Universitat de les Illes Balears

#SOM UIB

AprendeR: Introducción al tratamiento de datos con R y RStudio

Módulo 6

Lección 10
Descripción de datos cuantitativos multidimensionales

Campus Extens
UIB Virtual

http://moocs.uib.cat

Edició: abril 2016
Edita: Campus Extens – UIB Virtual
Disseny portada: Direcció de l'Estratègia de Comunicació i Promoció Institucional (dircom.uib.cat)

Lección 10

Descripción de datos cuantitativos multidimensionales

En general, los datos que se recogen en experimentos son multidimensionales: medimos varias variables aleatorias sobre una misma muestra de individuos, y organizamos esta información en tablas de datos en las que las filas representan los individuos observados y cada columna corresponde a una variable diferente. En lecciones anteriores ya han aparecido datos cualitativos y ordinales multidimensionales, para los que hemos calculado y representado gráficamente sus frecuencias globales y marginales; en esta lección estudiamos algunos estadísticos específicos para resumir y representar la relación existente entre diversas variables cuantitativas.

10.1. Matrices de datos cuantitativos

Supongamos que hemos medido p características de n individuos u objetos; como resultado, hemos obtenido p variables cuantitativas, cada una formada por n observaciones. Estas variables están emparejadas, en el sentido de que la primera entrada de cada variable corresponde a un mismo individuo, la segunda entrada a otro individuo, y así sucesivamente. Para trabajar con estas observaciones, las disponemos en una tabla de datos donde cada fila corresponde a un individuo y cada columna, a una variable. En `R`, definiremos esta tabla en forma de *data frame*, pero, por conveniencia de lenguaje, en el texto de esta lección la representaremos como una matriz

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}.$$

Utilizaremos las notaciones siguientes:

- Denotaremos la i-ésima fila de X por

  $$x_{i\bullet} = (x_{i1}, x_{i2}, \dots, x_{ip}).$$

  Este vector está compuesto por las observaciones de las p variables sobre el i-ésimo individuo.

- Denotaremos la j-ésima columna de X por

  $$x_{\bullet j} = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix}.$$

  Esta columna está formada por todos los valores de la j-ésima variable.

Observad que, en cada caso, el punto en el subíndice representa el índice «variable» de los elementos del vector o de la columna.

De esta manera, podremos expresar la matriz de datos X tanto por filas como por columnas:

$$X = \begin{pmatrix} x_{1\bullet} \\ x_{2\bullet} \\ \vdots \\ x_{n\bullet} \end{pmatrix} = (x_{\bullet 1}, x_{\bullet 2}, \ldots, x_{\bullet p}).$$

Con estas notaciones, podemos generalizar al caso multidimensional los estadísticos de una variable cuantitativa, definiéndolos como los vectores que se obtienen aplicando el estadístico concreto a cada columna de la tabla de datos. Así:

- El *vector de medias* de X es el vector formado por las medias aritméticas de sus columnas:

  $$\overline{X} = (\overline{x}_{\bullet 1}, \overline{x}_{\bullet 2}, \ldots, \overline{x}_{\bullet p}),$$

  donde, para cada $j = 1, \ldots, p$,

  $$\overline{x}_{\bullet j} = \frac{1}{n} \sum_{i=1}^{n} x_{ij}.$$

- El *vector de varianzas* de X es el vector formado por las varianzas de sus columnas:

  $$s_X^2 = (s_1^2, s_2^2, \ldots, s_p^2),$$

  donde

  $$s_j^2 = \frac{1}{n} \sum_{i=1}^{n} (x_{ij} - \overline{x}_{\bullet j})^2.$$

- El *vector de varianzas muestrales* de X está formado por las varianzas muestrales de sus columnas:

  $$\widetilde{s}_X^2 = (\widetilde{s}_1^2, \widetilde{s}_2^2, \ldots, \widetilde{s}_p^2),$$

  donde

  $$\widetilde{s}_j^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \overline{x}_{\bullet j})^2 = \frac{n}{n-1} s_j^2.$$

- Los *vectores de desviaciones típicas* s_X y *de desviaciones típicas muestrales* $\widetilde{s}_X$ de X son los formados por las desviaciones típicas y las desviaciones típicas muestrales de sus columnas, respectivamente:

  $$s_X = (s_1, s_2, \ldots, s_p) = (\sqrt{s_1^2}, \sqrt{s_2^2}, \ldots, \sqrt{s_p^2})$$
  $$\widetilde{s}_X = (\widetilde{s}_1, \widetilde{s}_2, \ldots, \widetilde{s}_p) = (\sqrt{\widetilde{s}_1^2}, \sqrt{\widetilde{s}_2^2}, \ldots, \sqrt{\widetilde{s}_p^2})$$

Estos vectores de estadísticos se pueden calcular con `R` aplicando la función correspondiente al estadístico a todas las columnas de la tabla de datos. La manera más sencilla de hacerlo en un solo paso es usando la función `sapply`, si tenemos guardada la tabla como un *data frame*, o `apply` con `MARGIN=2`, si la tenemos guardada en forma de matriz.

Ejemplo 10.1. Consideremos la tabla de datos

$$X = \begin{pmatrix} 1 & -1 & 3 \\ 1 & 0 & 3 \\ 2 & 3 & 0 \\ 3 & 0 & 1 \end{pmatrix}$$

formada por 4 observaciones de 3 variables; por lo tanto, $n = 4$ y $p = 3$. Vamos a guardarla en un *data frame* y a calcular sus estadísticos.

```
> X=data.frame(V1=c(1,1,2,3),V2=c(-1,0,3,0),V3=c(3,3,0,1))
> X
  V1 V2 V3
1  1 -1  3
2  1  0  3
3  2  3  0
4  3  0  1
> sapply(X, mean) #Vector de medias
  V1   V2   V3
1.75 0.50 1.75
> sapply(X, var) #Vector de varianzas muestrales
       V1        V2        V3
0.9166667 3.0000000 2.2500000
> sapply(X, sd) #Vector de desviaciones típicas muestrales
       V1        V2        V3
0.9574271 1.7320508 1.5000000
> var_ver=function(x){var(x)*(length(x)-1)/length(x)}   #Varianza
   "verdadera"
> sd_ver=function(x){sqrt(var_ver(x))} #Desv. típica "verdadera"
> sapply(X, var_ver) #Vector de varianzas "verdaderas"
    V1     V2     V3
0.6875 2.2500 1.6875
> sapply(X, sd_ver) #Vector de desviaciones típicas "verdaderas"
       V1        V2        V3
0.8291562 1.5000000 1.2990381
```

A veces es conveniente aplicar una transformación lineal a una tabla de datos X, sumando a cada columna un valor y luego multiplicando cada columna resultante por otro valor. El ejemplo más común de trasformación lineal es la *tipificación de datos*.

