Estadística Descriptiva con 

Estadística Descriptiva con R

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Departamento de Matemáticas

Fecha: 15 de septiembre de 2011

Índice general

Capítulo 0

Estadística Descriptiva con R

1. Introducción.

La *estadística descriptiva* es el conjunto de métodos diseñados para organizar, resumir y representar los datos recogidos en el curso de algún estudio. Su finalidad es convertir los datos brutos en información que pueda ser fácilmente entendida y asimilada. En este sentido, la estadística descriptiva es una herramienta indispensable para la *exploración* de los datos: descubrir tendencias, asociaciones, características relevantes, ...

Para poder aplicar los métodos de la estadística descriptiva de manera eficiente se hace necesario disponer de programas informáticos adecuados para ello, con capacidad para capturar datos desde distintas fuentes, procesarlos, transformarlos si es necesario, y generar tablas, gráficos y medidas de síntesis.

 http://www.r-project-org

En este curso proponemos la utilización del paquete estadístico R, que cuenta con numerosas ventajas: es gratuito, se actualiza constantemente, dispone de librerías adicionales para múltiples aplicaciones (genética, climatología, pesquerías, economía, ...), permite la realización de gráficos de alta calidad, incluye un lenguaje de programación que permite al usuario desarrollar funciones a medida y funciona en todas las plataformas (Windows, Linux y Mac).

Pretendemos además que este capítulo sea interactivo y que el alumno vaya aplicando las técnicas y métodos que en él se explican a medida que avanza en su lectura. Con este fin se han dispuesto en la web de la asignatura diversas bases de datos que pueden ser utilizadas libremente para el aprendizaje.

2. Objetivos.

Al finalizar el estudio de este tema, se espera que el alumno sea capaz de:

- Comprender la importancia de la exploración de los datos mediante tablas y gráficos.
- Distinguir los distintos tipos de variables y sus características.
- Calcular e interpretar correctamente la información aportada por las diferentes medidas de síntesis.
- Conocer los métodos de estadística descriptiva para el estudio conjunto de dos variables.
- Utilizar el programa R para la exploración y descripción de datos.

3. Población y Muestra.

Cuando se realiza un estudio de cualquier tipo (de investigación, de mercado, de evaluación de calidad, etc.), generalmente se observan características o magnitudes correspondientes a los elementos de una *población* de interés. Normalmente dicha población no suele ser accesible en su totalidad, y el estudio ha de reducirse a unos cuantos elementos escogidos de la misma. El subconjunto de objetos (o sujetos) de la población que son incluidos en el estudio, recibe el nombre de *muestra*. Así, por ejemplo, en el ámbito de las Ciencias Marinas:

- El estudio de las poblaciones biológicas –cefalópodos, crustáceos, peces, mamíferos marinos, ...– se realiza a partir de los datos aportados por los ejemplares que se capturan o se observan durante una campaña de muestreo.
- El estudio de parámetros físicos o químicos –temperatura, salinidad, velocidad de corriente, concentración de CO_2 disuelto, ...– se realiza a partir de los datos obtenidos por sensores que se colocan en los lugares de interés durante periodos concretos.

El proceso mediante el cual los resultados *particulares* obtenidos en un muestreo se emplean para responder cuestiones *generales* sobre la población recibe el nombre de *inferencia.* Cuando el muestreo es *aleatorio* (todos los elementos de la población tienen, a priori, la misma probabilidad de formar parte de la muestra[1]) el proceso de inferencia se lleva a cabo mediante métodos estadísticos basados en la probabilidad, y recibe el nombre de *Inferencia Estadística.*

[1]Ello garantiza al mismo tiempo que la muestra es *representativa* de la población, es decir, tiene sus mismas características generales. Un muestreo no aleatorio, en el que se seleccionan los objetos con unas características determinadas, puede resultar *tendencioso* y no representar para nada a la población de interés.

4. Tipos de datos.

Las magnitudes o atributos medidos sobre cada objeto de la muestra reciben el nombre de *variables estadísticas* (longitud, peso, duración, temperatura, ...). Los *datos* son los valores que toma la variable en cada objeto. Formalmente, una variable estadística X definida sobre una población Ω y con valores en un conjunto V es una función $X : \Omega \longrightarrow V$, que a cada objeto ω de Ω , le asigna un único valor en V. Cuando este conjunto es numérico ($V \subseteq \mathbb{R}$), la variable se dice *cuantitativa* o *numérica*, y en caso contrario *cualitativa* o *categórica*.

Las variables cuantitativas son *continuas* si pueden tomar cualquier valor dentro de un rango numérico (temperatura, peso, longitud, etc.); son *discretas* si no admiten todos los valores intermedios de un rango. Las variables discretas suelen tomar sólo valores *enteros* (número de hijos de una familia, número de fallos en un equipo técnico durante un año, etc.).

Las variables categóricas son *binarias* si solo toman dos valores (sano/enfermo, observado/no observado, etc.). Pueden ser además *nominales*, si los datos corresponden a categorías sin relación de orden entre sí (color, sexo, profesión,...), u *ordinales* cuando sí que hay relación de orden (curso escolar, posición en una cola,...).

Una vez que se han observado los valores que toman las variables de nuestro estudio es preciso guardar los datos en un archivo que pueda ser leido fácilmente por un programa estadístico, en nuestro caso R. Si la muestra está formada por n objetos $\omega_1, \omega_2, \ldots, \omega_n$, sobre los que se han medido p variables $X_1, X_2, \ldots, X_p$, los datos resultantes deberán organizarse, en general, en forma de una matriz con n filas (cada fila corresponde a un objeto) y p columnas (cada columna corresponde a una variable), tal como se muestra en la tabla 1. Denotamos por x_{ij} al valor observado de la variable X_j sobre el objeto ω_i.

	Variables					
Objetos	X_1	X_2	$\ldots$	X_j	$\ldots$	X_p
ω_1	x_{11}	x_{12}	$\ldots$	x_{1j}	$\ldots$	x_{ip}
ω_2	x_{21}	x_{22}	$\ldots$	x_{2j}	$\ldots$	x_{2p}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
ω_i	x_{i1}	x_{i2}	$\ldots$	x_{ij}	$\ldots$	x_{ip}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
ω_n	x_{n1}	x_{n2}	$\ldots$	x_{nj}	$\ldots$	x_{np}

Tabla 1: Organización de los datos para su tratamiento estadístico.

En la mayor parte de los casos la matriz de datos en bruto, aunque contiene toda la información recogida en el muestreo, no permite interpretar la información de forma clara. La percepción y resumen de las características de los datos se consigue fundamentalmente a través de:

1. Tablas de Frecuencias.
2. Representaciones Gráficas.
3. Medidas de Síntesis de datos numéricos.

4.1. Datos de ejemplo: acceso y lectura.

Para ilustrar los distintos métodos de la Estadística Descriptiva utilizaremos los datos que se encuentran en el archivo sargos.csv, que puede descargarse de la web de la asignatura[2]. Este archivo corresponde a un muestreo de sargos realizado sobre capturas de esta especie en las Islas Canarias durante el año 2005. La tabla 2 muestra datos relativos a 10 ejemplares, si bien la base de datos completa contiene 200. Sobre cada ejemplar se han medido las variables: *isla* (donde fue capturado), *sexo, long* (longitud total), *ldors* (longitud medida desde el morro hasta la aleta dorsal), *lpect* (longitud hasta la aleta pectoral), *loper* (longitud hasta el opérculo), *altop* (altura del pez en la región del opérculo), *peso* (peso total), *pgon* (peso de las gónadas), *phig* (peso del hígado), *ptdo* (variable que vale 1 si el pez está parasitado por larvas de anisákidos y 0 si no no está) y *larvas* (número de larvas de anisákidos encontradas en la cavidad abdominal del pez). Como puede apreciarse, el peso de las gónadas no está disponible para todos los peces. A estos valores no disponibles nos referiremos como *valores perdidos*.

isla	sexo	long	ldors	lpect	loper	altop	peso	pgon	phig	ptdo	larvas
GC	Macho	22,59	5,14	5,32	4,08	8	163,81		17,3	0	0
HI	Macho	26,35	6,44	6,02	5,36	8,89	277,04	6,86	22,3	0	0
FV	Macho	21,23	5,11	4,63	4,39	6,39	135,69	1,98	5,4	0	0
TF	Macho	22,7	5,35	4,61	4,95	7,33	167,54	1,65	27	1	5
LZ	Hembra	20,2	4,84	4,58	4,38	6,63	131,68		7,1	0	0
TF	Macho	21,6	5,5	5,56	3,83	6,08	176,21	4,54	22,9	0	0
GC	Hembra	25,18	5,73	5,52	5,72	8,14	257,38	37,01	12,4	0	0
GC	Macho	21,68	5,02	5,19	4,74	6,62	145,14		18,2	0	0
LP	Macho	23,29	6,03	5,4	5,34	6,95	201,82	3,55	12,7	0	0
TF	Hembra	16,39	4,31	3,54	3,57	5,21	78,54		6,4	0	0

Tabla 2: Datos recogidos en un muestreo de ejemplares de Sargo (*Diplodus Sargus*) en las Islas Canarias. Se muestran solo 10 ejemplares.

El archivo está en formato csv (*Comma Separated Values*), que es un archivo ASCII plano (es decir, sin información de formato de ningún tipo), en el que los distintos valores están separados por el símbolo *punto y coma* (;). Puede abrirse con cualquier editor de texto, si

[2]Este archivo puede descargarse también desde http://dl.dropbox.com/u/7610774/sargos.csv.

bien las hojas de cálculo estándar (OpenOffice o Microsoft Excel) nos lo muestran en forma de tabla visualmente más atractiva. En la primera fila del archivo se encuentran los nombres de las variables.

Supondremos que una vez descargado el archivo lo hemos guardado en el directorio[3]:

c:\documents and settings\fcmar\data\

Para leer este archivo con R utilizaremos los siguientes comandos:

```
> setwd("c:/documents and settings/fcmar/data/")
> sargos = read.table(file = "sargos.csv", sep = ";", dec = ",",
     header = TRUE)
```

El primer comando *setwd()* (acrónimo de *set working directory*) se encarga de indicar a R el directorio de trabajo, en el que se encuentran los datos (y en el que previsiblemente guardaremos los resultados).

Importante: Las barras empleadas para especificar el directorio deben ser de la forma "/" y no la habitual "\" en Windows.

La segunda linea es la que lee el fichero sargos.csv y asigna su contenido al objeto *sargos*. Indicamos además que los datos están separados por punto y coma (*sep=";"*), que el símbolo decimal que se usa en los valores numéricos es la coma (*dec=","*), y que el archivo tiene una cabecera con los nombres de las variables (*header=TRUE*).

Nota: si disponemos de un ordenador con conexión directa a internet, el fichero sargos.csv puede ser importado directamente desde la red con R mediante:

```
> sargos = read.table(file = "http://dl.dropbox.com/u/7610774/sargos.csv",
     sep = ";", dec = ",", header = TRUE)
```

El objeto en que R almacena la matriz de datos con la que vamos a trabajar –en el ejemplo, la tabla leída del archivo sargos.csv se ha almacenado en el objeto *sargos*– recibe el nombre de *data.frame*. En esencia, un *data.frame* es una matriz de datos cuyas columnas representan variables identificadas por su nombre.

[3]Suponemos que se utiliza un ordenador con sistema operativo Windows, que es la situación más habitual. En caso de utilizar Linux o Mac las rutas de directorio pueden ser ligeramente distintas. En lo que se refiere al funcionamiento de R, es idéntico en todos los sistemas operativos.

4.2. Acceso directo a las variables dentro de una matriz de datos.

En general, cuando deseamos acceder a una variable que está dentro de un *data.frame* deberemos anteponer al nombre de la variable el nombre del objeto que la contiene, separados por el símbolo $. Por ejemplo, para ver el contenido de la variable long deberíamos escribir sargos$long. Si hemos de trabajar con muchas variables, tener que escribir siempre el nombre de la matriz de datos puede llegar a hacerse muy tedioso. Podemos habilitar un *"acceso directo"* a las variables por su nombre utilizando la función:

```
> attach(sargos)
```

A partir de ahora todas las variables estarán disponibles directamente por su nombre. Para cancelar este acceso directo, deberemos ejecutar *detach(sargos)*.

4.3. Tipos de datos en R.

Hemos visto al comienzo de esta sección que las variables estadísticas pueden clasificarse en categóricas y numéricas, y estas últimas en discretas o continuas. R distingue las variables según su *clase*:

- *numeric*: variables numéricas continuas.
- *integer*: variables numéricas discretas.
- *character*: variables alfanuméricas; sus valores son combinaciones de cifras y letras.
- *factor*: variables categóricas; R almacena internamente los valores de un factor como números enteros, pero los muestra como valores alfanuméricos.

La función *str()* (acrónimo de *estructura*) muestra la estructura del objeto especificado. Así, si aplicamos esta función a nuestros datos de ejemplo obtenemos:

```
> str(sargos)
```

```
'data.frame':        200 obs. of  12 variables:
 $ isla  : Factor w/ 7 levels "FV","GC","HI",..: 2 3 1 7 6 7 2 2 5 7 ...
 $ sexo  : Factor w/ 2 levels "Hembra","Macho": 2 2 2 2 1 2 1 2 2 1 ...
 $ long  : num  22.6 26.4 21.2 22.7 20.2 ...
 $ ldors : num  5.49 5.49 5.36 4.5 5.36 5 5.66 4.78 4.83 3.79 ...
```

```
 $ lpect : num  5.32 6.02 4.63 4.61 4.58 5.56 5.52 5.19 5.4 3.54 ...
 $ loper : num  4.08 5.36 4.39 4.95 4.38 3.83 5.72 4.74 5.34 3.57 ...
 $ altop : num  8 8.89 6.39 7.33 6.63 6.08 8.14 6.62 6.95 5.21 ...
 $ peso  : num  164 277 136 168 132 ...
 $ pgon  : num  NA 6.86 1.98 1.65 NA ...
 $ phig  : num  17.3 22.3 5.4 27 7.1 22.9 12.4 18.2 12.7 6.4 ...
 $ ptdo  : int  0 0 0 1 0 0 0 0 0 0 ...
 $ larvas: int  0 0 0 5 0 0 0 0 0 0 ...
```

Podemos ver que las variables isla y sexo han sido identificadas como factores (*factor*); las variables long, ldors, lpect, loper, altop, peso, pgon y phig han sido identificadas como *numeric* (valores reales, variables numéricas continuas); y las variables ptdo y larvas han sido identificadas como *integer* (valores enteros, variables numéricas discretas).

