Predicting Wine Quality

By Andrew Zhang, Elena Joseph, Francesca Valdes, Mateo Pesa

Introduction

Wine is an alcoholic beverage enjoyed across the world. In 2020 alone, more than 234 million hectolitres of wine were consumed globally (Karrlson, 2021). As wine is routinely intertwined with people's lives, its quality is often the center of discussion. Everyday folk often see the hues, smell the aromas, swirl, sip, and savor the aftertaste of the wine to gain clarity on its quality (Sinkler, 2014). However, understanding wine quality is far from simple. There is artistry behind winemaking, often aided by generational practices and modern technological techniques. Moreover, external factors can also heavily impact wine quality, such as climate change. For example, the rising temperatures may change the ecosystems of vineyards, impacting the growing seasons of the grapes (Scott, 2022). Therefore, several factors impact wine quality. To gain a better understanding of this, we aim to explore the impact of various physiochemical features on wine quality and more specifically, if these features vary for red wine and white wine.

Data Description and Preprocessing

Data Description

The dataset that we will be analyzing is "wine-quality-white-and-red.csv". This dataset was created by merging "wine-quality-white.csv" and "wine-quality-red.csv", and adding a column to distinguish between the types of wine, where 0 represents red wine and 1 represents white wine. Another strategy that we employed was to create a new factor column of the labels "red" and "white".

The dataset analyzes various physicochemical attributes of red and white vinho verde wines. These wines are found in the Minho region of Portugal and were collected from May 2004 to February 2007 (Cortez et. al, 2009). This dataset is taken from the University of California Irvine Machine Learning Repository based on the aforementioned wines. The sample size has 6497 records, with 1599 records of red wine and 4898 records of white wine. No missing feature values are included in these records. There are 13 variables in this dataset associated with each sample including:

- Fixed.acidity: the concentration of tartaric acid in grams per cubic decimeter
- Volatile.acidity: the concentration of acetic acid in grams per cubic decimeter
- Citric.acid: the concentration of citric acid in grams per cubic decimeter
- Residual.sugar: the concentration of sugar in grams per cubic decimeter
- Chlorides: the concentration of sodium chloride in grams per cubic decimeter
- Free.sulfur.dioxide: the concentration of milligrams of unbound sulfur dioxide per cubic decimeter
- Total.sulfur.dioxide: the concentration of milligrams of bound sulfur dioxide per cubic decimeter
- Density: the mass within a volume of solution, measured in grams per cubic centimeter
- pH: the acidity of the wine from 0 to 14, where 0 is very acidic and 14 is very basic
- Sulphates: the concentration of potassium sulfate in grams per cubic decimeter
- Alcohol: the percentage of alcohol within the solution
- Quality: score between 0 and 10 based on sensory data
- Type: if the wine is red or white

type	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
white	7.0	0.27	0.36	20.7	0.045	45	170	1.0010	3.00	0.45	8.8	6
white	6.3	0.30	0.34	1.6	0.049	14	132	0.9940	3.30	0.49	9.5	6
white	8.1	0.28	0.40	6.9	0.050	30	97	0.9951	3.26	0.44	10.1	6

Figure 1. Example of 3 observations of Merged Dataset

Exploratory Data Analysis:

We first wanted to visualize the data that we were working with so we made two boxplots, as seen in Figure 4, *Jitter Plot Grouped by Wine Type,* side by side, of the white and red wine type and their quality level on the y axis. This allowed us to see that there is a larger variety of white wine data as there are more data points with quality below 4.25, as well as more data of white wine with quality higher than around 6.75.

Alongside this, we made a correlation matrix Figure 2, *Correlation Matrix of Predictors,* which showed us the following:

The predictors that may suffer from the issue of multicollinearity. Unbound Sulfur Dioxide and Total Sulfur Dioxide may suffer from multicollinearity due to their near-perfect correlation. Density and Alcohol may suffer from multicollinearity due to their negative correlation. Residual Sugar and Unbound Sulfur Dioxide may suffer from multicollinearity due to their positive correlation. Residual Sugar and Total Sulfur Dioxide may suffer from multicollinearity due to their positive correlation. Citric Acid and pH may suffer from multicollinearity due to their negative correlation. Fixed Acid and Unbound Sulfur Dioxide may suffer from multicollinearity due to their negative correlation. Fixed acid and Total Sulfur Dioxide may suffer from multicollinearity due to their negative correlation. Volatile Acid and Unbound Sulfur Dioxide may suffer from multicollinearity due to their negative correlation. Volatile Acid and Total Sulfur Dioxide may suffer from multicollinearity due to their negative correlation.

As seen in Figures 5-16, *Box Plot for "variable" Grouped by Wine Quality,* we created boxplots separated by the red wine data, on the left, and white on the right. We wanted to see if there were initial patterns that we could notice between the quality of the wine and every other variable, individually. These patterns are as follows:

For red wines, a decrease in volatile acidity is associated with higher wine quality. For white wines, volatile acidity remains relatively consistent across wine quality. For red wines, an increase in citric acid is associated with higher quality. For white wines, a slight increase in citric

acid is associated with higher quality. For red wines, residual sugar remains relatively constant across wine quality. For white wines, a decrease in residual sugar is associated with higher wine quality. For red wines, a decrease in Chlorides is associated with higher wine quality. For white wines, a slight decrease in Chlorides is associated with higher wine quality. For red wines, Unbound Sulfur Dioxide is relatively constant across wine quality. For white wines, Unbound Sulfur Dioxide is relatively constant across wine quality. For white wines, Total Sulfur Dioxide is relatively constant across wine quality. For red wines, a slight decrease in density is associated with higher wine quality. For white wines, a decrease in density is associated with higher wine quality. For red wines, a decrease in pH is associated with higher wine quality. For white wines, an increase in pH is associated with higher wine quality. For red wines, a strong increase in Alcohol is associated with higher wine quality. For white wines, a strong increase in Alcohol is associated with higher wine quality. For red wines, an increase in sulphates is associated with higher wine quality. For white wines, sulphates remain relatively constant across all wine qualities.

Methods

Multiple Linear Regression

The first analysis technique used was multiple linear regression. It was selected as all independent variables were continuous and the dependent variable was on a scale. The dependent variable for this model is Quality, describing the wine sample's quality on a scale of 0 to 10. Before proceeding, we ensure that the data meets the assumptions for linear regression by fitting a model on our training data for our full dataset, as displayed in the following diagnostic plots in Figure 17, *Diagnostic Plots for our Linear Model*.

1. We see that the residual plot does not show a fitted pattern. That is, the red line is approximately horizontal at zero. We do see various lines throughout the data but that is due to the nature of our quality being discrete integers for our training data but our predictions being continuous.
2. Our normal Q-Q plot is relatively linear, allowing us to assume normality.
3. The plot of our standardized residuals and our fitted values has a very clear pattern in its points,

however, our line seems relatively straight. We will again assume this is due to the discrete nature of our training data.

4. Our Residuals vs. Leverage plot is useful in identifying outliers in our data, but we see a relatively straight red line, and an outlier at index 2774 which may be influencing our model.

These plots were fit for our segmented red and white models as well, as seen in Figures 20, *Diagnostic Plots for Red Segmented Linear Model* and 23, *Diagnostic Plots for Segmented White Linear Model*. We came to similar conclusions, allowing us to fit a linear regression model for all 3 datasets.

First, we fitted a multiple linear regression model using both red and white wines, where its output is displayed in Figure 18, *Summary of our Grouped Linear Model*. This model has a low mean square error of 0.5312275. Moreover, all variables except chlorides and citric acid were found to be significant. However, this model has a very low adjusted R^2 of 0.2985, indicating that only 29.85% of the variance in quality can be explained by the predictors. As we noticed in the exploratory data analysis, there are differing distributions of various features for reds and whites. Therefore, we ran the regression on each wine type to determine if the model performance improved.

Running the regression for white wines, we find that the significant predictors, where the p-value is < 0.05, are volatile acidity, residual sugar, density, pH, sulphates, and alcohol as seen in Figure 24, *Summary of our White Linear Model*. This model has a slightly higher mean square error of 0.5553886 and again, a very low adjusted R^2 of 0.282, indicating that only 28.2% of variance in quality can be explained by the predictors. For the red wine linear regression, the significant predictors, where the p-value is less than 0.5, are volatile acidity, free sulfur dioxide, sulphates, and alcohol, as displayed in Figure 21, *Summary of our Red Linear Model*. The model has a slightly lower mean square error of 0.4172201 and a slightly higher adjusted R^2 of 0.3443, showcasing that the predictors can explain 34.43% of the variance in quality.

Best Subset Selection

We currently have a large number of parameters, so we will utilize best subset selection to see if a sparse model may be more effective in accurately predicting wine quality. This may occur if certain variables are redundant in our models

From Figure 26, *Criterion for BSS Variable Selection,* we see that the ideal number of variables in our model is 10 as the rate of change decreases greatly as we pass 10 towards 12. The variables we will use in our model, as deemed significant in the best subset selection, are type, fixed acidity, volatile acidity, residual sugar, free sulfur dioxide, total sulfur dioxide, density, pH, and sulphates. After running a linear regression using these variables, we see an increase in MSE with a value of 0.54. We will not utilize the best subset model in this scenario.

We ran a best subsets selection algorithm on our red and white models as well, and our results were as follows and are available in the appendix as Figures 27, *Criterion for BSS Variable Selection with Red Model* & 28, *Criterion for BSS Variable Selection with White Model*:

For our red model, we can clearly see that 6 variables will produce the best results in this case, and fit the model using the following variables: volatile acidity, chlorides, total sulfur dioxide, pH, sulphates, and alcohol. We found a test MSE of 0.41, which is slightly lower than our full model.

Through our plots for white wine we see our best number of variables being 8, and we fit our linear model using the following variables: fixed acidity, volatile acidity, residual sugar, free sulfur dioxide, density, pH, sulphates, alcohol. Our test MSE was higher than any of our other MSE's with a value of 0.60.

Regression Trees

The second method we used was regression trees. We followed the same pattern of attempting this on our full dataset first, and then on our segmented data sets of red and white wines. Our

results were as follows:

For our full dataset, our tree utilized only volatile acidity and alcohol, and the tree is available for viewing as Figure 29, *Regression Tree Visualized for Full Model,* in the appendix. This method gave us a surprisingly small test MSE for only using two predictor variables with a value of 0.56. We then analyzed our segmented sets as follows. We saw a much more complex tree when using our red dataset (see *Figure 30, Regression Tree Visualized for Red Model*), and arrived at a test MSE of 0.49. Our white dataset's tree (see *Figure 31, Regression Tree Visualized for White Model*) looked similar to that of our red data, but boasted a higher MSE with a value of 0.54.

K Nearest Neighbor

First, we decided to run 10-fold LOOCV On the Training Dataset to ensure that our K parameter for the KNN algorithm can be optimized. From the results of our LOOCV, it seems that K=9 is the value that optimizes our algorithm

See Figure 32 [KNN 10-fold Cross Validation]

Following implementation of our (K = 9) K-nearest neighbor algorithm, we receive a testing accuracy of 0.545 for predicting wine quality from the other 12 predictors (including wine type) in our dataset.

Results

As we compare each of our models, we have found that the model which performed the best on our data based on either its MSE or its Test Accuracy was our Grouped Model before utilizing Best Subset Selection. Almost all of our models floated around 0.5 for test MSE which lets us know that we might find similar results for any regression model unless we explore the data further, applying transformations to either our data itself or our models, such as trying a polynomial regression function.

When tackling the issue as a classification problem, we found a relatively low test accuracy, which indicates that although our response variable is technically discrete, we may find better results in using a regression model.

Conclusion

Our analysis of the wine quality dataset revealed that the models explored provided a basic understanding of the factors influencing quality. From the extensive exploration and analysis of the physicochemical attributes impacting wine quality, several key factors emerged. Separate models for red and white wines performed slightly better, highlighting the unique factors influencing each type. Certain attributes like volatile acidity, chlorides, and sulfur dioxide exhibited differing patterns across wine types concerning quality. However, even these models struggled to capture the full complexity, leaving a significant portion of the variance unexplained. This suggests that beyond the physicochemical properties we examined, other factors like tasting notes, production methods, and even external factors like climate likely play a crucial role. Future research incorporating these additional dimensions alongside non-linear regression techniques like random forests or support vector machines holds immense promise for unlocking the intricate secrets of wine quality. By delving deeper, we can benefit not only winemakers in their quest for crafting exceptional vintages, but also empower consumers to make informed choices based on their individual preferences.

