Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

CSE3005- Foundations of Data Analytics

J Component -Project Report

Fall Semester 2022-23

Topic: Wine Quality Analysis

By

UDAY SINGH SHERGILL	20BCE1806
SAI THARUN V	20BRS1066
NRHAAL REDDY A	20BPS1151

Submitted to

Dr. TRILOK NATH PANDEY
Assistant Professor,
SCOPE, VIT, Chennai

School of Computing Science and Engineering

VIT Chennai
Vandalur - Kelambakkam Road, Chennai - 600 127
FALL SEM 22-23

BONAFIDE CERTIFICATE

This is to certify that this project entitled "**Wine Quality Analysis**" submitted in partial fulfilment of the degree of B.Tech to Vellore Institute of Technology, Chennai, done by **Mr. Uday Singh Shergill, Regd. No.20BCE1806, Mr. Sai Tharun V, Regd. No.20BRS1066, Mr. Nehaal Reddy A, Regd. No.20BPS1151** is an authentic work carried out the project work for the subject **"Foundations of Data Analytics"** under my supervision for the fall semester 2022-23 by them under my guidance. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of our knowledge and belief.

SIGNATURE

Dr TRILOK NATH PANDEY
FACULTY NAME

Assistant Professor
SCOPE

ACKNOWLEDGEMENT

We would like to convey our sincere gratitude to our project advisor,
Dr. Priyadarshini R, SCOPE, for her constant support and insightful advice that she provided to us in a pleasant manner during the project work as well as for inspiring us to complete the study on the topic “Wine Quality Analysis”.

We would want to take this opportunity to thank the faculties of the college for their guidance and support throughout the project.

Lastly, we wish to express our gratitude to our parents, relatives, and friends for their support throughout the execution and helped us in every possible way and displayed appreciation for our project and for the opportunity they provided us in undergoing this course at such a prominent institution.

TABLE OF CONTENTS

ABSTRACT

Drinking wine may not be something we Indians may look up to but when we consider the broader world, wine drinking is a culture followed by many countries and people regard it essential for their living, and such wine is not cheap either. So, people expect to get quality goods for the money they pay. But there always are a set of people who aim to cheat people out of their money and earn some profit. It also involves health risks since drinking bad wine may cause altitude of problems for both those who drink it and are around them in various ways.

To produce good tasting and quality wine, there are many factors involved in this to get the same. Also, to check for the quality and taste we mostly hire wine tasters and get the opinion of each wine, using this data we analyse which wine has been preferred by most of the wine tasters. But it is not always feasible and accurate to predict the best factors effecting the wine without studying them.

So, to solve this setback our project aims predict the quality of a wine sample based on its physicochemical factors which will involve the creation of training sets using the data collected from the vineyards and testing based on new wine to test the quality. This will help the wine producers to improve their quality of wine. Using this analysis and result they can extract the important factors effecting the quality and taste of wine. Working on the concentrations of such factors will help them to increase the quality and the produce the wine that is preferred by the consumers at a higher rate. This will also increase their sale and give an interest at an increasing rate.

INTRODUCTION

Wine industry shows a recent growth spurt as social drinking is on the rise. The price of wine depends on a rather abstract concept of wine appreciation by wine tasters, opinions among whom may have a high degree of variability. Pricing of wine depends on such a volatile factor to some extent. Another key factor in wine certification and quality assessment is physicochemical tests which are laboratory-based and take into account factors like acidity, pH level, the presence of sugar and other chemical properties.

For the wine market, it would be of interest if human quality of tasting can be related to the chemical properties of wine so that certification and quality assessment and assurance process is more controlled. So, we want to study the importance of the features for the prediction of wine quality to see which yields the highest accuracy and to determine which features are the most indicative of a good quality wine.

We collected the dataset and used it as a source of description about each type of wine to analyse each variable. A predictive model developed on the dataset is expected to provide guidance to vineyards regarding quality and price expected on their produce without heavy reliance on the volatility of wine tasters. A prescriptive model developed on the dataset is expected to provide knowledge to vineyards about best quality wines and prescribing which variable is their wine lacking in order make it better quality for the consumers.

LITERATURE REVIEW

Nowadays, industry players are using product quality certifications to promote their products. This is a time-consuming process and requires the assessment given by human experts, which makes this process very expensive. A vital factor in wine certification and quality assessment is physicochemical tests, which are laboratory-based and consider factors like acidity, pH level, sugar, and other chemical properties. The wine market would be of interest if the human quality of tasting can be related to wine's chemical properties so that certification and quality assessment and assurance processes are more controlled.

- ## Real life situation:

Depending on a vineyard's size and willingness to invest in technology, there are endless opportunities to take advantage of data. Starting with the growing of the grapes, winemakers can measure the water each grape is getting. There are also weather stations that can gather data to help predict the best time to harvest the crops. Drones can help evenly apply any pest control.

After the wine has been prepared a sample can be sent to a lab for quality testing and its chemical factors can be measured there. Similarly, the data related to the wine from different wine yards has to be collected by a single body to make use of it for further research.

- ## Advantages of this data collection strategy:

The advantage of this strategy is that since a single governed body that won't be related to any vineyard will be collecting the data there won't be any altercations and the processing of the data can be done by the same institution to publish a standard result that will prefix the quality of wine.

- ## Limitations and risks of this data collection strategy:

The limitation in this method is that it will take a lot time to collect data all around the world from different vineyards and it may change over time. This will lead to an underwhelming process of wine quality maintenance that we aim to provide.

The risk in this process is, since the vineyards are old players, they may try to bribe the central institution's board to change their results which may lead to improper data collection which makes the whole process meaningless. There won't be any ethical issues since the data is about an inanimate object and it's a customer commodity.

RESEARCH PAPERS

Multivariate Methods Based Soft Measurement for Wine Quality Evaluation

- In this paper, the model-based modelling is utilized and the model is built by the multivariate methods based on soft measurement.
- In this paper, in order to evaluate red wine quality, the multivariate methods based on soft measurement are used. This algorithm can help construct the fitted wine quality model and predict wine quality.
- The physicochemical indexes that can impact the quality of wine are also proposed in this paper.

Conclusions

- In order to classify and identify red wine, proposed the technology based on the spectrum and pattern recognition.
- In this section, we use three multivariate statistical methods based on soft measurement, including principal component regression, partial least squares regression, and modified partial least squares regression to find which is the best method according to the relationship between the red wine physicochemical indexes and the wine quality.
- It is not utilized when the number of samples is smaller than the number of variables.
- With these methods and the real data obtained in practice, wine quality prediction models have been constructed.
- One purpose of this work is to choose the best method among OLSR, PLS, PCR, and MPLSR.
- With the science and technology transformed into productivity, the advanced technology is widely used in industry.

Multivariate Statistical Analysis Applied in Wine Quality Evaluation

- Cluster weights reflect the importance of the index, which is the advantage of weighted cluster analysis.
- And then the wine samples were clustered by Weighted Cluster Analysis, where weights were determined by information entropy.
- It is worth mentioning that the weights were respectively determined by information entropy method for red wine of the first category and the second category.

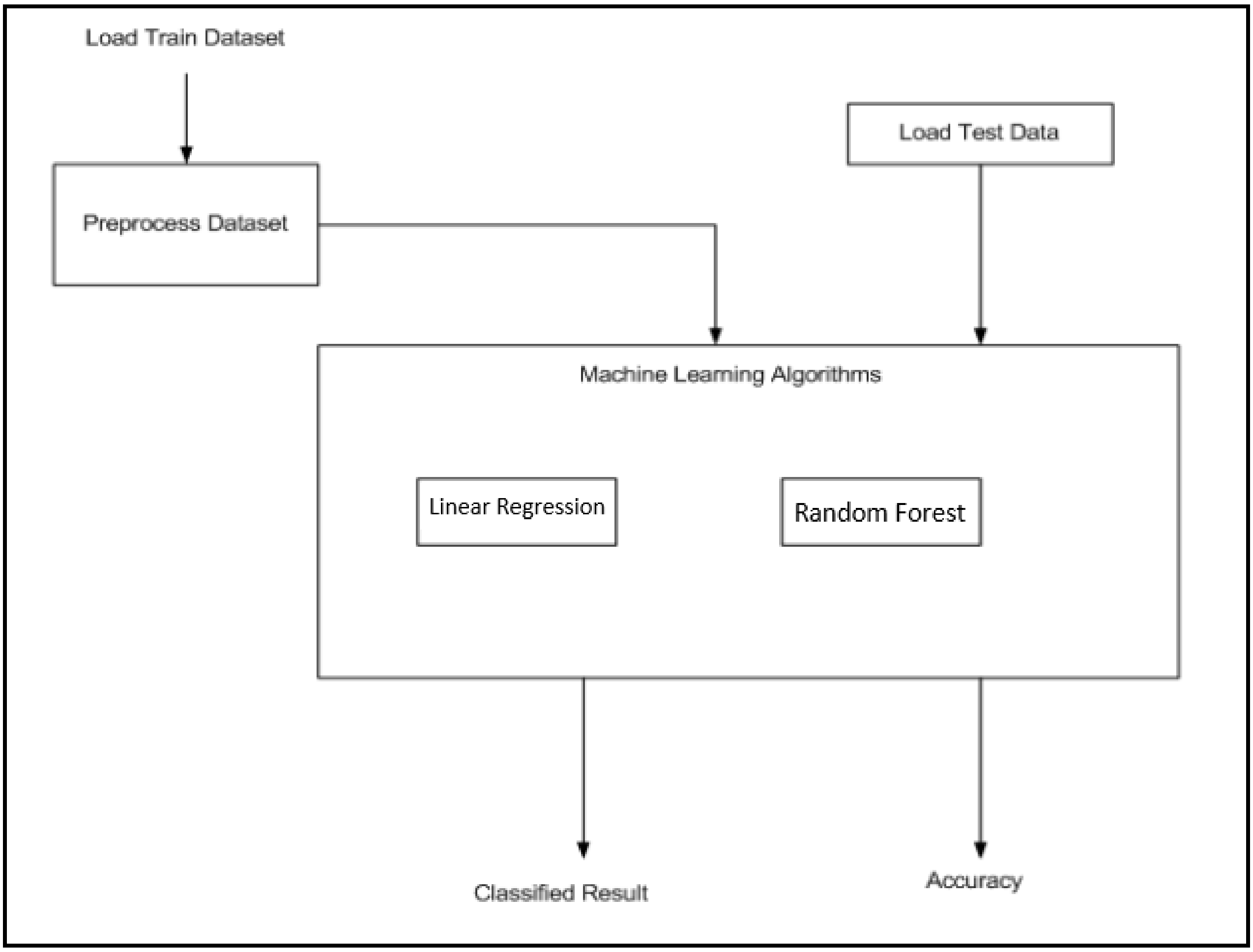
Results

- To be exact, the optimal value stood for the quality of wine.
- From, we found that most optimal values were less than 0.5.
- And the distance between the value of the best wine and that of the worst wine in three categories were bigger than 0.5, which showed good discrimination of using TOPSIS method.

Conclusions

- As is shown in, a total 89.06% of data information was explained by four principal components.
- From, we knew that component 1 of the red wine contained information of anthocyanin.
- Component 2 of the red wine contained information of a*, C, which could be named chromaticity factors.
- We grasped the principal components of the physicochemical indicators using Principal component analysis.
- The results of BP neural network showed that there was narrow difference between iterative weights and initial weights, which proved that weights determined by information entropy had a high accuracy.
- What's more, the model can be widely used in food and other quality evaluation.

ARCHITECTURE DESIGN



MODULE EXPLAINATION

1. Define the Problem:

- Show the contribution of each factor to the wine quality in your model
- Show which features are more important in determining the wine quality
- Show which features are less important in determining the wine quality

2. Data Gathering and understanding:

- Our analysis will use Wine Quality Data Set, available on the UCI machine learning repository (https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality.csv).
- The dataset contains a total of 12 variables, which were recorded for 4,898 observations. This data will allow us to create different regression models to determine how different independent variables help predict our dependent variable, quality. Knowing how each variable will impact the wine quality will help producers, distributors, and businesses in the red wine industry better assess their production, distribution, and pricing strategy.
- The aim of our project is to analyze wine dataset to predict the quality of a wine sample based on its physicochemical factors which will involve the creation of training sets using the data collected from the vineyards and testing based on new wine to test the quality.
- In our project the training and testing sets are created from the dataset we acquired and they will be used during the model building to provide a sample for the real-world solution.

3. Data Preparation and Exploration:

- Data Preparation:
  - ✓ First step to clean and prepare the data for analysis
  - ✓ First, we check the data types focusing on numerical and categorical.
  - ✓ Then, identifying missing values existing in our data set.
  - ✓ Lastly, checking for problems like outliers and abnormal distributions.

- Data Exploration:
    - ✓ Helps in interpreting each feature of wine data.
    - ✓ To know which variables are likely to affect the quality of wine the most.

4. Modelling with algorithms:

- Multivariable Regression
    - To build an optimal prediction model for wine quality.
- Random Forest
    - ✓ Helps to create a random sample of multiple regression decision trees.
    - ✓ Merges them to obtain a more stable and accurate prediction through cross-validation.

5. Performance Metrics:

In the end we will evaluate our model (Regression and Classification) using a few metrics:

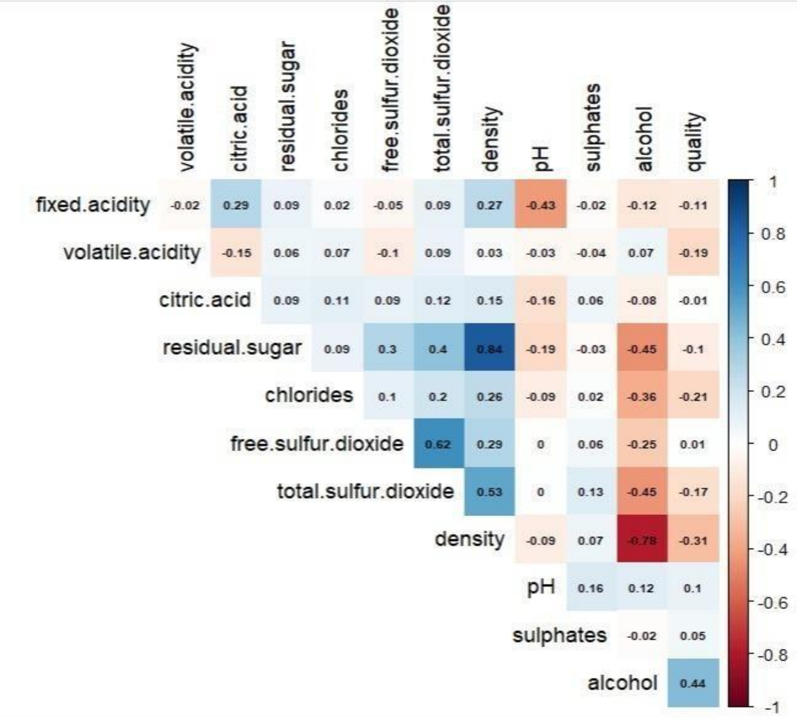
- Skew: a normal distribution close to zero is a perfect distribution
- MSE (Mean Squared Error): is an absolute measure of fit. Note that an MSE of 0 indicates a perfect fit
- RMSE (Root Mean Squared Error): is a good measure of how accurate the model predicts the target
- R-Squared: is a relative measure of fit

Also, the use of BIC (Bayesian Information Criterion) for model selection in measuring complexity

The model with the lower BIC Value is preferred.

IMPLEMENTATION

To see which variables are likely to affect the quality of red wine the most, I ran a correlation analysis of our independent variables against our dependent variable, quality. This analysis ended up with a list of variables of interest that had the highest correlations with quality.



In order of highest correlation, these variables are:

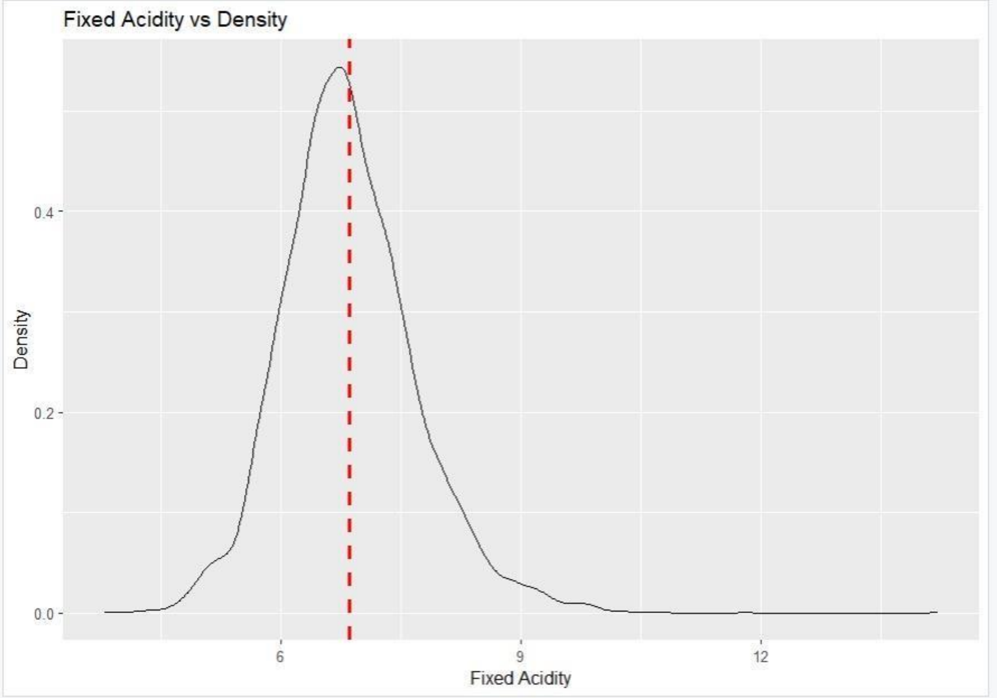
1. Alcohol: the amount of alcohol in wine
2. Volatile acidity: are high acetic acid in wine which leads to an unpleasant vinegar taste
3. Sulphates: a wine additive that contributes to SO2 levels and acts as an antimicrobial and antioxidant
4. Citric Acid: acts as a preservative to increase acidity (small quantities add freshness and flavor to wines)
5. Total Sulfur Dioxide: is the amount of free + bound forms of SO2
6. Density: sweeter wines have a higher density
7. Chlorides: the amount of salt in the wine
8. Fixed acidity: are non-volatile acids that do not evaporate readily

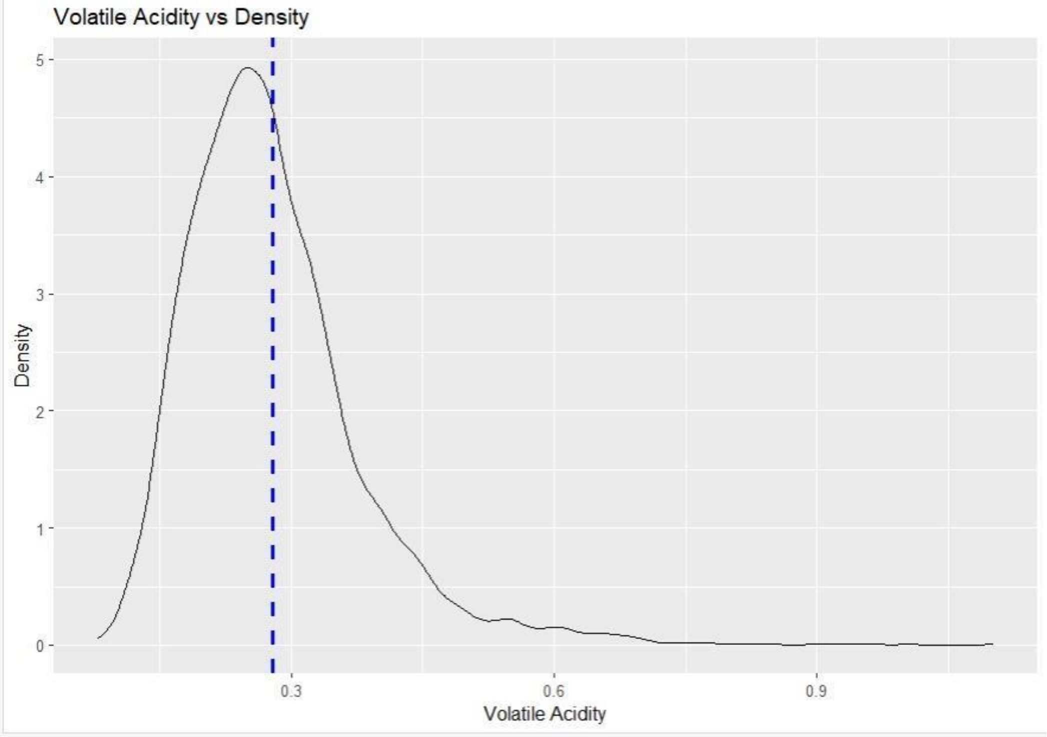
9. pH: the level of acidity

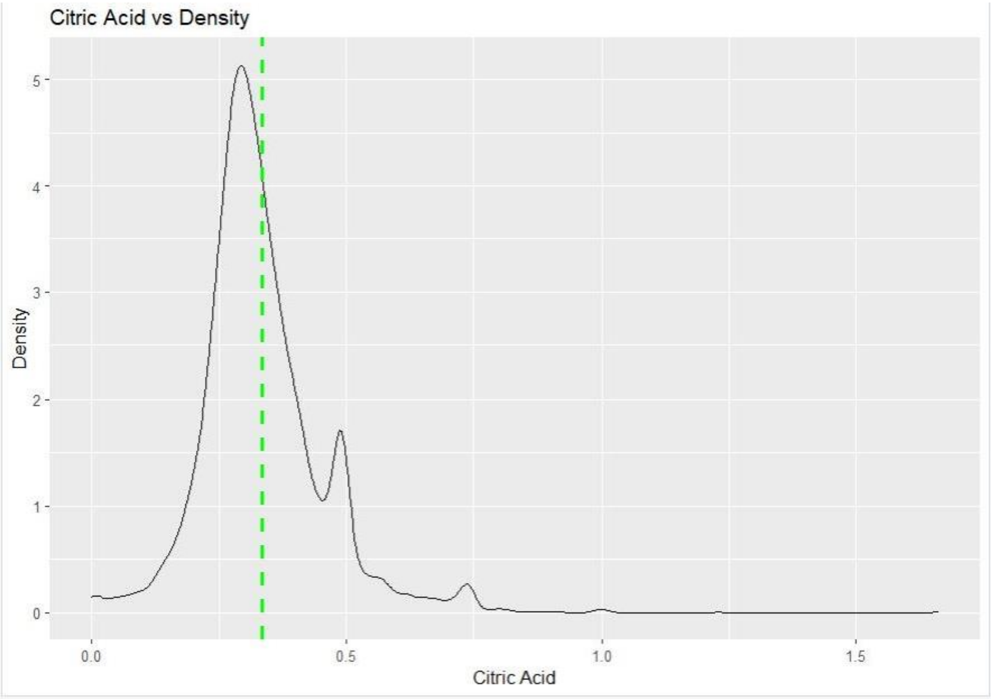
10. Free Sulfur Dioxide: it prevents microbial growth and the oxidation of wine

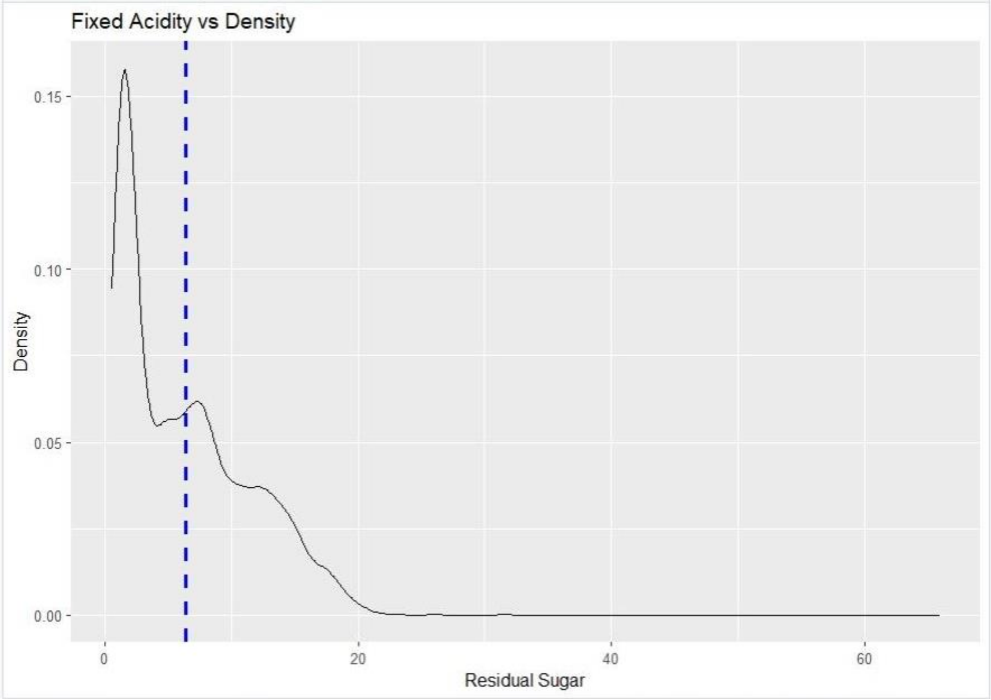
11. Residual sugar: is the amount of sugar remaining after fermentation stops. The key is to have a perfect balance between — sweetness and sourness (wines > 45g/ltrs are sweet)

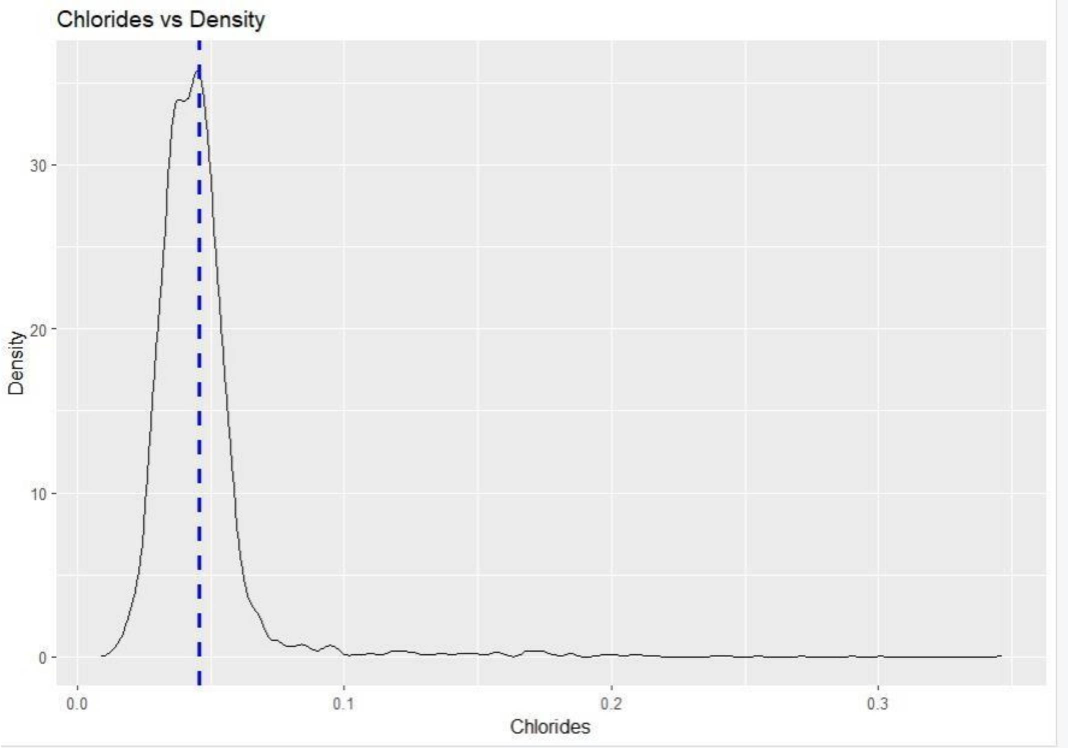
Next, for independent numerical variables, the first step to further analyze the relationship with our dependent variable was to create density plots visualizing the spread of the data.

