Análisis Estadístico Multivariante

Práctica 5: Análisis Discriminante

Francisco Javier Mercader Martínez

PRÁCTICA 5: ANÁLISIS DISCRIMINANTE

ANÁLISIS ESTADÍSTICO MULTIVARIANTE

GRADO EN CIENCIA E INGENIERÍA DE DATOS

Sumario: En esta práctica mostramos cómo clasificar individuos entre diversos grupos a partir de sus medidas en algunas variables. Para ello necesitaremos disponer de una muestra de las variables en estudio en individuos de cada grupo (al menos dos individuos por cada grupo) (clasificación supervisada). Necesitaremos instalar los siguientes paquetes: **MASS, dplyr, mvtnorm, mvnormtest**.

1. Estudio descriptivo inicial

Como en el resto de las técnicas aplicadas previamente, es conveniente hacer un análisis inicial del conjunto de datos. En este caso trataremos de estudiar las diferencias de las variables en cada uno de los grupos para dilucidar si serán de utilidad a la hora de clasificar (discriminar) a los individuos de los distintos grupos.

Comenzaremos leyendo los datos del objeto **d** del fichero **escarabajos.rda** (disponible en el aula virtual). Para cargar este archivo utilizamos la función *load*, indicando la ruta completa en dónde se encuentra el archivo o cambiando el directorio de trabajo.

```
load("../data/escarabajos.rda")
```

Empezamos mostrando la estructura del conjunto de datos:

```
## completar aquí
str(d)
```

```
## 'data.frame':    40 obs. of  6 variables:
##  $ surco  : num  189 192 217 221 171 192 213 192 170 201 ...
##  $ long   : num  245 260 276 299 239 262 278 255 244 276 ...
##  $ base2  : num  137 132 141 142 128 147 136 128 128 146 ...
##  $ base3  : num  163 217 192 213 158 173 201 185 192 186 ...
##  $ especie: Factor w/ 3 levels "* ","HC","HO": 3 3 3 3 3 3 3 3 3 3 ...
##  $ codigo : int  1 1 1 1 1 1 1 1 1 1 ...
```

Hacemos un resumen numérico:

```
## completar aquí
summary(d)
```

```
##      surco           long          base2          base3       especie
##  Min.   :158.0   Min.   :237.0   Min.   :121.0   Min.   :158.0   * : 1
##  1st Qu.:177.0   1st Qu.:262.8   1st Qu.:137.8   1st Qu.:187.5   HC:20
##  Median :183.1   Median :278.0   Median :146.0   Median :194.5   HO:19
```

```
##  Mean   :186.7   Mean   :279.0   Mean   :147.4   Mean   :197.7
##  3rd Qu.:192.8   3rd Qu.:299.0   3rd Qu.:160.5   3rd Qu.:213.0
##  Max.   :221.0   Max.   :317.0   Max.   :184.0   Max.   :235.0
##
##      codigo
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :2.000
##  Mean   :1.513
##  3rd Qu.:2.000
##  Max.   :2.000
##  NA's   :1
```

Si queremos visualizar el conjunto de datos completo podemos utilizar el comando *View(d)*. También podemos ver las primeras filas del conjunto de datos:

```
## completar aquí
head(d, n=6)
```

```
##   surco long base2 base3 especie codigo
## 1   189  245   137   163      HO      1
## 2   192  260   132   217      HO      1
## 3   217  276   141   192      HO      1
## 4   221  299   142   213      HO      1
## 5   171  239   128   158      HO      1
## 6   192  262   147   173      HO      1
```

Comprobaremos así que dicho fichero contiene una muestra de 40 escarabajos de dos especies diferentes Haltica Oleracea (HO) y Haltica Carduorum (HC) a los que se les han medido cuatro variables:

surco (X1): distancia desde el tórax al surco transversal (micras),

long (X2): longitud (0.01 mm.),

base2 (X3): longitud de la base de las antenas secundarias (micras),

base3 (X4): longitud de la base de las antenas terciarias (micras).

Las variables especie y código indican la especie a la que pertenece cada individuo (HO = 1, HC = 2). Puede observarse que hay un escarabajo (40) del que se desconoce la especie lo que en *R* se especifica con *NA* (*Not Available*).

Podemos comenzar estudiando cada variable por separado. Para ver solo los datos de la variable *surco* haremos:

```
d$surco
```

```
##  [1] 189.00 192.00 217.00 221.00 171.00 192.00 213.00 192.00 170.00 201.00
## [11] 195.00 205.00 180.00 192.00 200.00 192.00 200.00 181.00 192.00 181.00
## [21] 158.00 184.00 171.00 181.00 181.00 177.00 198.00 180.00 177.00 176.00
## [31] 192.00 176.00 169.00 164.00 181.00 192.00 181.00 175.00 197.00 182.22
```

```
## d[, 1] # Otra forma
```

Por ejemplo, para analizar el comportamiento de esta variable, podemos comenzar calculando sus estadísticos básicos (medias, cuartiles y valores extremos) en cada grupo haciendo:

```
tapply(d$surco, d$especie, summary)
```

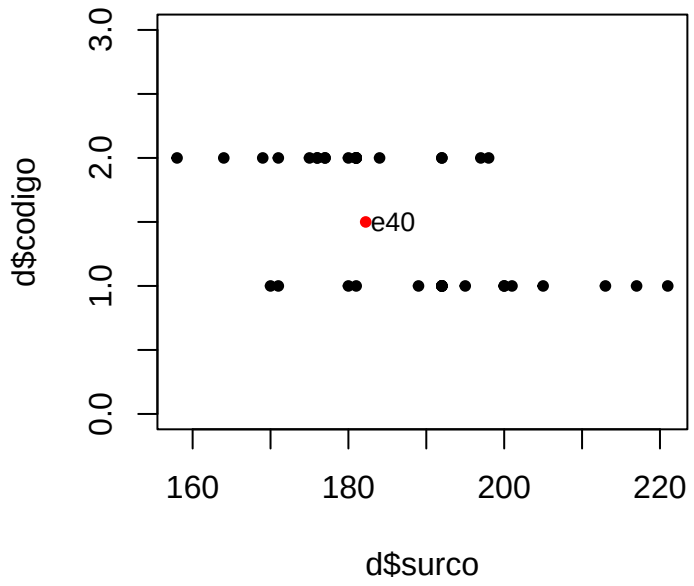
```
## $`* `
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   182.2   182.2   182.2   182.2   182.2   182.2
##
## $HC
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   158.0   175.8   180.5   179.6   181.8   198.0
##
## $HO
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   170.0   190.5   192.0   194.5   200.5   221.0
```

De esta forma observamos que la media de la variable surco es mayor en la especie HO (194.5) que en la HC (179.6) y que su valor en el escarabajo 40 (182.2) está más cerca de la media de la especie HC.

También podemos representarla gráficamente indicando dónde se encuentra el escarabajo 40 a la altura 1.5:

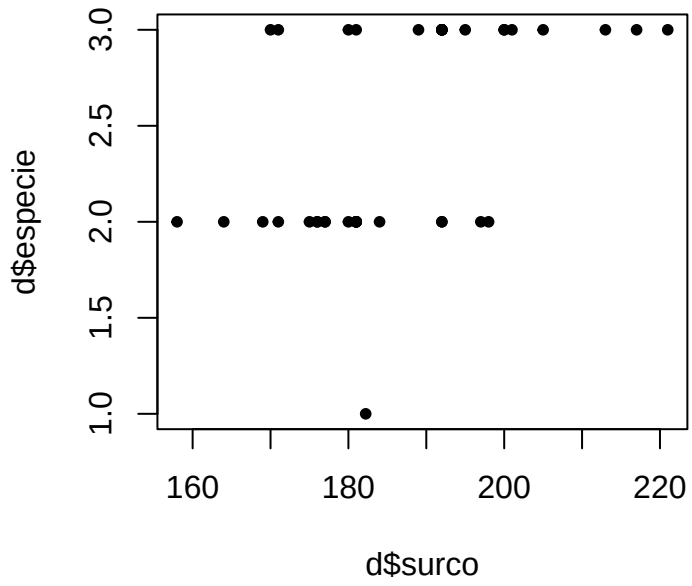
```
plot(d$surco, d$codigo, ylim = c(0, 3), pch = 20)
points(d$surco[40], 1.5, pch = 20, col = "red")
text(d$surco[40] + 3.5, 1.5, labels = "e40", cex = 0.8)
```



En esta gráfica podemos observar que la variable *surco* parece un poco mayor en el grupo 1 (HO) pero que no discrimina (separa) bien a los grupos. Con esta variable no es sencillo clasificar al escarabajo 40 pero si tenemos que elegir un grupo lo incluiríamos en el grupo 2 (HC) ya que está más cerca de su media.

Se obtiene una gráfica similar haciendo

```
plot(d$surco, d$especie, pch = 20)
```



En este caso, *R* etiqueta los datos por orden alfabético (ASCII) con * = 1, HC = 2 y HO = 3.

