Predicting wine quality using physio-chemical attributes

STAT 420 Group Data Project Report

Team AMB - Adriane Yi (adriane3), Baolong Truong (baolong3), Monika Janas (janas3)

2023-08-04

Introduction

In this report, we will be studying the impact of various physio-chemical attributes on predicting the quality of red and white wines.

The dataset we are using is from the UCI Machine Learning Repository (https://archive.ics.uci.edu/dataset/186/wine+quality). The dataset contains 5320 instances and 12 attributes, detailing red and white vinho verde wine samples from the northern region of Portugal. The original dataset is split into two csv files, `winequality-red.csv and winequality-white.csv` for red and white wines respectively, for the purpose of the study, we will combine these into one dataset with wine type as a 13th, categorical variable.

Our goal is to explore how effective regression algorithms are at predicting the quality of wine based on the provided variables, and which physio-chemical attributes contribute most to the quality of a wine, and if there is a difference in contributing attributes between red and white wines.

This study is interesting as it is a compelling exploration into the multifaceted relationship between wine quality and the various physio-chemical compounds. The complexity of the dataset accentuates the challenge of the potential of multi-layered connections between standardized measurements of these compounds and the subjective human sensations tied to preference and taste. Our study endeavor is directed towards constructing a model predicting wine quality, through series of explorations with various predictor variables and offering potential insights.

N	Variables	Type	Description
1	fixed acidity	numeric	(g(tartaric acid)/dm3)
2	volatile acidity	numeric	(g(acetic acid)/dm3)
3	citric acid	numeric	(g/dm3)
4	residual sugar	numeric	(g/dm3)
5	chlorides	numeric	(g(sodium chloride)/dm3)

N	Variables	Type	Description
6	free sulfur dioxide	numeric	(mg/dm3)
7	total sulfur dioxide	numeric	(mg/dm3)
8	density	numeric	(g/cm3)
9	pH	numeric	quantitative measure of the acidity or basicity
10	sulphates	numeric	(g(potassium sulphate)/dm3)
11	alcohol	numeric	alcohol (% vol.)
12	quality	numeric	Preference
13	type	factor	red(0) or white(1)

Citation:

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. Source Link (http://www3.dsi.uminho.pt/pcortez/wine/)

Methods

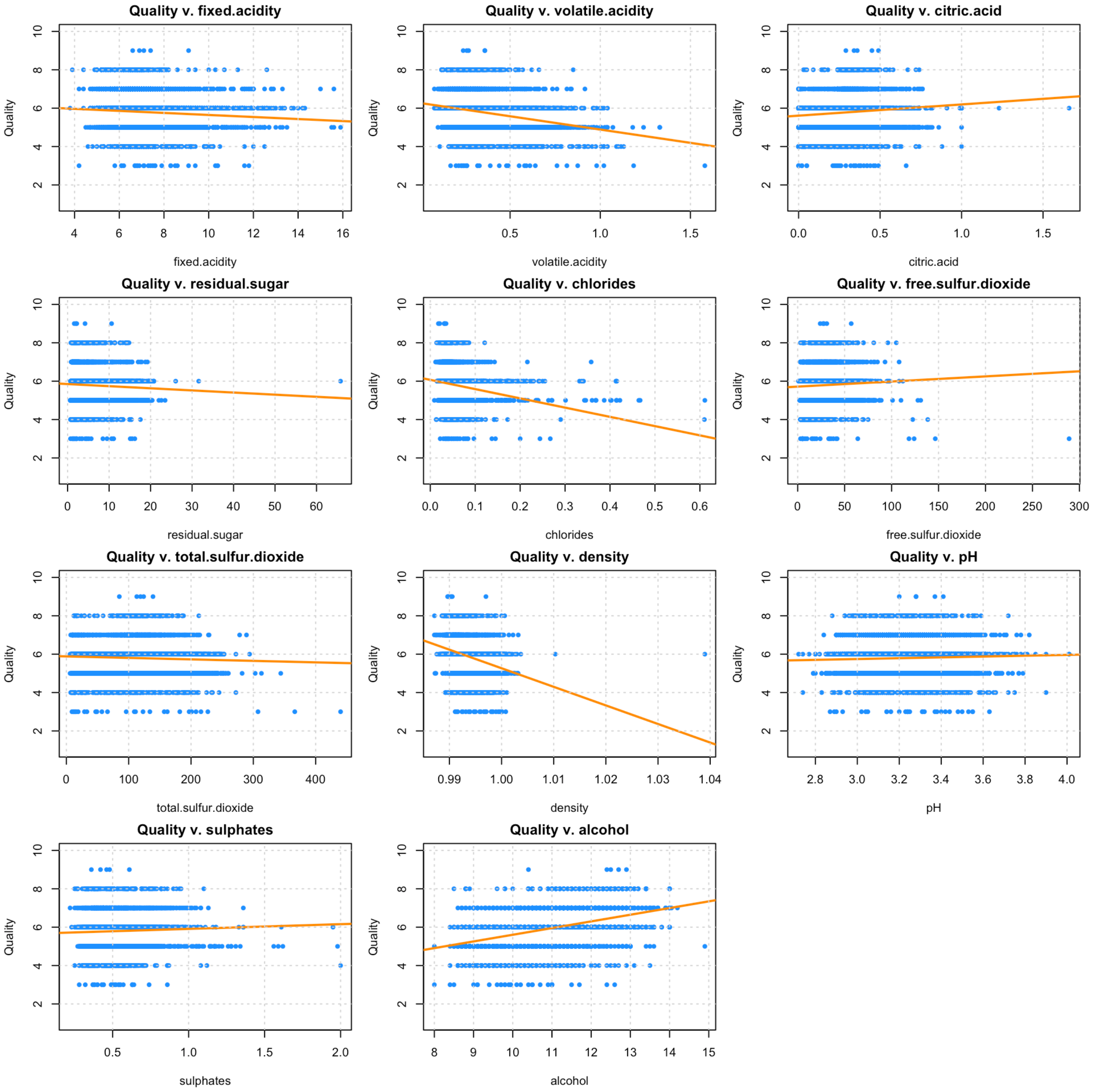
First, we will load in the dataset.

```r
library(readr)
library(knitr)
library(dplyr)
library(tibble)
library(lmtest)
library(car)

red = read.csv("winequality-red.csv", sep = ";")
red$type = "Red"
white = read.csv("winequality-white.csv", sep = ";")
white$type = "White"
winequality = bind_rows(red, white)

winequality$type = as.factor(winequality$type)
winequality$quality = as.numeric(winequality$quality)
winequality = na.omit(winequality)
winequality = unique(winequality)
```

```
set.seed(20230801)

trn_idx = sample(nrow(winequality), size = trunc(0.8 * nrow(winequality)))
wine_trn = winequality[ trn_idx, ]
wine_tst = winequality[-trn_idx, ]
```

