Choose Your Own: Predicting A Wine's Quality Rating

Marta Acedos Serrano

Content

Choose Your Own:

Predicting A Wine's Quality Rating

Marta Acedos Serrano

27/2/2021

1. Introduction

The dataset used has been downloaded from the website Kaggle, and the dataset comprises different qualities of different portuguese red wines and the quality rating for each wine. The dataset can be found in the following website: https://www.kaggle.com/rajyellow46/wine-quality. The variables of the dataset are the quality rating and the following wine characteristics: fixed acidity, citric acid, chlorides, total sulfur dioxide, pH, alcohol, volatile acidity, residual sugar, free sulfur dioxide, density and sulphates. The purpose of the project is to predict the quality rating of a wine given the different features of that wine. To do so we need to explore how each characteristic affects the wine's quality in order to study which features can help us predict the quality of the wine. Once this study is conducted, taking into account how each variable can explain a wine's quality a predictive model is constructed in order to predict a wine's quality rating given the relevant features of the wine.

On the other hand, once the best model to predict a wine's quality rating has been selected it is desirable to test its predictive power with a dataset that has not been used at all to fit it. Therefore, the original dataset is splitted into "wine" and "validation". The "wine" dataset is used to develop the different models, and once the best model is selected the "validation" set is used to test its predictive power: the model is used to predict the quality rating of the wines of this dataset and with these predictions the RMSE is computed. This is, "validation" will be used solely to test the final selected model. Since any of the data of "validation" will be used in the model development part -only the "wine" dataset will be used to develop the model-, the RMSE obtained when testing the model using "validation" constitutes a good measure of the predictive power of the model. Furthermore, since the original dataset is not too big and it is desirable to have enough data in "validation" to test the predictive power of the selected model, "validation" contains 15% of the original dataset and "wine" contains 85% of the original dataset.

2. Analysis

2.1. Data Management

First of all, we need to explore the data in order to assess wheter the data shall be cleaned and how it shall be cleaned. Thus, we check the class of each variable and whether there are any missing values -the code is not included in the report so that it is not too long, but it has been included in the script. In this preliminary exploration we don't find any missing values.

Additionally, the main statistics of each variable are computed and summarized in the following table:

Variables	Mean	SD	Max	Min	Median
Fixed Acidity	8.3174521	1.7153467	15.90000	4.6000	7.900000
Citric Acid	0.2683652	0.1944521	1.00000	0.0000	0.255000
Chlorides	0.0875810	0.0463835	0.61100	0.0120	0.079000
Total Sulfur Dioxide	46.8527246	33.1649152	289.00000	6.0000	38.000000
Density	0.9967688	0.0018521	1.00369	0.9902	0.996765
Sulphates	0.6591016	0.1727158	2.00000	0.3300	0.620000
Quality	5.6347570	0.8174765	8.00000	3.0000	6.000000

As it can be seen in the data, the lowest quality rating a wine can have is 3, and the maximum rating a wine can have is 8. Furthermore, the mean rating is approximately 5.

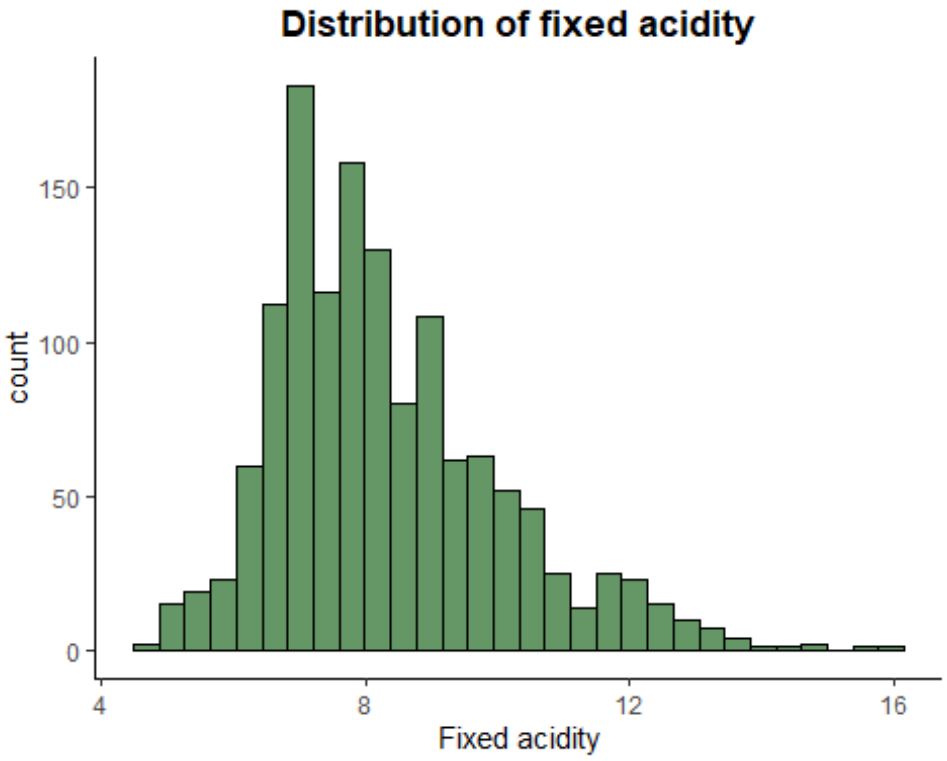
On the other hand, to enhance the exploratory data analysis a boolean variable indicating whether the wine is good or not is added. This variable ("good") takes the value 1 when the quality rating is equal or superior to 6, and 0 if the quality rating is less than 6. This is, a wine is considered to be good if it has a rating equal or superior to 6.

```
#add boolean variable indicating whether the quality is good (>=6)
#to enhance the exploratory data anaylisis
wine <- wine %>% mutate(good = ifelse(quality >=6,1,0))
```

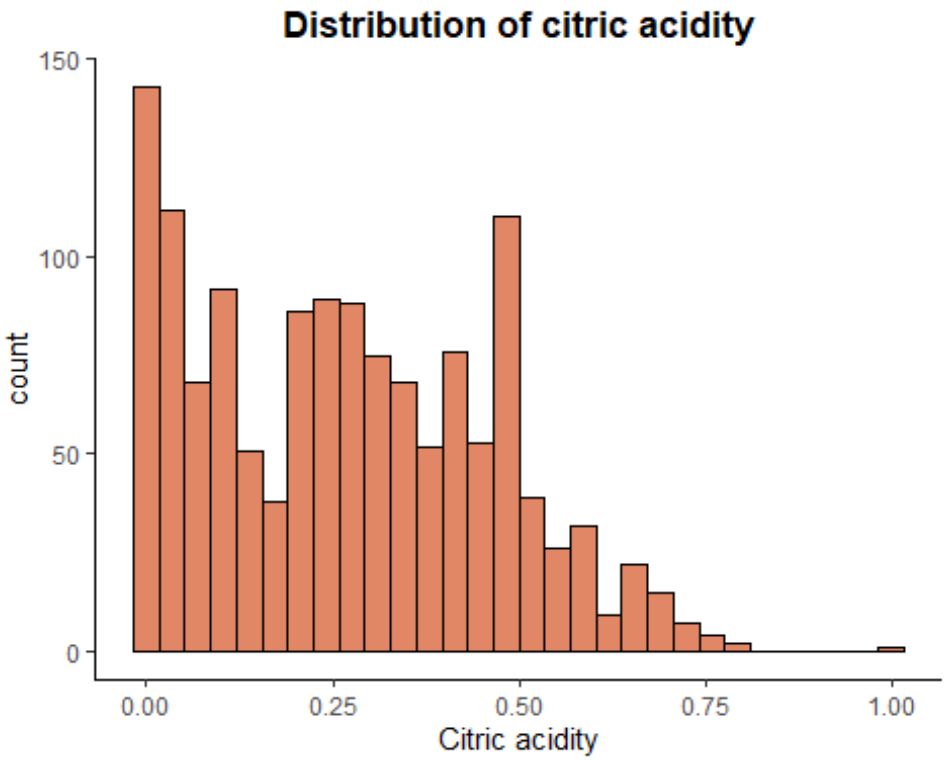
2.2. Exploratory Data Analysis

Once the data is loaded and the variable "good" is added we conduct an exploratory data analysis. Furthermore, the analysis is conducted using the "wine" dataset because the different models will be trained using this dataset. Thus, since this analysis aims to identify which variables shall be included in the models, it is conducted using the dataset that will be used to develop those models. Firstly, we explore the distribution of the different variables using histograms.

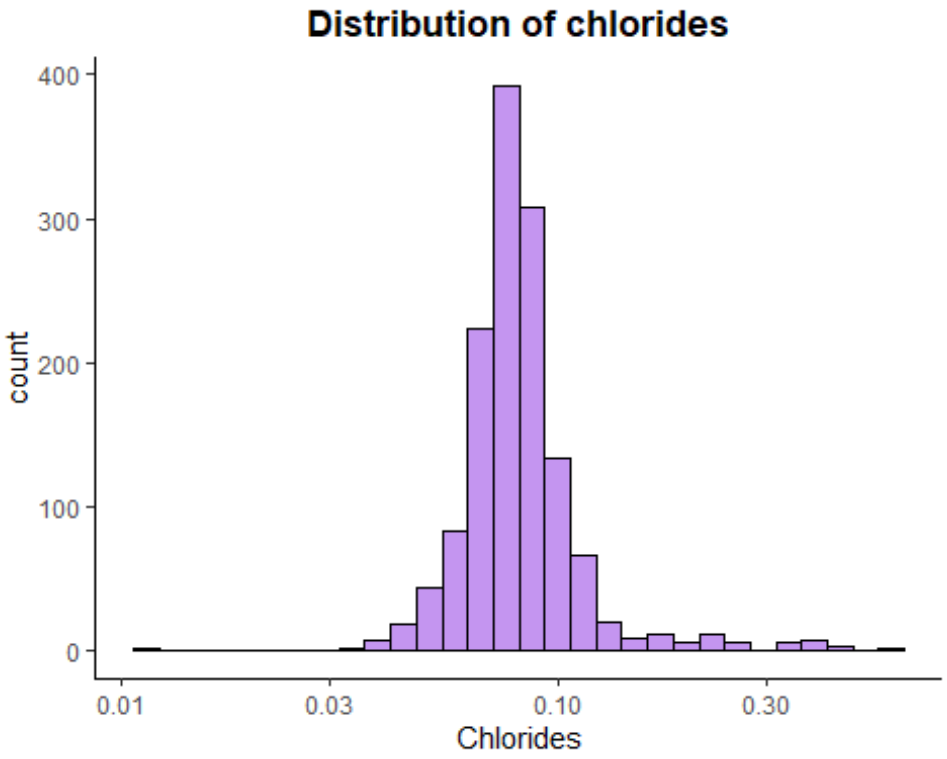
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



