

Un análisis con R

Datos multivariantes

Francesc Carmona
Departament d'Estadística

9 de febrero de 2015

1. Los datos

En este artículo vamos a utilizar un conjunto de datos del paquete MASS según las indicaciones del trabajo de Carey[1]. Se trata de los datos del artículo

Campbell, N.A. and Mahon, R.J. (1974) A multivariate study of variation in two species of rock crab of genus *Leptograpsus*. *Australian Journal of Zoology* **22**, 417-425.

en los que aparece información relativa a dos especies de cangrejos. Para acceder a los datos en R escribimos

```
> library(MASS)
> data(crabs)
> help(crabs)
```

La última instrucción nos proporcionará una somera descripción de su contenido. También podemos ver los primeros registros con

```
> crabs[1:3, ]

  sp sex index  FL  RW  CL  CW  BD
1  B  M     1  8.1 6.7 16.1 19.0 7.0
2  B  M     2  8.8 7.7 18.1 20.8 7.4
3  B  M     3  9.2 7.8 19.0 22.4 7.7
```

o la estructura de los datos con

```
> str(crabs)

'data.frame': 200 obs. of 8 variables:
 $ sp : Factor w/ 2 levels "B","O": 1 1 1 1 1 1 1 1 1 1 ...
 $ sex : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
 $ index: int 1 2 3 4 5 6 7 8 9 10 ...
 $ FL : num 8.1 8.8 9.2 9.6 9.8 10.8 11.1 11.6 11.8 11.8 ...
 $ RW : num 6.7 7.7 7.8 7.9 8 9 9.9 9.1 9.6 10.5 ...
 $ CL : num 16.1 18.1 19 20.1 20.3 23 23.8 24.5 24.2 25.2 ...
 $ CW : num 19 20.8 22.4 23.1 23 26.5 27.1 28.4 27.8 29.3 ...
 $ BD : num 7 7.4 7.7 8.2 8.2 9.8 9.8 10.4 9.7 10.3 ...
```

Las variables son

- `sp`, el código de la especie, O para naranja (orange) y B para azul (blue)
- `sex`, el código del sexo

- cinco medidas anatómicas en milímetros: FL (frontal lobe width - anchura frontal del lóbulo), RW (rear width - anchura trasera), CL (carapace length - longitud del caparazón), CW (carapace width - anchura del caparazón), BD (body depth - profundidad del cuerpo).

El objetivo es describir estas variables, estudiar la relación entre las características anatómicas y comprender cómo estas medidas se asocian con la especie y el sexo.

2. Carpeta de trabajo

Se trata de crear una carpeta de trabajo con el nombre “cangrejos” para dejar los archivos que utilizaremos en diversas sesiones. Esto depende del sistema operativo que utilizemos. En el caso de *Windows* crearemos una carpeta del tipo C:/cangrejos y la fijaremos en R como carpeta de trabajo:

```
> setwd("C:/cangrejos")
```

Esto también se puede hacer directamente desde el menú File → Change dir... de Rgui o en la pestaña Files → More → Set As Working Directory de RStudio.

3. Estadística descriptiva

En primer lugar queremos tener algún resumen de los datos que se puede obtener con la instrucción `summary`:

```
> attach(crabs)
> summary(crabs)
```

sp	sex	index	FL	RW	CL
B:100	F:100	Min. : 1.0	Min. : 7.20	Min. : 6.50	Min. :14.70
O:100	M:100	1st Qu.:13.0	1st Qu.:12.90	1st Qu.:11.00	1st Qu.:27.27
		Median :25.5	Median :15.55	Median :12.80	Median :32.10
		Mean :25.5	Mean :15.58	Mean :12.74	Mean :32.11
		3rd Qu.:38.0	3rd Qu.:18.05	3rd Qu.:14.30	3rd Qu.:37.23
		Max. :50.0	Max. :23.10	Max. :20.20	Max. :47.60
		CW	BD		
		Min. :17.10	Min. : 6.10		
		1st Qu.:31.50	1st Qu.:11.40		
		Median :36.80	Median :13.90		
		Mean :36.41	Mean :14.03		
		3rd Qu.:42.00	3rd Qu.:16.60		
		Max. :54.60	Max. :21.60		

También podemos cruzar datos cualitativos:

```
> table(sp, sex)
```

	sex	
sp	F	M
B	50	50
O	50	50

Con las variables cuantitativas es interesante calcular las medidas de localización y dispersión más habituales. Para ello podemos utilizar la instrucción `sapply` con las diversas funciones estadísticas como `mean`, `median`, `var`, `sd`, `IQR` o `range`

```
> crabs5 <- crabs[, 4:8]
> sapply(crabs5, mean)
```

