

Identificació de whiskies de contraban

Projecte d'Anàlisi de Dades - GCED

Fernando Gastón & Marc Gállego

Contents

1	Introducció	3
2	Descripció del conjunt de dades	3
3	Anàlisi	4
3.1	Anàlisi exploratori	4
3.2	Anàlisi discriminants	7
3.3	Resultats i conclusions	13
4	Bibliografia	13
5	Annex	15

1 Introducció

El principal objectiu d'aquest projecte és el de discriminar mescles de whisky de contraban d'entre un conjunt de whiskies de destil·leries escoceses.

Per a fer-ho disposem d'un dataset sobre la concentració de diferents químics en la composició dels whikies analitzats. És per això, que un objectiu alternatiu del projecte és determinar quins dels químics analitzats són més útils a l'hora de classificar els whiskies de contraban.

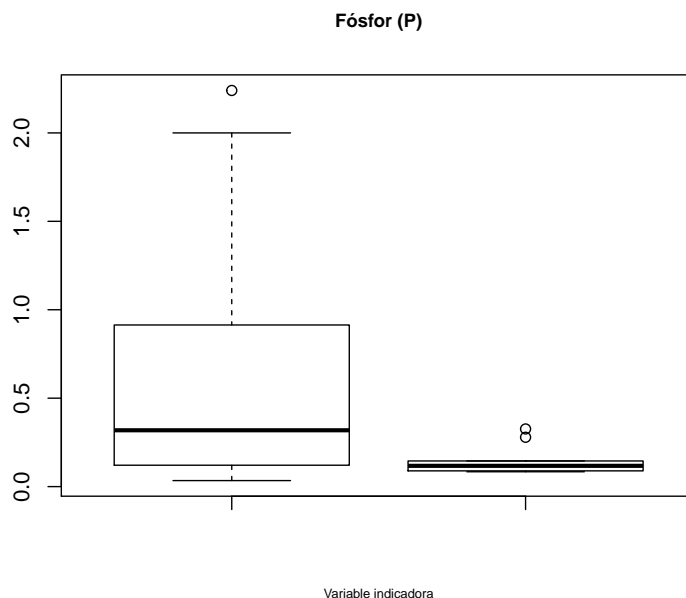
2 Descripció del conjunt de dades

Les dades utilitzades provenen d'un estudi realitzat per investigadors de The James Hutton Institute al Regne Unit i estan continguts al fitxer “**whiskey_chemo.csv**”. Aquest conté informació de la composició química de whiskies provinents de 32 destil·leries escoceses de les quals 5 són de contraban.

El fet que el nostre conjunt de dades estigui tan descompensat serà un fet que condicionarà molt el projecte i que, en general, no és gaire desitjable. Només tenim 5 observacions de whiskies de contraban, un nombre més petit i tot que el de variables.

A més, per cada destil·leria tenim un identificador numèric (**brandID**), el tipus de whisky (**type**), una variable indicadora respecte al contraban (**counterfeit**, 1 és contraban) i la concentració d'11 elements químics (P, S, Cl, K, Ca, Mn, Fe, Cu, Zn, Br, Rb).

Per cadascun d'aquests elements també disposem una variable indicadora (**P1-Rb1**) que indica si la concentració obtinguda per l'element és inferior al límit de detecció de la maquinària utilitzada. Això vol dir que concentracions baixes no són determinades adequadament (a efectes pràctics són 0). Això pot generar problemes en l'anàlisi de dades, per exemple, si es fan servir logaritmes. Tot i això, en el nostre cas són molt poques les dades per sota del límit de detecció i, a més, ja han estat imputades amb valors positius. Per exemple, per les dades del fòsfor (P) no s'observa aquest límit de detecció ja que s'han imputat les dades, sinó hauria de constar 0 o <0.0001 (o semblant):



Així doncs, no les tindrem en compte per a dur a terme els anàlisis.

El dataset resultant és:

counterfeit	P	S	Cl	K	Ca	Mn	Fe	Cu	Zn	Br	Rb
0	0.152	1.100	0.173	7.86	1.450	0.032	0.027	0.186	0.015	0.002	0.006
0	0.653	1.580	0.238	4.93	1.400	0.019	0.110	0.242	0.021	0.005	0.003
0	0.375	0.809	0.193	4.31	1.220	0.019	0.044	0.196	0.007	0.003	0.002
0	0.121	1.160	0.157	3.20	1.140	0.011	0.050	0.189	0.018	0.003	0.003
0	0.326	1.090	0.180	5.48	0.526	0.018	0.103	0.286	0.020	0.002	0.002
0	0.145	0.615	0.097	2.74	0.416	0.009	0.050	0.208	0.007	0.002	0.001

Respecte al tractament de dades previ a l'anàlisi, possibles dades faltants, no ens n'hem de preocupar ja que no n'hi ha.

