

# Capstone Project: Wine Quality

*Isaiah Lemons*

*12/31/2019*

## Introduction

This project is part of the HarvardX:PH125.9x Data Science Capstone project. There are two datasets used that provide multiple physiochemical tests based on red and white wine samples that came from northern Portugal. The goal of this project is to develop machine learning algorithms based on all the physiochemical test results provided, in attempt to predict if a certain wine will be of high quality.

The datasets used can be found at the link below. (<https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/>).

## Data Wrangling

```
# Install any neccessary libraries
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

The following code was used to import the data and split it into test and training sets for later models.

```
# Import the Red and White datasets
url_red <- "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv"
url_white <- "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv"

red_data <- read.csv(url_red, sep=';')
white_data <- read.csv(url_white, sep=';')

# Merge the two separate red and white wine datasets into one dataset
wine <- rbind(red_data, white_data)

# Adding a column to classify an excellent wine quality
table(wine$quality)
```

```
##
##      3      4      5      6      7      8      9
##    30   216  2138  2836  1079   193      5
```

```
wine <- wine %>% mutate(Excellent = ifelse(quality > 6, 1, 0))
wine$Excellent <- as.factor(wine$Excellent)

# remove files no longer necessary
rm(url_red, url_white, red_data, white_data)

# Splitting data into test and train sets 80/20 split
set.seed(42)
#set.seed(1, sample.kind="Rounding") #if using R 3.5 or later
```

```
test_index <- createDataPartition(wine$Excellent, times = 1, p = 0.2, list = FALSE)
train_set <- wine[-test_index,]
test_set <- wine[test_index,]
```

## Exploratory Analysis

Once the data is available initial analysis and research the dataset can begin. Using the code below we can see that this dataset is in tidy format, and it contains 6497 rows and 13 columns.

```
# data is in tidy format
wine %>% as.tibble()
```

```
## # A tibble: 6,497 x 13
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
##   <dbl>          <dbl>          <dbl>          <dbl>    <dbl>
## 1         7.4         0.7           0           1.9     0.076
## 2         7.8         0.88          0           2.6     0.098
## 3         7.8         0.76          0.04         2.3     0.092
## 4        11.2         0.28          0.56         1.9     0.075
## 5         7.4         0.7           0           1.9     0.076
## 6         7.4         0.66          0           1.8     0.075
## 7         7.9         0.6           0.06         1.6     0.069
## 8         7.3         0.65          0           1.2     0.065
## 9         7.8         0.580          0.02         2       0.073
## 10        7.5         0.5           0.36         6.1     0.071
## # ... with 6,487 more rows, and 8 more variables: free.sulfur.dioxide <dbl>,
## #   total.sulfur.dioxide <dbl>, density <dbl>, pH <dbl>, sulphates <dbl>,
## #   alcohol <dbl>, quality <int>, Excellent <fct>
```

```
# checking the structure of the data
str(wine)
```

```
## 'data.frame':   6497 obs. of  13 variables:
## $ fixed.acidity      : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
## $ volatile.acidity   : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
## $ citric.acid        : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
## $ residual.sugar     : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
## $ chlorides          : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
## $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
## $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
## $ density            : num  0.998 0.997 0.997 0.998 0.998 ...
## $ pH                 : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
## $ sulphates          : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
## $ alcohol            : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
## $ quality            : int   5 5 5 6 5 5 5 7 7 5 ...
## $ Excellent          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 2 1 ...
```

```
# checking basic summary statistics
summary(wine)
```

```
## fixed.acidity    volatile.acidity  citric.acid      residual.sugar
```

```
## Min. : 3.800 Min. :0.0800 Min. :0.0000 Min. : 0.600
## 1st Qu.: 6.400 1st Qu.:0.2300 1st Qu.:0.2500 1st Qu.: 1.800
## Median : 7.000 Median :0.2900 Median :0.3100 Median : 3.000
## Mean : 7.215 Mean :0.3397 Mean :0.3186 Mean : 5.443
## 3rd Qu.: 7.700 3rd Qu.:0.4000 3rd Qu.:0.3900 3rd Qu.: 8.100
## Max. :15.900 Max. :1.5800 Max. :1.6600 Max. :65.800
## chlorides free.sulfur.dioxide total.sulfur.dioxide density
## Min. :0.00900 Min. : 1.00 Min. : 6.0 Min. :0.9871
## 1st Qu.:0.03800 1st Qu.: 17.00 1st Qu.: 77.0 1st Qu.:0.9923
## Median :0.04700 Median : 29.00 Median :118.0 Median :0.9949
## Mean :0.05603 Mean : 30.53 Mean :115.7 Mean :0.9947
## 3rd Qu.:0.06500 3rd Qu.: 41.00 3rd Qu.:156.0 3rd Qu.:0.9970
## Max. :0.61100 Max. :289.00 Max. :440.0 Max. :1.0390
## pH sulphates alcohol quality Excellent
## Min. :2.720 Min. :0.2200 Min. : 8.00 Min. :3.000 0:5220
## 1st Qu.:3.110 1st Qu.:0.4300 1st Qu.: 9.50 1st Qu.:5.000 1:1277
## Median :3.210 Median :0.5100 Median :10.30 Median :6.000
## Mean :3.219 Mean :0.5313 Mean :10.49 Mean :5.818
## 3rd Qu.:3.320 3rd Qu.:0.6000 3rd Qu.:11.30 3rd Qu.:6.000
## Max. :4.010 Max. :2.0000 Max. :14.90 Max. :9.000
```

```
# Number of rows and columns
nrow(wine)
```

```
## [1] 6497
```

```
ncol(wine)
```

```
## [1] 13
```

```
# Check for missing values
any(is.na(wine))
```

```
## [1] FALSE
```

Installing additional libraries which may be useful for analysis and modeling

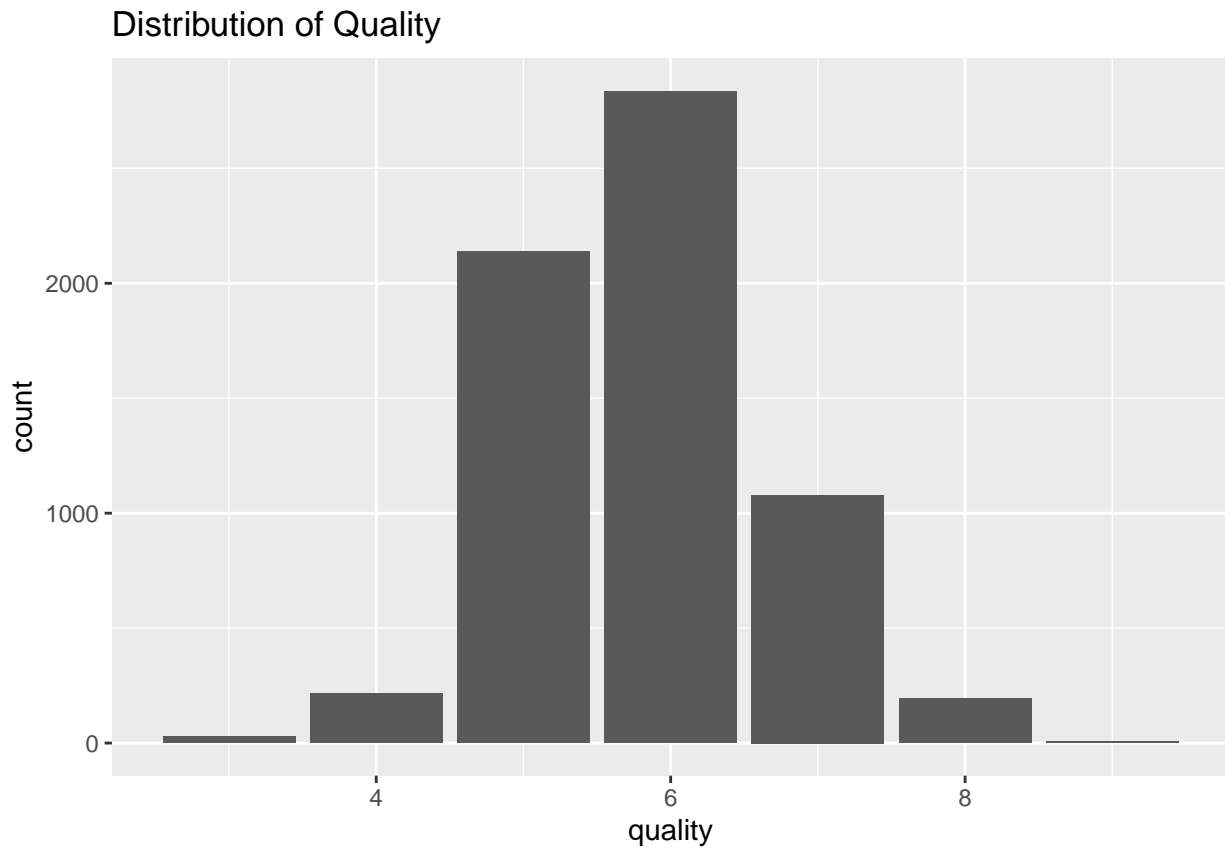
```
if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
if(!require(reshape2)) install.packages("reshape2", repos = "http://cran.us.r-project.org")
if(!require(lubridate)) install.packages("lubridate", repos = "http://cran.us.r-project.org")
if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(rpart)) install.packages("rpart", repos = "http://cran.us.r-project.org")
```

The first observation to check is the overall distribution of wines based on their quality.

```
# Overall Average Quality
mean(wine$quality)
```

```
## [1] 5.818378
```

```
# Distribution in Quality
wine %>%
  ggplot(aes(quality)) +
  geom_bar() +
  ggtitle("Distribution of Quality")
```



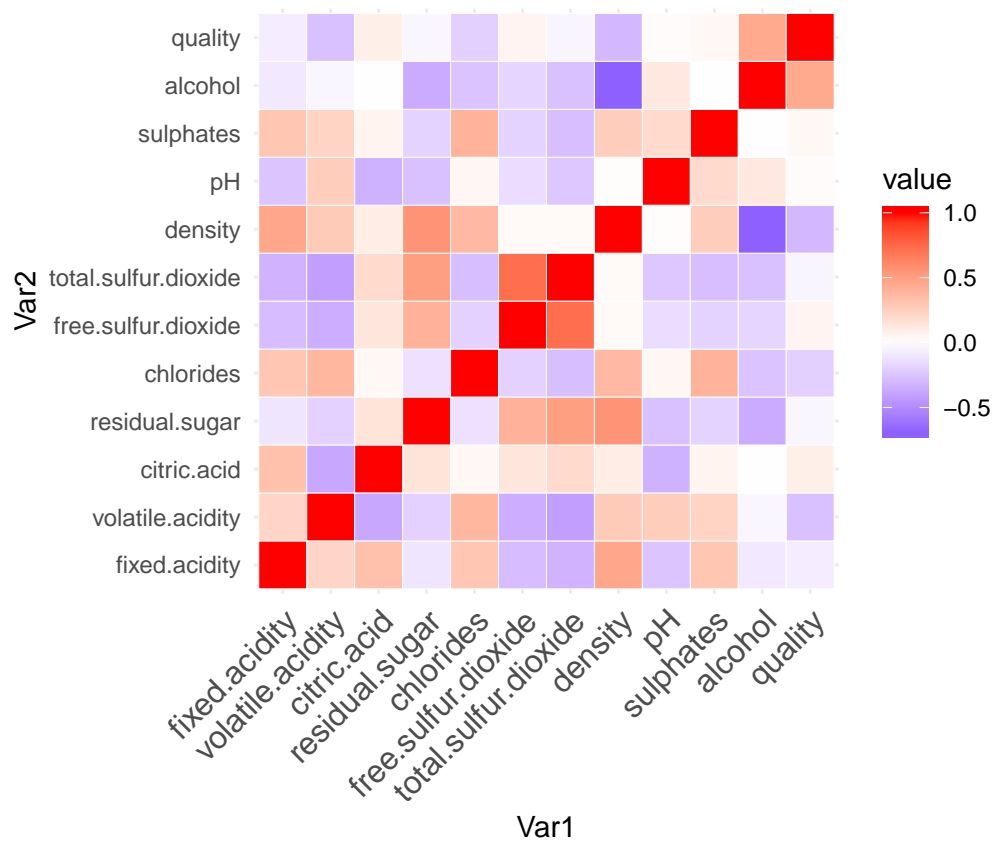
```
# Percentage of Excellent wines
mean(wine$Excellent == 1)
```