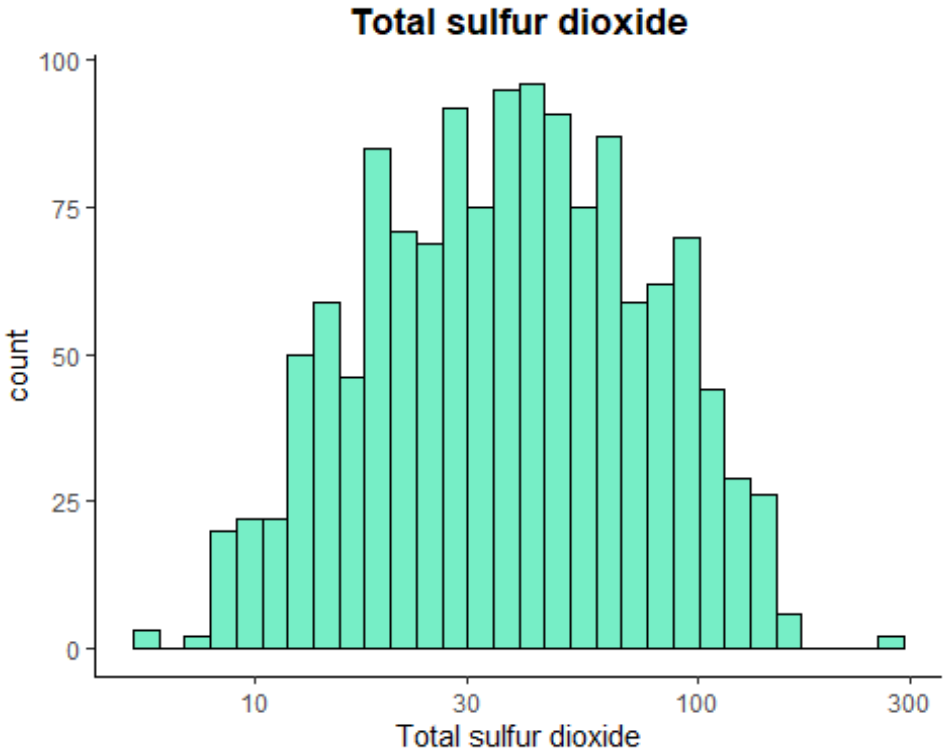

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



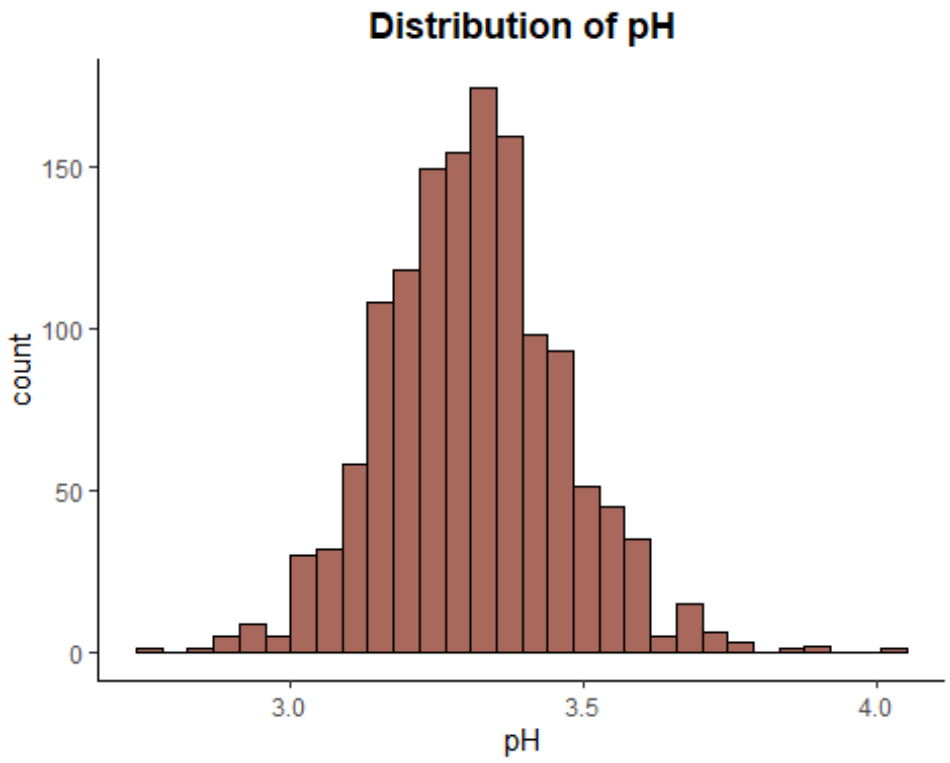

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



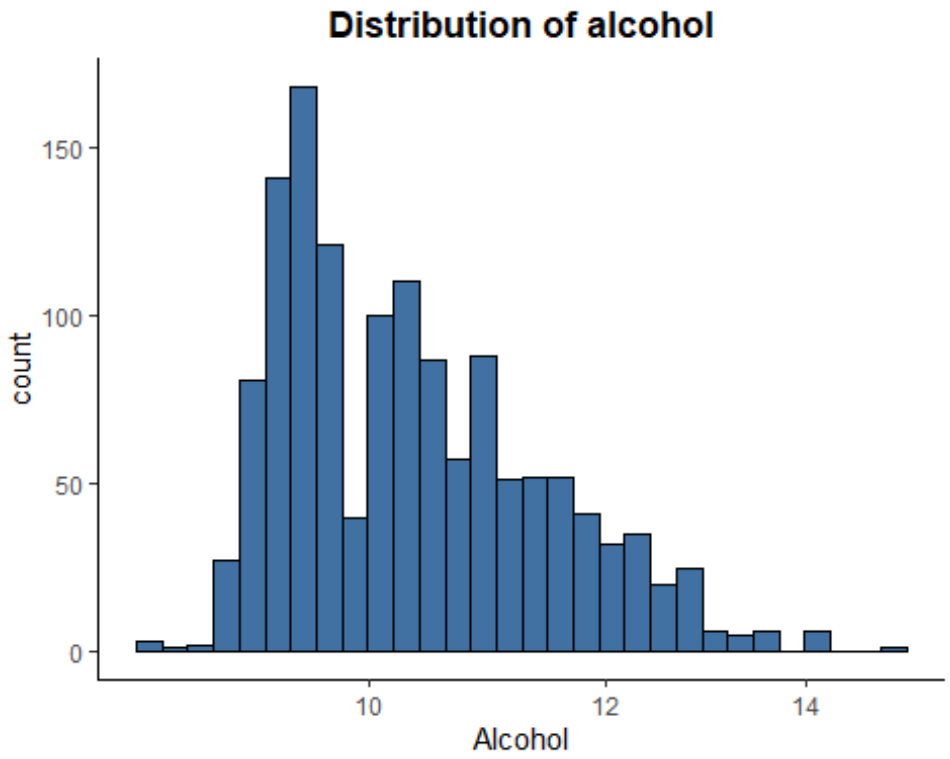

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



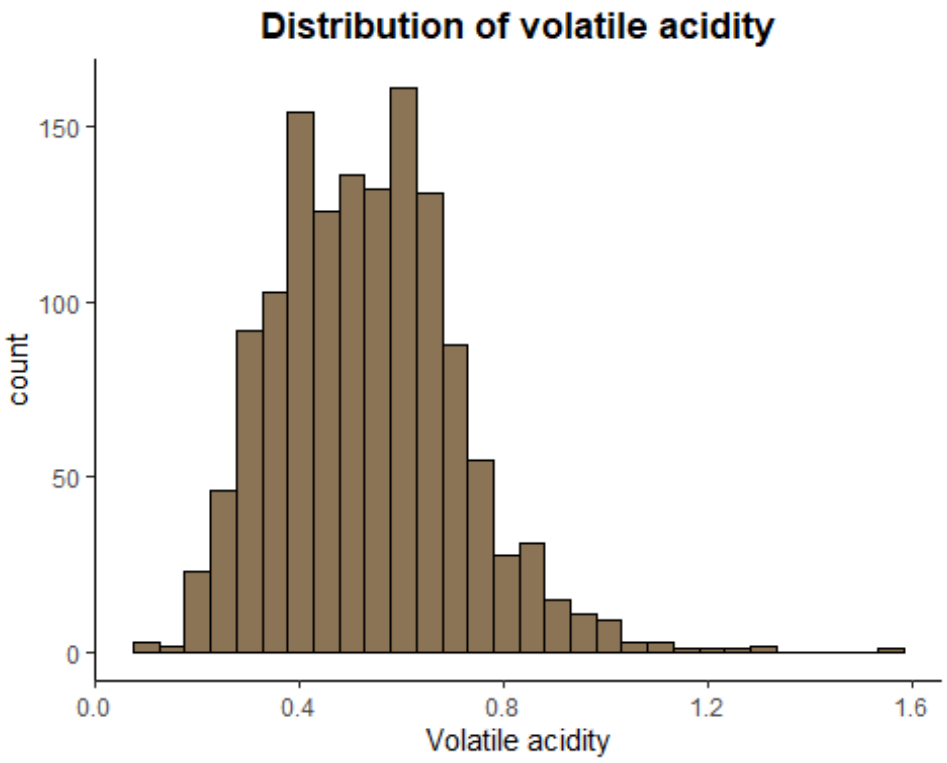

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



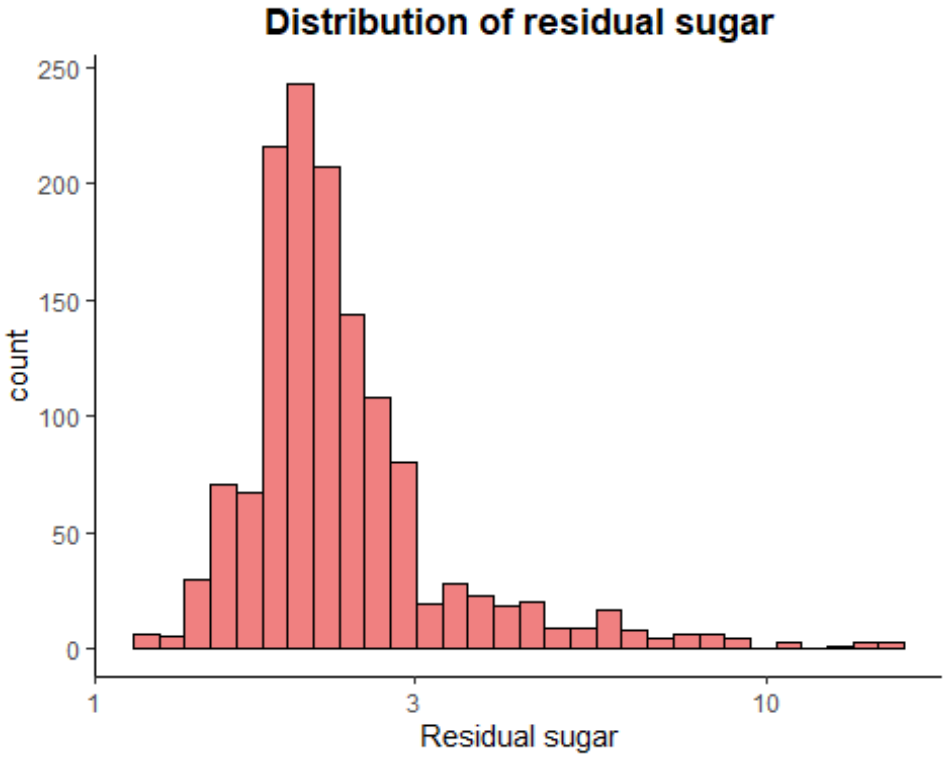

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



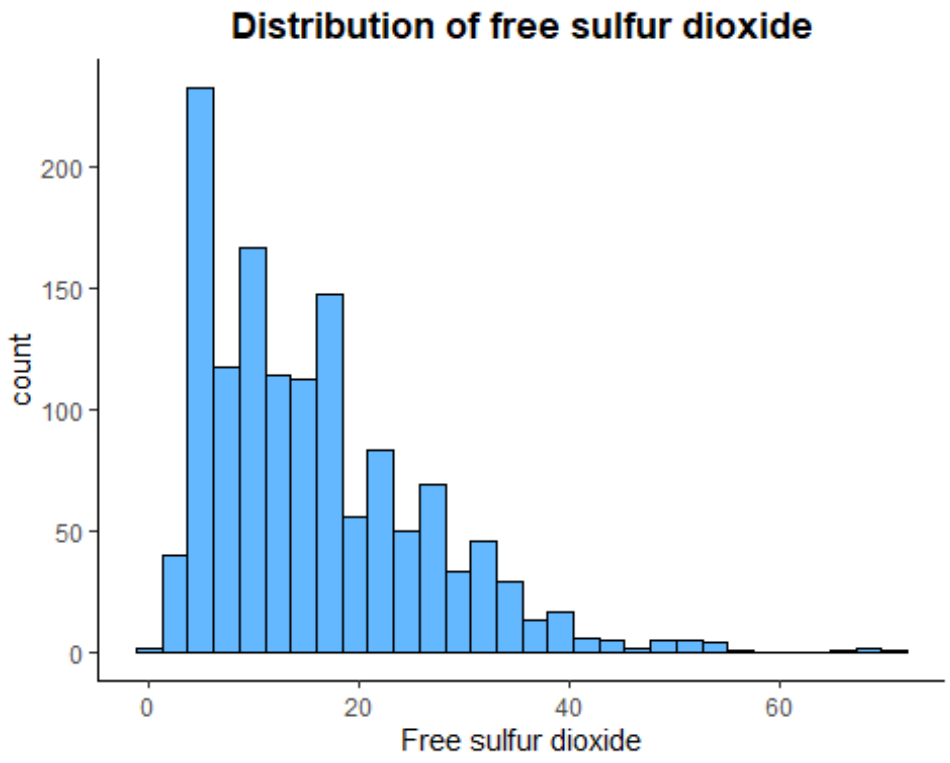

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



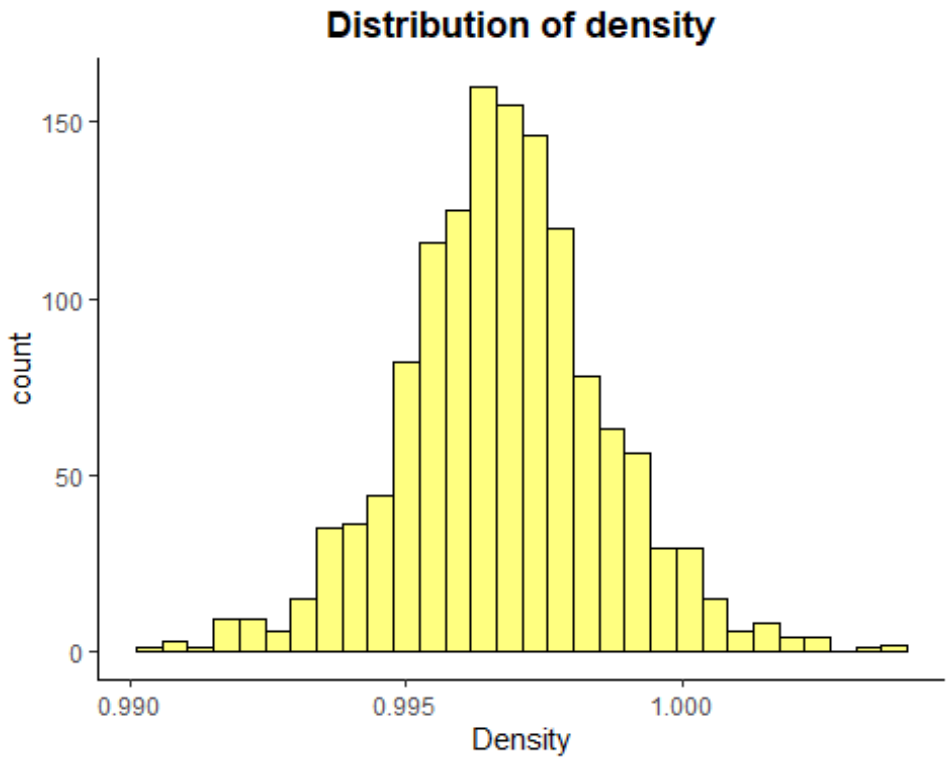

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



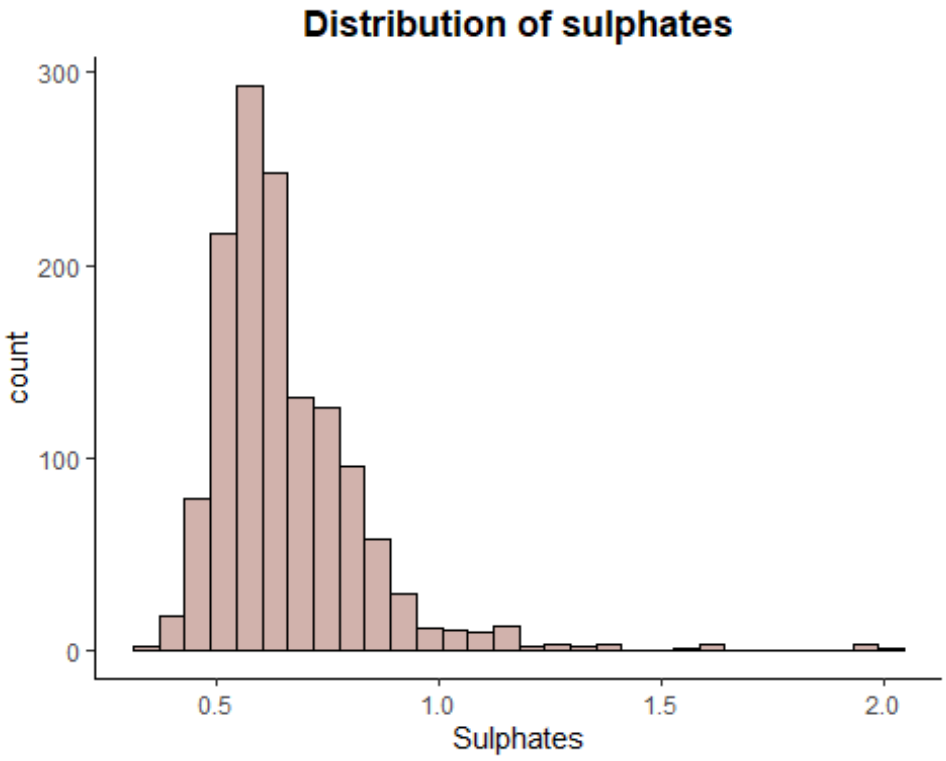

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

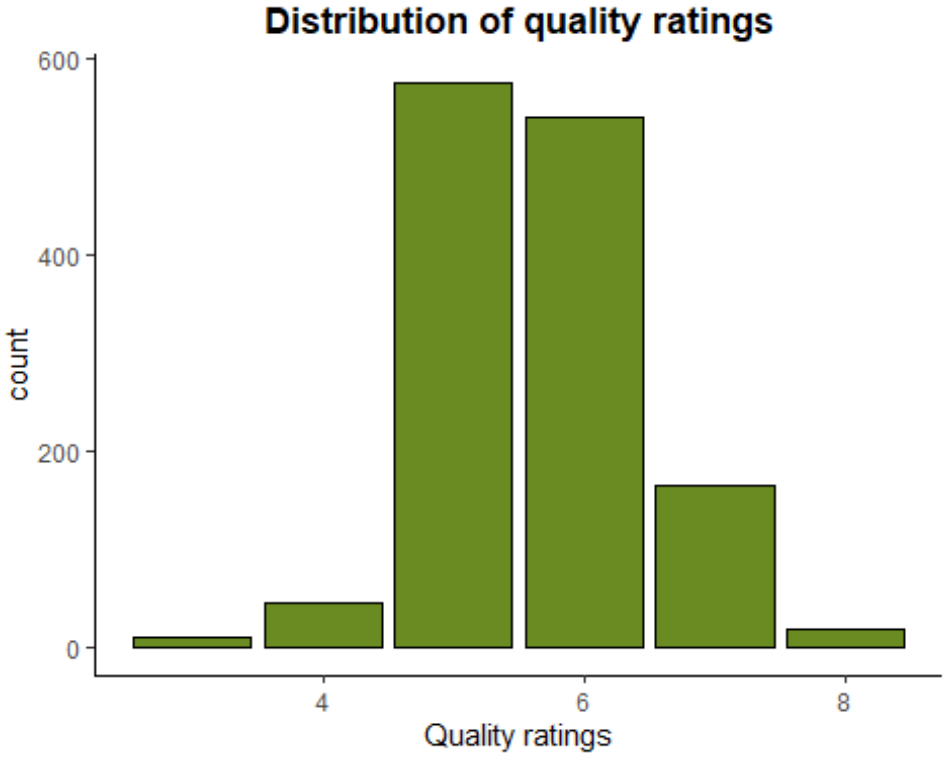



```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



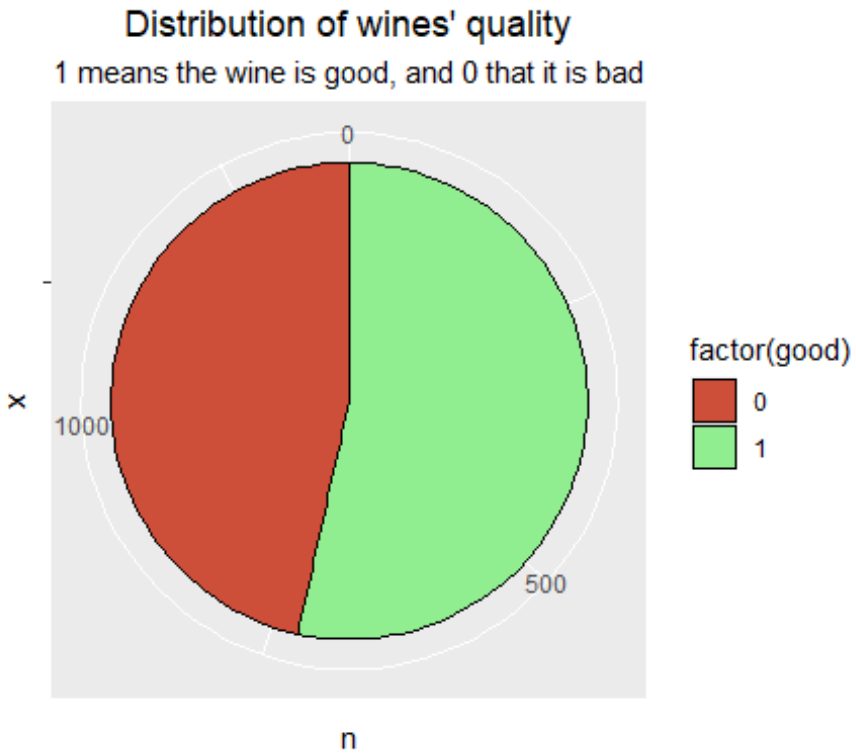

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```






```
## `summarise()` ungrouping output (override with `.groups` argument)
```

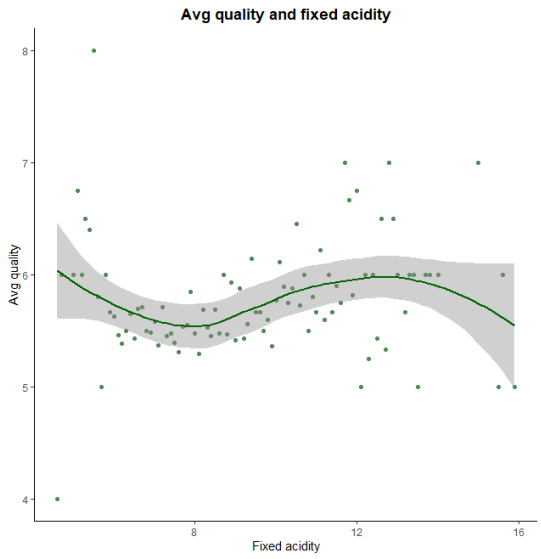
In the case of the variable “good”, since it is a dichotomous variable its distribution is explored using a pie chart.

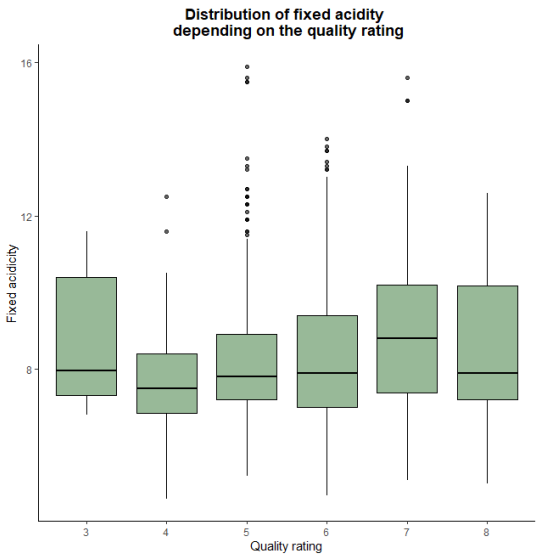


According to the previous histograms, the variables chlorides, pH and density appear to follow a normal distribution. Additionally, the distribution of the variables volatile, residual sugar, free sulfur dioxide and sulphates appear to be positively skewed. Regarding the wines’ quality raitings, according to the distribution of the variable “quality”, the most popular ratings are 5 and 6. Furthermore, more than 50% of the wines are classified as good. This is, more than half of the wines have a rating equal or superior to 6.

