STK-IN4300 oblig 1

anettfre

Autumn 2020

Problem 1. Reporting

Regression analysis

Introduction

In this report I am going to analyse a dataset of white wine quality from UCI Machine Learning Repository. The wines in the dataset is from north in Portugal. The purpose of this analysis is to se if I can predict the quality of a white wine given these variable inputs. This can be used to know if a wine is of good quality whitout tasing it and also help to choose a good wine for non-wineexperts.

I will use backward elimination with Akaike information criterion, and also use lasso regression to se if I can predict the quality of a white wine.

The output variable in the dataset is quality, it is in a range from 0 to 10, where 10 is the best quality. In the dataset it is a lot of wines with quality 5 or 6, i.e normal wines and not many excellent or poor wines. This will probably make it hard to separate the good and bad quality wine. The quality from each wine in the dataset is found by sensory data. Since the quality is determined from sensory data I assume that multiple people (wine experts) have tasted the different wines and graded the quality. This might have made the dataset more unrelaiable given that different people might have (minor) different opinion of the quality of a wine and this makes it more difficult for the model to predict the right quality.

The covatiates in the dataset is based on physicochemical tests. These are the different vaiables:

1 - fixed acidity, 2 - volatile acidity, 3 - citric acid, 4 - residual sugar, 5 - chlorides, 6 - free sulfur dioxide, 7 - total sulfur dioxide, 8 - density, 9 - pH, 10 - sulphates, 11 - alcohol,

First we set a seed to make results reproducible. We will also look at the different covariates.

```
library(ggplot2)
set.seed(1111)
white_wine = read.csv("winequality-white.csv", sep=";", header=TRUE)
head(white_wine)
```

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

```
## 5                  47                   186  0.9956 3.19      0.40     9.9
## 6                  30                    97  0.9951 3.26      0.44    10.1
##   quality
## 1       6
## 2       6
## 3       6
## 4       6
## 5       6
## 6       6
```

```r
summary(white_wine)
```

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

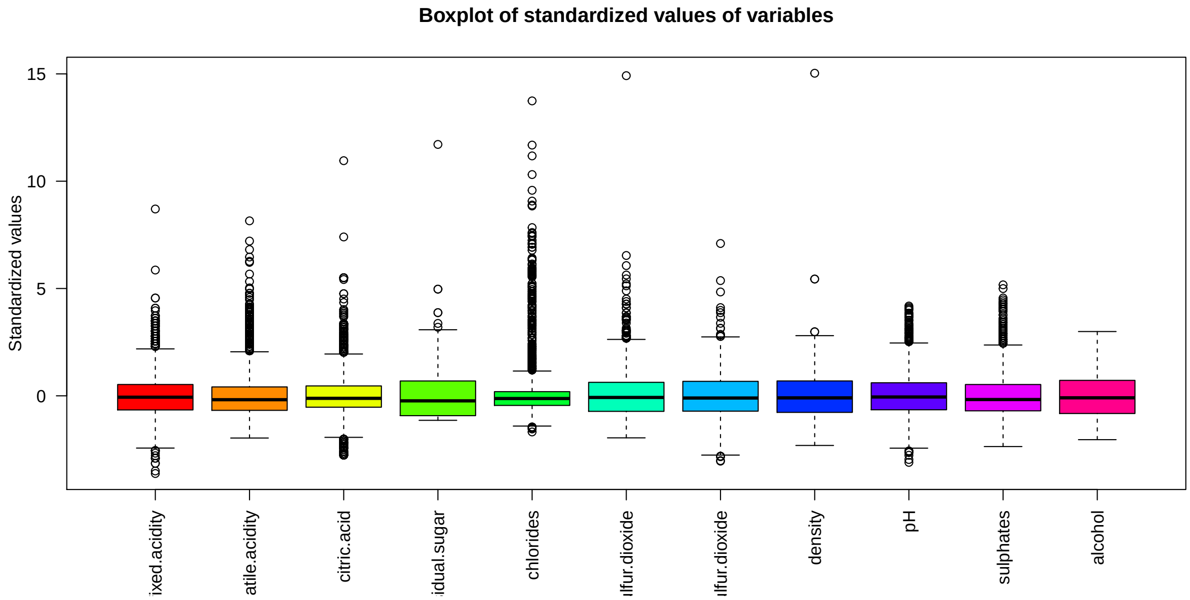
```r
dim(white_wine)
```

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

```r
which(is.na(white_wine))
```

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

```r
#the response variable
y <- white_wine[, 12]
#the explanatory variables
X <- white_wine[, 1:11]
boxplot(scale(X),las = 2, col=rainbow(length(unique(X))), main="Boxplot of standardized values of varia
mtext("Standardized values", side = 2, line = 2)
```



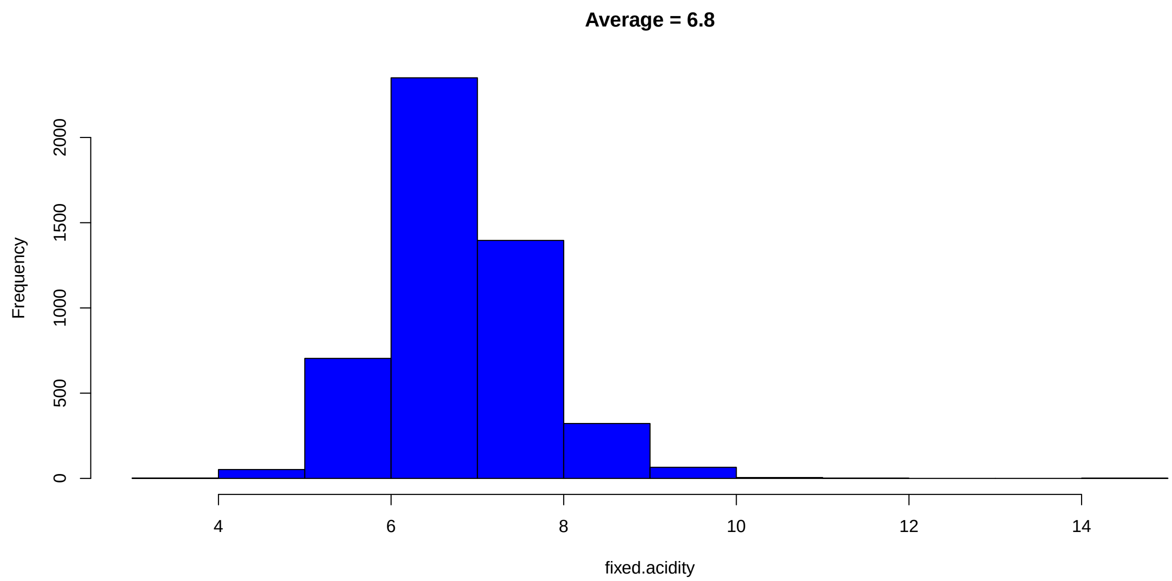

```
#mtext("Variables", side = 1, line = 7)
```

