

Análisis discriminante Soluciones

Francesc Carmona y Josep Gregori*

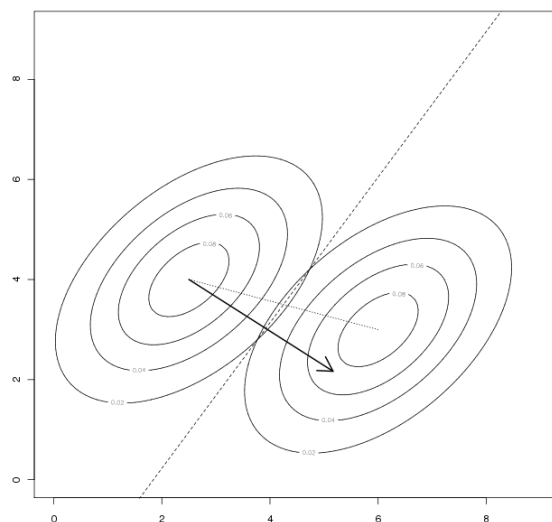
9 de junio de 2022

Ejercicio 1

Consideremos dos variables aleatorias (X_1, X_2) con distribución normal bivalente $N_2(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ para los elementos de una población y con distribución $N_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$ para los elementos de otra población. Los vectores de medias son $\boldsymbol{\mu}_1 = (2.5, 4)$ y $\boldsymbol{\mu}_2 = (6, 3)$ respectivamente, pero las dos poblaciones tienen la misma matriz de covarianzas

$$\boldsymbol{\Sigma} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

La siguiente imagen¹ nos puede ayudar



a) Hallar el discriminador lineal de Fisher

$$L(\mathbf{x}) = \mathbf{a}'\mathbf{x} - \mathbf{a}'\boldsymbol{\mu}_c = \mathbf{a}'(\mathbf{x} - \boldsymbol{\mu}_c)$$

donde $\boldsymbol{\mu}_c = (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2$, que asigna \mathbf{x} a una de las dos poblaciones según su signo.

Los datos son:

* Alumno del curso 2009-10

¹ El código para dibujarla, así como otros detalles, se hallan en el artículo

<http://erre-que-erre-paco.blogspot.com/2010/05/discriminador-lineal-de-fisher.html>

```
> Sigma <- matrix(c(2,1,1,2),nrow=2)
> mu1 <- c(2.5,4)
> mu2 <- c(6,3)
> mu.c <- (mu1+mu2)/2
```

Los coeficientes de discriminador lineal son:

```
> (a <- solve(Sigma) %*% (mu1-mu2))

      [,1]
[1,] -2.666667
[2,]  1.833333
```

y el término independiente:

```
> (k <- t(a) %*% mu.c)

      [,1]
[1,] -4.916667
```

Por todo ello el discriminador lineal de Fisher será:

$$L(x_1, x_2) = -2.6667x_1 + 1.8333x_2 + 4.9167$$

En otras soluciones, los coeficientes deben ser proporcionales a estos.

- b) *Comprobar matricialmente que $L(\mathbf{x})$ es básicamente la diferencia entre los cuadrados de las distancias de Mahalanobis del punto \mathbf{x} a las medias:*

$$L(\mathbf{x}) = \frac{1}{2} (d_M^2(\mathbf{x}, \boldsymbol{\mu}_2) - d_M^2(\mathbf{x}, \boldsymbol{\mu}_1))$$

Luego el criterio de Mahalanobis de asignar un punto a la población más cercana a su media coincide con el criterio de Fisher.

Hagamos el cálculo:

$$\begin{aligned} L(\mathbf{x}) &= \frac{1}{2} (d_M^2(\mathbf{x}, \boldsymbol{\mu}_2) - d_M^2(\mathbf{x}, \boldsymbol{\mu}_1)) \\ &= \frac{1}{2} ((\mathbf{x} - \boldsymbol{\mu}_2)' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_2) - (\mathbf{x} - \boldsymbol{\mu}_1)' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_1)) \\ &= \frac{1}{2} (\mathbf{x}' \boldsymbol{\Sigma}^{-1} \mathbf{x} - \boldsymbol{\mu}_2' \boldsymbol{\Sigma}^{-1} \mathbf{x} - \mathbf{x}' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 + \boldsymbol{\mu}_2' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 - \mathbf{x}' \boldsymbol{\Sigma}^{-1} \mathbf{x} + \boldsymbol{\mu}_1' \boldsymbol{\Sigma}^{-1} \mathbf{x} + \mathbf{x}' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 - \boldsymbol{\mu}_1' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1) \\ &= \frac{1}{2} (-2\boldsymbol{\mu}_2' \boldsymbol{\Sigma}^{-1} \mathbf{x} + 2\boldsymbol{\mu}_1' \boldsymbol{\Sigma}^{-1} \mathbf{x} + \boldsymbol{\mu}_2' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1) \\ &= (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} (\boldsymbol{\mu}_1' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2) \end{aligned}$$

De modo que los coeficientes son los del discriminador lineal de Fisher. Veamos si también coincide el término independiente.

$$\begin{aligned} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \boldsymbol{\Sigma}^{-1} \left(\frac{\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2}{2} \right) &= \frac{1}{2} (\boldsymbol{\mu}_1' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 + \boldsymbol{\mu}_1' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 - \boldsymbol{\mu}_2' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2) \\ &= \frac{1}{2} (\boldsymbol{\mu}_1' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2) \end{aligned}$$

Así pues, el criterio de Mahalanobis coincide con el criterio de Fisher.

Ejercicio 2

En el archivo `wine.data` se recogen los resultados de un análisis químico del vino criado en la misma región italiana pero de tres viticultores distintos. El análisis determinó las cantidades de 13 constituyentes hallados en cada uno de los tres tipos de vino.

```
> archivo <-
+   "http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data"
> vinos <- read.csv(archivo,header=F)
> names(vinos)<- c("tipo",paste("X",1:13,sep=""))
> vinos$tipo <- as.factor(vinos[,1])
> attach(vinos)
```

Considerando únicamente los tipos de vino 1 y 2 y las variables X_1 (Alcohol) y X_{13} (Proline),

```
> vinos0 <- vinos[vinos$tipo == 1 | vinos$tipo==2, c("tipo","X1","X13")]
> vinos0$tipo <- as.numeric(vinos0$tipo)
```

- a) Expresar la regla de clasificación lineal de Fisher para una nueva observación (x_1, x_{13}) .

El discriminador lineal de Fisher es

$$L(\mathbf{x}) = (\mathbf{x} - \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2))' \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

y la regla de clasificación según este discriminador:

Si $L(\mathbf{x}) > 0$ asignar ω a Ω_1
 en otro caso asignar ω a Ω_2

- b) Aplicar la regla anterior al caso concreto en que $(x_1, x_{13}) = (13.05, 515)$. ¿A qué tipo de vino corresponde?

El número de vinos de cada tipo es

```
> n1 <- sum(vinos0$tipo==1); n1
[1] 59

> n2 <- sum(vinos0$tipo==2); n2
[1] 71
```

Los vectores de medias de las dos poblaciones

```
> mu1 <- apply(vinos0[vinos0$tipo==1, 2:3], 2, mean); mu1

      X1      X13
13.74475 1115.71186

> mu2 <- apply(vinos0[vinos0$tipo==2, 2:3], 2, mean); mu2

      X1      X13
12.27873 519.50704
```

La matriz de varianzas-covarianzas de cada población

```
> S1 <- cov(vinos0[vinos0$tipo==1, 2:3]); S1
```

```
      X1      X13
X1  0.2135598  36.91949
X13 36.9194944 49071.45003
```

```
> S2 <- cov(vinos0[vinos0$tipo==2, 2:3]); S2
```

```
      X1      X13
X1  0.2894055   3.651366
X13 3.6513662 24715.367807
```

La matriz de covarianzas común es

```
> S <- ((n1-1)*S1+(n2-1)*S2)/(n1+n2-2); S
```

```
      X1      X13
X1  0.2550379  18.72599
X13 18.7259868 35751.71756
```

El punto a clasificar

```
> x <- c(13.05,515)
```

El discriminador lineal de Fisher es

```
> L.x <- as.numeric( t(x-(1/2)*(mu1+mu2)) %*% solve(S) %*% (mu1-mu2) ); L.x
```

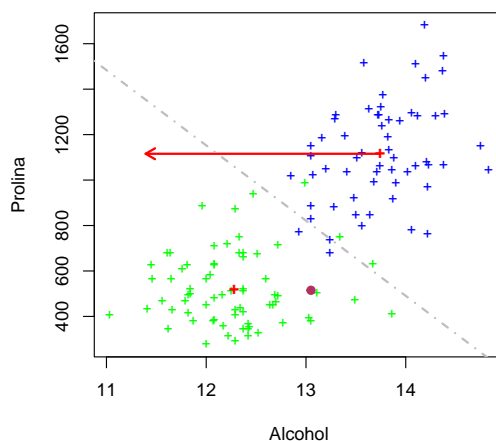
```
[1] -4.120689
```

luego el nuevo punto se clasifica como tipo 2 ya que el resultado no es positivo.

c) Representar gráficamente los puntos de las dos poblaciones y la recta equidistante de las medias².

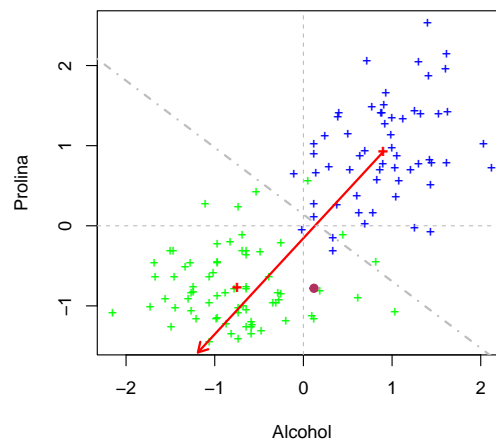
```
> plot(vinos0[,2],vinos0[,3],pch="+",xlab="Alcohol",ylab="Prolina",cex=0.8,
+      col=c("blue","green")[vinos0$tipo])
> text(mu1[1],mu1[2],"+",col="red",font=2,cex=1)
> text(mu2[1],mu2[2],"+",col="red",font=2,cex=1)
> # Vector del discriminador
> w <- solve(S)%*%(mu2-mu1)
> arrows(mu1[1],mu1[2],mu1[1]+w[1]/2,mu1[2]+w[2]/2,lwd=2,length=0.1,col="red")
> # Recta frontera
> pm <- (mu1+mu2)/2
> a <- w[1]*pm[1]/w[2]+pm[2]
> b <- -w[1]/w[2]
> abline(a,b,lty=4,col="gray",lwd=2)
> # Punto a clasificar
> points(x[1],x[2],pch=19,col="maroon",font=2,cex=1)
```

²Es la recta perpendicular al discriminador lineal de Fisher y que pasa por el punto medio entre los dos puntos medios de las poblaciones.



Notemos que en el gráfico la frontera no aparece perpendicular al vector director. Esto se debe a la gran diferencia de escala entre los dos ejes. Si el gráfico se dibuja proporcional todos los puntos aparecen en una estrecha franja vertical que no permite distinguir nada de nada. Para resolver este asunto se pueden centrar y escalar las variables como haremos a continuación:

```
> vinos0cs <- scale(vinos0[,2:3])
> mucs1 <- apply(vinos0cs[vinos0$tipo==1,],2,mean)
> mucs2 <- apply(vinos0cs[vinos0$tipo==2,],2,mean)
> Scs1 <- cov(vinos0cs[vinos0$tipo==1,])
> Scs2 <- cov(vinos0cs[vinos0$tipo==2,])
> Scs <- ((n1-1)*Scs1+(n2-1)*Scs2)/(n1+n2-2)
> # Representación gráfica de los puntos
> plot(vinos0cs,pch="+",xlab="Alcohol",ylab="Prolina",cex=0.8,
+      col=c("blue","green")[vinos$tipo])
> text(mucs1[1],mucs1[2],"+",col="red",font=2,cex=1)
> text(mucs2[1],mucs2[2],"+",col="red",font=2,cex=1)
> # Vector del discriminador
> wcs <- solve(Scs)%*(mucs2-mucs1)
> arrows(mucs1[1],mucs1[2],mucs1[1]+wcs[1]/2,mucs1[2]+wcs[2]/2,
+       lwd=2,length=0.1,col="red")
> # Recta frontera
> pm <- (mucs1+mucs2)/2
> a <- wcs[1]*pm[1]/wcs[2]+pm[2]
> b <- -wcs[1]/wcs[2]
> abline(a,b,lty=4,col="gray",lwd=2)
> abline(h=0,v=0,lty=2,col="gray")
> # Punto a clasificar
> xsc <- (x-attr(vinos0cs,"scaled:center"))/ attr(vinos0cs,"scaled:scale")
> points(xsc[1],xsc[2],pch=19,col="maroon",cex=1)
```



Ahora sí queda claro que son perpendiculares.