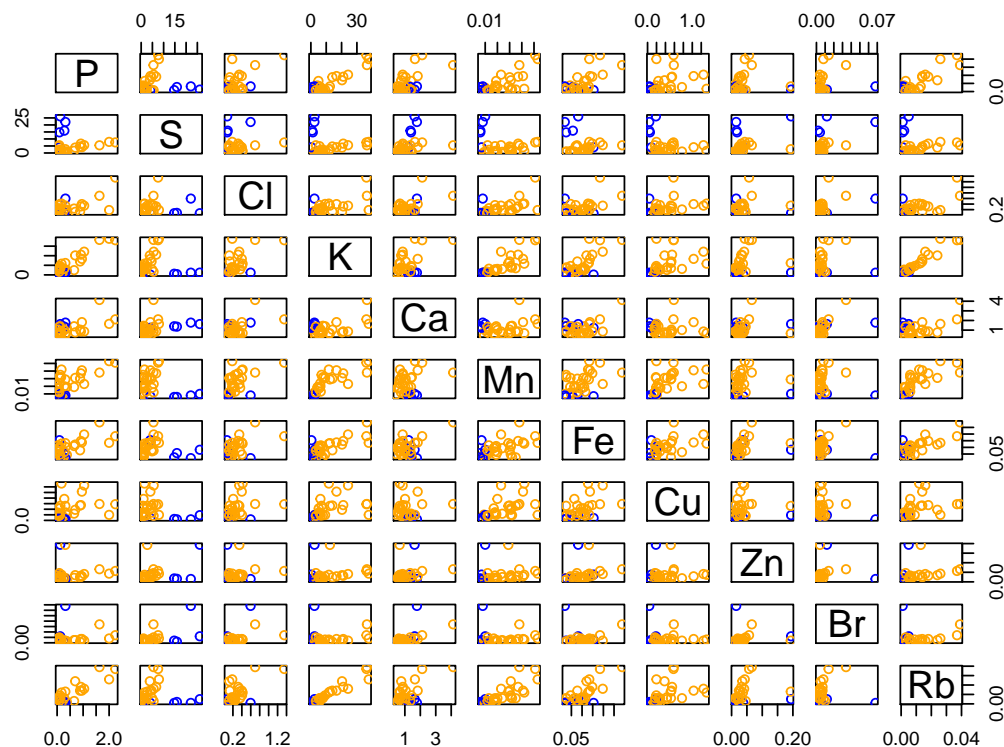
3 Anàlisi

En aquest apartat aplicarem els mètodes d'anàlisi de dades estudiats a l'assignatura. S'intercalerà part del codi utilitzat (la resta serà a l'annex) amb les explicacions pertinents sobre els mètodes, la seva aplicació i la interpretació dels resultats obtinguts.

3.1 Anàlisi exploratori

Com l'objectiu d'aquest treball es separar les dades de contraban, el primer que farem serà visualitzar i explorar les dades per veure si amb la informació disponible podem fer-ho.

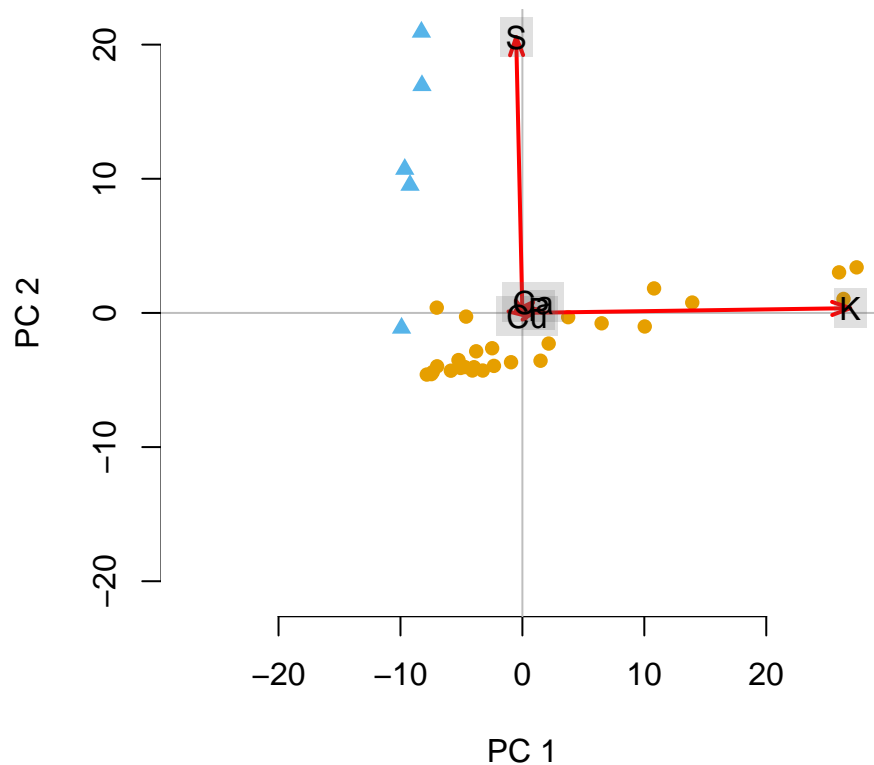
El primer que farem serà un simple scatter plot de tots els parells de variables (amb la funció **pairs**) per veure si hi observem una separació natural dels grups (en taronja el contraban):



Tot i que no s'aprecia gaire bé el detall (per la gran quantitat de plots) sí que podem observar que a tots els plots on intervé la variable **S** s'observa una bona separació dels grups, la qual cosa indica que probablement ens serà útil a l'hora de classificar les dades de contraban.

Per tal de fer una visualització més acurada de les dades farem servir la tècnica del Anàlisi de Components Principals (PCA) per tenir una representació de les dades en un espai de menys dimensions (2 o 3 dimensions) on es representi una gran variabilitat de les dades.

