SCI3501 Assignment:

Analysing the Physicochemistry and Perceived Quality of Vinho Verde Wines from North Portugal

Yesahel Scicluna

2023-01-12

Preparing the data for analysis

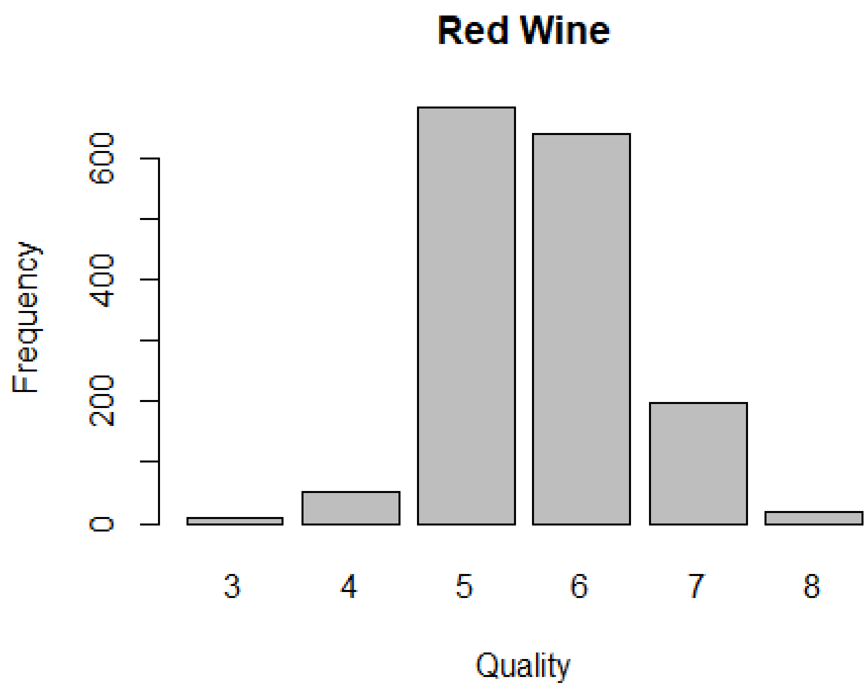
```
## Importing the data

library(readxl)
winequality_red_edited <- read_excel("C:/Users/zache/Desktop/R Assignment/0.
Raw data/winequality.red.edited.xlsx")
View(winequality_red_edited)

library(readxl)
winequality_white_edited <- read_excel("C:/Users/zache/Desktop/R
Assignment/0. Raw data/winequality.white.edited.xlsx")
View(winequality_white_edited)

## Removing the variables that will not be considered

winequality_red_edited$'fixed acidity'<-NULL
winequality_red_edited$'volatile acidity'<-NULL
winequality_red_edited$'citric acid'<-NULL
winequality_red_edited$'residual sugar'<-NULL
winequality_red_edited$'chlorides'<-NULL
winequality_red_edited$'free sulfur dioxide'<-NULL
winequality_red_edited$'total sulfur dioxide'<-NULL

winequality_white_edited$'fixed acidity'<-NULL
winequality_white_edited$'volatile acidity'<-NULL
winequality_white_edited$'citric acid'<-NULL
winequality_white_edited$'residual sugar'<-NULL
winequality_white_edited$'chlorides'<-NULL
winequality_white_edited$'free sulfur dioxide'<-NULL
winequality_white_edited$'total sulfur dioxide'<-NULL

## Creating a merged data set

winequality_merged<-rbind(winequality_red_edited, winequality_white_edited)
View(winequality_merged)
```

```r
## Reordering the columns

winequality_merged<-winequality_merged[,c(1,3:5,2,6)]

## Designating colour as a categorical variable

winequality_red_edited$colour <- factor(winequality_red_edited$colour)
levels(winequality_red_edited$colour)<-c("red", "white")

winequality_white_edited$colour <- factor(winequality_white_edited$colour)
levels(winequality_white_edited$colour)<-c("red", "white")

winequality_merged$colour <- factor(winequality_merged$colour)
levels(winequality_merged$colour)<-c("red", "white")

## Converting pH into a categorical variable, 'acidity'

acidity<-cut(winequality_merged$pH,breaks=c(-Inf, 3.5,
Inf),labels=c("strongly acidic","weakly acidic"),right=FALSE)
winequality_merged<-cbind(winequality_merged, acidity)
winequality_red_edited$'pH'<-NULL
winequality_white_edited$'pH'<-NULL
winequality_merged$'pH'<-NULL
```

