Datos cuantitativos agrupados

OScar Gerardo Hernández Martínez

Introducción

Aunque no seamos completamente conscientes de ello, tendemos a agrupar datos cuantitativos constantemente.

Sin ir más lejos, calificamos de excelente a todas las notas que están sobre el 9. También decimos que una persona tiene 20 años cuando se encuentra en el intervalo [20,21). Es decir, cuando ha cumplido los 20 pero aún no tiene los 21.

En estadística, existen innumerables motivos por los cuales nos interesa agrupar los datos cuando estos son cuantitativos. Uno de estos motivos puede ser perfectamente que los datos sean muy heterogéneos. En este caso, nos encontraríamos con que las frecuencias de los valores individuales serían todas muy similares, lo que daría lugar a un diagrama de barras muy difícil de interpretar, tal y como mostramos en el siguiente ejemplo.

Ejemplo 1

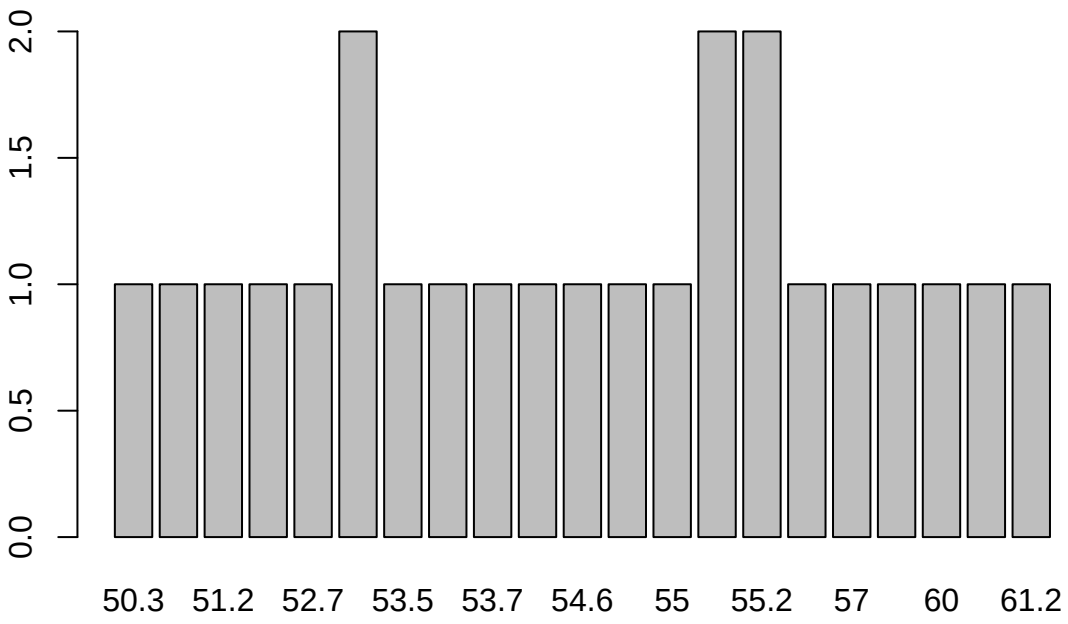
Consideremos la siguiente muestra de 24 pesos de estudiantes:

```
pesos = c(55.2,54.0,55.2,53.7,60.2,53.2,54.6,55.1,51.2,53.2,54.8,52.3,56.9,57.0,55.0,
          53.5,50.9,55.1,53.6,61.2,59.5,50.3,52.7,60.0)
```

El diagrama de barras de sus frecuencias absolutas, tomando como posibles niveles todos los pesos entre su mínimo y máximo se muestra en la siguiente diapositiva.

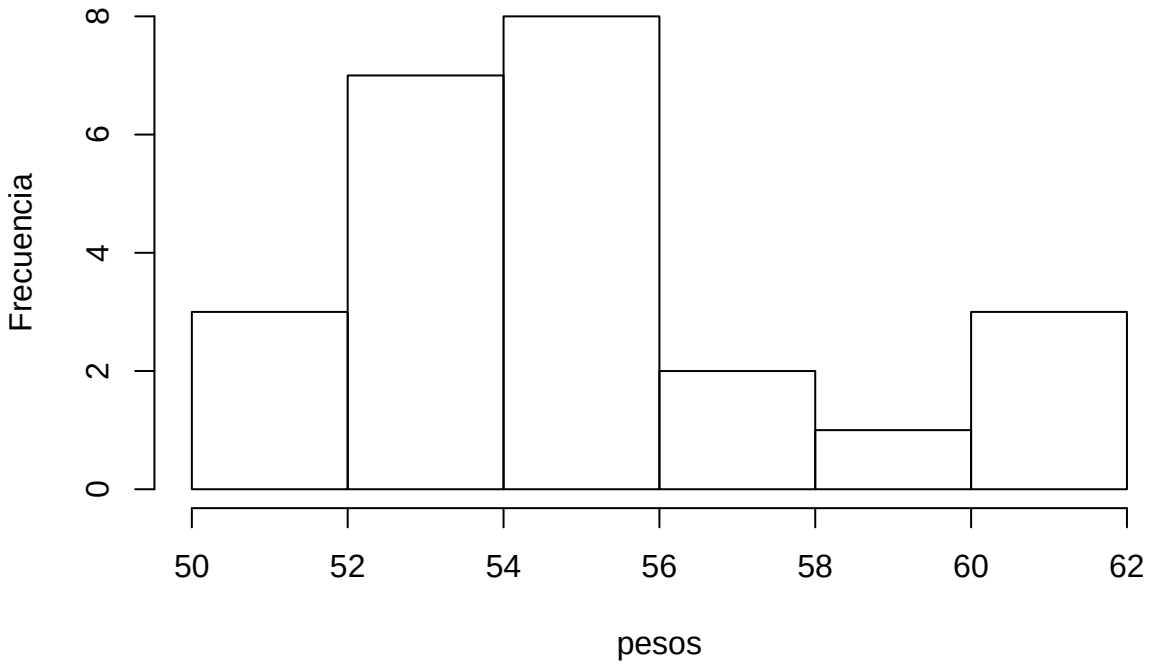
Como vemos, todas estas frecuencias se encuentran entre 0 y 2, cosa que no nos da mucha información.

```
barplot(table(pesos))
```



En cambio, si dividiésemos todos estos posibles valores que puede tomar la variable cuantitativa en intervalos y tomásemos como sus frecuencias las de todos los valores que caen en dicho intervalo, la cosa cambia.

En este caso, sería mucho más fácil interpretar los resultados, ya que estos darán mucha más información. Más adelante veremos como crear estos intervalos.



Otro de los motivos por el que necesitamos muchas veces agrupar los datos cuantitativos es porque, como ya dijimos en temas anteriores, la precisión infinita no existe. Por tanto, esta imposibilidad de medir de manera exacta muchas de las magnitudes continuas (tiempo, peso, altura...) nos obliga a trabajar con aproximaciones o redondeos de valores reales y que cada uno de estos represente todo un intervalo de posibles valores.

Por lo general, existen 3 situaciones en las cuales conviene sin lugar a dudas agrupar datos cuantitativos en intervalos, también llamados **clases**

- Cuando los datos son continuos, su redondeo ya define un agrupamiento debido a la inexistencia de precisión infinita
- Cuando los datos son discretos, pero con un número considerablemente grande de posibles valores
- Cuando tenemos muchísimos datos y estamos interesados en estudiar las frecuencias de sus valores

Cómo agrupar datos

Los 4 pasos

Antes de estudiar unos datos agrupados, hay que, obviamente, agruparlos. Este proceso consta de 4 pasos:

1. Decidir el número de intervalos que vamos a utilizar
2. Decidir la amplitud de estos intervalos
3. Acumular los extremos de los intervalos
4. Calcular el valor representativo de cada intervalo, **su marca de clase**

No hay una forma de agrupar datos mejor que otra. Eso sí, cada uno de los diferentes agrupamientos para un conjunto de datos podría sacar a la luz características diferentes del conjunto.

La función hist()

La función de R por excelencia para estudiar datos agrupados es `hist`. Dicha función implementa los 4 pasos del proceso.

Si le indicamos como argumentos el vector de datos y el número de intervalos que deseamos, o bien el método para determinarlo (cosa que veremos a continuación), la función agrupará los datos en el número de clases que le hemos introducido, más o menos. Eso sí, sin control de ningún tipo por nuestra parte sobre los intervalos que produce.

Esto puede venirnos bien en algunos casos, pero no en otros.

Estableciendo el número de clases

En este tema explicaremos una receta para agrupar datos. Lo dicho, ni mejor ni peor que el resto.

Lo primero es establecer el número k de clases en las que vamos a dividir nuestros datos. Podemos decidir en función de nuestros intereses o podemos hacer uso de alguna de las reglas existentes. Destacaremos las más populares. Sea n el número total de datos de la muestra

- **Regla de la raíz cuadrada:** $k = \lceil \sqrt{n} \ \rceil$
- **Regla de Sturges:** $k = \lceil 1 + \log_2(n) \rceil$
- **Regla de Scott:** Se determina primero la **amplitud teórica**, A_S de las clases

  $$A_S = 3.5 \cdot \tilde{s} \cdot n^{-\frac{1}{3}}$$

  donde $\tilde{s}$ es la desviación típica muestral. Luego se toma

  $$k = \left\lceil \frac{\max(x) - \min(x)}{A_S} \right\rceil$$

- **Regla de Freedman-Diaconis:** Se determina primero la **amplitud teórica**, A_{FD} de las clases

  $$A_{FD} = 2 \cdot (Q_{0.75} - Q_{0.25}) \cdot n^{-\frac{1}{3}}$$

  (donde, recordemos, $Q_{0.75} - Q_{0.25}$, es el rango intercuantílico) y entonces

  $$k = \left\lceil \frac{\max(x) - \min(x)}{A_{FD}} \right\rceil$$

Si os fijáis, las dos primeras solo dependen de n, mientras que las dos últimas también tienen en cuenta, de formas diferentes, la dispersión de los datos. De nuevo, no hay ninguna mejor que las demás. Pero sí puede ocurrir que métodos diferentes den lugar a la observación de características diferentes en los datos.

Estableciendo el número de clases con R

Las instrucciones para llevar a cabo las 3 últimas reglas con R son, respectivamente,

- `nclass.Sturges`
- `nclass.scott`
- `nclass.FD`

Puede ocurrir que las difrentes reglas den valores diferentes, o no.

Decidiendo la amplitud

Una vez determinado k, hay que decidir su amplitud.

La forma más fácil y la que nosotros utilizaremos por defecto es que la amplitud de todos los intervalos sea la misma, A. Esta forma no es la única.

Para calcular A, lo que haremos será dividir el rango de los datos entre k, el número de clases, y redondearemos por exceso a un valor de la precisión de la medida.

Si se da el improbable caso en que el cociente de exacto, tomaremos como A ese cociente más una unidad de precisión.

Extremos de los intervalos

Es la hora de calcular los extremos de los intervalos. Nosotros tomaremos estos intervalos siempre cerrados por su izquierda y abiertos por la derecha, debido a que esta es la forma en que R los construye y porque es así como se utilizan en Teoría de Probabilidades al definir la distribución de una variable aleatoria discreta y también en otras muchas situaciones cotidianas.

Utilizaremos la siguiente notación

$$[L_1, L_2), [L_2, L_3), \ldots, [L_k, L_{k+1})$$

donde los L_i denotan los extremos de los intervalos. Estos se calculan de la siguiente forma:

$$L_1 = \min(x) - \frac{1}{2} \cdot \text{precisión}$$

Extremos de los intervalos

A partir de L_1, el resto de intervalos se obtiene de forma recursiva:

$$L_2 = L_1 + A$$

$$L_3 = L_2 + A$$

$$\vdots$$

$$L_{k+1} = L_k + A$$

Si nos fijamos bien, los extremos forman una progresión aritmética de salto A:

$$L_i = L_1 + (i-1)A, \qquad i = 2, \ldots, k+1$$

De esta forma garantizamos que los extremos de los intervalos nunca coincidan con valores del conjunto de datos, puesto que tinen una precisión mayor.

Marca de clase

Solo nos queda determinar la **marca de clase**, X_i, de cada intervalo $[L_i, L_{i+1})$.

Este no es más que un valor del intervalo que utilizaremos para identificar la clase y para calcular algunos estadísticos.