References

Cortez, P., Cerdeira, A., Almeida, F., & Reis, J. (2009, June 9). Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems. https://www.sciencedirect.com/science/article/pii/S0167923609001377

Karlsson, P. and B. (2022, November 9). Wine consumption in the World 2020 in decline, a detailed look. Forbes. https://www.forbes.com/sites/karlsson/2021/12/31/wine-consumption-in-the-world-2020-in-decline-a-detailed-look/?sh=4b440c583f71

Scott, M. (2022, September 14). Hard-hit by climate change, winemakers turn to sustainability to ride the storms. Reuters. https://www.reuters.com/business/sustainable-business/hard-hit-by-climate-change-wine makers-turn-sustainability-ride-storms-2022-09-14/

Sinkler, S. (2020, March 2). How to swirl, sniff and SIP like a pro. thewineshack.wine. https://www.thewineshack.wine/how-to-swirl-sniff-and-sip-like-a-pro/

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. ISSN: 0167-9236.
[@Elsevier] http://dx.doi.org/10.1016/j.dss.2009.05.016
[Pre-press (pdf)] http://www3.dsi.uminho.pt/pcortez/winequality09.pdf
[bib] http://www3.dsi.uminho.pt/pcortez/dss09.bib

Code and Figures for Final Project

Data Description and Preproccessing

```
library(readr)
library(kableExtra)
library(MASS)
library(ordinal)

set.seed(1)
# Loading Data
wine.dat <- read_csv("wine-quality-white-and-red.csv")

## Rows: 6497 Columns: 13
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (1): type
## dbl (12): fixed acidity, volatile acidity, citric acid, residual sugar,
chlo...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.

wine.dat$type[wine.dat$type == "red"] <- 0
wine.dat$type[wine.dat$type == "white"] <- 1
red.dat <- wine.dat[wine.dat$type == 0, ]
white.dat <- wine.dat[wine.dat$type == 1, ]

"kable(head(wine.dat)) %>%
  kable_styling(full_width = FALSE)
red.dat <- wine.dat[which(wine.dat[,1]==0),]
white.dat<- wine.dat[which(wine.dat[,1]==1),]"

## [1] "kable(head(wine.dat)) %>%\n  kable_styling(full_width =
FALSE)\nred.dat <- wine.dat[which(wine.dat[,1]==0),]\nwhite.dat<-
wine.dat[which(wine.dat[,1]==1),]"

n <- nrow(wine.dat)
n.red<- nrow(red.dat)
n.white <-nrow(white.dat)

train.red <- sample(n.red, n.red/2)
train.white<-sample(n.white, n.white/2)

train.data.red <- red.dat[train.red,]
test.data.red<- red.dat[-train.red,]
```

```
train.data.white <-white.dat[train.white,]
test.data.white <-white.dat[-train.white,]

fin.train.data<- rbind(train.data.red,train.data.white)
fin.test.data<- rbind(test.data.red, test.data.white)

train.data.red <-train.data.red[,-1]
test.data.red <-test.data.red[,-1]
train.data.white <-train.data.white[,-1]
test.data.white <-test.data.white[,-1]
```

Figure 1: Head of the Full Dataset

Exploratory Data Analysis

```
library(ggplot2)
numeric_df <- wine.dat[, sapply(wine.dat, is.numeric)]
correlation <- cor(numeric_df)
pairs(correlation)
```

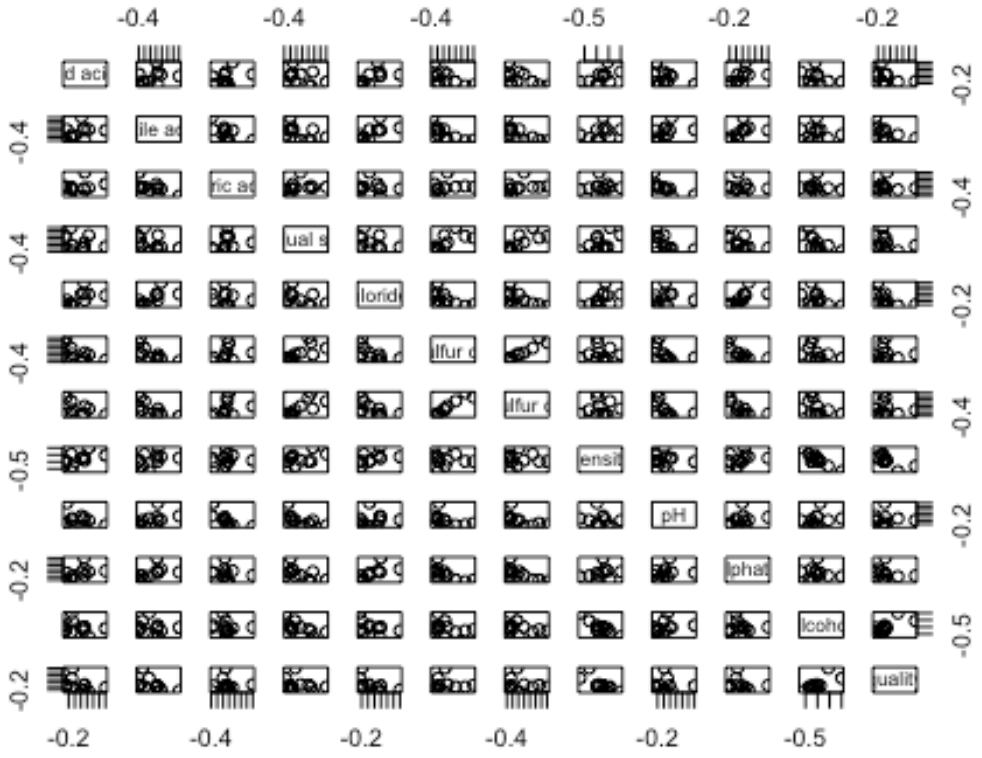

Figure 2: Correlation Matrix of Predictors

```r
ggplot(wine.dat, aes(x = type, y = quality)) +
  geom_boxplot() +
  labs(title = "Boxplot of Type of Wine by Wine Quality")
```

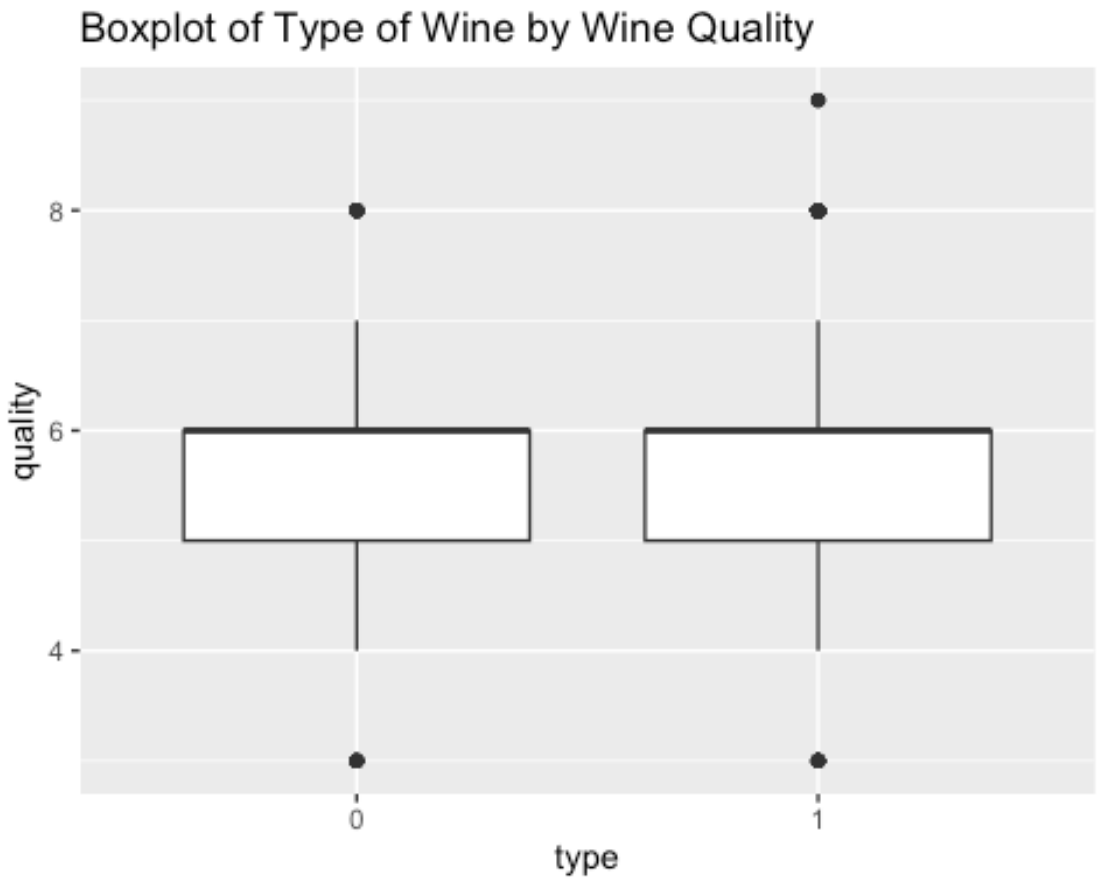

Figure 3: Boxplot of Wine Grouped by Wine Type

```r
ggplot(wine.dat, aes(x = type, y = quality, color = type)) +
  geom_jitter(width = 0.3, alpha = 0.6) +
  scale_color_manual (values = c("red" = "red", "white" = "yellow")) +
  labs (title = "Wine Type vs. Quality", x = "Wine Type", y = "Quality") +
  theme_minimal()
```

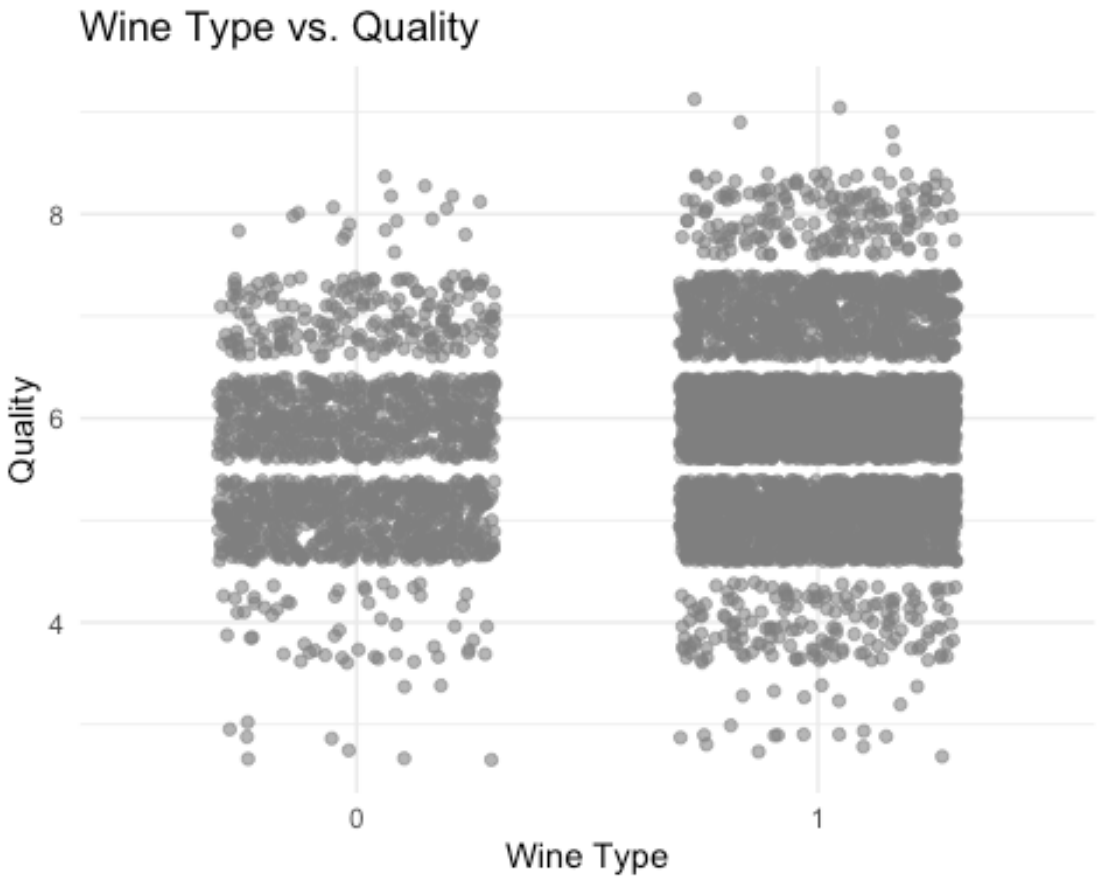

Figure 4: Jitter Plot Grouped by Wine Type

```
ggplot(wine.dat, aes(x = as.factor(quality), y = `fixed acidity`, fill =
as.factor(quality))) +
  geom_boxplot() +
  facet_wrap(~ type, nrow = 1) +
  labs(title = "Fixed Acidity by Wine Quality and Type", x = "Wine Quality",
y = "Fixed Acidity") +
  theme_minimal()
```

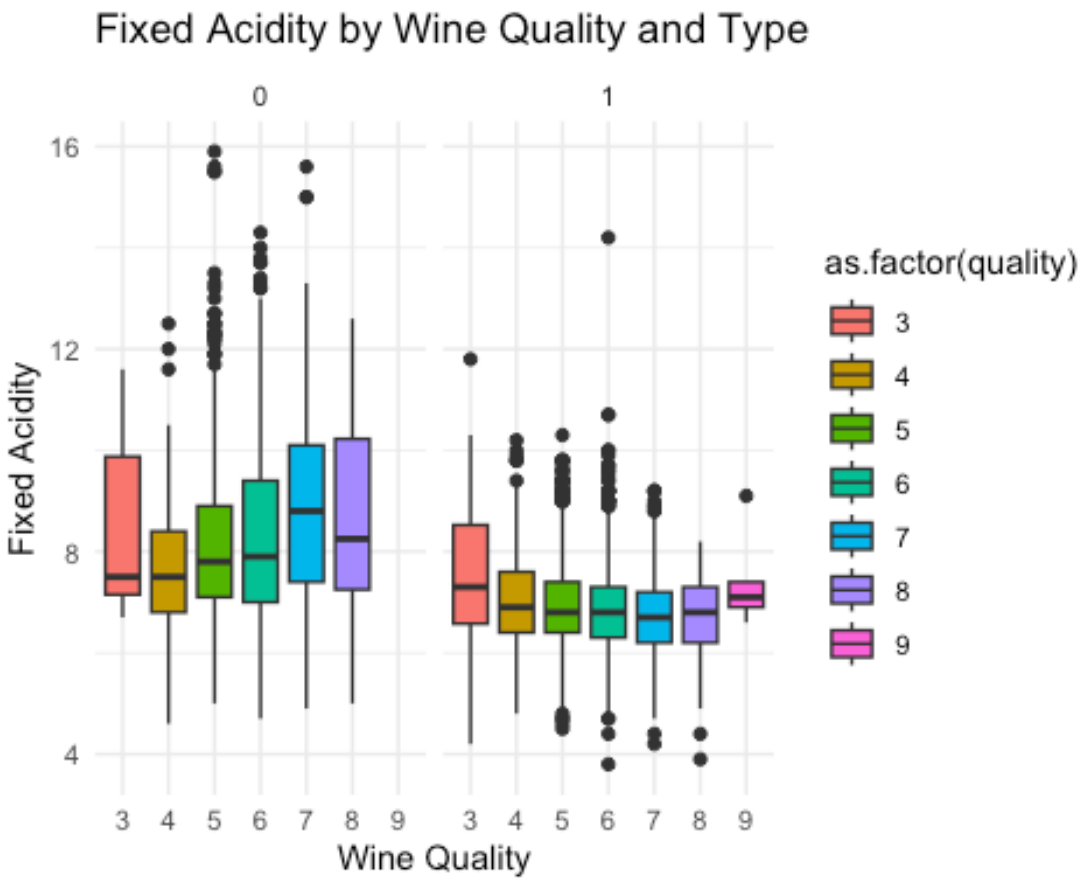

Figure 5: Box Plot for Fixed Acidity Grouped by Wine Quality

```
ggplot(wine.dat, aes(x = as.factor(quality), y = `volatile acidity`, fill =
as.factor(quality))) +
  geom_boxplot() +
  facet_wrap(~ type, nrow = 1) +
  labs(title = "Volatile Acidity by Wine Quality and Type", x = "Wine
Quality", y = "Volatile Acidity") +
  theme_minimal()
```

