Datos Ordinales

Oscar Gerardo Hernández Marínez

14/10/2019

Descripción de datos ordinales

Datos ordinales

Los datos ordinales son parecidos a los cualitativos, en el sentido de que son cualidades de los individuos u objetos.

La diferencia existente entre los datos cualitativos y los ordinales reside en las características que expresan. En el caso de los ordinales, éstas tienen un orden natural que permite "acumular" observaciones.

Frecuencias para datos ordinales

Frecuencia acumulada

Al trabajar con datos ordinales, el orden de los niveles de los datos nos permite calcular no solo frecuencias absolutas y relativas, sino también *frecuencias acumuladas*.

Es decir, podemos contar cuantas veces hemos observado un dato menor o igual a este.

Ejemplo 1

Supongamos que tenemos una muestra de 15 estudiantes de los cuales sabemos su nota en el examen de Estadística. Clasificamos todos estos resultados en Suspendido (S), Aprobado (A), Notable (N) y Excelente (Ex) y consideramos su orden natural $S < A < N < Ex$.

Las notas obtenidas han sido las siguientes

$$S,\ A,\ N,\ Ex,\ S,\ S,\ Ex,\ Ex,\ N,\ A,\ A,\ A,\ A,\ N,\ S$$

Como recordaremos, para saber cuantas hay de cada una (su frecuencia absoluta), utilizamos la función `table()`

```
notas = ordered(c("S","A", "N", "Ex", "S", "S", "Ex", "Ex", "N", "A", "A", "A",
                  "A", "N", "S"), levels = c("S", "A", "N", "Ex"))
table(notas)
```

```
## notas
##  S  A  N Ex
##  4  5  3  3
```

Como podréis observar, hay 4 S, 5 A, 3 N y 3 Ex.

En lo referente a **frecuencias absolutas acumuladas**, hay

- 4 estudiantes con S o menos. Ello implica que la frecuencia acumulada de S es 4
- 9 estudiantes que han obtenido A o menos. Entonces, la frecuencia acumulada de A es 9
- 12 estudiantes los cuales han obtenido N o menos. Así, la frecuencia acumulada de N es 12
- 15 estudiantes (todos) que han obtenido Ex o menos. De este modo, la frecuencia acumulada de Ex es 15, o sea, el total.

Frecuencia relativa acumulada.

Es la fracción del total de las observaciones en tanto por 1 que representa su frecuencia absoluta acumulada

Así, las recuencias relativas acumuladas respectivas son

- $S:\ \frac{4}{15} \approx 0.27$
- $A:\ \frac{9}{15} \approx 0.6$
- $N:\ \frac{12}{15} \approx 0.8$
- $Ex:\ \frac{15}{15} = 1$

En general, supongamos que realizamos n observaciones

$$x_1, \ldots, x_n$$

de un cierto tipo de datos ordinales, cuyos posibles niveles ordenados son

$$l_1 < l_2 < \cdots < l_k$$

Por tanto, cada una de las observaciones x_j es igual a algún l_i. Diremos que todas estas observaciones forman una *variable ordinal*. En nuestro ejemplo anterior, los 4 niveles eran

$$S < A < N < Ex$$

Además, nuestro $n = 15$ y nuestros $x_1, \ldots, x_{15}$ son las calificaciones obtenidas por los alumnos.

De este modo, con estas notaciones

- Las definiciones de frecuencias absolutas n_j y las relativas f_j, para cada nivel l_j son las mismas que en una variable cualitativa.
- Las frecuencia absoluta acumulada del nivel l_j en esta variable ordinal es el número N_j de observaciones x_i tales que $x_i \leq l_j$. Es decir,
$$N_j = \sum_{i=1}^{j} n_i$$
- La frecuencia relativa acumulada del nivel l_j en esta variable ordinal es la fracción en tanto por 1 F_j de observaciones x_i tales que $x_i \leq l_j$. Es decir,
$$F_j = \frac{N_j}{n} = \sum_{i=1}^{j} f_i$$

Ejemplo 2

En un estudio, a un grupo de clientes de un restaurante se les hizo la siguiente pregunta:

"¿Estás contento con el trato ofrecido por los trabajadores del establecimiento?"

