

# Tema 7 - Estadística descriptiva con datos ordinales

Juan Gabriel Gomila & María Santos

## 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

### Ejemplo 1

Suponed que tenemos una muestra de 15 estudiantes de los cuales sabemos su nota en el examen de Estadística. Clasificamos todos estos resultados en Suspenso ( $S$ ), 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 recordaréis, para saber cuantas hay de cada una (su frecuencia absoluta), utilizamos la función `table()`

## Ejemplo 1

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

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

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

## Ejemplo 1

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.



## Ejemplo 1

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

Así, las frecuencias relativas acumuladas respectivas son

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

## Frecuencia relativa acumulada

En general, supongamos que realizamos  $n$  observaciones

$$x_1, \dots, x_n$$

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

$$l_1 < l_2 < \dots < l_k$$

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

$$S < A < N < Ex$$

## Frecuencia relativa acumulada

Además, nuestro  $n = 15$  y nuestros  $x_1, \dots, x_{15}$  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.
- ▶ La 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,

$$N_j = \sum_{i=1}^j n_i$$

## Frecuencia relativa acumulada

- 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

### 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
[39] 2 1 2 1 3 2 1 2 3 3 1 2
```

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

## Ejemplo 2

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 tabla.

	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.** Calculad todas las frecuencias y comprobad que son exactamente estas.

## Frecuencia relativa acumulada

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()

¿Recordáis 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.

```
notas
```

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

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

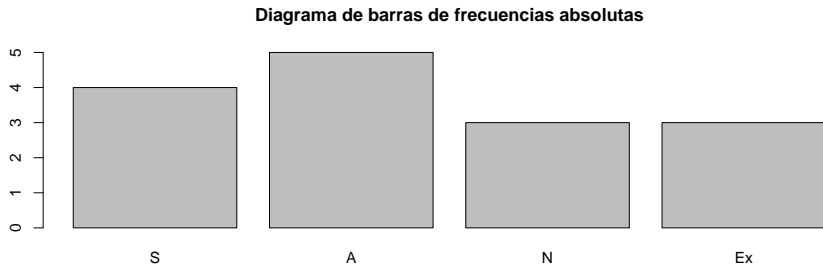
```
S  A  N Ex  
4  9 12 15
```

## Función cumsum()

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

	S	A	N	Ex
	0.2666667	0.6000000	0.8000000	1.0000000

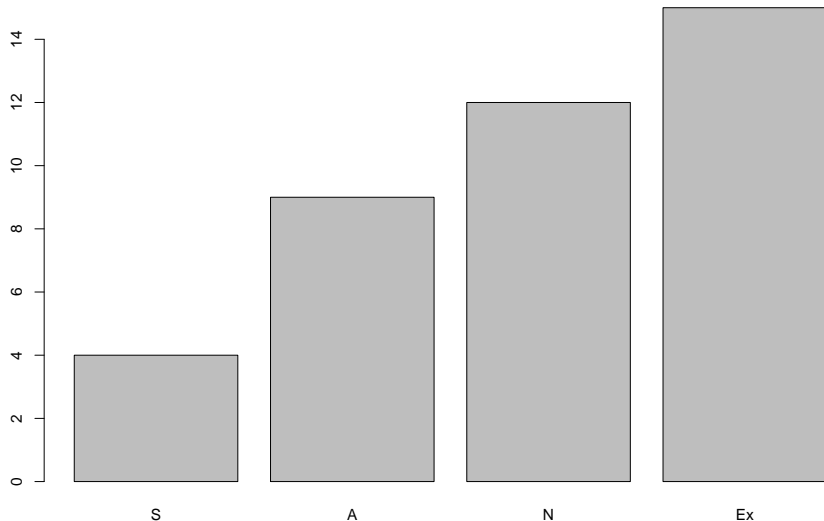
```
barplot(fAbs, main = "Diagrama de barras de frecuencias abs
```



## Función cumsum()

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

Diagrama de barras de frecuencias absolutas acumuladas



## Función cumsum()

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

## Función cumsum()

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

**Ejercicio.** Pensad qué ha entendido R que queríamos hacer con esta última instrucción.

## Ejemplo 3

### **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

## Ejemplo 3

longitud

[1]	Normal	Largo	Muy.largo	Corto	Muy.largo	Muy
[8]	Largo	Corto	Largo	Normal	Normal	Muy
[15]	Muy.largo	Normal	Muy.corto	Normal	Normal	Muy
[22]	Largo	Corto	Muy.largo	Normal	Largo	Muy
[29]	Corto	Corto	Muy.corto	Muy.largo	Muy.largo	Cor
[36]	Corto	Muy.largo	Muy.largo	Corto	Muy.corto	Cor
[43]	Normal	Corto	Muy.corto	Corto	Normal	Nor
[50]	Corto	Normal	Muy.corto	Largo	Largo	Cor
[57]	Corto	Normal	Normal	Normal	Normal	Muy
[64]	Muy.corto	Corto	Largo	Muy.corto	Corto	Muy
[71]	Muy.corto	Corto	Muy.largo	Largo	Muy.largo	Nor
[78]	Corto	Normal	Largo	Largo	Corto	Cor
[85]	Largo	Largo	Normal	Normal	Muy.corto	Nor
[92]	Normal	Muy.corto	Corto	Muy.corto	Normal	Cor
[99]	Muy.corto	Corto				

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

## Ejemplo 3

Estudiamos 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



## Ejemplo 3

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