```
## [1] 0.1965523
```

Based on the following code creating a heatmap can allow for pin-pointing a few attributes that have higher correlations than others. These variables may play a bigger part in predictions later on, so it's good to take a further look. (alcohol, total.sulfur.dioxide, free.sulfur.dioxide, residual.sugar, and density)

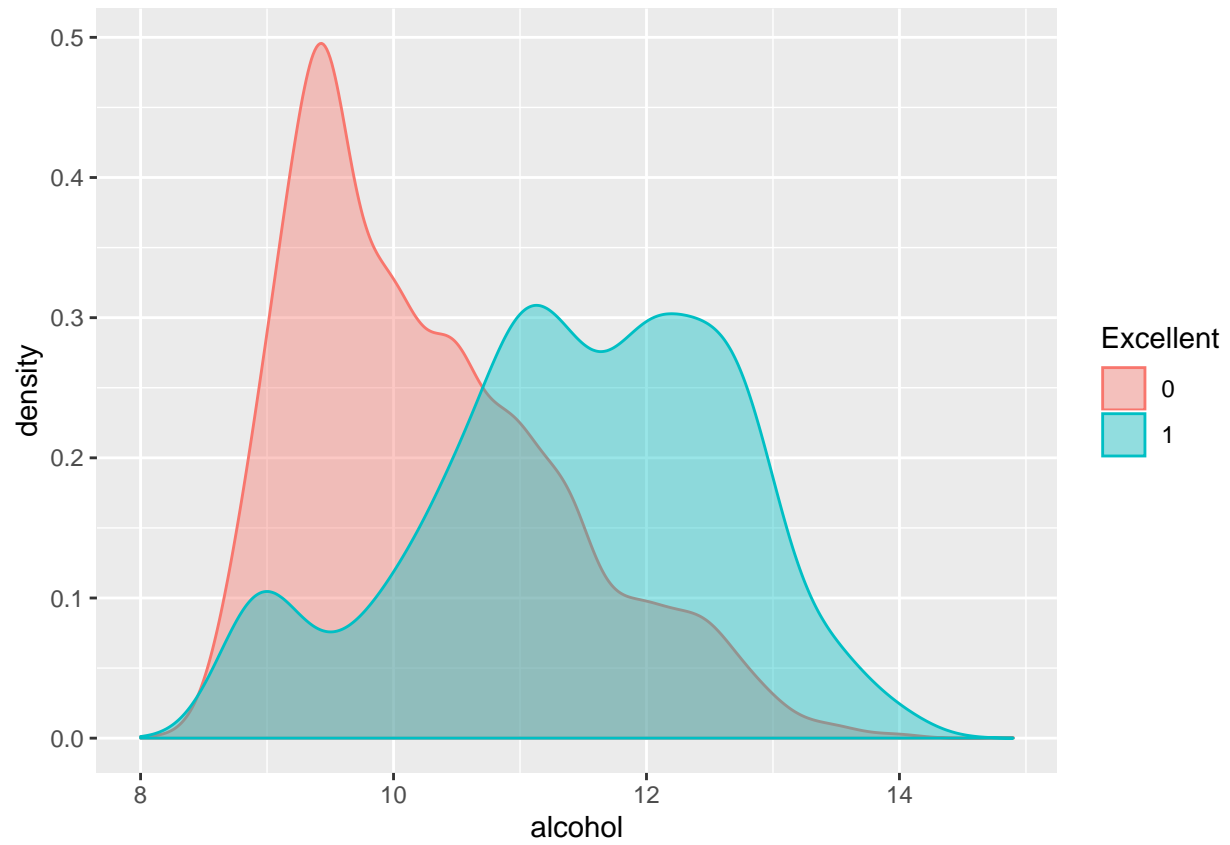
```
train.cor <- cor(subset(wine, select=-c(Excellent)))

ggplot(melt(train.cor), aes(Var1, Var2, fill=value)) +
  geom_tile(color = "white") + #color white is for border
  scale_fill_gradient2(low="blue", high="red", mid="white") +
  theme_minimal() + # minimal theme
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hjust = 1)) +
  coord_fixed()
```

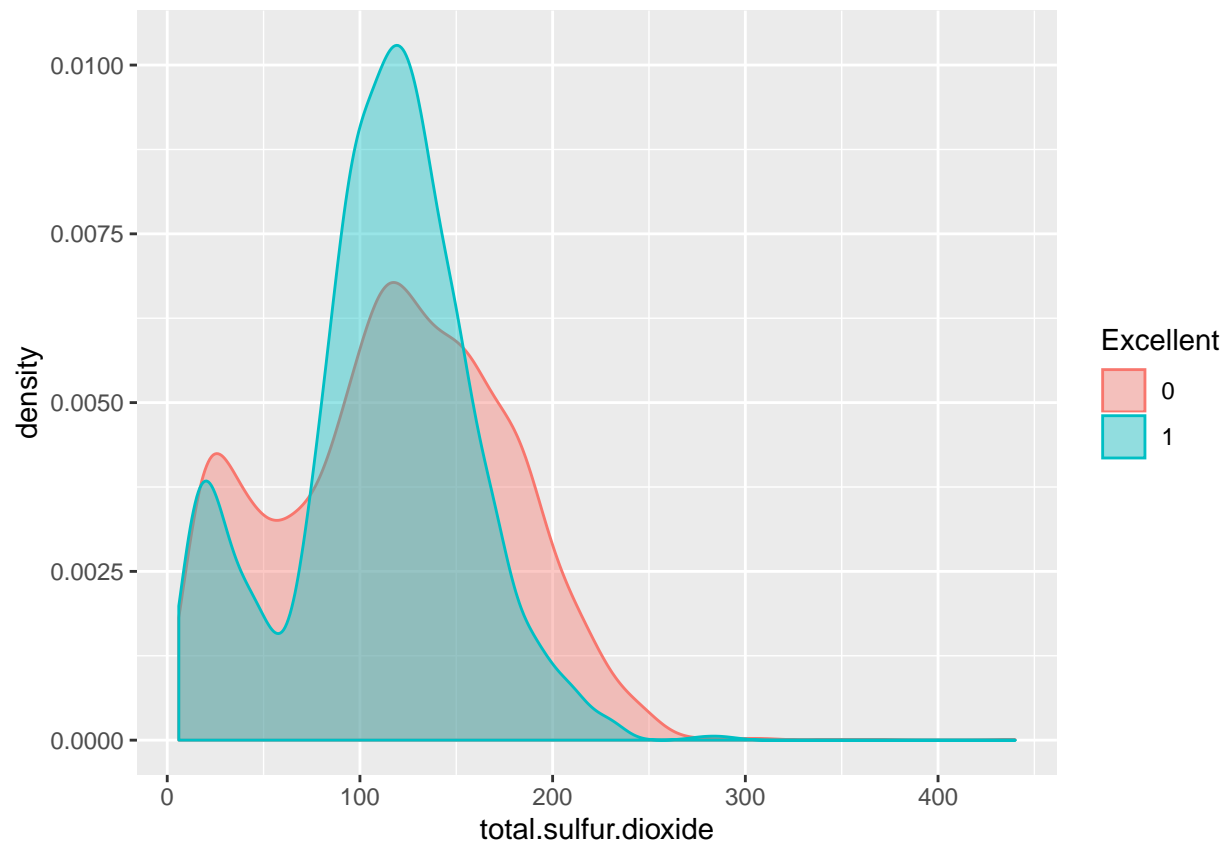


The below density plots check each of the physichemical tests to give a better understanding of their distributions. The flag created during data wrangling portion which specifies if a wine was considered excellent or not can also be added to give further insights.

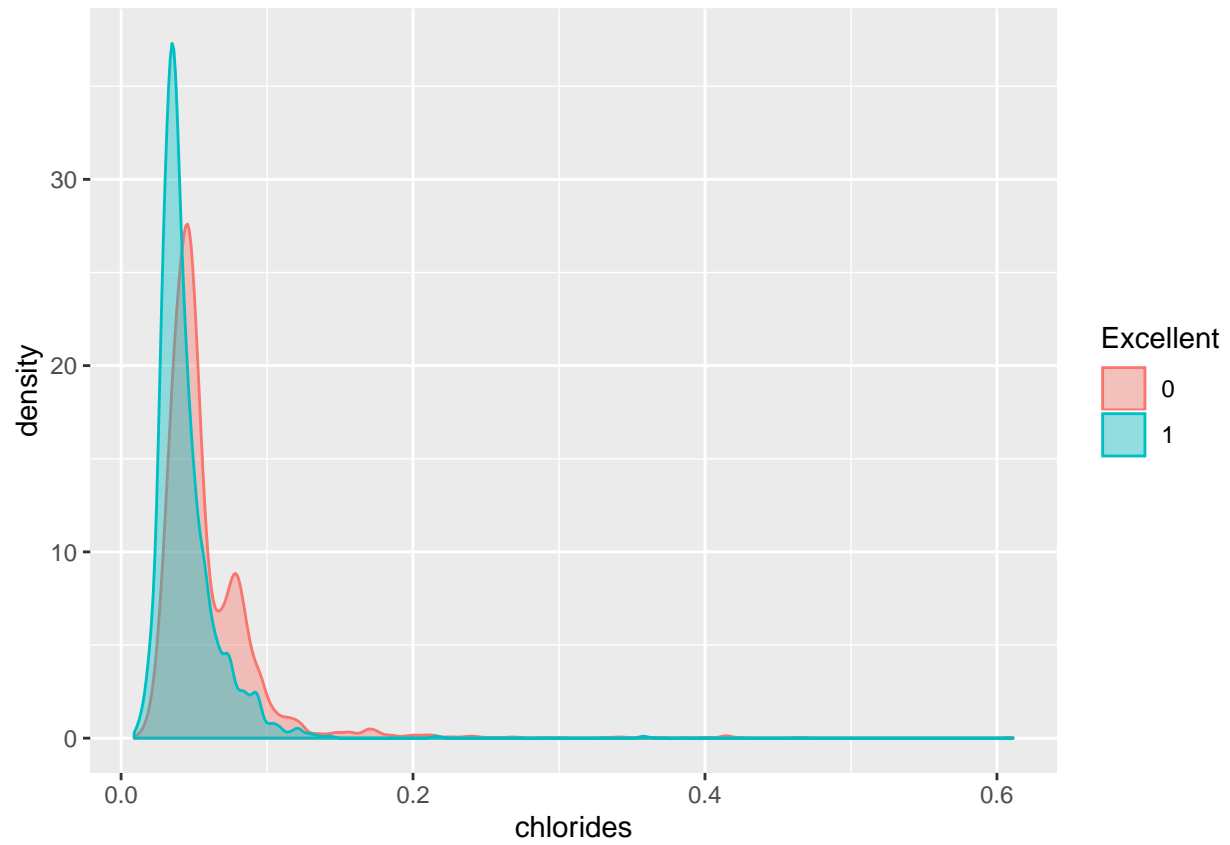
```
ggplot(wine, aes(alcohol, color=Excellent, fill=Excellent)) +  
  geom_density(alpha = 0.4)
```