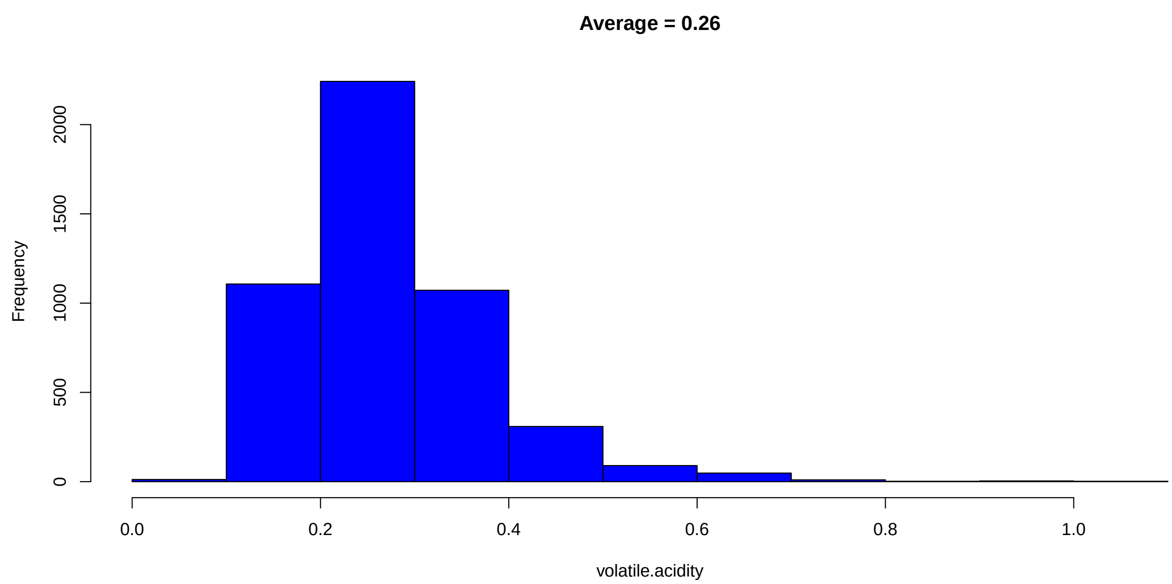
Information about the dataset

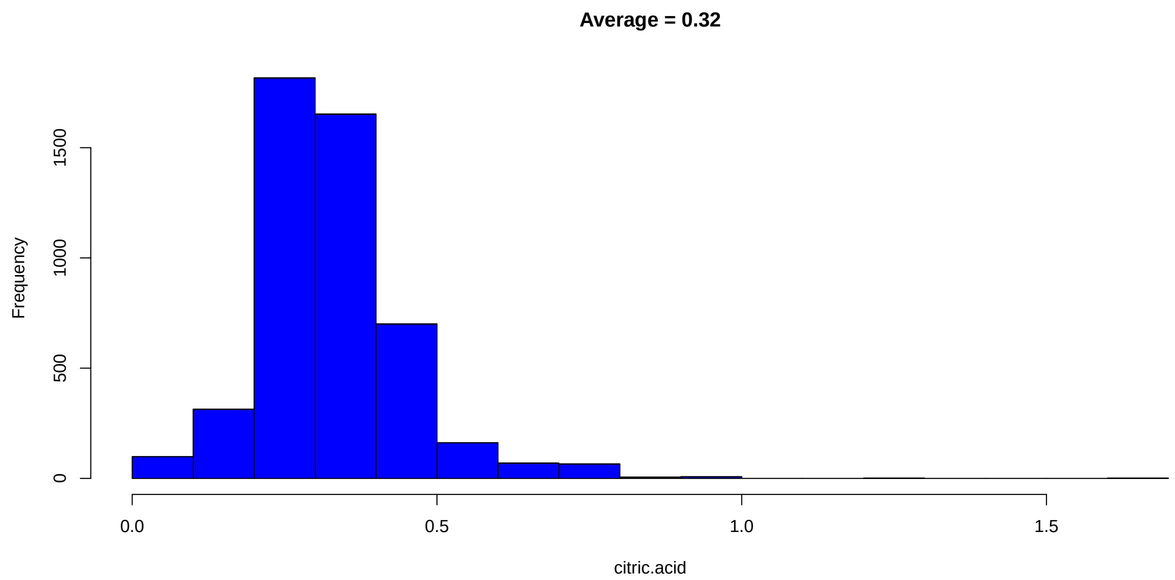
The dimension of the dataset is 4898 x 12, that means that the dataset has 4898 samples, 11 explanatory vaiables and a response vaiable, y, that is the quality of the wine. From the boxplot I see som points that is far from the average, these might be outliers, for this analysis I don't do something with them. I looked for missing values with is.na, there is not any missing values.

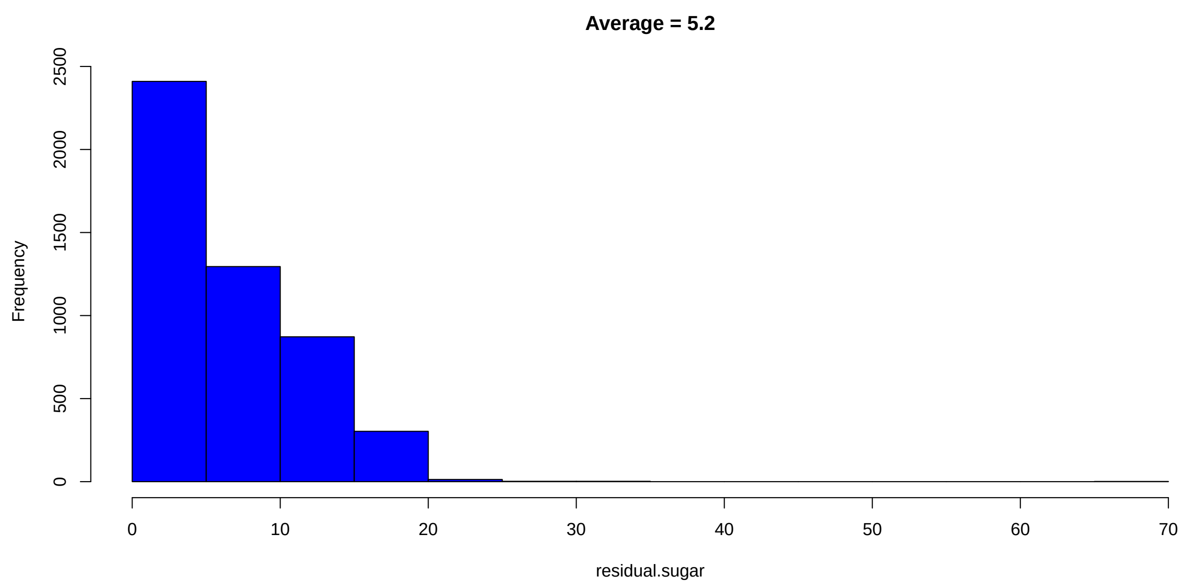
To look further on the covariates I plot a histogram of each of them and calculate the average I also do it for the quality. This gives a good visual presentation of the distribution of the covariates.

