Estadisticos para datos agrupados

Victor Lopez

2023-01-28

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.

Es lo mismo calcular los estadisticos que con datos no agrupados. La única diferencia es que sustituimos cada clase por su marca de clase y la contamos con su frecuencia.

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

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

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 critico (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

Aproximacion de la mediana real

Denotemos por n_c la frecuencia absoluta del intervalo critico, 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

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

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

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

Explicacion de lo que intenta hacer la formula: Lo que hace es una interpolacion entre el limite inferior y superior del intervalo donde caeria el cuantil teoricamente, se hace una regla de 3, es decir, una proporcion, de que tanta parte del intervalo hay que recorrer de forma lineal para encontrarnos con el cuantil presuponiendo un crecimiento lineal de los datos

Algo parecido ocurre con la de la mediana

Calculando los estadisticos en R

```
crabs = read.table("../../data/datacrab.txt", header = TRUE)
cw = cut(crabs$width,
    breaks =  c(20.95, 22.5, 23.55, 24.85, 26.15, 27.45, 28.75, 30.05, 31.35, 32.65, 33.95),
    right = FALSE)

TablaFrecs.L = function(x,L,V){
  x_cut = cut(x, breaks=L, right=FALSE, include.lowest=V, diag.lab = 5)
  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
}

tabla = TablaFrecs.L(crabs$width, c(20.95, 22.25, 23.55, 24.85, 26.15, 27.45, 28.75, 30.05, 31.35, 32.65
tabla
```

```
##      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  [28.8,30.1) 29.4     15        167 0.0867     0.9653
## 8  [30.1,31.4) 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
```

```r
n = tabla$Fr.cum.abs[length(tabla$Fr.cum.abs)]
n
```

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

```r
media = round(sum(tabla$Fr.abs * tabla$mc)/n, 3)
media
```

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

```r
varianza = round(sum(tabla$Fr.abs * tabla$mc^2)/n - media^2, 3)
varianza
```

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

```r
desvTipic = round(sqrt(varianza), 3)
desvTipic
```

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

```r
intervalo.modal = tabla$intervals[which(tabla$Fr.abs == max(tabla$Fr.abs))]
intervalo.modal
```

```
## [1] "[24.9,26.1)"
```

```r
intervalo.critico = tabla$intervals[which(tabla$Fr.cum.rel > 0.5)]
intervalo.critico[1]
```

```
## [1] "[24.9,26.1)"
```

```r
# which devuelve el indice de la fila donde se cumple la condicion.
```

```r
# Estimacion mediana datos reales

L = c(20.95, 22.25, 23.55, 24.85, 26.15, 27.45, 28.75, 30.05, 31.35, 32.65, 33.95)
n = n
Lc = L[4]
Lc.pos = L[5]
Ac = Lc.pos - Lc
Nc.ant = tabla$Fr.cum.abs[3]
nc = tabla$Fr.abs[4]
M = Lc + Ac *((n/2) - Nc.ant)/nc
M
```

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

```r
# Estimacion del cuantil aproximado

aprox.quantile.p = function(Lc, Ac, n, p, Nc.ant, nc){
  round(Lc + Ac * (p*n - Nc.ant)/nc, 3)
}

aprox.quantile.p(Lc, Ac, n, 0.25, Nc.ant, nc)
```

