Final Project

Thomas Kwok, Yanli Liu

November 20, 2018

```
setwd("C:/Users/thoma/Desktop/Wine Project/transformed")
train_data <- read.csv(file="train_data.csv", head=TRUE, sep=",")
test_data <- read.csv(file="test_data.csv", head=TRUE, sep=",")
```

Introduction:

Wine, many people drink it and all have their own rating system for it. It has been a hot topic studied for ages now, as people try to find what is the best conditions and formulations to make the best wine. It is a hot industry that generates a lot of revenue, with many successful vineyards around the world. The study of this research paper is on wine, and in this research, we observed twelve predictors to see how they relate to quality. There were approximately 6500 samples taken and 6500 different qualities listed. The specific data we had to work with was approximately 4600 samples with their own qualities assigned, and the remaining 1900 was unknown. Our hope was to create a model that could sufficiently predict the quality of wine.

From our data, we noticed that the first category was wine type; with a majority being white wine. This is particularly interesting because the process of making white wine and red wine do differ, which could mean that the formulation to make a good white wine may not be the same for red wine. The quality of the wine was also broken down into a scale from 1-10, with one being the worst and ten being the best. This was interesting because judging wine as good, bad, or average was already difficult; but to give it a numeric score? This could lead to even more variability.

According to many sources, the five fundamental traits of wine are sweetness, acidity, tannin, alcohol content, and body. This led us to first look at the acidities, sugar residual, and alcohol predictors in our data as those closely related to these fundamental traits. Our goal was to find the best method for predicting wine quality from the dataset. Our goal in this data was to find the best model that produces the highest accuracy and least error for predicting wine quality, if this model also let us know the predictors most correlated then it would be a win-win, but the main study was on the model and not as much on specific predictors.

Description of Data

The first thing we did with our data was run a summary of the training data we had. This gave us a minimum, maximum, median, mean, and first and third quadrant breakdown for each of the thirteen predictors. From this, we found out that the train data was approximately 76% white and 24% red. The quality of wine also ranged from a minimum of three to a maximum of nine, but most were graded around the median of six. We also noticed that some categories had a huge difference between the values: for example sulfur dioxide predictors had the minimum and maximum differ by more than a hundred. Others had a much smaller range, like acid, which means that scaling will be needed. Initially, we paid particular attention to the acidity categories (fixed, volatile, and citric), residual sugar, and alcohol, as those were the ones that most closely related to the fundamental traits of wine.

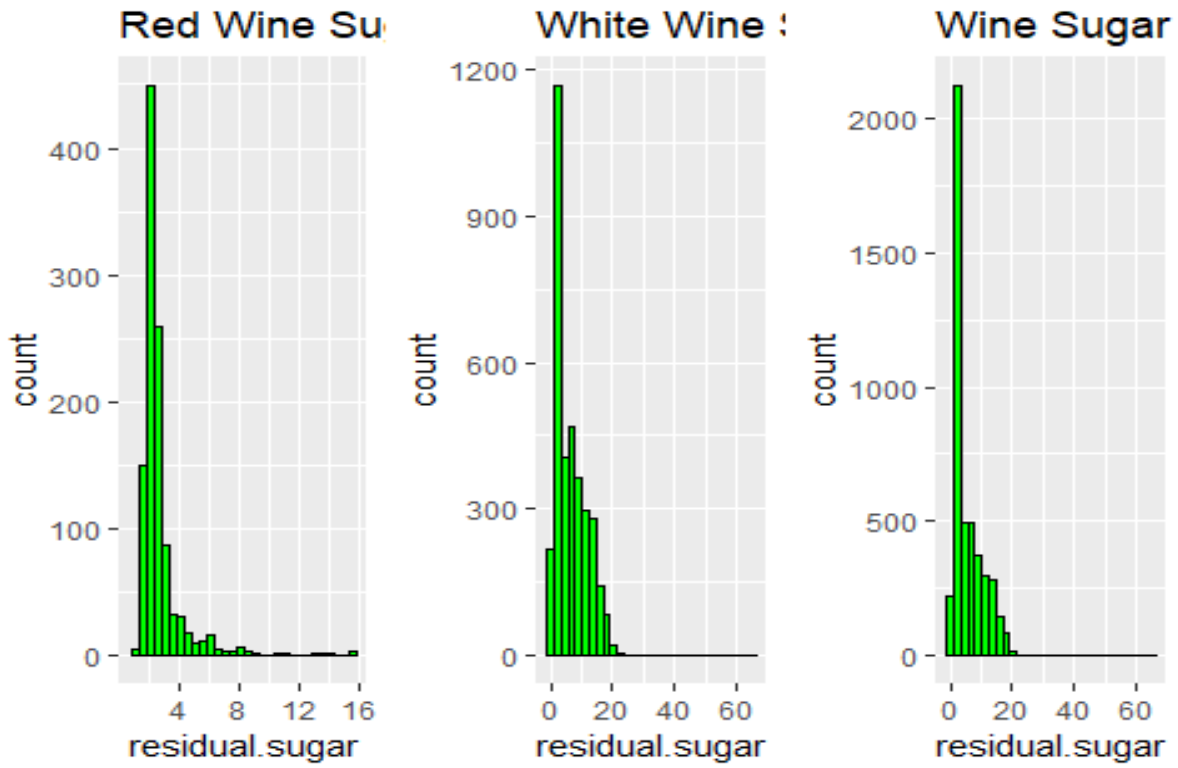
```
library(rpart)
library(rpart.plot)
library(tidyverse)
library(kernlab)
library(caret)
library(gridExtra)
library(pander)
library(glmnet)
library(randomForest)

wine <- train_data
summary(wine)
```

```
##  wine_type    fixed.acidity   volatile.acidity  citric.acid
##  red  :1096   Min.   : 3.90   Min.   :0.0800   Min.   :0.0000
##  white:3451   1st Qu.: 6.40   1st Qu.:0.2250   1st Qu.:0.2500
##               Median : 7.00   Median :0.2900   Median :0.3100
##               Mean   : 7.21   Mean   :0.3375   Mean   :0.3193
##               3rd Qu.: 7.70   3rd Qu.:0.4000   3rd Qu.:0.3900
##               Max.   :15.90   Max.   :1.5800   Max.   :1.2300
##  residual.sugar     chlorides       free.sulfur.dioxide
##  Min.   : 0.600   Min.   :0.00900   Min.   :  1.00
##  1st Qu.: 1.800   1st Qu.:0.03800   1st Qu.: 17.00
##  Median : 3.100   Median :0.04700   Median : 29.00
##  Mean   : 5.455   Mean   :0.05584   Mean   : 30.64
##  3rd Qu.: 8.125   3rd Qu.:0.06400   3rd Qu.: 42.00
##  Max.   :65.800   Max.   :0.61100   Max.   :289.00
##  total.sulfur.dioxide    density             pH          sulphates
##  Min.   :  6.0        Min.   :0.9874   Min.   :2.740   Min.   :0.2300
##  1st Qu.: 78.0        1st Qu.:0.9923   1st Qu.:3.110   1st Qu.:0.4300
##  Median :119.0        Median :0.9949   Median :3.200   Median :0.5000
##  Mean   :116.4        Mean   :0.9947   Mean   :3.216   Mean   :0.5285
##  3rd Qu.:156.0        3rd Qu.:0.9969   3rd Qu.:3.320   3rd Qu.:0.6000
##  Max.   :440.0        Max.   :1.0390   Max.   :4.010   Max.   :1.9800
##     alcohol         quality
##  Min.   : 8.00   Min.   :3.000
##  1st Qu.: 9.50   1st Qu.:5.000
##  Median :10.30   Median :6.000
##  Mean   :10.48   Mean   :5.821
##  3rd Qu.:11.30   3rd Qu.:6.000
##  Max.   :14.90   Max.   :9.000
```

The first thing we did was to make a visual representation of the data, in terms of histograms. We first looked at residual sugar level between the red and white wine, and then all the wine. From that, we saw that both red and white wine had a right tail skew, and the overall data had a right tail skew also. One thing to note though is that the x-value scale wasn't the same for each breakdown, though that is likely more to do with the fact that the outlier 65.8 residual sugar is part of the white wine so its x-values were more spread out.