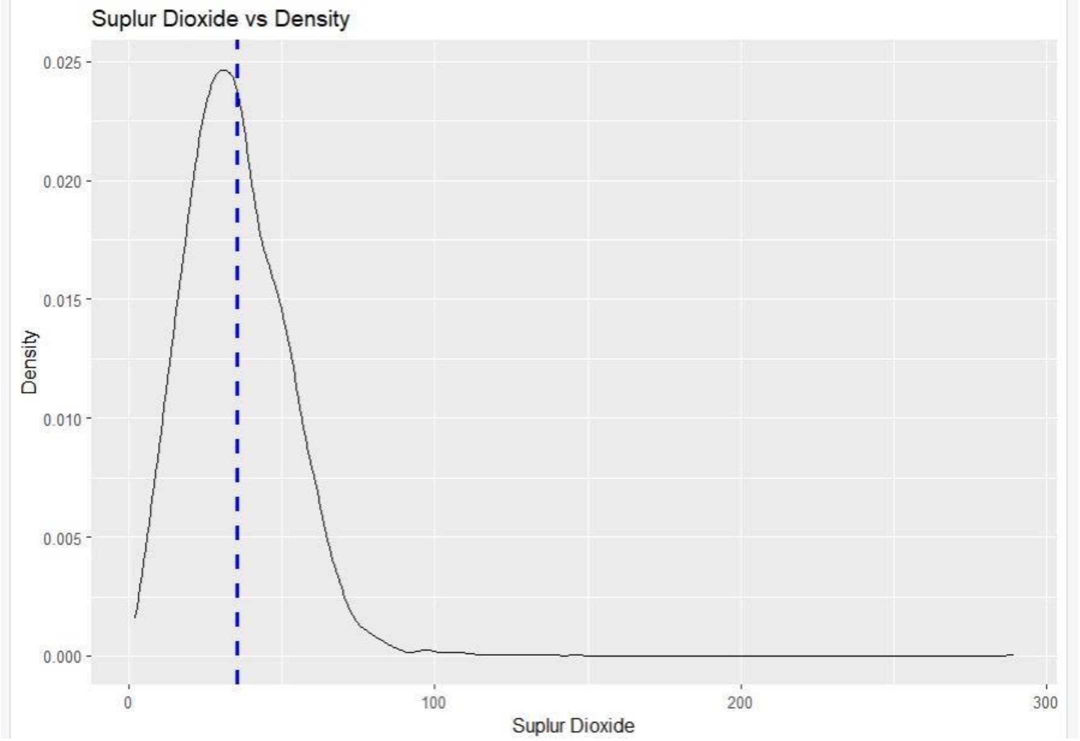


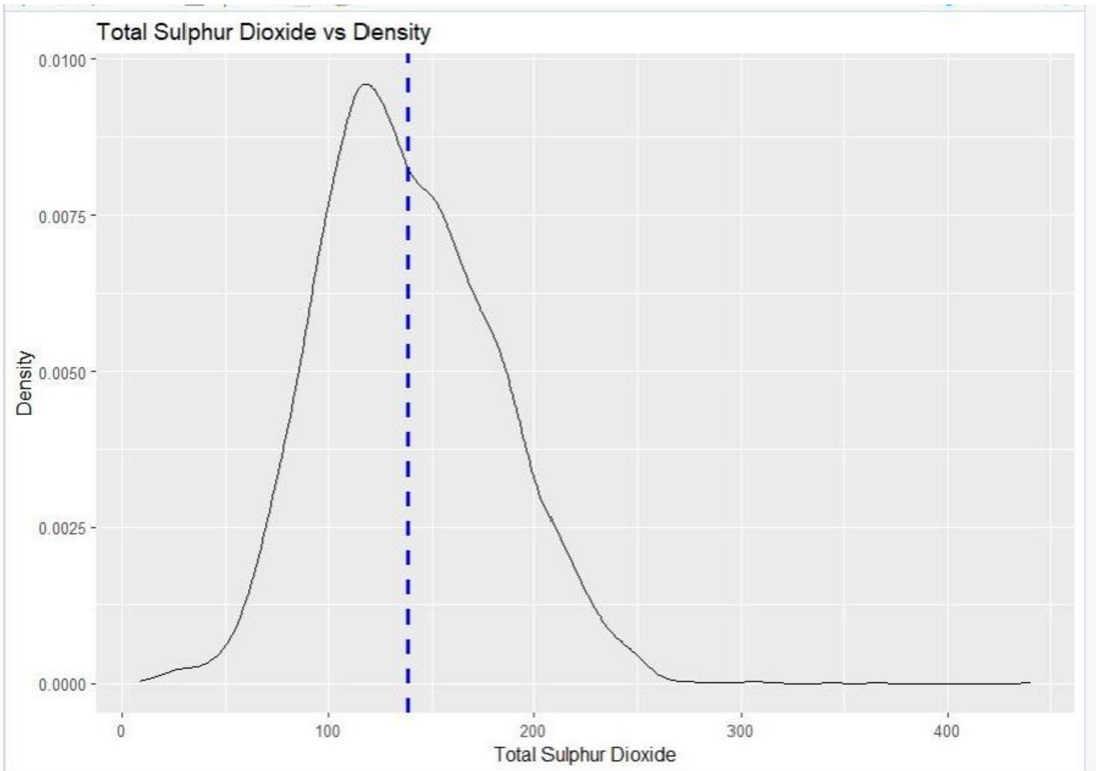


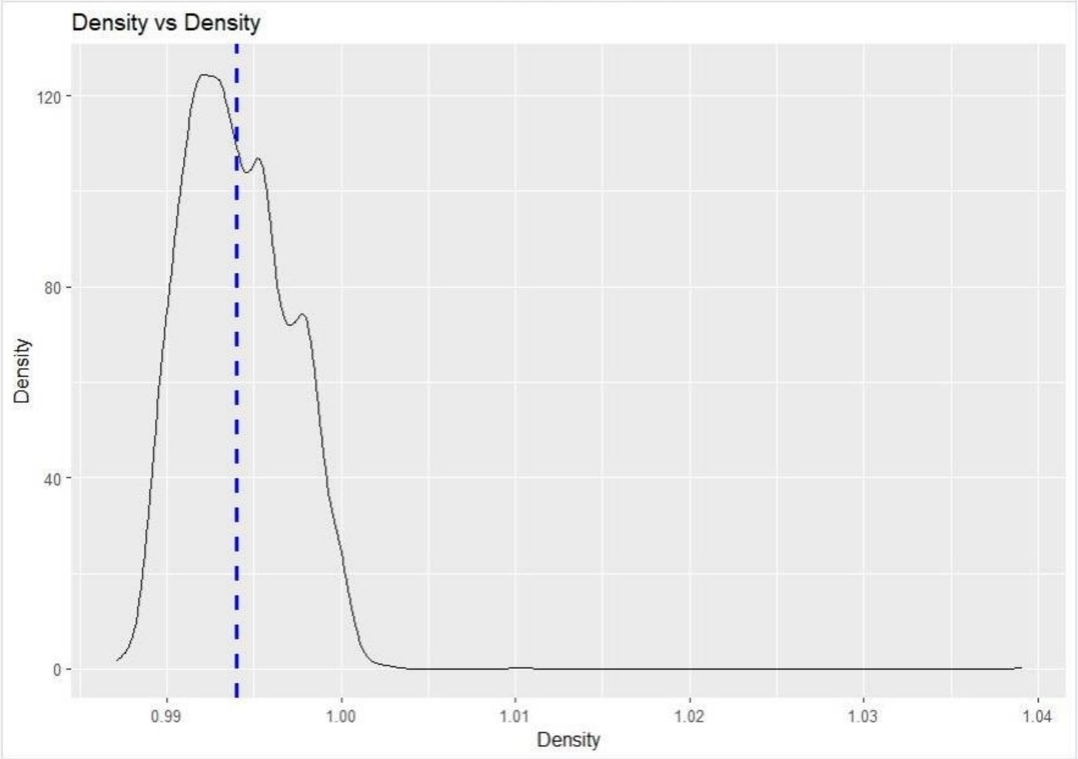

Citric Acid vs Density
5
4
3
2
1
0
Density
0.0
0.5
1.0
1.5
Citric Acid

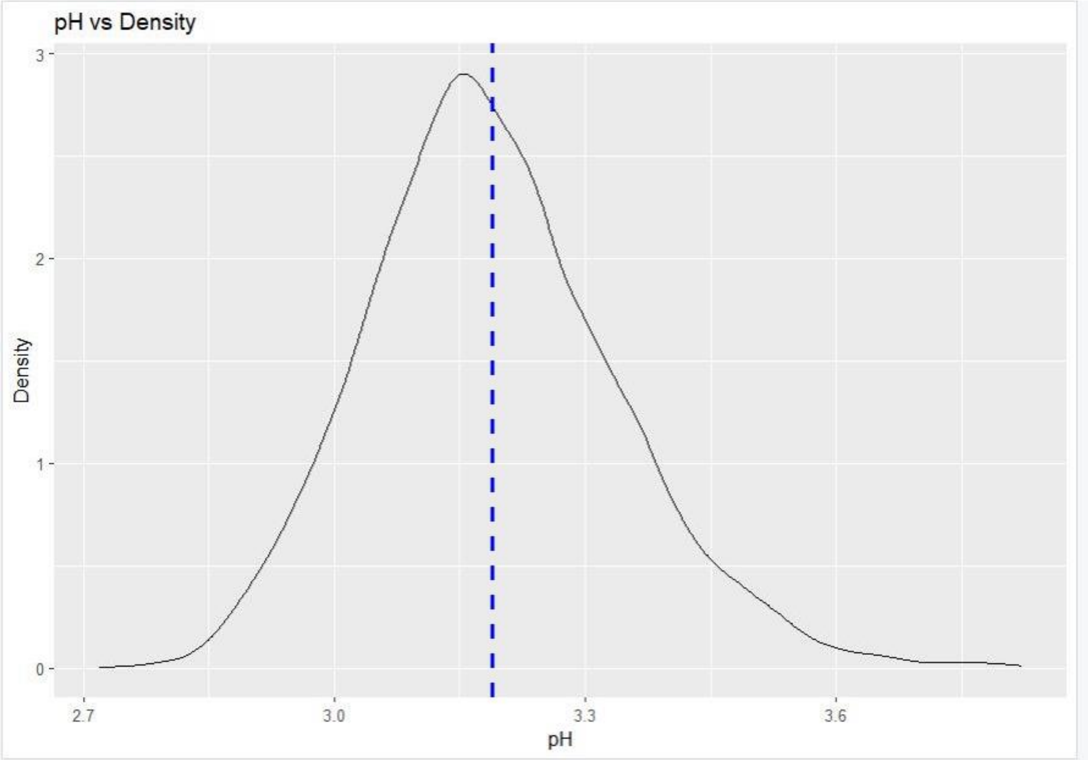

Fixed Acidity vs Density
0.15
0.10
0.05
0.00
Density
0
20
40
60
Residual Sugar

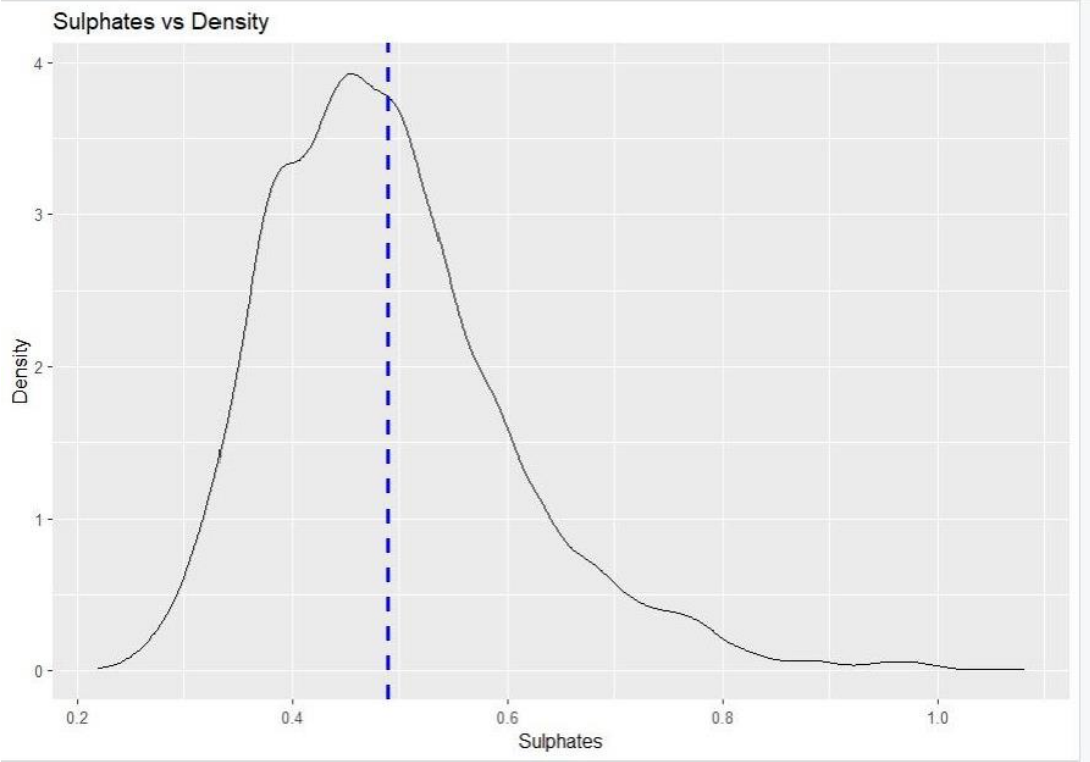

Chlorides vs Density
Density
30
20
10
0
0.0
0.1
0.2
0.3
Chlorides

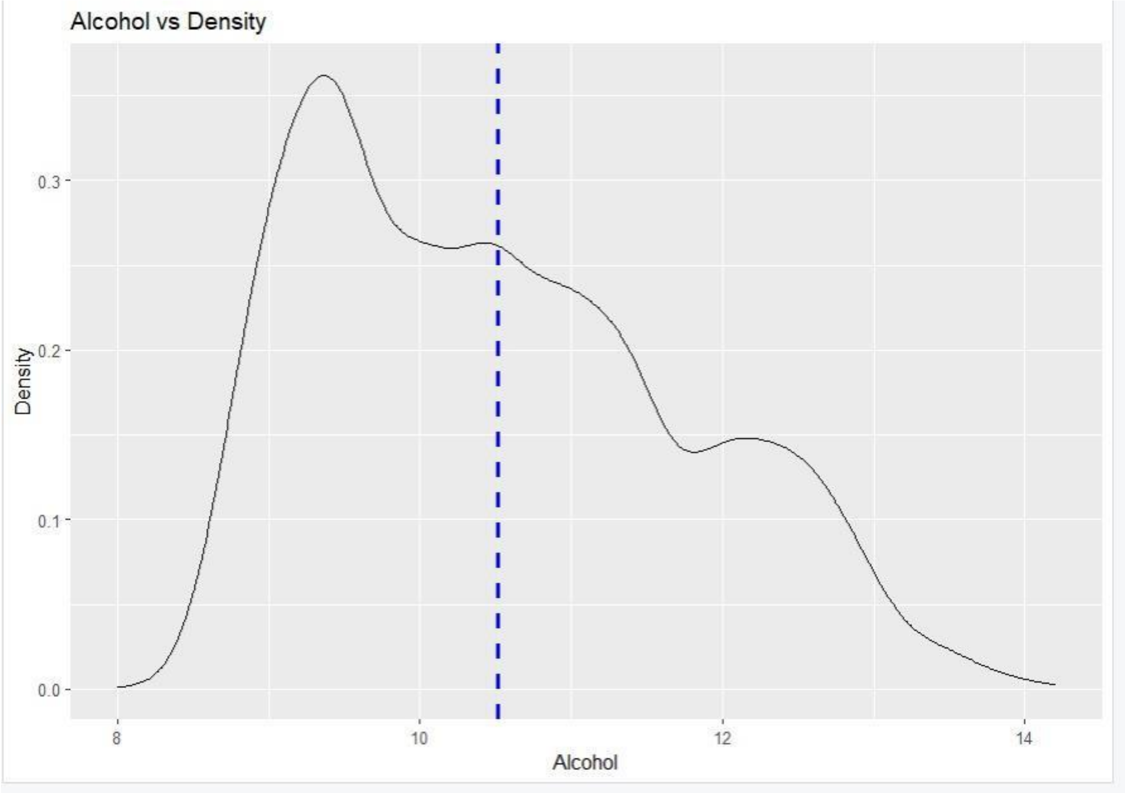

Suplur Dioxide vs Density
Density
0.025
0.020
0.015
0.010
0.005
0.000
0
100
200
300
Suplur Dioxide


Total Sulphur Dioxide vs Density
0.0100
0.0075
0.0050
0.0025
0.0000
Density
0
100
200
300
400
Total Sulphur Dioxide

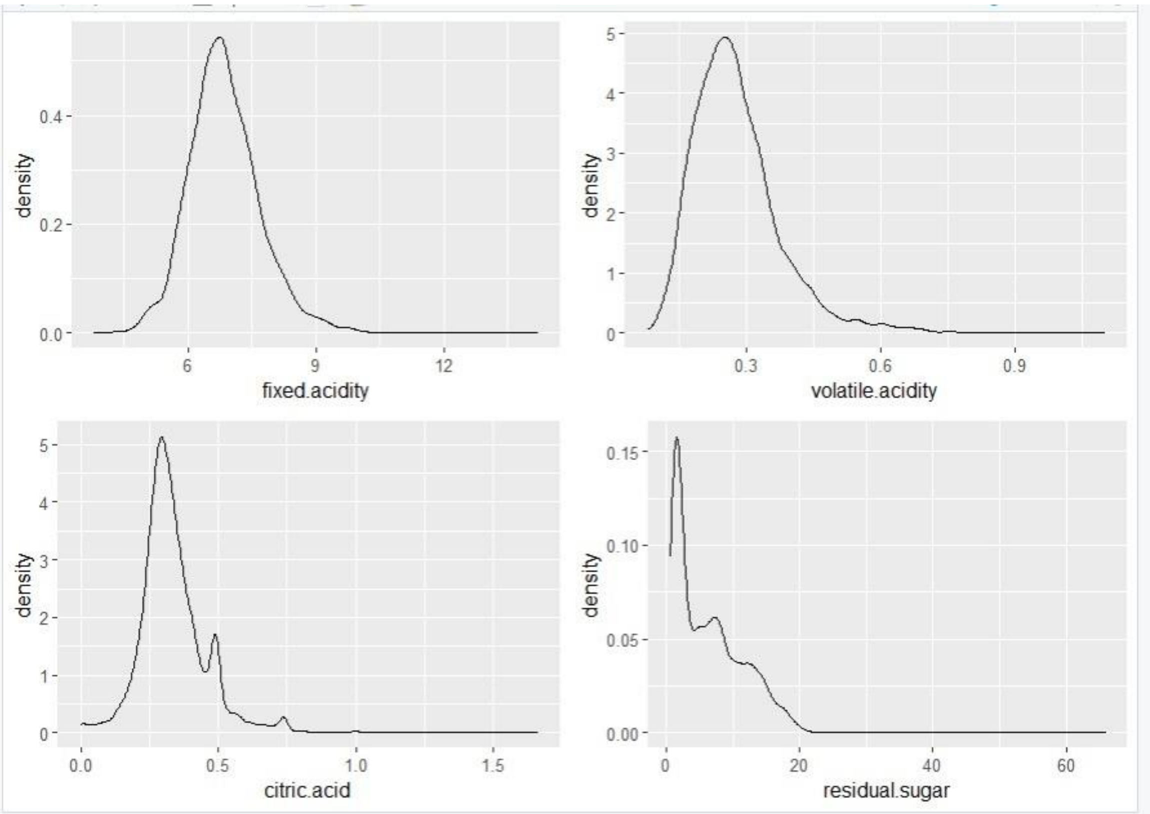

Density vs Density
120
80
40
0
Density
0.99
1.00
1.01
1.02
1.03
1.04
Density

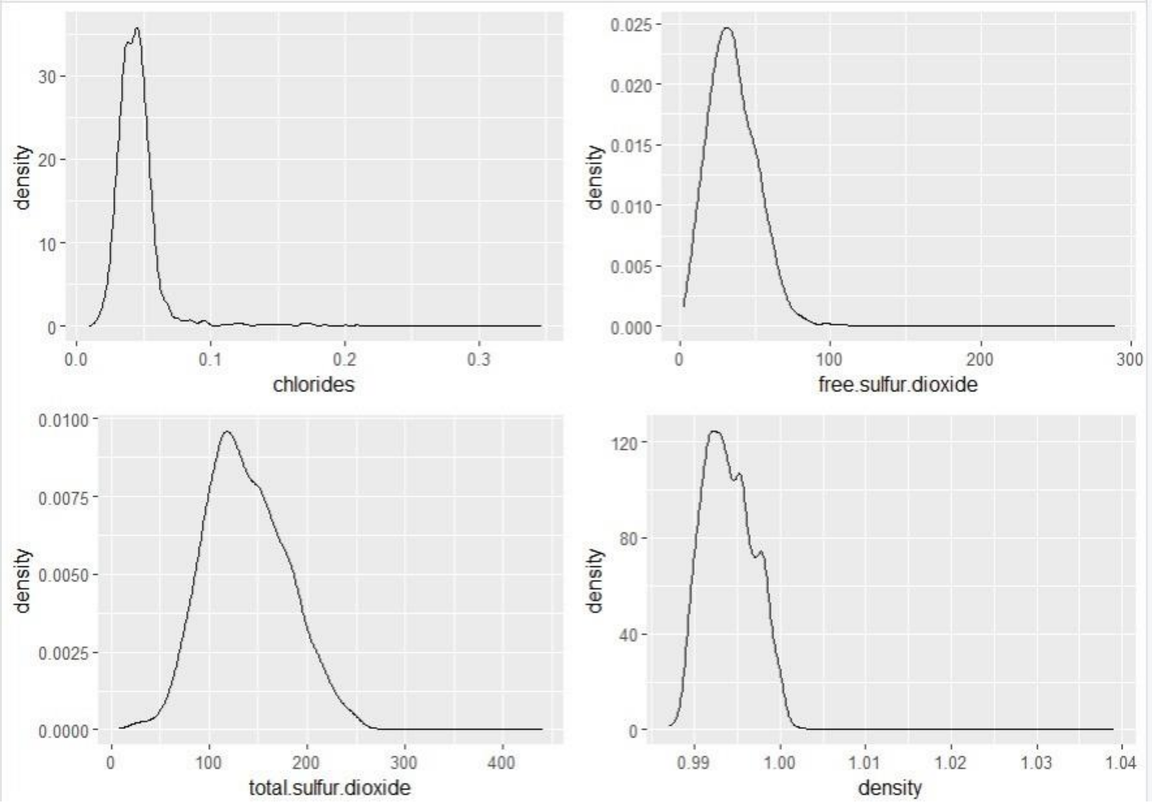

pH vs Density
Density
3
2
1
0
2.7
3.0
3.3
3.6
pH

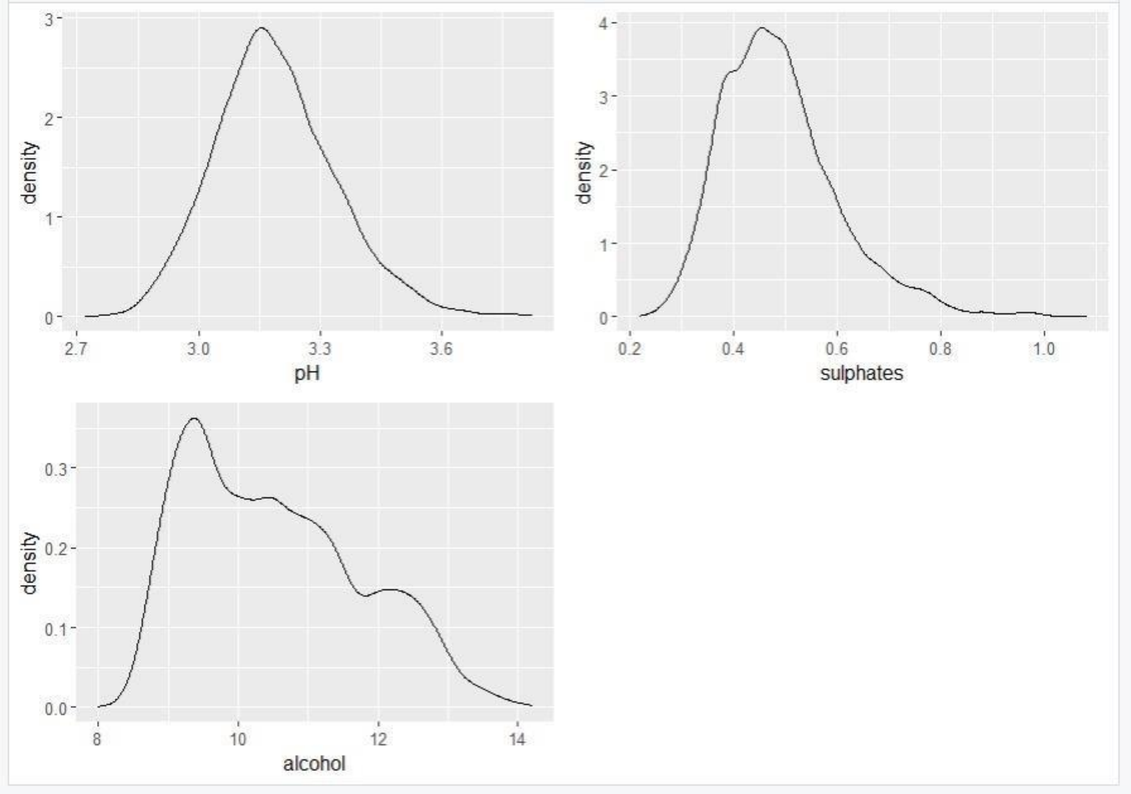

Sulphates vs Density
Density
4
3
2
1
0
0.2
0.4
0.6
0.8
1.0
Sulphates



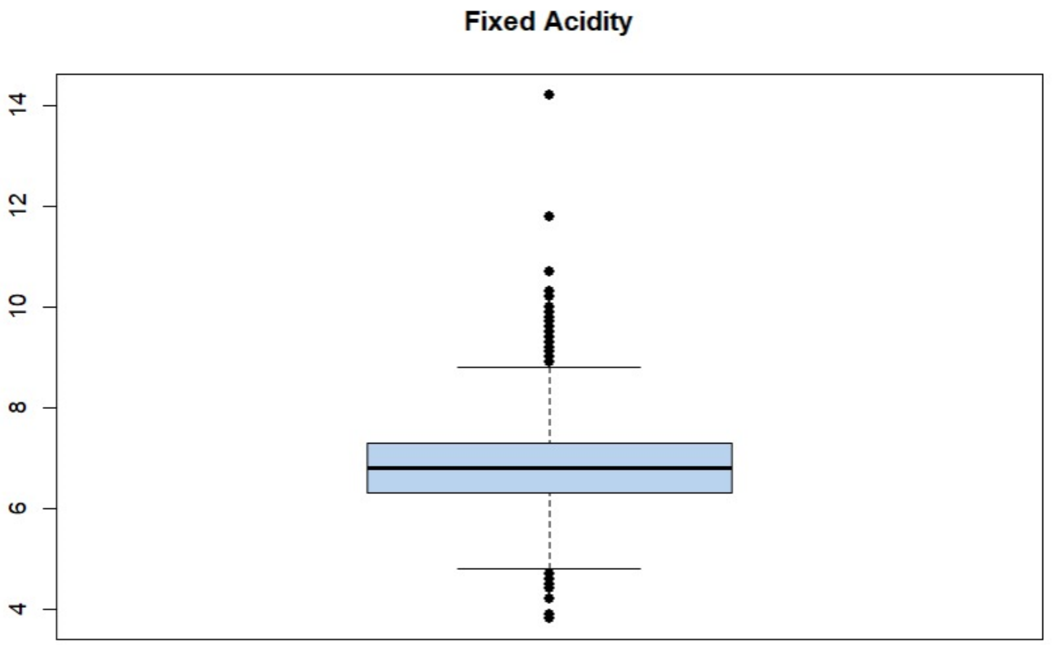
It can be seen that most red wines' pH levels are always between 3–4 and chlorides — the amount of salt is most prevalent at level 0.1. After analysing the density plots, we plotted the interaction between our numeric variables of interest and our dependent variable of quality.