FL	RW	CL	CW	BD
15.5830	12.7385	32.1055	36.4145	14.0305

```

> sapply(crabs5, median)

      FL      RW      CL      CW      BD
15.55 12.80 32.10 36.80 13.90

> sapply(crabs5, var)

      FL      RW      CL      CW      BD
12.217297  6.622078 50.679919 61.967678 11.729065

> sapply(crabs5, sd)

      FL      RW      CL      CW      BD
3.495325 2.573340 7.118983 7.871955 3.424772

> sapply(crabs5, IQR)

      FL      RW      CL      CW      BD
5.15   3.30   9.95 10.50   5.20

> sapply(crabs5, range)

      FL      RW      CL      CW      BD
[1,]  7.2   6.5 14.7 17.1   6.1
[2,] 23.1 20.2 47.6 54.6 21.6

```

Aunque la función `mean` se puede aplicar directamente a las columnas de un `data.frame`, es mejor utilizar la función `colMeans` o alguna de las funciones `sapply`. Sin embargo, algunas instrucciones sólo se pueden utilizar directamente sobre un único vector numérico. Por ejemplo, una instrucción que falla es

```

> median(crabs5)

```

Un resumen con los cinco números de Tukey se obtiene así:

```

> sapply(crabs5, fivenum)

      FL      RW      CL      CW      BD
[1,]  7.20   6.5 14.70 17.1   6.1
[2,] 12.90 11.0 27.25 31.5 11.4
[3,] 15.55 12.8 32.10 36.8 13.9
[4,] 18.10 14.3 37.25 42.0 16.6
[5,] 23.10 20.2 47.60 54.6 21.6

```

También podemos calcular algunos estadísticos sobre las poblaciones estudiadas por separado como:

```

> tapply(FL, sp, fivenum)

$B
[1]  7.20 11.80 14.45 16.15 21.30

$O
[1]  9.10 14.45 17.50 19.55 23.10

> tapply(FL, sex, fivenum)

$F
[1]  7.20 12.80 15.45 18.15 23.10

$M
[1]  8.10 13.15 15.70 18.10 23.10

```

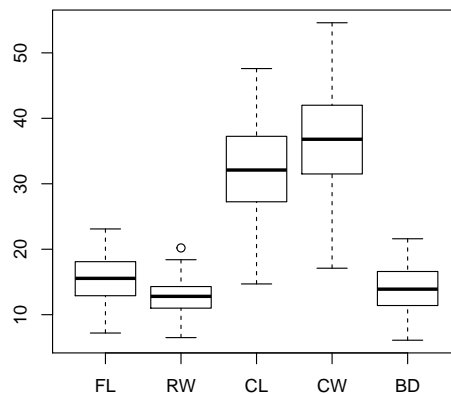


Figura 1: Gráficos de caja de las medidas numéricas para los datos de cangrejos

o incluso

```
> sapply(crabs5, function(x) tapply(x, sex, mean))
```

	FL	RW	CL	CW	BD
F	15.432	13.487	31.360	35.830	13.724
M	15.734	11.990	32.851	36.999	14.337

También podemos calcular la matriz de varianzas-covarianzas con la instrucción `var` y la matriz de correlaciones con la instrucción `cor`.

```
> var(crabs5)
```

	FL	RW	CL	CW	BD
FL	12.217297	8.158045	24.35668	26.55080	11.822581
RW	8.158045	6.622078	16.35466	18.23964	7.836659
CL	24.356677	16.354662	50.67992	55.76138	23.971389
CW	26.550801	18.239640	55.76138	61.96768	26.091867
BD	11.822581	7.836659	23.97139	26.09187	11.729065

```
> cor(crabs5)
```

	FL	RW	CL	CW	BD
FL	1.0000000	0.9069876	0.9788418	0.9649558	0.9876272
RW	0.9069876	1.0000000	0.8927430	0.9004021	0.8892054
CL	0.9788418	0.8927430	1.0000000	0.9950225	0.9832038
CW	0.9649558	0.9004021	0.9950225	1.0000000	0.9678117
BD	0.9876272	0.8892054	0.9832038	0.9678117	1.0000000

Esta última muestra una alta correlación entre las variables.

De forma gráfica podemos optar por los llamados gráficos de caja (*boxplot*) con la instrucción

```
> boxplot(crabs5)
```

que proporciona la figura 1.

También podemos optar por gráficos comparativos univariantes para cada población por separado.

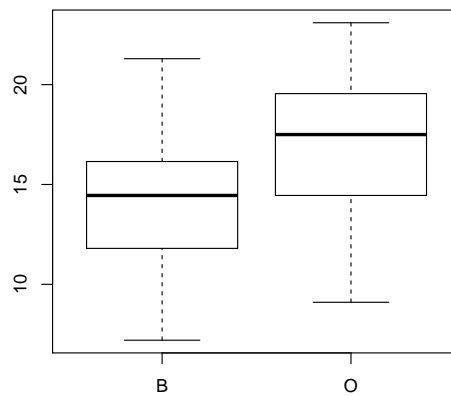


Figura 2: Gráficos de caja de la variable FL en cada especie

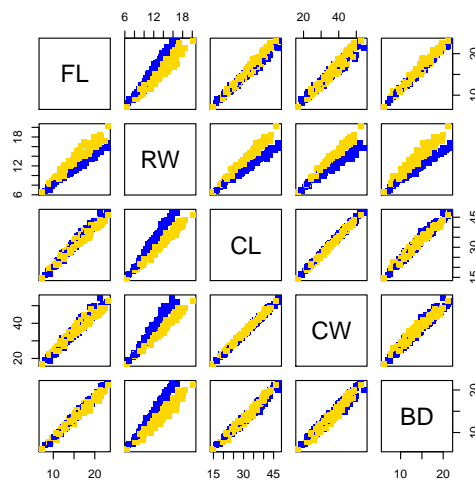


Figura 3: Gráficos de dispersión bivariantes

```
> boxplot(FL[sp == "O"], FL[sp == "B"], names = c("O", "B"))
```

Para mejorar la instrucción anterior se puede utilizar una fórmula

```
> boxplot(FL ~ sp, data = crabs)
```

que proporciona la figura 2.

