# HarvardX: Wine Quality Prediction Capstone Project

Adriana Xavier

7/24/2021

## Introduction

This document corresponds to the Choose Your Own Project for HarvardX PH125.9x course. The goal of this project is to apply machine learning techniques on a publicly available dataset to solve the problem of your choice.

In this report, I'm analysing the quality of red wines and predicting if the wine is good or bad.

### Dataset description

The data set is available publicly on Kaggles's website, the data was provided by "Vinho Verde, a Portugal based wine company. The data inputs include objective tests (e.g. PH values) and the output is based on sensory data (median of at least 3 evaluations made by wine experts). Each expert graded the wine quality between 0 (very bad) and 10 (very good).

Input variables (based on physicochemical tests):

1. fixed acidity
2. volatile acidity
3. citric acid
4. residual sugar
5. chlorides
6. free sulfur dioxide
7. total sulfur dioxide
8. density
9. pH
10. sulphates
11. alcohol

Output variable (based on sensory data): 12. quality (score between 0 and 10)

## Methods

I have given each wine a badge classifying it as Good or Bad based on the quality variable. I will use two methods to predict it, linear regression and random forest. Confusion matrix will be used to describe the performance of the classification model.

## Preparing the data

The data is available on Kaggle: https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009

The same data set can be downloaded on my GitHub. Kaggle requires to log in to download the data, for security reasons I didn't want to add my login and password in the code so I uploaded the CSV file to GitHub and the code used in this report downloads it from GitHub.

After importing the dataset in R, I renamed the column titles and converted the data to data frame and the quality variable to integer.

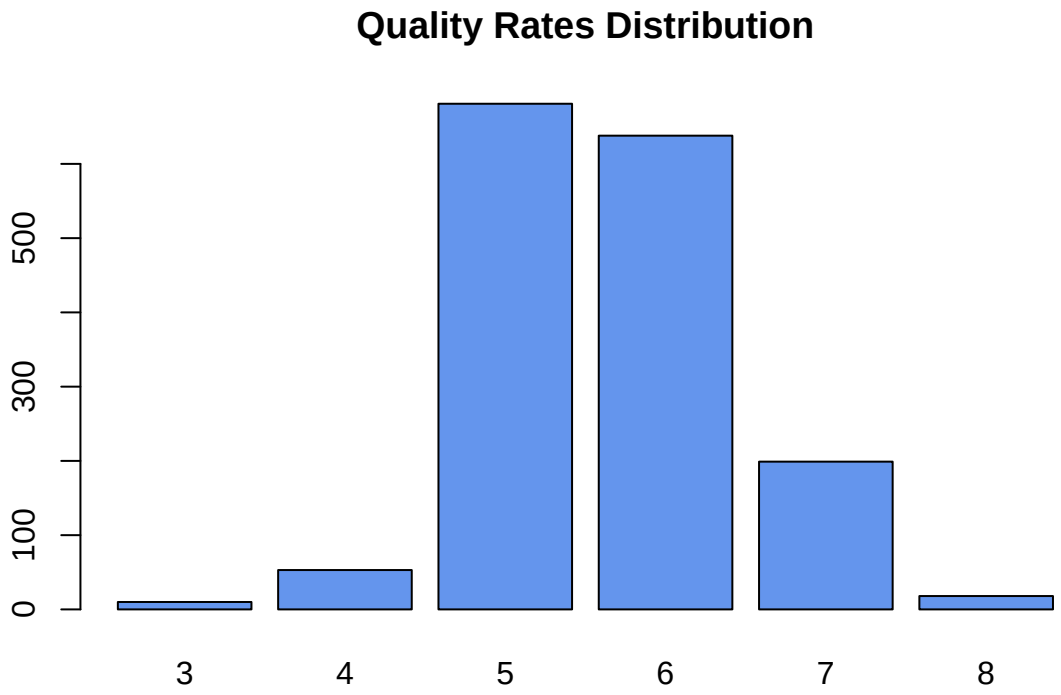
## Exploring the Data

We have 1599 observations and 12 variables in our dataset. Those 12 variables are:

Searching online on how to judge a good wine, the basics are that a good wine has balance. Balance: The relationship of four components - sweetness, acidity, tannin, and alcohol - to one another. Our data set has sweetness (residual sugar), acidity (ph, fixed acidity, volatile acidity and citric acid) and alcohol. Apparently, we don't have anything that can be related to tannin.

