandable to a general audience. The fact that the model abysmally predicts a small number of points should not be a cause for concern, but rather an opportunity to collect a broader data set.

**CONCLUSION/DISCUSSION**

Our work began with modeling wine quality, and due to modeling complications and lack of achieving a satisfying fit to the data, we opted to model wine alcohol content. We explored scatterplots of the predictors and identified interaction terms to address the potential differences in physicochemical properties between red wine and white wine. We will address some final thoughts regarding our work.

One potential future direction for this project is to use the split data sets for red and white wine and observe how well the final model fits the two data sets separately. This idea stems from the fact that the left-side of the $COVRATIO$ plots, which represented red wines in the data set, appeared to have a higher frequency of points that exceeded thresholds for both enhancing and degrading precision. One wonders if this is due to the model being accountable for both colors of wine.

Based on the work done regarding potential influential points, it would be interesting to see how the model statistics may differ with the removal of certain points according to a specified criteria such as removing points that exceed the interquartile range multiplied by a scalar. For example, we have discussed the considerable influence that point $4381$ exerted on the model. Out of curiosity, we examined the model fit with the specified point omitted. We observed a considerable increase in $R\_{adj}^2$, from $.8208$ to $.8416$, without changing the specification of the model. It would be interesting to have a model that performs even better for wines that do not possess extreme values of density and residual sugar.

Similarly, we could try a different direction, where we instead consider improving the fit of the model by collecting more data. Point $4381$ represents an interesting observation, being more dense and containing more residual sugar than the other points. Perhaps the reason the model does not fit particularly well is due to a lack of sampling wines with high density and high residual sugar. We could gather a larger sample size and pay special attention to these variables, ensuring that we have an adequate representation of various wines. Our result would be a more robust model that could accurately determine the alcohol content for many types of wines.

**Bibliography?**

[1] URL of Data Set [**https://archive.ics.uci.edu/ml/datasets/Wine+Quality**](https://archive.ics.uci.edu/ml/datasets/Wine+Quality)

[2] R documents[**https://www.rdocumentation.org/**](https://www.rdocumentation.org/)

[3] Definition of Mallow's Cp [**https://en.wikipedia.org/wiki/Mallows%27s\_Cp**](https://en.wikipedia.org/wiki/Mallows%27s_Cp)

**APPENDIX A: Plots**

For plots not already included in the paper

**APPENDIX B: R-Code**

**Model Code Section**

---

title: 'Wine Alcohol Linear Regression Analysis'

author: "Qian Meng & Louis Mutter"

date: "12/9/2020"

output:

word\_document: default

html\_document: default

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

# Project 2: Wine Quality Data

# By Qian Meng and Louis Mutter

# install.packages("leaps") # Run this line of code once

library(leaps) # used for variable selection leaps() function

library(dplyr) # used for arrange() function

# Set-up

# setwd() # Qian working directory

setwd("C:/Users/Louis/Desktop/School/Graduate/Math 261A/Project 2") # Louis working directory

wine <- read.csv("winequality.csv")

wine$color <- ifelse(wine$wine.color == "red", 1, 0)

y.wine <- wine[,12]

X.wine <- wine[, c(-1, -12)]

winedata <- as.data.frame(cbind(y.wine, X.wine))

names(winedata)[1] <- "alcohol"

```

## Abstract

The purpose of this project is to apply the theories and skills learned in Math 261A course to specify a linear regression model based on a multivariate data set. We fit a multiple linear regression model on $6,497$ observations of 12 variables. We seek to develop a robust multiple linear regression model using the following techniques: data exploration and visualization, multicollinearity diagnosis, variable selection, model fitting, assumptions checking, and diagnostics for leverage and influence. Exploratory data analysis informs the decision to include additional predictors as interaction terms, which improves explanatory power. At the end of this submission, we discuss and suggest potential future directions for the analysis to construct an even more robust or specialized regression model.

## Problem Description / Data Set Information

The data set used in the analysis is related to variants of the Portuguese "Vinho Verde" wine (hereinafter referred to as 'Wine'). The raw data was collected from the UCI Machine Learning Repository [1]. We combined two 12-variable data sets, one for red wine and one for white wine, to obtain our data set, ‘Wine'. When coding for wine color, we obtained a total of 13 variables (wine color, fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality) and 6,497 observations. The variable wine color, was added after we combined the two original data sets. There are two discrete variables: wine color and quality. The remaining 11 variables are continuous.

On the first model fitting trial, quality was set as the response, and the other 12 variables as predictors. Soon we noted that the explanatory power of the model was not desirable ($R^2$ was less than $0.3$), even though the model produced adequate residual plots. This is probably ascribed to the response’s discreteness. Thus, a continuous variable ‘alcohol’ was selected as the new response. The procedure on how we approached the final true model and the performance of the model will be explained later.

## Exploratory Data Analysis

The Wine data set contains 11 continuous variables and 2 discrete variables (wine color and quality). The value for wine color is 0 for white wine, and 1 for red wine. The range of quality is 3 - 9, 3 is for the lowest quality, and 9 is for the highest quality. Among all $6,497$ observations, there are $1,599$ observations of red wine and the other $4,898$ are white wine.

```{r}

summary(winedata)

```

By checking the distribution of all the variables, some of the data appear to be a bit right-skewed, which we determine by comparing the mean to the median. (Mean > median implies skewed right, while mean < median implies skewed left.) However, the quantitative assessment we tested in R suggests none of these skews are significant or need to be removed by variable transformation.

The boxplot and histogram of all predictors were plotted to explore the data, as well as a scatterplot of the response against each predictor (provided in Appendix A). According to the boxplots, almost all the continuous variables have outliers, and most outliers are located above the upper boundary of the boxes. This is in line with the previous right-skewed finding. We noticed the data point $4381$ is an outlier and an influential point based on our model, which will be discussed later. We created boxplots, histograms, and scatterplots between the response and each predictor in two steps: first, we plotted the entire data set, disregarding wine color. After plotting the entire data set, we plotted the data set according to wine color. We performed two passes of plotting to determine if our exploratory data analysis could reveal potential interactions with wine color.

## Multicollinearity Diagnosis

```{r, echo = F}

C <- round(cor(X.wine),2)

C

```

The correlation scores are obtained from the above correlation matrix. The highest correlation is between total sulfur dioxide and free sulfur dioxide, which is 0.72. The other correlations of concern are between volatile acidity and color (0.65) and total sulfur dioxide and color (-0.7). We pay special attention to these variables when reviewing the Variance Inflation Factors. The Variance Inflation Factors (VIFs) of the variables are low, most are less than 3. The high VIFs to take note of are: residual sugar, density, and color. Conceptually, these variables having high VIFs makes sense, as sugar is related to density (sugar is more dense than water), and wine color is related to sugar. However, the magnitude of the VIFs obtained are not a cause for concern, because these results indicate that we do not have severe multicollinearity among the variables. We list the VIFs in the table below:

$$

\begin{align\*}

\begin{array}{cc}

\text{Variable} & \text{Variance Inflation Factor} \\

\text{fixed acidity} & `r round(diag(solve(C))[1],4)` \\

\text{volatile acidity} & `r round(diag(solve(C))[2],4)` \\

\text{citric acid} & `r round(diag(solve(C))[3],4)` \\

\text{residual sugar} & `r round(diag(solve(C))[4],4)` \\

\text{chlorides} & `r round(diag(solve(C))[5],4)` \\

\text{free sulfur dioxide} & `r round(diag(solve(C))[6],4)` \\

\text{total sulfur dioxide} & `r round(diag(solve(C))[7],4)` \\

\text{density} & `r round(diag(solve(C))[8],4)` \\

\text{pH} & `r round(diag(solve(C))[9],4)` \\

\text{sulphates} & `r round(diag(solve(C))[10],4)` \\

\text{quality} & `r round(diag(solve(C))[11],4)` \\

\text{color} & `r round(diag(solve(C))[12],4)`

\end{array}

\end{align\*}

$$

Thus, prior to variable selection, we have the model:

$$

\begin{align\*}

y &= \beta\_{0} + \beta\_1 x\_{fixed.acidity} + \beta\_2 x\_{volatile.acidity} + \beta\_3 x\_{citric.acid} + \beta\_4 x\_{residual.sugar} +\\

&\beta\_5 x\_{chlorides} + \beta\_6 x\_{free.sulfur.dioxide} + \beta\_7 x\_{total.sulfur.dioxide} + \beta\_8 x\_{density} + \\

&\beta\_9 x\_{pH} + \beta\_{10} x\_{sulphates} + \beta\_{11} x\_{quality-4} + \beta\_{12} x\_{quality-5} + \beta\_{13} x\_{quality-6} +\\

&\beta\_{14} x\_{quality-7} + \beta\_{15} x\_{quality-8} + \beta\_{16} x\_{quality-9} + \beta\_{17} x\_{color-red} + \epsilon

\end{align\*}

$$

## Variable Selection

We were limited by a lack of understanding of the chemical properties of wine, so it was difficult for us to employ subject matter expertise to select the most important variables from the 12 predictors intuitively. This motivated the creation of the full model. Our full model contained many variables, so we performed variable selection by using the ```leaps()``` function in R on the data set. The leaps function searches the best subsets of the variables in predictors for response in linear regression, using an efficient branch-and-bound algorithm.

The variable alcohol is set as the response against the other 12 variables as predictors in the full model. Mallow's $C\_p$ statistic is selected as the method in the ```leaps``` function, and the algorithm seeks for a small value of $C\_p$. The smaller the value of $C\_p$, the more precise the model is, equivalent to the better model involving a subset of given predictors. We note that using the ```Cp``` method, the recommended model is the model that eliminates total sulfur dioxide.

The forward selection and backward elimination variable selection procedures are executed using the ```step``` function in R with scaled wine data set. We set the ```step``` function to select our regression model using the Bayesian Information Criterion. We set the ```direction``` argument in the function as ```backward```, ```forward```, and ```both```. For the base model, we used a simple linear regression model with alcohol regressed on wine color, due to the requirement of the model having to contain at least one categorical variable. The recommended model from each ```direction``` yielded the same model recommendation: one that excludes total sulfur dioxide. This result is in agreement with the result from the Mallow's $C\_p$ statistic.

We fit our final model based on the unscaled data points, with the two discrete variables treated with indicator variables. The summary statistics of the linear model indicates the model fit is not bad, with $MS\_{Res} = 0.25$, the p-value of the $F$-statistic is less than 2.2e-16 (significant), and $R\_{adj}^2 = 0.8227$. The high $R^2$ indicates that the model can explain more than 82% of the responses based on the given predictors.

When we reviewed the summary of the full model, the $t$-statistic of total sulfur dioxide is not significant. This is consistent with the results from the Mallow's $C\_p$ statistic and with all iterations of the ```step``` function. The indicator variable 'quality' has 7 distinct levels (3-9). The model summary showed only 2 (quality = 7,8) out of the 6 coefficients for quality as significant at level 0.01 and 0.001, respectively. Thus, we found it sensible to remove the predictors total sulfur dioxide and quality from the model.

After total sulfur dioxide and quality are eliminated, the interaction between the remaining indicator variable ```color-red``` with other predictors was tested. According to the scatterplots, the data points of red wine and white wine are distributed considerably differently in the fixed acidity and volatile acidity.