La variable isla es un factor; ello significa que si pedimos a R que nos muestre sus valores, nos los mostrará como alfanuméricos:

```
> isla
```

```
  [1] GC HI FV TF LZ TF GC GC LP TF GC GC LP LP GC HI GC FV FV FV GC
 [22] GC TF GC HI LZ GC GC LZ HI LG TF GC HI LZ HI LP LZ TF GC TF LP
 [43] LZ TF LP TF LG LZ FV TF TF GC GC LP TF FV LZ LZ TF TF LG FV GC
 [64] GC HI LZ LZ FV GC GC LG TF GC LZ LZ LP TF LP LZ LZ GC FV TF GC
 [85] LG FV FV GC TF FV TF GC LG LZ LZ TF HI TF LZ FV HI FV FV TF TF
[106] GC GC FV LP LZ FV LP GC HI LP LZ HI FV LZ TF TF FV LZ HI GC FV
[127] GC FV LG GC LZ GC FV LG FV GC FV LP FV FV LG TF HI TF TF GC LP
[148] LZ GC LP GC GC LZ LZ FV TF GC GC FV TF GC LP FV LP TF LP LZ TF
[169] LP LP TF TF GC GC LP GC LP GC TF TF LP TF LP LZ GC HI LZ FV HI
[190] TF FV FV GC GC GC LZ LZ LZ TF TF
Levels: FV GC HI LG LP LZ TF
```

Pero si ejecutamos la función *unclass()* vemos que internamente los valores de esta variable están almacenados como números enteros:

```
> unclass(isla)
```

```
  [1] 2 3 1 7 6 7 2 2 5 7 2 2 5 5 2 3 2 1 1 1 2 2 7 2 3 6 2 2 6 3 4 7
 [33] 2 3 6 3 5 6 7 2 7 5 6 7 5 7 4 6 1 7 7 2 2 5 7 1 6 6 7 7 4 1 2 2
 [65] 3 6 6 1 2 2 4 7 2 6 6 5 7 5 6 6 2 1 7 2 4 1 1 2 7 1 7 2 4 6 6 7
```

```
 [97] 3 7 6 1 3 1 1 7 7 2 2 1 5 6 1 5 2 3 5 6 3 1 6 7 7 1 6 3 2 1 2 1
[129] 4 2 6 2 1 4 1 2 1 5 1 1 4 7 3 7 7 2 5 6 2 5 2 2 6 6 1 7 2 2 1 7
[161] 2 5 1 5 7 5 6 7 5 5 7 7 2 2 5 2 5 2 7 7 5 7 5 6 2 3 6 1 3 7 1 1
[193] 2 2 2 6 6 6 7 7
attr(,"levels")
[1] "FV" "GC" "HI" "LG" "LP" "LZ" "TF"
```

4.4. Recodificación y etiquetado de niveles de los factores.

En muchas ocasiones, los niveles de un factor son poco ilustrativos de su significado. En los datos de nuestro ejemplo, la variable que indica si un pez está parasitado o no, ptdo, toma los valores 0 y 1, y éstos son los valores que aparecerán en las tablas y gráficos que podamos hacer con esta variable. Sería deseable que en su lugar apareciesen los términos "*No Parasitado*" y *"Parasitado",* ya que de esta forma la salida de resultados sería más clara e interpretable. Podemos conseguir este efecto creando un nuevo factor a partir de esta variable, y asignando etiquetas a sus valores mediante la siguiente sintaxis:

```
> fptdo = factor(ptdo, levels = c(0, 1), labels = c("No Parasitado",
     "Parasitado"))
```

Con ello hemos creado una nueva variable fptdo de clase *factor*; esta variable se construye a partir de ptdo, asignando a sus niveles originales, *levels=c(0,1)*, unas nuevas *etiquetas,* *labels=c("No Parasitado","Parasitado")* (las etiquetas deben asignarse en el mismo orden que en *levels()*). De esta manera, a partir de ahora, en todos los resultados que involucren a la variable fptdo (gráficos, tablas, etc.) sus valores aparecerán identificados como "*No Parasitado*" y *"Parasitado"*.

Nota: al crear una variable de clase *factor*, R almacena internamente sus valores como enteros consecutivos (1, 2, ...), si bien en todas las salidas se mostrarán exclusivamente las etiquetas que hayamos puesto. Puede observarse la codificación interna que se ha hecho de la variable fptdo mediante *unclass(fptdo)*.

Importante: si la variable que convertimos en factor tiene otros valores distintos que no han sido especificados en *levels*, tales valores se pierden: se convierten en *No Asignados* (*NA*), y no serán utilizados en los análisis que posteriormente podamos hacer de los datos.

¿Crear variables o recodificar variables existentes?

Acabamos de ver como se crea un factor (fptdo) a partir de una variable existente (ptdo). Si hubiésemos utilizado la sintaxis:

```
> ptdo = factor(ptdo, levels = c(0, 1), labels = c("No Parasitado",
     "Parasitado"))
```

en lugar de *crear* un nuevo factor, habríamos *recodificado* la variable ptdo ya existente, que de esta forma quedaría convertida directamente en factor (y habría perdido sus valores originales, en este caso 0 y 1)[4]. Podemos comprobarlo, por ejemplo, utilizando el comando ***unique()***, que muestra los valores *distintos* que toma la variable:

```
> unique(ptdo)
```

```
[1] No Parasitado Parasitado
Levels: No Parasitado Parasitado
```

¿Es mejor crear nuevas variables o recodificar las que ya existen? Si somos principiantes en R lo mejor es crear nuevas variables; de esta forma las variables originales estarán siempre disponibles y en caso de error podemos volver a utilizarlas. Si las recodificamos y nos hemos equivocado en la recodificación, tendríamos que recuperar la variable original, lo que a veces puede resultar complicado.

En este caso particular la recuperación resulta sencilla, ya que los valores originales de ptdo siguen almacenados en el data.frame sargos (vinculado al *entorno de trabajo* actual mediante el comando ***attach***). Si borramos la variable ptdo mediante:

```
> rm(ptdo)
```

en realidad sólo borramos la variable recodificada; la variable ptdo del data.frame original, que permanecía en el entorno de trabajo vuelve a ser accesible:

```
> unique(ptdo)
```

```
[1] 0 1
```

[4] En sentido estricto, la variable ptdo que pertenece al data.frame sargos, no se elimina de éste, sino que queda *oculta* por la nueva definición que se ha dado de dicha variable.

5. Tablas de frecuencias y representaciones gráficas.

5.1. Variables categóricas o numéricas discretas.

Cuando se observan variables categóricas tales como la isla en que fue capturado un pez, su sexo, y si está o no parasitado, muchos de sus valores aparecen repetidos. La *frecuencia absoluta* de la i-ésima categoría es el número de veces n_i que se repite dicha categoría en el total de observaciones. La *frecuencia relativa* es la proporción:

$$f_i = \frac{n_i}{n}$$

siendo $n = \sum_{i=1}^{k} n_i$ el número total de observaciones (k es el número de categorías). La frecuencia relativa suele también expresarse en porcentaje:

$$f_i = 100 \cdot \frac{n_i}{n} \,\%$$

Estas definiciones se extienden también a la construcción de tablas de frecuencias para variables numéricas discretas. En este último caso se suele considerar también la *frecuencia acumulada* hasta el valor x_i como el número $N_i = \sum_{j=1}^{i} n_j$ de observaciones menores o iguales que x_i. La *frecuencia acumulada relativa* es la proporción:

$$F_i = \frac{N_i}{n}$$

Estas frecuencias suelen presentarse como se muestra en la tabla 3. En la columna de la variable X se anotan sólo las k categorías o valores *distintos* que toma la variable, en orden creciente si X es numérica. Asimismo las frecuencias acumuladas sólo se incluyen cuando X es numérica.

X	Frecuencia Absoluta	Frecuencia Relativa	Frec. Acum. Absoluta	Frec. Acum. Relativa
x_1	n_1	f_1	N_1	F_1
x_2	n_2	f_2	N_2	F_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_k	n_k	f_k	N_k	F_k

Tabla 3: Tabla de frecuencias para variables categóricas o numéricas discretas.

Tablas de frecuencias para variables categóricas o discretas en R.

Los siguientes comandos nos muestran las tablas de frecuencias absolutas y relativas para la isla en que se han capturado los peces de nuestro ejemplo:

```
> table(isla)

isla
FV GC HI LG LP LZ TF
32 48 15  9 24 32 40

> prop.table(table(isla))

isla
   FV    GC    HI    LG    LP    LZ    TF
0.160 0.240 0.075 0.045 0.120 0.160 0.200
```

De igual modo, para el número de larvas:

```
> table(larvas)

larvas
  0   3   4   5   6   7   8   9
170   4   2   4   2   3   9   6

> prop.table(table(larvas))

larvas
    0     3     4     5     6     7     8     9
0.850 0.020 0.010 0.020 0.010 0.015 0.045 0.030
```

Para las frecuencias acumuladas utilizamos la función *cumsum()*:

```
> cumsum(table(larvas))

  0   3   4   5   6   7   8   9
170 174 176 180 182 185 194 200

> cumsum(prop.table(table(larvas)))
```

```
    0     3     4     5     6     7     8     9
0.850 0.870 0.880 0.900 0.910 0.925 0.970 1.000
```

Podemos construir una tabla más compacta para estas frecuencias del siguiente modo:

```
> tbl = table(larvas)
> nlarvas = names(tbl)
> fi = as.vector(tbl)
> fri = as.vector(prop.table(tbl))
> Fi = cumsum(fi)
> Fri = cumsum(fri)
> data.frame(nlarvas, fi, fri, Fi, Fri)
```

```
  nlarvas  fi   fri  Fi   Fri
1       0 170 0.850 170 0.850
2       3   4 0.020 174 0.870
3       4   2 0.010 176 0.880
4       5   4 0.020 180 0.900
5       6   2 0.010 182 0.910
6       7   3 0.015 185 0.925
7       8   9 0.045 194 0.970
8       9   6 0.030 200 1.000
```

Aquí hemos utilizado los siguientes comandos de R:

- *tbl=table(larvas)*: asignamos el contenido de la tabla de frecuencias al objeto *tbl*.
- *nlarvas=names(tbl)*: asigna a *nlarvas* los nombres (categorías) de la tabla anterior; en este ejemplo, las categorías son los distintos números de larvas encontrados. Utilizaremos estos nombres como primera columna de nuestra tabla compacta.
- *f=as.vector(tbl)*: la función *table(larvas)* como hemos visto antes, crea una tabla de frecuencias absolutas. En R una tabla es un objeto con una estructura muy particular, que contiene los nombres de las distintas categorías de la variable que se tabula y sus frecuencias. Al aplicar la función *as.vector()* a esta estructura, la convierte en un simple vector, sin nombres, sólo con los valores de las frecuencias, que se van a usar como segunda columna en la tabla.
- *data.frame()*: crea la matriz de datos que contiene la tabla de frecuencias que se presenta por pantalla.

Sugerencia: Si necesitáramos hacer frecuentemente tablas como ésta, resulta conveniente definir una función en R para ello, que nos ahorre tener que escribir todas estas líneas cada vez. Esta función podría ser, por ejemplo:

```
> tablaFrec = function(x) {
      tbl = table(x)
      categ = names(tbl)
      fi = as.vector(tbl)
      fri = as.vector(prop.table(tbl))
      Fi = cumsum(fi)
      Fri = cumsum(fri)
      tabla = data.frame(categ, fi, fri, Fi, Fri)
      names(tabla)[1] = deparse(substitute(x))
      return(tabla)
 }
```

Observemos que la función usa prácticamente los mismos comandos que acabamos de ver. Se ha añadido una linea al final para mejorar la presentación:

- *names(tabla)[1]=deparse(substitute(x))*: Nuestra función recibirá en general como argumento una variable arbitraria *x*. La función *deparse(substitute(x))* extrae su nombre, y *names(tabla)[1]=* lo asigna como cabecera de la primera columna de nuestra tabla.

Para aplicar la función que acabamos de definir a la variable *larvas* bastaría con introducir:

```
> tablaFrec(larvas)
```

A medida que vamos trabajando con R podemos ir construyendo nuestra colección de funciones útiles y guardarlas, por ejemplo, en el archivo *MisFunciones.R*. Para tenerlas disponibles cada vez que usemos R bastará con ejecutar al principio de nuestra sesión:

```
> source("MisFunciones.R")
```

Gráficos: diagramas de barras y diagramas de sectores.

Las tablas de frecuencias que hemos visto en esta sección se representan gráficamente mediante:

- *Diagramas de barras,* que en R se obtienen con el comando `barplot()`.
- *Diagramas de sectores,* que en R se obtienen con el comando `pie()`.

En la figura 1 se muestran ambos diagramas para el número de capturas de sargos por isla en la muestra que estamos utilizando como ejemplo. Para generar estos gráficos se ha utilizado la sintaxis:

```
> barplot(table(isla))
> pie(table(isla))
```

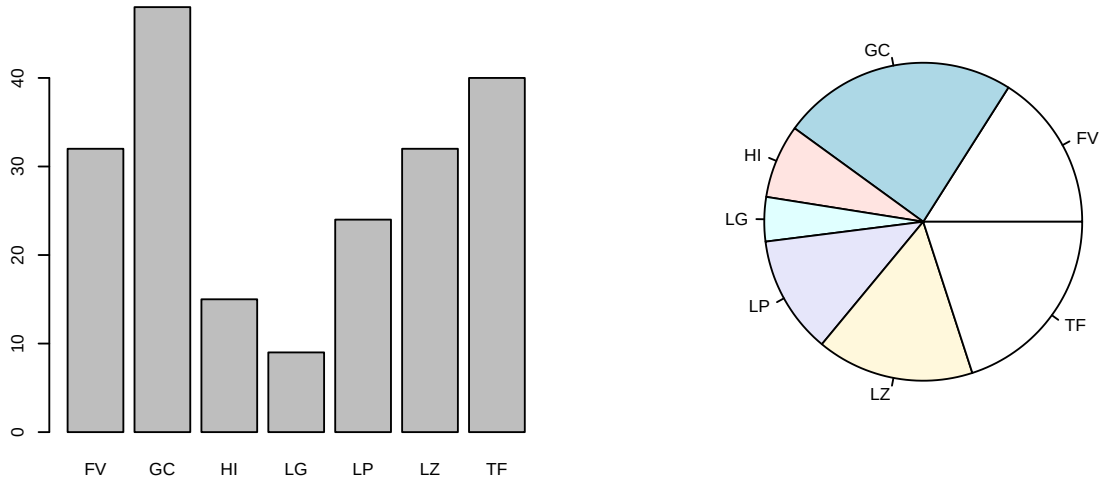

Figura 1: Izquierda: gráfico de barras del número de ejemplares capturados por isla. Derecha: gráfico de sectores con los mismos datos.

Como puede apreciarse en esta figura, en el diagrama de barras la altura de cada barra es igual a la frecuencia absoluta representada. Asimismo, en el diagrama de sectores, el ángulo del sector correspondiente a cada categoría es proporcional a su frecuencia. En el caso del diagrama de barras, si queremos que la altura de las barras represente frecuencias relativas, bastará emplear `prop.table()` del modo siguiente:

```
> barplot(prop.table(table(isla)))
```

Mejorando la presentación de los gráficos.

Los dos gráficos anteriores, si bien representan correctamente las frecuencias observadas, resultan poco informativos: carecen de título; las etiquetas de las barras o sectores (FV, GC, HI, etc) resultan poco claras (el lector del informe estadístico puede no saber qué significan estas siglas); estas etiquetas figuran en orden alfabético y quizás tuviese más sentido colocarlas en orden geográfico, con las islas de este a oeste; incluso el gráfico en tonos grises puede resultar visualmente poco atractivo.

Con R es sencillo mejorar el aspecto de los gráficos. La siguiente sintaxis produce el diagrama de barras mostrado en la figura 2, que mejora bastante al de la figura 1:

```
> isla = factor(isla, levels = c("HI", "LP", "LG", "TF",
     "GC", "FV", "LZ"), ordered = TRUE)
> par(cex.axis = 0.9, las = 1)
> barplot(prop.table(table(isla)), main = "Ejemplares capturados por isla",
     names.arg = c("Hierro", "La\nPalma", "La \nGomera",
         "Tenerife", "Gran \nCanaria", "Fuerte-\nventura",
         "Lanza-\nrote"), col = terrain.colors(12))
```

En la primera linea hemos redefinido el factor isla, simplemente colocando la lista de niveles de este factor en el orden Oeste-Este, e indicando a R que debe mantener esta ordenación (*ordered=TRUE*) en todas las representaciones que afecten a esta variable.

