Pràctica 2

M2.951 – Tipologia i cicle de vida de les dades

Alumne: Miquel Ribó i Pal

Nota: S'ha modificat lleugerament l'estructura d'aquesta memòria respecte de la que s'indica a l'enunciat de la pràctica:

- L'apartat 3 de l'enunciat (Neteja de dades) s'ha transformat en l'apartat 3. Pre-processament de les dades per tal d'afegir-hi les tasques de creació de noves variables discretes derivades i la normalització de les dades numèriques.
- El subapartat 4.3 s'ha expandit en diversos subapartats, un per a cada prova estadística o manipulació de les dades realitzada. Com que la fase de pre-processament de les dades és iterativa, hi ha també un apartat de pre-processament de les dades, de reducció de la numerositat.
- L'apartat 5 de l'enunciat s'ha fos amb el 4 de la memòria (els resultats es van mostrant a continuació de les anàlisis fetes.
- L'apartat 6 de l'enunciat (Resolució del problema) s'ha transformat en l'apartat 5. Conclusions.

1. Descripció del data set

S'ha triat el data set *Red Wine Quality* (https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009) (P. Cortez, A. Cerdeira, F. Almeida, T. Matos i J. Reis, "Modeling wine preferences by data mining from physicochemical properties", *Decision Support Systems*, Elsevier, 47(4):547-553, 2009) perquè és un bon exemple de l'aplicació de tècniques de Machine learning a un problema que té transcendència industrial i econòmica: el de discernir el grau de qualitat (subjectiu) esperable d'un vi (en aquest cas variants fetes amb raïm negre de Vinho Verde portuguès) a partir de les seves característiques (objectives) fisico-químiques. El fet que les característiques físico-químiques estiguin codificades amb variables reals contínues, i que hi hagi una única variable categòrica ordenada (de 0 a 10), la de la qualitat subjectiva percebuda del vi, fa que es puguin aplicar tant tècniques de regressió lineal com de classificació per a mirar d'obtenir informació sobre els paràmetres físico-químics rellevants en la determinació de la seva qualitat subjectiva.

Les dades del data set corresponen a un conjunt de 1599 observacions d'11 paràmetres físico-químics per a 1599 vins de la varietat comentada, així com la valoració subjectiva en una escala d'1 a 10 per a cadascun dels vins. Les variables dels data set són les següents:

- **`fixed acidity`**: Nivell d'acidesa deguda a àcids no volàtils del vi, que són la majoria, llevat de l'àcid acètic (https://en.wikipedia.org/wiki/Acids_in_wine). Els àcids no volàtils presents en més quantitat al vi són el tartàric, el màlic i el làctic.
- **`volatile acidity`**: Nivell d'àcid acètic, que en quantitats elevades dóna un gust avinagrat al vi.
- **`citric acid`**: Nivell d'àcid cítric. Es troba en petites quantitats al raïm (de l'ordre d'1/20 dels nivells d'àcid tartàric). Sovint s'afegeix artificialment al vi per en petites dosis per a eliminar-ne el coure o el ferro (https://en.wikipedia.org/wiki/Acids_in_wine). Pot donar aroma i frescor als vins.
- **`residual sugar`**: Nivell de sucre que roman al vi un cop acabada la fermentació. Sol ser superior a 1 gr/l. Vins amb nivells superiors a 45gr/l es consideren dolços.
- **`chorides`**: Nivell de sal al vi.
- **`free sulful dioxide`**: Nivell de SO_2 en forma lliure. Prevé la contaminació microbiana i l'oxidació del vi.
- **`total sulfur dioxide`**: Nivell de SO_2 en forma lliure i lligada. En concentracions baixes no afecta el gust del vi, però a partir de concentracions de 50 ppm (per a la forma lliure) n'afecta el gust i l'aroma.
- **`densitat`**: Més o menys propera a la de l'aigua en funció dels nivells de sucre i alcohol.
- **`pH`**: pH del vi, en una escala de 0 (molt àcid) i 14 (molt bàsic). Usualment entre 3 i 4.
- **`sulphates`**: Additius del vi que poden contribuir als nivells de SO_2.
- **`alcohol`**: Percentatge d'alcohol del vi.
- **`quality`**: Aualitat (subjectiva) del vi, basada en les seves característiques sensorials.

A partir de les dades que se subministren al lloc web esmentat, és impossible escatir les unitats d'alguns dels paràmetres que conté.

2. Integració i selecció de les dades d'interès a analitzar

El data set està contingut en un fitxer en format csv (amb capçaleres), `winequality-red.csv`, que pot descarregar-se del lloc web citat més amunt.

Si es carrega el fitxer a un `tibble` d'R, se substitueixen els espais de les capçaleres per "_" i se'n fa una anàlisi preliminar amb `summary()`:

```
library(tidyverse)
library(gridExtra)
library(stats)
library(class)

# Carrerguem el fitxer de dades al pickle df, mantenint el tipus int
# per a les dades subjectives de quality
df <- read_csv(".\\winequality-red.csv",
               col_names = TRUE,
               col_types = "ddddddddddddi")

# Substituim els espais als noms de columna per "_"
names(df) <- gsub(" ", "_", names(df))

df %>% select(1:4)  %>% summary() %>% print()
df %>% select(5:8)  %>% summary() %>% print()
df %>% select(9:12) %>% summary() %>% print()
```

```
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
```

Com es pot constatar els valors corresponen, quan a nivells i unitats, al que s'ha comentat més amunt. Cal recalcar que els nivells subjectius de qualitat es troben tots, per als vins presents al data set, entre els nivells 3 i 8.

En principi totes les variables que conté el data set són d'interès per a l'anàlisi que es vol fer perquè, tal com es comenta al lloc web, les dades de menús valor estadístic com ara preu, nom o bodega ja han estat eliminats del data set.

3. Pre/processament de les dades

Una vegada carregades les dades, es procedirà al seu pre-processament (Han et al, Data Mining. Concepts and Techniques). Com que no hi ha un nombre excessiu de dades, en principi no caldrà aplicar tècniques de reducció de dimensionalitat o numerositat, però si tècniques de neteja i de transformació de les dades.

3.1. Addició de variables categòriques

Com que es pretén analitzar fins a quin punt les dades físico-químiques permeten predir la qualitat subjectiva del vi, caldran, per a realitzar tasques de classificació, variables categòriques que defineixin aquesta qualitat. Se n'han definit tres:

- **`quality_c1`**: versió categòrica dels valors numèrics enters de la variable quality
- **`quality_c2`**: discretització dels valors enters de `quality` en tres grups ordenats: "Dolenta" (3 i 4), "Regular" (5 i 6) i "Bona" (7 i 8)
- **`quality_c3`**: discretització dels valors enters de `quality` en dos grups ordenats: "No acceptable" (3 a 6) i "Acceptable" (7 i 8).

D'aquesta manera es tindran diversos criteris de classificació possibles per a investigar.

```
# Transformem les dades numèriques enteres de qualitat, "quality"", en
# tres tipus de factors, un amb els mateixos valors numèrics considerats
# com a factors, un altre que divideix la qualitat del vi en "Bona",
# "Regular" i "Dolenta", i un altre en "Adequada" i "No adequada"

levels1 = as.character(0:10) # No emprat
levels2 = as.character(3:8)
levels3 = c("Dolenta","Regular", "Bona")
levels4 = c("No adequada", "Adequada")

aux1 <- df$quality %>% as.character() %>%
         factor(.,levels = levels2, ordered = TRUE)

aux2 <- ifelse(df$quality <= 4, "Dolenta", ifelse(df$quality >= 7, "Bona", "Regular"))
%>%
          factor(.,levels = levels3, ordered = TRUE)

aux3 <- ifelse(df$quality <= 6, "No adequada", "Adequada")

# Incorporem aquestes noves variables categòriques al tibble df
# per al seu ús posterior
df <- df %>% mutate(quality_c1 = aux1, quality_c2 = aux2, quality_c3 = aux3)

# Imprimim els còmputs de valors
cat("quality_c1")
print(table(df$quality_c1))
cat("\nquality_c2")
print(table(df$quality_c2))
cat("\nquality_c3")
print(table(df$quality_c3))
```

```
quality_c1
  3   4   5   6   7   8
 10  53 681 638 199  18

quality_c2
Dolenta Regular    Bona
     63    1319     217

quality_c3
   Adequada No adequada
        217     1382
```

3.2. Tractament de nuls i zeros

Primerament es comprovarà si al data set hi ha zeros o elements buits:

```
# Comprovem si el dataset té valors NA
aux1 <- df %>% select(-(quality:quality_c3)) %>%
               lapply(function(x) { length(which(is.na(x))) }) %>%
               as_tibble()

# Comprovem si el dataset té zeros
aux2 <- df %>% select(-(quality:quality_c3)) %>%
        lapply(function(x) { length(which(x<1e-6)) }) %>%
        as_tibble()

# Combinem els resultats per a imprimir-los
# Passem a data.frame per a afegir noms de fila
aux <- bind_rows(aux1, aux2) %>% as.data.frame()
row.names(aux) <- c("NA", "Zeros")
aux %>% t() %>% print()
```

```
                     NA Zeros
fixed_acidity         0     0
volatile_acidity      0     0
citric_acid           0   132
residual_sugar        0     0
chlorides             0     0
free_sulfur_dioxide   0     0
total_sulfur_dioxide  0     0
density               0     0
pH                    0     0
sulphates             0     0
alcohol               0     0
```

Com es pot observar el data set és ja força net. No conté valors nuls (NA) i només conté zeros (força) per a la variable `citric_acid`. Tot i que, com s'ha comentat més amunt, els nivells de cítric_acid poden ser petits (al data set hi ha valors de 0.04 i similars), uns valors estrictament iguals a 0 poden indicar que es tracta d'un paràmetre arrodonit, o no mesurat (un fals NA). A la vista d'això, hi ha diverses opcions possibles:

- Es poden eliminar els registres/files afectats per aquest problema.
- Es poden substituir els zeros per un valor que estadísticament tingui sentit, com ara la mitjana dels altres valors. Aquesta opció disminueix la variància de les dades.
- Es pot mirar d'inferir el valor correcte a partir d'una regressió lineal (perquè es tracta d'un valor continu) entrenada a partir dels casos en què la variable no pren el valor zero.