```{r, echo = F}

plot(wine$fixed.acidity, wine$alcohol, pch = 20, xlim = c(2,20),

xlab = "Fixed Acidity", ylab = "Alcohol", main = "Alcohol vs. Fixed Acidity Scatterplot",

col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

legend("bottomright", pch = 20, c("Red Wine","White Wine"), col=c("red","black"))

plot(wine$volatile.acidity, wine$alcohol, pch = 20, xlim = c(0,2),

xlab = "Volatile Acidity", ylab = "Alcohol", main = "Alcohol vs. Volatile Acidity Scatterplot",

col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

legend("bottomright", pch = 20, c("Red Wine","White Wine"), col=c("red","black"))

```

Even if the model does not improve greatly by adding the interaction ```color-red:fixed.acidity``` and ```color-red:volatile.acidity``` ($R\_{adj}^2 = .8227$ with interaction terms versus $R\_{adj}^2 = .8217$ without the interactions), the interaction items are left in the final models since we can tell the influence of color on fixed acidity and volatile acidity. As such, we obtain the final model:

$$

\begin{align\*}

\hat{y} &= \hat{\beta}\_{0} + \hat{\beta}\_1 x\_{fixed.acidity} + \hat{\beta}\_2 x\_{volatile.acidity} + \hat{\beta}\_3 x\_{citric.acid} + \hat{\beta}\_4 x\_{residual.sugar} +\\

&\hat{\beta}\_5 x\_{chlorides} +\hat{\beta}\_6 x\_{free.sulfur.dioxide} + \hat{\beta}\_7 x\_{density} + \hat{\beta}\_8 x\_{pH} + \hat{\beta}\_9 x\_{sulphates} + \hat{\beta}\_{10}x\_{color-red} + \\

&\hat{\beta}\_{11} x\_{color-red:fixed.acidity} + \hat{\beta}\_{12} x\_{color-red:volatile.acidity}

\end{align\*}

$$

```{r, echo = F}

winedata.scaled <- scale( winedata )

winedata.scaled <- as.data.frame(winedata.scaled)

alcohol <- winedata.scaled$alcohol # response

design <- winedata.scaled[, -1]

mymodel <- leaps( x = design, y = alcohol, method = "Cp", names = names( design ) )

# CpDiff <- mymodel$Cp - mymodel$size

# arrange( data.frame( mymodel$size, mymodel$which, mymodel$Cp, CpDiff ),CpDiff )

n <- dim(wine)[1]

lm.full <- lm( alcohol ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar + chlorides + free.sulfur.dioxide +

total.sulfur.dioxide + density + pH + sulphates + as.factor(color), data = winedata )

lm.base <- lm( alcohol ~ as.factor(color), data = winedata.scaled )

lm.back <- step( lm.full, scope = list( lower = ~1, upper = lm.full ), direction = "backward", trace = F, k = log(n) )

lm.forward <- step( lm.base, scope = list( lower = ~1, upper = lm.full ), direction = "forward", trace = F, k = log(n) )

lm.step <- step( lm.base, scope = list( lower = ~1, upper = lm.full ), direction = "both", trace = F, k = log(n) )

wine.final <- lm( alcohol ~ fixed.acidity + volatile.acidity + residual.sugar + chlorides + free.sulfur.dioxide + citric.acid +

density + pH + sulphates + as.factor(color) + as.factor(color):fixed.acidity + as.factor(color):volatile.acidity

, data = winedata )

wine.final.scaled <- lm( alcohol ~ fixed.acidity + volatile.acidity + residual.sugar + chlorides + free.sulfur.dioxide + citric.acid +

density + pH + sulphates + as.factor(color) + as.factor(color):fixed.acidity + as.factor(color):volatile.acidity + 0

, data = winedata.scaled )

yhat <- fitted(wine.final)

e <- residuals(wine.final)

d <- e/summary(wine.final)$sigma

r <- rstandard(wine.final)

t <- rstudent(wine.final)

n <- dim(wine)[1]

p <- length(wine.final$coefficients)

```

```{r, include = F}

plot(wine.final, 1, pch = 20)

plot(wine.final, 2, pch = 20)

plot(yhat, t, pch = 20, xlab = "Fitted values", ylab = "R-Student Residuals", main = "R-Student Residuals vs. Fitted Values Wine Alcohol")

abline(h = 0, lty = 2, col = "grey")

abline(h = 3, lty = 2, col = "red")

abline(h = -3, lty = 2, col = "red")

plot(hatvalues(wine.final), pch = 20, ylab = "Hat Values", main = "Wine Alcohol Leverage")

abline( h = 2 \* (p / n), lty = 2, lwd = 2, col = "red" )

plot(hatvalues(wine.final), rstudent(wine.final), pch = 20, xlab = "Hat Values", ylab = "R-Student Residuals", main = "R-Student vs. Hat Values Wine Alcohol")

abline( v = 2 \* (p / n), lty = 2, lwd = 2, col = "red" )

abline( h = 3, lty = 2, lwd = 2, col = "red" )

abline( h = -3, lty = 2, lwd = 2, col = "red" )

plot(cooks.distance(wine.final), pch = 20, ylab = "Cook's Distance", main = "Cook's D.")

abline( h = 1, lty = 2, lwd = 2, col = "red")

```

```{r, include = F}

lev.pts <- which(ifelse(hatvalues(wine.final) > 2 \* (p / n), T, F))

lev.cnt <- 0

for (i in lev.pts) {lev.cnt <- lev.cnt + 1}

lev.cnt/n

# dfbetas(wine.final)

dfb0 <- which(ifelse(abs(dfbetas(wine.final)[,1]) > (2/sqrt(n)), T, F))

dfb0.cnt <- rep(0, n)

for (i in dfb0){ dfb0.cnt[i] = dfb0.cnt[i] + 1 }

names(dfb0.cnt) <- c(paste(1:n, sep = ", "))

dfb1 <- which(ifelse(abs(dfbetas(wine.final)[,2]) > (2/sqrt(n)), T, F))

dfb1.cnt <- rep(0, n)

for (i in dfb1){ dfb1.cnt[i] = dfb1.cnt[i] + 1 }

names(dfb1.cnt) <- c(paste(1:n, sep = ", "))

dfb2 <- which(ifelse(abs(dfbetas(wine.final)[,3]) > (2/sqrt(n)), T, F))

dfb2.cnt <- rep(0, n)

for (i in dfb2){ dfb2.cnt[i] = dfb2.cnt[i] + 1 }

names(dfb2.cnt) <- c(paste(1:n, sep = ", "))

dfb3 <- which(ifelse(abs(dfbetas(wine.final)[,4]) > (2/sqrt(n)), T, F))

dfb3.cnt <- rep(0, n)

for (i in dfb3){ dfb3.cnt[i] = dfb3.cnt[i] + 1 }

names(dfb3.cnt) <- c(paste(1:n, sep = ", "))

dfb4 <- which(ifelse(abs(dfbetas(wine.final)[,5]) > (2/sqrt(n)), T, F))

dfb4.cnt <- rep(0, n)

for (i in dfb4){ dfb4.cnt[i] = dfb4.cnt[i] + 1 }

names(dfb4.cnt) <- c(paste(1:n, sep = ", "))

dfb5 <- which(ifelse(abs(dfbetas(wine.final)[,6]) > (2/sqrt(n)), T, F))

dfb5.cnt <- rep(0, n)

for (i in dfb5){ dfb5.cnt[i] = dfb5.cnt[i] + 1 }

names(dfb5.cnt) <- c(paste(1:n, sep = ", "))

dfb6 <- which(ifelse(abs(dfbetas(wine.final)[,7]) > (2/sqrt(n)), T, F))

dfb6.cnt <- rep(0, n)

for (i in dfb6){ dfb6.cnt[i] = dfb6.cnt[i] + 1 }

names(dfb6.cnt) <- c(paste(1:n, sep = ", "))

dfb7 <- which(ifelse(abs(dfbetas(wine.final)[,8]) > (2/sqrt(n)), T, F))

dfb7.cnt <- rep(0, n)

for (i in dfb7){ dfb7.cnt[i] = dfb7.cnt[i] + 1 }

names(dfb7.cnt) <- c(paste(1:n, sep = ", "))

dfb8 <- which(ifelse(abs(dfbetas(wine.final)[,9]) > (2/sqrt(n)), T, F))

dfb8.cnt <- rep(0, n)

for (i in dfb8){ dfb8.cnt[i] = dfb8.cnt[i] + 1 }

names(dfb8.cnt) <- c(paste(1:n, sep = ", "))

dfb9 <- which(ifelse(abs(dfbetas(wine.final)[,10]) > (2/sqrt(n)), T, F))

dfb9.cnt <- rep(0, n)

for (i in dfb9){ dfb9.cnt[i] = dfb9.cnt[i] + 1 }

names(dfb9.cnt) <- c(paste(1:n, sep = ", "))

dfb10 <- which(ifelse(abs(dfbetas(wine.final)[,11]) > (2/sqrt(n)), T, F))

dfb10.cnt <- rep(0, n)

for (i in dfb10){ dfb10.cnt[i] = dfb10.cnt[i] + 1 }

names(dfb10.cnt) <- c(paste(1:n, sep = ", "))

dfb11 <- which(ifelse(abs(dfbetas(wine.final)[,12]) > (2/sqrt(n)), T, F))

dfb11.cnt <- rep(0, n)

for (i in dfb11){ dfb11.cnt[i] = dfb11.cnt[i] + 1 }

names(dfb11.cnt) <- c(paste(1:n, sep = ", "))

dfb12 <- which(ifelse(abs(dfbetas(wine.final)[,13]) > (2/sqrt(n)), T, F))

dfb12.cnt <- rep(0, n)

for (i in dfb12){ dfb12.cnt[i] = dfb12.cnt[i] + 1 }

names(dfb12.cnt) <- c(paste(1:n, sep = ", "))

dfb.pts <- c(dfb0, dfb1, dfb2, dfb3, dfb4, dfb5, dfb6, dfb7, dfb8, dfb9, dfb10, dfb11,dfb12)