Continuamos estudiando las restantes variables. Para la variable *long* tendremos:

```
## completar aquí
d$long
```

```
##  [1] 245.00 260.00 276.00 299.00 239.00 262.00 278.00 255.00 244.00 276.00
## [11] 242.00 263.00 252.00 283.00 294.00 277.00 287.00 255.00 287.00 305.00
## [21] 237.00 300.00 273.00 297.00 308.00 301.00 308.00 286.00 299.00 317.00
## [31] 312.00 285.00 287.00 265.00 308.00 276.00 278.00 271.00 303.00 271.01
```

Y el siguiente resumen numérico por grupos:

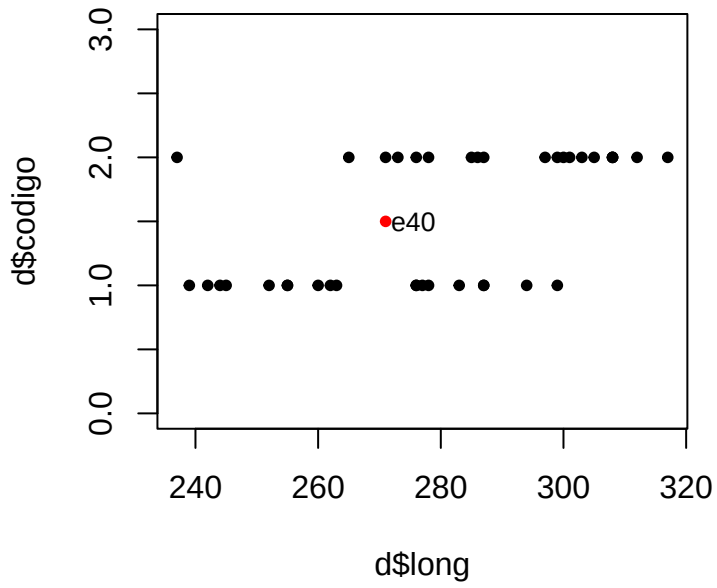
```
## completar aquí
tapply(d$long, d$especie, summary)
```

```
## $`* `
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     271     271     271     271     271     271
##
## $HC
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   237.0   277.5   298.0   290.8   305.8   317.0
##
## $HO
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   239.0   253.5   263.0   267.1   280.5   299.0
```

En este caso observamos que la media de la variable *long* es mayor en la especie HC (290.8) que en la HO (267.1) y que su valor en el escarabajo 40 (271) está más cerca de la media de la especie HO.

Y la representación gráfica:

```
## completar aquí
plot(d$long, d$codigo, ylim=c(0,3), pch=20)
points(d$long[40], 1.5, pch=20, col="red")
text(d$long[40] + 4.5, 1.5, labels = "e40", cex=0.8)
```



En esta gráfica podemos observar que la variable *long* parece un poco menor en el grupo 1 (HO) pero que no discrimina (separa) bien a los grupos. Según esta variable, al escarabajo 40 lo clasificaríamos en el grupo 1 (HO) ya que está más cerca de su media.

Para la variable *base2* tendremos:

```
## completar aquí
d$base2
```

```
##  [1] 137.00 132.00 141.00 142.00 128.00 147.00 136.00 128.00 128.00 146.00
## [11] 128.00 147.00 121.00 138.00 138.00 150.00 136.00 146.00 141.00 184.00
## [21] 133.00 166.00 162.00 163.00 160.00 166.00 141.00 146.00 171.00 166.00
## [31] 166.00 141.00 162.00 147.00 157.00 154.00 149.00 140.00 170.00 140.99
```

Y el siguiente resumen numérico por grupos:

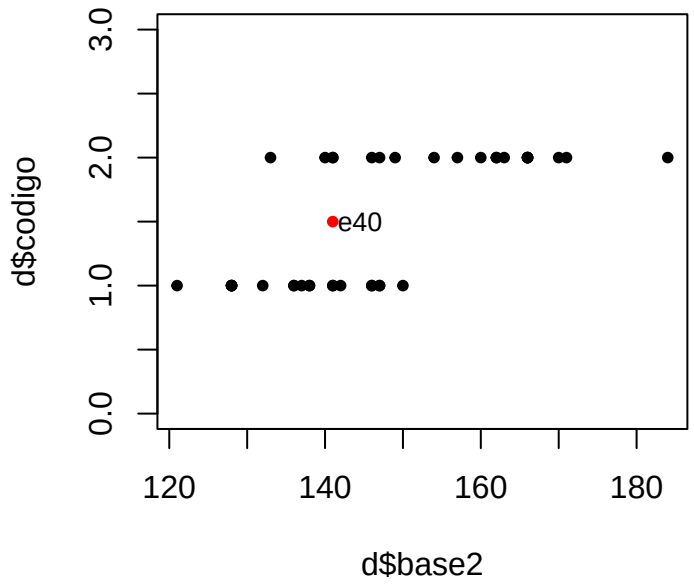
```
## completar aquí
tapply(d$base2, d$especie, summary)
```

```
## $`* `
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     141     141     141     141     141     141
##
## $HC
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##   133.0   146.8   161.0   157.2   166.0   184.0
##
## $HO
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   121.0   130.0   138.0   137.4   144.0   150.0
```

La media de la variable *base2* es mayor en la especie HC (157.2) que en la HO (137.4) y que su valor en el escarabajo 40 (141) está más cerca de la media de la especie HO. Si hacemos la representación gráfica:

```
## completar aquí
plot(d$base2, d$codigo, ylim=c(0,3), pch=20)
points(d$base2[40], 1.5, pch=20, col="red")
text(d$base2[40] + 3.5, 1.5, labels = "e40", cex=0.8)
```



Para la variable *base3* se observan los valores:

```
d$base3
```

```
##  [1] 163.00 217.00 192.00 213.00 158.00 173.00 201.00 185.00 192.00 186.00
## [11] 192.00 192.00 167.00 183.00 188.00 177.00 173.00 183.00 198.00 209.00
## [21] 188.00 231.00 213.00 224.00 223.00 221.00 197.00 214.00 192.00 213.00
## [31] 209.00 200.00 214.00 192.00 204.00 209.00 235.00 192.00 205.00 190.15
```

Y obtenemos el siguiente resumen numérico por grupos:

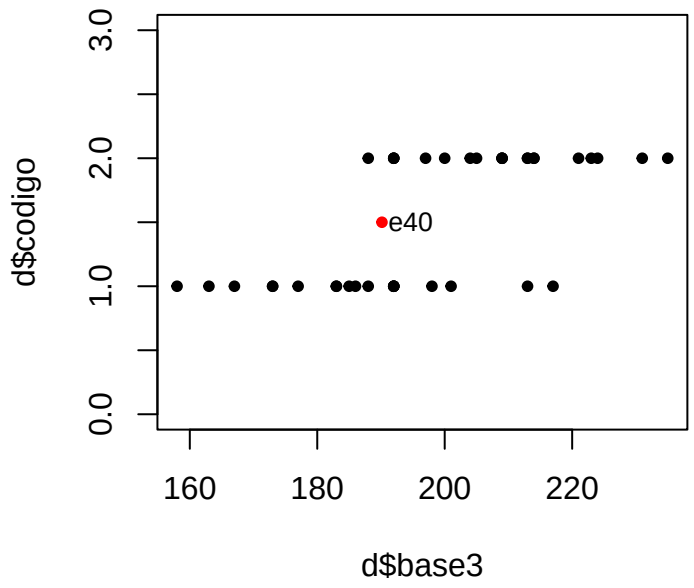
```
## completar aquí
tapply(d$base3, d$especie, summary)
```

```
## $`* `
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   190.1   190.1   190.1   190.1   190.1   190.1
##
```

```
## $HC
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   188.0   199.2   209.0   209.2   215.8   235.0
##
## $HO
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   158.0   175.0   186.0   185.9   192.0   217.0
```

