Tema 6 Apuntes

Carla Romero Sansano

28 de octubre de 2019

Tablas de frecuencias unidimensionales

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

```
aleatorio = sample(1:5, size = 12, replace = TRUE)
aleatorio
```

```
##  [1] 4 4 3 4 3 1 5 4 3 2 5 4
```

```
Respuestas=factor(sample(c("Si","No"), size = 12, replace = TRUE))
Respuestas
```

```
##  [1] Si No Si No No No Si No Si No No Si
## Levels: No Si
```

```
table(aleatorio)
```

```
## aleatorio
## 1 2 3 4 5
## 1 1 3 5 2
```

```
table(Respuestas)
```

```
## Respuestas
## No Si
##  7  5
```

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.

Tabla de contingencia: Tabla unidimensional formada por una fila con los niveles de la variable y una segunda fila con la frecuencia absoluta de cada uno de los niveles que ha aparecido en la muestra (debajo de cada nivel aparece su frecuencia absoluta en el vector).

names():Devuelve el nombre de los niveles presentes en la tabla de contingencia

En la **table** de un vector solo 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(aleatorio, levels = 1:7) #Los niveles serán ahora 1,2,3,4,5,6,7
z
```

```
##  [1] 4 4 3 4 3 1 5 4 3 2 5 4
## Levels: 1 2 3 4 5 6 7
```

```
table(z)
```

```
## z
## 1 2 3 4 5 6 7
## 1 1 3 5 2 0 0
```

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(aleatorio)[3] #La tercera columna de table(aleatorio)
```

```
## 3
## 3
```

```
table(z)[3]
```

```
## 3
## 3
```

```
table(aleatorio)["7"] #¿La columna de table(aleatorio) con nombre 7?
```

```
## <NA>
##   NA
```

```
table(z)["7"]
```

```
## 7
## 0
```

```
table(aleatorio)["5"] #La columna de table(aleatorio) con nombre 5
```

```
## 5
## 2
```

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

```
## 2
## 3
```

Las tablas de contingencia aceptan la mayoría de las funciones que ya hemos utilizado para vectores:

```
sum(table(aleatorio)) #Suma de las entradas de table(aleatorio)
```

```
## [1] 12
```

```
sqrt(table(Respuestas)) #Raíces cuadradas de las entradas de table(Respuestas)
```

```
## Respuestas
##       No       Si
## 2.645751 2.236068
```

Tablas de contingencia (práctica)

```
datos = factor(c("H","M","M","M","H","H","M","M"))
table(datos)
```

```
## datos
## H M
## 3 5
```

```
table(datos)["M"]
```

```
## M
## 5
```

```
sum(table(datos))
```

```
## [1] 8
```

Tabla de frecuencias relativas

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

```
prop.table(table(aleatorio))
```

```
## aleatorio
##          1          2          3          4          5
## 0.08333333 0.08333333 0.25000000 0.41666667 0.16666667
```

```
prop.table(table(Respuestas))
```

```
## Respuestas
##        No        Si
## 0.5833333 0.4166667
```

¡CUIDADO! 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 absoluta de un nivel diferente.

```
prop.table(aleatorio)
```

```
##  [1] 0.09523810 0.09523810 0.07142857 0.09523810 0.07142857 0.02380952
##  [7] 0.11904762 0.09523810 0.07142857 0.04761905 0.11904762 0.09523810
```

Otro ejemplo:

```
X = c(1,1,1)
prop.table(table(X))
```

```
## X
## 1
## 1
```

```
prop.table(X)
```

```
## [1] 0.3333333 0.3333333 0.3333333
```

También se puede calcular la tabla de frecuencias relativas a mano, es decir, dividiendo el resultado de `table` por el número total de observaciones.

```
table(aleatorio)/length(aleatorio)
```

```
## aleatorio
##          1          2          3          4          5
## 0.08333333 0.08333333 0.25000000 0.41666667 0.16666667
```

Frecuencias absolutas

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(aleatorio)
```

```
## aleatorio
## 1 2 3 4 5
```

```
## 1 1 3 5 2
```

```
names(which(table(aleatorio) == 3))
```

```
## [1] "3"
```

```
names(which(table(aleatorio) == 5))
```

```
## [1] "4"
```

Moda

Tuneando la instrucción anterior tendríamos la siguiente: `names(which(table(x) == max(table(x))))` que nos da los niveles de frecuencia máxima en **x**: su **moda**.

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

```
## [1] "4"
```

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

```
## [1] "No"
```

Ejercicio práctico

Recuperad el ejemplo de los 6 hombres y las 14 mujeres anterior y, utilizando R, calculad su tabla de frecuencias absolutas, su tabla de frecuencias relativas y la moda. Pista: usad la función **rep()** para no tener que escribir los datos a mano.

```
H <- rep("H", 6)
Muj <- rep("M", 14)
HM = c(H,Muj)
HM
```

```
##  [1] "H" "H" "H" "H" "H" "H" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M"
## [18] "M" "M" "M"
```

```
HyM= factor(sample(HM, 20, replace = TRUE))
#HyM = factor(sample(c(H,Muj), length(Muj),replace(TRUE)))
HyM
```

```
##  [1] H M M M H M M M M M M M H M H M H M M H
## Levels: H M
```

```
table(HyM) #TABLA FRECUENCIAS ABSOLUTAS
```

```
## HyM
##  H  M
##  6 14
```

```
prop.table(table(HyM)) #TABLA FRECUENCIAS RELATIVAS
```

```
## HyM
##   H   M
## 0.3 0.7
```

```
names(which(table(HyM) == max(table(HyM)))) #MODA
```

```
## [1] "M"
```

Otro ejercicio hecho por el profesor

```
datos = factor(c("H","M","M","M","H","H","M","M"))
table(datos)
```

```
## datos
## H M
## 3 5
```

```
table(datos)["M"]
```

```
## M
## 5
```

```
sum(table(datos))
```

```
## [1] 8
```

Frecuencias relativas

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

```
prop.table(table(datos))
```

```
## datos
##     H     M
## 0.375 0.625
```

```
100*prop.table(table(datos)) #Porcentaje
```

```
## datos
##    H    M
## 37.5 62.5
```

```
table(datos)/length(datos) #Forma manual
```

```
## datos
##     H     M
## 0.375 0.625
```

```
names(table(datos))
```

```
## [1] "H" "M"
```

```
names(which(table(datos)==3)) #Los que tienen una frecuencia absoluta igual a 3
```

```
## [1] "H"
```

```
names(which(table(datos)==max(table(datos)))) #MODA
```

```
## [1] "M"
```

```
#Moda creando una función:
moda <- function(d){
  names(which(table(d)==max(table(d))))
}
m_t = moda(datos)
```

La moda del data frame **datos** es: M.

Tablas de frecuencias bidimensionales

La función **table()** también permite construir tablas de frecuencias conjuntas de dos o más variables. Supongamos que el vector **Respuestas** 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 cuántas personas de cada sexo han dado cada respuesta.

```
Respuestas=factor(sample(c("Si","No"), size = 12, replace = TRUE))
Respuestas
```

```
##  [1] Si Si No Si Si Si Si Si No No Si Si
## Levels: No Si
```

```
Sexo=sample(c("H","M"), size = length(Respuestas), replace = T) #H = hombre, M = mujer (Muestra aleator
table(Respuestas,Sexo) #Respuesta en las filas y Sexo en las columnas
```

```
##           Sexo
## Respuestas H M
##         No 2 1
##         Si 4 5
```

Mini ejercicio

- Comprobad qué ocurre si cambiamos el orden de las columnas en la función table()

```
table(Sexo, Respuestas)
```

```
##     Respuestas
## Sexo No Si
##    H  2  4
##    M  1  5
```

-Usad la función t() para transponer ambas tablas y comprobad el resultado

```
t(table(Respuestas, Sexo)) #Nos da la table(Sexo, Respuestas)
```

```
##     Respuestas
## Sexo No Si
##    H  2  4
##    M  1  5
```

```
t(table(Sexo, Respuestas)) #Nos da la table(Respuestas, Sexo)
```

```
##           Sexo
## Respuestas H M
##         No 2 1
##         Si 4 5
```

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] #Fila: 1, columna: 2
```

```
## [1] 1
```

```
table(Respuestas,Sexo)["No", "M"]
```

```
## [1] 1
```

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:

- **Frecuencias relativas globales**: para cada par de niveles, uno de cara variable, la fracción de individuos que pertenecen a ambos niveles respecto del total de la muestra. (*Ejemplo*: proporción de mujeres que han dicho que sí respecto al total de individuos (muestra) (hombre y mujer))
- **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. (*Ejemplo*:dos familias de frecuencias marginales, proporción de mujeres que han dicho que sí respecto al total de mujeres o proporción de mujeres que han dicho que sí respecto al total de personas que han dicho que sí)

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**.

```
prop.table(table(Sexo,Respuestas)) #GLOBAL
```

```
##     Respuestas
## Sexo         No         Si
##    H 0.16666667 0.33333333
##    M 0.08333333 0.41666667
```

-Ejemplo, un 33% del total de la muestra son Hombres que han dicho que No.

De este modo, la tabla **prop.table(table(Sexo, Respuestas))** nos da la fracción total que representa cada pareja (sexo,respuesta).

Para obtener las marginales, debemos usar el parámetro **margin** al aplicar la función **prop.table()** a la **table**. Con **margin=1** obtenemos las frecuencias relativas de las filas y con **margin=2**, de las columnas.

```
prop.table(table(Sexo,Respuestas), margin=1) #Por Sexo
```

```
##     Respuestas
## Sexo        No        Si
##    H 0.3333333 0.6666667
##    M 0.1666667 0.8333333
```

```
prop.table(table(Sexo,Respuestas), margin=2) #Por Respuesta
```

```
##     Respuestas
## Sexo        No        Si
##    H 0.6666667 0.4444444
##    M 0.3333333 0.5555556
```

Primer caso (por sexo): -El 66% de los hombres respondió que No y el 33% de los hombres respondió que Si. -El 16% de las mujeres respondió que No y el 83% de las mujeres respondió que Si.

Segundo caso (por respuesta): -De los que han respondido que No, el 80% eran hombres y el 20% mujeres -De los que han respondido que Si, el 28% eran hombres y el 71% eran mujeres

Función CrossTable()

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 frecuen-

cias relativas de dos variables en un formato adecuado para su visualización.

La leyenda *Cell Contents* explica los contenidos de cada celda de la tabla: 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, y que podéis consultar en **help(CrossTable)**.

Paquete gmodels

```
#install.packages("gmodels")
library(gmodels)
```

```
## Warning: package 'gmodels' was built under R version 3.5.3
```

```
sex = factor(c("H","M","M","M","H","H","M","M"))
ans = factor(c("S","N","S","S","S","N","N","S"))

CrossTable(sex, ans, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  8
##
##
##              | ans
##          sex |         N |         S | Row Total |
## -------------|-----------|-----------|-----------|
##            H |         1 |         2 |         3 |
##              |     0.333 |     0.667 |     0.375 |
##              |     0.333 |     0.400 |           |
##              |     0.125 |     0.250 |           |
## -------------|-----------|-----------|-----------|
##            M |         2 |         3 |         5 |
##              |     0.400 |     0.600 |     0.625 |
##              |     0.667 |     0.600 |           |
##              |     0.250 |     0.375 |           |
## -------------|-----------|-----------|-----------|
## Column Total |         3 |         5 |         8 |
##              |     0.375 |     0.625 |           |
## -------------|-----------|-----------|-----------|
##
##
```

Sumas por filas y columnas

```
tt <- table(sex,ans)
tt #Frecuencias absolutas
```

```
##    ans
## sex N S
##   H 1 2
##   M 2 3
```

```
prop.table(tt) #Frecuencia relativa Global
```

```
##    ans
## sex     N     S
##   H 0.125 0.250
##   M 0.250 0.375
```

```
prop.table(tt, margin = 1) #Frecuencia relativa por sexo
```

```
##    ans
## sex         N         S
##   H 0.3333333 0.6666667
##   M 0.4000000 0.6000000
```

```
prop.table(tt, margin = 2) #Frecuencia relativa por respuesta
```

```
##    ans
## sex         N         S
##   H 0.3333333 0.4000000
##   M 0.6666667 0.6000000
```

```
colSums(tt) #Suma por columnas de las frecuencias abs.
```

```
## N S
## 3 5
```

```
rowSums(tt) #Suma por filas de las frecuencias abs.
```

```
## H M
## 3 5
```

```
colSums(prop.table(tt)) #Suma por col de las frec. relativas
```

```
##     N     S
## 0.375 0.625
```

```
rowSums(prop.table(tt)) #Suma por filas de las frec. rel.
```

```
##     H     M
## 0.375 0.625
```

```
apply(tt, FUN = sum, MARGIN = 1) #Suma por fila
```

```
## H M
## 3 5
```

```
apply(tt, FUN = sqrt, MARGIN = c(1,2)) #Raíz cuadrada por cada elemento (entrada) de la tabla
```

```
##    ans
## sex        N        S
##   H 1.000000 1.414214
##   M 1.414214 1.732051
```

Una **tabla de contingencia bidimensional** es, básicamente, una matriz con algunos atributos extra. En particular, podemos usar sobre estas tablas a mayoría de las funciones para matrices que tengan sentido para tablas:

- **rowSums()** y **colSums()** se pueden aplicar a una tabla y suman sus filas y sus columnas, respectivamente.
- También podemos usar sobre una tabla bidimensional (o, en general, multidimensional)la función **apply()** con la misma sintaxis que para matrices.

```
table(Sexo, Respuestas)
```

```
##     Respuestas
## Sexo No Si
##    H  2  4
##    M  1  5
```

```
#frecuencias absolutas:
colSums(table(Sexo, Respuestas))
```

```
## No Si
##  3  9
```

```
rowSums(table(Sexo, Respuestas))
```

```
## H M
## 6 6
```

```
#frecuencias relativas:
colSums(prop.table(table(Sexo, Respuestas)))
```

```
##   No   Si
## 0.25 0.75
```

```
rowSums(prop.table(table(Sexo, Respuestas)))
```

```
##   H   M
## 0.5 0.5
```

MULTIVARIANTE/ESTADÍSTICA MULTIDIMENSIONAL

En general, se pueden calcular tablas de frecuencias de cualquier número de variables (no solo una o dos). Ahora hay más variables a la hora de contar y calcular las frecuencias.

Ejemplo con tres dimensiones+

```
ans2 = sample(c("Si", "No"), size = 100, replace = TRUE)
sex2 = sample(c("H", "M"), size = 100, replace = TRUE)
place = sample(c("San Francisco", "Barcelona", "Valencia", "Cobija", "Asturias"), size = 100, replace =
table(sex2, ans2, place)
```

```
## , , place = Asturias
##
##     ans2
## sex2 No Si
##    H  8  5
##    M  4  3
##
## , , place = Barcelona
##
##     ans2
## sex2 No Si
```

```
##    H 10  1
##    M  5  7
##
## , , place = Cobija
##
##     ans2
## sex2 No Si
##    H  4  8
##    M  8  2
##
## , , place = San Francisco
##
##     ans2
## sex2 No Si
##    H  5  5
##    M  0  8
##
## , , place = Valencia
##
##     ans2
## sex2 No Si
##    H  4  2
##    M  4  7
```

Se obtiene una lista de tablas bidimensionales separando la población según el nivel de la tercera variable (lugar de procedencia). Si no nos gusta esta forma de visualizar los datos tenemos la función **ftable()** que nos crea una tabla en un formato plano (un poco más complicado de visualizar), juntando la información sin separarla en subtablas bidimensionales.En esta función se puede especificar qué variable queremos que aparezca como fila y qué variable queremos que aparezca como columna.

```
ftable(sex2,ans2,place)
```

```
##           place Asturias Barcelona Cobija San Francisco Valencia
## sex2 ans2
## H    No               8        10      4             5        4
##      Si               5         1      8             5        2
## M    No               4         5      8             0        4
##      Si               3         7      2             8        7
```

```
ftable(sex2,ans2,place, col.vars = c("sex2", "ans2")) #tabla más fácil de leer
```

```
##               sex2  H     M
##               ans2 No Si No Si
## place
## Asturias            8  5  4  3
## Barcelona          10  1  5  7
## Cobija              4  8  8  2
## San Francisco       5  5  0  8
## Valencia            4  2  4  7
```

Filtrar las tablas

```
table(sex2,ans2,place)["M", "Si", "San Francisco"]
```

```
## [1] 8
```

```r
table(sex2,ans2,place)[ , "Si", "Valencia"] #quiero los dos géneros que han dicho que si en Valencia
```

```
## H M
## 2 7
```

```r
table(sex2,ans2,place)[ , "No", ] #quiero toda la gente que ha dicho que no
```

```
##     place
## sex2 Asturias Barcelona Cobija San Francisco Valencia
##    H        8        10      4             5        4
##    M        4         5      8             0        4
```

```r
table(sex2,ans2,place)["M", , "Cobija"] #quiero todas las mujeres de Cobija independientemente de su re
```

```
## No Si
##  8  2
```

Frecuencias relativas

Si la dimensión crece, la visualización de los datos se complica.

```r
prop.table(table(sex2,ans2,place)) #Frecuencias relativas globales
```

```
## , , place = Asturias
##
##     ans2
## sex2   No   Si
##    H 0.08 0.05
##    M 0.04 0.03
##
## , , place = Barcelona
##
##     ans2
## sex2   No   Si
##    H 0.10 0.01
##    M 0.05 0.07
##
## , , place = Cobija
##
##     ans2
## sex2   No   Si
##    H 0.04 0.08
##    M 0.08 0.02
##
## , , place = San Francisco
##
##     ans2
## sex2   No   Si
##    H 0.05 0.05
##    M 0.00 0.08
##
## , , place = Valencia
##
##     ans2
## sex2   No   Si
##    H 0.04 0.02
```

```
##    M 0.04 0.07
```

```
#Ejemplo lectura tabla: los hombres que han dicho que Si en San Francisco representan el 8% del total d

prop.table(table(sex2,ans2,place), margin = 3) #Frecuencia relativa marginal por lugar
```

```
## , , place = Asturias
##
##     ans2
## sex2         No         Si
##    H 0.40000000 0.25000000
##    M 0.20000000 0.15000000
##
## , , place = Barcelona
##
##     ans2
## sex2         No         Si
##    H 0.43478261 0.04347826
##    M 0.21739130 0.30434783
##
## , , place = Cobija
##
##     ans2
## sex2         No         Si
##    H 0.18181818 0.36363636
##    M 0.36363636 0.09090909
##
## , , place = San Francisco
##
##     ans2
## sex2         No         Si
##    H 0.27777778 0.27777778
##    M 0.00000000 0.44444444
##
## , , place = Valencia
##
##     ans2
## sex2         No         Si
##    H 0.23529412 0.11764706
##    M 0.23529412 0.41176471
```

```
#Ejemplo lectura tabla: de todas las personas que contestaron en San Frascisco, el 22% eran mujeres que

prop.table(table(sex2,ans2,place), margin = c(1,3)) #Frecuencia relativa marginal por sexo y país
```

```
## , , place = Asturias
##
##     ans2
## sex2         No         Si
##    H 0.61538462 0.38461538
##    M 0.57142857 0.42857143
##
## , , place = Barcelona
##
##     ans2
```

```
## sex2         No         Si
##    H 0.90909091 0.09090909
##    M 0.41666667 0.58333333
##
## , , place = Cobija
##
##     ans2
## sex2         No         Si
##    H 0.33333333 0.66666667
##    M 0.80000000 0.20000000
##
## , , place = San Francisco
##
##     ans2
## sex2         No         Si
##    H 0.50000000 0.50000000
##    M 0.00000000 1.00000000
##
## , , place = Valencia
##
##     ans2
## sex2         No         Si
##    H 0.66666667 0.33333333
##    M 0.36363636 0.63636364
```

```
#Ejemplo lectura tabla: el 25% de las mujeres de San Francisco han dicho que No. El 75% restante de las

ftable(prop.table(table(sex2,ans2,place)))
```

```
##           place Asturias Barcelona Cobija San Francisco Valencia
## sex2 ans2
## H    No            0.08      0.10   0.04          0.05     0.04
##      Si            0.05      0.01   0.08          0.05     0.02
## M    No            0.04      0.05   0.08          0.00     0.04
##      Si            0.03      0.07   0.02          0.08     0.07
```

Ejemplo multivariante HairEyeColor (people)

Ejemplo de color de ojos y de pelo

R trae definido de serie **HairEyeColor** como una tabla de frecuencias de tres variables cualitativas.No todos los objetos de datos vienen en estructura de dataframe.

La información viene agregada, como si entráramos en el Instituto Nacional de Estadística y obtenemos los datos agregados por grupos, no los row data (en crudo) (individuo a individuo).

```
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
```

```
sum(HairEyeColor) #Da el número de individuos totales (592)
```

```
## [1] 592
```

```
sum(HairEyeColor) -> total
```

El total de individuos de la tabla de datos es 592.

Ya que viene hecha la tabla de frecuencias absolutas, podemos hacer la tabla de frecuencias relativas marginales.

```
prop.table(HairEyeColor, margin = 3) #Frecuencia relativa marginal por género
```

```
## , , 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
```

Lectura de la tabla: el 11.46% de los hombres respecto al total de hombres tiene pelo negro y ojos marrones. Solo el 1.07% de los hombres tiene el pelo negro y los ojos verdes. Solo el 0.6% de las mujeres tiene el pelo negro y los ojos verdes.

```
prop.table(HairEyeColor, margin = c(1,2)) #Proporciones relativas marginales del color de pelo y ojos r
```

```
## , , 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
```

Lectura de la tabla: del total de personas que tienen el pelo negro y los ojos marrones, el 47% son hombres y el 52% son mujeres.De las personas con pelo negro y ojos verdes, el 60% son hombres y el 40% son mujeres.

Por ejemplo, si queremos cambiar el orden de variables y poner **Género** contra **Pelo** filtrado por **color de ojos** se puede hacer con la función *aperm*:

```
#Permite cambiar el orden de las columnas
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
```

kable librería que forma parte de Knit, sirve para mostrar la información en forma de dataframe de cuatro columnas (pelo,ojo,sexo y frecuencias).

```
#install.packages("kableExtra") #Primero instalar el paquete
library(kableExtra) #importar la librería
```

```
## Warning: package 'kableExtra' was built under R version 3.5.3
```

```
kable(HairEyeColor)
```

Hair	Eye	Sex	Freq
Black	Brown	Male	32
Brown	Brown	Male	53
Red	Brown	Male	10
Blond	Brown	Male	3
Black	Blue	Male	11
Brown	Blue	Male	50
Red	Blue	Male	10
Blond	Blue	Male	30
Black	Hazel	Male	10
Brown	Hazel	Male	25
Red	Hazel	Male	7
Blond	Hazel	Male	5
Black	Green	Male	3
Brown	Green	Male	15
Red	Green	Male	7
Blond	Green	Male	8
Black	Brown	Female	36
Brown	Brown	Female	66
Red	Brown	Female	16
Blond	Brown	Female	4
Black	Blue	Female	9
Brown	Blue	Female	34
Red	Blue	Female	7
Blond	Blue	Female	64
Black	Hazel	Female	5
Brown	Hazel	Female	29
Red	Hazel	Female	7
Blond	Hazel	Female	5
Black	Green	Female	2
Brown	Green	Female	14
Red	Green	Female	7
Blond	Green	Female	8

Hay otra librería que se puede utilizar para formatear tablas pero tienen que tener dos dimensiones:

```
#install.packages("xtable")
library(xtable)
```

```
## Warning: package 'xtable' was built under R version 3.5.3
```

```
#No funciona para HairEyeColor (porque tiene 3 dimensiones)
xtable(table(sex2,ans2))
```

% latex table generated in R 3.5.1 by xtable 1.8-4 package % Fri Nov 08 19:33:47 2019

	No	Si
H	31	21
M	21	27