Tipologia i cicle de vida de les dades: Practica 2

Autors: Jonathan Mir Fernández-Aramburu i Dario Cabrera Gurillo

Maig 2022

Contents

---

1 Introducció

```
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')
```

1.1 Presentació

En aquest treball realitzarem un estudi sobre el dataset Red Wine Quality, el qual tenim a la plataforma de kaggle i correspon a una adaptació del dataset treobad en *UCI machine learning repository*, el qual agrupades les diferents caracteristiques dels vins blanc i rojos analitzats, amb la seua qualitat.

El repostiroy d'aquest treball es troba en https://github.com/jmirfern/data-lifecycle-pr2.

ELIMINAR ABANS D'ENTREGAR: PER A LA CONFIGURACIO DEL YAML, LES OPCIONS DEL PDF: https://bookdown.org/yihui/rmarkdown/pdf-document.html

2 Descripció del dataset

Aquest dataset no fa referencia a si el tipus de vi es blanc o roig, tenims les diferents caractersitiques del vi, i la seua puntuacio numerica de qualitat, dins del rang $[1, 10]$

1. **fixed acidity**: Quantitat d'àcids implicats al vi , en valor numeric.
2. **volatile acidity**: Quantitat d'àcid acètic al vi. on si tenim un nivell molt alt, aquest vi fara gust a vinagre.
3. **citric acid**: Aci ens trobem en la quantitat d'acid citric que te el vi, una variable numerica. Aquest valor ens diu la "frescor" dels vins.
4. **residual sugar:** La quantitat de sucre en el vi despres de la fermentació, almenys tots els vins han de tenir 1 gram/litre. Si hi ha mes de 45 grams/litres, es considera un vi dolç.
5. **chlorides:** valor numeric de la quantitat de sal al vi.
6. **free sulfur dioxide:** En diu el valor numeric de dioxid de sulfur lliure, element que impedeix el creixement bacteria i l'oxidació del vi.
7. **total sulfur dioxide:** Quantitat total de dioxid de sulfur. Encara que és necesari per a evitar la oxidació, un valor molt gran de concentracio desbaratara el gust i l'olor del vi.
8. **density:** Densitat del vi comparat amb la de l'aigua. Els vins solen ser un 8% mes densos que l'aigua.
9. **pH:** Valor numeric que ens diu el grau d'acidesa o alcanilitat del vi, per regla general, els vins solen ser algo acids (valors entre 3 i 4). Recordem que l'escala pH va del 0 (molt àcid) fins al 14 (molt bàsic).

10- **sulphates:** Valor numeric que ens diu la quantitat d'additiu del vi que contribueix a la creacio de diòxid de sofre.

11. **alcohol:** Valor numeric que ens diu el percentatge d'alcohol que te el nostre vi.
12. **quality:** Puntuaje rebut al vi, sense decimals, en l'escala de $[1, 10]$

2.1 Descripció de la PRA a realitzar

En aquesta activitat farem un analisis descriptius del dataset de vins aportat, en aquest mirarem en primer lloc com es distribuixen les dades, aixi com diagrames de caixa i bigots per a veure els valors extrems que podem trobar. Despres realitzarem un estudi de la normalitat de les variables, realitzant histogrames i QQ-plot, per acabar veient els diferents testos de normalitat que ens aporta R. A continuacio farem una comprovació de l'homoscedasticitat, per a veure si la variancia entre les dades amb la qualitat conserven la variancia. Per ultim realitzarem un diagrama de correlacions, un model de regresio logistica, i un model supervisat d'arbre.

3 Integracio i seleccio de dades a analitzar.

3.1 Elecció del conjunt de dades

Anem a realitzar un analisis previ

A continuació carreguem les dades:

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

3.2 Exploració del conjunt de dades

```
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.

Fixem-nos com estan distribuides les nostres variables, anem a veure el minim, els quartils i el maxim.

```
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 diferencia 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 apart, veiem que en aquesta base de dades no tenim distancio de si el vi es blanc o roig, aleshores no podem treballar les dades per separades, les tenim que treballar segons la qualitat del vi.

