PRACTICA 2 - Tipologia y ciclo de vida de los datos

Miguel Rodriguez Olmos

Contents

1 Dataset description. **1**

2 Data loading and feature selection. **1**

3 Cleaning the data. **3**

4 Data analysis. **6**
4.1 Feature selection. 6
4.2 Normality assumptions and homogeneity of variance. Statistical tests 12
4.3 A linear regression model. 20

5 Graphic representation of the results. **21**

6 Solution to the problem. **22**

1 Dataset description.

This dataset contains data about a red variant of the Portuguese 'vinho verde' (green wine). The main idea underlying this set is that the quality score of a given wine, which is in principle decided by experts, and is largely a subjective matter, can be awarded purely in terms of some chemical properties of the wine. Therefore, this dataset includes the results of 11 different chemical tests for a selection of wines,, as well as the quality score of the wine, ranking from 1 to 10.

2 Data loading and feature selection.

We start by loading the dataset for our analysis.

```
library(ggplot2)
df <- read.csv('winequality-red.csv')
```

We now inspect the first few observations of the dataset.

```
head(df)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.4             0.70        0.00            1.9     0.076
## 2           7.8             0.88        0.00            2.6     0.098
## 3           7.8             0.76        0.04            2.3     0.092
## 4          11.2             0.28        0.56            1.9     0.075
## 5           7.4             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  11                   34  0.9978 3.51      0.56     9.4
## 2                  25                   67  0.9968 3.20      0.68     9.8
## 3                  15                   54  0.9970 3.26      0.65     9.8
## 4                  17                   60  0.9980 3.16      0.58     9.8
```

```
## 5                 11                  34  0.9978 3.51      0.56     9.4
## 6                 13                  40  0.9978 3.51      0.56     9.4
##   quality
## 1       5
## 2       5
## 3       5
## 4       6
## 5       5
## 6       5
```

As a preliminary step, we will inspect the type of each variable. Notice that all variables consist on quantitative chemical tests as well as an ordered numeric quality score, so we expect all the variables to be numeric.

```
apply(df, 2, class)
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##            "numeric"            "numeric"            "numeric"
##       residual.sugar            chlorides  free.sulfur.dioxide
##            "numeric"            "numeric"            "numeric"
## total.sulfur.dioxide              density                   pH
##            "numeric"            "numeric"            "numeric"
##            sulphates              alcohol              quality
##            "numeric"            "numeric"            "numeric"
```

They are indeed, all numeric. Notice, however, that the quality variable is an integer. We are going to recode it into the real field since we are going to look at this variable as a dependent variable in a regression setting, and therefore we do not want to be regarded as a discrete categorical variable.

```
df$quality <- as.double(df$quality)
is.double(df$quality)
```

```
## [1] TRUE
```

Next, we are going to obtain a quick summary of our dataset.

```
summary(df)
```

```
##  fixed.acidity   volatile.acidity  citric.acid    residual.sugar
##  Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900
##  1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900
##  Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200
##  Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539
##  3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600
##  Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide
##  Min.   :0.01200   Min.   : 1.00       Min.   :  6.00
##  1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00
##  Median :0.07900   Median :14.00       Median : 38.00
##  Mean   :0.08747   Mean   :15.87       Mean   : 46.47
##  3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00
##  Max.   :0.61100   Max.   :72.00       Max.   :289.00
##     density             pH          sulphates         alcohol
##  Min.   :0.9901   Min.   :2.740   Min.   :0.3300   Min.   : 8.40
##  1st Qu.:0.9956   1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50
##  Median :0.9968   Median :3.310   Median :0.6200   Median :10.20
##  Mean   :0.9967   Mean   :3.311   Mean   :0.6581   Mean   :10.42
##  3rd Qu.:0.9978   3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10
##  Max.   :1.0037   Max.   :4.010   Max.   :2.0000   Max.   :14.90
```

```
##     quality
##  Min.   :3.000
##  1st Qu.:5.000
##  Median :6.000
##  Mean   :5.636
##  3rd Qu.:6.000
##  Max.   :8.000
```

We can see that the various variables are in different scales, which can be a problem for certain algorithms of machine learning, especially those based on distances. However we are not going to use those techniques in this work. We can also see that the dataset provided does not exhibit enough samples to cover the full range of the dependent variable `quality` which is supposed to range from 0 to 10.

As the end of the preliminary study of our data, we also look at the structure of the dataset.

```
str(df)
```

```
## 'data.frame':    1599 obs. of  12 variables:
##  $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
##  $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
##  $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : num  5 5 5 6 5 5 5 7 7 5 ...
```

Apart from double checking that all of our variables are numeric, we see that we have 1599 rows in our dataset.

3 Cleaning the data.

We are now going to see how tidy this dataset is. In particular we are going to focus on null values, as well as zeros and outliers. The case of zero values is included since it could be the case that in observations for which some variable is not informed, it may be filled as a zero, instead of a null value. Another possibility is that some extreme value (e.g. 999) is used to this end, in which case we would have extreme values in our dataset that actually correspond to null values. Finally, outliers could also come from legitimate rare situations or they could be due to some inaccurate measure, so ideally we should be able to distinguish among these cases.

We start by looking at zeros.

```
colSums(df==0)
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##                    0                    0                  132
##       residual.sugar            chlorides  free.sulfur.dioxide
##                    0                    0                    0
## total.sulfur.dioxide              density                   pH
##                    0                    0                    0
##            sulphates              alcohol              quality
##                    0                    0                    0
```

We can see that the variable `citric.acid` is the only one with zero values, but it has a large number of them, almost a 10%. Therefore we should be careful about what to do in this situation. A quick look on Wikipedia (https://en.wikipedia.org/wiki/Acids_in_wine#Citric_acid) reveals that the grape fruit has a minimal amount of citric acid. Actually, the citric acid found in wines is most of the time added as as a supplement in order to boost the total acidity of the wine. Therefore we are going to leave those zero values as they are since there are not reasons to believe that they have not been correctly informed.

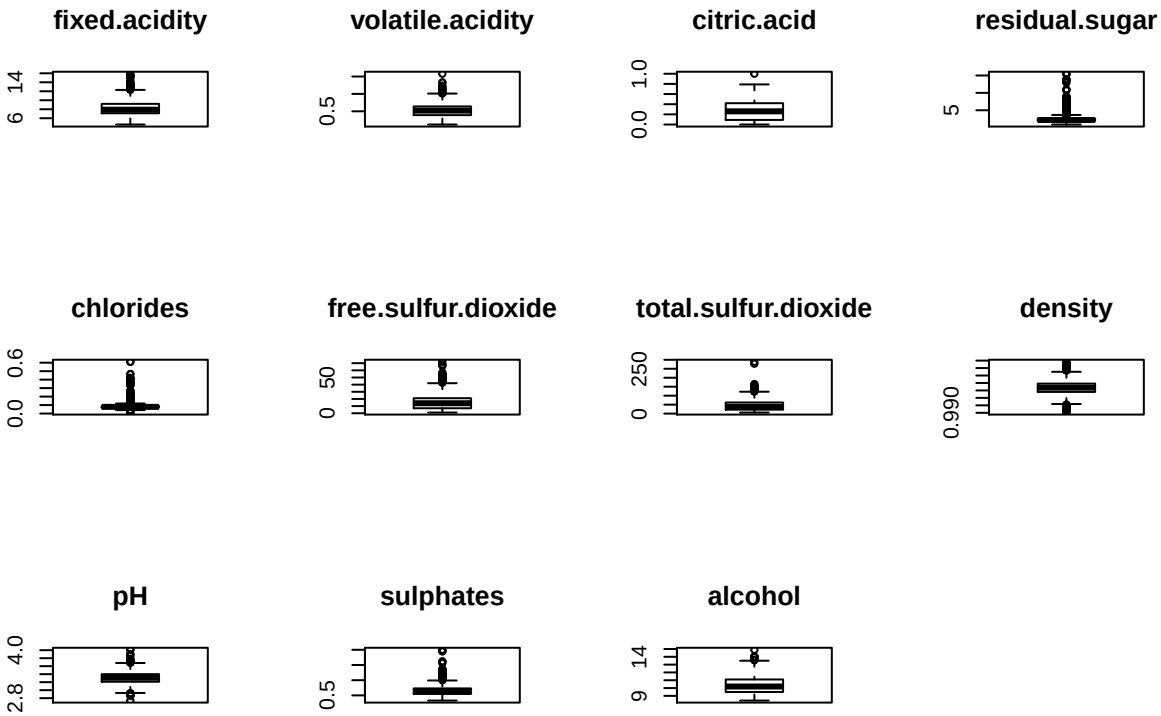
Now for the case of null values, we will count the total number existing in the dataset.

```
colSums(is.na(df))
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##                    0                    0                    0
##       residual.sugar            chlorides  free.sulfur.dioxide
##                    0                    0                    0
## total.sulfur.dioxide              density                   pH
##                    0                    0                    0
##            sulphates              alcohol              quality
##                    0                    0                    0
```

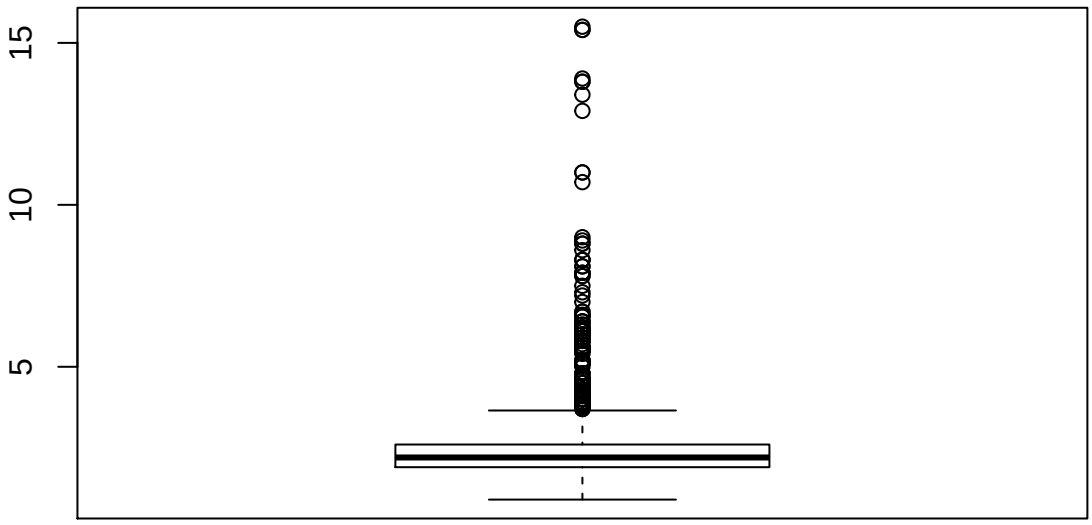
And this shows that there are not null values in this dataset. We now look at extreme values and outliers of each variable. We will visually examine this situation using boxplots.

```
par(mfrow = c(3, 4))
for (i in 1:(ncol(df)-1)){
  boxplot(df[i], main = colnames(df)[i])
}
```