# dfb0;dfb1;dfb2;dfb3;dfb4;dfb5;dfb6;dfb7;dfb8;dfb9;dfb10;dfb11;dfb12

dfb.cnt <- rep(0, n)

for (i in dfb.pts){ dfb.cnt[i] = dfb.cnt[i] + 1 }

names(dfb.cnt) <- c(paste(1:n, sep = ", "))

# names(sort(dfb.cnt, decreasing = T)[1:20])

# sort(dfb.cnt, decreasing = T)[1:20]

# dffits(wine.final)

par( mfrow = c(1,2) )

plot(dffits(wine.final), pch = 20, ylab = "DFFITS", main = "DFFITS")

abline( h = 2 \* sqrt( p / n ), lty = 2, lwd = 2, col = "red")

abline( h = -2 \* sqrt( p / n ), lty = 2, lwd = 2, col = "red")

plot(dffits(wine.final), pch = 20, ylim = c(-1.5,1.5), ylab = "DFFITS", main = "DFFITS (Pt 4381 Excluded)")

abline( h = 2 \* sqrt( p / n ), lty = 2, lwd = 2, col = "red")

abline( h = -2 \* sqrt( p / n ), lty = 2, lwd = 2, col = "red")

par( mfrow = c(1,1) )

dff <- which(ifelse(abs(dffits(wine.final)) > 2 \* sqrt( p / n ), T, F))

dff.cnt <- rep(0, n)

for(i in dff){ dff.cnt[i] <- dff.cnt[i] + 1}

names(dff.cnt) <- c(paste(1:n, sep = ", "))

# sort(dff.cnt, decreasing = T)[1:20]

sum(dff.cnt)

# dff.cnt[dff.cnt == 1]

# covratio(wine.final)

par( mfrow = c(1,2) )

plot(covratio(wine.final), pch = 20, ylab = "COVRATIO", main = "COVRATIO")

abline( h = 1 + 3 \* (p / n), lty = 2, lwd = 2, col = "red")

abline( h = 1 - 3 \* (p / n), lty = 2, lwd = 2, col = "red")

plot(covratio(wine.final), pch = 20, ylab = "COVRATIO", ylim = c(.9,1.1), main = "COVRATIO (Pt 4381 Excluded)")

abline( h = 1 + 3 \* (p / n), lty = 2, lwd = 2, col = "red")

abline( h = 1 - 3 \* (p / n), lty = 2, lwd = 2, col = "red")

par(mfrow = c(1,1) )

cov.help <- which(ifelse(covratio(wine.final) > 1 + 3 \* (p / n ), T, F))

cov.hurt <- which(ifelse(covratio(wine.final) < 1 - 3 \* (p / n ), T, F))

help.cnt <- rep(0,n)

for (i in cov.help) {help.cnt[i] <- help.cnt[i] + 1}

names(help.cnt) <- c(paste(1:n, sep = ", "))

hurt.cnt <- rep(0,n)

for (i in cov.hurt) {hurt.cnt[i] <- hurt.cnt[i] + 1}

names(hurt.cnt) <- c(paste(1:n, sep = ", "))

# hurt.cnt[hurt.cnt == 1]

```

For the linear regression model concerning wine alcohol content, we require the basic assumptions of linear regression to hold. That is, we require that the relationship between the response variable and the predictor variables is linear, the error term $\vec{\epsilon}$ has $\text{E}(\vec{\epsilon}) = 0$, the error term $\vec{\epsilon}$ has constant variance (homoscedastic), the errors are uncorrelated, and inference additionally requires that the errors are normally and identically distributed. To check our assumptions, we constructed scatterplots to ensure the relationship between the response and each predictor is adequately modeled by a line (included in Appendix A). We make use of the native ```plot()``` function in R to verify that the residuals are approximately normally distributed and that the variance is approximately constant. We begin with the Raw Residuals versus Fitted Values plot:

```{r, echo = F}

plot(wine.final, 1, pch = 20)

```

In the above plot, we notice immediately that there is one observation, point 4381, with an abnormally large raw residual and a negative fitted value. We will continue to monitor this point throughout the analysis, particularly when we discuss the leverage and influence of the model, the coefficients, and the fitted values. Regarding the remainder of the points, we see that the expected value remains around zero and that the constant variance assumption is satisfied. We also see that there is no clear curvature left in the residuals, which indicates that we do not require higher order terms or transformations. Based on our observations, we determined that the constant variance assumption is satisfied. Next, we examine the normality assumption of the residuals.

```{r, echo = F}

plot(wine.final, 2, pch = 20)

```

Using the qqplot to assess the normality of the residuals, we see that a majority of the points presented in this plot remain near the qqline. As we observed in the raw residuals plot, we observe that the point 4381 is quite a bit farther from the remainder of the points, which serves to obscure the right side of the plot. Upon closer inspection, we see this plot shows heavy tails compared to the normal distribution. As such, we may consider transformations if we are required to perform inference with this model. Next, we will discuss measures of diagnostics regarding leverage and influence of the model.

Regarding the measures of diagnostics for leverage and influence, we will examine the hat values, the R-Student residuals, Cook’s Distance, DFBETAS, and DFFITS for the model. First, we will review the leverage of the points in the model.

```{r, echo = F}

plot(hatvalues(wine.final), pch = 20, ylab = "Hat Values", main = "Wine Alcohol Leverage")

abline( h = 2 \* (p / n), lty = 2, lwd = 2, col = "red" )

```

With a sample size of $n = 6,497$ and $p = `r length(coefficients(wine.final))`$ parameters in the model, the commonly accepted threshold for assessing leverage $h\_{ii} \gt \dfrac{2p}{n}$ is fairly small, with $\dfrac{2p}{n} = `r round( 2 \* p / n, 4)`$. We determined that that $`r lev.cnt`$ points exceed this threshold, or approximately $`r round( lev.cnt / n, 4 ) \* 100`$% of the observations are categorized as leverage points by the definition from class. Next, we will investigate a plot of the $R$-Student residuals on the vertical axis against the hat values along the horizontal axis. The $R$-Student residuals can be a better residual type to use for examining constant variance by using deletion to determine the distance between the fitted value not fitted with the $i^{th}$ observation and observed value for $y\_i$ from the full data set.

```{r, echo = F}

plot(hatvalues(wine.final), rstudent(wine.final), pch = 20, xlab = "Hat Values", ylab = "R-Student Residuals", main = "R-Student vs. Hat Values Wine Alcohol")

abline( v = 2 \* (p / n), lty = 2, lwd = 2, col = "red" )

abline( h = 3, lty = 2, lwd = 2, col = "red" )

abline( h = -3, lty = 2, lwd = 2, col = "red" )

```

In the above plot, we can see there are quite a few points considered leverage points, by being located to the right of the vertical red line. We also observe that point $4381$ contains the most leverage and the largest R-Student residual, $t\_{4381} = `r round(max(rstudent(wine.final)),2)`$. As for other fitted values, we observe that points with R-Student residuals greater than $3$ possess much smaller hat values, indicating less leverage. Overall, we see that most points fall into an acceptable region in the above plot. We will now examine Cook’s Distance to assess the combination of leverage and influence for each observation in the data set. Cook’s Distance is a measure of the difference in the regression coefficients and fitted values from the model using all data points and the model with the $i^{th}$ response removed:

```{r, echo = F}

plot(cooks.distance(wine.final), pch = 20, ylab = "Cook's Distance", main = "Cook's D.")

abline( h = 1, lty = 2, lwd = 2, col = "red")

```

Upon reviewing the Cook’s Distance plot, we readily see point 4381 possesses a very large Cook’s Distance that compresses the vertical space on the graph for the other points. We see that point 4381 has a Cook’s Distance larger than $5$, where a Cook’s Distance greater than $1$ is considered to be a threshold for identifying outliers in a model. We observe that the remainder of the data is well below the threshold for determining outliers. We continue our analysis by examining $DFBETAS$ next.

We include all of the $DFBETAS$ plots in Appendix A. To summarize our findings, we constructed the following table that shows each $DFBETAS\_{j}$ for $j = 0, 1, \cdots, 12$ and the number of points in our data set that are considered to be influential to the calculation of $\beta\_{j}$.

$$

\begin{align\*}

\begin{array}{cc}

DFBETAS\_j & \text{Frequency of Points of Interest} \\

j = 0 & `r sum(dfb0.cnt)` \\

1 & `r sum(dfb1.cnt)` \\

2 & `r sum(dfb2.cnt)` \\

3 & `r sum(dfb3.cnt)` \\

4 & `r sum(dfb4.cnt)` \\

5 & `r sum(dfb5.cnt)` \\

6 & `r sum(dfb6.cnt)` \\

7 & `r sum(dfb7.cnt)` \\

8 & `r sum(dfb8.cnt)` \\

9 & `r sum(dfb9.cnt)` \\

10 & `r sum(dfb10.cnt)` \\

11 & `r sum(dfb11.cnt)` \\

12 & `r sum(dfb12.cnt)`

\end{array}

\end{align\*}

$$

In the above table, we can see that each parameter had a low frequency of points that were considered influential relative to the entire data set. For each point, we recorded the number of parameters it was influential in calculating, to serve for further investigation for potential recording errors. We note that two points, $4381$ and $5501$ were considered to be influential in calculating each model parameter. Next, we calculated the $DFFITS$ for the model.

```{r, echo = F}

par( mfrow = c(1,2) )

plot(dffits(wine.final), pch = 20, ylab = "DFFITS", main = "DFFITS")

abline( h = 2 \* sqrt( p / n ), lty = 2, lwd = 2, col = "red")

abline( h = -2 \* sqrt( p / n ), lty = 2, lwd = 2, col = "red")

plot(dffits(wine.final), pch = 20, ylim = c(-1.5,1.5), ylab = "DFFITS", main = "DFFITS (Pt 4381 Excluded)")

abline( h = 2 \* sqrt( p / n ), lty = 2, lwd = 2, col = "red")

abline( h = -2 \* sqrt( p / n ), lty = 2, lwd = 2, col = "red")

par( mfrow = c(1,1) )

```

We collected all points that exceeded the threshold used for testing how influential the $i^{th}$ observation is in the calculation of $\hat{y}\_i$, given by $|DFFITS\_i| > 2\sqrt{\frac{p}{n}}$. On the left plot, we notice that point 4381 obscures the view of the remaining points by vertically compressing the graph. On the right, with point 4381 excluded, we can more easily see points the collection of points that exceed the threshold. We note that points with lower index values appear to be more influential in the calculation of the lower indexed fitted values. This is interesting because the red wines are listed at lower indices, and white wines at higher indices. We observe that white wines appear to exceed the $DFFITS$ threshold less often than their red counterparts, but those that do are more pronounced in the distance away from the horizontal line at zero. Next, we will review the $COVRATIO$ for the data set.