La utilización de histogramas no se descarta, aunque se deben armonizar si queremos que sean realmente comparativos.

Otra opción es visualizar las correlaciones entre las variables numéricas con la instrucción `pairs`. El resultado puede verse en la figura 3.

```
> pairs(crabs5, col = ifelse(sex == "M", "blue", "gold"), pch = 15)
```

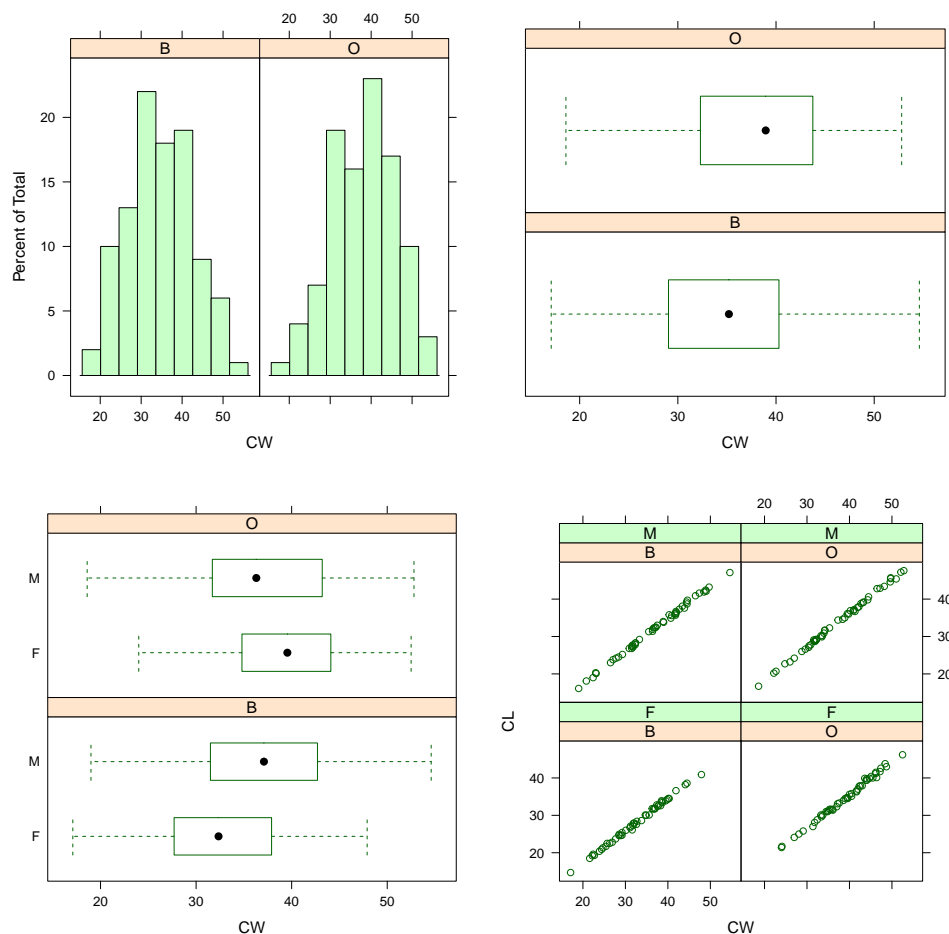


Figura 4: Gráficos obtenidos con el paquete lattice.

4. Paquetes gráficos

4.1. El paquete lattice

El paquete `lattice` es muy útil para describir gráficamente datos multivariantes. La idea consiste en que el gráfico está formado por un cierto número de paneles. Normalmente cada uno de ellos corresponde a alguno de los valores de una variable que condiciona. Es decir, un gráfico diferente para cada nivel del factor utilizado como condición. Las funciones se escriben con la notación de la fórmula del modelo. En los gráficos univariantes como los histogramas, la variable respuesta, a la izquierda, se deja vacía.

```
> library(lattice)
> trellis.device(color = TRUE, theme = "col.whitebg") # change default colors
> histogram(~CW | sp, data = crabs)
> bwplot(~CW | sp, data = crabs, layout = c(1, 2))
> bwplot(sex ~ CW | sp, data = crabs, layout = c(1, 2))
> xyplot(CL ~ CW | sp * sex, data = crabs)
```

El segundo gráfico de boxplot es muy interesante porque nos permite estudiar comparativamente el sexo separado por especies como se puede ver en la figura 4.

