Pràctica 2

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

Alumne: Miquel Ribó i Pal

Notes prèvies:

1. He mirar de donar resposta a tots els comentaris que vas fer-me a l'entrega parcial de la pràctica. En particular:

i. He afegit un apartat sobre selecció de variables en les classificacions amb regressions lineals
ii. He afegit un test de Kruskal-Wallis
iii. M'he decantat per a considerar els valors extrems com a legítims i no eliminar-los ni corregir-los
iv. Tot i que els nuls de la variable `cítric_acid` podrien ser legítims, he optat per corregir-los per a practicar tècniques de correcció de dades (en aquest cas a partir de regressions lineals)
v. Mantinc l'estructura de la memòria com a l'entrega parcial, ja que et va semblar correcta.

2. Probablement moltes de les manipulacions de dades que he hagut de fer a la pràctica (divisions estratificades, validacions creauades estratificades, variacions de paràmetres) es podrien haver fet de manera més compacta, elegant i eficient amb la llibreria `caret` d'`R`, l'existència de la qual vaig descobrir a mig (re)fer la pràctica. Vaig optar, no obstant això, per continuar programant aquestes manipulacions a partir d'instruccions bàsiques d'`R` per tal de millorar les meves capacitats de programació en `R`. Espero que el codi no hagi quedat, per aquest motiu, massa embolicat.

3. He afegit a l'anàlisi classificacions amb *Random Forest* i C5.0, per tal d'aconseguir resultats de classificació millors que els que havia aconseguit amb regressions lineals i kNN.

1. Descripció del data set

S'ha triat el data set *Red Wine Quality* [1, 2] 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 português) 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 [3]. Mesurada en g/l [4, 5]. Els àcids no volàtils predominants al vi són el tartàric (usualment en concentracions d'1 a 4 g/l), el màlic (usualment en concentracions de 0 a 8 g/l), el cítric (usualment en concentracions de 0 a 0.5 g/l) i el i el succínic (usualment en concentracions de 0.5 a 2 g/l).
- **`volatile acidity`**: Nivell d'àcid acètic, que en quantitats elevades dóna un gust avinagrat al vi. Mesurat en grams litre (g/l) [4, 5]. Poden donar-se concentracions des de pràcticament indetectables fins a 3 g/l. Per a donar un límit superior a les concentracions usuals, es pot comentar que als USA, la concentració màxima permesa d'àcids volàtils és de 1.2 g/l per al vi negre.
- **`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 [3]. Pot donar aroma i frescor als vins. Mesurat en g/l [4, 5]. Sol prendre valors entre 0 i 0.5 g/l.
- **`residual sugar`**: Nivell de sucre que roman al vi un cop acabada la fermentació. Mesurat en g/l. Sol ser superior a 1 g/l. Vins amb nivells superiors a 45g/l es consideren dolços.
- **`chorides`**: Nivell de sal (NaCl) al vi. Expressat en g (de NaCl)/l [4]. Per a tenir una idea de nivells, a Austràlia el nivell de clorurs al vi està limitat a un màxim d'1g/l [6].
- **`free sulful dioxide`**: Nivell de SO_2 en forma lliure. Prevé la contaminació microbiana i l'oxidació del vi. Mesurat en mg/l [4, 5].
- **`total sulfur dioxide`**: Nivell de SO_2 (també anomenat sulfit) 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. Mesurat en mg/l [4, 5], i regulat per la UE a un màxim de 160 mg/l per al vi negre [7]. Puix que la densitat del vi és aproximadament d'un Kg/l, també pot expressar-se de manera equivalent en unitats de parts per milió (ppm) o mg/Kg.
- **`densitat`**: Més o menys propera a la de l'aigua (que és 1 Kg/l) en funció dels nivells de sucre i alcohol. Mesurada en Kg/l [4].
- **`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. Nivells expressats en g (de sulfat de potassi)/l [4].
- **`alcohol`**: Percentatge (% del volum) d'alcohol del vi.

- **`quality`**: Qualitat (subjectiva) del vi, basada en les seves característiques sensorials. Pren valors de 0 a 10, tot i que al data set només hi ha registrats valors de 3 a 8.

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 de [1].

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)
library(randomForest)
library(C50)

# 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.

Cal recalcar que hi ha alguns vins amb nivells de sulfits per sobre dels nivells màxims permesos a la legislació (160 mg/l).

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 a [1], les dades de menys 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 [8]. Com que no hi ha un nombre excessiu de dades ni de dimensions, en principi no cal 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).

També s'han definit tres versions numèriques enteres d'aquestes variables, que serviran més endavant per a fer classificacions categòriques a partir de regressions lineals:

- **`quality_n1`**: versió numèrica entera de `quality_c1` (redundant amb `quality`, que es manté per tal de mantenir la coherència de les dades originals)
- **`quality_n2`**: versió numèrica entera de `quality_c2` (segons l'equivalència "Dolenta" == 1, "Regular" == 2, "Bona" == 3)
- **`quality_n3`**: versió numèrica entera de `quality_c3` (segons l'equivalència "No acceptable" == 0, "Acceptable" == 1).

D'aquesta manera es tindran diversos criteris de classificació/regressió 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". Generem
# també versions numèriques enteres de tots els factors.

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

aux1n <- df$quality
aux1c <- df$quality %>% as.character() %>%
         factor(.,levels = levels2, ordered = TRUE)

aux2n <- ifelse(df$quality <= 4L, 1L, ifelse(df$quality >= 7L, 3L, 2L))
aux2c <- ifelse(df$quality <= 4L, "Dolenta", ifelse(df$quality >= 7L, "Bona",
"Regular")) %>%
         factor(.,levels = levels3, ordered = TRUE)

aux3n <- ifelse(df$quality <= 6L, 0L, 1L)
aux3c <- ifelse(df$quality <= 6L, "No adequada", "Adequada") %>%
```

```
         factor(.,levels = levels4, ordered = TRUE)

# Incorporem aquestes noves variables categòriques al tibble df
# per al seu ús posterior
df <- df %>% mutate(quality_n1 = aux1n,
                    quality_c1 = aux1c,
                    quality_n2 = aux2n,
                    quality_c2 = aux2c,
                    quality_n3 = aux3n,
                    quality_c3 = aux3c)

# 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
No adequada    Adequada
       1382         217
```

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 `citric_acid` poden ser petits, típicament entre 0 i 0.5 g/l, uns valors estrictament iguals a 0.00 poden indicar que es tracta d'un paràmetre arrodonit, o no mesurat (un fals `NA`). El fet que el raïm presenti ja petites quantitats d'àcid cítric i que els nivells usuals d'aquest àcid siguin de l'ordre d'1/20 dels de l'àcid tartàric fa pensar que efectivament es pot tractar d'un cas de falsos nuls. A la vista d'això, hi ha diverses opcions possibles:

- Es pot deixar el data set tal com està considerant que els valors d'àcid cítric de valor 0.00 corresponen a valors reals (o a arrodoniments al segon decimal de valors reals).
- 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 [9].
- Es pot mirar d'inferir el valor correcte a partir d'un model de regressió lineal (perquè es tracta d'un valor continu) entrenat a partir dels casos en què la variable no pren el valor zero [9].

Tot i ser l'opció més complicada, s'ha pres la darrera opció perquè preserva la mida del data set i, en principi, també la variància de les dades [9]. S'ha generat, per tant, una nova variable, `citric_acid_corr`, que s'ha afegit al data set a continuació de `cítric_acid` per tal de no perdre les dades originals, amb valors corregits d'àcid per als zeros:

```
# 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
```

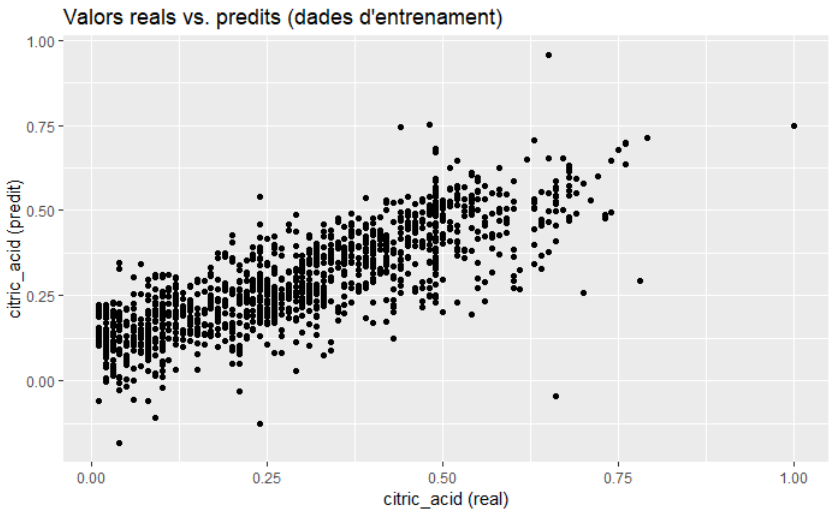

Fig. 1. Diagrama de dispersió per als valors reals i predits d'àcid cítric per al conjunt de dades d'entrenament.

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)

# Dibuixem l'histograma de només els valors corregits de citric_acid
ggplot(data = df_predict) +
  geom_histogram(mapping = aes(citric_acid_corr, fill = quality_c2), bins = 20)
```

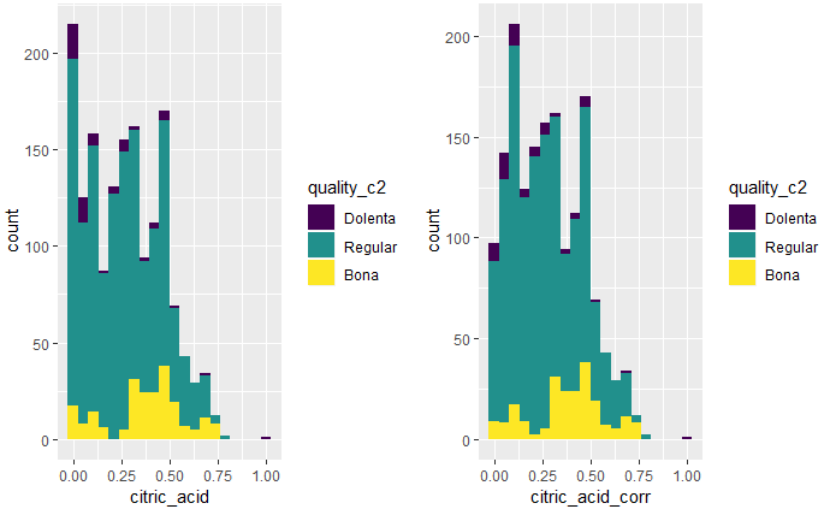

Fig. 2. Histogrames (apilats en funció de `quality_c2`) per als valors originals i corregits d'àcid cítric a les mostres.

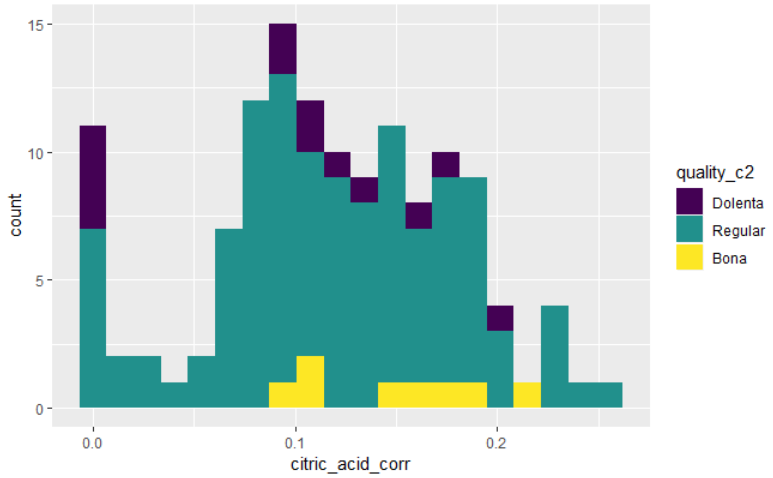

*Fig. 3. Histograma (aplita en funció de funció de `quality_c2`) només per **només** als valors corregits de zeros.*

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. Si es mira com han quedat distribuïts els valors corregits de zeros a partir de la regressió lineal feta, es pot constatar (Fig. 3) com els zeros han estat corregits en general a valors baixos de concentració d'àcid, fet que va preveure que no hi haurà massa diferències efectives entre emprar la variable original o la corregida.

Com que s'ha conservat també al data set els valors originals de la variable, es podran fer proves més endavant sobre la capacitat predictiva d'una o altra variable.

