White wine analysis

Alice Lo Gioco

2022-02-27

1 Introduction

In this report we analyze and we use the Wine Quality dataset, made available by the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Wine+Quality). In this link, two datasets are included, related to red and white Vinho Verde wine samples.

The Vinho verde wine is exclusively produced in the demarcated region of Vinho Verde in northwestern Portugal, it is only produced from the indigenous grape varieties of the region, preserving its typicity of aromas and flavors as unique in the world of wine. Already at the time of the monarchy, particularly during the reign of King Charles in 1908, the quality and genuineness of the Vinho Verde wine region was being officially recognized by the demarcation of region as a geographical area of production. The origin of the Vinho Verde name refers to the natural characteristics of the region, which produce dense green foliage, but which also contribute to the wine's profile with freshness and lightness. It is this youthfulness that the wine is named after, in comparison to other more complex and weighty wines.

Furthemore, the wine analysis is today a practice for oenologists, producers and wineries, indispensable during all the various stages of wine production, in order to:

- determine the expected quality grade
- verify the absence of alterations in the product
- confirm the desired organoleptic characteristics
- identify any problems that may arise in the various stages of preparation and storage.

In this report we take into consideration only the White Wine dataset and we build different classification models to predict if a white wine is good or not.

1.1 Preparation of the data to analyze

The dataset is saved in a csv file where the column are separated by ';'. With the code below we upload the dataset in R in order to use it for our analysis.

```
 # Note: this process could take a couple of minutes
dl <- tempfile()
download.file("https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white
WhiteWine <- read.csv(dl,sep = ';')
rm(dl)
```

We load the following libraries:

```r
if(!require(knitr)) install.packages("knitr", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: knitr
```

```r
if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v readr   2.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

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

```
## Loading required package: caret
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
if(!require(gtools)) install.packages("gtools", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: gtools
```

```r
if(!require(RColorBrewer)) install.packages("RColorBrewer", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: RColorBrewer
```

```r
if(!require(corrplot)) install.packages("corrplot", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: corrplot
```

```
## corrplot 0.92 loaded
```

```r
if(!require(gridExtra)) install.packages("gridExtra", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: gridExtra
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
if(!require(gam)) install.packages("gam", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: gam
```

```
## Loading required package: splines
```

```
## Loading required package: foreach
```

```
##
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
## Loaded gam 1.20
```

```r
library(knitr)
library(dplyr)
library(tidyverse)
library(ggplot2)
library(caret)
library(gtools)
library(RColorBrewer)
library(corrplot)
library (gridExtra)
library(gam)
```

1.2 Description of the dataset

First we start by exploring the dataset for understand the structure, the distribution of all the variables and the relationship of the predictors.

We start by seeing the structure and the first 6 rows in the dataset edx:

```
str(WhiteWine)
```

```
## 'data.frame':    4898 obs. of  12 variables:
##  $ fixed.acidity       : num  7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
##  $ volatile.acidity    : num  0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
##  $ citric.acid         : num  0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
##  $ residual.sugar      : num  20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
##  $ chlorides           : num  0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
##  $ free.sulfur.dioxide : num  45 14 30 47 47 30 30 45 14 28 ...
##  $ total.sulfur.dioxide: num  170 132 97 186 186 97 136 170 132 129 ...
##  $ density             : num  1.001 0.994 0.995 0.996 0.996 ...
##  $ pH                  : num  3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...
##  $ sulphates           : num  0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
##  $ alcohol             : num  8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
##  $ quality             : int  6 6 6 6 6 6 6 6 6 6 ...
```

```
head(WhiteWine)
```

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

The dataset is in tidy format, each row has one observation and the column names are the features. There are 12 columns:

- fixed.acidity (numeric): Amount of Tartaric Acid in wine, measured in g/dm3, they are non-volatile acids that do not evaporate easily;

- volatile.acidity (numeric): Amount of Acetic Acid in wine, measured in g/dm3; which leading to an unpleasant vinegar taste;
- citric.acid (numeric): Amount of citric acid in wine in g/dm3. Contributes to crispness of wine, acts as a preservative to increase acidity; Small quantities add freshness and flavor to wines;
- residual.sugar (numeric): amount of sugar left in wine after fermentation. Measured in in g/dm3 (wines with > 45g/ltrs are sweet);
- chlorides (numeric): the amount of Sodium Choloride (salt) in wine. Measured in g/dm3;
- free.sulfur.dioxide (numeric): Amount of SO2 in free form. Measured in mg/dm3, it prevents microbial growth and the oxidation of wine;
- total.sulfur.dioxide (numeric): total Amount of SO2. Too much SO2 can lead to a pungent smell. SO2 acts as antioxidant and antimicrobial agent;
- density (numeric): Density of Wine in g/dm3;
- pH (numeric): the level of acidity of the Wine on a scale of 0-14 . 0 means highly Acidic, while 14 means highly basic;
- sulphates (numeric): amount of Potassium Sulphate in wine, measured in g/dm3. Contributes to the formation of SO2 and acts as an antimicobial and antioxidant;
- alcohol (numeric) : the amount of alcohol in wine (in terms of % volume);
- quality (numeric): wine Quality graded on a scale of 1 - 10 (Higher is better)

We can group the variables in these categories:

- Acid: fixed.acidity, volatile.acidity, citric.acid
- Salt: chlorides
- Sugar: residual.sugar
- Physical: density
- Chemicals: free.sulfur.dioxide, total.sulfur.dioxide, pH, sulphates
- Alcohol: alcohol

The quality variable is not a wine's feature, but it is a rating given to the wine.

In the description of the dataset is declared that there aren't missing Attribute Values, that is confirmed below:

```
which(is.na(WhiteWine))
```

```
## integer(0)
```

Let's look at the summary and dimensions of the dataset:

```
summary(WhiteWine)
```

```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.300   1st Qu.:0.2100   1st Qu.:0.2700   1st Qu.: 1.700
##  Median : 6.800   Median :0.2600   Median :0.3200   Median : 5.200
##  Mean   : 6.855   Mean   :0.2782   Mean   :0.3342   Mean   : 6.391
##  3rd Qu.: 7.300   3rd Qu.:0.3200   3rd Qu.:0.3900   3rd Qu.: 9.900
##  Max.   :14.200   Max.   :1.1000   Max.   :1.6600   Max.   :65.800
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.00900   Min.   :  2.00      Min.   :  9.0        Min.   :0.9871
##  1st Qu.:0.03600   1st Qu.: 23.00      1st Qu.:108.0        1st Qu.:0.9917
##  Median :0.04300   Median : 34.00      Median :134.0        Median :0.9937
##  Mean   :0.04577   Mean   : 35.31      Mean   :138.4        Mean   :0.9940
##  3rd Qu.:0.05000   3rd Qu.: 46.00      3rd Qu.:167.0        3rd Qu.:0.9961
##  Max.   :0.34600   Max.   :289.00      Max.   :440.0        Max.   :1.0390
##        pH           sulphates          alcohol         quality
##  Min.   :2.720   Min.   :0.2200   Min.   : 8.00   Min.   :3.000
##  1st Qu.:3.090   1st Qu.:0.4100   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.180   Median :0.4700   Median :10.40   Median :6.000
##  Mean   :3.188   Mean   :0.4898   Mean   :10.51   Mean   :5.878
##  3rd Qu.:3.280   3rd Qu.:0.5500   3rd Qu.:11.40   3rd Qu.:6.000
##  Max.   :3.820   Max.   :1.0800   Max.   :14.20   Max.   :9.000
```

```
dim(WhiteWine)
```

```
## [1] 4898   12
```

Below some considerations:

- in the dataset there are 4898 white wine's observation;
- the alcohol value varies from 8 to 14.20;
- the pH value varies from 2.720 to 3.820;
- the sulfur dioxide has a large range of value, both for free and total one;
- the residual sugar value varies from 0.6 to 65.8;
- the quality values are from 3 to 9 and the mean is 5.878

2 Analysis and Methods

2.1 Data exploration & visualization

In this paragraph we analyze the distribution of all the variables in the dataset.

We start to analyze the quality information. As seen in the summary, that we reported below:

```
summary(WhiteWine$quality)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.878   6.000   9.000
```

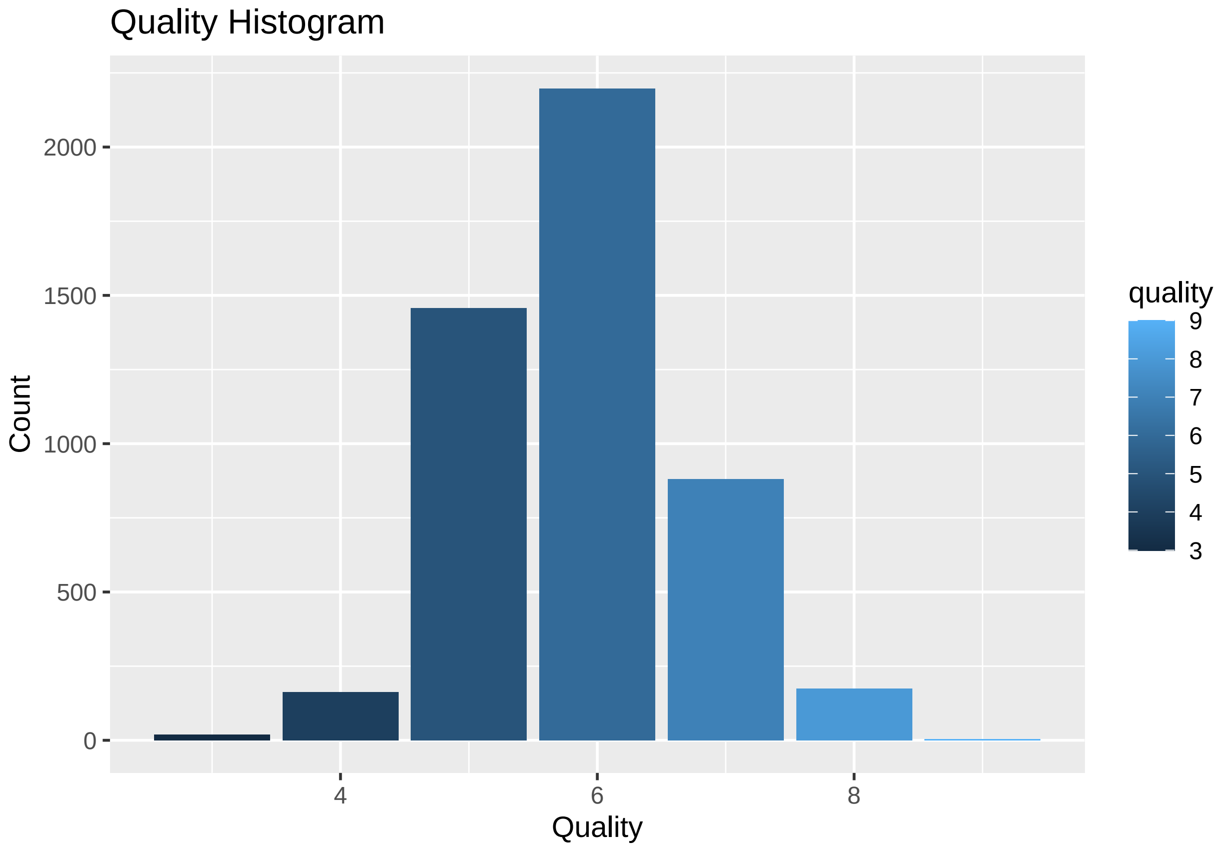
the quality values are from 3 to 9 and the mean is 5.878. Now we analyze the distribution of the wine quality, in a table:

```
table(WhiteWine$quality)
```

```
##
##    3    4    5    6    7    8    9
##   20  163 1457 2198  880  175    5
```

and graphically in a histogram:

```
WhiteWine%>%ggplot(aes(quality, fill=quality, group=quality)) +
  geom_bar() + ggtitle("Quality Histogram")+
  labs(x="Quality" , y="Count")
```

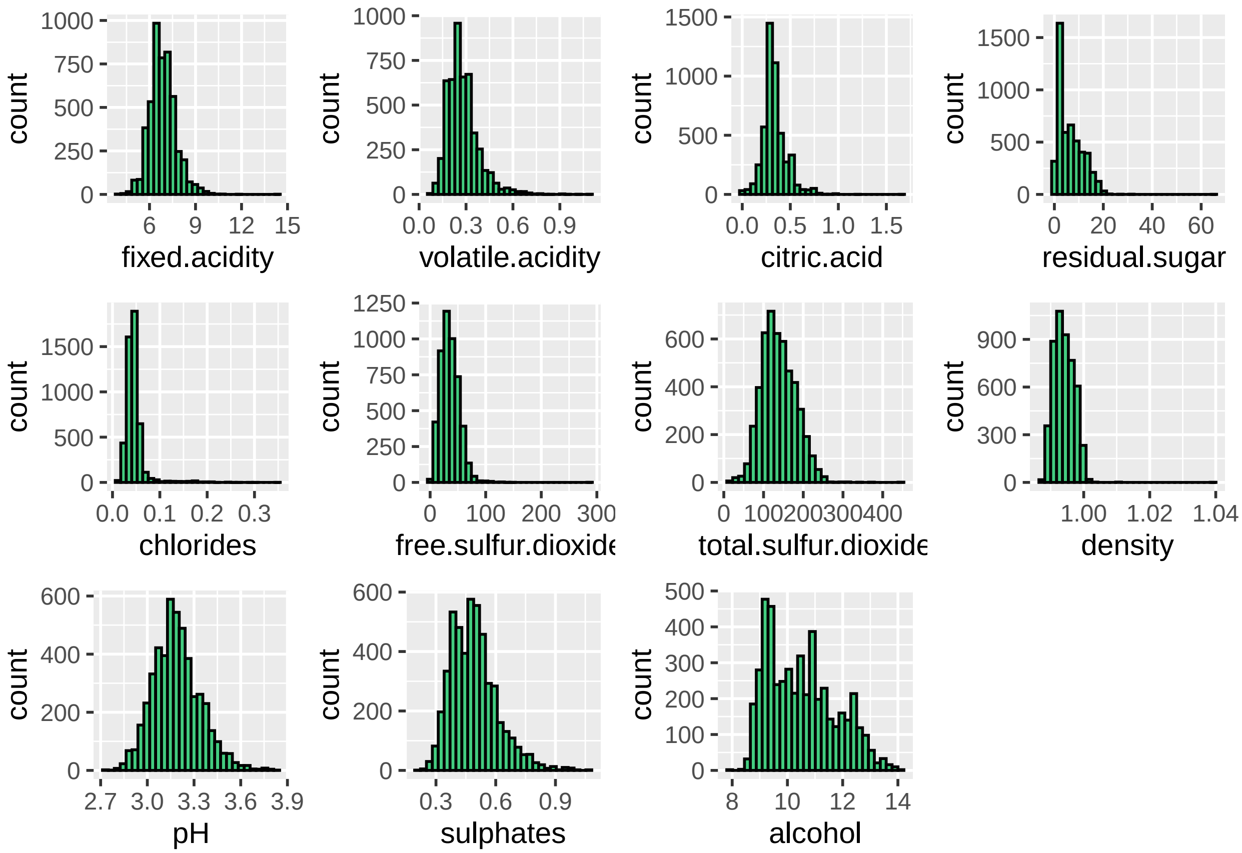


The quality's level 6 has the most observations (equal to 2198), while level 3 has the least (only 30 observations).

Now we explore the distribution of the others variables:

```
fa<-WhiteWine%>%ggplot(aes(fixed.acidity)) +
  geom_histogram(col="black", fill="seagreen3",bins=30)
va<-WhiteWine%>%ggplot(aes(volatile.acidity)) +
  geom_histogram(col="black", fill="seagreen3",bins=30)
ca<-WhiteWine%>%ggplot(aes(citric.acid)) +
  geom_histogram(col="black", fill="seagreen3",bins=30)
rs<-WhiteWine%>%ggplot(aes(residual.sugar)) +
  geom_histogram(col="black", fill="seagreen3",bins=30)
cl<-WhiteWine%>%ggplot(aes(chlorides)) +
  geom_histogram(col="black", fill="seagreen3",bins=30)
fs<-WhiteWine%>%ggplot(aes(free.sulfur.dioxide)) +
  geom_histogram(col="black", fill="seagreen3",bins=30)
ts<-WhiteWine%>%ggplot(aes(total.sulfur.dioxide)) +
  geom_histogram(col="black", fill="seagreen3",bins=30)
de<-WhiteWine%>%ggplot(aes(density)) +
  geom_histogram(col="black", fill="seagreen3",bins=30)
ph<-WhiteWine%>%ggplot(aes(pH)) +
  geom_histogram(col="black", fill="seagreen3",bins=30)
su<-WhiteWine%>%ggplot(aes(sulphates)) +
  geom_histogram(col="black", fill="seagreen3",bins=30)
al<-WhiteWine%>%ggplot(aes(alcohol)) +
  geom_histogram(col="black", fill="seagreen3",bins=30)
```

```r
grid.arrange(fa,va,ca,rs,cl,fs,ts,de,ph,su,al,ncol=4)
```



Below some observations from the plot:

- almost all the features seem to display a normal distribution;
- the residual sugar doesn't exceed to 20, so most of the wines aren't so sweet;
- the free sulfur dioxide seems to spread between 0 to 120 with peak exhibiting around 50;
- the total sulfur dioxide seems to have a spread between 0 and 250 with a peak around 150;
- the alcohol varies from 8 to 14 with major peaks around 10; the mean is equals to 5.878;
- the density's range values is small, the values are from 0.9871 to 1.0390

2.2 Correlation Matrix

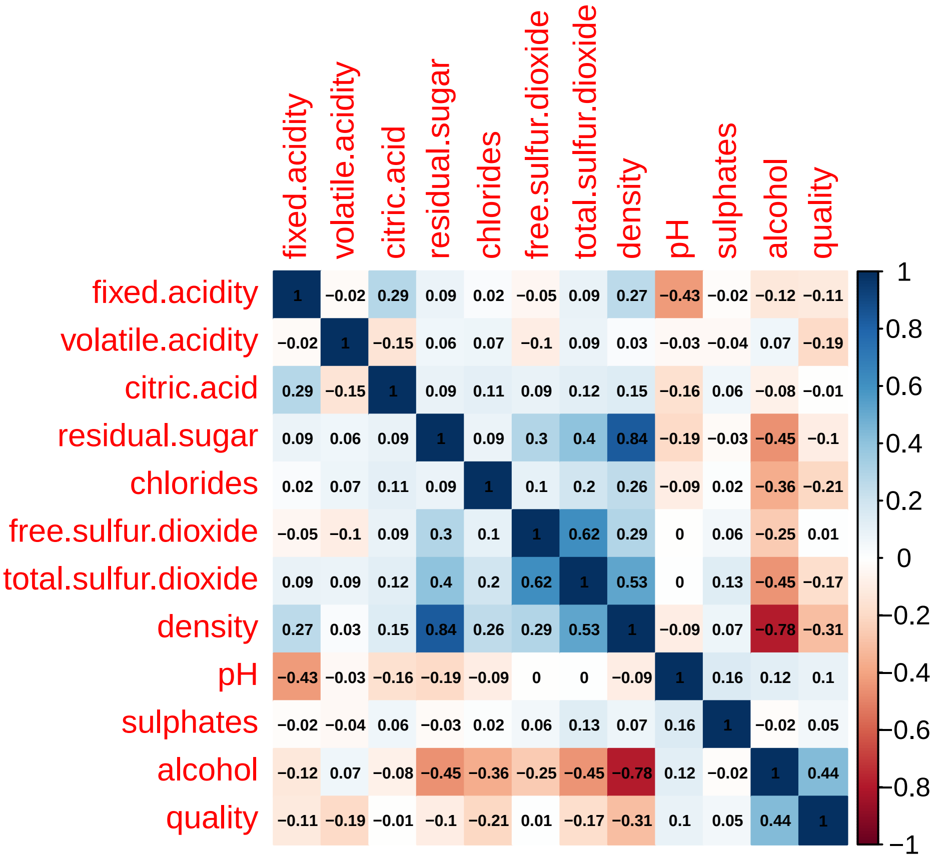
We want to understand the relationship between the variables.

We use the Pearson correlation coefficient, which is a measure of the linear association between two variables. It has a value between -1 and 1 where:

- -1 indicates a perfectly negative linear correlation between two variables
- 0 indicates no linear correlation between two variables
- 1 indicates a perfectly positive linear correlation between two variables

The further away the correlation coefficient is from zero, the stronger the relationship between the two variables. To better visualize the correlation, we use the correlogram, provided by the library corrplot. This is a graph of correlation matrix where the correlation coefficients are colored according to the value and it is useful to highlight the most correlated variables in a data table.

