Capítulo 3

Componentes principales

3.1. Introducción

En este tema nos ocupamos de problemas de reducción de dimensión. ¿Qué significa reducir la dimensión? Responder a esta pregunta es obvio si nos fijamos en los datos que tenemos. Trabajando con expresión de genes tenemos tantas filas como genes y tantas columnas como muestras. En resumen miles de filas y decenas o centenares de columnas. En temas anteriores hemos visto como seleccionar filas, esto es, seleccionar genes es una tarea incluso previa. Hemos de quedarnos con genes que tengan una expresión diferencial si consideramos alguna característica fenotípica o bien con genes que tengan una expresión mínima o bien con genes que tengan un cierto nivel de variación. ¿Qué hacemos con las columnas? O de otro modo: ¿qué hacemos con las muestras? Quizás la respuesta natural sería: si tenemos miles de filas, ¿por qué preocuparse de unas decenas de filas? No es una buena respuesta. Realmente tener 50 o 100 columnas son muchas a la hora de visualizar resultados o bien de aplicar tratamientos estadísticos. En este tema tratamos el tema de cómo reducir el número de columnas.

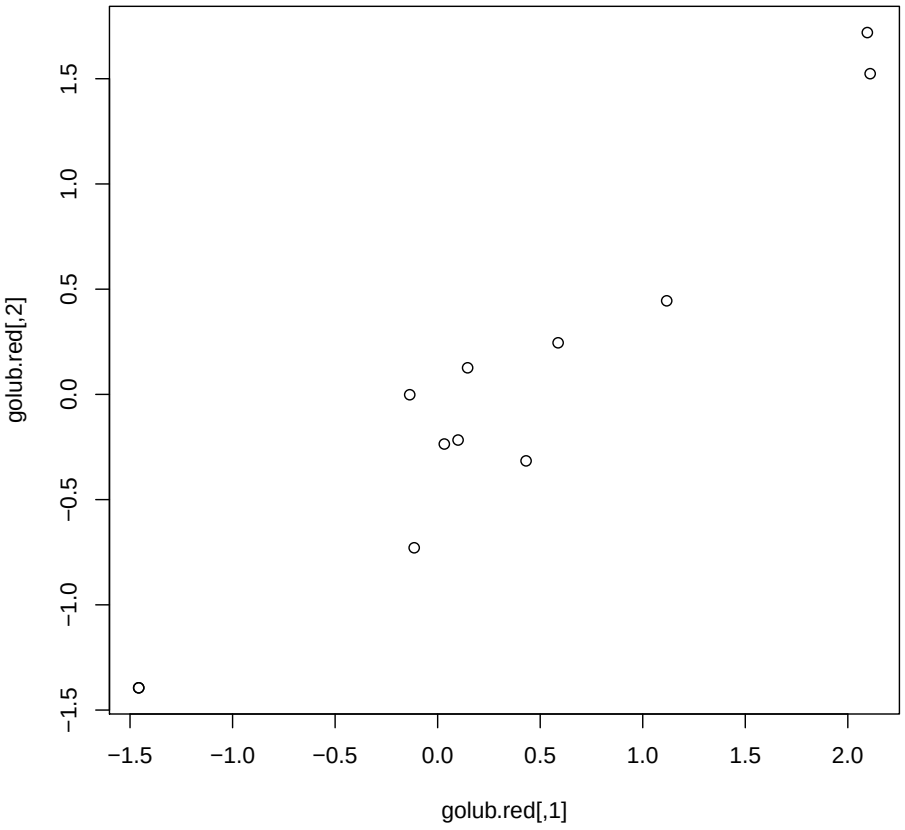
3.2. Componentes principales

Para ilustrar los conceptos vamos a considerar unos datos sencillos. Tomamos los datos *golub* y nos fijamos en los genes que tienen que ver con "Cyclin" (tienen esta palabra en su nombre). Vamos a considerar las dos primeras muestras, esto es, las dos primeras columnas.

```
library(multtest)
data(golub)
sel <- grep("Cyclin", golub.gnames[, 2])
golub.red <- golub[sel, 1:2]
```

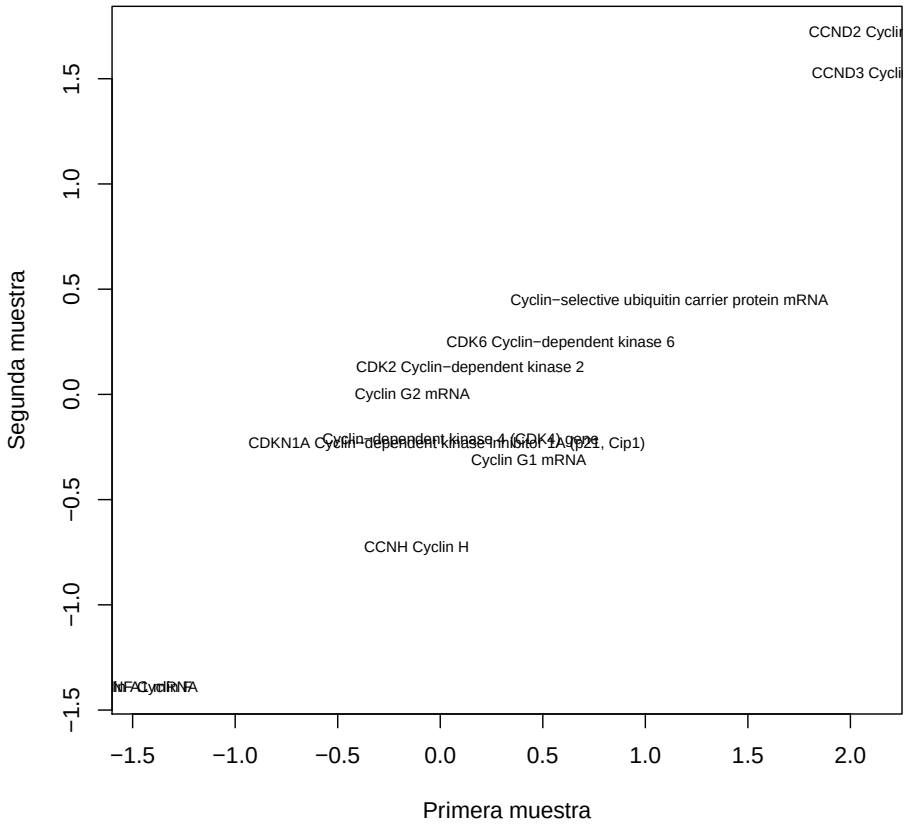
Los datos aparecen en el siguiente dibujo. Cada punto corresponde con uno de los genes seleccionados.

```
plot(golub.red)
```



Para la fila i (para el gen i) denotamos las expresiones observadas en las dos muestras como $x_i = (x_{i1}, x_{i1})$. Tenemos n filas y por lo tanto nuestros datos son x_i con $i = 1, \ldots, n$.

Vamos a repetir el dibujo anterior mostrando el nombre del gen.



Centramos los datos. Esto es, le restamos a cada columna la media de la columna. Para ello, primero calculamos las medias. El vector de medias lo vamos a denotar por $\bar{x} = (\bar{x}_1, \bar{x}_2)$ donde

$$\bar{x}_j = \sum_{i=1}^{n} \frac{x_{ij}}{n}$$

es decir, cada componente es la media de las componentes. En resumen el primer valor es la expresión media en la primera muestra para todos los genes. Podemos calcular fácilmente el vector de medias. Una función específica es la siguiente.

```
medias <- colMeans(golub.red)
```

También podemos usar la función genérica *apply* que nos hace lo mismo.

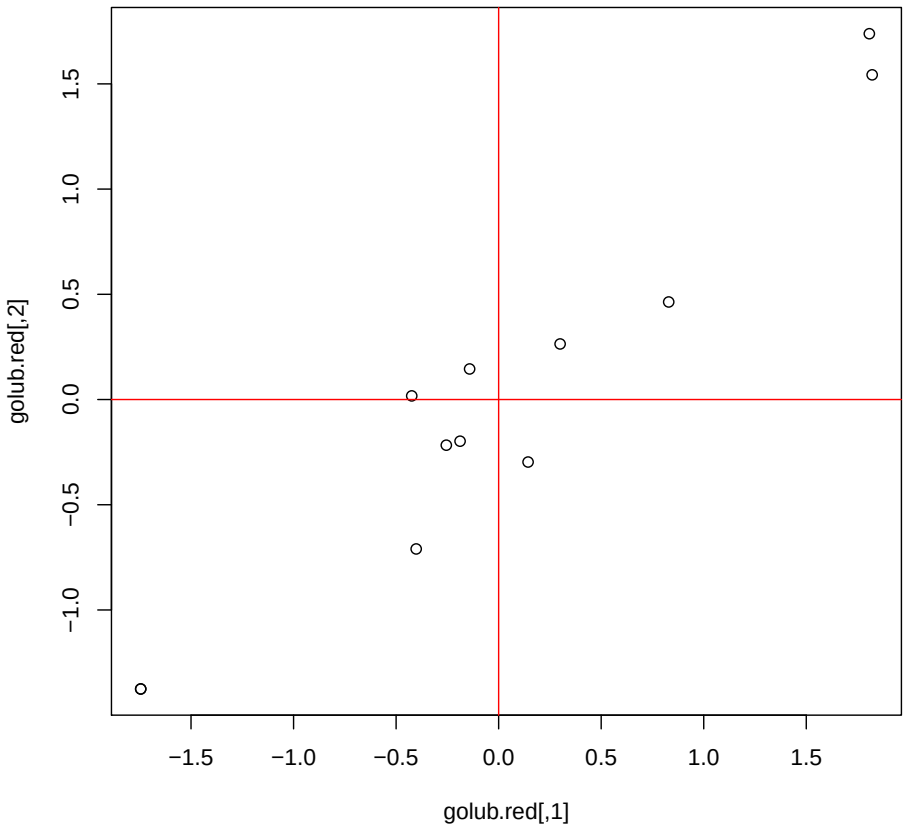
```
medias <- apply(golub.red, 2, mean)
```

Le restamos a cada columna su media.

```
golub.red <- sweep(golub.red, 2, medias)
```

En la siguiente figura reproducimos los datos centrados. Mostramos los ejes de coordenadas en rojo.

```
plot(golub.red)
abline(v = mean(golub.red[, 1]), col = "red")
abline(h = mean(golub.red[, 2]), col = "red")
```



Hemos trasladado los datos de modo que las medias de cada variable valen cero ahora. Esto es lo que se conoce como centrar los datos. Hemos *centrado los datos*. Podemos comprobar que los nuevos datos tienen una media nula.

```
colMeans(golub.red)

## [1] -3.007e-17  1.070e-17
```

Nuestros datos (filas) corresponden a las expresiones correspondientes a los genes. Los datos originales tienen dimensión 2 (dos variables correspondientes a las dos muestras) y supongamos que pretendemos reducir la dimensión a solo una, esto es, representar cada gen mediante un único número. La idea de las componentes principales es considerar una combinación lineal de los valores originales. Es decir, se pretende elegir un vector (de dimensión dos) $a_1 = (a_{11}, a_{12})$ de modo que en lugar de utilizar x_i consideremos (el resumen) $u_i = a_{11}x_{i1} + a_{12}x_{i2}$. ¿Qué a_1 elegimos? La idea es lograr que los valores u_i tengan la mayor variabilidad que se pueda con objeto de no perder información. Mantener la variabilidad original indica que mantenemos la información que los datos originales tienen. En concreto se elige a_1

de modo que maximizamos

$$\frac{1}{n}\sum_{i=1}^{n}(u_i - \bar{u})^2.$$

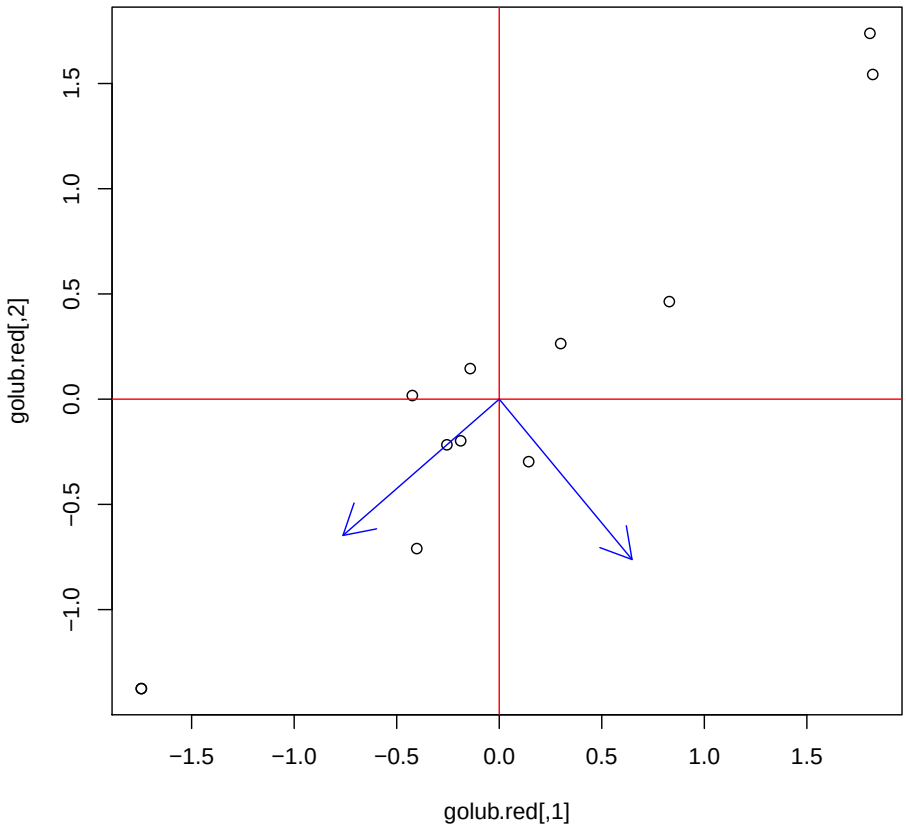
El vector a_1 nos indica la dirección sobre la cual proyectamos los datos originales. Las proyecciones sobre a_1, los valores u_i son la mejor descripción univariante de los datos.

La segunda mejor descripción que sea ortogonal a la anterior serían las proyecciones sobre la línea ortogonal a la primera que pasa por el origen de coordenadas.

Obtengamos las componentes principales.

```
a.pca <- prcomp(golub.red)
```

Vamos a representar los vectores directores de las líneas sobre las que proyectamos.

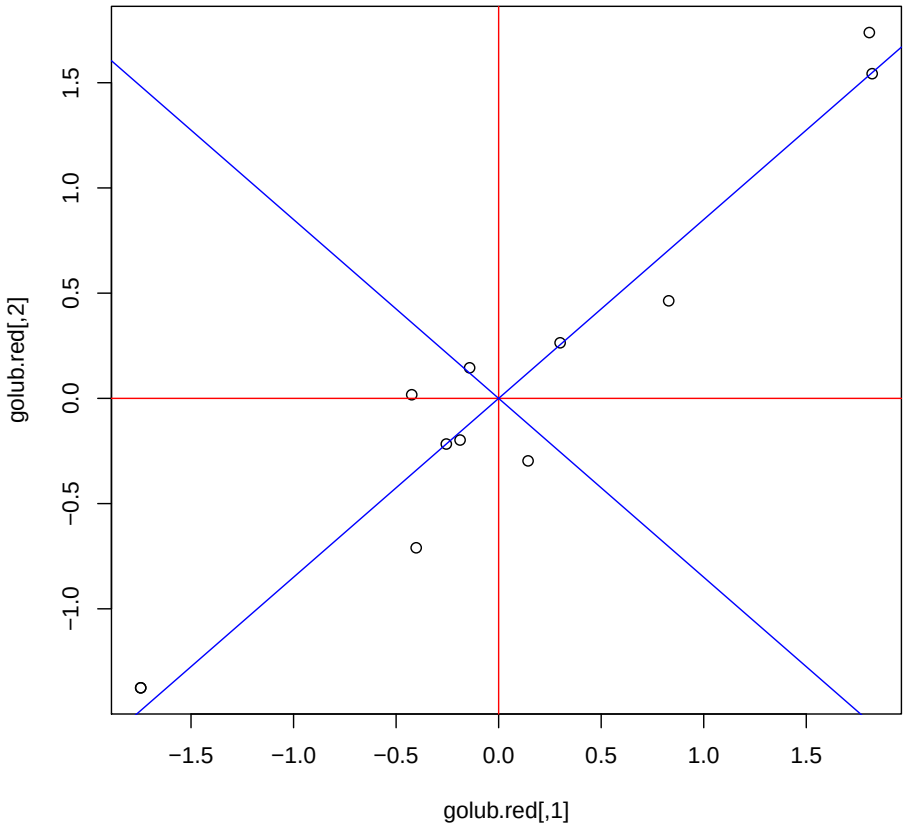


Estos vectores los podemos ver con

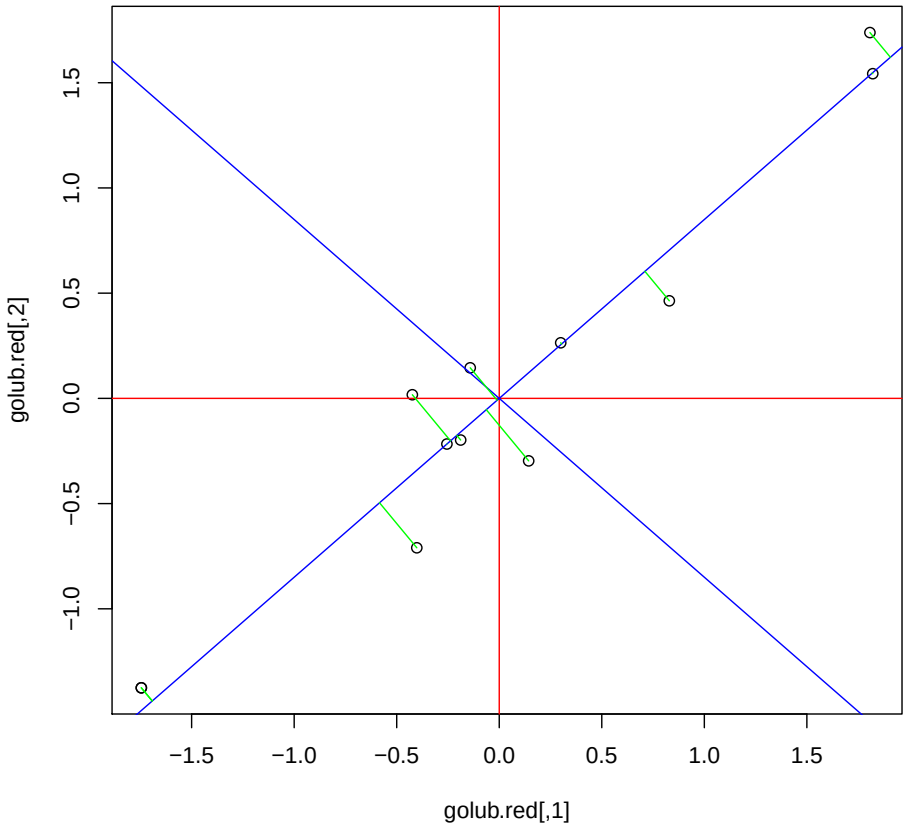
```
a.pca$rotation
```

```
##           PC1     PC2
## [1,] -0.7620  0.6476
## [2,] -0.6476 -0.7620
```

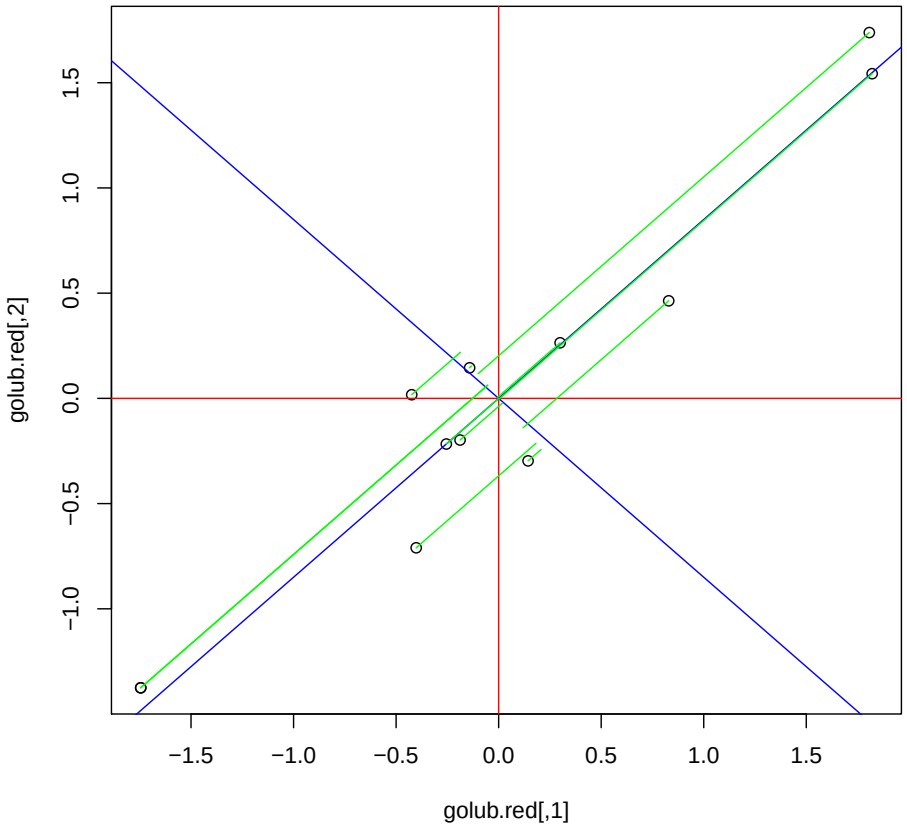
Y las líneas sobre las que proyectamos aparecen en el siguiente dibujo en azul.



Y finalmente podemos ver las proyecciones. En verde mostramos las proyecciones sobre la primera componente.



Y ahora consideremos la proyección sobre la segunda componente.