We can see that most variables present outliers, many of them in some cases. We will take, for instance the variable `residual.sugar` and count the number of outliers.

```
length(boxplot(df$residual.sugar)$out)
```



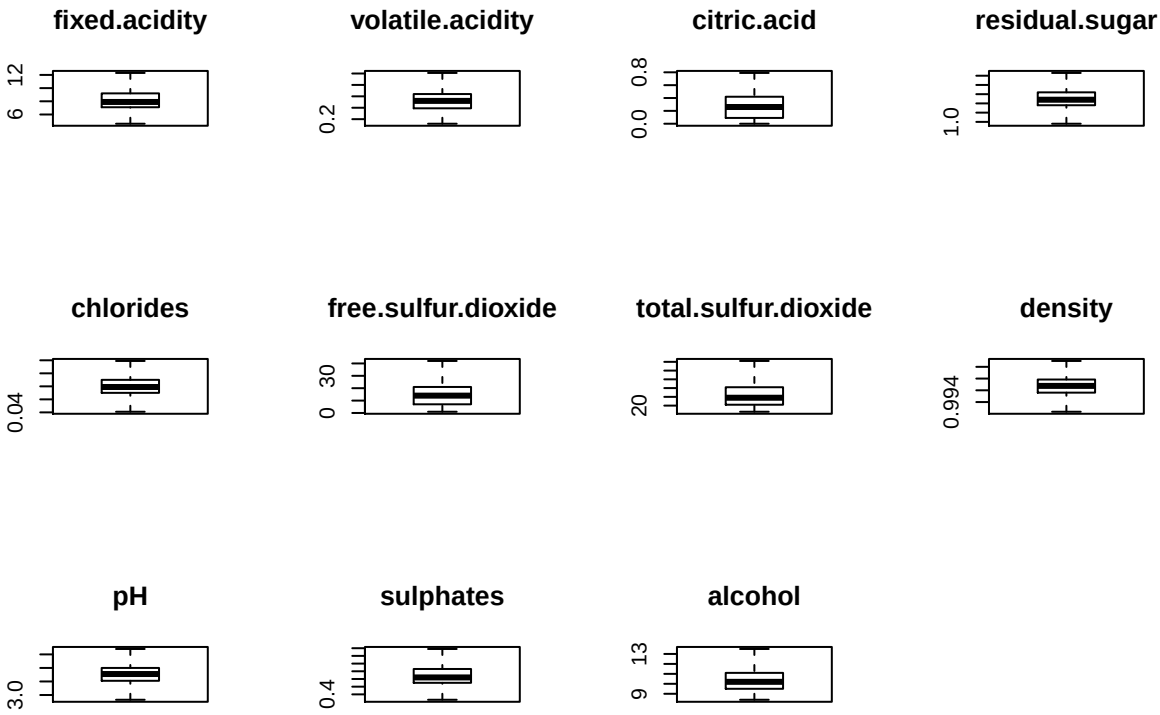

```
## [1] 155
```

We are getting roughly a 10% of observations detected as outliers by the boxplot function (values being more than 1.5 times IQL away from a whisker). These are obviously too many to be deleted, especially since they are distributed among many of the variables. Therefore the strategy will be to impute them with the corresponding whisker value. We will do this for all variables.

```
for (i in 1:(ncol(df)-1)){
  df[df[i] > boxplot(df[i])$stats[5,], i ] = boxplot(df[i])$stats[5,]
  df[df[i] < boxplot(df[i])$stats[1,], i ] = boxplot(df[i])$stats[1,]
}
```

We now generate again the boxplots for the variables of our dataset and check if there are still outliers.

```
par(mfrow = c(3, 4))
for (i in 1:(ncol(df)-1)){
  boxplot(df[i], main = colnames(df)[i])
}
```



As it should be the case, we are not getting any outliers this time.

4 Data analysis.

We will now analyze our clean data. We will start by deciding which of the variables are relevant to our analysis, and continue by testing the normality of the retained variables, which will be needed for the subsequent statistical analyses that may need this normality as a hypothesis. Finally, we will apply a linear regression in order to obtain the best linear function that explains the quality factor with respect to the relevant variables.

4.1 Feature selection.

In order to select appropriate variables for our analysis, we will perform a study of the linear correlation of the independent variables, fixing a threshold of 0.6 and removing all variables that exhibit a correlation coefficient higher than this figure with any other variable.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.1
```

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

```
correlationMatrix <- cor(df[,1:ncol(df)-1])
print(correlationMatrix)
```

```
##                      fixed.acidity volatile.acidity citric.acid
## fixed.acidity          1.00000000     -0.268474131  0.67935788
## volatile.acidity      -0.26847413      1.000000000 -0.56170521
## citric.acid            0.67935788     -0.561705208  1.00000000
## residual.sugar         0.21475784      0.039474103  0.18306211
## chlorides              0.22970119      0.133552160  0.14643613
```

```
## free.sulfur.dioxide    -0.15708874     -0.005263289 -0.06087960
## total.sulfur.dioxide   -0.11949044      0.091673643  0.01899204
## density                 0.66763018      0.016896851  0.36959694
## pH                     -0.69007120      0.235498899 -0.54586267
## sulphates               0.21577379     -0.317596905  0.33259922
## alcohol                -0.07049368     -0.209620036  0.11216376
##                      residual.sugar   chlorides free.sulfur.dioxide
## fixed.acidity            0.21475784  0.22970119        -0.157088745
## volatile.acidity         0.03947410  0.13355216        -0.005263289
## citric.acid              0.18306211  0.14643613        -0.060879601
## residual.sugar           1.00000000  0.20962988         0.082932518
## chlorides                0.20962988  1.00000000        -0.012045589
## free.sulfur.dioxide      0.08293252 -0.01204559         1.000000000
## total.sulfur.dioxide     0.15725020  0.09876342         0.687184488
## density                  0.42291570  0.40886365        -0.033836925
## pH                      -0.09452997 -0.25931554         0.075585326
## sulphates                0.03584784  0.10303843         0.049219601
## alcohol                  0.10711447 -0.29666085        -0.068099040
##                      total.sulfur.dioxide     density          pH
## fixed.acidity                -0.119490443  0.66763018 -0.69007120
## volatile.acidity              0.091673643  0.01689685  0.23549890
## citric.acid                   0.018992040  0.36959694 -0.54586267
## residual.sugar                0.157250195  0.42291570 -0.09452997
## chlorides                     0.098763419  0.40886365 -0.25931554
## free.sulfur.dioxide           0.687184488 -0.03383692  0.07558533
## total.sulfur.dioxide          1.000000000  0.09649019 -0.05191122
## density                       0.096490191  1.00000000 -0.33599217
## pH                           -0.051911224 -0.33599217  1.00000000
## sulphates                    -0.004658412  0.16108202 -0.13247253
## alcohol                      -0.228860469 -0.50112893  0.19488988
##                        sulphates     alcohol
## fixed.acidity         0.215773788 -0.07049368
## volatile.acidity     -0.317596905 -0.20962004
## citric.acid           0.332599220  0.11216376
## residual.sugar        0.035847838  0.10711447
## chlorides             0.103038429 -0.29666085
## free.sulfur.dioxide   0.049219601 -0.06809904
## total.sulfur.dioxide -0.004658412 -0.22886047
## density               0.161082023 -0.50112893
## pH                   -0.132472530  0.19488988
## sulphates             1.000000000  0.15771515
## alcohol               0.157715150  1.00000000
```

```
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.6)
print("the variables showing a high correlation are:")
```