Tot i ser l'opció més complicada, s'ha pres la darrera opció perquè preserva la mida del data set i, en principi, també més la variància de les dades (A.C. Rencher i W.F. Christensen, *Methods of Multivariate Analysis*, 3a ed. Wiley). S'ha creat una nova variable corregida per als nivells d'àcid cítric, `citric_acid_corr`, que s'ha afegit al data set a continuació de `cítric_acid`, que no s'ha esborrat:

```
# Predicció, emprant regressions lineals dels valors d'àcid cítric
# per a les mostres amb un valor idènticament igual a 0.

# Primerament generem una columna addcional, després de la de
# 'citric_acid', anomenada 'citric_acid_corr',  que contindrà
# els valors corregits de 'citric_acid'
df <- df %>% mutate(citric_acid_corr = citric_acid) %>%
  select(fixed_acidity:citric_acid, citric_acid_corr, everything())

# Generem un conjunt d'entrenament del model lineal amb les dades
# que tenen 'citric_acid' != 0, i un per a emprar en la predicció,
```

```
# per a les dades que tenen 'citric_acid' != 0.
# Emprem la funció filter de dplyr, que diferenciem de la de la
# llibreria base, per a augmentar la claredat
df_predict <- df %>% dplyr::filter(citric_acid < 1e-6)
df_train   <- df %>% dplyr::filter(citric_acid >= 1e-6)

# Generem una fórmula per a la regressió lineal de 'citric_acid'
# a partir d'una cadena de caràcters construïda a patir dels noms de les
# la resta de columnes del tibble df (llevat de les de quality, quality_c1
# i quality_c2 i citric_acid_corr)
f <- df %>% select(-citric_acid, -citric_acid_corr, -(quality:quality_c3)) %>%
  names() %>% paste(collapse = "+") %>% paste("citric_acid ~",.) %>%
  as.formula()

# Calculem els coeficients del model de regressió lineal emprant
# les dades d'entrenament, df_train, en què coneixem el valor exacte
# del paràmetre 'citric_acid'
model <- lm(f, data = df_train)

# Imprimim i dibuixem les dades rellevants de la fase d'entrenament
cat("Coeficients :\n")
print(model$coefficients)
cat("\nError absolut rms per mostra :\n",
    sqrt(sum(model$residuals**2)/nrow(df_train)), "\n")
cat("\nCorrelació entre predicció i valors d'entrenament :\n",
     cor(df_train$citric_acid, model$fitted.values), "\n")

ggplot() +
  geom_point(mapping = aes(x = df_train$citric_acid, y = model$fitted.values)) +
  xlab("citric_acid (real)") +
  ylab("citric_acid (predint)") +
  labs(title = "Valors reals vs. predits (dades d'entrenament)",
       substitle = NULL,
       tag = NULL)
```

```
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity       residual_sugar
chlorides  free_sulfur_dioxide
        -3.663733268          0.054570400         -0.424350425          0.004222231
0.758853520         -0.002025918
total_sulfur_dioxide              density                   pH            sulphates
alcohol
         0.001110236          3.426000925         -0.034135462          0.019525112
0.029056897

Error absolut rms per mostra :
 0.1085892

Correlació entre predicció i valors d'entrenament :
 0.809046
```

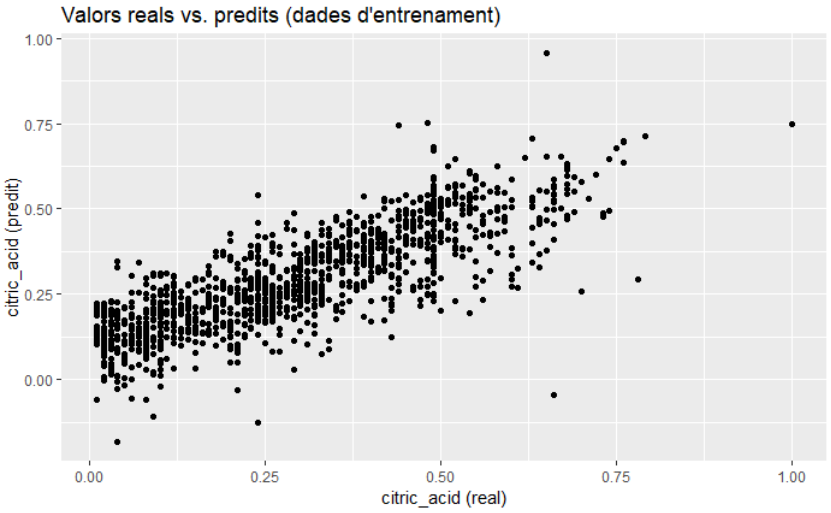

Fig. 1. Diagrama de dispersió per als valors reals i predits d'àcid cítric per al conjunt de dades d'entrenament.

Com es pot constatar la correlació entre el valor predit i real per a les dades d'entrenament és acceptable, amb un error rms de 0.108, que correspon a la faixa de dades de gruix aproximadament 0.2 de la gràfica de la Fig. 1. Si s'aplica el model ara a la correcció dels valors 0 de `citric_acid` (els valors diferents de zero no es modifiquen):

```
# Fem la predicció (correcció) dels valors de citric_acid per a les mostres
# amb citric_acid = 0, col·lapsant les prediccions menors que zero a zero
p <- predict(model, df_predict)
p0 <- ifelse(p>0, p, 0)

# Incorporem aquests nous valors al paràmetre 'citric_acid_corr' NOMÉS per
# a les dades que tenen 'citric_acid' == 0
df_predict <- df_predict %>% mutate(citric_acid_corr = p0)

# Recomposem el tibble original, df, tot i que amb un altre ordre de
# columnes, apilant les dades de df_train i # df_predict
df <- bind_rows(df_train, df_predict)

# Comparem les distribuicions estadístiques de 'citric_acid' i
# 'citric_acid_corr'
gg1 <- ggplot(data = df) +
  geom_histogram(mapping = aes(citric_acid, fill = quality_c2), bins = 20)

gg2 <- ggplot(data = df) +
  geom_histogram(mapping = aes(citric_acid_corr, fill = quality_c2), bins = 20)

grid.arrange(gg1, gg2, ncol = 2)
```

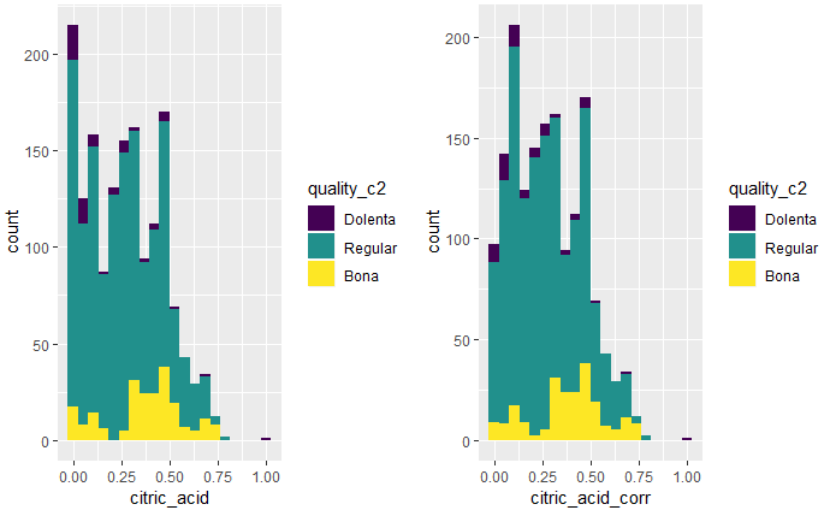

Fig. 2. Histogrames (apilats en funció de `quality_c2`) per als valors originals i corregits d'àcid cítric a les mostres.

Com es pot constatar als histogrames de la Fig. 2, els zeros afectaven si fa no fa a totes les categories de vi. Una vegada corregits, es té una distribució estadística de dades més en consonància amb el que es veurà més endavant que són les distribucions per a la resta de variables.

3.3. Normalització de les dades

Si es volen emprar algorismes de classificació cal normalitzar les dades per tal que la distància (euclidiana) entre elements no quedi dominada per paràmetres amb factors d'escala grans. Dues normalitzacions amb sentit són:

- Normalització de rang: les dades s'escalen per tal que tots els seus paràmetres ocupin el mateix rang de valors, per exemple a l'interval [0,1].
- Normalització de mitjana i variància: les dades s'escalen per tal que tots els seus paràmetres tinguin la mateixa mitjana (per exemple 0) i variància (per exemple 1).

S'ha triat la segona opció perquè conserva millor la dispersió *interna* de les dades. S'ha generat un nou data set, `dfn`, amb els valors normalitzats dels paràmetres objectius físico-químics:

```
# Normalitzem les dades del data set (llevat de les de qualitat del vi)
# per tal que tinguin mitja 0 i variança 1
dfn <- df %>% select(-(quality:quality_c3)) %>%
  lapply(function(x) {(x-mean(x))/sd(x)}) %>% as_tibble()
dfn <- dfn %>% mutate(quality    = df$quality,
                      quality_c1 = df$quality_c1,
                      quality_c2 = df$quality_c2,
                      quality_c3 = df$quality_c3)

#print(dfn %>% summary()%>% t())
```

3.4. Tractament dels *outliers*

Abans de tractar els *outliers* es pot fer una representació gràfica dels histogrames de les 12 variables del data set original (emprant la versió corregida de les dades de nivells d'àcid cítric, `citric_acid_corr`):

```
# Historgrames de les 12 variables numèriques del data set

aux <- df %>% select(-citric_acid)
```

```
for (i in 0:5) {
  aux1 <- ggplot(data = aux) +
    geom_histogram(mapping = aes(aux[[names(aux)[2*i+1]]],
                                 fill = quality_c2), bins = 20) +
    xlab(names(aux)[2*i+1])

  aux2 <- ggplot(data = aux) +
    geom_histogram(mapping = aes(aux[[names(aux)[2*i+2]]],
                                 fill = quality_c2), bins = 20) +
    xlab(names(aux)[2*i+2])

  grid.arrange(aux1, aux2, ncol = 2)
}
```

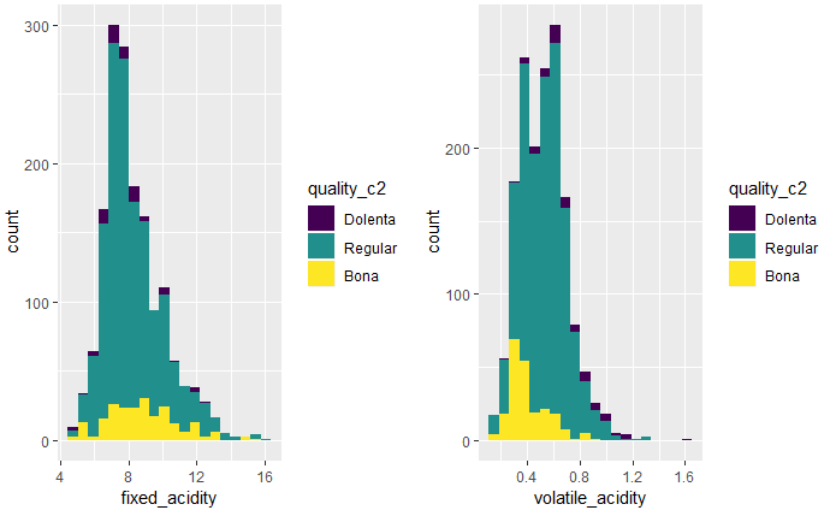

Fig. 3. Histogrames (apilats en funció de `quality_c2`) per a `fixed_acidity` i `volatie_acidity`.

