Minería de datos

Sesión 1: Análisis exploratorio de datos

Paloma Botella Rocamora
Paloma.Botella@gmail.com

Estructura de la sesión.

Estructura de la sesión.

- 1. Datos Multivariantes
- 2. Análisis descriptivo numérico
- 3. Representaciones gráficas
- 4. Outliers
- 5. Datos faltantes
- 6. Escala y medidas de disimilaridad
- 7. Breve guía de preparación de los datos

1. Datos Multivariantes

Datos Multivariantes

Cuando **en cada elemento** de la población **se mide** un **conjunto de variables estadísticas**, diremos que se ha definido una variable estadística **multivariante**, **vectorial** o **multidimensional**.

Consideraremos que las *variables* definidas sobre cada elemento de la población *son numéricas* (las *variables cualitativas se transformarán a numéricas*: a una o varias variables binarias 0-1)

- Los datos multivariantes se encontrarán en una matriz de datos $\mathbf{X}$.
- Cada **fila** representará los valores de todas las variables medidas en **un individuo** (consideraremos n individuos).
- Cada **columna** representará los valores de **una variable** medida en todos los individuos (consideraremos k variables).
- El **elemento** (i, j) de la matriz de datos X representará el valor de la **variable** j medido en el **individuo** i.

- Así, la matrix de datos X tiene dimensión $n \times k$ y será de la forma:

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \cdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nk} \end{pmatrix}$$

- Podemos representar la matriz X de diferentes formas:

por *columnas* (o **variables**):

$$X = \begin{pmatrix} \mathbf{x_1} & \mathbf{x_2} & \cdots & \mathbf{x_k} \end{pmatrix}$$

por *filas* (o **individuos**):

$$X = \begin{pmatrix} \mathbf{x_1'} \\ \mathbf{x_2'} \\ \vdots \\ \mathbf{x_n'} \end{pmatrix}$$

Algunos ejemplos: **medifis**

Datos **medifis**: Ocho variables físicas tomadas a 27 estudiantes:

- **sex**: Sexo (0=Mujer, 1=Hombre)
- **est**: Estatura (en cm)
- **pes**: Peso (en kg)
- **lpie**: Longitud del pie (en cm)
- **lbr**: Longitud del brazo (en cm)
- **aes**: Anchura de la espalda (en cm)
- **dcr**: Diámetro del cráneo (en cm)
- **lrt**: Longitud entre la rodilla y el tobillo (en cm)

Datos del libro *Análisis de datos multivariante* de Daniel Peña (2002).

Algunos ejemplos: **medifis**

```
kable(medifis, row.names = FALSE)
```

sex	est	pes	pie	lbr	aes	dcr	lrt
0	159	49	36.0	68.0	42.0	57.0	40.0
1	164	62	39.0	73.0	44.0	55.0	44.0
0	172	65	38.0	75.0	48.0	58.0	44.0
0	167	52	37.0	73.0	41.5	58.0	44.0
0	164	51	36.0	71.0	44.5	54.0	40.0
0	161	67	38.0	71.0	44.0	56.0	42.0
0	168	48	39.0	72.5	41.0	54.5	43.0
1	181	74	43.0	74.0	50.0	60.0	47.0
1	183	74	41.0	79.0	47.5	59.5	47.0
0	158	50	36.0	68.5	44.0	57.0	41.0
0	156	65	36.0	68.0	46.0	58.0	41.0
1	173	64	40.0	79.0	48.0	56.5	47.0
0	158	43	36.0	68.0	43.0	55.0	39.0
1	178	74	42.0	75.0	50.0	59.0	45.0
1	181	76	43.0	83.0	51.0	57.0	43.0
1	182	91	41.0	83.0	53.0	59.0	43.0
1	176	73	42.0	78.0	48.0	58.0	45.0
0	162	68	39.0	72.0	44.0	59.0	42.0
0	156	52	36.0	67.0	36.0	56.0	41.0
0	152	45	34.0	66.0	40.0	55.0	38.0
1	181	80	43.0	76.0	49.0	57.0	46.0
1	173	69	41.0	74.0	48.0	56.0	44.0

Algunos ejemplos: **airquality**

Datos **airquality**: New York Air Quality Measuraments

Seis variables numéricas de calidad de aire medidas diariamente en Nueva York (de mayo a septiembre de 1973)

- **Ozone**
- **Solar.R**
- **Wind**
- **Temp**
- **Month**
- **Day**

Algunos ejemplos: **airquality**

```
kable(airquality, row.names = FALSE)
```

Ozone	Solar.R	Wind	Temp	Month	Day
41	190	7.4	67	5	1
36	118	8.0	72	5	2
12	149	12.6	74	5	3
18	313	11.5	62	5	4
NA	NA	14.3	56	5	5
28	NA	14.9	66	5	6
23	299	8.6	65	5	7
19	99	13.8	59	5	8
8	19	20.1	61	5	9
NA	194	8.6	69	5	10
7	NA	6.9	74	5	11
16	256	9.7	69	5	12
11	290	9.2	66	5	13
14	274	10.9	68	5	14
18	65	13.2	58	5	15
14	334	11.5	64	5	16
34	307	12.0	66	5	17
6	78	18.4	57	5	18
30	322	11.5	68	5	19
11	44	9.7	62	5	20
1	8	9.7	59	5	21
11	320	16.6	73	5	22

2. Análisis descriptivo numérico

2.1 Análisis descriptivo numérico UNIVARIANTE

Univariante: vector de medias

Vector de medias: vector de dimensión k que recoge la **media** de cada una de las k variables.

$$\bar{\mathbf{x}} = \begin{pmatrix} \bar{\mathbf{x}}_1 \\ \bar{\mathbf{x}}_2 \\ \vdots \\ \bar{\mathbf{x}}_k \end{pmatrix} = \frac{\mathbf{1}}{\mathbf{n}} (\mathbf{x}_1 \mathbf{x}_2 \cdots \mathbf{x}_k)' \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = (\mathbf{1}_\mathbf{n}' \mathbf{1}_\mathbf{n})^{-1} \mathbf{X}' \mathbf{1}_\mathbf{n}$$

En *R* se puede obtener con el comando:

```
apply(x, 2, mean)
```

Univariante: vector de varianzas/desviaciones típicas

Vector de varianzas/desviaciones típicas: vector de dimensión k que recoge la **varianza** (**desviación típica**) de cada una de las k variables.

$$\mathbf{v}(\mathbf{X}) = \begin{pmatrix} var(\mathbf{x_1}) \\ var(\mathbf{x_2}) \\ \vdots \\ var(\mathbf{x_k}) \end{pmatrix}; \mathbf{sd}(\mathbf{X}) = \begin{pmatrix} sd(\mathbf{x_1}) \\ sd(\mathbf{x_2}) \\ \vdots \\ sd(\mathbf{x_k}) \end{pmatrix}$$

En R se puede obtener con el comando:

```
apply(x, 2, var)
apply(x, 2, sd)
```

Univariante: vector de coeficientes de asimetría

Vector de coeficientes de asimetría: vector de dimensión k que recoge el **coeficiente de asimetría** de cada una de las k variables.

- **Coef.Asimetría = 0** (distribución simétrica)
- **Coef.Asimetría > 0** (distribución asimétrica; mayor concentración de valores a la derecha de la media que a la izquierda)
- **Coef.Asimetría < 0** (distribución asimétrica; mayor concentración de valores a la izquierda de la media que a la derecha)

En *R* se puede obtener, por ejemplo, con la siguiente función de la librería *e1071*:

```
library(e1071)
apply(x, 2, skewness)
```

Univariante: vector de coeficientes de kurtosis

Vector de coeficientes de kurtosis: vector de dimensión k que recoge el **coeficiente de kurtosis** de cada una de las k variables.

- **Coef.Kurtosis = 0** (Igual apuntamiento que la Distrib.Normal)
- **Coef.Kurtosis > 0** (Mayor apuntamiento que la Distrib.Normal)
- **Coef.Kurtosis < 0** (Menor apuntamiento que la Distrib.Normal)

En *R* se puede obtener, por ejemplo, con la siguiente función de la librería *e1071* (o en otras como *fBasics*):

```
library(e1071)
apply(x, 2, kurtosis)
```

Univariante: Ejemplo

```
library(e1071)
# Media, Desv.Típica, Asimetría y Curtosis
round(apply(medifis, 2, mean), 2)
```

```
##    sex    est    pes    pie    lbr    aes    dcr    lrt
##   0.44 168.78  63.89  38.98  73.46  45.85  57.24  43.09
```

```
round(apply(medifis, 2, sd), 2)
```

```
##   sex   est   pes   pie   lbr   aes   dcr   lrt
##  0.51 10.20 12.80  2.86  4.96  4.02  1.84  3.16
```

```
round(apply(medifis, 2, skewness), 2)
```

```
##   sex   est   pes   pie   lbr   aes   dcr   lrt
##  0.21  0.15  0.17  0.27  0.37 -0.22  0.16  0.56
```

```
round(apply(medifis, 2, kurtosis), 2)
```

```
##   sex   est   pes   pie   lbr   aes   dcr   lrt
## -2.03 -1.19 -0.92 -1.07 -0.89 -0.36 -0.98  0.38
```

Univariante: Algunas funciones que realizan descriptivos...

```
library(Hmisc)
Hmisc::describe(medifis)
```

```
## medifis
##
##  8  Variables      27  Observations
## ---------------------------------------------------------------------------
## sex
##        n  missing distinct     Info      Sum     Mean      Gmd
##       27        0        2    0.742       12   0.4444   0.5128
##
## ---------------------------------------------------------------------------
## est
##        n  missing distinct     Info     Mean      Gmd      .05
##       27        0       19    0.997    168.8    11.91    155.3
##      .10      .25      .50      .75      .90      .95
##    156.0    160.0    168.0    177.0    181.4    182.7
##
## lowest : 152 155 156 158 159, highest: 178 181 182 183 189
##
## 152 (1, 0.037), 155 (1, 0.037), 156 (2, 0.074), 158 (2, 0.074), 159
## (1, 0.037), 161 (1, 0.037), 162 (1, 0.037), 164 (2, 0.074), 167 (1,
## 0.037), 168 (2, 0.074), 170 (2, 0.074), 172 (1, 0.037), 173 (2,
## 0.074), 176 (1, 0.037), 178 (1, 0.037), 181 (3, 0.111), 182 (1,
## 0.037), 183 (1, 0.037), 189 (1, 0.037)
## ---------------------------------------------------------------------------
## pes
##        n  missing distinct     Info     Mean      Gmd      .05
##       27        0       22    0.998    63.89    14.77     45.9
##      .10      .25      .50      .75      .90      .95
##     48.6     52.0     65.0     73.5     77.6     84.9
##
## lowest : 43 45 48 49 50, highest: 74 76 80 87 91
## ---------------------------------------------------------------------------
## pie
```

Univariante: Algunas funciones que realizan descriptivos...

```
library(pastecs)
stat.desc(medifis)
```

```
##                      sex          est          pes          pie
## nbr.val       27.00000000 2.700000e+01   27.0000000 2.700000e+01
## nbr.null      15.00000000 0.000000e+00    0.0000000 0.000000e+00
## nbr.na         0.00000000 0.000000e+00    0.0000000 0.000000e+00
## min            0.00000000 1.520000e+02   43.0000000 3.400000e+01
## max            1.00000000 1.890000e+02   91.0000000 4.500000e+01
## range          1.00000000 3.700000e+01   48.0000000 1.100000e+01
## sum           12.00000000 4.557000e+03 1725.0000000 1.052500e+03
## median         0.00000000 1.680000e+02   65.0000000 3.900000e+01
## mean           0.44444444 1.687778e+02   63.8888889 3.898148e+01
## SE.mean        0.09745089 1.962130e+00    2.4636002 5.511458e-01
## CI.mean.0.95   0.20031318 4.033215e+00    5.0640026 1.132897e+00
## var            0.25641026 1.039487e+02  163.8717949 8.201567e+00
## std.dev        0.50636968 1.019552e+01   12.8012419 2.863838e+00
## coef.var       1.13933179 6.040798e-02    0.2003673 7.346662e-02
##                      lbr          aes          dcr          lrt
## nbr.val       2.700000e+01 2.700000e+01 2.700000e+01 2.700000e+01
## nbr.null      0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## nbr.na        0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## min           6.600000e+01 3.600000e+01 5.400000e+01 3.800000e+01
## max           8.300000e+01 5.300000e+01 6.100000e+01 5.200000e+01
## range         1.700000e+01 1.700000e+01 7.000000e+00 1.400000e+01
## sum           1.983500e+03 1.238000e+03 1.545500e+03 1.163500e+03
## median        7.300000e+01 4.600000e+01 5.700000e+01 4.300000e+01
## mean          7.346296e+01 4.585185e+01 5.724074e+01 4.309259e+01
## SE.mean       9.540459e-01 7.738675e-01 3.544290e-01 6.074309e-01
## CI.mean.0.95  1.961070e+00 1.590707e+00 7.285393e-01 1.248592e+00
## var           2.457550e+01 1.616952e+01 3.391738e+00 9.962251e+00
## std.dev       4.957368e+00 4.021134e+00 1.841667e+00 3.156303e+00
## coef.var      6.748119e-02 8.769839e-02 3.217406e-02 7.324468e-02
```

Univariante: Algunas funciones que realizan descriptivos...

```
library(psych)
psych::describe(medifis)
```

```
##     vars  n   mean    sd median trimmed   mad min max range  skew
## sex    1 27   0.44  0.51      0    0.43  0.00   0   1     1  0.21
## est    2 27 168.78 10.20    168  168.61 13.34 152 189    37  0.15
## pes    3 27  63.89 12.80     65   63.43 13.34  43  91    48  0.17
## pie    4 27  38.98  2.86     39   38.89  4.45  34  45    11  0.27
## lbr    5 27  73.46  4.96     73   73.24  5.93  66  83    17  0.37
## aes    6 27  45.85  4.02     46   45.91  2.97  36  53    17 -0.22
## dcr    7 27  57.24  1.84     57   57.22  1.48  54  61     7  0.16
## lrt    8 27  43.09  3.16     43   42.98  2.97  38  52    14  0.56
##     kurtosis   se
## sex    -2.03 0.10
## est    -1.19 1.96
## pes    -0.92 2.46
## pie    -1.07 0.55
## lbr    -0.89 0.95
## aes    -0.36 0.77
## dcr    -0.98 0.35
## lrt     0.38 0.61
```

Univariante: Algunas funciones que realizan descriptivos...

```
psych::describeBy(medifis, group = "sex")
```

```
##
##  Descriptive statistics by group
## sex: 0
##     vars  n   mean   sd median trimmed  mad min max range  skew
## sex    1 15   0.00 0.00    0.0    0.00 0.00   0   0     0   NaN
## est    2 15 161.73 6.13  161.0  161.69 7.41 152 172    20  0.16
## pes    3 15  55.60 8.97   52.0   55.46 5.93  43  70    27  0.35
## pie    4 15  36.83 1.38   36.0   36.88 1.48  34  39     5 -0.03
## lbr    5 15  70.03 2.72   70.5   69.96 3.71  66  75     9  0.16
## aes    6 15  43.33 3.07   44.0   43.54 2.97  36  48    12 -0.52
## dcr    7 15  56.63 1.72   56.0   56.58 1.48  54  60     6  0.28
## lrt    8 15  41.07 1.94   41.0   41.08 1.48  38  44     6 -0.03
##     kurtosis   se
## sex      NaN 0.00
## est    -1.43 1.58
## pes    -1.51 2.32
## pie    -0.86 0.36
## lbr    -1.41 0.70
## aes     0.06 0.79
## dcr    -1.05 0.44
## lrt    -1.24 0.50
## ------------------------------------------------------
## sex: 1
##     vars  n   mean   sd median trimmed  mad min max range  skew
## sex    1 12   1.00 0.00    1.0    1.00 0.00   1   1     0   NaN
## est    2 12 177.58 6.75  179.5  177.80 5.19 164 189    25 -0.35
## pes    3 12  74.25 8.61   74.0   73.80 8.15  62  91    29  0.46
## pie    4 12  41.67 1.67   41.5   41.60 2.22  39  45     6  0.27
## lbr    5 12  77.75 3.55   77.5   77.70 4.45  73  83    10  0.24
## aes    6 12  49.00 2.60   48.5   49.10 2.22  44  53     9 -0.04
## dcr    7 12  58.00 1.77   58.0   58.00 1.85  55  61     6  0.00
## lrt    8 12  45.62 2.48   45.0   45.25 2.22  43  52     9  1.21
##     kurtosis   se
```

2.2 Análisis descriptivo numérico MULTIVARIANTE

Multivariante: matriz de varianzas-covarianzas

Matriz de varianzas-covarianzas (S): se trata de una matriz simétrica de dimensiones $k \times k$ que recoge la información sobre las relaciones lineales entre las variables. Los elementos de la diagonal representan la varianza de cada una de las variables, mientras que el resto de elementos fuera de la diagonal representan la covarianza entre cada par de variables.

$$S = \begin{pmatrix} S_{11} & \cdots & S_{1k} \\ \vdots & \vdots & \vdots \\ S_{1k} & \cdots & S_{kk} \end{pmatrix}$$

Si consideramos $X_c = X - 1_n \bar{X}'$ (matriz de datos centrados):

$$S = \frac{1}{n-1} X_c' X_c$$

En *R* se puede obtener con la función *cov*:

```
cov(x)
```

Ejemplo

```
round(cov(medifis), 2)
```

```
##       sex    est    pes   pie   lbr   aes   dcr   lrt
## sex  0.26   4.06   4.78  1.24  1.98  1.45  0.35  1.17
## est  4.06 103.95 108.36 27.09 45.86 34.43 11.04 27.16
## pes  4.78 108.36 163.87 31.15 52.05 43.21 14.60 29.03
## pie  1.24  27.09  31.15  8.20 12.10  9.19  2.89  7.69
## lbr  1.98  45.86  52.05 12.10 24.58 15.98  4.34 11.97
## aes  1.45  34.43  43.21  9.19 15.98 16.17  4.64  8.00
## dcr  0.35  11.04  14.60  2.89  4.34  4.64  3.39  3.23
## lrt  1.17  27.16  29.03  7.69 11.97  8.00  3.23  9.96
```

Multivariante: matriz de correlaciones

Matriz de correlaciones (R): se trata de una matriz simétrica y cuadrada, de dimensiones $k \times k$, que recoge la relación lineal entre las variables. Los elementos de la diagonal toman el valor 1, mientras que el resto de elementos fuera de la diagonal representan la correlación lineal entre cada par de variables.

$$R = \begin{pmatrix} 1 & \cdots & R_{1k} \\ \vdots & 1 & \vdots \\ R_{1k} & \cdots & 1 \end{pmatrix}$$

donde $R_{ij} = R_{ji} = \frac{S_{ij}}{\sqrt{S_{ii} S_{jj}}}$

Si definimos la matriz diagonal

$$D = \begin{pmatrix} \sqrt{S_{11}} & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & \sqrt{S_{kk}} \end{pmatrix}$$

se cumple que:

- $R = D^{-1} S D^{-1}$
- $S = DRD$

En *R* se puede obtener con la función *cor*:

```
cor(x)
```

Ejemplo

```
round(cor(medifis), 2)
```

```
##      sex  est  pes  pie  lbr  aes  dcr  lrt
## sex 1.00 0.79 0.74 0.85 0.79 0.71 0.38 0.73
## est 0.79 1.00 0.83 0.93 0.91 0.84 0.59 0.84
## pes 0.74 0.83 1.00 0.85 0.82 0.84 0.62 0.72
## pie 0.85 0.93 0.85 1.00 0.85 0.80 0.55 0.85
## lbr 0.79 0.91 0.82 0.85 1.00 0.80 0.47 0.77
## aes 0.71 0.84 0.84 0.80 0.80 1.00 0.63 0.63
## dcr 0.38 0.59 0.62 0.55 0.47 0.63 1.00 0.56
## lrt 0.73 0.84 0.72 0.85 0.77 0.63 0.56 1.00
```

Si queremos obtener la significación (en forma de p-valor) de esos coeficientes de correlación podemos usar la función *rcorr* de la librería *Hmisc*. El resultado es una lista con 3 elementos: la matriz de correlaciones (r), el número de observaciones utilizadas para calcular cada coeficiente de correlación (n) y la matriz con los p-valores obtenidos para cada coeficiente (P).

```
rcorr(as.matrix(x))
```

Ejemplo

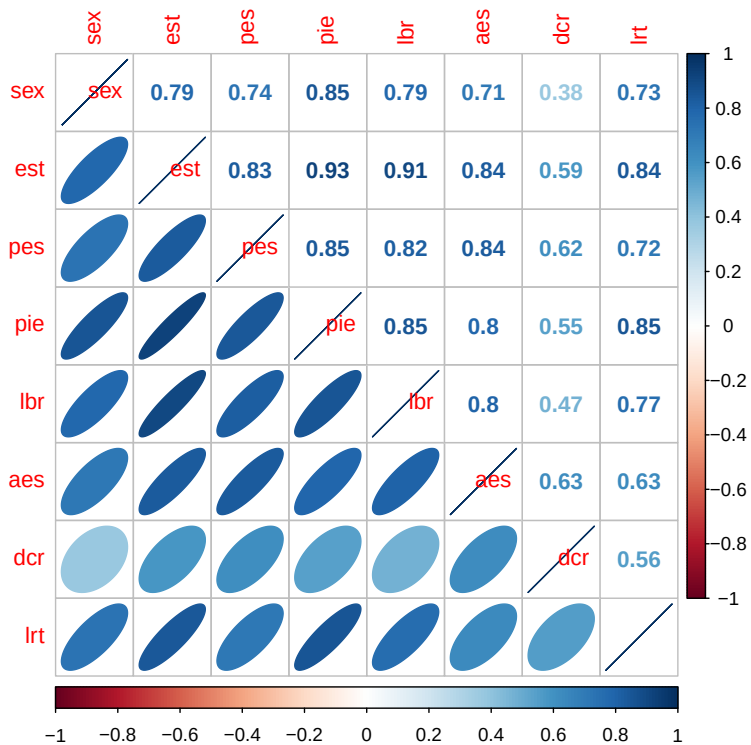
```
library(Hmisc)
mat.cor <- rcorr(as.matrix(medifis))
round(mat.cor$P, 4)
```

```
##        sex    est   pes    pie    lbr   aes    dcr    lrt
## sex     NA 0.0000 0e+00 0.0000 0.0000 0e+00 0.0534 0.0000
## est 0.0000     NA 0e+00 0.0000 0.0000 0e+00 0.0013 0.0000
## pes 0.0000 0.0000    NA 0.0000 0.0000 0e+00 0.0006 0.0000
## pie 0.0000 0.0000 0e+00     NA 0.0000 0e+00 0.0031 0.0000
## lbr 0.0000 0.0000 0e+00 0.0000     NA 0e+00 0.0123 0.0000
## aes 0.0000 0.0000 0e+00 0.0000 0.0000    NA 0.0005 0.0004
## dcr 0.0534 0.0013 6e-04 0.0031 0.0123 5e-04     NA 0.0027
## lrt 0.0000 0.0000 0e+00 0.0000 0.0000 4e-04 0.0027     NA
```

Para representar gráficamente matrices de correlación podéis explorar la función *corrplot* de la librería *corrplot*:

Ejemplo

```
library(corrplot)
matcor <- cor(medifis)
corrplot(matcor, method = "ellipse", type = "lower", diag = T)
corrplot(matcor, method = "number", type = "upper", diag = FALSE,
    add = T)
```



También podemos visualizar la matriz de gráficos de dispersión, los coeficientes de correlación y ver si son estadísticamente significativos mediante el comando *chart.Correlation* de la librería *PerformanceAnalytics*.

