# Prediction of wine scores using PLSRGLM

02/2022

Kamal Babaei
Company Interview for Gastrograph

## Introduction

You will see the explanation of my work here. There are two .R files attached to my email. One is the final function I wrote. The other one is a long step by step code to go through the data and understandings. You can run that code line to line and read the comments.

## Exploratory Analysis

At first we load the data, make a column ('Color') for each dataset and then combine the data frames by rows. (This column is considered a categorical predictor when plotting graphs and is considered numerical (with one HOT encoding) for regression purposes.

Eventually I shuffled the big data frame by rows to homogeneously mix both types of white and red wine rows. At first we do some general analysis:

```
> dim(data)
 6497   13
 summary(data)
 fixed.acidity    volatile.acidity  citric.acid     residual.sugar     chlorides
 Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600   Min.   :0.00900
 1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2500   1st Qu.: 1.800   1st Qu.:0.03800
 Median : 7.000   Median :0.2900   Median :0.3100   Median : 3.000   Median :0.04700
 Mean   : 7.215   Mean   :0.3397   Mean   :0.3186   Mean   : 5.443   Mean   :0.05603
 3rd Qu.: 7.700   3rd Qu.:0.4000   3rd Qu.:0.3900   3rd Qu.: 8.100   3rd Qu.:0.06500
 Max.   :15.900   Max.   :1.5800   Max.   :1.6600   Max.   :65.800   Max.   :0.61100
 free.sulfur.dioxide total.sulfur.dioxide    density           pH
 Min.   :  1.00      Min.   :  6.0        Min.   :0.9871   Min.   :2.720
 1st Qu.: 17.00      1st Qu.: 77.0        1st Qu.:0.9923   1st Qu.:3.110
 Median : 29.00      Median :118.0        Median :0.9949   Median :3.210
 Mean   : 30.53      Mean   :115.7        Mean   :0.9947   Mean   :3.219
 3rd Qu.: 41.00      3rd Qu.:156.0        3rd Qu.:0.9970   3rd Qu.:3.320
 Max.   :289.00      Max.   :440.0        Max.   :1.0390   Max.   :4.010
```

```
 sulphates        alcohol         color            quality
Min.   :0.2200   Min.   : 8.00   Length:6497        Min.   :3.000
1st Qu.:0.4300   1st Qu.: 9.50   Class :character   1st Qu.:5.000
Median :0.5100   Median :10.30   Mode  :character   Median :6.000
Mean   :0.5313   Mean   :10.49                      Mean   :5.818
3rd Qu.:0.6000   3rd Qu.:11.30                      3rd Qu.:6.000
Max.   :2.0000   Max.   :14.90                      Max.   :9.000
```

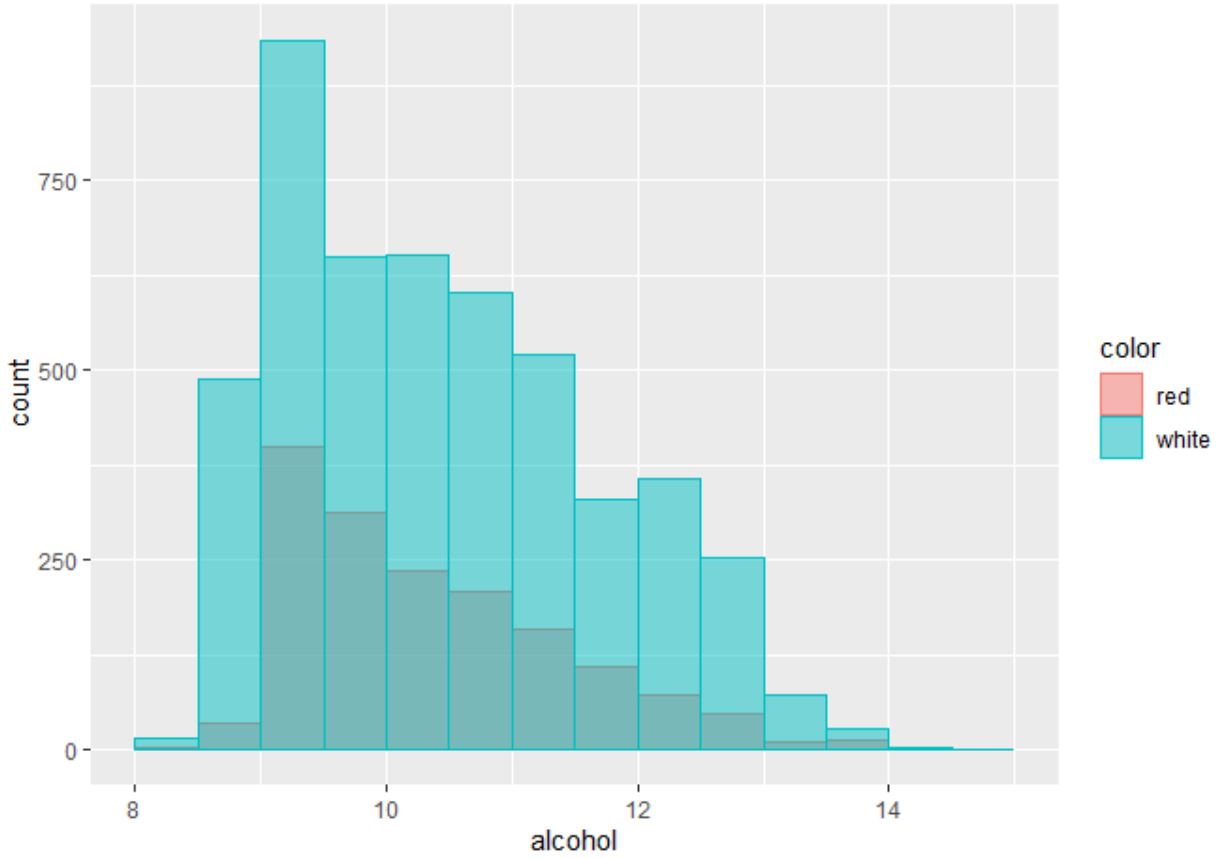
Looking for Missing Values shows that:

```
> sum(is.na(data))
[1] 0
```

great news! No N/A drama in this data set!

## Investigating single variables:

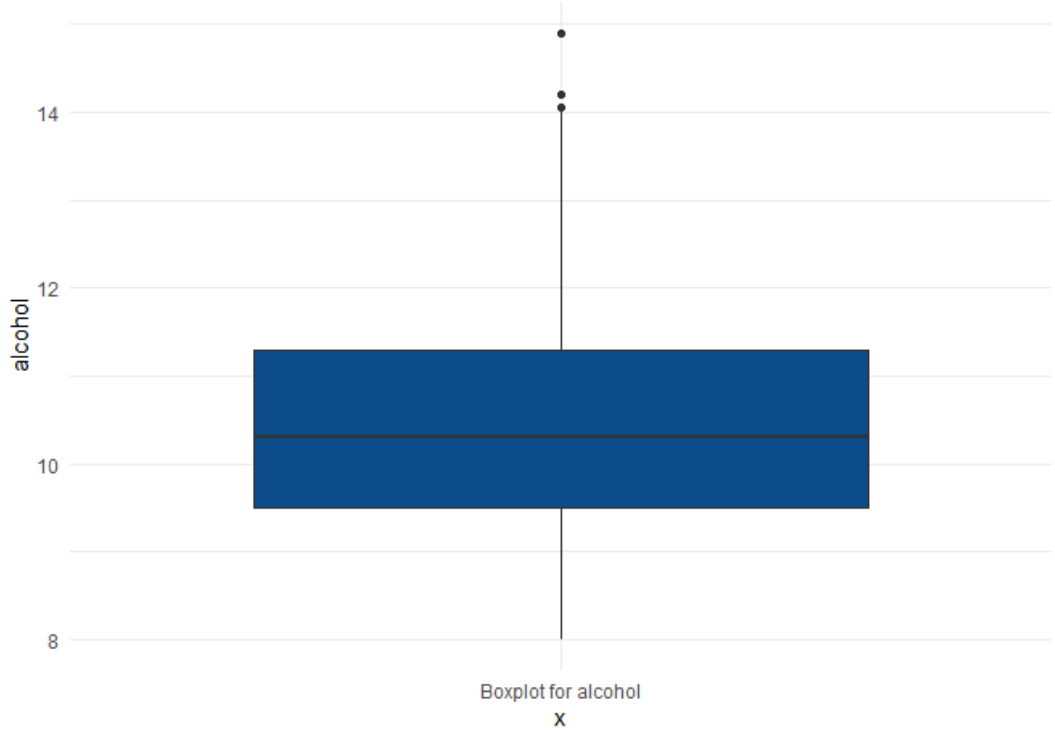
#Understanding the Distribution of Alcohol