```
ggplot(wine, aes(total.sulfur.dioxide, color=Excellent, fill=Excellent)) +  
  geom_density(alpha = 0.4)
```

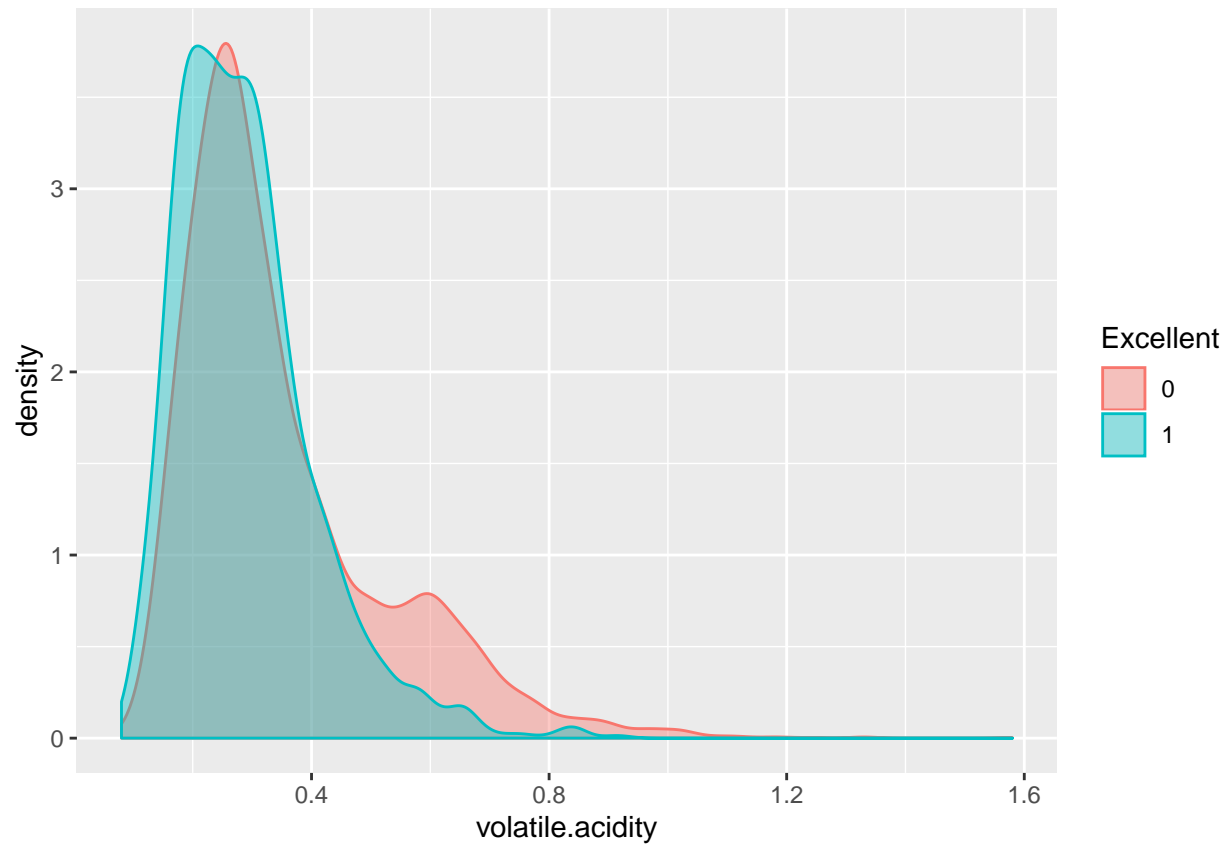


```
ggplot(wine, aes(chlorides, color=Excellent, fill=Excellent)) +  
  geom_density(alpha = 0.4)
```

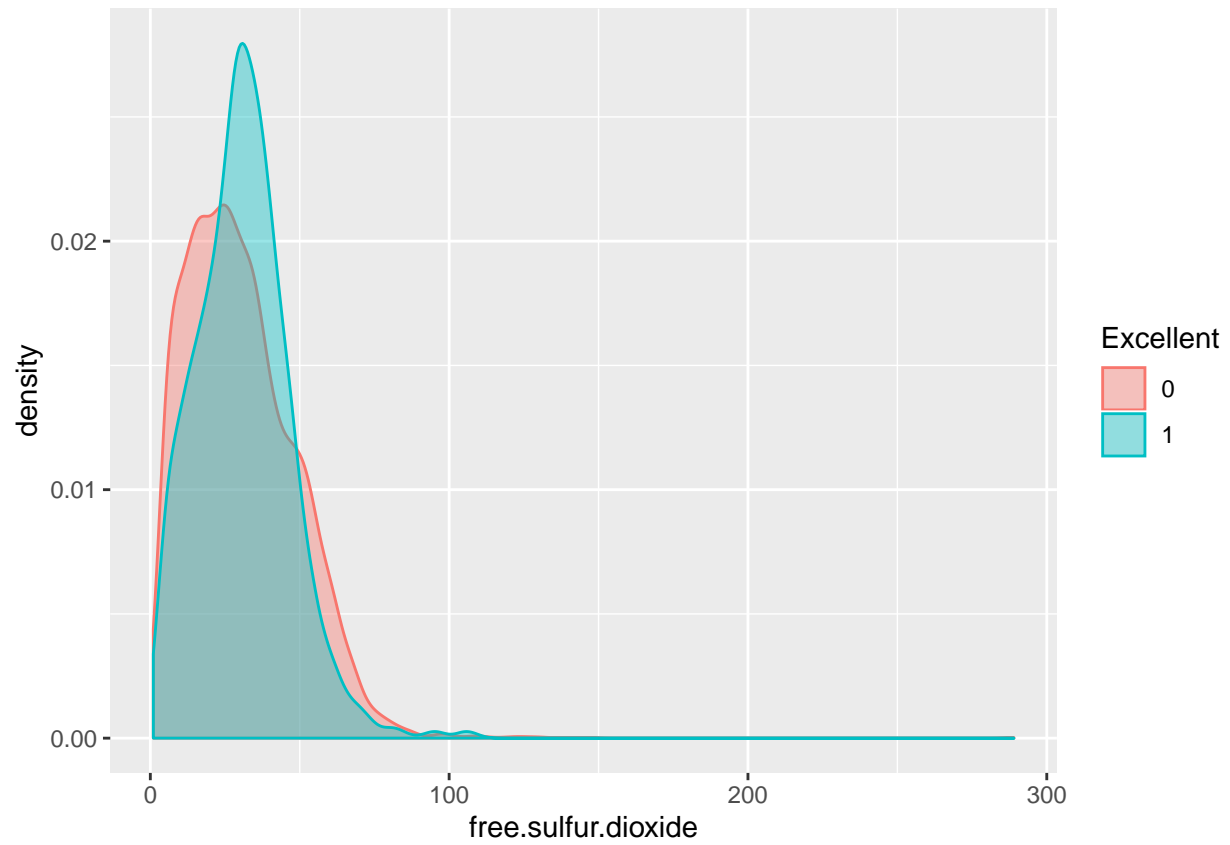


