Wine Quality Regression Problem

by Viviane Adohouannon, Kate Alexander, Diana Azbel, Igor Baranov

Abstract We are using a dataset related to red vinho verde wine samples, from the north of Portugal. The goal is to model wine quality based on physicochemical tests. 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. The method chosen to solve the problem is Linear Regression.

Introduction

Once viewed as a luxury good, nowadays wine is increasingly enjoyed by all consumers. Portugal is a top ten wine exporting country with 3.17% of the market share in 2005 (FAOSTAT). Exports of its vinho verde wine (from the northwest region) have increased by yearly. To support its growth, the wine industry is investing in new technologies for both wine making and selling processes. Wine certification and quality assessment are key elements within this context. Certification prevents the illegal adulteration of wines (to safeguard human health) and assures quality for the wine market. Quality evaluation is often part of the certification process and can be used to improve wine making (by identifying the most influential factors) and to stratify wines such as premium brands (useful for setting prices). Wine certification is generally assessed by physicochemical and sensory tests (Teranishi et al., 1999). Physicochemical laboratory tests routinely used to characterize wine include determination of density, alcohol or pH values, while sensory tests rely mainly on human experts. It should be stressed that taste is the least understood of the human senses, thus wine classification is a difficult task. Moreover, the relationships between the physicochemical and sensory analysis are complex and still not fully understood (Legin et al., 2003).

Background

The two datasets presented in (UCI Wine Data Set) are related to red and white variants of the Portuguese "Vinho Verde" wine. For more details, consult (Cortez et al., 1998). 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.

Due to specific purpose of this lab assignment, we are looking at Linear Regression problem only using red wine dataset. Full library of the wine datasets and their description are located here: (UCI Wine Data Set).

Objective

The objective of this article is to provide a reliable and feasible recommendation algorithm to predict wine quality based on physicochemical tests. The target value is a numeric value of wine 'quality', hence the task could be solved by Linear Regression methods. The following methodology 'check list' standard for a Linear Regression tasks will be applied to the problem at hand:

- Put all relevant variables in the model
- Leave the irrelevant variables out
- Check linearity
- Check regression assumptions:
  - Residuals have a mean of zero
  - Normality of errors
  - Residuals are not auto correlated
  - Linearity of variables
  - More data than independent variables is used in model building
  - No excessive collinearity

Data understanding

The dataset (UCI Wine Data Set) of red wine quality has 12 attributes and 1599 instances. For more information, read (Cortez et al., 1998). The following is the concept structure of the dataset:

```
Input variables (based on physicochemical tests):

1 - fixed acidity        (FA)
2 - volatile acidity     (VA)
3 - citric acid          (CA)
4 - residual sugar       (RS)
5 - chlorides            (CH)
6 - free sulfur dioxide  (FSD)
7 - total sulfur dioxide (TSD)
8 - density              (DEN)
9 - pH                   (pH)
10 - sulphates           (SUL)
11 - alcohol             (ALC)

Output variable (based on sensory data):
12 - quality (score between 0 and 10) - (QLT)
```

Preparation

To perform the analysis, certain R libraries were used. The code below was used to load and initialize the libraries. The first line invoking seed function was applied to enforce the repeatability of the calculation results.

Reading red wines dataset

The dataset was loaded directly from the site (UCI Wine Data Set) using the R statement below. Note that column names were assigned as the dataset headers had long name making it inconvenient to present it in tables and charts. Correspondence of variable codes to long names is shown in the previous section.

```
wines_red_data <- read.csv(
  paste ("http://archive.ics.uci.edu",
         "ml/machine-learning-databases/wine-quality/winequality-red.csv",
         sep = "/"),
 sep=";", header = TRUE,
 col.names = c("FA","VA","CA","RS","CH","FSD","TSD","DEN","pH","SUL","ALC","QLT"))
```

Preview of the data

To pretty-print the first rows of the dataset xtable (Dahl, 2016) library was used to generate Table 1.

	FA	VA	CA	RS	CH	FSD	TSD	DEN	pH	SUL	ALC	QLT
1	7.40	0.70	0.00	1.90	0.08	11.00	34.00	1.00	3.51	0.56	9.40	5
2	7.80	0.88	0.00	2.60	0.10	25.00	67.00	1.00	3.20	0.68	9.80	5
3	7.80	0.76	0.04	2.30	0.09	15.00	54.00	1.00	3.26	0.65	9.80	5
4	11.20	0.28	0.56	1.90	0.07	17.00	60.00	1.00	3.16	0.58	9.80	6
5	7.40	0.70	0.00	1.90	0.08	11.00	34.00	1.00	3.51	0.56	9.40	5
6	7.40	0.66	0.00	1.80	0.07	13.00	40.00	1.00	3.51	0.56	9.40	5
7	7.90	0.60	0.06	1.60	0.07	15.00	59.00	1.00	3.30	0.46	9.40	5
8	7.30	0.65	0.00	1.20	0.06	15.00	21.00	0.99	3.39	0.47	10.00	7
9	7.80	0.58	0.02	2.00	0.07	9.00	18.00	1.00	3.36	0.57	9.50	7
10	7.50	0.50	0.36	6.10	0.07	17.00	102.00	1.00	3.35	0.80	10.50	5

Table 1: Red Wines Quality Dataset - first rows

Data attributes summary

Quick view of the data attributes statistics presented in the Table 2. For each attribute in the dataset this table shows min, max, mean and normal distribution 1st and 3rd quartiles values.

FA	VA	CA	RS	CH	FSD
Min. : 4.60	Min. :0.1200	Min. :0.000	Min. : 0.900	Min. :0.01200	Min. : 1.00
1st Qu.: 7.10	1st Qu.:0.3900	1st Qu.:0.090	1st Qu.: 1.900	1st Qu.:0.07000	1st Qu.: 7.00
Median : 7.90	Median :0.5200	Median :0.260	Median : 2.200	Median :0.07900	Median :14.00
Mean : 8.32	Mean :0.5278	Mean :0.271	Mean : 2.539	Mean :0.08747	Mean :15.87
3rd Qu.: 9.20	3rd Qu.:0.6400	3rd Qu.:0.420	3rd Qu.: 2.600	3rd Qu.:0.09000	3rd Qu.:21.00
Max. :15.90	Max. :1.5800	Max. :1.000	Max. :15.500	Max. :0.61100	Max. :72.00

TSD	DEN	pH	SUL	ALC	QLT
Min. : 6.00	Min. :0.9901	Min. :2.740	Min. :0.3300	Min. : 8.40	Min. :3.000
1st Qu.: 22.00	1st Qu.:0.9956	1st Qu.:3.210	1st Qu.:0.5500	1st Qu.: 9.50	1st Qu.:5.000
Median : 38.00	Median :0.9968	Median :3.310	Median :0.6200	Median :10.20	Median :6.000
Mean : 46.47	Mean :0.9967	Mean :3.311	Mean :0.6581	Mean :10.42	Mean :5.636
3rd Qu.: 62.00	3rd Qu.:0.9978	3rd Qu.:3.400	3rd Qu.:0.7300	3rd Qu.:11.10	3rd Qu.:6.000
Max. :289.00	Max. :1.0037	Max. :4.010	Max. :2.0000	Max. :14.90	Max. :8.000

Table 2: `Red Wines Dataset Summary`

Check for missing values

The dataset has no missing values. Code below calculate number of rows with missing values and checks if there is at list one.

```
any(is.na(wines_red_data))
```

```
#> [1] FALSE
```

Distribution of target value in the dataset

As we mentioned before, the target value QLT of the wine quality is not equally distributed. The Figure 1 demonstrates the distribution. As we can see, dataset covers mostly medium-quality wines with QLT between 5 and 7 well, low and high quality wines represented poorly.

```
ggplot(data = wines_red_data, mapping = aes(x = QLT)) + geom_bar()
```