```
corrplot(cor(WhiteWine),method="color", addCoef.col = 'black', number.cex=0.5)
```



Each cell in the table shows the correlation between two specific variables.

The correlation coefficients along the diagonal of the table are all equals to 1 because each variable is perfectly correlated with itself, so these cells aren't useful for interpretation.

We can see that the following features are positively correlated:

- the correlation between "density" and "residual sugar" is equals to +0.84, so more "residual sugar" in the wine is strongly related to density, sweeter wines have a higher density;
- the correlation between "Total sulfur dioxide" with "Free sulfur dioxide" is equals to +0.62, so the "Total sulfur dioxide" is strongly related to "Free sulfur dioxide";
- the correlation between "alcohol" and "quality" is equals to +0.44, so more alcohol in the wine is strongly related to quality, more alcoholic wines have a higher quality;
- the correlation between "citric acid" with "fixed acidity" is equals to +0.29;
- the correlation between "fixed acidity" with "density" is equals to +0.27

And the following features are negatively correlated:

- the correlation between "density" and "alcohol" is equals to -0.78, which indicates that more alcohol in the wine is associated with less the density, more alcoholic wines have a lower density;
- the correlation between "pH" and "fixed acidity" is equals to -0.43, which indicates that more pH in the wine is associated with less the fixed acidity;
- the correlation between "residual sugar" and "alcohol" is equals to -0.45, which indicates that more "residual sugar" in the wine is associated with less alcohol;

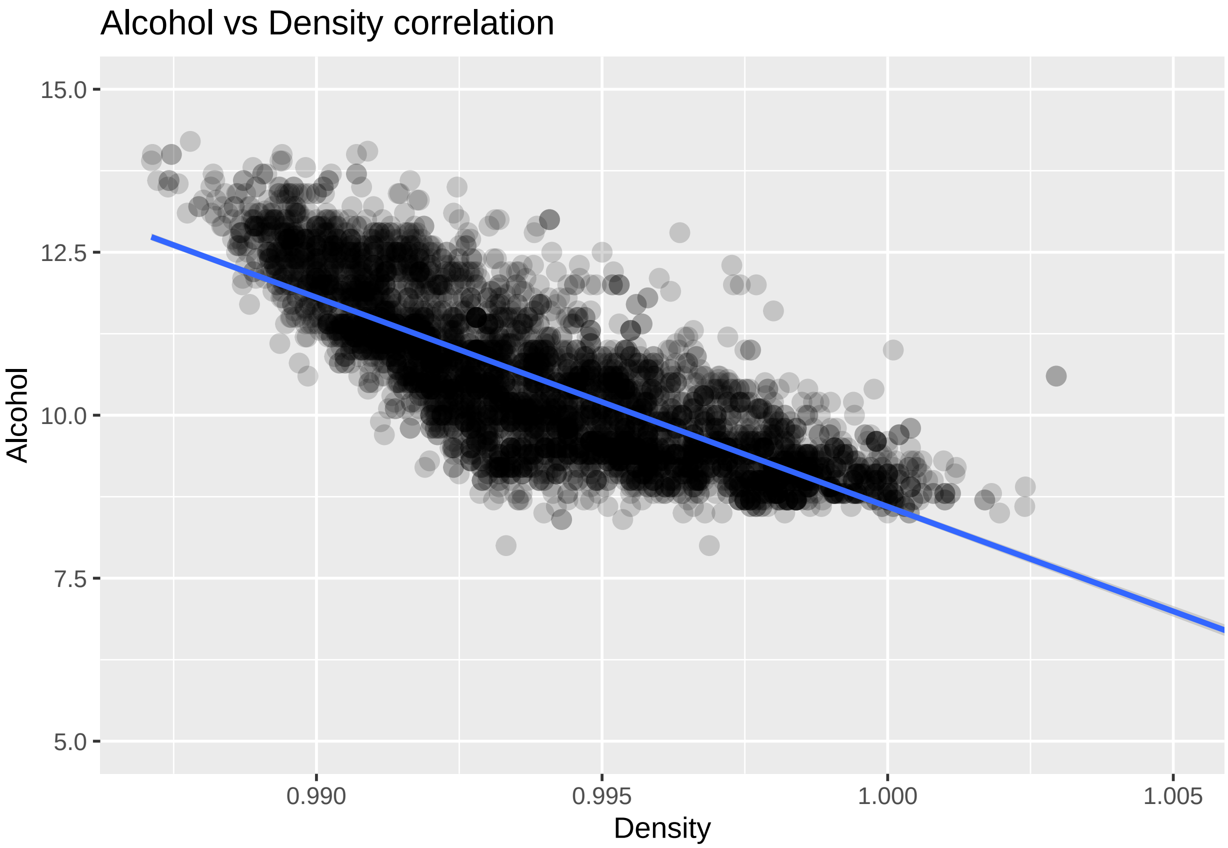
- the correlation between "volatile acidity" and "quality" is equals to -0.19, which indicates that more "volatile acidity" in the wine is associated with less quality; the volatile acidity can improve the vinegar taste, so the wine become unpleasant

2.2.1 Alcohol vs Density

As saw before, the Alcohol and the Density are negatively correlated, we plot below the correlation:

```
WhiteWine%>% ggplot(
       aes(x = density, y = alcohol)) +
  geom_point(alpha = 1/6, position = position_jitter(h = 0), size = 3) +
  geom_smooth(method = 'lm') +
  coord_cartesian(xlim=c(min(WhiteWine$density),1.005), ylim=c(5,15)) +
  xlab('Density') +
  ylab('Alcohol') +
  ggtitle('Alcohol vs Density correlation')
```

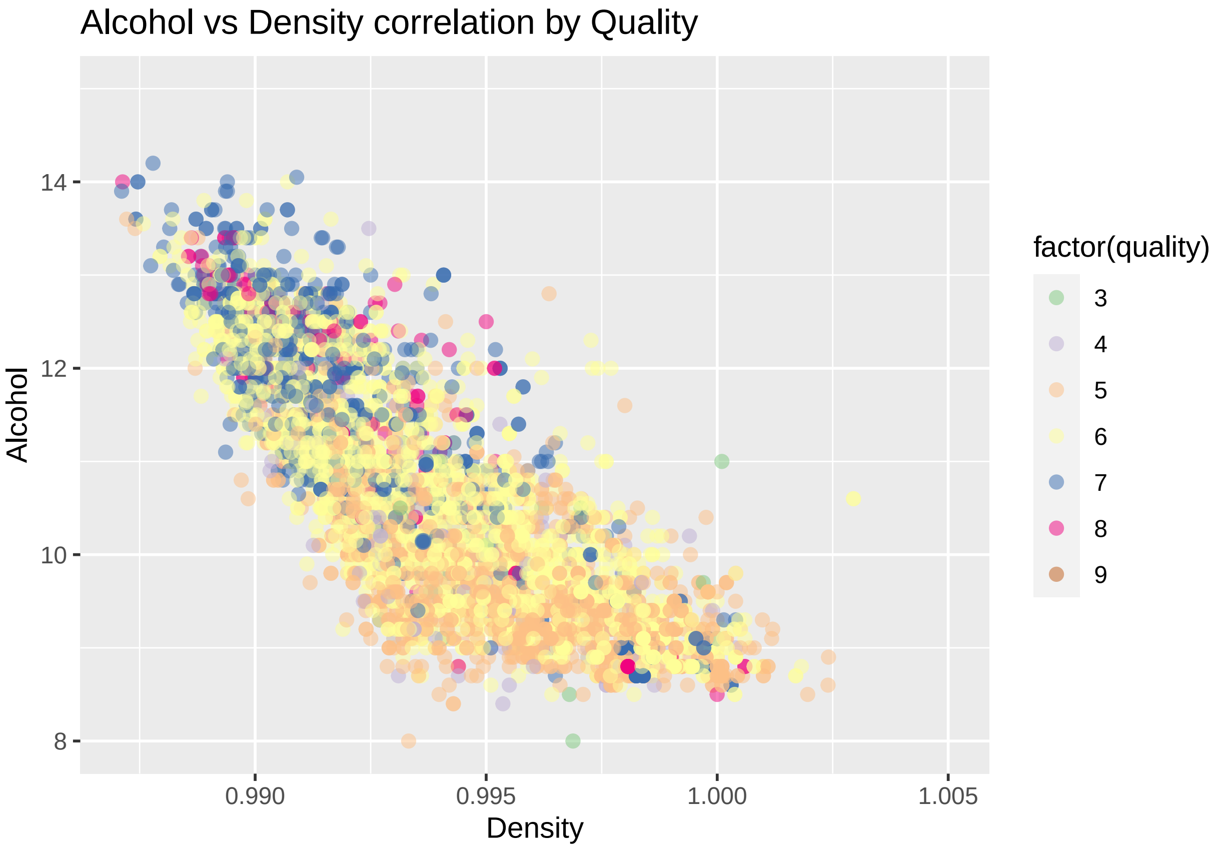
```
## 'geom_smooth()' using formula 'y ~ x'
```



We can see that when density level increase, alcohol decrease.

Below, we investigate the correlation of Alcohol vs Density related to the quality:

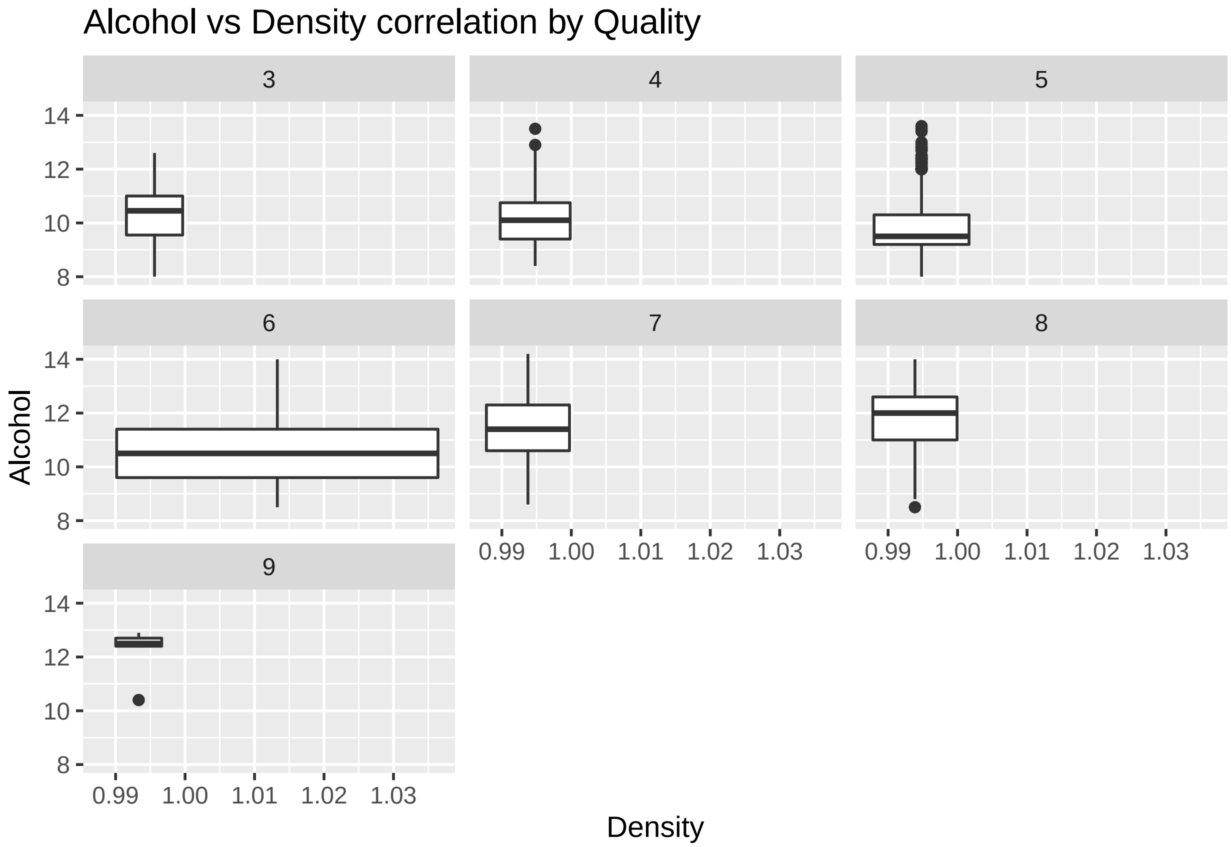
```
WhiteWine%>%  ggplot( aes(x = density, y = alcohol, color = factor(quality))) +
  geom_point(alpha = 1/2, position = position_jitter(h = 0), size = 2) +
  coord_cartesian(xlim=c(min(WhiteWine$density),1.005), ylim=c(8,15)) +
  scale_color_brewer(type='qual') +
  xlab('Density') +
  ylab('Alcohol') +
  ggtitle('Alcohol vs Density correlation by Quality')
```



The quality 7-8 levels wines are concentrated on the left top corner in the scatter plot which represent low density and high alcohol; which mean low density and high alcohol give better quality.

Below we take a look to the boxplot of the correlation between the Alcohol and the Density related by Quality:

