Mohammed Rahman, Amanda Mari, Rhea Ramsaywack

Wine Quality Analysis

I. Introduction

Wine is a beverage that people have enjoyed both alone and socially for several millennia (Choi). The consumption of this alcoholic drink continues to be on the rise in the United States, with the average American consuming about 2.94 gallons of wine in 2016 alone (Wine Institute). Consumers of wine have been found to not only care about the price and even the visual appeal of the bottles that they purchase, but ultimately to care about the perceived quality of the wine that they buy (Preszler and Shmitt). With sales in mind, those in the wine industry strive to produce wine that consumers would rate as "top notch." Since the overall quality of a bottle of wine is subjective to any individual taster, the wine industry has begun to review factors within their control that may affect wine quality ratings. In particular, producers of wine have evaluated the chemical properties of wine to see if they could predict humans' quality rating of wine.

The "Wine Quality" Data Set includes data on 4,898 varieties of wine from Portugal. For analysis purposes, we have chosen to "train" our models using only 4,547 of these observations. Each observation includes information on the type of wine, eleven chemical properties of the drink, and that wine's quality rating. The "type" of wine is binary; the wine is classified as either red or white. Within our training data set, 3,451 varieties are red and 1,096 are white. Next, the data set displays eleven chemical properties of the wine. These properties are fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol. These are all continuous variables that were collected through physiochemical tests. Finally, the “Wine Quality” dataset gives the output variable of quality, which is scored between 0 and 10. Wine quality is based on sensory data.

II. Description of Data

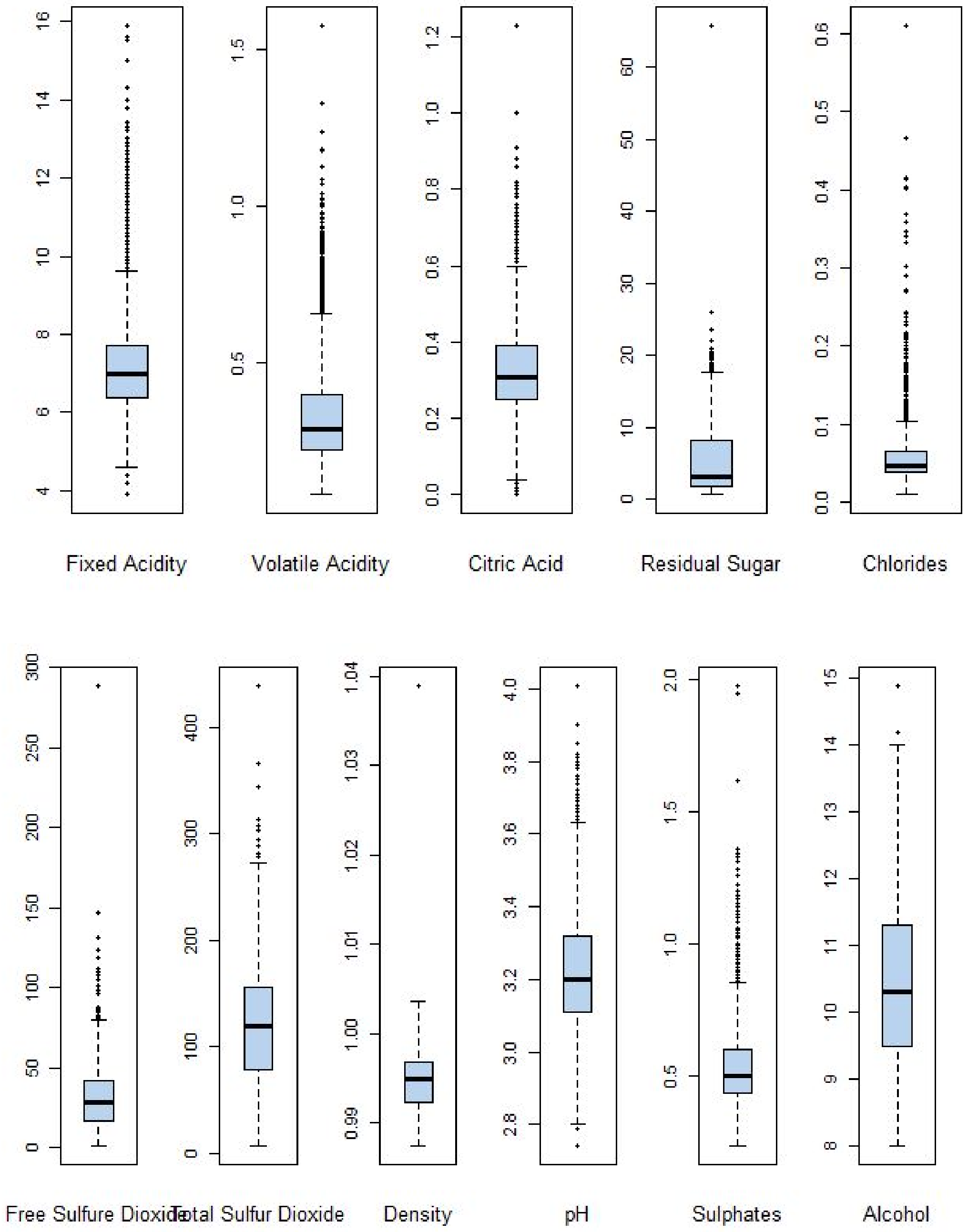
The first two dataset attributes all deal with the acidity of the wine: fixed acidity and citric acid. According to Waterhouse Lab, acidity is a major contributor to wine taste (The Regents of the University of California). The medians of fixed acidity and citric acid in the dataset are 7.0 and .310, respectively. Their modes are 6.8, and .3. In particular, volatile acidity is a measure of the wine's volatile (or gaseous) acids. The primary volatile acid in wine is acetic acid, which is also the primary acid associated with the smell and taste of vinegar. The most common VA concentrations in wine are around 0.4 g/L, with a legal limit of 1.2 g/L.