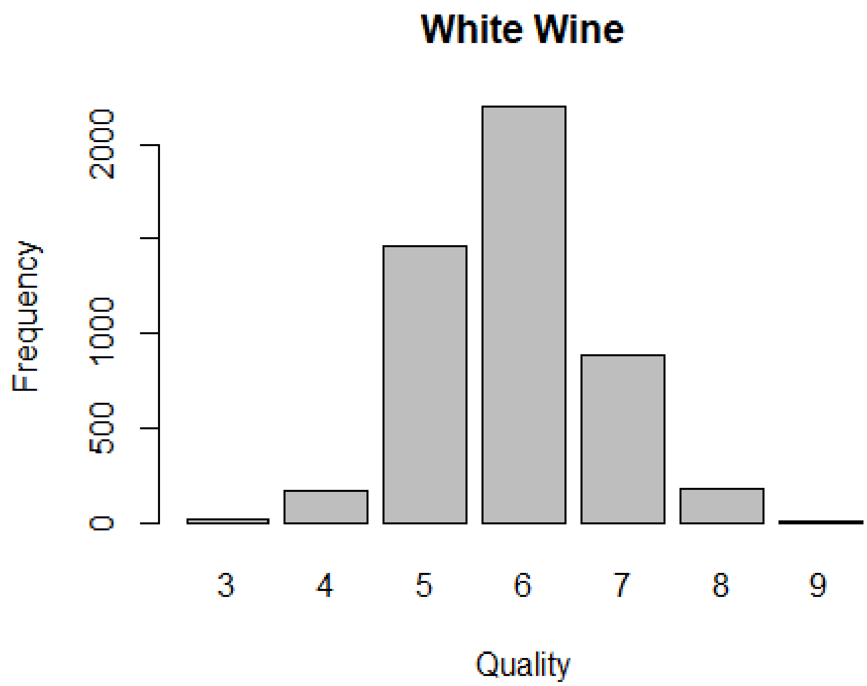
Exploratory analysis related to question 1

```r
## Checking the means and standard deviations of the scores for red and white
wines

summary(winequality_red_edited$quality)

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

sd(winequality_red_edited$quality)

## [1] 0.8075694

summary(winequality_white_edited$quality)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.878   6.000   9.000

sd(winequality_white_edited$quality)

## [1] 0.8856386

## Checking the distribution of scores for red and white wines
```

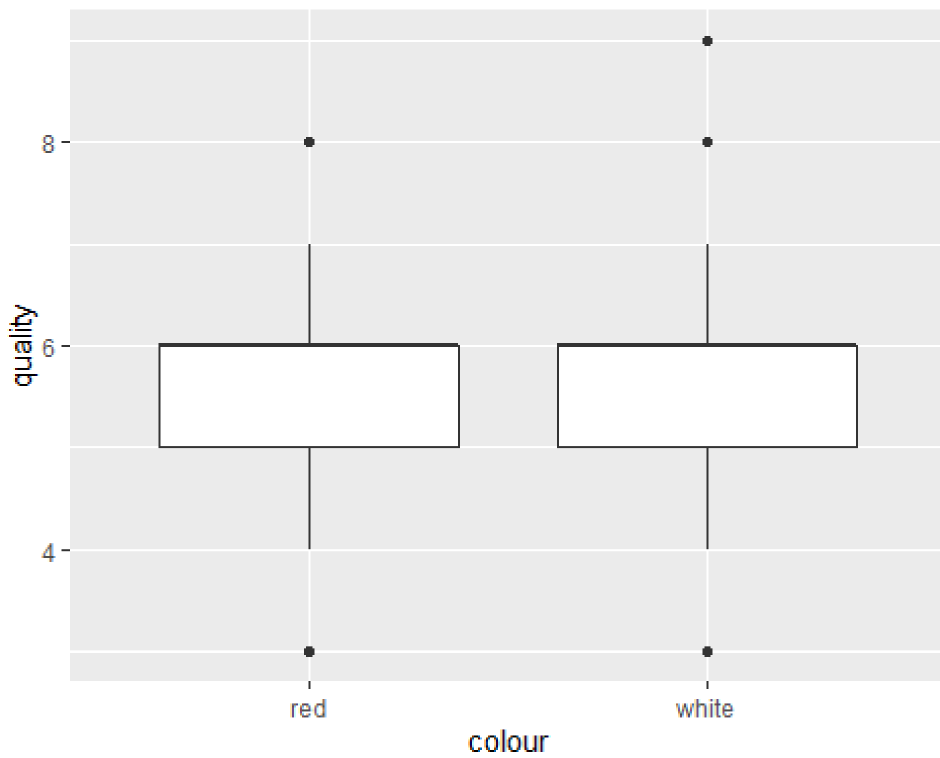
```r
frequency_redwine_quality <- table(winequality_red_edited$quality)
redwine_barchart<-barplot(frequency_redwine_quality, main="Red Wine",
xlab="Quality", ylab="Frequency",
names.arg=levels(winequality_red_edited$quality))
```




```r
frequency_whitewine_quality <- table(winequality_white_edited$quality)
whitewine_barchart<-barplot(frequency_whitewine_quality, main="White Wine",
xlab="Quality", ylab="Frequency",
names.arg=levels(winequality_white_edited$quality))
```