Las posibles respuestas forman una escala ordinal con $1 < 2 < 3 < 4 < 5$.

Supongamos que se recogieron las siguientes respuestas de 50 técnicos:

```
set.seed(2018)
clientes = sample(1:5, 50, replace = TRUE)
clientes
```

```
##  [1] 3 4 5 2 5 1 3 4 2 4 3 3 1 1 5 3 1 3 3 5 1 4 2 5 3 4 5 1 2 2 1 5 5 2 1
## [36] 2 5 5 2 1 2 1 3 2 1 2 3 3 1 2
```

```r
set.seed(NULL)
```

En este caso tenemos 5 niveles ($k = 5$) y 50 observaciones ($n = 50$) que forman una variable ordinal a la que hemos llamado `clientes`.

Hemos calculado todas sus frecuencias (absoluta, relativa, acumulada y relativa acumulada) y las hemos representado en la siguiente talbla.

```
##   Absoluta Relativa Acumulada Rel. Acumulada
## 1       12     0.24        12           0.24
## 2       12     0.24        24           0.48
## 3       11     0.22        35           0.70
## 4        5     0.10        40           0.80
## 5       10     0.20        50           1.00
```

Ejercicio. Calcula todas las frecuencias y comprueba que son exactamente estas.

Los gráficos para frecuencias absolutas y relativas absolutas de variables ordinales son exactamente los mismos que para las variables cualitativas.

También podemos utilizar diagramas de barras para describir frecuencias acumuladas: en este caso, la altura de cada barra debe ser igual a la frecuencia acumulada del nivel respectivo. Además, estos niveles deben de aparecer ordenados de manera ascendente, de forma que las alturas de las barras también tengan un orden ascendente.

No obstante, se recomienda no hacer uso de diagramas circulares a la hora de representar frecuencias acumuladas, debido a que éstos no representan la información sobre la acumulación de datos de forma fácil de entender a simple vista.

Descripción de datos ordinales con R

Función cumsum()

¿Recuerdas la función `cumsum()`? Pues esta puede ser utilizada a la hora de calcular frecuencias acumuladas.

Retomemos el ejemplo anterior de las notas de los estudiantes y calculemos y representemos en un diagrama de barras las frecuencias acumuladas de la muestra de notas.

```r
notas
```

```
##  [1] S  A  N  Ex S  S  Ex Ex N  A  A  A  A  N  S
## Levels: S < A < N < Ex
```

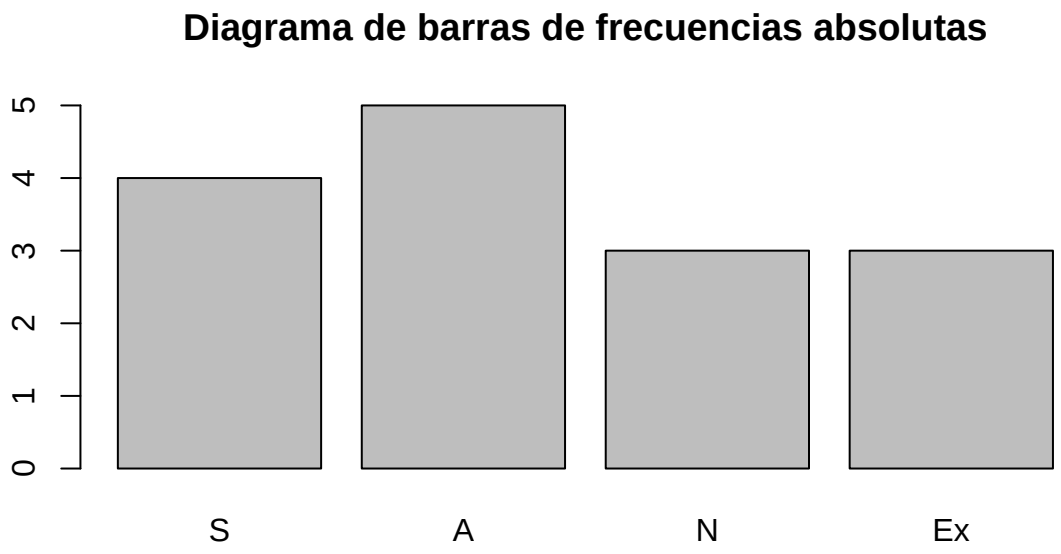
```r
fAbs = table(notas) #Frec. abs.
cumsum(fAbs) #Frec. abs. acumuladas
```

```
##  S  A  N Ex
##  4  9 12 15
```

