Tipologia i cicle de vida de les dades: Pràctica 2

Autors: Jonathan Mir Fernández-Aramburu i Dario Cabrera Gurillo

Maig 2022

Contents

---

1 Descripció del dataset

En aquest treball realitzarem un estudi sobre el dataset Red Wine Quality, el qual tenim disponible a la plataforma de kaggle i correspon al conjunt de dades originari del repositori *UCI machine learning repository.*

El dataset conté informació de diverses variants del vi portugués "Vinho Verde", incloent variables quantitatives com medicions Fisico-químiques i sensorials (qualitat del vi).Tanmateix per raons de privacitat i secret comercial s'exclouen dades comercials com la marca, el preu o el tipus de raïm emprat en l'elaboració dels vins.

Aquest dataset pot ser emprat per determinar quins factors fisicoquímics defineixen un bon vi, responent a les següents preguntes:

- Hi ha una combinació específica en les propietats Fisico-químiques que facin un vi de la millor qualitat?
- És un factor o hi ha diversos?
- Com és relacionen entre sí?

- Quines són les seves distribucions estadístiques?

El dataset conté 12 variables, on les 11 primeres poden considerar-se els inputs (factors Fisico-químics) i la última l'output (valoració de la qualitat del vi basada en una experiència sensorial) dins del rang $[1, 10]$.

Carreguem a continuació els paquets necessaris en R per tal de fer les anàlisi corresponents:

```
if (!require('dplyr')) install.packages('dplyr'); library('dplyr')
if (!require('nortest')) install.packages('nortest'); library('nortest')
if (!require('corrplot')) install.packages('corrplot'); library('corrplot')
if (!require('doBy')) install.packages('doBy'); library('doBy')
if (!require('caret')) install.packages('caret'); library('caret')
if (!require('tidyr')) install.packages('tydir'); library('tidyr')
if (!require('DescTools')) install.packages('DescTools'); library('DescTools')
if (!require('pROC')) install.packages('pROC'); library('pROC')
if (!require('rminer')) install.packages('rminer'); library('rminer')
if (!require('C50')) install.packages('C50'); library('C50')
```

---

2 Integració i selecció de dades a analitzar.

Donat que resulta fonamental conèixer el domini de les dades que pretenem analitzar i modelitzar passem a descriure-les a continuació:

1. **Fixed acitidy**: Quantitat d'àcids implicats al vi. La majoria dels àcids en el vi són fixos, és a dir, no s'evaporen fàcilment.
2. **volatile acidity**: Quantitat d'àcid acètic al vi. En altes quantitats, pot provocar un gust desagradable.
3. **citric acid**: Quantitat d'àcid cítric. Concentrat en petites quantitats, pot aportar frescor i sabor als vins.
4. **residual sugar:** La quantitat de sucre remanent un cop finalitzada la fermentació. Resulta estrany trobar vins amb menys d'un gram per litre. Vins amb més de 45 grams/litre són considerats dolços.
5. **Chlorides:** Quantitat de sal en el vi.
6. **Free sulfur dioxide:** SO2 en forma lliure existent en equilibri amb el SO2 mol · lecular (dissolt com un gas). És un element que impedeix el creixement bacterià i l'oxidació del vi.
7. **Total sulfur dioxide:** Quantitat total SO2. Encara que és necesari per a evitar la oxidació, en concentracions superiors als 50 ppm desbarata el gust i l'olor del vi.
8. **density:** Densitat del líquid en relació a la quantitat d'alcohol i sucre. Els vins solen ser un 8% més densos que l'aigua.
9. **pH:** Valor numèric que ens diu el grau d'acidesa o alcalinitat del vi. Descriu quan àcid o bàsic és un vi d'una escala des de 0 (molt àcid) a 14 (molt bàsic). La majoria de vins es situen entre 3 i 4.
10. **sulphates:** Quantitat d'additiu que actua com a antibacterià i antioxidant.
11. **Alcohol:** Percentatge d'alcohol present al vi.
12. **quality:** Variable output qualitativa basada en dades sensorials, en una escala de $[1, 10]$.

Per l'anàlisi que volem realitzar considerem que podem mantenir les 12 variables, per tant, no farem cap subselecció.

A continuació carreguem les dades a partir del csv descarregat a Kaggle:

```
library(readr)
B_vi <- read.csv("winequality-red.csv", sep= ",", header= TRUE, dec=".")
```

3 Neteja de les dades

3.1 Exploració del conjunt de dades

L'exploració inicial de les dades resulta fonamental per tenir una noció del domini de cada variable. Efectuarem a continuació una exploració preliminar del conjunt de dades, imprimint la capçalera amb un conjunt d'observacions:

```
str(B_vi)
```

```
## '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             : int  5 5 5 6 5 5 5 7 7 5 ...
```

Veiem que tenim un total de 1599 registres amb 12 variables.

Podem verificar a més les distribucions de les nostres variables tenint en compte estadístics com el mínim, els quartils (i mitjana) i el màxim, junt amb la mitja:

```
summary(B_vi)
```

```
##  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    density
##  Min.   :0.01200   Min.   : 1.00       Min.   :  6.00       Min.   :0.9901
##  1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00       1st Qu.:0.9956
##  Median :0.07900   Median :14.00       Median : 38.00       Median :0.9968
##  Mean   :0.08747   Mean   :15.87       Mean   : 46.47       Mean   :0.9967
```

```
##  3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00       3rd Qu.:0.9978
##  Max.   :0.61100   Max.   :72.00       Max.   :289.00       Max.   :1.0037
##        pH          sulphates         alcohol         quality
##  Min.   :2.740   Min.   :0.3300   Min.   : 8.40   Min.   :3.000
##  1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.310   Median :0.6200   Median :10.20   Median :6.000
##  Mean   :3.311   Mean   :0.6581   Mean   :10.42   Mean   :5.636
##  3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :8.000
```

Veiem que hi ha un gran diferència entre els valors de les variables **residual sugar, free sulfur dioxide i total sulfur dioxide**. També podem observar que la **mitjana de qualitat del vi és del 5.636 i la mitjana d' alcohol contingut en el vi és de 10.20**.