Looks like both types of wine have the same distribution with just different counts.

According to the following Boxplot, there are 3 outliers which I removed.

The good thing about this dataset is that it has enough number of rows compared to columns that we can carefully remove the outliers being sure we are not damaging the model

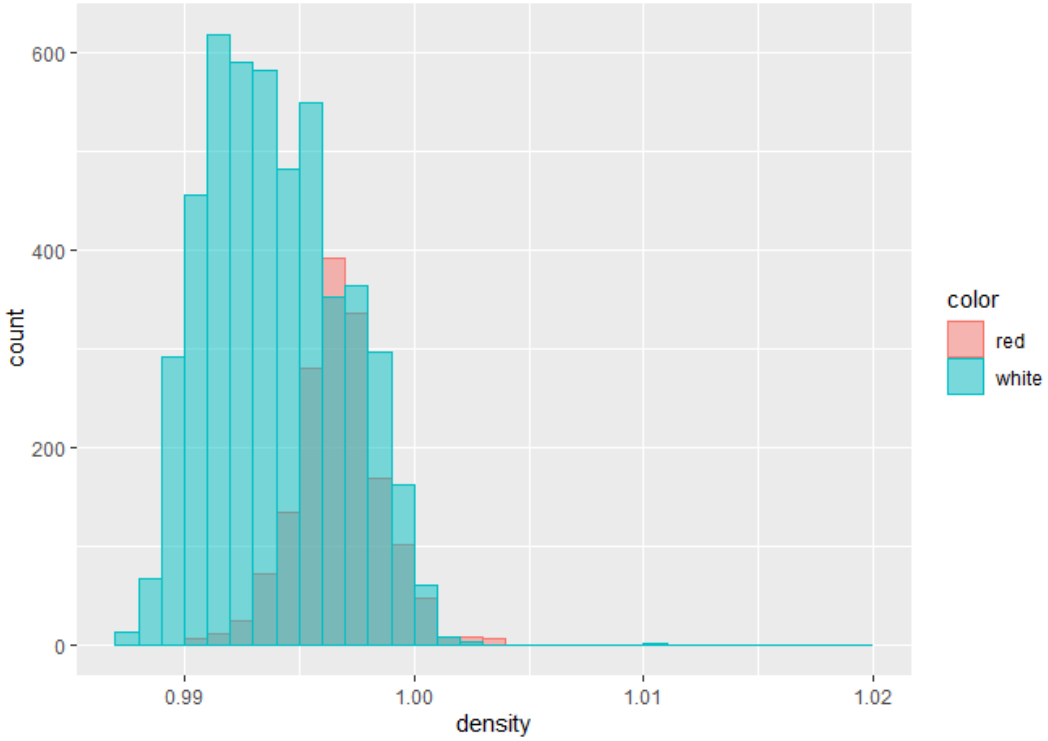


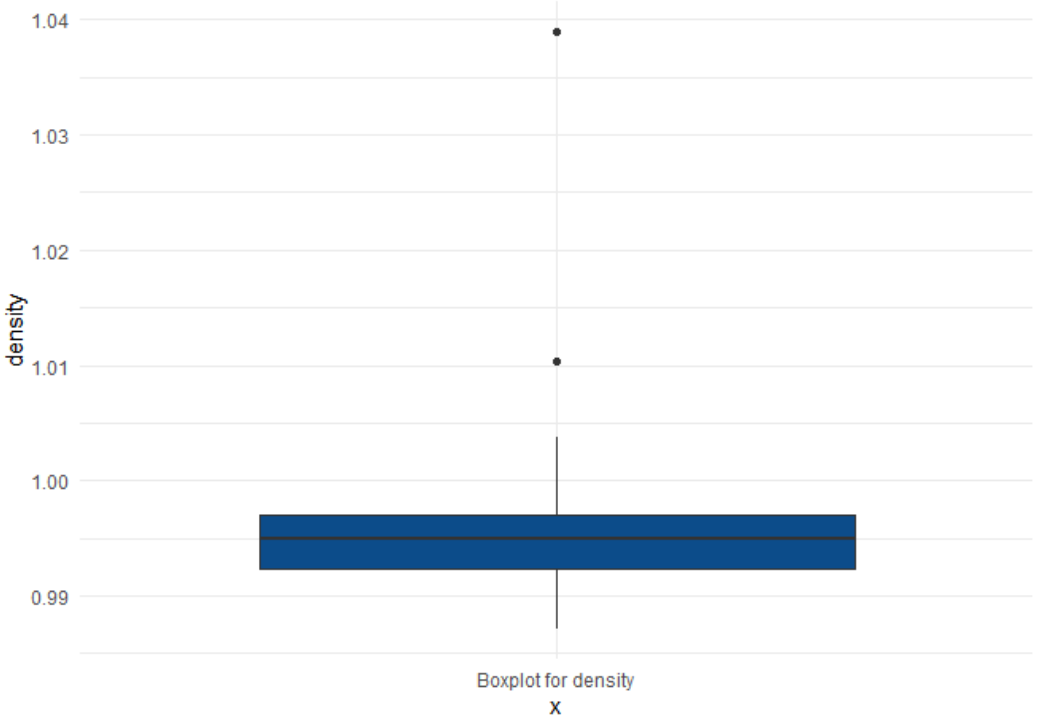

## #Understanding the Distribution of the density of wine

Looks like in general red wine is more dense than whte wine.

The distribution of red wine is almost normal, the white wine has a barely multimodal distribution.

According to the following Boxplot, there are 3 outliers which I removed.