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

Ejercicio de calcular los estadisticos

```
set.seed(4)
notas = sample(0:10,100, replace = TRUE)
set.seed(NULL)
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  4  0
##  [26]  5 10  4  3  0  7  5 10  3  4  8  1  9  3  7  9  1  9 10  5 10 10  9  5  0
##  [51]  3  1  3  2  0  6  6  4  7  4  7  3  9  0  7  0  3  0  3  3  1  4 10  9  1
##  [76]  4  0  6 10  0 10  1  0  2  6  4  8  2  3  7  7  3  3  8  2  6  6  2  8  9
```

```
t = TablaFrecs.L(notas, c(0, 5, 7, 9, 10), TRUE)
t
```

```
##   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
```

```
n = t$Fr.cum.abs[length(t$Fr.cum.abs)]
n
```

```
## [1] 100
```

```
media = round(sum(t$Fr.abs * t$mc)/n, 3)
media
```

```
## [1] 5.09
```

```
mean(notas)
```

```
## [1] 4.72
```

```
varianza = round(sum(t$Fr.abs * t$mc^2)/n - media^2, 3)
varianza
```

```
## [1] 8.552
```

```
var(notas)
```

```
## [1] 10.44606
```

```r
desvTipic = round(sqrt(varianza), 3)
desvTipic
```

```
## [1] 2.924
```

```r
sd(notas)
```

```
## [1] 3.232037
```

```r
intervalo.modal = t$intervals[which(t$Fr.abs == max(t$Fr.abs)) ]
intervalo.modal
```

```
## [1] "[0,5)"
```

```r
intervalo.critico = t$intervals[which(t$Fr.cum.abs > 0.5)]
intervalo.critico[1]
```

```
## [1] "[0,5)"
```

```r
L = c(0, 5, 7, 9, 10)
Lc = L[1]
Lc.pos = L[2]
Ac = Lc.pos - Lc
nc = t$Fr.abs[1]
Nc.ant = 0

mediana.real = Lc + Ac * (n/2 - Nc.ant)/nc
mediana.real
```

```
## [1] 4.716981
```

```r
median(notas)
```

```
## [1] 4
```

```r
quartil = aprox.quantile.p(Lc, Ac, n, 0.25, Nc.ant, nc)
quartil
```

```
## [1] 2.358
```

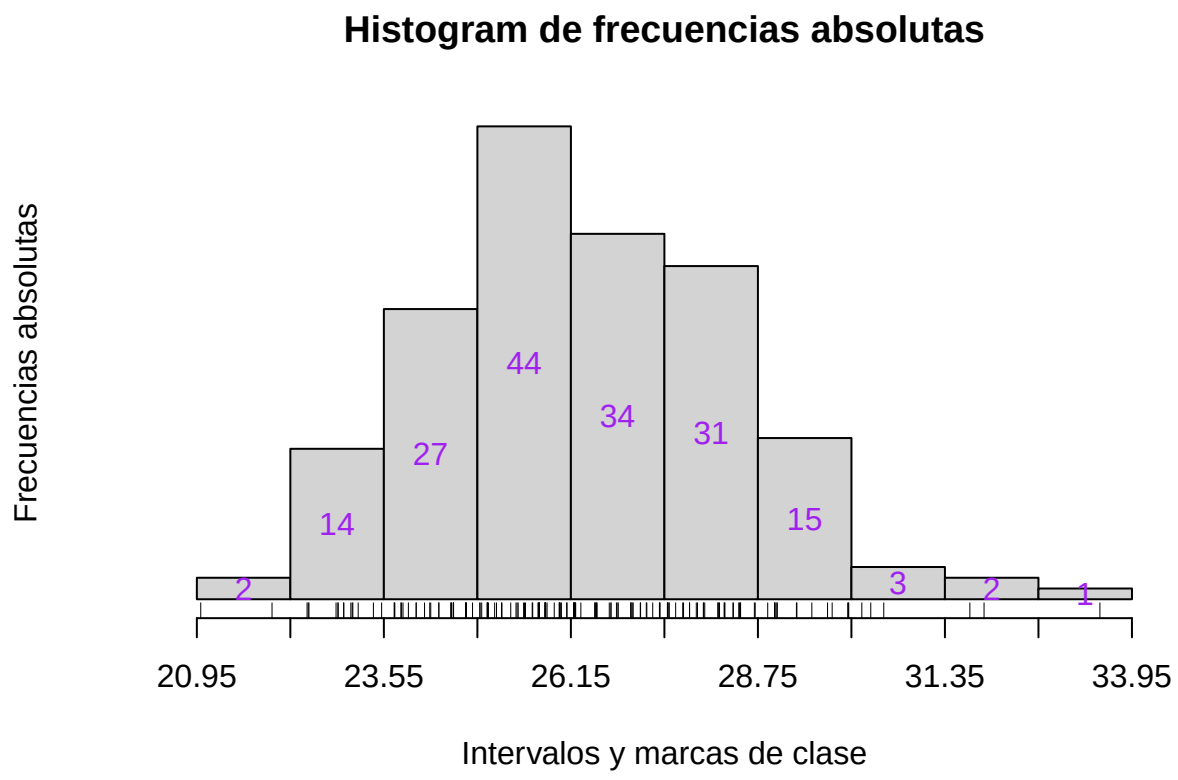
```r
quantile(notas, 0.25)
```

```
## 25%
##   2
```

