THE TRUTH ABOUT VINHO VERDE WINES

Correlations, regression and classification

Susana Reche Rodríguez
Student Number: *17165628*
Date: *27.04.2020*

Programming for Big Data

Higher Diploma in Data Analytics

Table of Contents

Table of Figures

Table of Tables

1. Executive Summary

Wine exists since 6,000 BC but still remains a mystery to discover. Some of the elements that influence wine are the terroir, the weather, the grapes used, the way the vineyards have been taking care, the type of methodologies used when producing the wine, the vintage, and many more. Even though some of them can be analyzed not all the characteristics of wine can be predicted.

"The truth about Vinho Verde wines" aspiration is to analyze a set of chemical compounds found in Vinho Verde wines with the aim to see if there are differences between red and white wines, qualities or if it exists any kind of relation between them. See a brief introduction to Vinho Verde in [27, Appendix I].

There is a first exploratory approach to the datasets, to understand which kind of wines are represented in them.

Once the wines in the data sets have been understood, the next step is to perform statistical tests to verify if there are differences between wines by type / quality.

The last step is trying to create some models to classify the wines or predict wines properties by using the datasets.

Note: *The 2,500 words allowed are accounted from Summary to Conclusion. Appendix, Code, Bibliography, Table of Content, Figures, and Tables are not included in the word count.*

2. Approach and Objectives

The research has a quantitative approach. As seen in Fig. 22, the data will need to be first prepared for the analysis. Then different hypotheses are formulated and tested (statistical tests, parametric or not depending on the nature of the distributions).

The main objectives of the research are:

- to identify if there is any difference between white and red wines based on their chemical properties (Insight 1),
- to identify if there is any difference between qualities based on their chemical properties (Insight 2),
- to assess the existence of a correlation between different chemical properties and if it is possible the creation of a simple linear model (Insight 3),
- to test if it is possible to classify the wines by type and/or quality (Insight 4).

In order to accomplish with those objectives Vinho Verde related data sets were researched, found in [1] [2] [3].

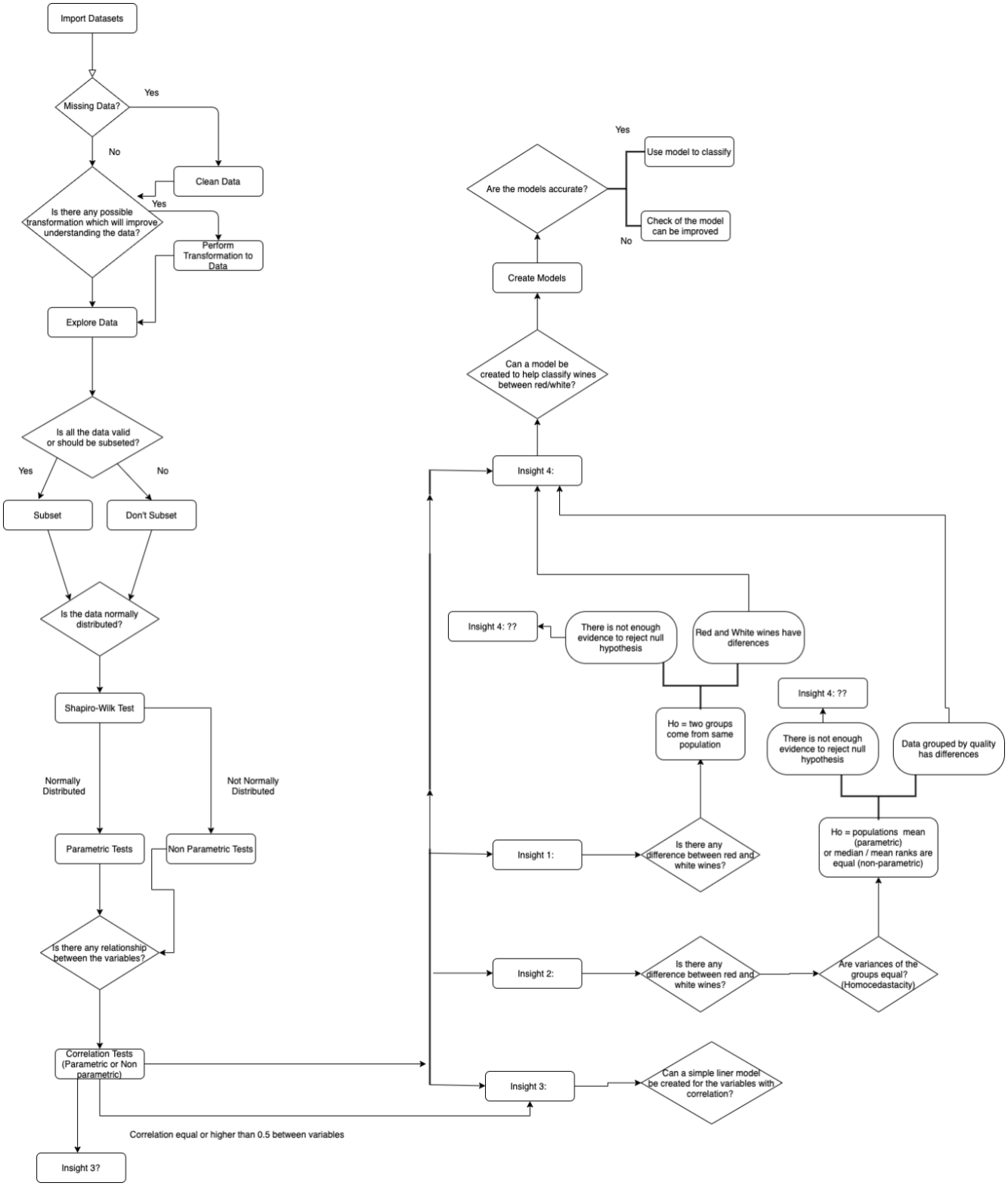


Figure 1: Research Approach Flowchart

3. Data Exploration

The datasets contains 1,599 records for red Vinho Verde wine and 4,898 records for white Vinho Verde wine, as illustrated in Fig. 3. Both data sets contain the same variables (chemical compounds), shown in Fig. 2. Chemical compounds explained in more detail in [27, Appendix II].

The datasets do not contain missing values or any other inconstancies.

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)

Figure 2: Variables Included on the Datasets, as seen in [2]

```
> str(redwine)
'data.frame':   1599 obs. of  12 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 ...
 $ class               : int  5 5 5 6 5 5 5 7 7 5 ...
> str(whitewine)
'data.frame':   4898 obs. of  12 variables:
 $ V1   : num  7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
 $ V2   : num  0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
 $ V3   : num  0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
 $ V4   : num  20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
 $ V5   : num  0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
 $ V6   : num  45 14 30 47 47 30 30 45 14 28 ...
 $ V7   : num  170 132 97 186 186 97 136 170 132 129 ...
 $ V8   : num  1.001 0.994 0.995 0.996 0.996 ...
 $ V9   : num  3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...
 $ V10  : num  0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
 $ V11  : num  8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
 $ Class: int  4 4 4 4 4 4 4 4 4 4 ...
```

Figure 3: Structure of raw datasets

Each data set contain 11 numerical variables referring to the chemical compounds and 1 integer related to the quality level.

3.1. Cleaning Data – Data Transformation

Some adjustments were needed to make the data more meaningful:

- to use same variable names in both datasets,
- to convert integer variable to factor with levels,

- to create a new variable **Type** as a factor (red, white),
- to create a new variable **Type by Sugar Level** as a factor with levels, according to [4],
- to create a new numerical variable **Total Acidity** (Fixed Acidity + Volatile Acidity),
- to create a new variable **PH Level** as a factor with levels,
- To create a new variable **Alcohol Level** as a factor with levels, according to [5].