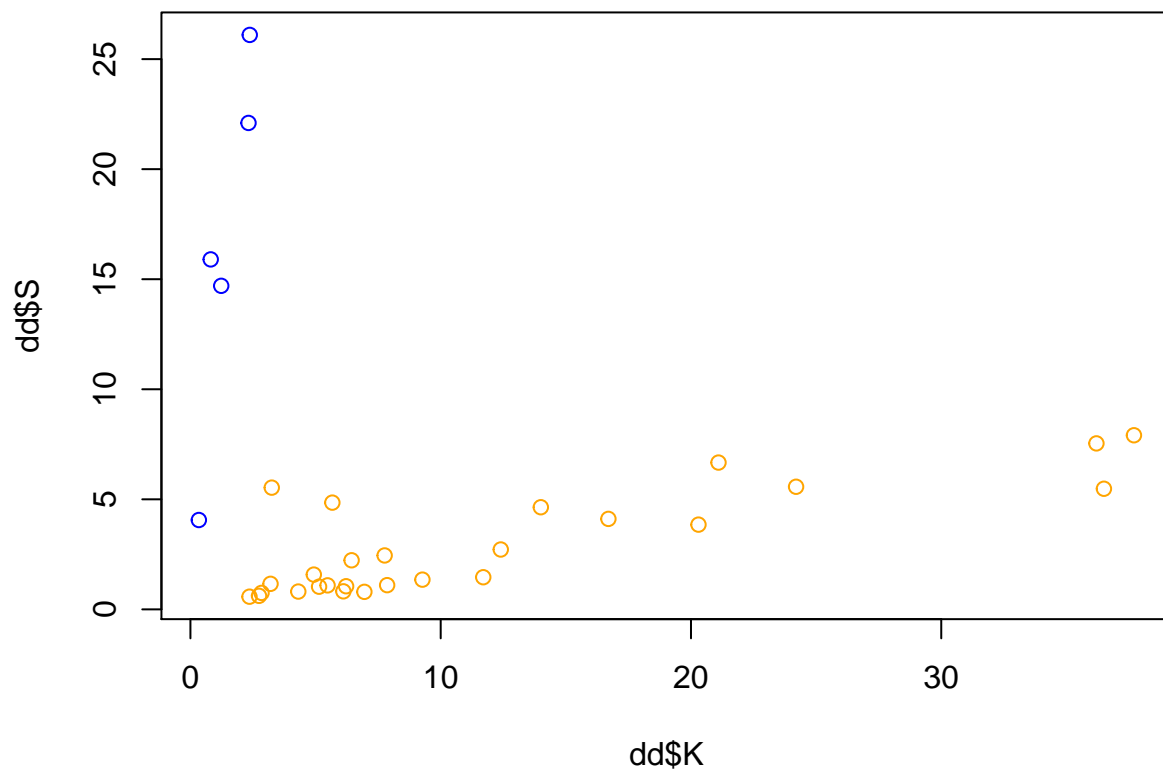
Aquesta tècnica permet reduir la dimensionalitat de les dades generant unes noves variables (els components principals) que són incorrelats i estan ordenats en funció del percentatge de la variància de les dades originals que expliquen. D'aquesta forma podem obtenir en poques dimensions una part important de la variabilitat observada en les dades. El biplot resultant de l'anàlisi de components principals es:



De les fletxes del biplot podem veure que els dos primers components principals són formats, quasi exclusivament, per una única variable, K i S respectivament:

	Comp.1	Comp.2
S	-0.0189307	0.9989165
K	0.9978127	0.0171457

Com ja podíem deduir del biplot, els coeficients per aquestes variables en els dos primers components principals són molt propers a 1. Per tant, aquest biplot és pràcticament igual al scatter plot entre les variables **S** i **K** en la matriu de plots realitzada amb **pairs**:



Cadascun dels components principals explica el següent percentatge de la variància total:

	% Var explicada
Comp.1	73.8612700
Comp.2	25.8420064
Comp.3	0.1969896
Comp.4	0.0515567
Comp.5	0.0286814
Comp.6	0.0176984
Comp.7	0.0011483
Comp.8	0.0005837
Comp.9	0.0000420
Comp.10	0.0000207
Comp.11	0.0000028

Per tant, amb aquests dos primers components principals (els que hem graficats al biplot) queda representat el 99.7% de la variabilitat de les dades.

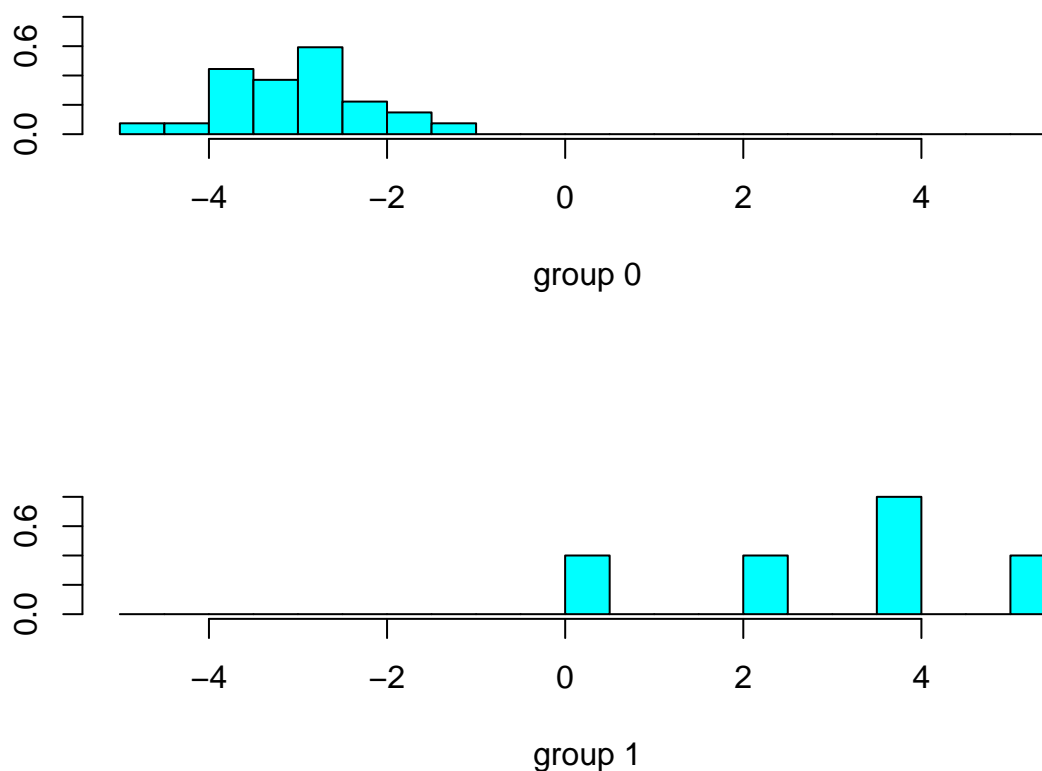
Una altra tècnica que ens pot servir per veure si existeixen diferències

Així doncs, a partir del biplot, veiem que les dades són, aparentment, fàcilment separables i que podrem classificar els whiskies fent servir les dades donades.