The next chemical property is residual sugar. Wine Folly says that residual sugar is the sugar left over in wine after fermentation, and that higher levels of residual sugar are usually associated with lower class wines. The residual sugar variable has a median of 3.1. It is important to note that there exists an outlier in the data, as one of the wines has a residual sugar amount of 65.8. On the other hand, the chlorides variable is concerned with the amount of salt in the wine (Resolution Oeno 6/91). The chlorides variable ranges from 0.009 to .611, and its median is .047. According to *The Wineoscope*, several wine producing countries have legal maximums for the amount of chloride in wine. This leads us to believe that too much salt in wine may lead to an unpleasant taste for consumers, and overall a lower quality rating for that wine.

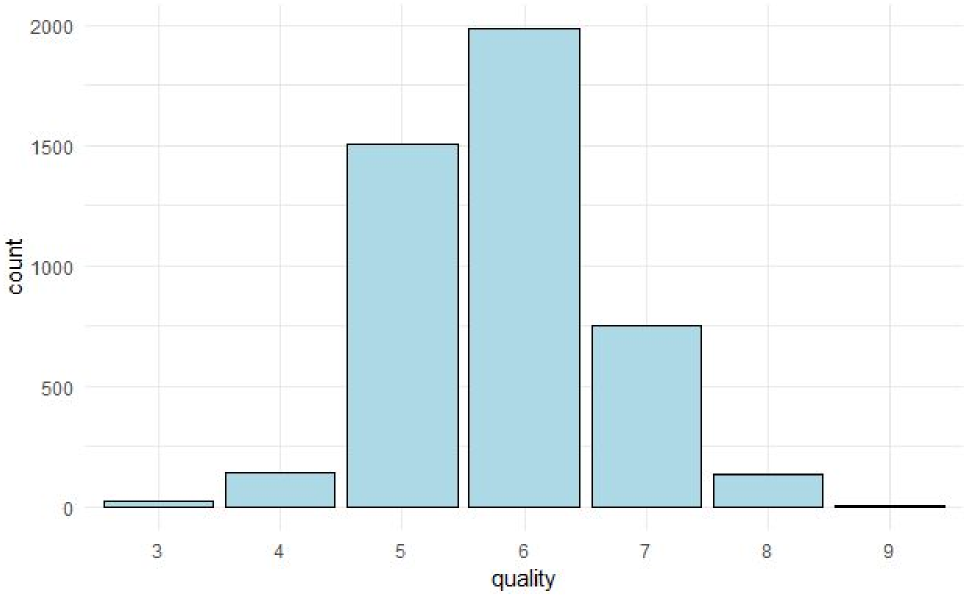
Sulfur dioxide tends to be added into wine by most producers since it has been found to prevent microbial growth and the oxidation of wine (Monro et al.) Similar to the chlorides variable, there are legal maximums for the number of sulfites that can be put into wine because it can affect the taste of the drink. Free sulfur dioxide ranges from 1.0 to 289 and the median of the variable is 29. Total sulfur dioxide ranged from 6 to 440, and its median is 119. Within the dataset, two observations are missing values for the amount of total sulfur dioxide.

Density is measured by the weight in grams of each milliliter of liquid (ETS Laboratories). It has a small range; density ranges from .9874 to 1.0390. The median and mean are approximately the same: .9949 and .9947 respectively. The next variable, pH, is said to be one of the four fundamental traits of wine (Wine Folly). pH is rated on a scale from 0 to 14, with 0-6 being acidic, 7 being neutral, and above 7 being basic. According to *WineSpectator*, most wine pH's are close to 3 or 4, and white wines tend to have a lower pH level than red. In the dataset, wine pH levels range from 2.74 to 4.01 and it has a median value of 3.2. The next variable examined in the data, sulfates, are preservatives added to the wine. The amount of sulfates in the wine varies from .23 to 1.98 with a median of .5. Finally, the last input variable is alcohol, which is the percent alcohol content of the wine. All of the wines range from having 8% alcohol to 14.9% alcohol. The median percent of alcohol is 10.3.