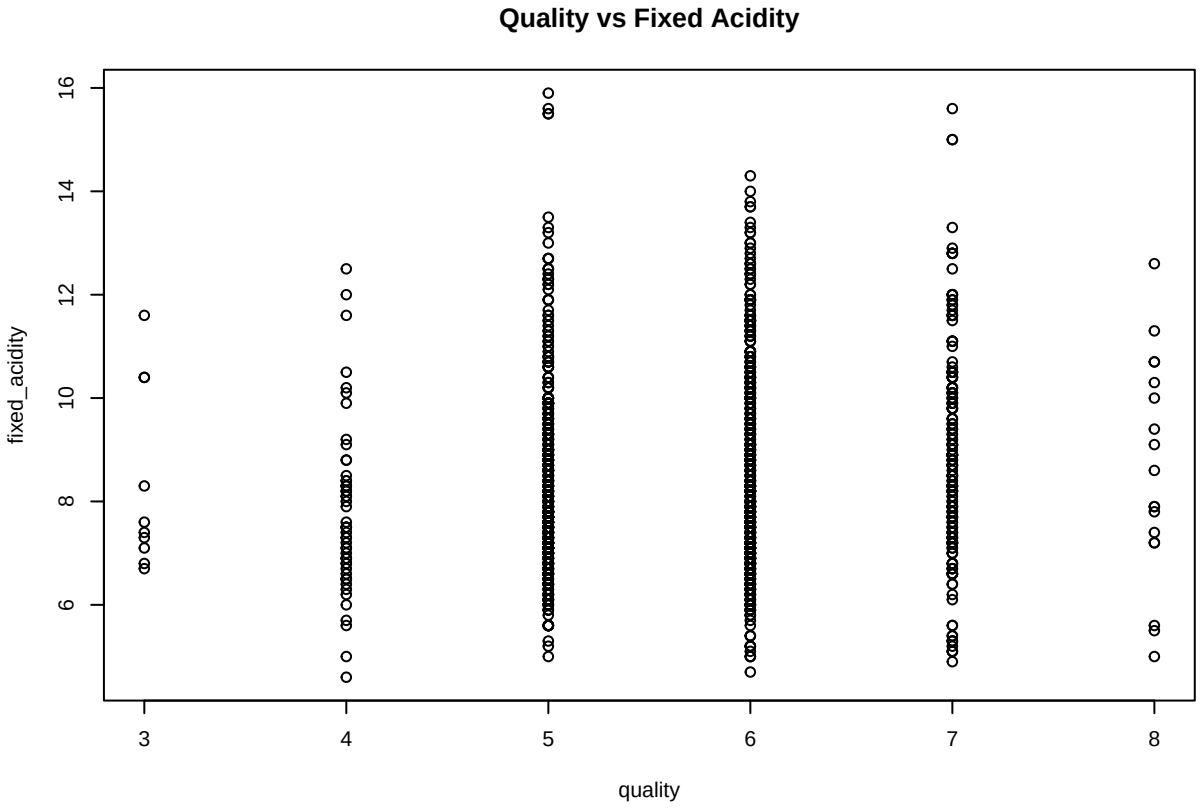
### Quality Distribution

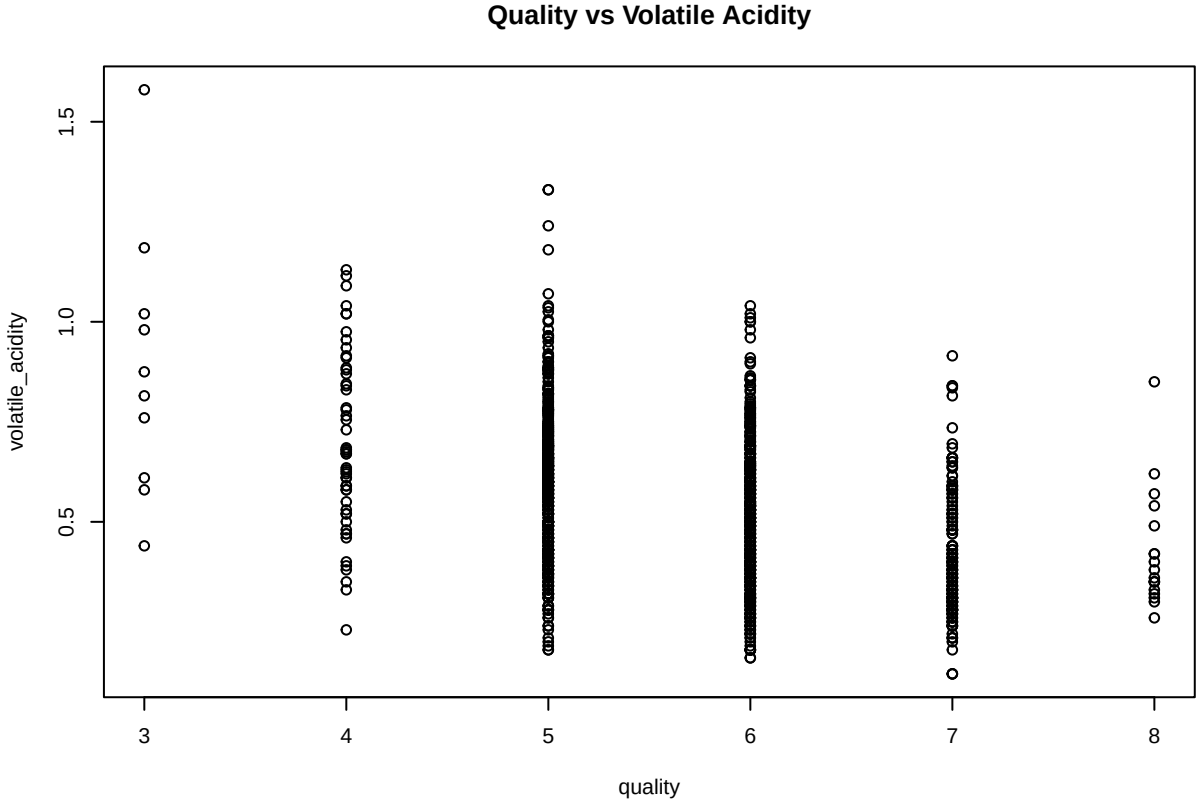
The plot below represents how many entries we have per quality rate.