3.2 Anàlisi discriminants

Per fer la classificació farem servir els dos mètodes d'anàlisi discriminant estudiats (`lda`, `qda`).

Primer, estudiarem el cas del discriminant lineal o LDA. Aquest classificador es basa en assumir normalitat multivariant per les dades de cada grup, assumint que per cada grup la matriu de covariàncies és igual. Una observació serà, doncs, classificada al grup pel qual maximitzi la funció de distribució. Això dóna lloc a fronteres de decisió lineals, vet aquí el nom de la tècnica. Vegem primer l'error de classificació que es cometria en aplicar LDA a les nostres dades:



En aquest plot podem observar un histograma del discriminant lineal per cada grup de dades (contraban / no contraban). Veiem que no hi ha solapament i, doncs, no hi ha errors de classificació a les dades d'entrenament. Per comprovar això últim generarem la matriu de confusió:

	0	1
0	27	0
1	0	5

La matriu de confusió és diagonal. Això vol dir que les prediccions del LDA es corresponen amb els grups mètode reporta un error de training del 0%.

Com hem estandarditzat les dades abans d'aplicar `lda`, podem interpretar els coeficients del discriminant lineal associats a cadascuna de les variables com la importància que tenen cadascuna d'aquestes en la discriminació. És a dir, com més alt sigui el coeficient associat a una variable, major serà l'efecte d'aquesta en el discriminant

lineal i, per tant, en la classificació. En el nostre lda els coeficients per cada variable són:

	LD1
P	0.2518360
S	2.3785201
Cl	-0.0839157
K	-2.0685275
Ca	0.1806522
Mn	0.2683782
Fe	1.0037762
Cu	-0.3327140
Zn	-0.4711936
Br	-0.2464910
Rb	0.1767580

Veiem que les variables que més influeixen en l'anàlisi discriminant són les variables S i K (curiosament coincideixen amb els dos primers components principals) i la que menys el clor Cl.

Tot i això, aquest error de training no és una bona mesura per determinar la *performance* de la regla de classificació i hem de fer servir un conjunt de dades de test per estimar l'error de predicció en dades futures. Com nosaltres no disposem de conjunt de test farem servir el mètode *leave-one-out crossvalidation* que ens permetrà fer una estimació d'aquest error.

```
sum=0
i = 0
for (i in 1:n){
  lda.mod<-lda(counterfeit~.,data=dd[-i,])
  err<-predict(lda.mod, newdata=dd[i,])$class==dd[i,"counterfeit"]
  sum=sum+err
}
(te.acc.lda<-sum/n)

[1] 0.9375
```

L'error de test estimat resultant és del 6.45%. Com era d'esperar, aquest error és major que el del primer model, ja que s'està estimant l'error en unes dades sobre les quals el model no ha estat entrenat.

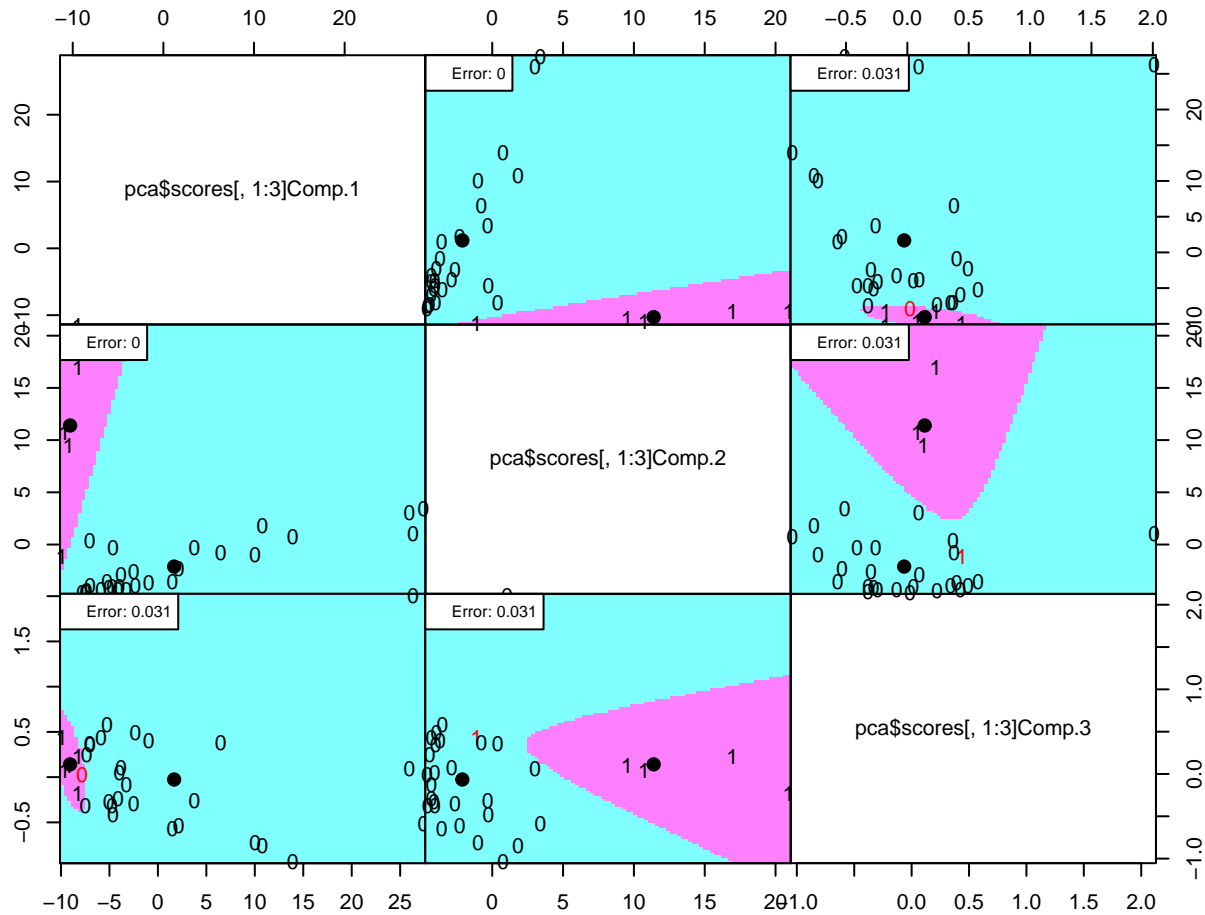
A continuació, analitzarem la regla de classificació del qda. La diferència fonamental respecte del LDA és que en el QDA no s'assumeix que les matrius de covariàncies pels diferents grups siguin iguals. Farem el mateix procediment que hem utilitzat per estudiar LDA: primer aplicarem el classificador sobre totes les dades i, després, realitzarem el LOOCV per estimar-ne l'error de test.

