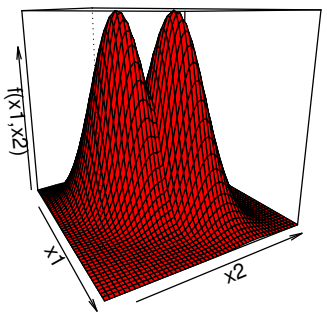
Universidad Politécnica de Cartagena

CAMPUS MARE NOSTRUM

UNIVERSIDAD DE MURCIA

Análisis Estadístico Multivariante



Jorge Navarro

2024

Índice general

Índice de figuras

Índice de tablas

Prefacio

Este libro corresponde a los contenidos de una asignatura de estadística multivariante para los grados en Matemáticas y Ciencias de Datos (o grados/master de ciencias en general). Como principal novedad incluye los comandos del programa estadístico R para la resolución de algunos problemas y prácticas e introducciones a las técnicas que se aplican en Aprendizaje Automático (Machine Learning) cuando los datos y/o las variables son muy numerosos.

El programa estadístico R (o RStudio) es de uso libre y se mejora con los procedimientos aportados por los propios usuarios mediante los "paquetes" incorporados a los diversos "repositorios" de internet. El programa R se puede descargar en:

`http://www.r-project.org`

y el programa RStudio en:

`http://www.rstudio.com/ide/download/desktop`.

Se debe instalar primero R y luego RStudio.

Es una gran verdad que cuando no está a nuestro alcance determinar lo que es verdadero,
debemos aceptar aquello que sea más probable.
René Descartes.

1

Estadística Multivariante

En este capítulo se introduce la notación y los resultados básicos necesarios para estudiar vectores aleatorios (es decir variables aleatorias relacionadas entre sí). Algunos de estos resultados ya se han estudiado en la asignatura de primero fundamentos de probabilidad y análisis exploratorio de datos.

1.1. Vectores aleatorios

Supondremos que queremos estudiar k variables sobre una población de "individuos" (objetos). Habitualmente estas variables serán numéricas y tendrán relaciones (correlaciones) entre ellas. En algunos casos tendremos variables cualitativas o discretas que nos indicarán grupos de individuos. Estas variables se representarán mediante vectores aleatorios sobre un espacio de probabilidad. La definición formal es la siguiente.

Definición 1.1. *Un vector aleatorio (v.a.) k dimensional sobre un espacio de probabilidad $(\Omega, \mathcal{S}, \Pr)$ es $X = (X_1, \ldots, X_k)$ tal que $X_i^{-1}(-\infty, x] \in \mathcal{S}$ para todo $x \in \mathbb{R}$ y todo $i = 1, \ldots, k$.*

Note que X es un vector columna (A' representa la traspuesta de A). Estas condiciones nos permiten definir su función de distribución conjunta como

$$F(x_1, \ldots, x_k) = \Pr(X_1 \leq x_1, \ldots, X_n \leq X_k)$$

para todo $x_1, \ldots, x_k \in \mathbb{R}$ (donde las comas en esa probabilidad indican intersecciones).

Diremos que esas variables son independientes si los sucesos

$$\{X_1 \leq x_1\}, \ldots, \{X_n \leq X_k\}$$

son independientes para todo $x_1, \ldots, x_k \in \mathbb{R}$. Esto es equivalente a

$$F(x_1, \ldots, x_k) = \Pr(X_1 \leq x_1) \ldots \Pr(X_n \leq X_k)$$

para todo $x_1, \ldots, x_k \in \mathbb{R}$.

La función $F_i(x) = \Pr(X_i \leq x_i)$ se denomina la marginal $i$ésima y, lógicamente, es la función de distribución de la variable aleatoria X_i.

La definición de v.a. absolutamente continuo es similar a la del caso continuo y se puede establecer como sigue.

Definición 1.2. *Diremos que un vector aleatorio X es absolutamente continuo si existe una función $f : \mathbb{R}^n \to \mathbb{R}$ no negativa (llamada función de densidad) tal que*

$$F(x_1, \ldots, x_k) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_k} f(z_1, \ldots, z_k) dz_k \ldots dz_1$$

para todo $x_1, \ldots, x_k \in \mathbb{R}$.

Conviene mencionar que existen v.a. cuya función de distribución es continua pero que no son absolutamente continuas (tienen una parte singular) y que puede ocurrir que $X_1, \ldots, X_k$ sean absolutamente continuas y que $(X_1, \ldots, X_k)$ no lo sea. Por ejemplo, si X_1 es una v.a. absolutamente continua, entonces el v.a. $X = (X_1, X_1)$ es continuo pero no absolutamente continuo. De hecho, es completamente singular ya que está contenido en la recta $y = x$ que tiene medida cero en $\mathbb{R}^2$. Estos vectores son bastante comunes en Estadística Multivariante. Por ejemplo, ocurre esto si consideramos la notas de unos alumnos y sus medias. En estos casos deberemos eliminar estas variables dependientes del vector.

El modelo más usado es el modelo normal $N_k(\mu, V)$ cuya densidad es

$$f(x) = \frac{1}{\sqrt{\mid V \mid (2\pi)^k}} \exp\left(-\frac{1}{2}(x-\mu)' V^{-1}(x-\mu)\right).$$

para $x \in \mathbb{R}^n$, donde μ es un vector k dimensional y V es una matriz simétrica $k \times k$ simétrica y definida positiva. Para calcular la inversa de V en R haremos

```
V<-matrix(NA,2,2)
V[1,]<-c(1,1/2)
V[2,]<-c(1/2,1)
solve(V)
```

La función de densidad puede calcularse en $x = (1, 1)$ con

```
mu<-c(0,0)
x<-c(1,1)
dmvnorm(x,mu,V)
```

donde necesitamos cargar el paquete `mvtnorm`. Se debe obtener 0.0943539. Para simular 50 datos de este modelo usaremos `rmvnorm(50,mu,V)`. Su gráfica puede verse en la Figura 4.5 (izquierda) los puntos simulados de ese modelo (izquierda). Para realizar el gráfico podemos hacer:

```
f<-function(x1,x2) dmvnorm(data.frame(x1,x2),mu,V)
x<-seq(-3,3,length=50)
y<-seq(-3,3,length=50)
```

```
z<-outer(x,y,f)
persp(x,y,z,xlab='x1',ylab='x2',zlab='f(x1,x2)',col='red')
```

En la gráfica de la derecha podemos ver sus curvas de nivel ($f(x_1, x_2) = c$) y 50 puntos de ese modelo. Se puede realizar con:

```
set.seed(123)
d<-rmvnorm(50,mu,V)
plot(d,xlab="X1",ylab="X2",pch=20,xlim=c(-3,3),lim=c(-3,3))
contour(x,y,z,levels=4,add=T,col='red')
```

donde el comando `set.seed(123)` sirve para que siempre se genere la misma muestra (si queremos otra basta borrarlo).

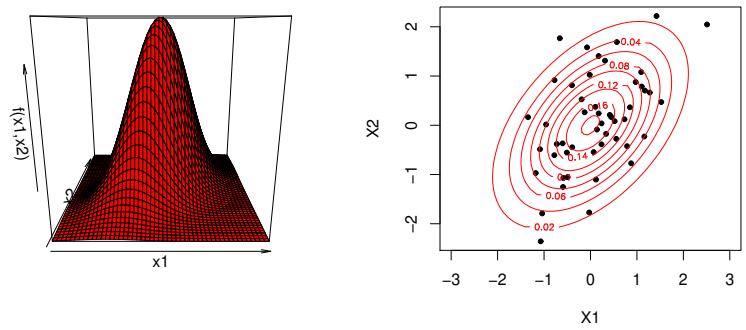


Figura 1.1: Función de densidad para una normal bidimensional con medias cero, varianzas 1 y covarianza 1/2 (izquierda) y sus curvas de nivel junto con una muestra de 50 puntos (derecha).

Aunque los usaremos poco, podemos comentar que un v.a. será discreto existe un conjunto numerable S tal que $\Pr(X \in S) = 1$. El modelo más conocido es la distribución multinomial $M_k(n, p_1, \ldots, p_k)$ donde $(X_1, \ldots, X_k)$ representan los valores observados en un experimento repetido n veces con k opciones con probabilidades constantes $p_i = \Pr(A_i)$ para $i = 1, \ldots, n$. Su función puntual de probabilidad es

$$p(x_1, \ldots, x_k) = \Pr(X_1 = x_1, \ldots, X_k = x_k) = \frac{n!}{x_1! \ldots, x_k!} p_1^{x_1} \ldots p_k^{x_k}$$

para enteros no negativos tales que $x_1 + \cdots + x_k = n$, y donde $p_i \in [0, 1]$ satisfacen $p_1 + \cdots + p_k = 1$. Note que $X_1 + \cdots + X_k = n$ y que X_i sigue una binomial $B(n, p_i)$ con media $E(X_i) = np_i$. Por ejemplo, si tiramos un dado, $n = 60$ veces, $p_i = 1/6$ y los valores esperados son $np_i = 10$ para $i = 1, \ldots, 6$. Para medir las discrepancias entre valores observados y esperados se usa el estadístico

de Pearson

$$T = \sum_{i=1}^{k} \frac{X_i - np_i}{np_i}$$

que sigue una distribución Chi-cuadrado χ^2_{k-1} de Pearson con $k-1$ grados de libertad cuando $n \to \infty$. Su función de distribución se calcula en R con `pchisq`.

Dado un vector aleatorio, su vector de medias es

$$\mu = E(X) = (\mu_1, \ldots, \mu_k) = (E(X_1), \ldots, E(X_k))'$$

(note que es un vector columna). Para medir las relaciones entre las variables X_i y X_j podemos usar la covarianza definida como

$$\sigma_{i,j} = Cov(X_i, X_j) = E((X_i - \mu_i)(X_j - \mu_j)) = E(X_i X_j) - \mu_i \mu_j.$$

Si son independientes, entonces $E(X_i X_j) = E(X_i)E(X_j)$ y, por lo tanto, $Cov(X_i, X_j) = 0$. El recíproco no es cierto.

Note que

$$\sigma_{i,i} = E((X_i - \mu_i)^2) = Var(X_i) = \sigma_i^2.$$

Con ellas se define la matriz de covarianzas (o varianzas-covarianzas) $V = (\sigma_{i,j})$ que es simétrica y definida semipositiva (es decir $x'Vx \geq 0$ para todo $x \in \mathbb{R}^k$). Para el modelo normal multivariante puede probarse que el vector μ y la matriz V de su densidad son el vector de medias y la matriz de covarianzas.

La matriz de covarianzas se puede obtener como

$$V = E\left[(X - \mu)(X - \mu)'\right] = E(XX') - E(\mu\mu'),$$

donde la esperanza de una matriz aleatoria se define como la matriz de las esperanzas de cada variable.

La correlación (lineal de Pearson) entre X_i y X_j se define como

$$\rho_{i,j} = Corr(X_i, X_j) = \frac{\sigma_{i,j}}{\sigma_i \sigma_j}$$

siendo $\rho_{i,i} = Corr(X_i, X_i) = 1$. Puede demostrarse que $-1 \leq \rho_{i,j} \leq 1$ y que mide el grado de relación lineal entre X_i y X_j. Diremos que X_i y X_j son incorreladas si $\rho_{i,j} = 0$. Como hemos mencionado anteriormente, si son independientes serán incorreladas pero el recíproco no es cierto. La matriz de correlaciones es $R = (\rho_{i,j})$.

Análogamente, si X e Y son vectores aleatorios (de dimensiones cualesquiera), se define su matriz de covarianzas como $Cov(X, Y) = (Cov(X_i, Y_j))$ y su matriz de correlaciones como $Corr(X, Y) = (Corr(X_i, Y_j))$. Puede demostrase que

$$Cov(X, Y) = E\left[(X - E(X))(Y - E(Y))'\right].$$

Evidentemente, $Cov(X) = Cov(X, X)$.

Si X, Y, Z son vectores (columna) aleatorios, $a_i \in \mathbb{R}$, $a, b \in \mathbb{R}^k$, y A y B son matrices (de las dimensiones adecuadas), se verifican las propiedades siguientes:

1) $E(a_1 g_1(X) + a_2 g_2(X)) = a_1 E(g_1(X)) + a_2 E(g_2(X))$; $a_1, a_2 \in \mathbb{R}$
2) $X = (Y, Z), E_X(g(Y)) = E_Y(g(Y))$
3) Si (X, Y) independientes, entonces $E(g_1(X)g_2(Y)) = E(g_1(X))E(g_2(Y))$
4) $E(AX + b) = AE(X) + b; A \in M_{m,k}, b' \in \mathbb{R}^k$
5) $Cov(X_i, X_j) = E(X_i X_j) - E(X_i)E(X_j)$
6) $Cov(X_i, X_j) = 0$ si $X_1, \ldots, X_k$ son independientes
7) $Var(X_i + X_j) = Var(X_i) + 2Cov(X_i, X_j) + Var(X_j)$
8) $Cov(aX_i + b, cX_j + d) = acCov(X_i, X_j)$
9) $Cov(X) = E((X - \mu)(X - \mu)') = E(XX') - \mu\mu'$
10) $Var(a'X) = a'Cov(X)a = \sum a_i a_j \sigma_{i,j}$
11) $Cov(AX + b) = ACov(X)A'$
12) $Corr(X_i, X_j) = 0$ si $X_1, \ldots, X_k$ son independientes
13) $Corr(aX_i + b, cX_j + d) = Corr(X_i, X_j)$
14) $-1 \leq Corr(X_i, X_j) \leq 1$
15) $Corr(X_i, aX_i + b) = \pm 1$ (según el signo de a)
16) $Corr(X) = \Delta^{-1} Cov(X) \Delta^{-1}$, donde Δ es la matriz diagonal formada por las desviaciones típicas ($\Delta = diag(\sigma_1, \ldots, \sigma_k)$).
17) $Cov(X, Y) = (Cov(X_i, Y_j)) = Cov(Y, X)'$
18) $Cov(X + Z, Y) = Cov(X, Y) + Cov(Z, Y)$
19) Si X e Y tienen la misma dimensión, entonces

$$Cov(X + Y) = Cov(X) + Cov(X, Y) + Cov(Y, X) + Cov(Y)$$

20) $Cov(AX, BY) = ACov(X, Y)B'$
21) Si X, Y independientes, entonces $Cov(X, Y) = 0$

Sus demostraciones son sencillas. Por ejemplo, la propiedad 10 se puede demostrar con

$$Var(a'X) = Cov(a'X, a'X) = E\left[a'(X - \mu)(X - \mu)'a\right] = a'Cov(X)a.$$

Como consecuencia, se obtiene que la matriz de covarianzas $Cov(X)$ es definida positiva ya que $Var(a'X) \geq 0$. Lo mismo le ocurre a la matriz de correlaciones ya que es la matriz de covarianzas de las v.a. tipificadas $Z_i = (X_i - \mu_i)/\sigma_i$.

1.2. Inferencia

En primer lugar debemos comentar que no queremos hacer un estudio de todas las técnicas de estadística multivariante (análogas a las univariantes como test de medias, covarianzas, etc. Simplemente daremos los resultados básicos para poder aplicar las técnicas que veremos en los capítulos siguientes.

En la práctica, todos los valores (medidas) definidas en la sección anterior serán desconocidas por lo que tendremos que estimarlas. Para ello dispondremos de una muestra de individuos (objetos) en los que se han medido todas las variables.

Por comodidad (aunque son algo antiguos) usaremos los ficheros de datos incluidos en R que se pueden ver con `data()`. Por ejemplo, podemos ver los datos del fichero `LifeCycleSavings` incluido en el programa R haciendo:

```
LifeCycleSavings
```

y para guardarlos en `d`

```
d<-LifeCycleSavings
```

El fichero contiene 5 variables medidas en 50 países diferentes. Los primeros datos se pueden ver en la Tabla 1.1.

Tabla 1.1: Primeros datos del fichero `LifeCycleSavings`.

	sr	pop15	pop75	dpi	ddpi
Australia	11.43	29.35	2.87	2329.68	2.87
Austria	12.07	23.32	4.41	1507.99	3.93
Belgium	13.17	23.80	4.43	2108.47	3.82
Bolivia	5.75	41.89	1.67	189.13	0.22
Brazil	12.88	42.19	0.83	728.47	4.56

La información proporcionada en R sobre datos se puede ver con:

```
help(LifeCycleSavings)
```

donde se indica que:
sr: incremento de los ahorros personales 1960-1970.
pop15: % población menor de 15 años.
pop75: % población mayor de 75.
dpi: ingresos per-capita.
ddpi: crecimiento del dpi 1960-1970.

En general, nuestra muestra se representará como:

La variable Y solo se usará cunado tengamos grupos para indicar a qué grupo pertenece cada dato. Si no hay grupos, supondremos que los objetos $O_i = (X_{i,1}, X_{i,2}, \ldots, X_{i,k})'$ son una muestra aletoria simple de X, es decir, serán vectores (columna) aleatorios independientes con la misma distribución que X. Si hay grupos supondremos lo mismo en cada grupo. En algunos casos usaremos

Tabla 1.2: Muestra.

i	X_1	X_2	...	X_k	Y
O_1	$X_{1,1}$	$X_{1,2}$	...	$X_{1,k}$	Y_1
...	...	...	...	...	...
O_i	$X_{i,1}$	$X_{i,2}$	...	$X_{i,k}$	Y_i
...	...	...	...	...	...
O_n	$X_{n,1}$	$X_{n,2}$	...	$X_{n,k}$	Y_n

la matriz de datos $M = (X_{i,j})$ que será una matriz aleatoria.

Para estima el vector de medias $\mu = E(X)$ usaremos el vector de medias muestrales (también llamado objeto medio)

$$\bar{O} = \bar{X} = (\bar{X}_1, \ldots, \bar{X}_k)' = \frac{1}{n}\sum_{i=1}^{n} O_i$$

donde $\bar{X}_j = \frac{1}{n}\sum_{i=1}^{n} X_{i,j}$. Puede considerarse como un individuo ficticio que obtiene el valor medio en cada variable. Puede demostrarse que es el punto de $\mathbb{R}^k$ que minimiza la suma de las distancias al cuadrado (error cuadrático medio, MSE en inglés), es decir, es la solución de

$$\min_{P\in\mathbb{R}^k} MSE = \sum_{i=1}^{n} d^2(O_i, P)$$

donde d representa la distancia Euclídea, es decir,

$$d(x, y) = \sqrt{\sum_{j=1}^{k}(x_j - y_j)^2}.$$

Se puede demostrar fácilmente que $E(\bar{O}) = \mu$ (es decir es un estimador centrado de μ) y $Cov(\bar{O}) = V/n$.

Para acceder a la primera columna del fichero de datos `LifeCycleSavings` usaremos:
`LifeCycleSavings$sr`

Si queremos acceder directamente a esas variables haremos
`attach(LifeCycleSavings)`
Tras esto basta teclear `sr` para ver estos datos. Si queremos calcular su media muestral haremos

```
mean(sr)
```

El resultado debe ser 9.671. Otra forma es

```
d<-LifeCycleSavings
```

```
mean(d[,1])
```

Para calcular todas las características de estas variables podemos usar:

```
summary(d)
```

Análogamente, para estimar las varianzas y covarianzas usaremos

$$S_{i,j} = \frac{1}{n-1}\sum_{l=1}^{n}(X_{l,i} - \bar{X}_i)(X_{l,j} - \bar{X}_j)$$

y para su matriz

$$S = (S_{i,j}) = \frac{1}{n-1}\sum_{l=1}^{n}(O_l - \overline{O})(O_l - \overline{O})'.$$

Para esta matriz se verifica $E(S) = V$.

Para calcularla y guardarla en R haremos

```
cov(d)
S<-cov(d)
```

En su diagonal tendremos las cuasivarianzas que estimarán las varianzas teóricas. Para comprobar que R divide por $n-1$ podemos hacer:

```
var(sr)
n<-length(sr)
sum((sr-mean(sr))^2)/n
sum((sr-mean(sr))^2)/(n-1)
sum((sr-mean(sr))*(pop15-mean(pop15)))/(n-1)
S[1,2]
```

La matriz

$$\hat{V} = \frac{1}{n}\sum_{l=1}^{n}(O_l - \overline{O})(O_l - \overline{O})'$$

también da buenas estimaciones de V.

Si X tiene una distribución normal $N_k(\mu, V)$ entonces $\bar{O}$ también es normal $N_k(\mu, V/n)$. Además, $\bar{O}$ y $\hat{V}$ (S) son independientes estre sí y son los estimadores máximo verosímiles de μ y V, respectivamente. La distribución de la matriz aleatoria $n\hat{V} = (n-1)S$ se conoce como distribución de Wishart.

Note que al aplicar estas formulas sobre nuestro fichero de datos estamos mezclando unidades diferentes. Esto no tiene sentido por lo que en este caso es mejor usar correlaciones muestrales que

eliminan el efecto de las unidades. Estas se obtienen mediante

$$R_{i,j} = \frac{S_{i,j}}{S_i S_j}$$

donde $S_i = \sqrt{S_{i,i}}$ y $S_j = \sqrt{S_{j,j}}$. Note que si usamos $\hat{V}$ se obtienen los mismo resultados. Esto es equivalente a estandarizar los datos y calcular la matriz de covarianzas muestrales de los datos estandarizados.

Para calcular esta matriz en R y guardarla haremos:

```
cor(d)
R<-cor(d)
```

El resultado puede verse en la Tabla 1.3. Observamos que algunas variables tienen correlaciones positivas y otras negativas.

Tabla 1.3: Correlaciones entre las 5 variables del fichero `LifeCycleSavings`.

	sr	pop15	pop75	dpi	ddpi
sr	1.0000000	-0.45553809	0.31652112	0.2203589	0.30478716
pop15	-0.4555381	1.00000000	-0.90847871	-0.7561881	-0.04782569
pop75	0.3165211	-0.90847871	1.00000000	0.7869995	0.02532138
dpi	0.2203589	-0.75618810	0.78699951	1.0000000	-0.12948552
ddpi	0.3047872	-0.04782569	0.02532138	-0.1294855	1.00000000

1.3. Problemas

1. Calcular la función de densidad normal bidimensional en $(1, 1)$ si las medias son cero, las varianzas 1 y 4 y la covarianza 1.

2. Demostrar las propiedades 1-21.

3. Demostrar el vector de medias muestral es el punto de $\mathbb{R}^k$ que minimiza la suma de las distancias al cuadrado (error cuadrático medio, MSE).

4. Genere una muestra de tamaño 100 de una normal bivariante y estime su media y su matriz de covarianzas usando esa muestra.

5. Calcule y estudie las caractrerísticas de un fichero de datos de R (se pueden ver con `data()`).

2

Regresión lineal

Esta es la técnica más usada por los científicos de datos y sirve de base para otras técnicas como la regresión logística o las redes neuronales. Trataremos de predecir una variable numérica a partir de k variables numéricas (variables predictoras) minimizando el error en la predicción. Para ello necesitamos disponer de una muestra en la que se conozcan dichas variables (aprendizaje supervisado). esta muestra se usará para elegir el mejor modelo y para validar su fiabilidad. Comenzaremos viendo la regresión lineal simple (univariante) que será adaptada para obtener la regresión lineal múltiple y polinómica. Comentaremos los errores más comunes (falta de ajuste, sobreajuste, extrapolación, etc.) y sus posibles soluciones así como técnicas numéricas (algoritmos) para resolver estos problemas de forma aproximada que podrán ser adaptados a casos en los que el número de variables o de datos sean muy grande. Como complemento (opcional) se comentan las principales técnicas de regresión cuantílica comentado sus ventajas e incovenientes.

2.1. Introducción al modelo de regresión lineal

Como hemos comentado, en este modelo se trata de predecir el valor (numérico) de una variable aleatoria (v.a.) Y a partir de unas v.a. predictoras $X_1, \ldots, X_k$). Para ello usaremos una función predictora lineal

$$h_\theta(x_1, \ldots, x_k) := \theta_0 + \theta_1 x_1 + \cdots + \theta_k x_k$$

donde $\theta = (\theta_0, \ldots, \theta_k) \in \mathbb{R}^{k+1}$ serán los parámetros del modelo que se deben elegir de forma que la estimación de Y sea óptima (el error sea mínimo).

Para ello necesitamos disponer de una muestra (training sample) de esas $k+1$ variables sobre n "individuos". Es muy importante que los valores de esta muestra sean correctos y que las predicciones se aplique a individuos "similares". La muestra se representará como

$$(x_1^{(i)}, \ldots, x_k^{(i)}, y^{(i)}), i = 1, \ldots, n.$$

Los datos se suelen representar en forma de tabla situando en la fila i y en la columna j los datos obtenidos para individuo i en la variable X_i para $i = 1, \ldots, k$ y en la última columna los datos para la variable Y.

Por comodidad, a modelo de ejemplo, usaremos en esta memoria conjuntos datos incluidos en el programa R (aunque sean algo antiguos). Estos datos se pueden ver tecleando: `data()` (esta fuente se usará en el libro para indicar que ese texto es un comando de R).

Por ejemplo, para cargar los datos denominados "USArrests" haremos:

```
d<-USArrests
```

La flecha `<-` indica que se guarde en `d` los valores del fichero "USArrests" (por comodidad). Si queremos analizar otro fichero basta cambiar esta orden.

Tecleando `view(d)` podemos ver los datos y con `help('USArrests')` podemos ver la fuente de los datos y explicaciones sobre el signicado de los datos. En este conjunto de datos tenemos cuatro variables númericas: Murder, Assault, UrbanPop, Rape que representan los ratios de arrestos por cada 100000 residente en cada uno de los 50 estados de la unión por asesinatos, asaltos y violaciones. La tercera variable representa el porcentaje de población que vive en áreas urbanas. Los primeros datos pueden verse en la tabla siguiente:

Tabla 2.1: Primeros datos del fichero `USArrests`.

State	Murder	Assault	UrbanPop	Rape
Alabama	13.2	236	58	21.2
Alaska	10.0	263	48	44.5
Arizona	8.1	294	80	31.0
Arkansas	8.8	190	50	19.5
California	9.0	276	91	40.6
Colorado	7.9	204	78	38.7

Aunque estos datos no se ajustan perfectamente a nuestro modelo ya que no está claro que sea una muestra ni qué variable queremos predecir en el futuro (ni en qué estado), nos servirán para presentar el problema.

Supongamos que queremos predecir el ratio de asesinatos $Y = Murder$ en función del porcentaje de población urbana $X = UrbanPop$. Para ver la relación entre estas variables podemos representarlas situando X en el eje horizontal e Y en el vertical ejecutando los comandos siguientes:

```
x<-d$UrbanPop #Elegimos x
y<-d$Murder   #Elegimos y
plot(x,y,xlab='UrbanPop',ylab='Murder') # Hacemos la gráfica x-y
```

La estructura `d$name` se usa para extraer los valores de esa columna y guardarlos en `x` o en `y`. El símbolo `#` sirve para incluir texto (explicaciones) que no se ejecutarán y servirán para aplicar este procedimiento a otros datos. Entonces, con `plot(x,y)` dibujamos los puntos de la muestra para esas variables obteniendo el gráfico de la Figura 2.1. Las opciones `xlab` y `ylab` sirven para poner

las etiquetas deseadas en cada eje. En la figura podemos observar que no parece existir ninguna relación entre esas variables por lo que la predicción no será muy buena. Sin embargo, si usamos como predictor la variable "Assault" cambiando X con `x<-d$Assault` obtenemos el gráfico de la derecha donde sí se aprecia una relación lineal (creciente).

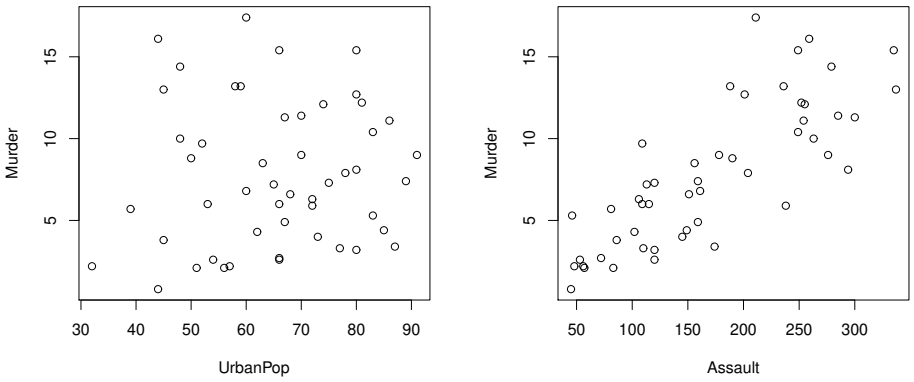


Figura 2.1: Gráficos bidimensionales para las variables UrbanPop y Murder (izquierda) y Assault y Murder (derecha) del fichero de R denominado `USArrests`.

Se pueden hacer todas estas gráficas de forma conjunta mediante `plot(d)` obteniéndose las gráficas de la Figura 2.2. Evidentemente, podemos intentar mejorar estas aproximaciones considerando

$$h_\theta = \theta_0 + \theta_1 Assault + \theta_2 UrbanPop + \theta_3 Rape$$

donde $\theta = (\theta_0, \theta_1, \theta_2, \theta_3) \in \mathbb{R}^4$ son los parámetros del modelo que debemos ajustar para obtener las mejores aproximaciones posibles. Los casos en los que solo usamos una variable están incluidos en este modelo haciendo que los parámetros de las otras variables sean cero (por eso se obtienen mejores resultados). También podemos intentar mejorar estas aproximaciones considerando otras funciones h (no lineales).

Las estadísticas descriptivas de estas variables se pueden obtener con `summary(d)` y se dan en la Tabla 2.2. Incluyen los extremos (mínimo y máximo), los cuartiles, la mediana y la media. Siempre es buena idea usar otras medidas y gráficas para analizar los datos antes de aplicar un procedimiento estadístico multivariante. Para calcular una estimación de las varianzas usaremos `var`. El programa calculará las cuasivarianzas. Podemos comprobarlo haciendo:

```
var(d$Murder)
mu<-mean(d$Murder)
sum((d$Murder-mu)^2)/50
sum((d$Murder-mu)^2)/49
```

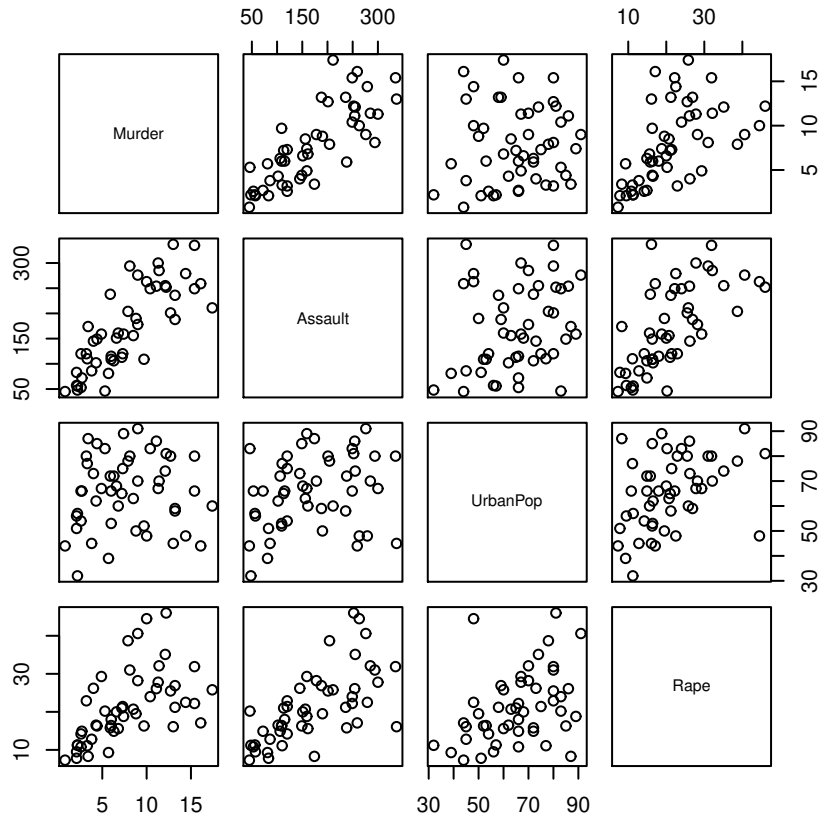


Figura 2.2: Gráficos bidimensionales para las variables del fichero de R denominado `USArrests`.

Tabla 2.2: Estadísticas básicas del fichero `USArrests`.

	Murder	Assault	UrbanPop	Rape
Min.	0.800	45.0	32.00	7.30
1st Qu.	4.075	109.0	54.50	15.07
Median	7.250	159.0	66.00	20.10
Mean	7.788	170.8	65.54	21.23
3rd Qu.	11.250	249.0	77.75	26.18
Max	17.400	337.0	91.00	46.00

2.2. Regresión lineal simple

En esta sección consideramos que se quiere estima una variable Y en función de otra X (es decir $k = 1$). Aunque este es un caso particular del caso general (que se verá posteriomente), nos permitirá entender el problema y sus diversas soluciones y además podremos mostrarlo gráficamente.

2.2.1. Modelo teórico

Desde el punto de vista teórico tendremos un vector aleatorio (X, Y) y queremos construir una nueva variable $h(X)$ que se "parezca" (aproxime) a Y. Los errores (residuos) serán otra variable aleatoria $R = Y - h(X)$ (note que pueden ser positivos o negativos).

Existen diversas reglas para determinar una función objetivo que mida cómo son esos errores y trate de minimizarlos. La más usada es el denominado error cuadrático medio (MSE en inglés) definido como:

$$MSE = E((h(X) - Y)^2).$$

Al elevar al cuadrado los errores conseguimos que sean todos positivos y que los errores pequeños (menores que 1) disminuyan y los grandes (mayores que 1) aumenten, penalizando mucho los errores muy grandes (ya que x^2 crece rápidamente). Consideraremos otras medidas de error posteriormente.

Se puede demostrar que la función h que minimiza el MSE es

$$h_{opt}(x) = E(Y|X = x) = \int_{-\infty}^{\infty} y f_{2|1}(y|x)dy,$$

donde $f_{2|1}(y|x) = f(x, y)/f_1(x)$ para x tales que $f_1(x) > 0$ es la función de densidad condicionada de $(Y|X = x)$ cuando (X, Y) tiene una distribución absolutamente continua con función de densidad conjunta f y marginales f_1 y f_2. Esta función se denomina **curva de regresión** y es el mejor predictor de Y dado X bajo el MSE.

Se puede demostrar que si (X, Y) tienen una distribución normal $N_2(\mu, V)$, donde $\mu = (\mu_X, \mu_Y)'$ es el vector de medias (A' representa la traspuesta de la matriz A),

$$V = \begin{pmatrix} \sigma_{1,1} & \sigma_{1,2} \\ \sigma_{2,1} & \sigma_{2,2} \end{pmatrix}$$

es la matriz de varianzas-covarianzas ($\sigma_{1,1} = \sigma_X^2 = Var(X) = E((X - \mu_X)^2)$, $\sigma_{2,2} = \sigma_Y^2 = Var(Y) = E((Y - \mu_Y)^2)$, y $\sigma_{1,2} = \sigma_{2,1} = Cov(X, Y) = E((X - \mu_X)(Y - \mu_Y)))$, entonces $(Y|X = x) \sim N_1(\bar{\mu}, \bar{\sigma}^2)$, con

$$h_{opt}(x) = \bar{\mu} = E(Y|X = x) = \mu_Y + \frac{\sigma_{1,2}}{\sigma_{1,1}}(x - \mu_X)$$

y

$$\bar{\sigma}^2 = Var(Y|X = x) = \sigma_{2,2} - \frac{\sigma_{1,2}^2}{\sigma_{1,1}}.$$

Note que, en la normal, la curva de regresión h_{opt} es siempre una recta y que la varianza $\bar{\sigma}^2$ no depende de x. Los residuos condicionados $R_x = R|X = x$ también serán normales $R_x \sim N_1(0, \bar{\sigma}^2)$ e idénticamente distribuidos. La curva (recta) de regresión para predecir Y en función de X se puede escribir como

$$\frac{y - \mu_Y}{\sigma_Y} = \rho \frac{x - \mu_X}{\sigma_X}$$

donde $\rho = Cov(X, Y)/(\sigma_X \sigma_Y)$ es el coeficiente de de correlación lineal de Pearson. Note que la recta siempre para por el punto formado por las media (μ_X, μ_Y), que el signo de su pendiente coincide con el signo de ρ y que

$$\bar{\sigma}^2 = \sigma_Y^2 - \frac{\sigma_{1,2}^2}{\sigma_X^2 \sigma_Y^2} \sigma_Y^2 = (1 - \rho^2)\sigma_Y^2 \geq 0$$

por lo que $-1 \leq \rho \leq 1$. Cuando ρ sea ± 1 tendremos ajustes perfectos con residuos cero. La recta (curve) para predecir X a partir de Y se calcula de forma similar y no coincide con la que acabamos de calcular salvo cuando $\rho = \pm 1$.

Para otras distribuciones bivariantes la curva de regresión no tiene porqué ser una recta. Cuando X e Y sean independientes, Y e $Y|X = x$ tiene la misma distribución y la curva óptima

$$h_{opt}(x) = E(Y|X = x) = E(Y)$$

es constante (por lo que también es una recta). En este caso el valor de X no influye en la predicción sobre Y y $\rho = 0$ (recta horizontal).

Si limitamos nuestra función h a una recta, es decir

$$h_\theta(x) = \theta_0 + \theta_1 x$$

y usamos el MSE el objetivo será

$$\min_{\theta \in \mathbb{R}^2} J(\theta_0, \theta_1) := E((h_\theta(X) - Y)^2) = E((\theta_0 + \theta_1 X - Y)^2)$$

donde J se conoce como función "costo" y $J(\theta_0, \theta_1) \geq 0$. Por lo tanto, se trata de minimizar una función real bivariante. Se puede comprobar que es convexa por lo que tendrá un único mínimo que se puede obtener resolviendo el sistema

$$\left.\begin{array}{rcll} \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) & = & 2E(\theta_0 + \theta_1 X - Y) & = 0 \\ \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) & = & 2E((\theta_0 + \theta_1 X - Y)X) & = 0 \end{array}\right\}$$

Estas ecuaciones se conocen como "ecuaciones normales". De la primera ecuación obtenemos

$$\theta_0 = E(Y) - \theta_1 E(X)$$

(con lo que la recta pasará por el punto formado con las medias) que junto con la segunda

$$\theta_0 E(X) + \theta_1 E(X^2) = E(XY)$$

dan

$$E(X)E(Y) - \theta_1 E^2(X) + \theta_1 E(X^2) = E(XY),$$

es decir,

$$\theta_1 Var(X) = Cov(X, Y)$$

ya que $Var(X) = E(X^2) - E^2(X)$ y $Cov(X, Y) = E(XY) - E(X)E(Y)$. Por lo tanto la solución del sistema es

$$\hat{\theta}_1 = \frac{Cov(X, Y)}{Var(X)} = \frac{\sigma_{X,Y}}{\sigma_X^2}$$

y

$$\hat{\theta}_0 = E(Y) - \hat{\theta}_1 E(X) = E(Y) - E(X)\frac{Cov(X, Y)}{Var(X)} = \mu_Y - \mu_X \frac{\sigma_{X,Y}}{\sigma_X^2}.$$

Se puede comprobar que ese punto es efectivamente donde se alcanza el mínimo de la función J que será

$$\min_{\theta \in \mathbb{R}^2} J(\theta_0, \theta_1) = J(\hat{\theta}_0, \hat{\theta}_1).$$

De esta forma, la recta de regresión para estima Y en función de X será

$$h_{\hat{\theta}}(x) = \mu_Y - \mu_X \frac{\sigma_{X,Y}}{\sigma_X^2} + \frac{\sigma_{X,Y}}{\sigma_X^2} x = \mu_Y + \frac{\sigma_{X,Y}}{\sigma_X^2}(x - \mu_X). \tag{2.1}$$

Note que la fórmula es la misma que la de la curva de regresión de la normal (obviamente porque es la que minimiza el MSE). Como antes, esta recta también se puede escribir como

$$\frac{y - \mu_Y}{\sigma_Y} = \rho_{X,Y} \frac{x - \mu_X}{\sigma_X}$$

donde $\rho_{X,Y} = \sigma_{X,Y}/(\sigma_X \sigma_Y)$ es el coeficiente de correlación lineal de Pearson.

Si definimos la variable aleatoria $\hat{Y} = h_{\hat{\theta}}(X) = \hat{\theta}_0 + \hat{\theta}_1 X$ que se usará para estimar Y (verificando $E(\hat{Y}) = E(Y)$ y los residuos como $R = Y - \hat{Y}$ (verificando $E(R) = 0$), tendremos $Y = \hat{Y} + R$, donde

$$\sigma_Y^2 = Var(Y) = Var(\hat{Y}) + Var(R)$$

ya que

$$Var(\hat{Y}) = \hat{\theta}_1^2 \sigma_X^2 = \frac{\sigma_{X,Y}^2}{\sigma_X^2} = \rho_{X,Y}^2 \sigma_Y^2,$$

y

$$Var(R) = Var(Y - \hat{\theta}_1 X) = (1 - \rho_{X,Y}^2)\sigma_Y^2,$$

es decir, la información (varianza) contenida en Y se descompone como

$$\sigma_Y^2 = \rho_{X,Y}^2 \sigma_Y^2 + (1 - \rho_{X,Y}^2)\sigma_Y^2,$$

donde el **coeficiente de determinación** $d_{X,Y} = \rho_{X,Y}^2$ el porcentaje (en tanto por 1) de la información de Y explicada por la recta de regresión (por relaciones lineales de X). Análogamente,

$1-d_{X,Y} = 1-\rho^2_{X,Y}$ indicaría la parte de Y no explicada por esa recta y que se queda en el residuo. Además, se tiene

$$E(\hat{Y}R) = 0,$$

es decir, la variable que se obtiene con la recta de regresión y los residuos son incorrelados. Bajo normalidad, ambas serán normales (por ser combinaciones lineales de X e Y) y, por lo tanto, serían independientes.

2.2.2. Inferencia y predicción. Análisis de los residuos

En la práctica tanto la distribución (densidad) de (X, Y) como todas esas medidas serán desconocidas por lo que tendrán que ser estimadas a partir de una muestra de esas variables (training sample). Si la muestra es grande, algunos datos se pueden dejar aparte para comprobar cómo de fiables serán nuestras estimaciones (en puntos no usados en el cálculo de la recta).

Como hemos comentado anteriormente la muestra se representará como $(x^{(i)}, y^{(i)})$ para $i = 1, \ldots, n$, donde n será el tamaño muestral. Los datos de cada variable se representarán como columnas y todos los datos como una matriz D.

Por comodidad trabajaremos con pocos datos (inventados) para que los cálculos sean más sencillos. Por ejemplo, supongamos que los datos son:

$$(0,1),\ (1,2),\ (1,3),\ (2,3),\ (3,4),\ (4,4) \tag{2.2}$$

con $n = 6$. Para introducir esos datos en R (o RStudio) haremos:

```
x<-c(0,1,1,2,3,4) #training sample
y<-c(1,2,3,3,4,4)
n<-length(x)  #tamaño muestral
```

Es fundamental que tengamos todos los datos, que esas columnas tengan la misma longitud y que no se cambie de orden una única columna (sin cambiar la otra). Para representar la **nube de puntos** (*scatterplot*) haremos: `plot(x,y,ylim=c(0,5))` (ver Figura 2.3, izquierda). Para ver los datos en forma de tabla (data frame) haremos:

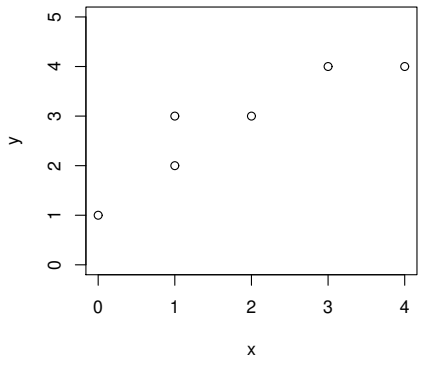
```
d<-data.frame(x,y)
View(d)
```

El resultado se puede ver en la Figura 2.3.

Como en la sección anterior supondremos que queremos aproximar los valores de Y mediante una recta (función lineal) de X, es decir

$$h_\theta(x) = \theta_0 + \theta_1 x$$

donde $\theta = (\theta_0, \theta_1)$ son dos parámetros desconocidos. Para calcular estos parámetros minimizaremos una **función coste empírica** J que nos mida el error cometido. La más utilizada es el error



Filter

	x	y
1	0	1
2	1	2
3	1	3
4	2	3
5	3	4
6	4	4

Showing 1 to 6 of 6 entries, 2 total colu

Figura 2.3: Nube de puntos (izquierda) y tabla de datos (derecha) para la muestra en (2.2).

cuadrático medio (o una función proporcional a él). por ejemplo podemos considerar

$$J(\theta_0,\theta_1) := \frac{1}{2n}\sum_{i=1}^{n}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 = \frac{1}{2n}\sum_{i=1}^{n}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

El objetivo es minimizar esta función en $\mathbb{R}^2$.

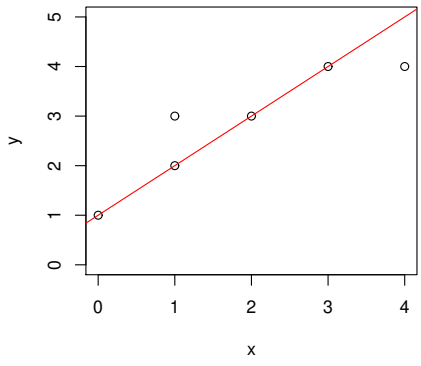
Note que ahora podemos calcular el valor de J para valores dados de los parámetros y que incluso podemos dibujar sus curvas de nivel (donde J es constante). Por ejemplo, para calcular J en $(1,1)$ haremos:

```
h<-function(theta0,theta1,z) theta0+theta1*z
J<-function(theta0,theta1) sum((h(theta0,theta1,x)-y)^2)/(2*n)
J(1,1)
```

Se debe obtener $J(1,1) = 1/6 = 0.1666667$. La recta de regresión y los cálculos se pueden ver el la Figura 2.4. El código es el siguiente:

```
abline(1,1,col='red')
yhat<-h(1,1,x)
error<-(h(1,1,x)-y)^2
d<-data.frame(x,y,yhat,error)
View(d)
2/(2*n)
```

En esa figura vemos que el ajuste es baste bueno pero queremos estar seguros de que esos parámetros son los que optimizan J. Si probamos con otras opciones obtenemos $J(0,2) = 1.916667$,



Filter

	x	y	yhat	error
1	0	1	1	0
2	1	2	2	0
3	1	3	2	1
4	2	3	3	0
5	3	4	4	0
6	4	4	5	1

Showing 1 to 6 of 6 entries, 4 total columns

Figura 2.4: Nube de puntos con la recta $y = 1 + x$ (izquierda) y tabla de datos para calcular el error cuadrático J (derecha) para la muestra en (2.2).

$J(2,0) = 0.9166667$, $J(2,2) = 5.583333$ y $J(2,1/2) = 0.1458333$. Observamos que este último resultado mejora el obtenido con $J(1,1)$.

Para obtener la solución exacta debemos diferenciar J con respecto a los parámetros obteniendo

$$\frac{\partial}{\partial\theta_0}J(\theta_0,\theta_1) = \frac{1}{n}\sum_{i=1}^{n}\left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

y

$$\frac{\partial}{\partial\theta_1}J(\theta_0,\theta_1) = \frac{1}{n}\sum_{i=1}^{n}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x^{(i)}$$

Igualando a cero obtenemos las ecuaciones normales empíricas. De la primera obtenemos

$$\frac{1}{n}\sum_{i=1}^{n}\left(h_\theta(x^{(i)}) - y^{(i)}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right) = 0$$

es decir,

$$\theta_0 + \theta_1\bar{x} - \bar{y} = 0$$

donde

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

y

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n}y^i$$

son las medias muestrales de X e Y, respectivamente. Esta ecuación nos dice que la solución óptima pasa por el punto medio $(\bar{x}, \bar{y})$ (individuo promedio). De la segunda ecuación obtenemos

$$\frac{1}{n}\sum_{i=1}^{n}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x^{(i)} = \frac{1}{n}\sum_{i=1}^{n}\left(\theta_0 x^{(i)} + \theta_1 (x^{(i)})^2 - x^{(i)}y^{(i)}\right) = 0,$$

es decir,

$$\theta_0\bar{x} + \theta_1 a(x,x) - a(x,y) = 0$$

donde

$$a(x,x) = \frac{1}{n}\sum_{i=1}^{n}(x^{(i)})^2$$

y

$$a(x,y) = \frac{1}{n}\sum_{i=1}^{n}x^{(i)}y^{(i)}.$$

Resolviendo este sistema de ecuaciones obtenemos

$$\theta_1(a(x,x) - (\bar{x})^2) = a(x,y) - \bar{x}\bar{y}$$

es decir, la solución óptima es

$$\hat{\theta}_1 = \frac{s_{x,y}}{s_x^2}$$

y

$$\hat{\theta}_0 = \bar{y} - \hat{\theta}_1\bar{x},$$

donde

$$s_{x,y} = a(x,y) - \bar{x}\bar{y} = \frac{1}{n}\sum_{i=1}^{n}x^{(i)}y^{(i)} - \bar{x}\bar{y} = \frac{1}{n}\sum_{i=1}^{n}(x^{(i)} - \bar{x})(y^{(i)} - \bar{y})$$

es la covarianza muestral y

$$s_x^2 = a(x,x) - (\bar{x})^2 = \frac{1}{n}\sum_{i=1}^{n}(x^{(i)})^2 - (\bar{x})^2 = \frac{1}{n}\sum_{i=1}^{n}(x^{(i)} - \bar{x})^2$$

es la varianza muestral de x. Aquí necesitamos suponer que esta varianza no es cero, es decir, que x toma más de un valor. Si no, el sistema tendría infinitas soluciones (la covarianza también sería cero y la segunda ecuación siempre sería cierta) y la más sencilla sería $\hat{\theta}_1 = 0$ y $\hat{\theta}_0 = \bar{y}$. Si la varianza no es cero, el punto es único.

Puede comprobarse que J es convexa y que por lo tanto ese punto es donde se alcanza el valor mínimo de J. Las segundas derivadas parciales son

$$D_{1,1} = \frac{\partial^2}{\partial\theta_0^2}J(\theta_0,\theta_1) = \frac{1}{n}\sum_{i=1}^{n}1 = 1,$$

$$D_{1,2} = \frac{\partial^2}{\partial\theta_1\theta_0} J(\theta_0,\theta_1) = \frac{1}{n}\sum_{i=1}^{n} x^{(i)} = \bar{x}$$

y

$$D_{2,2} = \frac{\partial^2}{\partial\theta_1^2} J(\theta_0,\theta_1) = \frac{1}{n}\sum_{i=1}^{n} \left(x^{(i)}\right)^2 = a(x,x)$$

verificándose $D_{1,1} = 1 > 0$ y si $D = (D_{i.j})$ es la matriz con esas derivadas, tenemos

$$|D| = \begin{vmatrix} 1 & \bar{x} \\ \bar{x} & a(x,x) \end{vmatrix} = a(x,x) - (\bar{x})^2 = s_x^2 > 0$$

por lo que el punto será un mínimo local de J. Como es el único mínimo local en $\mathbb{R}$ y J es continua, será el único mínimo global.

En nuestro ejemplo la varianza no es cero y obtenemos la única solución es

$$\hat{\theta}_1 = \frac{s_{x,y}}{s_x^2} = 0.7230769$$

y

$$\hat{\theta}_0 = \bar{y} - \hat{\theta}_1\bar{x} = 2.833333 - 0.7230769 \cdot 1.833333 = 1.507692$$

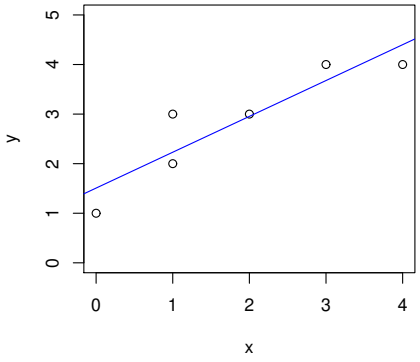
con $J(\hat{\theta}_0, \hat{\theta}_1) = 0.0974359$. El código en R para obtener esta solución es

```
mx<-mean(x)
my<-mean(y)
b<-cov(x,y)/var(x)
a<-my-b*mx
J(a,b)
```

Note que la solución óptima empírica coincide con la que obtendríamos sustituyendo en solución teórica (ver sección anterior) medias, varianzas y covarianzas por sus estimaciones. Los comandos var y cov calculan las cuasivarianzas (es decir dividen por $n-1$ y no por n) pero al calcular $\hat{\theta}_1$ esos denominadores se cancelan y el resultado es correcto. Ocurre lo mismo al calcular el coeficiente de correlación. Puede comprobarse ejecutando este código:

```
var(x)
sum((x-mx)^2)/n
sum((x-mx)^2)/(n-1)
cov(x,y)
sum((x-mx)*(y-my))/n
sum((x-mx)*(y-my))/(n-1)
cor(x,y)
```

obteniéndose $r_{x,y} = 0.9104356$ y $d_{x,y} = r_{x,y}^2 = 0.8288931$. Los parámetros pueden calcularse de forma automática en R con: `lm(y~x)`.



Filter

	x	y	yhat	error
1	0	1	1.507692	0.257751479
2	1	2	2.230769	0.053254438
3	1	3	2.230769	0.591715976
4	2	3	2.953846	0.002130178
5	3	4	3.676923	0.104378698
6	4	4	4.400000	0.160000000

Showing 1 to 6 of 6 entries, 4 total columns

Figura 2.5: Nube de puntos con la recta de regresión $y = \hat{\theta}_0 + \hat{\theta}_1 x$ (izquierda) y tabla de datos para predecir y y calcular el error cuadrático J (derecha) para la muestra en (2.2).

Ahora podemos volver a dibujar la nube de puntos añadiéndole la recta de regresión (óptima) para predecir y a partir de x (ver Figura 2.5). Podemos usar esa recta para precedir los valores de y y compararlos con sus verdaderos valores calculando los errores cuadráticos en cada caso y J. El código es:

```
plot(x,y,ylim=c(0,5))
abline(a,b,col='blue')
p<-function(x) a+b*x
yhat<-p(x)
error<-(yhat-y)^2
d<-data.frame(x,y,yhat,error)
View(d)
sum(error)/(2*n)
J(a,b)
```

Los errores en esos puntos siempre serán menores que los errores reales cuando estimemos y a partir de x en otros valores. Estas estimaciones no se deben aplicar fuera del rango de valores determinado pr la nube de puntos. Las posiciones relativas de los puntos con respecto a sus estimaciones (recta) sí se pueden utilizar para analizarlos. Por ejemplo, en el primer punto $(0, 1)$ el valor de y es un poquito menor del valor esperado $\hat{y} = 1.507692$. Si un punto (o varios) están muy lejos de la recta podría ser un valor atípico (es decir, no perteneciente a esa población). Si hay muchos puntos lejos de la recta, el ajuste no sería bueno y deberíamos buscar otra función h. Veremos diversas opciones en las secciones siguientes pero aquí también podríamos probar a cambiar las variable originales por sus logaritmos, exponenciales, etc. estudiando las nubes de puntos $(x, \log(y))$,

$(\log(x), \log(y))$,$(\log(x), \log(y))$,$(x, \exp(y))$,$(x, 1/y)$, etc. En nuestro ejemplo no es necesario porque el ajuste es bueno y tenemos un coeficiente de determinación de $d_{x,y} = r^2_{x,y} = 0.8288931$ que nos indica que la recta de regresión explica un 82.88931 % de la variabilidad de y.

Para calcular los errores de estimación reales debemos guardar una parte de la muestra (20 − 30 %) para probar nuestra función de predicción h. Para ello necesitamos un tamaño muestral mayor. Veamos un ejemplo.

Ejemplo 2.1. *Vamos a analizar los datos* `iris` *de R. Para verlos haremos*

```
d<-iris
View(d)
```

Podemos observar que tenemos distintas mediciones en 150 *flores de tres especies distintas de flores iris (se pueden ver más detalles con* `help(iris)`*).*

Vamos a analizar las dos primeras variables para la primera especie (no es conveniente mezclar grupos). Los datos están en las 50 *primeras filas. Usaremos los* 40 *primeros para calcular la recta (training sample) y los* 10 *últimos para medir su error (cross validation sample). Lo primero que haremos es dibujar la nube de puntos (ver Figura 2.6, izquierda) para comprobar la aleatoriedad de esta elección distinguiendo los puntos que vamos a usar (círculos negros) de los valores que usaremos para medir su efectividad (cruces rojas). El código usado es:*

```
x<-d$Sepal.Length[1:40]
y<-d$Sepal.Width[1:40]
plot(x,y,ylim=c(2,5))
text(d$Sepal.Length[41:50],d$Sepal.Width[41:50],'x',col='red')
```

A continuación calculamos la recta de regresión con los 40 *primeros datos y la incluímos en el gráfico con:*

```
lm(y~x)
abline(lm(y~x),col='blue')
```

obteniendo $h(x) = -0.2822 + 0.7414x$ *con* $r = cor(x, y) = 0.7438605$ *y* $d = r^2 = 0.5533285$*, es decir la recta solo explica el* 55.33285 % *de la variabilidad de* y*.*

A continuación predecimos los valores de y para todos los individuos calculando los errores cuadráticos medios en los primeros 40 *datos y en los* 10 *últimos obteniendo* $MSE_{TS} = 0.0567245$ *y* $MSE_{CV} = 0.09096423$*, respectivamente, con:*

```
h<-function(x) -0.2822+0.7414*x
x<-d$Sepal.Length[1:50]
y<-d$Sepal.Width[1:50]
yhat<-h(x)
error<-(y-h(x))^2
```

```
e<-data.frame(x,y,yhat,error)
View(e)
mean(error[1:40])
mean(error[41:50])
```

Los errores y los residuos los podemos ver en la Figura 2.7 y se obtienen con:

```
plot(error,ylab='Errores cuadráticos')
plot( y-h(x),ylab='Residuos y-h(x)')
abline(h=0)
```

Como esperábamos el error cuadrático medio es mayor en la muestra CV que en la muestra usada para construir la recta. Sin embargo, en las gráficas de errores podemos observar que gran parte de ese error proviene de la observación de la línea 42 *que es un poco atípica. Si la eliminamos (haciendo* `mean(error[c(41,43:50)])`*) el error cuadrático medio es* $MSE_{CV} = 0.03788617$.

Si (X, Y) *tienen una distribución normal bivariante, los residuos* $R_i = Y^{(i)} - h_{\hat{theta}}(X^{(i)})$ *son i.i.d. con una* $N(0, \sigma)$. *Esto nos permite dar intervalos de confianza para nuestras predicciones estimando* σ *con*

$$\hat{\sigma} = MSE_{CV}^{0.5} = 0.09096423^{0.5} = 0.3016028$$

(usamos la varianza muestral porque en este caso sabemos que la media es cero). Un intervalo de confianza para nuestras predicciones del 99.73002 *se obtendría con*

$$h(x) \pm 3 \cdot \hat{\sigma} = h(x) \pm 0.9048083.$$

En la Figura 2.8 pueden verse las **bandas de confianza** *que, en este caso, contienen a todos los puntos y residuos (aunque no sabemos si la distribución conjunta es normal).*

Si n no es muy grande, mediante programación el método de validación cruzada (CV) se puede aplicar de la forma siguiente. Usamos todos los puntos para calcular la recta de regresión pero para aproximar los errores, cuando vamos a predecir el valor de y para el individuo $i$ésimo, calcularemos la recta que se obtiene sin ese dato y la usaremos para esa predicción. Haciendo esto con los n datos obtenemos n estimaciones del error y haciendo su media estimaremos el funcionamiento en la práctica de nuestras predicciones.

Si n es grande, ese procedimiento no es necesario (además tardaría bastante). Si n es muy grande y los cálculos se relentizan (o no se computan bien), podemos usar solo unos pocos datos (al azar) para obtener la recta de regresión. Por ejemplo, para estimar medias, varianzas y covarianzas es suficiente con tener $n = 1000$ datos. El resto se puede usar para simular nuestro procedimiento de estimación y comprobar sus errores. En datos que dependan del tiempo se deberán usar los últimos n datos para detectar cambios temporales. Las predicciones lineales no se deben usar en valores muy lejanos de series temporales (porque siempre darán valores o muy grandes o muy pequeños). Por ejemplo, si queremos precedir la temperatura de mañana podemos usar los datos de los últimos días pero si queremos predecir la temperatura que hará el mismo día del año próximo deberemos usar los datos en registros históricos para ese día y ese lugar.

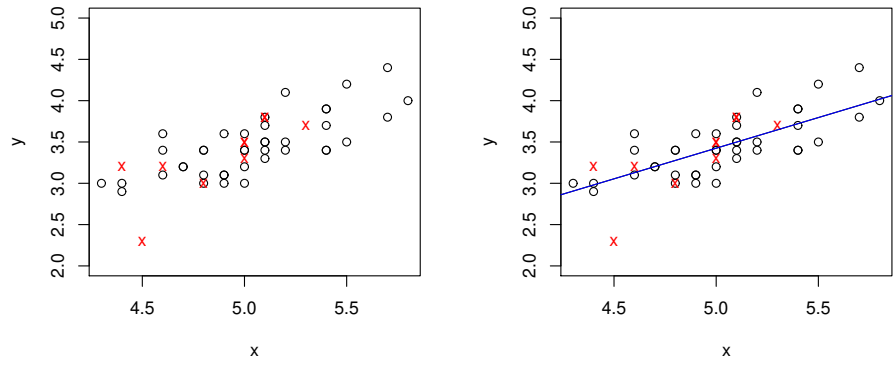


Figura 2.6: Nube de puntos para los 50 primeros datos del fichero `iris` de R (izquierda) y con la recta de regresión (derecha) obtenida con los 40 primeros datos (círculos negros).

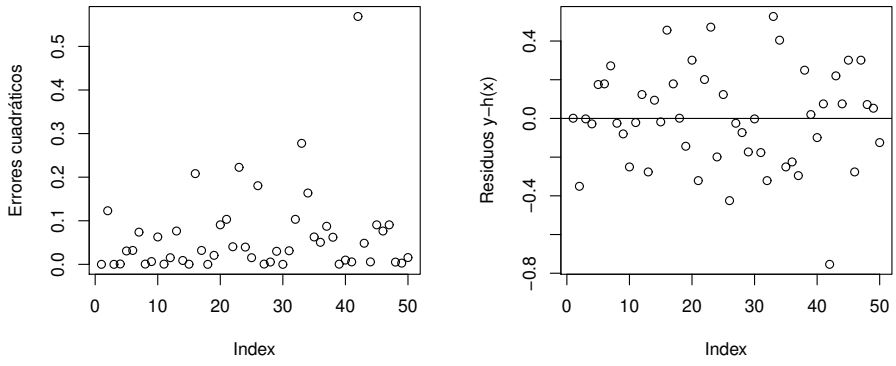


Figura 2.7: Errores cuadráticos (izquierda) y residuos para los 50 primeros datos del fichero `iris` de R con la recta de regresión obtenida con los 40 primeros datos.