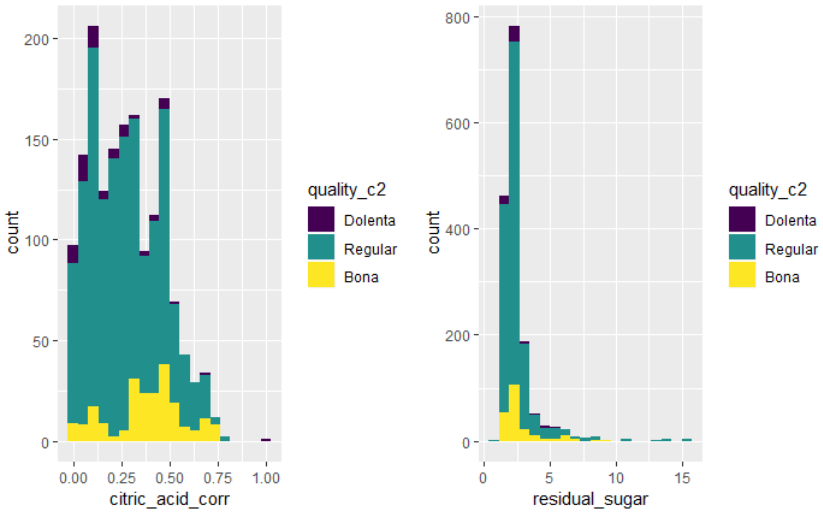

Fig. 4. Histogrames (apilats en funció de `quality_c2`) per a `cítric_acid_corr` i `residual_sugar`.

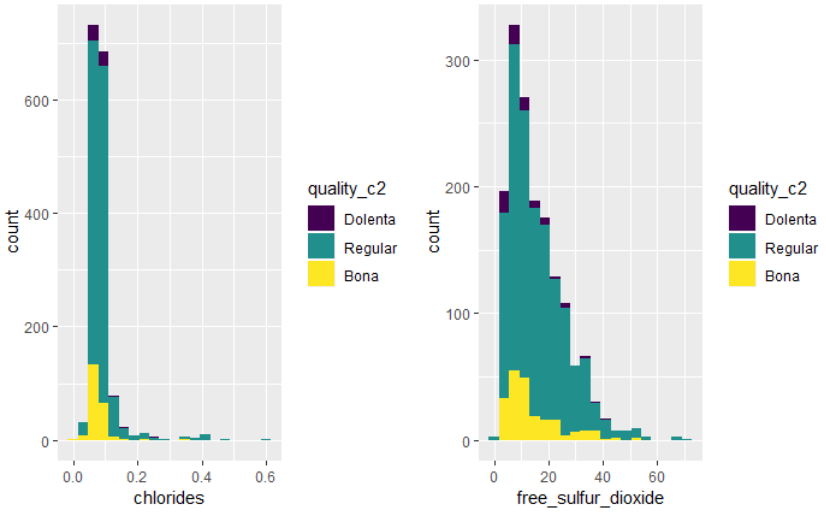

Fig. 5. Histogrames (apilats en funció de `quality_c2`) per a `chlorides` i `free_sulfur_dioxide`.

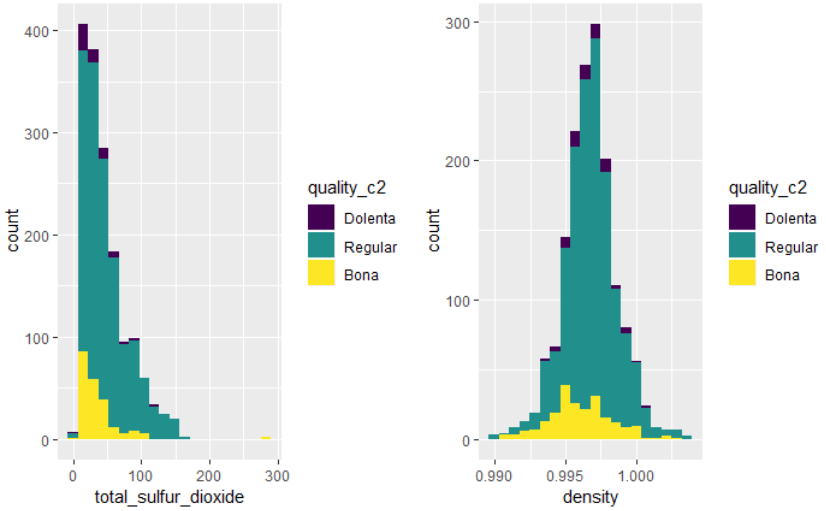

Fig. 6. Histogrames (apilats en funció de `quality_c2`) per a `total_sulfur_dioxide` i `density`.

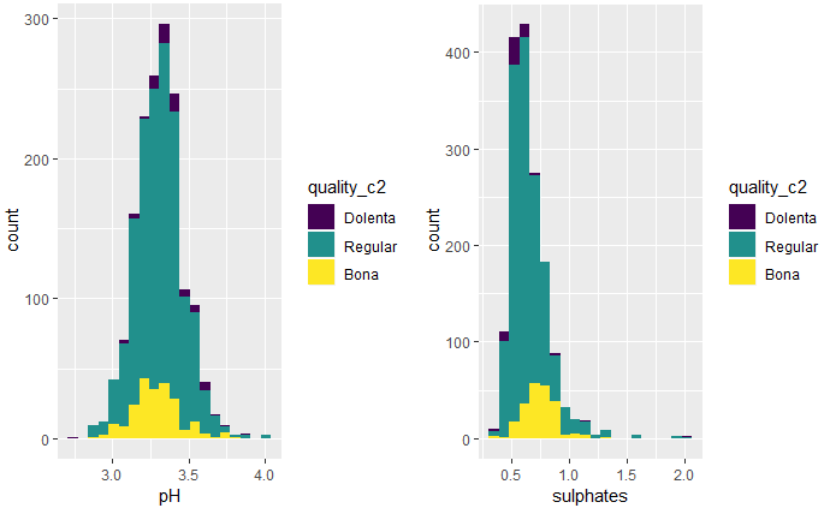

Fig. 7. Histogrames (apilats en funció de `quality_c2`) per a `pH` i `sulphates`.

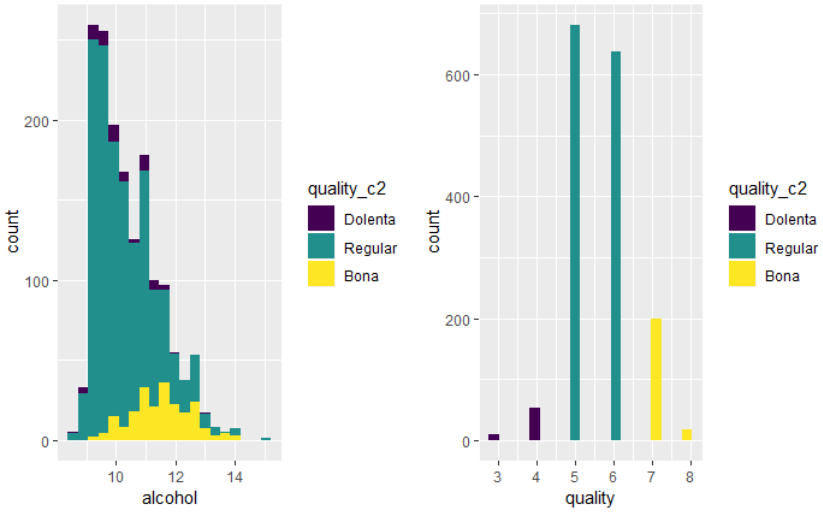

Fig. 8. Histogrames (apilats en funció de `quality_c2`) per a `alcohol` i `quality`.

Com es pot apreciar als histogrames de la Fig. 3 a la Fig. 8, per a bona part dels paràmetres hi ha mostres que queden força separades de la zona on es concentra la major part de les dades. També es pot comprovar que aquestes mostres solen correspondre a una qualitat subjectiva que s'ha etiquetat com a "Regular" (valors del paràmetre `quality` 5 o 6), que és, de llarg, la més abundant al data set.

Si es defineixen els *outliers* com aquelles mostres que se separen de la mitja més de tres desviacions estàndard, es pot analitzar com es distribueixen tant per files (mostres) com per columnes (paràmetres) del data set:

```
# Fem un còmput d'outliers per columna, definits com aqueslls valors
# que estan a més de 3*sigma de la mitjana. Emprem, per exemple, el
# tibble normalitzat (llevat de les columnes de valoració qualitativa)
cat("Nombre d'outliers per paràmetre: \n")
dfn %>% select(-(quality:quality_c3)) %>%
  lapply(function(x) { length(which(abs(x)>3)) }) %>%
  as_tibble() %>% t() %>% print()

# Comprovem també la quantitat d'outliers que té cada mostra
opm <- dfn %>% select(-(quality:quality_c3)) %>% as.matrix() %>%
  apply(1, function(x) { length(which(abs(x)>3)) })

ggplot() +
  geom_histogram(mapping = aes(opm, fill = dfn$quality_c2), bins = 6) +
  xlab("Nombre d'outliers per mostra")
```

```
Nombre d'outliers per paràmetre:
                     [,1]
fixed_acidity          12
volatile_acidity       10
citric_acid             1
citric_acid_corr        1
residual_sugar         30
chlorides              31
free_sulfur_dioxide    22
total_sulfur_dioxide   15
density                18
pH                      8
sulphates              27
alcohol                 8
```

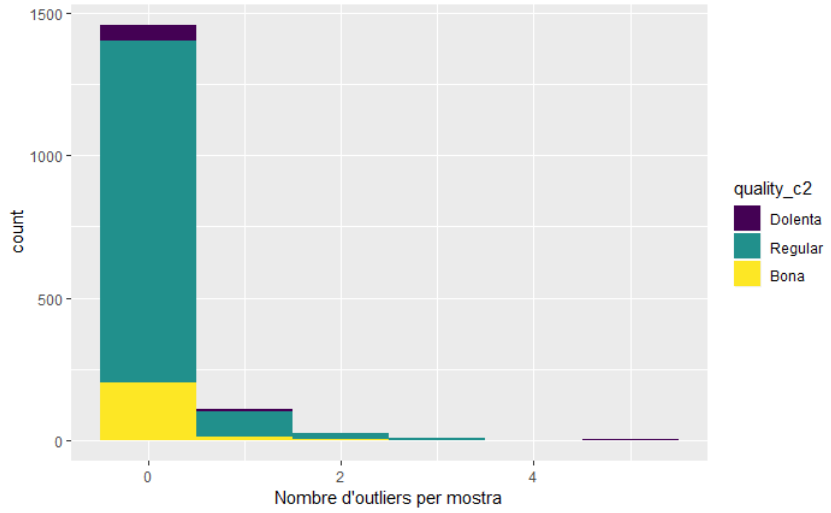

Fig. 9. Nombre d'outliers per mostra al data set.