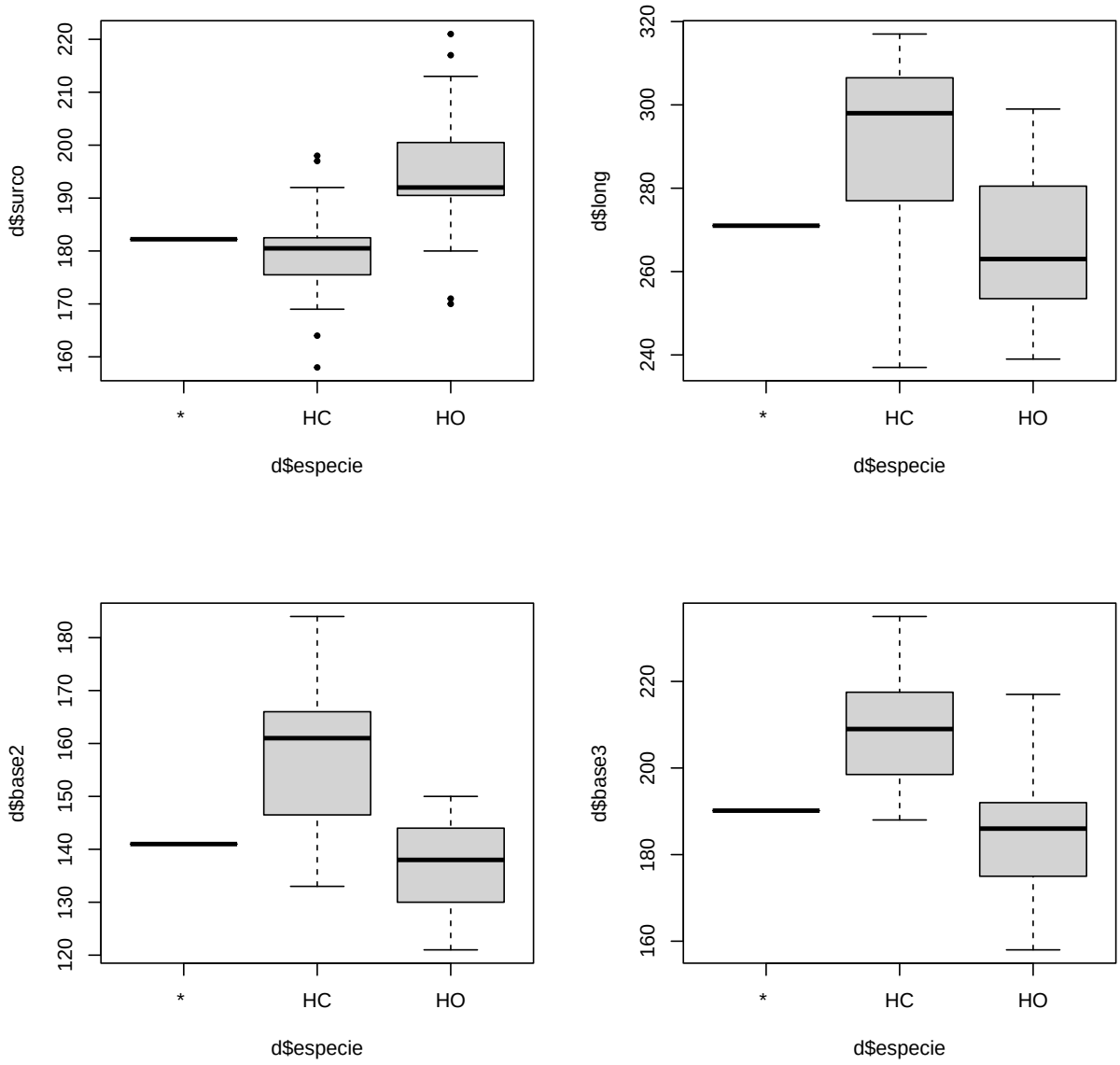
La media de la variable *base3* es mayor en la especie HC (209.2) que en la HO (185.9) y que el valor para el escarabajo 40 (190.1) está más cerca de la media de la especie HO. Y la representación gráfica vendrá dada por:

```
## completar aquí
plot(d$base3, d$codigo, ylim=c(0,3), pch=20)
points(d$base3[40], 1.5, pch=20, col="red")
text(d$base3[40] + 4.5, 1.5, labels = "e40", cex=0.8)
```



Para visualizar estas diferencias también podemos usar los gráficos caja-bigote por grupos.

```
par(mfrow = c(2, 2))
boxplot(d$surco ~ d$especie, pch = 20)
boxplot(d$long ~ d$especie, pch = 20)
boxplot(d$base2 ~ d$especie, pch = 20)
boxplot(d$base3 ~ d$especie, pch = 20)
```

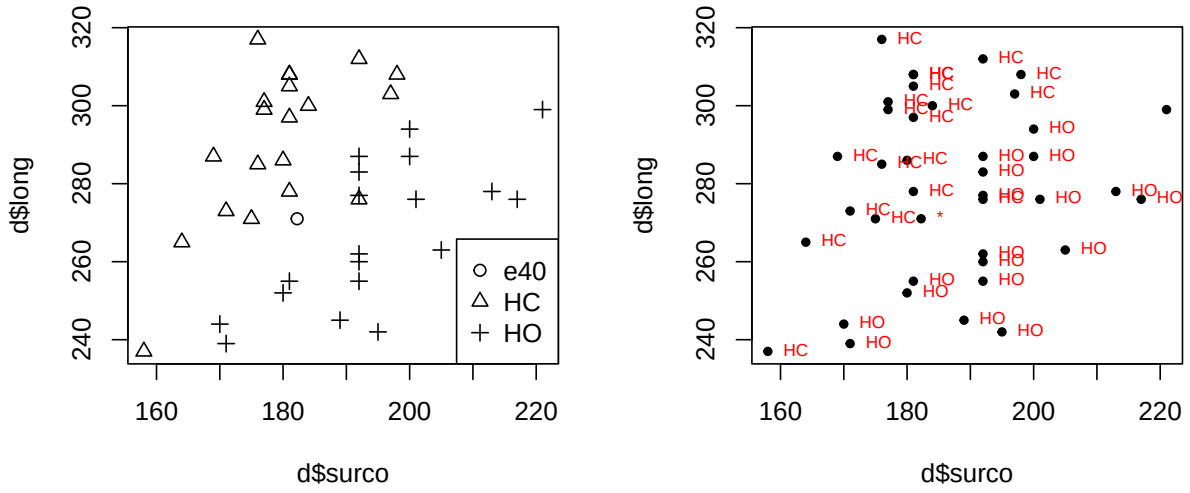


En estos gráficos apreciamos que si usáramos solo la variable *surco* para clasificar, como las cajas no se solapan, más del 75% de los individuos se clasificarían bien. También observamos que el escarabajo 40 estaría en la caja de la especie HC (por poco) pero que no sería un valor atípico en la HO. En ambas especies hay observaciones atípicas (para la distribución normal).

En segundo lugar podemos estudiar las variables por parejas. Por ejemplo, para analizar *surco* y *long*, podemos hacer:

```
par(mfrow = c(1, 2))
plot(d$surco, d$long, pch = as.integer(d$especie))
legend("bottomright", legend = c("e40", "HC", "HO"), pch = 1:3)

plot(d$surco, d$long, pch = 20)
text(d$surco, d$long, d$especie, cex = 0.7, pos = 4, col = "red")
```

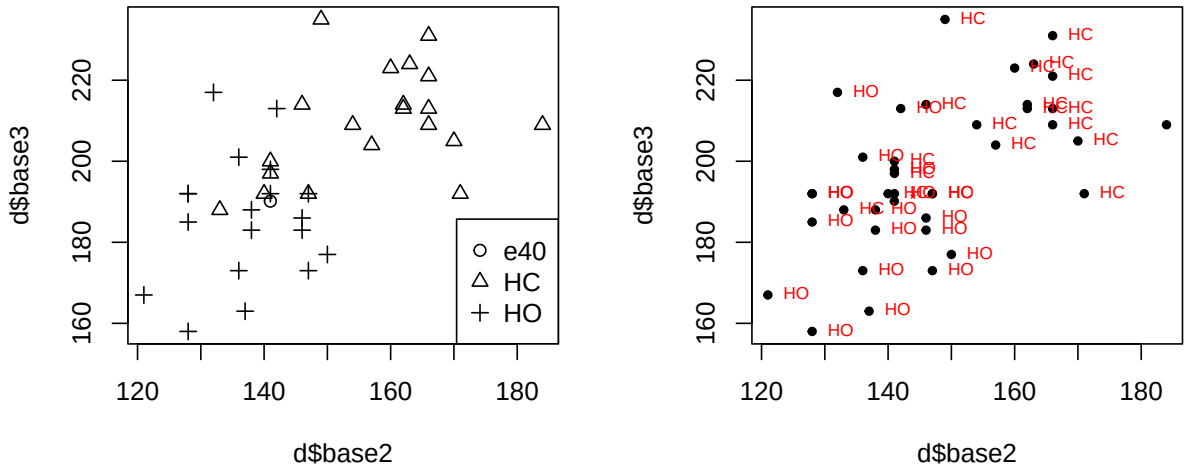


(el argumento cex indica el tamaño y pos la posición de la etiqueta). En esta última gráfica se observa que, con estas dos variables, los dos 2 grupos están bastante separados, pero que el escarabajo 40 estaría entre ambos grupos por lo que no es sencillo clasificarlo.