En la segunda linea hemos modificado algunos de los parámetros gráficos que usa R por defecto. En particular, *cex.axis=0.9* disminuye el tamaño de la letra que se usa para etiquetar las barras a un 90 % de su tamaño original (con objeto de que se puedan poner los nombres completos de las islas). A su vez *las=1* produce que las etiquetas en ambos ejes se escriban horizontalmente.

Por último, en la tercera linea se genera el diagrama de barras. Con la opción *main* se indica el título del gráfico. En *names.arg* se especifican los nombres que se van a utilizar como etiquetas de las barras. Si no se incluye esta opción, se usan las etiquetas del factor que se va a tabular. En este caso, hemos incluído los nombres de las islas para poder separar en dos lineas los nombres largos: para ello, hay que indicar con "\n" el lugar de la separación. La última opción, *col*, permite indicar los colores a utilizar. En este caso hemos utilizado la paleta *terrain.colors(n)* que genera n colores dentro de una misma gama[5]. Los colores para un gráfico pueden designarse también por su nombre (en inglés). Así, en este caso

[5]Si el número m de colores a representar es menor que n se utilizan los m primeros de esa gama. Y si el número es mayor, los colores se repiten hasta completar el gráfico.

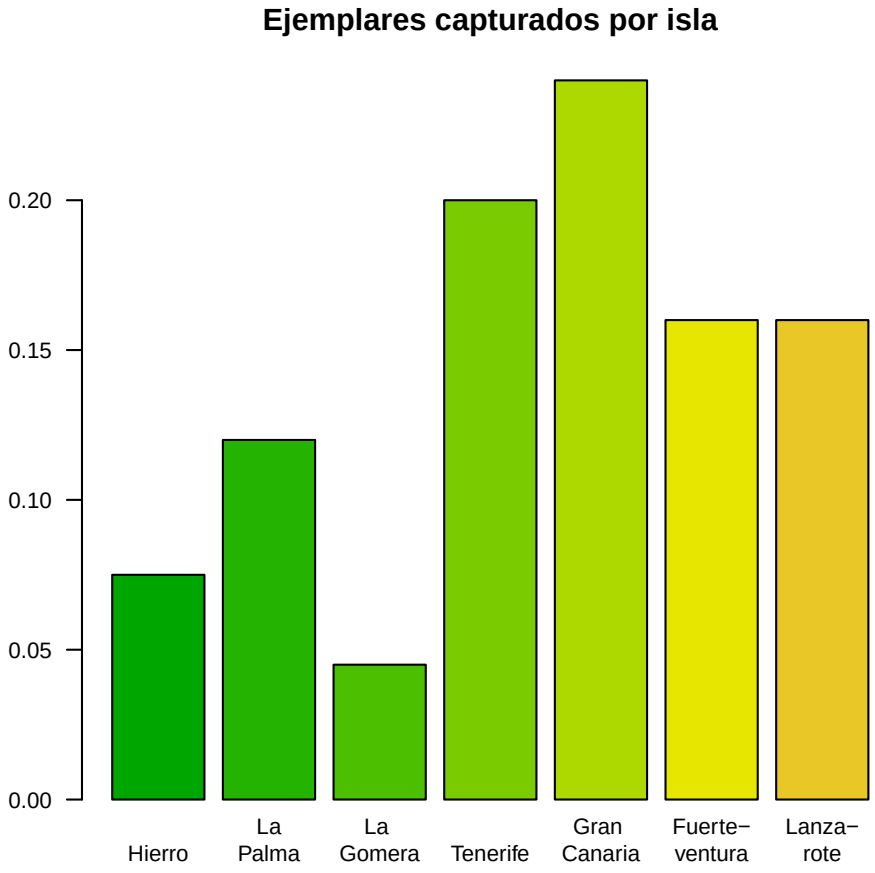

Figura 2: Diagrama de barras mejorado.

podíamos haber especificados los colores de cada barra, por ejemplo, mediante `col=c("red","blue","green","yellow","cyan","orange","magenta")`. Podemos obtener la lista de colores que maneja R mediante el comando `colours()`.

El gráfico de sectores de la figura 1 también puede mejorarse si se incluye el nombre completo de las islas y se indica además el porcentaje de capturas que corresponde a cada una. Requiere algo más de trabajo, pero el código es también muy simple:

```
> noms = c("Hierro", "La Palma", "La Gomera", "Tenerife",
    "Gran Canaria", "Fuerteventura", "Lanzarote")
> pct = prop.table(table(isla)) * 100
> etiquetas = paste(noms, " (", pct, "%)", sep = "")
> pie(table(isla), col = terrain.colors(7), labels = etiquetas,
    main = "Captura por isla")
```

En la primera línea hemos creado el vector `noms` que contiene los nombres de las islas.

En la segunda linea obtenemos la tabla de frecuencias relativas y la multiplicamos por 100; de esta forma sus valores, en lugar de estar expresados en tanto por uno, quedan expresados en tanto por ciento. La tabla se almacena en el objeto `pct`.

En la tercera linea se construyen las etiquetas que se van a añadir al diagrama de sectores; cada etiqueta será el nombre de la isla seguido del porcentaje de capturas obtenido en la misma entre paréntesis. Ello se consigue *"pegando"* mediante la función `paste()` los vectores `noms` y `pct`. La misma función `paste()` nos permite, como vemos, insertar los símbolos de paréntesis y de porcentaje.

Por último, en la cuarta linea, generamos el diagrama de sectores; utilizamos de nuevo la paleta de colores `terrain.colors()`, fijamos como etiquetas (`labels`) las que acabamos de generar, y añadimos un título al gráfico usando `main`. El resultado se muestra en la figura 3.

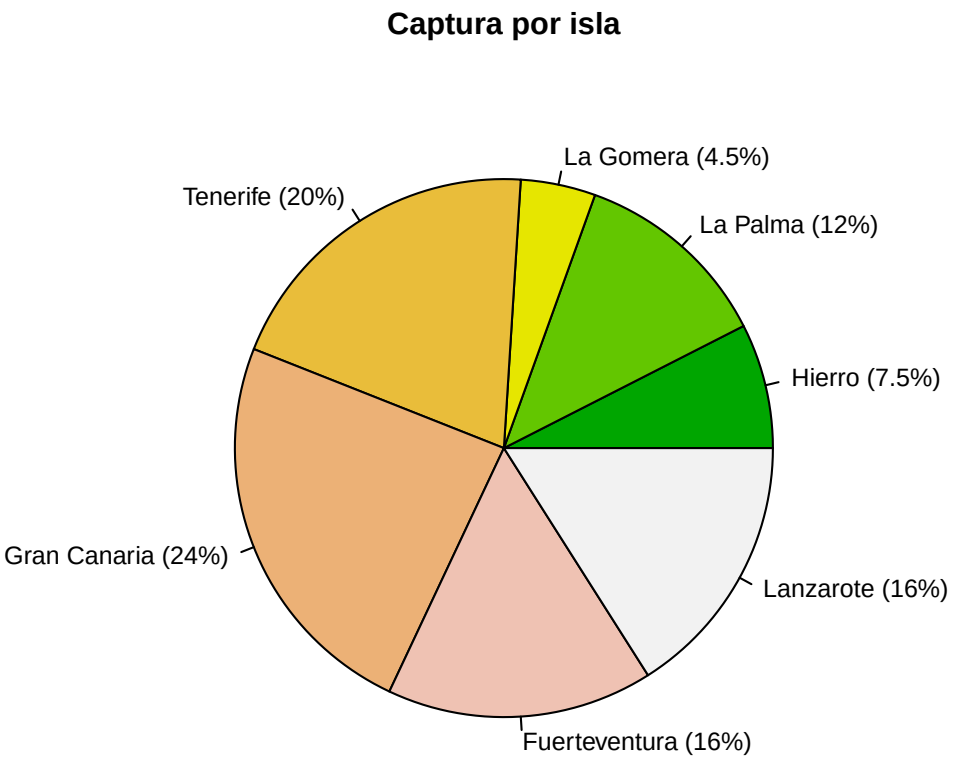

Figura 3: Diagrama de sectores mejorado.

Tablas cruzadas para variables categóricas o numéricas discretas.

Cuando se estudian conjuntamente dos variables categóricas o numéricas discretas, resulta de interés determinar qué valores aparecen juntos con más o menos frecuencia. Con este fin se construyen las *tablas de frecuencias cruzadas*. Si la variable X toma los valores $x_1, x_2, \ldots, x_k$ y la variable Y toma los valores $y_1, y_2, \ldots, y_m$, se denomina *frecuencia absoluta del par* (x_i, y_j) al número de veces n_{ij} que dicha pareja de valores aparecen juntos en la muestra. Las frecuencias absolutas se suelen presentar en una *tabla cruzada* como se muestra en la tabla 4.

	y_1	y_2	$\ldots$	y_m	*Totales*
x_1	n_{11}	n_{12}	$\ldots$	n_{1m}	$n_{1\bullet}$
x_2	n_{21}	n_{22}	$\ldots$	n_{2m}	$n_{2\bullet}$
$\vdots$					
x_k	n_{k1}	n_{k2}	$\ldots$	n_{km}	$n_{k\bullet}$
Totales	$n_{\bullet 1}$	$n_{\bullet 2}$		$n_{\bullet m}$	$n_{\bullet\bullet}$

Tabla 4: Tabla de frecuencias cruzadas.

El valor $n_{i\bullet}$ representa el total de la fila i, $\left(n_{i\bullet} = \sum_{j=1}^{m} n_{ij}\right)$, y por tanto es la frecuencia absoluta con que se observa el valor x_i. Asimismo, el valor $n_{\bullet j}$ representa el total de la fila j, $\left(n_{\bullet j} = \sum_{i=1}^{k} n_{ij}\right)$, y por tanto es la frecuencia absoluta con que se observa el valor y_j. Por último $n_{\bullet\bullet} = \sum_{i=1}^{k}\sum_{j=1}^{m} n_{ij}$ representa el total de valores observados y coincide con el tamaño de la muestra. Las frecuencias $n_{i\bullet}$ y $n_{\bullet j}$ reciben el nombre de *frecuencias marginales* de X e Y, respectivamente.

A partir de una tabla de frecuencias cruzadas absolutas es posible construir tres clases de tablas de frecuencias relativas:

- *Frecuencias relativas globales:* se calculan dividiendo cada frecuencia cruzada por el total de la tabla:
$$f_{ij} = \frac{n_{ij}}{n_{\bullet\bullet}}$$
- *Frecuencias relativas por filas:* se calculan dividiendo cada frecuencia cruzada por el total de su fila:
$$ff_{ij} = \frac{n_{ij}}{n_{i\bullet}}$$
Representan la frecuencia relativa con que se produce cada valor de Y cuando se fija el valor $X = x_i$. Por esta razón, suelen denominarse *frecuencias relativas de Y condicionadas por $X = x_i$*.
- *Frecuencias relativas por columnas:* se calculan dividiendo cada frecuencia cruzada por

el total de su columna:

$$fc_{ij} = \frac{n_{ij}}{n_{\bullet j}}$$

Representan la frecuencia relativa con que se produce cada valor de X cuando se fija el valor $Y = y_j$. Por esta razón, suelen denominarse *frecuencias relativas de X condicionadas por $Y = y_j$*.

Tablas cruzadas en R.

Las tablas cruzadas en R se generan también mediante la función *table()*, especificando ahora como argumento qué variables se desean cruzar. Así, en nuestros datos de ejemplo, si queremos evaluar el número de peces parasitados por *anisakis* capturados en cada una de las islas durante nuestra campaña de muestreo ejecutaríamos simplemente:

```
> table(fptdo, isla)

              isla
fptdo          HI LP LG TF GC FV LZ
  No Parasitado 14 19  8 31 44 28 26
  Parasitado     1  5  1  9  4  4  6
```

Podemos añadir los totales por filas y columnas mediante *addmargins*:

```
> addmargins(table(fptdo, isla))

              isla
fptdo           HI  LP  LG  TF  GC  FV  LZ Sum
  No Parasitado 14  19   8  31  44  28  26 170
  Parasitado     1   5   1   9   4   4   6  30
  Sum           15  24   9  40  48  32  32 200
```

Las distintas tablas cruzadas de frecuencias relativas se obtienen utilizando *prop.table()*:

- Frecuencias relativas globales:

```
> prop.table(table(fptdo, isla))
```

```
                isla
fptdo              HI    LP    LG    TF    GC    FV    LZ
  No Parasitado 0.070 0.095 0.040 0.155 0.220 0.140 0.130
  Parasitado    0.005 0.025 0.005 0.045 0.020 0.020 0.030
```

- Frecuencias relativas por filas: basta añadir a la función *prop.table()* el argumento *margin=1*. Aquí además redondeamos a tres decimales:

```
> round(prop.table(table(fptdo, isla), margin = 1), 3)

                isla
fptdo              HI    LP    LG    TF    GC    FV    LZ
  No Parasitado 0.082 0.112 0.047 0.182 0.259 0.165 0.153
  Parasitado    0.033 0.167 0.033 0.300 0.133 0.133 0.200
```

- Frecuencias relativas por columnas: Igual que en el caso anterior, pero utilizando el argumento *margin=2*:

```
> round(prop.table(table(fptdo, isla), margin = 2), 3)

                isla
fptdo              HI    LP    LG    TF    GC    FV    LZ
  No Parasitado 0.933 0.792 0.889 0.775 0.917 0.875 0.812
  Parasitado    0.067 0.208 0.111 0.225 0.083 0.125 0.188
```

Nota: Se puede omitir la palabra *margin* en los comandos anteriores. El resultado habría sido idéntico utilizando *prop.table(table(fptdo,isla),1)*.

Presentación gráfica de tablas cruzadas.