3.3 Analisis d'elements buis i 0

En les dades que acabem d'obtenir, veiem que no tenim valors nuls (almenys no identificats com a nuls). Per a comprovar-ho anem a veure les columnes.

```
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 tenim valors nuls, ja que aquest dataset pareix estar ben arreglat en la plataforma de Kaggle. Tampoc podem eliminar ningun valor en 0, ja que son valors que perfectament poden entendre's en el nostre dataset.

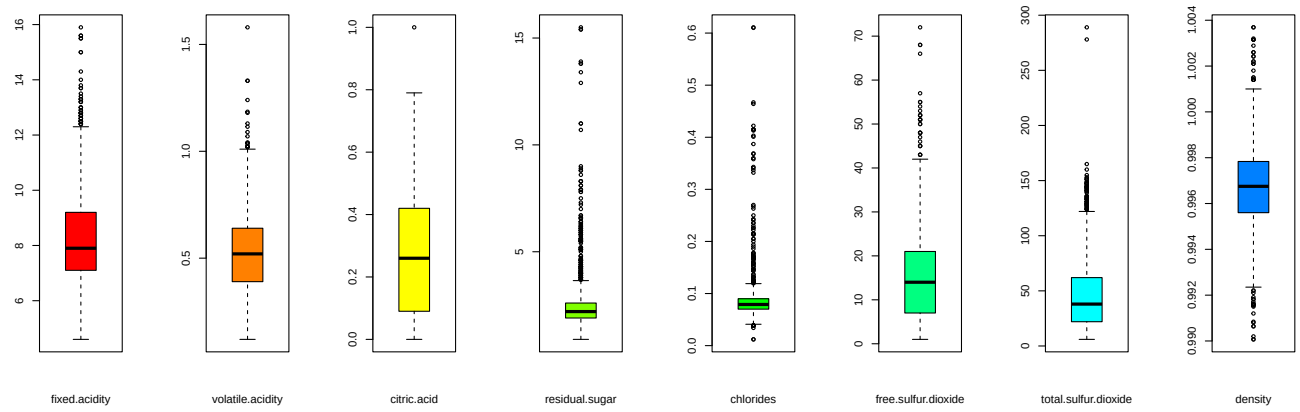
Ara veiem si tenim valors extrems, és a dir, *outliers*, per a veure-ho emprarem les grafiques Boxplot, i les dades considerades *outliers* son aquelles que ixen dels "bigots", és a dir, aquelles fora del rang

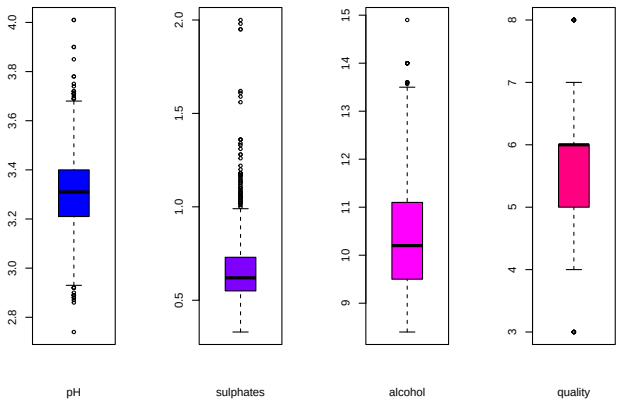
$$[Q_1 - 1.5 * IR, Q_3 + 1.5 * IRC]$$

, on IRC és el rang interquartilic, o lo 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
  }
```





Crearem un altre conjunt eliminant els valors extrems que veiem en el diagrama de caixa i bigots. Aquesta ho emprarem per al test de saphiro per veure si segueix una distribucio normal, o si realitzant alguna transformacio sense outliers segueixen una normal (tambe se li pot dir distribucio gaussiana).