Representamos ahora el gráfico de las otras dos variables:

```
## completar aquí
par(mfrow = c(1,2))
plot(d$base2, d$base3, pch=as.integer(d$especie))
legend("bottomright", legend = c("e40", "HC", "HO"), pch = 1:3)

plot(d$base2, d$base3, pch = 20)
text(d$base2, d$base3, d$especie, cex = 0.7, pos = 4, col = "red")
```



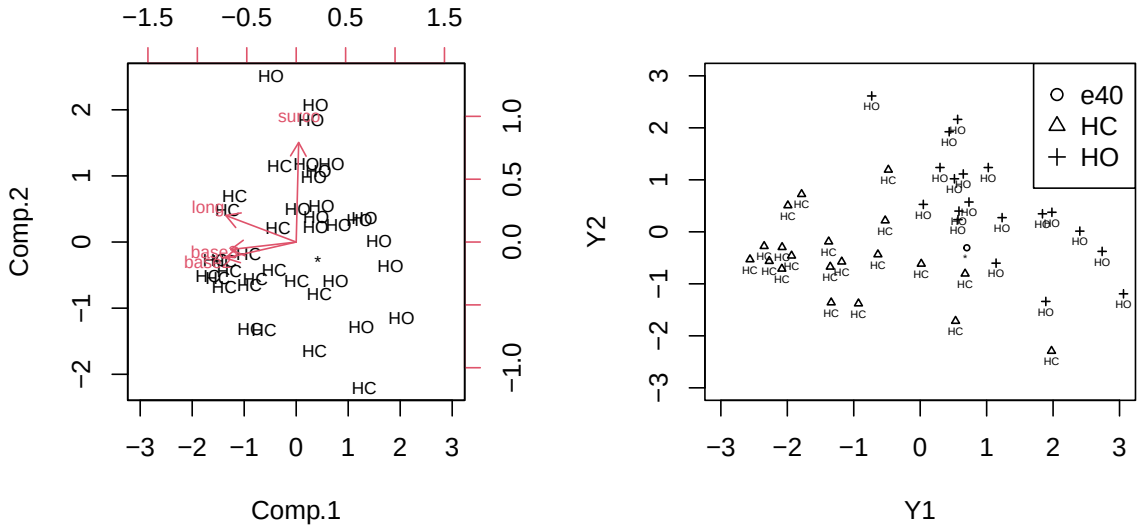
Finalmente, podemos hacer un gráfico similar al anterior pero usando las dos primeras componentes principales (ver práctica 4) que contienen información sobre todas las variables. Recordemos que las componentes principales (basadas en la matriz de correlaciones) y las puntuaciones (*scores*) se calculan con:

```
PCA <- princomp(d[, 1:4], cor = TRUE)
S <- PCA$scores
```

y que se pueden representar las dos primeras componentes por grupos con los gráficos:

```
par(mfrow = c(1, 2))
biplot(PCA, pc.biplot = TRUE, xlabs = d$especie, cex = 0.7, xlim = c(-3, 3))

plot(S[, 1], S[, 2], xlab = 'Y1', ylab = 'Y2', pch = as.integer(d$especie), cex = 0.5,
     xlim = c(-3, 3), ylim = c(-3, 3))
text(S[, 1], S[, 2]-0.2, labels = d$especie, cex = 0.4)
legend("topright", legend = c("e40", "HC", "HO"), pch = 1:3)
```



En estos gráficos también se aprecia que los grupos se pueden separar bastante bien teniendo el grupo HC medidas grandes en todas las variables excepto en surco. De nuevo, el escarabajo 40 aparece cerca de la frontera entre ambos grupos. Señalar no obstante que las dos primeras componentes principales no son necesariamente las mejores variables para clasificar a estos individuos (como veremos en las secciones siguientes).

2. Análisis Discriminante Lineal (LDA)

Para calcular la función discriminate lineal (FDL) de Fisher para distinguir entre dos grupos debemos suponer que sus matrices de covarianzas (teóricas) son iguales. Entonces la FDL valdrá $L(\mathbf{Z}) = \mathbf{a}'\mathbf{Z}$ donde $\mathbf{Z} = (Z_1, \ldots, Z_k)'$ son las medidas del individuo a clasificar y los coeficientes se calculan como

$$\mathbf{a}' = \lambda(\mu_{\mathbf{X}} - \mu_{\mathbf{Y}})'V^{-1},$$

donde λ es un número real cualquiera distinto de cero, V es la matriz de covarianzas común y $\mu_{\mathbf{X}}$ y $\mu_{\mathbf{Y}}$ son los vectores de medias en cada grupo de las variables usadas para clasificar. En la práctica estas medias teóricas se sustituyen por sus estimaciones $\overline{\mathbf{x}}$ e $\overline{\mathbf{y}}$ y V se estima mediante una media ponderada de las matrices de

cuasicovarianzas muestrales:

$$S = \frac{1}{n_1 + n_2 - 2}\left[(n_1 - 1)S_1 + (n_2 - 1)S_2\right]$$

siendo n_1 y n_2 los tamaños muestrales de cada grupo y S_1 y S_2 las matrices de cuasicovarianzas muestrales de cada grupo.

Para calcular (estimar) el vector **a** en *R* debemos cargar primero el paquete denominado *MASS* y, una vez cargado, utilizamos la función *lda*:

```
library("MASS")
LDA <- lda(x = d[1:39, 1:4], grouping = d[1:39, 6], prior = c(0.5, 0.5))
```

En la función *lda* hemos especificado que las probabilidades de pertenencia a priori asignadas a cada grupo valen 0.5. Si no se especifica nada se computan como si los individuos fuesen una muestra, esto es, como $19/39 = 0.4871795$ (HO) y $20/39 = 0.5128205$ (HC)). A partir del objeto **LDA**, estos valores también se pueden calcular:

```
## completar aquí
LDA$counts/LDA$N
```

```
##         1         2
## 0.4871795 0.5128205
```

(HO = 1, HC = 2), así como los vectores de medias de los grupos:

```
## completar aquí
LDA$means
```

```
##      surco     long    base2    base3
## 1 194.4737 267.0526 137.3684 185.9474
## 2 179.5500 290.8000 157.2000 209.2500
```

Los coeficientes estimados de la FDL son

```
## completar aquí
LDA$scaling
```

```
##               LD1
## surco -0.09327642
## long   0.03522706
## base2  0.02875538
## base3  0.03872998
```

Esto es,

$$\mathbf{a}' = (-0.09327642, 0.03522706, 0.02875538, 0.03872998).$$

Si queremos guardar estos coeficientes en el objeto **a** haremos:

```
a = LDA$scaling
```

Para clasificar a un individuo con medidas **z** calcularemos su proyección $L(\mathbf{z}) = \mathbf{a}'\mathbf{z}$ y las proyecciones de las medias de los grupos $L(\overline{\mathbf{x}})$ y $L(\overline{\mathbf{y}})$, clasificándolo en el grupo que tenga la media más cercana a su proyección. La frontera de las regiones de clasificación vendrá dada por la media de las proyecciones de las medias: $K = (L(\overline{\mathbf{x}}) + L(\overline{\mathbf{y}}))/2$. Para calcular L podemos definir la función:

```
L <- function(z) sum(a*z)
```

De esta forma, podemos calcular la proyección de la media $L(\overline{\mathbf{X}})$ de la especie HO haciendo:

```
mHO <- L(LDA$means[1, ])
```

obteniendo $mHO = L(\overline{\mathbf{x}}) = 2.419488$. Análogamente, podemos calcular

```
mHC <- L(LDA$means[2, ])
```

obteniendo $mHC = L(\overline{\mathbf{y}}) = 6.120841$. De esta forma, haciendo

```
K <- (mHC + mHO)/2
```

obtenemos $K = 4.270164$. Por lo tanto, la regla de decisión óptima según este criterio sería:

$$\text{Regla de decisión} = \begin{cases} \text{Se clasifica como HC (grupo 2)} & \text{si } L(\mathbf{z}) > K \\ \text{Se clasifica como HO (grupo 1)} & \text{si } L(\mathbf{z}) < K \end{cases}$$

Podemos calcular las proyecciones de los 40 escarabajos haciendo:

```
D <- vector("numeric", length = 40)
for (i in 1:40) D[i] <- L(d[i, 1:4])
```

Y podemos obtener la clasificación de cada individuo y si es correcta o no:

```
library("dplyr")
d$D = D
d <- d %>%
  mutate(clasificacion = ifelse(D < K,
                                "HO",
                                "HC"),
         error = ifelse(especie == clasificacion,
                        0,
                        1))
```

Para las primeras filas tendremos:

```
## completar aquí
head(d)
```

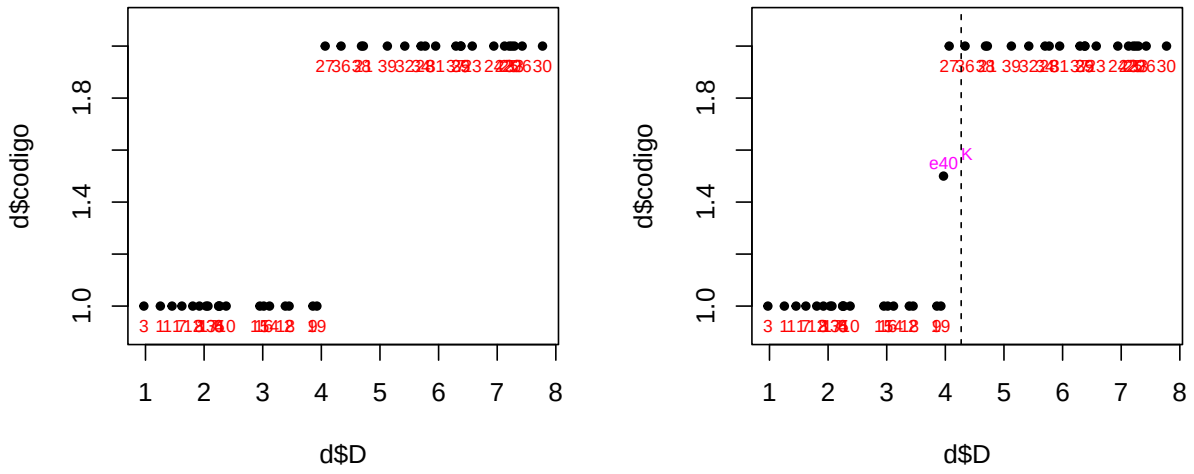
```
##   surco long base2 base3 especie codigo        D clasificacion error
## 1   189  245   137   163      HO      1 1.253859            HO     0
## 2   192  260   132   217      HO      1 3.450078            HO     0
## 3   217  276   141   192      HO      1 0.972349            HO     0
## 4   221  299   142   213      HO      1 2.251551            HO     0
## 5   171  239   128   158      HO      1 2.269024            HO     0
## 6   192  262   147   173      HO      1 2.247743            HO     0
```