Dada una variable cuantitativa,[1] su *variable tipificada* es el vector que se obtiene restando a cada entrada la media aritmética de la variable y dividiendo el resultado por su desviación típica; de esta manera, la variable tipificada tiene media aritmética 0 y varianza 1. Tipificar una variable es conveniente cuando se quiere trabajar con sus datos sin que influyan las unidades en los que están medidos: al dividir por su desviación típica, los valores resultantes son adimensionales.

Tipificar las variables de una tabla de datos permite compararlas dejando de lado las diferencias que pueda haber entre sus valores medios o sus varianzas. Llamaremos *matriz tipificada* de

[1] De ahora en adelante, supondremos que todas las variables cuantitativas que aparezcan en lo que queda de lección, incluidas las columnas de tablas de datos, son no constantes y, por lo tanto, tienen desviación típica no nula.

una matriz de datos X a la matriz Z que se obtiene tipificando cada columna; es decir, para tipificar una matriz de datos X, primero restamos a cada columna su media aritmética (llamaremos *matriz centrada* de X a la matriz obtenida en este paso), y, a continuación, dividimos cada columna de la matriz centrada por su desviación típica, que coincide con la desviación típica de la columna original de X:

$$Z = \begin{pmatrix} \frac{x_{11}-\overline{x}_{\bullet 1}}{s_1} & \frac{x_{12}-\overline{x}_{\bullet 2}}{s_2} & \cdots & \frac{x_{1p}-\overline{x}_{\bullet p}}{s_p} \\ \frac{x_{21}-\overline{x}_{\bullet 1}}{s_1} & \frac{x_{22}-\overline{x}_{\bullet 2}}{s_2} & \cdots & \frac{x_{2p}-\overline{x}_{\bullet p}}{s_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{x_{n1}-\overline{x}_{\bullet 1}}{s_1} & \frac{x_{n2}-\overline{x}_{\bullet 2}}{s_2} & \cdots & \frac{x_{np}-\overline{x}_{\bullet p}}{s_p} \end{pmatrix}.$$

La manera más sencilla de aplicar una transformación lineal a una tabla de datos X, y en particular de tipificarla, es usando la instrucción

```
scale(X, center=..., scale=...)
```

donde:

- `X` puede ser tanto una matriz como un *data frame*; el resultado será un objeto de la misma clase.
- El valor del parámetro `center` es el vector que restamos a sus columnas, en el sentido de que cada entrada de este vector se restará a todas las entradas de la columna correspondiente. Su valor por defecto (que no es necesario especificar, aunque también se puede especificar con `center=TRUE`) es el vector $\overline{X}$ de medias de X; para especificar que no se reste nada, podemos usar `center=FALSE`.
- El valor del parámetro `scale` es el vector por el que dividimos las columnas de la matriz obtenida tras restar el valor de `center`: cada columna se divide por la entrada correspondiente de este vector. Su valor por defecto (de nuevo, se puede especificar igualando el parámetro a `TRUE`) es el vector $\widetilde{s}_X$ de desviaciones típicas *muestrales*; para especificar que no se divida por nada, podemos usar `scale=FALSE`.

En particular, la instrucción `scale(X)` centra la tabla de datos X y divide sus columnas por sus *desviaciones típicas muestrales*; por lo tanto, no la tipifica según nuestra definición, ya que no las divide por sus desviaciones típicas «verdaderas».

Ejemplo 10.2. Vamos a centrar la tabla de datos X del Ejemplo 10.1.

```
> X
  V1 V2 V3
1  1 -1  3
2  1  0  3
3  2  3  0
4  3  0  1
> X_centrada=scale(X, center=TRUE, scale=FALSE)
> X_centrada
        V1   V2    V3
[1, ] -0.75 -1.5  1.25
```

```
[2, ] -0.75 -0.5  1.25
[3, ]  0.25  2.5 -1.75
[4, ]  1.25 -0.5 -0.75
attr(, "scaled:center")
  V1   V2   V3
1.75 0.50 1.75
```

Observad la estructura del resultado: en primer lugar nos da la matriz centrada, y a continuación nos dice que tiene un atributo llamado `"scaled:center"` cuyo valor es el vector usado para centrarla. Este atributo no interferirá para nada en las operaciones que se realicen con la matriz centrada, pero, si os molesta, recordad de la Lección 3 que se puede eliminar sustituyendo el resultado de centrar la matriz en los puntos suspensivos de la instrucción siguiente:

`attr(..., "scaled:center")=NULL`.

```
> attr(X_centrada, "scaled:center")=NULL
> X_centrada
        V1   V2    V3
[1, ] -0.75 -1.5  1.25
[2, ] -0.75 -0.5  1.25
[3, ]  0.25  2.5 -1.75
[4, ]  1.25 -0.5 -0.75
```

Como ya hemos avisado, para tipificar esta tabla de datos *no* podemos hacer lo siguiente:

```
> X_tip=scale(X)
> X_tip
             V1         V2         V3
[1, ] -0.7833495 -0.8660254  0.8333333
[2, ] -0.7833495 -0.2886751  0.8333333
[3, ]  0.2611165  1.4433757 -1.1666667
[4, ]  1.3055824 -0.2886751 -0.5000000
attr(, "scaled:center")
  V1   V2   V3
1.75 0.50 1.75
attr(, "scaled:scale")
       V1        V2        V3
0.9574271 1.7320508 1.5000000
```

Para hacerlo bien en base a la definición que hemos dado, tenemos dos opciones. Una posibilidad es multiplicar la matriz anterior por $\sqrt{n/(n-1)}$, donde n es el número de filas de la tabla.[2]

```
> n=dim(X)[1]  #Número de filas de X
> n
[1] 4
> X_tip=scale(X)*sqrt(n/(n-1))
> X_tip
```

[2] Como $\widetilde{s}_X = \sqrt{\frac{n}{n-1}} \cdot s_X$, se tiene que $\frac{1}{s_X} = \sqrt{\frac{n}{n-1}} \cdot \frac{1}{\widetilde{s}_X}$; por lo tanto, si queríamos dividir por s_X y `scale(X)` ha dividido por $\widetilde{s}_X$, basta multiplicar su resultado por $\sqrt{\frac{n}{n-1}}$.

```
              V1          V2          V3
[1, ] -0.9045340 -1.0000000  0.9622504
[2, ] -0.9045340 -0.3333333  0.9622504
[3, ]  0.3015113  1.6666667 -1.3471506
[4, ]  1.5075567 -0.3333333 -0.5773503
attr(, "scaled:center")
  V1   V2   V3
1.75 0.50 1.75
attr(, "scaled:scale")
       V1        V2        V3
0.9574271 1.7320508 1.5000000
```