After the transformation, the datasets contain 12 numerical variables and 5 factors (4 of them with ordered levels), as illustrated in Fig. 4.

```
> str(redwine_renamed)
'data.frame':   1599 obs. of  17 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             : Factor w/ 10 levels "1","2","3","4",..: 5 5 5 6 5 5 5 7 7 5 ...
 $ type                : Factor w/ 1 level "red": 1 1 1 1 1 1 1 1 1 1 ...
 $ type_by_sugar_level : Factor w/ 4 levels "Dry","Medium Dry",..: 1 1 1 1 1 1 1 1 1 2 ...
 $ total_acitity       : num  8.1 8.68 8.56 11.48 8.1 ...
 $ ph_level            : Factor w/ 5 levels "PH<=2.8","2.8>PH<=3",..: 4 3 3 3 4 4 3 3 3 3 ...
 $ alcohol_level       : Factor w/ 4 levels "Very Low","Moderately Low",..: 1 1 1 1 1 1 1 1 1 1 ...
> str(whitewine_renamed)
'data.frame':   4898 obs. of  17 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             : Factor w/ 10 levels "1","2","3","4",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ type                : Factor w/ 1 level "white": 1 1 1 1 1 1 1 1 1 1 ...
 $ type_by_sugar_level : Factor w/ 4 levels "Dry","Medium Dry",..: 3 1 2 2 2 2 2 3 1 1 ...
 $ total_acitity       : num  7.27 6.6 8.38 7.43 7.43 8.38 6.52 7.27 6.6 8.32 ...
 $ ph_level            : Factor w/ 5 levels "PH<=2.8","2.8>PH<=3",..: 2 3 3 3 3 3 3 2 3 3 ...
 $ alcohol_level       : Factor w/ 4 levels "Very Low","Moderately Low",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Figure 4: Structure of transformed datasets

3.2. About Vinho Verde (datasets)

Different graphical representations have been used to visualize the type of wines that are present in the data sets.

- **Quality:** As seen in Fig. 5, the average quality contained in the datasets is quite low for both white (average quality 3-4) and red (average quality 5-6) wines.

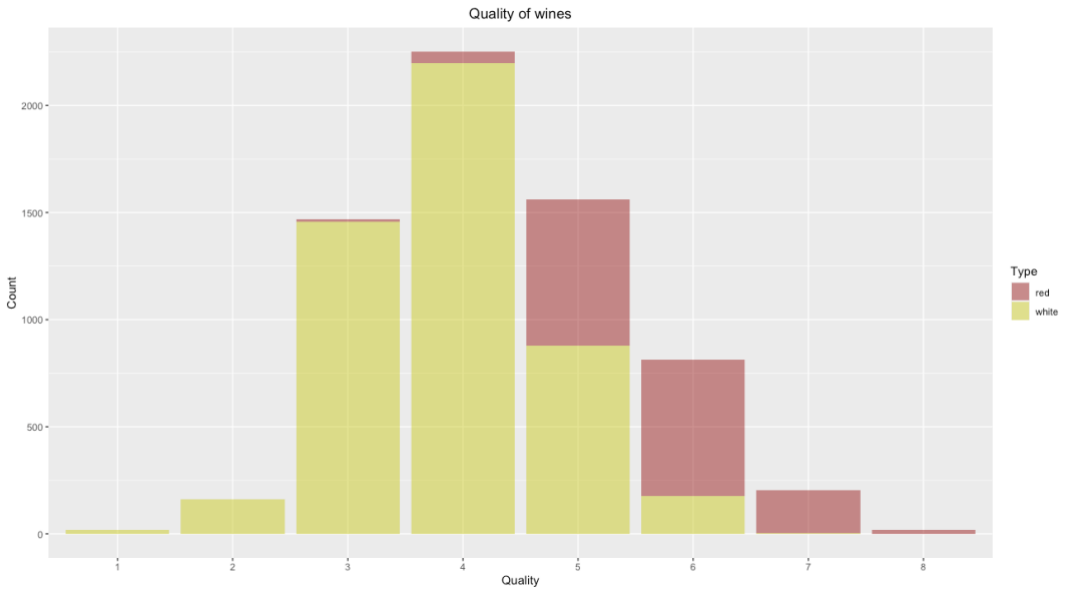


Figure 5: Quality of wines contained in the dataset

- **Total Acidity (mg/l) and PH:** Vinho Verde is well known for being fresh wines with high acidity, which matches with the information presented.

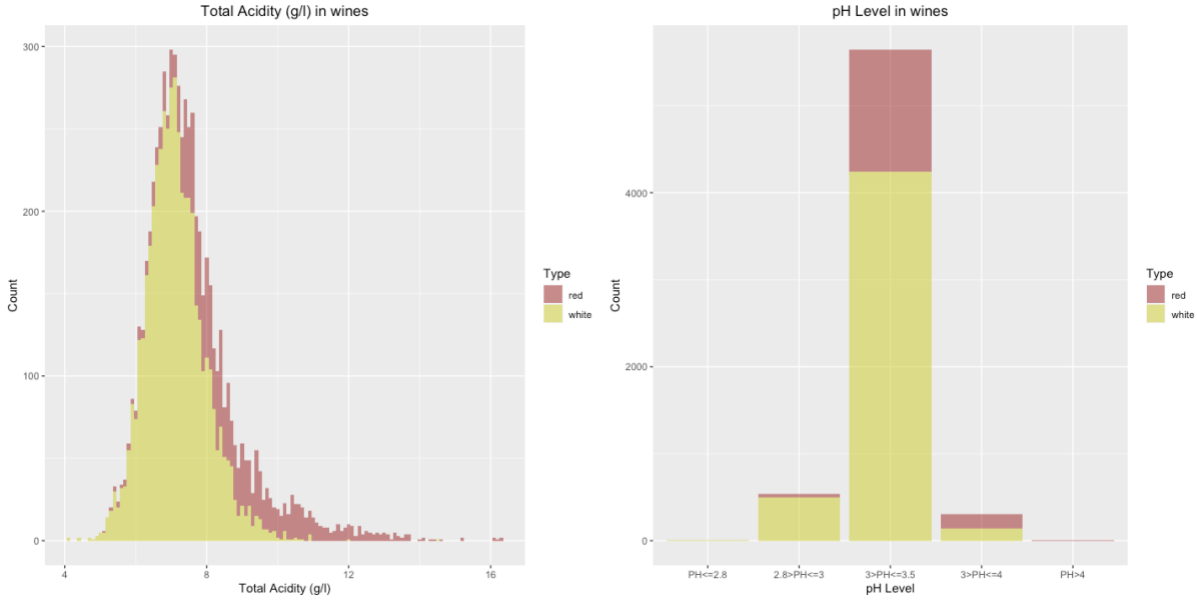


Figure 6: Total Acidity (g/l) and Level of pH in wines contained in the dataset

Some of the red wines seem to have a high volatile acidity (>0.7) which could make the wines taste vinegary.

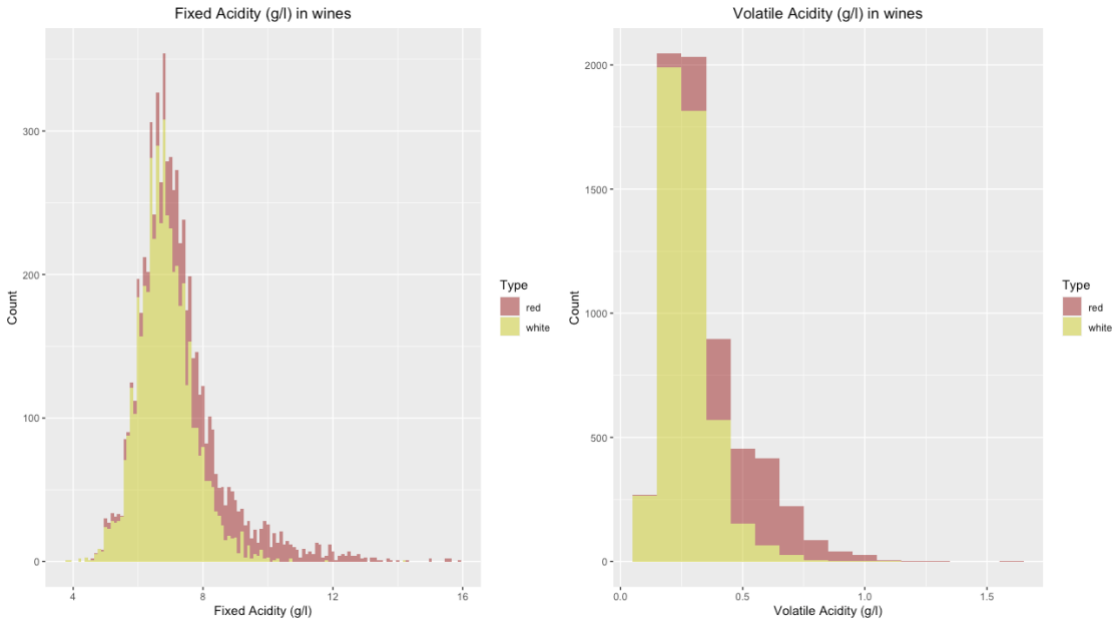


Figure 7: Fixed and Volatile Acidity (g/l) in wines contained in the dataset

- Sulphates (g/l), Total Sulfur Dioxide (mg/l) and Free Sulfur Dioxide (mg/l): Most wines seem to have a low level of sulphates which is good.

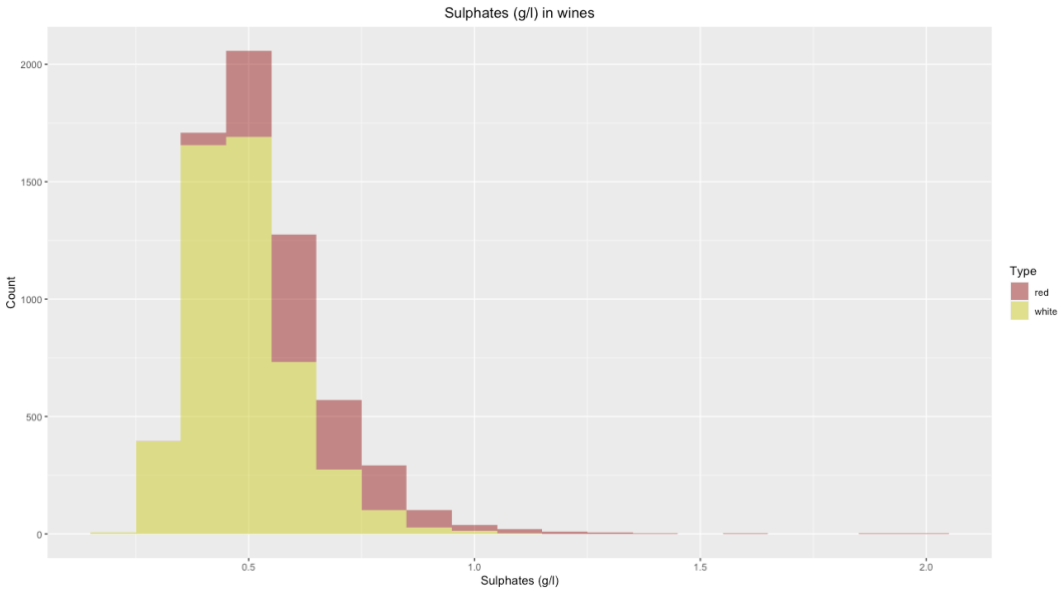


Figure 8: Sulphates (g/l) in wines contained in the dataset

The wines seem to follow the standard in relation to the amount of total sulfur dioxide. However, many white wines seem to have a higher amount (>35gr/l) of free sulfur dioxide than recommended, which can be contributing to the observed low quality.

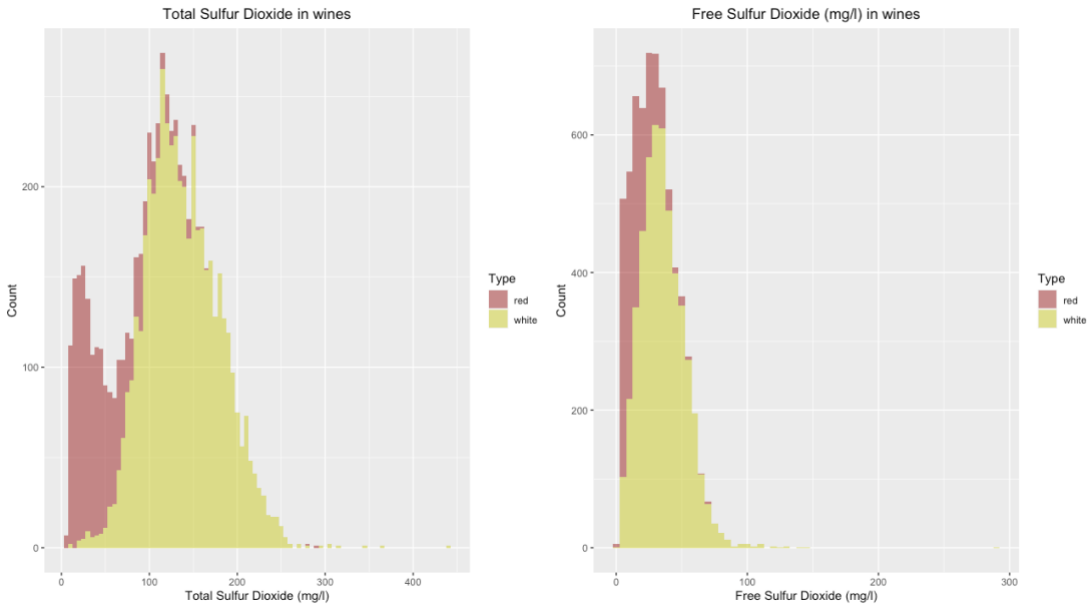


Figure 9: Total and Free Sulfur Dioxide (mg/l) in wines contained in the dataset

- **Type by Sugar Level, alcohol, chloride, and density:** Most the of wines on the dataset are dry or medium-dry, and especially the red wines are mostly dry.

  The level of alcohol in Vinho Verde is around 8.5 and 11, as it is considered a low-level alcohol wine. However, when using Alvarinho grape the alcohol goes higher to 11.5 to 14%, as seen in[6]. The higher levels of alcohol in some wines might indicate the use of Alvarinho grapes.

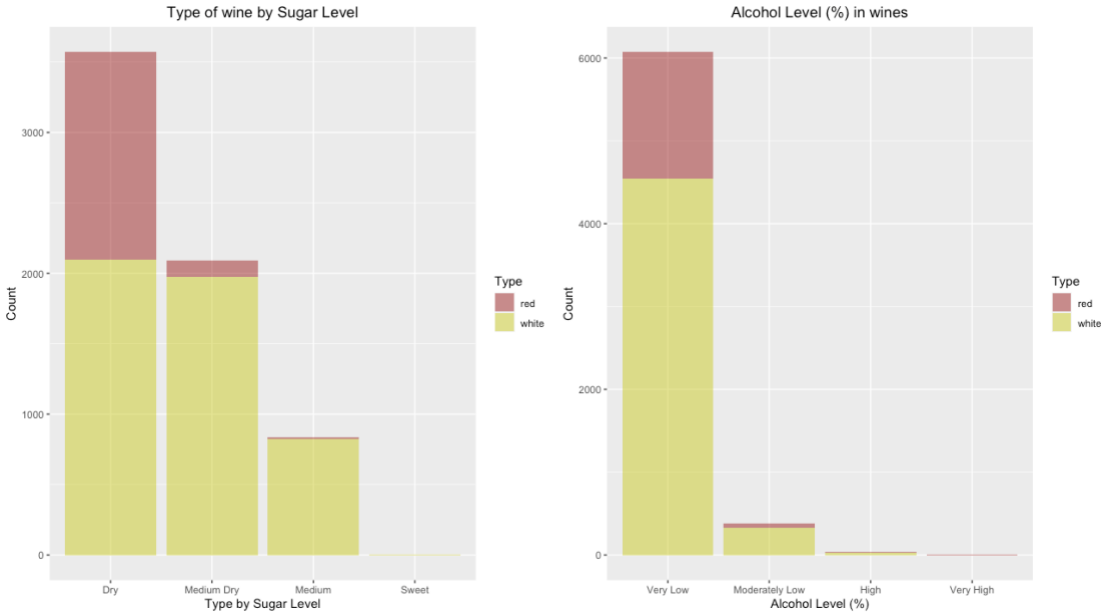


Figure 10: Type of wines by Sugar Level & Alcohol % in wines contained in the dataset

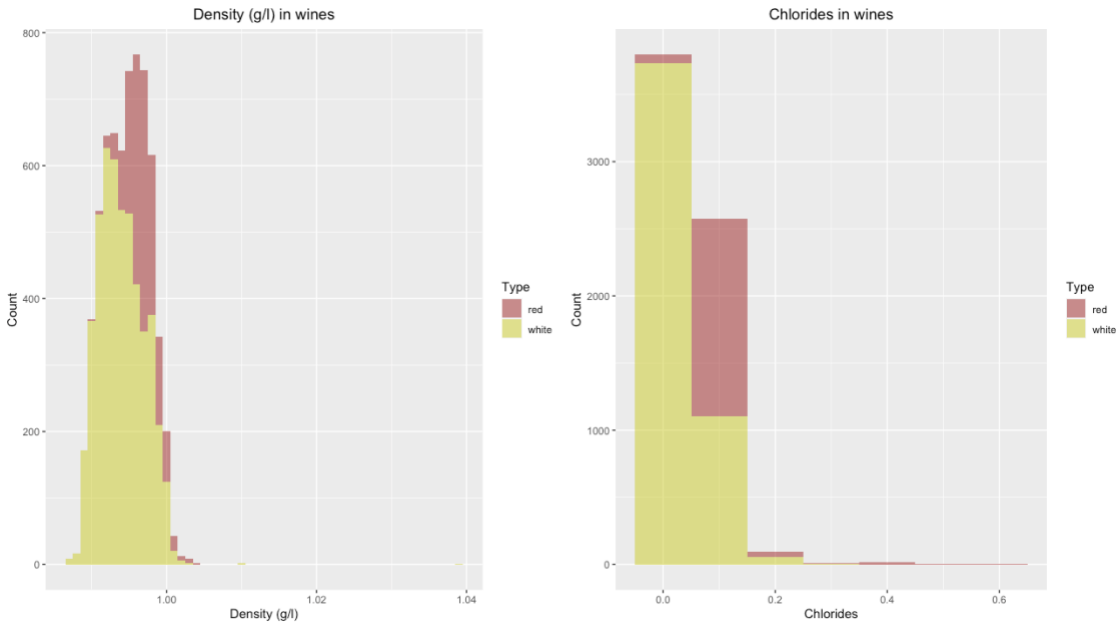


Figure 11: Density (g/l) & Chlorides in wines contained in the dataset

3.3. About dry Vinho Verde (data set)

As sweet white wine would have nothing to do in taste with dry red wine and comparing them will be like comparing oranges with apples, from this point only dry wines will be analyzed.

The dataset for dry wines contains 1,474 observations for red wine and 2,097 for white wine. A very similar pattern can be observed in dry wines compare to the complete data set.

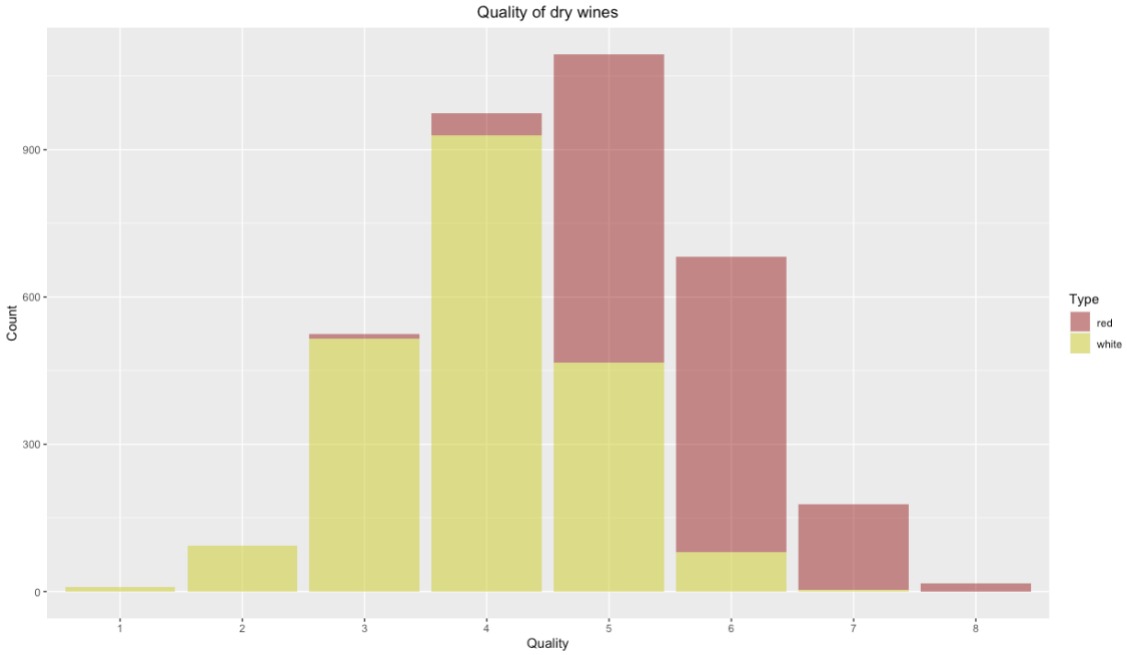


Figure 12:Quality of dry wines contained in the dataset

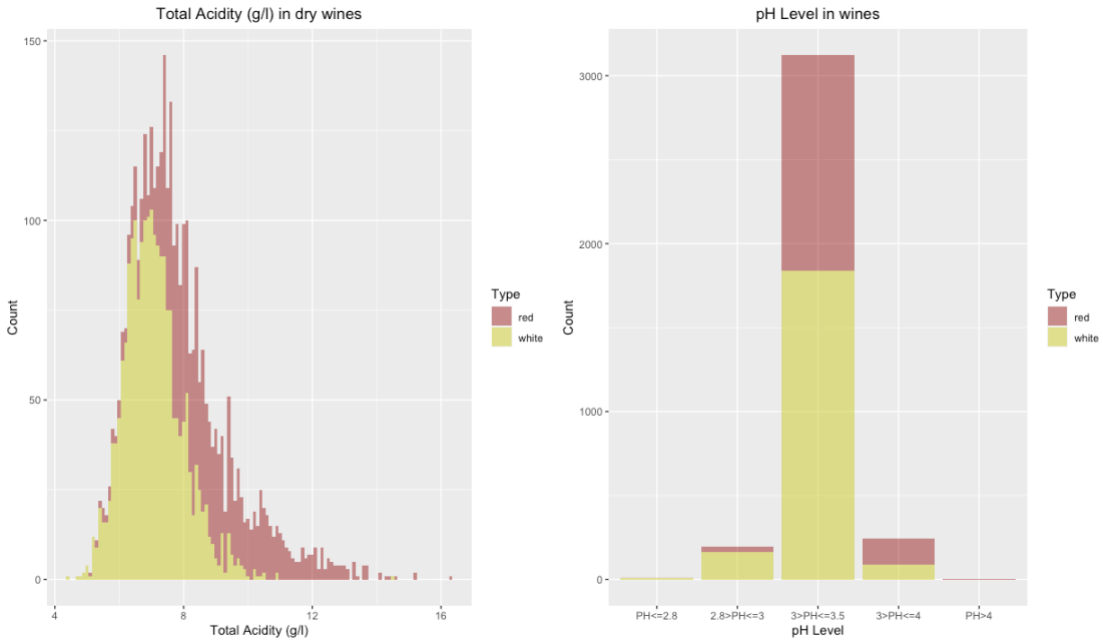


Figure 13: Fixed Acidity (g/l) and Level of pH in dry wines contained in the dataset

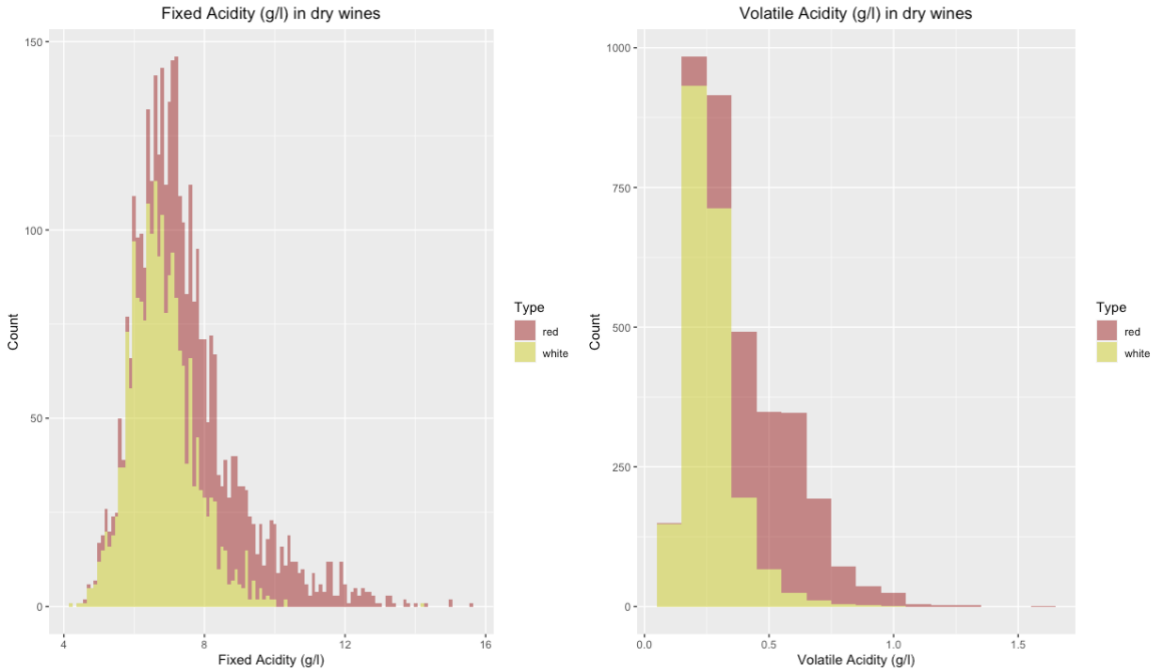


Figure 14: Fixed and Volatile Acidity (g/l) in dry wines contained in the dataset

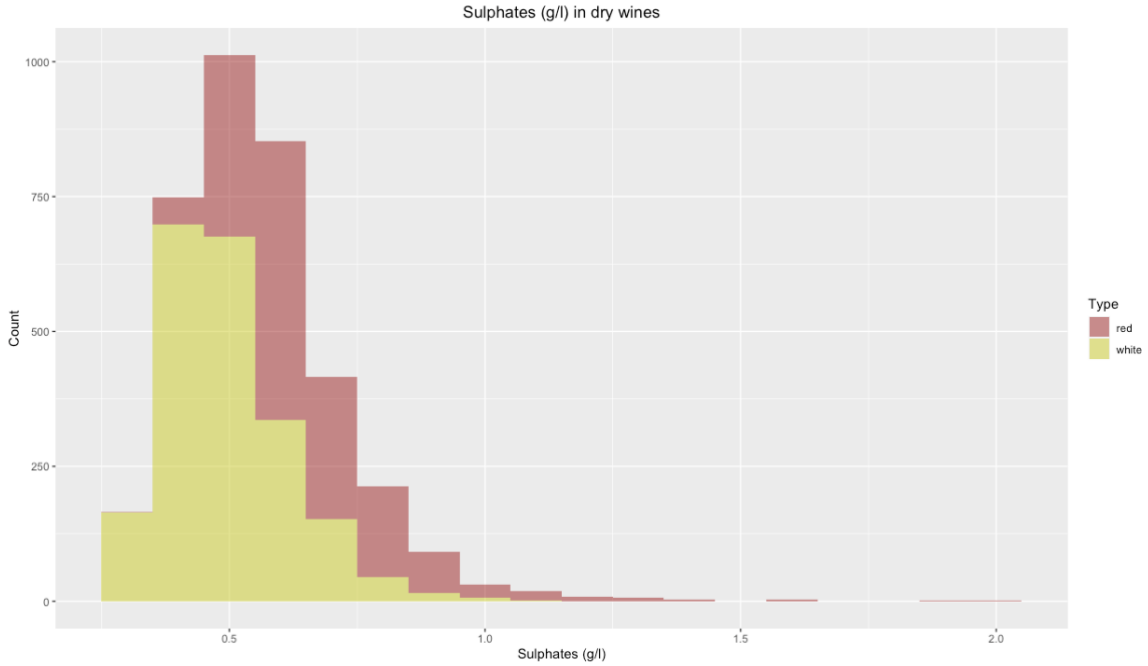


Figure 15: Sulphates (g/l) in dry wines contained in the dataset

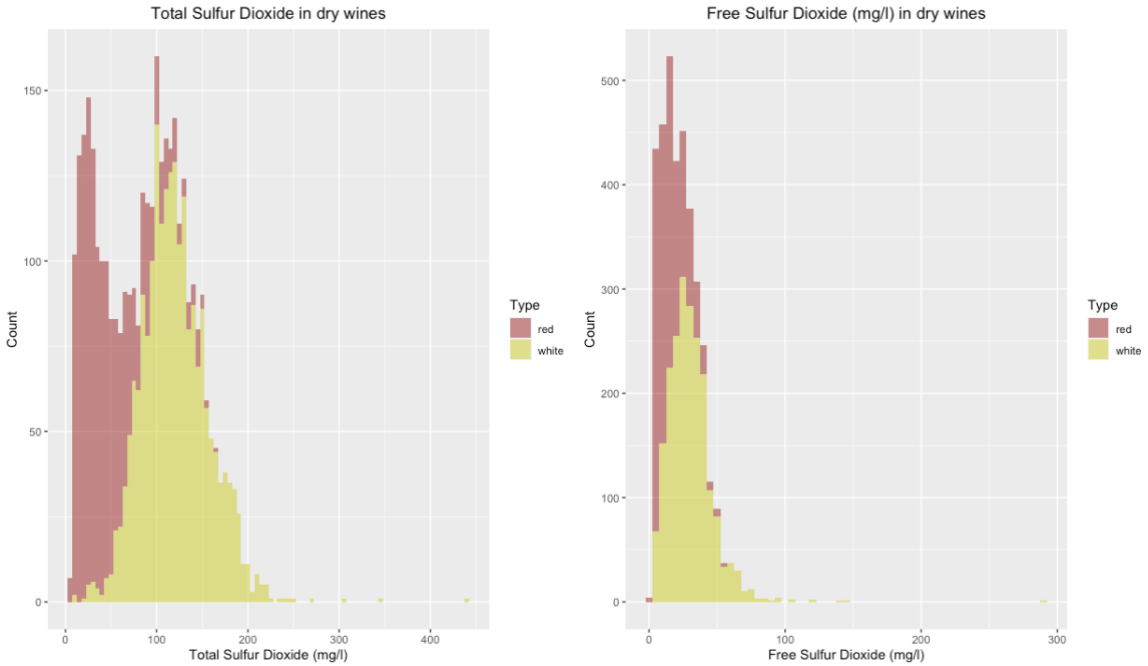


Figure 16: Total and Free Sulfur Dioxide (mg/l) in dry wines contained in the dataset

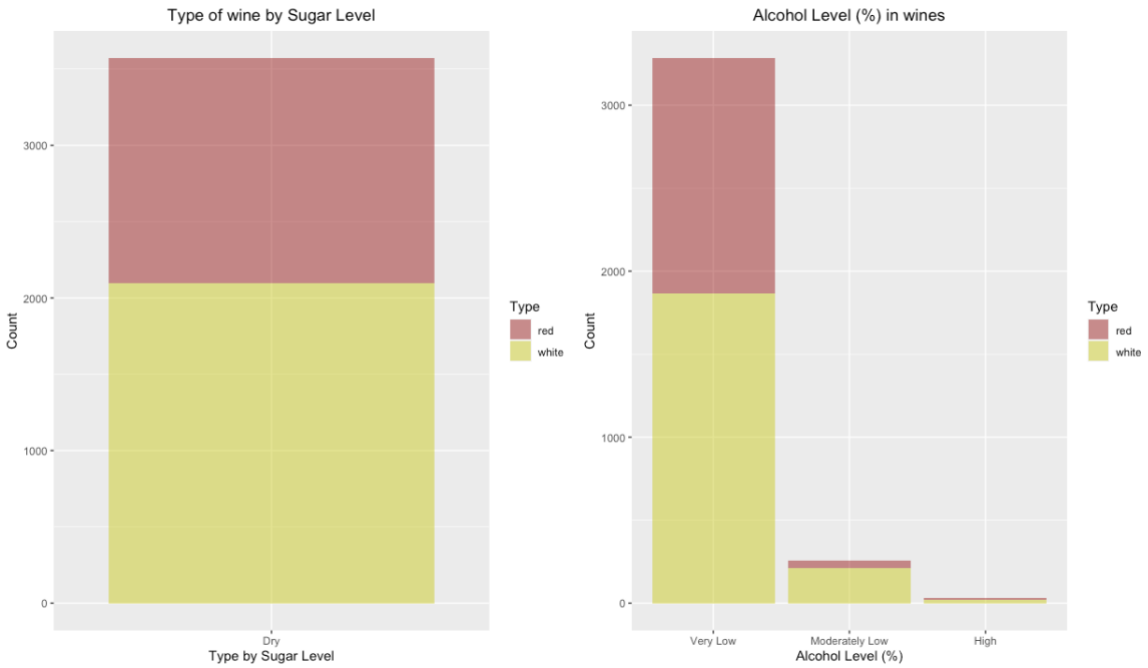


Figure 17:Type of wines by Sugar Level & Alcohol % in dry wines contained in the dataset

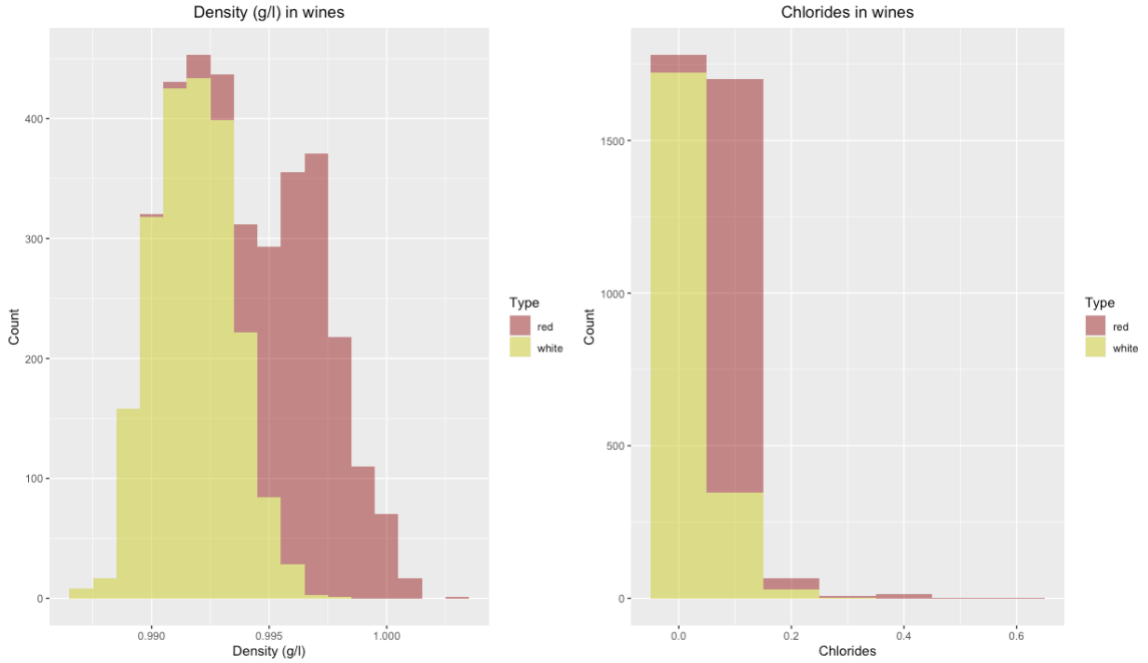


Figure 18: Density (g/l) & Chlorides in dry wines contained in the dataset

3.3.1. Normality Test

Assuming that red and white wines are different (hypothesis tested on Insight 1), the normality test is performed independently for each of them, for each of the variables.

By observing the QQ Plots in Fig. 19 and 20, the variables seem to pretty much follow the theoretical normal line except in the extremes, which will indicate normality.

However, the Shapiro-Wilk normality tests results indicate a **non-normal distribution**. For detailed tests information see [30, Appendix III].

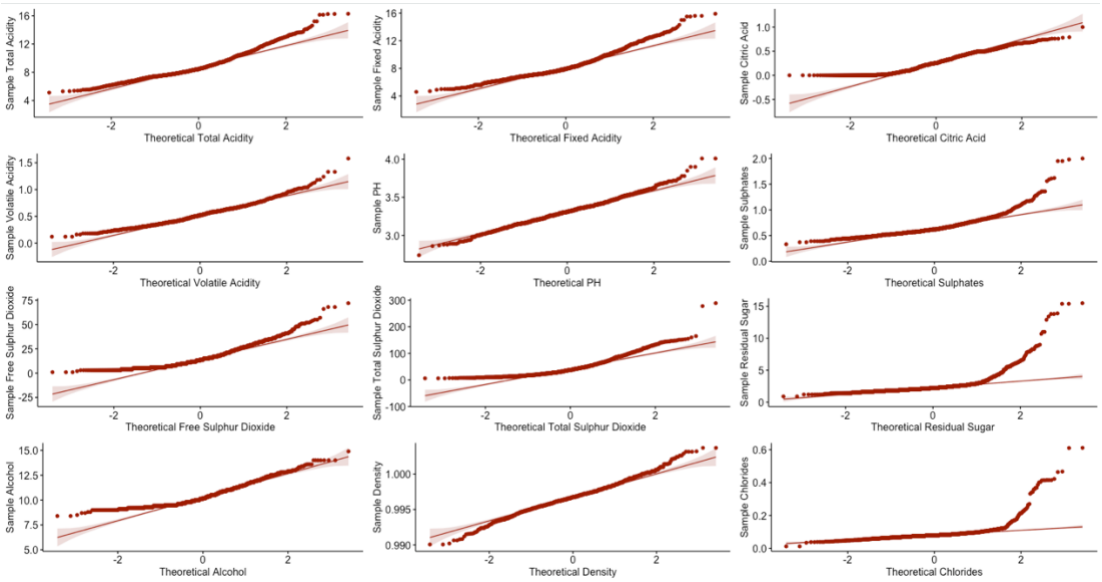


Figure 19: QQ Plots dry red wines

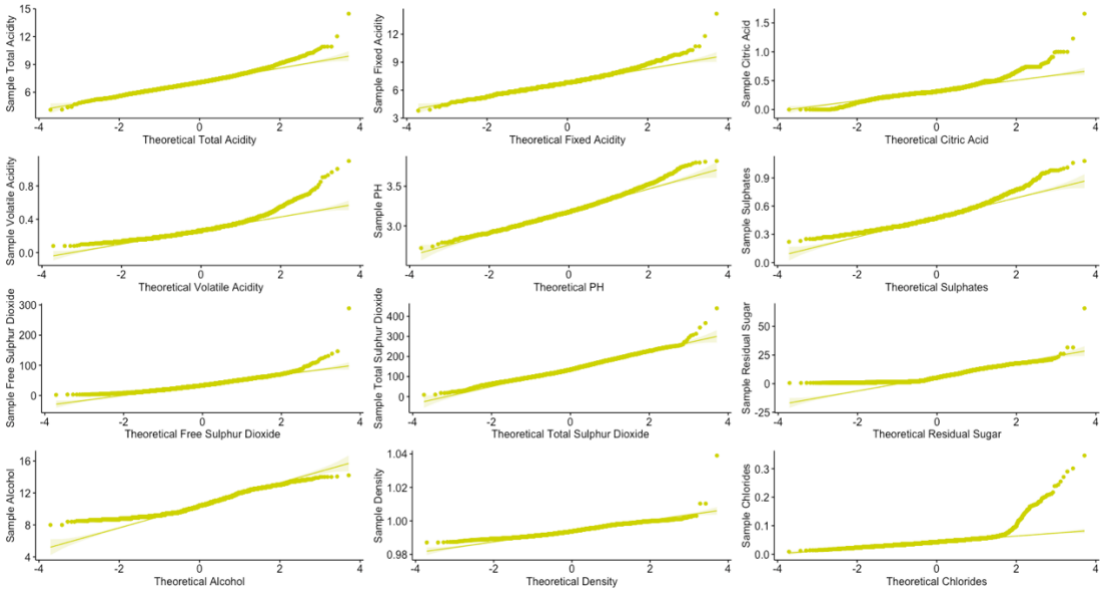


Figure 20: QQ Plots dry white wines

3.3.2. Correlations

For dry red wines, as seen in Fig. 21, a higher correlation than 0.5 are observed in:

- Fixed Acidity – Total Acidity (positive corr.)
- Free Sulfur Dioxide – Total Sulfur Dioxide (positive corr.)
- Total Acidity – pH (negative corr.)
- Fixed Acidity – pH (negative corr.)

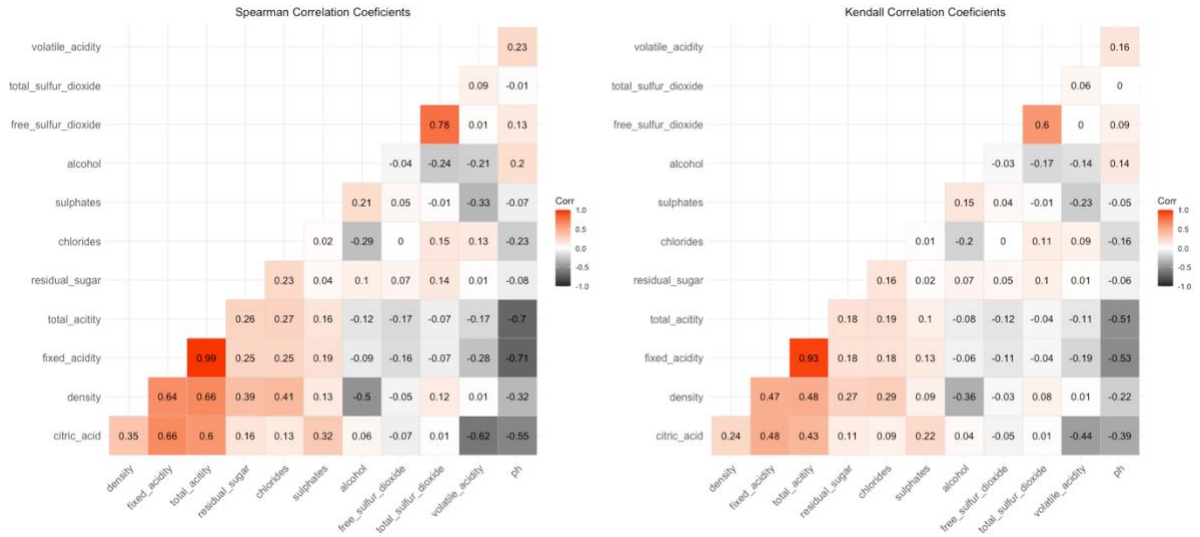


Figure 21: Spearman and Kendall Correlation Coefficients between numerical variables for dry red wines

For dry white wines, as seen in Fig. 21, a higher correlation than 0.5 are observed in:

- Fixed Acidity – Total Acidity (positive corr.)
- Density – Alcohol negative (negative corr.)

Figure 22: Spearman and Kendall Correlation Coefficients between numerical variables for dry white wines

4. Insights

4.1. Insight 1 – Is any difference between white and red Vinho Verde wines based on their chemical properties?

As maybe quality can influence the chemical properties observed on a wine (hypothesis tested on Insight 2), only quality 5 will be analyzed. The new subset contains 628 observations for red wine and 466 observations for white wine.

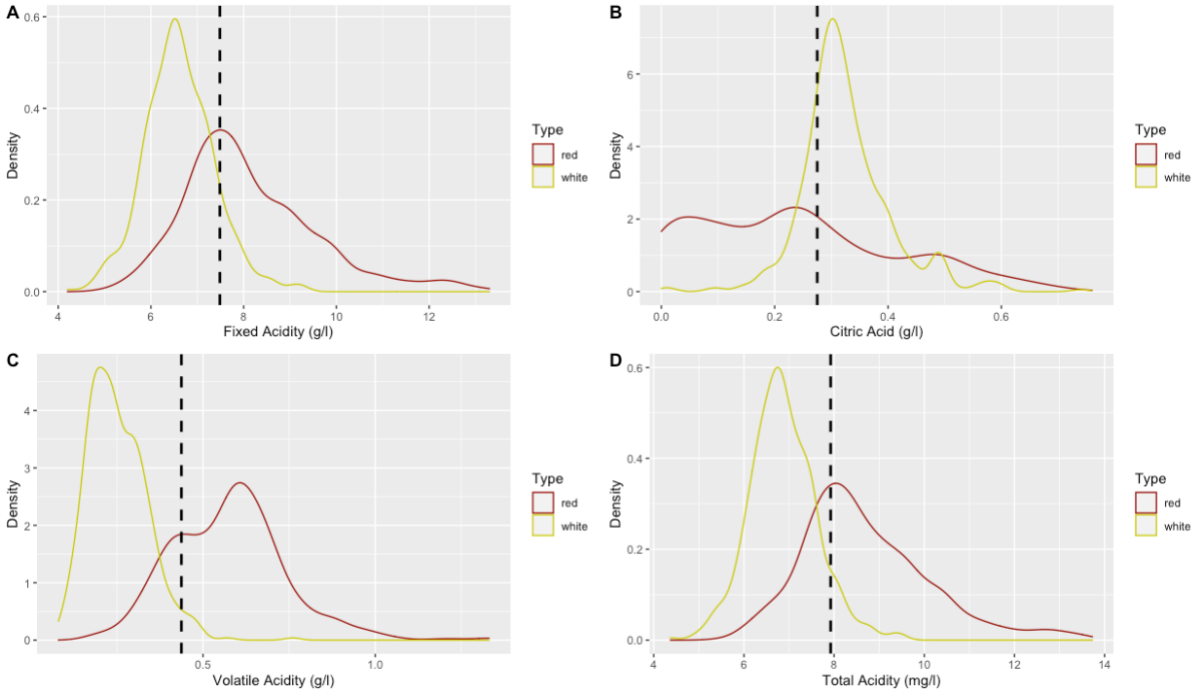


Figure 23: Fixed Acidity, Citric Acid, Volatile and Total Acidity density plots for dry wines with quality 5

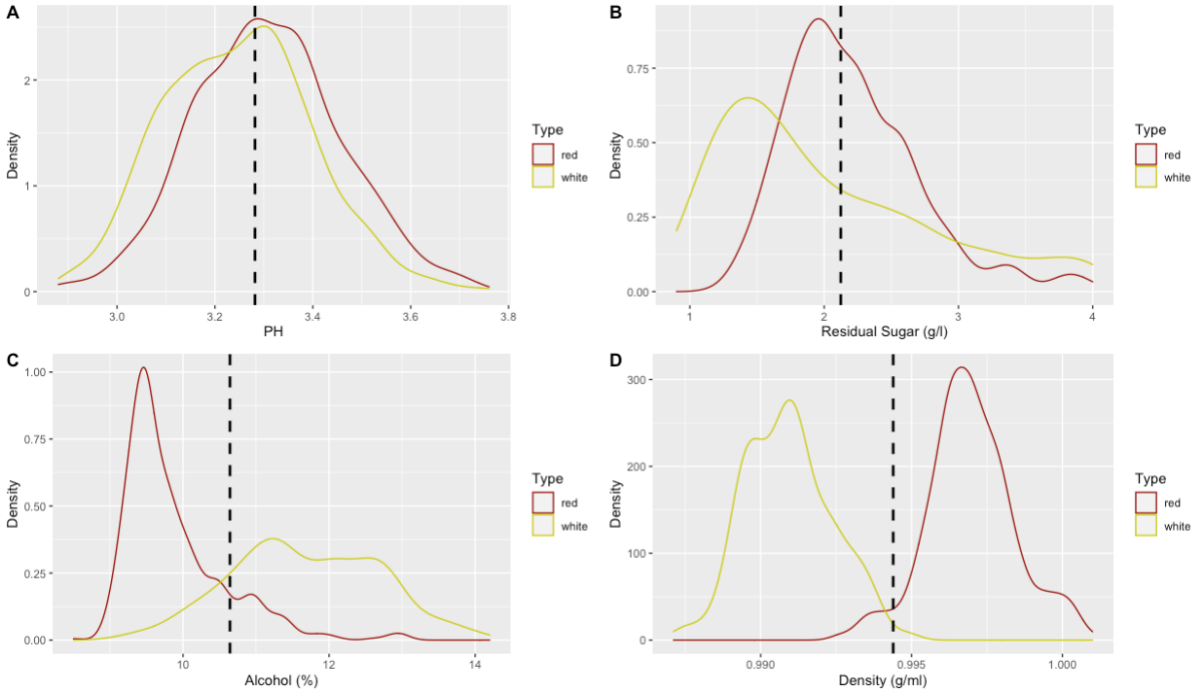


Figure 24: pH, Residual Sugar), Alcohol and Density density plots for dry wines with quality 5

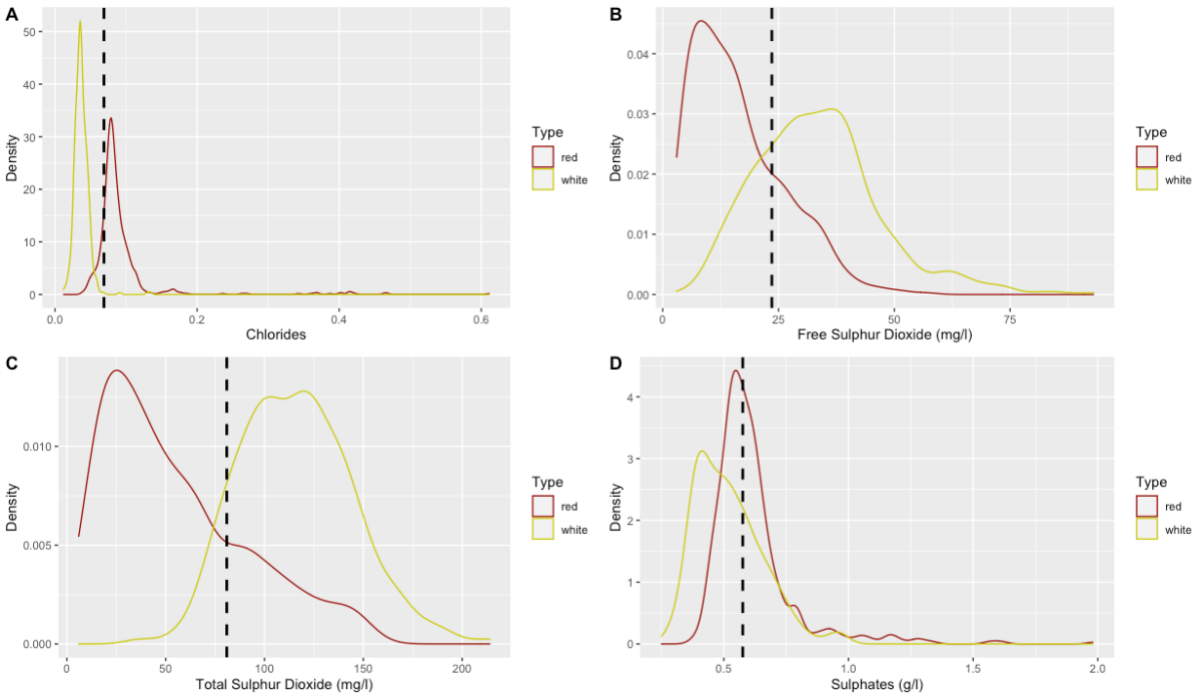


Figure 25: Chlorides, Free Sulfur and Total Sulfur Dioxide and Sulphates density plots for dry wines with quality 5

As seen in previous Fig. 23, 24, and 25, the variables have a positive skewness distribution, which confirms the Shapiro-Wilk normality test results of non-normality. Tab. 1, shows a high level of skewness for most variables, according to [7]. Complete descriptive statistics in [32, Appendix IV].

	Red Wine Skewness in Variable Distribution	White Wine Skewness in Variable Distribution
Fixed Acidity	0.9479009	0.3178317
Volatile Acidity	0.625286	0.9393171
Citric Acid	0.5406596	0.7150978
Residual Sugar	0.98784	0.885498
Chlorides	5.3611	3.354095
Free Sulfur Dioxide	0.9298675	0.8312963
Total Sulfur Dioxide	0.8567435	0.3465593
Density	0.07299805	0.2260654
pH	0.1177582	0.2140264
Sulphates	3.021923	0.8662798
Alcohol	1.627627	0.02077431
Total Acidity	0.8735447	0.3315459

Table 1: Skewness in variables distribution for dry wines with quality 5

The non-parametric Mann-Whitney U test has been used to test the null hypothesis of the populations (dry red and white wines with quality 5) being equal. The result is that we can reject the null hypothesis. Fig. 26, 27, and 28 show those differences between red and white wines with quality 5.

Mann-Whitney U test's p-values are shown in each of the box plots. For detailled Mann-Whitney U test see [33, Appendix IV].

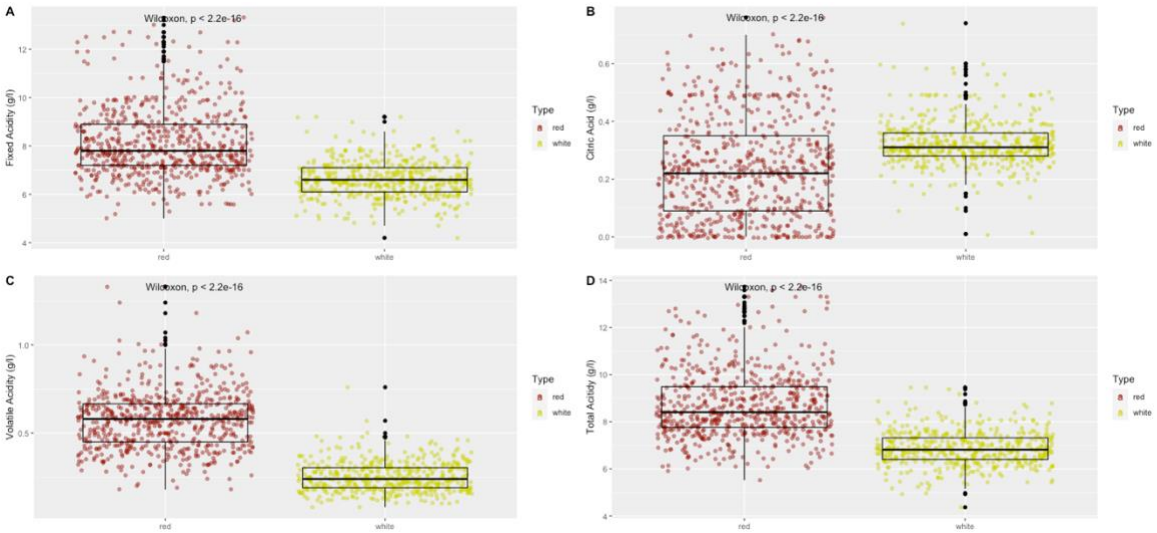


Figure 26: Box plots for Fixed Acidity, Citric Acid, Volatile and Total Acidity for dry wines with quality 5

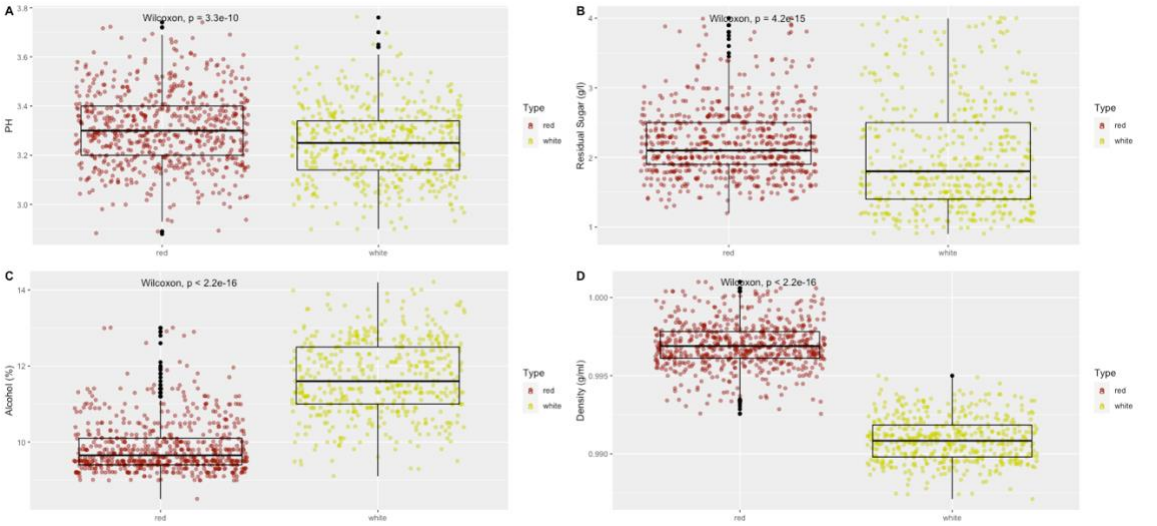


Figure 27: Box plots for pH, Residual Sugar, Alcohol and Density for dry wines with quality 5

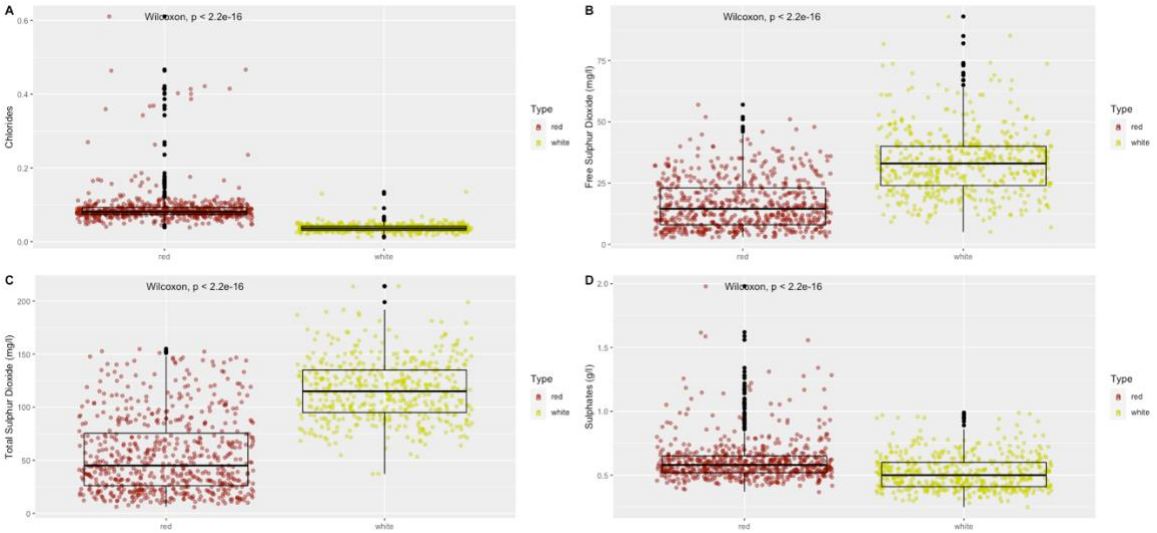


Figure 28: Box plots for Chlorides, Free and Total Sulfur Dioxide and Sulphates for dry wines with quality 5

4.2. Insight 2 – Is any difference between dry red Vinho Verde wines' quality based on their chemical properties?

The non-parametric Kruskal-Wallis H test reveals that there are differences between variables based on quality. For detailed information about the Kruskal-Wallis H test see [37, Appendix V].

However, it seems that the residual sugar, as shown in Fig. 30, doesn't show divergences for different levels of quality. But the variances of the different groups by quality for residual sugar are not equal, which is one of the Kruskal-Wallis H assumptions. For detailed Fligneen-Killeen test see [35, Appendix V].

The Kruskal Test was followed by Dunn's test to find out which are the groups with differences. See detailed results in [40, Appendix V].

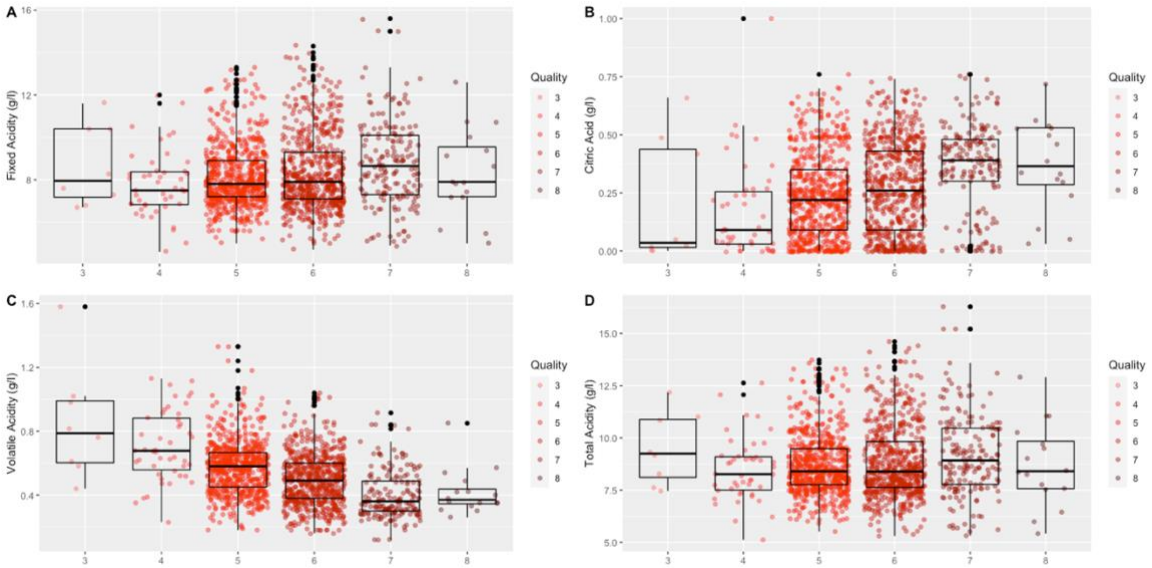


Figure 29: Box plot of Fixed Acidity, Acid Citric, Volatile and Total Acidity by quality (dry red wines)

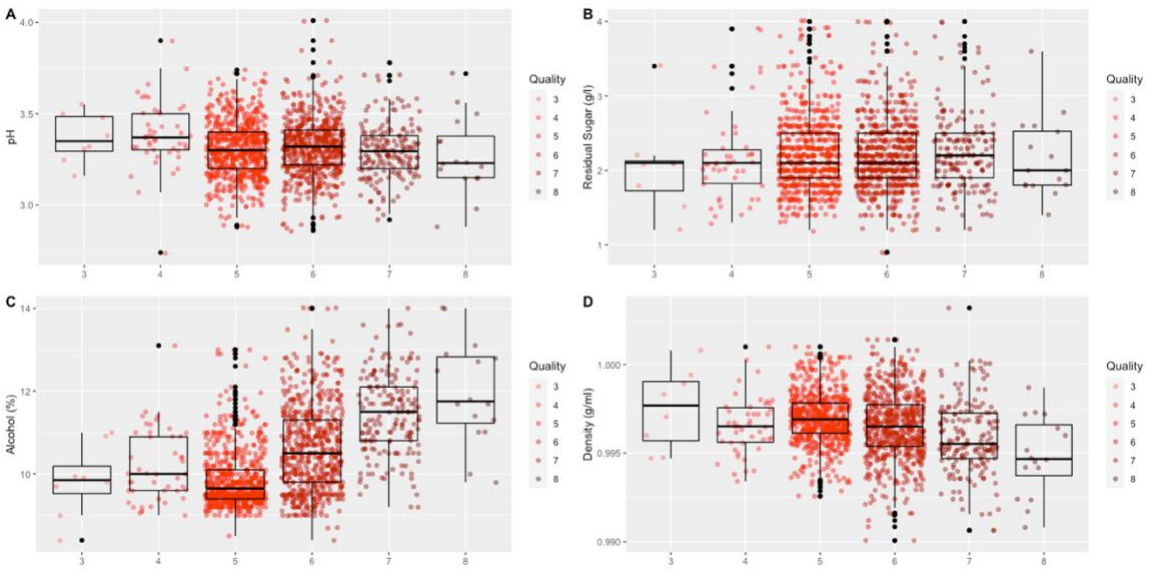


Figure 30: Box plot of pH, Residual Sugar, Alcohol and Density by quality (dry red wines)

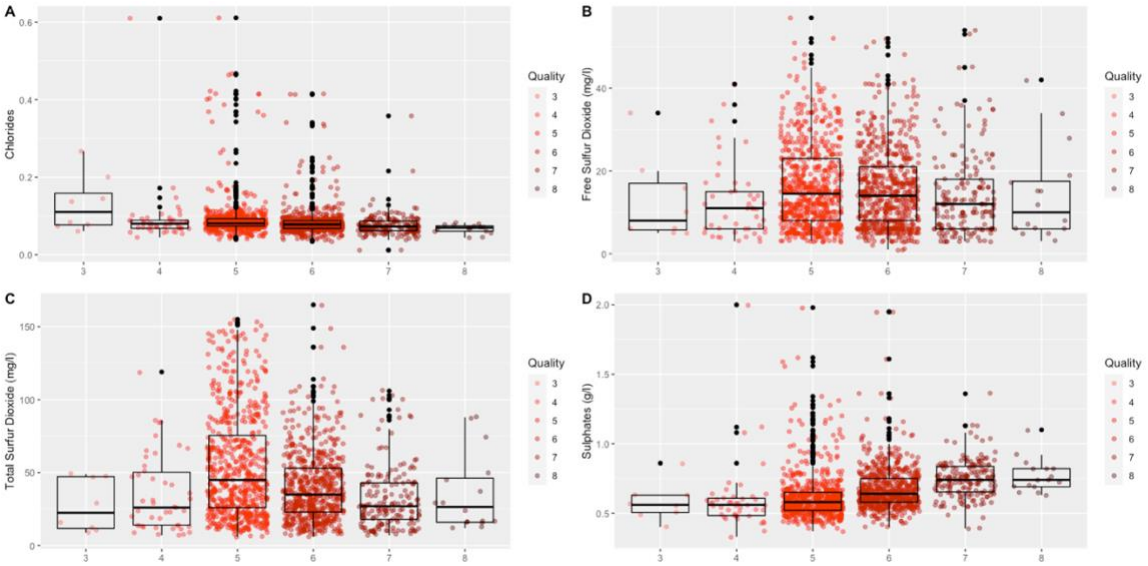


Figure 31: Box plot of Chlorides, Free and Total Sulfur Dioxide and Sulphates by quality (dry red wines)

4.2.1. Insight 2 – Is any difference between dry white Vinho Verde wines' quality based on their chemical properties?

We can conclude that there are differences between the groups by quality. For detailed information about the Kruskal-Wallis H test see [38, Appendix V].

The Kruskal Test was followed by Dunn's test to find out which are the groups with differences. See detailed results in [40, Appendix V].

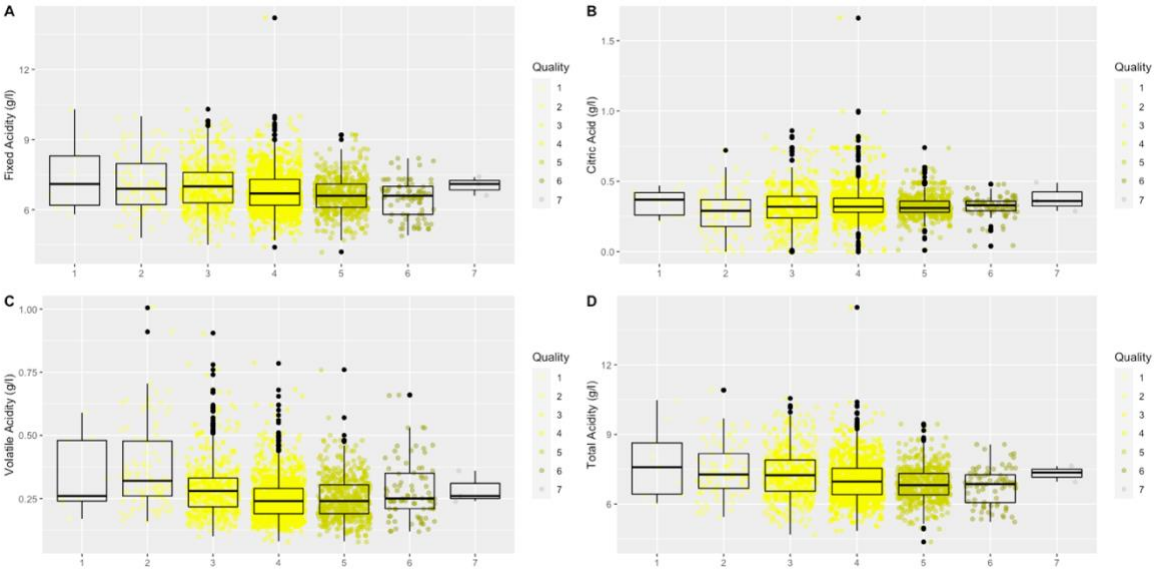


Figure 32: Box plot of Fixed Acidity, Acid Citric, Volatile and Total Acidity by quality (dry white wines)

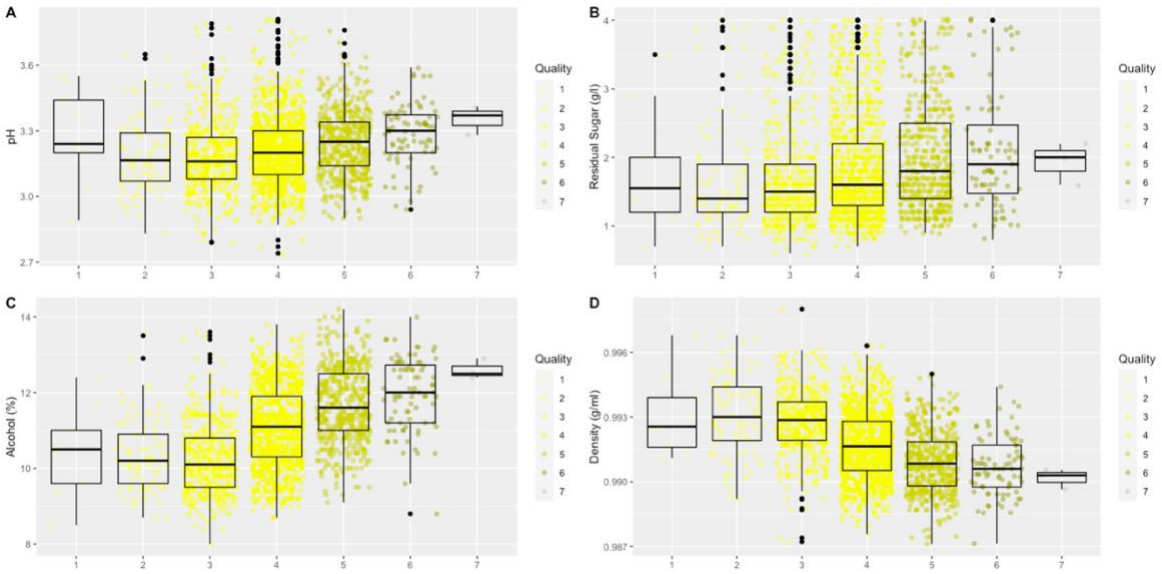


Figure 33: Box plot of pH, Residual Sugar, Alcohol and Density by quality (dry white wines)

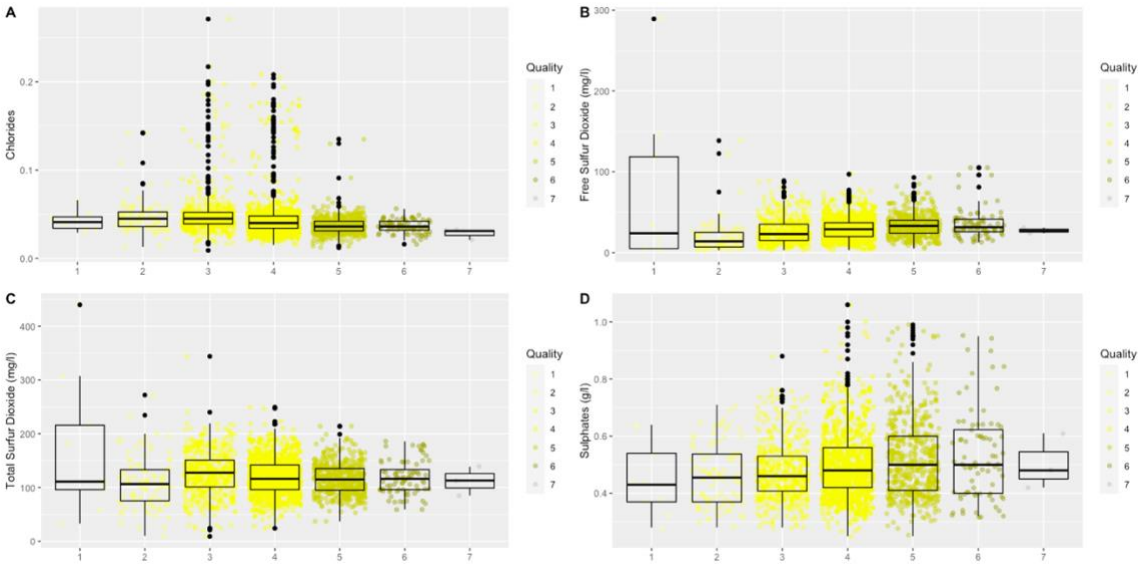


Figure 34: Box plot of Chlorides, Free and Total Sulfur Dioxide and Sulphates by quality (dry white wines)

4.3. Insight 3 – Is it possible the creation of a predictive simple linear model for some of the chemical properties of dry red Vinho Verde wine?

The simple linear models will be based on the variables with correlation from the previous exploration.

- Fixed Acidity – Total Acidity (positive corr.)
- Free Sulfur Dioxide – Total Sulfur Dioxide (positive corr.)
- Total Acidity – pH (negative corr.)
- Fixed Acidity – pH (negative corr.)

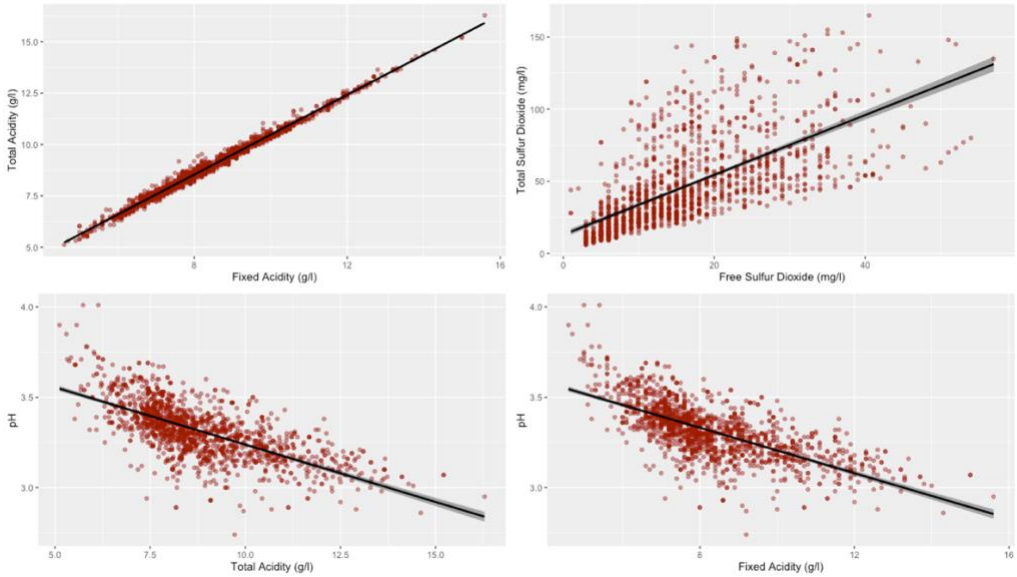


Figure 35: Scatter plots for chemical compounds with correlation. Simple Liner Model line in black.

4 models were created and tested:

- **Model 1:** Fixed Acidity (response variable) – Total Acidity

  Fixed Acidity = -0.6846 + 1.0181 Total Acidity

  The model is valid, there is a significant relationship between the variables, and it is highly effective as 98.9% of the changes in Fixed Acidity are explained by Total Acidity.

  Prediction for a red dry wine with Total Acidity of 7 g/l: Fixed Acidity of 6.441828 g/l(**Interval:** lower 5.985957 g/l – upper 6.8977 g/l).

- **Model 2:** Free Sulfur Dioxide (response variable) – Total Sulfur Dioxide

  Free Sulfur Dioxide = 6.0994 + 0.2095 Total Sulfur Dioxide

  The model is valid, there is a significant relationship between the variables, but only 43.43% of the changes in Free Sulfur Dioxide are explained by Total Sulfur Dioxide.

  Prediction for a red dry wine with Total Sulfur Dioxide of 100 mg/l: Free Sulfur Dioxide of 27.04616 mg/l (**Interval:** lower 8.01348 mg/l– upper 46.07884 mg/l).

- **Model 3:** Total Acidity (response variable) – pH

  Total Acidity = 32.561 – 7.167 pH

The model is valid, there is a significant relationship between the variables, but only 45.51% of the changes in Free Sulfur Dioxide are explained by Total Sulfur Dioxide.

Prediction for a red dry wine with a pH of 2.8: Total Acidity of 12.49274 g/l (**Interval:** lower 9.34958 g/l – upper 15.63589 g/l).

- **Model 4:** Fixed Acidity (response variable) – pH

  Fixed Acidity = 32.924 – 7.435 pH

  The model is valid, there is a significant relationship between the variables, but only 46.74% of the changes in Free Sulfur Dioxide are explained by Total Sulfur Dioxide.

  Prediction for a red dry wine with a pH of 2.8: Fixed Acidity of 12.10498 g/l (**Interval:** lower 8.923684 g/l – upper 15.28628 g/l).

For full results of each of the models see [46, Appendix VI].

VARIABLES	RANGE
Fixed Acidity	4.6 – 15.6
Total Acidity	5.120 – 16.285
pH	2.74 – 4.01
Free Sulfur Dioxide	1 – 57
Total Sulfur Dioxide	6 - 165

Table 2: Range for variables contained in subset dry red wines

were the Simple Linear Model can be applied.

4.4. Insight 4 – Is it possible to classify dry Vinho Verde wines by type (red / white)?

To classify red / white wines the chosen model is Random Forest. Only dry wines with quality 5 will be considered in the classification and prediction. From a total of 1,094 observations, 876 will be used for training purposes and the remaining 218 will be used to test the model.

The model has an accuracy of 99.08%, a sensitivity of 99.20%, and a precision of 98.92%. The model is considered a success. See detailed information about Confusion

Matrix in [48, Appendix VII]. As seen in Fig. 36, Density, Chlorides, and Volatile Acidity seems to be key for the Model.

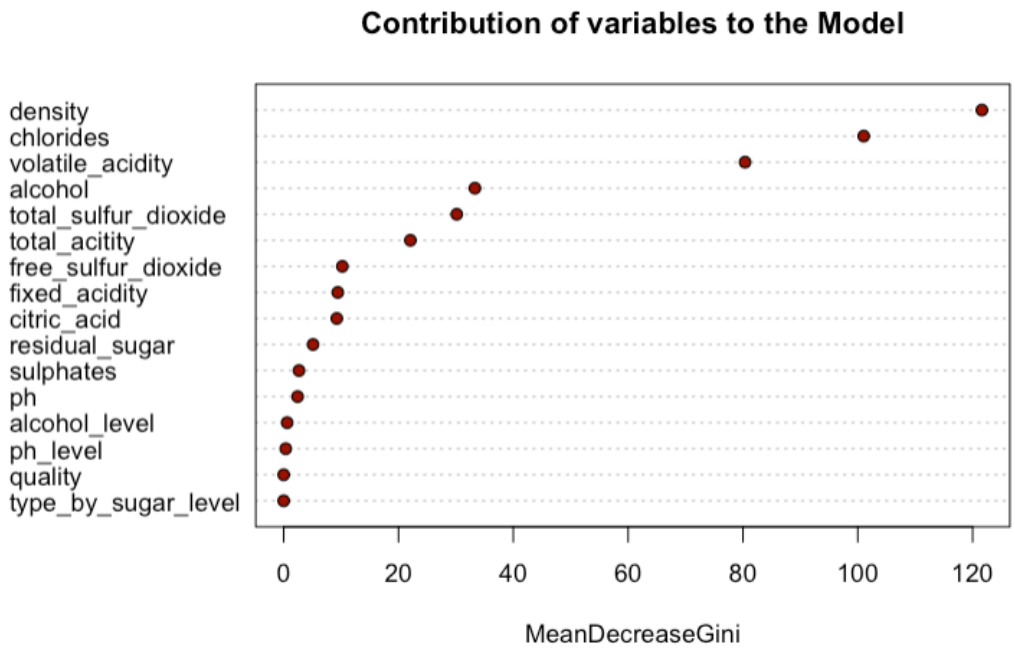


Figure 36: Contribution of variables to the Model - Dry Red/White Classification

4.4.1. Insight 4 – Is it possible to classify dry Vinho Verde wines by quality?

To classify red / white dry wines by quality the chosen model is again Random Forest.

Datasets

- Dry Red Wines: From a total of 1,474 observations, 1,182 will be used for training purposes and the remaining 292 will be used to test the model.

- Dry White Wines: From a total of 2,097 observations, 1,678 will be used for training purposes and the remaining 416 will be used to test the model.

The number of observations within the quality groups/classes is really low for some of them.

Models Result

- Dry Red Wines: The accuracy of the model is only 71.58%. As seen in Fig. 37, Alcohol, Sulphates, and Volatile Acidity seems to be key for the Model.

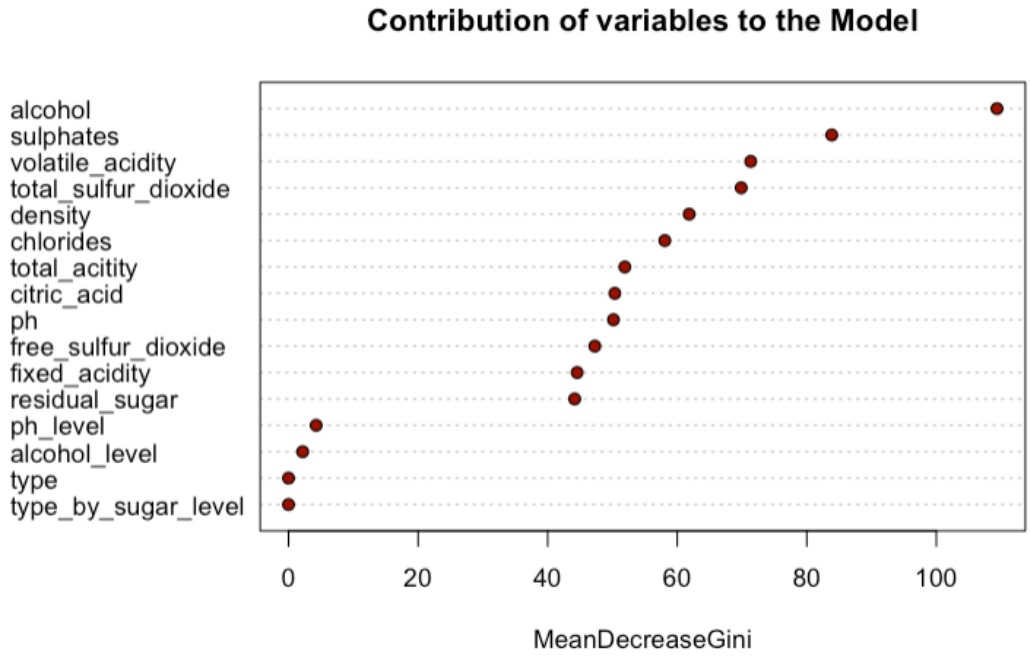


Figure 37: Contribution of variables to the Model - Dry Red Wine Quality Classification