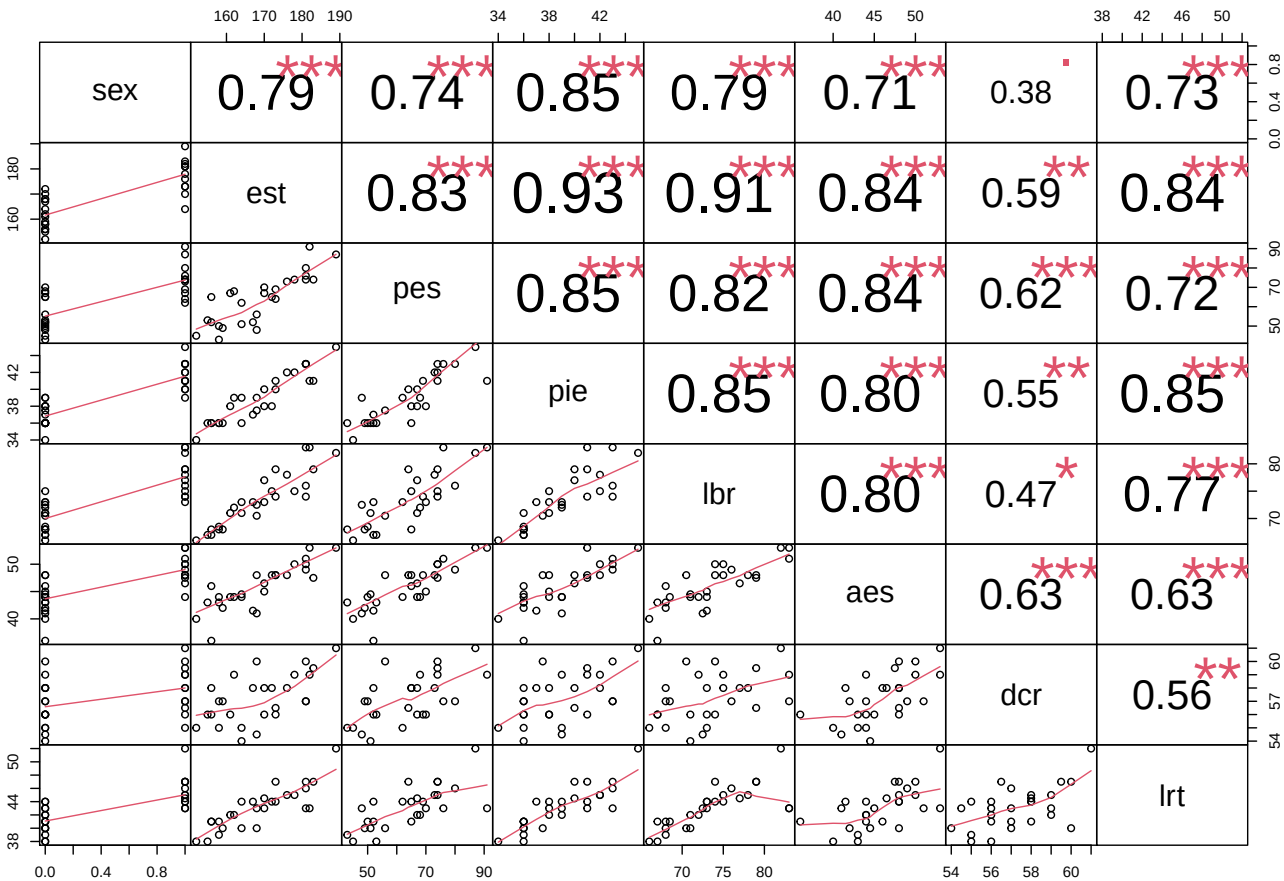
```
chart.Correlation(x, histogram = F)
```

Este gráfico mostrará diferentes símbolos (en color rojo) según el valor de p-valor:

- 0 - 0.001 : ***
- 0.001 - 0.01 : **
- 0.01 - 0.05 : *
- 0.05 - 0.1 : .

Ejemplo

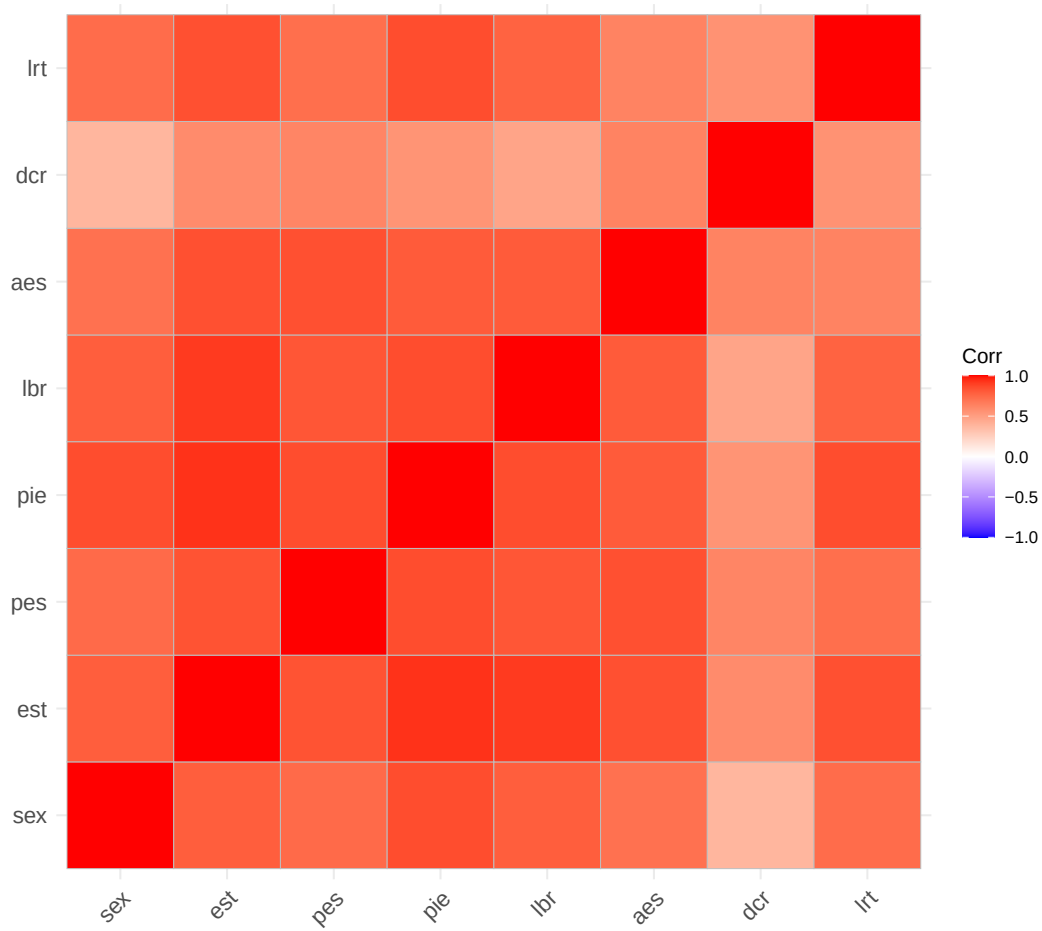
```
library(PerformanceAnalytics)
chart.Correlation(medifis, histogram = F)
```



En el entorno *ggplot2* también puedes explorar la función *ggcorrplot*:

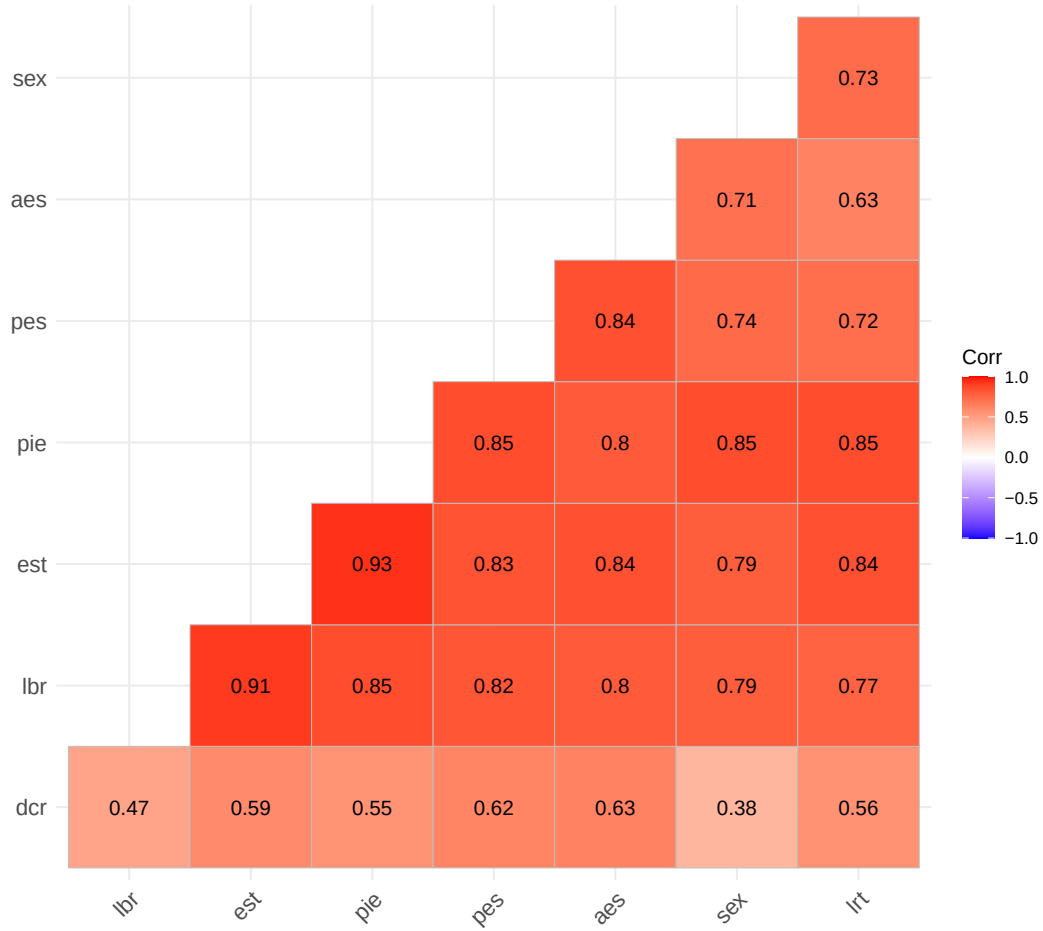
Ejemplo

```
library(ggplot2)
library(ggcorrplot)
matcor <- cor(medifis)
ggcorrplot(matcor)
```



Ejemplo

```
ggcorrplot(matcor, hc.order = TRUE, type = "lower", lab = "TRUE")
```



Distribuciones Multivariantes

De la misma forma que se definen distribuciones de probabilidad para variables aleatorias, discretas o continuas, que recogen la probabilidad de los distintos valores que puede tomar una variable, en el ámbito **multivariante** se definen distribuciones de probabilidad conjuntas para dos o más variables que recogen las probabilidades asociadas a las diferentes combinaciones de valores que pueden tomar todas estas variables simultáneamente.

La distribución más importante en el ámbito multivariante es la **Distribución Normal Multivariante**.

Distribución Normal Multivariante

Un vector aleatorio $\mathbf{X}$ sigue una distribución Normal k-variante o k-dimensional de medias

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_k \end{pmatrix}$$

y matriz de varianzas covarianzas

$$\Sigma = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1k} \\ \sigma_{21} & \cdots & \sigma_{2k} \\ \vdots & \ddots & \vdots \\ \sigma_{k1} & \cdots & \sigma_{kk} \end{pmatrix}$$

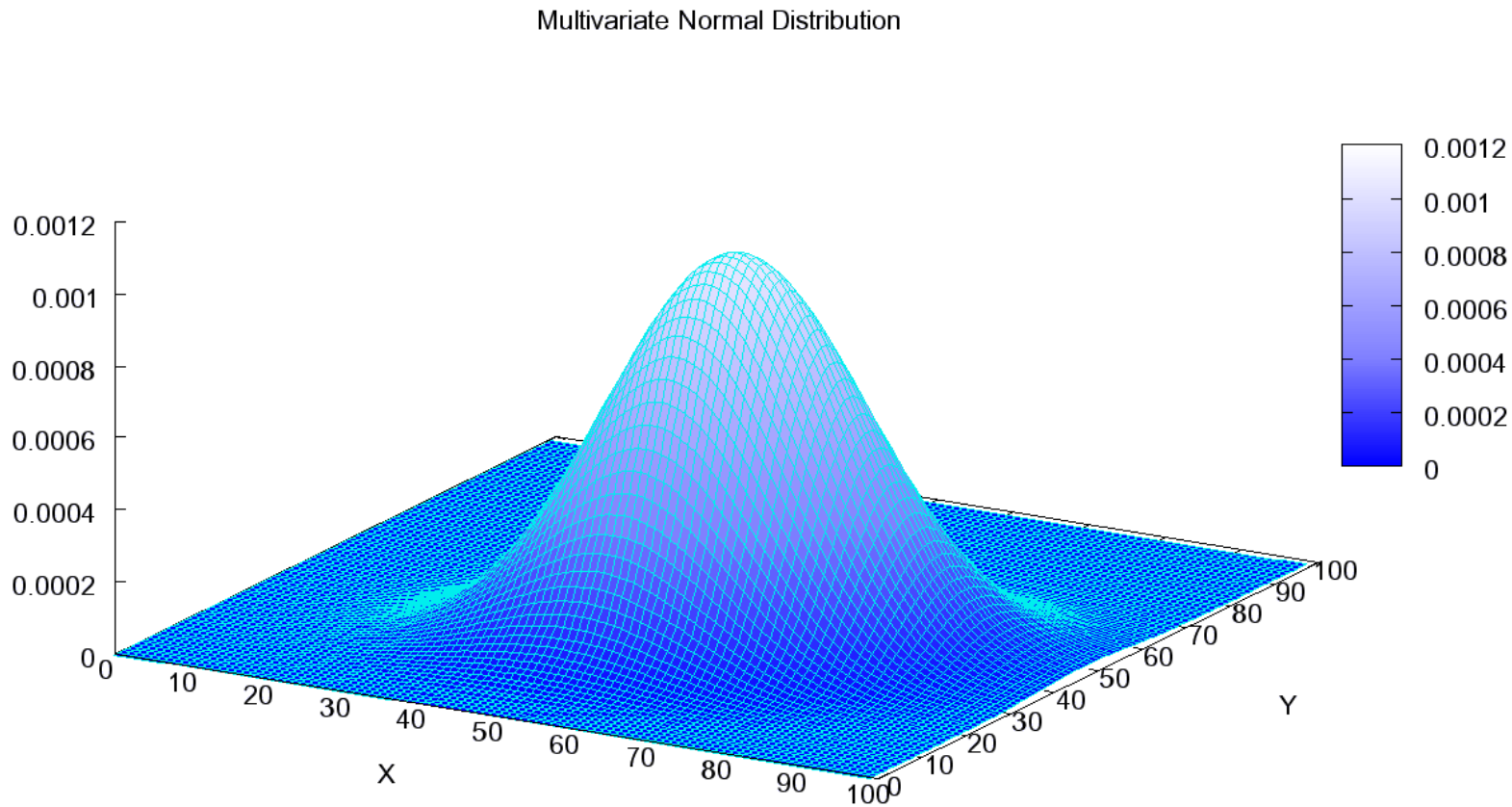
lo que se representa como

$$N_k(\mu, \Sigma)$$

si su función de densidad conjunta obedece a la expresión:

$$f(X) = f(x_1, x_2, ..., x_k) = \frac{e^{-\frac{1}{2}\left((X-\mu)'\Sigma^{-1}(X-\mu)\right)}}{\sqrt{(2\pi)^k|\Sigma|}}$$

Ejemplo de Distribución Normal Bivariante



3. Representaciones gráficas

Gráficos bi-dimensionales: gráficos de dispersión

Gráfico de dispersión o nubes de puntos: recoge la relación entre cada **par** de variables.

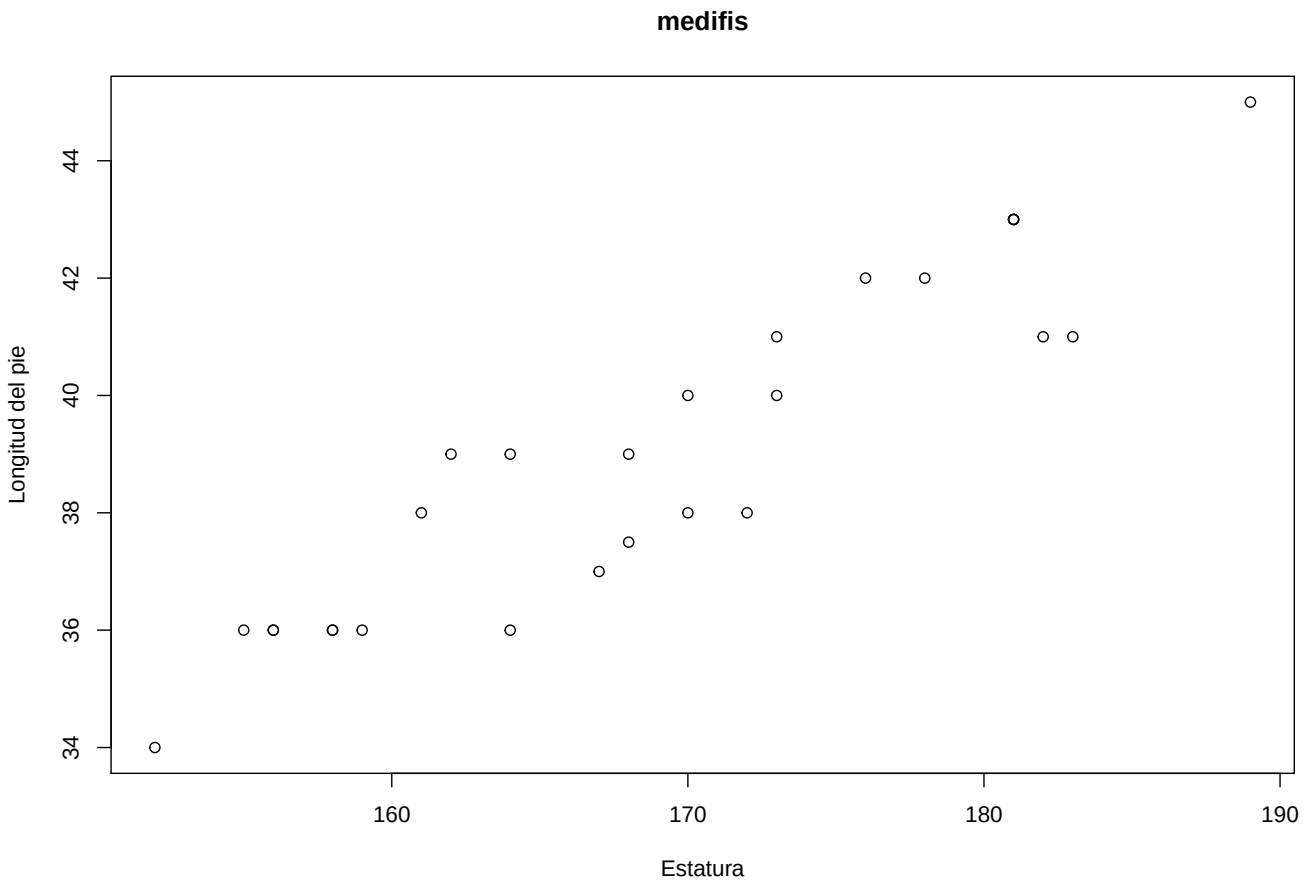
Se pueden considerar **variaciones** sobre estos gráficos:

- Añadir la **recta de regresión**.
- **Representar**, en lugar de un punto, los **nombres** de cada observación (si el banco de datos lo admite) para identificar los individuos.
- Con imaginación: se puede **añadir** a la representación **una variable más** mediante **gráficos de burbujas (3 variables en total)**.

Ejemplo:

Gráfico de dispersión

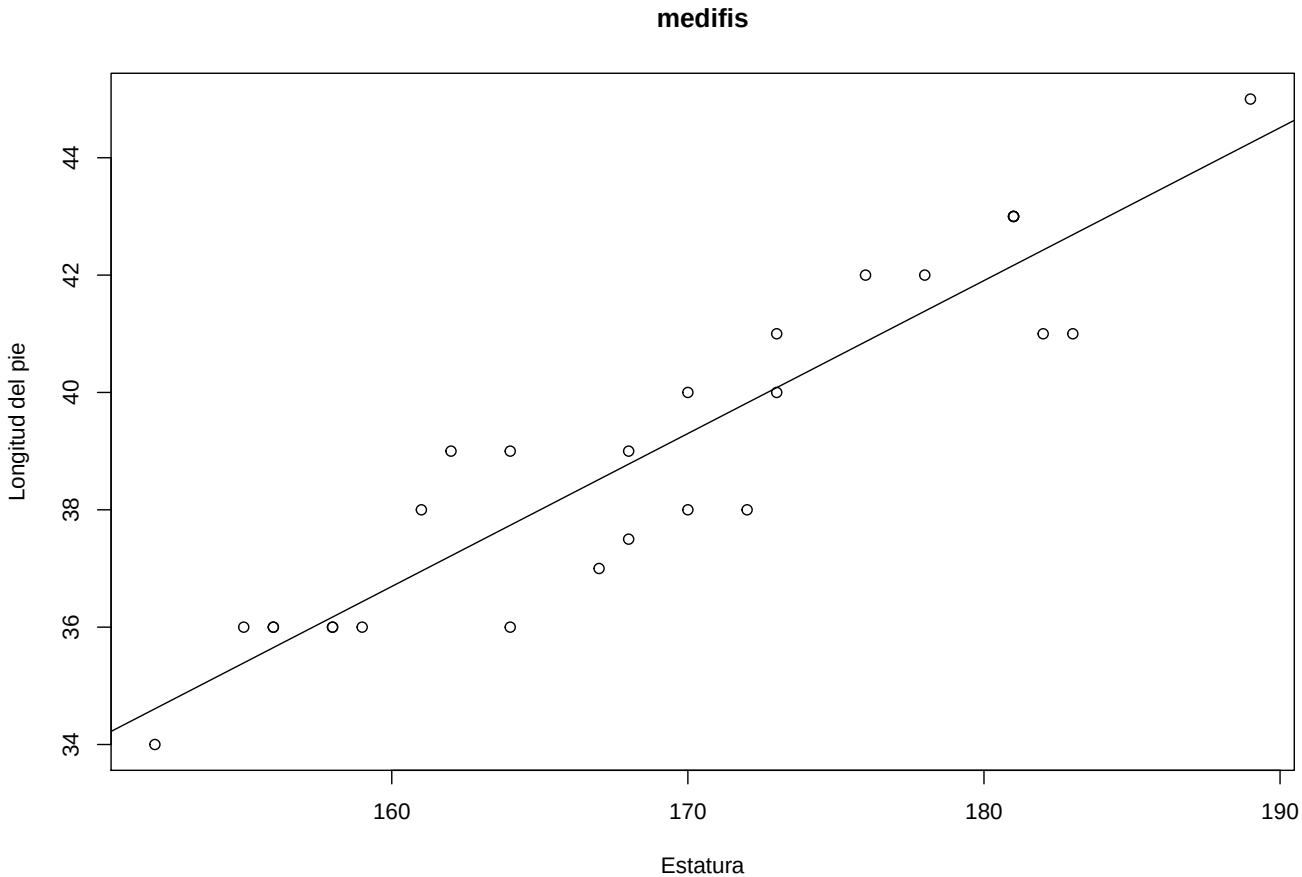
```
plot(medifis$est, medifis$pie, xlab = "Estatura", ylab = "Longitud del pie",
    main = "medifis")
```