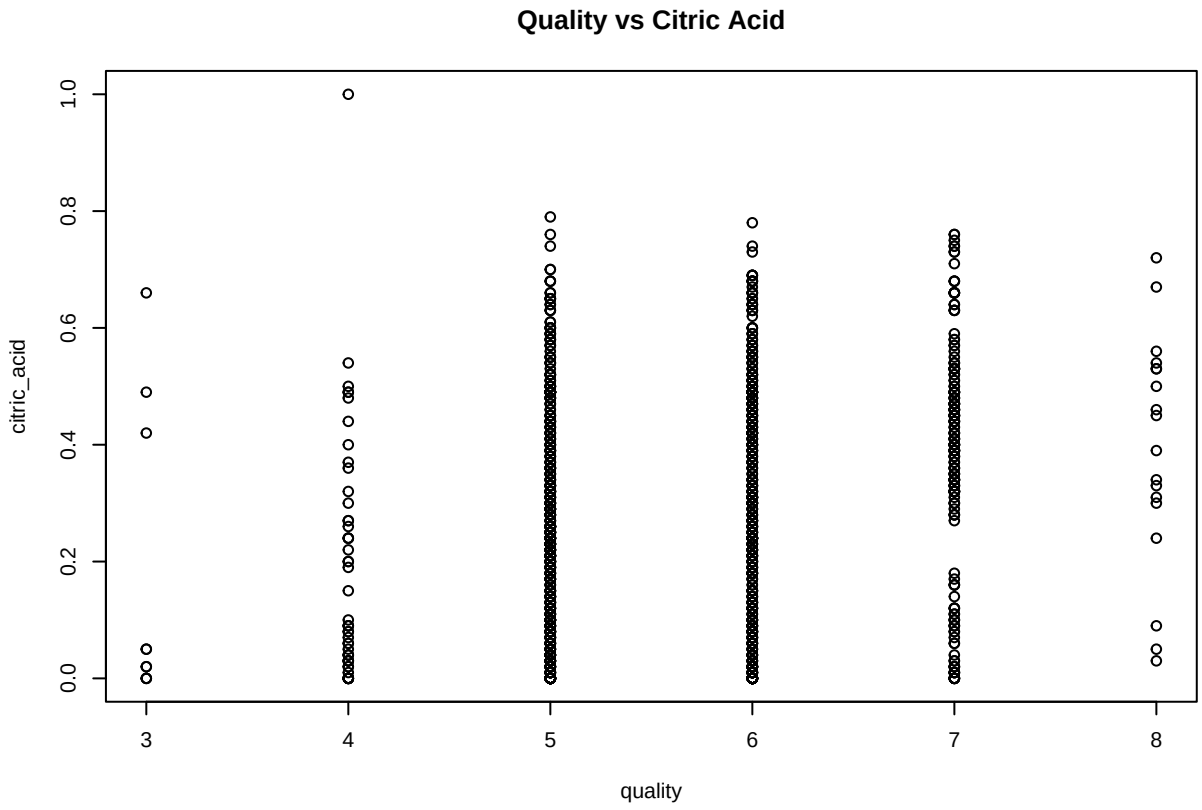


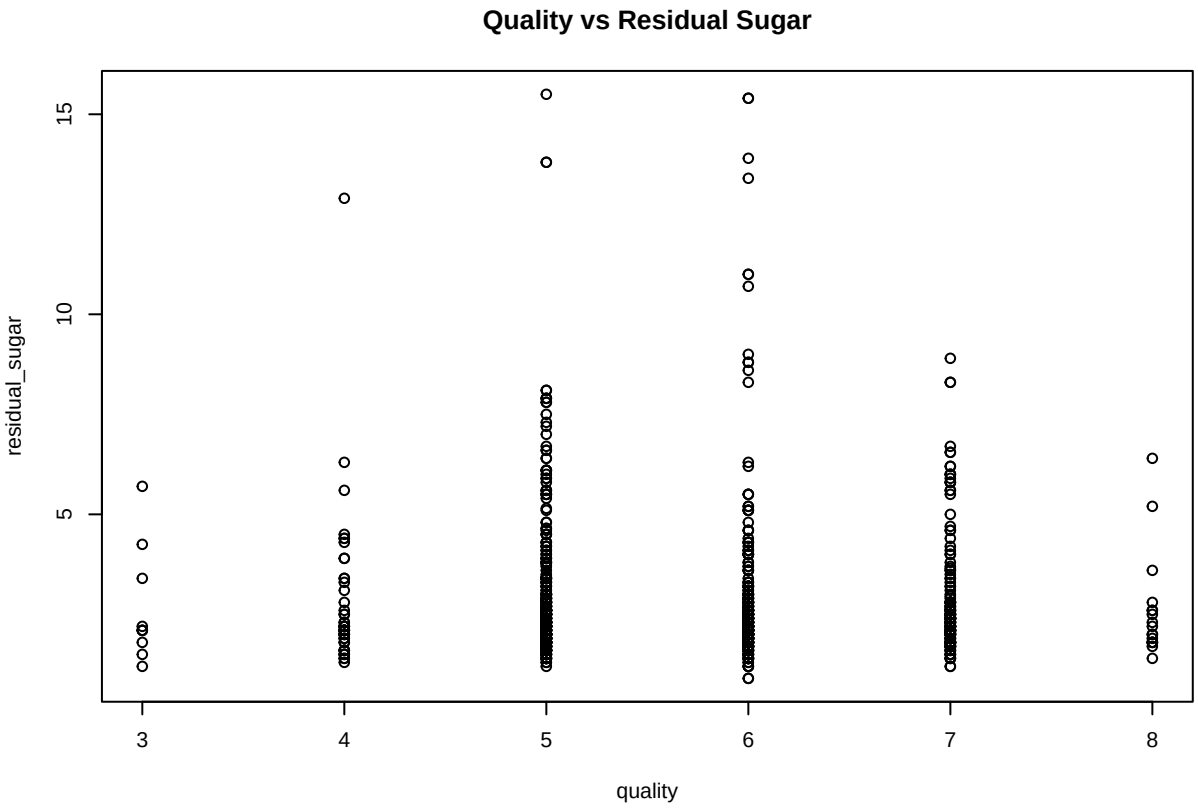

We can see that we have much more 'regular' wines than really bad or really good ones. If we consider a bad wine the ones rated from 3 to 5 and a good wine the ones rated from 6 to 8, the proportion seems to be balanced. we will classify the wines and check the same plot again later on.