3.3. Tractament dels *outliers*

Primerament es pot fer una representació gràfica dels histogrames de les 12 variables del data set original (emprant, per exemple, 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)
}
```

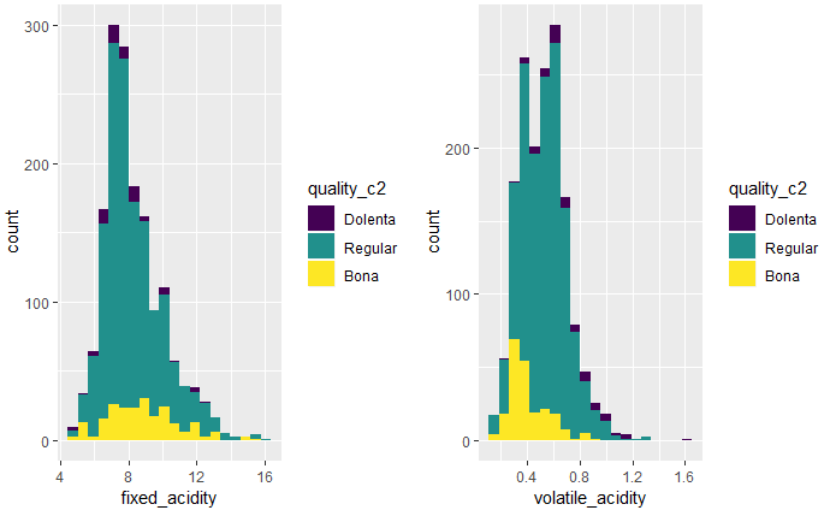

Fig. 4. Histogrames (apilats en funció de `quality_c2`) per a `fixed_acidity` i `volatie_acidity`.

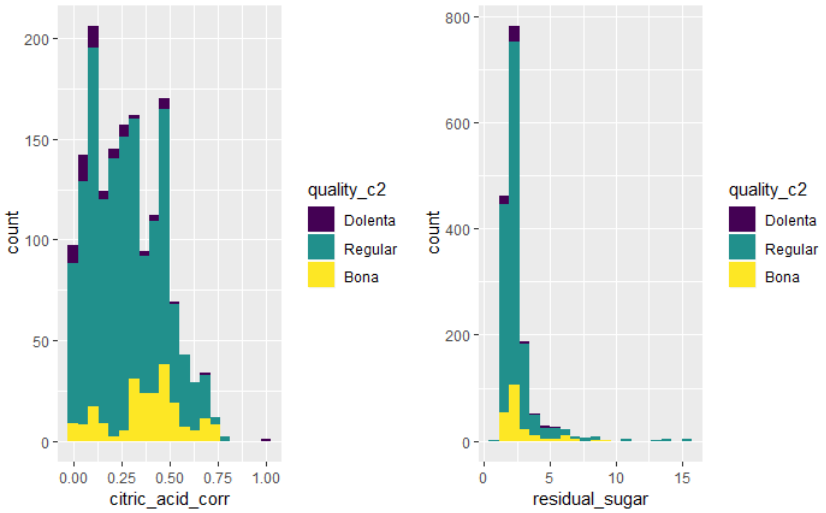

Fig. 5. Histogrames (apilats en funció de `quality_c2`) per a `cítric_acid_corr` i `residual_sugar`.

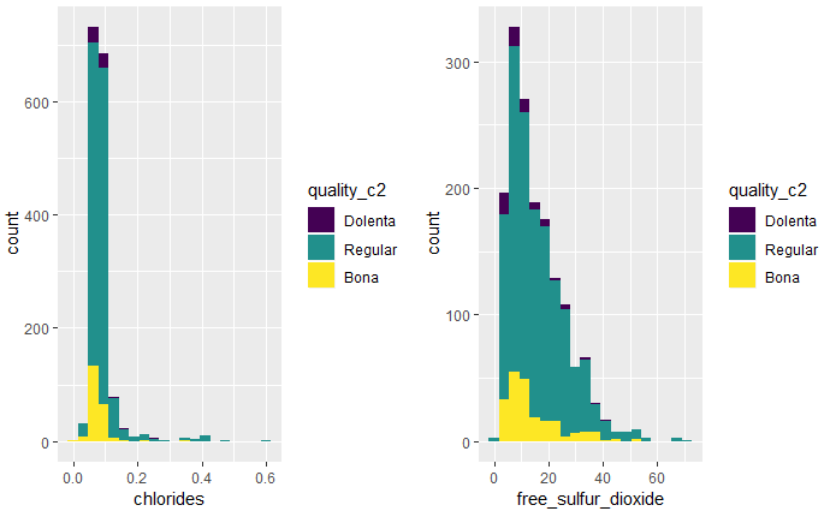

Fig. 6. Histogrames (apilats en funció de `quality_c2`) per a `chlorides` i `free_sulfur_dioxide`.

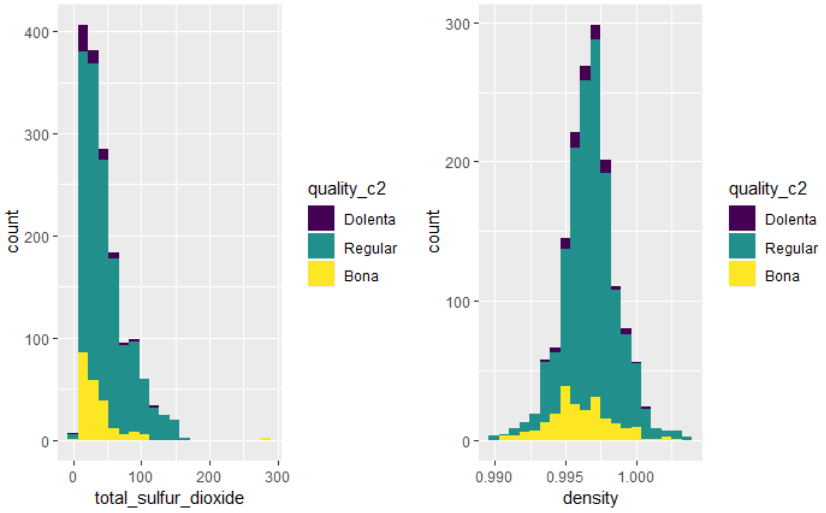

Fig. 7. Histogrames (apilats en funció de `quality_c2`) per a `total_sulfur_dioxide` i `density`.

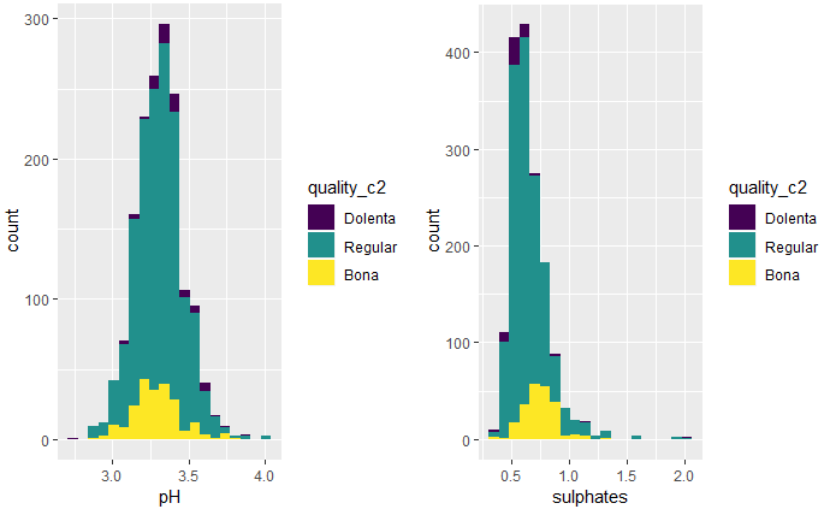

Fig. 8. Histogrames (apilats en funció de `quality_c2`) per a `pH` i `sulphates`.

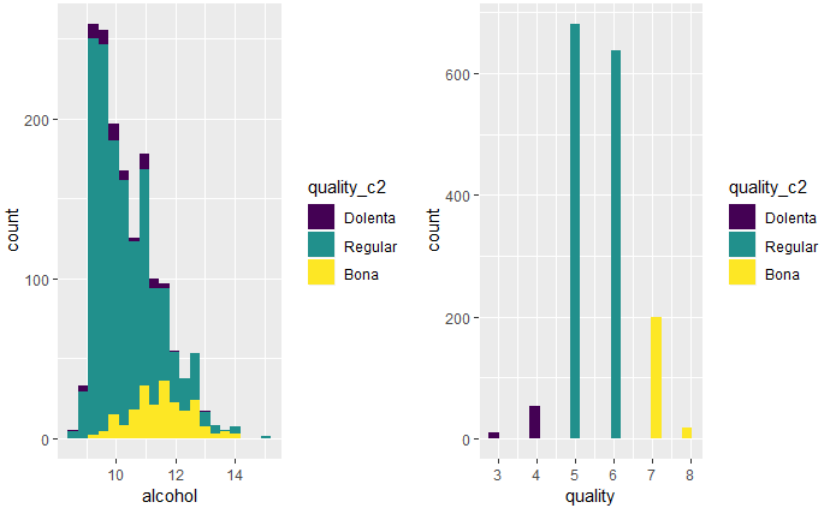

Fig. 9. Histogrames (apilats en funció de `quality_c2`) per a `alcohol` i `quality`.

Com es pot apreciar als histogrames de la Fig. 4 a la Fig. 9, per a bona part dels paràmetres hi ha algunes 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.

Per a tenir una idea quantitativa de l'abast de l'existència d'aquestes mostres aïllades, se'n pot calcular el nombre (definint-les per exemple com aquelles que són fora de l'intèrval ($m - 3s$, $m + 3s$), on m és la seva mitjana mostral i s la seva desviació estàndard mostral) per a cada paràmetre:

```
# Fem un còmput de valors per columna que estan a més
# de 3*sigma de la mitjana.
cat("Nombre de valors fora de (m-3s, m+3s) per paràmetre: \n")
df %>% select(-(quality:quality_c3)) %>%
  lapply(function(x) { length(which(abs(x-mean(x))>3*sd(x))) }) %>%
  as_tibble() %>% t() %>% print()
```

```
Nombre de valors fora de (m-3s, m+3s) 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
```

No obstant això, aquest fet no implica necessàriament que siguin *outliers*: poden ser mostres perfectament legítimes donada la funció de densitat de probabilitat multivariable subjacent a les dades i que és, malauradament, desconeguda i difícil d'inferir. Per exemple, es pot constatar com hi ha alguns valors força elevats d'acidesa fixa o de sucre residual. Aquests dos paràmetres, però, poden variar de manera harmònica per a donar un tipus de vi amb unes característiques de sequedat comparables amb les d'altres vins amb molt menys contingut de sucre residual perquè, a efectes de normativa [7], s'identifica un vi sec com aquell en què

- el nivell de sucre residual està per sota de 4 g/l, o
- el nivell de sucre residual està per sota de 9 g/l quan el nivell d'acidesa total no està més avall de 2 g/l respecte del nivell de sucre residual.

Per tant, tot i que es pot veure com la immensa majoria dels vins analitzats tenen un contingut de sucre per sota de 4 g/l (compleixen la 1a condició), n'hi pot haver amb continguts de sucre força més elevats que encara puguin considerar-se com a secs. De fet, si s'analitza quants vins compleixen la 1a condició, i quants la 2a però no la 1a:

```
# Anàlisi de vins segons acidesa i sucre residual

# Quantitat de vins dins la categoria "Sec"
df %>% filter((residual_sugar < 4) |
              ((residual_sugar < 9) &
               (residual_sugar-(fixed_acidity+volatile_acidity) < 2))) %>%
       count() %>% unlist() %>%
```

```
       cat("Quantitat de vins secs :", .)

# Quantitat de vins dins de la categoria "residual_sugar < 4"
df %>% filter(residual_sugar < 4) %>%
       count() %>% unlist() %>%
       cat("\nQuantitat de vins amb residual_sugar < 4 :", .)

# Quantitat de vins dins la categoria "Sec" amb "residual_sugar > 4"
df %>% filter((residual_sugar >= 4),
              ((residual_sugar < 9) &
               (residual_sugar-(fixed_acidity+volatile_acidity) < 2))) %>%
       count() %>% unlist() %>%
       cat("\nQuantitat de vins secs amb residual_sugar > 4 :", .)

# Vins fora de la categoria "Sec"
df %>% filter(residual_sugar >= 4,
              (residual_sugar >= 9) |
              (residual_sugar-(fixed_acidity+volatile_acidity)> 2)) %>%
       count() %>% unlist() %>%
       cat("\nQuantitat de vins no secs :", .)

# Interval (m-3s, m+3s) per a "residual_sugar"
cat("\n\nInterval (m-3s, m+3s) per a residual_sugar : (",
    mean(df$residual_sugar)-3*sd(df$residual_sugar), ",",
    mean(df$residual_sugar)+3*sd(df$residual_sugar),")")
# Nombre de vins amb "residual_sugar > m+3s"
df %>% dplyr::filter(residual_sugar >
                     mean(residual_sugar)+3*sd(residual_sugar)) %>%
       count() %>% unlist() %>%
       cat("\nQuantitat de vins amb residual_sugar > m+3s :", .)