Com apunt addicional, veiem que en aquesta base de dades no hi ha distinció de si el vi es blanc o roig. Per tant, no podem separar les dues tipologies sinó que les treballarem conjuntament segons la qualitat del vi (`quality`)

3.2 Anàlisi d'elements buits i zeros

Comptarems a continuació els valors nuls per a cada columna:

```
colSums(is.na(B_vi))
```

```
##        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
```

Com veiem, en el nostre cas, no hi ha valors nuls presents en el dataset, probablement perquè ja s'ha passat per un pre-processament de dades abans de pujar-se a Kaggle. Tampoc eliminem cap valor en 0, ja que són valors que en el seu context tenen un sentit (per exemple, un zero a alcohol significa que el vi no conté alcohol).

3.3 Anàlisi de valors extrems

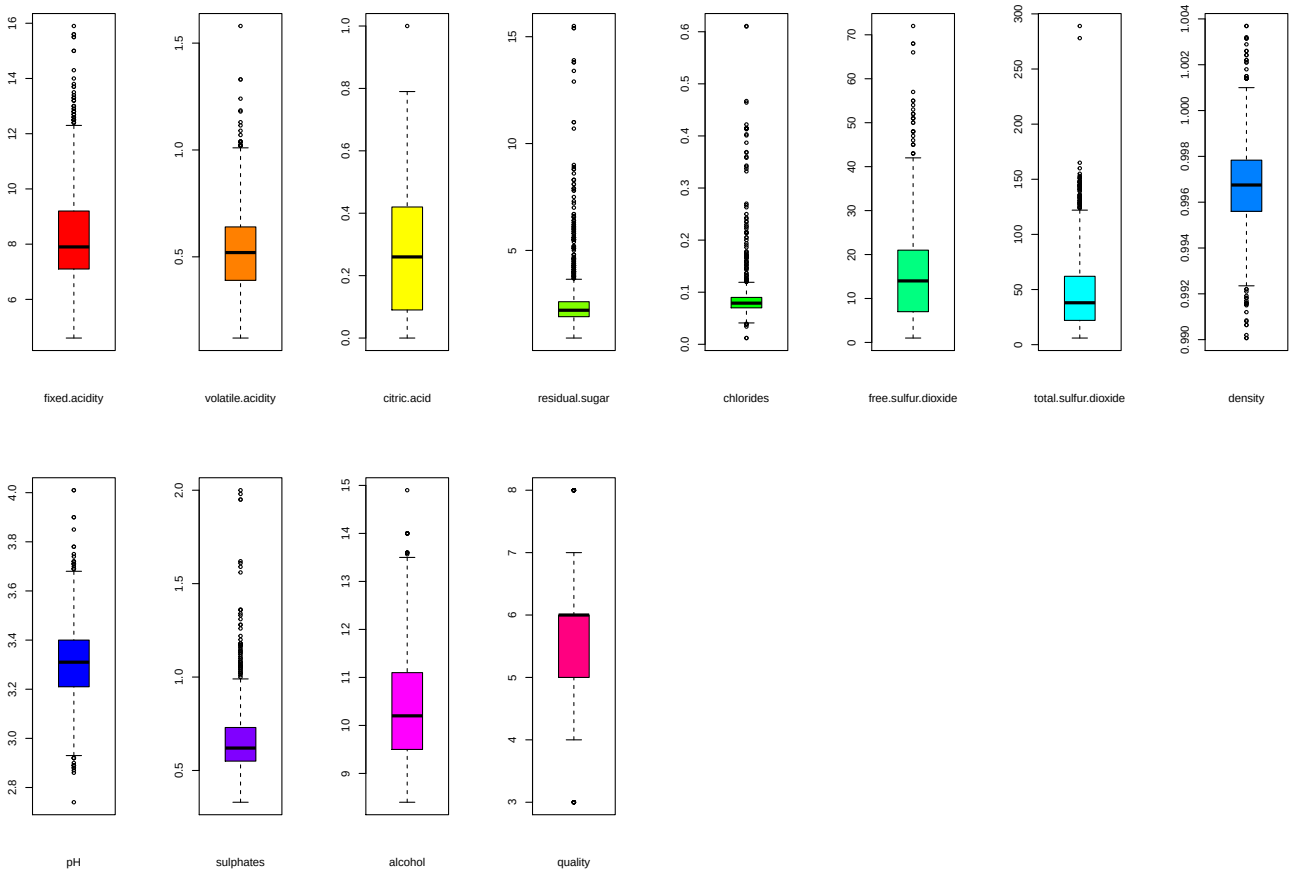
Ara analitzarem els valors extrems, és a dir, *outliers*. Per a visualitzar-los emprarem les gràfiques Boxplot. Les dades considerades *outliers* són aquelles que surten dels "bigots", és a dir, aquelles fora del rang

$$[Q_1 - 1.5 * IR, Q_3 + 1.5 * IRC]$$

, on IRC és el rang interquartilic, o el que és el mateix, $IRC = Q_3 - Q_1$, i Q_i és el percentil i-essim.

```
atributs <- names(B_vi)
p <- rainbow(12) #Colorets
k <- 1 # Per a reduir les lines de codi
for(i in 1:3){
   layout(matrix(c(1:4), nrow=1, byrow=FALSE)) #Matriu de grafiques 1x4
```

```
    for (j in k:(i*4)){
      boxplot(B_vi[,j], xlab=atributs[j], col=p[j]) #Boxplots
      }
  k <- 4*i+1
  }
```



Ara crearem un altre conjunt eliminant els valors extrems que veiem en el diagrama de caixa i bigots. Aquest l'emprarem per al test de saphiro per tal de veure si segueix una distribució normal o, si realitzant alguna transformació, sense outliers, segueixen una distribució normal o gaussiana.

```
# Llegim el document i el guardem en una altra variable
B_vi2 <- read.csv("winequality-red.csv", sep= ",", header= TRUE, dec=".")

# Convertim en valors nuls aquells que estan fora del rang interquartílic
for (i in 1:11){
  for (j in 1:1599){
    Hor <- B_vi2[,i]
    a <- quantile(Hor, 0.25, na.rm=TRUE)
    b <- quantile(Hor, 0.75, na.rm=TRUE)
    iqr <- (b-a)

    if (B_vi2[,i][j] <= (a-1.5*iqr)){
      B_vi2[,i][j] <- NA
```

```
    }
    else {if (B_vi2[,i][j] > (b+1.5*iqr)){
      B_vi2[,i][j] <- NA
    }
    }
  }
}
#Imprimim registres imputats com a valors nuls
print(colSums(is.na(B_vi2)))
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##                   57                   21                    1
##       residual.sugar            chlorides  free.sulfur.dioxide
##                  165                  133                   30
## total.sulfur.dioxide              density                   pH
##                   70                   45                   35
##            sulphates              alcohol              quality
##                   66                   13                    0
```

```
# Eliminem aquelles files que tenen valors nuls
B_vi2 <- drop_na(B_vi2)
```

En la captura anterior podem observar la quantitat de registres que hem eliminat per cada columna. Aquest nou conjunt de dades servirà posteriorment en la fase d'estudi de la normalitat de dades, per tal de veure si els valors extrems alteren la distribució general de cada característica. Alternativament, podríem aplicar altres mètodes per imputar valors sobre els extrems en lloc de simplement eliminar el registre, tals com imputació per un estadístic com la mitjana, la mitja o un percentil determinat.