Los diagramas de dispersión se obtienen con la función `xyplot` en lugar de `plot`. En este caso se necesitan dos variables. Como se puede ver en la figura 4, el resultado de esta instrucción es un gráfico con cuatro paneles donde podemos estudiar la relación entre dos variables según dos factores.

En algunos casos es preciso modificar la estructura de los datos para poder realizar algunos gráficos.

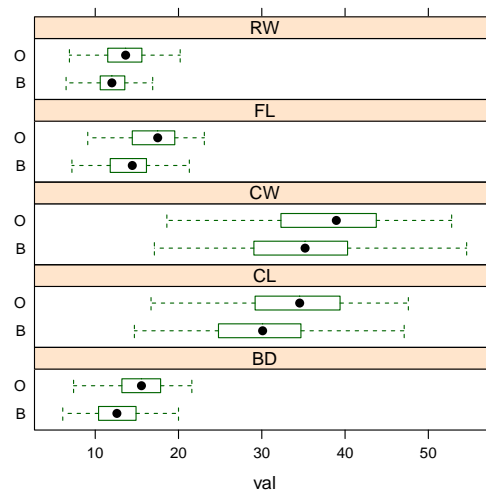


Figura 5: Gráficos de caja de cada variable, según especie

```
> vect <- as.vector(t(data.matrix(crabs5)))
> nombres <- factor(rep(names(crabs5), 200))
> especie <- rep(sp, each = 5)
> cangrejos <- data.frame(caract = nombres, val = vect, esp = especie)
> str(cangrejos)
```

```
'data.frame': 1000 obs. of 3 variables:
 $ caract: Factor w/ 5 levels "BD","CL","CW",...: 4 5 2 3 1 4 5 2 3 1 ...
 $ val : num 8.1 6.7 16.1 19 7 8.8 7.7 18.1 20.8 7.4 ...
 $ esp : Factor w/ 2 levels "B","O": 1 1 1 1 1 1 1 1 1 1 ...
```

En el siguiente apartado veremos otra forma de construir el *data.frame* cangrejos con la instrucción *melt* del paquete *reshape*.

Ahora, la siguiente instrucción dibuja el gráfico de la figura 5.

```
> bwplot(esp ~ val | caract, data = cangrejos, layout = c(1, 5))
```

4.2. El paquete ggplot2

Los mismos gráficos del apartado anterior se pueden reproducir con las instrucciones del paquete *ggplot2* y se obtiene la figura 6.

```
> library(ggplot2)
> ggplot(crabs, aes(CW)) + geom_histogram(binwidth = 5) + facet_wrap(~sp)
> ggplot(crabs, aes(sp, CW)) + geom_boxplot() + coord_flip()
> ggplot(crabs, aes(sex, CW)) +
+       geom_boxplot() + coord_flip() + facet_wrap(~sp, ncol=1)
> ggplot(crabs, aes(CW, CL)) + geom_point() + facet_wrap(~sex + sp)
```

También podemos reproducir la figura 5 con la ayuda de la función *melt* del paquete *reshape* para construir un nuevo *data.frame*. El resultado se muestra en la figura 7.

```
> library(reshape)
> mm <- melt(crabs5)
> especie <- rep(levels(sp), 5, each=100)
> cangrejos <- data.frame(caract = mm$variable, val = mm$value, esp = especie)
```