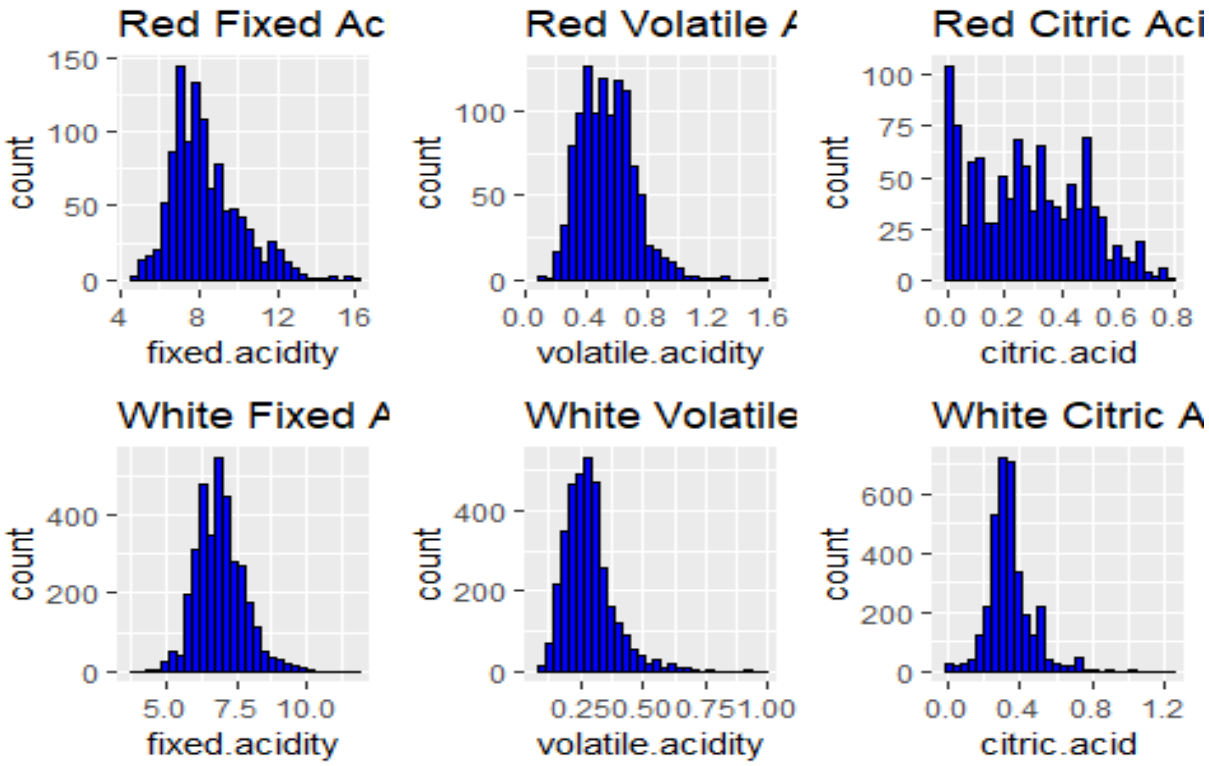
```
p1 <- ggplot(aes(x=residual.sugar), data = subset(wine, wine_type %in%
c("red")))+geom_histogram(color = I('black'), fill = I('green'))+ggtitle('Red
Wine Sugar')
p2 <- ggplot(aes(x=residual.sugar), data = subset(wine, wine_type %in%
c("white")))+geom_histogram(color = I('black'), fill =
I('green'))+ggtitle('White Wine Sugar')
```

```
p3 <- ggplot(data=wine, aes(x=residual.sugar))+geom_histogram(color='black',
fill=I('green'))+ggtitle('Wine Sugar')
grid.arrange(p1, p2, p3, ncol=3)
```

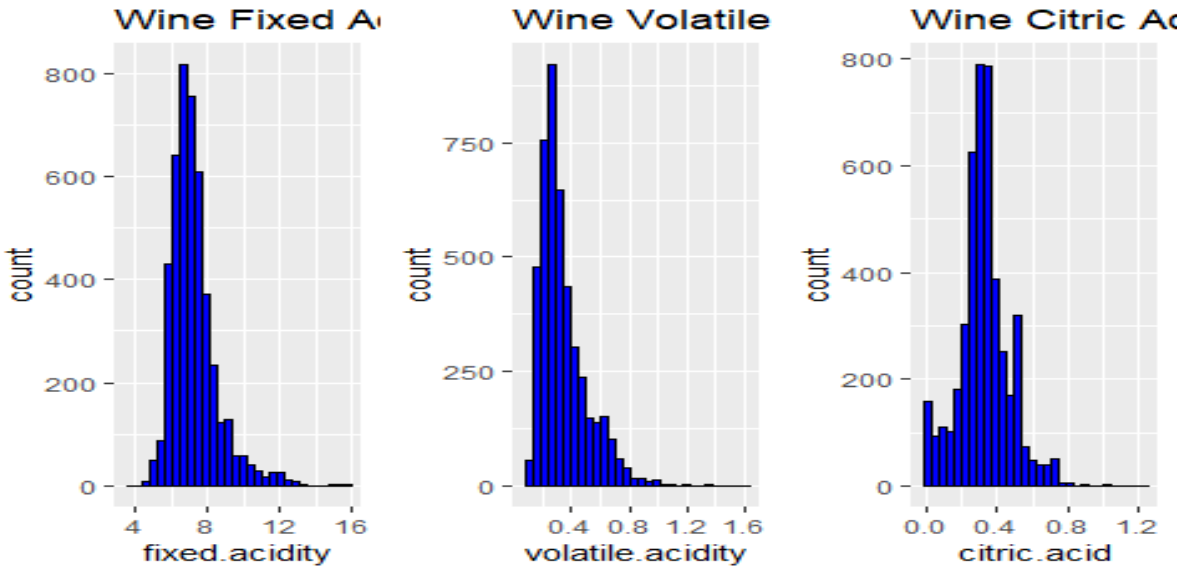


The next thing studied was the breakdown in difference in the acid predictors between red and white wine, and total wine acidity. The fixed acidity for red and white seemed to be about the same and the volatile data for both showed somewhat of a right tail skewness. Citric Acid between the two was very different though, which was interesting. When looking at all the wines, most of the data supported the white wine data, though that is likely because the majority of the data were white wine.

```
p4 <- ggplot(aes(x=fixed.acidity), data = subset(wine, wine_type %in%
c("red")))+geom_histogram(color = I('black'), fill = I('blue'))+ggtitle('Red
Fixed Acidity')
p5 <- ggplot(aes(x=volatile.acidity), data = subset(wine, wine_type %in%
c("red")))+geom_histogram(color = I('black'), fill = I('blue'))+ggtitle('Red
Volatile Acidity')
p6 <- ggplot(aes(x=citric.acid), data = subset(wine, wine_type %in%
c("red")))+geom_histogram(color = I('black'), fill = I('blue'))+ggtitle('Red
Citric Acid')
p7 <- ggplot(aes(x=fixed.acidity), data = subset(wine, wine_type %in%
c("white")))+geom_histogram(color = I('black'), fill =
I('blue'))+ggtitle('White Fixed Acidity')
p8 <- ggplot(aes(x=volatile.acidity), data = subset(wine, wine_type %in%
c("white")))+geom_histogram(color = I('black'), fill =
I('blue'))+ggtitle('White Volatile Acidity')
p9 <- ggplot(aes(x=citric.acid), data = subset(wine, wine_type %in%
c("white")))+geom_histogram(color = I('black'), fill =
I('blue'))+ggtitle('White Citric Acid')

grid.arrange(p4, p5, p6, p7, p8, p9, ncol=3)
```