Els resultats són coherents amb el que mostren els histogrames anteriors. Hi ha diverses estratègies per a limitar l'efecte dels *outliers* al procés posterior d'anàlisi:

- Es poden eliminar les mostres amb *outliers*
- Es poden col·lapsar els valors dels paràmetres que queden fora de l'interval (m–3s, m+3s), on m és la mitjana i s és la desviació estàndard de la mostra per a cada paràmetre als valors extrems de l'interval, m±3s
- Donat que la distribució de mostres és molt poc equilibrada (n'hi ha moltes més a la categoria de "Regular" que a les altres), es poden eliminar les mostres pertanyents a la categoria "Regular" i col·lapsar les altres als valors extrems dels intervals per al paràmetres afectats.

Per la seva senzillesa, i pel fet que no es perden dades que poden contenir altres valors rellevants, s'ha optat per la segona solució, col·lapsar els *outliers* per a cada paràmetre a m±3s:

```
# Eliminem els outliers de les mostres col·lapsant els seus valors
# a mitjana +/- 3 * sigma

# Funció que elimina els outliers d'una columna
el_outliers <- function(x) {
  mmm <- mean(x)
  sss <- sd(x)
  xn <- (x-mmm)/sss

  ifelse(xn < -3, -3, ifelse(xn > 3, 3, xn))
}

# Creem una versió sense outliers del tibbles df i dfn
df_no <- df %>%
         select(-(quality:quality_c3)) %>%
         lapply(el_outliers) %>%
         as_tibble() %>%
         mutate(quality    = df$quality,
                quality_c1 = df$quality_c1,
                quality_c2 = df$quality_c2,
                quality_c3 = df$quality_c3)

dfn_no <- dfn %>%
          select(-(quality:quality_c3)) %>%
```

```
          lapply(el_outliers) %>%
          as_tibble() %>%
          mutate(quality    = df$quality,
                 quality_c1 = df$quality_c1,
                 quality_c2 = df$quality_c2,
                 quality_c3 = df$quality_c3)
```

Cal notar que, fet això, caldria tornar a normalitzar el tibble `dfn_no` si es volgués que tots els seus paràmetres tinguessin exactament una mitjana 0 i una variància 1. Com que el que es pretenia amb la normalització era, essencialment, escalar els paràmetres perquè tinguessin nivells comparables, no s'ha tornat a normalitzar.

3.5. Gravació dels data sets pre-processats inicials

Per tal de conservar les tasques de pre-processament anteriors per a futures anàlisis, s'han desat els data sets `df_no` i `dfn_no` en fitxers amb format csv:

```
# Desem els data sets pre-processats
write_csv(df_no, "df_no.csv", col_names = TRUE)
write_csv(dfn_no, "dfn_no.csv", col_names = TRUE)
```

4. Anàlisi de les dades

4.1. Selecció dels grups de dades que es volen analitzar/comparar

Es pretén analitzar si es pot inferir la qualitat subjectiva de les mostres de vi a partir de les seves característiques físico-químiques. Donat que les dades per a tots els paràmetres (i fins i tot per a la qualitat subjectiva) són numèriques, es pot plantejar tant una anàlisi basada en regressions lineals com en algorismes de classificació (tractant la qualitat subjectiva `quality_c1` o alguna de les seves discretitzacions, `quality_c2` i `quality_c3`, com a variables categòriques).

Per a comprovar la bondat dels models de regressió o classificació, cal dividir les dades en conjunts d'entrenament i de test que, idealment, haurien de tenir una distribució estadística similar. Per tant, cal fer una divisió estratificada de les dades. A tal efecte s'ha implementat una funció de generació de conjunts estratificats (a partir de qualsevol variable categòrica que etiqueti tots els registres) d'entrenament i de test. La funció retorna quines files/registres/observacions pertanyen al conjunt d'entrenament o de test, ja sigui com a vector amb els nombres de files, ja sigui com a vector lògic indicant pertinença o no al conjunt.

```
# Dividim les dades en dos conjunts estratificats segons alguna
# de les variables categòriques quality_c1, quality_c2 o quality_c3


stratified_train_test_split <- function(estratificador,
                                        prop_dades_entrenament = 0.8,
                                        random_seed = 42) {
  # Funció que genera diversos vectors amb els ordinals o la posició
  # de les files per a una divisió estratificada d'un tibble/dataframe
  # de dades (representada per l'array de dades categòriques 'estratificador'
  # a partir de la qual generar la tria estratificada de files) en un
  # conjunt de files d'entrenament i un altre de test

  n_files = length(estratificador)

  set.seed(random_seed)

  # Emprem el paradigma SPLIT-APPLY-COMBINE
```

```
  # SPLIT: Separem els ordinals de fila del dataframe original en
  # una llista segons el valor de la variable categòrica
  # de la nostra elecció
  ldf <- split(1:n_files, estratificador)

  # APPLY: Per a  cada element de la llista (array d'ordinals que correspon
  # a un valor de la variable categòrica d'interès), generem un array
  # amb els ordinals de les files que corresponen a les mostres
  # entrenament, triades en un percentatge fix donat per
  # 'prop_dades_entrenament', fet que garanteix un mostreig
  # estratificat
  mostres <- lapply(ldf, function(x) {
    sample(x,as.integer(length(x)*prop_dades_entrenament),
           replace = FALSE)
  })

  # COMBINE: Combinem (concatenem amb la funció c()) els vectors obtinguts
  # amb la funció do.call(). Obtenin un vector amb els ordinals de les files
  # d'entrenament
  rows_train <- do.call(c, mostres)

  # Vector lògic que indica si una fila pertany al conjunt. d'entrenament:
  #   is_train[i] = TRUE si la fila i hi pertany; sinó, FALSE
  is_train <- is.element(1:n_files, rows_train)

  # Les files que no són d'entrenament són de test
  is_test <- !is_train
  rows_test <- which(is_test)

  list(rows_train = rows_train,
       is_train = is_train,
       rows_test = rows_test,
       is_test = is_test)
}

# Comprovem que funciona correctament...
aux <- stratified_train_test_split(df$quality_c3, 0.8)

cat("\nProp. de registres amb qualitat 'No adequada' al c. d'entrenament : \n",
    nrow(filter(df, aux$is_train, quality_c3 == 'No adequada')) /
    nrow(filter(df, quality_c3 == 'No adequada')))
cat("\nProp. de registres amb qualitat 'Adequada' al c. d'entrenament : \n",
    nrow(filter(df, aux$is_train, quality_c3 == 'Adequada')) /
    nrow(filter(df, quality_c3 == 'Adequada')))
cat("\nProp. de registres amb qualitat 'No adequada' al c. de test : \n",
    nrow(filter(df, aux$is_test, quality_c3 == 'No adequada')) /
    nrow(filter(df, quality_c3 == 'No adequada')))
cat("\nProp. de registres amb qualitat 'Adequada' al c. de test : : \n",
    nrow(filter(df, aux$is_test, quality_c3 == 'Adequada')) /
    nrow(filter(df, quality_c3 == 'Adequada')))
```

```
Prop. de registres amb qualitat 'No adequada' al c. d'entrenament :
 0.7995658
Prop. de registres amb qualitat 'Adequada' al c. d'entrenament :
 0.797235
Prop. de registres amb qualitat 'No adequada' al c. de test :
 0.2004342
Prop. de registres amb qualitat 'Adequada' al c. de test : :
 0.202765
```

Com es pot constatar, l'estratificació ha funcionat correctament.

4.2. Comprovació de la normalitat i homogeneïtat de la variància

Als histogrames de les dades de la Fig. 3 a la Fig. 8 ja s'ha pogut comprovar que la distribució de les dades no segueix, tot i ser acampanada, de manera molt exacta una distribució normal o gaussiana perquè a bona part d'elles hi ha un *skew* molt clar cap a la dreta.

Si es vol comprovar de manera rigorosa aquest fet amb un test estadístic, es pot emprar el test de Shapiro-Wilk (https://en.wikipedia.org/wiki/Shapiro-Wilk_test) per a cada característica o variable. En aquest test la hipòtesi nul·la és

H_0: La població té una distribució normal o gaussiana,

i l'alternativa és

H_a: la població no té una distribució normal o gaussiana.

Si s'estableix un nivell de significança usual, $\alpha = 0.05$, i s'exetuta el test per a obtenir els seus valors *p*:

```
# Mesures de normalitat de les dades mitjançant el test
# de Shapiro-Wilk
swt <- df_no %>% select(-(quality_c1:quality_c3)) %>%
                 lapply(shapiro.test)

# Imprimim els resultats...
tibble(parametre = names(swt),
       p_value = lapply(swt, function(x) x$p.value) %>% unlist()) %>%
  t() %>% t() %>% print()
```

```
      parametre             p_value
 [1,] "fixed_acidity"       "1.970872e-23"
 [2,] "volatile_acidity"    "4.343365e-13"
 [3,] "citric_acid"         "8.899014e-22"
 [4,] "citric_acid_corr"    "2.531365e-20"
 [5,] "residual_sugar"      "4.335346e-46"
 [6,] "chlorides"           "7.133154e-46"
 [7,] "free_sulfur_dioxide" "5.047708e-29"
 [8,] "total_sulfur_dioxide" "2.262108e-32"
 [9,] "density"             "2.179639e-07"
[10,] "pH"                  "7.814710e-04"
[11,] "sulphates"           "7.728106e-29"
[12,] "alcohol"             "8.704244e-27"
[13,] "quality"             "9.515085e-36"
```

Per tant, com que, per a cada paràmetre el valor *p* obtingut és (molt) menor que el nivell de significança establert, cal rebutjar la hipòtesi nul·la i acceptar la hipòtesi alternativa (la població no té una distribució normal o gaussiana) per a cap variable.