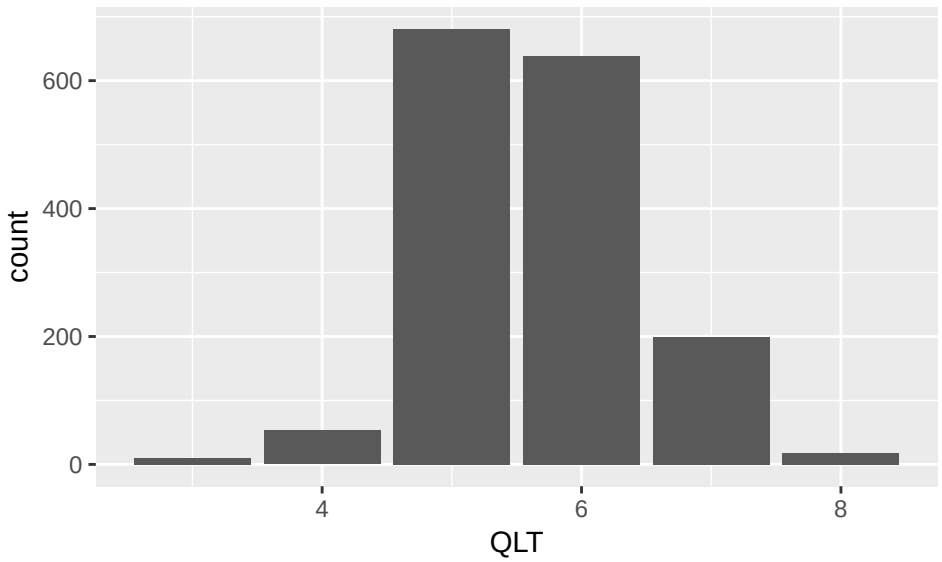

Figure 1: Distribution of Wine Quality Attribute

Linear Regression Modeling

Default Linear Regression fit

The following code calculates the default Ordinary Least Squares (OLS) model using all the independent variables. Results of the calculations and analysis are presented below.

```
wines_red_data.fit <- lm (QLT ~ ., data=wines_red_data)
fit.sum <- summary(wines_red_data.fit)
fit.sum

#>
#> Call:
#> lm(formula = QLT ~ ., data = wines_red_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
#> FA           2.499e-02  2.595e-02   0.963   0.3357
#> VA          -1.084e+00  1.211e-01  -8.948  < 2e-16 ***
#> CA          -1.826e-01  1.472e-01  -1.240   0.2150
#> RS           1.633e-02  1.500e-02   1.089   0.2765
#> CH          -1.874e+00  4.193e-01  -4.470 8.37e-06 ***
#> FSD          4.361e-03  2.171e-03   2.009   0.0447 *
#> TSD         -3.265e-03  7.287e-04  -4.480 8.00e-06 ***
#> DEN         -1.788e+01  2.163e+01  -0.827   0.4086
#> pH          -4.137e-01  1.916e-01  -2.159   0.0310 *
#> SUL          9.163e-01  1.143e-01   8.014 2.13e-15 ***
#> ALC          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
```

The first item shown in the output is the formula R used to fit the data. The next item in the model output talks about the residuals. Residuals are essentially the difference between the actual observed response values (distance to stop dist in our case) and the response values that the model predicted. We can see that those values are distributed with slight skewness. The next section in the model output talks about the coefficients of the model.

The coefficient Standard Error measures the average amount that the coefficient estimates vary from the actual average value of our response variable.

The coefficient t-value is a measure of how many standard deviations our coefficient estimate is far away from 0. We want it to be far away from zero as this would indicate we could reject the null hypothesis - that is, we could declare a relationship between speed and distance exist. In our example, the t-statistic values are relatively far away from zero for some attributes and close to others.

The model should be adjusted by removing some of the attributes from it. The Pr(>t) acronym found in the model output relates to the probability of observing any value equal or larger than t.

F-statistic is an indicator of whether there is a relationship between predictor and the response variables. When the number of data points is large, an F-statistic that is only a little bit larger than 1 is already sufficient to reject the null hypothesis (H0 : There is no relationship). The reverse is true, a large F-statistic is required to be able to ascertain that there may be a relationship between predictor and response variables. In our example the F-statistic is relatively larger than 1 given the size of our data. We could say that H0 hypothesis is not rejected in our model.

As we can see, Adjusted R-squared is pretty low at 0.3561195. In addition several variables like FA, CA, RS, and DEN have shown to be not significant at $p > 0.05$.

Adjusting the model

Let's remove the unsignificant attributes FA, CA, RS, and DEN from the model and check if this helps to increase the model reliability.

```
wines_red_data.fit1 <- lm (QLT ~ VA + CH + FSD + TSD + pH + SUL + ALC,
                           data=wines_red_data)
fit1.sum <- summary(wines_red_data.fit1)
fit1.sum
```

```
#>
#> Call:
#> lm(formula = QLT ~ VA + CH + FSD + TSD + pH + SUL + ALC, data = wines_red_data)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -2.68918 -0.36757 -0.04653  0.46081  2.02954
#>
#> Coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  4.4300987  0.4029168  10.995  < 2e-16 ***
#> VA          -1.0127527  0.1008429 -10.043  < 2e-16 ***
#> CH          -2.0178138  0.3975417  -5.076 4.31e-07 ***
#> FSD          0.0050774  0.0021255   2.389    0.017 *
#> TSD         -0.0034822  0.0006868  -5.070 4.43e-07 ***
#> pH          -0.4826614  0.1175581  -4.106 4.23e-05 ***
#> SUL          0.8826651  0.1099084   8.031 1.86e-15 ***
#> ALC          0.2893028  0.0167958  17.225  < 2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.6477 on 1591 degrees of freedom
#> Multiple R-squared:  0.3595, Adjusted R-squared:  0.3567
#> F-statistic: 127.6 on 7 and 1591 DF,  p-value: < 2.2e-16
```

As we can see, the reliability of the new model slightly increased. We now have Adjusted R-squared at 0.3566527 comparing to the previous 0.3561195.

Stepwise Regression

Stepwise Regression method starts with the full model and eliminates predictors one at a time, at each step considering whether the criterion will be improved. As we see from the algorithm output, the new model is identical to the one found the previous section manually.

```
library(MASS)
fit <- lm(QLT ~ .,data=wines_red_data)
step <- stepAIC(fit, direction="both", trace = FALSE)
step$anova
```

```
#> Stepwise Model Path
#> Analysis of Deviance Table
#>
#> Initial Model:
#> QLT ~ FA + VA + CA + RS + CH + FSD + TSD + DEN + pH + SUL + ALC
#>
#> Final Model:
#> QLT ~ VA + CH + FSD + TSD + pH + SUL + ALC
#>
#>
#>    Step Df  Deviance Resid. Df Resid. Dev       AIC
#> 1                         1587   666.4107 -1375.489
#> 2 - DEN  1 0.2868924      1588   666.6976 -1376.801
#> 3  - FA  1 0.1079824      1589   666.8056 -1378.542
#> 4  - RS  1 0.2566805      1590   667.0623 -1379.926
#> 5  - CA  1 0.4748034      1591   667.5371 -1380.789
```

Analysis of the final Linear Regression model

Checking intercolliearity of the model

To estimate intercollinearity of the model we start with generating correlation matrix of our dataset attributes included in the final model. This is presented in Table 3.

To present the same information graphically, we generate pairwise scatter plots (Figure 3). The analysis show that attributes FSD and TSD have noticeable correlation at 0.67. This should be tested and fixed is necessary.

	QLT	VA	CH	FSD	TSD	pH	SUL	ALC
QLT	1.00	-0.39	-0.13	-0.05	-0.19	-0.06	0.25	0.48
VA	-0.39	1.00	0.06	-0.01	0.08	0.23	-0.26	-0.20
CH	-0.13	0.06	1.00	0.01	0.05	-0.27	0.37	-0.22
FSD	-0.05	-0.01	0.01	1.00	0.67	0.07	0.05	-0.07
TSD	-0.19	0.08	0.05	0.67	1.00	-0.07	0.04	-0.21
pH	-0.06	0.23	-0.27	0.07	-0.07	1.00	-0.20	0.21
SUL	0.25	-0.26	0.37	0.05	0.04	-0.20	1.00	0.09
ALC	0.48	-0.20	-0.22	-0.07	-0.21	0.21	0.09	1.00

Table 3: `Red Wines Quality Dataset Correlation Matrix`

Checking model skewness

The code below generates a histogram of residuals 2 for the Linear Regression model that have so far. From the chart we can say that the resulting histogram is skewed. This should be addressed later.

```
hist(residuals(wines_red_data.fit1), xlab = "", main = "")
```