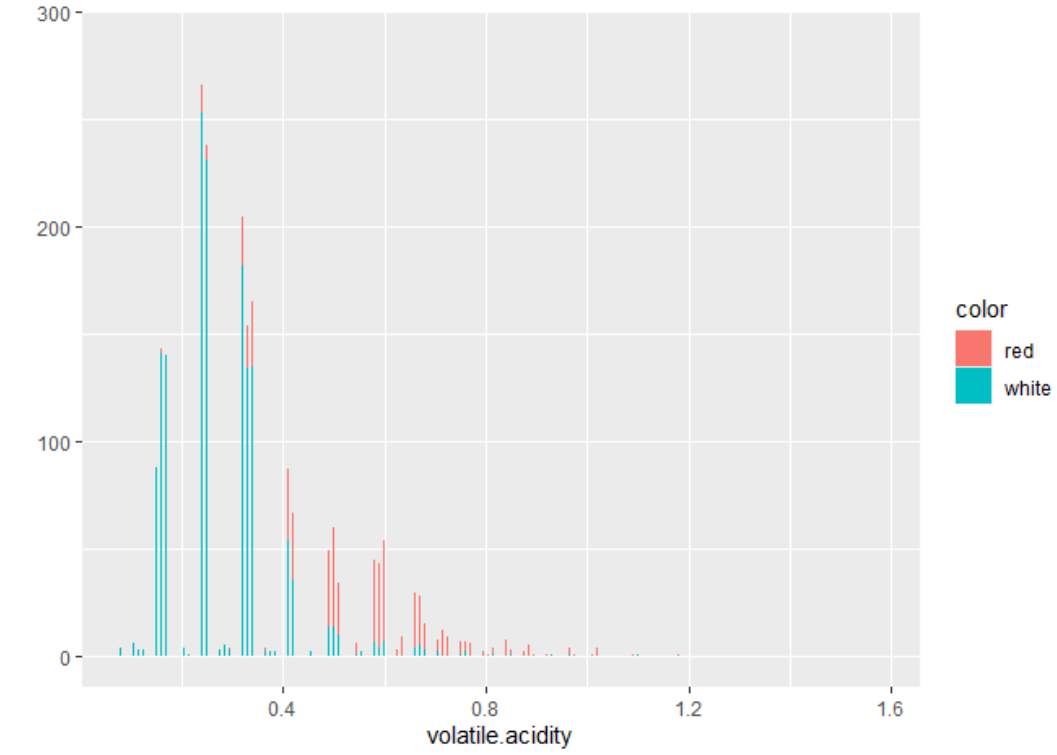

600
400
200
0
count
0.99
1.00
1.01
1.02
density
color
red
white

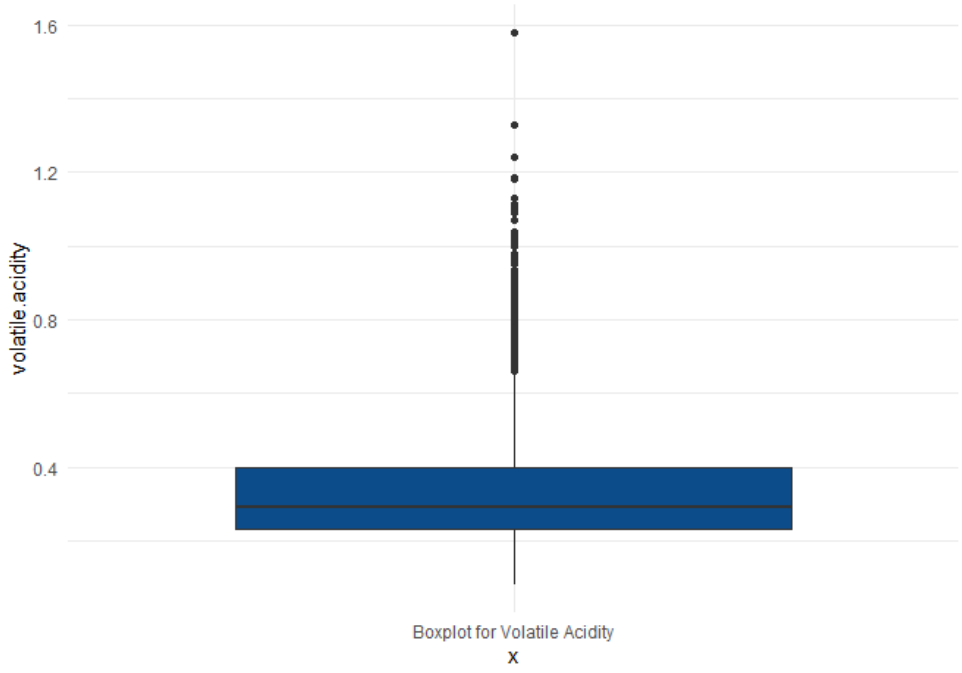

1.04
1.03
1.02
1.01
1.00
0.99
density
Boxplot for density
x

## #Understanding the Distribution of Level of Volatile Acidity

Looking at the boxplot we see that this one has too many outliers (all on the upper bound).

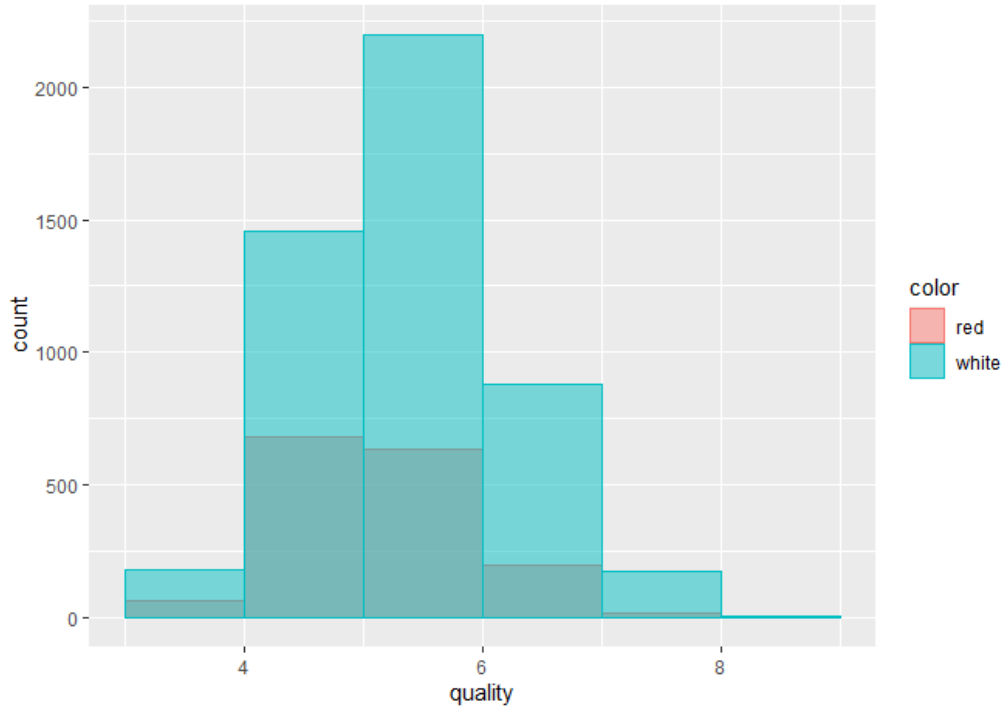
There were too many to delete so I changed the upper bound to (Q[2]+4*iqr) instead of (Q[2]+1.5*iqr)