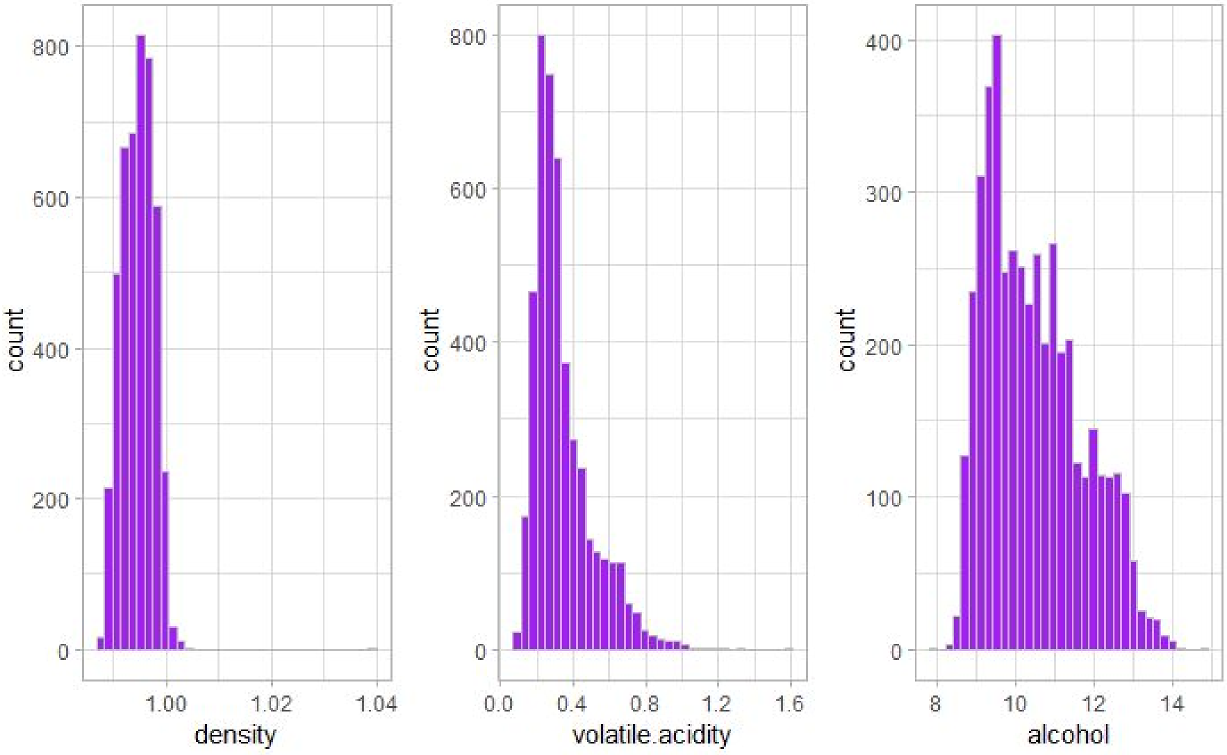
The averages for fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol are 7.21, 0.338, 0.319, 5.46, 0.056, 30.64, 116.38, 0.995, 3.22, 0.538, and 10.48 respectively.

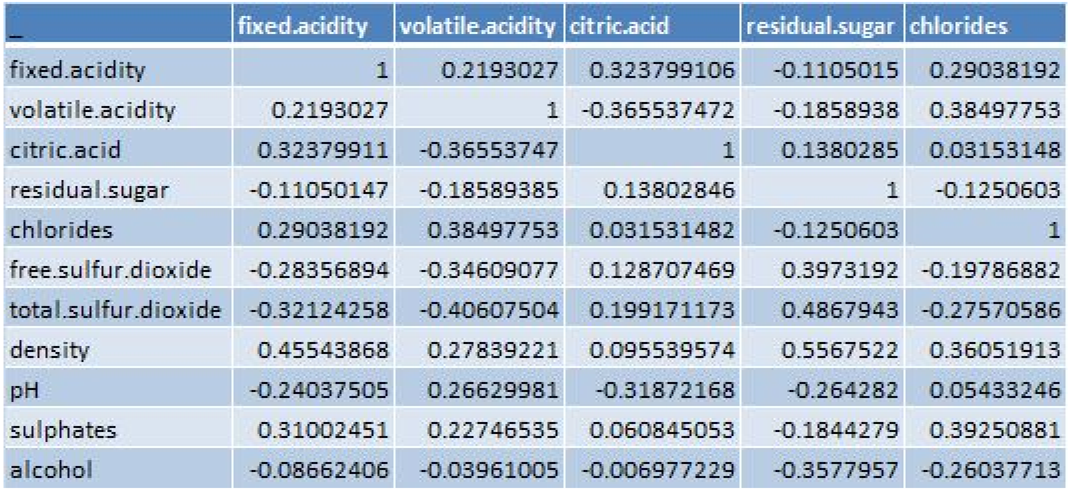

4
6
8
10
12
14
16
Fixed Acidity
0.5
1.0
1.5
Volatile Acidity
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Citric Acid
0
10
20
30
40
50
60
Residual Sugar
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Chlorides
0
50
100
150
200
250
300
Free Sulfure Dioxide
0
100
200
300
400
Total Sulfur Dioxide
0.99
1.00
1.01
1.02
1.03
1.04
Density
2.8
3.0
3.2
3.4
3.6
3.8
4.0
pH
0.5
1.0
1.5
2.0
Sulphates
8
9
10
11
12
13
14
15
Alcohol

It appears that each predictor variable has outliers.