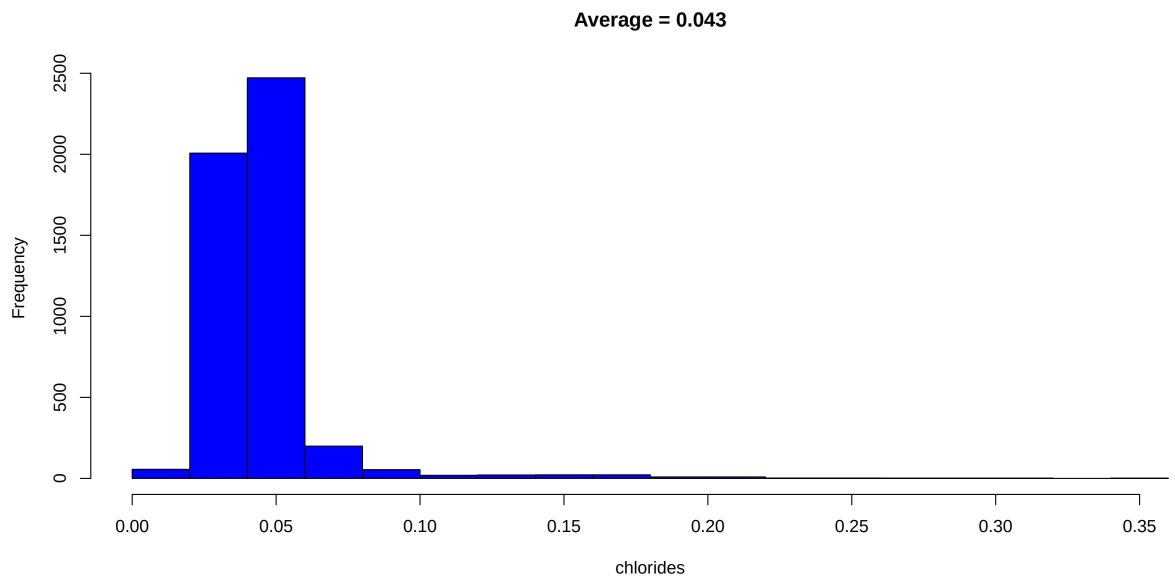
```
for (i in 1:12) {
  hist(white_wine[[i]], xlab = names(white_wine)[i],  col = "blue", main = paste("Average =",  mean(whi
}
```

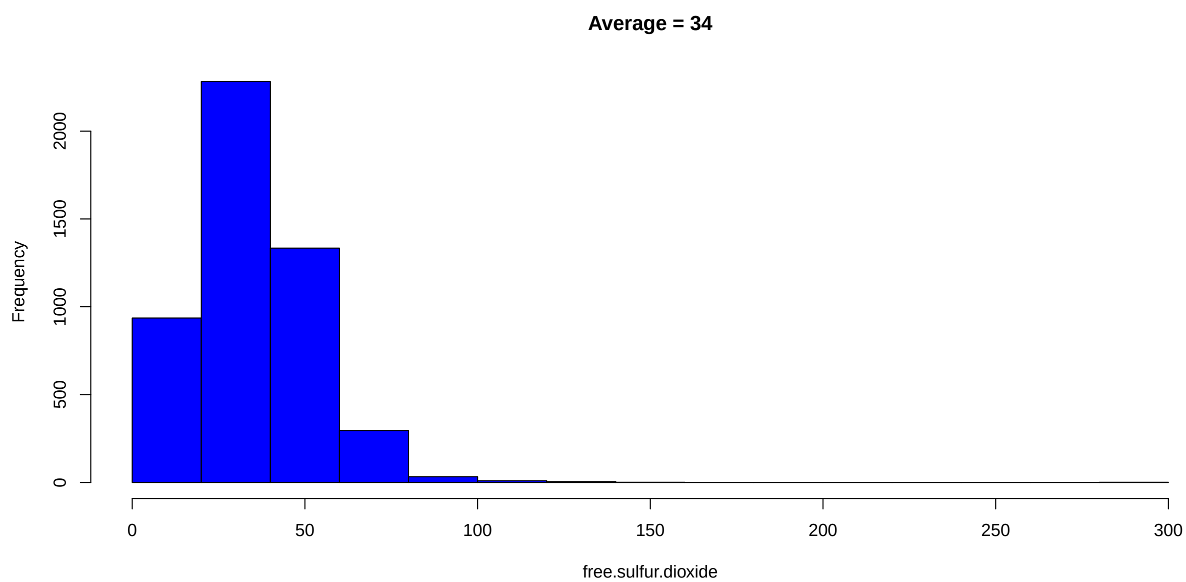

Average = 6.8
Frequency
0
500
1000
1500
2000
4
6
8
10
12
14
fixed.acidity

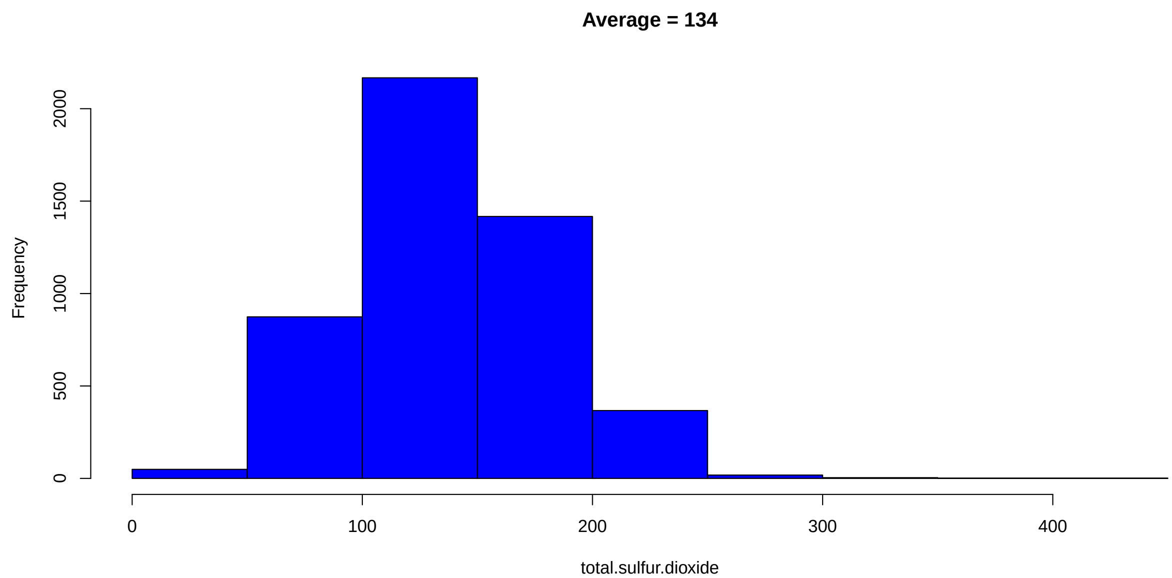

Average = 0.26
Frequency
0
500
1000
1500
2000
0.0
0.2
0.4
0.6
0.8
1.0
volatile.acidity