Histogramas

```r
# Histograma de frecuencias absolutas
histAbs = function(x, L) {
  h = hist(x, breaks = L, right = FALSE, freq = FALSE,
           xaxt = "n", yaxt = "n", col = "lightgray",
           main = "Histogram 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")
  rug(jitter(x))
}

# Debido a que el area de las barras es igual a la frecuencia absoluta de cada barra,
# para ver mejor los datos, es mejor poner en medio de las barras su frecuencia
# absoluta
# y quitar el eje de las ordenadas, para asi evitar confusiones.
# Para poder agregar su frecuencia absoluta en el centro de la barra, nos sale mejor
# utilizar un histograma de frecuencias relativas, para que asi este relacionada su
# altura con la densidad y poderle sacar la mitad para ahi poner las frecuencias
# absolutas de cada barra

# xaxt y yaxt = "n" eliminan los ejes

# freq = True, es su valor por defecto. Si es True dibuja un histograma de frecuencias
# absolutas, si es False, de relativas

# axis() tiene como parametro un 1 para agregar el eje x, 2 par agregar el eje y. El
# parametro at indica que numeros se representaran en ese eje.

# El hist devuelve una lista, que dentro tiene la propiedad $density, la cual es igual
# a la frecuencia relativa dividida entra la amplitud, es decir la altura de la barra
# necesaria para que al multiplicar base por altura, de como resultado la frecuencia
# relativa de ese intervalo. En resumen, $density es la altura de la barra.

# rug() combinado con jitter() agrega una alfombra que explica la distribucion de los
# datos dentro de ese intervalo y jitter() nos ayuda a que no se vea una encima de la
# otra si es que hay datos iguales
```

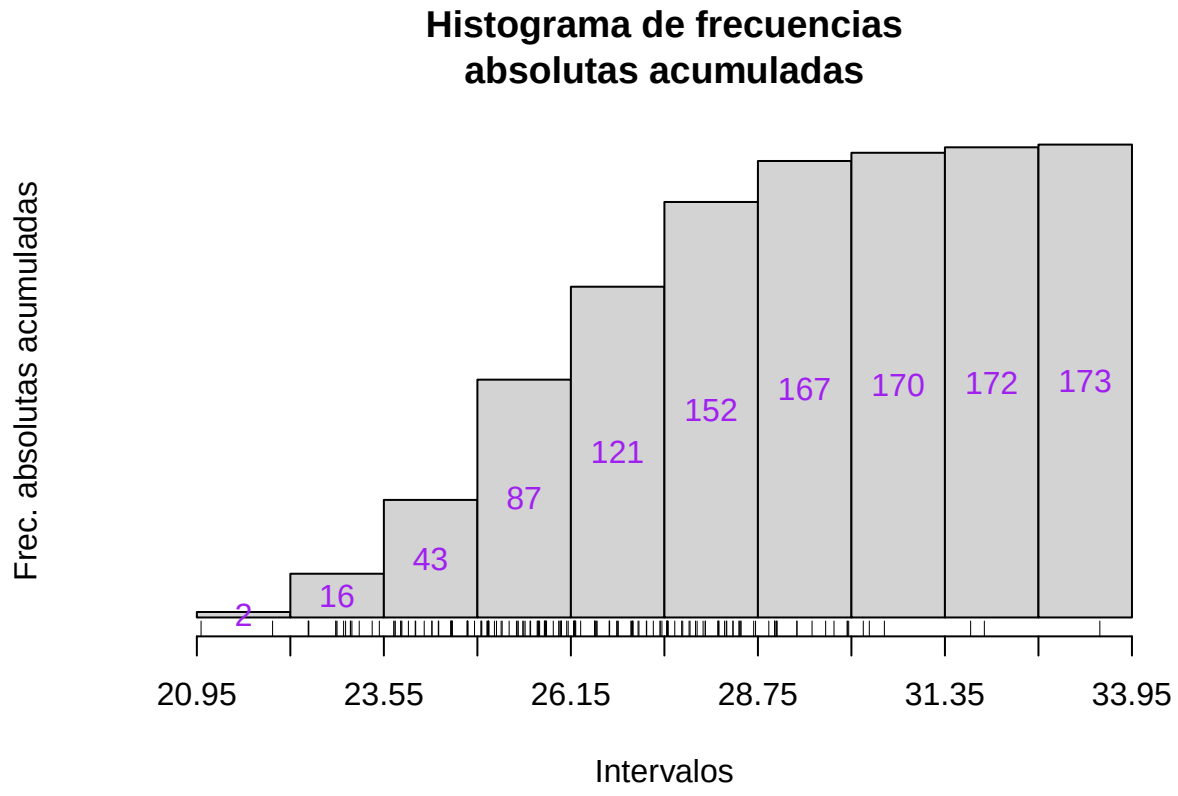
```r
histAbs(crabs$width, L = c(20.95, 22.25, 23.55, 24.85, 26.15, 27.45, 28.75, 30.05, 31.35, 32.65, 33.95))
```




```r
# Histograma de frecuencias absolutas acumuladas
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")
  rug(jitter(x))
}

# Para poder hacer que las alturas vayan creciendo, nos conviene utilizar el
# histograma pero que las densidades sean acumuladas para que las barras vayan
# creciendo
# Debido a que no podemos modificar este parametro antes de mostrar el grafico, nos
# conviene primero no mostralo, modificar ese parametro y una vez modificado,
# pasarselo como parametro a otro plot, el cual tambien representara el histograma.


# Si ponemos el freq = TRUE o FALSE, no cambia lo que devuelve la funcion hist().

# plot() solo hara uso de las caracteristicas que trae dentro de lo que devuelve
# hist(), es por eso que son necesarios los demas parametros en el plot(), ya que si
# fueron puestos en el hist(), no se le transmitiran al plot(). Por eso es que
# agregamos el freq = FALSE y los demas argumentos en el plot.
```

```
# Ahora en el text() hacemos un cumsum(h$counts) para mostrar las frecuencias
# absolutas acumuladas
```

```
histAbsCum(crabs$width, c(20.95, 22.25, 23.55, 24.85, 26.15, 27.45, 28.75, 30.05, 31.35, 32.65, 33.95))
```