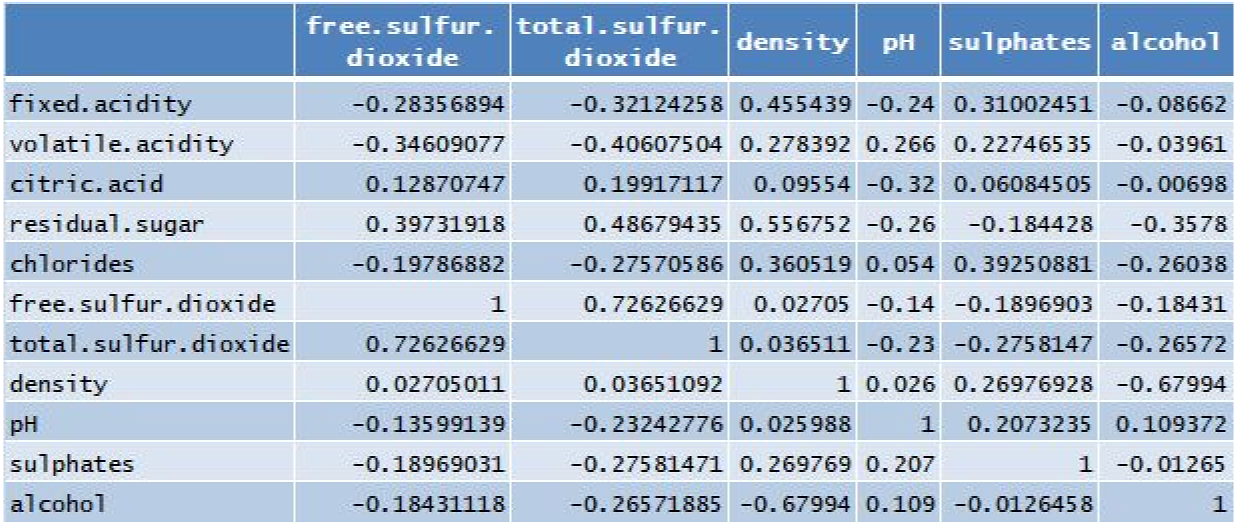


Although the response variable "quality" does range from 0 to 10, the lowest a wine was rated was 3 and the highest was 9. The median quality rating was a 6, and 6 is also the most frequent rating of the wines.

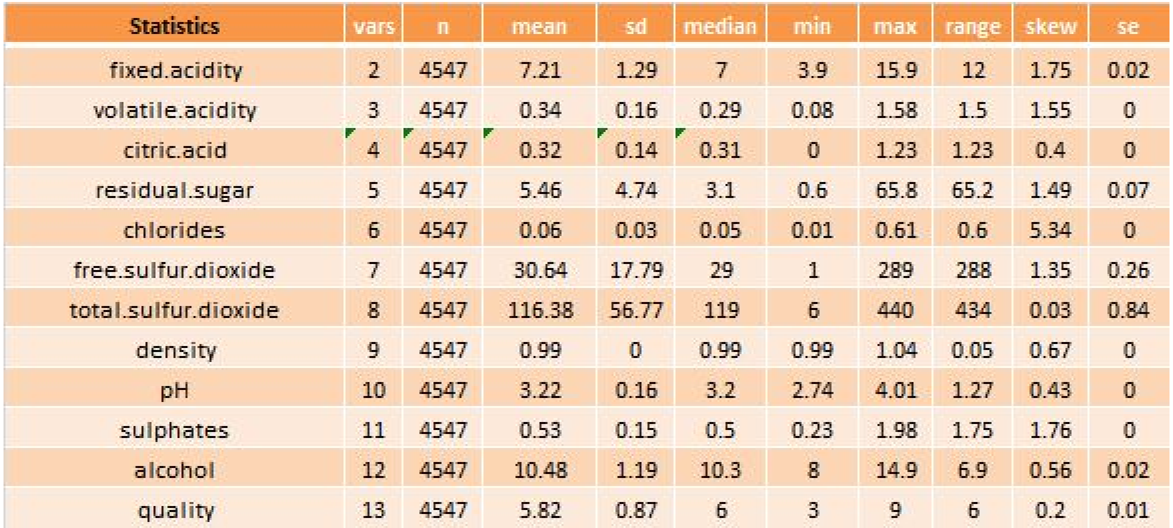




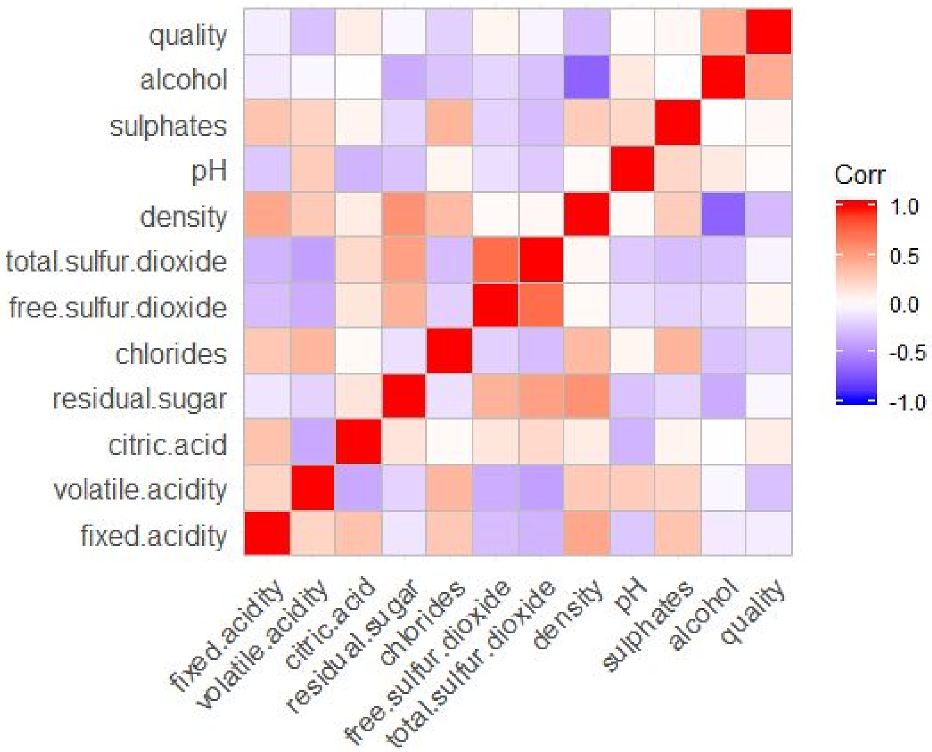
_	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides
fixed.acidity	1	0.2193027	0.323799106	-0.1105015	0.29038192
volatile.acidity	0.2193027	1	-0.365537472	-0.1858938	0.38497753
citric.acid	0.32379911	-0.36553747	1	0.1380285	0.03153148
residual.sugar	-0.11050147	-0.18589385	0.13802846	1	-0.1250603
chlorides	0.29038192	0.38497753	0.031531482	-0.1250603	1
free.sulfur.dioxide	-0.28356894	-0.34609077	0.128707469	0.3973192	-0.19786882
total.sulfur.dioxide	-0.32124258	-0.40607504	0.199171173	0.4867943	-0.27570586
density	0.45543868	0.27839221	0.095539574	0.5567522	0.36051913
pH	-0.24037505	0.26629981	-0.31872168	-0.264282	0.05433246
sulphates	0.31002451	0.22746535	0.060845053	-0.1844279	0.39250881
alcohol	-0.08662406	-0.03961005	-0.006977229	-0.3577957	-0.26037713