Este último gráfico se ha hecho con fines didácticos y, en general, no es necesario centrar y escalar los datos.

d) Probar la función `lda()` del paquete *MASS* para este conjunto restringido de datos.

```
> library(MASS)
> dis <- lda(vinos0[,2:3], grouping=vinos0$tipo); dis
```

Call:
`lda(vinos0[, 2:3], grouping = vinos0$tipo)`

Prior probabilities of groups:

	1	2
	0.4538462	0.5461538

Group means:

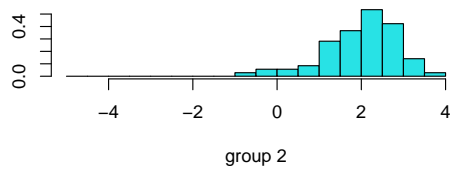
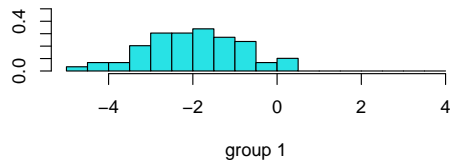
	X1	X13
1	13.74475	1115.712
2	12.27873	519.507

Coefficients of linear discriminants:

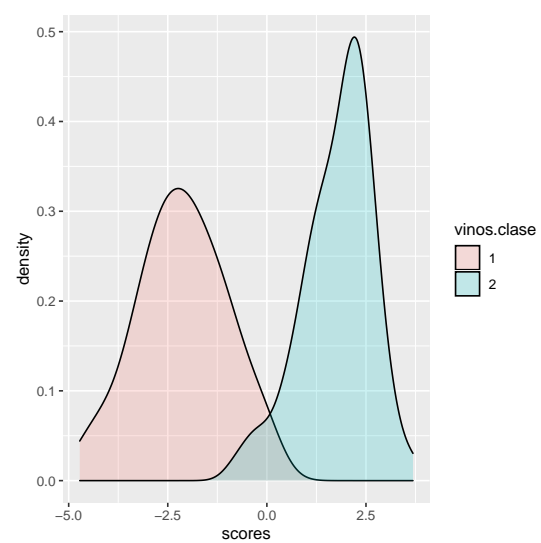
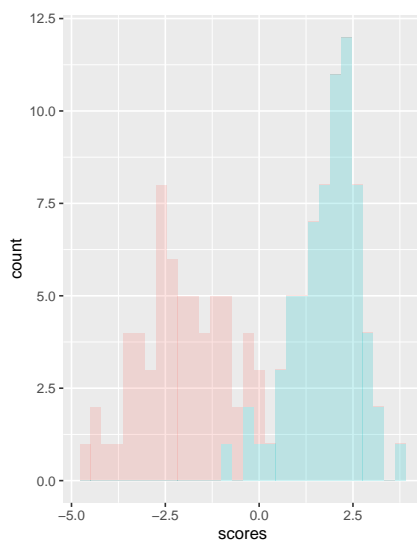
	LD1
X1	-1.200022623
X13	-0.003625041

e) Probar la función `plot()` sobre el objeto resultado de la función `lda()`. Obtener el gráfico con histogramas y con densidades (mejor con `ggplot2`).

```
> plot(dis)
```



```
> vinos.clase <- factor(vinos0$tipo)
> library(ggplot2)
> ggplot(vinos0, aes(predict(dis)$x, fill = vinos.clase)) +
+   geom_histogram(alpha = 0.2) + xlab("scores")
> ggplot(vinos0, aes(predict(dis)$x, fill = vinos.clase)) +
+   geom_density(alpha = 0.2) + xlab("scores")
```



f) Probar la función `predict()` para la observación $(x_1, x_{13}) = (13.05, 515)$.

```
> x

[1] 13.05 515.00

> predict(dis, newdata=x)

$class
[1] 2
```

```

Levels: 1 2

$posterior
      1      2
[1,] 0.01331012 0.9866899

$x
      LD1
[1,] 0.8701108

```

- g) Obtener la tabla de clasificación, también llamada tabla de confusión, donde se cruzan los datos reales con las clases obtenidas.

Calcular el error o *misclassification rate*

$$\text{Error} = 1 - \frac{\text{Total bien clasificados}}{\text{Total}} = 1 - \frac{\sum n_{ii}}{n}$$

```

> tt <- table(class=predict(dis)$class, true=vinos0$tipo); tt

      true
class  1  2
  1  56  3
  2   3 68

> # Error de clasificación
> 1 - sum(diag(tt))/sum(tt)

[1] 0.04615385

```

- h) Obtener la tabla de validación cruzada mediante el parámetro *CV = TRUE* que genera las predicciones por el sistema *leave one out*.

¡Atención! Cuando se utiliza este parámetro el objeto que resulta no es un objeto *lda*, lo que provoca que no se pueda utilizar con las funciones *plot()* y *predict()* asociadas a dichos objetos.

```

> # Validación cruzada per jackknife
> dis.CV <- lda(vinos0[,2:3],grouping=vinos0$tipo, CV=TRUE)
> tt <- table(class=dis.CV$class, true=vinos0$tipo); tt

      true
class  1  2
  1  56  3
  2   3 68

> # Error de clasificación
> 1 - sum(diag(tt))/sum(tt)

[1] 0.04615385

```

Con estos datos el error de clasificación es el mismo con el método de clasificación directa o por validación cruzada *jackknife*.

Ejercicio 3

Consideremos los datos sobre cráneos de varones egipcios de cinco épocas históricas que se pueden obtener en el siguiente enlace:

<http://lib.stat.cmu.edu/DASL/Datafiles/EgyptianSkulls.html>

También podemos bajarlos desde la página del libro de Everitt(2005) y así los podremos cargar directamente en **R** con las siguientes instrucciones:

```
> skulls <- source("chap5skulls.dat")$value
> str(skulls)

'data.frame': 150 obs. of 5 variables:
 $ EPOCH: Factor w/ 5 levels "c4000BC","c3300BC",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ MB : num 131 125 131 119 136 138 139 125 131 134 ...
 $ BH : num 138 131 132 132 143 137 130 136 134 134 ...
 $ BL : num 89 92 99 96 100 89 108 93 102 99 ...
 $ NH : num 49 48 50 44 54 56 48 48 51 51 ...

> attach(skulls)
```

Así tendremos la base de datos *skulls* con cinco variables. La primera variable es el factor *EPOCH* y las otras cuatro son las medidas biométricas estudiadas del cráneo.

- a) En primer lugar podemos realizar un MANOVA para contrastar la diferencia de medias entre los niveles del factor o poblaciones. No entraremos aquí en la comprobación de las hipótesis de normalidad y de igualdad de las matrices de covarianzas.

Realizar un test de Wilks.

```
> skulls.manova <- manova(cbind(MB,BH,BL,NH)~EPOCH, data=skulls)
> summary.aov(skulls.manova)

Response MB :
          Df Sum Sq Mean Sq F value    Pr(>F)
EPOCH         4  502.83  125.707   5.9546 0.0001826 ***
Residuals    145 3061.07   21.111
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Response BH :
          Df Sum Sq Mean Sq F value    Pr(>F)
EPOCH         4  229.9   57.477   2.4474 0.04897 *
Residuals    145 3405.3   23.485
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Response BL :
          Df Sum Sq Mean Sq F value    Pr(>F)
EPOCH         4  803.3  200.823   8.3057 4.636e-06 ***
Residuals    145 3506.0   24.179
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Response NH :
```

```

      Df Sum Sq Mean Sq F value Pr(>F)
EPOCH      4   61.2   15.300    1.507 0.2032
Residuals 145 1472.1   10.153

> summary(skulls.manova, test="Wilks") # test="Pillai" or "Hotelling" or "Roy"

      Df  Wilks approx F num Df den Df  Pr(>F)
EPOCH      4 0.66359    3.9009    16 434.45 7.01e-07 ***
Residuals 145
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

El test rechaza la igualdad de medias y, por lo tanto, justifica el análisis discriminante.

- b) Realizar un análisis lineal discriminante con la función `lda()` del paquete *MASS*.

Obtener los vectores de medias para las distintas poblaciones y los coeficientes de las variables canónicas.

Observar que con el nombre de *Coefficients of linear discriminants*³ justamente se obtienen los coeficientes de las variables canónicas. Éstos están normalizados respecto a la matriz **S** de covarianzas dentro de los grupos.

```

> dis <- lda(skulls[,2:5],grouping=skulls[,1]); dis

Call:
lda(skulls[, 2:5], grouping = skulls[, 1])

Prior probabilities of groups:
c4000BC c3300BC c1850BC c200BC cAD150
      0.2      0.2      0.2      0.2      0.2

Group means:
      MB      BH      BL      NH
c4000BC 131.3667 133.6000 99.16667 50.53333
c3300BC 132.3667 132.7000 99.06667 50.23333
c1850BC 134.4667 133.8000 96.03333 50.56667
c200BC  135.5000 132.3000 94.53333 51.96667
cAD150  136.1667 130.3333 93.50000 51.36667

Coefficients of linear discriminants:
      LD1      LD2      LD3      LD4
MB  0.12667629 0.03873784 0.09276835 0.1488398644
BH -0.03703209 0.21009773 -0.02456846 -0.0004200843
BL -0.14512512 -0.06811443 0.01474860 0.1325007670
NH  0.08285128 -0.07729281 -0.29458931 0.0668588797

Proportion of trace:
      LD1      LD2      LD3      LD4
0.8823 0.0809 0.0326 0.0042

```

Vamos a ver que estos valores son también los que se obtienen directamente del análisis canónico de poblaciones.

³Se obtienen con el valor `scaling`.

Necesitamos una función que calcule el análisis canónico de poblaciones. Esta función utiliza una función para calcular la diagonalización generalizada⁴:

```
> # H:operador S:métrica
> gen.eigen <- function(H,S){
+   T <- chol(S)
+   Tinv <- solve(T)
+   evr <- eigen( t(Tinv) %*% H %*% Tinv )
+   TV <- evr$vectors
+   V2 <- Tinv %*% TV
+   list(values=evr[[1]], vectors=evr[[2]],TinvV=V2)
+ }
```

Función de análisis canónico:

```
> # data (matrix) gfact (factor)
> anal.canonico <- function(data,gfact){
+   nT <- nrow(data)
+   p <- ncol(data)
+   g <- nlevels(gfact)
+   ni <- tapply(gfact,gfact,length)
+   # Vector de medias global
+   gmean <- apply(data,2,mean)
+   # Medias de los grupos
+   means <- aggregate(data,by=list(gfact),mean)
+   means <- as.matrix(means[,-1])
+   rownames(means) <- levels(gfact)
+   # Matriz de covarianzas común ponderada
+   S <- (nT-1)*var(data-means[gfact,])/(nT-g)
+   # Matriz de medias de los grupos centrada
+   M <- means-matrix(gmean,nrow=g,ncol=p,byrow=TRUE)
+   # Diagonalización en la métrica de S
+   geo <- gen.eigen(t(M)%*%M,S)
+   list(values=geo$values, TV=geo$vectors, V=geo$TinvV,
+         means=means, Y=means%*%geo$TinvV)
+ }
```

Ahora la aplicamos a los datos:

```
> aco <- anal.canonico(as.matrix(skulls[,2:5]),skulls$EPOCH); aco

$values
[1] 2.054627647 0.188494673 0.075904888 0.009765007

$TV
      [,1]      [,2]      [,3]      [,4]
[1,] 0.6154648 0.1447008 -0.2974889 0.7153776
[2,] -0.2865995 0.8997428 0.2759714 0.1793414
[3,] -0.6878362 -0.3348596 -0.0402855 0.6427500
[4,] 0.2567980 -0.2395695 0.9130812 0.2072295

$V
```

⁴Se puede utilizar la función `geigen()` del paquete `geigen` con el parámetro `symmetric = TRUE`.