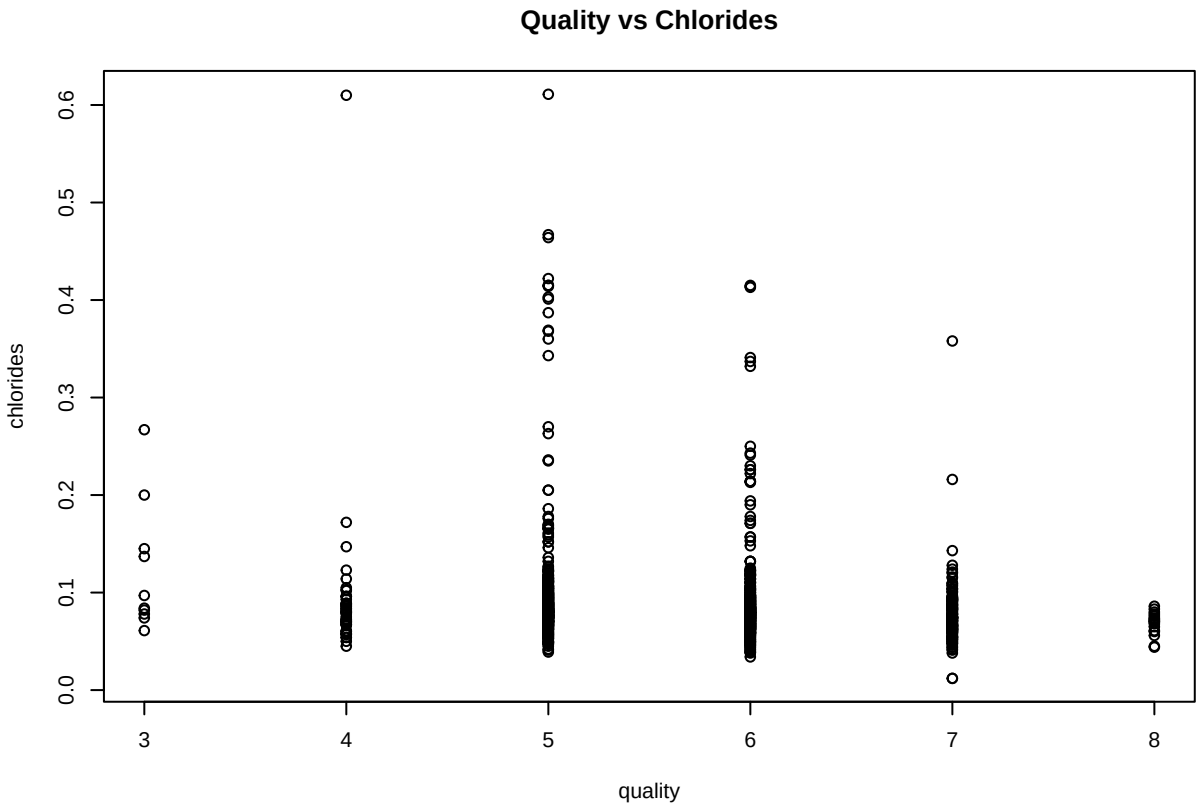
**Quality vs Variables**

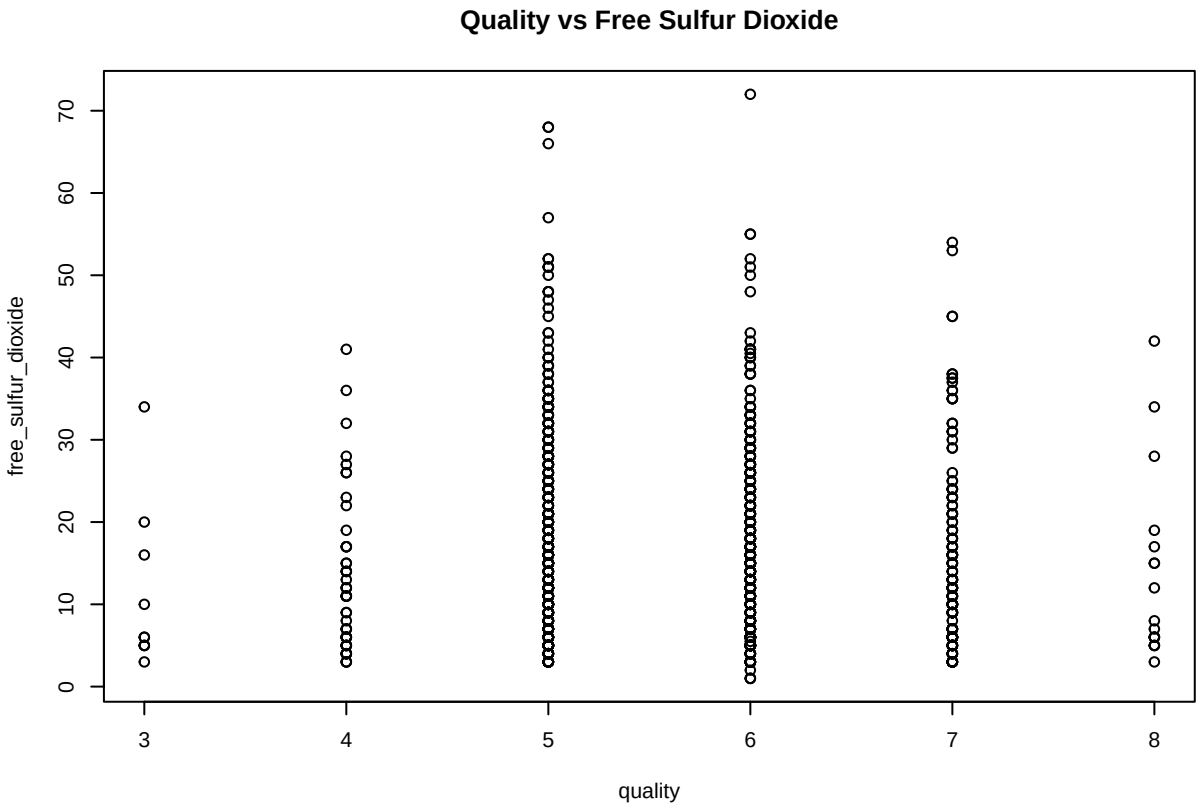


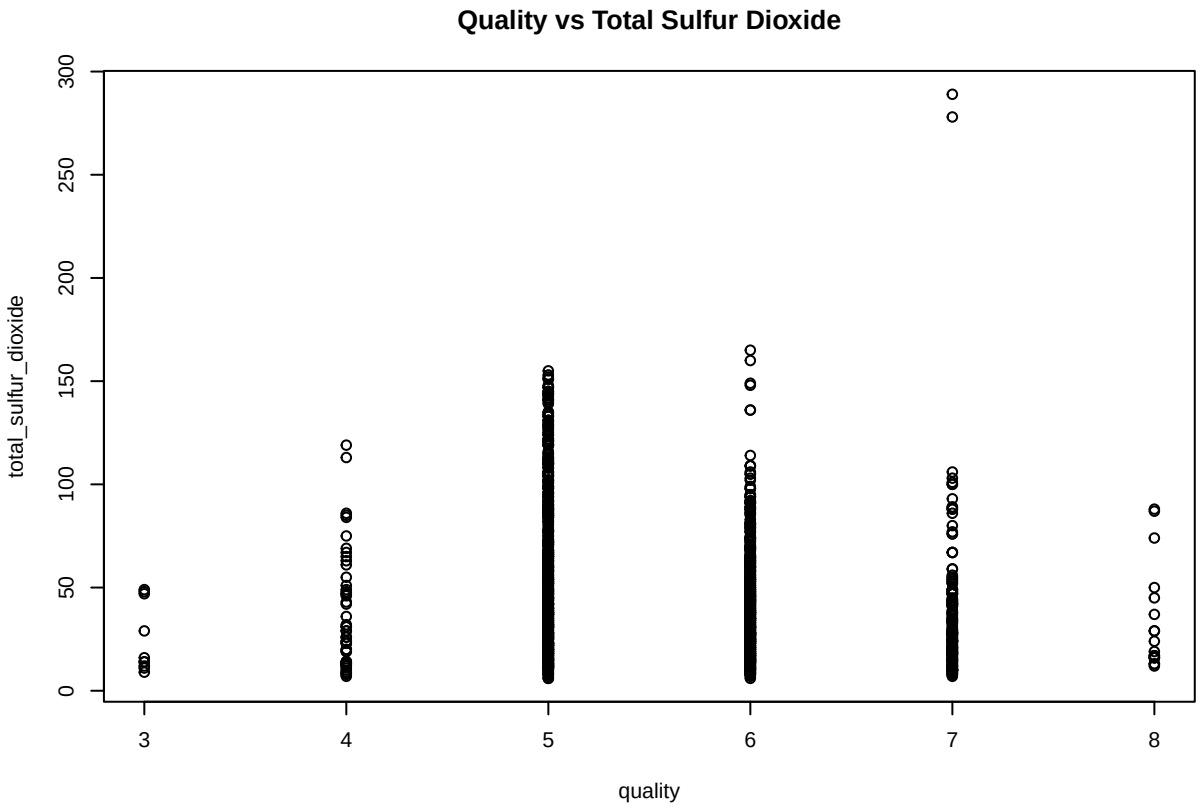



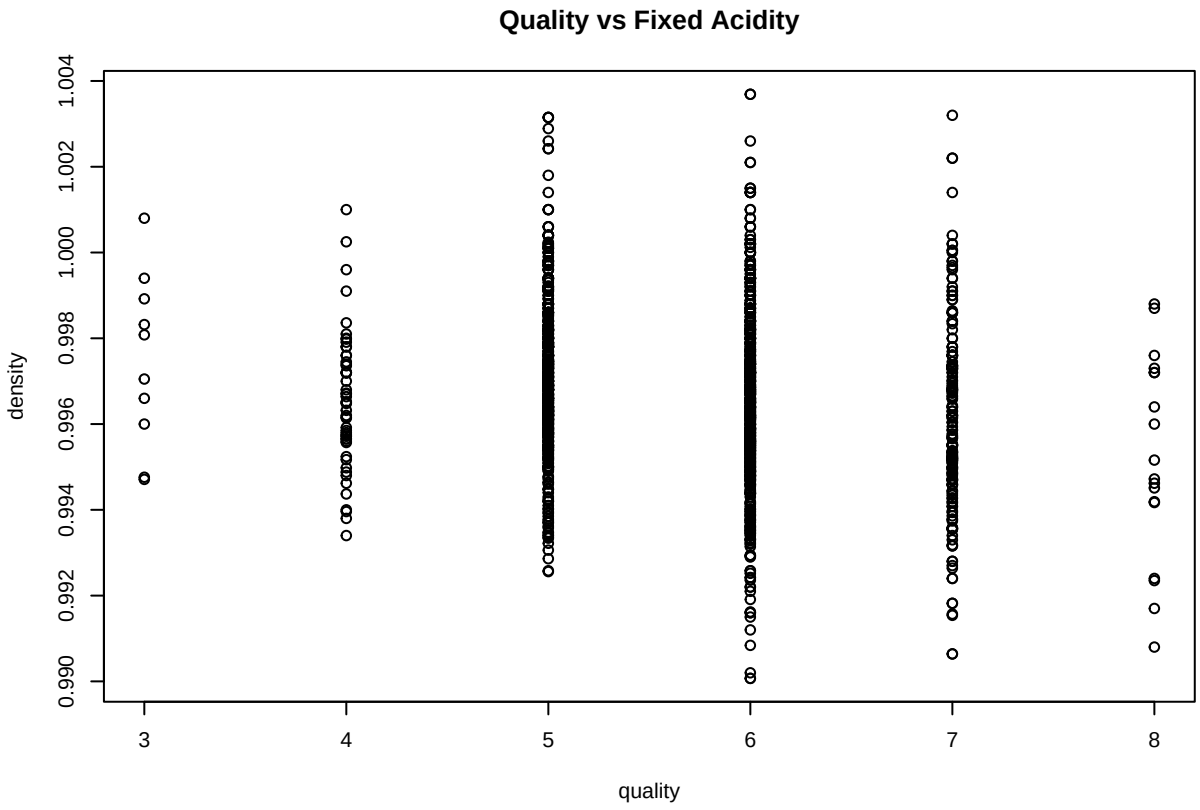

Quality vs Citric Acid
citric_acid
0.0
0.2
0.4
0.6
0.8
1.0
3
4
5
6
7
8
quality

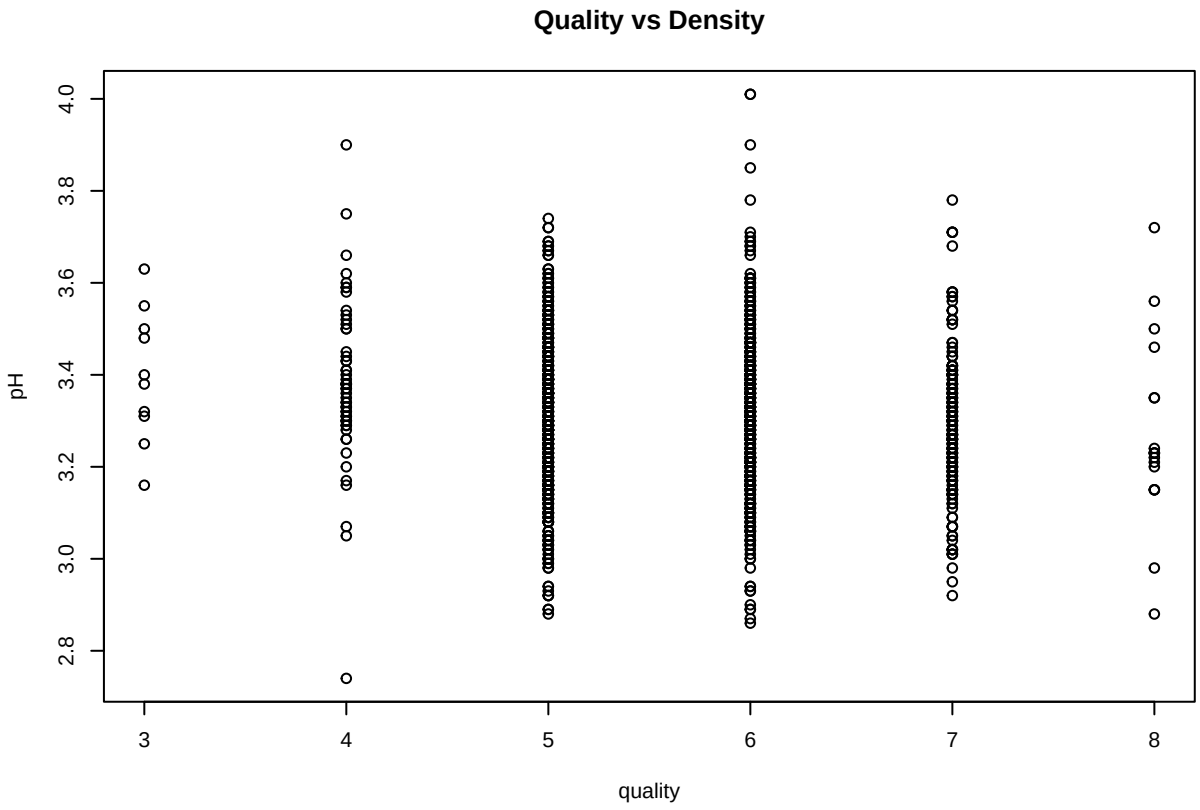

Quality vs Residual Sugar
residual_sugar
5
10
15
3
4
5
