Lección 7

Introducción a la estadística descriptiva

Habitualmente, dispondremos de una serie de datos que describirán algunos aspectos de un conjunto de individuos, y querremos analizarlos. El análisis estadístico de estos datos puede ser de dos tipos básicos:

- *Análisis exploratorio*, o *descriptivo*, si nuestro objetivo es resumir, representar y explicar los datos concretos de los que disponemos. La *estadística descriptiva* es el conjunto de técnicas que se usan con este fin.
- *Análisis inferencial*, si nuestro objetivo es deducir (*inferir*), a partir de estos datos, información significativa sobre el total de la población o las poblaciones de interés. Las técnicas que se usan en este caso forman la *estadística inferencial*.

Naturalmente, ambos análisis están relacionados. Así, por un lado, cualquier análisis inferencial se suele empezar explorando los datos que se usarán, y por otro, muchas técnicas descriptivas permiten estimar propiedades de la población de la que se ha extraído la muestra. Por citar un ejemplo de esto último, la media aritmética de las alturas de una muestra de individuos nos da un valor representativo de esta muestra, pero también *estima* la media de las alturas del total de la población.

En las lecciones que siguen explicaremos algunas técnicas básicas de estadística descriptiva orientadas al análisis de datos. Estas técnicas consistirán en una serie de medidas, gráficos y modelos descriptivos que nos permitirán resumir y explorar un conjunto de datos, con el objetivo final de entenderlos lo mejor posible.

Los datos de los que disponemos suelen ser multidimensionales, en el sentido de que observamos varias características de una serie de individuos. Estos datos se tienen que registrar de alguna manera. Normalmente, los guardaremos en un archivo de ordenador con un formato preestablecido. Los formatos de almacenamiento de datos en un ordenador son diversos: texto simple (codificado en diferentes formatos: ASCII, isolatin, utf8 ...), hojas de cálculo (archivos de *Open Office* o *Excel*), bases de datos, etc. Una de las maneras básicas de almacenar datos es en forma de tablas de datos (en `R`, *data frames*).

Como hemos visto en la Lección 6, en una tabla de datos cada columna expresa una variable, mientras que cada fila corresponde a las observaciones de estas variables para un individuo concreto. Los datos de una misma columna tienen que ser del mismo tipo, porque corresponden a observaciones de una misma propiedad. En cambio, las filas en principio son de naturaleza heterogénea, porque pueden contener datos de diferentes tipos: especie del individuo, sexo, peso, edad, alguna medida, etc. Los tipos de datos que consideramos en este curso son los siguientes:

- Datos de tipo *atributo*, o *cualitativos*. Son los que expresan una cualidad del individuo, tales como el sexo, el DNI, la especie ... En `R`, guardaremos las listas de datos cualitativos en vectores (habitualmente, de palabras), o en factores si vamos a usarlos para clasificar individuos.

- Datos *ordinales*. Son datos similares a los cualitativos, con la única diferencia de que se pueden ordenar de manera natural. Por ejemplo, los niveles de calidad ambiental de un ecosistema (malo, regular, normal, bueno, muy bueno) o las calificaciones en un examen (suspenso, aprobado, notable, sobresaliente) son datos ordinales. En cambio, no se pueden ordenar de manera significativa los sexos o las especies de los individuos. En `R`, guardaremos las listas de datos ordinales en factores ordenados.

- Datos *cuantitativos*. Son datos que se refieren a medidas, tales como edades, longitudes, pesos, tiempos, números de individuos, etc. En `R`, guardaremos las listas de datos cuantitativos en vectores de números.

El análisis, tanto descriptivo como inferencial, de un conjunto de datos es diferente según su tipo. Así, para datos cualitativos sólo tiene interés estudiar y representar las frecuencias con que aparecen sus diferentes valores, mientras que el análisis de datos cuantitativos suele involucrar el cálculo de medidas estadísticas que evaluen numéricamente sus propiedades.

A modo de ejemplo, recordad el *data frame* `iris` que lleva predefinido `R` y con el que ya trabajamos en la Lección 6.

```
> head(iris,5)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
> str(iris)
'data.frame': 150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1
    1 1 1 1 1 1 ...
```

Las cuatro primeras variables son cuantitativas: amplitudes y longitudes de pétalos y sépalos; en cambio, la última es cualitativa: la especie.

Lección 8

Descripción de datos cualitativos

Los datos cualitativos corresponden a observaciones sobre cualidades de un objeto o individuo tales como su color o su sexo. Suelen codificarse por medio de palabras, pero también se pueden usar números que jueguen el papel de etiquetas: a modo de ejemplo, es habitual representar «No» (o «Fracaso», «Ausente»...) con un 0, y «Sí» (o «Éxito», «Presente»...) con un 1.

En general, consideramos datos cualitativos aquellos que pueden ser iguales o diferentes, pero que no admiten ningún otro tipo de comparación significativa: por ejemplo, que no tenga ningún sentido preguntarse si uno es más grande que otro, ni efectuar operaciones aritméticas con ellos, aunque estén representados por números. Por lo tanto, un mismo conjunto de datos puede ser cualitativo o de otro tipo según el análisis que vayamos a hacer de él. Por poner un ejemplo, supongamos que hemos anotado durante unos años los días de la semana en los que ha llovido más de 10 mm en nuestra ciudad. Si sólo nos interesa contar cuántas veces ha ocurrido en lunes, cuántas en martes, etc., esta lista de nombres serán datos cualitativos. Si, en cambio, queremos estudiar cómo se comportan los días de lluvia según avanza la semana, y por lo tanto el orden de los días es relevante, serán datos ordinales.

Denominaremos *variable cualitativa* a una lista de observaciones de un tipo de datos cualitativos sobre un conjunto concreto de objetos. Llamaremos *niveles*, como en los factores, a los diferentes valores que pueden tomar estos datos. Por ejemplo, los dos niveles de una variable *Sexo* serían «Macho» y «Hembra», o sinónimos.

Con `R`, usaremos vectores y factores para representar variables cualitativas. Los factores nos servirán para agrupar las observaciones según los niveles de la variable. De esta manera podremos segmentar la población que representa la variable en grupos o subpoblaciones, asignando un grupo a cada nivel, y podremos comparar el comportamiento de otras variables sobre estos grupos.

8.1. Frecuencias

Los estadísticos básicos para datos cualitativos son sencillos: dada una variable cualitativa, para cada uno de sus niveles podemos contar cuántos datos hay en ese nivel (la *frecuencia absoluta* del nivel) y qué fracción del total representan (la *frecuencia relativa* del nivel) y nada más.

Ejemplo 8.1. Supongamos que se ha realizado un seguimiento a 20 personas ingresadas en un geriátrico. Uno de los datos que se han recogido sobre estas personas ha sido su sexo. El resultado ha sido una variable cualitativa formada por las 20 observaciones siguientes:

> Mujer, Mujer, Hombre, Mujer, Mujer, Mujer, Mujer, Mujer, Hombre, Mujer, Hombre, Hombre, Mujer, Mujer, Hombre, Mujer, Mujer, Mujer, Mujer, Hombre.

Sus dos niveles son «Hombre» y «Mujer». En esta variable hay 14 mujeres y 6 hombres. Por lo tanto, éstas son las frecuencias absolutas de estos niveles. Puesto que en total hay 20 individuos, sus frecuencias relativas son

$$\text{Hombre: } \frac{6}{20} = 0.3, \qquad \text{Mujer: } \frac{14}{20} = 0.7.$$

En general, supongamos que tenemos un tipo de datos cualitativos con niveles

$$l_1, l_2, \ldots, l_k.$$

Efectuamos n observaciones de este tipo de datos, y sean

$$x_1, x_2, \ldots, x_n$$

los resultados que obtenemos. Cada una de estas observaciones x_j toma como valor alguno de los niveles l_i. Estas observaciones forman una *variable cualitativa*.

Así, en el ejemplo anterior tendríamos que $l_1 =$ Hombre y $l_2 =$ Mujer, que $n = 20$ (el número de observaciones efectuadas), y $x_1, \ldots, x_{20}$ formarían la muestra de sexos.

Con estas notaciones:

- La *frecuencia absoluta* del nivel l_j en esta variable cualitativa, que denotaremos por n_j, es el número de observaciones en las que x_i toma el valor l_j.
- La *frecuencia relativa* del nivel l_j en esta variable cualitativa es la fracción $f_j = n_j/n$. Es decir, la frecuencia relativa del nivel l_j es la fracción (en tanto por uno) de observaciones que corresponden a este nivel. El tanto por ciento de observaciones del nivel l_j es entonces $f_j \cdot 100\,\%$.
- La *moda* de esta variable cualitativa es su nivel, o niveles, de mayor frecuencia (absoluta o relativa, tanto da).

La Tabla 8.1 resume las frecuencias absolutas y relativas de la variable cualitativa del Ejemplo 8.1, con las notaciones que acabamos de introducir. Su moda es el nivel *Mujer*.

Sexo	n_j	f_j	%
Hombre	6	0.3	30 %
Mujer	14	0.7	70 %
Total	20	1	100 %

Tabla 8.1. Frecuencias de la variable del Ejemplo 8.1.

8.2. Tablas unidimensionales de frecuencias

Supongamos que tenemos una variable cualitativa guardada en un vector o un factor,[1] como por ejemplo:

```
> x=c(3,2,5,1,3,1,5,6,2,2,2,1,3,5,2)
> x
[1] 3 2 5 1 3 1 5 6 2 2 2 1 3 5 2
> Respuestas=factor(c("No","No","Sí","No","Sí","No","No","Sí"))
> Respuestas
[1] No No Sí No Sí No No Sí
Levels: No Sí
```

[1] Para simplificar, en lo que queda de sección, diremos *vector* para referirnos genéricamente tanto a un vector como a un factor.

Con R, la *tabla de frecuencias absolutas* de un vector que representa una variable cualitativa se calcula con la función `table`.

```
> table(x)
x
1 2 3 5 6
3 5 3 3 1
> table(Respuestas)
Respuestas
No Sí
 5  3
```

El resultado de una función `table` es un objeto de datos de un tipo nuevo: una *tabla de contingencia*, una `table` en el argot de R. Como vemos, al aplicar `table` a un vector obtenemos una tabla unidimensional formada por una fila con los niveles de la variable y una segunda fila donde, debajo de cada nivel, aparece su frecuencia absoluta en el vector.