Los valores de estas proyecciones los obtenemos con

```
predict(a.pca)
```

```
##             PC1        PC2
##  [1,] -2.50309 -1.542e-01
##  [2,]  0.01369 -2.024e-01
##  [3,] -2.38702  3.714e-03
##  [4,]  0.33490 -6.847e-05
##  [5,]  0.76608  2.806e-01
##  [6,]  0.27145  2.900e-02
##  [7,]  0.31170 -2.876e-01
##  [8,]  2.22052 -8.232e-02
##  [9,] -0.93221  1.837e-01
## [10,] -0.39946 -7.240e-03
## [11,]  0.08294  3.192e-01
## [12,]  2.22052 -8.232e-02
```

Las desviaciones estándar de la primera y segunda componente principal son las siguientes

```
## [1] 1.469 0.185
```

Y las varianzas son los cuadrados de las desviaciones estándar.

```
a.pca$sdev^2

## [1] 2.15730 0.03421
```

¿Cómo de variables son nuestros datos? Podemos cuantificar el total de la variación de los datos sumando las varianzas de cada una de las dos coordenadas

```
var(golub.red[, 1])

## [1] 1.267

var(golub.red[, 2])

## [1] 0.9246
```

cuya suma es

```
var(golub.red[, 1]) + var(golub.red[, 2])

## [1] 2.192
```

Las nuevas coordenadas tienen la misma varianza total.

```
sum(a.pca$sdev^2)

## [1] 2.192
```

¿Y qué proporción de la varianza es atribuible a la primera componente? ¿Y a la segunda? Podemos dividir la varianza de cada componente por la suma total.

```
variacion.total <- sum(a.pca$sdev^2)
a.pca$sdev^2/variacion.total

## [1] 0.98439 0.01561
```

La primera componente explica un 98.44 % de la variación total. ¿Para qué necesitamos utilizar dos números por gen si con uno tenemos esencialmente la misma información.

3.3. Componentes principales de los datos golub

Hemos visto las componentes principales con dos variables (en nuestro caso dos muestras) con efecto de poder ver el significado geométrico de las componentes principales. Vamos a trabajar con el banco de datos completo: todos los datos golub que tienen 38 muestras (27 de un tipo de leucemia y 11 de otro tipo).

Obtengamos las componentes principales.

```
golub.pca <- prcomp(golub, scale = TRUE, center = TRUE)
```

El argumento `center=TRUE` centra los datos restando la media de la columna de modo que las variables tengan medias nulas. El argumento `scale=TRUE` hace que las variables originales sean divididas por su desviación estándar de modo que la varianza (y la desviación estándar) de las nuevas variables sea la unidad.

Diferentes criterios podemos aplicar a la hora de decidir con cuántas componentes nos quedamos.

1. Uno puede ser la proporción total explicada. Fijar un nivel mínimo y quedarnos con el número de componentes necesario para superar este valor mínimo.

2. El segundo puede ser que una componente no puede tener una desviación estándar menor que una de las variables originales. Si hemos escalado cada variable original dividiendo por su desviación estándar entonces la desviación estándar de cada componente ha de ser mayor que uno.

3. Otro criterio puede ser ver en qué momento se produce un descenso de la desviación estándar muy notable. Quedarnos con las componentes previas.

Un resumen de las componentes nos puede indicar con cuántas nos quedamos.

```
summary(golub.pca)

## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8    PC9
## Standard deviation     5.044 1.4407 1.1173 1.0351 0.8582 0.7440 0.7210 0.6923 0.6382
## Proportion of Variance 0.669 0.0546 0.0328 0.0282 0.0194 0.0146 0.0137 0.0126 0.0107
## Cumulative Proportion  0.669 0.7240 0.7569 0.7851 0.8045 0.8190 0.8327 0.8453 0.8561
##                           PC10    PC11    PC12    PC13    PC14    PC15    PC16    PC17
## Standard deviation     0.6363 0.56700 0.55263 0.53868 0.52011 0.49568 0.48402 0.47719
## Proportion of Variance 0.0106 0.00846 0.00804 0.00764 0.00712 0.00647 0.00617 0.00599
## Cumulative Proportion  0.8667 0.87518 0.88321 0.89085 0.89797 0.90443 0.91060 0.91659
##                           PC18    PC19    PC20    PC21    PC22    PC23    PC24    PC25
## Standard deviation     0.47068 0.45421 0.43795 0.43410 0.42475 0.41582 0.40718 0.40066
## Proportion of Variance 0.00583 0.00543 0.00505 0.00496 0.00475 0.00455 0.00436 0.00422
## Cumulative Proportion  0.92242 0.92785 0.93290 0.93786 0.94260 0.94715 0.95152 0.95574
##                           PC26    PC27    PC28    PC29    PC30    PC31   PC32   PC33
## Standard deviation     0.3948 0.38731 0.38417 0.37882 0.37124 0.36957 0.3596 0.3593
## Proportion of Variance 0.0041 0.00395 0.00388 0.00378 0.00363 0.00359 0.0034 0.0034
## Cumulative Proportion  0.9598 0.96379 0.96767 0.97145 0.97508 0.97867 0.9821 0.9855
##                           PC34    PC35    PC36    PC37    PC38
## Standard deviation     0.35276 0.34218 0.33228 0.32572 0.30667
## Proportion of Variance 0.00327 0.00308 0.00291 0.00279 0.00247
## Cumulative Proportion  0.98875 0.99183 0.99473 0.99753 1.00000
```

Atendiendo al segundo criterio nos quedaríamos con las cuatro primeras componentes. La quinta tiene una desviación inferior a uno. Atendiendo al tercer criterio vemos que a partir de la quinta es muy estable la desviación estándar. Si nos quedamos con las cinco primeras componentes estamos explicando un 80.44 % de la variación total. Puede ser una buena elección y una solución intermedia. Los nuevos datos los obtenemos con la función *predict*.

```
a <- predict(golub.pca)
```

Podemos ver todas las componentes para el primer gen (primera fila).

```
a[1, ]
```

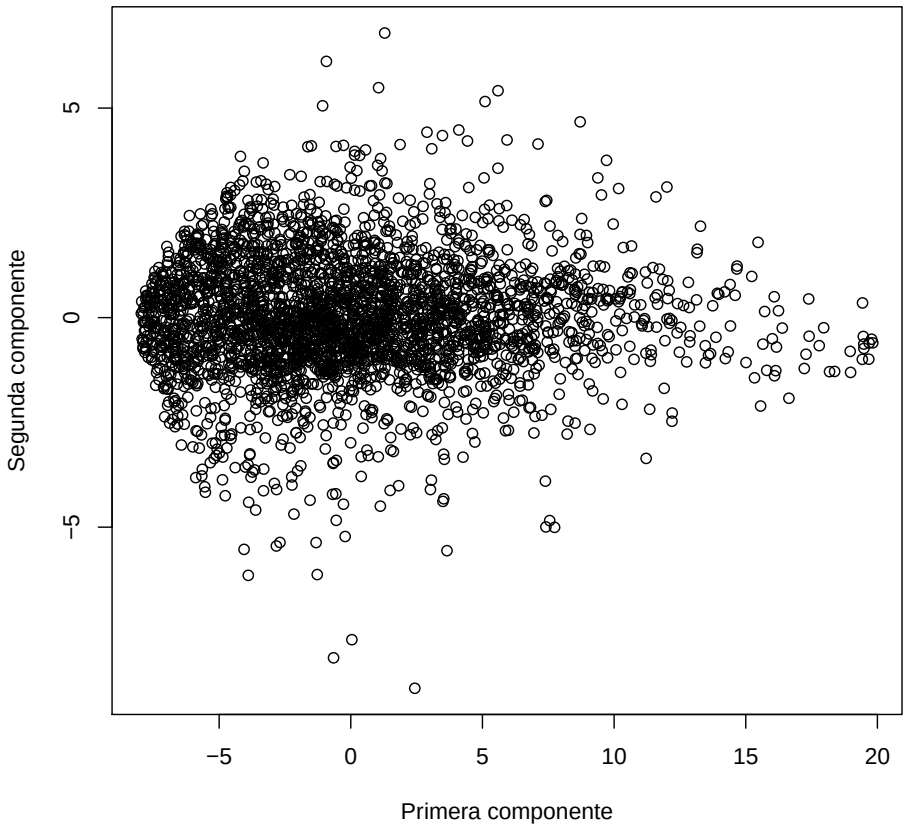
```
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8       PC9
## -7.037558 -1.611153 -0.580508  0.008742  0.538498  0.217864  0.095230 -0.918847  0.512902
##      PC10      PC11      PC12      PC13      PC14      PC15      PC16      PC17      PC18
##  0.863368 -0.199101 -0.661874  0.098494  1.167026 -0.080884  0.019310  0.311826 -0.734193
##      PC19      PC20      PC21      PC22      PC23      PC24      PC25      PC26      PC27
##  0.484427 -0.413978  0.861061  0.412109 -0.169220 -0.042500  0.392160 -0.810611 -0.724087
##      PC28      PC29      PC30      PC31      PC32      PC33      PC34      PC35      PC36
## -0.022861 -0.267373  0.223251  0.004499 -0.066890 -0.420015  0.043023  0.325942 -0.095873
##      PC37      PC38
##  0.451057  0.873975
```

Y ahora nos quedamos con las primeras cinco columnas correspondientes con las cinco primeras componentes principales como hemos decidido previamente.

```
a <- a[, 1:5]
```

Podemos representar, como es habitual, las dos primeras componentes.

```
plot(a[, 1], a[, 2], xlab = "Primera componente", ylab = "Segunda componente")
```



Es interesante observar los valores del vector asociado a la primera componente.

```
golub.pca$rotation[, 1]

##  [1] 0.1715 0.1691 0.1650 0.1727 0.1659 0.1669 0.1686 0.1602 0.1649 0.1688 0.1654 0.1694
## [13] 0.1629 0.1661 0.1648 0.1721 0.1559 0.1600 0.1677 0.1492 0.1273 0.1621 0.1644 0.1653
## [25] 0.1659 0.1690 0.1540 0.1689 0.1541 0.1517 0.1691 0.1682 0.1452 0.1675 0.1638 0.1509
## [37] 0.1476 0.1520
```

Podemos ver que son coeficientes muy parecidos, todos positivos. Básicamente tenemos la media muestral de todos los niveles de expresión en las 38 muestras. La primera componente es básicamente la media sobre las 38 muestras. ¿Y la segunda componente?

```
golub.pca$rotation[, 2]

##  [1]  0.104190 -0.036887  0.069109  0.100701  0.170952  0.028349  0.032391  0.000506
##  [9]  0.093594  0.023533  0.075376 -0.089381  0.233400  0.077939  0.237951  0.184072
## [17]  0.078197  0.041608  0.114629  0.247148  0.201580 -0.014148  0.037859  0.210586
## [25] -0.044465  0.122287  0.021439 -0.189279 -0.174593 -0.243776 -0.165316 -0.150156
## [33] -0.344035 -0.157688 -0.130649 -0.277921 -0.344829 -0.222766
```

Si observamos los coeficientes vemos que las primeros 27 valores son positivos y los 11 últimos son negativos. Además no hay una gran diferencia entre los 27 primeros y tampoco entre los 11 últimos. Básicamente estamos comparando, para cada gen, la media de los niveles de expresión sobre los datos ALL (leucemia linfoblástica aguda) con la media sobre los datos AML (leucemia mieloide aguda).

3.4. Un poco de teoría ⇑

Cuando tomamos medidas sobre personas, objetos, empresas, unidades experimentales de un modo genérico, se tiende a recoger el máximo de variables posible. En consecuencia tenemos dimensiones del vector de características X grandes.

Una opción consiste en sustituir la observación original, de dimensión d, por k combinaciones lineales de las mismas. Obviamente pretendemos que k sea mucho menor que d. El objetivo es elegir k de modo que expresen una proporción razonable de la dispersión o variación total cuantificada como la traza de la matriz de covarianza muestral, $tr(S)$,

Sea X un vector aleatorio de dimensión d con vector de medias μ y matriz de covarianzas Σ. Sea $T = (t_1, t_2, \ldots, t_d)$ (los t_i indican la i-ésima columna de la matriz) la matriz ortogonal tal que

$$T'\Sigma T = \Lambda = diag(\lambda_1, \ldots, \lambda_d), \tag{3.1}$$

donde $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d \geq 0$ son los valores propios de la matriz Σ. Sea

$$Y = T'(X - \mu). \tag{3.2}$$

Si denotamos la j-ésima componente de Y como Y_j entonces $Y_j = t_j'(X - \mu)$ con $j = 1, \ldots, d$. A la variable Y_j la llamamos la j-ésima **componente principal** de Y. La variable $Z_j = \sqrt{\lambda_j} Y_j$ es la j-ésima **componente principal estandarizada** de Y.

Estas componentes tienen algunas propiedades de gran interés.

Notemos que el vector t_j tiene longitud unitaria y, por lo tanto, Y_j no es más que la proyección ortogonal de $X - \mu$ en la dirección t_j.

Proposición 1 *1. Las variables Y_j son incorreladas y $var(Y_j) = \lambda_j$.*

2. *Las variables Z_j son incorreladas y con varianza unitaria.*

Demostración.

En cuanto al apartado primero tenemos que

$$var(Y) = var(T'(X - \mu) = T'var(Y)T = T'\Sigma T = \Lambda.$$

El segundo apartado es directo a partir del primero.

□

Se verifica el siguiente resultado.

Teorema 2 *Las componentes principales $Y_j = t_j'(X - \mu)$ con $j = 1, \ldots, d$ tienen las siguientes propiedades:*

1. *Para cualquier vector a_1 de longitud unitaria, $var(a_1'X)$ alcanza su valor máximo λ_1 cuando $a_1 = t_1$.*
2. *Para cualquier vector a_j de longitud unitaria tal que $a_j't_i = 0$ para $i = 1, \ldots, j-1$, se tiene que $var(a_j'X$ toma su valor máximo λ_j cuando $a_j = t_j$.*
3. *$\sum_{j=1}^d var(Y_j) = \sum_{j=1}^d var(X_j) = traza(\Sigma)$.*

La versión muestral de las componentes principales la obtenemos sustituyendo en lo anterior μ y Σ por $\bar{X}$ y $\hat{\Sigma}$ respectivamente. Es importante considerar el estimador de Σ que estamos utilizando (o bien el estimador insesgado donde dividimos por $n-1$ o bien el estimador en donde dividimos por n).

Si denotamos por $\hat{\lambda}_1 \geq \ldots \geq \hat{\lambda}_d$ los valores propios ordenados de $\hat{\Sigma}$ y la matriz $\hat{T} = (\hat{t}_1, \ldots, \hat{t}_d)$ es la matriz tal que cada columna es el correspondiente vector propio entonces tenemos las componentes principales muestrales dadas por $y_j = \hat{T}'(x_i - \bar{x})$. La nueva matriz de datos viene dada por

$$Y' = (y_1, \ldots, y_n) = \hat{T}'(x_1 - \bar{x}, \ldots, x_n - \bar{x}) \tag{3.3}$$

Finalmente, si las variables vienen dadas en unidades muy distintas puede ser conveniente sustituir la matriz de covarianzas (poblacional o muestral) por la correspondiente matriz de correlaciones. De hecho, una de los inconvenientes de las componentes principales como un modo de reducir la dimensión de los datos es precisamente que obtenemos resultados distintos si utilizamos las componentes principales obtenidas a partir de la matriz de covarianzas o bien las componentes principales obtenidas a partir de la matriz de correlaciones.

A partir de las d variables originales podemos obtener hasta d componentes principales. Sin embargo, hemos dicho que pretendemos reducir la dimensión del vector de datos. La pregunta a responder es: ¿con cuántas componentes nos quedamos?

Supongamos que estamos trabajando con la matriz de covarianzas Σ. Hemos de recordar que $var(y_j) = \lambda_j$ y que $\sum_{j=1}^d var(x_j) = \sum_{j=1}^d var(y_j) = \sum_{j=1}^d \lambda_j$. En consecuencia se suelen considerar los siguientes cocientes

$$\frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^d \lambda_j}, \text{ con } k = 1, \ldots, d,$$

de modo que, cuando para un cierto valor de k, estamos próximos a la unidad nos quedamos con ese valor de k. En la versión muestral trabajaremos o bien con los valores propios de la matriz de covarianzas muestral o la matriz de correlaciones muestrales.

Una referencia muy interesante sobre componentes principales es Abdi and Williams [2010].

Capítulo 4

Análisis cluster

En este tema vamos a tratar lo que en la literatura estadística recibe el nombre de *análisis cluster* [1] o, en mejor castellano, *análisis de conglomerados*. En la literatura de Inteligencia Artificial se utiliza la expresión *clasificación no supervisada*. Tenemos una muestra de observaciones multivariantes de dimensión d. ¿Qué pretendemos hacer con ellos? Encontrar grupos. Muy breve la respuesta pero: ¿qué son grupos? Imaginemos una imagen aérea fija de un patio de un colegio. En esta imagen los datos son las posiciones de los niños. ¿Se agrupan los niños formando grupos o todos están jugando con todos y los grupos son una consecuencia pasajera del juego (delanteros y defensas que aparecen agrupados en un ataque)?

Parece claro y simple el problema. Sí, lo parece. ¿Qué es un grupo? ¿Cómo defino un grupo? ¿Cuántos grupos distingo en los datos? Estamos viendo el efecto del ruido o realmente hay una estructura debajo que la vemos en un entorno con ruido.

¿Qué quiere decir encontrar grupos? Se trata de clasificar las observaciones en grupos de modo que las observaciones de un mismo grupo sean lo más similares que podamos y que los grupos entre sí sean muy distintos. El número de procedimientos que se han propuesto en la literatura es muy grande. La mayor parte de ellos no tienen un modelo probabilístico debajo, no son procedimientos *basados en modelo*. Son métodos que esencialmente utilizan el concepto de *proximidad*. Valoran de distintas formas lo próximos, lo cercanos que están los puntos, dentro de un mismo grupo y entre distintos grupos. Es pues, el primer punto a tratar: ¿cómo cuantificamos lo cerca o lejos que están los distintos puntos? En la sección 4.2 nos ocupamos de este punto. También será necesario, como veremos en el tema, valorar cuando dos conjuntos de puntos son más o menos parecidos, proximos, similares. En la misma sección nos ocupamos de ello. Supongamos que ya hemos clasificado en distintos grupos. ¿Hemos tenido éxito al hacerla? Cuando tenemos un análisis discriminante tenemos una muestra donde sabemos a qué grupo pertenece el individuo y dónde lo hemos clasificado. Esto nos permitía valorar si nuestro procedimiento clasifica bien o no. Aquí no vamos a tener esta referencia que nos da la muestra de entrenamiento. ¿Cómo valorarlo? Un concepto conocido por silueta y debido a Rousseeuw [Kaufman and Rousseeuw, 1990] nos va a servir para ello. No es ni tan simple ni tan satisfactorio como en análisis discriminante (como es de esperar si tenemos menos información para trabajar). Lo estudiamos en la sección 4.5.

Entre los muchos procedimientos de obtener los grupos a partir de los datos, los más utilizados son dos tipos: procedimientos jerárquicos y métodos de particionamiento. De los jerárquicos nos ocupamos en la sección 4.3. El método de las k-medias y el método de las k-mediodes (el castellano como siempre es muy sufrido

[1] Por cierto que la palabra cluster no existe en castellano

pues no existe la palabra) son métodos de particionamiento y los tratamos en la sección 4.4.

Una referencia muy adecuada que se puede consultar es el texto de [Kaufman and Rousseeuw, 1990]. Cualquier texto de reconocimiento de patrones es adecuado.

En lo que sigue vamos a basarnos fundamentalmente en la librería *cluster* [**?**] y denotaremos los datos a agrupar con x_i con $i = 1, \ldots n$ siendo $x_i = x_{i1}, \ldots, x_{id}$.

4.1. Algunos ejemplos

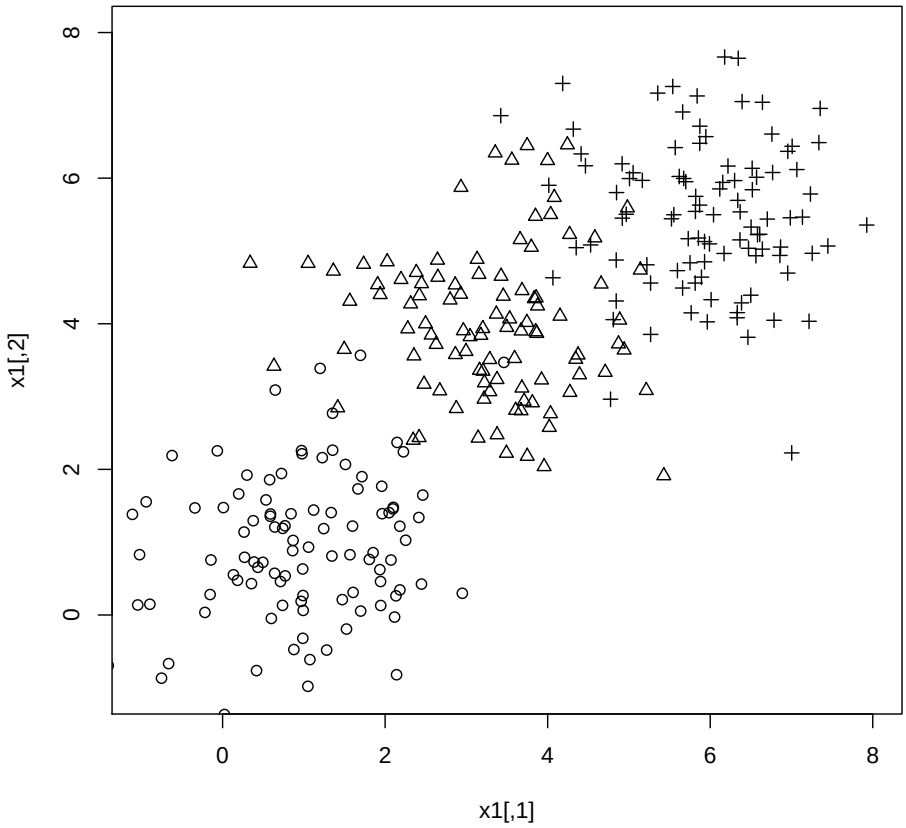
Empezamos viendo un conjunto de datos que nos sugieran el problema y cómo tratarlo.