Otra posibilidad es usar, como valor del parámetro `scale`, el vector s_X de desviaciones típicas de las columnas.

```
> sd_ver=function(x){sqrt(var(x)*(length(x)-1)/length(x))}
> X_dtv=sapply(X, sd_ver) #Desviaciones típicas "verdaderas"
> X_tip1=scale(X, scale=X_dtv) #Escalamos dividiendo las columnas
   por X_dtv
> X_tip1
              V1          V2          V3
[1, ] -0.9045340 -1.0000000  0.9622504
[2, ] -0.9045340 -0.3333333  0.9622504
[3, ]  0.3015113  1.6666667 -1.3471506
[4, ]  1.5075567 -0.3333333 -0.5773503
attr(, "scaled:center")
  V1   V2   V3
1.75 0.50 1.75
attr(, "scaled:scale")
       V1        V2        V3
0.8291562 1.5000000 1.2990381
```

Observaréis que la matriz resultante es la misma, pero el atributo que indica el vector por el que hemos dividido las columnas es diferente: ahora ha sido el de desviaciones típicas. En ambos casos, podemos usar la función `attr` para eliminar uno a uno los dos atributos, `"scaled:center"` y `"scaled:scale"`, que se han añadido a la matriz tipificada.

```
> attr(X_tip, "scaled:center")=NULL
> attr(X_tip, "scaled:scale")=NULL
> X_tip
              V1          V2          V3
[1, ] -0.9045340 -1.0000000  0.9622504
[2, ] -0.9045340 -0.3333333  0.9622504
[3, ]  0.3015113  1.6666667 -1.3471506
[4, ]  1.5075567 -0.3333333 -0.5773503
```

10.2. Covarianzas y correlaciones

La *covarianza* entre dos variables es una medida de la propensión que tienen ambas variables a variar conjuntamente. Cuando la covarianza es positiva, si una de las dos variables crece o

decrece, la otra tiene el mismo comportamiento; en cambio, cuando la covarianza es negativa, esta tendencia se invierte: si una variable crece, la otra decrece y viceversa. Por desgracia, interpretar el valor de la covarianza más allá de su signo es difícil, por lo que introduciremos una versión «normalizada» de la misma, la *correlación de Pearson*, que mide de manera más precisa la relación lineal entre dos variables.

La covarianza generaliza la varianza, en el sentido de que la varianza de una variable es su covarianza consigo misma. Y como en el caso de la varianza, definiremos dos versiones de la covarianza: la «*verdadera*», que se usa para medir la tendencia a la variación conjunta de dos conjuntos específicos de datos, y la *muestral*, que además aproxima mejor la covarianza de las variables definidas sobre la población total. La diferencia estará de nuevo en el denominador.

Formalmente, la *covarianza* de las variables $x_{\bullet i}$ y $x_{\bullet j}$ de una matriz de datos X es

$$s_{ij} = \frac{1}{n}\sum_{k=1}^{n}\big((x_{ki}-\overline{x}_{\bullet i})(x_{kj}-\overline{x}_{\bullet j})\big) = \frac{1}{n}\Big(\sum_{k=1}^{n} x_{ki}x_{kj}\Big) - \overline{x}_{\bullet i}\overline{x}_{\bullet j},$$

y su *covarianza muestral* es

$$\widetilde{s}_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}\big((x_{ki}-\overline{x}_{\bullet i})(x_{kj}-\overline{x}_{\bullet j})\big) = \frac{n}{n-1}s_{ij}.$$

Es inmediato comprobar a partir de sus definiciones que ambas covarianzas son simétricas, y que la covarianza, tanto «verdadera» como muestral, de una variable consigo misma es su correspondiente varianza:

$$s_{ij} = s_{ji}, \quad \widetilde{s}_{ij} = \widetilde{s}_{ji}, \quad s_{ii} = s_i^2, \quad \widetilde{s}_{ii} = \widetilde{s}_i^2.$$

Ejemplo 10.3. La covarianza de las dos primeras columnas de la matriz de datos

$$X = \begin{pmatrix} 1 & -1 & 3 \\ 1 & 0 & 3 \\ 2 & 3 & 0 \\ 3 & 0 & 1 \end{pmatrix}.$$

del Ejemplo 10.1 se calcularía de la manera siguiente, teniendo en cuenta que sus medias son 1.75 y 0.5, respectivamente:

$$s_{12} = \frac{1}{4}(1\cdot(-1)+1\cdot 0+2\cdot 3+3\cdot 0) - 1.75\cdot 0.5 = 1.25 - 0.875 = 0.375.$$

Su covarianza muestral se obtendría multiplicando este valor por $4/3$:

$$\widetilde{s}_{12} = \frac{4}{3}s_{12} = 0.5.$$

La covarianza *muestral* de dos vectores numéricos de la misma longitud n se puede calcular con `R` mediante la función `cov`. Para obtener su covarianza «verdadera», hay que multiplicar el resultado de `cov` por $(n-1)/n$.

```
> X
  V1 V2 V3
```

```
1  1 -1  3
2  1  0  3
3  2  3  0
4  3  0  1
> cov(X$V1, X$V2)   #Covarianza MUESTRAL
[1] 0.5
> (3/4)*cov(X$V1, X$V2)  #Covarianza "verdadera"
[1] 0.375
```

Queremos recalcar que, como en el caso de la varianza con `var`, `R` calcula con `cov` la versión muestral de la covarianza.

Las *matrices de covarianzas* y de *covarianzas muestrales* de una tabla de datos X son, respectivamente,

$$\mathbf{S} = \begin{pmatrix} s_{11} & s_{12} & \dots & s_{1p} \\ s_{21} & s_{22} & \dots & s_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{p1} & s_{p2} & \dots & s_{pp} \end{pmatrix}, \qquad \widetilde{\mathbf{S}} = \begin{pmatrix} \widetilde{s}_{11} & \widetilde{s}_{12} & \dots & \widetilde{s}_{1p} \\ \widetilde{s}_{21} & \widetilde{s}_{22} & \dots & \widetilde{s}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{s}_{p1} & \widetilde{s}_{p2} & \dots & \widetilde{s}_{pp} \end{pmatrix},$$

donde cada s_{ij} y cada $\widetilde{s}_{ij}$ son, respectivamente, la covarianza y la covarianza muestral de las correspondientes columnas $x_{\bullet i}$ y $x_{\bullet j}$. Estas matrices de covarianzas miden la tendencia a la variabilidad conjunta de los datos de X.