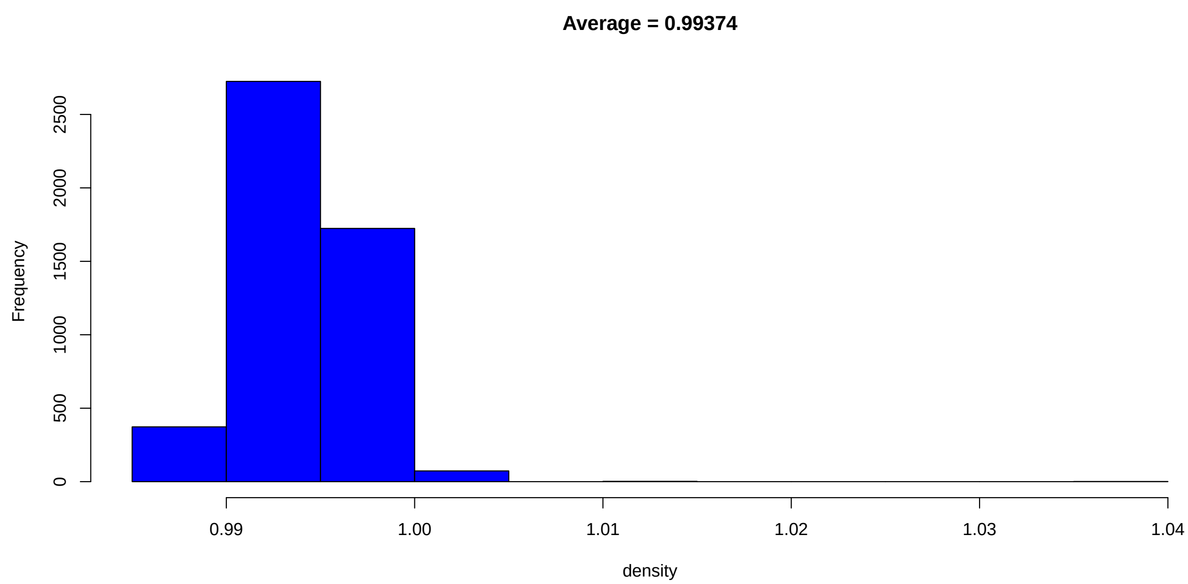

Average = 0.32
Frequency
0
500
1000
1500
0.0
0.5
1.0
1.5
citric.acid

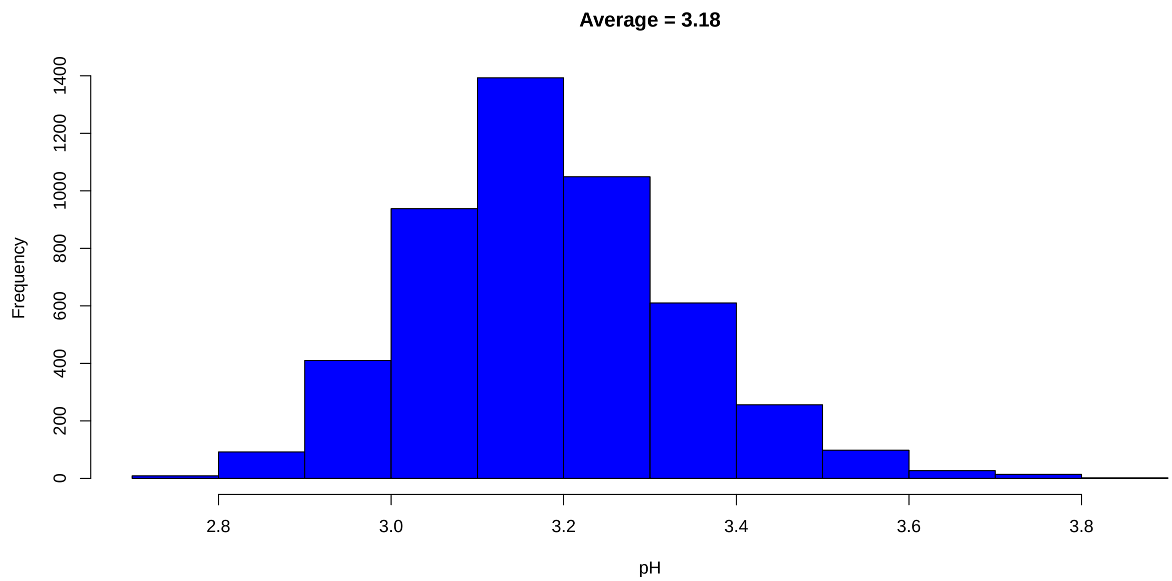

Average = 5.2
Frequency
0
500
1000
1500
2000
2500
0
10
20
30
40
50
60
70
residual.sugar


Average = 0.043
Frequency
0
500
1000
1500
2000
2500
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
chlorides

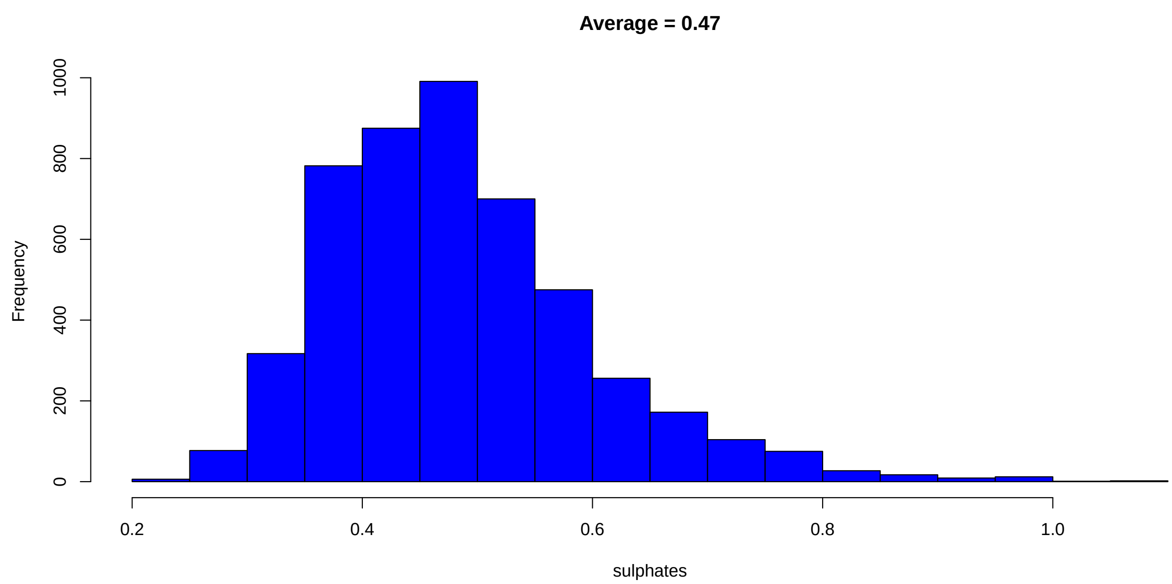

Average = 34
Frequency
0
500
1000
1500
2000
0
50
100
150
200
250
300
free.sulfur.dioxide