	free.sulfur. dioxide	total.sulfur. dioxide	density	pH	sulphates	alcohol
fixed.acidity	-0.28356894	-0.32124258	0.455439	-0.24	0.31002451	-0.08662
volatile.acidity	-0.34609077	-0.40607504	0.278392	0.266	0.22746535	-0.03961
citric.acid	0.12870747	0.19917117	0.09554	-0.32	0.06084505	-0.00698
residual.sugar	0.39731918	0.48679435	0.556752	-0.26	-0.184428	-0.3578
chlorides	-0.19786882	-0.27570586	0.360519	0.054	0.39250881	-0.26038
free.sulfur.dioxide	1	0.72626629	0.02705	-0.14	-0.1896903	-0.18431
total.sulfur.dioxide	0.72626629	1	0.036511	-0.23	-0.2758147	-0.26572
density	0.02705011	0.03651092	1	0.026	0.26976928	-0.67994
pH	-0.13599139	-0.23242776	0.025988	1	0.2073235	0.109372
sulphates	-0.18969031	-0.27581471	0.269769	0.207	1	-0.01265
alcohol	-0.18431118	-0.26571885	-0.67994	0.109	-0.0126458	1



Statistics	vars	n	mean	sd	median	min	max	range	skew	se
fixed.acidity	2	4547	7.21	1.29	7	3.9	15.9	12	1.75	0.02
volatile.acidity	3	4547	0.34	0.16	0.29	0.08	1.58	1.5	1.55	0
citric.acid	4	4547	0.32	0.14	0.31	0	1.23	1.23	0.4	0
residual.sugar	5	4547	5.46	4.74	3.1	0.6	65.8	65.2	1.49	0.07
chlorides	6	4547	0.06	0.03	0.05	0.01	0.61	0.6	5.34	0
free.sulfur.dioxide	7	4547	30.64	17.79	29	1	289	288	1.35	0.26
total.sulfur.dioxide	8	4547	116.38	56.77	119	6	440	434	0.03	0.84
density	9	4547	0.99	0	0.99	0.99	1.04	0.05	0.67	0
pH	10	4547	3.22	0.16	3.2	2.74	4.01	1.27	0.43	0
sulphates	11	4547	0.53	0.15	0.5	0.23	1.98	1.75	1.76	0
alcohol	12	4547	10.48	1.19	10.3	8	14.9	6.9	0.56	0.02
quality	13	4547	5.82	0.87	6	3	9	6	0.2	0.01



From this correlation plot, we can see that density, volatile acidity, and alcohol are the inputs that appear to be most related to quality. Density has a correlation of -0.299 with wine quality, which implies a negative relationship between the two variables. Similarly, volatile acidity has a negative relationship with quality, since its correlation coefficient is -.272. As opposed to density and volatile acidity, alcohol has a moderately strong and positive relationship with quality, since it has a correlation coefficient of .43. Based on our primary analyses, we predict that alcohol, volatile acidity and density will be significant predictors in our final model.

III. Methods and Results

Procedures:

Since quality is a variable that ranges from 0-10, we can treat our outcome as either a classification variable or a numeric variable (rounding the response to the closest rating). Based on this, we have decided to run each model twice. The first time we will treat quality as a factor, and the second time we will treat quality as a numeric variable.

1. Split the training data into 70% training and 30% test data. We will do this twice, and have training and test data sets for when we treat wine as a factor, called “classtrain” and “classtest.” Then, we will also have a training and test data set for when we treat wine as a numeric variable, called “regtrain” and “regtest.”
2. We will run various models on our “classtrain” and “regtrain” sets. We will then use each model to predict quality ratings on our “classtest” and “regtest” sets. To find the error rate of each model on the test set, we will calculate 1- mean (predicted quality ratings=actual quality ratings).
3. We will choose our best model by picking the model that has the lowest test error rate. We will use that model to predict wine quality on the assigned test data.