- Dry White Wines: The accuracy of the model is only 62.26%. As seen in Fig. 38, Density, Free Sulfur Dioxide, and Alcohol seems to be key for the Model.

  The fact that the number of observations is low for some of the qualities, has an impact on the accuracy, sensitivity, and precision of the model.

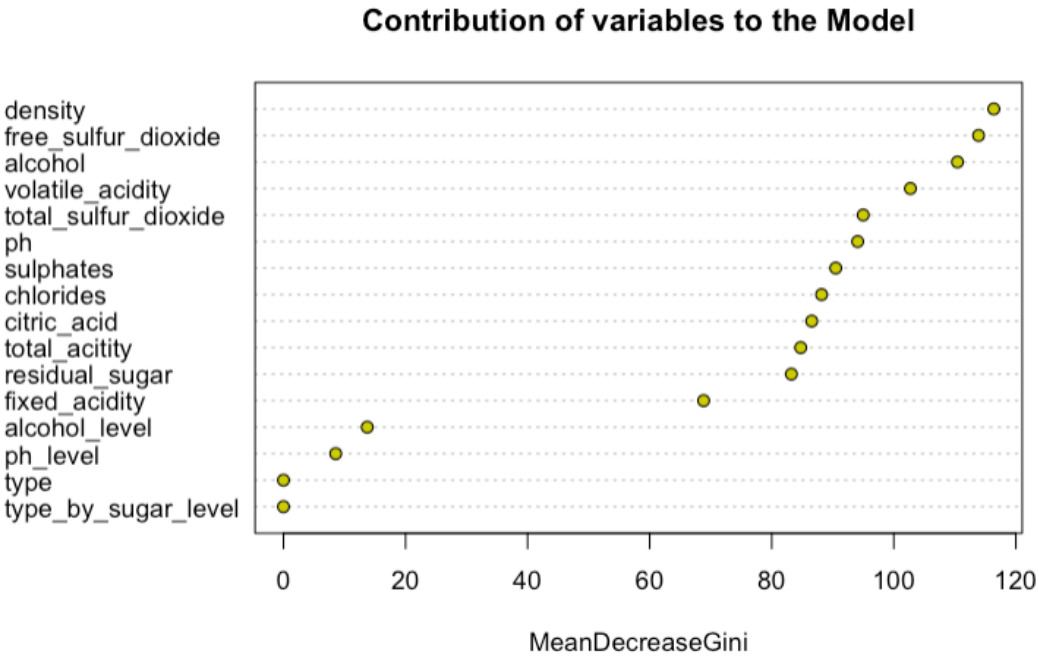


Figure 38: Contribution of variables to the Model - Dry White Wine Quality Classification

Only for qualities 6 and 7 in dry red wines, the first model seems accurate. The second model can't be considered valid. See detailed information about the Confusion Matrix for both models in [48, Appendix VII].

The next steps will be removing variables that have a low contribution to the model to see if the accuracy of the models improves.

5. Conclusion

Vinho Verde is a wine with normally high acidity and low alcohol levels. The datasets are limited and contain low to medium quality wine only. High level of Volatile Acidity and Free Sulfur Dioxide found in some of the wines can contribute to the low quality of them.

The distributions of the chemical compounds do not follow a normal distribution and are highly skewed in some cases, which determined the use of non-parametric statistical tests along with the analysis.

Some of the chemical compounds have correlation, which allowed the creation of 4 successful simple regression models, to predict the data within the red dry wines dataset limits.

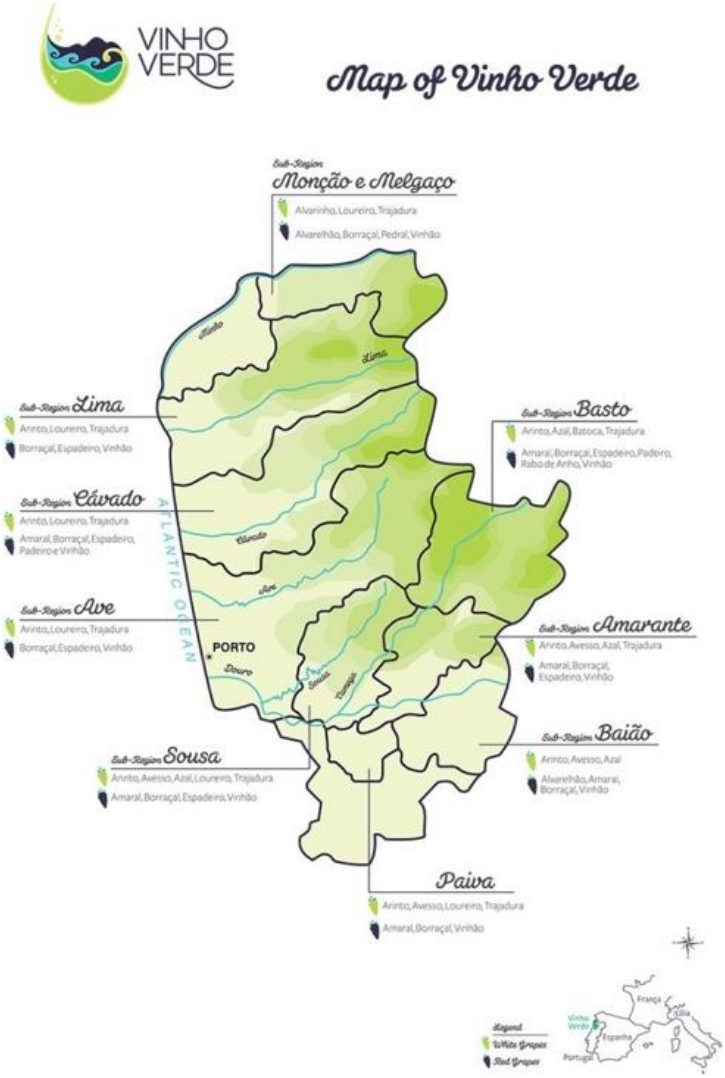
The analysis determined that red and white wines have different chemical properties and that those properties also can determine quality. Those insights allowed the creation of a successful model to classify dry wines by type (red/white). But the low diversity in the data sets related to quality resulted in two not valid models to classify wines by quality. The next steps will be improving the models.

6. Challenges

- Research about IEEE style was needed. [8] [9] [10] [11]
- Finding Datasets took me longer than expected.
- Trying to have a more innovative approach than existing projects using the same datasets was hard. See other projects in [12] [13] [14] [15] [16] [17] [18]
- A huge amount of my limited time was allocated for research on the topic and statistics.
- R Studio crashed many times and make me start from the start.
- The bibliography does not include a complete list of the huge number of sources I have checked when issues appeared while coding.

Appendix I – Brief Introduction to Vinho Verde

In the very north of Portugal, touching Spain is placed in the Minho region where Vinho Verde was born.



Vinho Verde's ("green wine") name refers to its original freshness and youthfulness and has nothing to do with the color of the wine. Historically Vinho Verde was drunk soon after was made, but nowadays we can find more serious wines as well.

Vinho Verde can be made out of 25 different grapes but the best wines come from Alvarinho, Trajadura, and Loureiro, as seen in [19].

Figure 39: Map of Vinho Verde, as seen in [20]

Appendix II – Wine Chemical Compounds in Datasets

Acidity:

Acidity plays such an important role in wine, balances the wine, and improves the aging capacity. Acidity would be recognized when tasting by making your mouth water.
Normally wines acidity is measured in total acidity and pH.

- **Total Acidity (g/l):** Fixed Acidity + Volatile Acidity. The total acidy in wines vary normally between 5.5 and 8.5, as seen in [21].

- **Fixed Acidity(g/l):** The fixed acids present in grapes are mainly acid tartaric, malic, citric and ascorbic (Vitamin C). During fermentation can appear acid succinic, and normally the acid ascorbic disappears then.

  Some producers can also add fumaric acid to the wine as a preservative.

- **Volatile Acidity(g/l):** The volatile acids appear during fermentation or due to microbial alterations. The number of volatile acids in wine is normally between 0.2 and 0.7 g/l. and winemakers try to keep it as lower as possible, as seen in [22] .

  The main acids in this category are acetic, formic, propionic, and butyric. Depending on the quantity of acetic acid and other factors can bring vinegary smell to it.

- **pH:** The PH of any substance is determining how acid or alkaline it is, being 7 the neutral value on a scale from 0 to 14. It represents the strengths of the acids. The PH in wines normally varies between 2.8 and 4, as seen in [21].

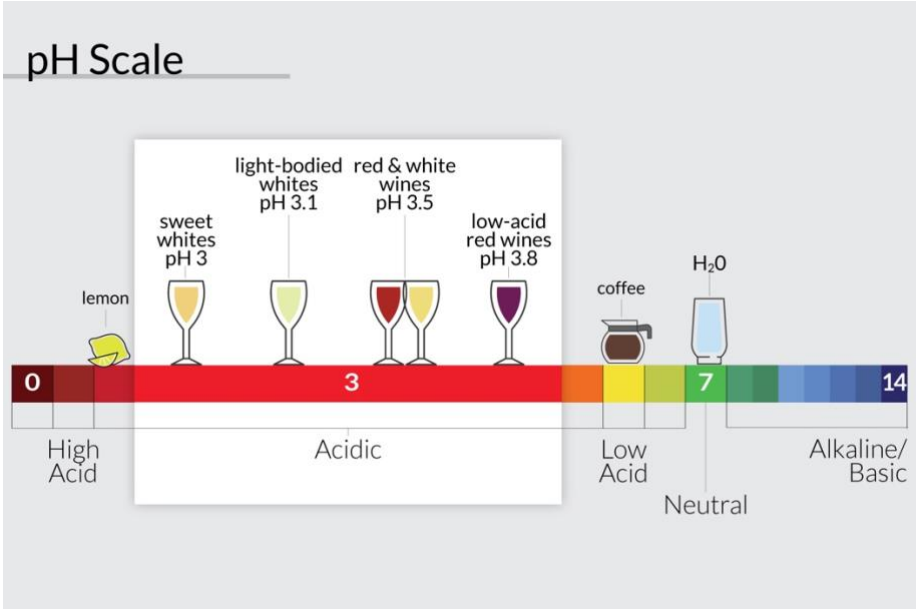


Figure 40: pH in wines, as seen in [23]

Normally after the alcoholic and malolactic fermentation, the total acidity gets reduced and the PH increased, resulting in less acidity in the wine.
White wines which have not malolactic fermentation will have a lower PH. Normally, even if not always, a lower PH can give a fresher and lower body wine, and the opposite when the Ph is high.

Sulphates (g/l):

Sometimes, in order to modify the conditions of the soil the vineyard's producers add sulphates. Those sulphates then can be found in wine. Normally we would like those to be zero or close to it.

Total Sulfur Dioxide (mg/l):

Sulfur Dioxide is normally added to wine to preserve it from microbial growth and protect it from oxidation. The amount needs to be less than 150 mg/l in red wines and less than200 mg/l in white wines (as red wines contains tannins which help preserving them naturally). If it contains more than 10 mg/l needs to be informed in the label as "contains sulfites", as seen in [23].

There are two types of sulfur dioxide, free and bound. As per [24]:

> "*The free sulfites are those available to react and thus exhibit both germicidal and antioxidant properties. The bound sulfites are those that have reacted (both reversibly and irreversibly) with other molecules within the wine medium."*

When the wine contains more than 35 mg/l of free sulfur dioxide can start being noticeable when tasting the wine as seen in [25][25].

Residual Sugar (g/l):

Naturally, grapes contain sugar, and this sugar disappears during alcoholic fermentation. Sometimes it can be residual sugar after this process is finished. Depending on the level of residual sugar the wines can be considered from dry to sweet, as seen in [26].

Chlorides:

Chlorides give saltiness to wine and the concentration depends on the terroir and the type of grape.

Density (g/ml):

Density is defined as mass per volume and alcohol has normally less than water. It is known that residual sugar and alcohol levels can influence the density observed in wine.

Alcohol (%):

Depending on the type of grape, weather, and when they have been harvested the initial sugar levels in grapes can be higher or lower. While the fermentation sugar is converted into alcohol, so grapes with a higher level of sugar will bring higher levels of alcohol to wine, as seen in [27].

Appendix III – Exploratory Analysis

Testing Normality

Null Hypothesis: the population is normally distributed

DRY RED WINE	DRY WHITE WINE
Shapiro-Wilk normality test – Fixed Acidity data: redwine_renamed_dry[, i] W = 0.95022, p-value < 2.2e-16	**Shapiro-Wilk normality test – Fixed Acidity** data: whitewine_renamed_dry[, i] W = 0.97096, p-value < 2.2e-16
Shapiro-Wilk normality test – Volatile Acidity data: redwine_renamed_dry[, i] W = 0.97353, p-value = 8.043e-16	**Shapiro-Wilk normality test – Volatile Acidity** data: whitewine_renamed_dry[, i] W = 0.88962, p-value < 2.2e-16
Shapiro-Wilk normality test – Citric Acid data: redwine_renamed_dry[, i] W = 0.95369, p-value < 2.2e-16	**Shapiro-Wilk normality test – Citric Acid** data: whitewine_renamed_dry[, i] W = 0.90968, p-value < 2.2e-16
Shapiro-Wilk normality test – Residual Sugar data: redwine_renamed_dry[, i] W = 0.944, p-value < 2.2e-16	**Shapiro-Wilk normality test – Residual Sugar** data: whitewine_renamed_dry[, i] W = 0.88418, p-value < 2.2e-16
Shapiro-Wilk normality test - Chlorides data: redwine_renamed_dry[, i] W = 0.46995, p-value < 2.2e-16	**Shapiro-Wilk normality test - Chlorides** data: whitewine_renamed_dry[, i] W = 0.59248, p-value < 2.2e-16
Shapiro-Wilk normality test – Free Sulfur Dioxide data: redwine_renamed_dry[, i] W = 0.91958, p-value < 2.2e-16	**Shapiro-Wilk normality test – Free Sulfur Dioxide** data: whitewine_renamed_dry[, i] W = 0.85834, p-value < 2.2e-16
Shapiro-Wilk normality test – Total Sulfur Dioxide data: redwine_renamed_dry[, i] W = 0.88884, p-value < 2.2e-16	**Shapiro-Wilk normality test – Total Sulfur Dioxide** data: whitewine_renamed_dry[, i] W = 0.97309, p-value < 2.2e-16
Shapiro-Wilk normality test - Density data: redwine_renamed_dry[, i] W = 0.99376, p-value = 7.345e-06	**Shapiro-Wilk normality test - Density** data: whitewine_renamed_dry[, i] W = 0.99572, p-value = 1.067e-05
Shapiro-Wilk normality test - pH data: redwine_renamed_dry[, i] W = 0.9928, p-value = 1.327e-06	**Shapiro-Wilk normality test - pH** data: whitewine_renamed_dry[, i] W = 0.98989, p-value = 5.972e-11

Shapiro-Wilk normality test - Sulphates data: redwine_renamed_dry[, i] W = 0.82412, p-value < 2.2e-16 Shapiro-Wilk normality test - Alcohol data: redwine_renamed_dry[, i] W = 0.92568, p-value < 2.2e-16 Shapiro-Wilk normality test – Total Acidity data: redwine_renamed_dry[, i] W = 0.95445, p-value < 2.2e-16	Shapiro-Wilk normality test - Sulphates data: whitewine_renamed_dry[, i] W = 0.9509, p-value < 2.2e-16 Shapiro-Wilk normality test - Alcohol data: whitewine_renamed_dry[, i] W = 0.98587, p-value = 1.521e-13 Shapiro-Wilk normality test – Total Acidity data: whitewine_renamed_dry[, i] W = 0.96949, p-value < 2.2e-16

Appendix IV – Insight 1

Insight 1 – Is any difference between white and red Vinho Verde wines based on their chemical properties?

Descriptive Statistics

DRY RED WHINES – QUALITY 5
Fixed Acidity
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 8.132166 7.8 2.098326 1.44856 5 13.3 0.9479009 4.051692
Volatile Acidity
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 0.5742994 0.58 0.02732766 0.1653108 0.18 1.33 0.625286 4.510644
Citric Acid
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 0.2381847 0.22 0.03117565 0.1765663 0 0.76 0.5406596 2.482748
Residual Sugar
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 2.211306 2.1 0.2516902 0.5016873 1.2 4 0.98784 4.221238
Chlorides
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 0.09234554 0.081 0.003018788 0.0549435 0.039 0.611 5.3611 35.9138
Free Sulfur Dioxide
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 16.28981 14.5 99.44379 9.972151 3 57 0.9298675 3.471384
Total Sulfur Dioxide
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 54.85669 45 1305.255 36.12832 6 155 0.8567435 2.849899
Density
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 0.9969653 0.9969 2.124367e-06 0.001457521 0.99256 1.001 0.07299805 3.447137
pH
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 3.307404 3.3 0.02263552 0.150451 2.88 3.74 0.1177582 2.926701
Sulphates
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 0.6166242 0.58 0.02872575 0.1694867 0.37 1.98 3.021923 16.66581
Alcohol
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 9.874363 9.65 0.481136 0.6936396 8.5 13 1.627627 6.299726
Total Acidity
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 8.706465 8.4025 2.039434 1.428087 5.52 13.73 0.8735447 3.92639
WHITE RED WHINES – QUALITY 5
Fixed Acidity
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 6.616202 6.6 0.5611832 0.7491216 4.2 9.2 0.3178317 3.813117
Volatile Acidity
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 0.2524034 0.24 0.007452168 0.08632594 0.08 0.76 0.9393171 5.290139
Citric Acid
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 0.3248498 0.31 0.006480945 0.08050432 0.01 0.74 0.7150978 6.216408
Residual Sugar
Mean Median Variance Standard Deviation Minimum Maximum Skewness Kurtosis
Stats 2.005258 1.8 0.6347411 0.7967064 0.9 4 0.885498 2.86742

Chlorides

	Mean	Median	Variance	Standard Deviation	Minimum	Maximum	Skewness	Kurtosis
Stats	0.03718026	0.036	0.0001146986	0.01070974	0.012	0.135	3.354095	29.76591

Free Sulfur Dioxide

	Mean	Median	Variance	Standard Deviation	Minimum	Maximum	Skewness	Kurtosis
Stats	33.35837	33	182.1261	13.49541	5	93	0.8312963	4.370637

Total Sulfur Dioxide

	Mean	Median	Variance	Standard Deviation	Minimum	Maximum	Skewness	Kurtosis
Stats	116.0236	115	820.2683	28.64033	37	214	0.3465593	3.149557

Density

	Mean	Median	Variance	Standard Deviation	Minimum	Maximum	Skewness	Kurtosis
Stats	0.9909233	0.99084	2.08336e-06	0.001443385	0.98711	0.995	0.2260654	2.729857

pH

	Mean	Median	Variance	Standard Deviation	Minimum	Maximum	Skewness	Kurtosis
Stats	3.247275	3.25	0.02269342	0.1506433	2.9	3.76	0.2140264	2.872974

Sulphates

	Mean	Median	Variance	Standard Deviation	Minimum	Maximum	Skewness	Kurtosis
Stats	0.5216524	0.5	0.01863791	0.1365207	0.25	0.99	0.8662798	3.668597

Alcohol

	Mean	Median	Variance	Standard Deviation	Minimum	Maximum	Skewness	Kurtosis
Stats	11.67786	11.6	1.007965	1.003974	9.1	14.2	0.02077431	2.463076

Total Acidity

	Mean	Median	Variance	Standard Deviation	Minimum	Maximum	Skewness	Kurtosis
Stats	6.868605	6.815	0.5466347	0.7393475	4.37	9.45	0.3315459	3.852505

Mann-Whitney U test

Null Hypothesis: the two groups come from the same population

Wilcoxon rank sum test with continuity correction – Fixed Acidity

data: redwine_renamed_dry_5[, i] and whitewine_renamed_dry_5[, i]
W = 246216, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

Wilcoxon rank sum test with continuity correction – Volatile Acidity

data: redwine_renamed_dry_5[, i] and whitewine_renamed_dry_5[, i]
W = 282626, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

Wilcoxon rank sum test with continuity correction – Citric Acid

data: redwine_renamed_dry_5[, i] and whitewine_renamed_dry_5[, i]
W = 88322, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

Wilcoxon rank sum test with continuity correction – Residual Sugar

data: redwine_renamed_dry_5[, i] and whitewine_renamed_dry_5[, i]
W = 186833, p-value = 4.183e-15
alternative hypothesis: true location shift is not equal to 0

Wilcoxon rank sum test with continuity correction - Chlorides

data: redwine_renamed_dry_5[, i] and whitewine_renamed_dry_5[, i]
W = 289817, p-value < 2.2e-16

alternative hypothesis: true location shift is not equal to 0

Wilcoxon rank sum test with continuity correction – Free Sulfur Dioxide

data: redwine_renamed_dry_5[, i] and whitewine_renamed_dry_5[, i]
W = 42484, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

Wilcoxon rank sum test with continuity correction – Total Sulfur Dioxide

data: redwine_renamed_dry_5[, i] and whitewine_renamed_dry_5[, i]
W = 31480, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

Wilcoxon rank sum test with continuity correction - Density

data: redwine_renamed_dry_5[, i] and whitewine_renamed_dry_5[, i]
W = 292028, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

Wilcoxon rank sum test with continuity correction - pH

data: redwine_renamed_dry_5[, i] and whitewine_renamed_dry_5[, i]
W = 178792, p-value = 3.292e-10
alternative hypothesis: true location shift is not equal to 0

Wilcoxon rank sum test with continuity correction - Sulphates

data: redwine_renamed_dry_5[, i] and whitewine_renamed_dry_5[, i]
W = 202410, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