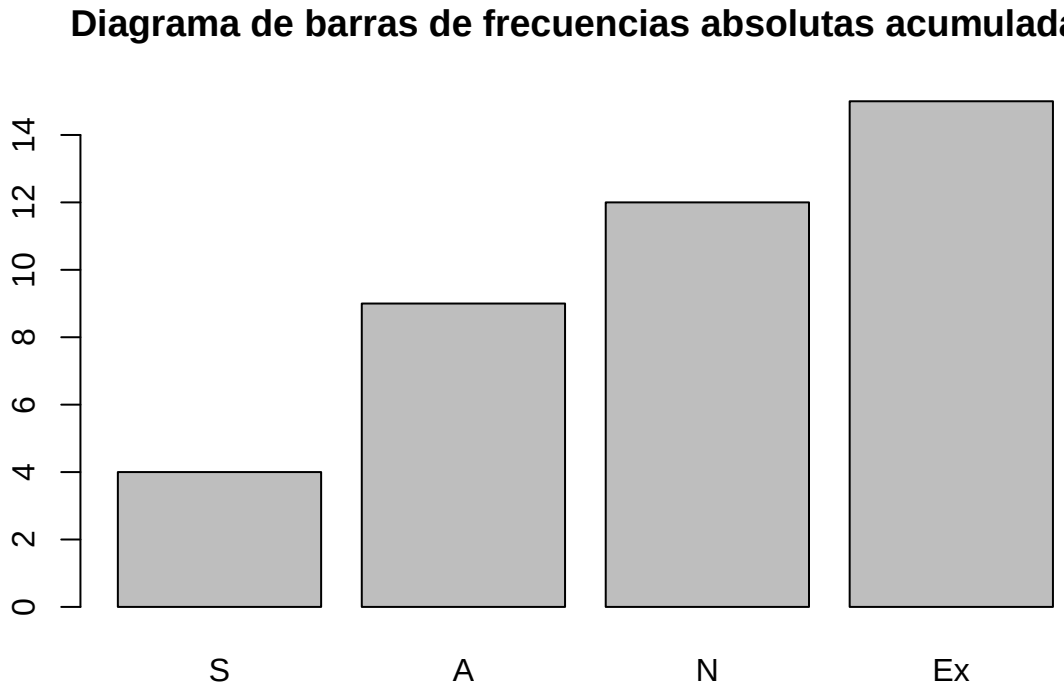
```r
cumsum(prop.table(fAbs)) #Frec. relativas acumuladas
```

```
##         S         A         N        Ex
## 0.2666667 0.6000000 0.8000000 1.0000000
```

```r
barplot(fAbs, main = "Diagrama de barras de frecuencias absolutas")
```




```
barplot(cumsum(fAbs), main = "Diagrama de barras de frecuencias absolutas acumuladas")
```



Podríamos haber calculado las frecuencias relativas acumuladas de la forma

```
cumsum(table(notas))/length(notas)
```

```
##         S         A         N        Ex
## 0.2666667 0.6000000 0.8000000 1.0000000
```

```
cumsum(table(notas)/length(notas))
```

```
##         S         A         N        Ex
## 0.2666667 0.6000000 0.8000000 1.0000000
```

Pero no podemos hacer `prop.table(cumsum(table(notas)))`.