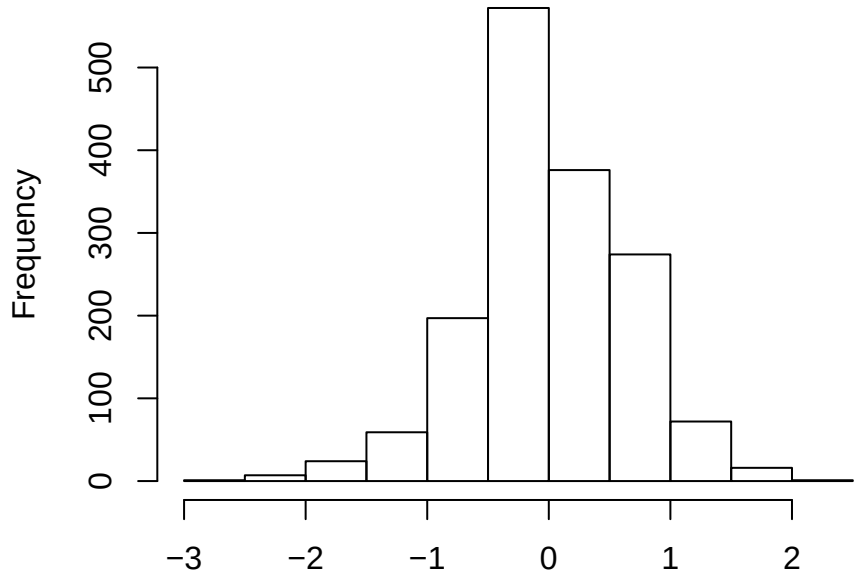

Figure 2: Histogram of Residual Errors

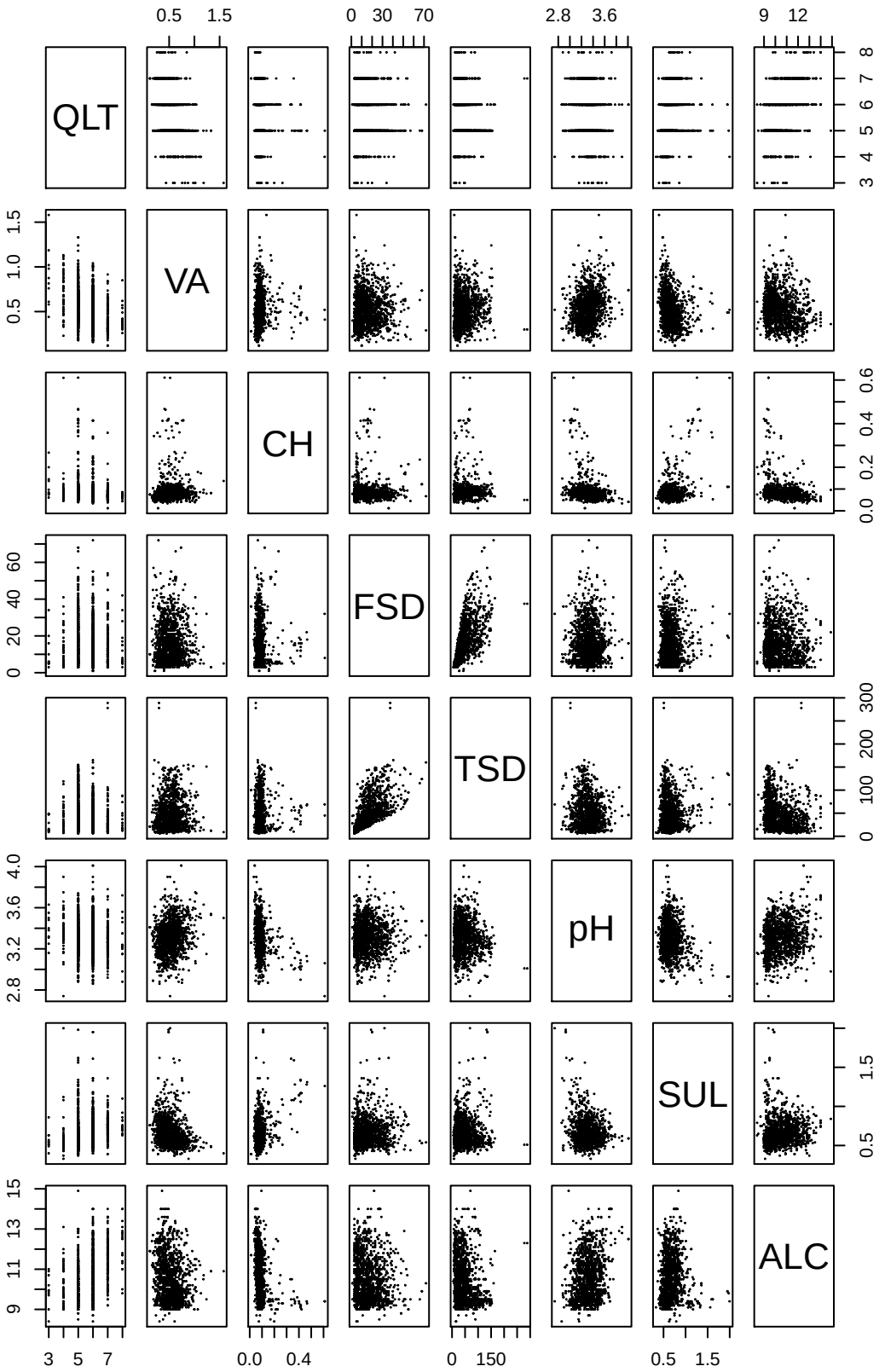

Figure 3: Relationships between the OSL model variables

Evaluate Nonlinearity via Component + Residual plots

A residual plot is a graph that shows the residuals on the vertical axis and the independent variable on the horizontal axis. If the points in a residual plot are randomly dispersed around the horizontal axis, a linear regression model is appropriate for the data; otherwise, a non-linear model is more appropriate. Our model's attributes are reasonably dispersed (see Figure 4) for most of the independent variables. Some pattern is shown for TSD and CH attributes. Code below generates the plot using srPlot function.

```
library(car)
crPlots(wines_red_data.fit1, layout = c(4,3), main = "")
```

Assessing Outliers

To evaluate the model we calculate Bonferonni p-value for most extreme observations (code below). Both p-value and Bonferroni-corrected p-value are smaller than 0.05, so the model is acceptable.

```
outlierTest(wines_red_data.fit)
```

```
#>      rstudent unadjusted p-value Bonferonni p
#> 833 -4.194088         2.8912e-05     0.046231
```

In addition we generate (code below) a QQ-Plot for Studentized Residuals. This graph is presented on Figure 5. Analyzing the chart we can see that our model has some deviation in the area of low QLT values. Most likely is explained by the low number of observation in than part of the dataset as was shown in Figure 1.

```
#> [1] 653 833
```

Another test model parameters is Leverage Plots (generated by the code below). Points further from the horizontal line than the slanted line effectively try to make the hypothesis test more significant, and those closer to the horizontal than the slanted line try to make the hypothesis test less significant. Figure 6 shows that our model is pretty well balanced and no collinearity between attributes.

```
leveragePlots(wines_red_data.fit, layout = c(4,3), main = "")
```

Test for Autocorrelated Errors

The Durbin Watson statistic is a number that tests for autocorrelation in the residuals from a statistical regression analysis. The Durbin-Watson statistic is always between 0 and 4. Values close to 2 low autocorrelation. Our model has the value of 1.76. Our model has low autocorrelation errors.

```
durbinWatsonTest(wines_red_data.fit)
```

```
#>  lag Autocorrelation D-W Statistic p-value
#>    1        0.121429       1.75714       0
#>  Alternative hypothesis: rho != 0
```