```{r, echo = F}

par( mfrow = c(1,2) )

plot(covratio(wine.final), pch = 20, ylab = "COVRATIO", main = "COVRATIO")

abline( h = 1 + 3 \* (p / n), lty = 2, lwd = 2, col = "red")

abline( h = 1 - 3 \* (p / n), lty = 2, lwd = 2, col = "red")

plot(covratio(wine.final), pch = 20, ylab = "COVRATIO", ylim = c(.9,1.1), main = "COVRATIO (Pt 4381 Excluded)")

abline( h = 1 + 3 \* (p / n), lty = 2, lwd = 2, col = "red")

abline( h = 1 - 3 \* (p / n), lty = 2, lwd = 2, col = "red")

par(mfrow = c(1,1) )

```

In the above plot on the left, we see that the inclusion of point $4381$ greatly reduces the precision of the model. However, we note that there are three other points that degrade model precision as well: $3253$, $3263$, and $5501$. Removing point $4381$ from the view, we obtain the above plot on the right. What we notice in this plot is that there is a large cluster of points on the left-hand side that fall below the lower threshold. The first $1,599$ points in the data set belong to red wine, while the remaining $4,898$ points belong to white wine. What we observe is that the model is trained largely to fit white wine, and the fitting of red wine appears to affect precision by including it in the same data set as the white wine. In general, we see more spread for the points on the left side of the axis on either $COVRATIO$ plot. Overall, we note that $`r sum(help.cnt)`$ points in the data set improve model precision by including them in the data set. On the other hand, we also note that $`r sum(hurt.cnt)`$ points in the data set degrade model precision with their inclusion in the data set.

Based on these analyses for detecting outliers and influential points, we have collected all points that could be considered outliers. As such, we could review the points to determine whether any of them were true recording errors, and simultaneously improve the fitting of the model. However, while this may improve the fit of the model, any potential tests may be hindered by the refining of the data. Next, we will discuss the interpretation of the model coefficients.

Below, we provide the model summaries for both the unscaled and the scaled models. Following the summaries, we interpret the model coefficients. Since the original data source does not specify exact units for the predictors, we adhere to a general interpretation of coefficients for the unscaled model.

```{r, echo = F}

summary(wine.final)

```

For the unscaled model, we have the following coefficient interpretations:

Fixed acidity: When fixed acidity increases by one unit, alcohol content increases by approximately $.5270$%, on average, when all other variables are held constant.

Volatile acidity: When volatile acidity increases by one unit, alcohol content increases by approximately $.9215$%, on average, when all other variables are held constant.

Residual sugar: When residual sugar increases by one unit, alcohol content increases by approximately $.2392$%, on average, when all other variables are held constant.

Chlorides: When chlorides increase by one unit, alcohol content decreases by approximately $.9834$%, on average, when all other variables are held constant.

Free sulfur dioxide: When free sulfur dioxide increases by one unit, alcohol content decreases by approximately $.0031$%, on average, when all other variables are held constant.

Citric acid: When citric acid increases by one unit, alcohol content increases by approximately $.4660$%, on average, when all other variables are held constant.

Density: When density increases by one unit, alcohol content decreases by approximately $681.6$%, on average, when all other variables are held constant.

pH: When pH increases by one unit, alcohol content increases by approximately $2.719$%, on average, when all other variables are held constant.

Sulphates: When sulphates increase by one unit, alcohol content increases by approximately $1.056$%, on average, when all other variables are held constant.

Red: Alcohol content for red wine is approximately $1.277$% higher than alcohol content for white wine, on average, when all other variables are equal and the only difference is wine color.

Fixed acidity:Red: Among red wines, when fixed acidity increases by one unit, alcohol content increases by approximately $.0257$%, on average, when all other variables are held constant.

Volatile acidity:Red: Among red wines, when volatile acidity increases by one unit, alcohol content decreases by approximately $.5900$%, on average, when all other variables are held constant.

Most of the coefficients in the model have a reasonable interpretation. The exception is the coefficient for density, which is too high to realistically have the effect that our model states. This is due to there being a low variance for most observations of the density predictor. We note that the observation that appeared on every influence diagnostic plot, $4381$, has a density that was considerably different from the other $6,496$ observations. In the below figure, we show how much point $4381$ varied in density from the other points.

```{r, echo = F}

plot(winedata$density, winedata$alcohol, pch = 20, col = ifelse(winedata$density >= 1.03, "red", "black"), xlab = "Density", ylab = "Alcohol")

```

Since the above plot shows that the value for density can create a large residual that shows up in each diagnostic plot for leverage and influence, we created a scaled data set to test whether the same issue occurred. In our scaled data set, we performed unit normal scaling for the response and each predictor. Below, we present the model summary:

```{r, echo = F}

summary(wine.final.scaled)

```

For the scaled model, we have the following coefficient interpretations:

Fixed acidity: When fixed acidity increases by one standard deviation, the mean standardized response increases by approximately $.5728$ when all other variables are held constant.

Volatile acidity: When volatile acidity increases by one standard deviation, the mean standardized response increases by approximately $.1272$ when all other variables are held constant.

Residual sugar: When residual sugar increases by one standard deviation, the mean standardized response increases by approximately $.9543$ when all other variables are held constant.

Chlorides: When chlorides increase by one standard deviation, the mean standardized response decreases by approximately $.0289$ when all other variables are held constant.

Free sulfur dioxide: When free sulfur dioxide increases by one standard deviation, the mean standardized response decreases by approximately $.0467$ when all other variables are held constant.

Citric acid: When citric acid increases by one standard deviation, the mean standardized response increases by approximately $.0568$ when all other variables are held constant.

Density: When density increases by one standard deviation, the mean standardized response decreases by approximately $1.7136$ when all other variables are held constant.

pH: When pH increases by one standard deviation, the mean standardized response increases by approximately $.3665$ when all other variables are held constant.

Sulphates: When sulphates increase by one standard deviation, the mean standardized response increases by approximately $.1317$ when all other variables are held constant.

Red: The standardized mean response for red wine is approximately $1.0583$ higher than for white wine, when all other variables are held constant and only differ by wine color.

Fixed acidity:Red: Among red wines, when fixed acidity increases by one standard deviation, the mean standardized response increases by approximately $.0280$ when all other variables are held constant.

Volatile acidity:Red: Among red wines, when volatile acidity increases by one standard deviation, the mean standardized response decreases by approximately $.0814$ when all other variables are held constant.

Comparing the magnitude of the coefficient estimates between the scaled and unscaled models, we notice that the scaled model can serve to inform future data collection methods and research, in particular as it relates to the density and residual sugar content of the wines. It appears that residual sugar, density, color, and both fixed and volatile acidity play a crucial role in determining alcohol content. We also note that these variables will likely differ between red and white wines, illustrated in the scatterplots color-coded by wine color, located in Appendix A.

Overall, we regard our preference for the unscaled model, as its units are more readily understandable to a general audience. The fact that the model abysmally predicts a small number of points should not be a cause for concern, but rather an opportunity to collect a broader data set.

## Conclusion

Our work began with modeling wine quality, and due to modeling complications and lack of achieving a satisfying fit to the data, we opted to model wine alcohol content. We explored scatterplots of the predictors and identified interaction terms to address the potential differences in physicochemical properties between red wine and white wine. The final model produced satisfactory model statistics, but we are concerned about the few points in the data set that were identified in our analysis to be given undue influence on the model. We will address some final thoughts regarding our work.

One potential future direction for this project is to use the split data sets for red and white wine and observe how well the final model fits the two data sets separately. This idea stems from the fact that the left-side of the $COVRATIO$ plots, which represented red wines in the data set, appeared to have a higher frequency of points that exceeded thresholds for both enhancing and degrading precision. One wonders if this is due to the model being accountable for both colors of wine.

Based on the work done regarding potential influential points, it would be interesting to see how the model statistics may differ with the removal of certain points according to a specified criteria such as removing points that exceed the interquartile range multiplied by some scalar. For example, we have discussed the considerable influence that point $4381$ exerted on the model. Out of curiosity, we examined the model fit with the specified point omitted. We observed a considerable increase in $R\_{adj}^2$, from $.8227$ to $.8412$, without changing the specification of the model. It would be interesting to have a model that performs even better for wines that do not possess extreme values of density and residual sugar.

Similarly, we could try a different direction, where we instead consider improving the fit of the model by collecting more data. Point $4381$ represents an interesting observation, being more dense and containing more residual sugar than the other points. Perhaps the reason the model does not fit particularly well is due to a lack of sampling wines with high density and high residual sugar. We could gather a larger sample size and pay special attention to these variables, ensuring that we have an adequate representation of various wines. Our result would be a more robust model that could accurately determine the alcohol content for many types of wines.

## References

1. 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.

## Appendix A: Exploratory Data Analysis Plots

```{r, echo = F}

# Scatterplots of response versus each predictor

plot(wine$fixed.acidity, wine$alcohol, pch = 20, xlab = "Fixed Acidity", ylab = "Alcohol", main = "Alcohol vs. Fixed Acidity Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$volatile.acidity, wine$alcohol, pch = 20, xlab = "Volatile Acidity", ylab = "Alcohol", main = "Alcohol vs. Volatile Acidity Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$citric.acid, wine$alcohol, pch = 20, xlab = "Citric Acid", ylab = "Alcohol", main = "Alcohol vs. Citric Acid Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$residual.sugar, wine$alcohol, pch = 20, xlab = "Residual Sugar", ylab = "Alcohol", main = "Alcohol vs. Residual Sugar Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$chlorides, wine$alcohol, pch = 20, xlab = "Chlorides", ylab = "Alcohol", main = "Alcohol vs. Chlorides Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$free.sulfur.dioxide, wine$alcohol, pch = 20, xlab = "Free Sulfur Dioxide", ylab = "Alcohol", main = "Alcohol vs. Free Sulfur Dioxide Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$total.sulfur.dioxide, wine$alcohol, pch = 20, xlab = "Total Sulfur Dioxide", ylab = "Alcohol", main = "Alcohol vs. Total Sulfur Dioxide Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$density, wine$alcohol, pch = 20, xlab = "Density", ylab = "Alcohol", main = "Alcohol vs. Density Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$pH, wine$alcohol, pch = 20, xlab = "pH", ylab = "Alcohol", main = "Alcohol vs. pH Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$sulphates, wine$alcohol, pch = 20, xlab = "Sulphates", ylab = "Alcohol", main = "Alcohol vs. Sulphates Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$color, wine$alcohol, pch = 20, xlab = "Color (Red = 1, White = 0)", ylab = "Alcohol", main = "Alcohol vs. Wine Color Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$quality, wine$alcohol, pch = 20, xlab = "Wine Quality", ylab = "Alcohol", main = "Alcohol vs. Quality Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