Las tablas de frecuencias cruzadas pueden representarse gráficamente también mediante *barplot()*. En la figura 4 se muestran dos diagramas de barras en los que se representa la distribución de sexos por isla. El gráfico (a) ha sido generado con la siguiente sintaxis:

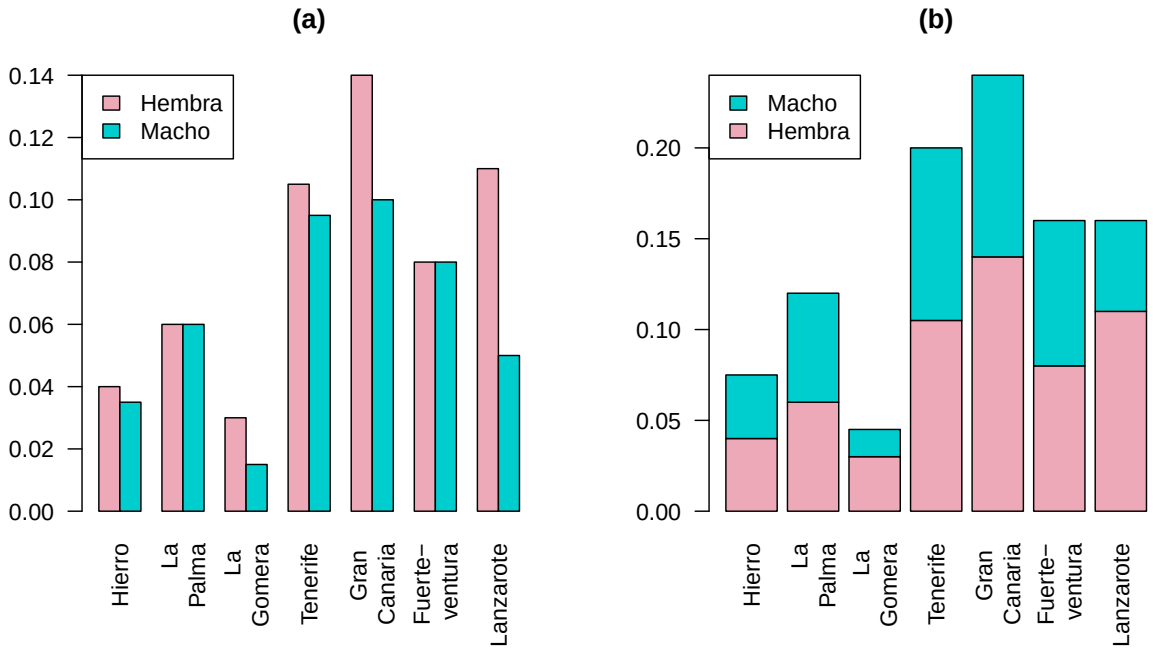

Figura 4: Representación gráfica de la distribución de sexos por isla. (a) Barras emparejadas (*beside=T*). (b) Barras apiladas. (*beside=F*)

```
> barplot(prop.table(table(sexo, isla)), col = c("pink2",
    "cyan3"), beside = TRUE, legend.text = TRUE, names.arg = c("Hierro",
    "La\nPalma", "La \nGomera", "Tenerife", "Gran \nCanaria",
    "Fuerteven-\ntura", "Lanza-\nrote"), las = 2)
```

El gráfico (b) ha sido generado con una sintaxis idéntica salvo que se ha especificado *beside=FALSE* para que las barras se presenten apiladas en lugar de una junto a otra. En este código se han especificado dos colores, uno para hembras y otro para machos. El orden en que se especifican los colores se corresponde con el orden alfabético de las etiquetas de la variable sexo. Por último, la opción *legend.text=TRUE* hace que se dibuje un recuadro en que se especifica qué color corresponde a cada categoría de la variable sexo.

5.2. Variables numéricas continuas.

Si la variable numérica es continua, no cabe esperar repeticiones de un mismo valor de la variable. En este caso, conviene sintetizar el conjunto de valores mediante agrupaciones de la variable en intervalos de clase $(x_{i-1}, x_i]$. En general, los intervalos deben ser de la misma longitud. Denominaremos "marca de clase" al punto medio del intervalo de clase, $m_i = \frac{x_{i-1}+x_i}{2}$. Para determinar el número de intervalos a construir suele emplearse la regla empírica de

Sturges que consiste tomar como número de intervalos un valor próximo a $k \approx 1+3{,}22\log(n)$, siendo n el número total de valores observados. Esta regla es la que emplea R por defecto en la construcción de tablas y gráficos de frecuencias para variables continuas.

Tablas de Frecuencias para variables continuas.

Una vez agrupados los datos en intervalos de clase, el cálculo de las frecuencias es análogo al caso anterior, con la única diferencia de que ahora n_i es el número de observaciones dentro del intervalo $(x_{i-1}, x_i]$, tal como se muestra en la tabla 5.

X (Intervalo)	Marca de clase	Frecuencia Absoluta	Frecuencia Relativa	Frec. Acum. Absoluta	Frec. Acum. Relativa
$[x_0, x_1]$	m_1	n_1	f_1	N_1	F_1
$(x_1, x_2]$	m_2	n_2	f_2	N_2	F_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(x_{k-1}, x_k]$	m_k	n_k	f_k	N_k	F_k

Tabla 5: Tabla de frecuencias para variables continuas.

Tablas de frecuencias para variables continuas en R

La configuración básica de R no dispone de ninguna función específica para la construcción de tablas de frecuencias para variables continuas. Sin embargo, si instalamos[6] el paquete *agricolae* tendremos a nuestra disposición la función *table.freq()*, muy similar a la que hemos construido más arriba para variables discretas. Veamos como utilizar esta función para construir una tabla de frecuencias de las longitudes de los sargos de nuestro ejemplo:

```
> library(agricolae)
> table.freq(hist(long, plot = F))

  Inf Sup MC fi   fri  Fi   Fri
    8  10  9  1 0.005   1 0.005
   10  12 11  1 0.005   2 0.010
   12  14 13  4 0.020   6 0.030
```

[6] Antes de usar una nueva librería –paquete de programas– en R por primera vez, será preciso descargarla e instalarla desde internet. Para ello, arrancamos R, y en el menú superior elegimos la opción *Paquetes →Instalar Paquete(s);* se abre una ventana en la que indicamos el país desde el que deseamos descargar el paquete. Elegimos un pais y a continuación se despliega la lista de paquetes disponibles, en la que seleccionamos el que nos interesa instalar.

```
14  16 15 10 0.050  16 0.080
16  18 17 28 0.140  44 0.220
18  20 19 33 0.165  77 0.385
20  22 21 39 0.195 116 0.580
22  24 23 34 0.170 150 0.750
24  26 25 24 0.120 174 0.870
26  28 27 16 0.080 190 0.950
28  30 29  8 0.040 198 0.990
30  32 31  2 0.010 200 1.000
```

Representación gráfica de las tablas de frecuencias para variables continuas.

Histogramas.

La distribución de frecuencias de variables continuas se representa habitualmente en un *histograma.* Este gráfico se construye levantando sobre cada intervalo un rectángulo de área proporcional a la frecuencia que se pretende representar. En R podemos obtener el histograma de las longitudes de los sargos de nuestra muestra mediante:

```
> hist(long, xlab = "longitud", ylab = "Frecuencia", freq = FALSE,
     main = "Longitudes observadas en la muestra", col = topo.colors(40))
```

En esta sintaxis hemos utilizado los comandos *xlab* e *ylab* para especificar etiquetas en los ejes X e Y respectivamente. Asimismo la opción *freq=FALSE* indica a R que en el eje Y represente frecuencias relativas. Las frecuencias absolutas se obtienen con *freq=TRUE*. El gráfico resultante se muestra en la figura 5.

Polígonos de frecuencias.

Los *polígonos de frecuencias* son representaciones similares al histograma, sustituyendo las barras por líneas que unen los distintos valores de frecuencia correspondientes a cada marca de clase. Suelen utilizarse también para representar las frecuencias acumuladas.

En R no existe ninguna función específica para dibujar polígonos de frecuencias. Sin embargo es muy sencillo construirlos a partir de la tabla de frecuencias:

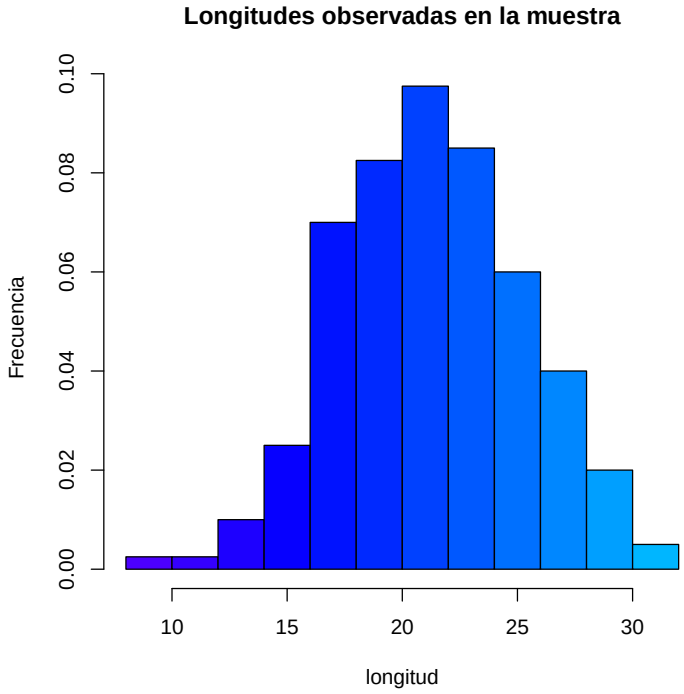

Figura 5: Histograma de longitudes de los sargos de la muestra.

```
> par(mfrow = c(1, 2))
> tbl = data.frame(table.freq(hist(long, plot = FALSE)))
> plot(tbl$MC, tbl$fi, type = "b", col = "red", lwd = 3,
     xlab = "Marca de Clase", ylab = "Frecuencia", sub = "(Longitud del sargo)",
     main = "Poligono de frecuencias absolutas")
> plot(tbl$MC, tbl$Fi, type = "b", col = "darkgreen", lwd = 3,
     xlab = "Marca de Clase", ylab = "Frecuencia", sub = "(Longitud del sargo)",
     main = "Poligono de frecuencias absolutas \nacumuladas")
```

6. Medidas de síntesis o resumen de variables numéricas.

Las variables numéricas pueden resumirse a través de diversas medidas que describen sus características de:

- **Posición:** percentiles y cuartiles
- **Tendencia central:** media, mediana y moda

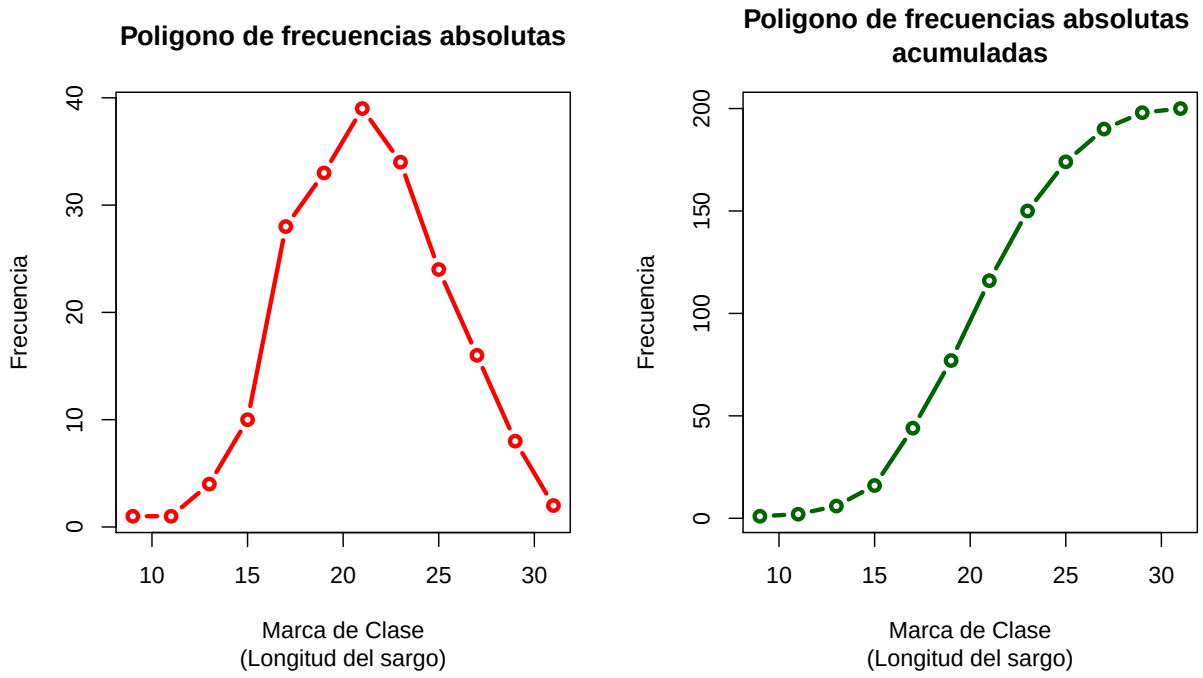

Figura 6: Polígonos de frecuencias para las longitudes de los sargos de la muestra.

- **Dispersión:** Varianza, desviación típica (o estándar), coeficiente de variación y rango.
- **Forma:** Asimetría, Apuntamiento (curtosis).

Pasamos a describir cada una de estas medidas.

6.1. Medidas de posición.

El *k-ésimo percentil* es un valor P_k tal que el $k\,\%$ de las observaciones de la variable tienen un valor menor o igual que P_k. Los percentiles 25, 50 y 75 reciben el nombre de *primer, segundo y tercer cuartiles,* respectivamente.

Los percentiles en R se calculan mediante la función *quantile()*. Así, para calcular los percentiles 0,05, 0,25, 0,50, 0,75, 0,9 y 0,95 de la longitud de los peces obtenidos durante la campaña de muestreo utilizaremos:

```
> quantile(long, probs = c(0.05, 0.25, 0.5, 0.75, 0.9,
    0.95))

    5%     25%     50%     75%     90%     95%
15.470 18.840 21.245 23.980 26.422 27.773
```

6.2. Medidas de tendencia central.

Mediana. Es el valor que ocupa la posición intermedia del conjunto de datos una vez que éstos se han ordenado de menor a mayor. La mediana es, por tanto, aquel valor que es mayor que la primera mitad de los datos, y menor que la segunda mitad. Obviamente, por su definición, coincide con el percentil 50, P_{50} y con el segundo cuartil. Si el número de datos es impar, se toma como mediana el valor que deja a derecha e izquierda el mismo número de datos. Si el número de datos es par , entonces la mediana es igual al promedio de los dos valores centrales.

En R la mediana se calcula mediante el comando *median()*. La longitud mediana de los sargos de la muestra es:

```
> median(long)

[1] 21.245
```

Media aritmética. Si en una muestra de una variable X se han observado los valores $x_1, x_2, \ldots, x_k$, siendo $n_1, n_2, \ldots, n_k$ sus frecuencias absolutas (número de veces que se ha observado cada valor), se define la *media aritmética* como:

$$\overline{x} = \frac{x_1 n_1 + x_2 n_2 + \ldots, + x_k n_k}{n} = \sum_{i=1}^{k} x_i \frac{n_i}{n} = \sum_{i=1}^{k} x_i f_i$$

siendo $n = \sum_{i=i}^{k} n_i$ el número total de observaciones y f_i la frecuencia relativa del valor x_i.

La media aritmética representa el centro de gravedad de los datos, por lo que efectivamente puede entenderse como medida de tendencia central.

En R la media se calcula mediante el comando *mean()*:

```
> mean(long)

[1] 21.3458
```

Moda. Es el valor que más veces se repite (esto es, el valor con mayor frecuencia absoluta). En el caso de datos agrupados suele sustituirse la moda por el *intervalo modal,* que se corresponde con el intervalo de mayor frecuencia absoluta observada. Tanto la moda como el intervalo modal pueden no ser únicos.

R no dispone de ninguna función para calcular la moda. En realidad tal función resulta innecesaria: si la variable que consideramos es categórica o discreta, una simple inspección de la tabla de frecuencias o del diagrama de barras nos indica cuál es el valor más frecuente (o los valores más frecuentes en caso de haber varios). En el caso de variables continuas, la inspección del histograma nos indica el intervalo modal.