```
ggplot(wine, aes(volatile.acidity, color=Excellent, fill=Excellent)) +  
  geom_density(alpha = 0.4)
```

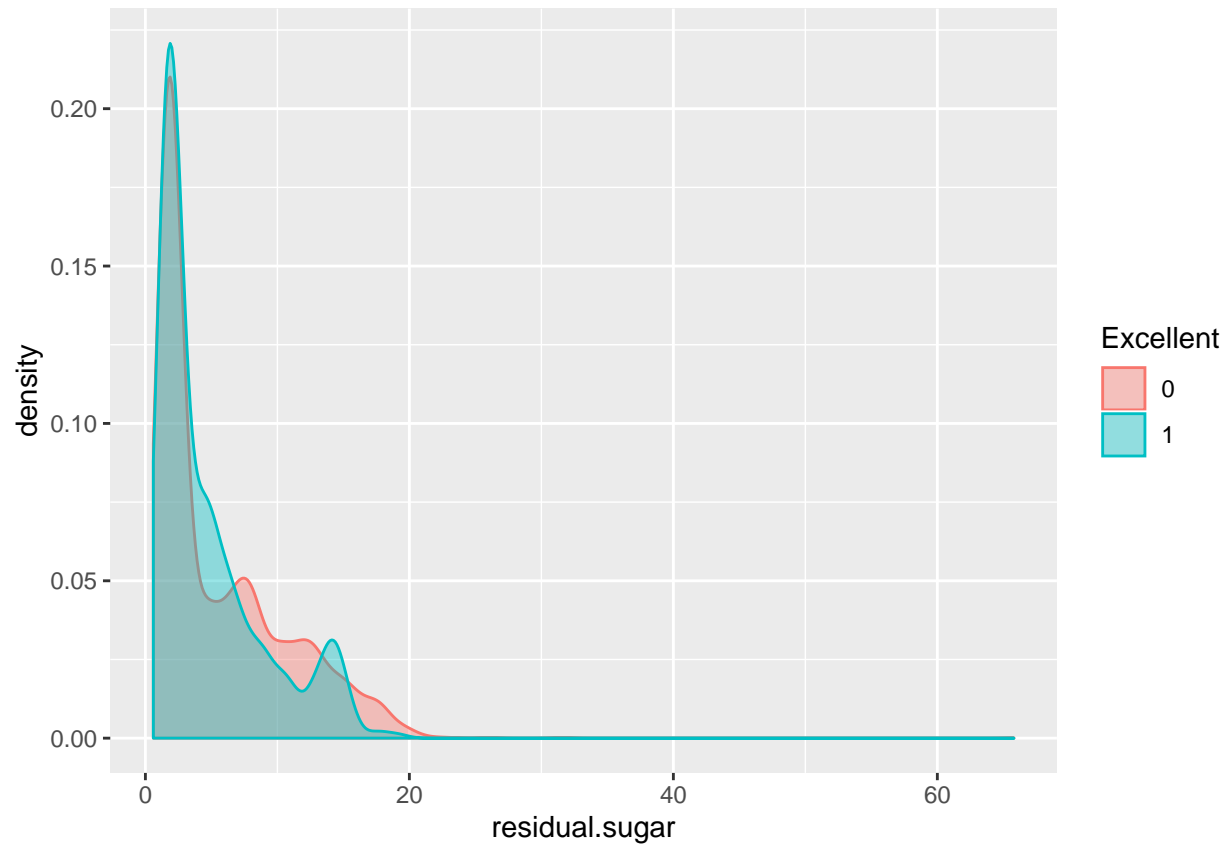




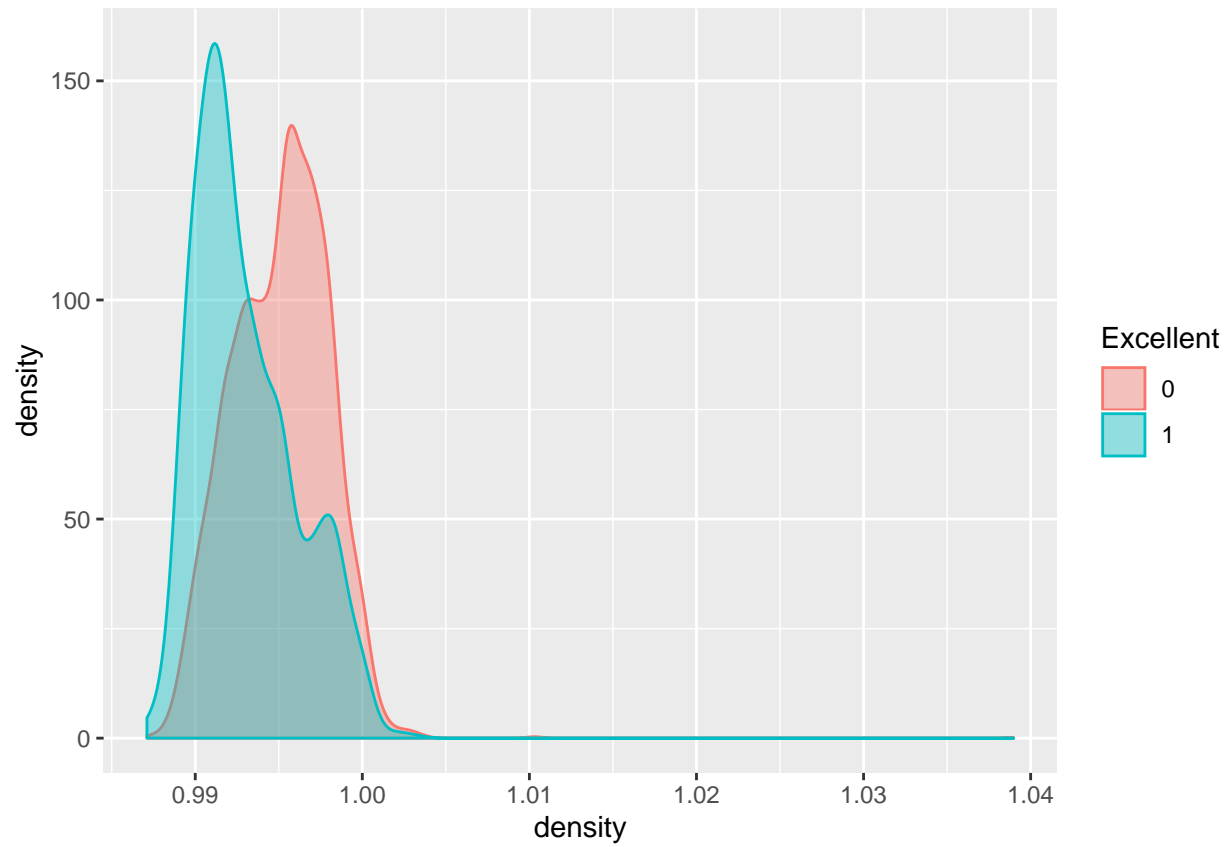
```
ggplot(wine, aes(free.sulfur.dioxide, color=Excellent, fill=Excellent)) +  
  geom_density(alpha = 0.4)
```



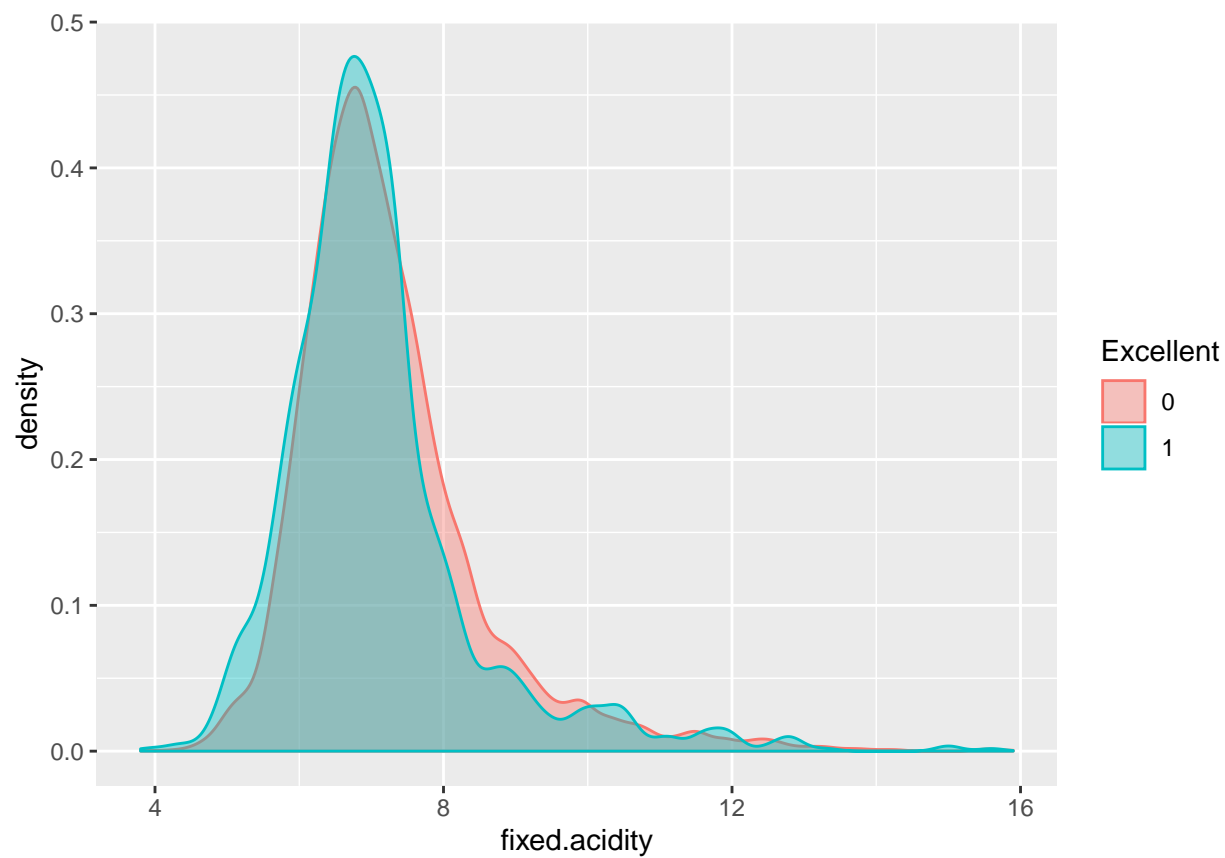
```
ggplot(wine, aes(residual.sugar, color=Excellent, fill=Excellent)) +  
  geom_density(alpha = 0.4)
```



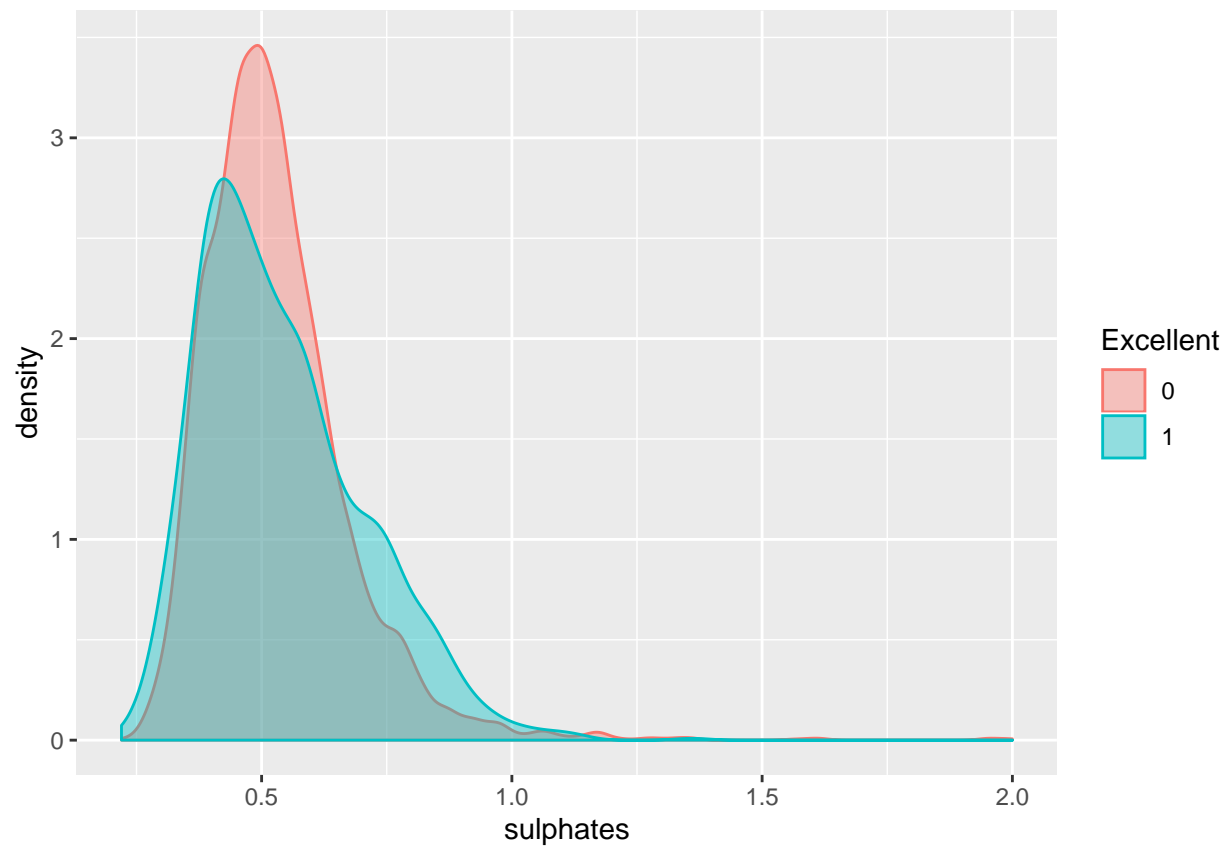
```
ggplot(wine, aes(density, color=Excellent, fill=Excellent)) +  
  geom_density(alpha = 0.4)
```



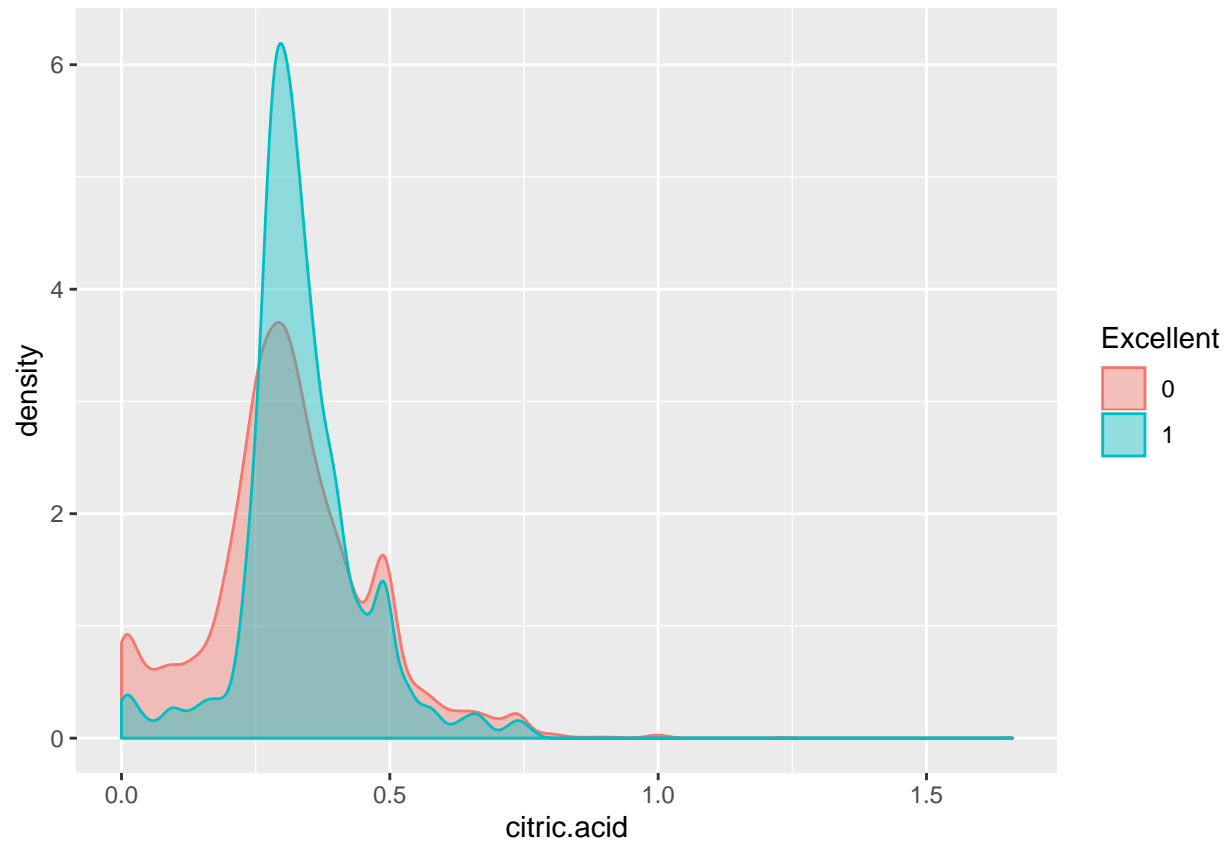
```
ggplot(wine, aes(fixed.acidity, color=Excellent, fill=Excellent)) +  
  geom_density(alpha = 0.4)
```



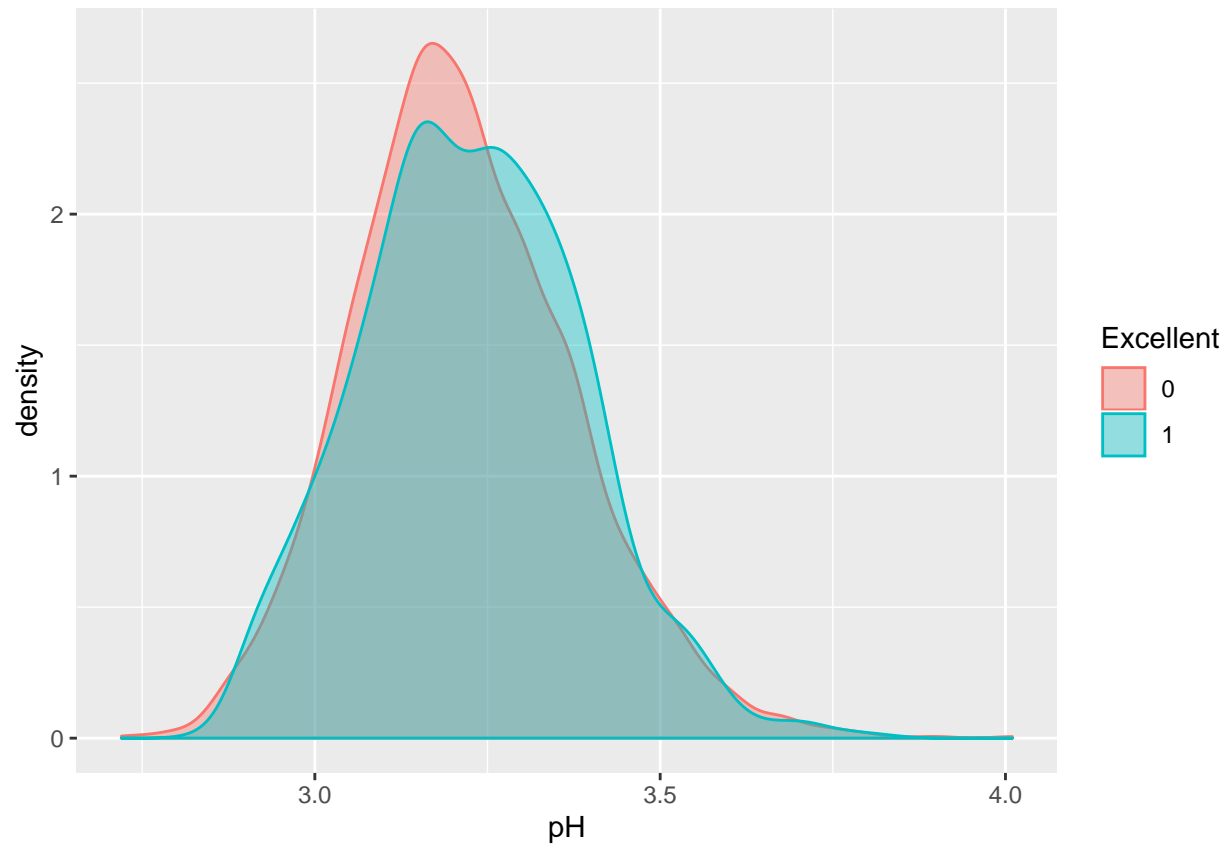
```
ggplot(wine, aes(sulphates, color=Excellent, fill=Excellent)) +  
  geom_density(alpha = 0.4)
```