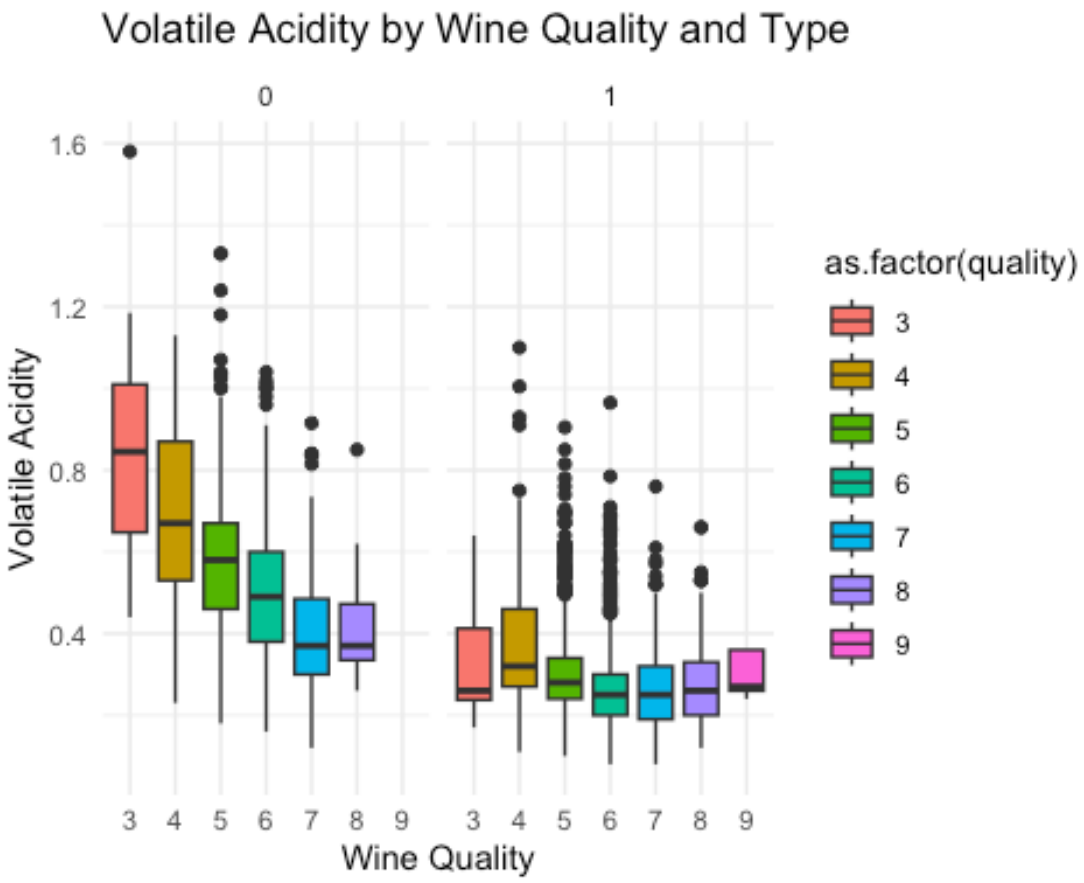

Figure 6: Box Plot for Volatile Acidity Grouped by Wine Quality

```
ggplot(wine.dat, aes(x = as.factor(quality), y = `citric acid`, fill =
as.factor(quality))) +
  geom_boxplot() +
  facet_wrap(~ type, nrow = 1) +
  labs(title = "Citric Acid by Wine Quality and Type", x = "Wine Quality", y
= "Citric Acid") +
  theme_minimal()
```

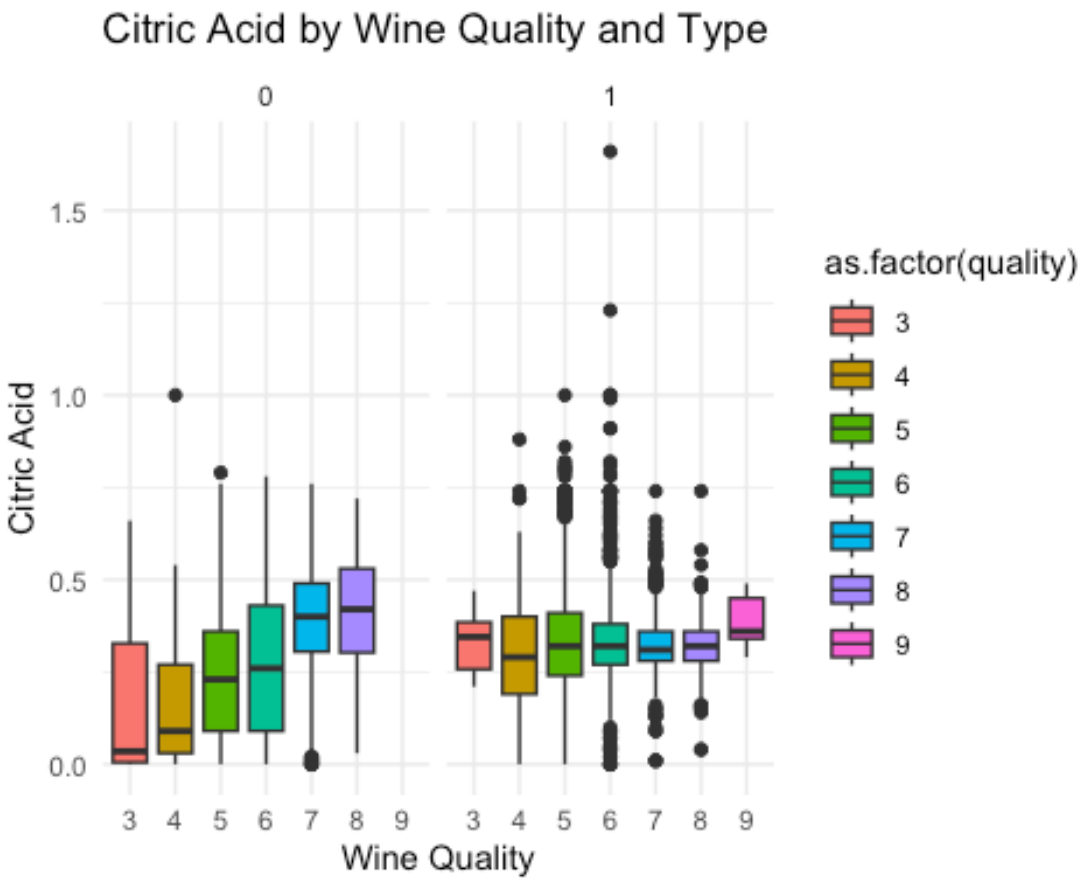

Figure 7: Box Plot for Citric Acid Grouped by Wine Quality

```
ggplot(wine.dat, aes(x = as.factor(quality), y = `residual sugar`, fill =
as.factor(quality))) +
  geom_boxplot() +
  facet_wrap(~ type, nrow = 1) +
  labs(title = "Residual Sugar by Wine Quality and Type", x = "Wine Quality",
y = "Residual Sugar") +
  theme_minimal()
```

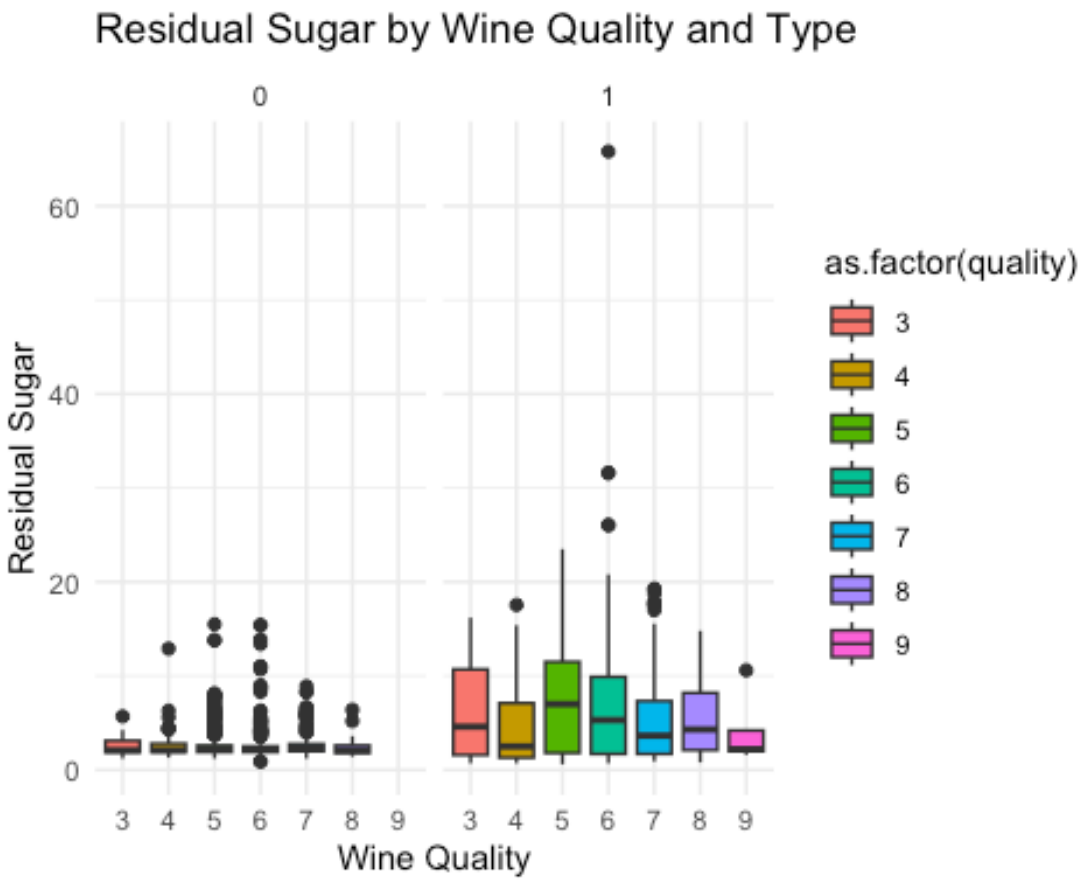

Figure 8: Box Plot for Residual Sugar Grouped by Wine Quality

```
ggplot(wine.dat, aes(x = as.factor(quality), y = `fixed acidity`, fill =
as.factor(quality))) +
  geom_boxplot() +
  facet_wrap(~ type, nrow = 1) +
  labs(title = "Fixed Acidity by Wine Quality and Type", x = "Wine Quality",
y = "Fixed Acidity") +
  theme_minimal()
```

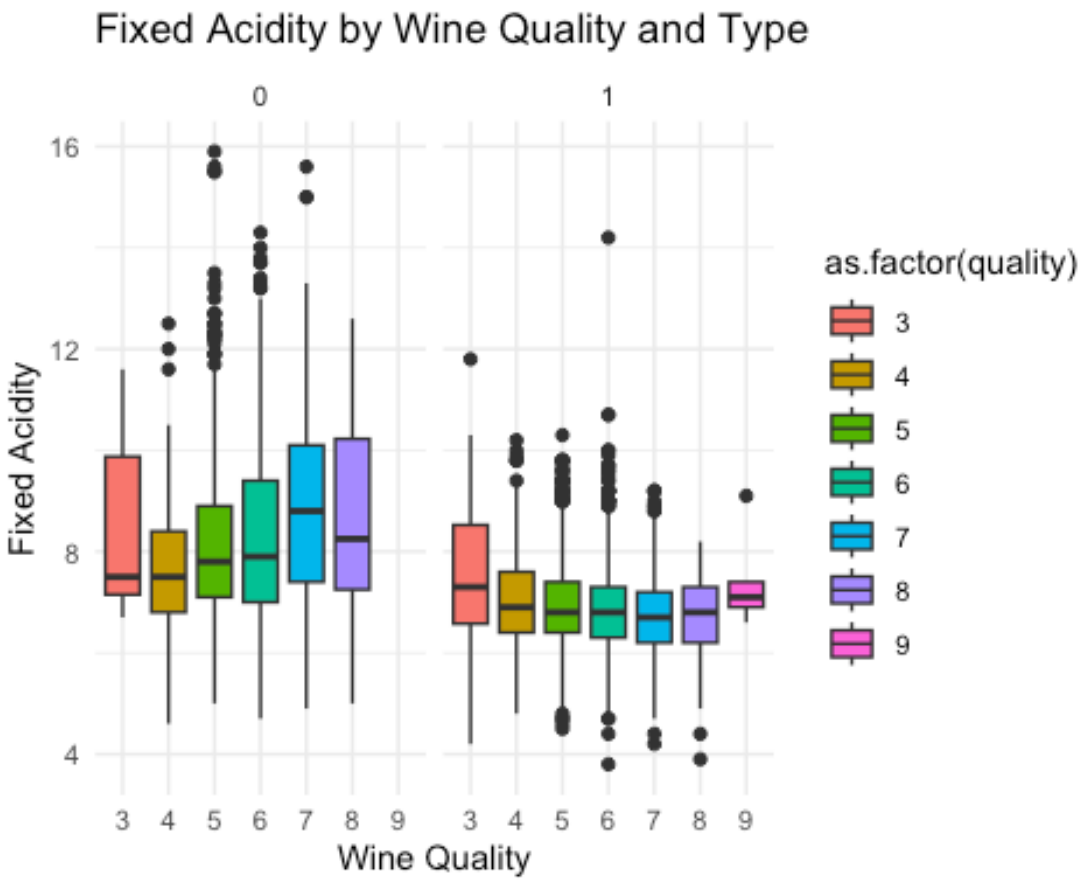

Figure 9: Box Plot for Fixed Acidity Grouped by Wine Quality

```
ggplot(wine.dat, aes(x = as.factor(quality), y = chlorides, fill =
as.factor(quality))) +
  geom_boxplot() +
  facet_wrap(~ type, nrow = 1) +
  labs(title = "Chlorides by Wine Quality and Type", x = "Wine Quality", y =
"Chlorides") +
  theme_minimal()
```