Nota de R 9 (Un ejemplo de datos a agrupar) *Generamos tres muestras correspondientes a distribuciones bivariates normales. La matriz de covarianzas vamos a suponer que es la matriz identidad. Los vectores de medias son* $\mu_1 = c(1, 1)$, $\mu_2 = c(3, 3)$ *y* $\mu_2 = c(7, 7)$. *Es resumen un vector* $X_i \sim N_d(\mu_i, I_{2\times 2}$. *Vamos a generar cien datos de cada grupo. Vamos a utilizar el paquete* mvtnor **?** *para simular las distintas normales bivariantes.*

```
library(mvtnorm)
x1 <- rmvnorm(n = 100, mean = c(1, 1))
x2 <- rmvnorm(n = 100, mean = c(3.3, 4.1))
x3 <- rmvnorm(n = 100, mean = c(6, 5.5))
```

El siguiente dibujo muestra los datos generados.

```
limite.x <- c(-1, 8)
limite.y <- limite.x
plot(x1, xlim = limite.x, ylim = limite.y)
points(x2, pch = 2)
points(x3, pch = 3)
```



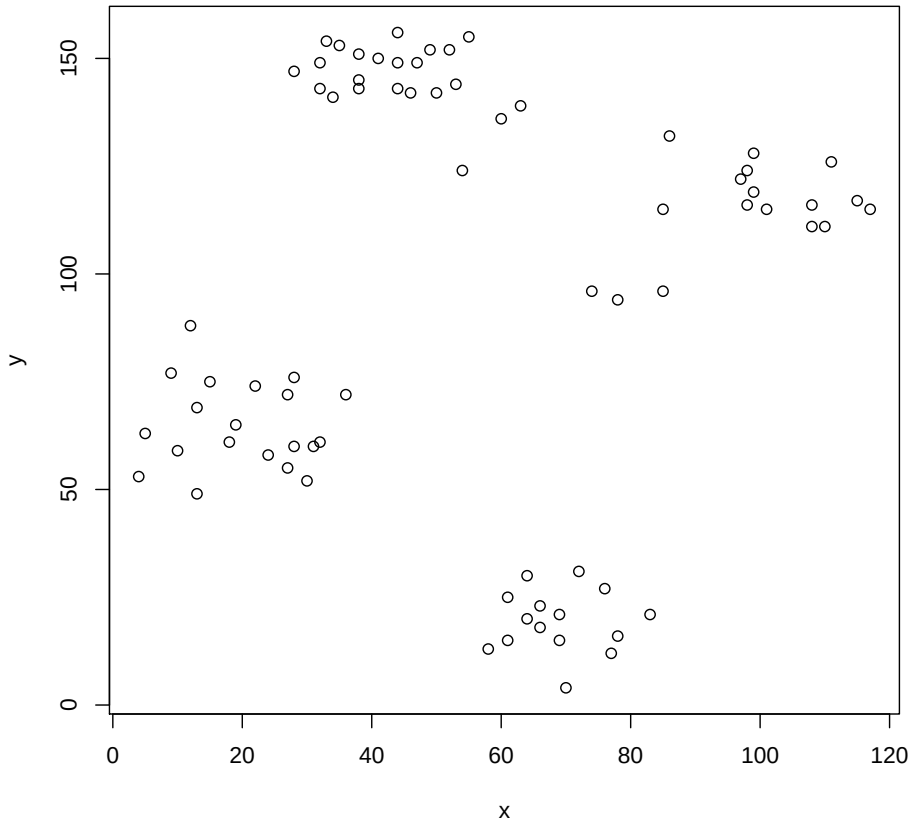
Se ve que hay tres grupos pero estos no estan muy claramente delimitados. Nosotros no disponemos de esta información. Conocemos los valores que componen los vectores de datos pero no conocemos el grupo al que podría pertenecer cada uno de ellos. Tampoco tenemos porqué tener prefijado el número de grupos. Los datos son artificiales pero ilustran bien el problema.

Nota de R 10 (Un ejemplo artificial: los datos Ruspini) *Son unos datos conocidos, los datos Ruspini. Están en el paquete* cluster **?**. *Cargamos el paquete y los datos.*

```
library(cluster)
data(ruspini)
```

Representamos los puntos que pretendemos clasificar.

```
plot(ruspini)
```



Son datos bivariantes. Visualmente vemos cómo se agrupan los puntos. Parece claro que podemos distinguir cuatro grupos.

Nota de R 11 (Un ejemplo con los datos golub) *Empezamos cargando los datos.*

```
library(multtest)
data(golub)
```

Previamente hemos visto que los valores de expresión de los genes "CCND3 Cyclin D3" y "Zyxin" permiten diferenciar entre ALL y AML. Localicemos las expresiones correspondientes a estos genes.

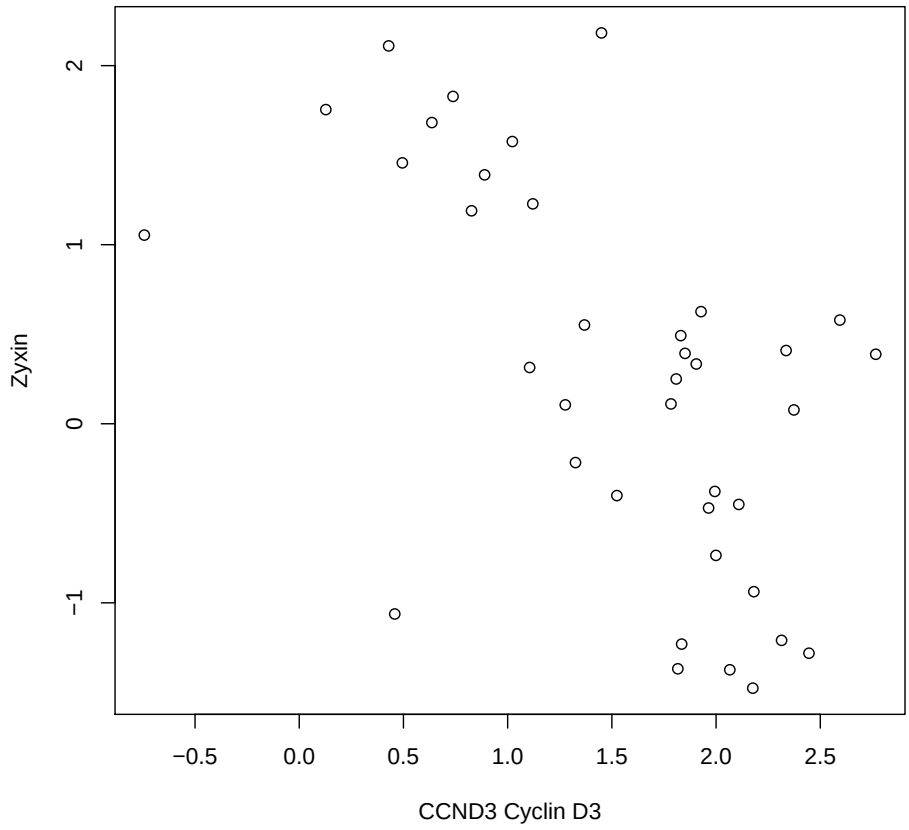
```
grep("CCND3 Cyclin D3", golub.gnames[, 2])

## [1] 1042

grep("Zyxin", golub.gnames[, 2])

## [1] 2124
```

Los datos aparecen en estas filas. Por lo tanto podemos construir la matriz de datos correspondiente.

```
cz.data <- data.frame(golub[1042, ], golub[2124, ])
colnames(cz.data) <- c("CCND3 Cyclin D3", "Zyxin")
```

Este será un segundo ejemplo para analizar. Veamos los datos.

```
plot(cz.data)
```



En este caso las observaciones corresponden a las muestras y las variables son los niveles de expresión de dos genes. ¿Hay grupos? Esto no son datos artificiales como los de Ruspini y ya no es tan claro.

4.2. Disimilaridades

4.2.1. Disimilaridades entre observaciones

Empezamos tratando el problema de cuantificar el grado de proximidad, de similaridad entre dos puntos en el espacio de dimensión d. Tradicionalmente este tema en Matemáticas se ha formalizado a través del concepto de *distancia* o *métrica*. Una métrica es una función que a cada par de puntos $x, y \in \mathbb{R}^d$ le asocia un valor positivo de modo que cuando mayor es más *distantes* son, más alejados están. Como siempre la formalización matemática de un concepto intuitivo ha de ser prudente y pedir que se verifiquen ciertos axiomas que resulten razonables y generalmente

admisibles. En concreto la función d definida en el espacio producto $\mathbb{R}^d \times \mathbb{R}^d$ se dice que es una métrica si verifica:

No negativa $d(x, y) \geq 0$.

Un punto dista 0 de sí mismo $d(x, x) = 0$.

Simetría $d(x, y) = d(y, x)$.

Desigualdad triangular $d(x, z) \leq d(x, y) + d(y, z)$, para todo $x, y, z \in \mathbb{R}^d$.

Las distancias más utilizadas en análisis cluster son la distancia euclídea y la distancia de Manhattan. Para dos vectores x e y (en $\mathbb{R}^d$) entonces la distancia euclídea se define como

$$d(x, y) = \sqrt{\sum_{k=1}^{d} (x_k - y_k)^2}, \tag{4.1}$$

con $x, y \in \mathbb{R}^d$. La distancia de Manhattan viene dada por

$$d(x, y) = \sum_{k=1}^{d} |x_k - y_k|. \tag{4.2}$$

Las distancias euclídea y de Manhattan son adecuadas cuando trabajamos con variables continuas y que además estén en una misma escala. [2] Notemos que cada una de las componentes del vector pesan igualmente. Si tenemos variables que no están igualmente escaladas estas distancias pueden pesar más unas variables que otras no por lo diferentes que sean entre los individuos sino simplemente por su escala.

Con mucha frecuencia nos encontramos trabajando con variables que aún siendo continuas están medidas en escalas muy diversas o bien tenemos variables que son continuas, otras que son binarias, otras categóricas con más de dos categorías o bien variable ordinales. En resumen, todos los posibles tipos de variables simultáneamente considerados. Es lo habitual. Una variable binaria la codificamos habitualmente como 1 y 0 indicando presencia o ausencia del atributo que estemos considerando. En una variable categórica la codificación es completamente arbitraria y por lo tanto no tiene sentido la aplicación de una de estas distancias.

Todo esto plantea el hecho de que no es recomendable, ni tampoco fácil, en un banco de datos con distintos tipos de variables considerar una métrica o distancia, esto es, algo que verifique las propiedades anteriores. Son demasiado exigentes estas propiedades. Lo que se hace es proponer medidas entre los vectores de características que tienen algunas de las propiedades y son, además, razonables. Por ello no hablaremos, en general, de una distancia o una métrica, sino de una medida de disimilaridad. Finalmente, valores grandes estarán asociados con vectores de características que tienen una mayor diferencia. Se han propuesto distintas medidas de disimilaridad entre variables cualitativas (binarias simétricas o asimétricas, cualitativas, ordinales) y cuantitativas. En lo que sigue comentamos con algún detalle la que lleva la función *daisy* de la librería *cluster* **?**. Es una opción muy genérica y razonable.

Consideramos el problema cuando solo tenemos un tipo de variable. Finalmente combinaremos todo en una sola medida de disimilaridad.

Supongamos descrita al individuo o caso mediante d variables binarias. Es natural construir la tabla de contingencia 2×2 que aparece en la tabla 4.1 donde las filas corresponden con el individuo i y las columnas con el individuo j. Según la tabla

[2] La función *dist* de **?** es una buena opción para el cálculo de estas y otras distancias. También lo es la función *daisy* del paquete *cluster* **?**.

Cuadro 4.1: Conteos asociados a dos casos descritos por variables binarias

	1	0	
1	A	B	A+B
0	C	D	C+D
	A+C	B+D	d=A+B+C+D

los individuos i y j coincidirían en la presencia de A atributos y en la no presencia de D atributos. Tenemos B atributos en i que no están en j y C atributos que no están en i pero sí que están en j.

El total de variables binarias es de $d = A+B+C+D$. Basándonos en esta tabla se pueden definir distintas medidas de disimilaridad. Vamos a considerar dos situaciones distintas. En la primera trabajamos con variables binarias simétricas y otra para variables binarias no simétricas. Una variable binaria es simétrica cuando las dos categorías que indica son intercambiables, cuando no tenemos una preferencia especial en qué resultado lo codificamos como 1 y qué resultado codificamos como 0. Un ejemplo frecuente es el sexo de la persona. Si las variables son todas binarias simétricas es natural utilizar como disimilaridad el *coeficiente de acoplamiento simple* definido como

$$d(i,j) = \frac{B+C}{A+B+C+D} = 1 - \frac{A+D}{A+B+C+D}.$$

La interpretación de esta medida de disimilaridad es simple. Dos individuos son tanto más disimilares cuantas más variables binarias tienen distintas. Notemos que la presencia o ausencia de un atributo tienen el mismo peso.

Supongamos que las variable que describen al individuo son binarias asimétricas. Ejemplos de esto pueden ser la presencia o ausencia de un atributo muy poco frecuente. Por ejemplo, tener o no tener sida. Dos personas que tienen el sida, tienen más es común, están más próximas, que dos personas que no lo tienen. Supongamos que codificamos el atributo menos frecuente como 1 y el más frecuente como 0. Está claro que un acoplamiento 1-1 o acoplamiento positivo es más significativo que un acoplamiento negativo o acoplamiento 0-0 por lo que A, número de acoplamientos positivos, ha de tener más peso que d o número de acoplamientos negativos. El más conocido es el *coeficiente de Jaccard* que se define como

$$d(i,j) = \frac{B+C}{A+B+C} = 1 - \frac{A}{A+B+C}$$

en el que simplemente no consideramos los acoplamientos negativos.

Consideremos ahora el caso de variables categóricas con más de dos categorías. Si todas las variables son de este tipo y tenemos un total de d variables entonces los individuos i y j son tanto más disimilares cuanto más variables categóricas son distintas. Si denotamos por u el número de variables en las que coinciden los dos individuos entonces la medida de disimilaridad sería

$$d(i,j) = \frac{d-u}{d}.$$

Finalmente veamos cómo tratar las variables ordinales. Lo que haremos para variables de este tipo es transformarlas al intervalo $[0,1]$. Si x_{ij} denota la j-ésima variable del i-ésimo individuo entonces consideramos la transformación

$$y_{ik} = \frac{x_{ik}-1}{M_k - 1}$$

siendo $1, \ldots, M_k$ los valores que puede tomar la j-ésima variable ordinal. Lo que estamos haciendo con este procedimiento es transformar la variable ordinal es una variable numérica con una escala común. En la medida en que el número de categorías sea mayor esta transformación tendrá más sentido.

Hemos visto cómo tratar cada tipo de variable aisladamente. El problema es combinar todas ellas en una sola medida de disimilaridad. La función *daisy* del paquete *cluster* **?** utiliza la siguiente medida:

$$d(i,j) = \frac{\sum_{k=1}^{d} \delta_{ij}^{(k)} d_{ij}^{(k)}}{\sum_{k=1}^{d} \delta_{ij}^{(k)}}, \tag{4.3}$$

donde:

- $\delta_{ij}^{(k)}$ vale uno cuando las medidas x_{ik} y x_{jk} no son valores faltantes y cero en otro caso;
- $\delta_{ij}^{(k)}$ vale 0 cuando la variable k es binaria asimétrica y tenemos entre los individuos i y j un acoplamiento 0-0;
- el valor $d_{ij}^{(k)}$ es lo que contribuye a la disimilaridad entre i y j la variable k.
  - Si la variable k es binaria o categórica entonces $d_{ij}^{(k)}$ es definida como $d_{ij}^{(k)} = 1$ si $x_{ik} \neq x_{jk}$ y 0 en otro caso.
  - Si la variable k es numérica entonces
    $$d_{ij}^{(k)} = \frac{|x_{ik} - x_{jk}|}{R_k}$$
    siendo R_k el rango de la variable k definido como
    $$R_k = \max_h x_{hk} - \min_h x_{hk}$$
    donde h varía entre todos los individuos con valor no faltante de la variable k.

Si todas las variables son categóricas entonces 4.3 nos da el número de acoplamientos del total de pares disponibles, en definitiva, el coeficiente de acoplamiento simple. Si todas son variables binarias simétricas entonces obtenemos otra vez el coeficiente de acoplamiento simple. Si las variables son binarias asimétricas entonces obtenemos el coeficiente de Jaccard. Cuando todas las variables con las que trabajamos son numéricas la medida de disimilaridad es la distancia de Manhattan donde cada variable está normalizada.

Dado un conjunto de datos x_i con $i = 1, \ldots, n$ tendremos, utilizando algunas de las medidas de disimilaridades comentadas, una matriz de dimensión $n \times n$ que tiene en la posición (i,j) la disimilaridad entre x_i y x_j, $d(i,j)$: $[d(i,j)]_{i,j=1,\ldots,n}$.

Con esta matriz cuantificamos la disimilaridad que hay entre los elementos originales de la muestra. Algunos de los procedimientos de agrupamiento que vamos a considerar en lo que sigue no necesitan conocer los datos originales. Pueden aplicarse con solo conocer esta matriz de disimilaridades. Otros no. Otros utilizan los datos a lo largo de las distintas etapas de aplicación del procedimiento.

4.2.2. Disimilaridades entre grupos de observaciones

En algunos procedimientos de agrupamiento (en particular, los jerárquicos) vamos a necesitar calcular disimilaridades entre conjuntos disjuntos de las observaciones originales. Estas disimilaridades las podemos calcular a partir de las disimilaridades originales punto a punto que hemos comentado en la sección 4.2.1.

Supongamos que tenemos un banco de datos con n individuos cuyos índices son $\{1, \ldots, n\}$. Sean A y B dos subconjuntos disjuntos del conjunto de índices de la muestra $\{1, \ldots, n\}$, esto es, dos subconjuntos de observaciones disjuntos. ¿Cómo podemos definir una disimilaridad entre A y B partiendo de las disimilaridades entre los datos individuales? Se han propuesto muchos procedimientos. Si denotamos la disimilaridad entre A y B como $d(A, B)$ entonces las disimilaridades más habitualmente utilizadas son las siguientes:

Enlace simple La disimilaridad entre los dos grupos es el mínimo de las disimilaridades entre las observaciones de uno y de otro. Tomamos la disimilaridad de los objetos que más se parecen en uno y otro grupo.

$$d(A, B) = \min_{a \in A, b \in B} d(a, b)$$

Enlace completo Ahora tomamos como disimilaridad entre los grupos como el máximo de las disimilaridades, en definitiva, la disimilaridad entre los objetos más alejados o más distintos.

$$d(A, B) = \max_{a \in A, b \in B} d(a, b)$$

Promedio La disimilaridad es el promedio de las disimilaridades entre todos los posibles pares.

$$d(A, B) = \frac{1}{|A| \times |B|} \sum_{a \in A, b \in B} d(a, b)$$

donde $|A|$ es el cardinal del conjunto A.

Es importante notar que solamente necesitamos conocer las disimilaridades entre los individuos para poder calcular las disimilaridades entre grupos de individuos.

En la siguiente sección nos vamos a ocupar de los métodos jerárquicos en los cuales es fundamental el procedimiento que elijamos para calcular distintas entre grupos.

4.3. Cluster jerárquico

La idea de estos procedimientos es construir una jerarquía de particiones del conjunto de índices.

Sea $\{1, \ldots, n\}$ el conjunto de índices que indexan las distintas observaciones. Supongamos que $\{C_1, \ldots, C_r\}$ es una partición de este conjunto de índices:

- $C_i \subset \{1, \ldots, n\}$; son disjuntos dos a dos, $C_i \cap C_j = \emptyset$ si $i \neq j$ con $i, j = 1, \ldots, n$ y
- $\cup_{i=}^{r} C_i = \{1, \ldots, n\}$.

Dada una partición del conjunto de índices podemos calcular la matriz $r \times r$ que en la posición (i, j) tiene la disimilaridad entre el conjunto C_i y C_j, $d(C_i, C_j)$ según alguno de los procedimientos antes indicados.

Veamos los procedimientos jerárquicos aglomerativos. En estos procedimientos vamos a iniciar el agrupamiento con la partición: $C_i = \{i\}$ con $i = 1, \ldots, n$, es decir, cada grupo es un individuo. En cada iteración vamos agrupando el par de conjuntos (elementos de la partición que tengamos en esa iteración) que estén *más próximos* según la disimilaridad entre grupos que estemos utilizando. El proceso continúa hasta que tengamos un único grupo.

Un esquema algorítmico del procedimiento indicado puede ser el siguiente:

Paso 0 Tenemos grupos unitarios formados por cada una de las observaciones. Tenemos pues una partición inicial $C_i = \{i\}$ con $i = 1, \ldots, n$. En un principio, cada dato es un grupo.

Paso 1 Calculamos las disimilaridades entre los elementos de la partición. Para ello utilizamos cualquiera de los procedimientos antes indicados.

Paso 2 Agrupamos los dos conjuntos de la partición más próximos y dejamos los demás conjuntos igual. Tenemos ahora C_i con $i = 1, \ldots, k$.

Paso 3 Si tenemos un solo conjunto en la partición paramos el procedimiento.

Paso 4 Volvemos al paso 1.

Hay una representación gráfica muy utilizada para describir los resultados de un cluster jerárquico aglomerativo como el que acabamos de describir. Esta representación tiene el nombre de **dendograma**. En el dendograma se va mostrando a qué valor de la medida de disimilaridad se produce la unión de los grupos y simultáneamente qué grupos se están uniendo para esa disimilaridad. También nos permite una valoración rápida de cuántos grupos puede haber en el banco de datos. Simplemente trazando una linea horizontal a la altura en que tengamos el número de grupos que *pensamos* que puede haber.

Nota de R 12 *Veamos un ejemplo de análisis cluster. Los datos han sido obtenidos de esta página. Tenemos cuatro variables que nos dan las puntuaciones obtenidas en 25 escuelas de New Haven en aritmética y lectura al principio del cuarto curso y al principio del sexto curso. Empezamos cargando el paquete* cluster **?** *y leyendo los datos.*

```
library(cluster)
x <- read.table("../data/achieve.txt")
names(x) <- c("centro", "lec4", "aritme4", "lec6", "aritme6")
attach(x)
```

Eliminamos la primera columna en que aparece el nombre de la escuela.

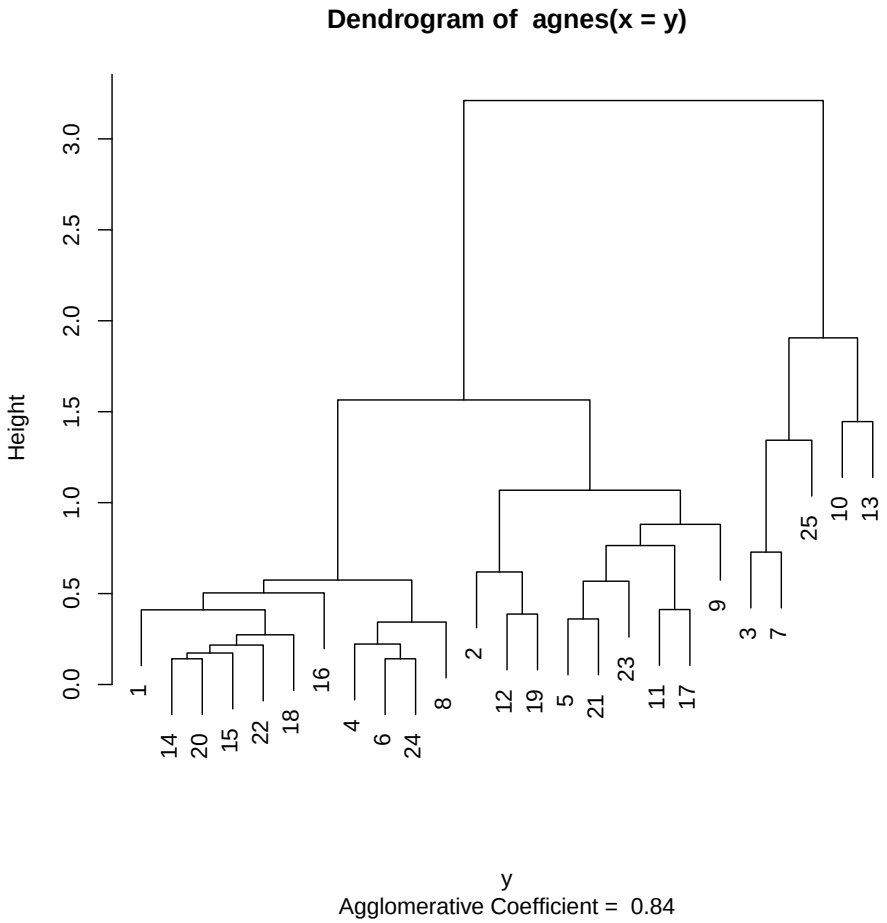
```
y <- x[, -1]
```

Hacemos un análisis cluster jerárquico utilizando la función agnes *del paquete* cluster.

```
y.ag <- agnes(y)
```

Veamos el dendograma.

```
plot(y.ag, which = 2)
```



Observando el dendograma parece razonable considerar tres grupos.