6
7
8
quality

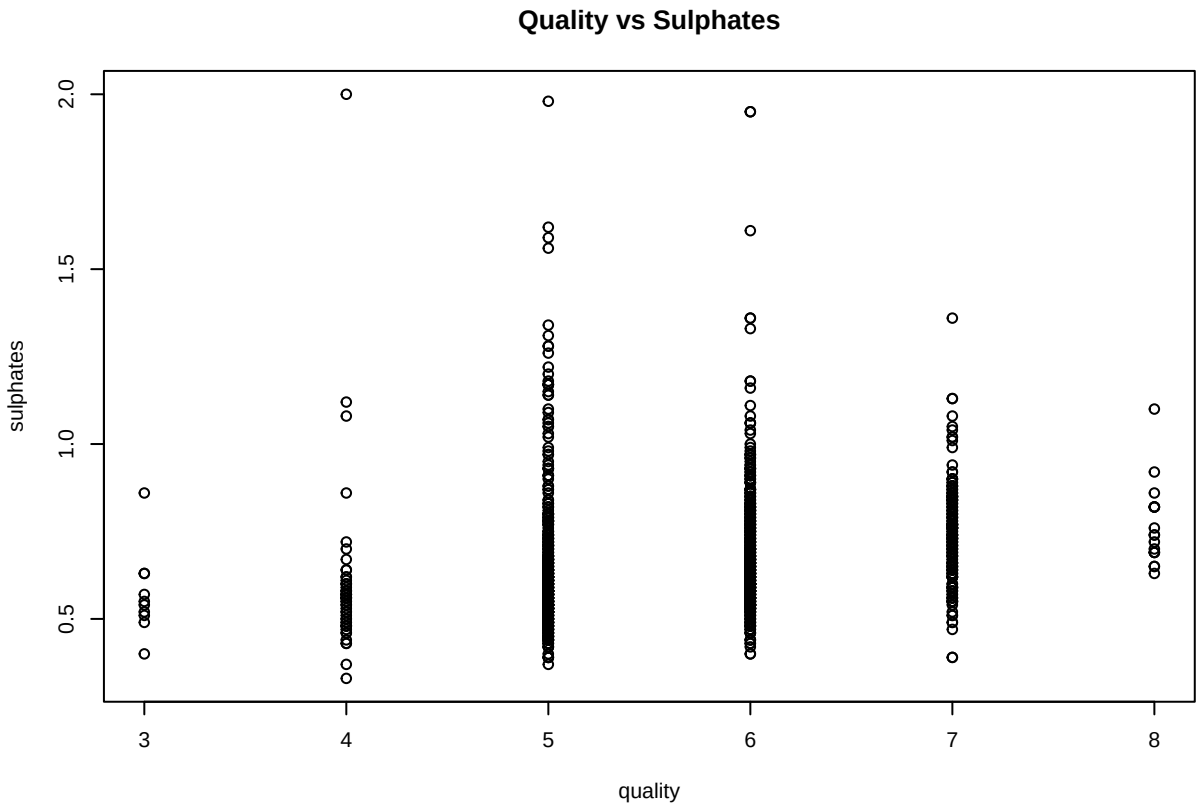

Quality vs Chlorides
chlorides
0.0
0.1
0.2
0.3
0.4
0.5
0.6
3
4
5
6
7
8
quality

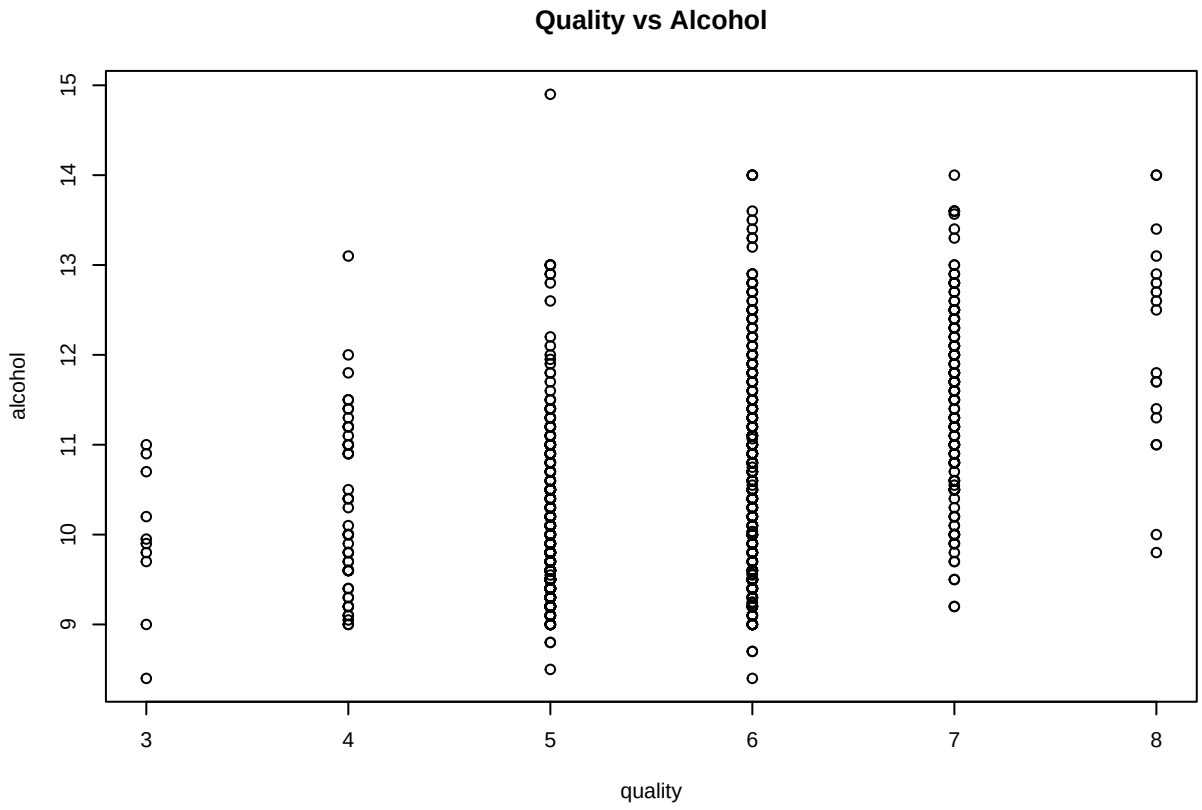

Quality vs Free Sulfur Dioxide
free_sulfur_dioxide
0
10
20
30
40
50
60
70
3
4
5
6
7
8
quality


Quality vs Total Sulfur Dioxide
total_sulfur_dioxide
0
50
100
150
200
250
300
3
4
5
6
7
8
quality