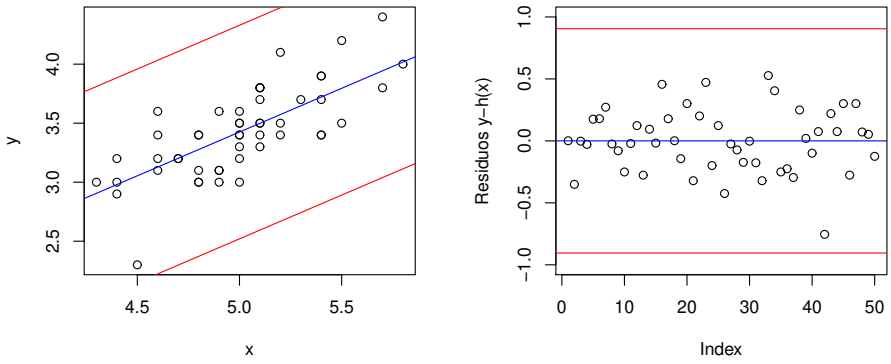


Figura 2.8: Errores cuadráticos (izquierda) y residuos para los 50 primeros datos del fichero `iris` de R con la recta de regresión obtenida con los 40 primeros datos.

2.2.3. Algoritmo del gradiente descendiente

El valor óptimo exacto de J obtenido en la sección anterior puede aproximarse de forma sencilla mediante el algoritmo del gradiente descendiente. En la regresión lineal simple no es necesario pero en otros casos veremos que este algoritmo es muy útil para calcular (aproximar) las soluciones óptimas. Lo mostramos aquí porque es más sencillo de ver (al ser J bivariante) y podremos comparar las soluciones aproximadas con las exactas. En procedimientos posteriores solo tendremos las soluciones aproximadas.

Para mostrar el algoritmo consideraremos primero una función coste univariante. Por ejemplo, supongamos que queremos minimizar $J(x) = x^2 + x + 1$ en $\mathbb{R}$. El algoritmo es el siguiente:

Algoritmo 1.1: Gradiente descendiente univariante:
Paso 0: Sean $x_1 \in \mathbb{R}$ y $\alpha \in \mathbb{R}^+$.
Paso 1: Hacer $x_{i+1} := x_i - \alpha J'(x_i)$.
Paso 2: Repetir paso 1 para $i = 1, ..., m$.

Note que si la derivada es positiva (negativa), es decir si J es creciente (decreciente) en ese punto, el algoritmo mueve el punto hacia la izquierda (derecha) para que J disminuya. El salto es proporcional a α y a la derivada. De esta forma, aunque α sea constante, el salto irá disminuyendo cuando nos acerquemos a la solución (que tiene derivada cero).

El valor inicial x_1 se puede fijar por el usuario o tomar de forma aleatoria (cuanto más cerca esté de la solución, más rápido irá el algoritmo). El paso o salto $\alpha > 0$ (learning rate) debe tener un valor intermedio (pequeño). Si es muy pequeño el algoritmo irá muy lento pero si es muy grande el algoritmo puede divergir. Si hay un único mínimo local de J, este algoritmo nos conducirá a él si tomamos un α adecuado. Si hay varios el algoritmo puede llevarnos a un mínimo local que no sea

el mínimo global.

Veamos como se aplica a esa función J (que tiene un único mínimo local en $x = -1/2$). Primero calculamos su derivada

$$J'(x) = 2x + 1.$$

Tomamos $x_1 = 2$ y $\alpha = 1$ (paso 1). El paso 2, haría

$$x_2 = x_1 - \alpha J'(x_1) = 2 - 1 \cdot 5 = -3$$

y el segundo

$$x_3 = x_2 - \alpha J'(x_2) = -3 - 1 \cdot (-5) = 2$$

con lo que la sucesión es $2, -3, 2, -3, \ldots$ (divergente). El código en R podría ser:

```
J<-function(x) x^2+x+1
Jp<-function(x) 2*x+1
m<-10
x<-1:(m+1)
x[1]<-2
alfa<-1
for (i in 1:m) x[i+1]<- x[i]-alfa*Jp(x[i])
x
```

En una primera iteración conviene usar m pequeño ($m = 10$) para probar el algoritmo. Claramente, en este caso, debemos disminuir α. Con $x_1 = 2$, $\alpha = 1/3$ y $m = 10$ obtenemos la sucesión:

i	2	3	4	5	6
x_i	0.3333333	-0.2222222	-0.4074074	-0.4691358	-0.4897119
$J(x_i)$	1.4444444	0.8271605	0.7585734	0.7509526	0.7501058
i	7	8	9	10	11
x_i	-0.4965706	-0.4988569	-0.4996190	-0.4998730	-0.4999577
$J(x_i)$	0.7500118	0.7500013	0.7500001	0.7500000	0.7500000

Vemos que con solo $m = 10$ iteraciones conseguimos alcanzar una buena aproximación del valor mínimo que se alcanza en $x = -1/2$ con $J(-1/2) = 1/4 - 1/2 + 1 = 3/4 = 0.75$. La convergencia se puede ver en la Figura 2.10. El código para obtener estos gráficos es:

```
plot(J(x),ylim=c(0,7),cex=0.5)
abline(h=0.75,col='blue')
curve(J(x),-2,2,ylim=c(0,7))
text(x,J(x),'x',col='red')
```

Sin embargo, si tomamos $\alpha = 0.1$ la convergencia es mucho más lenta y el mínimo se alcanza en la iteración $i = 81$ aunque con $i = 43$ ya se obtiene una buena aproximación (ver Figura **??**).

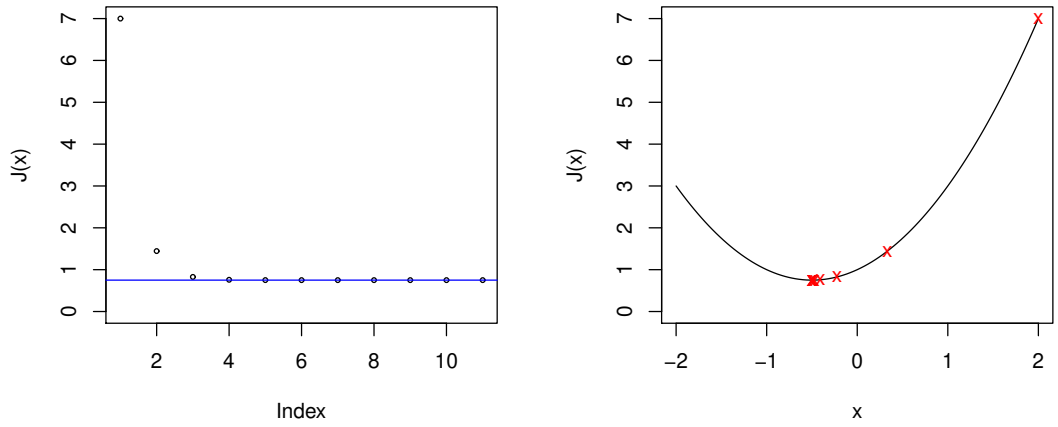


Figura 2.9: Convergencia del algoritmo gradiente descendiente para $J(x) = x^2 + x + 1$ con $x_1 = 2$, $\alpha = 1/3$ y $m = 10$.

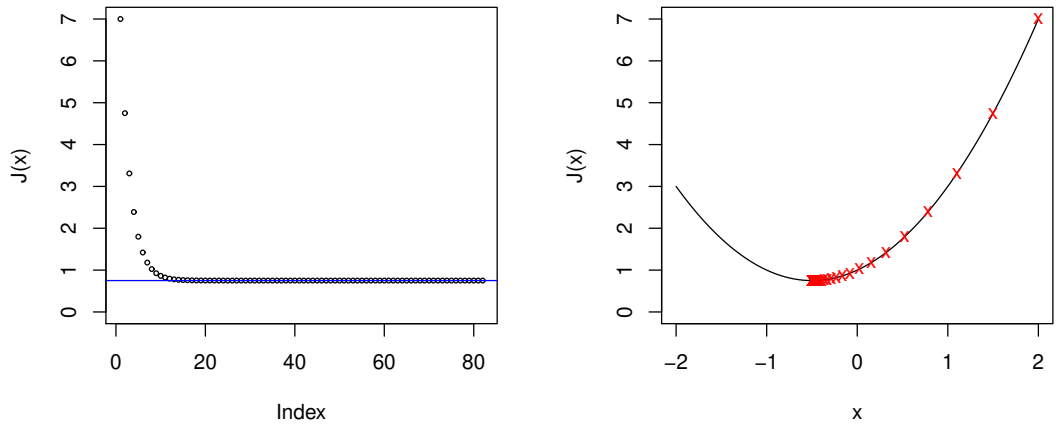


Figura 2.10: Convergencia del algoritmo gradiente descendiente para $J(x) = x^2 + x + 1$ con $x_1 = 2$, $\alpha = 0.1$ y $m = 100$.

Siempre es preferible que el algoritmo converja más lento a que no lo haga (ya que basta con aumentar m). Pruebe con distintos valores de α y distintas funciones J.

Una vez que tenemos claro el funcionamiento del algoritmo, podemos ver cómo aplicarlo a nuestro problema de regresión lineal que trata de minimizar la función costo J. En la mayoría de los problemas (datos) esta función tendrá un único mínimo. Además, podemos comprobar el funcionamiento del algoritmo comparando la solución aproximada con la exacta. La idea es la misma pero ahora la función J es bivariante con lo que el algoritmo será el siguiente:

Algoritmo 1.2: Gradiente Descendiente (GD) Bivariante:
Paso 0: Sean $\theta_0^{(1)}, \theta_1^{(1)} \in \mathbb{R}$ y $\alpha \in \mathbb{R}^+$.
Paso 1: Hacer (simultaneamente):

$$\theta_0^{(i+1)} := \theta_0^{(i)} - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0^{(i)}, \theta_1^{(i)}),$$

$$\theta_1^{(i+1)} := \theta_1^{(i)} - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0^{(i)}, \theta_1^{(i)}).$$

Paso 2: Repetir paso 1 para $i = 1, ..., m$.

En nuestro caso, las derivadas parciales valen

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right)$$

y

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{n} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x^{(i)}$$

y el algoritmo se puede programar como sigue:

```
# Linear Regression with GD:
x<-c(0,1,1,2,3,4) #training sample
y<-c(1,2,3,3,4,4)
n<-length(x)   # Tamaño muestral
h<-function(theta0,theta1,z) theta0+theta1*z
J<-function(theta0,theta1) sum((h(theta0,theta1,x)-y)^2)/(2*n)
J0<-function(theta0,theta1) sum((h(theta0,theta1,x)-y))/n
J1<-function(theta0,theta1) sum((h(theta0,theta1,x)-y)*x)/n
m<-10   # Número de interaciones
theta0<-1:(m+1)
theta1<-1:(m+1)
Js<-1:(m+1)
theta0[1]<-2 # valores iniciales
theta1[1]<-2
```

```
Js[1]<-J(theta0[1],theta1[1])
alfa<-0.1 # learning rate
for (i in 1:m) {
theta0[i+1]<-theta0[i]-alfa*J0(theta0[i],theta1[i])
theta1[i+1]<-theta1[i]-alfa*J1(theta0[i],theta1[i])
Js[i+1]<-J(theta0[i+1],theta1[i+1])
print(Js[i+1])
}
a<-data.frame(theta0,theta1,Js)
View(a)
plot(Js,ylim=c(0,7),cex=0.5)
lm(y∼x)
min<-J(1.5077,0.7231)
abline(h=min,col='blue')
Js[m+1]
min
```

El resultado puede verse en la Figura 2.11. El valor mínimo en la secuencia es $J(1.514462, 0.7207397) = 0.09744391$ mientras que el mínimo con la solución exacta es $J(1.5077, 0.7231) = 0.0974359$. Aumentando m podemos obtener un mejor resultado aunque este ya es bastante bueno. Note que el hecho de que los valores J permanezcan casi constantes nos indica que estamos cerca del mínimo (este podría ser otro criterio para parar el algoritmo). Lo mismo ocurre con la secuencia de puntos. Con $m = 100$ se obtiene $J(1.508093, 0.7229259) = 0.09743593$. La secuencia de rectas de regresión puede verse en la Figura 2.12, izquierda. También podemos ver la secuencia de puntos en la gráfica de curvas de nivel de J (Figura 2.12, derecha). El código para obtener este gráfico es:

```
z<-matrix(NA,1000,1000)
x1<-seq(0,2,length=1000)
x2<-seq(0,2,length=1000)
for (i in 1:1000)
for (j in 1:1000) z[i,j]<-J(x1[i],x2[j])
contour(x1,x2,z,xlab='theta0',ylab='theta1')
points(theta0[1:m],theta1[1:m],col='red',cex=0.5)
```

Obviamente, si punto inicial está cerca del óptimo la convergencia es más rápida. Si no tenemos una idea de donde está el mínimo una buena opción inicial es $\theta_0 = \bar{y}$ y $\theta_1 = 0$ (es decir una recta horizontal que pasa por $(\bar{x}, \bar{y})$. El algoritmo GD funciona mejor si x e y tienen escalas similares. En este caso el punto inicial puede ser $\theta_0 = \theta_1 = 0$. Para conseguir esto podemos dividir por las respectivas desviaciones estándar o (mejor) estandarizar ambas variables considerando

$$x^* = \frac{x - \bar{x}}{s_x}$$

e

$$y^* = \frac{y - \bar{y}}{s_y}$$

(o las que se obtienen con las respectivas cuasidesviaciones estándar). La solución óptima nos permitirá predecir y^* a partir de x^* mediante

$$y^* = \theta_0^* + \theta_1^* x^*.$$

Si queremos predecir y a partir de x deberemos usar:

$$y = s_y y^* - \bar{y} = s_y \left(\theta_0^* + \theta_1^* \frac{x - \bar{x}}{s_x} \right) - \bar{y} = s_y \theta_0^* - \bar{y} - \frac{s_y}{s_x} \theta_1^* \bar{x} + \frac{s_y}{s_x} \theta_1^* x,$$

es decir, $y = \theta_0 + \theta_1 x$ con

$$\theta_0 = s_y \theta_0^* - \bar{y} - \frac{s_y}{s_x} \theta_1^* \bar{x}$$

y

$$\theta_1 = \frac{s_y}{s_x} \theta_1^*.$$

Las correlaciones coinciden. Las variables (muestras) estandarizadas se pueden obtener en R con el comando `scale(x)` pero tenga en cuenta que R usa las cuasidesviaciones estándar. Si queremos usar las desviaciones estándar debemos hacer:

```
n<-length(x) sx<-(sum((x-mean(x))^2)/n)^0.5
xa<-(x-mean(x))/sx
```

Otra opción es definir una nueva función como

```
escala<-function(x) (x-mean(x))/(sum((x-mean(x))^2)/length(x))^0.5
escala(x)
escala(y)
```

2.3. Regresión lineal múltiple

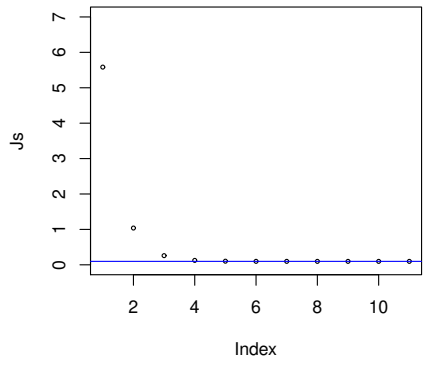
2.3.1. Modelo teórico

En el caso general queremos predecir Y (o X_{k+1}) a partir de k variables $X_1, \ldots, X_k$ (sobre el mismo espacio de probabilidad). En el modelo lineal queremos construir una función

$$h_\theta(x_1, \ldots, x_k) = \theta_0 + \theta_1 x_1 + \cdots + \theta_k x_k$$

de forma que $h_\theta(X_1, \ldots, X_k)$ esté lo más cerca posible de Y. Para medir el error usaremos de nuevo el error cuadrático medio

$$MSE(\theta) = E((h_\theta(X_1, \ldots, X_k) - Y)^2).$$



Filter

	theta0	theta1	Js
1	2.000000	2.0000000	5.58333333
2	1.716667	1.2500000	1.03840278
3	1.599167	0.9394444	0.25884366
4	1.550352	0.8108843	0.12513099
5	1.529988	0.7576962	0.10219563
6	1.521411	0.7357221	0.09826113
7	1.517721	0.7266736	0.09758573
8	1.516059	0.7229767	0.09746939
9	1.515241	0.7214946	0.09744896
10	1.514776	0.7209282	0.09744501
11	1.514462	0.7207397	0.09744391

Figura 2.11: Convergencia del algoritmo gradiente descendiente para regresión lineal con valores iniciales $\theta_0 = \theta_1 = 2$, $\alpha = 0.1$ y $m = 10$ iteraciones.

Si consideramos una nueva variable constante (degenerada) $X_0 = 1$, podemos escribir ese error como

$$MSE(\theta) = E((\theta' X - Y)^2)$$

donde $X = (X_1, \ldots, X_k)'$ (A' es la traspuesta de A) y $\theta' = (\theta_0, \theta_1, \ldots, \theta_k) \in \mathbb{R}^{k+1}$. Como en el caso $k = 1$ se puede comprobar que esta función es convexa por lo que tendrá un único mínimo $\hat{\theta}' \in \mathbb{R}^{k+1}$. Para detectarlo hacemos las derivadas parciales iguales a cero

$$\frac{\partial}{\partial \theta_j} MSE(\theta) = E(2(\theta' X - Y)X_j) = 0$$

obteniendo

$$\theta' E(XX_j) = E(YX_j)$$

para $j = 0, \ldots, k$, es decir,

$$\theta' E(XX') = E(YX')$$

o

$$E(XX')\theta = E(XY)$$

con lo que la solución es

$$\hat{\theta} = (E(XX'))^{-1} E(XY) \tag{2.3}$$

siempre que exista la inversa de la matriz simétrica $A = E(XX') = (E(X_i X_j))_{i,j}$ y donde (por convenio) $E(XY) = (E(X_0 Y), \ldots, E(X_k Y))'$.

Para medir el ajuste podemos usar el coeficiente de correlación múltiple definido como sigue.

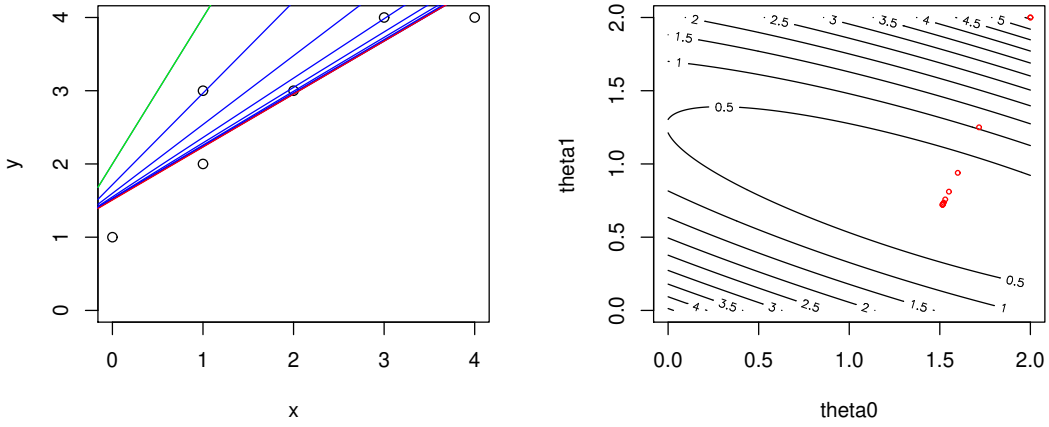


Figura 2.12: Convergencia de las rectas de regresión del algoritmo gradiente descendiente con valores iniciales $\theta_0 = \theta_1 = 2$, $\alpha = 0.1$ (recta verde) y $m = 10$ iteraciones. La recta roja representa la solución exacta.

Definición 2.1. *Si $Z = (X_1, ..., X_k, Y)'$ es un vector aleatorio se llama* ***coeficiente de correlación múltiple*** *al cuadrado de Y respecto de $X = (X_1, ..., X_k)'$ a*

$$Corr^2(X, Y) = \rho^2_{k+1(1,...,k)} = \frac{v'_{1,2} V_X^{-1} v_{1,2}}{\sigma^2_Y}$$

donde

$$Cov(Z) = \begin{pmatrix} V_X & v'_{1,2} \\ v_{1,2} & \sigma^2_Y \end{pmatrix},$$

$V_X = Cov(X)$, $\sigma^2_Y = Var(Y)$ *y*

$$v'_{1,2} = (\sigma_{1,k+1}, ..., \sigma_{k,k+1}) = (Cov(X_1, Y), \ldots, Cov(X_k, Y)).$$

Nótese que si $k = 2$, entonces $\rho^2_{2(1)} = \sigma_{1,2}\sigma^{-1}_{1,1}\sigma_{1,2}/\sigma_{2,2} = \rho^2_{1,2}$. La interpretación de este coeficiente se obtiene del resultado siguiente.

Proposición 2.1. *El coeficiente de correlación múltiple es el máximo de las correlaciones lineales al cuadrado de Y con combinaciones lineales de $X = (X_1, ..., X_k)'$, es decir*

$$\max_{\alpha} Corr^2(Y, \alpha' X) = \rho^2_{1(2,...,k)}$$

y ese máximo se obtiene con $\alpha = \lambda V_X^{-1} v_{1,2}$ para $\lambda \neq 0$.

Demostración. De la definición se tiene

$$\begin{aligned} Corr^2(Y, \alpha' X) &= \frac{(Cov(Y, \alpha' X))^2}{\sigma_Y^2 Var(\alpha' X)} \\ &= \frac{(Cov(Y, X)\alpha)^2}{\sigma_Y^2 Cov(\alpha' X, \alpha' X)} \\ &= \frac{(\alpha' v_{1,2})^2}{\sigma_Y^2 \alpha' V_X \alpha} \\ &= \frac{(\alpha' V_X^{1/2} V_X^{-1/2} v_{1,2})^2}{\sigma_Y^2 \alpha' V_X \alpha} \end{aligned}$$

y, usando la desigualdad de Cauchy-Schwarz (CS) (que dice que $(x'y)^2 \leq (x'x)(y'y)$) para $x' = \alpha' V_X^{1/2}$ e $y = V_X^{-1} v_{1,2}$, se tiene

$$Corr^2(Y, \alpha' X) \leq \frac{\alpha' V_X \alpha v_{1,2}' V_X^{-1} v_{1,2}}{\sigma_Y^2 \alpha' V_X \alpha} = \frac{v_{1,2}' V_X^{-1} v_{1,2}}{\sigma_Y^2},$$

es decir, $\rho^2_{1k+1(1,...,k)}$ es una cota superior. Además, la igualdad en CS se obtiene si y solo si los vectores x e y tienen la misma dirección

$$x = V_X^{1/2} \alpha = \lambda y = \lambda V_X^{-1/2} v_{1,2},$$

es decir, si $\alpha = \lambda V_X^{-1} v_{1,2}$ para $\lambda \neq 0$. □

Como consecuencia obtenemos el resultado siguiente para variables aleatorias independientes.

Proposición 2.2. *Si las variables de $X = (X_1, ..., X_k)'$ son independientes (o incorreladas) entre sí, entonces*

$$Corr^2(X, Y) = \sum_{j=1}^{k} Corr^2(X_j, Y).$$

Demostración. La demostración es inmediata ya que si $\sigma_{i,j} = 0$ para $i \neq j$, $i, j \in \{1, \ldots, n\}$, V_X es diagonal, y se tiene

$$Corr^2(X, Y) = \frac{v_{1,2}' V_X^{-1} v_{1,2}}{\sigma_Y^2} = \sum_{j=1}^{k} \frac{\sigma_{j,k+1}^2}{\sigma_Y^2 \sigma_{j,j}} = \sum_{j=1}^{k} \rho_{j,k+1}^2.$$

□

Observación 2.1. *Otra forma de obtener la solución óptima $\hat{\theta}$ en* (2.3) *es aplicar una regresión lineal simple a Y y $\alpha' X = v_{1,2} V_X^{-1} X$ (una solución de la Proposición 2.1). Recíprocamente, el coeficiente de correlación múltiple también se puede obtener como*

$$Corr^2(X, Y) = Corr^2(\hat{\theta}' X, Y).$$

Note que si sustituimos X_i e Y por $X_i - \mu_i$ e $Y - \mu_Y$ en (2.3), podemos eliminar X_0 (como la solución pasa por el vector de medias, en este caso $\theta_0 = 0$), la correlación no varía y tenemos

$$\hat{\theta} = (E((X - \mu_X)(X - \mu_X)'))^{-1} E((X - \mu_X)(Y - \mu_Y)) = V_X^{-1}\sigma_{1,2},$$

es decir, ambas soluciones coinciden. Así, podemos predecir Y usando

$$Y - \mu_Y = Cov(Y, X)V_X^{-1}(X - \mu_X)$$

es decir,

$$h_{\hat{\theta}}(x) = \mu_Y + Cov(Y, X)V_X^{-1}(x - \mu_X)$$

donde $Cov(Y, X) = (Cov(Y, X_1), \ldots, Cov(Y, X_k))$. Obviamente, si $k = 1$ obtenemos la expresión para el caso $k = 1$ dada en (2.1).

Observación 2.2 (Residuos y selección de variables). *En algunos casos podemos querer detectar las variables del conjunto $X_1, \ldots, X_k$ que mejor predicen Y. Podemos proceder de diversas formas.*

Una opción es seleccionar en primer lugar la variable $Z_1 = X_{j_1}$ que maximice la correlación al cuadrado con Y, es decir, solucionamos

$$j_1 : \max_{j=1,\ldots,k} Corr^2(X_j, Y).$$

A continuación calcularíamos la recta de regresión h_1 basada en Z_1 y el residuo $R_1 = h_1(Z_1) - Y$. En segundo lugar, seleccionamos la variable $Z_2 = X_{j_2}$ con $j_2 \neq j_1$ que más información tenga sobre ese residuo, es decir:

$$j_2 : \max_{j=1,\ldots,k, j\neq j_1} Corr^2(X_j, R_1).$$

Calcularíamos el segundo residuo $R_2 = h_2(Z_2) - R_1$ y continuaríamos así hasta obtener el número de variables deseado o hasta que la correlación múltiple sea tan grande como se desee.

Una segunda opción es fijar de antemano el número de variables deseadas $p < k$ y calcular las correlaciones multiples de todos los subconjuntos con p variables seleccionando al que tenga una mayor correlación múltiple.

Una solución más sencilla sería considerar desde el inicio variables estandarizadas, calcular $\hat{\theta}^*$ para estas variables, y seleccionar primero a las que tengan un mayor coeficiente θ_j^* en valor absoluto (son las que más influyen en el valor de Y ya que todas las variables tienen magnitudes similares). Veremos algunos ejemplos en la sección siguiente.

2.3.2. Inferencia y predicción

Como en el caso univariante, los valores teóricos deben ser estimados en la práctica. Una primera opción es estimar las medias, varianzas y cuasivarianzas y usarlas para estimar sus respectivos valores teóricos en las expresiones obtenidas en la sección anterior (asumiendo que tenemos una muestra aleatoria simple).

Otra opción es considerar el problema empírico. Partiremos de una muestra $(x_1^{(i)}, \ldots, x_k^{(i)}, y^{(i)})$, $i = 1, \ldots, n$ (training sample) donde conocemos los valores de y para esos valores de x (son n puntos de $\mathbb{R}^{k+1}$). Los datos se colocarán como $k+1$ columnas (variables) y n filas (objetos o individuos). Cada variable (sus datos) también se puede ver como un punto de $\mathbb{R}^n$.

Como en el caso teórico, para simplificar la notación es conveniente añadir una variable (columna) x_0 con n unos. De esta forma la función de predicción lineal será

$$h_\theta(x) = \theta' x = \theta_0 + \theta_1 x_1 + \ldots \theta_k x_k,$$

donde $\theta = (\theta_0, \ldots, \theta_k)'$ y $x = (x_0, \ldots, x_k)'$. La matriz de datos para x se representará como $M = (m_{i,j}) = (x_j^{(i)})$ para $i = 1, \ldots, n$ (fila) y $j = 0, \ldots, k$ (columna).

El objetivo será minimizar la función coste (proporcional al error cuadrático medio)

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2 = \frac{1}{2n} \sum_{i=1}^{n} (\theta' x^{(i)} - y^{(i)})^2,$$

donde $x^{(i)} = (x_0^{(i)}, \ldots, x_k^{(i)})$ e $y^{(i)}$ representan las medidas del individuo i-ésimo. Esta función también se puede escribir en forma matricial como

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (\theta' x^{(i)} - y^{(i)})(x^{(i)}\theta - y^{(i)}) = \frac{1}{2n} (M\theta - y)'(M\theta - y)$$

siendo $y = (y^{(1)}, \ldots, y^{(n)})$. En muchos programas, esta expresión es más fácil (rápida) de computar. Alternativamente, tenemos

$$J(\theta) = \frac{1}{2n} (\theta' M' - y')(M\theta - y) = \frac{1}{2n} (\theta' M' M\theta - 2\theta' M' y + y' y).$$

De nuevo tenemos una función J convexa y para detectar su valor mínimo haremos las derivadas parciales y trataremos de resolver

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{n} \sum_{i=1}^{n} (\theta' x^{(i)} - y^{(i)}) x_j^{(i)} = 0$$

para $j = 0, 1, \ldots, k$ lo que es equivalente a

$$\frac{1}{n} (\theta' M' - y') M = 0$$

o a

$$M' M\theta - M' y = 0$$

(ecuaciones normales), siendo 0 un vector de ceros (con la dimensión adecuada). Por lo tanto la solución es

$$\hat{\theta} = (M' M)^{-1} M' y \tag{2.4}$$

siempre que exista la inversa de $M'M$. Esta inversa puede no existir por que haya pocos datos ($n < k$) o porque algunas variables sean dependientes (por ejemplo si $X_2 = \lambda X_1$). En esos caso la solución no es única. Para evitarlos debemos tomar más datos en el primer caso y eliminar variables redundantes.

Como la matriz de datos M es una matriz $n \times k$, $M'M$ es una matriz $k \times k$. Si el número de variables k es muy grande (mayor que 10000), podemos tener problemas al calcular su inversa (el tiempo de cálculo es $O(k^3)$).

En esos casos usaremos el algoritmo gradiente descendiente para J visto en la sección anterior. El algoritmo sería el siguiente:

Algoritmo 1.3: Gradiente Descendiente (GD) Multivariante:
Paso 0: Sea $\theta^{(0)} \in \mathbb{R}^{k+1}$ y $\alpha \in \mathbb{R}^+$.
Paso 1: Hacer:

$$\theta_j^{(i+1)} := \theta_j^{(i)} - \alpha \frac{\partial}{\partial \theta_j} J(\theta^{(i)}),$$

para $j = 0, \ldots, k$ (simultáneamente).
Paso 2: Repetir paso 1 para $i = 1, ..., m$ (o hasta que la diferencia de J en cada paso sea menor que $\epsilon = 0.001$).

En este caso, las derivadas parciales valen

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

con lo que la iteración es

$$\theta_j^{(i+1)} := \theta_j^{(i)} - \alpha \frac{1}{n} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)},$$

para $j = 0, 1, \ldots, k$ (simultáneamente). Recordemos que $x_0^{(i)} = 1$ y que es importante que las variables $X_1, \ldots, X_k$ tengan escalas similares (si no, deberemos estandarizarlas). Veamos un ejemplo.

2.3.3. Ejemplo

Supongamos que queremos predecir la variable Y a partir de las variables X_1 y X_2. Para ello disponemos de la muestra siguiente:

i	X_0	X_1	X_2	Y
1	1	0	2	1
2	1	1	2	2
3	1	1	4	3
4	1	2	4	3
5	1	3	2	4
6	1	4	3	4

donde ya hemos incorporado la variable artificial X_0 que siempre vale 1. Note que tenemos $n = 6$ individuos y $k = 2$ variables para predecir Y mediante

$$h_\theta(x) = \theta' x = \theta_0 + \theta_1 x_1 + \theta_2 x_2.$$

Con la notación anterior tendríamos $x^{(2)} = (1, 1, 2)'$, $y = (1, 2, 3, 3, 4, 4)'$ con $x_3^{(2)} = 2$ e $y^{(2)} = 2$. Para introducir los datos en R haremos:

```
x0<-c(1,1,1,1,1,1) #training sample
x1<-c(0,1,1,2,3,4)
x2<-c(2,2,4,4,2,3)
y<-c(1,2,3,3,4,4)
n<-length(x1)   # Tamaño muestral
d<-data.frame(x0,x1,x2,y)
#Marco (tabla) de datos
View(d)
```

Para crear la matriz de datos de X podemos hacer:

```
k<-2
M<-matrix(1,n,k+1)
M[,2]<-x1
M[,3]<-x2
```

Tecleando `M` podemos ver que efectivamente es una matriz 6×3. Los datos también se pueden introducir directamente en M.

Para calcular la solución óptima usando (2.4) haremos:

```
#Solución óptima
A<-t(M)% *% M
B<-solve(A)
A%*%B
theta<-B%*%t(M)%*% y
theta
```

obteniendo $\hat{\theta} = (0.8129032, 0.7032258, 0.2580645)'$. De esta forma, la función óptima para predecir Y sería

$$h_{\hat{\theta}}(x_1, x_2) = 0.8129032 + 0.7032258 x_1 + 0.2580645 x_2.$$

En R se puede definir con

```
h<-function(z,x) sum(z*x)
```

donde `x` será un vector de longitud 3. Por ejemplo, si queremos predecir Y para $X_1 = X_2 = 1$, haremos

```
x<-c(1,1,1)
h(theta,x)
```

obteniendo $\hat{Y} = 1.774194$.

Para comprobar cómo funcionan estas predicciones en la muestra haremos:

```
ya<-1:n
error<-1:n
for (i in 1:n) {
   ya[i]<-h(M[i,])
   error[i]<- ya[i]-y[i]
}
d<-data.frame(x0,x1,x2,y,ya,error)
View(d)
```

obteniendo la tabla siguiente:

i	X_0	X_1	X_2	Y	$\hat{Y}$	Error $\hat{Y} - Y$
1	1	0	2	1	1.329032	0.32903226
2	1	1	2	2	2.032258	0.03225806
3	1	1	4	3	2.548387	−0.45161290
4	1	2	4	3	3.251613	0.25161290
5	1	3	2	4	3.438710	−0.56129032
6	1	4	3	4	4.400000	0.40000000

Las relaciones entre $\hat{Y}$ e Y y los errores pueden verse en las gráficas de la Figura 2.13. En la primera se incluye su recta de regresión ($y = x$, rojo) y la estimación para $X_1 = 1$ y $X_2 = 1$ (azul). En el segundo podemos estudiar si estos errores son pequeños y aleatorios, determinando los individuos donde la aproximación funciona peor y se podrían usar para detectar posibles valores atípicos (outliers). En este caso, el error mayor se obtiene en el individuo 5 pero son valores similares. El código para obtenerlas es:

```
#Gráficas
plot(ya,y,xlab='h(x)',ylim=c(0,5),xlim=c(0,5))
abline(0,1,col='red')
s<-seq(0,1.5,0.1)
text(1.774194,s,'|',col='blue')
s<-seq(-0.1,1.7,0.1)
text(s,1.774194,'-',col='blue')
text(ya[5]+0.3,y[5]+0.1,'5',cex=0.9,col='blue')
```

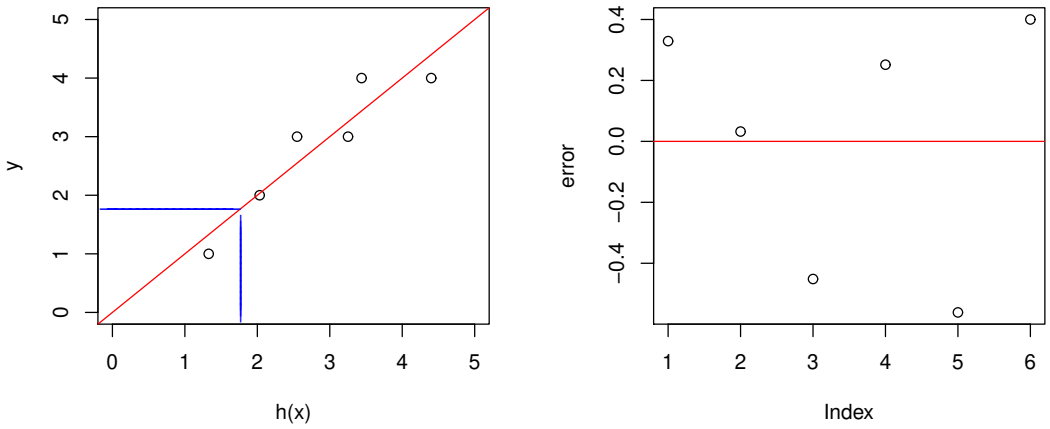


Figura 2.13: Relaciones entre las estimaciones $\hat{Y} = h_\theta(X)$ y los valores exactos de Y (izquierda) y errores $h_\theta(X) - Y$ (derecha).

```
plot(error)
abline(0,0,col='red')
```

El error cuadrático medio (MSE) se calcula con `mean(error^2)`, obteniendo: MSE=0.1419355. Este error será menor del que obtendremos cuando apliquemos este procedimiento a otros datos (ya que se basa en minimizar este error precisamente en esos datos). Para obtener una mejor estimación (no sesgada) podemos usar técnicas de validación cruzada eliminando el individuo en el que se quiere obtener una estimación del cálculo de los coeficientes (muestras pequeñas) o guardando una parte de la muestra para chequear su funcionamiento (muestras grandes). Debe coincidir con el doble de la función J en $\hat{\theta}$ que se puede calcular como:

```
J<-function(z) sum((M%*%z-y)^2)/(2*n)
J(theta)
2*J(theta)
```

Para obtener la superficie de regresión (plano en este caso) de forma automática en R haremos

```
data<-data.frame(y,x1,x2)
lm(data)
```

Note que se obtiene la misma solución que antes y que R toma de forma automática la primera variable como variable respuesta (por eso creamos ese nuevo marco de datos con T en la primera

columna). Alternativamente podemos teclear: `lm(y∼x1+x2)`. Para predecir nuevos valores de Y para, por ejemplo $X_1 = 1$ y $X_2 = 2$, haremos:

```
nuevos<-data.frame(x1=1,x2=2)
reg<-lm(data)
predict(reg,nuevos)
```

obteniendo

$$\hat{Y} = 0.8129 + 1 * 0.7032 + 2 * 0.2581 = 2.032258.$$

Se procederá de forma similar para obtener predicciones sobre un conjunto (marco) de datos. Si queremos dar las predicciones sobre los valores de la muestra de entrenamiento basta teclear `predict(reg)`. Estas estimaciones no son necesarias (ya que tenemos los valores de Y) y serán más precisas que las que obtendremos con valores nuevos (ya que h se ha calculado minimizando esos errores). Sí que se pueden usar para analizar cómo serán esos datos (si están por encima o debajo de lo esperado) y calcular los errores (residuos) utilizados en la gráfica de la Figura 2.13.

Para medir el ajuste podemos usar el coeficiente de correlación múltiple que, por la Proposición 2.1, coincidirá con la correlación entre Y e $\hat{Y}$. Con `corr(y,ya)^2` obtenemos

$$\hat{\rho}^2_{3|1,2} = 0.8753737$$

es decir, $h(x)$ explica un 87.53 % de la variabilidad de Y. Obviamente, esta correlación siempre será mayor que si solo usamos X_1 o X_2. En estos casos obtendríamos 0.8288931 y 0.1017662, respectivamente. Claramente, si tenemos que quedarnos solo con una variable, la mejor es la primera.

Para obtener las correlaciones y los residuos $Y - \hat{Y}$ de forma automática podemos hacer:

```
LM<-lm(data)
summary.lm(LM)
```

obteniendo Multiple R-squared: 0.8754, lo que coincide con el valor obtenido anteriormente (hay otro valor denominado ajustado que incluye una penalización por usar más variables).

La matrices de cuasi-covarianzas y correlaciones muestrales entre las variables originales se pueden calcular con:

```
cov(data.frame(x1,x2,y))
cor(data.frame(x1,x2,y))
```

obteniendo:

$\hat{V}$	X_1	X_2	Y
X_1	2.1666667	0.1666667	1.5666667
X_2	0.1666667	0.9666667	0.3666667
Y	1.5666667	0.3666667	1.3666667

y

$\hat{R}$	X_1	X_2	Y
X_1	1.0000000	0.1151634	0.9104356
X_2	0.1151634	1.0000000	0.3190081
Y	0.9104356	0.3190081	1.0000000

Recordemos que el coeficiente de correlación también se podía obtener como

$$\rho^2 = Corr^2(X,Y) = \rho^2_{k+1(1,\ldots,k)} = \frac{v'_{1,2}V_X^{-1}v_{1,2}}{\sigma^2_Y}.$$

Sustituyendo esos valores por sus estimaciones en estos datos obtenemos:

$$r^2 = \frac{\hat{v}'_{1,2}\hat{V}_X^{-1}\hat{v}_{1,2}}{\hat{\sigma}^2_Y} = 0.8753737.$$

El código en R para este cálculo es el siguiente (note que da igual si utilizamos varianzas-covarianzas o cuasivarianzas-covarianzas):

```
V<-cov(data.frame(x1,x2,y))
VX<-cov(data.frame(x1,x2))
VXI<-solve(VX)
v12<-matrix(NA,1,2)
v12[1,]<-V[3,1:2]
v12%*%VXI%*%t(v12)/var(y)
```

El plano de regresión también se puede calcular de este moda usando esa correlación máxima se obtiene con $\alpha' X$ siendo $\alpha = \lambda V_X^{-1}v_{1,2}$ con $\lambda \neq 0$. Tomando $\lambda = 1$ y sustituyendo los valores teóricos por sus estimaciones con

```
v12%*%VXI%*%t(v12)
```

obtenemos $\hat{\alpha}' = (0.7032258, 0.2580645)$ que son proporcionales (iguales en este ejemplo) a $\hat{\theta} = (0.7032258, 0.2580645)$. El término independiente $\hat{\theta}_1 = 0.8129$ se puede obtener usando regresión lineal univariante con

```
a<-VXI%*%t(v12)
Z<-a[1]*x1+a[2]*x2
lm(data.frame(y,Z))
```

Por último mostramos cómo se puede programar en R el algoritmo GD (con los datos anteriores).

```
# Multivariate Linear Regression with GD:
k<-2
M<-matrix(1,n,k+1)
```

```
M[,2]<-x1
M[,3]<-x2
n<-length(x1)
h<-function(theta,x) sum(theta*x)
J<-function(theta,M) 0.5*sum((M%*%theta- y)^2)/n
m<-10 # Número de interaciones
alfa<-0.1 # learning rate
Z<-matrix(NA,m,k+1)
s<-matrix(NA,n,k+1)
Z[1,]<-c(1,1,1) #valores iniciales
c<-k+1
for (j in 1:m) {
   for (i in 1:n) {s[i,]<-(h(Z[j,],M[i,])-y[i])*M[i,]}
   for (i in 1:c) {Z[j+1,i]<-Z[j,i]-alfa*(1/n)*sum(s[,i])}
}
```

Se puede programar de forma más sencilla (y más eficiente en cuanto a tiempo de ejecución) en forma matricial con

```
z<-c(1,1,1)
alpha<-0.1
m<-9
J2<-1:(m+1)
hv<-1:n
Z2<-matrix(NA,m+1,k+1)
J2[1]<-J(z,M)
Z2[1,]<-z
for (i in 1:m) {
   hv<-M%*%z
   z<-z-(alpha/n)*t(M)%*%(hv-y)
   J2[i+1]<-J(z,M)
   Z2[i+1,]<-z
}
View(data.frame(Z2,J2))
```

Se puede simplificar más no guardando los valores de z y $J(z)$. Después de $m = 10$ iteraciones

con $\alpha = 0.1$ y valor inicial $\theta = (1, 1, 1)$ se obtienen los valores siguientes:

m	$\hat{\theta}_0$	$\hat{\theta}_1$	$\hat{\theta}_2$	$J(\hat{\theta})$
1	1.0000000	1.0000000	1.0000000	4.41666667
2	0.7166667	0.4166667	0.1333333	0.63164352
3	0.8141667	0.6488889	0.4236111	0.15212943
4	0.7770972	0.5884398	0.3060000	0.08912125
5	0.7881402	0.6287448	0.3350212	0.07917442
6	0.7824670	0.6307229	0.3137822	0.07641776
7	0.7830161	0.6440466	0.3118567	0.07495118
8	0.7816132	0.6514127	0.3043705	0.07393387
9	0.7811212	0.6592228	0.2999661	0.07318649
10	0.7804946	0.6654369	0.2954262	0.07263175

Con $m = 50$ iteraciones obtenemos $\hat{\theta} = (0.7852326, 0.7047374, 0.2661078)$. Recordemos que la solución exacta era $\hat{\theta} = (0.8129, 0.7032, 0.2581)$, con $J(\hat{\theta}) = 0.07096774$. La convergencia se puede ver en la Figura 2.14. Observamos que la convergencia es muy rápida aunque se ralentiza cuando nos aceramos al óptimo. El código para obtener este gráfico es el siguiente:

```
Jm<-1:m
for (i in 1:m) Jm[i]<-J(Z[i,],M)
min<-J(c(0.8129, 0.7032,0.2581),M)
plot(Jm[1:10],col='blue',xlab='m',ylab='Jm')
abline(h=min)
```

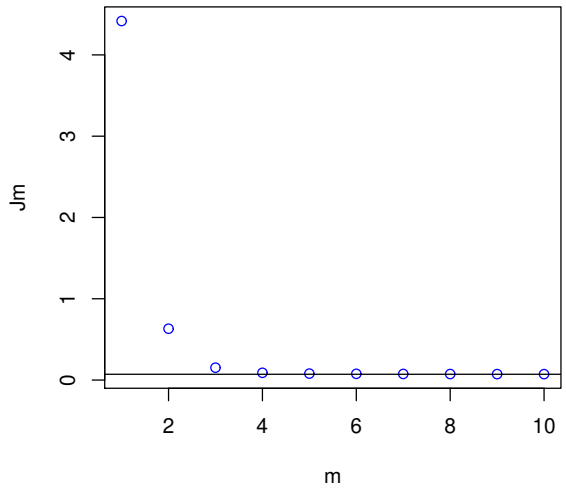


Figura 2.14: Convergencia del algoritmo gradiente descendiente para regresión lineal multivariante con valores iniciales $\theta_0 = \theta_1 = \theta_2 = 1$, $\alpha = 0.1$ y $m = 10$ iteraciones.

2.4. Extensiones del modelo de regresión múltiple

2.4.1. Planteamiento

El modelo estudiado en la sección anterior se puede usar para añadir variables a nuestro modelo inicial (univariante o multivariante). El modelo más típico es el modelo polinómico que transforma el modelos univariante (X, Y) considerando

$$h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_g x^g$$

donde el entero g representa el grado del polinomio que consideremos más adecuado (posteriormente veremos cómo se puede determinar el mejor g). Para hacer esto basta considerar el modelo de regresión lineal multivariante con $X_1 = X$, $X_2 = X^2$, $\ldots$, $X_g = X^g$. Por ejemplo, con los datos de la Figura 2.15, podemos considerar $g = 2$ para intentar mejorar el modelo lineal ajustado anteriormente.

Usaremos este modelo para mostrar el procedimiento pero se puede trabajar con otros modelos (que consideremos adecuados) de forma similar. Por ejemplo, si tenemos un modelo para predecir

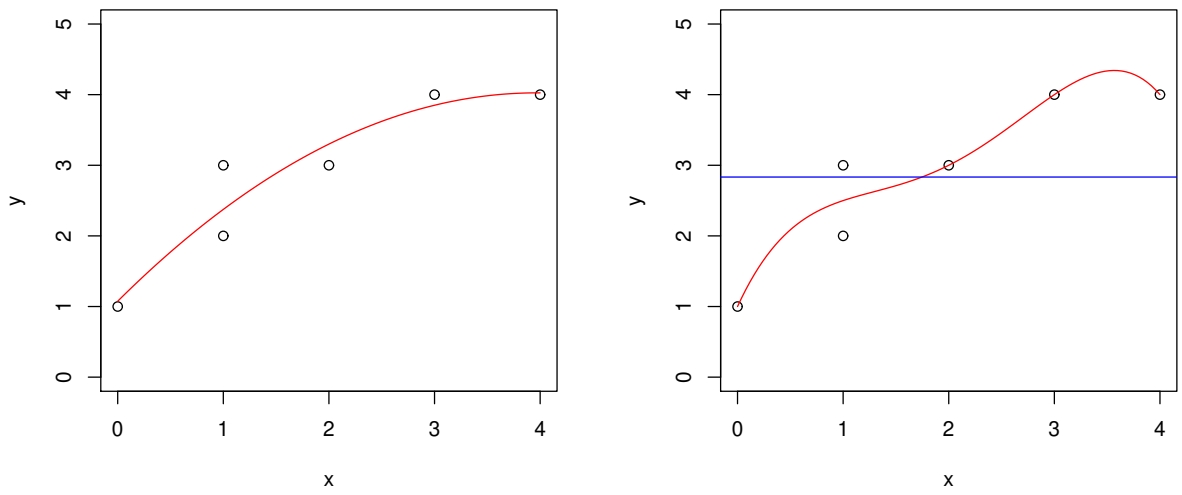


Figura 2.15: Nube de puntos con posible dependencia cuadrática (izquierda) y tabla de datos (derecha) para la muestra en (2.2).

Y a partir de X_1, X_2, y X_3, y no funciona muy bien podemos considerar

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_1 x_2 + \theta_5 x_1 x_3 + \theta_6 x_2 x_3.$$

Las gráficas bidimensionales (nubes de puntos) nos pueden dar una idea de las relaciones que pueden mejorar nuestras estimaciones. De esta forma podemos considerar también $\sqrt{x}$, $\ln x$, $\exp(x)$, etc. o cambiar y por expresiones similares basadas en y.

Volviendo a la regresión polinómica, es evidente que aumentando el grado, disminuirá el valor de J. Si los datos no están alineados (es decir no hay más de un dadto para cada x) y llegamos hasta $g = n$ podemos conseguir un ajuste prefecto (interpolación polinómica). Sin embargo, este ajuste perfecto para los datos de la muestra de entrenamiento, no tiene por qué funcionar mejor cuando lo usemos en otros datos. De hecho, casi siempre funciona peor. Este problema se conoce como **sobreajuste** (overfitting), ver Figura 2.15, derecha, curva roja. Note que no podemos tener un ajuste perfecto porque para $x01$ tenemos dos datos distintos.

También se nos puede dar el caso contrario, denominado **infraajuste** (underfitting). Por ejemplo, con $g = 0$ obtenemos la línea azul en ese gráfico que claramente dará malos resultados para predecir Y.

Si tenemos una muestra de entrenamiento grande, para evitar estos problemas y elegir el número óptimo de variables (y cuáles son las más adecuadas) podemos aplicar el procedimiento siguiente. Separaremos nuestra muestra (de forma aleatoria) en dos grupos. El primer grupo se usará para estimar los coeficientes óptimos para cada g. El segundo grupo se utilizará para calcular los errores

cuadráticos medios para cada g. Obviamente, escogeremos el grado (o grupo de variables) con menor error. De nuevo ese error nos dará una estimación menor del error real que se obtendrá con ese g óptimo. Para hacernos una idea del error real deberemos guardar un tercer grupo de datos para calcular el error en ellos.

El número de datos en cada grupo dependen de muchos factores: tamaño muestral n, número de variables consideradas k, tiempo de programación, etc. Por ejemplo, si tenemos $n = 100$ datos, podríamos dedicar 60 al cálculo de h, 20 para determinar el g óptimo y los otros 20 para estimar el error real en las predicciones futuras. Si nuestra muestra tienen pocos datos, para aplicar este procedimiento deberemos aplicarlo a cada dato eliminándolo del procedimiento para estimar h. Veamos un ejemplo.

2.4.2. Regresión polinómica

Consideramos los datos de la Figura 2.15. Se introducen en R con:

```
# Training sample
x<-c(0,1,1,2,3,4)
y<-c(1,2,3,3,4,4)
n<-length(x) #tamaño muestral
```

Lo primero que debemos hacer es dibujar la nube de puntos con `plot(y,x)` para estudiar sus relaciones. Claramente se observa que un modelo lineal o cuadrático nos puede dar un resultado bueno.

Aplicamos regresión polinómica con $g = 0, 1, 2, 3, 4$ obteniendo las funciones siguientes:

$$h_0(x) = 2.833333,$$

$$h_1(x) = 1.5077 + 0.7231x,$$

$$h_2(x) = 1.0750 + 1.4875x - 0.1875x^2,$$

$$h_3(x) = 1.04839 + 1.62097x - 0.28226x^2 + 0.01613x^3,$$

$$h_4(x) = 1.000 + 3.250x - 2.625x^2 + 1.000x^3 - 0.125x^4.$$

Por ejemplo, los coeficientes para $g = 2$ se pueden obtener con

```
d<-data.frame(y,x,x^2)
lm(d)
```

La gráficas de h_2, h_0 y h_4 pueden verse en la Figura 2.15, (izquierda, derecha azul, derecha roja). Obviamente, los errores cuadráticos medios en estas funciones son decrecientes en g obteniéndose:

$$1.13888889, 0.19487180, 0.10833333, 0.10752688, 0.08333333,$$

respectivamente. Para tener una mejor aproximación del error real, al calcular el error para el dato i, lo eliminaremos del cálculo de coeficientes. De esta forma, para el primer dato obtenemos los errores siguientes:

$$4.84, 0.94204152, 0.21301775, 9, 9$$

y los MSE siguientes (con los 6 datos):

$$1.64, 0.54059, 0.2588195, 3.406709, 3.46875$$

para $g = 0, 1, 2, 3, 4$, respectivamente. La gráfica con estos errores puede verse en la Figura 2.16. Claramente, el mejor resultado se obtiene con $g = 2$, aunque el resultado para $g = 1$ es similar. Por lo tanto, es mejor utilizar h_2 para predecir futuros valores de Y. Su gráfica puede verse en la Figura 2.15, izquierda.

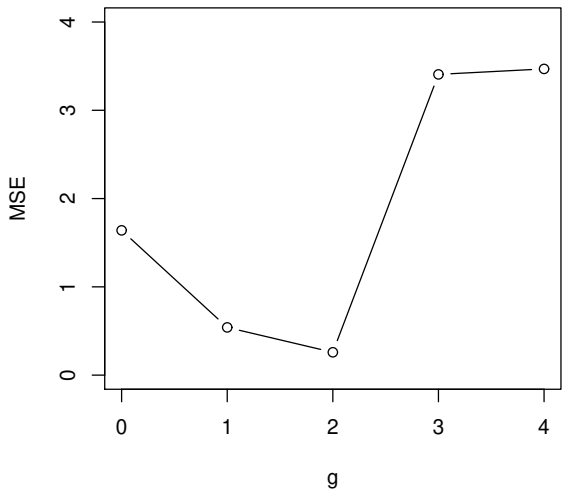


Figura 2.16: MSE para una regresión polinómica con de grado g.

2.5. Regresión cuantílica

2.5.1. Modelo teórico

Mostraremos nuestro modelo en el caso bivariante aunque se puede extender fácilmente al caso multivariante. Así, supondremo que tenmos dos v.a. X e Y y que se quiere predecir Y cuando se

conoce el valor de X. Para ello usaremos la mediana de la distribución condicionada de $(Y|X = x)$. Los cuantiles de esa distribución nos darán intervalos (bandas) de confianza para esas predicciones.

Supondremos que (X, Y) tiene una distribución absolutamente continua

$$F(x, y) = \Pr(X \leq x, Y \leq y).$$

Su función de densidad se obtendrá como

$$f(x, y) = \partial_{1,2} F(x, y)$$

y la de X como

$$f_1(x) = \int_{-\infty}^{\infty} f(x, y) dy.$$

Entonces la densidad condicionada de $(Y|X = x)$ se obtiene como

$$f_{2|1}(y|x) = \frac{f(x, y)}{f_1(x)}$$

para x tales que $f_(x) > 0$. Finalmente, la función de distribución condicionada se obtiene como

$$F_{2|1}(y|x) = \int_{-\infty}^{y} f_{2|1}(z|x) dz = \int_{-\infty}^{y} \frac{\partial_{1,2} F(x, z)}{f_1(x)} dz = \frac{\partial_1 F(x, y)}{f_1(x)} - \lim_{z \to -\infty} \frac{\partial_1 F(x, z)}{f_1(x)}.$$

En muchos modelos este límite es cero y, en ese caso, la distribución condicionada vale:

$$F_{2|1}(y|x) = \frac{\partial_1 F(x, y)}{f_1(x)}.$$

Para estimar Y usaremos la mediana de esta distribución que se puede obtener resolviendo

$$F_{2|1}(y|x) = 0.5$$

para cada x dado. Si esa función es estrictamente creciente y $F_{2|1}^{-1}$ es su función inversa en $(0, 1)$, la **curva de regresión cuantílica o mediana** será:

$$m(x) = F_{2|1}^{-1}(0.5|x). \tag{2.5}$$

Se puede demostrar que esta curva es la que minimiza el error absoluto medio (MAE en inglés)

$$MAE = E(|m(x) - Y|).$$

De forma análoga, las curva cuantílicas se pueden usar para dar intervalos de confianza para estas predicciones. Se definen como en (2.5) como

$$m_q(x) = F_{2|1}^{-1}(q|x) \tag{2.6}$$

para $0 < q < 1$. Por ejemplo, podemos obtener un intervalo (banda) centrado con un 50 % de confianza como

$$I_{50} = [m_{0.25}(x), m_{0.75}(x)]$$

y uno con un 90 % de confianza como

$$I_{90} = [m_{0.05}(x), m_{0.95}(x)].$$

Note que $\Pr(Y \in I_{50}) = 0.5$ y $\Pr(Y \in I_{90}) = 0.9$.

Si (X, Y) tiene una distribución normal, como comentamos anteriormente, la distribución condicionada también es normal $(Y|X = x) \sim N_1(\bar{\mu}, \bar{\sigma}^2)$, con

$$\bar{\mu} = E(Y|X = x) = \mu_Y + \frac{\sigma_{1,2}}{\sigma_{1,1}}(x - \mu_X)$$

y

$$\bar{\sigma}^2 = Var(Y|X = x) = \sigma_{2,2} - \frac{\sigma_{1,2}^2}{\sigma_{1,1}}.$$

Como además en el modelo normal la media y la mediana coinciden se tendrá

$$m(x) = E(Y|X = x) = \mu_Y + \frac{\sigma_{1,2}}{\sigma_{1,1}}(x - \mu_X),$$

es decir, la recta de regresión, la curva de regresión y la curva de regresión mediana coinciden y podemos usar (2.6) para dar bandas de confianza para esas predicciones.

La función condicionada inversa también sirve para obtener (simular) muestras de (X, Y) con el método de la transformada inversa.Se basa en la siguiente propiedad: Si U es un v.a. uniforme en $(0, 1)$ y F es una función de distribución, entonces $F^{-1}(U)$ tienen distribución F. Este método es el que usa R para generar datos de una distribución dada. El algoritmo para obtener muestras bivariantes sería el siguiente:

Método de la tranformada inversa bivariante
Paso 1: Generar U uniforme en $(0, 1)$.
Paso 2: Calcular $X = F_1^{-1}(U)$.
Paso 3: Generar V uniforme en $(0, 1)$.
Paso 4: Calcular $Y = F_{2|1}^{-1}(V|X)$.
Paso 5: Repetir pasos 1-4 n veces.
Veamos un ejemplo.

Ejemplo 2.2. *Supongamos que el peso de los hombres en un país Y (en Kg.) y su altura X (en cm.) siguen una distribución normal bivariante $(X, Y) \sim N_2(\mu, V)$ de media $\mu = (178, 77)$ y matriz de covarianzas*

$$V = \begin{pmatrix} 64 & 32 \\ 32 & 25 \end{pmatrix}.$$

Entonces

$$m(x) = \mu_Y + \frac{\sigma_{1,2}}{\sigma_{1,1}}(x - \mu_X) = 77 + \frac{32}{64}(x - 178)$$

y

$$\tilde{\sigma}^2 = \sigma_{2,2} - \frac{\sigma_{1,2}^2}{\sigma_{1,1}} = 25 - \frac{32^2}{64} = 9$$

.

El código en R para generar 100 *datos de este modelo es el siguiente:*

```
#Método de la transformada inversa
s<-3
m<-function(x) 77+32*(x-178)/64
FI<-function(q,x) qnorm(q,m(x),s)
n<-100
x<-1:n
y<-1:n
set.seed(201)
for (i in 1:n) {
   x[i]<-rnorm(1,178,8)
   y[i]<-FI(runif(1),x[i])
}
mean(x) #178.3829
mean(y) #77.21136
plot(x,y,xlab="X",ylab="Y",pch=20)
```

Para obtener otras muestras basta cambiar (o borrar) la orden `set.seed(201)`*. Los puntos pueden verse en la Figura 2.17 junto con las densidades condicionadas para* $x = 174, 178, 182$ *(rojo, negro azul).*

Para añadir la gráfica de m *en la nube de puntos hacemos:*

```
curve(m(x),add=T,col='red')
```

y para añadir los límites de la región de confianza hacemos:

```
curve(FI(0.25,x),add=T,col='blue')
curve(FI(0.75,x),add=T,col='blue')
curve(FI(0.05,x),add=T,col='blue',lty=2)
curve(FI(0.95,x),add=T,col='blue',lty=2)
```

La gráfica puede verse en la Figura 2.17, derecha. Podemos calcular cuantos puntos de nuestra muestras simulada caen en esas regiones con

```
sum( (y>FI(0.25,x))*(y<FI(0.75,x)))
sum( (y>FI(0.05,x))*(y<FI(0.95,x)))
```

obteniéndose 54 *puntos en* I_{50} *y* 93 *en* I_{90} *(más o menos como esperábamos). Con otras muestras se obtendrán otros valores (similares).*

Estas bandas sirven para predecir el peso de una persona a partir de su altura. Por ejemplo, para el primer individuo que tiene una altura de $x[1] = 180.2885$ *cm. predecimos un peso de* $m(180.2885) = 78.14424$ *Kg. con unos márgenes de* $[73.20968, 83.0788]$ *para el* $90\,\%$ *de la población. El peso real de esa persona (simulada) es* $y[1] = 77.1106$*, valor muy cercano a nuestra predicción y dentro de esos límites. En temas posteriores veremos cómo se pueden obtener otras regiones de confianza para* $(X.Y)$ *usando las curvas de nivel de la función de densidad* f*.*

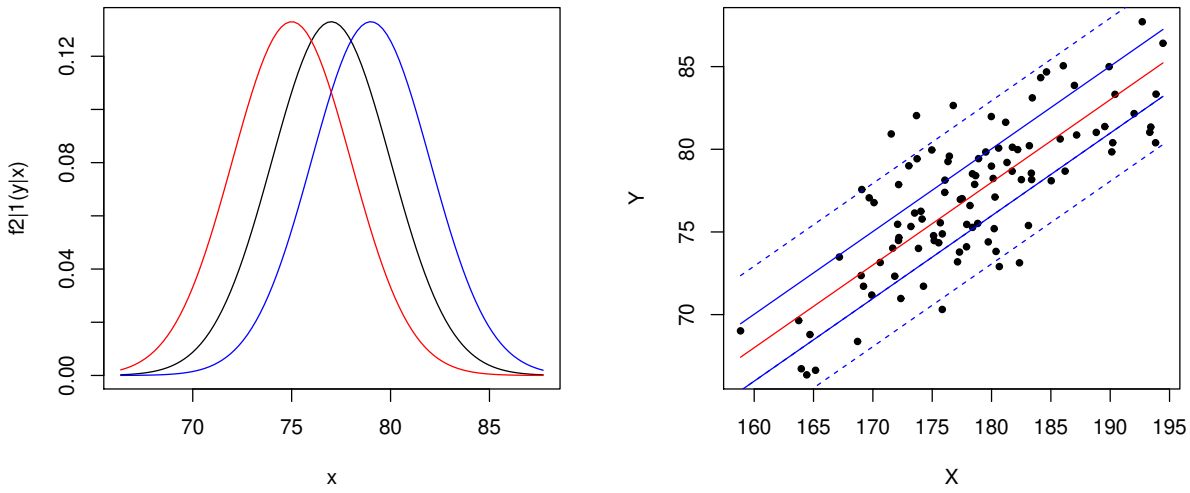


Figura 2.17: Densidades condiciondas (izquierda) y regresión cuantílica (derecha) para la normal del Ejemplo 2.2.

