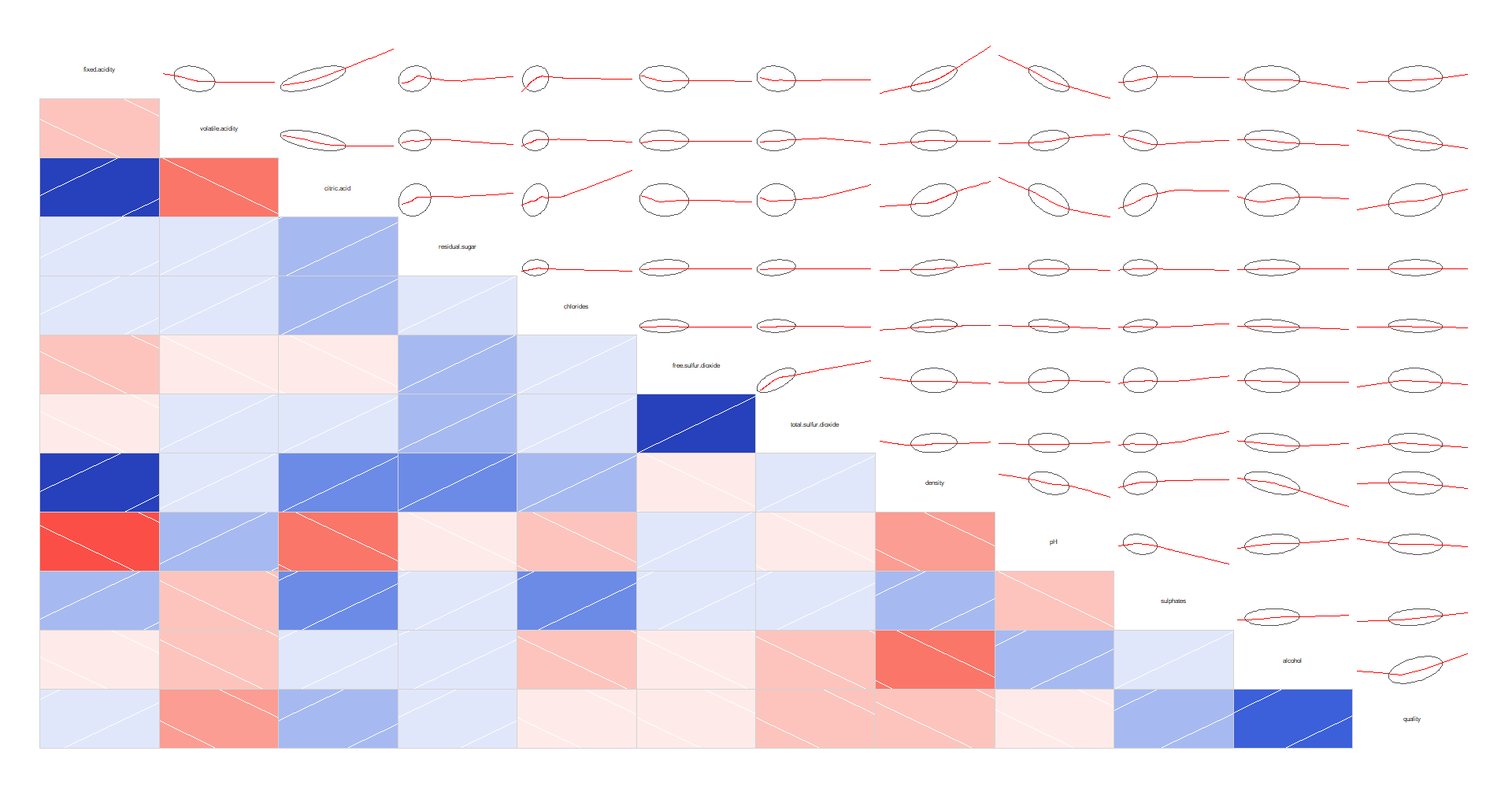
Multiple Linear Regression

Using R



Wine Quality Data Set for regression or classification modelling

By,

1.Abhiraj Suryawanshi – A00423508

2. Jinal Salvi – A00421173

3. Ranjit Mishra – A00425569

MCDA 5520 | Stats & Data Analytics | 3rd April, 2018

Table of Contents

[Abstract 2](#_Toc510563727)

[Introduction and background 2](#_Toc510563728)

[Goal 3](#_Toc510563729)

[Study methodology 3](#_Toc510563730)

[Step-By-Step Analysis 3](#_Toc510563731)

[1. Load the data in R software 3](#_Toc510563732)

[2. What is in the data? 3](#_Toc510563733)

[3. Summary of this data 4](#_Toc510563734)

[4. 1st Regression Model 4](#_Toc510563735)