# mean(wine$fixed.acidity[wine$wine.color == "red"])

# mean(wine$fixed.acidity[wine$wine.color == "white"])

# mean(wine$fixed.acidity)

# mean(wine$volatile.acidity[wine$wine.color == "red"])

# mean(wine$volatile.acidity[wine$wine.color == "white"])

# mean(wine$volatile.acidity)

# mean(wine$citric.acid[wine$wine.color == "red"])

# mean(wine$citric.acid[wine$wine.color == "white"])

# mean(wine$citric.acid)

# mean(wine$residual.sugar[wine$wine.color == "red"])

# mean(wine$residual.sugar[wine$wine.color == "white"])

# mean(wine$residual.sugar)

# mean(wine$chlorides[wine$wine.color == "red"])

# mean(wine$chlorides[wine$wine.color == "white"])

# mean(wine$chlorides)

# mean(wine$free.sulfur.dioxide[wine$wine.color == "red"])

# mean(wine$free.sulfur.dioxide[wine$wine.color == "white"])

# mean(wine$free.sulfur.dioxide)

# mean(wine$total.sulfur.dioxide[wine$wine.color == "red"])

# mean(wine$total.sulfur.dioxide[wine$wine.color == "white"])

# mean(wine$total.sulfur.dioxide)

# mean(wine$density[wine$wine.color == "red"])

# mean(wine$density[wine$wine.color == "white"])

# mean(wine$density)

# mean(wine$pH[wine$wine.color == "red"])

# mean(wine$pH[wine$wine.color == "white"])

# mean(wine$pH)

# mean(wine$sulphates[wine$wine.color == "red"])

# mean(wine$sulphates[wine$wine.color == "white"])

# mean(wine$sulphates)

# mean(wine$alcohol[wine$wine.color == "red"])

# mean(wine$alcohol[wine$wine.color == "white"])

# mean(wine$alcohol)

# mean(wine$quality[wine$wine.color == "red"])

# mean(wine$quality[wine$wine.color == "white"])

# mean(wine$quality)

```

```{r}

# Boxplots of all predictors by wine color and in total

# par(mfrow = c(1,3))

# boxplot(wine$fixed.acidity[wine$wine.color == "red"], ylab = "Fixed Acidity", xlab = "Red Wine")

# boxplot(wine$fixed.acidity[wine$wine.color == "white"], ylab = "Fixed Acidity", xlab = "White Wine")

# boxplot(wine$fixed.acidity, ylab = "Fixed Acidity", xlab = "Total Wine")

# par(mfrow = c(1,1))

#

# par(mfrow = c(1,3))

# boxplot(wine$volatile.acidity[wine$wine.color == "red"], ylab = "Volatile Acidity", xlab = "Red Wine")

# boxplot(wine$volatile.acidity[wine$wine.color == "white"], ylab = "Volatile Acidity", xlab = "White Wine")

# boxplot(wine$volatile.acidity, ylab = "Volatile Acidity", xlab = "Total Wine")

# par(mfrow = c(1,1))

#

# par(mfrow = c(1,3))

# boxplot(wine$citric.acid[wine$wine.color == "red"], ylab = "Citric Acid", xlab = "Red Wine")

# boxplot(wine$citric.acid[wine$wine.color == "white"], ylab = "Citric Acid", xlab = "White Wine")

# boxplot(wine$citric.acid, ylab = "Citric Acid", xlab = "Total Wine")

# par(mfrow = c(1,1))

#

# par(mfrow = c(1,3))

# boxplot(wine$residual.sugar[wine$wine.color == "red"], ylab = "Residual Sugar", xlab = "Red Wine")

# boxplot(wine$residual.sugar[wine$wine.color == "white"], ylab = "Residual Sugar", xlab = "White Wine")

# boxplot(wine$residual.sugar, ylab = "Residual Sugar", xlab = "Total Wine")

# par(mfrow = c(1,1))

#

# par(mfrow = c(1,3))

# boxplot(wine$chlorides[wine$wine.color == "red"], ylab = "Chlorides", xlab = "Red Wine")

# boxplot(wine$chlorides[wine$wine.color == "white"], ylab = "Chlorides", xlab = "White Wine")

# boxplot(wine$chlorides, ylab = "Chlorides", xlab = "Total Wine")

# par(mfrow = c(1,1))

#

# par(mfrow = c(1,3))

# boxplot(wine$free.sulfur.dioxide[wine$wine.color == "red"], ylab = "Free Sulfur Dioxide", xlab = "Red Wine")

# boxplot(wine$free.sulfur.dioxide[wine$wine.color == "white"], ylab = "Free Sulfur Dioxide", xlab = "White Wine")

# boxplot(wine$free.sulfur.dioxide, ylab = "Free Sulfur Dioxide", xlab = "Total Wine")

# par(mfrow = c(1,1))

#

# par(mfrow = c(1,3))

# boxplot(wine$total.sulfur.dioxide[wine$wine.color == "red"], ylab = "Total Sulfur Dioxide", xlab = "Red Wine")

# boxplot(wine$total.sulfur.dioxide[wine$wine.color == "white"], ylab = "Total Sulfur Dioxide", xlab = "White Wine")

# boxplot(wine$total.sulfur.dioxide, ylab = "Total Sulfur Dioxide", xlab = "Total Wine")

# par(mfrow = c(1,1))

#

# par(mfrow = c(1,3))

# boxplot(wine$density[wine$wine.color == "red"], ylab = "Density", xlab = "Red Wine")

# boxplot(wine$density[wine$wine.color == "white"], ylab = "Density", xlab = "White Wine")

# boxplot(wine$density, ylab = "Density", xlab = "Total Wine")

# par(mfrow = c(1,1))

#

# par(mfrow = c(1,3))

# boxplot(wine$pH[wine$wine.color == "red"], ylab = "pH", xlab = "Red Wine")

# boxplot(wine$pH[wine$wine.color == "white"], ylab = "pH", xlab = "White Wine")

# boxplot(wine$pH, ylab = "pH", xlab = "Total Wine")

# par(mfrow = c(1,1))

#

# par(mfrow = c(1,3))

# boxplot(wine$sulphates[wine$wine.color == "red"], ylab = "Sulphates", xlab = "Red Wine")

# boxplot(wine$sulphates[wine$wine.color == "white"], ylab = "Sulphates", xlab = "White Wine")

# boxplot(wine$sulphates, ylab = "Sulphates", xlab = "Total Wine")

# par(mfrow = c(1,1))

#

# par(mfrow = c(1,3))

# boxplot(wine$quality[wine$wine.color == "red"], ylab = "Quality", xlab = "Red Wine")

# boxplot(wine$quality[wine$wine.color == "white"], ylab = "Quality", xlab = "White Wine")

# boxplot(wine$quality, ylab = "Quality", xlab = "Total Wine")

# par(mfrow = c(1,1))

#

# par(mfrow = c(1,3))

# boxplot(wine$alcohol[wine$wine.color == "red"], ylab = "Alcohol", xlab = "Red Wine")

# boxplot(wine$alcohol[wine$wine.color == "white"], ylab = "Alcohol", xlab = "White Wine")

# boxplot(wine$alcohol, ylab = "Alcohol", xlab = "Total Wine")

# par(mfrow = c(1,1))

```

```{r}

# Barplot of color split

# wine.red <- sum(wine$wine.color == "red")

# wine.white <- sum(wine$wine.color == "white")

# wine.table <- with(wine, table(wine.color))

# barplot(wine.table, names.arg = c("Red", "White"), main = "Barplot of Wine Color", ylim = c(0,6000), beside = T)

```

```{r}

# Histograms displaying the distribution of predictor values in total and by wine color

# par( mfrow = c(1,3))

# hist(wine$fixed.acidity[wine$wine.color == "red"], xlab = "Red", main = "Fixed Acidity")

# hist(wine$fixed.acidity[wine$wine.color == "white"], xlab = "White", main = "Fixed Acidity")

# hist(wine$fixed.acidity, xlab = "Total", main = "Fixed Acidity")

# par( mfrow = c(1,1))

#

# par( mfrow = c(1,3))

# hist(wine$volatile.acidity[wine$wine.color == "red"], xlab = "Red", main = "Volatile Acidity")

# hist(wine$volatile.acidity[wine$wine.color == "white"], xlab = "White", main = "Volatile Acidity")

# hist(wine$volatile.acidity, xlab = "Total", main = "Volatile Acidity")

# par( mfrow = c(1,1))

#

# par( mfrow = c(1,3))

# hist(wine$citric.acid[wine$wine.color == "red"], xlab = "Red", main = "Citric Acid")

# hist(wine$citric.acid[wine$wine.color == "white"], xlab = "White", main = "Citric Acid")

# hist(wine$citric.acid, xlab = "Total", main = "Citric Acid")

# par( mfrow = c(1,1))

#

# par( mfrow = c(1,3))

# hist(wine$residual.sugar[wine$wine.color == "red"], xlab = "Red", main = "Residual Sugar")

# hist(wine$residual.sugar[wine$wine.color == "white"], xlab = "White", main = "Residual Sugar")

# hist(wine$residual.sugar, xlab = "Total", main = "Residual Sugar")

# par( mfrow = c(1,1))

#

# par( mfrow = c(1,3))

# hist(wine$chlorides[wine$wine.color == "red"], xlab = "Red", main = "Chlorides")

# hist(wine$chlorides[wine$wine.color == "white"], xlab = "White", main = "Chlorides")

# hist(wine$chlorides, xlab = "Total", main = "Chlorides")

# par( mfrow = c(1,1))

#

# par( mfrow = c(1,3))

# hist(wine$free.sulfur.dioxide[wine$wine.color == "red"], xlab = "Red", main = "Free Sulfur Dioxide")