# Interval (m-3s, m+3s) per a "fixed_acidity"
cat("\nInterval (m-3s, m+3s) per a fixed_acidity : (",
    mean(df$fixed_acidity)-3*sd(df$fixed_acidity), ",",
    mean(df$fixed_acidity)+3*sd(df$fixed_acidity),")")
# Nombre de vins amb "fixed_acidity > m+3s"
df %>% dplyr::filter(fixed_acidity >
                     mean(fixed_acidity)+3*sd(fixed_acidity)) %>%
       count() %>% unlist() %>%
       cat("\nQuantitat de vins amb fixed_acidity > m+3s :", .)
```

```
Quantitat de vins secs : 1587
Quantitat de vins amb residual_sugar < 4 : 1463
Quantitat de vins secs amb residual_sugar > 4 : 124
Quantitat de vins no secs : 12

Interval (m-3s, m+3s) per a residual_sugar : ( -1.690979 , 6.76859 )
Quantitat de vins amb residual_sugar > m+3s : 30
Interval (m-3s, m+3s) per a fixed_acidity : ( 3.096348 , 13.54293 )
Quantitat de vins amb fixed_acidity > m+3s : 12
```

Com es pot observar, hi ha força vins que (124) tenen una classificació de vi sec compensant nivells elevats d'acidesa amb nivells elevats de sucre residual. Per tant, el fet que hi hagi, per exemple, 30 vins amb nivells de sucre per sobre de $m + 3s$ no és un indicador precís que hi hagi un nivell gran d'*outliers* al data set quant a aquesta variable.

De fet, es podria considerar que els *outliers* del data set, quant a nivells d'acidesa i de sucre residual són els 12 vins que no corresponen a la classificació de vi sec.

De la mateixa manera es podria pensar que els pocs vins amb nivells de sulfats elevats (de l'ordre de 2g/l) o de clorurs (de l'ordre de més de 0.3) no tenen perquè ser, a priori, *outliers*, sinó vins amb unes característiques especials.

Finalment cal mencionar que, com s'ha comentat abans, hi ha una sèrie de vins que tenen uns nivells de sulfits per sobre dels màxims legals:

```
# Nivells il·legals de sulfits
df %>% filter(total_sulfur_dioxide > 160) %>%
       select(total_sulfur_dioxide, free_sulfur_dioxide, quality) %>%
       t() %>% print()
```

```
                       [,1]  [,2]  [,3]
total_sulfur_dioxide 165.0 278.0 289.0
free_sulfur_dioxide   40.5  37.5  37.5
quality                6.0   7.0   7.0
```

A la vista d'aquests resultats, cal preguntar-se quines de les mostres amb paràmetres discrepants formen part de l'univers d'anàlisi i quines no. Sembla que la pràctica totalitat dels vins analitzats corresponen a la categoria de vi sec. ¿Cal eliminar les poques mostres que no en formen part, tot i que probablement tinguin nivells de sucre residual i acidesa no gaire distants dels de la resta? ¿Cal eliminar de les mesures les tres mostres amb nivells de sulfits massa elevats, tot i que en general tenen bones percepcions de qualitat? ¿Cal eliminar les mostres amb nivells massa elevats de clorurs o de sulfats (o col·lapsar els seus valors a $m \pm 3s$, o corregir-los emprant una regressió lineal), tot i que no tenen perquè ser a priori errors de mesura sinó correspondre a opcions vàlides de disseny dels paràmetres del vi?

No sembla haver-hi una resposta a priori clara per a les qüestions anteriors. A la vista de tot els que s'ha comentat més amunt, potser sembla més adient treballar amb el data set sense mirar de fer cap filtratge o tractament d'*outliers*, ja que no s'ha identificat cap criteri convincent per a l'eliminació de mostres, o la correcció dels seus valors.

3.4. 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. La normalització també ajuda a interpretar la importància de cada variable en models de regressió lineal. 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$quality,
                      quality_n1 = df$quality_n1,
                      quality_c1 = df$quality_c1,
                      quality_n2 = df$quality_n2,
                      quality_c2 = df$quality_c2,
```

```
                        quality_n3 = df$quality_n3,
                        quality_c3 = df$quality_c3)
```

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` i `dfn` en fitxers amb format csv:

```
# Desem els data sets pre-processats
write_csv(df, "df.csv", col_names = TRUE)
write_csv(dfn, "dfn.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. 4 a la Fig. 9 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'executa el test per a obtenir els seus valors *p*:

```
# Mesures de normalitat de les dades mitjançant el test
# de Shapiro-Wilk
alpha <- 0.05

swt <- df %>% select(-(quality_n1:quality_c3)) %>%
              lapply(shapiro.test)

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

```
      parametre             p_value        distribucio_normal
 [1,] "fixed_acidity"       "1.525012e-24" "FALSE"
 [2,] "volatile_acidity"    "2.692935e-16" "FALSE"
 [3,] "citric_acid"         "1.021932e-21" "FALSE"
 [4,] "citric_acid_corr"    "2.877469e-20" "FALSE"
 [5,] "residual_sugar"      "1.020162e-52" "FALSE"
 [6,] "chlorides"           "1.179056e-55" "FALSE"
 [7,] "free_sulfur_dioxide" "7.694597e-31" "FALSE"
 [8,] "total_sulfur_dioxide" "3.573451e-34" "FALSE"
 [9,] "density"             "1.936053e-08" "FALSE"
[10,] "pH"                  "1.712237e-06" "FALSE"
[11,] "sulphates"           "5.823140e-38" "FALSE"
[12,] "alcohol"             "6.644057e-27" "FALSE"
[13,] "quality"             "9.515085e-36" "FALSE"
```

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 quantil-quantil 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)
}
```

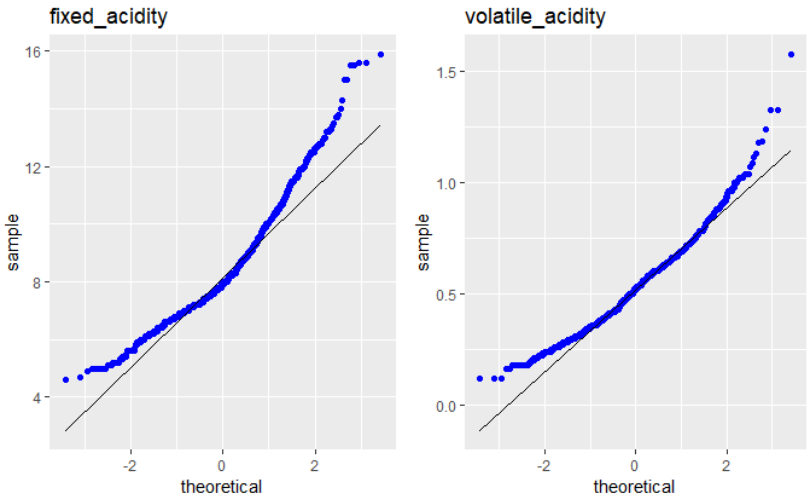

Fig. 10. Diagrames Q-Q per a `fixed_acidity` i `volatile_acidity`.

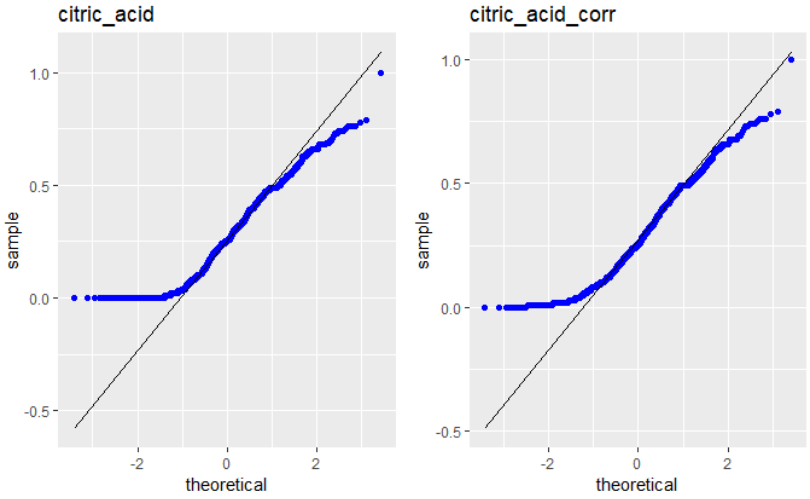

Fig. 11. Diagrames Q-Q per a `cítric_acid` i `cítric_acid_corr`.

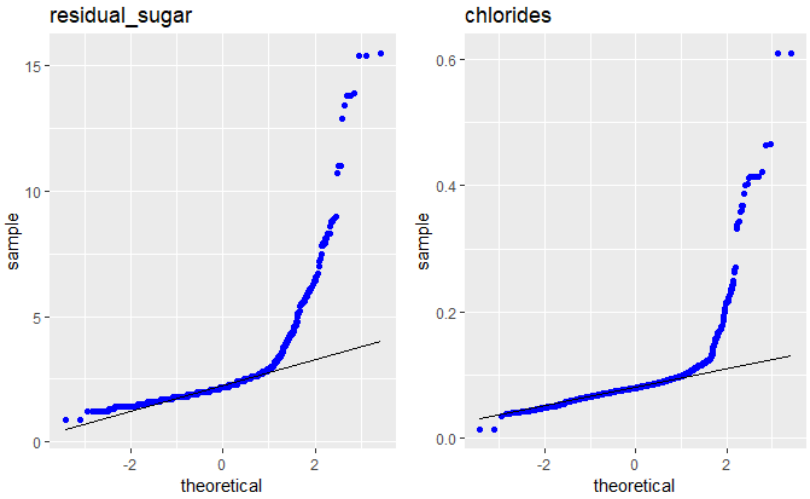

Fig. 12. Diagrames Q-Q per a `residual_sugar` i `chlorides`.

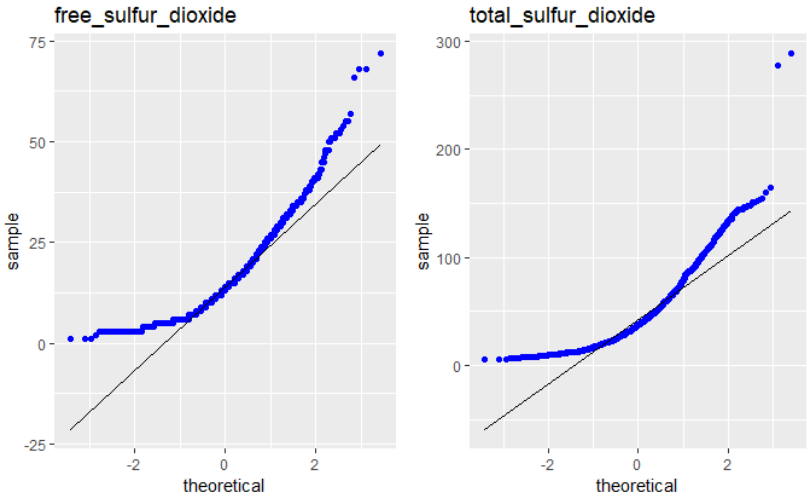

Fig. 13. Diagrames Q-Q per a `free_sulfur_dioxide` i `total_sulfur_dioxide`.

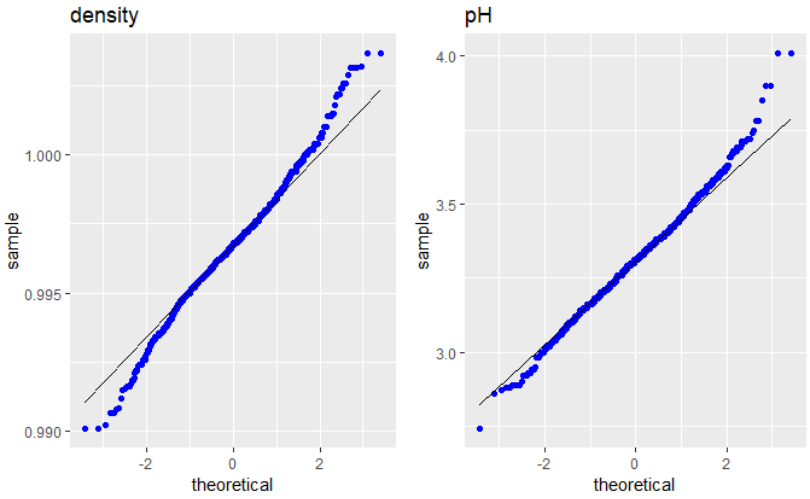

Fig. 14. Diagrames Q-Q per a `density` i `pH`.

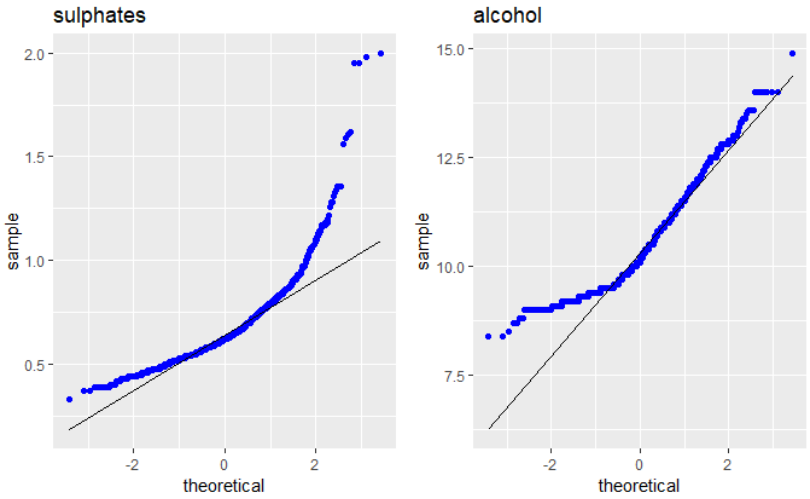

Fig. 15. Diagrames Q-Q per a `sulphates` i `alcohol`.

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 > alphanormalitat 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

alpha <- 0.05

df_aux <- df
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$p.value) %>% unlist(),
       variancies_iguals = (lapply(fkt,
                        function(x) x$p.value) %>% unlist()) > alpha) %>%
  t() %>% t() %>% print()
```

```
      parametre              p_value        variancies_iguals
 [1,] "fixed_acidity"        "4.785198e-05" "FALSE"
 [2,] "volatile_acidity"     "9.172531e-05" "FALSE"
 [3,] "citric_acid"          "8.965992e-01" " TRUE"
 [4,] "citric_acid_corr"     "9.483717e-01" " TRUE"
 [5,] "residual_sugar"       "6.210012e-02" " TRUE"
 [6,] "chlorides"            "4.878874e-01" " TRUE"
 [7,] "free_sulfur_dioxide"  "2.096175e-02" "FALSE"
 [8,] "total_sulfur_dioxide" "4.878458e-09" "FALSE"
 [9,] "density"              "3.615554e-05" "FALSE"
[10,] "pH"                   "7.704140e-01" " TRUE"
[11,] "sulphates"            "6.837163e-01" " TRUE"
[12,] "alcohol"              "8.011055e-02" " TRUE"
```