Y comprobamos que para el escarabajo 1 se obtiene $D[1] = 1.253859$ que, como es menor que $K = 4.270164$, nos conducirá a clasificarlo como del grupo HO (correctamente, error $= 0$). Análogamente, para el escarabajo 40, obtenemos $D[40] = 3.968782$ que, como es menor que K, nos conducirá a clasificarlo como del grupo HO (con un margen pequeño). Podemos representar estas **puntuaciones discriminates** e incluir la puntuación del escarabajo 40 y la constante K en el gráfico haciendo:

```
par(mfrow = c(1, 2))
plot(d$D, d$codigo, ylim = c(0.9, 2.1), pch = 20)
text(d$D, d$codigo, cex = 0.7, pos = 1, col = "red")

plot(d$D, d$codigo, ylim = c(0.9, 2.1), pch = 20)
text(d$D, d$codigo, cex = 0.7, pos = 1, col = "red")
```

```
points(D[40], 1.5, pch = 20)
text(D[40], 1.55, labels = "e40", cex =0.7, col = "magenta")
abline(v = K, lty = 2)
text(K + 0.1, 1.5, labels ="K", cex = 0.7, pos = 3, col ="magenta")
```



En este gráfico se observa que el escarabajo 27 es el único que se clasificaría erróneamente y que el 40 se clasificaría en el grupo 1 (HO) pero con un margen pequeño (está cerca de la frontera marcada por K).

Otros autores prefieren calcular las puntuaciones como $D - K$ con lo que la regla de decisión dependerá de si las puntuaciones son positivas o negativas. La puntuación $D - K$ se puede obtener de forma automática utilizando la función *predict*:

```
P <- predict(LDA, d[, 1:4])
```

En *P$x* están incluidos los valores de la función descriminante, para ver los primeros valores utilizaremos:

```
## completar aquí
head(P$x)
```

```
##             LD1
## [1,] -3.0163053
## [2,] -0.8200868
## [3,] -3.2978154
## [4,] -2.0186138
## [5,] -2.0011403
## [6,] -2.0224209
```

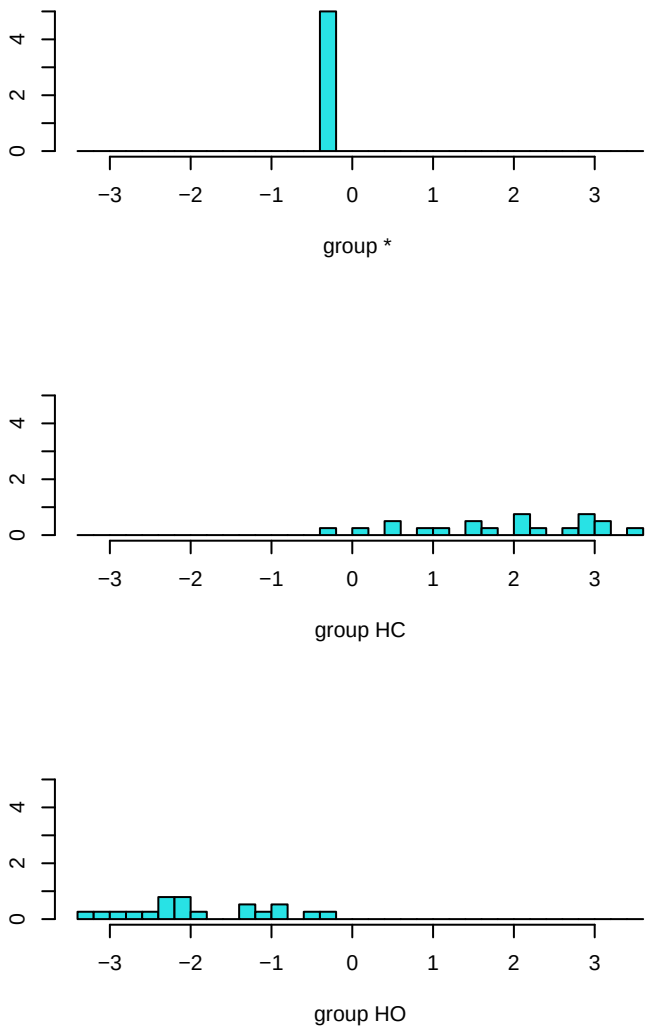
Y podemos comprobar que coinciden con los valores de *D-K*.

```
## completar aquí
d <- d %>%
  mutate(LD1 = D-K)
head(d$LD1)
```

```
## [1] -3.0163053 -0.8200868 -3.2978154 -2.0186138 -2.0011403 -2.0224209
```

Estos valores se pueden representar utilizando un histograma:

```
ldahist(P$x, g = d$especie)
```



Para ver en qué grupo se clasifican los 40 escarabajos accedemos a **P$class** y podemos ver si la clasificación es correcta para los 39 escarabajos de los que se conoce su grupo:

```
## completar aquí
P$class
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2
## [39] 2 1
## Levels: 1 2
```

```
P$class == d[, 6]
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [25]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
## [37]  TRUE  TRUE  TRUE    NA
```

Podemos hacer un recuento de estos resultados con:

```
## completar aquí
table(d[, 6], P$class)
```

```
##
##      1  2
##   1 19  0
##   2  1 19
```

Y resumir la información en la siguiente tabla (matriz de confusión):

Resumen	Clasificados en 1 (HO)	Clasificados en 2 (HC)	Total
Grupo verdadero: 1 (H0)	19	0	19
Grupo verdadero: 2 (HC)	1	19	20
Total	20	19	39

Esta tabla sirve para comprobar si este procedimiento de clasificación es adecuado. En este caso, obtenemos buenos resultados ya que todos los individuos del primer grupo se clasifican correctamente y sólo uno del grupo 2 (el escarabajo 27) se clasifica erróneamente como del grupo 1. Análogamente, comprobamos que todos los individuos clasificados como del grupo 2 se han clasificado correctamente pero que uno clasificado como del grupo 1, en realidad pertenecía al grupo 2 (de nuevo el 27).

Finalmente, accediendo a *P$posterior* podemos ver las **probabilidades a posteriori** (verosimilitudes normalizadas) de pertenencia a cada grupo bajo normalidad dadas por:

$$\Pr(i|\mathbf{z}) = \frac{\pi_i f_i(\mathbf{z})}{\pi_1 f_1(\mathbf{z}) + \pi_2 f_2(\mathbf{z})},$$

donde π_1 y π_2 son las probabilidades a priori (0.5 en este ejemplo) y f_1 y f_2 son las funciones de densidad normales obtenidas con los parámetros estimados en cada grupo. Para las primeras filas obtenemos:

```
## completar aquí
head(round(P$posterior, 2))
```

```
##         1    2
## [1,] 1.00 0.00
## [2,] 0.95 0.05
## [3,] 1.00 0.00
## [4,] 1.00 0.00
## [5,] 1.00 0.00
## [6,] 1.00 0.00
```

Podemos ver que las probabilidades de pertenencia para el escarabajo 40 valen $\Pr(1|z = e40) = 0.7531572$ y $\Pr(2|z = e40) = 0.2468428$, que nos muestran que para un individuo de estas medidas la clasificación no es muy fiable. Evidentemente, los individuos se clasifican usando LDA en el grupo en el que resultan más verosímiles (ambos métodos son equivalentes).