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

# Per a eliminar valors nuls seguin el rang interquartilic
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
    }
    }
  }
}
#Veiem 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)
```

4 Analisis de les dades

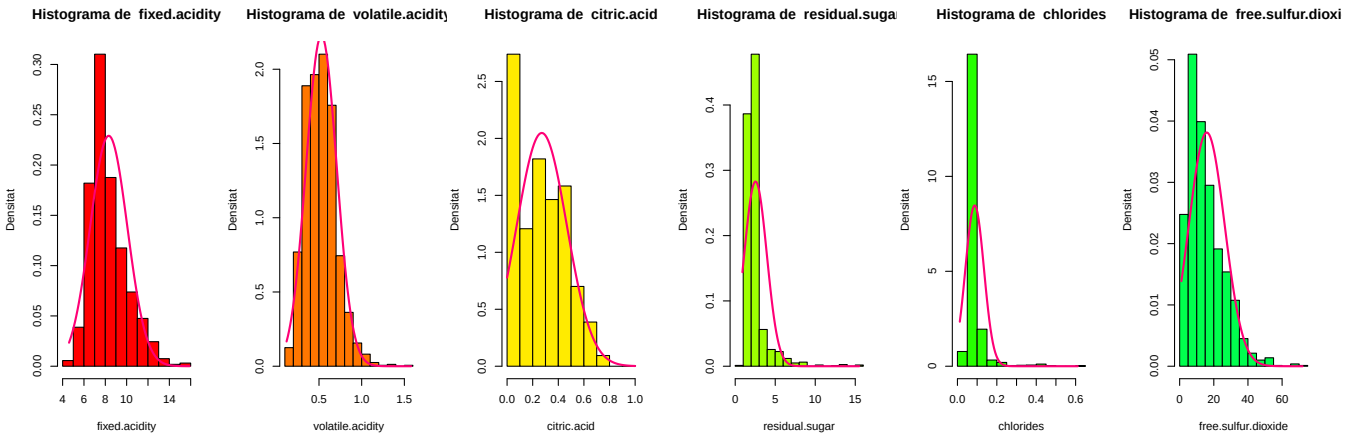
Recordem que volem veure si les nostres variables segueixen una normal, veurem com evoluciona la qualitat d'un vi segons els altres atributs que tenim. Segons si funcionen com una distribucio normal podem aplicar alguns procesos o uns altres, en cas de que funcionen, tindriem que els nostres models i resultats serien molt mes eficaços que en cas que no.

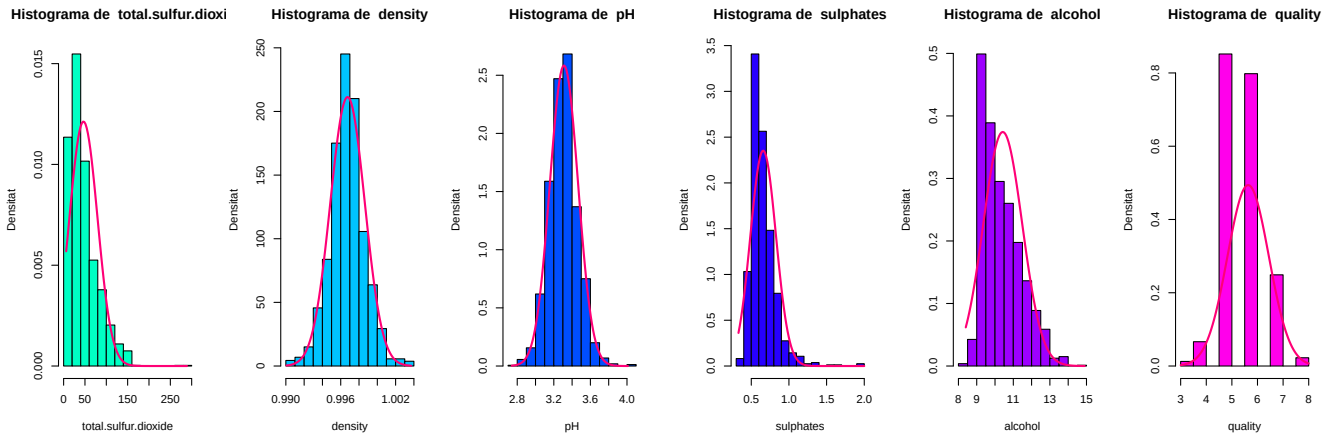
4.1 Estudi de la normalitat de les dades

Veiem de manera grtafica, mitjançant histogrames, com es comporten les nostres variables.