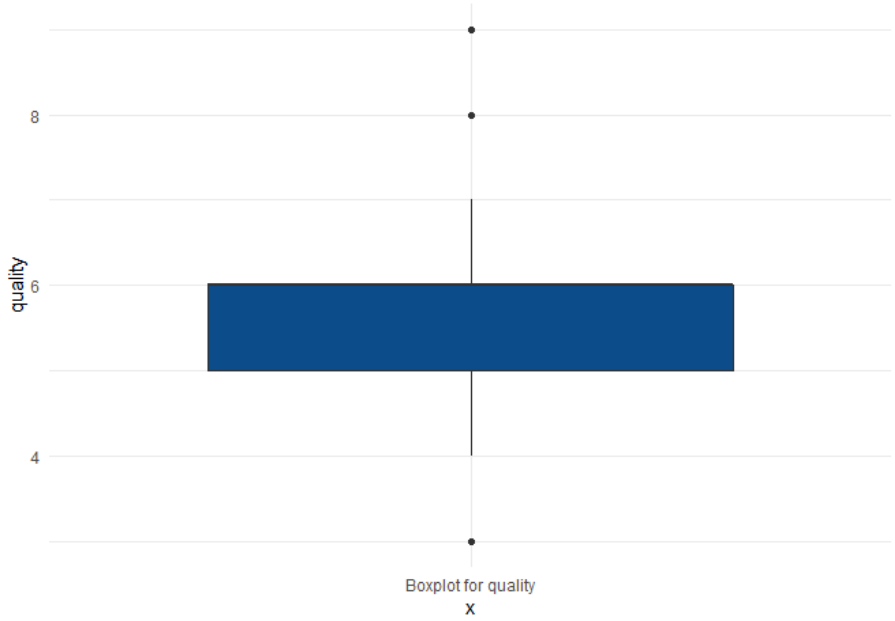


## #Understanding the Distribution of quality

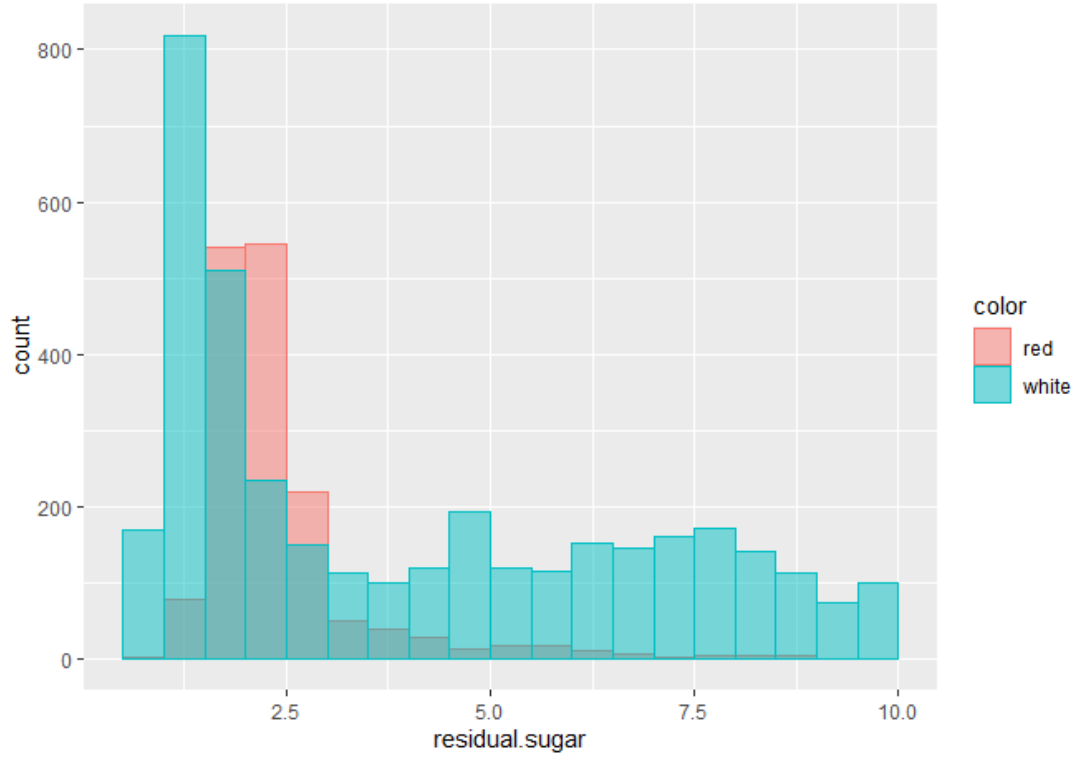
This histogram is the most important histogram. It looks like both red and white wines are almost normally distributed, with a slight positive skewness (to the right). It is very interesting that scores “3” and “9” have such small counts. The concentration in the middle is so big that both “8” and “9” are considered outliers according to the box plot. We will see what to do with them later on this report.