En cualquier caso, con variables categóricas podemos construir la siguiente función para obtener la moda:

```
> moda = function(x) {
     tbl = table(x)
     m = which(tbl == max(tbl))
     return(names(m))
 }
```

La aplicamos para determinar de qué isla procede la mayor parte de las capturas de sargos de la muestra:

```
> moda(isla)
```

```
[1] "GC"
```

En el caso de variables continuas, podemos usar la siguiente función para obtener el intervalo modal (o intervalos modales en caso de haber varios) a partir del histograma:

```
> intModal = function(x) {
     tbl = hist(x, plot = FALSE)
     m = which(tbl$counts == max(tbl$counts))
     im = data.frame(tbl$breaks[m], tbl$breaks[m + 1])
     names(im) = c("Inf", "Sup")
     return(im)
 }
```

Aplicamos esta función para hallar el intervalo modal de la longitud de los sargos de la muestra:

```
> intModal(long)
```

```
  Inf Sup
1  20  22
```

Media geométrica. Se define como:

$$\gamma = \{x_1 \cdot x_2 \cdot \cdots \cdot x_n\}^{1/n}$$

Suele utilizarse para promediar incrementos relativos, tales como los que se observan frecuentemente en Economía o Demografía. Por ejemplo, si el tamaño de una población se ha incrementado en un 50 % en un primer año, y ha disminuido un 50 % al año siguiente, la aplicación ingenua de la media aritmética nos llevaría a concluir que, por término medio, el tamaño de la población no cambia. Sin embargo un análisis más atento nos revela que si la población parte inicialmente de, digamos, 1000 individuos, el incremento inicial del 50 % significa una cifra de 1500 individuos al acabar el primer año, y la disminución posterior del 50 % deja la población en 750 individuos; por tanto, en los dos años ha habido un decremento global del 25 %. En realidad, la tasa media de variación interanual en este caso debe calcularse mediante la media geométrica: $\gamma = (1{,}50 \cdot 0{,}50)^{1/2} = 0{,}866$. Su interpretación es que, *por término medio*, cada año el tamaño de la población es un 86.6 % del tamaño del año anterior; dos años sucesivos con esta tasa media producen una tasa acumulada de $0{,}866 \cdot 0{,}866 = 0{,}75$, o lo que es lo mismo, un 75 % del tamaño inicial, lo que sí coincide con la cifra observada.

Si en la definición de media geométrica tomamos logaritmos resulta:

$$\log \gamma = \frac{1}{n} \sum_{i=i}^{n} \log (x_i)$$

Por tanto el logaritmo de la media geométrica coincide con la media aritmética de los logaritmos de los datos originales.

R tampoco dispone de ninguna función para el cálculo de la media geométrica. No obstante, es muy fácil de calcular utilizando la propiedad anterior:

```
> tasas = c(1.5, 0.5)
> exp(mean(log(tasas)))
```

```
[1] 0.8660254
```

O incluso aplicando directamente la definición:

```
> prod(tasas)^(1/length(tasas))
```

```
[1] 0.8660254
```

Hemos utilizado aquí la función *length(tasas)* que nos devuelve la longitud (número de elementos) del vector tasas. En este caso es innecesario (podíamos haber puesto directamente 2), pero de esta forma tenemos una expresión general que nos evita en otros casos tener que contar el número de términos cuya media geométrica se va a calcular.

6.3. Medidas de Dispersión.

Varianza. Si en una muestra de una variable X se han observado los valores $x_1, x_2, \ldots, x_k$, siendo $n_1, n_2, \ldots, n_k$ sus frecuencias absolutas (número de veces que se ha observado cada valor), se define la *varianza muestral* (o *cuasi-varianza*) como:

$$s^2 = var\,(X) = \frac{1}{n-1}\sum_{i=1}^{k}(x_i - \overline{x})^2 n_i = \frac{n}{n-1}\sum_{i=1}^{k}(x_i - \overline{x})^2 \frac{n_i}{n} = \frac{n}{n-1}\sum_{i=1}^{k}(x_i - \overline{x})^2 f_i$$

siendo $n = \sum_{i=i}^{k} n_i$ el número total de observaciones y f_i la frecuencia relativa del valor x_i. Obviamente la varianza es una medida de dispersión ya que cuanto más alejados entre sí se encuentren los valores x_i más lejos estarán de su media aritmética y mayor será el valor de la varianza; y a la inversa, cuánto más próximos entre sí, más cerca estarán de la media y menor será la varianza.

En R la varianza se calcula mediante la función *var()*:

```
> var(long)

[1] 15.12042
```

Desviación típica (o *Desviación estándar*). Es la raiz cuadrada de la varianza. Se obtiene así una medida de dispersión en las mismas unidades que la variable original:

$$s = \sqrt{s^2}$$

En R se obtiene con la función *sd()*:

```
> sd(long)

[1] 3.888498
```

Coeficiente de variación. La varianza y la desviación estándar son medidas de dispersión dependientes de las unidades en las que se mida la variable. El coeficiente de variación es una medida de dispersión adimensional que se define como:

$$cv(X) = \frac{s}{\overline{x}}$$

(siempre que $\overline{x} \neq 0$).

El coeficiente de variación resulta especialmente útil para comparar el grado de dispersión de variables que se miden en unidades diferentes. Por ejemplo si, en la muestra que estamos utilizando, queremos saber si los sargos presentan más dispersión en longitud o en peso, no tiene sentido comparar sus desviaciones típicas, medidas en centímetros y en gramos respectivamente. Sin embargo sus coeficientes de variación:

```
> sd(long)/mean(long)

[1] 0.1821669

> sd(peso)/mean(peso)

[1] 0.4552767
```

nos indican una mayor variabilidad en peso.

Rango y rango intercuartílico. El rango de una variable se define como la distancia entre los valores mínimo y máximo:

$$\text{rango}\,(X) = \text{máx}\,(X) - \text{mín}\,(X)$$

Asimismo, el rango intercuartílico es la distancia entre los cuartiles primero y tercero ($P_{75} - P_{25}$).

La función *range()* de R nos proporciona los valores mínimo y máximo de una variable. A su vez, como ya hemos visto, la función *quantile()* nos proporciona los cuartiles. La función *diff()* nos da la distancia entre valores:

```
> range(long)

[1]  9.74 30.65

> diff(range(long))
```

```
[1] 20.91

> quantile(long, probs = c(0.25, 0.75), names = FALSE)

[1] 18.84 23.98

> diff(quantile(long, probs = c(0.25, 0.75), names = FALSE))

[1] 5.14
```

6.4. Medidas de forma.

Coeficiente de asimetría. En los casos en que los datos estén distribuidos de forma simétrica, la media y mediana son medidas aproximadamente similares. Sin embargo, cuando los datos muestran largas colas a la derecha (valores altos muy alejados del resto de los datos), el valor de la media tenderá a ser mayor que el de la mediana. Así por ejemplo, para el conjunto de datos $\{1, 2, 2, 3, 3, 3, 4, 4, 5\}$ media y mediana coinciden en el valor 3. Por el contrario, si el conjunto de datos es $\{1, 2, 2, 3, 3, 3, 4, 4, 50\}$, la mediana sigue siendo el valor 3, mientras que la media aritmética se desplaza al valor 8. En estos casos, la mediana representa (localiza) mejor el centro de la distribución que la media aritmética.

Dada una muestra de una variable X formada por n observaciones, siendo $\overline{x}$ su media aritmética y s su desviación típica, la asimetría de la variable puede cuantificarse a través del *coeficiente de asimetría de Fisher*, definido como:

$$a_F = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{3/2}}$$

si bien en la práctica es preferible utilizar la siguiente versión corregida:

$$a_F = \frac{n\sqrt{(n-1)}}{n-2}\frac{\sum_{i=1}^{n}(x_i - \overline{x})^3}{\left(\sum_{i=1}^{n}(x_i - \overline{x})^2\right)^{3/2}} = \frac{n}{(n-1)(n-2)}\frac{\sum_{i=1}^{n}(x_i - \overline{x})^3}{s^3}$$

ya que esta última expresión tiende a producir valores más próximos a la asimetría de la variable en la población de la que se ha extraído la muestra. Cuando los datos son perfectamente simétricos este coeficiente es nulo. Cuando los valores se concentran a

la derecha, con largas colas a la izquierda este coeficiente es negativo (*asimetría a la izquierda o negativa*); y cuando los valores tienden a concentrarse a la izquierda, con largas colas a la derecha, el coeficiente es positivo (*asimetría a la derecha o positiva*).

El paquete base de R no contiene ninguna función para el cálculo del coeficiente de asimetría. Podríamos construir una función para su cálculo, pero en este caso ya existen varios paquetes que lo hacen, entre ellos el paquete *agricolae* que ya hemos usado con anterioridad. Para calcular la asimetría utilizamos la función *skewness()*:

```
> require(agricolae)
> skewness(ldors)

[1] -0.3480565

> skewness(phig)

[1] 1.400168
```

Como vemos, la distancia desde el morro del pez a la aleta dorsal (ldors) presenta asimetría negativa y el peso del hígado (phig) asimetría positiva. En la figura 7 podemos observar los histogramas de ambas variables y comprobar que son efectivamente asimétricos.

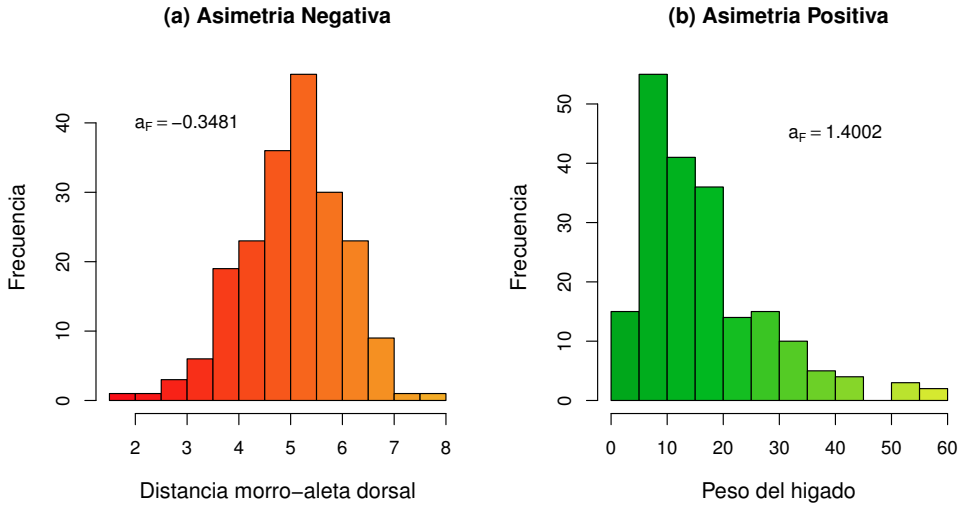

Figura 7: Variables que presentan asimetría (a) Histograma de la distancia del morro a la aleta dorsal (asimetría negativa) (b) Histograma del peso del hígado (asimetría positiva).

Nota: en el fragmento de código anterior hemos usado la función *require()*. Esta función comprueba si una librería –en este caso *agricolae*– ha sido ya cargada mediante *library()*. Si la librería ya ha sido cargada, *require()* no hace nada, y en caso contrario carga la librería.

Coeficiente_de_apuntamiento_(curtosis): mide el grado de concentración que presentan los valores alrededor de la zona central del conjunto de datos. La definición habitual de curtosis es:

$$\kappa = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^2} - 3$$

si bien, como ocurre con la asimetría, en la práctica se emplea una versión corregida (cuando n es grande produce prácticamente el mismo valor que la anterior, pero para valores de n pequeños tiende a producir valores de curtosis más próximos al verdadero valor en la población de la que se ha extraído la muestra):

$$\kappa = \frac{n\,(n+1)}{(n-1)\,(n-2)\,(n-3)}\frac{\sum_{i=1}^{n}(x_i - \bar{x})^4}{S^4} - 3\frac{(n-1)^2}{(n-2)\,(n-3)}$$

Si $\kappa > 0$ la forma del conjunto de datos es "puntiaguda" (*leptocúrtica*); por el contrario, si $\kappa < 0$, la forma es "aplastada" (*platicúrtica*). El caso $\kappa = 0$ corresponde a una forma "normal" (*mesocúrtica*), ni muy apuntada ni muy aplastada.

Al igual que ocurría con la asimetría, R no dispone en su paquete base de ninguna función para el cálculo de la curtosis, si bien podemos encontrarla en el paquete *agricolae*:

```
> kurtosis(ldors)

[1] 0.2372677

> kurtosis(phig)

[1] 2.168432
```

Como vemos, ambas variables presentan apuntamiento positivo (corresponden a distribuciones leptocúrticas), tal como podemos apreciar visualmente en los histogramas mostrados en la figura 7).

6.5. Valores perdidos.

En muchas ocasiones no se dispone de los valores de todas las variables, bien sea porque no se han podido medir sobre los objetos de la muestra, bien sea porque dichos valores no quedaron registrados en el archivo de datos. En cualquier caso, cuando R encuentra un espacio en blanco en una posición del archivo en la que esperaba encontrar un dato, considera que ese valor está perdido y lo codifica internamente como *NA* (*No Asignado*). A veces cuando un valor de la muestra se ha perdido, en lugar de dejar un espacio en blanco en el archivo de datos, se consigna con un valor identificativo (-1, 9999, "*",...). En tal caso, al leer el archivo hay que indicar a R que ese valor representa un valor perdido mediante la opción *na.strings*. Si, por ejemplo, los valores perdidos se identificaran con 9999, en el comando de lectura deberíamos especificar, junto a las opciones ya vistas en la sección 4.1:

```
> MisDatos = read.table(..., na.strings = "9999", ...)
```

La presencia de valores perdidos afecta a las funciones que calculan las medidas de síntesis (*mean, sd, quantile*, etc). Recordemos que en nuestro archivo de ejemplo, el peso de las gónadas no se había medido para todos los peces. Si quisiéramos calcular el peso medio de las gónadas obtendríamos:

```
> mean(pgon)

[1] NA
```

lo que indica que R no ha podido calcularlo debido a la presencia de valores perdidos. En realidad R sí que puede calcular el peso medio, y el hecho de que no lo calcule directamente significa más bien un aviso para que tengamos en cuenta la presencia de tales valores. Para calcular la media (o cualquier otra medida de síntesis) en estas condiciones, hay que añadir la opción *na.rm=TRUE* (acrónimo de *NA remove*):

```
> mean(pgon, na.rm = T)

[1] 11.48706
```

Nota: Bajo determinadas condiciones la existencia de valores perdidos (sobre todo si éstos constituyen una parte importante de la muestra) podría dar lugar a que la muestra no fuese realmente representativa de la población de la que se ha extraido y por tanto el análisis estadístico que hagamos de la misma tendría escaso valor.

6.6. Diagrama de cajas y barras (*boxplot*)

Estos diagramas representan los percentiles de una variable y son especialmente útiles para una comparación gráfica de varias poblaciones, así como para la detección de posibles valores anómalos (*outliers*). Su construcción se realiza de la siguiente forma: sea $\{x_1, \cdots, x_n\}$ el conjunto de datos correspondientes a una variable numérica X, y representemos por P_{25}, P_{50} y P_{75} los percentiles 25, 50 y 75 respectivamente; se dibuja un rectángulo vertical cuyos lados inferior y superior corresponden a P_{25} (primer cuartil) y P_{75} (tercer cuartil) respectivamente; a la altura P_{50} (mediana) se traza un segmento horizontal. Por último el rectángulo se une mediante lineas a dos barras correspondientes los extremos de la distribución, trazadas a alturas respectivas b y B:

1. *Barra superior*: $B = \text{mín}\left\{\text{máx}\left(X\right),\ P_{75} + 1{,}5\left(P_{75} - P_{25}\right)\right\}$
2. *Barra inferior*: $b = \text{máx}\left\{\text{mín}\left(X\right),\ P_{25} - 1{,}5\left(P_{75} - P_{25}\right)\right\}$

Los valores de los datos que quedan fuera de las barras superior e inferior se marcan con puntos y se entenderá que pueden ser anómalos, y deben ser revisados por si constituyeran errores de medida, datos correspondientes a otra población, etc.