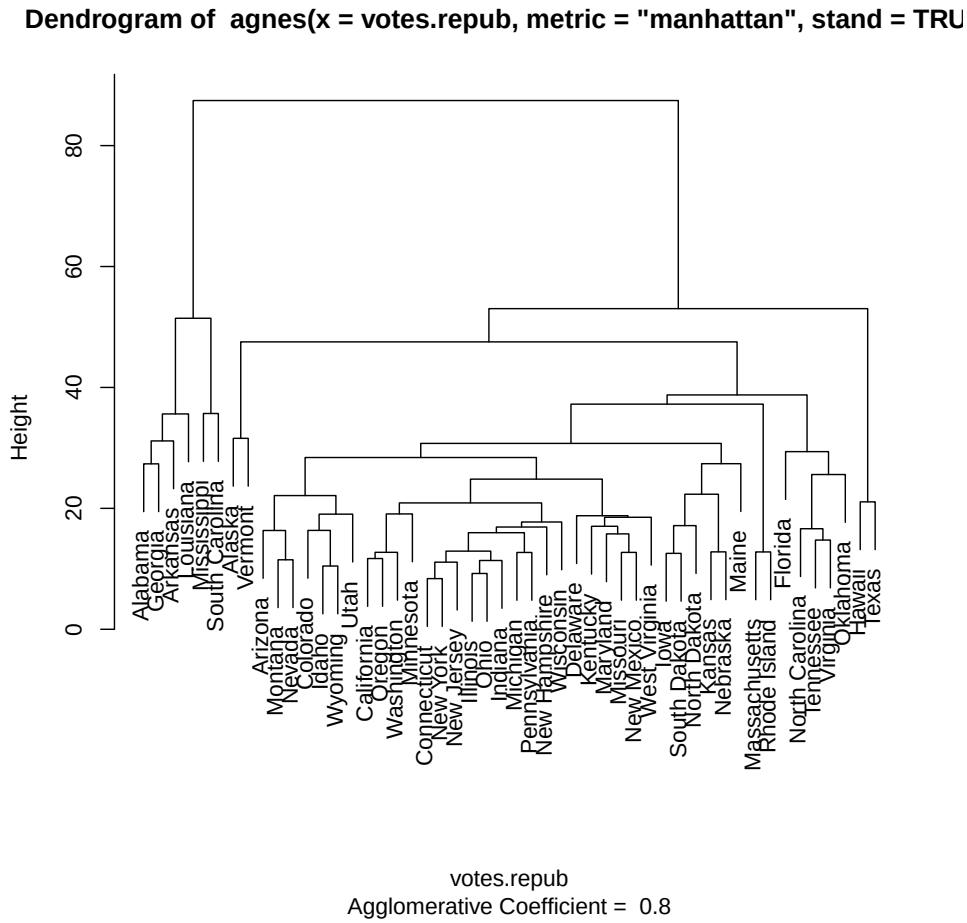
@

Nota de R 13 (Un ejemplo con votos republicanos) *Veamos un análisis cluster jerárquico. Los datos son los porcentajes de votas que recibió el candidato republicano en las elecciones americanas entre los años 1856 y1976. Cada observación corresponde a un estado y las variables corresponden con las distintas elecciones.*

```
library(cluster)
data(votes.repub)
agn1 <- agnes(votes.repub, metric = "manhattan", stand = TRUE)
```

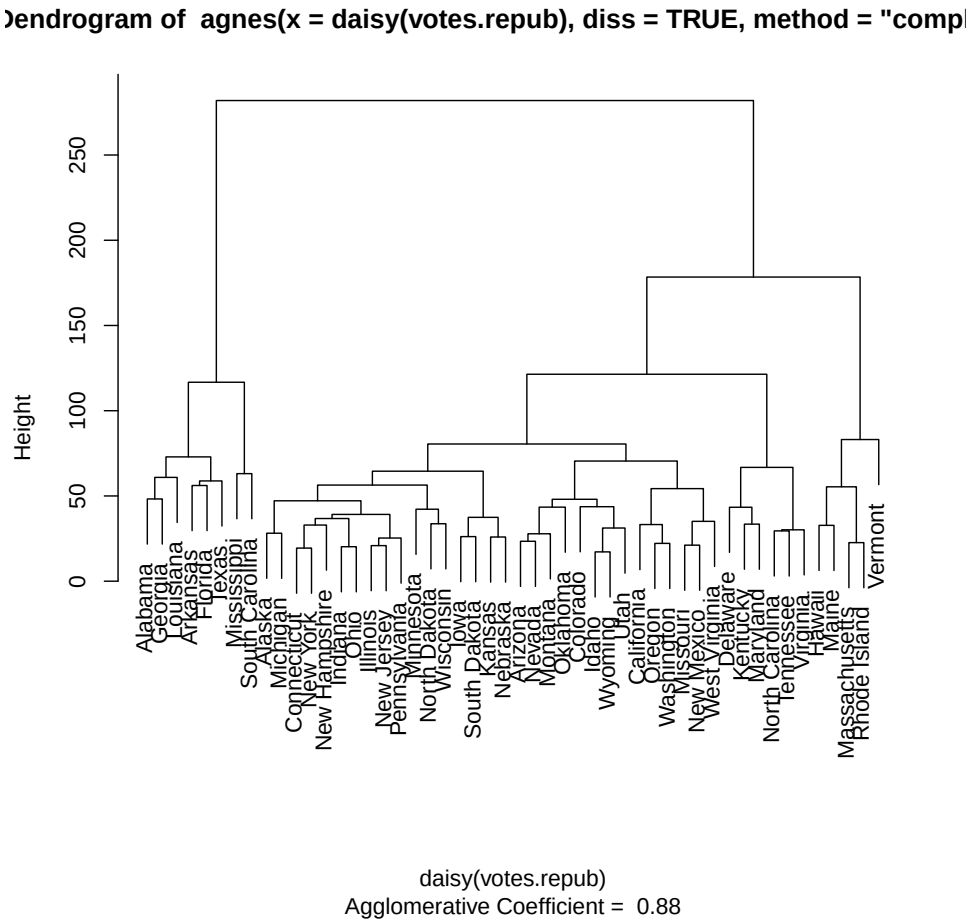
El dendograma sería el siguiente

```
plot(agn1, which = 2)
```



Pasamos ahora la matriz de distancias cambiamos el procedimiento para el cálculo de las disimilaridades entre grupos.

```
agn2 <- agnes(daisy(votes.repub), diss = TRUE, method = "complete")
plot(agn2, which = 2)
```

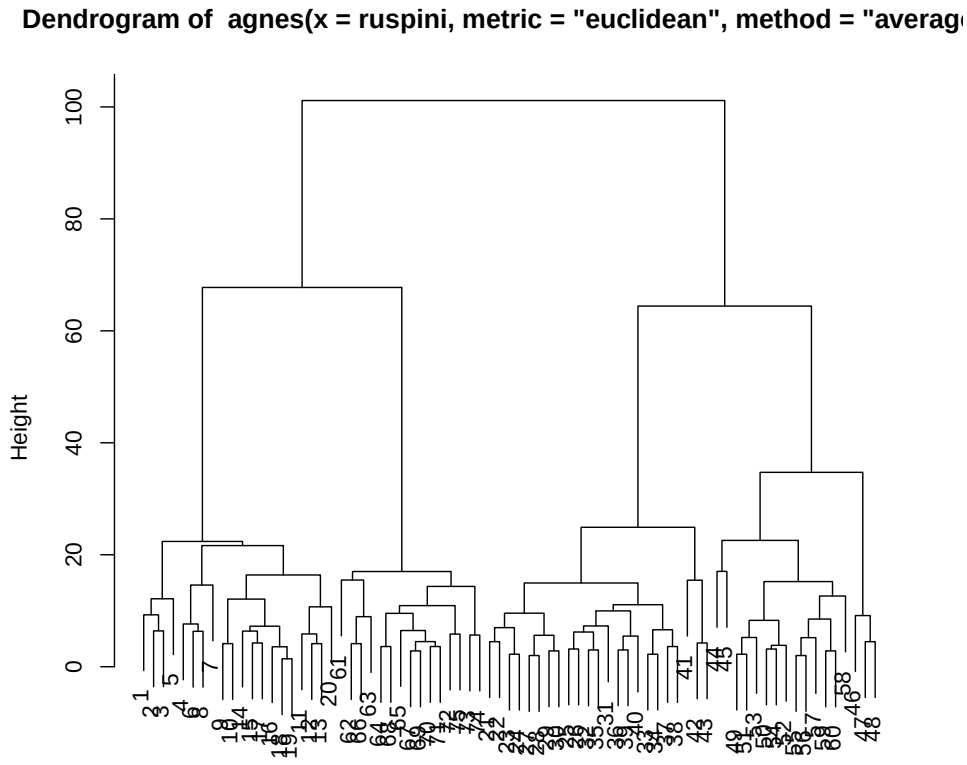


Nota de R 14 *Consideremos los datos ruspini. Apliquemos un cluster jerárquico aglomerativo utilizando como disimilaridad entre grupos el promedio de las disimilaridades y como medida de disimilaridad la distancia euclídea.*

```
ruspini.ag <- agnes(ruspini, metric = "euclidean", method = "average")
```

Representamos el dendograma.

```
plot(ruspini.ag, which = 2)
```



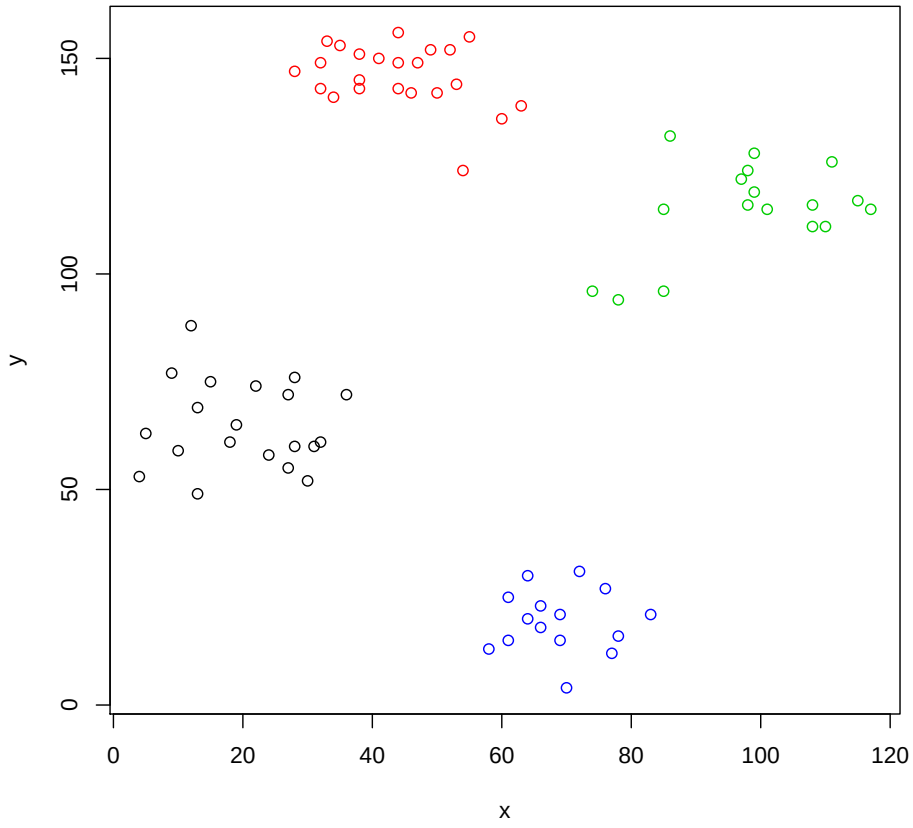
Supongamos que decidimos quedarnos con cuatro grupos. Las clasificaciones de los datos son las siguientes.

```
cutree(ruspini.ag, 4)
```

```
##  [1] 1 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 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [44] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
```

Y ahora podemos representar los datos de modo que ponemos en un mismo color los datos que hemos clasificado en un grupo.

```
plot(ruspini, col = cutree(ruspini.ag, 4))
```



Un procedimiento jerárquico aglomerativo tiene interés cuando el banco de datos no tiene muchos individuos. En otro caso es muy difícil de interpretar un dendograma o de seleccionar el número de grupos que hay. En todo caso lo hemos de ver como un análisis exploratorio para decidir el número de grupos y posteriormente utilizar algún método de particionamiento como los que vamos a ver.

4.4. Métodos de particionamiento

Suponemos ahora que tenemos una idea de cuántos grupos hay. Posiblemente hemos realizado un análisis jerárquico previo con todos los datos o, si eran muchos, con una selección aleatoria de los datos. Tenemos pues una idea de cuántos grupos tendremos que considerar. Obviamente podemos luego evaluar los resultados modificando el número de grupos. En principio, vamos a suponer que fijamos el número de grupos a considerar. Suponemos pues que *sabemos* el número de grupos y lo denotamos por k.

4.4.1. Método de las k-medias

El primer procedimiento que vamos a ver es el método de las *k-medias* (que por alguna extraña razón en la literatura de Inteligencia Artificial se le llama de las *c-medias* lo que demuestra que cada persona copia a sus amigos o, simplemente, conocidos). Supongamos que tenemos $C_1, \ldots, C_k$ una partición de $\{1, \ldots, n\}$. Un

modo bastante natural de valorar la calidad de del agrupamiento que la partición nos indica sería simplemente considerar la siguiente función.

$$\sum_{i=1}^{k} \sum_{j \in C_i} d_E(x_j, \bar{x}_{C_i})^2, \tag{4.4}$$

donde d_E denota aquí la distancia euclídea y

$$\bar{x}_{C_i} = \frac{1}{|C_i|} \sum_{j \in C_i} x_j, \tag{4.5}$$

es el vector de medias del grupo cuyos índices están en C_i. Una partición será tanto mejor cuanto menor sea el valor de la función dada en 4.4. El procedimiento de agrupamiento de las k-medias simplemente se basa en elegir como partición de los datos aquella que nos da el *mínimo* de la función objetivo considerada en ecuación 4.4. Notemos que en muchos textos se hablan del algoritmo de las k-medias y se identifica con un procedimiento concreto para encontrar el mínimo de la función. Aquí entendemos el procedimiento como la minimización de la función objetivo. De hecho, *R* ofrece hasta cuatro posibles procedimientos de los muchos que cabe proponer. Hay que diferenciar claramente el procedimiento del método de aplicación del mismo, del método de obtención de dicho mínimo.

Es importante darnos cuenta de que el procedimiento que acabamos de ver está basado en la utilización de la distancia euclídea y en que, dado un grupo, podemos calcular el vector de medias y esto solo lo podemos hacer si todas las variables son cuantitativas.

Nota de R 15 *Aplicamos el k-medias.*

```
ruspini.km <- kmeans(ruspini, 4)
```

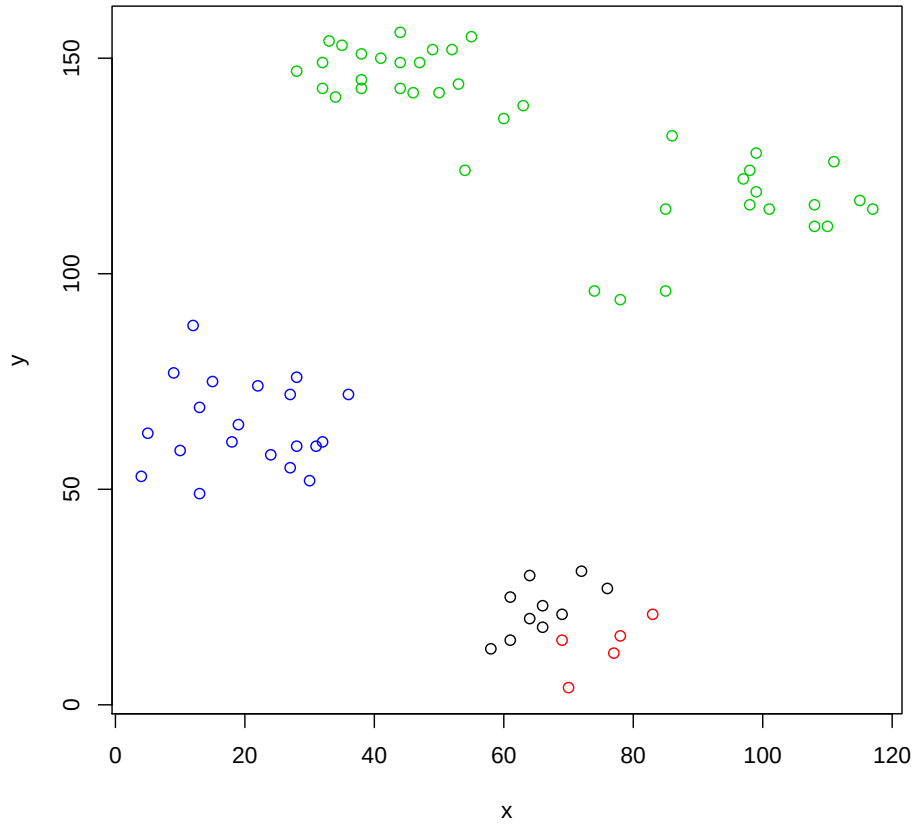
Las clasificaciones son

```
ruspini.km$cluster
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
##  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  3  3  3  3  3  3  3  3  3  3
## 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
##  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3
## 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
##  2  2  2  1  2  2  1  1  1  1  1  1  1  1  1
```

Y representamos los resultados.

```
plot(ruspini, col = ruspini.km$cluster)
```



Como vemos buenos resultados. Con ruspini todo va bien.

Nota de R 16 (cz.data) *Empezamos con el k-medias.*

```
cz.km <- kmeans(cz.data, 2)
```

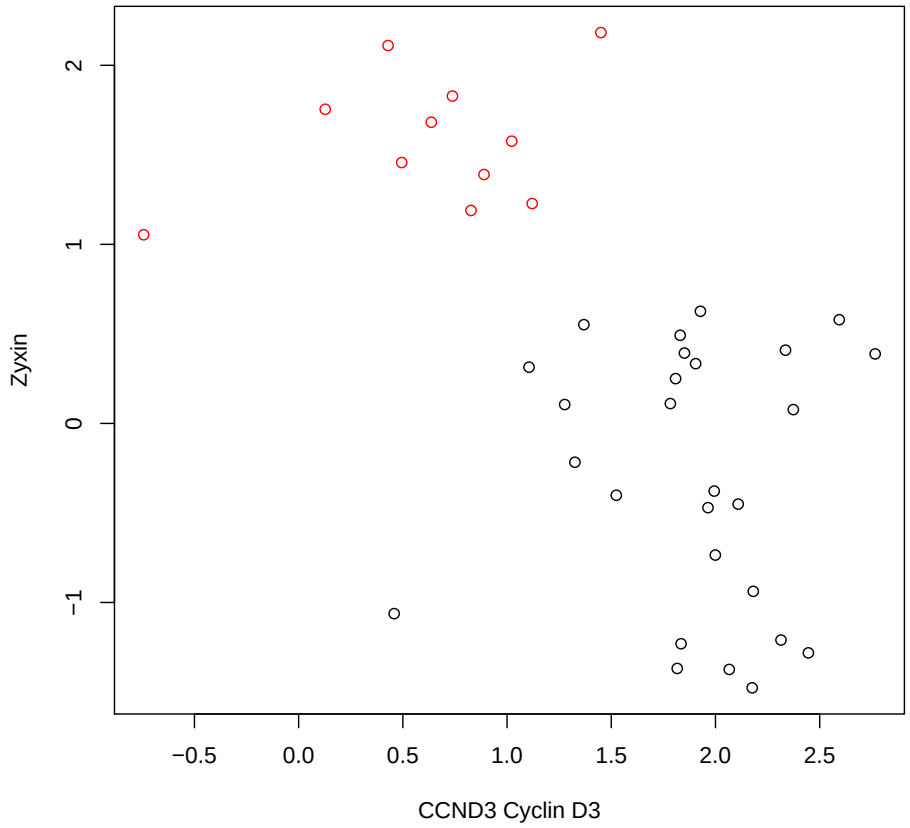
Las clasificaciones son

```
cz.km$cluster
```

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

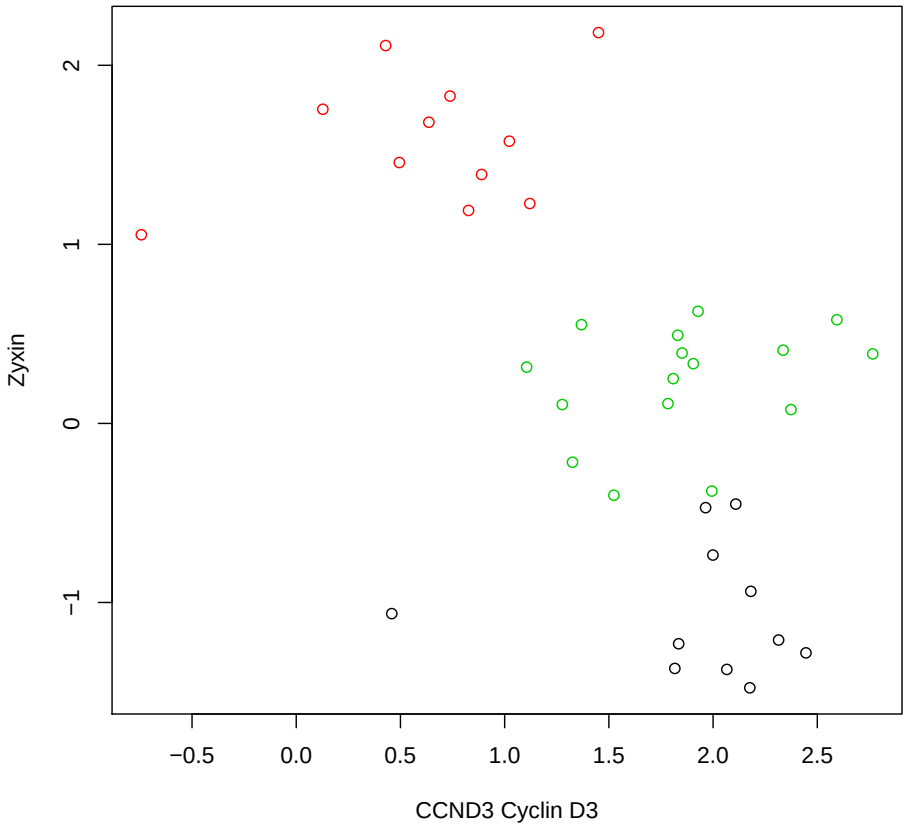
Y representamos los resultados.

```
plot(cz.data, col = cz.km$cluster)
```