Ejemplo:

Añadimos la **recta de regresión**

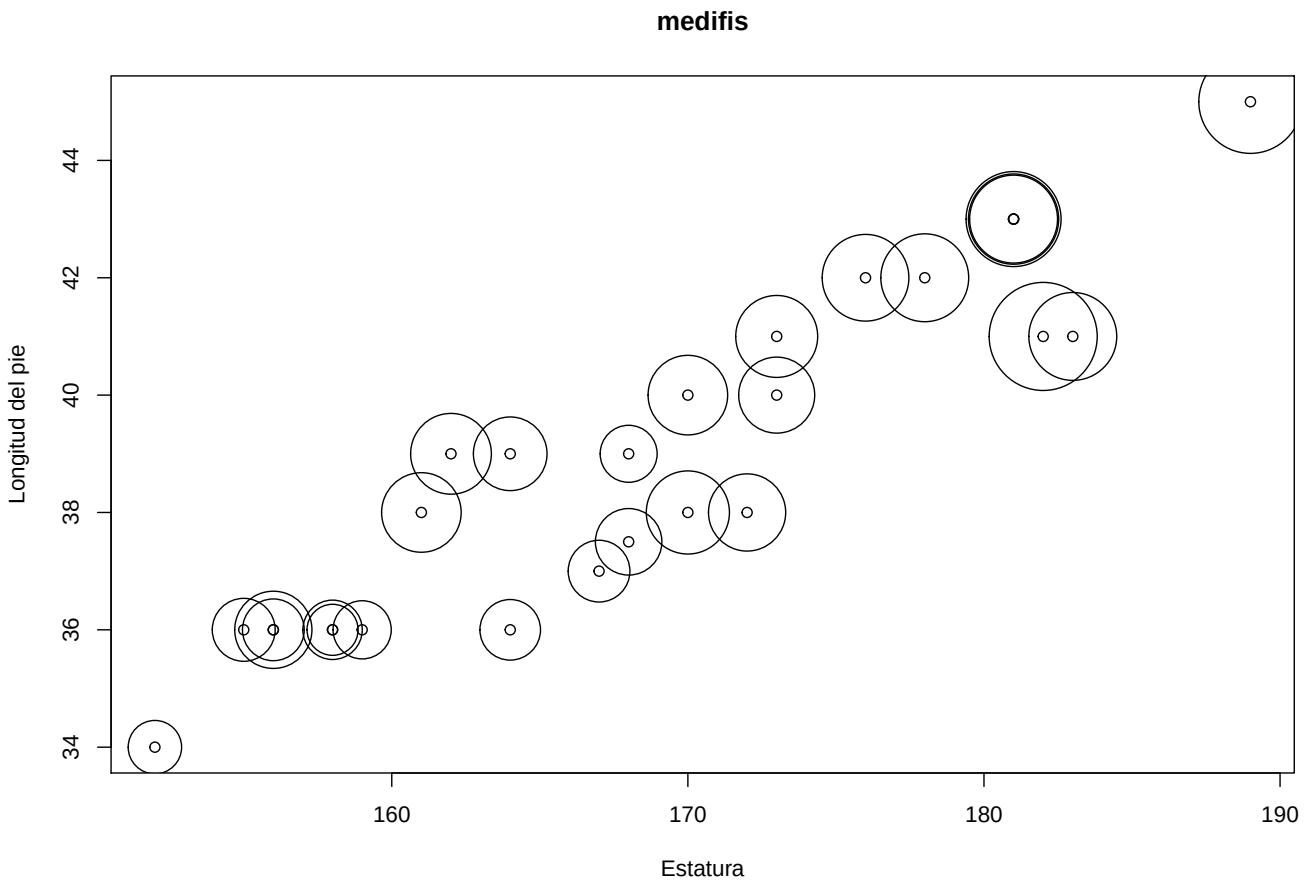
```
plot(medifis$est, medifis$pie, xlab = "Estatura", ylab = "Longitud del pie",
    main = "medifis")
abline(lm(medifis$pie ~ medifis$est))
```



Ejemplo:

Podemos añadir una variable más, *peso*, mediante **burbujas**

```
plot(medifis$est, medifis$pie, xlab = "Estatura", ylab = "Longitud del pie",
    main = "medifis")
symbols(medifis$est, medifis$pie, circles = medifis$pes, add = TRUE,
    inches = 0.4)
```



Gráficos bi-dimensionales: Matriz de dispersión

Matriz de gráficos de dispersión:matriz (gráfica) que recoge el **cruce de todas las variables dos a dos**.

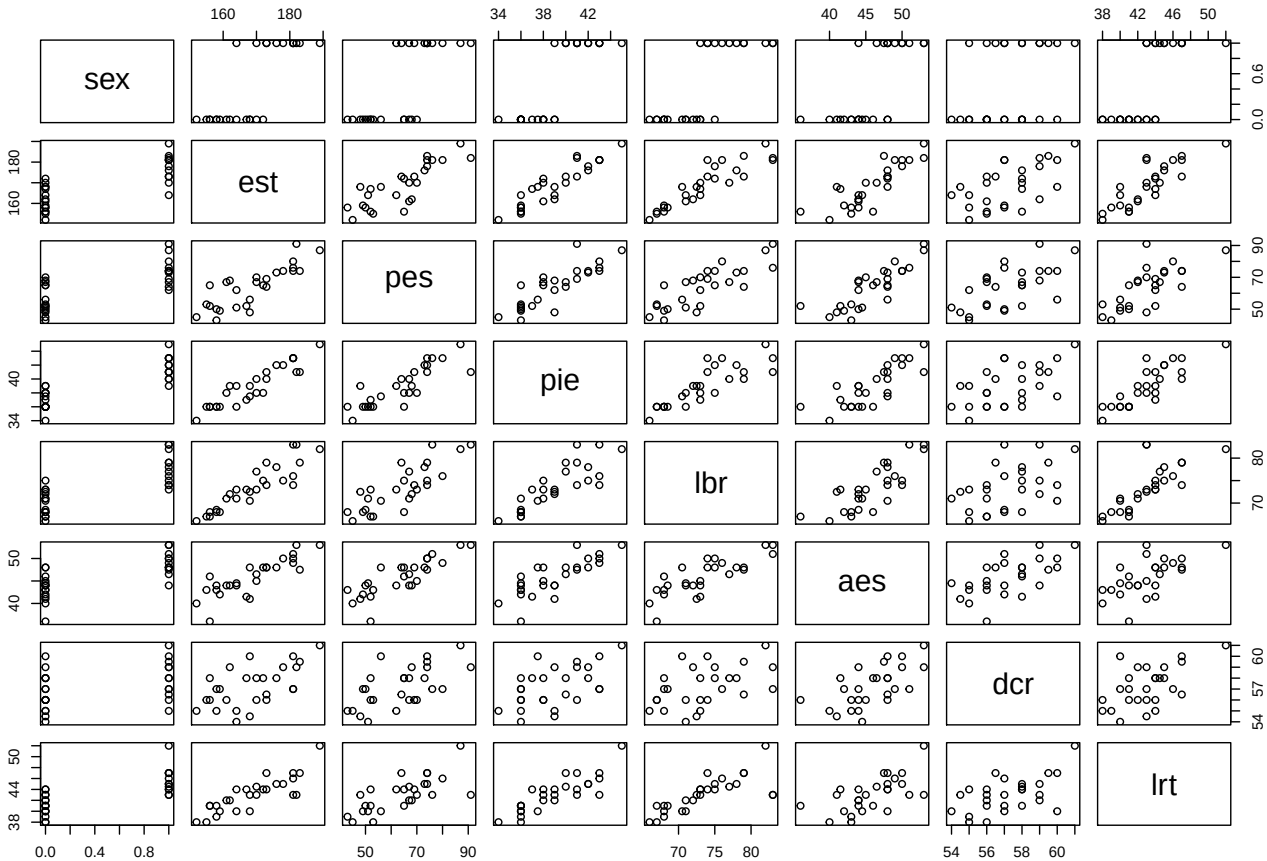
En *R* se puede obtener, entre otras, con las funciones:

```
pairs(x)


car::scatterplotMatrix(x)
```

Ejemplo:

Matriz de dispersión

```
pairs(medifis)
```



Ejemplo:

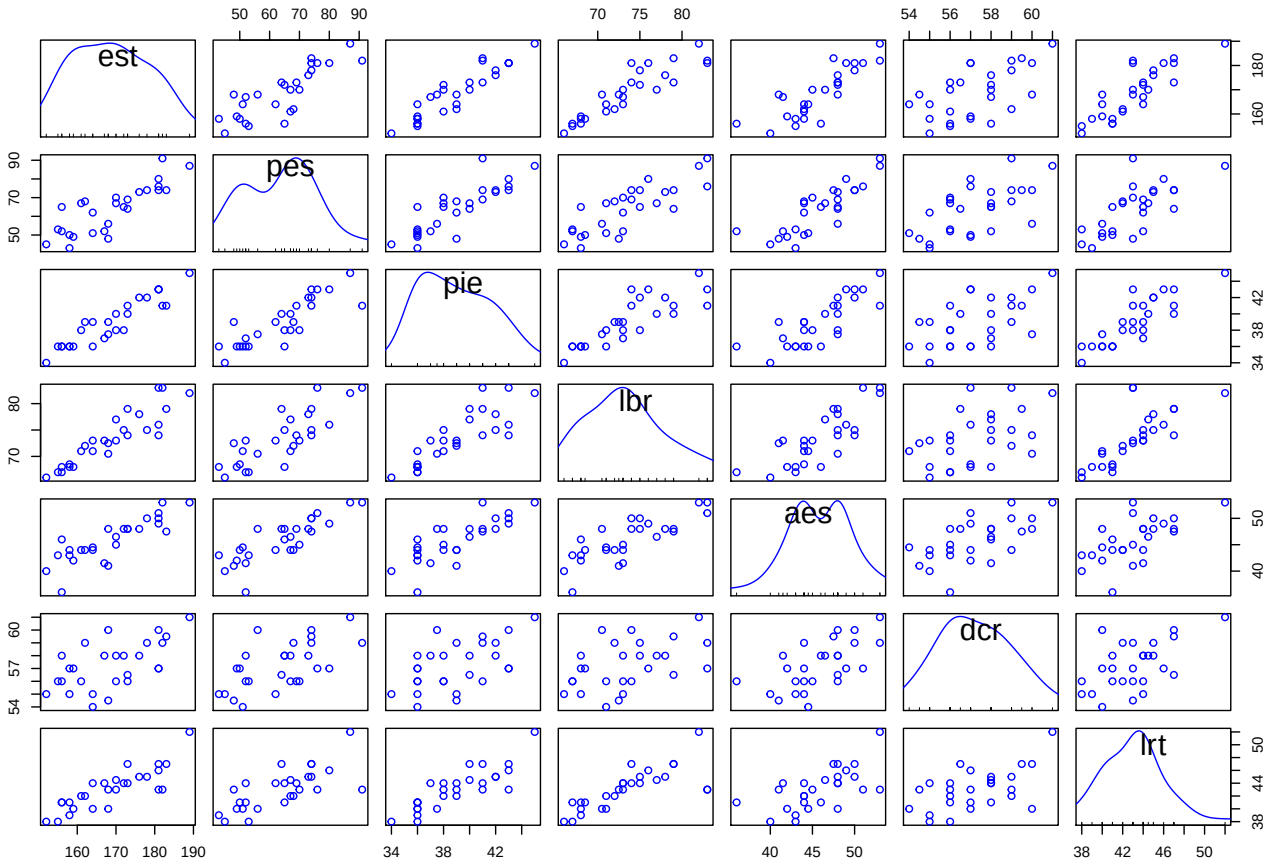
Matriz de dispersión: otras especificaciones mujeres / hombres

```
# medifis[,-1] representa el banco de datos expcepto sexo
pairs(medifis[, -1], panel = panel.smooth, col = 3 + medifis[, 1],
    main = "medifis")
```

Ejemplo:

Matriz de dispersión: otra función

```
car::scatterplotMatrix(medifis[, -1], regLine = FALSE, smooth = FALSE)
```



Gráficos condicionados: Coplots

Coplot: Se consideran diferentes particiones de una de las variables y para cada partición se representa una nube de puntos de otras dos. Se puede visualizar la relación entre las tres variables.

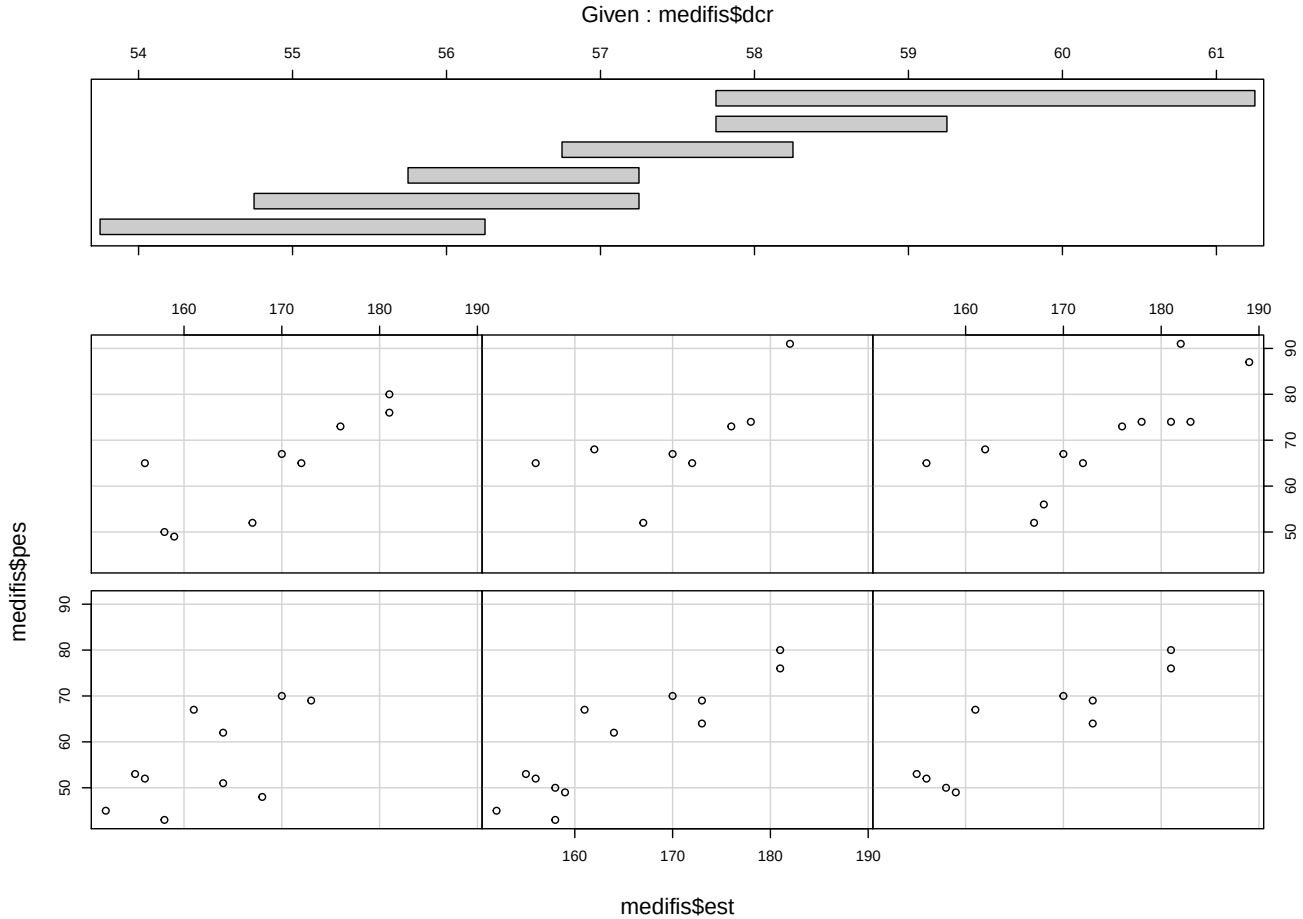
En *R* se puede obtener, entre otras, con las funciones de las siguientes librerías:

```
UsingR::coplot(x)

lattice::xyplot(x)

lattice::bwplot(x)
```

Ejemplo:

coplot (*medifis*)

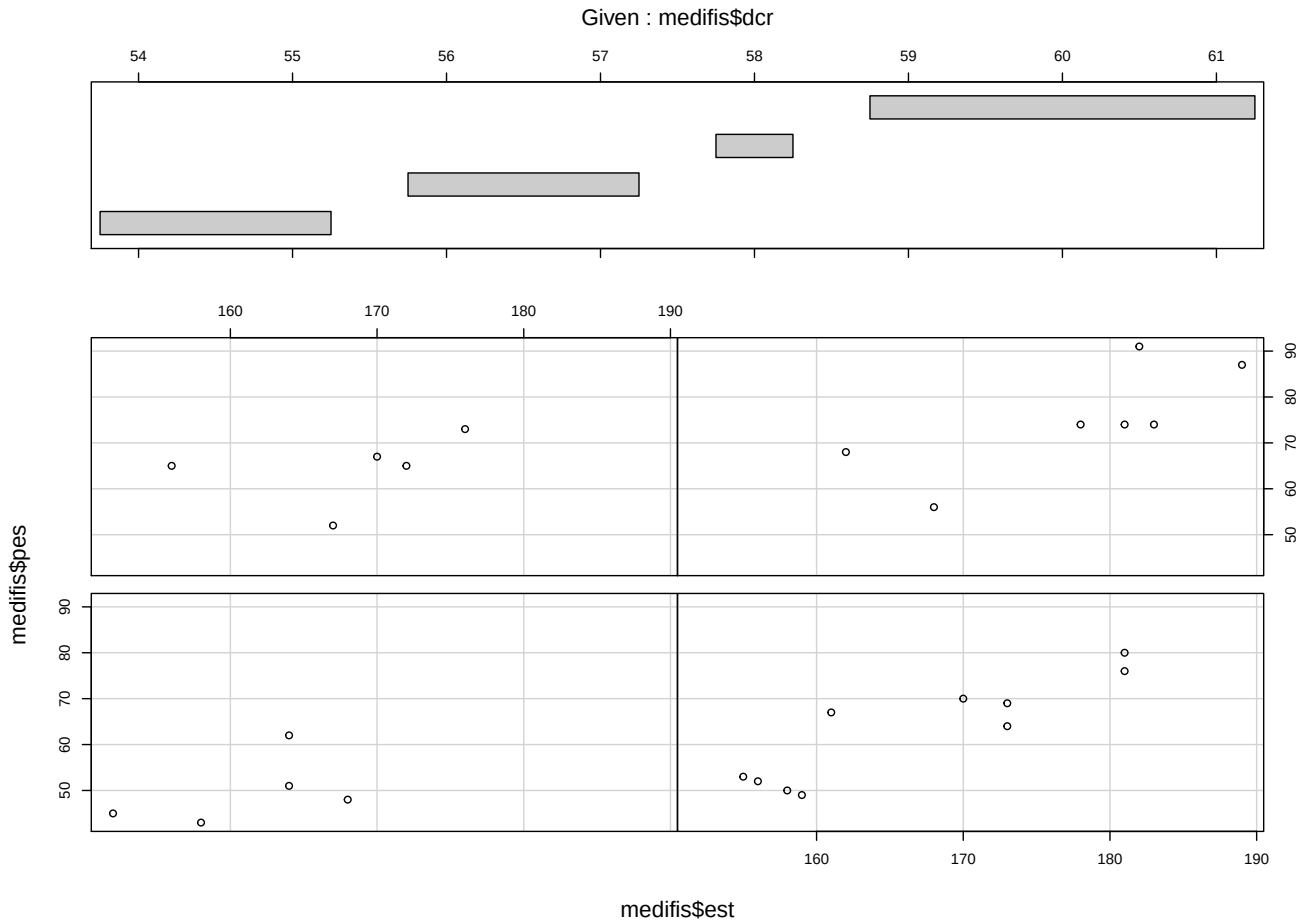
```
library(UsingR)
coplot(medifis$pes ~ medifis$est | medifis$dcr)
```



Ejemplo:

coplot (*medifis*)

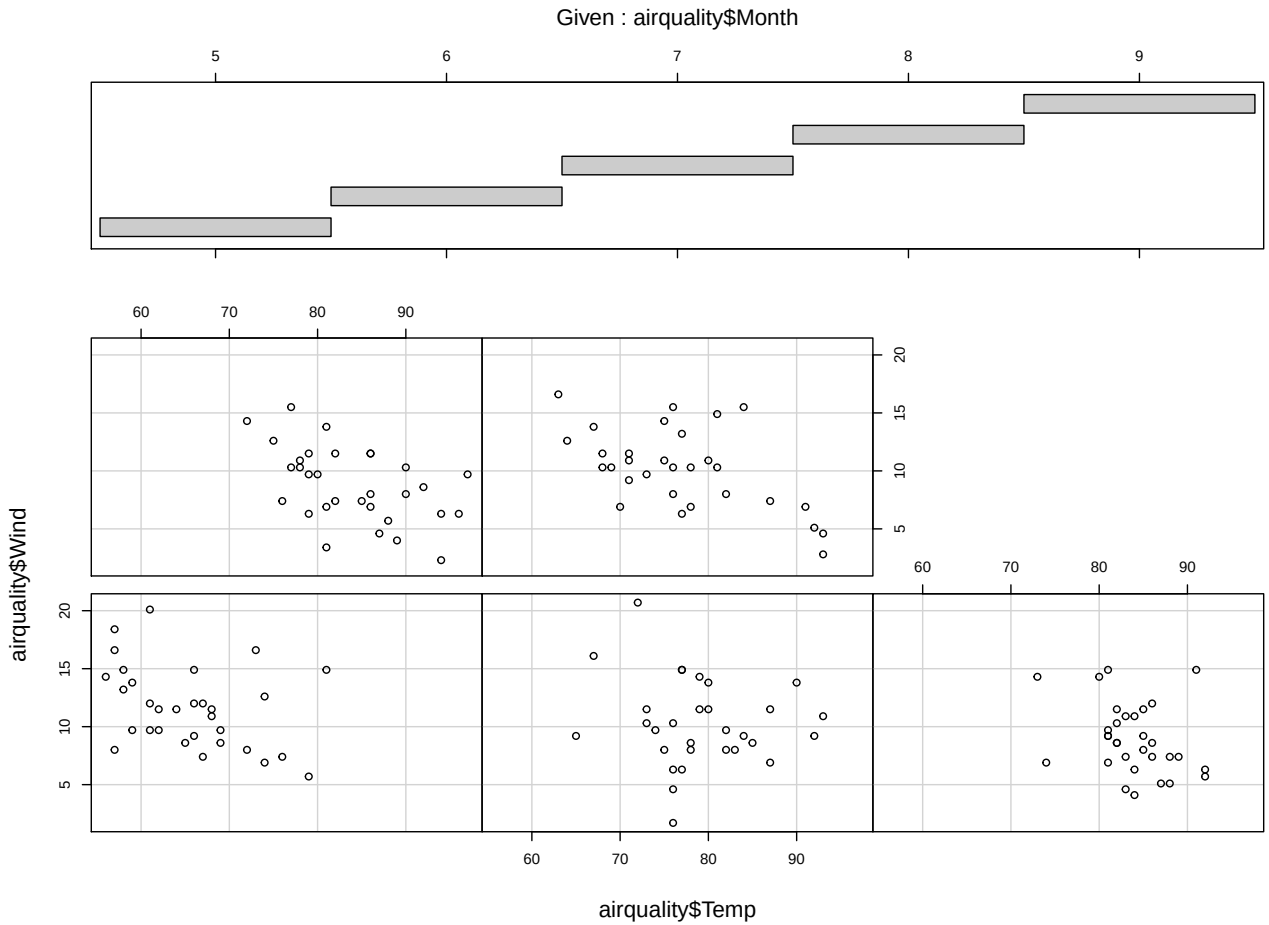
```
coplot(medifis$pes ~ medifis$est | medifis$dcr, number = 4, overlap = -0.5)
```