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. 4 a la Fig. 9 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 nivells alts d'encert els valors 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
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"       "2.796749e-05" "FALSE"
 [2,] "volatile_acidity"    "3.165494e-31" "FALSE"
 [3,] "citric_acid_corr"    "3.081771e-16" "FALSE"
 [4,] "residual_sugar"      "5.033308e-02" " TRUE"
 [5,] "chlorides"           "2.067627e-08" "FALSE"
 [6,] "free_sulfur_dioxide" "3.747866e-03" "FALSE"
 [7,] "total_sulfur_dioxide" "4.228165e-08" "FALSE"
 [8,] "density"             "2.775161e-07" "FALSE"
 [9,] "pH"                  "2.278993e-02" "FALSE"
[10,] "sulphates"           "9.945136e-20" "FALSE"
[11,] "alcohol"             "7.793167e-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, llevat de per als nivells residuals de sucre.

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

4.4. Test de Kruskall-Wallis

Del test de Welch sobre la igualtat de mitjanes entre dos grups, i el d'homogeneïtat de variàncies de Fligner-Killeen, aplicable a dos o més grups (s'ha aplicat, però, només a dos grups segons la classificació `quality_c3` ("No acceptable" i "Acceptable")), se'n desprèn que hi ha diferències significatives entre classes (de qualitat) de les dades. Fóra interessant tenir un test estadístic que refermés aquestes conclusions per a la categorització de qualitat `quality_c2` ("Dolenta", "Regular", "Bona"), més matisada que `quality_c3`. Això pot assolir-se amb el test de Kruskal-Wallis, que permet testejar si dos o més conjunts independents de dades (que poden tenir mides diferents, com en el nostre cas) provenen de la mateixa distribució (https://en.wikipedia.org/wiki/Kruskal%E2%80%93Wallis_one-way_analysis_of_variance). La hipòtesi nul·la és

H_0: No hi ha diferències significatives entre els conjunts o grups de dades

i l'alternativa és

H_a: Hi ha diferències significatives entre els conjunts o grups de dades.

```
# Test de Kruskal-Wallis de provinença de la mateixa distribució.
# Testeja si hi ha algun grup que en domina estocàsticament un altre

alpha <- 0.05

df_aux <- df
categ <- df_aux$quality_c2

# Apliquem el test a les variables físico-químiques
kwt <- df_aux %>% select(-(quality:quality_c3)) %>%
```

```
                  lapply(function(x) { kruskal.test(split(x,categ)) })

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

```
      parametre             p_value        grups_iguals
 [1,] "fixed_acidity"       "7.495141e-07" "FALSE"
 [2,] "volatile_acidity"    "2.653989e-37" "FALSE"
 [3,] "citric_acid"         "2.029460e-19" "FALSE"
 [4,] "citric_acid_corr"    "3.319494e-19" "FALSE"
 [5,] "residual_sugar"      "5.420483e-02" " TRUE"
 [6,] "chlorides"           "2.747248e-09" "FALSE"
 [7,] "free_sulfur_dioxide" "1.056971e-06" "FALSE"
 [8,] "total_sulfur_dioxide" "1.348635e-14" "FALSE"
 [9,] "density"             "7.286018e-09" "FALSE"
[10,] "pH"                  "3.491168e-05" "FALSE"
[11,] "sulphates"           "9.653449e-33" "FALSE"
[12,] "alcohol"             "1.313706e-51" "FALSE"
```

Llevat de per al nivell de sucre residual, cal concloure que, com que el valor *p* és (molt) menor que el nivell de significança triat (α= 0.05), que hi ha diferències estadístiques significatives entre alguns dels grups “Dolenta”, “Regular” i “Bona” per als diferents paràmetres.

4.5. 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
aux <- df %>% 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     -0.256130895       0.67257733
volatile_acidity        -0.25613089      1.000000000      -0.55183456
citric_acid_corr         0.67257733     -0.551834563       1.00000000
residual_sugar           0.11477672      0.001917882       0.14162767
chlorides                0.09370519      0.061297772       0.21176604
free_sulfur_dioxide     -0.15379419     -0.010503827      -0.06565463
total_sulfur_dioxide    -0.11318144      0.076470005       0.01612611
density                  0.66804729      0.022026232       0.36472008
pH                      -0.68297819      0.234937294      -0.53045919
sulphates                0.18300566     -0.260986685       0.31901714
alcohol                 -0.06166827     -0.202288027       0.12462896
quality                  0.12405165     -0.390557780       0.23411288
                      residual_sugar    chlorides free_sulfur_dioxide
fixed_acidity            0.114776724  0.093705186        -0.153794193
volatile_acidity         0.001917882  0.061297772        -0.010503827
citric_acid_corr         0.141627669  0.211766040        -0.065654635
residual_sugar           1.000000000  0.055609535         0.187048995
chlorides                0.055609535  1.000000000         0.005562147
free_sulfur_dioxide      0.187048995  0.005562147         1.000000000
```

```
total_sulfur_dioxide    0.203027882  0.047400468         0.667666450
density                 0.355283371  0.200632327        -0.021945831
pH                     -0.085652422 -0.265026131         0.070377499
sulphates               0.005527121  0.371260481         0.051657572
alcohol                 0.042075437 -0.221140545        -0.069408354
quality                 0.013731637 -0.128906560        -0.050656057
                     total_sulfur_dioxide     density          pH
fixed_acidity                -0.11318144  0.66804729 -0.68297819
volatile_acidity              0.07647000  0.02202623  0.23493729
citric_acid_corr              0.01612611  0.36472008 -0.53045919
residual_sugar                0.20302788  0.35528337 -0.08565242
chlorides                     0.04740047  0.20063233 -0.26502613
free_sulfur_dioxide           0.66766645 -0.02194583  0.07037750
total_sulfur_dioxide          1.00000000  0.07126948 -0.06649456
density                       0.07126948  1.00000000 -0.34169933
pH                           -0.06649456 -0.34169933  1.00000000
sulphates                     0.04294684  0.14850641 -0.19664760
alcohol                      -0.20565394 -0.49617977  0.20563251
quality                      -0.18510029 -0.17491923 -0.05773139
                        sulphates     alcohol     quality
fixed_acidity         0.183005664 -0.06166827  0.12405165
volatile_acidity     -0.260986685 -0.20228803 -0.39055778
citric_acid_corr      0.319017137  0.12462896  0.23411288
residual_sugar        0.005527121  0.04207544  0.01373164
chlorides             0.371260481 -0.22114054 -0.12890656
free_sulfur_dioxide   0.051657572 -0.06940835 -0.05065606
total_sulfur_dioxide  0.042946836 -0.20565394 -0.18510029
density               0.148506412 -0.49617977 -0.17491923
pH                   -0.196647602  0.20563251 -0.05773139
sulphates             1.000000000  0.09359475  0.25139708
alcohol               0.093594750  1.00000000  0.47616632
quality               0.251397079  0.47616632  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:

- `cor(fixed_acidity, cítric_acid_corr) = 0.67`, reflectint el fet que l'àcid cítric és un àcid no volàtil
- `cor(fixed_acidity, pH) = -0.68` i `cor(cítric_acid_corr, pH) = -0.53` perquè, a majors concentracions d'àcid, menor pH
- `cor(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.

4.6. 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 (numèrics) reals de qualitat (`quality_n1`, `quality_n2` o `quality_n3` en funció del nivell d'agregació triat per a la qualitat dels vins)). A continuació s'han predit els valors de qualitat numèrica 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` 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).

```
```{r}
# 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))
}

# Funció que retorna les prediccions numèriques i categòriques
# d'una regressió lineal
pred_reg_lineal <- function (df_aux_train,
                             df_aux_test,
                             variables,
                             objectiu_num,
                             objectiu_cat) {

    # Generem una fórmula per a la regressió lineal
    # a partir d'una cadena de caràcters construïda a patir
    # dels noms de les variables
    f_aux <- variables %>% paste(collapse = "+") %>%
             paste(objectiu_num, "~", .) %>%
             as.formula()

    # Generem un model de regressió lineal
    model_aux <- lm(f_aux, data = df_aux_train)

    # Fem la predicció dels valors de objectiu_num per a les mostres
    # de test
    objectiu_num_pred <- predict(model_aux, df_aux_test)
```
```

```
    # Transformem la predicció del model lineal en una
    # predicció categòrica/classificació (col·lapsem valors
    # massa petits o grans als valors extrems)
    aux <- arrodoneix_a_enter(objectiu_num_pred)
    objectiu_cat_pred <- ifelse(aux < min(df_aux_test[[objectiu_num]]),
                                min(df_aux_test[[objectiu_num]]),
                                ifelse(aux > max(df_aux_test[[objectiu_num]]),
                                       max(df_aux_test[[objectiu_num]]),
                                       aux)) %>%
                          sapply(function(x)  levels(df_aux_test[[objectiu_cat]])[x])
%>%
                          as.character()  %>%
                          factor(levels = levels(df_aux_test[[objectiu_cat]]),
                                 ordered = TRUE)

    # Retornem els valors rellevants
    return(list(model = model_aux,
                pred_num = objectiu_num_pred,
                pred_cat = objectiu_cat_pred))
}

# Funció que calcula l'score d'una classificació categòrica
precisio <- function(pred, real) {
    return(length(which(real == pred)) / length(real))
}

# Funció que calcula la precisió per tipus de mostra mitja
precisio_tmm <- function(pred, real) {
    return(tibble(valor = real,
                  pred = pred) %>%
           split(real) %>%
           lapply(function(x) {length(which(x$valor == x$pred)) /
                               length(x$valor) }) %>%
           do.call(c, .) %>%
           mean())
}

# Funció que imprimeix resultats d'una classificació categòrica
impr_resultats_class <- function(pred, real) {

    # Imprimim els valors de classificació obtinguts
    cat("Precisió : \n", precisio(pred, real))

    cat("\nPrecisió per tipus de mostra mitja :\n", precisio_tmm(pred, real))

    cat("\nTaula de contingència :\n")
    print(table(real, pred))

}

# Predicció de qualitat a partir d'una regressió lineal
reg_lineal <- function (dframe,
                        variables,
                        objectiu_num,
                        objectiu_cat,
                        prop_entrenament = 0.8,
                        random_state = 42) {

    # Generem un comjunt de test i un d'entrenament
```

```
    itts <- stratified_train_test_split(dframe[[objectiu_cat]],
                                        prop_entrenament,
                                        random_state)

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

    # Calculem les prediccions de regressió lineal
    prl <- pred_reg_lineal(df_aux_train,
                           df_aux_test,
                           variables,
                           objectiu_num,
                           objectiu_cat)


    # Imprimim les dades rellevants de la fase d'entrenament
    cat("\nFASE D'ENTRENAMENT\n")
    cat("Coeficients : \n")
    print(prl$model$coefficients)
    cat("Error absolut rms per mostra : \n",
        sqrt(sum(prl$model$residuals**2)/nrow(df_aux_train)))
    cat("\nCorrelació entre predicció i valors d'entrenament : \n",
        cor(df_aux_train[[objectiu_num]], prl$model$fitted.values))