```
## [1] "the variables showing a high correlation are:"
```

```
print(highlyCorrelated)
```

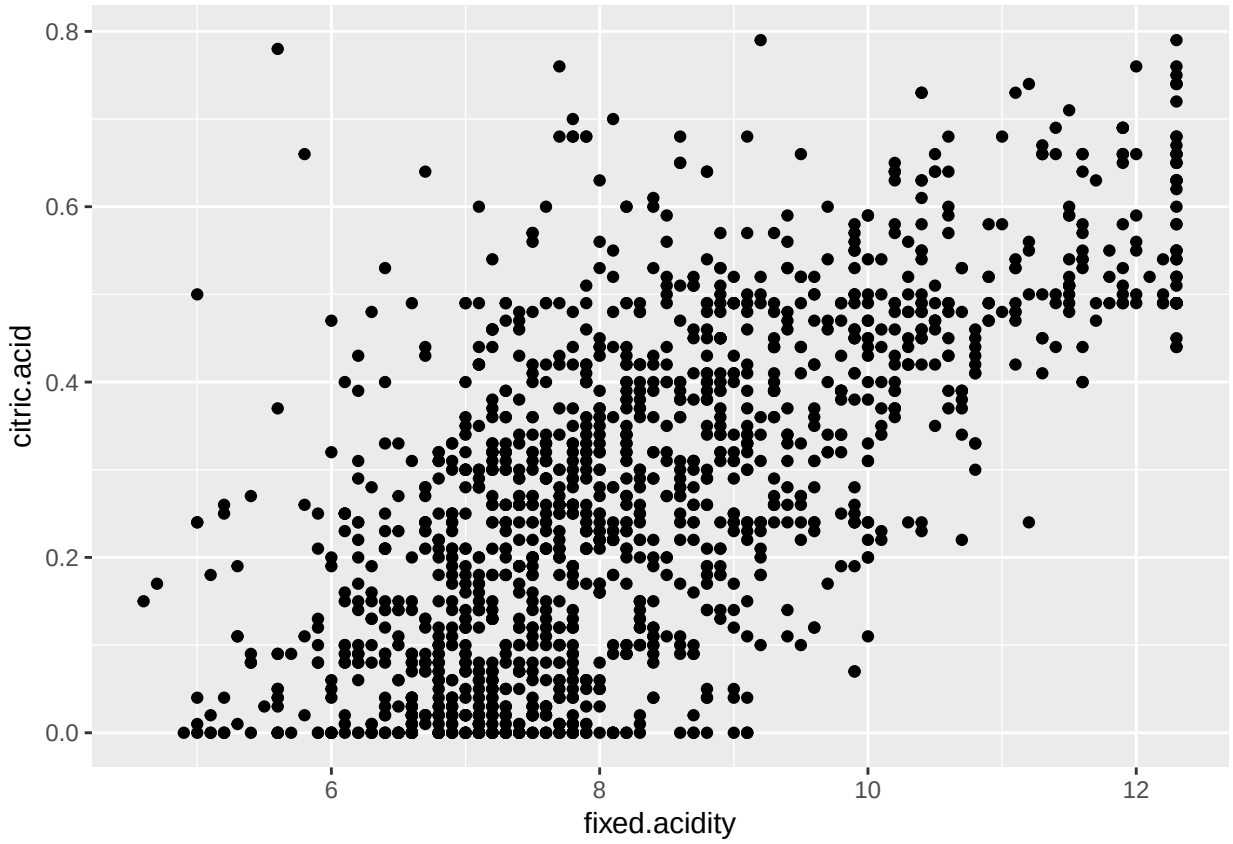
```
## [1] 1 6
```

As we can see, the variables with indexes 1 and 6 are those that are highly correlated with some other variable. From the above correlation matrix we see that these are `fixed.acidity` and `free.sulfur.dioxide`. By inspection of this matrix we can see that `fixed.acidity` is highly correlated with `citric.acid`, `density` and `pH`, while `free.sulfur.dioxide` is highly correlated with `total.sulfur.dioxide`. We will inspect visually

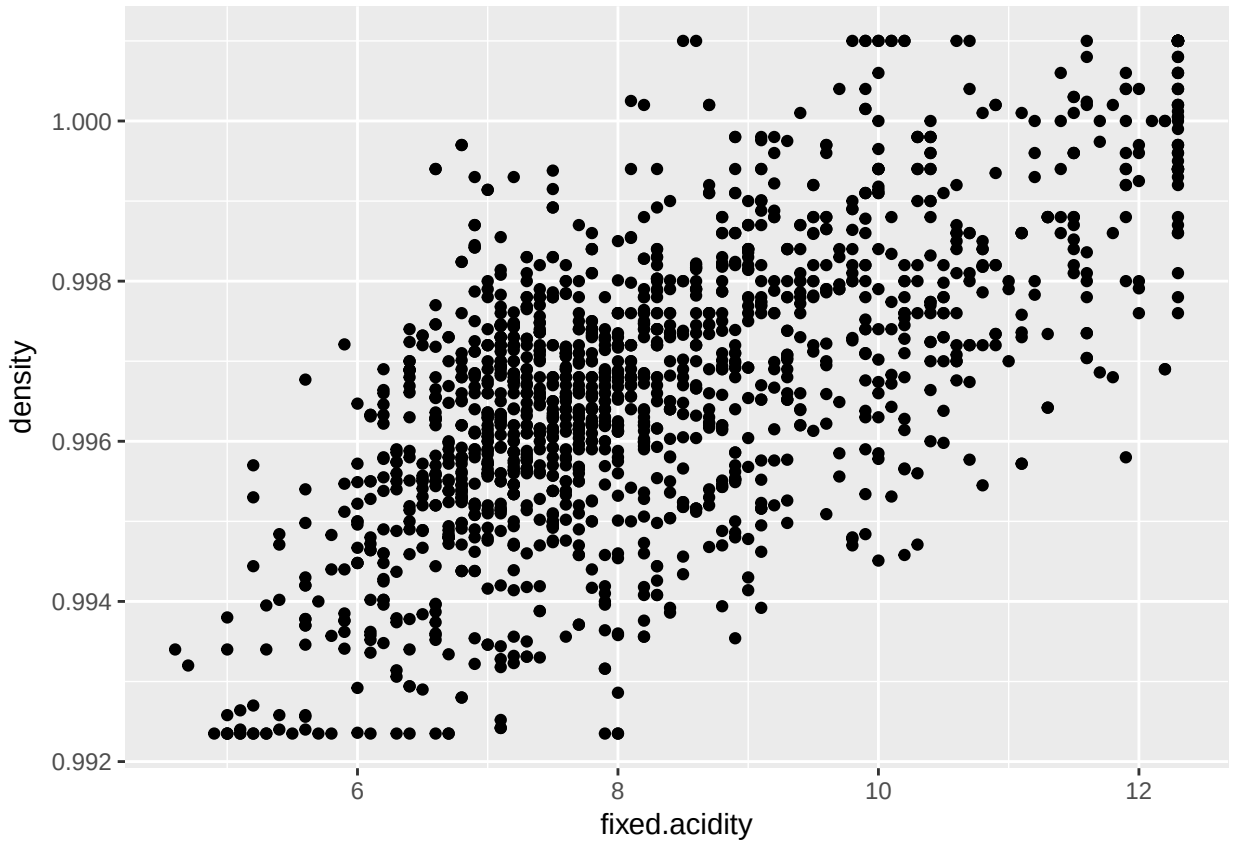
these correlations now.

For `fixed.acidity` we have

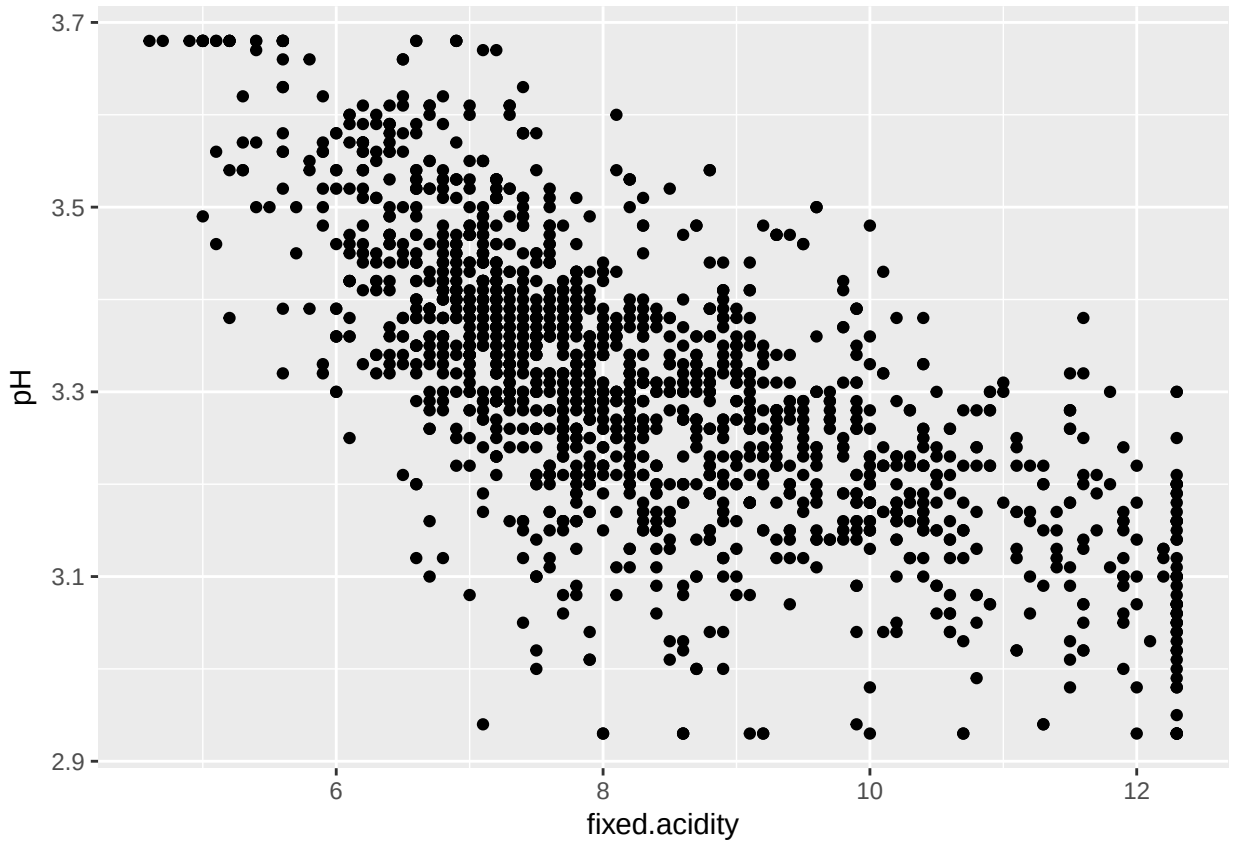
```
ggplot(data=df, aes(x = fixed.acidity, y = citric.acid)) + geom_point()
```