La función *predict* también se puede usar para clasificar a nuevos individuos. Por ejemplo, para clasificar a un escarabajo con medidas $\mathbf{z} = (185, 280, 150, 200)'$, haremos:

```
z <- c(185, 280, 150, 200)
predict(LDA, z)
```

```
## $class
## [1] 2
## Levels: 1 2
```

```
##
## $posterior
##               1         2
## [1,] 0.1872666 0.8127334
##
## $x
##            LD1
## [1,] 0.3965766
```

con lo que se obtiene que z se clasifica en el grupo 2, con una puntuación $D - K = 0.3965766$ y una probabilidad a posteriori de pertenencia al grupo 2 de 0.8127334. La clasificación de este escarabajo es un poco más fiable que la del escarabajo 40. Podemos comprobar que la puntuación coincide con $L(\mathbf{z}) - K$:

```
L(z) - K
```

```
## [1] 0.3965766
```

Los valores de la matriz de confusión se pueden usar para estimar las proporciones de acierto en cada caso. Por ejemplo, la probabilidad de acierto global estimada es $38/39 = 0.974359$. Estas estimaciones suelen dar valores ligeramente mayores que los valores reales ya que al clasificar a un individuo, se ha usado la información proporcionada por el propio individuo. Sin embargo, cuando se clasifica a un individuo nuevo (**e40**), éste no se usa en el procedimiento de clasificación. Para evitar esto, podemos usar la técnica de **validación cruzada LOOCV** (*leave-one-out cross validation*), que consiste en clasificar dejando fuera de la muestra cada observación individual. Cuando dispongamos de muestras de gran tamaño podemos dividir la muestra en conjuntos de entrenamiento y test, para medir la eficacia de nuestro método de clasificación. Para aplicar CV en R debemos incluir $CV = TRUE$ en la función *lda*:

```
LDACV <- lda(d[1:39, 1:4], d[1:39, 6], prior = c(0.5, 0.5), CV = TRUE)
table(d[1:39, 6], LDACV$class)
```

```
##
##      1  2
##   1 19  0
##   2  3 17
```

```
LDACV$class == d[1:39, 6]
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## [25]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## [37]  TRUE  TRUE  TRUE
```

De esta forma, podemos comprobar que hay 3 escarabajos del grupo 2 que se clasifican mal (21, 27 y 36) y el resumen correcto de clasificación sería el dado en la siguiente tabla:

Resumen	Clasificados en 1 (HO)	Clasificados en 2 (HC)	*Total*
Grupo verdadero: 1 (H0)	19	0	19
Grupo verdadero: 2 (HC)	3	17	20
Total	22	17	39

En ella comprobamos, por ejemplo, que la estimación (no sesgada) de la probabilidad global de acierto en este LDA es: $p = (19 + 17)/39 = 0.9230769$ (ligeramente menor que la calculada anteriormente sin CV). Al usar validación cruzada las probabilidades a posteriori de los individuos con grupos conocidos también cambian (ya que no se usan). Por ejemplo, para el escarabajo 21 obtenemos 0.5291374 y 0.4708626, mientras que antes eran 0.1606631 y 0.8393369. La validación cruzada no afecta a la clasificación de los individuos de los que se desconoce el grupo (ya que no se han usado para calcular L).

Tanto las probabilidades de pertenencia, como las puntuaciones (la constante K) y las clasificaciones finales se verán influenciadas por las probabilidades a priori. Por ejemplo, si no indicamos las probabilidades a priori (es decir, asumimos que éstas se calculen a partir de la muestra), para el escarabajo 40 se obtiene

```
## completar aquí
LDA_2 = lda(d[1:39, 1:4], d[1:39, 6])
predict(LDA_2, newdata = d[40, 1:4])
```

```
## $class
## [1] 1
## Levels: 1 2
##
## $posterior
##            1         2
## 40 0.7434978 0.2565022
##
## $x
##          LD1
## 40 -0.3488355
```

Esto es, una puntuación en $D-K$ de -0.34883551 y probabilidades a posteriori de $\Pr(1|e40) = 0.7434978$ y $\Pr(2|e40) = 0.2565022$, por lo que se sigue clasificando en el grupo 1. Los aciertos con estas probabilidades a priori son los mismos. Sin embargo, podemos comprobar que con las probabilidades a priori 0.2 y 0.8, existe un escarabajo (19) del grupo 1 que se clasifica en el 2 y que el escarabajo 40 se clasifica en el grupo 2.

```
## completar aquí
LDA_3 = lda(d[1:39, 1:4], d[1:39, 6], prior = c(0.2, 0.8))
predict(LDA_2, newdata = d[c(19, 40), 1:4])
```

```
## $class
## [1] 1 1
## Levels: 1 2
##
## $posterior
##            1         2
## 19 0.7737268 0.2262732
## 40 0.7434978 0.2565022
##
## $x
##          LD1
## 19 -0.3934805
## 40 -0.3488355
```

La clasificación será óptima cuando se usen las probabilidades de pertenencia reales en cada grupo (que suelen ser desconocidas).

Por último señalar que cuando se dispongan de $m > 2$ grupos, necesitaremos una función discriminante para distinguir entre cada pareja de grupos. En estos casos es mejor utilizar las probabilidades a posteriori y clasificarlo en donde sean máximas (que es un método equivalente).

3. Análisis Discriminante Cuadrático (QDA)

Cuando las variables usadas para clasificar sean normales (multivariantes) en cada grupo pero sus matrices de covarianzas (teóricas) no sean iguales, el procedimiento óptimo de clasificación consiste en comparar sus funciones de densidad (verosimilitudes o probabilidades a posteriori) estimadas mediante:

$$f_1(\mathbf{z}) = c \mid S_1 \mid^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{z}-\overline{\mathbf{X}})' S_1^{-1}(\mathbf{z}-\overline{\mathbf{X}})\right)$$
$$f_2(\mathbf{z}) = c \mid S_2 \mid^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{z}-\overline{\mathbf{Y}})' S_2^{-1}(\mathbf{z}-\overline{\mathbf{Y}})\right)$$

donde $c = (2\pi)^{-k/2}$.

En la sección anterior se estimaban usando la estimación de la matriz de varianzas común S. Note que ahora las matrices de covarianzas de cada grupo se estiman usando solo los datos de ese grupo con S_1 y S_2. Esto es equivalente a comparar las funciones discriminantes cuadráticas:

$$QDF_1(\mathbf{z}) = (\mathbf{z}-\overline{\mathbf{X}})' S_1^{-1}(\mathbf{z}-\overline{\mathbf{X}}) + \log \mid S_1 \mid \quad (1)$$
$$QDF_2(\mathbf{z}) = (\mathbf{z}-\overline{\mathbf{Y}})' S_2^{-1}(\mathbf{z}-\overline{\mathbf{Y}}) + \log \mid S_2 \mid \quad (2)$$

clasificando a un individuo en donde QDF sea mínima. Note que las funciones QDF son iguales a las distancias de Mahalanobis al cuadrado de cada grupo más una constante que depende del grupo. Cuando los determinantes sean iguales, el método será equivalente al de la distancia de Mahalanobis mínima.

Para realizar un QDA en R con los datos de los escarabajos incluidos en el objeto d debemos hacer:

```
QDA <- qda(d[1:39, 1:4], d[1:39, 6], prior = c(0.5, 0.5))
```

Comprobamos que en este procedimiento el objeto **QDA** no se incluyen los coeficientes de las QDF. Para obtener los coeficientes que convierten a los datos en esféricos debemos utilizar la instrucción:

```
## completar aquí
QDA$scaling
```

```
## , , 1
##
##                1           2           3           4
## surco 0.07301089 -0.07053717  0.01397441  0.04456840
## long  0.00000000  0.07481813  0.02436086  0.00916104
## base2 0.00000000  0.00000000 -0.14326815 -0.02396629
## base3 0.00000000  0.00000000  0.00000000 -0.07766942
##
## , , 2
##
##                 1           2           3            4
## surco -0.09909276 -0.08328527 -0.03045354 -0.009132788
## long   0.00000000  0.06623082  0.05430685 -0.006778529
## base2  0.00000000  0.00000000 -0.10417380 -0.024030738
## base3  0.00000000  0.00000000  0.00000000  0.082444581
```

Esta salida nos proporciona matrices triangulares U_i tales que $U_i U_i' = S_i^{-1}$. De esta forma las funciones discriminantes cuadráticas se pueden calcular como:

$$QDF_1(\mathbf{z}) = (U_1'\mathbf{z} - U_1'\overline{\mathbf{X}})'(U_1'\mathbf{z} - U_1'\overline{\mathbf{X}}) + \log \mid S_1 \mid \quad (3)$$
$$QDF_2(\mathbf{z}) = (U_2'\mathbf{z} - U_2'\overline{\mathbf{Y}})'(U_2'\mathbf{z} - U_2'\overline{\mathbf{Y}}) + \log \mid S_2 \mid, \quad (4)$$

es decir, la transformación $U_i'\mathbf{z}$ convierte a los datos del grupo i en esféricos ya que $Cov(U_i'\mathbf{z}) = U_i' S_i U_i$ y como $U_i U_i' = S_i^{-1}$, entonces $S_i = (U_i')^{-1} U_i^{-1}$ y

$$Cov(U_i'\mathbf{z}) = U_i' S_i U_i = U_i (U_i')^{-1} U_i^{-1} U_i = I.$$

Para obtener las constantes $\log|S_i|$ usamos:

```
## completar aquí
QDA$ldet
```

```
## [1] 19.41635 19.56726
```

comprobando que, efectivamente, $\log|S_1| = 19.41635$ y $\log|S_2| = 19.56726$.

Para obtener las predicciones basadas en estas funciones o, equivalentemente, en las probabilidades a posteriori, podemos utilizar como antes la función *predict*:

```
P <- predict(QDA, d[, 1:4])
```

Veamos ahora cuántos escarabajos hay mal clasificados:

```
## completar aquí
P$class == d[, 6]
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [25]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [37]  TRUE  TRUE  TRUE    NA
```

y comprobamos que solo hay un escarabajo mal clasificado (el 27). Accedemos a la clasificación del escarabajo 40 y a las probabilidades a posteriori:

```
## completar aquí
P$class[40]
```

```
## [1] 1
## Levels: 1 2
```

y con este procedimiento se ha clasificado en el grupo 1 (como en el LDA). En este caso, las **probabilidades** (verosimilitudes ponderadas) de pertenencia valen 0.5817418 y 0.4182582 por lo que esta clasificación no es nada fiable.