```

      [,1]      [,2]      [,3]      [,4]
MB  0.12667629  0.03873784 -0.09276835  0.1488398644
BH -0.03703209  0.21009773  0.02456846 -0.0004200843
BL -0.14512512 -0.06811443 -0.01474860  0.1325007670
NH  0.08285128 -0.07729281  0.29458931  0.0668588797

$means
      MB      BH      BL      NH
c4000BC 131.3667 133.6000 99.16667 50.53333
c3300BC 132.3667 132.7000 99.06667 50.23333
c1850BC 134.4667 133.8000 96.03333 50.56667
c200BC  135.5000 132.3000 94.53333 51.96667
cAD150  136.1667 130.3333 93.50000 51.36667

$Y
      [,1]      [,2]      [,3]      [,4]
c4000BC 1.488731 22.49737 4.519688 36.01474
c3300BC 1.638393 22.37702 4.317907 36.13065
c1850BC 2.331508 22.87033 4.293052 36.06311
c200BC  2.851635 22.58917 4.594887 36.11240
cAD150  3.109167 22.31857 4.323210 36.03542

```

Los coeficientes estandarizados de forma que $\mathbf{w}'\mathbf{S}\mathbf{w} = 1$ son:

```

> dis$scaling

      LD1      LD2      LD3      LD4
MB  0.12667629  0.03873784  0.09276835  0.1488398644
BH -0.03703209  0.21009773 -0.02456846 -0.0004200843
BL -0.14512512 -0.06811443  0.01474860  0.1325007670
NH  0.08285128 -0.07729281 -0.29458931  0.0668588797

```

- c) Calcular la matriz \mathbf{S} de covarianzas dentro de los grupos o *pooled within-groups covariance matrix* sobre los cinco grupos.

```

> medias <- dis$means
> X <- skulls[, -1]
> table(EPOCH)

EPOCH
c4000BC c3300BC c1850BC c200BC cAD150
      30      30      30      30      30

> n <- dim(X)[1]
> S <- (n-1)*var(X - medias[EPOCH,])/(n-nlevels(EPOCH))

```

Comprobar que los coeficientes \mathbf{b}_i de las variables canónicas verifican $\mathbf{b}_i'\mathbf{S}\mathbf{b}_i = 1$.

```

> t(dis$scaling) %*% S %*% dis$scaling

      LD1      LD2      LD3      LD4
LD1  1.000000e+00 -7.077672e-16  3.053113e-16  2.359224e-16

```

```
LD2 -7.823603e-16 1.000000e+00 -5.134781e-16 -1.578598e-16
LD3 3.608225e-16 -4.996004e-16 1.000000e+00 3.608225e-16
LD4 2.636780e-16 -1.387779e-16 2.775558e-16 1.000000e+00
```

- d) Calcular los coeficientes⁵ lineales $\bar{\mathbf{x}}'_i \mathbf{S}^{-1}$ y la “constante” $-\frac{1}{2} \bar{\mathbf{x}}'_i \mathbf{S}^{-1} \bar{\mathbf{x}}_i + \log(\pi_i)$ para cada población, donde π_i es la probabilidad a priori de la i -ésima población. En nuestro caso suponemos $\pi_i = 1/5$.

```
> Sinv <- solve(S)
> g <- nlevels(EPOCH)
> # Coeficientes de los indicadores lineales
> wg <- dis$means %*% Sinv; wg

              MB          BH          BL          NH
c4000BC 6.001231 4.767429 2.956873 2.123815
c3300BC 6.051498 4.731595 2.961685 2.093824
c1850BC 6.150664 4.808988 2.818914 2.101283
c200BC 6.184994 4.738051 2.764660 2.258320
cAD150 6.220880 4.665018 2.739524 2.215393

> # Constantes
> eddiag <- function(x,Op) t(x) %*% Op %*% x
> k <- apply(dis$means, 1, eddiag, Op=Sinv)
> k <- -1/2*k + log(1/g); k

      c4000BC   c3300BC   c1850BC   c200BC   cAD150
-914.5279 -915.3511 -925.3426 -923.4198 -914.1228
```

- e) Hallar los coeficientes de las funciones⁶ discriminantes $h_{ij}(\mathbf{x})$

$$\mathbf{a}'_{ij} = \bar{\mathbf{x}}'_i \mathbf{S}^{-1} - \bar{\mathbf{x}}'_j \mathbf{S}^{-1} = (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)' \mathbf{S}^{-1}$$

y su constante

$$\begin{aligned} c_{ij} &= -\frac{1}{2} \bar{\mathbf{x}}'_i \mathbf{S}^{-1} \bar{\mathbf{x}}_i + \log(1/5) - \left[-\frac{1}{2} \bar{\mathbf{x}}'_j \mathbf{S}^{-1} \bar{\mathbf{x}}_j + \log(1/5) \right] \\ &= -(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)' \mathbf{S}^{-1} \left[\frac{1}{2} (\bar{\mathbf{x}}_i + \bar{\mathbf{x}}_j) \right] \\ &= -\mathbf{a}'_{ij} \mathbf{m}_{ij} \end{aligned}$$

de modo que la función discriminante entre dos poblaciones es

$$\begin{aligned} h_{ij}(\mathbf{x}) &= \mathbf{a}'_{ij} \mathbf{x} + c_{ij} = \mathbf{a}'_{ij} \mathbf{x} - \mathbf{a}'_{ij} \mathbf{m}_{ij} = \mathbf{a}'_{ij} (\mathbf{x} - \mathbf{m}_{ij}) \\ &= (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)' \mathbf{S}^{-1} \left[\mathbf{x} - \frac{1}{2} (\bar{\mathbf{x}}_i + \bar{\mathbf{x}}_j) \right] \end{aligned}$$

Como los coeficientes de las funciones discriminantes tienen longitud 4, necesitamos un array para almacenar sus valores

⁵En la página 402 del libro de Peña(2002) se llaman $\mathbf{w}_g = \mathbf{V}^{-1} \boldsymbol{\mu}_g$.

⁶En la página 402 del libro de Peña(2002) esencialmente son las funciones $A_{ij}(\mathbf{x})$.

```

> a <- array(0,dim = c(g,g,ncol(dis$means)))
> ck <- matrix(0,nrow=g,ncol=g)
> for(i in 2:g){
+   for(j in 1:(i-1)){
+     a[i,j,] <- wg[i,]-wg[j,]
+     m_ij <- (dis$means[i,]+dis$means[j,])/2
+     ck[i,j] <- (-1) * as.numeric(t(a[i,j,]) %*% m_ij)
+   }
+ }

```

Los coeficientes de las funciones discriminantes entre las dos primeras poblaciones son:

```

> a[2,1,]

[1] 0.050267566 -0.035833910 0.004812059 -0.029991198

> ck[2,1]

[1] -0.8232301

```

- f) Obtener con la función `plot()` el gráfico que muestra la separación entre las poblaciones que proporciona la primera variable discriminante.

Un primer gráfico puede ser la proyección de las medias de cada grupo sobre la primera variable canónica:

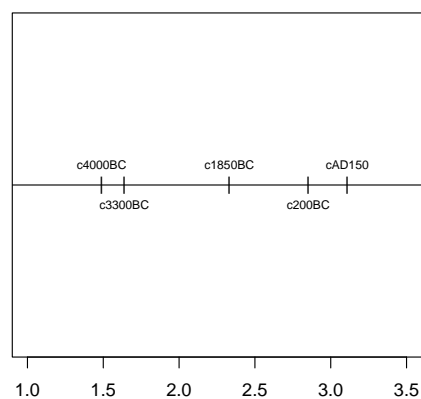
```

> (ld1 <- aco$Y[,1])

c4000BC c3300BC c1850BC c200BC cAD150
1.488731 1.638393 2.331508 2.851635 3.109167

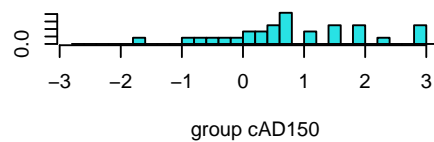
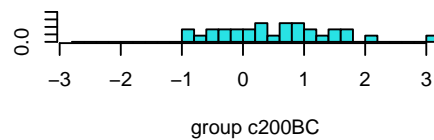
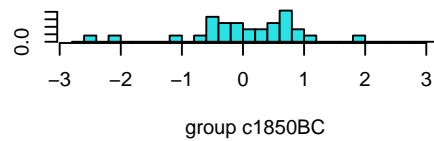
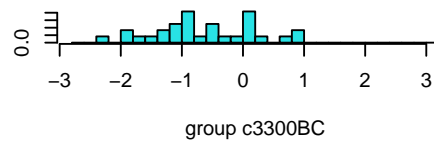
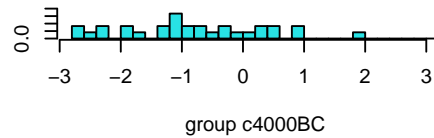
> stripchart(ld1, xlim=c(1,3.5), pch="|")
> abline(h=1)
> ct <- c(0.05,-0.05,0.05,-0.05,0.05)
> text(ld1,1+ct,levels(EPOCH),cex=0.7)

```



Otra posibilidad es la comparación de histogramas de datos proyectados sobre la primera variable canónica

```
> plot(dis, dimen=1)
```



g) Hallar la tabla de clasificación sencilla o *plug-in* y la tabla de validación cruzada.

La tabla de clasificación sencilla es

```
> cls.lda <- predict(dis)
> table(cls.lda$class)
```

```
c4000BC c3300BC c1850BC c200BC cAD150
      31      27      35      24      33
```

```
> table(cls.lda$class,real=skulls[,1])
```

	real				
	c4000BC	c3300BC	c1850BC	c200BC	cAD150
c4000BC	12	10	4	3	2
c3300BC	8	8	4	3	4
c1850BC	4	5	15	7	4
c200BC	4	4	2	5	9
cAD150	2	3	5	12	11

La tabla de validación cruzada:

```
> dis.CV <- lda(skulls[,2:5],grouping=skulls[,1],CV=TRUE)
> table(dis.CV$class)
```

c4000BC	c3300BC	c1850BC	c200BC	cAD150
31	28	33	25	33

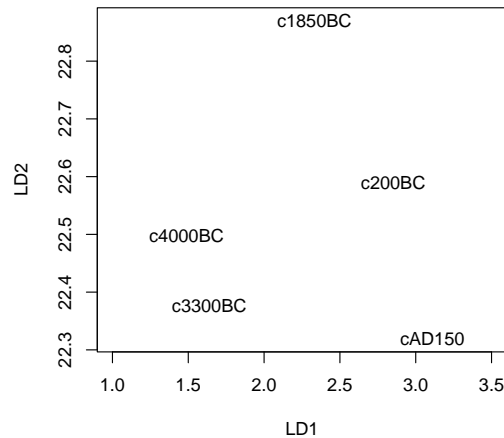
```
> table(dis.CV$class,real=skulls[,1])
```

	real				
	c4000BC	c3300BC	c1850BC	c200BC	cAD150
c4000BC	9	11	6	3	2
c3300BC	10	7	4	3	4
c1850BC	5	5	12	7	4
c200BC	4	4	2	5	10
cAD150	2	3	6	12	10

Curiosamente el método de validación cruzada resulta ser peor que el simple.

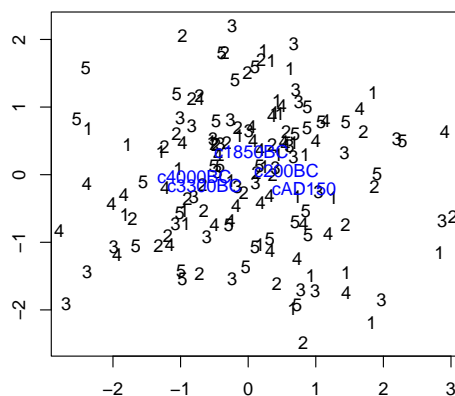
- h) Representar los puntos medios de las poblaciones en el diagrama con las dos primeras variables canónicas.