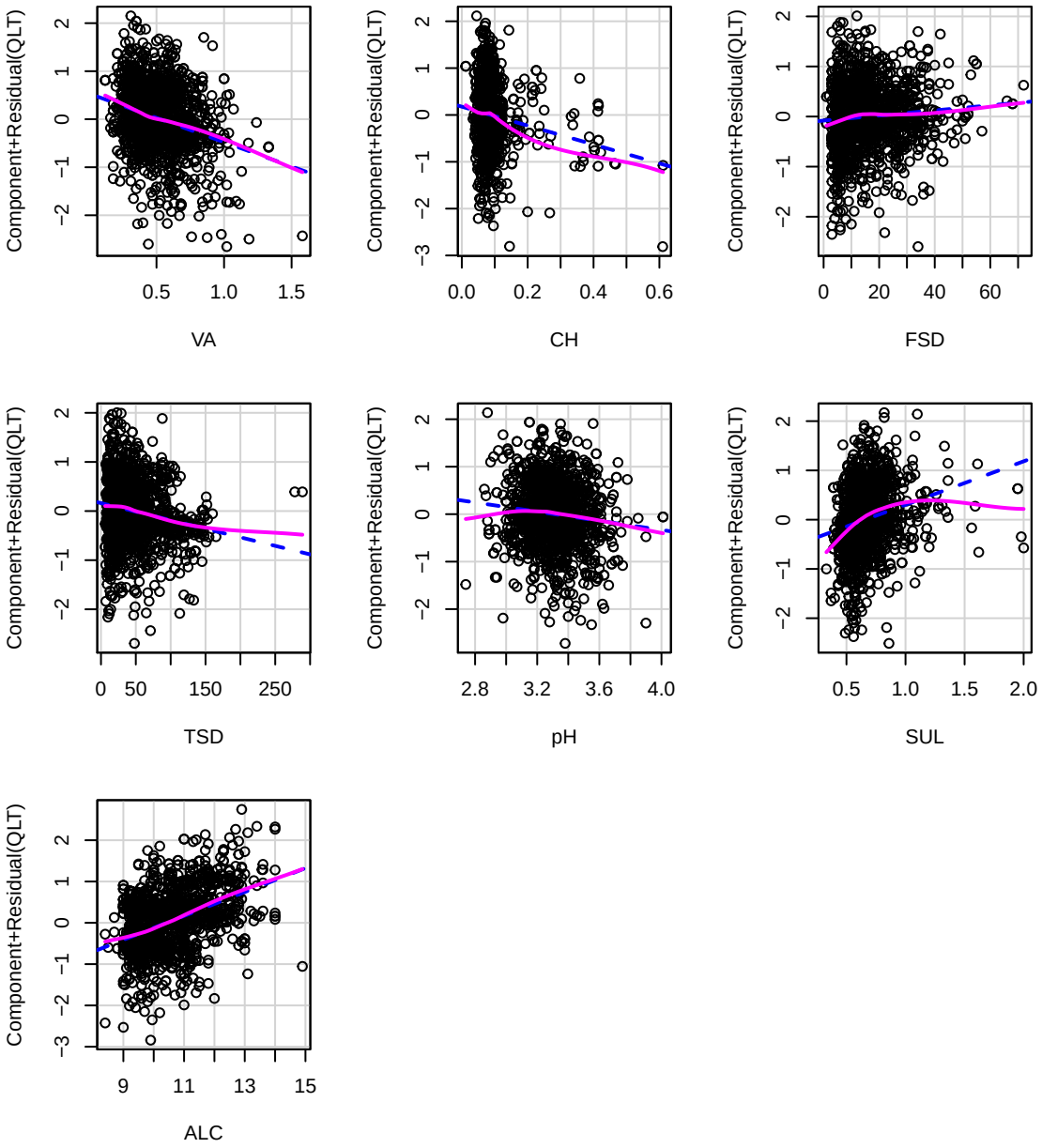

Figure 4: Component + Residuals Plot

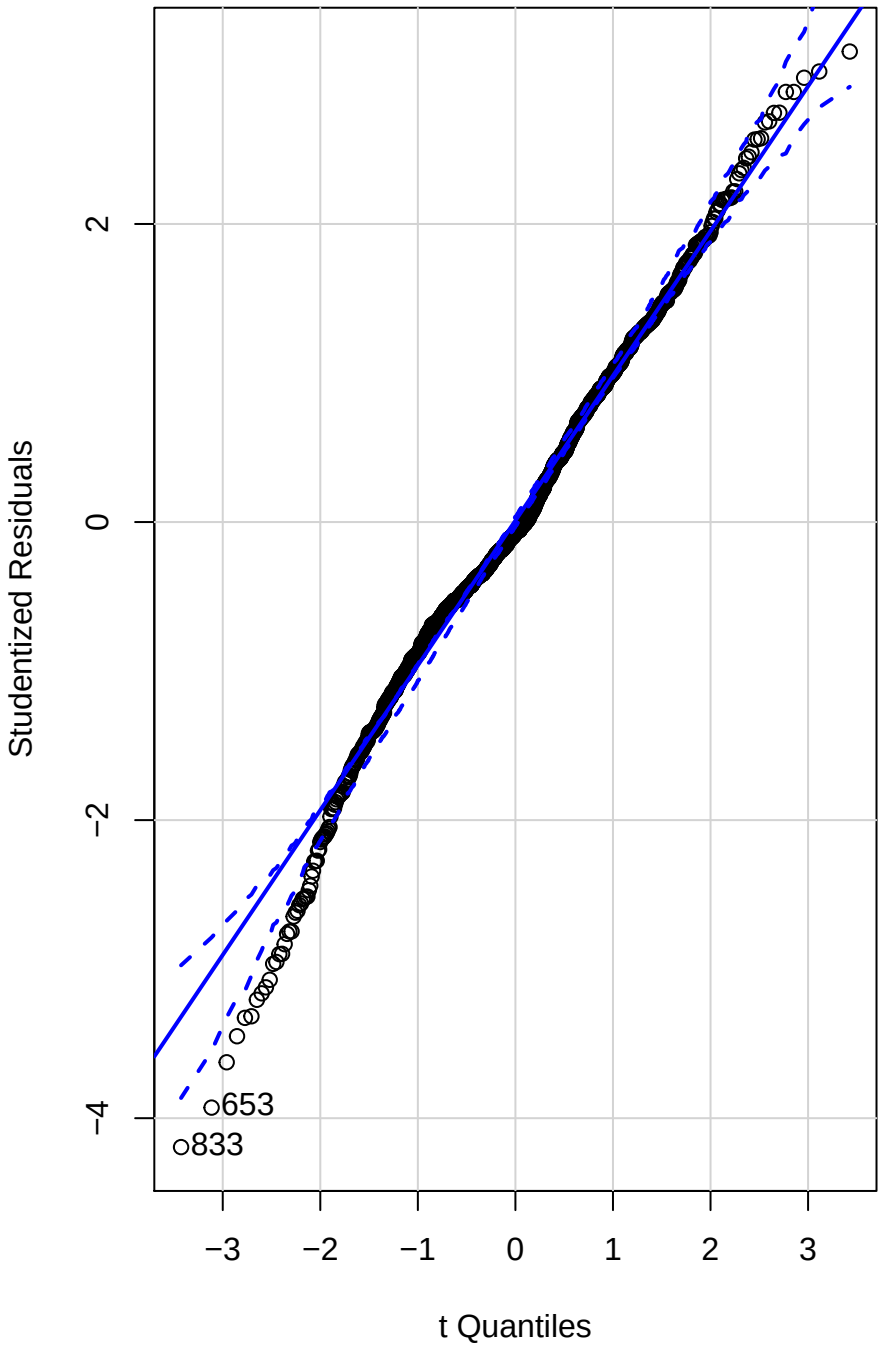

Figure 5: QQ Plot for Studentized Residuals

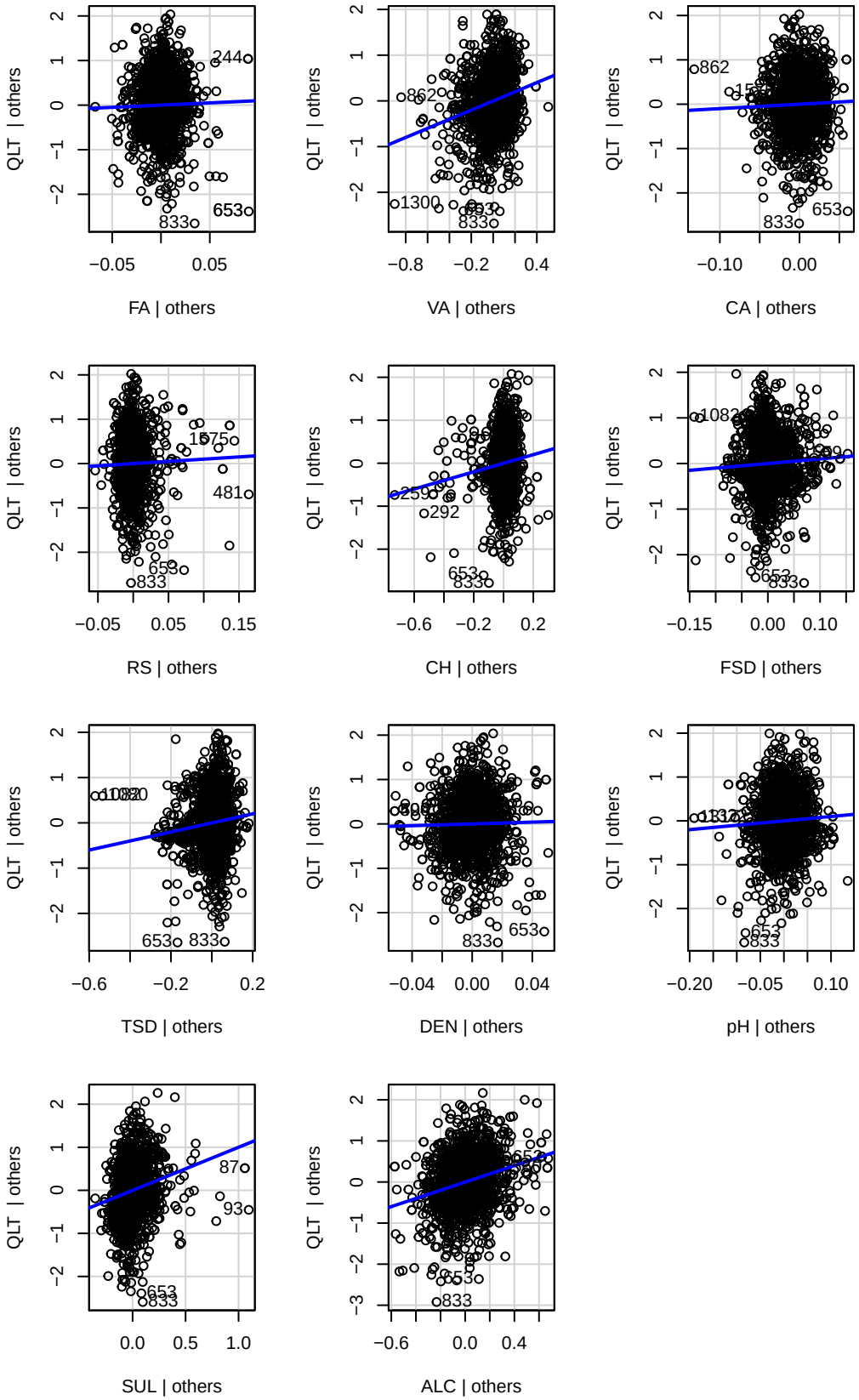

Figure 6: Red Wines - Leverage Plots

Global test of model assumptions

The following code tests for global model assumptions using gvmodel method (Peña and Slate, 2006) and is used for testing the four assumptions of the linear model. If the global procedure indicates a violation of at least one of the assumptions, then the components of the global test statistic can be used to gain insight into which assumptions have been violated. It can also be used in conjunction with associated deletion statistics to detect unusual observations.

Code below performs the tests and outputs the results. As we can see, 3 out of 5 global Linear Regression assumptions including the Skewness are not satisfied. We will address this in the next section.

```
library(gvlma)
gvmodel <- gvlma(wines_red_data.fit1)
summary(gvmodel)
```

```
#>
#> Call:
#> lm(formula = QLT ~ VA + CH + FSD + TSD + pH + SUL + ALC, data = wines_red_data)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -2.68918 -0.36757 -0.04653  0.46081  2.02954
#>
#> Coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  4.4300987  0.4029168  10.995  < 2e-16 ***
#> VA          -1.0127527  0.1008429 -10.043  < 2e-16 ***
#> CH          -2.0178138  0.3975417  -5.076 4.31e-07 ***
#> FSD          0.0050774  0.0021255   2.389    0.017 *
#> TSD         -0.0034822  0.0006868  -5.070 4.43e-07 ***
#> pH          -0.4826614  0.1175581  -4.106 4.23e-05 ***
#> SUL          0.8826651  0.1099084   8.031 1.86e-15 ***
#> ALC          0.2893028  0.0167958  17.225  < 2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.6477 on 1591 degrees of freedom
#> Multiple R-squared:  0.3595, Adjusted R-squared:  0.3567
#> F-statistic: 127.6 on 7 and 1591 DF,  p-value: < 2.2e-16
#>
#>
#> ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
#> USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
#> Level of Significance =  0.05
#>
#> Call:
#>  gvlma(x = wines_red_data.fit1)
#>
#>                      Value   p-value                   Decision
#> Global Stat        37.99895 1.121e-07 Assumptions NOT satisfied!
#> Skewness            6.45922 1.104e-02 Assumptions NOT satisfied!
#> Kurtosis           28.78544 8.086e-08 Assumptions NOT satisfied!
#> Link Function       2.70103 1.003e-01    Assumptions acceptable.
#> Heteroscedasticity  0.05326 8.175e-01    Assumptions acceptable.
```

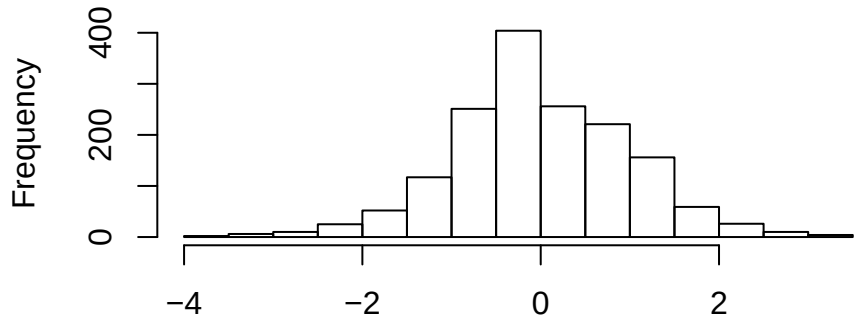

Figure 7: Histogram of residuals after correcting the skewness

Addressing Skewness Problem

To address the skewness of the model we will try to correct it applying the logarithmic transformation to the QLT target. The 1.25 parameter was obtained experimentally after several iterations. A histogram of residuals after correcting the skewness and presented in Figure 7

```
library(car)
summary(wines_red_data.fit2 <- lm (
  bcPower(QLT,1.25)  ~ VA + CH + FSD + TSD + pH + SUL + ALC, data=wines_red_data))

#>
#> Call:
#> lm(formula = bcPower(QLT, 1.25) ~ VA + CH + FSD + TSD + pH +
#>     SUL + ALC, data = wines_red_data)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -3.8872 -0.5845 -0.0839  0.7095  3.2585