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

    # Imprimim els valors de classificació obtinguts
    cat("\n\nCLASSIFICACIÓ DE LES MOSTRES DE TEST\n")
    impr_resultats_class(prl$pred_cat, df_aux_test[[objectiu_cat]])

}

cat("REGRESSIÓ LINEAL PER A quality_c1\n")
cat("---------------------------------\n")
reg_lineal(dfn,
           dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                   names(),
           "quality_n1",
           "quality_c1")


cat("\nREGRESSIÓ LINEAL PER A quality_c2\n")
cat("---------------------------------\n")
reg_lineal(dfn,
           dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                   names(),
           "quality_n2",
           "quality_c2")

cat("\nREGRESSIÓ LINEAL PER A quality_c2\n")
cat("---------------------------------\n")
reg_lineal(dfn,
           dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                   names(),
```

```
           "quality_n3",
           "quality_c3")
REGRESSIÓ LINEAL PER A quality_c1
---------------------------------

FASE D'ENTRENAMENT
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity     citric_acid_corr
          3.62803468           0.06557439          -0.19519695          -0.02544261
      residual_sugar            chlorides  free_sulfur_dioxide total_sulfur_dioxide
          0.02319482          -0.08464008           0.05151355          -0.13665911
             density                   pH            sulphates              alcohol
         -0.05365278          -0.03699814           0.14980268           0.26371164
Error absolut rms per mostra :
 0.6488139
Correlació entre predicció i valors d'entrenament :
 0.5933626

FASE DE TEST
Error absolut rms per mostra :
 0.6376379
Correlació entre predicció i valors de test :
 0.6221905

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió :
 0.5869565
Precisió per tipus de mostra mitja :
 0.2633212
Taula de contingència :
    pred
real   3   4   5   6   7   8
   3   0   0   2   0   0   0
   4   0   0  10   1   0   0
   5   0   0 100  37   0   0
   6   0   0  44  80   4   0
   7   0   0   0  31   9   0
   8   0   0   0   2   2   0


REGRESSIÓ LINEAL PER A quality_c2
---------------------------------

FASE D'ENTRENAMENT
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity     citric_acid_corr
          2.092419450          0.046051753         -0.081901231         -0.003384544
      residual_sugar            chlorides  free_sulfur_dioxide total_sulfur_dioxide
          0.018610484         -0.035428015          0.012829365         -0.010071598
             density                   pH            sulphates              alcohol
         -0.047205946         -0.017889538          0.057852587          0.094094466
Error absolut rms per mostra :
 0.3598746
Correlació entre predicció i valors d'entrenament :
 0.4649493

FASE DE TEST
Error absolut rms per mostra :
 0.3505717
Correlació entre predicció i valors de test :
```

```
 0.5248243

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió :
 0.8224299
Precisió per tipus de mostra mitja :
 0.3459596
Taula de contingència :
         pred
real      Dolenta Regular Bona
  Dolenta       0      13    0
  Regular       0     262    2
  Bona          0      42    2

REGRESSIÓ LINEAL PER A quality_c3
---------------------------------

FASE D'ENTRENAMENT
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity     citric_acid_corr
           1.13880546           0.07657589          -0.04141880           0.01107461
      residual_sugar            chlorides  free_sulfur_dioxide total_sulfur_dioxide
           0.03720582          -0.02456858          -0.00213920          -0.01761276
            density                   pH            sulphates              alcohol
          -0.08780975           0.01033631           0.05116438           0.07663043
Error absolut rms per mostra :
 0.2965753
Correlació entre predicció i valors d'entrenament :
 0.4985074

FASE DE TEST
Error absolut rms per mostra :
 0.3070339
Correlació entre predicció i valors de test :
 0.4548782

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió :
 0.8629283
Precisió per tipus de mostra mitja :
 0.5382343
Taula de contingència :
             pred
real          No adequada Adequada
  No adequada         273        4
  Adequada             40        4
```

Si s'analitzen els resultats per al cas de les dades de qualitat sense agregar (`quality_n1` / `quality_c1`), es pot constatar com 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. Aquest fet resulta en una capacitat de predicció o classificació del model força moderada, 58.6%. 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 massiva 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.

Per al cas de les dades amb nivells d'agregació de la qualitat majors, les prestacions del model milloren. Per a `quality_n2` / `quality_c2` ("Dolenta", "Regular", "Bona") i `quality_n3` / `quality_c3` ("No adequada", Adequada") es pot constatar com empitjora la correlació entre predicció i mesura, però millora la precisió de classificació categòrica. L'empitjorament de la correlació s'atribueix a la disminució de nivells, fet que fa que les prediccions numèriques tendeixin a estar més dispersades respecte dels valors discrets que haurien de prendre (una predicció numèrica de valor 5.1 abans estava a només 0.1 punts del valor discret 5, però ara n'estarà 0.4 del valor discret 5.5 (mitja entre 5 i 6), que podria encapsular numèricament el valor "Regular"). L'increment de la precisió també és esperable: a mesura que encapsulem valors de qualitat en unitats majors, els errors que abans es produïen en predir valors enters encapsulats a la mateixa unitat de qualitat superior ara queden emmascarats. Per a la classificació ternària de `quality_c2` s'aconsegueixen valors de precisió acceptablement bons (82.2%), tot i que estan subjectes a la mateixa problemàtica que per al cas de `quality_c1`: el predictor/classificador està molt escorat cap a la predicció correcta de la categoria "Regular", que és molt predominant en relació amb la resta.

Val a dir que s'han realitzat també les anàlisis anteriors emprant com a variable `citric_acid` en comptes de `citric_acid_corr`, amb diferències negligibles. Per tant, a partir d'ara es prendrà com a variable per a la resta d'anàlisis `citric_acid_corr`.

4.7. Refinament del procés pre-processament: reducció de la numerositat 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

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_red <- df_aux %>% dplyr::filter(es_reduida)

# Imprimim la nova distribució de qualitats
```

```
dfn_red %>% select(quality_c1) %>% table() %>% print().
```

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

Si ara es torna calcular els resultats de les regressions lineals anteriors per a aquest nou data set:

```
cat("REGRESSIÓ LINEAL PER A quality_c1\n")
cat("--------------------------------\n")
reg_lineal(dfn_red,
           dfn_red %>% select(-citric_acid, -(quality:quality_c3)) %>%
                   names(),
           "quality_n1",
           "quality_c1")

cat("\nREGRESSIÓ LINEAL PER A quality_c2\n")
cat("--------------------------------\n")
reg_lineal(dfn_red,
           dfn_red %>% select(-citric_acid, -(quality:quality_c3)) %>%
                   names(),
           "quality_n2",
           "quality_c2")# Predicció de qualitat a partir d'una regressió lineal

cat("\nREGRESSIÓ LINEAL PER A quality_c3\n")
cat("--------------------------------\n")
reg_lineal(dfn_red,
           dfn_red %>% select(-citric_acid, -(quality:quality_c3)) %>%
                   names(),
           "quality_n3",
           "quality_c3")
```

```
REGRESSIÓ LINEAL PER A quality_c1
--------------------------------

FASE D'ENTRENAMENT
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity     citric_acid_corr 
         3.721251243          0.016666743         -0.337245692         -0.080101790 
      residual_sugar            chlorides  free_sulfur_dioxide total_sulfur_dioxide 
         0.009162186         -0.261646384          0.084427012         -0.137822168 
             density                   pH            sulphates              alcohol 
         0.025625534         -0.239048716          0.267562940          0.438156322 
Error absolut rms per mostra :
 0.8083869 
Correlació entre predicció i valors d'entrenament :
 0.7208551 

FASE DE TEST
Error absolut rms per mostra :
 0.7270245 
Correlació entre predicció i valors de test :
 0.787649 

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió :
 0.5742574 
Precisió per tipus de mostra mitja :
 0.3693182 
Taula de contingència :
    pred
real  3  4  5  6  7  8
   3  0  1  1  0  0  0
```

```
   4  1  2  7  1  0  0
   5  0  0 17  5  0  0
   6  0  1  5 14  2  0
   7  0  0  1 14 25  0
   8  0  0  0  0  4  0

REGRESSIÓ LINEAL PER A quality_c2
---------------------------------

FASE D'ENTRENAMENT
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity     citric_acid_corr
           2.17525139           0.04155986          -0.17251474          -0.06409165
      residual_sugar            chlorides  free_sulfur_dioxide total_sulfur_dioxide
          -0.01821485          -0.12930673           0.04159839          -0.01728066
             density                   pH            sulphates              alcohol
           0.04690434          -0.13072060           0.14673932           0.28692937
Error absolut rms per mostra :
 0.4982546
Correlació entre predicció i valors d'entrenament :
 0.6817682

FASE DE TEST
Error absolut rms per mostra :
 0.4953164
Correlació entre predicció i valors de test :
 0.693613

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió :
 0.6831683
Precisió per tipus de mostra mitja :
 0.5949883
Taula de contingència :
         pred
real      Dolenta Regular Bona
  Dolenta       4       9    0
  Regular       0      33   11
  Bona          0      12   32


REGRESSIÓ LINEAL PER A quality_c3
---------------------------------

FASE D'ENTRENAMENT
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity     citric_acid_corr
           1.31665021           0.11792110          -0.06771782          -0.01269595
      residual_sugar            chlorides  free_sulfur_dioxide total_sulfur_dioxide
           0.03216890          -0.09489092          -0.00614578          -0.04876181
             density                   pH            sulphates              alcohol
          -0.06480972          -0.02620919           0.14900888           0.15400453
Error absolut rms per mostra :
 0.3583238
Correlació entre predicció i valors d'entrenament :
 0.6907906

FASE DE TEST
Error absolut rms per mostra :
 0.3665649
```

```
Correlació entre predicció i valors de test :
 0.6741969

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió :
 0.7920792
Precisió per tipus de mostra mitja :
 0.7924641
Taula de contingència :
             pred
real          No adequada Adequada
  No adequada          45       12
  Adequada              9       35
```

Com es pot observar, la precisió ha disminuït (lleugerament per a `quality_c1` i `quality_c3`) i de manera més notable per a `quality_c2`), però ha augmentat la capacitat del model de predir valors que no són els més freqüents. 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" (o 5 i 6), que eren aquelles que el classificador entrenat amb el data set de l'apartat anterior classificava millor per ser les més majoritàries: la precisió per tipus de mostra mitja ha incrementat en tots els casos. Ara, per exemple, per al cas de `quality_c3`, el model aconsegueix predir correctament la majoria de valors "Adequada", cosa que no passava abans.

4.8. Refinament del procés pre-processament: augment de la numerositat de les dades

Com a alternativa al procés anterior, en què s'han eliminat dades (i per tant informació potencialment rellevant del data set), es podria pensar en incrementar la quantitat de dades generant dades sintètiques per a les classes menys representades (ja sigui segons `quality_c1`, `quality_c2` o `quality_c3`). Això pot fer-se, per exemple, afegint dades sintètiques per a les classes menys representades. Aquestes dades sintètiques seran còpies de les reals existents per a la classe, però amb un soroll gaussià afegit de desviació estàndard una fracció `prop_sd_soroll` de la desviació estàndard de les dades originals de la classe.

L'algorisme que s'ha implementat completa les classes menys nombroses de dades afegint-hi mostres de la manera que s'acaba de comentar fins que la classes arribin a tenir una proporció `prop_min_files` de registres respecte de la classe més nombrosa del data set.

Només s'ha realitzat el procés d'increment de la numerositat per a les classes menys nombroses per al conjunt d'entrenament. D'aquesta manera no es contamina la capacitat de comparació dels resultats amb altres mètodes: les dades sobre les que s'aplica el model a la fase de test són dades reals (no sintètiques).