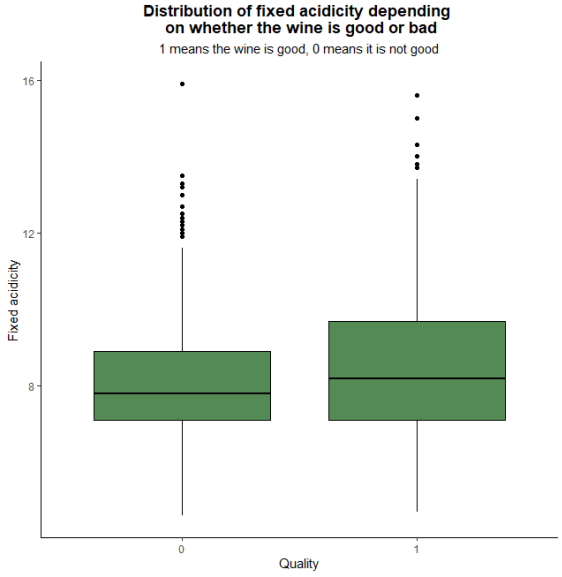
In the next section an exploratory analysis of the relationship between the wine’s quality and each feature is conducted. To conduct this analysis 4 graphs are used for each feature: the average quality rating depending on the feature’s value, a boxplot showing how the values of the feature are distributed depending on the quality rating, a boxplot showing how the values of the feature are distributed depending on whether the wine is good or bad (“good” variable) and a plot showing how the histogram of the feature changes depending on whether the wine is good or bad:

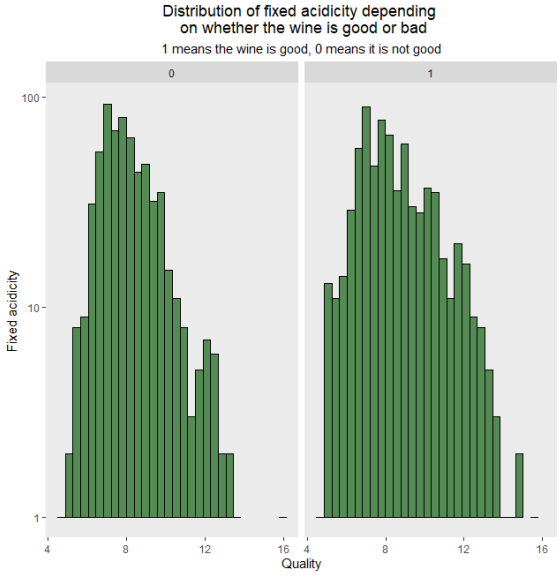
```
## `summarise()` ungrouping output (override with `.groups` argument)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```





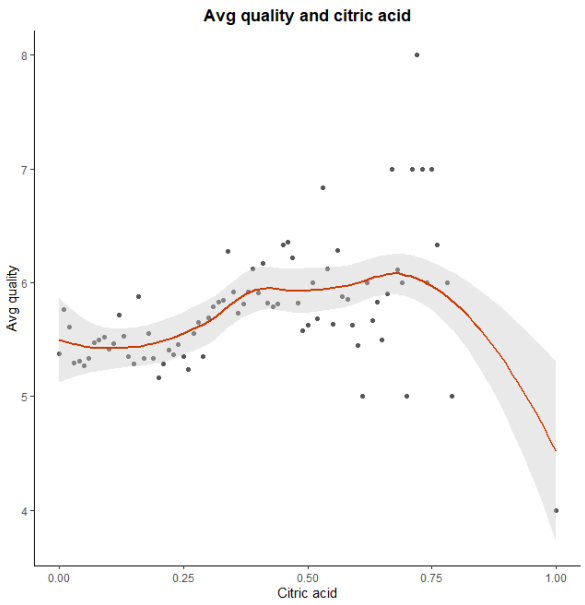

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 7 rows containing missing values (geom_bar).
```

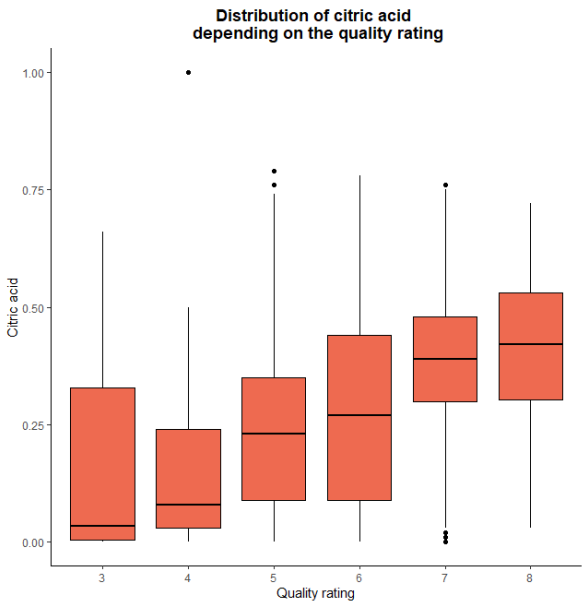




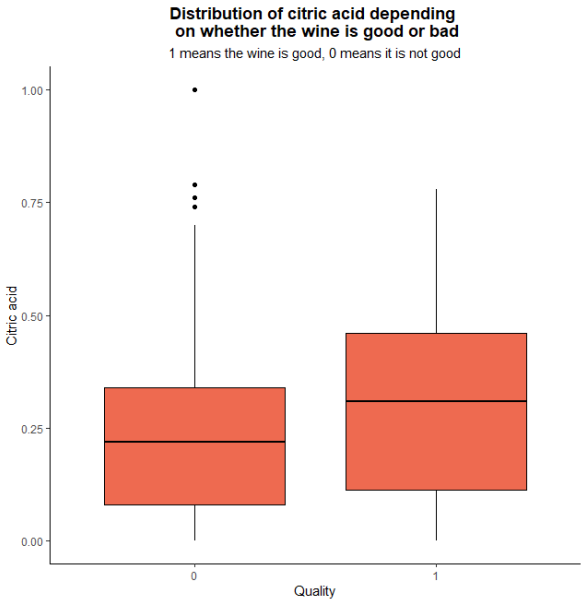
According to the graphs, there isn't a clear relationship between "fixed acidity" and the wine's quality. According to the boxplot of "fixed acidity" and "good", the wines classified as good seem to have slightly higher values of fixed acidity.

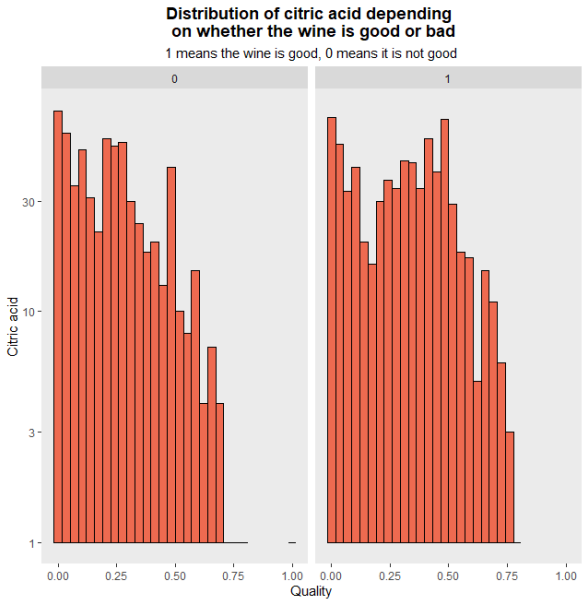
```
## `summarise()` ungrouping output (override with `.groups` argument)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```