Ejemplo:

coplot (*airquality*)

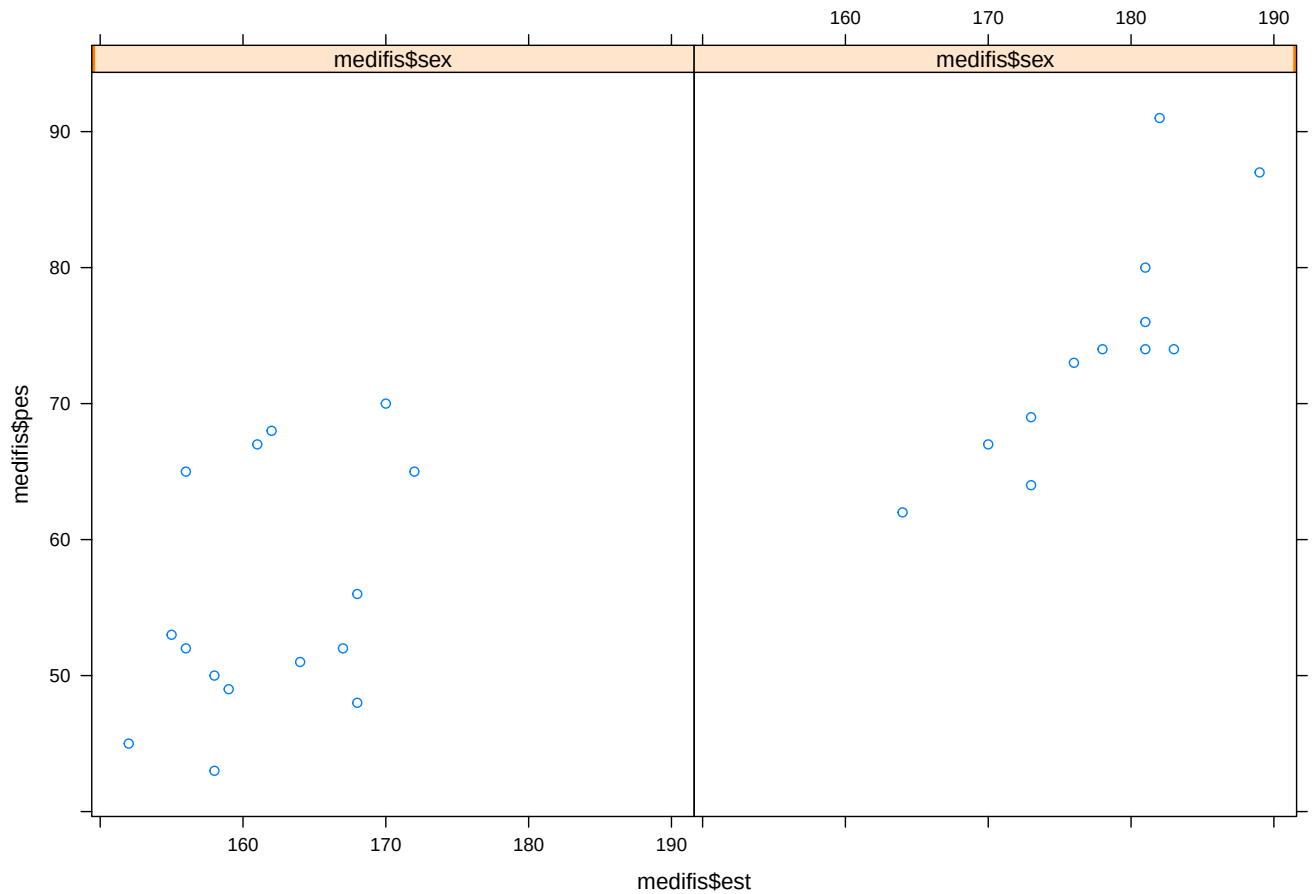
```
coplot(airquality$Wind ~ airquality$Temp | airquality$Month, number = 5,
    overlap = -0.5)
```



Ejemplo:

xyplot (*medifis*)

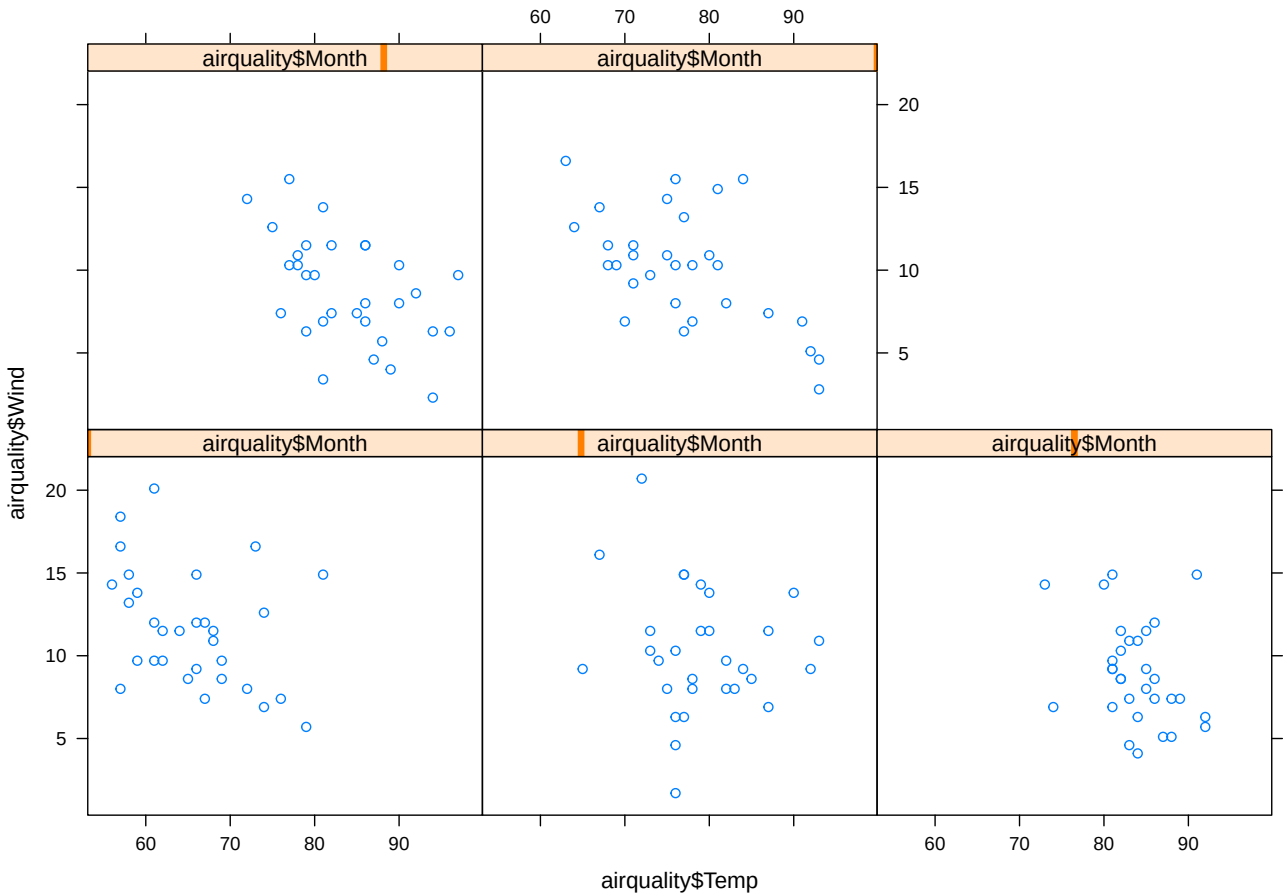
```
library(lattice)
xyplot(medifis$pes ~ medifis$est | medifis$sex)
```



Ejemplo:

xyplot (*airquality*)

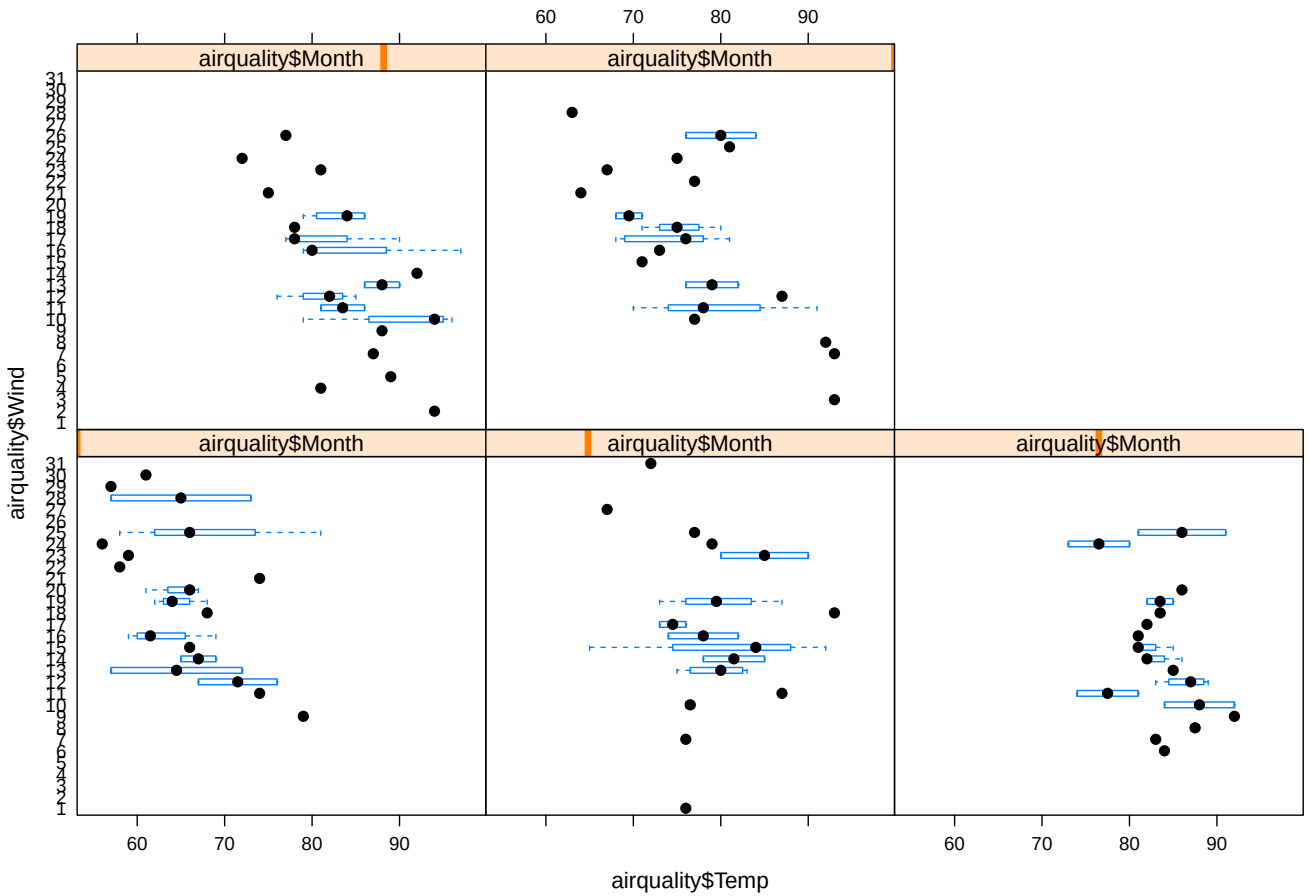
```
xyplot(airquality$Wind ~ airquality$Temp | airquality$Month)
```



Ejemplo:

bwplot (*airquality*)

```
bwplot(airquality$Wind ~ airquality$Temp | airquality$Month)
```



Gráficos tridimensionales

Gráficos tridimensionales: Se pueden representar tres variables cuantitativas simultáneamente (incluso separar por alguna otra)

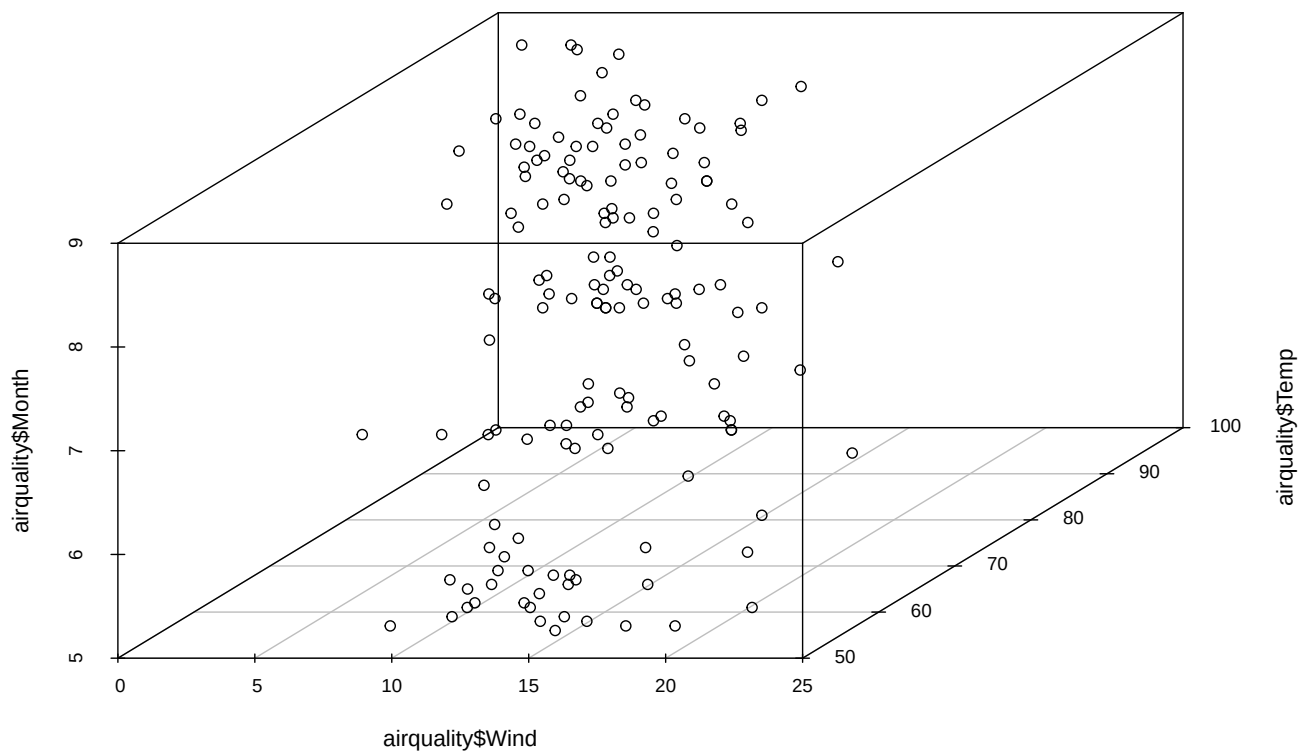
En *R* se puede obtener, entre otras, con las funciones de las siguientes librerías:

```
scatterplot3d::scatterplot3d(x)


lattice::cloud(x)
```

Ejemplo:

scatterplot3d (*airquality*)

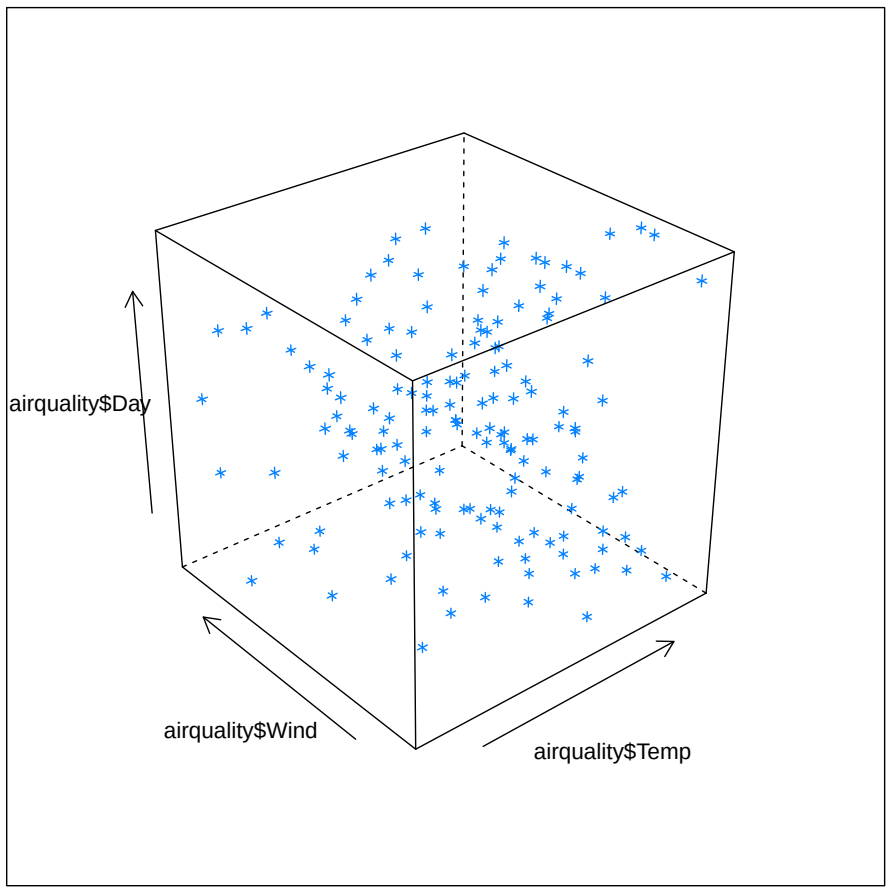
```
library(scatterplot3d)
scatterplot3d(airquality$Wind, airquality$Temp, airquality$Month)
```



Ejemplo:

cloud (*airquality*)

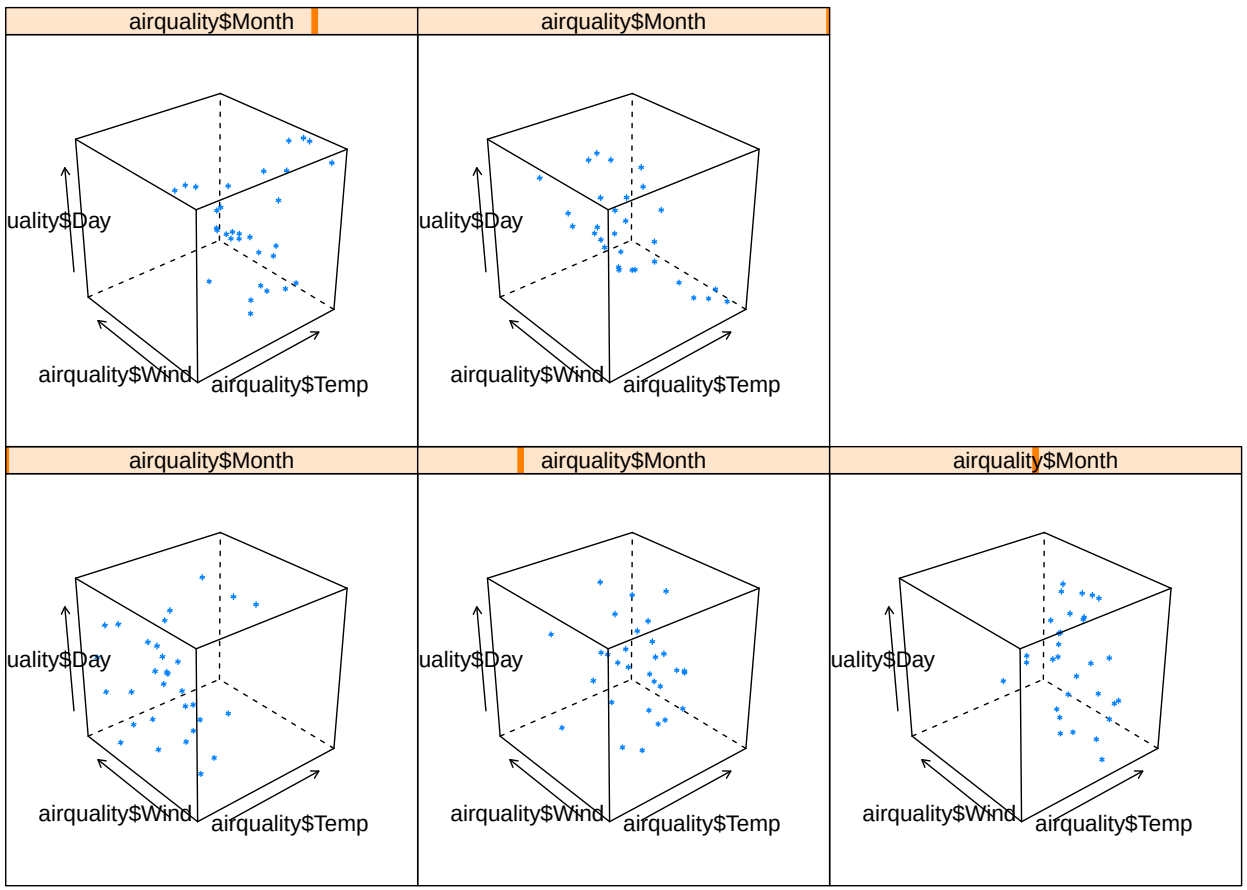
```
cloud(airquality$Day ~ airquality$Temp * airquality$Wind)
```



Ejemplo:

cloud (*airquality*)

```
cloud(airquality$Day ~ airquality$Temp * airquality$Wind | airquality$Month)
```



Gráficos multidimensionales: Caras de Chernoff

Caras de Chernoff: Podemos mostrar hasta 15 características por objeto representando cada individuo de la muestra con una cara en la que cada variable represente una característica:

- 1.altura de la cara
- 2.ancho de la cara
- 3.forma de la cara
- 4.altura de la boca
- 5.ancho de la boca
- 6.curva de la sonrisa
- 7.altura de los ojos
- 8.ancho de los ojos
- 9.altura del pelo
- 10.ancho del pelo
- 11.estilo de cabello
- 12.altura de la nariz
- 13.ancho de la nariz
- ...

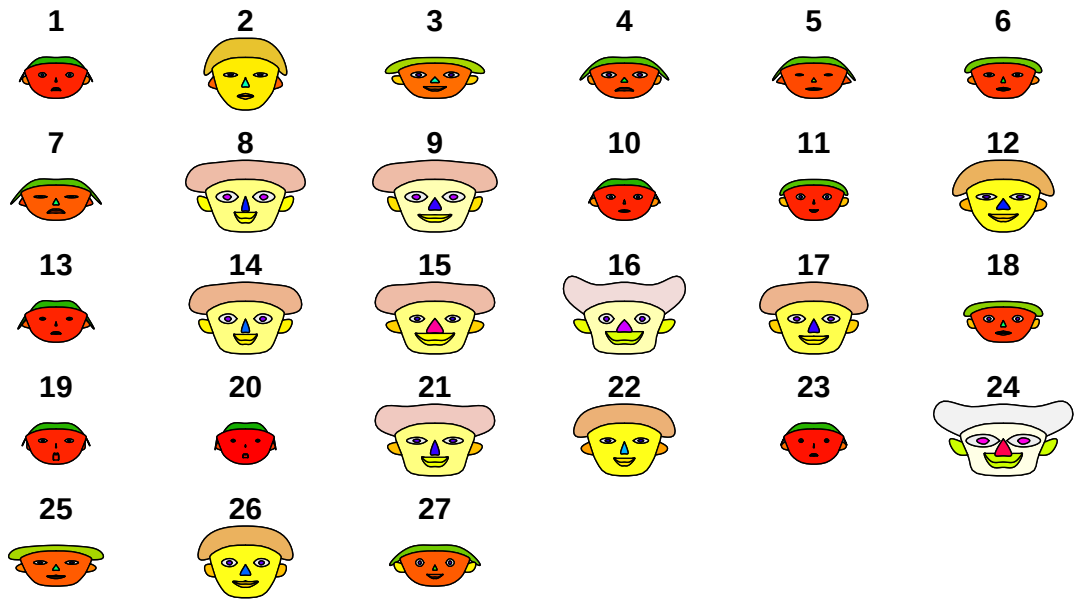
Se puede usar cuando el número de variables a comparar es *"razonable"*.