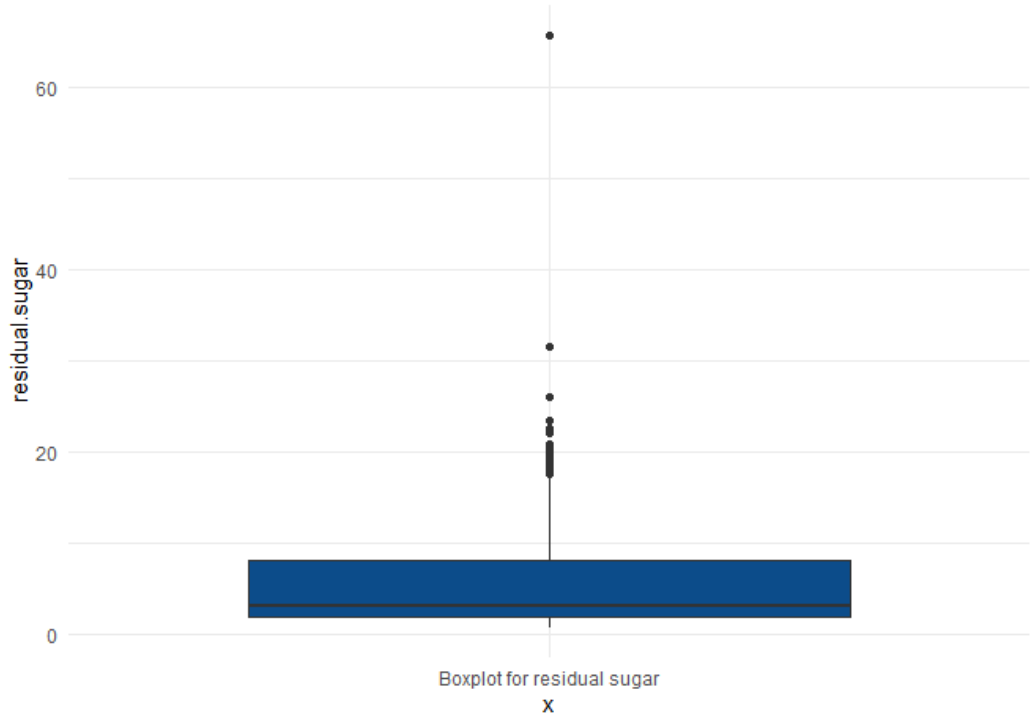


## #Understanding the Distribution of residual sugar

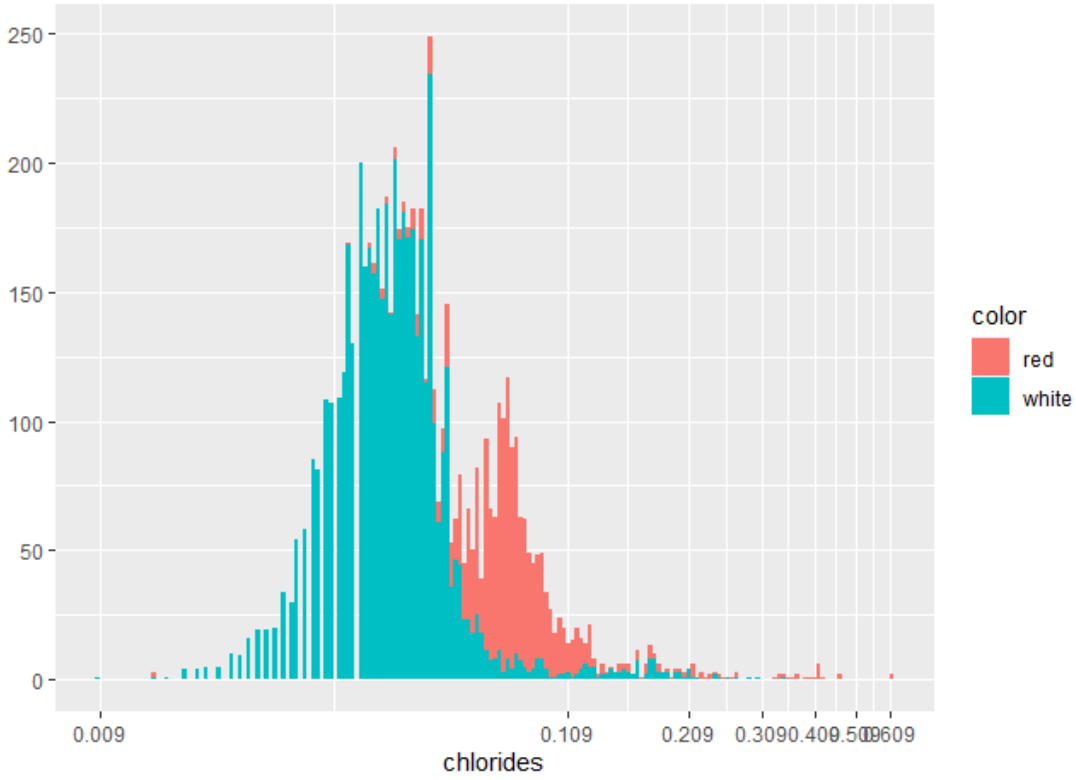
It could be seen that white wine has a little more sugar in general. The boxplot has several outliers but one very thick one. I cut off the top 3 outliers by changing the Q[2]+1.5*iqr to Q[2]+3*iqr

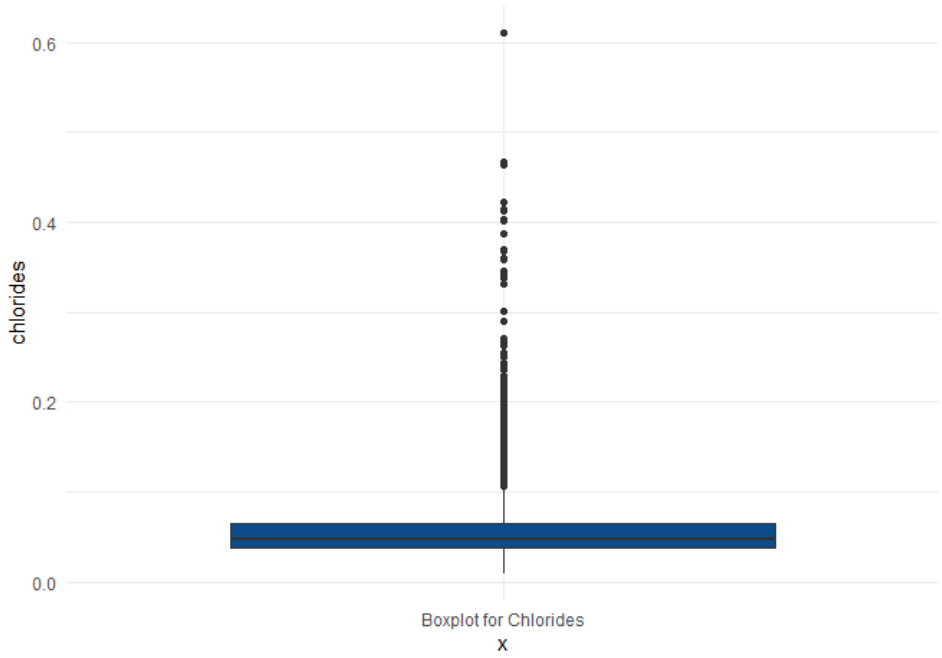


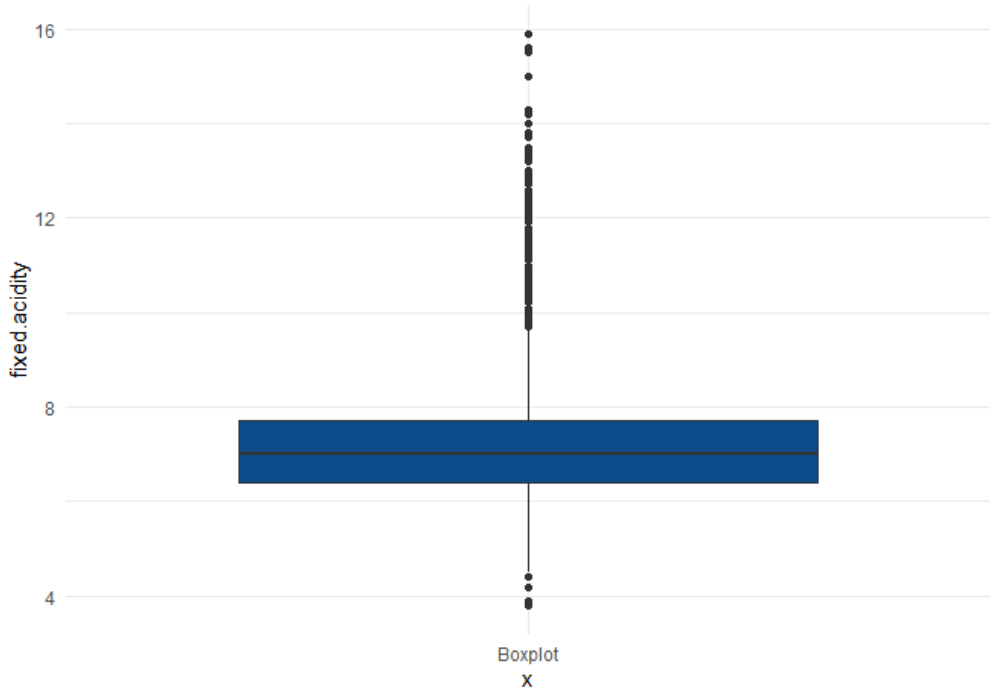
## #Understanding the Distribution of Level of Chlorides

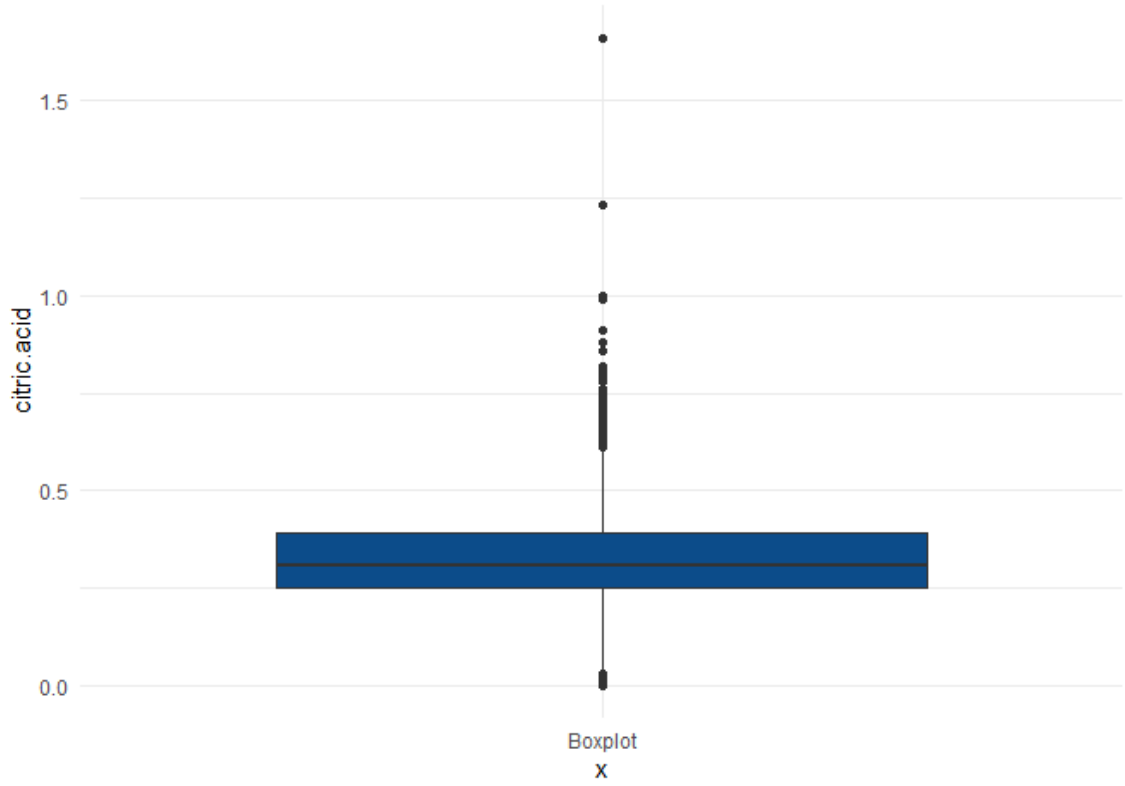
It looks like the level of Chlorides is more in general. In addition, again, we see several outliers but I decided to remove top 3 by changing Q[2]+1.5*iqr to (Q[2]+3.5*iqr.