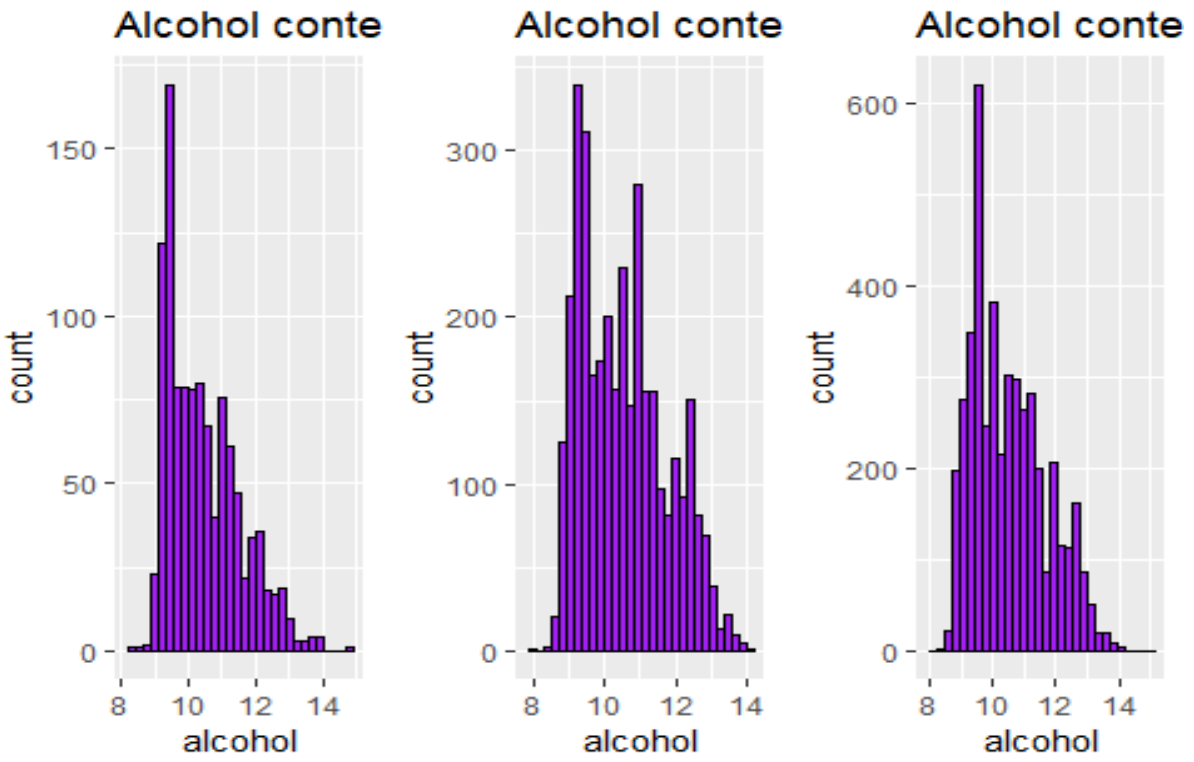



```
p10 <- ggplot(data = wine, aes(x = fixed.acidity)) +
  geom_histogram(color = 'black',fill = I('blue'))+ggtitle('Wine Fixed
Acidity')
p11 <- ggplot(data = wine, aes(x =
volatile.acidity))+geom_histogram(color='black', fill=I('blue'))+ggtitle('Wine
Volatile Acidity')
p12 <- ggplot(data=wine, aes(x=citric.acid))+geom_histogram(color='black',
fill=I('blue'))+ggtitle('Wine Citric Acid')
grid.arrange(p10, p11, p12, ncol=3)
```



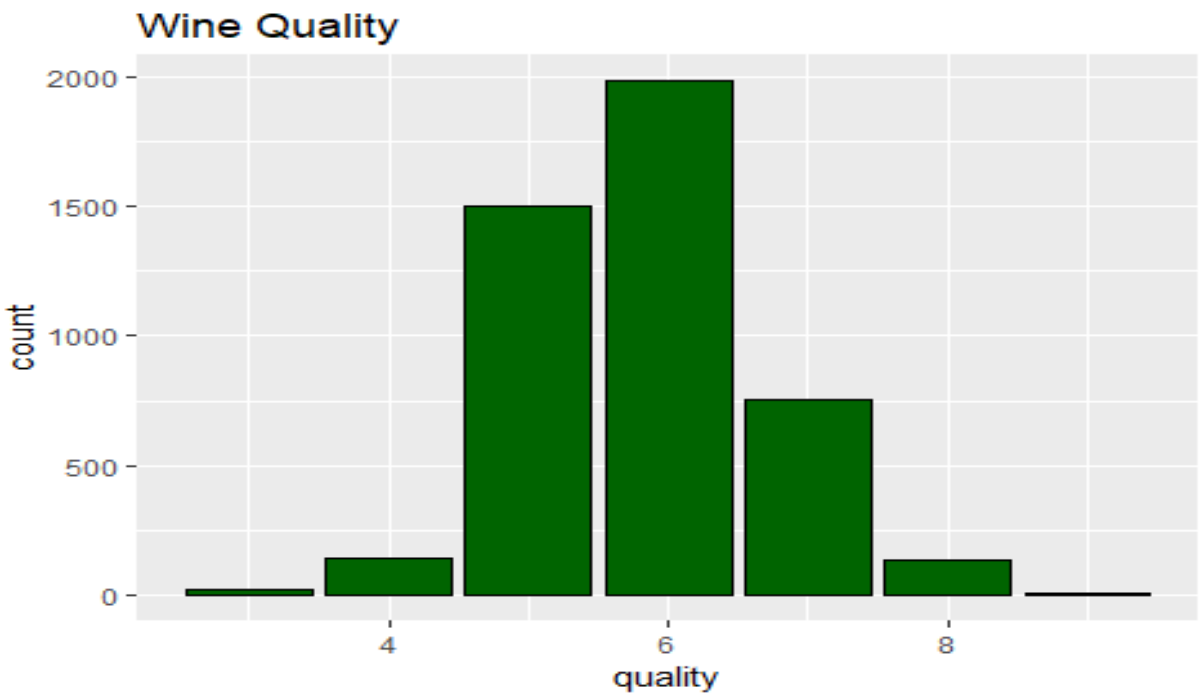
Next we ran a histogram of alcohol content between the red and white and all the wine, most of the wine fell between the nine to twelve percent alcohol content, with many of the red below ten percent alcohol content and white wine either below ten percent or between ten and twelve percent. When looking at all wines, most fell below ten percent like the red wine.

```
p13 <- ggplot(aes(x=alcohol), data = subset(wine, wine_type %in%
c("red")))+geom_histogram(color = I('black'), fill =
I('purple'))+ggtitle('Alcohol content Red')
p14 <- ggplot(aes(x=alcohol), data = subset(wine, wine_type %in%
c("white")))+geom_histogram(color = I('black'), fill =
I('purple'))+ggtitle('Alcohol content White')
p15 <- ggplot(data = wine, aes(x = alcohol)) +
  geom_histogram(color = 'black',fill = I('purple'))+ggtitle('Alcohol content
Wine')
grid.arrange(p13, p14, p15, ncol=3)
```



Lastly we ran a bar graph on overall wine quality and this was when the data became interesting. Most data fell either in the five or six range. There were few rated at three or nine, and four and eight were also relatively low. The graph as a whole looked like a normal curve, with the majority falling between five, six, and seven. This made us consider doing some classification of wine later, to look at the quality as a bad, average, and good wine breakdown.

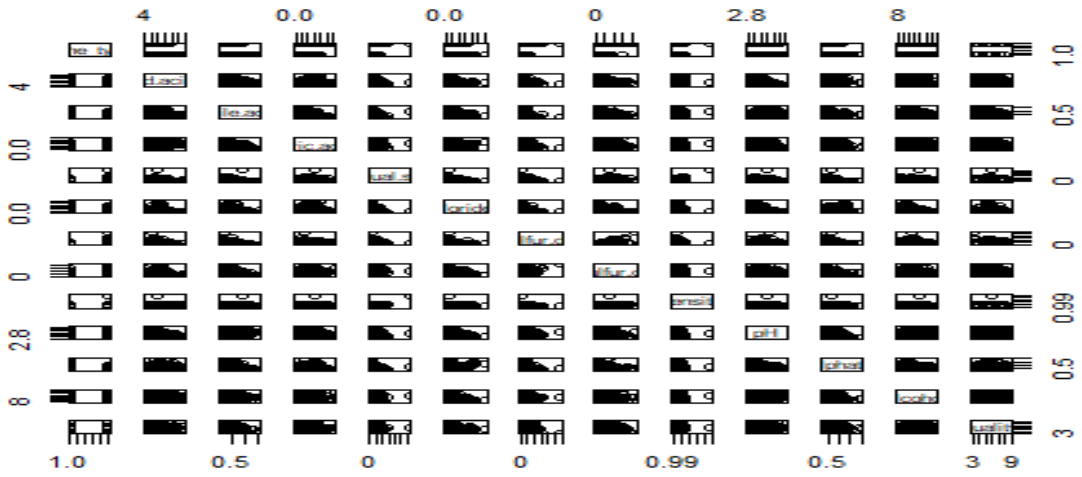
```
ggplot(data = wine, aes(x = quality)) +
  geom_bar(color = 'black',fill = I('dark green'))+ggtitle('Wine Quality')
```




```
table(wine$quality)
##    3    4    5    6    7    8    9
##   23  140 1505 1985  754  136    4
```