Quan intentem dur a terme el QDA, ens topem amb el següent error: `some group is too small for 'qda'`. Això es deu al fet que la implementació del QDA requereix tenir més observacions de cada grup que nombre de variables predictores:

```
#Fragment de codi en la implementació de la funció qda
g <- as.factor(grouping)
lev <- levels(g)
counts <- as.vector(table(g))
names(counts) <- lev
if(any(counts < p+1)) stop("some group is too small for 'qda'")
```

En el nostre cas disposem d'11 variables predictores i tan sols de 5 observacions de la classe `counterfeit`. Per a solventar aquest problema, hauríem d'usar les 4 primeres components principals com a entrada per a fer la QDA. Tot i això, quan duem a terme el LOOCV, en algunes iteracions les dades amb què entrenarem

el QDA només tindran 4 observacions de contraban (una d'elles serà utilitzada com a observació a predir). Per tant, haurem de fer servir únicament 3 components.



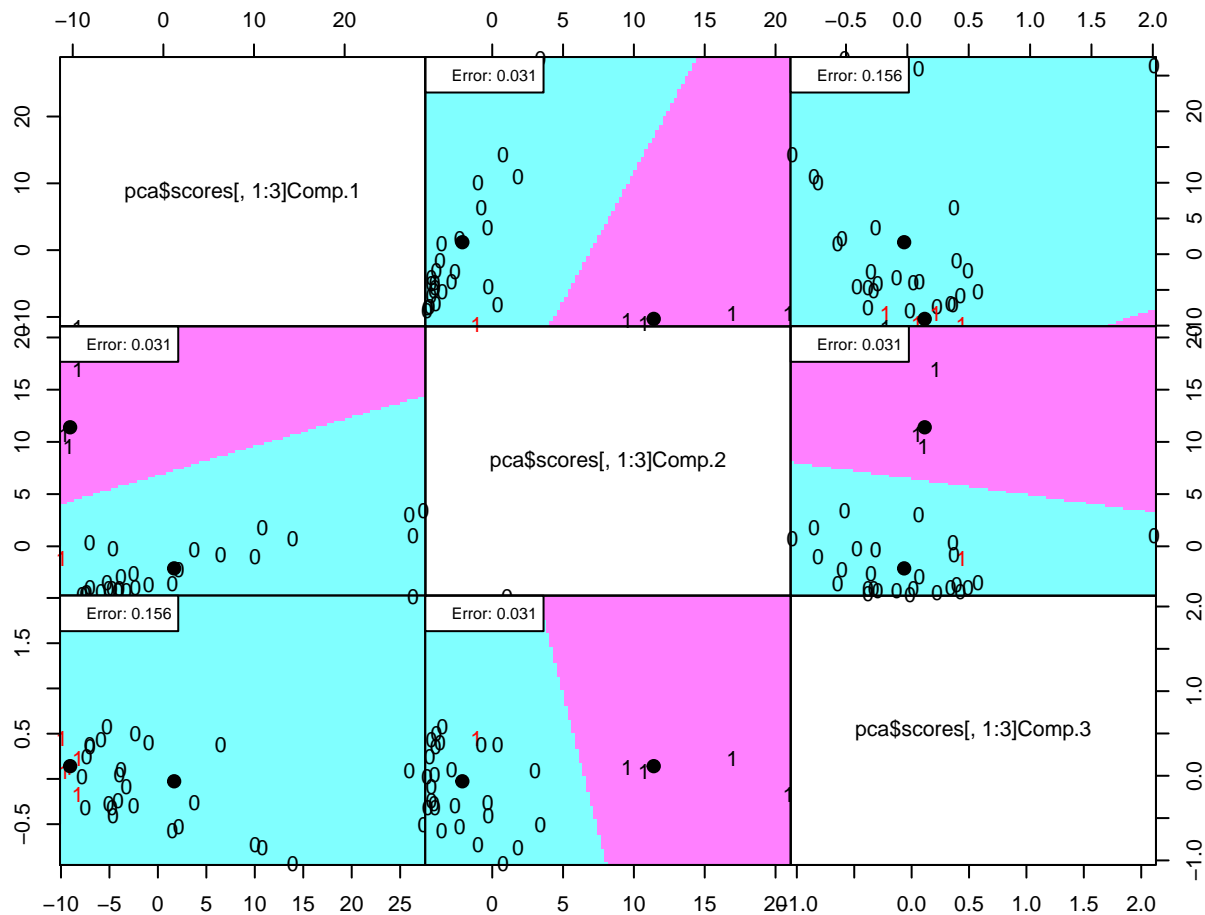
Veureu que hem afegit un plot de les regions de classificació fent servir la funció `partimat`. Els punts que quedessin a la zona blava serien classificats com a contraban i els punts que quedessin a la zona rosa no. Així doncs, apreciem com les decision boundaries del QDA no són lineals, com en el cas del LDA.