Genralmente,

$$X_i = \frac{L_i + L_{i+1}}{2}$$

es decir, X_i será el punto medio del intervalo, para así garantizar que el error máximo cometido al describir cualquier elemento del intervalo por medio de su marca de clase sea mínimo o igual a la mitad de la amplitud del respectivo intervalo.

Es sencillo concluir que, al tener todos los intervalos amplitud A, la distancia entre X_i y X_{i+1} tambien será A. Por consiguiente,

$$X_i = X_1 + (i-1)A, \qquad i = 2, \ldots, k$$

donde

$$X_1 = \frac{L_1 + L_2}{2}$$

Ejemplo 2

Vamos a considerar el conjunto de datos de `datacrab`. Para nuestro estudio, trabajaremos únicamente con la variable `width`.

Llevaremos a cabo los 4 pasos explicados con anterioridad: cálculo del número de intervalos, determinación de la amplitud, cálculo de los extremos y las marcas de clase.

Solución

En primer lugar, cargamos los datos en un data frame:

```
crabs = read.table("datacrab.txt", header = TRUE)
str(crabs)
```

```
## 'data.frame':    173 obs. of  6 variables:
##  $ input : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ 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 ...
```

```
cw = crabs$width
```

A continuación, definimos la variable `cw` que contiene los datos de la variable `width`.

Solución

Calculemos el número de clases según las diferentes reglas que hemos visto:

- Regla de la raíz cuadrada:

```
n = length(cw)
k1 = ceiling(sqrt(n))
k1
```

```
## [1] 14
```

- Regla de Sturges:

```
k2 = ceiling(1+log(n,2))
k2
```

```
## [1] 9
```

- Regla de Scott:

```
As = 3.5*sd(cw)*n^(-1/3) #Amplitud teórica
k3 = ceiling(diff(range(cw))/As)
k3
```

```
## [1] 10
```

- Regla de Freedman-Diaconis:

```
#Amplitud teórica
Afd = 2*(quantile(cw,0.75, names = FALSE)-quantile(cw,0.25,names = FALSE))*n^(-1/3)
k4 = ceiling(diff(range(cw))/Afd)
k4
```

```
## [1] 13
```

Podemos comprobar nuestros 3 últimos resultados con R:

```
nclass.Sturges(cw)
```

```
## [1] 9
```

```
nclass.scott(cw)
```

```
## [1] 10
```

```
nclass.FD(cw)
```

```
## [1] 13
```

De momento, vamos a seguir la Regla de Scott. Es decir, vamos a considerar 10 intervalos.

A continuación, debemos elegir la amplitud de los intervalos.

```
A = diff(range(cw)) / 10
A
```

```
## [1] 1.25
```

Como nuestros datos están expresados en mm con una precisión de una cifra decimal, debemos redondear por exceso a un cifra decimal el resultado obtenido. Por lo tanto, nuestra amplitud será de

```
A = 1.3
```

Recordad que si el cociente nos hubiera dado un valor exacto con respecto a la precisión, tendríamos que haberle sumado una unidad de precisión.

Ahora nos toca calcular los extremos $L_1, \ldots, L_{11}$ de los intervalos.

Recordad que nuestros intervalos tendrán la siguiente forma:

$$[L_1, L_2), \ \ldots, \ [L_{10}, L_{11})$$

Calculamos el primer extremo:

```
L1 = min(cw)-1/2*0.1
L1
```

```
## [1] 20.95
```

donde 0.1 es nuestra precisión (décimas de unidad, en este caso).

Y, el resto de extremos se calculan del siguiente modo:

```
L2 = L1 + A
L3 = L2 + A
L4 = L3 + A
L5 = L4 + A
L6 = L5 + A
L7 = L6 + A
L8 = L7 + A
L9 = L8 + A
```

```
L10 = L9 + A
L11 = L10 + A
L = c(L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11)
L
```

```
##  [1] 20.95 22.25 23.55 24.85 26.15 27.45 28.75 30.05 31.35 32.65 33.95
```

O bien, si queremos facilitarnos el trabajo, también los podemos calcular mucho más rápido del siguiente modo:

```
L = L1 + A*(0:10)
L
```

```
##  [1] 20.95 22.25 23.55 24.85 26.15 27.45 28.75 30.05 31.35 32.65 33.95
```

Así, nuestros intervalos serán los siguientes:

$$[20.95, 22.25),\ [22.25, 23.55),\ [23.55, 24.85),\ [24.85, 26.15),\ [26.15, 27.45),$$
$$[27.45, 28.75),\ [28.75, 30.05),\ [30.05, 31.35),\ [31.35, 32.65),\ [32.65, 33.95)$$

Y hemos llegado al úlitmo paso: calcular las marcas de clase.

Recordemos que $X_i = \frac{L_i + L_{i+1}}{2} \quad \forall i = 1, \ldots, 10$

Empecemos calculando X_1

```
X1 = (L[1]+L[2])/2
X1
```

```
## [1] 21.6
```

Y, el resto de marcas de clase se calculan del siguiente modo:

```
X2 = X1 + A
X3 = X2 + A
X4 = X3 + A
X5 = X4 + A
X6 = X5 + A
X7 = X6 + A
X8 = X7 + A
X9 = X8 + A
X10 = X9 + A
X = c(X1,X2,X3,X4,X5,X6,X7,X8,X9,X10)
X
```

```
##  [1] 21.6 22.9 24.2 25.5 26.8 28.1 29.4 30.7 32.0 33.3
```

O bien, si queremos facilitarnos el trabajo, también los podemos calcular mucho más rápido como sucesión:

```
X = X1 + A*(0:9)
X
```

```
##  [1] 21.6 22.9 24.2 25.5 26.8 28.1 29.4 30.7 32.0 33.3
```

o también, como punto medio del intervalo

```
X = (L[1:length(L)-1]+L[2:length(L)])/2
X
```

```
##  [1] 21.6 22.9 24.2 25.5 26.8 28.1 29.4 30.7 32.0 33.3
```

Ejercicio

Repetir este proceso para el número de clases obtenido con

- la regla de la raíz
- la regla de Sturges
- la regla de Freedman-Diaconis

Agrupando datos con R

Agrupando los datos con R

Al agrupar los datos, lo que hacemos es convertir nuestra variable cuantitativa en un factor cuyos niveles son las clases en que ha sido dividida e identificamos cada dato con su clase.

A la hora de etiquetar los niveles, podemos elegir 3 codificaciones:

- Los intervalos
- Las marcas de clase (el punto medio de cada intervalo)
- El número de orden de cada intervalo

La función cut

Esta función es la básica en R para agrupar un vector de datos numéricos y codificar sus valores con clases a las que pertenecen.

Su sintaxis básica es

```
cut(x, breaks=..., labels=..., right=...)
```

- `x` es el vector numérico, nuestra variable cuantitativa
- `breaks` puede ser un vector numérico formado por los extremos de los intervalos en los que queremos agrupar nuestros datos y que habremos calculado previamente. También puede ser un número k, en cuyo caso R agrupa los datos en k clases. Para este caso, R divide el intervalo comprendido entre los valores mínimo y máximo de x en k intervalos y, a continuación, desplaza ligeramente el extremo inferior del primer intervalo a la izquierda y el extremo del último, a la derecha.
- `labels` es un vector con las etiquetas de los intervalos. Su valor por defecto es utilizar la etiqueta de los mismos intervalos. Si especificamos `labels = FALSE`, obtendremos los intervalos etiquetados por medio de los números naturales correlativos, empezando por 1. Para utilizar como etiqueta las marcas de clase o cualquier otra codificación, hay que entrarlo como valor de este parámetro.
- `right` es un parámetro que igualadao a `FALSE` hace que los intervalos que consideremos sean cerrados por la izquierda y abiertos por la derecha. Este no es su valor por defecto.
- `include.lowest` igualdo a `TRUE` combinado con `right = FALSE` hace que el último intervalo sea cerrado. Puede sernos útil en algunos casos.

En cualquier caso, el resultado de la función `cut` es una lista con los elementos del vector original codificados con las etiquetas de las clases a las que pertenecen. Bien puede ser un factor o un vector.

Estudiando datos agrupados

Frecuencias

Una primera consideración es tratar las clases obtenidas en el paso anterior como los niveles de una variable ordinal y calcular sus frecuencias.

- La frecuencia absoluta de una clase será el número de datos originales que pertenecen a la clase
- La frecuencia absoluta acumulada de una clase será el número de datos que pertenecen a dicha clase o alguna de las anteriores

Tabla de frecuencias

Normalmente, las frecuencias de un conjunto de datos agrupados se suele representar de la siguiente forma

Intervalos	X_j	n_j	N_j	f_j	F_j
$[L_1, L_2)$	X_1	n_1	N_1	f_1	F_1
$[L_2, L_3)$	X_2	n_2	N_2	f_2	F_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$[L_k, L_{k+1})$	X_k	n_k	N_k	f_k	F_k

La función hist

El cálculo de las frecuencias con R podemos hacerlo mediante las funciones `table`, `prop.table` y `cumsum`.

También podemos utilizar la función `hist`, que internamente genera una list cuya componente `count` es el vector de frecuencias absolutas de las clases. Por consiguiente, para calcular estas frecuencias, podemos utilizar la sintaxis

```
hist(x, breaks=..., right=FALSE, plot=FALSE)$count
```

Conviene igualar el parámetro `breaks` al vector de los extremos del intervalo debido a que `cut` y `hist` hacen uso de diferentes métodos para agrupar los datos cuando se especifica solamente el número k de clases.

El resultado de `hist` incluye la componente `mids` que contiene el vector de puntos medios de los intervalos, es decir, nuestras marcas de clase.

Tabla de frecuencias con R

Podemos automatizar el cálculo de la ya tan mencionada tabla de frecuencias, utilizando las dos funciones que mostramos a continuación.

La primera sirve en el caso en que vayamos a tomar todas las clases de la misma amplitud. Sus parámetros son: x, el vector con los datos cuantitativos; k, el número de clases; A, su amplitud; y p, la precisión de los datos (p = 1 si la precisión son unidades, p = 0.1 si la precisión son décimas de unidad...).

Por su parte, la segunda es para cuando conocemos los extremos de las clases. Sus parámetros son: x, el vector con los datos cuantitativos; L, el vector de extremos de clases; y V , un valor lógico, que ha de ser `TRUE` si queremos que el último intervalo sea cerrado, y `FALSE` en caso contrario.

```
#Primera función
TablaFrecs = function(x,k,A,p){
  L = min(x)-p/2+A*(0:k)
  x_cut = cut(x, breaks = L, right=FALSE)
  intervals = levels(x_cut)
  mc = (L[1]+L[2])/2+A*(0:(k-1))
  Fr.abs = as.vector(table(x_cut))
  Fr.rel = round(Fr.abs/length(x),4)
  Fr.cum.abs = cumsum(Fr.abs)
  Fr.cum.rel = cumsum(Fr.rel)
  tabla = data.frame(intervals, mc, Fr.abs, Fr.cum.abs, Fr.rel, Fr.cum.rel)
  tabla
  }
```

```
TablaFrecs.L = function(x,L,V){
  x_cut = cut(x, breaks=L, right=FALSE, include.lowest=V)
  intervals = levels(x_cut)
  mc = (L[1:(length(L)-1)]+L[2:length(L)])/2
```

```
  Fr.abs = as.vector(table(x_cut))
  Fr.rel = round(Fr.abs/length(x),4)
  Fr.cum.abs = cumsum(Fr.abs)
  Fr.cum.rel = cumsum(Fr.rel)
  tabla = data.frame(intervals, mc, Fr.abs, Fr.cum.abs, Fr.rel, Fr.cum.rel)
  tabla
  }
```

El ejemplo que se realizó usando estas fórmulas, está en el archivo de Clases-Ejercicio.

Siguiendo con el ejemplo de las anchuras de los cangrejos, vamos a calcular sus tablas de frecuencias haciendo uso de todo lo aprendido anteriormente.