Linear Regression Model (Regression) & Logistic Regression Model (Classification)

```
set.seed(123)
reglog.fit <- lm(quality ~ ., data = regtrain)
reglog.pred <- round(predict(reglog.fit, regtest), 0)

table(reglog.pred, regtest$quality)
1 - mean(reglog.pred == regtest$quality)
```

```
reglog.pred   3   4   5   6   7   8   9
          4   0   1   3   1   0   0   0
          5   3  21 216  99   7   1   0
          6   4  20 230 456 175  27   1
          7   0   0   2  39  44  13   0
```

```
The error rate for the Linear Regression model is: 0.4739545
```

```
set.seed(123)
classlog.fit <- polr(quality ~ ., data = classtrain)
classlog.pred <- predict(classlog.fit, classtest, type = "class")
1 - mean(classlog.pred == classtest$quality)
```

```
The error rate for the Logistic Regression model is: 0.4680851
```

Feature Selection

Stepwise Logistic Regression Model to choose the best variables

```
nullmodel <- polr(quality ~ 1, data = classtrain)

step(nullmodel, list(
  upper = polr(quality ~ .,  data = classtrain),
  lower = polr(quality ~ 1,  data = classtrain)),
  direction = "both")
```

```
quality ~ alcohol + volatile.acidity + sulphates + residual.sugar + 
    wine_type + free.sulfur.dioxide + total.sulfur.dioxide + 
    pH + density + fixed.acidity + chlorides

                       Df     AIC
<none>                     9916.5
- chlorides             1  9917.1
+ citric.acid           1  9918.2
- wine_type             1  9932.9
- fixed.acidity         1  9936.9
- total.sulfur.dioxide  1  9942.9
- pH                    1  9948.3
- density               1  9949.4
- free.sulfur.dioxide   1  9959.9
- sulphates             1  9986.4
- residual.sugar        1  9986.9
- alcohol               1 10000.4
- volatile.acidity      1 10175.4
```

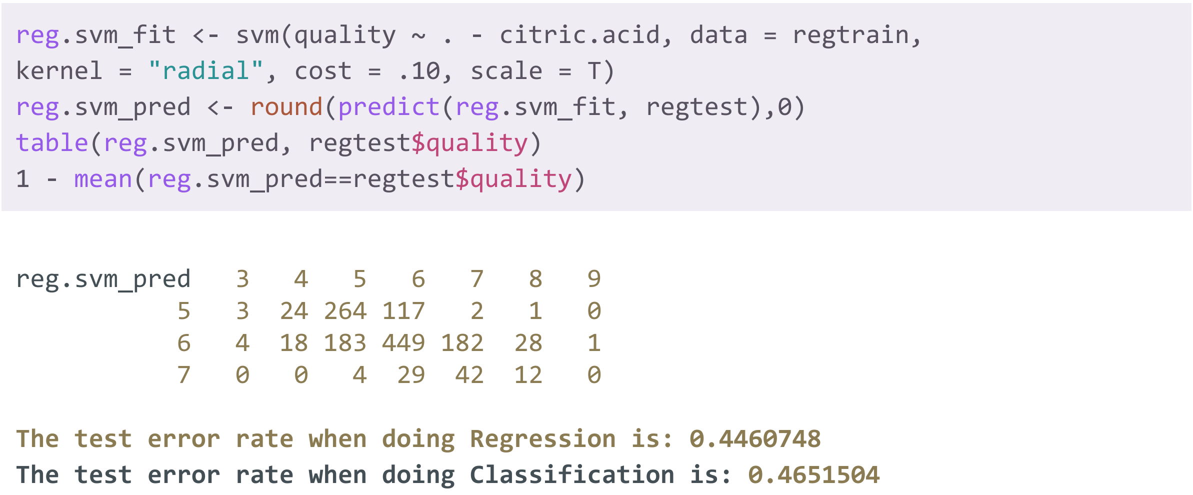
These results suggest that we should not include not include citric acid in our final model.

Polynomial Logistic Regression (Classification) & Polynomial Linear Regression (Regression)

```
regpoly.fit <- lm(quality ~ dummy + poly(fixed.acidity, 2) +
poly(volatile.acidity, 2) + poly(residual.sugar, 2) + poly(chlorides, 2) +
poly(free.sulfur.dioxide,2) + poly(total.sulfur.dioxide,2) + poly(density, 2) +
poly(pH, 2) + poly(sulphates,2) + poly(alcohol,2), regtrain)
regpolypred <- round(predict(regpoly.fit, regtest),0)
1 - mean(regpolypred == regtest$quality)
```

```
The error rate for the polynomial Linear Regression is: 0.4805576
The error rate for the polynomial Logistic Regression is: 0.4783566
```