Average = 134
Frequency
0
500
1000
1500
2000
0
100
200
300
400
total.sulfur.dioxide

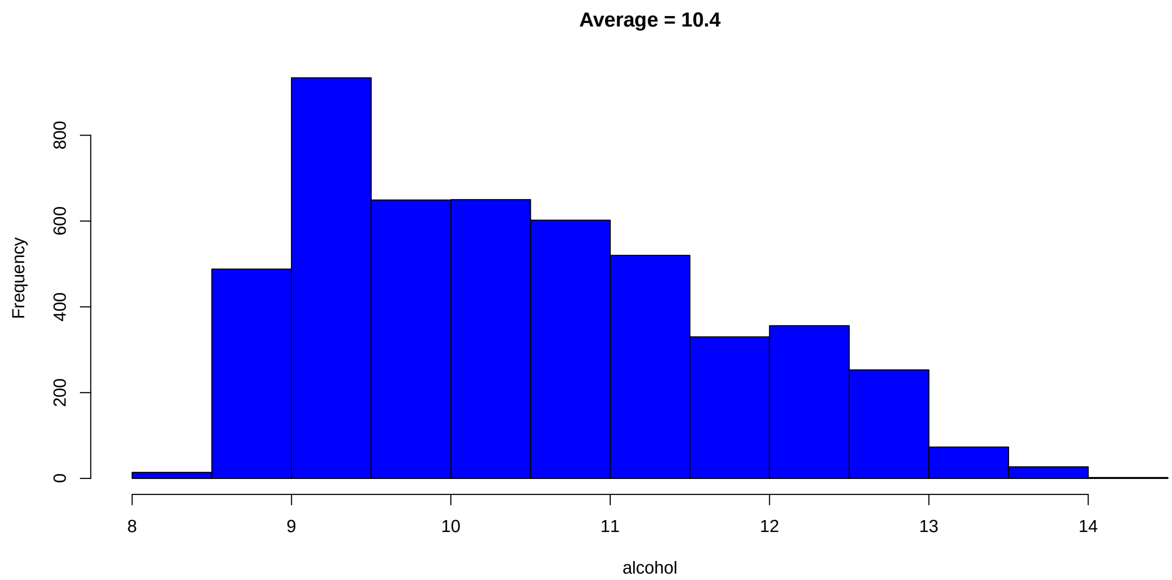

Average = 0.99374
Frequency
0
500
1000
1500
2000
2500
0.99
1.00
1.01
1.02
1.03
1.04
density

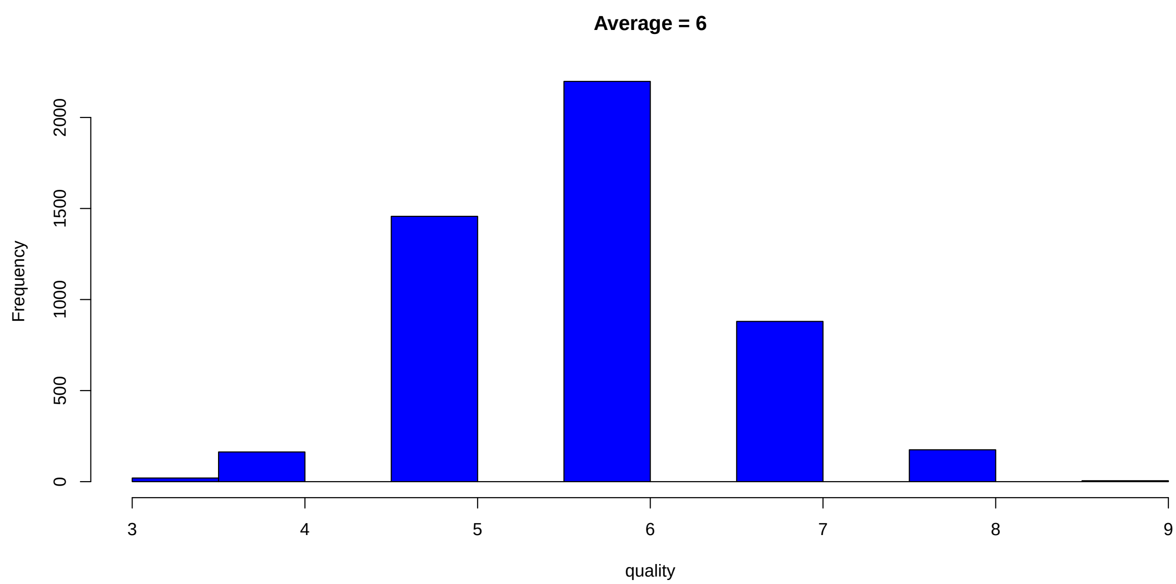


Average = 3.18
Frequency
0
200
400
600
800
1000
1200
1400
2.8
3.0
3.2
3.4
3.6
3.8
pH