De moment, conservem els dos datasets (amb i sense outliers) per a poder comparar l'efecte de la seva inclusió en la normalitat de les dades.

4 Anàlisi de les dades

4.1 Selecció dels grups de dades

Com s'ha mostrat abans, el conjunt original conté 12 variables, que conceptualment podem diferenciar entre:

- Característiques objetives i mesurables del vi (conté 11 diferents variables)
- Qualitat, que és una característica definida a partir de percepcions sensorials

Per tant, agruparem les dades en funció d'inputs i outout o variable objectiu. Sobre el primer grup realitzarem una anàlisi univariant i un anàlisi per correlacions. Posteriorment, entrenarem models predictius que relacionin les diferents característiques amb l'output com a variable explicada.

4.2 Estudi de la normalitat de les dades

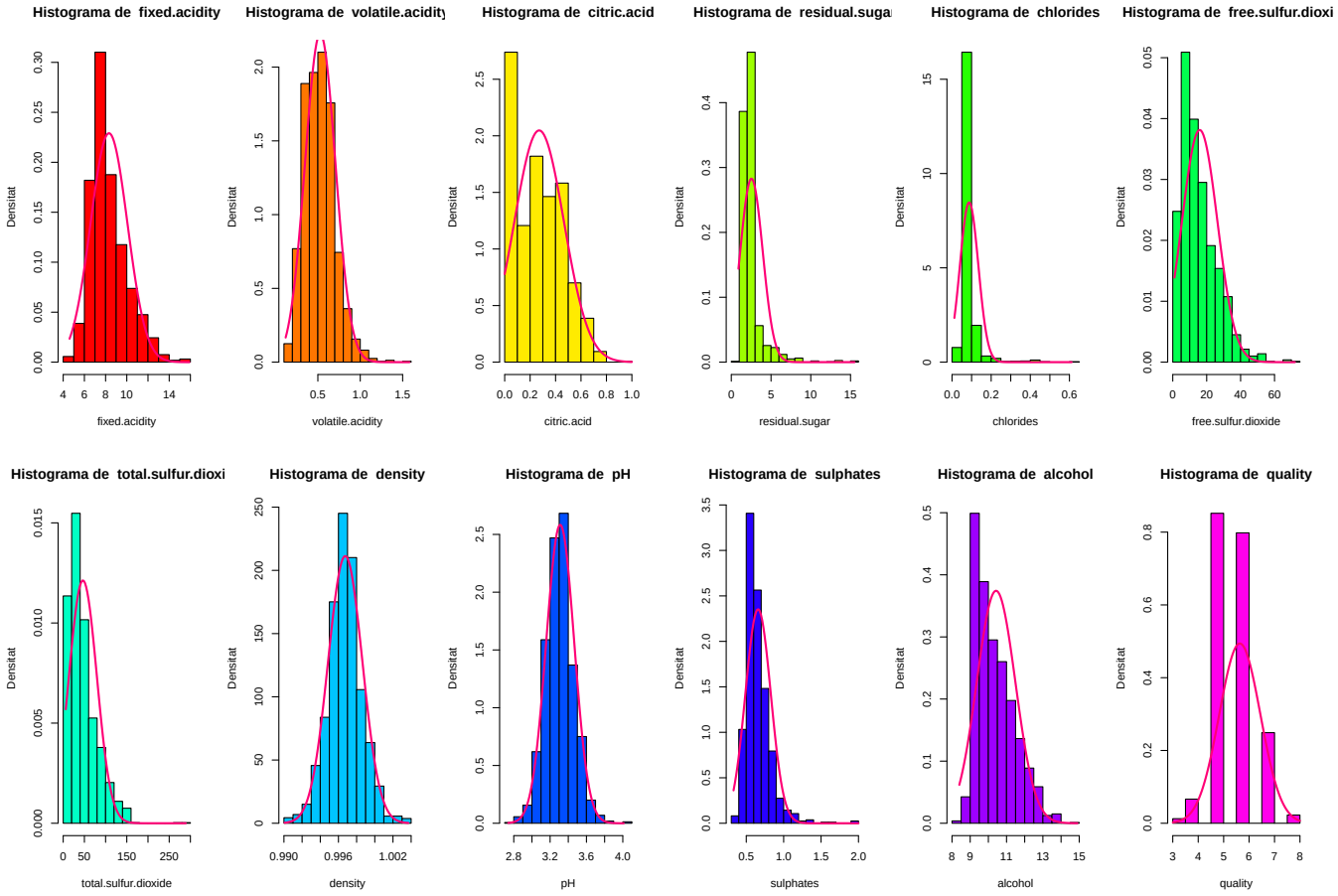
4.2.1 Visualització del conjunt de dades

Una assumpció molt important que determina l'anàlisi de les dades és si les variables segueixen una distribució normal, donat que en funció de la resposta, es pot aplicar un seguit de metodologies o un altre.