Wilcoxon rank sum test with continuity correction - Alcohol

data: redwine_renamed_dry_5[, i] and whitewine_renamed_dry_5[, i]
W = 21466, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

Wilcoxon rank sum test with continuity correction – Total Acidity

data: redwine_renamed_dry_5[, i] and whitewine_renamed_dry_5[, i]
W = 261994, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

Appendix V – Insight 2

Insight 2 – Is any difference between dry red Vinho Verde wines' quality based on their chemical properties? / Is any difference between dry white Vinho Verde wines' quality based on their chemical properties?

Fligner-Killeen test

Null Hypothesis: Variances are equal (Homoscedasticity)

The following variables do not have equal variance
Red wines: Residual Sugar, Free Sulfur Dioxide, pH
White wines: pH

DRY RED WHINES
Fligner-Killeen test of homogeneity of variances – Fixed Acidity data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality Fligner-Killeen:med chi-squared = 35.115, df = 5, p-value = 1.427e-06
Fligner-Killeen test of homogeneity of variances – Volatile Acidity data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality Fligner-Killeen:med chi-squared = 24.848, df = 5, p-value = 0.0001491
Fligner-Killeen test of homogeneity of variances – Citric Acid data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality Fligner-Killeen:med chi-squared = 16.358, df = 5, p-value = 0.005893
Fligner-Killeen test of homogeneity of variances – Residual Sugar data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality Fligner-Killeen:med chi-squared = 1.3788, df = 5, p-value = **0.9266**
Fligner-Killeen test of homogeneity of variances - Chlorides data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality Fligner-Killeen:med chi-squared = 30.428, df = 5, p-value = 1.215e-05
Fligner-Killeen test of homogeneity of variances – Free Sulfur Dioxide data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality Fligner-Killeen:med chi-squared = 6.2625, df = 5, p-value = **0.2815**
Fligner-Killeen test of homogeneity of variances – Total Sulfur Dioxide data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality Fligner-Killeen:med chi-squared = 118.75, df = 5, p-value < 2.2e-16

Fligner-Killeen test of homogeneity of variances - Density

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 48.674, df = 5, p-value = 2.587e-09

Fligner-Killeen test of homogeneity of variances - pH

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 2.5744, df = 5, p-value = **0.7653**

Fligner-Killeen test of homogeneity of variances - Sulphates

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 11.193, df = 5, p-value = 0.04768

Fligner-Killeen test of homogeneity of variances - Alcohol

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 133.68, df = 5, p-value < 2.2e-16

Fligner-Killeen test of homogeneity of variances – Total Acidity

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 29.399, df = 5, p-value = 1.936e-05

WHITE RED WHINES

Fligner-Killeen test of homogeneity of variances – Fixed Acidity

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 59.122, df = 6, p-value = 6.786e-11

Fligner-Killeen test of homogeneity of variances – Volatile Acidity

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 62.57, df = 6, p-value = 1.35e-11

Fligner-Killeen test of homogeneity of variances – Citric Acid

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 116.9, df = 6, p-value < 2.2e-16

Fligner-Killeen test of homogeneity of variances – Residual Sugar

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 30.114, df = 6, p-value = 3.739e-05

Fligner-Killeen test of homogeneity of variances - Chlorides

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 57.191, df = 6, p-value = 1.672e-10

Fligner-Killeen test of homogeneity of variances – Free Sulfur Dioxide

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 34.918, df = 6, p-value = 4.471e-06

Fligner-Killeen test of homogeneity of variances – Total Sulfur Dioxide

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 75.815, df = 6, p-value = 2.608e-14

Fligner-Killeen test of homogeneity of variances - Density

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 30.518, df = 6, p-value = 3.132e-05

Fligner-Killeen test of homogeneity of variances - pH

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 7.0139, df = 6, p-value = **0.3196**

Fligner-Killeen test of homogeneity of variances - Sulphates

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 54.533, df = 6, p-value = 5.76e-10

Fligner-Killeen test of homogeneity of variances - Alcohol

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 31.383, df = 6, p-value = 2.142e-05

Fligner-Killeen test of homogeneity of variances – Total Acidity

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Fligner-Killeen:med chi-squared = 61.007, df = 6, p-value = 2.809e-11

Kruskal-Wallis H test

Null Hypothesis: Mean ranks of the groups are the same

The following variables do not have different mean ranks:
Red wine: Residual Sugar

DRY RED WHINES
Kruskal-Wallis rank sum test – Fixed Acidity data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality Kruskal-Wallis chi-squared = 19.929, df = 5, p-value = 0.001289
Kruskal-Wallis rank sum test – Volatile Acidity data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality Kruskal-Wallis chi-squared = 222, df = 5, p-value < 2.2e-16

Kruskal-Wallis rank sum test – Citric Acid

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Kruskal-Wallis chi-squared = 76.051, df = 5, p-value = 5.614e-15

Kruskal-Wallis rank sum test – Residual Sugar

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Kruskal-Wallis chi-squared = 4.6468, df = 5, p-value = 0.4605

Kruskal-Wallis rank sum test - Chlorides

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Kruskal-Wallis chi-squared = 67.688, df = 5, p-value = 3.1e-13

Kruskal-Wallis rank sum test – Free Sulfur Dioxide

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Kruskal-Wallis chi-squared = 18.747, df = 5, p-value = 0.002142

Kruskal-Wallis rank sum test – Total Sulfur Dioxide

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Kruskal-Wallis chi-squared = 91.104, df = 5, p-value < 2.2e-16

Kruskal-Wallis rank sum test - Density

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Kruskal-Wallis chi-squared = 72.68, df = 5, p-value = 2.834e-14

Kruskal-Wallis rank sum test - pH

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Kruskal-Wallis chi-squared = 17.86, df = 5, p-value = 0.003127

Kruskal-Wallis rank sum test - Sulphates

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Kruskal-Wallis chi-squared = 232.61, df = 5, p-value < 2.2e-16

Kruskal-Wallis rank sum test - Alcohol

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Kruskal-Wallis chi-squared = 385.03, df = 5, p-value < 2.2e-16

Kruskal-Wallis rank sum test – Total Acidity

data: redwine_renamed_dry[, i] by redwine_renamed_dry$quality
Kruskal-Wallis chi-squared = 10.589, df = 5, p-value = 0.06016

WHITE RED WHINES

Kruskal-Wallis rank sum test – Fixed Acidity

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Kruskal-Wallis chi-squared = 50.68, df = 6, p-value = 3.434e-09

Kruskal-Wallis rank sum test – Volatile Acidity

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Kruskal-Wallis chi-squared = 110.27, df = 6, p-value < 2.2e-16

Kruskal-Wallis rank sum test – Citric Acid

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Kruskal-Wallis chi-squared = 17.565, df = 6, p-value = 0.007416

Kruskal-Wallis rank sum test – Residual Sugar

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Kruskal-Wallis chi-squared = 76.949, df = 6, p-value = 1.522e-14

Kruskal-Wallis rank sum test - Chlorides

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Kruskal-Wallis chi-squared = 197.25, df = 6, p-value < 2.2e-16

Kruskal-Wallis rank sum test – Free Sulfur Dioxide

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Kruskal-Wallis chi-squared = 164.54, df = 6, p-value < 2.2e-16

Kruskal-Wallis rank sum test – Total Sulfur Dioxide

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Kruskal-Wallis chi-squared = 34.52, df = 6, p-value = 5.338e-06

Kruskal-Wallis rank sum test - Density

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Kruskal-Wallis chi-squared = 405.16, df = 6, p-value < 2.2e-16

Kruskal-Wallis rank sum test - pH

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Kruskal-Wallis chi-squared = 79.096, df = 6, p-value = 5.491e-15

Kruskal-Wallis rank sum test - Sulphates

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Kruskal-Wallis chi-squared = 38.07, df = 6, p-value = 1.088e-06

Kruskal-Wallis rank sum test - Alcohol

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Kruskal-Wallis chi-squared = 488.5, df = 6, p-value < 2.2e-16

Kruskal-Wallis rank sum test – Total Acidity

data: whitewine_renamed_dry[, i] by whitewine_renamed_dry$quality
Kruskal-Wallis chi-squared = 67.318, df = 6, p-value = 1.449e-12

Dunn's test

Null Hypothesis: No difference between groups

DRY RED WHINES

Fixed Acidity
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	3 - 4	1.0931693	2.743195e-01	1.000000000
2	3 - 5	0.5442701	5.862556e-01	1.000000000
3	4 - 5	-1.4737169	1.405578e-01	1.000000000
4	3 - 6	0.3691211	7.120374e-01	1.000000000
5	4 - 6	-1.8786977	6.028578e-02	0.904286737
6	5 - 6	-1.0919133	2.748712e-01	1.000000000
7	3 - 7	-0.4078316	6.833973e-01	1.000000000
8	4 - 7	-3.4153072	6.371010e-04	0.009556516
9	5 - 7	-3.9817311	6.841515e-05	0.001026227
10	6 - 7	-3.2395961	1.196991e-03	0.017954866
11	3 - 8	0.2033442	8.388660e-01	1.000000000
12	4 - 8	-1.1394191	2.545284e-01	1.000000000
13	5 - 8	-0.4171185	6.765917e-01	1.000000000
14	6 - 8	-0.1710130	8.642136e-01	1.000000000
15	7 - 8	0.9015349	3.673040e-01	1.000000000

Volatile Acidity
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	3 - 4	0.6609640	5.086354e-01	1.000000e+00
2	3 - 5	2.1942041	2.822074e-02	4.233111e-01
3	4 - 5	3.4534438	5.534776e-04	8.302164e-03
4	3 - 6	3.4999575	4.653323e-04	6.979984e-03
5	4 - 6	6.4876447	8.718860e-11	1.307829e-09
6	5 - 6	8.1509172	3.611745e-16	5.417617e-15
7	3 - 7	5.0865789	3.645800e-07	5.468701e-06
8	4 - 7	9.5667056	1.103665e-21	1.655497e-20
9	5 - 7	12.3561470	4.512734e-35	6.769101e-34
10	6 - 7	6.8970536	5.309216e-12	7.963824e-11
11	3 - 8	4.1472879	3.364369e-05	5.046554e-04
12	4 - 8	5.3150393	1.066343e-07	1.599514e-06
13	5 - 8	4.0097791	6.077557e-05	9.116335e-04
14	6 - 8	2.1721694	2.984289e-02	4.476433e-01
15	7 - 8	-0.1662306	8.679755e-01	1.000000e+00

DRY WHITE WHINES

Fixed Acidity
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	1 - 2	0.40840355	6.829774e-01	1.000000e+00
2	1 - 3	0.55810093	5.767755e-01	1.000000e+00
3	2 - 3	0.40257489	6.872610e-01	1.000000e+00
4	1 - 4	1.20037511	2.299937e-01	1.000000e+00
5	2 - 4	2.39808656	1.648097e-02	3.461004e-01
6	3 - 4	3.90519614	9.414899e-05	1.977129e-03
7	1 - 5	1.69284033	9.048586e-02	1.000000e+00
8	2 - 5	3.77828387	1.579128e-04	3.316169e-03
9	3 - 5	5.97842687	2.253027e-09	4.731357e-08
10	4 - 5	2.95327036	3.144265e-03	6.602955e-02
11	1 - 6	1.93432653	5.307299e-02	1.000000e+00
12	2 - 6	3.53405589	4.092345e-04	8.593925e-03
13	3 - 6	4.09817095	4.164277e-05	8.744982e-04
14	4 - 6	2.38606363	1.702980e-02	3.576259e-01
15	5 - 6	0.91203351	3.617511e-01	1.000000e+00
16	1 - 7	-0.09048238	9.279039e-01	1.000000e+00
17	2 - 7	-0.34582672	7.294729e-01	1.000000e+00
18	3 - 7	-0.42825457	6.684658e-01	1.000000e+00
19	4 - 7	-0.79957589	4.239566e-01	1.000000e+00
20	5 - 7	-1.08773910	2.767103e-01	1.000000e+00
21	6 - 7	-1.25901985	2.080232e-01	1.000000e+00

Volatile Acidity
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	1 - 2	-0.64783924	5.170889e-01	1.000000e+00
2	1 - 3	0.66491262	5.061064e-01	1.000000e+00
3	2 - 3	4.00922008	6.091962e-05	1.279312e-03
4	1 - 4	1.88175330	5.986952e-02	1.000000e+00
5	2 - 4	7.91179810	2.536976e-15	5.327649e-14
6	3 - 4	7.40788751	1.283271e-13	2.694869e-12
7	1 - 5	1.74317245	8.130350e-02	1.000000e+00
8	2 - 5	7.18766194	6.591021e-13	1.384114e-11
9	3 - 5	5.68151262	1.335087e-08	2.803682e-07
10	4 - 5	-0.76878081	4.420234e-01	1.000000e+00
11	1 - 6	1.06908726	2.850304e-01	1.000000e+00
12	2 - 6	3.95707502	7.587311e-05	1.593335e-03
13	3 - 6	1.26759264	2.049435e-01	1.000000e+00
14	4 - 6	-2.18342768	2.900433e-02	6.090909e-01
15	5 - 6	-1.74159192	8.157988e-02	1.000000e+00
16	1 - 7	0.09640553	9.231985e-01	1.000000e+00
17	2 - 7	0.49500911	6.205937e-01	1.000000e+00
18	3 - 7	-0.27510279	7.832373e-01	1.000000e+00

Acid Citric
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	3 - 4	0.4113317	6.808293e-01	1.000000e+00
2	3 - 5	-0.7239539	4.690940e-01	1.000000e+00
3	4 - 5	-2.7178923	6.569923e-03	9.854884e-02
4	3 - 6	-1.1660991	2.435744e-01	1.000000e+00
5	4 - 6	-3.7430232	1.818195e-04	2.727292e-03
6	5 - 6	-2.7599750	5.780579e-03	8.670869e-02
7	3 - 7	-2.4799538	1.313994e-02	1.970991e-01
8	4 - 7	-6.3592139	2.027888e-10	3.041832e-09
9	5 - 7	-7.4604700	8.621435e-14	1.293215e-12
10	6 - 7	-5.5967271	2.184361e-08	3.276542e-07
11	3 - 8	-2.1885188	2.863183e-02	4.294775e-01
12	4 - 8	-3.8079667	1.401141e-04	2.101712e-03
13	5 - 8	-2.7257947	6.414688e-03	9.622032e-02
14	6 - 8	-2.1028299	3.548064e-02	5.322096e-01
15	7 - 8	-0.1949562	8.454273e-01	1.000000e+00

Residual Sugar
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	3 - 4	-0.55786659	0.5769355	1
2	3 - 5	-0.98690309	0.3236901	1
3	4 - 5	-0.89978461	0.3682349	1
4	3 - 6	-1.03865014	0.2989675	1
5	4 - 6	-1.01947664	0.3079767	1
6	5 - 6	-0.32454572	0.7455249	1
7	3 - 7	-1.31328128	0.1890882	1
8	4 - 7	-1.57530565	0.1151859	1
9	5 - 7	-1.44425750	0.1486666	1
10	6 - 7	-1.22246194	0.2215330	1
11	3 - 8	-0.52945512	0.5964898	1
12	4 - 8	-0.05361561	0.9572414	1
13	5 - 8	0.48141608	0.6302208	1
14	6 - 8	0.55424067	0.5794141	1
15	7 - 8	0.94015772	0.3471367	1

Chlorides
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

19	4 - 7	-0.97878205	3.276877e-01	1.000000e+00
20	5 - 7	-0.90187538	3.671231e-01	1.000000e+00
21	6 - 7	-0.52986917	5.962026e-01	1.000000e+00

Acid Citric
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	1 - 2	1.7626441	0.0779605272	1.000000000
2	1 - 3	1.0153916	0.3099191729	1.000000000
3	2 - 3	-2.4399844	0.0146878978	0.308445853
4	1 - 4	0.6633589	0.5071007362	1.000000000
5	2 - 4	-3.6295644	0.0002838998	0.005961897
6	3 - 4	-2.1713440	0.0299051754	0.628008683
7	1 - 5	0.9317563	0.3514624612	1.000000000
8	2 - 5	-2.6662184	0.0076709842	0.161090667
9	3 - 5	-0.4355265	0.6631802754	1.000000000
10	4 - 5	1.6098124	0.1074388107	1.000000000
11	1 - 6	0.7110566	0.4770491569	1.000000000
12	2 - 6	-2.3997759	0.0164051119	0.344507351
13	3 - 6	-0.7607216	0.4468233472	1.000000000
14	4 - 6	0.2386568	0.8113717005	1.000000000
15	5 - 6	-0.5253199	0.5993608225	1.000000000
16	1 - 7	-0.3670672	0.7135689169	1.000000000
17	2 - 7	-1.4659117	0.1426723378	1.000000000
18	3 - 7	-1.0122414	0.3114226327	1.000000000
19	4 - 7	-0.8073912	0.4194411407	1.000000000
20	5 - 7	-0.9638748	0.3351087200	1.000000000
21	6 - 7	-0.8412304	0.4002188752	1.000000000

Residual Sugar
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	1 - 2	0.5211208	6.022826e-01	1.000000e+00
2	1 - 3	0.3501150	7.262524e-01	1.000000e+00
3	2 - 3	-0.5717136	5.675160e-01	1.000000e+00
4	1 - 4	-0.2994950	7.645624e-01	1.000000e+00