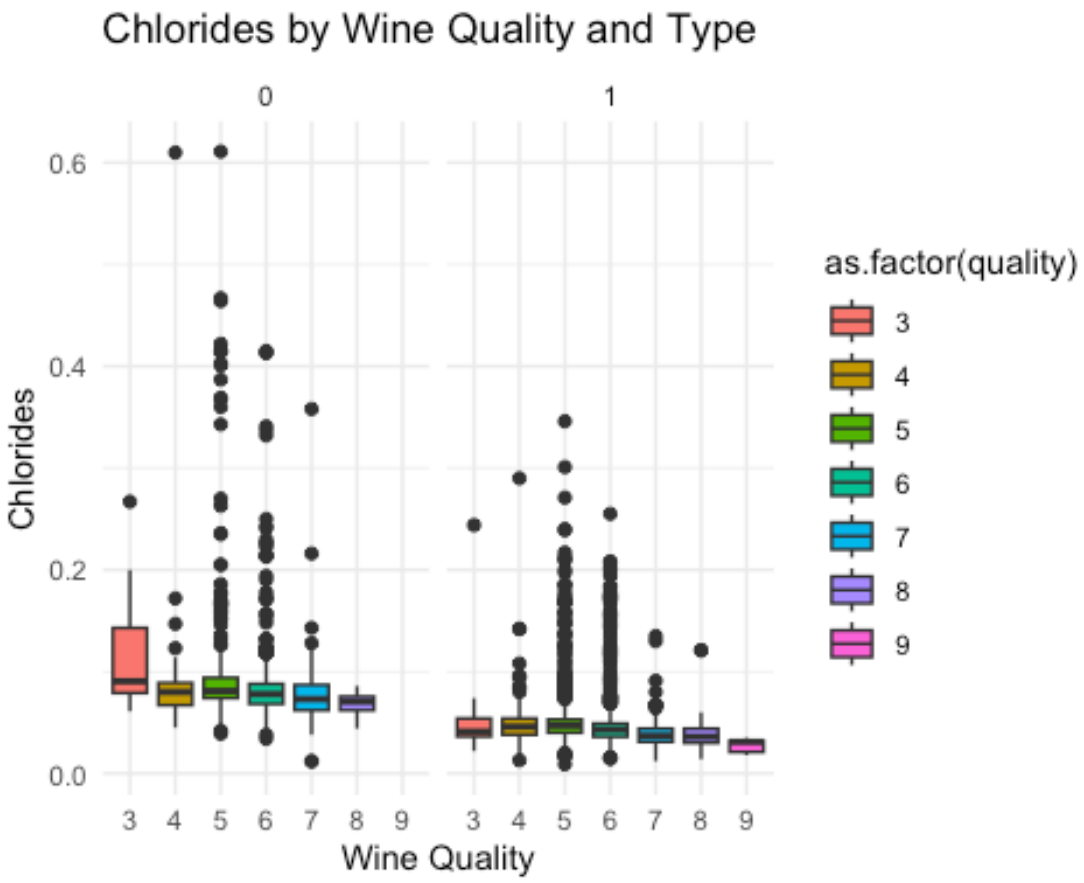

Figure 10: Box Plot for Chloride Grouped by Wine Quality

```
ggplot(wine.dat, aes(x = as.factor(quality), y = `free sulfur dioxide`, fill
= as.factor(quality))) +
  geom_boxplot() +
  facet_wrap(~ type, nrow = 1) +
  labs(title = "Free Sulfur Dioxide by Wine Quality and Type", x = "Wine
Quality", y = "Free Sulfur Dioxide") +
  theme_minimal()
```

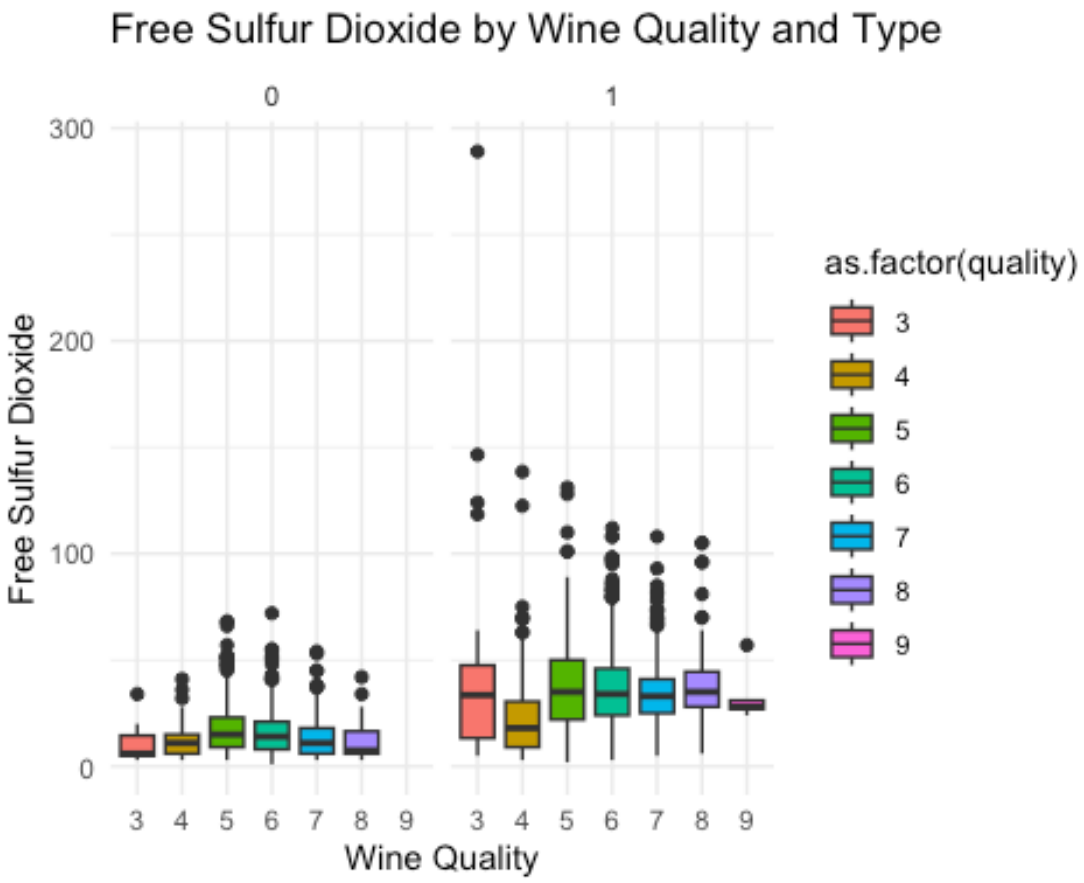

Figure 11: Box Plot for Free Sulfur Dioxide Grouped by Wine Quality

```
ggplot(wine.dat, aes(x = as.factor(quality), y = `total sulfur dioxide`, fill
= as.factor(quality))) +
  geom_boxplot() +
  facet_wrap(~ type, nrow = 1) +
  labs(title = "Total Sulfur Dioxide by Wine Quality and Type", x = "Wine
Quality", y = "Total Sulfur Dioxide") +
  theme_minimal()
```

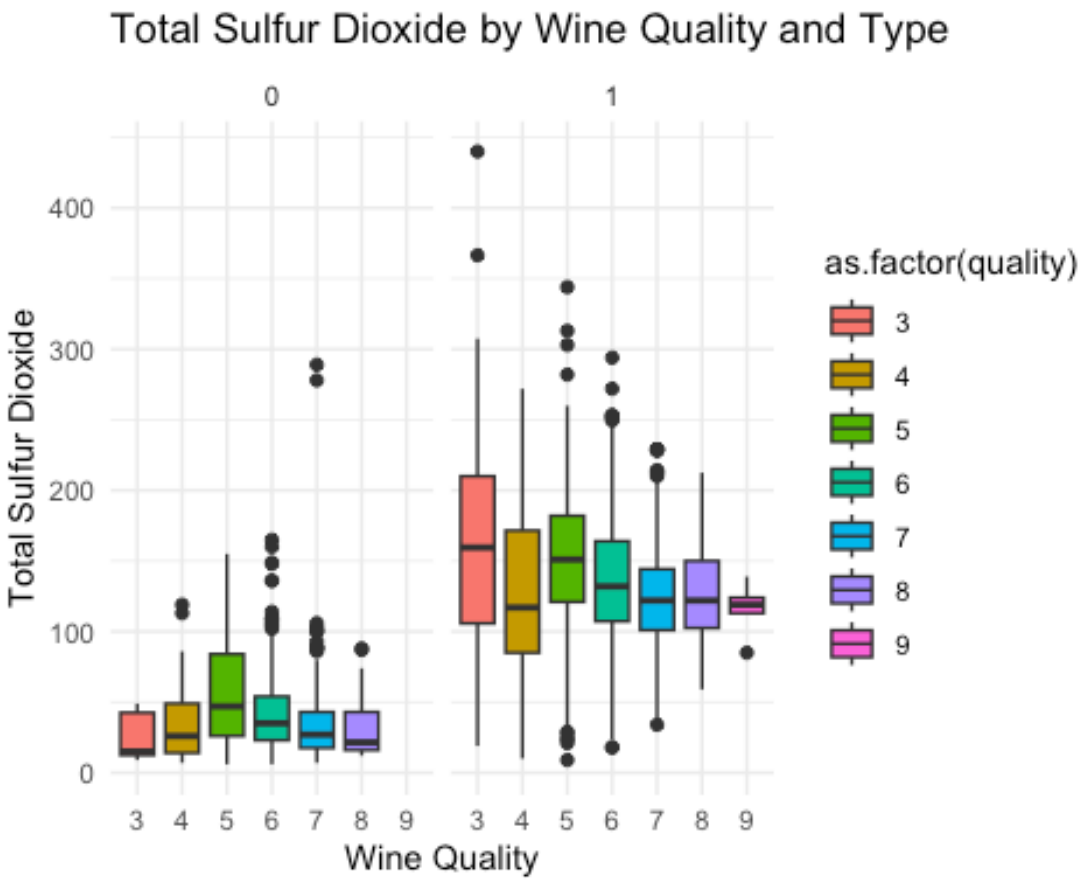

Figure 12: Box Plot for Total Sulfur Dioxide Grouped by Wine Quality

```
ggplot(wine.dat, aes(x = as.factor(quality), y = density, fill =
as.factor(quality))) +
  geom_boxplot() +
  facet_wrap(~ type, nrow = 1) +
  labs(title = "Density by Wine Quality and Type", x = "Wine Quality", y =
"Density") +
  theme_minimal()
```

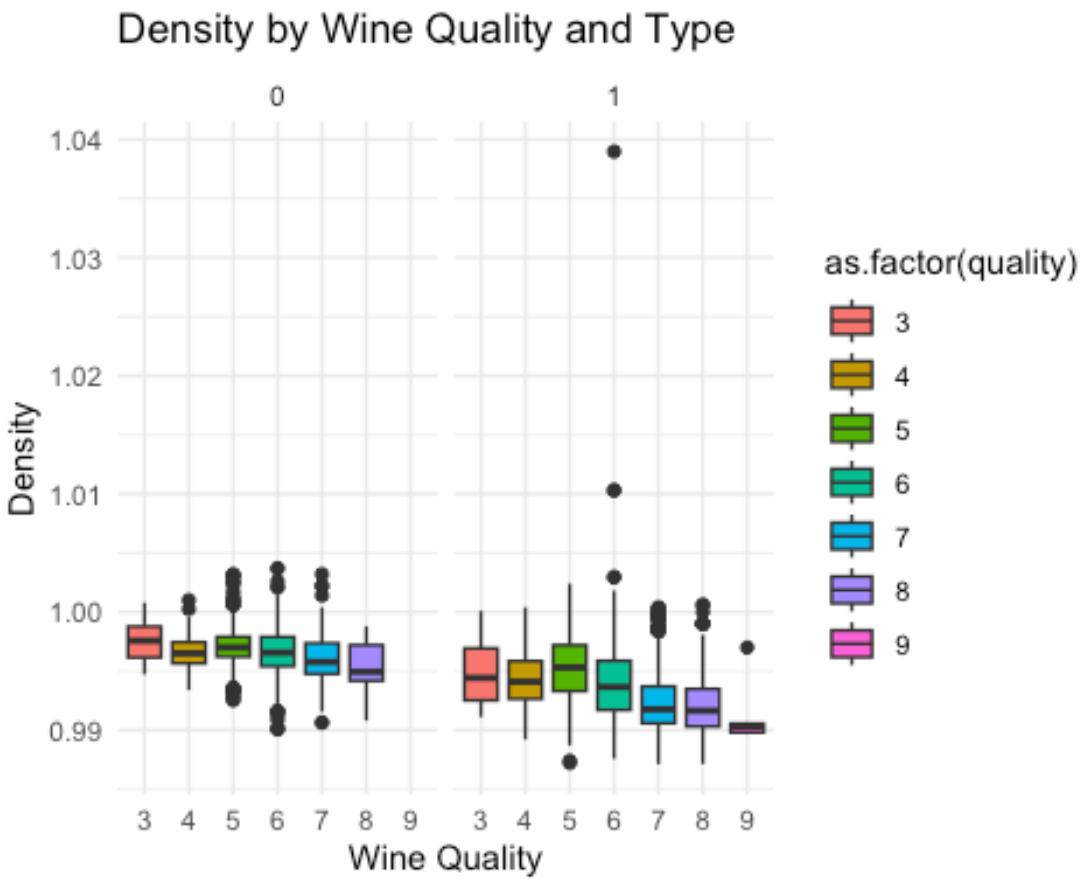

Figure 13: Box Plot for Density Grouped by Wine Quality

```
ggplot(wine.dat, aes(x = as.factor(quality), y = pH, fill =
as.factor(quality))) +
  geom_boxplot() +
  facet_wrap(~ type, nrow = 1) +
  labs(title = "pH by Wine Quality and Type", x = "Wine Quality", y = "pH") +
  theme_minimal()
```

