

# TabulaR - Sesión 2

*00R Team*

*Marzo 2018*

## Índice

|  |           |
|--|-----------|
| <b>1. Manipulación de los datos</b>                        | <b>2</b>  |
| 1.1. Preparando los datos . . . . .                        | 2         |
| 1.2. Transformaciones: códigos a factor . . . . .          | 2         |
| <b>2. Variables cuantitativas</b>                          | <b>3</b>  |
| 2.1. Descripción de las variables . . . . .                | 3         |
| 2.1.1. Descriptivos <i>ad hoc</i> . . . . .                | 4         |
| 2.2. Representación gráfica . . . . .                      | 6         |
| 2.2.1. Gráfico de dispersión . . . . .                     | 6         |
| 2.2.2. Histogramas y distribución de frecuencias . . . . . | 6         |
| <b>3. Variables cualitativas</b>                           | <b>8</b>  |
| 3.1. Tablas y gráficos de barras . . . . .                 | 8         |
| 3.1.1. Tabla de proporciones . . . . .                     | 9         |
| 3.2. Tabular con tres variables . . . . .                  | 11        |
| <b>4. Variables cuantitativas y factores</b>               | <b>12</b> |
| 4.1. Muchas variables. Tabular por estadísticos. . . . .   | 12        |
| 4.2. <i>Boxplot</i> . . . . .                              | 14        |
| <b>5. La librería <code>printr</code></b>                  | <b>15</b> |

# 1. Manipulación de los datos

Trabajaremos con datos biométricos de alumnos de la Licenciatura de Biología, Murcia en el curso 2003-04. Para cada individuo se anotaron los valores correspondientes a:

- Grupo: Grupo de prácticas al que pertenece el individuo
- Peso: Peso en kilogramos
- Altura: Estatura en cm
- Pie: Talla de calzado
- Hombros: Anchura de hombros en cm
- Brazos: Longitud del brazo en cm
- Caderas: Perímetro de caderas
- Sexo: 1 para mujeres y 2 para hombres
- Ojos: Color de ojos, 1 para los oscuros y 2 para los claros
- Tipo: Aspecto de los individuos, según ellos mismos, 1 atlético, 2 delgados y 3 pesados.

## 1.1. Preparando los datos

Leemos el fichero de datos

```
df <- read.table( "http://gauss.inf.um.es/datos/biom2003.dat", sep = ",",
                  header = TRUE )
```

Verificamos una lectura adecuada con un simple vistazo de las primeras filas de df:

```
head( df )
```

```
##      Grupo Peso Altura Pie Hombros Brazos Caderas Sexo Ojos Tipo
##  1      1   60   163  37    41     68     95    1    1    2
##  2      1   52   166  37    37     70     87    1    2    1
##  3      1   61   172  39    39     69     91    1    2    1
##  4      1   73   181  43    50     78    101    2    2    1
##  5      1   53   172  39    39     72     89    1    1    1
##  6      1   63   169  40    37     66     96    1    2    1
```

Comprobamos que el número de filas y columnas es adecuado:

```
dim( df )
```

```
## [1] 98 10
```

## 1.2. Transformaciones: códigos a factor

En muchos casos es necesaria la transformación de variables, por ejemplo, en esta matriz de datos df encontramos tres variables cualitativas que se tratan como cuantitativas: Sexo, Ojos y Tipo.

```
df$Sexo <- factor( df$Sexo, levels = c( 1, 2 ),
                  labels = c( "mujer", "hombre" ) )

df$Ojos <- factor( df$Ojos, levels = c( 1, 2 ),
                  labels = c( "oscuros", "claros" ) )

df$Tipo <- factor( df$Tipo, levels = c( 1, 2, 3 ),
                  labels = c( "atlético", "delgado", "pesado" ) )
```

Comprobamos que tenemos los datos en el formato que queríamos:

```
str( df )

## 'data.frame': 98 obs. of 10 variables:
## $ Grupo : int 1 1 1 1 1 1 1 1 1 1 ...
## $ Peso : int 60 52 61 73 53 63 55 56 64 72 ...
## $ Altura : int 163 166 172 181 172 169 170 169 180 178 ...
## $ Pie : int 37 37 39 43 39 40 38 42 41 43 ...
## $ Hombros: int 41 37 39 50 39 37 43 40 41 43 ...
## $ Brazos : int 68 70 69 78 72 66 74 70 78 80 ...
## $ Caderas: int 95 87 91 101 89 96 97 90 101 97 ...
## $ Sexo : Factor w/ 2 levels "mujer","hombre": 1 1 1 2 1 1 1 2 1 2 ...
## $ Ojos : Factor w/ 2 levels "oscuros","claros": 1 2 2 2 1 2 2 2 1 1 ...
## $ Tipo : Factor w/ 3 levels "atlético","delgado",...: 2 1 1 1 1 1 1 1 1 1 ...
```