Primer visualitzarem, mitjançant histogrames, com es comporten les nostres variables en relació a una distribució teòrica gaussiana:

```
atributs <- names(B_vi)
p <- rainbow(13) #Colorets
k <- 1 # Per a reduir les lines de codi
for(i in 1:4){
  layout(matrix(c(1:3), nrow=1, byrow=FALSE)) #Matriu de grafiques 1x4

  for (j in k:(i*3)){
    hist(B_vi[,j],prob=TRUE, xlab=atributs[j], ylab="Densitat", col=p[j],
         main=paste("Histograma de ",atributs[j])) # Histograma per densitats
    curve(dnorm(x,mean=mean(B_vi[,j]),sd=sd(B_vi[,j])), from=min(B_vi[,j]),
      to=max(B_vi[,j]), add=TRUE, col=p[13], lwd=2) #Curva normal
  }
  k <- 3*i+1
}
```

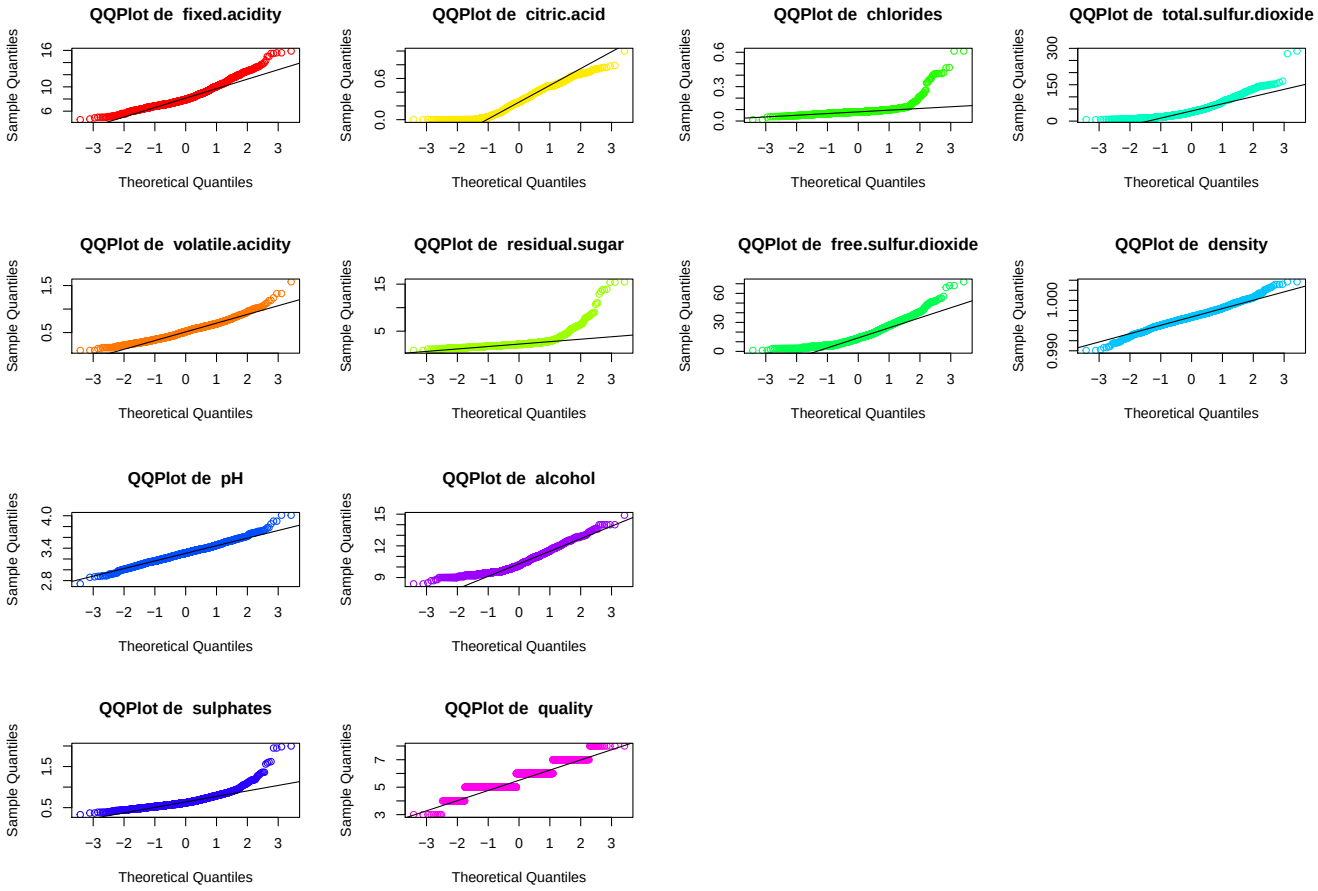


Com podem apreciar en els histogrames, sembla ser que les nostres variables estan desplaçades a l'esquerra. Una transformació convenient seria realitzar la transformació logaritmica o la inversa. Mes endavant, veurem si aquesta tranformació és suficient per a que les variables segueixin una normal emprant el test de Saphiro.

De moment, acabem de visualitzar la comparació amb la normal fent les QQ-plots:

```
k <- 1 # Per a reduir les lines de codi
for(i in 1:3){
   layout(matrix(c(1:4), nrow=2, byrow=FALSE)) #Matriu de grafiques 1x4

    for (j in k:(i*4)){
      qqnorm(B_vi[,j], main=paste("QQPlot de ",atributs[j]), col=p[j])
      qqline(B_vi[,j]) #Boxplots
      }
  k <- 4*i+1
  }
```



Analitzant els diferents **Q-Q Plots**, concloem que no semblen molt ajustats per a la normalitat: les distribucions més semblants a la gaussiana són per les variables de la densitat, el PH i el alcohol.

4.2.2 Tests de bondat de l'ajust

A continuació aplicarem el tests de Shapiro i Kolmogorov, per a comprovar si les dades efectivament segueixen una distribució normal, deixant fora la variable de qualificació, ja que serà el nostre target a analitzar.

Apliquem el següent codi per calcular el p-valor del test de shapiro sobre cada variable:

```
for (i in 1:11){
  p_val <- shapiro.test(B_vi[,i])
```

```
  print(paste("El p-valor del shapiro test de", atributs[i],
              "és:", p_val$p.value))
}
```

```
## [1] "El p-valor del shapiro test de fixed.acidity és: 1.52501179295091e-24"
## [1] "El p-valor del shapiro test de volatile.acidity és: 2.69293489456032e-16"
## [1] "El p-valor del shapiro test de citric.acid és: 1.02193162131975e-21"
## [1] "El p-valor del shapiro test de residual.sugar és: 1.02016171149076e-52"
## [1] "El p-valor del shapiro test de chlorides és: 1.17905575371677e-55"
## [1] "El p-valor del shapiro test de free.sulfur.dioxide és: 7.69459692029225e-31"
## [1] "El p-valor del shapiro test de total.sulfur.dioxide és: 3.57345139578549e-34"
## [1] "El p-valor del shapiro test de density és: 1.93605282884883e-08"
## [1] "El p-valor del shapiro test de pH és: 1.71223728301906e-06"
## [1] "El p-valor del shapiro test de sulphates és: 5.82314039765996e-38"
## [1] "El p-valor del shapiro test de alcohol és: 6.64405672007326e-27"
```

Els resultats anteriors mostren que les distribucions de les variables no segueixen cap normal ja que es rebutja la hipòtesi nul · la en tots els casos.