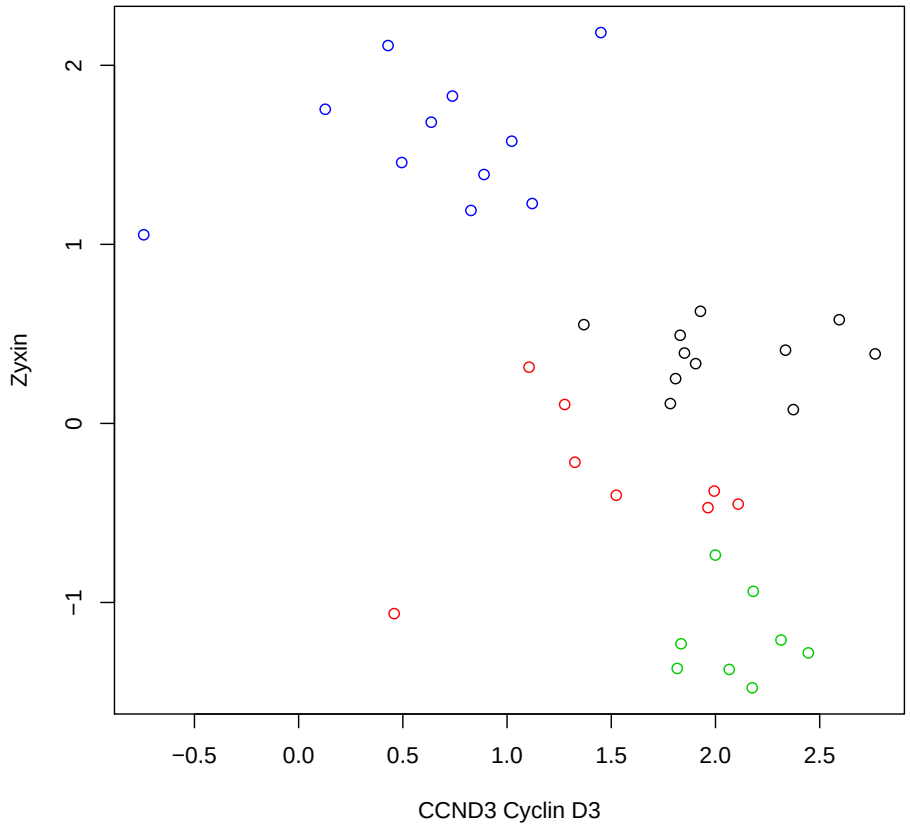
Supongamos que probamos con tres grupos.

```
cz.km <- kmeans(cz.data, 3)
plot(cz.data, col = cz.km$cluster)
```



Y finalmente con cuatro.

```
cz.km <- kmeans(cz.data, 4)
plot(cz.data, col = cz.km$cluster)
```



4.4.2. Particionamiento alrededor de los mediodes

¿Y si no podemos calcular el vector de medias? ¿Y si no tiene sentido calcular el vector de medias? ¿Cómo promediar dos configuraciones de puntos distintas? ¿Cómo promediamos dos formas distintas descritas numéricamente? Cuando el concepto de promedio aritmético no tiene sentido podemos generalizar el procedimiento anterior y hablar de (perdón por el neologismo) de *k-mediodes*.

La idea ahora es sustituir esos centros calculados como vectores de medias de los individuos de un mismo grupo por individuos bien centrados, por individuos típicos, que sustituyan a las medias.

Supongamos que tomamos k individuos de la muestra que denotamos por m_i con $i = 1, \ldots, k$. Particionamos la muestra en k grupos de modo que el grupo C_i está formado por los individuos más próximos a m_i que a cualquier otro m_j con $j \neq i$,

$$C_i = \{l : d(l, i) = \min_{j \neq i} d(l, j)\}.$$

Consideremos la siguiente cantidad:

$$\sum_{i=1}^{k} \sum_{j \in C_i} d(j, m_i). \tag{4.6}$$

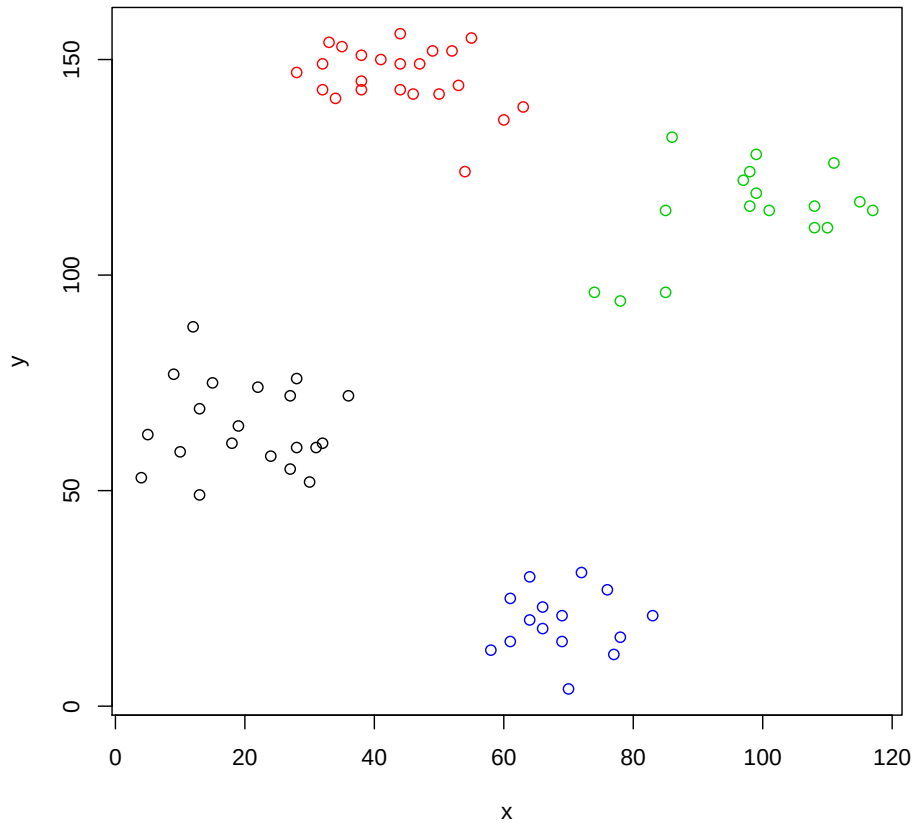
En el método de particionamiento alrededor de los mediodes nos planteamos encontrar las observaciones $m_1, \ldots, m_k$ que minimizan el valor dado en 4.6.

Nota de R 17 (ruspini) *Aplicamos PAM.*

```
ruspini.pam <- pam(ruspini, 4)
```

Y representamos los resultados.

```
plot(ruspini, col = ruspini.pam$cluster)
```



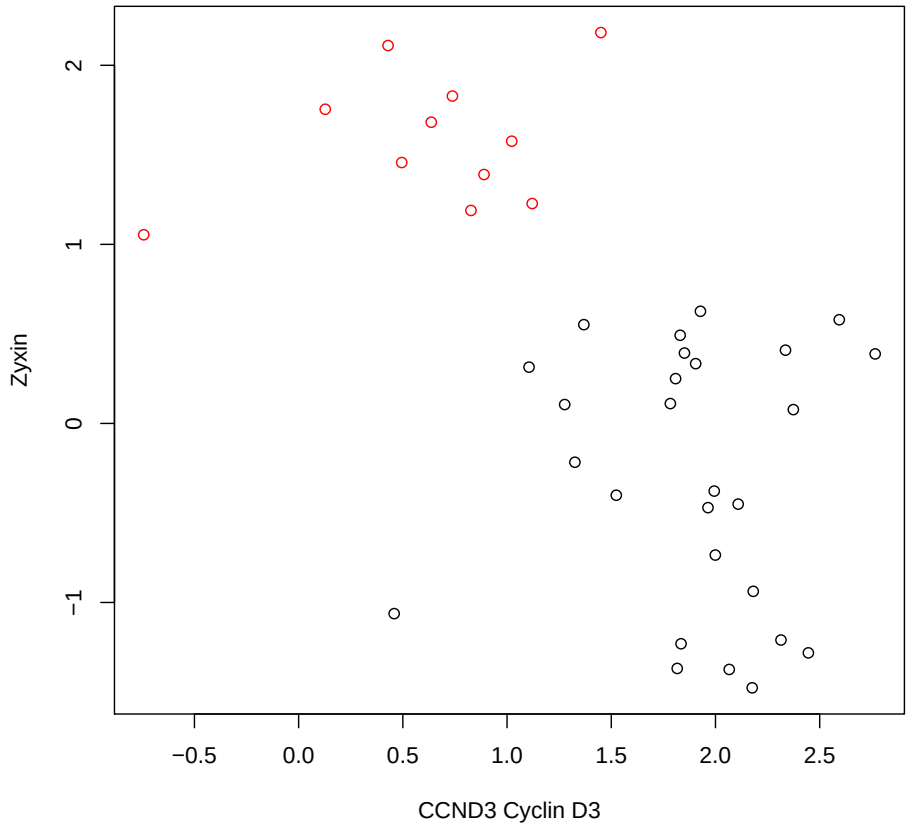
Lo dicho, con ruspini todo va bien.

Nota de R 18 (cz.data) *Empezamos con el k-medias.*

```
cz.pam <- pam(cz.data, 2)
```

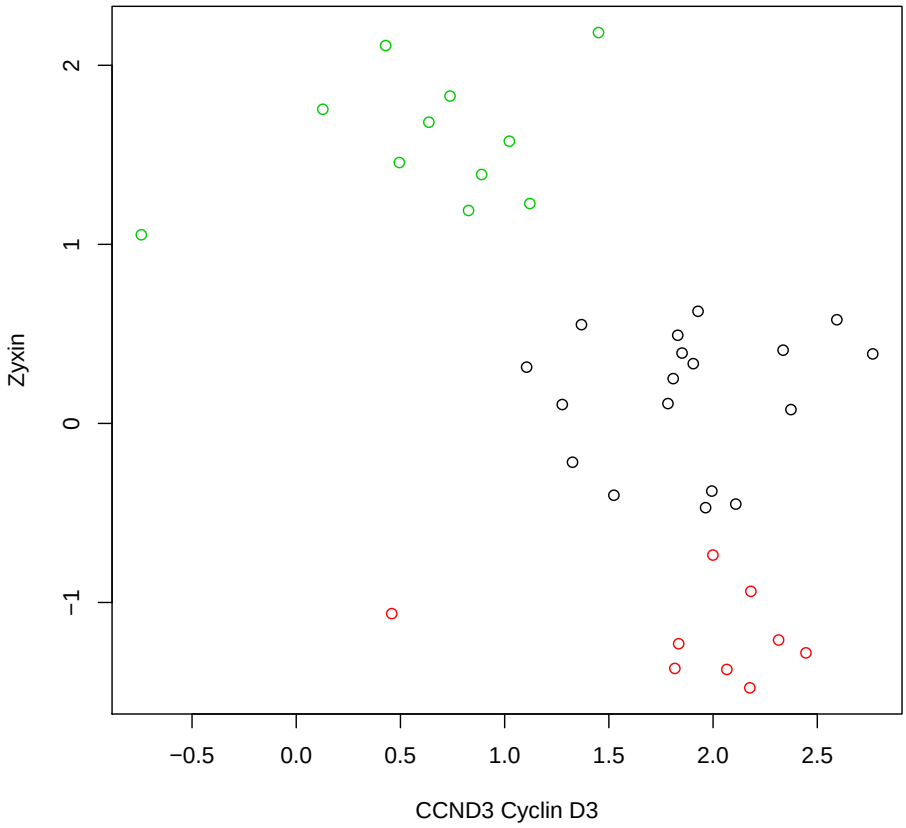
Y representamos los resultados.

```
plot(cz.data, col = cz.pam$cluster)
```



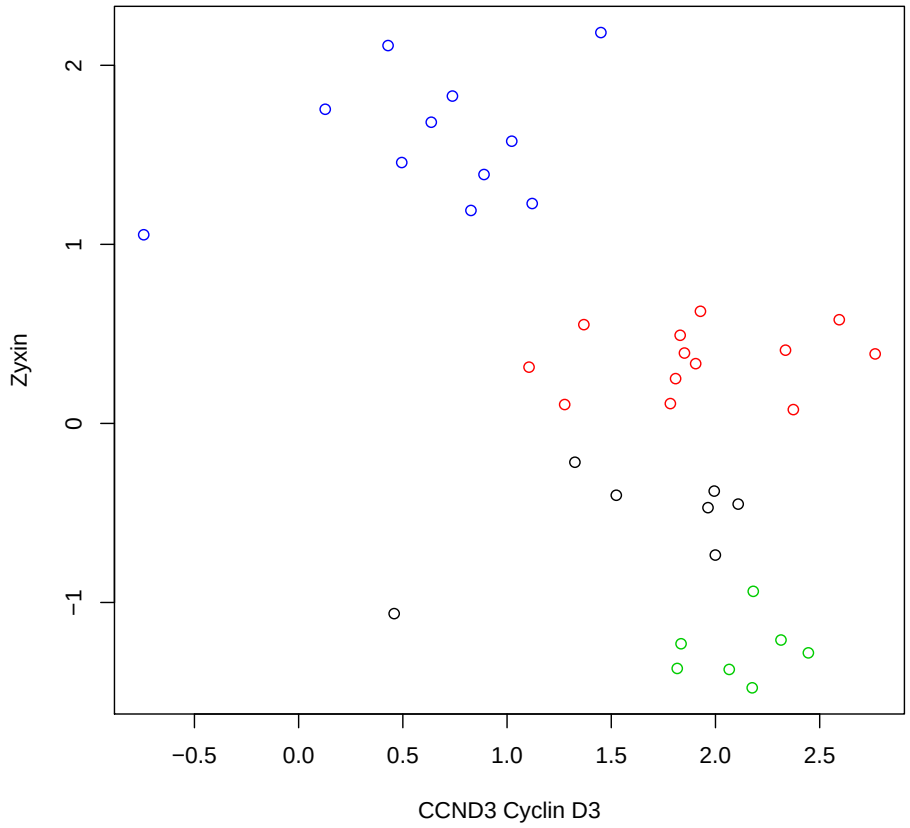
Supongamos que probamos con tres grupos.

```
cz.pam <- pam(cz.data, 3)
plot(cz.data, col = cz.pam$cluster)
```



Y finalmente con cuatro.

```
cz.pam <- pam(cz.data, 4)
plot(cz.data, col = cz.pam$cluster)
```



4.5. Silueta

Veamos cómo se construye la silueta. Para la observación i y el grupo C consideramos

$$\bar{d}(i, C) = \frac{1}{|C|} \sum_{j \in C} d(i, j),$$

la disimilaridad media i con los elementos del grupo C. Para cada observación i, sea A el cluster al cual lo ha asignado el procedimiento cluster que empleamos y calculamos $a(i)$ la disimilaridad media de i con todos los demás individuos del grupo A, $a(i) = \bar{d}(i, A)$. Obviamente estamos asumiendo que A contiene al menos otro objeto. Consideremos $\bar{d}(i, C)$ para todos los grupos $C \neq A$ y seleccionemos el que tiene el mínimo valor:

$$b(i) = \min_{C \neq A} \bar{d}(i, C).$$

Cuadro 4.2: Silueta media y estructura en un conjunto de datos

SC	Interpretación
$0{,}71 - 1{,}00$	Fuerte estructura
$0{,}51 - 0{,}70$	Estructura razonable
$0{,}26 - 0{,}50$	Estructura débil. Probar otros métodos
$\leq 0{,}25$	No se encuentra estructura

El grupo B donde se alcanza este mínimo, es decir, $\bar{d}(i, B) = b(i)$ se le llama vecino del objeto i. [3] Definimos $s(i)$ como

$$s(i) = 1 - \frac{a(i)}{b(i)} \text{ si } a(i) < b(i), \tag{4.7}$$

$$= 0 \text{ si } a(i) = b(i), \tag{4.8}$$

$$= \frac{b(i)}{a(i)} - 1 \text{ si } a(i) > b(i). \tag{4.9}$$

Esto se puede expresar en una única ecuación como

$$s(i) = \frac{b(i) - a(i)}{\text{máx}\{a(i), b(i)\}}.$$

En el caso en que el grupo A contenga un único objeto no está muy claro cómo definir $a(i)$. Tomaremos $s(i) = 0$ que es una elección arbitraria. Se comprueba con facilidad que $-1 \leq s(i) \leq 1$ para cualquier objeto i.

Para interpretar el significado de $s(i)$ es bueno ver los valores extremos. Si $s(i)$ es próximo a uno significa que $a(i)$ es mucho menor que $b(i)$ o lo que es lo mismo, que el objeto i está bien clasificado pues la disimilaridad con los de su propio grupo es mucho menor que la disimilaridad con los del grupo más próximo que no es el suyo. Un valor próximo a cero significa que $a(i)$ y $b(i)$ son similares y no tenemos muy claro si clasificarlo en A o en B. Finalmente un valor de $s(i)$ próximo a -1 significa que $a(i)$ es claramente mayor que $b(i)$. Su disimilaridad media con B es menor que la que tiene con A. Estaría mejor clasificado en B que en A. No está bien clasificado.

Los valores de $s(i)$ aparecerán representados para cada cluster en orden decreciente. Para cada objeto se representa una barra horizontal con longitud proporcional al valor $s(i)$. Una buena separación entre grupos o cluster viene indicada por unos valores positivos grandes de $s(i)$. Además de la representación gráfica se proporciona un análisis descriptivo. En concreto la media de los valores de la silueta dentro de cada cluster y la media de la silueta para todo el conjunto de datos. La clasificación será tanto mejor cuanto mayor sean estos valores medios. De hecho, se puede decidir el número de grupos en función del valor medio de las silueta sobre toda la muestra. Vamos probando distintos números de grupos y nos quedamos con el número que nos da la silueta media máxima.

¿Cuándo podemos decir que hay estructura de grupos en los datos que estamos analizando? Experiencias con datos sugieren la tabla 4.2.

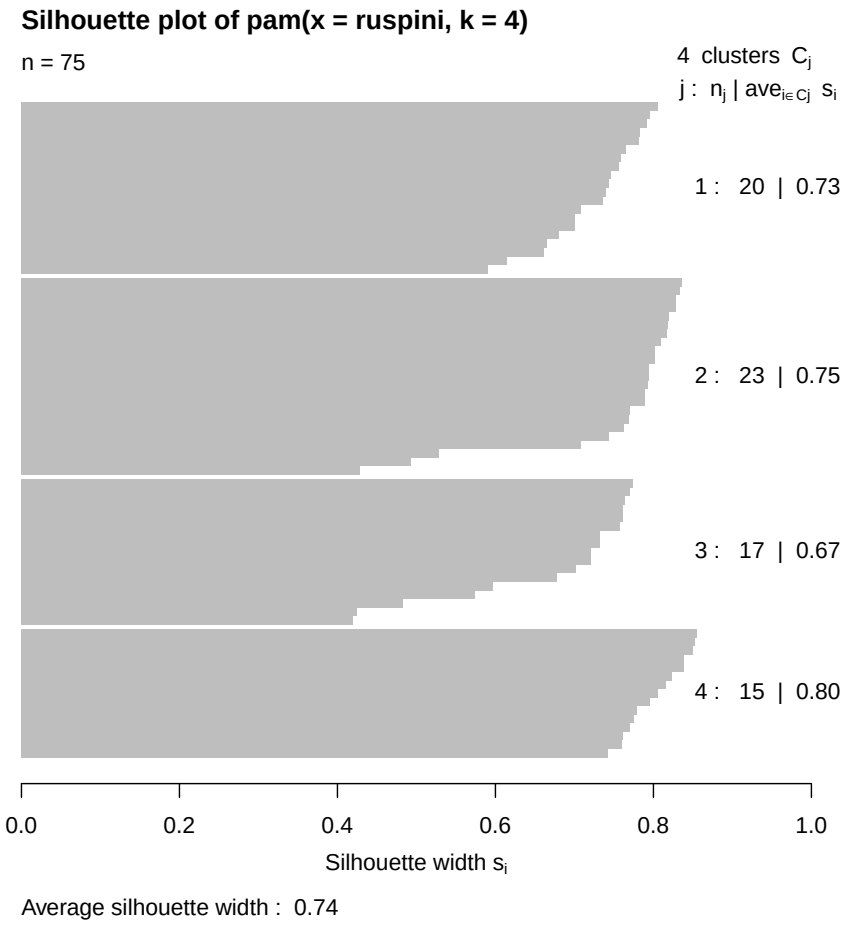
Nota de R 19 (ruspini) *Veamos el resumen de la silueta.*

[3] No parece un nombre inadecuado.

```
ruspini.pam <- pam(ruspini, 4)
summary(silhouette(ruspini.pam))

## Silhouette of 75 units in 4 clusters from pam(x = ruspini, k = 4) :
##  Cluster sizes and average silhouette widths:
##     20     23     17     15
## 0.7262 0.7548 0.6691 0.8042
## Individual silhouette widths:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.420   0.714   0.764   0.738   0.798   0.855
```

También podemos representarla gráficamente.