```
ggplot(data=df, aes(x = fixed.acidity, y = density)) + geom_point()
```

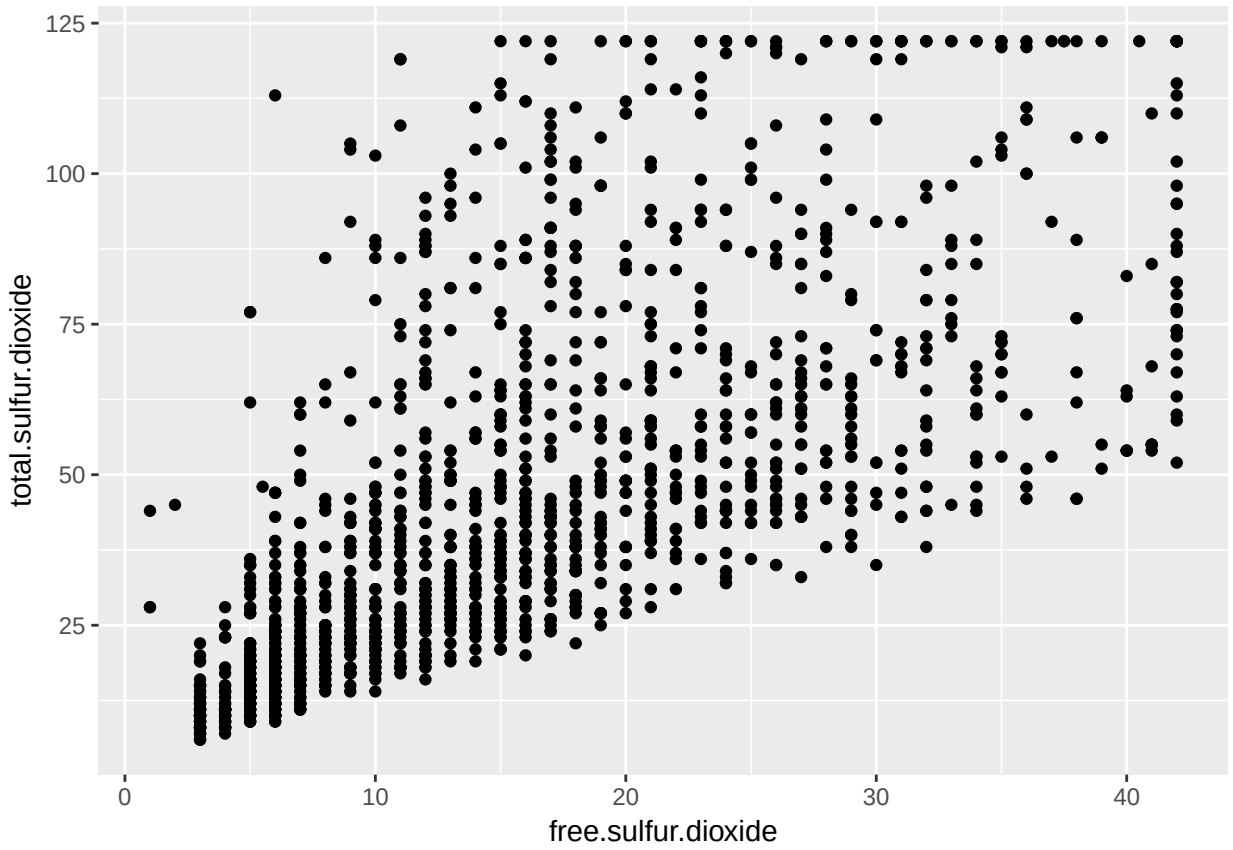



```
ggplot(data=df, aes(x = fixed.acidity, y = pH)) + geom_point()
```



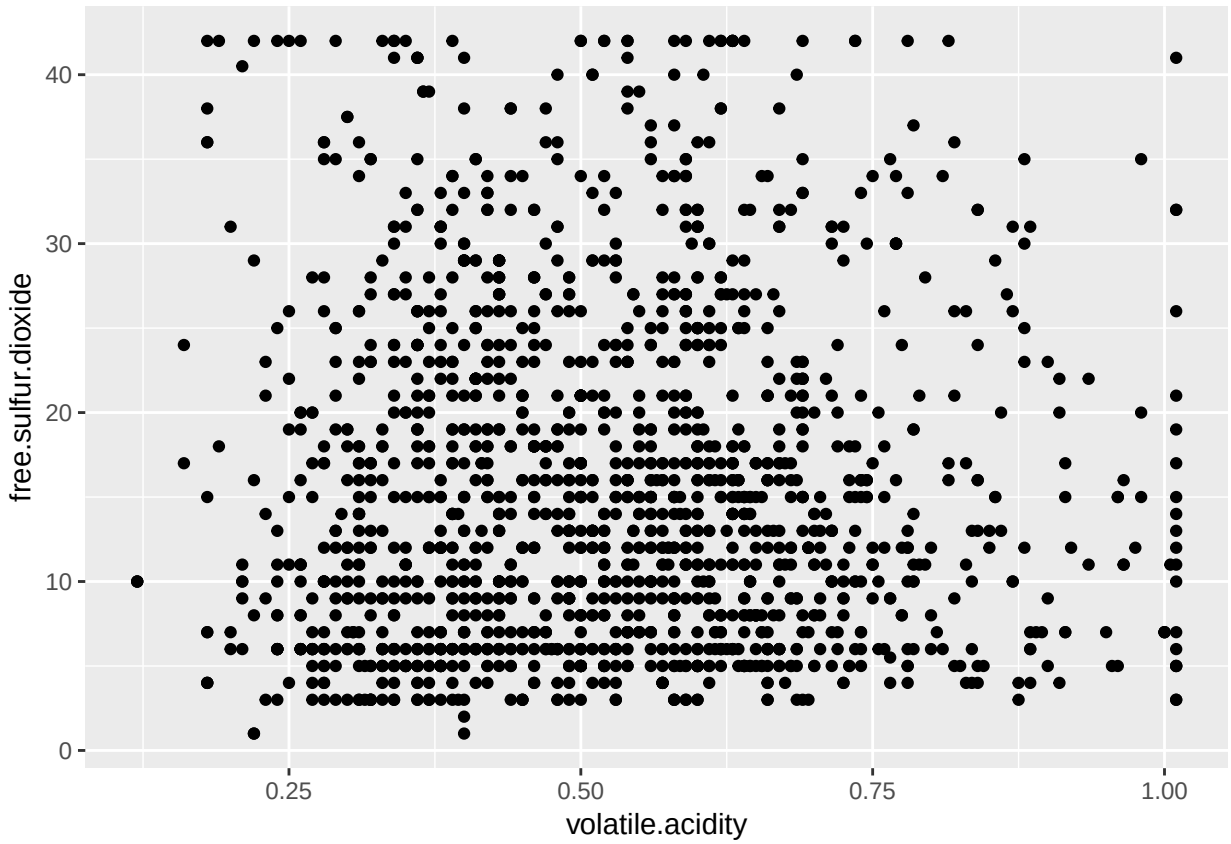
And for `free.sulfur.dioxide` we have

```
ggplot(data=df, aes(x = free.sulfur.dioxide, y = total.sulfur.dioxide)) + geom_point()
```



In order to look at these visualizations with some perspective, we will now plot two variables with a very low correlation.

```
ggplot(data=df, aes(x = volatile.acidity, y = free.sulfur.dioxide)) + geom_point()
```



We now update our dataframe by removing the two variables identified by the previous test.