Ejemplo 3

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:

$$\text{Muy.corto} < \text{Corto} < \text{Normal} < \text{Largo} < \text{Muy.largo}$$

Los valores obtenidos en dicho estudio han sido los siguientes

```
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
```

Estudiemos sus frecuencias

```
Fr.Abs = table(longitud)
Fr.Abs
```

```
## longitud
## Muy.corto     Corto    Normal     Largo Muy.largo
##        23        26        24        13        14
```

```
Fr.Rel = prop.table(Fr.Abs)
Fr.Rel
```

```
## longitud
## Muy.corto     Corto    Normal     Largo Muy.largo
##      0.23      0.26      0.24      0.13      0.14
```

```
Fr.Acum = cumsum(Fr.Abs)
Fr.Acum
```

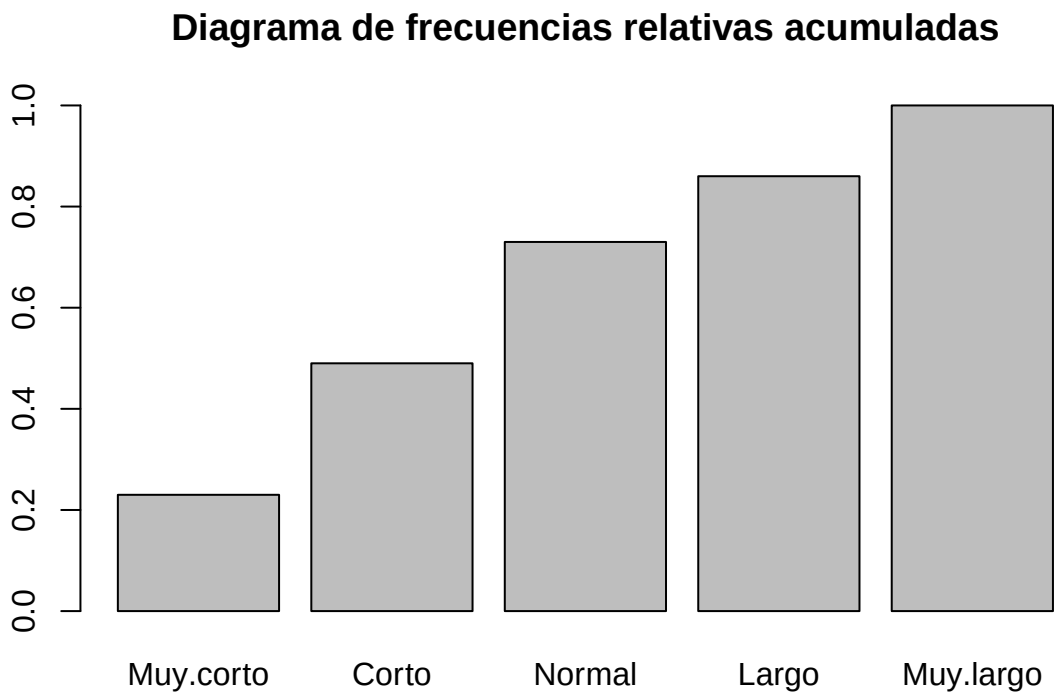
```
## Muy.corto      Corto     Normal      Largo Muy.largo
##        23         49         73         86        100
```

```
Fr.RAcum = cumsum(Fr.Rel)
Fr.RAcum
```

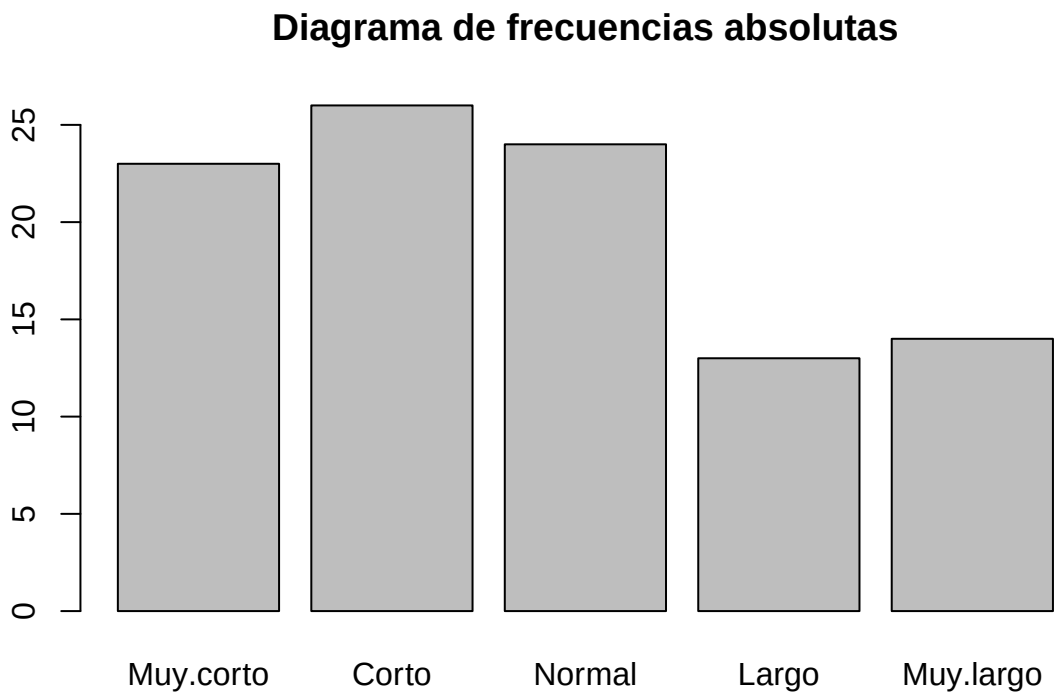
```
## Muy.corto      Corto     Normal      Largo Muy.largo
##      0.23       0.49       0.73       0.86       1.00
```