```
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
}
```



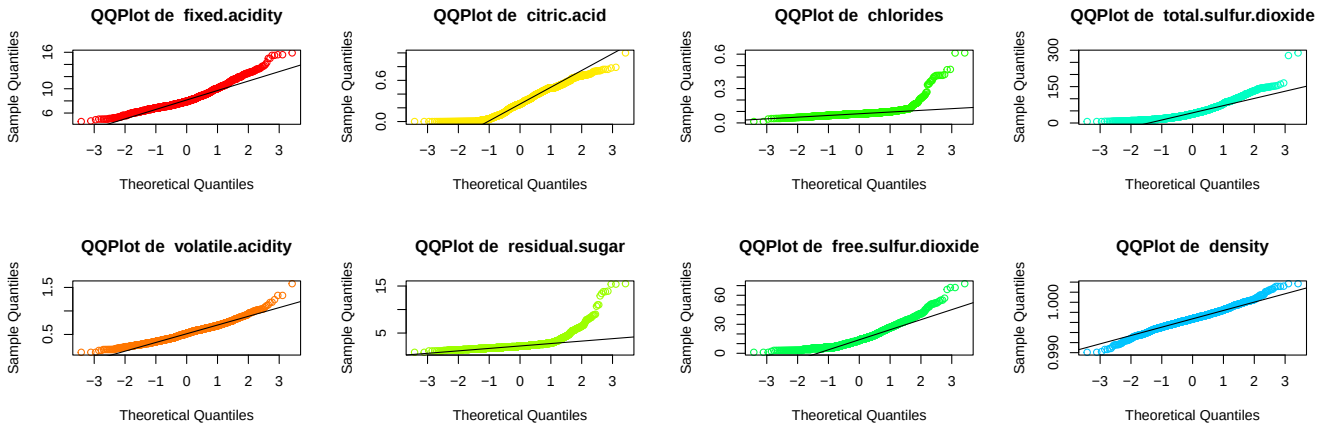


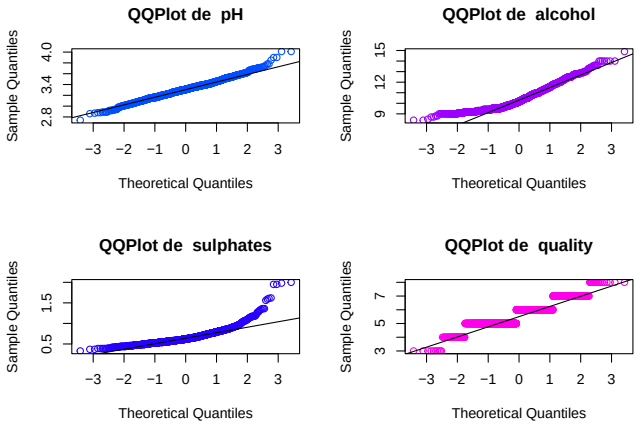
Com podem apreciar en els histogrames, pareix ser que les nostres variables estan desplaçades a l'esquerra, aleshores una transformacio convinient seria realitzar la tranformacio logaritmica o la inversa. Mes avanr veurem si aquesta tranformacio es suficient per a que les variables segueixen una normal, emprant el test de Saphiro.

Per ara acabem de visualitzar la comparacio en la normal fent grafiques QQ.

```
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
  }
```





Fixant-nos en els diferents **Q-Q Plots**, no pareixen molt bons per a la normalitat, les millors son la densitat, el PH i el alcohol. Despres realitzarem els diferents testos, per a vorer si de veritat segueixen una distribució normal, deixarem fora la variable de qualificació, ja que sera el nostre target a analitzar.

```
for (i in 1:11){
  p_val <- shapiro.test(B_vi[,i])
  print(paste("El p-valor del saphito test de", atributs[i],
              "es:", p_val$p.value))
}
```

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

Si no fem algo en els valors extrems, les nostres dades no segueixen una normal jeje.

```
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 es:", p_val$p.value))
}
```

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

```
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],
              "es:", p_val$p.value))
}
```