Lastly, we ran a correlation plot and matrix of the wine to see which predictors were positively and negatively correlated with one another, and which ones were correlated with the quality. We wanted to see if residual sugar, fixed acidity, volatile acidity, citric acid, and alcohol content actually had any correlation with quality. From our result, we found that alcohol had the most correlation to quality; a positive correlation meaning that the higher the alcohol content, the better the wine. One thing to note from the data was that the ones with the most correlation to quality was volatile acidity, chlorides, density, and alcohol. Though, the former three have a negative correlation with quality and the latter has a positive correlation. But this plot and table led to a model we ran often later.

```
pairs(wine)
```



```
wine_numeric <- cor(
  wine %>%
    dplyr::select(-wine_type)
)
emphasize.strong.cells(which(abs(wine_numeric) > .3 & wine_numeric != 1,
arr.ind = TRUE))
pandoc.table(wine_numeric)
```

```
## ------------------------------------------------------------------------
##                        fixed.acidity   volatile.acidity   citric.acid
## ------------------------- -------------- ------------------ -------------
##     **fixed.acidity**             1              0.2193        **0.3238**
##
##    **volatile.acidity**         0.2193              1           **-0.3655**
##
##      **citric.acid**          **0.3238**       **-0.3655**          1
##
##     **residual.sugar**          -0.1105          -0.1859           0.138
##
##       **chlorides**             0.2904           **0.385**         0.03153
##
##  **free.sulfur.dioxide**        -0.2836         **-0.3461**        0.1287
##
##  **total.sulfur.dioxide**     **-0.3212**       **-0.4061**        0.1992
##
##        **density**            **0.4554**          0.2784           0.09554
##
##           **pH**                -0.2404           0.2663         **-0.3187**
##
##       **sulphates**            **0.31**           0.2275           0.06085
##
##        **alcohol**             -0.08662          -0.03961        -0.006977
##
##        **quality**             -0.07724          -0.2724           0.08548
## ------------------------------------------------------------------------
## Table: Table continues below
## ------------------------------------------------------------------------
--
##                       residual.sugar   chlorides    free.sulfur.dioxide
## ------------------------- --------------- ------------ -------------------
--
##     **fixed.acidity**          -0.1105         0.2904           -0.2836
##
##    **volatile.acidity**        -0.1859        **0.385**        **-0.3461**
##
##      **citric.acid**            0.138          0.03153           0.1287
##
##     **residual.sugar**             1           -0.1251         **0.3973**
##
##       **chlorides**             -0.1251            1              -0.1979
##
##  **free.sulfur.dioxide**       **0.3973**       -0.1979             1
##
##  **total.sulfur.dioxide**      **0.4868**       -0.2757         **0.7263**
##
```

```
##        **density**             **0.5568**     **0.3605**          0.02705
##
##           **pH**                -0.2643         0.05433           -0.136
##
##       **sulphates**             -0.1844        **0.3925**         -0.1897
##
##        **alcohol**            **-0.3578**       -0.2604           -0.1843
##
##        **quality**              -0.03888        -0.1977           0.05253
## ------------------------------------------------------------------------
## Table: Table continues below
## ------------------------------------------------------------------------
-
##                         total.sulfur.dioxide     density          pH
## ------------------------- ---------------------- ------------- ------------
-
##     **fixed.acidity**            **-0.3212**         **0.4554**       -0.2404
##
##    **volatile.acidity**          **-0.4061**           0.2784          0.2663
##
##      **citric.acid**                0.1992             0.09554      **-0.3187**
##
##     **residual.sugar**            **0.4868**         **0.5568**       -0.2643
##
##       **chlorides**                -0.2757           **0.3605**       0.05433
##
##  **free.sulfur.dioxide**           **0.7263**           0.02705         -0.136
##
##  **total.sulfur.dioxide**             1                 0.03651         -0.2324
##
##        **density**                 0.03651                1            0.02599
##
##           **pH**                   -0.2324             0.02599            1
##
##       **sulphates**                -0.2758             0.2698          0.2073
##
##        **alcohol**                 -0.2657           **-0.6799**       0.1094
##
##        **quality**                 -0.0454             -0.2997         0.02299
## ------------------------------------------------------------------------
## Table: Table continues below
## ------------------------------------------------------------------
##                         sulphates      alcohol      quality
## ------------------------- ------------ ------------- ------------
##     **fixed.acidity**          **0.31**      -0.08662      -0.07724
##
##    **volatile.acidity**         0.2275       -0.03961      -0.2724
##
##      **citric.acid**           0.06085       -0.006977     0.08548
##
##     **residual.sugar**         -0.1844      **-0.3578**    -0.03888
##
##       **chlorides**           **0.3925**       -0.2604      -0.1977
##
##  **free.sulfur.dioxide**       -0.1897         -0.1843      0.05253
##
```

```
##  **total.sulfur.dioxide**    -0.2758       -0.2657      -0.0454
##
##        **density**            0.2698     **-0.6799**    -0.2997
##
##           **pH**              0.2073       0.1094       0.02299
##
##       **sulphates**             1          -0.01265      0.03639
##
##        **alcohol**           -0.01265         1        **0.4314**
##
##        **quality**           0.03639     **0.4314**        1
## ---------------------------------------------------------------
```

Methods and Results:

It should be noted that before we ran any model, we took a look at the data and found there were two null values for total sulfur dioxide. In order to make all data relevant, we decided to fill in those values by making them the average of sulfur dioxide. This could have impacted the data in some way, but with over 4600 points, we felt that this would only lead to a small error.

The first model we ran on our data was a linear regression model, which tested the quality against all predictors. The R-squared measurement, which measures how close the model fits a linear line was 0.2891. This meant that only 28.91% of the data could be explained with a linear line. The residual standard error is 0.7365 and the F-statistic was low.