We start with plotting the individual predictors vs. the response to check for linear relationships between each predictor variable and the response variable `quality` . Also correlation coefficient is calculated to see which predictors have a strong linear relationship with the response. From the correlation coefficient, it seems like `volatile acidity` , `density` , and `alcohol` have the strongest linear relationship with the response `quality` .

Predictor	Correlation
fixed.acidity	-0.0801
volatile.acidity	-0.2652
citric.acid	0.0980
residual.sugar	-0.0568
chlorides	-0.2021
free.sulfur.dioxide	0.0540
total.sulfur.dioxide	-0.0503
density	-0.3264
pH	0.0397
sulphates	0.0419
alcohol	0.4694



We first want to try an additive model with all of the possible predictors as a baseline for performance.

```
model_additive = lm(quality ~ ., data = wine_trn)
```

Our baseline additive model achieves an R^2 of 0.3089 and a LOOCV-RMSE of 0.7294. It consists of 13 predictors.

Before applying other regression techniques, we wanted to evaluate if our data had any collinearity between the predictors.

Predictor	VIF
fixed.acidity	4.932
volatile.acidity	2.190
citric.acid	1.667
residual.sugar	8.498
chlorides	1.669
free.sulfur.dioxide	2.239
total.sulfur.dioxide	4.055
density	20.066
pH	2.554
sulphates	1.570
alcohol	5.056
type	6.882

Looking at the variance inflation factor values for the predictors, there are potential collinearity issues with `density` and `residual sugar`. We then evaluated the correlation coefficients for all the predictors. Based on the results, we see that there is a high correlation between `free sulfur dioxide` and `total sulfur dioxide`. We would prefer not to include both these predictors in a model to mitigate the effects of their correlation.

	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfu
fixed.acidity	1.00	0.21	0.33	-0.10	0.29	
volatile.acidity	0.21	1.00	-0.38	-0.16	0.37	
citric.acid	0.33	-0.38	1.00	0.15	0.06	
residual.sugar	-0.10	-0.16	0.15	1.00	-0.12	
chlorides	0.29	0.37	0.06	-0.12	1.00	
free.sulfur.dioxide	-0.28	-0.35	0.13	0.40	-0.19	
total.sulfur.dioxide	-0.33	-0.40	0.19	0.49	-0.27	
density	0.48	0.31	0.09	0.52	0.37	
pH	-0.27	0.25	-0.34	-0.23	0.03	
sulphates	0.30	0.23	0.06	-0.17	0.41	
alcohol	-0.10	-0.07	-0.01	-0.31	-0.27	
quality	-0.08	-0.27	0.10	-0.06	-0.20	

Now that we have a good understanding of the collinearity and correlation coefficients of our predictors, we wanted to evaluate a two-way interaction model. We started with a model containing only the intercept, and used forward selection with AIC criteria to select the optimal two-way interaction model, scoped to the full two-way interaction model.

```
scope_formula_two <- formula(lm(quality ~ .^2, data = wine_trn))
base_model = lm(quality ~ 1, data = wine_trn)

model_twoway_interaction = step(base_model, direction = 'forward', scope = scope_formula_two, trace = 0, steps = 100)
```

Our two-way interaction model selected based on forward selection with AIC criteria achieves an R^2 of 0.3634 and a LOOCV-RMSE of 0.715. It consists of 41 predictors, which include all the predictors from the additive model except for `citric acid`.

We can now use an ANOVA test to compare our additive model to the selected two-way interaction model. Using an $\alpha = 0.05$, the p-value is 1.0858^{-56}, thus we reject H_0 and find that the interaction model is significant.

Next, we wanted to evaluate a three-way interaction model and compare it to our selected two-way interaction model. We again started with a model containing only the intercept, and used forward selection with AIC criteria to select the optimal three-way interaction model, scoped to the full three-way interaction model.

```
scope_formula_three <- formula(lm(quality ~ .^3, data = wine_trn))
model_threeway_interaction = step(base_model, direction = 'forward', scope = scope_formula_three, trace = 0, steps = 100)
```

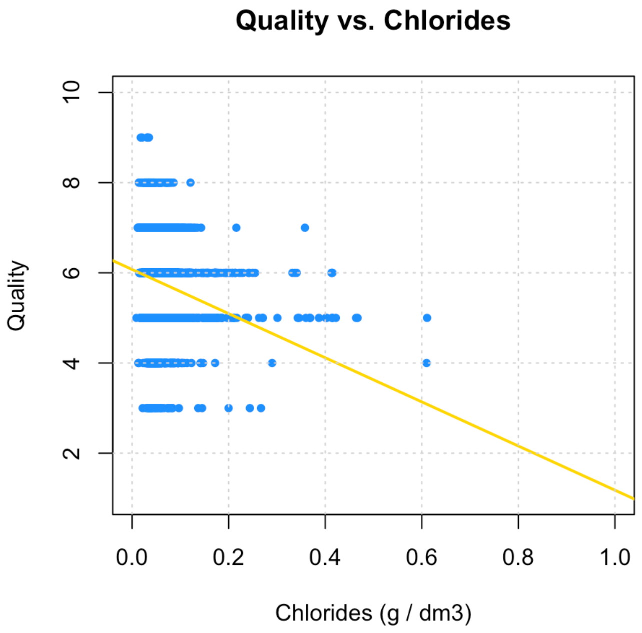
Our three-way interaction model selected based on forward selection with AIC criteria achieves an R^2 of 0.3732 and a LOOCV-RMSE of 0.7051. It consists of 49 predictors.

