

Vectorización

Luciano Selzer

28 June, 2018

R está vectorizado

La mayoría de las funciones están vectorizadas.

No hace falta hacer un bucle para operar sobre cada elemento.

- Es más legible
- Es más conciso
- Menor posibilidad de errores

```
x <- 1:4  
x * 2
```

```
[1] 2 4 6 8
```

La multiplicación ocurre sobre todo el vector

Podemos hacer adiciones

```
y <- 6:9  
x + y
```

```
[1] 7 9 11 13
```

```
x: 1 2 3 4  
   + + + +  
y: 6 7 8 9  
-----  
   7 9 11 13
```

Ejercicio 1

Intenta hacer esto en la columna `pop` en el dataset `gapminder`.

Haz una nueva columna en el data.frame `gapminder` que contenga la población en millones. Revisa el principio y el final del data.frame para asegurarte de que ha funcionado.

Ejercicio 2

En un solo gráfico grafica la población, en millones, versus el año, para todos los países. No te preocupes por identificar cada país

Repite el ejercicio, esta vez graficando solo China, India e Indonesia. Devuelta, no importa cual es cual.

Las operaciones de comparación, operadores lógicos y muchas otras funciones también están vectorizadas.

Operadores de comparación

```
x > 2
```

```
[1] FALSE FALSE TRUE TRUE
```

Operadores lógicos

```
a <- x > 3 # o, por claridad, a <- (x > 3)
a
```

```
[1] FALSE FALSE FALSE TRUE
```

La mayoría de las funciones operan sobre los elementos del vector.

Funciones

```
x <- 1:4
log(x)
```

```
[1] 0.0000000 0.6931472 1.0986123 1.3862944
```

Las operaciones vectorizadas trabajan en los elementos de las matrices:

```
m <- matrix(1:12, nrow = 3, ncol = 4)
m * -1
```

```
      [,1] [,2] [,3] [,4]
[1,]   -1   -4   -7  -10
[2,]   -2   -5   -8  -11
[3,]   -3   -6   -9  -12
```

Tip: multiplicación por elementos vs matricial

Muy importante: el operador `*` da la multiplicación por elementos!

Para hacer la multiplicación matricial, necesitamos usar el operador `%%`:

```
m %% matrix(1, nrow = 4, ncol = 1)
```

```
      [,1]
[1,]    22
[2,]    26
[3,]    30
```

Tip: multiplicación por elementos vs matricial

```
matrix(1:4, nrow = 1) %% matrix(1:4, ncol = 1)
```

```
      [,1]
[1,]    30
```

Para mas información sobre álgebra matricial puedan ver la guía de referencia de Quick-R

Ejercicio 3

Dada la siguiente matriz:

```
m <- matrix(1:12, nrow = 3, ncol = 4)
m
```

```
      [,1] [,2] [,3] [,4]
[1,]     1     4     7    10
[2,]     2     5     8    11
[3,]     3     6     9    12
```

Ejercicio 3

Escribe que crees que va a suceder cuando corras:

1. `m ^ -1`
2. `m * c(1, 0, -1)`
3. `m > c(0, 20)`
4. `m * c(1, 0, -1, 2)`

¿Dio la salida que esperabas? Si no fue así, pregunta.

Ejercicio 4

Estamos interesados en ver la suma de una secuencia de fracciones

```
x = 1/(1^2) + 1/(2^2) + 1/(3^2) + ... + 1/(n^2)
```

Esto sería tedioso de tipear e imposible de hacer para n grandes. Usa la vectorización para computar x cuando $n=100$ ¿Cuál es la suma cuando $n=10000$?