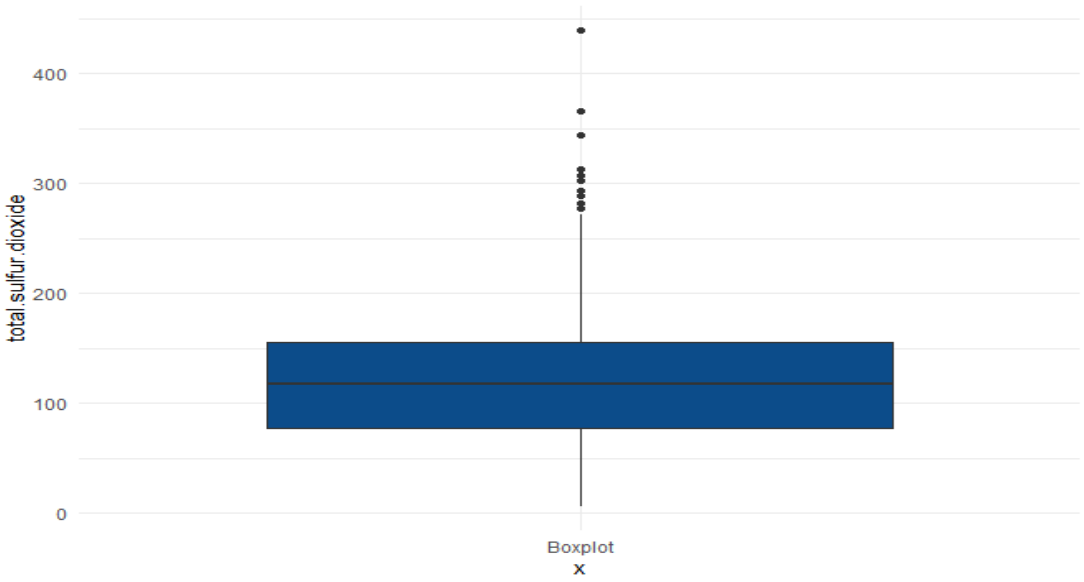


For the rest of single predictors, I will briefly explain the outlier situation, to save time and run to the training part. All additional information could be obtained by running the code. citric acid>1.2 removed (2 points). Total.sulfur.dioxide>350 removed (2 points).