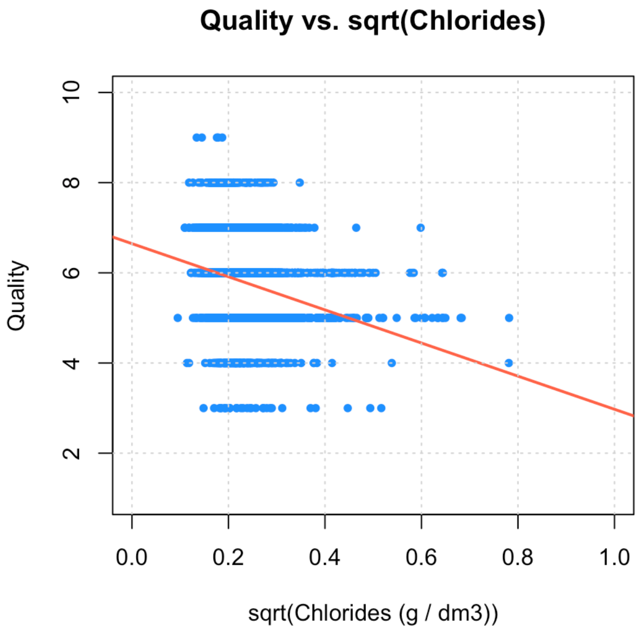
We can now use an ANOVA test to compare our selected two-way interaction model to the selected three-way interaction model. Using an $\alpha = 0.05$, the p-value is 4.1535^{-11}, thus we reject H_0 and find that the three-way interaction model is significant.

While reviewing the plots of predictors versus the response variable, we noticed that a stronger linear relationship between `chlorided` and `quality` could be achieved using a root transformation.

```
model_chlorides = lm(quality ~ chlorides, data = wine_trn)
model_sqrt_chlorides = lm(quality ~ sqrt(chlorides), data = wine_trn)
```

Model	Correlation
Chlorides	-0.2021
sqrt(Chlorides)	-0.2449





By applying a root transformation to `chlorides`, the correlation coefficient between it and `quality` changes from -0.2021 to -0.2449.

We now want to train a new three-way interaction model with the transformation applied to `chlorides` and evaluate it against our previous models.

```
# Create a copy of the dataset, with chlorides set to the sqrt(chlorides)
winequality_transform <- wine_trn
winequality_transform$chlorides = sqrt(winequality_transform$chlorides)

scope_formula_transform <- formula(lm(quality ~ .^3, data = winequality_transfor
m))
base_model_transform = lm(quality ~ 1, data = winequality_transform)

model_transform = step(base_model_transform, direction = "forward", scope = scope_
formula_transform, trace = 0, steps = 100)
```

Our new three-way interaction model selected based on forward selection with AIC criteria, and a transformation on `chlorides`, achieves an R^2 of 0.3713 and a LOOCV-RMSE of 0.706. It consists of 43 predictors.

We can now use an ANOVA test to compare our selected two-way interaction model to the selected three-way interaction model. Using an $\alpha = 0.05$, the p-value is 0.0453, thus we reject H_0 and find that the three-way interaction model with transformation on `chlorides` is significant, though less so when compared with the additive and two-way interaction models.

Lastly, we wanted to evaluate the impact of outlier and influential data points on our model.

```
cook_threshold = 4 / nrow(winequality_transform)
cook_distance = cooks.distance(model_transform)
outliers_cook = which(cook_distance > cook_threshold)
```

Using Cook's distance, we found 216 influential data points in our model, out of 4256 total. We next trained a new model based on the three-way interaction model with transformation, with the outliers removed.

```
winequality_transform_no_outliers = winequality_transform[-outliers_cook,]

scope_formula_transform_no_outliers <- formula(lm(quality ~ .^3, data = winequalit
y_transform_no_outliers))
base_model_transform_no_outliers = lm(quality ~ 1, data = winequality_transform_no
_outliers)

model_transform_no_outliers = step(base_model_transform_no_outliers, direction = "
forward", scope = scope_formula_transform_no_outliers, trace = 0, steps = 100)
```

Our new three-way interaction model selected based on forward selection with AIC criteria, with a transformation on `chlorides` , and outliers removed, achieves an R^2 of 0.4318 and a LOOCV-RMSE of 0.6147. It consists of 51 predictors.

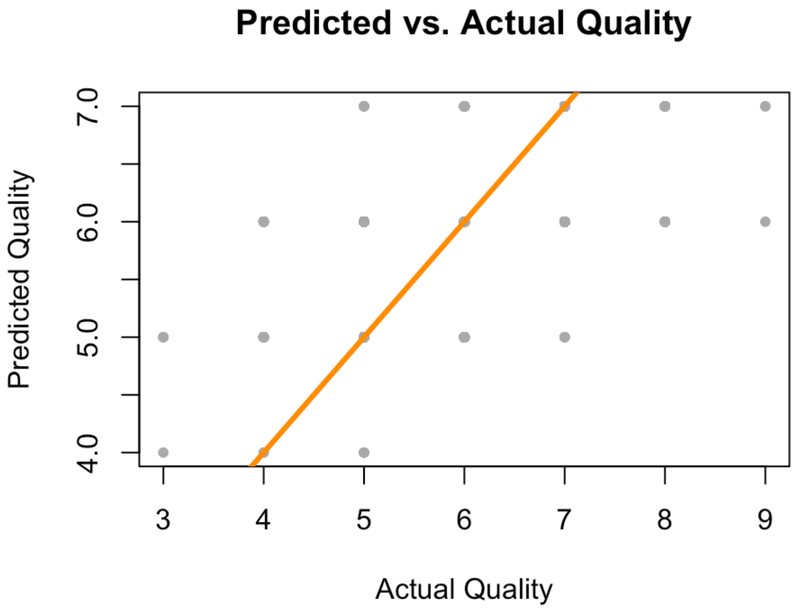
Because the models were fit to different datasets, we cannot compare them using an ANOVA F test. However, based on the improvement in the R^2 value and LOOCV-RMSE, we can set this as our final model and compare all of them together in the Results section.

```
model_final = model_transform_no_outliers
```

```
wine_transform_tst <- wine_tst
wine_transform_tst$chlorides = wine_transform_tst$chlorides^(1/2)

wine_transformed_no_outliers_tst = wine_transform_tst[-outliers_cook,]

actual = wine_transformed_no_outliers_tst$quality
predicted = predict(model_final, newdata = wine_transformed_no_outliers_tst)

predicted = round(unname(predicted), 0)
```

```
percentage_error = 100 * mean(abs(predicted - actual) / predicted)
```

```
plot(predicted ~ actual, col = "darkgrey", pch = 20,
     xlab = "Actual Quality",
     ylab = "Predicted Quality",
     main = "Predicted vs. Actual Quality",)
abline(a = 0, b = 1, lwd = 3, col = "darkorange")
```