Los nombres de las columnas de una tabla unidimensional se obtienen con la función `names`.

```
> names(table(x))
[1] "1" "2" "3" "5" "6"
> names(table(Respuestas))
[1] "No" "Sí"
```

Habréis observado que en la `table` de un vector sólo aparecen los nombres de los niveles presentes en el vector. Si el tipo de datos cualitativos usado tenía más niveles y queremos que aparezcan explícitamente en la tabla (con frecuencia 0), hay que transformar el vector en un factor con los niveles deseados.

```
> z=factor(x, levels=1:7)  #Los niveles serán 1,2,3,4,5,6,7
> z
[1] 3 2 5 1 3 1 5 6 2 2 2 1 3 5 2
Levels: 1 2 3 4 5 6 7
> table(z)
z
1 2 3 4 5 6 7
3 5 3 0 3 1 0
```

A efectos prácticos, podemos pensar que una tabla unidimensional es como un vector de números donde cada entrada está identificada por un nombre: el de su columna. Para referirnos a una entrada de una tabla unidimensional, podemos usar tanto su posición como su nombre (entre comillas, aunque sea un número).

```
> table(x)[4] #La cuarta columna de table(x)
5
3
> table(x)["4"] #¿La columna de table(x) con nombre 4?

NA
> table(x)["5"] #La columna de table(x) con nombre 5
5
3
```

```
> 3*table(x)[2] #El triple de la segunda columna de table(x)
2
15
```

Las tablas de contingencia aceptan la mayoría de las funciones explicadas para vectores.

```
> sum(table(x))  #Suma de las entradas de table(x)
[1] 15
> sqrt(table(Respuestas))  #Raíces cuadradas de las entradas de
   table(Respuestas)
Respuestas
      No       Sí
2.236068 1.732051
```

La *tabla de frecuencias relativas* de un vector se puede calcular aplicando la función `prop.table` a su `table`. El resultado vuelve a ser una tabla de contingencia unidimensional.

```
> prop.table(table(x))
x
         1          2          3          5          6
0.20000000 0.33333333 0.20000000 0.20000000 0.06666667
> prop.table(table(Respuestas))
Respuestas
   No    Sí
0.625 0.375
```

¡Atención! La función `prop.table` se tiene que aplicar al resultado de `table`, no al vector original. Si aplicamos `prop.table` a un vector de palabras o a un factor, dará un error, pero si la aplicamos a un vector de números, nos dará una tabla. Esta tabla *no es la tabla de frecuencias relativas* de la variable cualitativa representada por el vector, sino la tabla de frecuencias relativas de una variable que tuviera como tabla de frecuencias absolutas este vector de números, entendiendo que cada entrada del vector representa la frecuencia de un nivel diferente.

```
> prop.table(x)
 [1] 0.06976744 0.04651163 0.11627907 0.02325581 0.06976744
 [6] 0.02325581 0.11627907 0.13953488 0.04651163 0.04651163
[11] 0.04651163 0.02325581 0.06976744 0.11627907 0.04651163
> X=c(1,1,1)
> prop.table(table(X))
X
1
1
> prop.table(X)
[1] 0.3333333 0.3333333 0.3333333
```

También podemos calcular la tabla de frecuencias relativas de un vector dividiendo el resultado de `table` por el número de observaciones.

```
> table(x)/length(x)
x
```

```
          1           2           3           5           6
0.20000000 0.33333333 0.20000000 0.20000000 0.06666667
```

Dados un vector x y un número natural n, la instrucción

`names(which(table(`x`)==`n`))`

nos da los niveles que tienen frecuencia absoluta n en x.

```
> table(x)
x
1 2 3 5 6
3 5 3 3 1
> names(which(table(x)==3))
[1] "1" "3" "5"
> names(which(table(x)==4))
character(0)
```

En particular, por lo tanto,

`names(which(table(`x`)==max(table(`x`))))`

nos da los niveles de frecuencia máxima en x: su *moda*.

```
> names(which(table(x)==max(table(x))))
[1] "2"
> names(which(table(Respuestas)==max(table(Respuestas))))
[1] "No"
```

Ejemplo 8.2. Continuamos en la situación del Ejemplo 8.1. Para calcular las frecuencias y la moda con `R`, haríamos lo siguiente:

```
> Sexo_Ger=c("Mujer","Mujer","Hombre","Mujer","Mujer","Mujer",
  "Mujer","Mujer","Hombre","Mujer","Hombre","Hombre","Mujer",
  "Mujer","Hombre","Mujer","Mujer","Mujer","Mujer","Hombre")
> t0=table(Sexo_Ger)
> t0
Sexo_Ger
Hombre  Mujer
     6     14
> prop.table(t0)
Sexo_Ger
Hombre  Mujer
   0.3    0.7
> names(which(t0==max(t0)))
[1] "Mujer"
```

8.3. Tablas bidimensionales de frecuencias

La función `table` también permite construir tablas de frecuencias conjuntas de dos o más variables. A modo de ejemplo, supongamos que el vector `Respuestas`, de la sección anterior, contiene

las respuestas a una pregunta dadas por unos individuos cuyos sexos tenemos almacenados en un vector `Sexo`, en el mismo orden que sus respuestas. En este caso, podemos construir una tabla que nos diga cuántos individuos de cada sexo han dado cada respuesta.

```
> Respuestas=c("No","No","Sí","No","Sí","No","No","Sí")
> Sexo=c("M","M","M","H","H","H","H","H") #H es hombre, M es mujer
> table(Respuestas,Sexo)
          Sexo
Respuestas H M
        No 3 2
        Sí 2 1
> table(Sexo,Respuestas)
    Respuestas
Sexo No Sí
   H  3  2
   M  2  1
```

El resultado es, en ambos casos, una tabla de contingencia como antes, pero ahora *bidimensional*, puesto que cada entrada tiene dos dimensiones, una por cada variable, como en una matriz.

Como podemos ver, en una tabla bidimensional producida aplicando `table` a dos vectores, los niveles del primer vector en el argumento definen las filas, y los del segundo, las columnas. Así, en `table(Respuestas,Sexo)`, las filas corresponden a las respuestas y las columnas a los sexos. Para intercambiar filas por columnas, es decir, para «trasponer» la tabla sin tener que recalcularla, podemos usar la misma función `t` que usamos para trasponer matrices:

```
> t(table(Respuestas,Sexo))
    Respuestas
Sexo No Sí
   H  3  2
   M  2  1
```

En la práctica, tenéis que decidir si alguna de las asignaciones de variables a dimensiones es más conveniente que la otra. Por ejemplo, y teniendo en cuenta que nuestra manera natural de leer una tabla es por filas, si lo que nos interesa son las frecuencias de las respuestas entre las personas de cada sexo, seguramente lo más adecuado será elegir el sexo como variable para las filas.

Para referirnos a una entrada de una tabla bidimensional podemos usar el sufijo `[ , ]` como si estuviéramos en una matriz o un *data frame*. Dentro de los corchetes, tanto podemos usar los índices como los nombres (entre comillas) de los niveles.

```
> table(Respuestas,Sexo)[1,2]
[1] 2
> table(Respuestas,Sexo)["No","M"]
[1] 2
> table(Sexo,Respuestas)[1,2]
[1] 2
> table(Sexo,Respuestas)["H","Sí"]
[1] 2
```

Como en el caso unidimensional, la función `prop.table` sirve para calcular tablas bidimensionales de frecuencias relativas conjuntas de pares de variables. Pero en el caso bidimensional tenemos dos tipos de frecuencias relativas, que definen, para cada par de variables, tres tablas diferentes:

- Las frecuencias relativas *globales*: para cada par de niveles, uno de cada variable, la fracción de individuos que pertenecen a ambos niveles respecto del total de la muestra; por ejemplo, la fracción de mujeres que han contestado que sí respecto del total de la muestra sería la frecuencia relativa global del par (mujer, sí).
- Las frecuencias relativas *marginales*: dentro de cada nivel de una variable, y para cada nivel de la otra, la fracción de individuos que pertenecen al segundo nivel respecto del total de la subpoblación definida por el primer nivel.

  Dadas dos variables, se pueden calcular dos familias de frecuencias relativas marginales, según cuál sea la variable que defina las subpoblaciones en las que calculemos las frecuencias relativas de los niveles de la otra variable; no es lo mismo la fracción de mujeres que han contestado que sí respecto del total de mujeres, que la fracción de mujeres que han contestado que sí respecto del total de personas que han dado esta misma respuesta.

La tabla de frecuencias relativas globales se calcula aplicando sin más la función `prop.table` a la `table`. Por lo que se refiere a las frecuencias relativas marginales, la variable que define las subpoblaciones en las que las calculamos se indica con el parámetro `margin`. Así, con `margin=1` especificamos que la variable que define las subpoblaciones es la primera, y que, por lo tanto, las frecuencias relativas se calculan dentro de las filas. En cambio, con `margin=2` especificamos que la variable que define las subpoblaciones es la segunda, por lo que las frecuencias relativas se calculan por columnas.

```
> prop.table(table(Sexo,Respuestas)) #Global
    Respuestas
Sexo     No     Sí
   H  0.375  0.250
   M  0.250  0.125
> prop.table(table(Sexo,Respuestas), margin=1) #Por sexo
    Respuestas
Sexo         No         Sí
   H  0.6000000  0.4000000
   M  0.6666667  0.3333333
> prop.table(table(Sexo,Respuestas), margin=2) #Por respuesta
    Respuestas
Sexo         No         Sí
   H  0.6000000  0.6666667
   M  0.4000000  0.3333333
```

De esta manera:

- La tabla `prop.table(table(Sexo,Respuestas))` nos da la fracción del total que representa cada pareja (sexo, respuesta): por ejemplo, un 25 % del total de la muestra son mujeres que han contestado que no.