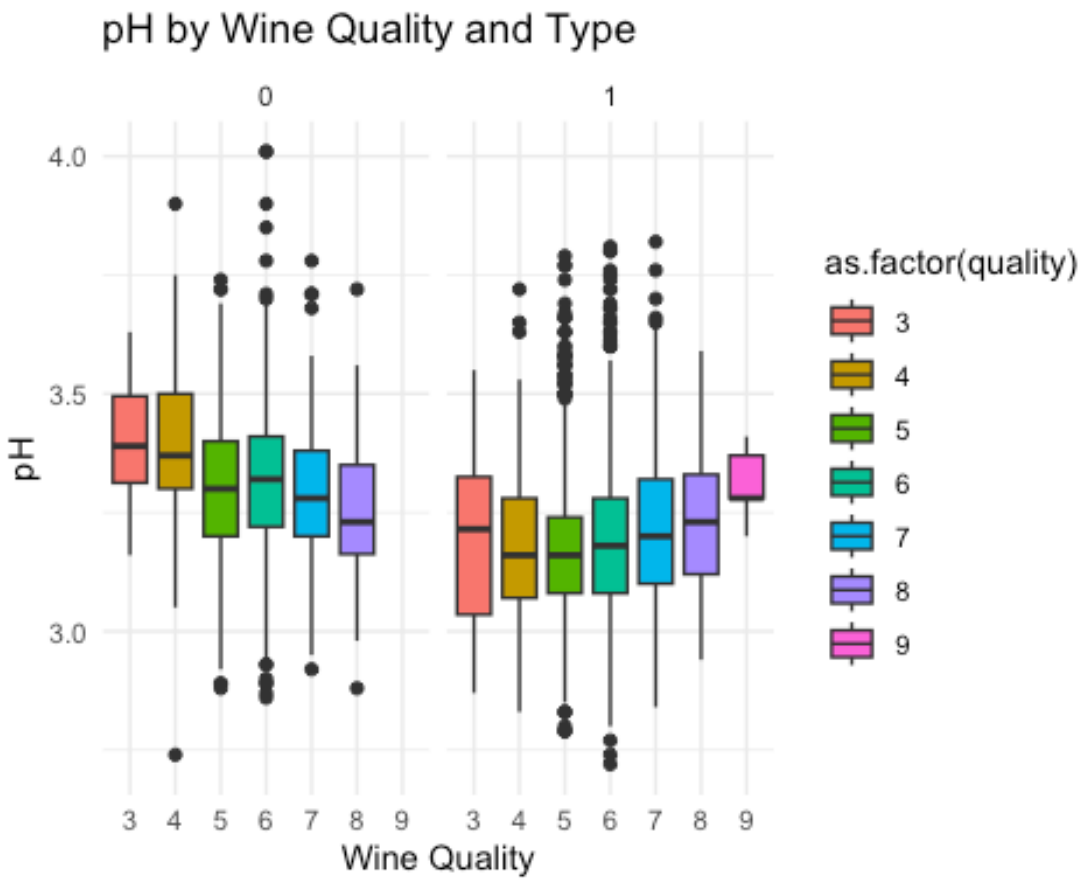

Figure 14: Box Plot for pH Grouped by Wine Quality

```
ggplot(wine.dat, aes(x = as.factor(quality), y = sulphates, fill =
as.factor(quality))) +
  geom_boxplot() +
  facet_wrap(~ type, nrow = 1) +
  labs(title = "Sulphates by Wine Quality and Type", x = "Wine Quality", y =
"Sulphates") +
  theme_minimal()
```

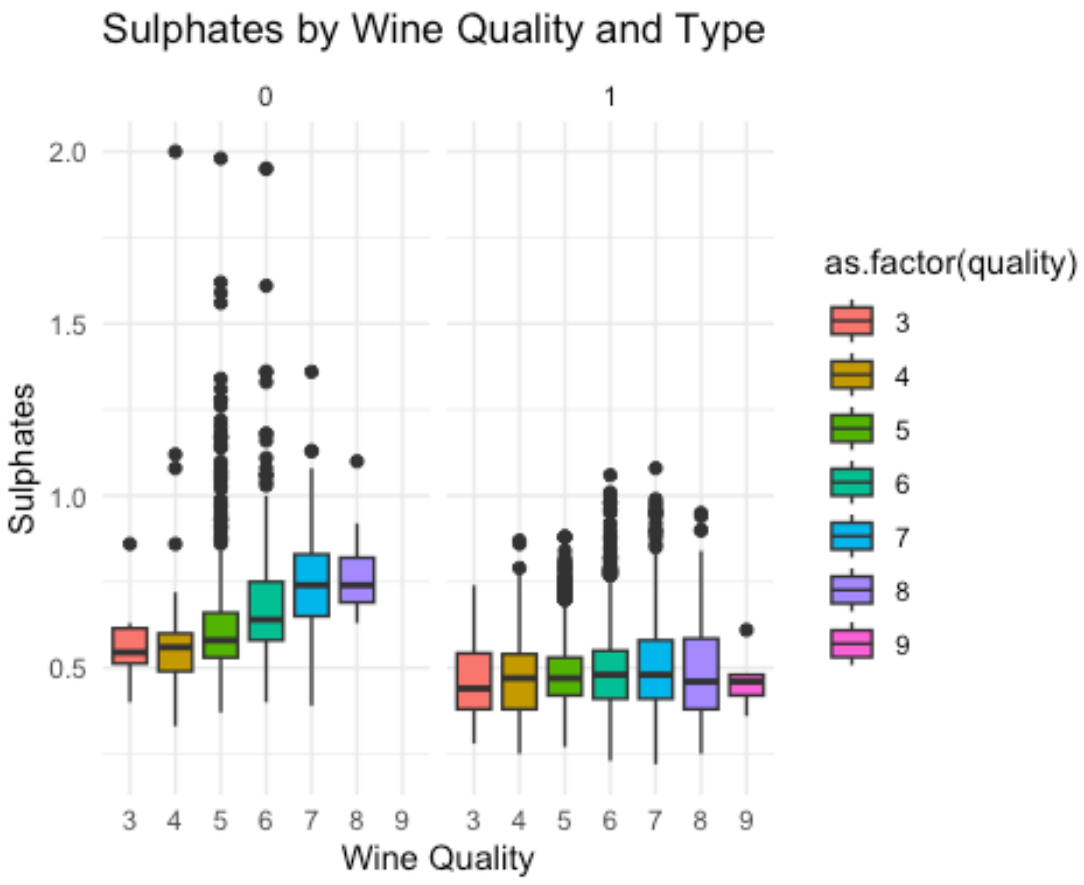

Figure 15: Box Plot for Sulphates Grouped by Wine Quality

```
ggplot(wine.dat, aes(x = as.factor(quality), y = alcohol, fill =
as.factor(quality))) +
  geom_boxplot() +
  facet_wrap(~ type, nrow = 1) +
  labs(title = "Alcohol by Wine Quality and Type", x = "Wine Quality", y =
"Alcohol") +
  theme_minimal()
```

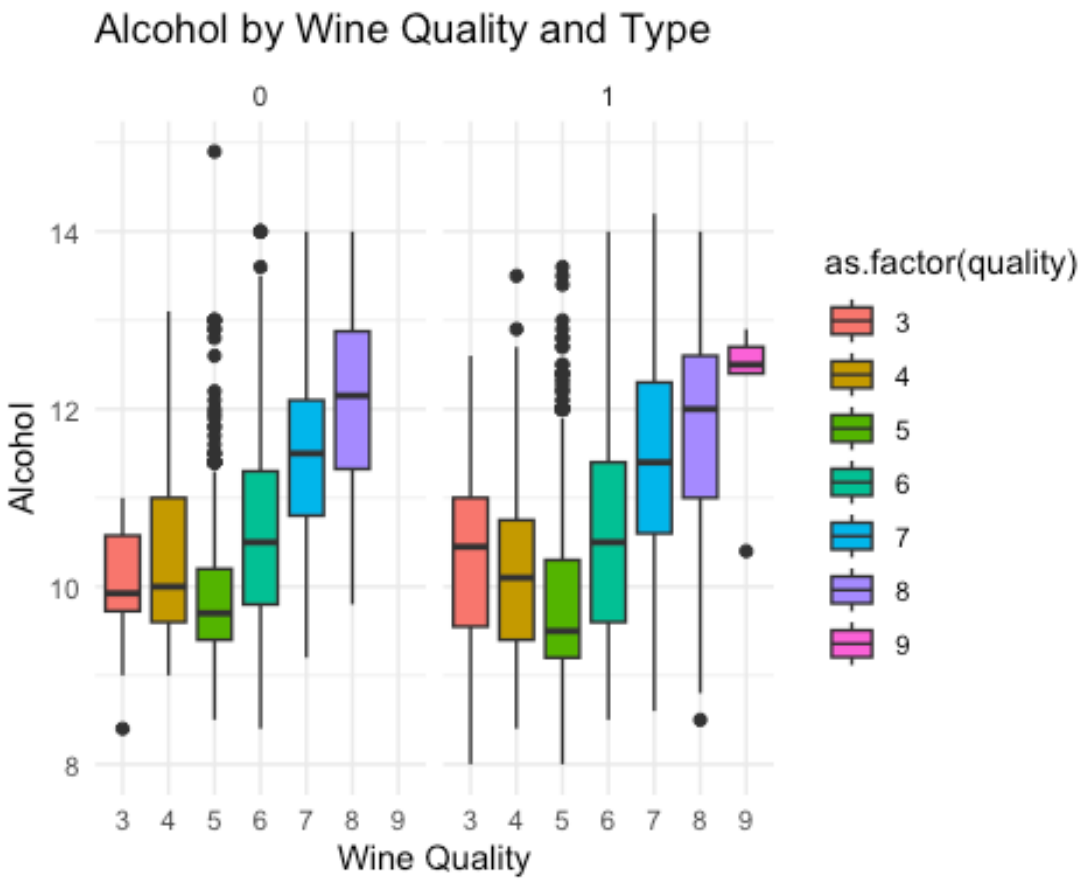

Figure 16: Box Plot for Alchohol Grouped by Wine Quality

Linear regression

```
#Dataset with both red and white
fin.train.data <- fin.train.data[-1993,]
lm.fit.both <- lm(quality ~ ., data = fin.train.data)
lm.both.sum<-summary(lm.fit.both)
lm.both.pred <- predict(lm.fit.both, fin.test.data)
lm.both.mse <- mean((lm.both.pred - fin.test.data$quality)^2)
lm.both.mse
```

```
## [1] 0.5312275
```

```
par(mfrow = c(2, 2))
plot(lm.fit.both)
```

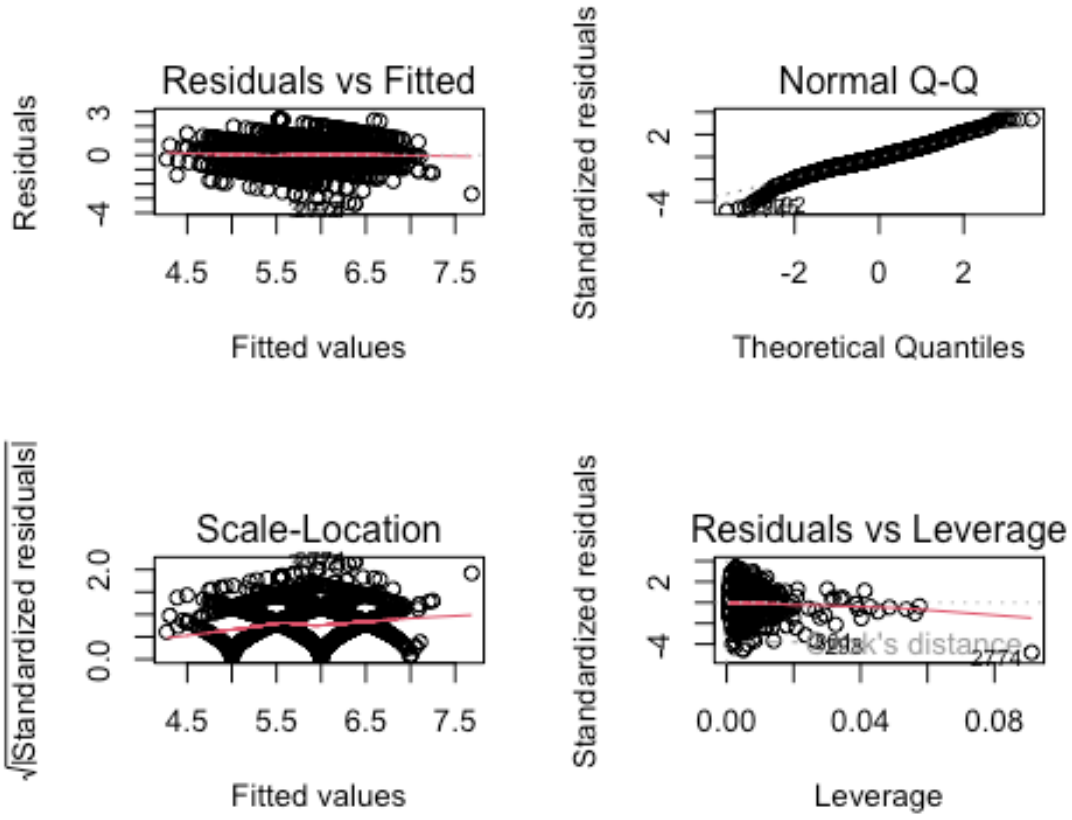

Figure 17: Diagnostic Plots for Grouped Linear Model

```
lm.both.sum

##
## Call:
## lm(formula = quality ~ ., data = fin.train.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3821 -0.4628 -0.0433  0.4468  2.4708
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.416e+02  2.181e+01   6.494 9.63e-11 ***
## type1                  -3.795e-01  8.392e-02  -4.523 6.32e-06 ***
## `fixed acidity`         1.069e-01  2.313e-02   4.623 3.94e-06 ***
## `volatile acidity`     -1.499e+00  1.163e-01 -12.885  < 2e-16 ***
## `citric acid`          -3.243e-02  1.127e-01  -0.288  0.77346
## `residual sugar`        7.645e-02  8.829e-03   8.660  < 2e-16 ***
## chlorides               1.941e-01  5.168e-01   0.376  0.70729
## `free sulfur dioxide`   3.397e-03  1.079e-03   3.147  0.00166 **
## `total sulfur dioxide` -1.410e-03  4.744e-04  -2.972  0.00298 **
```

```
## density                -1.413e+02  2.210e+01  -6.393 1.86e-10 ***
## pH                      6.264e-01  1.306e-01   4.798 1.68e-06 ***
## sulphates               7.334e-01  1.065e-01   6.884 6.95e-12 ***
## alcohol                 1.900e-01  2.748e-02   6.913 5.68e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7389 on 3234 degrees of freedom
## Multiple R-squared:  0.3011, Adjusted R-squared:  0.2985
## F-statistic: 116.1 on 12 and 3234 DF,  p-value: < 2.2e-16
```