[Visualizing variable relationships 5](#_Toc510563736)

[Interpreting the correlogram 6](#_Toc510563737)

[5. 2nd Regression Model (4 variable) 7](#_Toc510563738)

[6. 3rd Regression Model (3 variable) 8](#_Toc510563739)

[Collinearity and ANOVA test 8](#_Toc510563740)

[Linearity test by residual plots 9](#_Toc510563741)

[Final prediction 11](#_Toc510563742)

[Conclusions and future work 11](#_Toc510563743)

[References 11](#_Toc510563744)

# Abstract

The two datasets are related to red and white variants of the Portuguese "Vinho Verde" wine. For more details, consult the reference [Cortez et al., 2009]. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

These datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are much more normal wines than excellent or poor ones). Outlier detection algorithms could be used to detect the few excellent or poor wines. Also, we are not sure if all input variables are relevant. So, it could be interesting to test feature selection methods.

Relevant Papers:

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties.

In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

Available at: [[Web Link]](http://dx.doi.org/10.1016/j.dss.2009.05.016)

# Introduction and background

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Below is the detail snap shot of the data set. |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Data Set Characteristics:** | Multivariate | **Number of Instances:** | 4898 | **Area:** | Business | | **Attribute Characteristics:** | Real | **Number of Attributes:** | 12 | **Date Donated** | 2009-10-07 | | **Associated Tasks:** | Classification, Regression | **Missing Values?** | N/A | **Number of Web Hits:** | 620300 | |

For more information, read [Cortez et al., 2009].

Input variables (based on physicochemical tests):

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

Output variable (based on sensory data):

12 - quality (score between 0 and 10)

# Goal

Use machine learning to determine which physiochemical properties make a wine 'good' using R software.

# Study methodology

The goal is to model wine quality based on physicochemical tests. So, we are trying to solve it though regression model.

# Step-By-Step Analysis

## Load the data in R software

R commands to load the data.

*wine\_data <- read.csv("C:/Users/Ranjit/Desktop/Stats/winequality-red.csv", sep=";")*

## What is in the data?

In this data set, we have a variety of numerical categories, and the result at the end is a quality score. we used the **head()** function to show the first few rows of the table.

R command - *head(winequality.red)*

We can see each observation is a row, and there are 12 numerical data elements:

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

We can see the different scales of the data, and the first 11 are obviously objective variables involving chemical measurement.

The 12th dimension, quality, is the variable we're trying to predict.

## Summary of this data

R commands:

> *summary(wine\_data$quality)*

Min. 1st Qu. Median Mean 3rd Qu. Max.

3.000 5.000 6.000 5.636 6.000 8.000

> *table(wine\_data$quality)*

3 4 5 6 7 8

10 53 681 638 199 18

**Interpretation of data:**

First, we used summary () which gives summary statistics of the data. We see the data run from 3 to 8, and it's very interesting that all the quartiles are round numbers and that the median and the 3rd quartile (aka the 50th and 75th percentiles) are the same.

We used table () which gives a better idea of what the data look like. It's whole numbers from 3 to 8, with a lot of the data heaped about 5 and 6.

There are some ways we can try to predict these scores:

1. regression techniques which will output numbers
2. classification techniques, where each quality score is considered a class or category

## 1st Regression Model

Let's just do a linear regression for quality, using all the other variables as input.

**R command:**

*linear\_quality = lm(quality ~ fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+density+pH+sulphates+alcohol, data=wine\_data)*

*summary(linear\_quality)*

Call:

lm(formula = quality ~ fixed.acidity + volatile.acidity + citric.acid +

residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +

density + pH + sulphates + alcohol, data = wine\_data)

Residuals:

Min 1Q Median 3Q Max

-2.68911 -0.36652 -0.04699 0.45202 2.02498

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.197e+01 2.119e+01 1.036 0.3002

fixed.acidity 2.499e-02 2.595e-02 0.963 0.3357

volatile.acidity -1.084e+00 1.211e-01 -8.948 < 2e-16 \*\*\*

citric.acid -1.826e-01 1.472e-01 -1.240 0.2150

residual.sugar 1.633e-02 1.500e-02 1.089 0.2765

chlorides -1.874e+00 4.193e-01 -4.470 8.37e-06 \*\*\*

free.sulfur.dioxide 4.361e-03 2.171e-03 2.009 0.0447 \*

total.sulfur.dioxide -3.265e-03 7.287e-04 -4.480 8.00e-06 \*\*\*

density -1.788e+01 2.163e+01 -0.827 0.4086

pH -4.137e-01 1.916e-01 -2.159 0.0310 \*

sulphates 9.163e-01 1.143e-01 8.014 2.13e-15 \*\*\*

alcohol 2.762e-01 2.648e-02 10.429 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.648 on 1587 degrees of freedom