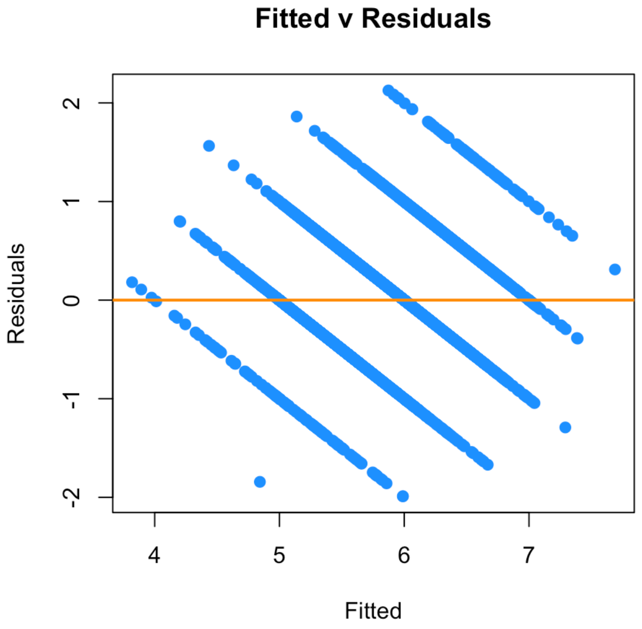


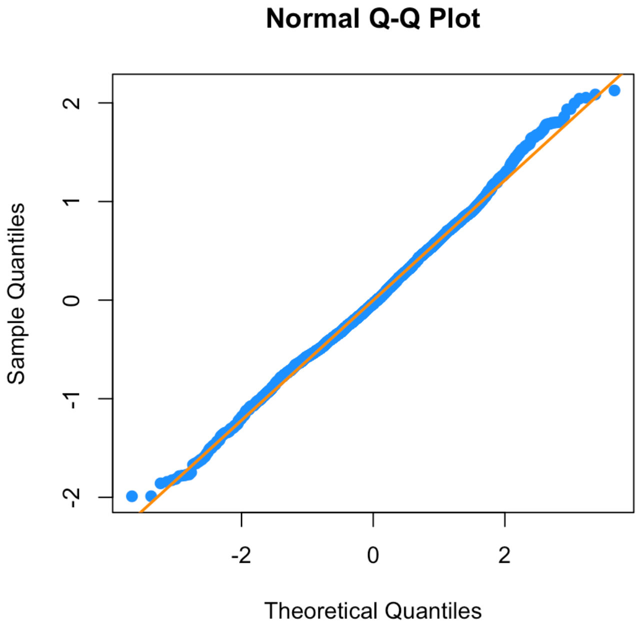
To evaluate the selected model, we computed the **Average Percent Error**, using test-data with outliers removed. The resulting percentage error for this test dataset was found to be 8.8965. Further visual inspection was conducted through plotting the *Predicted versus Actual Quality*, with a y = x line drawn in dark orange. This plot revealed an observable trend: data points situated further from the value 6 tended to deviate more noticeably. This tendency suggests a potential concern regarding the model's predictive consistency over the range of quality values.

Results

Model	Predictors	R_Squared	LOOCV_RMSE	PercentError
Additive	13	0.3089	0.7294	10.015
2-Way Interaction	41	0.3634	0.7150	9.697
3-Way Interaction	49	0.3732	0.7051	9.739
3-Way Interaction with Transformation	43	0.3713	0.7060	10.709
3-Way Interaction with Transformation, Outliers Removed	51	0.4318	0.6147	11.060

```
par(mfrow = c(1,2))
plot_fitted_resid(model_final, "Fitted v Residuals")
plot_qq(model_final)
```





Model	BreuschPagan	ShapiroWilk
Additive	Reject	Reject
2-Way Interaction	Reject	Reject
3-Way Interaction	Reject	Reject
3-Way Interaction with Transformation	Reject	Reject
3-Way Interaction with Transformation, Outliers Removed	Reject	Reject

Discussion

We started with an additive model as a baseline for performance and then studied the two-way and three-way interactions. To build the interaction models we decided to use a forward search with all possible two and three way interactions as the scope to systematically find which predictors and interactions were significant. The two-way interaction model was significantly better than the additive model, but the three-way model was only a slight improvement. We then attempted to transform the chlorides predictor to improve its linear relationship with the response. This also led to a very slight improvement. However, removing the outliers from the data led to a significant improvement from the three-way model with the transformation of chlorides, which became our final model since it had the best performance in the R^2 and LOOCV-RMSE metrics.

Our R^2 of 0.4287 and a LOOCV-RMSE of 0.6153 indicates that 42.87% of variation in quality of wine can be explained by the predictors in our final model and the average distance between the observed values and the predicted values from the model is 0.6153.

From the fitted vs. residual chart and the BP test with alpha 0.05, it is clear that the constant variance assumption is not met. However, this was expected because from our exploratory data analysis it was clear that the relationship between the predictors and the response was not linear. From the normal q-q plot and the Shapiro-Wilks test with alpha 0.05, it is also clear that the normality assumption is not met. However, this was also known since the response variable was discrete, meaning that it was clearly not normally distributed in the first place.