After that, we ran a step function to see how the values individually impacted quality. The result we saw was that alcohol was the best predictor, followed by volatile.acidity, sulphates, residual sugar, and then wine type. An interesting note is that citric acid is the only predictor to not effect the data at all in the end. This differed from the correlation matrix we ran earlier, which means that the correlation matrix likely had some multicollinearity.

```
stepfunction <- step(lm(quality ~ 1, wine), scope=list(lower=~1, upper =
~wine_type+fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+
free.sulfur.dioxide+total.sulfur.dioxide+density+pH+sulphates+alcohol),
direction = "forward")
```

```
## Start:  AIC=-1240.81
## quality ~ 1
##
##                        Df Sum of Sq    RSS     AIC
## + alcohol               1    643.74 2815.8 -2175.0
## + density               1    310.75 3148.8 -1666.8
## + volatile.acidity      1    256.71 3202.8 -1589.4
## + chlorides             1    135.16 3324.4 -1420.0
## + wine_type             1     51.17 3408.4 -1306.6
## + citric.acid           1     25.28 3434.3 -1272.2
## + fixed.acidity         1     20.64 3438.9 -1266.0
## + free.sulfur.dioxide   1      9.55 3450.0 -1251.4
## + total.sulfur.dioxide  1      7.13 3452.4 -1248.2
## + residual.sugar        1      5.23 3454.3 -1245.7
## + sulphates             1      4.58 3455.0 -1244.8
## + pH                    1      1.83 3457.7 -1241.2
## <none>                              3459.6 -1240.8
##
## Step:  AIC=-2174.99
```

```
## quality ~ alcohol
##
##                         Df Sum of Sq    RSS     AIC
## + volatile.acidity       1   225.872 2589.9 -2553.2
## + free.sulfur.dioxide    1    62.435 2753.4 -2274.9
## + residual.sugar         1    52.894 2762.9 -2259.2
## + wine_type              1    39.555 2776.3 -2237.3
## + citric.acid            1    27.093 2788.7 -2216.9
## + chlorides              1    27.029 2788.8 -2216.8
## + total.sulfur.dioxide   1    17.836 2798.0 -2201.9
## + sulphates              1     6.058 2809.8 -2182.8
## + fixed.acidity          1     5.541 2810.3 -2181.9
## + pH                     1     2.049 2813.8 -2176.3
## <none>                               2815.8 -2175.0
## + density                1     0.264 2815.6 -2173.4
##
## Step:  AIC=-2553.19
## quality ~ alcohol + volatile.acidity
##
##                         Df Sum of Sq    RSS     AIC
## + sulphates              1    36.397 2553.6 -2615.5
## + density                1    24.452 2565.5 -2594.3
## + wine_type              1    21.644 2568.3 -2589.3
## + residual.sugar         1    17.196 2572.8 -2581.5
## + pH                     1     7.663 2582.3 -2564.7
## + free.sulfur.dioxide    1     7.143 2582.8 -2563.7
## + total.sulfur.dioxide   1     6.375 2583.6 -2562.4
## <none>                               2589.9 -2553.2
## + fixed.acidity          1     0.860 2589.1 -2552.7
## + chlorides              1     0.479 2589.5 -2552.0
## + citric.acid            1     0.102 2589.8 -2551.4
##
## Step:  AIC=-2615.54
## quality ~ alcohol + volatile.acidity + sulphates
##
##                         Df Sum of Sq    RSS     AIC
## + residual.sugar         1   26.9311 2526.6 -2661.8
## + free.sulfur.dioxide    1   11.9324 2541.6 -2634.8
## + density                1   10.6565 2542.9 -2632.6
## + wine_type              1    4.2862 2549.3 -2621.2
## + pH                     1    3.3779 2550.2 -2619.6
## + chlorides              1    2.3166 2551.2 -2617.7
## + citric.acid            1    1.6742 2551.9 -2616.5
## + total.sulfur.dioxide   1    1.5371 2552.0 -2616.3
## <none>                               2553.6 -2615.5
## + fixed.acidity          1    0.5732 2553.0 -2614.6
##
## Step:  AIC=-2661.75
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar
##
##                         Df Sum of Sq    RSS     AIC
## + wine_type              1   13.6801 2512.9 -2684.4
## + total.sulfur.dioxide   1   11.4317 2515.2 -2680.4
## + pH                     1    7.7751 2518.8 -2673.8
## + free.sulfur.dioxide    1    4.0036 2522.6 -2667.0
## + citric.acid            1    3.3509 2523.3 -2665.8
```

```
## <none>                                2526.6 -2661.8
## + chlorides             1    0.7345 2525.9 -2661.1
## + fixed.acidity         1    0.1559 2526.5 -2660.0
## + density               1    0.0754 2526.5 -2659.9
##
## Step:  AIC=-2684.44
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar +
##     wine_type
##
##                         Df Sum of Sq    RSS     AIC
## + density                1   14.1135 2498.8 -2708.1
## + free.sulfur.dioxide    1    9.9062 2503.0 -2700.4
## + pH                     1    5.4572 2507.5 -2692.3
## + fixed.acidity          1    3.9328 2509.0 -2689.6
## + citric.acid            1    3.8652 2509.1 -2689.4
## + chlorides              1    3.2758 2509.7 -2688.4
## + total.sulfur.dioxide   1    2.1570 2510.8 -2686.3
## <none>                                2512.9 -2684.4
##
## Step:  AIC=-2708.05
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar +
##     wine_type + density
##
##                         Df Sum of Sq    RSS     AIC
## + free.sulfur.dioxide    1    8.1087 2490.7 -2720.8
## + pH                     1    7.4751 2491.3 -2719.7
## + chlorides              1    3.0495 2495.8 -2711.6
## + total.sulfur.dioxide   1    1.3630 2497.5 -2708.5
## <none>                                2498.8 -2708.1
## + citric.acid            1    0.7452 2498.1 -2707.4
## + fixed.acidity          1    0.2581 2498.6 -2706.5
##
## Step:  AIC=-2720.82
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar +
##     wine_type + density + free.sulfur.dioxide
##
##                         Df Sum of Sq    RSS     AIC
## + total.sulfur.dioxide   1   12.8267 2477.9 -2742.3
## + pH                     1    6.0488 2484.7 -2729.9
## + chlorides              1    3.0640 2487.7 -2724.4
## <none>                                2490.7 -2720.8
## + citric.acid            1    0.8636 2489.8 -2720.4
## + fixed.acidity          1    0.5940 2490.1 -2719.9
##
## Step:  AIC=-2742.3
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar +
##     wine_type + density + free.sulfur.dioxide + total.sulfur.dioxide
##
##                 Df Sum of Sq    RSS     AIC
## + pH             1    6.2585 2471.6 -2751.8
## + chlorides      1    3.1482 2474.7 -2746.1
## <none>                       2477.9 -2742.3
## + citric.acid    1    0.3953 2477.5 -2741.0
## + fixed.acidity  1    0.3289 2477.6 -2740.9
##
## Step:  AIC=-2751.8
```

```
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar +
##     wine_type + density + free.sulfur.dioxide + total.sulfur.dioxide +
##     pH
##
##                  Df Sum of Sq    RSS     AIC
## + fixed.acidity   1   11.0868 2460.6 -2770.2
## + chlorides       1    2.0097 2469.6 -2753.5
## <none>                        2471.6 -2751.8
## + citric.acid     1    0.0399 2471.6 -2749.9
##
## Step:  AIC=-2770.24
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar +
##     wine_type + density + free.sulfur.dioxide + total.sulfur.dioxide +
##     pH + fixed.acidity
##
##                Df Sum of Sq    RSS     AIC
## + chlorides     1   1.13737 2459.4 -2770.3
## <none>                      2460.6 -2770.2
## + citric.acid   1   0.22705 2460.3 -2768.7
##
## Step:  AIC=-2770.34
## quality ~ alcohol + volatile.acidity + sulphates + residual.sugar +
##     wine_type + density + free.sulfur.dioxide + total.sulfur.dioxide +
##     pH + fixed.acidity + chlorides
##
##                Df Sum of Sq    RSS     AIC
## <none>                      2459.4 -2770.3
## + citric.acid   1  0.096159 2459.3 -2768.5
```

```
summary(stepfunction)
```

```
##
## Call:
## lm(formula = quality ~ alcohol + volatile.acidity + sulphates +
##     residual.sugar + wine_type + density + free.sulfur.dioxide +
##     total.sulfur.dioxide + pH + fixed.acidity + chlorides, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7806 -0.4600 -0.0416  0.4532  3.0136
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.008e+02  1.642e+01   6.141 8.88e-10 ***
## alcohol               2.157e-01  2.094e-02  10.301  < 2e-16 ***
## volatile.acidity     -1.516e+00  9.246e-02 -16.397  < 2e-16 ***
## sulphates             7.349e-01  9.340e-02   7.868 4.46e-15 ***
## residual.sugar        6.079e-02  6.988e-03   8.699  < 2e-16 ***
## wine_typewhite       -3.125e-01  6.630e-02  -4.714 2.50e-06 ***
## density              -1.001e+02  1.666e+01  -6.008 2.03e-09 ***
## free.sulfur.dioxide   5.253e-03  9.239e-04   5.685 1.39e-08 ***
## total.sulfur.dioxide -1.754e-03  3.806e-04  -4.608 4.17e-06 ***
## pH                    5.627e-01  1.072e-01   5.249 1.60e-07 ***
## fixed.acidity         7.923e-02  1.826e-02   4.340 1.46e-05 ***
## chlorides            -5.832e-01  4.027e-01  -1.448    0.148
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.7364 on 4535 degrees of freedom
## Multiple R-squared:  0.2891, Adjusted R-squared:  0.2874
## F-statistic: 167.7 on 11 and 4535 DF,  p-value: < 2.2e-16
```

Next, we ran individual linear models to further break down the data. The first linear model looked at quality and alcohol alone, since alcohol had the biggest correlation to quality. The second compared quality to the predictors that had a positive correlation from our matrix, the third tested quality against the predictions that had the biggest correlation regardless of sign in the matrix, and the fourth tested quality against the first four predictors from the step function data. Alcohol alone had an R-squared of 0.18, while the positive correlants increased the R-square to 0.21. The third model increased the R-square again to 0.258, while the step function predictors increased had the highest R-square at 0.27. None of the data increased the R-square greater than the first linear model with all the predictors, but that makes sense as there was no penalty for using more predictors.

```
lm1 <- lm(quality ~ alcohol, data = wine)
lm2 <- lm(quality ~ alcohol+sulphates+pH+free.sulfur.dioxide+citric.acid,
data=wine)
lm3 <- lm(quality ~ alcohol+volatile.acidity+density+chlorides, data=wine)
lm4 <- lm(quality ~ alcohol+volatile.acidity+sulphates+residual.sugar,
data=wine)
summary(lm1)

##
## Call:
## lm(formula = quality ~ alcohol, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4927 -0.5094 -0.0486  0.5073  3.2051
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.49596    0.10380   24.05   <2e-16 ***
## alcohol      0.31720    0.00984   32.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7871 on 4545 degrees of freedom
## Multiple R-squared:  0.1861, Adjusted R-squared:  0.1859
## F-statistic:  1039 on 1 and 4545 DF,  p-value: < 2.2e-16

summary(lm2)

##
## Call:
## lm(formula = quality ~ alcohol + sulphates + pH + free.sulfur.dioxide +
##     citric.acid, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6256 -0.5016 -0.0501  0.4872  3.1986
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)          1.7453355  0.2687938   6.493 9.30e-11 ***
## alcohol              0.3372494  0.0099233  33.986  < 2e-16 ***
## sulphates            0.3875001  0.0827005   4.686 2.87e-06 ***
## pH                  -0.0015345  0.0781308  -0.020    0.984
## free.sulfur.dioxide  0.0069043  0.0006772  10.195  < 2e-16 ***
## citric.acid          0.4044647  0.0859550   4.706 2.61e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7742 on 4541 degrees of freedom
## Multiple R-squared:  0.2133, Adjusted R-squared:  0.2124
## F-statistic: 246.2 on 5 and 4541 DF,  p-value: < 2.2e-16
```

```
summary(lm3)
```

```
##
## Call:
## lm(formula = quality ~ alcohol + volatile.acidity + density +
##     chlorides, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4145 -0.4830 -0.0412  0.4658  3.0242
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -32.90912    5.51252  -5.970 2.56e-09 ***
## alcohol            0.36984    0.01318  28.071  < 2e-16 ***
## volatile.acidity  -1.51973    0.07656 -19.850  < 2e-16 ***
## density           35.55595    5.45763   6.515 8.06e-11 ***
## chlorides         -0.01345    0.36777  -0.037    0.971
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7516 on 4542 degrees of freedom
## Multiple R-squared:  0.2584, Adjusted R-squared:  0.2578
## F-statistic: 395.7 on 4 and 4542 DF,  p-value: < 2.2e-16
```

```
summary(lm4)
```

```
##
## Call:
## lm(formula = quality ~ alcohol + volatile.acidity + sulphates +
##     residual.sugar, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3309 -0.4754 -0.0322  0.4533  3.0685
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.288587   0.126067  18.154  < 2e-16 ***
## alcohol           0.336438   0.010074  33.397  < 2e-16 ***
## volatile.acidity -1.398197   0.070280 -19.895  < 2e-16 ***
## sulphates         0.717454   0.078784   9.107  < 2e-16 ***
## residual.sugar    0.018056   0.002595   6.958 3.95e-12 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7458 on 4542 degrees of freedom
## Multiple R-squared:  0.2697, Adjusted R-squared:  0.269
## F-statistic: 419.3 on 4 and 4542 DF,  p-value: < 2.2e-16
```

Next, we split the data into training and testing dataset. We created a partition of our data, with seventy percent training and thirty percent testing. This meant that 3184 observations were known and used to determine quality for the other 1363 observations. Then we compared our predicted quality to the actual quality from this testing sample. We first ran this method on repeated cross validation using K-Nearest Neighbors. From this result we found that the best result came at K = 15. This had an R-square of 0.35, which was higher than our linear model.