Sin embargo, en la mayoría de los casos, la curva de regresión, la recta de regresión y la curva de regresión mediana no coinciden. Vemos un ejemplo.

Ejemplo 2.3. *Supongamos que* (X, Y) *tienen función de distribución conjunta*

$$C(x, y) = \frac{xy}{x + y - xy}$$

para $x, y \in (0, 1)$ *(cópula de Clayton). Sus distribuciones marginales son*

$$F_1(x) = F_2(x) = C(x, 1) = C(1, x) = x$$

para $x \in (0, 1)$*, es decir son uniformes en el intervalo* $(0, 1)$ *(por eso es una cópula). Por lo tanto*

$E(X) = E(Y) = 0.5$ *y su distribución condicionada vale*

$$F_{2|1}(y|x) = \frac{y^2}{(x+y-xy)^2}$$

para $y \in (0,1)$ *ya que* $f_1(x) = 1$ *para* $x \in (0,1)$. *Su densidad condicionada (y la conjunta) es*

$$f_{2|1}(y|x) = F'_{2|1}(y|x) = \frac{2xy}{(x+y-xy)^3}$$

para $x, y \in (0,1)$. *Sus gráficas para* $x = 0.25, 0.5, 0.75$ *(roja,negra,azul) se pueden ver en la Figura 2.18, izquierda. La curva de regresión cuantílica es*

$$m(x) = F^{-1}_{2|1}(0.5|x).$$

Para calcular la inversa de $F_{2|1}(y|x)$ *para* $0 < q < 1$ *hacemos*

$$\frac{y^2}{(x+y-xy)^2} = q$$

lo que da

$$\frac{x+y-xy}{y} = q^{-1/2}$$

es decir

$$x = (q^{-1/2} + x - 1)y$$

con lo que

$$F^{-1}_{2|1}(q|x) = \frac{x}{q^{-1/2} + x - 1}$$

y

$$m(x) = \frac{x}{0.5^{-1/2} + x - 1}$$

para $x \in (0,1)$. *Su gráfica puede verse en la Figura 2.18, derecha, junto con los intervalos de confianza al* $50\,\%$ *y al* $90\,\%$ *y* 100 *datos generados al azar mediante el método de la transformada inversa. Si contamos los datos en esos intervalos obtenemos* 59 *y* 94 *(un poquito por encima de lo esperado). Nuestra predicción para el valor* Y *del primer dato sería*

$$m(x[1]) = m(0.6125842) = 0.5965967$$

con intervalo de confianza al $90\,\%$ *de* $[0.1499697, 0.9593175]$. *El valor real es* $y[1] = 0.2972452$ *que está dentro de ese intervalo pero lejos de la predicción (porque la variabilidad es muy grande). Esta es una de las ventajas de la regresión cuantílica ya que nos permite medir (y ver) las fiabilidades (o márgenes) de nuestras predicciones.*

Evidentemente, la curva de regresión cuantílica no coincidirá con la recta de regresión (porque no es una recta). Tampoco coincide con la curva de regresión se obtiene como

$$h(x) = \int_0^1 y f_{2|1}(y|x)dy = \int_0^1 \frac{2xy^2}{(x+y-xy)^3}dy$$

para $x \in (0,1)$.

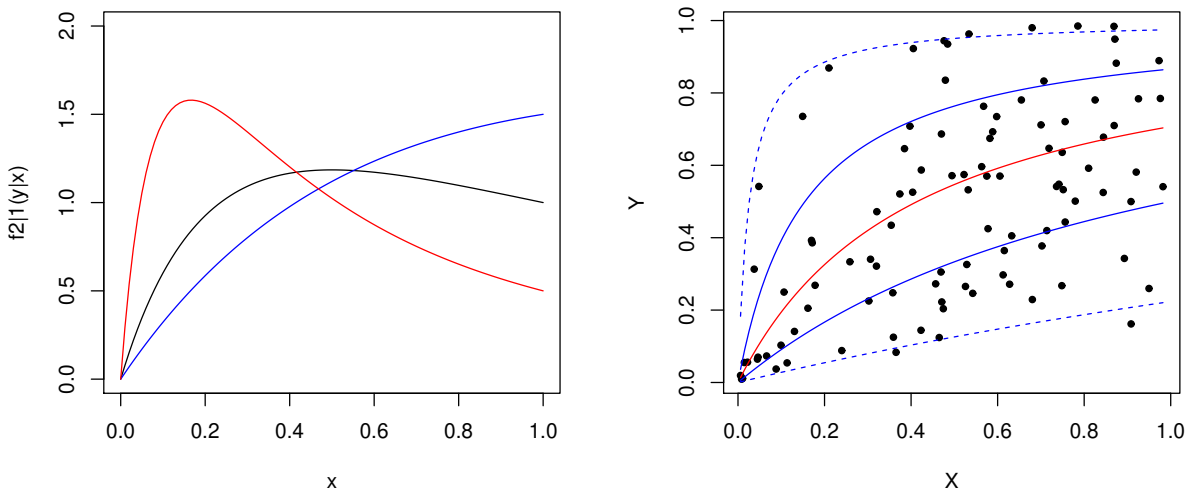


Figura 2.18: Densidades condiciondas (izquierda) y regresión cuantílica (derecha) para la distribución (cópula) de Clayton del Ejemplo 2.3.

2.5.2. Inferencia y predicción

En la práctica las curvas teóricas tendrán que ser estimadas. Para ello dispondremos de una muestra (training sample) en la que conocemos los valores de X y de Y, que representaremos como $(X_1, Y_1), \ldots, (X_n, Y_n)$. En este caso la función a minimizar será el MAE empírico:

$$J^* = \frac{1}{n} \sum_{i=1}^{n} |m(X_i) - Y_i|.$$

Para resolver este problema tenemos que suponer una forma específica para m (por ejemplo, una recta) o asumir un modelo paramétrico para la distribución conjunta de (X, Y) (e.g. podemos asumir que son normales), estimar los parámetros y aplicar las fórmulas obtenidas en la sección anterior con esas estimaciones. Existen modelos no paramétricos que, por ejemplo, estiman las medianas de $(Y|X = x)$ tomando los valores más cercanos a x (si n es muy grande).

Comenzaremos viendo el modelo paramétrico suponiendo que (X, Y) tienen una distribución normal $N_2(\mu, V)$. Obviamente, las medias se estimarán mediante las medias muestrales $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ e $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$. Para estimar las varianzas y covarianzas usaremos

$$S_{1,1} = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2,$$

$$S_{2,2} = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2,$$

y

$$S_{1,2} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y}).$$

Veamos un ejemplo.

Ejemplo 2.4. *Consideraremos la muestra obtenida en el Ejemplo 2.2 a partir de un modelo normal pero supondremos que sus parámetros son desconocidos. Por lo tanto, los estimaremos obteniendo:* $\bar{X} = 178.3829$ *e* $\bar{Y} = 77.21136$*,* $S_{1,1} = 58.31341$*,* $S_{2,2} = 19.97837$ *y* $S_{1,2} = 26.24986$*. Los comandos en R para obtener y guardar esos datos son*

```
xb<-mean(x)
yb<-mean(y)
Sx<-var(x)
Sy<-var(y)
Sxy<-cov(x,y)
```

La matriz de covarianzas también se puede calcular con `cov(data.frame(x,y))`*. La función deregresión cuantílica y la distribución condicionada inversa se definen como en el Ejemplo 2.2 pero usando esas estimaciones:*

```
me<-function(x) yb+Sxy*(x-xb)/Sx
se<-(Sy-Sxy^2/Sx)^0.5
FIe<-function(q,x) qnorm(q,me(x),se)
```

La estimación que se obtiene para la desviación típica residual $\tilde{\sigma}$ *es* $se = 2.856915$ *(su valor real es* 3*). Con esas funciones obtenemos las curvas de la Figura 2.19, derecha (líneas continuas) en la que también mostramos las curvas exactas (líneas discontinuas) y las densidades condicionadas (izquierda). Las rectas verdes y azules determinan los intervalos de confianza muestrales (o estimadas) del* 90 % *y el* 50 %*, respectivamente. Podemos observar que las estimaciones obtenidas con* 100 *datos son bastante buenas. La curva (recta) roja es la curva de regresión cuantílica estimada (que en este caso coincide con la curva y la recta de regresión empíricas). Estas curvas también se pueden usar para estimar los valores de* Y *en un individuo donde conocemos* X*. Por ejemplo, para el primer individuo obtenemos la estimación*

$$\hat{Y} = \hat{m}(x[1]) = 78.06917.$$

La estimación obtenida con el modelo exacto era 78.14424 *y el valor exacto es* $y[1] = 77.1106$*. Los intervalos de confianza empíricos para esa estimación son* $\hat{I}_{50} = [76.14221, 79.99613]$ *y* $\hat{I}_{90} = [73.36997, 82.76838]$*. Ambos contienen al valor exacto. Como ocurre en este caso, estas estimaciones serán mejores sobre los elementos de la muestra (ya que los parámetros del modelo se han ajustado*

para ellos). Como en las secciones anteriores, si queremos conocer el valor real de los errores deberemos testar esas funciones sobre otra muestra (o eliminar el dato a estimar de la estimación de los parámetros). Análogamente, el número de observaciones de la muestra dentro de las bandas de confianza puede ser un poco superior a lo esperado. En este caso se obtienen 48 *datos en* $\hat{I}_{50}$ *y* 94 *en* $\hat{I}_{90}$*. El error absoluto medio en esa muestra se calcula con* `mean(abs(me(x)-y))` *obteniéndose* $MAE = 2.293638$ *(también puede ser un poco mayor del real). También podemos calcular la mediana del error absoluto con* `median(abs(me(x)-y))` *obteniendo:* 1.952542*. Estos errores (residuos) se pueden analizar como en las secciones anteriores.*

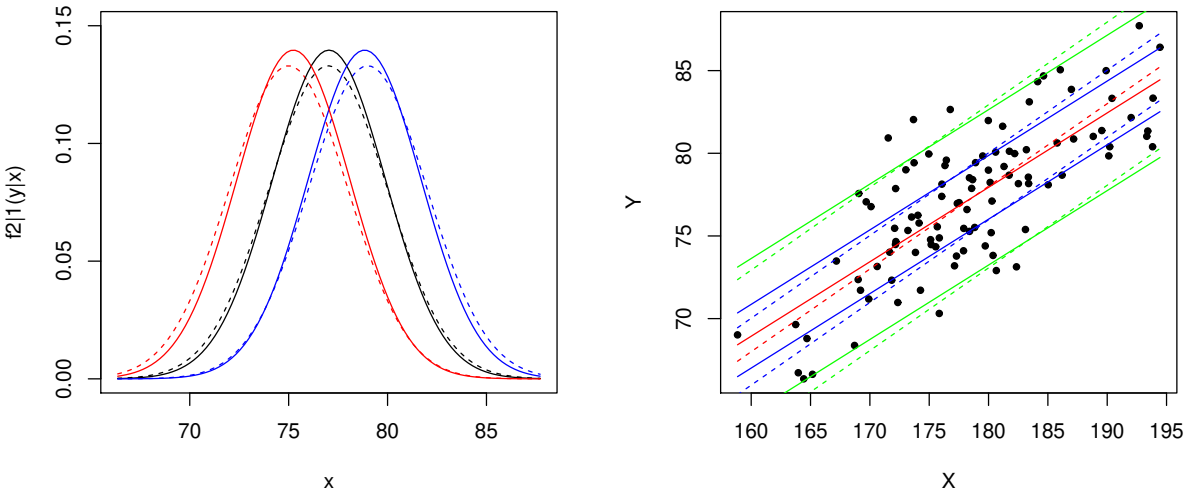


Figura 2.19: Densidades condiciondas (izquierda) y regresión cuantílica (derecha) estimadas para la distribución normal del Ejemplo 2.2.

Como comentamos anteriormente, otra opción es asumir una forma paramétrica para m y tratar de minimizar el MAE. Por ejemplo, en la regresión cuantílica lineal suponemos que $m_\theta(x) = \theta_0 + \theta_1 x$ con $\theta = (\theta_0, \theta_1)$ y trataremos de minimizar

$$J^*(\theta) = \sum_{i=1}^{n} |m(X_i) - Y_i| = \sum_{i=1}^{n} |\theta_0 + \theta_1 X_i - Y_i|.$$

Esta regresión (denominada quantile regression o QR) fue propuesta en por Koenker and Basset (1978) (ver también Koenker, 2005). El planteamiento es similar si X se reemplaza por $X_1, \ldots, X_k$.

Análogamente, la q recta cuantílitica $m_q(x) = a_q + b_q x$ se define como la que minimiza

$$J_q(a,b) = q \sum_{i:Y_i > a+bX_i} (Y_i - a - bX_i) + (1-q) \sum_{i:Y_i < a+bX_i} (a + bX_i - Y_i).$$

Las soluciones a esos problemas se pueden obtener en R con el paquete `quantreg`. Para cargarlo haremos:

`install.packages('quantreg')` para posteriormente marcarlo en la ventana `Pakages`. Si no se dispone de ese paquete tendrá que descargarlo usando `Tools>Install Packages` (indicando un repositiorio CRAN). Este paquete funciona de forma similar al paquete de regresión lineal. Por ejemplo si consideramos la muestra $(1,1)$, $(2,2)$ y $(3,1)$ el problema a resolver es

$$\text{mín}\, J(\theta) = |m_\theta(1) - 1| + |m_\theta(2) - 2| + |m_\theta(3) - 1|.$$

La solución óptima es $m(x) = 1$ con $J(1,0) = 1$. Se puede ver en la Figura 2.20 (línea roja) junto con la recta de regresión (verde) y la recta cuantílica con $q = 0.75$ (azul). la recta cuantílica con $q = 0.25$ coincide con m. El código para obtener estas curvas es:

```
# Regresión quantílica
x<-c(1,2,3)
y<-c(1,2,1)
rq(y∼x)
plot(x,y,xlab="X",ylab="Y",pch=20,xlim=c(0,4),ylim=c(0,3))
abline(rq(y∼x),col='red')
lm(y∼x)
abline(lm(y∼x),col='green')
rq(y∼x,1/4)
rq(y∼x,3/4)
abline(rq(y∼x,3/4),col='blue')
```

En algunos casos, si hay pocos puntos, estos problemas pueden tener diferentes óptimos. Sin embargo, si tenemos muchos puntos de una distribución bivariante continua (diferentes) entonces estas soluciones son únicas. Veamos un ejemplo.

Ejemplo 2.5. *Consideramos de nuevo la muestra del modelo normal obtenida en el Ejemplo 2.2. La recta de regresión mediana estimada es*

$$\hat{m}(x) = 1.2983842 + 0.4251036x.$$

Recordemos que la exacta era

$$m(x) = 77 + 32\frac{x - 178}{64} = -12 + 0.5x.$$

Las gráficas se pueden ver en la Figura 2.20, derecha, junto con la de las rectas cuantílicas para $q = 0.25, 0.75$ (azul) y $q = 0.05, 0.95$ (verde). El códgo para obtener ese rectas es:

```
plot(x,y,xlab='X',ylab='Y',pch=20)
d<-data.frame(y,x)
```

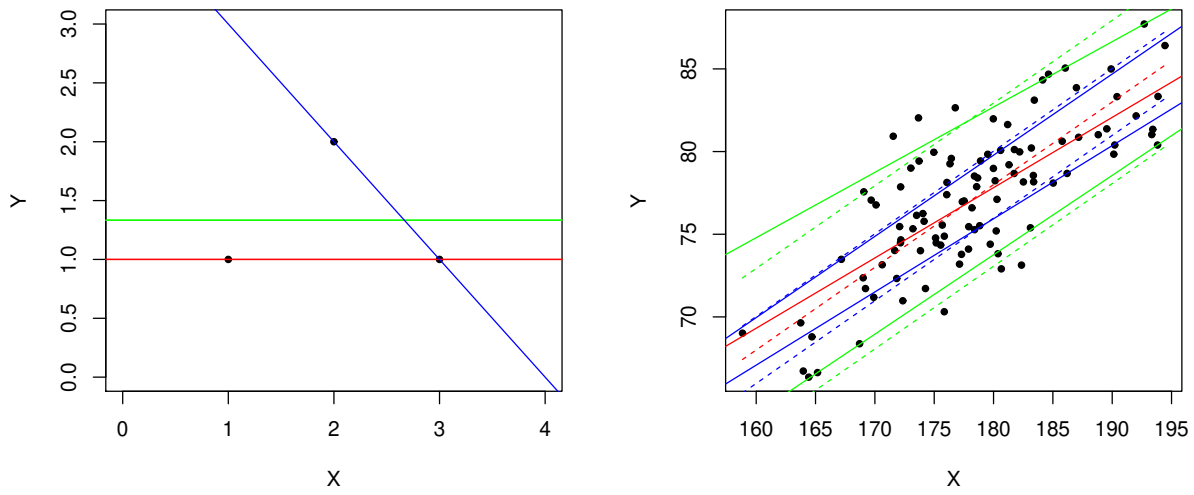


Figura 2.20: Regresión cuantílica lineal con tres puntos (izquierda) y con 100 puntos de la normal considerada en el Ejemplo 2.5.

```
abline(rq(d),col='red')
abline(rq(d,1/4),col='blue')
abline(rq(d,3/4),col='blue')
abline(rq(d,0.95),col='green')
abline(rq(d,0.05),col='green')
```

La estimación que se obtiene para el primer dato es $m(x[1]) = 77.93967$ *siendo el valor exacto* $y[1] = 77.1106$. *Los intervalos de confianza para esta estimación son* $\hat{Y}_{50} = [76.06201, 79.92128]$ *y* $\hat{Y}_{90} = [73.89545, 82.80867]$. *La recta de regresión es* $h(x) = -3.0879 + 0.4502x$ *y la estimación* $h(x[1]) = 78.07798$ *(peor en este caso). El valor óptimo de* J *es*

$$J(1.2983842, 0.4251036) = 228.6394$$

con un error absoluto medio (en la muestra) $MAE = 2.286394$ *Kg. Recordemos que el error real en futuras predicciones puede ser un poco mayor. Como en los ejemplos anteriores también podemos contar cuántos individuos de la muestra están dentro de las bandas de confianza obteniendo* 50 *y* 88, *respectivamente. En la distribución normal, el modelo lineal debe ser el que mejor soluciones de (ya que la curva exacta m es una recta). En otros (Clayton), otras curvas podrían dar mejores soluciones. Como en la regresión, para estudiar estos modelos basta añadir otras variables. Por ejemplo, en este caso, si queremos añadir* x^2 *y* x^3 *en m haremos:*

```
d<-data.frame(y,x,x^2,x^3)
rq(d,3/4)
```

obteniendo

$$m(x) = -528.9478 + 9.007978x - 0.04621452x^2 + 0.00008312606x^3.$$

Los bajos coeficientes en x^2 y en x^3 nos indican que estas variables no son necesarias y pueden dar perores estimaciones en otros datos (por sobreajuste).

2.6. Problemas

1. Calcular la curva de regresión en una normal con medias 1 y 2, varianzas 2 y correlación $-1/2$.
2. Calcular la curva de regresión en un vector (X, Y) con densidad $f(x, y) = c$ para $0 < x < y < 1$ (cero en otro caso). Dibujarla y predecir el valor de Y para $X = 2/3$.
3. Encontrar una densidad bivariante en la que la curva de regresión no sea una recta.
4. Encontrar la recta de regresión para los datos:

   $$(1, 4), (2, 2), (1, 5), (5, 3), (6, 2).$$

   Estimar y para $x = 3$. ¿Será fiable esa aproximación?
5. Programar y aplicar el algoritmo del gradiente descendiente a $J(x) = x^2(x + 5)^2$.
6. Programar y aplicar el algoritmo del gradiente descendiente a una función $J(x)$ con un único mínimo local.
7. Programar y aplicar el algoritmo del gradiente descendiente a los datos del problema 4.
8. Encontrar la recta de regresión para 10 datos inventados por ti. Estimar y para un valor de x. ¿Será fiable esa aproximación? Programar y aplicar el algoritmo del gradiente descendiente a esos.
9. Programar y aplicar el algoritmo del gradiente descendiente a una función $J(x, y)$ con un único mínimo local.
10. Usar los datos de R en el objeto `USArrests` para predecir el número de arrestados por asesinatos por cada 100.000 habitantes (Murder) en un estado a partir de la variable porcentaje de población urbana (UrbanPop). Estimar ese número para un estado con un 85 % de población urbana. ¿Será fiable esta estimación? (razone la respuesta).

11. Usar los datos de R en el objeto `USArrests` para predecir el número de arrestados por asesinatos por cada 100.000 habitantes (Murder) en un estado a partir de las variables: X_1 arrestos por asaltos por cada 100.000 habitantes (Assault), X_2 porcentaje de población urbana (UrbanPop) y X_3 arrestos por violación por cada 100.000 habitantes (Rape). Estimar ese número para un estado con 200 asaltos, un 85 % de población urbana y 40 violaciones. ¿Será fiable esta estimación? (razone la respuesta). Comprobar cómo funcionan las estimaciones en cada elemento de la muestra. ¿Qué variable es la que mejor predice? ¿Qué pareja de variables es la que mejor predice? Calcular las estimaciones en cada caso.

12. Usar los datos de R en el objeto `iris` para predecir la variable longitud del sépalo (`Sepal.Length`) a partir de las otras variables con todas las especies juntas y con las especies por separado. ¿Qué procedimiento funciona mejor?

13. Aplicar el algoritmo GD para regresión lineal multivariante para 4 variables inventadas por ti.

14. Mejorar el algoritmo GD para regresión lineal multivariante usando notación matricial. Aplicarlo a los datos usados en la sección 1.3 y comprobar que se obtienen los mismos resultados.

15. Usar los datos de R en el objeto `iris` para predecir la variable longitud del sépalo (`Sepal.Length`) a partir de la variable anchura del sépalo (`Sepal.Width`) usando regresión polinómica con todas las especies juntas y con las especies por separado. ¿Qué procedimiento funciona mejor? ¿Cuál es el grado óptimo para el polinomio?

16. Comprobar los valores y gráficas del Ejemplo 2.3 y realizar un estudio similar para otra cópula.

17. **(Tarea 1)** Aplicar regresión lineal y/o cuadrática a un conjunto de datos reales aplicando todas las técnicas que consideres oportunas.

3

Regresión logística

La regresión logística es un procedimiento de clasificación supervisado para dos grupos que es la base de las redes neuronales. Para clasificar se usa una función lineal de las variables y se necesitan los valores de estas variables en muestras de individuos de cada grupo. Las técnicas son similares a las usadas en regresión lineal y se pueden usar con muchos datos por medio del algoritmo gradiente-descendiente.

3.1. Modelo teórico

El planteamiento es similar al de regresión lineal múltiple con la única diferencia de que ahora nuestra variable respuesta Y solo toma los valores 0 y 1. Además, estos valores numéricos solo indicarán la pertenencia o no a un determinado grupo (es decir, en realidad no son números). El ejemplo típico es el de determinar si un paciente tiene o no una determinada enfermedad. En este caso el valor 1 suele indicar que sí la tiene y 0 que no.

Como en el tema anterior, para "predecir" Y dispondremos de diversas variables numéricas $X_1, \ldots, X_k$ que resumiremos en una única función

$$h_\theta(x) = g(\theta' x) = g(\theta_0 + \theta_1 x_1 + \cdots + \theta_k x_k),$$

donde el vector columna $\theta = (\theta_0, \ldots, \theta_k)'$ contiene todos los parámetros del modelo (que deberemos determinar para obtener los mejores valores posibles) y $x = (x_0, \ldots, x_k)'$ contiene las variables que podemos medir en nuestros individuos para predecir y. Como en el capítulo anterior, para mejorar la notación hemos incluido una variable artificial X_0 que siempre vale 1. Además, si queremos considerar modelos no lineales basta añadir variables como x_1^2, $x_1 x_2$, etc.

La función g debe transformar esos valores numéricos (lineales) en números entre 0 y 1 que nos indiquen cómo de cerca estará el individuo con medidas x de cada grupo, es decir, es como la "probabilidad" de que el individuo pertenezca al grupo 1. Por lo tanto

$$g : \mathbb{R} \to [0, 1]$$

y $h_\theta(x) \approx \Pr(Y = 1|X = x)$, donde $X = (X_0, \ldots, X_k)'$. De esta forma, la regla de decisión será:

$$h_\theta(x) \geq 0.5 \to \hat{y} = 1;$$

$$h_\theta(x) < 0.5 \to \hat{y} = 0;$$

donde $\hat{y}$ representa el valor que predecimos para Y cuando $X = x$.

Existen diversas opciones para determinar g pero la más popular es la función logística (o sigmoide)

$$g(z) = \frac{1}{1 + \exp(-z)} = \frac{\exp(z)}{1 + \exp(z)}$$

cuya gráfica puede verse en la Figura 3.1, izquierda. Por eso este modelo se denomina *Regresión Logística* (RL o logit model). Podemos observar que es continua, estrictamente creciente y que va de 0 a 1 (es una función de distribución). Por lo tanto transformará nuestro valor lineal $\theta' x \in \mathbb{R}$ en un valor $h_\theta(x) \in [0, 1]$. Además, $g(0) = 0.5$ con lo que la regla de decisión será:

$$\theta' x \geq 0 \to \hat{y} = 1;$$

$$\theta' x < 0 \to \hat{y} = 0.$$

Si la distribución logística se reemplaza por al distribución normal estándar el modelo se denomina regresión *probit*.

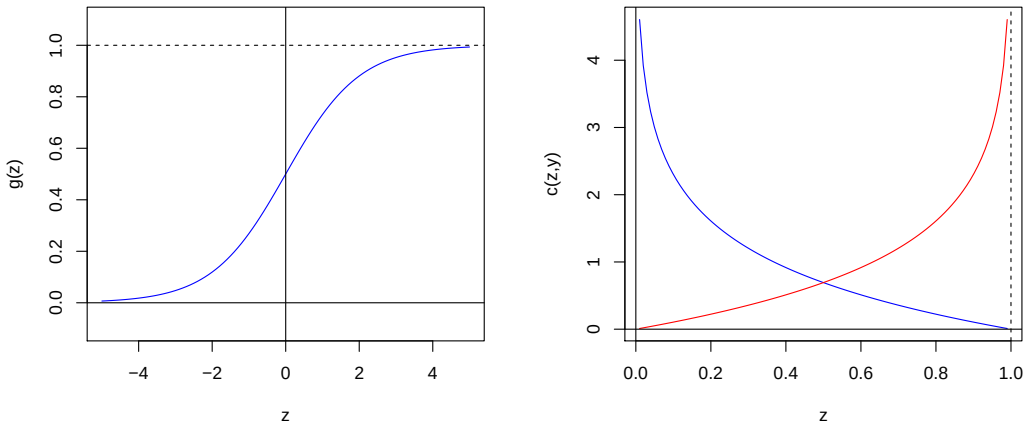


Figura 3.1: Función logística (o sigmoide) $g(z)$ (izquierda) y función de costo $c(z, y)$ asociada (derecha) para $y = 1$ (azul) e $y = 0$ (roja).

El siguiente paso consiste en determinar una función "costo" c que penalice las decisiones erróneas. Para ello, si $z = h_\theta(x)$ usaremos

$$c(z, y) = -\log(z)$$

si $y = 1$ y

$$c(z, y) = -\log(1 - z)$$

si $y = 0$, es decir,

$$c(z, y) = -y \log(z) - (1 - y) \log(1 - z),$$

donde $y \in \{0, 1\}$ (y los logaritmos son neperianos). Las gráficas pueden verse en la Figura 3.1, derecha ($y = 1$ azul, $y = 0$ roja). Note que $c \geq 0$ y que vale cero en $c(1, 1)$ y en $c(0, 0)$. Estos valores no se darán nunca ya que $z = h_\theta(x) = g(\theta' x) \in (0, 1)$, pero si $\theta' x$ es muy grande e $y = 1$, el costo será pequeño. Ocurrirá lo mismo si $\theta' x$ es muy pequeño e $y = 0$. Por contra, si $\theta' x$ es muy grande e $y = 0$, el costo será muy alto. Ocurrirá lo mismo si $\theta' x$ es muy pequeño e $y = 1$.

Una vez determinada la función costo trataremos de minimizar su valor esperado

$$\min_\theta J(\theta) = E(c(h_\theta(X), Y)).$$

Como en el tema anterior, para determinar los valores óptimos de los parámetros, necesitaremos disponer de una muestra (denominada "training sample") de individuos en los que se conozcan tanto los valores de x como los valores de y (aprendizaje supervisado). Calcularemos los costos en los valores muestrales y determinaremos los parámetros que minimizan estos costos (ver sección siguiente).

Si $p = \Pr(Y = 1)$, este modelo es equivalente a suponer que existe una relación lineal entre las variables $X_1, \ldots, X_k$ y la función *log-odd* de p, es decir,

$$\log\left(\frac{p}{1 - p}\right) = \theta' X.$$

Esto es equivalente a suponer que

$$p = \Pr(Y = 1) = \frac{\exp(\theta' X)}{1 + \exp(\theta' X)} = g(\theta' X))$$

como hemos establecido arriba. Si esto es cierto, podremos hablar de "probabilidades" de pertenencia al grupo 1 (o 0 con $1 - p$). Si no lo es, podremos considerar que lo que obtenemos es una aproximación de estas probabilidades (la mejor aproximación posible con nuestros datos y con esas variables). Como hemos mencionado arriba podremos tratar de mejorar estas aproximaciones añadiendo variables pero, como solo las medimos en los valores muestrales, también podemos caer en problemas de sobreajuste obteniendo un ajuste perfecto en los puntos de la muestra pero malas predicciones en otros valores. Para evitar ésto, deberemos usar el método de validación cruzada para testar nuestro modelo en datos que no se han usado para determinar θ.

Cuando haya mas de dos grupos (es decir $Y = 1, 2, \ldots, g$, podemos testar cada grupo frente al resto para, paso a paso llegar a una solución final. Existen distintos órdenes para esos test y deberemos estudiar cuál proporciona mejores resultados en la práctica.

3.2. Inferencia y predicción

3.2.1. Análisis inicial de los datos

Como en técnicas anteriores usaremos un ejemplo sencillo para comprobar cómo funciona nuestro modelo. Supongamos que tenemos dos variables predictoras X_1 y X_2 ($k = 2$) y los datos siguientes:

i	X_1	X_2	Y
1	1	2	0
2	2	1	0
3	3	1	0
4	2	2	0
5	5	1	1
6	5	3	1
7	3	2	0
8	4	3	1
9	4	4	4
10	5	4	1

Lo primero que tenemos que hacer (si es posible) es dibujar estos puntos añadiendo una etiqueta para distinguir los de cada grupo. Existen diversas formas para hacer esto en R. Por ejemplo, podemos añadir una etiqueta en cada dato para indicar el valor de Y. El resultado puede verse el la Figura 3.2, izquierda. El código para hacer esta gráfica es el siguiente:

```
X1<-c(1,2,3,2,5,5,3,4,4,5)
X2<-c(2,1,1,2,1,3,2,3,4,4)
Y<-c(0,0,0,0,1,1,0,1,1,1)
plot(X1,X2,xlab="X1",ylab="X2",pch=20,xlim=c(0,6),ylim=c(0,5))
text(X1+0.2,X2,Y,cex=0.5)
```

Otra opción es usar símbolos diferentes (o colores) para cada grupo. Por ejemplo, tecleando:

```
plot(X1,X2,xlab="X1",ylab="X2",pch=as.integer(Y),xlim=c(0,6),ylim=c(0,5),cex=0.7)
legend('topleft',legend=c('Y=0','Y=1'),pch=0:1,cex=0.7)
```

obtenemos la gráfica de la Figura 3.2, derecha. Hay que tener cuidado de que la caja con las etiquetas no tape ningún dato.

3.2.2. Modelo lineal

En ambas gráficas podemos observar que los dos grupos se pueden separar muy bien con rectas. Por lo tanto nuestro modelo será

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2).$$

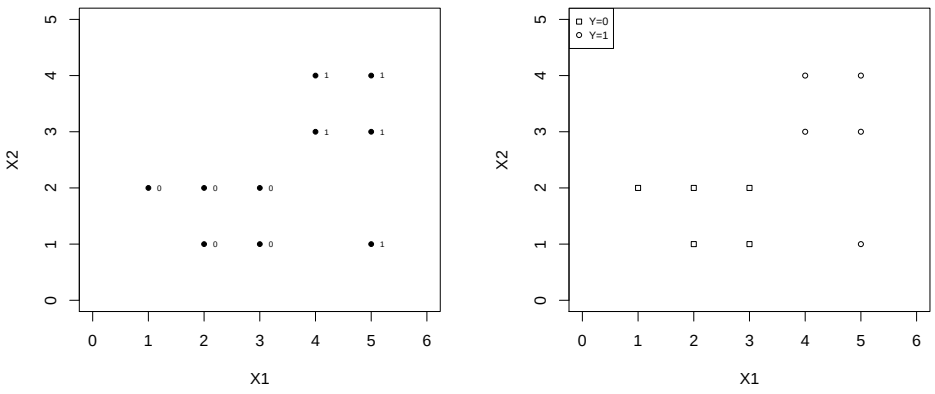


Figura 3.2: Valores muestrales para $y = 1$ e $y = 0$.

Otra forma de analizar los grupos es calcular medidas descriptivas en cada uno de ellos. Por ejemplo podemos calcular las medias en cada grupo con

```
tapply(X1,Y,mean)
tapply(X2,Y,mean)
```

obteniendo $\bar{X}_1 = 2.2$ y $\bar{X}_2 = 1.6$ en el grupo $Y = 0$ y $\bar{X}_1 = 4.6$ y $\bar{X}_2 = 3$ en el $Y = 1$. Estas diferencias también se pueden ver representado X_i frente a Y con

```
plot(X1,Y,ylim=c(-0.5,1.5))
plot(X2,Y,ylim=c(-0.5,1.5))
```

obteniendo la gráficas de la Figura 3.3. Podemos observar cómo la primera variable separa mejor a los grupos que la segunda (en los valores muestrales). De forma similar se pueden representar histogramas o diagramas caja-bigote para comparar las variables en cada grupo (ver sección siguiente).

De esta forma, la función de costo empírica será

$$J(\theta) := \frac{1}{n} \sum_{i=1}^{n} c(h_\theta(x^{(i)}), y^{(i)})$$

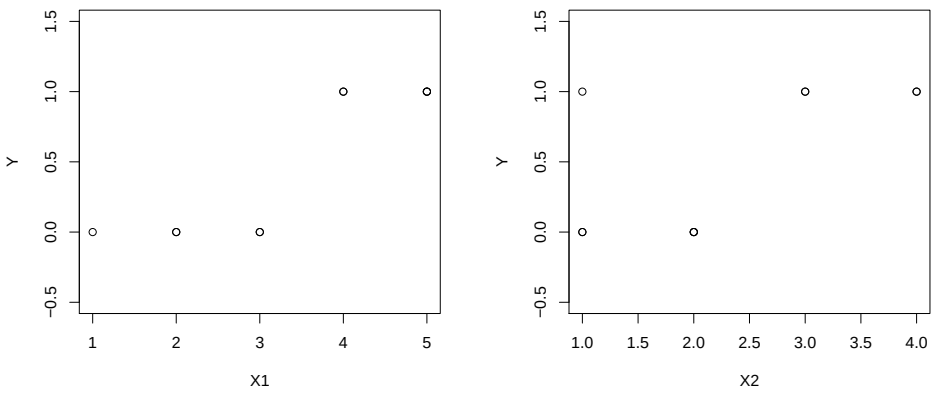


Figura 3.3: Valores muestrales para X_1 (izquierda) y X_2 (derecha).

donde $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$, representan los valores muestrales. Desarrollando c obtenemos

$$\begin{aligned} J(\theta) &= -\frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] \\ &= -\frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)}\log g(\theta' x^{(i)}) + (1-y^{(i)})\log\left(1-g(\theta' x^{(i)})\right)\right]. \end{aligned}$$

Esta función también se puede escribir en forma matricial. Así, si $M = (x_j^{(i)})$ representa la matriz de datos, y es el vector columna con los valores en Y y $h := g(M\theta)$ es el vector columna con los ajustes en cada individuo, entonces

$$J(\theta) = \frac{1}{n}\left[y'\log(h) + (1_n - y)'\log(1_n - h)\right]$$

donde 1_n representa un vector columna de dimensión n. Para calcular esta función en R con los datos anteriores podemos hacer:

```
n<-length(Y)
k<-2
M<-matrix(1,n,k+1)
M[,2]<-X1
M[,3]<-X2
J<-function(theta) -sum(Y*log(g(M%*%theta))+(1-Y)*log(1-g(M%*%theta)))/n
z<-c(0,0,1)
J(z) #0.9454044
```

Además, la hemos calculado para $\theta = (0, 0, 1)$ (es decir, usando solo la variable X_2) obteniendo $J(\theta) = 0.9454044$. Con esta elección todos los puntos se clasifican como $\hat{y} = 1$ (la frontera es la recta $x_2 = 0$). Podemos probar con otros valores y mejorar este resultado con `z<-c(-3.5,1,0)` que da $J(z) = 0.2982264$. Esta elección se representaría con la recta vertical $x_1 = 3.5$ que separa mejor los grupos (ver Figura 3.2).

3.2.3. Algoritmo GD

Como en el capítulo anterior, deberemos "ajustar" el parámetro θ para que J tome el menor valor posible. Para ello podemos usar de nuevo el método del gradiente descendiente (GD). Para ello, usaremos un valor inicial $\theta^{(0)}$ y la siguiente fórmula recursiva para obtener los nuevos valores:

$$\theta_i^{(j+1)} := \theta_i^{(j)} - \alpha \frac{\partial}{\partial \theta_i} J(\theta^{(j)}),$$

simultáneamente para $i = 0, \ldots, k$. La derivada parcial se puede calcular a partir de

$$\frac{\partial}{\partial \theta_i} J(\theta) = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial \theta_i} \left[-y^{(i)} \log g(\theta' x^{(i)}) - (1 - y^{(i)}) \log \left(1 - g(\theta' x^{(i)}) \right) \right].$$

Como $g(z) = (1 + \exp(-z))^{-1}$, tenemos $-\log(g(z)) = \log(1 + \exp(-z))$ y

$$(-\log(g(z)))' = -\frac{\exp(-z)}{1 + \exp(-z)} = \frac{-1}{1 + \exp(z)} = -g(-z) = g(z) - 1.$$

Análogamente

$$-\log(1 - g(z)) = -\log\left(1 - \frac{1}{1 + e^{-z}}\right) = \log\left(\frac{1 + e^{-z}}{e^{-z}}\right) = \log(1 + e^{z})$$

con lo que

$$(-\log(1 - g(z)))' = \frac{e^{z}}{1 + e^{z}} = \frac{1}{1 + e^{-z}} = g(z).$$

De esta forma, obtenemos

$$\begin{aligned}
\frac{\partial}{\partial \theta_i} J(\theta) &= \frac{1}{n} \sum_{i=1}^{n} \left[-y^{(i)} g(-\theta' x^{(i)}) + (1 - y^{(i)}) g(\theta' x^{(i)}) \right] x_j^{(i)} \\
&= \frac{1}{n} \sum_{i=1}^{n} \left[y^{(i)} (-1 + g(\theta' x^{(i)})) + (1 - y^{(i)}) g(\theta' x^{(i)}) \right] x_j^{(i)} \\
&= \frac{1}{n} \sum_{i=1}^{n} \left[g(\theta' x^{(i)}) - y^{(i)} \right] x_j^{(i)}
\end{aligned}$$

que es una expresión similar a la obtenida en regresión lineal.

De esta forma, el algoritmo GD quedaría como

$$\theta_i^{(j+1)} := \theta_i^{(j)} - \alpha \frac{1}{n} \sum_{i=1}^{n} \left[g((\theta^{(j)})' x^{(i)}) - y^{(i)} \right] x_j^{(i)},$$

simultáneamente para $i = 0, \ldots, k$. Como antes, en forma matricial se puede escribir como

$$\theta^{(j+1)} := \theta^{(j)} - \frac{\alpha}{n} M(h - y).$$

Por ejemplo, si tomamos $\theta^{(0)} = (-3.5, 1, 0)$ y $\alpha = 0.1$, el primer paso para este algoritmo se puede programar como

```
z<-c(-3.5,1,0)
alpha<-0.1
h<-g(M%*%z)
z<-z-(alpha/n)*t(M)%*%(h-Y)
z
J(z)
```

De esta forma, obtenemos $\theta^{(1)} = (-3.49893433, 1.02685904, 0.02270574)$ con $J(\theta^{(1)}) = 0.2879659$ que mejora un poco el valor anterior. Tras 500 iteraciones obtenemos $\theta^{(500)} = (-5.7435809, 1.3657064, 0.5798428)$ con $J(\theta^{(500)}) = 0.1639105$. La recta que marca la frontera de esta solución sería

$$-5.7435809 + 1.3657064 x_1 + 0.5798428 x_2 = 0$$

es decir

$$x_2 = 9.90541 - 2.355305 x_1.$$

Para añadir esta recta en la Figura3.2 basta hacer

```
abline(9.90541,-2.355305,col='red')
```

El resultado puede verse en la Figura 3.4, izquierda.

Si queremos predecir el grupo para un nuevo individuo con valores $X_1 = 3.5$ y $X_2 = 2$, podemos definir la función

```
hd<-function(x1,x2) -5.7435809+1.3657064*x1+0.5798428*x2
```

obteniendo $h_{\hat{\theta}}(3.5, 2) = 0.1960771 > 0$ por lo que se clasifica en el grupo $y = 1$. Para añadir este punto a la gráfica usaremos `text(3.5,2,'x')`.

Para comprobar si la convergencia es buena podemos representar $J(\theta^{(j)}$ frente a J, ver Figura 3.4, derecha. En esa gráfica usamos 1000 iteraciones y $\alpha = 0.1$ (como antes) y $\alpha = 1/3$ (para acelerar el algoritmo). Para calcular estas iteraciones podemos aplicar el código siguiente:

```
z<-c(-3.5,1,0)
alpha<-0.1
```

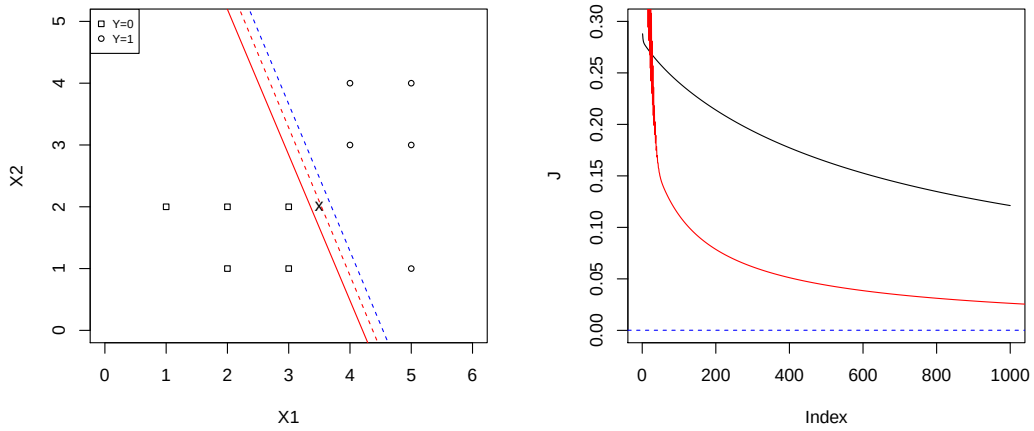


Figura 3.4: Valores muestrales para $y = 1$ e $y = 0$ junto con la recta (rojo) que separa los grupos tras 500 iteraciones del algoritmo GD (izquierda). Costo promedio $J(\theta^{(j)}$ para 1000 iteraciones con $\alpha = 0.1, 0.333$ (negra,roja). Las líneas azules representan los valores óptimos.

```
m<-1000
J1<-1:m
for (i in 1:m) {
   h<-g(M%*%z)
   z<-z-(alpha/n)*t(M)%*%(h-Y)
   J2[i]<-J(z)
}
```

Cambiando `m` y `alpha` podemos ver cómo converge el algoritmo. Observamos que J disminuye más en el segundo caso obteniéndose

$$\theta^{(1000)}_{\alpha=1/3} = (-10.750545, 2.459372, 1.028702)$$

con $J(\theta^{(1000)}_{\alpha=1/3}) = 0.05965094$. En este caso

$$h_{\hat{\theta}}(x_1, x_2) = -10.750545 + 2.459372x_1 + 1.028702x_2$$

y obtenemos $h_{\hat{\theta}}(3.5, 2) = -0.085339$ con lo que se clasificaría en el grupo $y = 0$. La frontera de clasificación sería la recta $x_2 = 10.45059 - 2.390753x_1$ (ver línea discontinua en Figura 3.4, izquierda).

Para medir cómo de fiable es esta clasificación podemos mirar esa gráfica (cuando sea posible) o calcular la "probabilidades a posteriori"

$$\Pr(Y = 1|X_1 = 3.5, X_2 = 2) \approx g(h_{\hat{\theta}}(3.5, 2)) = 0.4786782$$

y

$$\Pr(Y = 0|X_1 = 3.5, X_2 = 2) \approx 1 - g(h_{\hat{\theta}}(3.5, 2)) = 0.5213218.$$

Recordemos que en realidad no estamos seguros de que esos valores sean realmente esas probabilidades. De esta forma observamos que esta clasificación no es muy fiable ya que ese punto está muy cerca de la frontera.

3.2.4. Paquete `rms`

Las técnicas de regresión logística se pueden calcular de forma automática con el comando `lrm` del paquete `rms`. Para instalarlo en `Rstudio`, pinchamos en la pestaña `Packages` de la ventana inferior derecha. En la "lupa" ponemos "rms" y buscamos ese paquete. Si está basta "tiquearlo" para que se instale. Si no tendremos que pinchar en "install" para que lo busque en un repositorio (una vez instalado hay que tiquearlo para que se carge).

Un vez cargado el paquete (y los datos), para hacer una regresión logística basta con teclear:

`lrm(Y∼X1+X2)` De esta forma vemos que la solución óptima es

$$h_{\hat{\theta}}(x_1, x_2) = -47.8914 + 10.5524x_1 + 4.4340x_2$$

obteniendo $J(\hat{\theta}) = 0.0001226743$ (lo que mejora un poco nuestras aproximaciones con el algoritmo GD). La frontera que se obtiene con este valor aparece en la Figura 3.4, izquierda (líneas discontinuas azules). Observamos que hay pequeñas diferencias.

Para predecir Y en nuevos valores haremos:

```
LRM<-lrm(Y∼X1+X2)
predict(LRM,c(3.5,2))
```

En este caso obtenemos

$$h_{\hat{\theta}}(3.5, 2) = -2.089962$$

con probabilidades a posteriori

$$\Pr(Y = 1|X_1 = 3.5, X_2 = 2) \approx g(h_{\hat{\theta}}(3.5, 2)) = 0.1100763$$

y

$$\Pr(Y = 0|X_1 = 3.5, X_2 = 2) \approx 1 - g(h_{\hat{\theta}}(3.5, 2)) = 0.8899237.$$

Con esta solución sí que podemos afirmar con una mayor confianza que ese individuo debe estar en el grupo $Y = 0$.

Si son muchos datos en los que queremos predecir el grupo podemos usar

```
d<-data.frame(X1,X2)
predict(LRM,d)
```

Así observamos que todos los individuos de la muestra se clasificarían bien (tal y como esperábamos).

3.2.5. Modelo cuadrático

En otros casos, un modelo lineal puede no ser suficiente para separar los grupos. Por ejemplo con los datos:

i	X_1	X_2	Y
1	4	2	1
2	2	1	0
3	3	1	1
4	2	2	0
5	5	1	0
6	5	3	0
7	3	2	1
8	4	3	1
9	4	4	0
10	5	4	0

obtenemos la gráfica de la Figura 3.5. Claramente, un modelo lineal no es suficiente para separar estos grupos. Esta situación es bastante común cuando los valores centrales representan unidades bien fabricadas (de un modelo multivariante F) mientras que las periféricas representan unidades con algún tipo de defecto (no pertenecientes a F). También se puede utilizar para detectar valores atípicos (outliers).

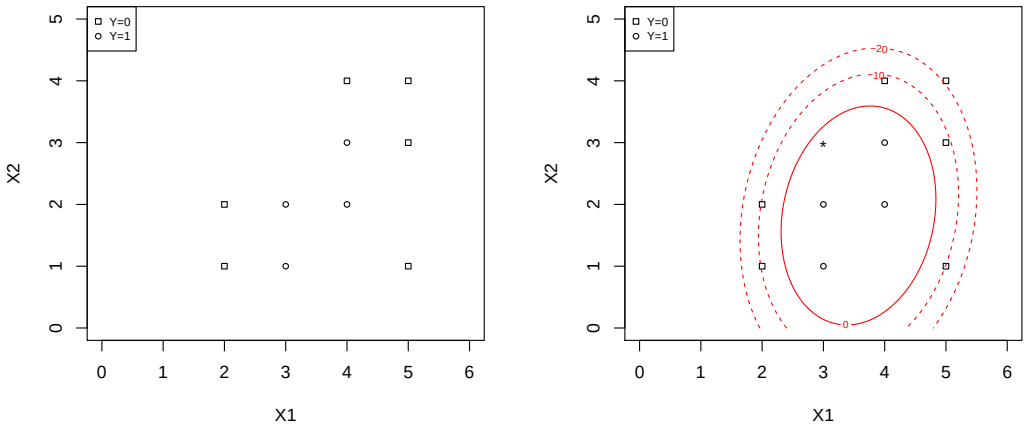


Figura 3.5: Valores muestrales para $y = 1$ e $y = 0$ junto con la frontera (línea continua roja) en una regresión logística cuadrática.

Para separar estos grupos debemos tomar un modelo cuadrático

$$h_\theta(x_1, x_2) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + +\theta_5 x_1 x_2.$$

Para hacer esto, basta considerar el modelo lineal añadiendo las variables $x_3 = x_1^2$, $x_4 = x_2^2$, $x_5 = x_1 x_2$. Así, tecleando:

```
X1<-c(4,2,3,2,5,5,3,4,4,5)
X2<-c(2,1,1,2,1,3,2,3,4,4)
Y<-c(1,0,1,0,0,0,1,1,0,0)
X3<-X1*X1
X4<-X2*X2
X5<-X1*X2
LRM<-lrm(Y∼X1+X2+X3+X4+X5)
LRM
```

obtenemos

$$h_{\hat\theta}(x_1, x_2) = -109.92 + 64.58 x_1 + 10.40 x_2 - 9.56 x_1^2 - 4.86 x_2^2 + 2.05 x_1 x_2.$$

De esta forma, si queremos predecir el grupo para un objeto con medidas $X_1 = X_2 = 3$, usaremos

```
predict(LRM,c(3,3,9,9,9))
```

obteniendo $h_{\hat\theta}(3, 3) = 3.640506$ con lo que se clasificaría como $Y = 1$ con probabilidades a posteriori

$$\Pr(Y = 1 | X_1 = 3, X_2 = 3) \approx g(h_{\hat\theta}(3, 3)) = 0.9744318$$

y

$$\Pr(Y = 0 | X_1 = 3, X_2 = 3) \approx 1 - g(h_{\hat\theta}(3, 3)) = 0.02556818.$$

Para representar este punto y la frontera debemos "correr" el código siguiente:

```
hc<-function(x1,x2)  -109.92 +64.58*x1+10.40*x2-9.56*x1*x1 -4.86*x2*x2+ 2.05*x1*x2
x<-seq(0,6,length=1000)
y<-seq(0,6,length=1000)
z<-outer(x,y,hc)
plot(X1,X2,pch=as.integer(Y),xlim=c(0,6),ylim=c(0,5),cex=0.7)
legend('topleft',legend=c('Y=0','Y=1'),pch=0:1,cex=0.7)
contour(x,y,z,levels=0,add=TRUE,col='red')
contour(x,y,z,levels=c(-10,-20),add=TRUE,col='red',lty=2)
text(3,3,'*')
```

El resultado puede verse en la Figura 3.5, derecha. En ella también hemos incluido las curvas de nivel $h = -10$ y $h = -20$ (líneas rojas discontinuas) que nos indicarán cómo de lejos estarán

los puntos en esas regiones del grupo central. Las probabilidades de pertenencia al grupo 1 en esas curvas son $g(-10) = 0.00004539787$ y $g(-20) = 0.000000002061154$, respectivamente.

3.3. Ejemplo

Por último mostramos cómo aplicar esta técnica a un conjunto de datos reales con tres grupos. Para ello usaremos los datos en el objeto `iris` que contienen cuatro medidas de 150 flores iris en tres especies diferentes (50 de cada especie).

Para ver los datos y calcular las medias y otras medidas en la primira variable por grupos haremos:

```
d<-iris
View(d)
tapply(d$Sepal.Length,d$Species,mean)
tapply(d$Sepal.Length,d$Species,summary)
```

Haciendo lo mismo con las otras variable obtenemos las medias siguientes:

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

Para representarlas por parejas y por grupos haremos:

```
plot(d$Sepal.Length,d$Sepal.Width,pch=as.integer(d$Specie),cex=0.5)
legend('topleft',legend=c('setosa','versicolor','virginica'),
   pch=1:3,cex=0.3)
```

Haciendo lo mismo con las otras dos variables obtenemos las gráficas de las Figura 3.6. En la primera podemos observar que el grupo 'setosa' se separa bien de los otros grupos teniendo una media más baja en la primera variable y más grande en la segunda. Los otros grupos aparecen más mezclados en esa gráfica. Sin embargo, en la segunda, las especies aparecen mucho más separadas con medias claramente ordenadas (los pétalos más pequeños corresponden a 'setosa', y los más grandes a 'virginica').

Otra forma sencilla de analizar los datos es realizar los gráficos caja-bigote por especies. Pueden verse en las Figuras 3.7 y 3.8. El primero se obtiene con:

```
boxplot(d$Sepal.Length~d$Species)
```

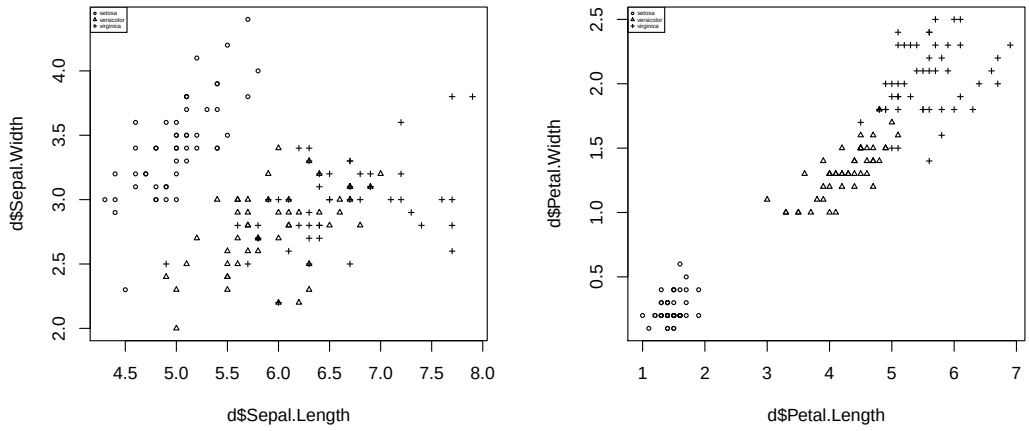


Figura 3.6: Valores muestrales para los datos de `iris` para las tres especies.

En los dos primeros gráficos (sépalos) observamos que los "bigotes' se solapan lo que nos permitirá separar bien al menos a un 75 % de los valores muestrales (usando solo una variable). En las dos últimas (pétalos) podemos ver que las flores de la primera especie (setosa) tiene los pétalos mucho más pequeños y menos dispersos por lo que se separa perfectamente de las otras dos que también se pueden separar bastante bien aunque los "bigotes" se solapan un poco.

En capítulos posteriores veremos cómo realizar una gráfico conjunto para estas variables usando las componentes principales.

Como hay tres grupos, para separarlos tenemos que analizarlos por parejas. Por ejemplo, para separar las dos primeras especies haremos:

```
X1<-d$Sepal.Length[1:100]
X2<-d$Sepal.Width[1:100]
X3<-d$Petal.Length[1:100]
X4<-d$Petal.Width[1:100]
Y<-d$Species[1:100]
LRM12<-lrm(Y∼X1+X2+X3+X4)
LRM12
```

El índice (función) que separa estas especies es

$$I_{12}(x1, x2, x3, x4) = 1.6445 - 2.5292 * x1 - 3.6365 * x2 + 5.8680 * x3 + 9.3695 * x4.$$

Si este índice es menor que cero, la flor se clasificará como de la especie "setosa" y si es positivo en la "versicolor". Para calcularlo en la muestra y dibujarlo podemos hacer:

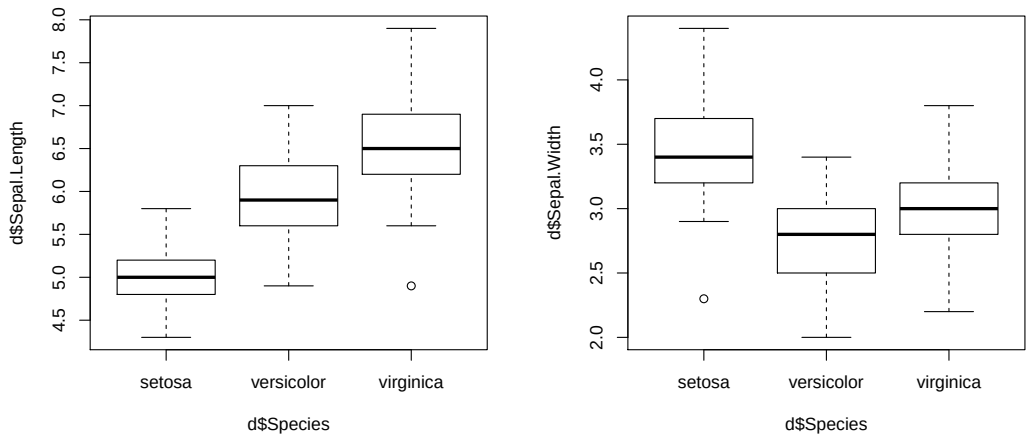


Figura 3.7: Gráficos caja-bigote para los datos de `iris` separados por especies.

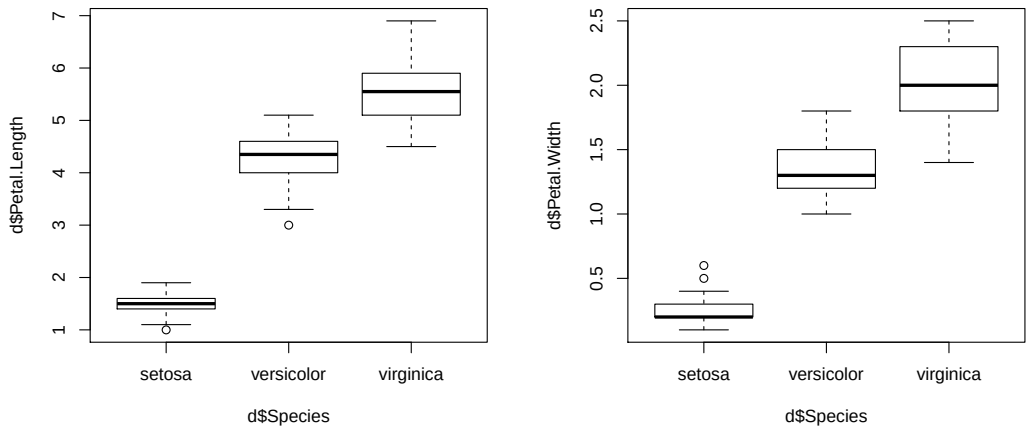


Figura 3.8: Gráficos caja-bigote para los datos de `iris` separados por especies.

```
F12<-function(x1,x2,x3,x4)  1.6445 -2.5292*x1-3.6365*x2+5.8680*x3+9.3695*x4
I12<-F1(d[,1],d[,2],d[,3],d[,4])
plot(I12,pch=as.integer(d$Specie),cex=0.5)
legend('topleft',legend=c('setosa','versicolor','virginica'),pch=1:3,cex=0.3)
```

```
abline(h=0,col='red')
boxplot(I12~d$Species)
abline(h=0,col='red')
```

El resultado puede verse en la Figura 3.9. Observamos que la separación es perfecta en los puntos de la muestra. También vemos que si aplicamos este índice a la tercera especie, todos sus individuos se clasificarían como de la especie segunda.

Si queremos predecir el grupo para una flor con medidas $5, 3, 3, 1$ haremos

```
predict(LRM12,c(5,3,3,1))
```

Se obtiene $I_{1,2} = 5.062433 > 0$ por lo que, entre estas dos especies, el punto está más cerca de las de la especie "vericolor". Nos faltaría calcular el índice para estas dos especies y comprobar en cual se clasifica. Hemos incluido este dato en esas figuras (punto azul) y observamos que efectivamente está más cerca de las de la segunda especie, cerca de la frontera con la especie primera. No parece que pertenezca a la tercera.

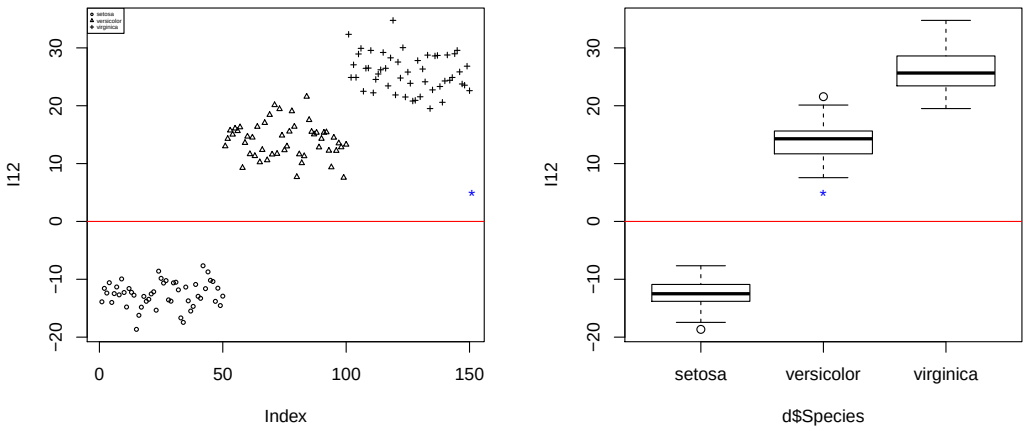


Figura 3.9: Índice para separar las dos primeras especies para los datos de `iris`.

Como estos datos se han usado para calcular el índice, este porcentaje de acierto podría ser un poquito mayor del que se obtendrá para una flor que no esté en esa muestra. Para estimar este porcentaje de acierto podemos guardar una parte de la muestra que no se usará en el cálculo del índice y usar esa muestra para testar su funcionamiento. Como los datos parecen obtenidos al azar (no se han ordenado), por ejemplo, podemos hacer:

```
X1<-d$Sepal.Length[11:90]
```

```
X2<-d$Sepal.Width[11:90]
X3<-d$Petal.Length[11:90]
X4<-d$Petal.Width[11:90]
Y<-d$Species[11:90]
LRM12XV<-lrm(Y∼X1+X2+X3+X4)
LRM12XV
I12XV<-function(x1,x2,x3,x4)  -3.1610-1.5998*x1-3.6784*x2+7.0801*x3+4.7190*x4
Z1<-d$Sepal.Length[c(1:10,91:100)]
Z2<-d$Sepal.Width[c(1:10,91:100)]
Z3<-d$Petal.Length[c(1:10,91:100)]
Z4<-d$Petal.Width[c(1:10,91:100)]
I12XV(Z1,Z2,Z3,Z4)
plot(I12XV(Z1,Z2,Z3,Z4))
abline(h=0)
```

Observamos que aunque el índice es distinto, todos los elementos de la muestra para validación cruzada (XV) se clasifican bien. Obviamente, esperamos que el primer índice dé mejores resultados (ya que se calcula con más valores muestrales).