En *R* se puede obtener, entre otras, con la función:

```
aplpack::faces(x)
```

Ejemplo:

```
library(aplpack)
faces(xy = medifis)
```



Ejemplo:

```
library(aplpack)
faces(xy = medifis, plot.faces = FALSE)
```

```
## effect of variables:
##  modified item       Var
##  "height of face   " "sex"
##  "width of face    " "est"
##  "structure of face" "pes"
##  "height of mouth  " "pie"
##  "width of mouth   " "lbr"
##  "smiling          " "aes"
##  "height of eyes   " "dcr"
##  "width of eyes    " "lrt"
##  "height of hair   " "sex"
##  "width of hair   "  "est"
##  "style of hair   "  "pes"
##  "height of nose  "  "pie"
##  "width of nose   "  "lbr"
##  "width of ear    "  "aes"
##  "height of ear   "  "dcr"
```

Ejemplo:

Explorad argumentos de la función. . . como *face.type*, pues hay alguno muy *navideño* ;-) (Por ejemplo con el parámetro *labels* podríamos indicar los nombres de cada individuo)

```
library(aplpack)
faces(xy = medifis, face.type = "2")
```

Gráficos multidimensionales: Stars

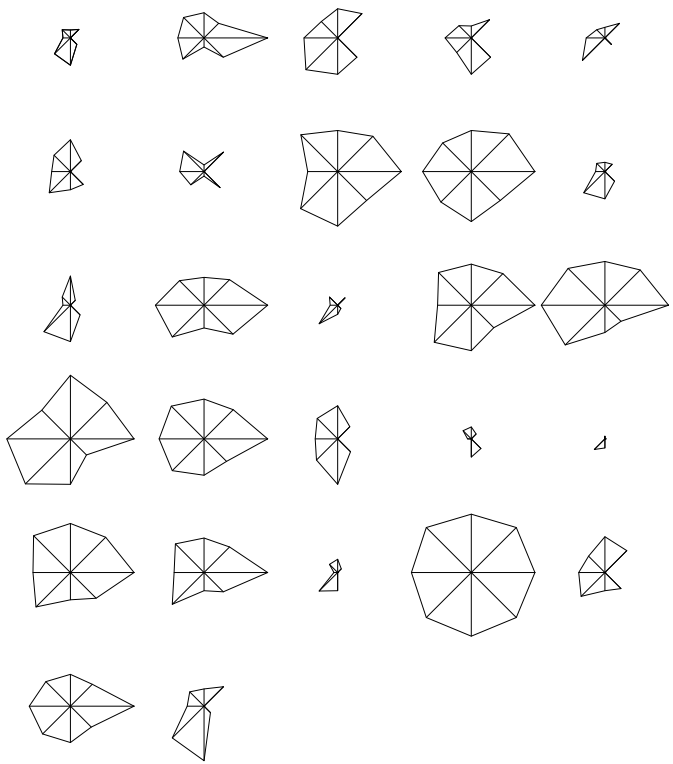
Gráficas de Estrellas: Consisten en representar los objetos con estrellas, cuyas puntas representan de forma proporcional el valor de una variable.

En *R* se puede obtener, entre otras, con la función *stars*, que posee diferentes parámetros que nos permiten definir diferentes formas del gráfico:

```
graphics::stars(x)
```

Ejemplo:

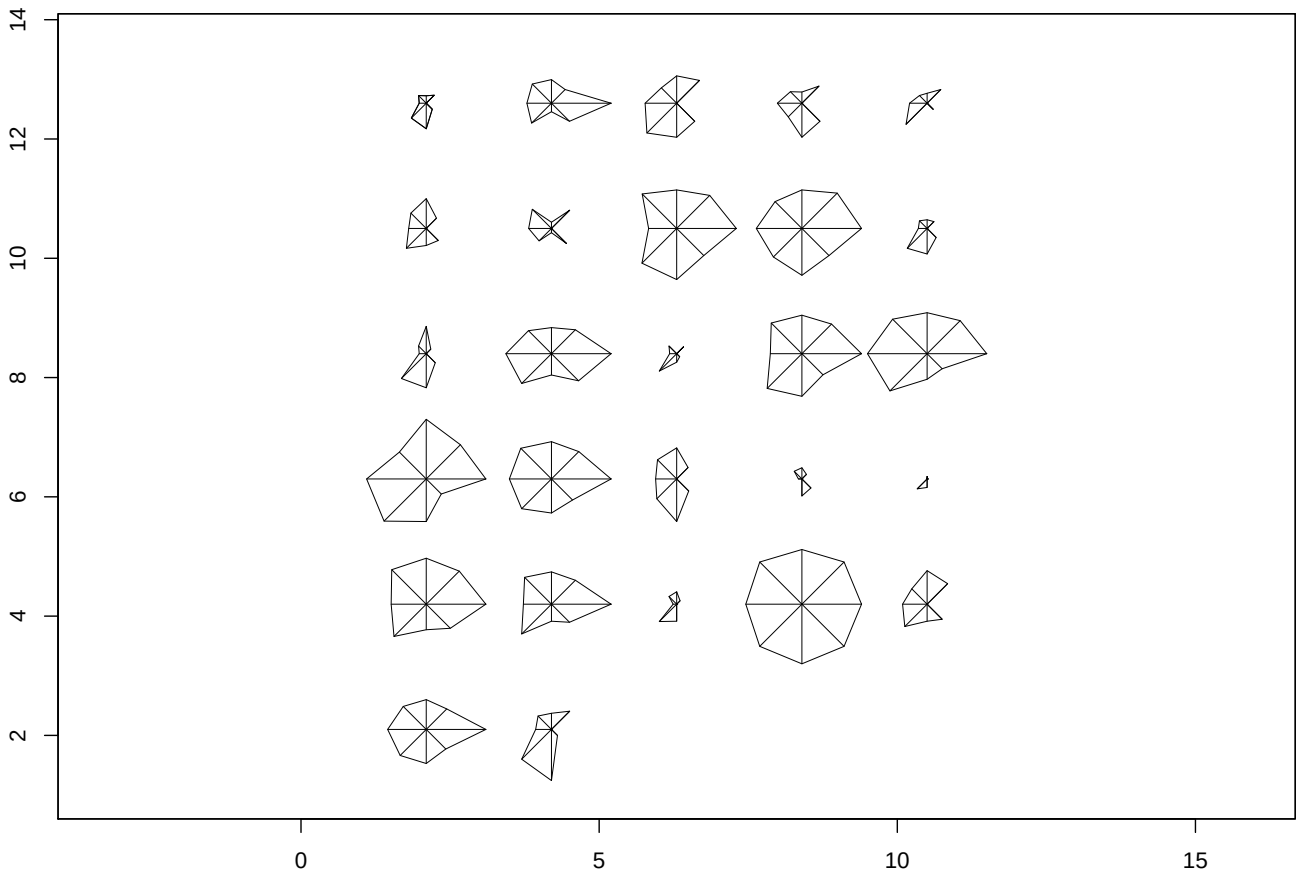
```
stars(medifis, col.lines = "black")
```



Ejemplo:

(Si queremos incluir una leyenda para conocer la dirección de cada variable podemos representar los ejes para obtener las coordenadas en las que queremos la leyenda)

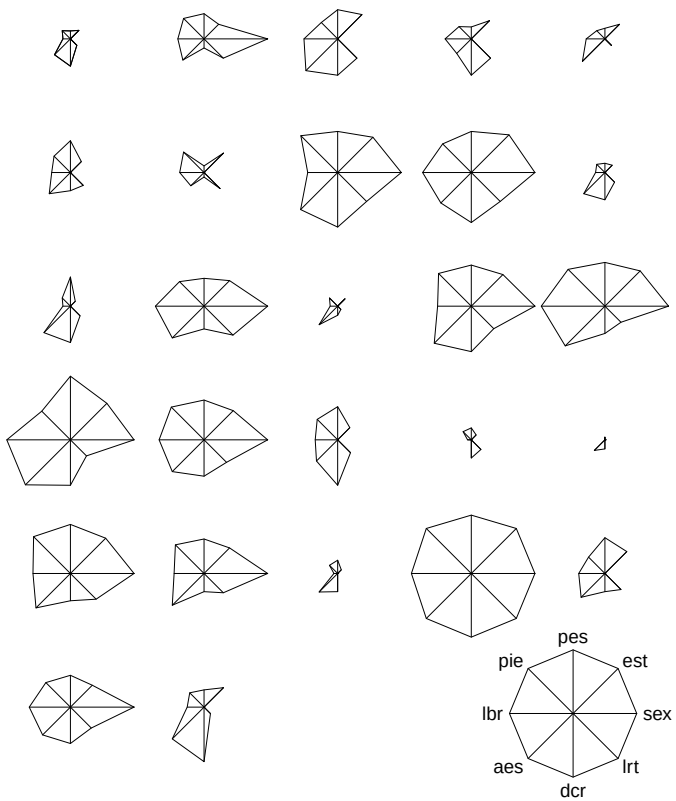
```
stars(medifis, col.lines = "black", axes = TRUE)
```



Ejemplo:

Con el parámetro *key.loc* indicamos las coordenadas en las que queremos situar la leyenda:

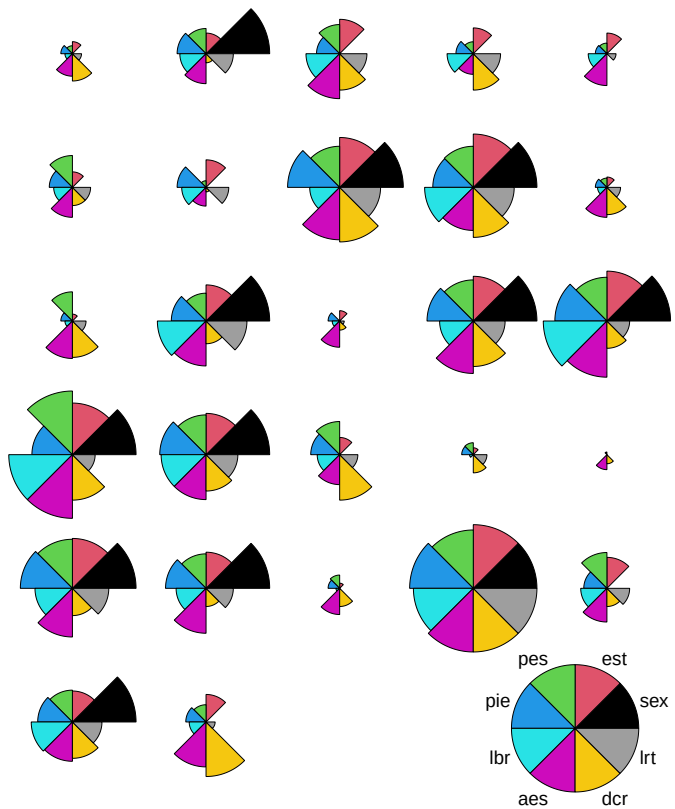
```
stars(medifis, col.lines = "black", key.loc = c(10, 2))
```



Ejemplo:

Podemos explorar los parámetros de esta función para obtener otras formas:

```
stars(medifis, key.loc = c(10, 2), draw.segments = TRUE)
```



(y con el parámetro *labels* podríamos indicar los nombres de cada individuo)

Radar charts

Para representar este tipo de gráficos, también denominados en la literatura como gráficos de radar o araña (*radar* o *spider* charts), Podemos usar también la función *radarchart* de la librería *fmsb*.

```
radarchart(x, maxmin = TRUE)
```

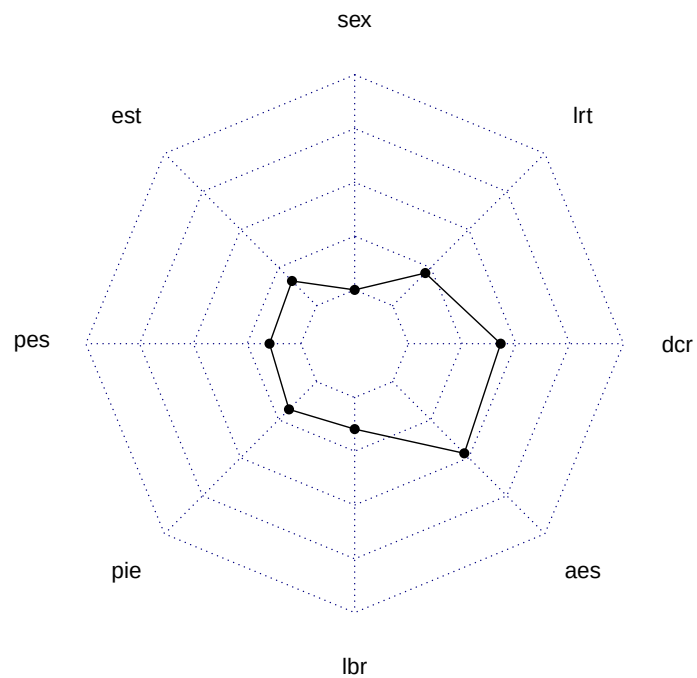
Esta función puede representar un individuo o varios simultánemanete sobre el mismo gráfico.

Debemos tener en cuenta el valor del parámetro *maxmin*. Si este parámetro tomar el valor **TRUE** (valor por defecto) la función espera encontrar en la primera fila de los datos los valores máximos de cada variable y en la segunda fila los valore mínimos que servirán de referencia para representar los individuos que se encontrarán de la fila 3 en adelante. Si este parámetro toma el valor **FALSE** los valores máximo y mínimo se calculan sobre el conjunto de filas que van a ser representadas (no sirve para representar un único individuo).

Ejemplo:

Representación de **un individuo**:

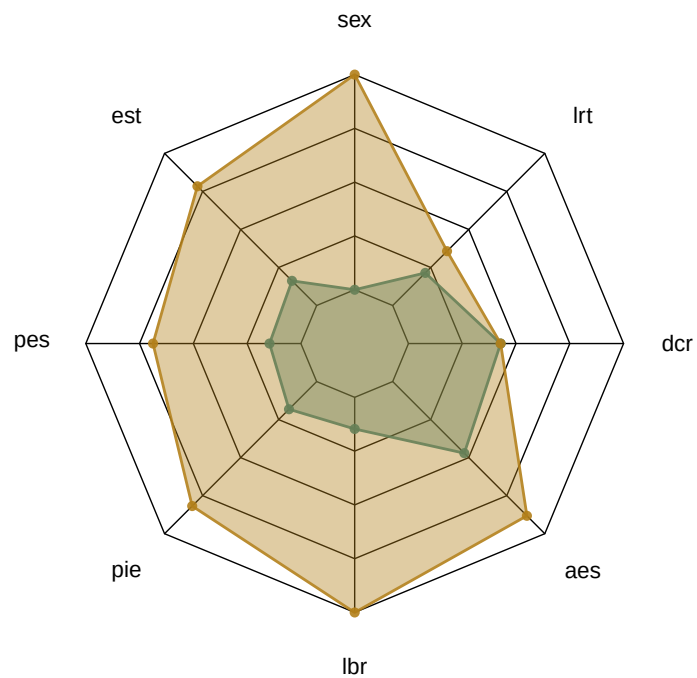
```
# Calculamos el vector de máximos
library(fmsb)
v.max <- apply(medifis, 2, max)
v.min <- apply(medifis, 2, min)
# Representamos al individuo 10
radarchart(rbind(v.max, v.min, medifis[10, ]))
```



Ejemplo:

Representación de **más de un individuo**:

```
#Representamos a los individuos 10 y 15 (y añadimos más parametrización):
colors_border=c( rgb(0.2,0.5,0.5,0.9),  rgb(0.7,0.5,0.1,0.9) )
colors_in=c( rgb(0.2,0.5,0.5,0.4) , rgb(0.7,0.5,0.1,0.4) )
radarchart(rbind(v.max,v.min,medifis[c(10,15),]),
           cglcol = "black",cglty = 1,lwd=1,#Grid
           pcol=colors_border,pfcol=colors_in,plwd = 2,plty = 1) #Polygon)
```



Gráficos multidimensionales: Fourier plot

Andrew's Fourier plot: Propone como representación de un vector k-dimensional $\mathbf{x} = (\mathbf{x_1}, ..., \mathbf{x_k})$ la función:

$$f_{\mathbf{x}}(t) = \frac{x_1}{\sqrt{2}} + x_2 \sin(t) + x_3 \cos(t) + x_4 \sin(2t) + x_5 \cos(2t) + ... \quad -\pi < t < \pi$$

Ejemplo:

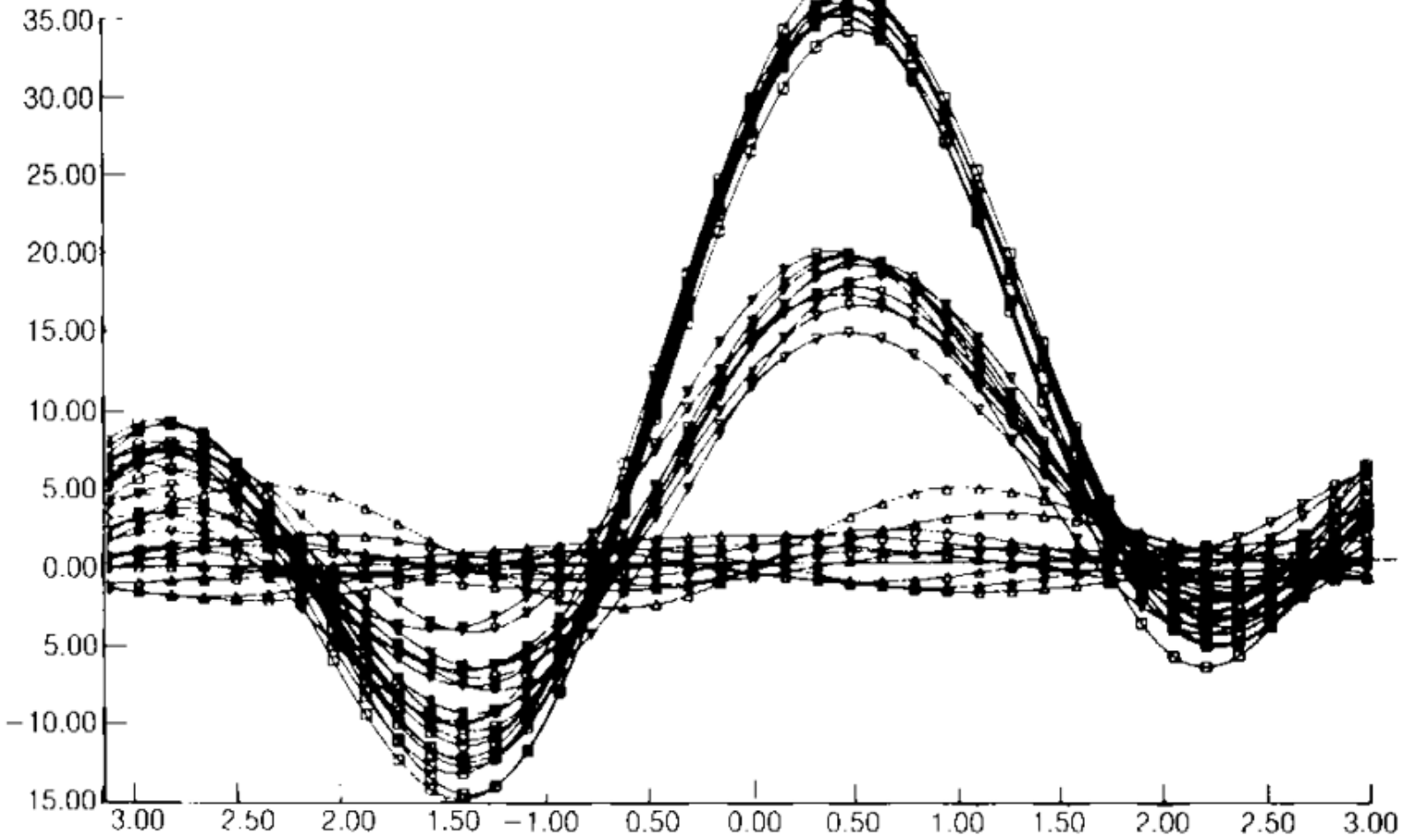


Figure 3.7 Andrews plot for 30 five-dimensional observations

Andrew's Fourier plot

- Un **conjunto de observaciones de cada individuo** multivariante aparece como **una línea en un gráfico**.
- Esta función **preserva la distancia euclídea** en el sentido:

$$\int_{-\pi}^{\pi} [f_x(t) - f_y(t)]^2 = \pi \sum_{i=1}^{p} (x_i - y_i)^2 = \pi ||x - y||^2$$

- **Útil** para **detectar observaciones aberrantes** y para **detectar grupos de observaciones** bien separados de otros grupos.
- **No recomendado** si el **número de individuos es elevado**, puede resultar poco clarificador.

4. Outliers (Datos atípicos)

Outliers

Denominamos como *outliers* o *casos atípicos* a aquellos casos para los que una (o varias) variables **toman valores extremos** que difieren del comportamiento del resto de la muestra.

Es **importante detectar** estos posibles valores debido a que **pueden distorsionar los resultados** (y en caso de necesitar el cumplimiento de alguna condición como la *Normalidad* para aplicar alguna técnica particular pueden afectar su cumplimiento).

Pueden producirse o bien **por un error** en el proceso de recogida de los datos **o bien** por **la variabilidad** de los propios datos.

¿Qué hacemos con los outliers una vez detectados?

- Si **podemos confirmar que se trata de un error**: tratar de **recuperar** la información correcta **o** bien **eliminarlos**.
- Si **confirmamos su autenticidad**: la solución no es tan clara... ¿Transformar la/s variable/s?, ¿Utilizar técnicas robustras frente a outliers (no paramétricas)?,...En el ámbito bayesiano, en el módulo de *modelos jerárquicos bayesianos*, se verá un ejemplo práctico para estos casos.

Detección univariante de casos atípicos.

Podemos hacer unas **comprobaciones univariantes** con unos diagramas de cajas para **todas las variables**.