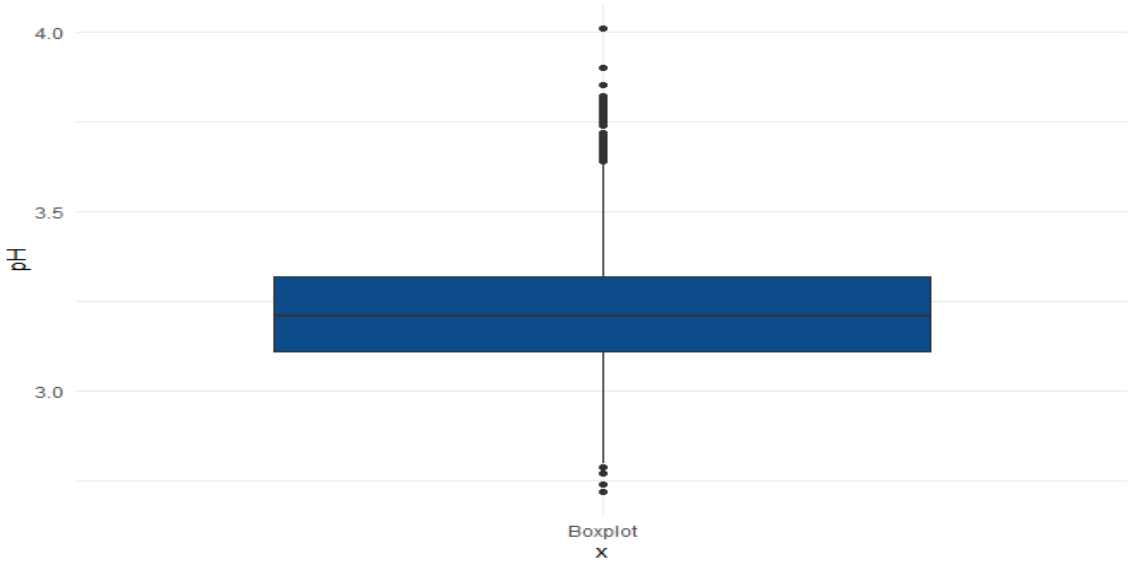


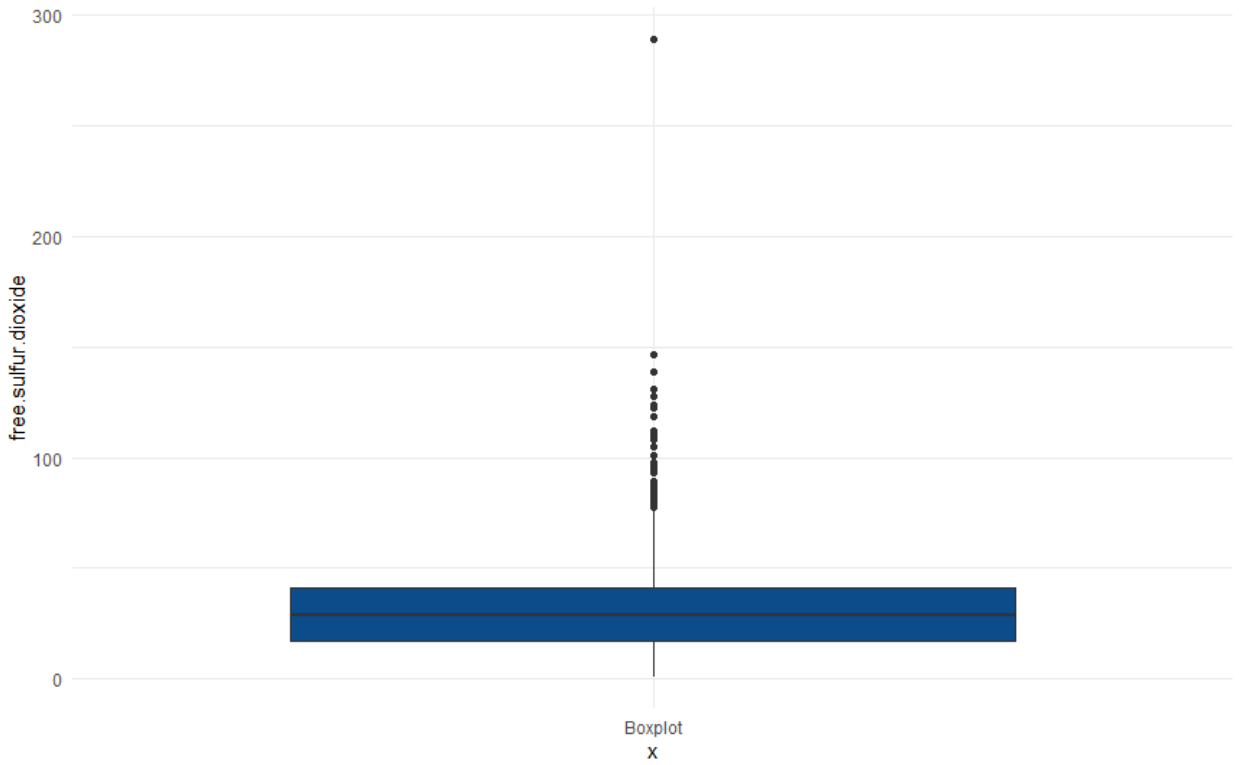



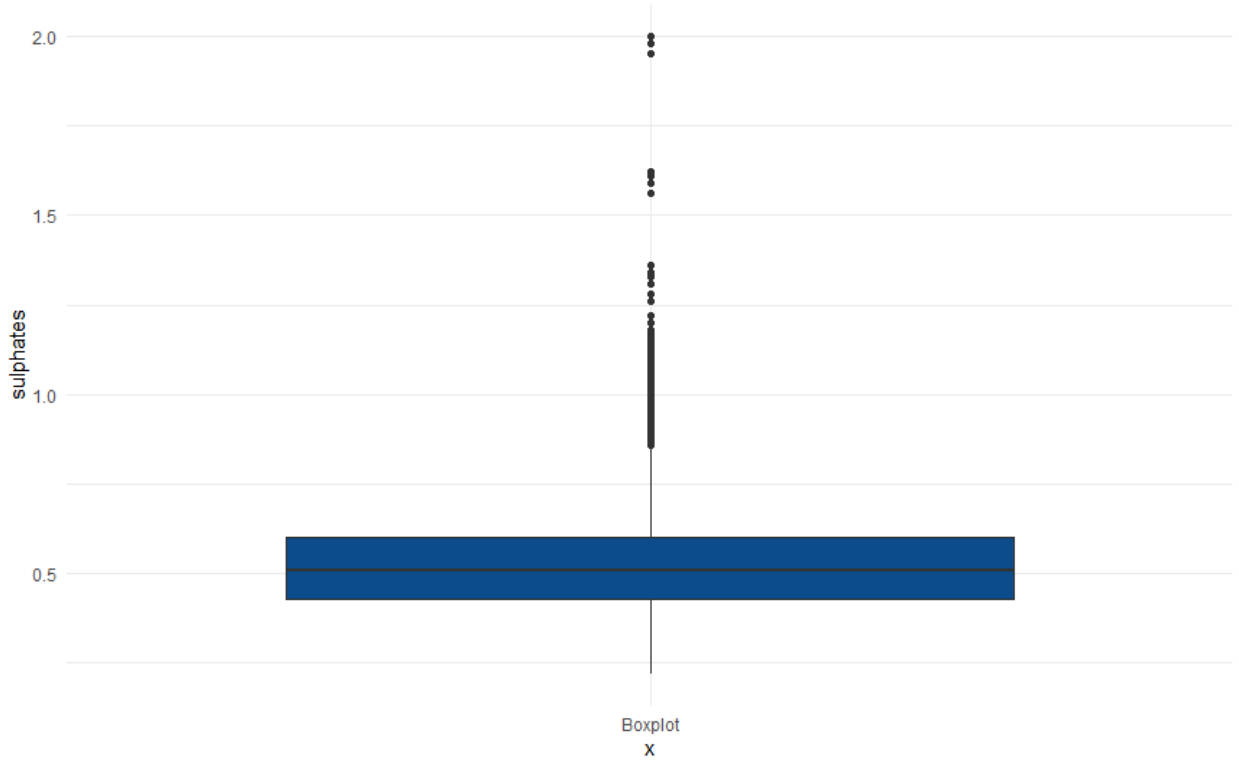



Sulfate has several outliers but the very top ones look really odd so I will just focus on those. I change the upper bound from Q[2]+1.5*iqr to (Q[2]+5*iqr.

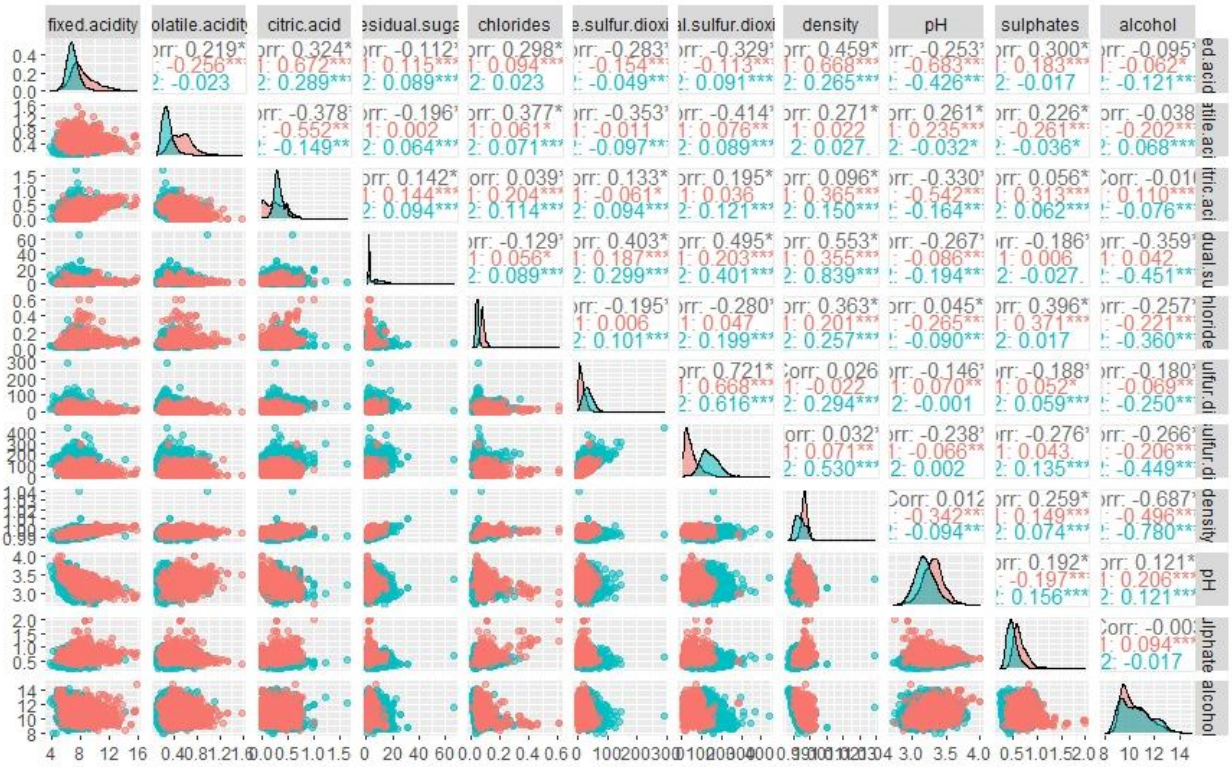
pH is not so unusually distributed. It looks good to me. This range of pH is to be expected.