Figure 18: Summary of our Grouped Linear Model

```
coefficients <- coef(lm.fit.both)[-1]
names(coefficients) <- names(lm.fit.both$coefficients)[-1]

barplot(coefficients, main = "Coefficients of Linear Regression Model for All
Wines",
        xlab = "Predictor Variables", ylab = "Increase in Quality per Unit
Increase",
        las = 2,  # Rotate x-axis labels by 90 degrees
        cex.names = 0.5,  # Adjust size of x-axis labels
        ylim = range(coefficients) * c(1.2, 1.3),  # Extend y-axis limits for
better spacing
        mar = c(5, 6, 4, 2)  # Adjust margin space: bottom, left, top, right
)
```

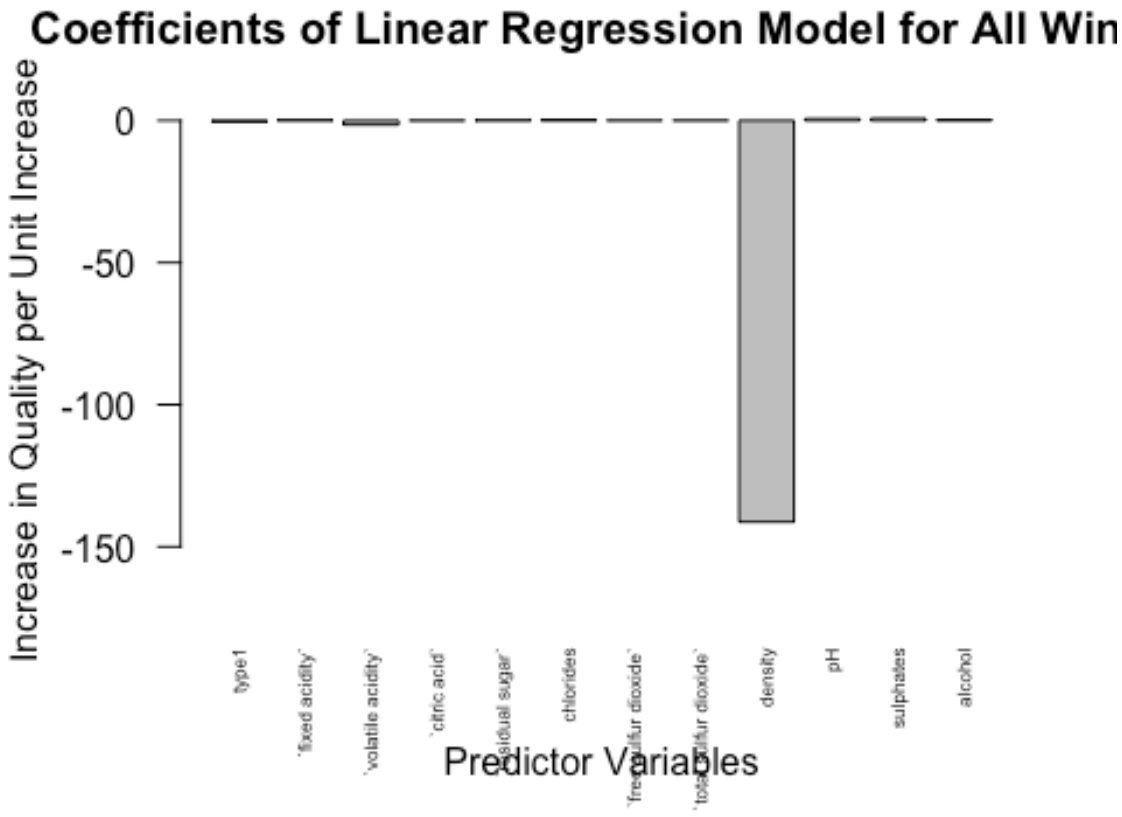

Figure 19: Plot of Coefficients for Grouped Linear Model

```
#Dataset with only red
lm.fit.red <- lm(quality ~ ., data = train.data.red)
lm.red.sum<-summary(lm.fit.red)
lm.red.pred <- predict(lm.fit.red, test.data.red)
lm.red.mse <- mean((lm.red.pred - test.data.red$quality)^2)
lm.red.mse
```

```
## [1] 0.4172201
```

```
par(mfrow = c(2, 2))
plot(lm.fit.red)
```

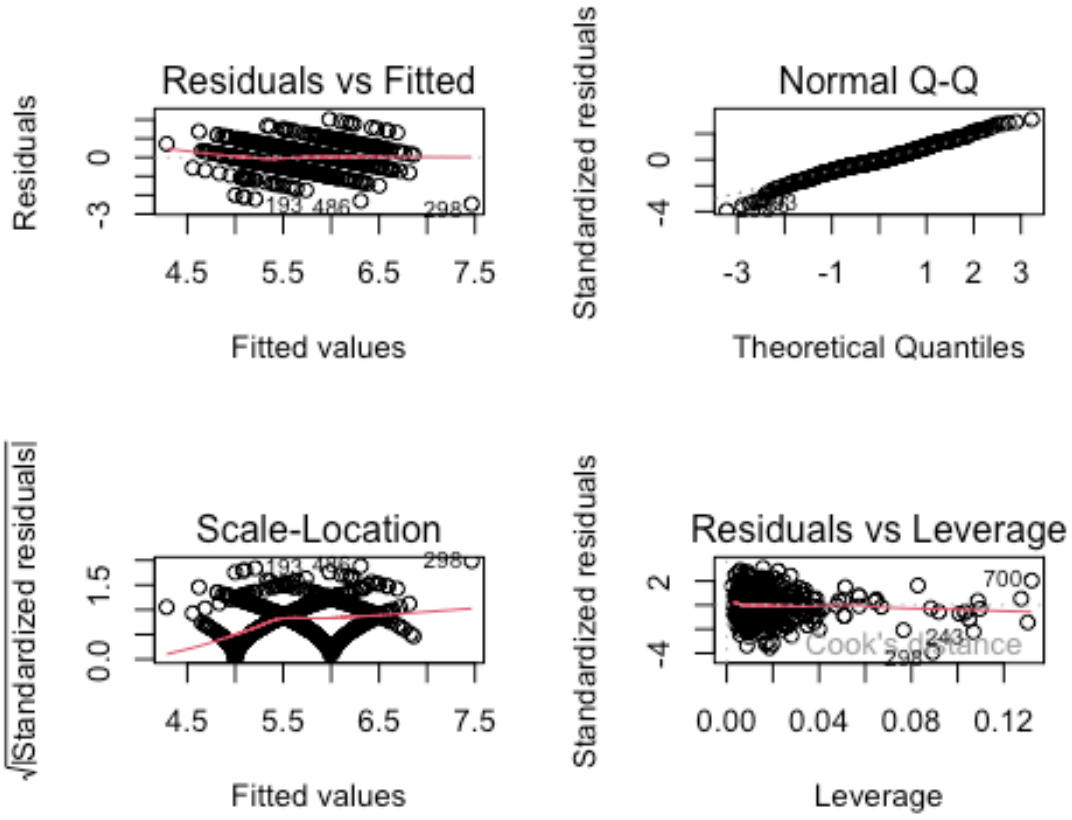

Figure 20: Diagnostic Plots for Red Segmented Linear Model

```
lm.red.sum

##
## Call:
## lm(formula = quality ~ ., data = train.data.red)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.46569 -0.36416 -0.03842  0.40413  2.01783
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             38.223749  29.372995   1.301 0.193529
## `fixed acidity`          0.038090   0.035736   1.066 0.286809
## `volatile acidity`      -1.156466   0.178125  -6.492 1.49e-10 ***
## `citric acid`           -0.186289   0.211867  -0.879 0.379521
## `residual sugar`         0.026133   0.020740   1.260 0.208028
## chlorides               -0.619347   0.677729  -0.914 0.361072
## `free sulfur dioxide`    0.004900   0.003247   1.509 0.131709
## `total sulfur dioxide`  -0.003773   0.001102  -3.424 0.000648 ***
## density                -34.706478  30.007968  -1.157 0.247797
```

```
## pH                    -0.264661   0.272789  -0.970 0.332243
## sulphates              0.860893   0.157445   5.468 6.12e-08 ***
## alcohol                0.260805   0.036803   7.087 3.06e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.656 on 787 degrees of freedom
## Multiple R-squared:  0.3534, Adjusted R-squared:  0.3443
## F-statistic:  39.1 on 11 and 787 DF,  p-value: < 2.2e-16
```

Figure 21: Summary of our Red Linear Model

```
coefficients <- coef(lm.fit.red)[-1]
names(coefficients) <- names(lm.fit.red$coefficients)[-1]

barplot(coefficients, main = "Coefficients of Linear Regression Model for Red
Wine",
        xlab = "Predictor Variables", ylab = "Increase in Quality per Unit
Increase",
        las = 2,  # Rotate x-axis labels by 90 degrees
        cex.names = 0.5,  # Adjust size of x-axis labels
        ylim = range(coefficients) * c(1.2, 1.3),  # Extend y-axis limits for
better spacing
        mar = c(5, 6, 4, 2)  # Adjust margin space: bottom, left, top, right
)
```

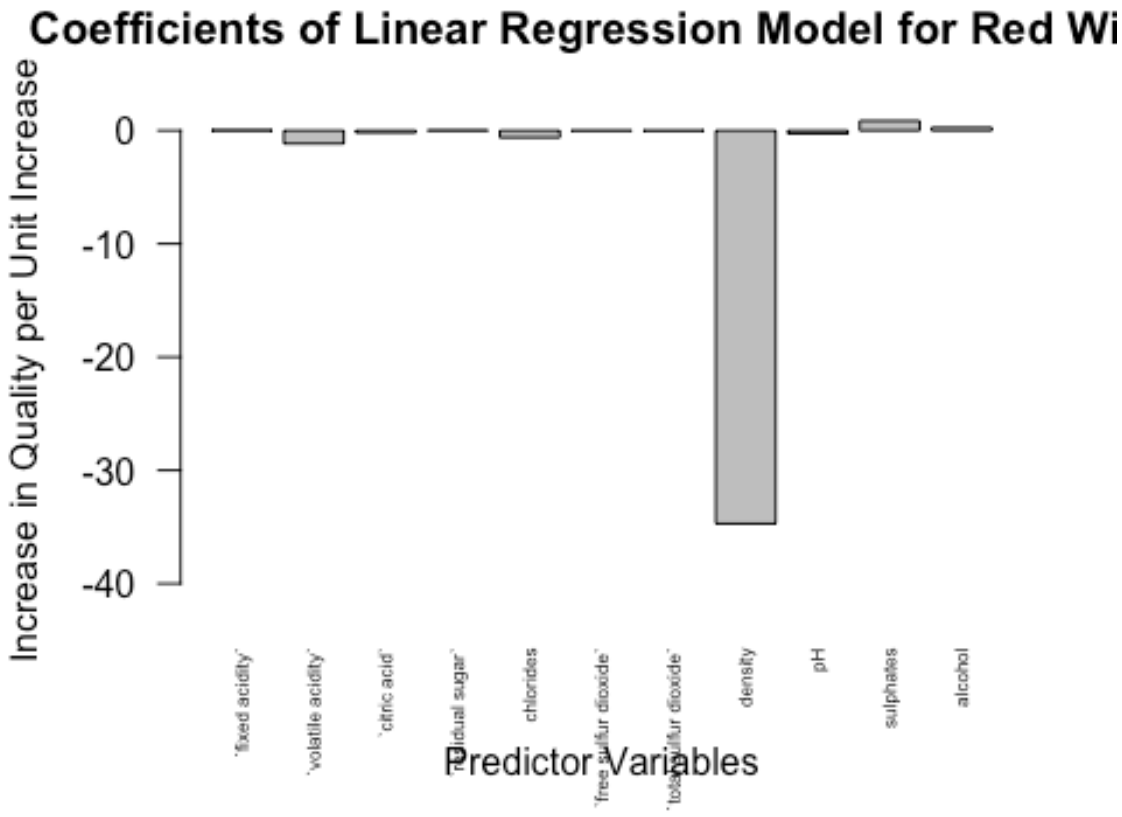

Figure 22: Plot of Coefficients for Segmented Red Linear Model

```
#Dataset with only white
lm.fit.white <- lm(quality ~ ., data = train.data.white)
lm.white.sum<-summary(lm.fit.white)
lm.white.pred <- predict(lm.fit.white, test.data.white)
lm.white.mse <- mean((lm.white.pred - test.data.white$quality)^2)
lm.white.mse

## [1] 0.5553886

par(mfrow = c(2, 2))
plot(lm.fit.white)
```