La instrucción `barplot` produce el siguiente diagrama de barras de frecuencias relativas acumuladas

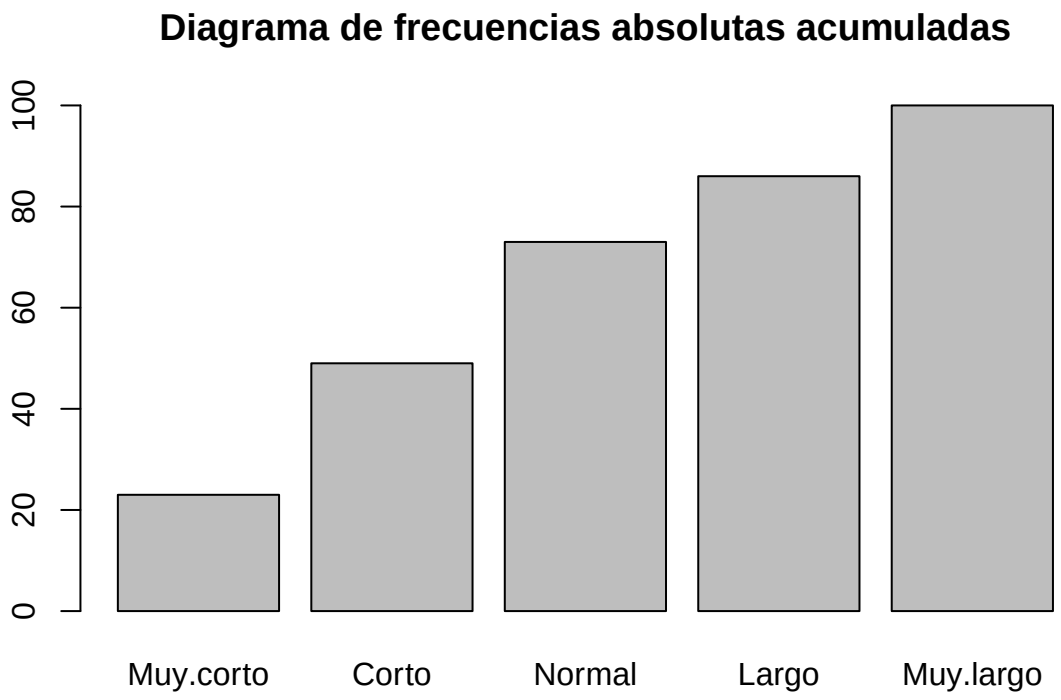
```
barplot(Fr.RAcum, main = "Diagrama de frecuencias relativas acumuladas")
```