```
WhiteWine%>%ggplot(aes(x = density, y = alcohol, group = quality) )+
  facet_wrap( ~ quality) +
  geom_boxplot() +
  xlab('Density') +
  ylab('Alcohol') +
  ggtitle('Alcohol vs Density correlation by Quality')
```



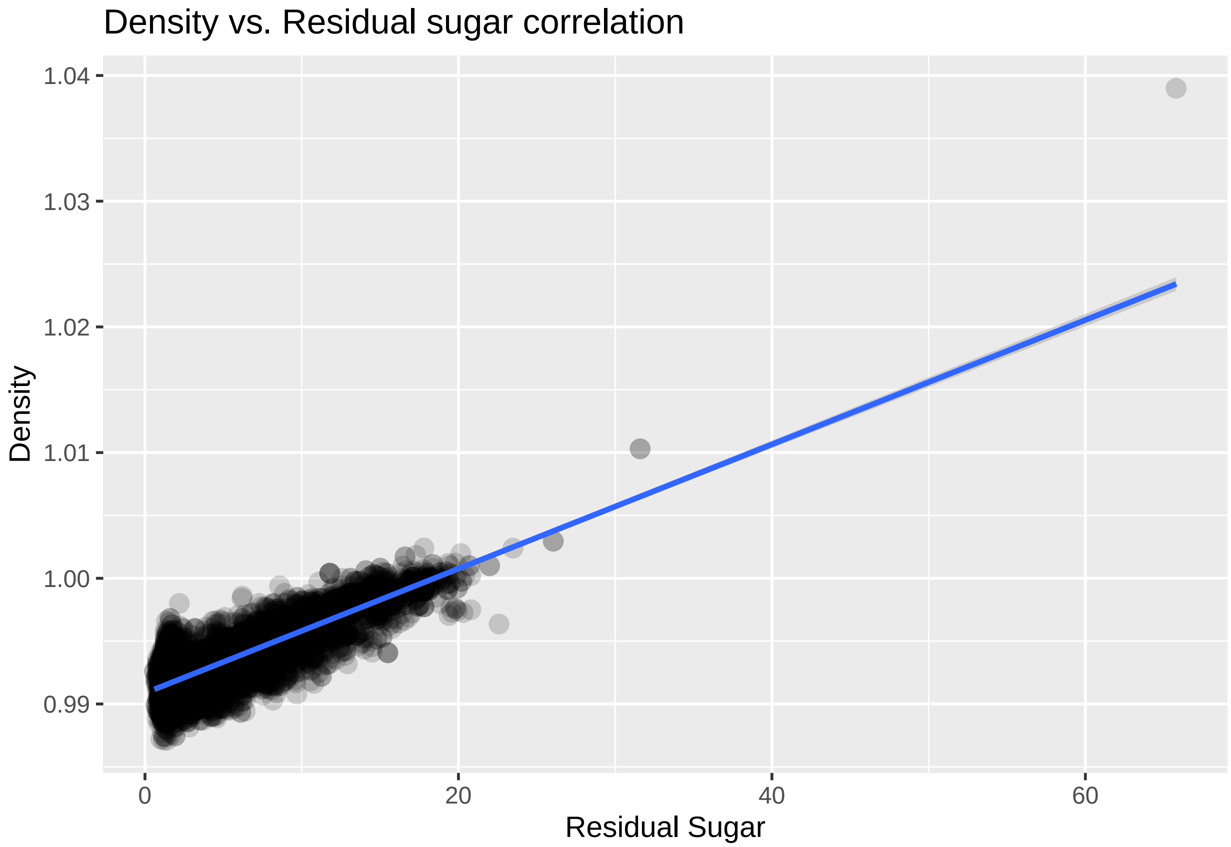
The wine with high alcohol percentage has quality level 7, wine with less alcohol percentage is quality level 5. Wine with quality levels 6 and 8 have various combinations of alcohol and density.

2.2.2 Density vs Residual Sugar

As saw before, the Residual sugar and the Density are positively correlated, we plot below the correlation:

```
WhiteWine%>% ggplot(
  aes(x = residual.sugar, y = density)) +
  geom_point(alpha = 1/6, position = position_jitter(h = 0), size = 3) +
  geom_smooth(method = 'lm') +
  coord_cartesian(xlim=c(min(WhiteWine$residual.sugar),max(WhiteWine$residual.sugar)),
                  ylim=c(min(WhiteWine$density),max(WhiteWine$density))) +
  xlab('Residual Sugar') +
  ylab('Density') +
  ggtitle('Density vs. Residual sugar correlation')
```

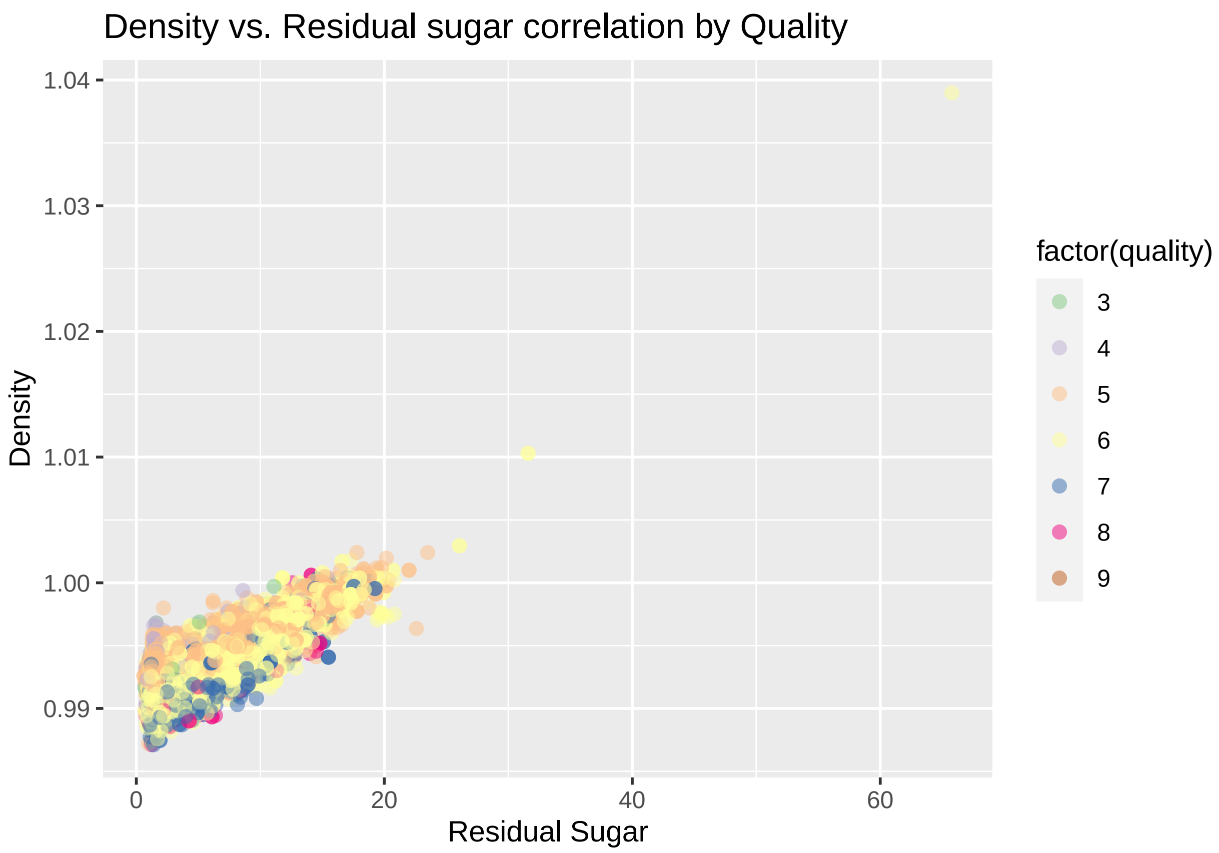
```
## 'geom_smooth()' using formula 'y ~ x'
```



We can see that when Residual sugar level increase, also density increase.

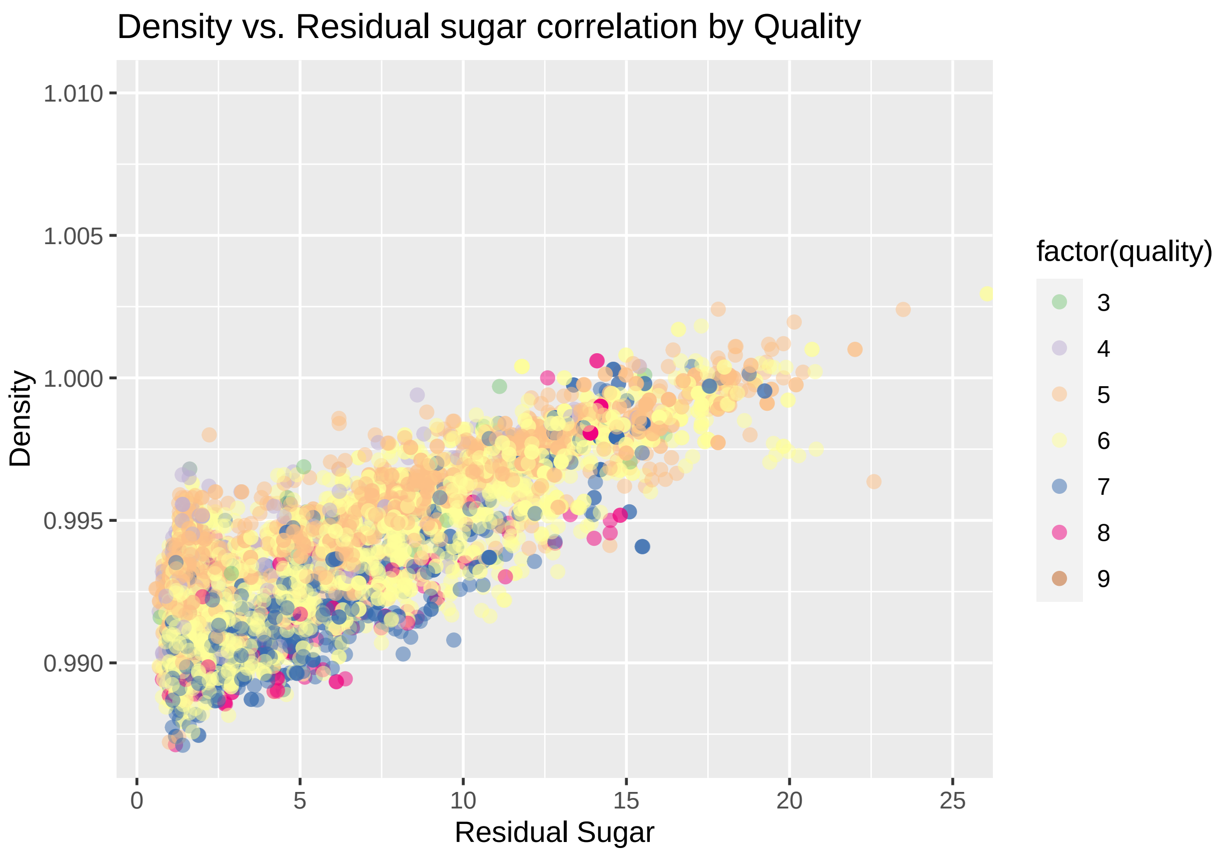
Below, we investigate the correlation between the density and Residual sugar related to the quality:

