DatosAgrupados

Gabriel

2022-04-13

Estadisticos para datos agrupados

Al tener datos numericos, es necesario calcular siempre antes de agrupar ciertos estadisticos, si no queda mas remedio, que la fuente los tenga agrupados o similares, aun siguie siendo posible calcular los estadisticos originales como aproximacion de los datos reales.

Estadisticos - Media - Varianza - Desviacion tipica - Moda

La diferencia es que es que ahora usaremos la marca de clase multiplicada para dicha clase respectivamente

En lo que se refiere a la moda, se cambia por el intercambio moda, viene a ser la clase que tiene mayor frecuencia absoluta y/o relativa

Le mediana se sustituye por el intervalo tipico para la mediana, un intervalo cuya su frecuencia relativa acumulada sea mayor a 0.5, el primero

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

Dicha formula se extiende para el calculo de cuantiles

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

Si no podemos acceder a los datos raw, hay que hacer estos calculos si tenemos agrupados

Ejercicio con agrupados

```
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
}

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
}
```

```
#Trabajamos nuevamente con data crabs
datacrab = read.table("../data/datacrab.txt", header = T)
cw = datacrab$width
#Determinando la tabla de frecuencias
k = 10
A = 1.3
L_1 = min(cw)-(1/2*0.1)
L = L_1 + A*(0:k)
mc = (L[1]+L[2])/2+A*(0:(k-1))
intervals = as.character(c("[20.95,22.25)","[22.25,23.55)","[23.55,24.85)","[24.85,26.15)","[26.15,27.45
cw_cut = cut(cw, breaks = L, right = F)
Fr.abs = as.vector(table(cw_cut))
Fr.rel = round(Fr.abs/length(cw),4)
Fr.cum.abs = cumsum(Fr.abs)
Fr.cum.rel = cumsum(Fr.rel)
cw_df = data.frame(intervals, mc, Fr.abs, Fr.cum.abs, Fr.rel, Fr.cum.rel)
cw_df
```

```
##        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
```

```
#Calculando estadisticos agrupados

#Total de muestras
TOT = cw_df$Fr.cum.abs[10]

#la sumatoria de las frecuencias absolutas de cada intervalo por la marca de clase de cada uno dividido
anchura.media = round(sum(cw_df$Fr.abs*cw_df$mc)/TOT,3)
anchura.media
```

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

```r
#la varianza, la sumatoria de las frecuencias absolutas de cada intervalo, por la marca de clase al cua
anchura.var = round(sum(cw_df$Fr.abs*cw_df$mc^2)/TOT - anchura.media^2, 3)
anchura.var
```

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

```r
#desviacion tipica, la raiz cuadrada de la varianza
anchura.dt = round(sqrt(anchura.var),3)
anchura.dt
```

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

```r
#Intervalo modal, le pido que de los intervalos, me de cuando la frecuencia absoluta sea igual al maxim
I.modal = cw_df$intervals[which(cw_df$Fr.abs == max(cw_df$Fr.abs))]
I.modal
```

```
## [1] "[24.85,26.15)"
```

```r
#Intervalo critico para la mediana, de los intervalos, los que tengan frecuencia relativa acumulada may
I.critic = cw_df$intervals[which(cw_df$Fr.cum.rel >= 0.5)]
I.critic[1]
```

```
## [1] "[24.85,26.15)"
```

```r
#Ahora vamos a la estimacion de la mediana real
n = TOT
Lc = L[4]
Lc.pos= L[5]
Ac = L[5]-L[4]
Nc.ant = cw_df$Fr.cum.abs[3]
nc = cw_df$Fr.abs[4]
M = Lc+Ac*((n/2)-Nc.ant)/nc
M # Aproximacion de la mediana en datos reales
```

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

```r
#Con la funcion median, si tengo disponibles los datos raw
median(cw)
```

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

```r
#Ver formulas para calcular quantiles
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
```

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

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

Histogramas de frecuencia

Si las amplitudes de los intervalos son distintas, no vamos a tener un histograma 100% representativo, en este caso hay que mirar las areas representadas y no las alturas

En frecuencias relativas, represento la densidad, si sumo todas las areas, me debe de dar 1

Frecuencias nulas, no es conveniente representar en el histograma, a no ser que lo quermos representar a proposito

Funcion `hist()`, x es el vector, breaks son los intervalos, se puede pasar k y entre comillas el metodo "Scott"

Calculo density del histograma, corresponden a las alturas de las barras, frecuencia relativa dividida por su amplitud

Histograma de frecuencia absoluta

```r
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")
  }
```

Histograma de recuencia absoluta acumulada

```r
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")
  }
```