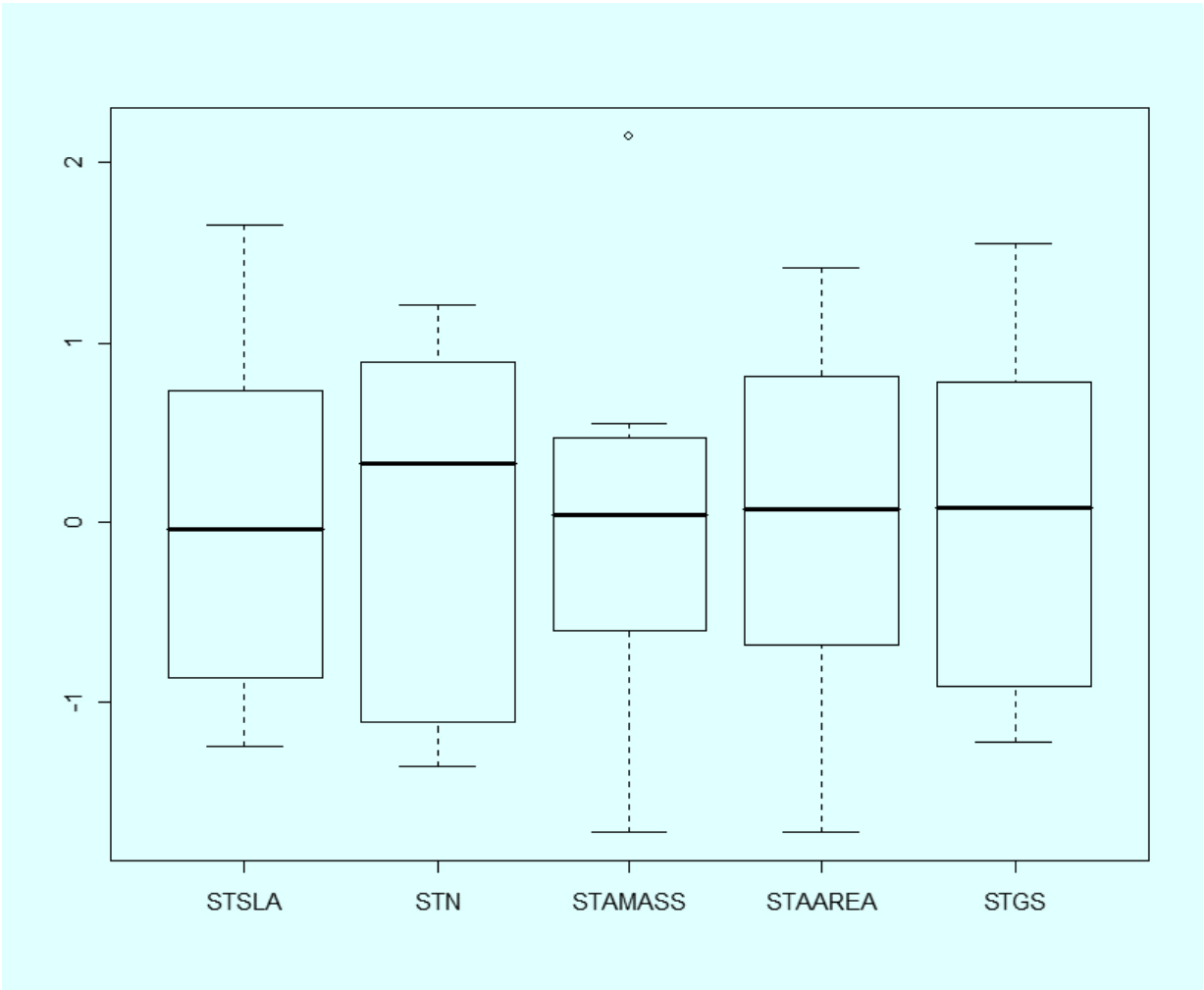
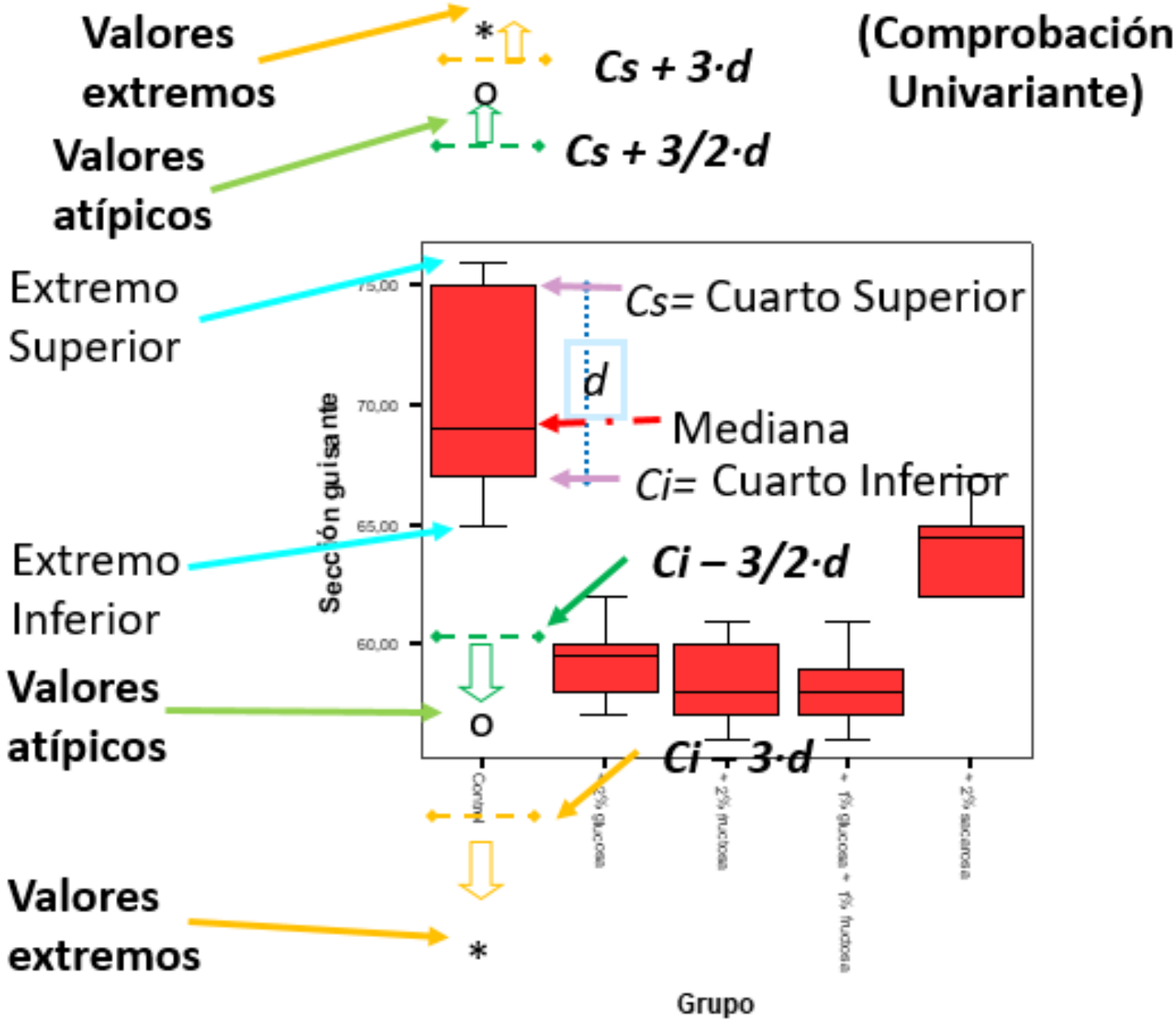


Diagrama de cajas



(Comprobación Univariante)

Detección multivariante de casos atípicos: **gráficos de coordenadas**.

En los **gráficos de coordenadas paralelas** se consideran varios ejes paralelos entre sí, y cada uno representa un atributo (o variable).

Los **valores de cada variable se escalan** para poder ser representados con la misma escala.

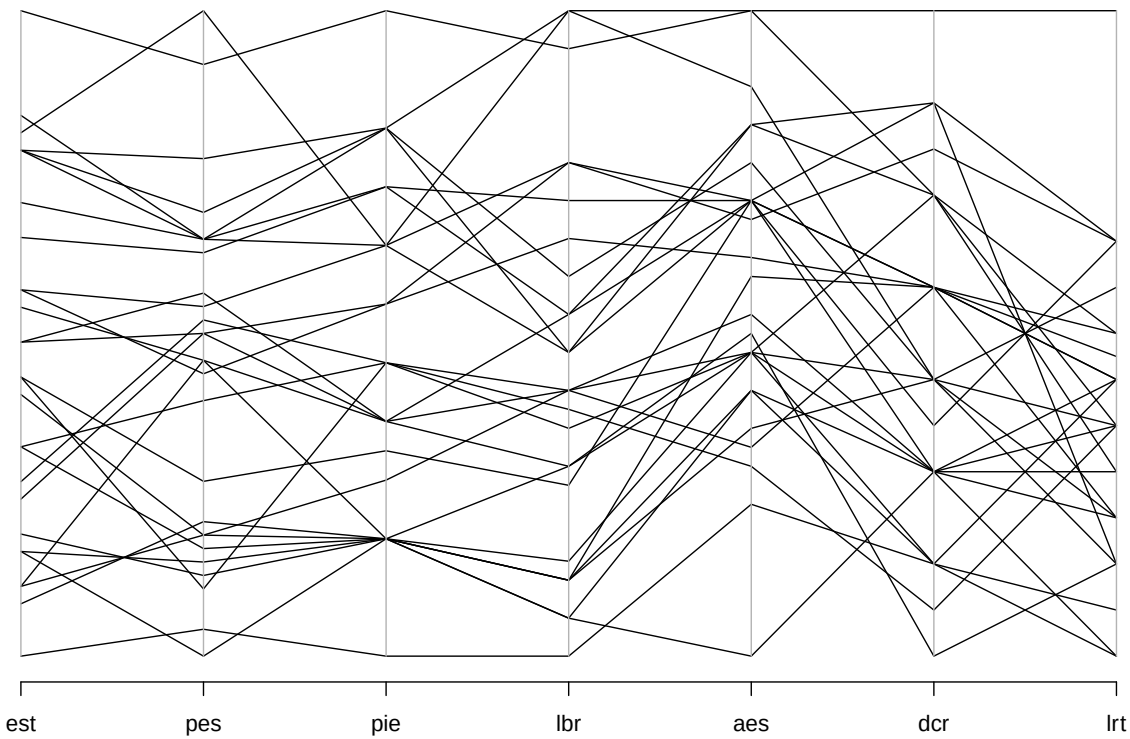
Cada individuo queda representado por **una línea** según los valores que obtiene en cada variable. Los individuos similares tienden a agruparse en líneas con trayectoria similar

En *R* se puede obtener, entre otras, con la función **parcoord** de la librería *MASS*:

```
MASS::parcoord(x)
```

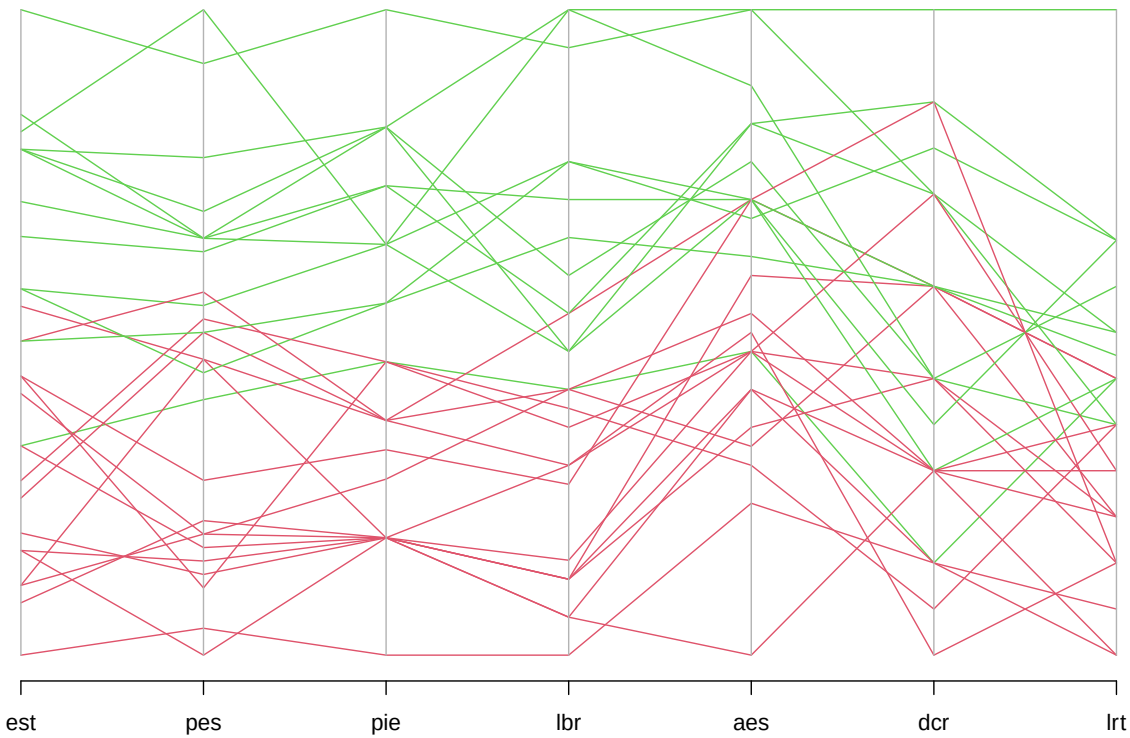
Detección multivariante de casos atípicos: gráficos de coordenadas.

```
MASS::parcoord(medifis[, 2:8])
```



Detección multivariante de casos atípicos: gráficos de coordenadas.

```
MASS::parcoord(medifis[, 2:8], col = medifis[, 1] + 2)
```



Detección multivariante de casos atípicos: **Distancia de Mahalanobis**.

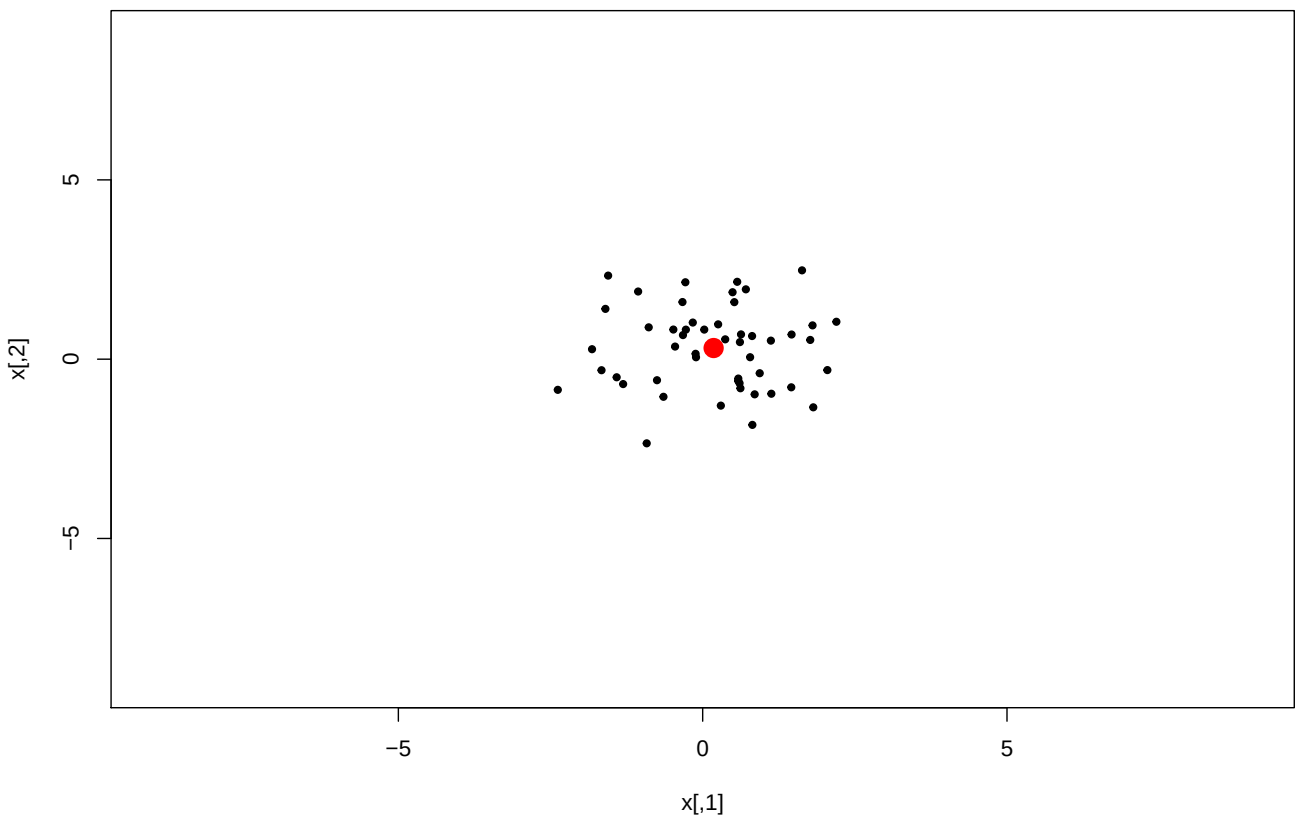
Para detectar los casos con valores de las variables *atípicos* desde un punto de vista multivariante, podemos considerar que cada individuo con sus k variables medidas representa un punto en un espacio de k dimensiones.

Podríamos considerar el **centroide** de la nube de puntos que forman todos los individuos y calcular la distancia de cada punto a ese centroide.

Ordenando dichas distancias podríamos explorar aquellas más grandes y establecer un límite por encima del cuál nos parecería necesario revisar el caso.

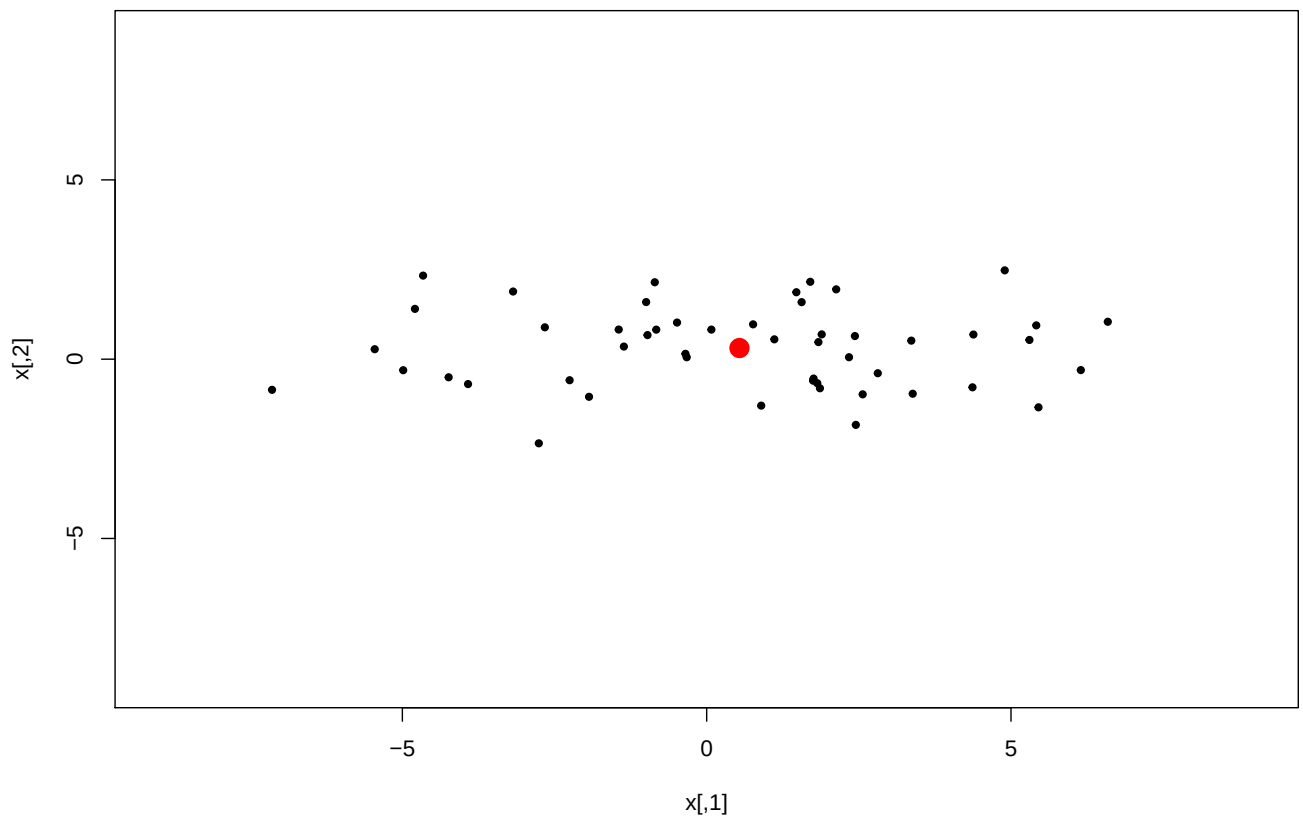
- Ejemplo: $k = 2$ variables independientes.

```
set.seed(seed = 12345)
x <- cbind(rnorm(50), rnorm(50))
plot(x, pch = 20, xlim = c(-9, 9), ylim = c(-9, 9))
mx <- apply(x, 2, mean)
points(mx[1], mx[2], col = "red", pch = 16, cex = 2)  #Centroide
```



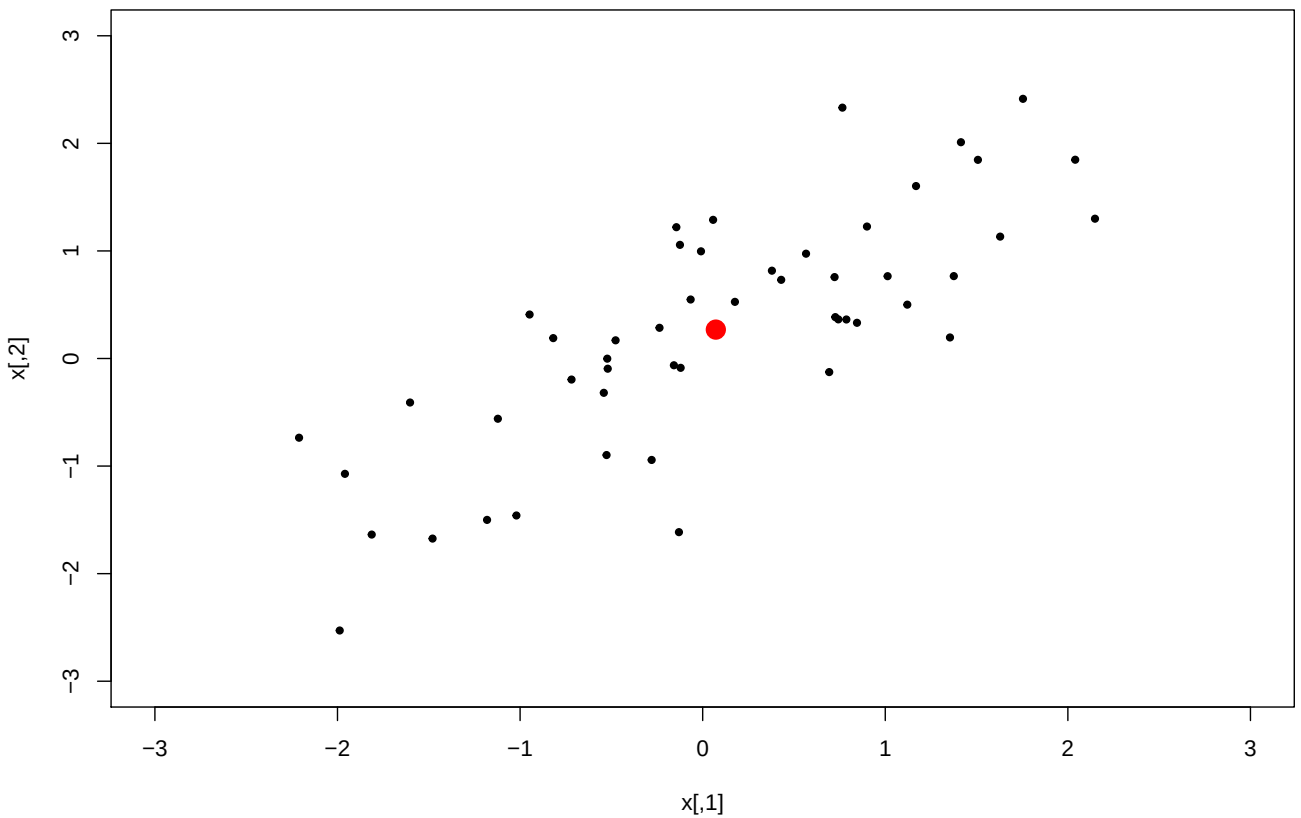
- Pero. . . ¿y si la varianza de ambas variables no fuera la misma?

```
set.seed(seed = 12345)
x <- cbind(rnorm(50, 0, 3), rnorm(50, 0, 1))
plot(x, pch = 20, xlim = c(-9, 9), ylim = c(-9, 9))
mx <- apply(x, 2, mean)
points(mx[1], mx[2], col = "red", pch = 16, cex = 2)  #Centroide
```