```
Fr.Acum
```

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

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

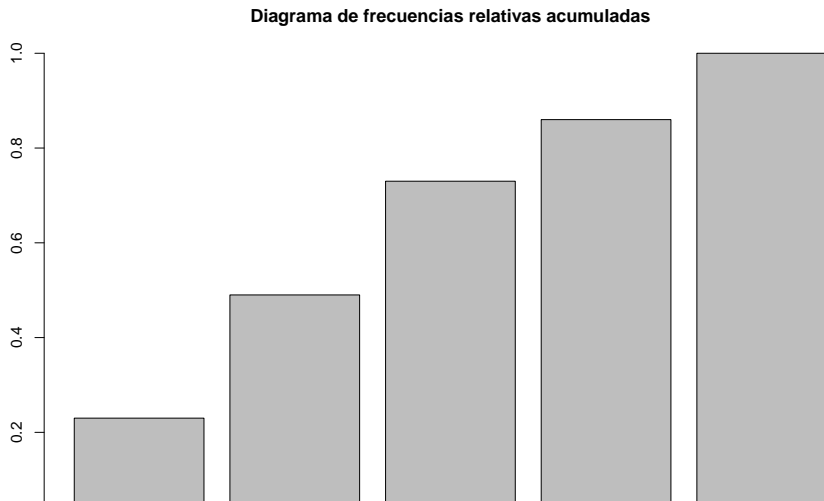
```
Fr.RAcum
```

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

## Ejemplo 3

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

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



## 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 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

### 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`.

## Ejemplo 4

```
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
```

```
$ longitud: Ord.factor w/ 5 levels "Muy.corto"<"Corto"<..
```

```
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

## Ejemplo 4

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

## Ejemplo 4

Fijaos 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 transponer 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

## Ejemplo 4

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,
```

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



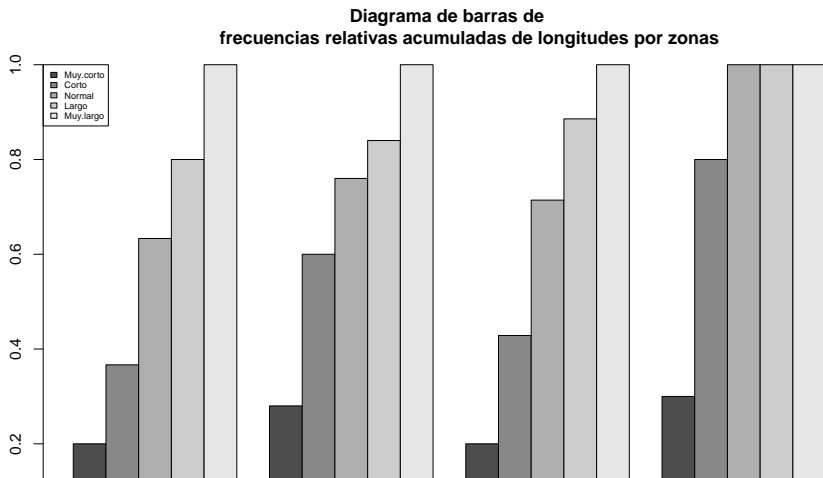
## Ejemplo 4

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.

## Ejemplo 4

```
Diagrama = apply(prop.table(table(jirafas), margin = 1), MARGIN=2, FUN=function(x){  
  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

### Ejemplo 5

Consideremos el data frame datacrab y arreglemos los datos.

```
crabs = read.table("../data/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
 $ satell: int    8 0 9 0 4 0 0 0 0 0 ...
 $ weight: int   3050 1550 2300 2100 2600 2100 2350 1900 195
```

La variable numérica width contiene la anchura de cada cangrejo

## Ejemplo 5

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

[illegible]

## Ejemplo 5

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  $\infty$ .

Por lo tanto, nosotros llevaremos a cabo la siguiente instrucción

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

## Ejemplo 5

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

## Ejemplo 5

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 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

## Ejemplo 5

```
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



## Ejemplo 5

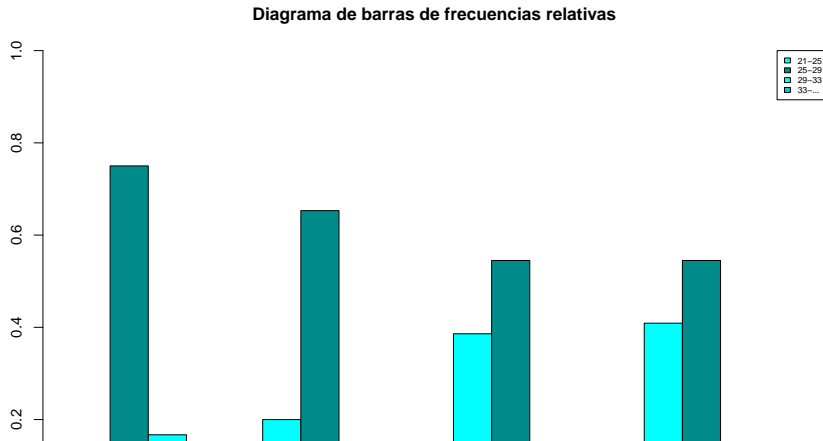
```
Fr.rel.acu = round(apply(prop.table(Tabla, margin = 1), MARGIN=2, FUN=FUN), 3)  
t(Fr.rel.acu)
```

	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

## Ejemplo 5

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

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



## Ejemplo 5

```
barplot(Fr.rel.acu, beside = TRUE, legend = TRUE, col = azu  
main = "Diagrama de barras de frecuencias relativas  
args.legend=list(x = "topleft", cex=0.55))
```

