

Estadística Descriptiva con Datos Cualitativos

Ramon Ceballos

24/1/2021

TABLAS DE FRECUENCIAS UNIDIMENSIONALES

Las tablas de frecuencias unidimensionales son aquellas que tienen una única variable.

Supongamos que tenemos una variable cualitativa guardada en un vector o un factor como la siguiente:

```
#Toma 12 número de la muestra que comprende los n°s del 1 al 5  
#replace=TRUE permite que se pueda obtener el mismo n° más de una vez  
#Es una variable cualitativa numérica  
x = sample(1:5, size = 12, replace = TRUE)  
x
```

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

```
#se genera otra variable cualitativa con sample()  
#En este caso toma los valores si y no  
#se genera un factor no ordenado  
Respuestas=factor(sample(c("Si", "No"), size = 12, replace = TRUE))  
Respuestas
```

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

1. Función table(). Frecuencias absolutas

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

```
#Proviene del ejemplo anterior  
table(x)
```

```
## x  
## 1 2 3 4 5  
## 2 1 2 6 1
```

```
#Proviene del ejemplo anterior  
table(Respuestas)
```

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

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.

Al aplicar `table()` a un vector obtenemos una tabla unidimensional formada por una fila (la primera) 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" "4" "5"
```

```
names(table(Respuestas))
```

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

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.

```
#Para asignar los niveles del factor se puede hacer lo siguiente
#También valdría para asignar más niveles a dichos datos cualitativos
z=factor(x, levels=1:7) #Los niveles serán 1,2,3,4,5,6,7
z
```

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

```
table(z)
```

```
## z
## 1 2 3 4 5 6 7
## 2 1 2 6 1 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(x)[3] #La tercera columna de table(x)
```

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

```
#No existe en esta tabla de contingencia
table(x)["7"] #¿La columna de table(x) con nombre 7?
```

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

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

```
## 5  
## 1
```

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

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

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

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

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

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

```
## Respuestas  
##      No      Si  
## 1.732051 3.000000
```

Ejemplos con la función `table()`.

Tablas de contingencia

```
datos = factor(c("H", "M", "M", "M", "H", "H", "M", "M"))  
#Creamos la tabla de contingencia del factor datos  
table(datos)
```

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

```
#Para acceder únicamente al nivel de mujeres se hace lo siguiente  
table(datos)["M"]
```

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

```
#Para obtener el total de observaciones dadas para dicha variable cualitativa  
sum(table(datos))
```

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

2. Función `prop.table()`. Frecuencias relativas

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      4      5
## 0.1666667 0.08333333 0.16666667 0.50000000 0.08333333
```

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

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

¡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 de un nivel diferente.

```
#No está bien hecho de este modo
```

```
prop.table(x)
```

```
## [1] 0.12820513 0.02564103 0.10256410 0.10256410 0.10256410 0.10256410
## [7] 0.07692308 0.07692308 0.10256410 0.05128205 0.10256410 0.02564103
```

```
#Vamos a ver un ejemplo más claro de la diferencia de lo dicho anteriormente
```

```
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.

```
#obtener frecuencias relativas a mano
```

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

```
## x
##      1      2      3      4      5
## 0.1666667 0.08333333 0.16666667 0.50000000 0.08333333
```

Dados un vector x y un número natural n , se puede hallar que niveles del factor tienen una frecuencia absoluta n en x con la instrucción:

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

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

```
table(x)
```

```
## x
## 1 2 3 4 5
## 2 1 2 6 1
```

```
#Nos da la table con los niveles que aparezcan una sola vez
#Solo pido las etiquetas de los niveles del factor
names(which(table(x)==1))
```

```
## [1] "2" "5"
```

En particular, por lo tanto, la instrucción:

```
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] "4"
```

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

```
## [1] "Si"
```

Ejemplos con la función `prop.table()`.

Frecuencias relativas

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

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

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

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

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

```
table(datos)/length(datos)
```

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

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

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

```
#Función para obtener la moda  
moda <- function(d){  
  names(which(table(d)==max(table(d))))  
}
```

```
#La moda sería:  
m_t = moda(datos)
```

La moda del data frame es: M = mujer.

EJERCICIOS PROPUESTOS

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.

```
#creo el vector gender  
gender = c(rep("H",6),rep("M",14))  
  
#frecuencia absoluta de gender  
fa = table(gender)  
fa
```

```
## gender
##   H   M
##  6 14
```

```
#frecuencias relativas de gender
```

```
fr = prop.table(fa)
fr
```

```
## gender
##   H   M
## 0.3 0.7
```

```
#moda de gender
```

```
moda_gender = names(which(table(gender)==max(table(gender))))
moda_gender
```

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