The failure to meet the equal variance and normality assumptions were the biggest shortcomings of our model. However, we believe that this is caused by the nature of our data rather than our methods. Since the response variable is discrete, our data was not normally distributed. Additionally, our exploratory data analysis showed weak linear relationships with the predictor and the response, indicating that a linear regression model was unlikely to be a good model for this data. Finally, a possible concern is the fact that the response variable was also measured by humans, which will introduce variance and inconsistencies in the response.

Although a linear model is clearly not the best model for our data, we still think our final model can be useful. Our final LOOCV-RMSE of 0.6153 indicates that our predictions were actually somewhat close to the observed data with an average error of less than 1 on a 10 point scale. Although our predictions may be inconsistent for different quality values, we still believe the model can predict the quality of a wine reliably in most cases.

Appendix

Helpful functions, plot and table code

```
# Creating the table listing the predictors
n = length(colnames(winequality))
vars = colnames(winequality)
nums = seq(1, n)

vars = c("fixed acidity", "volatile acidity",  "citric acid", "residual sugar", "c
hlorides", "free sulfur dioxide", "total sulfur dioxide",
         "density", "pH", "sulphates", "alcohol", "quality", "type")

data_types = c("numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "
numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "factor")

descriptions = c("(g(tartaric acid)/dm3)", "(g(acetic acid)/dm3)", "(g/dm3)", "(g/
dm3)", "(g(sodium chloride)/dm3)",
                 "(mg/dm3)", "(mg/dm3)", "(g/cm3)", "quantitative measure of the a
cidity or basicity", "(g(potassium sulphate)/dm3)",
                 "alcohol (% vol.)", "Preference", "red(0) or white(1)")

kable(data.frame(N = nums, Variables=vars, Type=data_types, Description=descriptio
ns))
```

```
# Calculate the LOOCV-RMSE value for a model
calc_loocv_rmse = function(model) {
  sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2))
}

calc_percent_error = function(predicted, actual) {
  n = length(actual)
  (1 / n) * sum(abs(predicted - actual) / predicted) * 100
}
```

```
# Create a table of the correlation coefficients of the predictors with the respon
se
x_variables <- setdiff(names(winequality), c("type", "quality"))

df_correlation = data.frame(
  Predictor = x_variables,
  Correlation = c(
    cor(winequality$fixed.acidity, winequality$quality),
    cor(winequality$volatile.acidity, winequality$quality),
    cor(winequality$citric.acid, winequality$quality),
    cor(winequality$residual.sugar, winequality$quality),
    cor(winequality$chlorides, winequality$quality),
    cor(winequality$free.sulfur.dioxide, winequality$quality),
    cor(winequality$total.sulfur.dioxide, winequality$quality),
    cor(winequality$density, winequality$quality),
    cor(winequality$pH, winequality$quality),
    cor(winequality$sulphates, winequality$quality),
    cor(winequality$alcohol, winequality$quality)
  )
)

kable(df_correlation, format = "markdown", padding = 1L)
```

```
# Create plots for the response v each predictor
par(mfrow = c(2, 3), mar = c(4, 4, 2, 1))
for (variable in x_variables) {
  plot(winequality[[variable]], winequality$quality, xlab = variable, ylab = "Qual
ity", ylim = c(1, 10), pch = 20, cex = 1, col = "dodgerblue")
  abline(lm(winequality$quality ~ winequality[[variable]]), col = "darkorange", lw
d = 2)
  title(main = paste("Quality v.", variable))
  grid()
}
```

```
# Create a table with the VIF information from the additive model
kable(data.frame(Predictor = names(vif(model_additive)), VIF = unname(vif(model_ad
ditive))), format = "markdown", padding = 1L)
```

```
}
```

```
# Create a table with the full correlation information for all predictors and the response
kable(round(cor(winequality[,-13]), 2), format = "markdown", padding = 1L)
```

```
# Create the table and plot for the chlorides correlation with the transformation
df_chlorides = data.frame(
  Model = c("Chlorides", "sqrt(Chlorides)"),
  Correlation = c(cor(winequality$chlorides, winequality$quality), cor(sqrt(winequality$chlorides), winequality$quality)))

kable(df_chlorides, format = "markdown", padding = 1L)

par(mfrow = c(1,2))
plot(quality ~ chlorides, data = winequality,
     xlab = "Chlorides (g / dm3)", ylab = "Quality", xlim = c(0, 1), ylim = c(1,10),
     main = paste("Quality vs. Chlorides"), pch = 20, cex = 1, col = "dodgerblue")
abline(model_chlorides, lwd = 2, col = "gold1")
grid()


plot(quality ~ sqrt(chlorides), data = winequality,
     xlab = "sqrt(Chlorides (g / dm3))", ylab = "Quality", xlim = c(0, 1), ylim = c(1,10),
     main = paste("Quality vs. sqrt(Chlorides)"), pch = 20, cex = 1, col = "dodgerblue")
abline(model_sqrt_chlorides, lwd = 2, col = "tomato1")
grid()
```