```
df <- df[,-highlyCorrelated]
head(df)
```

```
##   volatile.acidity citric.acid residual.sugar chlorides
## 1             0.70        0.00            1.9     0.076
## 2             0.88        0.00            2.6     0.098
## 3             0.76        0.04            2.3     0.092
## 4             0.28        0.56            1.9     0.075
## 5             0.70        0.00            1.9     0.076
## 6             0.66        0.00            1.8     0.075
##   total.sulfur.dioxide density   pH sulphates alcohol quality
## 1                   34  0.9978 3.51      0.56     9.4       5
## 2                   67  0.9968 3.20      0.68     9.8       5
## 3                   54  0.9970 3.26      0.65     9.8       5
## 4                   60  0.9980 3.16      0.58     9.8       6
## 5                   34  0.9978 3.51      0.56     9.4       5
## 6                   40  0.9978 3.51      0.56     9.4       5
```

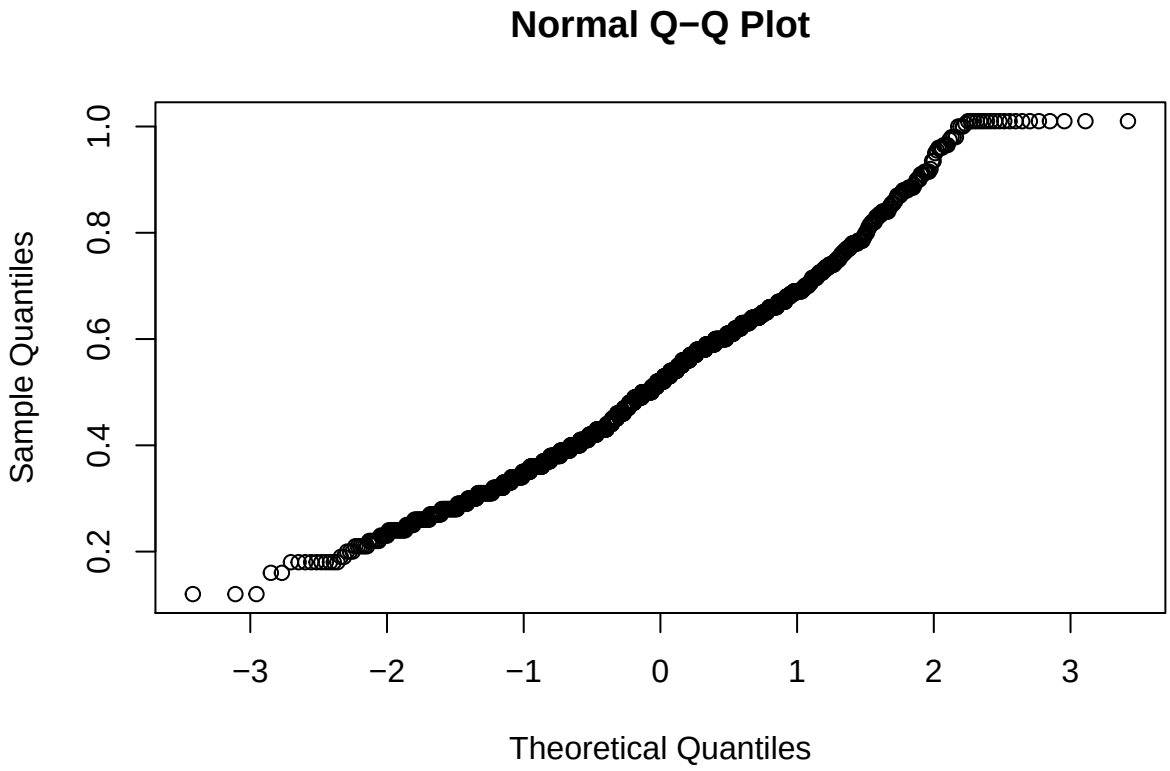
4.2 Normality assumptions and homogeneity of variance. Statistical tests

As a further step in our analysis we will be ultimately interested in performing a linear regression in order to model the quality of the wine with respect to the independent variables selected in the previous step. Although the computation of the best regression line does not require any particular statistical hypothesis about the distribution of our variables, should we want to provide confidence intervals for the regression coefficients the story is different. And in that situation we would be assuming that the variables follow a

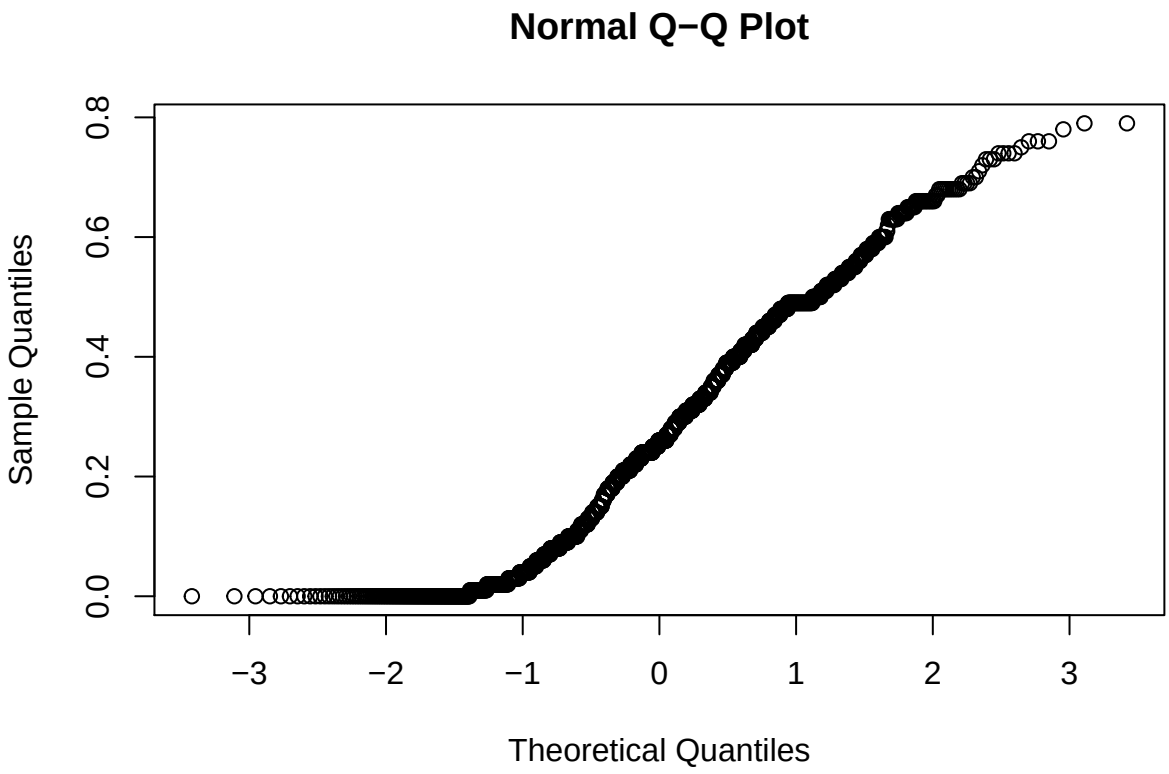
normal distribution. Therefore in this paragraph we are going to test for that hypothesis using Q-Q plots which gives a visual indication of the normality of a population. In our case, if a given variable is normally distributed (not necessarily standardized) we should see a straight line.

```
for (i in 1:ncol(df)){
  print(colnames(df)[i])
  qqnorm(df[,i])
}
```

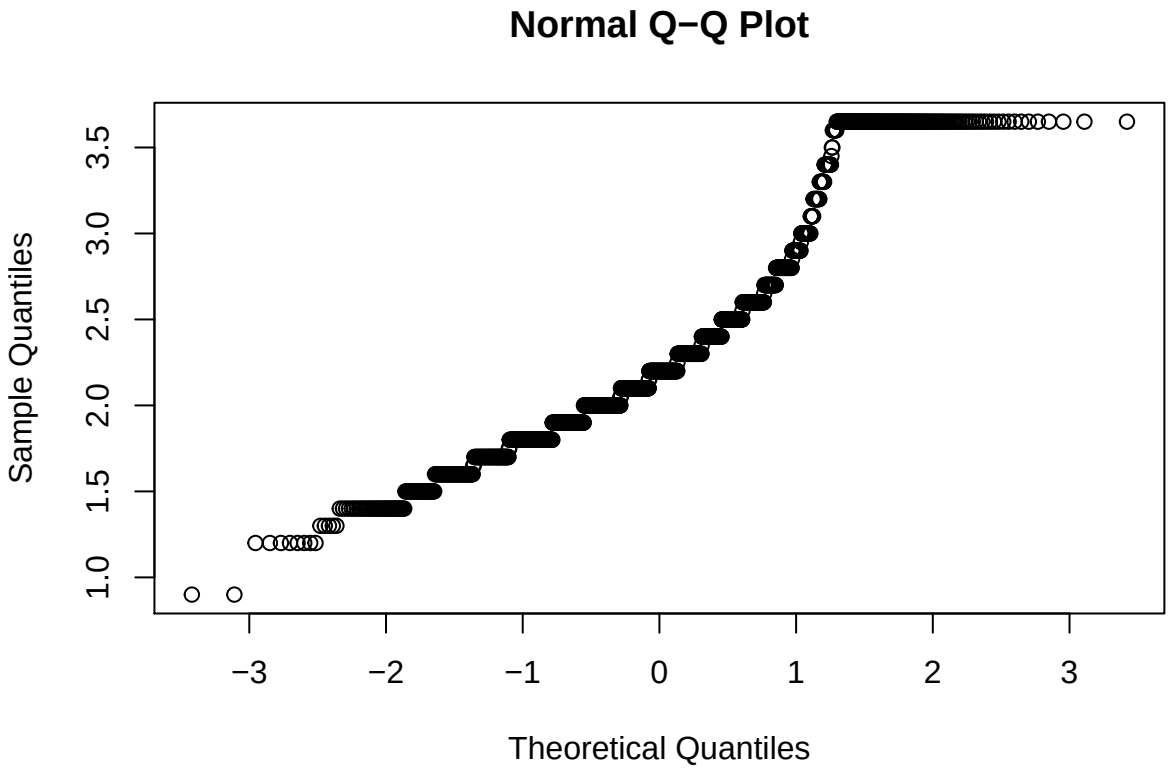
```
## [1] "volatile.acidity"
```



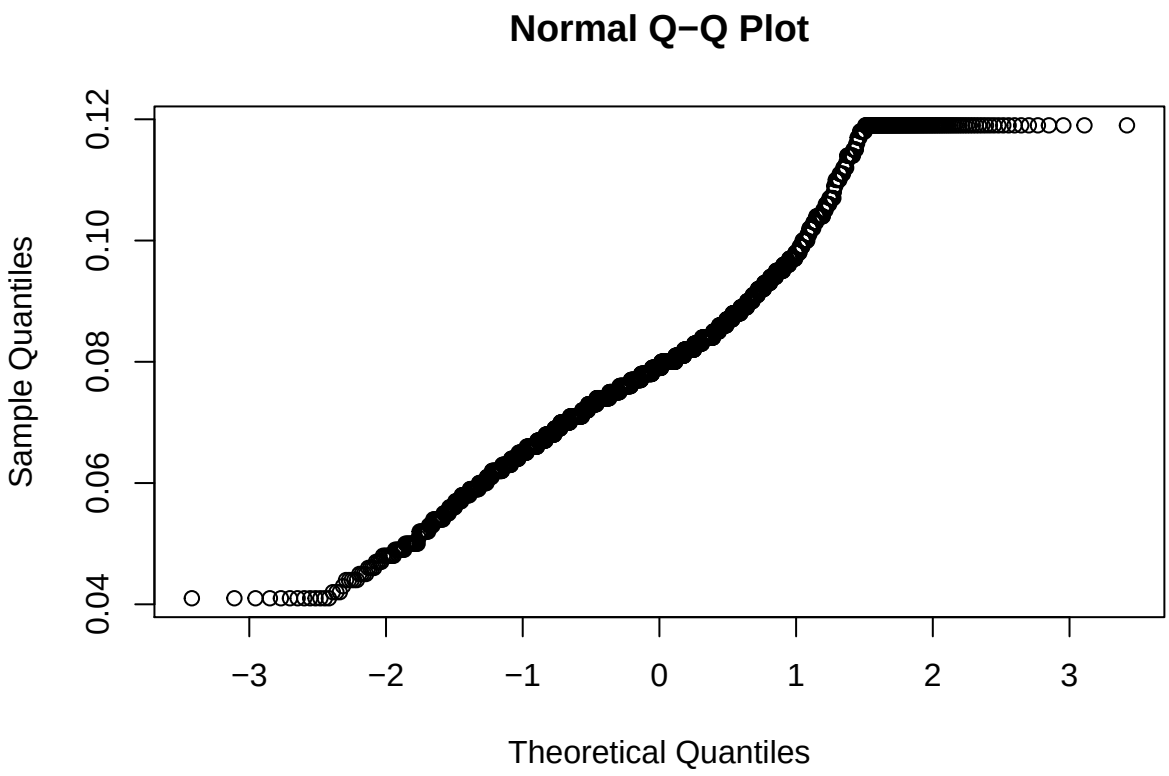