Para obtener en R el boxplot de la variable longitud, por ejemplo, ejecutaríamos simplemente la función:

```
> boxplot(long, col = "orange", main = "longitud")
```

6.7. Medidas de síntesis en subgrupos de la muestra.

En muchas ocasiones los objetos de la muestra pueden clasificarse según los valores de alguna variable categórica. Así, en los datos de nuestro ejemplo, podríamos clasificar los sargos en función de la isla de procedencia, o en función de su sexo. En la sección 5.1 ya hemos visto como construir tablas cruzadas para esta clase de variables. Cuando lo que nos interesa es calcular las distintas medidas de síntesis sobre cada uno de los grupos que forman la muestra, en R podemos utilizar los comandos *by()* y *aggregate()*.

Así, por ejemplo, para calcular la longitud media de los sargos según sexo usaríamos la función:

```
> by(long, sexo, mean)
```

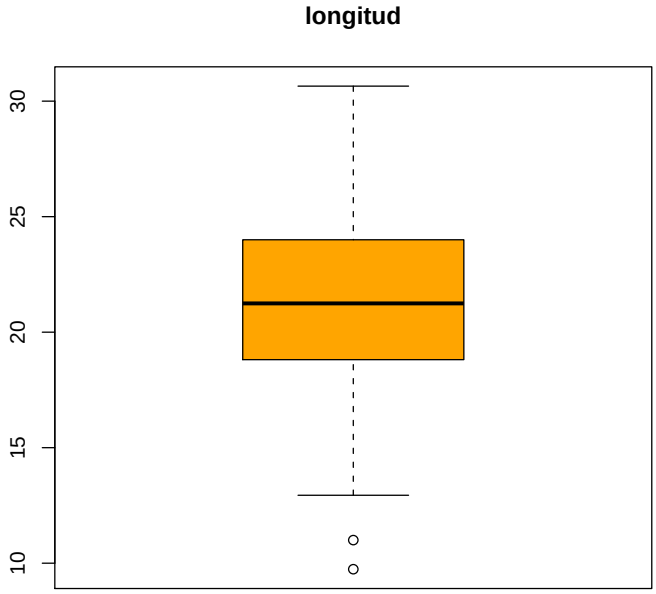

Figura 8: Diagrama de cajas y barras para la variable longitud.

```
sexo: Hembra
[1] 20.84080
------------------------------------------------------
sexo: Macho
[1] 22.00172
```

o de manera equivalente:

```
> aggregate(long, by = list(sexo), mean)

  Group.1        x
1  Hembra 20.84080
2   Macho 22.00172
```

La presentación de la tabla construida con el comando *aggregate()* mejora si:

- La variable (o variables, ya que pueden incluirse varias) a resumir se especifica como subconjunto (*subset()*) del conjunto de datos original.
- La variable (o variables, también podrían incluirse varias) que define los grupos se *renombra* dentro del comando *list()*.

Veamos el efecto de estos cambios, calculando la longitud y el peso medios por sexo y por isla en nuestra muestra:

```
> aggregate(subset(sargos, select = c(long, peso)), by = list(Sexo = sexo,
     Isla = isla), mean)

     Sexo Isla     long     peso
1  Hembra   HI 20.98250 156.9800
2   Macho   HI 22.78571 188.4914
3  Hembra   LP 20.46750 146.8017
4   Macho   LP 23.72500 216.5800
5  Hembra   LG 21.11167 158.4017
6   Macho   LG 22.08667 169.3333
7  Hembra   TF 21.77286 176.5952
8   Macho   TF 21.82632 174.2589
9  Hembra   GC 20.66786 152.8236
10  Macho   GC 22.39400 185.4225
11 Hembra   FV 20.07000 144.1612
12  Macho   FV 21.02563 161.7181
13 Hembra   LZ 20.81000 155.5855
14  Macho   LZ 20.47000 149.0160
```

Si quisiéramos calcular varias medidas de síntesis sobre los subgrupos de la muestra debemos definir una función con las medidas a calcular; así, por ejemplo, si de cada variable quisiéramos obtener la media, desviación típica, mínimo y máximo, construiríamos la función de resumen siguiente:

```
> resumen = function(x, ...) {
     m = mean(x, ...)
     s = sd(x, ...)
     mn = min(x, ...)
     mx = max(x, ...)
     output = round(c(m, s, mn, mx), 2)
     names(output) = c("media", "sd", "min", "max")
     return(output)
 }
```

Nota: los puntos sucesivos permiten que la función reciba otras opciones; por ejemplo, si al llamarla añadiésemos *na.rm=T* podríamos calcular todas las medidas de síntesis especificadas en presencia de valores perdidos.

Utilizamos esta función para resumir la variable peso según sexo:

```
> by(peso, sexo, resumen)

sexo: Hembra
 media     sd    min    max
156.50  73.00  27.09 371.89
------------------------------------------------
sexo: Macho
 media     sd    min    max
178.43  77.51  18.04 382.18
```

O, utilizando *aggregate()* para el peso del hígado, teniendo en cuenta la presencia de valores perdidos:

```
> aggregate(subset(sargos, select = phig), by = list(Sexo = sexo),
     resumen, na.rm = T)

    Sexo phig.media phig.sd phig.min phig.max
1 Hembra      15.36   11.66     1.70    59.00
2  Macho      18.06   10.43     0.70    55.00
```

Para concluir esta sección citemos que es posible utilizar la función *boxplot()* para hacer diagramas de cajas y barras según subgrupos de la muestra. El siguiente código genera los gráficos mostrados en la figura 9

```
> boxplot(peso ~ sexo, main = "Peso", col = c("pink2",
     "cyan3"))
> boxplot(peso ~ isla, main = "Peso", col = heat.colors(14))
```

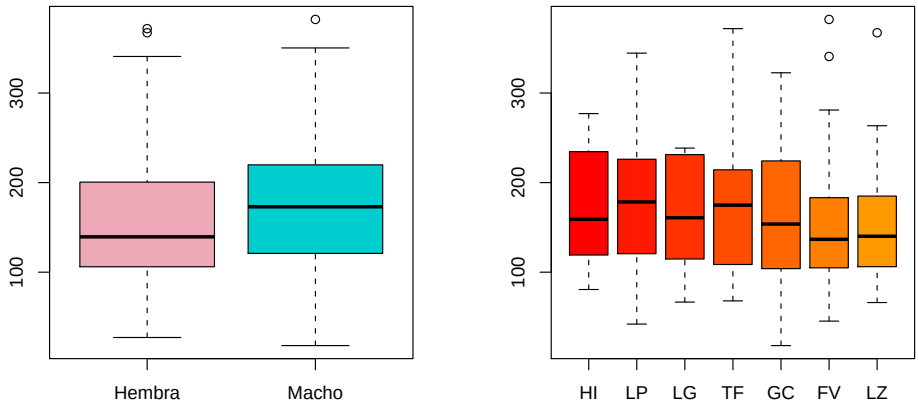

Figura 9: Boxplots para subgrupos de la muestra. Izquierda: peso según sexo. Derecha: peso según isla.

7. Asociación entre variables continuas.

En la sección 5.2 hemos llevado a cabo la descripción de datos correspondientes a variables continuas: tablas de frecuencias, histogramas y polígonos de frecuencias. Asimismo, en la sección 6 hemos presentado las medidas de síntesis que nos permiten resumir las características de estas variables en unos pocos valores. En ambos casos, el análisis de los datos ha sido univariante: cada variable se estudia aisladamente, sin conexión con las restantes variables continuas medidas en la muestra. Todo lo más, en 6.7 hemos visto como varía una variable continua en varios grupos definidos por una variable categórica.

Ahora bien, cuando se realiza el estudio conjunto de dos variables, normalmente el objetivo es determinar si existe algún tipo de asociación entre ellas o si, por el contrario, son independientes. En términos prácticos, la asociación significa que el conocimiento de los valores de una de las variables proporciona alguna información sobre los valores de la otra. Por ejemplo, conocer la estatura de una persona nos informa sobre su peso, ya que las personas más altas tienen, en general, un peso mayor que las personas más bajas. Esta asociación estadística, obviamente no es exacta: dos personas de la misma altura no tienen que tener exactamente el mismo peso, y una persona más alta puede pesar menos que una más baja. La figura 10 ilustra este tipo de asociación: valores altos de X tienden a ir acompañados de valores altos de Y, a la vez que valores bajos de X tienden a ir acompañados de valores bajos de Y, si bien no de manera exacta.

Al estudiar la asociación entre variables continuas podemos encontrarnos ante dos problemas distintos, según cuál sea el objetivo de nuestro estudio:

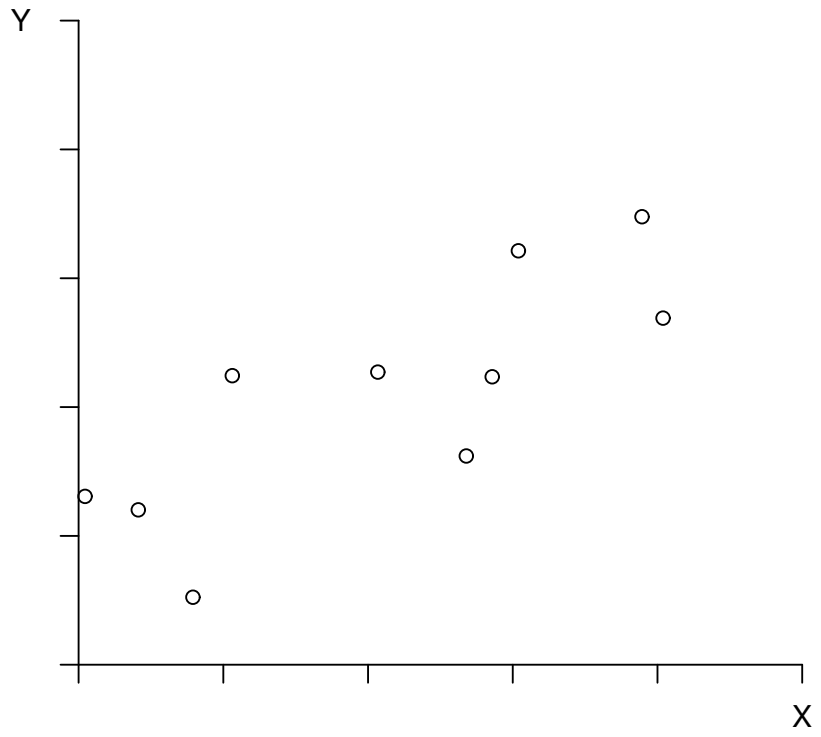

Figura 10: Nube de puntos correspondiente a la observación de dos variables X e Y sobre los sujetos de una muestra.

Análisis de regresión: nuestro objetivo es construir un modelo para *predecir* el valor de una variable Y cuando se conoce el valor de otra variable X. Esto es, si para el sujeto i-ésimo de la muestra sabemos que $X = x_i$, queremos hallar una función f tal que el valor de Y predicho para ese sujeto sea $y_i = f(x_i) + \varepsilon_i$. Los términos ε_i representan los *errores de predicción.* Cuando la función $f(X)$ es lineal nos hallamos ante un problema de *regresión lineal.* En caso contrario estaríamos ante un problema de *regresión no lineal.*

Análisis de correlación: nuestro objetivo es medir la intensidad de la asociación lineal entre dos variables X e Y. Una correlación alta indicaría una fuerte asociación y una correlación baja, una asociación débil. Las variables son tratadas de forma simétrica, no hay una variable predictora y una variable a predecir.

En un análisis de correlación ambas variables X e Y son *aleatorias*, lo que significa que sus valores no se conocen hasta haberlas observado. El observador usa la correlación para medir la asociación entre estas variables tal como se produce en la naturaleza. En la muestra que venimos utilizando como ejemplo, para cada sargo se mide su longitud y su peso; antes de

tomar la muestra estos valores son desconocidos, por lo que ambas variables son aleatorias. Sin embargo, en un análisis de regresión, si bien ambas variables pueder ser también aleatorias, es frecuente que el observador (o experimentador) fije de antemano los valores de la variable X y mida a continuación como responde la variable Y, que sería en tal caso la única aleatoria. Es importante señalar que en estas condiciones la asociación que se produzca entre X e Y puede ser muy distinta de la que se observa en condiciones naturales.

Nota: tanto en el caso de la regresión como en el de la correlación *no debe confundirse asociación con causalidad*. Podemos usar una regresión para predecir la edad de un niño a partir de su estatura, ya que niños más altos probablemente tienen mayor edad; pero evidentemente, la altura *no es la causa* de la edad. Podemos detectar una correlación –asociación– fuerte entre altos niveles de glucosa en sangre e hipertensión; sin embargo ello no quiere decir que la diabetes cause la hipertensión o que la hipertensión cause la diabetes; no puede descartarse la posibilidad de que exista una causa común –en este caso, el *síndrome metabólico*– que sea en realidad la que da lugar a la asociación entre ambas enfermedades.

Sólo los estudios experimentales pueden probar de manera concluyente una posible relación causal entre dos variables: en estos estudios el experimentador controla todos los posibles factores de confusión (terceras variables que puedan influir en la asociación) y las posibles fuentes de "ruido" en los datos; si en tales condiciones la modificación de X produce un cambio en Y, y se cuenta además con un mecanismo para explicar como se produce tal efecto, entonces y sólo entonces se puede hablar de causalidad, o al menos de influencia de X sobre Y.

7.1. Regresión lineal.

Una de las formas más comunes de asociación entre variables es la asociación lineal. Los valores representados en la figura 10 muestran precisamente este tipo de asociación. En la práctica resulta de interés determinar la ecuación de la recta que define esta relación y que permite aproximar el valor de Y cuando se conoce el valor de X. Esta recta se denomina *recta de regresión de Y sobre X*, y su ecuación es de la forma $Y = b_0 + b_1 X$.

La variable X recibe el nombre de *variable explicativa* (o *independiente*) y la Y el de *variable respuesta* (o *dependiente*). El valor de b_1 es la *pendiente* y b_0 es la *ordenada en el origen*. La pendiente representa el incremento (si b_1 es positivo) o decremento (si b_1 es negativo) que experimenta el valor promedio de Y por cada unidad de incremento en el valor de X.

Asimismo, la ordenada en el origen b_0 es el valor de Y cuando $X = 0$. Hay que señalar que, desde el punto de vista del análisis de los datos, esta interpretación solo debe realizarse cuando el valor $X = 0$ ha sido efectivamente observado. Si, por ejemplo, Y fuese el peso de una persona de altura X y se dispusiera de una recta de regresión $Y = b_0 + b_1X$ que relacionase ambas variables, dado que no existen personas de estatura $X = 0$ no tiene sentido decir que b_0 es el peso aproximado de tales personas.

Para calcular la recta de regresión de Y sobre X se utiliza habitualmente el *método de los mínimos cuadrados*. Supongamos que sobre una muestra de n objetos hemos medido el par de variables (X, Y), y que los valores observados han sido $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. Supongamos además que estos puntos se encuentran alineados a lo largo de una recta de ecuación $Y = b_0 + b_1X$, y llamemos $\hat{y}_i = b_0 + b_1x_i$ al valor que corresponde sobre la recta al punto x_i (*valor predicho por la recta*). El error de predicción sería entonces $e_i = y_i - \hat{y}_i$. El criterio de los mínimos cuadrados consiste en determinar los valores de b_0 y b_1 de forma que la suma de distancias al cuadrado entre observaciones y predicciones sea mínima, esto es:

$$\text{mín} \sum_{i=1}^{n} e_i^2 = \text{mín} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \min_{b_0,b_1} \sum_{i=1}^{n} (y_i - (b_0 + b_1x_i))^2$$

De esta forma se consigue que la recta pase simultáneamente lo más cerca posible de todos los puntos observados. La figura 11 ilustra gráficamente esta idea.