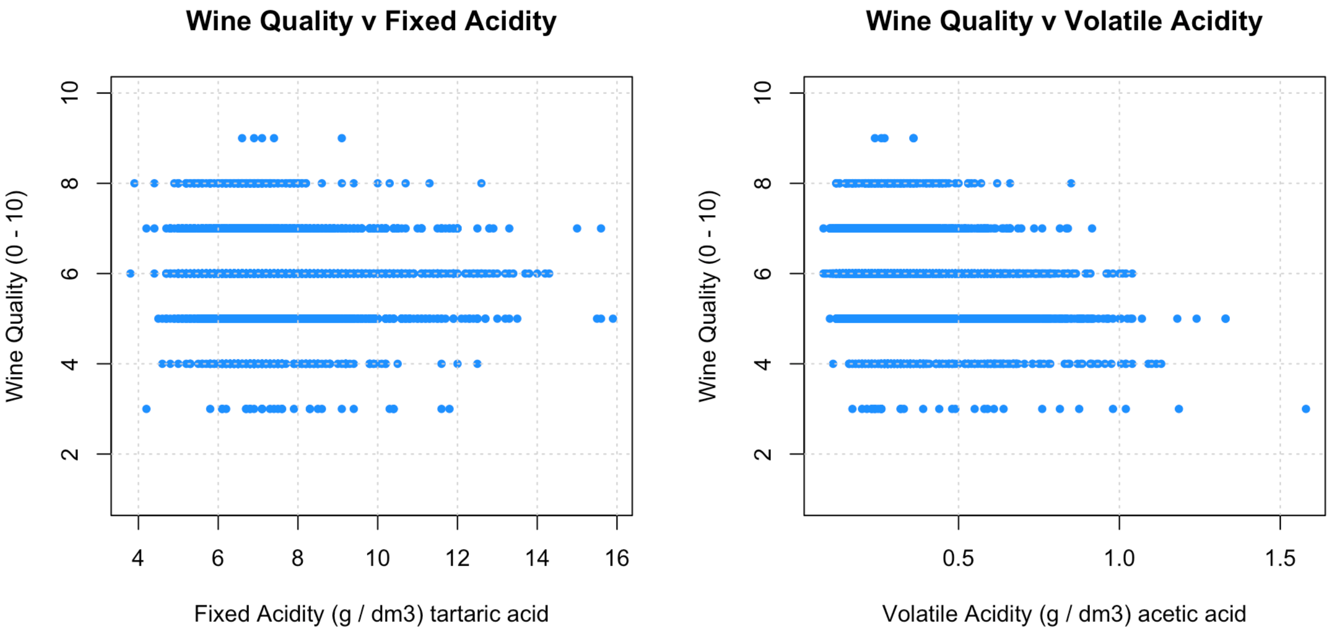
```
# Function to create Fitted v. Residuals plot
plot_fitted_resid = function(model, title, pointcol = "dodgerblue", linecol = "darkorange") {
  plot(fitted(model), resid(model),
       col = pointcol, pch = 20, cex = 1.5,
       xlab = "Fitted", ylab = "Residuals",
       main = title)
  abline(h = 0, col = linecol, lwd = 2)
}

# Function to create Normal QQ plot
plot_qq = function(model, pointcol = "dodgerblue", linecol = "darkorange") {
  qqnorm(resid(model), col = pointcol, pch = 20, cex = 1.5)
  qqline(resid(model), col = linecol, lwd = 2)
}
```

```
calc_bp = function(model_alpha) {
  bptest(model)$p.value
}

calc_bp_decision = function(model, alpha) {
  decide = unname(bptest(model)$p.value < alpha)
  ifelse(decide, "Reject", "Fail to Reject")
}

calc_sw_decision = function(model, alpha) {
  decide = unname(shapiro.test(resid(model))$p.value < alpha)
  ifelse(decide, "Reject", "Fail to Reject")
}
```

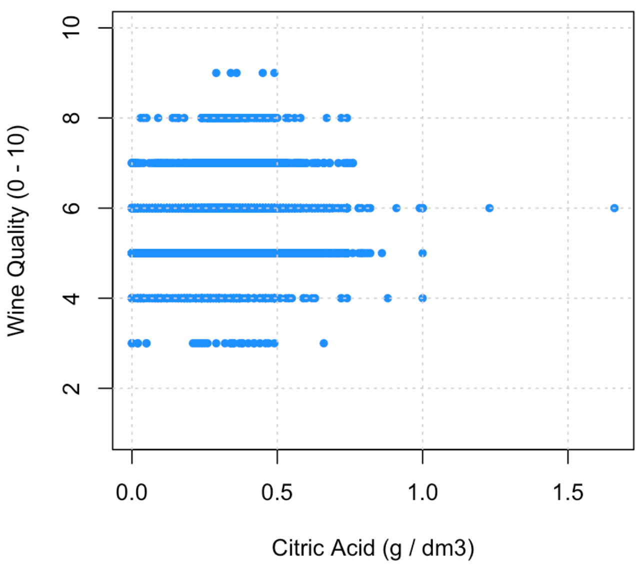
```
# Function to create Fitted v. Residuals plot
plot_fitted_resid = function(model, title, pointcol = "dodgerblue", linecol = "darkorange") {
  plot(fitted(model), resid(model),
       col = pointcol, pch = 20, cex = 1.5,
       xlab = "Fitted", ylab = "Residuals",
       main = title)
  abline(h = 0, col = linecol, lwd = 2)
}

# Function to create Normal QQ plot
plot_qq = function(model, pointcol = "dodgerblue", linecol = "darkorange") {
  qqnorm(resid(model), col = pointcol, pch = 20, cex = 1.5)
  qqline(resid(model), col = linecol, lwd = 2)
}
```

```
model_names = c(
  "Additive",
  "2-Way Interaction",
  "3-Way Interaction",
  "3-Way Interaction with Transformation",
  "3-Way Interaction with Transformation, Outliers Removed")

df_results = data.frame(
  Model = model_names,
  Predictors = c(
    length(coef(model_additive)),
    length(coef(model_twoway_interaction)),
    length(coef(model_threeway_interaction)),
    length(coef(model_transform)),
    length(coef(model_final))
  ),
  R_Squared = c(
    summary(model_additive)$r.squared,
    summary(model_twoway_interaction)$r.squared,
    summary(model_threeway_interaction)$r.squared,
    summary(model_transform)$r.squared,
    summary(model_final)$r.squared
  ),
  LOOCV_RMSE = c(
    calc_loocv_rmse(model_additive),
    calc_loocv_rmse(model_twoway_interaction),
    calc_loocv_rmse(model_threeway_interaction),
    calc_loocv_rmse(model_transform),
    calc_loocv_rmse(model_final)),
  PercentError = c(
    calc_percent_error(predict(model_additive, wine_tst), wine_tst$quality),
    calc_percent_error(predict(model_twoway_interaction, wine_tst), wine_tst$quali
ty),
    calc_percent_error(predict(model_threeway_interaction, wine_tst), wine_tst$qua
lity),
    calc_percent_error(predict(model_transform, wine_tst), wine_tst$quality),
    calc_percent_error(predict(model_final, wine_tst), wine_tst$quality))
)

knitr::kable(df_results, format = "markdown", padding = 1L)
```