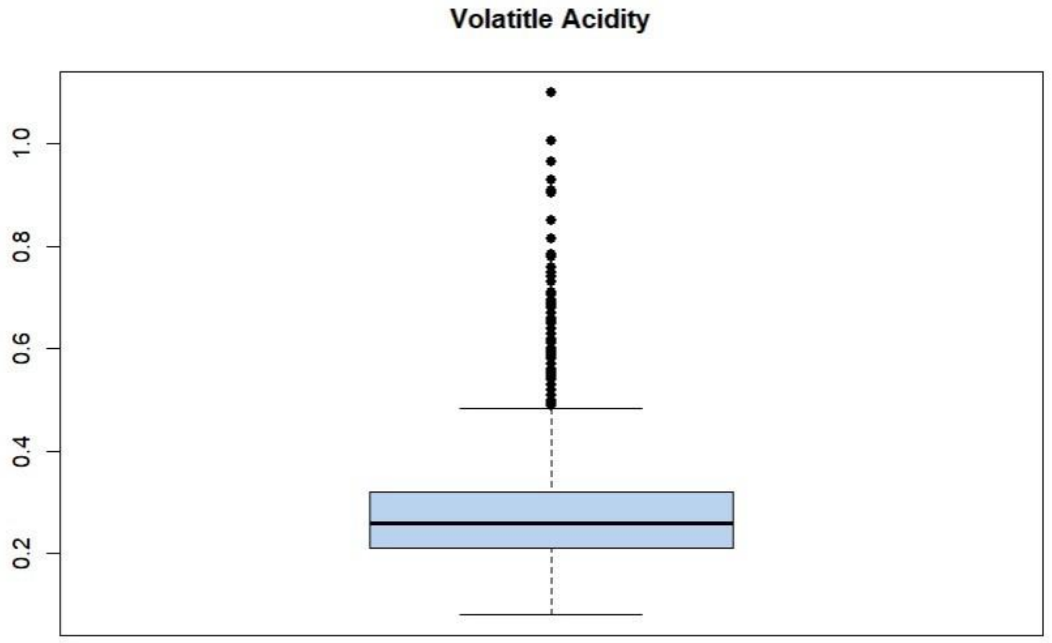

density
0.0
0.2
0.4
6
9
12
fixed.acidity
density
0
1
2
3
4
5
0.3
0.6
0.9
volatile.acidity
density
0
1
2
3
4
5
0.0
0.5
1.0
1.5
citric.acid
density
0.00
0.05
0.10
0.15
0
20
40
60
residual.sugar

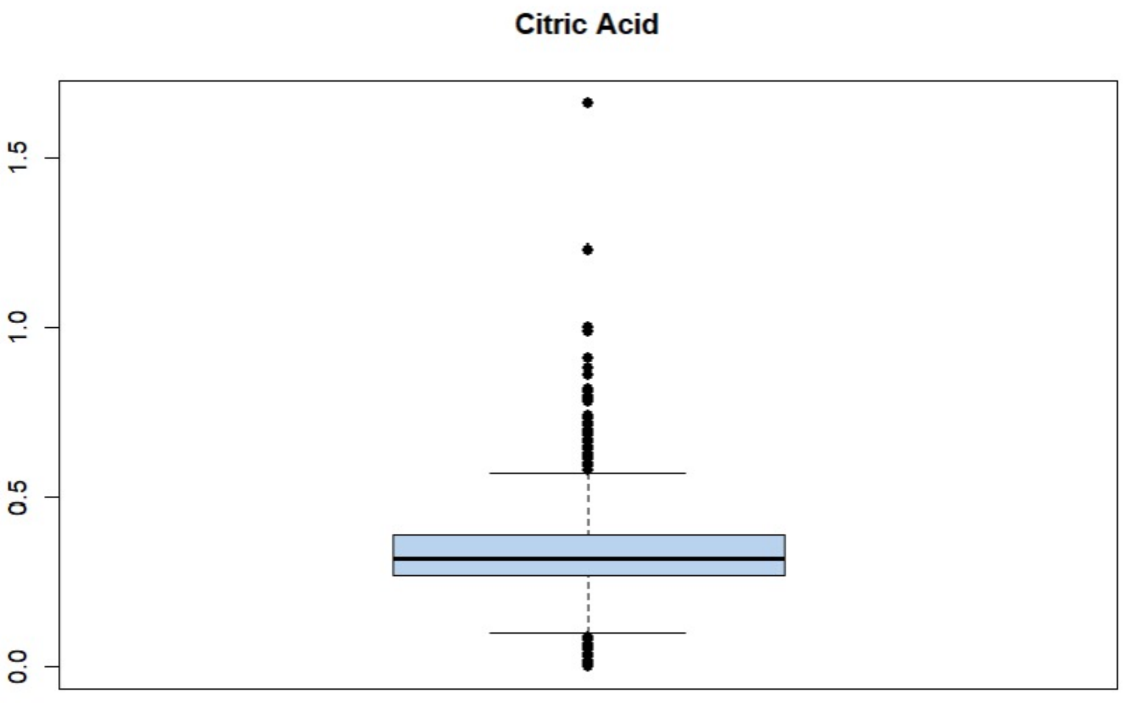

density
0
10
20
30
0.0
0.1
0.2
0.3
chlorides
density
0.000
0.005
0.010
0.015
0.020
0.025
0
100
200
300
free.sulfur.dioxide
density
0.0000
0.0025
0.0050
0.0075
0.0100
0
100
200
300
400
total.sulfur.dioxide
density
0
40
80
120
0.99
1.00
1.01
1.02
1.03
1.04
density

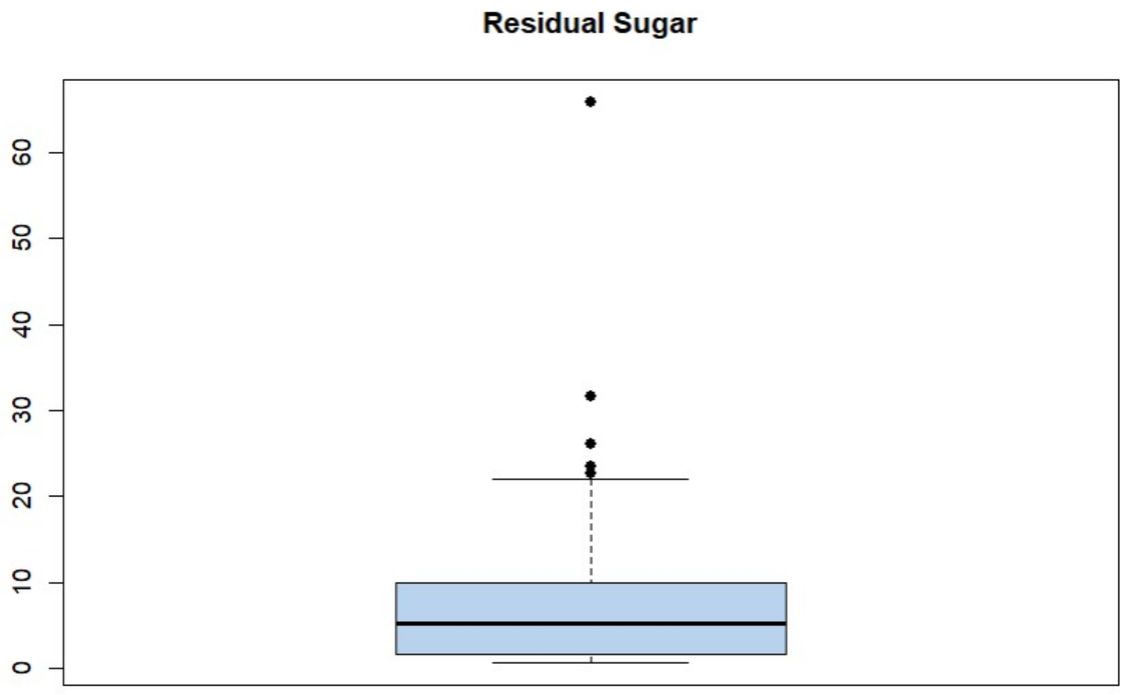

density
3
2
1
0
2.7
3.0
3.3
3.6
pH
density
4
3
2
1
0
0.2
0.4
0.6
0.8
1.0
sulphates
density
0.3
0.2
0.1
0.0
8
10
12
14
alcohol

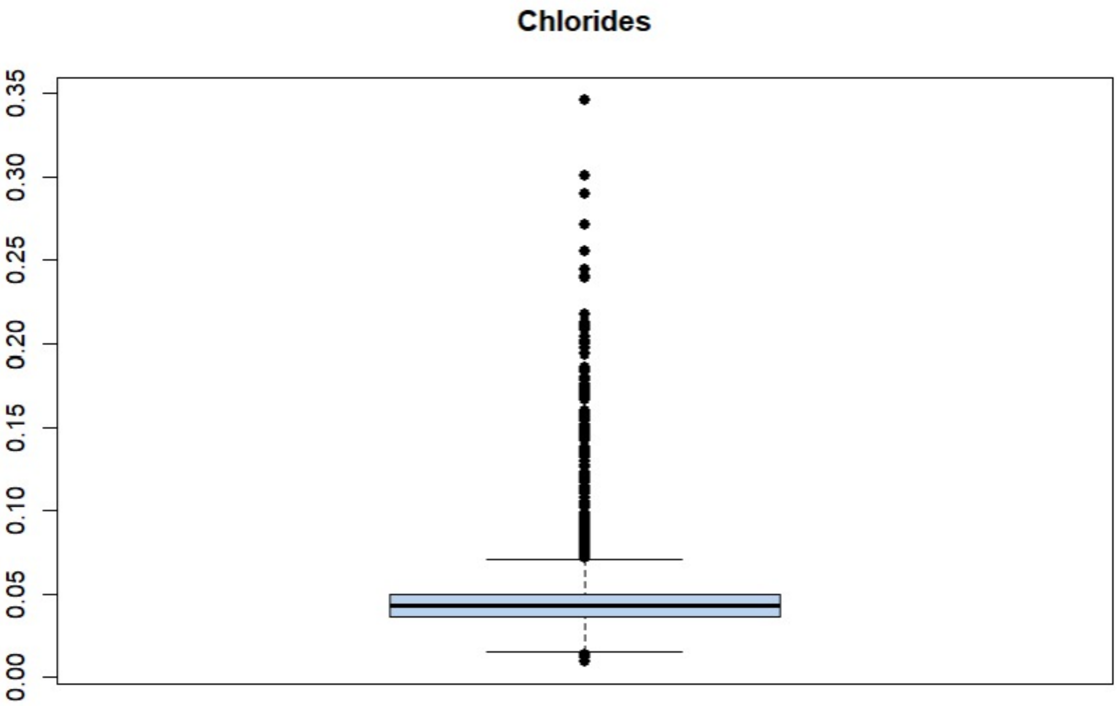
Now we plot Box Plot for each variable to get the Summary of each factor to predict and analyse the quality of wine. This will help us to predict each factor's relation to alter the quality and taste wine.