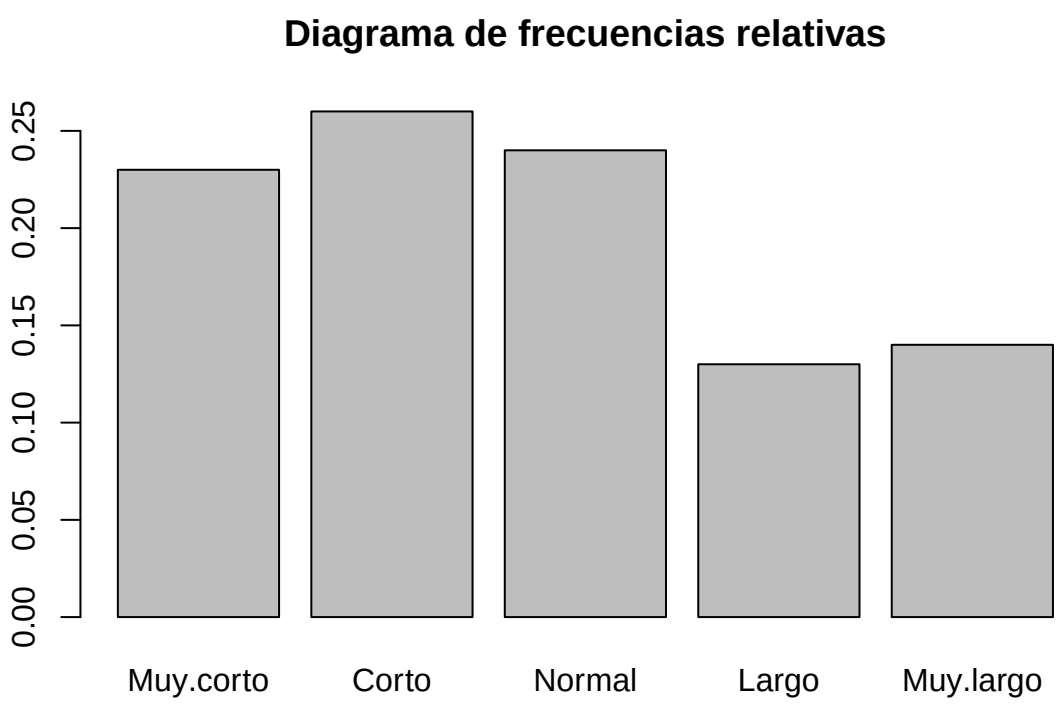

```
barplot(Fr.Abs, main = "Diagrama de frecuencias absolutas")
```




```r
barplot(Fr.Acum, main = "Diagrama de frecuencias absolutas acumuladas")
```




```r
barplot(Fr.Rel, main = "Diagrama de frecuencias relativas")
```



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 ser el de la dimensión en la que queremos acumular las frecuencias: 1 si queremos hacerlo por filas, 2 para hacerlo por columnas, etc. Lo veremos todo más claro con un ejemplo

Ejemplo 4

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.

```
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   : 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)
```

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

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`.

```
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
```

Observen que la tabla se ha traspuesto. 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`.

```
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
```

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.