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  4.258050   0.620654   6.861 9.79e-12 ***
#> VA          -1.538767   0.155339  -9.906  < 2e-16 ***
#> CH          -3.112317   0.612374  -5.082 4.17e-07 ***
#> FSD          0.007722   0.003274   2.358   0.0185 *
#> TSD         -0.005436   0.001058  -5.139 3.11e-07 ***
#> pH          -0.751939   0.181087  -4.152 3.46e-05 ***
#> SUL          1.372027   0.169303   8.104 1.05e-15 ***
#> ALC          0.452131   0.025872  17.475  < 2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.9978 on 1591 degrees of freedom
#> Multiple R-squared:  0.3625, Adjusted R-squared:  0.3597
#> F-statistic: 129.2 on 7 and 1591 DF,  p-value: < 2.2e-16
```

Re-testing Linear Regression global assumptions

Global assumptions were re-tested by the code below and the results presented. Results show that we fixed the skewness. Overall the model has Adjusted R-squared of 0.3597 which is slight improvement over previous values. Our model satisfied 3 out of 5 of the global Linear Regression assumptions. The Skewness problem has been resolved.

```
gvmodel <- gvlma(wines_red_data.fit2)
summary(gvmodel)
```

```
#>
#> Call:
#> lm(formula = bcPower(QLT, 1.25) ~ VA + CH + FSD + TSD + pH +
#>     SUL + ALC, data = wines_red_data)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -3.8872 -0.5845 -0.0839  0.7095  3.2585
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  4.258050   0.620654   6.861 9.79e-12 ***
#> VA          -1.538767   0.155339  -9.906  < 2e-16 ***
#> CH          -3.112317   0.612374  -5.082 4.17e-07 ***
#> FSD          0.007722   0.003274   2.358   0.0185 *
#> TSD         -0.005436   0.001058  -5.139 3.11e-07 ***
#> pH          -0.751939   0.181087  -4.152 3.46e-05 ***
#> SUL          1.372027   0.169303   8.104 1.05e-15 ***
#> ALC          0.452131   0.025872  17.475  < 2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.9978 on 1591 degrees of freedom
#> Multiple R-squared:  0.3625, Adjusted R-squared:  0.3597
#> F-statistic: 129.2 on 7 and 1591 DF,  p-value: < 2.2e-16
#>
#>
#> ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
#> USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
#> Level of Significance =  0.05
#>
#> Call:
#>  gvlma(x = wines_red_data.fit2)
#>
#>                      Value   p-value                   Decision
#> Global Stat        24.80148 5.515e-05 Assumptions NOT satisfied!
#> Skewness            0.50847 4.758e-01    Assumptions acceptable.
#> Kurtosis           20.50362 5.952e-06 Assumptions NOT satisfied!
#> Link Function       3.77416 5.205e-02    Assumptions acceptable.
#> Heteroscedasticity  0.01524 9.018e-01    Assumptions acceptable.
```

Tree-based regression methods

Tree-based methods, while simple and useful for interpretation, are typically not as competitive with the best supervised learning approaches such as polynomial regression. However, tree-based methods such as regression tree and random forests make up for this shortfall. By combining a large number of trees instead of one, the model usually results in dramatic improvements in terms of prediction accuracy. This improvement in accuracy comes at the expense of loss in interpretation.