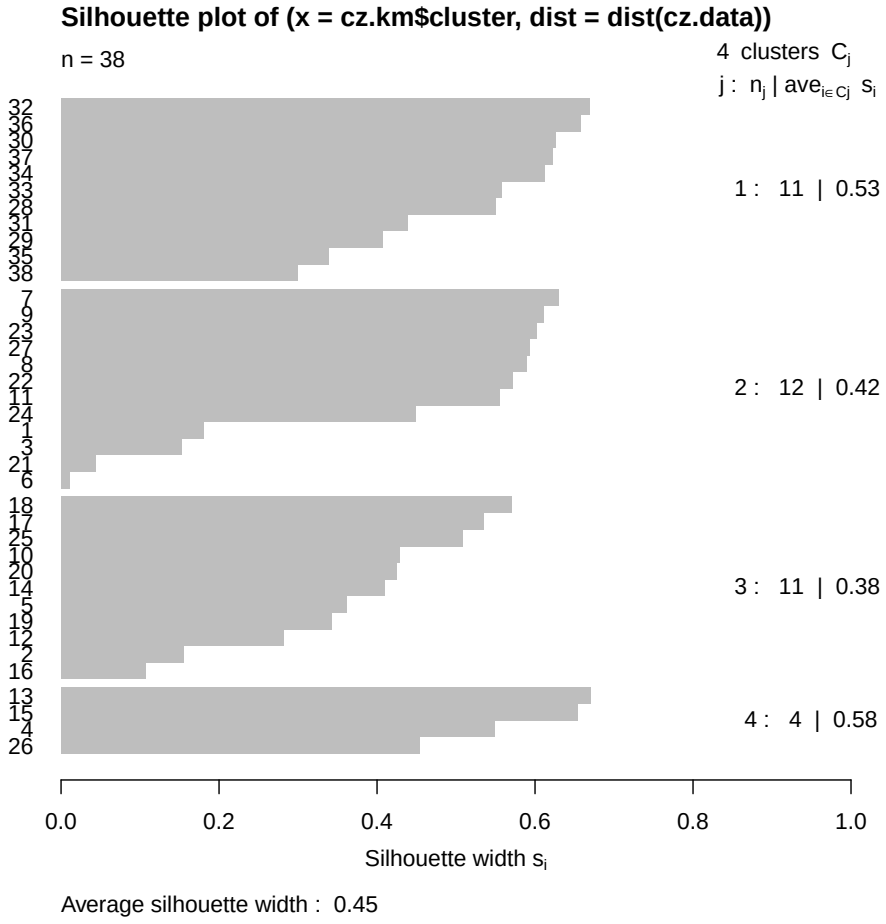
```
plot(silhouette(ruspini.pam))
```



Nota de R 20 (cz.data) *Para estos datos vamos a evaluar el procedimiento k-medias.*

```
cz.km <- kmeans(cz.data, 4)
summary(silhouette(cz.km$cluster, dist(cz.data)))

## Silhouette of 38 units in 4 clusters from silhouette.default(x = cz.km$cluster, dist = dis
##  Cluster sizes and average silhouette widths:
```

```
##     11     12     11      4
## 0.5254 0.4158 0.3751 0.5815
## Individual silhouette widths:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0113  0.3480  0.5220  0.4530  0.6000  0.6700

plot(silhouette(cz.km$cluster, dist(cz.data)))
```



4.6. Un ejemplo completo

Nota de R 21 (Un análisis completo de ALL) *En esta sección vamos a ver un análisis cluster completo.*

Lo tomamos de https://wiki.cgb.indiana.edu/display/r/4.+Bioconductor+Clustering+and+Visualization+Script. Como siempre empezamos cargando los datos.

```
library("ALL")
data("ALL")
```

Veamos los tipos de biología molecular y las frecuencias de cada tipo.

```
table(ALL$mol.biol)

##
## ALL1/AF4  BCR/ABL E2A/PBX1      NEG   NUP-98  p15/p16
##       10       37        5       74        1        1
```

En concreto vamos a quedarnos con dos tipos.

```
selSamples <- is.element(ALL$mol.biol, c("ALL1/AF4", "E2A/PBX1"))
```

Nos quedamos con las muestras (columnas) que corresponden a esa biología molecular.

```
ALLs <- ALL[, selSamples]
```

No vamos a trabajar con todos los genes. En lugar de ello vamos a filtrar atendiendo a criterios basados en sus niveles de expresión. Quizás podríamos decir que filtramos de acuerdo con criterios estadísticos. Obviamente otro filtraje de genes podría basarse en criterios biológicos atendiendo a algún conocimiento previo sobre los mismos.

En concreto los criterios estadísticos serán que el nivel medio de expresión sea mayor que un cierto valor mínimo tanto en un tipo de biología molecular como en el otro. Podemos utilizar la función apply *para calcular las medias. Notemos que la matriz que usamos son las expresiones de* ALLs `exprs(ALLs)` *y consideramos las columnas donde se verifica* `ALLs$mol.bio == "ALL1/AF4"`*, es decir, donde la biología molecular es la primera de las consideradas.*

```
m1 <- apply(exprs(ALLs)[, ALLs$mol.bio == "ALL1/AF4"], 1, mean)
```

Ahora podemos considerar qué filas son tales que esta media (calculada para cada gen) es mayor que el valor $\log_2(100)$*.*

```
s1 <- (m1 > log2(100))
```

¿Cuántos genes verifican este criterio?

```
table(s1)

## s1
## FALSE  TRUE
##  9186  3439
```

Vemos que hay 3439.
Hacemos lo mismo con el segundo tipo de biología molecular.

```
m2 <- apply(exprs(ALLs)[, ALLs$mol.bio == "E2A/PBX1"], 1, mean)
s2 <- (m2 > log2(100))
```

Podemos ver la tabla de los genes que verifican y no la condición.

```
table(s2)

## s2
## FALSE  TRUE
##  9118  3507
```

Podemos ver la tabla de contingencia en donde mostramos el número de genes que verifican las dos condiciones, solamente una o bien ninguna.

```
table(s1, s2)
```

```
##        s2
## s1      FALSE TRUE
##   FALSE  8863  323
##   TRUE    255 3184
```

A partir de ahora nos quedamos con los genes que verifican los dos criterios. Notemos que s1 es TRUE cuando se verifica la primera condición y s2 es TRUE cuando se verifica la segunda. Mediante el signo | indicamos la intersección, esto es, ha de darse la primera y la segunda.

```
ALLs <- ALLs[s1 | s2, ]
```

Podemos ver el número de filas y columnas de ALLs para comprobar que vamos bien.

```
dim(ALLs)
```

```
## Features  Samples
##     3762       15
```

Y vamos bien.

Vamos a considerar también que los niveles de expresión sean suficientemente variables*. En concreto la condición será que la desviación absoluta respecto de la mediana (o simplemente mad) supere un cierto valor. Primero hemos de calcular la desviación absoluta respecto de la mediana de cada fila.*

```
gen.mad <- apply(exprs(ALLs), 1, mad)
```

En concreto nos quedamos con los genes que tienen un valor de la desviación absoluta respecto de la mediana superior a 1.4.

```
ALLs <- ALLs[gen.mad > 1.4, ]
```

Otra vez, veamos qué datos tenemos.

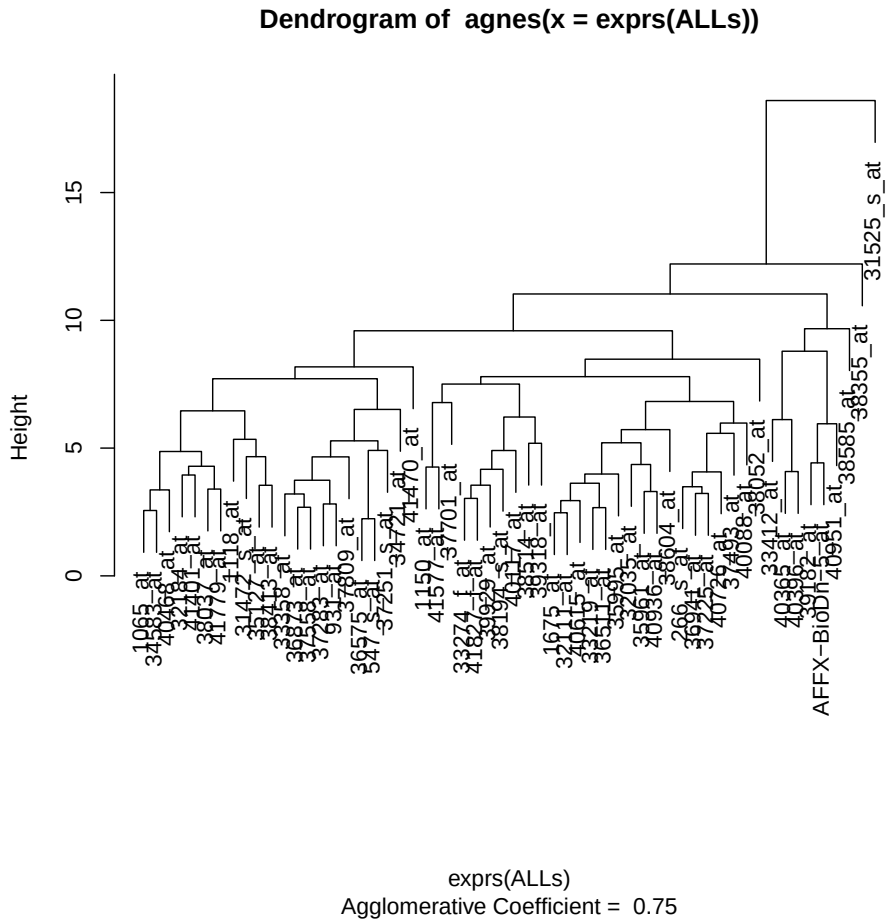
```
dim(ALLs)
```

```
## Features  Samples
##       58       15
```

Pocos. No tenemos muchos ahora. Mejor. Menos problemas para clasificar.

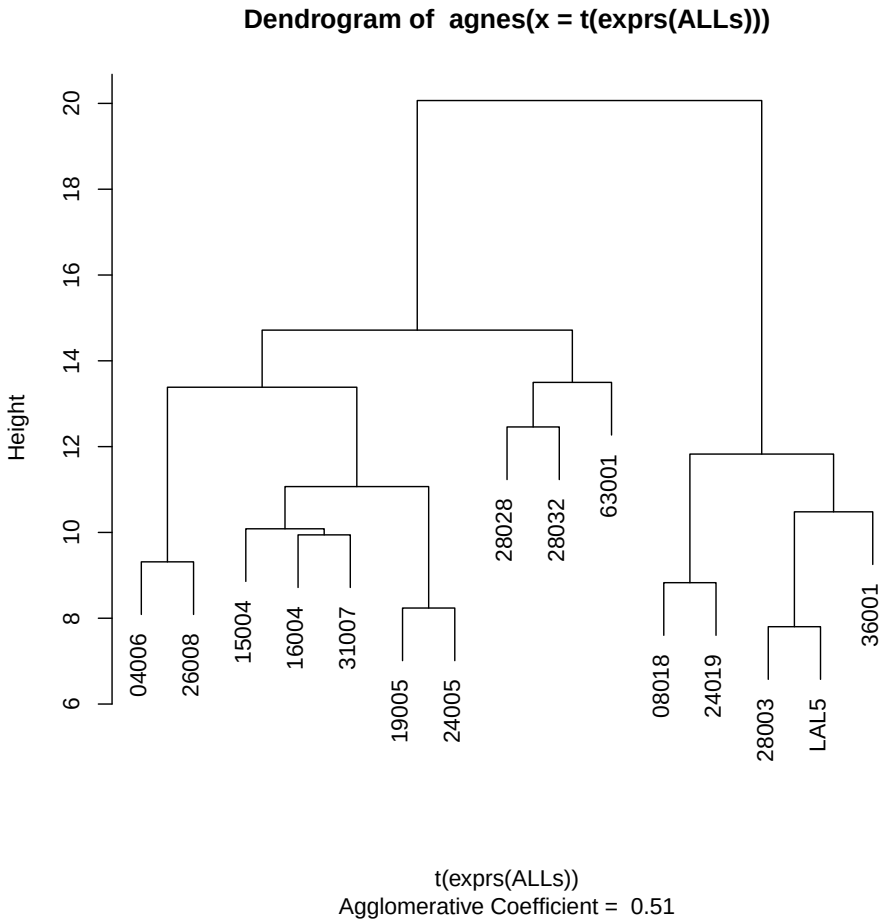
Vemos un cluster jerárquico aglomerativo de los genes.

```
genes.ag <- agnes(exprs(ALLs))
plot(genes.ag, which = 2)
```



O un análisis jerárquico de las muestras. Observemos que aplicamos la función agnes *a la matriz transpuesta de las expresiones* `t(exprs(ALLs))`.

```
muestras.ag <- agnes(t(exprs(ALLs)))
plot(muestras.ag, which = 2)
```


Dendrogram of agnes(x = t(exprs(ALLs)))
Height
6
8
10
12
14
16
18
20
04006
26008
15004
16004
31007
19005
24005
28028
28032
63001
08018
24019
28003
LAL5
36001
t(exprs(ALLs))
Agglomerative Coefficient = 0.51

Capítulo 5

Análisis discriminante o de cómo clasificar con muestra de entrenamiento

Discriminar es clasificar. Es una palabra mal vista. Cuando una persona le dice a otra: "Usted me discrimina" está indicando algo negativo en la medida en que piensa que lo está separando para perjudicarle. Se asume la discriminación como algo que se realiza en un sentido negativo. Pues no. El análisis discriminante simplemente se refiere al problema de clasificar en distintos grupos un conjunto de observaciones vectoriales. Clasificar y discriminar se usan como sinónimos.

Tenemos distintos conjuntos de datos multivariantes. ¿Qué quiere que tenemos grupos y qué pretendemos hacer con ellos? Una bióloga ha recogido una serie de animales y ha observado en cada uno unas características numéricas. Esta bióloga ha podido clasificar a los animales en distintas especies utilizando las variables x de las que dispone y, posiblemente, algunas otras cuya consecución ya no es tan simple y, en ocasiones, ni posible. Está interesada en diseñar un procedimiento que, partiendo de las variables de las que siempre dispone x, le permita decidir la especie a la que pertenece un animal futuro del cual solamente tiene las características x. Tiene esta persona un *problema de clasificación* que pretende hacer bien y clasificar a cada animal en su especie con un esfuerzo pequeño.

Otro ejemplo puede ser el siguiente. Una persona ha pasado una enfermedad (como la hidatidosis) y por lo tanto guarda en su organismo anticuerpos frente al virus que lo infectó y le causó la enfermedad. A este individuo se le controla a lo largo de los años. Cuando se le aplica un procedimiento diagnóstico el resultado puede ser positivo debido a dos razones: el individuo ha recaído en la enfermedad y por lo tanto ha de ser tratado. Sin embargo, si el procedimiento es muy sensible el resultado positivo del test puede ser simplemente producidos por los anticuerpos que la persona conserva. Diferenciar una situación de la otra supone otra exploración (placas del tórax) y, por ello, una complicación adicional. En la realización del test se han recogido la presencia o ausencia de una serie de aminoácidos. Tenemos que nuestro vector de características son variables binarias o dicotómicas y pretendemos poder decidir a partir de ellas si la persona está sana sin anticuerpos, sana con anticuerpos o finalmente enferma utilizando la presencia o ausencia de los distintos anticuerpos. Este es un segundo problema de interés a tratar aquí.

Pretendemos clasificar a un individuo dado utilizando algunas características del mismo. Pero para poder hacerlo tenemos que conocer para una muestra, que podemos llamar muestra de entrenamiento (training sample), en qué grupo está cada individuo con los que trabajamos. Esto hace bastante natural el nombre más

utilizado en el contexto de la Informática de *clasificación supervisada.*

Nota de R 22 (Cristales en la orina) *Tenemos pacientes de los cuales se conocen algunas variables obtenidas de un análisis de orina. En concreto las variables nos dan la gravidez específica (*grav*), la osmolaridad (*osmo*), la conductibilidad (*conduc*), la concentración de urea (*urea*) y la concentración de calcio (*calcio*). También tenemos una variable que nos indica la presencia o ausencia de cristales en la orina del individuo (*grupo *donde 1 indica ausencia y 2 indica presencia). El problema que nos planteamos es diseñar un procedimiento de clasificación de modo que, dado el vector de características de la orina, nos permita clasificar a un individuo en uno de los dos posibles grupos, esto es, que va a desarrollar cristales en la orina o no. En el diseño del procedimiento pretendemos utilizar la información que ya tenemos, esto es, conocemos para una serie de individuos si tiene o no cristales y los vectores de características asociados. En una observación futura tendremos el vector de características pero no conoceremos la clasificación. Leemos los datos de un fichero de SPSS utilizando el paquete* foreign **?** *y mostramos las seis primeras observaciones.*

```
library(foreign)
x <- read.spss(file = "../data/cristal.sav", to.data.frame = T)
x[x == -1] <- NA
cc <- complete.cases(x)
x <- x[cc, ]
```

Veamos algunos datos.

```
head(x)
```

```
##   IND                 GRUPO  GRAV   PH OSMO CONDUC UREA CALCIO
## 2   2 ausencia de cristales 1.017 5.74  577   20.0  296   4.49
## 3   3 ausencia de cristales 1.008 7.20  321   14.9  101   2.36
## 4   4 ausencia de cristales 1.011 5.51  408   12.6  224   2.15
## 5   5 ausencia de cristales 1.005 6.52  187    7.5   91   1.16
## 6   6 ausencia de cristales 1.020 5.27  668   25.3  252   3.34
## 7   7 ausencia de cristales 1.012 5.62  461   17.4  195   1.40
```

Incluimos un análisis descriptivo de los datos.

```
summary(x)
```

```
##       IND                             GRUPO         GRAV            PH            OSMO
##  Min.   : 2.0   ausencia de cristales :44   Min.   :1.00   Min.   :4.76   Min.   : 187
##  1st Qu.:21.0   presencia de cristales:33   1st Qu.:1.01   1st Qu.:5.53   1st Qu.: 410
##  Median :40.0                              Median :1.02   Median :5.94   Median : 594
##  Mean   :40.3                              Mean   :1.02   Mean   :6.04   Mean   : 614
##  3rd Qu.:60.0                              3rd Qu.:1.02   3rd Qu.:6.40   3rd Qu.: 803
##  Max.   :79.0                              Max.   :1.04   Max.   :7.94   Max.   :1236
##      CONDUC          UREA         CALCIO
##  Min.   : 5.1   Min.   : 10   Min.   : 0.17
##  1st Qu.:14.3   1st Qu.:159   1st Qu.: 1.45
##  Median :21.4   Median :255   Median : 3.16
##  Mean   :20.9   Mean   :262   Mean   : 4.16
##  3rd Qu.:27.0   3rd Qu.:362   3rd Qu.: 6.19
##  Max.   :38.0   Max.   :620   Max.   :14.34
```

Nota de R 23 (Diabetes) *Los datos corresponden a una serie de personas de las cuales conocemos información que previsiblemente nos permitirá predecir si son diabéticos o no. Incluso dentro de los diabéticos pretendemos discriminar (distinguir) entre diabetes clínica y diabetes manifiesta. La variable* tipo *nos indica en qué grupo está la persona observada de los tres grupos indicados. El resto de variables nos describen al paciente:* peso *es el peso relativo,* gpb *es la glucosa plasmática en ayunas,* garea *el área bajo la curva de la glucosa,* iarea *el área bajo la curva de insulina y* sspg *la glucosa plasmática en estado estacionario. Pretendemos clasificar a un individuo en uno de los tres grupos posibles teniendo en cuenta las variables consideradas.*

```
library(foreign)
x <- read.spss(file = "../data/diabetes.sav", to.data.frame = T)
head(x)
```

```
##   IND PESO GPB GAREA IAREA SSPG    TIPO LIAREA
## 1   1 0.81  80   356   124   55 control  4.820
## 2   2 0.95  97   289   117   76 control  4.762
## 3   3 0.94 105   319   143  105 control  4.963
## 4   4 1.04  90   356   199  108 control  5.293
## 5   5 1.00  90   323   240  143 control  5.481
## 6   6 0.76  86   381   157  165 control  5.056
```

Veamos un breve análisis descriptivo de los datos.

```
summary(x)
```

```
##       IND             PESO             GPB           GAREA          IAREA           SSPG
##  Min.   :  1   Min.   :0.710   Min.   : 70   Min.   : 269   Min.   : 10   Min.   : 29
##  1st Qu.: 37   1st Qu.:0.880   1st Qu.: 90   1st Qu.: 352   1st Qu.:118   1st Qu.:100
##  Median : 73   Median :0.980   Median : 97   Median : 413   Median :156   Median :159
##  Mean   : 73   Mean   :0.977   Mean   :122   Mean   : 544   Mean   :186   Mean   :184
##  3rd Qu.:109   3rd Qu.:1.080   3rd Qu.:112   3rd Qu.: 558   3rd Qu.:221   3rd Qu.:257
##  Max.   :145   Max.   :1.200   Max.   :353   Max.   :1568   Max.   :748   Max.   :480
##                    TIPO        LIAREA
##  diabetes manifiesta:33   Min.   :2.30
##  diabetes quimica   :36   1st Qu.:4.77
##  control            :76   Median :5.05
##                           Mean   :5.02
##                           3rd Qu.:5.40
##                           Max.   :6.62
```

El capítulo está organizado del siguiente modo. Empezamos (sección 5.1) recordando el teorema de Bayes con un ejemplo muy simple de urnas (no funerarias). De este modo vemos la idea básica del método de clasificación basado en probabilidades a posteriori. Consideramos, en la sección 5.2, el caso (de interés puramente académico) de dos poblaciones normales univariantes con la misma varianza y con los parámetros conocidos [1]. En la sección 5.3 abordamos la situación con dos poblaciones normales multivariantes. Allí consideramos tanto el caso en que las matrices de covarianzas son la misma como cuando son distintas. En la sección 5.4 nos planteamos la estimación de los vectores de medias y las matrices de covarianzas y vemos la implementación práctica del método. El problema de la reducción de la dimensión dentro del problema de la clasificación es considerado en la sección 5.7

[1] En datos reales los parámetros no son conocidos.

5.1. Un problema de probabilidad sencillo

Veamos un problema de probabilidad básico que nos servirá para introducir el procedimiento de clasificación que vamos a utilizar. No le falta ningún detalle y muchos lo hemos resuelto. Tenemos dos urnas. En la primera de ellas hay una bola blanca y dos negras mientras que en la segunda urna hay dos bolas blancas y una negra. Elegimos al azar una urna (no sabemos cuál es la elegida). Posteriormente de la urna elegida, elegimos a su vez una bola. Resulta que la bola elegida es blanca. La pregunta que nos hacemos es: ¿De qué urna la hemos elegido? La solución es una aplicación del teorema de Bayes (ver 1.2.3). Denotamos B_i el suceso consistente en que la bola ha sido extraída de la i-ésima urna y por el A el suceso de que la bola es blanca. A priori, antes de realizar el experimento, las dos urnas tenían la misma probabilidad (*elegimos al azar una de las urnas*) y por tanto la probabilidad (previa o a priori) de los sucesos B_i serían $P(B_i) = 1/2$. No sabemos si la urna elegida ha sido la primera o la segunda pero nos podemos plantear qué probabilidad tenemos de que sea blanca si efectivamente es la urna 1 la elegida y lo mismo para la dos. Es obvio que $P(A \mid B_1) = 1/3$ y $P(A \mid B_2) = 2/3$. Esta información se puede combinar aplicando el teorema de Bayes para determinar la probabilidad de que sea la primera o la segunda urna la elegida *sabiendo* (teniendo pues una información adicional sobre el experimento) que ha salido blanca. En concreto tenemos que $P(B_i \mid A) = \frac{P(A|B_i)P(B_i)}{\sum_{j=1}^{k} P(A|B_j)P(B_j)}$. Finalmente podemos comprobar que $P(B_1 \mid A) = 1/3$ y $P(B_2 \mid A) = 2/3$.