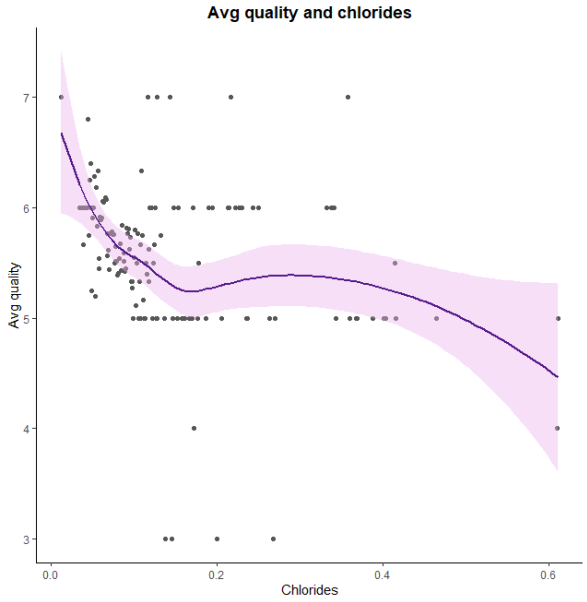

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 11 rows containing missing values (geom_bar).
```

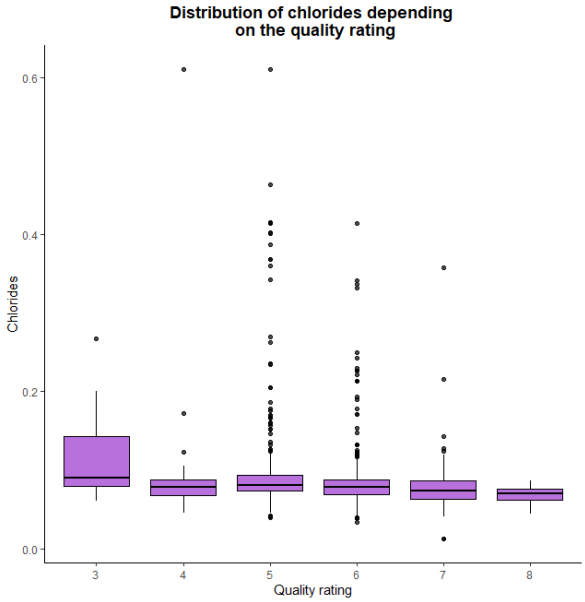




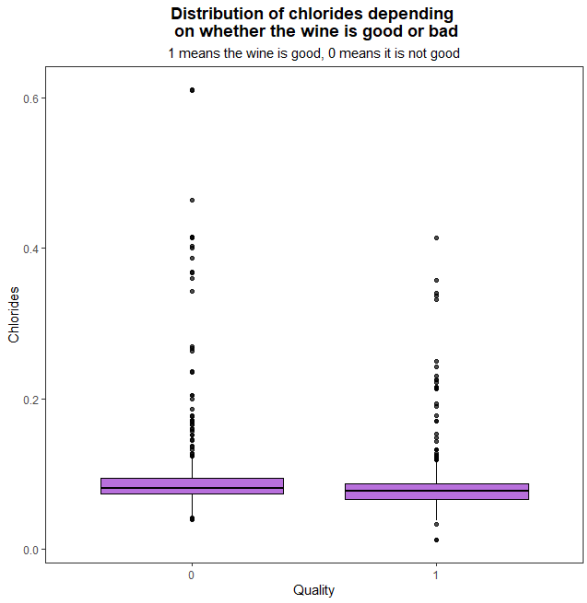
Citric acid appears to be a positive relationship with the wine's quality. The boxplot between "citric acid" and "quality" shows that for higher quality values the citric acid is distributed among higher values. Likewise, the boxplot between "citric acid" and "good" shows that the citric acid values are distributed among higher values for good wines.

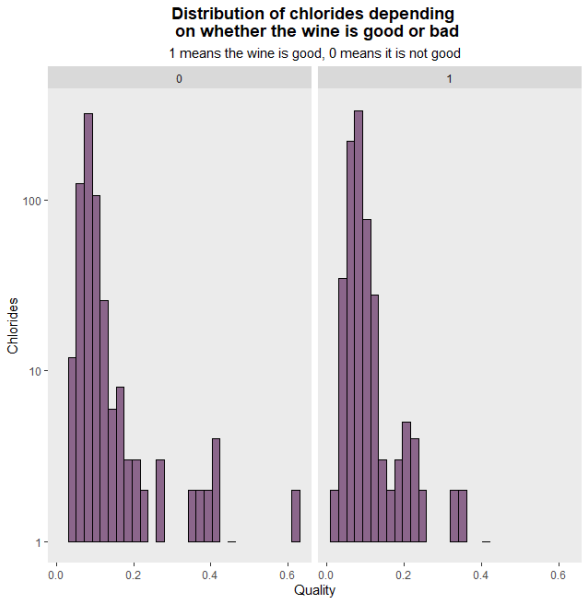
```
## `summarise()` ungrouping output (override with `.groups` argument)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```





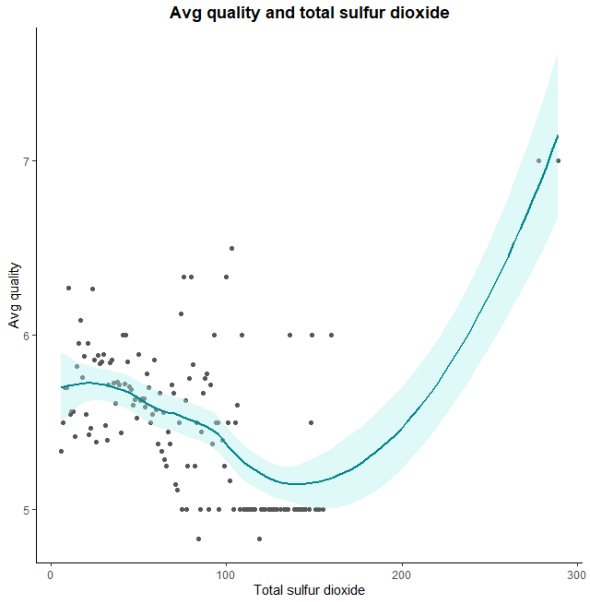

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 28 rows containing missing values (geom_bar).
```

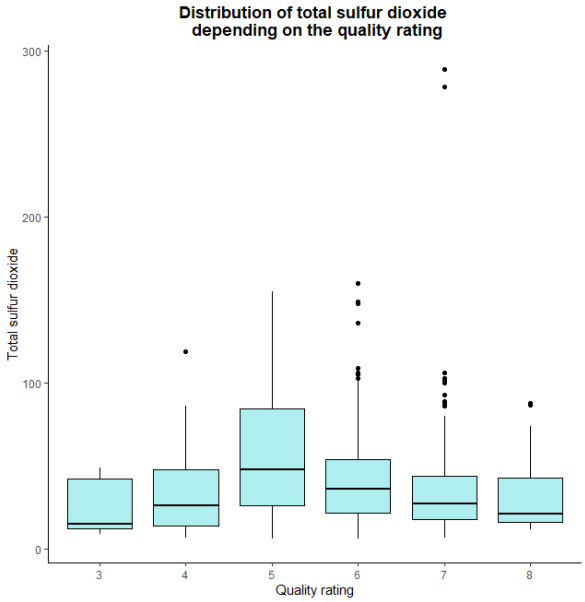




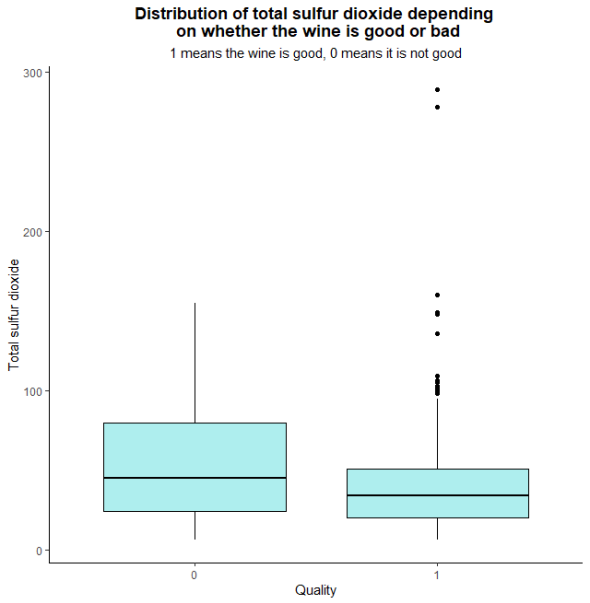
Regarding chlorides, it is not clear from the graphs whether more or less chlorides lead to more or less quality. As it can be seen in the two last graphs, the distribution of “chlorides” doesn’t seem to be different between good and bad wines. Thus, it cannot be concluded from the graph that there is a relationship between quality and chlorides.

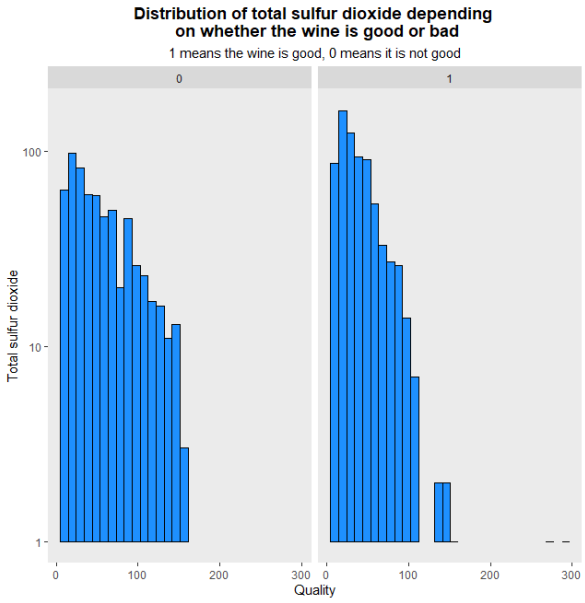
```
## `summarise()` ungrouping output (override with `.groups` argument)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```






```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 28 rows containing missing values (geom_bar).
```

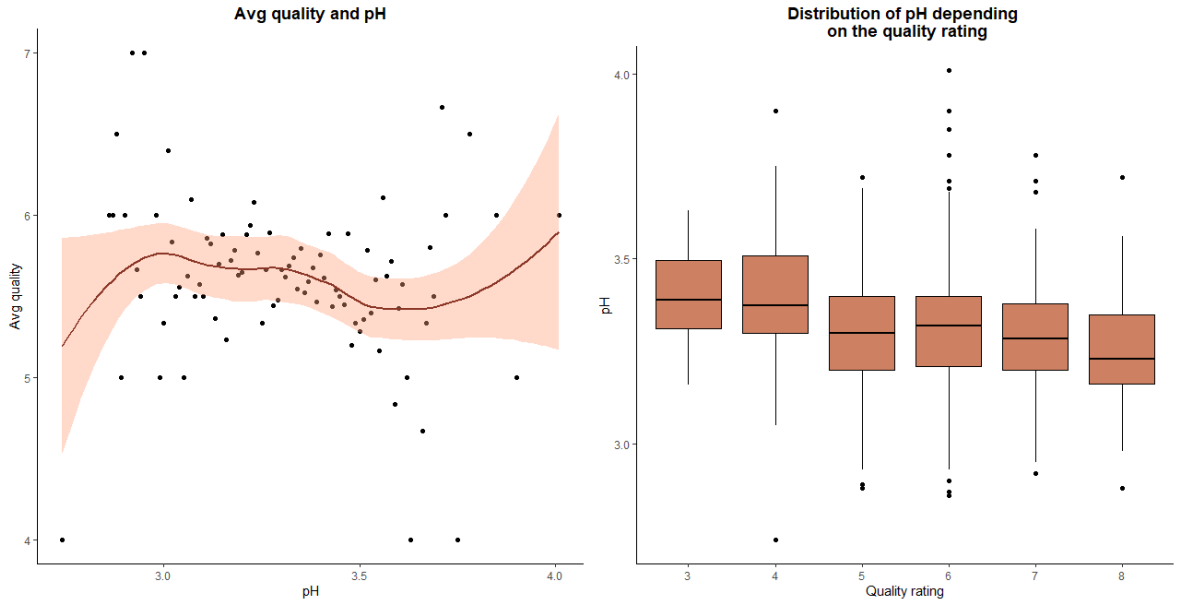




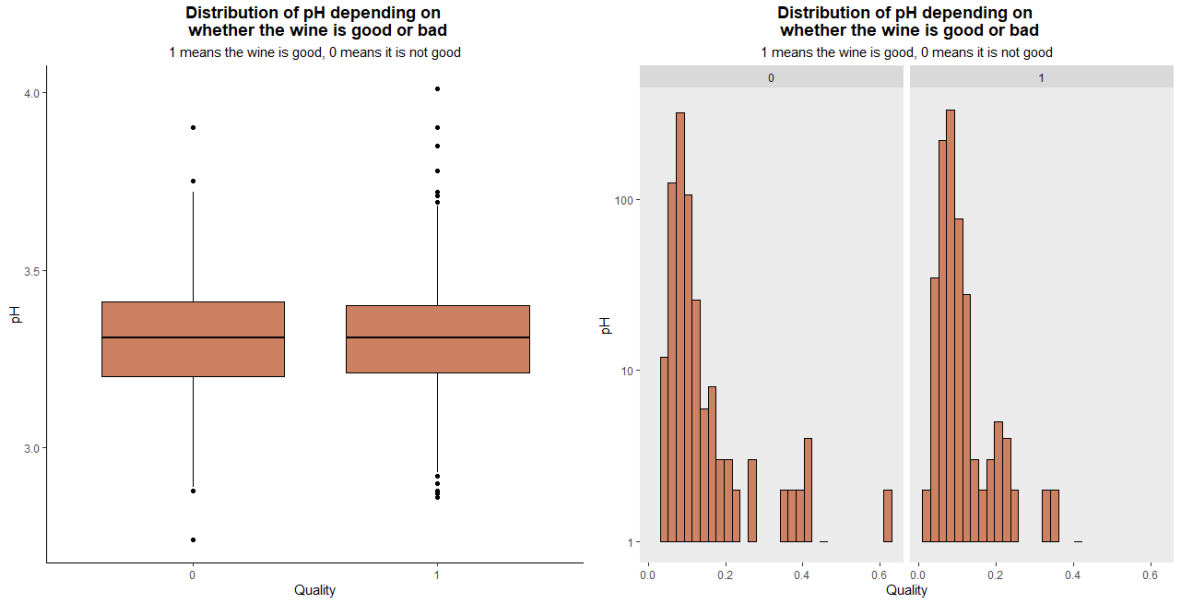
According to the graphs, total sulfur dioxide and quality seem to be negatively correlated: the boxplots show that wines classified as bad tend to have higher total sulfur dioxide values.