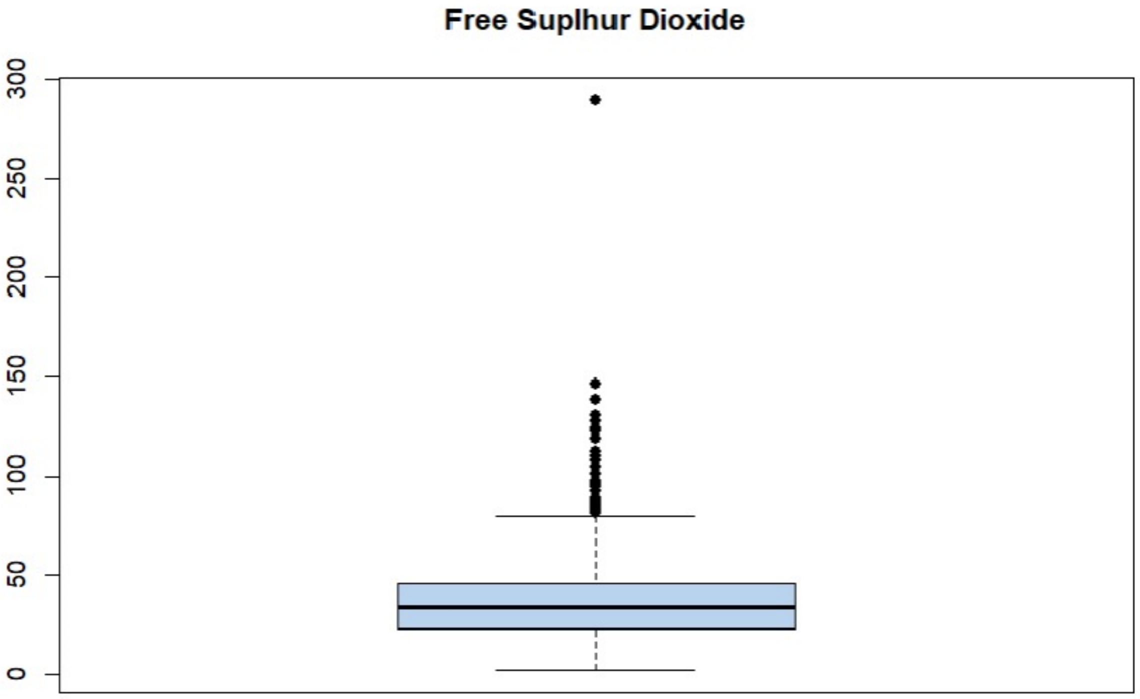


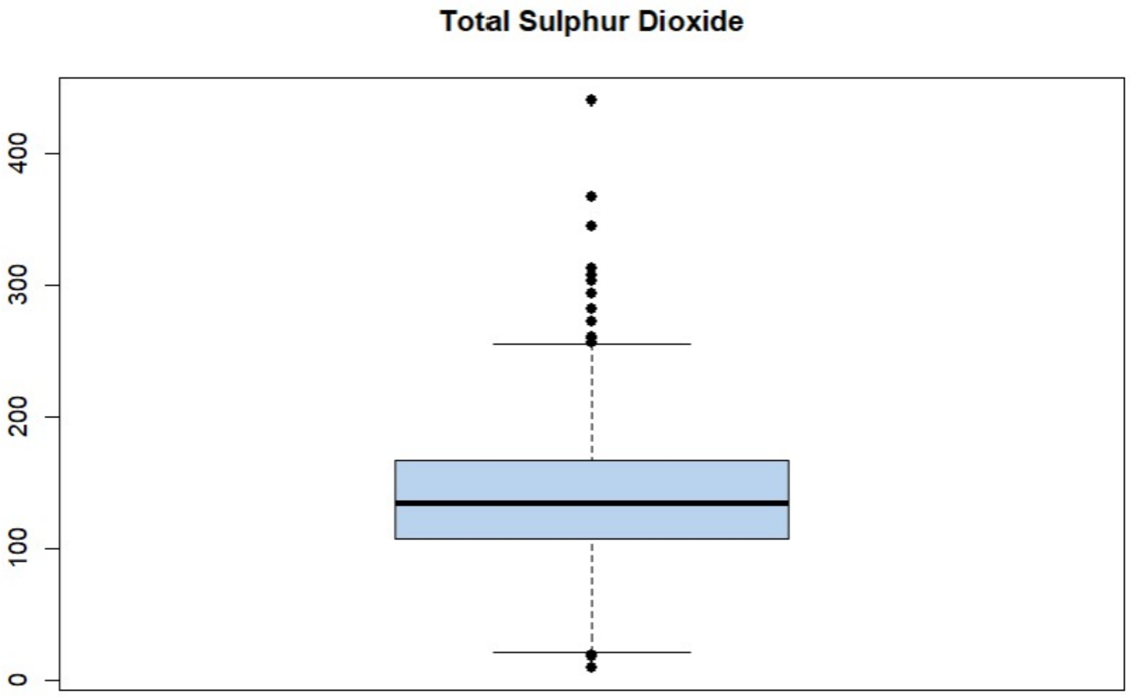


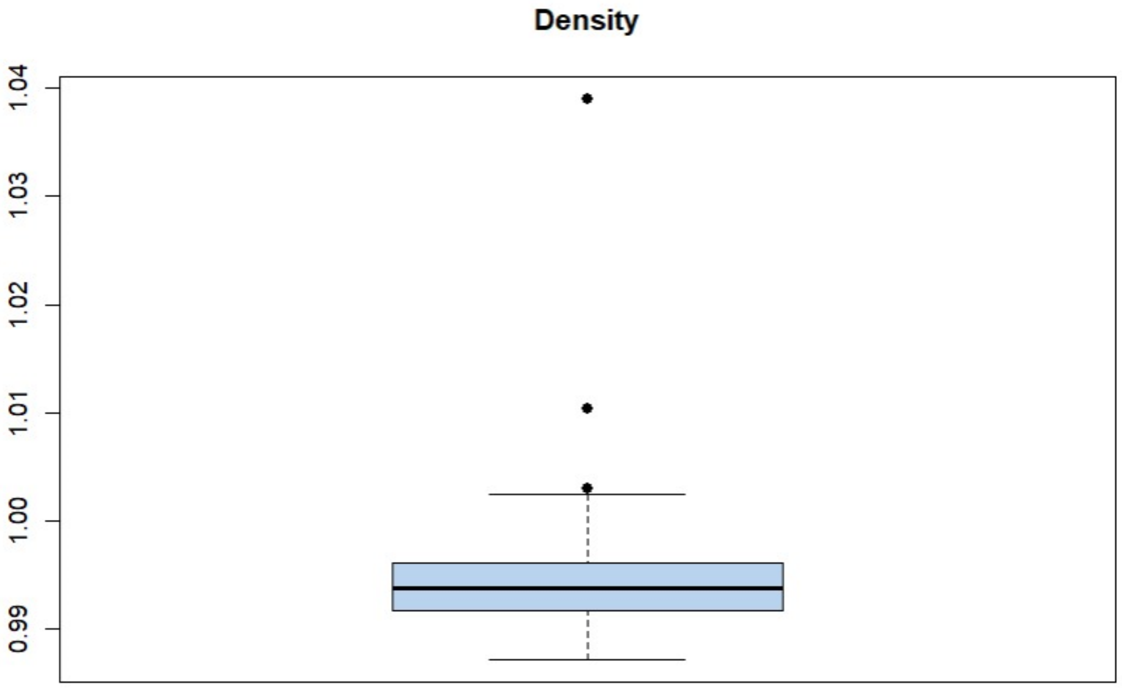

Citric Acid
0.0
0.5
1.0
1.5

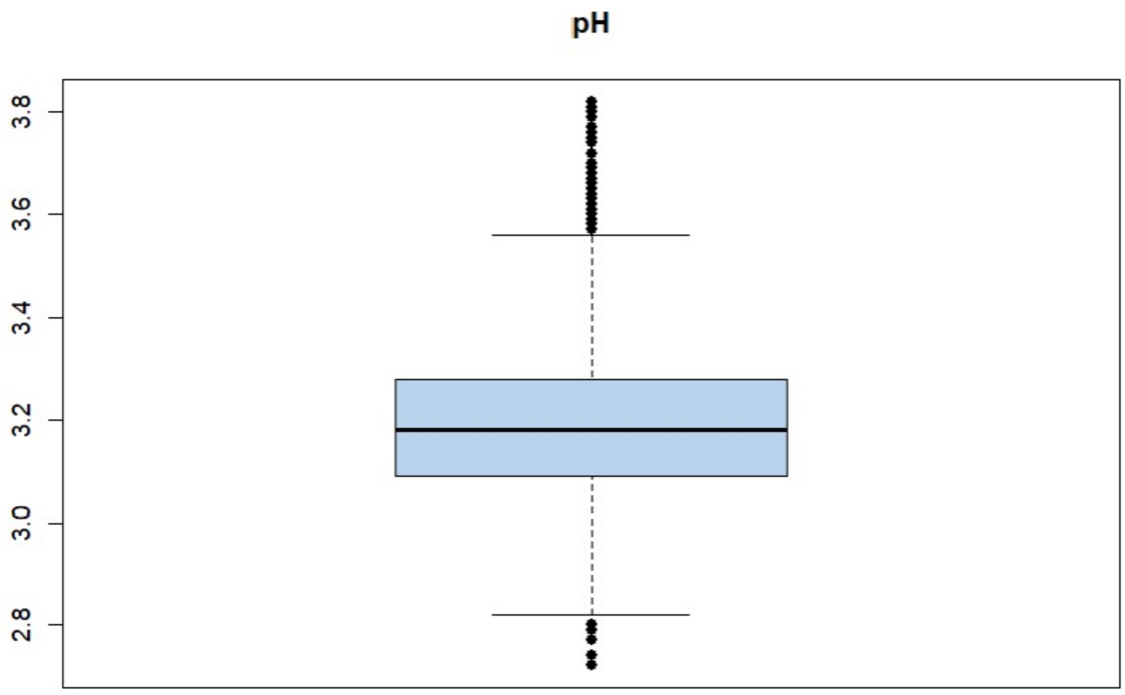

Residual Sugar
0
10
20
30
40
50
60

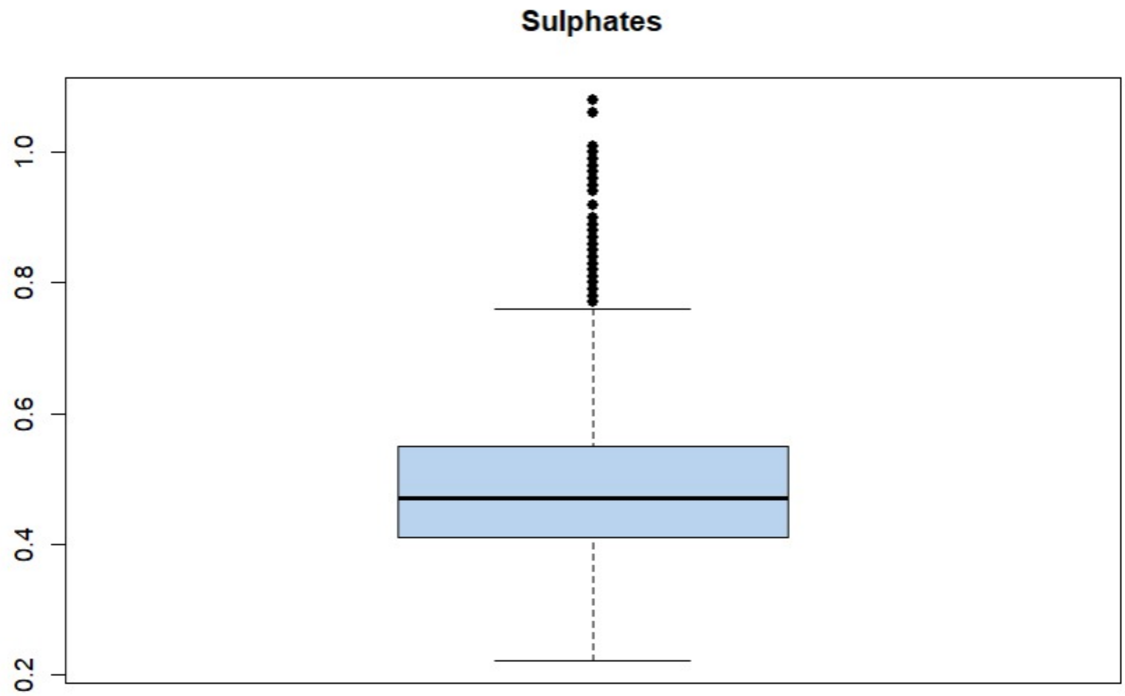

Chlorides
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35

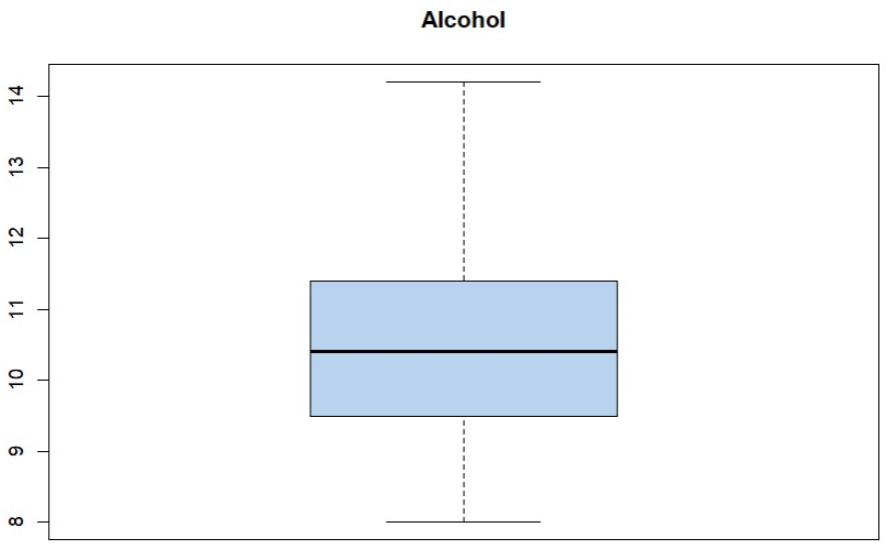

Free Suplhur Dioxide
0
50
100
150
200
250
300


Total Sulphur Dioxide
0
100
200
300
400


Density
0.99
1.00
1.01
1.02
1.03
1.04

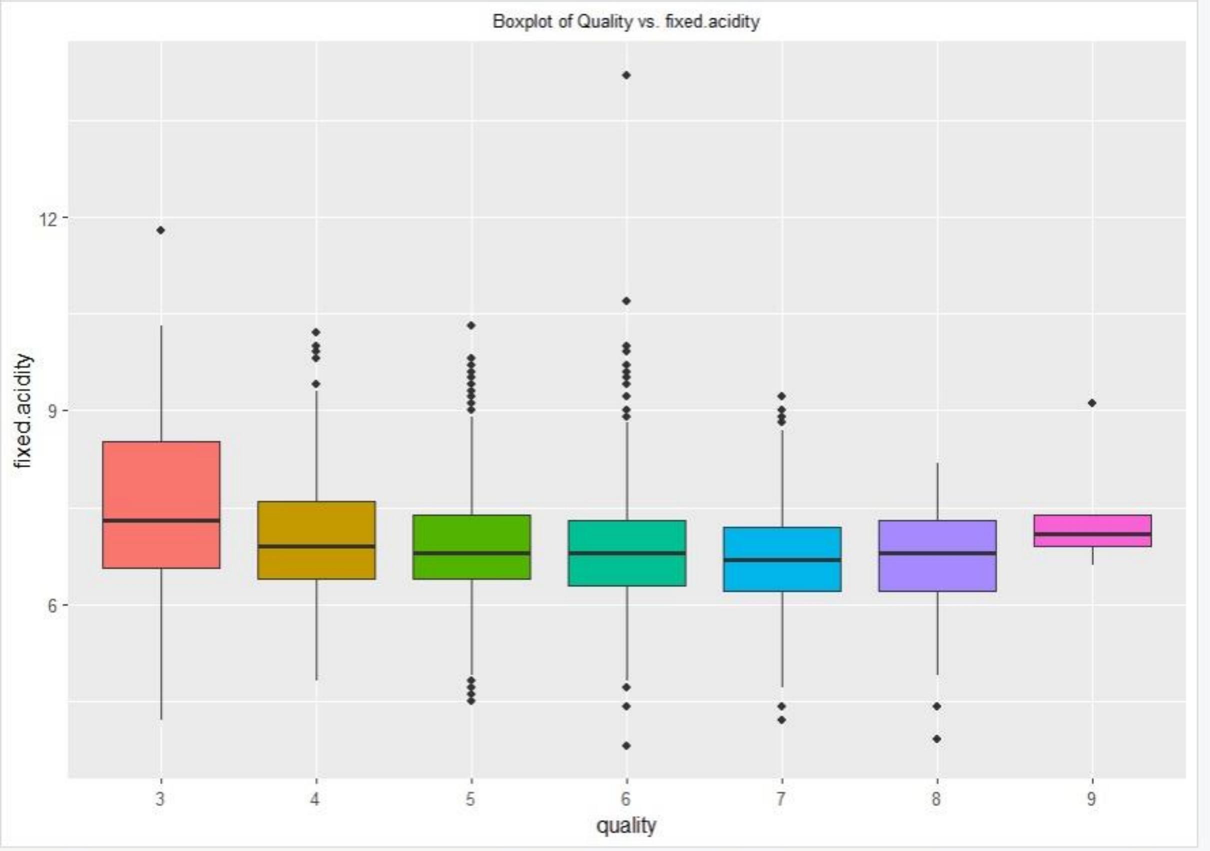

pH
3.8
3.6
3.4
3.2
3.0
2.8

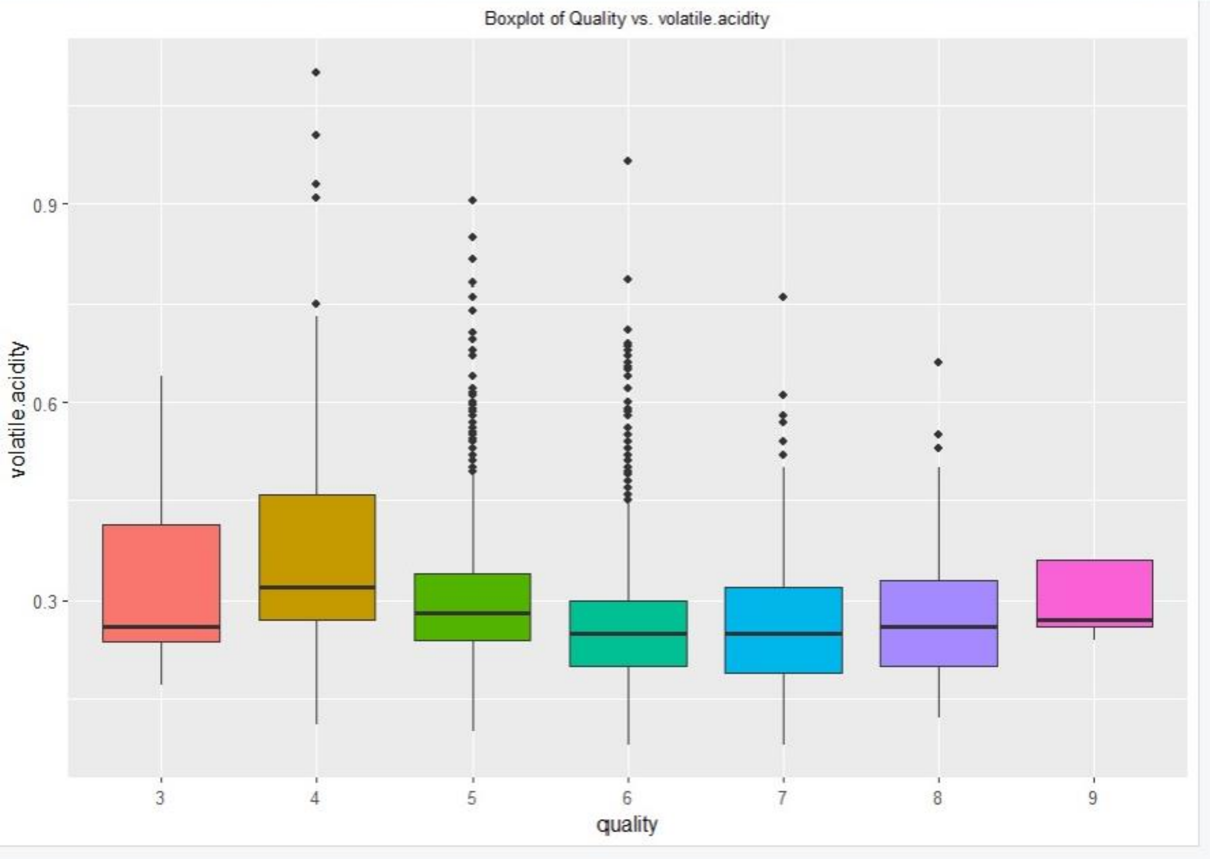

Sulphates
1.0
0.8
0.6
0.4
0.2

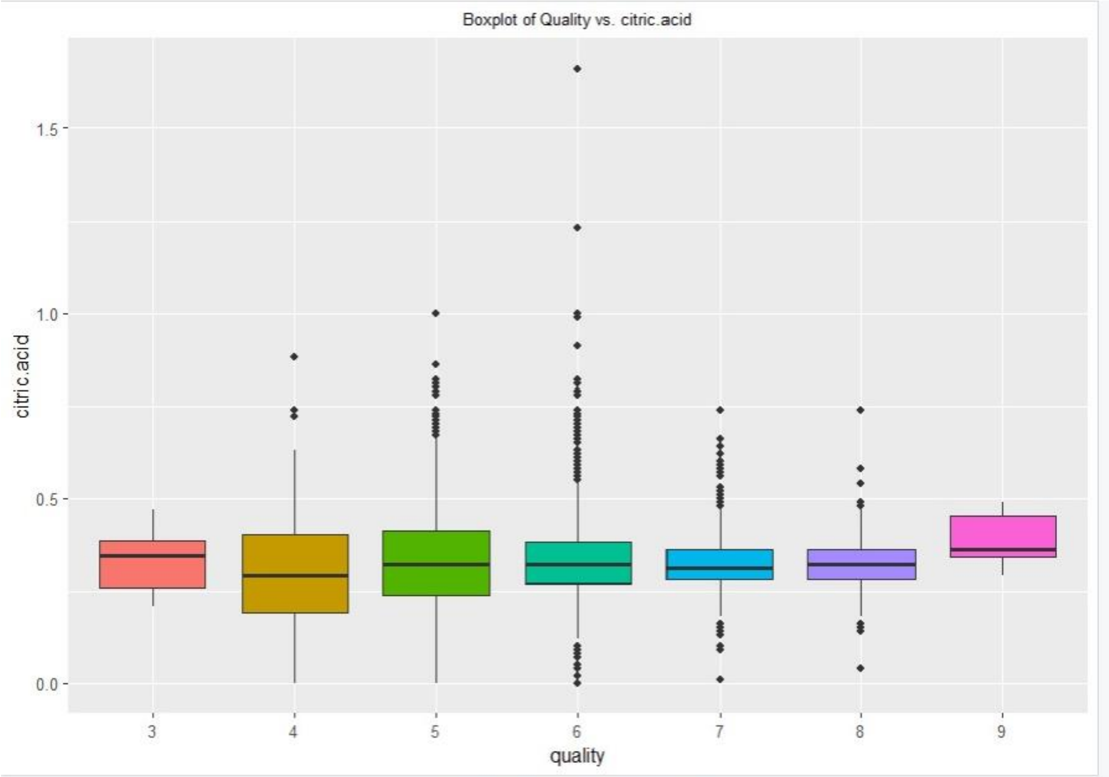


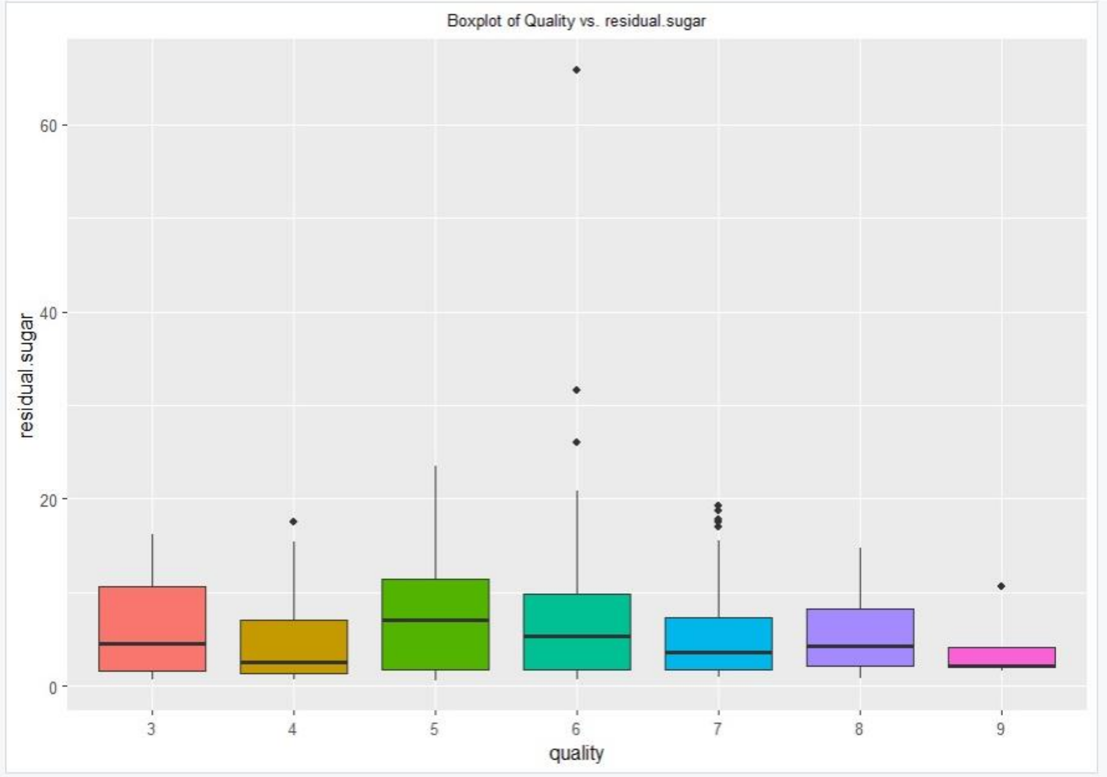
From these box plots of each factor gives us the summary of its distributions and importance to predict the quality of wine.

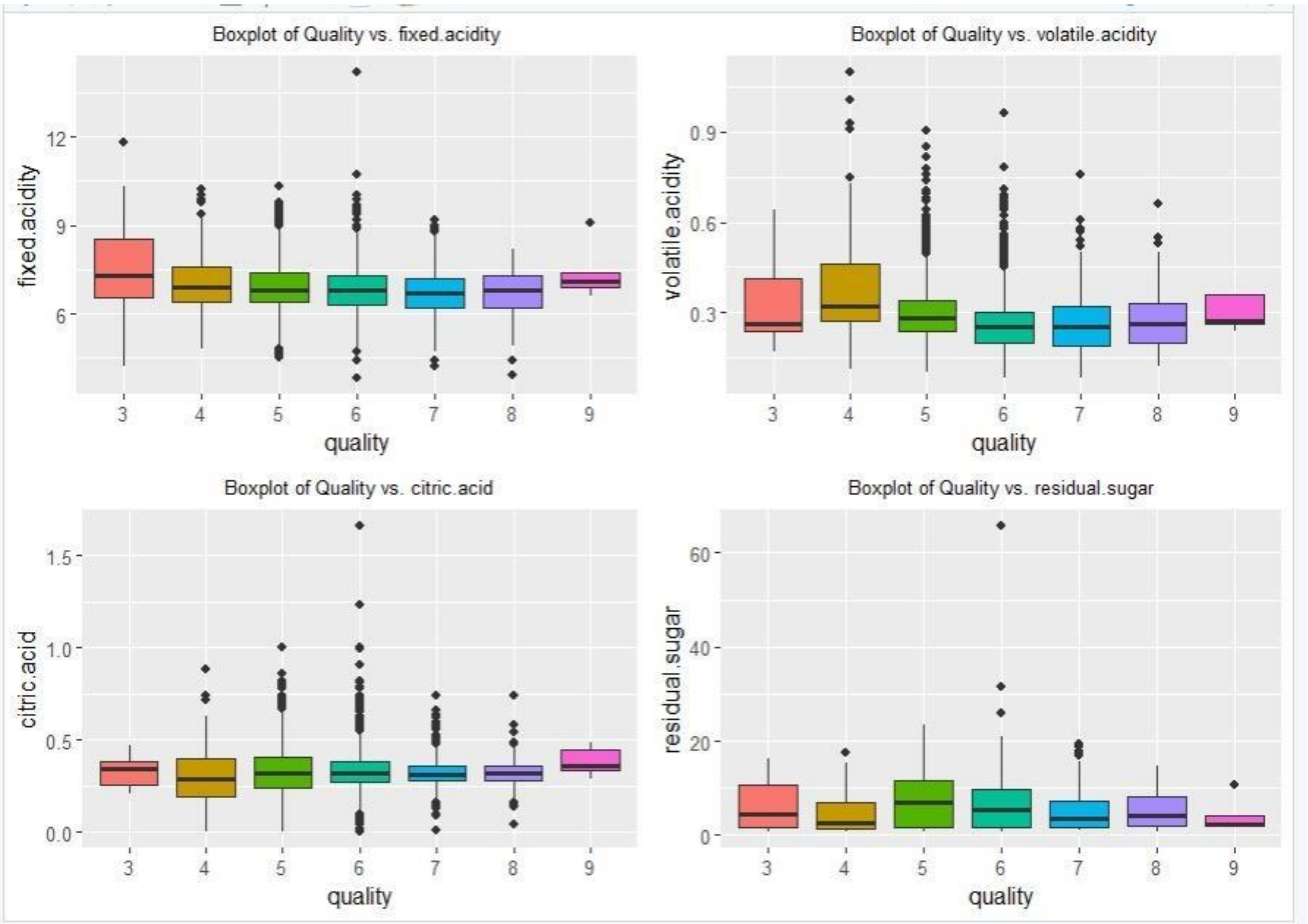
Now we inspect quality against variables of interest. We plot this in the required range of quality from 3-9 for each variable and fact

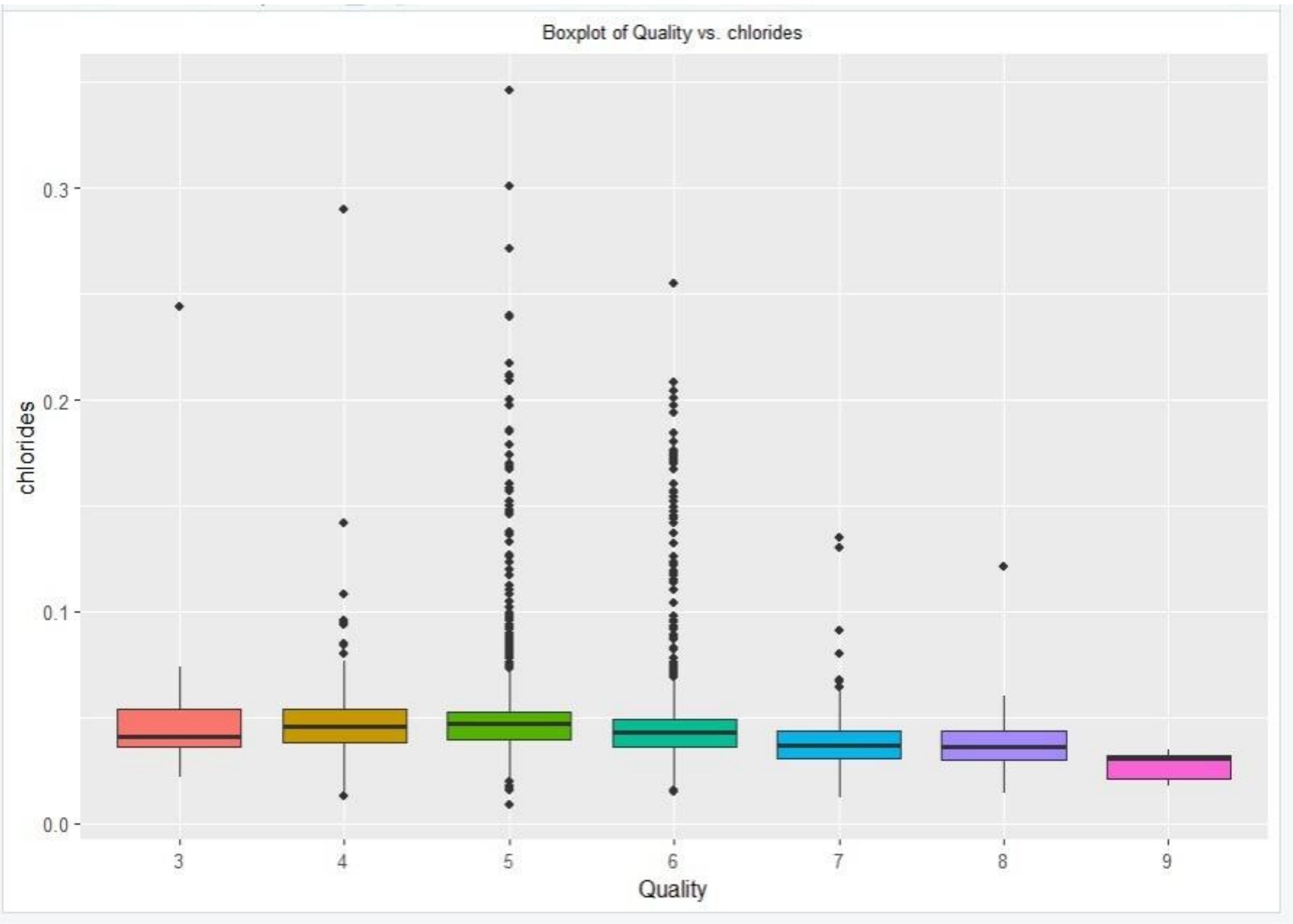


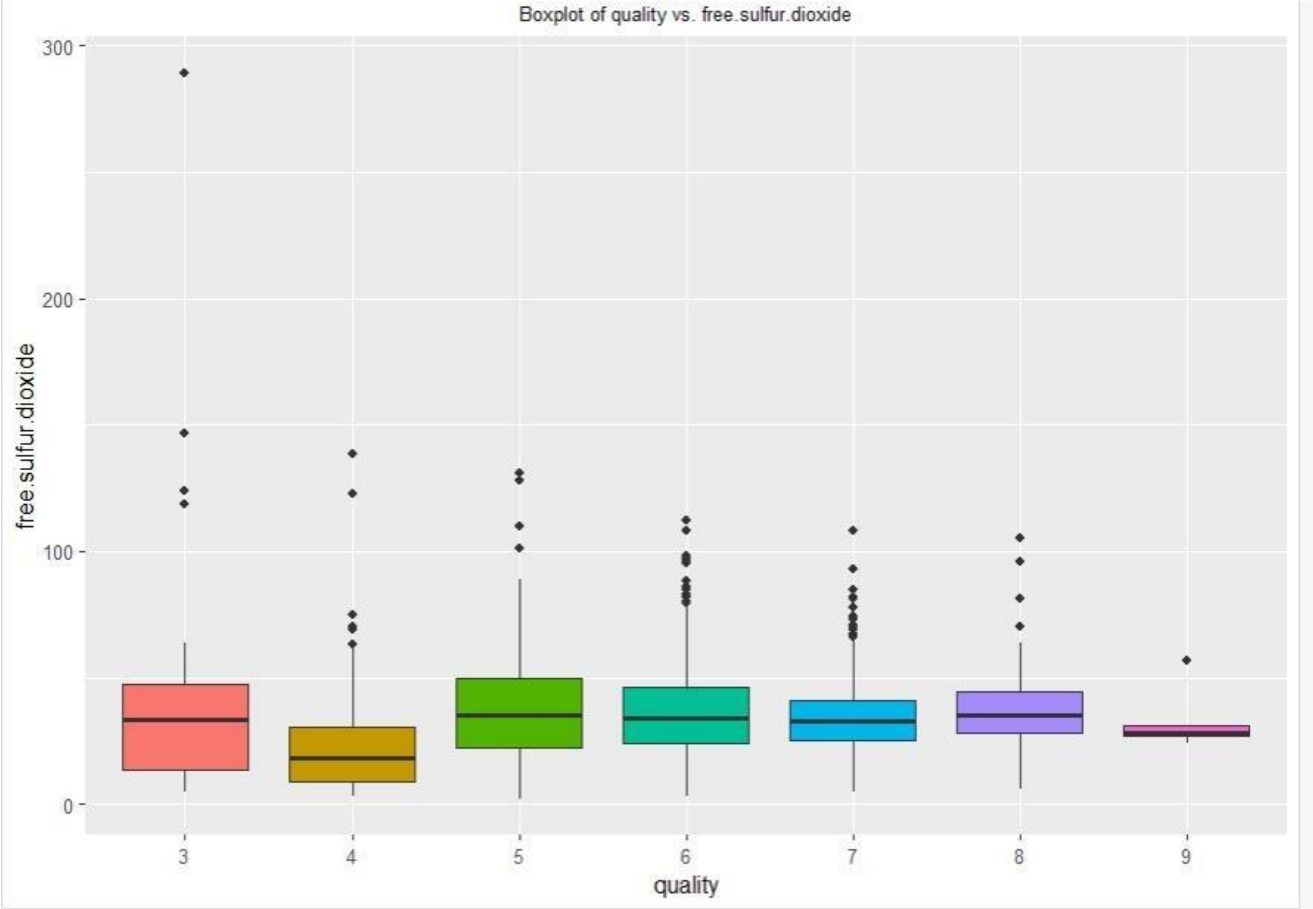


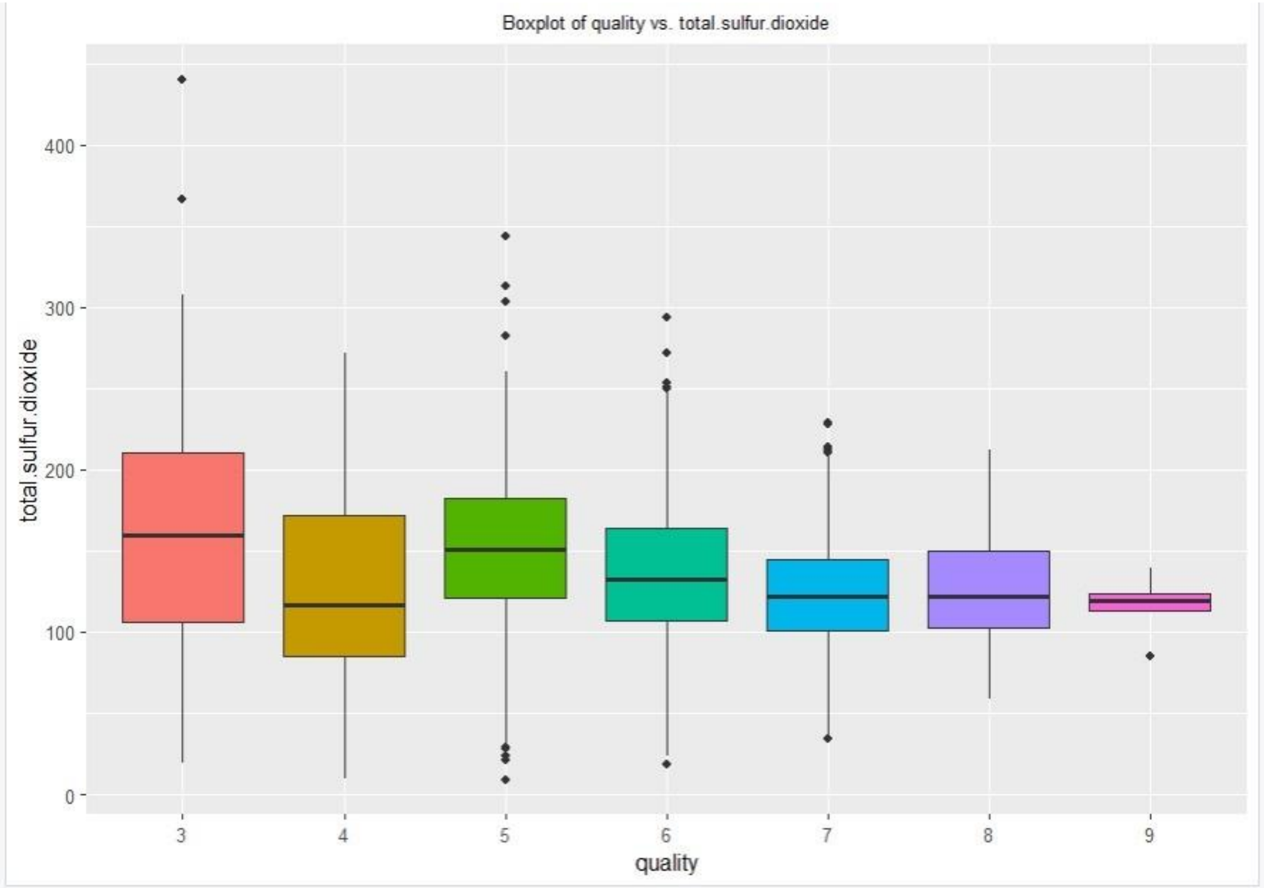

Boxplot of Quality vs. citric.acid
citric.acid
1.5
1.0
0.5
0.0
3
4
5
6
7
8
9
quality

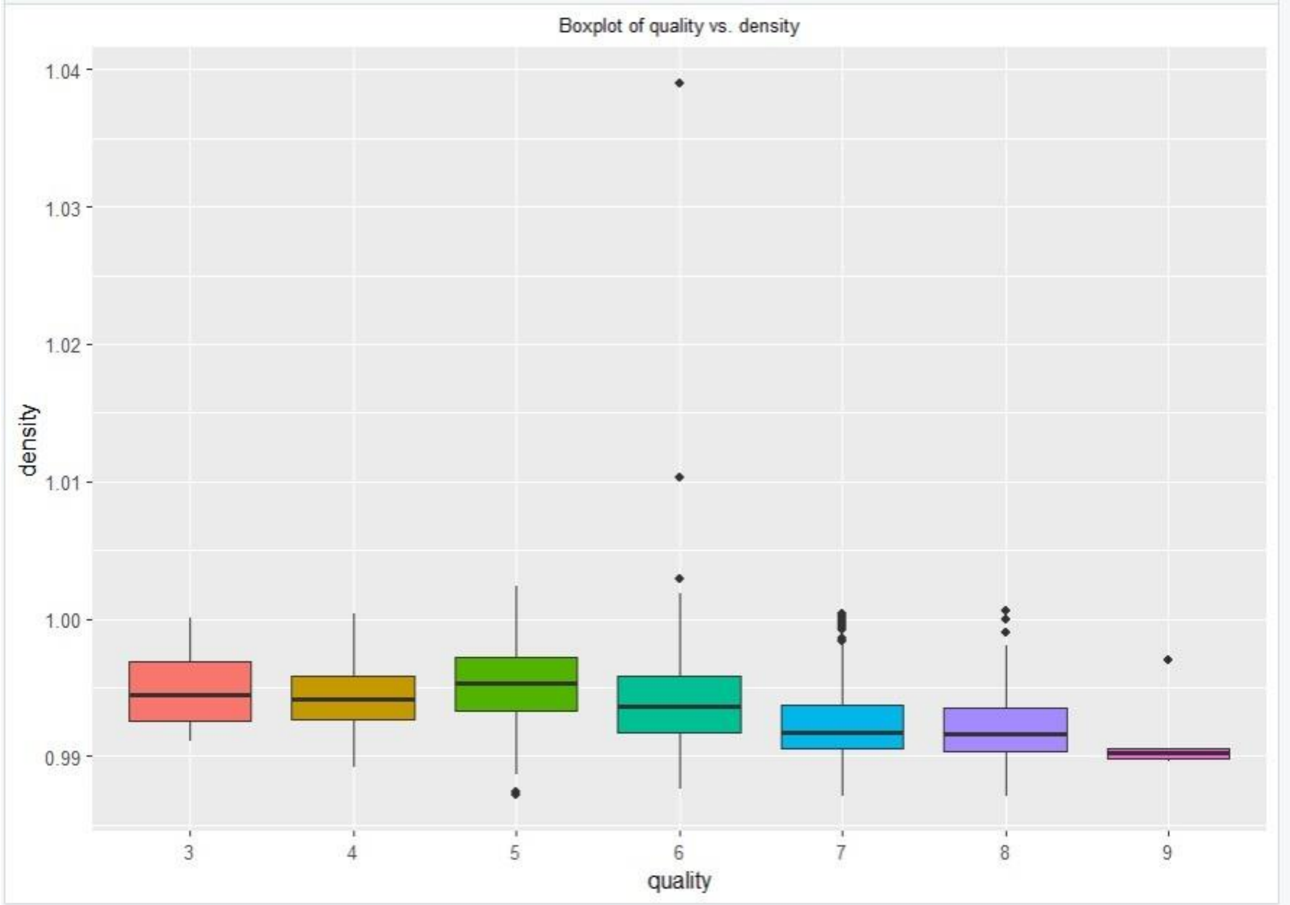

Boxplot of Quality vs. residual.sugar
residual.sugar
60
40
20
0
3
4
5
6
7
8
9
quality