Support Vector Machine

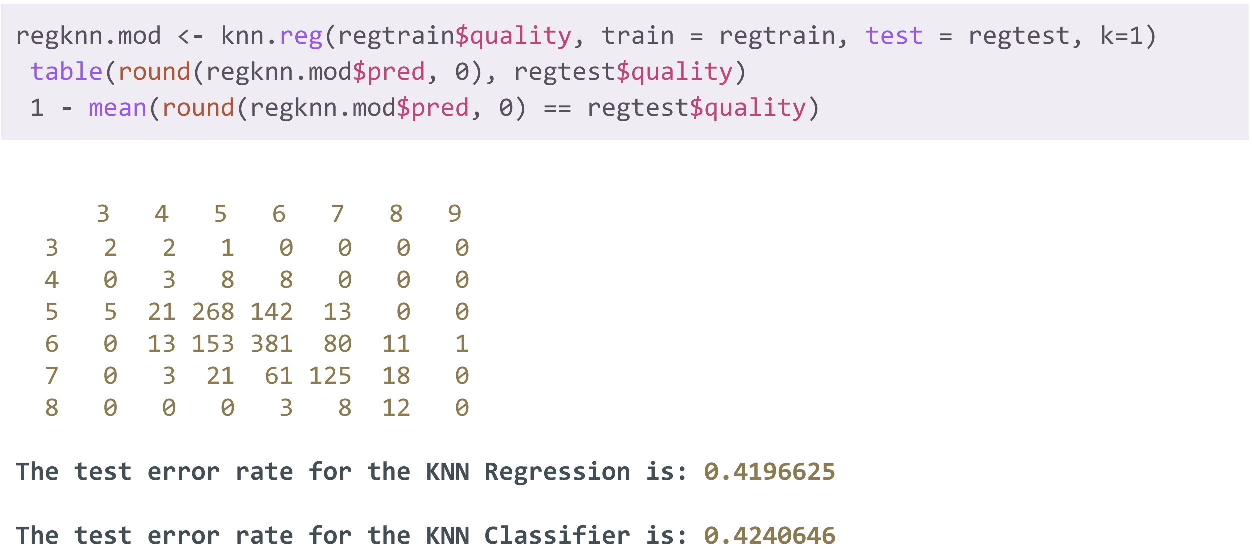


```
reg.svm_fit <- svm(quality ~ . - citric.acid, data = regtrain,
kernel = "radial", cost = .10, scale = T)
reg.svm_pred <- round(predict(reg.svm_fit, regtest),0)
table(reg.svm_pred, regtest$quality)
1 - mean(reg.svm_pred==regtest$quality)
```

```
reg.svm_pred   3   4   5   6   7   8   9
           5   3  24 264 117   2   1   0
           6   4  18 183 449 182  28   1
           7   0   0   4  29  42  12   0

The test error rate when doing Regression is: 0.4460748
The test error rate when doing Classification is: 0.4651504
```

K-Nearest Neighbors



```
regknn.mod <- knn.reg(regtrain$quality, train = regtrain, test = regtest, k=1)
 table(round(regknn.mod$pred, 0), regtest$quality)
 1 - mean(round(regknn.mod$pred, 0) == regtest$quality)
```

```
      3   4   5   6   7   8   9
  3   2   2   1   0   0   0   0
  4   0   3   8   8   0   0   0
  5   5  21 268 142  13   0   0
  6   0  13 153 381  80  11   1
  7   0   3  21  61 125  18   0
  8   0   0   0   3   8  12   0

The test error rate for the KNN Regression is: 0.4196625

The test error rate for the KNN Classifier is: 0.4240646
```

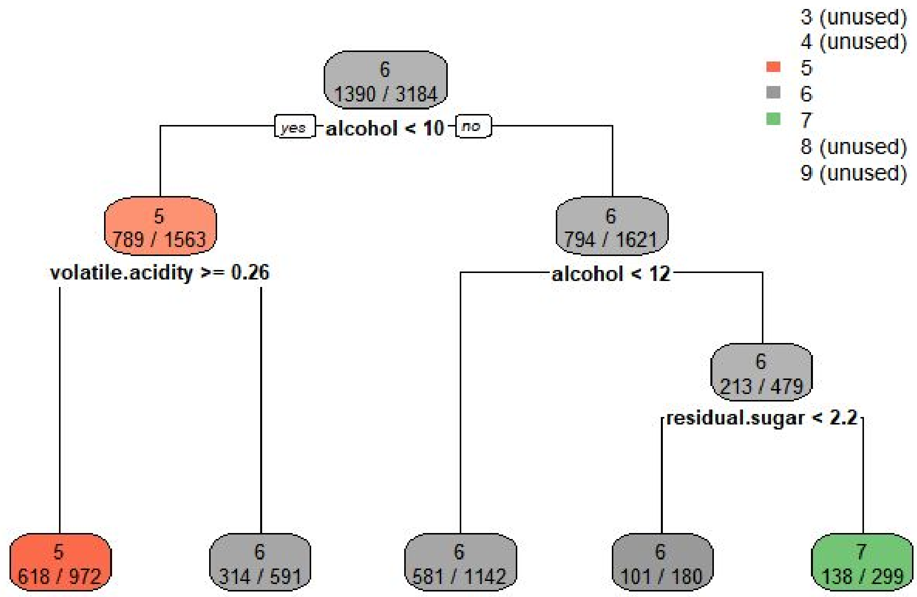
Pruned Tree Model

```r
classtree.cp <- rpart(data = classtrain, quality ~ . - citric.acid, control =
rpart.control(cp = 0.002, minsplit = 500))

regtreecppred <- round(predict(regtree.cp, regtest),0)
1 - mean(regtreecppred == regtest$quality)
```

The test error rate for the regression is: 0.4758034

The test error rate for the Classification is: 0.4834116



Random Forest

```r
t.ctrl <- trainControl(method = "optimism_boot", number = 5)
rf.grid <- expand.grid(mtry = 2)
regrf.train <- train(quality ~ .- citric.acid, data = regtrain, method = "rf",
                  trControl = t.ctrl, tuneGrid = rf.grid,
                  preProcess = c("center", "scale"), do.trace = F, n.tree =500)
regrf.pred <- round(predict(regrf.train, regtest),0)
table(regrf.pred, regtest$quality)
1 - mean(regrf.pred == regtest$quality)
```

```
regrf.pred   3   4   5   6   7   8   9
         5   3  26 288  87   1   1   0
         6   4  15 158 481 124  18   1
         7   0   1   5  27 101  21   0
         8   0   0   0   0   0   1   0
```

```
The test error rate for Regression is: 0.355099

The test error rate for Classification is: 0.3639032
```

IV. Discussion

The first model we tried out was a multiple linear regression with quality as a numeric variable. We used all of the predictors in this model and still achieved a high error rate of approximately .474. Then, we repeated this procedure using a logistic regression model, but we treated quality as a factor within this model. This did not improve our accuracy much, as we obtained an error rate of about .468, which is only slightly lower than that which we obtained from the multiple linear regression. However, we did have an inkling that these models would not perform well. Linear and logistic regression models do not work well on nonlinear data and we are unsure if there is multicollinearity in our data.