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



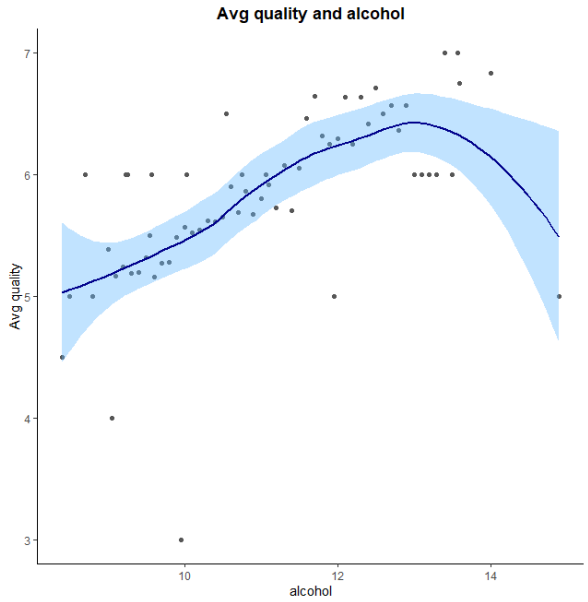

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 28 rows containing missing values (geom_bar).
```

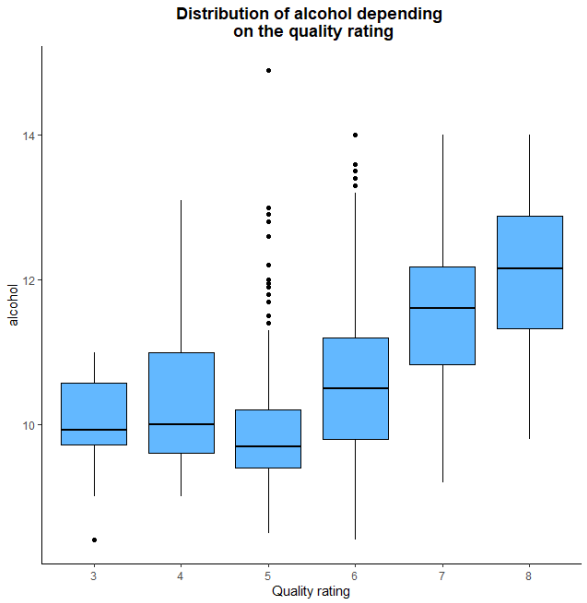


The graphs don't show a clear relationship between "quality" and "pH": the distribution of "pH" doesn't seem to differ between good and bad wines. Therefore, it seems that there isn't a relationship between these variables as "pH" doesn't seem to help explain the quality ratings.

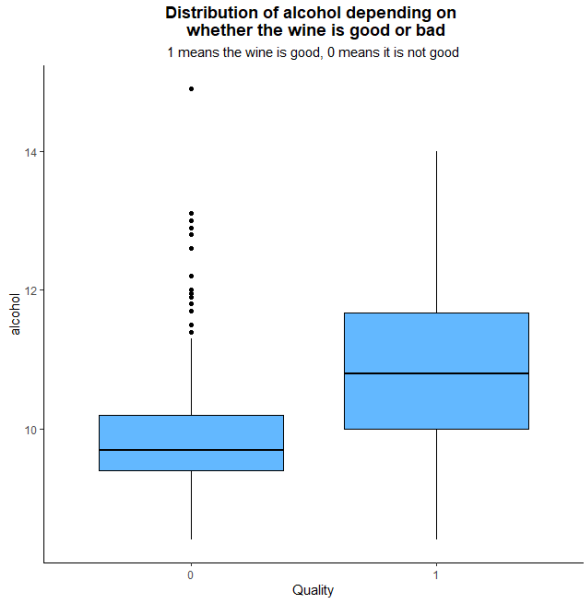
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

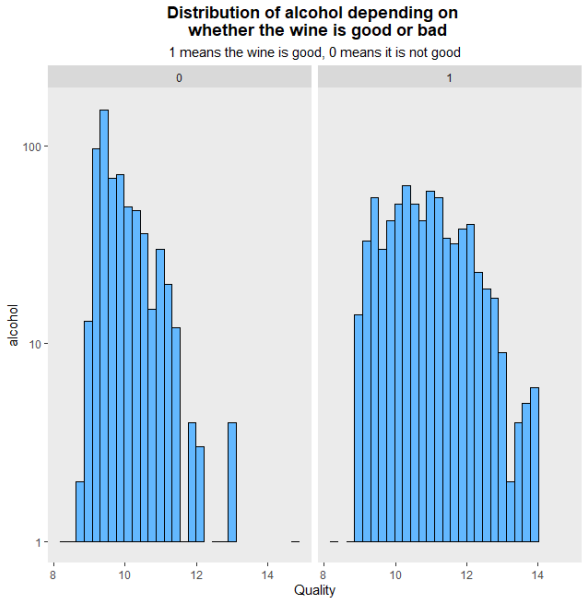
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```





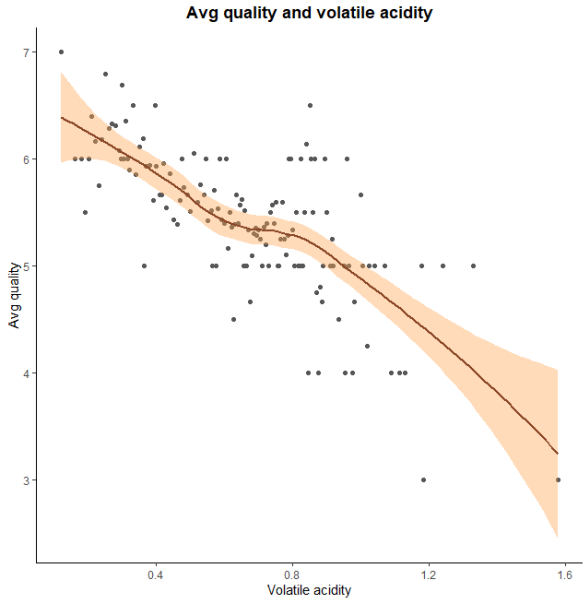

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 13 rows containing missing values (geom_bar).
```

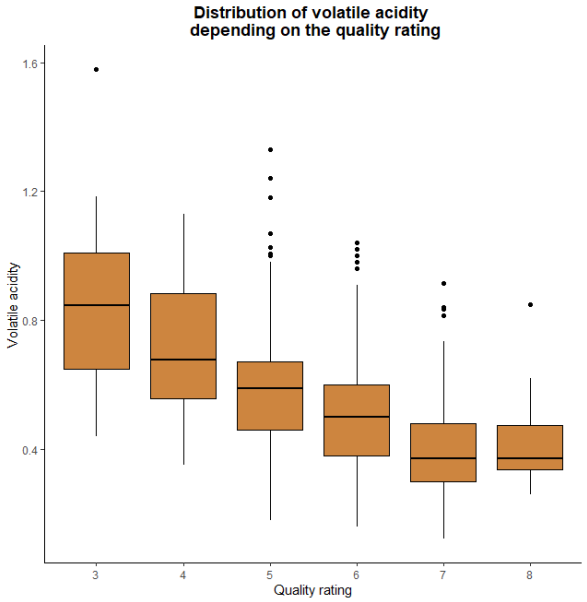




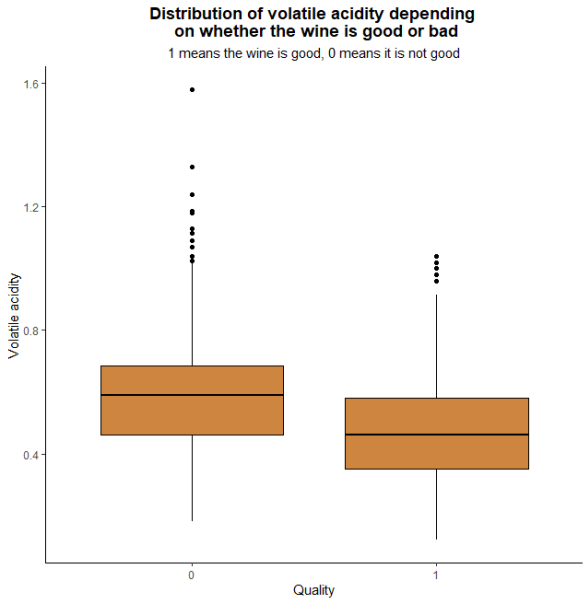
Alcohol and quality seem to be positively related: the boxplots show that the alcohol values are distributed among higher values for good wines than for bad wines, the histograms show that for good wines "rating" is distributed among higher values and the first graph shows that a higher level of alcohol tends to be associated with a higher average rating. Thus, this variable seems helpful to explain the wines' ratings.

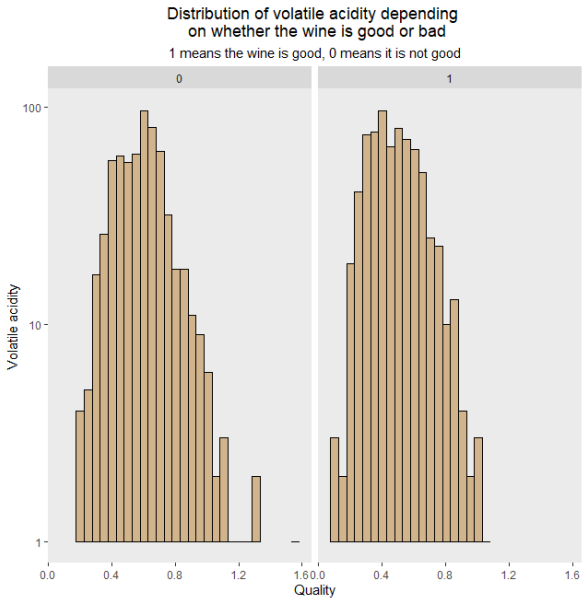
```
## `summarise()` ungrouping output (override with `.groups` argument)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```





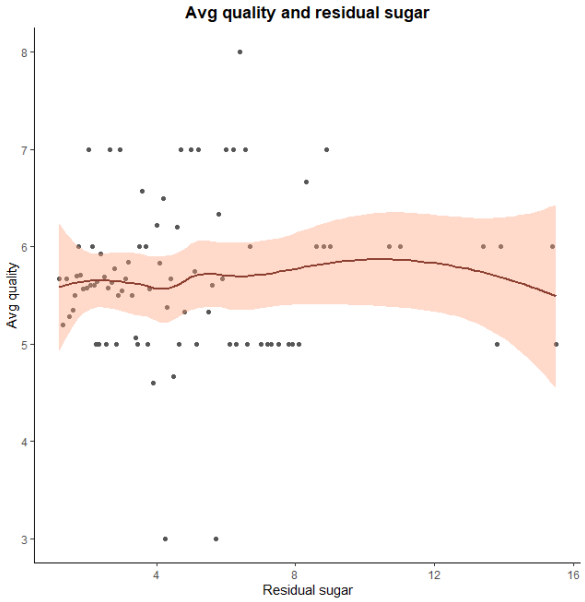

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 16 rows containing missing values (geom_bar).
```

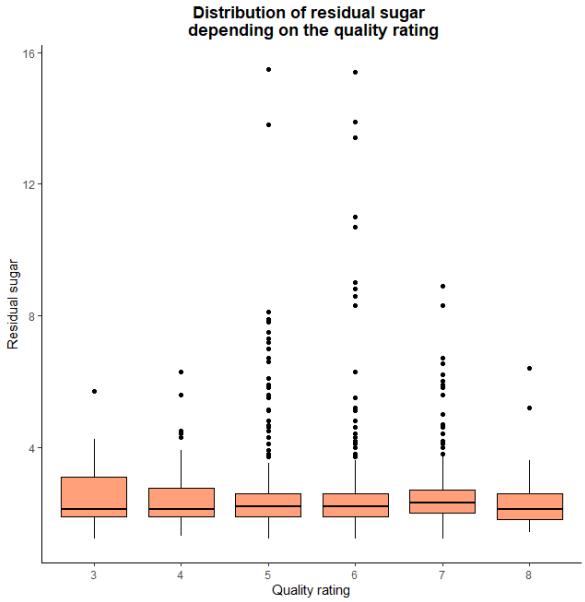




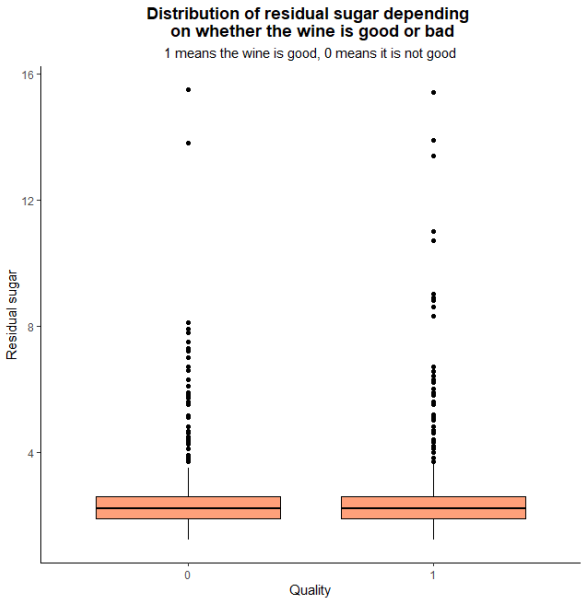
According to the graphs it can be concluded that volatile acidity and quality are negatively correlated: the first graphs shows that higher quality values tend to to have lower volatile acidity values, the second graph shows that for wines with higher volatile acidity "rating" is distributed among lower values and the last two graphs show that for good wines the variable "rating" is distributed among lower values.

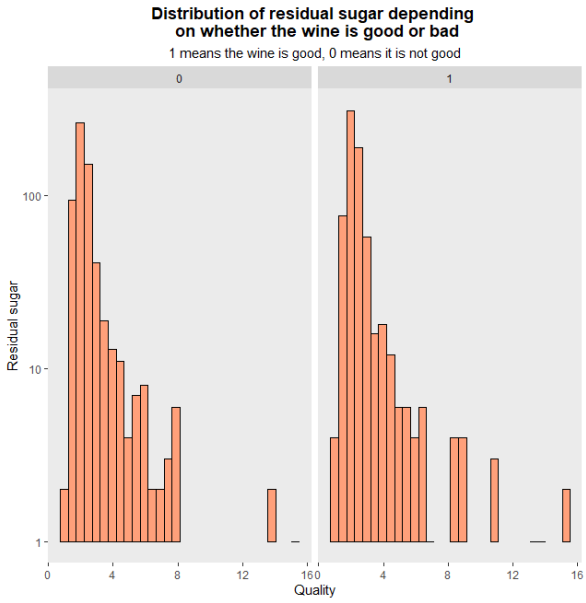
```
## `summarise()` ungrouping output (override with `.groups` argument)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```





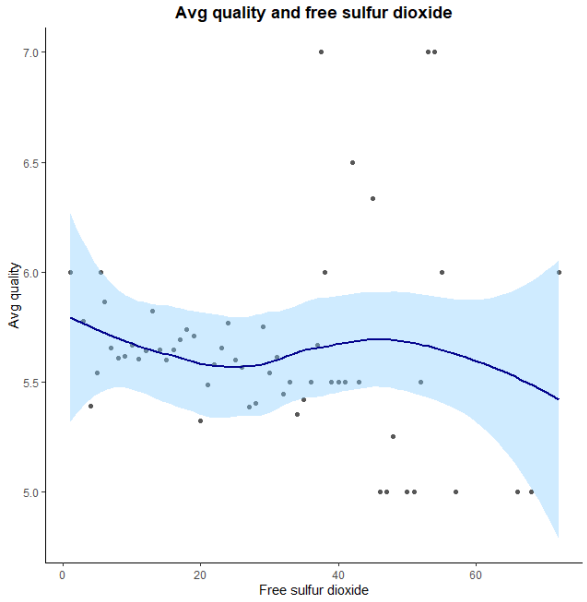

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 24 rows containing missing values (geom_bar).
```

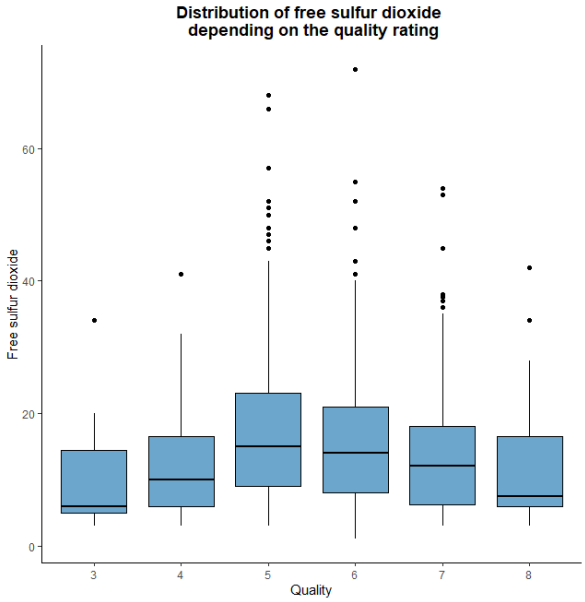




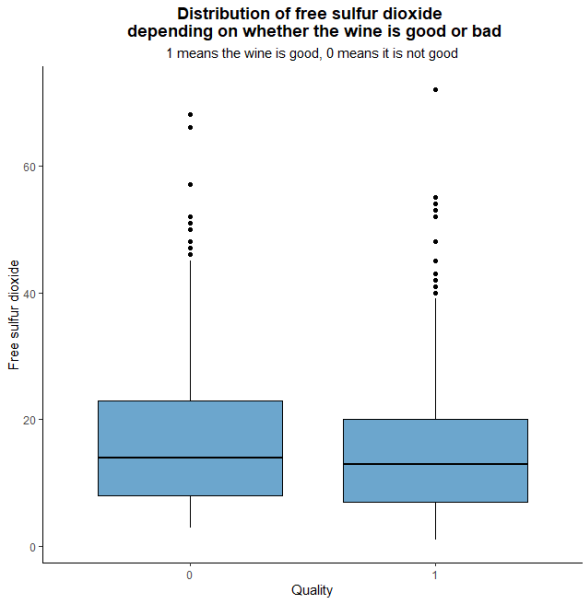
Regarding residual sugar, according to the graphs there isn't a relationship between this variable and "quality": higher levels of residual sugar don't seem to imply neither better nor worse ratings. Additionally, the distribution of "rating" doesn't seem to differ between good and bad wines.