Histograma de frecuencia relativa

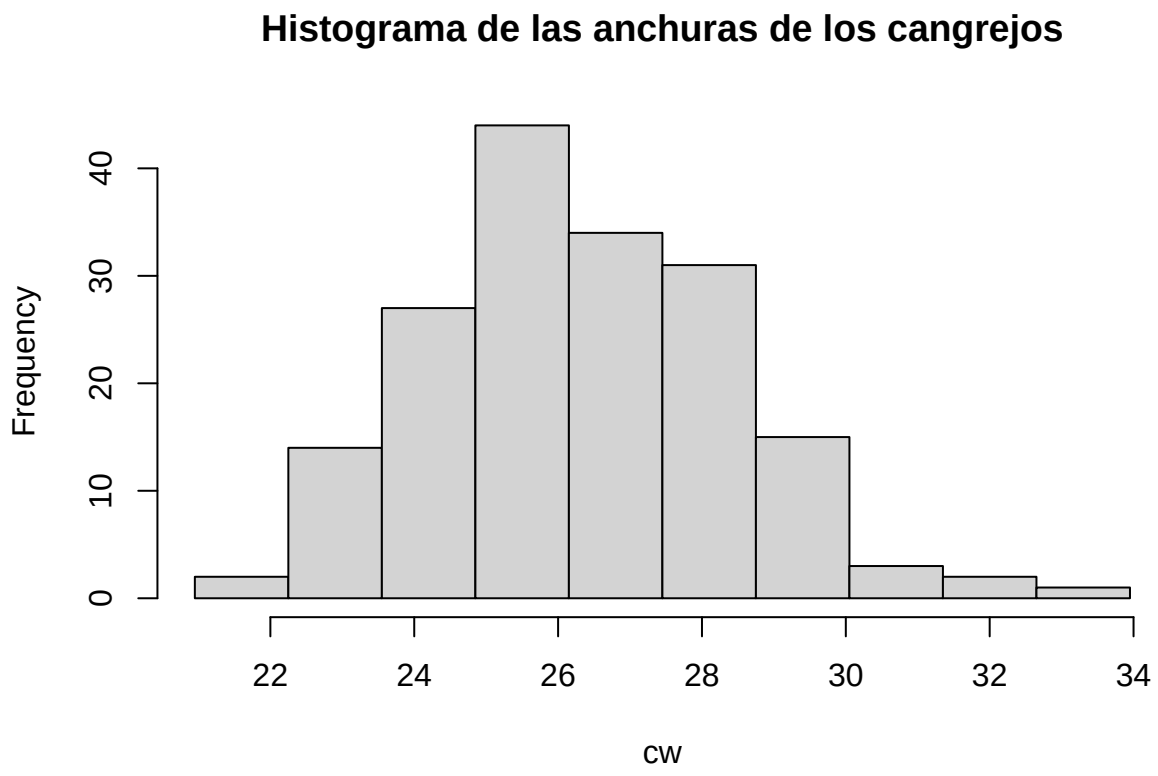
```r
histRel = function(x,L) {
  h = hist(x, breaks=L, right=FALSE , plot=FALSE)
  t = round(1.1*max(max(density(x)[[2]]),h$density),2)
  plot(h, freq = FALSE, col = "lightgray",
       main = "Histograma de frec. relativas\ny curva de densidad estimada",
       xaxt="n", ylim=c(0,t), xlab="Intervalos",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

Histograma de 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])
  lines(dens.x,col = "purple",lwd = 2)
}
```

Practica con los cangrejos

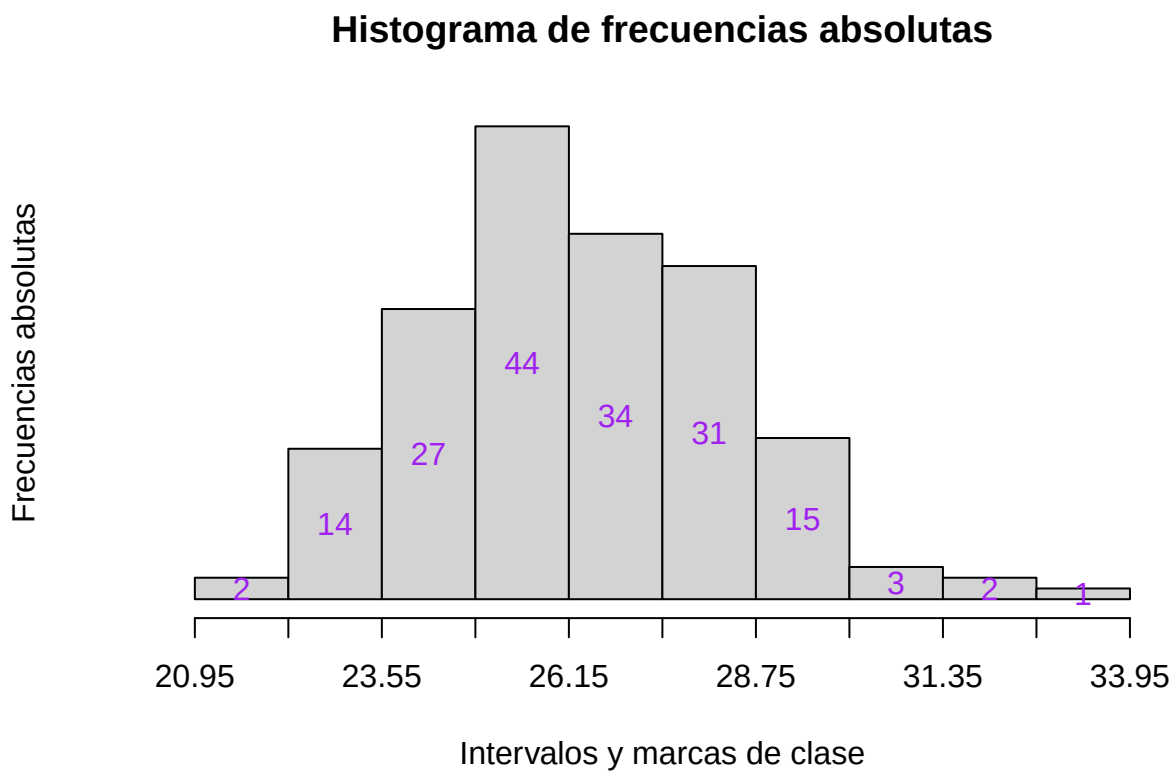
```
#Trabajando con cw, aplicando primero la funcion hist
hist(cw, breaks = L, right = F, main = "Histograma de las anchuras de los cangrejos")
```




```
#Mirando la estrucutra interna, con plot = F
hist(cw, breaks = L, right = F, plot = F)
```

```
## $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 0.137839040
##  [7] 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"
```

```
#Usando las funciones preparadas, primero absoluta, luego absoluta acumulada
histAbs(cw, L)
```



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