Llegado a este punto, podíamos pensar en guardar un conjunto de datos con las variables *anotadas* en un fichero `.Rdata`, de forma que a partir de ese momento podríamos trabajar directamente con con ese conjunto de datos, sin tener que realizar de nuevo la codificación de las variables.

```
save.image( "biom2003Anotado.RData" ) # guardar el conjunto de datos
load("biom2003Anotado.RData") # leer el conjunto de datos anotado
```

## 2. Variables cuantitativas

### 2.1. Descripción de las variables

Una vez tenemos una lectura adecuada de la tabla de datos, realizaremos una serie de descripciones y tabulaciones para comprobar que los valores son razonablemente correctos.

```
summary( df )

##      Grupo      Peso      Altura      Pie
##  Min.   :1.000   Min.   :40.00   Min.   :150.0   Min.   :35.00
##  1st Qu.:2.000   1st Qu.:56.00   1st Qu.:163.0   1st Qu.:38.00
##  Median :3.000   Median :61.50   Median :170.0   Median :39.00
##  Mean   :2.582   Mean   :63.16   Mean   :169.7   Mean   :39.61
##  3rd Qu.:3.000   3rd Qu.:70.00   3rd Qu.:175.8   3rd Qu.:42.00
##  Max.   :4.000   Max.   :95.00   Max.   :188.0   Max.   :47.00
##      Hombros      Brazos      Caderas      Sexo      Ojos
##  Min.   :32.00   Min.   :63.00   Min.   : 73.00   mujer :66   oscuros:18
##  1st Qu.:38.00   1st Qu.:69.00   1st Qu.: 90.00   hombre:32   claros :80
##  Median :40.00   Median :71.00   Median : 95.00
##  Mean   :40.86   Mean   :71.55   Mean   : 95.21
##  3rd Qu.:44.00   3rd Qu.:74.75   3rd Qu.:100.00
##  Max.   :53.00   Max.   :86.00   Max.   :113.00
##      Tipo
##  atlético:67
##  delgado :17
##  pesado  :14
##
##
```

¿Cómo trata a las variables categórica? ¿Qué estadísticos nos devuelve?

Para aquellos valores son numéricos cabe:

```
apply( df[, 2:7], 2, summary )
```

```
##           Peso  Altura      Pie Hombros  Brazos  Caderas
## Min.      40.00000 150.0000 35.00000 32.00000 63.00000 73.00000
## 1st Qu.  56.00000 163.0000 38.00000 38.00000 69.00000 90.00000
## Median   61.50000 170.0000 39.00000 40.00000 71.00000 95.00000
## Mean     63.16327 169.7143 39.61224 40.85714 71.55102 95.21429
## 3rd Qu.  70.00000 175.7500 42.00000 44.00000 74.75000 100.00000
## Max.     95.00000 188.0000 47.00000 53.00000 86.00000 113.00000
```

Si sólo queremos los descriptivos de las mujeres para las variables Peso, Altura y Caderas:

```
apply( df[df$Sexo == "mujer" , c(2:3,7)],
      2,
      summary )
```

```
##           Peso  Altura  Caderas
## Min.      40.00000 150.0000 73.00000
## 1st Qu.  53.00000 161.2500 89.00000
## Median   58.00000 165.0000 92.50000
## Mean     58.48485 165.6364 93.90909
## 3rd Qu.  62.75000 170.0000 98.75000
## Max.     79.00000 182.0000 113.00000
```

### 2.1.1. Descriptivos *ad hoc*

Podríamos crear *a mano* nuestras propias tablas de descriptivos con aquellos estadísticos que nos interesen utilizando la función `tapply()`:

```
# Peso medio según el género
tapply( df$Peso, df$Sexo, mean )
```

```
##      mujer  hombre
## 58.48485 72.81250
```

Creamos los descriptivos que nos interesan

```
# Creamos los descriptivos
media <- tapply ( df$Peso, df$Sexo, mean )
sd <- tapply ( df$Peso, df$Sexo, sd )
mediana <- tapply ( df$Peso, df$Sexo, median )
n <- tapply ( df$Peso, df$Sexo, length )
```

```
# los unimos en un único dataframe
cbind( media, sd, mediana, n )
```

```
##           media      sd mediana  n
## mujer 58.48485 7.614243  58.0 66
## hombre 72.81250 7.896253  71.5 32
```

Quizá sea algo tedioso repetir el código anterior para cada una de nuestras variables. Para evitarnos esto, podemos construir, de manera sencilla, una función *ad hoc* para estadísticos descriptivos.

```
f_descriptivos <- function( variable, factor ) {
  media <- tapply ( variable, factor, mean )
  sd <- tapply ( variable, factor, sd )
  mediana <- tapply ( variable, factor, median )
  n <- tapply ( variable, factor, length )
  cbind( media, sd, mediana, n )
}
```