Average = 0.47
Frequency
0
200
400
600
800
1000
0.2
0.4
0.6
0.8
1.0
sulphates

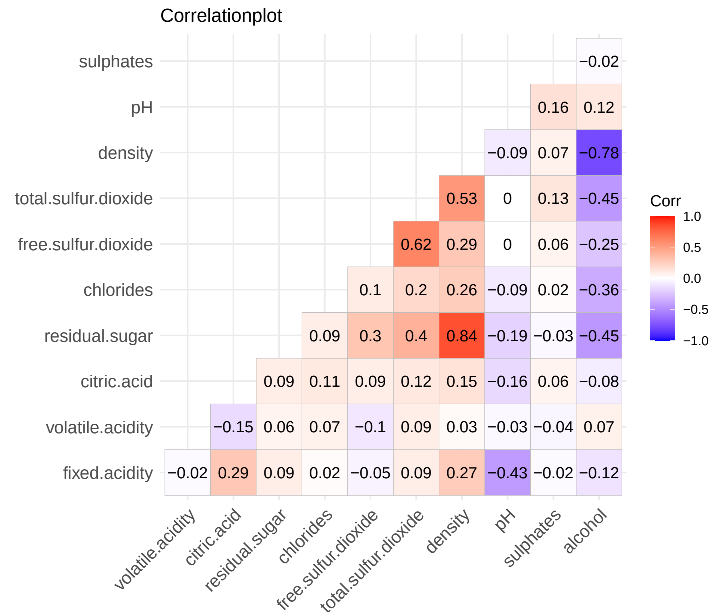




We can se from the histogram plots and the summary of the varables that: Fixed acidity has a average of 6.8 and there is almost a normal disribution of values. Volatile acidity has a average of 0.26 and is right-skewed. Citric acid has a average of 0.32 and is also right-skewed. Residual sugar has a average of 5.2 and we can se that most of the values is close to 0 and very few is over 20. Chlorides has a average of 0.043, most of the wines is between 0.00 and 0.10 but a few har chlorides from 0.10 to 0.20. Free sulfur dioxide has a average of 34. Total sulfur dioxide has a average of 134. Density has a average of 0.99, the wines values is distributed over a small range. PH has a average of 3.18. Sulphates has a average of 0.47. Alcohol has a average of 10.4, it is distributed over a longer intervall than the other variables. Quality has a average of 6 and in contrast to the explanatory variables this is not right-skewed, it is no values over 9 and under 3.

To compare the vaiables we scale the data.

```
library(ggcorrplot)
ggcorrplot(cor(scale(X)), lab = TRUE, type = "lower", title="Correlationplot")
```



Using ggcorrplot to visualise the correlation beween the explanitory vaiables. We can se that residual sugar and density, and also between density and alcohol have strong correlation. PH and total sulfur dioxide, pH and free sulfur dioxide has 0 correlation. Stronger correlation in the plot is shown with a darker color, red or blue.

```
sample <- sample.int(n = nrow(X), size = floor(.75*nrow(X)), replace = F)
train_data = X[sample, ]
train_y = y[sample]
scale_train = scale(train_data)

X.mean = apply(train_data, 2, mean)
X.sd = apply(train_data, 2, sd)

test_data = X[-sample, ]
test_y = y[-sample]
scale_test = sapply(1:ncol(test_data),
                  function(i, X.test, X.mean, X.sd) (X.test[, i] - X.mean[i])/X.sd[i],
                  X.test = test_data, X.mean = X.mean, X.sd = X.sd)
colnames(scale_test) = colnames(train_data)
```

It is important to scale the data after we divide in test and train data such that train data don't have information on test data to train on. So I have divided the dataset into test and train, and then scaled the explanatory vaiables. I scale the test data with the mean and standard deviation of the trainig set, such that if the test set consist of one variable the test (set) still get scaled.

Model selection

Using backward elemination we can find how many vaiables is best for the model, goind from a full model with all 11 vaiables to a null model with 0 vaiables. I use Akaike information criterion to find the best model.

```
library(MASS)
full.model = lm(train_y ~ ., data = as.data.frame(scale_train))
null.model = lm(train_y ~ 1, data = as.data.frame(scale_train))

model.backward.aic = stepAIC(object = full.model, scope = null.model, direction = 'backward')
```

```
## Start:  AIC=-2143.28
## train_y ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol
##
##                        Df Sum of Sq    RSS     AIC
## - total.sulfur.dioxide  1     0.063 2036.0 -2145.2
## - chlorides             1     0.099 2036.0 -2145.1
## - citric.acid           1     0.570 2036.5 -2144.2
## <none>                              2035.9 -2143.3
## - fixed.acidity         1     1.295 2037.2 -2142.9
## - free.sulfur.dioxide   1     4.985 2040.9 -2136.3
## - pH                    1    13.447 2049.4 -2121.1
## - sulphates             1    16.554 2052.5 -2115.5
## - density               1    18.520 2054.4 -2112.0
## - residual.sugar        1    41.819 2077.7 -2070.6
## - alcohol               1    44.404 2080.3 -2066.0
## - volatile.acidity      1   123.690 2159.6 -1928.7
##
## Step:  AIC=-2145.17
## train_y ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + density + pH + sulphates +
##     alcohol
##
##                       Df Sum of Sq    RSS     AIC
## - chlorides            1     0.099 2036.1 -2147.0
## - citric.acid          1     0.555 2036.5 -2146.2
## <none>                             2036.0 -2145.2
## - fixed.acidity        1     1.319 2037.3 -2144.8
## - free.sulfur.dioxide  1     6.845 2042.8 -2134.8
## - pH                   1    13.492 2049.5 -2122.9
## - sulphates            1    16.491 2052.5 -2117.5
## - density              1    19.678 2055.7 -2111.8
## - residual.sugar       1    42.977 2079.0 -2070.4
## - alcohol              1    44.343 2080.3 -2068.0
## - volatile.acidity     1   129.791 2165.8 -1920.2
##
## Step:  AIC=-2146.99
## train_y ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     free.sulfur.dioxide + density + pH + sulphates + alcohol
##
##                       Df Sum of Sq    RSS     AIC
## - citric.acid          1     0.507 2036.6 -2148.1
## <none>                             2036.1 -2147.0
## - fixed.acidity        1     1.497 2037.6 -2146.3
## - free.sulfur.dioxide  1     6.766 2042.8 -2136.8
## - pH                   1    14.316 2050.4 -2123.2
## - sulphates            1    16.626 2052.7 -2119.1
## - density              1    20.791 2056.9 -2111.7
## - alcohol              1    44.401 2080.5 -2069.8
## - residual.sugar       1    45.924 2082.0 -2067.1
## - volatile.acidity     1   131.778 2167.9 -1918.6
##
## Step:  AIC=-2148.07
```

```
## train_y ~ fixed.acidity + volatile.acidity + residual.sugar +
##     free.sulfur.dioxide + density + pH + sulphates + alcohol
##
##                       Df Sum of Sq    RSS     AIC
## <none>                             2036.6 -2148.1
## - fixed.acidity        1     1.717 2038.3 -2147.0
## - free.sulfur.dioxide  1     7.040 2043.6 -2137.4
## - pH                   1    13.911 2050.5 -2125.1
## - sulphates            1    16.950 2053.6 -2119.6
## - density              1    20.361 2057.0 -2113.5
## - residual.sugar       1    45.466 2082.1 -2069.0
## - alcohol              1    45.736 2082.3 -2068.5
## - volatile.acidity     1   137.376 2174.0 -1910.3
```