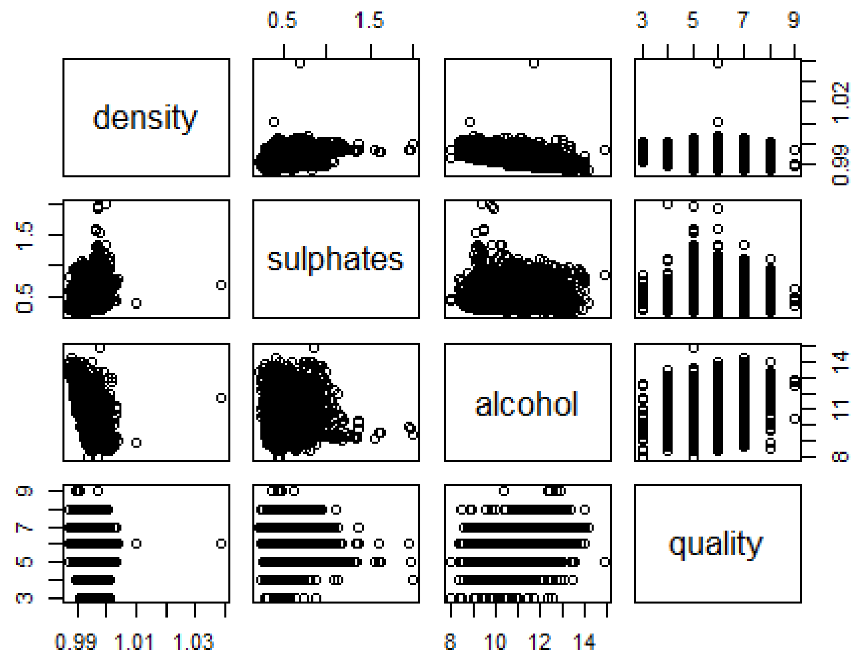



```
## Comparing the average scores of red and white wines

library(ggplot2)
score_boxplot <- ggplot(winequality_merged, aes(x=colour, y=quality, na.rm =
TRUE)) + geom_boxplot(na.rm = TRUE)
score_boxplot
```



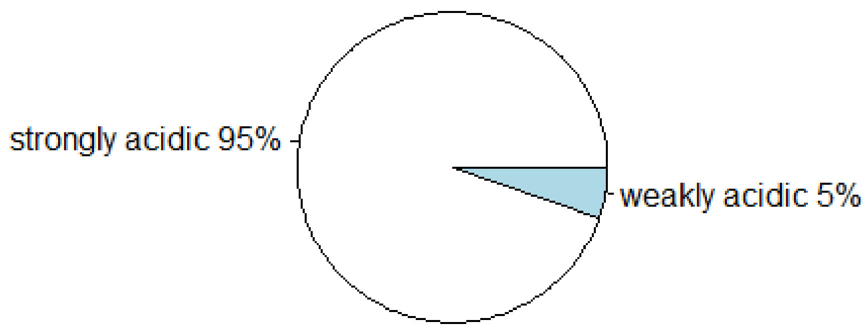
Exploratory analysis related to question 2

```
## Checking for any potential relationships between the different
quantitative variables

pairs(winequality_merged[1:4])
```