Las probabilidades $P(B_1)$ y $P(B_2)$ reciben el nombre de **probabilidades a priori**. Vamos a denotarlas en lo que sigue por π_i, esto es, la probabilidad de la urna i. Nuestra información consiste en la ocurrencia del suceso A (la bola ha sido blanca) de modo que las probabilidades $P(A \mid B_1)$ y $P(A \mid B_2)$ serían las **verosimilitudes** de que ocurra lo que ha ocurrido si la urna es la primera o la segunda. Finalmente tenemos $P(B_1 \mid A)$ y $P(B_2 \mid A)$ que nos darían las **probabilidades a posteriori**.

Hemos de tomar una decisión: ¿cuál fue la urna elegida? Parece natural elegir aquella que tiene a posteriori una máxima probabilidad y quedarnos con la segunda urna.

Vamos a reescribir lo que acabamos de hacer que nos acerque al planteamiento más genérico del problema. Supongamos que describimos el color de la bola elegida mediante una variable dicotómica o binaria. Consideramos la variable aleatoria X que vale uno si es blanca la bola y cero en otro caso. Es lo que se llama una variable indicatriz pues nos indica si se ha producido el suceso que nos interesa. [2] El comportamiento aleatorio de X, su distribución de probabilidad, depende que estemos extrayendo una bola de la primera o de la segunda urna. En concreto en la primera urna X sigue una distribución Bernoulli con probabilidad de éxito $p_1 = 1/3$ mientras que en la segunda urna tenemos una Bernoulli con probabilidad de éxito $p_2 = 2/3$. Cada urna es una población distinta donde el comportamiento aleatorio de la misma cantidad es distinto. X en la i-ésima población tiene una función de probabilidad

$$f_i(x) = p_i^x (1 - p_i)^{1-x} \text{ con } x = 0, 1.$$

Teníamos unas probabilidades a priori de que X estuviera siendo observada en la población i-ésima que denotamos por $\pi(i)$ donde $\pi(1) + \pi(2) = 1$ y $\pi(1), \pi(2) \geq 0$. Las probabilidades a posteriori obtenidas por la aplicación del teorema de Bayes vienen dadas por

$$\pi(i \mid x) = \frac{f_i(x)\pi(i)}{f_1(x)\pi(1) + f_2(x)\pi(2)}.$$

[2] Si A es el suceso de interés entonces $X(\omega) = 1$ si $\omega \in A$ y cero en otro caso. A veces se denota como $X(\omega) = 1_A(\omega)$.

Finalmente nos hemos quedado con la población i tal que tenía un valor de $\pi(i \mid x)$ mayor, aquella que, una vez observado el valor de $X = x$, hacía más probable la población.

5.2. Dos poblaciones normales

Supongamos ahora que tenemos que decidir entre dos poblaciones basándonos en un valor aleatorio continuo con distribución normal. En concreto supondremos que en la primera población X es normal con media μ_1 y varianza σ^2. En la segunda población X tiene distribución normal con media μ_2 y varianza σ^2. Gráficamente en la figura **??** aparece la situación con la que nos encontramos. Supongamos conocidos los valores de las media y la varianza común. Observamos un valor de la variable $X = x$: ¿cuál de las dos distribuciones lo ha generado? De otro modo: ¿a qué población pertenece este valor generado?

La idea para clasificar este valor generado es la misma de antes. Ahora tendremos

$$f_i(x) = f(x \mid \mu_i, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_i)^2 \right),$$

aunque $f_i(x)$ no es la probabilidad del valor x asumiendo que estamos en la población i. No obstante, $f_i(x)dx$ sí que tiene este sentido. Hablando en un sentido amplio tenemos una interpretación similar. La observación x la clasificaríamos en la población 1 si

$$\frac{\pi(1) f(x \mid \mu_1, \sigma^2)}{\pi(2) f(x \mid \mu_2, \sigma^2)} > 1,$$

Fácilmente comprobamos que esto equivale con que

$$\frac{1}{\sigma^2}(\mu_1 - \mu_2)\left(x - \frac{1}{2}(\mu_1 + \mu_2) \right) > \log \frac{\pi(2)}{\pi(1)}.$$

5.3. Dos normales multivariantes

En la sección 5.2 nos planteábamos la situación de dos normales univariantes. En las aplicaciones es más habitual el caso en que trabajamos con varias características simultáneamente. En particular, vamos a asumir ahora que X puede pertenecer a una de dos poblaciones normales multivariantes. La primera con vector de medias μ_1 y matriz de covarianzas Σ y la segunda con vector de medias μ_2 y matriz de covarianzas Σ. Dada una observación multivariante x, la clasificaremos en la población 1 si

$$\frac{\pi(1) f(x \mid \mu_1, \Sigma)}{\pi(2) f(x \mid \mu_2, \Sigma)} > 1,$$

pero,

$$\frac{f(x \mid \mu_1, \Sigma)}{f(x \mid \mu_2, \Sigma)} =$$

$$\exp\left[-\frac{1}{2}(x - \mu_1)' \Sigma^{-1} (x - \mu_1) + \frac{1}{2}(x - \mu_2)' \Sigma^{-1} (x - \mu_2) \right] =$$

$$\exp\left[(\mu_1 - \mu_2)' \Sigma^{-1} x - \frac{1}{2}(\mu_1 - \mu_2)' \Sigma^{-1} (\mu_1 + \mu_2) \right]. \quad (5.1)$$

Sea $\lambda = \Sigma^{-1}(\mu_1 - \mu_2)$, entonces la observación es asignada a la primera población si

$$D(x) = \lambda' \left[x - \frac{1}{2}(\mu_1 + \mu_2) \right] > \log \frac{\pi(2)}{\pi(1)}. \quad (5.2)$$

Notemos que la ecuación $D(x) = \log \frac{\pi(2)}{\pi(1)}$ nos define un hiperplano que separa las dos poblaciones.

¿Qué ocurre si no asumimos que tenemos una misma matriz de covarianzas? En este caso se tiene que:

$$Q(x) = \log \frac{f(x \mid \mu_1, \Sigma_1)}{f(x \mid \mu_2, \Sigma_2)} = \\ \frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2}(x-\mu_1)'\Sigma_1^{-1}(x-\mu_1) + \frac{1}{2}(x-\mu_2)'\Sigma_2^{-1}(x-\mu_2) = \\ \frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2}\Big[x'(\Sigma_1^{-1} - \Sigma_2^{-1})x - 2x'(\Sigma_1^{-1}\mu_1 - \Sigma_2^{-1}\mu_2)\Big]. \quad (5.3)$$

Como en el caso anterior asignamos la observación x a la primera población si

$$Q(x) > \log \frac{\pi(2)}{\pi(1)}.$$

Notemos que ahora $Q(x) = \log \frac{\pi(2)}{\pi(1)}$ no es un hiperplano sino que tenemos una superficie no plana.

5.4. Dos poblaciones normales multivariantes con parámetros desconocidos

Lo visto en las secciones anteriores tenía como objeto mostrar de un modo suave la transición desde el resultado probabilístico básico, el teorema de Bayes, y su aplicación en el problema de la clasificación. Sin embargo, no es real asumir que conocemos completamente la distribución de las observaciones en cada población o clase. En las aplicaciones los vectores de medias y la matriz o matrices de covarianzas no son conocidas. Hemos de estimarlas a partir de los datos. Veamos primero cómo hacerlo y luego cómo usar estos parámetros en el procedimiento de clasificación.

Empezamos por el caso en que tenemos dos poblaciones normales con vectores de medias μ_1 y μ_2 y matrices de covarianzas Σ_1 y Σ_2. Lo que tenemos son dos muestras aleatorias correspondientes a cada una de las poblaciones.

Supongamos que tenemos n_i individuos de la población i y los vectores de características son los vectores columna $x_{ij} \in \mathbb{R}^d$ (con $i = 1, 2$ y $j = 1, \ldots, n_i$). Denotamos

$$\bar{x}_{i\cdot} = \frac{\sum_{j=1}^{n_i} x_{ij}}{n_i}, \quad \bar{x}_{\cdot\cdot} = \frac{\sum_{i=1}^{2}\sum_{j=1}^{n_i} x_{ij}}{n} \quad (5.4)$$

donde $n = n_1 + n_2$. Sea S_i la matriz de varianzas o de dispersión de la población i, es decir,

$$S_i = \frac{\sum_{j=1}^{n_i}(x_{ij} - \bar{x}_{i\cdot})(x_{ij} - \bar{x}_{i\cdot})'}{n_i - 1}. \quad (5.5)$$

El vector μ_i es estimado mediante $\hat{\mu}_i = \bar{x}_{i\cdot}$. La matriz Σ_i la estimamos mediante S_i. En el caso particular en que asumamos que $\Sigma = \Sigma_1 = \Sigma_2$ entonces la matriz de covarianzas común la estimamos con

$$\mathbf{S}_p = \frac{\sum_{i=1}^{2}(n_i - 1)S_i}{n - 2}.$$

¿Cómo clasificamos? Las distribuciones teóricas que suponíamos conocidas son reemplazadas por las distribuciones normales con los parámetros estimados.

Si asumimos una matriz de covarianza común a ambas poblaciones entonces asignamos x a la primera población si

$$D_s(x) > \log \frac{\pi(2)}{\pi(1)}, \tag{5.6}$$

donde

$$D_s(x) = \hat{\lambda}'(x - \frac{1}{2}(\bar{x}_{1\cdot} + \bar{x}_{2\cdot}) \tag{5.7}$$

y

$$\hat{\lambda} = \mathbf{S}_p^{-1}(\bar{x}_{1\cdot} - \bar{x}_{2\cdot}). \tag{5.8}$$

La función D_s recibe el nombre de *función discriminante lineal*. La razón es obvia: clasificamos en uno o en otro grupo utilizando una función lineal de las distintas variables.

En el caso particular en que $\pi(1) = \pi(2)$, esto es, consideramos a priori igualmente probables ambos grupos entonces la regla de clasificación propuesta sería: clasificamos en la población o clase 1 si,

$$\hat{\lambda}'x > \frac{1}{2}(\hat{\lambda}'\bar{x}_{1\cdot} + \hat{\lambda}'\bar{x}_{2\cdot}).$$

Es es el procedimiento que propuso R.A. Fisher en 1936.

Notemos que las probabilidades de pertenencia a posteriori a cada una de las poblaciones pueden ser estimadas mediante

$$\hat{\pi}(i|x) = \frac{\pi(i)f(x|\bar{x}_{i\cdot}, \mathbf{S}_p)}{\pi(1)f(x|\bar{x}_{1\cdot}, \mathbf{S}_p) + \pi(2)f(x|\bar{x}_{2\cdot}, \mathbf{S}_p)}. \tag{5.9}$$

Una vez tenemos las probabilidades a posteriori estimadas el individuo es clasificado en el grupo que tiene una mayor probabilidad a posteriori.

En la situación más general no asumiremos una misma matriz de covarianzas en las dos poblaciones. En este caso estimamos la matriz Σ_i mediante la matriz S_i dada en la ecuación 5.5. Las probabilidades a posteriori las estimamos como

$$\hat{\pi}(i|x) = \frac{\pi(i)f(x|\bar{x}_{i\cdot}, S_i)}{\pi(1)f(x|\bar{x}_{1\cdot}, S_1) + \pi(2)f(x|\bar{x}_{2\cdot}, S_2)}. \tag{5.10}$$

Nota de R 24 *Vamos a trabajar con los datos de cristales en la orina. Esta nota es un ejemplo de análisis discriminante lineal con dos grupos. Consideramos dos casos. En el primero las probabilidades a priori de cada grupo se asumen iguales entre sí y, por lo tanto, iguales a* 0,5*. En el segundo caso, las probabilidades a priori coinciden con las proporciones observadas dentro de la muestra de cada una de las poblaciones o clases. Leemos los datos.*

```
library(foreign)
x <- read.spss(file = "../data/cristal.sav", to.data.frame = T)
```

Definimos el valor -1 como dato faltante.

```
x[x == -1] <- NA
```

Eliminamos del estudio todos los casos en los que hay algún dato faltante.

```
cc <- complete.cases(x)
attach(x[cc, ])
```

Suponemos matrices de covarianzas iguales y probabilidades a priori iguales.

```
library(MASS)
z <- lda(GRUPO ~ CALCIO + CONDUC + GRAV + OSMO + PH + UREA, prior = c(1, 1)/2)
```

Veamos cómo recuperar los distintos elementos del análisis.

```
attributes(z)

## $names
##  [1] "prior"   "counts"  "means"   "scaling" "lev"     "svd"     "N"       "call"
##  [9] "terms"   "xlevels"
##
## $class
## [1] "lda"
```

Las probabilidades a priori vienen dadas por

```
z$prior

##  ausencia de cristales presencia de cristales
##                    0.5                    0.5
```

El número de datos por grupo es

```
z$counts

##  ausencia de cristales presencia de cristales
##                     44                     33
```

El vector de medias estimado en cada grupo lo obtenemos con

```
z$means

##                         CALCIO CONDUC  GRAV  OSMO    PH  UREA
## ausencia de cristales    2.629  20.55 1.015 561.7 6.126 232.4
## presencia de cristales   6.202  21.38 1.022 682.9 5.927 302.4
```

Vamos a obtener las probabilidades a posteriori. Observar la opción CV=TRUE.

```
z <- lda(GRUPO ~ CALCIO + CONDUC + GRAV + OSMO + PH + UREA, prior = c(1, 1)/2, CV = TRUE)
attributes(z)

## $names
## [1] "class"     "posterior" "terms"     "call"      "xlevels"
```

Obtengamos las probabilidades a posteriori.

```
head(z$posterior)

##   ausencia de cristales presencia de cristales
## 1                0.6411                0.35893
## 2                0.8700                0.12997
## 3                0.8484                0.15162
## 4                0.9053                0.09466
## 5                0.5762                0.42381
## 6                0.8968                0.10321
```

y las clasificaciones para cada los distintos casos.

```
head(z$class)

## [1] ausencia de cristales ausencia de cristales ausencia de cristales
## [4] ausencia de cristales ausencia de cristales ausencia de cristales
## Levels: ausencia de cristales presencia de cristales
```

Las probabilidades a priori corresponden con proporciones observadas.

```
z1 <- lda(GRUPO ~ CALCIO + CONDUC + GRAV + OSMO + PH + UREA, CV = TRUE)
```

Las probabilidades a posteriori son ahora

```
head(z1$posterior)

##   ausencia de cristales presencia de cristales
## 1                0.7043                0.29573
## 2                0.8992                0.10075
## 3                0.8818                0.11820
## 4                0.9273                0.07272
## 5                0.6445                0.35553
## 6                0.9205                0.07946
```

5.5. Análisis discriminante con más de dos poblaciones normales

Supongamos que tenemos unas probabilidades a priori $\pi(i)$ de que el caso pertenezca al grupo i con $i = 1, \ldots, g$ (obviamente $\sum_{i=1}^{g} \pi(i) = 1$). Si x son las características de un caso entonces vamos a asumir que x tiene una distribución normal multivariante con media μ_i y matriz de varianzas Σ_i en la clase i. Su densidad de probabilidad viene dada por

$$f(x \mid \mu_i, \Sigma_i) = (2\pi)^{-d/2} \mid \Sigma_i \mid^{-1/2} \exp\{-\frac{1}{2}(x - \mu_i)'\Sigma_i^{-1}(x - \mu_i)\}. \tag{5.11}$$

Utilizando el teorema de Bayes tenemos que las probabilidades a posteriori vienen dadas por

$$\pi(i \mid x) = \frac{\pi(i) f(x \mid \mu_i, \Sigma_i)}{\sum_{j=1}^{g} \pi(j) f(x \mid \mu_j, \Sigma_j)}. \tag{5.12}$$

Un método de clasificación consiste en clasificar al individuo con características x en la clase o grupo tal que la probabilidad a posteriori $\pi(i \mid x)$ es máxima o lo que es lo mismo: clasificamos en el grupo tal que

$$\pi(i) f(x|\mu_i, \Sigma_i) = \max_j \pi_j f(x|\mu_j, \Sigma_j).$$

Obviamente, habitualmente no conocemos los parámetros (μ_i, Σ_i) de las distintas clases por lo que hemos de estimarlos.

Supongamos que tenemos n_i individuos en la clase i y los vectores de características son los vectores columna $x_{ij} \in \mathbb{R}^d$ (con $i = 1, \ldots, g$ y $j = 1, \ldots, n_i$). Denotamos

$$\bar{x}_{i\cdot} = \frac{\sum_{j=1}^{n_i} x_{ij}}{n_i}, \quad \bar{x}_{\cdot\cdot} = \frac{\sum_{i=1}^{g} \sum_{j=1}^{n_i} x_{ij}}{n} \tag{5.13}$$

donde $n = \sum_{i=1}^{g} n_i$. Sea S_i la matriz de varianzas o de dispersión de la clase i, es decir,

$$S_i = \frac{\sum_{j=1}^{n_i}(x_{ij} - \bar{x}_{i\cdot})(x_{ij} - \bar{x}_{i\cdot})'}{n_i - 1}. \tag{5.14}$$

El vector μ_i es estimado mediante $\hat{\mu}_i = \bar{x}_{i\cdot}$. En cuanto a la estimación de las matrices Σ_i se utilizan dos estimadores. En el caso en que asumamos que todas son iguales entonces el estimador de $\Sigma = \Sigma_1 = \ldots = \Sigma_g$ es $\hat{\Sigma} = \mathbf{S}_p$ donde

$$\mathbf{S}_p = \frac{\sum_{i=1}^{g}(n_i - 1)S_i}{n - g}.$$

Si no asumimos que las distintas matrices de varianzas son iguales entonces cada Σ_i es estimada mediante S_i.

Es claro que el procedimiento indicado en la ecuación 5.5 no es aplicable pues no conocemos los parámetros. Veamos como queda el procedimiento en las dos situaciones posibles: asumiendo igualdad de las matrices de covarianzas y asumiendo que son distintas.

Bajo la hipótesis de matriz de covarianza común tendremos que

$$\log[\pi(i)f(x|\bar{x}_{i\cdot}, \mathbf{S}_p)] = \log \pi(i) + c - \frac{1}{2}(x - \bar{x}_{i\cdot})'\mathbf{S}_p^{-1}(x - \bar{x}_{i\cdot}).$$

Le quitamos a $\log[\pi(i)f(x|\bar{x}_{i\cdot}, \mathbf{S}_p)]$ la parte que no depende de i dada por $c - \frac{1}{2}x'\mathbf{S}_p^{-1}x$ y obtenemos la función

$$L_i(x) = \log \pi(i) + \bar{x}_{i\cdot}'\mathbf{S}_p^{-1}(x - \frac{1}{2}\bar{x}_{i\cdot}).$$

Asignamos x al grupo que tiene un valor mayor de la función $L_i(x)$. Estas funciones reciben el nombre de *funciones discriminantes*. Observemos que las diferencias entre distintas funciones L_i son hiperplanos y por ello se habla de *análisis discriminante lineal*.

En el caso en que **no se asume una matriz de varianzas común** entonces la regla de clasificación consiste en clasificar donde es máxima la siguiente función

$$Q_i(x) = 2\log \pi(i) - \log |S_i| - (x - \bar{x}_{i\cdot})'S_i^{-1}(x - \bar{x}_{i\cdot}). \tag{5.15}$$

Notemos que el último término no es más que la distancia de Mahalanobis de x al centro estimado de la clase, $\bar{x}_{i\cdot}$. La diferencia entre las funciones Q_i para dos clases distintas es una función cuadrática y por ello el método recibe el nombre de *análisis discriminante cuadrático*.

5.6. Valoración del procedimiento de clasificación

Nuestros datos son (x_i, y_i) con $i = 1, \ldots, n$ siendo x_i el vector de característica del individuo y denotando y_i la clase población o grupo al que pertenece realmente el individuo. Tenemos unas probabilidades a posteriori $\hat{\pi}(j|x)$ para cada x que nos queramos plantearla. Clasificamos x en la clase j tal que tiene máxima probabilidad de las posibles. Pero el método así construido, ¿va bien o es un desastre? Parece que todos coincidimos en que ir bien quiere decir clasificar a los individuos en los grupos a los que realmente pertenecen.

Una primera práctica que pretende valorar la probabilidad (y por lo tanto la frecuencia de veces que ocurre) de una clasificación correcta es, una vez estimadas las probabilidades a posteriori para los propios elementos de la muestra. Esto es, nos planteamos clasificar a los individuos utilizados para construir el procedimiento

de clasificación. Tendremos para cada data, y_i, el grupo al que pertenece e y_i^* el grupo en el que lo clasificamos. Podemos considerar una valoración del resultado de la clasificación la siguiente cantidad,

$$I = \sum_{i=1}^{n} \frac{\delta_{y_i, y_i^*}}{n}, \tag{5.16}$$

donde $\delta_{y,y^*} = 1$ si $y = y^*$ y cero en otro caso. La cantidad definida en 5.16 es de uso habitual en la literatura de reconocimiento de patrones. Es, sin duda, un modo razonable de valorar la calidad del procedimiento de clasificación. Se queda pobre. Al menos parece insuficiente. ¿Cómo nos equivocamos cuando clasificamos? La siguiente opción habitual es utilizar la **tabla de clasificación** en donde cruzamos los valores (y_i, y_i^*). En esta tabla tendremos en la fila (r, c) el número de casos que originalmente son de la clase r y los hemos clasificado en el grupo c. Valorando la tabla de clasificación podemos valorar el método de clasificación. Es importante tener en cuenta aquí que no todos los errores de clasificación tienen la misma importancia.

Independientemente de la valoración numérica que hagamos del procedimiento de clasificación hemos de tener en cuenta sobre qué casos estamos realizando esta valoración. Si un mismo caso lo utilizamos para construir el procedimiento de valoración y lo volvemos a utilizar para clasificarlo estamos sesgando la valoración del procedimiento a favor del mismo. Un procedimiento de clasificación siempre irá mejor con los casos que utilizamos para construirlo y peor sobre otros casos. Hemos de intentar corregirlo.