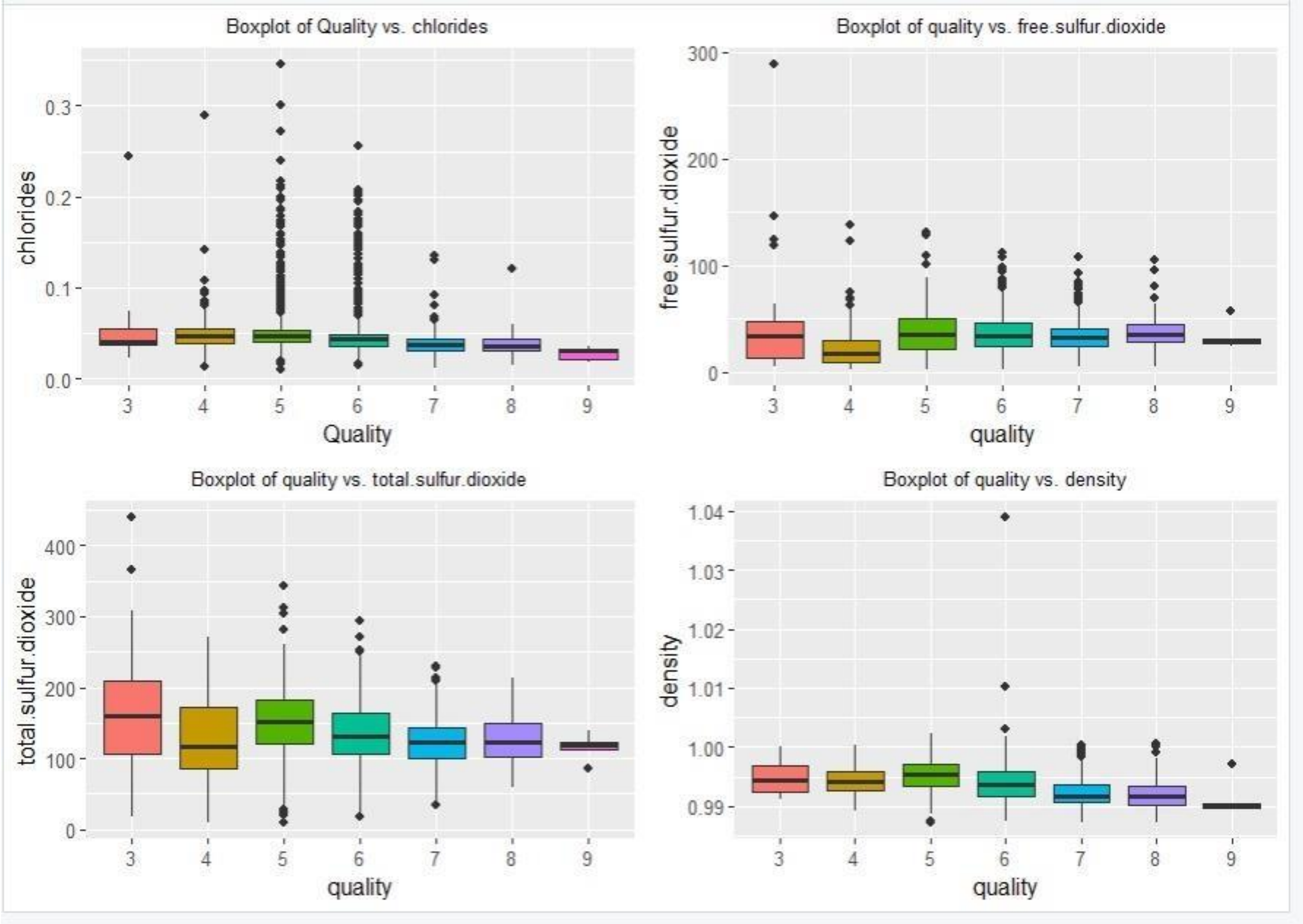

Boxplot of Quality vs. fixed.acidity
fixed.acidity
12
9
6
3
4
5
6
7
8
9
quality
Boxplot of Quality vs. volatile.acidity
volatile.acidity
0.9
0.6
0.3
3
4
5
6
7
8
9
quality
Boxplot of Quality vs. citric.acid
citric.acid
1.5
1.0
0.5
0.0
3
4
5
6
7
8
9
quality
Boxplot of Quality vs. residual.sugar
residual.sugar
60
40
20
0
3
4
5
6
7
8
9
quality

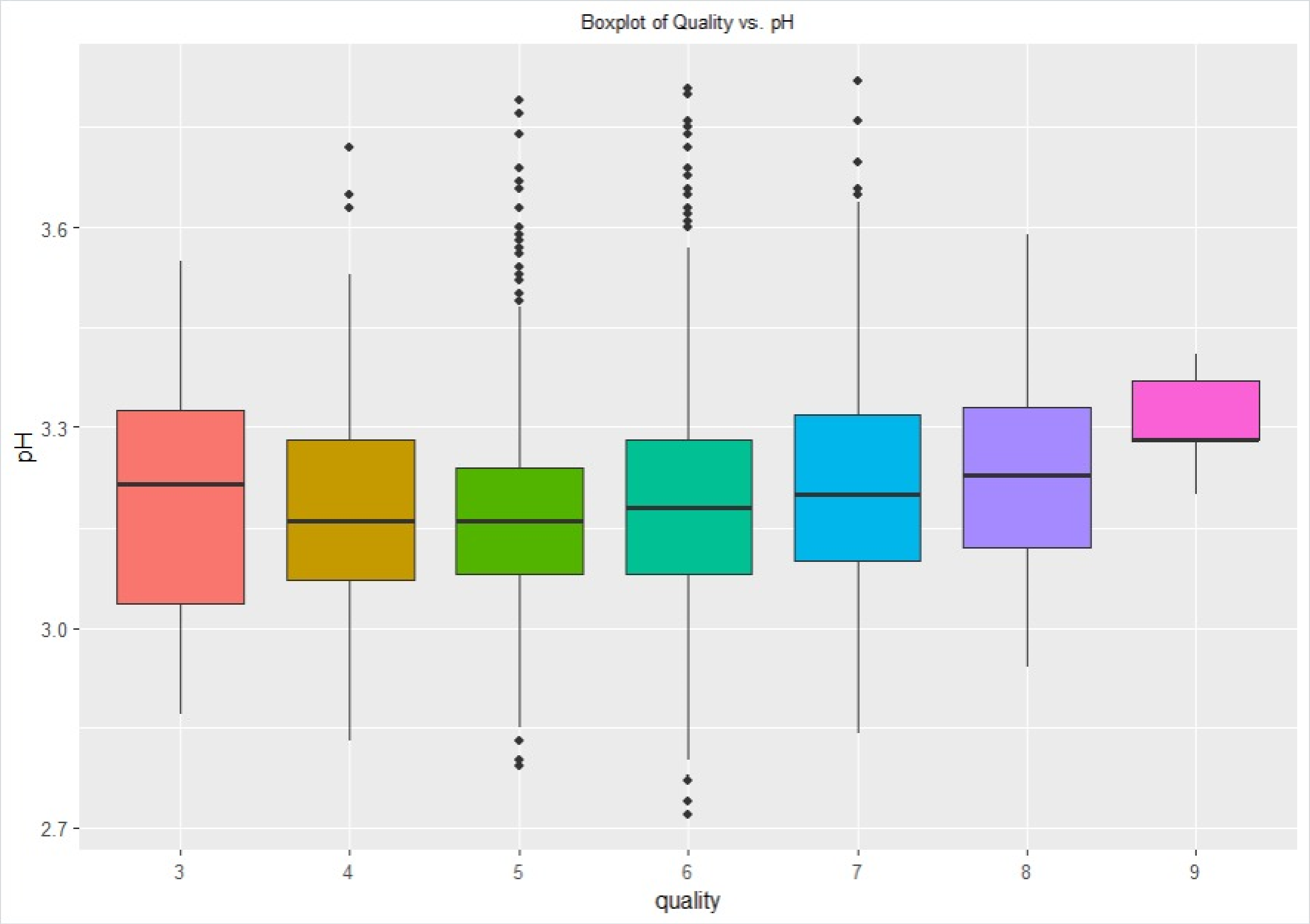

Boxplot of Quality vs. chlorides
chlorides
0.0
0.1
0.2
0.3
3
4
5
6
7
8
9
Quality

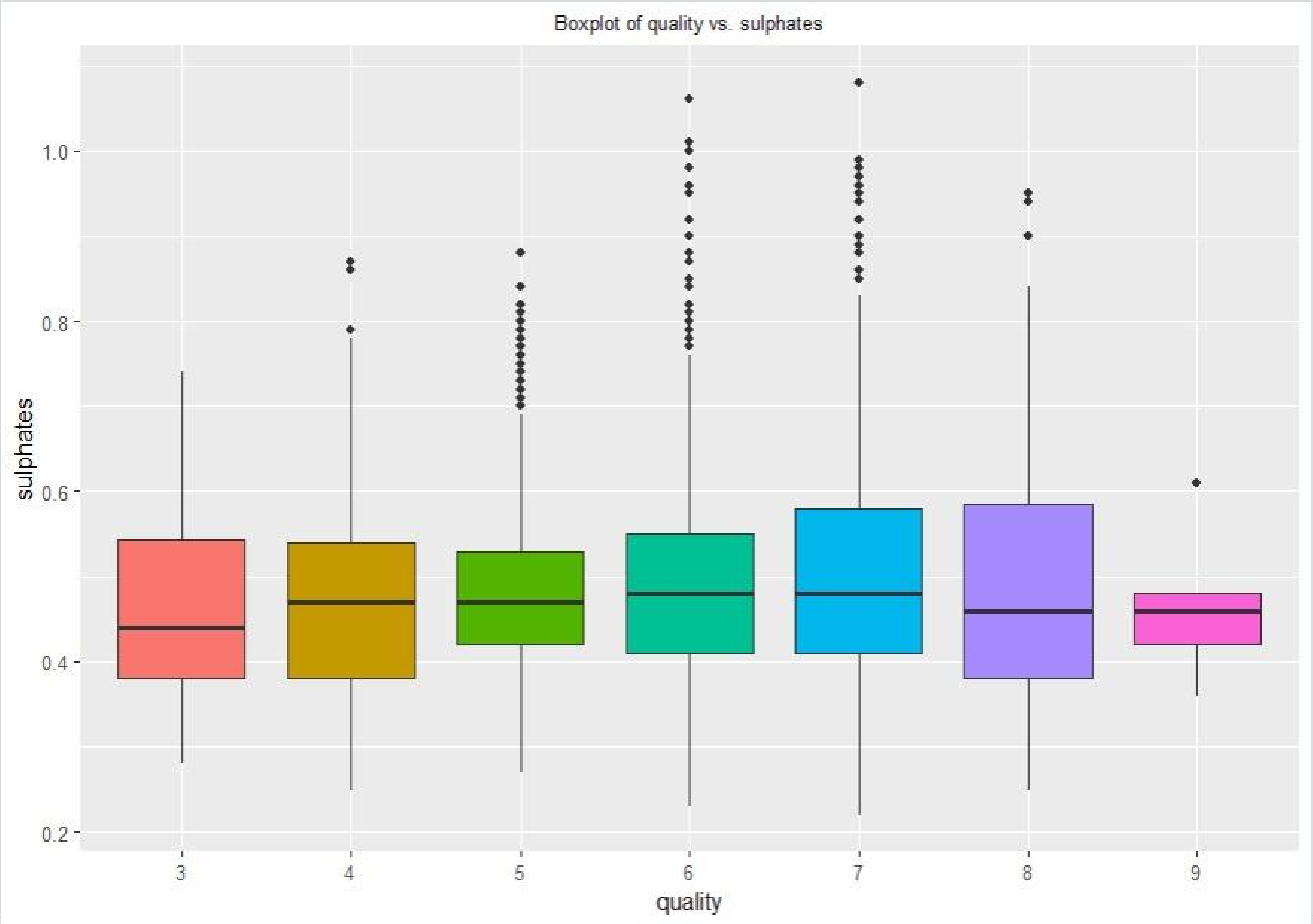

Boxplot of quality vs. free.sulfur.dioxide
free.sulfur.dioxide
0
100
200
300
3
4
5
6
7
8
9
quality

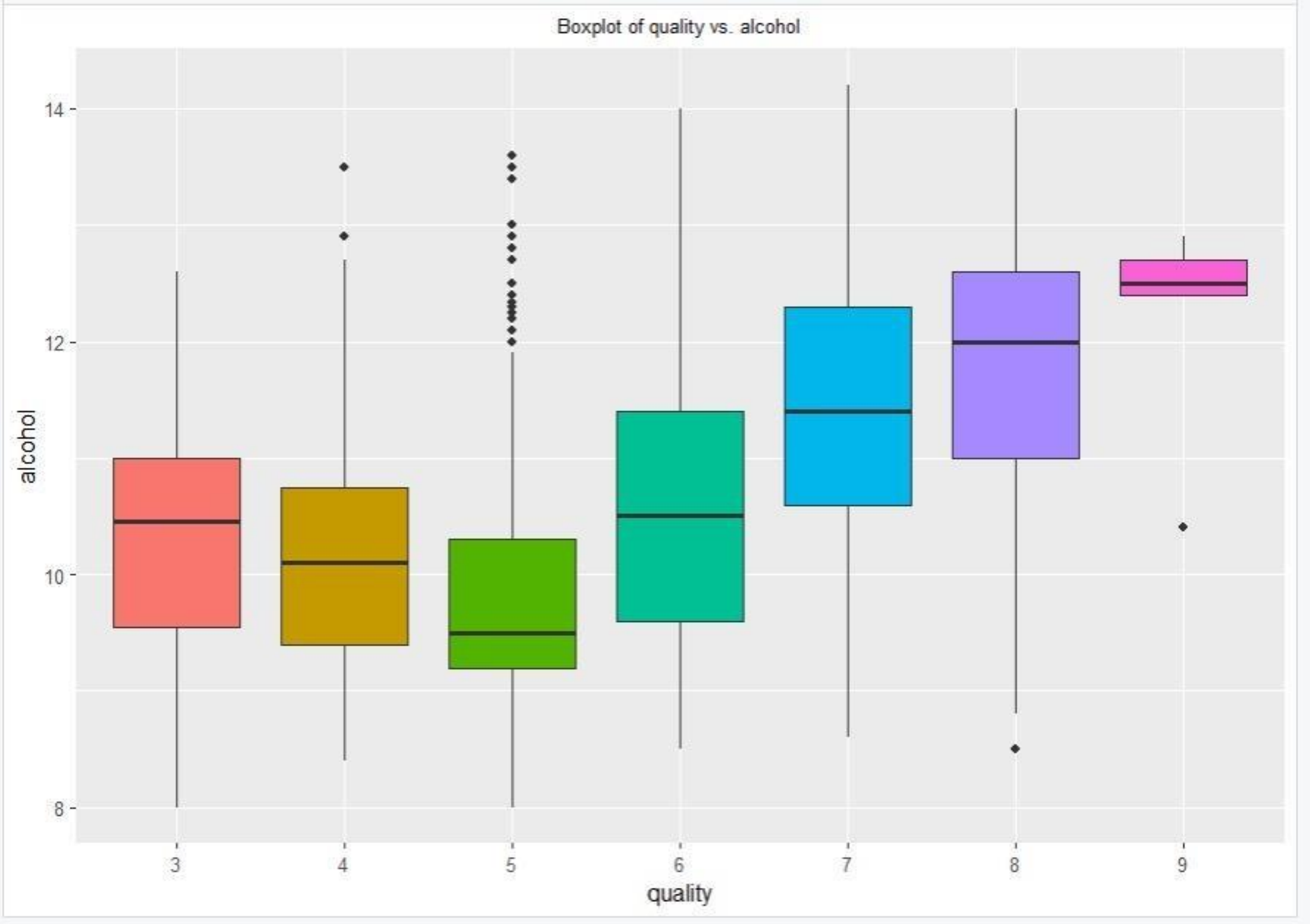

Boxplot of quality vs. total.sulfur.dioxide
total.sulfur.dioxide
400
300
200
100
0
3
4
5
6
7
8
9
quality

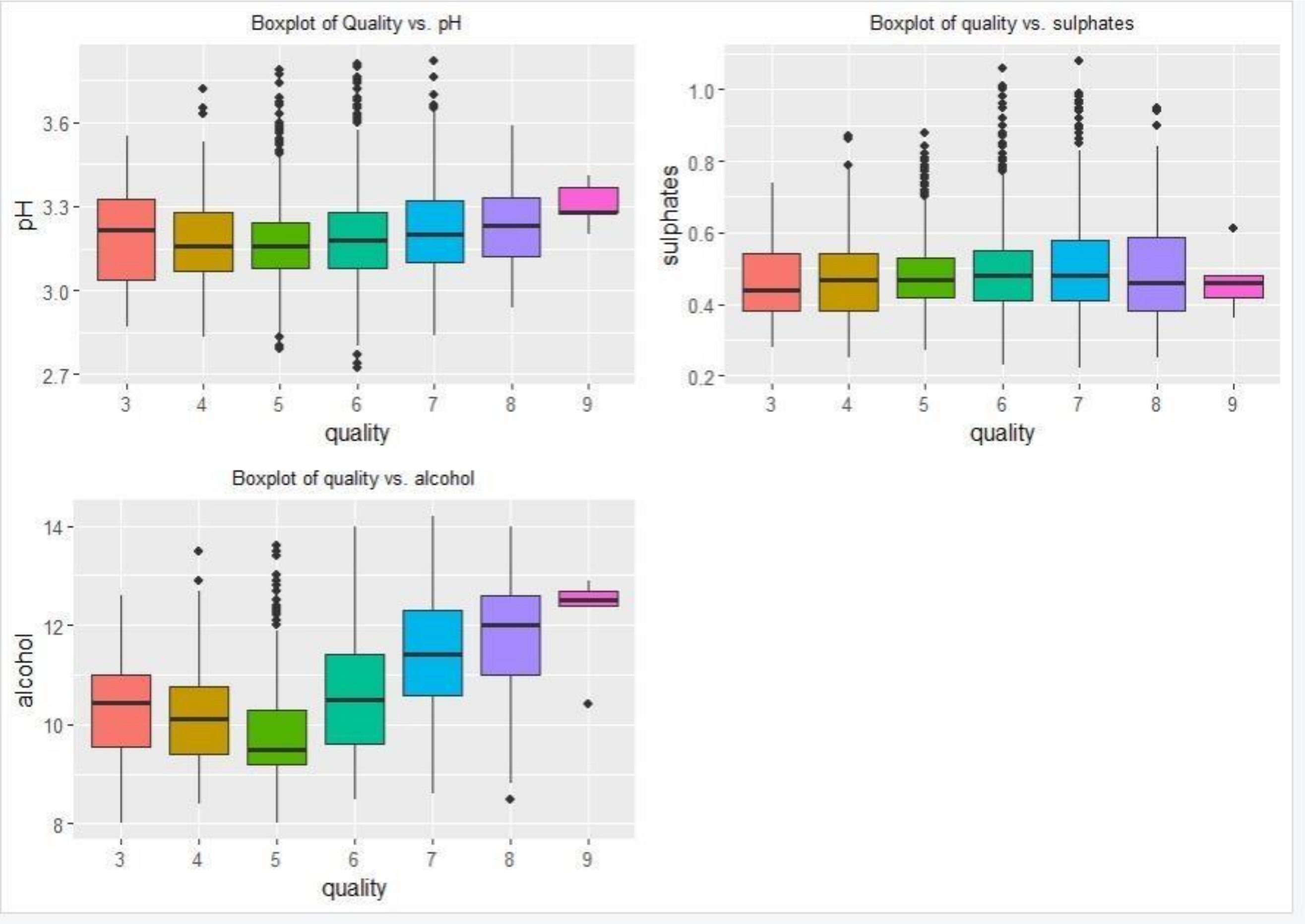

Boxplot of quality vs. density
density
1.04
1.03
1.02
1.01
1.00
0.99
3
4
5
6
7
8
9
quality


Boxplot of Quality vs. chlorides
chlorides
0.3
0.2
0.1
0.0
3
4
5
6
7
8
9
Quality
Boxplot of quality vs. free.sulfur.dioxide
free.sulfur.dioxide
300
200
100
0
3
4
5
6
7
8
9
quality
Boxplot of quality vs. total.sulfur.dioxide
total.sulfur.dioxide
400
300
200
100
0
3
4
5
6
7
8
9
quality
Boxplot of quality vs. density
density
1.04
1.03
1.02
1.01
1.00
0.99
3
4
5
6
7
8
9
quality


Boxplot of Quality vs. pH
pH
3.6
3.3
3.0
2.7
3
4
5
6
7
8
9
quality

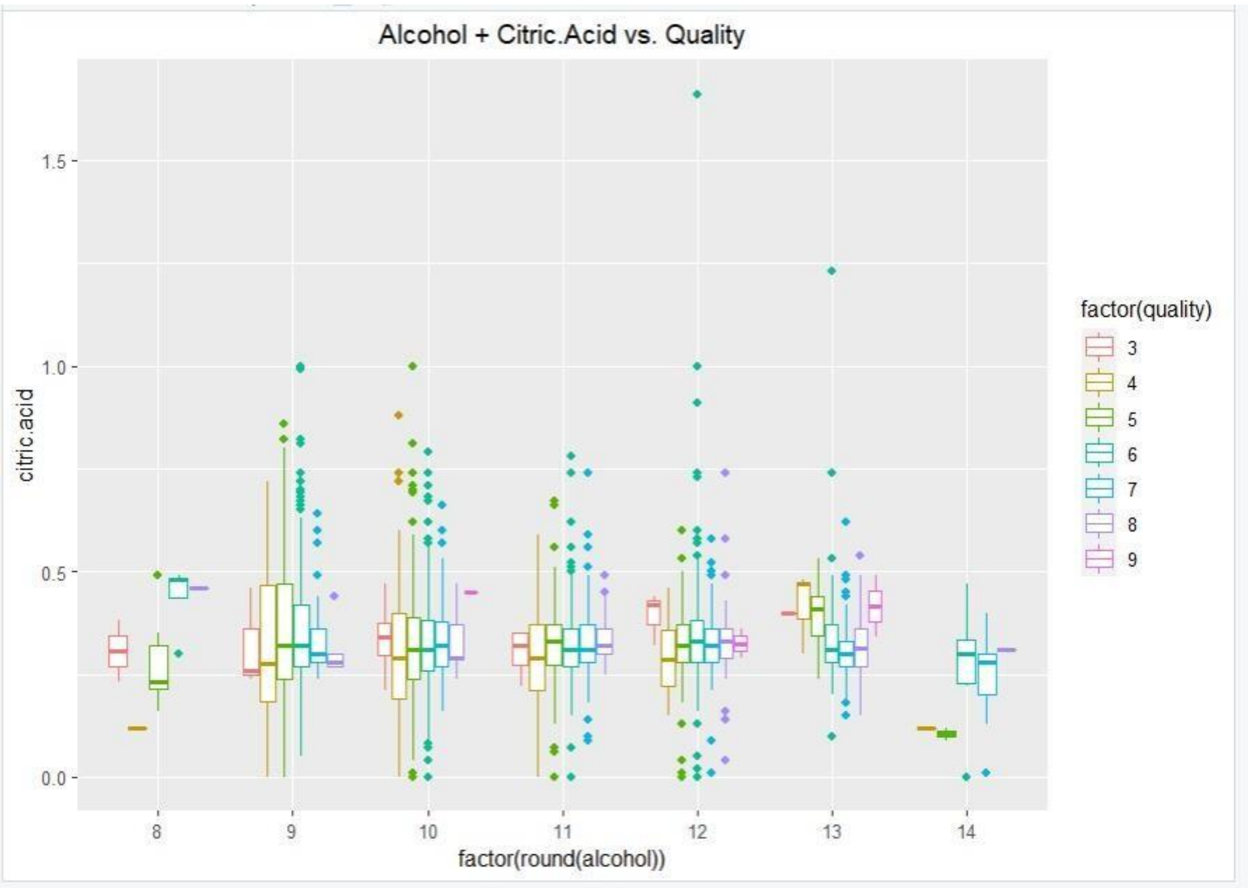

Boxplot of quality vs. sulphates
sulphates
1.0
0.8
0.6
0.4
0.2
3
4
5
6
7
8
9
quality


Boxplot of quality vs. alcohol
alcohol
14
12
10
8
quality
3
4
5
6
7
8
9

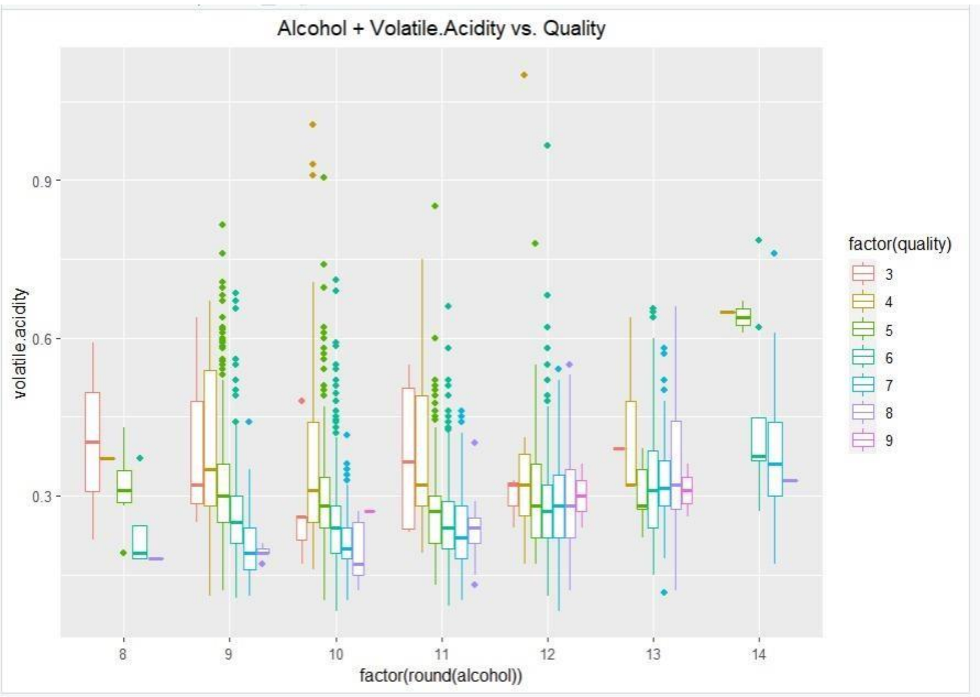

Boxplot of Quality vs. pH
pH
3.6
3.3
3.0
2.7
quality
3
4
5
6
7
8
9
Boxplot of quality vs. sulphates
sulphates
1.0
0.8
0.6
0.4
0.2
quality
3
4
5
6
7
8
9
Boxplot of quality vs. alcohol
alcohol
14
12
10
8
quality
3
4
5
6
7
8
9

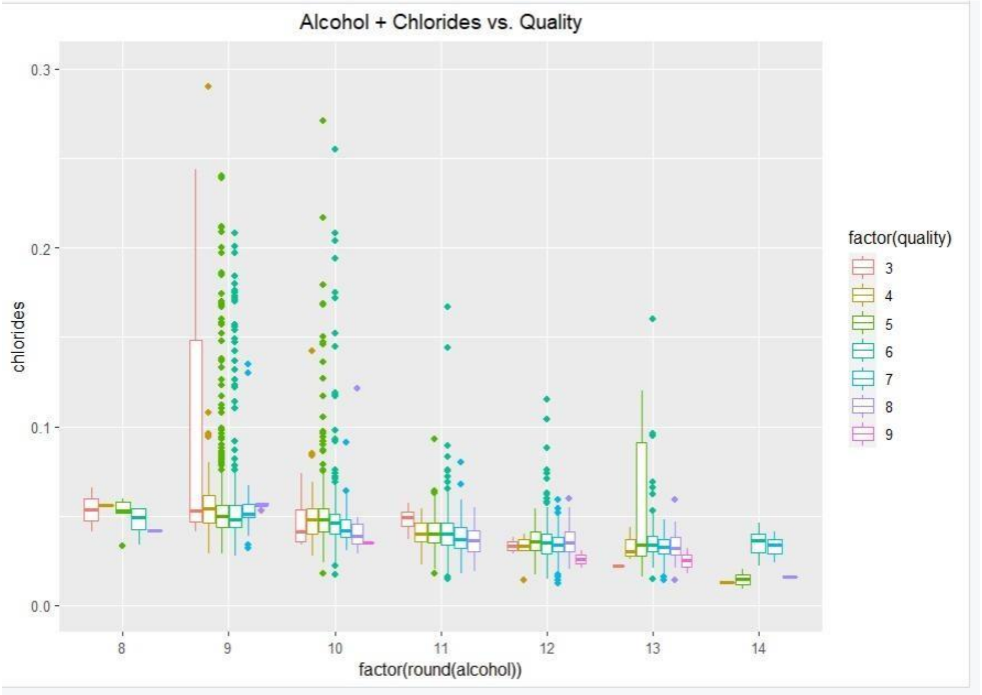
Three different patterns can be observed. First, there are positive relationships between quality and critic.acid, alcohol, and sulphates. Even though wines with a higher level of alcohol may make them less popular, they should be highly rated in quality. Second, there are negative relationships between quality and volatile.acidity, density, and pH. It is reasonable that less sweet wines and a lower level of acidity are favored in quality testings. Last, these independent variables show no significant relationship with quality: residual.sugar, chlorides, and total.sulfur.dioxide.