```
## [1] "citric.acid"
```



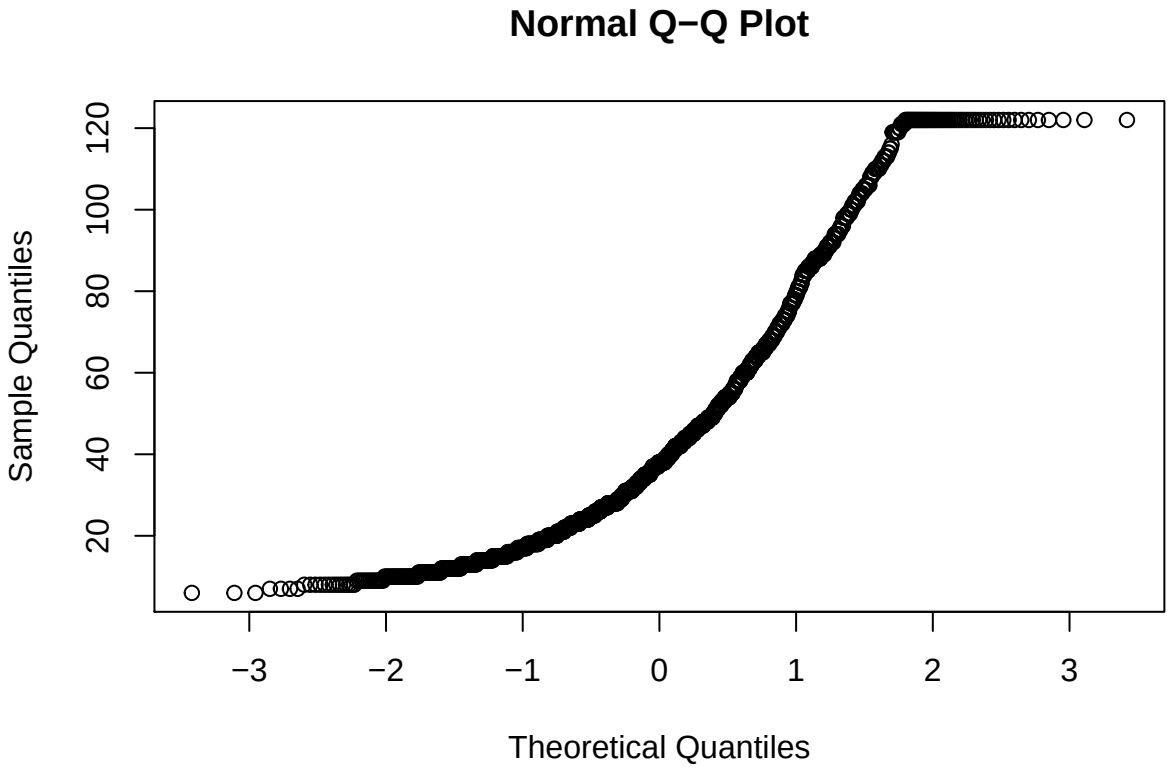

```
## [1] "residual.sugar"
```



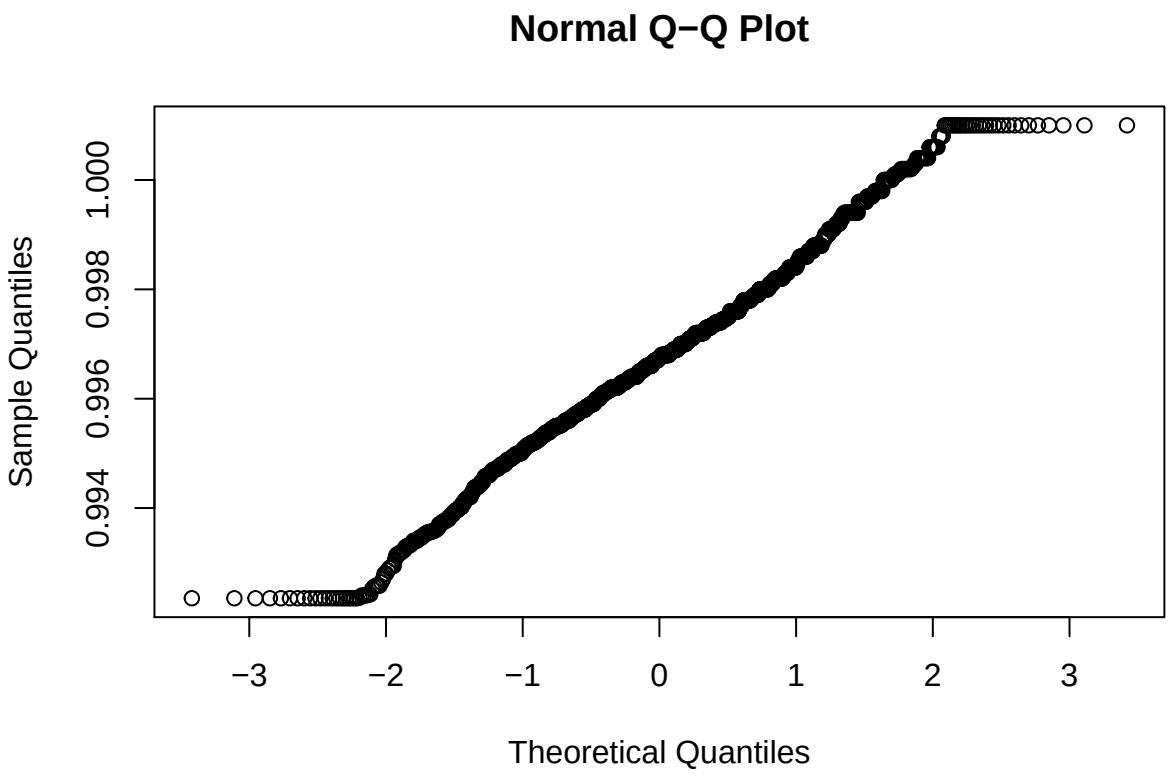

```
## [1] "chlorides"
```



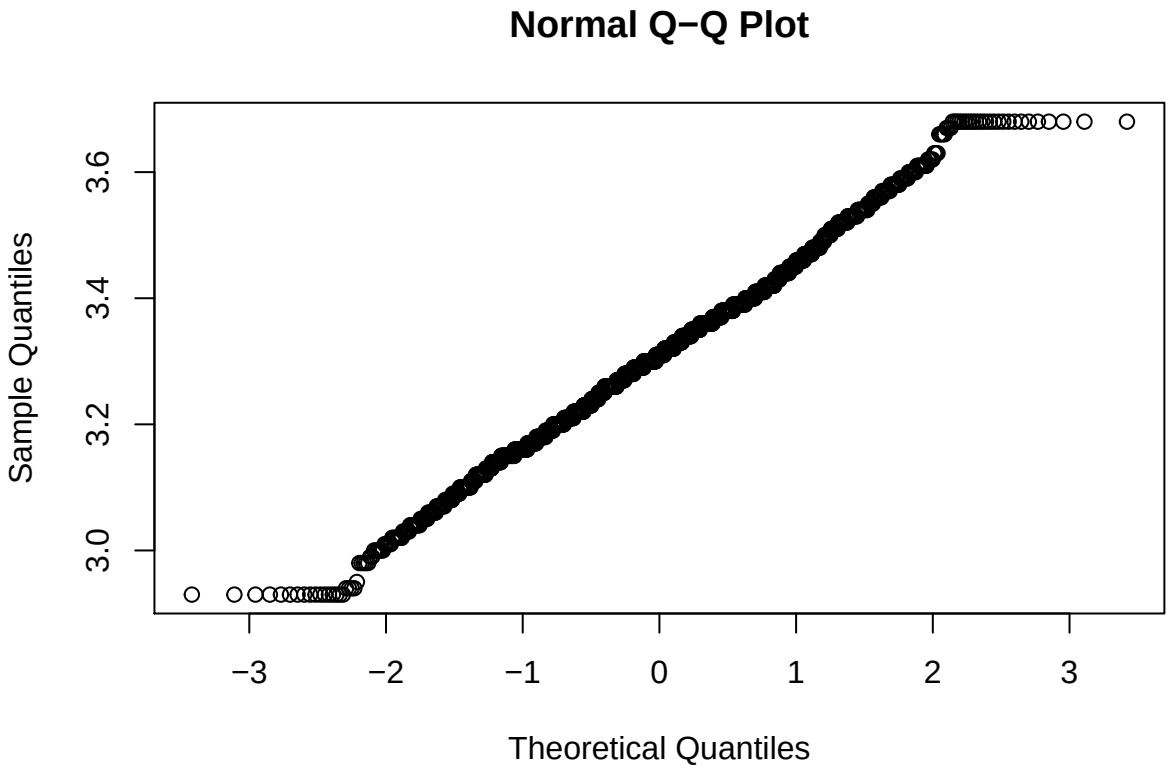

```
## [1] "total.sulfur.dioxide"
```



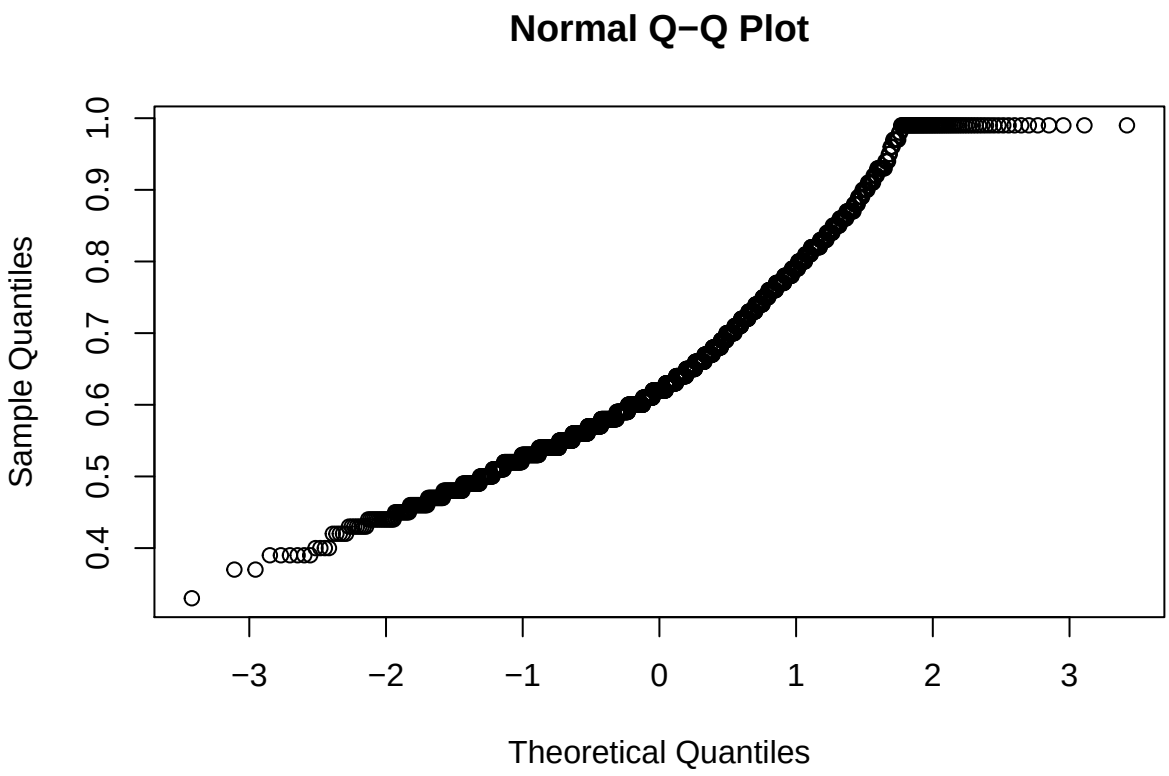