Una primera idea es *dejar uno fuera cada vez*. Para cada j consideramos toda la muestra menos x_j. Utilizando el resto de la muestra estimamos los vectores de medias y las matrices de covarianzas y, finalmente, las probabilidades a posteriori del individuo j en cada clase. Lo clasificamos del modo habitual. Repetimos el procedimiento para cada j y construimos la tabla correspondiente. En inglés es la técnica conocida como *leaving one out*. Realmente el método de clasificación que valoramos en cada ocasión no es exactamente el mismo pero no es muy diferente. En cada ocasión solamente estamos prescindiendo de un caso y los vectores de medias y matrices de covarianzas no se modifican mucho. Estamos valorando esencialmente el mismo método.

Una segunda opción; mejor en su validez peor en sus necesidades. Si tenemos una muestra de tamaño n elegimos una muestra sin reemplazamiento de tamaño m. Los m datos seleccionados son utilizadas para estimar las probabilidades a posteriori y los $n - m$ restantes son clasificados y podemos valorar las distintas proporciones de error. Es una estimación del error basada en un *método de aleatorización*. Si elegimos las sucesivas muestras con reemplazamiento tendríamos un método bootstrap.

Nota de R 25 (Los datos iris de Fisher) *Consideremos los datos iris tratados originalmente por Fisher. Vemos cómo se utiliza una muestra para estimar las matrices de covarianzas y los vectores de medias mientras que clasificamos a los individuos no utilizados en la estimación de los parámetros. Se utilizan los datos iris de Fisher.*

```
library(MASS)
data(iris3)
Iris <- data.frame(rbind(iris3[, , 1], iris3[, , 2], iris3[, , 3]), Sp = rep(c("s", "c", "v"),
    rep(50, 3)))
```

Tomamos una muestra y con esta muestra estimamos los vectores de medias y la matriz de covarianzas.

```
train <- sample(1:150, 75)
table(Iris$Sp[train])
z <- lda(Sp ~ ., Iris, prior = c(1, 1, 1)/3, subset = train)
```

Con los estimadores podemos ahora clasificar los demás datos de la muestra.

```
predict(z, Iris[-train, ])$class

##  [1] s s s s s s s s s s s s s s s s s s s s s s s s s c c c c c c c c c c c c c c c v c
## [44] c c c c c c v v v v v v v v v v v v v v v v v v v c v v v v v v v v v
## Levels: c s v
```

Nota de R 26 *Seguimos con los datos de la orina. Notemos que consideramos probabilidades a priori correspondientes a las proporciones en la muestra. Nos limitamos a construir la tabla de clasificación. Suponemos matrices de covarianzas iguales y probabilidades a priori dadas por las proporciones de cada clase en la muestra.*

```
library(foreign)
x <- read.spss(file = "../data/cristal.sav", to.data.frame = T)
x[x == -1] <- NA
cc <- complete.cases(x)
x <- x[cc, ]
attach(x)
```

Realizamos un análisis discriminante lineal.

```
z1 <- lda(GRUPO ~ CALCIO + CONDUC + GRAV + OSMO + PH + UREA, CV = TRUE)
```

Construimos la tabla de clasificaciones.

```
table(GRUPO, z1$class)

##
## GRUPO                    ausencia de cristales presencia de cristales
##   ausencia de cristales                     42                      2
##   presencia de cristales                    14                     19
```

Nota de R 27 (Correo basura) *Consideramos unos datos de correo electrónico. El objetivo del análisis es decidir si el correo es basura o no basándonos en información de dicho correo. Estos datos se pueden encontrar en D.J. Newman and Merz [1998]. Los datos los podemos encontrar en* ***http://mlearn.ics.uci.edu/databases/spambase/*** *y en particular la descripción de estos datos la tenemos en* ***http://mlearn.ics.uci.edu/databases/spambase/spambase.DOCUMENTATION****.*

Realizamos un análisis discriminante linear y un análisis cuadrático. Vemos que el lineal nos proporciona mejores resultados.

```
library(MASS)
x <- read.table(file = "../data/spambase_data", sep = ",")
attach(x)
xnam <- paste("V", 1:57, sep = "")
(fmla <- as.formula(paste("y ~ ", paste(xnam, collapse = "+"))))
y <- x[, 58]
```

Realizamos el análisis discriminante lineal.

```
z <- lda(fmla, data = x, prior = c(1, 1)/2, CV = T)
```

La tabla de clasificación es la siguiente.

```
table(V58, z$class)

##
## V58    0    1
##   0 2625  163
##   1  265 1548
```

Realizamos el análisis discriminante cuadrático y obtenemos la nueva tabla de clasificación.

```
z <- qda(fmla, data = x, prior = c(1, 1)/2, CV = T)
table(V58, z$class)

##
## V58    0    1
##   0 2086  695
##   1   86 1723
```

Vamos a realizar una valoración de los resultados de clasificación con el análisis discriminante lineal utilizando remuestreo. Elegimos la muestra de entrenamiento como una selección aleatoria de los datos.

```
entrenamiento <- sample(nrow(x), 2000)
```

Vemos la distribución del correo en la muestra de entrenamiento.

```
table(y[entrenamiento])

##
##    0    1
## 1207  793
```

Realizamos el análisis discriminante lineal.

```
z <- lda(fmla, data = x, prior = c(1, 1)/2, subset = entrenamiento)
```

Vemos la tabla de clasificación sobre los datos de entrenamiento.

```
table(predict(z, x[entrenamiento, ])$class, y[entrenamiento])

##
##        0    1
##   0 1149  108
##   1   58  685
```

Vemos la tabla de clasificación sobre el resto de los datos.

```
table(predict(z, x[-entrenamiento, ])$class, y[-entrenamiento])

##
##        0    1
##   0 1486  162
##   1   95  858
```

Repetimos en muestras con reemplazamiento.

```
entrenamiento <- sample(nrow(x), 2000, replace = T)
table(y[entrenamiento])

##
##    0    1
## 1220  780

z <- lda(fmla, data = x, prior = c(1, 1)/2, subset = entrenamiento)
```

Vemos la tabla de clasificación sobre los datos de entrenamiento.

```
table(predict(z, x[entrenamiento, ])$class, y[entrenamiento])

##
##        0    1
##   0 1155  112
##   1   65  668
```

Vemos la tabla de clasificación sobre el resto de los datos.

```
table(predict(z, x[-entrenamiento, ])$class, y[-entrenamiento])

##
##        0    1
##   0 1687  190
##   1  101  978
```

5.7. Variables discriminantes canónicas o discriminantes lineales

Vamos a estudiar una técnica de reducción de la dimensión relacionada con el planteamiento que del análisis discriminante lineal hizo Fisher. Consideramos las matrices W y B definidas como

$$W = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{ij} - x_{i\cdot})(x_{ij} - \bar{x}_{i\cdot})' = \sum_{i=1}^{g} (n_i - 1) S_i, \tag{5.17}$$

y

$$B = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (\bar{x}_{i\cdot} - \bar{x}_{\cdot\cdot})(\bar{x}_{i\cdot} - \bar{x}_{\cdot\cdot})' = \sum_{i=1}^{g} n_i (\bar{x}_{i\cdot} - \bar{x}_{\cdot\cdot})(\bar{x}_{i\cdot} - \bar{x}_{\cdot\cdot})' \tag{5.18}$$

Notemos que

$$\mathbf{S}_p = \frac{W}{n - g} \tag{5.19}$$

Estas matrices reciben el nombre de matrices *intra grupos* y *entre grupos* respectivamente. Son las versiones matriciales de las sumas de cuadrados intra y entre grupos habituales en análisis de la varianza.

Es claro que cuando más agrupados estén los datos dentro de los grupos y más separados estén para grupos distintos tendremos que la magnitud de W ha de ser menor que la de B. Supongamos que reducimos las observaciones multivariantes x_{ij} a datos univariantes mediante tomando $z_{ij} = c'x_{ij}$. Las sumas de cuadrados intra y entre vendrían dadas por $c'Wc$ y $c'Bc$. El cociente $F_c = c'Bc/c'Wc$ nos compara la variabilidad intra con la variabilidad entre. Fisher (1936) introdujo el análisis discriminante lineal buscando el vector c tal que el cociente F_c sea el mayor posible. Ese fue su objetivo inicial.

La matriz W es suma de matrices semidefinidas positivas por lo que es definida positiva y consideramos su descomposición de Cholesky dada por $W = T'T$. Tomamos $b = Tc$. Se tiene

$$F_c = \frac{c'Bc}{c'Wc} = \frac{b'(T')^{-1}BT^{-1}b}{b'b} = \frac{b'Ab}{b'b} = a'Aa, \tag{5.20}$$

donde $a = b/ \parallel b \parallel$, esto es, a tiene módulo unitario y $A = (T')^{-1}BT^{-1}$. Se nos plantea el problema de maximizar $a'Aa$ con la restriccion de $\parallel a'a \parallel= 1$. Por resultados estándar del álgebra lineal se tiene que a_1 es el vector propio de A con el mayor propio λ_1 verificando que $\lambda_1 = a_1'Aa_i$. Hemos encontrado una combinación lineal que, en el sentido que hemos indicado, es óptima a la hora de separar los grupos. Parece lógico buscar la siguiente combinación lineal que verifique el mismo criterio de optimalidad pero que el vector correspondiente sea ortogonal al ya calculado. Nos planteamos pues maximizar $a'Aa$ con la restriccion de $\parallel a \parallel= 1$ y que sea ortogonal con el anterior. La solución viene dada por el vector propio de A asociado a su segundo valor propio por orden de magnitud, λ_2 ($Aa_2 = \lambda_2 a_2$ por lo que $\lambda_2 = a_2'Aa_2$). Procedemos del mismo modo obteniendo k direcciones ortogonales que nos dan las combinaciones óptimas que separan a los grupos. El valor de k es el mínimo entre el número de grupos menos uno, $g-1$ y el número de datos n, $k = \min g-1, n$. Notemos que los sucesivos a_r constituyen una base ortonormal tales que

$$(T')^{-1}BT^{-1}a_r = Aa_r = \lambda_r a_r,$$

con $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$. Si multiplicamos por la izquierda por la matriz $T^{-1}(n-g)^{1/2}$ se deduce que

$$W^{-1}Bc_r = \lambda_r c_r,$$

donde $c_r = (n-g)^{1/2}T^{-1}a_r$. En consecuencia $W^{-1}B$ tiene valores propios λ_r y vectores propios c_r con $r = 1, \ldots, k$. Además los vectores $a_r = Tc_r(n-g)^{-1/2}$ constituyen una base ortonormal. Consideremos la matriz C que tiene por fila r-ésima el vector c_r. Sea $z_{ij} = Cx_{ij}$. Estos valores reciben el nombre de *coordenadas discriminantes*. Vemos que estas coordenadas pretenden destacar las diferencias entre los grupos con un orden decreciente de relevancia. Tenemos que decidir cón cuantas de ellas nos quedamos. Es habitual estudiar los cocientes

$$\frac{\sum_{i=1}^{j}\lambda_i}{\sum_{i=1}^{k}\lambda_i} \tag{5.21}$$

como función de j y quedarse con las coordenadas discriminantes hasta un j próximo a uno.

Es muy importante darse cuenta que las coordenadas discriminantes están tipificadas y son independientes entre ellas. Recordemos que $W = T'T$ y que la matriz de covarianzas agrupada viene dada por $\mathbf{S}_p = W/(g-1)$. Por tanto tendremos que

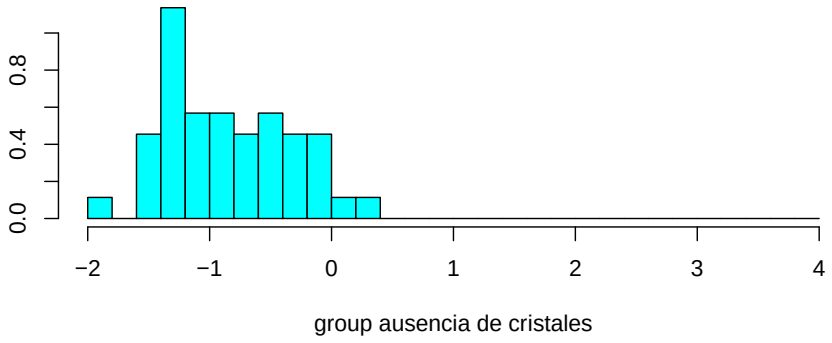
$$c_r'\mathbf{S}_p c_s = (n-g)^{-1}c_r'T'Tc_s = a_r'a_s = \delta_{rs}, \tag{5.22}$$

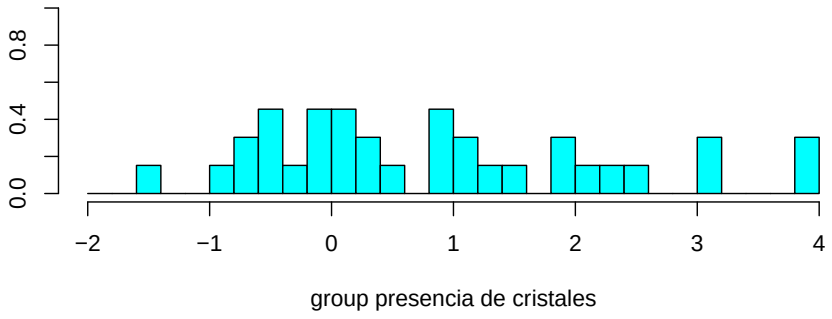
donde $\delta_{rs} = 1$ si $r = s$ y cero en otro caso. Tenemos pues que $C\mathbf{S}_p C' = I_k$. Asumiendo que las matrices de dispersión son iguales para los distintos grupos tendremos que $cov(c_r' x_{ij}, c_s' x_{ij}) = c_r' \Sigma c_s$ y reemplazando Σ por $\mathbf{S}_p$ tenemos que los z_{ij} tienen covarianzas muestrales nulas y varianzas muestrales unitarias.

Nota de R 28 (Variables discriminantes canónicas) *Repetimos el análisis discriminante tanto para los datos de cristales en la orina como para los datos de la diabetes.*

En la siguietne figura vemos un histograma de la primera variable discriminante canónica en el primer y segundo grupo.

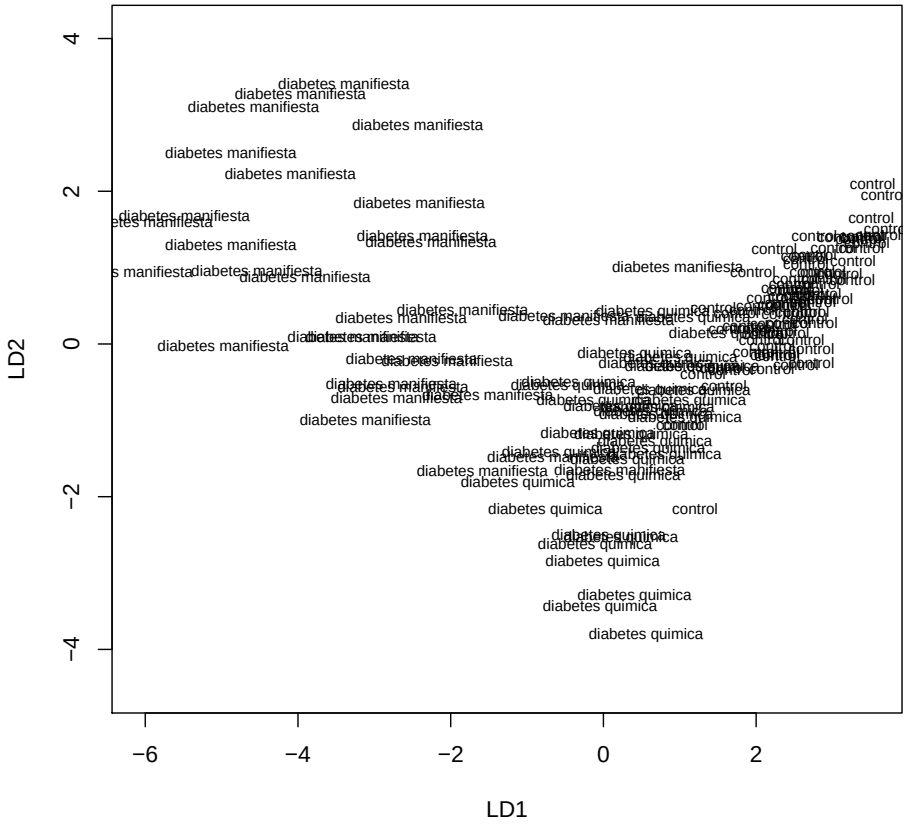
```
plot(z)
```



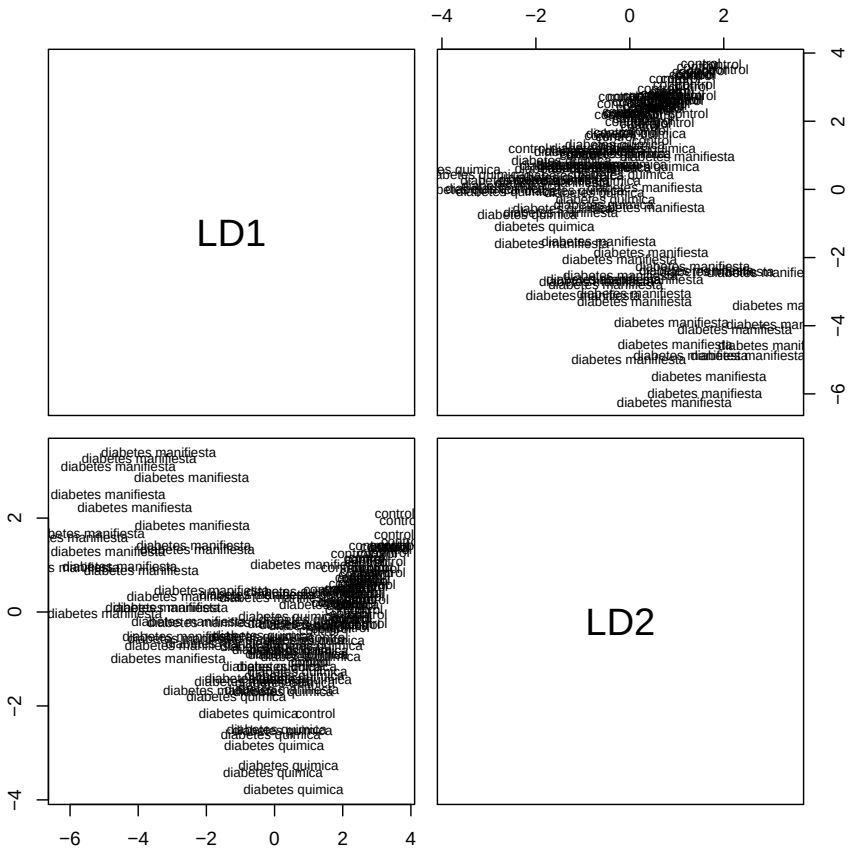


Para los datos de las diabetes podemos tendremos dos variables discriminantes canónicas. Mostramos dos posibles representaciones gráficas.

```
plot(z)
```


LD2
LD1
4
2
0
−2
−4
−6
−4
−2
0
2
diabetes manifiesta
control
diabetes quimica

```
pairs(z)
```



5.8. Algunos ejemplos

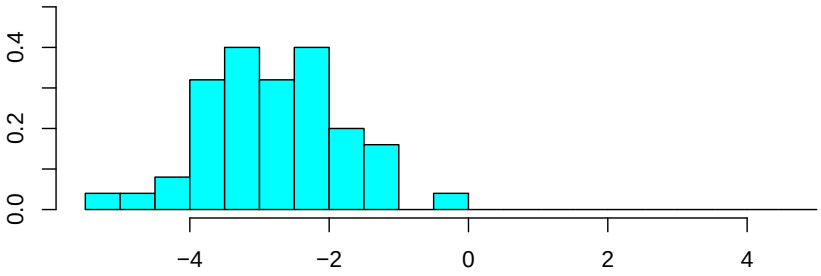
Nota de R 29 *En la sección* **??** *tenemos una descripción de los mismos. Realizamos un análisis discriminante en donde a partir de las características morfológicas pretendemos saber si es una abeja reina o bien es una obrera.*

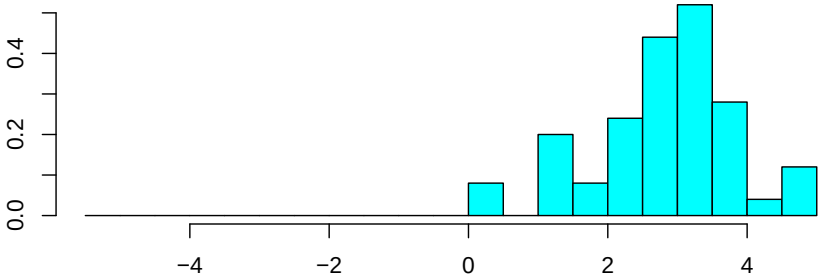
```
library(MASS)
x <- read.table(file = "../data/wasp.dat", header = T)
attach(x)
```

Aplicamos un análisis discriminante lineal y mostramos los histogramas de la variable discriminante canónica.

```
z <- lda(caste ~ TL + WL + HH + HW + TH + TW + G1L + G2Wa, prior = c(1, 1)/2)
```

```
plot(z)
```





Los histogramas muestran que los valores de la variable discriminante son claramente distintos en cada grupo. La tabla de clasificación es la siguiente.

```
z <- lda(caste ~ TL + WL + HH + HW + TH + TW + G1L + G2Wa, prior = c(1, 1)/2, CV = TRUE)
table(caste, z$class)

##
## caste  Q  W
##     Q 49  1
##     W  1 49
```

Mostrando que los resultados de la clasificación son realmente buenos.

Nota de R 30 (Datos wbca) *En la sección* **??** *tenemos la descripción de los datos. Básicamente tenemos personas enfermas de cáncer y no enfermos y pretendemos clasificar en estos dos posibles grupos. Pretendemos clasificar a la paciente como enferma o no de cáncer utilizando el resto de las variables.*

```
library(faraway)
library(MASS)
attach(wbca)
```

Realizamos para ello un análisis discriminante lineal y, como tenemos solamente dos grupos, mostramos los histogramas de la variable discriminante canónica en cada uno de los grupos.

```
wbca.lda <- lda(Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick + UShap + USize
    CV = T)
table(Class, wbca.lda$class)

##
## Class   0   1
##     0 219  19
##     1   7 436
```

Tenemos una muestra bastante grande. Elegimos una muestra de entrenamiento para estimar las probabilidades a posteriori. Para ellos elegimos al azar 400 mujeres de la muestra inicial. Mostramos la tabla de clasificación.

```
train <- sample(1:nrow(wbca), 400)
wbca.lda0 <- lda(Class ~ Adhes + BNucl + Chrom + Epith + Mitos + NNucl + Thick + UShap + USiz
    subset = train, CV = T)
table(wbca.lda0$class[train], Class[train])

##
##      0  1
##   0 36 53
##   1 65 91
```