

Subvectores

Téllez Gerardo Rubén

14/3/2021

Para marcar parte de un vector

- **vector[i]**: da la i-esima entrada del vector
 - Los índices de R empiezan en 1 (no como en Py)
 - **vector[length(vector)]**: nos da la última entrada del vector
 - **vector[a:b]**: si a y b son dos números naturales, nos da el subvector con las entradas del vector original que van de la posición a-ésima hasta la b-ésima
 - **vector[-i]**: si i es un número, este subvector tendrá todas las entradas excepto la i-ésima.
 - **vector[i]**: Si i es un vector actúa como vector de índices, y crea un vector con los i índices.
 - **vector[-x]**: si x es un vector, otorga el complemento del vector x
 - **vector[length(x)]**: otorga la última entrada, $length(x)$ en realidad devuelve el índice último, por lo que puede restarse algún número para obtener las entradas de índices anteriores

```
a <- c(1, 2, 3, 4, 5, 5.2)
```

```
i = 1:3  
a[i]
```

```
## [1] 1 2 3
```

```
j = 3:1  
a[j]
```

```
## [1] 3 2 1
```

```
x <- seq(3, 50, by=3.5)  
x
```

```
## [1] 3.0 6.5 10.0 13.5 17.0 20.5 24.0 27.5 31.0 34.5 38.0 41.5 45.0 48.5
```

```
x[3]
```

```
## [1] 10
```

```
x[length(x)]
```

```
## [1] 48.5
```

```
x[length(x)-2]
```

```
## [1] 41.5
```

```
x[-i]
```

```
## [1] 13.5 17.0 20.5 24.0 27.5 31.0 34.5 38.0 41.5 45.0 48.5
```

```
x[i]
```

```
## [1] 3.0 6.5 10.0
```

- También podemos usar operadores lógicos:

- ==
- !=
- =
- <=
- <
-
- !: “NO” LÓGICO
- &: “Y” lógico
- |: “O” lógico

```
v = c(14, 5, 8, 40, 8, 71, 4)
v[3]
```

```
## [1] 8
```

```
#Para quitar las entradas 3 y 5
v[-c(3,5)]
```

```
## [1] 14 5 40 71 4
```

```
#Diferente de 40 y más grande de 10
v[v != 40 & v>10]
```

```
## [1] 14 71
```

```
#Aquellos más grandes que 10 o iguales a 5
v[v > 10 | v == 5]
```

```
## [1] 14 5 40 71
```

```
#Para filtrar aquellos que no sean menores a 10  
v[!x <10]
```

```
## [1] 8 40 8 71 4 NA NA NA NA NA NA NA
```

Condicionales de vectores

- **which(x cumple condición):** Para obtener los índices de las entradas del vector `x` que satisfacen la condición dada
- **which.min(x):** Para obtener la posición en el que x tiene su valor mínimo (en caso de empate se da el primero)
- **which(x==min(x)):** Da todas las posiciones en las que el vector x tiene su valor mínimo (siempre se dan todas las posiciones)
- **which.max(x):** Para obtener la primera posición donde x obtiene su valor máximo
- **which(x==max(x)):** Da todas las posiciones en las que el vector x tiene su valor máximo (siempre se dan todas las posiciones)

```
x <- seq(3, 50, by=3.5)  
x
```

```
## [1] 3.0 6.5 10.0 13.5 17.0 20.5 24.0 27.5 31.0 34.5 38.0 41.5 45.0 48.5
```

```
#Para obtener las entradas de índices pares  
x[seq(1, length(x), by=2)]
```

```
## [1] 3 10 17 24 31 38 45
```

```
#Para quitar los de posición par  
x[-seq(1, length(x), by=2)]
```

```
## [1] 6.5 13.5 20.5 27.5 34.5 41.5 48.5
```

```
#Para obtener los valores pares  
x[which(x %% 2 == 0)]
```

```
## [1] 10 24 38
```

```
#Para obtener los últimos 3 elementos  
x[(length(x)-2) :length(x)]
```

```
## [1] 41.5 45.0 48.5
```

```
#Para filtrar el cuarto, sexto y décimo  
x[c(4, 5, 10)]
```

```
## [1] 13.5 17.0 34.5
```

Regresar cadenas booleanas

```
bx = x < 5
bx
```

```
## [1] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE
```

```
y = seq(3, 50, length.out=14)
x < y
```

```
## [1] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [13] TRUE TRUE
```

Vector vacío

Un vector vacío se regresa cuando no hay elementos que satisfagan la condición.

```
x[x == 50]
```

```
## numeric(0)
```

```
which(x < 2)
```

```
## integer(0)
```

```
v = c()
v
```

```
## NULL
```