Podemos aplicar nuestra función `f_descriptivos()` a los datos de `biom`

```
f_descriptivos( df$Pie, df$Sexo )

##           media          sd mediana  n
## mujer  38.10606 1.579959      38 66
## hombre 42.71875 1.954884      42 32
```

```
f_descriptivos( df$Altura, df$Tipo )

##           media          sd mediana  n
## atlético 168.9403 8.502906   170.0 67
## delgado  166.2941 6.835246   166.0 17
## pesado   177.5714 7.693297   176.5 14
```

y presentarlos mediante una tabla “elegante” mediante la función `kable()` de la librería `knitr`.

```
des <- f_descriptivos(df$Hombros, df$Ojos)

# library( knitr )
kable( des, caption = "Descriptivos adhoc",
       align = c('lcc'),
       col.names = c("Media", "SD", "Mediana", "n"),
       row.names = TRUE,
       digits = 1,
       format.args = list( decimal.mark = "," )
     )
```

Cuadro 1: Descriptivos adhoc

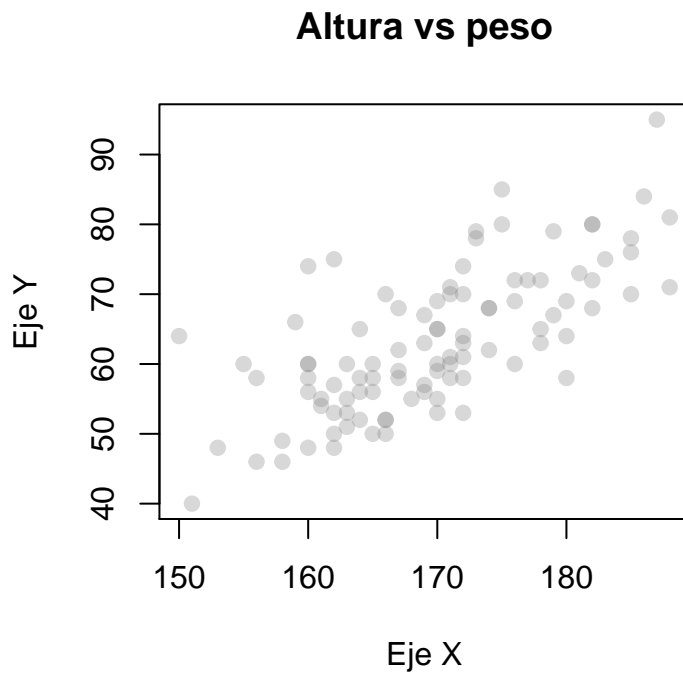
|         | Media | SD  | Mediana | n  |
|---------|-------|-----|---------|----|
| oscuros | 39,5  | 3,8 | 40      | 18 |
| claros  | 41,2  | 4,8 | 40      | 80 |

## 2.2. Representación gráfica

### 2.2.1. Gráfico de dispersión

Realizamos el gráfico de dispersión de dos variables:

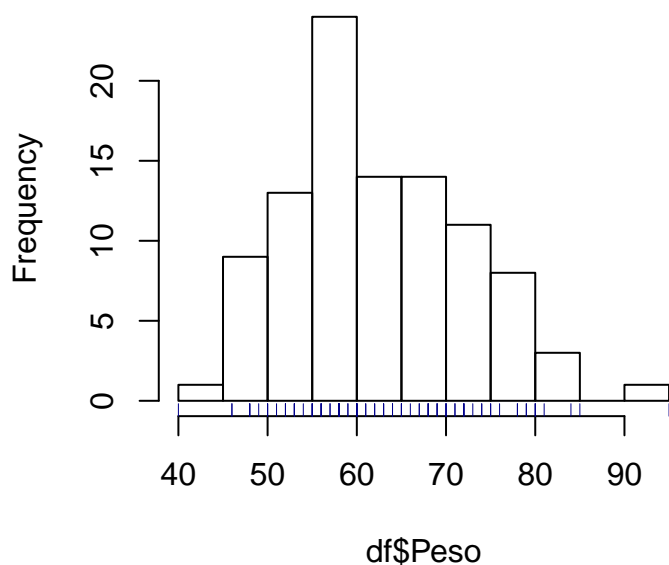
```
plot( df$Altura, df$Peso, pch = 20,  
      main = "Altura vs peso",  
      xlab = "Eje X", ylab= "Eje Y",  
      col = rgb( .5, .5, .5, 0.3 ), cex = 1.5 )
```



### 2.2.2. Histogramas y distribución de frecuencias

```
hist( df$Peso, main = "Histograma de Peso" )  
rug( df$Peso, col = "darkblue" )
```