Que les dades no segueixen una distribució normal també pot constatar-se visualment a partir de gràfics quartil-quartil o Q-Q:

```
# Q-Q plots per a avaluar visualment la normalitat de les dades

df_aux <- df_no

for (i in 0:5) {
  aux1 <- ggplot(data = df_aux, aes(sample = df_aux[[names(df)[2*i+1]]])) +
    stat_qq(color = "blue") +
    stat_qq_line() +
```

```
    labs(title = names(df_aux)[2*i+1])

  aux2 <- ggplot(data = df_aux, aes(sample = df_aux[[names(df)[2*i+2]]])) +
    stat_qq(color = "blue") +
    stat_qq_line() +
    labs(title = names(df_aux)[2*i+2])

  grid.arrange(aux1, aux2, ncol = 2)
}
```

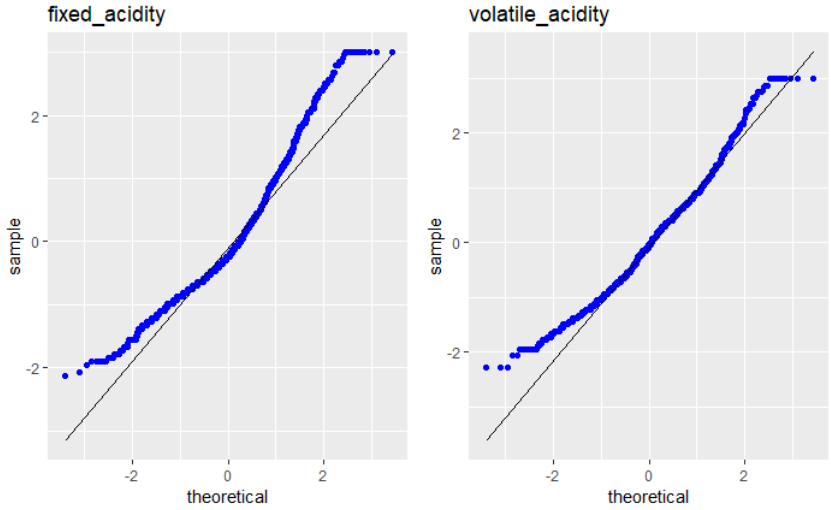

Fig. 10. Diagrames Q-Q per a `fixed_acidity` i `volatile_acidity`.

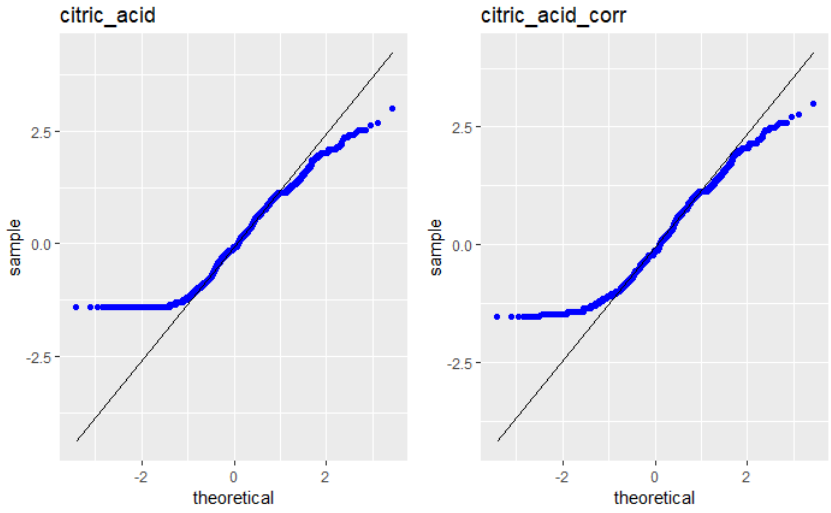

Fig. 11. Diagrames Q-Q per a `cítric_acid` i `cítric_acid_corr`.

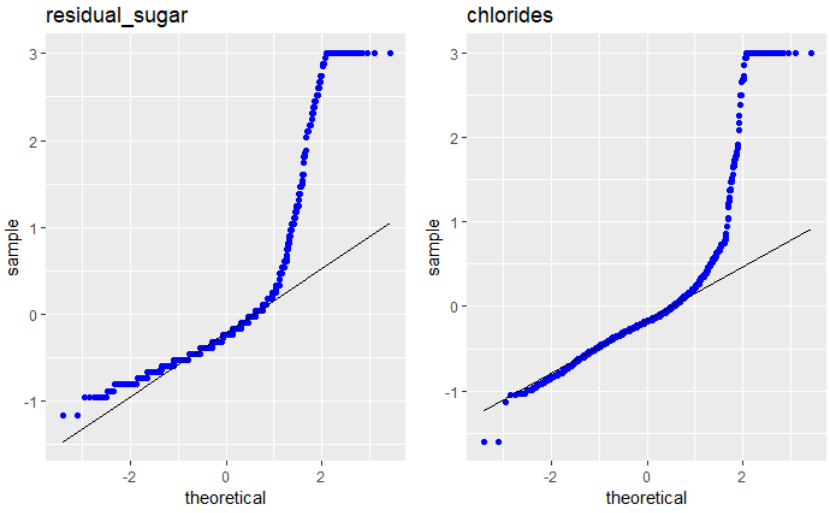

Fig. 12. Diagrames Q-Q per a `residual_sugar` i `chlorides`.

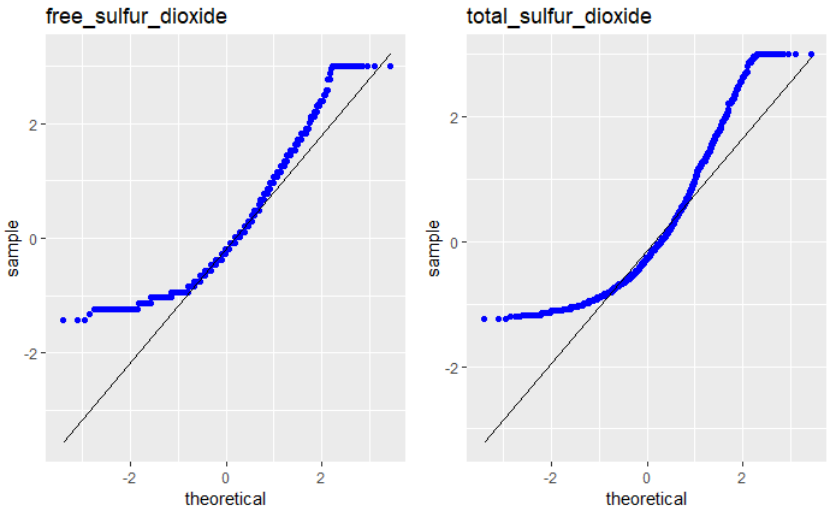

Fig. 13. Diagrames Q-Q per a `free_sulfur_dioxide` i `total_sulfur_dioxide`.

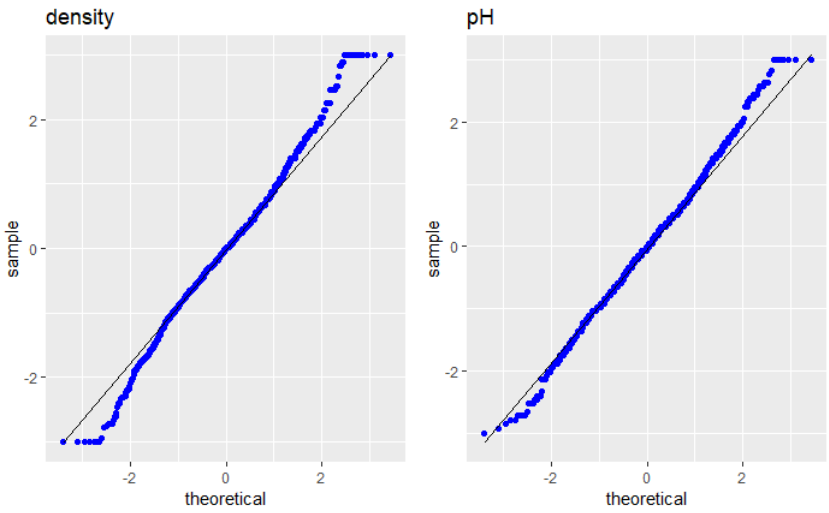

Fig. 14. Diagrames Q-Q per a `density` i `pH`.

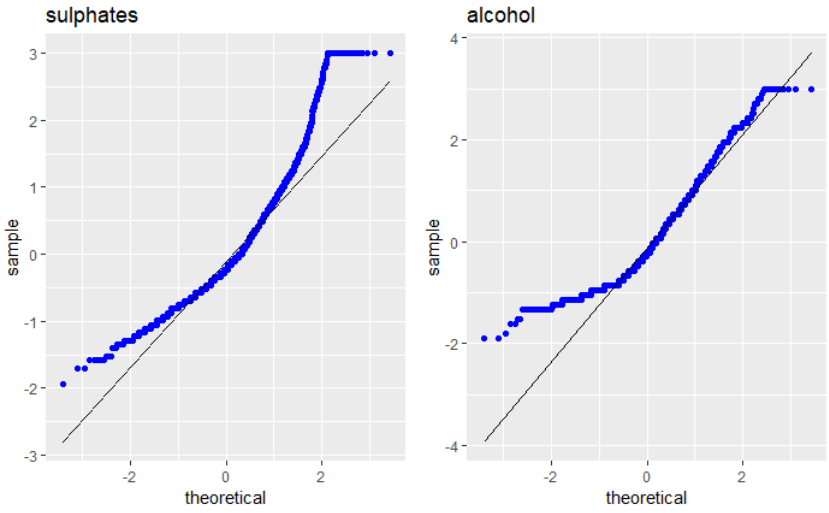

Fig. 15. Diagrames Q-Q per a `sulphates` i `alcohol`.

Com es pot constatar de la Fig. 10 a la Fig. 15, i de manera coherent amb els resultats del test de Shapiro-Wilk, les variables que més s'acosten a la normalitat són de de `density` i `pH`, i en menor mesura, la de `volatile_acidity` (són les que tenen valors *p* més elevats).

Com que les dades no s'adiuen gaire bé a una distribució normal, si es vol fer una anàlisi de la homogeneïtat de la variància entre mostres (per exemple entre mostres classificades segons la classificació binària de qualitat `quality_c3`, que pot prendre els valors "No adequada" (que correspon a les classificacions segons `quality_c2` "Dolenta" i "Regular") i "Adequada" (que correspon a la classificació segons `quality_c2` "Bona")) caldrà un test d'homogeneïtat de variàncies robust a distribucions no normals de les dades, el test de Fligner-Killeen. En aquest test, la hipòtesi nul·la és

H_0: Les variàncies dels dos o més grups són iguals

i l'alternativa és

H_a: Les variàncies dels dos o més grups són diferents.