To dive deep into relationships within independent variables and with quality, I built different three-dimensional plots. When inspecting the two variables, alcohol and volatile.acidity with quality, we can see that with red wines' alcohol level between 9% to 12%, the level of volatile acidity decreases as the wines' alcohol level increases. For higher alcohol content (>12%), the pattern reverses, implying high-quality wines' popularity.



Finally, an interaction analysis using chlorides in relationships with alcohol and quality shows that the wines' quality decreases when chloride level decreases at the alcohol before 12%. However, the quality of red wine increases as the chloride level increases at the alcohol level from 12%.





Last, we considered if the collinearity problem existed in our data. As a result of correlation analysis and VIF verification, we discovered some variables with slightly high correlations. To deal with such a potential problem, we will take advantage of the LASSO regularization technique in the next modelling part.

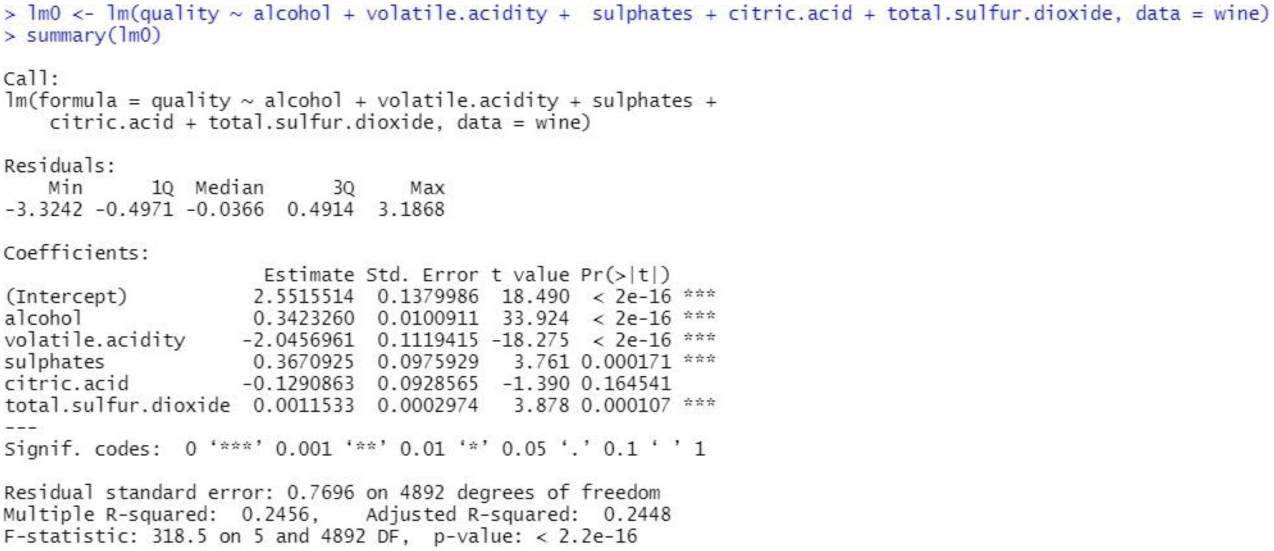
Modelling:

Based on the EDA and correlation analysis, three potential models were used in the modelling part.

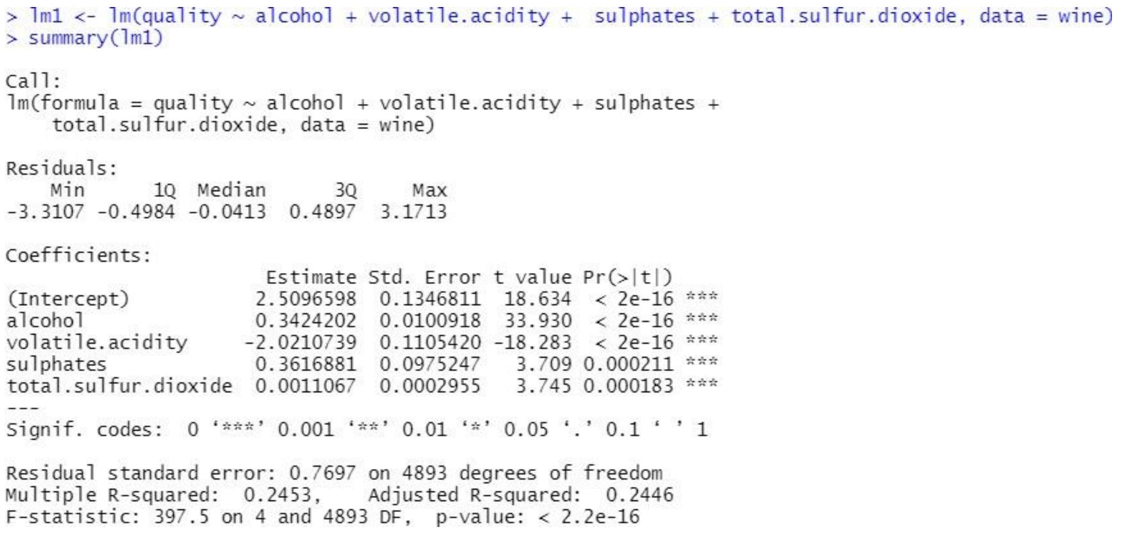
Diving deep into variable selection, we have the top 10 predictors most important to the model. It is done by using MDI (Gini Importance or Mean Decrease in Impurity) that calculates each feature's importance as the sum over the number of splits (across all trees) that include the feature, proportionally to the number of samples it splits. In comparison with Model 1 and Model 2, we have additional insights into such variables as density and pH.

Model-1:

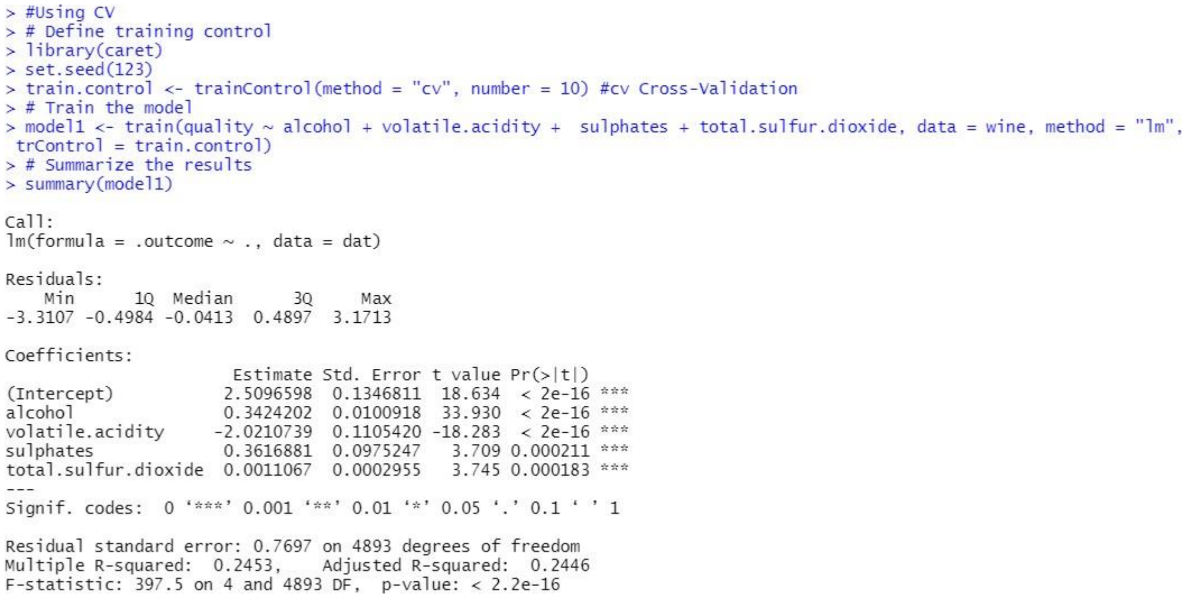
Since the correlation analysis shows that quality is highly correlated with a subset of variables (top 5), we employed multi-linear regression to build an optimal prediction model for the red wine quality. Removing a non-significant independent variable from the initial model, we got "Model 1", which included top 4 explanatory variables.



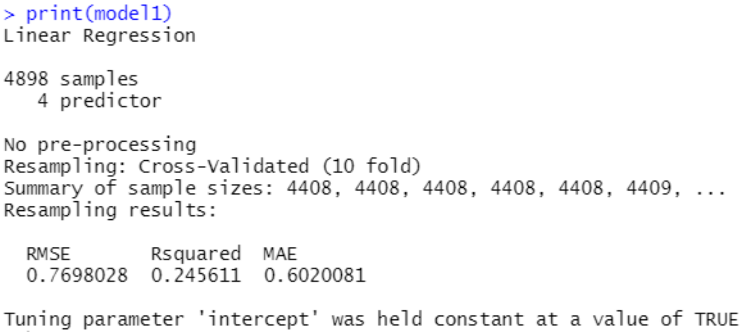
```
> lm0 <- lm(quality ~ alcohol + volatile.acidity +  sulphates + citric.acid + total.sulfur.dioxide, data = wine)
> summary(lm0)

Call:
lm(formula = quality ~ alcohol + volatile.acidity + sulphates + 
    citric.acid + total.sulfur.dioxide, data = wine)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.3242 -0.4971 -0.0366  0.4914  3.1868 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          2.5515514  0.1379986  18.490  < 2e-16 ***
alcohol              0.3423260  0.0100911  33.924  < 2e-16 ***
volatile.acidity    -2.0456961  0.1119415 -18.275  < 2e-16 ***
sulphates            0.3670925  0.0975929   3.761 0.000171 ***
citric.acid         -0.1290863  0.0928565  -1.390 0.164541    
total.sulfur.dioxide  0.0011533  0.0002974   3.878 0.000107 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7696 on 4892 degrees of freedom
Multiple R-squared:  0.2456,	Adjusted R-squared:  0.2448 
F-statistic: 318.5 on 5 and 4892 DF,  p-value: < 2.2e-16
```



```
> lm1 <- lm(quality ~ alcohol + volatile.acidity +  sulphates + total.sulfur.dioxide, data = wine)
> summary(lm1)

Call:
lm(formula = quality ~ alcohol + volatile.acidity + sulphates + 
    total.sulfur.dioxide, data = wine)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.3107 -0.4984 -0.0413  0.4897  3.1713 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          2.5096598  0.1346811  18.634  < 2e-16 ***
alcohol              0.3424202  0.0100918  33.930  < 2e-16 ***
volatile.acidity    -2.0210739  0.1105420 -18.283  < 2e-16 ***
sulphates            0.3616881  0.0975247   3.709 0.000211 ***
total.sulfur.dioxide 0.0011067  0.0002955   3.745 0.000183 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7697 on 4893 degrees of freedom
Multiple R-squared:  0.2453,	Adjusted R-squared:  0.2446 
F-statistic: 397.5 on 4 and 4893 DF,  p-value: < 2.2e-16
```



```
> #Using CV
> # Define training control
> library(caret)
> set.seed(123)
> train.control <- trainControl(method = "cv", number = 10) #cv Cross-Validation
> # Train the model
> model1 <- train(quality ~ alcohol + volatile.acidity +  sulphates + total.sulfur.dioxide, data = wine, method = "lm",
 trControl = train.control)
> # Summarize the results
> summary(model1)

Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.3107 -0.4984 -0.0413  0.4897  3.1713 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          2.5096598  0.1346811  18.634  < 2e-16 ***
alcohol              0.3424202  0.0100918  33.930  < 2e-16 ***
volatile.acidity    -2.0210739  0.1105420 -18.283  < 2e-16 ***
sulphates            0.3616881  0.0975247   3.709 0.000211 ***
total.sulfur.dioxide 0.0011067  0.0002955   3.745 0.000183 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7697 on 4893 degrees of freedom
Multiple R-squared:  0.2453,	Adjusted R-squared:  0.2446 
F-statistic: 397.5 on 4 and 4893 DF,  p-value: < 2.2e-16
```

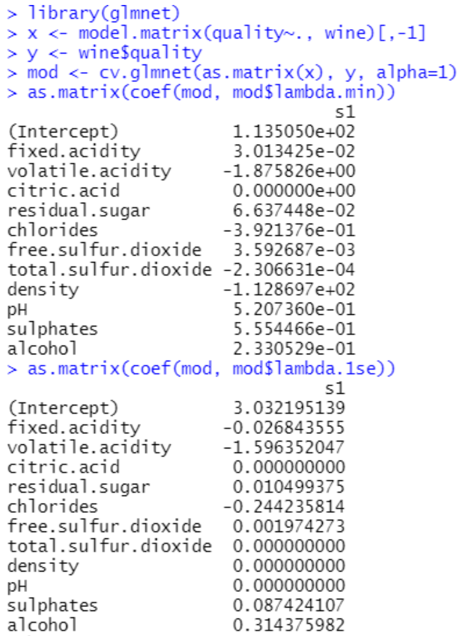


```
> print(model1)
Linear Regression 

4898 samples
   4 predictor

No pre-processing
Resampling: Cross-Validated (10 fold) 
Summary of sample sizes: 4408, 4408, 4408, 4408, 4408, 4409, ... 
Resampling results:

  RMSE       Rsquared  MAE      
  0.7698028  0.245611  0.6020081

Tuning parameter 'intercept' was held constant at a value of TRUE
```

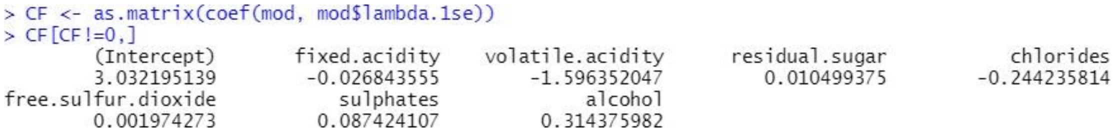
In Model 1, all identified variables are highly correlated with our target variable (quality) and show statistical significance. Alcohol and sulphates have positive relationships with quality, implying that the more level of alcohol and sulphates will translate into a higher quality of red wine. Reversely, there are negative relationships between both volatile.acidity and total.sulfur.dioxide and quality, showing that people expect a low level of acetic acid and SO2 in high-quality wine. A large amount of acetic acid may lead to an unpleasant vinegar taste, for example.

Model-2: *Lasso*

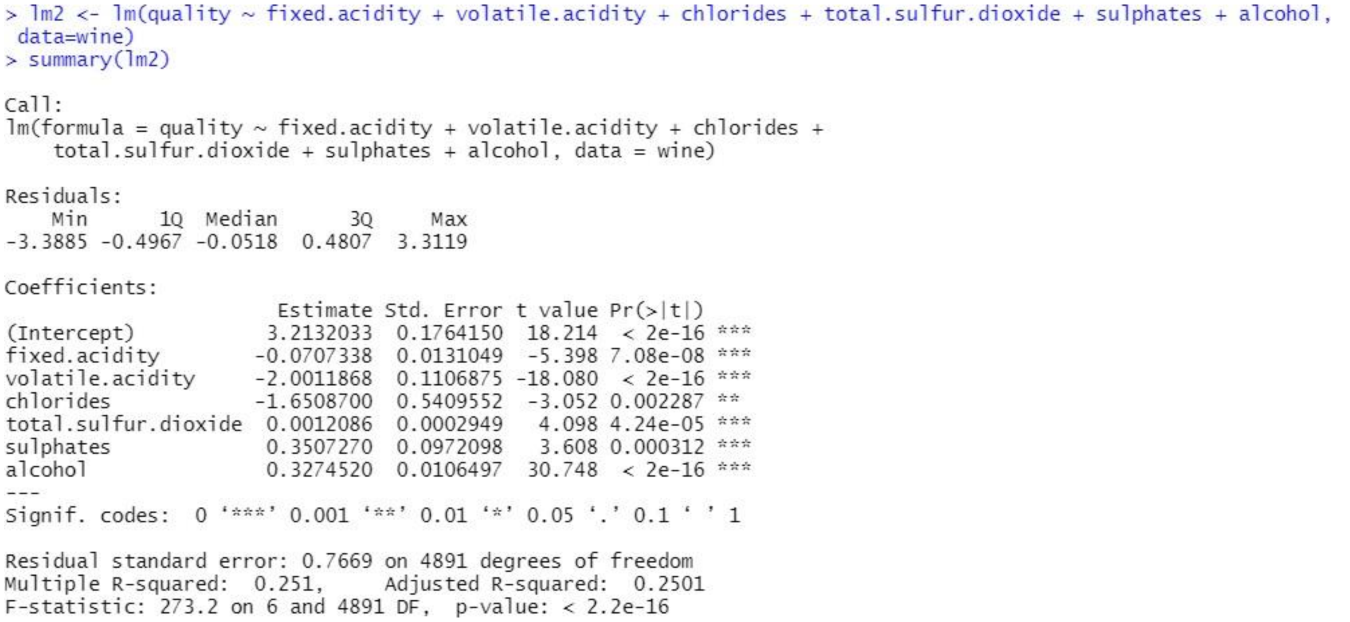
Next, using the LASSO method, I came up with the second model (“Model 2”) that performs both variable selection and regularization. This resulted in a subset of predictors (our “Top 6”) that minimizes prediction error for a quantitative response variable — quality. This subset includes six variables: fixed.acidity, volatile.acidity, chlorides, total.sulfur.dioxide, sulphates, and alcohol.



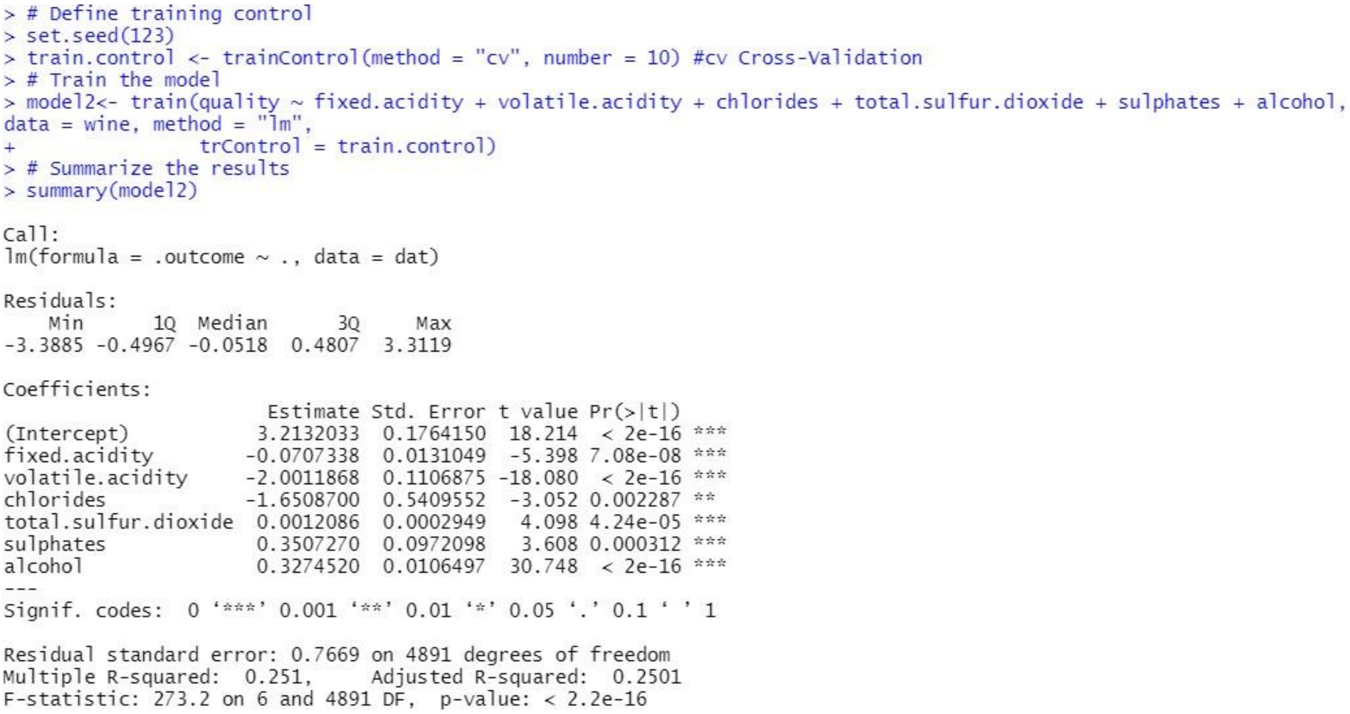
```
> library(glmnet)
> x <- model.matrix(quality~., wine)[,-1]
> y <- wine$quality
> mod <- cv.glmnet(as.matrix(x), y, alpha=1)
> as.matrix(coef(mod, mod$lambda.min))
                                s1
(Intercept)           1.135050e+02
fixed.acidity         3.013425e-02
volatile.acidity     -1.875826e+00
citric.acid           0.000000e+00
residual.sugar        6.637448e-02
chlorides            -3.921376e-01
free.sulfur.dioxide   3.592687e-03
total.sulfur.dioxide -2.306631e-04
density              -1.128697e+02
pH                    5.207360e-01
sulphates             5.554466e-01
alcohol               2.330529e-01
> as.matrix(coef(mod, mod$lambda.1se))
                               s1
(Intercept)           3.032195139
fixed.acidity        -0.026843555
volatile.acidity     -1.596352047
citric.acid           0.000000000
residual.sugar        0.010499375
chlorides            -0.244235814
free.sulfur.dioxide   0.001974273
total.sulfur.dioxide  0.000000000
density               0.000000000
pH                    0.000000000
sulphates             0.087424107
alcohol               0.314375982
```



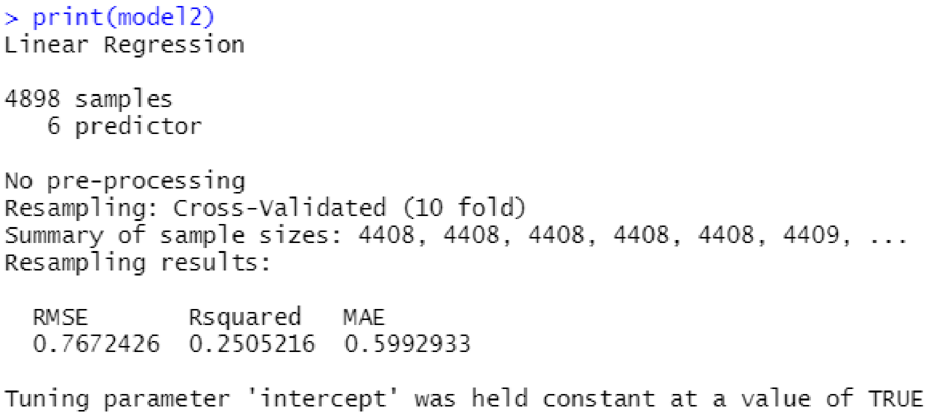
```
> CF <- as.matrix(coef(mod, mod$lambda.1se))
> CF[CF!=0,]
        (Intercept)       fixed.acidity    volatile.acidity      residual.sugar           chlorides
        3.032195139        -0.026843555        -1.596352047         0.010499375        -0.244235814
free.sulfur.dioxide           sulphates             alcohol
        0.001974273         0.087424107         0.314375982
```



```
> lm2 <- lm(quality ~ fixed.acidity + volatile.acidity + chlorides + total.sulfur.dioxide + sulphates + alcohol,
 data=wine)
> summary(lm2)

Call:
lm(formula = quality ~ fixed.acidity + volatile.acidity + chlorides + 
    total.sulfur.dioxide + sulphates + alcohol, data = wine)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.3885 -0.4967 -0.0518  0.4807  3.3119 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)           3.2132033  0.1764150  18.214  < 2e-16 ***
fixed.acidity        -0.0707338  0.0131049  -5.398 7.08e-08 ***
volatile.acidity     -2.0011868  0.1106875 -18.080  < 2e-16 ***
chlorides            -1.6508700  0.5409552  -3.052 0.002287 ** 
total.sulfur.dioxide  0.0012086  0.0002949   4.098 4.24e-05 ***
sulphates             0.3507270  0.0972098   3.608 0.000312 ***
alcohol               0.3274520  0.0106497  30.748  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7669 on 4891 degrees of freedom
Multiple R-squared:  0.251,	Adjusted R-squared:  0.2501 
F-statistic: 273.2 on 6 and 4891 DF,  p-value: < 2.2e-16
```



```
> # Define training control
> set.seed(123)
> train.control <- trainControl(method = "cv", number = 10) #cv Cross-Validation
> # Train the model
> model2<- train(quality ~ fixed.acidity + volatile.acidity + chlorides + total.sulfur.dioxide + sulphates + alcohol,
data = wine, method = "lm",
+                trControl = train.control)
> # Summarize the results
> summary(model2)

Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.3885 -0.4967 -0.0518  0.4807  3.3119 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)           3.2132033  0.1764150  18.214  < 2e-16 ***
fixed.acidity        -0.0707338  0.0131049  -5.398 7.08e-08 ***
volatile.acidity     -2.0011868  0.1106875 -18.080  < 2e-16 ***
chlorides            -1.6508700  0.5409552  -3.052 0.002287 ** 
total.sulfur.dioxide  0.0012086  0.0002949   4.098 4.24e-05 ***
sulphates             0.3507270  0.0972098   3.608 0.000312 ***
alcohol               0.3274520  0.0106497  30.748  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7669 on 4891 degrees of freedom
Multiple R-squared:  0.251,	Adjusted R-squared:  0.2501 
F-statistic: 273.2 on 6 and 4891 DF,  p-value: < 2.2e-16
```

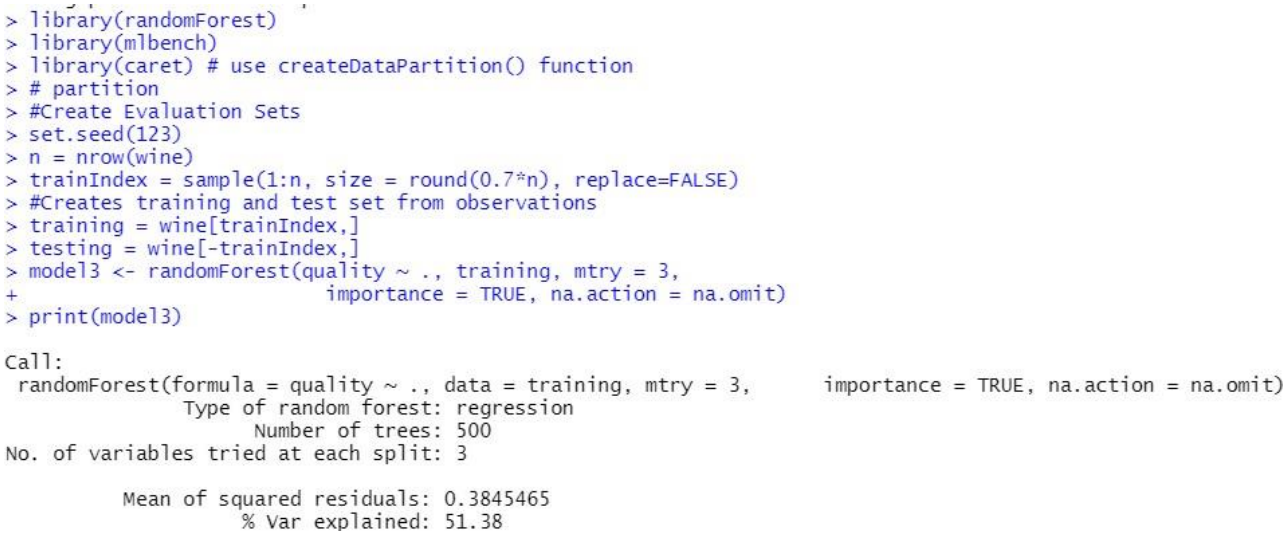


```
> print(model2)
Linear Regression 

4898 samples
   6 predictor

No pre-processing
Resampling: Cross-Validated (10 fold) 
Summary of sample sizes: 4408, 4408, 4408, 4408, 4408, 4409, ... 
Resampling results:

  RMSE       Rsquared   MAE      
  0.7672426  0.2505216  0.5992933

Tuning parameter 'intercept' was held constant at a value of TRUE
```

All six variables have a strong correlation with our dependent variable (quality) and are statistically significant. Compared to Model 1, the new model contains two additional variables: fixed.acidity and chlories, whose marginal effects on water quality are opposite. A negative chloride estimation coefficient indicates that higher quality wine should contain less salt. In the meantime, there is a weakly positive correlation between fixed acidity and quality, indicating that non-volatile acids that do not readily evaporate should be an indicator of wine of high quality.