Solución

La tabla queda del siguiente modo:

Intervalos	X_j	n_j	N_j	f_j	F_j
[20.95, 22.25)	21.6	2	2	0.0116	0.0116
[22.25, 23.55)	22.9	14	16	0.0809	0.0925
[23.55, 24.85)	24.2	27	43	0.1561	0.2486
[24.85, 26.15)	25.5	44	87	0.2543	0.5029
[26.15, 27.45)	26.8	34	121	0.1965	0.6994
[27.45, 28.75)	28.1	31	152	0.1792	0.8786

Intervalos	X_j	n_j	N_j	f_j	F_j
[28.75, 30.05)	29.4	15	167	0.0867	0.9653
[30.05, 31.35)	30.7	3	170	0.0173	0.9826
[31.35, 32.65)	32	2	172	0.0116	0.9942
[32.65, 33.95)	33.3	1	173	0.0058	1

Y, ahora, lo haremos con las funciones que os hemos proporcionado:

```
TablaFrecs(cw,10,1.3,0.1)
```

```
##      intervals   mc Fr.abs Fr.cum.abs Fr.rel Fr.cum.rel
## 1  [20.9,22.2) 21.6      2          2 0.0116     0.0116
## 2  [22.2,23.6) 22.9     14         16 0.0809     0.0925
## 3  [23.6,24.9) 24.2     27         43 0.1561     0.2486
## 4  [24.9,26.1) 25.5     44         87 0.2543     0.5029
## 5  [26.1,27.4) 26.8     34        121 0.1965     0.6994
## 6  [27.4,28.8) 28.1     31        152 0.1792     0.8786
## 7    [28.8,30) 29.4     15        167 0.0867     0.9653
## 8    [30,31.4) 30.7      3        170 0.0173     0.9826
## 9  [31.4,32.6) 32.0      2        172 0.0116     0.9942
## 10   [32.6,34) 33.3      1        173 0.0058     1.0000
```

```
TablaFrecs.L(cw,L,FALSE)
```

```
##      intervals   mc Fr.abs Fr.cum.abs Fr.rel Fr.cum.rel
## 1  [20.9,22.2) 21.6      2          2 0.0116     0.0116
## 2  [22.2,23.6) 22.9     14         16 0.0809     0.0925
## 3  [23.6,24.9) 24.2     27         43 0.1561     0.2486
```

```
## 4  [24.9,26.1) 25.5     44         87 0.2543     0.5029
## 5  [26.1,27.4) 26.8     34        121 0.1965     0.6994
## 6  [27.4,28.8) 28.1     31        152 0.1792     0.8786
## 7    [28.8,30) 29.4     15        167 0.0867     0.9653
## 8    [30,31.4) 30.7      3        170 0.0173     0.9826
## 9  [31.4,32.6) 32.0      2        172 0.0116     0.9942
## 10   [32.6,34) 33.3      1        173 0.0058     1.0000
```

Fijaos que los intervalos no terminan de ser los que hemos calculado nosotros, pero eso se debe a como funciona la función `cut`.

Ejemplo 3

Se han recogido las notas de un examen de historia a los 100 alumnos de primero de bachillerato de un instituto.

Vamos a hacer uso de todo lo aprendido para obtener la mayor información posible utilizando las funciones `cut` e `hist` y también, las proporcionadas por nosotros.

Solución

Los resultados obtenidos en la encuesta han sido:

```
notas
```

```
##   [1]  7 10  2  2  6  2  5  4  9  2  7  5  1  7  0  3 10  2 10  4  1  4  5
##  [24]  4  0  5 10  4  3  0  7  5 10  3  4  8  1  9  3  7  9  1  9 10  5 10
##  [47] 10  9  5  0  3  1  3  2  0  6  6  4  7  4  7  3  9  0  7  0  3  0  3
##  [70]  3  1  4 10  9  1  4  0  6 10  0 10  1  0  2  6  4  8  2  3  7  7  3
##  [93]  3  8  2  6  6  2  8  9
```

Vamos a agrupar las notas en los siguientes intervalos:

$$[0,5),\ [5,7),\ [7,9),\ [9,10]$$

Claramente, estos 4 intervalos no tienen la misma amplitud.

Fijémonos también en que el último intervalo está cerrado por la derecha.

```
#Definimos vector de extremos
L = c(0,5,7,9,10)
#Definimos notas1 como el resultado de la codificación en intervalos utilizando como
#etiquetas los propios intervalos
notas1 = cut(notas, breaks = L, right = FALSE, include.lowest = TRUE)
notas1
```

```
##   [1] [7,9)  [9,10] [0,5)  [0,5)  [5,7)  [0,5)  [5,7)  [0,5)  [9,10] [0,5)
##  [11] [7,9)  [5,7)  [0,5)  [7,9)  [0,5)  [0,5)  [9,10] [0,5)  [9,10] [0,5)
##  [21] [0,5)  [0,5)  [5,7)  [0,5)  [0,5)  [5,7)  [9,10] [0,5)  [0,5)  [0,5)
##  [31] [7,9)  [5,7)  [9,10] [0,5)  [0,5)  [7,9)  [0,5)  [9,10] [0,5)  [7,9)
##  [41] [9,10] [0,5)  [9,10] [9,10] [5,7)  [9,10] [9,10] [9,10] [5,7)  [0,5)
##  [51] [0,5)  [0,5)  [0,5)  [0,5)  [0,5)  [5,7)  [5,7)  [0,5)  [7,9)  [0,5)
##  [61] [7,9)  [0,5)  [9,10] [0,5)  [7,9)  [0,5)  [0,5)  [0,5)  [0,5)  [0,5)
##  [71] [0,5)  [0,5)  [9,10] [9,10] [0,5)  [0,5)  [0,5)  [5,7)  [9,10] [0,5)
##  [81] [9,10] [0,5)  [0,5)  [0,5)  [5,7)  [0,5)  [7,9)  [0,5)  [0,5)  [7,9)
##  [91] [7,9)  [0,5)  [0,5)  [7,9)  [0,5)  [5,7)  [5,7)  [0,5)  [7,9)  [9,10]
## Levels: [0,5) [5,7) [7,9) [9,10]
```

```r
#Definimos las marcas de clase
MC = (L[1:length(L)-1]+L[2:length(L)])/2
#Definimos notas2 como el resultado de la codificación en intervalos utilizando como
#etiquetas las marcas de clase
notas2 = cut(notas, breaks = L, labels = MC, right = FALSE, include.lowest = TRUE)
notas2
```

```
##   [1] 8   9.5 2.5 2.5 6   2.5 6   2.5 9.5 2.5 8   6   2.5 8   2.5 2.5 9.5
##  [18] 2.5 9.5 2.5 2.5 2.5 6   2.5 2.5 6   9.5 2.5 2.5 2.5 8   6   9.5 2.5
##  [35] 2.5 8   2.5 9.5 2.5 8   9.5 2.5 9.5 9.5 6   9.5 9.5 9.5 6   2.5 2.5
##  [52] 2.5 2.5 2.5 2.5 6   6   2.5 8   2.5 8   2.5 9.5 2.5 8   2.5 2.5 2.5
##  [69] 2.5 2.5 2.5 2.5 9.5 9.5 2.5 2.5 2.5 6   9.5 2.5 9.5 2.5 2.5 2.5 6
##  [86] 2.5 8   2.5 2.5 8   8   2.5 2.5 8   2.5 6   6   2.5 8   9.5
## Levels: 2.5 6 8 9.5
```

```r
#Definimos notas3 como el resultado de la codificación en intervalos utilizando como
#etiquetas la posición ordenada del intervalo (1, 2, 3 o 4)
notas3 = cut(notas, breaks = L, labels = FALSE, right = FALSE, include.lowest = TRUE)
notas3
```

```
##   [1] 3 4 1 1 2 1 2 1 4 1 3 2 1 3 1 1 4 1 4 1 1 1 2 1 1 2 4 1 1 1 3 2 4 1 1
##  [36] 3 1 4 1 3 4 1 4 4 2 4 4 4 2 1 1 1 1 1 1 2 2 1 3 1 3 1 4 1 3 1 1 1 1 1
##  [71] 1 1 4 4 1 1 1 2 4 1 4 1 1 1 2 1 3 1 1 3 3 1 1 3 1 2 2 1 3 4
```

```r
#Definimos notas4 como el resultado de la codificación en intervalos utilizando como
#etiquetas Susp, Aprob, Not y Exc
notas4 = cut(notas, breaks = L, labels = c("Susp", "Aprob", "Not", "Exc"), right = FALSE, include.lowes
notas4
```

```
##   [1] Not   Exc   Susp  Susp  Aprob Susp  Aprob Susp  Exc   Susp  Not
##  [12] Aprob Susp  Not   Susp  Susp  Exc   Susp  Exc   Susp  Susp  Susp
##  [23] Aprob Susp  Susp  Aprob Exc   Susp  Susp  Susp  Not   Aprob Exc
##  [34] Susp  Susp  Not   Susp  Exc   Susp  Not   Exc   Susp  Exc   Exc
##  [45] Aprob Exc   Exc   Exc   Aprob Susp  Susp  Susp  Susp  Susp  Susp
##  [56] Aprob Aprob Susp  Not   Susp  Not   Susp  Exc   Susp  Not   Susp
##  [67] Susp  Susp  Susp  Susp  Susp  Susp  Exc   Exc   Susp  Susp  Susp
##  [78] Aprob Exc   Susp  Exc   Susp  Susp  Susp  Aprob Susp  Not   Susp
##  [89] Susp  Not   Not   Susp  Susp  Not   Susp  Aprob Aprob Susp  Not
## [100] Exc
## Levels: Susp Aprob Not Exc
```

El resultado de `cut` ha sido, en cada caso, una lista con los elementos del vector original codificados con las etiquetas de las clases a las que pertenecen.

Las dos primeras aplicaciones de la función `cut` han producido factores (cuyos niveles son los intervalos y las marcas de clase, respectivamente, en ambos casos ordenados de manera natural), mientras que aplicándole `labels = FALSE` hemos obtenido un vector.

¿Qué habría ocurrido si le hubiéramos pedido a R que cortase los datos en 4 intervalos?

Pues en este caso no nos hubiera servido de mucho, sobre todo porque la amplitud de nuestros intervalos era, desde buen inicio, diferente.

```r
cut(notas, breaks = 4, right = FALSE, include.lowest = TRUE)
```

```
##   [1] [5,7.5)     [7.5,10]    [-0.01,2.5) [-0.01,2.5) [5,7.5)
##   [6] [-0.01,2.5) [5,7.5)     [2.5,5)     [7.5,10]    [-0.01,2.5)
##  [11] [5,7.5)     [5,7.5)     [-0.01,2.5) [5,7.5)     [-0.01,2.5)
##  [16] [2.5,5)     [7.5,10]    [-0.01,2.5) [7.5,10]    [2.5,5)
```

```
##  [21] [-0.01,2.5) [2.5,5)     [5,7.5)     [2.5,5)     [-0.01,2.5)
##  [26] [5,7.5)     [7.5,10]    [2.5,5)     [2.5,5)     [-0.01,2.5)
##  [31] [5,7.5)     [5,7.5)     [7.5,10]    [2.5,5)     [2.5,5)
##  [36] [7.5,10]    [-0.01,2.5) [7.5,10]    [2.5,5)     [5,7.5)
##  [41] [7.5,10]    [-0.01,2.5) [7.5,10]    [7.5,10]    [5,7.5)
##  [46] [7.5,10]    [7.5,10]    [7.5,10]    [5,7.5)     [-0.01,2.5)
##  [51] [2.5,5)     [-0.01,2.5) [2.5,5)     [-0.01,2.5) [-0.01,2.5)
##  [56] [5,7.5)     [5,7.5)     [2.5,5)     [5,7.5)     [2.5,5)
##  [61] [5,7.5)     [2.5,5)     [7.5,10]    [-0.01,2.5) [5,7.5)
##  [66] [-0.01,2.5) [2.5,5)     [-0.01,2.5) [2.5,5)     [2.5,5)
##  [71] [-0.01,2.5) [2.5,5)     [7.5,10]    [7.5,10]    [-0.01,2.5)
##  [76] [2.5,5)     [-0.01,2.5) [5,7.5)     [7.5,10]    [-0.01,2.5)
##  [81] [7.5,10]    [-0.01,2.5) [-0.01,2.5) [-0.01,2.5) [5,7.5)
##  [86] [2.5,5)     [7.5,10]    [-0.01,2.5) [2.5,5)     [5,7.5)
##  [91] [5,7.5)     [2.5,5)     [2.5,5)     [7.5,10]    [-0.01,2.5)
##  [96] [5,7.5)     [5,7.5)     [-0.01,2.5) [7.5,10]    [7.5,10]
## Levels: [-0.01,2.5) [2.5,5) [5,7.5) [7.5,10]
```

R ha repartido los datos en 4 intervalos de longitud 2.5, y ha desplazado ligeramente a la izquierda el extremo izquierdo del primer intervalo.