## Histograma de Peso

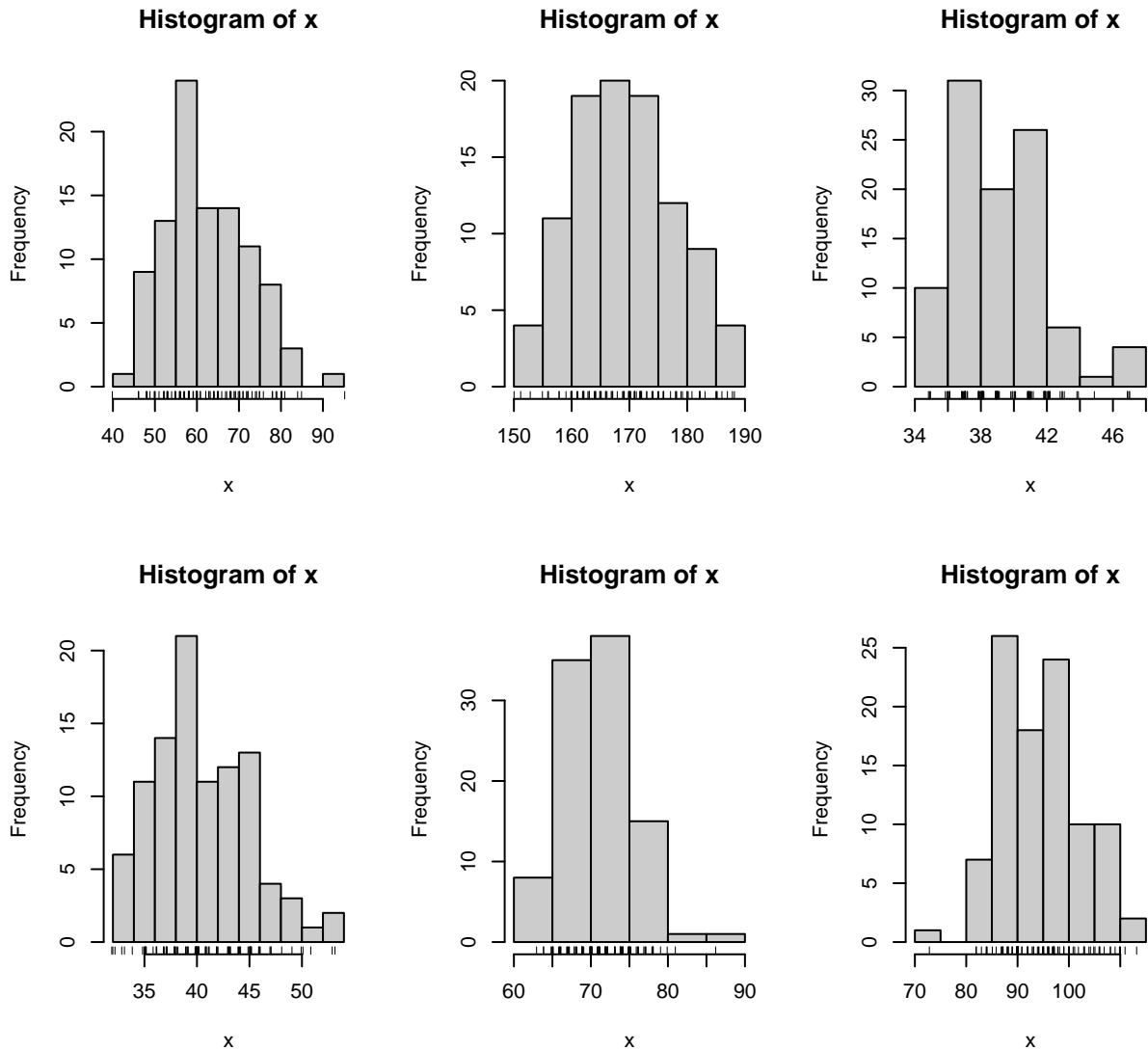


Podemos personalizar el gráfico definiendo nuestra propia función:

```
miHist <- function( x) {  
  hist ( x, col = "grey80" )  
  rug( jitter( x ) )  
}
```

Y aplicar nuestra función `miHist()` a las variables del conjunto de datos, que representamos en un sólo *device* gráfico :

```
par( mfrow = c( 2, 3 ) ) # defino cómo va a ser el device  
apply( df[ , 2:7 ], 2, miHist ) # ejecuto mi función las vbles cuantitativas
```



```
par( mfrow = c(1, 1) ) # restaura el device original
```

Esto nos permite analizar de un vistazo la distribución de todas las variables.