De nuevo podemos obtener una tabla resumen de las clasificaciones con:

```
## completar aquí
P$posterior[40, ]
```

```
##         1         2
## 0.5817418 0.4182582
```

Para que esta tabla sea más realista (no sea sesgada) debemos usar validación cruzada haciendo:

```
QDACV <- qda(d[1:39, 1:4], d[1:39, 6], prior = c(0.5, 0.5), CV = TRUE)
table(d[1:39, 6], QDACV$class)
```

```
##
##      1  2
##   1 17  2
##   2  2 18
```

De esta forma se obtienen los resultados de la siguiente tabla:

Resumen	Clasificados en 1 (HO)	Clasificados en 2 (HC)	*Total*
Grupo verdadero: 1 (H0)	17	2	19
Grupo verdadero: 2 (HC)	2	18	20
Total	19	20	39

Los resultados son similares a los obtenidos con el LDA (aquí hay un error más) con una probabilidad global de acierto (eficiencia) estimada de $\hat{p}_{(QDA)} = 35/39 = 0.8974359$.

De nuevo, la función *predict* se puede usar para clasificar a nuevos individuos. Por ejemplo, para clasificar a un escarabajo con medidas $z = (185, 280, 150, 200)$, haremos:

```
## completar aquí
z <- c(185, 280, 150, 200)
predict(QDA, z)
```

```
## $class
## [1] 2
## Levels: 1 2
##
## $posterior
##               1         2
## [1,] 0.03632458 0.9636754
```

con lo que se obtiene que **z** se clasifica en el grupo 2, con probabilidad a posteriori de pertenencia al grupo 2 de 0.9636754. Esta clasificación sí es fiable (bajo la hipótesis de normalidad).

4. Comprobaciones

En primer lugar podemos tener la duda de si es mejor aplicar LDA o QDA. El primer método funciona bien si las matrices de covarianzas teóricas son iguales y el segundo si los datos son normales en cada grupo. En todo caso, se cumplan o no esas hipótesis, el método de validación cruzada nos proporciona estimaciones de las probabilidades de acierto en cada caso y nos permite la comparación global de las técnicas LDA y QDA. También tenemos la opción de usar ambas técnicas y comprobar si los resultados coinciden.

Si queremos estudiar si los datos cumplen las hipótesis del LDA, la matriz de cuasicovarianzas del primer grupo se puede calcular con:

```
S1 <- cov(d[1:19, 1:4])
```

También se pueden separar los datos del grupo 1 con:

```
d1 <- d[d$especie == "HO", 1:4]
## d1 <- d %>%
##    filter(especie == "HO")
```

y su matriz de cuasicovarianzas se calcula como *cov(d1)*. Análogamente, se calcula la del segundo grupo

```
S2 <- cov(d[20:39, 1:4])
```

obteniéndose:

$$S_1 = \begin{pmatrix} 187.596 & 176.863 & 48.371 & 113.582 \\ 176.863 & 345.386 & 75.980 & 118.781 \\ 48.371 & 75.980 & 66.357 & 16.243 \\ 113.582 & 118.781 & 16.243 & 239.942 \end{pmatrix} \text{ y } S_2 = \begin{pmatrix} 101.839 & 128.063 & 36.989 & 32.592 \\ 128.063 & 389.011 & 165.358 & 94.368 \\ 36.989 & 165.358 & 167.537 & 66.526 \\ 32.592 & 94.368 & 66.526 & 177.882 \end{pmatrix}$$

De esta forma, comprobamos que las matrices de covarianzas muestrales de los grupos son bastante diferentes y que no parecen estimaciones de una misma matriz V (incluso teniendo en cuenta que los tamaños muestrales son pequeños).

Antes de aplicar un QDA debemos comprobar si los datos son normales. Para hacer un test de normalidad multivariante (Shapiro-Wilk) debemos cargar el paquete *mvnormtest* y aplicar la función:

```r
library("mvnormtest")
```

```
## Warning: package 'mvnormtest' was built under R version 4.3.3
```

```r
mshapiro.test(t(d[1:19, 1:4]))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Z
## W = 0.93356, p-value = 0.2013
```

obteniendo un p-valor de 0.2013, por lo que el primer grupo pasaría el test de normalidad. Análogamente, para el segundo

```r
## completar aquí
mshapiro.test(t(d[20:39, 1:4]))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Z
## W = 0.90772, p-value = 0.05769
```

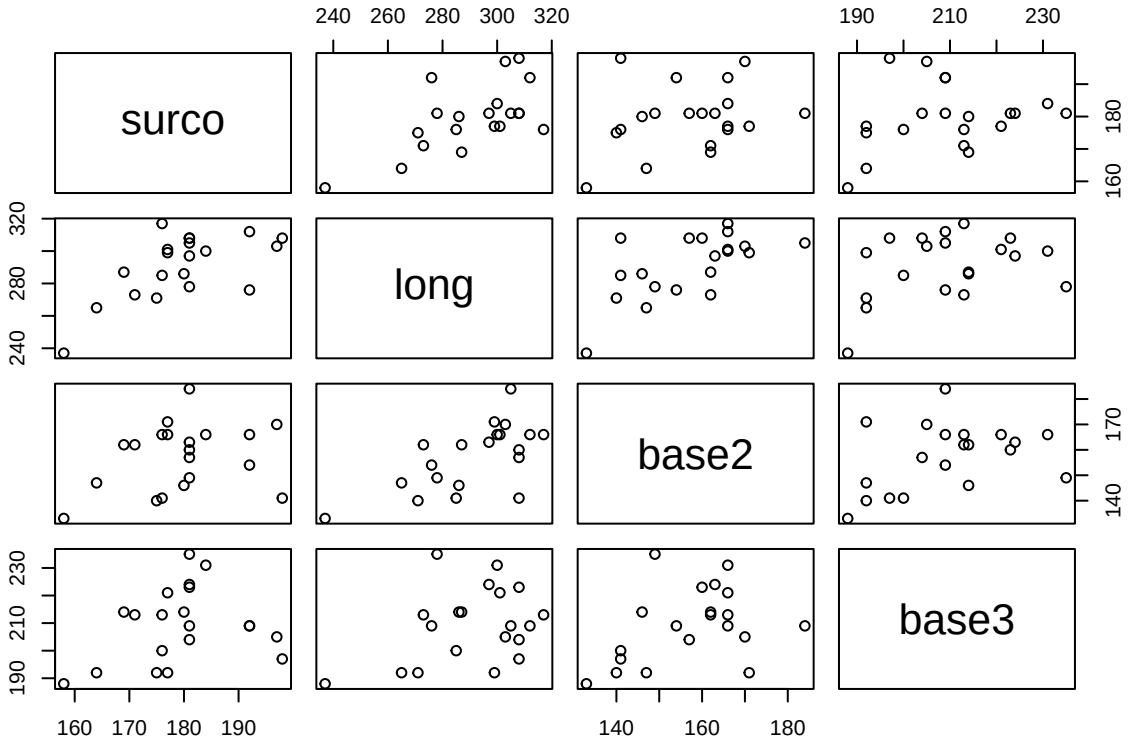
se obtiene un p-valor de 0.05769, que nos conduciría a aceptar la normalidad con $\alpha = 0.05$ por muy poco. Esto se puede deber al escarabajo 27 que, como hemos visto durante toda la práctica tiene unas medidas raras para ser del grupo 2. Sin embargo, al eliminarlo, el test no mejora.

```r
## completar aquí
mshapiro.test(t(d[c(20:26, 28:39), 1:4]))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Z
## W = 0.88566, p-value = 0.02697
```

Los datos para este grupo se pueden ver haciendo:

```r
## completar aquí
plot(d[20:39, 1:4])
```



Para comprobar que las computaciones de R para los coeficientes del LDA son correctas podemos calcular la estimación de la matriz de covarianzas común V con

$$S = \frac{1}{n_1 + n_2 - 2}[(n_1 - 1)S_1 + (n_2 - 1)S_2].$$

Para ello, utilizamos la instrucción:

```
S <-(18*S1 + 19*S2)/37
```

y obtenemos

$$S = \begin{pmatrix} 143.559 & 151.803 & 42.527 & 71.993 \\ 151.803 & 367.788 & 121.877 & 106.245 \\ 42.527 & 121.877 & 118.314 & 42.064 \\ 71.993 & 106.245 & 42.064 & 208.073 \end{pmatrix}.$$

Esta estimación no depende de las probabilidades a priori (se da mayor peso a las estimaciones con más datos). Recordemos que las medias de los grupos se pueden calcular con:

```
m1 <- LDA$means[1, ]
m2 <- LDA$means[2,]
## colMeans(d1) #Otra forma
```

y los coeficientes como

```
a2 <- (m1 - m2) %*% solve(S)
a2
```

```
##          surco       long      base2      base3
## [1,] 0.345249 -0.1303878 -0.1064338 -0.1433533
```

(donde %*% denota el producto de matrices en R) obteniendo

$$\mathbf{a}' = (0.345249, -0.1303878, -0.1064338, -0.1433533).$$

Para comprobar que son proporcionales a los obtenidos por R haremos:

```
LDA$scaling/t(a2)
```

```
##              LD1
## surco -0.2701715
## long  -0.2701715
## base2 -0.2701715
## base3 -0.2701715
```

donde **t(a2)** es el traspuesto de **a2**. Así obtenemos que en este ejemplo R usa $\lambda = -0.2701715$.