# hist(wine$free.sulfur.dioxide[wine$wine.color == "white"], xlab = "White", main = "Free Sulfur Dioxide")

# hist(wine$free.sulfur.dioxide, xlab = "Total", main = "Free Sulfur Dioxide")

# par( mfrow = c(1,1))

#

# par( mfrow = c(1,3))

# hist(wine$total.sulfur.dioxide[wine$wine.color == "red"], xlab = "Red", main = "Total Sulfur Dioxide")

# hist(wine$total.sulfur.dioxide[wine$wine.color == "white"], xlab = "White", main = "Total Sulfur Dioxide")

# hist(wine$total.sulfur.dioxide, xlab = "Total", main = "Total Sulfur Dioxide")

# par( mfrow = c(1,1))

#

# par( mfrow = c(1,3))

# hist(wine$density[wine$wine.color == "red"], xlab = "Red", main = "Density")

# hist(wine$density[wine$wine.color == "white"], xlab = "White", main = "Density")

# hist(wine$density, xlab = "Total", main = "Density")

# par( mfrow = c(1,1))

#

# par( mfrow = c(1,3))

# hist(wine$pH[wine$wine.color == "red"], xlab = "Red", main = "pH")

# hist(wine$pH[wine$wine.color == "white"], xlab = "White", main = "pH")

# hist(wine$pH, xlab = "Total", main = "pH")

# par( mfrow = c(1,1))

#

# par( mfrow = c(1,3))

# hist(wine$sulphates[wine$wine.color == "red"], xlab = "Red", main = "Sulphates")

# hist(wine$sulphates[wine$wine.color == "white"], xlab = "White", main = "Sulphates")

# hist(wine$sulphates, xlab = "Total", main = "Sulphates")

# par( mfrow = c(1,1))

#

# par( mfrow = c(1,3))

# hist(wine$alcohol[wine$wine.color == "red"], xlab = "Red", main = "Alcohol")

# hist(wine$alcohol[wine$wine.color == "white"], xlab = "White", main = "Alcohol")

# hist(wine$alcohol, xlab = "Total", main = "Alcohol")

# par( mfrow = c(1,1))

#

# par( mfrow = c(1,3))

# hist(wine$quality[wine$wine.color == "red"], xlab = "Red", main = "Quality")

# hist(wine$quality[wine$wine.color == "white"], xlab = "White", main = "Quality")

# hist(wine$quality, xlab = "Total", main = "Quality")

# par( mfrow = c(1,1))

#

# par( mfrow = c(1,3))

# hist(wine$alcohol[wine$wine.color == "red"], xlab = "Red", main = "Alcohol")

# hist(wine$alcohol[wine$wine.color == "white"], xlab = "White", main = "Alcohol")

# hist(wine$alcohol, xlab = "Total", main = "Alcohol")

# par( mfrow = c(1,1))

```

```{r, echo = F}

# DFBETAS PLOTS

plot(dfbetas(wine.final)[,1], ylab = "DFBETAS\_0", pch = 20, main = "DFBETAS\_0")

abline(h = 2/sqrt(n), lty = 2, col = "red")

abline(h = -2/sqrt(n), lty = 2, col = "red")

plot(dfbetas(wine.final)[,2], ylab = "DFBETAS\_1", pch = 20, main = "DFBETAS\_1")

abline(h = 2/sqrt(n), lty = 2, col = "red")

abline(h = -2/sqrt(n), lty = 2, col = "red")

plot(dfbetas(wine.final)[,3], ylab = "DFBETAS\_2", pch = 20, main = "DFBETAS\_2")

abline(h = 2/sqrt(n), lty = 2, col = "red")

abline(h = -2/sqrt(n), lty = 2, col = "red")

plot(dfbetas(wine.final)[,4], ylab = "DFBETAS\_3", pch = 20, main = "DFBETAS\_3")

abline(h = 2/sqrt(n), lty = 2, col = "red")

abline(h = -2/sqrt(n), lty = 2, col = "red")

plot(dfbetas(wine.final)[,5], ylab = "DFBETAS\_4", pch = 20, main = "DFBETAS\_4")

abline(h = 2/sqrt(n), lty = 2, col = "red")

abline(h = -2/sqrt(n), lty = 2, col = "red")

plot(dfbetas(wine.final)[,6], ylab = "DFBETAS\_5", pch = 20, main = "DFBETAS\_5")

abline(h = 2/sqrt(n), lty = 2, col = "red")

abline(h = -2/sqrt(n), lty = 2, col = "red")

plot(dfbetas(wine.final)[,7], ylab = "DFBETAS\_6", pch = 20, main = "DFBETAS\_6")

abline(h = 2/sqrt(n), lty = 2, col = "red")

abline(h = -2/sqrt(n), lty = 2, col = "red")

plot(dfbetas(wine.final)[,8], ylab = "DFBETAS\_7", pch = 20, main = "DFBETAS\_7")

abline(h = 2/sqrt(n), lty = 2, col = "red")

abline(h = -2/sqrt(n), lty = 2, col = "red")

plot(dfbetas(wine.final)[,9], ylab = "DFBETAS\_8", pch = 20, main = "DFBETAS\_8")

abline(h = 2/sqrt(n), lty = 2, col = "red")

abline(h = -2/sqrt(n), lty = 2, col = "red")

plot(dfbetas(wine.final)[,10], ylab = "DFBETAS\_9", pch = 20, main = "DFBETAS\_9")

abline(h = 2/sqrt(n), lty = 2, col = "red")

abline(h = -2/sqrt(n), lty = 2, col = "red")

plot(dfbetas(wine.final)[,11], ylab = "DFBETAS\_10", pch = 20, main = "DFBETAS\_10")

abline(h = 2/sqrt(n), lty = 2, col = "red")

abline(h = -2/sqrt(n), lty = 2, col = "red")

plot(dfbetas(wine.final)[,12], ylab = "DFBETAS\_11", pch = 20, main = "DFBETAS\_11")

abline(h = 2/sqrt(n), lty = 2, col = "red")

abline(h = -2/sqrt(n), lty = 2, col = "red")

plot(dfbetas(wine.final)[,13], ylab = "DFBETAS\_12", pch = 20, main = "DFBETAS\_12")

abline(h = 2/sqrt(n), lty = 2, col = "red")

abline(h = -2/sqrt(n), lty = 2, col = "red")

```

## Appendix B: R-Code

```

library(leaps) # used for variable selection leaps() function

library(dplyr) # used for arrange() function

wine <- read.csv("winequality.csv")

wine$color <- ifelse(wine$wine.color == "red", 1, 0)

y.wine <- wine[,12]

X.wine <- wine[, c(-1, -12)]

winedata <- as.data.frame(cbind(y.wine, X.wine))

names(winedata)[1] <- "alcohol"

summary(winedata)

C <- cor(X.wine)

C

winedata.scaled <- scale( winedata )

winedata.scaled <- as.data.frame(winedata.scaled)

alcohol <- winedata.scaled$alcohol # response

design <- winedata.scaled[, -1]

mymodel <- leaps( x = design, y = alcohol, method = "Cp", names = names( design ) )

# CpDiff <- mymodel$Cp - mymodel$size

# arrange( data.frame( mymodel$size, mymodel$which, mymodel$Cp, CpDiff ),CpDiff )

n <- dim(wine)[1]

lm.full <- lm( alcohol ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar + chlorides + free.sulfur.dioxide +

total.sulfur.dioxide + density + pH + sulphates + as.factor(color), data = winedata.scaled )

lm.base <- lm( alcohol ~ as.factor(color), data = winedata.scaled )

lm.back <- step( lm.full, scope = list( lower = ~1, upper = lm.full ), direction = "backward", trace = F, k = log(n) )

lm.forward <- step( lm.base, scope = list( lower = ~1, upper = lm.full ), direction = "forward", trace = F, k = log(n) )

lm.step <- step( lm.base, scope = list( lower = ~1, upper = lm.full ), direction = "both", trace = F, k = log(n) )

wine.final <- lm( alcohol ~ fixed.acidity + volatile.acidity + residual.sugar + chlorides + free.sulfur.dioxide + citric.acid +

density + pH + sulphates + as.factor(color) + as.factor(color):fixed.acidity + as.factor(color):volatile.acidity

, data = winedata )

wine.final.scaled <- lm( alcohol ~ fixed.acidity + volatile.acidity + residual.sugar + chlorides + free.sulfur.dioxide + citric.acid +

density + pH + sulphates + as.factor(color) + as.factor(color):fixed.acidity + as.factor(color):volatile.acidity + 0

, data = winedata.scaled )

yhat <- fitted(wine.final)

e <- residuals(wine.final)

d <- e/summary(wine.final)$sigma

r <- rstandard(wine.final)

t <- rstudent(wine.final)

n <- dim(wine)[1]

p <- length(wine.final$coefficients)

plot(wine.final, 1, pch = 20)

plot(wine.final, 2, pch = 20)

plot(yhat, t, pch = 20, xlab = "Fitted values", ylab = "R-Student Residuals", main = "R-Student Residuals vs. Fitted Values Wine Alcohol")

abline(h = 0, lty = 2, col = "grey")

abline(h = 3, lty = 2, col = "red")

abline(h = -3, lty = 2, col = "red")

plot(hatvalues(wine.final), pch = 20, ylab = "Hat Values", main = "Wine Alcohol Leverage")

abline( h = 2 \* (p / n), lty = 2, lwd = 2, col = "red" )

plot(hatvalues(wine.final), rstudent(wine.final), pch = 20, xlab = "Hat Values", ylab = "R-Student Residuals", main = "R-Student vs. Hat Values Wine Alcohol")

abline( v = 2 \* (p / n), lty = 2, lwd = 2, col = "red" )

abline( h = 3, lty = 2, lwd = 2, col = "red" )

abline( h = -3, lty = 2, lwd = 2, col = "red" )

plot(cooks.distance(wine.final), pch = 20, ylab = "Cook's Distance", main = "Cook's D.")