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

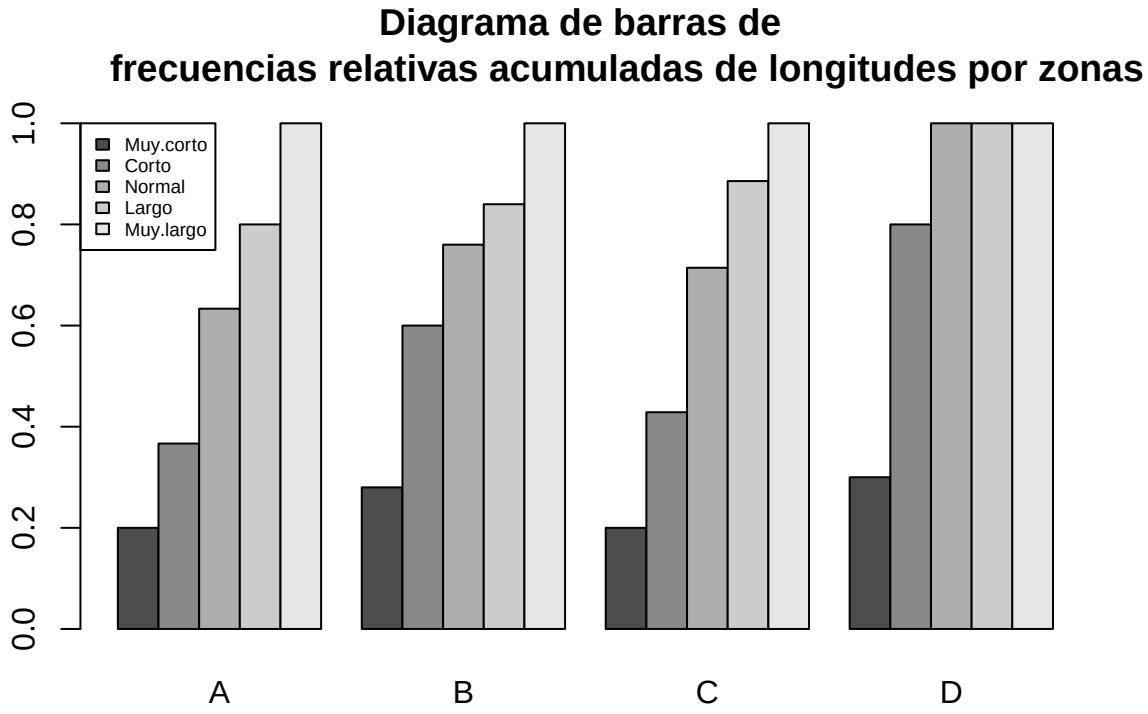
```
##      longitud
## zonas Muy.corto     Corto    Normal     Largo Muy.largo
##     A      0.20 0.3666667 0.6333333 0.8000000         1
##     B      0.28 0.6000000 0.7600000 0.8400000         1
##     C      0.20 0.4285714 0.7142857 0.8857143         1
##     D      0.30 0.8000000 1.0000000 1.0000000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

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=0.55))
```



Ejemplo 5

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

```
crabs = read.table("datacrab.txt", header = TRUE)
crabs = crabs[,-1] #Omitimos la primera 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 ...
```

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
##    1    1    3    3    2    3    1    1    1    3    3    1    2    1    2
## 24.3 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
##    2    7    5    1    3    6    2    2    1    3    3    2    6    7    1
##   26 26.1 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
```

```
##    6    2    8    1    6    3    3    5    2    2    1    3    6    1    2
## 27.8 27.9   28 28.2 28.3 28.4 28.5 28.7 28.9   29 29.3 29.5 29.7 29.8   30
##    2    2    3    4    3    2    4    2    1    6    2    1    1    1    3
## 30.2 30.3 30.5 31.7 31.9 33.5
##    1    1    1    1    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, e `Inf` indica ∞.

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-..."))
```

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.

```
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 frecuencais 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 y última columnas.

```
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
```