Ara aplicarem la transformació de BoxCox per tal de poder verificar si després de la transformació les dades segueixen una distribució normal sobre el conjunt de dades on s'han eliminat els outliers:

```
for (i in 1:11){
  p_val <- shapiro.test(BoxCox(B_vi2[,j], lambda = BoxCoxLambda(B_vi2[,j])))
  print(paste("El p-valor del saphito test de", atributs[i],
              "convertida és:", p_val$p.value))
}
```

```
## [1] "El p-valor del saphito test de fixed.acidity convertida és: 6.06563410461918e-32"
## [1] "El p-valor del saphito test de volatile.acidity convertida és: 6.06563410461918e-32"
## [1] "El p-valor del saphito test de citric.acid convertida és: 6.06563410461918e-32"
## [1] "El p-valor del saphito test de residual.sugar convertida és: 6.06563410461918e-32"
## [1] "El p-valor del saphito test de chlorides convertida és: 6.06563410461918e-32"
## [1] "El p-valor del saphito test de free.sulfur.dioxide convertida és: 6.06563410461918e-32"
## [1] "El p-valor del saphito test de total.sulfur.dioxide convertida és: 6.06563410461918e-32"
## [1] "El p-valor del saphito test de density convertida és: 6.06563410461918e-32"
## [1] "El p-valor del saphito test de pH convertida és: 6.06563410461918e-32"
## [1] "El p-valor del saphito test de sulphates convertida és: 6.06563410461918e-32"
## [1] "El p-valor del saphito test de alcohol convertida és: 6.06563410461918e-32"
```

Novament es rebutja la hipòtesi nul · la en tots els casos.

Apliquem també el test no paramètric de Kolmogorov-Smirnov (sobre el conjunt on no hem eliminat outliers):

```
for (i in 1:11){
  p_val <- ks.test(B_vi[,i], pnorm, mean(B_vi[,i]), sd(B_vi[,i]))
  print(paste("El p-valor del Kologomorov de", atributs[i],
              "és:", p_val$p.value))
}
```

```
## [1] "El p-valor del Kologomorov de fixed.acidity és: 0"
## [1] "El p-valor del Kologomorov de volatile.acidity és: 0.000141611830835164"
```

```
## [1] "El p-valor del Kologomorov de citric.acid és: 3.40695693878956e-10"
## [1] "El p-valor del Kologomorov de residual.sugar és: 0"
## [1] "El p-valor del Kologomorov de chlorides és: 0"
## [1] "El p-valor del Kologomorov de free.sulfur.dioxide és: 0"
## [1] "El p-valor del Kologomorov de total.sulfur.dioxide és: 0"
## [1] "El p-valor del Kologomorov de density és: 0.00327426744089643"
## [1] "El p-valor del Kologomorov de pH és: 0.0109069785673132"
## [1] "El p-valor del Kologomorov de sulphates és: 0"
## [1] "El p-valor del Kologomorov de alcohol és: 0"
```

Després dels resultats anteriors, arribem a la conclusió de que les variables, originals i transformades, no segueixen una normal. Per tant, no és convenient aplicar models que assumeixin normalitat en les dades, tals com la regressió lineal.

A més, hem aplicat diverses transformacions, tant la logarítmica com la inversa, per tal de corregir la curtosi i la assimetria de les distribucions en relació a la distribució gaussiana. Tanmateix, cap transformació genera dades normals.

Addicionalment, s'ha intentat realitzar una normalització per escala i per transformació de Box-Cox, però els resultats no són en cap cas satisfactoris. Alguns d'aquests resultats no els mostrem per estalvi d'espai en la documentació de la pràctica.

4.2.3 Test de homoscedasticitat

En definitiva, com les nostres dades no segueiexen una distribucio normal. Ara comprovarem l'homoscedasticitat emprant una prova de Finger-Killen per verificar si la variància és constant per la variable resposta:

```
for (i in 1:11){
  p_val <- fligner.test(B_vi[,12]~ B_vi[,i])
  print(paste("El test homoscedicitat de", atributs[i],
              "amb quality és:", p_val$p.value))
}
```

```
## [1] "El test homoscedicitat de fixed.acidity amb quality és: 0.981773123740242"
## [1] "El test homoscedicitat de volatile.acidity amb quality és: 0.362141747253417"
## [1] "El test homoscedicitat de citric.acid amb quality és: 0.236197626264641"
## [1] "El test homoscedicitat de residual.sugar amb quality és: 0.603307502614504"
## [1] "El test homoscedicitat de chlorides amb quality és: 0.564220853343594"
## [1] "El test homoscedicitat de free.sulfur.dioxide amb quality és: 0.695479131023802"
## [1] "El test homoscedicitat de total.sulfur.dioxide amb quality és: 0.0183192310199995"
## [1] "El test homoscedicitat de density amb quality és: 0.993806366292141"
## [1] "El test homoscedicitat de pH amb quality és: 0.523517118615673"
## [1] "El test homoscedicitat de sulphates amb quality és: 0.0413823548768257"
## [1] "El test homoscedicitat de alcohol amb quality és: 4.15745166691387e-07"
```

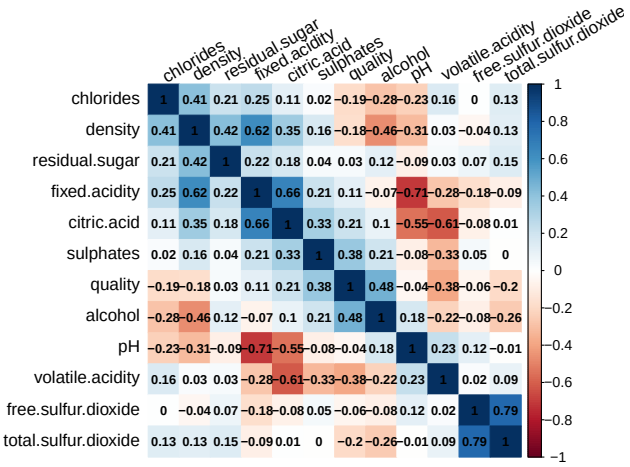
Com veiem, per a p-valors superiors al 0.05, tenim que sí presenten homoscedasticitat envers la variable resposta. Tanmateix, com les nostres dades no segueixen una distribució normal, realitzar un model de regressió lineal no és el mètode més eficaç. Ajustarem un model logístic per veure els resultats.

