

Tabla de Frecuencias Unidimensionales

Santiago Pérez Moncada

3/6/2020

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

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

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

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

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

Tabla de Contingencia

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

La primera fila de la tabla tiene los niveles de la variable y la segunda fila contiene la frecuencia absoluta de cada nivel respectivamente.

```
table(x)
```

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

```
table(Respuestas)
```

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

Obtenemos los nombres de los niveles de la variable de la siguiente forma

```
names(table(x))
```

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

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

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

```
z = factor(x, levels = 1:7) #Los niveles seran 1,2,3,4,5,6,7  
z
```

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

```
table(z)
```

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

Podemos pensar que la tabla unidimensional es como un vector de numeros donde cada entrada esta identificada por un nombre:El de su columna. podemos usar tanto su posicion como su nombre.

```
table(x)[3] # la tercer columna de table(x)
```

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

```
table(Respuestas)['Si'] # la columna de si
```

```
## Si  
## 7
```

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

```
sum(table(x))
```

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

```
sqrt(table(Respuestas))
```

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

Ejemplo de Tablas de Contingencia

```
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

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

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

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

```
table(datos)/length(datos) * 100
```

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

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

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

```
moda <- function(d){  
  names(which(table(d) == max(table(d))))  
}
```

```
m_t = moda(datos) # Moda
```

La moda del dataFrame es: M.

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  
## 0.1666667 0.2500000 0.2500000 0.3333333
```

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

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

Tambien podemos calcular la tabla de frecuencias relativas de un vector dividiendo el resultado de table por el numero de observaciones.

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

```
## x
##           1           2           3           5
## 0.1666667 0.2500000 0.2500000 0.3333333
```

Dados un vector \vec{x} y un $n \in \mathbb{N}$, la instrucción `names(which(table(x)==n))` nos da los niveles que tienen frecuencia absoluta n en \vec{x}

```
table(x)
```

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

```
names(which(table(x)==1))
```

```
## character(0)
```

podemos obtener la **Moda** de la siguiente manera

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

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

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

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

Ejercicio

```
datos = c(rep("H",6),rep("M",14))
```

Tabla de Frecuencias Absolutas

```
table(datos)
```

```
## datos  
## H M  
## 6 14
```

Tabla de Frecuencias Relativas

```
prop.table(table(datos)) ## Forma una de hacerlo
```

```
## datos  
## H M  
## 0.3 0.7
```

```
table(datos)/length(datos) ## forma dos de hacerlo
```

```
## datos  
## H M  
## 0.3 0.7
```

Moda

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

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

```
table(datos)[table(datos) == max(table(datos))]
```

```
## M  
## 14
```