```
ggplot(wine, aes(citric.acid, color=Excellent, fill=Excellent)) +  
  geom_density(alpha = 0.4)
```



```
ggplot(wine, aes(pH, color=Excellent, fill=Excellent)) +  
  geom_density(alpha = 0.4)
```

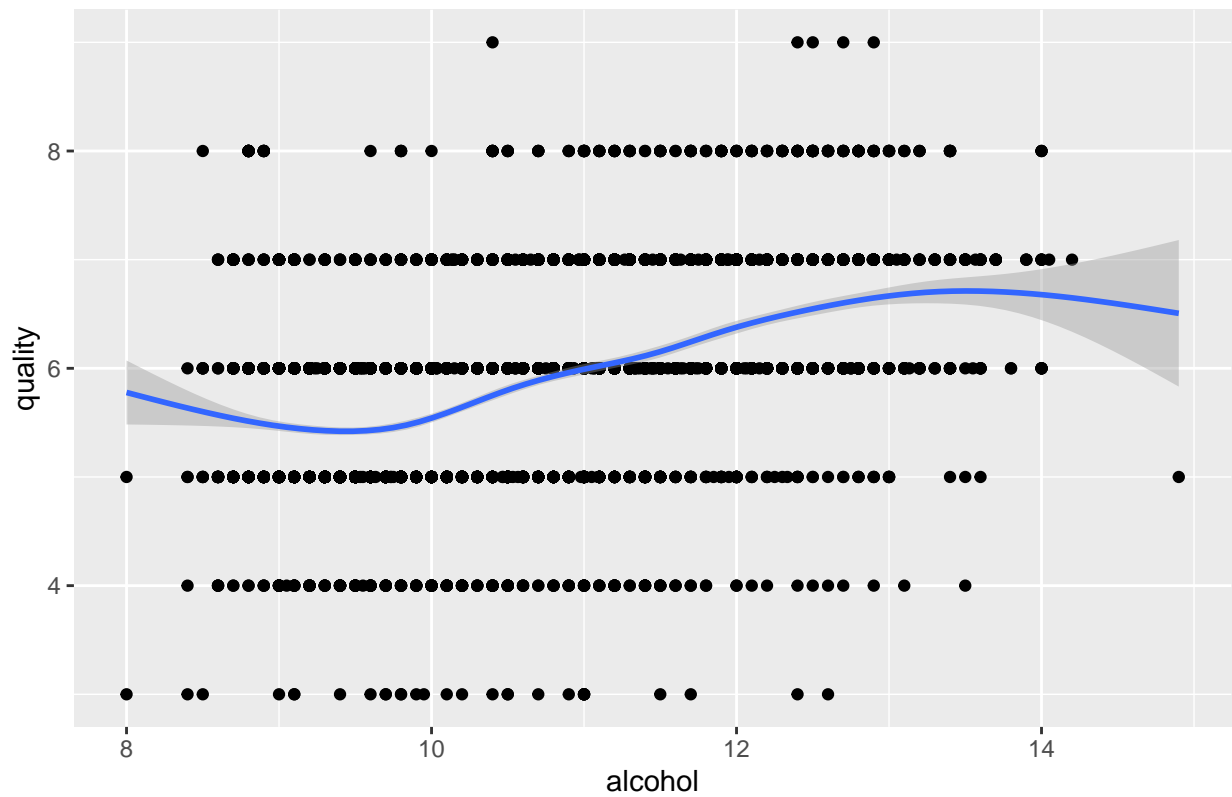


Taking a look at a couple scatterplots may show how certain attributes directly correlate with the quality. As shown in the plots below, when the alcohol content increases the quality generally increases as well. Density however, the quality tends to decrease as density increases.

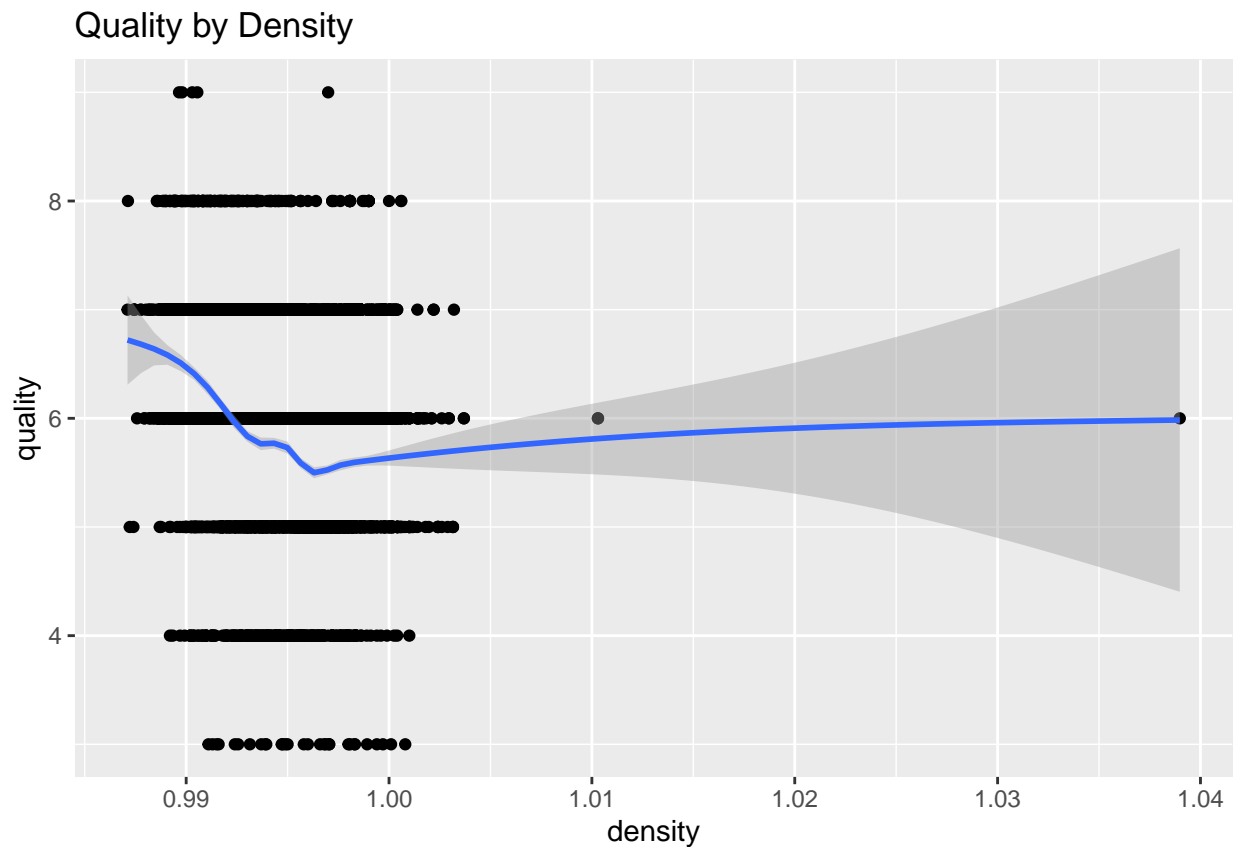
```
ggplot(wine, aes(alcohol, quality)) +  
  geom_point() +  
  geom_smooth() +  
  ggtitle("Quality by Alcohol")
```



Quality by Alcohol

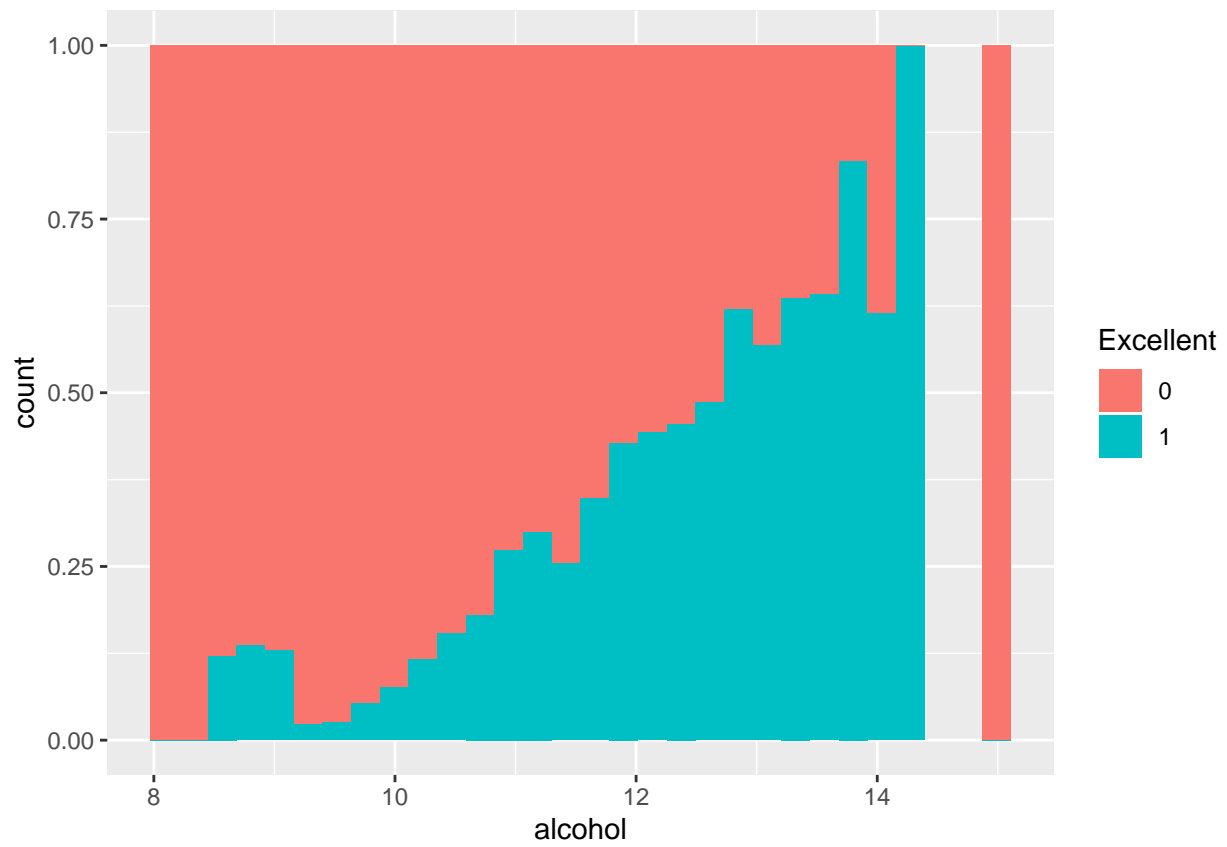