```
WhiteWine%>%  ggplot( aes(x = residual.sugar, y = density, color = factor(quality))) +
  geom_point(alpha = 1/2, position = position_jitter(h = 0), size = 2) +
  coord_cartesian(xlim=c(min(WhiteWine$residual.sugar),max(WhiteWine$residual.sugar)),
                  ylim=c(min(WhiteWine$density),max(WhiteWine$density))) +
  scale_color_brewer(type='qual') +
  xlab('Residual Sugar') +
  ylab('Density') +
  ggtitle('Density vs. Residual sugar correlation by Quality')
```



There are two outlier related to a 6 level quality, we zoom on the scatterplot to see better the distribution:

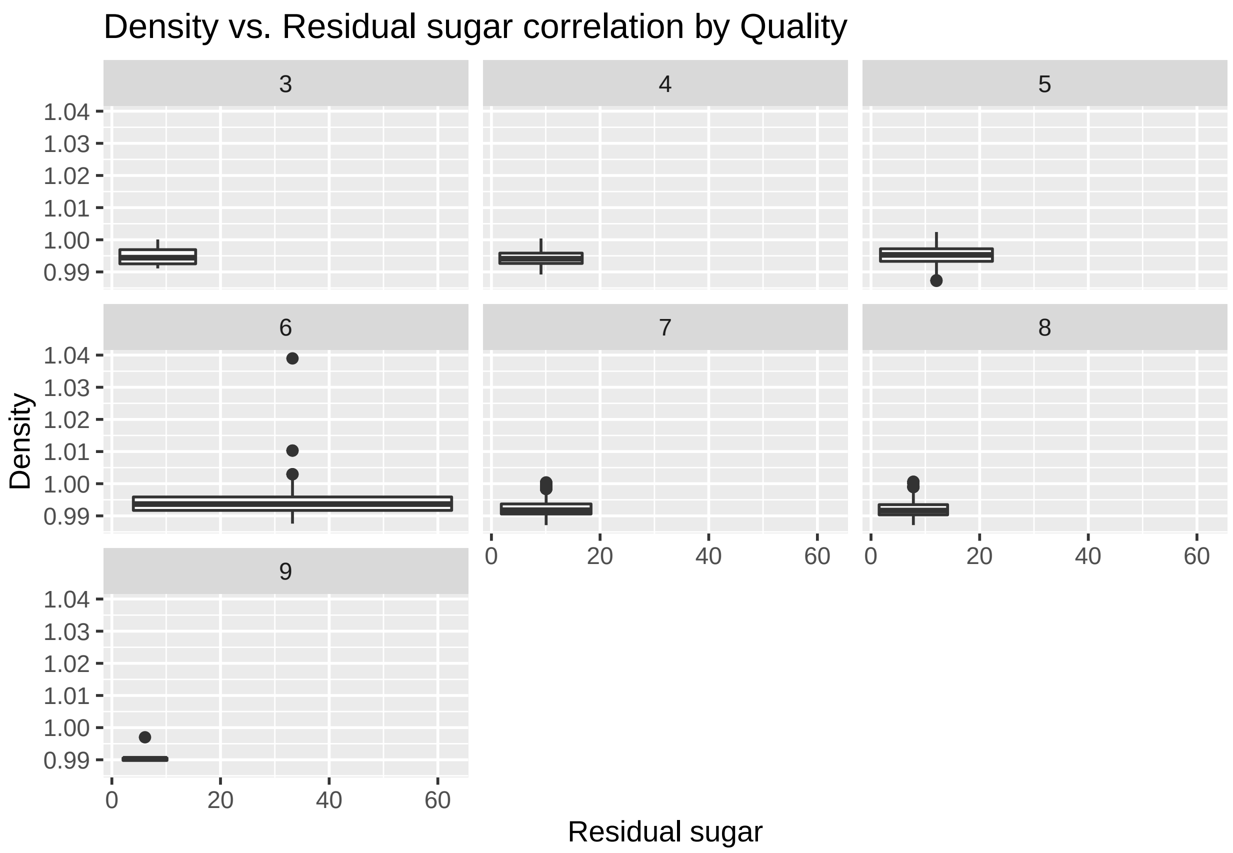
```
WhiteWine%>%  ggplot( aes(x = residual.sugar, y = density, color = factor(quality))) +
  geom_point(alpha = 1/2, position = position_jitter(h = 0), size = 2) +
  coord_cartesian(xlim=c(min(WhiteWine$residual.sugar),25),
                  ylim=c(min(WhiteWine$density),1.01)) +
  scale_color_brewer(type='qual') +
  xlab('Residual Sugar') +
  ylab('Density') +
  ggtitle('Density vs. Residual sugar correlation by Quality')
```



The higher quality (i.e. quality = 7) wines are concentrated on the left bottom corner in the scatter plot which represent low Residual sugar and low density; which mean low density and low sugar level give better quality.

Below we take a look to the boxplot of the correlation between Density and Residual Sugar related by Quality:

```
WhiteWine%>%ggplot(aes(x = residual.sugar, y = density, group = quality) )+
  facet_wrap( ~ quality) +
  geom_boxplot() +
  xlab('Residual sugar') +
  ylab('Density') +
  ggtitle('Density vs. Residual sugar correlation by Quality')
```



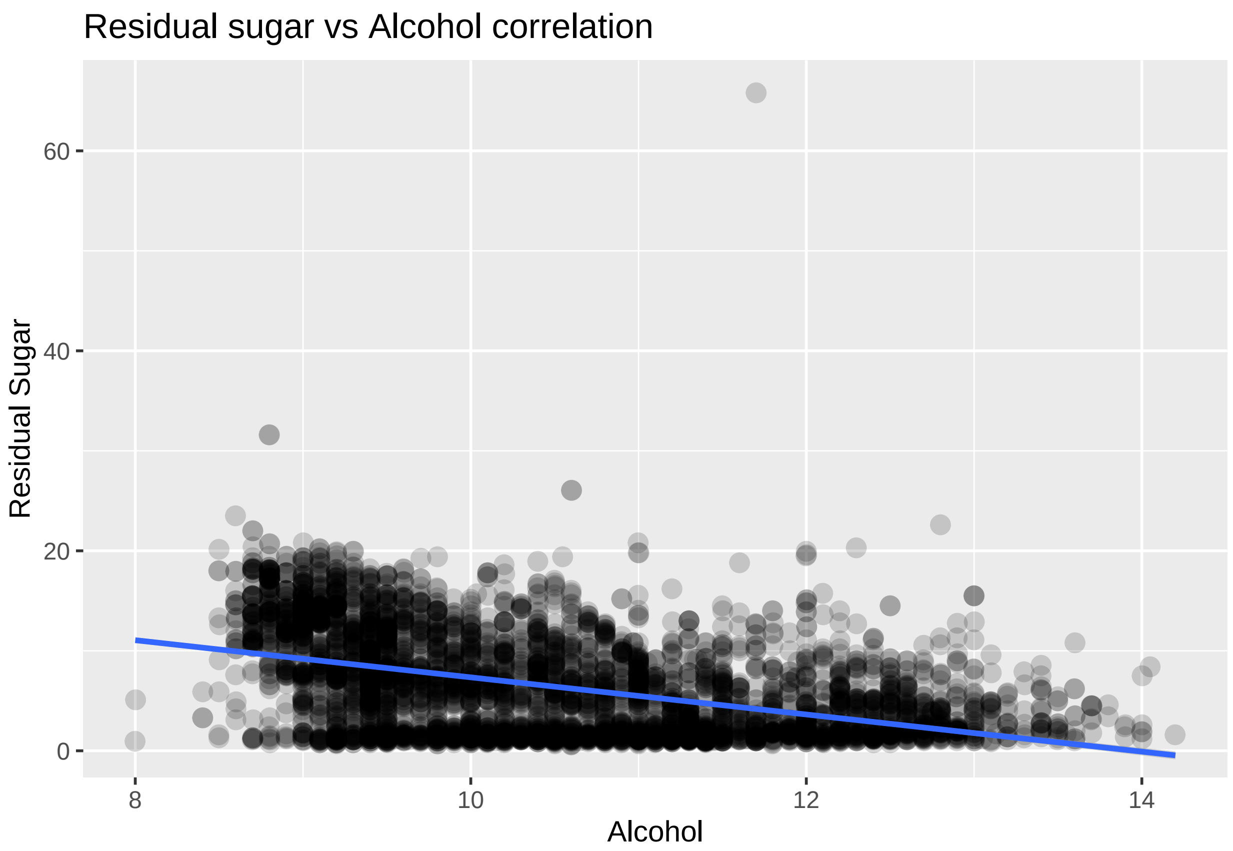
We can confirm that the quality is higher when the density and the Residual Sugar are low.

2.2.3 Residual Sugar vs Alcohol

As saw before, the Residual Sugar and Alcohol are negatively correlated, we plot below the correlation:

