

Las Notas de Bachillerato

Santiago Pérez Moncada

13/7/2020

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

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

```
notas = sample(0:10,100,replace = TRUE)
notas
```

```
## [1] 7 1 7 6 4 9 7 6 7 5 0 0 2 8 10 2 2 0 9 1 7 9 6 2 9
## [26] 7 3 8 4 0 1 2 3 3 4 9 1 4 6 0 5 2 0 10 5 8 7 8 4 3
## [51] 3 2 10 8 1 3 4 1 1 9 7 5 9 7 8 3 8 6 2 0 2 10 10 3 0
## [76] 6 3 7 0 6 0 9 1 8 10 9 6 8 9 9 3 3 2 1 9 2 4 2 9 8
```

Solución

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

```
#Definimos el 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) [0,5) [7,9) [5,7) [0,5) [9,10] [7,9) [5,7) [7,9) [5,7)
## [11] [0,5) [0,5) [0,5) [7,9) [9,10] [0,5) [0,5) [0,5) [9,10] [0,5)
## [21] [7,9) [9,10] [5,7) [0,5) [9,10] [7,9) [0,5) [7,9) [0,5) [0,5)
## [31] [0,5) [0,5) [0,5) [0,5) [0,5) [9,10] [0,5) [0,5) [5,7) [0,5)
## [41] [5,7) [0,5) [0,5) [9,10] [5,7) [7,9) [7,9) [7,9) [0,5) [0,5)
## [51] [0,5) [0,5) [9,10] [7,9) [0,5) [0,5) [0,5) [0,5) [0,5) [9,10]
## [61] [7,9) [5,7) [9,10] [7,9) [7,9) [0,5) [7,9) [5,7) [0,5) [0,5)
## [71] [0,5) [9,10] [9,10] [0,5) [0,5) [5,7) [0,5) [7,9) [0,5) [5,7)
## [81] [0,5) [9,10] [0,5) [7,9) [9,10] [9,10] [5,7) [7,9) [9,10] [9,10]
## [91] [0,5) [0,5) [0,5) [0,5) [9,10] [0,5) [0,5) [0,5) [9,10] [7,9)
## 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
#Definamos 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 2.5 8 6 2.5 9.5 8 6 8 6 2.5 2.5 2.5 8 9.5 2.5 2.5 2.5
## [19] 9.5 2.5 8 9.5 6 2.5 9.5 8 2.5 8 2.5 2.5 2.5 2.5 2.5 2.5 2.5 9.5
## [37] 2.5 2.5 6 2.5 6 2.5 2.5 9.5 6 8 8 8 2.5 2.5 2.5 2.5 9.5 8
## [55] 2.5 2.5 2.5 2.5 2.5 9.5 8 6 9.5 8 8 2.5 8 6 2.5 2.5 2.5 9.5
## [73] 9.5 2.5 2.5 6 2.5 8 2.5 6 2.5 9.5 2.5 8 9.5 9.5 6 8 9.5 9.5
## [91] 2.5 2.5 2.5 2.5 9.5 2.5 2.5 2.5 9.5 8
## Levels: 2.5 6 8 9.5
```

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

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

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

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

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

```
## [1] [5,7.5) [-0.01,2.5) [5,7.5) [5,7.5) [2.5,5) [7.5,10]
## [7] [5,7.5) [5,7.5) [5,7.5) [5,7.5) [-0.01,2.5) [-0.01,2.5)
## [13] [-0.01,2.5) [7.5,10] [7.5,10] [-0.01,2.5) [-0.01,2.5) [-0.01,2.5)
## [19] [7.5,10] [-0.01,2.5) [5,7.5) [7.5,10] [5,7.5) [-0.01,2.5)
## [25] [7.5,10] [5,7.5) [2.5,5) [7.5,10] [2.5,5) [-0.01,2.5)
## [31] [-0.01,2.5) [-0.01,2.5) [2.5,5) [2.5,5) [2.5,5) [7.5,10]
## [37] [-0.01,2.5) [2.5,5) [5,7.5) [-0.01,2.5) [5,7.5) [-0.01,2.5)
## [43] [-0.01,2.5) [7.5,10] [5,7.5) [7.5,10] [5,7.5) [7.5,10]
## [49] [2.5,5) [2.5,5) [2.5,5) [-0.01,2.5) [7.5,10] [7.5,10]
## [55] [-0.01,2.5) [2.5,5) [2.5,5) [-0.01,2.5) [-0.01,2.5) [7.5,10]
## [61] [5,7.5) [5,7.5) [7.5,10] [5,7.5) [7.5,10] [2.5,5)
## [67] [7.5,10] [5,7.5) [-0.01,2.5) [-0.01,2.5) [-0.01,2.5) [7.5,10]
## [73] [7.5,10] [2.5,5) [-0.01,2.5) [5,7.5) [2.5,5) [5,7.5)
## [79] [-0.01,2.5) [5,7.5) [-0.01,2.5) [7.5,10] [-0.01,2.5) [7.5,10]
## [85] [7.5,10] [7.5,10] [5,7.5) [7.5,10] [7.5,10] [7.5,10]
## [91] [2.5,5) [2.5,5) [-0.01,2.5) [-0.01,2.5) [7.5,10] [-0.01,2.5)
## [97] [2.5,5) [-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 extremos izquierdo del primer intervalo.

Trabajaremos ahora con `notas4` y calculemos sus frecuencias:

```
table(notas4) #Frec.Absolutas
```

```
## notas4
## Susp Aprob Not Exc
## 49 12 20 19
```

```
prop.table(table(notas4)) #Frec.Relativas
```

```
## notas4
## Susp Aprob Not Exc
## 0.49 0.12 0.20 0.19
```

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

```
## Susp Aprob Not Exc
## 49 61 81 100
```

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

```
## Susp Aprob Not Exc
## 0.49 0.61 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	49	49	0.49	0.49
## 2	[5,7)	Aprobado	6.0	12	61	0.12	0.61
## 3	[7,9)	Notable	8.0	20	81	0.20	0.81
## 4	[9,10]	Excelente	9.5	19	100	0.19	1.00