Completamos este estudio calculando los índices que servirán para separar las especies 1 y 3 y las 2 y 3 obteniendo

$$I_{1,3}(x_1, x_2, x_3, x_4) = -4.9285 - 1.7473x_1 - 1.6775x_2 + 4.6475x_3 + 3.9767x_4$$

$$I_{2,3}(x_1, x_2, x_3, x_4) = -42.6356 - 2.4652x_1 - 6.6806x_2 + 9.4290x_3 + 18.2854x_4.$$

El resultado en los valores de la muestra puede verse en la Figura 3.10 donde también podemos ver los valores que se obtienen para esa flor que queremos clasificar que, de forma clara, se clasifica como de la especie "versicolor" ya que $I_{2,3} = -28.431$. También podemos observar que los grupos 2 y 3 están más mezclados y que dos flores de la muestra de esas especies se clasificarían mal. Para detectarlas y contarlas podemos usar

```
I23<0
sum(I23[101:150]<0)
sum(I23[51:100]>0)
```

observando que las flores mal clasificadas son la 84 y la 134. El porcentaje de acierto estimado sería $98/100 = 0.98$ pero, como antes, advertimos que el resultado en los valores muestrales puede ser un poco mejor que el que obtendremos en otras flores.

Si queremos representar estos índices en una solo gráfica, observamos que en este ejemplo, la primera especie se para muy bien de las otras dos, siendo la especie tercera la más lejana de la primera. teniendo esto en cuenta bastaría representar de forma conjunta los índices $I_{1,2}$ e $I_{2,3}$ representados en la Figura 3.11, izquierda. En la parte derecha representamos $I_{1,3}$ e $I_{2,3}$ que, en este caso, creemos que no será necesario. De hecho, si observamos la flor que queremos clasificar (punto azul) observamos que si primero comparamos los grupos 1 y 3, se clasificaría en el 1, y para

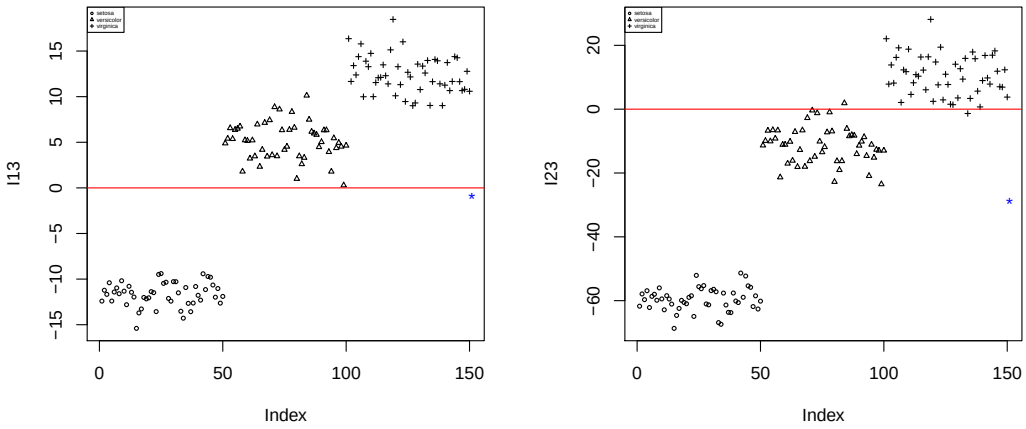


Figura 3.10: Índice para separar las especies 1 y 3 (izquierda) y las 2 y 3 (derecha) para los datos de `iris`.

poder clasificarla bien deberíamos usar la gráfica de la izquierda (con la derecha se clasificaría en 1).

El código para la primera gráfica es

```
especies<-c('setosa','versicolor','virginica')
plot(I12,I23,pch=as.integer(d$Specie),cex=0.5)
legend('topleft',legend=especies,pch=1:3,cex=0.3)
abline(h=0,col='red')
abline(v=0,col='red')
text(F12(5,3,3,1),F23(5,3,3,1),'*',col='blue')
```

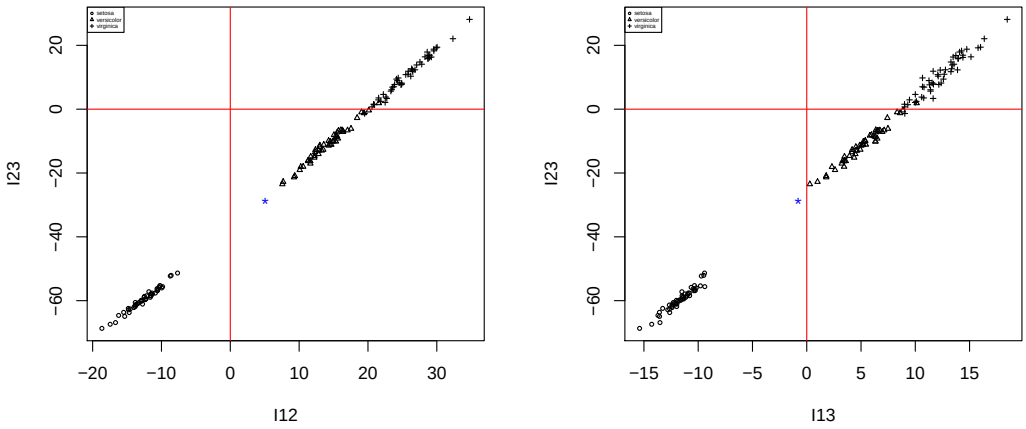


Figura 3.11: Índice para separar las especies 1 y 2 y 2 y 3 (izquierda) y las 1 y 3 y las 2 y 3 (derecha) para los datos de `iris`.

3.4. Problemas

1. Aplicar regresión logística lineal a una muestra de tamaño 10 con dos variables y dos grupos. Calcular el índice de separación y realizar las gráficas pertinentes. Comprobar los resultados usando al algoritmo GD.

2. Aplicar regresión logística cuadráticas a una muestra de tamaño 10 con dos variables y dos grupos. Calcular el índice de separación y realizar las gráficas pertinentes. Comprobar los resultados usando al algoritmo GD.

3. Aplicar regresión logística lineal a una muestra de tamaño 10 con dos variables y tres grupos. Calcular el índice de separación y realizar las gráficas pertinentes.

4. Estudiar la fiabilidad de los procedimientos de clasificación para separar las especies 2 y 3 de los datos `iris`.

5. Estudiar la fiabilidad del ejercicio anterior se puede mejorar usando regresión logística cuadrática para separar las especies 2 y 3 de los datos `iris`.

6. **(Tarea 2)** Aplicar regresión logística a un conjunto de datos reales aplicando todas las técnicas que consideres oportunas.

4

Análisis de componentes principales

En este capítulo mostramos cómo calcular las componentes principales asociadas a un conjunto de variables aleatorias tanto desde el punto de vista teórico como empíricamente. El Análisis de las Componentes Principales (ACP o PCA en inglés) permitirá resumir la información contenida en las variables, mostrar sus principales relaciones y analizar las características de los individuos de la población usando sus valores (puntuaciones) en las componentes principales.

4.1. Introducción

El primer investigador que se percató de la necesidad de eliminar la información redundante en un conjunto de variables aleatorias fue Galton (Francis, Inglaterra, 1822-1911) quién criticó un intento de identificar criminales a partir de 12 medidas corporales alegando que varias de las medidas tomadas estarían altamente correlacionadas. Posteriormente, en 1901, un colaborador de Pearson (Karl, Inglaterra 1857-1936), McDonald hizo un estudio similar sobre 7 variables y 3000 criminales, publicando los resultados en una matriz de correlaciones, con la idea de encontrar algún índice que resumiera la información contenida en los datos. Pearson estaba convencido que los "índices" ideales coincidirían con los ejes del elipsoide de concentración. Pearson probó que el plano formado por estos ejes y que pasaba por la media era el que minimizaba la suma de las distancias al cuadrado con cada punto original. Finalmente, en 1933, fue Hotelling (Harold, USA 1895-1973) el que encontró un algoritmo para calcular dichos ejes lo que, en esencia, requería la obtención de los valores propios de una matriz simétrica y definida positiva. También debemos destacar las aportaciones de Rao (Calyampudi Radhakrishna, India, 1920). Con la llegada de los ordenadores la resolución de este problema para muchas variables (matrices de gran dimensión) ha supuesto uno de los principales problemas de la Programación Matemática y del Big Data.

Para presentar el problema usaremos varios ejemplos.

Ejemplo 4.1. *En el primer ejemplo consideramos un conjunto de datos denominado* `LifeCycleSavings` *incluido en el programa R. Los ficheros de datos incluidos en R se pueden ver con* `data()`*. Para ver los datos de ese fichero haremos:*

```
LifeCycleSavings
```

y para guardarlos en `d`

```
d<-LifeCycleSavings
```

El fichero contiene 5 *variables medidas en* 50 *países diferentes. Los primeros datos se pueden ver en la Tabla 4.1.*

Tabla 4.1: Primeros datos del fichero `LifeCycleSavings`.

	sr	pop15	pop75	dpi	ddpi
Australia	11.43	29.35	2.87	2329.68	2.87
Austria	12.07	23.32	4.41	1507.99	3.93
Belgium	13.17	23.80	4.43	2108.47	3.82
Bolivia	5.75	41.89	1.67	189.13	0.22
Brazil	12.88	42.19	0.83	728.47	4.56

La información proporcionada en R sobre datos se puede ver con:

```
help(LifeCycleSavings)
```

donde se indica que:
sr: incremento de los ahorros personales 1960-1970.
pop15: % población menor de 15 años.
pop75: % población mayor de 75.
dpi: ingresos per-capita.
ddpi: crecimiento del dpi 1960-1970.

Para estudiar las relaciones entre las variables podemos hacer

```
plot(d)
```

La gráfica se puede ver en la Figura 4.1.

Podemos calcular las matrices de covarianza y correlación con `cov(d)` *y* `cor(d)`*, respectivamente. Las correlaciones se pueden ver la Tabla 4.2. Se aprecia que existen variables con correlaciones (lineales) positivas, negativas y casi nulas. Estas relaciones se verán reflejadas en las componentes principales que calcularemos posteriormente.*

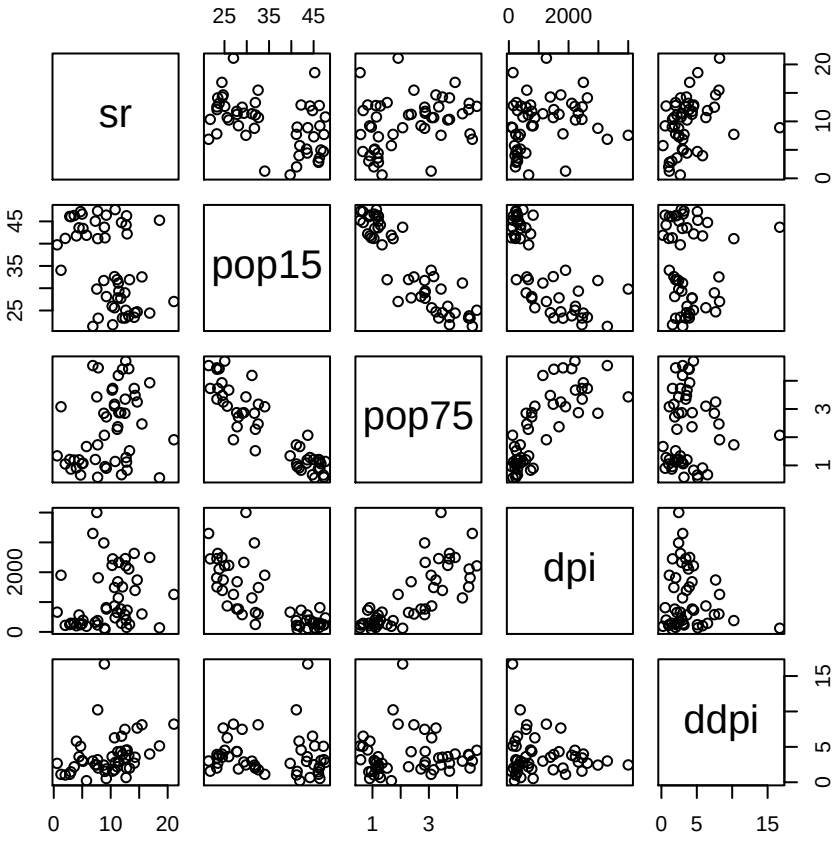


Figura 4.1: Gráficos bidimensionales para todas las variables del fichero de R `LifeCycleSavings`.

Tabla 4.2: Correlaciones entre las 5 variables del fichero `LifeCycleSavings`.

	sr	pop15	pop75	dpi	ddpi
sr	1.0000000	-0.45553809	0.31652112	0.2203589	0.30478716
pop15	-0.4555381	1.00000000	-0.90847871	-0.7561881	-0.04782569
pop75	0.3165211	-0.90847871	1.00000000	0.7869995	0.02532138
dpi	0.2203589	-0.75618810	0.78699951	1.0000000	-0.12948552
ddpi	0.3047872	-0.04782569	0.02532138	-0.1294855	1.00000000

Ejemplo 4.2. *En este ejemplo analizaremos el objeto* `d` *del fichero* `nota.rda` *(Aula virtual*[1]*) que contiene las notas (sobre 100) de 88 alumnos de matemáticas en una universidad americana (Fuente: Rencher, 1995). Los primeros datos se pueden ver en la Tabla 4.3.*

Tabla 4.3: Primeros datos del fichero `nota.rda`.

	Mecanica	Vectores	Algebra	Analisis	Estadist
1	77	82	67	67	81
2	63	78	80	70	81
3	75	73	71	66	81
4	55	72	63	70	68
5	63	63	65	70	63
6	53	61	72	64	73
7	51	67	65	65	68
8	59	70	68	62	56
9	62	60	58	62	70
10	64	72	60	62	45

Para estudiar las relaciones entre las variables podemos hacer `plot(d)`*. La gráfica se puede ver en la Figura 4.2.*

Podemos calcular las matrices de covarianza y correlación con `cov(d)` *y* `cor(d)`*, respectivamente. Las correlaciones se pueden ver la Tabla 4.4. Se aprecia que todas las variables tienen correlaciones (lineales) positivas. En la matriz de covarianzas se aprecia que aunque las cinco variables se miden en las mismas unidades (notas sobre 100), las cuasivarianzas son bastante diferentes. Estas relaciones se verán reflejadas en las componentes principales.*

Tabla 4.4: Correlaciones entre las 5 variables del fichero `nota.rda`.

	Mecanica	Vectores	Algebra	Analisis	Estadist
Mecanica	1.0000000	0.5534052	0.5467511	0.4093920	0.3890993
Vectores	0.5534052	1.0000000	0.6096447	0.4850813	0.4364487
Algebra	0.5467511	0.6096447	1.0000000	0.7108059	0.6647357
Analisis	0.4093920	0.4850813	0.7108059	1.0000000	0.6071743
Estadist	0.3890993	0.4364487	0.6647357	0.6071743	1.0000000

[1]Para leer este tipo de archivos teclear `load('f:/.../name.rda')` indicando la ruta completa en donde se encuentra el archivo y reemplazando name por el nombre del archivo. Alternativamente, se puede cambiar el directorio de trabajo en Session>Set working directory y teclear `load('name.rda')`. Obviamente b orrará el contenido anterior de `d`.

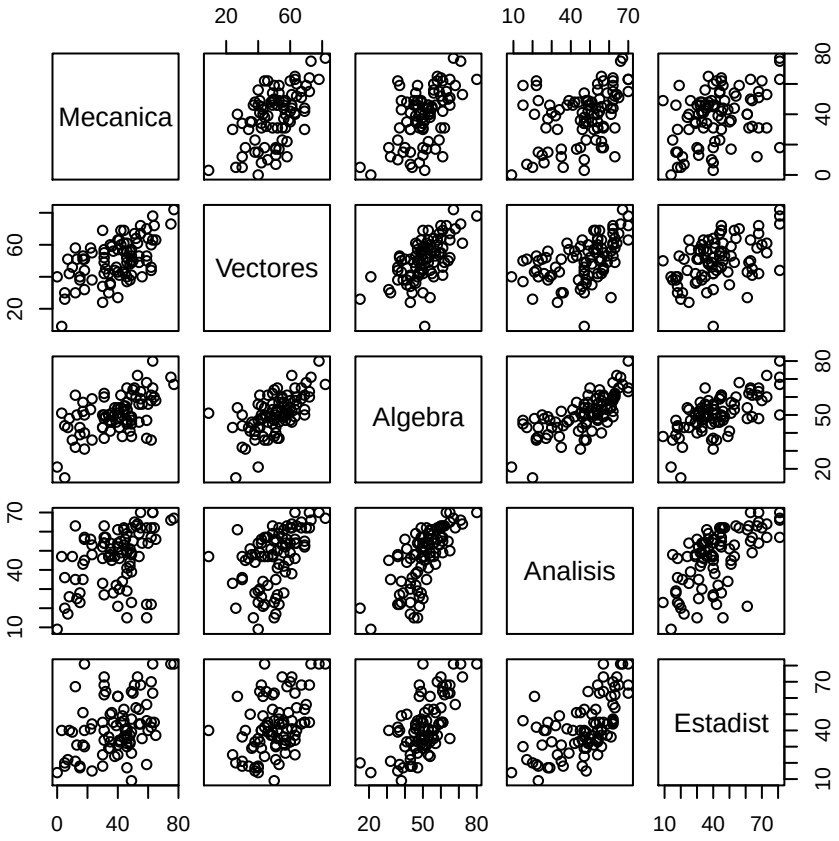


Figura 4.2: Gráficos bidimensionales para las 5 variables del fichero `nota.rda`.

Ejemplo 4.3. *En el tercer ejemplo analizamos los datos del fichero* `heptathlon` *del paquete de R* `MVA`[2] *correspondientes a los resultados en la prueba femenina de heptatlon en las olimpiadas de Seul 1988 (ver Tabla 4.5). Las variables corresponden a las pruebas: hurdles (110 m. vallas, en segundos), highjump (salto de altura, en metros), shot (lanzamiento de peso, en metros), run200m (carrera de 200m., en segundos), longjump (salto de longitud, en metros), javelin (lanzamiento de jabalina, en metros), run800m (carrera de 800m., en segundos) y score (puntuación final).*

[2]Para leer este conjunto de datos hay que instalar el paquete `MVA` pinchando en el menú: `Paquetes >Instalar Paquetes` seleccionando `MVA` o tecleando en R: `library('MVA')`.

Tabla 4.5: Resultados de Heptatlon en la Olimpiada de Seul 1988.

	hurd.	HJ	shot	200m	LJ	jav.	800m	score
1	12.69	1.86	15.80	22.56	7.27	45.66	128.51	7291
2	12.85	1.80	16.23	23.65	6.71	42.56	126.12	6897
3	13.20	1.83	14.20	23.10	6.68	44.54	124.20	6858
4	13.61	1.80	15.23	23.92	6.25	42.78	132.24	6540
5	13.51	1.74	14.76	23.93	6.32	47.46	127.90	6540
6	13.75	1.83	13.50	24.65	6.33	42.82	125.79	6411
7	13.38	1.80	12.88	23.59	6.37	40.28	132.54	6351
8	13.55	1.80	14.13	24.48	6.47	38.00	133.65	6297
9	13.63	1.83	14.28	24.86	6.11	42.20	136.05	6252
10	13.25	1.77	12.62	23.59	6.28	39.06	134.74	6252
11	13.75	1.86	13.01	25.03	6.34	37.86	131.49	6205
12	13.24	1.80	12.88	23.59	6.37	40.28	132.54	6171
13	13.85	1.86	11.58	24.87	6.05	47.50	134.93	6137
14	13.71	1.83	13.16	24.78	6.12	44.58	142.82	6109
15	13.79	1.80	12.32	24.61	6.08	45.44	137.06	6101
16	13.93	1.86	14.21	25.00	6.40	38.60	146.67	6087
17	13.47	1.80	12.75	25.47	6.34	35.76	138.48	5975
18	14.07	1.83	12.69	24.83	6.13	44.34	146.43	5972
19	14.39	1.71	12.68	24.92	6.10	37.76	138.02	5746
20	14.04	1.77	11.81	25.61	5.99	35.68	133.90	5734
21	14.31	1.77	11.66	25.69	5.75	39.48	133.35	5686
22	14.23	1.71	12.95	25.50	5.50	39.64	144.02	5508
23	14.85	1.68	10.00	25.23	5.47	39.14	137.30	5290
24	14.53	1.71	10.83	26.61	5.50	39.26	139.17	5289
25	16.42	1.50	11.78	26.16	4.88	46.38	163.43	4566

Para estudiar las relaciones entre las variables podemos hacer

```
plot(heptathlon)
```

La gráfica se puede ver en la Figura 4.3.

Podemos calcular las matrices de covarianza y correlación con `cov(d)` *y* `cor(d)`*, respectivamente. Las correlaciones se pueden ver la Tabla 4.6. Se aprecia que algunas variables tienen correlaciones (lineales) positivas y otras negativas. Note que en algunas variables (las carreras) es mejor tener valores bajos (poco tiempo) mientras que en otras es mejor tener valores altos (lanzamientos y saltos). Estas relaciones se verán reflejadas en las componentes principales.*

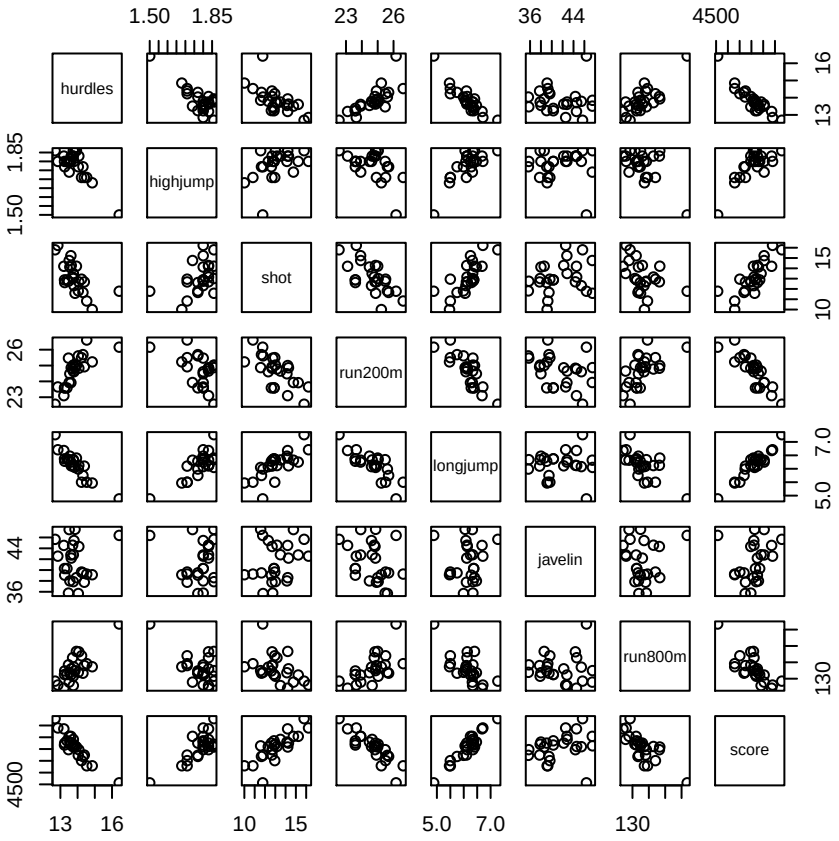


Figura 4.3: Gráficos bidimensionales para las variables del fichero `heptathlon`.

Pueden pensarse ejemplos similares. Por ejemplo: ¿Cómo construir unos índices que midan los cambios de precios en un país? ¿Qué relación hay entre las variables de un recién nacido, una persona o una animal? ¿Cómo resumir los indicadores económicos de diversos países?

Otras veces simplemente se querrá disminuir el número de variables porque son muchas perdiendo la menor información posible (comprimiendo los datos). De hecho, esta es una de las técnicas más usadas en el Big Data para este propósito. En particular, este procedimiento se puede usar para representar nuestros datos en un único gráfico bidimensional (en cualquiera de las técnicas vistas en este libro).

Tabla 4.6: Correlaciones entre las variables del fichero `heptathlon`.

	hurd.	HJ	shot	200m
hurd.	1.000000000	-0.811402536	-0.6513347	0.7737205
HJ	-0.811402536	1.000000000	0.4407861	-0.4876637
shot	-0.651334688	0.440786140	1.0000000	-0.6826704
200m	0.773720543	-0.487663685	-0.6826704	1.0000000
LJ	-0.912133617	0.782442273	0.7430730	-0.8172053
jav.	-0.007762549	0.002153016	0.2689888	-0.3330427
800m	0.779257110	-0.591162823	-0.4196196	0.6168101
score	-0.923198458	0.767358719	0.7996987	-0.8648825
	LJ	jav.	800m	score
hurd.	-0.91213362	-0.007762549	0.77925711	-0.9231985
HJ	0.78244227	0.002153016	-0.59116282	0.7673587
shot	0.74307300	0.268988837	-0.41961957	0.7996987
200m	-0.81720530	-0.333042722	0.61681006	-0.8648825
LJ	1.00000000	0.067108409	-0.69951116	0.9504368
jav.	0.06710841	1.000000000	0.02004909	0.2531466
800m	-0.69951116	0.020049088	1.00000000	-0.7727757
score	0.95043678	0.253146604	-0.77277571	1.0000000

4.2. Componentes principales

Desde el punto de vista teórico la idea es resumir la información de un vector aleatorio (v.a.) $k-$dimensional $X = (X_1, \ldots, X_k)'$ (recuerde que A' denota la traspuesta de A, es decir, X es un vector columna) en unas "pocas" variables que proporcionen la información más relevante. Se puede dar una aproximación geométrica mediante el concepto de elipsoide de concentración.

Definición 4.1. *Si X es un vector aleatorio de dimensión k, media μ y matriz de covarianzas $V = (\sigma_{i,j})$ definida positiva, se define el* ***elipsoide de concentración*** *de X como*

$$E_k = \{x \in \mathbb{R}^k : (x-\mu)'V^{-1}(x-\mu) \leq k+2\}.$$

Puede probarse que existe un v.a. uniforme sobre E_k con media μ y matriz de covarianzas V (ver Zoroa y Zoroa 2008, p. 206). En la definición del elipsoide interviene la **distancia de Mahalanobis** basada en la matriz V entre x y la media μ dada por

$$d_V(x,\mu) = \sqrt{(x-\mu)'V^{-1}(x-\mu)}.$$

Esta distancia al cuadrado se puede calcular en R con `mahalanobis(x,mu,V)`. Por ejemplo, si queremos calcular las distancias al cuadrado de las 10 primeras atletas de `hepthatlon` a la media teclearemos:

```
mu<- colMeans(d)
mahalanobis(d[1:10,],mu,cov(d))
```

Además, si X es normal, el elipsoide se puede definir a partir de las curvas de nivel de la función de densidad ($f(x) = cte$) ya que

$$f(x) = \frac{1}{\sqrt{|V|(2\pi)^k}} \exp\left(-\frac{1}{2}(x-\mu)'V^{-1}(x-\mu)\right).$$

La mayor parte de los individuos (puntos) estarán dentro de este elipsoide y, si queremos distinguirlos con una única variable, parece claro que lo mejor sería proyectarlos sobre el eje mayor. Por ejemplo, para una Normal bivariante

$$N_2\left(\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, V = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}\right)$$

se tiene

$$\begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \end{pmatrix} = \frac{4}{3}x^2 - \frac{4}{3}xy + \frac{4}{3}y^2$$

por lo que el elipsoide de concentración sería

$$\frac{4}{3}x_1^2 - \frac{4}{3}x_1y_2 + \frac{4}{3}x_2^2 \leq 4.$$

Para calcular la inversa de V en R podemos hacer

```
V<-matrix(NA,2,2)
V[1,]<-c(1,1/2)
V[2,]<-c(1/2,1)
solve(V)
```

Para representar el elipsoide como en la Figura 4.4 podemos hacer

```
hc<-function(x1,x2) (4/3)*x1^2-(4/3)*x1*x2+(4/3)*x2^2
x<-seq(-3,3,length=1000)
y<-seq(-3,3,length=1000)
z<-outer(x,y,hc)
contour(x,y,z,levels=4)
contour(x,y,z,levels=c(1:6))
```

La función de densidad puede calcularse en $(1,1)$ con

```
mu<-c(0,0)
```

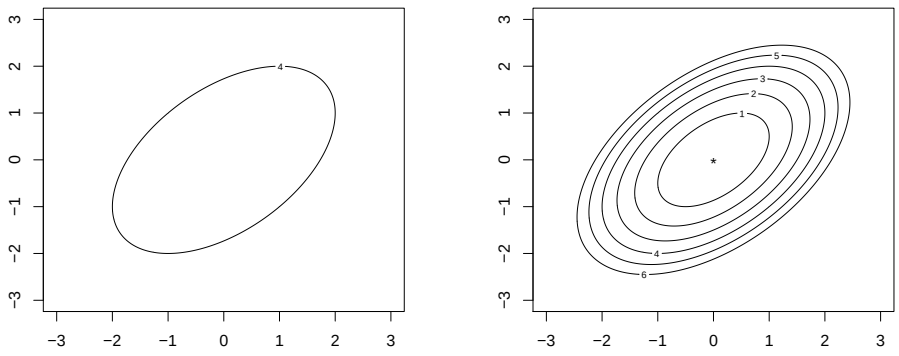


Figura 4.4: Elipsoide de concentración para una normal bidimensional con medias 0, varianzas 1 y correlación 1/2 (izquierda) y elipsoides obtenidos con otros niveles (circunferencias de Mahalanobis).

```
x<-c(1,1)
dmvnorm(x,mu,V)
```

Se debe obtener 0.0943539. Para simular 50 datos de este modelo usaremos `rmvnorm(50,mu,V)`. Su gráfica puede verse en la Figura 4.5 (derecha) junto con el elipsoide de concentración y los puntos simulados de ese modelo (izquierda). Las curvas de nivel son similares a las de la Figura 4.4, derecha. Para hacer gráficas 3D en R hay diversas opciones. Esa gráfica de la distribución normal se obtiene con:

```
f<-function(x1,x2)
dmvnorm(data.frame(x1,x2),mu,V)
x<-seq(-3,3,length=50)
y<-seq(-3,3,length=50)
z<-outer(x,y,f)
persp(x,y,z,xlab='x1',ylab='x2',zlab='f(x1,x2)',col='red')
```

Para realizar el gráfico de la izquierda podemos hacer:

```
set.seed(123)
d<-rmvnorm(50,mu,V)
plot(d,xlab="X1",ylab="X2",pch=20,xlim=c(-3,3),lim=c(-3,3))
contour(x,y,z,levels=4,add=T,col='red')
```

donde el comando `set.seed(123)` sirve para que siempre se genere la misma muestra (si queremos

otra basta borrarlo). La gráfica de la derecha también se puede obtener con:

```
f<-function(x1,x2)
exp(-(4/6)*x1*x1+(4/6)*x1*x2-(4/6)*x2*x2)
x<-seq(-3,3,length=50)
y<-seq(-3,3,length=50)
z<-outer(x,y,f)
persp(x,y,z,xlab='x1',ylab='x2',zlab='f(x1,x2)',col='red')
```

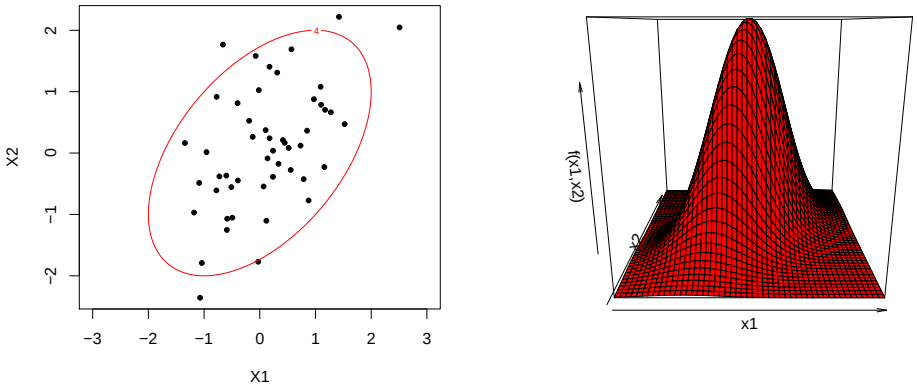


Figura 4.5: Elisoide de concentración (izquierda) y función de densidad para una normal bidimensional con medias cero, varianzas 1 y correlación 1/2.

Teniendo en cuenta esos gráficos, si queremos reducir las dos variables a solo una, la mejor "proyección", es decir la que mejor separa los puntos (varianza máxima), es la proporcionada por el eje principal del elipsoide (o curvas de nivel de la normal) que en este ejemplo viene dado por la recta $x_2 - x_1 = 0$. Lo mismo ocurre en dimensión k y en un modelo general como veremos en la sección siguiente.

4.3. Definición y cálculo teórico

Supongamos que $X = (X_1, \ldots, X_k)'$ es un v.a. k dimensional con vector de medias μ y matriz de covarianzas V semidefinida positiva. Entonces la primera componente principal será la v.a. unidimensional $Y_1 = a_1X_1 + \cdots + a_kX_k$ con $a_1^2 + \cdots + a_k^2 = 1$ cuya varianza es máxima. Nótese que si no se normaliza la combinación lineal, la variable Y_1 puede tener varianza tan grande como

queramos. Geométricamente, hacemos un cambio de variable (primer eje) para que la dispersión sea máxima y la normalización equivale a mantener la escala original (proyectar).

El problema puede expresarse de la forma siguiente:

$$\left.\begin{array}{l}\text{máx}\, Var(a'X)\\ s.a.: a'a=1\end{array}\right\}$$

donde $a=(a_1,\ldots,a_k)'\in\mathbb{R}^k$.

Una vez calculada una primera componente principal Y_1, la segunda componente principal Y_2 debe verificar $Cov(Y_1,Y_2)=0$ (no debe contener información ya incluida en Y_1) y debe tener varianza máxima, es decir

$$\left.\begin{array}{ll}\text{máx}\, Var(a'X) & \\ s.a.: & a'a=1\\ & Cov(Y_1,a'X)=0\end{array}\right\}$$

Así, sucesivamente, por inducción, se definen las siguientes componentes principales como la (una) solución de

$$\left.\begin{array}{ll}\text{máx}\, Var(a'X) & \\ s.a.: & a'a=1\\ & Cov(Y_i,a'X)=0, i=1,\ldots,j-1\end{array}\right\}$$

La solución general viene dada en el teorema siguiente que prueba la existencia de las (unas) componentes principales y muestra cómo calcularlas. Además, se demuestra que las componentes principales no son únicas (puede haber más soluciones).

Teorema 4.1. *Si X es un v.a. k dimensional con matriz de covarianzas V definida positiva, las (unas) componentes principales valen*

$$Y=(Y_1,\ldots,Y_k)'=T'X=\begin{pmatrix}t_{1,1} & \ldots & t_{k,1}\\ \ldots & \ldots & \ldots\\ t_{1,k} & \ldots & t_{k,k}\end{pmatrix}\begin{pmatrix}X_1\\ \ldots\\ X_k\end{pmatrix}$$

donde T es una matriz ortogonal ($T'T=TT'=I$) tal que $T'VT=D=diag(\lambda_1,\ldots,\lambda_k)$ con $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_k>0$.

Demostración. Como V es una matriz simétrica y definida positiva, existe una matriz $T=(t_{i,j})$ ortogonal ($T'T=TT'=I$) tal que $T'VT=D=diag(\lambda_1,\ldots,\lambda_k)$ con los valores propios verificando $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_k>0$ (ver Burgos 1994, pág. 630). De esta forma, si

$$Y=\begin{pmatrix}Y_1\\ \ldots\\ Y_k\end{pmatrix}=T'X=\begin{pmatrix}t_{1,1} & \ldots & t_{k,1}\\ \ldots & \ldots & \ldots\\ t_{1,k} & \ldots & t_{k,k}\end{pmatrix}\begin{pmatrix}X_1\\ \ldots\\ X_k\end{pmatrix}$$

entonces $Y_1,\ldots,Y_k$ verifican

$$Cov(Y)=Cov(T'X)=E(T'(X-\mu)(X-\mu)'T)=T'VT=D$$

lo que implica que $Cov(Y_i, Y_j) = 0$ para $i \neq j$ y $Var(Y_j) = \lambda_j$. Los vectores columnas de T (filas de T') $t_j = (t_{1,j}, \ldots, t_{k,j})'$, serán una base ortonormal de vectores propios de V verificándose $Y_j = t_j' X$ y $Vt_j = \lambda_j t_j$ para $j = 1, \ldots, k$.

Para comprobar que Y_1 es una primera componente principal, supongamos que $a'X$ es una combinación lineal con $a'a = 1$. Entonces, como los vectores propios son una base, existirán $c_1, \ldots, c_k$ números reales tales que

$$a = c_1 t_1 + \cdots + c_k t_k, \text{ con } c = (c_1, \ldots, c_k)'$$

con lo que

$$\begin{aligned} Var(a'X) &= E(a'(X-\mu)(X-\mu)'a) \\ &= a'Va \\ &= \left(\sum_{i=1}^{k} c_i t_i'\right) V \left(\sum_{i=1}^{k} c_i t_i\right) \\ &= \left(\sum_{i=1}^{k} c_i t_i'\right) \left(\sum_{i=1}^{k} c_i V t_i\right) \\ &= \left(\sum_{i=1}^{k} c_i t_i'\right) \left(\sum_{j=1}^{k} c_j \lambda_j t_j\right) \\ &= \sum_{i,j} c_i c_j \lambda_j t_i' t_j \\ &= \sum_{i=1}^{k} c_i^2 \lambda_i \end{aligned}$$

y, como

$$a'a = \left(\sum_{i=1}^{k} c_i t_i'\right) \left(\sum_{j=1}^{k} c_j t_j\right) = \sum_{i,j} c_i c_j t_i' t_j = \sum_{i=1}^{k} c_i^2 = c'c = 1$$

la varianza será máxima si $c_1^2 = 1, c_2 = 0, \ldots, c_k = 0$ ya que

$$Var(\pm t_1' X) = \lambda_1 = \sum_{i=1}^{k} c_i^2 \lambda_1 \geq \sum_{i=1}^{k} c_i^2 \lambda_i = Var(a'X),$$

para todo a tal que $a'a = 1$, es decir, $Y_1 = \pm t_1' X$ es una primera componente principal (puede haber otras soluciones si $\lambda_1 = \lambda_2$).

Por inducción, supongamos que $Y_1 = t_1' X, \ldots, Y_{j-1} = t_{j-1}' X$ son las primeras $(j-1)$ componentes principales y veamos que $Y_j = t_j' X$ es la (una) solución de

$$\left.\begin{array}{ll} \text{máx } Var(a'X) & \\ s.a.: & a'a = 1 \\ & Cov(a'X, Y_i) = 0, \ i = 1, \ldots, j-1 \end{array}\right\}$$

Como se debe verificar

$$Cov(a'X, Y_i) = Cov(a'X, t_i'X) = E(a'(X-\mu)(X-\mu)t_i) = a'Vt_i$$
$$= \lambda_i a't_i = \lambda_i \left(\sum_s c_s t_s'\right) t_i = \lambda_i c_i = 0$$

para $i = 1, \ldots, j-1$, y $\lambda_i > 0$, se tiene $c_1 = \cdots = c_{j-1} = 0$. Entonces, la varianza será máxima si $c_j = 1$ y $c_i = 0$ para $i > j$, ya que

$$Var(\pm t_j X) = \lambda_j = c'c\lambda_j = \sum_{i=j}^{k} c_i^2 \lambda_j \geq \sum_{i=j}^{k} c_i^2 \lambda_i = Var(a'X),$$

para todo a tal que $a'a = 1$ y $Cov(a'X, Y_i) = 0$, $i = 1, \ldots, j-1$, es decir, $Y_j = \pm t_j'X$, es una componente principal j-ésima (no necesariamente la única). □

Como consecuencia inmediata de la demostración anterior se tiene el corolario siguiente.

Corolario 4.1. *Si $\lambda_1 > \lambda_2 > \ldots > \lambda_k$, entonces las componentes principales son únicas salvo signo.*

Observación 4.1. *Nótese que la componente principal j-ésima se obtiene multiplicando la fila j-ésima de T' (la columna j-ésima de T) por X, es decir, $Y_j = t_j'X$ donde $t_j' = (t_{1,j}, \ldots, t_{k,j})$ es un vector propio unitario correspondiente al j-ésimo valor propio (vectores columna de T). Además, $Var(Y_j) = \lambda_j$ y*

$$traza(V) = \sum_{j=1}^{k} \sigma_{j,j} = \sum_{j=1}^{k} Var(X_j) = \sum_{j=1}^{k} Var(Y_j) = \sum_{j=1}^{k} \lambda_j$$

*(las matrices semejantes tienen las trazas iguales), es decir, la variabilidad (información) de las variables originales es igual a la suma de las variabilidades de las componentes principales. La **cantidad de información** (%) contenida en cada componente será $I_j = 100\lambda_j/(\sum_{i=1}^{k} \lambda_i)\,\%$. Por esto, la traza se usa como una medida unidimensional de la dispersión de una variable k-dimensional. La otra medida es el determinante de V para el que también se verifica:*

$$|V| = \lambda_1 \ldots \lambda_k = |Cov(Y)|$$

(es decir, la variabilidad se mide calculando el área encerrada en el paralelogramo de lados iguales a los valores propios).

Observación 4.2. *Otros autores llaman componentes principales a $Y = T'(X-\mu)$ con lo que, además, se consigue que sean centradas ($E(Y_j) = 0$). También se pueden definir las componentes principales estandarizadas $Z_j = t_j'(X-\mu)\lambda_j^{-1/2}$ ($Z = D^{-1/2}T'(X-\mu)$) que además de ser centradas tendrán varianza 1.*

Observación 4.3. *Cuando hay valores propios iguales a cero (V es semidefinida positiva) no suelen considerarse sus correspondientes componentes principales (degeneradas) y se puede conservar toda la información en las componentes principales de valores propios distintos de cero. En este caso hay variables que pueden obtenerse como combinación lineal de las restantes (aunque no siempre pueden eliminarse del análisis).*

Observación 4.4. *Como comentamos al inicio, geométricamente, las componentes principales se corresponden con los ejes principales del elipsoide de concentración. Como $Y = T'X$, podemos interpretar las componentes en función de los pesos que tengan en ellas las variables originales. Si ponemos X en función de Y como $X = TY$, entonces las variables originales se pueden interpretar en función de las componentes principales e incluso, podemos representar aproximadamente, las variables originales usando las dos (tres) primeras componentes.*

Si la población X es normal, entonces las componentes principales son normales e independientes entre sí, ya que en éstas poblaciones equivalen independencia e incorrelación (independencia lineal) y Z será una normal estándar multivariante ($N_k(0, I)$).

Proposición 4.1. *Si Y son las componentes principales obtenidas a partir de X, entonces X es normal multivariante si, y solo si $Y_1, \ldots, Y_k$ son independientes y normales univariantes para todo $j = 1, \ldots, k$.*

La demostración es inmediata. Esta propiedad puede ser utilizada para estudiar la normalidad multivariante a partir de un test de normalidad univariante sobre las componentes principales. Incluso si la normal multivariante no es de rango completo (V no es definida positiva), puede utilizarse con las m primeras componentes con valores propios distintos de cero (las otras serán degeneradas) coincidiendo m con el rango de V.

Ejemplo 4.4. *Para la v.a. normal de media $\mu = (0, 0)'$ y matriz de covarianzas*

$$V = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

(ver Figura 4.5), sus componentes principales se calcularán diagonalizando V mediante

$$|V - \lambda I| = \begin{vmatrix} 1-\lambda & 0.5 \\ 0.5 & 1-\lambda \end{vmatrix} = 1 - 2\lambda + \lambda^2 - 1/4 = 0$$

que tiene soluciones

$$\lambda = \frac{2 \pm \sqrt{4 - 4(1 - 1/4)}}{2} = 1 \pm 0.5,$$

y la primera componente se obtendrá resolviendo

$$\begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = (1 + 0.5) \begin{pmatrix} x \\ y \end{pmatrix},$$
$$\begin{pmatrix} -0.5x + 0.5y \\ 0.5x - 0.5y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

lo que da $x = y$, es decir, sus vectores propios son $v = \lambda(1,1)'$. Como usamos vectores normalizados (de norma 1), una primera componente valdrá

$$Y_1 = \left(1/\sqrt{2}, 1/\sqrt{2}\right) \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = (X_1 + X_2)/\sqrt{2}$$

y su varianza es $\lambda_1 = 1.5$. Análogamente, la segunda valdrá $Y_2 = (X_1 - X_2)/\sqrt{2}$ (ya que tiene que ser perpendicular a la primera) y tendrá varianza $\lambda_2 = 0.5$. Es decir, tenemos

$$\begin{aligned} Y_1 &= (X_1 + X_2)/\sqrt{2} \\ Y_2 &= (X_1 - X_2)/\sqrt{2} \end{aligned}$$

por lo que

$$Y = T'X = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}.$$

La primera componente explicará un $I_1 = 100\lambda_1/(\lambda_1 + \lambda_2)\,\% = 75\,\%$ de la varianza total y la segunda un $I_2 = 100\lambda_2/(\lambda_1 + \lambda_2)\,\% = 25\,\%$. Como las varianzas iniciales son iguales, ambas tienen igual peso en las componentes con distinto signo en el caso de la segunda de ellas. Nótese que aunque las varianzas iniciales sean todas iguales (1) las componentes principales tienen varianzas (en general) distintas. Si X_1 fuese el peso de una persona y X_2 su altura (estandarizadas), la primera componente se podría interpretar como lo "grande" que es dicha persona, mientras que la segunda estaría relacionada con su "constitución" (Y_2 grande significaría gran peso y poca altura, es decir, complexión fuerte). Despejando, se tiene

$$\begin{aligned} X_1 &= (Y_1 + Y_2)/\sqrt{2} \\ X_2 &= (Y_1 - Y_2)/\sqrt{2}, \end{aligned}$$

lo que nos permite representar las variables X_1, X_2 en función de las componentes Y_1, Y_2 (ver Figura 4.6). Nótese que Y_1 aumenta si lo hacen X_1 y X_2 mientras que Y_2 aumenta si aumenta X_1 y disminuye X_2. Estas relaciones servirán para interpretar (dar significado) a las componentes principales.

Para realizar estos cálculos en R introduciremos los comandos siguientes. El primer lugar definimos e introducimos V con:

```
V<-matrix(nrow=2,ncol=2)
V[1,1]<-1
V[2,2]<-1
V[2,1]<-1/2
V[1,2]<-1/2
```

Tecleando `V` podemos comprobar que hemos introducido los datos bien. Para calcular los valores y vectores propios haremos: `eigen(V)`. Si queremos guardar la matriz T de vectores propios haremos

```
eigen(V)$vectors->T
```

Recuerde que los vectores normalizados aparecen en las columnas de T. Para comprobar que T es una matriz ortogonal haremos `t(T)%*%T` *donde* `t(A)` *es la traspuesta de A y* `A%*%B` *es el producto de las matrices A y B en R. De esta forma, si queremos comprobar que T diagonaliza a V haremos*

```
t(T)%*%V%*%T
```

lo que nos dará (aproximadamente) la matriz diagonal con los valores 1.5 y 0.5 en la diagonal. Como $Y = T'X$, la primera componente principal será $Y_1 = 0.7071068X_1 + 0.7071068X_2$ y la segunda $Y_1 = -0.7071068X_1 + 0.7071068X_2$. Para calcular las informaciones contenidas en cada una (en tanto por 100) haremos:

```
100*eigen(V)$values/sum(eigen(V)$values)
```

obteniendo 75 y 25.

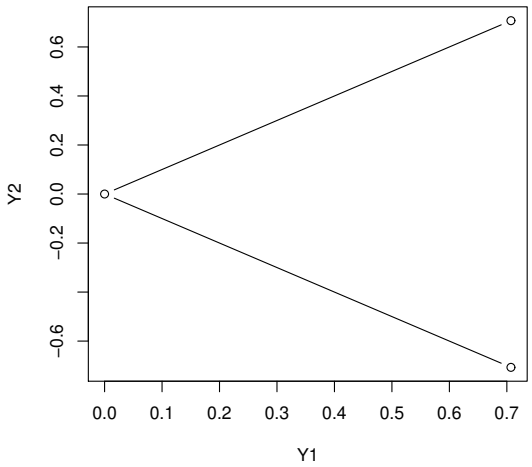


Figura 4.6: Variables del Ejemplo 4.4 en función de las componentes principales.

La desigualdad de Chebyshev para variables aleatorias da una cota inferior en función de la varianza para el porcentaje de valores a una distancia dada de la media. Se puede obtener de la desigualdad de Markov siguiente. Si Z es una variable aleatoria no negativa con media finita $E(Z)$

y $\varepsilon > 0$, entonces

$$\varepsilon \Pr(Z \geq \varepsilon) = \varepsilon \int_{[\varepsilon,\infty)} dF_Z(x) \leq \int_{[\varepsilon,\infty)} x dF_Z(x) \leq \int_{[0,\infty)} x dF_Z(x) = E(Z)$$

(donde $F_Z(x) = \Pr(Z \leq x)$ es su función de distribución), es decir

$$\Pr(Z \geq \varepsilon) \leq \frac{E(Z)}{\varepsilon}. \tag{4.1}$$

La desigualdad de Chebyshev se obtiene como sigue. Si X es una variable aleatoria con media finita $\mu = E(X)$ y varianza $\sigma^2 = Var(X) > 0$, entonces tomando $Z = (X-\mu)^2/\sigma^2 \geq 0$ en (4.1), tenemos

$$\Pr\left(\frac{(X-\mu)^2}{\sigma^2} \geq \varepsilon\right) \leq \frac{1}{\varepsilon} \tag{4.2}$$

para todo $\varepsilon > 0$. También se puede escribir como

$$\Pr((X-\mu)^2 < \varepsilon\sigma^2) \geq 1 - \frac{1}{\varepsilon}$$

o como

$$\Pr(|X-\mu| < r) \leq 1 - \frac{\sigma^2}{r^2}$$

para todo $r > 0$. De forma análoga, la desigualdad de Chebyshev multivariante se obtiene usando las componentes principales como sigue.

Teorema 4.2. *Sea $X = (X_1, \ldots, X_k)'$ un vector aleatorio con vector de medias finito $\mu = E(X)$ y matriz de covarianzas definida positiva V, entonces*

$$\Pr((X-\mu)'V^{-1}(X-\mu) \geq \varepsilon) \leq \frac{k}{\varepsilon} \tag{4.3}$$

para todo $\varepsilon > 0$.

Demostración. Como V es definida positiva y simétrica existe T tal que $TT' = T'T = I$ y $T'VT = D$, donde $D = diag(\lambda_1, \ldots, \lambda_k)$ es la matriz diagonal con los valores propios ordenados $\lambda_1 \geq \cdots \geq \lambda_k > 0$. Entonces $V = TDT'$ y $V^{-1} = TD^{-1}T'$. Entonces la variable aleatoria no negativa

$$Z = (X-\mu)'V^{-1}(X-\mu)$$

se puede escribir como

$$Z = (X-\mu)'TD^{-1}T'(X-\mu) = [D^{-1/2}T'(X-\mu)]'[D^{-1/2}T'(X-\mu)] = \mathbf{Z}'\mathbf{Z},$$

donde

$$\mathbf{Z} = D^{-1/2}T'(X-\mu)$$

y $D^{-1/2} = diag(\lambda_1^{-1/2}, \ldots, \lambda_n^{-1/2})$. El vector aleatorio $\mathbf{Z} = (Z_1, \ldots, Z_k)$ (con las componentes principales estandarizadas) verifica

$$E(\mathbf{Z}) = E(D^{-1/2}T'(X-\mu)) = D^{-1/2}T'E(X-\mu) = \mathbf{0}_k$$

y

$$Cov(\mathbf{Z}) = Cov(D^{-1/2}T'(X-\mu)) = D^{-1/2}T'VTD^{-1/2} = D^{-1/2}DD^{-1/2} = I_k.$$

Por lo tanto

$$E(Z) = E(\mathbf{Z}'\mathbf{Z}) = E\left(\sum_{i=1}^{k} Z_i^2\right) = \sum_{i=1}^{k} E(Z_i^2) = \sum_{i=1}^{k} Var(Z_i) = k.$$

Entonces, usando la desigualdad de Markov (4.1), tenemos

$$\Pr(Z \geq \varepsilon) = \Pr((X-\mu)'V^{-1}(X-\mu) \geq \varepsilon) \leq \frac{E(Z)}{\varepsilon} = \frac{k}{\varepsilon}$$

para todo $\varepsilon > 0$, lo que finaliza la demostración. □

La desigualdad (4.3) también se puede escribir como

$$\Pr((X-\mu)'V^{-1}(X-\mu) < \varepsilon) \geq 1 - \frac{k}{\varepsilon} \tag{4.4}$$

para todo $\varepsilon > 0$. En particular, para el elipsoide de concentración

$$E_k = \{x \in \mathbb{R}^k : (x-\mu)'V^{-1}(x-\mu) \leq k+2\},$$

obtenemos

$$\Pr(X \in E_k) \geq 1 - \frac{k}{k+2} = \frac{2}{k+2}.$$

Para obtener regiones con más datos podemos tomar $\varepsilon = ck$, resultando

$$\Pr((X-\mu)'V^{-1}(X-\mu) < ck) \geq 1 - \frac{k}{\varepsilon} = 1 - \frac{1}{c} = \frac{c-1}{c}. \tag{4.5}$$

Si X es normal, entonces $Z_1, \ldots, Z_n$ son normales independientes y $Z = \sum_{i=1}^{k} Z_i^2$ sigue una distribución chi-cuadrado con k grados de libertad (ya que es la suma de k normales $N(0,1)$ independientes).

4.4. Propiedades

En primer lugar estudiaremos las relaciones entre la nuevas variables y las componentes principales obtenidas mediante la matriz de covarianzas V.

Proposición 4.2. *Si Y son las componentes principales obtenidas a partir de X, entonces*

$$\begin{aligned} Cov(X,Y) &= TD \\ Corr(X,Y) &= diag(V)^{-1/2}TD^{1/2} \end{aligned} \tag{4.6}$$

donde $diag(V) = diag(\sigma_1^2, \ldots, \sigma_k^2)$.

Demostración. En primer lugar señalaremos que

$$Cov(X,Y) = Cov(X, T'X) = VT$$

y, como $T'VT = D$ y T es ortogonal, entonces $VT = TD$ y $Cov(X,Y) = TD$.

Por otro lado se tiene que como

$$Corr(X_i, Y_j) = \frac{Cov(X_i, Y_j)}{\sigma_i \lambda_j^{1/2}},$$

entonces $Corr(X,Y) = diag(V)^{-1/2} Cov(X,Y) D^{-1/2}$ y

$$Corr(X,Y) = diag(V)^{-1/2} TDD^{-1/2} = diag(V)^{-1/2} TD^{1/2}.$$

□

Corolario 4.2. *En las condiciones de la proposición anterior se tiene:*

$$Cov(X_i, Y_j) = t_{i,j}\lambda_j$$

y

$$Corr(X_i, Y_j) = \frac{t_{i,j}}{\sigma_i}\lambda_j^{1/2}$$

para todo i, j.

Definición 4.2. *Llamaremos **matriz de saturaciones** a $A = Corr(X,Y)$.*

Ejemplo 4.5. *Para el vector aleatorio $(X_1, X_2)'$ con normal de media $\mu = (0,0)'$ y matriz de covarianzas*

$$V = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix},$$

se obtiene

$$T = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix}, D = \begin{pmatrix} 1.5 & 0 \\ 0 & 0.5 \end{pmatrix}$$

por lo que la matriz de saturaciones valdrá:

$$A = TD^{1/2} = \frac{1}{2}\begin{pmatrix} \sqrt{3} & 1 \\ \sqrt{3} & -1 \end{pmatrix} = \begin{pmatrix} 0.86603 & 0.5 \\ 0.86603 & -0.5 \end{pmatrix}.$$

Nótese que la primera componente explica un 75 % ($0.86603^2 100$) de las variables X_1 y X_2, mientras que la segunda solo un 25 %. Las saturaciones y sus cuadrados suelen representarse en tablas de la forma siguiente:

$a_{i,j}$	Y_1	Y_2
X_1	*0.866*	*0.5*
X_2	*0.866*	*-0.5*

$a_{i,j}^2$	Y_1	Y_2	*total*
X_1	*0.75*	*0.25*	*1*
X_2	*0.75*	*0.25*	*1*

lo que nos puede ayudar a "interpretar" las componentes principales. Las saturaciones también se pueden representar gráficamente igual que hacíamos con los coeficientes en la Figura 4.6. Aunque en este ejemplo, las saturaciones con las distintas variables coincidan, esto no siempre es así, y tendremos variables mejor explicadas por las componentes elegidas que otras.

Proposición 4.3. *Si A es la matriz de saturaciones, entonces*

$$AA' = Corr(X)$$

Demostración. Si multiplicamos se obtiene

$$\begin{aligned} AA' &= diag(V)^{-1/2} TD^{1/2} (diag(V)^{-1/2} TD^{1/2})' \\ &= diag(V)^{-1/2} TD^{1/2} D^{1/2} T' diag(V)^{-1/2} \\ &= diag(V)^{-1/2} TDT' diag(V)^{-1/2} \\ &= diag(V)^{-1/2} V diag(V)^{-1/2} \\ &= Corr(X). \end{aligned}$$

□

También se pueden obtener las correlaciones múltiples entre cada variable original y el grupo de componentes principales elegidas lo que nos dará una idea de lo bueno que es el modelo formado por las componentes principales (y los correspondientes coeficientes) para predecir cada variable original. Recordaremos la definición de coeficiente de correlación múltiple.

Definición 4.3. *Si $X = (X_1, \ldots, X_k)'$ es un vector aleatorio se llama* ***coeficiente de correlación múltiple****(al cuadrado) de X_1 respecto de $Z = (X_2, \ldots, X_k)'$ a*

$$Corr^2(X_1, Z) = \rho^2_{1(2,\ldots,k)} = \frac{v'_{1,2} V_{2,2}^{-1} v_{1,2}}{\sigma_{1,1}}$$

donde $Cov(X) = \begin{pmatrix} \sigma_{1,1} & v'_{1,2} \\ v_{1,2} & V_{2,2} \end{pmatrix}$, $Cov(Z) = V_{2,2}$, $\sigma_{1,1} = Var(X_1)$ *y* $v'_{1,2} = (\sigma_{1,2}, \ldots, \sigma_{1,k}) = Cov(X_1, Z)$.

Nótese que si $k = 2$, entonces $\rho^2_{1(2)} = \sigma_{1,2}\sigma^{-1}_{2,2}\sigma_{1,2}/\sigma_{1,1} = \rho^2_{1,2}$. La interpretación de este coeficiente se obtiene del resultado siguiente.

Proposición 4.4. *El coeficiente de correlación múltiple es el máximo de las correlaciones (al cuadrado) de X_1 con combinaciones lineales de $Z = (X_2, \ldots, X_k)'$, es decir*

$$\max_{\alpha} Corr^2(X_1, \alpha' Z) = \rho^2_{1(2,\ldots,k)}$$

Demostración. De la definición se tiene

$$\begin{aligned} Corr^2(X_1, \alpha' Z) &= \frac{(Cov(X_1, \alpha' Z))^2}{\sigma_{1,1} Var(\alpha' Z)} = \frac{(Cov(X_1, Z)\alpha)^2}{\sigma_{1,1} Cov(\alpha' Z, \alpha' Z)} \\ &= \frac{(\alpha' v_{1,2})^2}{\sigma_{1,1}\alpha' V_{2,2}\alpha} = \frac{(\alpha' V^{1/2}_{2,2} V^{-1/2}_{2,2} v_{1,2})^2}{\sigma_{1,1}\alpha' V_{2,2}\alpha} \end{aligned}$$

y, usando la desigualdad de Cauchy-Schwarz ($(x'y)^2 \leq (x'x)(y'y)$) para $x' = \alpha' V^{1/2}_{2,2}$ e $y = V^{-1}_{2,2} v_{1,2}$, se tiene

$$Corr^2(X_1, \alpha' Z) \leq \frac{\alpha' V_{2,2}\alpha v'_{1,2} V^{-1}_{2,2} v_{1,2}}{\sigma_{1,1}\alpha' V_{2,2}\alpha} = \frac{v'_{1,2} V^{-1}_{2,2} v_{1,2}}{\sigma_{1,1}},$$

es decir $\rho^2_{1(2,\ldots,k)}$ es una cota superior. Además, la igualdad se obtiene si x e y tienen la misma dirección

$$x = V^{1/2}_{2,2}\alpha = \lambda y = \lambda V^{-1/2}_{2,2} v_{1,2},$$

es decir, si $\alpha = \lambda V^{-1}_{2,2} v_{1,2}$. □

Proposición 4.5. *Si las variables de $Z = (X_2, \ldots, X_k)'$ son independientes (incorreladas) entre sí, entonces*

$$Corr^2(X_1, Z) = \sum_{j=2}^{k} Corr^2(X_1, X_j).$$

Demostración. La demostración es inmediata ya que al ser $V_{2,2}$ diagonal, se tiene

$$Corr^2(X_1, Z) = \frac{v'_{1,2} V^{-1}_{2,2} v_{1,2}}{\sigma_{1,1}} = \sum_{j=2}^{k} \frac{\sigma^2_{1,j}}{\sigma_{1,1}\sigma_{j,j}} = \sum_{j=2}^{k} \rho^2_{1,j}.$$

□

Por lo tanto, es interesante calcular las correlaciones múltiples de cada variable original con el grupo de las p primeras componentes principales elegidas ($p \leq k$) para medir el máximo que podemos explicar de cada variable original a partir de combinaciones lineales esas componentes principales.