- La tabla `prop.table(table(Sexo,Respuestas), margin=1)` nos da la fracción que representa cada respuesta dentro de cada sexo: por ejemplo, un 66.66 % de las mujeres han contestado que no.
- La tabla `prop.table(table(Sexo,Respuestas), margin=2)` nos da la fracción que representa cada sexo dentro de cada respuesta: por ejemplo, las mujeres representan el 40 % del total de las personas que han contestado que no.

La función `CrossTable` del paquete `gmodels` permite producir (especificando el parámetro `prop.chisq=FALSE`) un resumen de la tabla de frecuencias absolutas y las tres tablas de frecuencias relativas de dos variables en un formato adecuado para su visualización:

```
> #Instalamos y cargamos el paquete gmodels
...
> CrossTable(Sexo,Respuestas,prop.chisq=FALSE)

   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  8


             | Respuestas
        Sexo |        No |        Sí | Row Total |
-------------|-----------|-----------|-----------|
           H |         3 |         2 |         5 |
             |     0.600 |     0.400 |     0.625 |
             |     0.600 |     0.667 |           |
             |     0.375 |     0.250 |           |
-------------|-----------|-----------|-----------|
           M |         2 |         1 |         3 |
             |     0.667 |     0.333 |     0.375 |
             |     0.400 |     0.333 |           |
             |     0.250 |     0.125 |           |
-------------|-----------|-----------|-----------|
Column Total |         5 |         3 |         8 |
             |     0.625 |     0.375 |           |
-------------|-----------|-----------|-----------|
```

La leyenda `Cell Contents` explica los contenidos de cada celda de la tabla: en este caso, y en orden descendente, la frecuencia absoluta, la frecuencia relativa por filas, la frecuencia relativa por columnas, y la frecuencia relativa global. Esta función dispone de muchos parámetros que permiten modificar el contenido de las celdas, los podéis consultar en `help(CrossTable)`.

Una tabla de contingencia bidimensional es, básicamente, una matriz con algunos atributos extra. En particular, podemos usar sobre estas tablas la mayoría de las funciones para matrices

que tengan sentido para tablas; por ejemplo, `rowSums` y `colSums` se pueden aplicar a una tabla y suman sus filas y sus columnas, respectivamente:

```
> table(Sexo,Respuestas)
    Respuestas
Sexo No Sí
   H  3  2
   M  2  1
> colSums(table(Sexo,Respuestas))
 No Sí
  5  3
> rowSums(table(Sexo,Respuestas))
H M
5 3
> colSums(prop.table(table(Sexo,Respuestas)))
   No    Sí
0.625 0.375
> rowSums(prop.table(table(Sexo,Respuestas)))
    H     M
0.625 0.375
```

También podemos usar sobre una tabla bidimensional (o, en general, multidimensional) la función `apply` con la misma sintaxis que para matrices.

8.4. Tablas multidimensionales de frecuencias (opcional)

En general, podemos calcular tablas de frecuencias de cualquier número de variables, no sólo de una o dos. El manejo de estas tablas multidimensionales es similar al caso bidimensional, simplemente recordando que ahora hay más variables que tener en cuenta a la hora, por ejemplo, de especificar entradas o de calcular frecuencias relativas marginales.

Veamos un ejemplo tridimensional. Supongamos que, además de los vectores

```
> Respuestas=c("No","No","Sí","No","Sí","No","No","Sí")
> Sexo=c("M","M","M","H","H","H","H","H")
```

tenemos un tercer vector con las nacionalidades de los individuos representados en estos dos vectores:

```
> País=c("Francia","Alemania","España","España","España","España",
  "Alemania","Francia")
```

Podemos calcular entonces una tabla de frecuencias absolutas para las ternas (sexo, respuesta, país).

```
> table(Sexo,Respuestas,País)
, , País = Alemania

    Respuestas
Sexo No Sí
   H  1  0
   M  1  0
```

```
, , País = España

    Respuestas
Sexo No Sí
   H  2  1
   M  0  1

, , País = Francia

    Respuestas
Sexo No Sí
   H  0  1
   M  1  0
```

R muestra la *tabla tridimensional* que obtenemos como una lista de tablas bidimensionales `table(Sexo,Respuestas)`, separando la población según el nivel de la tercera variable. Si no os gusta esta manera de visualizar una tabla tridimensional, una alternativa es usar la función `ftable`, que la mostrará en lo que se llama *formato plano*:

```
> ftable(Sexo,Respuestas,País)
                País Alemania España Francia
Sexo Respuestas
H    No                    1      2       0
     Sí                    0      1       1
M    No                    1      0       1
     Sí                    0      1       0
```

Los parámetros `row.vars` y `col.vars` de `ftable` permiten especificar qué variables queremos que aparezcan como filas o como columnas.

```
> ftable(Sexo,Respuestas,País, col.vars=c("Sexo","Respuesta"))
         Sexo         H     M
         Respuestas  No Sí No Sí
País
Alemania             1  0  1  0
España               2  1  0  1
Francia              0  1  1  0
```

Para referirnos a una entrada, o a una subtabla, de una tabla podemos usar los corchetes.

```
> table(Sexo,Respuestas,País)["H","Sí","España"]
[1] 1
> table(Sexo,Respuestas,País)[ , ,"España"]
     Respuestas
Sexo No Sí
   H  2  1
   M  0  1
> table(Sexo,Respuestas,País)[  ,"Sí","España"]
H M
1 1
> table(Sexo,Respuestas,País)["M", ,"España"]
```

```
 No Sí
  0  1
```

En una tabla multidimensional, podemos calcular frecuencias relativas marginales respecto de los niveles de una variable o respecto de combinaciones de niveles de varias variables: por ejemplo, las frecuencias relativas marginales de las respuestas en cada combinación (sexo, país). Como en el caso bidimensional, las tablas correspondientes se calculan aplicando `prop.table` a la tabla de frecuencias absolutas, y especificando con el parámetro `margin` las dimensiones, o combinaciones de dimensiones, respecto de las que calculamos las frecuencias relativas. Si no se especifica el parámetro `margin`, se obtiene la tabla de frecuencias relativas globales.

```
> prop.table(table(Sexo,Respuestas,País)) #Frecuencias relativas
   globales
, , País = Alemania

    Respuestas
Sexo    No    Sí
   H 0.125 0.000
   M 0.125 0.000

, , País = España

    Respuestas
Sexo    No    Sí
   H 0.250 0.125
   M 0.000 0.125

, , País = Francia

    Respuestas
Sexo    No    Sí
   H 0.000 0.125
   M 0.125 0.000

> prop.table(table(Sexo,Respuestas,País), margin=3)   #Frecuencias
   relativas por país
, , País = Alemania

    Respuestas
Sexo   No   Sí
   H 0.50 0.00
   M 0.50 0.00

, , País = España

    Respuestas
Sexo   No   Sí
   H 0.50 0.25
   M 0.00 0.25

, , País = Francia
```

```
    Respuestas
Sexo   No   Sí
   H 0.00 0.50
   M 0.50 0.00

> prop.table(table(Sexo,Respuestas,País), margin=c(1,3)) #
   Frecuencias relativas por sexo y país
, , País = Alemania

    Respuestas
Sexo        No        Sí
   H 1.0000000 0.0000000
   M 1.0000000 0.0000000

, , País = España

    Respuestas
Sexo        No        Sí
   H 0.6666667 0.3333333
   M 0.0000000 1.0000000

, , País = Francia

    Respuestas
Sexo        No        Sí
   H 0.0000000 1.0000000
   M 1.0000000 0.0000000
```

En este caso:

- La tabla `prop.table(table(Sexo,Respuestas,País))` nos da la fracción que representa cada pareja (sexo, respuesta, país) dentro del total de la muestra: por ejemplo, los hombres españoles que han contestado afirmativamente forman un 12.5 % del total de individuos.
- La tabla `prop.table(table(Sexo,Respuestas,País), margin=3)` nos da la fracción que representa cada pareja (sexo, respuesta) dentro de cada país: por ejemplo, los hombres que han contestado que sí representan un 25 % del total de individuos españoles.
- La tabla `prop.table(table(Sexo,Respuestas,País), margin=c(1,3))` nos da la fracción que representa cada respuesta dentro de cada combinación de (sexo, país): por ejemplo, un 33.33 % del total de hombres españoles ha contestado que sí.

La función `ftable` se puede aplicar a una tabla, y nos la muestra en formato plano. En particular, puede usarse para visualizar en este formato una tabla de frecuencias relativas multidimensional.

```
> ftable(prop.table(table(Sexo,Respuestas,País)))
                País Alemania España Francia
Sexo Respuestas
H    No                 0.125  0.250   0.000
```

```
     Sí                  0.000   0.125   0.125
M    No                  0.125   0.000   0.125
     Sí                  0.000   0.125   0.000
```

Hasta ahora hemos manipulado tablas de frecuencias que hemos construido nosotros mismos a partir de variables cualitativas. Todo lo que hemos hecho con estas tablas se puede también hacer con las tablas de frecuencias que lleva `R` predefinidas o que obtengamos de otra manera. Por ejemplo, el objeto de datos `HairEyeColor` que lleva predefinido `R` es una tabla de frecuencias de tres variables cualitativas: color de cabello (`Hair`), color de los ojos (`Eye`) y sexo (`Sex`).

```
> HairEyeColor
, , Sex = Male

       Eye
Hair    Brown Blue Hazel Green
  Black    32   11    10     3
  Brown    53   50    25    15
  Red      10   10     7     7
  Blond     3   30     5     8

, , Sex = Female

       Eye
Hair    Brown Blue Hazel Green
  Black    36    9     5     2
  Brown    66   34    29    14
  Red      16    7     7     7
  Blond     4   64     5     8
```