Multiple R-squared: 0.3606, Adjusted R-squared: 0.3561

F-statistic: 81.35 on 11 and 1587 DF, p-value: < 2.2e-16

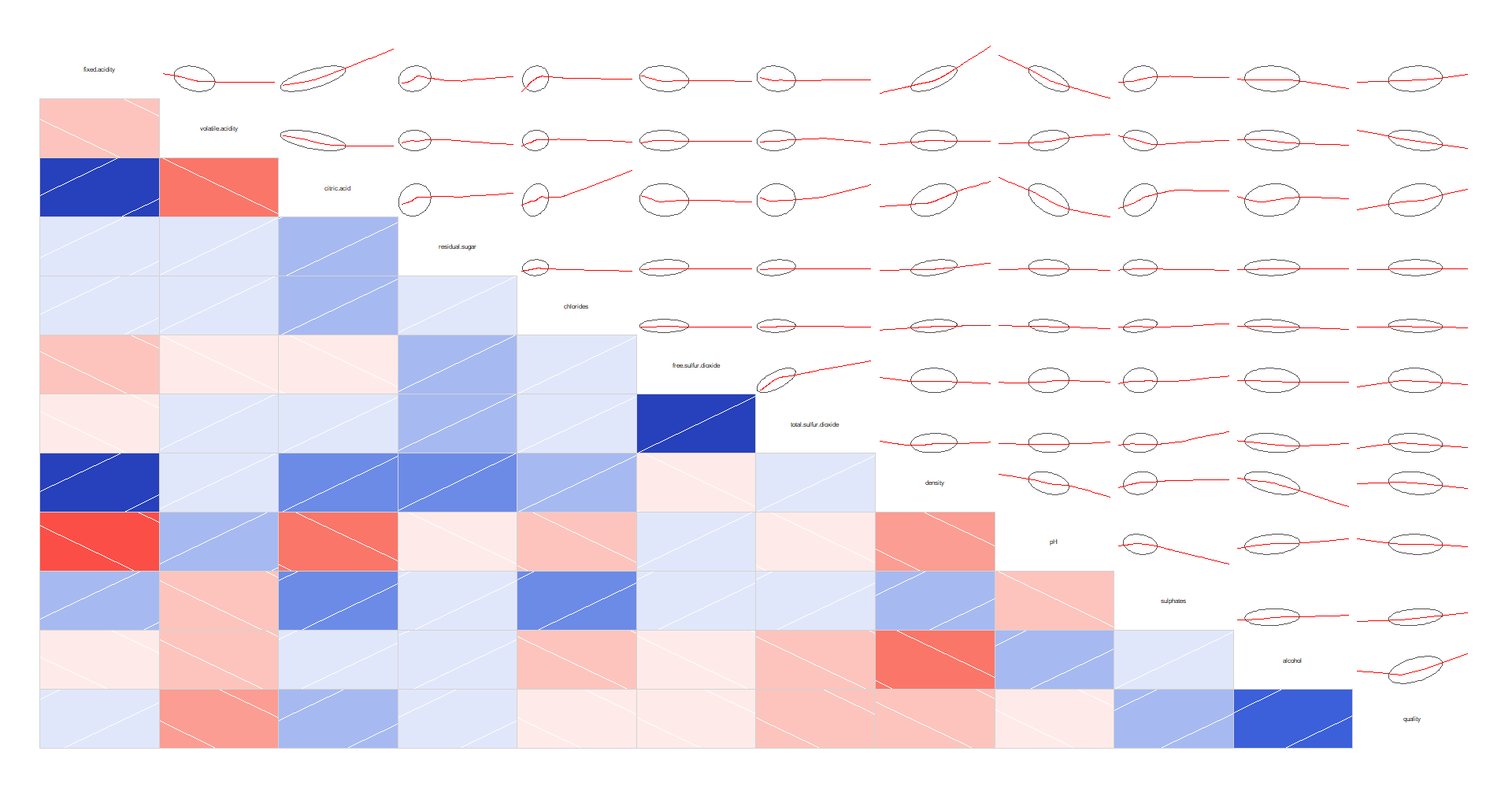
Summary: It looks bad. For some of the item it's hard to tell how well they work together. Just because something has a very small significance value, may not mean it really is significant.

Perhaps it would be helpful to visualize the data, so we can better decide the best input variables to use.

### Visualizing variable relationships

We use below correlogram, which reflects pairwise correlations.

**R command:** *corrgram(wine\_data, lower.panel=panel.shade, upper.panel=panel.ellipse)*



In the lower left half of the display, hcolor saturation indicates strength of correlation and the colors indicate direction of correlation (red=negative, blue=positive...for the colorblind, we also get white lines indicating direction).