4.3 Aplicació de proves estadístiques

4.3.1 Correlacions

Abans de continuar, realitzarem un estudi sobre la correlació de les nostres variables. Ens centrarem especialment en la correlació que hi ha en les variables explicatives i la variable qualificació obtinguda. Com no segueixen una distribució normal emprarem el mètode *spearman*.

```
M = cor(B_vi, method="spearman")
corrplot(M,method="color",tl.col="black", tl.srt=30, order = "AOE",
number.cex=0.75,sig.level = 0.01, addCoef.col = "black")
```



Com podem observar, les variables que més correlació en sentit positiu (és a dir, directament relacionades) són: alcohol (0.48) i sulphates (0.38). I de manera negativa (relacionades de manera inversa) trobem volatile.acidity (-0.38).

Com és lògic pensar, els atributs que més inversament relacionats entre sí són el ph del vi amb fixed.acidityм, amb un valor de -0.71.

4.3.2 Regressio Logistica

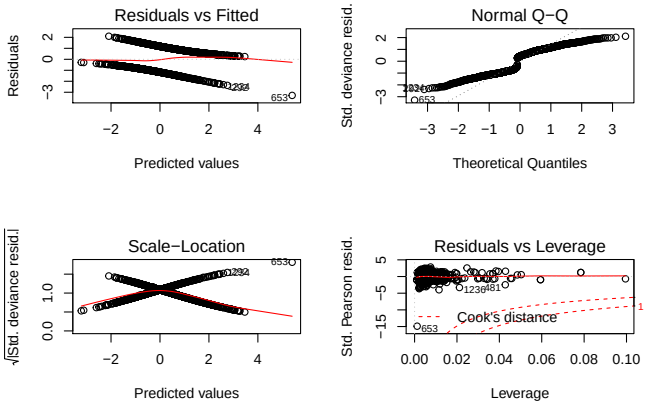
Ara realitzarem una regressió logística transformant l'atribut target, que actualment és numèric en una escala de l'1 al 10, a un atribut binari o dicotòmic, on aquelles puntuacions superiors o iguals a 6 seràn de la classe positiva (aprovades) i inferiors a aquest valor formaran part de la classe negativa (suspès). Aquesta partició es realitza perquè en el histograma anterior veiem que la majoria de les dades es troben al voltant de les valoracions 5 i 6. Les variables dependents que emprarem seran aquelles que aconsegueixen explicar certa variància total de la variable explicada "qualificacio".

El model desenvolupat serà el següent:

```
set.seed(200)
B_vi[,"quality_range"] <- cut(B_vi$quality, breaks=c(0,5.9,10),
                              labels=c("suspens", "aprovat"))
B_vi <- select(B_vi, -quality)
m1 <- glm(quality_range ~fixed.acidity+volatile.acidity+citric.acid+
            residual.sugar+chlorides+residual.sugar+free.sulfur.dioxide+
            density+pH, data=B_vi, family=binomial)
summary(m1)
```

```
##
## Call:
## glm(formula = quality_range ~ fixed.acidity + volatile.acidity +
##     citric.acid + residual.sugar + chlorides + residual.sugar +
##     free.sulfur.dioxide + density + pH, family = binomial, data = B_vi)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2877  -0.9900   0.4287   0.9634   2.1010
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          5.695e+02  5.338e+01  10.668  < 2e-16 ***
## fixed.acidity        7.309e-01  7.841e-02   9.321  < 2e-16 ***
## volatile.acidity    -3.899e+00  4.335e-01  -8.994  < 2e-16 ***
## citric.acid         -1.008e+00  4.907e-01  -2.054   0.0400 *
## residual.sugar       2.421e-01  4.715e-02   5.136 2.81e-07 ***
## chlorides            1.062e+00  1.406e+00   0.755   0.4501
## free.sulfur.dioxide -1.034e-02  5.658e-03  -1.827   0.0678 .
## density             -5.881e+02  5.468e+01 -10.756  < 2e-16 ***
## pH                   3.813e+00  5.762e-01   6.618 3.65e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2209  on 1598  degrees of freedom
## Residual deviance: 1862  on 1590  degrees of freedom
## AIC: 1880
##
## Number of Fisher Scoring iterations: 4
```

```
# Dibuixem
par(mfrow = c(2, 2))
plot(m1)
```



Donat que hem ajustat un model multivariant amb 11 dimensions, només analitzem els residus produits pel nostre model. En el gràfic de residius vs valors predits, observem que les dades es separen especialment en els extrem, però hi ha un bon ajust general al llarg de la recta. En el QQ plots, també podem concloure que

els residus segueixen una distribució normal, un resultat que és desitjable ja que ens assegura que no tenim patrons addicionals que distorsionin la distribuciío de residus.

D'altrabanda, el nostre model té una puntuació en termes d'AIC de 1880. Aquest valor no és directament interpretable, només podem afirmar que quant més petit millor. A banda, totes les variables tenen un valor menor al p-valor $\alpha = 0.05$, a excepció del chlorides i free.sulfur.dioxide. Resulta convenient eliminar aquests atributs si ajustem un altre model logístic.

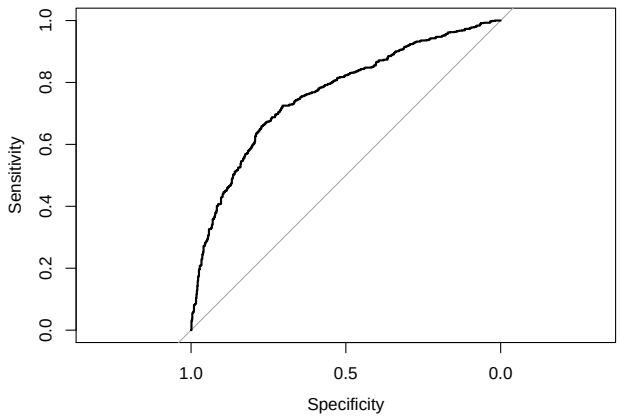
Per veure com funciona el nostre model, realitzarem una comprovació gràfica mitjançant la corba ROC. Aquesta corba realitza una gràfica i segons l'àrea compresa entre la corba i la recta $y = x$, ens indica el grau de capacitat predictiva del model. La puntuació oscil · la entre 0.5 i 1, on 1 és indicatiu d'un model perfectament predictiu i 0.5 és un model on la predicció és completament aleatòria. Mostrem a continuació la corba ROC resultant del model:

```
prob=predict(m1,B_vi,type="response")
r=roc(B_vi$quality_range, prob, data=B_vi)
```

```
## Setting levels: control = suspens, case = aprovat
```

```
## Setting direction: controls < cases
```

```
plot(r)
```




```
auc(r)
```

```
## Area under the curve: 0.7616
```

Obtenim un valor de 0.767 a la corba ROC, que és indicatiu d'un model correcte. Tot i així, el model pot millorar-se aplicant feature engineering o transformant variables i eliminant atributs que no aporten capacitat explicativa a la predicció.