Si queremos estudiar qué variables influyen más en los procedimientos de clasificación LDA, como las variables originales pueden tener escalas diferentes (como ocurre en nuestro ejemplo), no podemos comparar directamente los coeficientes obtenidos con ellas. Sin embargo, si estandarizamos las variables originales, como éstas tendrán valores similares, los coeficientes obtenidos con ellas en el LDA sí se podrán usar para estudiar la influencia de las variables en la clasificación. Al contrario de lo que ocurría en el PCA, los procedimientos de clasificación LDA y QDA dan el mismo resultado si se usan las variables estandarizadas (no se ven afectados por cambios de escala y/o localización). Para estandarizar los datos haremos:

```
ds <- scale(d[, 1:4])
```

y calculando los coeficientes con:

```
LDAds = lda(x = ds[1:39, 1:4], grouping = d[1:39, 6], prior = c(0.5, 0.5))
```

obtenemos:

```
## completar aquí
LDAds$scaling
```

```
##              LD1
## surco -1.2937164
## long   0.7809833
## base2  0.4182667
## base3  0.7084167
```

Por lo tanto, la variable que más influye (mejor discrimina) es *surco* y la que menos *base2*.

También nos podemos plantear si podemos eliminar alguna variable y cuál sería la más adecuada. Para esto podemos usar los procedimientos de validación cruzada y estudiar qué opción proporciona los mejores resultados teniendo claro que la mejor opción es siempre usarlas todas. Por ejemplo, si eliminamos *surco* haciendo:

```
LDA_sin_surco <- lda(d[1:39, 2:4], d[1:39, 6], prior = c(0.5, 0.5), CV = TRUE)
LDA_sin_surco$class == d[1:39, 6]
```

```
##  [1]  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## [25]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE
## [37]  TRUE FALSE  TRUE
```

```
sum(LDA_sin_surco$class != d[1:39, 6])
```

```
## [1] 7
```

comprobamos que hay 7 escarabajos que se clasifican mal. Eliminado las otras variables

```
LDA_sin_long <- lda(d[1:39, c(1,3:4)], d[1:39, 6], prior = c(0.5, 0.5), CV = TRUE)
LDA_sin_long$class == d[1:39, 6]
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [25]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [37]  TRUE  TRUE  TRUE
```

```
sum(LDA_sin_long$class != d[1:39, 6])
```

```
## [1] 2
```

```
LDA_sin_base2<- lda(d[1:39, c(1:2, 4)], d[1:39, 6], prior = c(0.5, 0.5), CV = TRUE)
LDA_sin_base2$class == d[1:39, 6]
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [25]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## [37]  TRUE  TRUE  TRUE
```

```
sum(LDA_sin_base2$class != d[1:39, 6])
```

```
## [1] 2
```

```
LDA_sin_base3 <- lda(d[1:39, 1:3], d[1:39, 6], prior = c(0.5, 0.5), CV = TRUE)
LDA_sin_base3$class == d[1:39, 6]
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## [25]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
## [37]  TRUE  TRUE  TRUE
```

```
sum(LDA_sin_base3$class != d[1:39, 6])
```

```
## [1] 3
```

comprobamos que las mejores opciones son eliminar la variable *long* o la variable *base*2 (en ambos casos solo hay 2 escarabajos que se clasifiquen mal). Análogamente, podemos estudiar cuál es la mejor pareja de variables (o la variable) que mejor discriminan. Se puede aplicar un procedimiento similar en el QDA.

También podemos comprobar cómo se calculan las probabilidades a posteriori. Para ello debemos cargar el paquete *mvtnorm* e incluir las siguientes instrucciones:

```
library("mvtnorm")
f1 <- dmvnorm(d[40, 1:4], m1, S)
f2 <- dmvnorm(d[40, 1:4], m2, S)
f1/(f1+f2)
```

```
##        40
## 0.7531572
```

De esta forma se obtiene la probabilidad a posteriori del escarabajo 40 en el grupo 1, $\Pr(1|z = e40) = 0.7531572$ (como en la sección 2), en el caso de probabilidades a priori iguales. Para obtener la que se obtiene con las probabilidades a priori proporcionadas por los grupos debemos hacer:

```
19*f1/(19*f1+20*f2)
```

```
##        40
## 0.7434978
```

obteniendo $\Pr(1|e40) = 0.743497$.

Por último, señalar que para que estas **probabilidades** (verosimilitudes) sean correctas, las variables deben ser normales multivariantes en cada grupo. Esta hipótesis también se usa en el QDA.

Cuando en un LDA hay más de dos grupos, algunos autores prefieren calcular las funciones discriminantes lineales por grupos dadas por:

$$L_i(\mathbf{z}) = \mathbf{m}_i' S^{-1}\mathbf{z} - \mathbf{m}_i' S^{-1}\mathbf{m}_i/2,$$

donde S es la matriz de covarianzas ponderada (calculada anteriormente) y $\mathbf{m}_i$ son las medias muestrales de los grupos. Para calcularlas en R haremos:

```
## completar aquí
t(m1) %*% solve(S)
```

```
##          surco        long     base2     base3
## [1,] 0.9557217 -0.02086224 0.6842504 0.4353125
```

```
t(m2) %*% solve(S)
```

```
##          surco      long     base2     base3
## [1,] 0.6104728 0.1095255 0.7906842 0.5786658
```

```
-05*t(m1) %*% solve(S) %*% m1
```

```
##           [,1]
## [1,] -1776.155
```

obteniendo:

$$L_1(\mathbf{z}) = 0.9557217z_1 - 0.0208622z_2 + 0.6842504z_3 + 0.4353125z_4 - 177.6155$$

y

$$L_2(\mathbf{z}) = 0.6104728z_1 + 0.1095255z_2 + 0.7906842z_3 + 0.5786658z_4 - 193.4209$$

Los individuos se clasificarán en el grupo con valor máximo de estas funciones. Este método es equivalente al de máxima verosimilitud (probabilidad a posteriori) con matrices de covarianzas iguales por lo que se obtendrán los mismos resultados que en la sección 2. También es equivalente a usar las funciones discriminantes de Fisher paso a paso. De hecho, estas se obtienen restando las funciones discriminates de los grupos, es decir: $L_1(\mathbf{z}) - L_2(\mathbf{z}) = \mathbf{a}'\mathbf{z} - K$. Por ejemplo, para el escarabajo 40 obtenemos:

$$L_1(182.22, 271.01, 140.99, 190.15) = 170.1294$$

y

$$L_2(182.22, 271.01, 140.99, 190.15) = 169.0138$$

por lo que se clasificaría en el grupo 1 (HO).

De forma análoga, en el QDA se pueden calcular las funciones cuadráticas definidas previamente (QDF). En este caso, los individuos se incluyen en el grupo con el valor mínimo para esas funciones. Esto es equivalente a usar el método de máxima verosimilitud (probabilidad a posteriori) con matrices de covarianzas distintas por lo que se obtendrán las mismas clasificaciones que en la sección 3. Para el escarabajo 40 se obtiene:

$$QDF_1(\mathbf{z}) = (\mathbf{z} - \overline{\mathbf{x}})' S_1^{-1}(\mathbf{z} - \overline{\mathbf{x}}) + \log|S_1| = 22.76789$$

y

$$QDF_2(\mathbf{z}) = (\mathbf{z} - \overline{\mathbf{y}})' S_2^{-1}(\mathbf{z} - \overline{\mathbf{y}}) + \log|S_2| = 23.42774$$

por lo que se clasificaría (de nuevo) en el grupo 1.

De forma similar se pueden calcular las distancias de Mahalanobis (al cuadrado) dadas en el QDA por:

$$D_1^2(z) = (\mathbf{z} - \overline{\mathbf{x}})' S_1^{-1} (\mathbf{z} - \overline{\mathbf{x}}) \text{ y } D_2^2(\mathbf{z}) = (\mathbf{z} - \overline{\mathbf{y}})' S_2^{-1} (\mathbf{z} - \overline{\mathbf{y}})$$

obteniendo para el escarabajo 40: $D_1^2(\mathbf{z}) = 3.351539$ y $D_2^2(\mathbf{z}) = 3.860477$ por lo que se clasificaría en el grupo 1 (en el más cercano). También se puede usar el comando *mahalanobis(x, y, V)*. Los métodos de clasificación son equivalentes si los determinantes de las matrices de covarianzas de los grupos son iguales. Por lo tanto, se pueden obtener resultados diferentes de los obtenidos en la sección 3 si usamos esta distancia para clasificar. Los valores del QDA también se pueden obtener con:

```
## completar aquí
e40 = as.numeric(d[40, 1:4])
M1 <- mahalanobis(m1, e40, S1)
M2 <- mahalanobis(m1, e40, S2)
QDA$ldet + c(M1, M2)
```

```
## [1] 22.76789 22.85001
```