In the upper right half of the display, we see an ellipse and a red line. The red line is showing a smoothed line indicating relationship between the two variables, and the ellipse provide the confidence intervals.

### Interpreting the correlogram

We're trying to predict quality, so we care about the final row and column in order to choose our variables. We may wish to look elsewhere in the matrix, too, to make sure that we're not picking independent variables that are quite correlated: for example, the free.sulphur.dioxide and total.sulfur.dioxide have extremely strong correlation, so we would not want to use both variables in a simplified model.

Our first try was using all the variables. Now let's use fewer meaningful ones for our next try.

## 2nd Regression Model (4 variable)

R command:

*linear\_quality\_4 = lm(quality ~ alcohol + volatile.acidity + citric.acid + sulphates, data = wine\_data)*

*summary(linear\_quality\_4)*

Call:

lm(formula = quality ~ alcohol + volatile.acidity + citric.acid +

sulphates, data = wine\_data)

Residuals:

Min 1Q Median 3Q Max

-2.71408 -0.38590 -0.06402 0.46657 2.20393

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.64592 0.20106 13.160 < 2e-16 \*\*\*

alcohol 0.30908 0.01581 19.553 < 2e-16 \*\*\*

volatile.acidity -1.26506 0.11266 -11.229 < 2e-16 \*\*\*

citric.acid -0.07913 0.10381 -0.762 0.446

sulphates 0.69552 0.10311 6.746 2.12e-11 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6588 on 1594 degrees of freedom

Multiple R-squared: 0.3361, Adjusted R-squared: 0.3345

F-statistic: 201.8 on 4 and 1594 DF, p-value: < 2.2e-16

**Interpreting the data:**

From above table its clear that its much better then then first one, but still there is one parameter “citric.acid” which is insignificant to total data. Now let’s try to remove it and check again.

## 3rd Regression Model (3 variable)

**R command:**

linear\_quality\_5 = lm(quality ~ alcohol + volatile.acidity + sulphates, data = wine\_data)

summary(linear\_quality\_5)

Call:

lm(formula = quality ~ alcohol + volatile.acidity + sulphates,

data = wine\_data)

Residuals:

Min 1Q Median 3Q Max

-2.7186 -0.3820 -0.0641 0.4746 2.1807

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.61083 0.19569 13.342 < 2e-16 \*\*\*

alcohol 0.30922 0.01580 19.566 < 2e-16 \*\*\*

volatile.acidity -1.22140 0.09701 -12.591 < 2e-16 \*\*\*

sulphates 0.67903 0.10080 6.737 2.26e-11 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6587 on 1595 degrees of freedom

Multiple R-squared: 0.3359, Adjusted R-squared: 0.3346

F-statistic: 268.9 on 3 and 1595 DF, p-value: < 2.2e-16

Interpreting data:

This model did not improve much from previous model, but it removed one unnecessary parameter. Now it got all the parameter which are very significant to it.

Now let’s check the collinearity(VIF) and ANOVA test of these parameters to conclude on this model.

### Collinearity and ANOVA test

*> anova(linear\_quality\_5)*

Analysis of Variance Table

Response: quality

Df Sum Sq Mean Sq F value Pr(>F)