### 3. Variables cualitativas

#### 3.1. Tablas y gráficos de barras

Imaginemos ahora que queremos analizar la relación entre dos variables cualitativas.

```
table( df$Sexo, df$Ojos )
```

```
##
##          oscuros claros
##  mujer      15     51
##  hombre       3     29
```

```
table( df$Ojos, df$Tipo, dnn = c("Color ojos", "Complexion") ) # nombres dimensiones
```



```
##           Complexion
## Color ojos atlético delgado pesado
##   oscuros      12      6      0
##   claros       55     11     14
```

```
library( knitr )
```

```
kable( table( df$Sexo, df$Tipo ),
       caption = "Color de ojos según el género",
       align = c('lcc'))
```

Cuadro 2: Color de ojos según el género

|        | atlético | delgado | pesado |
|--------|----------|---------|--------|
| mujer  | 49       | 14      | 3      |
| hombre | 18       | 3       | 11     |

Nos podría interesar también saber los totales

```
addmargins( table( df$Sexo, df$Ojos ) ) # total de filas y columnas
```

```
##
##           oscuros claros Sum
##   mujer         15     51  66
##   hombre         3     29  32
##   Sum            18     80  98
```

```
addmargins( table( df$Sexo, df$Ojos ), 1 ) # total para cada columna
```

```
##
##           oscuros claros
##   mujer         15     51
##   hombre         3     29
##   Sum            18     80
```

```
kable( addmargins( table( df$Sexo, df$Ojos ), 2 ),
       col.names = c("oscuros", "claros", "Total")) # total para cada fila
```

|        | oscuros | claros | Total |
|--------|---------|--------|-------|
| mujer  | 15      | 51     | 66    |
| hombre | 3       | 29     | 32    |

Podéis encontrar más opciones para representar tablas en esta entrada del blog del SAE blog

### 3.1.1. Tabla de proporciones

Utilizamos la función `prop.table ( x, margin )` que tiene dos argumentos:

- `x`: tabla creada anteriormente con la función `table()`
- `margin`: si se calcula la proporción para todo o por filas o columnas
  - Nulo, calcula  $x/\text{suma}(x)$ , como en el ejemplo anterior.
  - 1, calcula la proporción por filas.
  - 2, calcula la proporción por columnas.

```
tt <- table( df$Sexo, df$Ojos )
prop.table( tt ) # porcentaje celdas sobre el total
```

```
##
##           oscuros   claros
##  mujer 0.15306122 0.52040816
##  hombre 0.03061224 0.29591837
```

```
kable( prop.table( tt, 1 ), digits = 2 ) # porcentajes por filas
```

|        | oscuros | claros |
|--------|---------|--------|
| mujer  | 0.23    | 0.77   |
| hombre | 0.09    | 0.91   |

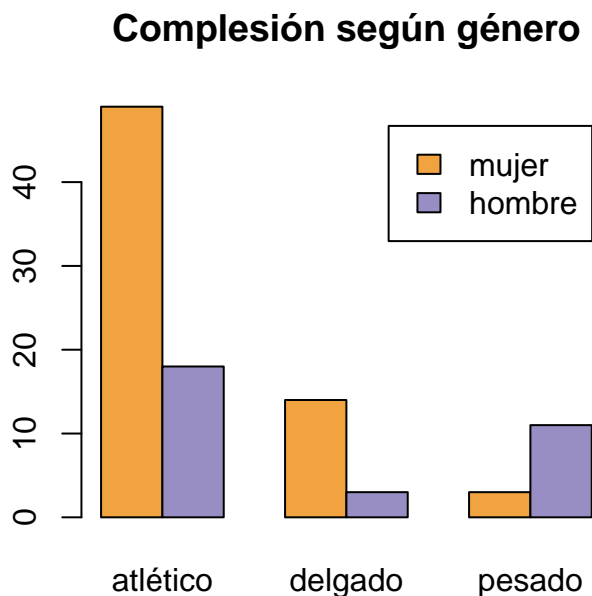
```
kable( prop.table( tt, 2 ), digits = 2,
caption = "Distribución color de ojos por género") # porcentajes por columnas
```

Cuadro 5: Distribución color de ojos por género

|        | oscuros | claros |
|--------|---------|--------|
| mujer  | 0.83    | 0.64   |
| hombre | 0.17    | 0.36   |

Podemos utilizar la tabla generada para crear un gráfico de barras:

```
# Barras agrupadas
barplot( table( df$Sexo, df$Tipo ), beside = TRUE,
col = c( "#f1a340", "#998ec3" ),
legend = levels( df$Sexo ),
main = "Compleción según género" )
```



### 3.2. Tabular con tres variables

Imaginemos ahora que queremos una tabla de tres dimensiones

```
table( df$Sexo, df$Ojos, df$Tipo )
```

```
## , , = atlético
##
##          oscuros claros
##  mujer         11    38
##  hombre         1    17
##
## , , = delgado
##
##          oscuros claros
##  mujer          4    10
##  hombre         2     1
##
## , , = pesado
##
##          oscuros claros
##  mujer          0     3
##  hombre         0    11
```