```
WhiteWine%>% ggplot(
  aes(x = alcohol, y = residual.sugar)) +
  geom_point(alpha = 1/6, position = position_jitter(h = 0), size = 3) +
  geom_smooth(method = 'lm') +
  coord_cartesian(xlim=c(min(WhiteWine$alcohol),max(WhiteWine$alcohol)),
                  ylim=c(min(WhiteWine$residual.sugar),max(WhiteWine$residual.sugar))) +
  xlab('Alcohol') +
  ylab('Residual Sugar') +
  ggtitle('Residual sugar vs Alcohol correlation')
```

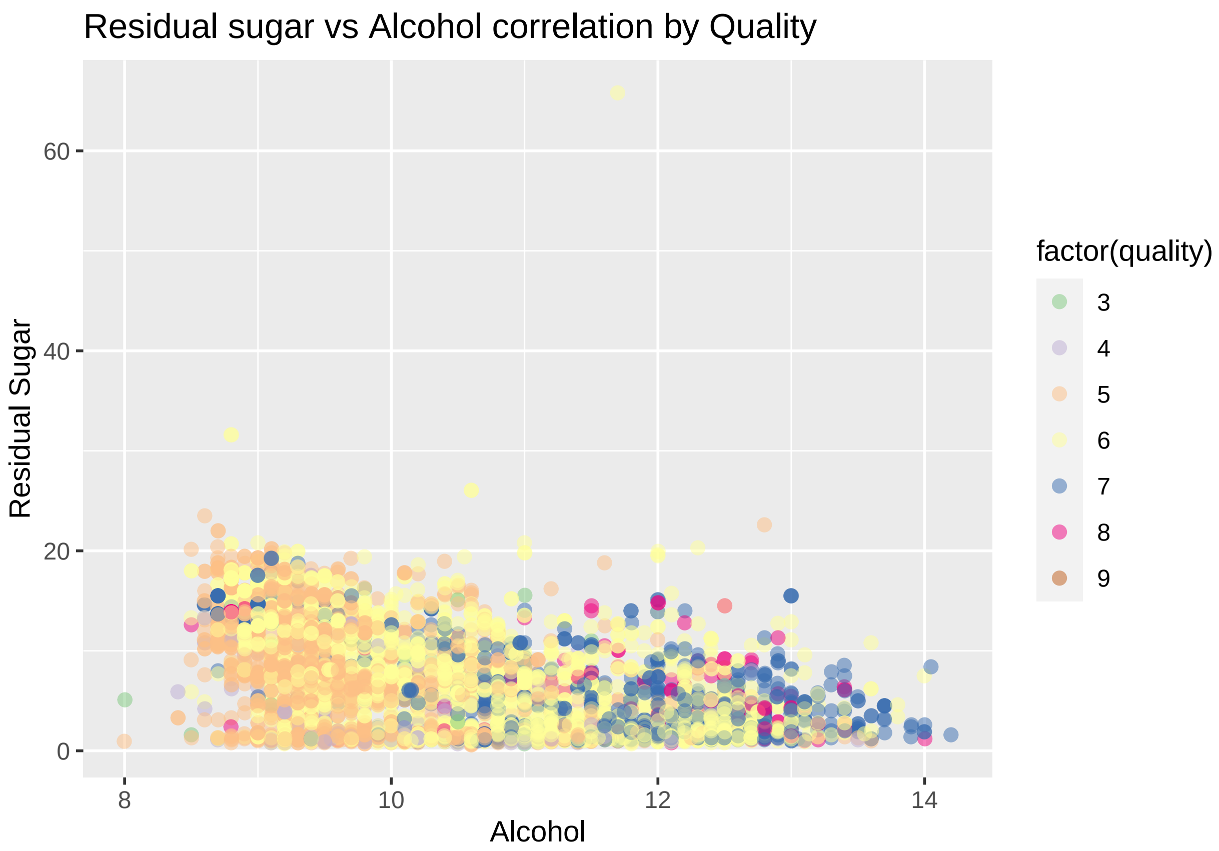
```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



We can see that when Alcohol increase, the Residual Sugar decrease.

Below, we investigate the correlation between the Residual Sugar and Alcohol related to the quality:

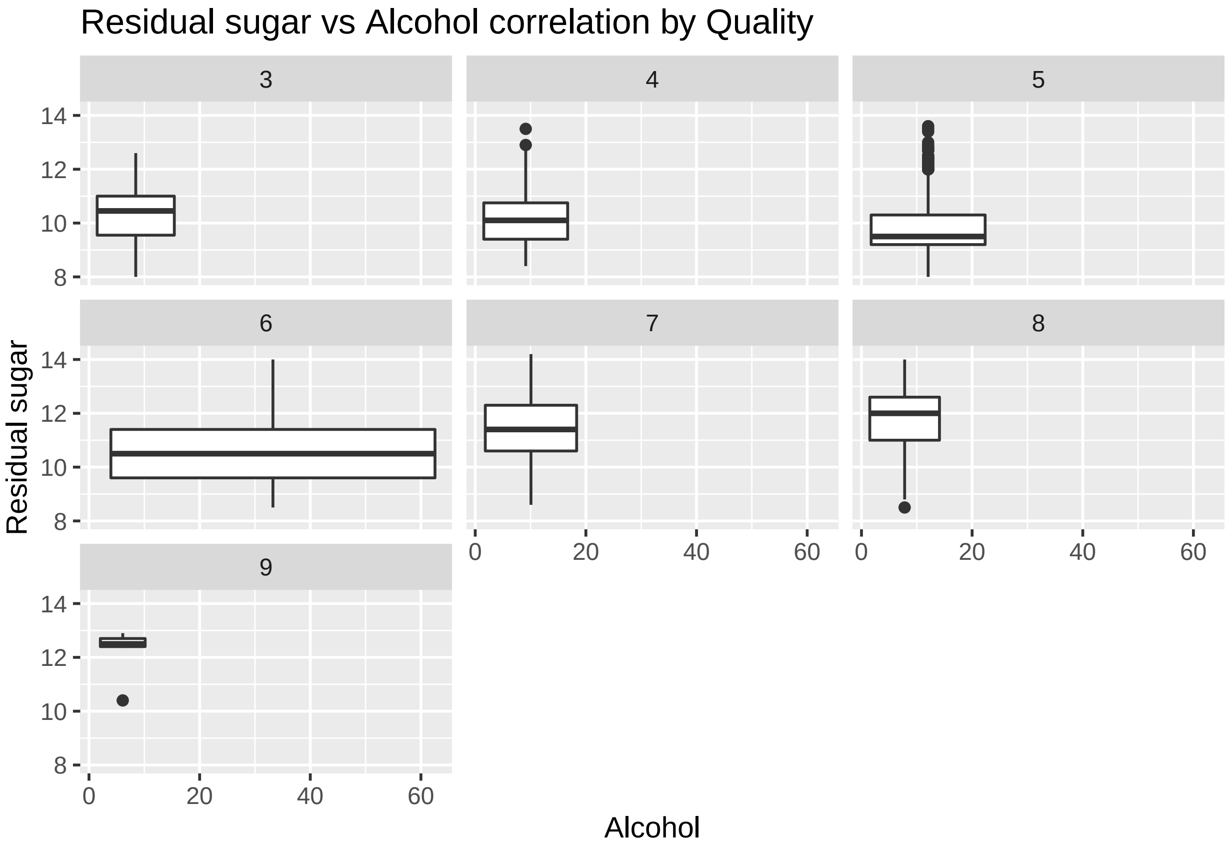
```
WhiteWine%>%  ggplot( aes(x = alcohol , y = residual.sugar, color = factor(quality))) +
  geom_point(alpha = 1/2, position = position_jitter(h = 0), size = 2) +
  coord_cartesian(xlim=c(min(WhiteWine$alcohol),max(WhiteWine$alcohol)),
                  ylim=c(min(WhiteWine$residual.sugar),max(WhiteWine$residual.sugar))) +
  scale_color_brewer(type='qual') +
  xlab('Alcohol') +
  ylab('Residual Sugar') +
  ggtitle('Residual sugar vs Alcohol correlation by Quality')
```



The quality 7-8 levels wines are concentrated on the right bottom corner in the scatter plot which represent low Residual Sugar and high alcohol.

Below we take a look to the boxplot of the correlation between Residual Sugar and Alcohol related by Quality:

```
WhiteWine%>%ggplot(aes(x = residual.sugar, y = alcohol, group = quality) )+
  facet_wrap( ~ quality) +
  geom_boxplot() +
  xlab('Alcohol') +
  ylab('Residual sugar') +
  ggtitle('Residual sugar vs Alcohol correlation by Quality')
```



We can confirm that low Residual Sugar and high alcohol give better quality.

From these analysis we saw that there is a relation between the quality, the alcohol and the sugar. In particular:

- when the density is low, the quality is high
- when the density is low, the sugar is low
- when the sugar is low, the alcohol is high

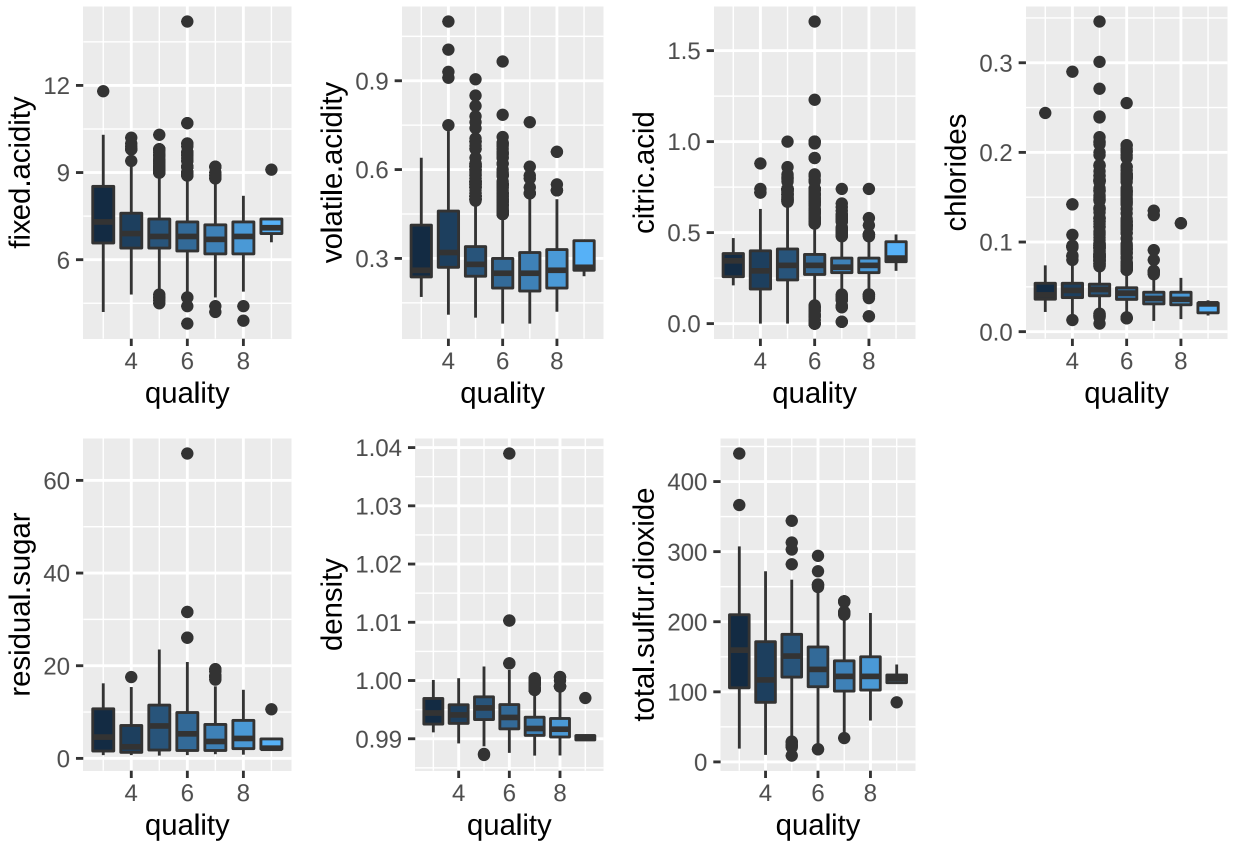
From these statement, we can think that when alcohol is high, the quality is high. This is confirmed by the positive correlation value that is equals to +0.44.