```
> dsfs <- as.matrix(skulls[,,-1]) %*% dis$scaling[,1:2]
> medias.cv <- aggregate(dsfs, skulls["EPOCH"], mean)
> plot(medias.cv[,2:3], type="n", xlim=c(1,3.5))
> text(medias.cv[,2:3], labels=medias.cv[,1])
```

o con los datos de las variables canónicas centrados

```
> skulls.pr <- predict(dis, dimen=2)
> skulls.t <- skulls.pr$x[,1:2] # =dsfs pero con columnas centradas
> skulls.m <- lda(skulls.t, EPOCH)$means
> # plot
> eqscplot(skulls.t, type="n")
> text(skulls.t, labels=1:5)
> text(skulls.m, labels=rownames(skulls.m), col="blue")
```



Ejercicio 4

Vamos a construir una regla para predecir el sexo de un futuro cangrejo *Leptograpsus* de especie desconocida tomando como muestra la base de datos *crabs* del paquete *MASS* de **R**. Los autores del paquete Venables & Ripley(2003, pág. 334-336) sugieren eliminar del análisis la variable *BD* ya que se midió de forma distinta para machos y hembras. También argumentan la utilización de los logaritmos de las variables.

```
> library(MASS)
> data(crabs)
> lcrabs <- log(crabs[,4:8])
```

- a) Realizar un análisis discriminante lineal con la función `lda()` para el sexo con las cuatro variables numéricas transformadas. Calcular la tabla de clasificación.

```
> lda.sex <- lda(lcrabs,crabs$sex)
> # Tabla de clasificación
> cls1 <- predict(lda.sex)
> table(cls1$class)

  F   M
100 100

> # Tabla de clasificación cruzada
> table(lda=cls1$class,real=crabs$sex)

      real
lda   F   M
  F  97   3
  M   3  97
```

- b) Repetir el análisis teniendo en cuenta las cuatro clases en función del sexo y la especie. Los machos se codifican con una letra mayúscula y las hembras con una minúscula:

```
> crabs.grp <- as.factor(c("B","b","O","o")[rep(1:4, each=50)])
> lda.4 <- lda(lcrabs,crabs.grp)
> # Tabla de clasificación
> cls4 <- predict(lda.4)
> table(cls4$class)

  b   B   o   O
53 47 47 53

> # Tabla de clasificación cruzada
> table(lda=cls4$class,real=crabs.grp)

      real
lda   b   B   o   O
  b  49   4   0   0
  B   1  46   0   0
  o   0   0  47   0
  O   0   0   3  50
```

- c) Como las dos primeras variables canónicas dominan la variación entre grupos, dibujar el gráfico de dispersión de los datos con ellas.

```
> (dcrabs.lda4 <- lda(crabs.grp ~ FL + RW + CL + CW, lcrabs))
```

Call:

```
lda(crabs.grp ~ FL + RW + CL + CW, data = lcrabs)
```

Prior probabilities of groups:

b	B	o	0
0.25	0.25	0.25	0.25

Group means:

	FL	RW	CL	CW
b	2.564985	2.475174	3.312685	3.462327
B	2.672724	2.443774	3.437968	3.578077
o	2.852455	2.683831	3.529370	3.649555
0	2.787885	2.489921	3.490431	3.589426

Coefficients of linear discriminants:

	LD1	LD2	LD3
FL	36.25600	-4.844633	-19.10647
RW	13.38368	22.786954	7.07711
CL	20.28868	-48.380432	58.34517
CW	-65.64476	33.710217	-49.51270

Proportion of trace:

	LD1	LD2	LD3
	0.6422	0.3491	0.0087

```
> dcrabs.pr4 <- predict(dcrabs.lda4, dimen=2)
```

```
>
```

```
> # Male posterior probabilities
```

```
> dcrabs.pr2 <- dcrabs.pr4$pos[,"B"] + dcrabs.pr4$pos[,"0"]
```

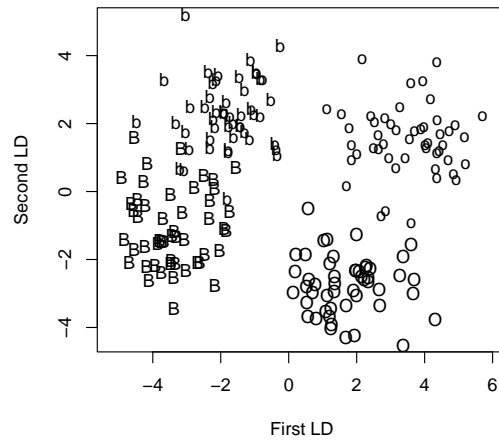
```
> table(crabs$sex, dcrabs.pr2 > 0.5)
```

	FALSE	TRUE
F	96	4
M	3	97

```
> cr.t <- dcrabs.pr4$x[,1:2]
```

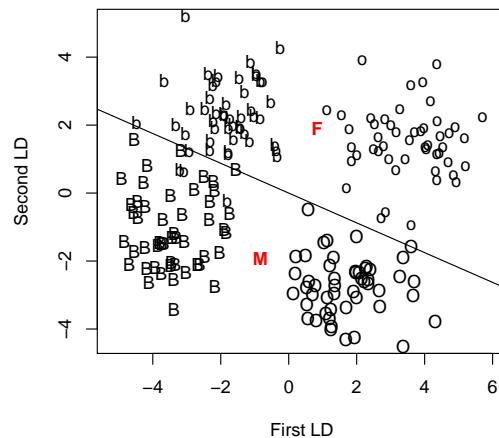
```
> eqscplot(cr.t, type="n", xlab="First LD", ylab="Second LD")
```

```
> text(cr.t, labels=as.character(crabs.grp))
```



- d) Añadir al gráfico anterior los puntos medios que representan a los dos sexos. Añadir también la recta discriminante.

```
> # recta perpendicular al segmento que une dos puntos
> perp <- function(x,y) {
+   m <- (x+y)/2
+   s <- - (x[1] - y[1])/(x[2] - y[2])
+   abline(c(m[2] - s*m[1], s))
+   invisible()
+ }
> # puntos medios
> cr.m <- lda(cr.t, crabs$sex)$means
> # gráfico
> eqscplot(cr.t, type="n", xlab="First LD", ylab="Second LD")
> text(cr.t, labels=as.character(crabs.grp))
> text(cr.m, labels=levels(crabs$sex), font=2, col="red")
> perp(cr.m[1,], cr.m[2,])
```



Ejercicio 5

Los científicos que han estudiado algunos de los insectos sociales, tales como las abejas y las hormigas, han observado que las reinas y las trabajadoras tienen diferentes tamaños y formas. En la página web

<https://pages.stat.wisc.edu/~yandell/pda/data/Wasp/>

se describen los datos y las variables de un experimento sobre las abejas.

Para cargar esos datos en **R** hacemos:

```
> wasp <- read.table("https://pages.stat.wisc.edu/~yandell/pda/data/Wasp/wasp.dat",
+                    header=T, stringsAsFactors = T)
> attach(wasp)
```

- a) Analizar las características morfológicas de las abejas de forma descriptiva univariante y multivariante según el factor **caste**. Añadir algunos gráficos ilustrativos.

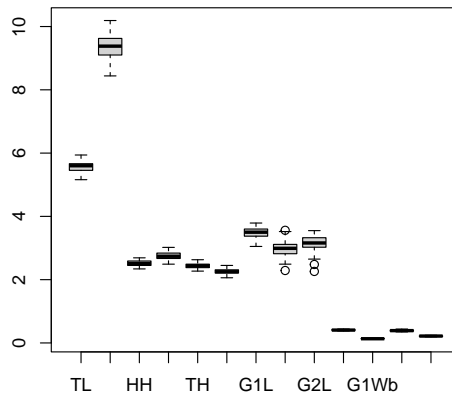
```
> library(psych)
> describe(wasp[, -1])[, -1]
```

	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
TL	100	5.57	0.15	5.60	5.57	0.16	5.16	5.94	0.78	-0.23	-0.16	0.02
WL	100	9.38	0.38	9.38	9.39	0.37	8.44	10.19	1.75	-0.08	-0.53	0.04
HH	100	2.52	0.08	2.51	2.52	0.10	2.34	2.69	0.35	0.04	-0.98	0.01
HW	100	2.76	0.12	2.74	2.76	0.13	2.49	3.02	0.53	0.08	-0.73	0.01
TH	100	2.44	0.08	2.44	2.44	0.07	2.27	2.63	0.36	0.18	-0.50	0.01
TW	100	2.26	0.07	2.25	2.26	0.08	2.06	2.45	0.39	0.15	-0.16	0.01
G1L	100	3.49	0.15	3.50	3.49	0.16	3.05	3.79	0.74	-0.53	-0.22	0.01
G2Wa	100	2.98	0.23	2.99	2.98	0.22	2.29	3.56	1.27	-0.08	0.14	0.02
G2L	100	3.15	0.23	3.16	3.16	0.22	2.26	3.55	1.29	-0.82	1.32	0.02
HL	100	0.41	0.02	0.41	0.41	0.02	0.38	0.44	0.06	0.13	-1.01	0.00
G1Wb	100	0.13	0.01	0.13	0.13	0.01	0.12	0.15	0.03	-0.04	-0.36	0.00
G1Wa	100	0.39	0.02	0.39	0.39	0.03	0.35	0.44	0.09	0.09	-0.96	0.00
G1H	100	0.22	0.01	0.22	0.22	0.01	0.19	0.24	0.05	0.01	-0.35	0.00

```
> table(caste)

caste
  Q   W
50 50

> boxplot(wasp[, -1])
```



La siguiente función calcula estadísticos globales y por grupo:

```
> gfn <- function(data,g,FUN){
+   gbl <- sapply(data, FUN)
+   byg <- function(x) tapply(x,g,FUN)
+   gbl <- rbind(gbl,apply(data,2,byg))
+   t(gbl)
+ }
```

Las desviaciones típicas globales, por grupo y dentro de grupo son:

```
> s <- gfn(wasp[, -1], caste, sd)
> n <- nrow(wasp)
> s <- cbind(s, pooled=sqrt((n/2-1)*(s[,2]^2+s[,3]^2)/(n-2)))
> round(s,5)
```

	gbl	Q	W	pooled
TL	0.15330	0.13805	0.14155	0.13981
WL	0.38222	0.25725	0.27220	0.26483
HH	0.08067	0.04479	0.04885	0.04686
HW	0.11933	0.06585	0.08026	0.07341
TH	0.07615	0.05844	0.08093	0.07058
TW	0.07135	0.06936	0.07218	0.07079
G1L	0.14550	0.11259	0.06672	0.09254
G2Wa	0.22982	0.21354	0.18069	0.19780
G2L	0.23220	0.20215	0.13753	0.17289
HL	0.01602	0.00872	0.01061	0.00971
G1Wb	0.00529	0.00517	0.00526	0.00521
G1Wa	0.02422	0.01424	0.01288	0.01358
G1H	0.01044	0.01083	0.00911	0.01001

Las medias globales, por grupo, diferencia, estadístico t y p -valor (ordenado):

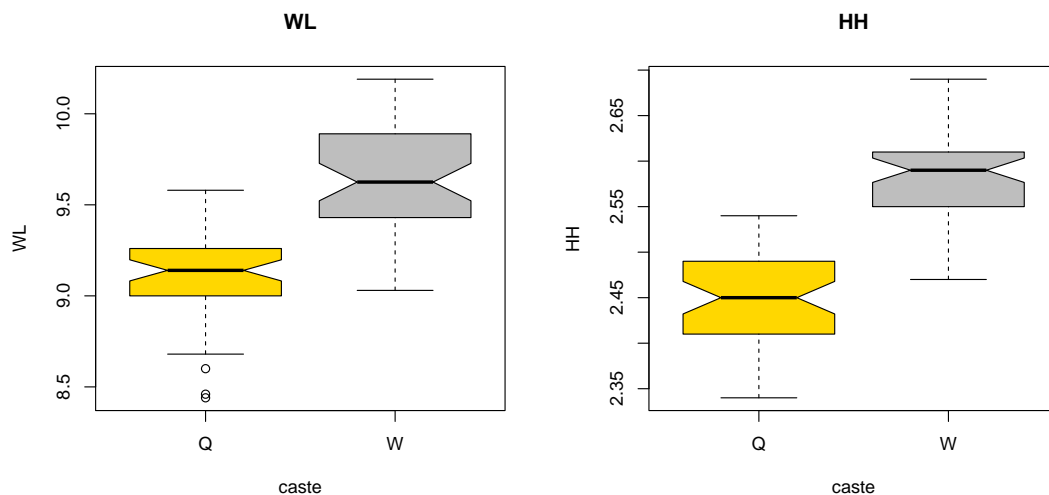
```
> m <- gfn(wasp[, -1], caste, mean)
> t.cv <- qt(0.975, 49)
```

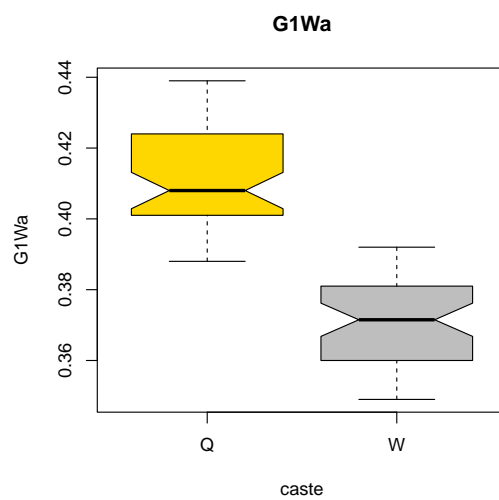
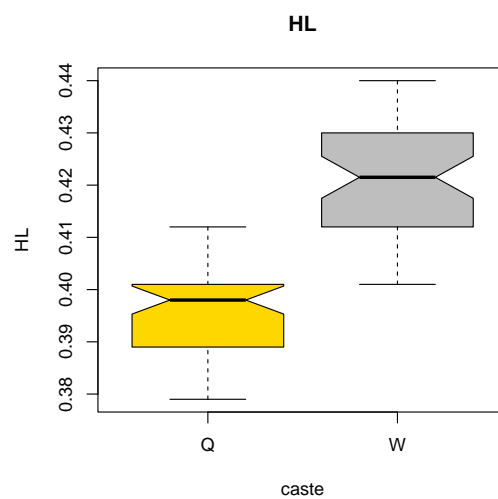
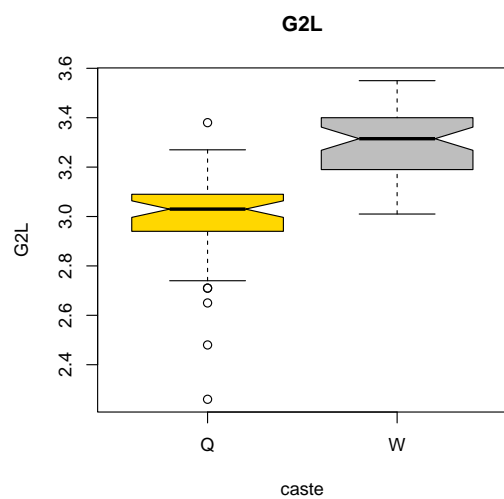
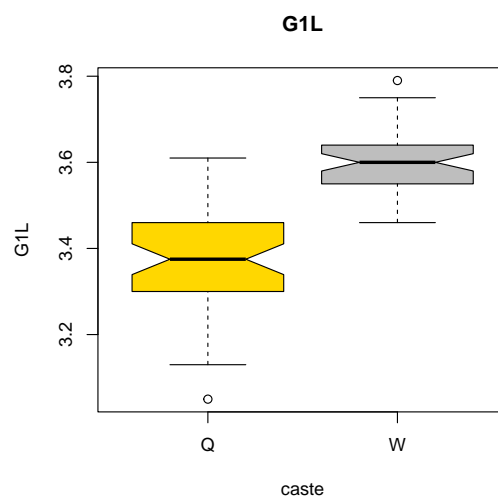
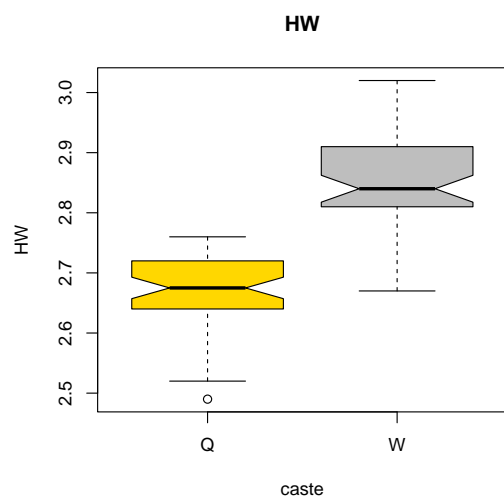
```
> t<-(m[,2]-m[,3])/s[,4]
> m <- cbind(m,diff=m[,2]-m[,3],t=t,p= 2*pt(abs(t),49,lower.tail=F))
> round(m[order(m[,6]),],4)
```

	gbl	Q	W	diff	t	p
G1Wa	0.3907	0.4107	0.3707	0.0400	2.9462	0.0049
HH	2.5157	2.4502	2.5812	-0.1310	-2.7954	0.0074
HL	0.4084	0.3956	0.4211	-0.0254	-2.6178	0.0117
HW	2.7589	2.6650	2.8528	-0.1878	-2.5584	0.0137
G1L	3.4855	3.3734	3.5976	-0.2242	-2.4227	0.0191
WL	9.3841	9.1086	9.6596	-0.5510	-2.0806	0.0427
G2L	3.1474	2.9922	3.3026	-0.3104	-1.7954	0.0788
G2Wa	2.9789	3.0970	2.8608	0.2362	1.1941	0.2382
TL	5.5697	5.5056	5.6338	-0.1282	-0.9170	0.3637
TH	2.4405	2.4112	2.4698	-0.0586	-0.8302	0.4104
G1H	0.2164	0.2195	0.2133	0.0062	0.6235	0.5358
G1Wb	0.1336	0.1346	0.1325	0.0021	0.4028	0.6889
TW	2.2612	2.2726	2.2498	0.0228	0.3221	0.7488

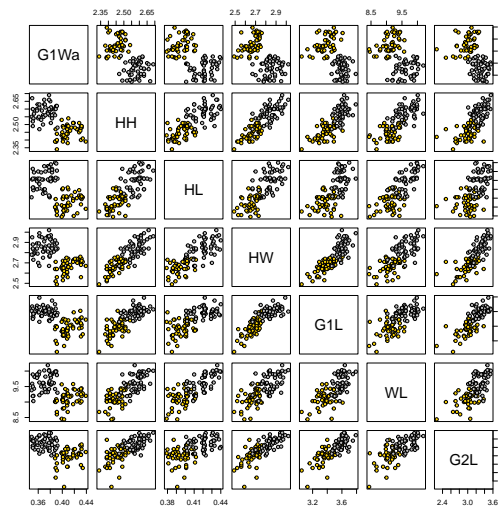
A la vista de estos resultados, las variables que pueden mostrar diferencias entre uno y otro grupo son: WL, HH, HW, G1L, HL y G1Wa.

```
> boxplot(WL~caste,notch=T,col=c("gold","gray"),main="WL")
> boxplot(HH~caste,notch=T,col=c("gold","gray"),main="HH")
> boxplot(HW~caste,notch=T,col=c("gold","gray"),main="HW")
> boxplot(G1L~caste,notch=T,col=c("gold","gray"),main="G1L")
> boxplot(G2L~caste,notch=T,col=c("gold","gray"),main="G2L")
> boxplot(HL~caste,notch=T,col=c("gold","gray"),main="HL")
> boxplot(G1Wa~caste,notch=T,col=c("gold","gray"),main="G1Wa")
```






```
> # Subconjunto de datos con variables significativas
> svars <- c("G1Wa", "HH", "HL", "HW", "G1L", "WL", "G2L")
> # Gráfico por parejas
> pnfn <- function(x,y) points(x,y,pch=21,bg=c("gold","gray")[caste])
> pairs(wasp[,svars],panel=pnfn)
```



b) Realizar un análisis discriminante lineal. La función `lda()` del paquete *MASS* puede servir.

Antes de realizar el análisis discriminante lineal, podemos estudiar si las 7 variables más significativas pueden ser suficientes para la discriminación. Este estudio lo haremos con los análisis canónicos del conjunto reducido y del total.

```
> # ... con las 7 variables más significativas
> aco.7vs <- anal.canonico(wasp[,svars], wasp$caste); aco.7vs$values

[1] 1.777943e+01 1.065814e-14 9.115364e-16 7.018183e-16 5.534527e-16
[6] -4.691258e-17 -5.517048e-16

> # ... con todas las variables
> aco.14v <- anal.canonico(wasp[,,-1],caste); aco.14v$values

[1] 2.511175e+01 7.105427e-15 2.028966e-15 1.306172e-15 1.157706e-15
[6] 5.781206e-16 4.887713e-16 2.725270e-16 1.495885e-16 -1.886984e-17
[11] -5.670527e-17 -7.173977e-16 -1.608410e-15
```

Separación entre los centros

```
> # ... con las 7 variables más significativas
> (df1 <- as.numeric(diff(ac0.7vs$Y[,1])))

[1] -5.963125

> # ... con todas las variables
> (df2 <- as.numeric(diff(ac0.14v$Y[,1])))

[1] -7.086854
```

De modo que el tanto por ciento de separación debido a las 7 variables menos significativas es sólo de:

```
> round(abs((df1-df2)/df2)*100,2)

[1] 15.86
```

Así pues parece razonable construir el discriminador únicamente con las 7 variables más significativas:

```
> (zws <- lda(caste~G1Wa+HH+HL+HW+G1L+WL+G2L,data=wasp))

Call:
lda(caste ~ G1Wa + HH + HL + HW + G1L + WL + G2L, data = wasp)

Prior probabilities of groups:
  Q  W
0.5 0.5

Group means:
      G1Wa      HH      HL      HW      G1L      WL      G2L
Q 0.41072 2.4502 0.39564 2.6650 3.3734 9.1086 2.9922
W 0.37072 2.5812 0.42106 2.8528 3.5976 9.6596 3.3026

Coefficients of linear discriminants:
      LD1
G1Wa -66.8282034
HH      4.6354054
HL     40.9199663
HW      2.7897518
G1L     3.6262165
WL      0.8715826
G2L    -0.5624420

> cls.w <- predict(zws,wasp[,svars])
> sum(cls.w$class==caste)

[1] 100
```

Todos los ejemplares se clasifican correctamente.

- c) (**) Comparar las matrices de covarianzas de las dos poblaciones con el test de la razón de verosimilitudes (ver el ejercicio 13e del segundo módulo).

También se puede aplicar el test M de Box⁷

Un código para su cálculo en **R**: <http://finzi.psych.upenn.edu/R/Rhelp02a/archive/33330.html>. Ambos son muy sensibles a la no normalidad de los datos y tienden a rechazar la igualdad de covarianzas.