A continuació, mostrem la matriu de confusió per les dades d'entrenament:

	0	1
0	27	0
1	0	5

A diferència de quan fèiem LDA, en aquest cas no podem interpretar la importància de les variables tan fàcilment, ja que els discriminants no són combinacions lineals d'aquestes.

Si duem a terme l'equivalent al model anterior però usant LDA veurem la diferència entre els dos mètodes de classificació: aquí els decision boundaries són totalment lineals. Aquest model té un comportament molt similar a l'LDA original (probablement per les components principals tan particulars que ens han sortit).



Aquesta és perfectament diagonal i, per tant, no s'han comès errors a la classificació (error de training del 0%).

Ara duem a terme el LOOCV. A cada iteració fem el PCA de les $n - 1$ observacions que tenim disponibles per entrenar el QDA amb aquestes i no pas les que hem dut a terme abans, ja que estariem afegint en la regla de classificació informació de l'observació que volem predir.

```
sum=0
for (i in 1:n){
  comps <- princomp(dd[-i,2:12])$scores[,1:2]
  qda.mod <- qda(comps, grouping=dd[-i,]$counterfeit)
  obs <- princomp(dd[,2:12])$scores[i,1:2]
  pred <- predict(qda.mod, newdata=obs)
  err <- pred$class==dd[i,"counterfeit"]
  sum <- sum+err
}
(te.err.qda <- sum/n)

[1] 0.96875
```

Un cop estudiats els models QDA i LDA i les seves capacitats discriminants ens centrarem en analitzar la correctesa de les assumpcions que fan aquestes regles de classificació.

Com ja hem esmentat, el LDA assumeix normalitat multivariant de les dades (chisq plot) i igualtat de matriu de covariàncies pels dos grups (BoxM package heplots o). D'aquestes assumpcions, el QDA només en fa la primera.

Per analitzar la normalitat multivariant farem servir el chi-squared plot que consisteix fer un plot dels quantils teòrics contra les nostres dades fent servir la següent propietat de les dades multivariants:

$$(x - \mu)' \Sigma^{-1} (x - \mu)$$

En cas que les nostres dades s'ajustin a una distribució normal multivariant, el plot resultarà en una línia. Farem servir la funció `chisqplot` que vam desenvolupar en la pràctica sobre anàlisi multivariant:

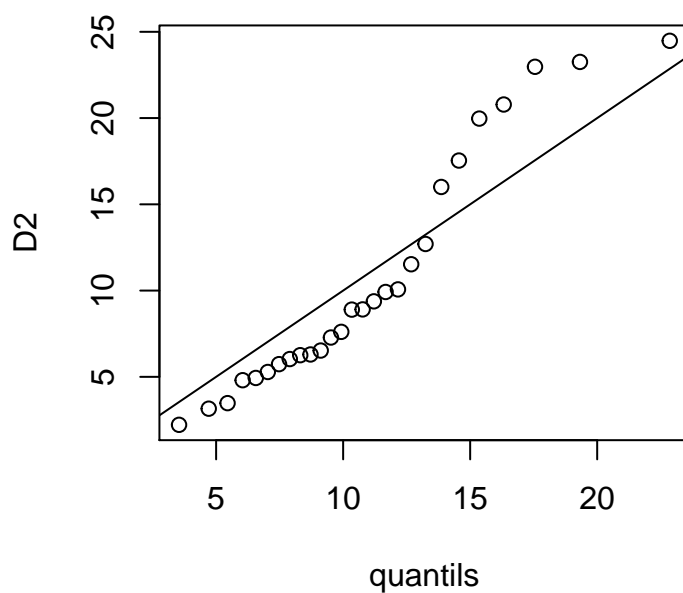
```
chisqplot<-function(data, main){
  X <- as.matrix(data)
  Xcentr <- scale(X, scale=F)
  cov <- cov(X)
  mineig <- min(abs(eigen(cov)$values))
  if(mineig<10^-8) cat("Error: Matriu de covariàncies singular. Ab Minimum eigenvalue: ", mineig, ".\n")
  else{
    D2<-diag(Xcentr%*%as.matrix(solve(cov))%*%t(Xcentr))
    (D2<-sort(D2))
    n<-length(D2)
    (quantils<-qchisq((c(1:n)-0.5)/n, dim(data)[2]))
    plot(quantils,D2, main=main)
    abline(0, 1)
  }
}
```

Per testar la igualtat de matrius de covariàncies farem servir el Box's M Test, un test d'hipòtesi que contrasta igualtat de matriu de covariàncies Farem servir la implementació de la funció `boxM` del paquet `heplots`. La hipòtesi nul·la del test és la igualtat de matrius de covariància, per tant, un p-valor menor al nivell de significació ($\alpha = 0.05$) indicaria que les matrius de covariància són diferents.

Les dades utilitzades per al LDA són les 11 variables sobre la concentració dels elements, així que aplicarem `chisqplot` i `boxM` a aquestes dades. Pel QDA hem fet servir el resultat dels components principals, per tant, els aplicarem en aquests.

Primer tractarem la normalitat multivariant:

Chi-sq plot (Dades no contraban)

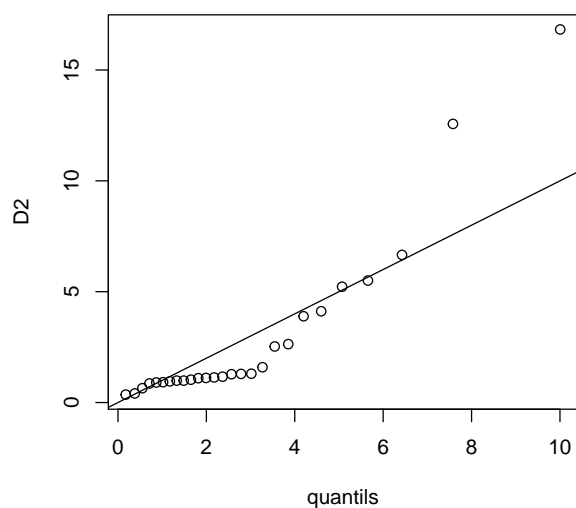


Error: Matriu de covariàncies singular. Ab Minimum eigenvalue: $7.484401e-22$.
No es pot fer el chi-squared plot