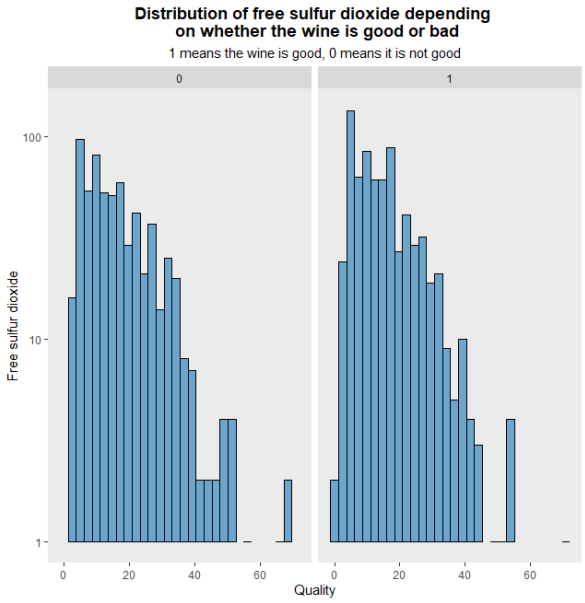
```
## `summarise()` ungrouping output (override with `.groups` argument)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```





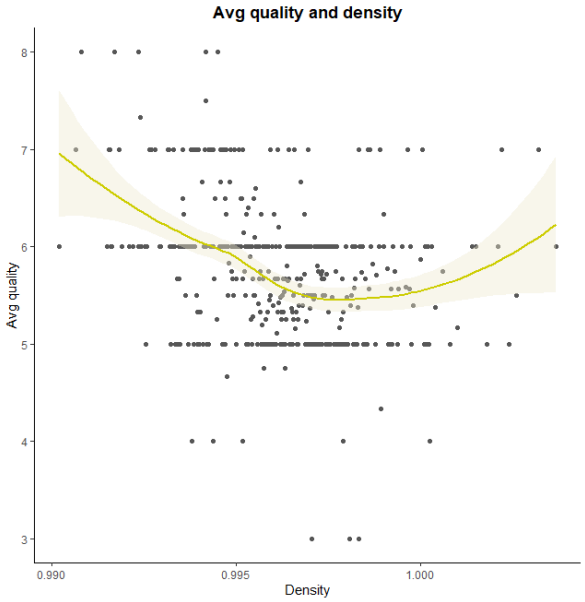

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 13 rows containing missing values (geom_bar).
```

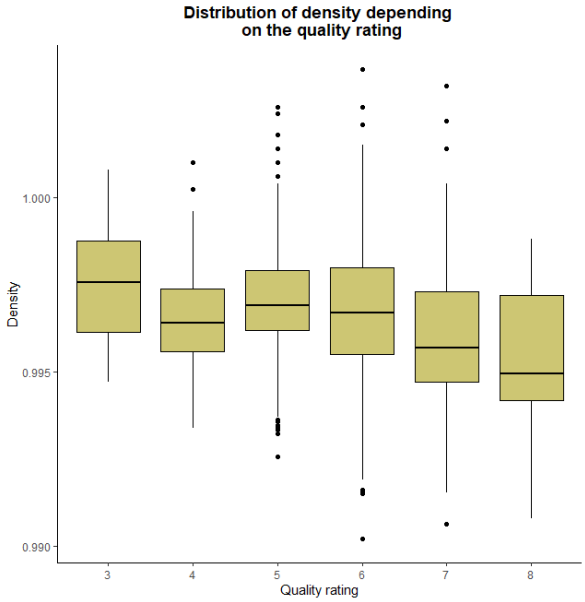




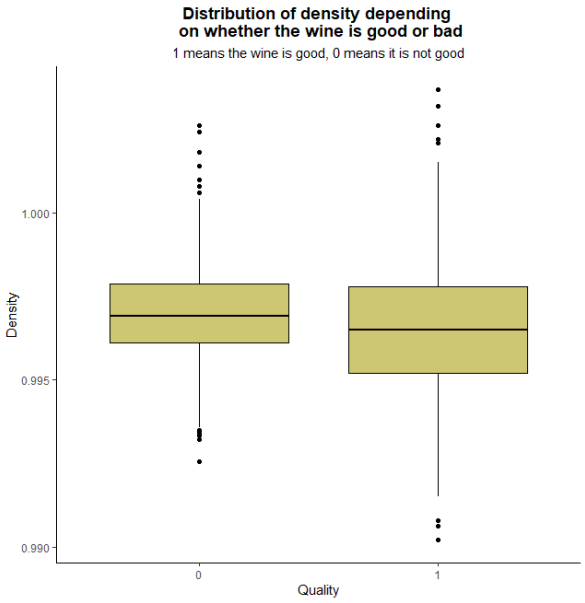
Similarly, the graphs don't suggest a relationship between "quality" and "free sulfur dioxide". The distribution of "free sulfur dioxide" doesn't differ much between good and bad wines and higher levels of free sulfur dioxide aren't associated to neither better nor worse ratings.

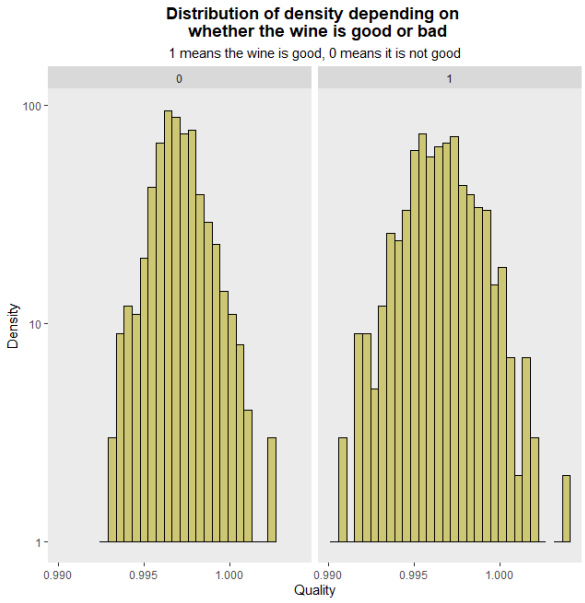
```
## `summarise()` ungrouping output (override with `.groups` argument)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```





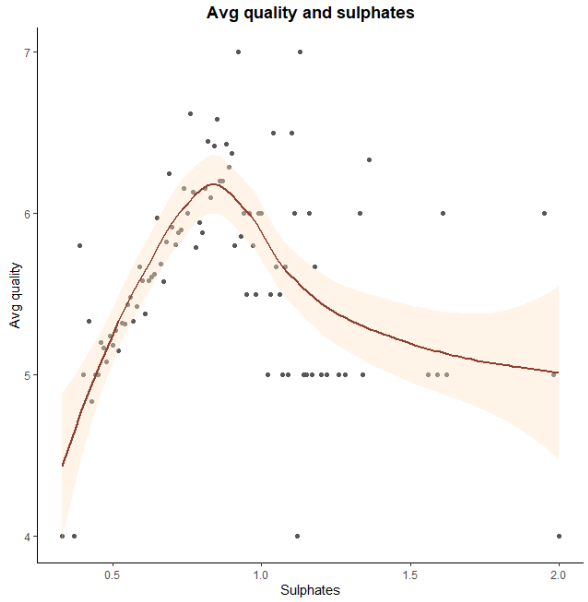

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 9 rows containing missing values (geom_bar).
```

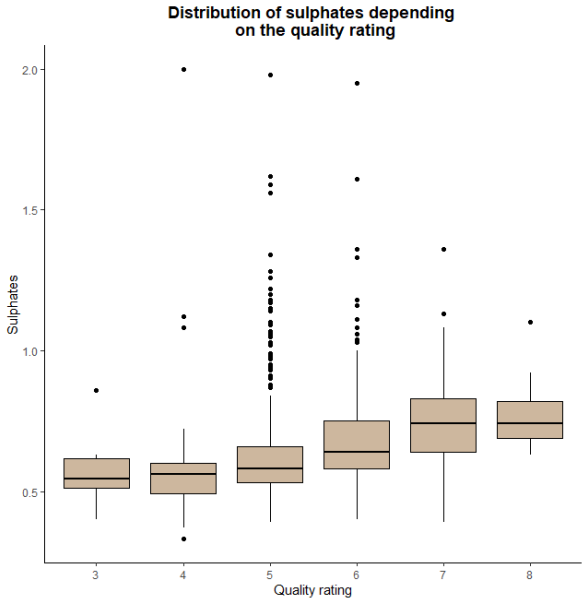




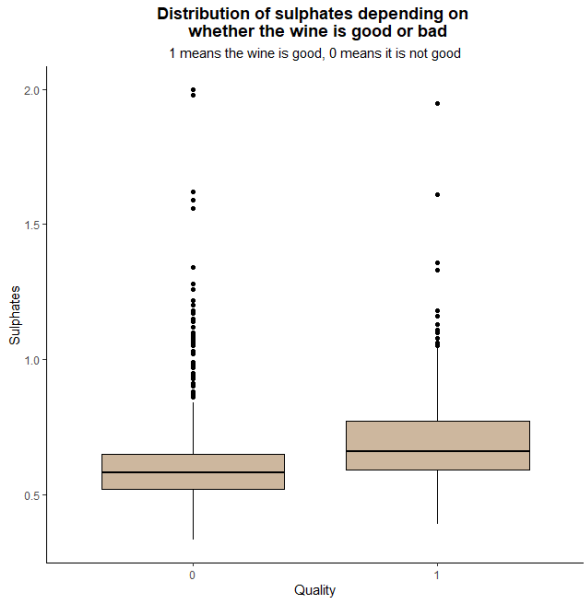
According to the boxplots, for better ratings the variable “density” is distributed among lower values. However, the first and the last plots don’t suggest that these variables are correlated. However, in the following sections it is furtherly studied whether these variables are correlated or not.

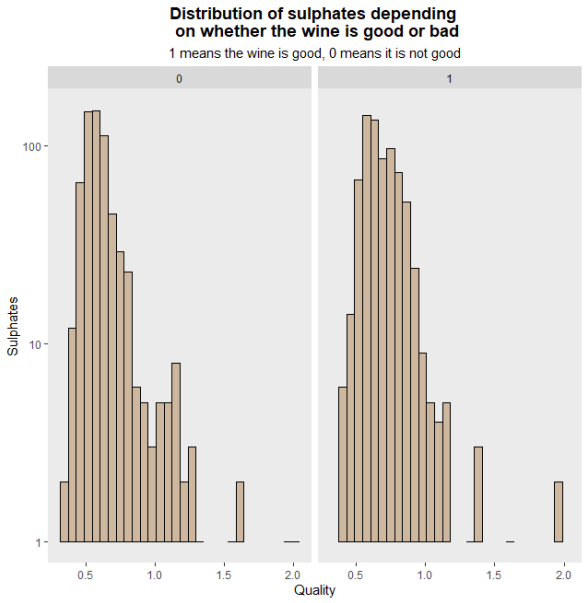
```
## `summarise()` ungrouping output (override with `.groups` argument)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```






```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 20 rows containing missing values (geom_bar).
```





Finally, according to the boxplots sulphates and quality appear to be positively correlated: the boxplots show that better wines tend to have higher values of sulphates and the first graph shows that higher level of sulphates are associated with higher average ratings -although this relationship seems to be reverted for values of "sulphates" above 1. Additionally, the last graph doesn't suggest that the distribution of "sulphates" differs for good and bad wines. Nonetheless,

this relationship is furtherly explored in the following sections in order to empirically determine if these variables are correlated.

To sum up, the following conclusions can be extracted from this section:

- The variables "fixed acidity", "chlorides", "pH", "residual sugar", "free sulfur dioxide" and "density" don't seem to be relevant variables to explain the wines' quality ratings. According to the graphs explored in this section these variables don't seem to have a relationship with "quality". Nonetheless, it is necessary to further study these variables in the following section before concluding whether they can be useful to predict a wine's quality rating or not.
- The rest of the variables seem to have a relationship with "quality", and "alcohol", "volatile acidity" and "sulphates" seem to be the variables that have the strongest relationship with "quality". This is, according to the graphs from this section these variables seem to be the most useful ones to predict a wine's quality rating. However, in this is case it is also necessary to conduct a deeper analysis of the relationship of these variables and "quality" before concluding how strong those relationships are.

2.3. Correlations

After conducting the exploratory data analysis, the relationships identified through this analysis are empirically studied. We compute the correlation to check if the variables that, according to the plots, appeared to be related are ineed related. Additionally, with the test of Pearson's correlation we test whether this correlation is statistically significant. The null hypothesis of this test is:

$$H_0 : \rho_{x,y} = 0$$

$$H_1 : \rho_{x,y} \neq 0$$

The following table summarizes the correlation coefficients obtained as well as the results from the test of Pearson's correlation:

Variables	Pearson.Correlation	Test.Results
Fixed Acidity	0.1365083	Reject Null Hypothesis
Citric Acid	0.2351280	Reject Null Hypothesis
Chlorides	-0.1349521	Reject Null Hypothesis
Total Sulfur Dioxide	-0.1722475	Reject Null Hypothesis
pH	-0.0655435	Reject Null Hypothesis
Alcohol	0.4897872	Reject Null Hypothesis
Volatile Acidity	-0.4013427	Reject Null Hypothesis
Residual Sugar	0.0312401	Not Reject Null Hypothesis
Free Sulfur Dioxide	-0.0484563	Not Reject Null Hypothesis
Density	-0.1718335	Reject Null Hypothesis
Sulphates	0.2465831	Reject Null Hypothesis

Most of the results obtained are consistent with the insights obtained from the exploratory data analysis:

- In the previous section it was concluded that "alcohol", "volatile acidity" and "sulphates" had a strong relationship with "quality", and as it can be seen in the table these variables have the highest correlations -in absolute values- with "quality". Furthermore, for the 3 variables it can be concluded that their correlation with "quality" is statistically significant. Additionally, the sign of this correlation is also consistent with the previous section: "alcohol" and "sulphates" seemed to be positively correlated with "rating" and the correlation is indeed positive whereas "volatile acidity" seemed to be negatively correlated with "quality" and the correlation obtained is negative.

- For most of the variables that seemed to not have a relationship with "quality" ("fixed acidity", "chlorides", "pH", "residual sugar" and "free sulfur dioxide"), the correlation coefficient obtained is small. Furthermore, in the case of "residual sugar" and "free sulfur dioxide" it cannot be concluded that the correlation is statistically significant, suggesting that these variables don't have a relationship with "quality". For the other variables, although it can be concluded that the correlation coefficient obtained is statistically significant, this coefficient is small. This is, they have a relationship with "quality", but it is weak.

- In the case of "density", in the previous section it seemed that this variable didn't have a relationship with "quality". However, the correlation coefficient obtained is quite high in comparison to the other variables that also seemed to be uncorrelated with "quality". Moreover, according to the results of the Pearson's correlation test we can conclude that this correlation is statistically significant. Although this result may not seem consistent with the results from the previous section, it is necessary to take into account that for this variable the graphs suggested different things: while the boxplots suggested that for better ratings this variable is distributed among lower values the last plot didn't show different "density" distributions between good and bad wines. Thus, whereas the plots didn't show a clear relationship between this variable and "quality", after conducting a deeper analysis we can conclude that they have a negative relationship.

3. Model Development

Once we have explored how each variable is related to the quality of the wine, we construct the prediction model. The aim of the model is to predict the wine's quality given its characteristics, including those features that have a relationship with the wine's quality and thus can help us predict it. As the "validation" dataset will only be used to test the final selected model, the "wine" dataset is splitted into "train_set" -which contains 90% of the "wine" dataset- and "test_set" -which includes 10% of the "wine" dataset. "Train_set" will be used to train the different models, and each of them will be tested using "test_set".

3.1. Generalized Linear Models (GLM)

We start by fitting a Generalized Linear Model -GLM- including as predictors only the two variables that have the strongest relationship with the target variable –"quality". Thus, the predictors are "alcohol" and "volatile acidity".