abline( h = 1, lty = 2, lwd = 2, col = "red")

lev.pts <- which(ifelse(hatvalues(wine.final) > 2 \* (p / n), T, F))

lev.cnt <- 0

for (i in lev.pts) {lev.cnt <- lev.cnt + 1}

lev.cnt/n

# dfbetas(wine.final)

dfb0 <- which(ifelse(abs(dfbetas(wine.final)[,1]) > (2/sqrt(n)), T, F))

dfb0.cnt <- rep(0, n)

for (i in dfb0){ dfb0.cnt[i] = dfb0.cnt[i] + 1 }

names(dfb0.cnt) <- c(paste(1:n, sep = ", "))

dfb1 <- which(ifelse(abs(dfbetas(wine.final)[,2]) > (2/sqrt(n)), T, F))

dfb1.cnt <- rep(0, n)

for (i in dfb1){ dfb1.cnt[i] = dfb1.cnt[i] + 1 }

names(dfb1.cnt) <- c(paste(1:n, sep = ", "))

dfb2 <- which(ifelse(abs(dfbetas(wine.final)[,3]) > (2/sqrt(n)), T, F))

dfb2.cnt <- rep(0, n)

for (i in dfb2){ dfb2.cnt[i] = dfb2.cnt[i] + 1 }

names(dfb2.cnt) <- c(paste(1:n, sep = ", "))

dfb3 <- which(ifelse(abs(dfbetas(wine.final)[,4]) > (2/sqrt(n)), T, F))

dfb3.cnt <- rep(0, n)

for (i in dfb3){ dfb3.cnt[i] = dfb3.cnt[i] + 1 }

names(dfb3.cnt) <- c(paste(1:n, sep = ", "))

dfb4 <- which(ifelse(abs(dfbetas(wine.final)[,5]) > (2/sqrt(n)), T, F))

dfb4.cnt <- rep(0, n)

for (i in dfb4){ dfb4.cnt[i] = dfb4.cnt[i] + 1 }

names(dfb4.cnt) <- c(paste(1:n, sep = ", "))

dfb5 <- which(ifelse(abs(dfbetas(wine.final)[,6]) > (2/sqrt(n)), T, F))

dfb5.cnt <- rep(0, n)

for (i in dfb5){ dfb5.cnt[i] = dfb5.cnt[i] + 1 }

names(dfb5.cnt) <- c(paste(1:n, sep = ", "))

dfb6 <- which(ifelse(abs(dfbetas(wine.final)[,7]) > (2/sqrt(n)), T, F))

dfb6.cnt <- rep(0, n)

for (i in dfb6){ dfb6.cnt[i] = dfb6.cnt[i] + 1 }

names(dfb6.cnt) <- c(paste(1:n, sep = ", "))

dfb7 <- which(ifelse(abs(dfbetas(wine.final)[,8]) > (2/sqrt(n)), T, F))

dfb7.cnt <- rep(0, n)

for (i in dfb7){ dfb7.cnt[i] = dfb7.cnt[i] + 1 }

names(dfb7.cnt) <- c(paste(1:n, sep = ", "))

dfb8 <- which(ifelse(abs(dfbetas(wine.final)[,9]) > (2/sqrt(n)), T, F))

dfb8.cnt <- rep(0, n)

for (i in dfb8){ dfb8.cnt[i] = dfb8.cnt[i] + 1 }

names(dfb8.cnt) <- c(paste(1:n, sep = ", "))

dfb9 <- which(ifelse(abs(dfbetas(wine.final)[,10]) > (2/sqrt(n)), T, F))

dfb9.cnt <- rep(0, n)

for (i in dfb9){ dfb9.cnt[i] = dfb9.cnt[i] + 1 }

names(dfb9.cnt) <- c(paste(1:n, sep = ", "))

dfb10 <- which(ifelse(abs(dfbetas(wine.final)[,11]) > (2/sqrt(n)), T, F))

dfb10.cnt <- rep(0, n)

for (i in dfb10){ dfb10.cnt[i] = dfb10.cnt[i] + 1 }

names(dfb10.cnt) <- c(paste(1:n, sep = ", "))

dfb11 <- which(ifelse(abs(dfbetas(wine.final)[,12]) > (2/sqrt(n)), T, F))

dfb11.cnt <- rep(0, n)

for (i in dfb11){ dfb11.cnt[i] = dfb11.cnt[i] + 1 }

names(dfb11.cnt) <- c(paste(1:n, sep = ", "))

dfb12 <- which(ifelse(abs(dfbetas(wine.final)[,13]) > (2/sqrt(n)), T, F))

dfb12.cnt <- rep(0, n)

for (i in dfb12){ dfb12.cnt[i] = dfb12.cnt[i] + 1 }

names(dfb12.cnt) <- c(paste(1:n, sep = ", "))

dfb.pts <- c(dfb0, dfb1, dfb2, dfb3, dfb4, dfb5, dfb6, dfb7, dfb8, dfb9, dfb10, dfb11,dfb12)

# dfb0;dfb1;dfb2;dfb3;dfb4;dfb5;dfb6;dfb7;dfb8;dfb9;dfb10;dfb11;dfb12

dfb.cnt <- rep(0, n)

for (i in dfb.pts){ dfb.cnt[i] = dfb.cnt[i] + 1 }

names(dfb.cnt) <- c(paste(1:n, sep = ", "))

# names(sort(dfb.cnt, decreasing = T)[1:20])

# sort(dfb.cnt, decreasing = T)[1:20]

# dffits(wine.final)

par( mfrow = c(1,2) )

plot(dffits(wine.final), pch = 20, ylab = "DFFITS", main = "DFFITS")

abline( h = 2 \* sqrt( p / n ), lty = 2, lwd = 2, col = "red")

abline( h = -2 \* sqrt( p / n ), lty = 2, lwd = 2, col = "red")

plot(dffits(wine.final), pch = 20, ylim = c(-1.5,1.5), ylab = "DFFITS", main = "DFFITS (Pt 4381 Excluded)")

abline( h = 2 \* sqrt( p / n ), lty = 2, lwd = 2, col = "red")

abline( h = -2 \* sqrt( p / n ), lty = 2, lwd = 2, col = "red")

par( mfrow = c(1,1) )

dff <- which(ifelse(abs(dffits(wine.final)) > 2 \* sqrt( p / n ), T, F))

dff.cnt <- rep(0, n)

for(i in dff){ dff.cnt[i] <- dff.cnt[i] + 1}

names(dff.cnt) <- c(paste(1:n, sep = ", "))

# sort(dff.cnt, decreasing = T)[1:20]

sum(dff.cnt)

# dff.cnt[dff.cnt == 1]

# covratio(wine.final)

par( mfrow = c(1,2) )

plot(covratio(wine.final), pch = 20, ylab = "COVRATIO", main = "COVRATIO")

abline( h = 1 + 3 \* (p / n), lty = 2, lwd = 2, col = "red")

abline( h = 1 - 3 \* (p / n), lty = 2, lwd = 2, col = "red")

plot(covratio(wine.final), pch = 20, ylab = "COVRATIO", ylim = c(.9,1.1), main = "COVRATIO (Pt 4381 Excluded)")

abline( h = 1 + 3 \* (p / n), lty = 2, lwd = 2, col = "red")

abline( h = 1 - 3 \* (p / n), lty = 2, lwd = 2, col = "red")

par(mfrow = c(1,1) )

cov.help <- which(ifelse(covratio(wine.final) > 1 + 3 \* (p / n ), T, F))

cov.hurt <- which(ifelse(covratio(wine.final) < 1 - 3 \* (p / n ), T, F))

help.cnt <- rep(0,n)

for (i in cov.help) {help.cnt[i] <- help.cnt[i] + 1}

names(help.cnt) <- c(paste(1:n, sep = ", "))

hurt.cnt <- rep(0,n)

for (i in cov.hurt) {hurt.cnt[i] <- hurt.cnt[i] + 1}

names(hurt.cnt) <- c(paste(1:n, sep = ", "))

# hurt.cnt[hurt.cnt == 1]

# Scatterplots of response versus each predictor

plot(wine$fixed.acidity, wine$alcohol, pch = 20, xlab = "Fixed Acidity", ylab = "Alcohol", main = "Alcohol vs. Fixed Acidity Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$volatile.acidity, wine$alcohol, pch = 20, xlab = "Volatile Acidity", ylab = "Alcohol", main = "Alcohol vs. Volatile Acidity Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$citric.acid, wine$alcohol, pch = 20, xlab = "Citric Acid", ylab = "Alcohol", main = "Alcohol vs. Citric Acid Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$residual.sugar, wine$alcohol, pch = 20, xlab = "Residual Sugar", ylab = "Alcohol", main = "Alcohol vs. Residual Sugar Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$chlorides, wine$alcohol, pch = 20, xlab = "Chlorides", ylab = "Alcohol", main = "Alcohol vs. Chlorides Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$free.sulfur.dioxide, wine$alcohol, pch = 20, xlab = "Free Sulfur Dioxide", ylab = "Alcohol", main = "Alcohol vs. Free Sulfur Dioxide Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$total.sulfur.dioxide, wine$alcohol, pch = 20, xlab = "Total Sulfur Dioxide", ylab = "Alcohol", main = "Alcohol vs. Total Sulfur Dioxide Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$density, wine$alcohol, pch = 20, xlab = "Density", ylab = "Alcohol", main = "Alcohol vs. Density Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$pH, wine$alcohol, pch = 20, xlab = "pH", ylab = "Alcohol", main = "Alcohol vs. pH Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$sulphates, wine$alcohol, pch = 20, xlab = "Sulphates", ylab = "Alcohol", main = "Alcohol vs. Sulphates Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$color, wine$alcohol, pch = 20, xlab = "Color (Red = 1, White = 0)", ylab = "Alcohol", main = "Alcohol vs. Wine Color Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

plot(wine$quality, wine$alcohol, pch = 20, xlab = "Wine Quality", ylab = "Alcohol", main = "Alcohol vs. Quality Scatterplot", col = ifelse(wine$wine.color == "red", "red", "black"), cex = 2)

# Means of our predictors, total and by color

mean(wine$fixed.acidity[wine$wine.color == "red"])

mean(wine$fixed.acidity[wine$wine.color == "white"])

mean(wine$fixed.acidity)

mean(wine$volatile.acidity[wine$wine.color == "red"])

mean(wine$volatile.acidity[wine$wine.color == "white"])

mean(wine$volatile.acidity)

mean(wine$citric.acid[wine$wine.color == "red"])

mean(wine$citric.acid[wine$wine.color == "white"])

mean(wine$citric.acid)

mean(wine$residual.sugar[wine$wine.color == "red"])

mean(wine$residual.sugar[wine$wine.color == "white"])

mean(wine$residual.sugar)

mean(wine$chlorides[wine$wine.color == "red"])

mean(wine$chlorides[wine$wine.color == "white"])

mean(wine$chlorides)

mean(wine$free.sulfur.dioxide[wine$wine.color == "red"])

mean(wine$free.sulfur.dioxide[wine$wine.color == "white"])

mean(wine$free.sulfur.dioxide)

mean(wine$total.sulfur.dioxide[wine$wine.color == "red"])

mean(wine$total.sulfur.dioxide[wine$wine.color == "white"])

mean(wine$total.sulfur.dioxide)

mean(wine$density[wine$wine.color == "red"])

mean(wine$density[wine$wine.color == "white"])

mean(wine$density)

mean(wine$pH[wine$wine.color == "red"])

mean(wine$pH[wine$wine.color == "white"])

mean(wine$pH)

mean(wine$sulphates[wine$wine.color == "red"])

mean(wine$sulphates[wine$wine.color == "white"])

mean(wine$sulphates)

mean(wine$alcohol[wine$wine.color == "red"])

mean(wine$alcohol[wine$wine.color == "white"])

mean(wine$alcohol)

mean(wine$quality[wine$wine.color == "red"])

mean(wine$quality[wine$wine.color == "white"])

mean(wine$quality)