alcohol 1 236.29 236.295 544.56 < 2.2e-16 \*\*\*

volatile.acidity 1 94.07 94.074 216.80 < 2.2e-16 \*\*\*

sulphates 1 19.69 19.692 45.38 2.258e-11 \*\*\*

Residuals 1595 692.10 0.434

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

*> vif(linear\_quality\_5)*

alcohol volatile.acidity sulphates

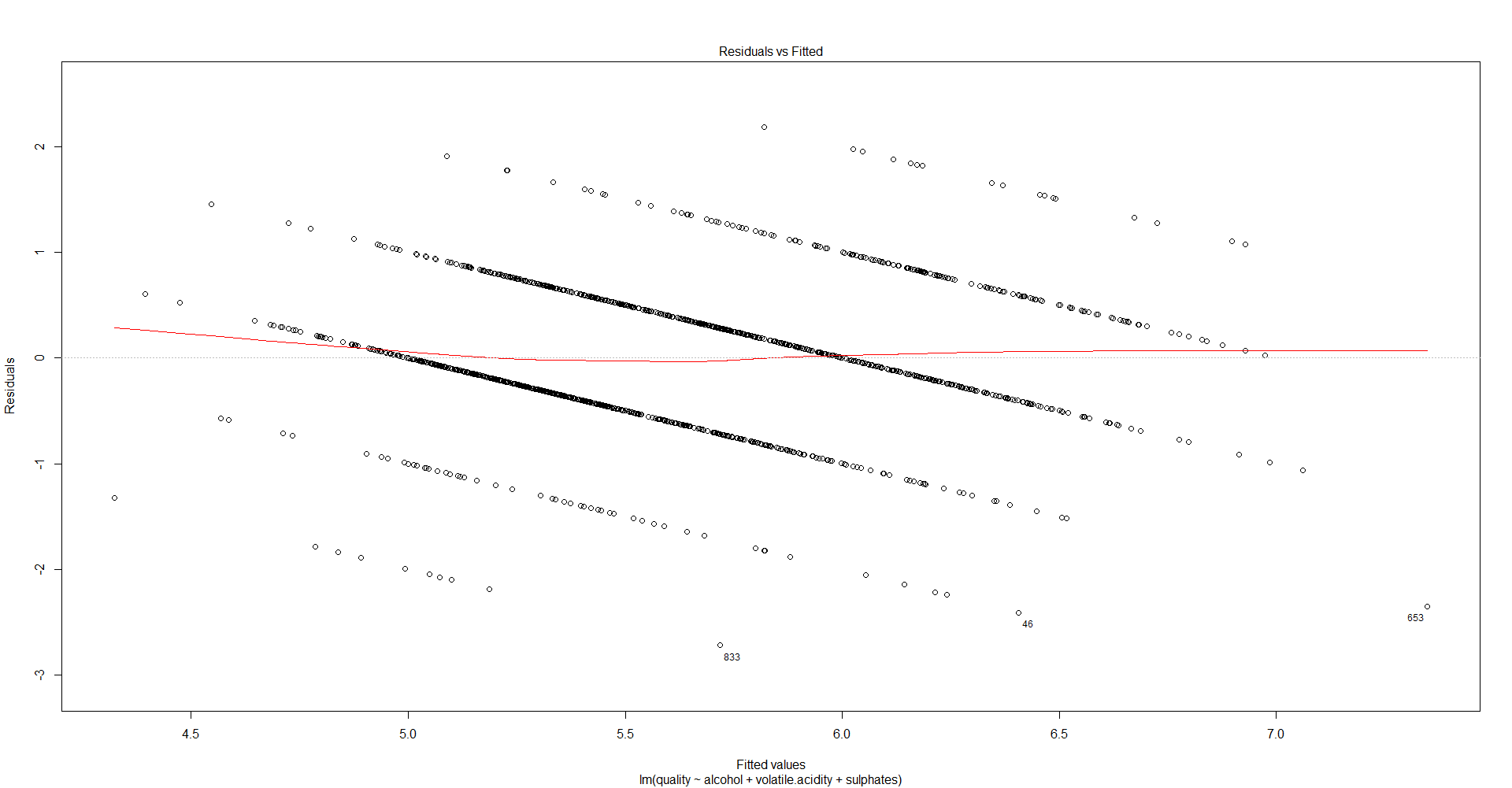
1.044612 1.111146 1.075095

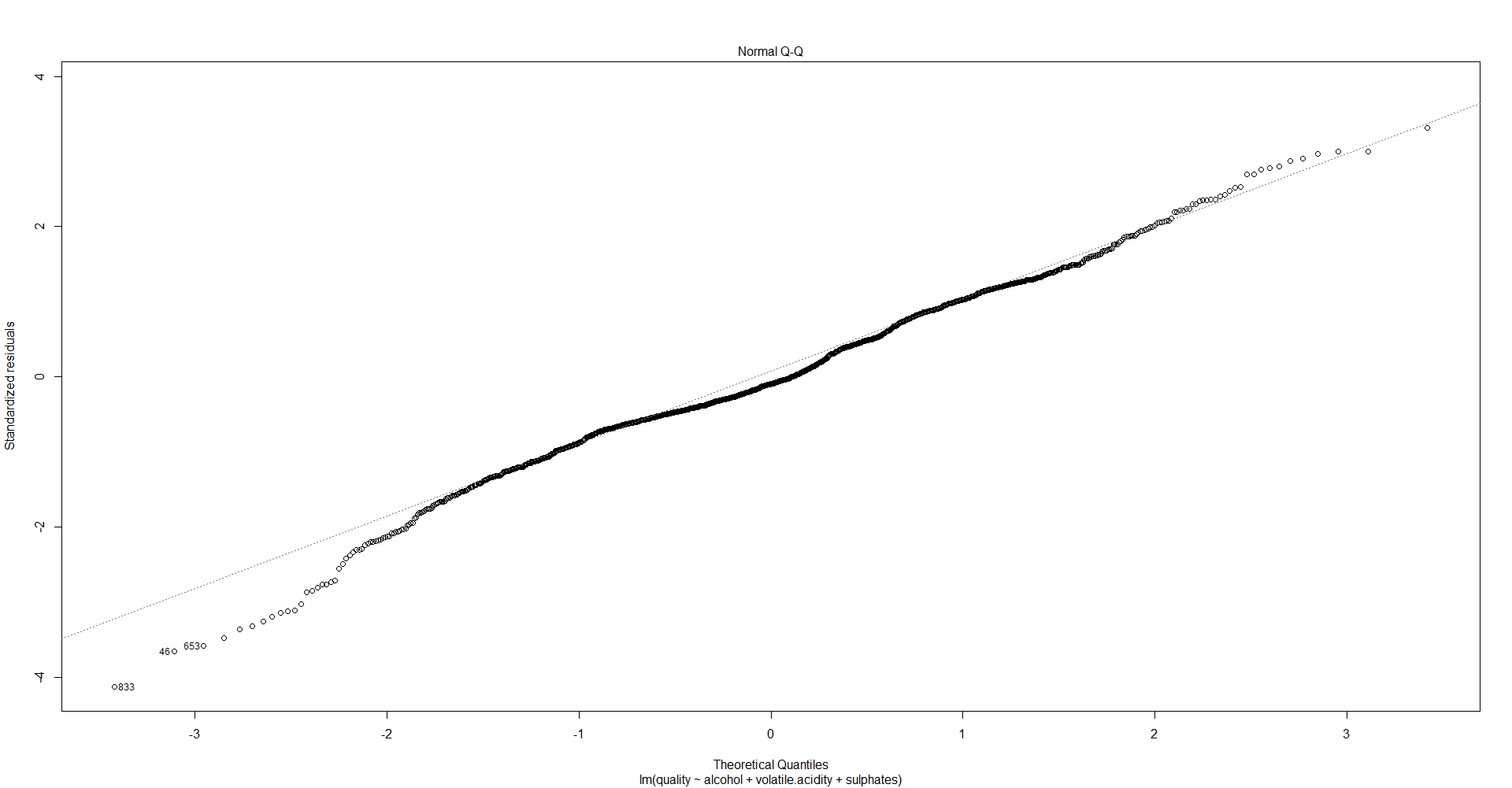
**Analysis:**

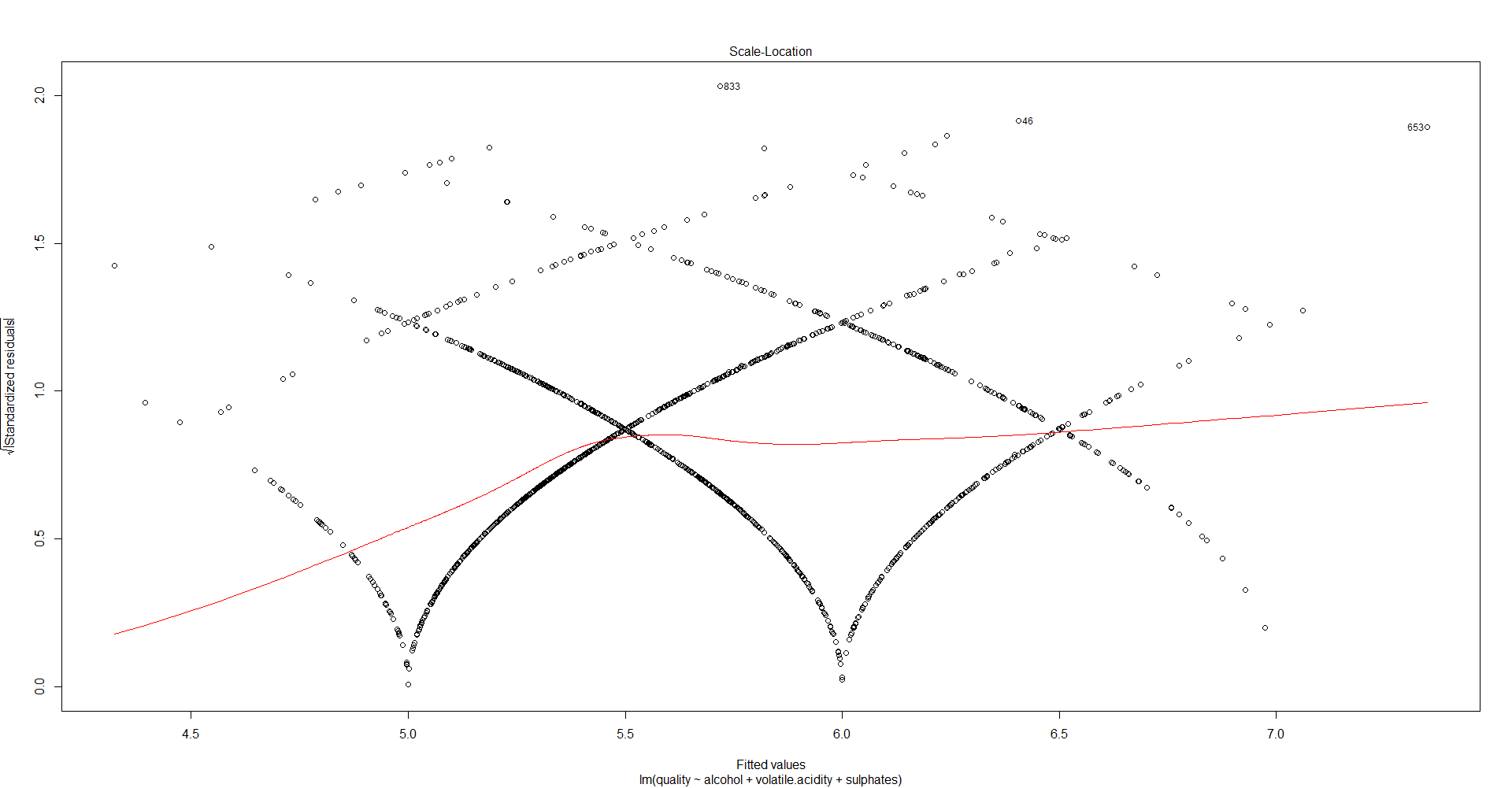
From VIF test we can see that all the values are below 5. So, we can conclude that the collinearity of these variables is not present. Similarly, ANOVA test shows that the remaining all the variables are significant (less than 0.001) and hence we can conclude that for our confidence interval **we accept the alternative hypothesis H1** that there is a significant relationship between Quality of wine and these chemicals.