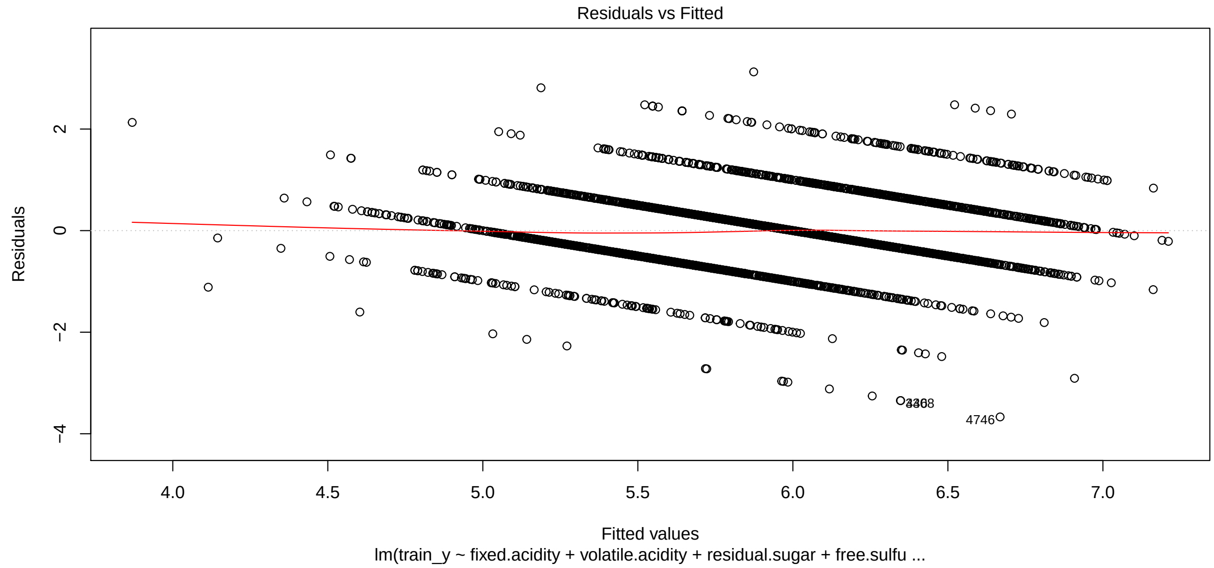
```
summary(model.backward.aic)
```

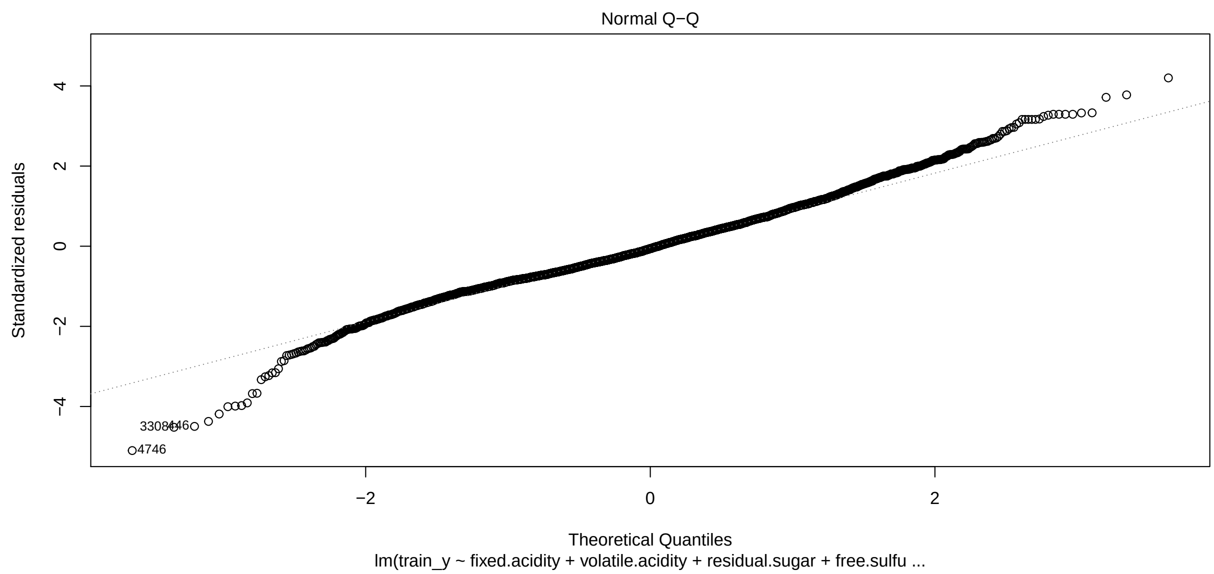
```
##
## Call:
## lm(formula = train_y ~ fixed.acidity + volatile.acidity + residual.sugar +
##     free.sulfur.dioxide + density + pH + sulphates + alcohol,
##     data = as.data.frame(scale_train))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6684 -0.4909 -0.0437  0.4422  3.1263
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.88211    0.01230 478.155  < 2e-16 ***
## fixed.acidity        0.03393    0.01931   1.758 0.078916 .
## volatile.acidity    -0.19894    0.01265 -15.721  < 2e-16 ***
## residual.sugar       0.37746    0.04174   9.044  < 2e-16 ***
## free.sulfur.dioxide  0.04682    0.01315   3.559 0.000377 ***
## density             -0.37100    0.06130  -6.052 1.57e-09 ***
## pH                   0.08912    0.01781   5.003 5.92e-07 ***
## sulphates            0.07221    0.01308   5.522 3.58e-08 ***
## alcohol              0.29917    0.03298   9.071  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7455 on 3664 degrees of freedom
## Multiple R-squared:  0.2928, Adjusted R-squared:  0.2913
## F-statistic: 189.7 on 8 and 3664 DF,  p-value: < 2.2e-16
```

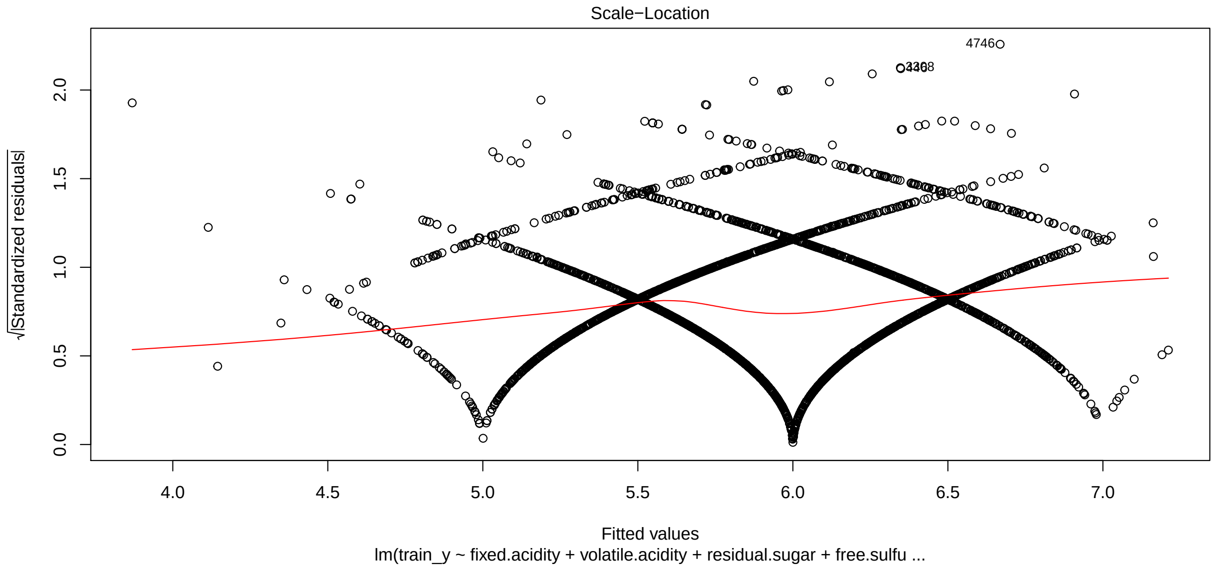
```
mean(model.backward.aic$residuals^2)
```