```
ggplot(wine, aes(density, quality)) +  
  geom_point() +  
  geom_smooth() +  
  ggtitle("Quality by Density")
```

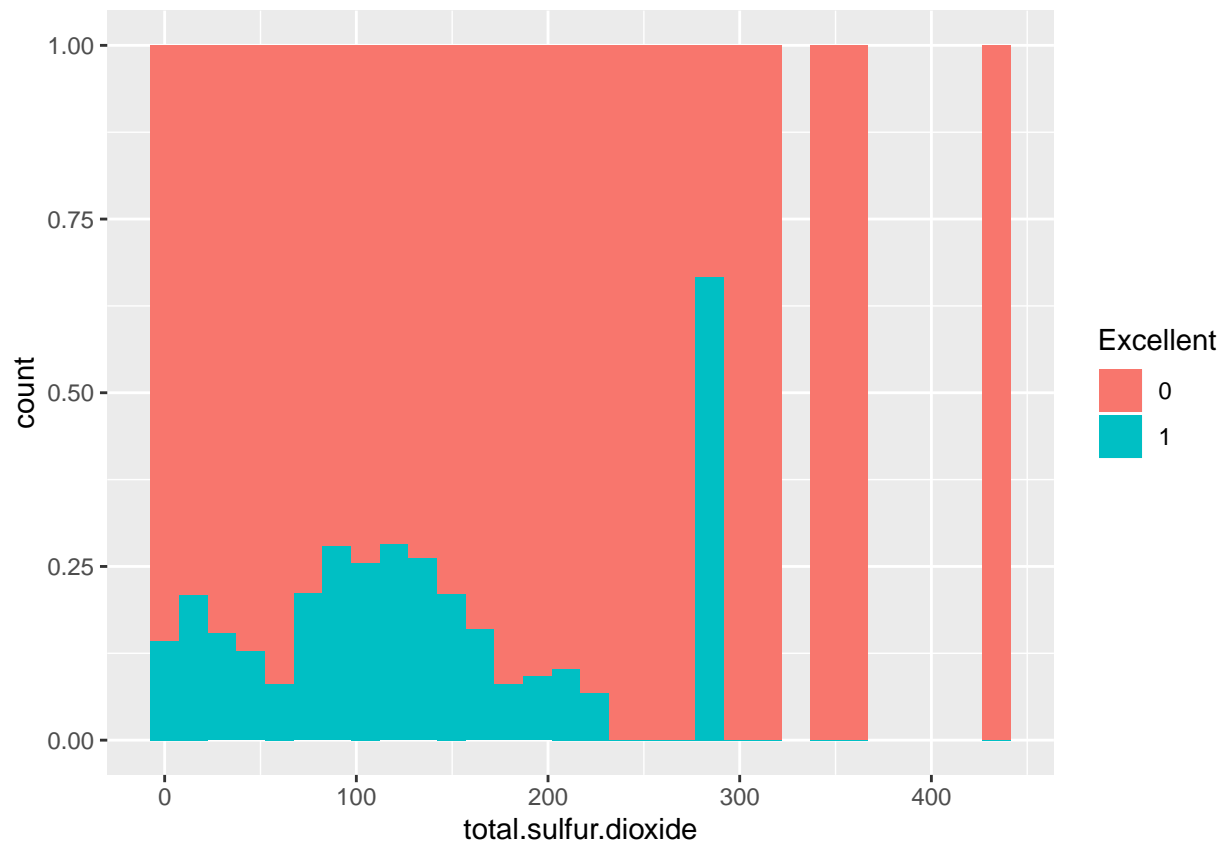


Stacked bar charts to easily show the percentage of excellent wines overall, based on particular variables. The images below continue to show similar results as previously mentioned, where higher quality wines tend to have higher alcohol content and lower densities.

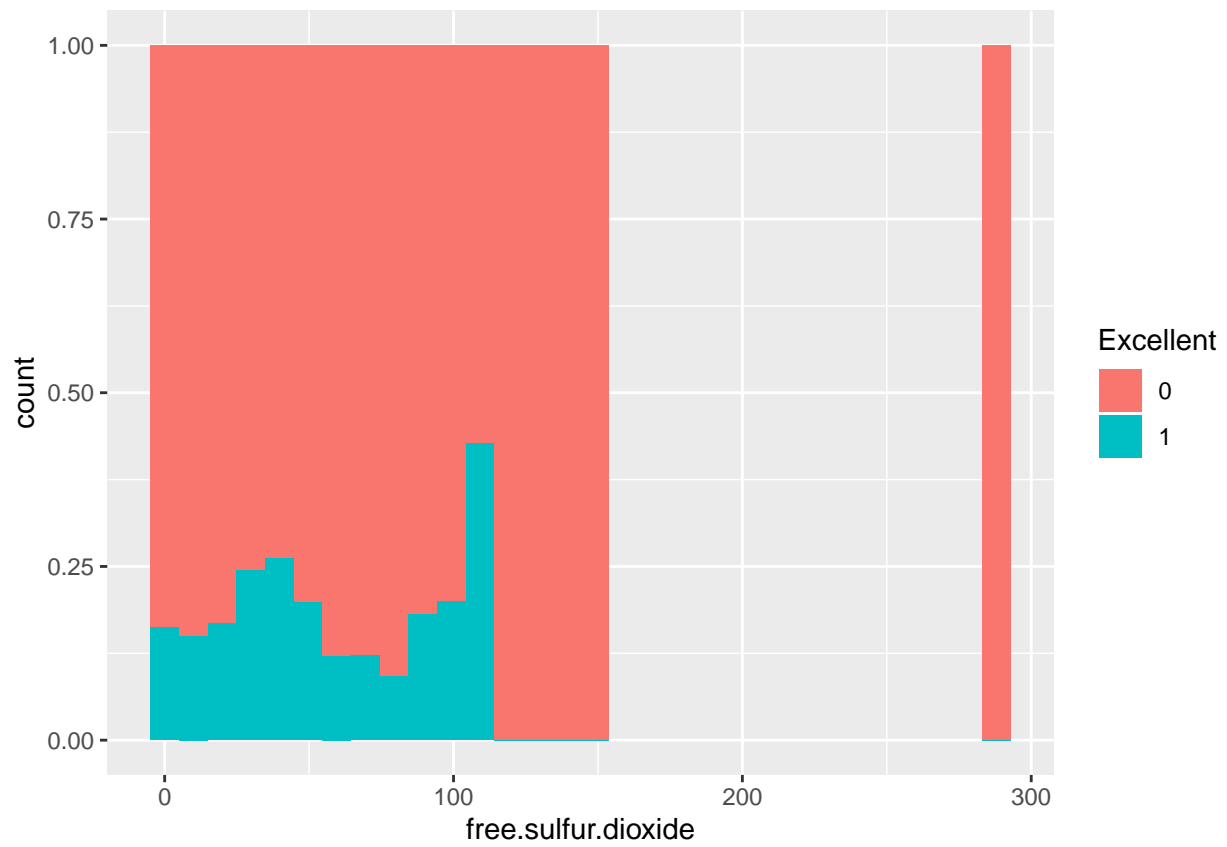
```
ggplot(wine, aes(alcohol, fill=Excellent)) +  
  geom_histogram(bins=30, position="fill")
```



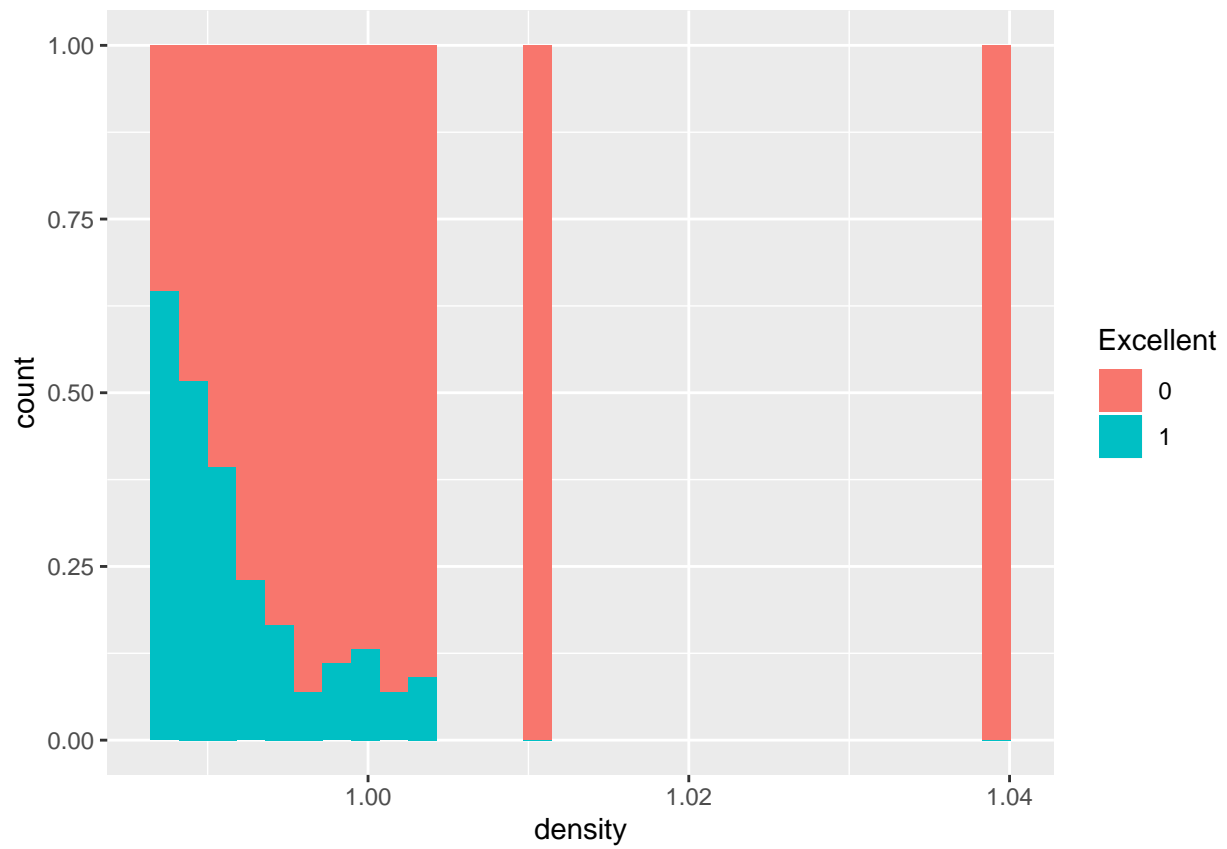
```
ggplot(wine, aes(total.sulfur.dioxide, fill=Excellent)) +  
  geom_histogram(bins=30, position="fill")
```



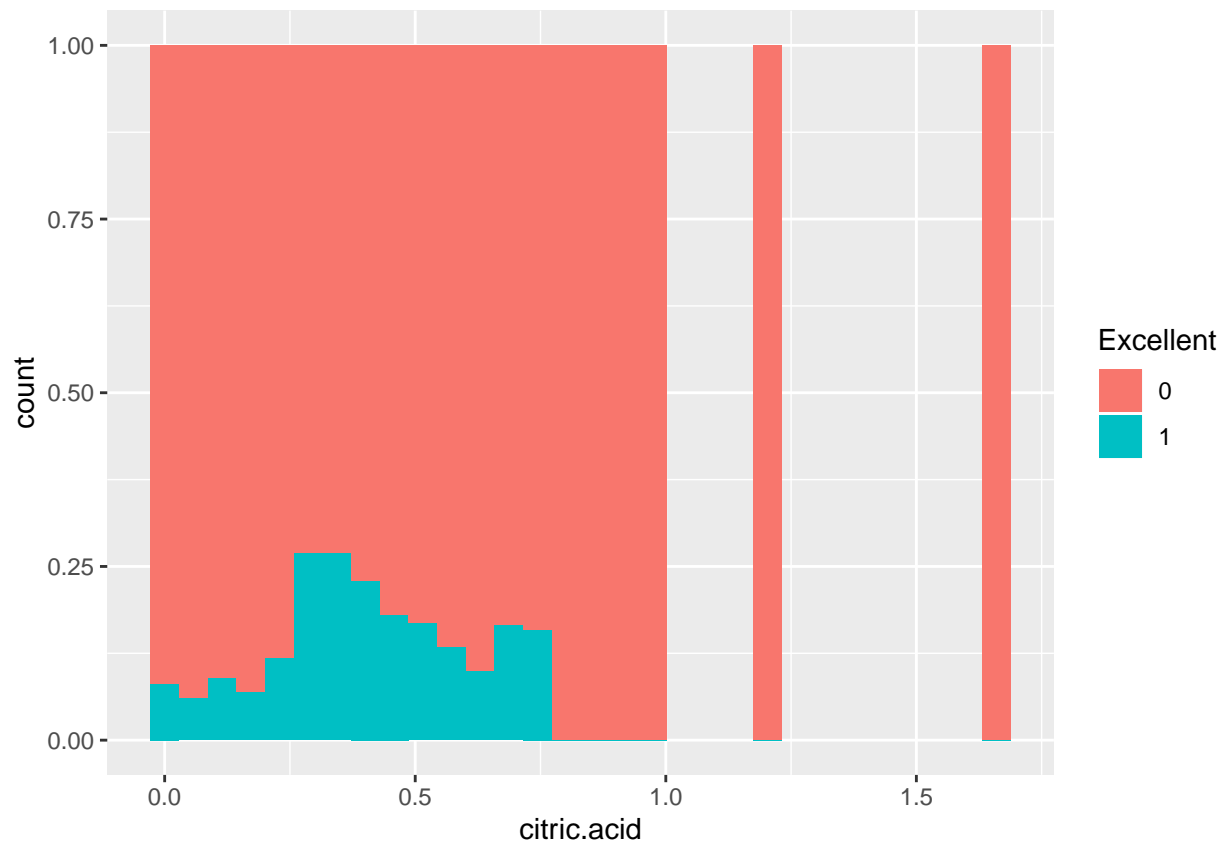
```
ggplot(wine, aes(free.sulfur.dioxide, fill=Excellent)) +  
  geom_histogram(bins=30, position="fill")
```