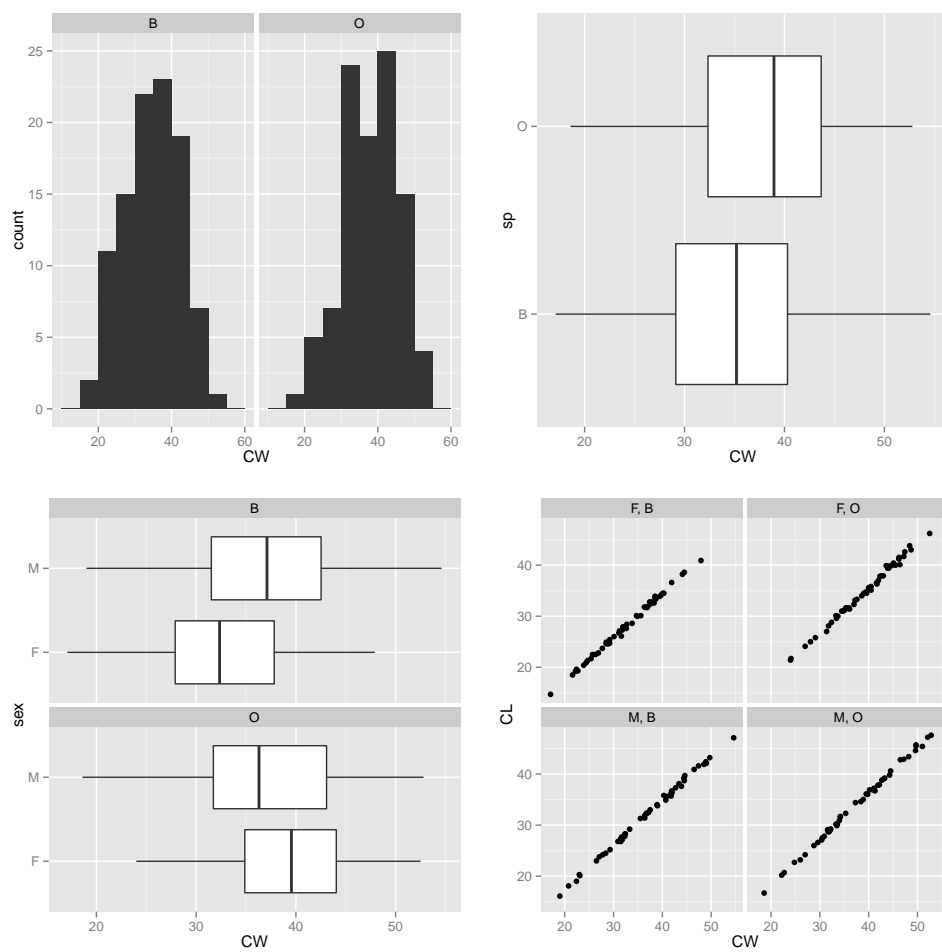


Figura 6: Gráficos obtenidos con el paquete ggplot2.

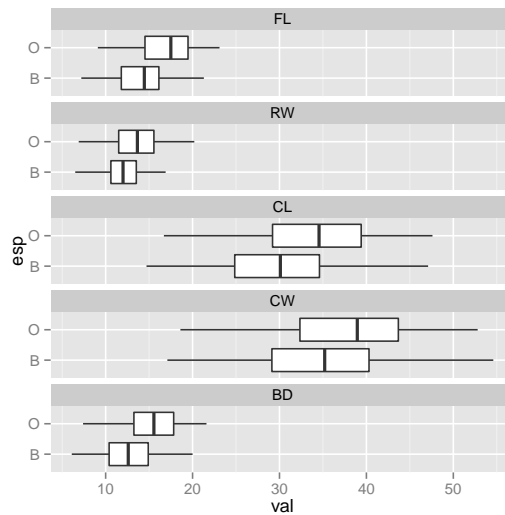


Figura 7: Gráficos de caja de cada variable, según especie

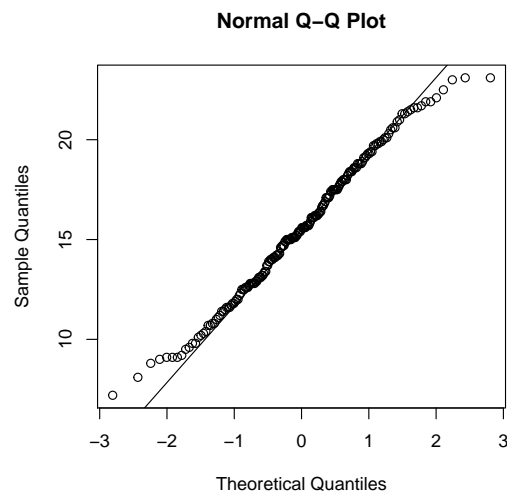


Figura 8: Gráfico para la evaluación de la normalidad de la variable FL

```
> ggplot(cangrejos, aes(esp, val)) +  
+   geom_boxplot() + coord_flip() + facet_wrap(~caract, ncol=1)
```

5. Contrastes entre dos poblaciones

En primer lugar podemos estudiar la previsible normalidad de los datos mediante el gráfico de la figura 8.

```
> qqnorm(crabs[, "FL"])  
> qqline(crabs[, "FL"])
```

Al mismo tiempo podemos realizar un test de normalidad. En la configuración básica se dispone del siguiente:

```
> shapiro.test(crabs[, "FL"])
```

Shapiro-Wilk normality test

```
data: crabs[, "FL"]  
W = 0.9904, p-value = 0.2023
```

Ahora, si queremos comparar las dos especies podemos hacer un test t de comparación de medias. Aunque previamente deberemos comparar las varianzas y proceder en consecuencia. El gráfico de la figura 2 nos indica el resultado final.

```
> var.test(FL[sp == "B"], FL[sp == "O"])  
  
F test to compare two variances  
  
data: FL[sp == "B"] and FL[sp == "O"]  
F = 0.8498, num df = 99, denom df = 99, p-value = 0.4196  
alternative hypothesis: true ratio of variances is not equal to 1  
95 percent confidence interval:  
 0.5717937 1.2630299  
sample estimates:  
ratio of variances  
 0.8498192  
  
> t.test(FL[sp == "B"], FL[sp == "O"], var.equal = TRUE)  
  
Two Sample t-test  
  
data: FL[sp == "B"] and FL[sp == "O"]  
t = -6.8551, df = 198, p-value = 8.842e-11  
alternative hypothesis: true difference in means is not equal to 0  
95 percent confidence interval:  
 -3.932543 -2.175457  
sample estimates:  
mean of x mean of y  
 14.056 17.110
```

Si dudamos de la normalidad de los datos, deberemos utilizar un test no paramétrico como `wilcox.test` o `ks.test`.