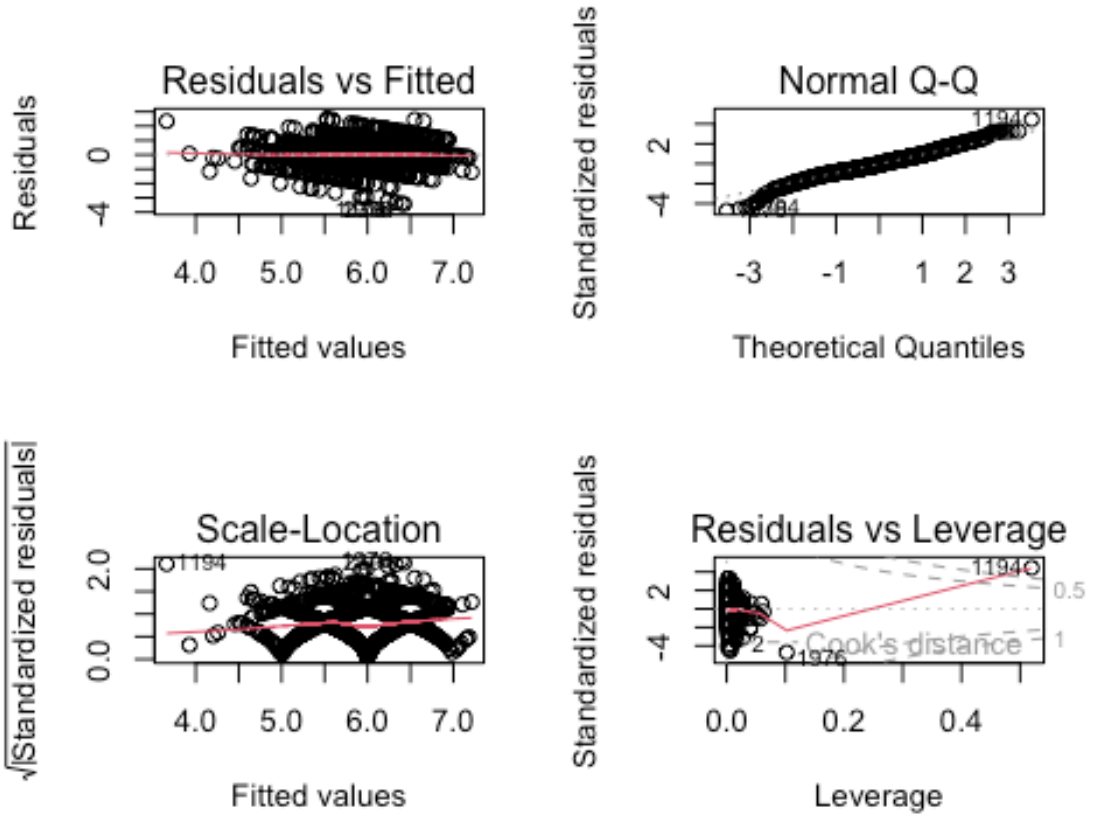

Figure 23: Diagnostic Plots for Segmented White Linear Model

```
lm.white.sum

##
## Call:
## lm(formula = quality ~ ., data = train.data.white)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4272 -0.4928 -0.0398  0.4506  2.4701
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.383e+02  2.354e+01   5.874 4.82e-09 ***
## `fixed acidity`        4.827e-02  2.781e-02   1.736   0.0828 .
## `volatile acidity`    -1.799e+00  1.607e-01 -11.196  < 2e-16 ***
## `citric acid`          4.293e-02  1.345e-01   0.319   0.7497
## `residual sugar`       7.989e-02  9.907e-03   8.064 1.15e-15 ***
## chlorides              1.882e-01  7.493e-01   0.251   0.8017
## `free sulfur dioxide`  2.276e-03  1.181e-03   1.927   0.0541 .
## `total sulfur dioxide` -4.427e-04  5.491e-04  -0.806   0.4202
## density                -1.382e+02  2.389e+01  -5.786 8.16e-09 ***
```

```
## pH                    6.386e-01  1.440e-01   4.434 9.68e-06 ***
## sulphates             5.566e-01  1.404e-01   3.964 7.57e-05 ***
## alcohol               2.191e-01  3.068e-02   7.142 1.21e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7592 on 2437 degrees of freedom
## Multiple R-squared:  0.2853, Adjusted R-squared:  0.282
## F-statistic: 88.42 on 11 and 2437 DF,  p-value: < 2.2e-16
```

Figure 24: Summary of our White Linear Model

```r
coefficients <- coef(lm.fit.white)[-1]
names(coefficients) <- names(lm.fit.white$coefficients)[-1]

barplot(coefficients, main = "Coefficients of Linear Regression Model for
White Wines",
        xlab = "Predictor Variables", ylab = "Increase in Quality per Unit
Increase",
        las = 2,  # Rotate x-axis labels by 90 degrees
        cex.names = 0.5,  # Adjust size of x-axis labels
        ylim = range(coefficients) * c(1.2, 1.3),  # Extend y-axis limits for
better spacing
        mar = c(5, 6, 4, 2)  # Adjust margin space: bottom, left, top, right
)
```

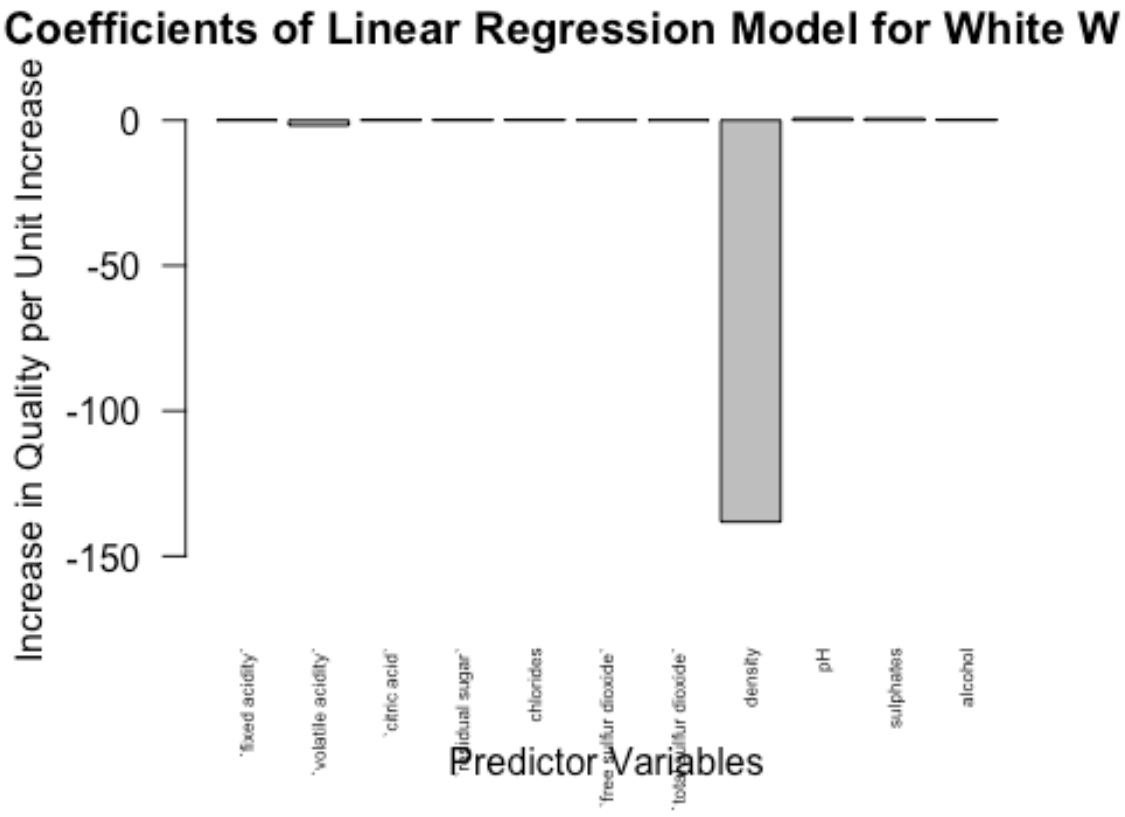

Figure 25: Plot of Coefficients for Segmented White Linear Model

Best Subset Selection

```
library(leaps)
wine.fit.full<- regsubsets(quality~., data=wine.dat, nvmax=12)
wine.fit.sum<- summary(wine.fit.full)

par(mfrow=c(2,2))

plot(wine.fit.sum$adjr2, type='l', main='Adj. R-Squared vs. # of Variables',
xlab='# Variables', ylab='Adj. R-Squared')
plot(wine.fit.sum$cp, type='l', main='Cp vs. # of Variables', xlab='#
Variables', ylab='Cp')
plot(wine.fit.sum$bic, type='l', main='BIC vs. # of Variables', xlab='#
Variables', ylab='BIC')
plot(wine.fit.sum$rss, type='l', main='RSS vs. # of Variables', xlab='#
Variables', ylab='RSS')
```

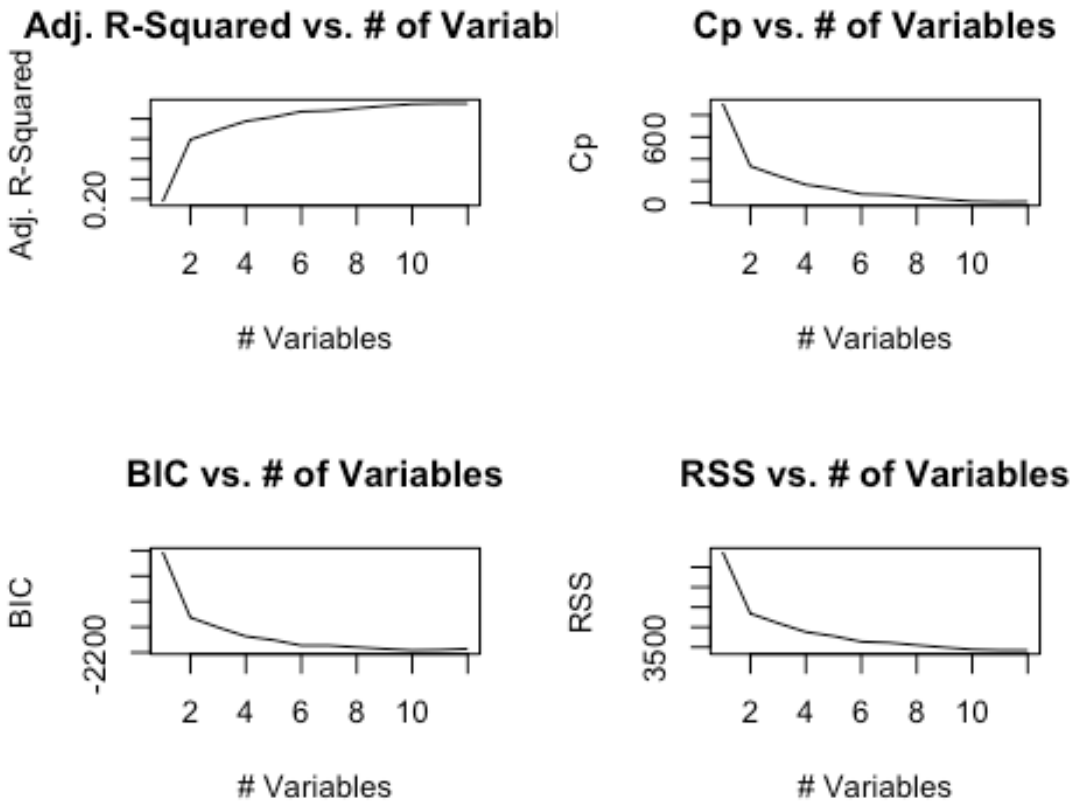

Figure 26: Criterion for BSS Variable Selection

```
#Dataset with both red and white
lm.both.reduced <- lm(quality ~type+`fixed acidity`+`volatile
acidity`+`residual sugar`+`free sulfur dioxide`+`total sulfur
dioxide`+density+pH+sulphates, data = fin.train.data)

reduced.lm.sum<-summary(lm.both.reduced)
lm.reduced.pred <- predict(lm.both.reduced, fin.test.data)
lm.both.mse <- mean((lm.reduced.pred - fin.test.data$quality)^2)
lm.both.mse

## [1] 0.5400551

#Just red with BSS
wine.fit.red<- regsubsets(quality~., data=red.dat[,-1], nvmax=12)
wine.fit.red.sum<- summary(wine.fit.red)

par(mfrow=c(2,2))

plot(wine.fit.red.sum$adjr2, type='l', main='Adj. R-Squared vs. # of
Variables', xlab='# Variables', ylab='Adj. R-Squared')
plot(wine.fit.red.sum$cp, type='l', main='Cp vs. # of Variables', xlab='#
```

```
Variables', ylab='Cp')
plot(wine.fit.red.sum$bic, type='l', main='BIC vs. # of Variables', xlab='#
Variables', ylab='BIC')
plot(wine.fit.red.sum$rss, type='l', main='RSS vs. # of Variables', xlab='#
Variables', ylab='RSS')
```

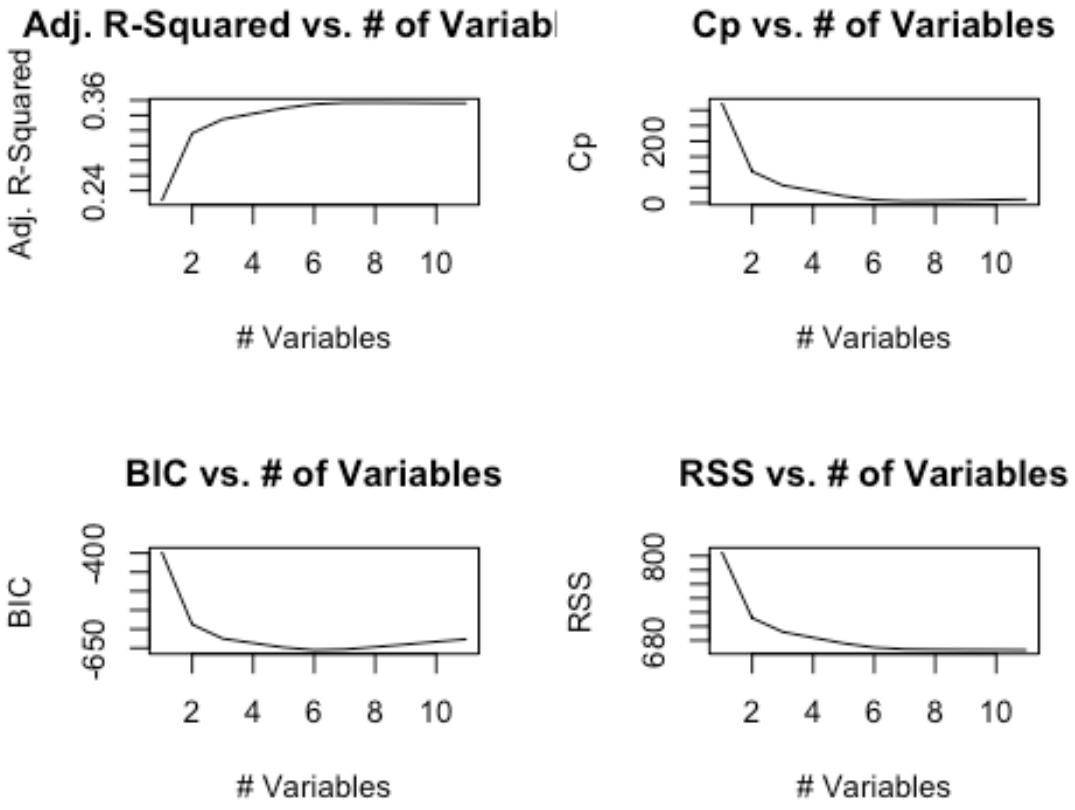

Figure 27: Criterion for BSS Variable Selection with Red Model

```
names(coef(wine.fit.red,6))

## [1] "(Intercept)"            "`volatile acidity`"     "chlorides"
## [4] "`total sulfur dioxide`" "pH"                     "sulphates"
## [7] "alcohol"

lm.red.reduced <- lm(quality ~`volatile acidity`+`chlorides`++`total sulfur
dioxide`+pH+sulphates+alcohol, data = train.data.red)
reduced.red.sum<-summary(lm.red.reduced)
lm.reduced.red.pred <- predict(lm.red.reduced, test.data.red)
lm.red.mse.reduced <- mean((lm.reduced.red.pred - test.data.red$quality)^2)
lm.red.mse.reduced

## [1] 0.4172256
```

```r
#Just white with BSS
wine.fit.white<- regsubsets(quality~., data=white.dat[,-1], nvmax=12)
wine.fit.white.sum<- summary(wine.fit.white)

par(mfrow=c(2,2))

plot(wine.fit.white.sum$adjr2, type='l', main='Adj. R-Squared vs. # of
Variables', xlab='# Variables', ylab='Adj. R-Squared')
plot(wine.fit.white.sum$cp, type='l', main='Cp vs. # of Variables', xlab='#
Variables', ylab='Cp')
plot(wine.fit.white.sum$bic, type='l', main='BIC vs. # of Variables', xlab='#
Variables', ylab='BIC')
plot(wine.fit.white.sum$rss, type='l', main='RSS vs. # of Variables', xlab='#
Variables', ylab='RSS')
```