2.2.4 Quality

From the correlation matrix, we can see that the Quality is negatively correlated with the major of variables:

- Acid: fixed.acidity(-0.11), volatile.acidity(-0.19), citric.acid(-0.01)
- Salt: chlorides(-0.21)
- Sugar: residual.sugar(-0.1)
- Physical: density(-0.31)
- Chemicals: total.sulfur.dioxide(-0.17)

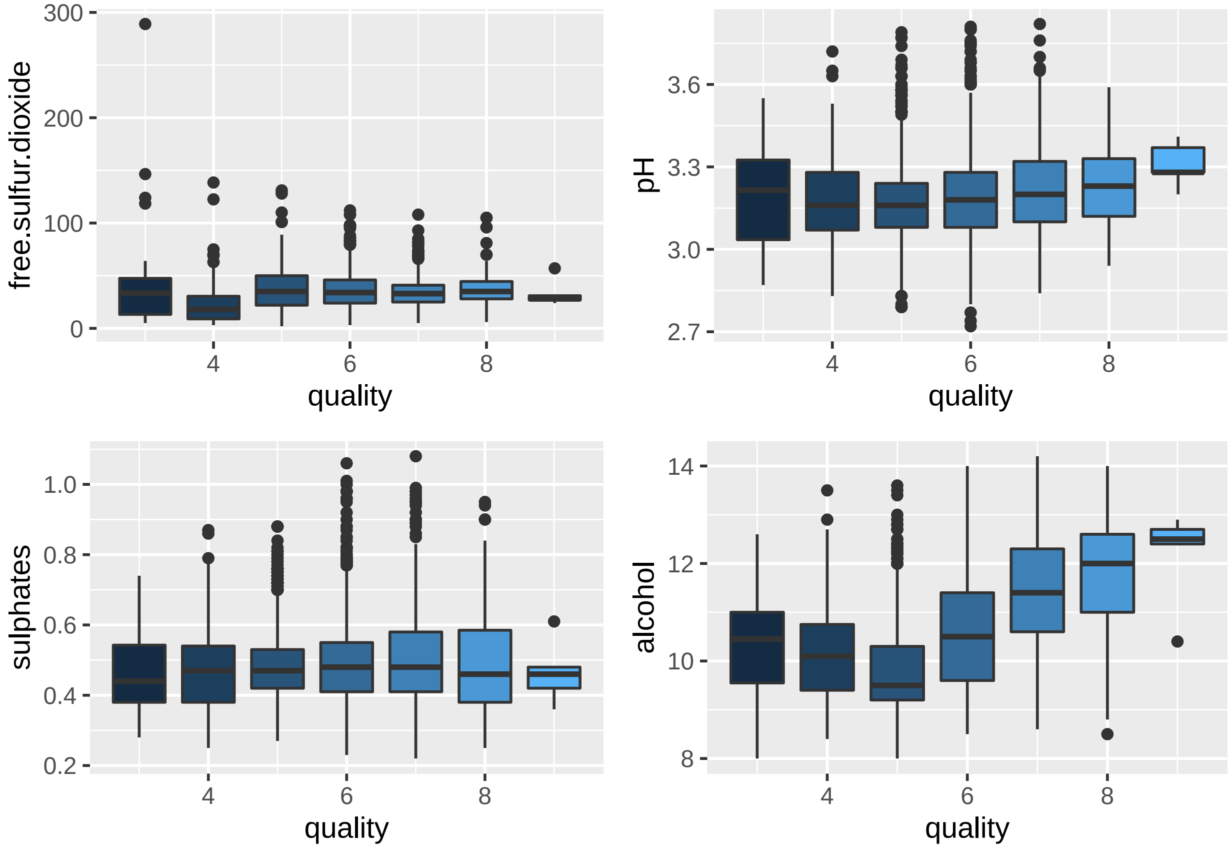
```
fa_n<-WhiteWine%>%ggplot(aes(x = quality, y = fixed.acidity, group=quality, fill=quality) )+
  geom_boxplot(show.legend = FALSE)
va_n<-WhiteWine%>%ggplot(aes(x = quality, y = volatile.acidity, group=quality, fill=quality) )+
  geom_boxplot(show.legend = FALSE)
ca_n<-WhiteWine%>%ggplot(aes(x = quality, y = citric.acid, group=quality, fill=quality) )+
  geom_boxplot(show.legend = FALSE)
cl_n<-WhiteWine%>%ggplot(aes(x = quality, y = chlorides, group=quality, fill=quality) )+
  geom_boxplot(show.legend = FALSE)
rs_n<-WhiteWine%>%ggplot(aes(x = quality, y = residual.sugar, group=quality, fill=quality) )+
  geom_boxplot(show.legend = FALSE)
de_n<-WhiteWine%>%ggplot(aes(x = quality, y = density, group=quality, fill=quality) )+
  geom_boxplot(show.legend = FALSE)
ts_n<-WhiteWine%>%ggplot(aes(x = quality, y = total.sulfur.dioxide, group=quality, fill=quality) )+
  geom_boxplot(show.legend = FALSE)
grid.arrange(fa_n,va_n,ca_n,cl_n,rs_n,de_n,ts_n,ncol=4)
```



But positively correlated with the following:

- Chemicals: free.sulfur.dioxide(0.01), pH(0.1), sulphates(0.05)
- Alcohol: alcohol(0.44)

```
fs_p<-WhiteWine%>%ggplot(aes(x = quality, y = free.sulfur.dioxide, group=quality, fill=quality) )+
  geom_boxplot(show.legend = FALSE)
ph_p<-WhiteWine%>%ggplot(aes(x = quality, y = pH, group=quality, fill=quality) )+
  geom_boxplot(show.legend = FALSE)
su_p <-WhiteWine%>%ggplot(aes(x = quality, y = sulphates, group=quality, fill=quality) )+
  geom_boxplot(show.legend = FALSE)
al_p<-WhiteWine%>%ggplot(aes(x = quality, y = alcohol, group=quality, fill=quality) )+
  geom_boxplot(show.legend = FALSE)
grid.arrange(fs_p,ph_p,su_p,al_p,ncol=2)
```



2.3 Data preparation & Data Cleaning

In order to create our classification models, we add a new column, named Goodness, which indicates whether the wine is good or not base on the column “quality”. In particular, we create the binary column Goodness that is equals to 1 if the quality’s value is greater than 5 (i.e. the wine is good) and equals to 0 if the quality’s value is less than or equals to 5 (i.e. the wine is not good):

```
WhiteWine <-WhiteWine %>% mutate(Goodness = as.factor(ifelse(quality >5, 1,0)))
head(WhiteWine)
```

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

Below we see how many good wines and how many not good wine are in the dataset, using the new column Goodness:

```
kable(table((WhiteWine$Goodness)))
```

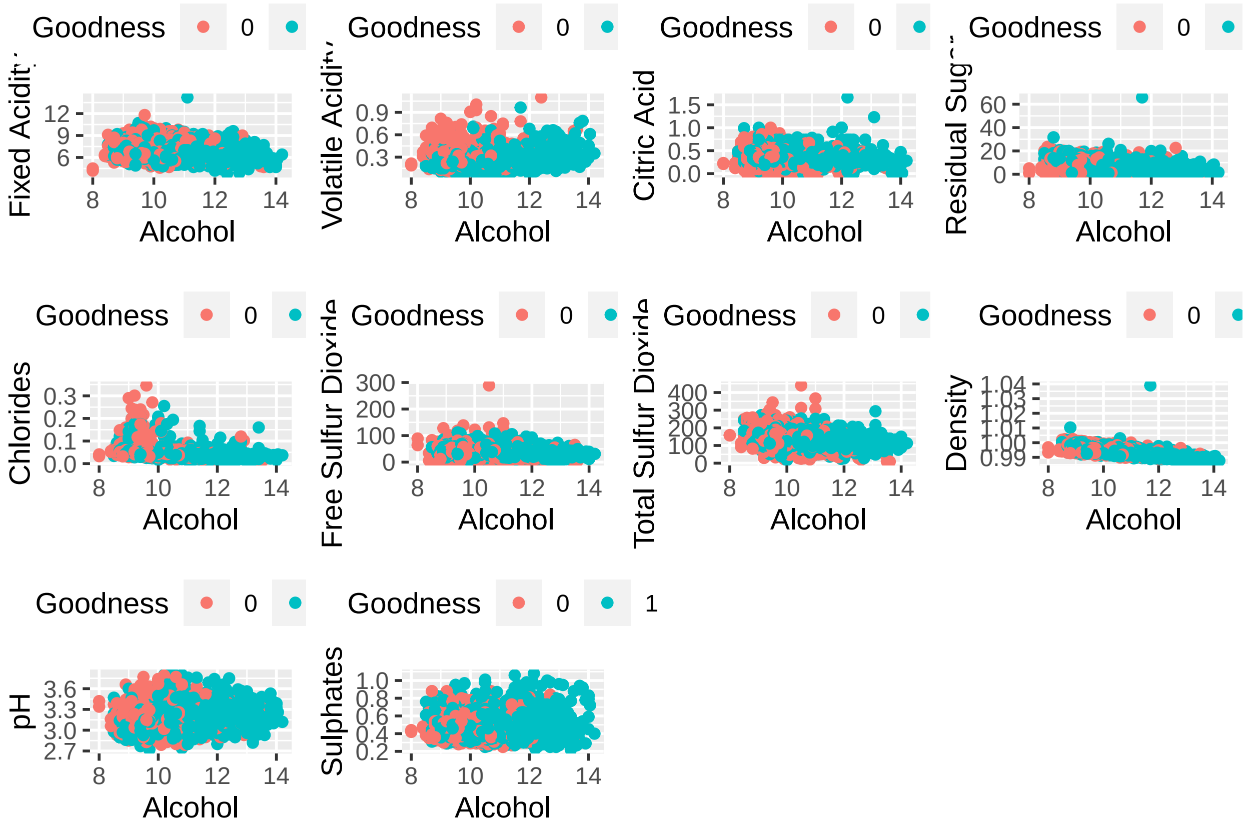
Var1	Freq
0	1640
1	3258

We have 1640 observation of a not good wine and 3258 of a good wine. Based on these results, it seem like a far enough number of good and not good. So we decide do not apply any resampling of the dataset.