6. Análisis de la varianza

Elegimos una variable numérica como FL y vamos a comparar sus valores teniendo en cuenta los factores especie y sexo.

En primer lugar creamos el modelo lineal cuya variable respuesta es FL y a continuación calculamos la tabla ANOVA:

```
> g <- lm(FL ~ sp * sex, data = crabs)  
> gan <- anova(g)
```

```
> library(xtable)  
> xtable(gan, caption = "Tabla ANOVA", floating = F, label = "tab:anova")
```

Como se puede observar en el código anterior, la tabla 1 se produce en \LaTeX con la ayuda de la función `xtable` del paquete `xtable` de R.

7. Clasificación

Vamos a fijar nuestra atención en 40 observaciones de los datos originales.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
sp	1	466.35	466.35	48.63	0.0000
sex	1	4.56	4.56	0.48	0.4913
sp:sex	1	80.65	80.65	8.41	0.0042
Residuals	196	1879.69	9.59		

Cuadro 1: Tabla ANOVA

```
> matriz <- data.matrix(crabs5)
> ind <- c(1:10, 51:60, 101:110, 151:160)
> submatriz <- matriz[ind, ]
```

La distancia euclídea entre dos puntos \mathbf{x}_1 y \mathbf{x}_2 de un espacio de dimensión n es

$$d_E(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2}$$

El cálculo para las variables numéricas de los cangrejos es

```
> dist(t(submatriz))
```

	FL	RW	CL	CW
RW	11.665333			
CL	73.631583	84.572986		
CW	93.532134	104.387164	20.090545	
BD	9.119759	5.172040	82.454836	102.369576

Otra definición de distancia es la llamada de *Canberra*

$$d_C(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=1}^n \frac{|x_{1i} - x_{2i}|}{|x_{1i} + x_{2i}|}$$

que se puede calcular con la instrucción

```
> dist(t(submatriz), method = "canberra")
```

	FL	RW	CL	CW
RW	3.269761			
CL	13.547798	16.362142		
CW	15.790481	18.466711	2.594875	
BD	2.763221	1.326496	15.932914	18.052531

En R podemos utilizar estas distancias para clasificar las observaciones con diversos métodos. El paquete básico `stats` incluye la función `hclust` que calcula un árbol de clasificación jerárquica con varios criterios, y `kmeans` que divide el espacio en k regiones (con k a elegir por el usuario). El paquete `cluster` incluye una función `pam`, similar al procedimiento `kmeans`.

Las siguientes instrucciones comparan y dibujan cuatro tipos de clasificaciones (ver figura 9).

```
> par(mfrow = c(2, 2))
> clas1 <- hclust(dist(submatriz))
> plot(clas1, main = "hclust por defecto")
> clas2 <- hclust(dist(submatriz), method = "single")
> plot(clas2, main = "conexion simple")
> clas3 <- hclust(dist(submatriz, method = "canberra"))
> library(cluster)
> clas4 <- pam(dist(submatriz), 4)
> clusplot(clas4)
> plot(silhouette(clas4))
> par(mfrow = c(1, 1))
```