Proposición 4.6. *Si Y son las componentes principales obtenidas a partir de X, entonces*

$$Corr^2(X_i,(Y_1,\ldots,Y_p)) = \sum_{j=1}^{p} Corr^2(X_i,Y_j) = \frac{1}{\sigma_{i,i}}\sum_{j=1}^{p} t_{i,j}^2\lambda_j = \sum_{j=1}^{p} a_{i,j}^2.$$

La demostración es inmediata ya que las componentes son incorreladas entre sí. A estas correlaciones se las suele denominar **comunalidades**

$$c_i = Corr^2(X_i,(Y_1,\ldots,Y_p))$$

y se suelen representar en la tabla de las saturaciones al cuadrado (como totales de las filas). Además, el máximo de la correlación se obtiene con la combinación lineal $\alpha_i'(Y_1,\ldots,Y_p)'$ con

$$\alpha_i = \lambda V_{2,2}^{-1} v_{1,2} = \lambda \begin{pmatrix} \lambda_1^{-1} & \ldots & 0 \\ \ldots & \ldots & \ldots \\ 0 & \ldots & \lambda_p^{-1} \end{pmatrix} \begin{pmatrix} t_{i,1}\lambda_1 \\ \ldots \\ t_{i,p}\lambda_p \end{pmatrix} = \lambda \begin{pmatrix} t_{i,1} \\ \ldots \\ t_{i,p} \end{pmatrix}.$$

Es decir, si tenemos que obtener X en función de las p primeras componentes principales, lo haremos a partir de la relación $X = TY$ eliminando el resto de las componentes. Lógicamente, si $p = k$, se obtiene $\alpha_i'(Y_1,\ldots,Y_k)' = \lambda X_i$ y $Corr^2(X_i,(Y_1,\ldots,Y_k)) = 1$ y la igualdad

$$Corr^2(X_i,(Y_1,\ldots,Y_k)) = \sum_{j=1}^{k} Corr^2(X_i,Y_j) = \frac{1}{\sigma_{i,i}}\sum_{j=1}^{k} t_{i,j}^2\lambda_j = 1.$$

Recíprocamente, la información contenida en la componente principal j-ésima vale:

$$\lambda_j = \lambda_j \sum_{i=1}^{k} t_{i,j}^2 = \sum_{i=1}^{k} \sigma_{i,i}\frac{1}{\sigma_{i,i}} t_{i,j}^2\lambda_j = \sum_{i=1}^{k} \sigma_{i,i} Corr^2(X_i,Y_j),$$

ya que $1 = \sum_{i=1}^{k} t_{i,j}^2$ es el módulo al cuadrado del vector propio t_j (columnas de T) y la información (variación) total contenida en las p primeras componentes principales vale:

$$\sum_{j=1}^{p}\lambda_j = \sum_{j=1}^{p}\sum_{i=1}^{k} \sigma_{i,i} Corr^2(X_i,Y_j) = \sum_{i=1}^{k} \sigma_{i,i} Corr^2(X_i,(Y_1,\ldots,Y_p)) = \sum_{i=1}^{k} c_i\sigma_i^2.$$

Si todas las varianzas son 1, la información total $\sum_{j=1}^{p}\lambda_j$ será la suma de la comunalidades, es decir, la suma de la información que se tiene de cada variable original. Si $p = k$, entonces $c_i = 1$ y se tiene $sum_{j=1}^{k}\lambda_j = \sum_{i=1}^{k}\sigma_i^2$.

En el ejemplo anterior tenemos

$a_{i,j}^2$	Y_1	Y_2	Total
X_1	0.75	0.25	1
X_2	0.75	0.25	1
Total	1.5	0.5	2

donde si $p = 1$ se tiene $\lambda_1 = 3/2 = 0.75 + 0.75$ y si $p = 2$, se tiene $\lambda_1 + \lambda_2 = 3/2 + 1/2 = 2 = 1 + 1 = \sigma_1^2 + \sigma_2^2$.

4.5. Cálculo a partir de la matriz de correlaciones

Cuando se estudian variables en las que se usan unidades diferentes o queremos que éstas no sean significativas (todas las variables sean iguales a priori), las componentes principales suelen calcularse a partir de la matriz de correlaciones $\Pi = (\rho_{i,j})$ con $\rho_{i,j} = \sigma_{i,j}/(\sigma_i\sigma_j)$, lo que equivale a considerar desde el principio las variables estandarizadas $Z_i = (X_i - \mu_i)/\sigma_i$ (se igualan las varianzas a 1). De esta forma, usando el teorema principal, se obtienen las componentes

$$\widetilde{Y} = \widetilde{T}'Z = \widetilde{T}'diag(V)^{-1/2}(X - \mu)$$
$$\widetilde{Y}_j = \widetilde{t}_j'Z = \sum_{i=1}^{k} \widetilde{t}_{i,j} Z_i = \sum_{i=1}^{k} \widetilde{t}_{i,j} \frac{X_i - \mu_i}{\sigma_i},$$

donde $\widetilde{T}$ es la matriz ortogonal que diagonaliza $\Pi = Corr(X) = Cov(Z)$

$$\widetilde{T}'\Pi\widetilde{T} = diag(\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_k) = \widetilde{D}$$

$\Pi\widetilde{t}_j = \lambda_j\widetilde{t}_j$ y $Z = (Z_1, \ldots, Z_k)'$. De esta forma, se obtiene

$$Cov(\widetilde{Y}) = Cov(\widetilde{T}'Z) = \widetilde{T}'\Pi\widetilde{T} = \widetilde{D}.$$

Es decir, las componentes principales obtenidas a partir de la matriz de correlaciones serán las variables incorreladas con varianza máxima que se pueden obtener a partir de combinaciones lineales de las variables estandarizadas $Z = diag(V)^{-1/2}(X - \mu)$. Sin embargo, los resultados que se obtienen son (en general) diferentes de los que se obtienen a partir de V.

Proposición 4.7. *Si $\widetilde{Y}$ son las componentes principales obtenidas a partir de la matriz de correlaciones de X, entonces*

$$Corr(X, \widetilde{Y}) = \widetilde{T}\widetilde{D}^{1/2}.$$

En efecto, si $\widetilde{Y} = \widetilde{T}'Z = \widetilde{T}'diag(V)^{-1/2}(X - \mu)$, entonces

$$Cov(Z, \widetilde{Y}) = Cov(Z, \widetilde{T}'Z) = \Pi\widetilde{T} = \widetilde{T}\widetilde{D}.$$
$$Corr(X, \widetilde{Y}) = Corr(Z, \widetilde{Y}) = Cov(Z, \widetilde{Y})\widetilde{D}^{-1/2} = \widetilde{T}\widetilde{D}\widetilde{D}^{-1/2} = \widetilde{T}\widetilde{D}^{1/2}.$$

Observación 4.5. *Nótese que las correlaciones con la componente $\widetilde{Y}_j$ son proporcionales al vector propio $\widetilde{t}_j$ (columnas de $\widetilde{T}$) con constante de proporcionalidad $\widetilde{\lambda}_j^{1/2}$ ($Corr(X_i, \widetilde{Y}_j) = \widetilde{t}_{i,j}\widetilde{\lambda}_j^{1/2}$) y que*

$$\sum_{i=1}^{k} Corr^2(X_i, \widetilde{Y}_j) = \sum_{i=1}^{k} \widetilde{t}_{i,j}\widetilde{\lambda}_j = \widetilde{\lambda}_j.$$

De forma similar, se define la matriz de saturaciones $\widetilde{A} = Corr(X, \widetilde{Y})$, que verifica

$$\widetilde{A}\widetilde{A}' = \widetilde{T}\widetilde{D}^{1/2}\widetilde{D}^{1/2}\widetilde{T}' = Cov(Z) = Corr(X)$$

(como vimos en la sección anterior) y

$$\widetilde{A}'\widetilde{A} = \widetilde{D}^{1/2}\widetilde{T}'\widetilde{T}\widetilde{D}^{1/2} = \widetilde{D},$$

es decir, la matriz de saturación es una matriz que factoriza Π junto a su traspuesta de forma que si las multiplicamos al revés nos da una matriz diagonal.

4.6. Cálculo práctico de las componentes principales

Si estudiamos k variables (numéricas) en una determinada población usando una muestra de n individuos, tendremos una tabla de datos de la forma siguiente:

Datos	X_1	...	X_k
O_1'	$X_{1,1}$	...	$X_{1,k}$
...	...	...	...
O_n'	$X_{n,1}$	...	$X_{n,k}$

Esta tabla será una m.a.s. $O_1, \ldots, O_n$ (formada por n vectores aleatorios columna independientes e idénticamente distribuidos) de la variable aleatoria k dimensional $X = (X_1, \ldots, X_k)'$ que, en muchas ocasiones, podremos suponer normal. Sin embargo, otras veces prescindiremos de estas hipótesis y únicamente analizaremos una tabla de datos, tratando de condensar la información contenida en la misma y de analizar (de forma descriptiva) las relaciones entre las variables y los individuos.

Así, en la práctica, tendremos que la matriz de covarianzas V es desconocida, por lo que tendremos que estimarla y, una vez estimada, procederemos al cálculo de las componentes principales. De esta forma, las componentes principales (y los valores de la matriz T) dependerán de los valores muestrales y, por lo tanto serán v.a. (con individuos distintos, obtendremos componentes distintas) y lo mismo les ocurrirá a los valores propios (serán estimaciones de los verdaderos valores propios).

Para estimar V podemos utilizar la matriz de cuasicovarianzas muestrales S calculada como:

$$\begin{aligned} O_l &= (X_{l,1}, \ldots, X_{l,k})' \\ \overline{X}_j &= \frac{1}{n}\sum_{l=1}^{n} X_{l,j} \\ \overline{O} &= (\overline{X}_1, \ldots, \overline{X}_k)' = \frac{1}{n}\sum_{l=1}^{n} O_l \\ S &= \frac{1}{n-1}\sum_{l=1}^{n}(O_l - \overline{O})(O_l - \overline{O})' = (S_{i,j}) \\ S_{i,j} &= \frac{1}{n-1}\sum_{l=1}^{n}(X_{l,i} - \overline{X}_i)(X_{l,j} - \overline{X}_j). \end{aligned}$$

También podemos usar la matriz de covarianzas muestrales

$$\widehat{V} = \frac{n-1}{n} S.$$

Ambas tendrán los mismos vectores propios y, si n es grande, casi los mismos valores propios.

4.6.1. Cálculo a partir de una muestra

Como no conocemos V, la aproximaremos mediante S o $\widehat{V}$, las diagonalizaremos (calcularemos los ejes de sus elipsoides) y podremos calcular las componentes principales definidas como sigue.

Definición 4.4. *Llamaremos* ***componentes principales muestrales*** *a las variables $\widehat{Y} = \widehat{T}'X$, donde $\widehat{T}$ es la matriz ortogonal que diagonaliza S ($\widehat{V}$) y llamaremos* ***valores propios muestrales*** *$\widehat{\lambda}_j$ a los valores propios de S ($\widehat{V}$). Los valores de $\widehat{T}$ serán las* ***cargas*** *("loadings") o* ***coeficientes muestrales****. Llamaremos* ***puntuaciones muestrales*** *("scores") a los valores que obtendríamos para cada individuo en la componentes muestrales $P_{l,j} = Y_j(O_l) = \widehat{t}_j' O_l$.*

Si optamos por calcular las componentes principales a partir de la matriz de correlaciones, como también es desconocida, en su lugar se usará la matriz de correlaciones (de Pearson) muestrales

$$\begin{aligned} R &= diag(S)^{-1/2} S diag(S)^{-1/2} = (R_{i,j}) \\ R_{i,j} &= S_{i,j}(S_{i,i}S_{j,j})^{-1/2} = \widehat{V}_{i,j}(\widehat{V}_{i,i}\widehat{V}_{j,j})^{-1/2}. \end{aligned}$$

Esto equivaldría a estandarizar las variables iniciales restándoles sus medias muestrales y dividiéndolas por sus cuasivarianzas (es decir, hacer que todas tengan la misma variabilidad). En este caso, las puntuaciones se calcularán como:

$$P_{l,j} = Y_j(O_l) = \widehat{t}_j' O_l^*$$

donde $\widehat{t_j}$ es el vector propio j-ésimo de R y los datos estandarizados se obtienen (estiman) como

$$O_l^* = \left(\frac{X_{l,1} - \overline{X}_1}{S_1}, \ldots, \frac{X_{l,k} - \overline{X}_k}{S_k} \right)$$

siendo $S_i = \sqrt{S_{i,i}}$ la cuasidesviación típica de la variable X_i. La cuasidesviación típica S_j puede ser reemplazada por la desviación típica muestra $\widehat{V}_j = \sqrt{\widehat{V}_{j,j}}$ (como hace el programa R).

Si n es grande, ambas matrices ($\widehat{V}$ y S) son prácticamente iguales. Si X es normal, $\widehat{V}$ es máximo verosímil y S es insesgado para V, teniendo $(n-1)S$ una distribución (en el muestreo) Wishart $W_k(n-1, V)$. A partir de este resultado, se puede obtener la distribución exacta de los estimadores de los valores propios, pero ésta es bastante complicada. Si usamos $\widehat{V}$ y todos sus valores propios son distintos, se obtendrán estimadores máximo verosímiles para $t_{i,j}$ y λ_j ya que se verifica el resultado siguiente:

Proposición 4.8. *Si $\widehat{\theta}$ es máximo verosímil para θ, entonces $g(\widehat{\theta})$ es máximo verosímil para $g(\theta)$.*

Si X es normal, puede probarse que asintóticamente, la distribución conjunta de los estimadores de los valores propios es normal multivariante y que la de los estimadores de los valores $t_{i,j}$, también lo es, siendo ambas independientes entre sí.

Los valores asintóticos que se obtienen son los siguientes:

Teorema 4.3 (Anderson, 1974)**.** *Si X es normal con matriz de covarianzas con valores propios distintos y $O_1, \ldots, O_n$ es una m.a.s., entonces:*

1) $E(\widehat{\lambda}_j) = \lambda_j + \frac{1}{n} \sum_{i \neq j} \lambda_j \lambda_i (\lambda_j - \lambda_i)^{-1} + O(n^{-2})$
2) $Var(\widehat{\lambda}_j) = \frac{1}{n} 2\lambda_j^2 (1 - \frac{1}{n} \sum_{i \neq j} \lambda_i^2 (\lambda_j - \lambda_i)^{-2}) + O(n^{-3})$
3) $Cov(\widehat{\lambda}_i, \widehat{\lambda}_j) = O(n^{-2})$ si $i \neq j$
4) $\widehat{\lambda} \to_{n\to\infty} N_k(\lambda, 2D^2/n)$
5) $E(\widehat{t_j}) = t_j + O_k(n^{-1})$
6) $Cov(\widehat{t_j}) = \frac{1}{n} \sum_{i \neq j} \lambda_j \lambda_i (\lambda_j - \lambda_i)^{-2} t_j t_j' + O_{k,k}(n^{-2})$
7) $\widehat{t_j} \to_{n\to\infty} N_k(t_j, \frac{1}{n} \sum_{i \neq j} \lambda_j \lambda_i (\lambda_j - \lambda_i)^{-2} t_j t_j')$
8) $Cov(\widehat{t_i}, \widehat{t_j}) = -\frac{1}{n} \lambda_j \lambda_i (\lambda_j - \lambda_i)^{-2} t_i t_j' + O_{k,k}(n^{-2})$
9) $Cov(\widehat{\lambda}, \widehat{t_i}) \to_{n\to\infty} 0$

Las convergencias de variables aleatorias son convergencias en ley,

$$\lim_{n\to\infty} O(a_n)/a_n = cte$$

y $O_k(a_n)$ y $O_{k,k}(a_n)$ representan a un vector de dimensión k y a una matriz de dimensión $k \times k$ cuyos términos son $O(a_n)$, respectivamente.

Nótese que todos los estimadores son asintóticamente centrados y sus varianzas tienden a cero, siendo además $\widehat{\lambda}_i$ y $\widehat{\lambda}_j$ asintóticamente independientes (incorrelados). No ocurrirá esto si hay dos valores propios iguales ya que, entonces $(\lambda_j - \lambda_i)^{-1} \to \infty$.

4.6.2. Cálculo maximizando la varianza muestral

El cálculo de las componentes principales muestrales se puede enfocar de otra forma buscando la variable $a'X$ (combinación lineal de las originales) con $a'a = 1$ que aplicada a los individuos de la muestra nos de una variable con varianza (o cuasivarianza) muestral máxima. La **puntuación** o contador ("scores") del individuo j en esta nueva variable sería $a'O_j$, su media muestral sería

$$\frac{1}{n}\sum_{j=1}^{n} a'O_j = a'\frac{1}{n}\sum_{j=1}^{n} O_j = a'\overline{O}$$

y su cuasivarianza sería

$$\frac{1}{n-1}\sum_{j=1}^{n}(a'O_j - a'\overline{O})^2 = \frac{1}{n-1}\sum_{j=1}^{n} a'(O_j - \overline{O})(O_j - \overline{O})'a = a'Sa \tag{4.7}$$

cuyo máximo se alcanza si a es un vector propio del mayor de los valores propios de S. De forma análoga, se procedería para el cálculo de las restantes componentes principales muestrales. Si, por inducción, suponemos que los primeros $i-1$ vectores propios $\hat{t}_j$ de S nos dan las variables incorreladas con mayor varianza y buscamos maximizar la varianza muestral de $a'O_j$ (es decir $a'Sa$) para $a'a = 0$ haciendo que la covarianza muestral

$$\frac{1}{n-1}\sum_{j=1}^{n}(a'O_j - a'\overline{O})(\hat{t}_j'O_j - \hat{t}_j'\overline{O}) = a'S\hat{t}_j = \hat{\lambda}_j a'\hat{t}_j$$

sea cero para $j = 1, \ldots, i-1$. Escribiendo a en función de la base de vectores propios y procediendo como en el teorema principal se obtiene que el óptimo es $a = \hat{t}_i$.

De esta forma, podemos representar a los individuos mediante sus puntuaciones en las dos o tres primeras componentes manteniendo de ellos la mayor información (variabilidad o dispersión) posible (aunque $=_1, \ldots, O_n$ no sea una m.a.s.).

4.6.3. Cálculo minimizando las distancias cuadráticas

Geométricamente, el espacio formado por las m primeras componentes y que pasa por el punto $\overline{O}$ sería el espacio de dimensión m que minimiza la suma de las distancias al cuadrado de los individuos a dicho espacio (regresión perpendicular). De esta forma, el ACP sería como realizar una regresión mínimo cuadrática usando las distancias mínimas (regresión ortogonal) en lugar de las distancias "verticales" de la regresión clásica (para predecir Y en función de X).

Veamos que es cierto para $m = 1$. Para la primera componente, la suma de las distancias al cuadrado de los puntos O_j a la recta $\overline{O} + \lambda a$ vale

$$\sum_{j=1}^{n} d_j^2 = \sum_{j=1}^{n} d^2(O_j, \overline{O}) - p_j^2 = \sum_{j=1}^{n}(O_j - \overline{O})'(O_j - \overline{O}) - \sum_{j=1}^{n} a'(O_j - \overline{O})(O_j - \overline{O})'a,$$

donde p_j es la proyección del vector $O_j - \overline{O}$ sobre la recta $\overline{O} + \lambda a$. El mínimo para $a'a = 1$ coincide con el máximo de (4.7), es decir, se alcanza haciendo que a sea el primer vector propio muestral. Si consideramos cualquier otra recta paralela $P + \lambda a$, se tendrá

$$\sum_{j=1}^{n} d_j^2 = \sum_{j=1}^{n} d^2(O_j, P) - p_j^2 = \sum_{j=1}^{n} (O_j - P)'(O_j - P) - \sum_{j=1}^{n} a'(O_j - \overline{O})(O_j - \overline{O})'a,$$

ya que eligiendo P de forma adecuada podemos conseguir que las proyecciones sean las mismas. Entonces, de nuevo su mínimo para $a'a = 1$ coincide con el máximo de (4.7). Por último, si queremos minimizar

$$\sum_{j=1}^{n} (O_j - P)'(O_j - P)$$

en P tenemos que

$$\begin{aligned}
\sum_{j=1}^{n} d_j^2 &= \sum_{j=1}^{n} (O_j - P)'(O_j - P) \\
&= \sum_{j=1}^{n} (O_j - \overline{O} + \overline{O} - P)'(O_j - \overline{O} + \overline{O} - P) \\
&= n(\overline{O} - P)'(\overline{O} - P) + \sum_{j=1}^{n} (O_j - \overline{O})'(O_j - \overline{O}) + 2\sum_{j=1}^{n} (\overline{O} - P)'(O_j - \overline{O}) \\
&= n(\overline{O} - P)'(\overline{O} - P) + \sum_{j=1}^{n} (O_j - \overline{O})'(O_j - \overline{O}) + 2(\overline{O} - P)'\sum_{j=1}^{n} (O_j - \overline{O}) \\
&= n(\overline{O} - P)'(\overline{O} - P) + \sum_{j=1}^{n} (O_j - \overline{O})'(O_j - \overline{O}),
\end{aligned}$$

donde el segundo sumando es constante y el mínimo del primer sumando se alcanza con $P = \overline{O}$ (ya que es positivo).

4.7. Análisis de componentes principales en R

Veamos como realizar una ACP (PCA en inglés) en R sobre un conjunto de datos. Los conjuntos de datos disponibles en R en el "paquete" *Datasets* se pueden ver mediante: `data()`

Otros "paquetes" incluyen otros ficheros de datos. Consideraremos los datos del Ejemplo 4.1 denominados `LifeCycleSavings`. Recordemos que para ver estos datos en R debemos hacer:

```
LifeCycleSavings
```

y para guardarlos en `d`

```
d<-LifeCycleSavings
```

Como ya hemos comentado el fichero contiene 5 variables medidas en 50 países diferentes (ver Tabla 4.7).

Tabla 4.7: Primeros datos del fichero `LifeCycleSavings`.

	sr	pop15	pop75	dpi	ddpi
Australia	11.43	29.35	2.87	2329.68	2.87
Austria	12.07	23.32	4.41	1507.99	3.93
Belgium	13.17	23.80	4.43	2108.47	3.82
Bolivia	5.75	41.89	1.67	189.13	0.22
Brazil	12.88	42.19	0.83	728.47	4.56

Los detalles sobre estos datos se pueden ver tecleando:

```
help(LifeCycleSavings)
```

donde se indica que:

sr: incremento de los ahorros personales 1960-1970 (crecimiento ahorros dividido por ingresos)
pop15: % población menor de 15 años.
pop75: % población mayor de 75.
dpi: ingresos per-capita.
ddpi: crecimiento del dpi 1960-1970.

4.7.1. Análisis inicial de los datos

Antes de aplicar cualquier técnica estadística es conveniente hacer un estudio preliminar de los datos. Podemos empezar estudiando las variables por separado para detectar valores atípicos, falta de simetría o normalidad, etc. En realidad podemos usar las técnicas que consideremos oportunas. Para resumir estas variables podemos hacer:

```
summary(d)
```

obteniendo medias, medianas, cuartiles, mínimos y máximos (ver Tabla 4.8).

Para obtener los valores de la primera variable `sr` podemos hacer: `d$sr` o `d[,1]`. Podemos dibujarlos con:

Tabla 4.8: Principales características (media, mediana, etc.) de todas las variables del fichero de R `LifeCycleSavings`.

	sr	pop15	pop75	dpi	ddpi
Min.	0.600	21.44	0.560	88.94	0.220
1st Qu.	6.970	26.21	1.125	288.21	2.002
Median	10.510	32.58	2.175	695.66	3.000
Mean	9.671	35.09	2.293	1106.76	3.758
3rd Qu.	12.617	44.06	3.325	1795.62	4.478
Max.	21.100	47.64	4.700	4001.89	16.710

```
plot(d[,1])
```

o con:

```
boxplot(d[,1])
```

(ver Figura 4.7). En este caso no se observan ni datos atípicos, ni falta de simetrías y tendencias (falta de aleatoriedad).

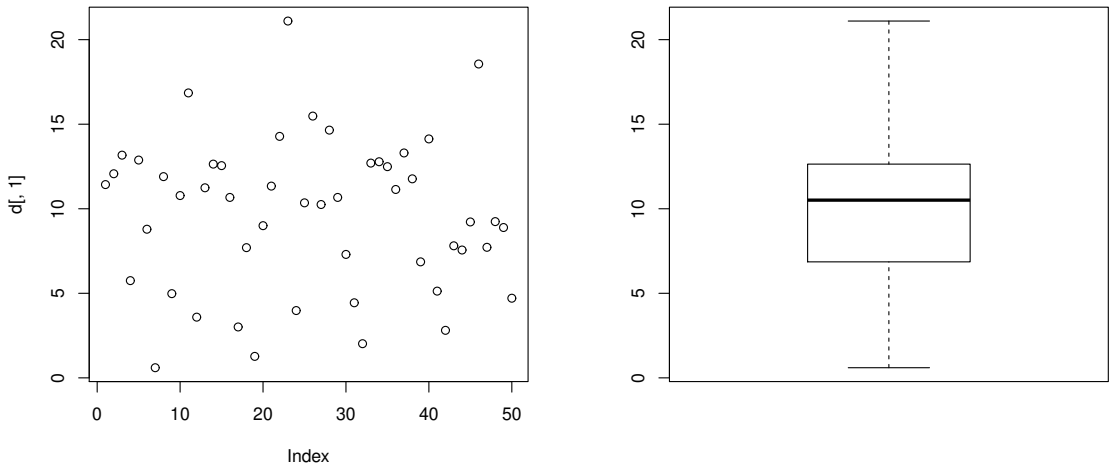


Figura 4.7: Datos y gráfico caja-bigote de la primera variable del fichero `LifeCycleSavings`.

Otra opción interesante son los histogramas que se pueden realizar con `hist(d$sr)` o con `hist(d[,1])` (ver Figura 4.8). Estos gráficos nos permiten estudiar simetrías y normalidad. En este caso se observa asimetría y falta de normalidad (el gráfico no se parece a la campana de Gauss).

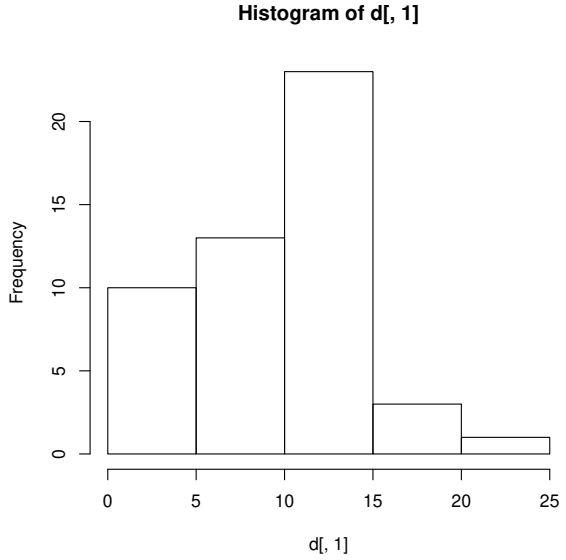


Figura 4.8: Histograma de la primera variable del fichero `LifeCycleSavings`.

Para estudiar las relaciones entre las variables podemos hacer todos los gráficos por parejas mediante

```
boxplot(d)
plot(d)
```

Las gráficas se pueden ver en las Figura 4.9. En la primera se observa que una variable (dpi) tiene mucha mas dispersión que las demás (esto es muy importante para decidir si hacemos un PCA con la matriz de covarianzas o la de correlaciones). En la segunda se aprecia que hay variables poco relacionadas con las demás (por ejemplo, sr) y que en algunas variables hay valores atípicos. Por ejemplo, en `ddpi` hay dos países con un valores muy altos (Jamaica 10.23 y Libya 16.71). Podemos usar los comandos `which.max(d[,5])`, `sort(d[,5])` o `order(d[,5])` para detectar los valores extremos.

Podemos calcular las matrices de covarianzas y correlaciones con `cov(d)` y `cor(d)`, respectivamente. Las correlaciones se pueden ver con `View(cor(d))` (ver Tabla 4.9). Se aprecia que existen variables con correlaciones (lineales) positivas, negativas y casi nulas. Estas relaciones se verán reflejadas en las componentes principales.

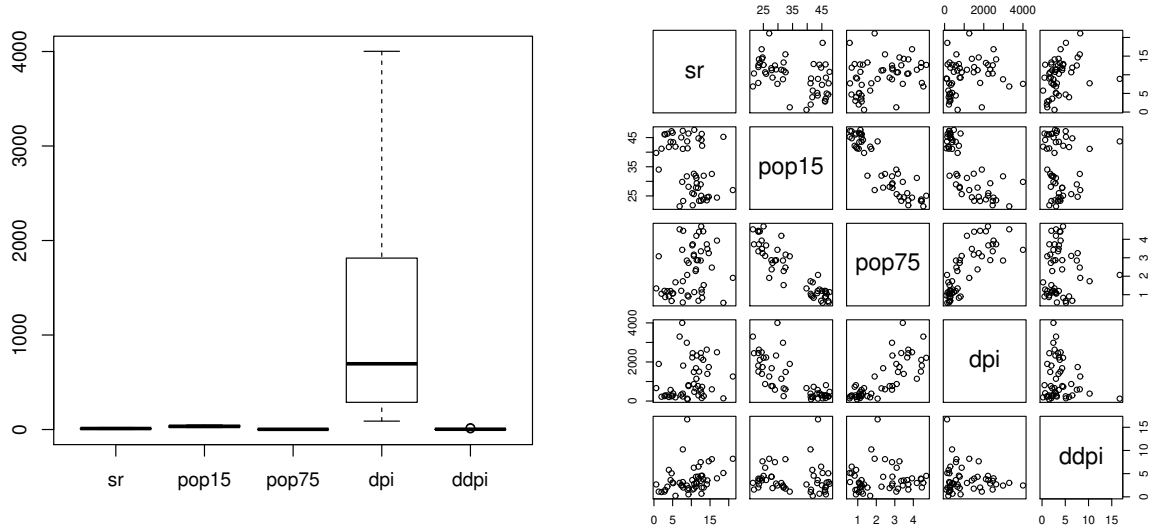


Figura 4.9: Gráficos caja-bigote (izquierda) y bidimensionales (derecha) de los datos de todas las variables del fichero de R `LifeCycleSavings`.

Tabla 4.9: Correlaciones entre las 5 variables del fichero `LifeCycleSavings`.

	sr	pop15	pop75	dpi	ddpi
sr	1.0000000	-0.45553809	0.31652112	0.2203589	0.30478716
pop15	-0.4555381	1.00000000	-0.90847871	-0.7561881	-0.04782569
pop75	0.3165211	-0.90847871	1.00000000	0.7869995	0.02532138
dpi	0.2203589	-0.75618810	0.78699951	1.0000000	-0.12948552
ddpi	0.3047872	-0.04782569	0.02532138	-0.1294855	1.00000000

En las gráficas anteriores y en la matriz de covarianzas se aprecia que las variables se miden en unidades muy diferentes (recuerde que en la diagonal aparecen las cuasivarianzas) por lo que el ACP se deberá realizar sobre la matriz de correlaciones (es decir, el programa lo hará sobre las v.a. estandarizadas).

4.7.2. Cálculo de las componentes principales

En el programa R disponemos de dos comandos diferentes para calcular las componentes principales: `princomp` y `prcomp`. Como las componentes son únicas (salvo cambio de signo) cuando los valores propios estimados son distintos, los resultados serán muy similares, pero puede haber

pequeñas diferencias debidas a los métodos numéricos usados para su cálculo.

Para hacer un PCA con `princomp` usando la matriz de correlaciones de los datos guardados en `d` basta teclear:

```
PCA<-princomp(d,cor=TRUE)
```

Para ver las características principales haremos:

```
summary(PCA,loadings=TRUE)
```

obteniendo:

Importance:	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5
Stan. dev.	1.6799041	1.1207437	0.777512	0.4895354	0.278721
Prop. Var.	0.5644156	0.2512133	0.120905	0.0479299	0.015537
Cum. Prop.	0.5644156	0.8156289	0.936534	0.984463	1

Loadings:	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5
sr	0.308	0.554	0.750	-0.130	-0.134
pop15	-0.571			-0.416	-0.707
pop75	0.560	-0.101	-0.212	0.390	-0.692
dpi	0.514	-0.266	-0.145	-0.801	
ddpi		0.782	-0.609	-0.123	

La importancia de las componentes se mide con sus desviaciones estándar (raíces cuadradas de los valores propios de la matriz de correlaciones ordenados de mayor a menor) estimadas, la proporción en tanto por uno de sus varianzas estimadas y las proporciones acumuladas. En este caso, las varianzas iniciales suman 5 (la traza de la matriz de correlaciones muestral) por lo que el primer valor de las proporciones 0.5644156, se calcula como:

```
(1.6799041^2)/5
```

Las proporciones acumuladas se calculan sumando las de las componentes anteriores. Por ejemplo, $0.8156289 = 0.5644156 + 0.2512133$. Estos valores nos indican que la primera componente mantiene un 56.44156 % de la información inicial, la segunda un 25.12133 % y las dos juntas un 81.56289 %.

Las cargas (*loadings*) son los vectores propios unitarios de los valores propios anteriores. Los valores ausentes son números pequeños (pero no necesariamente cero). Si queremos guardarlos podemos hacer:

```
PCA$loadings->T
```

De esta forma, tecleando `T[,1]` obtenemos el primer vector propio que se utiliza para calcular la primera componente principal. Su último coeficiente es 0.03787232 (pequeño pero no cero). Se puede comprobar que este vector tiene norma 1. Por lo tanto la primera componente principal (muestral) se calcularía como:

$$\hat{Y}_1 = 0.308 * X_1^* - 0.571 * X_2^* + 0.560 * X_3^* + 0.514 * X_4^* + 0.0379 * X_5^* \tag{4.8}$$

(los coeficientes se han redondeado), donde $X_i^* = (X_i - mean(X_i))/sd(X_i)$ es la variable i-ésima estandarizada (muestral). Si usamos la matriz de covarianzas para el PCA, en esta fórmula se usarán las variables originales.

Análogamente, las puntuaciones (*scores*), es decir, los valores que obtendrían los individuos de la muestra (países en este caso) en las componentes principales (usando esas fórmulas), se pueden calcular mediante:

```
PCA$scores->S
```

Tecleando `S` comprobamos que, por ejemplo, la puntuación en la primera componente de Australia es 1.36528994. El significado de estos valores se verá posteriormente.

Para comprobar que los valores obtenidos con `princomp` son los correctos podemos hacer: `eigen(cor(d))` con lo que obtenemos los valores propios ordenados (varianzas de las componentes) y los vectores propios (cargas o coeficientes). Para calcular las puntuaciones debemos primero estandarizar (por columnas) los datos iniciales (este paso no será necesario cuando usemos la matriz de covarianzas). Para ello usaremos:

```
z<-scale(d)
```

y para calcular las puntuaciones de la primera componente haremos:

```
y1<-0.30846174*z[,1]-0.57065322*z[,2]+0.56043119*z[,3]+0.51350640*z[,4]
   +0.03787232*z[,5]
```

De esta forma, para Australia obtenemos 1.35156808, que es similar al valor obtenido antes.

Las componentes principales se pueden calcular aunque no se dispongan de los datos completos usando únicamente la matriz de correlaciones (o covarianza). Para ello haremos:

```
princomp(covmat=cor(d))
```

sustituyendo `cor(d)` por la matriz de correlación (o covarianzas) de los datos. Las cargas se calcularán como antes pero, en este caso, no podremos calcular las puntuaciones y no podremos hacer los gráficos de las componentes principales (objetos).

Para calcular las componentes principales con `prcomp` usando la matriz de correlaciones debemos hacer:

```
PCAbis<-prcomp(d,scale=TRUE)
```

Haciendo `summary(PCAbis)` se obtiene la importancia de las componentes, con `PCAbis` las cargas y con `PCAbis$x` las puntuaciones (usa las cuasivarianzas). Nótese que los valores son similares pero que se han cambiado de signo las dos primeras componentes (su interpretación será opuesta). Compruebe que se obtienen resultados totalmente diferentes (erróneos en este caso) si usamos la matriz de covarianzas tecleando `princomp(d)` (procure no alterar los objetos usados anteriormente ya que se usarán en las secciones siguientes).

4.7.3. Análisis de las componentes principales

Para analizar las componentes principales calculadas en la sección anterior primero debemos fijarnos en la importancia de cada una. Hablaremos posteriormente sobre el número adecuado de componentes pero, antes de analizarlas, debemos tener en cuenta que la primera tiene un 56.44156 % de la información inicial, la segunda un 25.12133 % y las dos juntas un 81.56289 %. Por lo tanto, en este caso, la información proporcionada por la primera será en general el doble de importante que la que proporciona la segunda, etc. Esto es un cómputo global por lo que puede haber variables que estén mejor representadas en Y_2 que en Y_1 (por ejemplo `ddpi`).

En segundo lugar miraremos las cargas (loadings) o coeficientes de las componentes que queremos analizar para poder dar un significado a estas variables nuevas denominadas componentes principales. Si miramos las cargas de Y_1 dadas en (4.8) y guardadas en `T[,1]`, teniendo en cuenta que las variables están estandarizadas (y tendrán valores similares), podemos afirmar que las variables que más influyen en $Y1$ son (por orden de influencia): pop15 (negativa), pop75 (positiva), dpi (positiva) y sr (positiva). Por lo tanto, Y_1 tomará valores grandes en los países con valores pequeños en pop15 y grandes en las otras tres. Por lo tanto, Y_1 nos indicará los países que tienen poblaciones envejecidas (alta pop75 y baja pop15) y ricos (altos valores en dpi y sr). Estas suelen ser características de países muy desarrollados.

Una vez que ya hemos interpretado una componente, podemos analizar sus puntuaciones para decir cómo serán (aproximadamente) los individuos de la muestra según los valores que toman en esa componente. Usando `summary(S[,1])` y/o `plot(S[,1])` (ver Figura 4.10) observamos que los valores de la muestra en Y_1 están entre -2.258755 y 2.787708 (su media es cero ya que hemos usado variables estandarizadas). Haciendo `which.max(S[,1])` comprobamos que el valor mayor en Y_1 corresponde a Suecia (2.787708) y el menor a Malasia (-2.258755). Por lo tanto, Australia con 1.36528994 sería, en aquella época, un país bastante desarrollado y España con 0.69294913 estaría un poco por encima de la media (ver Figura 4.10). En esta gráfica se observa que casi no hay valores entre 0 y -1, por lo que la mayoría de los países se podrían clasificar como del tercer o del primer mundo (en esa época). Analice la segunda componente y estudie las puntuaciones de estos países en esa componente.

Como las componentes son incorreladas (independientes si las variables iniciales son normales), las podemos estudiar por separado. Sin embargo, muchas veces resulta conveniente representarlas

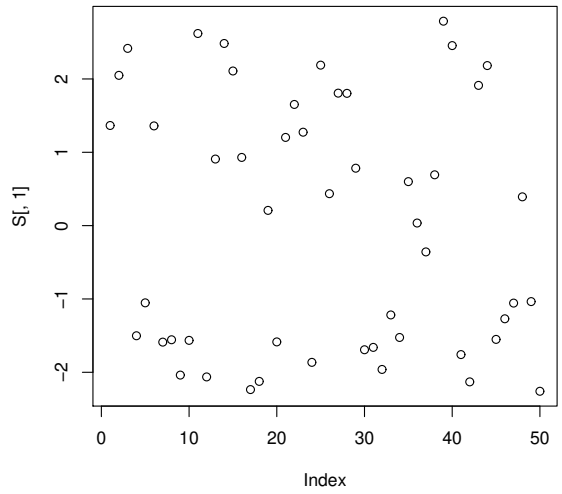


Figura 4.10: Gráfico de puntuaciones para la primera componente principal.

por parejas en gráficos bidimensionales. Para representar las cargas y las puntuaciones de las dos primeras componentes haremos:

```
biplot(PCA,pc.biplot=TRUE)
```

El resultado puede verse en la Figura 4.11. Las cargas aparecen como vectores en rojo con las escalas en la derecha y arriba y las puntuaciones en negro con las etiquetas de los datos (nombres de los países) con las escalas abajo (Y_1) y en la izquierda (Y_2). Las puntuaciones en el gráfico se reescalan para tener varianza 1 (componentes principales estandarizadas).

Este gráfico es la ('mejor') proyección bidimensional ('foto') de los ejes iniciales de las cinco variables estandarizadas y de las puntuaciones de los individuos (países) de la muestra. Las cargas de este gráfico se pueden usar (igual que antes) para interpretar las componentes. Las variables con vectores largos (norma cercana a 1) estarán bien representadas por las dos primeras componentes, mientras que las que tengan vectores cortos estarán mal representadas (se pierden al proyectar por ser casi perpendiculares). En este ejemplo todas las variables están bien representadas en este gráfico. Además, se aprecia que `pop75`, `dpi` y, en menor medida, `sr`, hacen crecer la primera componente, mientras que `pop15` la hace disminuir y `ddpi` no influye en ella. Lógicamente, la interpretación de Y_1 es la misma que antes. También podemos observar que la segunda componente crece cuando crece `ddpi` (incremento ingresos per-capita) y en menor medida `sr` (incremento de los ahorros personales), decrece un poco si crece `dpi` y casi no se ve afectada por el envejecimiento de la población. Por lo tanto se puede interpretar como un índice del crecimiento de los países en esa década (los valores grandes corresponderán a los países que más han crecido).

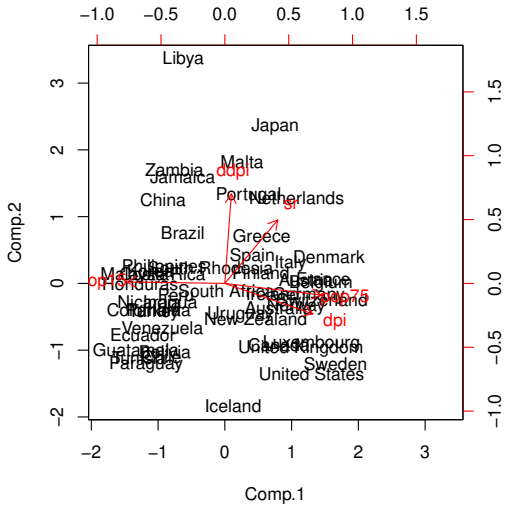


Figura 4.11: Gráfico de las dos primeras componentes principales estandarizadas.

Las puntuaciones se usarán para decir cómo serán (aproximadamente) los individuos de la muestra (países) en esas características. A la derecha tendremos a los países más desarrollados y con poblaciones envejecidas (Suecia, US, etc.) a la izquierda lo contrario (Honduras, Guatemala, etc.), arriba a los países que más se desarrollaron durante esa época (Líbia, Japón, etc.) y debajo los que menos (Islandia, US, Paraguay, etc.). También nos podemos fijar en una variable concreta. Por ejemplo, con respecto a `sr` podríamos decir que los países con mayores incrementos de los ahorros personales (valores sr) deberían ser Japón, Malta y Holanda. Si vemos los datos de `sr` (con `d` y `sort(d[,1])`) podemos comprobar que efectivamente Japón es el que tienen un valor mayor (21.10) pero que el segundo es Zambia (18.56). Es lógico que al proyectar las variables originales, se pierda algo de la información contenida en ellas.

Como las etiquetas de los datos (nombres de los países) son muy grandes, algunos de ellos no se aprecian bien en el gráfico. Para sustituirlos por sus números de línea podemos hacer:

```
biplot(PCA,pc.biplot=TRUE,xlabs=1:50)
```

Si queremos hacer un gráfico de las componentes tercera y cuarta haremos:

```
biplot(PCA,pc.biplot=TRUE,choices=c(3,4),xlabs=1:50)
```

También podemos hacer un gráfico solo de las puntuaciones de las dos primeras componentes (sin estandarizar) con:

```
plot(S[,1],S[,2],xlab='Y1',ylab='Y2')
```

Para encontrar un individuo (país) bastar mirar sus puntuaciones. La mayor dispersión (varianza) de la primera componente indica que ésta es más importante (tiene más información) a la hora de distinguir los datos. Además podemos localizar cualquier país usando sus puntuaciones. Por ejemplo, encuentre España y diga como serán sus medidas según su posición en el gráfico. Para poner una etiqueta al dato $i = 38$ haremos:

```
text(S[38,1],S[38,2],labels='Esp')
```

Las cargas se pueden representar de forma similar. Aunque no son muy habituales, las tres primeras componentes se podrían representar en gráficos 3D. Es mucho mejor realizar el gráfico siguiente que contiene las tres primeras componentes (sin estandarizar):

```
pairs(PCA$scores[,1:3])
```

4.7.4. Saturaciones

Para medir las relaciones lineales entre las variables iniciales y las componentes principales, se puede calcular la matriz de correlaciones conocida como matriz de saturaciones mediante:

$$Corr(X_i, Y_j) = \frac{t_{i,j}}{\sigma_i} \lambda_j^{1/2}.$$

En la práctica trabajaremos con sus estimaciones. Si el PCA se ha realizado con la matriz de correlaciones (o si todas las varianzas son 1) bastará con multiplicar la columna de los coeficientes (cargas) de cada componente principal por la raíz cuadrada de su valor propio (su desviación estándar). Las saturaciones al cuadrado nos indicarán cuánta información (en tanto por 1) tendrá cada componente de cada variable. En nuestro ejemplo, las saturaciones de la primera componente principal se calcularán con:

```
S1<-T[,1]*1.6799041
```

y las saturaciones al cuadrado con `S1^2` obteniendo los valores de la Tabla 4.10.

Tabla 4.10: Saturaciones de la primera componente principal.

	sr	pop15	pop75	dpi	ddpi
Sat.	0.5181861	-0.9586427	0.9414706	0.8626415	0.0636219
Inf.	0.26851688	0.91899580	0.88636698	0.74415038	0.00404774

De esta forma comprobamos que la variable mejor representada en Y_1 es *pop15* con un 91.89958 % y que de la última variable Y_1 prácticamente no tiene información.

Para calcular todas las saturaciones (usemos o no la matriz de correlaciones) podemos hacer:

```
SAT<-cor(d,S)
```

obteniendo:

Sat.	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5
sr	0.5181861	0.6211673	0.5832461	-0.0637125	-0.03740324
pop15	-0.9586427	0.0142035	0.0206425	-0.2037487	-0.19713655
pop75	0.9414706	-0.1131882	-0.1647000	0.1911276	-0.19278378
dpi	0.8626415	-0.2984564	-0.1127514	-0.3922864	0.01310945
ddpi	0.0636219	0.8764293	-0.4733754	-0.0604464	0.00927110

Como las componentes son incorreladas, las correlaciones múltiples al cuadrado o comunalidades serán la suma de las correlaciones al cuadrado:

$$Corr^2(X_i,(Y_1,\ldots,Y_P)) = \sum_{j=1}^{p} Corr^2(X_i,Y_j).$$

Estos valores nos indicarán la información (en tanto por 1) que matienen las p primeras componentes sobre cada variable. Por ejemplo, si decidimos usar las dos primeras componentes, las correlaciones multiples se calcularán mediante:

```
SAT[,1]^2+ SAT[,2]^2
```

obteniendo:

Inf.	Comp.1	Comp.2	Comunalidad
sr	0.268516885	0.38584877	0.6543657
pop15	0.918995810	0.00020174	0.9191976
pop75	0.886366987	0.01281156	0.8991785
dpi	0.744150387	0.08907624	0.8332266
ddpi	0.004047742	0.76812824	0.7721760

Se observa que la variable mejor representada por las dos primeras componentes (gráfico *biplot*) es `pop15` de la que se mantiene un 91.91976 % de su información y la peor representada es `sr` con un 65.43657 % (no es necesario usar tantos decimales). Se puede comprobar que la media de los valores de la última columna es 0.8156289 que coincide con la información que (en promedio) mantienen Y_1 y Y_2 (calculada anteriormente con `summary(PCA)`). Compruebe que ocurre lo mismo con las

informaciones individuales de cada componente. También se puede comprobar que la suma de los valores de cada columna nos dan los valores propios (informaciones) de cada componente principal (solo si usamos la matriz de correlaciones). Por ejemplo, compruebe que sumando los valores de la primera obtenemos 2.822078, es decir, el mayor valor propio de la matriz de correlaciones.

La correlación múltiple al cuadrado $Corr^2(X_i, (Y_1, \ldots, Y_p))$ es el máximo de las correlaciones que se pueden obtener con combinaciones lineales de las componentes $Y_1, \ldots, Y_p$. Además, el máximo de esas correlaciones se obtiene con los coeficientes incluidos en la matriz T. Por ejemplo, la mejor combinación lineal de las dos primeras componentes para aproximar *sr* es la que se obtiene cortando el vector fila `T[1,]`, es decir, $Z_1 = 0.3084617 * Y_1 + 0.5542456 * Y_2$. De esta forma, si calculamos Z_1 con:

```
Z1<-0.3084617*S[,1]+ 0.5542456*S[,2]
```

y calculamos `cor(d[,1],Z1)^2`, se obtiene 0.6543657 que coincide con la información que mantienen esas dos componentes sobre `sr` (correlación al cuadrado máxima). La variable Z_1 se podría usar para predecir `sr` usando las técnicas de regresión lineal vistas en las prácticas anteriores.

4.8. Número de componentes

Una vez realizado un PCA podemos preguntarnos con cuántas componentes principales debemos quedarnos. Las respuesta no es única y puede depender de factores subjetivos. Todas las soluciones serán correctas ya que lo que estamos haciendo es perder algo de información (la menor posible) a cambio de reducir la dimensión (número de variables) inicial. A continuación comentamos algunas de las técnicas más usadas. En todos ellas el número de componentes elegidas se representará por m y, lógicamente, se tomarán siempre las m primeras componentes principales (ya que son las que más información tienen).

4.8.1. Fijar un número concreto de componentes

Una opción válida es fijar un número de componentes concreto. Por ejemplo, si queremos hacer una única gráfica bidimensional, evidentemente debemos tomar $m = 2$, con lo que únicamente analizaremos Y_1 e Y_2. En esta opción es fundamental informar de la información total mantenida por las componentes elegidas y advertir si ese número es bajo. Se suelen tomar números pares de componentes para poder realizar gráficas bidimensionales y el valor más usual es $m = 2$.

Tecleando `summary(PCA)` comprobamos que, en nuestro ejemplo, si tomamos $m = 2$, mantendríamos $p = 81.56\,\%$ de la información inicial lo que podemos considerar como aceptable al reducir la dimensión de 5 a 2. También podríamos informar sobre las comunalidades, es decir, sobre la información mantenida por esas componentes de cada variable (ver sección anterior). En nuestro ejemplo, para $m = 2$, la variable peor representada es *sr* de la que mantienen un $65.44\,\%$. Por lo

tanto, todas las variables están bien representadas. En otros ejemplos nos podremos encontrar con variables que no están representadas en las componentes elegidas. En estos casos es importante señalarlo.

4.8.2. Fijar un porcentaje mínimo de información mantenida

Si queremos mantener un porcentaje p de la variabilidad inicial deberemos quedarnos con las primeras m componentes que verifiquen

$$\frac{\widehat{\lambda}_1 + \cdots + \widehat{\lambda}_m}{\widehat{\lambda}_1 + \cdots + \widehat{\lambda}_k} \geq \frac{p}{100},$$

donde $\widehat{\lambda}_i$ representan las estimaciones de los valores propios. En nuestro ejemplo, si queremos mantener más de un $p = 90\,\%$, debemos tomar $m = 3$ con lo que mantendríamos un $93.65\,\%$.

Otra regla (diferente) podría ser el fijar un porcentaje mínimo para las comunalidades. De esta forma nos aseguramos de que todas las variables originales (sean importantes o no), estén representadas en las componentes. En nuestro ejemplo, si queremos que las comunalidades sean mayores que 0.5 (es decir queremos mantener al menos un $50\,\%$ de todas las variables), debemos tomar $m = 2$. Nótese que con esta regla, en nuestro ejemplo, nunca obtendríamos $m = 1$ a pesar de que Y_1 tiene un $56\,\%$ de la información total.

4.8.3. Regla de Rao

Esta regla establece que solo serán relevantes las componentes que tengan una variabilidad (varianza o valor propio) mayor que la variabilidad mínima de las variables originales. De esta forma, se tiene

$$\text{máx}\, m : \widehat{\lambda}_m \geq \text{mín}\{S_j^2\},$$

donde S_j^2 representan a las cuasivarianzas muestrales de las variables originales. Si las componentes se calculan usando la matriz de correlaciones, como esto es equivalente a usar las variables estandarizadas, se entiende que las varianzas son 1 y, por lo tanto, se toman solo las componentes con valores propios (varianzas o desviaciones típicas) mayores que uno. En nuestro ejemplo, esta regla nos conduce a $m = 2$ ya que

$$\widehat{\lambda}_2 = 1.256066 > 1 > \widehat{\lambda}_3 = 0.6045255.$$

Si calculásemos las componentes con la matriz de covarianzas (aunque ya hemos comentado que esto no sería correcto), el mínimo de las cuasivarianzas muestrales corresponde a la variable `pop75` y vale 1.66609082 (hacer `var(d$pop75)` o `cov(d)`) y los valores propios de la matriz de covarianzas valen: 981871.2, 43.14338, 13.68328, 6.629537 y 0.2351568 por lo que con este criterio tomaríamos $m = 4$.

4.8.4. Regla de Kaiser

Esta regla es similar a la anterior y establece que solo serán relevantes las componentes que tengan una variabilidad mayor que la variabilidad media de las variables originales. De esta forma, se tiene

$$\text{máx}\, m : \widehat{\lambda}_m \geq \frac{1}{k}\sum_{j=1}^{k} S_j^2.$$

Si usamos la matriz de correlaciones para calcular las componentes, como las varianzas iniciales son 1, su media es 1 y este criterio coincide con el de Rao, por lo que, en nuestro ejemplo, obtenemos el mismo resultado $m = 2$. Si calculásemos las componentes con la matriz de covarianzas (aunque ya hemos comentado que esto no sería correcto con estos datos), la media de las cuasivarianzas muestrales es 196387 y los valores propios de la matriz de covarianzas valen: 981871.2, 43.14338, 13.68328, 6.629537 y 0.2351568 por lo que con este criterio tomaríamos $m = 1$.

4.8.5. Regla del codo o del gráfico de sedimentación

Es uno de los métodos más usados y suele ir incluido en casi todos los programas de estadística. El método consiste en representar j (eje x) frente a los valores propios estimados $\widehat{\lambda}_j$ obteniéndose el denominado gráfico de sedimentación o desmoronamiento (*scree graph*). El gráfico será similar a la acumulación de sedimentos en la ladera de una montaña (cono de desmoronamiento). Se trataría de separar "la montaña" de los "sedimentos". La regla establece que serán representativas las componentes hasta el primer "codo" (sin incluirlo) de la gráfica o hasta que comience la línea recta aproximada final. Para realizar este gráfico en R haremos:

```
screeplot(PCA)
```

con lo que se obtiene el gráfico de la Figura 4.12. Se puede obtener un gráfico similar mediante:

```
plot(eigen(cor(d))$values,type='l',ylab='valores propios')
```

En estos gráficos, aunque no está muy claro, parece que el codo (los sedimentos) se encuentra en $j = 3$, por lo que tomaríamos las dos primeras componentes ($m = 2$). Las soluciones $m = 1$ y $m = 3$ también serían aceptables. En otras ocasiones el "codo" aparece más claro y solo hay una opción para m con esta regla.

4.8.6. Test de esfericidad

Esta regla se basa en la contrastación de la hipótesis $H_0 : \lambda_{m+1} = \cdots = \lambda_k$, cuyo significado es que para un m dado, las componentes restantes tienen igual variabilidad teórica (las diferencias en las varianzas muestrales se deben al azar) y, por lo tanto, no debemos dar preferencia a una sobre otra (también se dice que hay "esfericidad" en $Y_{m+1}, \ldots, Y_k$). El test de esfericidad de Bartlett se

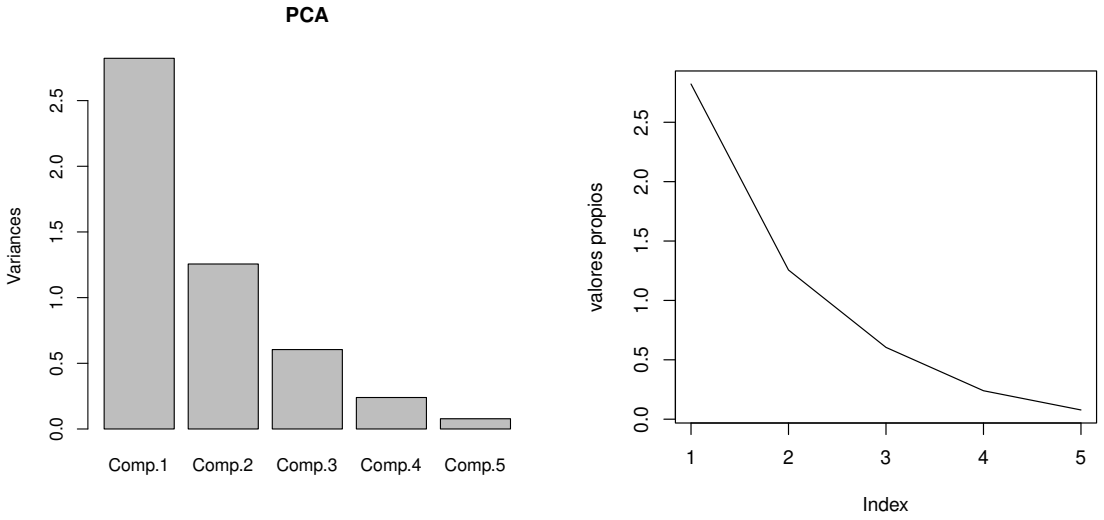


Figura 4.12: Gráfico de sedimentación (screeplot).

basa en el test de razón de verosimilitudes que da el estadístico:

$$T = \left(n - \frac{2k+11}{6}\right)(k-m)\ln\left(\frac{m_a}{m_g}\right),$$

donde $m_a = \frac{1}{k-m}\sum_{i=m+1}^{k}\widehat{\lambda}_i$ y $m_g = (\prod_{i=m+1}^{k}\widehat{\lambda}_i)^{1/(k-m)}$ (medias aritmética y geométrica de los últimos valores propios). En condiciones de normalidad de los datos iniciales y cuando H_0 es cierta, T sigue una distribución chi-cuadrado χ^2_{gl} con $gl = 0.5(k-m-1)(k-m+2)$ grados de libertad. Si H_0 no es cierta, T tiende a tomar valores mayores por lo que la región de rechazo sería de la forma $T > \chi^2_{1-\alpha,gl}$, donde $\chi^2_{1-\alpha,gl}$ es el cuantil $1-\alpha$ de esa distribución chi-cuadrado.

Para aplicar este test a nuestro ejemplo con $m = 2$ calcularemos

```
eigen(cor(d))
(0.60452546+0.23964496+0.07768522)/3->ma
(0.60452546*0.23964496*0.07768522)^(1/3)->mg
(50-(2*5+11)/6)*(5-2)*log(ma/mg)->T
0.5*(5-2-1)*(5-2+2)->gl
1-pchisq(T,gl)
```

obteniendo:

$$
\begin{aligned}
m_a &= \frac{1}{k-m}\sum_{i=m+1}^{k}\widehat{\lambda}_i = 0.3072852\\
m_g &= \left(\prod_{i=m+1}^{k}\widehat{\lambda}_i\right)^{1/(k-m)} = 0.2240993\\
T &= \left(n-\frac{2k+11}{6}\right)(k-m)\ln\left(\frac{m_a}{m_g}\right) = 44.03837\\
gl &= 0.5(k-m-1)(k-m+2) = 5\\
P-valor &= \Pr(\chi^2_5 > 44.03837) = 2.27505\cdot 10^{-8}
\end{aligned}
$$

y, como el P-valor obtenido es muy pequeño (menor que 0.05), rechazaremos la esfericidad de las tres últimas componentes (H_0) por lo que, si queremos, podemos calcular más componentes (y éstas no serán al azar). La región crítica (de rechazo) para este test con $\alpha = 0.05$ es $(11.0705, \infty)$ donde $11.0705 = \chi^2_{0.95,5}$ se calcula en R mediante `qchisq(0.95,gl)`. La gráfica de la función de densidad de una χ^2_5 se puede obtener con:

```
curve(dchisq(x,5),0,50)
```

obteniéndose la gráfica de la Figura 4.13.

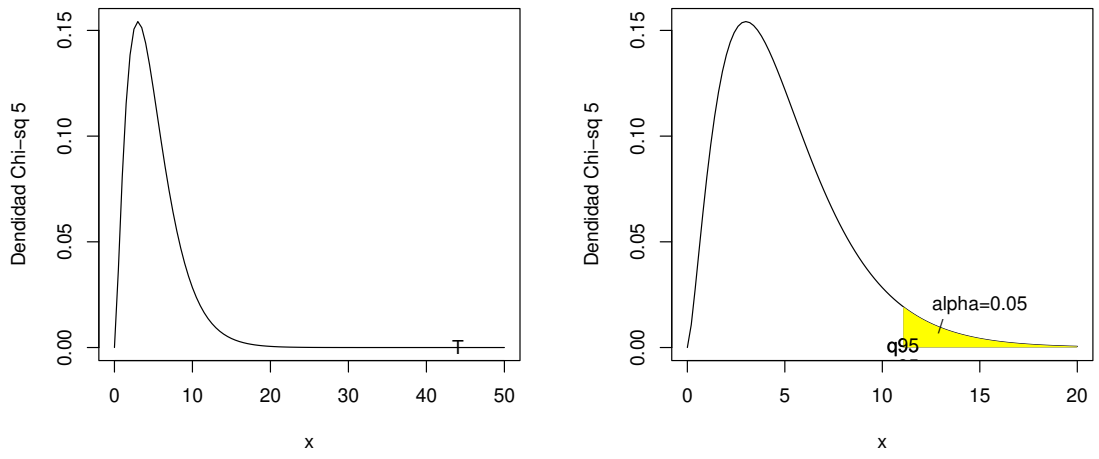


Figura 4.13: Gráfico de la función de densidad Chi-cuadrado con 5 grados de libertad y región crítica para el test de esfericidad.

Note que es muy posible que no haya esfericidad para ningún m ($m = 1, 2, 3$) (los valores propios teóricos son todos diferentes), pero esto no implica que tengamos que tomar todas las componentes principales. Si para algún m se acepta la esfericidad, no sería conveniente aumentar las componentes (ya que estás podrían obtenerse por azar) y sí podríamos intentar disminuir m.

4.9. Problemas

1. Calcular las componentes principales para una variable bidimensional con matriz de covarianzas
$$V = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}.$$
Calcular la información que tiene cada componente.

2. Calcular las componentes principales para una variable bidimensional con matriz de correlaciones
$$\Pi = \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix}.$$
¿Qué condiciones debe verificar r? Calcular la información que tiene cada componente.

3. Calcular las componentes principales para una variable bidimensional con matriz de covarianzas
$$\begin{pmatrix} 10 & -3 \\ -3 & 2 \end{pmatrix}.$$

4. Calcular la primera componente principal para una variable tridimensional con media cero y matriz de correlaciones
$$\Pi = \begin{pmatrix} 1 & 0.8 & 0.8 \\ 0.8 & 1 & 0.8 \\ 0.8 & 0.8 & 1 \end{pmatrix}.$$

5. Calcular las componentes principales para una variable tridimensional con media cero y matriz de covarianzas
$$\Sigma = \begin{pmatrix} \beta^2 + \delta & \beta & \beta \\ \beta & 1 + \delta & 1 \\ \beta & 1 & 1 + \delta \end{pmatrix}.$$
(Indicación: $\Sigma - \delta I = (\beta, 1, 1)'(\beta, 1, 1)$).

6. Demostrar que si las varianzas iniciales son iguales entonces las componentes principales que se obtienen con la matriz de covarianzas son iguales a las que se obtienen con la matriz de correlaciones.

7. Calcular las componentes principales de k variables con media cero, varianza uno y correlaciones iguales a r. ¿Qué condiciones debe verificar r? Calcular la información que tiene cada componente.

8. Demostrar que las componentes principales no son invariantes por cambio de escala.

9. (Teorema Perron-Frobenius). Si A es una matriz simétrica con todos sus elementos positivos, entonces todos los coeficientes del vector propio del mayor valor propio de A se pueden tomar todos positivos.

10. Aplicar un PCA a los datos del fichero: `USArrests` que contiene datos sobre los arrestos por cada 100000 residentes por asesinato, asalto o violación en cada uno de los 50 estados de USA en 1973. También se incluye el porcentaje de población que vive en las áreas urbanas. Fuente: `help(USArrests)`.

11. Aplicar un PCA a los datos del fichero de R: `USJudgeRatings`. Fuente: `help(USJudgeRatings)`.

12. Aplicar un PCA a la matriz de covarianza incluida en el fichero: `ability.cov` sobre diversos tests de inteligencia. Fuente: `help(ability.cov)`.

13. Aplicar un PCA a los datos de las columnas 5-10 del objeto `d` del fichero `bears.rda` ([3]). Esas columnas contienen diversas medidas de 143 osos (Head.L= longitud de la cabeza (pulgadas), Head.W=anchura de la cabeza (pulgadas), Neck.G=perímetro cuello (pulgadas), Length=altura (pulgadas), Chest.G=perímetro pecho (pulgadas), Weight=peso (libras)). Fuente: Minitab15. (Indicación: Para aplicar un PCA a las columnas 5-10 del objeto `d` debemos teclear: `PCA<-princomp(d[,5:10])`).

