

10-Script para calcular la tabla de frecuencias con datos agrupados

Adrian

19/1/2022

Frecuencias

- La frecuencia abs de una clase sera el numero de datos originales que pertenecen a la clase.
- La frecuencia abs acumulada de una clase sera el numero de datos que pertenecen a dicha clase o alguna de las anteriores.

Primer Forma de obtener la tabla de frecuencias

Para cuando todas las clases son de la misma amplitud

```
TablaFrecs = function(x,k,A,p){  
  L = min(x)-p/2+A*(0:k)  
  x_cut = cut(x, breaks = L, right = F)  
  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  
}
```

Segunda Forma de obtener la tabla de frecuencias

Para cuando todas las clases no son de la misma amplitud

```
TablaFrecs.L = function(x,L,V){  
  x_cut = cut(x, breaks = L, right = F, 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  
}
```

Tabla de frecuencias de la longitud de pétalos de iris es:

```
petals = iris$Petal.Length
TablaFrecs(petals, k=6, A=1, p=0.1)
```

```
##      intervals    mc Fr.abs Fr.cum.abs Fr.rel Fr.cum.rel
## 1 [0.95,1.95) 1.45     50          50 0.3333     0.3333
## 2 [1.95,2.95) 2.45      0          50 0.0000     0.3333
## 3 [2.95,3.95) 3.45     11         61 0.0733     0.4066
## 4 [3.95,4.95) 4.45     43        104 0.2867     0.6933
## 5 [4.95,5.95) 5.45     35        139 0.2333     0.9266
## 6 [5.95,6.95) 6.45     11        150 0.0733     0.9999
```

```
TablaFrecs.L(petals, L=1:7, V=FALSE)
```

```
##      intervals    mc Fr.abs Fr.cum.abs Fr.rel Fr.cum.rel
## 1      [1,2) 1.5      50          50 0.3333     0.3333
## 2      [2,3) 2.5       0          50 0.0000     0.3333
## 3      [3,4) 3.5      11         61 0.0733     0.4066
## 4      [4,5) 4.5      43        104 0.2867     0.6933
## 5      [5,6) 5.5      35        139 0.2333     0.9266
## 6      [6,7) 6.5      11        150 0.0733     0.9999
```

Ejemplo con todas las funciones

```
notas = sample(1:10, 100, replace = T)
notas
```

```
##      [1] 10  4  5  5  2  3 10  5  4  7  1  3  7  7  1  5  2  9  6  7  4  8  2  5 10
##     [26]  5  8  7  9  3  3  4  5  5  5  8  3  4  8  5  1  6  8  3  5  4  1  4  4  4
##     [51] 10  4  1  6  8  2  3  8  3  1  6  6  5  7  5  6  8  1  1 10  6  2  2  8  8
##     [76]  4  4  6  7  6  9  6  7  7  7  5  6  4  8  3  5  9  4  8  3  3  4  6  4  4
```

```
# Vamos a agrupar las notas en [0,5), [5,7), [7,9), [9,10].
```

```
# 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 puntos
```

```
notas1 = cut(notas, breaks = L, right = F, include.lowest = T)
```

```
notas1
```

