

# Estructura de datos con R - Subvectores

Ramon Ceballos

15/1/2021

## Subvectores

Vamos a ver diferentes notaciones que nos permiten trabajar con partes del total del vector.

- `vector[i]`: da la  $i$ -ésima entrada del vector
  - Los índices en R empiezan en 1
  - `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 está formado por todas las entradas del vector original menos la entrada  $i$ -ésima que hayamos dado. Si  $i$  resulta ser un vector, entonces es un vector de índices y crea un nuevo vector con las entradas del vector original, cuyos índices pertenecen a  $i$
  - `vector[-x]`: si  $x$  es un vector (de índices), entonces este es el complementario de `vector[x]`

## Operadores lógicos

También a la hora de acceder a posiciones podemos utilizar **operadores lógicos**:

```
- '==' : =
- '!=' :  $\neq$ 
- '>' :  $\geq$ 
- '<=' :  $\leq$ 
- '<' :  $<$ 
- '>' :  $>$ 
- '!' : NO lógico
- '&' : Y lógico
- '|' : O lógico
```

```
v = c(14,5,6,19,32,0,8)
#Muestra el elemento de la posición 2
v[2]
```

```
## [1] 5
```

```
#Da un vector que elimina los elemntos de las posiciones 3 y 5
v[-c(3,5)]
```

```
## [1] 14 5 19 0 8
```

```
#Solo apareceran en el nuevo vector los elementos distintos de 19 y mayores de 15  
v[v != 19 & v>15]
```

```
## [1] 32
```

## Condicionales

Existen las funciones `which`, que dado un vector `x` buscan los índices que cumplen cierta **condición**.

- `which(x cumple condición)`: para obtener los índices de las entradas del vector `x` que satisfacen la condición dada
- `which.min(x)`: nos da la primera posición en la que el vector `x` toma su valor mínimo
- `which(x==min(x))`: da todas las posiciones en las que el vector `x` toma sus valores mínimos
- `which.max(x)`: nos da la primera posición en la que el vector `x` toma su valor máximo
- `which(x==max(x))`: da todas las posiciones en las que el vector `x` toma sus valores máximos

## Ejemplos en chunks

*Operadores lógicos*

```
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
```

```
#posición 8  
x[8]
```

```
## [1] 27.5
```

```
#última posición  
x[length(x)]
```

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

```
#quitar n°  
x[-3]
```

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

```
#nuevo vector  
x[4:9]
```

```
## [1] 13.5 17.0 20.5 24.0 27.5 31.0
```

```
#Obtener n° pares del vector  
x[seq(2,length(x),by=2)]
```

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

```
#Quitar los n° pares  
x[-seq(2,length(x),by=2)]
```

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

```
#Filtrar elementos  
x[c(1,3,5)]
```

```
## [1] 3 10 17
```

```
#elementos mayores de 30  
x[x>30]
```

```
## [1] 31.0 34.5 38.0 41.5 45.0 48.5
```

```
#elementos menores de 10 y mayores de 35  
x[x<10|x>35]
```

```
## [1] 3.0 6.5 38.0 41.5 45.0 48.5
```

```
#SE PUEDE FILTRAR UN VECTOR SEGUN LOS VALORES DE OTRO VECTOR  
v1 <- c(1,3,5,7)  
v2 <- c(-9,2,5,1)  
v1[v2>0]
```

```
## [1] 3 5 7
```

*Condicionales*

```
#si aplicas which directamente da las posiciones del vector  
#x[which] da el valor de las posiciones  
s <- c(2,6,9,14,-5)  
which(s>4)
```

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

```
s[which(s>4)]
```

```
## [1] 6 9 14
```

```
s[s>4]
```

```
## [1] 6 9 14
```

```
#Posición del elemento más pequeño  
which.min(s) #si solo hay uno
```

```
## [1] 5
```

```
which(s == min(s)) #si hay varios minimos en el vectores
```

```
## [1] 5
```