Llamemos:

$$L(b_0, b_1) = \sum_{i=1}^{n} (y_i - b_0 - b_1x_i)^2$$

Para obtener los valores de b_0 y b_1 que minimizan esta expresión derivamos con respecto a b_0 y a b_1 e igualamos a 0, obteniendo las llamadas ecuaciones normales de mínimos cuadrados:

$$\begin{array}{rcccl} \frac{\partial L(b_0,b_1)}{\partial b_0} & = & -2\sum_{i=1}^{n} (y_i - b_0 - b_1x_i) & = & 0 \\ \frac{\partial L(b_0,b_1)}{\partial b_1} & = & -2\sum_{i=1}^{n} (y_i - b_0 - b_1x_i)x_i & = & 0 \end{array}$$

De la primera ecuación se tiene:

$$-2\sum_{i=1}^{n} (y_i - b_0 - b_1x_i) = 0 \Rightarrow \sum_{i=1}^{n} (y_i - b_0 - b_1x_i) = 0 \Rightarrow \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} b_0 - \sum_{i=1}^{n} b_1x_i = 0$$
$$\Rightarrow \sum_{i=1}^{n} y_i - nb_0 - b_1 \sum_{i=1}^{n} x_i = 0 \Rightarrow b_0 = \frac{\sum_{i=1}^{n} y_i}{n} - b_1 \frac{\sum_{i=1}^{n} x_i}{n} \Rightarrow b_0 = \bar{y} - b_1\bar{x}$$

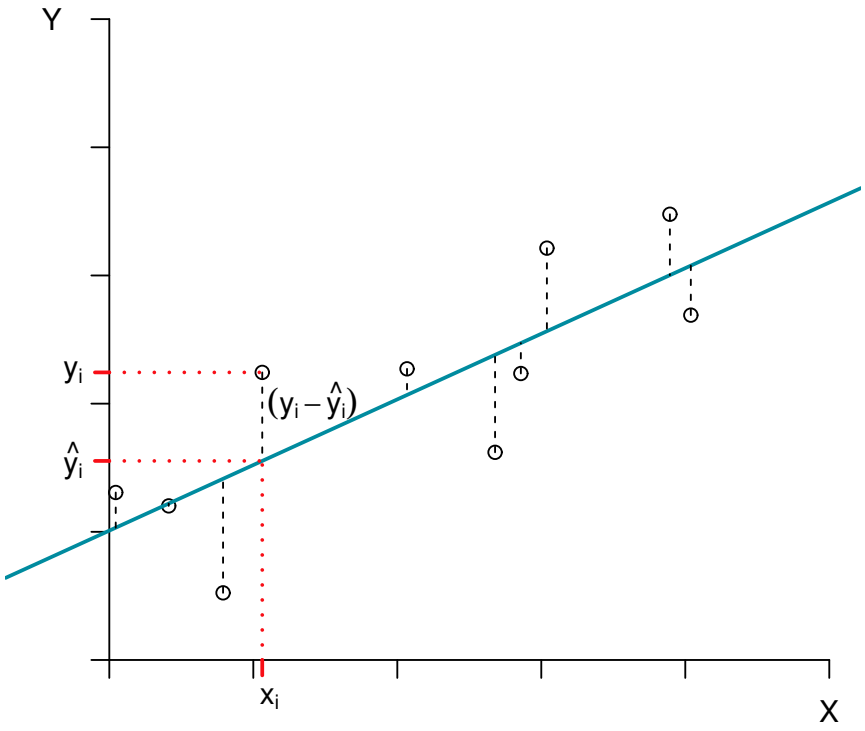

Figura 11: Recta de regresión ajustada a la nube de puntos de la figura 10. Las lineas a trazos verticales representan las distancias de los puntos a la recta. El método de los mínimos cuadrados busca la recta que minimiza la suma de los cuadrados de estas distancias.

Sustituyendo en la segunda ecuación:

$$-2\sum_{i=1}^{n}(y_i - b_0 - b_1 x_i)x_i \;=\; 0 \Rightarrow \sum_{i=1}^{n}(y_i - (\bar{y} - b_1\bar{x}) - b_1 x_i)x_i = 0 \Rightarrow$$

$$\sum_{i=1}^{n}(y_i - \overline{y})\,x_i - b_1\sum_{i=1}^{n}(x_i - \overline{x})\,x_i = 0 \Rightarrow b_1 = \frac{\sum_{i=1}^{n}(y_i - \overline{y})\,x_i}{\sum_{i=1}^{n}(x_i - \overline{x})\,x_i} = \frac{\sum_{i=1}^{n} y_i x_i - \overline{y}\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i^2 - \overline{x}\sum_{i=1}^{n} x_i}$$

Si tenemos en cuenta que:

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

podemos sustituir en la expresión anterior y nos queda:

$$b_1 = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2}$$

Una vez obtenido el valor de b_1, el valor de b_0 se despeja de:

$$b_0 = \bar{y} - b_1\bar{x}$$

En R es muy sencillo obtener la recta de regresión. La siguiente sintaxis muestra como calcular la recta de regresión entre la longitud total del pez y la distancia desde el morro a la aleta dorsal:

```
> lm(peso ~ long)

Call:
lm(formula = peso ~ long)

Coefficients:
(Intercept)         long
    -236.20        18.84
```

El valor indicado como intercept es la ordenada en el origen b_0, mientras que el valor bajo el nombre de la variable es la pendiente b_1. Para representar esta recta gráficamente podemos utilizar la siguiente sintaxis, cuyo resultado se muestra en la figura 12.

```
> plot(long, ldors, xlab = "Longitud total", ylab = "Distancia morro-aleta dorsal",
     main = "Regresión Longitud-Distancia a la aleta dorsal")
> recta = lm(ldors ~ long)
> abline(recta, col = "darkgreen", lwd = 2)
```

Con R es posible dibujar en un mismo gráfico nubes de puntos correspondientes a distintos grupos de datos, mostrando el ajuste de regresión para cada uno. Por ejemplo, la siguiente sintaxis repite el gráfico anterior pero dibujando de color distinto machos y hembras, y ajustando una recta de regresión a cada grupo:

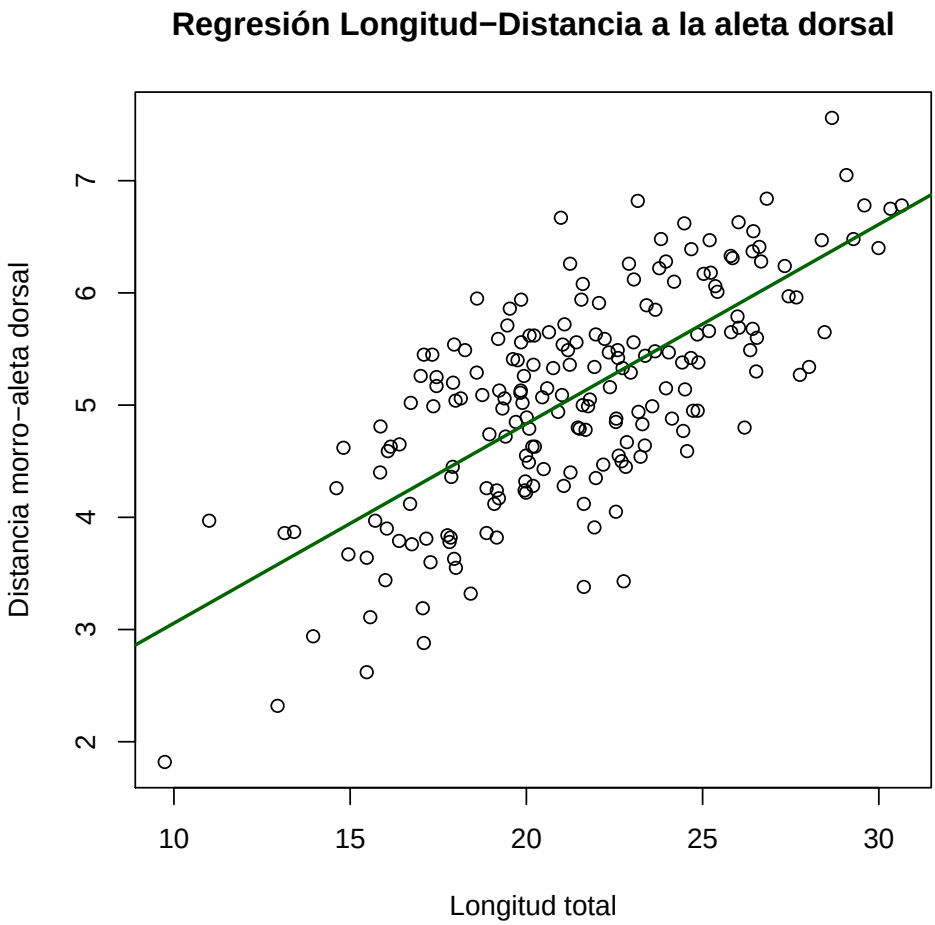

Figura 12: Recta de regresión para la distancia desde el morro a la aleta dorsal frente a la longitud total del pez.

```
> plot(long, ldors, xlab = "Longitud total", ylab = "Distancia morro-aleta dorsal",
     main = "Regresión Longitud-Distancia a la aleta dorsal",
     type = "n")
> with(subset(sargos, sexo == "Hembra"), {
     points(long, ldors, col = "pink3", pch = 19)
     abline(lm(ldors ~ long), col = "pink3", lwd = 2)
 })
> with(subset(sargos, sexo == "Macho"), {
     points(long, ldors, col = "cyan4", pch = 19)
     abline(lm(ldors ~ long), col = "cyan4", lwd = 2)
 })
> legend("topleft", c("Hembra", "Macho"), col = c("pink3",
     "cyan4"), pch = 19, lty = 2, bty = "n")
```

El resultado de esta sintaxis se muestra en la figura 13 .

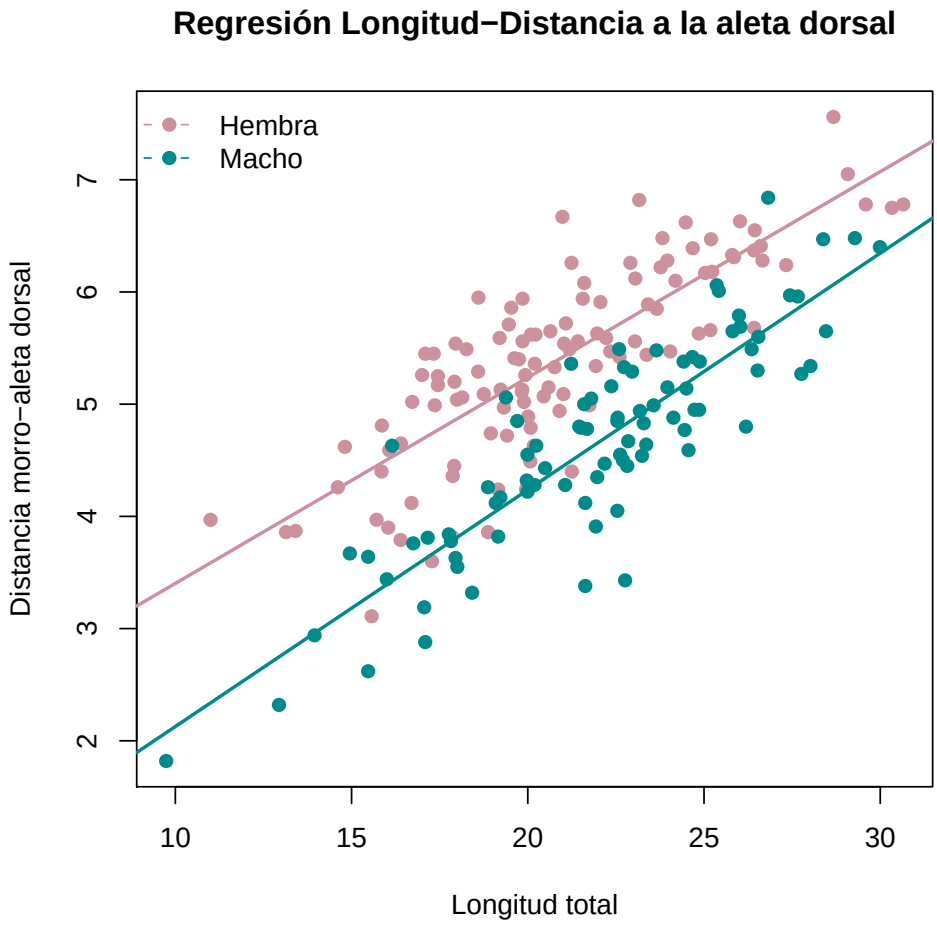

Figura 13: Rectas de regresión para la distancia desde el morro a la aleta dorsal frente a la longitud total del pez, ajustadas para cada sexo.

Nota: el paquete *lattice* contiene funciones gráficas de alto nivel que permiten construir este gráfico (y otros más complejos) de manera mucho más simple.

Si queremos obtener los valores numéricos de las ecuaciones de ambas rectas bastará con ejecutar:

```
> lm(ldors ~ long, data = subset(sargos, sexo == "Hembra"))

Call:
lm(formula = ldors ~ long, data = subset(sargos, sexo == "Hembra"))
```

```
Coefficients:
(Intercept)         long
     1.5677       0.1835

> lm(ldors ~ long, data = subset(sargos, sexo == "Macho"))

Call:
lm(formula = ldors ~ long, data = subset(sargos, sexo == "Macho"))

Coefficients:
(Intercept)         long
    0.01804      0.21091
```

7.2. Covarianza y correlación

La figura 14 nos muestra dos nubes de puntos. Se aprecia claramente que los datos de la nube (a) muestran una asociación lineal muy fuerte, mientras que en la nube (b) esta asociación es más débil.

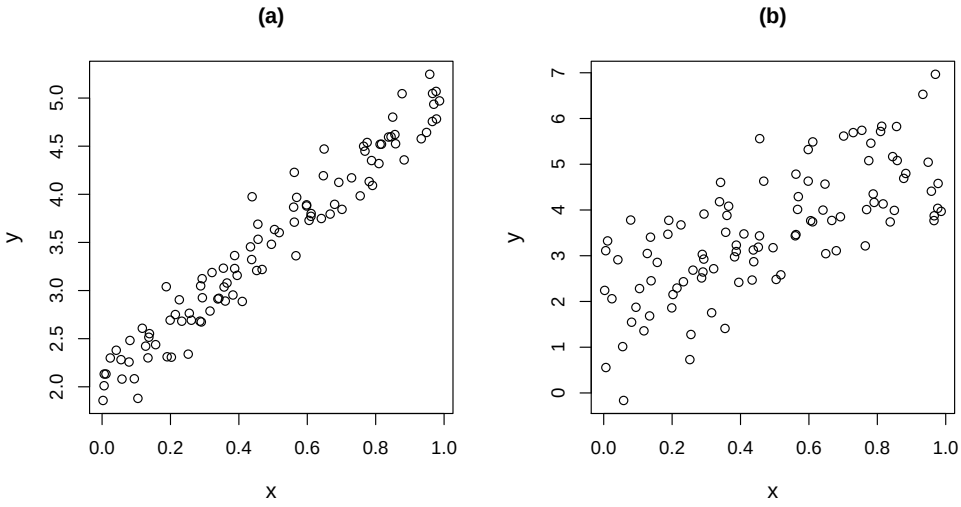

Figura 14: Nubes de puntos con distintos grado de asociación (a) Asociación lineal fuerte. (b) Asociación lineal débil.