Trabajaremos ahora con `notas4` y calcularemos sus frecuencias:

```
table(notas4) #Fr. Abs
```

```
## notas4
##  Susp Aprob   Not   Exc
##    53    14    14    19
```

```
prop.table(table(notas4)) #Fr. Rel
```

```
## notas4
##  Susp Aprob   Not   Exc
##  0.53  0.14  0.14  0.19
```

```
cumsum(table(notas4)) #Fr. Abs. Cum
```

```
##  Susp Aprob   Not   Exc
##    53    67    81   100
```

```
cumsum(prop.table(table(notas4))) #Fr. Rel. Cum
```

```
##  Susp Aprob   Not   Exc
##  0.53  0.67  0.81  1.00
```

Podríamos haber obtenido todo lo anterior haciendo uso de la función `hist`.

```
notasHist = hist(notas, breaks = L, right = FALSE, include.lowest = TRUE, plot = FALSE)
FAbs = notasHist$count
FRel = prop.table(FAbs)
FAbsCum = cumsum(FAbs)
FRelCum = cumsum(FRel)
```

Ahora ya podemos crear un data frame con todas estas frecuencias:

```
intervalos = c("[0,5)","[5,7)","[7,9)","[9,10]")
calificacion = c("Suspenso", "Aprobado", "Notable", "Excelente")
marcas = notasHist$mids
tabla.Fr = data.frame(intervalos,calificacion,marcas,FAbs,FAbsCum,FRel,FRelCum)
tabla.Fr
```

```
##   intervalos calificacion marcas FAbs FAbsCum FRel FRelCum
## 1      [0,5)     Suspenso    2.5   53      53 0.53    0.53
## 2      [5,7)     Aprobado    6.0   14      67 0.14    0.67
## 3      [7,9)      Notable    8.0   14      81 0.14    0.81
## 4     [9,10]    Excelente    9.5   19     100 0.19    1.00
```

O bien, podríamos haber utilizado las funciones que os hemos proporcionado:

```
TablaFrecs.L(notas, L, TRUE)
```

```
##   intervals  mc Fr.abs Fr.cum.abs Fr.rel Fr.cum.rel
## 1     [0,5) 2.5     53         53   0.53       0.53
## 2     [5,7) 6.0     14         67   0.14       0.67
## 3     [7,9) 8.0     14         81   0.14       0.81
## 4    [9,10] 9.5     19        100   0.19       1.00
```

Estadísticos para datos agrupados

Estadísticos para datos agrupados

Al tener una muestra de datos numéricos, conviene calcular los *estadísticos* antes de realizar los agrupamientos, puesto que de lo contrario podemos perder información.

No obstante, hay situaciones en que los datos los obtenemos ya agrupados. En estos casos, aún sigue siendo posible calcular los estadísticos y utilizarlos como aproximaciones de los estadísticos de los datos "reales", los cuales no conocemos.

La media $\bar{x}$, la varianza, s^2, la varianza muestral, $\tilde{s}^2$, la desviación típica, s, y la desviación típica muestral, $\tilde{s}$ de un conjunto de datos agrupados se calculan mediante las mismas fórmulas que para los datos no agrupados con la única diferencia de que sustituimos cada clase por su marca de clase y la contamos con su frecuencia.

Es decir, si tenemos k clases, con sus respectivas marcas $X_1, \ldots, X_k$ con frecuencias absolutas $n_1, \ldots, n_k$ de forma que $n = \sum_{j=1}^{k} n_j$. Entonces

$$\bar{x} = \frac{\sum_{j=1}^{k} n_j X_j}{n}, \quad s^2 = \frac{\sum_{j=1}^{k} n_j X_j^2}{n} - \bar{x}^2, \quad \tilde{s}^2 = \frac{n}{n-1} \cdot s^2$$

$$s = \sqrt{s^2}, \quad \tilde{s} = \sqrt{\tilde{s}^2}$$

Intervalo modal

En lo referente a la moda, esta se sustituye por el intervalo modal, que es la clase con mayor frecuencia (absoluta o relativa, tanto da).

En el caso en que un valor numérico fuera necesario, se tomaría su marca de clase.

Intervalo crítico para la mediana

Se conoce como intervalo crítico para la mediana, $[L_c, L_{c+1})$, al primer intervalo donde la frecuencia relativa acumulada sea mayor o igual que 0.5

Denotemos por n_c su frecuencia absoluta, por $A_c = L_{c+1} - L_c$ su amplitud y por N_{c-1} la frecuencia acumalada del intervalo inmediantamente anterior (en caso de ser $[L_c, L_{c+1}) = [L_1, L_2)$, entonces $N_{c-1} = 0$). Entonces, M será una aproximación para la mediana de los datos "reales" a partir de los agrupados

$$M = L_c + A_c \cdot \frac{\frac{n}{2} - N_{c-1}}{n_c}$$

Aproximación de los cuantiles

La fórmula anterior nos permite aproximar el cuantil Q_p de los datos "reales" a partir de los datos agrupados:

$$Q_p = L_p + A_p \cdot \frac{p \cdot n - N_{p-1}}{n_p}$$

donde el intervalo $[L_p, L_{p+1})$ denota el primer intervalo cuya frecuencia relativa acumulada es mayor o igual a p

Vamos a seguir trabajando con nuestra variable `cw` y, esta vez, lo que haremos será calcular los estadísticos de la variable con los datos agrupados y, para acabar, estimaremos la mediana y algunos cuantiles.

Solución

Recordemos todo lo que habíamos obtenido sobre nuestra variable `cw`:

```
##  [1] 20.95 22.25 23.55 24.85 26.15 27.45 28.75 30.05 31.35 32.65 33.95
```

```
##        intervals   mc Fr.abs Fr.cum.abs Fr.rel Fr.cum.rel
## 1  [20.95,22.25) 21.6      2          2 0.0116     0.0116
## 2  [22.25,23.55) 22.9     14         16 0.0809     0.0925
## 3  [23.55,24.85) 24.2     27         43 0.1561     0.2486
## 4  [24.85,26.15) 25.5     44         87 0.2543     0.5029
## 5  [26.15,27.45) 26.8     34        121 0.1965     0.6994
## 6  [27.45,28.75) 28.1     31        152 0.1792     0.8786
## 7  [28.75,30.05) 29.4     15        167 0.0867     0.9653
## 8  [30.05,31.35) 30.7      3        170 0.0173     0.9826
## 9  [31.35,32.65) 32.0      2        172 0.0116     0.9942
## 10 [32.65,33.95) 33.3      1        173 0.0058     1.0000
```

Ahora ya podemos calcular los estadísticos:

```
TOT = tabla$Fr.cum.abs[10] #Como el total de todas
#las observaciones está en la fila 10, accedemos al valor 10
TOT
```

```
## [1] 173
```

```
anchura.media = round(sum(tabla$Fr.abs*tabla$mc)/TOT,3)
anchura.media #Media
```

```
## [1] 26.312
```

```
anchura.var = round(sum(tabla$Fr.abs*tabla$mc^2)/TOT-anchura.media^2,3)
anchura.var #Varianza
```

```
## [1] 4.476
```

```
anchura.dt = round(sqrt(anchura.var),3)
anchura.dt #Desviación típica
```

```
## [1] 2.116
```

```
I.modal = tabla$intervals[which(tabla$Fr.abs == max(tabla$Fr.abs))]
I.modal #Intervalo modal
```

```
## [1] [24.85,26.15)
## 10 Levels: [20.95,22.25) [22.25,23.55) [23.55,24.85) ... [32.65,33.95)
```

Por lo tanto, con los datos de los que disponemos, podemos afirmar que la anchura media de los cangrejos de la muestra es de 26.312mm, con una desviación típica de unos 4.476mm, y que el grupo de anchuras más numeroso era el de [24.85,26.15).

Pasemos ahora a calcular el intervalo crítico para la mediana.

```
I.critic = tabla$intervals[which(tabla$Fr.cum.rel >= 0.5)]
I.critic[1] #Intervalo critic
```

```
## [1] [24.85,26.15)
## 10 Levels: [20.95,22.25) [22.25,23.55) [23.55,24.85) ... [32.65,33.95)
```

```
#Como which regresa todos los intervalos que cumplan esa condición,
#solo necesitaremos acceder al primero.
```

Ahora, ya podemos calcular una estimación de la mediana de los datos "reales".

```
n = TOT
Lc = L[4]
Lc.pos = L[5]
Ac = L[5]-L[4]
Nc.ant = tabla$Fr.cum.abs[3]
nc = tabla$Fr.abs[4]
M = Lc+Ac*((n/2)-Nc.ant)/nc
M #Aproximación de la mediana de los datos "reales"
```

```
## [1] 26.13523
```

```
median(cw) #Mediana de los datos "reales"
```

```
## [1] 26.1
```

También podemos hacer aproximaciones de los cuantiles. Hemos creado una función `aprox.quantile.p` para no tener que copiar la operación cada vez que queramos calcular un cuantil aproximado.

```
aprox.quantile.p = function(Lcrit,Acrit,n,p,Ncrit.ant,ncrit){
  round(Lcrit+Acrit*(p*n-Ncrit.ant)/ncrit,3)
}
aprox.quantile.p(Lc,Ac,n,0.25,Nc.ant,nc) #Primer cuartil
```

```
## [1] 24.857
```

```
aprox.quantile.p(Lc,Ac,n,0.75,Nc.ant,nc) #Tercer cuartil
```

```
## [1] 27.413
```

Y ahora, calculemos los cuartiles de los datos "reales"

```
quantile(cw,0.25)
```

```
##  25%
## 24.9
```

```
quantile(cw,0.75)
```

```
##  75%
## 27.7
```

Repetir este ejemplo para la muestra de notas del Ejemplo 3.

Histogramas

La mejor manera de representar datos agrupados es mediante unos diagramas de barras especiales conocidos como **histogramas.**

En ellos se dibuja sobre cada clase una barra cuya área representa su frecuencia. Podéis comprobar que el producto de la base por la altura de cada barra es igual a la frecuencia de la clase correspondiente.

El uso de histogramas

Si todas las clases tienen la misma amplitud, las alturas de estas barras son proporcionales a las frecuencias de sus clases, con lo cual podemos marcar sin ningún problema las frecuencias sobre el eje vertical. Pero si las amplitudes de las clases no son iguales, las alturas de las barras en un histograma no representan correctamente las frecuencias de las clases.

En este último caso, las alturas de las barras son las necesarias para que el área de cada barra sea igual a la frecuencia de la clase correspondiente y como las bases son de amplitudes diferentes, estas alturas no son proporcionales a las frecuencias de las clases, por lo que no tiene sentido marcar las frecuencias en el eje vertical

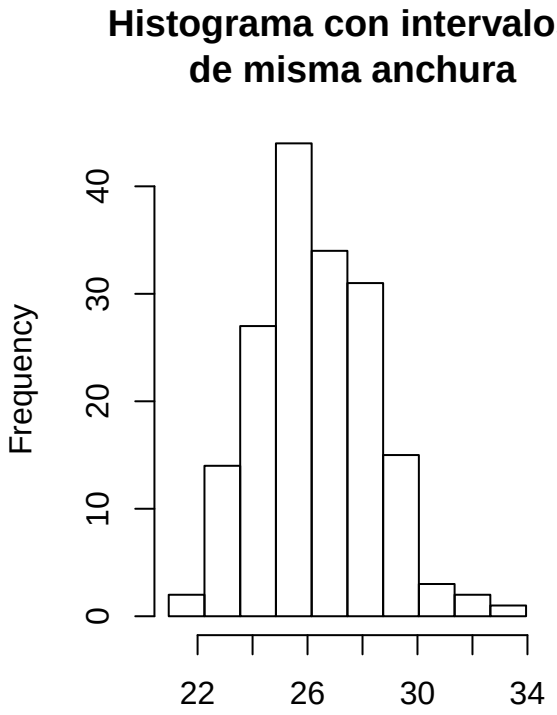
El uso de histogramas

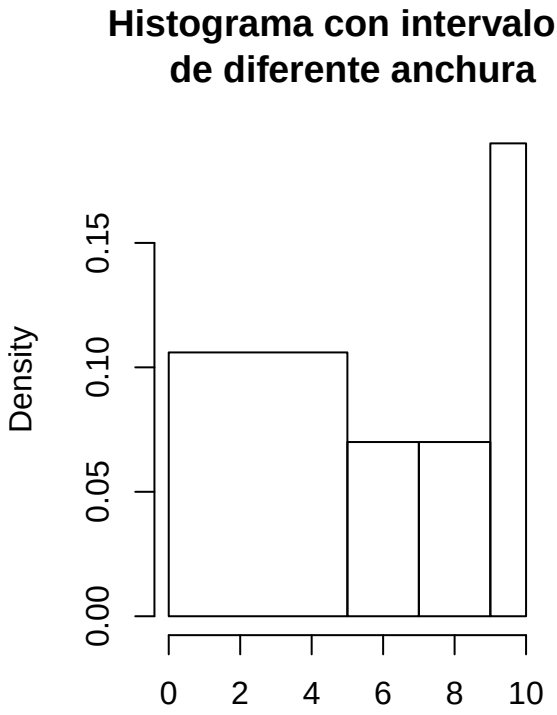
Los histogramas también son utilizados para representar frecuencias acumuladas de datos agrupados. En este caso, las alturas representan las frecuencias independientemente de la base debido a que éstas deben ir creciendo.