```
## [1] "density"
```



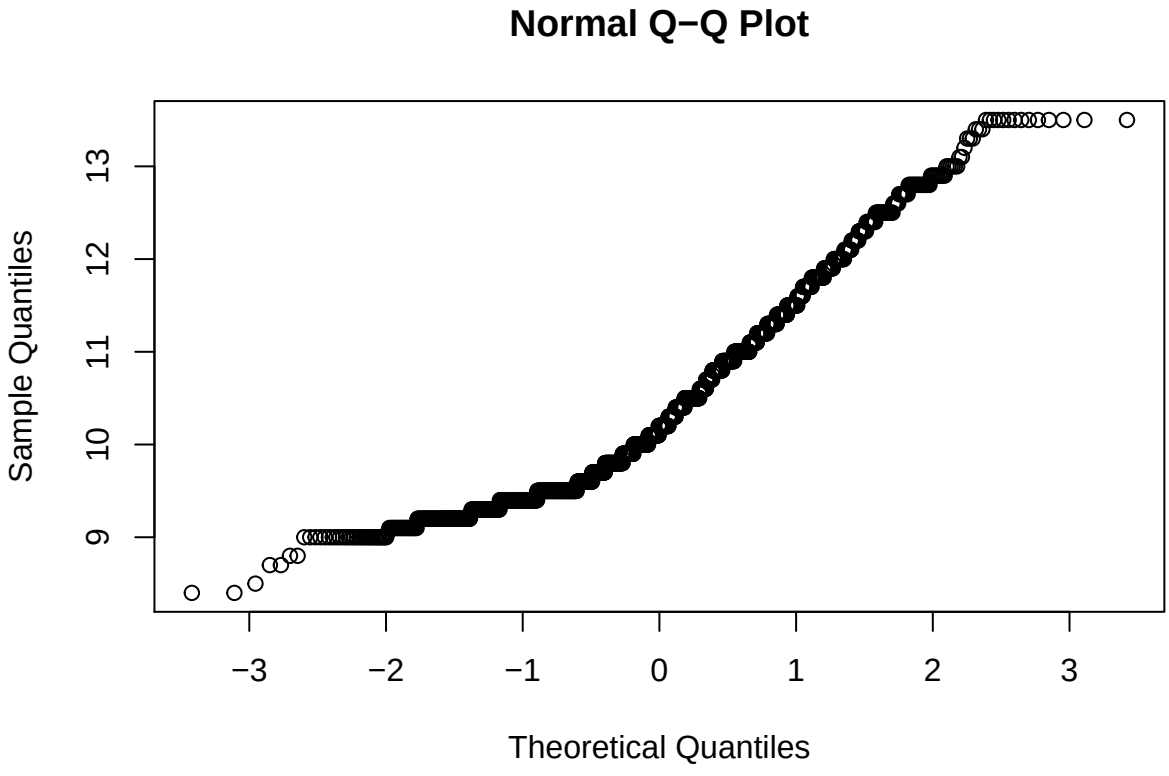

```
## [1] "pH"
```



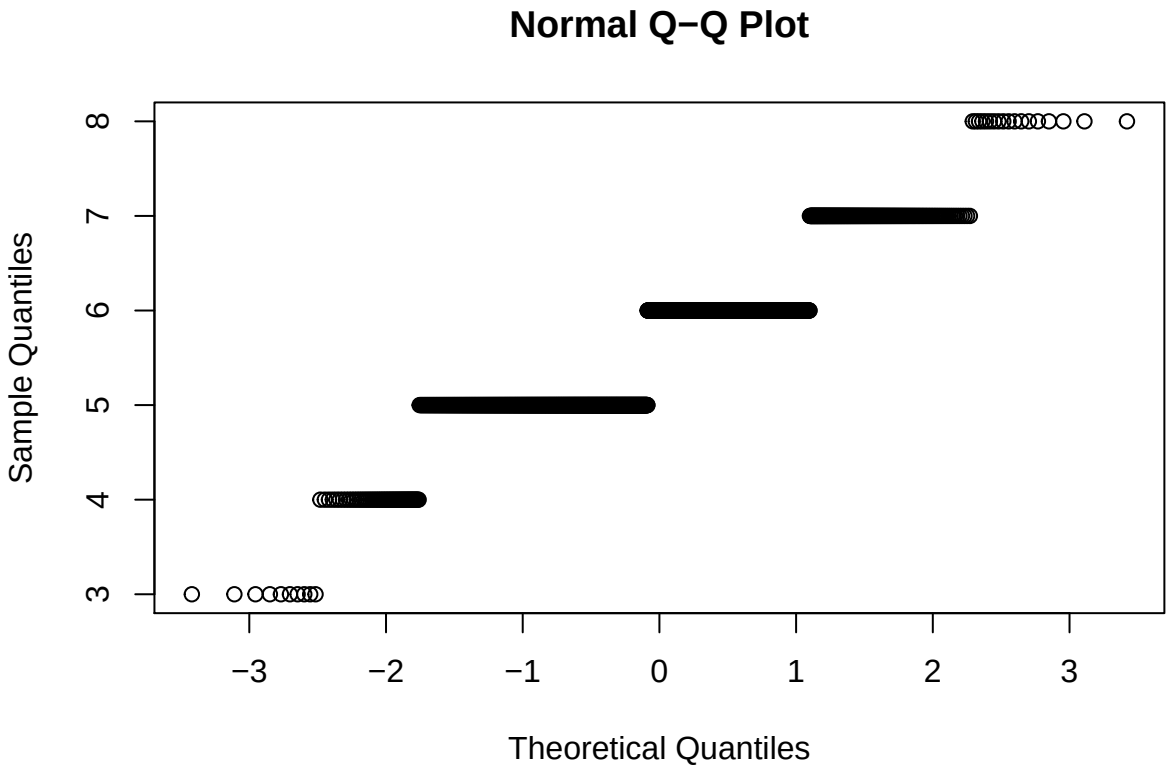

```
## [1] "sulphates"
```




```
## [1] "alcohol"
```




```
## [1] "quality"
```



We see that most variables follow a distribution close to normality within the intermediate range of their values, which is a common behavior. Exceptions to this are `residual.sugar` and `total.sulfur.dioxide`.

As a possible interesting question about this particular dataset, we can check if there is a significant difference in the levels of alcohol between the highest and lowest rated wines, and see if this level is in average the same. To this end, we will perform a 2-sample T test. We will start by subsetting our dataframe in order to get a smaller one only including those wines with quality rankings of 3 or and 7 or 8, which are the two lowest and two highest levels that we have.

```
df_small = df[df$quality == 3 | df$quality == 4 | df$quality == 7 | df$quality == 8,]
head(df_small)
```

```
##    volatile.acidity citric.acid residual.sugar chlorides
## 8              0.65        0.00           1.20     0.065
## 9              0.58        0.02           2.00     0.073
## 17             0.28        0.56           1.80     0.092
## 19             0.59        0.08           3.65     0.086
## 38             0.38        0.28           2.10     0.066
## 39             1.01        0.09           1.50     0.119
##    total.sulfur.dioxide density   pH sulphates alcohol quality
## 8                    21  0.9946 3.39      0.47    10.0       7
## 9                    18  0.9968 3.36      0.57     9.5       7
## 17                  103  0.9969 3.30      0.75    10.5       7
## 19                   29  0.9974 3.38      0.50     9.0       4
## 38                   30  0.9968 3.23      0.73     9.7       7
## 39                   19  0.9940 3.50      0.48     9.8       4
```

We will now create a label with two values 'L' or 'H' corresponding to the low and high levels.

```
df_small$label[df_small$quality == 3 | df_small$quality == 4] = 'L'
df_small$label[df_small$quality == 7 | df_small$quality == 8] = 'H'
head(df_small)
```

```
##    volatile.acidity citric.acid residual.sugar chlorides
## 8              0.65        0.00           1.20     0.065
## 9              0.58        0.02           2.00     0.073
## 17             0.28        0.56           1.80     0.092
## 19             0.59        0.08           3.65     0.086
## 38             0.38        0.28           2.10     0.066
## 39             1.01        0.09           1.50     0.119
##    total.sulfur.dioxide density   pH sulphates alcohol quality label
## 8                    21  0.9946 3.39      0.47    10.0       7     H
## 9                    18  0.9968 3.36      0.57     9.5       7     H
## 17                  103  0.9969 3.30      0.75    10.5       7     H
## 19                   29  0.9974 3.38      0.50     9.0       4     L
## 38                   30  0.9968 3.23      0.73     9.7       7     H
## 39                   19  0.9940 3.50      0.48     9.8       4     L
```

We now compare the means of the `alcohol` variable according to the L and H groups. The null hypothesis will be that these means are equal, and therefore the amount of alcohol in low quality and high quality wines are the same. The alternative hypothesis is that these means are different. Notice that

1. In order to apply this test, which is a parametric test, we are assuming normality of the distributions and
2. This is a bilateral contrast, and therefore, if the null hypothesis is rejected the best thing we can say is that both means are significantly different, without distinguishing which one is higher.