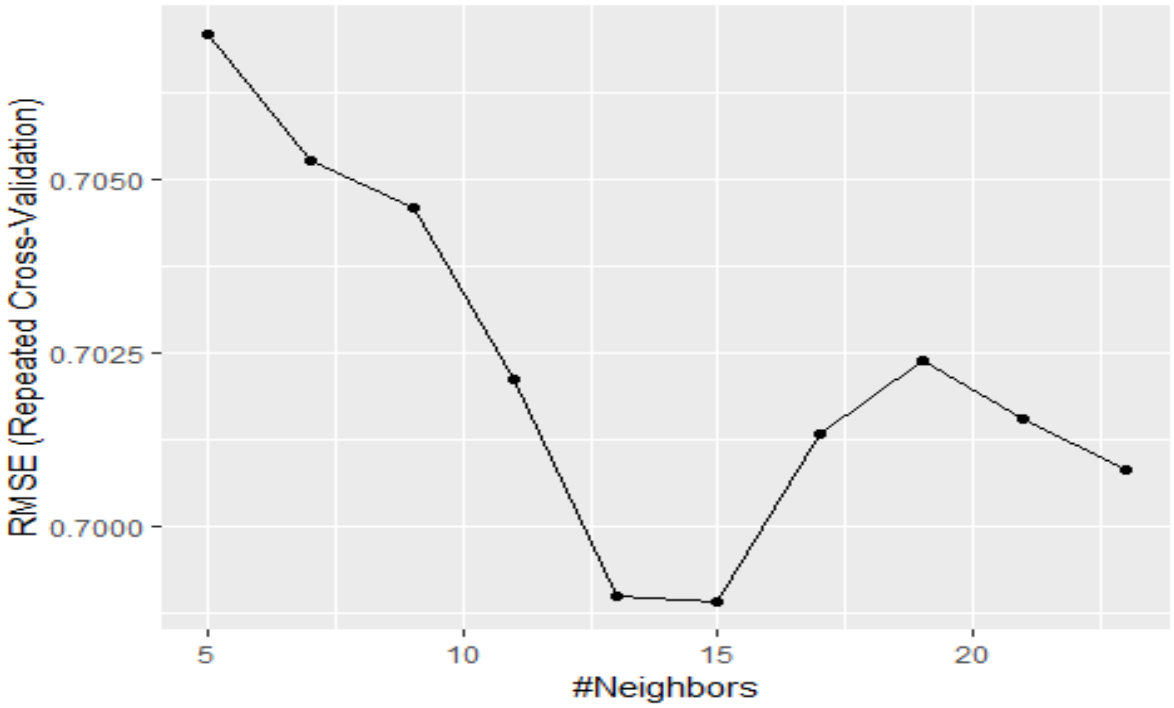
```
set.seed(1234)

sample <- createDataPartition(wine$quality,
                              p = 0.7,
                              list = FALSE)
training <- wine[ sample ,]
testing <- wine[ -sample, ]

trctrl <- trainControl(method = "repeatedcv", number = 10)
knn_fit <- train(quality ~., data = training, method = "knn", trControl=trctrl,
preProcess = c("center", "scale"), tuneLength = 10)
knn_fit
```

```
## k-Nearest Neighbors
##
## 3184 samples
##   12 predictor
##
## Pre-processing: centered (12), scaled (12)
## Resampling: Cross-Validated (10 fold, repeated 1 times)
## Summary of sample sizes: 2866, 2865, 2867, 2866, 2866, 2865, ...
## Resampling results across tuning parameters:
##
##   k   RMSE       Rsquared   MAE
##    5  0.7070818  0.3455450  0.5440655
##    7  0.7052668  0.3418834  0.5467277
##    9  0.7045984  0.3394662  0.5503547
##   11  0.7021234  0.3425838  0.5512339
##   13  0.6989951  0.3479956  0.5510159
##   15  0.6989248  0.3478133  0.5504886
##   17  0.7013256  0.3431795  0.5532384
##   19  0.7023960  0.3414775  0.5553954
##   21  0.7015539  0.3436457  0.5549760
##   23  0.7008191  0.3453385  0.5555221
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 15.
```

From here, we plotted the KNN fit, to show the K Neighbors and mean square error relation. Interestingly, K= 13 was closer to the lowest KNN than k=17. We also made a conscious

decision here to keep seed the same even though changing that value could have possibly led to better accuracy. We also kept the tuning parameter as 10 instead of changing that.

```
ggplot(knn_fit, xlabs = 'K Neighbors', ylab='Accuracy')
```



Next we wanted to run a confusion matrix to see how our model performed, but we ran into an issue of a nonsquare matrix. Our model only showed three predictor quality, 5, 6, and 7, while the actual data had seven. Thus instead of running the confusion matrix function, we manually calculated our accuracy by taking the correct predictions and dividing by the total qualities. Our KNN model had an accurate of 54% in predicting the quality of wine.

We then took our model and ran it against the test data for R-square and Mean Square Error, and this showed that the R-square was 0.24, which is lower than our prior two models' R-square, and a Mean Square Error higher than the other two models also.

```
r2_pred <- predict(knn_fit, newdata = testing)
r2_predround <- round(r2_pred)
r2_knn <- R2(r2_predround, testing$quality)
r2_mse <- RMSE(r2_predround, testing$quality)

r2_knn

## [1] 0.2406392

r2_mse

## [1] 0.7989355

table(r2_predround, testing$quality)

##
## r2_predround   3   4   5   6   7   8   9
##            5   3  20 244 101   3   0   0
```

```
##            6   6  29 193 421 144  23   1
##            7   0   0   5  73  76  20   1

(244+421+76)/(1363)

## [1] 0.5436537
```

Next, we ran a SVM, or Support Vector Machines model, to see how it compared to our KNN. Here we used the same training and testing data, and set a tuning length of 10. Here we got a 64.4% accuracy model and an R-square of ~0.42 and Mean Square Error of 0.68.

```
set.seed(1234)
SVM_1 <- train(quality ~.,
                  data = wine,
                method = "svmRadial",
                tuneLength = 10,
                trControl = trctrl)

SVM_pred <- predict(SVM_1, newdata = testing)
svm_predround <- round(SVM_pred, digits=0)
r2_svm <- R2(svm_predround, testing$quality)
rmse_svm <- RMSE(svm_predround, testing$quality)

r2_svm

## [1] 0.4179606

rmse_svm

## [1] 0.6847034

table(svm_predround, testing$quality)

##
## svm_predround   3   4   5   6   7   8   9
##             4   2   3   0   0   0   0   0
##             5   6  34 324  88   1   0   0
##             6   1  12 117 467 138  25   1
##             7   0   0   1  40  84  18   1

(3+324+467+84)/(1363)

## [1] 0.6441673
```

After that, we decided to run some classification analysis, first by breaking the wine into good, bad, and average. Our grading scale here was anything less than 5 in quality was considered bad, anything above five but below seven was average, and seven and above good. We created a separate data set called wines for classification models, mainly because the original kept breaking when we would try it for linear models after.