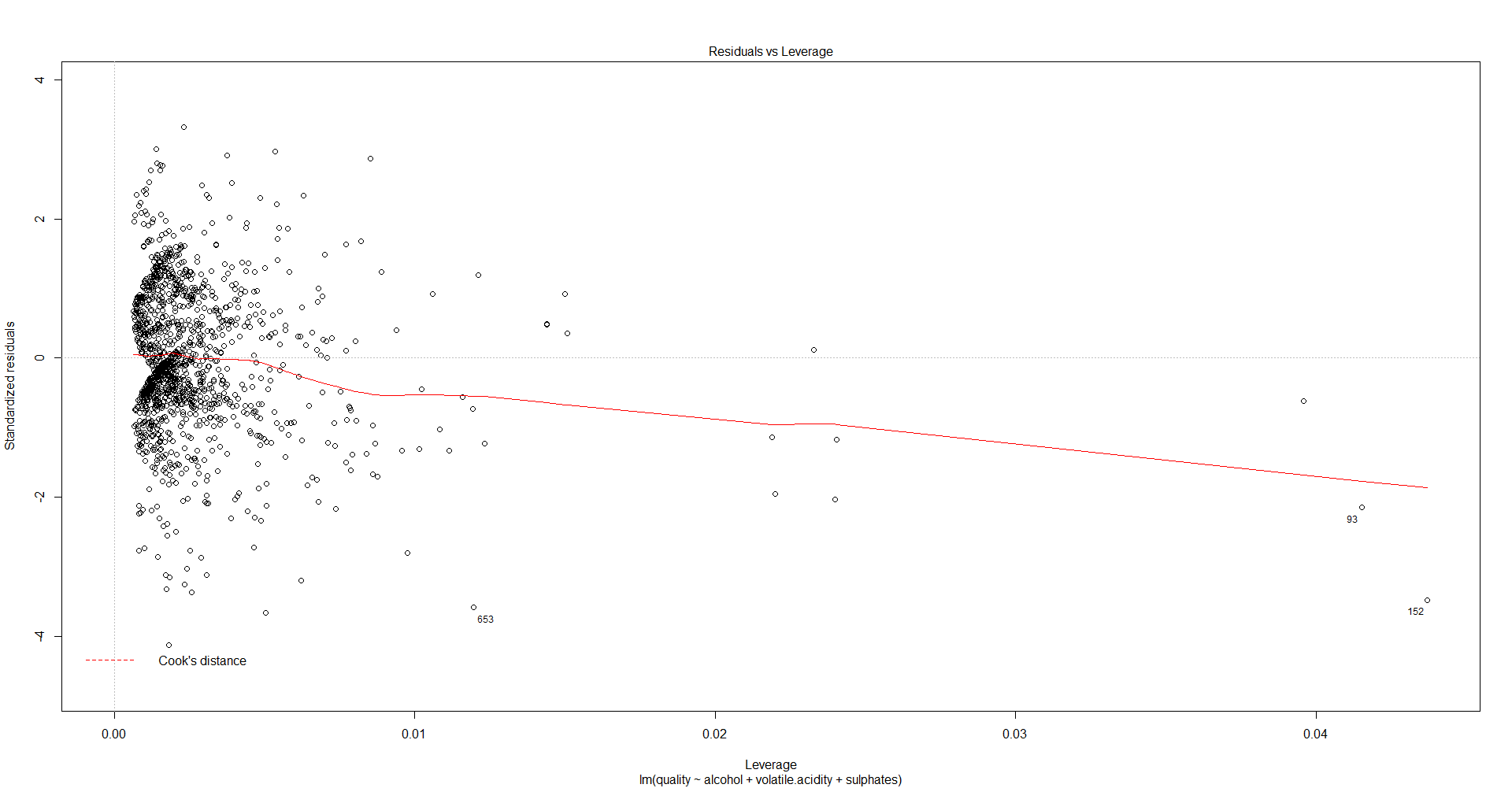
### Linearity test by residual plots

We had had assumed that the data are fit for leaner regression. Now at the end of this test let’s just verity that the data were indeed fit for linear regression by plotting their residual plots.









Interpreting data:

The above residual plot shows that the data was fit for linear regression model. It in homoscedasticity and does not cross to cook’s distance line.

# Final prediction

It’s time to test the model.

# Split data into training and validation samples

# We will use (train.size)% for training and (100-train.size)% for validation

> set.seed(2018)

> train.size <- 0.8

> train.index <- sample.int(length(wine\_data$quality), round(length(wine\_data$quality) \* train.size))

> train.sample <- wine\_data[train.index,]

> valid.sample <- wine\_data[-train.index,]

##### Now evaluate the final linear model

# Find all predicted values for both a training set and a validation set

> train.sample$Pred.quality <- predict(linear\_quality\_5,

+ newdata = subset(train.sample, select=c(quality, alcohol, volatile.acidity, citric.acid, sulphates)))

> valid.sample$Pred.quality <- predict(linear\_quality\_5,

+ newdata = subset(valid.sample, select=c(quality, alcohol, volatile.acidity, citric.acid, sulphates)))

> # The theoretical model performance is defined here as R-Squared

> summary(linear\_quality\_5)

Call:

lm(formula = quality ~ alcohol + volatile.acidity + sulphates,

data = wine\_data)

Residuals:

Min 1Q Median 3Q Max

-2.7186 -0.3820 -0.0641 0.4746 2.1807

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.61083 0.19569 13.342 < 2e-16 \*\*\*

alcohol 0.30922 0.01580 19.566 < 2e-16 \*\*\*

volatile.acidity -1.22140 0.09701 -12.591 < 2e-16 \*\*\*

sulphates 0.67903 0.10080 6.737 2.26e-11 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6587 on 1595 degrees of freedom

Multiple R-squared: 0.3359, Adjusted R-squared: 0.3346

F-statistic: 268.9 on 3 and 1595 DF, p-value: < 2.2e-16

# Check how good is the model on the training set - correlation^2, RME and MAE

> train.corr <- round(cor(train.sample$Pred.quality, train.sample$quality), 2)

> train.RMSE <- round(sqrt(mean((train.sample$Pred.quality - train.sample$quality)^2)))

> train.MAE <- round(mean(abs(train.sample$Pred.quality - train.sample$quality)))

> c(train.corr^2, train.RMSE, train.MAE)

[1] 0.3364 1.0000 1.0000

# Check how good is the model on the validation set - correlation^2, RME and MAE

> valid.corr <- round(cor(valid.sample$Pred.quality, valid.sample$quality), 2)

> valid.RMSE <- round(sqrt(mean((valid.sample$Pred.quality - valid.sample$quality)^2)))

> valid.MAE <- round(mean(abs(valid.sample$Pred.quality - valid.sample$quality)))

> c(valid.corr^2, valid.RMSE, valid.MAE)

[1] 0.3481 1.0000 0.0000

The above corr, RMSE and MAE shows that the model prediction is great and same as the R-squared value.

# Conclusions and future work

The data is fit for linear regression as it follows homoscedasticity. We also identified all the chemicals which makes a positive impact on the wine quality.

Moreover, above mentioned 3rd model (3 variable linear modem) is the best suited multiple linear regression for the wine quality prediction. It’s Multiple R-squared: 0.3359, Adjusted R-squared: 0.3346 and F-statistic: 268.9 value also looks good.

Unfortunately, with the given data set above is the maximum prediction level we can achieve though multiple linear regression. Next ideal step will be to try a different model (probably decision tree model) and check if that model can predict better results compared to multiple linear regression.

# References

1. This dataset is also available from the UCI machine learning repository, <https://archive.ics.uci.edu/ml/datasets/wine+quality> . We are not the owner of this dataset.
2. The script file of R is also attached below for reference