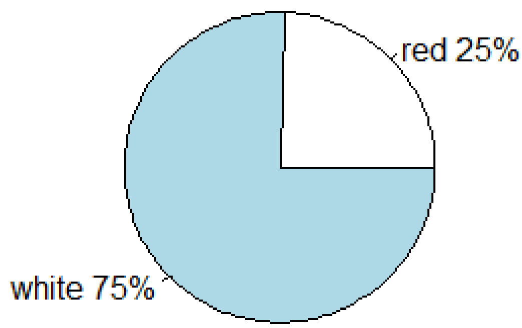


Exploratory analysis related to question 3

```
## Checking the frequency of stronger/weaker wines and red/white wines

summary(winequality_merged$acidity)

## strongly acidic   weakly acidic
##            6159             338

summary(winequality_merged$colour)

##   red white
##  1599  4898

## Expressing the above frequencies as percentages

slices<-summary(winequality_merged$acidity)
lbls<-levels(winequality_merged$acidity)
prcnt<-round(slices/sum(slices)*100)
lbls<-paste(lbls, prcnt)
lbls <- paste(lbls,"%",sep="")
pie(slices, labels=lbls)
```




```
slices2<-summary(winequality_merged$colour)
lbls2<-levels(winequality_merged$colour)
prcnt2<-round(slices2/sum(slices2)*100)
lbls2<-paste(lbls2, prcnt2)
lbls2<- paste(lbls2,"%",sep="")
pie(slices2, labels=lbls2)
```



Data analysis related to question 1

```
## Testing if the red and white wine scores are normally distributed

by(winequality_merged$quality, winequality_merged$colour, shapiro.test)

## winequality_merged$colour: red
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.85759, p-value < 2.2e-16
##
## ------------------------------------------------------------
## winequality_merged$colour: white
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.88904, p-value < 2.2e-16

## Testing whether the red and white wines are equal in variance

library(lawstat)
levene.test(winequality_merged$quality,winequality_merged$colour,location='me
an')
```

```
##
##  Classical Levene's test based on the absolute deviations from the mean
##  ( none not applied because the location is not set to median )
##
## data:  winequality_merged$quality
## Test Statistic = 0.62133, p-value = 0.4306
```

```
## Testing if the average red and white wine scores are different

wilcox.test(quality ~ colour, data=winequality_merged, exact=FALSE)

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  quality by colour
## W = 3311514, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Data analysis related to question 2

```
## Testing if density, alcohol, sulfates and score exhibit multivariate
normal distribution

library(energy)
mvnorm.etest(winequality_merged[,1:4],R=200)

##
##  Energy test of multivariate normality: estimated parameters
##
## data:  x, sample size 6497, dimension 4, replicates 200
## E-statistic = 69.196, p-value < 2.2e-16

## Testing for correlations between density, alcohol, sulfates and score
library(Hmisc)

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units

corr_data<-as.matrix(winequality_merged[c(1:4)])
rcorr(corr_data, type="spearman")
```

```
##           density sulphates alcohol quality
## density      1.00      0.27   -0.70   -0.32
## sulphates    0.27      1.00    0.00    0.03
## alcohol     -0.70      0.00    1.00    0.45
## quality     -0.32      0.03    0.45    1.00
##
## n= 6497
##
##
## P
##           density sulphates alcohol quality
## density           0.0000    0.0000  0.0000
## sulphates 0.0000            0.7119  0.0162
## alcohol   0.0000  0.7119            0.0000
## quality   0.0000  0.0162    0.0000

## Fitting a multiple linear regression model

y<-winequality_merged[,4]
x<-winequality_merged[,2:3]
mlrmodel<-lm(y~.,x)
summary(mlrmodel)

##
## Call:
## lm(formula = y ~ ., data = x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4667 -0.4953 -0.0349  0.5072  3.2282
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.280158   0.092678  24.603  < 2e-16 ***
## sulphates   0.233749   0.065175   3.586 0.000338 ***
## alcohol     0.325400   0.008131  40.018  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7817 on 6494 degrees of freedom
## Multiple R-squared:  0.199,  Adjusted R-squared:  0.1988
## F-statistic: 806.7 on 2 and 6494 DF,  p-value: < 2.2e-16

## Testing if the residuals exhibit normal distribution

library(MASS)
mlrresiduals<-studres(mlrmodel)
library(nortest)
ad.test(mlrresiduals)
```