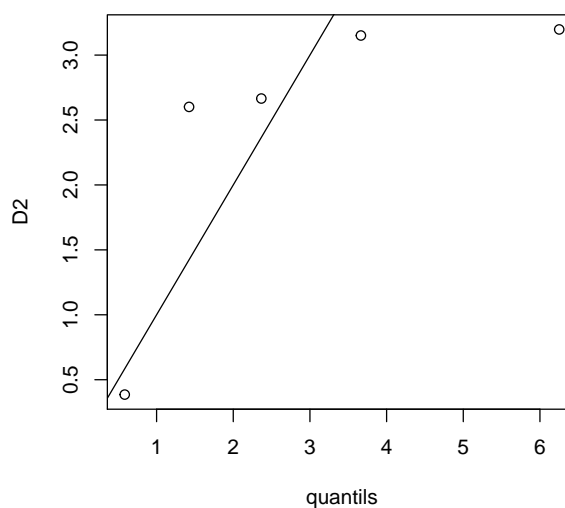
Fent el chi-squared plot de les dades de no contraban observem que aquestes no s'ajusten gaire bé a una normal multivariant: el plot obtingut no és lineal, hi ha uns puja-baixes que ens fan descartar la normalitat.

Pel que fa a les dades de contraban, la matriu de covariàncies és singular i, per tant, la distribució de les nostres dades no és normal multivariant: resulta impossible fer el chi-squared plot atès que les fórmules utilitzades requereixen la inversió d'aquesta matriu.

Chi-sq plot (PCA no contraban)



Chi-sq plot (PCA contraban)



Per les dades de contraban observem un altre cop que les dades no s'ajusten gaire bé a la normalitat multivariant suposada.

A continuació els resultats del Box's M test per les dades de concentracions químiques i pels PCAs, respectivament:

```
Warning in boxM.default(dd[, 2:12], group = dd[, 1]): there are one or more
levels with less observations than variables!
```

Box's M-test for Homogeneity of Covariance Matrices

```
data: dd[, 2:12]
Chi-Sq (approx.) = 34.119, df = 66, p-value = 0.9996
```

No descartem igualtat de matrius de covariància ja que el p-valor es major que el nivell de significació. Es correspon amb les assumpcions del lda.

Box's M-test for Homogeneity of Covariance Matrices

```
data: pca$scores[, 1:3]
Chi-Sq (approx.) = 53.118, df = 6, p-value = 1.111e-09
```

Pels PCAs, la hipòtesi nula és descartada, pel p-valor ($1.11e-09$), i, per tant, acceptem que les matrius de covariància són diferents. Les assumpcions del qda sobre la matriu de covariàncies es compleixen.

Tot i això, és sabut que aquest test no és gaire fiable quan les dades no segueixen molt bé una normal multivariant. Fent els chi-squared plot hem vist que no és el cas per les nostres dades. A més, també es veu molt afectat per una mostra petita i, aquest és el cas pel nostre conjunt de dades, com ja hem mencionat anteriorment.

Sigui com sigui, els dos models QDA i LDA són molt bons en el que es refereix a la classificació de les dades, tot i no complir les assumpcions de normalitat multivariant.

3.3 Resultats i conclusions

En aquest projecte hem pogut usar la majoria de mètodes estudiats a l'assignatura d'anàlisi de dades per a variables contínues.

Una vegada més veiem que, tot i que no existeix cap model correcte, n'hi ha uns quants que són d'allò més útils. Per exemple, tant LDA com QDA assumeixen que les dades es distribueixin seguint una normal multivariant i, com hem comprovat funcionen molt bé. Utilitzant el mètode del LOOCV hem aconseguit estimar la precisió de la predicció dels dos models: 93.75% per LDA i 96.87% per QDA.

A més hem vist quins dels components químics són més importants per identificar els whiskies de contraban en el mètode LDA: el sofre i el potassi.

En conclusió, hem pogut utilitzar les tècniques d'anàlisi discriminant amb èxit encara que les assumpcions fetes pels mètodes no es complissin gaire bé. Per una banda, les dades utilitzades no s'ajustaven gaire bé a la normalitat multivariant. Per l'altra, les assumpcions sobre les matrius de covariàncies dels grups sí que se satisfien, però aquests resultats no són del tot fiables perquè el test utilitzat era molt sensible a la mida del dataset (molt petit) i a la manca de normalitat multivariant.

4 Bibliografia

J. Graffelman (2020) "Apunts de l'assignatura AD-GCED"

C.A. Shand, R. Wendler, L. Dawson, K. Yates, H. Stephenson (2017). "Multivariate Analysis of Scotch Whisky by Total Reflection X-Ray Fluorescence and Chemometric Methods: A Potential Tool in the Identification of Counterfeits," *Analytica Chimica Acta*, Vol. 976, pp. 14-24.