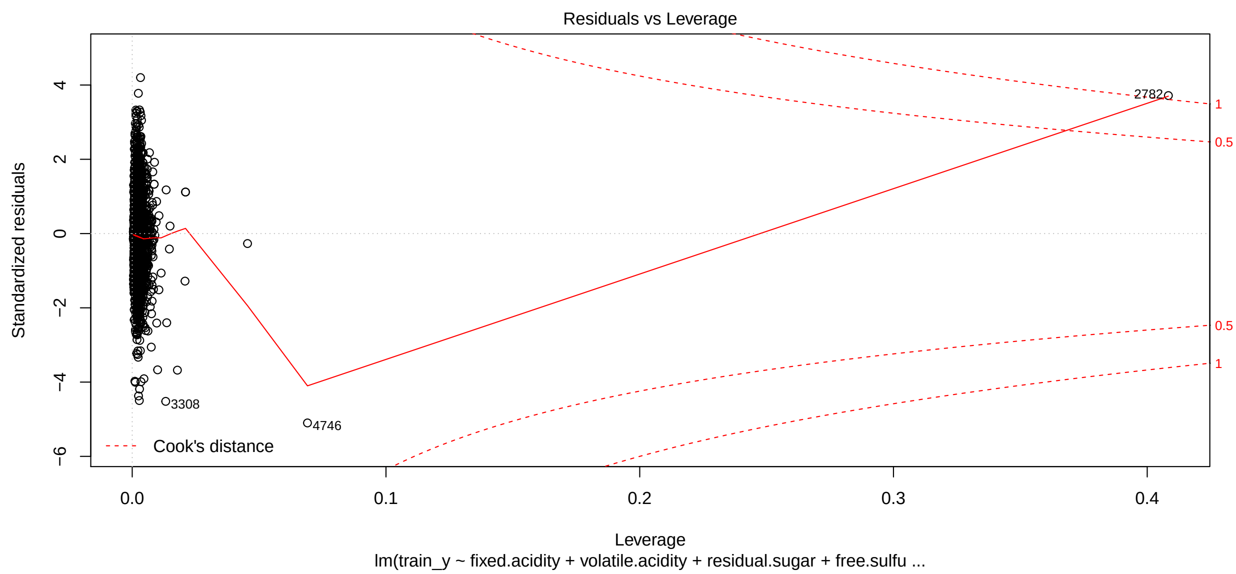
```
## [1] 0.5544775
```

```
plot(model.backward.aic)
```


Residuals vs Fitted
Residuals
Fitted values
lm(train_y ~ fixed.acidity + volatile.acidity + residual.sugar + free.sulfu ...
4746
4.0
4.5
5.0
5.5
6.0
6.5
7.0


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(train_y ~ fixed.acidity + volatile.acidity + residual.sugar + free.sulfu ...
4746





Using backward elimination we get the best model with 8 features, that are: fixed acidity, volatile avidity, residual sugar, free sulfur dioxide, density, pH, sulphates and alcohol. Then the adjusted R-squared is 0.29 and a mean squared error of 0.55.

The pattern in the Residuals vs fitted does not look random, this might sugest that a non-linear model will expalin the data better, the points is distributed around the 0 line which is good. From the normal Q-Q plot we can se that it might be a better model to explain our data since it is a couple of points in both ends who are far from the line. We can se from the Scale-Location plot that our residuals is not homoscedastic, since the points is not spread equally along the predictor range. From the Residuals vs Leverage it looks like sample 2782 is an outlier. Removing this might have given a better result.

```r
library(glmnet)
```

Lasso regression

```
## Loading required package: Matrix

## Loaded glmnet 3.0-2
```

```
cv_lasso = cv.glmnet(x = scale_train, y = train_y, alpha = 1)
lambda_cv = cv_lasso$lambda.min
cbind(cv_lasso$lambda.min, cv_lasso$lambda.1se)
```

```
##             [,1]       [,2]
## [1,] 0.004155489 0.03531141
```

```
mod_lasso = glmnet(x = scale_train, y = train_y, lambda = lambda_cv, alpha = 1)
mod_lasso$beta
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                                 s0
## fixed.acidity         .
## volatile.acidity     -0.1965563860
## citric.acid           0.0073134287
## residual.sugar        0.2809208289
## chlorides            -0.0087708851
## free.sulfur.dioxide   0.0453954340
## total.sulfur.dioxide -0.0007075601
## density              -0.2351678131
## pH                    0.0597690449
## sulphates             0.0601851891
## alcohol               0.3528069794
```

```
lasso.train.error = mean((train_y - cbind(1, scale_train) %*% c(mean(train_y),as.vector(mod_lasso$beta)
lasso.test.error = mean((test_y - cbind(1, scale_test) %*% c(mean(train_y), as.vector(mod_lasso$beta)))
best.model.error.train = mean((predict(model.backward.aic, as.data.frame(scale_train)) - train_y)^2)
best.mode.error.test = mean((predict(model.backward.aic, as.data.frame(scale_test)) - test_y)^2)
cbind(lasso.train.error, lasso.test.error, best.model.error.train, best.mode.error.test)
```

```
##      lasso.train.error lasso.test.error best.model.error.train
## [1,]         0.5551774        0.5960834              0.5544775
##      best.mode.error.test
## [1,]            0.5924736
```

The error for lasso on training is 0.555 and test 0.599, and for the linear model with 8 vaiables a train error of 0.553 and test error of 0.595, this is a high error. We don't know if when the model predict wrong it is way off or only by 1. It is worse if it predict a wine is a 10 when it is a 1, than predictin it is 5 when it is 6. This can be analysed further.

Conclution

The MSE for lasso and for the best linear model found using AIC is not that good, so linear model and lasso regression is not great to predict if a wine is of good quality given these explanatory vaiables using the methods in this report.

Librarys used

ggplot2: for boxplot

ggcorrplot: for correlationpolt

MASS: for AIC

glmnet: for lasso regression

References:

http://archive.ics.uci.edu/ml/datasets/Wine+Quality

https://medium.com/data-distilled/residual-plots-part-3-scale-location-plot-113e469b99c

https://www.r-bloggers.com/2013/06/box-plot-with-r-tutorial/

Lecture notes