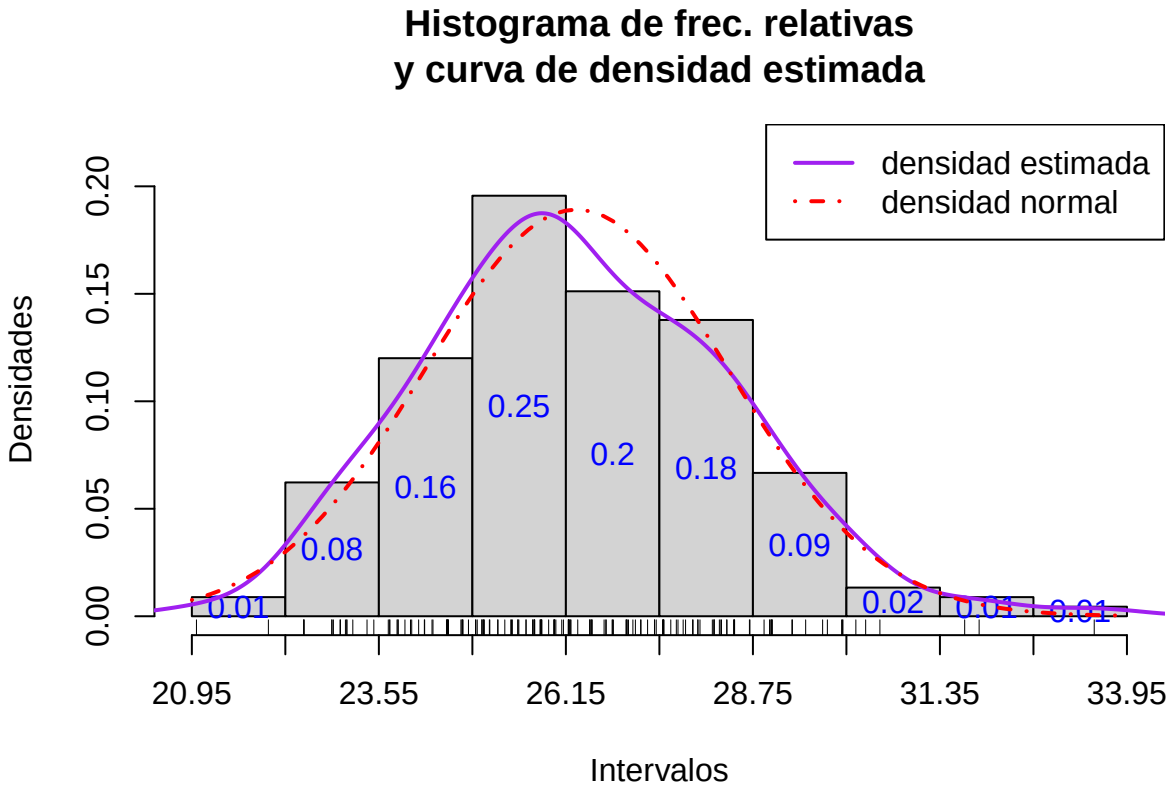



```
# Histograma de frecuencias relativas

# 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.

# La curva de densidad que se estima, se le conoce como Kernel Density Estimation
# (KDE), para recuerdes como funciona en caso de que se te olvide, te dejo este video:
# https://www.youtube.com/watch?v=MZigYoDwKDM

# Lo mas importante es saber que describe la probabilidad de que la variable tome un
# valor determinado. Conociendo la funcion de densidad, podemos calcular la
# probabilidad de que el valor de nuestra variable caiga en una region especifica.
# Dicha probabilidad se obtiene calculando la integral de la funcion de densidad
# comprendida entre los limites superior e inferior de la region en cuestion
```

```r
# El area bajo la curva representa el 100% de los datos del intervalo que abarca.

# Los programas que calculan el KDE, ya saben por defecto la mejor manera de
# calcularlo, en este caso, lo hace la funcion density()
```

```r
histRel = function(x,L) {
  h = hist(x, breaks=L, right=FALSE , plot=FALSE)
  t = round(1.1*max(max(density(x)[[2]]),h$density),2) # Indicamos la altura que
  # tendra el eje de las "y", el cual sera 10% mas grande que el valor mas grande
  # encontrado, ya sea del KDE, o de las alturas de la barra
  plot(h, freq = FALSE, col = "lightgray",
       main = "Histograma de frec. relativas\ny curva de densidad estimada",
       xaxt="n", ylim=c(0,t), xlab="Intervalos", ylab="Densidades")
  axis(1, at = L)
  text(h$mids, h$density/2, labels = round(h$counts/length(x),2), col = "blue")
  lines(density(x), col = "purple", lwd = 2) # Los puntos del KDE se calculan con
  # density() y lines() la dibuja. Se puede acceder a x e y de density con
  # str(density(x))
  rug(jitter(x))