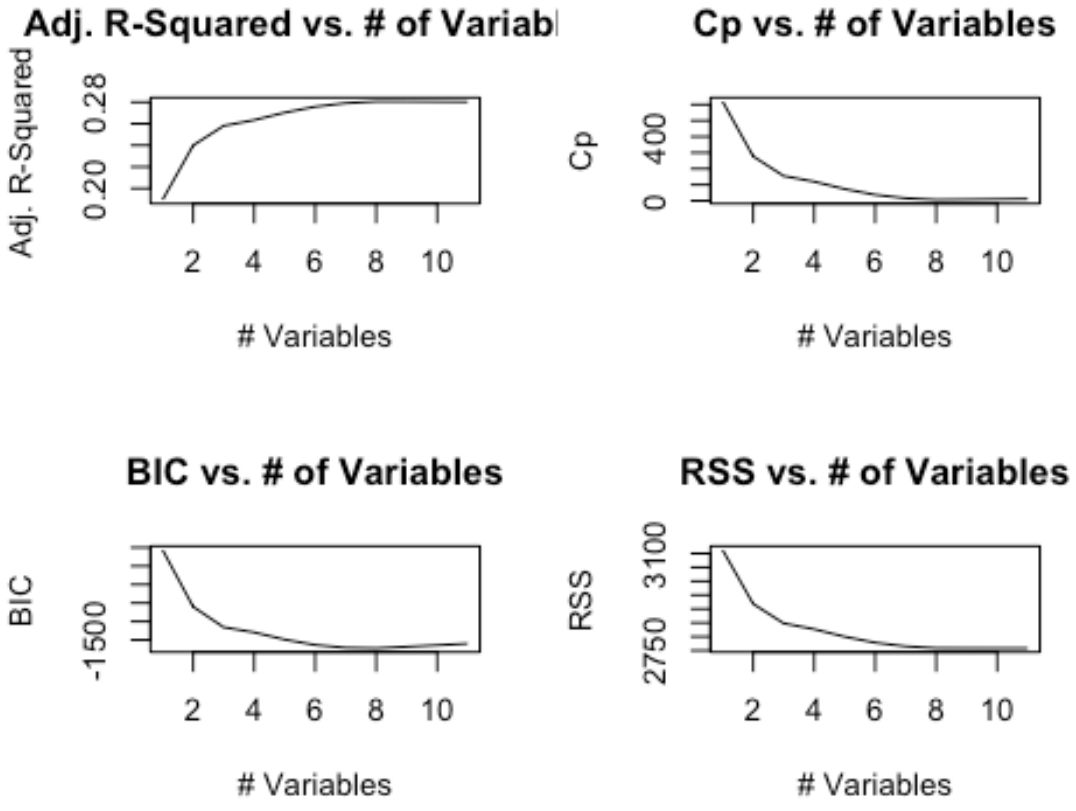

Figure 28: Criterion for BSS Variable Selection with White Model

```r
names(coef(wine.fit.white,8))
```

```
## [1] "(Intercept)"           "`fixed acidity`"       "`volatile acidity`"
## [4] "`residual sugar`"      "`free sulfur dioxide`" "density"
## [7] "pH"                    "sulphates"             "alcohol"
```

```
lm.white.reduced <- lm(quality ~`fixed acidity`+`volatile acidity`+`residual
sugar`+`free sulfur dioxide`+density+pH+sulphates+alcohol, data =
train.data.red)
reduced.white.sum<-summary(lm.white.reduced)
lm.reduced.white.pred <- predict(lm.white.reduced, test.data.white)
lm.white.mse.reduced <- mean((lm.reduced.white.pred -
test.data.white$quality)^2)
lm.white.mse.reduced

## [1] 0.6032922
```

Classification Tree

```
library(tree)
colnames(fin.train.data) <- gsub(" ", "_", colnames(fin.train.data))
colnames(fin.test.data) <- gsub(" ", "_", colnames(fin.test.data))

both.tree<-tree(quality~., data=fin.train.data)
plot(both.tree)
text(both.tree, pretty=0)
```

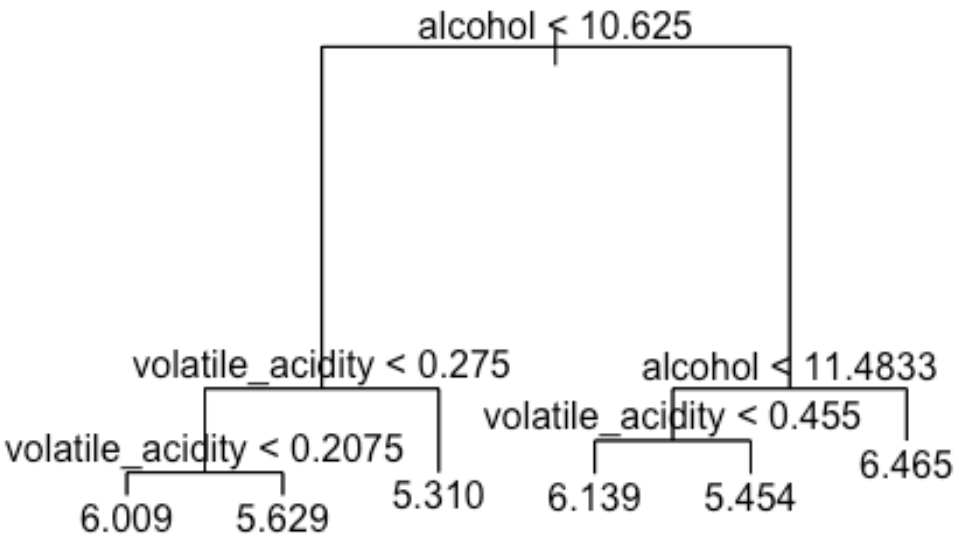

Figure 29: Regression Tree Visualized for Full Model

```
pred <- predict(both.tree, newdata = fin.test.data)
test.mse <- mean((pred - fin.test.data$quality)^2)
test.mse
```

```
## [1] 0.5624006
```

```
colnames(train.data.red) <- gsub(" ", "_", colnames(train.data.red))
colnames(test.data.red) <- gsub(" ", "_", colnames(test.data.red))

red.tree<-tree(quality~., data=train.data.red)
plot(red.tree)
text(red.tree, pretty=0)
```

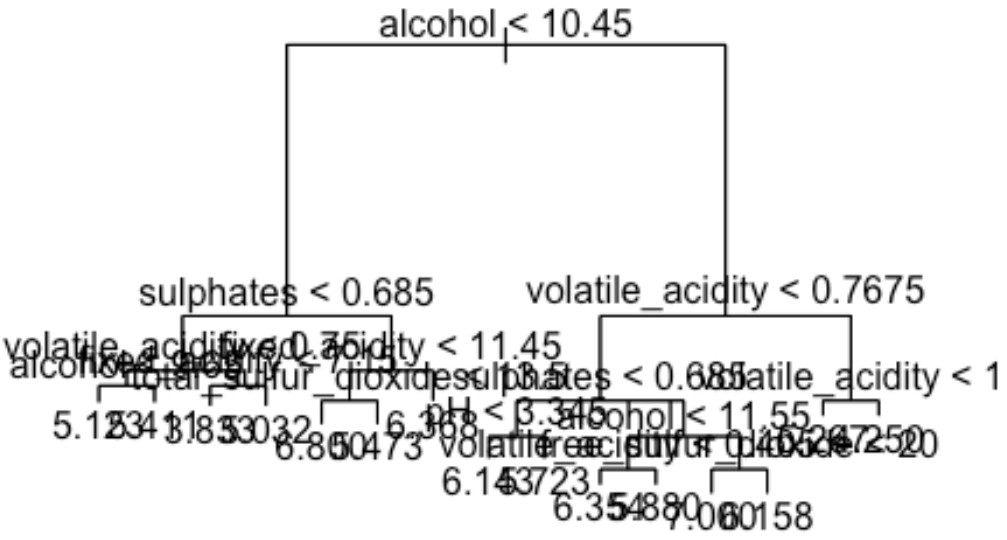

Figure 30: Regression Tree Visualized for Red Model

```
pred.red <- predict(red.tree, newdata = test.data.red)
red.test.mse <- mean((pred.red - test.data.red$quality)^2)
red.test.mse
```

```
## [1] 0.4998054
```

```
colnames(train.data.white) <- gsub(" ", "_", colnames(train.data.white))
colnames(test.data.white) <- gsub(" ", "_", colnames(test.data.white))
```

```
white.tree<-tree(quality
~`fixed_acidity`+`volatile_acidity`+`residual_sugar`+`free_sulfur_dioxide`+de
nsity+pH+sulphates+alcohol, data=train.data.white)
plot(white.tree)
text(white.tree, pretty=0)
```

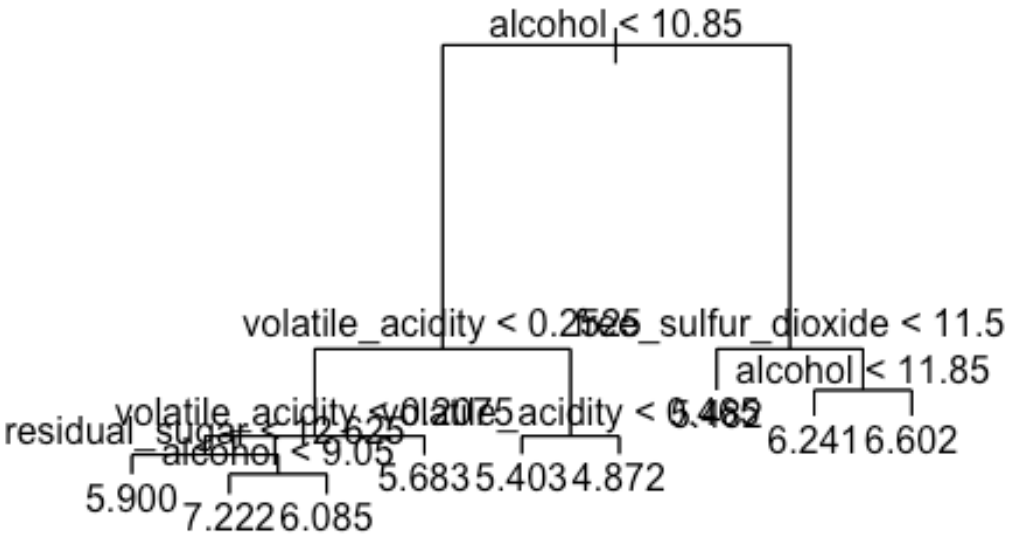

Figure 31: Regression Tree Visualized for White Model

```
pred.white <- predict(white.tree, newdata = test.data.white)
white.test.mse <- mean((pred.white - test.data.white$quality)^2)
white.test.mse
```

```
## [1] 0.5489912
```

KNN

```
#K-Nearest Neighbors Algorithm On Wines DataSet (az296)

#Fitting our KNN model on the training data
library(caret)
```

```
## Loading required package: lattice
```

```
set.seed(1)
trainControl <- trainControl(method="repeatedcv",number=10,repeats=3)
```

```
fit.knn <- train(quality~.,
                 data = fin.train.data,
                 method="knn",
                 preProcess=c("center","scale"),
                 trcontrol=trainControl)
print(fit.knn)
```

```
## k-Nearest Neighbors
##
## 3247 samples
##   12 predictor
##
## Pre-processing: centered (12), scaled (12)
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 3247, 3247, 3247, 3247, 3247, 3247, ...
## Resampling results across tuning parameters:
##
##   k  RMSE       Rsquared   MAE
##   5  0.7897954  0.2625325  0.5944420
##   7  0.7702026  0.2755425  0.5890834
##   9  0.7607554  0.2822192  0.5860804
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 9.
```

Figure 32: KNN Output

```
#Thus, according to our K-Fold Cross validation, we will proceed with the k=9
nearest neighbors
#model since it has the lowest Training RMSE
test_pred <- predict(fit.knn,newdata=fin.test.data)
test_pred <- round(test_pred)
fin.test.data["pred_quality"] <- test_pred
accuracy <- mean(fin.test.data$quality == fin.test.data$pred_quality)
statement_KNN <- paste0("The test accuracy of our KNN algorithm with K=9 is:
", round(accuracy,digits=3))
print(statement_KNN)
```

```
## [1] "The test accuracy of our KNN algorithm with K=9 is: 0.556"
```