Tal com s'ha fet la programació dels algorismes, la divisió de les dades dels data sets en dades d'entrenament i de test es fa a l'interior de les funcions de càlcul. Per tant, l'augment de la numerositat mitjançant dades sintètiques per al conjunt de test no s'ha fet en un data set independent sinó que s'ha implementat com una funció, `equilibra_numerositat()`, que opera sobre una versió modificada de la funció `reg_lineal()` anterior, `reg_lineal_bis()`.

```
# Funció que equilibra la numerositat d'un data set
equilibra_numerositat <- function(df_aux_train,
                                  objectiu_cat,
                                  prop_min_files = 0.25,
                                  prop_sd_soroll = 0.33) {
```

```
    # Calculem el nombre mínim de registres/files que trindrà cada
    # classe com a proporció del nombre de registres que té la
    # classe més nombrosa
    aux <- table(df_aux_train[[objectiu_cat]]) %>% as.vector()
        min_files <- as.integer(prop_min_files*max(aux))

    # SPLIT
    aux1 <- df_aux_train %>% split(df_aux_train[[objectiu_cat]])

    # APPLY
    aux2 <- aux1 %>% lapply(function(dfaux) {
        if(nrow(dfaux)>min_files) {
            # Si la classe té prou files, la retornem
            # sense modificar
            return(dfaux)
        }

        # Generem dues "parts"" del data frame amb les files addicionals
        # reptetint les existents
        filaux <- rep(1:nrow(dfaux),
                      length.out = min_files - nrow(dfaux))

        dfaux2 <- dfaux[filaux,] %>% select(-(quality:quality_c3))
        dfaux3 <- dfaux[filaux,] %>% select(quality:quality_c3)

        # A cada columna de paràmetres fisico-químics afegim
        # soroll a les files afegides
        dfaux2 <- dfaux2 %>% lapply(function(colaux) {

            if(length(colaux)<5) {
                # Si hi ha pocs elements (sd(colaux) pot ser estranya)
                # els retornem sense modificar
                return(colaux)
            } else {
                # Sinó, hi afegim soroll gaussià de sd = sd(colaux)*prop_sd_soroll
                return (colaux + rnorm(length(colaux),
                                       0, sd(colaux)*prop_sd_soroll))
            }

        }) %>% as_tibble()

        # Unim les columnes amb dades numèriques amb les columnes amb dades
        # categòriques i qualitatives del data frame amb les dades sintètiques
        # per a la classe que estem modificant
        dfaux4 <- bind_cols(dfaux2, dfaux3)

        # Afegim les files sintètiques a les originals de la classe i retornem
        # el valor
        return(bind_rows(dfaux, dfaux4))

    })

    # COMBINE
    aux3 <- do.call(bind_rows, aux2)

    # Retornem el data frame amb la numerositat augmentada
    return(aux3)
}
```

```
# Predicció de qualitat a partir d'una regressió lineal
reg_lineal_bis <- function (dframe,
                            variables,
                            objectiu_num,
                            objectiu_cat,
                            prop_entrenament = 0.8,
                            random_state = 42,
                            prop_min_files = 0.25,
                            prop_sd_soroll = 0.33) {

    # Generem un comjunt de test i un d'entrenament
    itts <- stratified_train_test_split(dframe[[objectiu_cat]],
                                        prop_entrenament,
                                        random_state)

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

    # Equilibrem la numerositat de les dades d'entrenament
    df_aux_train <- equilibra_numerositat(df_aux_train,
                                          objectiu_cat,
                                          prop_min_files = prop_min_files,
                                          prop_sd_soroll = prop_sd_soroll)

    # Calculem les prediccions de regressió lineal
    prl <- pred_reg_lineal(df_aux_train,
                           df_aux_test,
                           variables,
                           objectiu_num,
                           objectiu_cat)

    # Imprimim les dades rellevants de la fase d'entrenament
    cat("\n\nFASE D'ENTRENAMENT\n")
    cat("Coeficients : \n")
    print(prl$model$coefficients)
    cat("Error absolut rms per mostra : \n",
        sqrt(sum(prl$model$residuals**2)/nrow(df_aux_train)))
    cat("\nCorrelació entre predicció i valors d'entrenament : \n",
        cor(df_aux_train[[objectiu_num]], prl$model$fitted.values))

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

    # Imprimim els valors de classificació obtinguts
    cat("\n\nCLASSIFICACIÓ DE LES MOSTRES DE TEST\n")
    impr_resultats_class(prl$pred_cat, df_aux_test[[objectiu_cat]])

}

cat("REGRESSIÓ LINEAL PER A quality_c1\n")
cat("---------------------------------\n")
reg_lineal_bis(dfn,
           dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                   names(),
           "quality_n1",
```

```
           "quality_c1",
           prop_min_files = 0.50,
           prop_sd_soroll = 0.33)

cat("\nREGRESSIÓ LINEAL PER A quality_c2\n")
cat("---------------------------------\n")
reg_lineal_bis(dfn,
           dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                   names(),
           "quality_n2",
           "quality_c2",
           prop_min_files = 0.50,
           prop_sd_soroll = 0.33)

cat("\nREGRESSIÓ LINEAL PER A quality_c3\n")
cat("---------------------------------\n")
reg_lineal_bis(dfn,
           dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                   names(),
           "quality_n3",
           "quality_c3",
           prop_min_files = 0.50,
           prop_sd_soroll = 0.33)
```

```
REGRESSIÓ LINEAL PER A quality_c1
---------------------------------

FASE D'ENTRENAMENT
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity     citric_acid_corr
          3.522478660         -0.018690913         -0.390779994         -0.020304288
      residual_sugar            chlorides  free_sulfur_dioxide total_sulfur_dioxide
        -0.090858202         -0.251052117         -0.045288047          0.006884528
              density                   pH            sulphates              alcohol
        -0.125729015         -0.299465517          0.228431560          0.554501799
Error absolut rms per mostra :
 0.9737501
Correlació entre predicció i valors d'entrenament :
 0.764483

FASE DE TEST
Error absolut rms per mostra :
 0.8273406
Correlació entre predicció i valors de test :
 0.6173542

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió :
 0.4658385
Precisió per tipus de mostra mitja :
 0.3169457
Taula de contingència :
    pred
real  3  4  5  6  7  8
   3  0  2  0  0  0  0
   4  1  2  7  1  0  0
   5  3 29 77 24  4  0
   6  1  7 52 49 17  2
   7  0  0  2 12 21  5
   8  0  0  0  0  3  1
```

```
REGRESSIÓ LINEAL PER A quality_c2
---------------------------------

FASE D'ENTRENAMENT
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity     citric_acid_corr
         1.995304245          0.034131703         -0.200369499         -0.044630922
      residual_sugar            chlorides  free_sulfur_dioxide total_sulfur_dioxide
        -0.049255650         -0.078244358          0.068245580         -0.001102022
             density                   pH            sulphates              alcohol
         0.009506849         -0.100158513          0.086410135          0.240409929
Error absolut rms per mostra :
 0.5572199
Correlació entre predicció i valors d'entrenament :
 0.6154002

FASE DE TEST
Error absolut rms per mostra :
 0.4223107
Correlació entre predicció i valors de test :
 0.5054493

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió :
 0.7788162
Precisió per tipus de mostra mitja :
 0.6541375
Taula de contingència :
         pred
real      Dolenta Regular Bona
  Dolenta      10       3    0
  Regular      26     225   13
  Bona          0      29   15

REGRESSIÓ LINEAL PER A quality_c3
---------------------------------

FASE D'ENTRENAMENT
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity     citric_acid_corr
        1.2629766060         0.0827854891        -0.0732885672         0.0137182143
      residual_sugar            chlorides  free_sulfur_dioxide total_sulfur_dioxide
        0.0472768381        -0.0434515629         0.0022047864        -0.0417491453
             density                   pH            sulphates              alcohol
       -0.0844067952        -0.0001346961         0.0852820307         0.1407177941
Error absolut rms per mostra :
 0.3699652
Correlació entre predicció i valors d'entrenament :
 0.6195804

FASE DE TEST
Error absolut rms per mostra :
 0.3551262
Correlació entre predicció i valors de test :
 0.4670232

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió :
 0.8286604
Precisió per tipus de mostra mitja :
```

```
 0.7955776
Taula de contingència :
             pred
real          No adequada Adequada
  No adequada         233        44
  Adequada             11        33
```

Com es pot constatar comparant aquests resultats amb els de l'apartat anterior (reducció de la numerositat), hi ha millores en la precisió quan es classifica segons (`quality_c2`) però lleugeres disminucions de la precisió en els altres dos casos. Com que no s'ha produït una millora de la qualitat notable en tots els casos respecte de la reducció de la numerositat, s'abandonarà aquesta opció perquè es basa en dades sintètiques i és, per essència, menys recomanable que les anteriors.

4.9. Anàlisi de la dimensionalitat del data set per a classificacions basades en regressions lineals

Com a darrer estudi realitzat sobre la capacitat predictiva de models basats en regressions lineals s'ha realitzat un estudi d'influència de la reducció de la dimensionalitat de les dades. A tal efecte s'ha implementat un algorisme de validació creuada sobre les dades de test a partir d'una funció que genera els índex dels diversos *folds* mantenint la seva estratificació:

```
# Divisió dels nombres de fila d'un data frame en k folds estratificats
# en funció de la variable objectiu_cat
stratified_k_folds <- function(k, dframe, objectiu_cat) {

    # Dividim els nombres de fila en funció d'objectiu_cat
    aux1 <- 1:nrow(dframe) %>% split(dframe[[objectiu_cat]])

    # Assignem un nombre de fold a cada element de cada divisió anterior
    aux2 <- aux1 %>% lapply(function(x) { cut(1:length(x), k,
                                             labels = FALSE,
                                             include.lowest = TRUE,
                                             right = TRUE) })

    # Reconstruim la sortida desitjada, seleccionant els elements de cada fold
    # de la llista aux1
    sortida <- list()

    for(i in 1:k) {
        acum <- c()
        for(n in names(aux1)) {
            acum <- c(acum, aux1[[n]][aux2[[n]] == i])
        }

    sortida[[i]] <- acum

    }

return(sortida)

}

# Comprovem que funciona...
#
# aux <- stratified_k_folds(10, dfn, "quality_c1")
#
# for(i in 1:10) {
#     table(dfn[l[[i]],"quality_c1"]) %>% print()
# }
```

A continuació s'ha programat un esquema de validació creuada sobre les dades d'entrenament (amb 5 folds), i les prestacions de la millor opció trobada s'han avaluat amb les dades de test.