From the breakdown, we saw that 3490 of the training wine was considered average (between 5 and 6), 163 was bad (3 and 4), and 894 was good (7, 8, 9). When we ran the model, we found out that the best KNN was at k=23 which gave a 79.4% accuracy. When we ran a Confusion Matrix against our test data, we got a 78.9% accuracy for prediction. Though one notable thing about the prediction was that the model struggled in predicting bad and good wine, but the accuracy was high for average wines.

```r
wines <- wine
wines$qualitynum = wines$quality
wines$quality[which(wines$quality %in% c(3,4))] = 'bad'
wines$quality[which(wines$quality %in% c(5, 6))] = 'average'
wines$quality[which(wines$quality %in% c(7,8,9))] = 'good'
table(wines$quality)
```

```
##
## average     bad    good
##    3490     163     894
```

```r
set.seed(1234)
train.set <- createDataPartition(wines$quality, p=0.7, list= FALSE)
exclude <- which(names(wines) %in% c('wine_type', 'qualitynum'))
train <- wines[train.set, -exclude]
test <- wines[-train.set, -exclude]

ctrl <- trainControl(method= "repeatedcv", repeats=10, classProbs=TRUE)

knn.mod <- train(quality ~., data = train, method = 'knn', preProcess =
c("center", "scale"), metric = "Accuracy", trControl=ctrl, tuneLength = 10)

knn.mod
```

```
## k-Nearest Neighbors
##
## 3184 samples
##   11 predictor
##    3 classes: 'average', 'bad', 'good'
##
## Pre-processing: centered (11), scaled (11)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 2864, 2866, 2867, 2866, 2866, 2865, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.7827357  0.3381662
##    7  0.7836764  0.3309469
##    9  0.7867555  0.3337386
##   11  0.7886358  0.3327494
##   13  0.7881026  0.3223203
##   15  0.7888220  0.3191825
##   17  0.7892308  0.3148263
##   19  0.7913962  0.3168436
##   21  0.7931857  0.3186447
##   23  0.7940954  0.3165399
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 23.
```

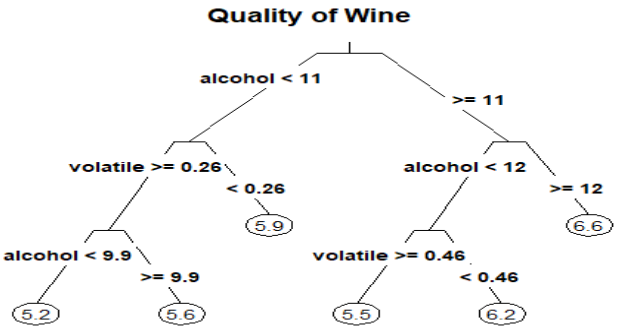
```r
test_pred <- predict(knn.mod, newdata = test)
confusionMatrix(table(test_pred, test$quality))
```

```
## Confusion Matrix and Statistics

## test_pred average bad good
##   average     987  47  179
```

```
##   bad           0   0    0
##   good         60   1   89
##
## Overall Statistics
##
##                Accuracy : 0.7894
##                  95% CI : (0.7668, 0.8108)
##     No Information Rate : 0.7682
##     P-Value [Acc > NIR] : 0.03263
##
##                   Kappa : 0.2856
##  Mcnemar's Test P-Value : < 2e-16
##
## Statistics by Class:
##
##                      Class: average Class: bad Class: good
## Sensitivity                  0.9427    0.00000      0.3321
## Specificity                  0.2848    1.00000      0.9443
## Pos Pred Value               0.8137        NaN      0.5933
## Neg Pred Value               0.6000    0.96478      0.8524
## Prevalence                   0.7682    0.03522      0.1966
## Detection Rate               0.7241    0.00000      0.0653
## Detection Prevalence         0.8899    0.00000      0.1101
## Balanced Accuracy            0.6138    0.50000      0.6382
```

Next we ran a tree model on the wine data to visually see which predictors impacted quality and in what way. No surprise at all, alcohol would have the biggest impact for quality, with branches at less than 11% and greater than or equal to 11%. Volatile acidity also had a big impact, which seem to also impact alcohol content and quality. Most wine predicted from the tree fell between five and seven in the grading scale.

```
set.seed(1234)

tree <- rpart(wine$quality ~ ., data = wine)
prp(tree, type=3, tweak=1, main="Quality of Wine", compress=TRUE )
```



Using the tree model, we wanted to run a prediction to see how the data could compare to our testing data from our train_data sample. In this, we saw that the wine could only predict quality at 5, 6, or 7, and gave an accuracy of 52% after rounding.

```
tree_model<-rpart(training$quality~., data=training)
tree_pred<-predict(tree_model, testing)
```

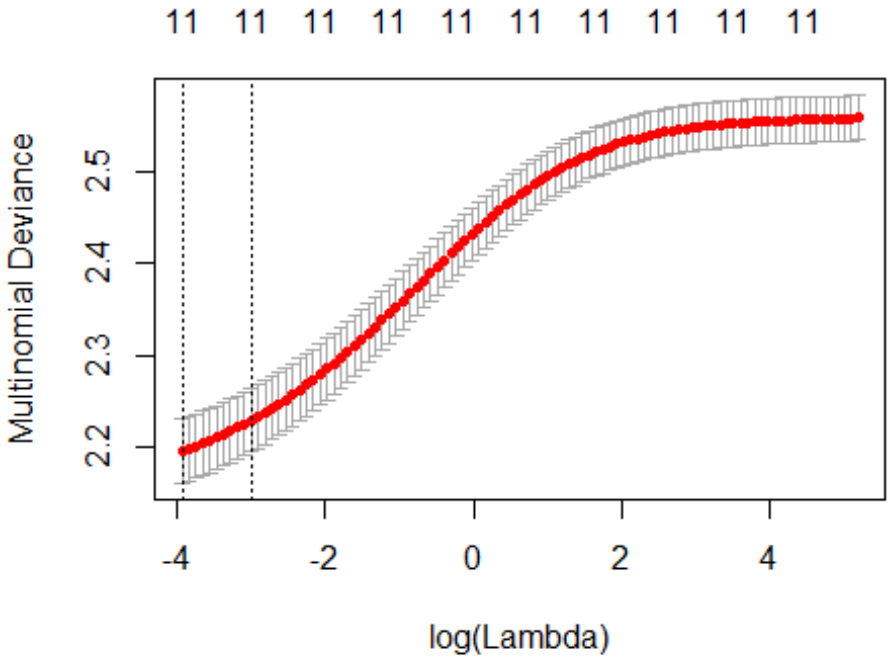
```
round_tree <- round(tree_pred)
table(round_tree, testing$quality)

##
## round_tree   3   4   5   6   7   8   9
##          5   3  16 225  87   3   0   0
##          6   5  31 210 424 153  23   1
##          7   1   2   7  84  67  20   1

(225+424+67)/(1363)

## [1] 0.5253118
```

Next, we ran a random forest with 500 trees and scaling, which predicted scores of five to eight, and an accuracy of approximately 66%.

```
set.seed(1234)
rf <- randomForest(quality ~ ., data=training, ntree = 500, scale=TRUE)
pred <- predict(rf, newdata = testing)
roundpred <- round(pred)
table(roundpred, testing$quality)

##
## roundpred   3   4   5   6   7   8   9
##         5   6  29 297  66   3   2   0
##         6   3  19 144 488 111  16   1
##         7   0   1   1  41 109  20   1
##         8   0   0   0   0   0   5   0

(297+488+109+5)/(1363)

## [1] 0.6595745
```

Lastly, we decided to run ridge regression analysis. Here we needed to create another partition because we were getting error message from the other two data. Our data was 70% training and 30% testing and we looked at the predictors 2 to 12, skipping wine_type as it was a classification. We ran a multinomial test and again the prediction only could account for wine quality of 5, 6, and 7. Our accuracy from the ridge regression analysis was 56.2%.

```
ridge_wine <- train_data
set.seed(1234)
ridge_size <- ceiling(nrow(ridge_wine)*.3)
Ind_test <- sample(c(1:nrow(ridge_wine)),size=ridge_size,replace=FALSE)
Ind_train <- setdiff(c(1:nrow(ridge_wine)),Ind_test)
ridge_train <- ridge_wine[Ind_train,]
ridge_test <- ridge_wine[Ind_test,]

ridge_CV <-
cv.glmnet(x=as.matrix(ridge_train[,c(2:12)]),y=ridge_train[,13],family='multino
mial',alpha=0)

plot(ridge_CV)
```




```
ridge_predict <-
as.numeric(predict.cv.glmnet(ridge_CV,as.matrix(ridge_test[,c(2:12)]),s='lambda
.min',type='class'))
table(ridge_predict,ridge_test$quality)
```