5	2 - 4	-2.6068075	9.139071e-03	1.919205e-01
6	3 - 4	-3.9711831	7.151657e-05	1.501848e-03
7	1 - 5	-1.0896724	2.758575e-01	1.000000e+00
8	2 - 5	-4.8515124	1.225236e-06	2.572995e-05
9	3 - 5	-7.5804682	3.443102e-14	7.230514e-13
10	4 - 5	-4.6929862	2.692457e-06	5.654159e-05
11	1 - 6	-1.3404651	1.800942e-01	1.000000e+00
12	2 - 6	-4.2936454	1.757631e-05	3.691026e-04
13	3 - 6	-4.9019063	9.491108e-07	1.993133e-05
14	4 - 6	-3.1838160	1.453474e-03	3.052295e-02
15	5 - 6	-0.8640762	3.875460e-01	1.000000e+00
16	1 - 7	-0.9626619	3.357172e-01	1.000000e+00
17	2 - 7	-1.4042966	1.602306e-01	1.000000e+00
18	3 - 7	-1.3116729	1.896305e-01	1.000000e+00
19	4 - 7	-0.9363265	3.491051e-01	1.000000e+00
20	5 - 7	-0.4748912	6.348645e-01	1.000000e+00
21	6 - 7	-0.2899096	7.718854e-01	1.000000e+00

Chlorides
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	3 - 4	1.6072298	1.080040e-01	1.000000e+00
2	3 - 5	1.1835416	2.365946e-01	1.000000e+00
3	4 - 5	-1.2738246	2.027256e-01	1.000000e+00
4	3 - 6	1.9728271	4.851527e-02	7.277290e-01
5	4 - 6	0.5651039	5.720031e-01	1.000000e+00
6	5 - 6	4.9267222	8.362051e-07	1.254308e-05
7	3 - 7	2.7495179	5.968300e-03	8.952449e-02
8	4 - 7	2.2831515	2.242145e-02	3.363217e-01
9	5 - 7	6.6895086	2.239212e-11	3.358818e-10
10	6 - 7	3.3934507	6.901798e-04	1.035270e-02
11	3 - 8	3.3766972	7.336175e-04	1.100426e-02
12	4 - 8	2.9164803	3.540051e-03	5.310077e-02
13	5 - 8	4.1121487	3.919937e-05	5.879906e-04
14	6 - 8	3.0005228	2.695166e-03	4.042749e-02
15	7 - 8	1.7912750	7.324918e-02	1.000000e+00

	Comparison	Z	P.unadj	P.adj
1	1 - 2	-0.6479771	5.169998e-01	1.000000e+00
2	1 - 3	-0.9303390	3.521956e-01	1.000000e+00
3	2 - 3	-0.7731995	4.394043e-01	1.000000e+00
4	1 - 4	0.2775328	7.813710e-01	1.000000e+00
5	2 - 4	2.9478015	3.200425e-03	6.720893e-02
6	3 - 4	7.3904696	1.463112e-13	3.072535e-12
7	1 - 5	1.5067857	1.318656e-01	1.000000e+00
8	2 - 5	6.4844990	8.902711e-11	1.869569e-09
9	3 - 5	12.8298073	1.116416e-37	2.344473e-36
10	4 - 5	7.2954380	2.976890e-13	6.251468e-12
11	1 - 6	1.6016613	1.092305e-01	1.000000e+00
12	2 - 6	5.1883485	2.121673e-07	4.455513e-06
13	3 - 6	7.2897570	3.105147e-13	6.520808e-12
14	4 - 6	4.0350879	5.458185e-05	1.146219e-03
15	5 - 6	0.4629729	6.433838e-01	1.000000e+00
16	1 - 7	2.0863269	3.694902e-02	7.759294e-01
17	2 - 7	2.7570429	5.832670e-03	1.224861e-01
18	3 - 7	2.9423374	3.257448e-03	6.840640e-02
19	4 - 7	2.2444538	2.480322e-02	5.208675e-01
20	5 - 7	1.5258754	1.270409e-01	1.000000e+00
21	6 - 7	1.4075845	1.592541e-01	1.000000e+00

Free Sulfur Dioxide
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	3 - 4	0.0288250	0.977004160	1.00000000
2	3 - 5	-1.2654437	0.205712328	1.00000000
3	4 - 5	-3.0199350	0.002528290	0.03792434
4	3 - 6	-1.1154776	0.264645827	1.00000000
5	4 - 6	-2.6673905	0.007644278	0.11466418
6	5 - 6	0.9335641	0.350528804	1.00000000
7	3 - 7	-0.5231564	0.600865411	1.00000000
8	4 - 7	-1.2076126	0.227196321	1.00000000
9	5 - 7	3.0474169	0.002308173	0.03462260
10	6 - 7	2.4145667	0.015753938	0.23630908
11	3 - 8	-0.3122580	0.754844437	1.00000000
12	4 - 8	-0.5039060	0.614327466	1.00000000
13	5 - 8	1.2443714	0.213362913	1.00000000
14	6 - 8	1.0334801	0.301379263	1.00000000
15	7 - 8	0.2065380	0.836370712	1.00000000

Free Sulfur Dioxide
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	1 - 2	2.339713298	1.929855e-02	4.052695e-01
2	1 - 3	0.536451644	5.916464e-01	1.000000e+00
3	2 - 3	-5.671430663	1.416099e-08	2.973808e-07
4	1 - 4	-0.409872650	6.818994e-01	1.000000e+00
5	2 - 4	-8.811152715	1.238655e-18	2.601175e-17
6	3 - 4	-5.785668985	7.222431e-09	1.516711e-07
7	1 - 5	-1.119701543	2.628410e-01	1.000000e+00
8	2 - 5	-10.553073223	4.916223e-26	1.032407e-24
9	3 - 5	-8.718985296	2.807060e-18	5.894826e-17
10	4 - 5	-4.219735205	2.445894e-05	5.136378e-04
11	1 - 6	-1.225826713	2.202639e-01	1.000000e+00
12	2 - 6	-8.200282064	2.398236e-16	5.036297e-15
13	3 - 6	-5.087882645	3.620833e-07	7.603749e-06
14	4 - 6	-2.520610475	1.171515e-02	2.460181e-01
15	5 - 6	-0.447527357	6.544943e-01	1.000000e+00
16	1 - 7	-0.008948197	9.928605e-01	1.000000e+00
17	2 - 7	-1.402157112	1.608683e-01	1.000000e+00
18	3 - 7	-0.321808230	7.475980e-01	1.000000e+00
19	4 - 7	0.227084735	8.203579e-01	1.000000e+00
20	5 - 7	0.640282596	5.219889e-01	1.000000e+00
21	6 - 7	0.722721688	4.698509e-01	1.000000e+00

Total Sulfur Dioxide
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	3 - 4	-0.64386871	5.196606e-01	1.000000e+00
2	3 - 5	-2.45533818	1.407521e-02	2.111282e-01
3	4 - 5	-4.10458591	4.050397e-05	6.075596e-04

Total Sulfur Dioxide
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	1 - 2	1.49857530	1.339838e-01	1.000000e+00
2	1 - 3	0.04715138	9.623926e-01	1.000000e+00
3	2 - 3	-4.52131444	6.145682e-06	1.290593e-04

4	3 - 6	-1.45875233	1.446333e-01	1.000000e+00
5	4 - 6	-1.78149967	7.483086e-02	1.000000e+00
6	5 - 6	6.21400055	5.165235e-10	7.747852e-09
7	3 - 7	-0.51153730	6.089749e-01	1.000000e+00
8	4 - 7	0.37202059	7.098775e-01	1.000000e+00
9	5 - 7	8.03819272	9.117316e-16	1.367597e-14
10	6 - 7	3.88278166	1.032683e-04	1.549024e-03
11	3 - 8	-0.57669929	5.641426e-01	1.000000e+00
12	4 - 8	-0.01059217	9.915488e-01	1.000000e+00
13	5 - 8	2.46435156	1.372614e-02	2.058921e-01
14	6 - 8	1.06373387	2.874493e-01	1.000000e+00
15	7 - 8	-0.24786064	8.042422e-01	1.000000e+00

Density
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	3 - 4	1.15580840	2.477595e-01	1.000000e+00
2	3 - 5	0.50141898	6.160763e-01	1.000000e+00
3	4 - 5	-1.73062111	8.351936e-02	1.000000e+00
4	3 - 6	1.25736574	2.086212e-01	1.000000e+00
5	4 - 6	0.03098429	9.752821e-01	1.000000e+00
6	5 - 6	4.71755102	2.387005e-06	3.580508e-05
7	3 - 7	2.25760834	2.397009e-02	3.595513e-01
8	4 - 7	2.25332151	2.423888e-02	3.635833e-01
9	5 - 7	7.44622581	9.604835e-14	1.440725e-12
10	6 - 7	4.28526489	1.825216e-05	2.737824e-04
11	3 - 8	2.69934804	6.947548e-03	1.042132e-01
12	4 - 8	2.50173057	1.235879e-02	1.853819e-01
13	5 - 8	3.91226772	9.143347e-05	1.371502e-03
14	6 - 8	2.84785106	4.401552e-03	6.602328e-02
15	7 - 8	1.34941808	1.772027e-01	1.000000e+00

pH
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	3 - 4	-0.2230591	0.8234895144	1.000000000
2	3 - 5	1.1924010	0.2331040326	1.000000000
3	4 - 5	3.3369042	0.0008471715	0.012707572
4	3 - 6	0.9424410	0.3459669220	1.000000000
5	4 - 6	2.7512071	0.0059376095	0.089064142
6	5 - 6	-1.5575965	0.1193289396	1.000000000
7	3 - 7	1.4105992	0.1583628322	1.000000000
8	4 - 7	3.5919243	0.0003282453	0.004923679
9	5 - 7	1.0015553	0.3165584250	1.000000000

4	1 - 4	0.69730278	4.856133e-01	1.000000e+00
5	2 - 4	-2.67321745	7.512753e-03	1.577678e-01
6	3 - 4	3.96519785	7.333509e-05	1.540037e-03
7	1 - 5	0.91769866	3.587766e-01	1.000000e+00
8	2 - 5	-1.89314290	5.833887e-02	1.000000e+00
9	3 - 5	4.58467573	4.546915e-06	9.548522e-05
10	4 - 5	1.32618652	1.847779e-01	1.000000e+00
11	1 - 6	0.76634321	4.434721e-01	1.000000e+00
12	2 - 6	-1.66625189	9.566325e-02	1.000000e+00
13	3 - 6	2.11038670	3.482506e-02	7.313263e-01
14	4 - 6	0.30790420	7.581552e-01	1.000000e+00
15	5 - 6	-0.32561136	7.447184e-01	1.000000e+00
16	1 - 7	0.61522735	5.384046e-01	1.000000e+00
17	2 - 7	-0.19222877	8.475630e-01	1.000000e+00
18	3 - 7	0.68096742	4.958921e-01	1.000000e+00
19	4 - 7	0.30537733	7.600788e-01	1.000000e+00
20	5 - 7	0.17491570	8.611459e-01	1.000000e+00
21	6 - 7	0.23928503	8.108846e-01	1.000000e+00

Density
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	1 - 2	-0.09971241	9.205727e-01	1.000000e+00
2	1 - 3	0.06750454	9.461800e-01	1.000000e+00
3	2 - 3	0.51263684	6.082054e-01	1.000000e+00
4	1 - 4	2.06926284	3.852143e-02	8.089500e-01
5	2 - 4	6.72501897	1.755696e-11	3.686961e-10
6	3 - 4	12.21030737	2.738526e-34	5.750905e-33
7	1 - 5	3.42328338	6.186954e-04	1.299260e-02
8	2 - 5	10.49687118	8.929078e-26	1.875106e-24
9	3 - 5	17.67241948	6.839935e-70	1.436386e-68
10	4 - 5	8.08539787	6.196159e-16	1.301193e-14
11	1 - 6	3.69411859	2.206508e-04	4.633667e-03
12	2 - 6	8.76707539	1.833670e-18	3.850707e-17
13	3 - 6	10.62012771	2.402322e-26	5.044877e-25
14	4 - 6	5.19838150	2.010312e-07	4.221655e-06
15	5 - 6	1.21244938	2.253404e-01	1.000000e+00
16	1 - 7	2.56413182	1.034343e-02	2.172119e-01
17	2 - 7	2.97398192	2.939624e-03	6.173210e-02
18	3 - 7	2.91303636	3.579329e-03	7.516591e-02
19	4 - 7	1.75750646	7.883152e-02	1.000000e+00
20	5 - 7	0.96228385	3.359070e-01	1.000000e+00
21	6 - 7	0.69825889	4.850153e-01	1.000000e+00

pH
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	1 - 2	1.6079207	1.078526e-01	1.000000e+00
2	1 - 3	1.8036914	7.127970e-02	1.000000e+00
3	2 - 3	0.4048867	6.855608e-01	1.000000e+00
4	1 - 4	1.2568807	2.087968e-01	1.000000e+00
5	2 - 4	-1.2940477	1.956489e-01	1.000000e+00
6	3 - 4	-3.3780330	7.300633e-04	1.533133e-02
7	1 - 5	0.4596424	6.457729e-01	1.000000e+00
8	2 - 5	-3.5939548	3.256964e-04	6.839624e-03
9	3 - 5	-7.0692634	1.557583e-12	3.270924e-11

10	6 - 7	2.0291136	0.0424467172	0.636700758
11	3 - 8	1.5556537	0.1197904572	1.000000000
12	4 - 8	2.6152998	0.0089149158	0.133723738
13	5 - 8	0.9849860	0.3246309239	1.000000000
14	6 - 8	1.3352046	0.1818093954	1.000000000
15	7 - 8	0.6260860	0.5312585276	1.000000000

Sulphates
Dunn (1964) Kruskal-Wallis multiple comparison p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	3 - 4	0.1831084	8.547130e-01	1.000000e+00
2	3 - 5	-0.4987057	6.179867e-01	1.000000e+00
3	4 - 5	-1.6208627	1.050471e-01	1.000000e+00
4	3 - 6	-2.0418644	4.116498e-02	6.174747e-01
5	4 - 6	-5.2090214	1.898392e-07	2.847588e-06
6	5 - 6	-9.6293262	6.012240e-22	9.018359e-21
7	3 - 7	-3.5429835	3.956275e-04	5.934412e-03
8	4 - 7	-8.1503742	3.627998e-16	5.441997e-15
9	5 - 7	-12.8826628	5.635701e-38	8.453551e-37
10	6 - 7	-6.4413792	1.183926e-10	1.775890e-09
11	3 - 8	-3.4120909	6.446661e-04	9.669992e-03
12	4 - 8	-5.3322159	9.702147e-08	1.455322e-06
13	5 - 8	-5.1351559	2.819103e-07	4.228654e-06
14	6 - 8	-2.9640280	3.036406e-03	4.554609e-02
15	7 - 8	-0.7516887	4.522383e-01	1.000000e+00

Alcohol
Dunn (1964) Kruskal-Wallis multiple comparison p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	3 - 4	-0.6413026	5.213261e-01	1.000000e+00
2	3 - 5	0.2395813	8.106549e-01	1.000000e+00
3	4 - 5	2.1663573	3.028389e-02	4.542584e-01
4	3 - 6	-1.8844762	5.950059e-02	8.925089e-01
5	4 - 6	-2.7783912	5.462881e-03	8.194321e-02
6	5 - 6	-13.2525717	4.360024e-40	6.540036e-39
7	3 - 7	-3.8289382	1.286973e-04	1.930460e-03
8	4 - 7	-6.8692034	6.456138e-12	9.684207e-11
9	5 - 7	-17.1557588	5.692167e-66	8.538251e-65
10	6 - 7	-8.2934727	1.099892e-16	1.649839e-15
11	3 - 8	-3.6944664	2.203490e-04	3.305235e-03
12	4 - 8	-4.6654184	3.079896e-06	4.619844e-05
13	5 - 8	-6.6557209	2.819147e-11	4.228721e-10
14	6 - 8	-3.6678792	2.445706e-04	3.668560e-03
15	7 - 8	-0.8239356	4.099761e-01	1.000000e+00

10	4 - 5	-4.6911776	2.716371e-06	5.704379e-05
11	1 - 6	-0.3386918	7.348419e-01	1.000000e+00
12	2 - 6	-4.4711823	7.778836e-06	1.633556e-04
13	3 - 6	-6.0381215	1.559187e-09	3.274292e-08
14	4 - 6	-4.6350231	3.568977e-06	7.494852e-05
15	5 - 6	-2.2621412	2.368868e-02	4.974623e-01
16	1 - 7	-1.0057176	3.145514e-01	1.000000e+00
17	2 - 7	-2.0998173	3.574492e-02	7.506432e-01
18	3 - 7	-2.2053052	2.743268e-02	5.760863e-01
19	4 - 7	-1.8874245	5.910325e-02	1.000000e+00
20	5 - 7	-1.4246492	1.542586e-01	1.000000e+00
21	6 - 7	-0.9376338	3.484327e-01	1.000000e+00

Sulphates
Dunn (1964) Kruskal-Wallis multiple comparison p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	1 - 2	-0.38286780	7.018178e-01	1.000000e+00
2	1 - 3	-0.82931676	4.069252e-01	1.000000e+00
3	2 - 3	-1.29518039	1.952580e-01	1.000000e+00
4	1 - 4	-1.37186206	1.701064e-01	1.000000e+00
5	2 - 4	-3.01109424	2.603080e-03	5.466469e-02
6	3 - 4	-3.29045112	1.000269e-03	2.100564e-02
7	1 - 5	-1.76044842	7.833181e-02	1.000000e+00
8	2 - 5	-4.05831788	4.942746e-05	1.037977e-03
9	3 - 5	-4.90750615	9.224176e-07	1.937077e-05
10	4 - 5	-2.34223397	1.916869e-02	4.025426e-01
11	1 - 6	-1.79454735	7.272585e-02	1.000000e+00
12	2 - 6	-3.26955481	1.077169e-03	2.262054e-02
13	3 - 6	-2.93025722	3.386815e-03	7.112312e-02
14	4 - 6	-1.47133205	1.412013e-01	1.000000e+00
15	5 - 6	-0.31794584	7.505260e-01	1.000000e+00
16	1 - 7	-0.86934962	3.846559e-01	1.000000e+00
17	2 - 7	-0.76041047	4.470093e-01	1.000000e+00
18	3 - 7	-0.51936704	6.035048e-01	1.000000e+00
19	4 - 7	-0.20763121	8.355169e-01	1.000000e+00
20	5 - 7	0.02225341	9.822458e-01	1.000000e+00
21	6 - 7	0.08734793	9.303950e-01	1.000000e+00

Alcohol
Dunn (1964) Kruskal-Wallis multiple comparison p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	1 - 2	0.5933594	5.529406e-01	1.000000e+00
2	1 - 3	0.8282015	4.075564e-01	1.000000e+00
3	2 - 3	0.6369107	5.241830e-01	1.000000e+00
4	1 - 4	-1.3828002	1.667262e-01	1.000000e+00
5	2 - 4	-6.1920997	5.936800e-10	1.246728e-08
6	3 - 4	-13.5077547	1.407489e-41	2.955728e-40
7	1 - 5	-2.8925634	3.821121e-03	8.024354e-02
8	2 - 5	-10.4406170	1.617559e-25	3.396875e-24
9	3 - 5	-19.5901647	1.875826e-85	3.939235e-84
10	4 - 5	-8.9894606	2.484468e-19	5.217383e-18
11	1 - 6	-3.4407215	5.801653e-04	1.218347e-02
12	2 - 6	-9.3137489	1.233982e-20	2.591362e-19
13	3 - 6	-12.3850591	3.148527e-35	6.611908e-34
14	4 - 6	-6.4070911	1.483222e-10	3.114767e-09
15	5 - 6	-1.9521312	5.092264e-02	1.000000e+00

Total Acidity
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	3 - 4	1.49234003	0.135610029	1.0000000
2	3 - 5	1.20578172	0.227901668	1.0000000
3	4 - 5	-0.93389339	0.350358890	1.0000000
4	3 - 6	1.17825975	0.238693067	1.0000000
5	4 - 6	-0.99578940	0.319352483	1.0000000
6	5 - 6	-0.16967948	0.865262210	1.0000000
7	3 - 7	0.54437538	0.586183184	1.0000000
8	4 - 7	-2.26083481	0.023769488	0.3565423
9	5 - 7	-2.71007698	0.006726759	0.1009014
10	6 - 7	-2.58502198	0.009737279	0.1460592
11	3 - 8	1.00169870	0.316489137	1.0000000
12	4 - 8	-0.47517824	0.634659935	1.0000000
13	5 - 8	0.01869652	0.985083203	1.0000000
14	6 - 8	0.05689569	0.954628289	1.0000000
15	7 - 8	0.90685327	0.364484397	1.0000000
16	1 - 7	-2.6525438	7.988775e-03	1.677643e-01
17	2 - 7	-3.3681699	7.566896e-04	1.589048e-02
18	3 - 7	-3.5349470	4.078569e-04	8.564995e-03
19	4 - 7	-2.2570345	2.400592e-02	5.041243e-01
20	5 - 7	-1.3724113	1.699354e-01	1.000000e+00
21	6 - 7	-0.9499808	3.421220e-01	1.000000e+00

Total Acidity
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

	Comparison	Z	P.unadj	P.adj
1	1 - 2	0.2182070	8.272678e-01	1.000000e+00
2	1 - 3	0.5955361	5.514851e-01	1.000000e+00
3	2 - 3	1.1065918	2.684704e-01	1.000000e+00
4	1 - 4	1.3787020	1.679867e-01	1.000000e+00
5	2 - 4	3.5630937	3.665097e-04	7.696704e-03
6	3 - 4	4.7638416	1.899416e-06	3.988774e-05
7	1 - 5	1.8758791	6.067189e-02	1.000000e+00
8	2 - 5	4.9100327	9.106121e-07	1.912285e-05
9	3 - 5	6.7453806	1.526265e-11	3.205156e-10
10	4 - 5	2.9862084	2.824601e-03	5.931661e-02
11	1 - 6	2.0119776	4.422229e-02	9.286682e-01
12	2 - 6	4.1498188	3.327386e-05	6.987511e-04
13	3 - 6	4.2206271	2.436236e-05	5.116096e-04
14	4 - 6	2.1077489	3.505272e-02	7.361071e-01
15	5 - 6	0.6286232	5.295958e-01	1.000000e+00
16	1 - 7	-0.0949640	9.243434e-01	1.000000e+00
17	2 - 7	-0.2377657	8.120628e-01	1.000000e+00
18	3 - 7	-0.4551523	6.489997e-01	1.000000e+00
19	4 - 7	-0.9080304	3.638622e-01	1.000000e+00
20	5 - 7	-1.1992487	2.304312e-01	1.000000e+00
21	6 - 7	-1.3105244	1.900185e-01	1.000000e+00

Appendix VI – Insight 3

Insight 3 – Is it possible the creation of a predictive simple linear model for some of the chemical properties of dry red Vinho Verde wine?

Linear Regression Models

Null Hypothesis: beta coefficient associated with the variables is equal to zero.

MODEL 1