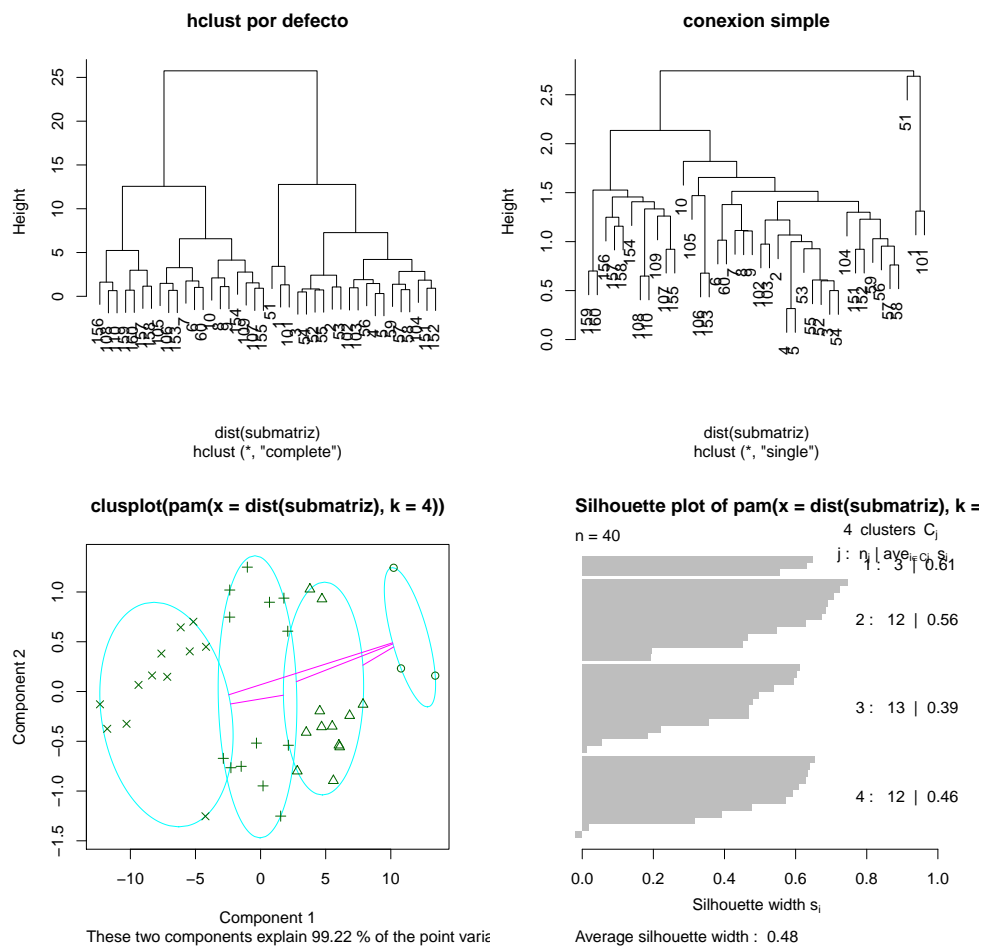


Figura 9: Gráficos de algunos procedimientos de clasificación y de evaluación

El último de los gráficos de la figura 9 es un gráfico de silueta que representa medidas de separación entre clases y dentro de las clases (ver la documentación de la función `silhouette`).

Para convertir un árbol de clasificación, como los que crea la función `hclust`, en un conjunto de etiquetas, se utiliza la función `cutree`. Con ella se pueden comparar las diversas clasificaciones.

```
> c1 <- cutree(clas1, k = 4)
> table(c1)

c1
 1  2  3  4
3 17 13  7

> c2 <- cutree(clas2, k = 4)
> table(c1, c2)

      c2
c1     1  2  3  4
 1     1  2  0  1  0
 2     0 17  0  0
 3     0  9  0  4
 4     0  0  0  7
```

Un método de partición como `pam` devuelve un vector de etiquetas directamente.

```
> table(clas4$clustering, c1)

      c1
      1  2  3  4
 1     3  0  0  0
 2     0 12  0  0
 3     0  5  8  0
 4     0  0  5  7
```

Con nuestros datos podemos dividir y etiquetar los cangrejos en cuatro clases según el sexo y la especie.

```
> subsp <- crabs$sp[ind]
> subsex <- crabs$sex[ind]
> clases <- paste(as.character(subsp), as.character(subsex), sep = "")
> table(clases, clas4$clustering)

clases 1 2 3 4
  BF 1 6 3 0
  BM 1 4 4 1
  OF 0 0 3 7
  OM 1 2 3 4
```

A la vista del resultado, no parece que exista un buen ajuste entre las cuatro clases definidas y la clasificación proporcionada por el `pam`.

8. Componentes principales

El análisis de las componentes principales ACP es una técnica multivariante de reducción de la dimensión. Se trata de obtener factores combinación lineal de las variables originales con máxima variabilidad y ortogonales entre sí. En la figura 3 se observa una gran correlación entre las variables estudiadas, esto facilita el ACP.

```
> cp <- prcomp(crabs5)
> cp
```

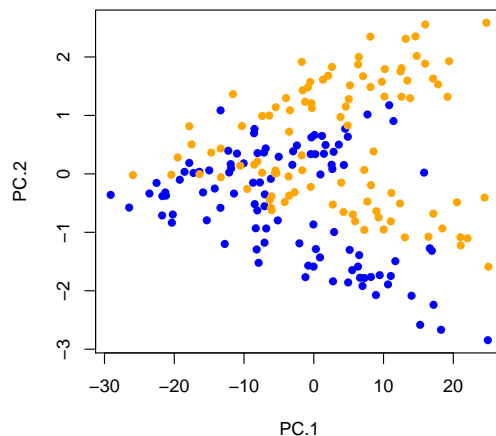


Figura 10: Gráfico de las dos primeras componentes principales

```
Standard deviations:
[1] 11.8619441  1.1387874  1.0001346  0.3678306  0.2791312
```

```
Rotation:
      PC1      PC2      PC3      PC4      PC5
FL 0.2889810  0.3232500 -0.5071698  0.7342907  0.1248816
RW 0.1972824  0.8647159  0.4141356 -0.1483092 -0.1408623
CL 0.5993986 -0.1982263 -0.1753299 -0.1435941 -0.7416656
CW 0.6616550 -0.2879790  0.4913755  0.1256282  0.4712202
BD 0.2837317  0.1598447 -0.5468821 -0.6343657  0.4386868
```

Este resultado indica que la mayor variabilidad se retiene con las dos primeras componentes principales. Éstas se ajustan a la clásica interpretación de tamaño, la primera, y forma, la segunda. El tamaño es una combinación ponderada de todas las variables con especial peso de las dos del caparazón. La forma viene definida por la contraposición de las variables del caparazón frente a las otras. En la figura 10 tenemos los datos de las dos primeras componentes principales con diferentes colores para las especies.

```
> plot(cp$x[, 1], cp$x[, 2],
+       col = ifelse(crabs$sp == "O", "orange", "blue"),
+       pch = 16, xlab = "PC.1", ylab = "PC.2")
```