Si triem un valor de significança usual, $\alpha = 0.05$, i executem el test per a obtenir els seus valors *p*:

```
# Test de Fligner-Killeen d'homogeneïtat de les variàncies

df_aux <- df_no
categ <- df_aux$quality_c3

# Apliquem el test a les variables físico-químiques
fkt <- df_aux %>% select(-(quality:quality_c3)) %>%
                  lapply(function(x) { fligner.test(split(x,categ)) })

# Imprimim els resultats...
tibble(parametre = names(fkt),
       p_value = lapply(fkt, function(x) x$p.value) %>% unlist()) %>%
  t() %>% t() %>% print()
```

```
     parametre             p_value
 [1,] "fixed_acidity"       "1.088917e-04"
 [2,] "volatile_acidity"    "1.195257e-04"
 [3,] "citric_acid"         "8.108406e-01"
 [4,] "citric_acid_corr"    "9.378638e-01"
```

```
 [5,] "residual_sugar"       "6.478662e-02"
 [6,] "chlorides"            "4.074174e-01"
 [7,] "free_sulfur_dioxide"  "2.881859e-02"
 [8,] "total_sulfur_dioxide" "4.675805e-09"
 [9,] "density"              "2.484952e-05"
[10,] "pH"                   "8.042085e-01"
[11,] "sulphates"            "6.071179e-01"
[12,] "alcohol"              "9.319097e-02"
```

Per als paràmetres en què el valor *p* obtingut és major que el nivell de significança establert, cal acceptar la hipòtesi nul·la (la variància és igual); per als que els valor *p* és menor que el nivell de significança establert , cal rebutjar la hipòtesi nul·la i acceptar la hipòtesi alternativa (les variàncies no són iguals). Per tant, per als paràmetres `fixed_acidity`, `volatile_acidity`, `free_sulfur_dioxide`, `total_sulfur_dioxide` i `density` les variàncies per als grups de mostres caracteritzades per `quality_c3` "No adequada" i "Adequada" seran diferents.

4.3. Igualtat de mitjanes per grups

Dels histogrames dels paràmetres presentats de la Fig. 2 a la Fig. 8 sembla desprendre-se'n que força d'entre ells presenten distribucions diferents en funció de la qualitat del vi ("Bona"– "Acceptable" i la resta). Si això és així, sembla que models de classificació o de regressió lineal haurien de tenir força possibilitats de discernir amb garanties els nivells de qualitat dels vi en funció dels seus paràmetres físico-químics.

Per tal d'escatir si les mostres tenen mitjanes diferents per als seus paràmetres en funció de la qualitat del vi es pot fer un test sobre la igualtat de mitjanes de les mostres segons la seva pertinença als grups definits per `quality_c3` ("No acceptable" o "Acceptable"). Com que les variàncies dels diversos grups en general poden ser diferents, tal com s'ha vist més amunt, caldrà emprar un test robust tant a variàncies diferents com a nombre de mostres diferents, com ara el test de Welch, en què la hipòtesi nul·la és

H_0: les mitjanes entre grups de mostres són iguals

i la hipòtesi alternativa és

H_a: les mitjanes entre grups de mostres són diferents:

```
# Test de Welch sobre la igualtat de mitjanes entre mostres

# Nivell de significança del test
alpha <- 0.05

df_aux <- df_no
noms <- df_aux %>% select(-citric_acid, -(quality:quality_c3)) %>%
                   names()
wtt = list()

for (nom in noms) {
  wtt[[nom]] <- t.test(df_aux %>% dplyr::filter(quality_c3 == "Adequada") %>%
                                  select(nom) %>%
                                  unlist(),
                       df_aux %>% dplyr::filter(quality_c3 == "No adequada") %>%
                                  select(nom) %>%
                                  unlist(),
                       alternative = "two.sided",
                       mu = 0,
                       var.equal = FALSE)$p.value
```

```
}

# Imprimim els resultats
tibble(parametre = names(wtt),
       p_value = unlist(wtt),
       mitjanes_iguals = unlist(wtt) >= alpha ) %>%
       t() %>% t() %>% print()
```

```
      parametre             p_value        mitjanes_iguals
 [1,] "fixed_acidity"       "2.529312e-05" "FALSE"
 [2,] "volatile_acidity"    "5.846986e-31" "FALSE"
 [3,] "citric_acid_corr"    "2.856100e-16" "FALSE"
 [4,] "residual_sugar"      "8.790867e-03" "FALSE"
 [5,] "chlorides"           "2.509605e-09" "FALSE"
 [6,] "free_sulfur_dioxide" "3.741139e-03" "FALSE"
 [7,] "total_sulfur_dioxide" "8.071217e-14" "FALSE"
 [8,] "density"             "2.324370e-07" "FALSE"
 [9,] "pH"                  "2.504388e-02" "FALSE"
[10,] "sulphates"           "2.140163e-22" "FALSE"
[11,] "alcohol"             "1.866307e-47" "FALSE"
```

Com es pot observar, per al nivell de significança usual, $\alpha = 0.05$, cal rebutjar la hipòtesi nul·la que les mitjanes entre grups mostrals són iguals.

Per tant, sembla que les perspectives d'algorismes de classificació o de regressió hagin de ser bones.

4.4. Correlacions entre les dades

Un altre paràmetre important a l'hora de veure la capacitat predictiva de les dades és la correlació que hi ha entre els seus diferents paràmetres:

```
# Calculem les correlacions entre columnes, tant per a les
# dades originals com per a les sense outliers
aux <- df_no %>% select(-citric_acid, -(quality_c1:quality_c3)) %>% cor()

aux[,1:3] %>% print()
aux[,4:6] %>% print()
aux[,7:9] %>% print()
aux[,10:12] %>% print()
```

```
                      fixed_acidity volatile_acidity citric_acid_corr
fixed_acidity           1.00000000     -0.264524447      0.678242580
volatile_acidity       -0.26452445      1.000000000     -0.558264233
citric_acid_corr        0.67824258     -0.558264233      1.000000000
residual_sugar          0.14071814      0.029704444      0.159096727
chlorides               0.14410034      0.096368588      0.179152461
free_sulfur_dioxide    -0.15483108     -0.006727315     -0.065320020
total_sulfur_dioxide   -0.11517824      0.090605852      0.003880329
density                 0.66961467      0.022370669      0.367208530
pH                     -0.68671196      0.235013880     -0.531028753
sulphates               0.20272765     -0.292267750      0.336911676
alcohol                -0.06724148     -0.207900780      0.125596379
quality                 0.12551624     -0.388451829      0.235712282

                      residual_sugar    chlorides free_sulfur_dioxide
fixed_acidity             0.14071814  0.144100337        -0.154831075
volatile_acidity          0.02970444  0.096368588        -0.006727315
citric_acid_corr          0.15909673  0.179152461        -0.065320020
residual_sugar            1.00000000  0.094152173         0.125197530
chlorides                 0.09415217  1.000000000        -0.004447748
free_sulfur_dioxide       0.12519753 -0.004447748         1.000000000
```

```
total_sulfur_dioxide     0.17892489  0.066323831         0.678786088
density                  0.37127206  0.286930267        -0.030847924
pH                      -0.08392028 -0.259551515         0.073442306
sulphates                0.02249843  0.256725390         0.048879498
alcohol                  0.08661815 -0.259254607        -0.069045047
quality                  0.02152858 -0.149763874        -0.050637377

                     total_sulfur_dioxide     density          pH
fixed_acidity                -0.115178242  0.66961467 -0.68671196
volatile_acidity              0.090605852  0.02237067  0.23501388
citric_acid_corr              0.003880329  0.36720853 -0.53102875
residual_sugar                0.178924892  0.37127206 -0.08392028
chlorides                     0.066323831  0.28693027 -0.25955151
free_sulfur_dioxide           0.678786088 -0.03084792  0.07344231
total_sulfur_dioxide          1.000000000  0.08669018 -0.05890716
density                       0.086690176  1.00000000 -0.34037538
pH                           -0.058907157 -0.34037538  1.00000000
sulphates                     0.015766168  0.15814245 -0.16222363
alcohol                      -0.225013984 -0.49810805  0.20100817
quality                      -0.200205453 -0.17580805 -0.05966150

                       sulphates     alcohol     quality
fixed_acidity         0.20272765 -0.06724148  0.12551624
volatile_acidity     -0.29226775 -0.20790078 -0.38845183
citric_acid_corr      0.33691168  0.12559638  0.23571228
residual_sugar        0.02249843  0.08661815  0.02152858
chlorides             0.25672539 -0.25925461 -0.14976387
free_sulfur_dioxide   0.04887950 -0.06904505 -0.05063738
total_sulfur_dioxide  0.01576617 -0.22501398 -0.20020545
density               0.15814245 -0.49810805 -0.17580805
pH                   -0.16222363  0.20100817 -0.05966150
sulphates             1.00000000  0.12488170  0.29494054
alcohol               0.12488170  1.00000000  0.47835764
quality               0.29494054  0.47835764  1.00000000
```

Com es pot constatar, els nivells de correlacions entre paràmetres i amb la qualitat subjectiva del vi no són gaire elevats en la majoria dels casos. Hi ha correlacions que, tot i no ser excessivament grans, són esperables entre paràmetres:

- `corr(fixed_acidity, cítric_acid_corr) = 0.67`, reflectint el fet que l'àcid cítric és un àcid no volàtil
- `corr(fixed_acidity, pH) = -0.68` i `corr(cítric_acid_corr, pH) = -0.53` perquè, a majors concentracions d'àcid, menor pH
- `corr(alcohol, density) = -0.49` perquè a més percentatge d'alcohol, menys densitat del vi.

Quant a la correlació entre els diversos paràmetres i la qualitat percebuda del vi, cal destacar les dues més elevades, una positiva amb el nivell d'alcohol (0.47) i una de negativa amb el nivell d'àcids volàtils (àcid acètic), que donen un gust avinagrat al vi. La segona és esperable. La primera aporta una informació rellevant sobre un paràmetre important a l'hora de formar-se una opinió d'un vi: els vins de més graduació tenen tendència a ser valorats millor.

De totes maneres, cap d'aquestes correlacions és prou alta perquè es pugui eliminar de la llista de paràmetres un dels dos paràmetres correlats. Si es vol investigar més a fons quina relació hi pot haver entre paràmetres i qualitat percebuda del vi caldrà recórrer a models de regressió o de classificació.

4.5. Generació d'un model de regressió lineal per a les dades

Per tal d'escatir si és possible predir o inferir la qualitat subjectiva d'un vi a partir dels seus paràmetres físico-químics, es pot generar un model de regressió lineal que predigui un valor (real) de qualitat per a un conjunt donat de característiques.

A tal efecte s'han generat dos conjunts estratificats de mostres, un d'entrenament amb el 80% de les mostres i un altre de test amb el 20% restant. S'ha generat un model de regressió lineal a partir de les dades d'entrenament, i se n'han mostrat els paràmetres més rellevants (coeficients de la regressió, error rms per mostra i correlació amb els valors reals de qualitat (`quality`)). A continuació s'han predit els valors de qualitat amb el conjunt de test i s'ha calculat l'error rms per mostra i correlació entre predicció i valor real. Finalment s'han discretitzat (arrodonint-los a l'enter més proper) els valors de predicció per al conjunt de test per tal de poder calcular

- la precisió de predicció del model, definida com la proporció de prediccions correctes,
- la precisió per tipus de mostra mitja, definida com la mitjana de les precisions que s'han obtingut per a cada classe o categoria (aquest factor de mèrit dóna idea de la capacitat que té el model de predir correctament de manera homogènia independentment de la classe de les dades d'entrada), i
- la taula de contingència entre valors predits (enters) i valors reals de qualitat.