- Pero. . . ¿y si las variables no fueran independientes?

```
library(MASS)  #Simularemos valores de una distrib.Normal Multivariante
set.seed(seed = 12345)
x <- mvrnorm(50, mu = c(0, 0), Sigma = matrix(c(1, 0.8, 0.8, 1), nrow = 2))
plot(x, pch = 20, xlim = c(-3, 3), ylim = c(-3, 3))
mx <- apply(x, 2, mean)
points(mx[1], mx[2], col = "red", pch = 16, cex = 2)  #Centroide
```



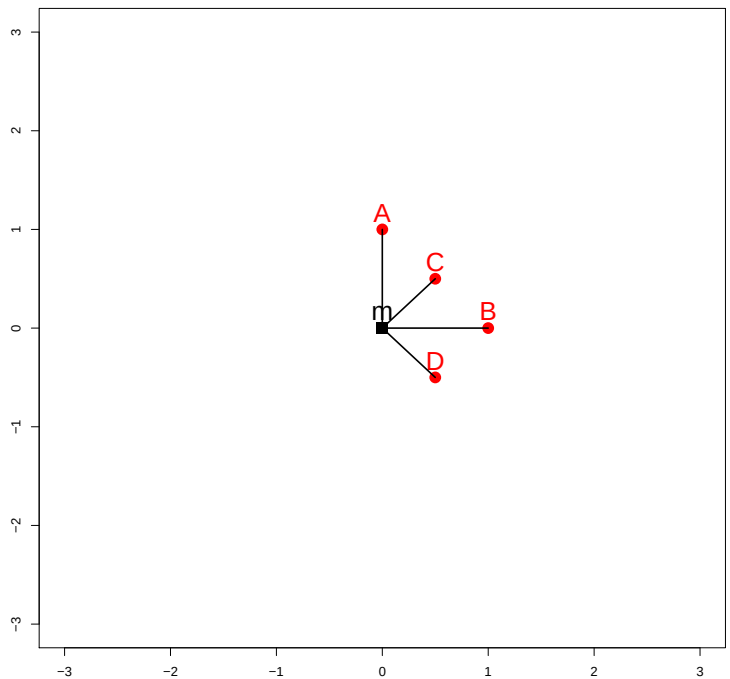
- Para evitar el efecto de la varianza en el cálculo de las distancias tenemos la posibilidad de *estandarizar* las variables (transformarlas para hacer que todas ellas tengan la misma varianza).
- Pero esta acción no corrige la posible covarianza existente entre ellas.
- La **distancia de Mahalanobis** es una distancia *estadística* que tiene en cuenta la covarianza.
- Si los datos son **continuos** y podemos **suponer normalidad multivariante**, podemos usar la distancia de Mahalanobis.
- Si consideramos que S es la matriz de varianzas-covarianzas de un conjunto de datos X, y $\overline{\mathbf{x}}$ representa su vector de medias, podemos calcular la **distancia de Mahalanobis** de un individuo i al centroide de la nube de puntos como:

$$d_m(i) = \left((\mathbf{x_i} - \overline{\mathbf{x}})' \mathbf{S^{-1}} (\mathbf{x_i} - \overline{\mathbf{x}})\right)^{\frac{1}{2}}$$

- Ejemplo:

Consideremos los siguientes puntos A, B, C y D y el centroide de un banco de datos que consideraremos $m = c(0, 0)$:

```
A <- c(0, 1)
B <- c(1, 0)
C <- c(0.5, 0.5)
D <- c(0.5, -0.5)
m <- c(0, 0)   #Centroide
```



- Escenario 1: Dos variables **incorreladas** con la **misma varianza**.

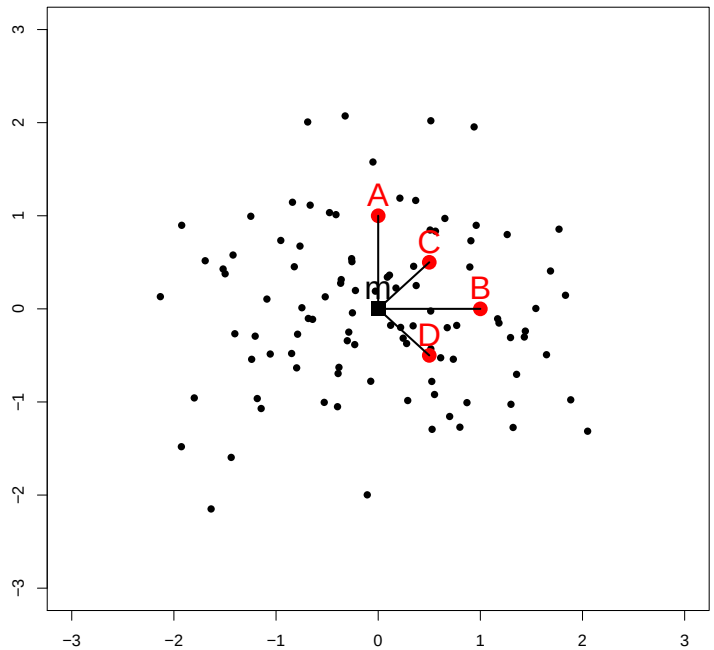
$$\mathbf{y} \sim \mathbf{N_2}\left(\mathbf{m} = \begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix}, \mathbf{S} = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\right)$$

$$d_m^2 = (y_1 - m_1, y_2 - m_2)\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}^{-1}\begin{pmatrix}y_1 - m_1\\ y_2 - m_2\end{pmatrix} =$$

$$= (y_1 - m_1)^2 + (y_2 - m_2)^2 = d_{euclidea}^2$$

- Escenario 1: Dos variables **incorreladas** con la **misma varianza**.

$$d_m^2(A, \mathbf{m}) = \mathbf{1}$$
$$d_m^2(B, \mathbf{m}) = \mathbf{1}$$
$$d_m^2(C, \mathbf{m}) = \mathbf{0.5}$$
$$d_m^2(D, \mathbf{m}) = \mathbf{0.5}$$



- Escenario 2: Dos variables **incorreladas** con **distinta varianza**.

$$\mathbf{y} \sim \mathbf{N_2}\left(\mathbf{m} = \begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix}, \mathbf{S} = \begin{pmatrix}0.5 & 0\\ 0 & 1\end{pmatrix}\right)$$

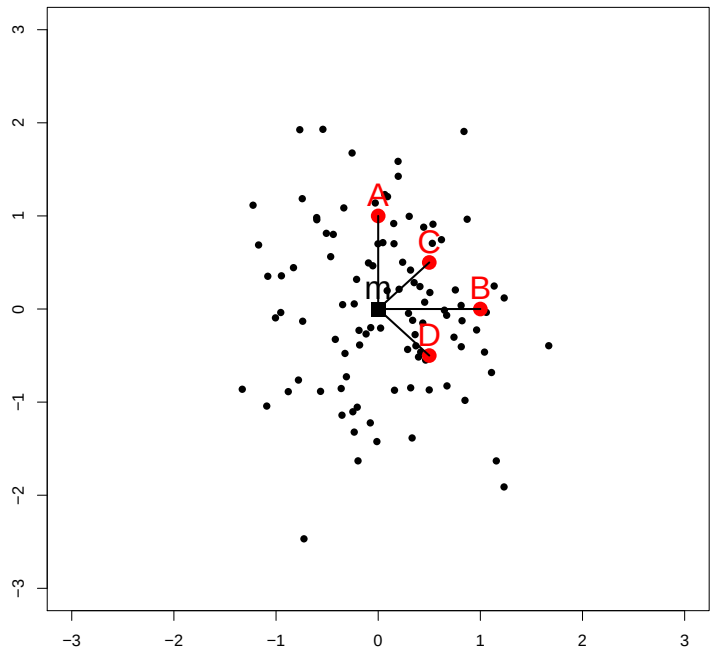
$$d_m^2 = (y_1 - m_1, y_2 - m_2)\begin{pmatrix}0.5 & 0\\ 0 & 1\end{pmatrix}^{-1}\begin{pmatrix}y_1 - m_1\\ y_2 - m_2\end{pmatrix} =$$

$$= (y_1 - m_1, y_2 - m_2)\begin{pmatrix}\frac{1}{0.5} & 0\\ 0 & \frac{1}{1}\end{pmatrix}\begin{pmatrix}y_1 - m_1\\ y_2 - m_2\end{pmatrix} =$$

$$= \left(\frac{y_1 - m_1}{0.5}\right)^2 + \left(\frac{y_2 - m_2}{1}\right)^2 \neq d^2_{euclidea}$$

- Escenario 2: Dos variables **incorreladas** con **distinta varianza**.

$$d_m^2(A, \mathbf{m}) = \mathbf{1}$$
$$d_m^2(B, \mathbf{m}) = \mathbf{2}$$
$$d_m^2(C, \mathbf{m}) = \mathbf{0.75}$$
$$d_m^2(D, \mathbf{m}) = \mathbf{0.75}$$



- Escenario 3: Dos variables **correlacionadas ($r = -0.70$)** con **distinta varianza**.

$$\mathbf{y} \sim \mathbf{N_2}\left(\mathbf{m} = \begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix}, \mathbf{S} = \begin{pmatrix} 0.5 & -0.49 \\ -0.49 & 1 \end{pmatrix}\right)$$

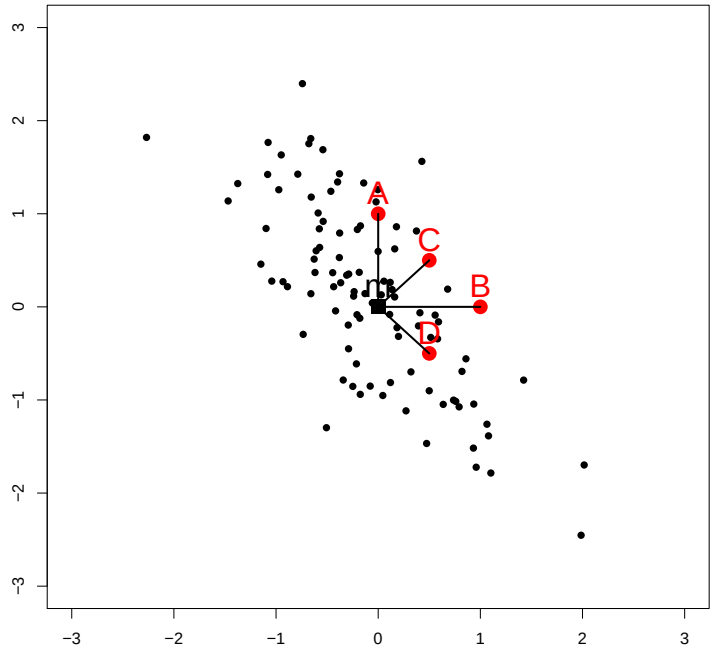
$$d_m^2 = (y_1 - m_1, y_2 - m_2) \begin{pmatrix} 0.5 & -0.49 \\ -0.49 & 1 \end{pmatrix}^{-1} \begin{pmatrix} y_1 - m_1 \\ y_2 - m_2 \end{pmatrix} =$$

$$= (y_1 - m_1, y_2 - m_2) \begin{pmatrix} 3.92 & 1.94 \\ 1.94 & 1.96 \end{pmatrix} \begin{pmatrix} y_1 - m_1 \\ y_2 - m_2 \end{pmatrix} =$$

$$= 3.92 \times (y_1 - m_1)^2 + 1.96 \times (y_2 - m_2)^2 + 2 \times 1.94 \times (y_1 - m_1) \times (y_2 - m_2)$$

- Escenario 3: Dos variables **correlacionadas ($r = -0.70$)** con **distinta varianza**.

$$d_m^2(A, \mathbf{m}) = \mathbf{1.96}$$
$$d_m^2(B, \mathbf{m}) = \mathbf{3.92}$$
$$d_m^2(C, \mathbf{m}) = \mathbf{2.44}$$
$$d_m^2(D, \mathbf{m}) = \mathbf{0.50}$$



Detección multivariante de casos atípicos: **Distancia de Mahalanobis**.

Para detectar los casos con valores de las variables *atípicos* desde un punto de vista multivariante podemos considerar el **centroide** de la nube de puntos que forman todos los individuos y calcular la distancia de **Mahalanobis** de cada individuo a la media.

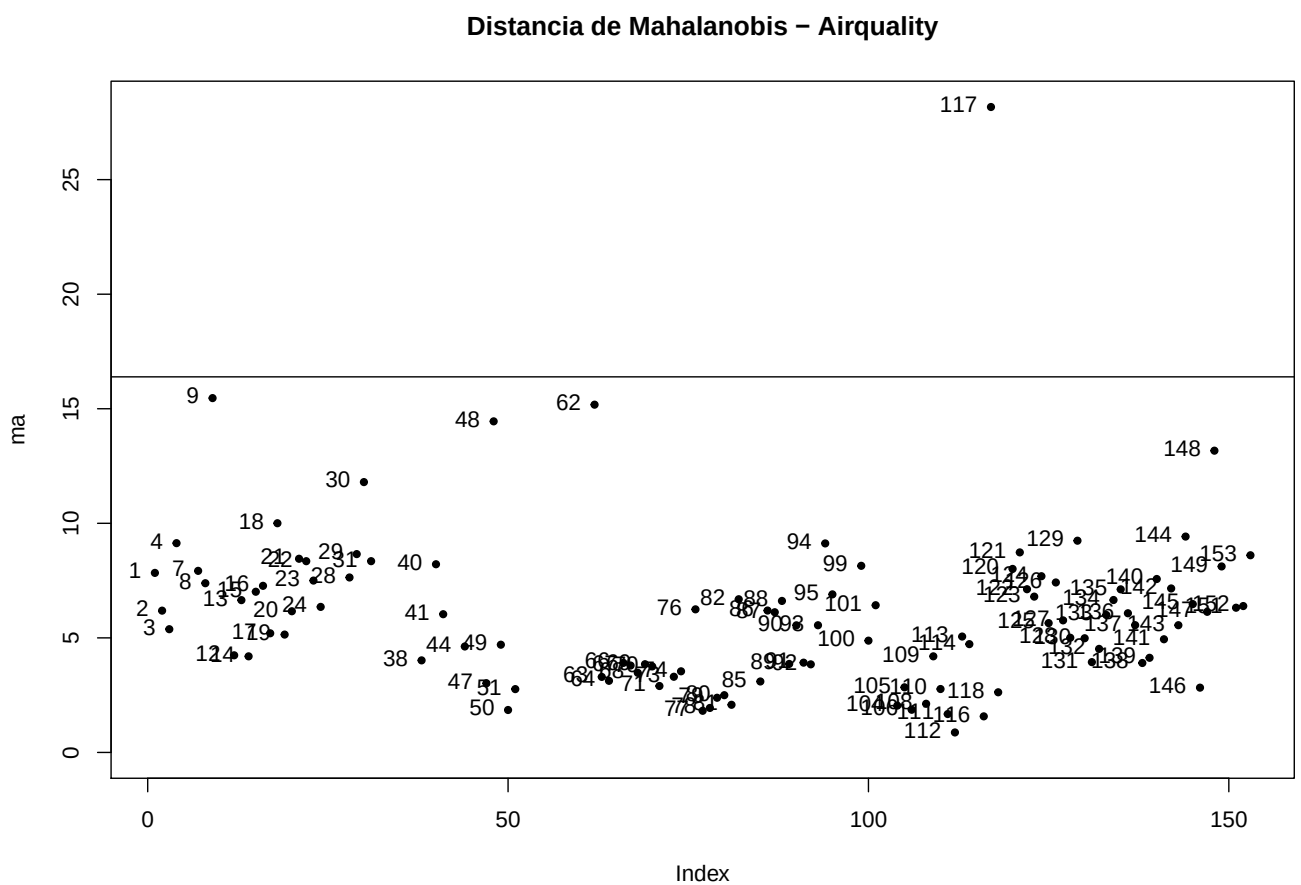
Se establece como límite **razonable** para esta distancia el valor:

$$k + 3\sqrt{2k}$$

donde k representa el número de variables.

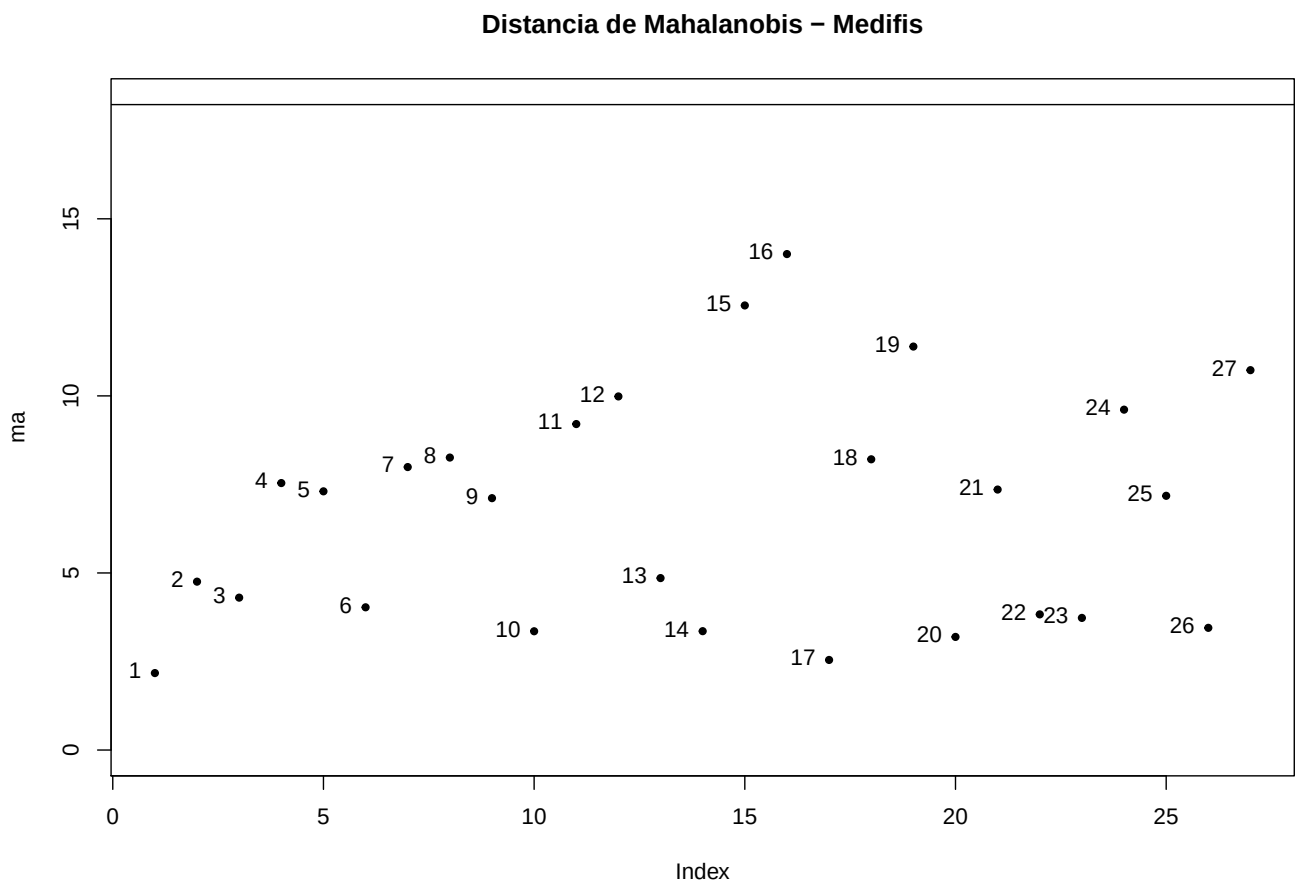
Ejemplo:

```
x <- airquality
ma <- mahalanobis(x, apply(x, 2, mean, na.rm = TRUE), cov(x, use = "na.or.compl
k <- dim(x)[2]  #Número de variables
Lim <- k + 3 * sqrt(k * 2)  #Límite distancia de Mahalanobis
plot(ma, pch = 20, ylim = c(0, max(ma, Lim, na.rm = TRUE)))
text(ma, rownames(x), pos = 2)
abline(h = Lim)
title("Distancia de Mahalanobis - Airquality")
```



Ejemplo:

```
x <- medifis[, -1]
ma <- mahalanobis(x, apply(x, 2, mean, na.rm = TRUE), cov(x, use = "na.or.compl
k <- dim(x)[2]  #Número de variables
Lim <- k + 3 * sqrt(k * 2)  #Límite distancia de Mahalanobis
plot(ma, pch = 20, ylim = c(0, max(ma, Lim)))
text(ma, rownames(x), pos = 2)
abline(h = Lim)
title("Distancia de Mahalanobis - Medifis")
```



5. Datos faltantes

Datos faltantes

En cualquier banco de datos podemos tener **datos faltantes** (valores perdidos).

Antes de realizar cualquier análisis es muy importante **conocer la cumplimentación de las variables**.

Posteriormente será necesario plantearse **cómo es el tipo de valores perdidos** y **la necesidad o no de realizar alguna acción** sobre los valores faltantes.