Obtenemos una tabla que no es muy bonita. Para este caso podemos usar la función `ftable()` con la que podemos crear lo que se llama una tabla de contingencia plana.

```
ftable( df$Sexo, df$Ojos, df$Tipo )
```

```
##          atlético delgado pesado
##
##  mujer  oscuros         11     4     0
##          claros         38    10     3
##  hombre  oscuros         1     2     0
##          claros         17     1    11
```

Lamentablemente esta tabla no la podemos presentar mediante `kable()` pues la deforma completamente.

```
tt <- ftable( df$Sexo, df$Ojos, df$Tipo )
kable( tt )
```

---

|        |         |          |    |
|--------|---------|----------|----|
| mujer  | oscuros | atlético | 11 |
| hombre | oscuros | atlético | 1  |
| mujer  | claros  | atlético | 38 |
| hombre | claros  | atlético | 17 |
| mujer  | oscuros | delgado  | 4  |
| hombre | oscuros | delgado  | 2  |
| mujer  | claros  | delgado  | 10 |
| hombre | claros  | delgado  | 1  |
| mujer  | oscuros | pesado   | 0  |
| hombre | oscuros | pesado   | 0  |
| mujer  | claros  | pesado   | 3  |
| hombre | claros  | pesado   | 11 |

---

## 4. Variables cuantitativas y factores

Es posible realizar un gran número de combinaciones en este caso, por ejemplo un cálculo condicionado de estadísticos:

```
tapply( df$Altura, list(df$Sexo, df$Ojos), mean )
```

```
##           oscuros   claros
## mujer  167.6667 165.0392
## hombre 175.3333 178.4138
```

Podríamos querer presentar esta tabla como resultado:

```
tt <- tapply( df$Altura,
              list( df$Sexo, df$Ojos ),
              mean,
              simplify = TRUE )

kable( tt,
       digits = 3,
       align = c('c1l1'),
       caption = "Media de altura según género y color de ojos")
```

Cuadro 7: Media de altura según género y color de ojos

|        | oscuros | claros  |
|--------|---------|---------|
| mujer  | 167.667 | 165.039 |
| hombre | 175.333 | 178.414 |

¿Qué ocurre si queremos la media de más de una variable?

```
tapply( df[,2:7], list(df$Sexo, df$Ojos ), mean )
```

¿Y si queremos partir por tres variables cualitativas?

```
tapply( df$Pie, list( df$Sexo, df$Ojos, df$Tipo ), median )
```

### 4.1. Muchas variables. Tabular por estadísticos.

El cálculo condicionado del estadístico también puede resultar de ayuda para este proceso de descripción y localización de errores. La función `aggregate()` nos calculará los estadísticos para un conjunto de variables, por ejemplo:

```
aggregate( df[ , 2:7], list( df$Sexo ), mean )
```

```
##   Group.1  Peso  Altura  Pie Hombros  Brazos  Caderas
## 1  mujer 58.48485 165.6364 38.10606 38.53030 69.77273 93.90909
## 2  hombre 72.81250 178.1250 42.71875 45.65625 75.21875 97.90625
```

Puede obtenerse fácilmente los valores del rango para cada una de las variables:

```
aggregate( df[ , 2:4], list( Sexo = df$Sexo ), sd )
```

```
##   Sexo  Peso  Altura  Pie
## 1  mujer 7.614243 6.815449 1.579959
## 2  hombre 7.896253 5.712183 1.954884
```

Las variables cualitativas pueden ser varias, consiguiendo el detalle descriptivo que nos interese:

```
aggregate( df[ , 2:4],
           list( Sexo = df$Sexo,
                `Color ojos` = df$Ojos ),
           median )
```

```
##      Sexo Color ojos Peso Altura Pie
## 1  mujer   oscuros   58   166  38
## 2 hombre   oscuros   78   175  42
## 3  mujer   claros   58   165  38
## 4 hombre   claros   71   178  42
```

Con tres variables categóricas:

```
aggregate( df[ , 2:7],
           list( Sexo = df$Sexo,
                `Color ojos` = df$Ojos,
                Tipo = df$Tipo ),
           median )
```

```
##      Sexo Color ojos      Tipo Peso Altura Pie Hombros Brazos Caderas
## 1  mujer   oscuros atlético 55.0  168.0 38.0   40.0   72.0   92.0
## 2 hombre   oscuros atlético 72.0  178.0 43.0   43.0   80.0   97.0
## 3  mujer   claros  atlético 57.0  165.0 38.0   38.0   69.5   90.0
## 4 hombre   claros  atlético 69.0  178.0 42.0   44.0   75.0   96.0
## 5  mujer   oscuros  delgado 62.5  163.5 37.5   39.5   68.0  104.5
## 6 hombre   oscuros  delgado 81.5  174.0 42.0   46.0   74.5  107.0
## 7  mujer   claros  delgado 66.5  165.0 38.5   40.5   70.0  101.0
## 8 hombre   claros  delgado 79.0  179.0 42.0   45.0   70.0  100.0
## 9  mujer   claros  pesado  58.0  167.0 39.0   38.0   69.0   88.0
## 10 hombre  claros  pesado  76.0  182.0 42.0   47.0   75.0   99.0
```