```
t.test(df_small$alcohol~df_small$label)
```

```
##
##  Welch Two Sample t-test
##
## data:  df_small$alcohol by df_small$label
## t = 9.6963, df = 106.35, p-value = 2.627e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.029087 1.558061
## sample estimates:
## mean in group H mean in group L
##        11.50945        10.21587
```

Since the confidence interval for the difference of means does not contain 0, we find that at a 0.05 significance level the means are not equal for low and high quality wines.

As for the final statistical test that we will apply to this dataset, we are going to study the homogeneity of the variance for the variable `alcohol` among the groups given by the quality scores. To this end, we will apply the Fligner-Killeen test. As for the previous case, the null hypothesis will be that the variances are equal, and the alternative hypothesis that they are not.

```
fligner.test(alcohol~quality, data = df)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  alcohol by quality
## Fligner-Killeen:med chi-squared = 131.49, df = 5, p-value <
## 2.2e-16
```

Since the p-value obtained is smaller than 0.05 we can conclude that the variances are not homogeneous among all quality groups at a 0.05 significance level.

4.3 A linear regression model.

Finally we are going to train a regression model for the full dataset in order to obtain the best possible linear function that predicts the dependent variable taking as arguments the surviving variables after the feature selection process. For this, we use the `lm` function implemented in base `R`.

```
model <- lm(quality ~ ., data = df)
summary(model)
```

```
##
## Call:
## lm(formula = quality ~ ., data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.60961 -0.36133 -0.03315  0.42583  1.92336
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -0.030855  15.280699  -0.002  0.99839
## volatile.acidity     -1.098308   0.122229  -8.986  < 2e-16 ***
## citric.acid          -0.302375   0.136345  -2.218  0.02672 *
## residual.sugar        0.016144   0.033990   0.475  0.63489
## chlorides            -3.429585   1.043664  -3.286  0.00104 **
## total.sulfur.dioxide -0.002400   0.000569  -4.218 2.61e-05 ***
## density               4.815427  15.277908   0.315  0.75266
## pH                   -0.609663   0.136720  -4.459 8.80e-06 ***
## sulphates             1.241553   0.133012   9.334  < 2e-16 ***
## alcohol               0.296507   0.023636  12.545  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6443 on 1589 degrees of freedom
## Multiple R-squared:  0.367,  Adjusted R-squared:  0.3634
## F-statistic: 102.4 on 9 and 1589 DF,  p-value: < 2.2e-16
```

The p-values obtained show that the significant (non-zero coefficients) for this linear model at at least a 0.05 significant level are those marked with stars. As we see, this regression line does not use all the variables and in particular there is no intercept (constant) term.

The total variance explained by the model is given by the adjusted R^2 coefficient which has a value of 0.3651. This is not a very good fit since the maximum value is 1. In our case, this coefficient indicates that 36.51% of the total variance of the dataset is explained by this model. We will now perform a prediction on the full dataset and compute the RMSE error.

```
pred <- predict(model, df[,1:(ncol(df)-1)])
res <- pred - df$quality
RMSE <- sqrt((1/nrow(df))*sum( res^2   ))
RMSE
```

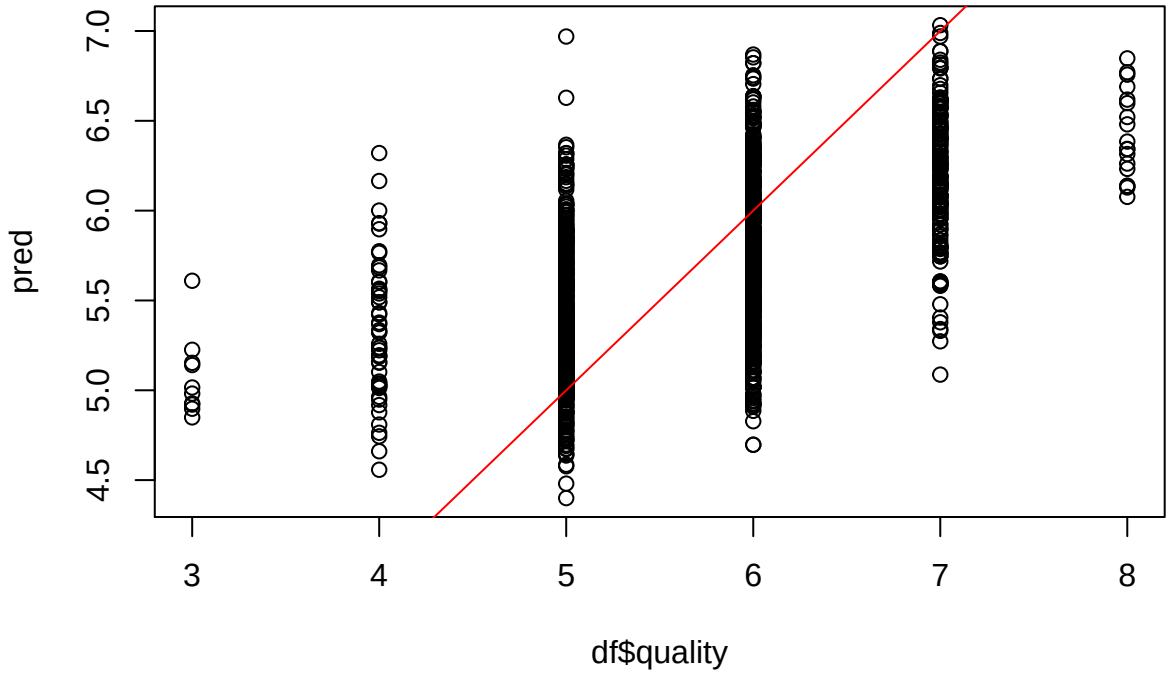
```
## [1] 0.6423209
```

This means that the typical error made by using this linear model is of 0.64, which is slightly less than one quality score point, since the RMSE value is expressed in the same units as the dependent variable.

5 Graphic representation of the results.

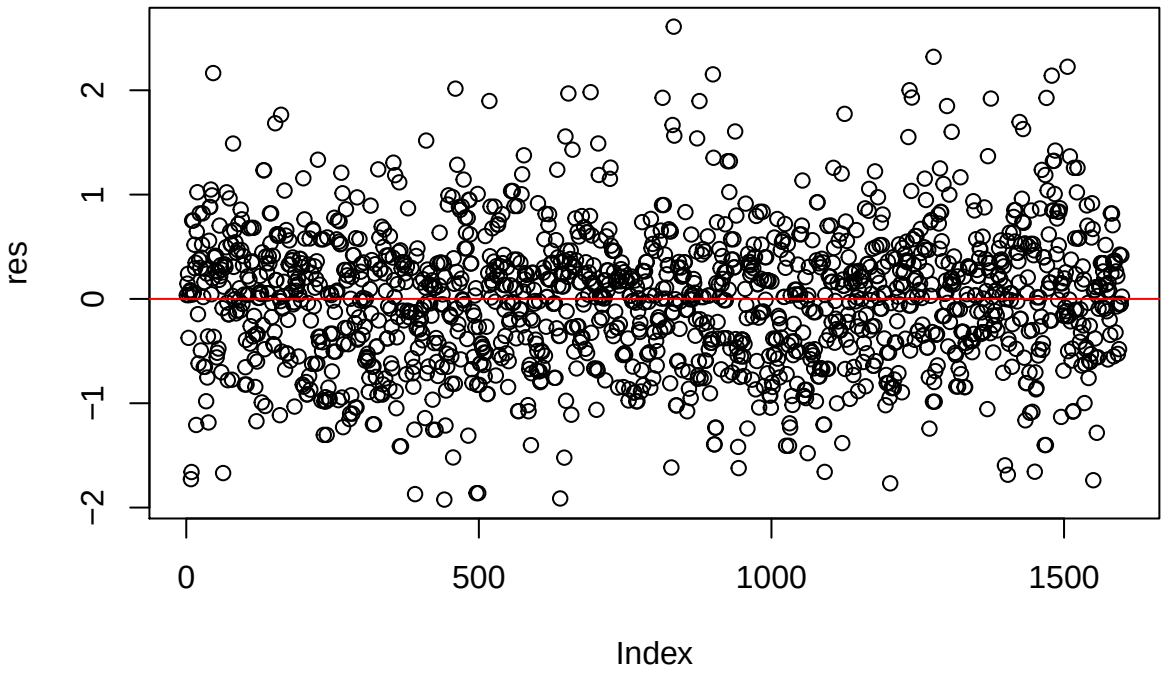
We will now use two different graphical indicators to visually assess the quality of the fit for this linear model. First, we plot the actual values of `quality` versus the line $y = x$, which correspond to a perfect fit. All points outside this line are errors of the predictions of the model. The larger the vertical distance from a point to the line, the higher the error made.

```
plot(df$quality, pred)
abline(a=0, b=1, col = 'red')
```



Next, we are going to plot the residuals, that is, the differences of the predicted values and the actual values. This visualization should show a pattern without any particular structure, and symmetric with respect to the horizontal line $y = 0$, meaning that the residuals have mean 0. That is, they are random noise.

```
plot(res)
abline(a=0, b=0, col = 'red')
```



We can actually compute this mean, which is equal to

```
mean(res)
```

```
## [1] -5.857311e-14
```

This negligible figure, together with the plot of the residuals show that, indeed, the residuals are random noise.

6 Solution to the problem.

We have cleaned the `winequality-red` dataset, in particular treating the outliers. We have not found zero or not informed variables. Afterwards, we have performed a feature selection analysis by removing highly correlated variables in order to drop redundant information.

Since the goal of this analysis was to establish the best (multi-)linear relationship among the independent variables and the `quality` score, we have tested for the normality assumption inherent to the linear regression procedure.

We have also performed statistical tests on the average alcohol amount in wines of low and high quality, as well as on the variances of the distribution of alcohol levels among the different quality values. Finally we have trained a linear model, obtaining a rather modest result, which suggests that the functional relationship in this problem, if exists, is more complicated than a linear function. Here we should stress the fact that for this particular problem, the quality score is a rather subjective quantity, and this fact goes along the lines of the results obtained.

We have investigated the fit of the model by making predictions on the dataset and comparing them to the actual values for the `quality` score, and also studied the distribution and mean of the residuals.