```
Call:
lm(formula = fixed_acidity ~ total_acitity, data = redwine_renamed_dry)

Residuals:
    Min      1Q  Median      3Q     Max
-1.06121 -0.10328  0.00727  0.13005  0.40168

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.68461    0.02509  -27.28   <2e-16 ***
total_acitity  1.01806    0.00280  363.64   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1766 on 1472 degrees of freedom
Multiple R-squared:  0.989,    Adjusted R-squared:  0.989
F-statistic: 1.322e+05 on 1 and 1472 DF,  p-value: < 2.2e-16
```

MODEL 2

```
Call:
lm(formula = free_sulfur_dioxide ~ total_sulfur_dioxide, data = redwine_renamed_dry)

Residuals:
   Min     1Q  Median     3Q    Max
-21.053  -4.369  -1.664   3.637  31.143

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         6.099351   0.341345   17.87   <2e-16 ***
total_sulfur_dioxide 0.209468   0.006232   33.61   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.369 on 1472 degrees of freedom
Multiple R-squared:  0.4343,   Adjusted R-squared:  0.4339
F-statistic:  1130 on 1 and 1472 DF,  p-value: < 2.2e-16
```

MODEL 3

```
Call:
lm(formula = total_acitity ~ ph, data = redwine_renamed_dry)

Residuals:
   Min     1Q  Median     3Q    Max
-4.0793 -0.8095 -0.1600  0.6842  4.8673

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  32.5607     0.6780   48.02   <2e-16 ***
ph           -7.1671     0.2044  -35.06   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.214 on 1472 degrees of freedom
Multiple R-squared:  0.4551,   Adjusted R-squared:  0.4548
F-statistic:  1230 on 1 and 1472 DF,  p-value: < 2.2e-16
```

MODEL 4

```
Call:
lm(formula = fixed_acidity ~ ph, data = redwine_renamed_dry)

Residuals:
   Min     1Q  Median     3Q    Max
-3.9640 -0.8257 -0.1669  0.6742  4.9025

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  32.9236     0.6862   47.98   <2e-16 ***
ph           -7.4352     0.2069  -35.94   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.228 on 1472 degrees of freedom
Multiple R-squared:  0.4674,   Adjusted R-squared:  0.467
F-statistic:  1292 on 1 and 1472 DF,  p-value: < 2.2e-16
```

Appendix VII – Insight 4

Insight 4 – Is it possible to classify dry Vinho Verde wines by type (red / white)? / Is it possible to classify dry Vinho Verde wines by quality?

Confusion Matrix and Statistic

DRY WINE – QUALITY 5 – BY TYPE

Confusion Matrix and Statistics

```
          Reference
Prediction red white
     red   124     1
     white   1    92

               Accuracy : 0.9908
                 95% CI : (0.9673, 0.9989)
    No Information Rate : 0.5734
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9812

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9920
            Specificity : 0.9892
         Pos Pred Value : 0.9920
         Neg Pred Value : 0.9892
             Prevalence : 0.5734
         Detection Rate : 0.5688
   Detection Prevalence : 0.5734
      Balanced Accuracy : 0.9906

       'Positive' Class : red
```

DRY RED WINE – BY QUALITY

Confusion Matrix and Statistics

```
          Reference
Prediction  3  4  5  6  7  8
         3  0  0  0  0  0  0
         4  0  0  0  0  0  0
         5  0  5 97 23  1  0
         6  1  4 27 92 13  2
         7  0  0  1  5 20  1
         8  0  0  0  0  0  0

Overall Statistics

               Accuracy : 0.7158
                 95% CI : (0.6603, 0.7668)
    No Information Rate : 0.4281
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5332
```

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: 3	Class: 4	Class: 5	Class: 6	Class: 7	Class: 8
Sensitivity	0.000000	0.00000	0.7760	0.7667	0.58824	0.00000
Specificity	1.000000	1.00000	0.8263	0.7267	0.97287	1.00000
Pos Pred Value	NaN	NaN	0.7698	0.6619	0.74074	NaN
Neg Pred Value	0.996575	0.96918	0.8313	0.8170	0.94717	0.98973
Prevalence	0.003425	0.03082	0.4281	0.4110	0.11644	0.01027
Detection Rate	0.000000	0.00000	0.3322	0.3151	0.06849	0.00000
Detection Prevalence	0.000000	0.00000	0.4315	0.4760	0.09247	0.00000
Balanced Accuracy	0.500000	0.50000	0.8012	0.7467	0.78055	0.50000

DRY WHITE WINE – BY QUALITY

Confusion Matrix and Statistics

Prediction \ Reference	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	5	0	0	0	0
3	0	6	62	17	2	0
4	1	6	38	151	53	9
5	0	1	3	16	38	4
6	0	0	0	1	0	3

Overall Statistics

Accuracy : 0.6226
95% CI : (0.5741, 0.6694)
No Information Rate : 0.4447
P-Value [Acc > NIR] : 2.358e-13

Kappa : 0.4086

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: 1	Class: 2	Class: 3	Class: 4	Class: 5	Class: 6
Sensitivity	0.000000	0.27778	0.6019	0.8162	0.40860	0.187500
Specificity	1.000000	1.00000	0.9201	0.5368	0.92570	0.997500
Pos Pred Value	NaN	1.00000	0.7126	0.5853	0.61290	0.750000
Neg Pred Value	0.997596	0.96837	0.8754	0.7848	0.84463	0.968447
Prevalence	0.002404	0.04327	0.2476	0.4447	0.22356	0.038462
Detection Rate	0.000000	0.01202	0.1490	0.3630	0.09135	0.007212
Detection Prevalence	0.000000	0.01202	0.2091	0.6202	0.14904	0.009615
Balanced Accuracy	0.500000	0.63889	0.7610	0.6765	0.66715	0.592500

Bibliography

[1] A. C. F. A. T. M. a. J. R. P. Cortez, “Modeling wine preferences by data mining from physicochemical properties,” In Decision Support Systems, Elsevier, 47(4):547-553. ISSN: 0167-9236..
[2] “Wine Quality Wine,” 29 Jul. 2016. [Online]. Available: https://www.openml.org/d/40498. [Accessed 17 Apr. 2020].
[3] “Wine Quality Red,” 6 Apr. 2017. [Online]. Available: https://www.openml.org/d/40691. [Accessed 16 Apr. 2020].
[4] “Sweetness of wine,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Sweetness_of_wine. [Accessed 18 Apr. 2020].
[5] B. Gold, “A Handy Guide To The Alcohol Content in Every Type of Wine—Because Information Is Power,” 11 Jul. 2019. [Online]. Available: https://www.realsimple.com/holidays-entertaining/entertaining/food-drink/alcohol-content-wine. [Accessed 26 Apr. 2020].
[6] “Vinho Verde,” [Online]. Available: https://en.wikipedia.org/wiki/Vinho_Verde. [Accessed 20 Apr. 2020].
[7] “Normality Testing - Skewness and Kurtosis,” Good Data, [Online]. Available: https://help.gooddata.com/doc/en/reporting-and-dashboards/maql-analytical-query-language/maql-expression-reference/aggregation-functions/statistical-functions/predictive-statistical-use-cases/normality-testing-skewness-and-kurtosis. [Accessed 22 Apr. 2020].
[8] IEEE Periodicals Transactions/Journals Department, “IEEE Editorial Style Manual For Authors,” IEEE Author Center, 2019. [Online]. Available: https://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE_Style_Manual.pdf. [Accessed 5 Apr. 2020].
[9] J. A. Longo, “Preparing a Research Paper in IEEE Format,” University of Nevada, [Online]. Available: https://www.unlv.edu/sites/default/files/page_files/27/Engineering-PreparingPaperIEEE-Sept15.pdf. [Accessed 5 Apr. 2020].
[10] Victoria University, “IEEE Referencing,” [Online]. Available: http://libraryguides.vu.edu.au/ieeereferencing/webbaseddocument. [Accessed 5 Apr. 2020].
[11] “Writing a Scientific Research Article,” Columbia University, [Online]. Available: http://www.columbia.edu/cu/biology/ug/research/paper.html. [Accessed 13 Apr. 2020].
[12] D. Alekseeva, “Red and White Wine Quality,” [Online]. Available: https://rstudio-pubs-static.s3.amazonaws.com/57835_c4ace81da9dc45438ad0c286bcbb4224.html. [Accessed 21 Apr. 2020].
[13] “Analysis of Wine Quality Data,” PennState - Eberly College of Science, [Online]. Available: https://online.stat.psu.edu/stat508/lesson/analysis-wine-quality-data. [Accessed 17 Apr. 2020].
[14] S. Bhattacharjee, “An analysis of classification techniques in data mining for determining quality of wine product: red & white,” Jun. 2016. [Online]. Available: https://www.researchgate.net/publication/304180279_AN_ANALYSIS_OF_CLASSIFICATION_TECHNIQUES_IN_DATA_MINING_FOR_DETERMINING_QUALITY_OF_WINE_PRODUCT_RED_WHITE . [Accessed 16 Apr. 2020].
[15] N. Dhalia, “REPORT: The Effect Of Physicochemical On The Wine Quality,” 28 Jan. 2018. [Online]. Available: https://rstudio-pubs-static.s3.amazonaws.com/354404_c136afece292494593e4632ec8a2d65c.html. [Accessed 18 Apr. 2020].

[16] A. Hariharan, “Game of Wines - Using Data Science to understand what makes wines taste good,” 7 Feb. 2018. [Online]. Available: https://medium.com/free-code-camp/using-data-science-to-understand-what-makes-wine-taste-good-669b496c67ee. [Accessed 17 Apr. 2020].
[17] M. MacArdle, “Red Wine Analysis by Mark MacArdle,” [Online]. Available: https://markmacardle.com/redwine_eda/index.html. [Accessed 16 Apr. 2020].
[18] “Predicting wine quality,” 7 Dec. 2012. [Online]. Available: http://fastml.com/predicting-wine-quality/. [Accessed 16 Apr. 2020].
[19] J. M. &. T. Sincich, Statistics, London: Pearson Education Limited, 2017.
[20] “Ep 291: Vinho Verde (has so much more to it than you know!),” A Cast, [Online]. Available: https://play.acast.com/s/winefornormalpeople/d4caaa0ea638463d8707eb26db5ee17a. [Accessed 24 Apr. 2020].
[21] “PH y Vino,” Aprender de Vino, [Online]. Available: https://www.aprenderdevino.es/ph-y-vino/. [Accessed 17 Apr. 2020].
[22] “Acidez volátil en vinos,” El Blog de QuecusLab, [Online]. Available: https://quercuslab.es/blog/determinacion-acidez-volatil-en-vinos/. [Accessed 17 Apr. 2020].
[23] N. Hale, “What is Acidity in Wine?,” [Online]. Available: https://www.winemag.com/2019/06/19/what-is-acidity-in-wine/. [Accessed 25 Apr. 2020].
[24] “Sulphur Dioxide and Sulphites,” Food Safety Authority of Ireland, [Online]. Available: https://www.fsai.ie/faq/additives/sulphur_dioxide_sulphites.html. [Accessed 17 Apr. 2020].
[25] 1. R. L. M. M.-C. N. H. E.-H. G. K. S. G. M. E. a. D. K. T. Tanya M. Monro, “Sensing Free Sulfur Dioxide in Wine,” US National Library of Medicine National Institutes of Health , 6 Aug. 2012. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3472855/. [Accessed 17 Apr. 2020].
[26] “Sabes que son los sulfitos?,” Vinetur, [Online]. Available: https://www.vinetur.com/2015040218835/sabes-que-son-los-sulfitos.html. [Accessed 17 Apr. 2020].
[27] M. Puckette, “Wine Folly.,” [Online]. Available: https://winefolly.com. [Accessed 18 Mar. 2020].
[28] B. Gold, “A Handy Guide To The Alcohol Content in Every Type of Wine—Because Information Is Power,” 11 Jul. 2019. [Online]. Available: https://www.realsimple.com/holidays-entertaining/entertaining/food-drink/alcohol-content-wine. [Accessed 25 Apr. 2020].
[29] G. G. &. H. Wickham, R for Data Science, Sebastopol: O-Reilly, 2017.
[30] R. Stinerock, Statictics with R, A Beguinner's Guide, London: SAGE Publications Ltd, 2018.
[31] K. MacNeil, The Wine Bible, New York: Workman Publishing CO, INC., 2015.
[32] Trust, Wine & Spirit Education, Wines and Spirits, Uckfield: Looking Behind the Label, 2017.
[33] “Tidyverse,” [Online]. Available: https://www.tidyverse.org/. [Accessed 17 Apr. 2020].
[34] Comissao de Viticultura da Regiao Dos Vinhos Verdes, “Vinho Verde,” [Online]. Available: https://www.vinhoverde.pt/en/homepage. [Accessed 17 Apr. 2020].
[35] “RGB Color Codes Chart,” RapidTables, [Online]. Available: http://rapidtables.com/web/color/RGB_Color.html. [Accessed 17 Apr. 2020].
[36] “Scientific Notation Converter,” JustinTools, [Online]. Available: https://www.justintools.com/calculators/scientificnumber.php. [Accessed 19 Apr. 2020].

[37] R. Martin, “Using Linear Regression for Predictive Modeling in R,” DataQuest, [Online]. Available: https://www.dataquest.io/blog/statistical-learning-for-predictive-modeling-r/. [Accessed 5 Apr. 2020].
[38] C. G. &. R. Lovelace, “Efficient R programming,” Dec. 2016. [Online]. Available: https://csgillespie.github.io/efficientR/5-3-importing-data.html. [Accessed 13 Apr. 2020].
[39] C. A. Engel, “Data Wrangling with R,” Oct. 2019. [Online]. Available: https://cengel.github.io/R-data-wrangling/data-visualization-with-ggplot2.html. [Accessed 13 Apr. 2020].
[40] “Side by Side plots with ggplot2,” StackOverflow, [Online]. Available: https://stackoverflow.com/questions/1249548/side-by-side-plots-with-ggplot2. [Accessed 13 Apr. 2020].
[41] Kassambara, “GGplot Colors Best Tricks You Will Love,” Data Novia, [Online]. Available: https://www.datanovia.com/en/blog/ggplot-colors-best-tricks-you-will-love/. [Accessed 17 Apr. 2020].
[42] D. Ranzolin, “Creating New Variables in R with mutate() and ifelse(),” 6 Oct. 2015. [Online]. Available: https://rstudio-pubs-static.s3.amazonaws.com/116317_e6922e81e72e4e3f83995485ce686c14.html#/. [Accessed 17 Apr. 2020].
[43] “Drop columns in R using Dplyr – drop variables,” DataSicience Made Simple, [Online]. Available: http://www.datasciencemadesimple.com/drop-variables-columns-r-using-dplyr/. [Accessed 17 Apr. 2020].
[44] “Parametros Analiticos,” Instituto da Vinha e do Vinho, [Online]. Available: https://www.ivv.gov.pt/np4/np4/89. [Accessed 17 Apr. 2020].
[45] A. Soetewey, “How to do a t-test or ANOVA for more than one variable at once in R and communicate the results in a better way,” Mar 2019. [Online]. Available: https://towardsdatascience.com/how-to-do-a-t-test-or-anova-for-many-variables-at-once-in-r-and-communicate-the-results-in-a-6defaa712e5. [Accessed 20 Apr. 2020].
[46] P. R. d. l. Santos, “Tipos de aprendizaje en Machine Learning: supervisado y no supervisado,” 16 Nov. 2017. [Online]. Available: https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje/. [Accessed 21 Apr. 2020].
[47] J. O. Alvear, “Arboles de Decisión - Parte I,” 12-16 Nov. 2018. [Online]. Available: https://bookdown.org/content/2031/arboles-de-decision-parte-i.html. [Accessed 21 Apr. 2020].
[48] Kassambara, “Add P-values and Significance Levels to ggplots,” STHDA Statistical tools for high-throughput data analysis, 31 Aug. 2017. [Online]. Available: http://www.sthda.com/english/articles/24-ggpubr-publication-ready-plots/76-add-p-values-and-significance-levels-to-ggplots/. [Accessed 21 Apr. 2020].
[49] “Error in shapiro.test : sample size must be between,” Stackoverflow, Feb. 2015. [Online]. Available: https://stackoverflow.com/questions/28217306/error-in-shapiro-test-sample-size-must-be-between. [Accessed 21 Apr. 2020].
[50] “Acidez Total del Vino,” BODEGAS URBINA - LA RIOJA, 31 Jan. 2011. [Online]. Available: http://urbinavinos.blogspot.com/2011/01/acidez-total.html. [Accessed 22 Apr. 2020].
[51] “What’s the difference between sulfates and sulfites?,” Wine Spectator, 14 Apr. 2017. [Online]. Available: https://www.winespectator.com/articles/difference-between-sulfites-sulfates-wine-54706. [Accessed 22 Apr. 2020].

[52] D. Vinny, “What's in Wine?,” UCDavis, [Online]. Available: https://waterhouse.ucdavis.edu/whats-in-wine. [Accessed 5 Apr. 2020].
[53] E. Brown, “Making Wine with High and Low pH Juice,” 11 Nov. 2017. [Online]. Available: https://aces.nmsu.edu/ces/viticulture/documents/making-wine-with-high-and-low-ph-juiceada.pdf. [Accessed 22 Apr. 2020].
[54] “R Kruskal-Wallis test,” R Statistics and Research, 7 Oct 2017. [Online]. Available: https://www.youtube.com/watch?v=Y1qeAFAV5yQ. [Accessed 22 Apr. 2020].
[55] Z. Geer, “A Comprehensive Guide to Random Forest in R,” 8 Nov. 2019. [Online]. Available: https://dzone.com/articles/a-comprehensive-guide-to-random-forest-in-r. [Accessed 25 Apr. 2020].
[56] J. Brownlee, “Your First Machine Learning Project in R Step-By-Step,” Machine Learning Mastery, 8 Oct. 2019. [Online]. Available: https://machinelearningmastery.com/machine-learning-in-r-step-by-step/. [Accessed 25 Apr. 2020].
[57] J. Brownlee, “What is a Confusion Matrix in Machine Learning,” Machine Leaning Mastery, 12 Jan. 2020. [Online]. Available: https://machinelearningmastery.com/confusion-matrix-machine-learning/. [Accessed 25 Apr. 2020].
[58] D. Radecic, “A Non-Confusing Guide to Confusion Matrix,” 28 Sep. 2019. [Online]. Available: https://towardsdatascience.com/a-non-confusing-guide-to-confusion-matrix-7071d2c2204f. [Accessed 25 Apr. 2020].
[59] “Density Curves and their Properties (5.1),” Simple Learning Pro, 28 Jun. 2018. [Online]. Available: https://www.youtube.com/watch?v=Txlm4ORI4Gs. [Accessed 18 Apr. 2020].
[60] J. Brown, “Interpreting R Output For Simple Linear Regression Part 1 (EPSY 5262),” 31 Jan 2016. [Online]. Available: https://www.youtube.com/watch?v=u7TxjUI4PRI. [Accessed 19 Apr. 2020].
[61] J. Brown, “ 0:02 / 12:09 Interpreting R Output for Simple Linear Regression Part 2 (EPSY 5262),” 7 Feb. 2016. [Online]. Available: https://www.youtube.com/watch?v=sKW2umonEvY. [Accessed 18 Apr. 2020].
[62] B. Foltz, “Statistics 101: Is My Data Normal?,” 27 Jan. 2013. [Online]. Available: https://www.youtube.com/watch?v=9IcaQwQkE9I. [Accessed 20 Apr. 2020].
[63] M. E. Clapham, “9: Shapiro-Wilk test,” 18 Jan. 2016. [Online]. Available: https://www.youtube.com/watch?v=dRAqSsgkCUc. [Accessed 20 Apr. 2020].
[64] P. Chan, “Kruskal-Wallis post hoc test in R interpretation,” 21 Nov. 2012. [Online]. Available: https://www.youtube.com/watch?v=3n15iBJar4E. [Accessed 23 Apr. 2020].
[65] P. Chan, “Mann-Whitney U test interpretation in SPSS,” 10 Sep. 2012. [Online]. Available: https://www.youtube.com/watch?v=nUudt9ffUP4. [Accessed 23 Apr. 2020].
[66] J. Lambert, “SPSS: Non-parametric Mann-Whitney U Test - Part 1,” Maths and Stats, 22 Sep. 2017. [Online]. Available: https://www.youtube.com/watch?v=xHXYR5m-Od0. [Accessed 23 Apr. 2020].
[67] J. Lambert, “SPSS: Non-parametric Mann-Whitney U Test - Part 2,” Maths and Stats, 22 Sep. 2017. [Online]. Available: https://www.youtube.com/watch?v=LYKVsVgV3Ng. [Accessed 23 Apr. 2020].
[68] “Confusion Matrices - Machine Learning with caret,” Dragonfly Statistics, 16 Oct. 2017. [Online]. Available: https://www.youtube.com/watch?v=4TH5XTw8lrE. [Accessed 23 Apr. 2020].

[69] “Random Forest prediction model in R,” [Online]. Available: https://rpubs.com/markloessi/498787. [Accessed 25 Apr. 2020].
[70] V. Shelunts, “Wine Quality Decision Tree and Random Forest,” 2018. [Online]. Available: https://www.kaggle.com/vshelunts/wine-quality-decision-tree-and-random-forest. [Accessed 24 Apr. 2020].
[71] M. P. &. E. Corrales, “Analisis Calidad de Vino,” 13 Apr. 2019. [Online]. Available: https://rpubs.com/mparedess/486447. [Accessed 17 Apr. 2020].
[72] J. P. Espinosa, “Vino y algo de python(y pandas),” [Online]. Available: http://javierpalmaespinosa.cl/science/vino-python. [Accessed 18 Apr. 2020].
[73] J. Fernández, “Data_preprocess,” 5 Jan. 2019. [Online]. Available: https://github.com/joaquinfdez/Data_preprocess. [Accessed 18 Apr. 2020].
[74] F. P. C. Flores, “Beginners Guide to EDA and Random Forest using R,” 5 Dec. 2017. [Online]. Available: https://www.kaggle.com/grosvenpaul/beginners-guide-to-eda-and-random-forest-using-r. [Accessed 20 Apr. 2020].
[75] S. Khan, “Quality Analysis of Vinho Verde Wines,” 1 May 2015. [Online]. Available: https://saadkhan321.github.io/docs/ProjectReport3_DataAnalyst_NanoDegree.html . [Accessed 15 Apr. 2020].

Code

```
#SOURCES
################################################################################################################
###
################################################################################################################
###
#wine-quality-red https://www.openml.org/d/40691
#wine-quality-white https://www.openml.org/d/40498

#LIBRARIES NEEDED
################################################################################################################
###
################################################################################################################
###
library(tidyverse) #tidiversey contains other packages as
library(tidyr)
library(tibble)
library(ggplot2) #graphics
library(readr) #library to read csv
library(dplyr)
library(openintro)
library(tools)
library(GGally)
library(forcats)
library(ggpubr)
library(mvShapiroTest)
library(gridExtra)
library(moments)
library(cowplot) #add different plots together
library(nortest) # to be able to perform Anderson-Darling normality test
library(rstatix)
library(FSA) #Dunn Test
library(ggcorrplot) #correlation matrix
library(rpart)
library(rpart.plot)
library(caret)
library(e1071)
library(randomForest)
library(gdata)
library(DiagrammeR)

#IMPORTING DATA SETS
################################################################################################################
###
################################################################################################################
###
redwine <- read.csv("wine-quality-red.csv", head=TRUE, sep=",")
whitewine <- read.csv("wine-quality-white.csv", head=TRUE, sep=",")

#DATA SET EXPLORATION
################################################################################################################
###
################################################################################################################
###

#Check the number of rows of each data set
nrow(redwine)
nrow(whitewine)

#Check first rows of each of the data sets
head(redwine)
```

```
head(whitewine)

#Check the last rows of each of the data sets
tail(redwine)
tail(whitewine)

#Information: Number of observations. Number of valiables.
#Variables name and class, as well as some of the data for each of them.
str(redwine)
str(whitewine)

#Check only names of variables for each of the data sets
names(redwine) #class variable will be more meaningful with the name quality
names(whitewine) #We will need to change the names so are the same as in the other data set

#Check if the data sets has any missing value
redwine[!complete.cases(redwine), ] #no missing rows
whitewine[!complete.cases(whitewine), ] #no missing rows



#CLEANING DATA / DATA TRANSFORMATION
#########################################################################################################
###
#########################################################################################################
###
#Renaming both files with same variable names
redwine_renamed <- redwine %>% rename("quality" = "class",
                                      "ph" = "pH")
whitewine_renamed <- whitewine %>% rename(
  "fixed_acidity" = "V1",
  "volatile_acidity" = "V2",
  "citric_acid" = "V3",
  "residual_sugar" = "V4",
  "chlorides" = "V5",
  "free_sulfur_dioxide" = "V6",
  "total_sulfur_dioxide" = "V7",
  "density" = "V8",
  "ph" = "V9",
  "sulphates" = "V10",
  "alcohol" = "V11",
  "quality" = "Class")

#Check the names of variables again to ensure they were changed
names(redwine_renamed)
names(whitewine_renamed)



#Changing "Quality" from integer to factor with ordered levels
#########################################################################################################
###
#Checking current class of variable
class(redwine_renamed$quality)
#Changing quality from character to factor
redwine_renamed <- mutate_at(redwine_renamed,vars(quality), as.factor)
#Adding levels to factor
redwine_renamed$quality <- factor(redwine_renamed$quality, levels = c("1","2","3","4","5","6","7","8","9","10"))
#Check if the variable is now a factor with levels
class(redwine_renamed$quality)
levels(redwine_renamed$quality)
```

```
#Checking current class of variable
class(whitewine_renamed$quality)
#Changing quality from character to factor
whitewine_renamed <- mutate_at(whitewine_renamed,vars(quality), as.factor)
#Adding levels to factor
whitewine_renamed$quality <- factor(whitewine_renamed$quality, levels = c("1","2","3","4","5","6","7","8","9","10"))
#Check if the variable is now a factor with levels
class(whitewine_renamed$quality)
levels(whitewine_renamed$quality)



#Creating new variable "Type" which will include if the wine is red or white and changing it from character to factor
#########################################################################################################
###
#New variable for red wine
redwine_renamed["type"] = "red"
#Checking class of new variable
class(redwine_renamed$type)
#Change from character to factor
redwine_renamed <- mutate_at(redwine_renamed, vars(type), as.factor)
#Check if the variable is now a factor
class(redwine_renamed$type)

#New variable for white wine
whitewine_renamed["type"] = "white"
#checking class of new variable
class(whitewine_renamed$type)
#Change from character to factor
whitewine_renamed <- mutate_at(whitewine_renamed, vars(type), as.factor)

#Check if the variable is now a factor
class(whitewine_renamed$type)


#Creating new valiable "Type by Sugar Level", which will help understanding the data set better
#########################################################################################################
###
#Following the guidance of https://en.wikipedia.org/wiki/Sweetness_of_wine
#Creating new variable Type by Sugar Level in red wine set
redwine_renamed <- redwine_renamed %>%
  add_column(type_by_sugar_level = ifelse (redwine_renamed$residual_sugar <= 4 ,"Dry",
                                ifelse (redwine_renamed$residual_sugar >4 &  redwine_renamed$residual_sugar <=12,"Medium
Dry",
                                        ifelse (redwine_renamed$residual_sugar >12 &  redwine_renamed$residual_sugar
<=45,"Medium","Sweet"))))
#Chaning new variable from character to factor
redwine_renamed <- mutate_at(redwine_renamed, vars(type_by_sugar_level), as.factor)
#Adding ordered levels to factor variable
redwine_renamed$type_by_sugar_level <- factor(redwine_renamed$type_by_sugar_level, levels = c("Dry","Medium
Dry","Medium","Sweet"))
#Check if the variable is now a factor with levels
class(redwine_renamed$type_by_sugar_level)
levels(redwine_renamed$type_by_sugar_level)
#Checking number of samples inside each of the newest created variable
redwine_renamed %>%
  group_by(redwine_renamed$type_by_sugar_level) %>%
  summarise(n = n())
```