Interpretación de los histogramas

El eje de las abcisas representa los datos. Aquí marcamos los extremos de las clases y se dibuja una barra sobre cada una de ellas. Esta barra tiene significados diferentes en función del tipo de histograma, pero en general representa la frecuencia de su clase

- Histograma de frecuencias absolutas: la altura de cada barra es la necesaria para que el área de la barra sea igual a la frecuencia absoluta de la clase. Las amplitudes de las clases pueden ser todas iguales o no. En el primer caso, las alturas son proporcionales a las frecuencias. En el segundo caso, no existe tal proporcionalidad. De todas formas, sea cual sea el caso, conviene indicar de alguna forma la frecuencia que representa cada barra.





- Histograma de frecuencias relativas: la altura, **densidad**, de cada barra es la necesaria para que el área sea igual a la frecuencia relativa de la clase. La suma de todas las áreas debe ser 1. De nuevo, conviene indicar de alguna forma la frecuencia que representa cada barra.
- Histogramas de frecuencias acumuladas: las alturas de las barras son iguales a las frecuencias acumuladas de las clases, independientemente de su amplitud.

Frecuencias nulas

No es conveniente que en un histograma aparezcan clases con frecuencia nula, exceptuando el caso en que represente poblaciones muy diferentes y separadas sin individuos intermedios.

Si apareciesen clases vacías, convendría utilizar un número menor de clases, o bien unir las clases vacías con alguna de sus adyacentes. De este último modo romperíamos nuestro modo de trabajar con clases de la misma amplitud.

Dibujando histogramas con R

Lo hacemos con la función `hist`, la cual ya conocemos. Su sintaxis es

```
hist(x, breaks=..., freq=..., right=..., ...)
```

- `x`: vector de los datos
- `breaks`: vector con los extremos de los intervalos o el número k de intervalos. Incluso podemos indicar, entre comillas, el método que deseemos para calcular el número de clases: `"Scott"`, `"Sturges"`... Eso sí, para cualquiera de las dos últimas opciones, no siempre obtendréis el número deseado de intervalos, puesto que R lo considerará solo como sugerencia. Además, recordad que el método para calcular los intervalos es diferente al de la función `cut`. Por tanto, se recomienda hacer uso de la primera opción.

- `freq=TRUE`, que es su valor por defecto, produce el histograma de frecuencias absolutas si los intervalos son todos de la misma amplitud y de frecuencias relativas en caso contrario. `freq=FALSE` nos produce siempre el de frecuencias relativas.
- `right` funciona exactamente igual que en la función `cut`.
- `include.lowest = TRUE` también funciona exactamente igual que en la función `cut`.
- También podéis utilizar los parámetros de la función `plot` que tengan sentido

`hist` titula por defecto los histogramas del siguiente modo: "Histogram of" seguido del nombre del vector de datos. No suele quedar muy bien si no estamos haciendo nuestro análisis en inglés.

Recordemos que el parámetro `plot` igualado a `FALSE` no dibujaba, pero sí calculaba el histograma.

La función `hist` contiene mucha información en su estructura interna

- `breaks` contiene el vector de extremos de los intervalos: $L_1, \ldots, L_{k+1}$
- `mids` contiene los puntos medios de los intervalos, lo que nosotros consideramos las marcas de clase: $X_1, \ldots, X_k$
- `counts` contiene el vector de frecuencias absolutas de los intervalos: $n_1, \ldots, n_k$
- `density` contiene el vector de las densidades de los intervalos. Estas se corresponden con las alturas de las barras del histograma de frecuencias relativas. Recordemos, la densidad de un intervalo es su frecuencia relativa divida por su amplitud.

Aquí os dejamos una función útil para calcular histogramas de frecuencias absolutas más completos:

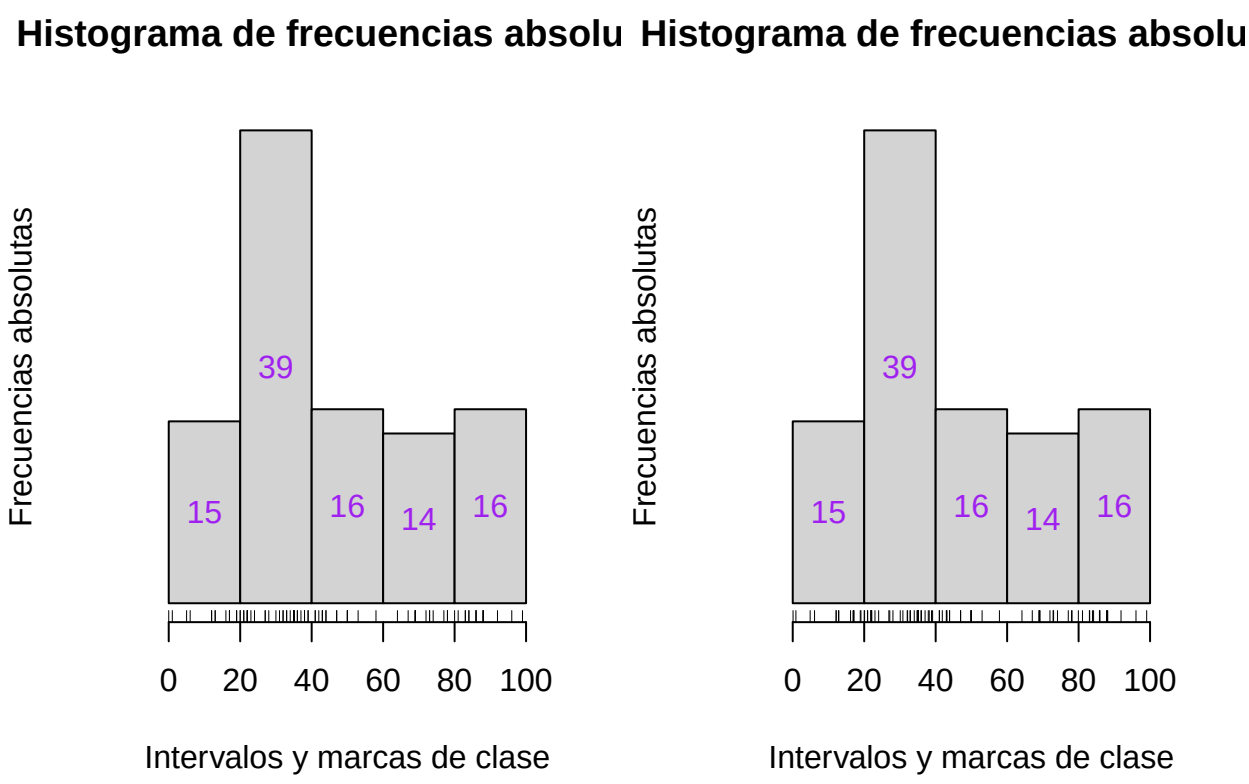
```
histAbs = function(x,L) {
  h = hist(x, breaks = L, right = FALSE, freq = FALSE,
           xaxt = "n", yaxt = "n", col = "lightgray",
           main = "Histograma de frecuencias absolutas",
           xlab = "Intervalos y marcas de clase",ylab = "Frecuencias absolutas")
  axis(1, at=L)
  text(h$mids, h$density/2, labels=h$counts, col="purple")
  }
```

- `xaxt="n"` e `yaxt="n"` especifican que, por ahora, la función no dibuje los ejes de abcisas y ordenadas, respectivamente.
- `axis(i, at=...)` dibuja el eje correspondiente al valor de i con marcas en los lugares indicados por el vector definido mediante `at`. Si $i = 1$, el de abcisas; si $i = 2$, el de ordenadas.

Os habréis fijado que con `freq = FALSE` en realidad hemos dibujado un histograma de frecuencias relativas, pero al haber omitido el eje de ordenadas, da lo mismo. En cambio, sí que nos ha sido útil para poder añadir, con la función `text`, la frecuencia absoluta de cada clase sobre el punto medio de su intervalo, los valores `h$mids` y a media algura de su barra, correspondiente a `h$density` gracias a que, con `freq = FALSE` estas alturas se corresponden con la densidad.

Otra forma de indicar las frecuencias absolutas de las barras es utilizar la función `rug`, la cual permite añadir al histograma una "alfombra" con marcas en todos los valores del vector, donde el grosor de cada marca es proporcional a la frecuencia del valor que representa.

Existe la posibilidad de añadir un poco de ruido a los datos de un vector para deshacer posibles empates. Esto lo conseguimos combinando la función `rug` con `jitter`.



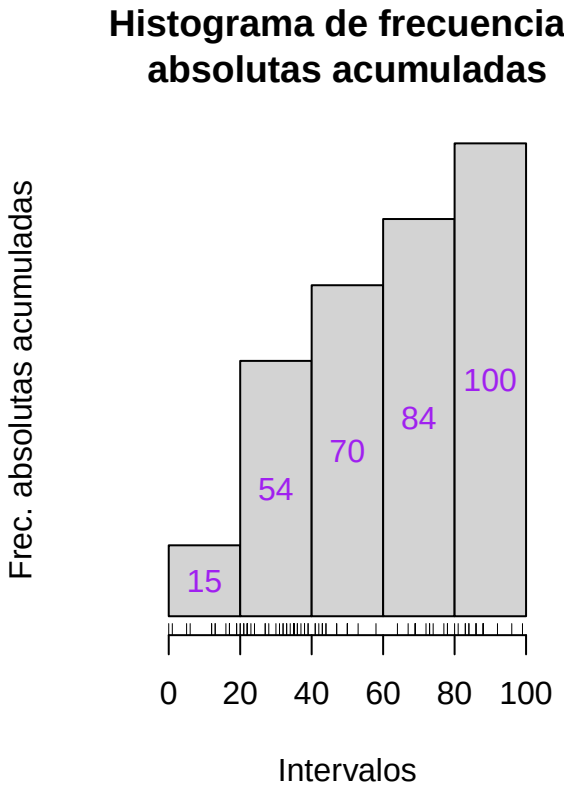
Aquí os dejamos una función útil para calcular histogramas de frecuencias absolutas acumuladas más completos:

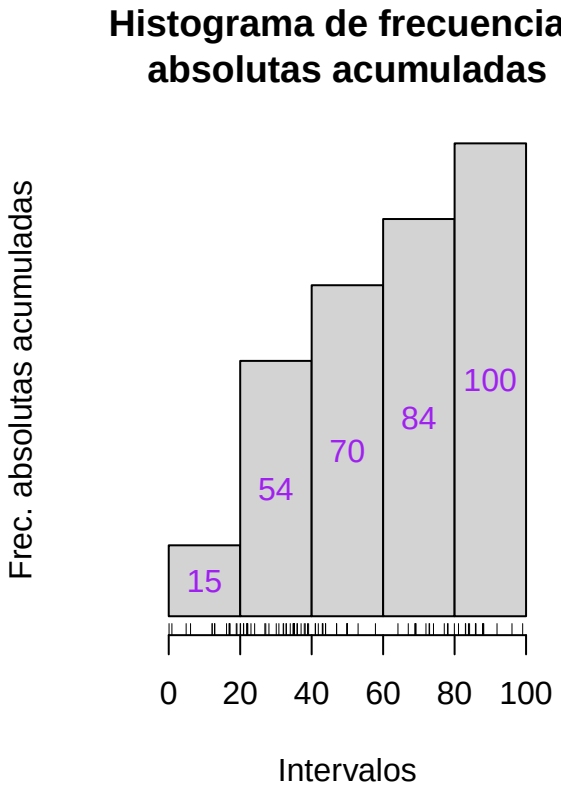
```
histAbsCum = function(x,L) {
  h = hist(x, breaks = L, right = FALSE , plot = FALSE)
  h$density = cumsum(h$density)
  plot(h, freq = FALSE, xaxt = "n", yaxt = "n", col = "lightgray",
       main = "Histograma de frecuencias\nabsolutas acumuladas", xlab = "Intervalos",
       ylab = "Frec. absolutas acumuladas")
  axis(1, at=L)
  text(h$mids, h$density/2, labels = cumsum(h$counts), col = "purple")
  }
```

Con la función anterior, lo que hacemos es, en primer lugar, producir el histograma básico de los datos, sin dibujarlo para a continuación modificar la componente `density` para que contenga las sumas acumuladas de esta componente del histograma original.