S'ha emprat per a realitzar la regressió el conjunt de dades normalitzades `dfn_no` per tal que els valors dels coeficients de regressió donin directament idea de la importància de cada característica físico-química en la determinació de la qualitat percebuda del vi (emprar o no dades normalitzades no té cap efecte en els resultats.

```
# Predicció de qualitat a partir d'una regressió lineal
df_aux <- dfn_no

# Generem un comjunt de test i un d'entrenament
prop_entrenament <- 0.8
itts <- stratified_train_test_split(df_aux$quality_c1, prop_entrenament)

df_aux_train <- df_aux %>% dplyr::filter(itts$is_train)
df_aux_test <- df_aux %>% dplyr::filter(itts$is_test)

# Entrenem un model lineal. Emprem les dades corregides de 'citric_acid_corr'

# Generem la fórmula
# Generem una fórmula per a la regressió lineal de 'citric_acid'
# a partir d'una cadena de caràcters construïda a patir dels noms de les
# la resta de columnes del tibble df (llevat de les de quality, quality_c1
# i quality_c2 i citric_acid_corr)
f2 <- df_aux %>% select(-citric_acid, -(quality:quality_c3)) %>%
  names() %>% paste(collapse = "+") %>% paste("quality ~",.) %>%
  as.formula()

model2 <- lm(f2, data = df_aux_train)


# Imprimim i dibuixem les dades rellevants de la fase d'entrenament
cat("FASE D'ENTRENAMENT\n")
cat("Coeficients : \n")
print(model2$coefficients)
cat("\nError absolut rms per mostra : \n",
    sqrt(sum(model2$residuals**2)/nrow(df_aux_train)))
cat("\nCorrelació entre predicció i valors d'entrenament : \n",
    cor(df_aux_train$quality, model2$fitted.values))
```

```
# Fem la predicció  dels valors de quality per a les mostres
# de test
quality_pred <- predict(model2, df_aux_test)

# Imprimim les dades rellevants de la fase de test
cat("\n\nFASE DE TEST\n")
cat("Error absolut rms per mostra : \n",
    sqrt(sum((quality_pred-df_aux_test$quality)**2) /
             nrow(df_aux_test)))
cat("\nCorrelació entre predicció i valors de test : \n",
    cor(quality_pred, df_aux_test$quality))

# Funció robusta que arrodoneix a un enter
arrodoneix_a_enter <- function(x) {
  aux <- as.integer(x)
  ifelse(abs(x-aux) < 0.5 , aux, ifelse(x < 0, aux-1L, aux+1L))
}

# Transformem la predicció del model lineal en una
# predicció categòrica/classificació
quality_c1_pred <- quality_pred %>%
                arrodoneix_a_enter() %>%
                as.character() %>%
                factor(levels = levels2, ordered = TRUE)

# Imprimim els valors de classificació obtinguts
cat("\n\nCLASSIFICACIÓ DE LES MOSTRES DE TEST\n")
cat("Precisió de la classificació : \n",
    length(which(df_aux_test$quality_c1 == quality_c1_pred)) /
           length(quality_c1_pred))
cat("\nPrecissió per tipus de mostra mitja :\n",
    tibble(valor = df_aux_test$quality_c1,
           pred = quality_c1_pred) %>%
    split(df_aux_test$quality_c1) %>%
    lapply(function(x) {length(which(x$valor == x$pred)) /
                        length(x$valor) }) %>%
    do.call(c, .) %>% mean())
cat("\nTaula de contingència de la classificació :\n")
print(table(df_aux_test$quality_c1, quality_c1_pred))
```

```
FASE D'ENTRENAMENT
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity     citric_acid_corr
          5.62714164           0.08141680          -0.19142290          -0.04335335
      residual_sugar            chlorides  free_sulfur_dioxide total_sulfur_dioxide
          0.02623345          -0.10593236           0.04558396          -0.12377905
             density                   pH            sulphates              alcohol
         -0.06032274          -0.03855833           0.18375447           0.26533412

Error absolut rms per mostra :
 0.6471073
Correlació entre predicció i valors d'entrenament :
 0.5962242

FASE DE TEST
Error absolut rms per mostra :
 0.6283758
Correlació entre predicció i valors de test :
 0.6365514
```

```
CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió de la classificació :
 0.5838509
Precisió per tipus de mostra mitja :
 0.2621046
Taula de contingència de la classificació :
   quality_c1_pred
     3  4  5  6  7  8
  3  0  0  2  0  0  0
  4  0  0 10  1  0  0
  5  0  0 99 38  0  0
  6  0  0 46 80  2  0
  7  0  0  0 31  9  0
  8  0  0  0  2  2  0
```

Com es pot constatar els coeficients de la regressió amb més pes corresponen a aquelles característiques físico-químiques que ja mostraven més correlació amb la qualitat percebuda del vi: `alcohol` (positiu), `volatile_acidity` (negatiu) i `sulphates` (positiu). L'error absolut rms per mostra i la correlació no són gaire bons, tant per a les dades d'entrenament com, després, per a les de test. La taula de contingència, que mostra el nivell enter de qualitat associat a cada predicció real/decimal de qualitat del model de regressió, pot indicar quin és el problema: la presència molt més important de dades amb qualitats de valor 5 i 6 produeix un efecte de sobre-entrenament que fa decantar el model cap a predir valors al voltant de 5 i 6, fet que minimitzarà l'error quadràtic total de predicció a la fase d'entrenament, però que acaba generant una capacitat de predicció molt dolenta per a vins amb altres qualitats, tal com demostra la precisió per tipus de mostra mitja, que és de només 0.26.

4.6. Refinament del procés pre-processament de les dades

Una solució al problema anterior pot ser una disminució de la numerositat del data set per a les qualitats que corresponen a valors numèrics 5 i 6. Pensant en termes de la variable categòrica quality_c2, caldria disminuir la quantitat de dades amb una qualitat "Regular" (1319) de tal manera que fos equiparable a la de dades amb qualitat "Bona" (217) i "Dolenta" (63). Una possible tria *ad hoc* fóra triar 110 mostres amb qualitat 5 i 110 més amb qualitat 6, per tal que la quantitat de dades amb qualitat "Regular" fos equiparable, si més no, a la de dades amb qualitat "Bona", puix que la quantitat de dades amb qualitat "Dolenta" és minsa:

```
# Disminució de la numerositat per a mostres del data set amb qualitat 5 o 6
df_aux <- dfn_no

set.seed(42)

# Generem un vector de nombres de fila amb la numerositat
# reduïda

# SPLIT
aux <- split(1:nrow(df_aux), df_aux$quality_c1)
# APPLY
aux[['5']]<- sample(aux[['5']], 110, replace = FALSE)
aux[['6']]<- sample(aux[['6']], 110, replace = FALSE)
# COMBINE
m_reduides <- do.call(c, aux)
es_reduida <- is.element(1:nrow(df_aux), m_reduides)

# Generem un nou dada set amb la numerositat reduïda
# per a les qualitats "Regular" (5 i 6)
```

```
dfn_no_red <- df_aux %>% dplyr::filter(es_reduida)

# Imprimim la nova distribució de qualitats
dfn_no_red %>% select(quality_c1) %>% table() %>% print()
```

```
.
  3   4   5   6   7   8
 10  53 110 110 199  18
```

Si ara es torna a executar el mateix codi de regressió lineal per a aquest nou data set amb la numerositat reduïda, `dfn_no_red`, fent el canvi de la instrucció

```
df_aux <- dfn_no
```

per la instrucció

```
df_aux <- dfn_no_red
```

per a adaptar-lo al nou data set:

```
FASE D'ENTRENAMENT
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity     citric_acid_corr
        5.6988790441        -0.0005072521        -0.3424881391        -0.0882774688
      residual_sugar            chlorides  free_sulfur_dioxide total_sulfur_dioxide
       -0.0017383066        -0.3201439885         0.1032597248        -0.1802644889
             density                   pH            sulphates              alcohol
        0.0574547825        -0.2423542742         0.2840353305         0.4421392958

Error absolut rms per mostra :
 0.8106715
Correlació entre predicció i valors d'entrenament :
 0.7189667

FASE DE TEST
Error absolut rms per mostra :
 0.7157923
Correlació entre predicció i valors de test :
 0.7953291

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió de la classificació :
 0.5445545
Precisió per tipus de mostra mitja :
 0.3534091
Taula de contingència de la classificació :
   quality_c1_pred
     3  4  5  6  7  8
  3  0  1  1  0  0  0
  4  1  2  7  1  0  0
  5  0  1 16  5  0  0
  6  0  0  6 14  2  0
  7  0  0  0 17 23  0
  8  0  0  0  0  4  0
```

Com es pot constatar, hi ha hagut un re-balanceig dels coeficients de la regressió lineal que ha fet que, tot i que hagi augmentat l'error absolut rms per mostra, la capacitat del model d'adaptar-se a la realitat hagi millorat, tal com ho mosta l'increment notable en la correlació entre predicció i valor real de la qualitat. Tot i que la precisió de classificació (que distingeix barroerament entre ben classificat i mal classificat sigui una mica pitjor ara), si es mira amb detall com es classifica es pot observar que el nou classificador classifica millor en el sentit que classifica millor totes les mostres, i no només aquelles

de qualitat “Regular”, que era aquelles que el classificador entrenat amb el data set de l’apartat anterior classificava millor: la precisió per tipus de mostra mitja ha incrementat fins a 0.35.

4.7. Predicció de qualitat a partir d’un classificador kNN

Com a alternativa a una predicció de la qualitat a partir d’un model de regressió lineal, es pot emprar un classificador (per exemple un de molt senzill com el kNN) aprofitant el fet que les variables subjectives de qualitat són, en essència, categòriques, ja sigui `quality` com les seves diverses discretitzacions categòriques `quality_c1`, `quality_c2` o `quality_c3`.

Per a un classificador com kNN cal emprar dades normalitzades per tal de no esbiaixar la distància euclidià entre mostres.

Tenint en compte el que s’ha vist abans quant als efectes perniciosos del biaix de les prediccions cap a les classes més nombroses, es farà el procés de classificació/predicció tant amb el data set `dfn_no` com amb el de numerositat reduïda per a les mostres de qualitat `quality_c2` “Regular”, `dfn_no_red`. S’analitzarà la capacitat de predicció del classificador segons les tres modalitats de qualitat categòrica definides al començament:

- `quality_c1` : “3”, “4”, “5”, “6”, “7” i “8”
- `quality_c2` : “Dolenta”, “Regular” i “Bona”
- `quality_c3` : “No acceptable” i “Acceptable”.

