

# Estadística descriptiva con datos ordinales

Ramon Ceballos

29/1/2021

## 1. Tablas multidimensionales para datos ordinales

Para calcular frecuencias acumuladas en una tabla multidimensional, hay que aplicar a la tabla la función **cumsum** mediante la función **apply** que ya explicábamos para matrices. En este caso en concreto, la sintaxis de la instrucción sería: **apply(tabla, MARGIN = ..., FUN = cumsum)**

En esta expresión, el valor **MARGIN** ha de ser el de la dimensión en la que queremos acumular las frecuencias: 1 si queremos hacerlo por filas, 2 para hacerlo por columnas (también se podría hacer para una tercera variable y más). Lo veremos todo más claro con un ejemplo.

### Ejemplo 1

Se ha evaluado el tamaño de los cuellos de 100 jirafas. Los niveles que se han utilizado se los considera ordenados de la siguiente manera:

Muy.corto < Corto < Normal < Largo < Muy.largo

Los valores obtenidos en dicho estudio han sido los siguientes:

```
set.seed(2018)

longitud = sample(1:5,100, replace = TRUE)

longitud = ordered(longitud)

levels(longitud) = c("Muy.corto","Corto","Normal","Largo","Muy.largo")
```

```
#vector a estudiar del cuello de las jirafas
longitud
```

```
## [1] Normal      Largo      Muy.largo Corto      Muy.largo Muy.corto Normal
## [8] Largo      Corto      Largo      Normal      Normal    Muy.corto Muy.corto
## [15] Muy.largo Normal      Muy.corto Normal      Normal    Muy.largo Muy.corto
## [22] Largo      Corto      Muy.largo Normal      Largo      Muy.largo Muy.corto
## [29] Corto      Corto      Muy.corto Muy.largo Muy.largo Corto      Muy.corto
## [36] Corto      Muy.largo Muy.largo Corto      Muy.corto Corto      Muy.corto
## [43] Normal      Corto      Muy.corto Corto      Normal    Normal    Muy.corto
## [50] Corto      Normal      Muy.corto Largo      Largo      Corto      Muy.corto
## [57] Corto      Normal      Normal      Normal      Normal    Muy.corto Normal
```

```
## [64] Muy.corto Corto      Largo      Muy.corto Corto      Muy.corto Muy.largo
## [71] Muy.corto Corto      Muy.largo Largo      Muy.largo Normal    Corto
## [78] Corto      Normal    Largo      Largo      Corto      Corto      Muy.largo
## [85] Largo      Largo      Normal    Normal    Muy.corto Normal    Corto
## [92] Normal    Muy.corto Corto      Muy.corto Normal    Corto      Corto
## [99] Muy.corto Corto
## Levels: Muy.corto < Corto < Normal < Largo < Muy.largo
```

Supongamos que en el ejemplo anterior, el de las jirafas, estas provienen de 4 zonas diferentes (A, B, C y D), de manera que las 30 primeras son de la zona A, las 25 siguientes de la B, las 35 siguientes de la C y las 10 últimas de la D. Nos interesa estudiar la distribución de las longitudes según la zona.

Vamos a organizar todos estos datos en un data frame llamado **jirafas**. Para que nos sea más fácil visualizar la información, es conveniente que las filas de las tablas de frecuencias correspondan a las zonas. Por lo tanto, al definir el data frame, entraremos como primera variable la de la muestra las zonas. Así, conseguiremos que éstas aparezcan en las filas al aplicarle la función **table()**.

```
#se define el array de zonas (no es un vector ordenado)
zonas = rep(c("A","B","C","D"), c(30,25,35,10))
```

```
#configuramos el DF jirafas
jirafas = data.frame(zonas,longitud)
```

```
#Se disciernen dos variables, siendo:
# - Un factor sin ordenar.
# - Un factor ordenado.
str(jirafas)
```

```
## 'data.frame': 100 obs. of 2 variables:
## $ zonas : Factor w/ 4 levels "A","B","C","D": 1 1 1 1 1 1 1 1 1 1 ...
## $ longitud: Ord.factor w/ 5 levels "Muy.corto"<"Corto"<...: 3 4 5 2 5 1 3 4 2 4 ...
```

```
head(jirafas,4)
```

```
## zonas longitud
## 1 A Normal
## 2 A Largo
## 3 A Muy.largo
## 4 A Corto
```

### Tabla de frecuencias absolutas

Para calcular la tabla de frecuencias absolutas acumuladas de las longitudes por zonas y como las zonas definen las filas de la tabla anterior, debemos utilizar la función **apply** con **MARGIN = 1**.

```
#Se generan las frecuencias acumuladas por zonas (A,B,C,D)
apply(table(jirafas), MARGIN = 1, FUN = cumsum)
```

```
##          zonas
## longitud  A B C D
## Muy.corto 6 7 7 3
## Corto     11 15 15 8
## Normal    19 19 25 10
## Largo     24 21 31 10
## Muy.largo 30 25 35 10
```

Fijaos que la tabla se ha traspuesto la fila y la columna. Resulta que cuando se aplica **apply** a una **table** bidimensional, R intercambia, en caso de ser necesario, filas por columnas en el resultado para que la dimensión de la tabla resultante en la que se haya aplicado la función sea la de las columnas.

Con lo cual, para volver a tener las zonas en las filas, hay que **trasponer el resultado de la función apply**.

```
#transpuesta de la tabla que genera las frecuencias acumuladas por zonas
t(apply(table(jirafas), MARGIN = 1, FUN = cumsum))
```

```
##      longitud
## zonas Muy.corto Corto Normal Largo Muy.largo
##   A           6    11     19    24      30
##   B           7    15     19    21      25
##   C           7    15     25    31      35
##   D           3     8     10    10      10
```

### Tabla de frecuencias relativas acumuladas

Vamos ahora a calcular la tabla de frecuencias relativas acumuladas de las longitudes de cuello por zonas. Para conseguirlo, y en una única instrucción, primero calculamos la tabla de *frecuencias relativas por filas*, a continuación, con las funciones **apply** y **cumsum** las acumulamos y, finalmente, trasponemos el resultado.

```
#Obtenemos la tabla de frecuencias acumuladas por filas (zonas)
t(apply(prop.table(table(jirafas), margin = 1), MARGIN = 1, FUN = cumsum))
```

```
##      longitud
## zonas Muy.corto   Corto   Normal   Largo Muy.largo
##   A      0.20 0.3666667 0.6333333 0.8000000      1
##   B      0.28 0.6000000 0.7600000 0.8400000      1
##   C      0.20 0.4285714 0.7142857 0.8857143      1
##   D      0.30 0.8000000 1.0000000 1.0000000      1
```

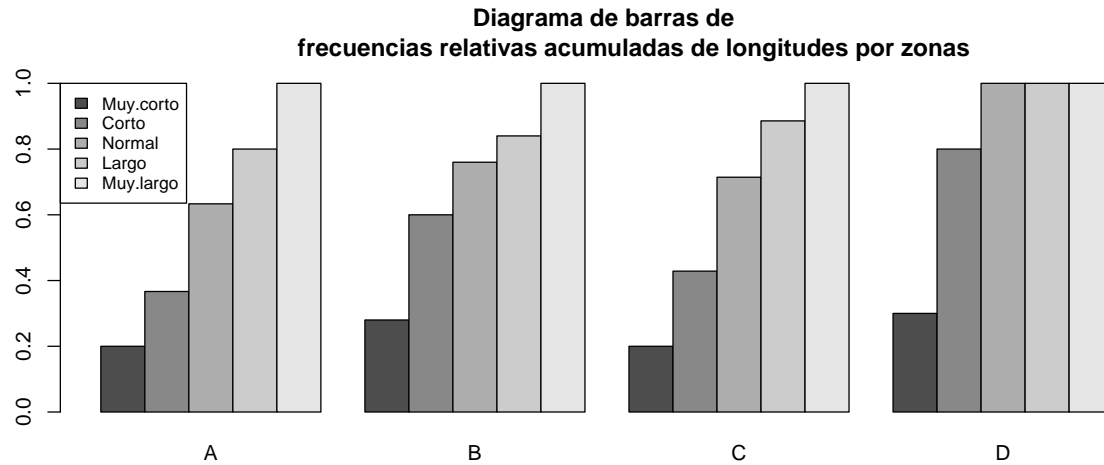
### Diagrama de Barras por bloques

Vamos ahora a dibujar el diagrama de barras por bloques de esta tabla. Nos interesa que las barras de este diagrama se agrupen por zonas. Entonces, tendremos que aplicar **barplot** a la tabla sin trasponer.

Además, vamos a colocar la leyenda en la esquina superior izquierda para que no se superponga a ninguna barra. También reduciremos el tamaño del texto de la leyenda para que quepa completamente.

```
Diagrama = apply(prop.table(table(jirafas), margin = 1), MARGIN = 1, FUN = cumsum)

barplot(Diagrama,
        beside = TRUE,
        legend = TRUE,
        main = "Diagrama de barras de
frecuencias relativas acumuladas de longitudes por zonas",
        args.legend=list(x="topleft", cex=0.8))
```



Se ha representado con las frecuencias relativas acumuladas en vez de las absolutas, para poder comparar las zonas con mayor facilidad.

```
Diagrama2 = t(prop.table(table(jirafas), margin = 1))

barplot(Diagrama2,
  beside = TRUE,
  legend = TRUE,
  main = "Diagrama de barras de
frecuencias relativas de longitudes por zonas",
  args.legend=list(x="topleft", cex=1),
  col=c("black","brown","red","lightgreen","lightblue"),
  ylim=c(0,0.6))
```