}
```

```r
histRel(crabs$width, c(20.95, 22.25, 23.55, 24.85, 26.15, 27.45, 28.75, 30.05, 31.35, 32.65, 33.95))
curve(dnorm(x, mean(crabs$width), sd(crabs$width)), # x va si la queremos en el eje x
      col = "red", lty = 4, lwd = 2, add = TRUE)
legend("topright", lwd = c(2, 2), lty = c(1,4), col = c("purple", "red"),
       legend = c("densidad estimada", "densidad normal"))
```




```
# Agregamos una campana de Gauss para compararlos.
# Si se parecen, afirmamos que los datos de los cangrejos tienen una distribucion
# normal
```

```
# Histograma de frecuencias relativas acumuladas

# En este caso la curva no sera de densidad, sera la funcion de distribucion estmiada.
# Esta funcion, en cada punto nos da la fraccion de individuos de la poblacion que
# caen a la izquierda de este punto: su frecuencia relativa acumulada

# En general, la funcion de distribucion en un valor determinado, se obtiene hallando
# el area de la funcion de densidad que hay a la izquierda del valor. Solo se puede
# tomar un valor (no es como la densidad), y apartir de ese, toda el area de la
# izquierda, sera la frecuencia relativa acumulada

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]) # Lo que se hace aqui, es
```

```
  # hacer que la curva vaya subiendo con un cumsum(dens.x$y). Pero si solo hacemos
  # eso, la curva sera casi como una linea que no parara de subir, entonces lo que se
  # hace es multiplicarlo por la diferencia que hay entre cada uno de los valores de
  # x, para que haga el correcto efecto de la funcion de distribucion
  lines(dens.x,col = "purple",lwd = 2)
}
```

```
histRelCum(crabs$width, c(20.95, 22.25, 23.55, 24.85, 26.15, 27.45, 28.75, 30.05, 31.35, 32.65, 33.95))
```