14. Aplicar un PCA a los datos del fichero `heptathlon` del paquete `MVA`[4] correspondientes a los resultados en la prueba femenina de heptatlon en las olimpiadas de Seul 1988.

15. Aplicar un PCA a los datos del fichero `pottery` del paquete `MVA`[4] que contiene resultados de análisis químicos de cerámica británica de la época romana de diversas regiones y hornos (kiln). La región 1 corresponde al horno 1, la región 2 a los hornos 2 y 3, y la región 3 a los hornos 4 y 5. ¿Podemos usar estas medidas para determinar el origen de la cerámica?

16. Aplicar un PCA a los datos del objeto `d` del fichero `nota.rda`[3] que contiene las notas (sobre 100) de alumnos de matemáticas en una universidad americana. Fuente: Rencher (1995, Methods of Multivariate Analysis, Wiley).

17. Aplicar un PCA a los datos del objeto `d` del fichero `madres.rda`[3] que contiene las medidas de madres y sus bebés recién nacidos. Las variables son: PESOM (peso madre), TALLAM (altura de la madre), SEM (semanas de gestación), PASM (presión sanguínea sistólica de la

[3]Para este tipo de archivos teclear `load('f:/name.rda')` indicando la ruta completa en donde se encuentra el archivo y reemplazando name por el nombre del archivo.

[4]Para leer este conjunto de datos hay que instalar el paquete `MVA` pinchando en el menú: `Paquete >Instalar Paquete` seleccionando `MVA` y tecleando en R: `library('MVA')` (o indicando en el menú que se carge este paquete).

madre), PADM (presión sanguínea diastólica de la madre), PESOR (peso del recién nacido), TALLAR (altura recién del nacido), PTR (perímetro torácico del recién nacido), PCR (perímetro craneal del recién nacido).

18. Aplicar un PCA a los datos del objeto `d` del fichero `decatlon.rda`[3] que contiene los resultados obtenidos por 24 atletas en las 10 pruebas de décatlon en los Juegos Olímpicos de Seul 1988. Las variables corresponden a las pruebas siguientes: X1 100 metros lisos (en segundos), X2 salto de longitud (metros), X3 lanzamiento de peso (metros), X4 salto de altura (metros), X5 400 metros lisos (segundos), X6 110 metros vallas (segundos), X7 lanzamiento de disco (metros), X8 salto con pértiga (metros), X9 lanzamiento de jabalina (metros), X10 1500 metros lisos (segundos) y X11 puntuación.

5

Análisis discriminante

En este capítulo mostramos cómo clasificar individuos entre varios grupos a partir de sus medidas en diversas variables aleatorias. Para ello necesitaremos disponer de una muestra de las variables en estudio en individuos de cada grupo (al menos dos individuos por cada grupo) y de las medidas de los elementos a clasificar en esas variables.

5.1. Introducción

El análisis discriminante trata de decidir si uno (o varios) "individuos" sobre el que se han medido una serie de características (variables) pertenece a una de las poblaciones existentes "a priori". Para ello construiremos *funciones discriminantes* que servirán para decidir en qué población incluimos a cada sujeto. Los ejemplo típicos son la diagnosis de enfermedades, las clasificación de individuos de diferentes especies, diagnosis de autoría en obras de arte, clasificación de perfiles de clientes (por ejemplo en la concesión de créditos), diseño de máquinas de clasificación automática en ingeniería, etc., aunque esta técnica se puede aplicar a muy diferentes situaciones (Economía, Psicología, Meteorología, Genética, etc.). Las variables estudiadas sobre los individuos deben ser numéricas (en muchos casos normales multivariantes) y, lógicamente, cuando no se conozcan las características de las poblaciones en las que se pueden clasificar los individuos, necesitaremos una muestra de individuos de cada una de ellas. El ACP (ver capítulo anterior) y otras técnicas de estadística descriptiva pueden servir de ayuda a la hora de visualizar las diferencias entre los individuos de distintas poblaciones, así como de los que están por clasificar, aunque veremos que pueden existir otras direcciones de proyección que permitan separar mejor a los grupos. Usaremos la técnica denominada *validación cruzada* para dar una estimación de las probabilidades de cada uno de los errores posibles (clasificar a un individuo de la población 1 en la población 2, etc...). La principal diferencia del Análisis Discriminante con el Análisis Cluster (de grupos) es que, en el primer caso, las poblaciones están establecidas de antemano, mientras que en el segundo la clasificación se realiza a posteriori (pudiéndose elegir el número de grupos deseados o el índice de "afinidad" deseada para los individuos de un determinado grupo). Cuando algunas de las variables de clasificación sean de tipo discreto es preferible utilizar otras técnicas de clasificación como la

regresión logística (ver Peña 2002).

El primero que claramente estudió un problema de Análisis Discriminante (AD o DA) fue Fisher (Ronald Aylmer, Inglaterra 1890-1962) quién fue consultado por Barnard en 1935 para clasificar restos de esqueletos. En 1936 Fisher introdujo la función discriminante para clasificar a un individuo en una de dos poblaciones normales con una matriz de covarianzas común. Básicamente veremos que esta clasificación se basa en la distancia de Mahalanobis del individuo a cada una de las poblaciones (sus medias). La utilización de esta distancia es equivalente bajo normalidad a la utilización del criterio de máxima verosimilitud que clasificará a un individuo en donde sus medidas sean "más probables" (verosímiles), es decir, donde la función de densidad sea más grande. Este segundo criterio permitirá la extensión de dicha clasificación a más de dos poblaciones con diferentes matrices de covarianzas incluso sin la necesidad de la normalidad de las mismas. Esta extensión fue llevada a cabo entre otros, por Welch (1939), Anderson (1951) y Okamoto (1963).

Recordemos que si X es una v.a. de dimensión k, media μ y matriz de covarianzas $V = (\sigma_{i,j})$ definida positiva, se define la **distancia de Mahalanobis** (Prasanta Chandra Mahalanobis, India 1893-1972) de $x, y \in \mathbb{R}^k$ como

$$\triangle(x, y) = \sqrt{(x-y)'V^{-1}(x-y).}$$

Obviamente, si V es la matriz identidad, obtenemos la distancia Euclídea. En la Figura 5.1 pueden verse "circunferencias" para la distancia de Mahalanobis con centro en la media para una Normal bivariante

$$N_2\left(\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, V = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}\right).$$

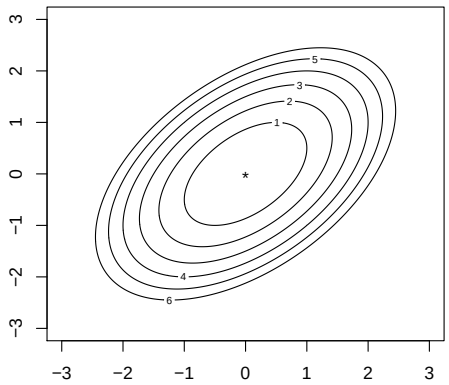


Figura 5.1: Circunferencias para la distancia de Mahalanobis en una población normal bidimensional con medias 0, varianzas 1 y correlación 1/2.

Como la función de densidad de la normal es

$$f(x) = \frac{1}{\sqrt{|V|(2\pi)^k}} \exp\left(-\frac{1}{2}(x-\mu)'V^{-1}(x-\mu)\right)$$

las circunferencias para la distancia de Mahalanobis con centro en μ coincidirán con las curvas de nivel de la función de densidad ($f(x) = cte$).

5.2. Clasificación teórica

5.2.1. Dos poblaciones normales con la misma matriz de covarianza

Supongamos que $X = (X_1, \ldots, X_k)'$ e $Y = (Y_1, \ldots, Y_k)'$ son dos v.a. normales k dimensionales con vectores de medias μ_X y μ_Y y matriz de covarianzas común V definida positiva. Supongamos que $Z = (Z_1, \ldots, Z_k)$ representa las medidas obtenidas para el individuo que se quiere clasificar y que Z proviene de X o de Y. Es decir, Z será una v.a. normal k dimensional con media igual a μ_X o μ_Y y matriz de covarianzas V. En la práctica z será un punto de $\mathbb{R}^k$ que debemos clasificar en X o en Y.

La idea de Fisher es usar una función discriminante D unidimensional lineal basada en Z. De esta forma,

$$D = a'Z = a_1Z_1 + \cdots + a_kZ_k$$

donde $a \in \mathbb{R}^k$ y si $Z \equiv N_k(\mu, V)$, entonces

$$D = a'Z \equiv N_1\left(a'\mu, \sqrt{a'Va}\right)$$

ya que $E(a'Z) = a'E(Z)$ y

$$Var(a'Z) = Cov(a'Z) = a'Cov(Z)a = a'Va,$$

donde $\mu = E(Z) = \mu_X$ ó μ_Y.

Esta función debe elegirse de forma que discrimine (aleje) a los individuos de X de los de Y, es decir, debemos resolver el problema siguiente:

$$\max_a \frac{(a'\mu_X - a'\mu_Y)^2}{a'Va}. \tag{5.1}$$

Nótese que el objetivo es alejar las "proyecciones" de las medias $a'\mu_X$ y $a'\mu_Y$ y disminuir la varianza común $\sigma^2 = a'Va$ (ver Figura 5.2). La solución se obtiene en el teorema siguiente.

Teorema 5.1. *Si V es definida positiva, la solución general de (5.1) es*

$$a = \lambda V^{-1}(\mu_X - \mu_y)$$

para $\lambda \neq 0$, y el máximo vale $\triangle^2(\mu_X, \mu_Y)$.

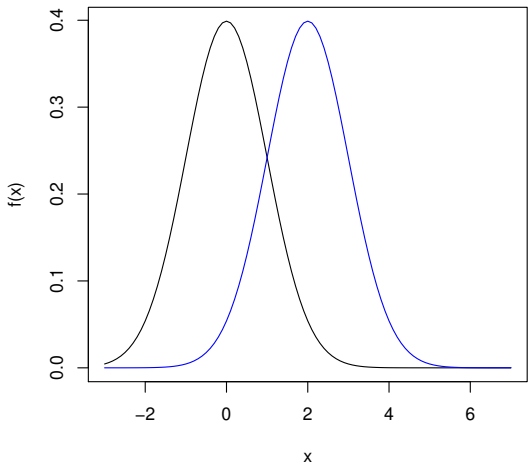


Figura 5.2: Funciones de densidad de las proyecciones en cada grupo.

Demostración. La demostración se basa en la desigualdad de Cauchy-Schwarz:

$$(x'y)^2 \leq (x'x)(y'y),$$

donde se da la igualdad si, y solo si, $x = \lambda y$. Como V es definida positiva, existe su inversa V^{-1} y $a'Va > 0$ para todo vector $a \neq 0$. Entonces, tenemos

$$\begin{aligned}
\frac{(a'\mu_X - a'\mu_Y)^2}{a'Va} &= \frac{(a'V^{1/2}V^{-1/2}(\mu_X - \mu_Y))^2}{a'Va} \\
&\leq \frac{a'Va(\mu_X - \mu_Y)'V^{-1}(\mu_X - \mu_Y)}{a'Va} \\
&= (\mu_X - \mu_Y)'V^{-1}(\mu_X - \mu_Y) \\
&= \triangle^2(\mu_X, \mu_Y),
\end{aligned}$$

donde $x' = a'V^{1/2}$ e $y = V^{-1/2}(\mu_X - \mu_Y)$. Además, se verifica la igualdad si, y solo si $x = \lambda y$, es decir, si

$$V^{1/2}a = \lambda V^{-1/2}(\mu_X - \mu_Y),$$

lo que implica

$$a = \lambda V^{-1}(\mu_X - \mu_Y).$$

□

Definición 5.1. *Llamaremos* ***función discriminante de Fisher*** *a la v.a.*

$$D = L(Z) = a'Z = (\mu_X - \mu_Y)'V^{-1}Z.$$

Nótese que en la proposición anterior no es necesario que las variables sean normales. Si las variables X e Y son normales, entonces la nueva variable D será normal

$$D \sim N_1((\mu_X - \mu_Y)'V^{-1}\mu, \triangle(\mu_X, \mu_Y)),$$

donde $\mu = E(Z)$ es igual a μ_X o μ_Y (ver Figura 5.2). Nótese que hemos tomado $\lambda = 1$, pero que esto no influye en la clasificación ya que podemos tomar cualquier otro λ no nulo. Por ejemplo, si tomamos $\lambda = 1/ \mid a \mid$ obtenemos una proyección en la dirección de a.

Con la función discriminante de Fisher, la regla de discriminación será:

- Si $L(Z) > K$, entonces Z es clasificado en X;

- Si $L(Z) < K$, entonces Z es clasificado en Y;

donde $K = L((\mu_X + \mu_Y)/2)$. En realidad clasificamos a un individuo con características z según $a'z$ esté más cerca de $a'\mu_X$ o de $a'\mu_Y$, ya que, como

$$(\mu_X - \mu_Y)'V^{-1}(\mu_X - \mu_Y) \geq 0,$$

entonces

$$a'\mu_X = (\mu_X - \mu_Y)'V^{-1}\mu_X \geq (\mu_X - \mu_Y)'V^{-1}\mu_Y = a'\mu_Y,$$

es decir, con esta función discriminante, la proyección de la media de X será siempre mayor que la proyección de la media de Y. Ocurrirá lo mismo si tomamos $\lambda > 0$ y lo contrario si tomamos $\lambda < 0$.

De esta forma, se crean dos regiones en el conjunto de posibles valores de Z, la región de individuos que serán clasificados en X y la de los que lo serán en Y:

$$R_X = \{z \in \mathbb{R}^k : L(z) > K\},$$
$$R_Y = \{z \in \mathbb{R}^k : L(z) < K\}.$$

Lógicamente, debemos dar una medida de lo "buena" que es la función discriminante obtenida. Está claro que será mejor cuanto más alejadas estén las medias $a'\mu_X$ y $a'\mu_Y$, y cuanto más pequeña sea la varianza $a'Va$. Así, el cociente

$$\frac{(a'\mu_X - a'\mu_Y)^2}{a'Va} = (\mu_X - \mu_Y)'V^{-1}(\mu_X - \mu_Y) = \triangle^2(\mu_X, \mu_Y)$$

(que no depende de λ) puede servir para comparar una función de discriminación con otra. Nótese que la discriminación será buena si las medias poblacionales (las poblaciones) están alejadas según la distancia de Mahalanobis asociada a V.

Otra forma de medir la bondad de un criterio de clasificación es la que calcula las probabilidades de malas (buenas) clasificaciones. Si llamamos error tipo 1, e_1, al que clasifica a un individuo de la

población X en la población Y, entonces

$$\begin{aligned}\Pr(e_1) &= \Pr(Z \in R_Y \mid Z \equiv X)\\ &= \Pr(L(X) < K)\\ &= \Pr\left(a'X < a'\frac{\mu_X+\mu_Y}{2}\right)\\ &= \Pr\left(\frac{a'X - a'\mu_X}{\sqrt{a'Va}} < \frac{a'(\mu_Y-\mu_X)}{2\sqrt{a'Va}}\right)\\ &= \Pr\left(U < \frac{(\mu_X-\mu_Y)'V^{-1}(\mu_Y-\mu_X)}{2\sqrt{(\mu_X-\mu_Y)'V^{-1}(\mu_X-\mu_Y)}}\right)\\ &= \Pr\left(U < -\frac{1}{2}\triangle(\mu_X,\mu_Y)\right),\end{aligned}$$

donde $U \equiv N_1(0,1)$. De forma análoga, puede comprobarse que

$$\Pr(e_2) = \Pr(Z \in R_X \mid Z \equiv Y) = \Pr\left(U > \frac{1}{2}\triangle(\mu_X,\mu_Y)\right) = \Pr(e_1).$$

Por lo tanto, las probabilidades de clasificaciones erróneas son iguales y solo dependen de la distancia de Mahalanobis entre las poblaciones. Lógicamente las probabilidades de clasificaciones correctas vienen dadas por:

$$\Pr(c_1) = \Pr(Z \in R_X \mid Z \equiv X) = 1 - \Pr(e_1),$$
$$\Pr(c_2) = \Pr(Z \in R_Y \mid Z \equiv Y) = 1 - \Pr(e_2),$$

y también son iguales.

Ejemplo 5.1. *Supongamos que tenemos que decidir si un individuo con medidas* $z = (z_1, z_2)' = (2, 0.9)'$ *se clasifica en una población normal bivariante de media* $\mu_X = (0,0)'$ *o en una de media* $\mu_Y = (1,2)'$ *siendo la matriz de covarianzas común*

$$V = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}.$$

Sus gráficas pueden verse en la Figura 5.3 y se obtienen con:

```
V<-matrix(NA,2,2)
V[1,]<-c(1,1/2)
V[2,]<-c(1/2,1)
muX<-c(0,0)
fX<-function(x1,x2)
dmvnorm(data.frame(x1,x2),muX,V)
muY<-c(1,2)
fY<-function(x1,x2)
```

```
dmvnorm(data.frame(x1,x2),muY,V)
f<-function(x1,x2) pmax(fX(x1,x2),fY(x1,x2))
x<-seq(-3,4,length=50)
y<-seq(-3,6,length=50)
z<-outer(x,y,f)
persp(x,y,z,xlab='x1',ylab='x2',zlab='f(x1,x2)',col='red',theta=60)
```

donde para usar `dmvnorm` *debemos cargar el paquete* `mvtnorm`.

La función discriminante será:

$$\begin{aligned} D &= L(Z) = a'Z = (\mu_X - \mu_Y)'V^{-1}Z \\ &= -(1,2)\begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}^{-1} Z \\ &= -(1,2)\begin{pmatrix} 4/3 & -2/3 \\ -2/3 & 4/3 \end{pmatrix} Z \\ &= -(0,2)\,Z \\ &= -2Z_2, \end{aligned}$$

es decir, $L(z_1, z_2) = -z_2$. *La distancia de Mahalanobis entre las dos poblaciones vale*

$$\begin{aligned} \triangle^2(\mu_X, \mu_Y) &= (\mu_X - \mu_Y)'V^{-1}(\mu_X - \mu_Y) \\ &= (0,2)(1,2)' = 4. \end{aligned}$$

Un individuo Z *será clasificado en la primera población si*

$$-2Z_2 > K = a'\frac{\mu_X + \mu_Y}{2} = -(0,2)(0.5,1)' = -2,$$

es decir, si $Z_2 < 1$. *En este caso,* $z = (1, 0.9)$ *será clasificado en* X, *con una probabilidad de error global*

$$\begin{aligned} \Pr(e_2) &= \Pr(Z \in R_X \mid Z \equiv Y) \\ &= \Pr(U > \frac{1}{2} \triangle (\mu_X, \mu_Y)) \\ &= \Pr(U > 1) \\ &= 1 - F_U(1) \\ &= 1 - 0.8413 = 0.1587, \end{aligned}$$

donde la función de distribución normal estándar $F_U(1)$ *se calcula en la Tabla 7.3 o en R haciendo:* `pnorm(1)`.

Note que otra función discriminate equivalente sería

$$L^*(z_1, z_2) = z_2,$$

(proyección sobre el eje y) con la que obtendríamos

$$L^*(\mu_X) = L^*(0,0) = 0,$$
$$L^*(\mu_Y) = L^*(1,2) = 2,$$
$$K^* = (L^*(\mu_X) + L^*(\mu_X))/2 = 1$$

y

$$L^*(z) = L^*(1,0.9) = 0.9,$$

con lo que z se clasificaría en X. Las proyecciones con esta función serán $N(0,1)$ ($L^*(X)$) y $N(2,1)$ ($L^*(Y)$). Las densidades de las proyecciones (sobre el eje y) pueden verse en la Figura 5.4, izquierda. La probabilidad de error 0.1587 *corresponde a las áreas menores determinadas por la recta vertical en el punto de corte de las densidades en $K = 1$. Para dibujarlas en R podemos hacer:*

```
curve(dnorm(x,0,1),-3,7,ylab='f(x)')
curve(dnorm(x,2,1),add=TRUE,col='blue')
```

Si hacemos cualquier otra proyección, los grupos aparecerán más mezclados. Por ejemplo, si proyectamos sobre el eje x, obtendremos las densidades de la Figura 5.4, derecha. ¿Cuál sería la probabilidad de error si usáramos estas proyecciones sobre el eje x?

Welch (1939) probó que, si las poblaciones son normales, el procedimiento de clasificación mediante la función discriminante de Fisher es máximo verosímil, es decir, que se clasifica a un individuo con características z en X si y solo si $f_X(z) > f_Y(z)$. Además, se comprueba que también es equivalente al criterio de clasificación basado en la distancia de Mahalanovis mínima.

Teorema 5.2. *Si las variables X e Y son normales multivariantes con matriz de covarianzas común V y función discriminante de Fisher L, entonces equivalen:*
1) $L(z) > K$.
2) $\triangle_V(z, \mu_X) < \triangle_V(z, \mu_Y)$.
3) $f_X(z) > f_Y(z)$.

Demostración. La primera condición $L(z) > K$ es

$$a'z > a'\frac{\mu_X + \mu_Y}{2},$$

con $a' = (\mu_X - \mu_Y)'V^{-1}$, es decir,

$$\begin{aligned}(\mu_X - \mu_Y)'V^{-1}z &> \frac{1}{2}(\mu_X - \mu_Y)'V^{-1}(\mu_X + \mu_Y)\\ &= \frac{1}{2}\mu_X'V^{-1}\mu_X - \frac{1}{2}\mu_Y'V^{-1}\mu_Y\end{aligned}$$

lo que equivale a

$$2\mu_X'V^{-1}z - 2\mu_Y'V^{-1}z > \mu_X'V^{-1}\mu_X - \mu_Y'V^{-1}\mu_Y.$$

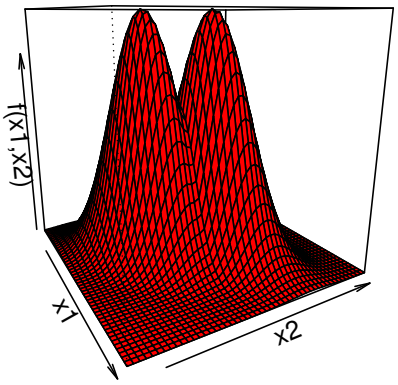


Figura 5.3: Funciones de densidad bivariantes para las poblaciones del Ejemplo 5.1 con medias $(0, 0)$ y $(1, 2)$.

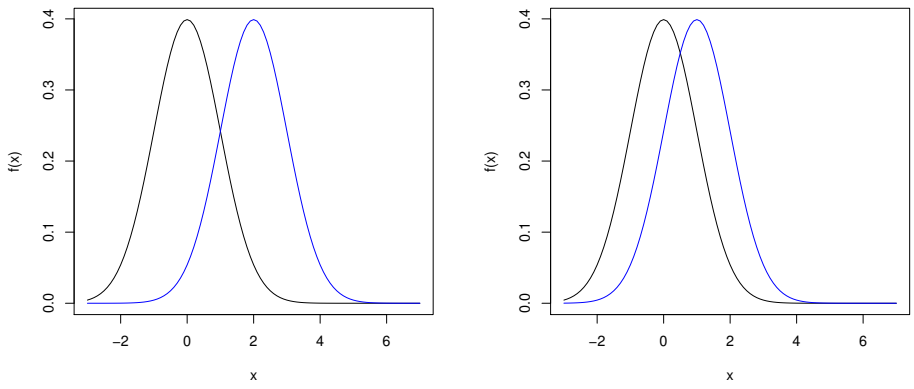


Figura 5.4: Funciones de densidad de las proyecciones sobre el eje y (izquierda) y el eje x (derecha) en cada grupo para las poblaciones del Ejemplo 5.1.

La segunda condición es equivalente a

$$(z-\mu_X)'V^{-1}(z-\mu_X) < (z-\mu_Y)'V^{-1}(z-\mu_Y) \tag{5.2}$$

y, desarrollando, se obtiene

$$z'V^{-1}z - 2\mu_X'V^{-1}z + \mu_X'V^{-1}\mu_X < z'V^{-1}z - 2\mu_Y'V^{-1}z + \mu_Y'V^{-1}\mu_Y,$$

es decir,

$$2\mu_X'V^{-1}z - 2\mu_Y'V^{-1}z > \mu_X'V^{-1}\mu_X - \mu_Y'V^{-1}\mu_Y$$

y, por tanto, las dos primeras condiciones son equivalentes.

Si las poblaciones son normales, la tercera condición es

$$c\exp\left(-\frac{1}{2}(z-\mu_X)'V^{-1}(z-\mu_X)\right) > c\exp\left(-\frac{1}{2}(z-\mu_Y)'V^{-1}(z-\mu_Y)\right),$$

es decir,

$$(z-\mu_X)'V^{-1}(z-\mu_X) < (z-\mu_Y)'V^{-1}(z-\mu_Y)$$

lo que es equivalente a la condición segunda. □

Observación 5.1. *Note que en la demostración anterior la hipótesis de normalidad no es necesaria para demostrar la equivalencia entre las dos primeras condiciones.*

Observación 5.2. *En ocasiones no es conveniente dar la misma importancia a los dos tipos de errores. Así, por ejemplo, podemos usar el criterio utilizado en los contrastes de hipótesis (Neyman-Pearson) que fija un máximo para uno de los errores* $\Pr(e_1) \leq \alpha$ *e intenta reducir la probabilidad del otro error. Usando este criterio sobre la función discriminate de Fisher, cambiaría la constante* K*, que ahora se calcularía a partir de la relación*

$$\Pr(e_1) = \Pr(Z \in R_Y | Z \equiv X) = \Pr(L(X) < K_\alpha) = \alpha,$$

donde $L(X) \equiv N_1((\mu_X - \mu_Y)'V^{-1}\mu_X, \sigma)$ *y*

$$\sigma^2 = \triangle^2(\mu_X, \mu_Y) = (\mu_X - \mu_Y)'V^{-1}(\mu_X - \mu_Y).$$

De esta forma, la probabilidad del otro error valdrá

$$\Pr(e_2) = \Pr(L(Y) > K_\alpha)$$

donde $L(Y) \equiv N_1((\mu_X - \mu_Y)'V^{-1}\mu_Y, \sigma)$.

Observación 5.3. Criterio de mínimo coste (probabilidad de error) *Otras veces se da un coste a cada uno de los posibles errores* $(c_1, c_2 > 0)$ *y, si se conocen las probabilidades "a priori" de pertenencia a cada una de las poblaciones* $q_1 = \Pr(Z \equiv X)$ *y* $q_2 = \Pr(Z \equiv Y)$*, entonces usando el teorema de la probabilidad total, se tiene*

$$\begin{aligned}\Pr(error) &= \Pr(Z \in R_Y \mid Z \equiv X)\Pr(Z \equiv X) + \Pr(Z \in R_X \mid Z \equiv Y)\Pr(Z \equiv Y) \\ &= \Pr(e_1)q_1 + \Pr(e_2)q_2\end{aligned}$$

y el coste esperado asociado para una constante k será

$$c(k) = c_1 \Pr(e_1) q_1 + c_2 \Pr(e_2) q_2,$$

donde

$$\Pr(e_1) = \Pr(L(X) < k) = G\left(\frac{k - L(\mu_X)}{\sigma}\right)$$

y

$$\Pr(e_2) = \Pr(L(Y) > k) = 1 - G\left(\frac{k - L(\mu_Y)}{\sigma}\right).$$

Por lo tanto,

$$c(k) = c_1 q_1 G\left(\frac{k - L(\mu_X)}{\sigma}\right) + c_2 q_2 - c_2 q_2 G\left(\frac{k - L(\mu_Y)}{\sigma}\right)$$

y, derivando, tenemos

$$c'(k) = \frac{c_1 q_1}{\sigma} g\left(\frac{k - L(\mu_X)}{\sigma}\right) - \frac{c_2 q_2}{\sigma} g\left(\frac{k - L(\mu_Y)}{\sigma}\right)$$

donde $g(u) = c\exp(-u^2/2)$ es la función de densidad normal estándar. Igualando a cero, obtenemos

$$c_1 q_1 g\left(\frac{k - L(\mu_X)}{\sigma}\right) = c_2 q_2 g\left(\frac{k - L(\mu_Y)}{\sigma}\right).$$

Despejando, se obtiene

$$\frac{(k - L(\mu_Y))^2}{\sigma} - \frac{(k - L(\mu_X))^2}{\sigma} = 2\log\left(\frac{c_2 q_2}{c_1 q_1}\right),$$

$$(L(\mu_Y))^2 - (L(\mu_X))^2 - 2k(L(\mu_Y) - L(\mu_X)) = 2\sigma^2 \log\left(\frac{c_2 q_2}{c_1 q_1}\right)$$

y, finalmente,

$$k = \frac{L(\mu_Y) + L(\mu_X)}{2} + \frac{\sigma^2}{L(\mu_X) - L(\mu_Y)} \log\left(\frac{c_2 q_2}{c_1 q_1}\right),$$

donde

$$L(\mu_X) - L(\mu_Y) = a'(\mu_X - \mu_Y) = (\mu_X - \mu_Y)' V^{-1} (\mu_X - \mu_Y) = \sigma^2$$

por lo que, para minimizar el coste esperado, debe tomarse

$$k = a' \frac{\mu_X + \mu_Y}{2} + \log\left(\frac{c_2 q_2}{c_1 q_1}\right).$$

Cuando $c_1 = c_2$ lo que se hace es minimizar la probabilidad total de error (mala clasificación). Si $c_1 q_1 = c_2 q_2$, la constante k coincide con la del criterio de Fisher $K = (L(\mu_X) + L(\mu_Y))/2$. Así demostramos que este criterio es el que minimiza la suma de las probabilidades de error.

Observación 5.4. Criterio de máxima probabilidad a posteriori. *Cuando se conozcan las probabilidades "a priori" q_1 y q_2, también se pueden calcular las probabilidades "a posteriori" (es decir, cuando conocemos sus los valores de Z) para un individuo con medidas z mediante el Teorema de Bayes como:*

$$\Pr(Z \equiv X \mid Z = z) = \frac{\Pr(Z = z \mid Z \equiv X)\Pr(Z \equiv X)}{\Pr(Z = z)},$$

con

$$\Pr(Z = z) = \Pr(Z = z \mid Z \equiv X)\Pr(Z \equiv X) + \Pr(Z = z \mid Z \equiv Y)\Pr(Z \equiv Y).$$

Si las variables son discretas, podremos calcular esas probabilidades. Si son continuas, reemplazaremos las probabilidades puntuales por las respectivas funciones de densidad obteniendo:

$$\Pr(Z \equiv X \mid Z = z) = \frac{q_1 f_X(z)}{q_1 f_X(z) + q_2 f_Y(z)}$$

y

$$\Pr(Z \equiv Y \mid Z = z) = \frac{q_2 f_Y(z)}{f_X(z) q_1 + q_2 f_Y(z)},$$

clasificándose a un individuo con características z en la población en la que tenga mayor "probabilidad a posteriori" (pero note que esos valores no son probabilidades sino verosimilitudes ponderadas para que sumen 1). En cualquier caso, esos valores nos indicarán la fiabilidad de la clasificación de un individuo z con esas medidas.

En la práctica, las probabilidades "a priori" q_1 y q_2 suelen ser desconocidas por lo que se tendrán que estimar (si es posible) o suponer iguales (si es razonable). Obviamente, si $q_1 = q_2$, entonces el criterio coincide con el de máxima verosimilitud. En general, se puede probar el resultado siguiente.

Proposición 5.1. *Los criterios de mínima probabilidad de error y de máxima probabilidad a posteriori son equivalentes.*

Demostración. Un individuo con medidas z se clasifica en X usando el criterio de máxima probabilidad a posteriori si y solo si

$$q_1 f_X(z) > q_2 f_Y(z).$$

Esta condición es equivalente a

$$\exp(-\Delta^2(z, \mu_X)/2) > \frac{q_2}{q_1} \exp(-\Delta^2(z, \mu_Y)/2),$$

es decir, a

$$\Delta^2(z, \mu_Y) - \Delta^2(z, \mu_X) > 2 \log \frac{q_2}{q_1}.$$

Operando se obtiene que esta condición es equivalente a

$$2(\mu_X - \mu_Y)' V^{-1} z > 2 \log \frac{q_2}{q_1} + \mu_X' V^{-1} \mu_X - \mu_Y' V^{-1} \mu_Y \tag{5.3}$$

donde

$$\mu_X' V^{-1} \mu_X - \mu_Y' V^{-1} \mu_Y = (\mu_X - \mu_Y)' V^{-1} \mu_X + (\mu_X - \mu_Y)' V^{-1} \mu_Y.$$

Finalmente, si $a' = (\mu_X - \mu_Y)'V^{-1}$, la condición (5.3) puede escribirse como

$$a'z > \log \frac{q_2}{q_1} + \frac{a'\mu_X + a'\mu_Y}{2}$$

donde la expresión de la derecha coincide con la constante k obtenida en el criterio de mínimo coste cuando $c_1 = c_2 = 1$ (es decir, en el criterio de mínima probabilidad de error total). □

Observación 5.5. *Cuando el análisis discriminante se aplica a datos biomédicos (tests) siendo X la población de los individuos que tienen una determinada enfermedad e Y los que no la tienen, suelen utilizarse los términos "sensibilidad" (o Recall) y "especificidad" siendo*

$$sensibilidad = S = 100(1 - \Pr(e_1)) = 100 \Pr(Z \in R_X \mid Z \equiv X) \approx \frac{TP}{TP + FN} 100$$

$$especificidad = E = 100(1 - \Pr(e_2)) = 100 \Pr(Z \in R_Y \mid Z \equiv Y) \approx \frac{TN}{TN + FP} 100,$$

donde TP=verdaderos positivos, FN=falsos negativos, TN=verdaderos de negativos y FP=falsos positivos. Es decir, la sensibilidad es el porcentaje de individuos con la enfermedad detectados y la especificidad es el porcentaje de individuos sin la enfermedad descartados. Los resultados suelen resumirse con la denominada matriz de confusión, ver Tabla 5.1.

Resumen	Prediction Positive (PP)	Prediction Negative (PN)	Total
Real value Positive	True Positive (TP)	False Negative (FN)	P
Real value Negative	False Positive (FP)	True Negative (TN)	N
Total	PP	PN	Total Population

Tabla 5.1: Matriz de confusión.

También se habla del "valor predictivo" del test como el porcentaje de individuos bien clasificados entre los que han sido clasificados dentro de una de las poblaciones:

$$valor\ predictivo\ positivos = 100 \Pr(Z \equiv X \mid Z \in R_X) \approx 100 \frac{TP}{TP + FP}$$

$$valor\ predictivo\ negativos = 100 \Pr(Z \equiv Y \mid Z \in R_Y) \approx 100 \frac{TN}{TN + FN},$$

es decir, nos dice cuántos de los positivos (negativos) son realmente positivos (negativos). El valor predictivo de los positivos también se conoce como "precisión" (P). Se puede usar, junto con la sensibilidad, para obtener la puntuación F_1 (F score) como

$$F_1 = 2\left(\frac{1}{P} + \frac{1}{S}\right)^{-1} = \frac{2PS}{P + S},$$

es decir, es la media armóninca de la precisión y la sensibilidad. Este índice está entre 0 y 1. Vale cero si $P = 0$ o $S = 0$ y vale 1 si $P = S = 1$. En general, el índice F-beta se define como

$$F_\beta = (1 + \beta^2) \frac{PS}{\beta^2 P + S}$$

para $\beta > 0$. Si $\beta = 0$ solo se tiene en cuenta la precisión P y si $\beta = \infty$ solo la sensibilidad S.Con $\beta = 1$ hacemos un tratamiento simétrico de los dos errores. La "eficiencia" (accuracy) de una prueba indica el porcentaje de individuos bien clasificados

$$eficiencia = q_1(1 - \Pr(e_1)) + q_2(1 - \Pr(e_2)) \approx \frac{TP + TN}{total}.$$

Los resultados pueden ser muy diferentes cuando q_1 y q_2 sean desconocidas. Por último, la "prevalencia" de una enfermedad es el porcentaje de enfermos en la población total. En la práctica todas estas cantidades se pueden aproximar a partir de una muestra test diferente de la usada para ajustar los parámetros.

5.2.2. Varias poblaciones con la misma matriz de covarianza

Cuando tengamos más de dos poblaciones con matriz de covarianzas común V, podemos usar el criterio de mínima distancia de Mahalanobis a las medias de los grupos. Para ello, si queremos clasificar a Z entre diversos grupos con variables $X^{(i)} = (Z \mid G = i)$ de medias $\mu^{(i)} = E(X^{(i)})$, para $i = 1, \ldots, m$, calcularemos

$$\begin{aligned}\Delta^2(z, \mu^{(i)}) &= (z - \mu^{(i)})' V^{-1} (z - \mu^{(i)}) \\ &= z' V^{-1} z - 2(\mu^{(i)})' V^{-1} z + (\mu^{(i)})' V^{-1} \mu^{(i)}.\end{aligned}$$

Como la parte cuadrática $z' V^{-1} z$ es común, podemos quedarnos solo con la parte lineal (en realidad, su opuesta) dada por

$$L_i(z) = (\mu^{(i)})' V^{-1} z - (\mu^{(i)})' V^{-1} \mu^{(i)}/2$$

conocida como **función discriminante lineal (FDL)**, clasificándose un individuo con características z en el grupo en el que tenga un valor máximo dicha función discriminante. Como consecuencia se obtiene el teorema siguiente.

Teorema 5.3. *Si $X^{(i)}$ tienen medias $\mu^{(i)} = E(X^{(i)})$ y matriz de covarianzas común V para $i = 1, \ldots, m$, entonces equivalen:*
1) $L_i(z) \geq L_j(z)$ para todo j.
2) $\Delta^2(z, \mu^{(i)}) \leq \Delta^2(z, \mu^{(j)})$ para todo j.

Corolario 5.1. *Si solo hay dos grupos, este criterio de clasificación es equivalente a usar la función discriminante de Fisher.*

La demostración del corolario es inmediata ya que demostramos en la sección anterior que el criterio de mínima distancia de Mahalanobis era equivalente a usar la función discriminante de Fisher.

En este caso también podemos aplicar el criterio de máxima verosimilitud clasificando a Z en el grupo para el que $f_i(z)$ sea máxima, donde f_i representa la densidad del grupo i. Bajo normalidad, tenemos el resultado siguiente.

Teorema 5.4. *Si* $X^{(j)} \sim N(\mu^{(j)}, V)$ *para* $j = 1, \ldots, m$, *entonces equivalen:*
1) $L_i(z) \geq L_j(z)$ *para todo* j.
2) $\Delta^2(z, \mu^{(i)}) \leq \Delta^2(z, \mu^{(j)})$ *para todo* j.
3) $f_i(z) \geq f_j(z)$ *para todo* j.

La demostración es inmediata ya que la densidad normal multivariante del grupo i vale

$$f_i(z) = \frac{1}{\sqrt{\mid V \mid (2\pi)^k}} \exp\{-\frac{1}{2}(z - \mu^{(i)})' V^{-1}(z - \mu^{(i)})\}$$

y será máxima cuando la distancia de Mahalanobis al cuadrado

$$\Delta^2(z, \mu^{(i)}) = (x - \mu^{(i)})' V^{-1}(x - \mu^{(i)})$$

sea mínima.

Nótese que esto no será, en general, cierto si las poblaciones no son normales o si tienen distintas matrices de covarianzas. Como consecuencia inmediata, se obtiene el resultado siguiente.

Proposición 5.2. *Si todas las poblaciones son normales con matriz de covarianzas común* V, *entonces los criterios de clasificación de máxima verosimilitud y de mínima distancia de Mahalanobis son equivalentes a aplicar el criterio de discriminación de Fisher paso a paso tomando las poblaciones de dos en dos.*

De esta forma, podríamos estudiar primero si z se clasifica en la población 1 o en la 2. En el segundo paso discriminaríamos entre la 3 y la ganadora del primer paso y así, sucesivamente. Sin embargo, este método no se puede aplicar en la práctica ya que al discriminar entre las poblaciones 1 y 2 solo se utilizarían los individuos de estas poblaciones para estimar V (ver siguiente sección).

El método de proyección de Fisher puede generalizarse para más de dos grupos obteniéndose las denominadas **proyecciones canónicas** que permiten separar (discriminar) mejor a los grupos i y j. La idea se basa en la representación de la función discriminante de Fisher siguiente:

$$\begin{aligned} L(Z) &= (\mu_X - \mu_Y)' V^{-1} Z && (5.4)\\ &= (\mu_X - \mu_Y)' V^{-1/2} V^{-1/2} Z && (5.5)\\ &= (V^{-1/2}\mu_X - V^{-1/2}\mu_Y)' V^{-1/2} Z, && (5.6) \end{aligned}$$

donde $Z^* = V^{-1/2}Z$ es una variable con $Cov(V^{-1/2}Z) = V^{-1/2}VV^{-1/2} = I$, $V^{-1/2} = T'D^{-1/2}T$ y donde T es la matriz ortonormal que diagonaliza a V ($T'VT = D$, $T'T = TT' = I$). Es decir, el criterio basado en la función discriminante de Fisher equivale a, primero estandarizar las variables haciendo que tengan covarianza I y medias $\mu_X^* = V^{-1/2}\mu_X'$ y $\mu_Y^* = V^{-1/2}\mu_Y'$, y luego proyectarlas en la dirección de la recta que une ambas medias $\mu_X^* - \mu_Y^*$. Esto corresponde con la solución óptima para la distancia de Mahalanobis cuando $V = I$, es decir, cuando usamos la distancia Euclídea. Así, una vez proyectados el punto z y las medias sobre el espacio canónico (Euclídeo), z se clasificará en el grupo cuya proyección de la media esté más cercana a su proyección en la distancia euclídea

(unimos las medias y trazamos la mediatriz para separar los grupos). Este criterio también es equivalente al criterio de mínima distancia de Mahalanobis ya que las distancias euclídeas de la proyección de z a los grupos verificarán:

$$d_i^2(z) = d^2(V^{-1/2}z, V^{-1/2}\mu^{(i)}) = (z-\mu^{(i)})'V^{-1/2}V^{-1/2}(z-\mu^{(i)}) = \Delta^2(z,\mu^{(i)})$$

para todo i.

Si solo hay dos grupos, podemos representar los puntos proyectados usando la función discriminante de Fisher, es decir, las proyecciones de los puntos transformados $z^* = V^{-1/2}z$ sobre la recta que une las dos medias transformadas $\mu_X^* = V^{-1/2}\mu_X$ y $\mu_Y^* = V^{-1/2}\mu_Y$. Si hay tres grupos, podemos representar las proyecciones de los puntos transformados $z^* = V^{-1/2}z$ sobre el plano que forman las tres medias transformadas $\mu_X^* = V^{-1/2}\mu_X$, $\mu_Y^* = V^{-1/2}\mu_Y$ y $\mu_Z^* = V^{-1/2}\mu_Z$. Si estos puntos forman un triángulo, sus mediatrices y su circuncentro determinarán las regiones de clasificación (ya que, en este espacio, podemos usar la distancia Euclídea). Si los puntos están alineados, las regiones de clasificación vendrán dadas por las dos mediatrices obtenidas con el punto central y los dos extremos. En este último caso, en realidad bastaría con proyectar sobre la recta formada por estos puntos. Se procede de forma análoga si hay más grupos. Finalmente mencionar que la matriz $V^{-1/2}$ se puede reemplazar por cualquier matriz U' no singular (invertible) tal que $UU' = V^{-1}$ ya que entonces $V = (UU')^{-1} = (U')^{-1}U^{-1}$ y

$$Cov(U'Z) = U'VU = U'(U')^{-1}U^{-1}U = I.$$

Ejemplo 5.2. *Supongamos que tenemos que decidir si un individuo con medidas* $z = (x,y)' = (1, 0.9)'$ *se clasifica en una población normal bivariante de media* $(0,0)'$, *en una de media* $(1,2)'$ *o en una de media* $(-1/2, 1)$ *siendo la matriz de covarianzas común*

$$V = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}.$$

Entonces las funciones discriminantes lineales (LDF) para $z = (x,y)'$ *serán:*

$$L_1(x,y) = (\mu^{(1)})'V^{-1}z - (\mu^{(1)})'V^{-1}\mu^{(1)}/2 = 0,$$

$$\begin{aligned} L_2(x,y) &= (\mu^{(2)})'V^{-1}z - (\mu^{(2)})'V^{-1}\mu^{(2)}/2 \\ &= (1,2)\begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}^{-1}\begin{pmatrix} x \\ y \end{pmatrix} - \frac{1}{2}(1,2)\begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}^{-1}\begin{pmatrix} 1 \\ 2 \end{pmatrix} \\ &= (1,2)\begin{pmatrix} 4/3 & -2/3 \\ -2/3 & 4/3 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} - \frac{1}{2}(1,2)\begin{pmatrix} 4/3 & -2/3 \\ -2/3 & 4/3 \end{pmatrix}\begin{pmatrix} 1 \\ 2 \end{pmatrix} \\ &= 2y - 2 \end{aligned}$$

y

$$\begin{aligned} L_3(x,y) &= (\mu^{(3)})'V^{-1}z - \frac{1}{2}(\mu^{(3)})'V^{-1}\mu^{(3)} \\ &= (-1/2,1)\begin{pmatrix} 4/3 & -2/3 \\ -2/3 & 4/3 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \\ &\quad - \frac{1}{2}(-1/2,1)\begin{pmatrix} 4/3 & -2/3 \\ -2/3 & 4/3 \end{pmatrix}\begin{pmatrix} -1/2 \\ 1 \end{pmatrix} \\ &= -\frac{4}{3}x + \frac{5}{3}y - \frac{7}{6}. \end{aligned}$$

En particular, para $z = (1, 0.9)'$ *obtenemos*

$$\begin{aligned} L_1(1,0.9) &= 0 \\ L_2(1,0.9) &= 2(0.9) - 2 = -0.2 \\ L_3(1,0.9) &= -\frac{4}{3} + \frac{5}{3}0.9 - \frac{7}{6} = -1, \end{aligned}$$

por lo que z se clasificará en la primera población (donde L_i *es máxima).*

Para calcular las regiones de clasificación para los grupos 1 y 2 haremos $L_1 = L_2$*, es decir,*

$$0 = 2y - 2$$

obteniendo $y = 1$ *(como ya vimos anteriormente usando la función discriminante de Fisher). Para los grupos 1 y 3, obtenemos*

$$0 = -\frac{4}{3}x + \frac{5}{3}y - \frac{7}{6},$$

es decir, $y = \frac{4}{5}x + \frac{7}{10}$ *y para los grupos 2 y 3, obtenemos*

$$2y - 2 = -\frac{4}{3}x + \frac{5}{3}y - \frac{7}{6},$$

es decir, $y = -4x + \frac{5}{2}$*. Las regiones de clasificación se pueden ver en la Figura 5.5. Las tres rectas se cortan en el punto* $(3/8, 1)$*. Para representarlas en R haremos:*

```
curve(4*x/5+7/10,-3,5,ylab='y',axes=TRUE)
curve(1+x-x,add=TRUE)
curve(-4*x+5/2,add=TRUE)
text(0,0,labels='mu1')
text(1,2,labels='mu2')
text(-0.5,1,labels='mu3')
text(1,0.9,labels='z')
```

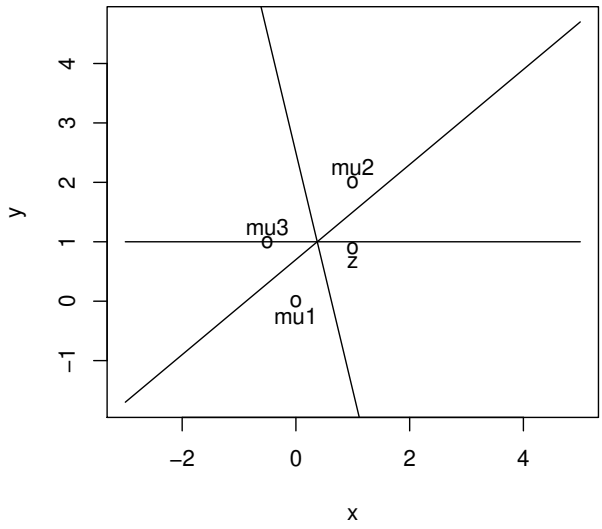


Figura 5.5: Regiones de clasificación para las poblaciones del Ejemplo 5.2.

Para calcular las proyecciones canónicas $z^* = V^{-1/2}z$ *al espacio Euclídeo, debemos calcular la matriz ortogonal* T *tal que* $T'VT = D$*, donde* D *es diagonal con lo que*

$$V^{-1/2} = TD^{-1/2}T'.$$

Las matrices T *y* D *se calcularon en el Ejemplo 4.4, obteniéndose*

$$T = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix}$$

y

$$D = \begin{pmatrix} 3/2 & 0 \\ 0 & 1/2 \end{pmatrix},$$

con lo que

$$\begin{aligned} V^{-1/2} &= TD^{-1/2}T' \\ &= \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \begin{pmatrix} \sqrt{2/3} & 0 \\ 0 & \sqrt{2} \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{6}\sqrt{6} + \frac{1}{2}\sqrt{2} & \frac{1}{6}\sqrt{6} - \frac{1}{2}\sqrt{2} \\ \frac{1}{6}\sqrt{6} - \frac{1}{2}\sqrt{2} & \frac{1}{6}\sqrt{6} + \frac{1}{2}\sqrt{2} \end{pmatrix}. \end{aligned}$$

De esta forma, se tiene:

$$
\begin{aligned}
V^{-1/2}z &= \begin{pmatrix} \frac{1}{6}\sqrt{6}+\frac{1}{2}\sqrt{2} & \frac{1}{6}\sqrt{6}-\frac{1}{2}\sqrt{2} \\ \frac{1}{6}\sqrt{6}-\frac{1}{2}\sqrt{2} & \frac{1}{6}\sqrt{6}+\frac{1}{2}\sqrt{2} \end{pmatrix} \begin{pmatrix} 1 \\ 0.9 \end{pmatrix} = \begin{pmatrix} 0.846382 \\ 0.704961 \end{pmatrix} \\
V^{-1/2}\mu_1 &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\
V^{-1/2}\mu_2 &= \begin{pmatrix} \frac{1}{6}\sqrt{6}+\frac{1}{2}\sqrt{2} & \frac{1}{6}\sqrt{6}-\frac{1}{2}\sqrt{2} \\ \frac{1}{6}\sqrt{6}-\frac{1}{2}\sqrt{2} & \frac{1}{6}\sqrt{6}+\frac{1}{2}\sqrt{2} \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0.517638 \\ 1.938516 \end{pmatrix} \\
V^{-1/2}\mu_3 &= \begin{pmatrix} \frac{1}{6}\sqrt{6}+\frac{1}{2}\sqrt{2} & \frac{1}{6}\sqrt{6}-\frac{1}{2}\sqrt{2} \\ \frac{1}{6}\sqrt{6}-\frac{1}{2}\sqrt{2} & \frac{1}{6}\sqrt{6}+\frac{1}{2}\sqrt{2} \end{pmatrix} \begin{pmatrix} -1/2 \\ 1 \end{pmatrix} = \begin{pmatrix} -0.856536 \\ 1.264784 \end{pmatrix}.
\end{aligned}
$$

Por último, calculamos las distancias euclídeas entre la proyección de z y las proyecciones de las medias obteniendo

$$
\begin{aligned}
d_1 &= \sqrt{0.846382^2 + 0.704961^2} \approx 1.101514 \\
d_2 &= \sqrt{(0.846382 - 0.517638)^2 + (0.704961 - 1.938516)^2} \approx 1.276609 \\
d_3 &= \sqrt{(0.846382 + 0.856536)^2 + (0.704961 - 1.264784)^2} \approx 1.792577,
\end{aligned}
$$

con lo que (como ya sabíamos) z se clasifica (por poco) en el grupo 1

5.2.3. Varias poblaciones con distintas matrices de covarianza

Los criterios de clasificación por máxima verosimilitud o por mínima distancia de Mahalanobis a las medias de los grupos pueden utilizarse aunque las poblaciones no tengan la misma matriz de covarianzas. De hecho, tampoco es necesario que las poblaciones sean normales, pudiéndose aplicar incluso a poblaciones de tipo discreto (siempre que se conozcan las densidades o las funciones puntuales de probabilidad). Cuando las poblaciones sean normales suele hablarse de **Análisis Discriminante Cuadrático** (ADC o QDA) ya que las funciones que determinan las regiones de clasificación son polinomios de grado 2. Sin embargo, en este caso, las funciones discriminantes para mínima distancia o máxima verosimilitud no coinciden. El criterio de máxima verosimilitud buscaría el máximo de

$$
f_i(z) = \frac{1}{\sqrt{\mid V_i \mid (2\pi)^k}} \exp\left(-\frac{1}{2}(z-\mu^{(i)})' V_i^{-1}(z-\mu^{(i)})\right)
$$

o, equivalentemente, el mínimo de

$$
Q_i(z) = c - 2\log f_i(z) = (z-\mu^{(i)})' V_i^{-1}(z-\mu^{(i)}) + \log \mid V_i \mid,
$$

conocida como **función discriminante cuadrática (QDF)** para $i = 1, \ldots, m$. Un individuo con medidas z se clasificará en el grupo donde la función discriminante cuadrática sea mínima (máxima verosimilitud).

Sin embargo, el criterio basado en la distancia de Mahalanobis, usará la función discriminante cuadrática

$$Q_i^*(z) = \Delta^2(z, \mu^{(i)}) = (z - \mu^{(i)})' V_i^{-1} (z - \mu^{(i)}).$$

Por lo tanto, los resultados pueden ser diferentes (cuando los determinantes de las matrices de covarianzas sean diferentes).

En general, las funciones discriminantes cuadráticas son muy "sensibles" cuando las poblaciones no son normales, por lo que no es muy recomendable su uso en este caso, siendo preferible usar funciones discriminantes lineales. Como veremos posteriormente, en la práctica podemos usar las técnicas de *validación cruzada* para elegir el mejor método posible con los datos disponibles.

Veamos un ejemplo.

Ejemplo 5.3. *Sean dos poblaciones normales bidimensionales con medias* $\mu_1 = (2,0)'$ *y* $\mu_2 = (0,0)'$ *y matrices de covarianzas*

$$V_1 = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}$$

y

$$V_2 = \begin{pmatrix} 2 & 0 \\ 0 & 1/2 \end{pmatrix}.$$

Se pide: Calcular las funciones discriminates cuadráticas, dar el criterio de clasificación, dibujar las regiones de clasificación en R^2 *y clasificar a* $z = (1,1)'$.

Como

$$V_1^{-1} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix},$$

la primera QDF es

$$\begin{aligned} Q_1(x,y) &= \begin{pmatrix} x-2 & y \end{pmatrix} V_1^{-1} \begin{pmatrix} x-2 \\ y \end{pmatrix} \\ &= (x-2+y)(x-2) + (x-2+2y)\, y \\ &= x^2 - 4x + 4 + 2yx - 4y + 2y^2. \end{aligned}$$

Análogamente, para el segundo grupo tenemos:

$$\begin{pmatrix} 2 & 0 \\ 0 & 1/2 \end{pmatrix}^{-1} = \begin{pmatrix} 1/2 & 0 \\ 0 & 2 \end{pmatrix}$$

con lo que la segunda QDF será

$$Q_2 = \begin{pmatrix} x & y \end{pmatrix} V_2^{-1} \begin{pmatrix} x \\ y \end{pmatrix} = 0.5x^2 + 2y^2,$$

es decir,

$$Q_2(x,y) = 0.5x^2 + 2y^2.$$

Entonces $z = (x, y)$ se clasifica en 1 si y solo si $Q_1 < Q_2$, es decir, si

$$(2x - 4)y < 4x - 4 - x^2/2.$$

Como:

$$Q_1(1, 1) = 1$$

y

$$Q_2(1, 1) = 5/2,$$

se clasifica en 1.

Para calcular las regiones de clasificación notamos que, en la ecuación anterior, tenemos tres casos:

1) Si $x > 2$, $Q_1 < Q_2$, si y solo si, $y < (4x - 4 - x^2/2)/(2x - 4)$.

2) Si $x < 2$, $Q_1 < Q_2$, si y solo si, $y < (4x - 4 - x^2/2)/(2x - 4)$.

3) Si $x = 2$, $0 < 8 - 4 - 4/2 = 2$, por lo que $Q_1 < Q_2$ y se clasifica en 1.

Las regiones de clasificación se pueden ver en la Figura 5.6. El código para realizarla es:

```
D<-function(x) (4*x-4-x*x/2)/(2*x-4)
x<-seq(-1,5,by=0.01)
f<-D(x)
min<- x-x-6
max<- x-x+6
plot(x,f,type='l',ylim=c(-5,5),xlim=c(-1,5),ylab='y',xlab='x')
i<-1:301
polygon(c(x[i],rev(x[i])),c(min[i],rev(f[i])),col="#00009920",border=NA)
points(0,0,pch=20)
text(0.35,0,'mu2',col='red', cex=0.8)
points(2,0,pch=20)
text(2.35,0,'mu1',col='red', cex=0.8)
i<-315:601
polygon(c(x[i],rev(x[i])),c(max[i],rev(f[i])),col="#00009920",border=NA)
points(1,1,pch=20)
text(1.2,1,'z',col='red', cex=0.8)
```

Observación 5.6 (**Detectando anomalías**). *En otras ocasiones lo que tenemos es un conjunto de unidades de la población correcta (por ejemplo unidades bien fabricadas) y otras (pocas) que no pertenecen a dicha población (unidades defectuosas). El objetivo es detectar esas unidades. La principal diferencia es que, en este caso, las unidades defectuosas no forman un grupo homogéneo que provengan de una determinada población. Es más, puede que las unidades defectuosas futuras*

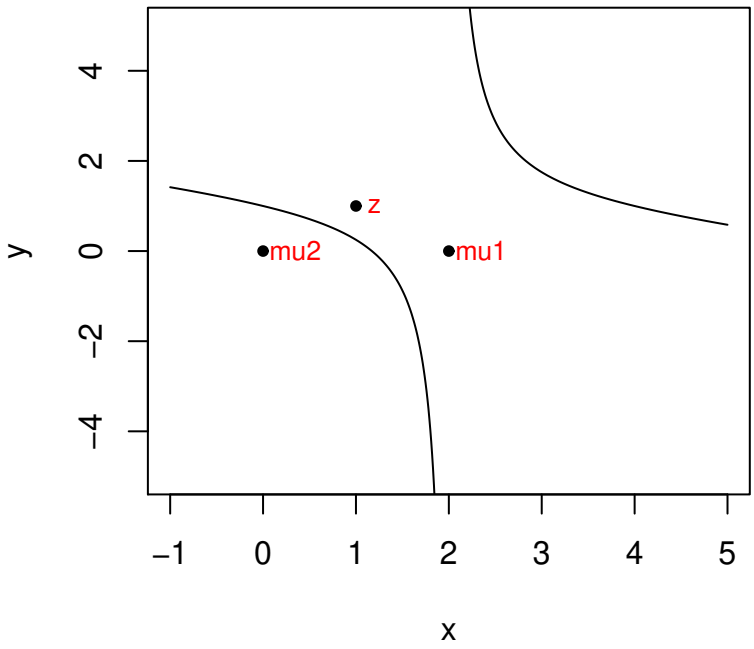


Figura 5.6: Regiones de clasificación Ejemplo 5.3.

sean completamente diferentes de las que tenemos en la muestra. En este caso, la regla de decisión bajo normalidad para una unidad con medidas $z = (z_1, \ldots, z_k)$*, será:*

Regla	y	*Grupo*
$f(z; \mu, V) < \epsilon$	$y = 1$	*Unidad defectuosa*
$f(z; \mu, V) \geq \epsilon$	$y = 0$	*Unidad no defectuosa*

donde f *es la función de densidad normal multivariante* $N_k(\mu, V)$ *y* $\epsilon > 0$ *es un parámetro a determinar a partir de la muestra optimizando la detección de errores y la detección de unidades bien fabricadas (usando, por ejemplo,* F_1*). Claramente, ésto es equivalente a usar la distancia de Mahalanobis (o la desigualdad de Tchebychev) para detectar esas unidades mal fabricadas (outliers). En la práctica, el vector de medias y la matriz de covarianzas se reemplazarán por sus estimaciones a partir de unidades bien fabricadas. La región de aceptación de las unidades bien fabricadas será un elipsoide de concentración con centro en* μ *y la de las defectuosas su parte exterior. En este caso, lo típico es que en la muestra (donde conocemos los grupos de cada unidad), haya pocos datos del grupo de las unidades defectuosas. Todos esos datos de unidades defectuosas se pueden usar junto con un número similar de unidades bien fabricadas (no usadas para estimar* μ *y* V*) para estimar el* ϵ *óptimo (muestra de validación cruzada, CV) y para estimar la fiabilidad de nuestro test (e.g.* F_1*). Como* ϵ *se toma para maximizar la eficacia en la muestra de CV, los datos de esa muestra no se pueden usar para determinar la eficacia del test (no se deben incluir en la muestra test). Si las variables son muy numerosas (*k *muy grande) el cálculo de la inversa de* V *puede ser lento. En este caso se puede asumir que las variables son independientes (usando el productor de las marginales normales) o usar un PCA para reducir el número de variables manteniendo una gran cantidad de la información inicial (e.g. un* $99\,\%$*). Si no estamos seguros de la normalidad de la población inicial, la función de densidad* f *se puede sustituir por su estimador tipo núcleo (Kernel estimator)*

$$\hat{f}(z) = \frac{1}{n} \sum_{i=1}^{n} f(z; x^{(i)}, h_n I_k),$$

donde $x^{(i)} = (x_1^{(i)}, \ldots, x_k^{(i)})$*,* $i = 1, \ldots, n$*, son los elementos de la muestra (de entrenamiento) bien fabricadas,* n *es el tamaño de esa muestra,* I_k *es la matriz identidad de tamaño* k*, y* $h_n > 0$ *es el ancho de banda (que depende de* n*). Este estimador es equivalente a sustituir cada punto de la muestra por una mixtura (combinación lineal convexa) uniforme de distribuciones normales con variables independientes, media en los puntos de la muestra y varianza común* h_n*. La regla de decisión es similar (en este caso no nos saldrán elipsoides) y donde los parámetros* ϵ *y* h *se pueden determinar usando la muestra CV. De nuevo usaremos una muestra nueva (test) para estimar la eficacia de nuestro procedimiento. Por ejemplo, si tenemos* 40 *unidades defectuosas, podemos usar* 20 *para determinar los parámetros (junto con otras* 20 *bien fabricadas) y las otras* 20 *para estimar la eficacia del test (junto con otras 20 bien fabricadas, no usadas anteriormente).*

5.3. Clasificación a partir de una muestra

En la práctica, los valores de las medias y las matrices de covarianzas teóricos usados en los criterios de clasificación dados en la sección anterior serán desconocidos, por lo que tendrán que ser estimados. Estas estimaciones dependerán de las hipótesis de partida (normalidad, igualdad de matrices de covarianzas, etc...), hipótesis que en muchos casos deberán ser corroboradas mediante algún procedimiento cuando no se esté muy seguro de su validez.

En general, si estudiamos k variables numéricas $(Z_1, \ldots, Z_k)$ en m distintas poblaciones indicadas por una variable discreta G (grupo) para una muestra de n individuos (muestra de entrenamiento), tendremos una tabla de datos de la forma

	Z_1	...	Z_k	G
ω_1	$z_{1,1}$	...	$z_{1,k}$	g_1
...	...	...	...	...
ω_n	$z_{n,1}$	...	$z_{n,k}$	g_n

donde $g_i \in \{1, \ldots, m\}$ para todo i.