```
## [1] "El p-valor del Kologomorov de fixed.acidity es: 0"
## [1] "El p-valor del Kologomorov de volatile.acidity es: 0.000141611830835164"
## [1] "El p-valor del Kologomorov de citric.acid es: 3.40695693878956e-10"
## [1] "El p-valor del Kologomorov de residual.sugar es: 0"
## [1] "El p-valor del Kologomorov de chlorides es: 0"
## [1] "El p-valor del Kologomorov de free.sulfur.dioxide es: 0"
## [1] "El p-valor del Kologomorov de total.sulfur.dioxide es: 0"
## [1] "El p-valor del Kologomorov de density es: 0.00327426744089643"
## [1] "El p-valor del Kologomorov de pH es: 0.0109069785673132"
## [1] "El p-valor del Kologomorov de sulphates es: 0"
## [1] "El p-valor del Kologomorov de alcohol es: 0"
```

Ninguna segueix la normalitat, per aquest motiu no seria convenient aplicar una regresio lineal a les dades, ja que seria mes fiable tirar una moneda al aire que fiar-nos d'un model linial. Per aquest moti realitzarem una tranformacio de la nostra variable qualitat, i realitzar un model de regresio logistica.

S'ha intentat realitzar les tranformacions tant logaritmica com inversa, ja que com veiem en els histogrames, tenim la cua desenvolupada per la dreta del histograma, pero aixi i tot no segueix una distribucio normal. A banda, tambe s'ha intentat realitzar una normalitzacio per escala i una tranformacio Box-Cox. aquesta tampoc amb resultats correctes. Si es necesari i cap s'incoporaran.

Com les nostres dades no segueiexen una distribucio normal, anem a veure una comporvacio de l'homoscedasticitat emprant una prova de Finger-Killen

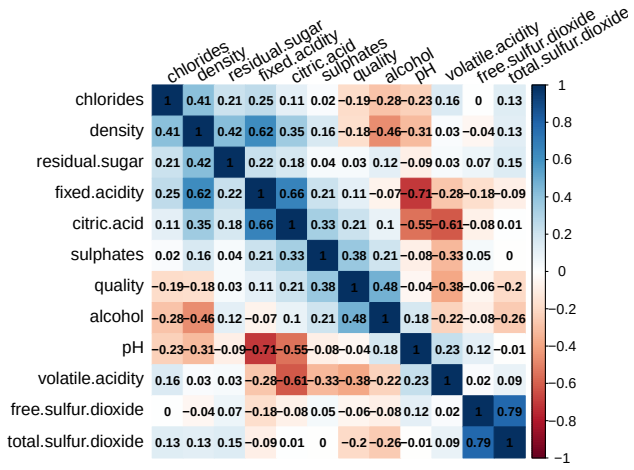
```
for (i in 1:11){
  p_val <- fligner.test(B_vi[,12]~ B_vi[,i])
  print(paste("El test homoscedicitat de", atributs[i],
              "amb quality es:", p_val$p.value))
}
```

```
## [1] "El test homoscedicitat de fixed.acidity amb quality es: 0.98177312374024"
## [1] "El test homoscedicitat de volatile.acidity amb quality es: 0.362141747253411"
## [1] "El test homoscedicitat de citric.acid amb quality es: 0.236197626264642"
## [1] "El test homoscedicitat de residual.sugar amb quality es: 0.603307502614511"
## [1] "El test homoscedicitat de chlorides amb quality es: 0.56422085334359"
## [1] "El test homoscedicitat de free.sulfur.dioxide amb quality es: 0.695479131023783"
## [1] "El test homoscedicitat de total.sulfur.dioxide amb quality es: 0.0183192310200004"
## [1] "El test homoscedicitat de density amb quality es: 0.993806366292141"
## [1] "El test homoscedicitat de pH amb quality es: 0.523517118615679"
## [1] "El test homoscedicitat de sulphates amb quality es: 0.0413823548768257"
## [1] "El test homoscedicitat de alcohol amb quality es: 4.15745166691327e-07"
```

Com veiem, per als p-valors superiors als 0.05, tenim que si son dades homoscebla, en canvi, per a les ods que son menors, tenen una relacio heterovlavla. Com les nostres dades no segueixen una distribucio normal, realitzar un model de regresio lineal no es el metode mes eficaç. Farem un model logistic a veure si aquest funciona.

Abans de continuar, realitzarem un estudi sobre la correlacio de les nostres variables. Ens concentrarem mes en la correlacio que hi han en les variables segons la qualificacio obtinguda, com no segueixen una distribucio normal emprarem el metode *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 mes correlacio tenen son, de manera positiva (és a dir, si la qualificacio es mes alta, estes creixen amb la qualificacio) son: alcohol (0.48) i luphatos (0.38). I de manera negativa (estan relacionades de manera inversa) és volatile.acidity (-0.38).

Com es logic pensar, els atributs que mes relacionades son, en manera negativa, es el ph del vi amb fixed.aciditym, amb un valor de -0.71.

Ara realitzarem una regresio logistica, per aquest motiu transforame l'atribut target, que actualment es troba de manera numerica, a un atribut logistic, on aquelles puntuacions superiors o igal a 6 considerem que estan aprovades i inferiors a aquest valor seran suspeses. Aquesta particio es realitza perque en el histograma anterior veiem que la majoria de les dades es troben al voltant del 5 i el 6. Les variables dependents que emprarem serna aquelles que consegueix guardar certa variancia amb la qualificacio, vist abans en el test de l'homoscedasticitat.