```
> p <- ncol(wasp)-1
> n1 <- table(caste)[1]
> n2 <- table(caste)[2]
```

⁷http://publib.boulder.ibm.com/infocenter/spssstat/v20r0m0/topic/com.ibm.spss.statistics.help/alg_boxs-m.htm

```

> S1 <- (n1-1)*cov(wasp[caste=="W",-1])/n1
> S2 <- (n2-1)*cov(wasp[caste=="Q",-1])/n2
> S <- (n1*S1+n2*S2)/(n1+n2)
> llr <- (n1+n2)*log(det(S))-n1*log(det(S1))-n2*log(det(S2))
> p.valor <- pchisq(llr, p*(p+1)/2, lower.tail=FALSE)
> c(llr=as.numeric(llr),p.valor=as.numeric(p.valor))

          llr          p.valor
1.679195e+02 1.677558e-06

```

Con el test M de Box:

```

> biotools::boxM(wasp[, -1], caste)

Box's M-test for Homogeneity of Covariance Matrices

data:  wasp[, -1]
Chi-Sq (approx.) = 142.01, df = 91, p-value = 0.0005014

```

Con ambos tests rechazamos la igualdad de las matrices de covarianzas.

- d) *En el caso de poblaciones normales con diferentes matrices de covarianzas se clasificará cada observación en el grupo con máxima probabilidad a posteriori, pero entonces las funciones discriminantes no son lineales, ya que tienen un término de segundo grado.*

Realizar un análisis discriminante cuadrático.

La función `qda()` del paquete `MASS` nos ayudará.

```

> (qda.w <- qda(caste~G1Wa+HH+HL+HW+G1L+WL+G2L,data=wasp))

Call:
qda(caste ~ G1Wa + HH + HL + HW + G1L + WL + G2L, data = wasp)

Prior probabilities of groups:
  Q  W
0.5 0.5

Group means:
      G1Wa      HH      HL      HW      G1L      WL      G2L
Q 0.41072 2.4502 0.39564 2.6650 3.3734 9.1086 2.9922
W 0.37072 2.5812 0.42106 2.8528 3.5976 9.6596 3.3026

> clq <- predict(qda.w, wasp[, -1])$class
> sum(clq==caste)

[1] 100

```

- e) *Calcular el número de parámetros que hay que estimar en la discriminación lineal y en la cuadrática.*

En el caso lineal hay que estimar las medias de cada grupo y la matriz de covarianzas común. Dada la simetría de la matriz de covarianzas, para p variables habrá que estimar $p(p+1)/2$, es decir, los

elementos de la diagonal más los de la matriz triangular inferior (o superior). Así pues, en el caso de G grupos y p variables, el número de variables a estimar es:

$$Gp + p(p + 1)/2$$

En el caso cuadrático hay que estimar la media y la matriz de covarianzas de cada grupo, que ya no es común. Así pues, habrá que estimar:

$$G(p + p(p + 1)/2)$$

Consideraando los dos grupos y las 13 variables de nuestro caso, el número de parámetros a estimar es de 117 para el discriminador lineal, y de 208 para el discriminador cuadrático. Si decidimos tomar únicamente las 7 variables más significativas, estos números se reducen a 42 y 70. Como el discriminador lineal con 7 variables consigue clasificar bien todos los datos, lo más aconsejable en este caso es desestimar la opción más compleja. Cuantos menos parámetros hay que estimar, más robusto resulta el discriminador, y más fácil su interpretación.

f) *Calcular los errores de clasificación con ambas reglas utilizando validación cruzada.*

Si son similares, nos quedaremos con el análisis lineal que además es más robusto y de mejor interpretación.

```
> lda.cv <- lda(caste~G1Wa+HH+HL+HW+G1L+WL+G2L,data=wasps,CV=T)
> table(lda.cv$class,caste)

      caste
      Q  W
Q  50  0
W   0 50

> qda.cv <- qda(caste~G1Wa+HH+HL+HW+G1L+WL+G2L,data=wasps,CV=T)
> table(qda.cv$class,caste)

      caste
      Q  W
Q  50  0
W   0 50
```

Como ya hemos dicho nos quedamos con el discriminador lineal.

Ejercicio 6

El banco de datos **wbca** contiene información sobre un estudio de cáncer de mama en el estado norteamericano de Wisconsin. La variable **Class** indica si el tumor es maligno (valor 0) o si es benigno (valor 1). El resto son variables que describen el estado de la paciente en una escala de 1 (normal) a 10 (muy poco normal) evaluadas por la observación de un médico. El objetivo es clasificar a la paciente como enferma o no de cáncer utilizando estas variables en lugar de algún otro método más invasivo.

```
> data(wbca, package = "faraway")
> attach(wbca)
```

a) *Realizar un análisis discriminante lineal.*

Añadir algún gráfico ilustrativo.

```
> (lda.wbca <- lda(wbca[, -1], Class))

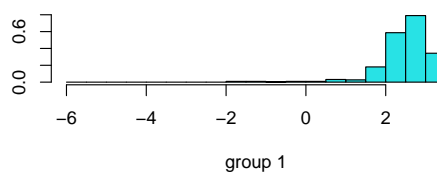
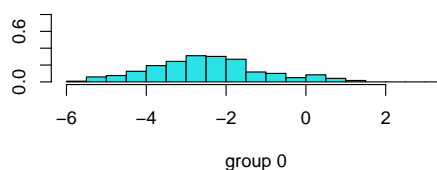
Call:
lda(wbca[, -1], Class)

Prior probabilities of groups:
      0      1 
0.349486 0.650514 

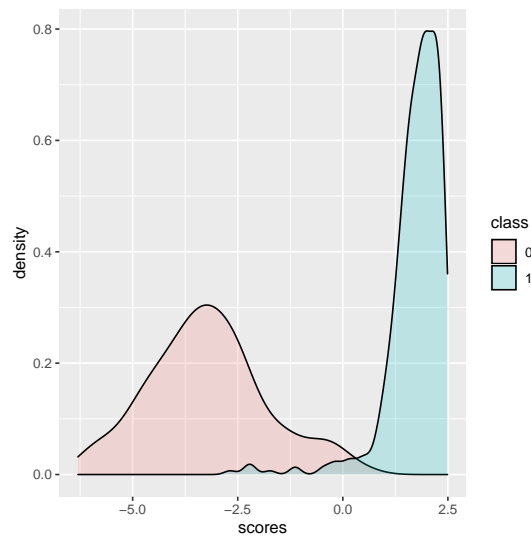
Group means:
      Adhes      BNucl      Chrom      Epith      Mitos      NNucl      Thick      UShap
0 5.567227 7.651261 5.966387 5.327731 2.609244 5.861345 7.197479 6.546218
1 1.338600 1.334086 2.072235 2.103837 1.063205 1.246050 2.952596 1.408578
      USize
0 6.563025
1 1.300226

Coefficients of linear discriminants:
      LD1
Adhes -0.045370376
BNucl -0.268564522
Chrom -0.116445050
Epith -0.058806527
Mitos -0.006909037
NNucl -0.115720030
Thick -0.189671159
UShap -0.082727580
USize -0.120895945

> plot(lda.wbca)
```



```
> class <- factor(Class)
> ggplot(wbca, aes(predict(lda.wbca)$x, fill = class)) +
+   geom_density(alpha = 0.2) + xlab("scores")
```



- b) Como tenemos una muestra muy grande, podemos elegir una muestra de entrenamiento para estimar mejor los errores de clasificación.

```
> set.seed(123)
> train <- sample(1:nrow(wbca), 400)
```

Utilizar el parámetro `subset = train` para limitar el análisis a este subconjunto. Predecir el resto de elementos con las variables obtenidas y calcular la tabla de clasificación.

Comparar el resultado con las tablas de clasificación *plug-in* y de valoración cruzada con todas las pacientes, normalmente más optimistas.

```
> (lda.w2 <- lda(wbca[, -1], Class, subset=train))
```

Call:

```
lda(wbca[, -1], Class, subset = train)
```

Prior probabilities of groups:

```
  0    1
0.345 0.655
```

Group means:

```
      Adhes  BNucl  Chrom  Epith  Mitos  NNucl  Thick  UShap
0  5.376812 7.739130 5.978261 5.282609 2.731884 5.789855 7.072464 6.586957
1  1.290076 1.339695 2.072519 2.057252 1.041985 1.259542 2.961832 1.374046
      USize
0  6.652174
1  1.290076
```

Coefficients of linear discriminants:

```
      LD1
Adhes -0.03201002
BNucl -0.30042194
Chrom -0.14283115
Epith -0.00375152
Mitos  0.01226397
```

```

NNucl -0.11958161
Thick -0.15965319
UShap -0.09800947
USize -0.14321783

> pred1 <- predict(lda.w2,newdata=wbca[-train,-1])
> # Tabla de clasificación
> table(pred1$class)

  0   1
92 189

> # Tabla de clasificación cruzada
> table(pred=pred1$class,true=Class[-train])

      true
pred   0   1
  0  89   3
  1  11 178

```

Con todas las pacientes:

```

> pred2 <- predict(lda.wbca,newdata=wbca[-train,-1])
> # Tabla de clasificación
> table(pred2$class)

  0   1
93 188

> # Tabla de clasificación cruzada
> table(pred=pred2$class,true=Class[-train])

      true
pred   0   1
  0  90   3
  1  10 178

```

- c) (*) Cuando el factor tiene dos clases, la primera combinación discriminante es equivalente a la que se obtiene en una regresión múltiple⁸ tomando como variable respuesta una variable numérica con valores $-1/n_1$ y $1/n_2$, según la clase, y donde n_1 y n_2 son los tamaños de los dos grupos.

```

> n1 <- table(Class)[1]
> n2 <- table(Class)[2]
> y <- c(-1/n1,1/n2)[Class+1]
> X <- data.frame(y,scale(wbca[, -1],scale=FALSE))
> (lmd <- lm(y~.-1, data=X))

```

⁸Ver el apéndice 13.2 de Peña(2002).

```
Call:
lm(formula = y ~ . - 1, data = X)

Coefficients:
      Adhes      BNucl      Chrom      Epith      Mitos      NNucl
-5.027e-05 -2.975e-04 -1.290e-04 -6.515e-05 -7.655e-06 -1.282e-04
      Thick      UShap      USize
-2.101e-04 -9.165e-05 -1.339e-04
```

La matriz de covarianzas común es:

```
> S1 <- (n1-1)*cov(wbca[Class==0,-1])/n1
> S2 <- (n2-1)*cov(wbca[Class==1,-1])/n2
> S <- (n1*S1+n2*S2)/(n1+n2-2)
```

Con esta matriz podemos normalizar el vector de coeficientes:

```
> (k <- sqrt( as.numeric(t(lmd$coeff) %*% S %*% lmd$coeff) ))

[1] 0.001107903

> lmd$coeff/k

      Adhes      BNucl      Chrom      Epith      Mitos
-0.045370376 -0.268564522 -0.116445050 -0.058806527 -0.006909037
      NNucl      Thick      UShap      USize
-0.115720030 -0.189671159 -0.082727580 -0.120895945
```

Resultado que coincide con la primera coordenada canónica:

```
> lda.wbca$scaling

      LD1
Adhes -0.045370376
BNucl -0.268564522
Chrom -0.116445050
Epith -0.058806527
Mitos -0.006909037
NNucl -0.115720030
Thick -0.189671159
UShap -0.082727580
USize -0.120895945
```

- d) (**) Como el factor considerado sólo tiene dos estados, también podemos acometer una regresión logística:

```
> (logres <- glm(Class ~ ., wbca, family = binomial, subset = train))

Call:  glm(formula = Class ~ ., family = binomial, data = wbca, subset = train)
```



```

Coefficients:
(Intercept)      Adhes      BNucl      Chrom      Epith
    10.92199    -0.45355    -0.56901    -0.45318     0.03100
      Mitos      NNucl      Thick      UShap      USize
    -0.62181    -0.20128    -0.47704    -0.41011    -0.05273

Degrees of Freedom: 399 Total (i.e. Null);  390 Residual
Null Deviance:      515.4
Residual Deviance: 46.26  AIC: 66.26

```

Predicciones sobre el conjunto de verificación:

```

> cls.lr <- (predict(logres, newdata=wbca[-train,-1], type="response")>0.5)
> cls.lr <- ifelse(cls.lr,1,0)
> # Tabla cruzada de clasificación
> table(pred=cls.lr,true=Class[-train])

      true
pred   0   1
   0  95   4
   1   5 177

```

Ejercicio 7

Consideremos un grupo de pacientes de los cuales se conocen algunas variables obtenidas de un análisis de orina. En concreto las variables nos dan la gravedad específica *grav*, la osmolaridad *osmo*, la conductibilidad *conduc*, la concentración de urea *urea* y la concentración de calcio *calcio*. También tenemos una variable *grupo* que nos indica la presencia o ausencia de cristales en la orina del individuo, donde 1 indica la ausencia y 2 la presencia. El siguiente código prepara los datos para su análisis:

```

> x <- read.table("T44.1", header=F)
> x <- x[, -(1:4)]
> x[x == -9999] <- NA
> x[, 1] <- factor(x[, 1], levels=1:2, labels=c("A", "P"))
> cristal <- x; remove(x)
> names(cristal) <- c("grupo", "calcio", "conduc", "grav", "osmo", "ph", "urea")
> attach(cristal)

```

Observemos la substitución del valor original *missing* -9999 por el *NA* de **R**. También es conveniente tratar a la variable que indica el grupo de los individuos como un factor.

- a) Realizar un análisis discriminante lineal con la función *lda()* del paquete *MASS* y probabilidades a priori iguales a 0.5.

Habrà que utilizar el parámetro *prior* y también⁹ *na.action = na.omit*.