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

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

```
## [1] 9.5 2.5 6 6 2.5 2.5 9.5 6 2.5 8 2.5 2.5 8 8 2.5 6 2.5 9.5
## [19] 6 8 2.5 8 2.5 6 9.5 6 8 8 9.5 2.5 2.5 2.5 6 6 6 8
## [37] 2.5 2.5 8 6 2.5 6 8 2.5 6 2.5 2.5 2.5 2.5 2.5 9.5 2.5 2.5 6
## [55] 8 2.5 2.5 8 2.5 2.5 6 6 6 8 6 6 8 2.5 2.5 9.5 6 2.5
## [73] 2.5 8 8 2.5 2.5 6 8 6 9.5 6 8 8 8 6 6 2.5 8 2.5
## [91] 6 9.5 2.5 8 2.5 2.5 2.5 6 2.5 2.5
## Levels: 2.5 6 8 9.5
```

```
# Definimos notas3 como el resultado de la codificacion en intervalos utilizando como etiquetas la posición
notas3 = cut(notas, breaks = L, labels = F, right = F, include.lowest = T)
notas3
```

```
## [1] 4 1 2 2 1 1 4 2 1 3 1 1 3 3 1 2 1 4 2 3 1 3 1 2 4 2 3 3 4 1 1 1 2 2 2 3 1
## [38] 1 3 2 1 2 3 1 2 1 1 1 1 1 4 1 1 2 3 1 1 3 1 1 2 2 2 3 2 2 3 1 1 4 2 1 1 3
## [75] 3 1 1 2 3 2 4 2 3 3 3 2 2 1 3 1 2 4 1 3 1 1 1 2 1 1
```

```
# Definimos notas4 como el resultado de la codificacion en intervalos utilizando como etiquetas (Susp, Aprob, Not, Exc)
notas4 = cut(notas, breaks = L, labels = c("Susp", "Aprob", "Not", "Exc"), right = F, include.lowest = T)
notas4
```

```
## [1] Exc Susp Aprob Aprob Susp Susp Exc Aprob Susp Not Susp Susp
## [13] Not Not Susp Aprob Susp Exc Aprob Not Susp Not Susp Aprob
## [25] Exc Aprob Not Not Exc Susp Susp Susp Aprob Aprob Aprob Not
## [37] Susp Susp Not Aprob Susp Aprob Not Susp Aprob Susp Susp Susp
## [49] Susp Susp Exc Susp Susp Aprob Not Susp Susp Not Susp Susp
## [61] Aprob Aprob Aprob Not Aprob Aprob Not Susp Susp Exc Aprob Susp
## [73] Susp Not Not Susp Susp Aprob Not Aprob Exc Aprob Not Not
## [85] Not Aprob Aprob Susp Not Susp Aprob Exc Susp Not Susp Susp
## [97] Susp Aprob Susp Susp
## Levels: Susp Aprob Not Exc
```

#notas1 y notas2 han producido factores mientras que notas3 y notas4 han producido un vector

```
# Frecuencia absoluta
table(notas4)
```

```
## notas4
## Susp Aprob Not Exc
## 42 27 22 9
```

```
# Frecuencia relativa
prop.table(table(notas4))
```

```
## notas4
## Susp Aprob Not Exc
## 0.42 0.27 0.22 0.09
```

```
# Frecuencia acumulada absoluta
cumsum(table(notas4))
```

```
##  Susp Aprob  Not   Exc
##    42    69   91  100
```

```
# Frecuencia acumulada relativa
cumsum(prop.table(table(notas4)))
```

```
##  Susp Aprob  Not   Exc
##  0.42  0.69  0.91  1.00
```

Obtener todo lo anterior con hist()

```
notasHist = hist(notas, breaks = L, right = F, include.lowest = T, plot = F)
# Frecuencia absoluta
FAbs = notasHist$count
# Frecuencia relativa
FRel = prop.table(FAbs)
# Frecuencia acumulada absoluta
FAbsCum = cumsum(FAbs)
# Frecuencia acumulada relativa
FRelCum = cumsum(FRel)
```

Crear un data frame con los datos

```
intervalos = c("[0,5)", "[5,7)", "[7,9)", "[9,10]")
calificacion = c("Suspendo", "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)    Suspendo    2.5   42      42 0.42    0.42
## 2  [5,7)    Aprobado    6.0   27      69 0.27    0.69
## 3  [7,9)    Notable     8.0   22      91 0.22    0.91
## 4  [9,10]   Excelente    9.5    9     100 0.09    1.00
```

```
TablaFreCs.L(notas, L, T)
```

O usando nuestra funcion

```
##  intervals mc Fr.abs Fr.cum.abs Fr.rel Fr.cum.rel
## 1  [0,5) 2.5   42      42   0.42      0.42
## 2  [5,7) 6.0   27      69   0.27      0.69
## 3  [7,9) 8.0   22      91   0.22      0.91
## 4  [9,10] 9.5    9     100   0.09      1.00
```