5 Annex

A continuació podeu trobar tot el codi que hem usat per a dur a terme aquest anàlisi:

```
raw <- read.csv("whiskey_chemo.csv", sep=',')

#rownames(raw)<-raw$distillery
dd <- raw
dd <- dd[,4:15]
dd$counterfeit<-as.factor(dd$counterfeit)
n<-dim(dd)[1]
p<-dim(dd)[2]

dd.factor<-raw[,16:26] # Variables indicadores
boxplot(dd[dd.factor[, "P1"]==0, "P"], dd[dd.factor[, "P1"]==1, "P"], main="Fósfor (P)",
        xlab="Variable indicadora", cex.main=0.8, cex.lab=0.6)

require(knitr)
kable(head(dd))

pairs(dd[,c(2:12)], col=ifelse(dd$counterfeit==0, "orange", "blue"))

pca<-princomp(dd[,2:12])
library(pca3d)
pca2d(pca, group=dd$counterfeit, biplot=TRUE, asp=1)

kable(pca$loadings[c("S", "K"),1:2])

plot(dd$K, dd$S, col=ifelse(dd$counterfeit==0, "orange", "blue"))
tab <- as.data.frame(100*(pca$sdev)^2/sum((pca$sdev)^2))
colnames(tab) <- c("% Var explicada")
kable(tab) #Variança explicada pels components

require(MASS)
lda <- lda(scale(dd[, -1]), grouping=dd$counterfeit)
plot(lda)

kable(table(predict(lda)$class, dd$counterfeit))

kable(lda$scaling)

sum=0
i = 0
for (i in 1:n){
  lda.mod<-lda(counterfeit~., data=dd[-i,])
  err<-predict(lda.mod, newdata=dd[i,])$class==dd[i, "counterfeit"]
  sum=sum+err
}
(te.acc.lda<-sum/n)

qda <- qda(dd[,2:12], grouping=dd$counterfeit)

#Fragment de codi en la implementació de la funció qda
g <- as.factor(grouping)
lev <- levels(g)
```

```

counts <- as.vector(table(g))
names(counts) <- lev
if(any(counts < p+1)) stop("some group is too small for 'qda'")

require(klaR)
qda <- qda(pca$scores[,1:3], grouping=dd$counterfeit)
partimat(as.factor(counterfeit) ~ pca$scores[,1:3], data=dd, method="qda", plot.matrix = TRUE)

kable(table(predict(qda)$class,dd$counterfeit))

partimat(as.factor(counterfeit) ~ pca$scores[,1:3], data=dd, method="lda", plot.matrix = TRUE)
sum=0
for (i in 1:n){
  comps <- princomp(dd[-i,2:12])$scores[,1:2]
  qda.mod <- qda(comps, grouping=dd[-i,]$counterfeit)
  obs <- princomp(dd[,2:12])$scores[i,1:2]
  pred <- predict(qda.mod, newdata=obs)
  err <- pred$class==dd[i,"counterfeit"]
  sum <- sum+err
}
(te.err.qda <- sum/n)

chisqplot<-function(data, main){
  X <- as.matrix(data)
  Xcentr <- scale(X, scale=F)
  cov <- cov(X)
  mineig <- min(abs(eigen(cov)$values))
  if(mineig<10^-8) cat("Error: Matriu de covariàncies singular. Ab Minimum eigenvalue: ", mineig, ".\n")
  else{
    D2<-diag(Xcentr%*%as.matrix(solve(cov))%*%t(Xcentr))
    (D2<-sort(D2))
    n<-length(D2)
    (quantils<-qchisq((c(1:n)-0.5)/n, dim(data)[2]))
    plot(quantils,D2, main=main)
    abline(0, 1)
  }
}

chisqplot(dd[dd$counterfeit==0, 2:12], main="Chi-sq plot (Dades no contraban)")
chisqplot(dd[dd$counterfeit==1, 2:12], main="Chi-sq plot (Dades contraban)")

par(mfrow=c(1,2))
chisqplot(pca$scores[dd$counterfeit==0,1:3], main="Chi-sq plot (PCA no contraban)")
chisqplot(pca$scores[dd$counterfeit==1,1:3], main="Chi-sq plot (PCA contraban)")

library(heplots)
boxM(dd[,2:12],group=dd[,1])

boxM(pca$scores[,1:3],group=dd[,1])

labs = knitr::all_labels()
labs = setdiff(labs, c("setup", "get-labels"))
sessionInfo()

```


Aquest informe ha estat generat usant R 3.6.3 i els paquets detallats a continuació:

R version 3.6.3 (2020-02-29)

Platform: x86_64-pc-linux-gnu (64-bit)

Running under: Ubuntu 20.04 LTS

Matrix products: default

BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0

LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0

locale:

```
[1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
[3] LC_TIME=es_ES.UTF-8      LC_COLLATE=en_US.UTF-8
[5] LC_MONETARY=es_ES.UTF-8  LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=es_ES.UTF-8     LC_NAME=C
[9] LC_ADDRESS=C             LC_TELEPHONE=C
[11] LC_MEASUREMENT=es_ES.UTF-8 LC_IDENTIFICATION=C
```

attached base packages:

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

other attached packages:

```
[1] heplots_1.3-5  car_3.0-7      carData_3.0-3  klaR_0.6-15    MASS_7.3-51.5
[6] pca3d_0.10.1  knitr_1.28
```

loaded via a namespace (and not attached):

```
[1] xfun_0.13          haven_2.2.0      vctrs_0.2.4
[4] miniUI_0.1.1.1     htmltools_0.4.0  yaml_2.2.1
[7] rlang_0.4.6        manipulateWidget_0.10.1 later_1.0.0
[10] pillar_1.4.4       foreign_0.8-75   readxl_1.3.1
[13] lifecycle_0.2.0    questionr_0.7.0  stringr_1.4.0
[16] cellranger_1.1.0   combinat_0.0-8   zip_2.0.4
[19] htmlwidgets_1.5.1  evaluate_0.14    rio_0.5.16
[22] forcats_0.5.0      fastmap_1.0.1    httpuv_1.5.2
[25] crosstalk_1.1.0.1  curl_4.3         highr_0.8
[28] Rcpp_1.0.4.6       xtable_1.8-4     promises_1.1.0
[31] webshot_0.5.2      jsonlite_1.6.1   abind_1.4-5
[34] mime_0.9           ellipse_0.4.1    hms_0.5.3
[37] digest_0.6.25      stringi_1.4.6    openxlsx_4.1.4
[40] shiny_1.4.0.2      tools_3.6.3      magrittr_1.5
[43] rgl_0.100.54       tibble_3.0.1     crayon_1.3.4
[46] pkgconfig_2.0.3    ellipsis_0.3.0   data.table_1.12.8
[49] rmarkdown_2.1      rstudioapi_0.11  R6_2.4.1
[52] compiler_3.6.3
```