Seguidamente, dibujamos el nuevo histograma resultante, aplicando la función `plot`. Es aquí donde debemos especificar los parámetros y no en el histograma original.

Finalmente, añadimos el eje de abcisas y las frecuencias acumuladas en color lila.





Histogramas de frecuencias relativas

En estos histogramas, es común superponer una curva que estime la densidad de la distribución de la variable cuantitativa definida por la característica que estamos midiendo.

La densidad de una variable es una curva cuya área comprendida entre el eje de las abcisas y la propia curva sobre un intervalo es igual a la fracción de individuos de la población que caen dentro de ese intervalo.

Para hacernos una idea visual, imaginad que vais aumentando el tamaño de la muestra a la vez que agrupáis los datos en un conjunto cada vez mayor de clases. Si el rango de los datos se mantiene constante, la amplitud de las clases del histograma irá menguando. Además, cuando n, el tamaño de la muestra, tiende a infinito, los intervalos tienden a ser puntos y, a su vez, las barras tienden a ser líneas verticales. Pues bien, los extremos superiores de estas líneas serán los que dibujen la densidad de la variable.

Campana de Gauss

Es la densidad más famosa: la Campana de Gauss**<- Oprime aquí** Ésta se corresponde con una variable que siga una distribución nomal.

La forma de la campana depende de dos parámetros: el valor medio, μ, y su desviación típica, σ.

Dibujando la curva de densidad

Existen muchos métodos con los cuales estimar la densidad de distribución a partir de una muestra.

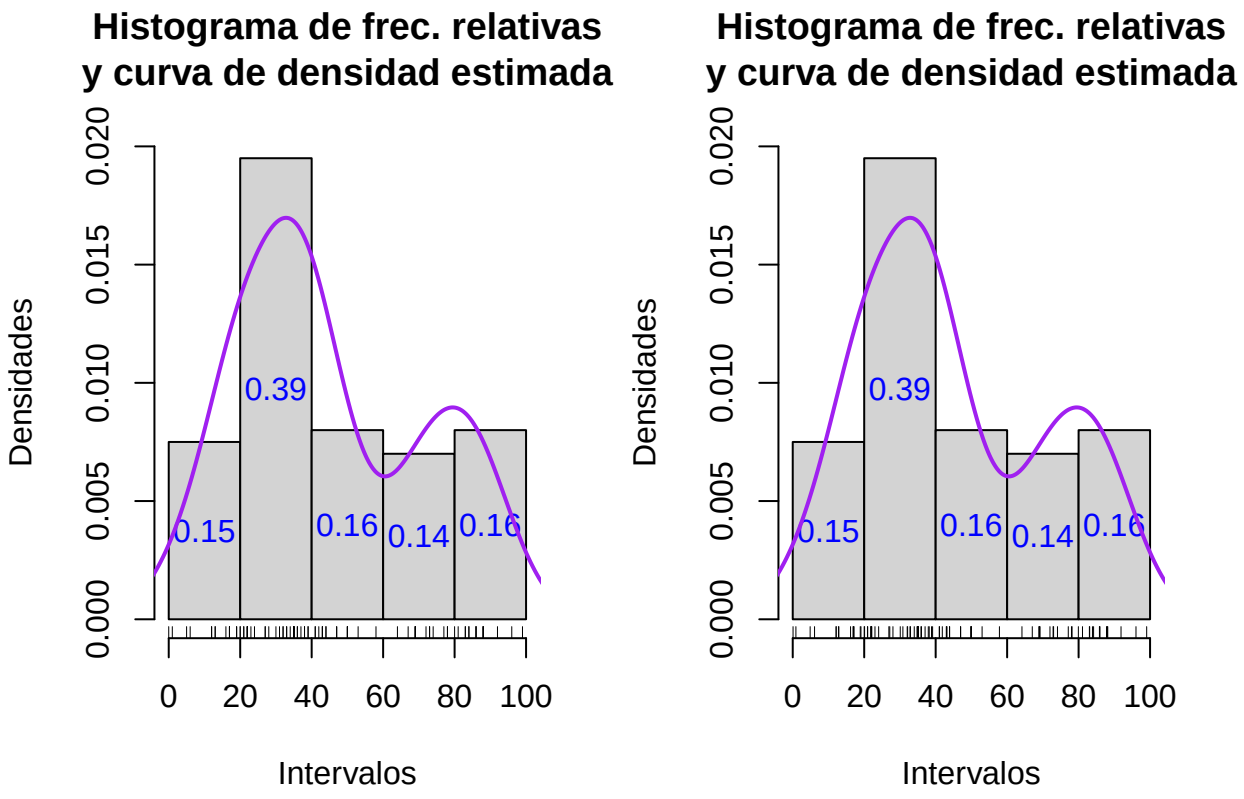
Una de ellas es mediante la función `density` de R. Al aplicarla a un conjunto de datos, produce una `list` que incluye los vectores `x` e `y` que continen la primera y segunda coordenadas, respectivamente, de 512 puntos de la forma (x, y) sobre la curva de densidad estimada.

Aplicando `plot` o `lines` a este resultado según pertoque, obtenemos la representación gráfica de esta curva.

Histogramas de frecuencias relativas

Aquí os dejamos una función útil para calcular histogramas de frecuencias relativas más completos:

```
histRel = function(x,L) {
  h = hist(x, breaks=L, right=FALSE , plot=FALSE)
  t = round(1.1*max(max(density(x)[[2]]),h$density),2)
  plot(h, freq = FALSE, col = "lightgray",
       main = "Histograma de frec. relativas\ny curva de densidad estimada",
       xaxt="n", ylim=c(0,t), xlab="Intervalos", ylab="Densidades")
  axis(1, at = L)
  text(h$mids, h$density/2, labels = round(h$counts/length(x),2), col = "blue")
  lines(density(x), col = "purple", lwd = 2)
  }
```



Histogramas de frecuencias relativas acumuladas

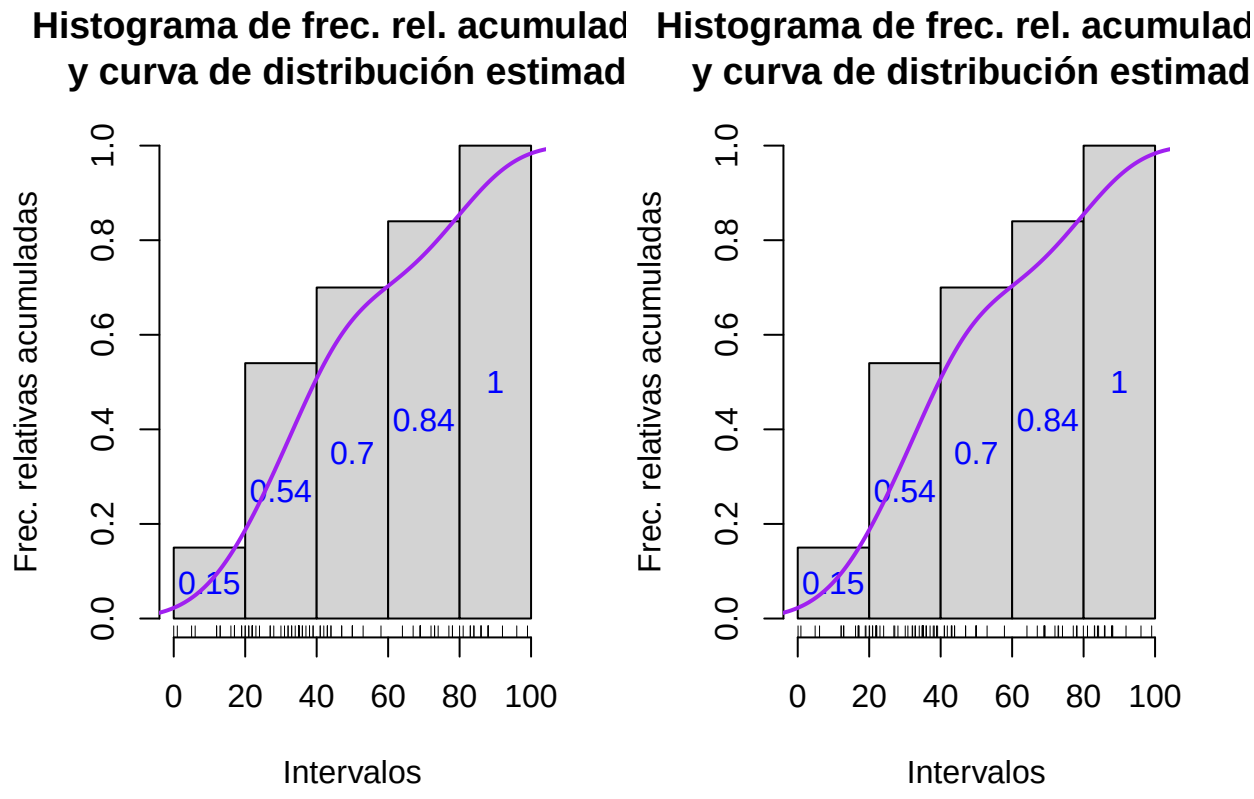
En este último tipo de histograma, se suele superponer una curva que estime la función de distribución de la variable definida por la característica que estamos midiendo.

Esta función de distribución, en cada punto nos da la fracción de individuos de la población que caen a la izquierda de este punto: su frecuencia relativa acumulada.

En general, la función de distribución en un valor determinado se obtiene hallando el área de la función de densidad que hay a la izquierda del valor.

Aquí os dejamos una función útil para calcular histogramas de frecuencias relativas acumuladas más completos:

```
histRelCum = function(x,L){
  h = hist(x, breaks = L, right = FALSE , plot = FALSE)
  h$density = cumsum(h$counts)/length(x)
  plot(h, freq = FALSE,
       main = "Histograma de frec. rel. acumuladas\n y curva de distribución estimada",
       xaxt = "n", col = "lightgray", xlab = "Intervalos",
       ylab = "Frec. relativas acumuladas")
  axis(1, at = L)
  text(h$mids, h$density/2, labels = round(h$density ,2), col = "blue")
  dens.x = density(x)
  dens.x$y = cumsum(dens.x$y)*(dens.x$x[2]-dens.x$x[1])
  lines(dens.x,col = "purple",lwd = 2)
}
```

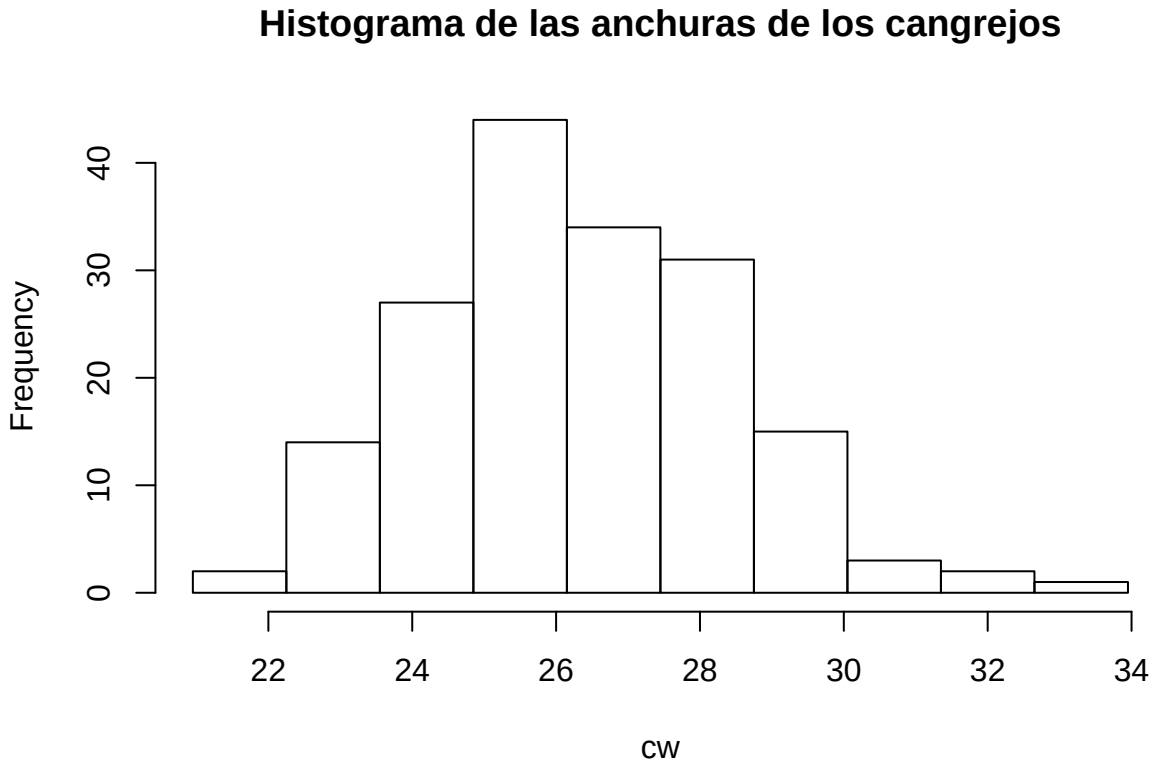


Vamos a seguir trabajando con nuestra variable `cw` y, esta vez, lo que haremos será calcular histogramas de todas las formas explicadas anteriormente.