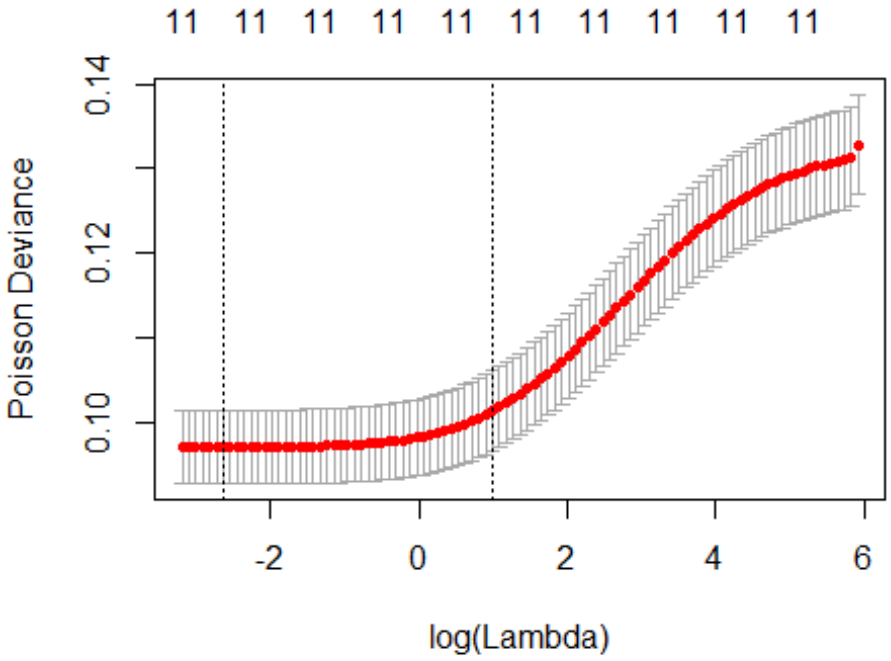
```
##
## ridge_predict   3   4   5   6   7   8
##             5   3  33 298 144  12   2
##             6   0  19 177 441 146  34
##             7   0   0   1  19  28   8
```

```
(298+441+28)/(1365)
```

```
## [1] 0.5619048
```

Next we ran a poisson family ridge regression, and got an accuracy of 55%.

```
poisson_ridge <-
cv.glmnet(x=as.matrix(ridge_train[,c(2:12)]),y=ridge_train[,13],family='poisson
',alpha=0)
plot(poisson_ridge)
```




```
poisson_ridge_pred <-
as.numeric(predict.cv.glmnet(poisson_ridge,as.matrix(ridge_test[,c(2:12)]),s='l
ambda.min',type='response'))
roundpoisson <- round(poisson_ridge_pred)
table(roundpoisson,ridge_test$quality)

##
## roundpoisson   3   4   5   6   7   8
##            4   0   0   2   0   0   0
##            5   3  28 249 106   6   0
##            6   0  24 221 464 142  30
##            7   0   0   4  34  38  14

(0+249+464+38)/(1365)

## [1] 0.5501832
```

Discussion:

From that, we had concluded our modeling and found out that linear regression performed the worst on the data, and that classification models performed better than regression models. The reason for this could be that classification models grouped the quality into three categories as opposed to seven, so the prediction data just needed to be within the range to be considered accurate as opposed to getting the exactly quality number.

For regression, our best model was random forest, which had an accuracy of approximately 66%. Random Forest did a very solid job at predicting average wine between the scores of five and six, but not at predicting wine in outside those numbers.

```
set.seed(1234)
rf <- randomForest(quality ~ ., data=training, ntree = 500, scale=TRUE)
pred <- predict(rf, newdata = testing)
```

```
roundpred <- round(pred)
table(roundpred, testing$quality)

##
## roundpred   3   4   5   6   7   8   9
##         5   6  29 297  66   3   2   0
##         6   3  19 144 488 111  16   1
##         7   0   1   1  41 109  20   1
##         8   0   0   0   0   0   5   0

(297+488+109+5)/(1363)

## [1] 0.6595745
```

If we ran classification analysis using KNN, we found that the best accuracy came with K = 23 and this gave a 79% accuracy of wine prediction. In this case the data was unable to predict bad wine at all, similar to the random forest for regression analysis, and it predicted more good wine as average than as good. Both models were successful at predicting "average" wine, not good or bad wine.

```
wines <- wine
wines$qualitynum = wines$quality
wines$quality[which(wines$quality %in% c(3,4))] = 'bad'
wines$quality[which(wines$quality %in% c(5, 6))] = 'average'
wines$quality[which(wines$quality %in% c(7,8,9))] = 'good'

set.seed(1234)
train.set <- createDataPartition(wines$quality, p=0.7, list= FALSE)
exclude <- which(names(wines) %in% c('wine_type', 'qualitynum'))
train <- wines[train.set, -exclude]
test <- wines[-train.set, -exclude]

ctrl <- trainControl(method= "repeatedcv", repeats=10, classProbs=TRUE)

knn.mod <- train(quality ~., data = train, method = 'knn', preProcess =
c("center", "scale"), metric = "Accuracy", trControl=ctrl, tuneLength = 10)

test_pred <- predict(knn.mod, newdata = test)
confusionMatrix(table(test_pred, test$quality))

## Confusion Matrix and Statistics
##
##
## test_pred average bad good
##   average     987  47  179
##   bad           0   0    0
##   good         60   1   89
##
## Overall Statistics
##
##                Accuracy : 0.7894
##                  95% CI : (0.7668, 0.8108)
##     No Information Rate : 0.7682
##     P-Value [Acc > NIR] : 0.03263
##
##                   Kappa : 0.2856
```

```
##  Mcnemar's Test P-Value : < 2e-16
##
## Statistics by Class:
##
##                      Class: average Class: bad Class: good
## Sensitivity                  0.9427    0.00000      0.3321
## Specificity                  0.2848    1.00000      0.9443
## Pos Pred Value               0.8137        NaN      0.5933
## Neg Pred Value               0.6000    0.96478      0.8524
## Prevalence                   0.7682    0.03522      0.1966
## Detection Rate               0.7241    0.00000      0.0653
## Detection Prevalence         0.8899    0.00000      0.1101
## Balanced Accuracy            0.6138    0.50000      0.6382
```

In the end, it should be noted that the models that performed best used clustering, which could be explained by looking at the training data to begin with, since most fell within the five and six range. From our initial prediction, we found the model supported by alcohol and volatile acidity had a big impact on quality, although residual sugar, fixed acidity, and especially citric acid weren't impactful

In the future, a few things could be done to improve our result. The first would be to break the wine into red and white, as the two wines are made differently, so studying them individually as opposed to combined could have given more accurate predictions. The models we used to predict good red wine may not give good white wine. Another improvement could be to use a different study variable instead of quality, one that was more quantifiable. Quality is very suggestive and changes from person to person; a quality 5 wine for one person could be a quality 6 wine for another person, or even a quality 4 for a third person. In the future if we looked at alcohol percentage for example as our y-variable, we may be able to better predict.

Lastly, we added the prediction table to the test data with unknown quality below, with both regression and classification. From our model, we predict that most of the unknown are average wine between five and six in quality, and few fall within the seven and eight range.

```
tests <- test_data
tests$id = NULL
quality <- predict(knn.mod, newdata = tests)
classification_tests <- cbind(test_data, quality)
table(classification_tests$quality)

##
## average     bad    good
##    1716       0     234

write.csv(classification_tests,file = file.choose(new = T))

pred <- predict(rf, newdata = tests)
quality <- round(pred)
regression_test <- cbind(test_data, quality)
table(regression_test$quality)

##
##    5    6    7    8
##  592 1099  255    4
```

```
write.csv(regression_test, file= file.choose(new = T))
```

Sources:

https://winefolly.com/review/wine-characteristics/
https://vincarta.com/blog/assessing-quality/
http://rstudio-pubs-static.s3.amazonaws.com/24803_abbae17a5e154b259f6f9225da6dade0.html
http://rpubs.com/beka/red-wine_data-analysis
https://rpubs.com/datascientiest/237405
https://www.kaggle.com/sagarnildass/red-wine-analysis-by-r
https://rstudio-pubs-static.s3.amazonaws.com/33876_1d7794d9a86647ca90c4f182df93f0e8.html
https://www.r-bloggers.com/predicting-wine-quality-using-random-forests/
https://www.researchgate.net/publication/324870033_Comparing_Linear_Ridge_and_Lasso_Regressions
http://rstudio-pubs-static.s3.amazonaws.com/299637_2ba434e6967240c8b8da4511cb42318f.html
https://www.kaggle.com/ssudeep/red-wine-quality-prediction-using-ridge-regression
https://rpubs.com/jeknov/redwine
http://rstudio-pubs-static.s3.amazonaws.com/175762_83cf2d7b322c4c63bf9ba2487b79e77e.html
https://rpubs.com/Daria/57835
https://www.kaggle.com/umutozdemir/comparison-of-different-regression-models
https://www.kaggle.com/vishalyo990/prediction-of-quality-of-wine
https://www.kaggle.com/grosvenpaul/beginners-guide-to-eda-and-random-forest-using-r
https://www.kaggle.com/meepbobeep/intro-to-regression-and-classification-in-r
https://www.kaggle.com/aleixdorca/keras-caret-with-the-wine-dataset
https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009/kernels