La matriz de covarianzas muestrales de X se calcula aplicando la función `cov` al *data frame* o a la matriz que contenga dicha tabla. Para obtener su matriz de covarianzas «verdaderas», es suficiente multiplicar el resultado de `cov` por $(n-1)/n$, donde n es el número de filas de X.

```
> X
  V1 V2 V3
1  1 -1  3
2  1  0  3
3  2  3  0
4  3  0  1
> n=dim(X)[1]
> cov(X)  #Matriz de covarianzas muestrales
            V1        V2        V3
V1  0.9166667  0.500000 -1.083333
V2  0.5000000  3.000000 -2.166667
V3 -1.0833333 -2.166667  2.250000
> ((n-1)/n)*cov(X)  #Matriz de covarianzas "verdaderas"
        V1     V2      V3
V1  0.6875  0.375 -0.8125
V2  0.3750  2.250 -1.6250
V3 -0.8125 -1.625  1.6875
```

La *correlación lineal de Pearson* (o, de ahora en adelante, simplemente *correlación*) de las variables $x_{\bullet i}$ y $x_{\bullet j}$ de X es

$$r_{ij} = \frac{s_{ij}}{s_i s_j}.$$

Observad que

$$\frac{\widetilde{s}_{ij}}{\widetilde{s}_i \cdot \widetilde{s}_j} = \frac{\frac{n}{n-1} \cdot s_{ij}}{\sqrt{\frac{n}{n-1}} \cdot s_i \cdot \sqrt{\frac{n}{n-1}} \cdot s_j} = \frac{s_{ij}}{s_i \cdot s_j} = r_{ij},$$

por lo que esta correlación se puede calcular también a partir de las versiones muestrales de la covarianza y las desviaciones típicas por medio de la misma fórmula.

La correlación r_{ij} tiene las propiedades siguientes:

- Es símétrica: $r_{ij} = r_{ji}$.
- $-1 \leqslant r_{ij} \leqslant 1$.
- $r_{ii} = 1$.
- r_{ij} tiene el mismo signo que s_{ij}.
- $r_{ij} = \pm 1$ si y, sólo si, existe una relación lineal perfecta entre las variables $x_{\bullet i}$ y $x_{\bullet j}$: es decir, si, y sólo si, existen valores $a, b \in \mathbb{R}$ tales que

$$\begin{pmatrix} x_{1j} \\ \vdots \\ x_{nj} \end{pmatrix} = a \cdot \begin{pmatrix} x_{1i} \\ \vdots \\ x_{ni} \end{pmatrix} + b.$$

  La pendiente a de esta relación lineal tiene el mismo signo que r_{ij}.
- El coeficiente de determinación R^2 de la regresión lineal por mínimos cuadrados de $x_{\bullet j}$ respecto de $x_{\bullet i}$ (véase la Lección 2) es igual al cuadrado de su correlación, r_{ij}^2; por lo tanto, cuánto más se aproxime el valor absoluto de r_{ij} a 1, más se acercan las variables $x_{\bullet i}$ y $x_{\bullet j}$ a depender linealmente la una de la otra.

Así pues, la correlación entre dos variables viene a ser una covarianza «normalizada», ya que, como vemos, su valor está entre -1 y 1, y mide la tendencia de las variables a estar relacionadas según una función lineal. En concreto, cuanto más se acerca la correlación a 1 (respectivamente, a -1), más se acerca una (cualquiera) de las variables a ser función lineal creciente (respectivamente, decreciente) de la otra.

Con `R`, la correlación de Pearson de dos vectores se puede calcular por medio de la función `cor`.

Ejemplo 10.4. En ejemplos anteriores hemos calculado la covarianza y las varianzas de las dos primeras columnas de la matriz de datos

$$X = \begin{pmatrix} 1 & -1 & 3 \\ 1 & 0 & 3 \\ 2 & 3 & 0 \\ 3 & 0 & 1 \end{pmatrix}.$$

Hemos obtenido los valores siguientes:

$$s_{12} = 0.375, \quad s_1 = \frac{\sqrt{11}}{4} = 0.82916, \quad s_2 = \frac{3}{2} = 1.5.$$

Por lo tanto, su correlación es

$$r_{12} = \frac{0.375}{0.82916 \cdot 1.5} = 0.3015.$$

Ahora vamos a calcularla con `R`, y aprovecharemos para confirmar su relación con el valor de R^2 de la regresión lineal de la segunda columna respecto de la primera.

```
> X
  V1 V2 V3
1  1 -1  3
2  1  0  3
3  2  3  0
4  3  0  1
> cor(X$V1, X$V2)
[1] 0.3015113
> cor(X$V1, X$V2)^2
[1] 0.09090909
> summary(lm(X$V2~X$V1))$r.squared
[1] 0.09090909
```

La *matriz de correlaciones* de X es

$$\mathbf{R} = \begin{pmatrix} 1 & r_{12} & \dots & r_{1p} \\ r_{21} & 1 & \dots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} & \dots & 1 \end{pmatrix},$$

donde cada r_{ij} es la correlación de las columnas correspondientes de X. Esta matriz de correlaciones se puede calcular con `R` con la misma instrucción `cor`, que se puede aplicar tanto a una matriz como a un *data frame*.

```
> X
  V1 V2 V3
1  1 -1  3
2  1  0  3
3  2  3  0
4  3  0  1
> cor(X)
           V1         V2         V3
V1  1.0000000  0.3015113 -0.7543365
V2  0.3015113  1.0000000 -0.8339504
V3 -0.7543365 -0.8339504  1.0000000
```

`R` también dispone de la función `cov2cor` que, aplicada a la matriz de covarianzas (muestrales o «verdaderas»), calcula la matriz de correlaciones de los datos originales.

```
> S=cov(X)
> cov2cor(S)
           V1         V2         V3
V1  1.0000000  0.3015113 -0.7543365
V2  0.3015113  1.0000000 -0.8339504
V3 -0.7543365 -0.8339504  1.0000000
```

Se tiene el teorema siguiente, que se puede demostrar mediante un simple, aunque farragoso, cálculo algebraico:

Teorema 10.1. La matriz de correlaciones de X es igual a la matriz de covarianzas de su matriz tipificada.

La importancia de este resultado es que, si la tabla de datos es muy grande, suele ser más eficiente tipificar primero la tabla y luego calcular la matriz de covarianzas de la tabla tipificada que calcular directamente la matriz de correlaciones de la tabla original. Comprobemos que el teorema es cierto para nuestra matriz de datos.

```
> X
  V1 V2 V3
1  1 -1  3
2  1  0  3
3  2  3  0
4  3  0  1
> n=dim(X)[1]
> X_tip=scale(X)*sqrt(n/(n-1))
> cov(X_tip)*(n-1)/n
           V1          V2          V3
V1  1.0000000  0.3015113 -0.7543365
V2  0.3015113  1.0000000 -0.8339504
V3 -0.7543365 -0.8339504  1.0000000
```

Cuando se calcula la covarianza o la correlación de dos vectores que contienen `NA`, lo usual es no tenerlos en cuenta: es decir, si un vector contiene un `NA` en una posición, se eliminan de ambos vectores sus entradas en dicha posición. De esta manera, se tomaría como covarianza de

$$\begin{pmatrix} 1 \\ 2 \\ NA \\ 4 \\ 6 \\ 2 \end{pmatrix} \text{ y } \begin{pmatrix} 2 \\ 4 \\ -3 \\ 5 \\ 7 \\ NA \end{pmatrix}$$

la de

$$\begin{pmatrix} 1 \\ 2 \\ 4 \\ 6 \end{pmatrix} \text{ y } \begin{pmatrix} 2 \\ 4 \\ 5 \\ 7 \end{pmatrix}.$$

