

# Tablas de Frecuencias Bidimensionales

Santiago Pérez Moncada

3/6/2020

## Frecuencias Absolutas

La función `table()` también permite construir tablas de frecuencias conjuntas de dos o más variables.

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

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

Supongamos que el vector `Respuestas` anterior contiene las respuestas de 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.

```
#H = hombre M = Mujer
Sexo = sample(c("H","M"), size = length(Respuestas), replace = TRUE)
Sexo
```

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

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

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

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

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

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

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

Para referirnos a una entrada de una tabla bidimensional podemos usar lo siguiente

```
table(Respuestas,Sexo)[1,2]
```

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

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

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

## Frecuencias Relativas

Como en el caso unidimensional, la funcion `prop.table()` sirve para calcular tablas bidimensionales de frecuencias relativas conjunta de pares de variable. Pero en el caso bidimensional tenemos dos **tipos de Frecuencias relativas**:

- **Frecuencias relativas globales:** para cada par e niveles, uno de cada variable, la fraccion de individuos que pertenece a ambos niveles respecto del total de la muestra.
- **Frecuencias relativas marginales:** dentro de cada nivel de una variable y para cada nivel de la otra, la fraccion de los individuos que pertenecen al segundo nivel respecto del total de la subpoblacion definida por el primer nivel.

Dadas dos variables, se pueden calcular dos familias de frecuencias relativas marginales, segun cual 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 fraccion de mujeres que han conterstado que sí respecto del toatal de mujeres, que la fraccion de mujeres que han contestado que sí respecto del total de personas que han dado la misma respuesta.

## Tabla de Frecuencias Relativas Globales

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

```
##      Respuestas
## Sexo      No      Si
##    H 0.08333333 0.3333333
##    M 0.41666667 0.1666667
```

## Tablas de Frecuencias Relativas Marginales

Para obtener las marginales, debemos usar el parametro `margin` al aplicar la funcion `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 0.2000000 0.8000000
##   M 0.7142857 0.2857143
```

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

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

## Tabla de frecuencias bidimensionales

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

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

## Paquete `gmodels`

```
library(gmodels)
sex = factor(c("H","M","M","M","M","H","H","M","M"))
ans = factor(c("S","N","S","N","N","S","N","S","S"))
CrossTable(sex, ans, prop.chisq = FALSE)
```

```
##
##
##      Cell Contents
## |-----|
## |                      N |
## |          N / Row Total |
## |          N / Col Total |
## |          N / Table Total |
## |-----|
##
##
## Total Observations in Table:  9
##
##
##          | ans
##      sex |      N |      S | Row Total |
## -----|-----|-----|-----|
```

```
##           H |           1 |           2 |           3 |
##           |           0.333 |           0.667 |           0.333 |
##           |           0.250 |           0.400 |           |
##           |           0.111 |           0.222 |           |
## -----|-----|-----|-----|
##           M |           3 |           3 |           6 |
##           |           0.500 |           0.500 |           0.667 |
##           |           0.750 |           0.600 |           |
##           |           0.333 |           0.333 |           |
## -----|-----|-----|-----|
## Column Total |           4 |           5 |           9 |
##           |           0.444 |           0.556 |           |
## -----|-----|-----|-----|
##
##
```

Una **tabla de contingencia bidimensional** es basicamente, una matriz con algunos atributos extras. En particular, podemos usar sobre estas tablas la 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 columnas respectivamente.
- También podemos usar sobre una tabla bidimensional (o , en general, multidimensional) la función `apply()` con la misma sintaxis para matrices.

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

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

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

```
## No Si
##   6   6
```

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

```
## H M
## 5 7
```

## Ejemplo

Sin usar `gmodels`

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

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

```
prop.table(tt) # relativas globales
```

```
##      ans
## sex      N      S
##   H 0.111111 0.222222
##   M 0.333333 0.333333
```

```
prop.table(tt, margin = 1) # relativas por filas(sexo)
```

```
##      ans
## sex      N      S
##   H 0.333333 0.666667
##   M 0.500000 0.500000
```

```
prop.table(tt, margin = 2) # relativas por columna(Respuesta)
```

```
##      ans
## sex      N      S
##   H 0.25 0.40
##   M 0.75 0.60
```

```
colSums(tt)
```

```
## N S
## 4 5
```

```
rowSums(tt)
```

```
## H M
## 3 6
```

```
colSums(prop.table(tt))
```

```
##      N      S
## 0.444444 0.555556
```

```
rowSums(prop.table(tt))
```

```
##      H      M
## 0.333333 0.666667
```