Con el paquete `scatterplot3d` (hay que instalarlo) podemos examinar los gráficos en tres dimensiones de tres variables, CL, CW, RW en la figura 11, y de las tres primeras componentes principales en la figura 12. Se han fijado diferentes tipos de puntos para representar a las especies y el sexo.

```
> library(scatterplot3d)
> scatterplot3d(CL, CW, RW, color = ifelse(crabs$sp == "O", "orange", "blue"))
> scatterplot3d(cp$x[, 1], cp$x[, 2], cp$x[, 3],
+               color = ifelse(crabs$sp == "O", "orange", "blue"),
+               pch = ifelse(crabs$sex == "M", 1, 15))
```

También se pueden interpretar los gráficos de dispersión dos a dos de todas las componentes principales de la figura 13.

```
> pairs(cp$x, col = ifelse(crabs$sp == "O", "orange", "blue"),
+       pch = ifelse(crabs$sex == "M", 1, 16))
```

Finalmente, la representación de los datos se puede hacer con puntos y variables conjuntamente mediante una función `biplot` de las componentes principales (ver figura 14).

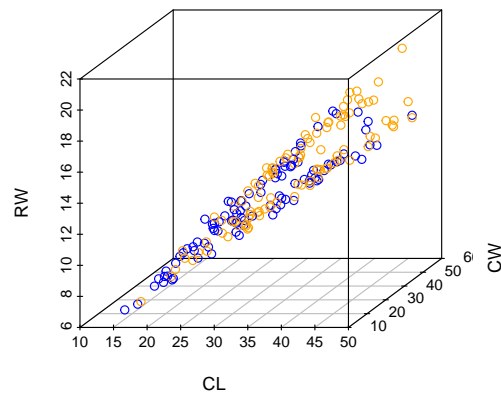


Figura 11: Gráfico 3D de las variables CL, CW y RW.

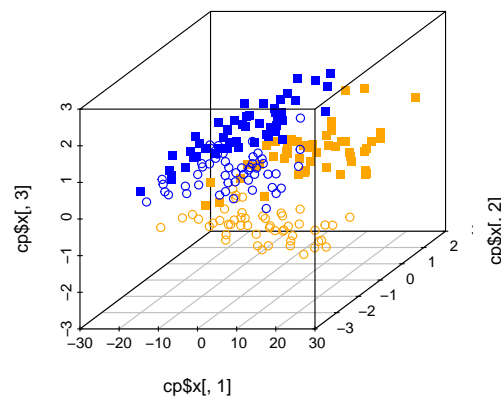


Figura 12: Gráfico 3D de las tres primeras componentes principales

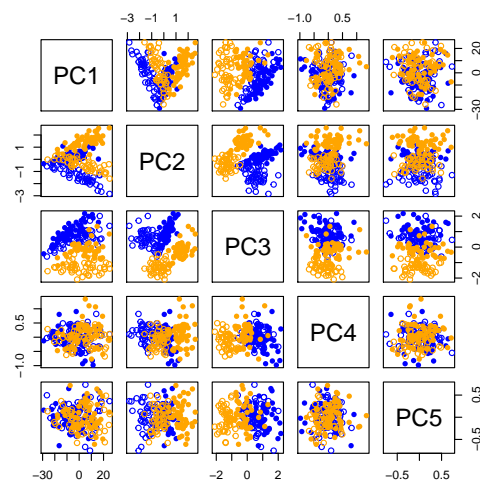


Figura 13: Gráfico de todas las componentes principales

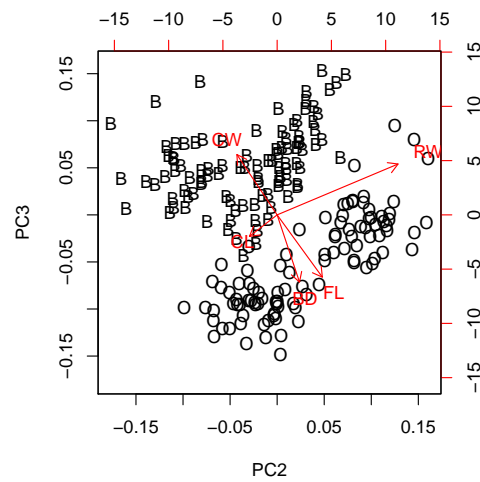


Figura 14: Gráfico biplot con PC2 y PC3

```
> rownames(matriz) <- as.character(crabs$sp)
> biplot(prcomp(matriz), choice = 2:3)
```


Referencias

- [1] V.J. Carey, *Machine learning in Bioconductor: Practical exercises*. 2005.
- [2] F. Carmona, *Modelos lineales*, Publicacions UB, 2005.
- [3] J.J. Faraway, *Linear Models with R*, Chapman & Hall/CRC, 2004.
- [4] P. Murrell, *R Graphics*, Chapman & Hall/CRC, 2005.
- [5] J. Verzani, *Using R for Introductory Statistics*. Chapman & Hall/CRC, 2004.