Al aplicar `cov` o `cor` a un par de vectores que contengan `NA`, se obtiene, por defecto, `NA`. Si se quiere que `R` calcule el valor sin tener en cuenta los `NA`, se ha de especificar añadiendo el parámetro `use="complete.obs"` (que le indica que ha de usar las *observaciones completas*, es decir, las posiciones que no tienen `NA` en ninguno de los dos vectores).

```
> x=c(1,2,NA,4,6,2)
> y=c(2,4,-3,5,7,NA)
> x1=c(1,2,4,6)  #Quitamos las entradas 3a y 6a
> x2=c(2,4,5,7)  #Quitamos las entradas 3a y 6a
> cov(x, y)
[1] NA
> cov(x, y, use="complete.obs")
[1] 4.5
> cov(x1, x2)
[1] 4.5
> cor(x, y)
```

```
[1] NA
> cor(x, y, use="complete.obs")
[1] 0.9749135
> cor(x1, x2)
[1] 0.9749135
```

Al calcular las matrices de covarianzas, covarianzas muestrales o correlaciones de una tabla de datos que contenga `NA`, se suele seguir una de las dos estrategias siguientes, según lo que interese al usuario:

- Para cada par de columnas, se calcula su covarianza o su correlación con la estrategia explicada más arriba para dos vectores, obviando el hecho de que forman parte de una tabla de datos mayor; es decir, al efectuar el cálculo para un par de columnas concreto, se eliminan de cada una de ellas sus entradas `NA` y aquellas en cuya fila la otra tiene un `NA`. Esta opción se especifica dentro de la función `cov` o `cor` con el parámetro `use="pairwise.complete.obs"` (observaciones completas por parejas).
- Antes de nada, se eliminan las filas de la tabla que contienen algún `NA` en alguna columna, dejando solo en la tabla las filas «completas», las que no contienen ningún `NA`. Luego se calcula la matriz de covarianzas o de correlaciones de la tabla resultante. Esta opción se especifica con el parámetro `use="complete.obs"`.

Veamos un ejemplo:

```
> X=cbind(c(1,2,NA,4,6,2), c(2,4,-3,5,7,NA), c(-2,1,0,2,3,0))
> X
     [, 1] [, 2] [, 3]
[1, ]     1     2   -2
[2, ]     2     4    1
[3, ]    NA    -3    0
[4, ]     4     5    2
[5, ]     6     7    3
[6, ]     2    NA    0
> cov(X)
     [, 1] [, 2]      [, 3]
[1, ]    NA    NA         NA
[2, ]    NA    NA         NA
[3, ]    NA    NA 3.066667
> cov(X, use="pairwise.complete.obs")
     [, 1]  [, 2]     [, 3]
[1, ]   4.0   4.50 3.500000
[2, ]   4.5 14.50 4.750000
[3, ]   3.5   4.75 3.066667
> cov(X, use="complete.obs")
          [, 1]     [, 2]     [, 3]
[1, ] 4.916667 4.500000 4.333333
[2, ] 4.500000 4.333333 4.333333
[3, ] 4.333333 4.333333 4.666667
> Y=cbind(c(1,2,4,6), c(2,4,5,7), c(-2,1,2,3)) #Eliminamos las
   filas con algún NA
> Y
```

```
     [, 1] [, 2] [, 3]
[1, ]    1    2   -2
[2, ]    2    4    1
[3, ]    4    5    2
[4, ]    6    7    3
> cov(Y)  #Dará lo mismo que con use="complete.obs"
         [, 1]     [, 2]     [, 3]
[1, ] 4.916667 4.500000 4.333333
[2, ] 4.500000 4.333333 4.333333
[3, ] 4.333333 4.333333 4.666667
```

Ejemplo 10.5. Recordaréis el *data frame* `iris`, que tabulaba las longitudes y anchuras de los pétalos y los sépalos de una muestra de flores iris de tres especies. Vamos a extraer un sub*data frame* con sus cuatro variables numéricas y calcularemos sus matrices de covarianzas y correlaciones.

```
 > str(iris)
'data.frame': 150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa", "versicolor", ..: 1 1
    1 1 1 1 1 1 1 1 ...
> iris_num=iris[, 1:4]
> cov(iris_num)  #Covarianzas muestrales
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    0.6856935  -0.0424340    1.2743154   0.5162707
Sepal.Width    -0.0424340   0.1899794   -0.3296564  -0.1216394
Petal.Length    1.2743154  -0.3296564    3.1162779   1.2956094
Petal.Width     0.5162707  -0.1216394    1.2956094   0.5810063
> n=dim(iris_num)[1]    #Número de filas; son 150, recordemos
> cov(iris_num)*(n-1)/n  #Covarianzas "verdaderas"
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length   0.68112222 -0.04215111    1.2658200   0.5128289
Sepal.Width   -0.04215111  0.18871289   -0.3274587  -0.1208284
Petal.Length   1.26582000 -0.32745867    3.0955027   1.2869720
Petal.Width    0.51282889 -0.12082844    1.2869720   0.5771329
> cor(iris_num)    #Correlaciones
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

Observamos, por ejemplo, una gran correlación lineal positiva entre la longitud y la anchura de los pétalos, 0.9628654, lo que indica una estrecha relación lineal con pendiente positiva entre estas magnitudes. Valdría la pena, entonces, calcular la recta de regresión lineal de una de estas medidas en función de la otra:

```
> lm(Petal.Length~Petal.Width, data=iris_num)
```

```
Call:
lm(formula = Petal.Length ~ Petal.Width, data = iris_num)

Coefficients:
(Intercept)  Petal.Width
      1.084        2.230

> summary(lm(Petal.Length~Petal.Width, data=iris_num))$r.squared
[1] 0.9271098
```

En cambio, la correlación entre la longitud y la anchura de los sépalos es -0.1175698, muy cercana a cero, lo que es señal de que la variación conjunta de las longitudes y anchuras de los sépalos no tiene una tendencia clara.

Vamos a ordenar ahora los pares de variables numéricas de `iris` en orden decreciente de su correlación en valor absoluto, para saber cuáles están más correlacionadas. Para ello, en primer lugar creamos un *data frame* cuyas filas están formadas por pares diferentes de variables numéricas de `iris`, su correlación y el valor absoluto de ésta, y a continuación ordenamos las filas de este *data frame* en orden decreciente de estos valores absolutos. Todo esto lo llevamos a cabo con el código siguiente, que luego explicamos:

```
> medidas=names(iris_num)
> n=length(medidas)  #En este caso, n=4
> indices=upper.tri(diag(n))
> medida1=matrix(rep(medidas, times=n), nrow=n,
  byrow=FALSE)[indices]
> medida2=matrix(rep(medidas, times=n), nrow=n,
  byrow=TRUE)[indices]
> corrs=as.vector(cor(iris_num))[indices]
> corrs.abs=abs(corrs)
> corrs_df=data.frame(medida1, medida2, corrs, corrs.abs)
> corrs_df
       medida1      medida2       corrs corrs.abs
1 Sepal.Length  Sepal.Width -0.1175698 0.1175698
2 Sepal.Length Petal.Length  0.8717538 0.8717538
3  Sepal.Width Petal.Length -0.4284401 0.4284401
4 Sepal.Length  Petal.Width  0.8179411 0.8179411
5  Sepal.Width  Petal.Width -0.3661259 0.3661259
6 Petal.Length  Petal.Width  0.9628654 0.9628654
> corrs_df_sort=corrs_df[order(corrs_df$corrs.abs,
   decreasing=TRUE), ]
> corrs_df_sort
       medida1      medida2       corrs corrs.abs
6 Petal.Length  Petal.Width  0.9628654 0.9628654
2 Sepal.Length Petal.Length  0.8717538 0.8717538
4 Sepal.Length  Petal.Width  0.8179411 0.8179411
3  Sepal.Width Petal.Length -0.4284401 0.4284401
5  Sepal.Width  Petal.Width -0.3661259 0.3661259
1 Sepal.Length  Sepal.Width -0.1175698 0.1175698
```

Vemos que el par de variables con mayor correlación en valor absoluto son `Petal.Length` y `Petal.Width`, como ya habíamos observado, seguidos por `Petal.Length` y `Sepal.Length`.

Vamos a explicar el código. La función `upper.tri`, aplicada a una matriz cuadrada M, produce la matriz *triangular superior* de valores lógicos del mismo orden que M, cuyas entradas (i, j) con $i < j$ son todas `TRUE` y el resto todas `FALSE`. Existe una función similar, `lower.tri`, para producir matrices *triangulares inferiores* de valores lógicos.

```
> upper.tri(diag(4))
      [,1]  [,2]  [,3]  [,4]
[1,] FALSE  TRUE  TRUE  TRUE
[2,] FALSE FALSE  TRUE  TRUE
[3,] FALSE FALSE FALSE  TRUE
[4,] FALSE FALSE FALSE FALSE
> lower.tri(diag(4))
      [,1]  [,2]  [,3]  [,4]
[1,] FALSE FALSE FALSE FALSE
[2,]  TRUE FALSE FALSE FALSE
[3,]  TRUE  TRUE FALSE FALSE
[4,]  TRUE  TRUE  TRUE FALSE
```

Ambas funciones disponen del parámetro `diag` que, igualado a `TRUE`, define también como `TRUE` las entradas de la diagonal principal.

```
> upper.tri(diag(4), diag=TRUE)
      [,1]  [,2]  [,3] [,4]
[1,]  TRUE  TRUE  TRUE TRUE
[2,] FALSE  TRUE  TRUE TRUE
[3,] FALSE FALSE  TRUE TRUE
[4,] FALSE FALSE FALSE TRUE
```

Si M es una matriz y L es una matriz de valores lógicos del mismo orden, `M[L]` produce el vector construido de la manera siguiente: de cada columna, se queda sólo con las entradas de M cuya entrada correspondiente en L es `TRUE`, y a continuación concatena estas columnas, de izquierda a derecha, en un vector.

```
> M=matrix(1:16, nrow=4, byrow=T)
> M
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
[4,]   13   14   15   16
> M[upper.tri(diag(4))] #Las entradas del triángulo superior, por
   columnas
[1]  2  3  7  4  8 12
```

Ahora, tenemos las matrices siguientes:

```
> matrix(rep(medidas, times=4), nrow=4, byrow=FALSE)
     [,1]           [,2]           [,3]           [,4]
[1,] "Sepal.Length" "Sepal.Length" "Sepal.Length" "Sepal.Length"
[2,] "Sepal.Width"  "Sepal.Width"  "Sepal.Width"  "Sepal.Width"
```

```
[3,] "Petal.Length" "Petal.Length" "Petal.Length" "Petal.Length"
[4,] "Petal.Width"  "Petal.Width"  "Petal.Width"  "Petal.Width"
> matrix(rep(medidas, times=4), nrow=4, byrow=TRUE)
     [,1]           [,2]          [,3]           [,4]
[1,] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
[2,] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
[3,] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
[4,] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
```

Por lo tanto, al aplicar estas matrices a la matriz de valores lógicos `upper.tri(diag(4))` obtenemos los nombres de las variables correspondientes a las filas y las columnas del triángulo superior, respectivamente, y al aplicar la matriz de correlaciones a esta matriz de valores lógicos, obtenemos sus entradas en este triángulo; en los tres vectores, las entradas siguen el mismo orden. Esto nos permite construir el *data frame* `corrs_df` cuyas filas están formadas por pares diferentes de variables numéricas de `iris`, su correlación y, aplicando `abs` a esta última variable, su correlación en valor absoluto.

Finalmente, la función `order` ordena los valores del vector al que se aplica, en orden decreciente si se especifica el parámetro `decreasing=TRUE`. Cuando aplicamos un *data frame* a una de sus variables reordenada de esta manera, reordena sus filas según el orden de esta variable. En este caso hubiéramos conseguido lo mismo con la función `sort`, pero la función `order` se puede aplicar a más de una variable del *data frame*: esto permite ordenar las filas del *data frame* en el orden de la primera variable de manera que, en caso de empate, queden ordenadas por la segunda variable, y así sucesivamente.

10.3. Representación gráfica de datos multidimensionales

La representación gráfica de tablas de datos multidimensionales tiene la dificultad de las dimensiones; para dos o tres variables es sencillo visualizar las relaciones entre las mismas, pero para más variables ya no nos bastan nuestras tres dimensiones espaciales y tenemos que usar algunos trucos, tales como representaciones gráficas conjuntas de pares de variables.

Cuando tenemos una tabla de datos formada por dos variables numéricas, la manera más sencilla de representarlos gráficamente es mediante la función `plot` aplicada a la matriz de datos o al *data frame*. Con esta función obtenemos un gráfico de los puntos definidos por las filas de la tabla: en el contexto de la estadística multidimensional, se le llama el *diagrama de dispersión* (*scatter plot*) de los datos.

A modo de ejemplo, si extrajéramos de la tabla `iris` una subtabla conteniendo sólo las longitudes y anchuras de los pétalos y quisiéramos visualizar las relación entre estas dimensiones, podríamos dibujar su diagrama de dispersión de la manera siguiente:

```
> iris.pet=iris[ ,c("Petal.Length","Petal.Width")]
> plot(iris.pet, pch=20, xlab="Largo", ylab="Ancho")
```

El resultado es la Figura 10.1, que muestra una clara tendencia positiva: cuanto más largos son los pétalos, más anchos tienden a ser. Esto se corresponde con la correlación de 0.9628654 que hemos obtenido en el Ejemplo 10.5.

Para tablas de datos de tres columnas numéricas, podemos usar con un fin similar la instrucción `scatterplot3d` del paquete homónimo, que dibuja un diagrama de dispersión tridimensio-

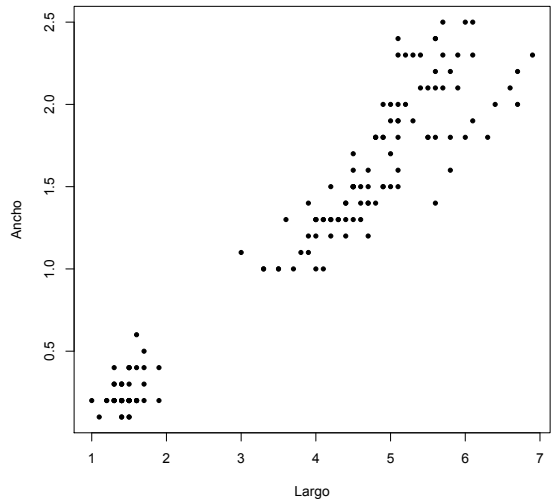

Figura 10.1. Diagrama de dispersión de las longitudes y anchuras de los pétalos de las flores representadas en la tabla `iris`.

nal. Como `plot`, se puede aplicar a un *data frame* o a una matriz; por ejemplo, para representar gráficamente las tres primeras variables numéricas de `iris`, podríamos hacer lo siguiente:

```
> #Instalamos y cargamos el paquete scatterplot3d
...
> scatterplot3d(iris[ , 1:3], pch=20)
```

Obtendríamos la Figura 10.2. Podéis consultar la Ayuda de la instrucción para saber cómo modificar su apariencia: cómo ponerle un título, etiquetar los ejes, usar colores, cambiar el estilo del gráfico, etc.

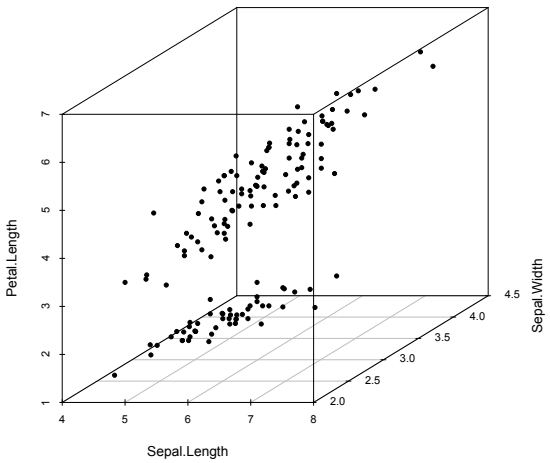

Figura 10.2. Diagrama de dispersión tridimensional de las tres primeras columnas de la tabla `iris`.

Una representación gráfica muy popular de las tablas de datos de tres o más columnas numéricas son las matrices formadas por los diagramas de dispersión de todos sus pares de columnas.

Si la tabla de datos es un *data frame*, esta matriz de diagramas de dispersión se obtiene simplemente aplicando la función `plot` al *data frame*; por ejemplo,

```
> plot(iris[ , 1:4])
```

produce el gráfico de la izquierda de la Figura 10.3. En este gráfico, los cuadrados en la diagonal indican a qué variables corresponden cada fila y cada columna, de manera que podamos identificar fácilmente qué variables compara cada diagrama de dispersión; así, en el diagrama de la primera fila y segunda columna de esta figura, las abscisas corresponden a anchuras de sépalos y las ordenadas a longitudes de sépalos. Observad que la nube de puntos no muestra una tendencia clara y en todo caso ligeramente negativa, lo que se corresponde con la correlación entre estas variables de -0.11 que hemos obtenido en el Ejemplo 10.5.

Podemos usar los parámetros usuales de `plot` para mejorar el gráfico resultante; por ejemplo, podemos usar colores para distinguir las flores según su especie:

```
> plot(iris[ , 1:4], col=iris$Species)
```

produce el gráfico de la derecha de la Figura 10.3.

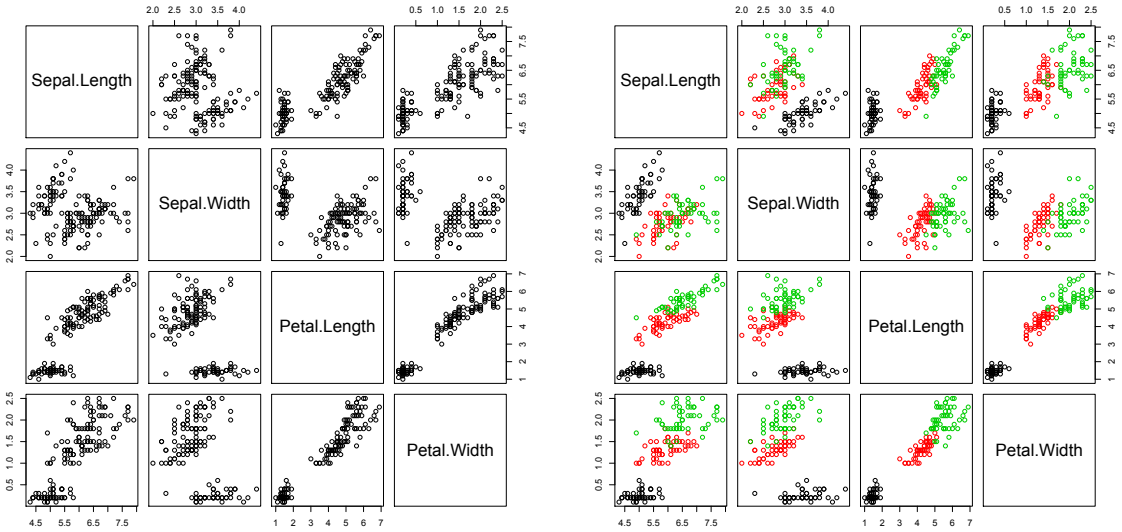

Figura 10.3. Matrices de diagramas de dispersión de la tabla `iris`; en la de la derecha, las especies se distinguen por medio de colores.

La matriz de diagramas de dispersión de una tabla de datos multidimensional también se puede calcular con la función `pairs`: así, `pairs(iris[, 1:4])` produce exactamente el mismo gráfico que `plot(iris[, 1:4])`. La ventaja de `pairs` es que se puede aplicar a una matriz para obtener la matriz de diagramas de dispersión de sus columnas, mientras que `plot` no.

El paquete `car` incorpora la función `spm` que genera matrices de diagramas de dispersión enriquecidos con información descriptiva de las variables de la tabla de datos. Por ejemplo,

```
> #Instalamos y cargamos el paquete car
> ...
> spm(iris[ , 1:4])
```

produce el gráfico de la izquierda de la Figura 10.4. Observaréis para empezar que en los cuadrados de la diagonal ha dibujado unas curvas: se trata de la curva de densidad de la variable correspondiente de la que ya hablábamos en la Sección 11.4. La información gráfica contenida en estos cuadrados de la diagonal se puede modificar con el parámetro `diagonal`: podemos pedir, por ejemplo, que dibuje un histograma de cada variable (con `diagonal="histogram"`) o su *boxplot* (con `diagonal="boxplot"`). Así,