Para presentar estos datos en una tabla:

```
# Guardamos la tabla en una variable
tt <- aggregate( df[ , 2:4],
                 list( Sexo = df$Sexo,
                      `Color ojos` = df$Ojos ),
                 median )

# Presentamos la tabla
kable( tt,
       digits = 3,
       caption = "Media de altura según género y color de ojos")
```

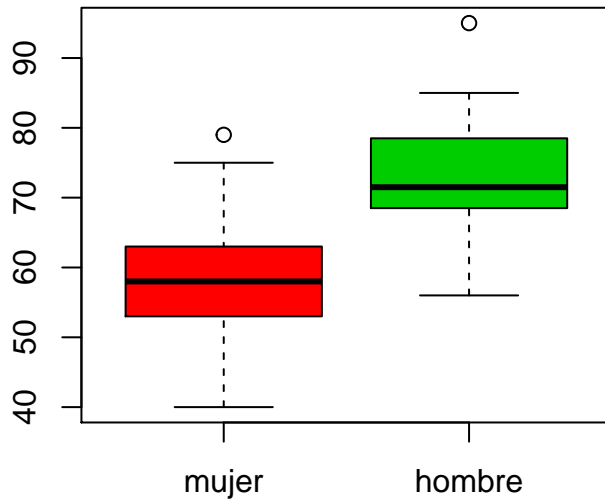
Cuadro 8: Media de altura según género y color de ojos

| Sexo   | Color ojos | Peso | Altura | Pie |
|--------|------------|------|--------|-----|
| mujer  | oscuros    | 58   | 166    | 38  |
| hombre | oscuros    | 78   | 175    | 42  |
| mujer  | claros     | 58   | 165    | 38  |
| hombre | claros     | 71   | 178    | 42  |

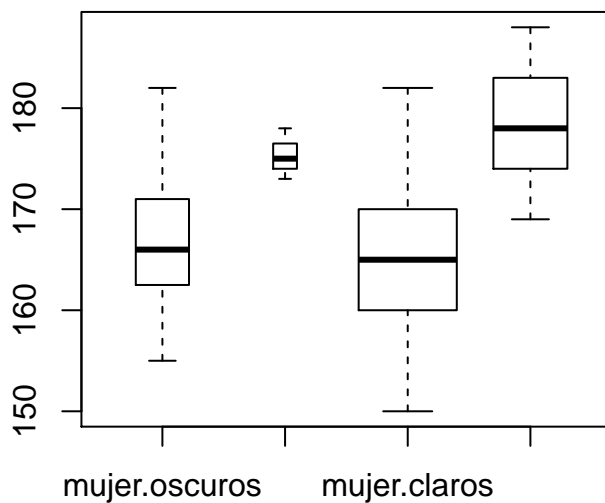
## 4.2. *Boxplot*

La función `boxplot` representa los valores extremos fuera de los rangos “aceptables” (utilizando un punto), considerándolos sospechosos de ser *outliers* o valores espurios, que deben confirmarse necesariamente por si se trata de errores de mecanización o realmente valores muy desviados.

```
boxplot( df$Peso ~ df$Sexo, col=2:3 )
```

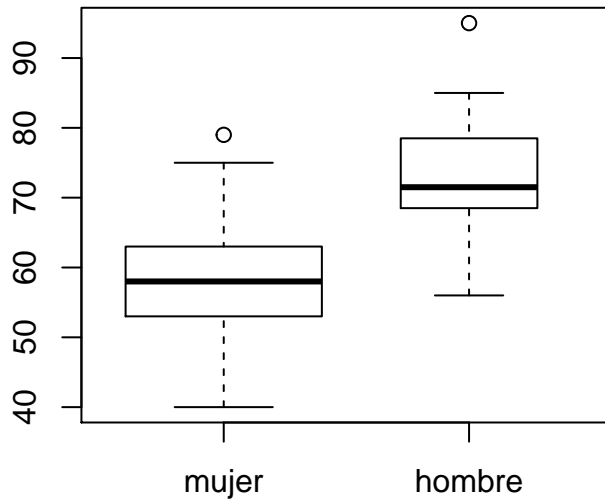


```
# Anchura proporcional a número de elementos en cada grupo.  
boxplot( df$Altura ~ df$Sexo + df$Ojos, varwidth = TRUE )
```