```
ggplot(wine, aes(density, fill=Excellent)) +  
  geom_histogram(bins=30, position="fill")
```



```
ggplot(wine, aes(citric.acid, fill=Excellent)) +  
  geom_histogram(bins=30, position="fill")
```



## Modeling

Each of the models below are using all the physiochemical tests variables available to predict if a wine will be considered excellent. The quality column is not included as a predictor of Excellent since it was directly used to create the classification flags.

```
# Logistic Regression Model
set.seed(1)
train_glm <- train(Excellent ~ .-quality, method = "glm", data = train_set)
glm_pred <- predict(train_glm, test_set, type = "raw")
confusionMatrix(glm_pred, test_set$Excellent)$overall[["Accuracy"]]
```

```
## [1] 0.8323077
```

```
# LDA Model
set.seed(1)
train_lda <- train(Excellent ~ .-quality, method = "lda", data = train_set)
lda_pred <- predict(train_lda, test_set)
confusionMatrix(lda_pred, test_set$Excellent)$overall[["Accuracy"]]
```

```
## [1] 0.8323077
```

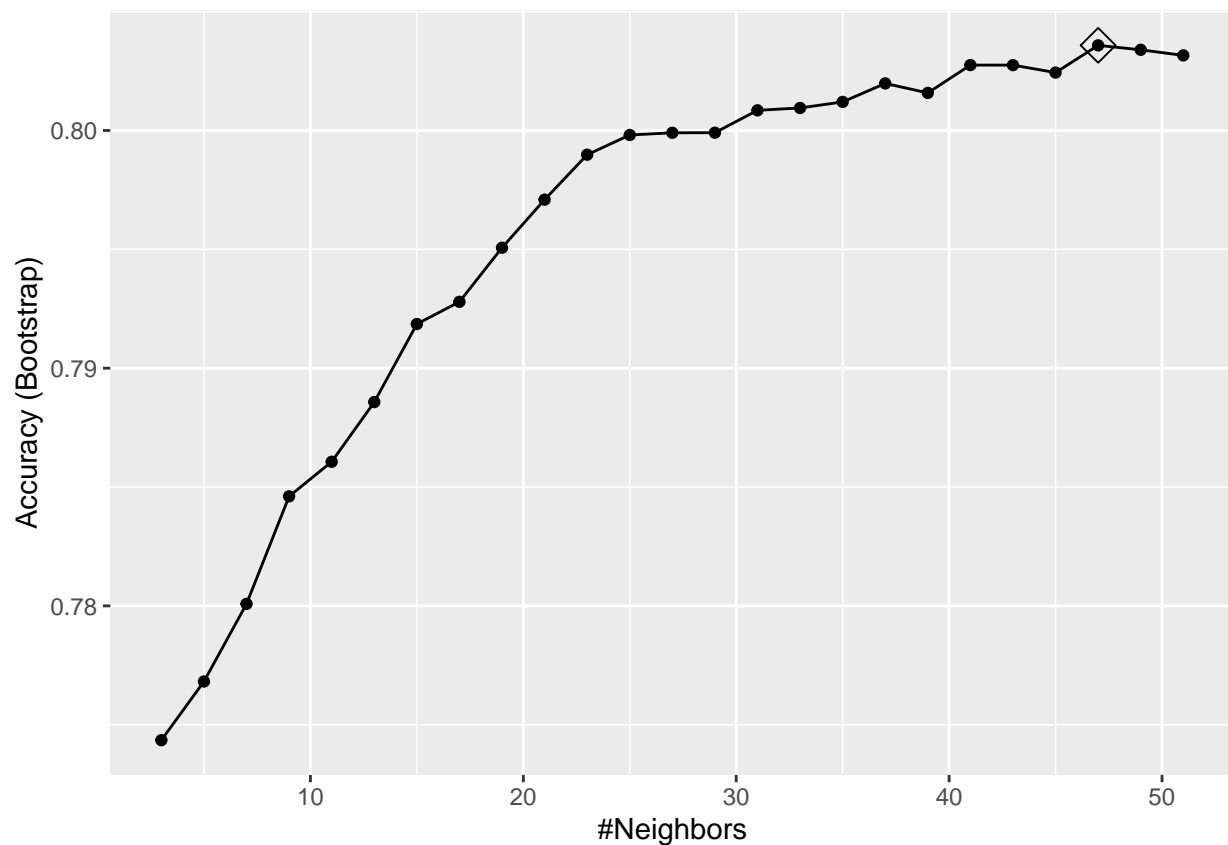
```
# QDA Model
set.seed(1)
train_lda <- train(Excellent ~ .-quality, method = "qda", data = train_set)
qda_pred <- predict(train_lda, test_set)
mean(lda_pred == test_set$Excellent)
```

```
## [1] 0.8323077
```

```
# Loess Model
set.seed(1)
train_loess <- train(Excellent ~ .-quality,
                     data = train_set,
                     method = "gamLoess")
loess_pred <- predict(train_loess, test_set)
mean(loess_pred == test_set$Excellent)
```

```
## [1] 0.8376923
```

```
# K Nearest Neighbors Model
set.seed(1)
train_knn <- train(Excellent ~ .-quality,
                  method = "knn",
                  data = train_set,
                  tuneGrid = data.frame(k = seq(3,51,2)))
ggplot(train_knn, highlight = TRUE)
```





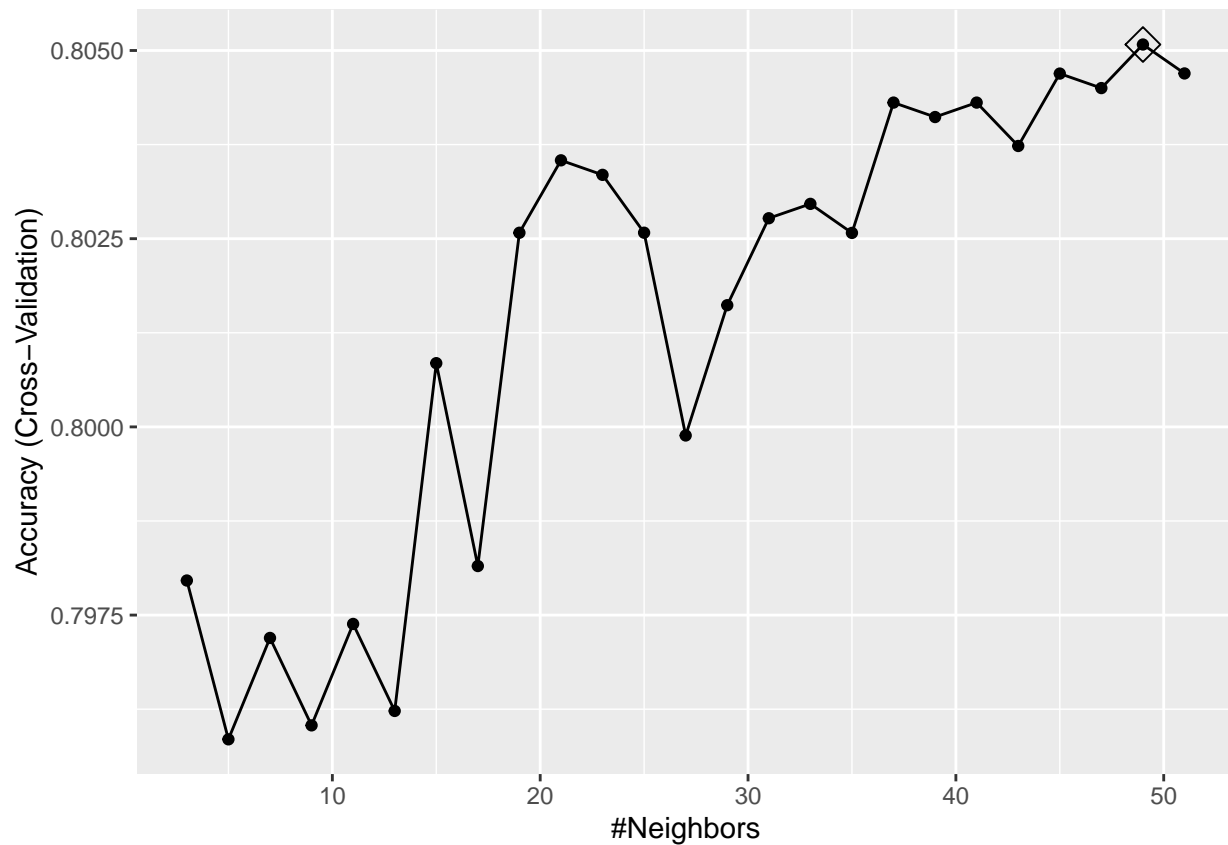
```
train_knn$bestTune
```