```
##
##  Anderson-Darling normality test
##
## data:  mlrresiduals
## A = 29.757, p-value < 2.2e-16
```

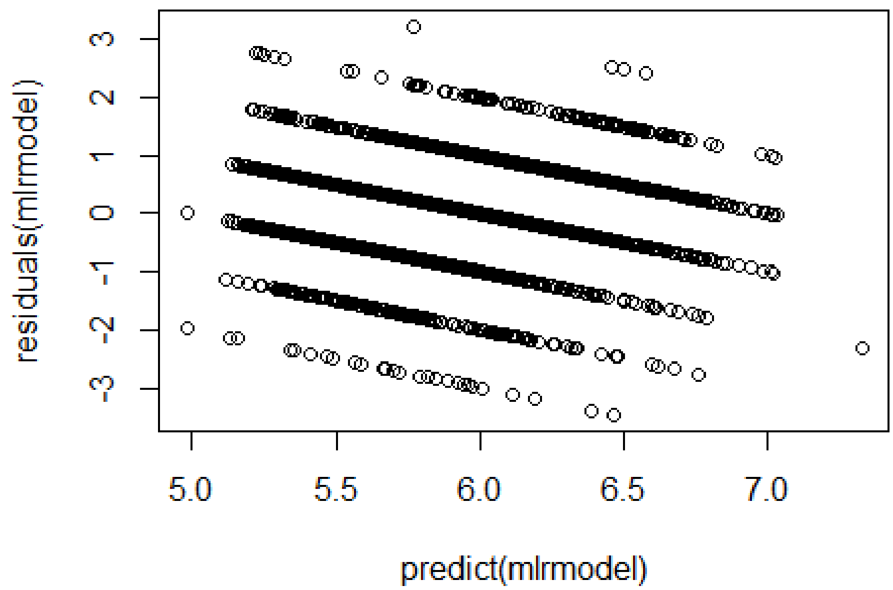
```r
## Testing if the residuals are independent of each other

library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
dwtest(mlrmodel)
```