¿Cómo podemos saber cuales son los valores atípicos?

```
bx <- boxplot( df$Peso ~ df$Sexo, varwidth = TRUE )
```



```
str( bx )
```

```
## List of 6
## $ stats: integer [1:5, 1:2] 40 53 58 63 75 56 68.5 71.5 78.5 85
## $ n : num [1:2] 66 32
## $ conf : num [1:2, 1:2] 56.1 59.9 68.7 74.3
## $ out : num [1:2] 79 95
## $ group: num [1:2] 1 2
## $ names: chr [1:2] "mujer" "hombre"
```

```
bx$out
```

```
## [1] 79 95
```

## 5. La librería `printr`

A lo largo del documento nos hemos encontrado con muchos objetos que nos gustaría presentar tablas resultado en nuestros trabajos. Una forma de hacerlo, y que funciona para la mayoría de los objetos, es utilizar la función `kable()` del paquete `knitr`, que incluso veremos más adelante que podemos perfeccionar con la librería `kableExtra`.

Ahora bien, si lo que queremos es hacer un informe rápido al mismo tiempo que presentamos los resultados de una forma “decente” lo podemos hacer mediante la librería `printr`, desarrollada por *Yihui Xie*, autor de `knitr`, que busca mejorar la salida de consola que utiliza por defecto dicha librería.

Puedes encontrar una *Viñeta* de uso de la librería `printr` aquí

```
library(printr)
```

```
apply( df[, 2:7], 2, summary, digits = 3 )
```

|         | Peso | Altura | Pie  | Hombros | Brazos | Caderas |
|---------|------|--------|------|---------|--------|---------|
| Min.    | 40.0 | 150    | 35.0 | 32.0    | 63.0   | 73.0    |
| 1st Qu. | 56.0 | 163    | 38.0 | 38.0    | 69.0   | 90.0    |
| Median  | 61.5 | 170    | 39.0 | 40.0    | 71.0   | 95.0    |
| Mean    | 63.2 | 170    | 39.6 | 40.9    | 71.6   | 95.2    |
| 3rd Qu. | 70.0 | 176    | 42.0 | 44.0    | 74.8   | 100.0   |
| Max.    | 95.0 | 188    | 47.0 | 53.0    | 86.0   | 113.0   |

```
table( df$Ojos, df$Tipo, dnn = c("Color ojos", "Complexion") )
```

| Color ojos/Complexion | atlético | delgado | pesado |
|-----------------------|----------|---------|--------|
| oscuros               | 12       | 6       | 0      |
| claros                | 55       | 11      | 14     |

```
table( df$Sexo, df$Ojos, df$Tipo, dnn = c("Sexo", "Ojos", "Tipo") )
```

| Sexo   | Ojos    | Tipo     | Freq |
|--------|---------|----------|------|
| hombre | claros  | atlético | 17   |
|        |         | delgado  | 1    |
|        |         | pesado   | 11   |
|        | oscuros | atlético | 1    |
|        |         | delgado  | 2    |
|        |         | pesado   | 0    |
| mujer  | claros  | atlético | 38   |
|        |         | delgado  | 10   |
|        |         | pesado   | 3    |
|        | oscuros | atlético | 11   |
|        |         | delgado  | 4    |
|        |         | pesado   | 0    |

```
aggregate( df[ , 2:4],
  list( Sexo = df$Sexo,
        `Color ojos` = df$Ojos ),
  median )
```

| Sexo   | Color ojos | Peso | Altura | Pie |
|--------|------------|------|--------|-----|
| mujer  | oscuros    | 58   | 166    | 38  |
| hombre | oscuros    | 78   | 175    | 42  |
| mujer  | claros     | 58   | 165    | 38  |
| hombre | claros     | 71   | 178    | 42  |

Aún así, vemos que siguen habiendo tablas como la siguiente, que solo se pueden presentar por consola:

```
tapply( df$Pie, list( df$Sexo, df$Ojos, df$Tipo ), median )
```

```
## , , atlético
##
##      oscuros claros
## mujer      38     38
## hombre     43     42
##
## , , delgado
##
##      oscuros claros
## mujer     37.5   38.5
## hombre    42.0   42.0
##
```



```
## , , pesado
##
##      oscuros claros
## mujer      NA     39
## hombre     NA     42
```

Otro problema a tener en cuenta es que no podemos modificar el estilo por defecto tabla, ni añadir, por ejemplo, pies de tabla. En esos caso que usar funciones y librerías concretas de presentación de tablas como

- `kable()` del paquete `knitr`
- `pandoc.table()` de `pander`
- `tabular()` de la librería `tables`