S’ha pres un valor de *k* = 20 per a tots els classificadors, que dóna uns percentatges de classificació bons.

```
# Classificació kNN. Emprem els mateixos conjunts d'entremanemt i de test

dfn_aux <- dfn_no

k = 20

# Generem un comjunt de test i un d'entrenament
prop_entrenament <- 0.8
itts <- stratified_train_test_split(dfn_aux$quality_c1, prop_entrenament)

dfn_aux_train <- dfn_aux %>% dplyr::filter(itts$is_train)
dfn_aux_test <- dfn_aux %>% dplyr::filter(itts$is_test)


# PREDICCIÓ AMB QUALITY_C1
quality_c1_pred_knn    <-    knn(dfn_aux_train    %>%    select(-citric_acid,    -
(quality:quality_c3)),
                           dfn_aux_test       %>%       select(-citric_acid,       -
(quality:quality_c3)),
                           dfn_aux_train$quality_c1,
                           k = k) %>%
                           as.character() %>%
                           factor(levels = levels2, ordered = TRUE)

cat("PREDICCIÓ AMB 'quality_c1' (3,4,5,6,7,8) \n")
cat("Precisió : \n",
    length(which(dfn_aux_test$quality_c1 == quality_c1_pred_knn)) /
    length(quality_c1_pred_knn))
cat("\nPrecisió per tipus de mostra mitja :\n",
    tibble(valor = dfn_aux_test$quality_c1,
           pred = quality_c1_pred_knn) %>%
```

```
    split(dfn_aux_test$quality_c1) %>%
    lapply(function(x) {length(which(x$valor == x$pred)) /
                        length(x$valor) }) %>%
    do.call(c, .) %>% mean())
cat("\nTaula de contingència : \n")
print(table(dfn_aux_test$quality_c1, quality_c1_pred_knn))

# PREDICCIÓ AMB QUALITY_C2
quality_c2_pred_knn    <-    knn(dfn_aux_train    %>%    select(-citric_acid,    -
(quality:quality_c3)),
                           dfn_aux_test       %>%       select(-citric_acid,       -
(quality:quality_c3)),
                           dfn_aux_train$quality_c2,
                           k = k) %>%
                           as.character() %>%
                           factor(levels = levels3, ordered = TRUE)

cat("\n\nPREDICCIÓ AMB 'quality_c2' (Dolenta, Regular, Bona)\n")
cat("Precisió : \n",
    length(which(dfn_aux_test$quality_c2 == quality_c2_pred_knn)) /
    length(quality_c2_pred_knn))
cat("\nPrecisió per tipus de mostra mitja :\n",
    tibble(valor = dfn_aux_test$quality_c2,
           pred = quality_c2_pred_knn) %>%
    split(dfn_aux_test$quality_c2) %>%
    lapply(function(x) {length(which(x$valor == x$pred)) /
                        length(x$valor) }) %>%
    do.call(c, .) %>% mean())
cat("\nTaula de contingència : \n")
print(table(dfn_aux_test$quality_c2, quality_c2_pred_knn))

# PREDICCIÓ AMB QUALITY_C3
quality_c3_pred_knn    <-    knn(dfn_aux_train    %>%    select(-citric_acid,    -
(quality:quality_c3)),
                           dfn_aux_test       %>%       select(-citric_acid,       -
(quality:quality_c3)),
                           dfn_aux_train$quality_c3,
                           k = k) %>%
                           as.character() %>%
                           factor(levels = levels4, ordered = TRUE)

cat("\n\nPREDICCIÓ AMB 'quality_c3' (No acceptable, Acceptable)\n")
cat("Precisió : \n",
    length(which(dfn_aux_test$quality_c3 == quality_c3_pred_knn)) /
    length(quality_c3_pred_knn))
cat("\nPrecisió per tipus de mostra mitja :\n",
    tibble(valor = dfn_aux_test$quality_c3,
           pred = quality_c3_pred_knn) %>%
    split(dfn_aux_test$quality_c3) %>%
    lapply(function(x) {length(which(x$valor == x$pred)) /
                        length(x$valor) }) %>%
    do.call(c, .) %>% mean())
cat("\nTaula de contingència : \n")
table(dfn_aux_test$quality_c3, quality_c3_pred_knn)
```

```
PREDICCIÓ AMB 'quality_c1' (3,4,5,6,7,8)
Precisió :
 0.5465839
Precisió per tipus de mostra mitja :
 0.2522943
Taula de contingència :
```

```
   quality_c1_pred_knn
     3  4  5  6  7  8
  3  0  0  1  1  0  0
  4  0  0  9  2  0  0
  5  0  0 98 36  3  0
  6  0  0 49 67 12  0
  7  0  0  1 28 11  0
  8  0  0  0  3  1  0


PREDICCIÓ AMB 'quality_c2' (Dolenta, Regular, Bona)
Precisió :
 0.8229814
Precisió per tipus de mostra mitja :
 0.4091481
Taula de contingència :
         quality_c2_pred_knn
          Dolenta Regular Bona
  Dolenta       0      13    0
  Regular       0     253   12
  Bona          0      32   12


PREDICCIÓ AMB 'quality_c3' (No acceptable, Acceptable)
Precisió :
 0.8664596
Precisió per tipus de mostra mitja :
 0.6261445
Taula de contingència :
             quality_c3_pred_knn
              No adequada Adequada
  Adequada             31       13
  No adequada         266       12
```

Com es pot constatar, la capacitat de predicció és lleugerament inferior que la de la regressió lineal quan es tracta de predir els valors enters de qualitat. Quan aquests valors es van agrupant en criteris de qualitat més englobants (`quality_c2` o `quality_c3`), la capacitat predictiva del model augmenta. De totes maneres cal recalcar que un altre cop el model està sobre-entrenat per a detectar correctament els valors de la classe més nombrosa (la de qualitat `quality_c2` "Regular") i que, per tant, comet molts errors de classificació per a les menys nombroses: només cal veure la capacitat pèssima que té de predir la classe "Bona"-"Acceptable".

Si es mira d'arreglar aquesta situació, entrenant els classificadors kNN amb el data set amb numerositat reduïda (canviant la instrucció

```
dfn_aux <- dfn_no
```

per la instrucció

```
dfn_aux <- dfn_no_red
```

per a adaptar-lo al nou data set):

```
PREDICCIÓ AMB 'quality_c1' (3,4,5,6,7,8)
Precisió :
 0.5742574
Precisió per tipus de mostra mitja :
 0.3386364
Taula de contingència :
   quality_c1_pred_knn
```

```
     3  4  5  6  7  8
  3  0  0  2  0  0  0
  4  0  2  5  3  1  0
  5  0  0 17  1  4  0
  6  0  2  4  5 11  0
  7  0  0  1  5 34  0
  8  0  0  0  0  4  0


PREDICCIÓ AMB 'quality_c2' (Dolenta, Regular, Bona)
Precisió :
 0.7227723
Precisió per tipus de mostra mitja :
 0.5891608
Taula de contingència :
         quality_c2_pred_knn
          Dolenta Regular Bona
  Dolenta       2      11    0
  Regular       1      34    9
  Bona          0       7   37


PREDICCIÓ AMB 'quality_c3' (No acceptable, Acceptable)
Precisió :
 0.8217822
Precisió per tipus de mostra mitja :
 0.8161882
Taula de contingència :
             quality_c3_pred_knn
              No adequada Adequada
  Adequada             10       34
  No adequada          49        8
```

Com es pot constatar, la precisió millora per a `qualiti_c1` i empitjora en la resta de casos. El que és evident és que ara el model té una capacitat predictiva més homogènia amb la classe d'entrada, fet que es tradueix en un increment notable de la precisió per tipus de mostra mitja en tots tres casos. Per exemple, per a la classificació amb `quality_c3` en aquest segon cas és capaç de predir molt millor element de qualitat "Adequada", que abans eren classificats majoritàriament com a "No adequada" pel biaix de classificació que introduïa la quantitat enorme de dades d'aquesta classe.

5. Conclusions

En aquesta pràctica s'ha realitzat un petit projecte de mineria de dades seguint les etapes d'un projecte analític, fent especial èmfasi en les tasques de pre-processament i neteja de les dades, i en les d'anàlisi inicial de les dades. Quant a les tècniques de pre-processament i neteja de dades, s'han emprat tècniques de

- normalització de dades,
- generació de nous tipus de dades a partir de discretització de dades existents,
- tractament de zeros (el data set triat no tenia valors nuls) a partir de tècniques de regressió lineal,
- tractament d'outliers,
- reducció de la numerositat per a les classes més representades del data set per a mitigar els efectes de sobre-entrenament dels mètodes de predicció emprats que produeixen un biaix important dels predictors cap a aquestes classes.

S'ha pogut comprovar que els processos de pre-processament i neteja de dades són processos iteratius que cal anar repetint en funció de les necessitats posteriors d'anàlisi.

Quant a les tècniques d'anàlisi emprades, s'han emprat dues tècniques de predicció molt senzilles, regressions lineals i classificadors kNN, per entendre que l'èmfasi de la pràctica havia de ser en l'anàlisi i pre-processament inicial de les dades. S'ha pogut constatar que un data set amb un conjunt de dades amb un biaix clar cap a unes determinades classes és difícil de tractar amb mètodes supervisats elementals perquè les solucions que minimitzen els erros de predicció o de classificació tendeixen a predir o classificar malament les mostres de classes minoritàries. S'ha constatat com la homogeneïtat de la capacitat predictiva dels models millora (però no necessàriament la seva precisió absoluta) quan s'equiparen les numerositats de les mostres corresponents a cada classe. Per al data set emprat, una equiparació total no semblava, però, aconsellable vist el petit nombre de mostres amb qualitats subjectives extremes (de 3 o de 8).

Quant al comportament dels models desenvolupats per a predir la qualitat subjectiva del vi a partir dels seus paràmetres físico-químics objectius, s'ha vist que aquests en condicionen de manera clara la qualitat (els models trobats tenen un comportament força millor que el *random*, i la major part dels errors de classificació corresponen a classificacions de vins en classes errònies però contigües a la correcta). Per a millorar la capacitat de predicció dels models, caldria provar nous tipus de classificadors més complexos, com ara màquines de suport vectorial, xarxes neuronals o arbres de classificació, o combinacions de classificadors.

6. Codi

Tot el codi executat es troba, en format R Markdown, al fitxer `Practica_02.Rmd`. El codi presentat als apartats anteriors correspon als diversos *chunks* de codi R que es troben al fitxer.