```
#Creating new variable Type by Sugar Level in white wine set
whitewine_renamed <- whitewine_renamed %>%
  add_column(type_by_sugar_level =ifelse (whitewine_renamed$residual_sugar <=4,"Dry",
                                ifelse (whitewine_renamed$residual_sugar >4 &  whitewine_renamed$residual_sugar
<=12,"Medium Dry",
                                        ifelse (whitewine_renamed$residual_sugar >12 &  whitewine_renamed$residual_sugar
<=45,"Medium","Sweet"))))
#Chaning new variable from character to factor
whitewine_renamed <- mutate_at(whitewine_renamed, vars(type_by_sugar_level), as.factor)
#Adding ordered levels to factor variable
whitewine_renamed$type_by_sugar_level <- factor(whitewine_renamed$type_by_sugar_level, levels = c("Dry","Medium
Dry","Medium","Sweet"))
#Check if the variable is now a factor with levels
class(whitewine_renamed$type_by_sugar_level)
levels(whitewine_renamed$type_by_sugar_level)
#Checking number of samples inside each of the newest created variable
whitewine_renamed %>%
  group_by(whitewine_renamed$type_by_sugar_level) %>%
  summarise(n = n())



#Creating new valiable numerical variable "Total Acidity", which will help understanding the data set better
##############################################################################################################
###
#Creating new variable Total Acidity for red wine
redwine_renamed <- redwine_renamed %>%
  mutate(total_acitity = fixed_acidity + volatile_acidity)
#Checking the variable was created
head(redwine_renamed)
#Checking the class is numeric
class(redwine_renamed$total_acitity)


#Creating new variable Total Acidity for white wine
whitewine_renamed <- whitewine_renamed %>%
  mutate(total_acitity = fixed_acidity + volatile_acidity)
#Checking the variable was created
head(whitewine_renamed)
#Checking the class is numeric
class(whitewine_renamed$total_acitity)



#Creating new variable "PH Level" as factor with levels
##############################################################################################################
###
#Creating new variable for red wine
redwine_renamed <- redwine_renamed %>%
  add_column(ph_level = ifelse (redwine_renamed$ph <= 2.8 ,"PH<=2.8",
                        ifelse (redwine_renamed$ph >2.8 &  redwine_renamed$ph <=3,"2.8>PH<=3",
                              ifelse (redwine_renamed$ph >3 &  redwine_renamed$ph <=3.5,"3>PH<=3.5",
                                    ifelse (redwine_renamed$ph >3.5 &  redwine_renamed$ph <=4,"3>PH<=4", "PH>4")))))
#Checking class for new variable
class(redwine_renamed$ph_level)
#Change from character to factor
redwine_renamed <- mutate_at(redwine_renamed, vars(ph_level), as.factor)
#Add levels to the factor
```

```
redwine_renamed$ph_level <- factor(redwine_renamed$ph_level, levels =
c("PH<=2.8","2.8>PH<=3","3>PH<=3.5","3>PH<=4","PH>4"))
#Check class again to confirm is a factor
class(redwine_renamed$ph_level)
#Check levels of factor
levels(redwine_renamed$ph_level)
#Check number of samples for each type of PH category
redwine_renamed %>%
 group_by(redwine_renamed$ph_level) %>%
 summarise(n = n())
```

```
#Creating new variable for white wine
whitewine_renamed <- whitewine_renamed %>%
 add_column(ph_level = ifelse (whitewine_renamed$ph <= 2.8 ,"PH<=2.8",
                        ifelse (whitewine_renamed$ph >2.8 &  whitewine_renamed$ph <=3,"2.8>PH<=3",
                              ifelse (whitewine_renamed$ph >3 &  whitewine_renamed$ph <=3.5,"3>PH<=3.5",
                                    ifelse (whitewine_renamed$ph >3.5 &  whitewine_renamed$ph <=4,"3>PH<=4", "PH>4")))))
#Checking class for new variable
class(whitewine_renamed$ph_level)
#Change from character to factor
whitewine_renamed <- mutate_at(whitewine_renamed, vars(ph_level), as.factor)
#Add levels to the factor
whitewine_renamed$ph_level <- factor(whitewine_renamed$ph_level, levels =
c("PH<=2.8","2.8>PH<=3","3>PH<=3.5","3>PH<=4","PH>4"))
#Check class again to confirm is a factor
class(whitewine_renamed$ph_level)
#Check levels of factor
levels(whitewine_renamed$ph_level)
#Check number of samples for each type of PH category
whitewine_renamed %>%
 group_by(whitewine_renamed$ph_level) %>%
 summarise(n = n())
```

```
#Creating new variable "Alcohol Level" as factor with levels
#########################################################################################################
###
#Following https://www.realsimple.com/holidays-entertaining/entertaining/food-drink/alcohol-content-wine
#Creating new variable for red wine
redwine_renamed <- redwine_renamed %>%
 add_column(alcohol_level = ifelse (redwine_renamed$alcohol <= 12.5 ,"Very Low",
                        ifelse (redwine_renamed$alcohol >12.5 &  redwine_renamed$alcohol <=13.5,"Moderately Low",
                              ifelse (redwine_renamed$alcohol >13.5 &  redwine_renamed$alcohol <=14.5,"High", "Very High"))))
#Checking class for new variable
class(redwine_renamed$alcohol_level)
#Change from character to factor
redwine_renamed <- mutate_at(redwine_renamed, vars(alcohol_level), as.factor)
#Add levels to the factor
redwine_renamed$alcohol_level <- factor(redwine_renamed$alcohol_level, levels = c("Very Low","Moderately
Low","High","Very High"))
#Check class again to confirm is a factor
class(redwine_renamed$alcohol_level)
#Check levels of factor
levels(redwine_renamed$alcohol_level)
#Check number of samples for each type of PH category
redwine_renamed %>%
 group_by(redwine_renamed$alcohol_level) %>%
 summarise(n = n())
```

```
#Creating new variable for white wine
whitewine_renamed <- whitewine_renamed %>%
  add_column(alcohol_level = ifelse (whitewine_renamed$alcohol <= 12.5 ,"Very Low",
                           ifelse (whitewine_renamed$alcohol >12.5 &  whitewine_renamed$alcohol <=13.5,"Moderately Low",
                                ifelse (whitewine_renamed$alcohol >13.5 &  whitewine_renamed$alcohol <=14.5,"High", "Very
High"))))
#Checking class for new variable
class(whitewine_renamed$alcohol_level)
#Change from character to factor
whitewine_renamed <- mutate_at(whitewine_renamed, vars(alcohol_level), as.factor)
#Add levels to the factor
whitewine_renamed$alcohol_level <- factor(whitewine_renamed$alcohol_level, levels = c("Very Low","Moderately
Low","High","Very High"))
#Check class again to confirm is a factor
class(whitewine_renamed$alcohol_level)
#Check levels of factor
levels(whitewine_renamed$alcohol_level)
#Check number of samples for each type of PH category
whitewine_renamed %>%
  group_by(whitewine_renamed$alcohol_level) %>%
  summarise(n = n())



#CLEAN DATA CHECK
##############################################################################################################
###
##############################################################################################################
###
str(redwine_renamed)
str(whitewine_renamed)

#Summary as all the variables are numerical, informs about Minimun Value, 1st Quartile, Median, Mean, 3rd Quartile and
Maximum Value for each of the variables.
summary(redwine_renamed)
summary(whitewine_renamed)




#COMBINE DATA SETS
##############################################################################################################
###
##############################################################################################################
###
#Combine both data sets
head(redwine_renamed)
head(whitewine_renamed)
winequality <- rbind(redwine_renamed,whitewine_renamed)
as.data.frame(winequality)
head(winequality)
tail(winequality)
```

```
#ABOUT THE DATA
#########################################################################################################
###
#########################################################################################################
###

#Create bar plot for red and white wines based on "Quality"
data_quality <- ggplot(winequality, aes(x = quality)) +
  labs(title = "Quality of wines", x = "Quality",y="Count",fill="Type")+
  geom_bar(aes(fill = type), alpha = 0.5) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))



#Create bar plot for red and white wines based on "Total Acidity"
data_total_acidity <- ggplot(winequality, aes(x = total_acitity)) +
  labs(title = "Total Acidity (g/l) in wines",x = "Total Acidity (g/l)",y="Count",fill="Type")+
  geom_histogram(aes(fill = type), alpha = 0.5, binwidth = 0.1) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for red and white wines based on "Volatile Acidity"
data_volatile_acidity <- ggplot(winequality, aes(x = volatile_acidity)) +
  labs(title = "Volatile Acidity (g/l) in wines",x = "Volatile Acidity (g/l)",y="Count",fill="Type")+
  geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 0.1) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for red and white wines based on "Fixed Acidity"
data_fixed_acidity <- ggplot(winequality, aes(x = fixed_acidity)) +
  labs(title = "Fixed Acidity (g/l) in wines",x = "Fixed Acidity (g/l)",y="Count",fill="Type")+
  geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 0.1) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for red and white wines based on "PH Level"
data_ph_level <- ggplot(winequality, aes(x = ph_level)) +
  labs(title = "pH Level in wines", x="pH Level",y="Count",fill="Type")+
  geom_bar(aes(fill = type), alpha = 0.5)+
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))



#Create bar plot for red and white wines based on "Sulphates"
data_sulphates <- ggplot(winequality, aes(x = sulphates)) +
  labs(title = "Sulphates (g/l) in wines",x = "Sulphates (g/l)",y="Count",fill="Type")+
  geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 0.1) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for red and white wines based on "Total Sulfur Dioxide"
data_total_sulfur_dioxide <- ggplot(winequality, aes(x = total_sulfur_dioxide)) +
  labs(title = "Total Sulfur Dioxide in wines",x = "Total Sulfur Dioxide (mg/l)",y="Count",fill="Type")+
```

```
  geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 5) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for red and white wines based on "Free Sulfur Dioxide"
data_free_sulfur_dioxide <- ggplot(winequality, aes(x = free_sulfur_dioxide)) +
  labs(title = "Free Sulfur Dioxide (mg/l) in wines",x = "Free Sulfur Dioxide (mg/l)",y="Count",fill="Type")+
  geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 5) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))




#Create bar plot for red and white wines based on "Alcohol Level"
data_alcohol<- ggplot(winequality, aes(x = alcohol_level)) +
  labs(title = "Alcohol Level (%) in wines", x="Alcohol Level (%)",y="Count",fill="Type")+
  geom_bar(aes(fill = type), alpha = 0.5)+
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for red and white wines based on "Density"
data_density <- ggplot(winequality, aes(x = density)) +
  labs(title = "Density (g/ml) in wines", x = "Density (g/ml)",y="Count",fill="Type")+
  geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 0.001) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for red and white wines based on "Residual Sugar Level"
data_sugar_level <- ggplot(winequality, aes(x = type_by_sugar_level)) +
  labs(title = "Type of wine by Sugar Level", x = "Type by Sugar Level",y="Count",fill="Type")+
  geom_bar(aes(fill = type), alpha = 0.5) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for red and white wines based on "Chlorides"
data_chlorides <- ggplot(winequality, aes(x = chlorides)) +
  labs(title = "Chlorides in wines", x = "Chlorides",y="Count",fill="Type")+
  geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 0.1) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))




#After creating all the different bar plots for each of the variables I add some of them together for better visualisation
data_quality

plot_grid(data_total_acidity,data_ph_level, nrow = 1)

plot_grid(data_fixed_acidity, data_volatile_acidity, nrow = 1)

data_sulphates

plot_grid(data_total_sulfur_dioxide,data_free_sulfur_dioxide)
```

```
plot_grid(data_sugar_level,data_alcohol, nrow = 1)

plot_grid(data_density,data_chlorides, nrow = 1)




#CREATING SUBSET WITH ONLY DRY WINES
#########################################################################################################
###
#########################################################################################################
###
#Subset the combined data set and chose just dry wines
winequality_dry <- subset(winequality, winequality$type_by_sugar_level =="Dry")

#Subset the red wine data set
redwine_renamed_dry <- subset(redwine_renamed, redwine_renamed$type_by_sugar_level == "Dry")

#Subset the red wine data set
whitewine_renamed_dry <- subset(whitewine_renamed, whitewine_renamed$type_by_sugar_level == "Dry")


#information about new data sets
str(winequality_dry)
str(redwine_renamed_dry)
str(whitewine_renamed_dry)




#ABOUT DRY VINHO VERDE
#########################################################################################################
###
#########################################################################################################
###


#Create bar plot for dry red and white wines based on "Quality"
data_quality_dry <- ggplot(winequality_dry, aes(x = quality)) +
 labs(title = "Quality of dry wines", x = "Quality",y="Count",fill="Type")+
 geom_bar(aes(fill = type), alpha = 0.5) +
 scale_color_manual(values = c("#990000","#CCCC00"))+
 scale_fill_manual(values = c("#990000","#CCCC00"))+
 theme(plot.title = element_text(hjust = 0.5))




#Create bar plot for dry red and white wines based on "Total Acidity"
data_total_acidity_dry <- ggplot(winequality_dry, aes(x = total_acitity)) +
 labs(title = "Total Acidity (g/l) in dry wines",x = "Total Acidity (g/l)",y="Count",fill="Type")+
 geom_histogram(aes(fill = type), alpha = 0.5, binwidth = 0.1) +
 scale_color_manual(values = c("#990000","#CCCC00"))+
 scale_fill_manual(values = c("#990000","#CCCC00"))+
 theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for dry red and white wines based on "Volatile Acidity"
data_volatile_acidity_dry <- ggplot(winequality_dry, aes(x = volatile_acidity)) +
 labs(title = "Volatile Acidity (g/l) in dry wines",x = "Volatile Acidity (g/l)",y="Count",fill="Type")+
 geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 0.1) +
```

```
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for dry red and white wines based on "Fixed Acidity"
data_fixed_acidity_dry <- ggplot(winequality_dry, aes(x = fixed_acidity)) +
  labs(title = "Fixed Acidity (g/l) in dry wines",x = "Fixed Acidity (g/l)",y="Count",fill="Type")+
  geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 0.1) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for dry red and white wines based on "PH Level"
data_ph_level_dry <- ggplot(winequality_dry, aes(x = ph_level)) +
  labs(title = "pH Level in wines", x="pH Level",y="Count",fill="Type")+
  geom_bar(aes(fill = type), alpha = 0.5)+
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))



#Create bar plot for dry red and white wines based on "Sulphates"
data_sulphates_dry <- ggplot(winequality_dry, aes(x = sulphates)) +
  labs(title = "Sulphates (g/l) in dry wines",x = "Sulphates (g/l)",y="Count",fill="Type")+
  geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 0.1) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for dry red and white wines based on "Total Sulfur Dioxide"
data_total_sulfur_dioxide_dry <- ggplot(winequality_dry, aes(x = total_sulfur_dioxide)) +
  labs(title = "Total Sulfur Dioxide in dry wines",x = "Total Sulfur Dioxide (mg/l)",y="Count",fill="Type")+
  geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 5) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for dry red and white wines based on "Free Sulfur Dioxide"
data_free_sulfur_dioxide_dry <- ggplot(winequality_dry, aes(x = free_sulfur_dioxide)) +
  labs(title = "Free Sulfur Dioxide (mg/l) in dry wines",x = "Free Sulfur Dioxide (mg/l)",y="Count",fill="Type")+
  geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 5) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))



#Create bar plot for dry red and white wines based on "Alcohol Level"
data_alcohol_dry<- ggplot(winequality_dry, aes(x = alcohol_level)) +
  labs(title = "Alcohol Level (%) in wines", x="Alcohol Level (%)",y="Count",fill="Type")+
  geom_bar(aes(fill = type), alpha = 0.5)+
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for dry red and white wines based on "Density"
data_density_dry <- ggplot(winequality_dry, aes(x = density)) +
  labs(title = "Density (g/ml) in wines", x = "Density (g/ml)",y="Count",fill="Type")+
  geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 0.001) +
```

```
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for dry red and white wines based on "Residual Sugar Level"
data_sugar_level_dry <- ggplot(winequality_dry, aes(x = type_by_sugar_level)) +
  labs(title = "Type of wine by Sugar Level", x = "Type by Sugar Level",y="Count",fill="Type")+
  geom_bar(aes(fill = type), alpha = 0.5) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))
#
#Create bar plot for dry red and white wines based on "Chlorides"
data_chlorides_dry <- ggplot(winequality_dry, aes(x = chlorides)) +
  labs(title = "Chlorides in wines", x = "Chlorides",y="Count",fill="Type")+
  geom_histogram(aes(fill = type), alpha = 0.5,binwidth = 0.1) +
  scale_color_manual(values = c("#990000","#CCCC00"))+
  scale_fill_manual(values = c("#990000","#CCCC00"))+
  theme(plot.title = element_text(hjust = 0.5))



#After creating all the different bar plots for each of the variables I add some of them together for better visualisation
data_quality_dry

plot_grid(data_total_acidity_dry,data_ph_level_dry, nrow = 1)

plot_grid(data_fixed_acidity_dry, data_volatile_acidity_dry, nrow = 1)

data_sulphates_dry

plot_grid(data_total_sulfur_dioxide_dry,data_free_sulfur_dioxide_dry)

plot_grid(data_sugar_level_dry,data_alcohol_dry, nrow = 1)

plot_grid(data_density_dry,data_chlorides_dry, nrow = 1)




#TESTING NORMALITY
#########################################################################################################
###
#########################################################################################################
###
#qq plots for numerical variables red wine
qq_red_fixed_acidity <- ggqqplot(redwine_renamed$fixed_acidity, color = "#990000")+
  labs(x="Theoretical Fixed Acidity", y="Sample Fixed Acidity")
qq_red_citric_acid <- ggqqplot(redwine_renamed$citric_acid, color = "#990000")+
  labs(x="Theoretical Citric Acid", y="Sample Citric Acid")
qq_red_volatile_acidity <- ggqqplot(redwine_renamed$volatile_acidity, color = "#990000")+
  labs(x="Theoretical Volatile Acidity", y="Sample Volatile Acidity")
qq_red_total_acidity <- ggqqplot(redwine_renamed$total_acitity, color = "#990000")+
  labs(x="Theoretical Total Acidity", y="Sample Total Acidity")
qq_red_ph <- ggqqplot(redwine_renamed$ph, color = "#990000")+
  labs(x="Theoretical PH", y="Sample PH")
qq_red_residual_sugar <- ggqqplot(redwine_renamed$residual_sugar, color = "#990000")+
  labs(x="Theoretical Residual Sugar", y="Sample Residual Sugar")
qq_red_alcohol <- ggqqplot(redwine_renamed$alcohol, color = "#990000")+
  labs(x="Theoretical Alcohol", y="Sample Alcohol")
```

```
qq_red_density <- ggqqplot(redwine_renamed$density, color = "#990000")+
  labs(x="Theoretical Density", y="Sample Density")
qq_red_chlorides <- ggqqplot(redwine_renamed$chlorides, color = "#990000")+
  labs(x="Theoretical Chlorides", y="Sample Chlorides")
qq_red_free_sulfur_dioxide <- ggqqplot(redwine_renamed$free_sulfur_dioxide, color = "#990000")+
  labs(x="Theoretical Free Sulphur Dioxide", y="Sample Free Sulphur Dioxide")
qq_red_total_sulfur_dioxide <- ggqqplot(redwine_renamed$total_sulfur_dioxide, color = "#990000")+
  labs(x="Theoretical Total Sulphur Dioxide", y="Sample Total Sulphur Dioxide")
qq_red_sulphates <- ggqqplot(redwine_renamed$sulphates, color = "#990000")+
  labs(x="Theoretical Sulphates", y="Sample Sulphates")
#After creating all the different qq plots for each of the variables we add them together for better visualisation
grid.arrange(qq_red_total_acidity, qq_red_fixed_acidity,qq_red_citric_acid, qq_red_volatile_acidity,
          qq_red_ph, qq_red_sulphates, qq_red_free_sulfur_dioxide, qq_red_total_sulfur_dioxide,
          qq_red_residual_sugar, qq_red_alcohol, qq_red_density, qq_red_chlorides)


#qq plots for numerical variables white wine
qq_white_fixed_acidity <- ggqqplot(whitewine_renamed$fixed_acidity, color = "#CCCC00")+
  labs(x="Theoretical Fixed Acidity", y="Sample Fixed Acidity")
qq_white_citric_acid <- ggqqplot(whitewine_renamed$citric_acid, color = "#CCCC00")+
  labs(x="Theoretical Citric Acid", y="Sample Citric Acid")
qq_white_volatile_acidity <- ggqqplot(whitewine_renamed$volatile_acidity, color = "#CCCC00")+
  labs(x="Theoretical Volatile Acidity", y="Sample Volatile Acidity")
qq_white_total_acidity <- ggqqplot(whitewine_renamed$total_acitity, color = "#CCCC00")+
  labs(x="Theoretical Total Acidity", y="Sample Total Acidity")
qq_white_ph <- ggqqplot(whitewine_renamed$ph, color = "#CCCC00")+
  labs(x="Theoretical PH", y="Sample PH")
qq_white_residual_sugar <- ggqqplot(whitewine_renamed$residual_sugar, color = "#CCCC00")+
  labs(x="Theoretical Residual Sugar", y="Sample Residual Sugar")
qq_white_alcohol <- ggqqplot(whitewine_renamed$alcohol, color = "#CCCC00")+
  labs(x="Theoretical Alcohol", y="Sample Alcohol")
qq_white_density <- ggqqplot(whitewine_renamed$density, color = "#CCCC00")+
  labs(x="Theoretical Density", y="Sample Density")
qq_white_chlorides <- ggqqplot(whitewine_renamed$chlorides, color = "#CCCC00")+
  labs(x="Theoretical Chlorides", y="Sample Chlorides")
qq_white_free_sulfur_dioxide <- ggqqplot(whitewine_renamed$free_sulfur_dioxide, color = "#CCCC00")+
  labs(x="Theoretical Free Sulphur Dioxide", y="Sample Free Sulphur Dioxide")
qq_white_total_sulfur_dioxide <- ggqqplot(whitewine_renamed$total_sulfur_dioxide, color = "#CCCC00")+
  labs(x="Theoretical Total Sulphur Dioxide", y="Sample Total Sulphur Dioxide")
qq_white_sulphates <- ggqqplot(whitewine_renamed$sulphates, color = "#CCCC00")+
  labs(x="Theoretical Sulphates", y="Sample Sulphates")
#After creating all the different qq plots for each of the variables we add them together for better visualisation
grid.arrange(qq_white_total_acidity, qq_white_fixed_acidity,qq_white_citric_acid, qq_white_volatile_acidity,
          qq_white_ph, qq_white_sulphates, qq_white_free_sulfur_dioxide, qq_white_total_sulfur_dioxide,
          qq_white_residual_sugar, qq_white_alcohol, qq_white_density, qq_white_chlorides)


str(redwine_renamed_dry)

#Shapiro-Wilk normality test
#create variable so we can run the test just for the numerical values
x <- c(1:11,15)
#Red wine
for (i in x) {
  print(shapiro.test(redwine_renamed_dry[, i]))
}
#White wine
for (i in x) {
  print(shapiro.test(whitewine_renamed_dry[, i]))
}
```

```
#CORRELATIONS (https://www.dataquest.io/blog/statistical-learning-for-predictive-modeling-r/)
#########################################################################################################
###
#########################################################################################################
###
#Correlation dry red wines
#
x <- c(1:11,15)
#
#Spearman Correlations
corr_spearman_redwine_renamed_dry <- round(cor(redwine_renamed_dry[,x], method = "spearman"), 2)
corr_spearman_redwine_renamed_dry
#
#Kendall Correlations
corr_kendall_redwine_renamed_dry <- round(cor(redwine_renamed_dry[,x], method = "kendall"), 2)
corr_kendall_redwine_renamed_dry
#
#correlation matrix: Spearman Correlations
plot_corr_spearman_redwine_renamed_dry <- ggcorrplot(corr_spearman_redwine_renamed_dry,
        hc.order = TRUE,
        type = "lower",
        lab = TRUE,
        colors = c("#202020","white","#FF0000"))+
        labs (title = "Spearman Correlation Coeficients")+
        theme(plot.title = element_text(hjust = 0.5))
#
#correlation matrix: Kendall Correlations
plot_corr_kendall_redwine_renamed_dry <- ggcorrplot(corr_kendall_redwine_renamed_dry,
        hc.order = TRUE,
        type = "lower",
        lab = TRUE,
        colors = c("#202020","white","#FF0000"))+
        labs (title = "Kendall Correlation Coeficients")+
        theme(plot.title = element_text(hjust = 0.5))
#Combining both matrix
grid.arrange(plot_corr_spearman_redwine_renamed_dry, plot_corr_kendall_redwine_renamed_dry, nrow=1)



#Correlation dry white wines
#
x <- c(1:11,15)
#
#Spearman Correlations
corr_spearman_whitewine_renamed_dry <- round(cor(whitewine_renamed_dry[,x], method = "spearman"), 2)
corr_spearman_whitewine_renamed_dry
#
#Kendall Correlations
corr_kendall_whitewine_renamed_dry <- round(cor(whitewine_renamed_dry[,x], method = "kendall"), 2)
corr_kendall_whitewine_renamed_dry
#
#correlation matrix: Spearman Correlations
plot_corr_spearman_whitewine_renamed_dry <- ggcorrplot(corr_spearman_whitewine_renamed_dry,
        hc.order = TRUE,
        type = "lower",
        lab = TRUE,
        colors = c("#FFFF33","white","#CCCC00"))+
        labs (title = "Spearman Correlation Coeficients")+
        theme(plot.title = element_text(hjust = 0.5))
#
#correlation matrix: Kendall Correlations
plot_corr_kendall_whitewine_renamed_dry <- ggcorrplot(corr_kendall_whitewine_renamed_dry,
        hc.order = TRUE,
```

```
        type = "lower",
        lab = TRUE,
        colors = c("#FFFF33","white","#CCCC00"))+
        labs (title = "Kendall Correlation Coeficients")+
        theme(plot.title = element_text(hjust = 0.5))
#Combining both matrix
grid.arrange(plot_corr_spearman_whitewine_renamed_dry, plot_corr_kendall_whitewine_renamed_dry, nrow=1)




#INSIGHT 1 - IS THERE ANY DIFFERENCE BETWEEN DRY RED AND WHITE VINHO VERDE WINES
#BASED ON THEIR CHEMICAL PROPERTIES?
##############################################################################################################
###
##############################################################################################################
###
#Subset by just dry wines with quality 5
winequality_dry_5 <- subset(winequality_dry, winequality_dry$quality == "5")
redwine_renamed_dry_5 <- subset(redwine_renamed_dry, redwine_renamed_dry$quality == "5")
whitewine_renamed_dry_5 <- subset(whitewine_renamed_dry, whitewine_renamed_dry$quality == "5")



#Descriptive Statistics
#Creationg of function which will give more information than summary function does
#Same as function created in class with kurtosis
Stats <- function(stats){
  newMatrix <- matrix(1:8, nrow=1) #creating a blank matrix
  colnames(newMatrix) <- c("Mean","Median","Variance","Standard
Deviation","Minimum","Maximum","Skewness","Kurtosis")
  rownames(newMatrix) <- "Stats"
  newMatrix[1, ] <- c(mean(stats),median(stats),var(stats),sd(stats),min(stats),max(stats),skewness(stats),kurtosis(stats))
  newMatrix
}


#Using stats for both subsets
#
x <- c(1:11,15)
#
for (i in x) {
  print(Stats(redwine_renamed_dry_5[, i]))
}
#
for (i in x) {
  print(Stats(whitewine_renamed_dry_5[, i]))
}




#Density plots
#
#Fixed Acidity
density_red_white_fixed_acidity <- ggplot(winequality_dry_5, aes(x = fixed_acidity, color = type)) +
  geom_density()+
  geom_vline(aes(xintercept=mean(fixed_acidity)),
             color="black", linetype="dashed", size=1)+
  labs (x = "Fixed Acidity (g/l)", y = "Density", color = "Type")+
  scale_color_manual(values=c("#990000","#CCCC00"))
```

```
#
#Volatile Acidity
density_red_white_volatile_acidity <- ggplot(winequality_dry_5, aes(x = volatile_acidity, color = type)) +
 geom_density()+
 geom_vline(aes(xintercept=mean(volatile_acidity)),
            color="black", linetype="dashed", size=1)+
 labs (x = "Volatile Acidity (g/l)", y = "Density", color = "Type")+
 scale_color_manual(values=c("#990000","#CCCC00"))
#
#Citric Acid
density_red_white_citric_acid <- ggplot(winequality_dry_5, aes(x = citric_acid, color = type)) +
 geom_density()+
 geom_vline(aes(xintercept=mean(citric_acid)),
            color="black", linetype="dashed", size=1)+
 labs (x = "Citric Acid (g/l)", y = "Density", color = "Type")+
 scale_color_manual(values=c("#990000","#CCCC00"))
#
#Residual Sugar
density_red_white_residual_sugar <- ggplot(winequality_dry_5, aes(x = residual_sugar, color = type)) +
 geom_density()+
 geom_vline(aes(xintercept=mean(residual_sugar)),
            color="black", linetype="dashed", size=1)+
 labs (x = "Residual Sugar (g/l)", y = "Density", color = "Type")+
 scale_color_manual(values=c("#990000","#CCCC00"))
#
#Chlorides
density_red_white_chlorides <- ggplot(winequality_dry_5, aes(x = chlorides, color = type)) +
 geom_density()+
 geom_vline(aes(xintercept=mean(chlorides)),
            color="black", linetype="dashed", size=1)+
 labs (x = "Chlorides", y = "Density", color = "Type")+
 scale_color_manual(values=c("#990000","#CCCC00"))
#
#Free Sulphur Dioxide
density_red_white_free_sulfur_dioxide <- ggplot(winequality_dry_5, aes(x = free_sulfur_dioxide, color = type)) +
 geom_density()+
 geom_vline(aes(xintercept=mean(free_sulfur_dioxide)),
            color="black", linetype="dashed", size=1)+
 labs (x = "Free Sulphur Dioxide (mg/l)", y = "Density", color = "Type")+
 scale_color_manual(values=c("#990000","#CCCC00"))
#
#Total Sulphur Dioxide
density_red_white_total_sulfur_dioxide <- ggplot(winequality_dry_5, aes(x = total_sulfur_dioxide, color = type)) +
 geom_density()+
 geom_vline(aes(xintercept=mean(total_sulfur_dioxide)),
            color="black", linetype="dashed", size=1)+
 labs (x = "Total Sulphur Dioxide (mg/l)", y = "Density", color = "Type")+
 scale_color_manual(values=c("#990000","#CCCC00"))
#
#Density
density_red_white_density <- ggplot(winequality_dry_5, aes(x = density, color = type)) +
 geom_density()+
 geom_vline(aes(xintercept=mean(density)),
            color="black", linetype="dashed", size=1)+
 labs (x = "Density (g/ml)", y = "Density", color = "Type")+
 scale_color_manual(values=c("#990000","#CCCC00"))
#
#PH
density_red_white_ph <- ggplot(winequality_dry_5, aes(x = ph, color = type)) +
 geom_density()+
 geom_vline(aes(xintercept=mean(ph)),
            color="black", linetype="dashed", size=1)+
 labs (x = "PH", y = "Density", color = "Type")+
```