```
> spm(iris[ , 1:4], diagonal="boxplot")
```

produce el gráfico de la derecha de la Figura 10.4.

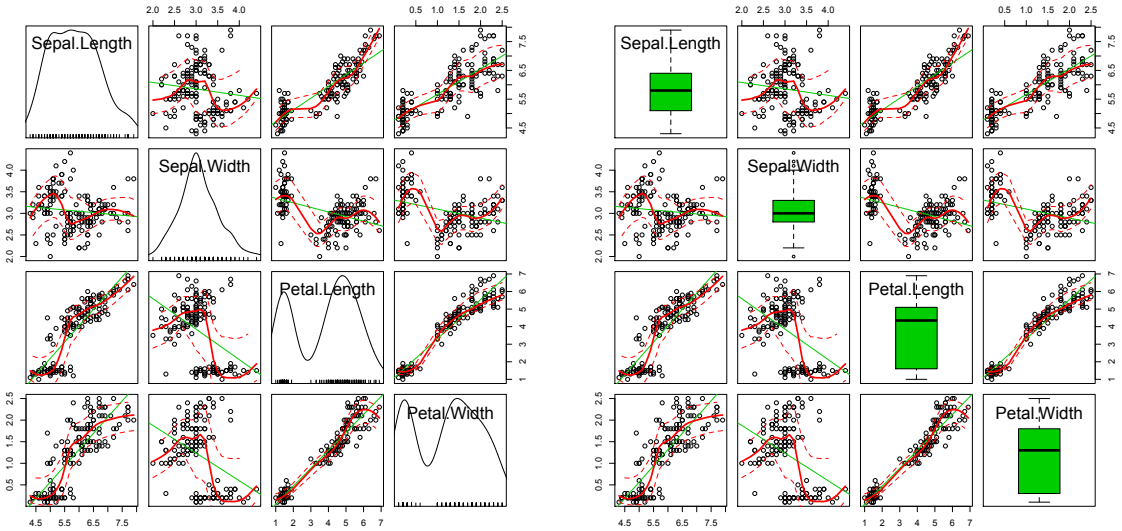

Figura 10.4. Matrices de diagramas de dispersión de la tabla `iris` producidos con `spm`.

Asimismo, observaréis que los diagramas de dispersión de la matriz producida con `spm` contienen curvas. La línea recta verde es la recta de regresión por mínimos cuadrados y, sin entrar en detalle sobre su significado exacto, las curvas rojas continuas representan la tendencia de los datos.

10.4. Guía rápida

- `sapply(`*data frame*`,`*función*`)` aplica la *función* a las columnas de un *data frame*.
- `scale` sirve para aplicar una transformación lineal a una matriz o a un *data frame*. Sus parámetros son:
  - `center`: especifica el vector que restamos a sus columnas.
  - `scale`: especifica el vector por el que a continuación dividimos sus columnas.
- `cov`, aplicada a dos vectores, calcula su covarianza muestral; aplicada a un *data frame* o a una matriz, calcula su matriz de covarianzas muestrales. Dispone del parámetro `use`:
  - Igualado a `"pairwise.complete.obs"`, calcula la covarianza de cada par de columnas teniendo en cuenta sólo sus observacions completas (las filas en las que ninguna de las dos tiene un `NA`), independientemente del resto de la tabla.

    - Igualado a `"complete.obs"`, calcula las covarianzas de las columnas teniendo en cuenta sólo las filas completas de toda la matriz.

- `cor`, aplicada a dos vectores, calcula su correlación; aplicada a un *data frame* o a una matriz, calcula su matriz de correlaciones. Dispone del mismo parámetro `use` que `cov`.
- `cov2cor`, aplicada a la matriz de covarianzas, calcula la matriz de correlaciones.
- `upper.tri`, aplicada a una matriz cuadrada M, produce la matriz triangular superior de valores lógicos del mismo orden que M. Con el parámetro `diag=TRUE` se impone que el triángulo de valores `TRUE` incluya la diagonal principal.
- `lower.tri`, aplicada a una matriz cuadrada M, produce la matriz triangular inferior de valores lógicos del mismo orden que M. Dispone del mismo parámetro `diag=TRUE`.
- `order` ordena el primer vector al que se aplica, desempatando empates mediante el orden de los vectores subsiguientes a los que se aplica; el parámetro `decreasing=TRUE` sirve para especificar que sea en orden decreciente.
- `plot`, aplicado a un *data frame* de dos variables numéricas, dibuja su diagrama de dispersión; aplicado a un *data frame* de más de dos variables numéricas, produce la matriz formada por los diagramas de dispersión de todos sus pares de variables.
- `pairs` es equivalente a `plot` en el sentido anterior, y se puede aplicar a matrices.
- `scatterplot3d`, del paquete del mismo nombre, dibuja diagramas de dispersión tridimensionales.

10.5. Ejercicio

El fichero `http://aprender.uib.es/Rdir/NotasMatesI14.txt` recoge las notas medias (sobre 100) obtenidas, en las diferentes actividades de evaluación de la asignatura Matemáticas I del grado de Biología en el curso 2013/14, por parte de los estudiantes que fueron considerados «presentados» en la primera convocatoria. Estas actividades consistieron en:

- Dos controles (columnas `Control1` y `Control2`).
- Talleres de resolución de problemas (columna `Talleres`).
- Ejercicios para resolver en casa (columna `Casa`).
- Cuestionarios en línea sobre los contenidos de la asignatura y sobre `R` (columnas `TestsCont` y `TestsR`, respectivamente).

Cargad este fichero en un *data frame*.

(a) Calculad el vector de medias y el vector de desviaciones típicas de esta tabla de datos. ¿Cuáles son las actividades de evaluación cuyas notas presentan mayor y menor variabilidad?

(b) Calculad las matrices de covarianzas y de correlaciones de esta tabla de datos.

(c) ¿Qué variable tiene la mayor correlación media con las otras variables? ¿Cuál tiene la menor?

(d) Ordenad los pares de variables de esta tabla por su correlación. ¿Cuáles son los dos pares con mayor correlación positiva? ¿Cuáles son los dos pares con menor correlación negativa?

(e) Comprobad en esta tabla de datos que su matriz de correlaciones es igual a la matriz de covarianzas de su tabla tipificada.

(f) Dibujad una matriz de diagramas de dispersión de estas notas. ¿Se pueden ver en este diagrama los dos pares de actividades de evaluación con mayor correlación y los dos pares con menor correlación que habéis encontrado en el apartado (d)?

Universitat de les Illes Balears

Campus Extens
UIB Virtual

http://**campusextens**.uib.cat


@campusextensUIB


http://www.scoop.it/t/recursos-i-eines-per-al-professorat


http://campusextensrecursos.uib.es/