With the new column, we can see some relation between the variables:

Below the comparison between the Alcohol and te other features:

```
faA<-WhiteWine%>%ggplot(aes(x=alcohol, y=fixed.acidity,  group=Goodness, col=Goodness)) +
  geom_point() +  labs(x="Alcohol" , y="Fixed Acidity")  + theme(legend.position="top")
vaA<-WhiteWine%>%ggplot(aes(x=alcohol, y=volatile.acidity,  group=Goodness, col=Goodness)) +
  geom_point() +  labs(x="Alcohol" , y="Volatile Acidity")  + theme(legend.position="top")
caA<-WhiteWine%>%ggplot(aes(x=alcohol, y=citric.acid,  group=Goodness, col=Goodness)) +
  geom_point() +  labs(x="Alcohol" , y="Citric Acid")  + theme(legend.position="top")
rsA<-WhiteWine%>%ggplot(aes(x=alcohol, y=residual.sugar,  group=Goodness, col=Goodness)) +
  geom_point() +  labs(x="Alcohol" , y="Residual Sugar")  + theme(legend.position="top")
clA<-WhiteWine%>%ggplot(aes(x=alcohol, y=chlorides,  group=Goodness, col=Goodness)) +
  geom_point() +  labs(x="Alcohol" , y="Chlorides")  + theme(legend.position="top")
fsA<-WhiteWine%>%ggplot(aes(x=alcohol, y=free.sulfur.dioxide,  group=Goodness, col=Goodness)) +
  geom_point() +  labs(x="Alcohol" , y="Free Sulfur Dioxide") + theme(legend.position="top")
tsA<-WhiteWine%>%ggplot(aes(x=alcohol, y=total.sulfur.dioxide,  group=Goodness, col=Goodness)) +
  geom_point() +  labs(x="Alcohol" , y="Total Sulfur Dioxide") + theme(legend.position="top")
deA<-WhiteWine%>%ggplot(aes(x=alcohol, y=density,  group=Goodness, col=Goodness)) +
  geom_point() +  labs(x="Alcohol" , y="Density") + theme(legend.position="top")
pHA<-WhiteWine%>%ggplot(aes(x=alcohol, y=pH,  group=Goodness, col=Goodness)) +
  geom_point() +  labs(x="Alcohol" , y="pH")  + theme(legend.position="top")
suA<-WhiteWine%>%ggplot(aes(x=alcohol, y=sulphates,  group=Goodness, col=Goodness)) +
  geom_point() +  labs(x="Alcohol" , y="Sulphates") + theme(legend.position="top")
grid.arrange(faA,vaA,caA,rsA,clA,fsA,tsA,deA,pHA,suA,ncol=4)
```



Some considerations:

- In all the plot, we see that the blue dots (good wines) are on the right and the red dots (not good wine) on the left
- the good wines have high alcohol and low respective feature (i.e. density, Residual Sugar etc);
- the alcohol is positively correlated with "volatile acidity" and pH;
- the are lightly positively correlated (+0.12). In the plot of pH vs Alcohol, we see that the good wines (blue dots) are more on the middle right side, which means high alcohol and pH between 3.0 and 3.45.

From this point forward, we analyze the dataset using the new column Goodness, so we remove the column quality from the dataset that is not useful:

```
WhiteWine <-WhiteWine %>% select(-quality)
```

2.3.1 Modeling approach

From the previous analysis we discovered that the alcohol and the quality are positively correlated, a good wine has high alcohol. Furthermore the alcohol is correlated with all the other features. So the quality is also implicitly correlated with them. We decide to valuate our model with all the wine features.

2.3.1.1 Feature variable and target variable

We now separate the feature variables and the target variable.

We create the feature variable WhiteWine_x which is the data set, without the feature goodness that we are trying to predict:

```
WhiteWine_x <- WhiteWine[,-12]
head(WhiteWine_x)
```

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

and the target variable WhiteWine_y which is the feature we are trying to predict:

```
WhiteWine_y <- WhiteWine$Goodness
head(WhiteWine_y)
```

```
## [1] 1 1 1 1 1 1
## Levels: 0 1
```

We verify their dimensions:

```
dim(WhiteWine_x)
```

```
## [1] 4898   11
```

```
length(WhiteWine_y)
```

```
## [1] 4898
```

The WhiteWine_x is a dataset with 4898 rows and 11 columns, the WhiteWine_y is a vector with 4898 rows.

2.3.1.2 The training and test set

We create the training set and the test set from the two variables created, WhiteWine_x and WhiteWine_y. In particular, we create the train_WWx that is the 90% of WhiteWine_x and the test_WWx that is the remaining 10%. Equally, the train_WWy and the test_WWy.

```
set.seed(3,sample.kind = "Rounding")# if using R 3.5 or earlier, use `set.seed(3)`
```

```
## Warning in set.seed(3, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
test_index <- createDataPartition(WhiteWine_y,times = 1, p=0.2,list = FALSE)
test_WWx <- WhiteWine_x[test_index,]
test_WWy <- WhiteWine_y[test_index]
train_WWx <- WhiteWine_x[-test_index,]
train_WWy <- WhiteWine_y[-test_index]
```

We look at their dimensions:

```
dim(train_WWx)
```

```
## [1] 3918   11
```

```
dim(test_WWx)
```

```
## [1] 980  11
```

```
length(train_WWy)
```

```
## [1] 3918
```

```
length(test_WWy)
```

```
## [1] 980
```

The train_WWx has 3918 rows and the test_WWx has 980 rows, both with 11 columns. So, equally, the train_WWy has 3918 rows and the test_WWy has 980 rows.

We check if the training and test sets have similar proportions of good and not good wine:

```
meanGoodtrain_WWy <-mean(train_WWy == 1)
meanGoodtest_WWy <-mean(test_WWy == 1)
meanNotGoodtrain_WWy <-mean(train_WWy == 0)
meanNotGoodtest_WWy <- mean(test_WWy == 0)

MeanSet <- c("Mean Good train_WWy","Mean Good test_WWy", "Mean not Good train_WWy","Mean not Good test_
Value <- c(meanGoodtrain_WWy,meanGoodtest_WWy,meanNotGoodtrain_WWy,meanNotGoodtest_WWy)

data.frame(MeanSet= MeanSet, Value = Value)
```

```
##                   MeanSet     Value
## 1     Mean Good train_WWy 0.6651353
## 2      Mean Good test_WWy 0.6653061
## 3 Mean not Good train_WWy 0.3348647
## 4  Mean not Good test_WWy 0.3346939
```

The mean between the “Good train set” and the “Good test set” are almost the same, also between the “Not Good train set” and the “Not Good test set”.

3 Results

In this chapter, we create six different machine learning models: K nearest neighbors, Logistic regression, Loess, LDA, QDA and Random forest. Finally, we compare them by their accuracy.

3.1 K-nearest neighbors model

We train a K-nearest neighbors model on the training set using the caret package. We consider a tuning vector with odd values from 3 to 21:

```
tuning <- data.frame(k = seq(3, 21, 2))
WW_train_knn <- train(train_WWx, train_WWy,
                      method = "knn",
                      tuneGrid = tuning)
WW_knn_preds <- predict(WW_train_knn, test_WWx)
WW_knnV <- mean(WW_knn_preds == test_WWy)
```

The final value of k used in the model is 15 and the accuracy of the K-nearest neighbors model on the test set is equals to 0.6959184.

3.2 Logistic regression model

We fit a Logistic regression model on the training set with caret::train():

```
WW_glm <- train(train_WWx, train_WWy, method = "glm")
WW_glm_preds <- predict(WW_glm, test_WWx)
WW_glmV<-mean(WW_glm_preds == test_WWy)
```

The accuracy of the logistic regression model on the test set is equals to 0.7357143.

3.3 Loess model

We fit a Loess model on the training set with the caret package. We use the default tuning grid:

```
WW_loess <- train(train_WWx, train_WWy, method = "gamLoess")
WW_loess_preds <- predict(WW_loess, test_WWx)
WW_loessV<-mean(WW_loess_preds == test_WWy)
```

The accuracy of the loess model on the test set is equals to 0.7571429.

3.4 LDA model and a QDA model

Now we fit a LDA model and a QDA model on the training set:

```
WW_lda <- train(train_WWx, train_WWy, method = "lda")
WW_lda_preds <- predict(WW_lda, test_WWx)
WW_ldaV <-mean(WW_lda_preds == test_WWy)

WW_qda <- train(train_WWx, train_WWy, method = "qda")
WW_qda_preds <- predict(WW_qda, test_WWx)
WW_qdaV<-mean(WW_qda_preds == test_WWy)
```

The accuracy of the LDA model on the test set is equals to 0.7326531 and the accuracy of the QDA on the test set is equals to 0.7326531.

3.5 Random Forest model

We train a Random Forest on the training set using the caret package. It takes some minutes:

```
tuningRF <- data.frame(mtry = c(3, 5, 7, 9))
WW_RF <- train(train_WWx, train_WWy,
                  method = "rf",
                  tuneGrid = tuningRF,
                  importance = TRUE)
WW_RF_preds <- predict(WW_RF, test_WWx)
WW_RFV <-mean(WW_RF_preds == test_WWy)
```

The value of tuningRF which gives the highest accuracy is 3; the accuracy of the Random Forest model on the test set is equals to 0.8489796.

3.6 Final results

We show below a table with all the accuracy obtained by the 6 models:

```
models <- c("K nearest neighbors", "Logistic regression","Loess", "LDA", "QDA","Random forest" )
accuracy <- c( WW_knnV,
              WW_glmV,
              WW_loessV,
              WW_ldaV,
              WW_qdaV,
              WW_RFV)
data.frame(Model = models, Accuracy = accuracy)
```

```
##                 Model  Accuracy
## 1 K nearest neighbors 0.6959184
## 2 Logistic regression 0.7357143
## 3               Loess 0.7571429
## 4                 LDA 0.7326531
## 5                 QDA 0.7326531
## 6       Random forest 0.8489796
```

We can see that the best accuracy is given by the Random Forest's model and the worst one is given by the K nearest neighbors's model.

Conclusion

In this report we showed the creation of some machine learning algorithms to predict if a wine is good or not. We started by studying the dataset, the correlation between the variables and studying the different models.

We found that the final Random Forest model is the better model which gives us an accuracy equals to 0.8489796 that is a good accuracy.

Other next possible works:

- Analyze also the Red Wine dataset on correlation and relationship about the quality and valuate the accuracy on the same machine learning models
- Use the two dataset together, for seeing possible differences in the features and valuate the accuracy of the Machine Learning models using more data