```

> (lda.cr <- lda(grupo ~ calcio+conduc+grav+osmo+ph+urea,
+               prior = c(0.5,0.5), na.action=na.omit))

Call:
lda(grupo ~ calcio + conduc + grav + osmo + ph + urea, prior = c(0.5,

```

⁹Otra posibilidad es excluir de la base de datos todos los individuos con algún valor *missing*: *cc <- complete.cases(x); cristal <- x[cc,]*.

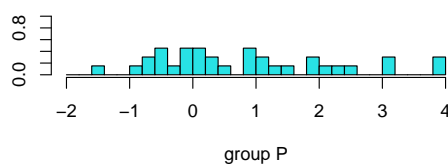
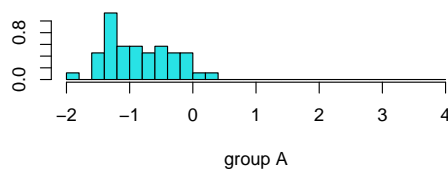
```
0.5), na.action = na.omit)

Prior probabilities of groups:
  A   P
0.5 0.5

Group means:
      calcio  conduc   grav   osmo   ph   urea
A 1.015364 6.125682 561.6591 20.55000 232.4318 2.628864
P 1.021576 5.927273 682.8788 21.37879 302.3636 6.202424

Coefficients of linear discriminants:
          LD1
calcio 140.610784925
conduc  -0.125723350
grav     0.003408930
osmo    -0.117156709
ph      -0.008383726
urea     0.301091251

> plot(lda.cr)
```



La tabla de clasificación cruzada es:

```
> cc <- complete.cases(cristal)
> table(pred=predict(lda.cr)$class,true=grupo[cc])

      true
pred  A  P
A   42 11
P    2 22
```

- b) Realizar otro análisis discriminante lineal con la función `lda()` pero con las probabilidades a priori que coincidan con las proporciones observadas dentro de la muestra.

```
> (lda.cr2 <- lda(grupo ~ calcio+conduc+grav+osmo+ph+urea, na.action=na.omit))

Call:
lda(grupo ~ calcio + conduc + grav + osmo + ph + urea, na.action = na.omit)

Prior probabilities of groups:
      A      P
0.5714286 0.4285714

Group means:
      calcio  conduc   grav   osmo    ph    urea
A 1.015364 6.125682 561.6591 20.55000 232.4318 2.628864
P 1.021576 5.927273 682.8788 21.37879 302.3636 6.202424

Coefficients of linear discriminants:
      LD1
calcio 140.610784925
conduc -0.125723350
grav    0.003408930
osmo    -0.117156709
ph       -0.008383726
urea     0.301091251

> table(pred=predict(lda.cr2)$class,true=grupo[cc])

      true
pred  A  P
  A 42 14
  P  2 19
```

Los coeficientes del discriminante lineal son los mismos, pero la asignación no.

- c) Hallar las probabilidades a posteriori en ambos casos. Deberemos utilizar el parámetro $CV = T$.

```
> lda.cr3 <- lda(grupo ~ calcio+conduc+grav+osmo+ph+urea,
+               prior = c(0.5,0.5), na.action=na.omit, CV=T)
> table(pred=lda.cr3$class,true=grupo[cc])

      true
pred  A  P
  A 41 14
  P  3 19

> lda.cr4 <- lda(grupo ~ calcio+conduc+grav+osmo+ph+urea, na.action=na.omit, CV=T)
> table(pred=lda.cr4$class,true=grupo[cc])

      true
pred  A  P
  A 42 14
  P  2 19
```

Las probabilidades a posteriori se hallan con

```
> lda.cr3$posterior
> lda.cr4$posterior
```

Ejercicio 8

En este ejercicio vamos a utilizar los famosos datos *iris* que Fisher usó para introducir la metodología. Tenemos tres especies de iris y cuatro características numéricas medidas sobre cada iris. Los datos están en el *data.frame* *iris* del paquete *MASS*. El objetivo es capacitarnos para clasificar un futuro individuo en la especie correcta basándonos únicamente en las medidas físicas.

En primer lugar recodificaremos el factor para utilizarlo mejor en los gráficos.

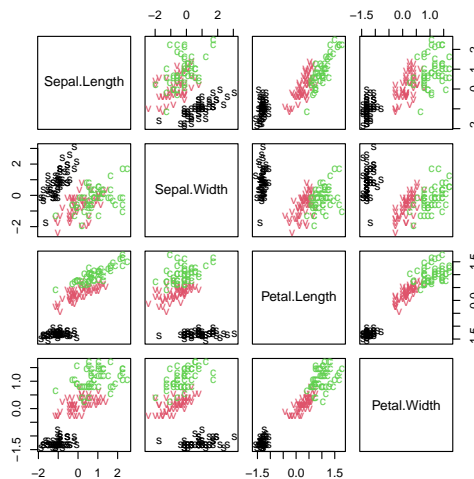
```
> data(iris)
> levels(iris$Species)

[1] "setosa"      "versicolor" "virginica"

> Sp <- c("s", "v", "c")[as.numeric(iris$Species)]
```

- a) Realizar un gráfico con la función `pairs()` de las variables numéricas con distintos símbolos y colores para cada especie.

```
> pairs(scale(iris[, -5]), pch = Sp, col = as.numeric(iris$Species))
```



- b) Antes de proceder con un análisis discriminante lineal, realizar un análisis de clasificación jerárquico y otro por el método *k-medias*. Obtener las tablas de clasificación por ambos métodos.

Cluster jerárquico con distancia media:

```
> hc <- hclust(dist(iris[, 1:4]), method="average")
> # Tres clusters
> hr <- cutree(hc, 3)
> # Tabla de clasificación
> table(hr)
```

```
hr
  1  2  3
50 64 36

> # Tabla clasificación cruzada
> table(hr,Sp)

      Sp
hr     c  s  v
  1    0 50  0
  2   14  0 50
  3   36  0  0
```

La especie virginica no se clasifica correctamente.

Clasificación por k -means en tres clusters:

```
> iris.km <- kmeans(iris[,1:4], centers=3)
> # Tabla de clasificación
> table(km=iris.km$cluster)

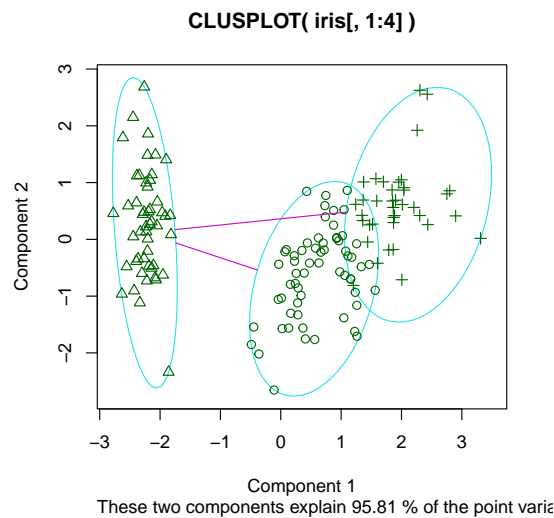
km
  1  2  3
62 50 38

> # Tabla de clasificación cruzada
> table(km=iris.km$cluster,Sp)

      Sp
km     c  s  v
  1   14  0 48
  2    0 50  0
  3   36  0  2
```

Representación de los grupos formados:

```
> library(cluster)
> clusplot(iris[,1:4],iris.km$cluster,diss=FALSE)
```



Esta representación es excelente.

- c) Realizar un análisis discriminante lineal y obtener la tabla de clasificación con la primera variable discriminante.

```
> (lda.iris <- lda(iris[, -5], grouping=iris$Species))
```

Call:
lda(iris[, -5], grouping = iris\$Species)

Prior probabilities of groups:

setosa	versicolor	virginica
0.3333333	0.3333333	0.3333333

Group means:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
setosa	5.006	3.428	1.462	0.246
versicolor	5.936	2.770	4.260	1.326
virginica	6.588	2.974	5.552	2.026

Coefficients of linear discriminants:

	LD1	LD2
Sepal.Length	0.8293776	0.02410215
Sepal.Width	1.5344731	2.16452123
Petal.Length	-2.2012117	-0.93192121
Petal.Width	-2.8104603	2.83918785

Proportion of trace:

	LD1	LD2
	0.9912	0.0088

La discriminación por una sola dimensión es

```
> cls.iris.d1 <- predict(lda.iris, dimen=1)
> # Tabla de clasificación
> table(cls.iris.d1$class)
```

```

      setosa versicolor  virginica
      50         48         52

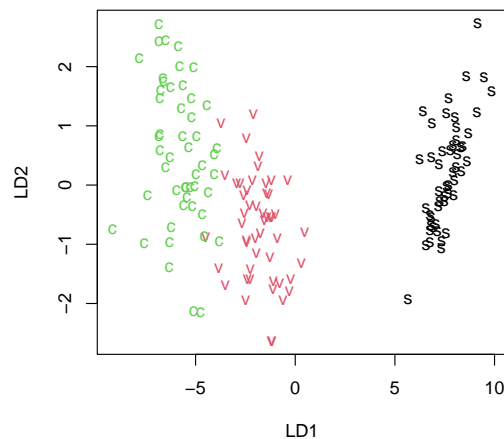
> # Tabla de clasificación cruzada
> table(lda=cls.iris.d1$class,iris$Species)

lda      setosa versicolor virginica
setosa      50         0         0
versicolor  0         48         0
virginica   0          2        50

```

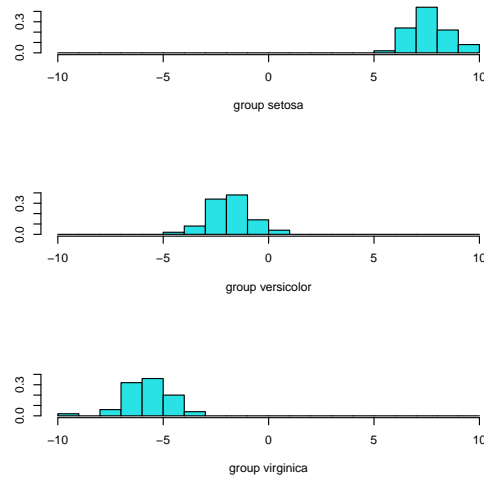
d) Dibujar el gráfico de dispersión de los datos transformados por las dos primeras variables canónicas.