Efectuemos algunas operaciones sobre esta tabla, para ilustrar como podemos trabajar con ella:

```
> sum(HairEyeColor)   #Número total de individuos en la muestra
[1] 592
> HairEyeColor[ , ,"Male"]  #Subtabla de hombres
       Eye
Hair    Brown Blue Hazel Green
  Black    32   11    10     3
  Brown    53   50    25    15
  Red      10   10     7     7
  Blond     3   30     5     8

> prop.table(HairEyeColor, margin=3)   #Frecuencias relativas de
   las combinaciones (color de cabello, color de ojos) en cada sexo
, , Sex = Male

       Eye
Hair           Brown        Blue       Hazel       Green
  Black 0.114695341 0.039426523 0.035842294 0.010752688
  Brown 0.189964158 0.179211470 0.089605735 0.053763441
  Red   0.035842294 0.035842294 0.025089606 0.025089606
```

```
  Blond 0.010752688 0.107526882 0.017921147 0.028673835

, , Sex = Female

       Eye
Hair          Brown        Blue       Hazel       Green
  Black 0.115015974 0.028753994 0.015974441 0.006389776
  Brown 0.210862620 0.108626198 0.092651757 0.044728435
  Red   0.051118211 0.022364217 0.022364217 0.022364217
  Blond 0.012779553 0.204472843 0.015974441 0.025559105

> prop.table(HairEyeColor, margin=c(1,2)) #Frecuencias relativas de
    los sexos en cada combinación (color de cabello, color de ojos)
, , Sex = Male

       Eye
Hair         Brown      Blue     Hazel     Green
  Black 0.4705882 0.5500000 0.6666667 0.6000000
  Brown 0.4453782 0.5952381 0.4629630 0.5172414
  Red   0.3846154 0.5882353 0.5000000 0.5000000
  Blond 0.4285714 0.3191489 0.5000000 0.5000000
, , Sex = Female

       Eye
Hair         Brown      Blue     Hazel     Green
  Black 0.5294118 0.4500000 0.3333333 0.4000000
  Brown 0.5546218 0.4047619 0.5370370 0.4827586
  Red   0.6153846 0.4117647 0.5000000 0.5000000
  Blond 0.5714286 0.6808511 0.5000000 0.5000000
```

Para cambiar el orden de las variables en una *tabla* multidimensional, se puede usar la instrucción

`aperm(`*tabla*`, perm=...)`,

igualando el parámetro `perm` a la lista de las variables en el orden deseado. Por ejemplo, si queremos una tabla equivalente a `HairEyeColor`, pero con primera variable `Sex`, segunda variable `Hair` y tercera variable `Eye`, podemos hacer:

```
> aperm(HairEyeColor, perm=c("Sex", "Hair", "Eye"))
, , Eye = Brown

        Hair
Sex      Black Brown Red Blond
  Male      32    53  10     3
  Female    36    66  16     4

, , Eye = Blue

        Hair
Sex      Black Brown Red Blond
  Male      11    50  10    30
```

```
  Female     9    34    7    64

, , Eye = Hazel

        Hair
Sex      Black Brown Red Blond
  Male      10    25   7     5
  Female     5    29   7     5

, , Eye = Green

        Hair
Sex      Black Brown Red Blond
  Male       3    15   7     8
  Female     2    14   7     8
```

8.5. Tablas a partir de *data frames* de variables cualitativas

Como ya hemos comentado en varias ocasiones, la manera natural de organizar datos multidimensionales en `R` es en forma de *data frame*. En esta sección explicaremos algunas instrucciones para calcular tablas de frecuencias absolutas a partir de un *data frame* de variables cualitativas. Para ilustrarla, usaremos el fichero que se encuentra en el *url*

`http://bioinfo.uib.es/~recerca/RMOOC/bebenerg.txt`

Este fichero consiste en una tabla de datos con la siguiente información sobre 122 estudiantes de la Escuela Politécnica Superior de la UIB: su sexo (variable `sexo`), el grado en el que están matriculados (variable `estudio`) y si consumen habitualmente bebidas energéticas (variable `bebe`).

```
> Beb_Energ=read.table("http://bioinfo.uib.es/~recerca/RMOOC/
   bebenerg.txt",header=TRUE)
> str(Beb_Energ)
'data.frame': 122 obs. of  3 variables:
 $ estudio: Factor w/ 4 levels "Informática",..: 1 3 2 1 2 3 1 2 1
   1 ...
 $ bebe   : Factor w/ 2 levels "No","Sí": 1 1 2 2 1 1 2 1 1 1 ...
 $ sexo   : Factor w/ 2 levels "Hombre","Mujer": 2 1 2 1 2 2 1 1 1
   1 ...
> head(Beb_Energ)
         estudio bebe   sexo
1    Informática   No  Mujer
2    Matemáticas   No Hombre
3 Ing.Industrial   Sí  Mujer
4    Informática   Sí Hombre
5 Ing.Industrial   No  Mujer
6    Matemáticas   No  Mujer
```

Aplicando la función `summary` a un *data frame* de variables cualitativas, obtenemos, a modo de resumen, una tabla con las frecuencias absolutas de cada variable.

```
> summary(Beb_Energ)
           estudio    bebe         sexo
 Informática    :53   No:97   Hombre:83
 Ing.Industrial:37   Si:25   Mujer :39
 Matemáticas    :16
 Telemática     :16
```

Esta tabla sólo sirve para ver la información, porque sus entradas son palabras.

```
> summary(Beb_Energ)[,2]

"No:97   " "Sí:25   "         NA         NA
```

Para calcular en un solo paso la `table` de cada variable, podemos usar la función `apply` de la manera siguiente:

```
> apply(Beb_Energ, MARGIN=2, FUN=table)
$estudio

   Informática Ing.Industrial     Matemáticas      Telemática
            53              37              16              16

$bebe

No Sí
97 25

$sexo

Hombre  Mujer
    83     39
```

De esta manera, obtenemos una `list` cuyas componentes son las tablas que queríamos.

```
> apply(Beb_Energ,MARGIN=2,FUN=table)$sexo

Hombre  Mujer
    83     39
> table(Beb_Energ$sexo)

Hombre  Mujer
    83     39
```

Si aplicamos la función `table` a un *data frame* de variables cualitativas, obtenemos su tabla de frecuencias absolutas, con las variables ordenadas tal y como aparecen en el *data frame*.

```
> table(Beb_Energ)
, , sexo = Hombre

                bebe
estudio          No Sí
  Informática    30  7
```

```
  Ing.Industrial 19  6
  Matemáticas     8  1
  Telemática     10  2

, , sexo = Mujer

                bebe
estudio          No Sí
  Informática    11  5
  Ing.Industrial 10  2
  Matemáticas     6  1
  Telemática      3  1

> table(Beb_Energ[c(1,3)])
                sexo
estudio          Hombre Mujer
  Informática        37    16
  Ing.Industrial     25    12
  Matemáticas         9     7
  Telemática         12     4
```

Otra opción es usar la función `ftable`, que produce la misma tabla de frecuencias pero en formato plano.

```
> ftable(Beb_Energ)
                    sexo Hombre Mujer
estudio         bebe
Informática     No            30    11
                Sí             7     5
Ing.Industrial  No            19    10
                Sí             6     2
Matemáticas     No             8     6
                Sí             1     1
Telemática      No            10     3
                Sí             2     1
```

8.6. Diagramas de barras

El tipo de gráfico más usado para representar variables cualitativas son los diagramas de barras (*bar plots*). Como su nombre indica, un *diagrama de barras* contiene, para cada nivel de la variable cualitativa, una barra de altura su frecuencia; por ejemplo, la Figura 8.1 es un diagrama de barras de las frecuencias absolutas de los dos niveles de la muestra de sexos del Ejemplo 8.1. El código que lo ha producido, y que explicaremos en esta sección, es el siguiente:

```
> barplot(table(Sexo_Ger), col=c("lightblue","pink"),
  main="Diagrama de barras de las frecuencias absolutas\n de la
   variable \"Sexo_Ger\"")
```

La función `\n` dentro de una frase entrada entre comillas introduce un cambio de línea. El mismo efecto se obtiene con un cambio de línea dentro del valor de `main`. Id con cuidado,

porque ambos efectos se acumulan, así que si cambiáis de línea después del `\n`, obtendréis una línea en blanco. Por otro lado, `\"` escribe unas comillas en el texto entrado entre comillas.

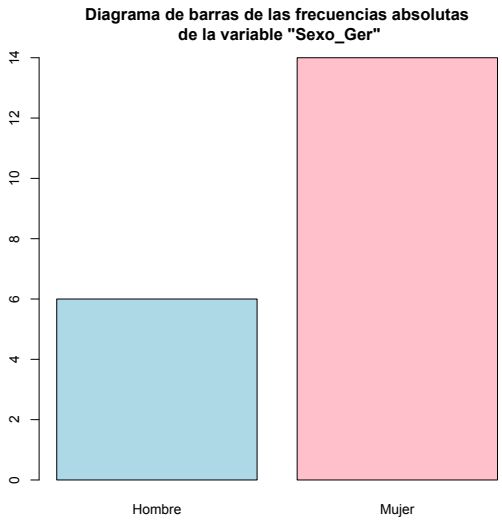

Figura 8.1. Diagrama de barras de las frecuencias absolutas de los datos del Ejemplo 8.1.

La manera más sencilla de dibujar un diagrama de barras de las frecuencias absolutas o relativas de una variable cualitativa es usando la instrucción `barplot` aplicada a la tabla correspondiente. Así,

```
> x=c(3,2,5,1,3,1,5,6,2,2,2,1,3,5,2)
> Respuestas=c("No","No","Sí","No","Sí","No","No","Sí")
> barplot(table(x), main="Diagrama de barras de frecuencias
   absolutas de la variable \"x\"")
> barplot(prop.table(table(Respuestas)), main="Diagrama de barras
   de frecuencias relativas\n de la variable \"Respuestas\"")
```

produce los diagramas de la Figura 8.2.

¡Atención! Como pasaba con `prop.table`, el argumento de `barplot` ha de ser una tabla, y, por consiguiente, se ha de aplicar al resultado de `table` o de `prop.table`, nunca al vector de datos original.

Habréis observado que en las funciones `barplot` anteriores hemos usado el parámetro `main` para poner título a los diagramas; en general, la función `barplot` admite los parámetros de `plot` que tienen sentido en el contexto de los diagramas de barras: `xlab`, `ylab`, `main`, etc. Los parámetros disponibles se pueden consultar en `help(barplot)`. Aquí sólo vamos a comentar algunos.

Se pueden especificar los colores de las barras usando el parámetro `col`. Si se iguala a un solo color, todas las barras serán de este color, pero también se puede especificar un color para cada barra, igualando `col` a un vector de colores. Por ejemplo,

```
> barplot(table(Respuestas), col=c("green"))
> barplot(table(Respuestas), col=c("red","blue"))
```

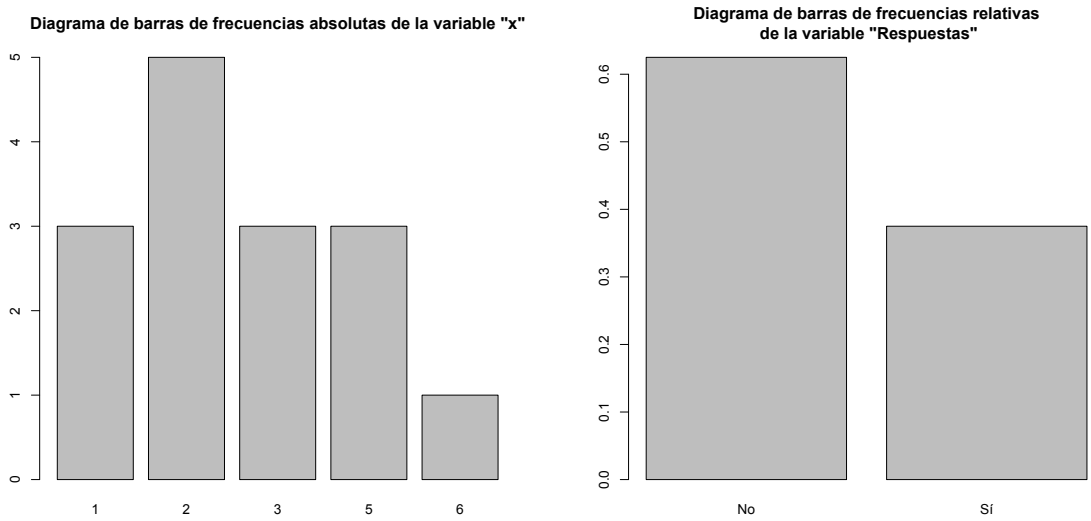

Figura 8.2. Diagramas de barras.

producen, respectivamente, los dos diagramas de la Figura 8.3. En un diagrama con muchas barras, es conveniente usar un esquema adecuado de colores para ellas. Para ello se puede usar el paquete `RColorBrewer`, del que hablaremos en detalle en la Sección 12.3.

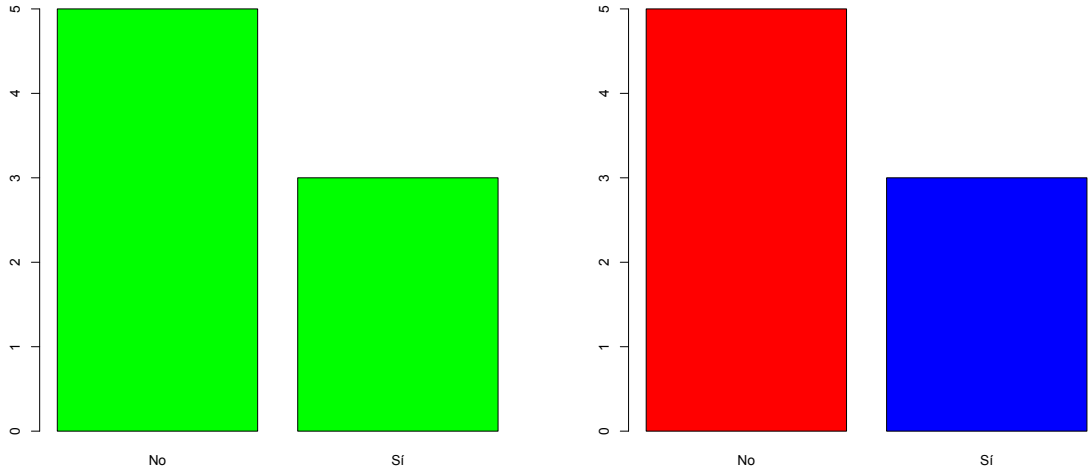

Figura 8.3. Ejemplos de diagramas de barras de colores.

Una opción interesante es dibujar las barras horizontales en vez de verticales: para hacerlo, se tiene que añadir el parámetro `horiz=TRUE`. Así, la Figura 8.4 se obtiene con la siguiente instrucción:

```
> barplot(table(x), horiz=TRUE)
```

Si se aplica `barplot` a una tabla bidimensional, por defecto dibuja las barras de la segunda variable cortadas por la frecuencia de la primera variable: se le llama un *diagrama de barras apiladas*. Por ejemplo, las instrucciones siguientes producen la Figura 8.5.

```
> Respuestas=c("No","No","Sí","No","Sí","No","No","Sí")
```

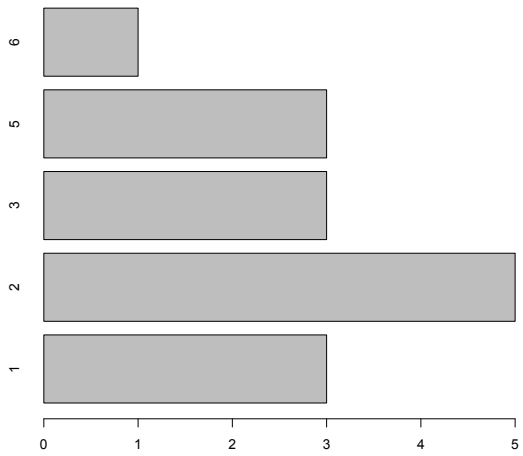

Figura 8.4. Un diagrama de barras horizontales.

```
> Sexo=c("M","M","M","H","H","H","H","H")
> table(Sexo,Respuestas)
    Respuestas
Sexo No Sí
   H  3  2
   M  2  1
> barplot(table(Sexo,Respuestas))
```

En un diagrama de barras apiladas, las barras globales corresponden a los niveles de la variable que definen las columnas de la tabla, es decir, la segunda variable especificada dentro de `table`: en el de la Figura 8.5, se trata de la variable `Respuestas`, de niveles `No` y `Sí`. Cada una de estas barras se divide verticalmente en sectores que representan los niveles de la otra variable, en orden ascendente: en el ejemplo que nos ocupa, la zona inferior de cada barra representa el nivel `H` de la variable `Sexo` y la zona superior su nivel `M`.

En vez de organizar las barras de la primera variable en una sola barra vertical, se pueden dibujar una junto a la otra añadiendo el parámetro `beside=TRUE`, obteniéndose de esta manera un *diagrama de barras por bloques*. Así,

```
> barplot(table(Sexo,Respuestas), beside=TRUE)
```

produce el diagrama de barras de la izquierda de la Figura 8.6. En este diagrama, cada bloque de barras representa un nivel de la variable de las columnas (`No` y `Sí`), y en cada uno de estos bloques las barras representan los niveles de las filas en su orden (en cada bloque, la barra de la izquierda corresponde a `H` y la de la derecha a `M`).

Los diagramas de barras tienen que mostrar la información de la manera más adecuada. Por ejemplo, en un diagrama de barras por bloques, si lo que nos interesa es la distribución de las respuestas por sexo, los bloques de barras deberían corresponder a los sexos, y las barras dentro de cada bloque a las respuestas. En este caso, convendría cambiar el orden de los vectores dentro del `table` al que aplicamos `barplot`, o trasponer la tabla antes de aplicarle `barplot`.

Suele ser conveniente añadir a un diagrama de barras de dos variables una leyenda que indique cada sector (en los diagramas de barras apiladas) o cada barra (en los diagramas de barras

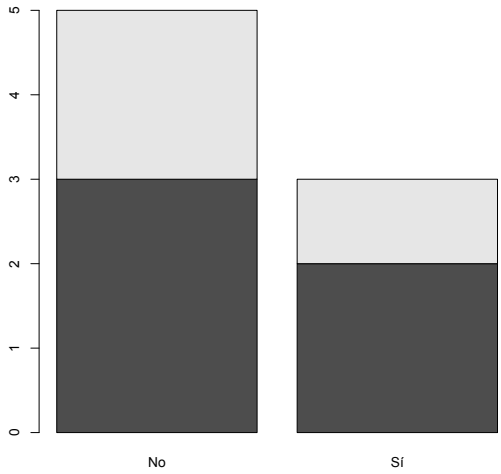

Figura 8.5. Diagrama de barras apiladas de una tabla bidimensional.

por bloques) qué nivel representa. Esto se puede realizar entrando el parámetro `legend.text` igualado a `TRUE`, si no queremos modificar el nombre de los niveles de las filas, o igualado a un vector con los nombres que les queremos asignar (en el orden que toque). Por ejemplo,

```
> barplot(table(Sexo,Respuestas), beside=TRUE, legend.text=TRUE)
```

produce el diagrama de la derecha de la Figura 8.6.

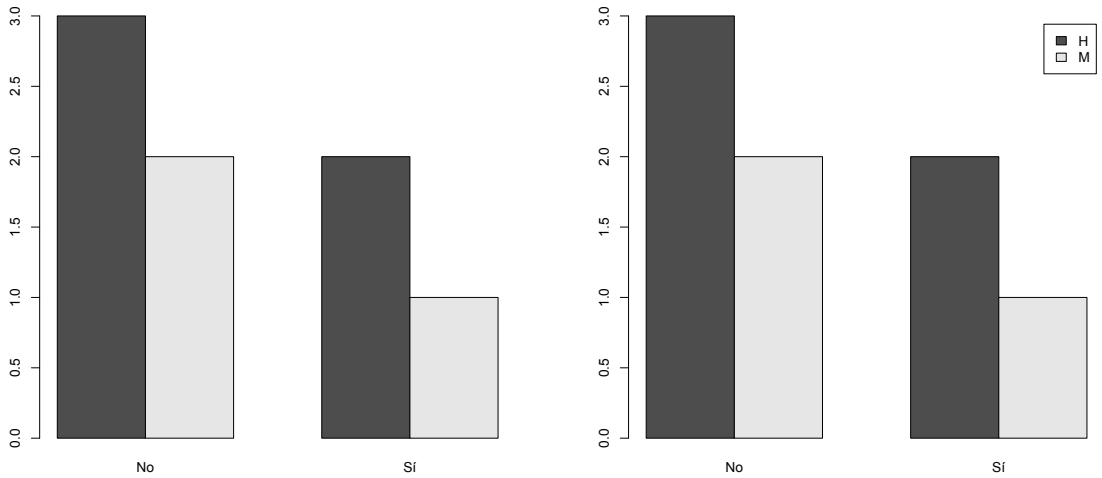

Figura 8.6. Diagramas de barras por bloques con `beside=TRUE`.

La leyenda que genera `R` se puede modificar usando el parámetro `args.legend` igualado a una `list` con los parámetros que usaríamos en la función `legend` que explicamos en la Lección 5: `x`, para indicar la posición de la leyenda, `cex` para indicar el factor por el cual se quiere multiplicar su tamaño, etc. Podéis consultar el resto de parámetros en `help(legend)`.

Se pueden cambiar los colores de las barras usando el parámetro `col` como en los diagramas de barras de tablas unidimensionales. La función `legend.text` importa estos colores, no hace falta especificarlos en el `args.legend`.

También puede ser conveniente poner nombres más informativos a los niveles de las variables. El parámetro `names` dentro de `barplot` permite cambiar los nombres de los niveles que muestra debajo del eje horizontal: en un diagrama de barras de una variable, los de sus niveles, y en un diagrama bidimensional, los de los niveles de la variable de las columnas.

Veamos un ejemplo usando `col` y con los nombres que se muestran de los niveles de ambas variables traducidos al inglés; el resultado es la Figura 8.7:

```
> barplot(table(Respuestas,Sexo), beside=TRUE, names=c("Men",
  "Women"), col=c("red","blue"), legend.text=c("No","Yes"))
```

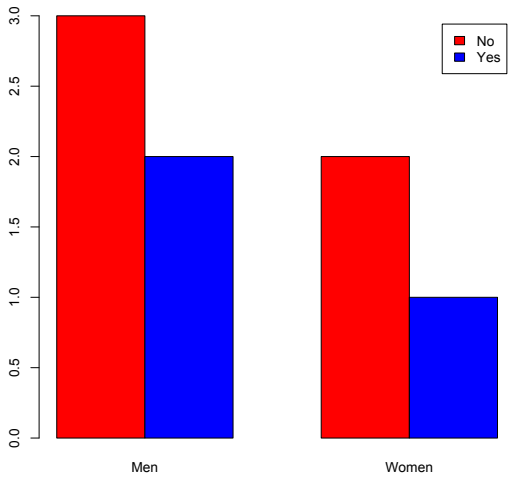

Figura 8.7. Diagrama de barras para visualizar la distribución de las respuestas por sexo.

8.7. Otros gráficos básicos para datos cualitativos

Un tipo muy popular de representación gráfica de variables cualitativas son los diagramas circulares. En un *diagrama circular* (*pie chart*) se representan los niveles de una variable cualitativa como sectores circulares de un círculo, de manera que el ángulo (o equivalentemente, el área) de cada sector sea proporcional a la frecuencia del nivel al que corresponde. Con `R`, este tipo de diagramas se producen con la instrucción `pie`, de nuevo aplicada a una tabla de frecuencias y no al vector original. La función `pie` admite muchos parámetros para modificar el resultado: se pueden cambiar los colores con `col`, se pueden cambiar los nombres de los niveles con `names`, se puede poner un título con `main`, etc.; podéis consultar la lista completa de parámetros en `help(pie)`. Así, por ejemplo,

```
> x=c(3,2,5,1,3,1,5,6,2,2,2,1,3,5,2)
> Respuestas=c("No","No","Sí","No","Sí","No","No","Sí")
> pie(table(x), main="Diagrama circular de la variable x")
```

```
> pie(table(Respuestas), main="Diagrama circular de la variable
   Respuestas")
```

produce los diagramas de la Figura 8.8.

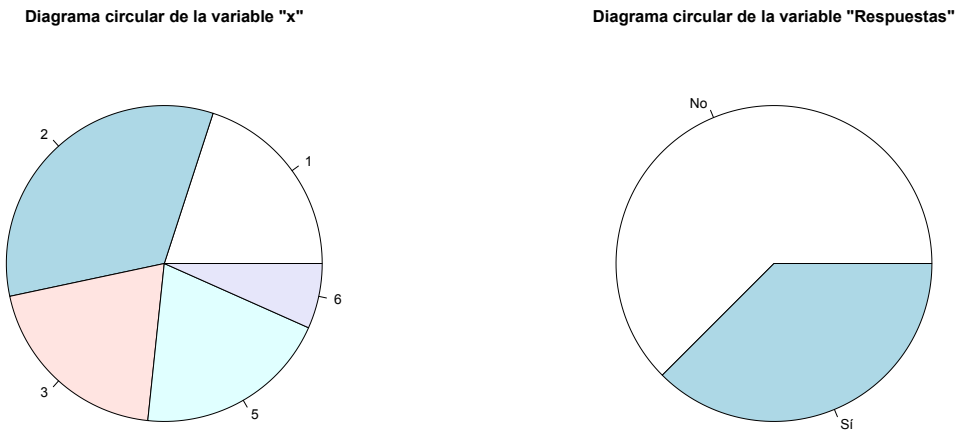

Figura 8.8. Diagramas circulares.

Pese a su popularidad, es poco recomendable usar diagramas circulares porque a veces es difícil, a simple vista, comprender las relaciones entre las frecuencias que representan. Para convencerse, basta comparar los diagramas de barras y los diagramas circulares de la Figura 8.9, extraída de la entrada sobre diagramas circulares de la *Wikipedia*.[2]

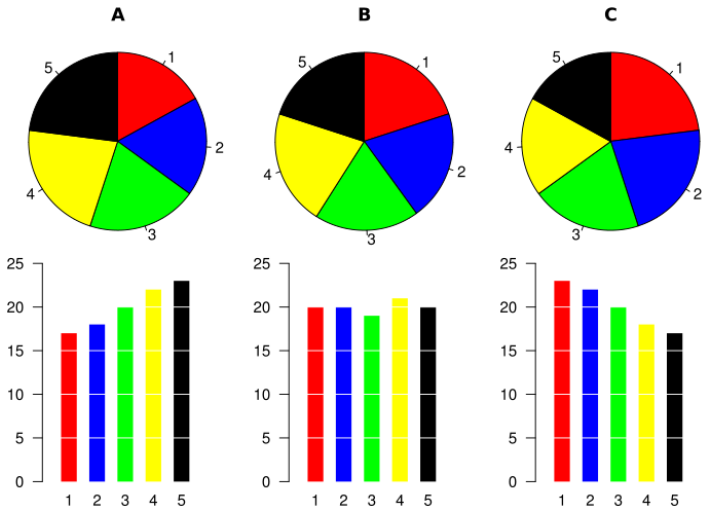

Figura 8.9. Diagramas de barras *vs* diagramas circulares.

Otra representación de las tablas multidimensionales de frecuencias son los *gráficos de mosaico*. Estos gráficos se obtienen substituyendo cada entrada de la tabla de frecuencias por una región rectangular de área proporcional a su valor. En concreto, para obtener el gráfico de mosaico de una tabla bidimensional, se parte de un cuadrado de lado 1, primero se divide en barras

[2] `http://en.wikipedia.org/wiki/Pie_chart`.

verticales de amplitudes iguales a las frecuencias relativas de una variable, y luego cada barra se divide, a lo alto, en regiones de alturas proporcionales a las frecuencias relativas marginales de cada nivel de la otra variable, dentro del nivel correspondiente de la primera variable.

Un gráfico de mosaico de una tabla se obtiene con `R` aplicando la función `plot` a la tabla, o también la función `mosaicplot`. Esta última también se puede aplicar a matrices. Por ejemplo,

```
> Respuestas=c("No","No","Sí","No","Sí","No","No","Sí")
> Sexo=c("M","M","M","H","H","H","H","H")
> plot(table(Sexo,Respuestas), main="Gráfico de mosaico de las
   variables \"Sexo\" y \"Respuestas\"")
```

produce el gráfico de mosaico de la izquierda en la Figura 8.10.

En el gráfico de mosaico de una tabla tridimensional, primero se divide el cuadrado en barras verticales de amplitudes iguales a las frecuencias relativas de una variable, luego cada barra se divide, a lo alto, en regiones de alturas proporcionales a las frecuencias relativas marginales de cada nivel de una segunda variable, dentro del nivel correspondiente de la primera variable, y finalmente cada sector rectangular se vuelve a dividir a lo ancho en regiones de amplitudes proporcionales a las frecuencias relativas marginales de cada nivel de la tercera variable dentro de la combinación correspondiente de niveles de las otras dos. Por ejemplo,

```
> plot(HairEyeColor,  main="Gráfico de mosaico de la tabla
   HairEyeColor", col=c("pink","lightblue"))
```

produce el gráfico de la derecha en la Figura 8.10.

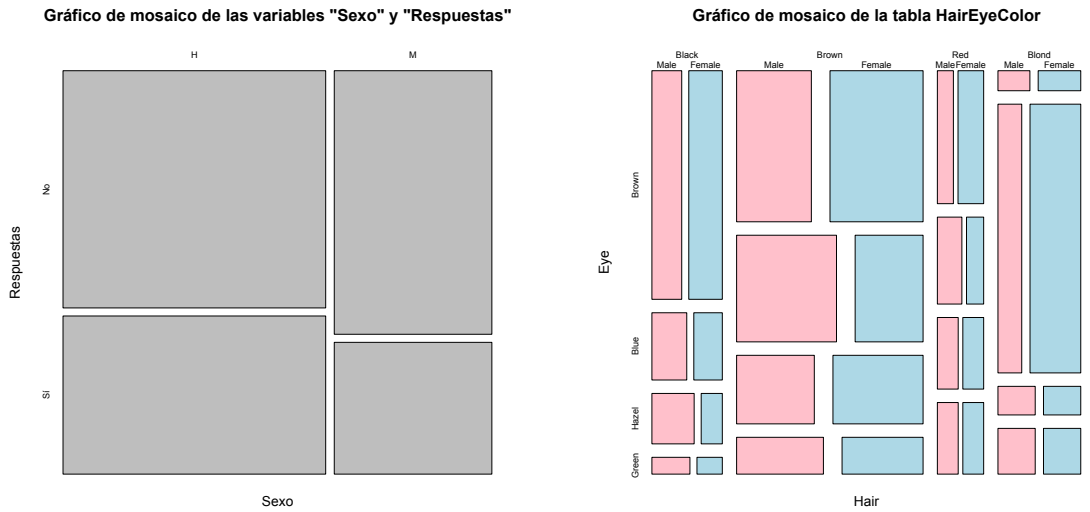

Figura 8.10. Diagramas de mosaico bidimensional y tridimensional.

Además de sus parámetros usuales, la función `plot` admite algunos parámetros específicos cuando se usa para producir el gráfico de mosaico de una tabla. Estos parámetros se pueden consultar en `help(mosaicplot)`.

Los paquetes `vcd` y `vcdExtra` incluyen otras funciones que producen representaciones gráficas interesantes de tablas tridimensionales. Por ejemplo:

- La función `cotabplot` de `vcd` produce un diagrama de mosaico para cada nivel de la tercera variable.
- La función `mosaic3d` de `vcdExtra` produce un diagrama de mosaico tridimensional en una ventana de una aplicación para gráficos 3D interactivos.

Por ejemplo, con

```
> #Instalamos y cargamos el paquete vcd
...
> cotabplot(table(Sexo,Respuestas,País))
> #Instalamos y cargamos el paquete vcdExtra
...
> mosaic3d(HairEyeColor, type="expected", box=TRUE,
  col=c("pink","lightblue"))
```

obtenemos los dos gráficos de la Figura 8.11 .

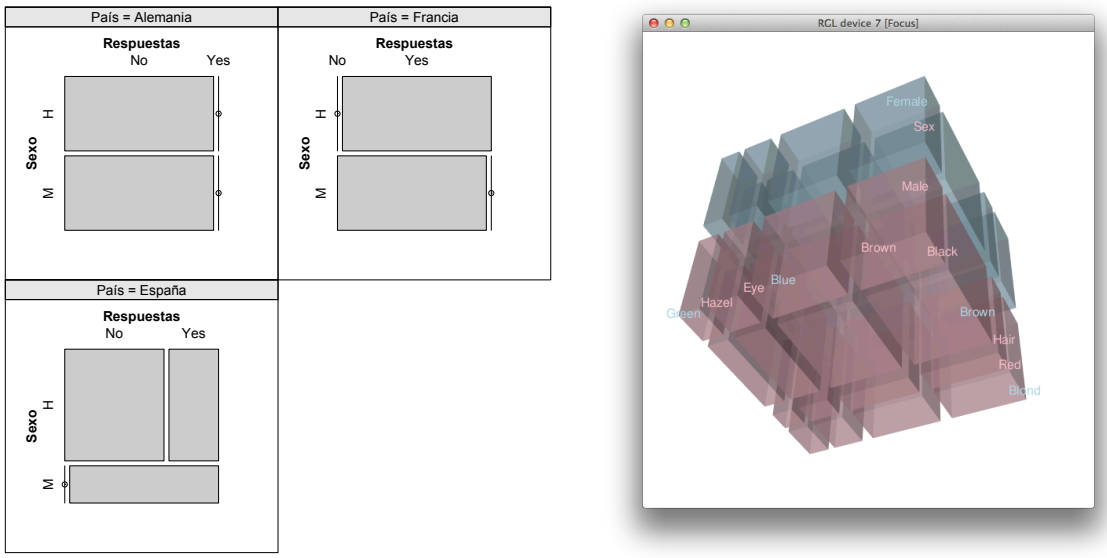

Figura 8.11. Ejemplos de gráficos obtenido con `cotabplot` (izquierda) y `mosaic3d` (derecha).

8.8. Un ejemplo completo

Vamos a llevar a cabo un análisis completo de un ejemplo con lo que hemos aprendido en esta lección, y aprovecharemos para aprender algo nuevo.

Como ya hemos comentado, el objeto de datos `HairEyeColor` que lleva predefinido `R` es una tabla de frecuencias absolutas de tres variables cualitativas: color de cabello (`Hair`), color de los ojos (`Eye`) y sexo (`Sex`). Vamos a extraer de esta tabla una tabla bidimensional de frecuencias absolutas de las variables `Eye` y `Hair`, sin distinguir según el sexo. La manera más sencilla de obtener esta tabla es sumando las subtablas de frecuencias para hombres y mujeres, y aplicando `as.table` al resultado para transformarlo en una `table` por si no lo es.[3]

[3] En realidad, `HairEyeColor[,,1]` y `HairEyeColor[,,2]` son matrices, y, por lo tanto, su suma también lo es. Pa-

```
> HEC=as.table(HairEyeColor[ , , 1]+ HairEyeColor[ , , 2])
> HEC
       Eye
Hair    Brown Blue Hazel Green
  Black    68   20    15     5
  Brown   119   84    54    29
  Red      26   17    14    14
  Blond     7   94    10    16
```

Vamos a traducir al castellano los nombres de las variables de esta tabla y de sus niveles. Esto lo podemos llevar a cabo en un solo paso con la función `dimnames` que ya usamos sobre *data frames*. El resultado de aplicar esta función a una `table` es una `list` cuyas componentes son los niveles de cada variable.

```
> dimnames(HEC)
$Hair
[1] "Black" "Brown" "Red"   "Blond"

$Eye
[1] "Brown" "Blue"  "Hazel" "Green"
```

Por lo tanto, para reescribir los nombres de las variables y sus niveles, basta redefinir esta `list` de la manera siguiente:

```
> dimnames(HEC)=list(Cabello=c("Negro","Castaño","Rojo","Rubio"),
   Ojos=c("Marrones","Azules","Pardos","Verdes"))
> HEC
         Ojos
Cabello   Marrones Azules Pardos Verdes
  Negro          68     20     15      5
  Castaño       119     84     54     29
  Rojo           26     17     14     14
  Rubio           7     94     10     16
```

Vamos a dibujar un diagrama de mosaico de esta tabla, para visualizar gráficamente sus entradas.

```
> plot(HEC,col=c("lightblue"),main="Diagrama de mosaico de la tabla
    bidimensional de frecuencias\n de colores de cabellos y ojos")
```

Obtenemos la Figura 8.12. A simple vista, vemos que las combinaciones de colores de cabellos y ojos más frecuentes son los cabellos castaños con los ojos marrones, y los cabellos rubios con los ojos azules.

---

ra conocer de qué clase es un objeto, le podemos aplicar la función `class`. Por ejemplo, `class(HairEyeColor)` da `"table"`, pero si entramos `class(HairEyeColor[,,1]+ HairEyeColor[,,2])`, da `"matrix"`. Por lo tanto, sin el `as.table`, la tabla que hubiéramos construido sería una realidad una matriz.

A este nivel de manejo de `R`, que esta tabla no sea una `table` no es grave. La única diferencia sería que, como podéis comprobar por vuestra cuenta, si le aplicásemos la función `plot` no obtendríamos un diagrama de mosaico, y tendríamos que usar `mosaicplot`. Pero no cuesta nada convertirla en una `table` con la función `as.table`, y así no tenemos que preocuparnos de este tipo de cuestiones.

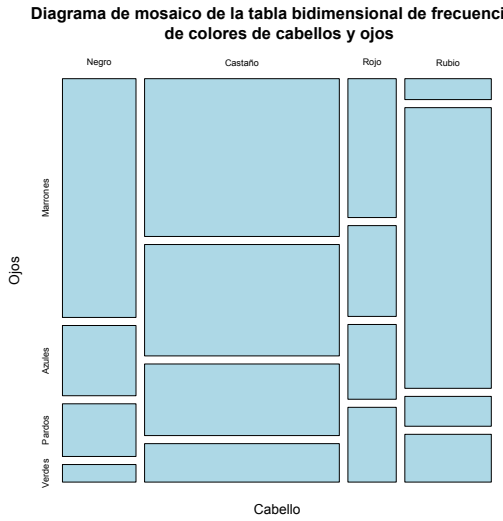

Figura 8.12. Diagrama de mosaico de las frecuencias conjuntas de colores de ojos y de cabellos en `HairEyeColor`.

A continuación, vamos a calcular el número total de individuos representados en esta tabla, así como las tablas de frecuencias absolutas y relativas de cada variable, y representaremos estas últimas en sendos diagramas de barras.

```
> sum(HEC)  #Número total de individuos
[1] 592
> colSums(HEC)    #Frec. abs. de Ojos
Marrones   Azules   Pardos   Verdes
     220       215       93       64
> rowSums(HEC)       #Frec. abs. de Cabello
  Negro Castaño    Rojo   Rubio
    108     286      71     127
> round(prop.table(colSums(HEC)),3)   #Frec. rel. de Ojos
Marrones   Azules   Pardos   Verdes
   0.372    0.363    0.157    0.108
> round(prop.table(rowSums(HEC)),3)   #Frec. rel. de Cabello
  Negro Castaño    Rojo   Rubio
  0.182   0.483   0.120   0.215
> barplot(prop.table(colSums(HEC)), ylim=c(0,0.4),
  col=c("burlywood4","lightblue","orange3","lightgreen"),
  main="Frecuencias relativas de colores de ojos")
> barplot(prop.table(rowSums(HEC)),
  col=c("black","brown","red","gold"), ylim=c(0,0.5),
  main="Frecuencias relativas de colores de cabello")
```

Los diagramas de barras producidos con la dos últimas instrucciones son los de la Figura 8.13. Vemos que el color dominante de cabellos es el castaño, mientras que en el color de ojos el marrón y el azul están prácticamente empatados.

Pasamos ahora a calcular las tablas de frecuencias relativas y dibujar los dos diagramas de barras de las frecuencias relativas marginales.

```
> round(prop.table(HEC), 3)  #Frec. rel. globales
```

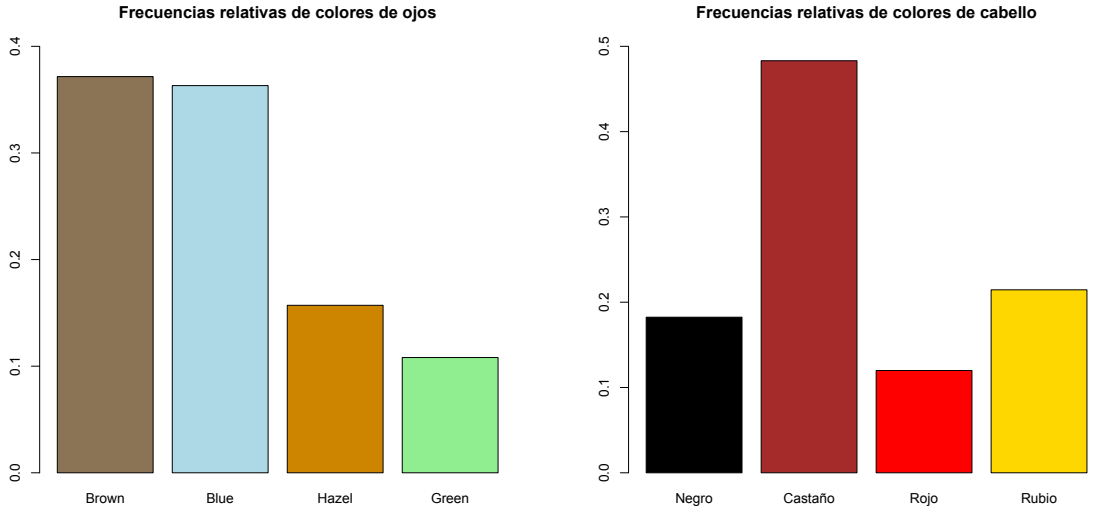

Figura 8.13. Diagrama de barras de frecuencias relativas de colores de ojos y de cabellos en `HairEyeColor`.

```
         Ojos
Cabello   Marrones Azules Pardos Verdes
  Negro      0.115  0.034  0.025  0.008
  Castaño    0.201  0.142  0.091  0.049
  Rojo       0.044  0.029  0.024  0.024
  Rubio      0.012  0.159  0.017  0.027
> round(prop.table(HEC, margin=1), 3)  #Frec. rel. por filas
         Ojos
Cabello   Marrones Azules Pardos Verdes
  Negro      0.630  0.185  0.139  0.046
  Castaño    0.416  0.294  0.189  0.101
  Rojo       0.366  0.239  0.197  0.197
  Rubio      0.055  0.740  0.079  0.126
> round(prop.table(HEC, margin=2), 3)  #Frec. rel. por columnas
         Ojos
Cabello   Marrones Azules Pardos Verdes
  Negro      0.309  0.093  0.161  0.078
  Castaño    0.541  0.391  0.581  0.453
  Rojo       0.118  0.079  0.151  0.219
  Rubio      0.032  0.437  0.108  0.250
> barplot(prop.table(HEC, margin=1), beside=TRUE, legend.text=TRUE,
   col=c("black","brown","red","gold"), ylim=c(0,0.8),
   main="Frecuencias relativas de colores de cabello\n en cada
   color de ojos")
> barplot(t(prop.table(HEC, margin=2)), beside=TRUE,
   legend.text=TRUE, ylim=c(0,0.6),
   col=c("burlywood4","lightblue","orange3","lightgreen"),
   main="Frecuencias relativas de colores de ojo\n
   en cada color de cabellos")
```

Los diagramas de barras producidos con la dos últimas instrucciones son los de la Figura 8.14. Vemos, por ejemplo, que entre las personas de ojos azules, los cabellos rubios son los más

frecuentes, y que entre las personas castañas el color de ojos más frecuente es el pardo.

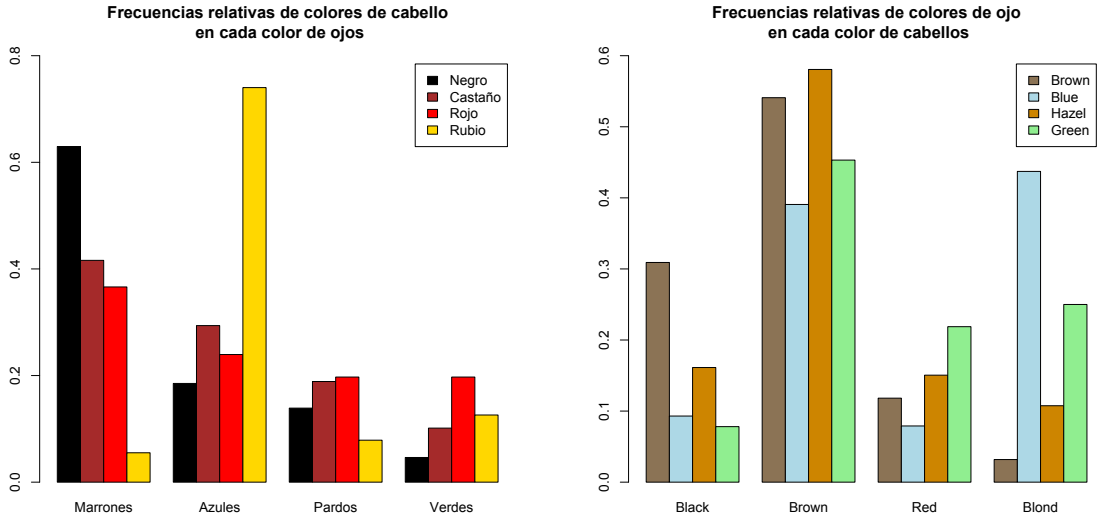

Figura 8.14. Diagrama de barras de frecuencias relativas marginales de colores de ojos y de cabellos en `HairEyeColor`.

8.9. Guía rápida

- `table` calcula la tabla de frecuencias absolutas de un vector o un factor.
- `as.table` convierte un objeto (por ejemplo, una matriz) en una tabla de contingencia.
- `ftable` muestra una tabla multidimensional en formato plano.
- `t` sirve para trasponer una tabla bidimensional.
- `aperm` sirve para permutar las variables de una tabla multidimensional.
- `prop.table` calcula la tabla de frecuencias relativas de un vector o un factor a partir de su tabla de frecuencias absolutas. Tiene el parámetro siguiente:
  - `margin`: sirve para especificar las dimensiones en cuyos niveles se calcularán las frecuencias relativas marginales. Si no se especifica, se calculan las frecuencias relativas globales.
- `CrossTable`, del paquete `gmodels`, produce, en el caso bidimensional, una tabla conjunta de frecuencias absolutas y de frecuencias relativas globales y marginales.
- `names` da los nombres de las columnas de una tabla unidimensional, y sirve también para modificar estos nombres.
- `dimnames` da una `list` con los vectores de los nombres de los niveles de las diferentes variables de una tabla multidimensional, y sirve también para modificar los nombres tanto de las variables como de sus niveles.
- *tabla*`[...]` se usa para especificar un elemento, una fila, una columna o una subtabla de la *tabla*.

- `barplot` dibuja el diagrama de barras de un vector o un factor a partir de una tabla de frecuencias. Algunos parámetros importantes:
  - `col`: sirve para especificar los colores de las barras.
  - `horiz=TRUE`: sirve para dibujar el diagrama horizontal.
  - `beside=TRUE`: sirve para especificar que el diagrama sea por bloques.
  - `legend.text`: sirve para añadir una leyenda que asigne barras a los niveles de la primera variable.
  - `args.legend`: sirve para modificar las características de esta leyenda, igualándolo a una `list` con los valores de los parámetros de la función `legend` que queramos especificar.
  - `names`: sirve para cambiar en el diagrama los nombres de los niveles de la segunda variable.
  - `main`, `xlab`, `ylab` y el resto de parámetros de `plot` que tengan sentido para diagramas de barras.
- `pie` dibuja el diagrama circular de un vector o un factor a partir de una tabla de frecuencias. Algunos parámetros importantes:
  - `col`: sirve para especificar los colores de los sectores.
  - `names`: sirve para cambiar en el diagrama los nombres de los niveles.
  - `main`, `xlab`, `ylab` y el resto de parámetros de `plot` que tengan sentido para diagramas circulares.
- `plot` y `mosaicplot` dibujan el diagrama de mosaico de una tabla de frecuencias.
- `cotabplot`, del paquete `vcd`, produce una tabla con un diagrama de mosaico para cada nivel de la última variable.
- `mosaic3d`, del paquete `vcdExtra`, produce un diagrama de mosaico tridimensional.
- `\n` indica un cambio de línea en un título o etiqueta.
- `\"` escribe unas comillas en el texto de un título o etiqueta.
- `class` sirve para conocer de qué clase es un objeto de `R`.

8.10. Ejercicio

Instalad y cargad el paquete `MASS`. Este paquete lleva una tabla de datos llamada `birthwt` sobre factores que pueden incidir en el peso de los niños al nacer. Con `str` y `head`, explorad la estructura, y con `help`, mirad el significado de cada variable.

(a) Calculad una tabla bidimensional de frecuencias relativas marginales de los pares (raza de la madre, peso inferior a 2.5 kg o no) que permita ver, fácilmente, si la raza de la madre influye en el peso del bebé. Dibujad un diagrama de mosaico de esta tabla.

Asimismo, dibujad un diagrama bidimensional de barras, con las barras organizadas en bloques, que permita visualizar esta información. Poned nombres adecuados a los bloques, colores a las barras, y añadid una leyenda que explique qué representa cada barra. ¿Se puede obtener alguna conclusión de esta tabla y de este diagrama de barras?

(b) Repetid el punto anterior para los pares (madre fumadora o no, peso inferior a 2.5 kg o no) y para los pares (madre hipertensa o no, peso inferior a 2.5 kg o no).

(c) Calculad una tabla de frecuencias relativas marginales de las ternas (raza de la madre, madre fumadora o no, peso inferior a 2.5 kg o no) que permita ver, fácilmente, si la raza de la madre y su condición de fumadora o no fumadora influyen en el peso del bebé. Dibujad un diagrama de mosaico de esta tabla.