Splitting the dataset into train and test

The dataset has been split in such a way that train and test sets would have the same distribution of the 'QLT' attribute. The reason for this stratification strategy is to focus on the priority on the target value. We used 60:34 split ratio. The code below also checks for distribution of QLT in train and test sets. As we can see, those values match well.

```
library(caret)
train.rows<- createDataPartition(y= wines_red_data$QLT, p=0.6, list = FALSE)
train.data<- wines_red_data[train.rows,]
prop.table((table(train.data$QLT)))

#>
#>           3           4           5           6           7           8
#> 0.005202914 0.029136316 0.430801249 0.398543184 0.124869927 0.011446410

test.data<- wines_red_data[-train.rows,]
prop.table((table(test.data$QLT)))

#>
#>           3           4           5           6           7           8
#> 0.007836991 0.039184953 0.418495298 0.399686520 0.123824451 0.010971787
```

Regression Tree fit

The Regression Tree method (Therneau and Atkinson, 2018) is known to be computationally fast but not very precise, we have use all default parameters and all attributes of the train dataset in the resulting Decision Tree presented in Figure 8. In a regression tree, the tree arranges or segments observations into regions of a predictor space. Regression tree, since the target variable is a real valued number, fits a regression model to the target variable using each of the independent variables. Then for each independent variable, the data is split at several split points. It calculates Sum of Squared Error(SSE) at each split point between the predicted value and the actual values. The variable resulting in minimum SSE is selected for the node. Then this process is recursively continued till the entire data is covered.

```
library(rpart)
library(rpart.plot)
library(rattle)
library(caret)

reg.tree <- rpart(QLT ~ ., method="anova", data = train.data)
fancyRpartPlot(reg.tree, main="", sub="")
```

Regression Tree model evaluation

This Decision Tree favors the following attributes in order of their importance for the prediction of the target attribute: ALC, VA, SUL, TSD and FSDs. It does not consider the rest of the attributes as important. Let's apply the model to the test set and evaluate accuracy of the predictions. As we can see, the accuracy of RT at 0.5480833 is 50% higher than OLS.

```
dtPrediction <- predict(reg.tree, test.data)
cor(dtPrediction,test.data$QLT)

#> [1] 0.5480833
```

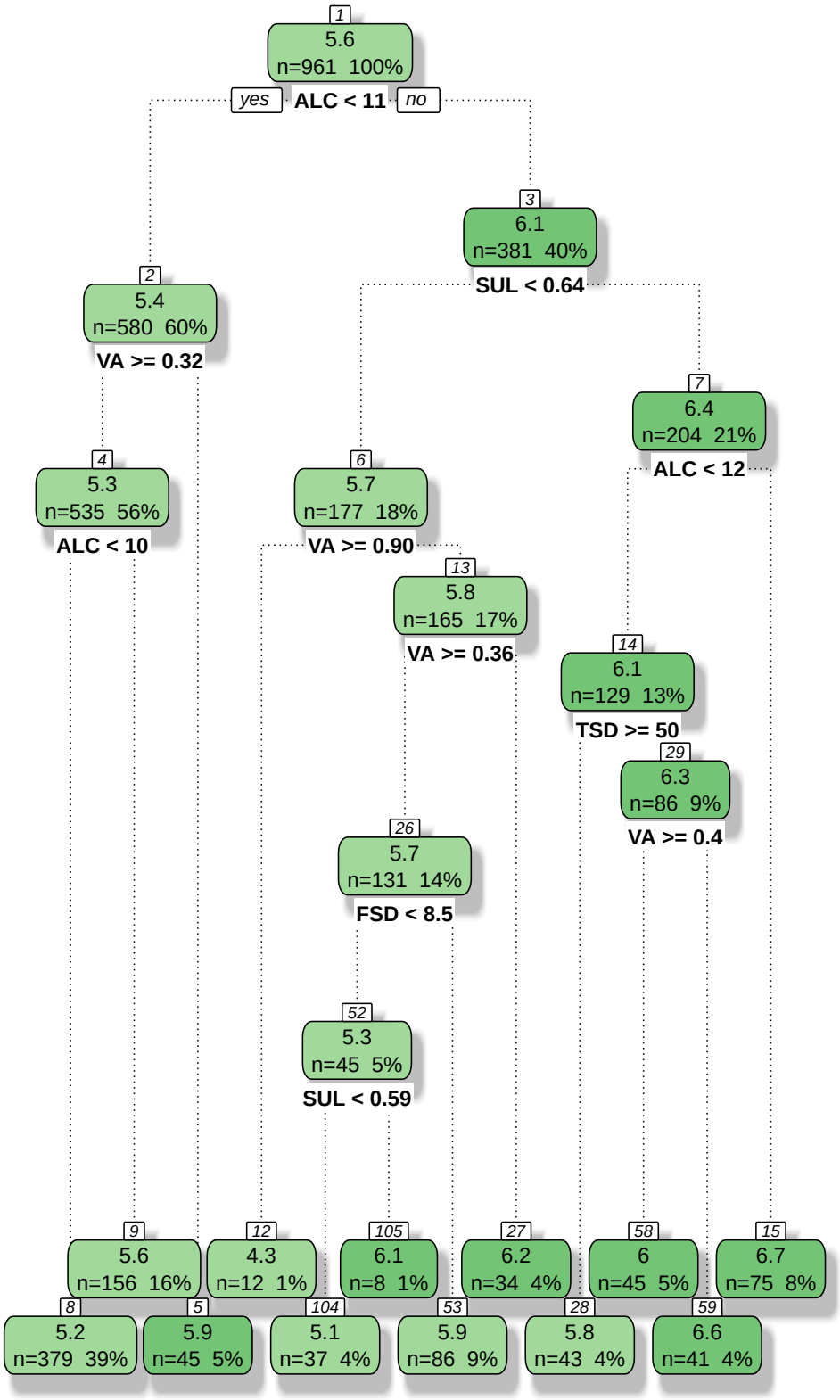

Figure 8: Regression Tree Diagram

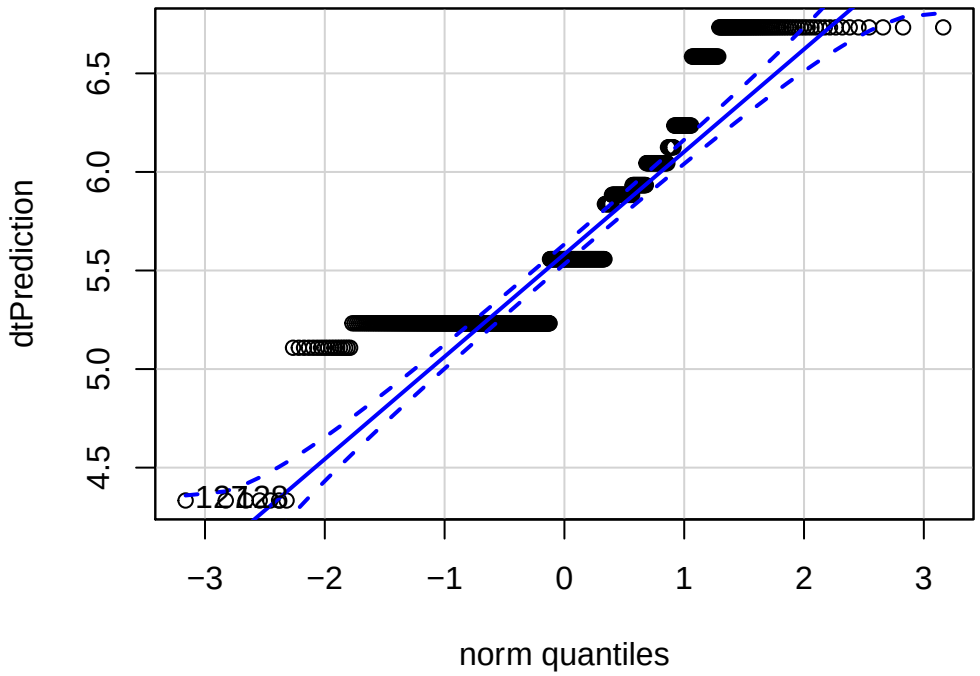

Figure 9: Regression Tree Prediction QQ Plot

To visualize the results we generated (code below) a QQ-Plot for Studentized Residuals. This graph is presented on Figure 9. Analyzing the chart we can see that our model has comparable deviations in the area of a low, middle and high QLT values.

```
qqPlot(dtPrediction, main="" )
```

```
#> 127 128
#>  44  45
```