Solución

Dibujamos el histograma con `hist` y luego observamos su información interna.

```r
hist(cw, breaks = L, right = FALSE, main = "Histograma de las anchuras de los cangrejos")
```

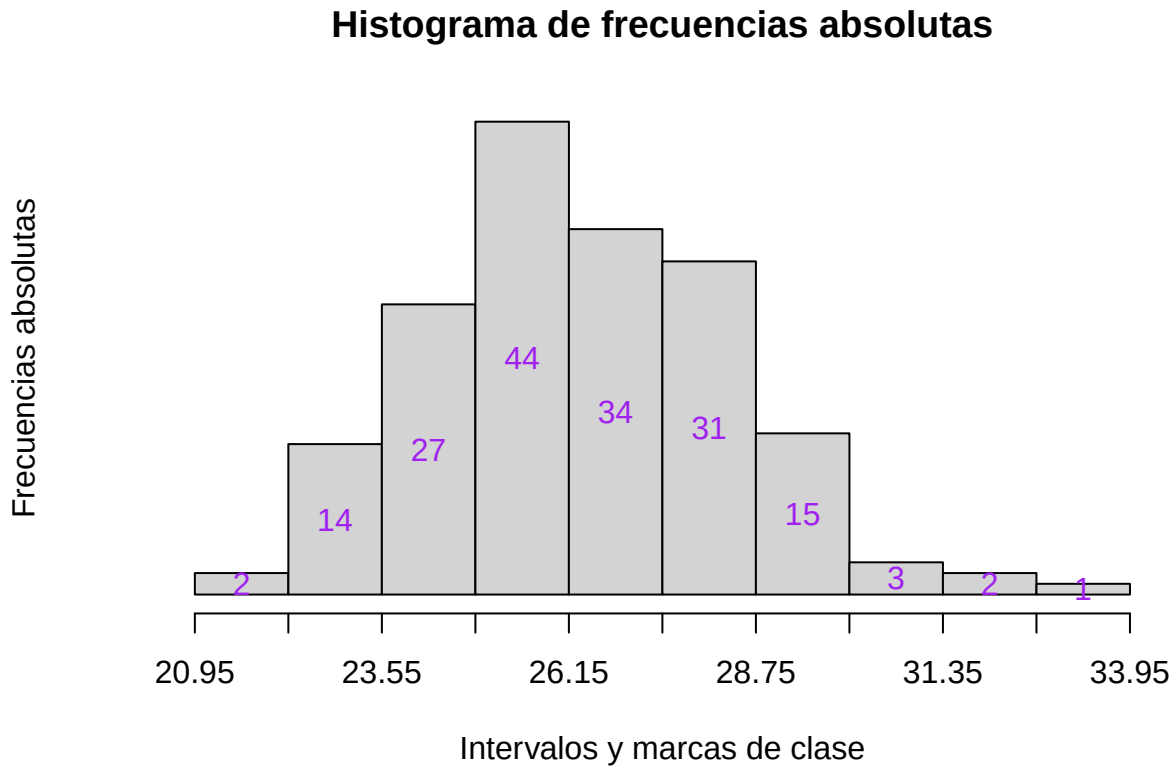



```r
hist(cw, breaks = L, right = FALSE, plot = FALSE)
```

```
## $breaks
##  [1] 20.95 22.25 23.55 24.85 26.15 27.45 28.75 30.05 31.35 32.65 33.95
##
## $counts
##  [1]  2 14 27 44 34 31 15  3  2  1
##
## $density
##  [1] 0.008892841 0.062249889 0.120053357 0.195642508 0.151178301
##  [6] 0.137839040 0.066696309 0.013339262 0.008892841 0.004446421
##
## $mids
##  [1] 21.6 22.9 24.2 25.5 26.8 28.1 29.4 30.7 32.0 33.3
##
## $xname
## [1] "cw"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

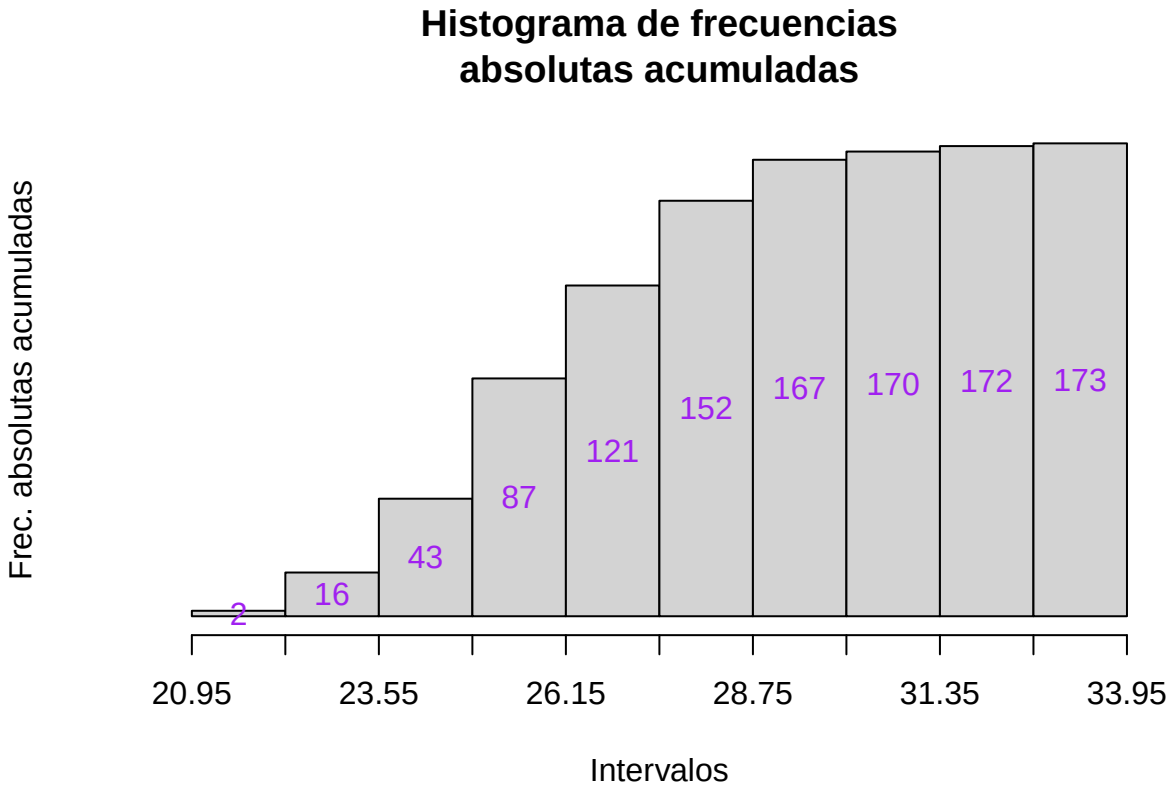
Dibujamos el histograma con `histAbs`.

```
histAbs(cw,L)
```



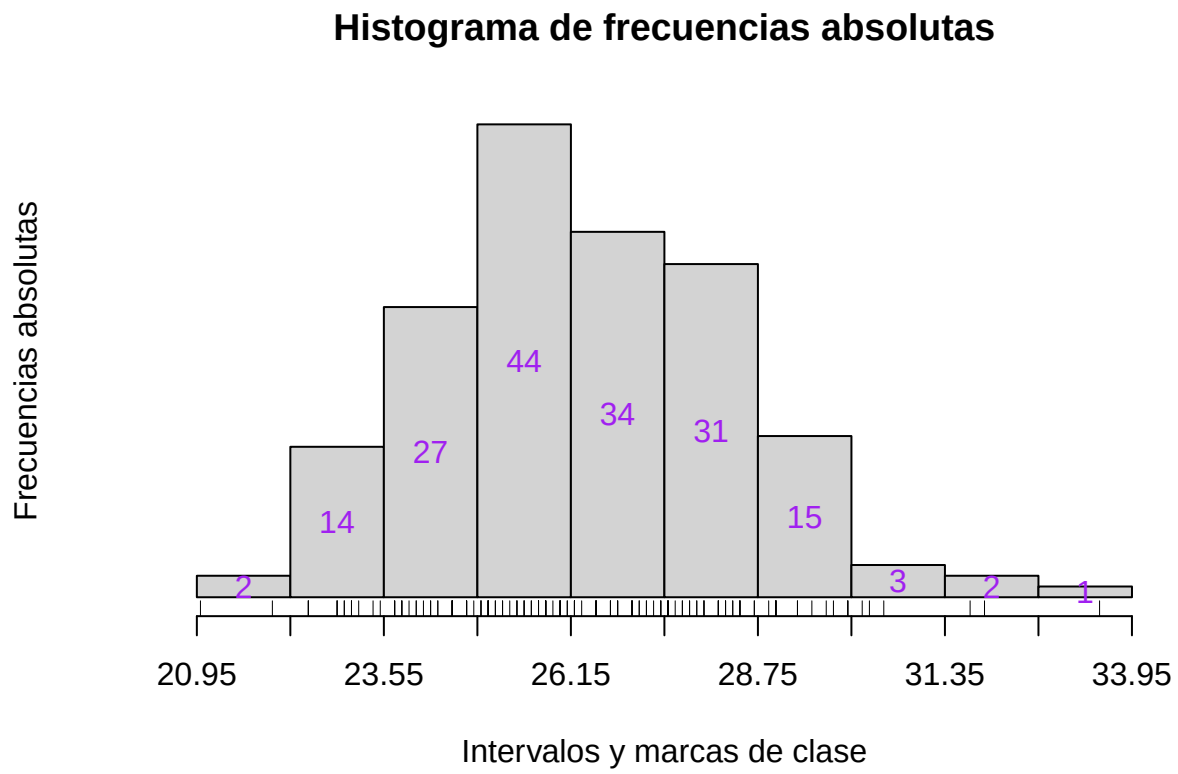
Dibujamos el histograma con `histAbsCum`.