```
##GLM
#start with a simple model: the variables with the highest correlation
fit_glm1 <- glm(quality~alcohol + volatile.acidity, data= train_set)
pred_glm1 <- predict.glm(fit_glm1, test_set)
pred_glm_21 <- predict(fit_glm1, train_set)
RMSE_glm_test1 <- sqrt(mean((pred_glm1-test_set$quality)^2))
RMSE_glm1 <- sqrt(mean((pred_glm_21 - train_set$quality)^2))

RMSE_glm_test1

## [1] 0.7021208
```

Although it seems like a good RMSE, there are other variables that, although they don't have such a strong relationship with "quality", they can also help explain a wine's quality because they have a relationship with this variable. Thus, we subsequently add more variables, including first the ones that have a stronger relationship with "quality". The code of the subsequent models is not included so that the report is not too long, but it can be found in the script. The models include the following variables:

-GLM 2: alcohol + volatile acidity + citric acid + sulphates

-GLM 3: alcohol + volatile acidity + citric acid + sulphates + total sulfur dioxide + density

-GLM 4: alcohol + volatile acidity + citric acid + sulphates + total sulfur dioxide + density + fixed acidity + chlorides

-GLM 5: alcohol + volatile acidity + citric acid + sulphates + total sulfur dioxide + density + fixed acidity + chlorides + pH + free sulfur dioxide

-GLM 6: alcohol + volatile acidity + citric acid + sulphates + total sulfur dioxide + density + fixed acidity + chlorides + pH + free sulfur dioxide + residual sugar

Models	RMSE.on.train.set	RMSE.on.test.set
GLM 1	0.6631834	0.7021208
GLM 2	0.6550826	0.6893489
GLM 3	0.6515777	0.6914678
GLM 4	0.6443952	0.6888163
GLM 5	0.6420485	0.6866897
GLM 6	0.6415740	0.6866264

Variables that don't have a relationship with "quality"- "residual sugar" and "free sulfur dioxide- were included in the previous models. However, according to the results from the previous section, for these variables the null hypothesis that the correlation with "quality" is 0 could not be rejected. Thus, these variables don't seem to be helpful in order to predict a wine's quality rating. Therefore, a model excluding this variable is fitted:

```
#remove residual sugar and free sulfur dioxide
fit_glm7 <- glm(quality~alcohol + volatile.acidity + citric.acid +
                  sulphates + total.sulfur.dioxide + density +
                  fixed.acidity + chlorides + pH,
                , data=train_set)
pred_glm7 <- predict.glm(fit_glm7, test_set)
pred_glm_27 <- predict(fit_glm7, train_set)
RMSE_glm_test7 <- sqrt(mean((pred_glm7-test_set$quality)^2))
RMSE_glm7 <- sqrt(mean((pred_glm_27 - train_set$quality)^2))
RMSE_glm_test7

## [1] 0.6818731
```

-GLM 7: alcohol + volatile acidity + citric acid + sulphates + total sulfur dioxide + density + fixed acidity + chlorides + pH

Models	RMSE.on.train.set	RMSE.on.test.set
GLM 1	0.6631834	0.7021208
GLM 2	0.6550826	0.6893489
GLM 3	0.6515777	0.6914678
GLM 4	0.6443952	0.6888163
GLM 5	0.6420485	0.6866897
GLM 6	0.6415740	0.6866264
GLM 7	0.6434701	0.6818731

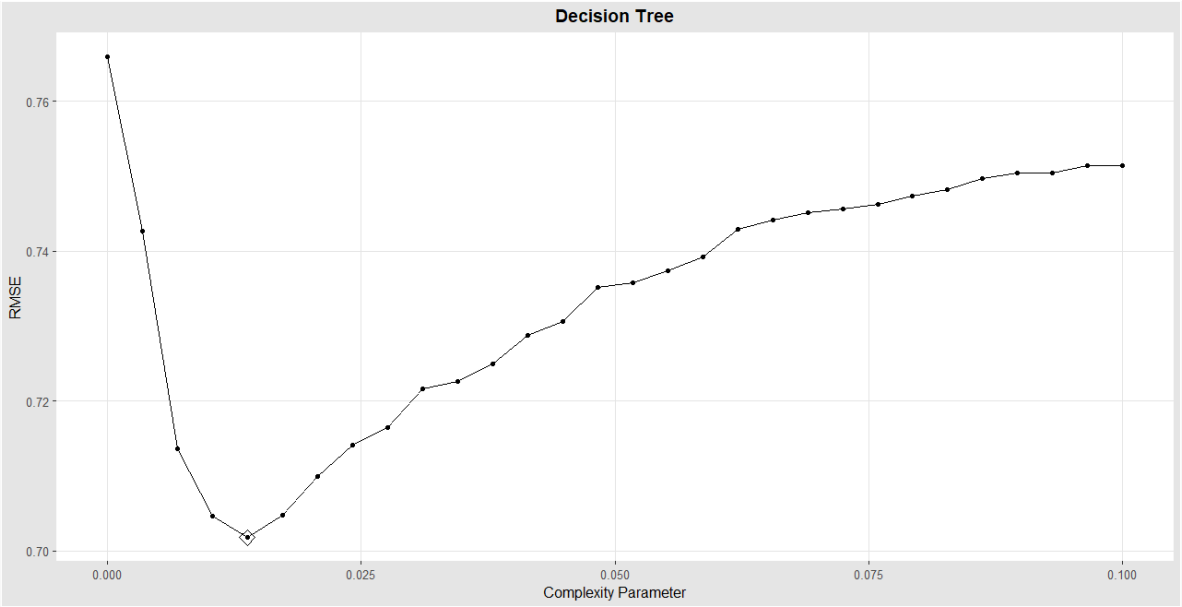
As it can be seen in the previous table, excluding the variables that didn't have a relationship with "quality", and thus they did not help explain a wine's quality rating, leads to a significant improvement in the RMSE computed using the "test_set" dataset. Thus, those variables were overfitting the model and leading to worse results. To sum up, according to the RMSE computed in the "test_set" dataset the best model is the GLM 7.

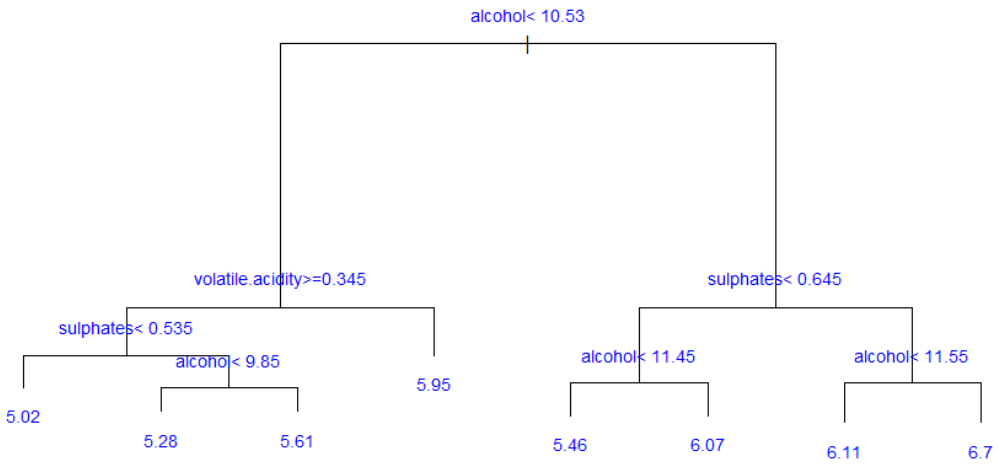
3.2. Decision Tree (CART)

Parting from the best model from the previous section -GLM 7- a decision tree is fitted. This is, a decision tree is fitted using the same predictors as in the GLM 7: all the variables except for "residual sugar" and "free sulfur dioxide".

```
fit_rpart <- train(quality~alcohol + volatile.acidity + citric.acid +
                  sulphates + total.sulfur.dioxide + density +
                  fixed.acidity + chlorides +pH,
                   method = "rpart", tuneGrid=data.frame(cp=seq(0,0.1,len=30)),
                   data = train_set)
predict_rpart<- predict(fit_rpart, test_set, norm.votes=T, predict.all=F, proxim
ity=F, nodes=F)
predict_rpart2 <- predict(fit_rpart, train_set, norm.votes=T, predict.all=F, pro
ximity=F, nodes=F)
```

```
RMSE_rpart_test <- sqrt(mean((predict_rpart-test_set$quality)^2))
RMSE_rpart <-  sqrt(mean((predict_rpart2-train_set$quality)^2))
RMSE_rpart_test

## [1] 0.7345924

fit_rpart

## CART
##
## 1220 samples
##    9 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 1220, 1220, 1220, 1220, 1220, 1220, ...
## Resampling results across tuning parameters:
##
##   cp           RMSE       Rsquared   MAE
##   0.000000000  0.7658729  0.2441765  0.5597718
##   0.003448276  0.7427194  0.2564131  0.5437488
##   0.006896552  0.7136006  0.2741391  0.5359907
##   0.010344828  0.7046337  0.2752141  0.5400733
##   0.013793103  0.7017546  0.2741809  0.5424565
##   0.017241379  0.7047728  0.2647987  0.5482079
##   0.020689655  0.7099766  0.2530835  0.5554577
##   0.024137931  0.7141132  0.2424151  0.5618704
##   0.027586207  0.7164741  0.2372950  0.5655218
##   0.031034483  0.7216066  0.2267658  0.5742097
##   0.034482759  0.7226071  0.2233494  0.5778178
##   0.037931034  0.7249583  0.2178060  0.5836957
##   0.041379310  0.7287561  0.2093438  0.5914940
##   0.044827586  0.7306612  0.2052697  0.5917671
##   0.048275862  0.7351017  0.1953015  0.5953818
##   0.051724138  0.7358014  0.1935883  0.5960662
##   0.055172414  0.7374305  0.1904631  0.5977655
##   0.058620690  0.7391630  0.1867106  0.5990260
##   0.062068966  0.7429446  0.1770659  0.6030937
##   0.065517241  0.7441260  0.1741418  0.6022883
##   0.068965517  0.7451799  0.1710582  0.6029477
##   0.072413793  0.7455786  0.1701249  0.6024845
##   0.075862069  0.7462888  0.1686439  0.6016789
##   0.079310345  0.7473108  0.1659111  0.6039100
##   0.082758621  0.7481532  0.1642961  0.6056642
##   0.086206897  0.7496729  0.1609686  0.6091182
##   0.089655172  0.7504317  0.1589810  0.6104760
##   0.093103448  0.7504317  0.1589810  0.6104760
##   0.096551724  0.7514049  0.1564868  0.6121768
##   0.100000000  0.7514049  0.1564868  0.6121768
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.0137931.
```



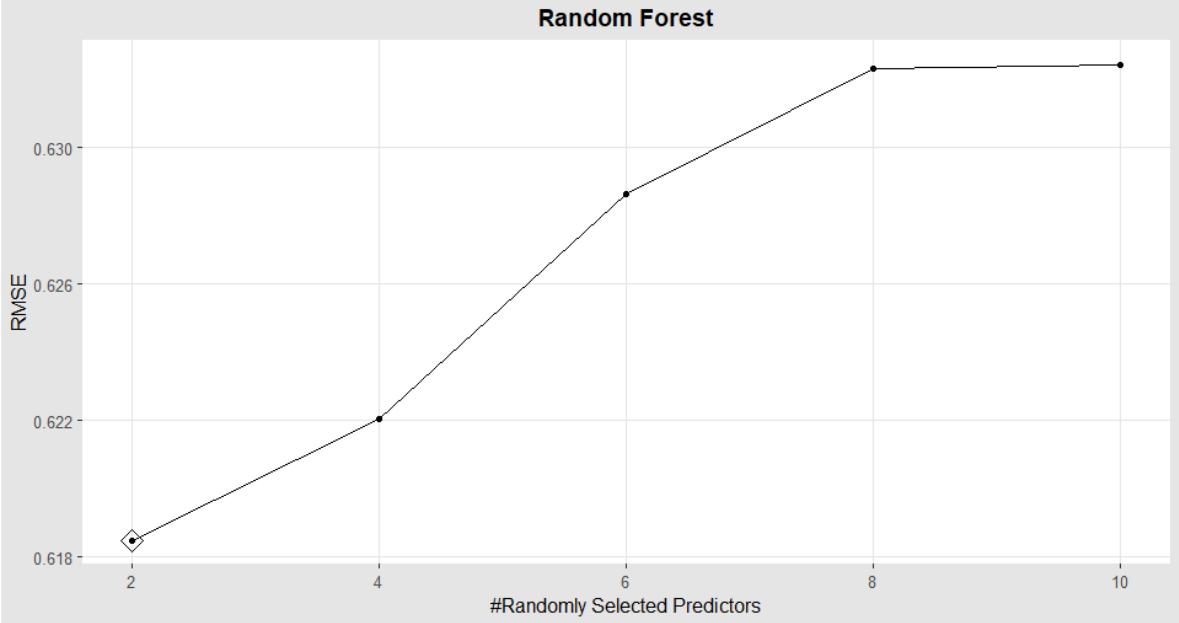


3.3. Random Forest

In this section a Random Forest model is fitted. To fit this model all the predictors used for the GLM 7 are included -this is, all variables except for "residual sugar" and "free sulfur dioxide". First of all, we find the optimal number of predictors randomly selected in each split:

```
#first we find the optimal number of predictors randomly selected in each split
tunegrid <- expand.grid(.mtry = seq(2,10,2))
rf_optim <- train(quality~alcohol + volatile.acidity + citric.acid +
                  sulphates + total.sulfur.dioxide + density +
                  fixed.acidity + chlorides +pH,
                       data= train_set,ntree=80, method="rf", tuneGrid= tunegrid
)
```

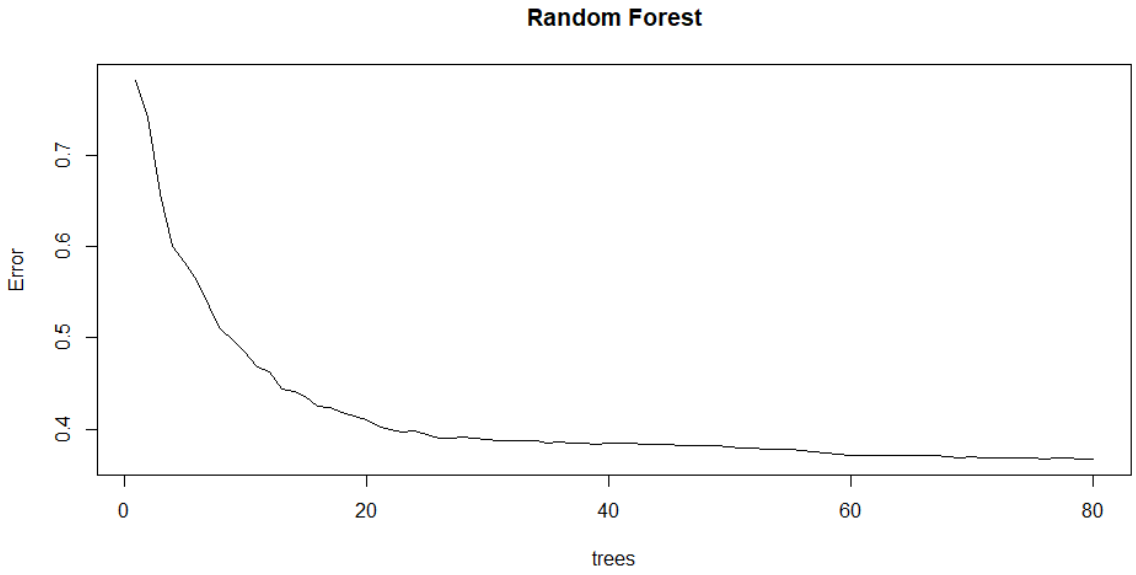
```
ggplot(rf_optim, highlight = T) + theme_igray() + labs(X="Randoly Selected Predictos",
                                                       y = "RMSE", title="Random Forest") +
  theme(plot.title = element_text(hjust=0.5,face="bold"))
```



Once the optimal number of predictors to be randomly selected out of the available predictors -the variables included in the model- in each split is computed, we fit the Random Forest model specifying that number -which is 2: in each split 2 variables out of the variables included will be randomly selected to predict the wine's rating. Furthermore, the model is fit with 80 trees:

```
mtry_value <- rf_optim$bestTune$mtry
fit_rf <- randomForest(quality~alcohol + volatile.acidity + citric.acid +
                  sulphates + total.sulfur.dioxide + density +
                  fixed.acidity + chlorides +pH,
                       data= train_set,ntree=80, mtry= mtry_value)
predict_rf <- predict(fit_rf, test_set)
predict_rf_2 <- predict(fit_rf, train_set)
RMSE_rf_test <- sqrt(mean((predict_rf-test_set$quality)^2))
RMSE_rf <- sqrt(mean((predict_rf_2-train_set$quality)^2))
RMSE_rf_test

## [1] 0.6195952

fit_rf

##
## Call:
##  randomForest(formula = quality ~ alcohol + volatile.acidity +      citric.ac
id + sulphates + total.sulfur.dioxide + density +      fixed.acidity + chlorides
+ pH, data = train_set, ntree = 80,      mtry = mtry_value)
##                Type of random forest: regression
##                      Number of trees: 80
## No. of variables tried at each split: 2
##
```

```
##           Mean of squared residuals: 0.3667309
##                     % Var explained: 44.06

plot(fit_rf,main="Random Forest")
```



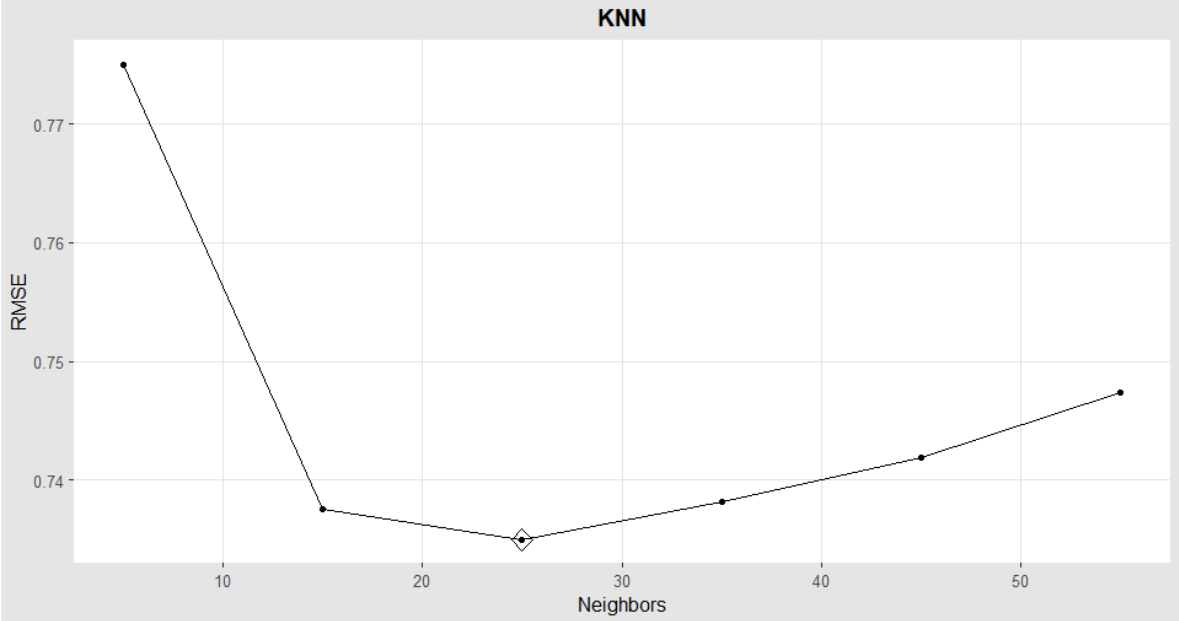
3.4. KNN

In this section a K-Nearest Neighbor Model is fitted, and to find the optimal K we try with different values: 5, 15, 25, 35, 45 and 55. As it can be seen in the results below, the best result is obtained when using K = 25.

```
#Knn
fit_knn <- train(quality~alcohol + volatile.acidity + citric.acid +
                   sulphates + total.sulfur.dioxide + density +
                   fixed.acidity + chlorides +pH,
                 method="knn", data=train_set,
                 tuneGrid = data.frame(k = seq(5,55,10)))
predict_knn <- predict(fit_knn, test_set)
predict_knn_2 <- predict(fit_knn, train_set)
RMSE_knn_test <- sqrt(mean((predict_knn-test_set$quality)^2))
RMSE_knn <- sqrt(mean((predict_knn_2-train_set$quality)^2))
RMSE_knn_test

## [1] 0.8194026

fit_knn

## k-Nearest Neighbors
##
## 1220 samples
##    9 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
```

```
## Summary of sample sizes: 1220, 1220, 1220, 1220, 1220, 1220, ...
## Resampling results across tuning parameters:
##
##   k   RMSE       Rsquared   MAE
##    5  0.7750179  0.1533733  0.5801109
##   15  0.7375092  0.1616846  0.5733693
##   25  0.7349631  0.1578473  0.5762071
##   35  0.7382064  0.1498495  0.5817276
##   45  0.7418539  0.1422688  0.5871085
##   55  0.7473905  0.1296883  0.5942566
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 25.
```

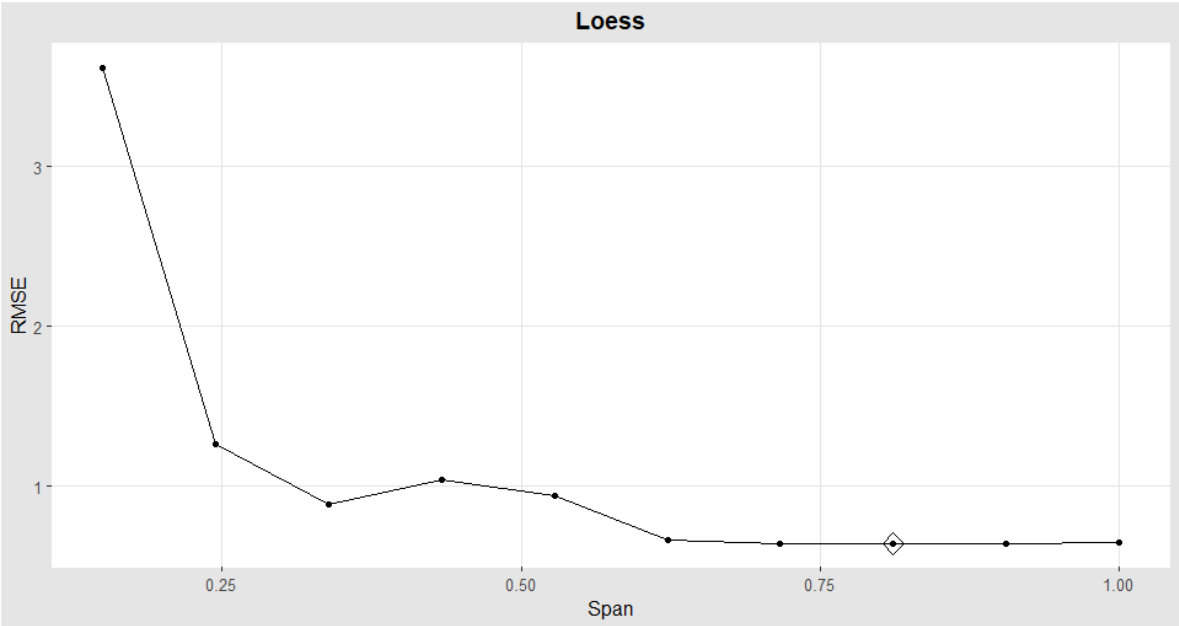
```
ggplot(fit_knn,highlight=T) + theme_igray() +
  labs(x="Neighbors", y="RMSE", title="KNN") +
  theme(plot.title = element_text(hjust=0.5,face="bold"))
```



3.5. LOESS

In this section a "Loess" model is fitted, and to find the optimal span we try different values: 0.15, 0.244, 0.338, 0.433, 0.527, 0.622, 0.716, 0.811, 0.905 and 1. As it can be seen in the results below, the best result is obtained when using a span of 0.811.

```
#Loess model
grid <- expand.grid(span = seq(0.15, 1, len = 10), degree = 1)
fit_loess <- train(quality~alcohol + volatile.acidity + citric.acid +
                   sulphates + total.sulfur.dioxide + density +
                   fixed.acidity + chlorides +pH,
                    method = "gamLoess", tuneGrid=grid, data = train_set)

predict_loess2 <- predict(fit_loess, train_set)
RMSE_loess_test <- sqrt(mean((predict_loess-test_set$quality)^2))
RMSE_loess <- sqrt(mean((predict_loess2-train_set$quality)^2))
RMSE_loess_test

## [1] 0.6588411

fit_loess

## Generalized Additive Model using LOESS
##
## 1220 samples
##    9 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 1220, 1220, 1220, 1220, 1220, 1220, ...
## Resampling results across tuning parameters:
##
##   span       RMSE       Rsquared   MAE
##   0.1500000  3.6168086  0.1351605  0.6956437
##   0.2444444  1.2670597  0.2051980  0.5524647
##   0.3388889  0.8924204  0.2369090  0.5227954
##   0.4333333  1.0404623  0.2246906  0.5288668
##   0.5277778  0.9414909  0.2336988  0.5224692
##   0.6222222  0.6670614  0.3386633  0.5024510
##   0.7166667  0.6440707  0.3695778  0.4982119
##   0.8111111  0.6427422  0.3718657  0.4978450
##   0.9055556  0.6435092  0.3706468  0.4976960
##   1.0000000  0.6473594  0.3643415  0.4995188
##
## Tuning parameter 'degree' was held constant at a value of 1
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were span = 0.8111111 and degree = 1.

ggplot(fit_loess, highlight=TRUE) + theme_igray() +
  labs(x="Span", y="RMSE", title="Loess") +
  theme(plot.title = element_text(hjust=0.5,face="bold"))
```



3.6. Model Comparison

The following table summarizes the RMSE computed both in the "train_set" and in the "test_set" for the different models:

Model	RMSE.test	RMSE.train
GLM	0.6818731	0.6434701
Decision Tree CART	0.7345924	0.6480868
Random Forest	0.6195952	0.2847609
KNN	0.8194026	0.6998118
Loess	0.6588411	0.6262135

Although 2 measures of the RMSE are given for each model, the conclusions are made taking into account the RMSE computed using the "test_set". This dataset hasn't been used to train the model , so the RMSE obtained using this dataset measures how well the model performs when used with "out-of-sample" data- this is, data that was not used to fit it. According to the RMSE computed in the "test_set", the model with the best predictive power is the "Random Forest". Therefore, the selected model is a "Random Forest" with 80 trees that uses in each split 2 predictors randomly selected. Furthermore, the variables included in this model -out of which 2 predictors are randomly selected in each split- are: alcohol + volatile acidity + citric acid + sulphates + total sulfur dioxide + density + fixed acidity + chlorides + pH.

3.7. Testing the selected model

As it was explained in the previous section, the selected model is a "Random Forest" model in which 2 predictors randomly selected in each split. Additionally, it is a "Random Forest" with 80 trees. As it was argued in the introduction, the original dataset was splitted into "wine" and "validation" in order to keep part of the data to test the selected model. This is, "wine" was used to develop the model: a part of "wine" was used to fit the different models ("train_set") and the other part of "wine" was used to test each of the fitted models ("test_set"). Once all the models have been developed and the best one has been selected, this selected model is tested again with a dataset that has not been used at all in the development of this model: it hasn't been used neither to fit it nor to compare it with other models and choose it based on the RMSE. Thus, the RMSE computed with the "validation" dataset constitutes a very good measure of the predictive power of this model because this dataset hasn't been used at all to develop this model -not even to choose it based on the RMSEs computed in the development part.

```
#RMSE on validation
pred_validation <- predict(fit_rf, validation)
rmse_validation <- sqrt(mean((pred_validation - validation$quality)^2))
```

The following table reports the RMSE computed on the "validation" dataset: the model trained in the previous section is used to predict the quality rating of the wines of this dataset, and those predictions are compared with the real ratings to compute the RMSE:

Dataset	RMSE
Validation	0.5646233

4. Conclusions

The aim of this project was to build a model able to predict a wine's quality rating given the wine's features. To build that model firstly an exploratory data analysis was conducted to explore how each feature can explain a wine's rating. Once these relationships have been preliminary explored, the correlation between each variable and "rating" is computed and the test of Pearson's correlation is employed to test if each variable is correlated with "rating". After identifying the variables that are correlated with "rating" -and thus can help us explain a wine's rating- we fit different models to predict a wine's quality rating using those variables. These variables that help us explain and predict a wine's quality are: "alcohol", "volatile acidity", "citric acid", "sulphates", "total sulfur dioxide", "density", "fixed acidity", "chlorides" and "pH".

For each model the RMSE is computed and the model that yields the best accuracy when predicting a wine's rating is selected -this is, the model with the lowest RMSE is chosen. This selected model is a "Random Forest" with 80 trees and with 2 randomly selected predictors, and its RMSE when tested with the "validation" dataset is 0.5646233. As it

has been argued, this constitutes a good measure of the predictive power of the model because it was been computed with data that has not been used at all neither to fit the model nor to choose it.

The RMSE obtained can be argued to be a good RMSE. Nonetheless, it could be improved with a bigger dataset. This is precisely one of the limitations of the model: the original dataset wasn't big, and we needed to keep some data to test the predictive power of the model. Thus, the data available to develop the model wasn't too big. On the other hand, another limitation of this analysis is that not all the features that might explain a wine's quality are available in the dataset. For instance, characteristics as the tannin levels, age, region… could help us explain a wine's quality rating and thus including them could lead to more accurate predictions. Consequently, these limitations lead to opportunities to enhance this analysis in the future: building a predictive model with a dataset with more observations and more wines' features.