In addition we generated a visual representation of the RT method prediction using a scatter plot in Figure 10.

```
library(ggplot2)
df1 = data.frame(as.factor(test.data$QLT), dtPrediction)
colnames(df1) <- c("Test","Prediction")
ggplot(df1, aes(x = Test, y = Prediction)) +
        geom_boxplot(outlier.colour = "red") +
        geom_jitter(width = 0.25, pch=20, col=rgb(0.1, 0.2, 0.8, 0.3))
```

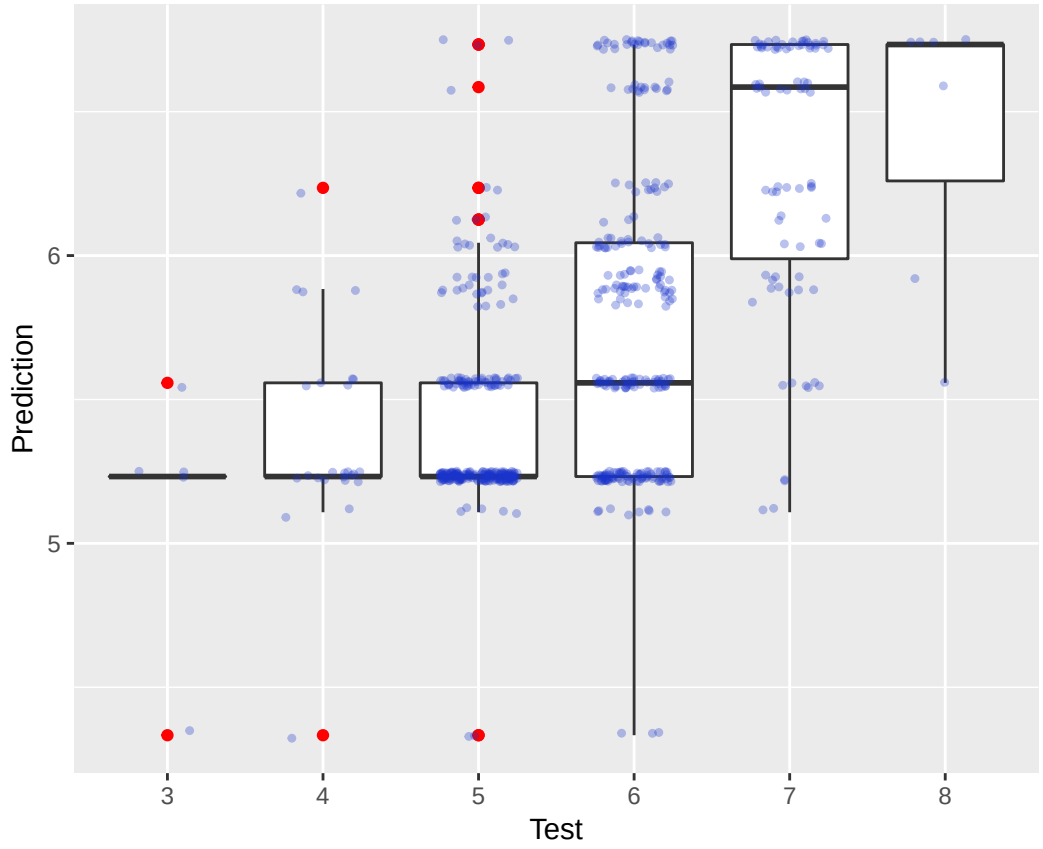

Figure 10: Regression Tree Prediction

Random Forest model fit

Next step is to use more advanced but more computationally demanding Random Forest Regressor (Liaw and Wiener, 2002). Random forests can be used for regression analysis and are in fact called Regression Forests. They are an ensemble of different Regression Trees and are used for nonlinear multiple regression. Each leaf contains a distribution for the continuous output variables.

Random Forest Regressor is an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

In addition, Random Forest Regressor algorithms can produce a graph explaining the importance independent attributes in the output result generation. The code below runs the calculations.

```
library(randomForest)
fitRF1 <- randomForest(
  QLT ~ ., method="anova",
  data=train.data, importance=TRUE, ntree=1000)
```

Importance of the dataset attributes for the prediction of the "class" attribute shown in Figure 11. Analyzing this chart we conclude that the proper order of attributes importance for the prediction of the target attribute is: ALC, SUL VA, TSD, DEN, FA, CA, CH, FSD, pH and RS. It contradicts the absolute values of the Linear Regression coeffitients assigned to independent attributes, i.e. SUL at 0.88 seems to be more important than ALC at 0.29. This difference could be explained by limitation of OLS .

```
varImpPlot(fitRF1, main="")
```

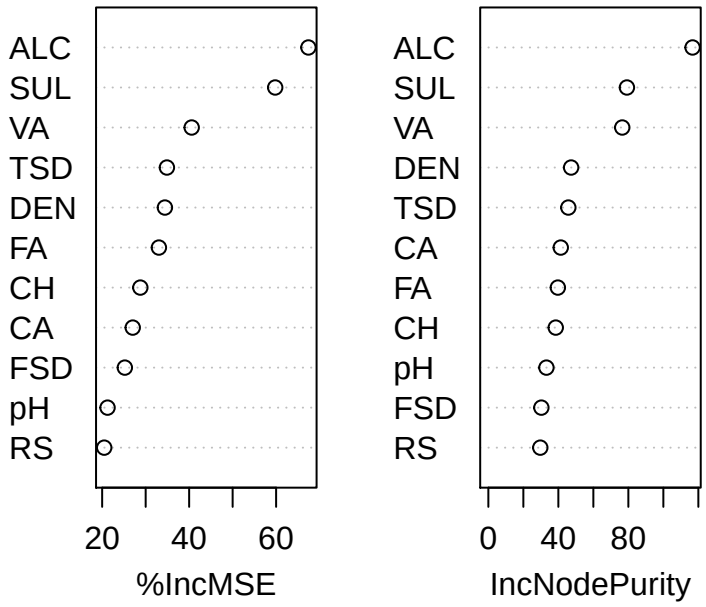

Figure 11: Importance of the dataset attributes for the prediction of the 'class' attribute

Random Forest Regressor prediction and evaluation

Code below applies fitted Random Forest Regressor model to the test data. The calculations show that RFR gives 80% better accuracy at 0.6734 than OLS at 0.3597.

```
PredictionRF1 <- predict(fitRF1, test.data)
cor(PredictionRF1,test.data$QLT)

#> [1] 0.6757805
```

To visualize the results of the predictions, the code below generates a scatter plot of the Predictor vs Test values (Figure 12). Also a QQ-Plot for Studentized Residuals is generated and presented in Figure 13.

```
library(ggplot2)
df2 = data.frame(as.factor(test.data$QLT), PredictionRF1)
colnames(df2) <- c("Test","Prediction")
ggplot(df2, aes(x = Test, y = Prediction)) +
        geom_boxplot(outlier.colour = "red") +
        geom_jitter(width = 0.25, pch=20, col=rgb(0.1, 0.2, 0.8, 0.3))


qqPlot(PredictionRF1, main="")


#> 199 128
#>  75  45
```