```
##      k  
## 23 47
```

```
knn_pred <- predict(train_knn, test_set, type = "raw")  
confusionMatrix(knn_pred, test_set$Excellent)$overall["Accuracy"]
```

```
## Accuracy  
## 0.8030769
```

```
# Cross Validation Model  
set.seed(1)  
train_knn_cross <- train(Excellent ~ .-quality,  
  method = "knn",  
  data = train_set,  
  tuneGrid = data.frame(k = seq(3,51,2)),  
  trControl = trainControl(method = "cv", number = 10, p = .9))  
ggplot(train_knn_cross, highlight = TRUE)
```



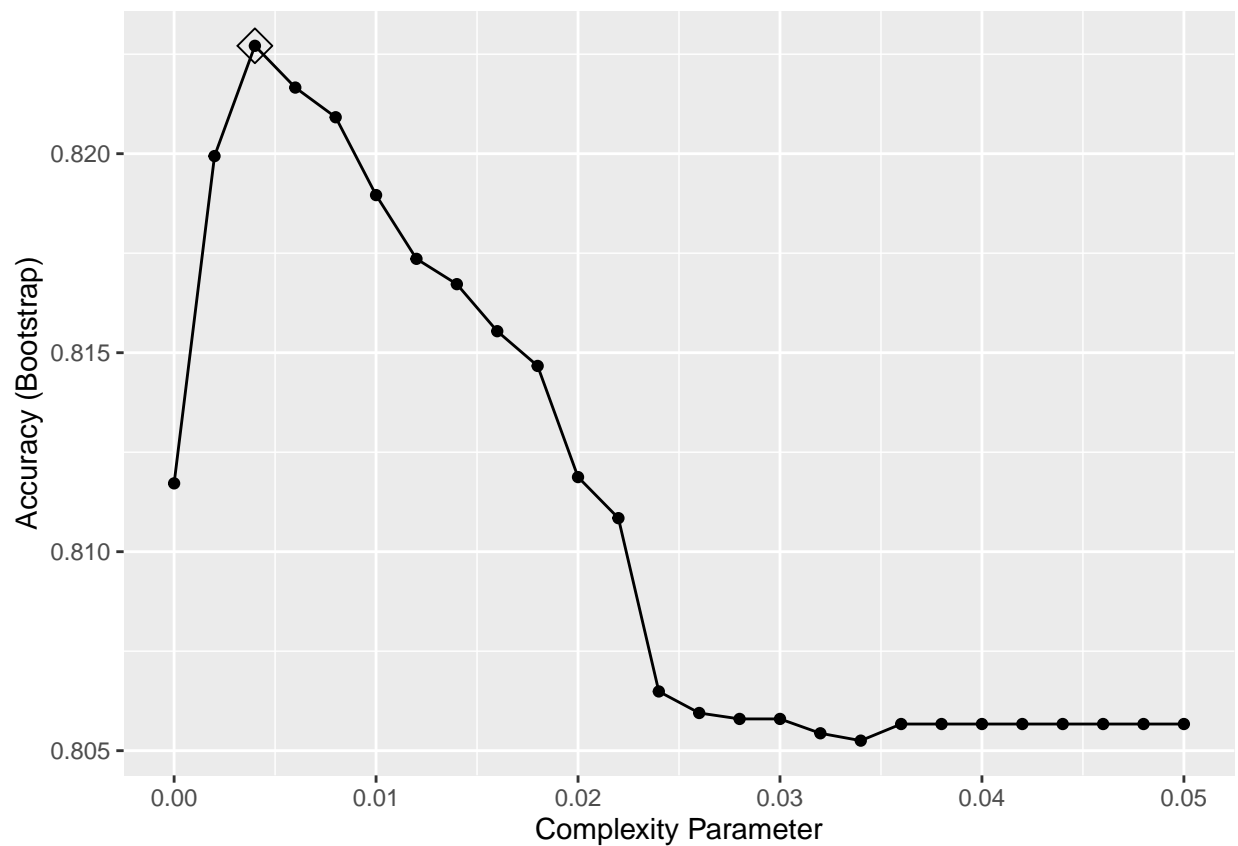
```
train_knn_cross$bestTune
```

```
##      k  
## 24 49
```

```
knn_cross_pred <- predict(train_knn_cross, test_set, type = "raw")
confusionMatrix(knn_cross_pred, test_set$Excellent)$overall["Accuracy"]
```

```
## Accuracy
## 0.8038462
```

```
# Rpart
set.seed(1)
train_tree <- train(Excellent ~ .-quality,
                    method = "rpart",
                    data = train_set,
                    tuneGrid = data.frame(cp = seq(0, 0.05, 0.002)))
ggplot(train_tree, highlight = TRUE)
```



```
rpart_pred <- predict(train_tree, test_set, type = "raw")
confusionMatrix(rpart_pred, test_set$Excellent)$overall["Accuracy"]
```

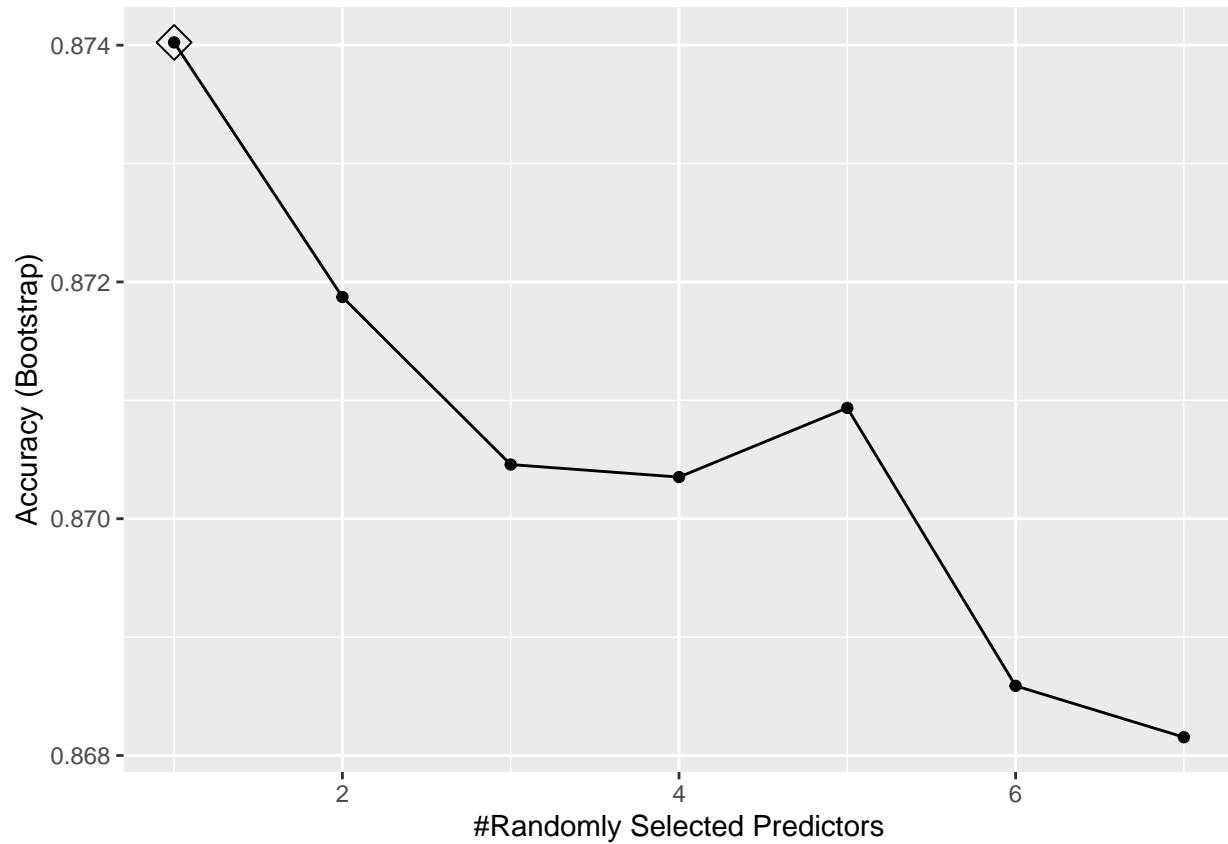
```
## Accuracy
## 0.8346154
```

```
# Random Forest Model
set.seed(1)
train_rf <- train(Excellent ~ .-quality,
```

```

method = "rf",
data = train_set,
tuneGrid = data.frame(mtry = seq(1:7)),
ntree = 100)
ggplot(train_rf, highlight = TRUE)

```



```

rf_pred <- predict(train_rf, test_set, type = "raw")
confusionMatrix(rf_pred, test_set$Excellent)$overall["Accuracy"]

```

```

## Accuracy
## 0.9053846

```

```

#importance of variables
varImp(train_rf)

```

```

## rf variable importance
##
## Overall
## alcohol      100.000
## density      76.208
## chlorides    39.376
## volatile.acidity 19.975
## total.sulfur.dioxide 17.175
## sulphates    14.729

```

```
## residual.sugar      13.334
## free.sulfur.dioxide 12.734
## citric.acid         11.799
## pH                  9.679
## fixed.acidity       0.000
```

## Results

```
models <- c("Logistic Regression", "LDA", "QDA", "Loess",
            "K nearest neighbors", "Cross Validation", "Rpart", "Random forest")
accuracy <- c(mean(glm_pred == test_set$Excellent),
              mean(lda_pred == test_set$Excellent),
              mean(qda_pred == test_set$Excellent),
              mean(loess_pred == test_set$Excellent),
              mean(knn_pred == test_set$Excellent),
              mean(knn_cross_pred == test_set$Excellent),
              mean(rpart_pred == test_set$Excellent),
              mean(rf_pred == test_set$Excellent))
Model_Results <- data.frame(Model = models, Accuracy = accuracy)
Model_Results
```

```
##           Model Accuracy
## 1 Logistic Regression 0.8323077
## 2           LDA 0.8323077
## 3           QDA 0.7792308
## 4           Loess 0.8376923
## 5 K nearest neighbors 0.8030769
## 6   Cross Validation 0.8038462
## 7           Rpart 0.8346154
## 8   Random forest 0.9053846
```

When comparing the results from all the different models performance it is clear that the highest performing model was the Random Forest which had a 90.5% accuracy. This is large improvement from the lowest performing model which was QDA at 77.9%.

## Conclusion

Through our analysis we were able to interpret the visualizations and determine a few variables that play a higher role in predicting the outcome. These variables were later confirmed with the random forest model which showed both alcohol and density as having the highest importance for predictions. Since this data didn't contain every aspect of wine data, things like brand name, age, sell price could all potentially improve these models and make it easier to classify an excellent wine from a mediocre.