```
##
##  Durbin-Watson test
##
## data:  mlrmodel
## DW = 1.6245, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

```r
## Visualising the homoscedasticity of the residuals

plot(predict(mlrmodel),residuals(mlrmodel))
```




```
## Testing the homoscedasticity of the residuals

library(lmtest)
bptest(mlrmodel)

##
##  studentized Breusch-Pagan test
##
## data:  mlrmodel
## BP = 23.419, df = 2, p-value = 8.215e-06

# Checking for outliers using Mahalanobis distances

m_dist<-mahalanobis(x, colMeans(x), cov(x))
cutof_mah<-qchisq(0.95, 2, lower.tail = TRUE, log.p = FALSE)
cutof_mah

## [1] 5.991465

out_mah<-which(m_dist>cutof_mah)
out_mah

##   [1]   14   15   16   18   20   23   28   43   44   70   80   82   84
87   89
##  [16]   92   93  107  111  115  129  143  145  152  162  170  182  198
199  202
##  [31]  211  227  241  246  250  259  268  269  270  272  277  278  279
```

```
282  290
## [46]  336  339  340  341  347  348  349  351  354  362  366  370  372
373  377
## [61]  378  379  391  416  424  431  435  439  445  452  456  466  468
475  478
## [76]  482  483  484  485  489  492  493  502  503  504  505  507  516
521  523
## [91]  545  571  587  589  615  618  624  640  653  690  693  724  755
774  775
## [106]  796  803  809  822  853  920  923  927  947  983 1052 1054 1071
1094 1099
## [121] 1115 1121 1127 1133 1151 1158 1159 1166 1167 1168 1208 1229 1261
1270 1271
## [136] 1289 1290 1320 1368 1371 1372 1373 1404 1406 1407 1408 1409 1410
1413 1414
## [151] 1430 1476 1478 1517 1523 1571 1589 2301 2358 2359 2452 2454 2466
2574 2616
## [166] 2699 2726 2838 2842 2843 2883 2893 2986 2994 3064 3190 3203 3341
3462 3552
## [181] 4003 4041 4194 4234 4237 4268 4348 4350 4393 4396 4472 4473 4474
4493 4517
## [196] 4522 4530 4545 4598 4608 4656 4657 4683 4686 4750 4806 4825 4844
4884 4891
## [211] 4901 4903 5058 5068 5076 5082 5083 5084 5103 5107 5114 5117 5119
5224 5255
## [226] 5271 5273 5276 5310 5328 5335 5336 5354 5364 5373 5451 5501 5504
5510 5515
## [241] 5516 5518 5531 5598 5599 5600 5612 5665 5729 5749 5795 5839 6001
6046 6062
## [256] 6091 6103 6145 6152 6160 6182 6196 6217 6245 6258 6281 6296 6356
6357 6389
## [271] 6392 6415 6418 6465 6486 6487

length(out_mah)

## [1] 276

## Checking for outliers using Leverages

cutof_lev<-2*3/(length(y))
cutof_lev

## [1] 0.0009235032

leverages<-as.data.frame(hatvalues(mlrmodel, type='rstandard'))
out_lev<-which(leverages>cutof_lev)
out_lev

##   [1]   14   15   16   18   20   23   28   43   44   70   80   82   84
87   89
##  [16]   92   93  107  111  115  129  143  145  152  162  170  182  198
```

```
199  202
##  [31]  210  211  227  241  244  245  246  250  259  265  268  269  270
272  277
##  [46]  278  279  282  290  336  339  340  341  342  347  348  349  350
351  354
##  [61]  357  362  364  366  370  372  373  376  377  378  379  391  416
424  431
##  [76]  435  439  445  452  456  466  468  475  478  482  483  484  485
489  492
##  [91]  493  502  503  504  505  506  507  516  521  523  531  536  545
571  587
## [106]  589  607  615  618  624  640  653  690  693  724  755  774  775
796  803
## [121]  806  808  809  822  829  833  834  853  897  899  911  920  923
926  927
## [136]  939  947  966  971  972  983 1003 1007 1008 1017 1039 1052 1054
1071 1094
## [151] 1099 1101 1108 1115 1119 1121 1127 1133 1147 1151 1158 1159 1166
1167 1168
## [166] 1193 1208 1210 1218 1229 1261 1268 1270 1271 1288 1289 1290 1320
1368 1371
## [181] 1372 1373 1403 1404 1406 1407 1408 1409 1410 1413 1414 1416 1430
1433 1476
## [196] 1478 1517 1523 1571 1586 1587 1589 2301 2357 2358 2359 2452 2454
2465 2466
## [211] 2467 2468 2479 2574 2616 2636 2699 2726 2772 2790 2828 2838 2842
2843 2880
## [226] 2883 2885 2893 2894 2920 2921 2933 2986 2987 2992 2994 3012 3026
3064 3190
## [241] 3203 3341 3407 3409 3414 3419 3422 3448 3462 3552 3595 3597 3598
3606 3657
## [256] 4003 4020 4041 4194 4234 4237 4252 4268 4280 4286 4348 4350 4372
4393 4396
## [271] 4414 4417 4472 4473 4474 4483 4489 4493 4517 4522 4526 4530 4531
4545 4559
## [286] 4584 4590 4598 4607 4608 4656 4657 4679 4683 4686 4722 4750 4752
4806 4807
## [301] 4825 4844 4884 4891 4901 4903 4967 4970 4973 5022 5036 5058 5068
5076 5082
## [316] 5083 5084 5099 5103 5107 5114 5116 5117 5119 5129 5139 5140 5224
5241 5242
## [331] 5255 5259 5264 5265 5271 5272 5273 5276 5310 5328 5335 5336 5354
5364 5373
## [346] 5385 5415 5429 5443 5451 5458 5501 5504 5507 5510 5515 5516 5518
5519 5522
## [361] 5531 5577 5598 5599 5600 5612 5665 5729 5749 5767 5795 5839 5903
5912 5950
## [376] 6001 6006 6009 6031 6046 6062 6080 6088 6089 6091 6103 6110 6145
6152 6160
## [391] 6182 6196 6217 6245 6258 6281 6296 6337 6356 6357 6386 6387 6389
```

```
6392 6402
## [406] 6415 6418 6437 6463 6465 6467 6486 6487

length(out_lev)

## [1] 413

# Checking for outliers using Cook's distances

cook<-cooks.distance(mlrmodel, type='rstandard')
which(cook>=1)

## named integer(0)
```

Data analysis related to question 3

```
## Running a chi-squared test on the acidity and colour variables

X2_data<-matrix(c(6159,338,1599,4898),nrow=2,byrow=TRUE)
colnames(X2_data) <- c("strongly acidic","weakly acidic")
rownames(X2_data) <- c("red","white")
chisq.test(X2_data,correct=FALSE)

##
##  Pearson's Chi-squared test
##
## data:  X2_data
## X-squared = 6651.6, df = 1, p-value < 2.2e-16
```