# Boxplots of all predictors by wine color and in total

par(mfrow = c(1,3))

boxplot(wine$fixed.acidity[wine$wine.color == "red"], ylab = "Fixed Acidity", xlab = "Red Wine")

boxplot(wine$fixed.acidity[wine$wine.color == "white"], ylab = "Fixed Acidity", xlab = "White Wine")

boxplot(wine$fixed.acidity, ylab = "Fixed Acidity", xlab = "Total Wine")

par(mfrow = c(1,1))

par(mfrow = c(1,3))

boxplot(wine$volatile.acidity[wine$wine.color == "red"], ylab = "Volatile Acidity", xlab = "Red Wine")

boxplot(wine$volatile.acidity[wine$wine.color == "white"], ylab = "Volatile Acidity", xlab = "White Wine")

boxplot(wine$volatile.acidity, ylab = "Volatile Acidity", xlab = "Total Wine")

par(mfrow = c(1,1))

par(mfrow = c(1,3))

boxplot(wine$citric.acid[wine$wine.color == "red"], ylab = "Citric Acid", xlab = "Red Wine")

boxplot(wine$citric.acid[wine$wine.color == "white"], ylab = "Citric Acid", xlab = "White Wine")

boxplot(wine$citric.acid, ylab = "Citric Acid", xlab = "Total Wine")

par(mfrow = c(1,1))

par(mfrow = c(1,3))

boxplot(wine$residual.sugar[wine$wine.color == "red"], ylab = "Residual Sugar", xlab = "Red Wine")

boxplot(wine$residual.sugar[wine$wine.color == "white"], ylab = "Residual Sugar", xlab = "White Wine")

boxplot(wine$residual.sugar, ylab = "Residual Sugar", xlab = "Total Wine")

par(mfrow = c(1,1))

par(mfrow = c(1,3))

boxplot(wine$chlorides[wine$wine.color == "red"], ylab = "Chlorides", xlab = "Red Wine")

boxplot(wine$chlorides[wine$wine.color == "white"], ylab = "Chlorides", xlab = "White Wine")

boxplot(wine$chlorides, ylab = "Chlorides", xlab = "Total Wine")

par(mfrow = c(1,1))

par(mfrow = c(1,3))

boxplot(wine$free.sulfur.dioxide[wine$wine.color == "red"], ylab = "Free Sulfur Dioxide", xlab = "Red Wine")

boxplot(wine$free.sulfur.dioxide[wine$wine.color == "white"], ylab = "Free Sulfur Dioxide", xlab = "White Wine")

boxplot(wine$free.sulfur.dioxide, ylab = "Free Sulfur Dioxide", xlab = "Total Wine")

par(mfrow = c(1,1))

par(mfrow = c(1,3))

boxplot(wine$total.sulfur.dioxide[wine$wine.color == "red"], ylab = "Total Sulfur Dioxide", xlab = "Red Wine")

boxplot(wine$total.sulfur.dioxide[wine$wine.color == "white"], ylab = "Total Sulfur Dioxide", xlab = "White Wine")

boxplot(wine$total.sulfur.dioxide, ylab = "Total Sulfur Dioxide", xlab = "Total Wine")

par(mfrow = c(1,1))

par(mfrow = c(1,3))

boxplot(wine$density[wine$wine.color == "red"], ylab = "Density", xlab = "Red Wine")

boxplot(wine$density[wine$wine.color == "white"], ylab = "Density", xlab = "White Wine")

boxplot(wine$density, ylab = "Density", xlab = "Total Wine")

par(mfrow = c(1,1))

par(mfrow = c(1,3))

boxplot(wine$pH[wine$wine.color == "red"], ylab = "pH", xlab = "Red Wine")

boxplot(wine$pH[wine$wine.color == "white"], ylab = "pH", xlab = "White Wine")

boxplot(wine$pH, ylab = "pH", xlab = "Total Wine")

par(mfrow = c(1,1))

par(mfrow = c(1,3))

boxplot(wine$sulphates[wine$wine.color == "red"], ylab = "Sulphates", xlab = "Red Wine")

boxplot(wine$sulphates[wine$wine.color == "white"], ylab = "Sulphates", xlab = "White Wine")

boxplot(wine$sulphates, ylab = "Sulphates", xlab = "Total Wine")

par(mfrow = c(1,1))

par(mfrow = c(1,3))

boxplot(wine$quality[wine$wine.color == "red"], ylab = "Quality", xlab = "Red Wine")

boxplot(wine$quality[wine$wine.color == "white"], ylab = "Quality", xlab = "White Wine")

boxplot(wine$quality, ylab = "Quality", xlab = "Total Wine")

par(mfrow = c(1,1))

par(mfrow = c(1,3))

boxplot(wine$alcohol[wine$wine.color == "red"], ylab = "Alcohol", xlab = "Red Wine")

boxplot(wine$alcohol[wine$wine.color == "white"], ylab = "Alcohol", xlab = "White Wine")

boxplot(wine$alcohol, ylab = "Alcohol", xlab = "Total Wine")

par(mfrow = c(1,1))

# Barplot of color split

wine.red <- sum(wine$wine.color == "red")

wine.white <- sum(wine$wine.color == "white")

wine.table <- with(wine, table(wine.color))

barplot(wine.table, names.arg = c("Red", "White"), main = "Barplot of Wine Color", ylim = c(0,6000), beside = T)

# Histograms displaying the distribution of predictor values in total and by wine color

par( mfrow = c(1,3))

hist(wine$fixed.acidity[wine$wine.color == "red"], xlab = "Red", main = "Fixed Acidity")

hist(wine$fixed.acidity[wine$wine.color == "white"], xlab = "White", main = "Fixed Acidity")

hist(wine$fixed.acidity, xlab = "Total", main = "Fixed Acidity")

par( mfrow = c(1,1))

par( mfrow = c(1,3))

hist(wine$volatile.acidity[wine$wine.color == "red"], xlab = "Red", main = "Volatile Acidity")

hist(wine$volatile.acidity[wine$wine.color == "white"], xlab = "White", main = "Volatile Acidity")

hist(wine$volatile.acidity, xlab = "Total", main = "Volatile Acidity")

par( mfrow = c(1,1))

par( mfrow = c(1,3))

hist(wine$citric.acid[wine$wine.color == "red"], xlab = "Red", main = "Citric Acid")

hist(wine$citric.acid[wine$wine.color == "white"], xlab = "White", main = "Citric Acid")

hist(wine$citric.acid, xlab = "Total", main = "Citric Acid")

par( mfrow = c(1,1))

par( mfrow = c(1,3))

hist(wine$residual.sugar[wine$wine.color == "red"], xlab = "Red", main = "Residual Sugar")

hist(wine$residual.sugar[wine$wine.color == "white"], xlab = "White", main = "Residual Sugar")

hist(wine$residual.sugar, xlab = "Total", main = "Residual Sugar")

par( mfrow = c(1,1))

par( mfrow = c(1,3))

hist(wine$chlorides[wine$wine.color == "red"], xlab = "Red", main = "Chlorides")

hist(wine$chlorides[wine$wine.color == "white"], xlab = "White", main = "Chlorides")

hist(wine$chlorides, xlab = "Total", main = "Chlorides")

par( mfrow = c(1,1))

par( mfrow = c(1,3))

hist(wine$free.sulfur.dioxide[wine$wine.color == "red"], xlab = "Red", main = "Free Sulfur Dioxide")

hist(wine$free.sulfur.dioxide[wine$wine.color == "white"], xlab = "White", main = "Free Sulfur Dioxide")

hist(wine$free.sulfur.dioxide, xlab = "Total", main = "Free Sulfur Dioxide")

par( mfrow = c(1,1))

par( mfrow = c(1,3))

hist(wine$total.sulfur.dioxide[wine$wine.color == "red"], xlab = "Red", main = "Total Sulfur Dioxide")

hist(wine$total.sulfur.dioxide[wine$wine.color == "white"], xlab = "White", main = "Total Sulfur Dioxide")

hist(wine$total.sulfur.dioxide, xlab = "Total", main = "Total Sulfur Dioxide")

par( mfrow = c(1,1))

par( mfrow = c(1,3))

hist(wine$density[wine$wine.color == "red"], xlab = "Red", main = "Density")

hist(wine$density[wine$wine.color == "white"], xlab = "White", main = "Density")

hist(wine$density, xlab = "Total", main = "Density")

par( mfrow = c(1,1))

par( mfrow = c(1,3))

hist(wine$pH[wine$wine.color == "red"], xlab = "Red", main = "pH")

hist(wine$pH[wine$wine.color == "white"], xlab = "White", main = "pH")

hist(wine$pH, xlab = "Total", main = "pH")

par( mfrow = c(1,1))

par( mfrow = c(1,3))

hist(wine$sulphates[wine$wine.color == "red"], xlab = "Red", main = "Sulphates")

hist(wine$sulphates[wine$wine.color == "white"], xlab = "White", main = "Sulphates")

hist(wine$sulphates, xlab = "Total", main = "Sulphates")

par( mfrow = c(1,1))

par( mfrow = c(1,3))

hist(wine$alcohol[wine$wine.color == "red"], xlab = "Red", main = "Alcohol")

hist(wine$alcohol[wine$wine.color == "white"], xlab = "White", main = "Alcohol")

hist(wine$alcohol, xlab = "Total", main = "Alcohol")

par( mfrow = c(1,1))

par( mfrow = c(1,3))

hist(wine$quality[wine$wine.color == "red"], xlab = "Red", main = "Quality")

hist(wine$quality[wine$wine.color == "white"], xlab = "White", main = "Quality")

hist(wine$quality, xlab = "Total", main = "Quality")

par( mfrow = c(1,1))

par( mfrow = c(1,3))

hist(wine$alcohol[wine$wine.color == "red"], xlab = "Red", main = "Alcohol")

hist(wine$alcohol[wine$wine.color == "white"], xlab = "White", main = "Alcohol")

hist(wine$alcohol, xlab = "Total", main = "Alcohol")

par( mfrow = c(1,1))

```