4.3.3 Model Supervisat

Finalment, entrenarem un model supervisat del tipus arbre de decissió. Hem escollit l'algoritme C5.0. Aquest model ens realitza un diagrama d'arbre, on el resultat es decideix en funció de les regles definides per l'arbre en cada node.

```r
set.seed(200)

# Per a graficar l'arbre
gr = expand.grid(trials = c(1, 2),
model = c("tree"), winnow = c(TRUE, FALSE))

# Conjunt de entrenament i test
sep <- holdout(B_vi$quality_range, ratio=2/3, mode="stratified")
train <- B_vi[sep$tr,]
test <- B_vi[sep$ts,]

# A veure la distbucio
print(table(train$quality_range))
```

```
##
## suspens aprovat
##     496     570
```

```r
print(table(test$quality_range))
```

```
##
## suspens aprovat
##     248     285
```

```r
# Creacio del model
train_control<- trainControl(method="repeatedcv", number=2, repeats=5)
model <- train(quality_range~., data=train, trControl = train_control,
               method="C5.0", tuneGrid=gr)


#Apliquem el millor model posible
c5model = C5.0.default(x = select(train, -quality_range), y = train$quality_range,
trials = model$bestTune$trials, rules = model$bestTune$model == "rules",
control = C5.0Control(winnow = model$bestTune$winnow))

summary(c5model)
```

```
##
## Call:
## C5.0.default(x = select(train, -quality_range), y = train$quality_range,
##  trials = model$bestTune$trials, rules = model$bestTune$model ==
##  "rules", control = C5.0Control(winnow = model$bestTune$winnow))
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon May 30 22:02:15 2022
## -------------------------------
##
## Class specified by attribute 'outcome'
##
## Read 1066 cases (12 attributes) from undefined.data
##
## Decision tree:
```

```
##
## alcohol > 10.5:
## :...sulphates <= 0.58:
## :   :...alcohol > 11.4:
## :   :   :...volatile.acidity <= 0.55: aprovat (33/2)
## :   :   :   volatile.acidity > 0.55:
## :   :   :   :...citric.acid <= 0.05: aprovat (14/3)
## :   :   :       citric.acid > 0.05: suspens (7/1)
## :   :   alcohol <= 11.4:
## :   :   :...density > 0.99612:
## :   :       :...chlorides <= 0.114: suspens (25/2)
## :   :       :   chlorides > 0.114: aprovat (2)
## :   :       density <= 0.99612:
## :   :       :...chlorides <= 0.049: suspens (4)
## :   :           chlorides > 0.049:
## :   :           :...pH <= 3.46: aprovat (20/4)
## :   :               pH > 3.46:
## :   :               :...alcohol <= 10.8: aprovat (2)
## :   :                   alcohol > 10.8: suspens (8/1)
## :   sulphates > 0.58:
## :   :...alcohol > 11.5: aprovat (126/7)
## :       alcohol <= 11.5:
## :       :...total.sulfur.dioxide <= 61: aprovat (154/22)
## :           total.sulfur.dioxide > 61:
## :           :...pH <= 3.32: aprovat (8)
## :               pH > 3.32:
## :               :...alcohol <= 11.3: suspens (8)
## :                   alcohol > 11.3:
## :                   :...fixed.acidity <= 5.7: suspens (2)
## :                       fixed.acidity > 5.7: aprovat (3)
## alcohol <= 10.5:
## :...sulphates <= 0.58:
##     :...alcohol <= 9.7: suspens (181/26)
##     :   alcohol > 9.7:
##     :   :...sulphates > 0.54: aprovat (40/17)
##     :       sulphates <= 0.54:
##     :       :...volatile.acidity <= 0.48:
##     :           :...sulphates <= 0.45: suspens (2)
##     :           :   sulphates > 0.45:
##     :           :   :...citric.acid <= 0.23: aprovat (5)
##     :           :       citric.acid > 0.23:
##     :           :       :...citric.acid <= 0.3: suspens (3)
##     :           :           citric.acid > 0.3: aprovat (5/1)
##     :           volatile.acidity > 0.48:
##     :           :...alcohol > 10.03333: suspens (19)
##     :               alcohol <= 10.03333:
##     :               :...chlorides <= 0.069: aprovat (4/1)
##     :                   chlorides > 0.069:
##     :                   :...density <= 0.99651: suspens (24)
##     :                       density > 0.99651:
##     :                       :...volatile.acidity <= 0.67: aprovat (3)
##     :                           volatile.acidity > 0.67: suspens (9/1)
##     sulphates > 0.58:
##     :...total.sulfur.dioxide > 82:
```

```
##         :...pH <= 2.93: aprovat (3)
##         :   pH > 2.93: suspens (54/4)
##         total.sulfur.dioxide <= 82:
##         :...volatile.acidity > 0.545:
##             :...alcohol > 9.8: aprovat (58/24)
##             :   alcohol <= 9.8:
##             :   :...total.sulfur.dioxide > 76: aprovat (4)
##             :       total.sulfur.dioxide <= 76:
##             :       :...residual.sugar <= 2.3: suspens (51/9)
##             :           residual.sugar > 2.3:
##             :           :...density > 0.9997: suspens (4)
##             :               density <= 0.9997:
##             :               :...pH > 3.27: aprovat (10/1)
##             :                   pH <= 3.27:
##             :                   :...volatile.acidity <= 0.585: aprovat (2)
##             :                       volatile.acidity > 0.585: suspens (8/1)
##             volatile.acidity <= 0.545:
##             :...sulphates > 0.66:
##                 :...chlorides <= 0.097: aprovat (72/9)
##                 :   chlorides > 0.097:
##                 :   :...residual.sugar <= 1.65: suspens (4)
##                 :       residual.sugar > 1.65:
##                 :       :...fixed.acidity <= 8.3: aprovat (5)
##                 :           fixed.acidity > 8.3:
##                 :           :...fixed.acidity <= 10.6: suspens (6)
##                 :               fixed.acidity > 10.6: aprovat (6/1)
##                 sulphates <= 0.66:
##                 :...free.sulfur.dioxide <= 5: suspens (6)
##                     free.sulfur.dioxide > 5:
##                     :...free.sulfur.dioxide <= 6: aprovat (12)
##                         free.sulfur.dioxide > 6:
##                         :...alcohol > 10.2:
##                             :...total.sulfur.dioxide <= 52: suspens (8)
##                             :   total.sulfur.dioxide > 52:
##                             :   :...free.sulfur.dioxide <= 28: aprovat (3)
##                             :       free.sulfur.dioxide > 28: suspens (2)
##                             alcohol <= 10.2:
##                             :...chlorides <= 0.071: suspens (7/2)
##                                 chlorides > 0.071:
##                                 :...volatile.acidity > 0.48: aprovat (8)
##                                     volatile.acidity <= 0.48:
##                                     :...alcohol <= 9.25: suspens (3)
##                                         alcohol > 9.25:
##                                         :...residual.sugar <= 2.05: aprovat (8)
##                                             residual.sugar > 2.05:
##                                             :...alcohol <= 9.9: suspens (7/1)
##                                                 alcohol > 9.9: aprovat (4)
##
##
## Evaluation on training data (1066 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
```