L'algorisme emprat per a avaluar les possibilitats de reducció de la dimensionalitat quant a prediccions basades en regressions lineals es basa en anar eliminant variables de la regressió lineal mentre s'aconsegueixi una millora en la precisió de la classificació. A cada iteració s'elimina la variable amb què s'aconsegueix més millora de precisió. Una vegada eliminada una variable, es torna a començar el procés amb les restants, fins que en una iteració no s'ha aconseguit cap millora.

```
reg_lineal_sel_param <- function(dframe,
                                 variables,
                                 objectiu_num,
                                 objectiu_cat,
                                 prop_entrenament = 0.8,
                                 k = 5,
                                 random_state = 42) {

    # Generem un comjunt de test i un d'entrenament
    itts <- stratified_train_test_split(dframe[[objectiu_cat]],
                                        prop_entrenament,
                                        random_state)

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


    # Calculem les precisons de referència per a les dades
    # segons un esquema de validació creuada amb les dades
    # d'entrenament
    folds <- stratified_k_folds(k, df_aux_train, objectiu_cat)

    # Acumuladors per a les precisions
    p_0 <- c()
    p_tmm0 <- c()
    for(i in 1:k) {

        # Seleccionen el fold de test i prenen els altres com d'entrenament
        trn <- df_aux_train %>%
               dplyr::filter(!is.element(1:nrow(df_aux_train), folds[[i]]))
        tst <- df_aux_train %>%
               dplyr::filter(is.element(1:nrow(df_aux_train), folds[[i]]))

        # Calculem els paràemtres de la regressió lineal i acumulem les precisions
        rlin <- pred_reg_lineal(trn,
                                tst,
                                variables,
                                objectiu_num,
                                objectiu_cat)
        p_0 <- c(p_0, precisio(rlin$pred_cat, tst[[objectiu_cat]]))
        p_tmm0 <- c(p_tmm0, precisio_tmm(rlin$pred_cat, tst[[objectiu_cat]]))
    }

    # Calculem els valors mitjos
    p_0 <- mean(p_0)
    p_tmm0 <- mean(p_tmm0)

    # Emprant els mateixos folds, implementem un esquema de selecció de
    # variables. Eliminem una variable cada vegada, i ens quedem amb el resultat
    # millor
```

```
p_max <- p_0
p_t_mmmax <- p_tmm0
variables_max <- variables

millorat <- TRUE

while ((length(variables) >= 2) && millorat) {

    millorat <- FALSE

    for(v in variables) {

        variables_0 <- variables[!(variables == v)]
        p_0 <- c()
        p_tmm0 <- c()

        for(i in 1:k) {

            # Seleccionen el fold de test i prenen els altres com d'entrenament
            trn <- df_aux_train %>%
                   dplyr::filter(!is.element(1:nrow(df_aux_train), folds[[i]]))
            tst <- df_aux_train %>%
                   dplyr::filter(is.element(1:nrow(df_aux_train), folds[[i]]))

            # Calculem els paràemtres de la regressió lineal i acumulem les
            # precisions
            rlin <- pred_reg_lineal(trn,
                                    tst,
                                    variables_0,
                                    objectiu_num,
                                    objectiu_cat)
            p_0 <- c(p_0, precisio(rlin$pred_cat, tst[[objectiu_cat]]))
            p_tmm0 <- c(p_tmm0, precisio_tmm(rlin$pred_cat,
                                             tst[[objectiu_cat]]))
        }

        # Calculem els valors mitjos
        p_0 <- mean(p_0)
        p_tmm0 <- mean(p_tmm0)

        if(p_0 > p_max) {
            p_max <- p_0
            p_t_mmmax <- p_tmm0
            variables_max <- variables_0
            millorat <- TRUE
        }
    }
variables <- variables_max
}

# Una vegada tenim el conjunt de variables òptim, implementem la regressió
# lineal amb totes les dades d'entrenament i de test

cat("El conjunt òptim de variables és : \n")
print(variables_max)
cat("La precisió mitja aconseguida és : ", p_max,"\n")

# Calculem les prediccions de regressió lineal
prl <- pred_reg_lineal(df_aux_train,
                       df_aux_test,
```

```
                          variables_max,
                          objectiu_num,
                          objectiu_cat)

    # Imprimim les dades rellevants de la fase d'entrenament
    cat("\nFASE D'ENTRENAMENT\n")
    cat("Coeficients : \n")
    print(prl$model$coefficients)
    cat("Error absolut rms per mostra : \n",
        sqrt(sum(prl$model$residuals**2)/nrow(df_aux_train)))
    cat("\nCorrelació entre predicció i valors d'entrenament : \n",
        cor(df_aux_train[[objectiu_num]], prl$model$fitted.values))

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

    # Imprimim els valors de classificació obtinguts
    cat("\n\nCLASSIFICACIÓ DE LES MOSTRES DE TEST\n")
    impr_resultats_class(prl$pred_cat, df_aux_test[[objectiu_cat]])

}
cat("REGRESSIÓ LINEAL PER A quality_c1\n")
cat("---------------------------------\n")
reg_lineal_sel_param(dfn,
                     dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                             names(),
                     "quality_n1",
                     "quality_c1",
                     prop_entrenament = 0.8,
                     k = 5,
                     random_state = 42)

cat("\nREGRESSIÓ LINEAL PER A quality_c2\n")
cat("---------------------------------\n")
reg_lineal_sel_param(dfn,
                     dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                             names(),
                     "quality_n2",
                     "quality_c2",
                     prop_entrenament = 0.8,
                     k = 5,
                     random_state = 42)

cat("\nREGRESSIÓ LINEAL PER A quality_c3\n")
cat("---------------------------------\n")
reg_lineal_sel_param(dfn,
                     dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                             names(),
                     "quality_n3",
                     "quality_c3",
                     prop_entrenament = 0.8,
                     k = 5,
                     random_state = 42)
```

```
REGRESSIÓ LINEAL PER A quality_c1
---------------------------------
```

```
El conjunt òptim de variables és :
[1] "fixed_acidity"        "volatile_acidity"     "chlorides"
[4] "free_sulfur_dioxide"  "total_sulfur_dioxide" "density"
[7] "sulphates"            "alcohol"
La precisió mitja aconseguida és :  0.6000009

FASE D'ENTRENAMENT
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity            chlorides
           3.62766474           0.08120904          -0.18785717          -0.08184976
 free_sulfur_dioxide total_sulfur_dioxide              density            sulphates
           0.05311500          -0.13216898          -0.05905183            0.14873747
              alcohol
           0.25597172
Error absolut rms per mostra :
 0.6497194
Correlació entre predicció i valors d'entrenament :
 0.5918356

FASE DE TEST
Error absolut rms per mostra :
 0.6426532
Correlació entre predicció i valors de test :
 0.6140908

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió :
 0.5838509
Precisió per tipus de mostra mitja :
 0.2620191
Taula de contingència :
    pred
real   3   4   5   6   7   8
   3   0   0   2   0   0   0
   4   0   0  10   1   0   0
   5   0   0 100  37   0   0
   6   0   0  45  79   4   0
   7   0   0   1  30   9   0
   8   0   0   0   2   2   0


REGRESSIÓ LINEAL PER A quality_c2
---------------------------------
El conjunt òptim de variables és :
[1] "fixed_acidity"      "volatile_acidity"   "citric_acid_corr"   "residual_sugar"
[5] "free_sulfur_dioxide" "density"             "pH"                 "sulphates"
La precisió mitja aconseguida és :  0.8247335

FASE D'ENTRENAMENT
Coeficients :
        (Intercept)       fixed_acidity    volatile_acidity    citric_acid_corr
          2.092511512         0.162739750        -0.089316753        -0.005799213
     residual_sugar free_sulfur_dioxide             density                  pH
          0.056531887         0.002774871        -0.167892500         0.055021223
            sulphates
          0.062635926
Error absolut rms per mostra :
 0.3655837
Correlació entre predicció i valors d'entrenament :
 0.4371628
```

```
FASE DE TEST
Error absolut rms per mostra :
 0.36505
Correlació entre predicció i valors de test :
 0.4584236

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió :
 0.8255452
Precisió per tipus de mostra mitja :
 0.3472222
Taula de contingència :
         pred
real      Dolenta Regular Bona
  Dolenta       0      13    0
  Regular       0     263    1
  Bona          0      42    2


REGRESSIÓ LINEAL PER A quality_c3
---------------------------------
El conjunt òptim de variables és :
 [1] "fixed_acidity"        "volatile_acidity"     "citric_acid_corr"
 [4] "residual_sugar"       "chlorides"            "free_sulfur_dioxide"
 [7] "total_sulfur_dioxide" "density"              "sulphates"
[10] "alcohol"
La precisió mitja aconseguida és :  0.8740288

FASE D'ENTRENAMENT
Coeficients :
         (Intercept)        fixed_acidity     volatile_acidity     citric_acid_corr
         1.138774576          0.065100042         -0.041105547          0.010904650
      residual_sugar            chlorides  free_sulfur_dioxide total_sulfur_dioxide
         0.034673653         -0.026273592         -0.001052116         -0.019231586
             density            sulphates              alcohol
        -0.079548738          0.050274155          0.081800249
Error absolut rms per mostra :
 0.2966291
Correlació entre predicció i valors d'entrenament :
 0.4982342

FASE DE TEST
Error absolut rms per mostra :
 0.3066803
Correlació entre predicció i valors de test :
 0.4566371

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió :
 0.8629283
Precisió per tipus de mostra mitja :
 0.5382343
Taula de contingència :
             pred
real          No adequada Adequada
  No adequada         273        4
  Adequada             40        4
```

Com es pot constatar, les possibilitats de reducció de la dimensionalitat no són gaire elevades. On es podria eliminar més variables sense afectar (de fet, millorant fins i tot) la precisió dels models és en el cas de classificació directa a partir de les notes de qualitat, en què podrien eliminar-se tres variables, `cítric_acid_corr`, `residual_sugar` i `pH`. Aquestes variables, no obstant això, semblen tenir influència en les altres classificacions. Per tant, no queda clar, de l'anàlisi feta, si realment es pot operar una reducció de la dimensionalitat basada en l'eliminació de variables.

Val a dir que el mètode emprat d'anàlisi de la dimensionalitat no és òptim perquè no assegura que no hi hagi combinacions no visitades que puguin donar resultats més favorables. Potser si s'hagués fet una cerca aleatòria de combinacions de paràmetres, o l'hagués implementat un algorisme de cerca basat en afegir variables en comptes de treure'n, els resultats haguessin estat més clarificadors.

Si es torna a repetir l'experiment amb les dades del data set de dimensionalitat reduïda, s'obtenen conjunts de variables no coincidents amb els anteriors. Curiosament s'obté una reducció de la dimensionalitat molt bona per al criteri `quality_c2`:

```
cat("\nREGRESSIÓ LINEAL PER A quality_c2\n")
cat("---------------------------------\n")
reg_lineal_sel_param(dfn_red,
                     dfn_red %>% select(-citric_acid, -(quality:quality_c3)) %>%
                             names(),
                     "quality_n2",
                     "quality_c2",
                     prop_entrenament = 0.8,
                     k = 5,
                     random_state = 42)
```

```
REGRESSIÓ LINEAL PER A quality_c2
---------------------------------
El conjunt òptim de variables és :
[1] "volatile_acidity" "chlorides"        "pH"               "sulphates"
[5] "alcohol"
La precisió mitja aconseguida és :  0.7142819

FASE D'ENTRENAMENT
Coeficients :
     (Intercept) volatile_acidity        chlorides               pH        sulphates
       2.1747428       -0.1518446       -0.1452275       -0.1347802        0.1597419
         alcohol
       0.2399664
Error absolut rms per mostra :
 0.502207
Correlació entre predicció i valors d'entrenament :
 0.6754876

FASE DE TEST
Error absolut rms per mostra :
 0.4867194
Correlació entre predicció i valors de test :
 0.7046101

CLASSIFICACIÓ DE LES MOSTRES DE TEST
Precisió :
 0.7029703
Precisió per tipus de mostra mitja :
 0.6101399
Taula de contingència :
         pred
```

```
real      Dolenta Regular Bona
  Dolenta       4       9    0
  Regular       0      35    9
  Bona          0      12   32
```

Tot i això, la discrepància amb els altres valors obtinguts (al cap i a la fi representen el mateix data set) fa creure que aquesta reducció és espúria i que no representa la variabillitat del data set.

Per tant, vistos els resultats, es continuarà treballant amb totes les dimensions disponibles.

4.10. 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 ara kNN cal emprar dades normalitzades per tal de no esbiaixar la distància euclidià entre mostres. S'ha pres un valor de *k* = 20 per a tots els classificadors kNN.

```
# Predicció de qualitat a partir d'un classificador kNN

class_kNN <- function (dframe,
                       variables,
                       objectiu_cat,
                       k = 20,
                       prop_entrenament = 0.8,
                       random_state = 42) {

    # Generem un comjunt de test i un d'entrenament
    itts <- stratified_train_test_split(dframe[[objectiu_cat]],
                                        prop_entrenament,
                                        random_state)

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

    # Predicció
    quality_pred_knn <- knn(df_aux_train %>% select(variables),
                            df_aux_test %>% select(variables),
                            df_aux_train[[objectiu_cat]],
                            k = k) %>%
                            as.character() %>%
                            factor(levels = levels(df_aux_test[[objectiu_cat]]),
                                                   ordered = TRUE)

    # Imprimim els valors de classificació obtinguts
    impr_resultats_class(quality_pred_knn, df_aux_test[[objectiu_cat]])

}

cat("CLASSIFICACIÓ KNN PER A quality_c1\n")
cat("---------------------------------\n")
class_kNN(dfn,
          dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                  names(),
          "quality_c1",
          k = 20,
          prop_entrenament = 0.8,
          random_state = 42)
```

```
cat("\nCLASSIFICACIÓ KNN PER A quality_c2\n")
cat("---------------------------------\n")
class_kNN(dfn,
          dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                  names(),
          "quality_c2",
          k = 20,
          prop_entrenament = 0.8,
          random_state = 42)

cat("\nCLASSIFICACIÓ KNN PER A quality_c2\n")
cat("---------------------------------\n")
class_kNN(dfn,
          dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                  names(),
          "quality_c3",
          k = 20,
          prop_entrenament = 0.8,
          random_state = 42)
```

```
CLASSIFICACIÓ KNN PER A quality_c1
---------------------------------
Precisió :
 0.5652174
Precisió per tipus de mostra mitja :
 0.2628003
Taula de contingència :
    pred
real   3   4   5   6   7   8
   3   0   0   1   1   0   0
   4   0   0   9   2   0   0
   5   0   1 100  33   3   0
   6   0   0  47  70  11   0
   7   0   0   3  25  12   0
   8   0   0   0   2   2   0

CLASSIFICACIÓ KNN PER A quality_c2
---------------------------------
Precisió :
 0.8286604
Precisió per tipus de mostra mitja :
 0.3989899
Taula de contingència :
         pred
real      Dolenta Regular Bona
  Dolenta       0      13    0
  Regular       0     256    8
  Bona          0      34   10

CLASSIFICACIÓ KNN PER A quality_c3
---------------------------------
Precisió :
 0.8598131
Precisió per tipus de mostra mitja :
 0.6320151
Taula de contingència :
             pred
real          No adequada Adequada
  No adequada         262       15
  Adequada             30       14
```

Com es pot constatar, els resultats són molt similars als obtinguts classificant a partir d'una regressió lineal. Si es vol millorar la capacitat de predicció dels models, caldrà anar cap a models més complicats (i menys explicatius de la realitat).

En cas d'emprar el data set amb numerositat reduïda `dfn_red`, les conclusions foren anàlogues a les ja extretes per al cas de regressions lineals: hi ha una tendència a classificar millor els valors de qualitat menys nombrosos a costa d'una reducció de la precisió, pels mateixos motius que ja s'han comentat abans.