```
histAbsCum(cw,L)
```

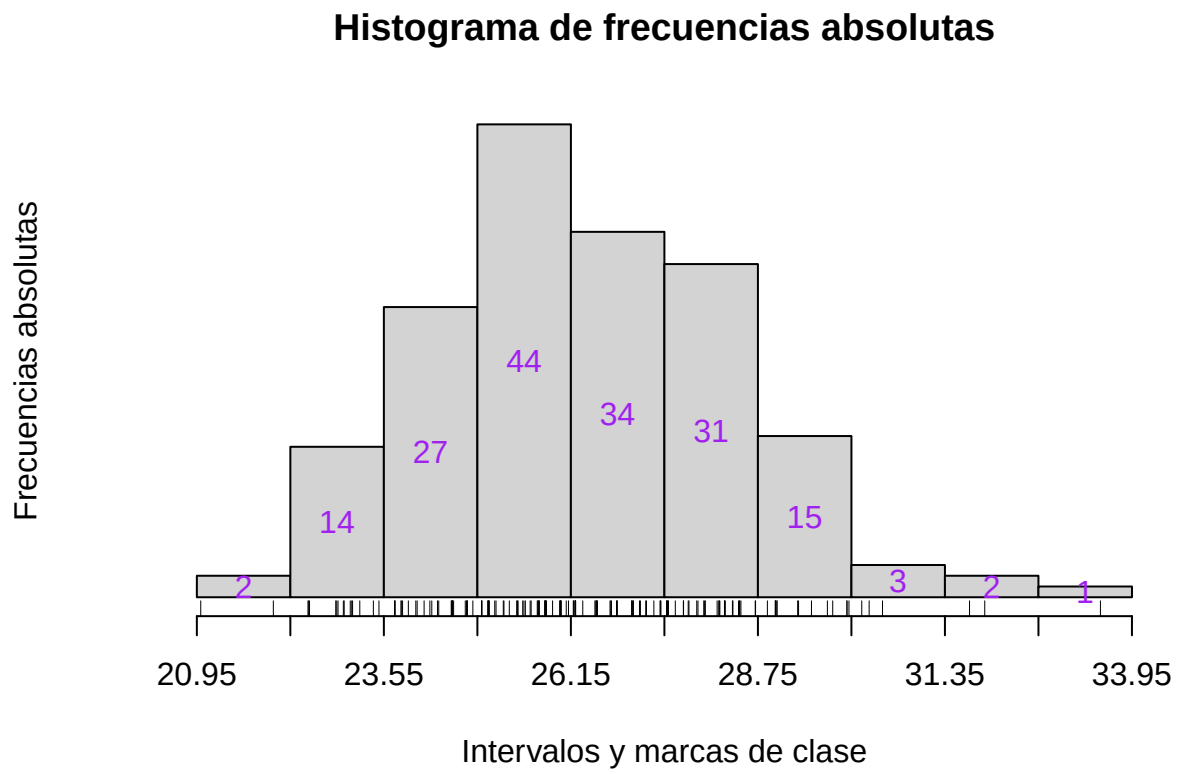


Hacemos uso de las funciones `rug` y `jitter`

```
histAbs(cw,L)
rug(cw)
```




```
histAbs(cw,L)
rug(jitter(cw))
```

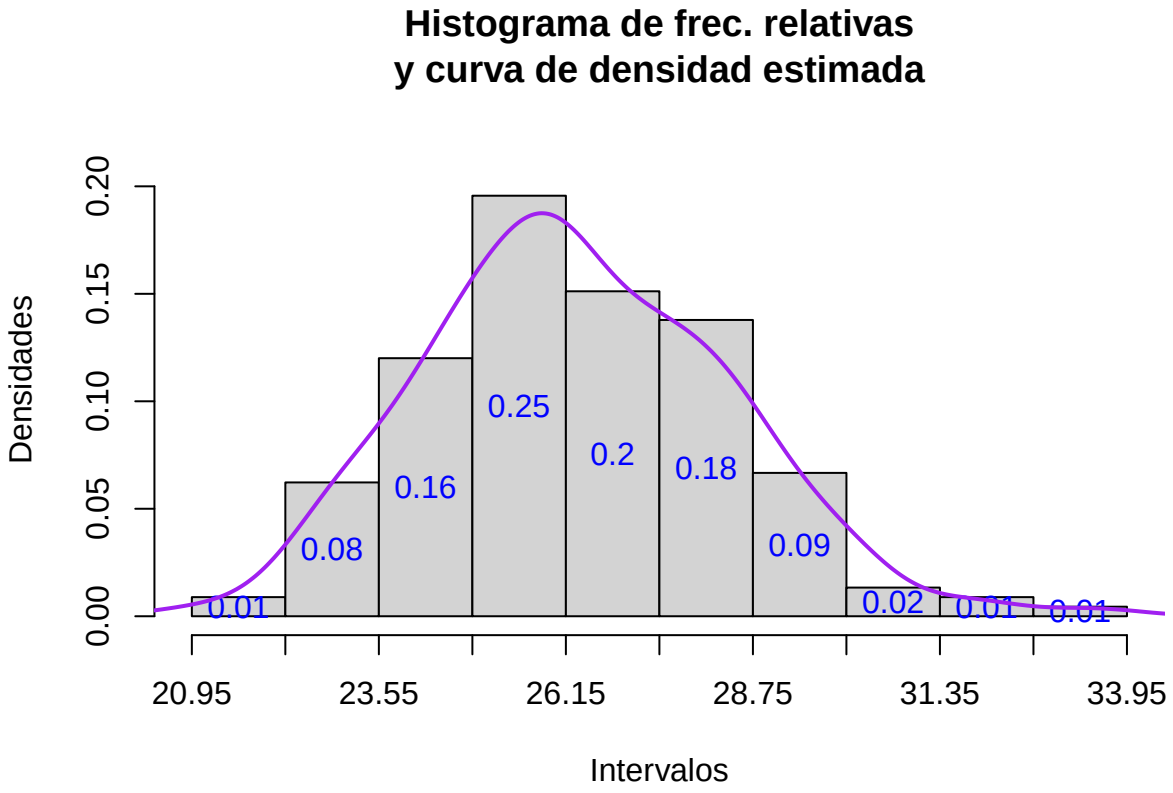


A continuación, calculamos la densidad de `cw` y la representamos con `histRel`

```
str(density(cw))
```

```
## List of 7
##  $ x        : num [1:512] 19 19 19.1 19.1 19.1 ...
##  $ y        : num [1:512] 3.90e-05 4.50e-05 5.17e-05 5.94e-05 6.82e-05 ...
##  $ bw       : num 0.671
##  $ n        : int 173
##  $ call     : language density.default(x = cw)
##  $ data.name: chr "cw"
##  $ has.na   : logi FALSE
##  - attr(*, "class")= chr "density"
```

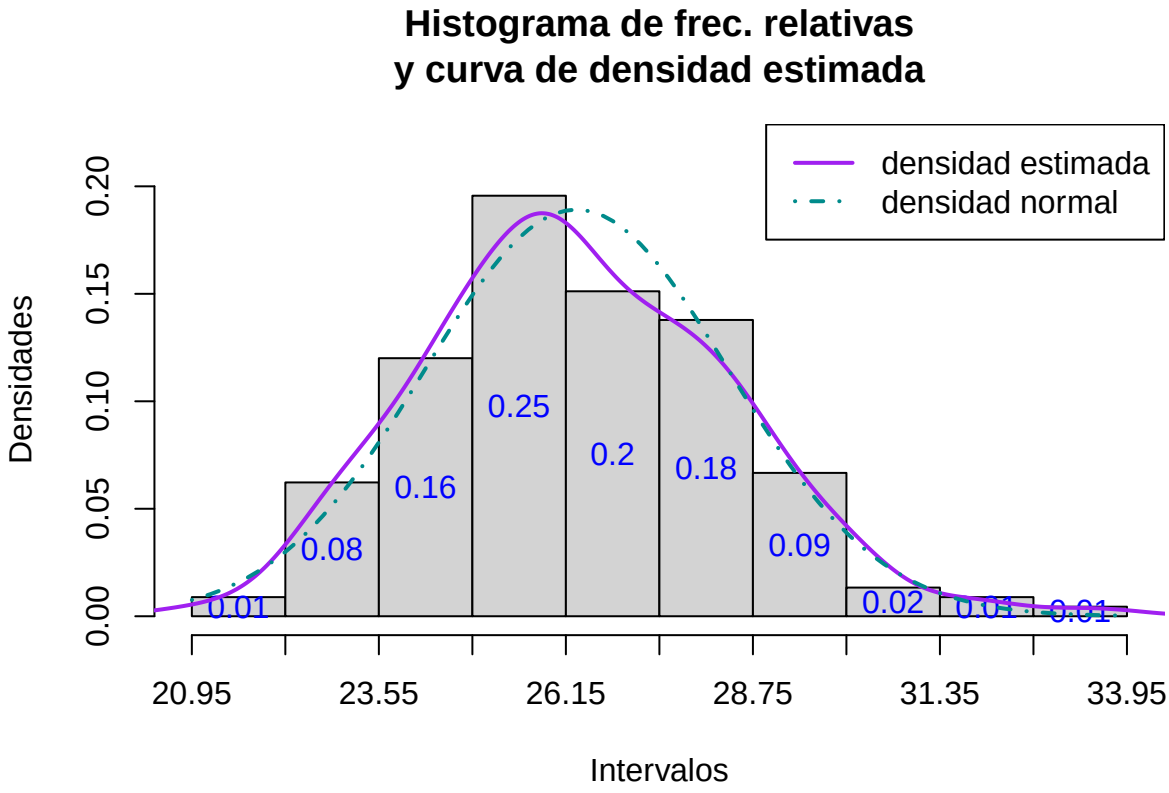
```
histRel(cw,L)
```



La curva de densidad que hemos obtenemos en este gráfico tiene una forma de campana que nos recuerda la campana de Gauss. Para explorar este parecido, vamos a añadir al histograma la gráfica de la función densidad de una distribución normal de media y desviación típica las del conjunto de datos original

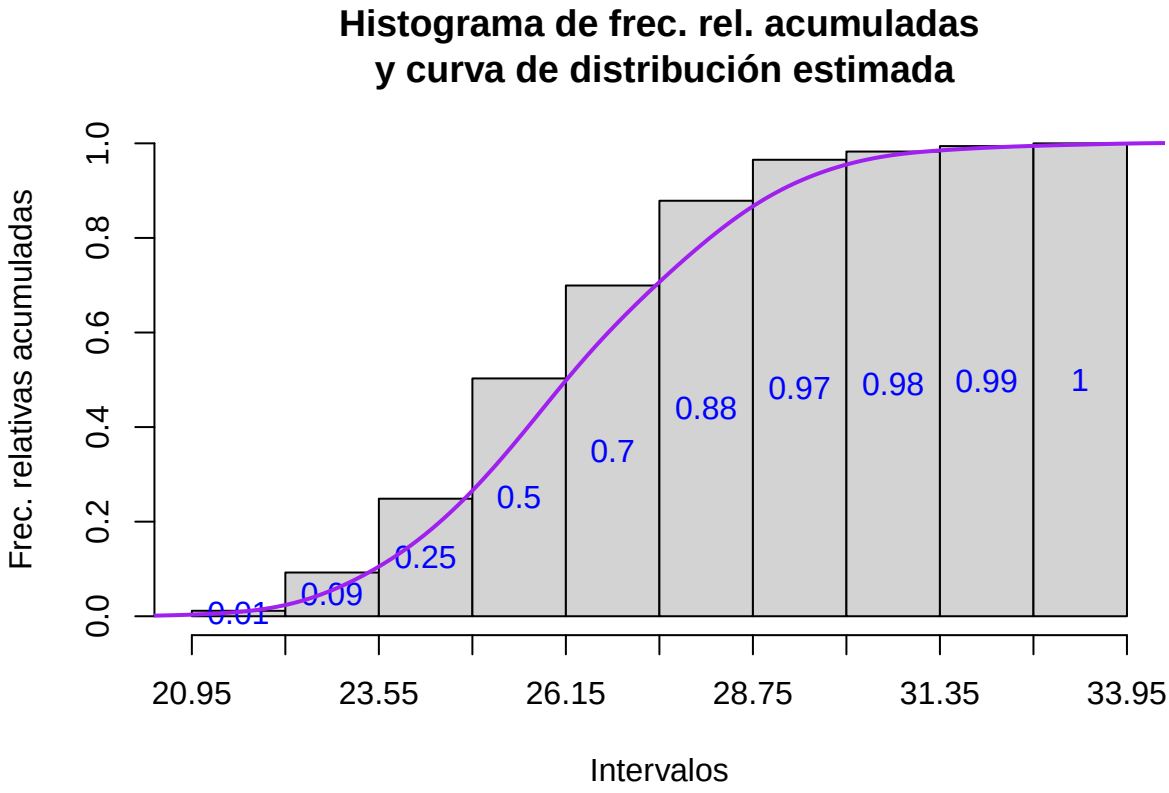
Así, aplicando las instrucciones siguientes, acabamos obteniendo

```
histRel(cw,L)
curve(dnorm(x, mean(cw), sd(cw)), col="cyan4", lty=4, lwd=2,
add=TRUE)
legend("topright", lwd=c(2,2), lty=c(1,4), col=c("purple","cyan4"),
       legend=c("densidad estimada","densidad normal"))
```



Dibujamos el histograma con `histRelCum`.

```
histRelCum(cw,L)
```


Histograma de frec. rel. acumuladas
y curva de distribución estimada
Frec. relativas acumuladas
0.0
0.2
0.4
0.6
0.8
1.0
0.01
0.09
0.25
0.5
0.7
0.88
0.97
0.98
0.99
1
20.95
23.55
26.15
28.75
31.35
33.95
Intervalos