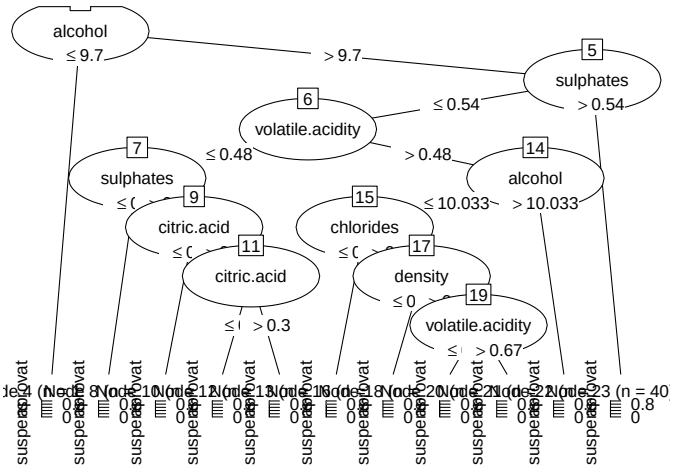
```
##
##      51  140(13.1%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     404    92    (a): class suspens
##      48   522    (b): class aprovat
##
##
##  Attribute usage:
##
##  100.00% sulphates
##  100.00% alcohol
##   49.72% total.sulfur.dioxide
##   39.96% volatile.acidity
##   21.67% chlorides
##   12.01% pH
##   11.35% density
##   10.79% residual.sugar
##    6.38% free.sulfur.dioxide
##    3.19% citric.acid
##    2.06% fixed.acidity
##
##
## Time: 0.0 secs
```

```
pred2 <- predict(c5model, newdata=test)
confusionMatrix(pred2, test$quality_range)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction suspens aprovat
##    suspens     170      58
##    aprovat      78     227
##
##                Accuracy : 0.7448
##                  95% CI : (0.7056, 0.7813)
##     No Information Rate : 0.5347
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.4845
##
##  Mcnemar's Test P-Value : 0.1033
##
##             Sensitivity : 0.6855
##             Specificity : 0.7965
##          Pos Pred Value : 0.7456
##          Neg Pred Value : 0.7443
##              Prevalence : 0.4653
##          Detection Rate : 0.3189
##    Detection Prevalence : 0.4278
##       Balanced Accuracy : 0.7410
```

```
##
##         'Positive' Class : suspens
##
```

```
plot(c5model, subtree= 3)
```



De manera visual, el nostre arbre es difícil de llegir, pero veiem que l'alcohol té un pes significatiu en el nostre model. En els resultats observem que el p-valor és molt petit, fet indicatiu de que el model és significatiu. A més, tal i com es mostra en el digrama, les variables alcohol i sulphates són les més significatives.

Analitzant la matriu de confusió també podem veure que hi ha 140 dades incorrectament classificades en la partició d'entrenament. En la partició de test, el model partim de 248 registres classificats com suspesos (puntuació inferior a 6) i 285 registres com a aprovats (puntuació igual o superior a 6), un total de 533 dades. En relació als falsos negatius tenim 58 registres i 78 falsos positius. Per tant, hi ha major tendència a que el model faci una predicció erronia cap a un fals positiu. L'exactitud total del model és del 0.745.

Per últim, exportem el csv amb el dataset modificat amb les prediccions definitives:

```
write.csv(B_vi, "Vins_categoritzats.csv")
```

5 Conclusions

Al principi plantejàvem si podríem explicar la qualitat d'un bon vi a partir de diferents propietats fisico-químiques mesurables.

Hem vist que les dades amb les que podem treballar estan força allunyades de les ideals per a modelitzar aquest problema amb una regressió lineal, que pren com a principal hipòtesi la normalitat de les dades. Hem aplicat diversos tractaments per tal d'ajustar-les a la distribució gaussiana, aplicant diverses transformacions (com BoxCox) o eliminant outliers, sense que els resultats milloréssin l'ajust.

Finalment, hem optat per generar una matriu de correlacions amb el mètode d'Spearman, que no requereix normalitat. Addicionalment, hem ajustat dos models, una regressió logística i un arbre de decissió, on hem inclós totes les variables explicatives possibles i com a variable objectiu hem definit la qualitat del vi.

En ambdos models hem obtingut una bona capacitat predictiva, cosa que demostra que les propietats fisico-químiques són explicatives de la qualitat. Addicionalment, hem vist que no totes les variables contribueixen a la predicció en la mateixa mesura, éssent l'alcohol, els sufactes i l'acidesa volàtil de les més rellevants.

Els resultats mostren que les metologies d'anàlisi emprades han permés resoldre el problema, tot i que el model pot ser millorat amb altres tècniques més avançades i amb més qualitat de dades.

Contribucions	Firma
Investigació Prèvia	JMF, DCG
Redacció de les respostes	JMF, DCG
Desenvolupament Codi	JMF, DCG