Estadística descriptiva con datos ordinales

Ramon Ceballos

29/1/2021

EJEMPLO FINAL

Consideremos el data frame **datacrab** (medidas de los cangrejos) y arreglemos los datos.

```
#cargar el fichero datacrab
crabs = read.table("../../../data/datacrab.txt", header = TRUE)

#veo cual es la estructura de la tabla cargada
head(crabs,4)
```

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

```
#Omito la primera columna porque no aporta información útil (data wringling)
crabs = crabs[,-1] #Omitimos la primera columna

#Observo de nuevo el DF
head(crabs,4)
```

```
##   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
```

```
#Observo la estructura del DF
#Son variables numéricas
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 ...
```

La variable numérica **width** contiene la anchura de cada cangrejo. A priori no parece que sea una variable ordinal.

```
#Exploro específicamente la columna width
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 las entradas de la variable original en niveles.

La manera más sencilla de llevarlo a cabo es utilizando la función **cut()**, que estudiaremos en detalle en lecciones posteriores. Por ahora, basta con saber que la instrucción dividirá el vector numérico **crabs$width** en intervalos de extremos los puntos especificados en el argumento **breaks**. El parámetro **right = FALSE** sirve para indicar que los puntos de corte pertenecen la intervalo de su derecha o izquierda, e **Inf** indica ∞.

Por lo tanto, nosotros llevaremos a cabo la siguiente instrucción.

```
#En este caso los ptos de corte no estan incluidos en el intervalo que definen
#Están incluidos en el siguiente intervalo
#Se pueden establecer etiquetas (labels)
intervalos = cut(crabs$width,
                 breaks = c(21,25,29,33,Inf),
                 right = FALSE,
                 labels = c("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 nostros vamos a usar estos intervalos como niveles de una variable ordinal, además convertiremos este factor en ordenado.

```
#Se aplica al rango de la variable que queremos transformar a ordinal
#un ordered() del parámetros creado anteriormente
#Se genera una nueva columna en el DF llamada width.rank (última columna)
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 anchuras de los cangrejos según el número de colores. Por lo tanto, vamos a calcular las tablas bidimensionales de frecuencias relativas y relativas acumuladas de los intervalos de las anchuras 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 columna (color) y última columna (width.rank).

Tabla de frecuencias absolutas

```
#Tabla bidimensional (colores vs width.rank)
Tabla = table(crabs[,c(1,6)])
Tabla #FREC. ABS.
```

```
##       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
```

Tabla de frecuencias relativas marginales por fila

```
Fr.rel = round(prop.table(Tabla,margin = 1),3)
Fr.rel #FREC. REL. MARG. POR FILAS (COLOR)
```

```
##       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
```

Tabla de frecuencias relativas marginales por fila acumuladas

```
Fr.rel.acu = round(apply(prop.table(Tabla, margin = 1), MARGIN = 1, FUN = cumsum), 3)
t(Fr.rel.acu) #FREC. REL. MARG. POR FILAS ACUMULADA
```

```
##       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
```

Diagrama de Barras de las frecuecnias relativas marginales por filas

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

#barplot
#se ha transpuesto la tabla de frecuencias para asignar los colores
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=0.9))
```

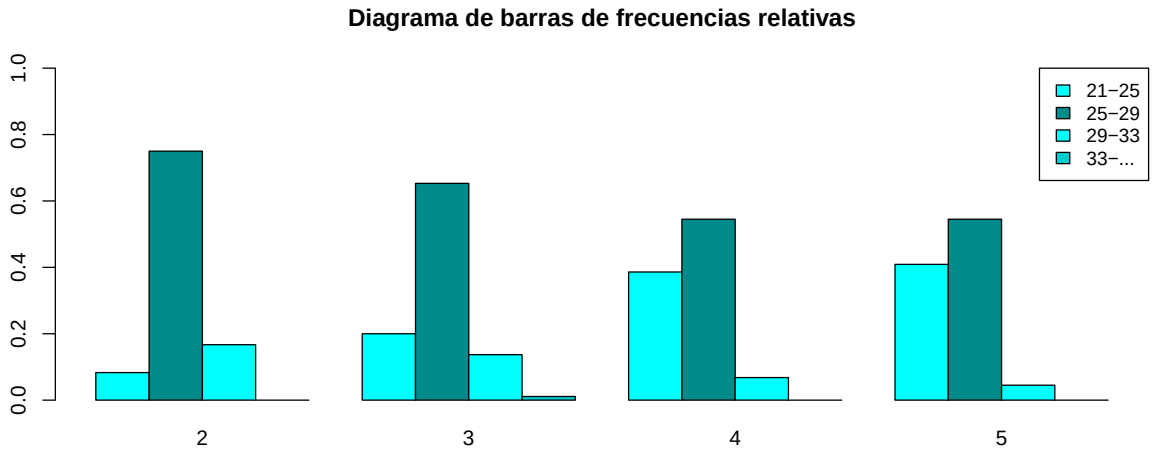

Diagrama de Barras de las frecuecnias relativas marginales por filas acumuladas

```
barplot(Fr.rel.acu,
        beside = TRUE,
        legend = TRUE,
        col = azul,
        main = "Diagrama de barras de frecuencias relativas acumuladas",
        args.legend=list(x = "topleft", cex=0.65))
```

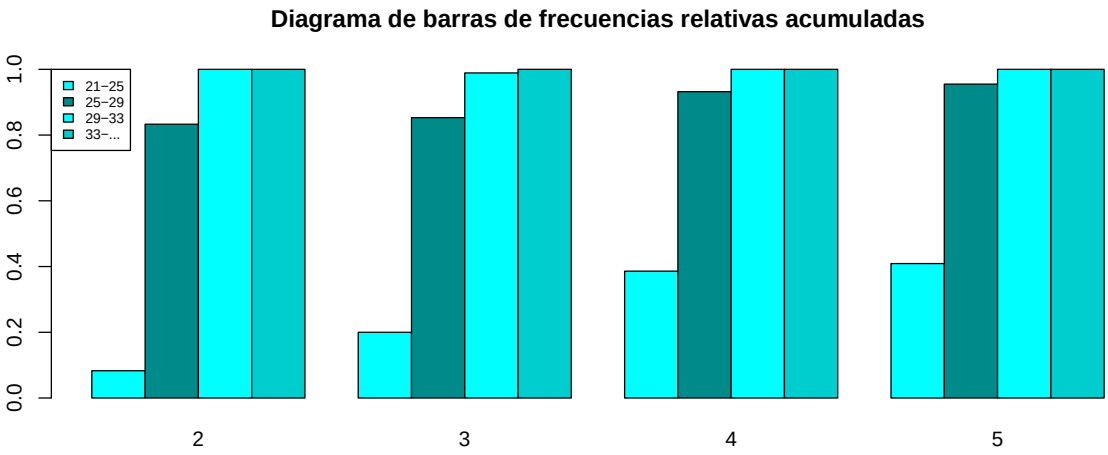
Diagrama de barras de frecuencias relativas acumuladas
21−25
25−29
29−33
33−...
0.0
0.2
0.4
0.6
0.8
1.0
2
3
4
5