Agrupando datos con R

Santiago Pérez Moncada

4/7/2020

Al agrupar los datos, lo que hacenmos 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 entro los valores mínimo y máximo de x en k intervalos y a continuación, dezplaza ligeramente el extremo inferior del primer intervalo a la izquierda y ek 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`, obtenemos 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 igualado 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** igualado 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.

Ejemplo

```
iris_df = iris
petals = iris$Petal.Lengt
head(petals)
```

```
## [1] 1.4 1.4 1.3 1.5 1.4 1.7
```

```
iris_df$div1 = cut(petals, breaks = 5, right = FALSE)

head(cut(petals, breaks = ceiling(sqrt(length(petals))), right = FALSE))
```

```
## [1] [0.994,1.45) [0.994,1.45) [0.994,1.45) [1.45,1.91)  [0.994,1.45)
## [6] [1.45,1.91)
## 13 Levels: [0.994,1.45) [1.45,1.91) [1.91,2.36) [2.36,2.82) ... [6.45,6.91)
```

```
iris_df$div2 = cut(petals, breaks = c(1,2,3,4,5,6,7), right = FALSE)

head(cut(petals, breaks = 5, right = FALSE, labels = FALSE))
```

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

```
iris_df$div3 = cut(petals, breaks = 5, right = FALSE,
                labels = c("Peq","Norm","Grand", "XGran","Gigan"))

head(iris_df)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species         div1  div2
## 1          5.1         3.5          1.4         0.2  setosa [0.994,2.18) [1,2)
## 2          4.9         3.0          1.4         0.2  setosa [0.994,2.18) [1,2)
## 3          4.7         3.2          1.3         0.2  setosa [0.994,2.18) [1,2)
## 4          4.6         3.1          1.5         0.2  setosa [0.994,2.18) [1,2)
## 5          5.0         3.6          1.4         0.2  setosa [0.994,2.18) [1,2)
## 6          5.4         3.9          1.7         0.4  setosa [0.994,2.18) [1,2)
##   div3
## 1  Peq
## 2  Peq
## 3  Peq
## 4  Peq
## 5  Peq
## 6  Peq
```