Quality vs Fixed Acidity
density
0.990
0.992
0.994
0.996
0.998
1.000
1.002
1.004
3
4
5
6
7
8
quality

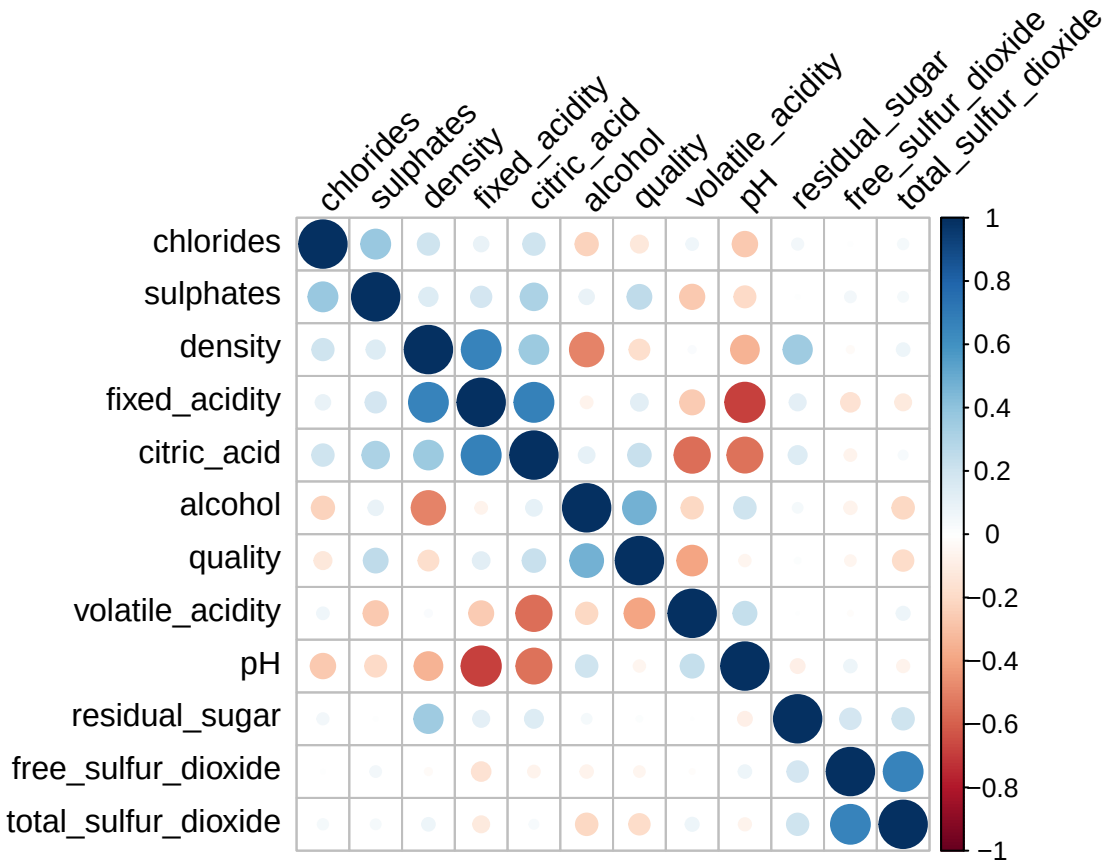

Quality vs Density
pH
2.8
3.0
3.2
3.4
3.6
3.8
4.0
3
4
5
6
7
8
quality


Quality vs Sulphates
sulphates
0.5
1.0
1.5
2.0
3
4
5
6
7
8
quality




### Correlation

Checking which variables have a higher correlation with quality.