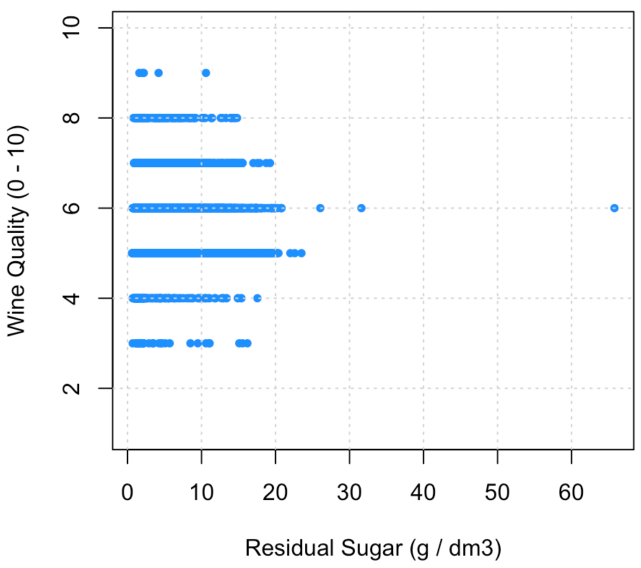
```r
alpha = 0.05

df_results = data.frame(
  Model = model_names,
  BreuschPagan = c(
    calc_bp_decision(model_additive, alpha),
    calc_bp_decision(model_twoway_interaction, alpha),
    calc_bp_decision(model_threeway_interaction, alpha),
    calc_bp_decision(model_transform, alpha),
    calc_bp_decision(model_final, alpha)
  ),
  ShapiroWilk = c(
    calc_sw_decision(model_additive, alpha),
    calc_sw_decision(model_twoway_interaction, alpha),
    calc_sw_decision(model_threeway_interaction, alpha),
    calc_sw_decision(model_transform, alpha),
    calc_sw_decision(model_final, alpha)
  )
)

knitr::kable(df_results, format = "markdown", padding = 1L)
```