Para medir numéricamente la intensidad de la asociación lineal entre dos variables se utiliza

la *covarianza*, definida como:

$$S_{XY} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i y_i - n \cdot \bar{x}\,\bar{y}\right)$$

Esta medida es positiva si los datos presentan tendencia lineal creciente; es negativa si presentan tendencia lineal decreciente; y es nula si los datos no presentan tendencia lineal.

Nota: La ausencia de tendencia lineal no significa que no exista algún otro tipo de asociación (no lineal) entre X e Y.

La figura 15 muestra cuatro nubes de puntos con distinta covarianza. La figuras (a) y (b) presentan asociación lineal, el caso (a) con pendiente positiva, y por tanto con covarianza positiva, y el caso (b) con pendiente (y por tanto covarianza) negativa. A su vez las figuras (c) y (d) presentan covarianza nula; en el caso (a) porque no existe asociación entre X e Y, y en el caso (d) porque, aún existiendo asociación, esta es claramente no lineal.

La covarianza, como medida de la asociación lineal entre variables presenta un problema práctico: depende de las unidades de X e Y, y por tanto su magnitud, en términos absolutos, sea grande o pequeña puede depender más de las escalas de medida que de la fuerza de la asociación lineal entre ambas variables (por ejemplo, si X e Y son longitudes, el valor de la covarianza entre ambas será un número mucho mayor si X e Y se miden en centímetros que si se miden en metros). Por tanto es preciso introducir una nueva medida de asociación lineal que no dependa de las unidades de X e Y. Esta medida es el *coeficiente de correlación de Pearson,* definido como:

$$r = \frac{S_{XY}}{S_X S_Y}$$

siendo S_X y S_Y las desviaciones típicas respectivas de las variables X e Y. Como éstas son siempre positivas, es obvio que el signo de r coincide con el signo de S_{XY}. Además, se cumple que:

$$-1 \leq r \leq 1$$

siendo el valor absoluto de r igual a 1 cuando los puntos están *exactamente* sobre una recta. La figura 16 muestra cuatro nubes de puntos con distintos valores de correlación lineal.

Así pues:

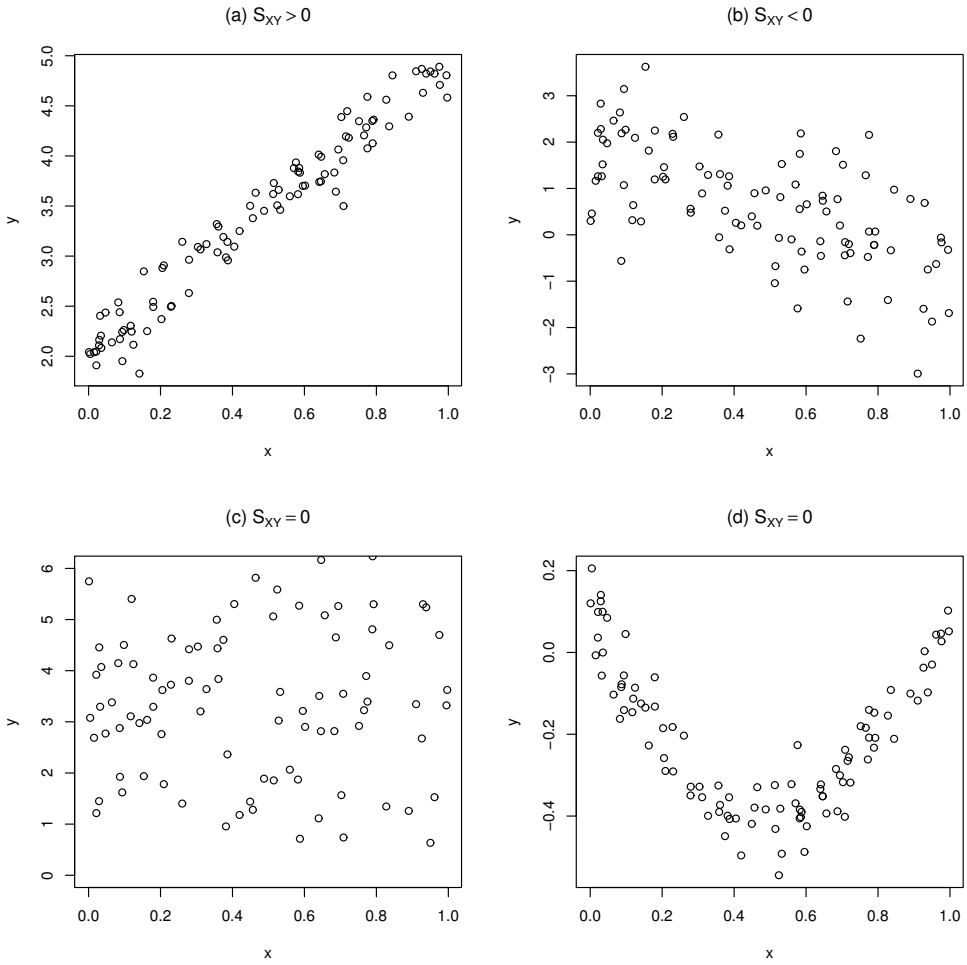

Figura 15: Nubes de puntos con distinta covarianza.

- $r > 0$: indica la presencia de una asociación lineal positiva (recta creciente). Esta asociación es tanto más fuerte (más se ajustan los puntos a la recta) cuanto más se aproxime el valor de r a 1.

- $r < 0$: indica la presencia de una asociación lineal negativa (recta decreciente); cuando aumenta el valor de X, el valor de Y disminuye proporcionalmente). Cuanto más se aproxime r a -1 tanto mejor es el ajuste a una recta.

- $r = 0$: indica la ausencia de asociación lineal entre X e Y: podría haber una ausencia absoluta de asociación como en la figura 15(c), o bien podría existir algún tipo de relación no lineal como en la figura 15(d).

Para determinar si el coeficiente de correlación es una medida adecuada de la asociación entre variables, el primer paso debe ser siempre dibujar un gráfico de la nube de puntos correspondiente a las observaciones. En los siguientes casos no es apropiado utilizar el coeficiente de

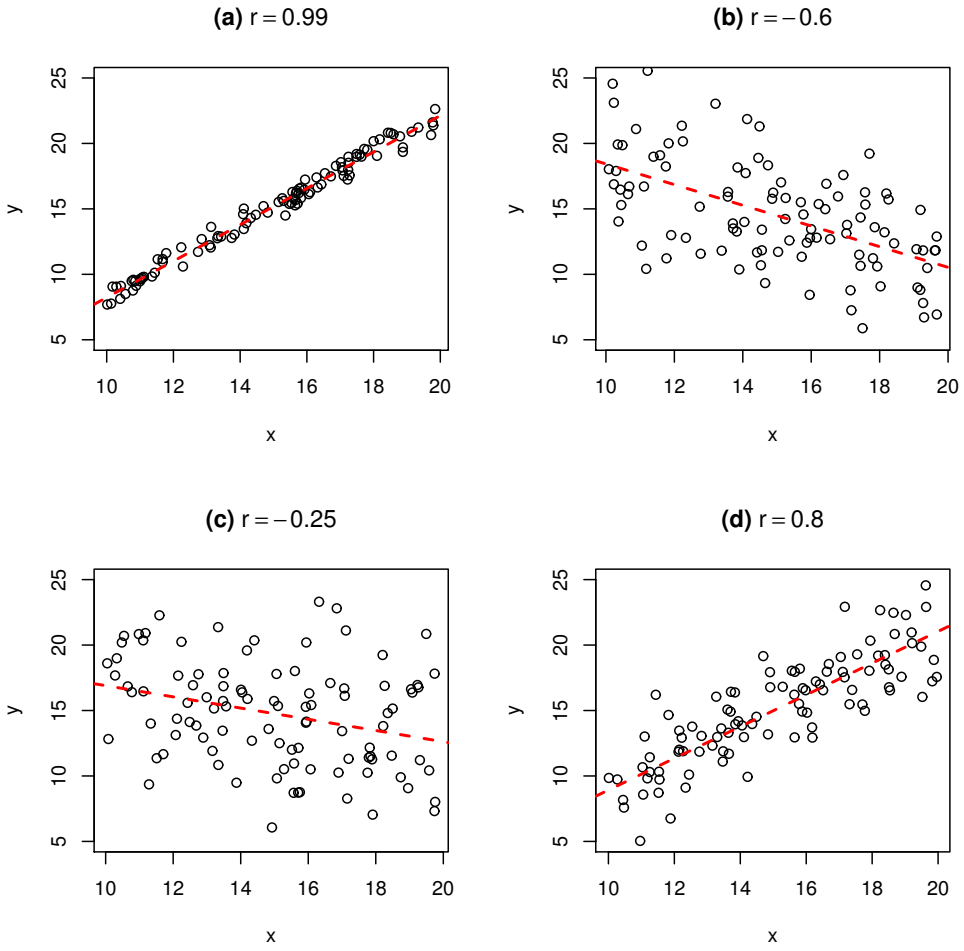

Figura 16: Nubes de puntos con distintos valores de correlación lineal.

correlación:

1. **La relación entre las variables es no lineal:** se observa que los puntos se distribuyen a lo largo de alguna figura geométrica regular distinta de una recta. En este caso lo mejor es tratar de encontrar el modelo matemático que mejor se ajusta a las observaciones. Ello puede significar utilizar, por ejemplo, regresión lineal múltiple (regresión lineal con varias variables independientes) o regresión no lineal. En la figura 17(a) vemos un ejemplo de esta situación. El coeficiente de correlación es alto (0.888), pero la nube de puntos tiene una forma claramente no lineal.

2. **Presencia de valores anómalos (outliers)**: El coeficiente de correlación debe usarse con precaución en presencia de estos valores. Gráficamente, un outlier es un punto que se aparta notoriamente del cuerpo principal de las observaciones y puede incrementar o disminuir artificialmente el valor de r. Así en la figura 17(b) vemos un caso en que hay

una nube de puntos con un ajuste lineal muy bueno. Un único valor alejado de esa nube da lugar a que la correlación sea prácticamente nula (incluso ligeramente negativa, aún cuando la tendencia de la nube de puntos es creciente). En la figura 17(c) vemos la situación contraria: una nube de puntos que no presenta asociación, y un punto aislado; la correlación global de este conjunto de puntos es, sin embargo, muy alta, 0.9.

3. **Presencia de grupos distintos de datos.** El coeficiente de correlación también debe usarse con precaución cuando las variables se miden sobre varios grupos distintos, ya que la correlación global puede llegar a diferir mucho de la correlación en cada grupo. En la imagen mostrada en la figura 17(d) se aprecia que hay dos grupos de datos, cada uno de ellos con una fuerte correlación negativa. Sin embargo, cuando la correlación se calcula globalmente para todos los puntos, sin distinguir grupos, se obtiene un valor positivo relativamente alto (0.743).

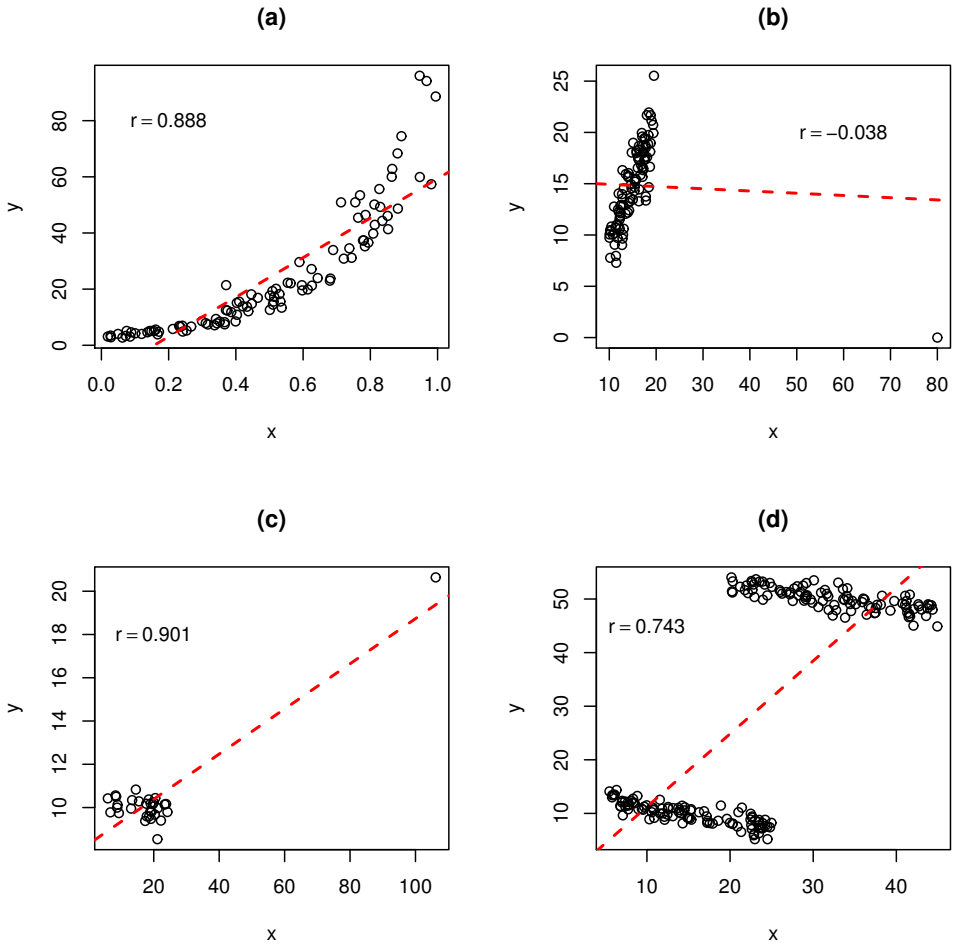

Figura 17: Diversos casos en que el coeficiente de correlación no resulta apropiado como medida de ajuste lineal.

En R la covarianza se calcula mediante la función *cov()* y la correlación mediante *cor()*. Veamos un ejemplo:

```
> cov(long, ldors)

[1] 2.686969

> cor(long, ldors)

[1] 0.7150845
```

Estas funciones pueden aplicarse a más de dos variables, en cuyo caso se obtienen las correspondientes matrices de covarianzas o correlaciones:

```
> cov(data.frame(long, ldors, lpect, peso))

            long      ldors      lpect       peso
long   15.120419  2.6869694  3.6571723  284.92959
ldors   2.686969  0.9337849  0.6619590   50.11847
lpect   3.657172  0.6619590  0.9677932   69.29353
peso  284.929587 50.1184671 69.2935278 5714.58082

> cor(data.frame(long, ldors, lpect, peso))

           long     ldors     lpect      peso
long  1.0000000 0.7150845 0.9560315 0.9693117
ldors 0.7150845 1.0000000 0.6963321 0.6860917
lpect 0.9560315 0.6963321 1.0000000 0.9317710
peso  0.9693117 0.6860917 0.9317710 1.0000000
```

Podemos calcular correlaciones y covarianzas en grupos separados de datos utilizando la función *by* de modo similar a como hemos visto ya en 6.7. La siguiente sintaxis nos proporciona la correlación entre longitud y peso para cada sexo:

```
> by(data.frame(long, peso), sexo, cor)

sexo: Hembra
          long     peso
long 1.000000 0.976949
peso 0.976949 1.000000
```

```
------------------------------------------------------
sexo: Macho
          long     peso
long 1.000000 0.958976
peso 0.958976 1.000000
```