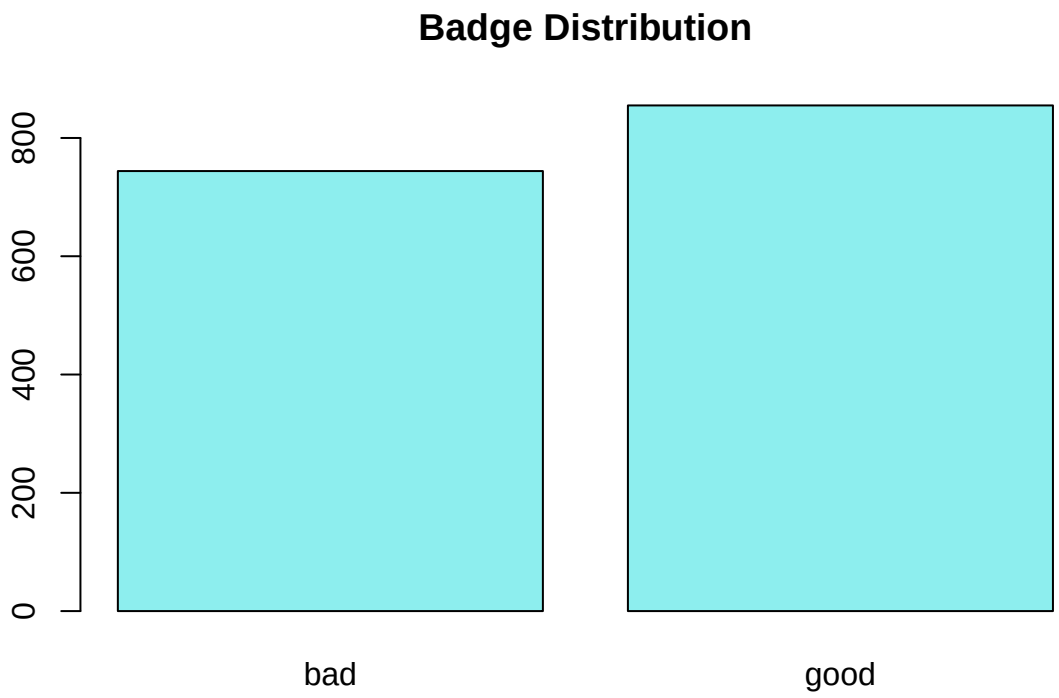



We can see that the 3 variables with higher quality correlation are alcohol, sulphates and citric acid.

## Classification

I will classify the wines as good or bad. For that, I created a new column on the dataset called "badge".

```
#Classifying the quality between good or bad

winedata<- winedata %>% mutate(badge = ifelse(quality >= 6, 'good', 'bad'))
winedata <- mutate(winedata, quality = as.factor(quality),badge = as.factor(badge))

str(winedata)
```




### Train and Test Set

Now we will split the dataset into training and testing sets. The train data will be used to build the model, while the test set will be used to predict new values and evaluate the results.

```
# The variables that will be used for predicting are all the variables but quality
winevar <- c("fixed_acidity", "volatile_acidity", "citric_acid",
          "residual_sugar", "chlorides", "free_sulfur_dioxide",
          "total_sulfur_dioxide", "density", "pH",
          "sulphates", "alcohol")


#Splitting thedata into training and test set
set.seed(1, sample.kind="Rounding")
train_index <- createDataPartition(y = winedata$badge,
                                                    times = 1,
                                                    p = 0.7,
                                                    list = FALSE)

train <- winedata[train_index, ]
test <- winedata[-train_index, ]


anyNA(train)
```

```r
str(train)
summary(train)
```

## Validation

### Confusion Matrix

The confusion matrix basically tabulates each combination of prediction and actual value. To provide precise definitions, we name the four entries of the confusion matrix:

| | Actual Positive | Actual Negative |
|---|---|---|
| Predicted Positive | True Positive (TP) | False Positive (FP) |
| Predicted Negative | False Negative (FN) | True Negative (TN) |

The multiple names can be confusing, so we include a table to help us remember the terms. The table includes a column that shows the definition if we think of the proportions as probabilities.

| Measure of | Name | Definition |
|---|---|---|
| sensitivity | True positive rate (TPR) | $\frac{TP}{TP+FN}$ |
| Specificity | True negative rate (TNR) | $\frac{TN}{TN+FP}$ |
| Specificity | Positive predictive value (PPV) | $\frac{TP}{TP+FP}$ |

## Prediction

### Linear Regression

Linear regression is a basic and commonly used type of predictive analysis. Linear regression approximates the multidimensional space of predictors $X$ and outcomes $Y$ into a function. The outcome, which is the wine badge, is categorical, so we need to convert to number before building the function. Here, we assign 1 to good quality wines and 0 to bad quality wines.

We will use only the three main variables to predict wine badge, according to our correlation analysis the main variables are alcohol, sulphates and citric acid.

```r
# Linear Regression
# Predict the wine badge (good/bad) based on citric_acid + chlorides + alcohol

# Train the linear regression model
fit_lm <- train %>%
  mutate(badge = ifelse(badge == "good", 1, 0)) %>%
  lm(badge ~ citric_acid + chlorides + alcohol, data = .)

# Predict
p_hat_lm <- predict(fit_lm, newdata = test)

# Convert the predicted value to factor
y_hat_lm <- factor(ifelse(p_hat_lm >= 1, "good", "bad"))

# Results
```

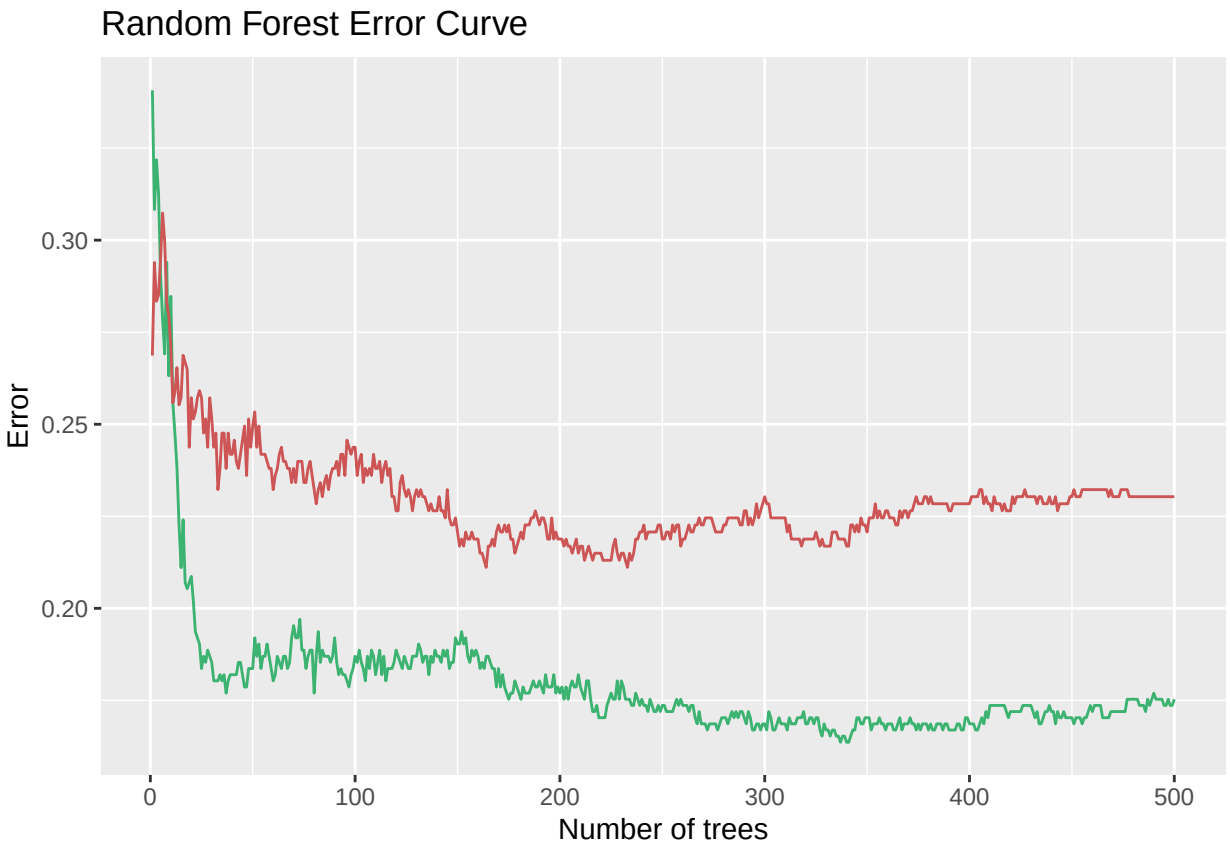
```
caret::confusionMatrix(y_hat_lm, test$badge)

result_lm <- caret::confusionMatrix(y_hat_lm, test$badge)
```