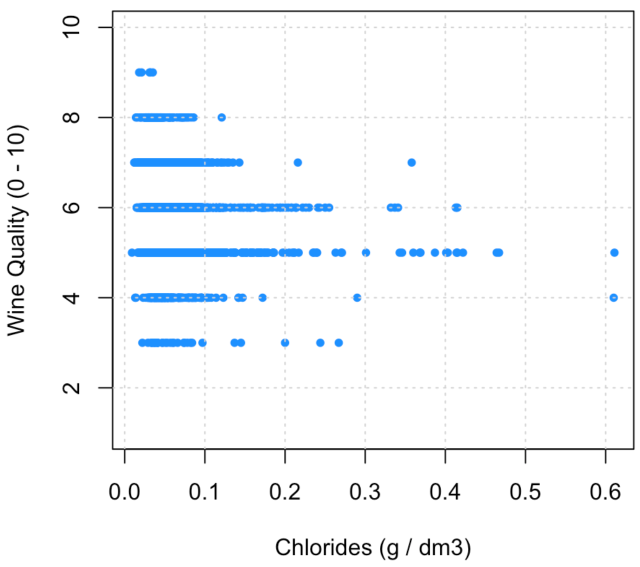
Wine Quality v Citric Acid



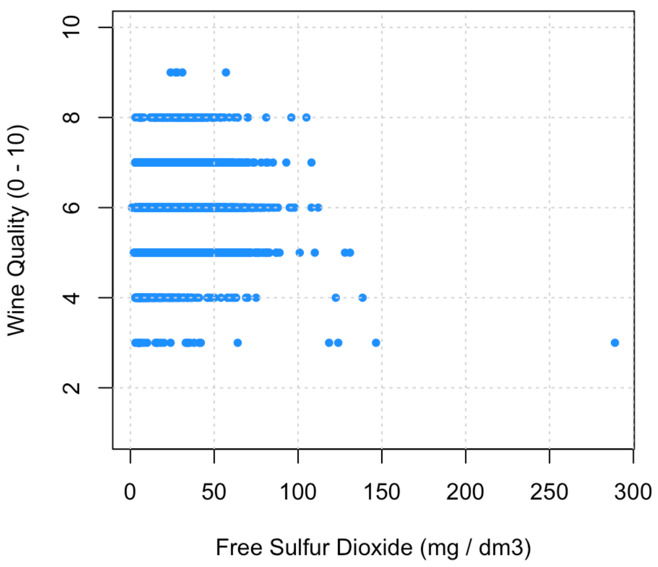
Wine Quality v Residual Sugar



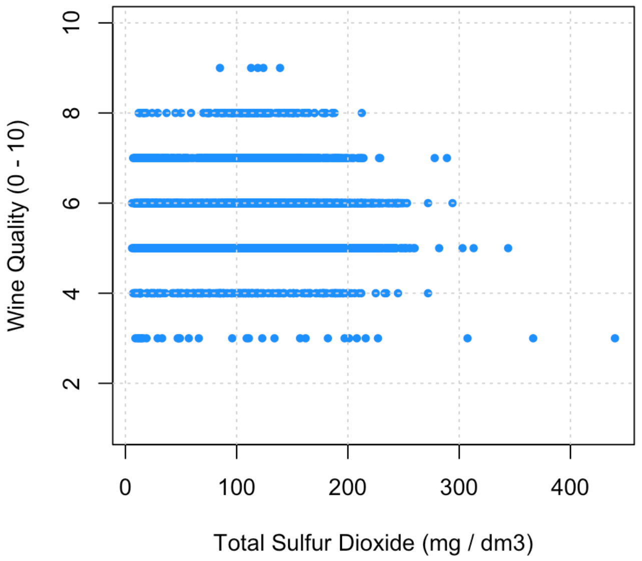
Wine Quality v Chlorides



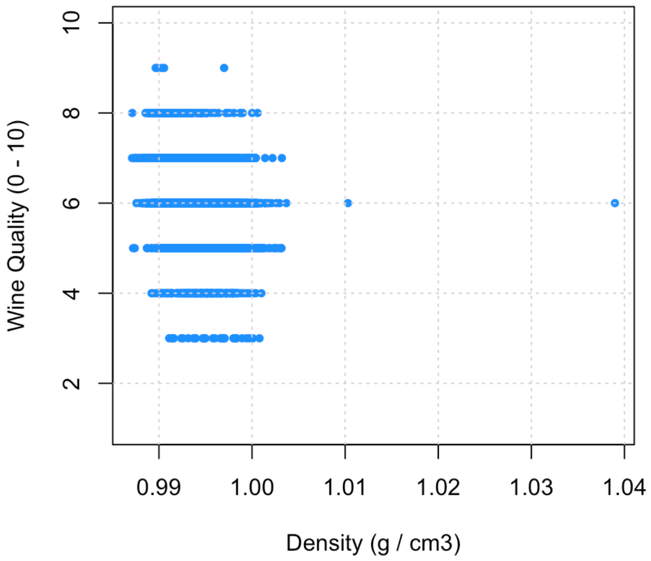
Wine Quality v Free Sulfur Dioxide



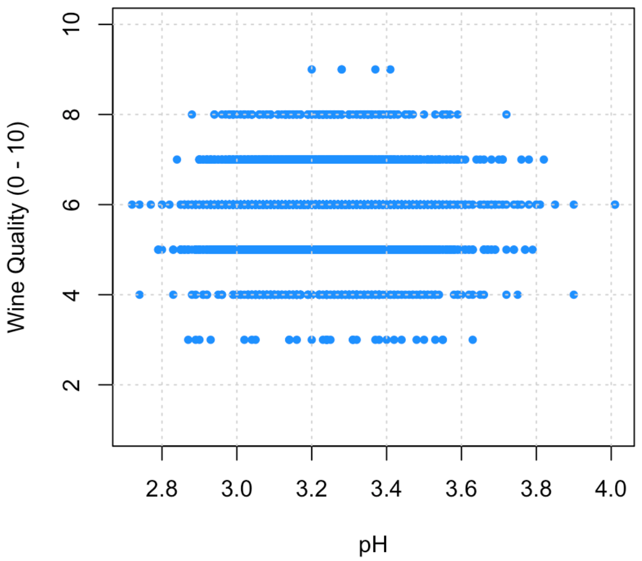
Wine Quality v Total Sulfur Dioxide



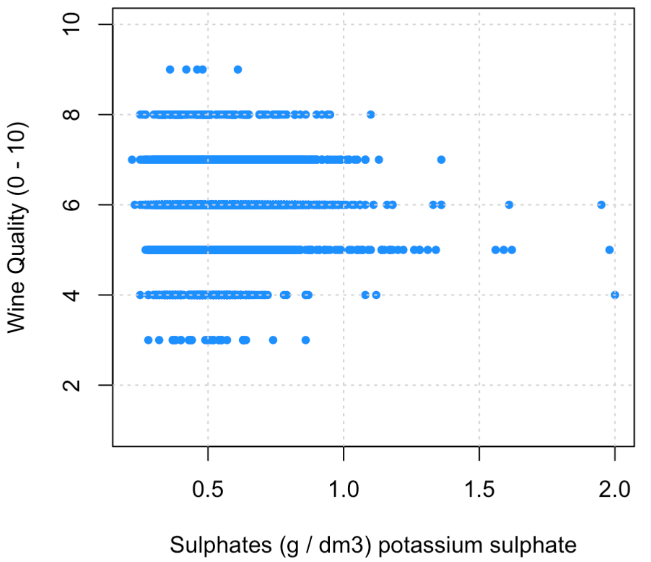
Wine Quality v Density

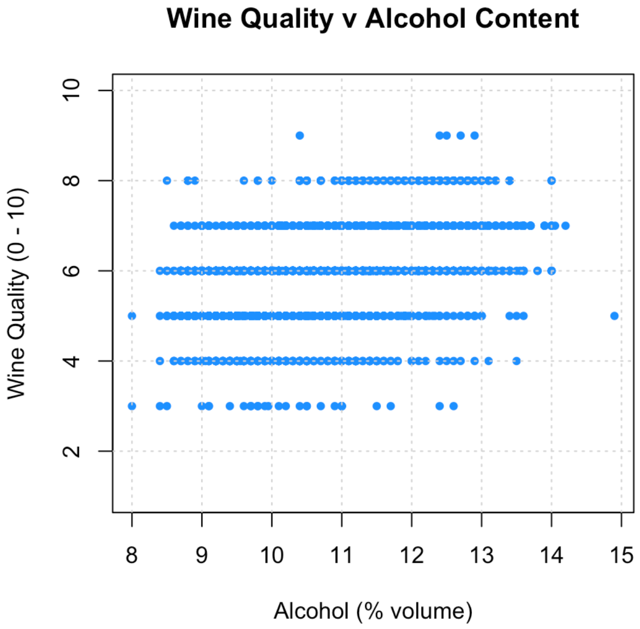


Wine Quality v pH

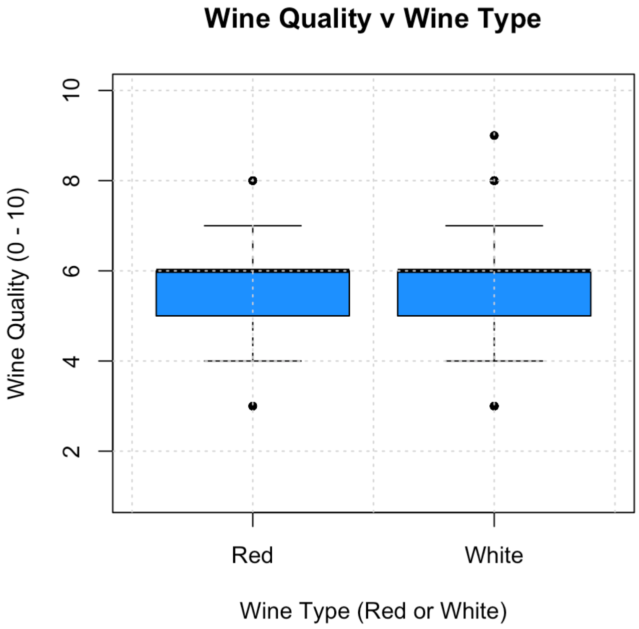


Wine Quality v Sulphates





Wine Quality v Alcohol Content



Wine Quality v Wine Type

“Team AMB - Adriane Yi (adriane3), Baolong Truong (baolong3), Monika Janas (janas3)”