```
 scale_color_manual(values=c("#990000","#CCCC00"))
#
#Sulphates
density_red_white_sulphates <- ggplot(winequality_dry_5, aes(x = sulphates, color = type)) +
 geom_density()+
 geom_vline(aes(xintercept=mean(sulphates)),
            color="black", linetype="dashed", size=1)+
 labs (x = "Sulphates (g/l)", y = "Density", color = "Type")+
 scale_color_manual(values=c("#990000","#CCCC00"))
#
#Alcohol
density_red_white_alcohol <- ggplot(winequality_dry_5, aes(x = alcohol, color = type)) +
 geom_density()+
 geom_vline(aes(xintercept=mean(alcohol)),
            color="black", linetype="dashed", size=1)+
 labs (x = "Alcohol (%)", y = "Density", color = "Type")+
 scale_color_manual(values=c("#990000","#CCCC00"))
#
#Total Acidity
density_red_white_total_acidity <- ggplot(winequality_dry_5, aes(x = total_acitity, color = type)) +
 geom_density()+
 geom_vline(aes(xintercept=mean(total_acitity)),
            color="black", linetype="dashed", size=1)+
 labs (x = "Total Acidity (mg/l)", y = "Density", color = "Type")+
 scale_color_manual(values=c("#990000","#CCCC00"))


#After creating each of the density plots we add them all together for easier visualisation
plot_grid(density_red_white_fixed_acidity, density_red_white_citric_acid, density_red_white_volatile_acidity,
          density_red_white_total_acidity, labels = "AUTO", ncol = 2)
#
plot_grid(density_red_white_ph,density_red_white_residual_sugar, density_red_white_alcohol,
          density_red_white_density,labels = "AUTO", ncol = 2)
#
plot_grid(density_red_white_chlorides,density_red_white_free_sulfur_dioxide, density_red_white_total_sulfur_dioxide,
          density_red_white_sulphates,labels = "AUTO", ncol = 2)


#Creating box plot comparing both red and wine
#
#The boc plot contain the Mann-Whitney p-value. I have not added "alternative = "two.sided"" as is not recognised in the
graph.
#However the results are the same as in the full test
#fixed acidity
red_white_fixed_acidity <- ggplot(winequality_dry_5, aes(x=type, y=fixed_acidity, color = type)) +
 geom_jitter(alpha = 0.5) +
 stat_boxplot(fill = NA,color = "Black") +
 labs (x = element_blank(), y = "Fixed Acidity (g/l)", color = "Type") +
 scale_color_manual(values=c("#990000","#CCCC00"))+
 stat_compare_means(method = "wilcox.test", paired = FALSE)
#
#citric acid
red_white_citric_acid <- ggplot(winequality_dry_5, aes(x=type, y=citric_acid, color = type)) +
 geom_jitter(alpha = 0.5) +
 stat_boxplot(fill = NA,color = "Black") +
 labs (x = element_blank(), y = "Citric Acid (g/l)", color = "Type") +
 scale_color_manual(values=c("#990000","#CCCC00"))+
 stat_compare_means(method = "wilcox.test", paired = FALSE)
#
#volatile acidity
red_white_volatile_acidity <- ggplot(winequality_dry_5, aes(x=type, y=volatile_acidity, color = type)) +
 geom_jitter(alpha = 0.5) +
 stat_boxplot(fill = NA,color = "Black") +
```

```
  labs (x = element_blank(), y = "Volatile Acidity (g/l)", color = "Type") +
  scale_color_manual(values=c("#990000","#CCCC00"))+
  stat_compare_means(method = "wilcox.test", paired = FALSE)
#
#total acidity
red_white_total_acidity <- ggplot(winequality_dry_5, aes(x=type, y=total_acitity, color = type)) +
  geom_jitter(alpha = 0.5) +
  stat_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Total Acitidy (g/l)", color = "Type") +
  scale_color_manual(values=c("#990000","#CCCC00"))+
  stat_compare_means(method = "wilcox.test", paired = FALSE)
#
#ph
red_white_ph <- ggplot(winequality_dry_5, aes(x=type, y=ph, color = type)) +
  geom_jitter(alpha = 0.5) +
  stat_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "PH", color = "Type") +
  scale_color_manual(values=c("#990000","#CCCC00"))+
  stat_compare_means(method = "wilcox.test", paired = FALSE)
#
#residual sugar
red_white_residual_sugar <- ggplot(winequality_dry_5, aes(x=type, y=residual_sugar, color = type)) +
  geom_jitter(alpha = 0.5) +
  stat_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Residual Sugar (g/l)", color = "Type") +
  scale_color_manual(values=c("#990000","#CCCC00"))+
  stat_compare_means(method = "wilcox.test", paired = FALSE)
#
#alcohol
red_white_alcohol <- ggplot(winequality_dry_5, aes(x=type, y=alcohol, color = type)) +
  geom_jitter(alpha = 0.5) +
  stat_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Alcohol (%)", color = "Type") +
  scale_color_manual(values=c("#990000","#CCCC00"))+
  stat_compare_means(method = "wilcox.test", paired = FALSE)
#
#density
red_white_density <- ggplot(winequality_dry_5, aes(x=type, y=density, color = type)) +
  geom_jitter(alpha = 0.5) +
  stat_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Density (g/ml)", color = "Type") +
  scale_color_manual(values=c("#990000","#CCCC00"))+
  stat_compare_means(method = "wilcox.test", paired = FALSE)
#
#chlorides
red_white_chlorides <- ggplot(winequality_dry_5, aes(x=type, y=chlorides, color = type)) +
  geom_jitter(alpha = 0.5) +
  stat_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Chlorides", color = "Type") +
  scale_color_manual(values=c("#990000","#CCCC00"))+
  stat_compare_means(method = "wilcox.test", paired = FALSE)
#
#free sulfur dioxide
red_white_free_sulfur_dioxide <- ggplot(winequality_dry_5, aes(x=type, y=free_sulfur_dioxide, color = type)) +
  geom_jitter(alpha = 0.5) +
  stat_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Free Sulphur Dioxide (mg/l)", color = "Type") +
  scale_color_manual(values=c("#990000","#CCCC00"))+
  stat_compare_means(method = "wilcox.test", paired = FALSE)
#
#total sulfur dioxide
red_white_total_sulfur_dioxide <- ggplot(winequality_dry_5, aes(x=type, y=total_sulfur_dioxide, color = type)) +
  geom_jitter(alpha = 0.5) +
```

```
  stat_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Total Sulphur Dioxide (mg/l)", color = "Type") +
  scale_color_manual(values=c("#990000","#CCCC00"))+
  stat_compare_means(method = "wilcox.test", paired = FALSE)
#
#sulphates
red_white_sulphates <- ggplot(winequality_dry_5, aes(x=type, y=sulphates, color = type)) +
  geom_jitter(alpha=0.5) +
  stat_boxplot(fill = NA, color = "Black") +
  labs (x = element_blank(), y = "Sulphates (g/l)", color = "Type") +
  scale_color_manual(values=c("#990000","#CCCC00"))+
  stat_compare_means(method = "wilcox.test", paired = FALSE)



#After creating each of the box plots we add them all together for easier visualisation
#I create a couple of them so that the graphs stay a bit bigger
#
plot_grid(red_white_fixed_acidity, red_white_citric_acid, red_white_volatile_acidity,
          red_white_total_acidity, labels = "AUTO", ncol = 2)
#

plot_grid(red_white_ph,red_white_residual_sugar,red_white_alcohol, red_white_density,
          labels = "AUTO", ncol = 2)
#
plot_grid(red_white_chlorides,red_white_free_sulfur_dioxide,red_white_total_sulfur_dioxide,
          red_white_sulphates, labels = "AUTO", ncol = 2)



#Full Mann-Whitney U test results appearing in index
#
str(redwine_renamed_dry_5)
#
x <- c(1:11,15)
#
for (i in x) {
print(wilcox.test(redwine_renamed_dry_5[,i],whitewine_renamed_dry_5[,i]),alternative = "two.sided", paired = FALSE)
}
#Doing the test for paired samples the same result appears, we reject the null hypothesis of populations being equal
for (i in x) {
  print(wilcox.test(redwine_renamed_dry_5[,i],whitewine_renamed_dry_5[,i]),alternative = "two.sided", paired = TRUE)
}
```

```
#INSIGHT 2 - IS THERE ANY DIFFERENCE BETWEEN DRY RED VINHO VERDE WINES' QUALITY BASED ON THEIR CHEMICAL
PROPERTIES?
#########################################################################################################
###
#########################################################################################################
###
#Creating box plot comparing dry red wines by quality
#
#fixed acidity
quality_red_fixed_acidity <- ggplot(redwine_renamed_dry, aes(x=quality, y=fixed_acidity, color = quality)) +
```

```
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Fixed Acidity (g/l)", color = "Quality") +
  scale_color_manual(values=c("#FF6666","#FF3333","#FF0000","#CC0000","#990000","#660000"))
#
#citric acid
quality_red_citric_acid <- ggplot(redwine_renamed_dry, aes(x=quality, y=citric_acid, color = quality)) +
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Citric Acid (g/l)", color = "Quality") +
  scale_color_manual(values=c("#FF6666","#FF3333","#FF0000","#CC0000","#990000","#660000"))
#
#volatile acidity
quality_red_volatile_acidity <- ggplot(redwine_renamed_dry, aes(x=quality, y=volatile_acidity, color = quality)) +
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Volatile Acidity (g/l)", color = "Quality") +
  scale_color_manual(values=c("#FF6666","#FF3333","#FF0000","#CC0000","#990000","#660000"))
#
#total acidity
quality_red_total_acidity <- ggplot(redwine_renamed_dry, aes(x=quality, y=total_acitity, color = quality)) +
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Total Acidity (g/l)", color = "Quality") +
  scale_color_manual(values=c("#FF6666","#FF3333","#FF0000","#CC0000","#990000","#660000"))
#
#ph
quality_red_ph <- ggplot(redwine_renamed_dry, aes(x=quality, y=ph, color = quality)) +
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "pH", color = "Quality") +
  scale_color_manual(values=c("#FF6666","#FF3333","#FF0000","#CC0000","#990000","#660000"))
#
#residual sugar
quality_red_residual_sugar <- ggplot(redwine_renamed_dry, aes(x=quality, y=residual_sugar, color = quality)) +
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Residual Sugar (g/l)", color = "Quality") +
  scale_color_manual(values=c("#FF6666","#FF3333","#FF0000","#CC0000","#990000","#660000"))
#
#alcohol
quality_red_alcohol <- ggplot(redwine_renamed_dry, aes(x=quality, y=alcohol, color = quality)) +
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Alcohol (%)", color = "Quality") +
  scale_color_manual(values=c("#FF6666","#FF3333","#FF0000","#CC0000","#990000","#660000"))
#
#density
quality_red_density <- ggplot(redwine_renamed_dry, aes(x=quality, y=density, color = quality)) +
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Density (g/ml)", color = "Quality") +
  scale_color_manual(values=c("#FF6666","#FF3333","#FF0000","#CC0000","#990000","#660000"))
#
#chlorides
quality_red_chlorides <- ggplot(redwine_renamed_dry, aes(x=quality, y=chlorides, color = quality)) +
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Chlorides", color = "Quality") +
  scale_color_manual(values=c("#FF6666","#FF3333","#FF0000","#CC0000","#990000","#660000"))
#
#free sulfur dioxide
quality_red_free_sulfur_dioxide <- ggplot(redwine_renamed_dry, aes(x=quality, y=free_sulfur_dioxide, color = quality)) +
```

```
 geom_jitter(alpha = 0.5) +
 geom_boxplot(fill = NA,color = "Black") +
 labs (x = element_blank(), y = "Free Sulfur Dioxide (mg/l)", color = "Quality") +
 scale_color_manual(values=c("#FF6666","#FF3333","#FF0000","#CC0000","#990000","#660000"))
#
#total sulfur dioxide
quality_red_total_sulfur_dioxide <- ggplot(redwine_renamed_dry, aes(x=quality, y=total_sulfur_dioxide, color = quality)) +
 geom_jitter(alpha = 0.5) +
 geom_boxplot(fill = NA,color = "Black") +
 labs (x = element_blank(), y = "Total Surfur Dioxide (mg/l)", color = "Quality") +
 scale_color_manual(values=c("#FF6666","#FF3333","#FF0000","#CC0000","#990000","#660000"))
#
#sulphates
quality_red_sulphates <- ggplot(redwine_renamed_dry, aes(x=quality, y=sulphates, color = quality)) +
 geom_jitter(alpha = 0.5) +
 geom_boxplot(fill = NA,color = "Black") +
 labs (x = element_blank(), y = "Sulphates (g/l)", color = "Quality") +
 scale_color_manual(values=c("#FF6666","#FF3333","#FF0000","#CC0000","#990000","#660000"))



#After creating each of the box plots we add them all together for easier visualisation
#I create a couple of them so that the graphs stay a bit bigger
#
plot_grid(quality_red_fixed_acidity, quality_red_citric_acid, quality_red_volatile_acidity,
        quality_red_total_acidity, labels = "AUTO", ncol = 2)
#

plot_grid(quality_red_ph,quality_red_residual_sugar,quality_red_alcohol, quality_red_density,
        labels = "AUTO", ncol = 2)
#
plot_grid(quality_red_chlorides,quality_red_free_sulfur_dioxide,quality_red_total_sulfur_dioxide,
        quality_red_sulphates, labels = "AUTO", ncol = 2)

#In order to run Kruskal Test:
#Assumptions: independent varaiables, dependant variable is continous, homogeanity of variances
#Checking homogeanity of variances with Fligner-Killeen test
#
#
x <- c(1:11,15)
#
for (i in x) {
 print(fligner.test(redwine_renamed_dry[, i] ~ redwine_renamed_dry$quality, data = redwine_renamed_dry))
}



#Kruskal-Wallis test
#
x <- c(1:11,15)
#
for (i in x) {
 print(kruskal.test(redwine_renamed_dry[, i] ~ redwine_renamed_dry$quality, data = redwine_renamed_dry))
}


#Dunn's test to identity which groups are different
#
x <- c(1:11,15)
#
for (i in x) {
 print(dunnTest(redwine_renamed_dry[, i] ~ redwine_renamed_dry$quality, data = redwine_renamed_dry,
            method = "bonferroni"))
```

```
}



#INSIGHT 2 - IS THERE ANY DIFFERENCE BETWEEN DRY WHITE VINHO VERDE WINES' QUALITY BASED ON THEIR CHEMICAL PROPERTIES?
#####################################################################################################################
###
#####################################################################################################################
###
#Creating box plot comparing dry white wines by quality
#
#fixed acidity
quality_white_fixed_acidity <- ggplot(whitewine_renamed_dry, aes(x=quality, y=fixed_acidity, color = quality)) +
 geom_jitter(alpha = 0.5) +
 geom_boxplot(fill = NA,color = "Black") +
 labs (x = element_blank(), y = "Fixed Acidity (g/l)", color = "Quality") +
 scale_color_manual(values=c("#FFFF99","#FFFF66","#FFFF33","#FFFF00","#CCCC00","#999900","666600"))
#
#citric acid
quality_white_citric_acid <- ggplot(whitewine_renamed_dry, aes(x=quality, y=citric_acid, color = quality)) +
 geom_jitter(alpha = 0.5) +
 geom_boxplot(fill = NA,color = "Black") +
 labs (x = element_blank(), y = "Citric Acid (g/l)", color = "Quality") +
 scale_color_manual(values=c("#FFFF99","#FFFF66","#FFFF33","#FFFF00","#CCCC00","#999900","666600"))
#
#volatile acidity
quality_white_volatile_acidity <- ggplot(whitewine_renamed_dry, aes(x=quality, y=volatile_acidity, color = quality)) +
 geom_jitter(alpha = 0.5) +
 geom_boxplot(fill = NA,color = "Black") +
 labs (x = element_blank(), y = "Volatile Acidity (g/l)", color = "Quality") +
 scale_color_manual(values=c("#FFFF99","#FFFF66","#FFFF33","#FFFF00","#CCCC00","#999900","666600"))
#
#total acidity
quality_white_total_acidity <- ggplot(whitewine_renamed_dry, aes(x=quality, y=total_acitity, color = quality)) +
 geom_jitter(alpha = 0.5) +
 geom_boxplot(fill = NA,color = "Black") +
 labs (x = element_blank(), y = "Total Acidity (g/l)", color = "Quality") +
 scale_color_manual(values=c("#FFFF99","#FFFF66","#FFFF33","#FFFF00","#CCCC00","#999900","666600"))
#
#ph
quality_white_ph <- ggplot(whitewine_renamed_dry, aes(x=quality, y=ph, color = quality)) +
 geom_jitter(alpha = 0.5) +
 geom_boxplot(fill = NA,color = "Black") +
 labs (x = element_blank(), y = "pH", color = "Quality") +
 scale_color_manual(values=c("#FFFF99","#FFFF66","#FFFF33","#FFFF00","#CCCC00","#999900","666600"))
#
#residual sugar
quality_white_residual_sugar <- ggplot(whitewine_renamed_dry, aes(x=quality, y=residual_sugar, color = quality)) +
 geom_jitter(alpha = 0.5) +
 geom_boxplot(fill = NA,color = "Black") +
 labs (x = element_blank(), y = "Residual Sugar (g/l)", color = "Quality") +
 scale_color_manual(values=c("#FFFF99","#FFFF66","#FFFF33","#FFFF00","#CCCC00","#999900","666600"))
#
#alcohol
quality_white_alcohol <- ggplot(whitewine_renamed_dry, aes(x=quality, y=alcohol, color = quality)) +
 geom_jitter(alpha = 0.5) +
 geom_boxplot(fill = NA,color = "Black") +
 labs (x = element_blank(), y = "Alcohol (%)", color = "Quality") +
 scale_color_manual(values=c("#FFFF99","#FFFF66","#FFFF33","#FFFF00","#CCCC00","#999900","666600"))
```

```
#
#density
quality_white_density <- ggplot(whitewine_renamed_dry, aes(x=quality, y=density, color = quality)) +
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Density (g/ml)", color = "Quality") +
  scale_color_manual(values=c("#FFFF99","#FFFF66","#FFFF33","#FFFF00","#CCCC00","#999900","666600"))
#
#chlorides
quality_white_chlorides <- ggplot(whitewine_renamed_dry, aes(x=quality, y=chlorides, color = quality)) +
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Chlorides", color = "Quality") +
  scale_color_manual(values=c("#FFFF99","#FFFF66","#FFFF33","#FFFF00","#CCCC00","#999900","666600"))
#
#free sulfur dioxide
quality_white_free_sulfur_dioxide <- ggplot(whitewine_renamed_dry, aes(x=quality, y=free_sulfur_dioxide, color = quality))
+
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Free Sulfur Dioxide (mg/l)", color = "Quality") +
  scale_color_manual(values=c("#FFFF99","#FFFF66","#FFFF33","#FFFF00","#CCCC00","#999900","666600"))
#
#total sulfur dioxide
quality_white_total_sulfur_dioxide <- ggplot(whitewine_renamed_dry, aes(x=quality, y=total_sulfur_dioxide, color =
quality)) +
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Total Surfur Dioxide (mg/l)", color = "Quality") +
  scale_color_manual(values=c("#FFFF99","#FFFF66","#FFFF33","#FFFF00","#CCCC00","#999900","666600"))
#
#sulphates
quality_white_sulphates <- ggplot(whitewine_renamed_dry, aes(x=quality, y=sulphates, color = quality)) +
  geom_jitter(alpha = 0.5) +
  geom_boxplot(fill = NA,color = "Black") +
  labs (x = element_blank(), y = "Sulphates (g/l)", color = "Quality") +
  scale_color_manual(values=c("#FFFF99","#FFFF66","#FFFF33","#FFFF00","#CCCC00","#999900","666600"))



#After creating each of the box plots we add them all together for easier visualisation
#I create a couple of them so that the graphs stay a bit bigger
#
plot_grid(quality_white_fixed_acidity, quality_white_citric_acid, quality_white_volatile_acidity,
          quality_white_total_acidity, labels = "AUTO", ncol = 2)
#

plot_grid(quality_white_ph,quality_white_residual_sugar,quality_white_alcohol, quality_white_density,
          labels = "AUTO", ncol = 2)
#
plot_grid(quality_white_chlorides,quality_white_free_sulfur_dioxide,quality_white_total_sulfur_dioxide,
          quality_white_sulphates, labels = "AUTO", ncol = 2)


str(whitewine_renamed)



#In order to run Kruskal Test:
#Assumptions: independent varaiables, dependant variable is continous, homogeanity of variances
#Checking homogeanity of variances with Fligner-Killeen test
#
#
```

```
x <- c(1:11,15)
#
for (i in x) {
  print(fligner.test(whitewine_renamed_dry[, i] ~ whitewine_renamed_dry$quality, data = whitewine_renamed_dry))
}

#Kruskal-Wallis test
#
x <- c(1:11,15)
#
for (i in x) {
  print(kruskal.test(whitewine_renamed_dry[, i] ~ whitewine_renamed_dry$quality, data = whitewine_renamed_dry))
}


#Dunn's test to identity which groups are different
#
for (i in x) {
  print(dunnTest(whitewine_renamed_dry[, i] ~ whitewine_renamed_dry$quality, data = whitewine_renamed_dry,
                 method = "bonferroni"))
}




#INSIGHT 3 - IS IT POSSIBLE THE CREATION OF A PREDICTIVE SIMPLE LINEAR MODEL FOR SOME OF
#THE CHEMICAL PROPERTIES OF DRY RED VINHO WINE?
#############################################################################################################
###
#############################################################################################################
###
#Scatter Plots
scat_redwine_renamed_1 <- ggplot(redwine_renamed_dry, aes(x=fixed_acidity, y=total_acitity)) +
  geom_point(color = "#990000", alpha = 0.5)+
  geom_smooth(method = lm, color = "black", fill="black")+
  labs (x = "Fixed Acidity (g/l)", y = "Total Acidity (g/l)")

scat_redwine_renamed_2 <- ggplot(redwine_renamed_dry, aes(x=free_sulfur_dioxide, y=total_sulfur_dioxide)) +
  geom_point(color = "#990000", alpha = 0.5)+
  geom_smooth(method = lm, color = "black", fill="black")+
  labs (x = "Free Sulfur Dioxide (mg/l)", y = "Total Sulfur Dioxide (mg/l)")

scat_redwine_renamed_3 <- ggplot(redwine_renamed_dry, aes(x=total_acitity, y=ph)) +
  geom_point(color = "#990000", alpha = 0.5)+
  geom_smooth(method = lm, color = "black", fill="black")+
  labs (x = "Total Acidity (g/l)", y = "pH")

scat_redwine_renamed_4 <- ggplot(redwine_renamed_dry, aes(x=fixed_acidity, y=ph)) +
  geom_point(color = "#990000", alpha = 0.5)+
  geom_smooth(method = lm, color = "black", fill="black")+
  labs (x = "Fixed Acidity (g/l)", y = "pH")


plot_grid(scat_redwine_renamed_1,scat_redwine_renamed_2,scat_redwine_renamed_3,scat_redwine_renamed_4)



#Checking the range for the selected variables
range(redwine_renamed_dry$fixed_acidity)
range(redwine_renamed_dry$total_acitity)
range(redwine_renamed_dry$ph)
range(redwine_renamed_dry$free_sulfur_dioxide)
```

```
range(redwine_renamed_dry$total_sulfur_dioxide)
#Creating some linear models
model_1 <- lm(fixed_acidity ~ total_acitity, data = redwine_renamed_dry)
model_2 <- lm(free_sulfur_dioxide ~ total_sulfur_dioxide, data = redwine_renamed_dry)
model_3 <- lm(total_acitity ~ ph, data = redwine_renamed_dry)
model_4 <- lm(fixed_acidity ~ ph, data = redwine_renamed_dry)
#Having access to the beta and the alpha to form the equations
print(model_1)
print(model_2)
print(model_3)
print(model_4)
#Statistical summary of the models
summary(model_1)
summary(model_2)
summary(model_3)
summary(model_4)
#Predictions with the models
predict(model_1, data.frame(total_acitity = 7), level = 0.99, interval ="prediction")
predict(model_2, data.frame(total_sulfur_dioxide = 100), level = 0.99, interval ="prediction")
predict(model_3, data.frame(ph = 2.80), level = 0.99, interval ="prediction")
predict(model_4, data.frame(ph = 2.80), level = 0.99, interval ="prediction")




#INSIGHT 4 - IS IT POSSIBLE TO CLASSIFY DRY VINHO VERDE WINES BY TYPE (RED/WHITE)?
##############################################################################################################
###
##############################################################################################################
###
#Following different manuals:
#https://www.kaggle.com/vshelunts/wine-quality-decision-tree-and-random-forest
#https://dzone.com/articles/a-comprehensive-guide-to-random-forest-in-r
#https://machinelearningmastery.com/machine-learning-in-r-step-by-step/

#Model to predict if a wine is red
#Create data partition
validation_index_dry_5 <- createDataPartition(winequality_dry_5$type, p=0.80, list=FALSE)
# 80% of data for training purposes
training_dry_5 <- winequality_dry_5[validation_index_dry_5,]
# 20% of the data for validation
validation_dry_5 <- winequality_dry_5[-validation_index_dry_5,]
#checking number of rows for each partition
nrow(winequality_dry_5)
nrow(training_dry_5)
nrow(validation_dry_5)

#Creating model
model_classification_dry_5_type<- randomForest(type~., training_dry_5, ntree=50)
#Checking which variables contribute more to the model
varImpPlot(model_classification_dry_5_type, bg = "#990000", main="Contribution of variables to the Model")


#Testing the model - check acuracy in predicting the observations appearing in the validation subset
prediction_model_classification_dry_5_type <- predict(model_classification_dry_5_type, validation_dry_5)
#Confusion Matrix to check the validity of the model
confusionMatrix(prediction_model_classification_dry_5_type, validation_dry_5$type)


importance(model_classification_dry_5_type)
```

```
#INSIGHT 4 - IS IT POSSIBLE TO CLASSIFY DRY VINHO VERDE WINES BY QUALITY?
#########################################################################################################################
###
#########################################################################################################################
###
#Model to classify red wine by quality
#Create data partition
validation_index_dry_red <- createDataPartition(redwine_renamed_dry$quality, p=0.80, list=FALSE)
# 80% of data for training purposes
training_dry_red <- redwine_renamed_dry[validation_index_dry_red,]
# 20% of the data for validation
validation_dry_red <- redwine_renamed_dry[-validation_index_dry_red,]

#Checking existing levels for quality variable
levels(training_dry_red$quality)
#Drop unused levels from traning subset
training_dry_red$quality <- drop.levels(training_dry_red$quality)
#Confirm the levels were droped
levels(validation_dry_red$quality)
#Drop unused levels from validation subset
validation_dry_red$quality <- drop.levels(validation_dry_red$quality)
#Confirm the levels were droped
levels(validation_dry_red$quality)

#checking number of rows for each partition
nrow(redwine_renamed_dry)
nrow(training_dry_red)
nrow(validation_dry_red)

#Creating model
model_classification_dry_red_quality<- randomForest(quality~., training_dry_red, ntree=50)
#Checking which variables contribute more to the model
varImpPlot(model_classification_dry_red_quality, bg = "#990000", main="Contribution of variables to the Model")

#Testing the model - check acuracy in predicting the observations appearing in the validation subset
prediction_model_classification_dry_red_quality <- predict(model_classification_dry_red_quality, validation_dry_red)
#Confusion Matrix to check the validity of the model
confusionMatrix(prediction_model_classification_dry_red_quality, validation_dry_red$quality)




#Model to classify white wine by quality
#Creating a new subset from white dry wines where quality is 1 to 6
whitewine_renamed_dry_for_model <- subset(whitewine_renamed_dry, whitewine_renamed_dry$quality != 7 )


#Visualising the observations for the new variable quality
whitewine_renamed_dry_for_model$quality

#Create data partition
validation_index_dry_white <- createDataPartition(whitewine_renamed_dry_for_model$quality, p=0.80, list=FALSE)
# 20% of the data for validation
validation_dry_white <- whitewine_renamed_dry_for_model[-validation_index_dry_white,]
# 80% of data for training purposes
training_dry_white <- whitewine_renamed_dry_for_model[validation_index_dry_white,]
```

```
#checking number of rows for each partition
nrow(whitewine_renamed_dry)
nrow(training_dry_white)
nrow(validation_dry_white)

#Checking existing levels for quality variable
levels(training_dry_white$quality)
#Drop unused levels from traning subset
training_dry_white$quality <- drop.levels(training_dry_white$quality)
#Confirm the levels were droped
levels(training_dry_white$quality)
#Confirm the levels were droped
levels(validation_dry_white$quality)
#Drop unused levels from validation subset
validation_dry_white$quality <- drop.levels(validation_dry_white$quality)
#Confirm the levels were droped
levels(validation_dry_white$quality)


#Creating model
model_classification_dry_white_quality<- randomForest(quality~., training_dry_white, ntree=50)
#Checking which variables contribute more to the model
varImpPlot(model_classification_dry_white_quality,bg = "#CCCC00", main="Contribution of variables to the Model")

#Testing the model - check acuracy in predicting the observations appearing in the validation subset
prediction_model_classification_dry_white_quality <- predict(model_classification_dry_white_quality,
validation_dry_white)
#Confusion Matrix to check the validity of the model
confusionMatrix(prediction_model_classification_dry_white_quality, validation_dry_white$quality)
```