Model-3:

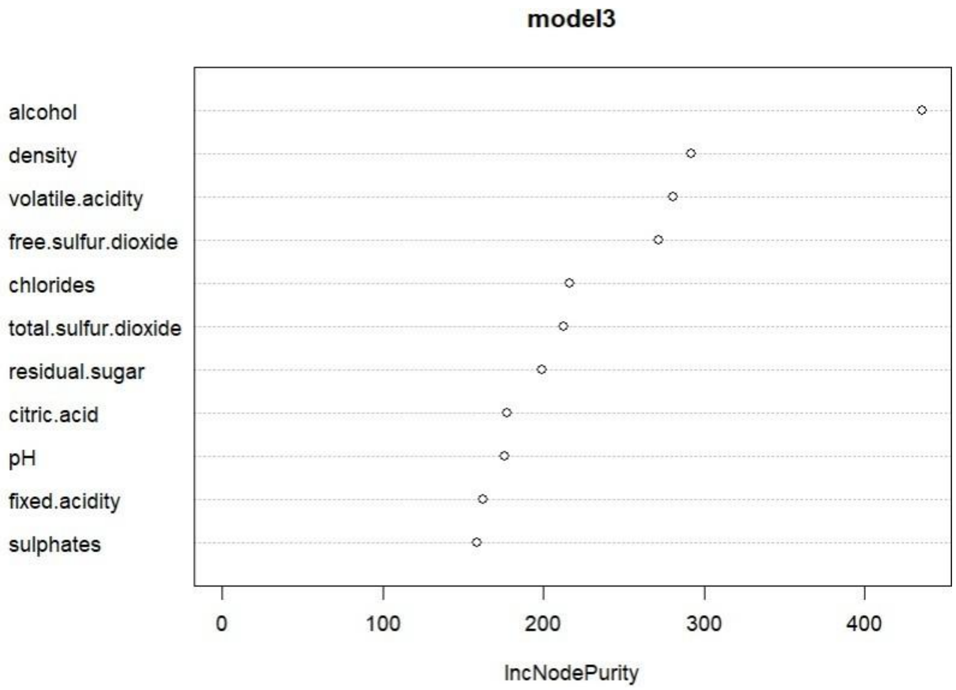
Last, we ran Random Forest as a machine learning regression tree algorithm used in the modeling process. This helps to create a random sample of multiple regression decision trees and merges them to obtain a more stable and accurate prediction through cross-validation.



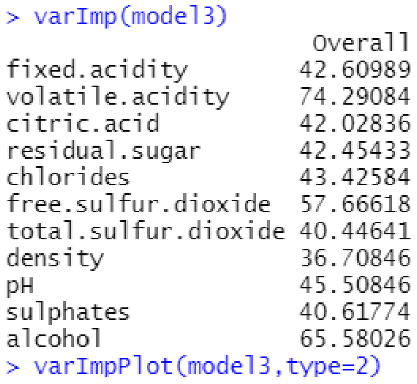
```
> library(randomForest)
> library(mlbench)
> library(caret) # use createDataPartition() function
> # partition
> #Create Evaluation Sets
> set.seed(123)
> n = nrow(wine)
> trainIndex = sample(1:n, size = round(0.7*n), replace=FALSE)
> #Creates training and test set from observations
> training = wine[trainIndex,]
> testing = wine[-trainIndex,]
> model3 <- randomForest(quality ~ ., training, mtry = 3,
+                        importance = TRUE, na.action = na.omit)
> print(model3)

Call:
 randomForest(formula = quality ~ ., data = training, mtry = 3,      importance = TRUE, na.action = na.omit)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 0.3845465
                    % Var explained: 51.38
```

Diving deep into variable selection, we have the top 10 predictors most important to the model. It is done by using MDI (Gini Importance or Mean Decrease in Impurity) that calculates each feature’s importance as the sum over the number of splits (across all trees) that include the feature, proportionally to the number of samples it splits. In comparison with Model 1 and Model 2, we have additional insights into such variables as density and pH.

The error vs the number of trees graph

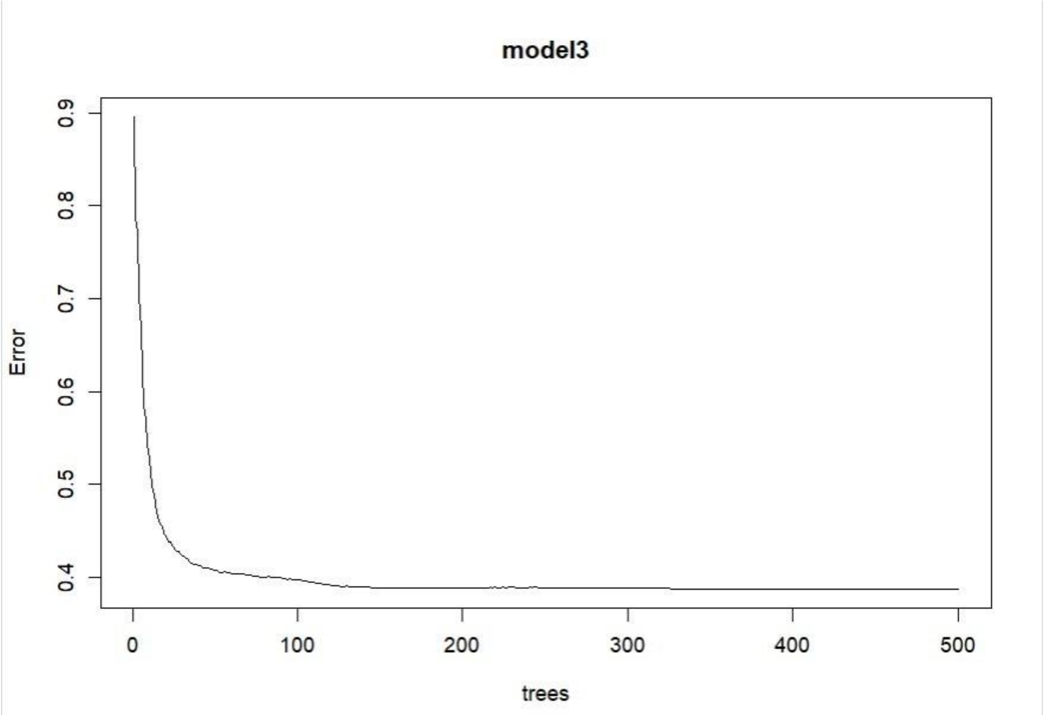


Variable Importance

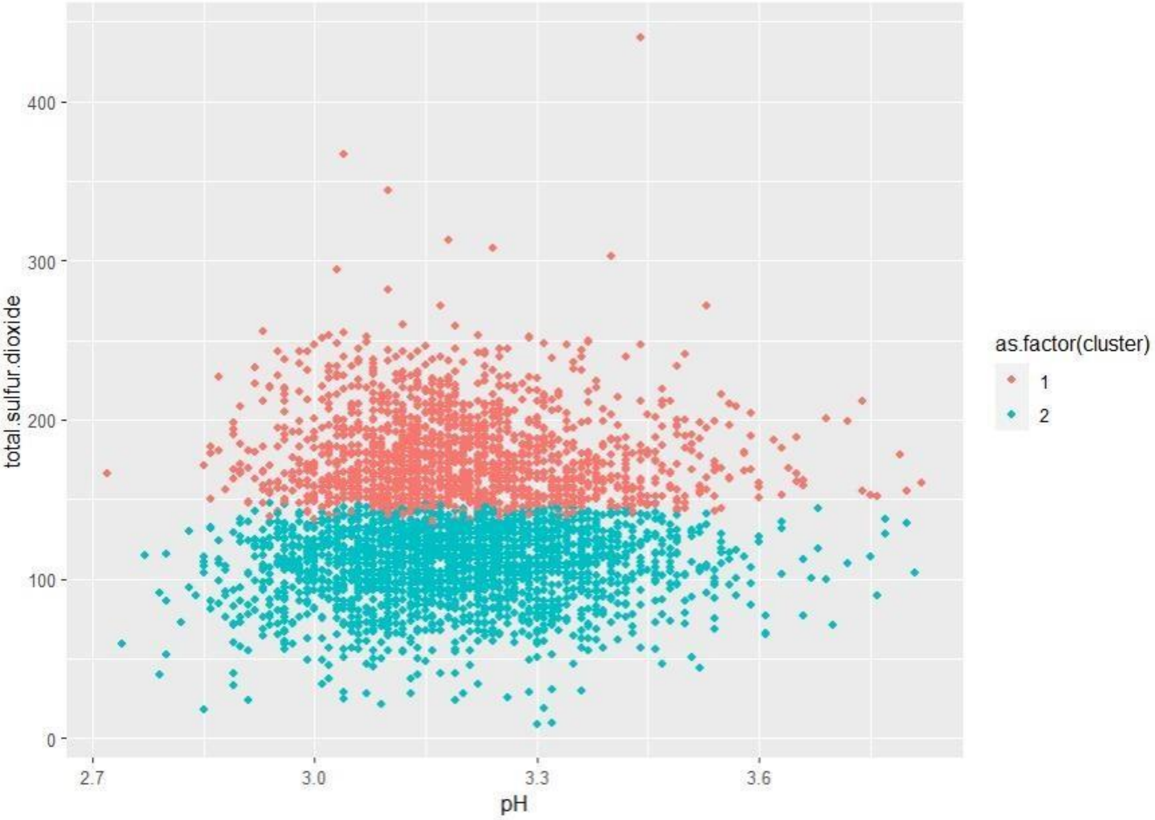


```
> varImp(model3)
                       Overall
fixed.acidity         42.60989
volatile.acidity      74.29084
citric.acid           42.02836
residual.sugar        42.45433
chlorides             43.42584
free.sulfur.dioxide   57.66618
total.sulfur.dioxide  40.44641
density               36.70846
pH                    45.50846
sulphates             40.61774
alcohol               65.58026
> varImpPlot(model3,type=2)
```

Variable Importance Plot



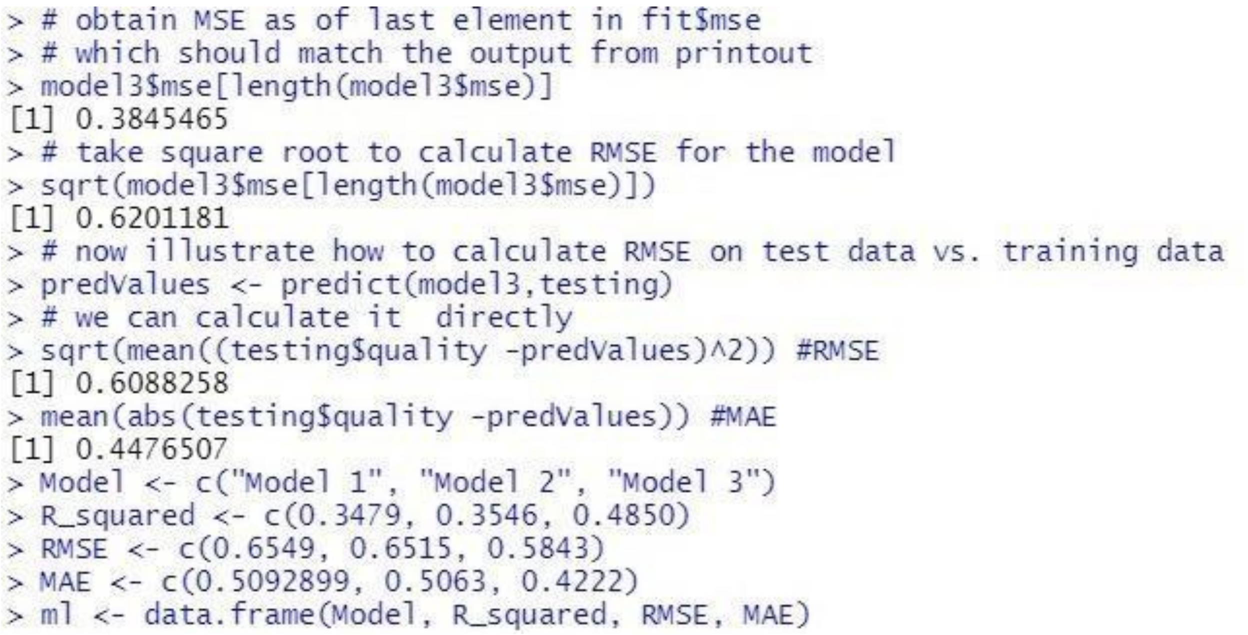
Using K-means:



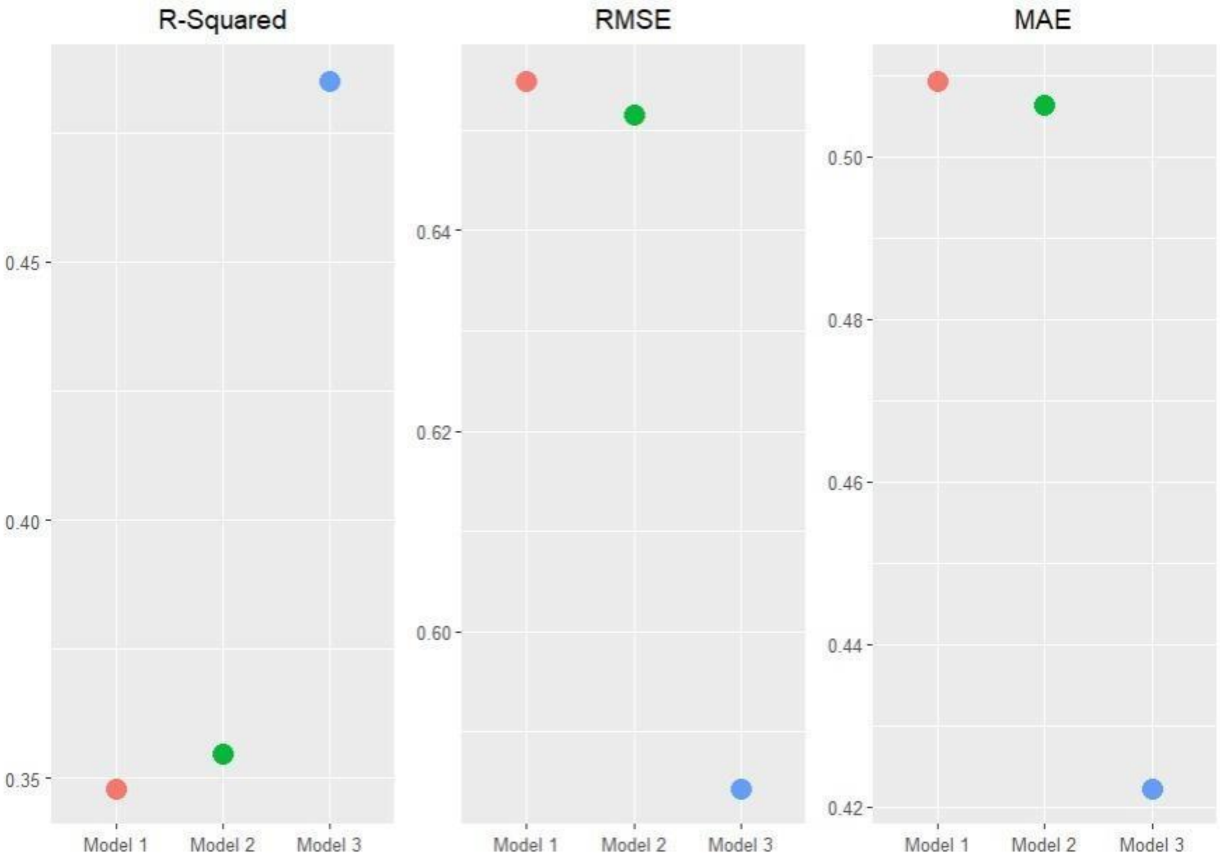
PERFORMANCE ANALYSIS

We evaluated the performance of our three models' model prediction using three metrics: R-squared, RMSE, and MAE. As anticipated, Model 3 has the best R-Squared, RMSE, and MAE values across all three metrics, with a score of 48.50%, 0.5843, and 0.4222. Models 1 and 2, whose predictors were chosen using our regularisation and correlation analysis methodologies, don't show much variation in these performance indicators.

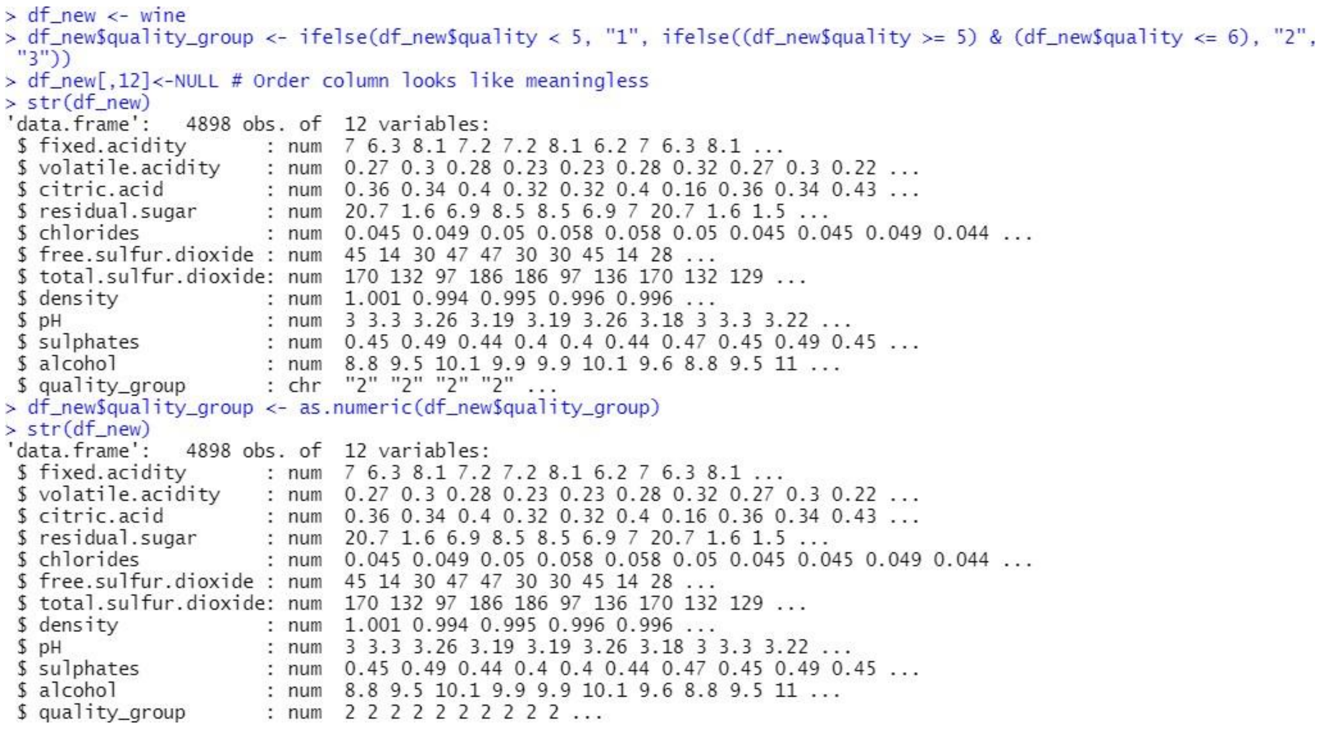
It makes sense that Model 3's Random Forest provides us with better "predictions". Models 1 and 2 may come out on top despite having subpar performance measures from the standpoint of "marginal impact" interpretation. Model 3 will be the most effective in the context of our business question, which focuses on predicting the quality of red wine.



```
> # obtain MSE as of last element in fit$mse
> # which should match the output from printout
> model3$mse[length(model3$mse)]
[1] 0.3845465
> # take square root to calculate RMSE for the model
> sqrt(model3$mse[length(model3$mse)])
[1] 0.6201181
> # now illustrate how to calculate RMSE on test data vs. training data
> predValues <- predict(model3,testing)
> # we can calculate it  directly
> sqrt(mean((testing$quality -predValues)^2)) #RMSE
[1] 0.6088258
> mean(abs(testing$quality -predValues)) #MAE
[1] 0.4476507
> Model <- c("Model 1", "Model 2", "Model 3")
> R_squared <- c(0.3479, 0.3546, 0.4850)
> RMSE <- c(0.6549, 0.6515, 0.5843)
> MAE <- c(0.5092899, 0.5063, 0.4222)
> ml <- data.frame(Model, R_squared, RMSE, MAE)
```

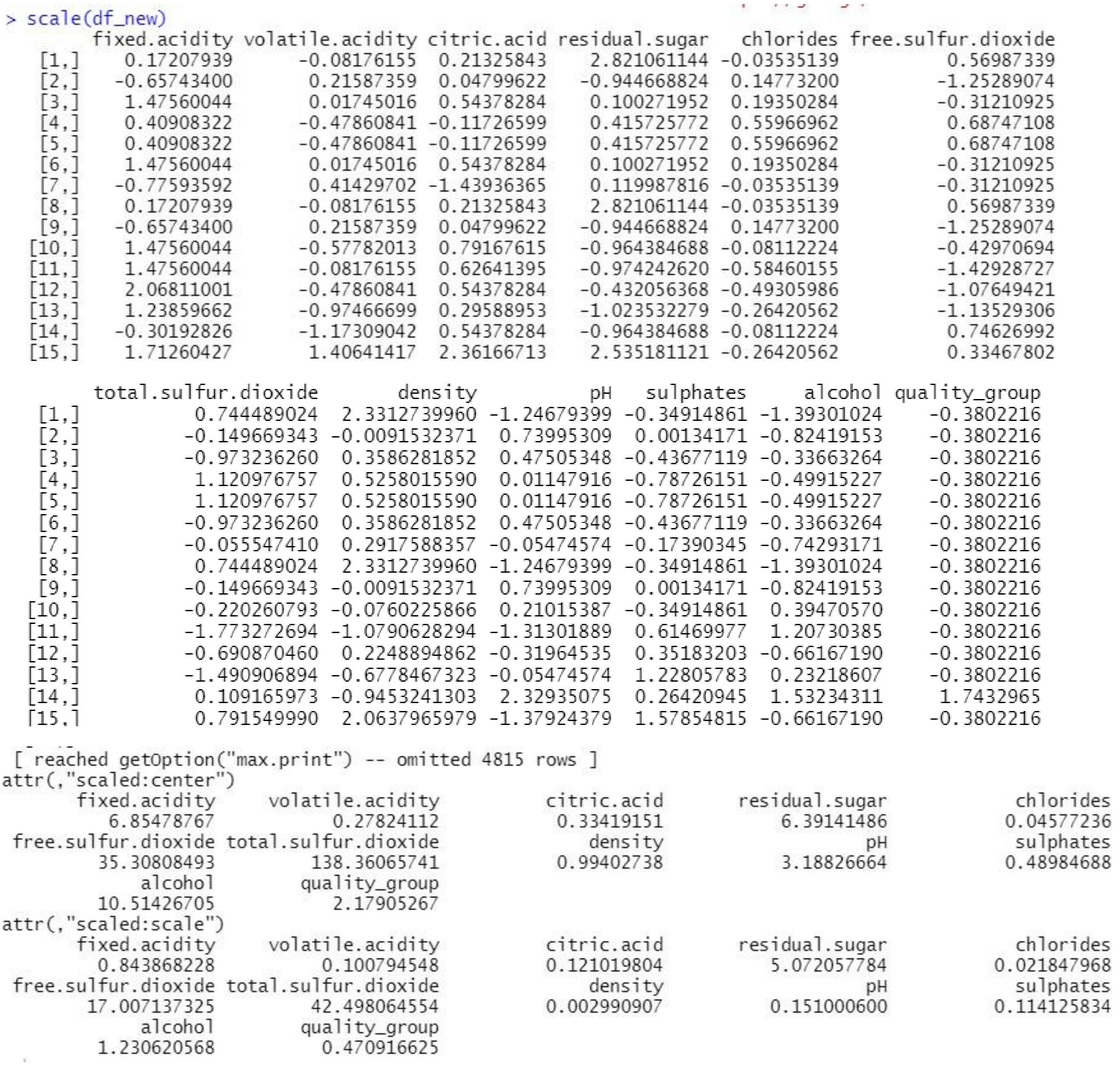


Clustering – Clearing Data



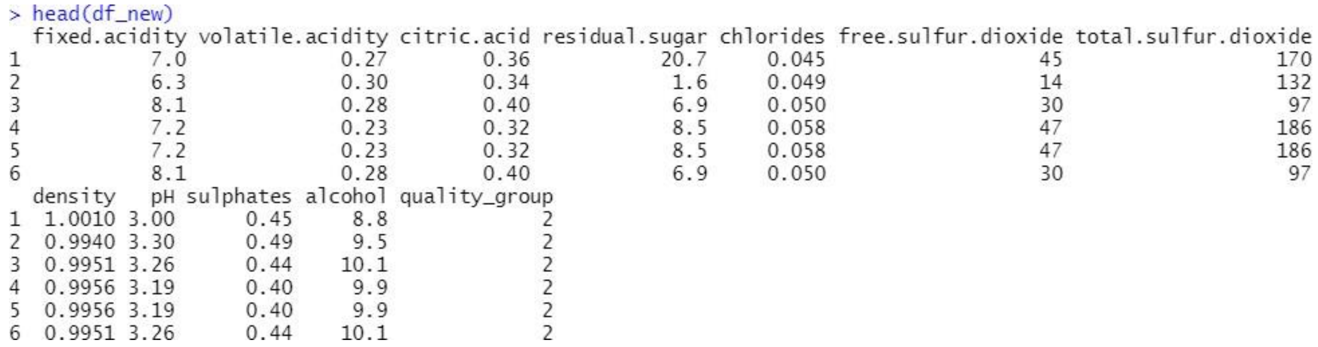
```
> df_new <- wine
> df_new$quality_group <- ifelse(df_new$quality < 5, "1", ifelse((df_new$quality >= 5) & (df_new$quality <= 6), "2",
 "3"))
> df_new[,12]<-NULL # Order column looks like meaningless
> str(df_new)
'data.frame':	4898 obs. of  12 variables:
 $ fixed.acidity       : num  7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
 $ volatile.acidity    : num  0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
 $ citric.acid         : num  0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
 $ residual.sugar      : num  20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
 $ chlorides           : num  0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
 $ free.sulfur.dioxide : num  45 14 30 47 47 30 30 45 14 28 ...
 $ total.sulfur.dioxide: num  170 132 97 186 186 97 136 170 132 129 ...
 $ density             : num  1.001 0.994 0.995 0.996 0.996 ...
 $ pH                  : num  3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...
 $ sulphates           : num  0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
 $ alcohol             : num  8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
 $ quality_group       : chr  "2" "2" "2" "2" ...
> df_new$quality_group <- as.numeric(df_new$quality_group)
> str(df_new)
'data.frame':	4898 obs. of  12 variables:
 $ fixed.acidity       : num  7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
 $ volatile.acidity    : num  0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
 $ citric.acid         : num  0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
 $ residual.sugar      : num  20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
 $ chlorides           : num  0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
 $ free.sulfur.dioxide : num  45 14 30 47 47 30 30 45 14 28 ...
 $ total.sulfur.dioxide: num  170 132 97 186 186 97 136 170 132 129 ...
 $ density             : num  1.001 0.994 0.995 0.996 0.996 ...
 $ pH                  : num  3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...
 $ sulphates           : num  0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
 $ alcohol             : num  8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
 $ quality_group       : num  2 2 2 2 2 2 2 2 2 2 ...
```