```
> plot(predict(lda.iris)$x, pch=Sp, col=as.numeric(iris$Species))
```



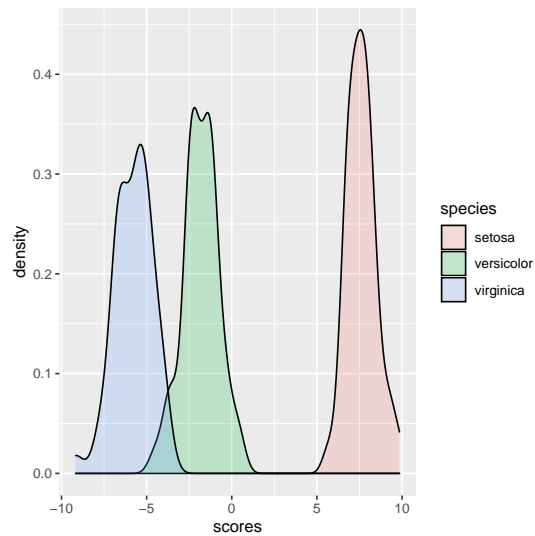
e) Dibujar las distribuciones de los grupos en la primera variable discriminante.

```
> plot(lda.iris, dimen = 1)
```



También

```
> species <- iris$Species
> ggplot(iris, aes(predict(lda.iris)$x[,1], fill = species)) +
+   geom_density(alpha = 0.2) + xlab("scores")
```



En este ejercicio se ve muy bien la diferencia entre la clasificación supervisada, cuando se conocen las etiquetas de todos los datos, por análisis discriminante, y la clasificación no supervisada, cuando se trata de formar grupos homogéneos y se desconoce la filiación de cada dato, por el análisis cluster.

También ilustra bastante bien, como en el caso del análisis de orina o el de los datos wbca, el problema de clasificación y/o discriminación que presentan los casos con poblaciones que se solapan mucho. La única solución admisible en estos casos pasa por la construcción de una función de coste que contemple los costes asociados a los falsos positivos y a los falsos negativos. Las poblaciones no dejan de estar igualmente solapadas, pero en la decisión intervienen condicionantes económicos, psicológicos o de otro tipo que tienen en cuenta las pérdidas asociadas a cada tipo de error. Entonces lo que minimiza la función discriminante es el total de pérdidas.

Ejercicio 9

El método *k*-nn (*k*-nearest neighbors, Fix y Hodges, 1951) es un método de clasificación supervisada (o aprendizaje con estimación basada en un conjunto de entrenamiento y prototipos) que sirve para estimar la función de densidad $f(\mathbf{x}|C_j)$ de las variables predictoras \mathbf{x} en cada clase C_j , $j = 1, \dots, g$.

Este es un método de clasificación no paramétrico, que estima el valor de la función de densidad de probabilidad o directamente la probabilidad a posteriori de que un elemento \mathbf{x} pertenezca a la clase C_j a partir de la información proporcionada por el conjunto de prototipos. En el proceso de aprendizaje no se hace ninguna suposición acerca de la distribución de las variables predictoras.

Otros métodos no paramétricos difieren en la elección del conjunto de prototipos de referencia. El *k*-nn usa el conjunto de entrenamiento o un subconjunto de él. La “cercanía” se mide con la distancia euclídea. Aplicar la función `knn()` del paquete *class* al conjunto de entrenamiento de 400 datos de *wbca* para predecir el tipo de tumor del resto.

```
> library(class)
> cls.knn <- knn(wbca[train,-1],wbca[-train,-1],wbca$Class[train],k=5)
> table(cls.knn)

cls.knn
 0   1
100 181

> table(knn=cls.knn,real=wbca$Class[-train])

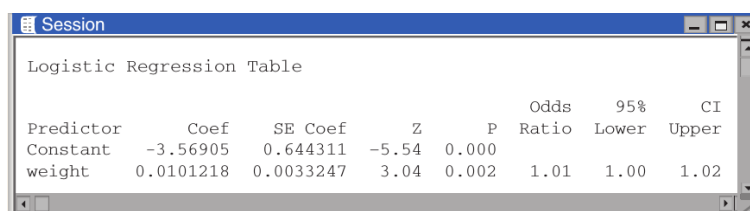
      real
knn    0   1
 0   95   5
 1    5 176
```

En este caso funciona bastante bien.

Ejercicio 10

Un estudio analizó una muestra aleatoria de personas adultas con diabetes de tipo 2 con el objetivo de identificar algunos de los factores de riesgo asociados con la enfermedad. El archivo `ex28_49.dat` contiene datos¹⁰ sobre el estado de la diabetes, peso (en libras), circunferencia de la cintura (en pulgadas) y relación de colesterol (relación entre el colesterol HDL y el colesterol total en sangre) para todos los 386 sujetos. Cincuenta y nueve de los 386 se diagnosticaron con diabetes tipo 2 (*diabetes* = 1).

La diabetes de aparición en adultos o tipo 2 se ha asociado con la obesidad. La salida de la siguiente figura muestra los resultados de una regresión logística con la variable respuesta *diabetes* y *weight* como variable explicativa. Dar la ecuación del modelo de regresión logística. ¿Es significativo el coeficiente de la pendiente? ¿El peso parece aumentar las probabilidades de tener diabetes tipo 2?



Predictor	Coef	SE Coef	Z	P	Odds Ratio	95% Lower	95% Upper
Constant	-3.56905	0.644311	-5.54	0.000			
weight	0.0101218	0.0033247	3.04	0.002	1.01	1.00	1.02

Los datos se encuentran disponibles en el archivo `ex28_49.dat`.

¹⁰Este ejercicio se ha extraído del capítulo suplementario sobre regresión múltiple y logística del libro *La práctica de la estadística para las ciencias de la vida* de Baldi y Moore.

```
> ex28_49 <- read.table("ex28_49.dat", sep=",", header = T, row.names=1)
> ex28_49$diabetes <- factor(ex28_49$diabetes, labels=c("No", "Yes"))
```

La ecuación que representa el modelo de regresión logística en este caso es:

$$\log\left(\frac{p}{1-p}\right) = -3.56905 + 0.0101218 * weight$$

de donde

$$p = \frac{\exp(-3.56905 + 0.0101218 * weight)}{1 + \exp(-3.56905 + 0.0101218 * weight)}$$

El p -valor para la variable **weight** es 0.002, por lo tanto el test de Wald indica que es significativamente diferente de cero.

La *odds ratio* es sólo de 1.01. Así pues, el hecho de aumentar en una unidad (1 pound) el peso prácticamente no aumenta la probabilidad de padecer diabetes 2.

- a) Utilizar **R** para obtener el análisis de la regresión logística con la **diabetes** como variable respuesta y **weight** como variable explicativa. Se pueden obtener los mismos resultados que los mostrados en la figura anterior, salvo errores de redondeo.

```
> lmod1 <- glm(diabetes ~ weight, data=ex28_49, family=binomial())
> summary(lmod1)
```

Call:
glm(formula = diabetes ~ weight, family = binomial(), data = ex28_49)

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.0613	-0.5856	-0.5258	-0.4385	2.1988

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-3.569051	0.644306	-5.539	3.04e-08 ***
weight	0.010122	0.003325	3.044	0.00233 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 330.12 on 385 degrees of freedom
Residual deviance: 321.01 on 384 degrees of freedom
AIC: 325.01

Number of Fisher Scoring iterations: 4

```
> exp(coef(lmod1)["weight"])

weight
1.010173

> exp(confint(lmod1,parm="weight"))

2.5 % 97.5 %
1.003592 1.016821
```

Obtener la tabla de clasificación, también llamada tabla de confusión, y calcular el error o *misclassification rate*.

```
> pred.prob <- predict(lmod1, type="response")
> pred.prob = ifelse(pred.prob > 0.5, "Yes", "No")
> table(Predicted=pred.prob, Diabetes=ex28_49$diabetes)
```

	Diabetes	
Predicted	No	Yes
No	327	59

¡Un desastre! De manera que el error es:

```
> 1-(327+0)/386

[1] 0.1528497
```

Este error basado en la proporción de los bien clasificados (*accuracy*) no refleja el desastre de esta clasificación. En cambio, la razón de los falsos negativos en este caso es 1 y, por tanto, la sensibilidad o razón de los verdaderos positivos es 0.

- b) La investigación sugiere que el lugar donde se almacena la grasa corporal es un factor importante en la predicción de la diabetes. Calcular el modelo con ambas variables, *weight* y *waist*, como explicativas. ¿Son ambas significativas? ¿Como queda la tabla de clasificación ahora?

```
> lmod2 <- glm(diabetes ~ weight+waist, data=ex28_49, family=binomial())
> summary(lmod2)
```

Call:

```
glm(formula = diabetes ~ weight + waist, family = binomial(),
    data = ex28_49)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.1793	-0.6012	-0.4753	-0.3652	2.3030

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.487312	1.081683	-5.997	2e-09 ***
weight	-0.009302	0.006379	-1.458	0.144786
waist	0.165519	0.046306	3.574	0.000351 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 330.12 on 385 degrees of freedom
 Residual deviance: 307.99 on 383 degrees of freedom
 AIC: 313.99

Number of Fisher Scoring iterations: 5

Con este modelo únicamente la variable **waist** tiene un coeficiente significativo.

La clasificación es:

```
> pred.prob <- predict(lmod2, type="response")
> pred.prob = ifelse(pred.prob > 0.5, "Yes", "No")
> table(Predicted=pred.prob, Diabetes=ex28_49$diabetes)
```

	Diabetes	
Predicted	No	Yes
No	326	58
Yes	1	1

No hemos mejorado mucho.

- c) Entre la lista de los factores de riesgo potenciales para la diabetes de tipo 2 se halla la razón de colesterol personal (**cholratio**), es decir, la razón entre el colesterol bueno o HDL y el total. Ajustar un modelo logístico con las variables **waist** y **cholratio** y estudiar el resultado.

```
> lmod3 <- glm(diabetes ~ waist+cholratio, data=ex28_49, family=binomial())
> summary(lmod3)
```

Call:

```
glm(formula = diabetes ~ waist + cholratio, family = binomial(),
    data = ex28_49)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.3892	-0.5966	-0.4420	-0.2937	2.4992

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.84900	1.11265	-6.156	7.48e-10 ***
waist	0.08822	0.02675	3.299	0.000972 ***
cholratio	0.34153	0.08902	3.837	0.000125 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 330.12 on 385 degrees of freedom

Residual deviance: 293.01 on 383 degrees of freedom

AIC: 299.01

Number of Fisher Scoring iterations: 5

```
> pred.prob <- predict(lmod3, type="response")
> pred.prob = ifelse(pred.prob > 0.5, "Yes", "No")
> table(Predicted=pred.prob, Diabetes=ex28_49$diabetes)
```

	Diabetes	
Predicted	No	Yes
No	325	55
Yes	2	4

d) Realizar un análisis LDA con las tres variables explicativas y estudiar el resultado.

```
> library(MASS)
> lda.model <- lda(diabetes ~ weight+waist+cholratio, data=ex28_49)
> lda.model
```

Call:
lda(diabetes ~ weight + waist + cholratio, data = ex28_49)

Prior probabilities of groups:

	No	Yes
	0.8471503	0.1528497

Group means:

	weight	waist	cholratio
No	175.0183	37.38226	4.334557
Yes	192.7119	41.06780	5.676271

Coefficients of linear discriminants:

	LD1
weight	-0.01217453
waist	0.16305721
cholratio	0.42878957

```
> pred.lda <- predict(lda.model)
> table(Predicted=pred.lda$class, Diabetes=ex28_49$diabetes)
```

	Diabetes	
Predicted	No	Yes
No	322	53
Yes	5	6

La regresión logística y el análisis LDA proporcionan resultados muy similares. Con estas variables explicativas es difícil discriminar la diabetes de tipo 2.