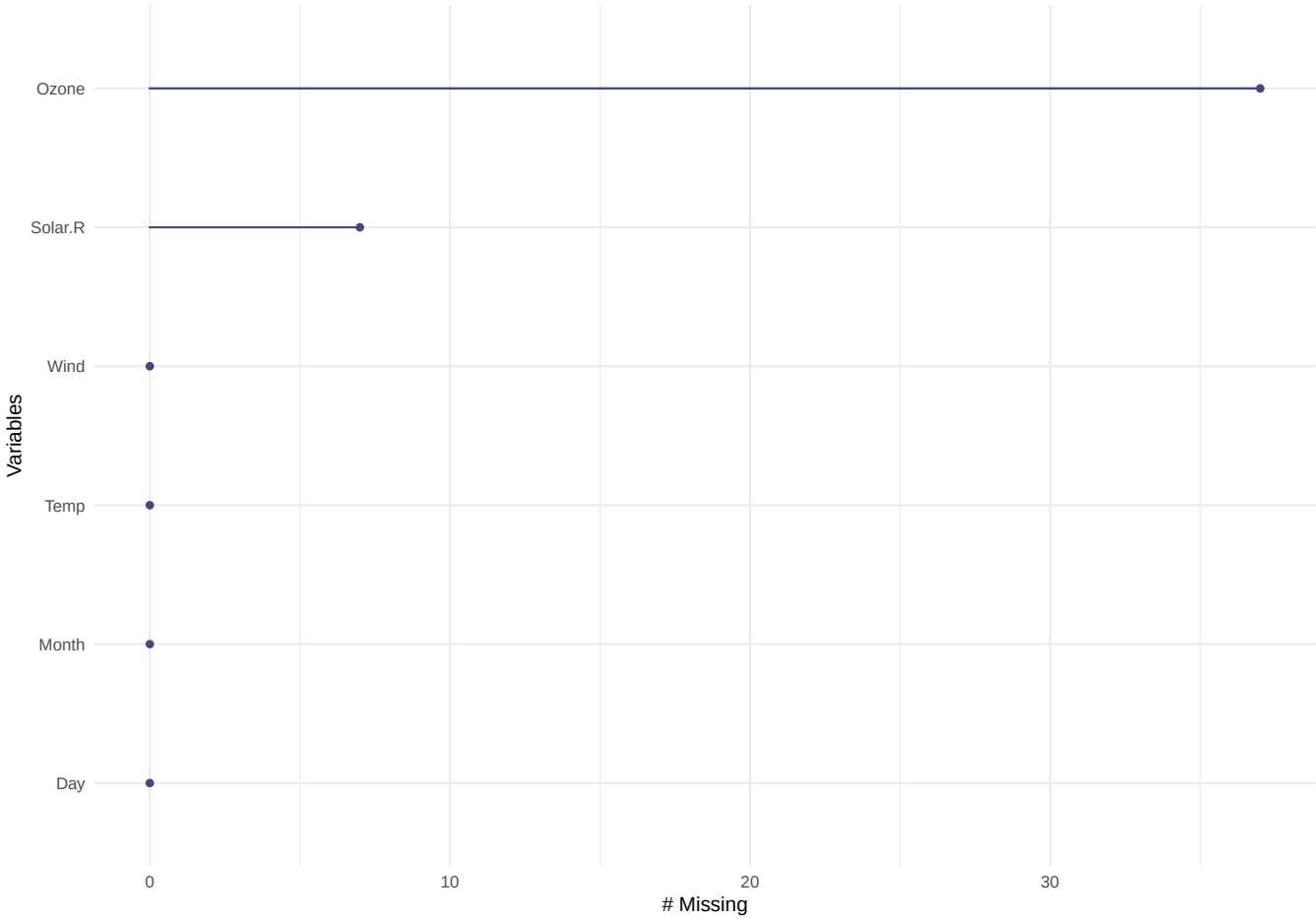
Conocer la cumplimentación de las variables

Existen diferentes librerías en R que contienen funciones que nos ayudan a explorar los valores perdidos de nuestro banco de datos, como por ejemplo la librería **naniar**.

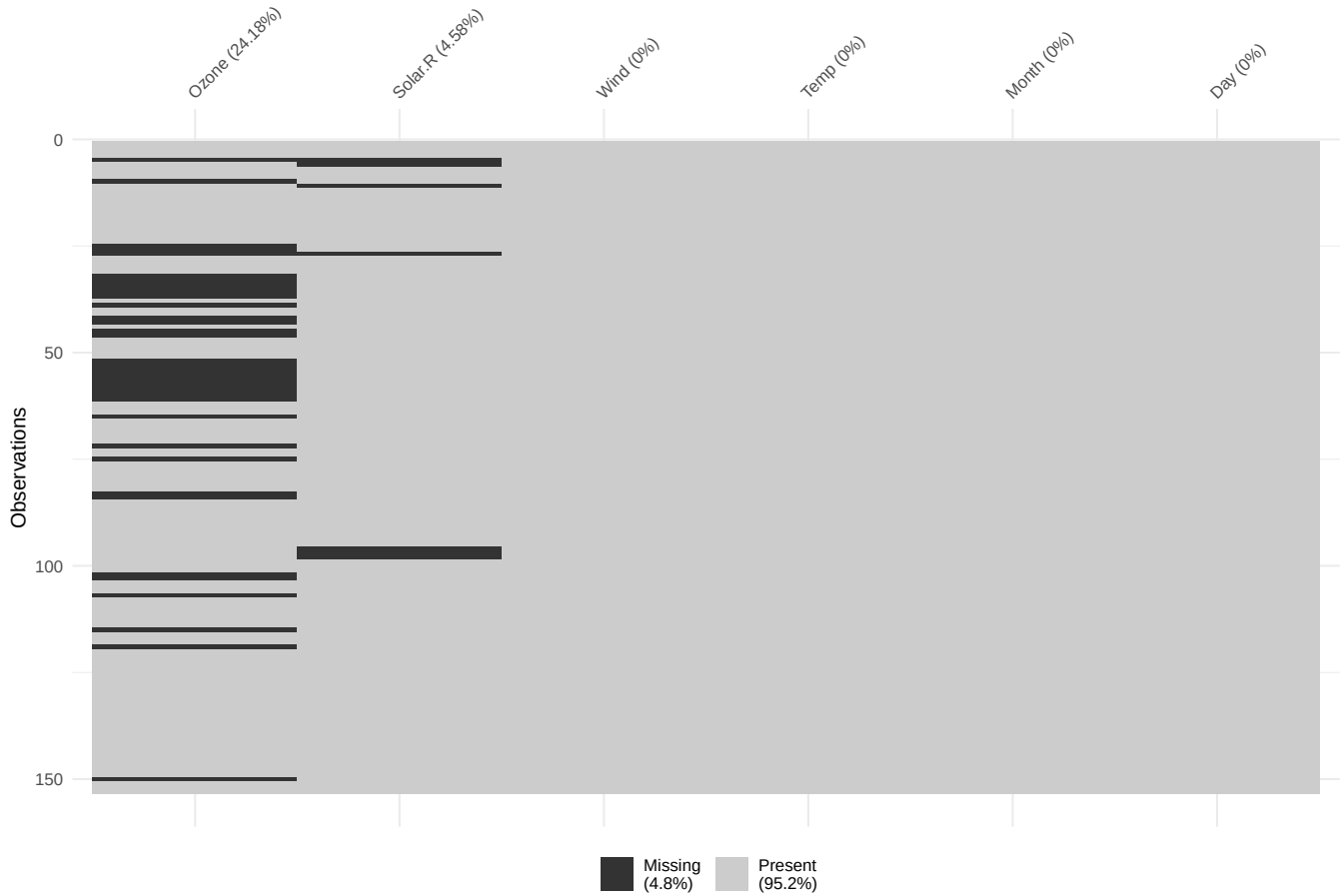
Podemos realizar un gráfico que nos resuma la cumplimentación de las variables de nuestro banco de datos de la siguiente forma:

```
library(naniar)
gg_miss_var(airquality)
vis_miss(airquality)
```

```
library(naniar)
gg_miss_var(airquality)
```



```
vis_miss(airquality)
```



Tipos de datos faltantes

Podemos tener **datos faltantes** por diferentes motivos:

- **MCAR** (Missing Completly At Random)

Pérdida **completamente aleatoria**, la ausencia de un dato no depende de los valores ausentes ni de los no ausentes.

Ejemplo: se borró un dato, el que introdujo el dato puso alguna letra en un valor numérico, no se entiende la grafía del dato original, etc,. . .

Tipos de datos faltantes

Podemos tener **datos faltantes** por diferentes motivos:

- **MAR** (Missing At Random)

La pérdida de un dato en una variable depende de otra variable.

Ejemplo: en una encuesta sobre depresión los hombres tienden a responder menos que las mujeres, aunque no tiene nada que ver con si tienen depresión o no, ni de su nivel...

Tipos de datos faltantes

Podemos tener **datos faltantes** por diferentes motivos:

- **NMAR** (No Missing At Random)

La pérdida depende del valor de la variable (el hecho de no tener valor informa del propio valor).

Ejemplo: una medida extrema hace que el contador no la marque, en una encuesta sobre depresión tras tomar uno de varios tratamientos los que han tomado uno de ellos están más afectados y no quieren responder...

Qué podemos hacer con los datos faltantes

- **Eliminación**:
  - Eliminar los casos que tengan variables con valores perdidos, si perdemos pocos casos y los NA son *MCAR*
  - Eliminar aquellas variables que tienen muchos valores perdidos.

Qué podemos hacer con los datos faltantes

- **Imputación simple**: se puede imputar (sustituir) el dato faltante por una estimación del mismo:
  - Por la media de la variable (luego infraestimaríamos la varianza)
  - Por regresión, glm, u otro modelo (si es MAR)
  - Por un dato o función de datos de objetos similares

Qué podemos hacer con los datos faltantes

Ejemplo de un método de imputación simple

La función *kNN* de la librería *VIM* imputa los valores perdidos de las variables de un caso tomando los *k* casos más **similares** que no tienen valores perdidos. Si la variable que tiene un valor perdido es cuantitativa imputa en el valor perdido el valor medio de esa variable para los *k* casos. Si la variable con el dato perdido es cualitativa, imputaría la categoría más frecuente de esa variable en los *k* casos.

```
library(VIM)
kNN(data, variable, metric, k, dist_var, ...)
```

La *imputación simple* puede sesgar los resultados de nuestro análisis, puesto que **no recoge la incertidumbre** sobre el valor desconocido que realimente ha sido *inventado* (imputado).

Ejemplo de un método de imputación simple

```
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

Ejemplo de un método de imputación simple

```
library(VIM)
new.airquality <- kNN(data = airquality, variable = "Solar.R")
head(new.airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day Solar.R_imp
## 1    41     190  7.4   67     5   1       FALSE
## 2    36     118  8.0   72     5   2       FALSE
## 3    12     149 12.6   74     5   3       FALSE
## 4    18     313 11.5   62     5   4       FALSE
## 5    NA     242 14.3   56     5   5        TRUE
## 6    28     299 14.9   66     5   6        TRUE
```

¿solución?

Imputación múltiple: La imputación de los datos se hace varias veces creando varios posibles bancos de datos. Luego los análisis se hacen para cada banco de datos posible y se ponderan las soluciones para obtener una única solución. *Es necesario buscar el software adecuado para análisis posteriores, la base de datos se convierte en tridimensional.*

En R podemos utilizar las siguientes librerías:

Amelia, mitools, mice, mvnmle, norm, cat, mix, pan, VIM, Hmisc, EMV, monomvm

¿otra solución?

Imputación bayesiana: Los valores faltantes son considerados parámetros por su estatus de *desconocidos*. Si se va a realizar una modelización sobre el banco de datos, los datos faltantes se pueden estimar con el mismo marco de modelización.

Se verá un ejemplo en el módulo de Modelos Jerárquicos Bayesianos.

6 Escala y medidas de disimilaridad

Elección escala

La elección de la **escala** y la medida para valorar la *distancia* o *disimilaridad* entre individuos (o variables) es crucial en el análisis multivariante.

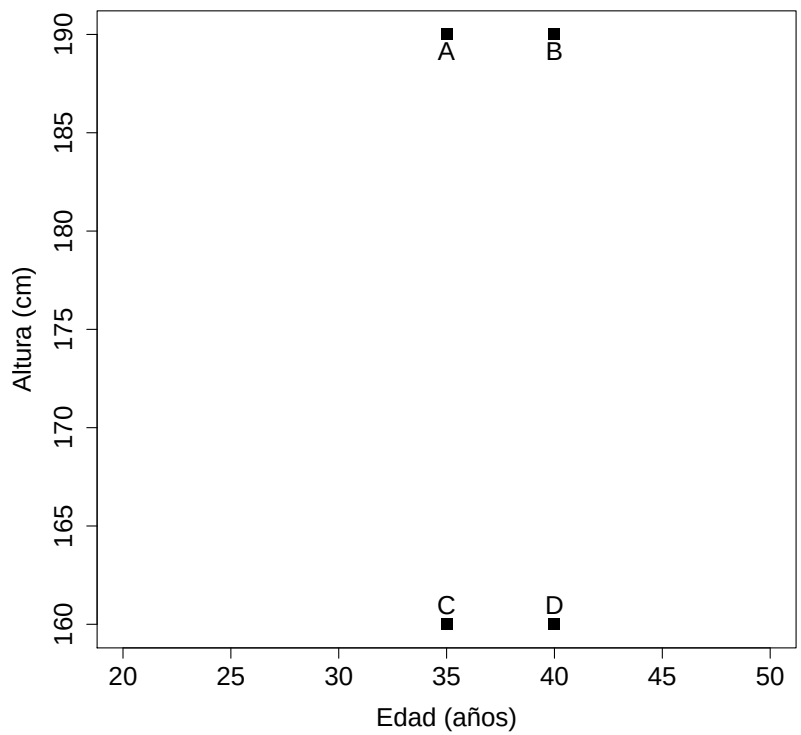
Ejemplo ilustrativo Pensemos en el siguiente banco de datos:

```
##   Persona Edad_años Altura_cm Altura_ft Edad_estand Altura_estand
## 1       A        35       190       6.2          -1             1
## 2       B        40       190       6.2           1             1
## 3       C        35       160       5.2          -1            -1
## 4       D        40       160       5.2           1            -1
```

Elección escala

¿Qué individuos piensas que son más parecidos a la vista de este gráfico?

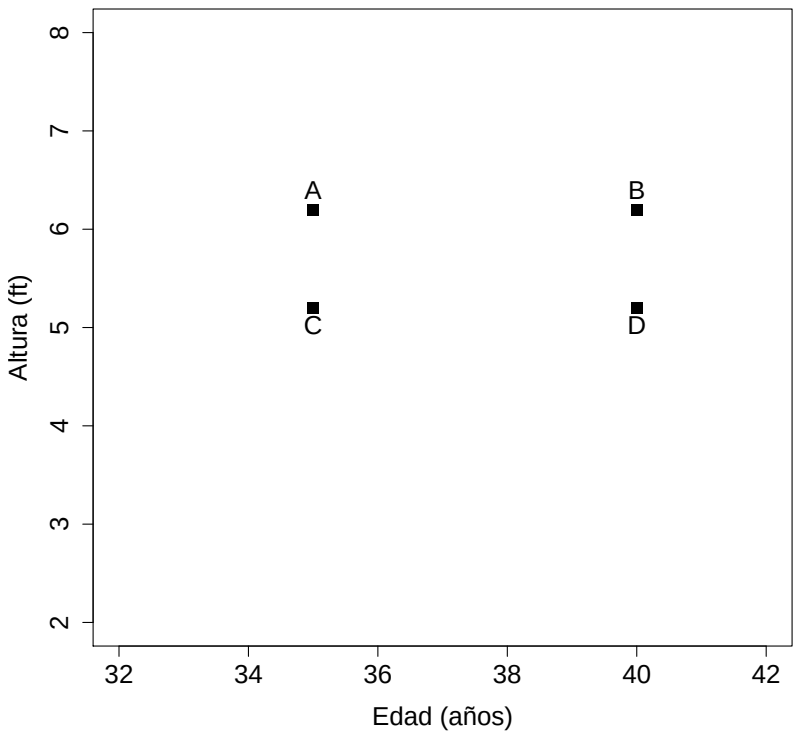
Ejemplo ilustrativo:



Elección escala

¿Y ahora?

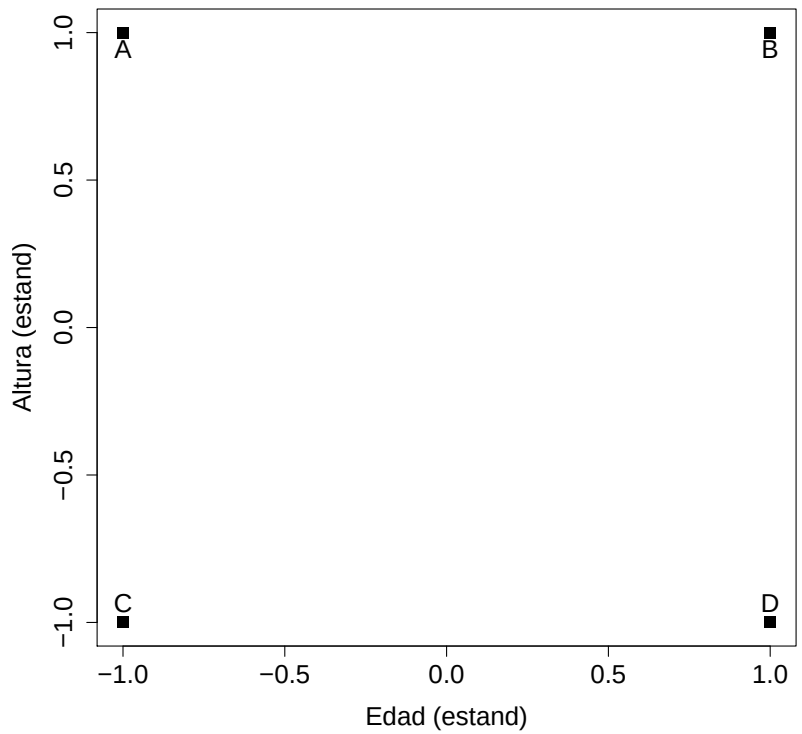
Ejemplo ilustrativo:



Elección escala

¿Y ahora?

Ejemplo ilustrativo:



Elección escala

En ocasiones es necesario transformar las variables cuantitativas para poder compararlas. Las transformaciones más habituales son:

- Pasar a puntuaciones z ($Z_i = \frac{x_i - \overline{x}}{S}$)
- Rango 0 a 1, o de -1 a 1.

Medidas de disimilaridad

Para medir la **disimilaridad** entre dos objetos, dos individuos del banco de datos, y una vez considerada la escala de las variables, tenemos una gran cantidad de funciones que **miden lo diferentes que son dos objetos**.

La elección de la función de **distancia** o **disimilaridad** depende del tipo de variables que tengamos en el banco de datos.

Cuando las variables son **cuantitativas** la distancia más habitual es la distancia **Euclídea**:

$$d(\mathbf{x_i}, \overline{\mathbf{x}}) = \sqrt{\sum_{\mathbf{j=1}}^{\mathbf{k}} (\mathbf{x_{ij}} - \overline{\mathbf{x}})^{\mathbf{2}}}$$

Medidas de disimilaridad

Distancias para variables **cuantitativas** que pueden ser apropiadas según la ocasión:

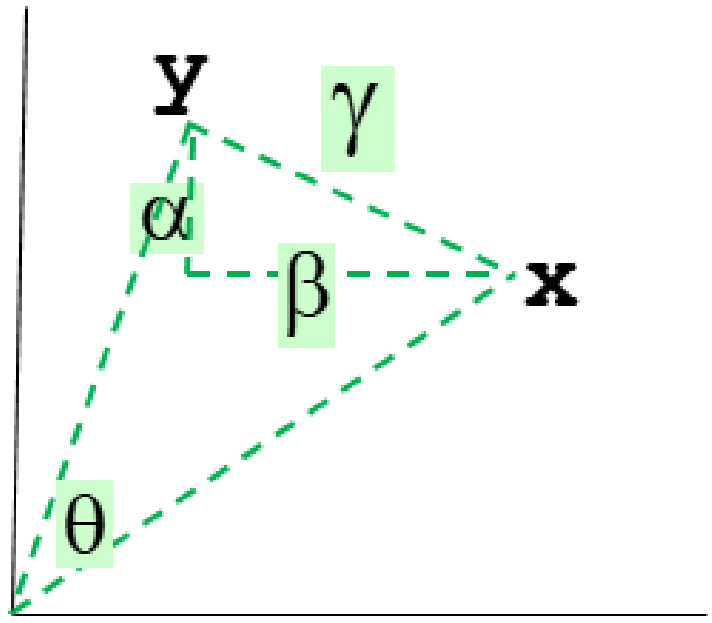
- **Minkowski**

$$d(\mathbf{x_i}, \overline{\mathbf{x}}) = \left(\sum_{\mathbf{j=1}}^{\mathbf{k}} |\mathbf{x_{ij}} - \overline{\mathbf{x}}|^{\lambda} \right)^{\frac{\mathbf{1}}{\lambda}}$$

- **Euclídea** $(\lambda = 2)$
- **City block (Manhattan)** $(\lambda = 1)$

$$d(\mathbf{x_i}, \overline{\mathbf{x}}) = \sum_{\mathbf{j=1}}^{\mathbf{k}} |\mathbf{x_{ij}} - \overline{\mathbf{x}}|$$

- Ilustramos algunas distancias:



γ = ***Distancia Euclídea***

β = ***Distancia Chebychev***

$\alpha + \beta$ = ***Distancia Ciudad***

$\cos(\theta)$ = ***Similitud Coseno***

Medidas de disimilaridad

Para variables **binarias**$(0/1)$ se suelen emplear otro tipo de distancias según el significado de los valores de la variable. Por ejemplo, para variables **binarias** cuyos valores representan **ausencia/presencia** es muy utilizado el **coeficiente de Jaccard**:

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}	V_{12}
O_1	0	0	1	0	1	1	1	0	1	0	0	1
O_2	0	1	1	0	0	1	1	0	0	0	1	0

		O_2	
		1	0
O_1	1	a=3	b=3
	0	c=2	d=4

el **coeficiente de disimilitud de Jaccard** se calcula como:
$1 - \frac{a}{a+b+c} = 1 - \frac{3}{8} = \frac{5}{8}$

Representa el número de veces en que se produce presencia de las dos variables en relación al número de veces en que hay presencia de una, de la otra o de las dos variables.

Medidas de disimilaridad

Sobre estas ideas de la elección de la medida de distancia adecuada en la comparación de los individuos y su similaridad o disimilaridad profundizaremos en la sesión de **análisis de agrupamiento**.

7. Breve guía de preparación de los datos.

Breve guía de preparación de los datos.

Antes de comenzar con cualquier análisis debes tener en cuenta algunas cuestiones importantes que enumeramos a continuación.

- **1. Importa el banco de datos a R para comenzar a trabajar.** El formato de origen del banco de datos en la mayoría de los casos no será un fichero de R (*Rdata*). Identifica el formato de los datos y busca en R la función adecuada para poder leerlo.

¿Se trata de un fichero de texto cuyos campos tienen una amplitud determinada? (Ejemplo: *utils::read.fwf*)

¿Se trata de un fichero de texto cuyos campos están separados por algún caracter (, ; . . .)? (Ejemplo: *utils::read.table*, *read.csv*,. . .)

¿El orgien es Excel, Access, otras bases de datos que requieren conexión ODBC?. . . Debemos buscar las fucniones de R adecuadas.

Breve guía de preparación de los datos.

- **2. Revisa la estructura del banco de datos.** ¿Es la estructura del banco de datos la adecuada? La mayoría de funciones de R esperan que la estructura del banco de datos sea la clásica de *una fila por individuo* y *una columna por variable*.

Si la estructura del banco de datos no es la adecuada debemos preparar el banco de datos en este sentido.

Breve guía de preparación de los datos.

- **3. Conoce el banco de datos.** ¿Conoces las variables que componen el banco de datos? ¿El tipo de variables es el adecuado?

Comprueba si las variables están definidas como numéricas, carácter, factor,. . . según corresponda con los objetivos propuestos. ¿El número de niveles para los factores es adecuado o al ser demasiados convendría agrupar algunos de ellos definiendo un nuevo factor?

¿Hay muchos valores perdidos? En ese caso ¿en alguna de las variables o en todas?

¿Hay valores atípicos?

- **4. Nombres de las filas y las columnas.** Comprueba si las variables tienen un nombre adecuado, que permite identificar cada una de ellas, pues en muchas salidas de funciones de R, numéricas o gráficas, aparecerán con su nombre. Por otro lado, piensa si sería adecuado que las filas tuvieran o no un nombre adecuado, pues en algúnos análisis que se centran en los individuos identifican a los mismos mediante el nombre de cada fila correspondiente.

Conclusiones

Conclusiones

- En cualquier análisis de datos es importante realizar un buen **análisis exploratorio**, tanto **univariante** como **multivariante**.
- El *análisis exploratorio multivariante* nos va a permitir conocer tanto la **relación entre las variables** del banco de datos como la **relación entre individuos**.
- Antes de abordar cualquier análisis más profundo de los datos es importante tener conocimiento de los **valores ausentes** en el banco de datos, así como realizar un estudio tanto univariante como multivariante de los **valores atípicos**.