Next, we attempted a stepwise logistic regression. We performed this analysis solely to gain insight into which variables were the most significant. Based on the slight increase in AIC when we added “citric acid” into our stepwise regression, it indicated that our model would perform better if we removed the predictor in its entirety. This information led us to exclude “citric acid” in the rest of our analyses.

Naturally, we attempted a polynomial logistic regression model and a polynomial linear regression model next. The error rates for the polynomial logistic model and the polynomial linear model were about .478 and .481, respectively. Once again, the logistic classification model outperformed the linear regression model, but only ever so slightly. However, we are confident that these models were not strong due to the underlying assumptions they have regarding the linearity of the data.

Since the linear models were not performing well on our test data, we decided to try a support vector machine model next. An appealing feature of a support vector machine model is that it can perform well with a non-linear boundary so long as the appropriate kernel is used. For this reason, we chose a radial kernel to fit our data. We modeled the data twice, once with quality as a factor and the second time with quality as a numeric outcome, and received error rates of about .465 and .446 respectively. Both models had lower error rates than the linear models before it, suggesting that we should focus primarily on models that do not assume linearity.

After trying to model our data using Support Vector Machines, we wanted to see how K-Nearest Neighbors would perform on our data. We predicted that K.N.N. would perform at least a little better than our previous models, since it is a nonparametric method that makes no assumptions about the true form of the data. As expected, our accuracy did increase by a few percentage points, since our error rates for K.N.N. regression and classifier were about .420 and .424, respectively. However, we believed that this model would not be our best due to the fact that each rating or “class” did not show up with equal probability, and K.N.N. suffers when the class distributions are not normal.

The almost four percent decrease in error rate that stemmed from our K.N.N. models inspired us to pursue additional non-parametric methods to best fit our data. Next, we tried a pruned decision tree. When we treated “quality” as a numeric outcome, the test error rate was about .434. The tree initially divided the data into wines that had alcohol contents less than 10, and wines that did not. If the alcohol content was less than 10, then the model divided that subset of data by wine that had a volatile acidity of .26 or higher, and wines that did not. But, if the alcohol content was greater than 10, the tree branches continued to divide the remaining data by alcohol content less than 12, then residual sugar less than 2.2. This does confirm our initial hypothesis that alcohol and volatile acidity are two significant predictors. It also brings to our attention that “residual sugar” is one of the most important variables in our model. However, an apparent issue with this model is that it only predicts quality ratings of “5-7,” even though we know that the less frequent values of 3,4,8, and 9 do exist in our dataset. Similarly, when we treated quality as a factor, we had a test error rate of approximately .453. But, once again, the pruned tree does not predict the less common quality ratings that occur within the dataset.

Finally, we attempted to model our data by creating random forests. When we treated quality as a factor, we obtained a test error rate of about .363. However, when we treated quality as a numeric variable, we achieved a test error rate of approximately .355, which implies the model is about 65% accurate. Thus, the random forests model that treated quality as a numeric outcome was ultimately our best model. We believe that this model outperformed the others due to the fact that random forest models are typically more flexible and they lower variance fairly quickly.

Initially, we suspected that the predictors “alcohol,” “volatile acidity,” and “density” would be significant independent variables in our final model. As shown by the pruned tree, “alcohol” and “volatile acidity” did indeed prove to be important predictors in wine quality. However, a variable that we initially did not consider to be that important, “residual sugar,” played more of a role in our models than density did. Also, in a majority of the models, we learned that treating quality as either numeric or a factor did not change the test error rate by much. Another point that should be made about our analyses is that the issues of multicollinearity, outliers, and non-scaled data were not explored. Furthermore, the primary issue that we encountered when we tried to create models is that the data is imbalanced. As previously stated in our discussion of the K.N.N models, the imbalanced data made it nearly impossible for the models to predict the less frequent ratings of 3,4,8, and 9. Therefore, we believe that our analyses could be improved with additional data, more specifically data on wines with “lower” or “higher” ratings. Additionally, if we could have binned the ratings together into “below average,” “average,” and “above average,” we could have achieved models with better accuracy. Finally, two other factors that could have improved our analyses are more predictors and creating two separate models: one that predicts white wine quality and one that predicts red wine quality.

Resources:

https://www.livescience.com/48958-human-origins-alcohol-consumption.html

https://www.wineinstitute.org/resources/statistics/article86

https://ageconsearch.umn.edu/bitstream/99488/2/Factors%20pg%2016-30.pdf

http://waterhouse.ucdavis.edu/whats-in-wine/fixed-acidity

https://winefolly.com/update/sugar-in-wine-misunderstanding/

http://www.oiv.int/public/medias/2604/oiv-ma-d1-03.pdf

http://wineoscope.com/2015/10/02/when-a-wine-is-salty-and-why-it-shouldnt-be/

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3472855/

https://www.etslabs.com/analyses/DEN

https://winefolly.com/review/understanding-acidity-in-wine/

https://www.winespectator.com/drvinny/show/id/5035