Standardizing Data



```
> scale(df_new)
      fixed.acidity volatile.acidity citric.acid residual.sugar   chlorides free.sulfur.dioxide
 [1,]    0.17207939      -0.08176155  0.21325843    2.821061144 -0.03535139          0.56987339
 [2,]   -0.65743400       0.21587359  0.04799622   -0.944668824  0.14773200         -1.25289074
 [3,]    1.47560044       0.01745016  0.54378284    0.100271952  0.19350284         -0.31210925
 [4,]    0.40908322      -0.47860841 -0.11726599    0.415725772  0.55966962          0.68747108
 [5,]    0.40908322      -0.47860841 -0.11726599    0.415725772  0.55966962          0.68747108
 [6,]    1.47560044       0.01745016  0.54378284    0.100271952  0.19350284         -0.31210925
 [7,]   -0.77593592       0.41429702 -1.43936365    0.119987816 -0.03535139         -0.31210925
 [8,]    0.17207939      -0.08176155  0.21325843    2.821061144 -0.03535139          0.56987339
 [9,]   -0.65743400       0.21587359  0.04799622   -0.944668824  0.14773200         -1.25289074
[10,]    1.47560044      -0.57782013  0.79167615   -0.964384688 -0.08112224         -0.42970694
[11,]    1.47560044      -0.08176155  0.62641395   -0.974242620 -0.58460155         -1.42928727
[12,]    2.06811001      -0.47860841  0.54378284   -0.432056368 -0.49305986         -1.07649421
[13,]    1.23859662      -0.97466699  0.29588953   -1.023532279 -0.26420562         -1.13529306
[14,]   -0.30192826      -1.17309042  0.54378284   -0.964384688 -0.08112224          0.74626992
[15,]    1.71260427       1.40641417  2.36166713    2.535181121 -0.26420562          0.33467802

      total.sulfur.dioxide       density          pH   sulphates     alcohol quality_group
 [1,]          0.744489024  2.3312739960 -1.24679399 -0.34914861 -1.39301024    -0.3802216
 [2,]         -0.149669343 -0.0091532371  0.73995309  0.00134171 -0.82419153    -0.3802216
 [3,]         -0.973236260  0.3586281852  0.47505348 -0.43677119 -0.33663264    -0.3802216
 [4,]          1.120976757  0.5258015590  0.01147916 -0.78726151 -0.49915227    -0.3802216
 [5,]          1.120976757  0.5258015590  0.01147916 -0.78726151 -0.49915227    -0.3802216
 [6,]         -0.973236260  0.3586281852  0.47505348 -0.43677119 -0.33663264    -0.3802216
 [7,]         -0.055547410  0.2917588357 -0.05474574 -0.17390345 -0.74293171    -0.3802216
 [8,]          0.744489024  2.3312739960 -1.24679399 -0.34914861 -1.39301024    -0.3802216
 [9,]         -0.149669343 -0.0091532371  0.73995309  0.00134171 -0.82419153    -0.3802216
[10,]         -0.220260793 -0.0760225866  0.21015387 -0.34914861  0.39470570    -0.3802216
[11,]         -1.773272694 -1.0790628294 -1.31301889  0.61469977  1.20730385    -0.3802216
[12,]         -0.690870460  0.2248894862 -0.31964535  0.35183203 -0.66167190    -0.3802216
[13,]         -1.490906894 -0.6778467323 -0.05474574  1.22805783  0.23218607    -0.3802216
[14,]          0.109165973 -0.9453241303  2.32935075  0.26420945  1.53234311     1.7432965
[15,]          0.791549990  2.0637965979 -1.37924379  1.57854815 -0.66167190    -0.3802216

 [ reached getOption("max.print") -- omitted 4815 rows ]
attr(,"scaled:center")
      fixed.acidity    volatile.acidity         citric.acid      residual.sugar           chlorides
         6.85478767          0.27824112          0.33419151          6.39141486          0.04577236
free.sulfur.dioxide total.sulfur.dioxide            density                  pH           sulphates
        35.30808493        138.36065741          0.99402738          3.18826664          0.48984688
            alcohol       quality_group
        10.51426705          2.17905267
attr(,"scaled:scale")
      fixed.acidity    volatile.acidity         citric.acid      residual.sugar           chlorides
        0.843868228         0.100794548         0.121019804         5.072057784         0.021847968
free.sulfur.dioxide total.sulfur.dioxide            density                  pH           sulphates
       17.007137325        42.498064554         0.002990907         0.151000600         0.114125834
            alcohol       quality_group
        1.230620568         0.470916625
```

Getting Top 6 Records of Standardized Data using Head



```
> head(df_new)
  fixed.acidity volatile.acidity citric.acid residual.sugar chlorides free.sulfur.dioxide total.sulfur.dioxide
1           7.0             0.27        0.36           20.7     0.045                  45                  170
2           6.3             0.30        0.34            1.6     0.049                  14                  132
3           8.1             0.28        0.40            6.9     0.050                  30                   97
4           7.2             0.23        0.32            8.5     0.058                  47                  186
5           7.2             0.23        0.32            8.5     0.058                  47                  186
6           8.1             0.28        0.40            6.9     0.050                  30                   97
  density   pH sulphates alcohol quality_group
1  1.0010 3.00      0.45     8.8             2
2  0.9940 3.30      0.49     9.5             2
3  0.9951 3.26      0.44    10.1             2
4  0.9956 3.19      0.40     9.9             2
5  0.9956 3.19      0.40     9.9             2
6  0.9951 3.26      0.44    10.1             2
```

LIMITATIONS

The main problem came from the fact that our data set was unbalanced. A majority of the quality values were “regular” (5 and 6), which made no significant contribution to finding an optimal model. These values made it harder to identify each factor’s different influence on a “high” or “low” quality of the wine, which was the main focus of this analysis. In order to improve our predictive model, we need more balanced data. Another limitation worth mentioned from the data set was it only had 12 attributes, which can narrow down the accuracy of our predicting quality of red wine. The solution for this is to include more relevant data features, like the year of harvest, brew time, location, or wine type.

CONCLUSION

In the aim of analysing the Wine Quality, we have performed Data preparation under which we have performed data cleaning, data exploration and transformation where we removed duplicate records, checked for NA values and derived each feature's statistical summary to detect any problem like outliers and abnormal distributions.

We tried to find which variables are likely to affect the quality of the red wine the most. We have performed a correlation analysis of the independent variables against variable, quality. The analysis gave us a list of variables of interest that had highest correlations with quality.

Then we created density plot visualizing the spread of the data, in order to analyze the relationship with the dependent variable.

After analysing the density plots, we plot the interaction between the numeric variables of interest and the dependent variable of quantity.

We performed modelling to estimate the relations between the factors effecting the quality of wine. And finally, evaluation, where we evaluated the relations and factors to predict the best quality of wine.

By analysing the physicochemical tests samples data of wines, we were able to create a model that can help industry producers, distributors, and sellers predict the quality of wine products and have a better understanding of each critical and up-to-date features. We found that the **Model 3 — Random Forest-based** feature sets performed better than others. In general, using Model 3 as our best model for prediction, we determined four of the features as the most influential: volatile acidity, citric acid, sulphates, and alcohol. To be more specific, high-quality wines seem to have lower volatile acidity, higher alcohol, and medium-high sulphate values. Meanwhile, lower-quality wines tend to have low values for citric acid.

FUTURE WORKS

In the future, we can try to analyse and predict quality of other products such as edible oils, hair oils, medicines, soaps and also use other and better machine learning techniques and performance metrics for better performance and comparison of results. We can also further work on this model to improve its performance and efficiency so that the prediction of the highly correlated and important factors and their concentrations to produce a better tasting and quality wine that will help the wine producers boost their sales.

APPENDIX – I (Codes)

```
wine = read.csv("C:\\Users\\lalli\\Downloads\\whiteWine.csv",header = TRUE, sep = ",")
wine


library("dplyr")


distinct(wine)
View(wine)


write.csv(wine,"C:\\Users\\lalli\\Downloads\\whiteWineC.csv", row.names = FALSE)


dim(wine)
str(wine)
summary(wine)


# 3.2. Plotting the marginal distributions of key numerical quantities of interest
p1 <- ggplot(wine, aes(x=fixed.acidity)) +geom_density()
```

```
p1 + geom_vline(aes(xintercept=mean(fixed.acidity)),color="red", linetype="dashed",
size=1)+ggtitle("Fixed Acidity vs Density")+labs(x="Fixed Acidity",y="Density")


p2 <- ggplot(wine, aes(x=volatile.acidity)) + geom_density()

p2 + geom_vline(aes(xintercept=mean(volatile.acidity)),color="blue",
linetype="dashed", size=1)+ggtitle("Volatile Acidity vs Density")+labs(x="Volatile
Acidity",y="Density")


p3 <- ggplot(wine, aes(x=citric.acid)) + geom_density()

p3 + geom_vline(aes(xintercept=mean(citric.acid)),color="green", linetype="dashed",
size=1)+ggtitle("Citric Acid vs Density")+labs(x="Citric Acid",y="Density")


p4 <- ggplot(wine, aes(x=residual.sugar)) + geom_density()

p4 + geom_vline(aes(xintercept=mean(residual.sugar)),color="blue",
linetype="dashed", size=1)+ggtitle("Fixed Acidity vs Density")+labs(x="Residual
Sugar",y="Density")


p5 <- ggplot(wine, aes(x=chlorides)) + geom_density()

p5 + geom_vline(aes(xintercept=mean(chlorides)),color="blue", linetype="dashed",
size=1)+ggtitle("Chlorides vs Density")+labs(x="Chlorides",y="Density")


p6 <- ggplot(wine, aes(x=free.sulfur.dioxide)) + geom_density()

p6 + geom_vline(aes(xintercept=mean(free.sulfur.dioxide)),color="blue",
linetype="dashed", size=1)+ggtitle("Suplur Dioxide vs
Density")+labs(x="SuplurDioxide",y="Density")


p7 <- ggplot(wine, aes(x=total.sulfur.dioxide)) + geom_density()

p7 + geom_vline(aes(xintercept=mean(total.sulfur.dioxide)),color="blue",
linetype="dashed", size=1)+ggtitle("Total Sulphur Dioxide vs Density")+labs(x="Total
Sulphur Dioxide",y="Density")


p8 <- ggplot(wine, aes(x=density)) + geom_density()
```

```
p8 + geom_vline(aes(xintercept=mean(density)),color="blue", linetype="dashed",
size=1)+ggtitle("Density vs Density")+labs(x="Density",y="Density")


p9 <- ggplot(wine, aes(x=pH)) + geom_density()
p9 + geom_vline(aes(xintercept=mean(pH)),color="blue", linetype="dashed",
size=1)+ggtitle("pH vs Density")+labs(x="pH",y="Density")


p10 <- ggplot(wine, aes(x=sulphates)) + geom_density()
p10 + geom_vline(aes(xintercept=mean(sulphates)),color="blue", linetype="dashed",
size=1)+ggtitle("Sulphates vs Density")+labs(x="Sulphates",y="Density")


p11 <- ggplot(wine, aes(x=alcohol)) + geom_density()
p11 + geom_vline(aes(xintercept=mean(alcohol)),color="blue", linetype="dashed",
size=1)+ggtitle("Alcohol vs Density")+labs(x="Alcohol",y="Density")


install.packages("ggpubr")
library("ggpubr")


ggarrange(p1, p2, p3, p4, nrow = 2, ncol =2)
ggarrange(p5, p6, p7, p8, nrow = 2, ncol =2)
ggarrange(p9, p10, p11, nrow = 2, ncol =2)




b1 <- boxplot(wine$fixed.acidity, col="slategray2", pch=19,main="Fixed Acidity")
b2 <- boxplot(wine$volatile.acidity, col="slategray2", pch=19,main="Volatitle Acidity")
b3 <- boxplot(wine$citric.acid, col="slategray2", pch=19,main="Citric Acid")
b4 <- boxplot(wine$residual.sugar, col="slategray2", pch=19,main="Residual Sugar")
b5 <- boxplot(wine$chlorides, col="slategray2", pch=19,main="Chlorides")
```

```
b6 <- boxplot(wine$free.sulfur.dioxide, col="slategray2", pch=19,main="Free Suplhur Dioxide")

b7 <- boxplot(wine$total.sulfur.dioxide, col="slategray2", pch=19,main="Total Sulphur Dioxide")

b8 <- boxplot(wine$density, col="slategray2", pch=19,main="Density")

b9 <- boxplot(wine$pH, col="slategray2", pch=19,main="pH")

b10 <- boxplot(wine$sulphates, col="slategray2", pch=19,main="Sulphates")

b11 <- boxplot(wine$alcohol, col="slategray2", pch=19,main="Alcohol")




## 3.3. Inspecting quality against numerical variables of interest


g1 <- ggplot(wine, aes(factor(quality), fixed.acidity, fill=factor(quality))) + geom_boxplot() +labs(x = "quality", y = "fixed.acidity", title = "Boxplot of Quality vs. fixed.acidity") + theme(legend.position = 'none', plot.title = element_text(size = 9, hjust=0.5))

g1


g2 <- ggplot(wine, aes(factor(quality), volatile.acidity, fill=factor(quality))) + geom_boxplot() +labs(x = "quality", y = "volatile.acidity", title = "Boxplot of Quality vs. volatile.acidity") + theme(legend.position = 'none', plot.title = element_text(size = 9, hjust=0.5))

g2


g3 <- ggplot(wine, aes(factor(quality), citric.acid, fill=factor(quality))) + geom_boxplot() +labs(x = "quality", y = "citric.acid", title = "Boxplot of Quality vs. citric.acid") + theme(legend.position = 'none', plot.title = element_text(size = 9, hjust=0.5))

g3


g4 <- ggplot(wine, aes(factor(quality), residual.sugar, fill=factor(quality))) + geom_boxplot() +labs(x = "quality", y = "residual.sugar", title = "Boxplot of Quality vs.
residual.sugar") + theme(legend.position = 'none', plot.title = element_text(size = 9, hjust=0.5))

g4


ggarrange(g1, g2, g3, g4, nrow = 2, ncol =2)

# It seems there’s positive relationship between citric acid and quality

# It seems there’s negative relationship between volatile acidity and quality


g5 <- ggplot(wine, aes(factor(quality), chlorides, fill=factor(quality))) + geom_boxplot() +labs(x = "Quality", y = "chlorides", title = "Boxplot of Quality vs. chlorides") + theme(legend.position = 'none', plot.title = element_text(size = 9, hjust=0.5))

g5


g6 <- ggplot(wine, aes(factor(quality), free.sulfur.dioxide, fill=factor(quality))) + geom_boxplot() +labs(x = "quality", y = "free.sulfur.dioxide", title = "Boxplot of quality vs. free.sulfur.dioxide") + theme(legend.position = 'none', plot.title = element_text(size = 9, hjust=0.5))

g6


g7 <- ggplot(wine, aes(factor(quality), total.sulfur.dioxide, fill=factor(quality))) + geom_boxplot() +labs(x = "quality", y = "total.sulfur.dioxide", title = "Boxplot of quality vs. total.sulfur.dioxide") + theme(legend.position = 'none', plot.title = element_text(size = 9, hjust=0.5))

g7


g8 <- ggplot(wine, aes(factor(quality), density, fill=factor(quality))) + geom_boxplot() +labs(x = "quality", y = "density", title = "Boxplot of quality vs. density") + theme(legend.position = 'none', plot.title = element_text(size = 9, hjust=0.5))

g8


ggarrange(g5, g6, g7, g8, nrow = 2, ncol =2)

# It seems there’s negative relationship between density acid and quality
```

```
g9 <- ggplot(wine, aes(factor(quality), pH, fill=factor(quality))) + geom_boxplot()
+labs(x = "quality", y = "pH", title = "Boxplot of Quality vs. pH") +
theme(legend.position = 'none', plot.title = element_text(size = 9, hjust=0.5))

g9


g10 <- ggplot(wine, aes(factor(quality), sulphates, fill=factor(quality))) +
geom_boxplot() +labs(x = "quality", y = "sulphates", title = "Boxplot of quality vs.
sulphates") + theme(legend.position = 'none', plot.title = element_text(size = 9,
hjust=0.5))

g10


g11 <- ggplot(wine, aes(factor(quality), alcohol, fill=factor(quality))) + geom_boxplot()
+labs(x = "quality", y = "alcohol", title = "Boxplot of quality vs. alcohol") +
theme(legend.position = 'none', plot.title = element_text(size = 9, hjust=0.5))

g11


ggarrange(g9, g10, g11, nrow = 2, ncol =2)



# 3.4. Inspecting 3D

i1<- ggplot(wine, aes(x=factor(round(alcohol)), y=citric.acid)) +
geom_boxplot(aes(colour = factor(quality))) +labs(title="Alcohol + Citric.Acid vs.
Quality") + theme(plot.title=element_text(hjust=0.5))

i1



i2 <- ggplot(wine, aes(x=factor(round(alcohol)), y=volatile.acidity)) +
geom_boxplot(aes(colour = factor(quality))) +labs(title="Alcohol + Volatile.Acidity vs.
Quality") + theme(plot.title=element_text(hjust=0.5))

i2
```

```
i3 <- ggplot(wine, aes(x=factor(round(alcohol)), y=chlorides)) + geom_boxplot(aes(colour = factor(quality))) +labs(title="Alcohol + Chlorides vs. Quality") + ylim(0, 0.3)+theme(plot.title=element_text(hjust=0.5))

i3



#MODELLING


# 4. Modeling

## 4.1. Modeling with top 5 variables


## 4.1. Model 1 with top 5 highest correlation with TotalIncidents


lm0 <- lm(quality ~ alcohol + volatile.acidity + sulphates + citric.acid + total.sulfur.dioxide, data = wine)

summary(lm0)


lm1 <- lm(quality ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide, data = wine)

summary(lm1)

lm0 <- lm(quality ~ alcohol + volatile.acidity + sulphates + citric.acid + total.sulfur.dioxide, data = wine)

summary(lm0)


lm1 <- lm(quality ~ alcohol + volatile.acidity + sulphates + total.sulfur.dioxide, data = wine)

summary(lm1)



#Using CV
```

```
# Define training control

set.seed(123)

train.control<- trainControl(method = "cv", number = 10) #cv Cross-Validation

# Train the model

model1 <- train(quality ~ alcohol + volatile.acidity +  sulphates + total.sulfur.dioxide,
data = wine, method = "lm", trControl = train.control)

# Summarize the results

summary(model1)

print(model1)



## 4.2. LASSO


library(glmnet)

x <- model.matrix(quality~., wine)[,-1]

y <- wine$quality

mod <- cv.glmnet(as.matrix(x), y, alpha=1)



#To see the coefficients with the minimum cross-validation error

#To see the coefficients with the largest value of lambda such that error is within 1
standard error of the minimum:


as.matrix(coef(mod, mod$lambda.min))

as.matrix(coef(mod, mod$lambda.1se))



#You can also select any other value of lambda that you want. Coefficients that are 0
have been dropped out of the model
```

```
CF <- as.matrix(coef(mod, mod$lambda.1se))

CF[CF!=0,]



#Using Model 2 using above independent variables


lm2 <- lm(quality ~ fixed.acidity + volatile.acidity + chlorides + total.sulfur.dioxide +
sulphates + alcohol, data=wine)

summary(lm2)


#Using CV


# Define training control

set.seed(123)

train.control<- trainControl(method = "cv", number = 10) #cv Cross-Validation

# Train the model

model2<- train(quality ~ fixed.acidity + volatile.acidity + chlorides +
total.sulfur.dioxide + sulphates + alcohol,data = wine, method = "lm",

trControl = train.control)

# Summarize the results

summary(model2)

print(model2)


## 4.3 Random Forest Model 3


library(randomForest)

library(mlbench)

library(caret) # use createDataPartition() function

# partition
```

```
#Create Evaluation Sets
set.seed(123)
n = nrow(wine)
trainIndex = sample(1:n, size = round(0.7*n), replace=FALSE)

#Creates training and test set from observations
training = wine[trainIndex,]
testing = wine[-trainIndex,]
model3 <- randomForest(quality ~ ., training, mtry = 3,
                       importance = TRUE, na.action = na.omit)
print(model3)
#Plot the error vs the number of trees graph
plot(model3)

varImp(model3)
varImpPlot(model3,type=2)



# obtain MSE as of last element in fit$mse
# which should match the output from printout
model3$mse[length(model3$mse)]
# take square root to calculate RMSE for the model
sqrt(model3$mse[length(model3$mse)])



# now illustrate how to calculate RMSE on test data vs. training data
predValues<- predict(model3,testing)
```

```
# we can calculate it directly
sqrt(mean((testing$quality -predValues)^2)) #RMSE
mean(abs(testing$quality -predValues)) #MAE



#Evaluation dataframe


Model <- c("Model 1", "Model 2", "Model 3")
R_squared<- c(0.3479, 0.3546, 0.4850)
RMSE <- c(0.6549, 0.6515, 0.5843)
MAE <- c(0.5092899, 0.5063, 0.4222)
ml <- data.frame(Model, R_squared, RMSE, MAE)



#Plot


library(gridExtra)
p1 <- ggplot(ml, aes(Model, RMSE)) + geom_point(aes(colour = factor(Model), size = 4)) + labs(title="RMSE") + theme(plot.title=element_text(hjust=0.5), axis.title.y = element_blank(),axis.title.x = element_blank(), legend.position="none")
p2 <- ggplot(ml, aes(Model, R_squared)) + geom_point(aes(colour = factor(Model), size = 4)) + labs(title="R-Squared") + theme(plot.title=element_text(hjust=0.5), axis.title.y = element_blank(),axis.title.x = element_blank(), legend.position="none")
p3 <- ggplot(ml, aes(Model, MAE)) + geom_point(aes(colour = factor(Model), size = 4)) + labs(title="MAE") + theme(plot.title=element_text(hjust=0.5), axis.title.y = element_blank(),axis.title.x = element_blank(), legend.position="none")
grid.arrange(p2,p1,p3, ncol=3)



## 4.4. Clustering
```

```
#Cleaning data

df_new<- wine
df_new$quality_group<- ifelse(df_new$quality< 5, "1", ifelse((df_new$quality>= 5) & (df_new$quality<= 6), "2", "3"))
df_new[,12]<-NULL # Order column looks like meaningless
str(df_new)


df_new$quality_group<- as.numeric(df_new$quality_group)
str(df_new)
View(df_new)

library(tidyverse) # data manipulation
library(cluster)    # clustering algorithms
library(factoextra) # clustering algorithms & visualization


#Standardize data

scale(df_new)
head(df_new)


#Elbow Method

library(factoextra)
library(NbClust)
```

```
set.seed(123)
a1 = fviz_nbclust(df_new, kmeans, method = "wss")
a1



#Using K-means

set.seed(123)
km <- kmeans(df_new, 2, nstart = 25)
df_new$cluster = km$cluster
View(df_new)
ggplot(df_new,aes(pH,alcohol,color=as.factor(cluster)))+geom_point()
ggplot(df_new,aes(pH,sulphates,color=as.factor(cluster)))+geom_point()
ggplot(df_new,aes(pH,total.sulfur.dioxide,color=as.factor(cluster)))+geom_point()
ggplot(df_new,aes(sulphates,total.sulfur.dioxide,color=as.factor(cluster)))+geom_point()
ggplot(df_new,aes(alcohol,sulphates,color=as.factor(cluster)))+geom_point()
ggplot(df_new,aes(alcohol,volatile.acidity,color=as.factor(cluster)))+geom_point()
```

REFERENCES

Multivariate Methods Based Soft Measurement for Wine Quality Evaluation

- https://www.hindawi.com/journals/aaa/2014/740754/

Improving Wine Data, Meredith Galante, Jan 28, 2020, Dimensional Insight:

- *https://www.dimins.com/blog/2020/01/28/improving-wine-data/*

Multivariate Statistical Analysis Applied in Wine Quality Evaluation:

- *https://www.researchgate.net/publication/282982301_Multivariate_Statistical_Analysis_Applied_in_Wine_Quality_Evaluation*

Random Forest Classification in prediction of best quality wine, Sailaja Konda, Apr 13 2020, Medium

- *https://medium.com/@sailajakonda2012/random-forest-classification- in-prediction-of-best-quality-wine-d0d7591a7c17*

Analysis of Wine Quality Data

- *https://online.stat.psu.edu/stat508/book/export/html/804*

WHAT WE DID TO MAKE OUR PROJECT PUBLIC

GitHub Link: https://github.com/Lallith-Prasath/Wine-Quality-Analysis.git