### Random Forest

Random forests is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

```
# Random Forest
# Predict the wine badge (good/bad) based on all the variables

#Formula
fml <- as.formula(paste("badge", "~",
                        paste(winevar, collapse=' + ')))

# Train the model
fit_rf <- randomForest(formula = fml, data = train)

# Predict
y_rf <- predict(object = fit_rf, newdata = test)

# Results
caret::confusionMatrix(data = y_rf,
                       reference = test$badge,
                       positive = "good")

result_rf <- caret::confusionMatrix(data = y_rf,
                       reference = test$badge,
                       positive = "good")
```

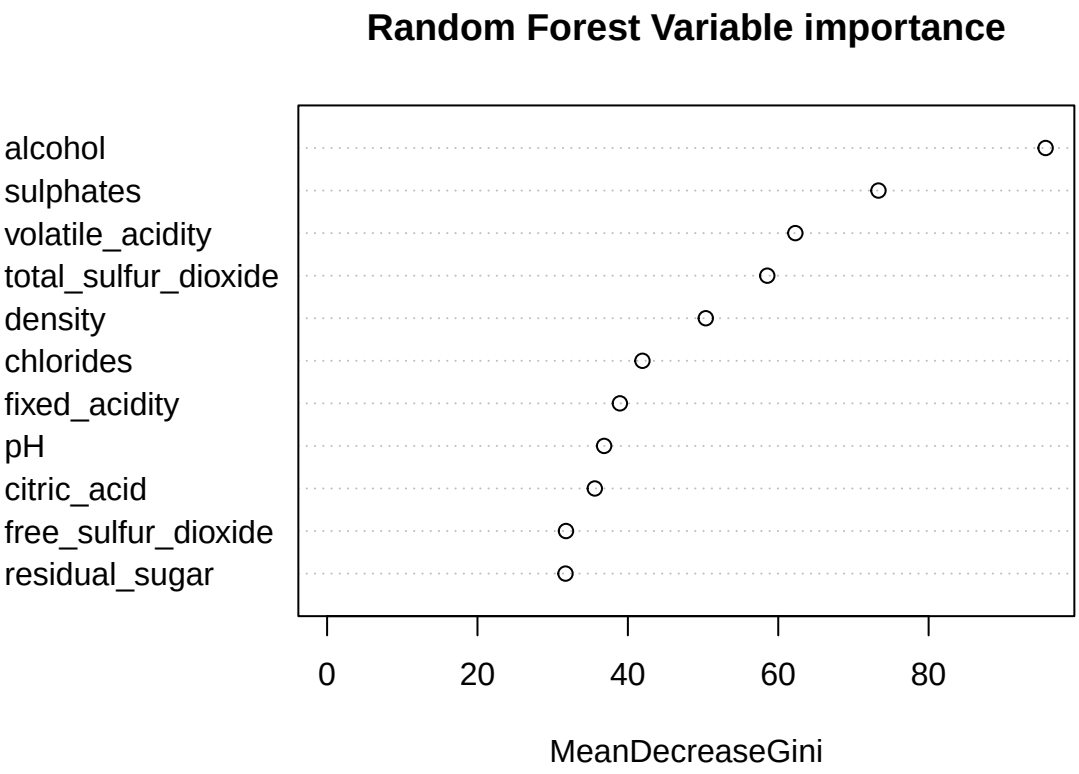
The chart below shows the error curve for good quality (green line) and bad quality (red line) wines.




The error decreases as the number of trees grows, stabilizing around 50 trees. The error for bad quality wines is higher than for good quality wines.

The most important variables for the random forest prediction are alcohol, sulphates and volatile acidity while residual sugar, free sulfur dioxide and citric acid have low predictive power.

```
# Variable importance plot
varImpPlot(fit_rf, main = "Random Forest Variable importance")
```




## Results

Random forest achieved much higher accuracy than the linear regression model.

**Linear Regression Model Accuracy:**

```
result_lm$overall["Accuracy"]
```

```
##  Accuracy
## 0.4947808
```

**Random Forest Model Accurancy**

```
result_rf$overall["Accuracy"]
```

```
##  Accuracy
## 0.8308977
```

## Conclusion

This report uses a publicly available data set of red wines physicochemical tests and quality based on sensory data. The data set was used to predict the wine quality badge (good or bad) based on the physicochemical variables.

The report shows the data exploration showing quality distribution, quality vs variable and variables correlation.

On classification, we classified the data by good quality or bad quality.

Before starting the predictions, the data has been split into train and test sets.

For predicting the wine quality badge we used linear regression model and random forest model and for validating the result we used the confusion matrix method.

### Limitations

The machine learning models used in this research aren't able to predict red wine quality badge with very high accuracy, specificity and sensitivity. This might be partially explained by the low prevalence of quality levels 3, 4 and 8.

Also, the wine quality is based on sensory data and each person has a different perception and personal taste on the quality.

## References


Rafael A. Irizarry, (2021) - "Introduction to Data Science" - https://rafalab.github.io/dsbook/

https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009 Data Source: Created by: Paulo Cortez (Univ. Minho), Antonio Cerdeira, Fernando Almeida, Telmo Matos and Jose Reis (CVRVV) @ 2009 Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

Leo Breiman and Adele Cutler (2018) - "Package 'randomForest' " https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

How to discern wine quality - https://www.dummies.com/food-drink/drinks/wine/how-to-discern-wine-quality/

R Markdown Cheat Sheet - https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf

Colorsin R - http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf