Datos Ordinales Multidimensionales

Santiago Pérez Moncada

18/6/2020

Función `cumsum()`

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)
```

donde el valor `MARGIN` ha de sr el de la dimension en la que queremos acumularlas frecuencias: 1 si queremos hacerlo por filas, 2 para hacerlo por columnas, etc. Lo veremos todo más claro con un ejemplo.

Ejemplo 1

Supongamos que el ejemplo anterior, el de las jirafas, estas provienen de 4 zonas diferentes, de manera que las primeras 30 son de la zona `A`, las 25 siguientes de la `B`, las 35 siguientes de la zona `C` y las 10 ultimas de la `D`. Nos interesa estudiar la distribución de las longuitudes de los cuellos segun la zona.

Vamos a organizar todos estos datos en un dataframe llamado `jirafas`.Para que nos sea más facil visualizar la informacion, 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 de las zonas.Así, conseguiremos que éstas aparezcan en las filas al aplicarle la función table.

```
longitud = ordered(sample(c("Muy corto","Corto","Normal","Largo","Muy largo"),
                          size = 100, replace = TRUE))

zonas = rep(c("A","B","C","D"), c(30,25,35,10))

jirafas = data.frame(zonas,longitud)

str(jirafas)
```

```
## 'data.frame':    100 obs. of  2 variables:
##  $ zonas   : chr  "A" "A" "A" "A" ...
##  $ longitud: Ord.factor w/ 5 levels "Corto"<"Largo"<..: 5 2 1 4 5 1 3 4 4 5 ...
```

Veamos como quedo nuestro DataFrame

```
head(jirafas)
```

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

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

```
apply(table(jirafas), MARGIN = 1, FUN = cumsum)
```

```
##            zonas
## longitud     A  B  C  D
##   Corto      6  5 11  4
##   Largo     11 13 18  6
##   Muy corto 14 16 25  6
##   Muy largo 22 22 32  7
##   Normal    30 25 35 10
```

Veamos que la tabla se ha transpuesto.Resulta que cuando se aplica `apply` a una `tabla` 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 de las columnas.

Con lo cual, para volver a tener las zonas en las filas, hay que transponer el resultado de la fúncion `apply`.

```
t(apply(table(jirafas), MARGIN = 1, FUN = cumsum))
```

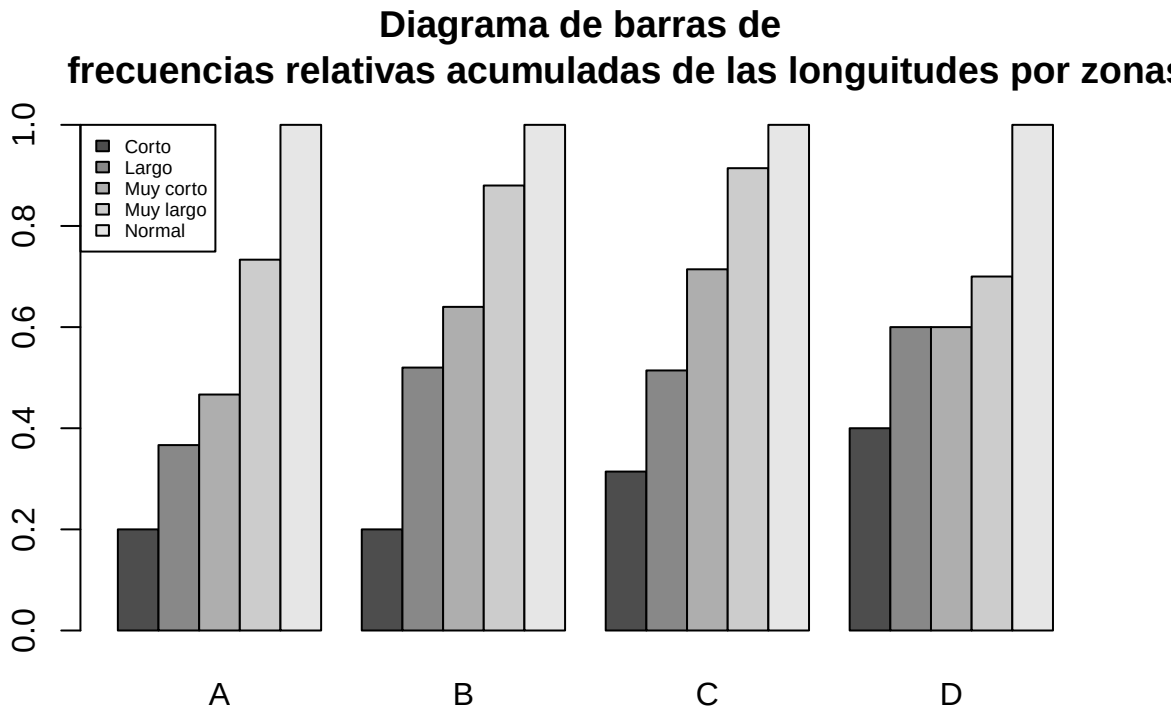
```
##      longitud
## zonas Corto Largo Muy corto Muy largo Normal
##     A     6    11        14        22     30
##     B     5    13        16        22     25
##     C    11    18        25        32     35
##     D     4     6         6         7     10
```

Vamos ahora a calcular la tabla de frecuencias relativas acumuladas de las longuitudes 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 transponemos el resultado para que las zonas me queden en las filas y sea mas sencillo de leer.

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

```
##      longitud
## zonas     Corto     Largo Muy corto Muy largo Normal
##     A 0.2000000 0.3666667 0.4666667 0.7333333      1
##     B 0.2000000 0.5200000 0.6400000 0.8800000      1
##     C 0.3142857 0.5142857 0.7142857 0.9142857      1
##     D 0.4000000 0.6000000 0.6000000 0.7000000      1
```

```r
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 las longuitudes por zonas",
        args.legend = list(x="topleft",cex=0.55))
```



Ejemplo 2

Consideremos el data frame `datacrab` y arreglemos los datos.

```r
crabs = read.table("../data/datacrab.txt", header = TRUE)
crabs = crabs[,2:length(crabs)] #omitimos la primer columna
str(crabs)
```

```
## 'data.frame':    173 obs. of  5 variables:
##  $ color : int  3 4 2 4 4 3 2 4 3 4 ...
##  $ spine : int  3 3 1 3 3 3 1 2 1 3 ...
##  $ width : num  28.3 22.5 26 24.8 26 23.8 26.5 24.7 23.7 25.6 ...
##  $ satell: int  8 0 9 0 4 0 0 0 0 0 ...
##  $ weight: int  3050 1550 2300 2100 2600 2100 2350 1900 1950 2150 ...
```

```r
head(crabs)
```

```
##   color spine width satell weight
## 1     3     3  28.3      8   3050
## 2     4     3  22.5      0   1550
## 3     2     1  26.0      9   2300
## 4     4     3  24.8      0   2100
## 5     4     3  26.0      4   2600
## 6     3     3  23.8      0   2100
```

La variable numérica `width` contiene la anchura de cada cangrejo.

```
table(crabs$width)
```

```
##
##   21   22 22.5 22.9   23 23.1 23.2 23.4 23.5 23.7 23.8 23.9   24 24.1 24.2 24.3
##    1    1    3    3    2    3    1    1    1    3    3    1    2    1    2    2
## 24.5 24.7 24.8 24.9   25 25.1 25.2 25.3 25.4 25.5 25.6 25.7 25.8 25.9   26 26.1
##    7    5    1    3    6    2    2    1    3    3    2    6    7    1    6    2
## 26.2 26.3 26.5 26.7 26.8   27 27.1 27.2 27.3 27.4 27.5 27.6 27.7 27.8 27.9   28
##    8    1    6    3    3    5    2    2    1    3    6    1    2    2    2    3
## 28.2 28.3 28.4 28.5 28.7 28.9   29 29.3 29.5 29.7 29.8   30 30.2 30.3 30.5 31.7
##    4    3    2    4    2    1    6    2    1    1    1    3    1    1    1    1
## 31.9 33.5
##    1    1
```

Vamos a convertir a la variable `width` en una variable ordinal que agrupe a las entradas de la variable original en niveles.

La manera mas sencilla de llevarlo a cabo es utilizando la funcion `cut` que estudiaremos en detalle en lecciones posteriores.Por ahora basta saber que la instrucción dividira el vector numérico `crabs$width` en intervalos de extremos los puntos especificados en el argumento `breaks`.El parámetro `right = FALSE` sirve pa indicar que los puntos de corte pertenecen al intervalos de su derecha e `Inf` indica infinito.

Por lo tanto, nosotros llevaremos a cabo la siguiente instrucción

```
intervalos = cut(crabs$width, breaks = c(21,25,29,33,Inf), right = FALSE,
                 labels = c("21-25","25-29","29-33","33-.."))

head(intervalos)
```

```
## [1] 25-29 21-25 25-29 21-25 25-29 21-25
## Levels: 21-25 25-29 29-33 33-..
```

El resultado de la instrucción es un factor que tiene como niveles estos intervalos, identificados con las etiquetas especificadas en el parámetro `labels`. Como nosotros vamos a usar estos intervalos como niveles de una variable ordinal, además convertiremos este factor en ordenado.

```
crabs$width.rank = ordered(intervalos)
str(crabs)
```

```
## 'data.frame':    173 obs. of  6 variables:
##  $ color     : int  3 4 2 4 4 3 2 4 3 4 ...
##  $ spine     : int  3 3 1 3 3 3 1 2 1 3 ...
##  $ width     : num  28.3 22.5 26 24.8 26 23.8 26.5 24.7 23.7 25.6 ...
```

```
##  $ satell    : int  8 0 9 0 4 0 0 0 0 0 ...
##  $ weight    : int  3050 1550 2300 2100 2600 2100 2350 1900 1950 2150 ...
##  $ width.rank: Ord.factor w/ 4 levels "21-25"<"25-29"<..: 2 1 2 1 2 1 2 1 1 2 ...
```

Nos interesa estudiar la distribución de las anchiras de los cangrejos según el numero de colores.Por lo tanto, vamos a calcular las tablas bidimensionales de frecuencias relativas y relativas acumuladas de los intervalos de las anchura en cada nivel de `color` y las representaremos por medio de diagramas de barras.

La tabla de frecuencias absolutas de los pares se puede obtener aplicando `table` al data frame formado por la primera y última columnas.

Frecuancia Absoluta

```
Tabla = table(crabs[,c(1,6)])
Tabla
```

```
##      width.rank
## color 21-25 25-29 29-33 33-..
##     2     1     9     2     0
##     3    19    62    13     1
##     4    17    24     3     0
##     5     9    12     1     0
```

Frecuancia relativa por fila

```
Fr.rel = round(prop.table(Tabla,margin = 1),3)
Fr.rel
```

```
##      width.rank
## color 21-25 25-29 29-33 33-..
##     2 0.083 0.750 0.167 0.000
##     3 0.200 0.653 0.137 0.011
##     4 0.386 0.545 0.068 0.000
##     5 0.409 0.545 0.045 0.000
```

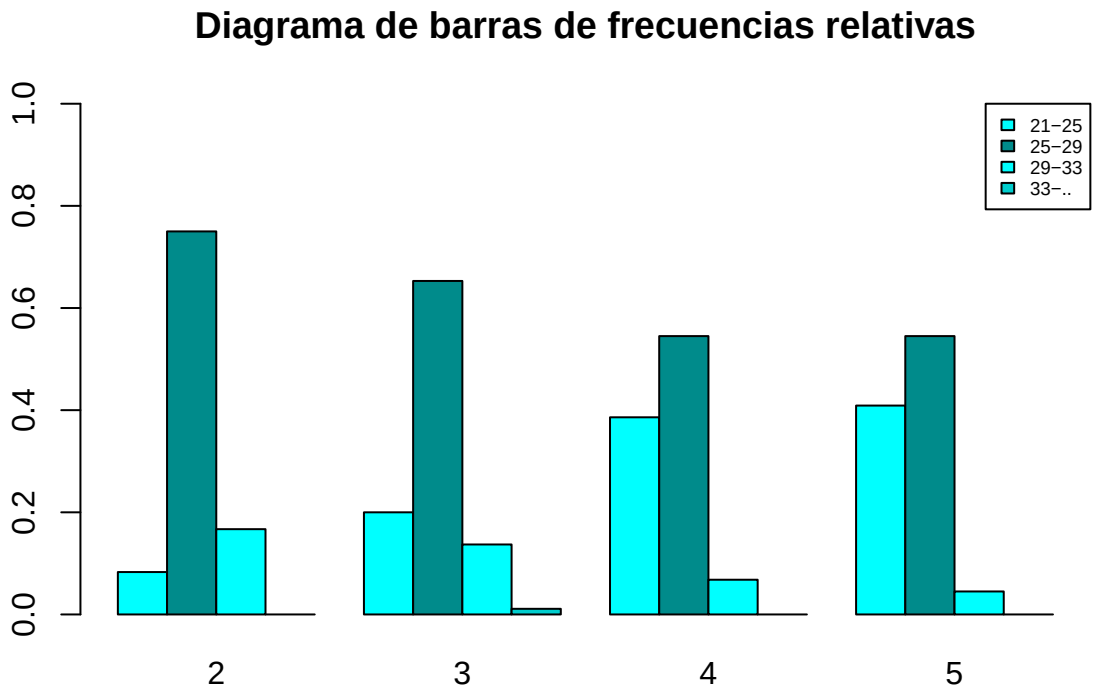
Frecuancia relativa acumulada por fila

```
Fr.rel.acu = round(apply(prop.table(Tabla, margin = 1), MARGIN = 1,FUN = cumsum),3)
t(Fr.rel.acu)
```

```
##      width.rank
## color 21-25 25-29 29-33 33-..
##     2 0.083 0.833 1.000     1
##     3 0.200 0.853 0.989     1
##     4 0.386 0.932 1.000     1
##     5 0.409 0.955 1.000     1
```

```r
azul = c("cyan","cyan4","cyan1","cyan3")

barplot(t(Fr.rel),beside = TRUE, legend=TRUE, ylim = c(0,1), col = azul,
        main = "Diagrama de barras de frecuencias relativas",
        args.legend = list(x="topright", cex=0.55))
```




```r
barplot(Fr.rel.acu,beside = TRUE, legend=TRUE, ylim = c(0,1), col = azul,
        main = "Diagrama de barras de frecuencias relativas acumuladas",
        args.legend = list(x="topleft", cex=0.55))
```

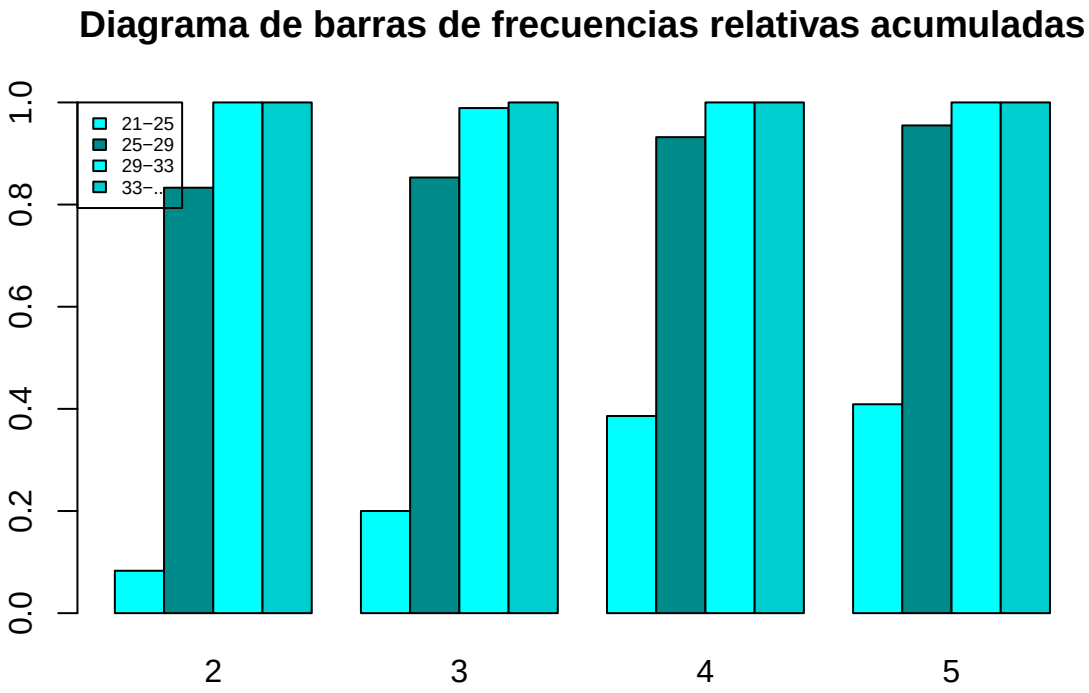


Diagrama de barras de frecuencias relativas acumuladas
21−25
25−29
29−33
33−..
0.0
0.2
0.4
0.6
0.8
1.0
2
3
4
5