300
200
100
0
free.sulfur.dioxide
Boxplot
x


2.0
1.5
1.0
0.5
sulphates
Boxplot
x

## ##Correlation of every pair




The correlations look pretty small/average. there is only one 0.72 which is cor(free sulfur dioxide and total sulfur dioxide). Intuitively, that is to be expected. The rest of the correlations are pretty tolerable. If I saw large correlations I would think of combining columns or deleting one column. But here correlations are not that big, so we do not have to worry about multicollinearity.

**quality has the strongest correlation with alcohol** (Wine tasters love alcohol don't they!)

the more volatile.acidity and chlorides, the (slightly) less score. I think it could be interpreted as the taste of volatile acid could be really annoying if it is slightly over the tolerance of a taster. But we cannot be sure until we build the complete model. The rest of my observation is as follows.

A few predictors are significantly different in white and red wines:

Fixed Acidity, Volatile Acidity, Residual Sugar, Total Sulfur Dioxide

# Model Building

Initially I used the "train" function from "caret" and picked my training method as: 'plsRglm'

Manipulating the inputs in this function I figured out a few very important facts:

-**PREPROCESS:** changing the "Preprocess=center, scale" argument does not seem to make a big difference at all. I think this happens because, PLS method already picks the best directions of components which describe the data matrix (as well as account for the relationship between X and Y). Therefore, the components are picked almost regardless of the scaling and centering of predictors.

-**ROUNDING:** When I looked at the outcomes of my model prediction, I noticed they are real numbers but what we really want to predict is an integer. So I thought of rounding the predicted Y-response up or down to make it an integer. Rounding led to a substantial decrease in MAE (at the expense of a very tiny raise in RMSE). So I decided to do it on the next runs; because it also makes sense. When the machine gives you a prediction of "6.8", don't you think "Oh you mean 7!".

**-Range of predictions:** I noticed that my model is predicting the test data in the range of (4.35,7.1)

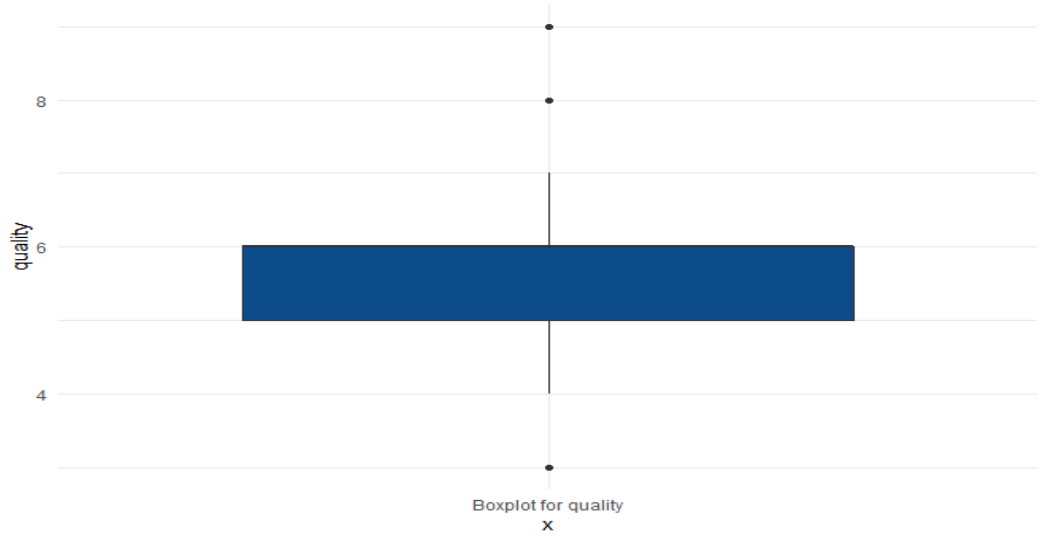
Looking at the range of "Quality" column, this makes sense:

```
> table(data$quality)

  3   4    5    6    7   8   9
 30 216 2138 2836 1079 193   5
```

When the machine is learning, it is not even becoming familiar with real high or real low scores; what the machine sees is mainly 4,5,6,7. So it is expected to predict the majority in the mentioned range.

**Let's dig more into this**, when we saw the Boxplot for quality, quality scores such as (3,9) were outliers. I thought and researched a lot about what to do with these.



First idea that might come to mind was to look at “quality” as a categorical value and therefore do some OverSampling or UnderSampling to deal with the imbalanced data. I rejected this; I think it is not in our benefit to do that. Imagine if I over sampled the minority class and now have a huge number of “8”s and “9”s. One would say, dude where do you live that everyone likes to give such high scores to wines (or any other survey). The idea looked unreal. I did not want to do just ANYTHING to decrease my MAE.

Second Idea is to delete the extreme outliers. Initially I thought of cutting equally from both tails; however, because I see a skewness (to the right), I prefer to cut the extreme values of the right tail (score=8,9) and also the far lower bound outlier (score=3). In other words: keep the values which are between $\mu-3\sigma$ and $\mu+2\sigma$.

**Trying a model without the “color” predictor:**

I trained the model without this column to see if anything significant happens. I saw no change.

**Trying the model for different colors individually:**

The prediction of MAE is much better for the “red” data set. Maybe the reason is that Red does not have any "9" score; nor a huge percentage of "8" or "3" either. In general its histogram is much more dense. That is why it has less error. Therefore, I decided to take advantage of this fact and build two different models based on color.

**Trying to train the model with high importance predictors:**

I extracted the high importance predictors (Alcohol, Density, Volatile Acidity) and trained the model based on them. I did not see any improvement in the result. I think this happens because PLS already takes care of the information in predictors.

**Trying to implement PCA:**

I tried preprocessing my data matrix with the PCA method before training the model with it. It did not help. MAE increased but Rsquared decreased. RMSE slightly increased.

**plsRglm function:**

plsRglm() does the modeling just like train(method=’plsRglm’) does. The difference is that train() function gives you different combinations of alpha.pvals.expli and nt (up to 3 components); however, plsRglm() lets you choose the hyperparameters, I wrote a grid to pick the two hyperparameters which give out the best test MAE. Even though I did not see a big change over different values of alpha.pvals.expli. Alpha has a very minor effect on the final MAE but I included it anyway. In order to make it more optimal, I wrote the grid for both colors of wine.

plsRglm has 3 other modeles as well: “Gamma”, “Poisson” and “inverse Gaussian”. As I had predicted before running them, they all gave terrible results (MAE around 3.5 to 6) so I did not consider trying them. I think the reason they did not work is that our wine data set seems to have such a “normal”

distribution. It is not “the number of cars passing through the same street hourly” or anything like Poisson. It definitely is far from Gamma and Inverse Gaussian. So let's stick to Gaussian.

**My Function:**

My function (AFS) takes two arguments (train, test) and outputs the best MAE using plsRglm. It takes advantage of all the findings explained above. It treats the outliers, trains on two different colors, rounds the fitted values to make them integer scores and picks the best MAE after gridding over all hyperparameters. The intuition behind using two machines (one for each color) was that I noticed how red wine data is better at predicting response so I wanted to take advantage of this. The overall best MAE that the function gives, is the weighted mean of both MAEs from both colors. I interpret it as “On average(any color), how far is your estimation of score from reality”. I ran my function 10 times (with different splits of data) and the average of all the responses was:  0.5013

Thank you for the opportunity; it was fun working and thinking and researching!