4.11. Predicció de qualitat a partir d'un classificador *Random Forest*

A la vista dels resultats de la classificació amb un classificador kNN, s'ha optat per a emprar classificadors molt més sofisticats, com ara un classificador *Random Forest*, que empra una gran quantitat d'arbres de classificació (100 en el nostre cas) en un esquema de *bagging* per a votar la classe de les mostres.

```
# Predicció de qualitat a partir d'un classificador Random Forest

class_RF <- function (dframe,
                      variables,
                      objectiu_cat,
                      prop_entrenament = 0.8,
                      random_state = 42) {

    # Generem un comjunt de test i un d'entrenament
    itts <- stratified_train_test_split(dframe[[objectiu_cat]],
                                        prop_entrenament,
                                        random_state)

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

    # Generem un classificador Random Forest
    set.seed(random_state)
    rfc <- df_aux_train %>%
           select(variables) %>%
           randomForest(df_aux_train[[objectiu_cat]],
                        n_tree = 100,
                        replace = TRUE)

    # Generem la predicció amb el conjunt de test
    objectiu_cat_pred <- df_aux_test %>%
                         select(variables) %>%
                         predict(rfc, .) %>%
                         as.character() %>%
                         factor(levels = levels(df_aux_test[[objectiu_cat]]),
                                ordered = TRUE)

    # Imprimim els valors de classificació obtinguts
    impr_resultats_class(objectiu_cat_pred, df_aux_test[[objectiu_cat]])

}

cat("CLASSIFICACIÓ RANDOM FOREST PER A quality_c1\n")
cat("-------------------------------------------\n")
class_RF(dfn,
         dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                 names(),
```

```
         "quality_c1")

cat("\nCLASSIFICACIÓ RANDOM FOREST PER A quality_c2\n")
cat("-------------------------------------------\n")
class_RF(dfn,
         dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                 names(),
         "quality_c2")

cat("\nCLASSIFICACIÓ RANDOM FOREST PER A quality_c3\n")
cat("-------------------------------------------\n")
class_RF(dfn,
         dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                 names(),
         "quality_c3")
```

```
CLASSIFICACIÓ RANDOM FOREST PER A quality_c1
-------------------------------------------
Precisió :
 0.689441
Precisió per tipus de mostra mitja :
 0.3489165
Taula de contingència :
    pred
real   3   4   5   6   7   8
   3   0   0   1   1   0   0
   4   0   0   9   2   0   0
   5   2   0 104  30   1   0
   6   0   0  30  94   4   0
   7   0   0   1  14  24   1
   8   0   0   0   2   2   0

CLASSIFICACIÓ RANDOM FOREST PER A quality_c2
-------------------------------------------
Precisió :
 0.8785047
Precisió per tipus de mostra mitja :
 0.525641
Taula de contingència :
         pred
real      Dolenta Regular Bona
  Dolenta       1      12    0
  Regular       0     258    6
  Bona          0      21   23

CLASSIFICACIÓ RANDOM FOREST PER A quality_c3
-------------------------------------------
Precisió :
 0.9096573
Precisió per tipus de mostra mitja :
 0.7469232
Taula de contingència :
             pred
real          No adequada Adequada
  No adequada         269        8
  Adequada             21       23
```

Com es pot observar, en aquest cas la millora és evident: la precisió ha augmentat dràsticament en tots els casos, i per a `quality_c2` i `quality_c3` a valors que es podrien començar a considerar

acceptables. També ha millora en general, i molt, la precisió per tipus de mostra mitja, fet que palesa la major capacitat d'aquest algorisme per a classificar bé fins i tot membres de classes minoritàries.

Cal recalcar que, si es mira la classificació segons quality_c1, l'algorisme és capaç de classificar la pràctica totalitat de les mostres amb un error enter de classificació de, com a molt, ±1, fet que demostra que hi ha uns relació evident entre les dades físico-químiques dels vins analitzats i la seva qualitat percebuda.

Si ara es prova la classificació al conjunt de dades de numerositat reduïda:

```
cat("CLASSIFICACIÓ RANDOM FOREST PER A quality_c1\n")
cat("-------------------------------------------\n")
class_RF(dfn_red,
         dfn_red %>% select(-citric_acid, -(quality:quality_c3)) %>%
                 names(),
         "quality_c1")

cat("\nCLASSIFICACIÓ RANDOM FOREST PER A quality_c2\n")
cat("-------------------------------------------\n")
class_RF(dfn_red,
         dfn_red %>% select(-citric_acid, -(quality:quality_c3)) %>%
                 names(),
         "quality_c2")

cat("\nCLASSIFICACIÓ RANDOM FOREST PER A quality_c3\n")
cat("-------------------------------------------\n")
class_RF(dfn_red,
         dfn_red %>% select(-citric_acid, -(quality:quality_c3)) %>%
                 names(),
         "quality_c3")
```

```
CLASSIFICACIÓ RANDOM FOREST PER A quality_c1
-------------------------------------------
Precisió :
 0.5940594
Precisió per tipus de mostra mitja :
 0.3545455
Taula de contingència :
    pred
real  3  4  5  6  7  8
   3  0  1  0  1  0  0
   4  0  3  5  2  1  0
   5  0  2 15  3  2  0
   6  0  0  7  6  9  0
   7  0  0  0  3 36  1
   8  0  0  0  0  4  0

CLASSIFICACIÓ RANDOM FOREST PER A quality_c2
-------------------------------------------
Precisió :
 0.7425743
Precisió per tipus de mostra mitja :
 0.6585082
Taula de contingència :
         pred
real      Dolenta Regular Bona
  Dolenta       5       7    1
  Regular       2      31   11
  Bona          1       4   39
```

```
CLASSIFICACIÓ RANDOM FOREST PER A quality_c3
-----------------------------------------
Precisió :
 0.8712871
Precisió per tipus de mostra mitja :
 0.8652313
Taula de contingència :
             pred
real          No adequada Adequada
  No adequada          52         5
  Adequada              8        36
```

De manera coherent amb el que ja s'ha constatat en apartats anteriors, la capacitat de predicció continua sent acceptable en general (tot i que lleugerament pitjor), però es classifiquen millor els elements de clsses minoritàries.

4.12. Predicció de qualitat a partir d'un classificador C5.0 amb *boosting*

Com a darrer experiment, s'ha implementat un classificador C5.0 amb *boosting*. Un altre cop cal complicar el model (en aquest cas amb tècniques de *boosting*) per a mirar d'obtenir bons resultats:

```
# Predicció de qualitat a partir d'un classificador C5.0

class_C5.0 <- function(dframe,
                       variables,
                       objectiu_cat,
                       prop_entrenament = 0.8,
                       random_state = 42) {

    # Generem un comjunt de test i un d'entrenament
    itts <- stratified_train_test_split(dframe[[objectiu_cat]],
                                        prop_entrenament,
                                        random_state)

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

    # Generem un classificador C5.0
    set.seed(random_state)
    c5.0c <- df_aux_train %>%
             select(variables) %>%
             C5.0(df_aux_train[[objectiu_cat]],
             trials = 10,
             rules = FALSE)

    # Generem la predicció amb el conjunt de test
    objectiu_cat_pred <- df_aux_test %>%
                         select(variables) %>%
                         predict(c5.0c, .) %>%
                         as.character() %>%
                         factor(levels = levels(df_aux_test[[objectiu_cat]]),
                                ordered = TRUE)

    # Imprimim els valors de classificació obtinguts
    impr_resultats_class(objectiu_cat_pred, df_aux_test[[objectiu_cat]])

}

cat("CLASSIFICACIÓ C5.0 PER A quality_c1\n")
cat("-----------------------------------\n")
```

```
class_C5.0(dfn,
           dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                  names(),
           "quality_c1")

cat("\nCLASSIFICACIÓ C5.0 PER A quality_c2\n")
cat("-----------------------------------\n")
class_C5.0(dfn,
           dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                   names(),
           "quality_c2")

cat("\nCLASSIFICACIÓ C5.0 PER A quality_c3\n")
cat("-----------------------------------\n")

class_C5.0(dfn,
           dfn %>% select(-citric_acid, -(quality:quality_c3)) %>%
                   names(),
           "quality_c3")
```

```
CLASSIFICACIÓ C5.0 PER A quality_c1
-----------------------------------
Precisió :
 0.6552795
Precisió per tipus de mostra mitja :
 0.3373726
Taula de contingència :
    pred
real   3   4   5   6   7   8
   3   0   0   0   2   0   0
   4   1   0   8   2   0   0
   5   2   2 105  26   2   0
   6   0   0  39  81   8   0
   7   0   0   2  13  25   0
   8   0   0   0   2   2   0

CLASSIFICACIÓ C5.0 PER A quality_c2
-----------------------------------
Precisió :
 0.8785047
Precisió per tipus de mostra mitja :
 0.5075758
Taula de contingència :
         pred
real      Dolenta Regular Bona
  Dolenta       0      13    0
  Regular       0     258    6
  Bona          0      20   24

CLASSIFICACIÓ C5.0 PER A quality_c3
-----------------------------------
Precisió :
 0.8816199
Precisió per tipus de mostra mitja :
 0.7497949
Taula de contingència :
             pred
real          No adequada Adequada
  No adequada         258       19
  Adequada             19       25
```

En aquest cas els resultats són bons, però lleugerament pitjors que per a la classificació emprant un classificador Random Forest.

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 categòriques 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,
- identificació (i possible processament) d'*outliers*,
- reducció de la numerositat per a les classes més nombroses 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,
- ampliació de numerositat de les classes menys nombroses a partir de dades sintètiques, i
- reducció de la dimensionalitat (per a models basats en regressions lineals).

S'ha decidit no eliminar ni modificar cap valor extrem del conjunt de dades, en no tenir argument fefaents per a discernir si es tractava de valors legítims o de valors que realment no pertanyien a l'univers de les dades que es volien analitzar (*outliers*). Cal recalcar que els resultats obtinguts han estat molt similars, i de vegades millors, als obtinguts en versions prèvies d'aquest treball, en què sí es van eliminar aquests valors extrems. Quant al tractament dels zeros (entesos com a falsos nuls) que potser apareixien a les concentracions d'àcid cítric, se n'ha fet una correcció basada en regressions lineals. Els valors corregits continuaven tenint valors baixos, amb què el fet de treballar amb dades originals o corregides no afectava quasibé les anàlisis posteriors.

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. Com a fruit d'aquest procés iteratiu, s'ha investigat una tècnica d'augment de la numerositat per a les classes menys representades basada en l'agregació de còpies sorolloses dels registres de la classe. Els resultats no han estat en general millors que els de reducció de la numerositat, i s'ha desentimat la tècnica.

S'ha realitzat un estudi estadístic de les dades destinat a conèixer-ne les seves característiques. S'ha fet un èmfasi especial en mirar de discernir si hi havia diferències notables entre els paràmetres estadísitics bàsics en funció de la classe a què pertanyien les mostres.

Quant a les tècniques d'anàlisi emprades, s'han emprat tècniques de regressió/classificació basades en regressions lineals, i tècniques de classificació basades en classificadors kNN, *Random Forest* i C5.0 amb *boosting*. De manera general 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).

Pel que fa a models basats en regressions lineals, que eren atractius per la seva senzillesa d'interpretació, s'ha pogut constatar que tenien una capacitat predictiva relativament baixa, probablement a causa de les assumpcions de linealitat, que són molt restrictives. S'ha intentat, per a aquests models, operar una reducció de la dimensionalitat de les dades perquè podia ser molt informativa (donada la facilitat d'interpretació dels models) sobre quins paràmetres realment determinaven la qualitat subjectiva del vins analitzats. No s'han obtingut resultats concloents, potser a causa de l'algorisme emprat.

Quant als models basats en classificadors senzills com el kNN, s'ha constatat que cal emprar classificadors sofisticats per a obtenir bones capacitats predictives. Malauradament aquests darrers classificadors són força complicats: no poden interpretar-se fàcilment i cal considerar-los com a un model de caixa negra. Per tant, quan s'ha pogut predir amb una precisió acceptable la qualitat subjectiva del vi a partir dels sesu paràmetres físico-químics, no s'ha estat capaç d'avaluar la influència de cada paràmetre individual en la fixació de la qualitat del vi.

De manera general, i com a conclusió de l'anàlisi feta, es pot afirmar que els paràmetres fisico-químics mesurats dels vins analitzats en determinen de manera força aproximada la qualitat subjectiva predita: analitzant les prediccions per als diversos models presentats i per als nivells de qualitat originals (valorats amb un enter de 3 a 8), es pot constatar que algorismes complexos com el *Random Forest* són capaços de classificar la pràctica totalitat de les mostres de test amb un error de classificació 0 o ±1.

Per a millorar les capacitats predictives del model caldria, potser, implantar classificadors encara més complexos, com ara classificadors basats en *stacking*. També es podria mirar d'operar una reducció de la dimensionalitat de les dades amb estratègies similars a les intentades per al cas de regressions lineals, però basada ara en models de classificació..

6. Codi

Tot el codi executat es troba, en format R Markdown, al fitxer `Practica_02_v03.Rmd`. El codi presentat als apartats anteriors correspon als diversos *chunks* de codi `R` que es troben al fitxer.

7. Referències

1. https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009
2. 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
3. https://en.wikipedia.org/wiki/Acids_in_wine
4. https://www.researchgate.net/publication/311919082_The_Classification_of_White_Wine_and_Red_Wine_According_to_Their_Physicochemical_Qualities
5. http://waterhouse.ucdavis.edu/whats-in-wine
6. https://www.awri.com.au/wp-content/uploads/2018/08/s1530.pdf
7. http://adlib.everysite.co.uk/resources/000/264/146/euwineregs.pdf
8. J. Ha, M. Kamber i J. Pei, *Data Mining. Concepts and Techniques*, 3a ed., Elsevier
9. A. C. Rencher, W. F. Christensen, *Methods of Multivariate Analysis*, 3a ed. Wiley