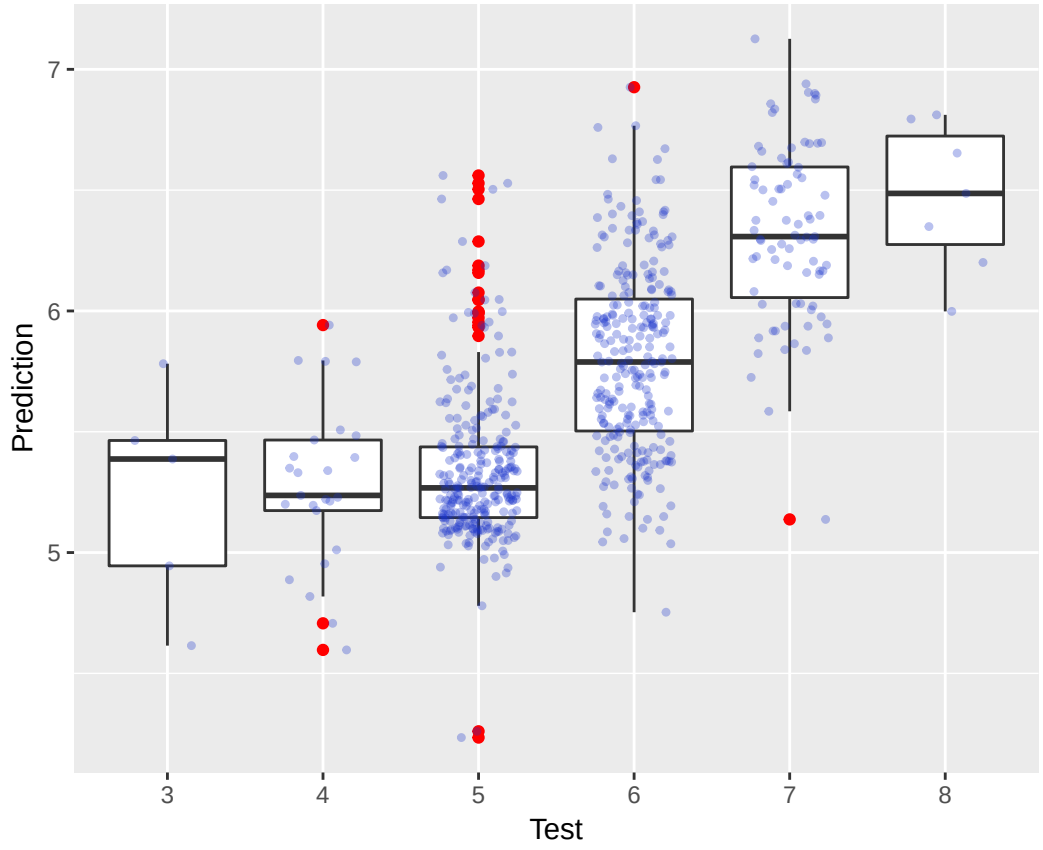

Figure 12: Random Forest Prediction

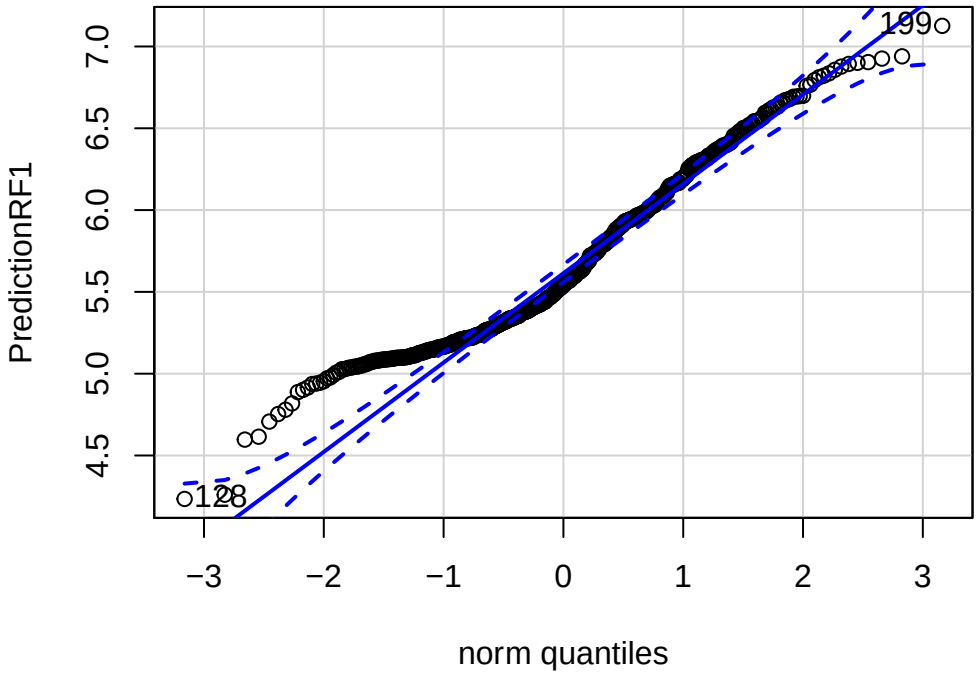

Figure 13: Random Forest Prediction QQ Plot

Conclusion

Through exploring the Red Wine Quality dataset and using a different methods of Linear Regression we developed an algorithm to predict the wine quality using its chemical characteristics.

First we applied the Linear Regression OLS method and through several steps of correcting the model and adjusting for skewness and then comparing it to automated Stepwise Regression method we achieved the accuracy of RT at 0.5480833 is 50% higher than OLS.

Next we applied tree-based regression methods. First we used a Regression Tree which gave us accuracy of 0.5480833. The final method was the Random Forest Regressor and achieved 80% better accuracy at 0.6734 comparing to OLS.

The project was a success, however, none of the Linear Regression methods used would give us reliable precision. All the plots show the presence of outliers and inaccuracy in the areas of low and high scores. We conclude that the current problem could not be solved by the Linear Regression methods only, it looks that additional methods like Clustering is required to split the dataset into smaller sets to satisfy Linear Regression limitations.

Bibliography

P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 1998. [p1, 2]

D. B. Dahl. *xtable: Export Tables to LaTeX or HTML*, 2016. URL https://CRAN.R-project.org/package=xtable. R package version 1.8-2. [p2]

FAOSTAT. Faostat. URL http://faostat.fao.org/site/535/DesktopDefault.aspx?PageID=535. [p1]

A. Legin, A. Rudnitskaya, L. Lvova, Y. Vlasov, C. Di Natale, and A. D'Amico. Evaluation of Italian wine by the electronic tongue: recognition, quantitative analysis and correlation with human sensory perception. *Analytica Chimica Acta*, 484(1):33–44, May 2003. ISSN 00032670. doi: 10.1016/S0003-2670(03)00301-5. URL http://linkinghub.elsevier.com/retrieve/pii/S0003267003003015. [p1]

A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002. URL https://CRAN.R-project.org/doc/Rnews/. [p18]

E. A. Peña and E. H. Slate. Global validation of linear model assumptions. *Journal of the American Statistical Association*, 101(473):341–354, Mar. 2006. ISSN 0162-1459, 1537-274X. doi: 10.1198/016214505000000637. URL http://www.tandfonline.com/doi/abs/10.1198/016214505000000637. [p12]

R. Teranishi, E. L. Wick, and I. Hornstein, editors. *Flavor chemistry: thirty years of progress*. Kluwer Academic/Plenum Publishers, New York, 1999. ISBN 9780306461996. [p1]

T. Therneau and B. Atkinson. *rpart: Recursive Partitioning and Regression Trees*, 2018. URL https://CRAN.R-project.org/package=rpart. R package version 4.1-13. [p15]

UCI Wine Data Set. Uci machine learning repository: wine quality data set. URL http://archive.ics.uci.edu/ml/datasets/Wine+Quality. [p1, 2]

Note from the Authors

This file was generated using *The R Journal* style article template, additional information on how to prepare articles for submission is here - Instructions for Authors. The article itself is an executable R Markdown file that could be downloaded from Github with all the necessary artifacts.

Viviane Adohouannon
York University School of Continuing Studies

https://learn.continue.yorku.ca/user/view.php?id=21444

Kate Alexander
York University School of Continuing Studies

https://learn.continue.yorku.ca/user/view.php?id=21524

Diana Azbel
York University School of Continuing Studies

https://learn.continue.yorku.ca/user/view.php?id=20687

Igor Baranov
York University School of Continuing Studies

https://learn.continue.yorku.ca/user/profile.php?id=21219