Para cada valor de $G = j$, esta tabla proporcionará una m.a.s. de la variable aleatoria k dimensional $Z = (Z_1, \ldots, Z_k)'$ condicionada por $G = j$ que, en muchas ocasiones, podremos suponer normal. Es decir, en realidad tendremos m muestras de m poblaciones normales k dimensionales $N(\mu_j, V_j)$.

Así, en la práctica, tendremos m medias muestrales teóricas y m matrices de covarianzas desconocidas, por lo que tendremos que estimarlas. Para dichas estimaciones usaremos:

$$\widehat{\mu}^{(j)} = \frac{1}{n_j} \sum_{i=1}^{n} \omega_j 1(G(\omega_i) = j)$$

$$\widehat{V}_j = \frac{1}{n_j - 1} \sum_{i=1}^{n} 1(G(\omega_i) = j)(\omega_i - \widehat{\mu}^{(j)})(\omega_i - \widehat{\mu}^{(j)})'$$

$$\omega_i = (Z_{1,i}, \ldots, Z_{k,i})'$$

$$n_j = \sum_{i=1}^{n} 1(G(\omega_i) = j)$$

$$n = n_1 + \cdots + n_m,$$

donde $1(G(\omega_i) = j)$ indica si el individuo i pertenece (1) o no (0) a la población j-ésima y, por lo tanto, n_j es el número de individuos de la muestra pertenecientes a la población j-ésima. Note que necesitamos $n_j > 1$ para todo j. Si $(Z|G = j)$ es normal, $\widehat{V}_j$ es insesgado para V_j, teniendo $(n_j - 1)\widehat{V}_j$ una distribución (en el muestreo) Wishart $W_k(n_j - 1, V_j)$.

Si suponemos que las matrices de covarianzas teóricas son todas iguales ($V_1 = \cdots = V_m = V$), entonces la matriz de covarianzas común V se aproximará mediante la matriz de covarianzas ponderada (pooled)

$$\widehat{V} = \frac{1}{n - m} \sum_{j=1}^{m} (n_j - 1)\widehat{V}_j$$

que será un estimador insesgado para V.

A partir de estos estimadores se pueden obtener estimaciones de las distintas funciones discriminantes y con ellas, obtener clasificaciones (empíricas) para nuevos individuos. Como los estimadores se aproximan a los verdaderos valores de los parámetros, las clasificaciones "se parecerán" (si n es grande) a las que se obtendrían usando los verdaderos parámetros. Por ejemplo, para el caso de dos poblaciones con la misma matriz de covarianzas, la función discriminante de Fisher se estimará mediante

$$\widehat{D} = \widehat{L}(Z) = \widehat{a}'Z = (\widehat{\mu}_X - \widehat{\mu}_Y)'\widehat{V}^{-1}Z.$$

De esta forma, la probabilidad del error tipo 1 se estimará mediante

$$\Pr(e_1) = \Pr\left(U < -\frac{1}{2}\widehat{\triangle}\right),$$

donde $U \equiv N_1(0,1)$ y

$$\widehat{\triangle} = \sqrt{(\widehat{\mu}_X - \widehat{\mu}_Y)'\widehat{V}^{-1}(\widehat{\mu}_X - \widehat{\mu}_Y)}$$

es la distancia de Mahalanobis muestral entre la medias (muestrales) de los grupos.

Bajo la hipótesis de normalidad, estos estimadores son asintóticamente insesgados. En Srivastava y Carter (1983, pag. 238) pueden verse otros estimadores basados en las distribuciones obtenidas para los distintos errores a partir de la hipótesis de normalidad.

Se procederá de forma similar en los otros casos. Por ejemplo, las Funciones Discriminantes Lineales (FDL) muestrales serán

$$\widehat{L}_i(z) = (\widehat{\mu}^{(i)})'\widehat{V}^{-1}z - (\widehat{\mu}^{(i)})'\widehat{V}^{-1}\widehat{\mu}^{(i)}/2,$$

clasificándose z en $G_i : \widehat{L}_i(z) \geq \widehat{L}_j(z)$ para todo j. Análogamente, las proyecciones canónicas muestrales serán $Z^* = \widehat{V}^{-1/2}Z$ y las funciones discriminantes cuadráticas (FDC o QDF) muestrales serán

$$\widehat{Q}_i(z) = c - 2\log \widehat{f}_i(z) = (z - \widehat{\mu}^{(i)})'\widehat{V}_i^{-1}(z - \widehat{\mu}^{(i)}) + \log \mid \widehat{V}_i \mid,$$

clasificándose z en $G_i : \widehat{Q}_i(z) \leq \widehat{Q}_j(z)$ para todo j.

Veamos otra forma de aproximar los errores de clasificación que no precisa de la hipótesis de normalidad.

5.3.1. Validación cruzada

Una forma de estimar las probabilidades de los posibles errores consiste en clasificar a los individuos de la muestra (de los que se conoce su verdadero grupo) usando las funciones discriminantes obtenidas y hacer un recuento de los individuos mal (o bien) clasificados según el grupo al que pertenecen y/o el grupo en el que son clasificados por error.

Cuando se clasifica a los individuos de los que se conoce su grupo se está usando la propia información que ellos han proporcionado a la función discriminante. Esto puede influir en las proporciones de individuos bien clasificados ya que si en la muestra hay un individuo con las mismas características que las del que se quiere clasificar, es bastante probable que coincidan los grupos.

Para evitar esto suele utilizarse la técnica denominada **validación cruzada** (CV=cross validation, también llamada leave-one-out o jackknife) que deja fuera del análisis (se tacha) al individuo que se quiere clasificar.

Es decir, para aplicar esta técnica a un individuo deberemos recalcular las funciones discriminantes excluyéndolo de la muestra y aplicárselas, observando si se clasifica o no en su verdadero grupo. Repitiendo esto con cada individuo (o con un número suficiente de ellos) obtendremos estimaciones de las probabilidades de clasificación errónea (o correcta).

Lógicamente este proceso conlleva numerosos cálculos por lo que solo se puede realizar ayudándonos de un ordenador e incluso, si la muestra es muy grande, reduciendo el número de individuos a los que se le aplica (o haciendo grupos). Los errores de estimación de las distintas proporciones (probabilidades) son fáciles de calcular ya que se trata de variables Binomiales y, por lo tanto, incluso podemos calcular los intervalos de confianza para las distintas proporciones. La ventaja de usar validación cruzada es que al ser una prueba empírica, no es necesaria ninguna comprobación estadística (no se necesitan hipótesis iniciales).

También se puede utilizar esta técnica para ver qué variables son las que más influyen a la hora de clasificar correctamente a los individuos. Para ello podemos eliminar variables y comprobar cómo afecta esta eliminación al porcentaje de correctos. También se puede realizar una análisis discriminante lineal con las variables estandarizadas $(X_i - \mu_i)/\sigma_i$ (sustituyendo las medias y las varianzas por sus estimaciones) y, en este caso, la variable que más influya será aquella que tenga un coeficiente mayor en valor absoluto en la función discriminante lineal de Fisher. La estandarización no influirá en la clasificación (se obtienen los mismos resultados) pero sí nos permitirá el poder comparar los coeficientes al tener las variables el mismo rango de variación. Si no estandarizamos, los coeficientes dependerán de las unidades usadas en cada variable.

5.4. Ejemplos

5.4.1. Ejemplo con dos grupos

En el primer ejemplo consideramos el objeto d del fichero `escarabajos.rda` (Aula virtual) que contiene una muestra de 40 escarabajos de dos especies diferentes (Haltica oleracea y Haltica carduorum) a los que se les han medido 4 variables. Para leer este archivo debemos teclear

```
load('f:/escarabajos.rda')
```

indicando la ruta completa en donde se encuentra el archivo. Para ver los datos basta teclear `d` (o `View(d)`). Las variables son: distancia desde el tórax al surco transversal X_1 (micras), longitud X_2 (0.01mm.), longitud de la base de las antenas secundarias X_3 y terciarias X_4 (en micras). La variable código (X_6) indica la especie a la que pertenece cada individuo (HO=1, HC=2). Puede observarse que hay un individuo (40) del que se desconoce la especie lo que en R se escribe como `NA`.

Podemos comenzar estudiando las variables por separado. Si queremos ver solo los datos de la variable `surco`, haremos: `d$surco` o `d[,1]`. Por ejemplo, para estudiar esta variable podemos comenzar calculando sus estadísticos básicos (medias, cuartiles y valores extremos) en cada grupo haciendo:

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

De esta forma observamos que la media de la variable `surco` es más grande en la especie HO (194.5) que en la HC (179.6) y que su valor en el escarabajo 40 (182.2) está más cerca de la media de la especie HC. También podemos representarla gráficamente tecleando:

```
plot(d$surco,d$codigo)
```

Si queremos que aparezca el escarabajo 40 podemos hacer:

```
text(d$surco[40],1.5,labels='e40')
```

obteniendo el gráfico de la Figura 5.7 (izquierda). En esta gráfica podemos observar que la variable `surco` parece un poco mayor en el grupo 1 (HO) pero que no discrimina (separa) bien a los grupos. Con esta variable no es sencillo clasificar al escarabajo 40 pero, si tenemos que elegir un grupo, lo incluiríamos en el grupo 2 (HC) ya que está más cerca de su media. Se obtiene una gráfica similar haciendo `plot(d$surco,d$especie)`. En este caso, R etiqueta los datos por orden alfabético (ASCII) con *=1, HC=2 y HO=3. Estudie las restantes variables.

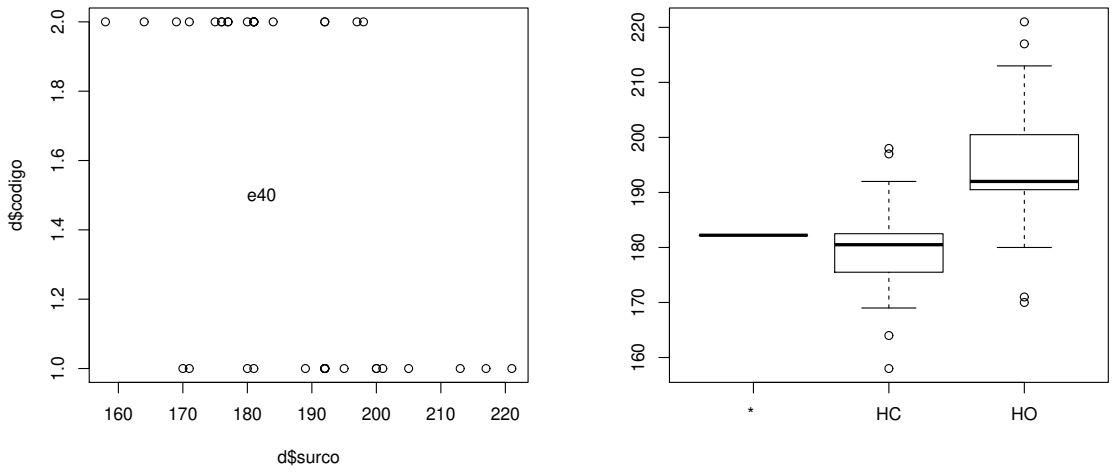


Figura 5.7: Gráficos de la variable `surco` por grupos.

También se pueden hacer los gráficos caja-bigote por grupos con:

```
boxplot(d$surco~d$especie)
```

(el símbolo $\sim$ se puede escribir pulsando simultáneamente Alt y 126) obteniendo el gráfico de la Figura 5.7 (derecha). Las cajas contienen al 50 % de los datos, los "bigotes" al 25 % y los puntos señalan valores atípicos (para una distribución normal). En este gráfico apreciamos que si usáramos solo la variable surco para clasificar, como las cajas no se solapan, más del 75 % de los individuos se clasificarían bien. También observamos que el escarabajo 40 estaría en la caja de la especie HC (por poco) pero que no sería un valor atípico en la HO. Estudie las restantes variables.

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

```
plot(d$surco,d$long,pch=as.integer(d$especie))
legend('topright',legend=c('e40','HC','HO'),pch=1:3)
```

obteniendo el gráfico de la Figura 5.8 en el que se observa que, con estas dos variables, los dos grupos están bastante separados, pero que el escarabajo 40 estaría entre ambos grupos por lo que no es sencillo clasificarlo. Estudie las otras dos variables. Se obtiene un gráfico similar haciendo

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

(`cex` indica el tamaño y `pos` la posición de la etiqueta).

Finalmente podemos hacer un gráfico similar al de la Figura 5.8, izquierda, pero usando las dos primeras componentes principales (ver tema anterior) que contienen información sobre todas las variables. Recordemos que las componentes principales se calculan con:

```
pca<-princomp(d[,1:4],cor=TRUE)
```

y que se pueden representar las dos primeras componentes por grupos haciendo:

```
biplot(pca,pc.biplot=TRUE,xlabs=d$especie)
```

El resultado puede verse en la Figura 5.8 (derecha). En este gráfico también se aprecia que los grupos se pueden separar bastante bien. Señalar no obstante que las dos primeras componentes principales no son necesariamente las mejores variables para clasificar a estos individuos.

Comenzaremos realizando un **Análisis Discriminante Lineal** (LDA). Para calcular la función discriminate lineal (FDL) de Fisher para distinguir entre dos grupos debemos suponer que sus matrices de covarianzas (teóricas) son iguales. Entonces, la FDL valdrá $L = L(Z) = \mathbf{a}'Z$ donde

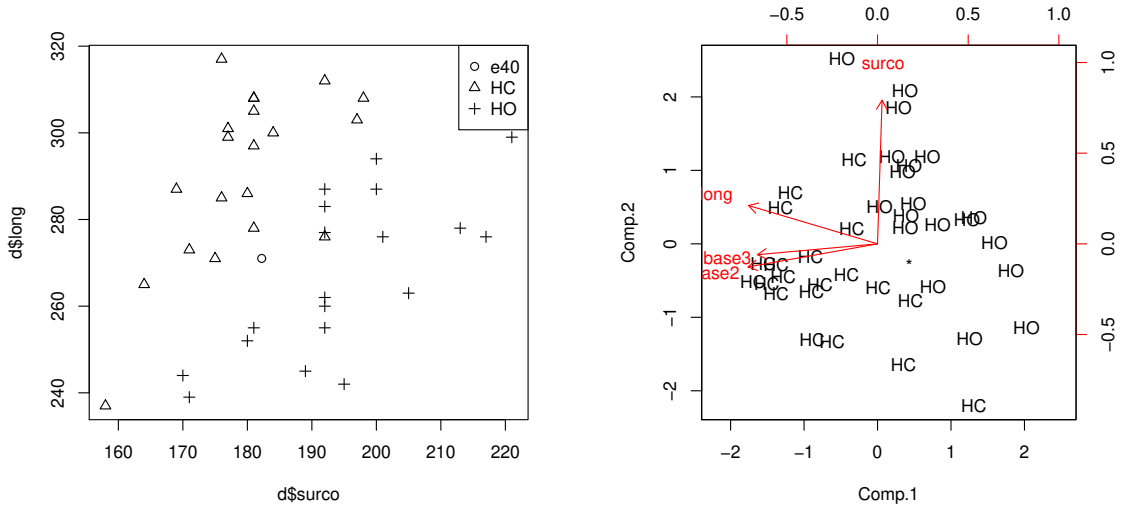


Figura 5.8: Gráfico conjunto de las variables `surco` y `long` (izquierda) y gráfico de las dos primeras componentes principales (derecha) por grupos.

$(Z_1, \ldots, Z_k)'$ son las medidas del individuo a clasificar y los coeficientes teóricos se calculan como

$$\mathbf{a}' = \lambda(\mu_X - \mu_Y)'V^{-1}, \tag{5.7}$$

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

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

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

Para calcular (estimar) **a** en R debemos cargar primero el "paquete" denominado **MASS**. Una vez cargado, debemos hacer:

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

Tecleando `LDA` comprobamos que las probabilidades de pertenencia a priori asignadas a cada grupo valen 0.5 (si no se especifica nada se computan como si los individuos fuesen una muestra, es decir, como $19/39 = 0.4871795$ (HO) y $20/39 = 0.5128205$ (HC) en este ejemplo), los vectores de medias de los grupos son:

$$\overline{X} = (194.4737, 267.0526, 137.3684, 185.9474)$$

y

$$\overline{Y} = (179.5500, 290.8000, 157.2000, 209.2500)$$

y que los (unos) coeficientes estimados de la FDL son

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

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

```
a<-LDA$scaling
```

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

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

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

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

obteniendo $L(\overline{X}) = 2.419488$. Análogamente, podemos calcular `mHC` obteniendo $L(\overline{Y}) = 6.120841$. De esta forma, haciendo `K<-(mHC+mHO)/2`, obtenemos $K = 4.270164$. Por lo tanto, la regla de decisión óptima según este criterio sería: Si $L(z) > K$, se clasifica como HC (grupo 2) y si no como HO (grupo 1). Podemos calcular las proyecciones de los 40 escarabajos haciendo:

```
z<-d[,1:4]
D<-1:40
for (i in 1:40) D[i]<-L(z[i,])
```

Tecleando `D` comprobamos que para el escarabajo 1 se obtiene $D[1] = 1.253859$ que, como es menor que $K = 4.270164$, nos conduciría a clasificarlo como del grupo HO (correctamente). Análogamente, para el escarabajo 40, obtenemos $D[40] = 3.968782$ que, como es menor que K, nos conduciría a clasificarlo como del grupo HO (con un margen pequeño). Podemos representar estas "puntuaciones discriminates" haciendo:

```
plot(D,d$codigo)
text(D,d$codigo,cex=0.7,pos=3,col='red')
```

Podemos incluir la puntuación del escarabajo 40 y la constante K en el gráfico haciendo:

```
text(D[40],1.5,labels='*')
```

```
text(D[40],1.5,labels='e40',cex=0.7, pos=3,col='red')
text(K,1.5,labels='|')
text(K,1.5,labels='K',cex=0.7,pos=3,col='red')
```

De esta forma se obtiene el gráfico de la Figura 5.9. En este gráfico se observa que el escarabajo 27 se clasificaría erróneamente y que el 40 se clasificaría en el grupo 1 (HO) pero con un margen pequeño (cerca de K).

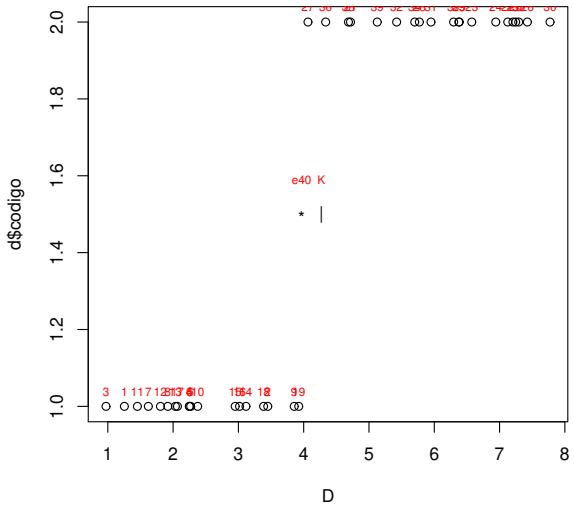


Figura 5.9: Gráfico de las puntuaciones discriminantes.

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

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

Las puntuaciones se obtienen haciendo `P` o `P$x`. Compruebe que coinciden con los valores de `D-K`. Estos valores se pueden representar como en la Figura 5.9 o en forma de histograma haciendo:

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

Haciendo `P$class` podemos ver en qué grupo se clasifican los 40 escarabajos. Tecleando

```
P$class==d[,6]->Resumen
```

podemos ver cuando la clasificación es correcta (para los 39 escarabajos de los que se conoce su grupo). Podemos hacer un recuento de estos resultados con:

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

Estos valores se pueden resumir con los valores de la Tabla 5.2 (matriz de confusión). Esta tabla sirve para comprobar si este procedimiento de clasificación es adecuado. En este caso, obtenemos buenos resultados ya que todos los individuos del primer grupo se clasifican correctamente y solo uno del grupo 2 (el escarabajo 27) se clasifica erróneamente como del grupo 1. Análogamente, comprobamos que todos los individuos clasificados como del grupo 2 se han clasificado correctamente pero que uno clasificado como del grupo 1, en realidad pertenecía al grupo 2 (de nuevo el 27).

Tabla 5.2: Resumen de los resultados de clasificación usando LDA sin validación cruzada.

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

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

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

donde π_1 y π_2 son las probabilidades a priori (0.5 en este caso) y f_1 y f_2 son las funciones de densidad normales estimadas de cada grupo. Aquí podemos ver que las probabilidades de pertenencia para el escarabajo 40 valen $\Pr(1|z = e40) = 0.7531572$ y $\Pr(2|z = e40) = 0.2468428$, que nos muestran que para un individuo de estas medidas la clasificación no es muy fiable. Evidentemente, los individuos se clasifican usando LDA en el grupo en el que resultan más verosímiles (ambos métodos son equivalentes).

La función `predict` también se puede usar para clasificar a nuevos individuos. Por ejemplo, para clasificar a un escarabajo con las medidas siguientes $z = (185, 280, 150, 200)$, haremos:

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

con lo que z se obtiene que se clasifica en el grupo 2, con una puntuación 0.3965766 y una probabilidad a posteriori de pertenencia al grupo 2 de 0.8127334. Compruebe que la puntuación coincide con $L(z) - K$.

Los valores de la Tabla 5.2 se pueden usar para estimar las proporciones de acierto en cada caso. Por ejemplo, la probabilidad de acierto global estimada es $38/39 = 0.974359$. Estas estimaciones

suelen dar valores ligeramente mayores que los reales ya que al clasificar a un individuo, se ha usado la información proporcionada por el propio individuo. Sin embargo, cuando se clasifica a un individuo nuevo (e40), éste no se usa en el procedimiento de clasificación. Para evitar esto, podemos usar la técnica denominada *validación cruzada* (*cross validation* o CV) que consiste en que, al clasificar a los individuos de los que se conoce su grupo, el individuo a clasificar no se usa en el procedimiento de clasificación (se tacha). Para hacer esto en R debemos teclear:

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

De esta forma, podemos comprobar que hay 3 escarabajos del grupo 2 que se clasifican mal (21, 27 y 36) con validación cruzada y el resumen correcto de clasificación sería el dado en la Tabla 5.3. En ella comprobamos, por ejemplo, que la verdadera (no sesgada) estimación de la probabilidad global de acierto es: $p_{LDA} = (19+17)/39 = 0.9230769$ (ligeramente menor que la calculada anteriormente sin CV). Al usar validación cruzada las probabilidades a posteriori de los individuos con grupos conocidos también cambian (ya que no se usan). Por ejemplo, para el escarabajo 21 obtenemos 0.5291374 y 0.4708626, mientras que antes eran 0.1606631 y 0.8393369. La validación cruzada no afecta a la clasificación de los individuos de los que se desconoce el grupo.

Tabla 5.3: Resumen de los resultados de clasificación usando LDA y validación cruzada.

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

Tanto las probabilidades de pertenencia, como las puntuaciones (la constante K) y las clasificaciones finales se verán influenciadas por las probabilidades a priori. Por ejemplo, si no indicamos las probabilidades a priori (es decir, asumimos que éstas se calculen a partir de la muestra), para el escarabajo 40 se obtiene una puntuación en $D-K$ de -0.34883551 y probabilidades a posteriori de $\Pr(1|e40) = 0.7434978$ y $\Pr(2|e40) = 0.2565022$, por lo que se sigue clasificando en el grupo 1. Los aciertos con estas probabilidades a priori son los mismos. Sin embargo, compruebe que con las probabilidades a priori 0.2 y 0.8, existe un escarabajo (19) del grupo 1 que se clasifica en el 2 y que el escarabajo 40 se clasifica en el grupo 2. La clasificación será óptima cuando se usen las probabilidades de pertenencia reales en cada grupo (que suelen ser desconocidas).

Cuando las variables usadas para clasificar sean normales (multivariantes) en cada grupo pero sus matrices de covarianzas (teóricas) no sean iguales, el procedimiento óptimo de clasificación será el proporcionado por el **Análisis Discriminante Cuadrático** (QDA) que consiste en comparar

sus funciones de densidad (verosimilitudes o probabilidades a posteriori) estimadas mediante:

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

En el LDA estas funciones se estimaban usando la estimación de la matriz de varianzas común S. Ahora note que las matrices de covarianzas de cada grupo se estiman usando solo los datos de ese grupo. Esto es equivalente a comparar las funciones discriminantes cuadráticas:

$$QDF_1(z) = (z-\overline{X})' S_1^{-1}(z-\overline{X}) + \log \mid S_1 \mid \tag{5.8}$$
$$QDF_2(z) = (z-\overline{Y})' S_2^{-1}(z-\overline{Y}) + \log \mid S_2 \mid, \tag{5.9}$$

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

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

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

Tecleando QDA comprobamos que en este procedimiento no aparecen los coeficientes de las QDF. Para obtener los coeficientes que convierten a los datos en esféricos y las constantes debemos teclear:

```
QDA$scaling
QDA$ldet
```

respectivamente. Compruebe que con la segunda opción se obtiene $\log \mid S_1 \mid = 19.41635$. La primera opción nos proporciona matrices triangulares U_i tales que $U_i U_i' = S_i^{-1}$. De esta forma, las funciones discriminantes cuadráticas se pueden calcular como:

$$QDF_1(z) = (U_1'z - U_1'\overline{X})'(U_1'z - U_1'\overline{X}) + \log \mid S_1 \mid \tag{5.10}$$
$$QDF_2(z) = (U_2'z - U_2'\overline{Y})'(U_2'z - U_2'\overline{Y}) + \log \mid S_2 \mid, \tag{5.11}$$

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

$$Cov(U_i'Z) = U_i' S_i U_i = U_i (U_i')^{-1} U_i^{-1} U_i = I$$

cuando Z pertenece al grupo i.

Para obtener las predicciones basadas en las probabilidades a posteriori podemos hacer:

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

Tecleando `P` comprobamos que solo hay un escarabajo mal clasificado (el 27) y que el escarabajo 40 se clasifica en el grupo 1 (como en el LDA). En este caso, las probabilidades de pertenencia valen 0.5817418 y 0.4182582 por lo que esta clasificación no es fiable.

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

```
table(d$codigo,P$class)
```

Para que esta tabla sea más realista debemos usar validación cruzada haciendo:

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

obteniendo los resultados de la Tabla 5.4. Los resultados son similares a los obtenidos con el LDA con una probabilidad global de acierto estimada de $p_{QDA} = 35/39 = 0.8974359$.

Tabla 5.4: Resumen de los resultados de clasificación usando QDA y validación cruzada.

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

La función `predict` también se puede usar para clasificar a nuevos individuos. Por ejemplo, para clasificar a un escarabajo con medidas

$$z = (185, 280, 150, 200),$$

haremos:

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

con lo que se obtiene que se clasifica en el grupo 2, con probabilidad a posteriori de pertenencia al grupo 2 de 0.9636754 con lo que esta clasificación sí es fiable (bajo la hipótesis de normalidad). En este ejemplo, los dos procedimientos dan buenos resultados y proporcionan las mismas clasificaciones para $e40$ y este z.

Finalmente, podemos realizar algunas comprobaciones sobre los modelos usados. En primer lugar, podemos tener la duda de si es mejor aplicar LDA o QDA. El primer método funciona bien si

las matrices de covarianzas teóricas son iguales y el segundo si los datos son normales en cada grupo. Se cumplan o no esas hipótesis, el método de validación cruzada nos proporciona estimaciones de las probabilidades de acierto en cada caso y nos permite la comparación de las técnicas LDA y QDA. También tenemos la opción de usar ambas técnicas y comprobar si los resultados coinciden.

Si queremos estudiar las hipótesis del LDA, la matriz de cuasicovarianzas del primer grupo se puede calcular con: `S1<-cov(d[1:19,1:4])`. También se pueden separar los datos del grupo 1 con:

```
d1<-d[d$especie=='HO',1:4]
```

y así, su matriz de cuasicovarianzas se calcula con `cov(d1)`. Análogamente, se calcula la del segundo grupo obteniéndose:

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

De esta forma, comprobamos que las matrices de covarianzas de los grupos son bastante diferentes y no parecen una estimación de una misma matriz de covarianzas V.

Para comprobar que las computaciones de R para los coeficientes del LDA dados en (5.7) son correctas podemos calcular la estimación de la matriz de covarianzas común V con

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

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

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

Su inversa se calcula con: `solve(S)->In`. Las medias de los grupos se calculan con:

```
LDA$means[1,]->m1
LDA$means[2,]->m2
```

(o con `mean(d1)`) y los coeficientes como `(m1-m2)%*%In->a` (donde `%*%` denota el producto de matrices en R) obteniendo

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

Para comprobar que son proporcionales a los obtenidos por R haremos: `LDA$scaling/t(a)` donde `t(a)` denota el vector traspuesto de `a`. Note que la constante de proporcionalidad es negativa (-0.2701715) y, por eso, la media del grupo 2 es mayor que la del 1.

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

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

y calculando los coeficientes con:

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

obtenemos: -1.2937164 (surco), 0.7809833 (long), 0.4182667 (base2) y 0.7084167 (base3). Por lo tanto, la variable que más influye (mejor discrimina) es surco (con un peso negativo) y la que menos base2.

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

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

comprobamos que hay 7 escarabajos que se clasifican mal. Eliminado las otras variables comprobamos que las mejores opciones son eliminar la variable *long* o la variable *base*2 (en ambos casos solo hay 2 escarabajos que se clasifiquen mal). Análogamente, podemos estudiar cuál es la mejor pareja de variables (o la variable individual) que mejor discriminan. Se puede aplicar un procedimiento similar en el QDA.

También podemos comprobar cómo se calculan las probabilidades a posteriori. Para ello debemos cargar el "paquete" **mvtnorm** y teclear:

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

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

`19*f1/(19*f1+20*f2)` obteniendo $\Pr(1|e40) = 0.743497$ (como anteriormente). Compruebe usando un procedimiento análogo (pero sustituyendo S por $S1$ y $S2$) las probabilidades a posteriori calculadas en el QDA.

Por último, señalar que para que estas "probabilidades" (verosimilitudes) sean correctas, las variables deben ser normales en cada grupo. Esta hipótesis también se usa en el QDA. Para hacer un test de normalidad multivariante (Shapiro-Wilk) debemos cargar el paquete: **mvnormtest** y hacer:

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

obteniendo un p-valor de 0.2013 por lo que el primer grupo pasaría el test de normalidad. Análogamente, para el segundo se obtiene un p-valor de 0.05769 que nos conduciría a aceptar la normalidad con $\alpha = 0.05$ por muy poco. Esto se puede deber al escarabajo 27 que, como hemos visto durante toda la práctica tiene unas medidas raras para ser del grupo 2. Los datos para este grupo se pueden ver haciendo `plot(d[20:39,1:4])`.

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

$$L_i(z) = z'S^{-1}m_i - m_i'S^{-1}m_i/2,$$

donde S es la matriz de covarianzas ponderada (calculada anteriormente) y m_i son las medias muestrales de los grupos. Para calcularlas en R haremos:

```
solve(S)%*%m1
solve(S)%*%m2
-0.5*t(m1)%*%solve(S)%*%m1
-0.5*t(m2)%*%solve(S)%*%m2
```

obteniendo:

$$L_1(z) = 0.9557217z_1 - 0.0208622z_2 + 0.6842504z_3 + 0.4353125z_4 - 177.6155,$$

$$L_2(z) = 0.6104728z_1 + 0.1095255z_2 + 0.7906842z_3 + 0.5786658z_4 - 193.4209.$$

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

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

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

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

De forma análoga, en el QDA se pueden calcular las funciones cuadráticas definidas por (5.9). En este caso, los individuos se incluyen en el grupo con el valor mínimo para esas funciones. Esto es equivalente a usar el método se máxima verosimilitud (probabilidad a posteriori) con matrices de covarianzas distintas por lo que se obtendrán las mismas clasificaciones que en la sección 3. Para el escarabajo 40 se obtiene:

$$QDF_1(z) = 22.76789,$$

$$QDF_2(z) = 23.42774,$$

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

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

$$D_1^2(z) = (z - \overline{X})' S_1^{-1} (z - \overline{X}),$$
$$D_2^2(z) = (z - \overline{Y})' S_2^{-1} (z - \overline{Y}),$$

obteniendo para el escarabajo 40: $D_1^2(z) = 3.351539$ y $D_2^2(z) = 3.860477$, por lo que se clasificaría en el grupo 1 (en el más cercano). En este caso, los métodos solo son equivalentes si los determinantes de las matrices de covarianzas de los grupos son iguales. Por lo tanto, se pueden obtener resultados diferentes de los obtenidos con las QDF. Estas distancias también se pueden calcular usando las transformaciones proporcionadas por las matrices U_i incluidas en `QDA$scaling`. Por ejemplo, los transformados en el grupo 2 de la media del grupo 2 y el escarabajo 40 son

$$U_2' \overline{Y} = (-17.792105, 4.306052, -6.051622, 9.862908)$$

y

$$U_2' z = (-18.056683, 2.772973, -5.519009, 8.787518),$$

respectivamente, y su distancia Euclídea al cuadrado es 3.860477.

Para calcular estas distancias en el LDA debemos reemplazar S_1 y S_2 por S obteniendo para el escarabajo 40: $D_1^2(z) = 2.801345$ y $D_2^2(z) = 5.03239$ por lo que se clasificaría en el grupo 1 (en el más cercano). En este caso, los métodos son equivalentes por lo que se obtendrán los mismos resultados de antes (con probabilidades a priori iguales). Cuando hay más de dos grupos, `LDA$scaling` proporciona la matriz U tal que $UU' = S^{-1}$, es decir, la transformación $U'z$ es esférica en todos los grupos. Con `predict(LDA)` podemos ver los transformados de los individuos que se pueden representar con `plot`. Si solo hay dos grupos, los transformados esféricos se proyectan sobre la recta formada por los transformados de las dos medias (función de Fisher). Veamos un ejemplo con tres grupos.

5.4.2. Ejemplo con tres grupos

Vamos a aplicar un DA a los datos del fichero `wine.R` (Aula virtual) que se pueden leer en R con: `source('F:/Ruta/wine.R')` o con

```
wine<-read.table('http://archive.ics.uci.edu/ml/machine-learning-databases
/wine/wine.data',sep=',')
```

Los datos contienen resultados de 13 diferentes análisis químicos en vinos de la misma región de Italia producidos tres cultivos diferentes (indicadas en la primera columna). Fuente: http://little-book-of-r-for-multivariate-analysis.readthedocs.org.

Como en el ejemplo anterior comenzaremos haciendo un estudio por separado de cada variable. Por ejemplo, para calcular las principales característica de la segunda variable por grupos haremos:

```
tapply(wine$V2, wine$V1, summary)
```

Estos datos se pueden representar con

```
plot(wine$V2,wine$V1)
```

obteniéndose la gráfica de la Figura 5.10. En la gráfica se observa que esta variable discrimina bien a los grupos 1 y 2 pero que los grupos 1 y 3 aparecerían muy mezclados. Estudiando las otras variables observamos que para separar a estos dos últimos grupos podemos usar la variable V8 o V13.

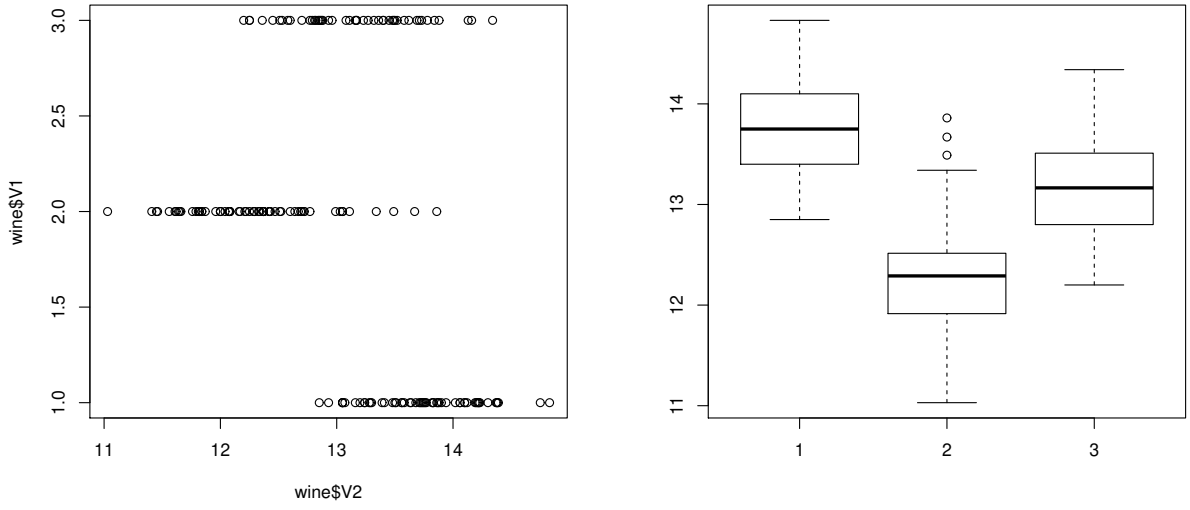


Figura 5.10: Gráficos de la segunda variable del fichero `wine` por grupos.

Otra opción sería realizar los gráficos caja-bigote por grupos. Por ejemplo, los de la Figura 5.10,

derecha, se obtienen haciendo:

```
boxplot(wine$V2∼wine$V1)
```

Para estudiar las variables por parejas en los grupos podemos hacer los gráficos bidimensionales con

```
plot(wine$V2,wine$V8,pch=as.integer(wine$V1))
legend('topright',legend=c('1','2','3'),pch=1:3)
```

obteniendo el gráfico de la Figura 5.11. En este gráfico se aprecia que, con estas dos variables, se pueden separar bastante bien a los tres grupos (aunque hay algunos elementos mezclados). Se obtiene un gráfico similar haciendo:

```
plot(wine$V2,wine$V8)
text(wine$V2,wine$V8,wine$V1,cex=0.7,pos=4,col='red')
```

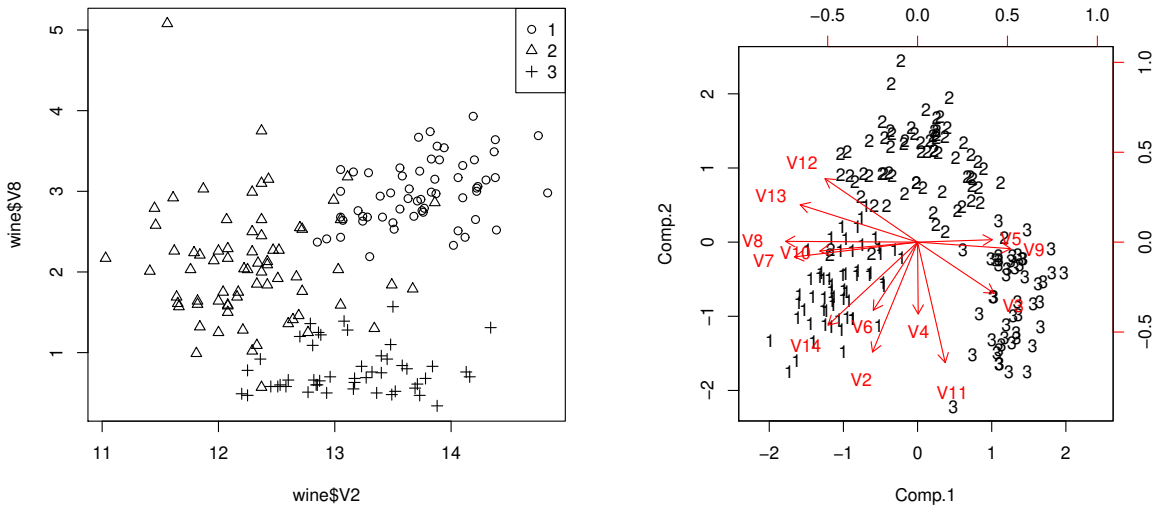


Figura 5.11: Gráficos bidimensionales de las variables V2 y V8 (izquierda) y de las dos primeras componentes principales (derecha) para los datos del fichero `wine` por grupos.

Otra opción es calcular las componentes principales (ver tema anterior) y representar los puntos de cada grupo. Para realizar el gráfico de las dos primeras componentes haremos:

```
pca<-princomp(wine[,2:14],cor=TRUE)
biplot(pca,pc.biplot=TRUE,xlabs=wine$V1)
```

obteniendo el gráfico de la Figura 5.11, derecha.

Tecleando `summary(pca)` podemos ver que estas dos primeras componentes mantienen un 55.4 % de la información de las 13 variables numéricas. La primera componente destaca a los vinos que tienen puntuaciones altas en V3, V5 y V9 y bajas en V7, V8, V10 y V13 (derecha del gráfico). Esta primera componente distingue perfectamente a los grupos 3 (con puntuaciones altas en Y_1) y 1 (con puntuaciones bajas en Y_1). La segunda componente destaca a los vinos que tienen puntuaciones altas en V12 y bajas en V2, V4, V6, V11 y V14 (arriba en el gráfico). Esta segunda componente distingue perfectamente a los grupos 2 (con puntuaciones altas en Y_2) y 1 y 3 (con puntuaciones bajas en Y_2). Algunos elementos del grupo 2 aparecen mezclados con los otros grupos. Para detectar estos elementos podemos ver las puntuaciones en Y_2 con `pca$scores[,2]` o representarlas con:

```
plot(pca$scores[,2])
text(pca$scores[,2],cex=0.7,pos=4,col='red')
```

obteniendo el gráfico de la Figura 5.12.

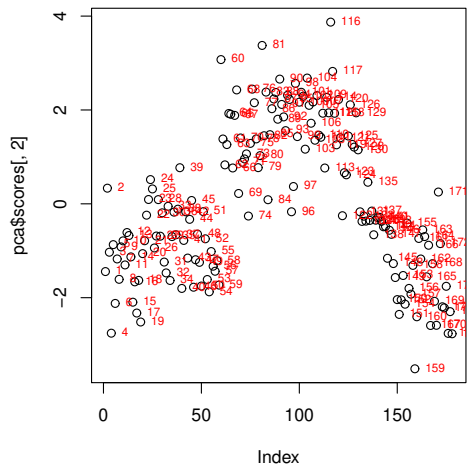


Figura 5.12: Gráfico de la segunda componente principal para los datos del fichero `wine`.

Para hacer una Análisis Discriminante (DA) debemos cargar primero el paquete **MASS** pinchando en el menú superior del programa R en *Paquetes-Cargar paquetes*. Una vez cargado, para hacer un LDA con probabilidades a priori iguales para todos los grupos debemos teclear:

```
LDA<-lda(wine[,2:14],wine[,1],prior=c(1/3,1/3,1/3))
```

Tecleando LDA podemos ver las medias de los grupos en las variables y los coeficientes para las proyecciones canónicas sobre el plano formado por las tres medias. Para guardar estos coeficientes podemos hacer `a<-LDA$scaling`. Podemos calcular las puntuaciones de los vinos con estos coeficientes haciendo

```
predict(LDA,wine[,2:14])->P
```

Tecleando `P` podemos ver los grupos donde se clasificarían los 178 casos, las probabilidades de pertenencia a cada grupo (bajo normalidad) y las puntuaciones canónicas proyectadas. Para representar estas proyecciones por grupos podemos hacer:

```
plot(P$x, pch = as.integer(wine$V1))
legend('bottomright',legend=c('1','2','3'),pch=1:3)
```

obteniendo el gráfico de la Figura 5.13 donde apreciamos que los grupos se pueden separar perfectamente con estas proyecciones (mucho mejor que con las componentes principales).

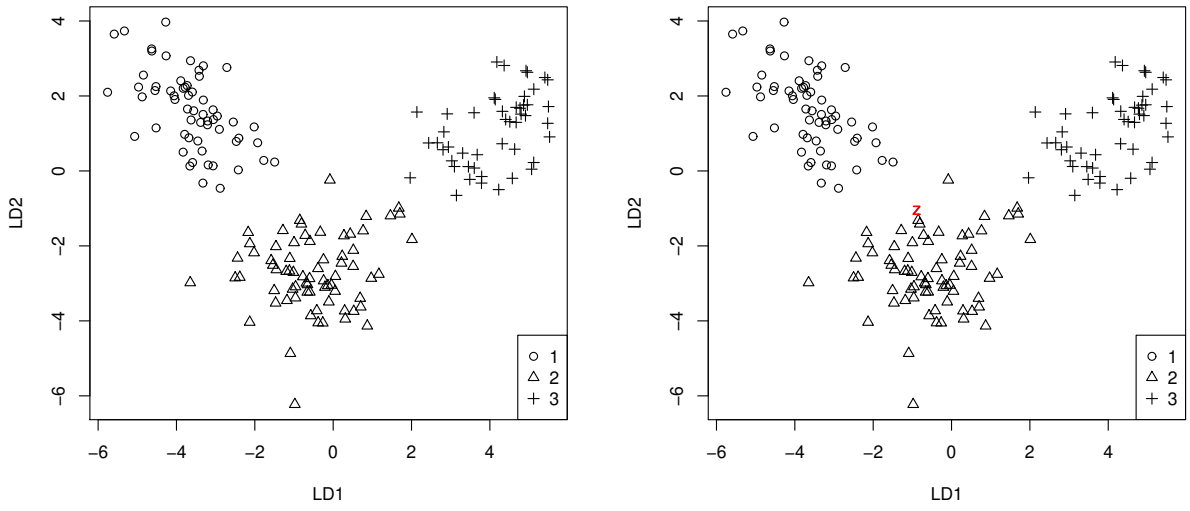


Figura 5.13: Gráfico de las puntuaciones canónicas para los datos del fichero `wine` por grupos incluyendo al proyectado del punto z en la derecha.

Podemos comprobar que todas las observaciones de la muestra se clasificarían correctamente haciendo

```
P$class==wine[,1]
```

o

```
table(P$class,wine[,1])
```

Como comentamos en la sección anterior estas estimaciones de las proporciones de acierto son ligeramente superiores a las reales porque al clasificar a un individuo se están usando sus propias medidas. Para evitar esto y dar estimaciones mejores (no sesgadas) podemos usar las técnicas de validación cruzada (que eliminan al individuo a clasificar) haciendo:

```
LDACV<-lda(wine[,2:14],wine[,1],prior=c(1/3,1/3,1/3),CV=TRUE)
table(LDACV$class,wine[,1])
```

observando que hay dos elementos del grupo 2 que se clasifican erróneamente en los grupos 1 y 3. Los resultados se pueden ver en la Tabla 5.5.

Tabla 5.5: Resumen de los resultados de clasificación usando LDA y validación cruzada.

Clasificados en el grupo	1	2	3	Total
Grupo verdadero grupo 1	59	0	0	59
Grupo verdadero grupo 2	1	69	1	71
Grupo verdadero grupo 3	0	0	48	48
Total	60	69	49	178

Para determinar qué individuos se clasifican mal podemos hacer: `LDACV$class` observando que corresponden a las filas 122 (se clasifica en el grupo 1 siendo del 2) y 97 (se clasifica en el grupo 3 siendo del 2). En todo caso, las proporciones de acierto (estimadas) son grandes en todos los casos. Por ejemplo, la proporción global de clasificación correcta es $176/178 = 0.988764$, la de clasificación correcta para individuos del grupo 2 es $69/71 = 0.971831$ y la de clasificación correcta para individuos clasificados en el primer grupo es $59/60 = 0.9833333$.

Si queremos predecir la forma de cultivo de un vino con medidas

$$z = (13, 2, 2, 19, 100, 2, 2, 0.3, 1.6, 5, 1, 3, 750)$$

teclearemos:

```
z<-c(13,2,2,19,100,2, 2, 0.3,1.6,5,1,3,750)
predict(LDA,z)
```

obteniendo que z se clasifica en el grupo 2 con una probabilidad de pertenencia a este grupo (bajo normalidad) de 0.996975. También proporciona las coordenadas para incluirlo en la Figura 5.13 haciendo

```
text(-0.8813738, -1.039406, labels ='z',col='red')
```

obteniéndose la Figura 5.13, derecha.

Análogamente, para hacer un análisis discriminante cuadrático (QDA) con probabilidades a priori iguales, debemos hacer:

```
QDA<-qda(wine[,2:14],wine[,1],prior=c(1/3,1/3,1/3))
```

Para obtener las predicciones para el punto z haremos:

```
predict(QDA,z)
```

obteniendo que, de nuevo, se clasifica en el grupo 2 con una probabilidad de pertenencia estimada (bajo normalidad) de 0.8616003. Para aplicar validación cruzada hacemos:

```
QDA<-qda(wine[,2:14],wine[,1],prior=c(1/3,1/3,1/3),CV=TRUE)
```

Para contar los aciertos y fallos haremos:

```
table(QDA$class,wine$V1)
```

obteniendo que solo un individuo del grupo 2 se clasifica mal en el grupo 1. Para ver cuál es podemos hacer: `QDA$class==wine$V1` obteniendo que el vino mal clasificado es el 82. De esta forma, concluimos que ambas formas de clasificación dan muy buenos resultados y que el vino a clasificar debe pertenecer al cultivo tipo 2 (o es muy parecido a los de ese tipo) con una probabilidad de acierto alta.

5.5. Problemas

1. Dadas tres poblaciones normales bidimensionales con medias $\mu_1 = (1,0)'$, $\mu_2 = (0,1)'$ y $\mu_3 = (0,0)'$ y matrices de covarianzas iguales a

$$V = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$$

   se pide:

(i) Obtener las funciones discriminantes lineales.

(ii) Clasificar a $z = (2, 2)'$.

(iii) Dibujar las regiones de clasificación para cada grupo.

2. Dadas tres poblaciones normales bidimensionales con medias $\mu_1 = (0, 0)'$, $\mu_2 = (1, 1)'$ y $\mu_3 = (2, 0)'$ y matrices de covarianzas iguales a

$$\begin{pmatrix} 5 & 2 \\ 2 & 1 \end{pmatrix},$$

se pide:

(i) Obtener las funciones discriminantes lineales.

(ii) Clasificar a $z = (1, 1/2)'$.

(iii) Obtener la función discriminante de Fisher, la constante K y el criterio de clasificación para distinguir entre las poblaciones 2 y 3.

(iv) Dibujar las regiones de clasificación para cada grupo.

3. Dadas tres poblaciones normales bivariantes con matriz de covarianzas común

$$V = \begin{pmatrix} 5 & 2 \\ 2 & 1 \end{pmatrix}$$

y medias $(-1, 2), (0, -2)$ y $(3, 5)$, respectivamente.

a) Obtener las funciones discriminantes.

b) Si $z = (2, 1)$, ¿en qué población se clasificaría?

4. Dadas tres poblaciones normales bivariantes con matriz de covarianzas común

$$V = \begin{pmatrix} 5 & 2 \\ 2 & 1 \end{pmatrix}$$

y medias $(1, 2), (0, 2)$ y $(3, 0)$, respectivamente.

a) Obtener las funciones discriminantes.

b) Si $z = (2, 1)$, ¿en qué población se clasificaría?

5. Dadas tres poblaciones normales bidimensionales con medias $\mu_1 = (0, 1)'$, $\mu_2 = (1, 0)'$ y $\mu_3 = (2, 2)'$ y matriz de covarianzas común

$$V = \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}$$

obtener las funciones discriminantes y clasificar a $z = (1, 3/2)'$.

6. Dadas tres poblaciones normales con matriz de covarianzas común

$$V = \begin{pmatrix} 6 & 2 \\ 2 & 1 \end{pmatrix}$$

y medias $(0,1), (1,0)$ y $(1,1)$, respectivamente, obtener las funciones discriminantes y el criterio de clasificación.

7. Dados dos vectores aleatorios bidimensionales con medias $(0,0)$ y $(3,0)$ y matrices de covarianzas

$$V_1 = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix} \text{ y } V_2 = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix},$$

se pide:

a) Calcular las funciones discriminantes cuadráticas.

b) Clasificar a $z = (1, -4)$ usando dichas funciones.

c) Representar gráficamente las regiones de clasificación.

Dados dos vectores aleatorios bidimensionales con medias $(0,0)$ y $(1,1)$ y matrices de covarianzas

$$V_1 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \text{ y } V_2 = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix},$$

respectivamente, se pide:

a) Calcular las funciones discriminantes cuadráticas.

b) Clasificar a $z = (1, 0)$ usando dichas funciones.

c) Representar gráficamente las regiones de clasificación para un punto cualquiera del plano $z = (x, y)$.

8. Dadas dos poblaciones normales bidimensionales con medias $\mu_1 = (1,0)'$ y $\mu_2 = (0,0)'$ y matrices de covarianzas

$$V_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1/2 \end{pmatrix} \text{ y } V_2 = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$$

se pide:

(i) Obtener las funciones discriminantes cuadráticas y clasificar a $z = (1,1)'$.

(ii) Clasificar a z usando el criterio de mínima distancia de Mahalanobis y representar las regiones de clasificación con este criterio para cada grupo.

9. Obtener un criterio de clasificación para dos poblaciones Exponenciales unidimensionales con medias distintas usando máxima verosimilitud. Calcular las probabilidades de error. ¿Cuál deberá ser el criterio de clasificación para que las probabilidades de ambos errores sean iguales?. Clasificar a $z = 1.5$ entre dos poblaciones exponenciales con medias 2 y 1 usando ambos criterios. (Indicación: La función de densidad de la distribución exponencial es $f(x) = (1/\mu)\exp(-x/\mu)$ para $x \geq 0$).

10. Aplicar un DA a los datos de las columnas 5-10 del objeto d del fichero `bears.rda`[1]) para estudiar si esas medidas sirven para determinar el sexo del oso. Las variables son: Head.L= longitud de la cabeza (pulgadas), Head.W=anchura de la cabeza (pulgadas), Neck.G=perímetro cuello (pulgadas), Length=altura (pulgadas), Chest.G=perímetro pecho (pulgadas), Weight=peso (libras). Fuente: Minitab15.

11. Aplicar un DA a los datos del objeto d del fichero `pulgas.rda`[1].

[1]Para leer este tipo de archivos en R teclear `load('f:/name.rda')` indicando la ruta completa en donde se encuentra el archivo y reemplazando `name` por el nombre del archivo.

6

Análisis cluster

En este capítulo mostramos cómo agrupar observaciones estableciendo grupos (o clusters) con las más similares. A diferencia de la Regresión Logistica o del Análisis Discriminante estudiados en los capítulos 3 y 5 (aprendizaje supervisado), en este caso no tenemos una muestra inicial donde se nos diga a qué grupo pertenece cada observación (aprendizaje no supervisado o automático). De hecho, en algunas ocasiones podemos decidir cuántos grupos queremos establecer. Según la Wikipedia: "El Análisis Cluster (CA) es la tarea principal de la minería de datos exploratoria y es una técnica común en el análisis de datos estadísticos".

6.1. Introducción

Como en los capítulos anteriores dispondremos de una muestra (o población) de n individuos (objetos) en los que hemos medido k variables numéricas $(X_1, \ldots, X_k)$. Sin embargo, en este caso, no dispondremos de una variable Y que nos diga a qué grupo (población) pertenece cada observación. Incluso, en algunos casos, no sabremos ni siquiera el número de grupos. De hecho, lo que haremos será determinar los valores de Y que nos asigne los grupos que minimicen una función costo adecuada.

Para ello tendremos que utilizar una función "distancia" que nos mida cómo de similares son dos observaciones (individuos). La elección de esta distancia es muy importante y la solución final dependerá de la distancia elegida. La más popular es la distancia Euclídea definida como

$$d_E(x, c) = \sqrt{(x-c)'(x-c)} = \sqrt{\sum_{j=1}^{k} (x_j - c_j)^2}$$

para todo $x, c \in \mathbb{R}^k$ (vectores columna). Aquí, habitualmente $x = (x_1, \ldots, x_n)$ representará un "individuo" y $c = (c_1, \ldots, c_n)$ el "centroide" de un grupo. En R se puede computar como:

```
dE<-function(x,C) sqrt(sum((x-C)*(x-C)))
```

`dE(x,C)`
donde los valores se deben incluir antes en los vectores columna `x` y `C` [1].

Otra opción es la distancia de Mahalanobis que usa la métrica de los datos. Se define como

$$d_M(x,c) = \sqrt{(x-c)'V^{-1}(x-c)},$$

donde $V = Cov(X_1,\ldots,X_n)$. El principal problema es que si hay grupos, esta matriz puede ser distinta en cada grupo. Incluso, aunque supongamos que todos los grupos tienen la misma matriz de covarianzas, éstos tendrán medias distintas y, como desconocemos los grupos, no podemos estimar V (como hacíamos en el Análisis Discriminante). Una solución es suponer inicialmete que todos los individuos están en un mismo grupo (población) y calcular (estimar) la media y la covarianza en ella. En R se puede calcular con

```
dM<-function(x,C,V) sqrt(sum(t(x-C)%*%solve(V)%*%(x-C)))
```

También se puede calcular con el comando `mahalanobis(x,C,V)` que proporciona el cuadrado de esta distancia. Obviamente, si $V = I$ (matriz identidad), se obtiene la distancia Euclídea que, por lo tanto, representará a v.a. independientes con varianza uno. En otros casos, la distancia de Mahalanobis tendrá en cuenta las varianzas de las variables y sus covarianzas (correlaciones o dependencia).

Las circunferencias (elipsoides) obtenidas con $d_M(x,\mu,V) = cte$ coincidirán con las curvas de nivel de la distribución normal multivariante $N_k(\mu,V)$ cuya función de densidad es

$$f(x) = \frac{1}{\sqrt{(2\pi)^k|V|}} \exp\left(-\frac{1}{2}(x-\mu)'V^{-1}(x-\mu)\right).$$

De esta forma, bajo este modelo y si conocemos V, el individuo con medidas x se asignará al grupo en dónde sea más verosímil, es decir, donde $f_i(x)$ sea máxima, siendo f_i la densidad $N_k(\mu_i,V)$ (tal y como hacíamos en AD). Ahora el problema es que no sabemos como estimar μ_i y V y tampoco sabemos si hay una V común.

Otras distancias interesantes son la distancia absoluta (Manhattan, de ciudad o geometría del taxista)

$$d_A(x,C) = \sum_{j=1}^{k} |x_j - C_j|$$

(que usa las cuadrículas como caminos), la distancia L_s

$$d_s(x,c) = \left(\sum_{j=1}^{k}(x_j-c_j)^s\right)^{1/s}$$

[1] En R no se puede usar la letra "c" ya que es la que se usa para definir los vectores columna. Lo mismo ocurre con la letra "q" (se usa para cerrar R o "T" y "F" que se usan para verdadero y falso).

para $s > 0$, o la distancia de Pearson

$$d_P(x, c) = \sqrt{\sum_{j=1}^{k} \left(\frac{x_j - c_j}{\sigma_i} \right)^2}$$

donde σ_i es la desviación típica de X_i. Este último caso es equivalente a estandarizar los datos usando $Z_i = X_i/\sigma_i$ o $Z_i = (X_i - \mu_i)/\sigma_i$ lo que nos asegura que las variables tendrán magnitudes similares aunque se usen unidades diferentes en ellas (ésto no ocurre en la distancia Euclídea). El principal problema es que desconocemos σ_i y μ_i que tendrán que ser estimados usando todos los datos (sin grupos). A pesar de este hándicap, es preferible cometer este error a usar variables con escalas o unidades muy diferentes. Obviamente, es equivalente a usar la distancia Euclidea con los datos estandarizados. La distancia no dependerá de las unidades usadas en cada variable (es invariante por cambio de escala).

Además de definir las distancias entre individuos, también tendremos que definir distancias de individuos a grupos o distancias entre grupos, lo que nos llevará a definir diversas funciones "coste" que determinarán diferentes soluciones finales. Éstas vendrán determinadas por el problema que queremos resolver. Por ejemplo, si queremos calcular la distancia de un individuo x a un grupo $\{z_i : i \in G\}$ formado por $m = |G|$ individuos podemos definir las distancias siguientes:

$$d_1(x, G) := d(x, C),\ C = \frac{1}{|G|} \sum_{i \in G} z_i$$

$$d_2(x, G) := \min_{i \in G} d(x, z_i),$$

$$d_3(x, G) := \max_{i \in G} d(x, z_i),$$

$$d_4(x, G) := \sum_{i \in G} d(x, z_i),$$

o

$$d_5(x, G) := \sum_{i \in G} d^2(x, z_i),$$

donde d es una distancia entre individuos. Otra opción interesante es calcular (o estimar) una función de densidad para los individuos de un mismo grupo y calcular las distancias como

$$d(x, G_j) = 1 - \frac{f_j(x)}{f_1(x) + \cdots + f_m(x)}.$$

Análogamente, para las distancias entre grupos se pueden usar:

$$D_1(G_1, G_2) = d(C_1, C_2),\ C_j = \frac{1}{|G_j|} \sum_{i \in G_j} z_i, j = 1, 2$$

$$D_2(G_1, G_2) = \min_{i \in G_1, j \in G_2} d(z_i, z_j),$$

$$D_3(G_1, G_2) = \max_{i \in G_1, j \in G_2} d(z_i, z_j)$$

$$D_4(G_1, G_2) = \frac{1}{|G_1||G_2|} \sum_{i \in G_1, j \in G_2} d(z_i, z_j)$$

o

$$D_5(G_1, G_2) = \frac{1}{|G_1||G_2|} \sum_{i \in G_1, j \in G_2} d^2(z_i, z_j).$$

En d_1 o en D_1 podemos utilizar otros "centroides" C_1 y C_2 distintos de la media de cada grupo.

Estas distancias entre grupos nos permitirán representar sus distancias y, posteriormente establecer a partir de qué nivel uniremos los grupos formando los gráficos denominados "dendogramas".

Finalmente debemos definir una función costo que trataremos de minimizar para obtener la solución óptima de ese problema. Por ejemplo, una vez asignados los n individuos a un grupo mediante una variable Y que nos indicará con $y_i = j$ que el individuo i se asigna al grupo j, podemos definir el costo

$$J(y) = \sum_j \sum_{i: y_i = j} d(x_i, G_j), \tag{6.1}$$

donde $\sum_{j:y_i=j} 1 = 1$ para todo i (cada elemento se asigna a un único grupo). También se pueden usar distancias al cuadrado. En este caso, tenemos que fijar un número máximo de grupos ya que si no, la solución óptima será tener n grupos (uno para cada elemento). Otra opción podría ser maximizar

$$J(y) = \sum_{i<j} d(G_i, G_j).$$

Todas estas opciones nos llevarán a problemas diferentes que tendrán que resolverse (cuando sea posible) usando sus técnicas específicas (la mayoría de Investigación Operativa). Estos métodos se pueden dividir en dos grandes grupos: los métodos jerárquicos y los no jerárquicos. Los primeros parten de la idea de juntar las unidades (individuos o grupos) más similares (cercanas). En los no jerárquicos estableceremos un determinado número de grupos e iremos asignando cada individuo al grupo más cercano. Veamos dos ejemplos que coinciden con los más utilizados en la práctica.

6.2. Método no jerárquico de las K-medias

El método de las K-medias (K-means) es sin duda el método no jerárquico más popular. Habitualmente usa la distancia Euclídea con los datos sin estandarizar (cuando tienen escalas similares) o estandarizados (distancia de Pearson, cuando tienen escalas diferentes) pero se puede aplicar a otras distancias.

En este caso tenemos que fijar un número de grupos predeterminado K con $1 < K \leq n/2$ (note que K es el número de grupos, k es el número de variables, n es el número de observaciones y que estos números pueden ser diferentes). Posteriormente podremos si debemos aumentar o disminuir K según la solución obtenida. El algoritmo se puede establecer como sigue.

Algoritmo 3.1: Análisis Cluster con K-medias

Paso 0: Determinar K centroides $C_1^0, \ldots, C_K^0 \in \mathbb{R}^k$ al azar.
Paso1: Formar el grupo G_j^m con las observaciones que estén más cercanas al centroide C_j^{m-1} para $j = 1, \ldots, K$.
Paso 2: Calcular el centroide C_j^m del grupo G_j^m definido como el punto que minimiza $\sum_{i \in G_j^m} d(x_i, C_j^m)$ o $\sum_{i \in G_j^m} d^2(x_i, C_j^m)$ para $j = 1, \ldots, K$.
Paso 3: Repetir pasos 1 y 2 hasta que no se produzcan cambios en los grupos del paso 1 o un número determinado de veces.