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

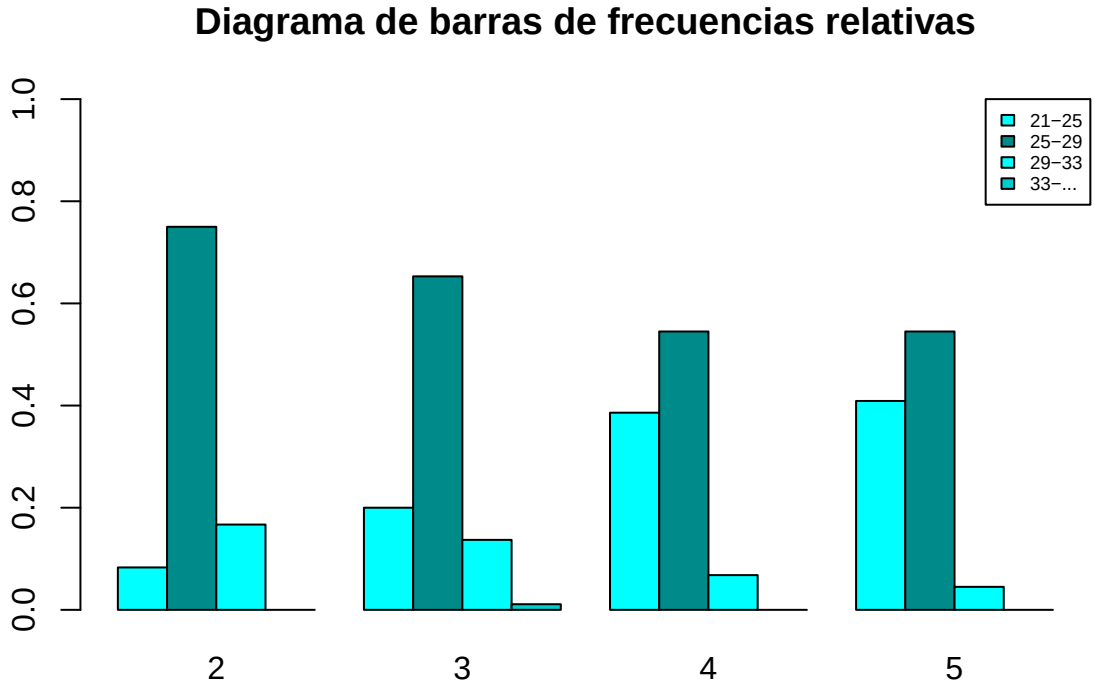
```
##      width.rank
## color 21-25 25-29 29-33 33-...
##     2 0.083 0.750 0.167  0.000
##     3 0.200 0.653 0.137  0.011
##     4 0.386 0.545 0.068  0.000
##     5 0.409 0.545 0.045  0.000
```

```r
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, col = azul,
        main = "Diagrama de barras de frecuencias relativas acumuladas",
        args.legend=list(x = "topleft", cex=0.55))
```

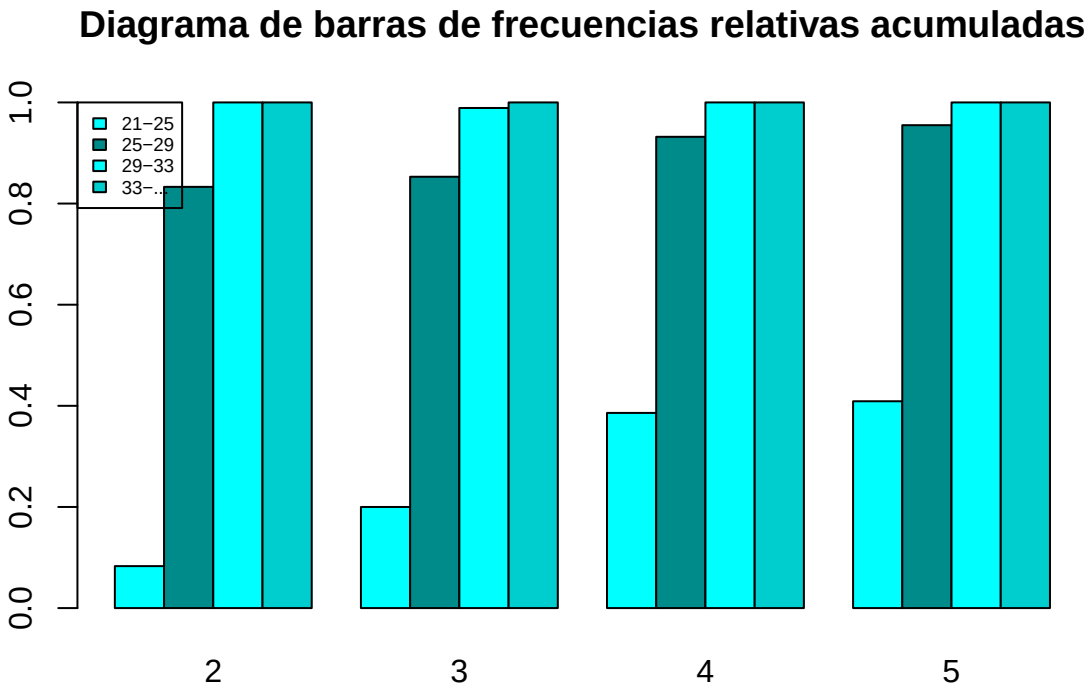


Diagrama de barras de frecuencias relativas acumuladas
21−25
25−29
29−33
33−..
0.0
0.2
0.4
0.6
0.8
1.0
2
3
4
5