```
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
```

Com veiem, el nostre model te una puntuacio de AIC de 1880, aquest valor quant mes petit millor (revisar). A banda, totes les variables tenen un valor menor al valor $\alpha = 0.05$, a excepció del chlorides i free.sulfur.dioxide. Seria convenient eliminar aquests atributs si realitzem un altre model logisitic.

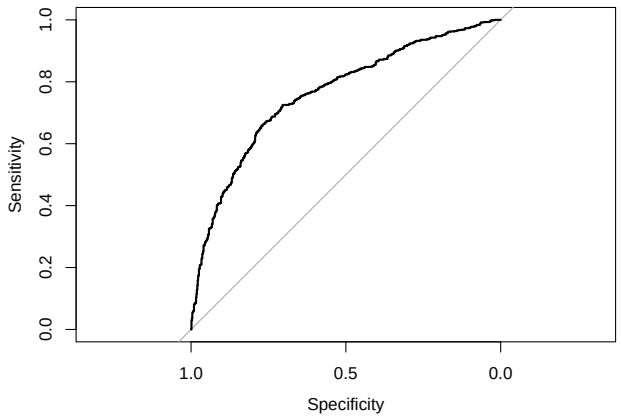
Per a veure com funciona el nostre model, realitzarem una comprovacio grafica mitjançant la corba ROC, aquesta corba realitza una grafica i segons el area que queda per sota la corva amb la recta $y = x$, en diu com va el nostre test. Una puntuacio de 0.5 vol dir que el nostre test no funciona correctament, i una puntuacio de 1 vol dir que es perfecte. A veure el resultat que obtenim. EXPLICAR MILLOR LA CORBA ROC.

```
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
```

un valor de 0.767, és un valor molt bo de prediccio, pero podria ser millorable, ja sigui realitzant transformacions de les nostres variables o eliminar aquells atributs que fan mal be la prediccio.

Per ultim, realitzarem un model supervisat, en aquest cas hem elegit el C5.0, vist en anterioritat en altres

assignatures. Aquest model ens realitza un diagrama d'arbre, paregut a un arbre de decisio, on segons el resultat de la variable a analitzar decidirem un cami o un altre, acabant en una fulla.

```
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
```

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

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

```
# 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 16 23:27:48 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
##
```

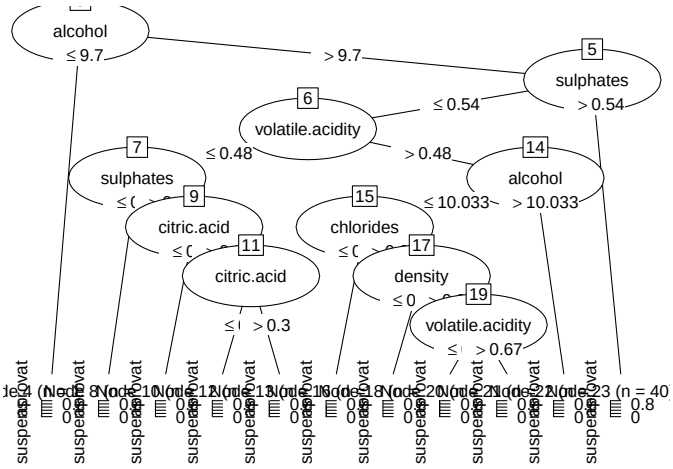
```
##      51  140(13.1%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     404    92    (a): class suspens
##      48   522    (b): class aprovat
##
##
##  Attribute usage:
##
##  100.00% sulphates
##  100.00% alcohol
##   49.72% total.sulfur.dioxide
##   39.96% volatile.acidity
##   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 dificil de llegir, pero veiem que el alcohol te un pes significatiu en el nostre model. amb les dades veiem que el nostre p-valor es molt petit, cosa que ens dona bona señal. Com hem dit abans, ara de manera textual, la variable de mes pes en el nostre model es la quantitat d'alcohol que te el vi, acompanyat despres dels sulphates. Tambe veiem que el model ha clasificat incorrectament 140 dades de les dades d'entrenament, que no son poques.

Per ultim comentar la matriu de confusio amb les dades de test, en aquest cas partim de 248 dades clasificades com suspenses i 285 dades com a aprovades, un total de 533 dades. Correctament han sigut clasificades 170 aprovades i 227 suspenses. Com a aprovades incorrectament tenim 58 dades i suspenses incorrectament 78 dades, ens fa pensar que el nostre model és mes facil que un vi siga bo (és a dir que aprove), a que siga roin. L'exactitud del nostre model es de 0.745, un resultat prou bo (pero millorable).

Per ultim extraiem el csv del nostre archiu modificat.

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

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