Si usamos la distancia Euclídea y el error cuadrático los centroides del paso 2 serán las medias aritméticas de los datos de cada grupo. De esta forma este paso es inmediato y en el paso 1 simplemente calculamos las distancias a estos K centroides (medias) asignando cada individuo al grupo del centroide más cercano (distancia d_1). Además, si usamos como función de coste la dada en (6.1) y hay cambios en los grupos, esta función es estrictamente decreciente en el paso 2. Como las opciones del paso 1 son finitas ($VR_m^K = K^m$), este algoritmo conducirá hasta una solución óptima local en un número finito de pasos, que puede ser muy grande. Para evitar este problema podemos aplicar el algoritmo varias veces con centroides iniciales diferentes y comparar las soluciones óptimas finales de cada algoritmo. Veremos que con unos pocos pasos podemos obtener soluciones muy buenas.

Como en capítulos anteriores usaremos unos datos inventados para mostrar el funcionamiento del algoritmo. Para ello usaremos los datos análizados previamente con regresión logística pero ahora supondremos que no conocemos los grupos de esa muestra. Los datos son los siguientes y pueden verse en la Figura 6.1.

i	X_1	X_2	Y
1	1	2	
2	2	1	
3	3	1	
4	2	2	
5	5	1	
6	5	3	
7	3	2	
8	4	3	
9	4	4	
10	5	4	

Observamos que tienen unidades similares (por lo que podremos usar la distancia Euclídea) y que parecen formar dos grupos diferentes. El objetivo es determinar la variable Y que nos asigne cada individuo a un grupo. La dejamos en blanco para señalar que en este caso no tenemos una muestra de entrenamiento y, por lo tanto, no podremos saber cuál es la solución óptima (que mejor clasifique a los individuos). Esto es lo que se denomina "análisis no supervisado" (o automático).

Para aplicar el algoritmo con $K = 2$ medias (grupos) elegimos dos centroides al azar (dentro de

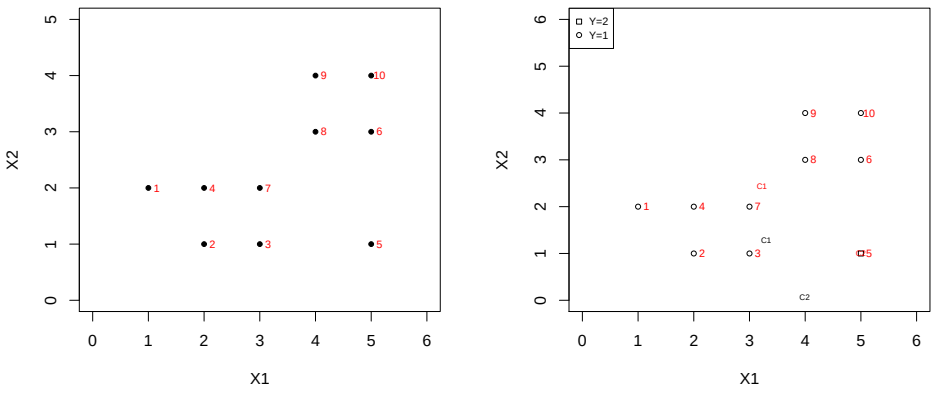


Figura 6.1: Individuos sin agrupamiento inicial (izquierda) y agrupados en el primer paso del algoritmo K-means con los centroides iniciales (negro) y los nuevos (rojo).

la zona donde están los individuos). Para ello usamos `runif(2,0,6)` obteniendo

$$C_1^0 = (2.482099, 2.270985)$$

y

$$C_2^0 = (3.851084, 5.344700).$$

Otra opción sería usar dos de esos puntos al azar. Con estos centroides obtenemos las distancias y agrupaciones siguientes:

i	X_1	X_2	d_1	d_2	Y
1	1	2	1.5066686	4.394963	1
2	2	1	1.3593463	4.722598	1
3	3	1	1.3724519	4.427275	1
4	2	2	0.5530392	3.822765	1
5	5	1	2.8205014	4.494043	1
6	5	3	2.6213142	2.611058	2
7	3	2	0.5845120	3.451284	1
8	4	3	1.6838902	2.349424	1
9	4	4	2.3007643	1.352921	2
10	5	4	3.0543933	1.768679	1

Lógicamente cada individuo se asigna al grupo más cercano (midiendo su distancia a cada centroide). El resultado puede verse en la Figura 6.1, derecha, donde además hemos incluido los centroides iniciales (negro) y los nuevos (rojo) que son las medias de los individuos de cada grupo. Estos centroides son

$$C_1^1 = (2.857143, 1.714286)$$

y

$$C_2^1 = (4.666667, 3.666667).$$

Repetimos los cálculos con los nuevos centroides obteniendo la gráfica de la Figura 6.2,izquierda, y los nuevos centroides (medias)

$$C_1^2 = (2.666667, 1.500000)$$

y

$$C_2^2 = (4.5, 3.5).$$

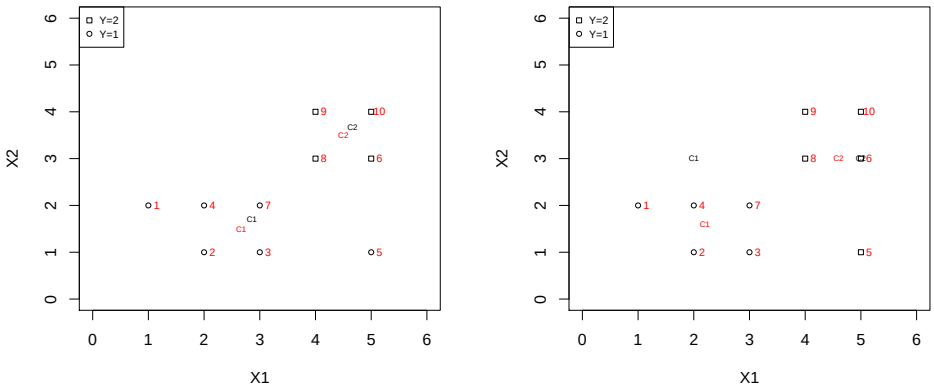


Figura 6.2: Individuos agrupados en el segundo paso (izquierda) y los obtenidos con otros centroides iniciales (derecha).

En la siguiente iteración los grupos no varían por lo que los centroides son los mismos y el algoritmo se detiene.

Este algoritmo puede depender de los valores iniciales. Por ejemplo, si tomamos los valores iniciales

$$C_1^0 = (2, 3)$$

y

$$C_2^0 = (5, 3),$$

obtenemos los grupos de la Figura 6.2, derecha con centroides finales

$$C_1 = (2.2, 1.6)$$

y

$$C_2 = (4.6, 3.0).$$

En este caso obtenemos los mismos grupos que se suponían inicialmente en Regresión Logística.

Para comparar ambas soluciones podemos utilizar diversas medidas. Por ejemplo podíamos usar (6.1) obteniendo

$$J(y_{sol1}) = 9.823299 > J(y_{sol2}) = 9.521839$$

con distancias Euclídeas y

$$J^*(y_{sol1}) = 12.83333 > J^*(y_{sol2}) = 11.2$$

con distancias Euclídeas al cuadrado. En ambos casos la solución segunda parece dar mejores resultados.

Tarea 6.1: Programe este algoritmo y compruebe esos resultados. Pruebe con otros valores iniciales y con $K = 3$ grupos.

El algoritmo K-means se puede ejecutar de forma automática en R con el comando `Kmean`. Por defecto, usa el algoritmo de Hartigan and Wong (1979). Para ejecutarlo en este ejemplo basta teclear:

```
X1<-c(1,2,3,2,5,5,3,4,4,5)
X2<-c(2,1,1,2,1,3,2,3,4,4)
d<-data.frame(X1,X2)
CA<-kmeans(d,2)
```

Tecleando `CA` podemos ver los grupos que coinciden con los obtenidos en la segunda solución anterior (óptima). También se pueden obtener y guardar con

```
CA$centers
CA1<-CA$centers[1,]
CA2<-CA$centers[2,]
```

Análogamente, los grupos se obtienen con

```
CA$cluster
```

La solución coincide con la representada en la Figura 6.2, derecha.

Las sumas de las distancias al cuadrado en los grupos se obtienen con

```
CA$withinss
```

obteniendo $7.2 + 4.0 = 11.2$ (como antes). El comando

```
CA$totss
```

proporciona la suma de las distancias al cuadrado sin grupos (o con un único grupo) obteniéndose 30.5 por lo que al agruparlos se ha producido una disminución del

$$1 - \frac{11.2}{30.5} = 0.6327869$$

por uno, es decir, con dos grupos la "variabilidad" se reduce un 63.28 %.

Los gráficos con 2 y 3 grupos pueden verse en la Figura 6.3. El código para generar el gráfico con tres grupos es el siguiente (note que el grupo dos solo tiene un dato que, por lo tanto, será su centroide):

```
K<-3
CA<-kmeans(d,K)
plot(X1,X2,xlab="X1",ylab="X2",pch=as.integer(CA$cluster),
   xlim=c(0,6),ylim=c(0,6),cex=0.7)
legend('topleft',legend=c('Y=1','Y=2','Y=3'),pch=1:K,cex=0.7)
text(CA$centers[1,1]+0.15,CA$centers[1,2],'C1',cex=0.5,col='red')
text(CA$centers[2,1]-0.20,CA$centers[2,2],'C2',cex=0.5,col='red')
text(CA$centers[3,1]+0.15,CA$centers[3,2],'C3',cex=0.5,col='red')
text(CA$centers[1,1],CA$centers[1,2],'*',cex=1)
text(CA$centers[3,1],CA$centers[3,2],'*',cex=1)
text(X1+0.15,X2,1:n,cex=0.7,col='red')
```

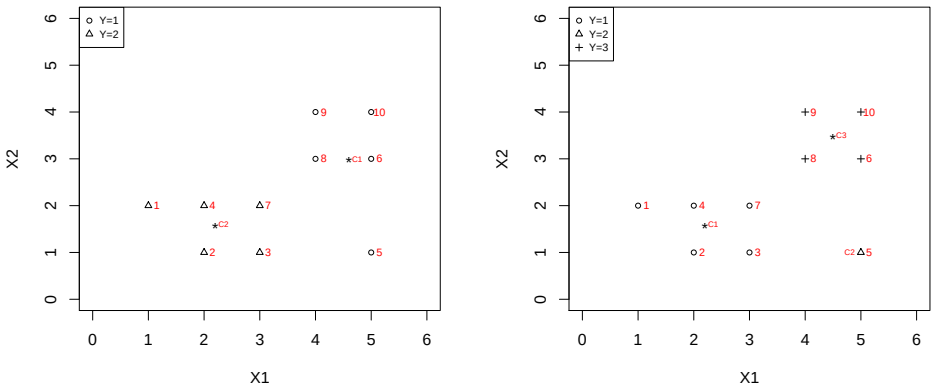


Figura 6.3: Individuos agrupados con Kmeans en 2 y 3 grupos.

6.3. Método jerárquico

En este caso no fijamos de antemano un número de grupos. Lo que haremos es, dada una distancia, definir un índice de similaridad entre dos observaciones con

$$I(x^{(i)}, x^{(j)}) = 1 - \frac{d(x^{(i)}, x^{(j)})}{\max_{r,s} d(x^{(r)}, x^{(s)})} \in [0, 1].$$

Se puede dar una definición análoga para grupos. La idea es establecer clasificaciones calculando los índices de similitud (o distancias) que se van obteniendo. Finalmente, dependiendo del índice de similitud elegido, obtendremos un número determinado de grupos (uniendo los que tienen similitud menor que ese índice).

Consideraremos dos algoritmos. En el primero partiremos de n grupos formados por un individuo cada uno. En el primer paso uniremos las dos observaciones más cercanas (distancia mínima) que serán las que tengan un índice de similitud mayor. Recalculamos las distancias para estos grupos y unimos los dos grupos más cercanos, continuamos así hasta conseguir un único grupo.

En el segundo, procederemos de forma inversa, es decir partiremos de un único grupo que separaremos en dos de forma que las distancias entre estos dos grupos sea máxima (o las distancias a esos dos grupos de sus individuos sea mínima). En el siguiente paso formaremos un tercer grupo tomando individuos de los grupos 1 y 2 con un criterio similar. Procederemos así hasta conseguir n grupos. Claramente, este método es más lento que el anterior.

Para ver un ejemplo analizaremos los datos de la sección anterior usando el primer método. En primer lugar calculamos las distancias Euclídeas entre todos los individuos:

D	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}
O_1	0	1.41	2.24	**1**	4.12	4.12	2	3.16	3.61	**4.47**
O_2	1.41	0	1	1	3	3.61	1.41	2.83	3.61	4.24
O_3	2.24	1	0	1.41	2	2.83	1	2.24	3.16	3.61
O_4	1	1	1.41	0	3.16	3.16	1	2.24	2.83	3.61
O_5	4.12	3	2	3.16	0	2	2.24	2.24	3.16	3
O_6	4.12	3.61	2.83	3.16	2	0	2.24	1	1.41	1
O_7	2	1.41	1	1	2.24	2.24	0	1.41	2.24	2.83
O_8	3.16	2.83	2.24	2.24	2.24	1	1.41	0	1	1.41
O_9	3.61	3.61	3.16	2.83	3.16	1.41	2.24	1	0	1
O_{10}	**4.47**	4.24	3.61	3.61	3	1	2.83	1.41	1	0

El código para calcularlas es el siguiente:

```
n<-length(X1)
D<-matrix(NA,n,n)
for (i in 1:n) {
   for (j in 1:n) D[i,j]<-dE(d[i,],d[j,])
}
```

Observamos que el máximo se alcanza en $D_{1,10} = 4.47$ y el mínimo eliminando los ceros es 1 y se alcanza en varios puntos (esto se debe a que los puntos son discretos). El primero que detecta el programa es $D_{1,4} = 1$ por lo que sería el primer grupo $G_1 = \{1, 4\}$. El índice de similaridad será

$$I(x^{(1)}, x^{(4)}) = 1 - \frac{d(x^{(1)}, x^{(4)})}{\text{máx}_{r,s}\, d(x^{(r)}, x^{(s)})} = 1 - \frac{1}{4.472136} = 0.7763932.$$

El siguiente paso dependerá de la distancia entre grupos elegida. Si queremos mantener esas distancias y detectar esos empates debemos elegir la distancia del "vecino más próximo" (D_2). Todas las demás nos darán valores mayores. Con esta distancia (tras varias iteraciones) uniríamos todos los puntos que estén a distancia 1 de alguno del grupo obteniendo:

$$G_1 = \{1, 2, 3, 4, 7\},\ G_2 = \{5\}, G_3 = \{6, 8, 9, 10\}.$$

El resultado puede verse en la Figura 6.4, izquierda.

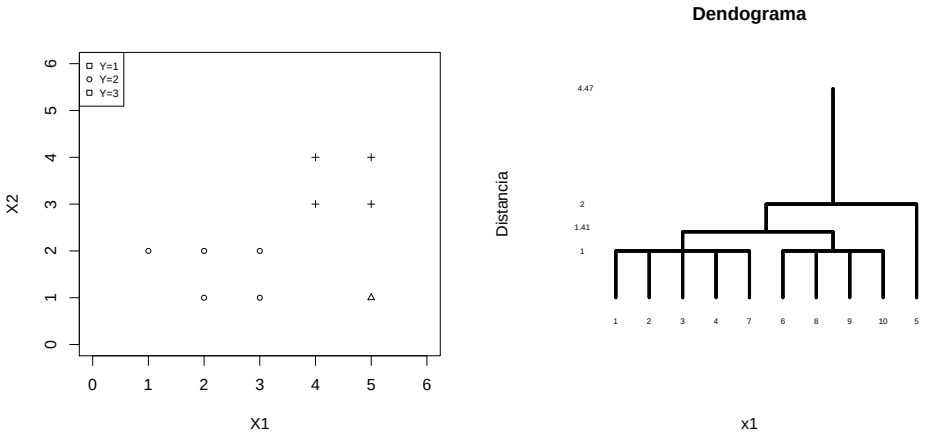


Figura 6.4: Individuos agrupados en el primer paso usando el método jerárquico con la distancia del vecino más próximo (izquierda) y dendograma (derecha).

La matriz de distancias D_2 para estos tres grupos será

D_2	G_1	G_2	G_3
G_1	0	2	**1.41**
G_2	2	0	2
G_3	**1.41**	2	0

El mínimo se alcanza en $d(G_1, G_3) = d(x^{(7)}, x^{(8)}) = 1.41$ por lo que en el segundo paso uniríamos los grupos G_1 y G_3 que tendrían un índice de similaridad

$$I(x^{(7)}, x^{(8)}) = 1 - \frac{d(x^{(7)}, x^{(8)})}{\text{máx}_{r,s}\, d(x^{(r)}, x^{(s)})} = 1 - \frac{1.41}{4.472136} = 0.6847144.$$

En el último paso uniríamos el grupo G_2 con $G_1 \cup G_3$ a distancia 2 y similaridad

$$I(x^{(3)}, x^{(5)}) = 1 - \frac{d(x^{(3)}, x^{(5)})}{\text{máx}_{r,s}\, d(x^{(r)}, x^{(s)})} = 1 - \frac{2}{4.472136} = 0.5527864.$$

El dendograma debe mostrar estas uniones usando las distancias o los índices de similaridad. El gráfico con distancias (vecino más próximo) puede verse en la Figura 6.4, derecha. Observamos que a distancia cero (similaridad 1) tenemos diez grupos, a distancia uno (similaridad 0.7763932), tres, a distancia 1.41 (similaridad 0.6847144), dos y por último, a distancia dos (similaridad 0.5527864) un único grupo (con el vecino más próximo) siendo 4.47 la distancia máxima entre individuos (similaridad cero).

Obviamente, con otras distancias y/o usando el segundo método (que parte de un único grupo que se separa en dos) podemos obtener resultados diferentes. También observamos que los resultados no tienen por qué coincidir con los obtenidos con el algoritmo K-medias. La elección de un método u otro dependerá de los datos que tengamos y del problema que se quiera resolver ("costo"). Por ejemplo si lo que queremos es agrupar a los usuarios para ser atendidos por centros deberemos usar distancias basadas en centroides que representarán dónde se situarán (aproximadamente) esos centroides. Por contra, si lo que queremos es simplemente clasificar empresas o países según sus características, estos centroides no serán tan importantes.

Para realizar este agrupamiento de forma automática en R podemos usar el comando `hclust`. En primer lugar calcularemos las distancias con

```
D <- dist(d, method = 'euclidean')
```

representadas en forma de vector. Para verlo en forma de matriz usaremos el comando `as.matrix(D)[1:10, 1:10]`. Ahora, para hacer un CA basta teclear

```
CA2<- hclust(D, method='complete')
```

Existen diversos métodos que pueden verse con `help(hclust)`. Nosotros hemos usado "complete". Si queremos obtener $K = 4$ grupos podemos hacer

```
grupos<- cutree(CA2, k=4)
```

Para representar esos grupos y el dendograma como en la Figura 6.5 podemos hacer:

```
plot(X1,X2,pch=as.integer(grupos),xlim=c(0,6),ylim=c(0,6),cex=0.7)
legend('topleft',legend=c('Y=1','Y=2','Y=3','Y=4'),pch=1:4,cex=0.7)
text(X1+0.15,X2,1:n,cex=0.7,col='red')
plot(CA2,cex=0.8,main='Dendograma',ylab='Distancia',xlab='Observaciones',sub='')
abline(h=2,col='red')
```

Note que en el dendograma primero se unen los puntos que están a distancia uno y, posteriormente se calculan las distancias entre grupos usando la distancia al vecino más lejano (D_3). Por

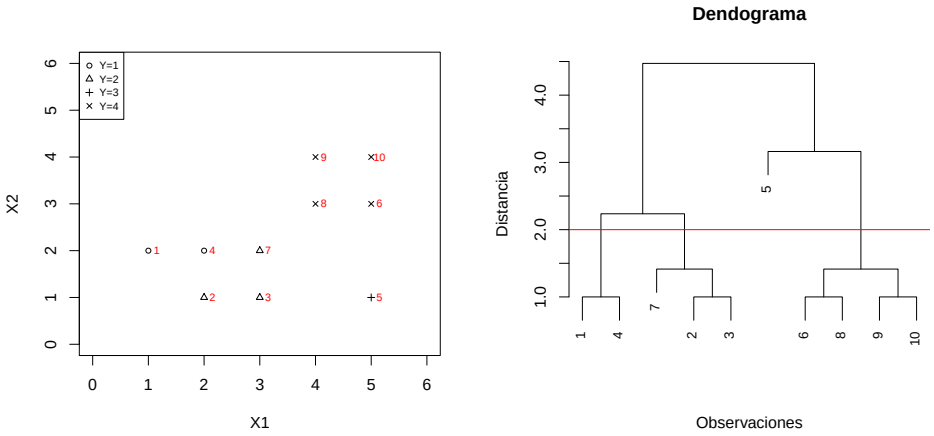


Figura 6.5: Cluster análisis con cuatro grupos (izquierda) y dendograma (derecha). La línea roja en el dendograma representa la distancia que nos da 4 grupos.

ejemplo, la distancia de la observación 7 al grupo $\{2, 3\}$ es

```
dE(d[7,],d[2,])
```

es decir, 1.414214. Lo mismo ocurre con la distancia entre los grupos $\{6, 8\}$ y $\{9, 10\}$. La distancia mayor es la de la observación 5 al grupo $\{6, 8, 9, 10\}$ obtenida con

```
dE(d[5,],d[9,])
```

y que vale 3.162278.

6.4. Ejemplo

Como en capítulos anteriores usaremos los datos del archivo `iris` de R. Para cargarlo y calcular las distancias Euclídeas haremos

```
d<-iris[,1:4]
D <- dist(d, method = 'euclidean')
D[1]
dE(d[1,],d[2,])
```

Con la última orden comprobamos que el primer dato de D es la distancia Euclídea entre las

dos primeras flores. Así, comprobamos que D es un vector de longitud

$$\frac{n(n-1)}{2} = \frac{150 * 149}{2} = 11175$$

donde se han omitido las distancias de un individuo consigo mismo y las distancias simétricas. Por lo tanto, `D[2]` contendrá la distancia entre las flores 1 y 3, etc. Las distintas distancias incluidas pueden verse con `help(dist)`. Si queremos tener una matriz con los tres primeros datos haremos

```
as.matrix(D)[1:3, 1:3]
```

Los máximos y mínimos pueden calcularse con

```
max(D)
min(D)
which.max(D)
which.min(D)
```

El máximo es 7.085196 y el mínimo 0 (es decir, hay dos flores distintas con las mismas medidas). Estos valores se alcanzan con las flores 14 y 119 (el máximo) y las flores 102 y 143. Para detectarlas podemos hacer:

```
which.max(D)
sum(149:136)
150-136
150+which.max(D)-sum(149:136)
dE(d[14,],d[119,])
max(D)
```

Las flores con sus distintas especies pueden representarse con dos gráficos como en el Capítulo 3. El resultado puede verse en la Figura 6.6. También se pueden representar en un único gráfico usando PCA.

Para aplicar un análisis cluster (CA en inglés) con K-means y tres grupos teclearemos:

```
CA<-kmeans(d,3)
```

Tecleando `CA` podemos ver los centroides de los tres grupos

$$C_1 = (5.006000, 3.428000, 1.462000, 0.246000),$$

$$C_2 = (6.850000, 3.073684, 5.742105, 2.071053)$$

y

$$C_3 = (5.901613, 2.748387, 4.393548, 1.433871)$$

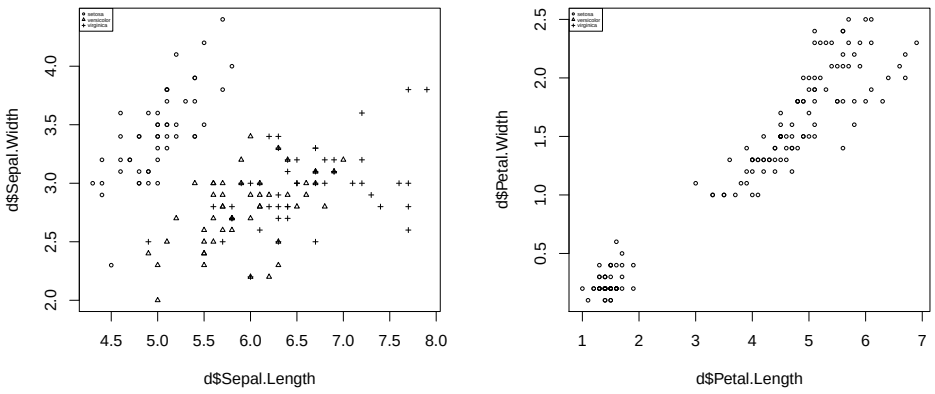


Figura 6.6: Flores del archivo `iris` clasificadas por especies.

y las asignaciones para cada flor. Si queremos guardar estas asignaciones haremos

```
y<-CA$cluster
```

Para realizar los gráficos con estos grupos haremos:

```
plot(d$Sepal.Length,d$Sepal.Width,pch=as.integer(y),cex=0.5)
legend('topleft',legend=c('1','2','3'),pch=1:3,cex=0.3)
text(5.006000,3.428000,'C1',cex=0.5,col='red')
text(6.850000,3.073684,'C2',cex=0.5,col='red')
text(5.901613,2.748387,'C3',cex=0.5,col='red')
plot(d$Petal.Length,d$Petal.Width,pch=as.integer(y),cex=0.5)
legend('topleft',legend=c('1','2','3'),pch=1:3,cex=0.3)
text(1.462000,0.246000,'C1',cex=0.5,col='red')
text( 5.742105,2.071053,'C2',cex=0.5,col='red')
text( 4.393548,1.433871,'C3',cex=0.5,col='red')
```

El resultado puede verse en la Figura 6.7. Allí observamos que los grupos se establecen principalmente en las dos últimas variables.

Para aplicar un método jerárquico podemos usar

```
CA2<-hclust(D, method='complete')
plot(CA2,cex=0.2)
```

La segunda orden sirve para representar el dendograma que puede verse en la Figura 6.8.

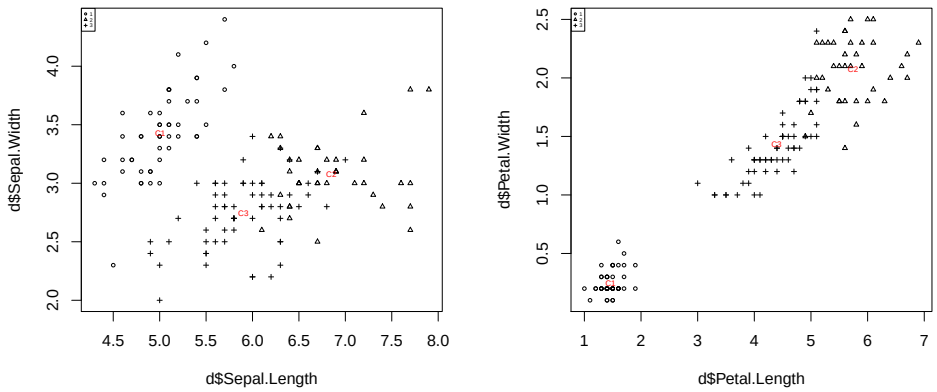


Figura 6.7: Flores del archivo `iris` clasificadas por usando K-means en tres grupos.

Observamos que las dos primeras flores que se unen son la 102 y la 143 cuya distancia es cero como podemos comprobar con `dE(d[102,],d[143,])`.

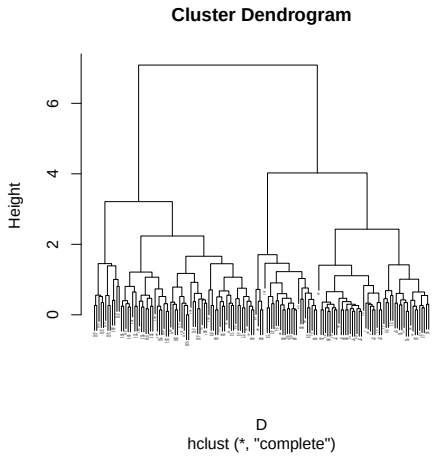


Figura 6.8: Flores del archivo `iris` clasificadas jerárquicamente en tres grupos.

Existen diversos métodos que pueden verse con `help(hclust)`. Nosotros hemos usado "complete". Si queremos obtener un número determinado de grupos podemos hacer

```
grupos<- cutree(CA2, k=3)
```

En este caso obtenemos tres grupos que se han representado en la Figura 6.9. Haciendo

```
sum(y==grupos)
```

comprobamos que 116 flores se clasifican en los mismos grupos del algoritmo K-means.

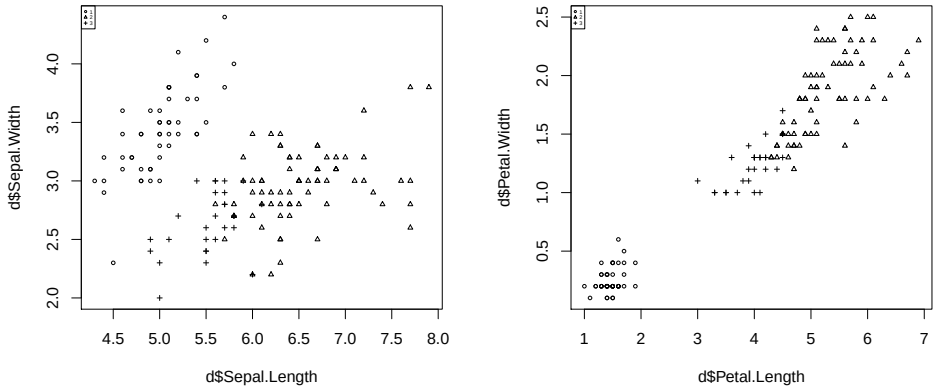


Figura 6.9: Flores del archivo `iris` clasificadas jerárquicamente en tres grupos.

Tarea 6.2: Aplicar un CA con tres grupos usando los datos de `iris` estandarizados. Comparar con los resultados sin estandarizar. Repetir esta comparación usando cinco grupos.

6.5. Problemas

1. Aplicar un análisis cluster usando K-means a una muestra de tamaño 10 con dos variables y dos grupos. Calcular el índice de coste y realizar las gráficas pertinentes. Comprobar los resultados usando el comando `kmeans` de R.

2. Aplicar un análisis cluster jerárquico a una muestra de tamaño 10 con dos variables. Calcular el índice de coste y realizar el dendograma. Comprobar los resultados usando el comando `hclust` de R.

3. Aplicar un análisis cluster al fichero `USArrests` clasificando a los estados según sus índices de delitos (usar help para ver las descripciones de las variables).

4. Aplicar un análisis cluster al fichero `heptathlon` clasificando a los atletas en dos grupos (usar help para ver las descripciones de las variables).

5. Aplicar un análisis cluster al fichero `decatlon.rda` clasificando a los atletas en dos grupos.

6. Aplicar un análisis cluster al fichero `madres.rda` clasificando a los atletas en dos grupos.
7. Aplicar un CA a un conjunto de datos reales aplicando todas las técnicas que consideres oportunas.

7

Apéndice

7.1. Formulario

Álgebra:

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in M_{n\times 1}, A = (a_{i,j}) = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,m} \\ ... & ... & ... & ... \\ a_{n,1} & a_{n,2} & & a_{n,m} \end{pmatrix} \in M_{n\times m}$$

1. Traspuesta: $x' = (x_1, \dots, x_n)$, $A' = (a_{j,i})$
2. Norma euclídea: $\| x \| = \sqrt{(x'x)} = \sqrt{x_1^2 + \dots + x_n^2}$
3. Desigualdad de Cauchy-Schwarz: $(x'y)^2 \leq (x'x)(y'y)$, es decir,

   $$| x'y | \leq \| x \| \| y \| .$$

   Además se da la igualdad si y solo si $y = \lambda x$.
4. Inversa generalizada A^- tal que $AA^-A = A$
5. Determinante $|A|$, si $A \in M_{n\times n}$
6. Matriz no singular si $|A| \neq 0$. Si A es no singular, existe A^{-1}
7. $I_n = \begin{pmatrix} 1 & 0 & ... & 0 \\ 0 & 1 & ... & 0 \\ ... & ... & ... & ... \\ 0 & ... & 0 & 1 \end{pmatrix}, 1_n = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$
8. Matriz diagonal $diag(d_1, \dots, d_n) = \begin{pmatrix} d_1 & 0 & ... & 0 \\ 0 & d_2 & ... & 0 \\ ... & ... & ... & ... \\ 0 & ... & 0 & d_n \end{pmatrix}$
9. Una matriz cuadrada es semidefinida positiva ($A \geq 0$) si $x'Ax \geq 0$, para todo x.
10. Todos los valores propios de una matriz simétrica (real) son números reales.
11. Una matriz cuadrada es semidefinida positiva si todos sus valores propios son mayores o iguales que cero.
12. Teorema espectral: Si A es una matriz (real) simétrica, entonces existe una matriz ortogonal T tal que $T'AT$ es diagonal.
13. $A \leq B$ si $B - A \geq 0$ (si $B - A$ es semidefinida positiva).
14. Rango de una matriz $R(A)$ es la dimensión del espacio vectorial generado por sus vectores columnas.

15. Núcleo de una matriz $N(A)$ es el espacio vectorial $\{x : Ax = 0\}$
16. Matriz idempotente $A^2 = A$
17. Traza $traza(ABC) = traza(BCA) = \sum valores\ propios$
18. A simétrica e idempotente, entonces sus valores propios son cero o uno y $rango = traza$.
19. Producto de Kroneker (directo), $A \otimes B = (a_{ij}B) \in M_{nm\times nm}, A \in M_{m\times m}$ y $B \in M_{n\times n}$.

Esperanza y covarianza:

X, Y, Z vectores (columna) aleatorios. A y B matrices reales.

1. $E(a_1g_1(X) + a_2g_2(X)) = a_1E(g_1(X)) + a_2E(g_2(X))$; $a_1, a_2 \in \mathbb{R}$
2. $X = (Y, Z), E_X(g(Y)) = E_Y(g(Y))$
3. Si (X, Y) independientes, entonces $E(g_1(X)g_2(Y)) = E(g_1(X))E(g_2(Y))$
4. $E(AX + b) = AE(X) + b; A \in M_{m,k}, b' \in \mathbb{R}^k$
5. $Cov(X_i, X_j) = E(X_iX_j) - E(X_i)E(X_j)$
6. $Cov(X_i, X_j) = 0$ si $X_1, \ldots, X_k$ son independientes
7. $Var(X_i + X_j) = Var(X_i) + 2Cov(X_i, X_j) + Var(X_j)$
8. $Cov(aX_i + b, cX_j + d) = acCov(X_i, X_j)$
9. $Cov(X) = E((X - \mu)(X - \mu)') = E(XX') - \mu\mu'$
10. $Var(a'X) = a'Cov(X)a = \sum a_ia_j\sigma_{i,j}$
11. $Cov(AX + b) = ACov(X)A'$
12. $Corr(X_i, X_j) = 0$ si $X_1, \ldots, X_k$ son independientes
13. $Corr(aX_i + b, cX_j + d) = Corr(X_i, X_j)$
14. $-1 \leq Corr(X_i, X_j) \leq 1$
15. $Corr(X_i, aX_i + b) = \pm 1$ (según el signo de a)
16. $Corr(X) = \Delta^{-1}Cov(X)\Delta^{-1}$, donde Δ es la matriz diagonal formada por las desviaciones típicas ($\Delta = diag(\sigma_1, \ldots, \sigma_k)$).
17. $Cov(X, Y) = (Cov(X_i, Y_j)) = Cov(Y, X)'$

18. $Cov(X+Z,Y) = Cov(X,Y) + Cov(Z,Y)$
19. Si X e Y tienen la misma dimensión, entonces

$$Cov(X+Y) = Cov(X) + Cov(X,Y) + Cov(Y,X) + Cov(Y)$$

20. $Cov(AX,BY) = ACov(X,Y)B'$
21. Si X,Y independientes, entonces $Cov(X,Y) = 0$

Modelos:

1. Distribución multinomial $M_k(n,p_1,\ldots,p_k)$

$$p(x_1,\ldots,x_k) = \frac{n!}{x_1!\ldots,x_k!} p_1^{x_1}\ldots p_k^{x_k}$$

2. Normal $N_k(\mu,V)$

$$f(x) = \frac{1}{\sqrt{\mid V \mid (2\pi)^k}} \exp\left(-\frac{1}{2}(x-\mu)'V^{-1}(x-\mu)\right).$$

3. Distribución Wishart $W_k(m,V) = \sum_{j=1}^m Y_j Y_j'$, con $Y_j \equiv N_k(0,V)$ indep.

$$f(w_{11},\ldots,w_{kk}) = 2^{-mk/2} \mid V \mid^{-m/2} \Gamma_k^{-1}(\frac{m}{2}) \mid W \mid^{(m-k-1)/2} e^{tr(-V^{-1}W/2)}$$

siendo $W = (w_{ij})$, $\Gamma_k(\frac{m}{2}) = \pi^{k(k-1)/4}\prod_{j=1}^k \Gamma(\frac{m+1-j}{2})$ y $\Gamma(a) = \int_0^\infty y^{a-1}e^{-y}dy$.

4. Distribución T^2 de Hotelling $T^2_{k,m} = mZ'W^{-1}Z$, con $Z \equiv N_k(0,I)$ y $W \equiv W_k(m,I)$ independientes. Verifica $\frac{m+1-k}{mk}T^2_{k,m} \equiv F_{k,m-k+1}$.
5. Distribución F de Snedecord $F_{n,m}$:

$$f(x) = \frac{\sqrt{n^n m^m}}{\beta(n/2,m/2)}\sqrt{\frac{x^{m-2}}{(m+nx)^{n+m}}} \text{ si } x > 0$$

Inferencia:

1. Media muestral: $\overline{X} = \overline{O} = (\overline{X}_j) = \frac{1}{n}\sum_{i=1}^n O_i$, con $\overline{X}_j = \frac{1}{n}\sum_{i=1}^n X_{ij}$. Propiedades $E(\overline{X}) = E(X)$, $Cov(\overline{X}) = \frac{1}{n}Cov(X)$ y $\overline{X} \equiv N_k(\mu, V/n)$ si $X \equiv N_k(\mu,V)$

2. Cuasivarianza muestral:

$$S = (S_{ij}) = \frac{1}{n-1}\sum_{l=1}^{n}(O_l - \overline{O})(O_l - \overline{O})',$$

con

$$S_{ij} = \frac{1}{n-1}\sum_{l=1}^{n}(X_{li} - \overline{X}_i)(X_{lj} - \overline{X}_j)',$$

$E(S) = V$ y $(n-1)S \equiv W_k(n-1, V)$.

3. Correlación: $R = (r_{ij}) = D^{-1}SD^{-1}, D = diag(S_j)$,con $r_{ij} = S_{i,j}/(S_iS_j)$.

4. Distancia de Mahalanobis:

$$\triangle(x, y) = \sqrt{(x-y)'V^{-1}(x-y)}$$

con $\triangle(X, \mu) \equiv \chi^2_n$ para $X \equiv N_k(\mu, V)$.

5. Distancia de Mahalanobis muestral:

$$D(x, y) = \sqrt{(x-y)'S^{-1}(x-y)}$$

con

$$nD^2(\overline{X}, \mu) = n(\overline{X} - \mu)'S^{-1}(\overline{X} - \mu) \equiv T^2_{k,n-1}$$

de Hotelling para $X \equiv N_k(\mu, V)$.

Análisis de Componentes Principales (PCA).

PCA1) Definición Y_1:

$$\left.\begin{array}{l}\max Var(a'X) \\ s.a.: \quad a'a = 1\end{array}\right\}$$

PCA2) Definición Y_j:

$$\left.\begin{array}{ll}\max Var(a'X) & \\ s.a.: & a'a = 1 \\ & Cov(Y_i, a'X) = 0, i = 1, \ldots, j-1.\end{array}\right\}$$

PCA3) Cálculo teórico: $Y = T'X$, donde $TT' = T'T = I$ y $T'VT = D = diag(\lambda_1, \ldots, \lambda_k)$ con $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k > 0$.

PCA4) $Y_j = t'_jX$, $Var(Y_j) = \lambda_j$ y

$$traza(V) = \sum_{j=1}^{k} Var(X_j) = \sum_{j=1}^{k} Var(Y_j) = \sum_{j=1}^{k} \lambda_j.$$

PCA5) Información en Y_j: $I_j = 100\lambda_j/(\sum_{i=1}^{k}\lambda_i)\,\%$.
PCA6) $|V| = \lambda_1 \ldots \lambda_k = |Cov(Y)|$.
PCA7) Relaciones entre X e $Y = T'X$:

$$\begin{aligned} Cov(X,Y) &= TD \\ Corr(X,Y) &= diag(V)^{-1/2}TD^{1/2} \end{aligned} \tag{7.1}$$

donde $diag(V) = diag(\sigma_1^2, \ldots, \sigma_k^2)$.
PCA8) Matriz de saturaciones $A = Corr(X,Y)$ con

$$a_{i,j} = Corr(X_i, Y_j) = t_{i,j}\sqrt{\lambda_j}/\sigma_i.$$

PCA9) Información en Y_j sobre X_i:

$$100Corr^2(X_i, Y_j)\,\% = 100t_{i,j}^2\lambda_j/\sigma_i^2\,\%.$$

Análisis Discriminante (DA)

DA1) Función discriminante de Fisher a la v.a.

$$D = L(Z) = a'Z = (\mu_X - \mu_Y)'V^{-1}Z.$$

DA2) Si X e Y son normales con $Cov(X) = Cov(Y) = V$,

$$D \sim N_1((\mu_X - \mu_Y)'V^{-1}\mu, \triangle(\mu_X, \mu_Y))$$

donde $\mu = E(Z)$ es igual a μ_X o μ_Y.
DA3) Regla de discriminación:

Si $L(Z) > K$, entonces Z es clasificado en X

Si $L(Z) < K$, entonces Z es clasificado en Y

donde $K = L((\mu_X + \mu_Y)/2)$.
DA4) Probabilidad de errores 1 y 2

$$\begin{aligned} \Pr(e_1) &= \Pr(Z \in R_Y \mid Z \equiv X) \\ &= \Pr\left(U < -\frac{1}{2}\triangle(\mu_X, \mu_Y)\right) \\ &= \Pr(e_2) \\ &= \Pr(Z \in R_X \mid Z \equiv Y), \end{aligned}$$

donde $U \equiv N_1(0,1)$.

DA5) Probabilidad error total

$$\begin{aligned}\Pr(error) &= \Pr(Z \in R_Y \mid Z \equiv X)\Pr(Z \equiv X) + \Pr(Z \in R_X \mid Z \equiv Y)\Pr(Z \equiv Y) \\ &= \Pr(e_1)q_1 + \Pr(e_2)q_2\end{aligned}$$

con $q_1 = \Pr(Z \equiv X)$ y $q_2 = \Pr(Z \equiv Y)$ (probabilidades a priori).

DA6) Se minimiza el coste esperado

$$c(K) = c_1 \Pr(e_1)q_1 + c_2 \Pr(e_2)q_2$$

si

$$K = a' \frac{\mu_X + \mu_Y}{2} + \log\left(\frac{c_2 q_2}{c_1 q_1}\right).$$

DA7) Probabilidades a posteriori

$$\Pr(Z \equiv X \mid Z = z) = \frac{f_X(z)q_1}{f_X(z)q_1 + f_Y(z)q_2}.$$

DA8) Función discriminante lineal (FDL)

$$L_i(z) = (\mu^{(i)})' V^{-1} z - (\mu^{(i)})' V^{-1} \mu^{(i)}/2,$$

clasificándose z en $G_i : L_i(z) \geq L_j(z)$ para todo j.

DA9) Proyecciones canónicas $Z^* = V^{-1/2}Z$ con $Cov(V^{-1/2}Z) = I$,

$$L(Z) = (V^{-1/2}\mu_X - V^{-1/2}\mu_Y)' V^{-1/2} Z$$

y

$$d_i^2(z) = d^2(V^{-1/2}z, V^{-1/2}\mu^{(i)}) = \Delta^2(z, \mu^{(i)}).$$

DA10) Función discriminante cuadrática (QDF)

$$QDF_i(z) = c - 2\log f_i(z) = (z - \mu^{(i)})' V_i^{-1} (z - \mu^{(i)}) + \log \mid V_i \mid,$$

clasificándose z en $G_i : QDF_i(z) \leq QDF_j(z)$ para todo j.

DA11) Función discriminante cuadrática basada en la distancia de Mahalanobis

$$QDF_i^*(z) = \Delta^2(z, \mu^{(i)}) = (z - \mu^{(i)})' V_i^{-1} (z - \mu^{(i)}).$$

DA12) Estimaciones por grupos

$$\begin{aligned}\widehat{\mu}_j &= \frac{1}{n_j} \sum_{i=1}^{n} \omega_j 1(Y(\omega_i) = j) \\ \widehat{V}_j &= \frac{1}{n_j - 1} \sum_{i=1}^{n} 1(Y(\omega_i) = j)(\omega_i - \widehat{\mu}_j)(\omega_i - \widehat{\mu}_j)' \\ \omega_i &= (X_{1,i}, \ldots, X_{k,i})' \\ n_j &= \sum_{i=1}^{n} 1(Y(\omega_i) = j),\end{aligned}$$

donde $1(Y(\omega_i) = j) = 1$ si $\omega_i \in G_j$ (cero si no).

DA13) Matriz de covarianzas ponderada (pooled)

$$\widehat{V} = \frac{1}{n-m}\sum_{j=1}^{m}(n_j - 1)\widehat{V}_j.$$

DA14) Función discriminante de Fisher muestral

$$\widehat{D} = \widehat{L}(Z) = \widehat{a}'Z = (\widehat{\mu}_X - \widehat{\mu}_Y)'\widehat{V}^{-1}Z.$$

DA15) Probabilidad del error tipo 1 muestral

$$\Pr(e_1) = \Pr\left(U < -\frac{1}{2}\widehat{\triangle}\right)$$

donde $U \equiv N_1(0,1)$ y

$$\widehat{\triangle} = \sqrt{(\widehat{\mu}_X - \widehat{\mu}_Y)'\widehat{V}^{-1}(\widehat{\mu}_X - \widehat{\mu}_Y)}$$

es la distancia de Mahalanobis muestral.

DA16) Funciones discriminantes lineales muestrales

$$\widehat{L}_i(z) = (\widehat{\mu}^{(i)})'\widehat{V}^{-1}z - (\widehat{\mu}^{(i)})'\widehat{V}^{-1}\widehat{\mu}^{(i)}/2,$$

clasificándose z en $G_i : \widehat{L}_i(z) \geq \widehat{L}_j(z)$ para todo j.

DA17) Proyecciones canónicas muestrales $Z^* = \widehat{V}^{-1/2}Z$.

DA18) Funciones discriminantes cuadráticas muestrales

$$\widehat{Q}_i(z) = c - 2\log\widehat{f}_i(z) = (z - \widehat{\mu}^{(i)})'\widehat{V}_i^{-1}(z - \widehat{\mu}^{(i)}) + \log \mid \widehat{V}_i \mid,$$

clasificándose z en $G_i : \widehat{Q}_i(z) \leq \widehat{Q}_j(z)$ para todo j.

7.2. Tablas

Tabla 7.1: Características de los modelos discretos más usuales.

Modelo	$E(X)$	$Var(X)$	γ_1	γ_2
Binomial	np	npq	$\frac{1-2p}{\sqrt{npq}}$	$\frac{1}{npq} - \frac{6}{n}$
Geométrica	q/p	q/p^2	$\frac{1+q}{\sqrt{q}}$	$6 + \frac{p^2}{q}$
Bin. Neg.	nq/p	nq/p^2	$\frac{1+q}{\sqrt{nq}}$	$\frac{6q+p^2}{nq}$
Hipergeo.	$n\frac{a}{N}$	$n\frac{a}{N}\frac{N-a}{N}\frac{N-n}{N-1}$	$\frac{(N-2a)(N-2n)}{N-2}\sqrt{\frac{(N-1)}{Na(N-a)(N-n)}}$	$*$
Poisson	λ	λ	$1/\sqrt{\lambda}$	$1/\lambda$

$H(N, a, n) \approx B(n, p = a/N)$ si $N/n > 10$.
$B(n, p) \approx P(\lambda = np)$ si $np > 1$ y $p < 0.1$.
$B(n, p) \approx N(\mu = np, \sigma = \sqrt{npq})$ si $npq > 5$.
$P(\lambda) \approx N(\mu = \lambda, \sigma = \sqrt{\lambda})$ si $\lambda > 5$.

Tabla 7.2: Características de los modelos continuos más usuales.

Modelo	$E(X)$	$Var(X)$	γ_1	γ_2
Normal	μ	σ^2	0	0
Uniforme	$(a+b)/2$	$(b-a)^2/12$	0	$-6/5$
Exponencial	$1/\lambda$	$2/\lambda^2$	2	6
Gamma	a/b	a/b^2	$2/\sqrt{a}$	$6/a$
Beta	$\frac{a}{a+b}$	$\frac{ab}{(a+b+1)(a+b)^2}$	$*$	$*$
Weibull	$b\Gamma(1+1/a)$	$b^2\Gamma(1+2/a) - b^2\Gamma^2(1+1/a)$	$*$	$*$
Chi-cuadrado	n	$2n$	$\sqrt{8/n}$	$12/n$
t-Student	0	$\frac{n}{n-2}$	0	$\frac{6}{n-4}$
F-Snedecor	$\frac{n}{n-2}$	$\frac{2n^2(m+n-2)}{m(n-2)^2(n-4)}$	$*$	$*$

$\Gamma(p) = \int_0^\infty t^{p-1} \exp(-t)\, dt$.
$\beta(a, b) = \int_0^1 x^{a-1}(1-x)^{b-1} dx = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$.

Tabla 7.3: Función de distribución Normal $N(0,1)$.

z	G(z)	z	G(z)	z	G(z)
0.1	0.5398278	1.1	0.8643339	2.1	0.9821356
0.2	0.5792597	1.2	0.8849303	2.2	0.9860966
0.3	0.6179114	1.3	0.9031995	2.3	0.9892759
0.4	0.6554217	1.4	0.9192433	2.4	0.9918025
0.5	0.6914625	1.5	0.9331928	2.5	0.9937903
0.6	0.7257469	1.6	0.9452007	2.6	0.9953388
0.7	0.7580363	1.7	0.9554345	2.7	0.9965330
0.8	0.7881446	1.8	0.9640697	2.8	0.9974449
0.9	0.8159399	1.9	0.9712834	2.9	0.9981342
1	0.8413447	2	0.9772499	3	0.9986501

Tabla 7.4: Cuantiles de la distribución Normal $N(0,1)$.

z	G(z)	z	G(z)	z	G(z)
0	0.50	0.25334710	0.60	0.52440051	0.70
0.02506891	0.51	0.27931903	0.61	0.55338472	0.71
0.05015358	0.52	0.30548079	0.62	0.58284151	0.72
0.07526986	0.53	0.33185335	0.63	0.61281299	0.73
0.10043372	0.54	0.35845879	0.64	0.64334541	0.74
0.12566135	0.55	0.38532047	0.65	0.67448975	0.75
0.15096922	0.56	0.41246313	0.66	0.70630256	0.76
0.17637416	0.57	0.43991317	0.67	0.73884685	0.77
0.20189348	0.58	0.46769880	0.68	0.77219321	0.78
0.22754498	0.59	0.49585035	0.69	0.80642125	0.79

z	G(z)	z	G(z)	z	G(z)
0.84162123	0.80	1.28155157	0.90	2.575829	0.995
0.87789630	0.81	1.34075503	0.91	3.090232	0.999
0.91536509	0.82	1.40507156	0.92	3.290527	0.9995
0.95416525	0.83	1.47579103	0.93	3.719016	0.9999
0.99445788	0.84	1.55477359	0.94		
1.03643339	0.85	1.64485363	0.95		
1.08031934	0.86	1.75068607	0.96		
1.12639113	0.87	1.88079361	0.97		
1.17498679	0.88	2.05374891	0.98		
1.22652812	0.89	2.32634787	0.99		

Tabla 7.5: Comandos en R más usuales

Comando	Significado
`m^n`	m^n
`factorial(m)`	$m!$
`choose(m,n)`	$\binom{m}{n}$
`exp(x)`	e^x
`log(x)`	$\ln(x)$
`curve(f(x),a,b,ylab='f(x)')`	Dibuja f en (a,b)
`curve(g(x),add=TRUE)`	Añade la gráfica de g
`plot(x,y)`	Dibuja los puntos (x,y)
`plot(x,y,type='l')`	Dibuja los (x,y) y los une
`plot(x)`	Dibuja la serie (i,x_i)
`text(x,y,z)`	Pone la etiqueta z en (x,y)
`barplot(x,f)`	Histograma de frecuencias (x,f_x)
`f<-function(x) 2x+3`	Define la función $f(x)=2x+3$
`gamma(p)`	$\Gamma(p)=\int_0^\infty t^{p-1}\exp(-t)\,dt$
`beta(a,b)`	$\beta(a,b)=\int_0^1 x^{a-1}(1-x)^{b-1}$
`mean(X)`	$\overline{X}=\frac{1}{n}=\sum_{i=1}^n X_i$
`var(X)`	$S^2=\frac{1}{n-1}\sum_{i=1}^n(X_i-\overline{X})^2$
`v<-vector(length=3)`	Define el vector (columna) v
`M<-matrix(nrow=2,ncol=2)`	Define la matriz M
`M[1,2]`	Elemento de la fila 1 y columna 2 de M
`t(M)`	Matriz (o vector) transpuesto
`A%*%B`	Producto de matrices
`solve(M)`	Matriz inversa
`eigen(M)`	Vectores y valores propios de M
`mahalanobis(x,y,V)`	Distancia de Mahalanobis

Tabla 7.6: Nombres en R de los modelos discretos más usuales.

Modelo	$p(x)$	x	Nombre en R
Binomial	$\binom{n}{x}p^x(1-p)^{n-x}$	0,. . . ,n	$binom(x,n,p)$
Geométrica	$p(1-p)^x$	0,1,. . .	$geom(x,p)$
Bin. Neg.	$\binom{n+x-1}{x}p^n(1-p)^x$	$0,1,\ldots$	$nbinom(x,n,p)$
Hipergeo.	$\binom{a}{x}\binom{b}{n-x}/\binom{N}{n}$	$0,\ldots,\text{mín}(n,a)$	$hyper(x,a,b,n)$
Poisson	$\exp(-\lambda)\lambda^x/x!$	$0,1,\ldots$	$pois(x,\lambda)$

La función de distribución $F(x)$ se obtiene haciendo *pnombre*, la función puntual de probabilidad $p(x)$ con *dnombre*, los cuantiles $F^{-1}(x)$ con *qnombre* y podemos generar m datos con $rnombre(m,a,b)$.

Tabla 7.7: Nombres en R de los modelos continuos más usuales.

Modelo	$f(x)$	x	Nombre en R
Normal	$\frac{1}{\sigma\sqrt{2\pi}}\exp(-(x-\mu)^2/(2\sigma^2))$	$\mathbb{R}$	$norm(x,\mu,\sigma)$
Uniforme	$1/(b-a)$	(a,b)	$unif(x,a,b)$
Exponencial	$\lambda\exp(-\lambda x)$	$(0,\infty)$	$exp(x,\lambda)$
Gamma	$\frac{b^a}{\Gamma(a)}x^{a-1}e^{-bx}$	$(0,\infty)$	$gamma(x,a,b)$
Beta	$\frac{1}{\beta(a,b)}x^{a-1}(1-x)^{b-1}$	$(0,1)$	$beta(x,a,b)$
Weibull	$b^{-a}x^{a-1}\exp(-(x/b)^a)$	$(0,\infty)$	$weibull(x,a,b)$
Chi-cuadrado	$\frac{2^{-n/2}}{\Gamma(n/2)}x^{n/2-1}e^{-x/2}$	$(0,\infty)$	$chisq(x,n)$
t-Student	$\frac{1}{\beta(1/2,n/2)}\sqrt{\frac{n^n}{(n+x^2)^{n+1}}}$	$\mathbb{R}$	$t(x,n)$
F-Snedecor	$\frac{n^{n/2}m^{m/2}}{\beta(n/2,m/2)}\frac{x^{m/2-1}}{(n+mx)^{(n+m)/2}}$	$(0,\infty)$	$f(x,m,n)$
$N_k(\mu,V)$	$\frac{1}{\sqrt{\lvert V\rvert(2\pi)^k}}\exp(-\Delta^2(x,\mu)/2)$	$\mathbb{R}^k$	$mvnorm(x,\mu,V)$

La función de distribución $F(x)$ se obtiene haciendo *pnombre*, la función de densidad de probabilidad $f(x)$ con *dnombre*, los cuantiles $F^{-1}(x)$ con *qnombre* y podemos generar m datos con $rnombre(m,a,b)$. Para la normal multivariante hay que cargar el paquete **mvtnorm**.

Tabla 7.8: Comandos en R para Análisis de Componentes Principales.

Comando	Significado
`data()`	Datos en R
`d<-LifeCycleSavings`	Guardar datos
`summary(d)`	Resumen de los datos
`View(d)`	Ver los datos
`plot(d)`	Gráficos bidimensionales
`cov(d)`	Matriz de covarianzas
`cor(d)`	Matriz de correlaciones
`boxplot(d)`	Diagramas caja-bigote
`boxplot(d[,1])`	Diagrama caja-bigote
`which.max(d[,1])`	Índice que da el máximo
`sort(d[,1])`	Valores ordenados
`hist(d[,1])`	Histograma
`PCA<-princomp(d)`	PCA de covarianzas
`PCA<-princomp(d,cor=TRUE)`	PCA de correlaciones
`princomp(covmat=cor(d))`	PCA basado en una matriz
`PCAbis<-prcomp(d,scale=TRUE)`	PCA de correlaciones
`summary(PCA,loadings=TRUE)`	Resumen PCA
`PCA$loadings->T`	Matriz de cargas
`PCA$scores->S`	Matriz de puntuaciones
`z<-scale(d)`	Variables estandarizadas
`biplot(PCA,pc.biplot=TRUE)`	Gráfico $Y_1 - Y_2$
`biplot(PCA,pc.biplot=TRUE,choices=c(3,4))`	Gráfico $Y_3 - Y_4$
`biplot(PCA,pc.biplot=TRUE,xlabs=1:50)`	Gráfico con etiquetas
`plot(S[,1],S[,2],xlab='Y1',ylab='Y2')`	Gráfico $Y_1 - Y_2$
`text(S[38,1],S[38,2],labels='Esp')`	Etiqueta del dato 38
`pairs(PCA$scores[,1:3])`	Gráficos $Y_1 - Y_2 - Y_3$
`SAT<-cor(d,S)`	Matriz de saturaciones
`SAT[,1]^2`	Informaciones en Y_1
`SAT[,1]^2+ SAT[,2]^2`	Comunalidades en $Y_1 - Y_2$
`screeplot(PCA)`	Gráfico de sedimentación
`plot(eigen(cor(d))$values,type='l')`	Gráfico de sedimentación

Tabla 7.9: Comandos en R para Análisis Discriminante

Comando	Significado
`load('e:/tal/escarabajos.rda')`	Leer escarabajos.rda
`dump('d','g:/nombre/datos1.R')`	Guardar el objeto d
`source('g:/nombre/datos1.R'`	Leer el objeto d
`tapply(d$surco,d$especie,summary)`	Resumen por grupos
`plot(d$surco,d$codigo)`	Gráfica por grupos
`text(d$surco[40],1.5,labels='40')`	Etiqueta para el dato 40
`boxplot(d$surco~d$especie)`	Gráficos caja-bigote por grupos
`plot(d$surco,d$long,pch=as.integer(d$especie))`	Gráficos $x-y$ por grupos
`legend('topright',legend=c('40','2','1'),pch=1:3)`	Explicación símbolos
`plot(d$surco,d$long)`	Gráfico bidimensional
`text(d$surco,d$long,d$especie)`	Etiquetas
`pca<-princomp(d[,1:4],cor=TRUE)`	Cálculo PCA
`biplot(pca,pc.biplot=TRUE,xlabs=d$especie)`	Gráfico PCA por grupos
`library('MASS')`	Carga paquete MASS
`LDA<-lda(d[1:39,1:4],d[1:39,6],prior=c(1/2,1/2))`	Cálculo LDA
`a<-LDA$scaling`	Vector a
`L<-function(z) sum(a*z)`	Función de Fisher
`z<-d[,1:4]`	Función discriminante
`D<-1:40`	Función discriminante
`for (i in 1:40) D[i]<-L(z[i,])`	Función discriminante FD
`plot(D,d$codigo)`	Gráfico FD
`text(D,d$codigo,pos=3,col='red')`	Etiquetas D
`text(D[40],labels='*')`	Etiqueta dato 40
`text(D[40],labels='40')`	Etiqueta dato 40
`P<-predict(LDA,d[,1:4])`	Predicciones con LDA
`P$x`	Puntuaciones
`ldahist(P$x,g=d$especie)`	Histograma por grupos
`P$class`	Predicciones
`P$class==d[,6]`	Aciertos y fallos
`table(P$class,d[,6])`	Resumen aciertos
`P$posterior`	Probabilidades a posteriori
`predict(LDA,c(185,280,150,200))`	Predicción
`CV<-lda(...,CV=TRUE)`	Validación cruzada
`table(CV$class,d[1:39,6])`	Aciertos CV
`QDA<-qda(d[1:39,1:4],d[1:39,6],prior=c(0.5,0.5))`	QDA
`predict(QDA,d[,1:4])->P`	Predicciones QDA
`table(P$class,d$codigo)`	Aciertos QDA
`qda(...,CV=TRUE)`	Validación cruzada
`table(QDACV$class,d[1:39,6])`	Aciertos CV
`S1<-cov(d[1:19,1:4])`	Matriz covarianza MC1
`S2<-cov(d[20:39,1:4])`	Matriz covarianza MC2
`(18*S1+19*S2)/37->S`	MC ponderada

8

Índice alfabético

Índice alfabético

9

Bibliografía

Anderson, T.W. (1974). A Introduction to Multivariate Statistical Analysis. Wiley.

Burgos, J. (1994). Curso de Álgebra y Geometría. Alhambra Longman.

Cuadras, C.M. (1991). Métodos de análisis multivariante. PPU.

Guillamón, A.; Navarro, J. (2002). Probabilidad y Estadística. Fundamentos (2ª ed.). DM.

Hartigan, J. A.; Wong, M. A. (1979). Algorithm AS 136: A K-means clustering algorithm. Applied Statistics, 28, 100–108. 10.2307/2346830.

Hastie, T. Tibshirani, R. (2009). The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Springer Series in Statistics. Springer.

Kotz, S.; Balakrishnan, N.; Jonhson, N.L. (2000). Continuous multivariate distributions.

Koenker, R. (2005). Quantile Regression. Cambridge University Press.

Koenker, R.; Bassett Jr., G. (1978). Regression quantiles. Econometrica 46(1), 33–50.

Mardia, K.V.; Kent, J.T; Bibby, J.M. (1997). Multivariate Analysis. Academic Press.

Navarro, J.; Franco, M.; Guillamón, A. (1999). Probabilidad y Estadística. Problemas. DM.

Peña, D. (2002). Análisis de datos multivariantes. McGraw Hill.

Rencher, A.C. (1995). Methods of Multivariate Analysis. Wiley.

Srivastava, M.S.; Carter, E.M. (1983). A Introduction to Applied Multivariate Statistics. North-Holland.

Zoroa, P.; Zoroa, N. (2008). Elementos de Probabilidades. Diego Marín.