

Tarea 1 - ACP-149 - Estadística

Julio César Ramirez-Pacheco

2/10/2021

El lenguaje R, vectores

El lenguaje R, como se vió en sesiones anteriores tiene como un tipo de datos importante a los vectores. Los vectores atómicos se pueden generar de diversas formas, por ejemplo usando el comando `vector()` el cual toma dos parámetros, el tipo de vector atómico (`character`, `double`, etc) y el número de valores de ese tipo. Por ejemplo:

```
# Con el comando vector generamos vectores de la siguiente manera  
vector("numeric", 5) # vector(tipo_elemento, no_de_elementos)
```

```
## [1] 0 0 0 0 0
```

Además de `numeric` también podemos generar vectores del tipo `character`, `double`, `integer` y `logical`.

Actividad

- Genere un vector de 10 valores tipo `integer`.

```
vector(_, 10)
```

- Genere un vector de 15 valores tipo `character`

```
vector("character", _)
```

Generación de vectores atómicos

Note que el comando `vector`, solo genera vectores con valores por defecto, por ejemplo para los del tipo `numeric` genera puros 0s. Para generar vectores más interesantes, éstos los podemos generar mediante el comando `c()` que es una abreviación de `combine` o combinar. Por ejemplo, para generar un vector con los nombres de 5 personas que viven en Cancún, lo podemos hacer de la siguiente forma:

```
# Código para generar un vector con los nombres de 5 personas  
personas <- c("Eduardo", "Maria", "Mercedes", "Iván", "Eugenio")  
print(personas) # Con este comando imprimimos el vector personas.
```

```
## [1] "Eduardo" "Maria" "Mercedes" "Iván" "Eugenio"
```

Actividad

Genere los siguientes vectores

1. Un vector con diez números enteros e imprímalos

```
enteros <- c(, , , , , , , , , )
print( )
```

2. Un vector con 5 valores lógicos (TRUE o FALSE)

```
logicos <- c(, , , , )
print(logicos)
```

3. Un vector con 10 nombres de compañeros de la clase.

```
compClase <- c("","","","","","","","","","")
print( )
```

R como calculadora

R, como se vió en sesiones previas, permite realizar cálculos matemáticos tal cual fuese una calculadora. R permite realizar casi cualquier tipo de cálculo desde aritmética básica hasta el cálculo de la derivada de una función. Por ejemplo:

```
# R como calculadora
sin(pi) # Cálculo del seno de pi
```

```
## [1] 1.224647e-16
```

```
sqrt(4) # Cálculo de la raíz cuadrada de 4
```

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

```
min(c(4,9)) # Valor mínimo entre 4 y 9
```

```
## [1] 4
```

Actividades

Hallar las siguientes cantidades usando R

- $\cos(2\pi)$

```
cos(2*_)
```

- $\sqrt{2\pi} / \max(3, 2, 4, 1, 8)$

```
sqrt(*_)/max(c(, , , , , ))
```

- $\log_{10}(10000)$

```
log(, base=10)
```

Generando secuencias en R

R permite generar secuencias de valores con el comando `seq`, el comando `seq` genera secuencias entre un valor mínimo y máximo y puede generar secuencias de acuerdo a cierto incremento (con el parámetro `by=`) o una longitud de valores entre mínimo y máximo (con el parámetro `length=`). Por ejemplo:

```
# Uso del comando seq()
seq(1,6) # Genera valores entre 1 y 6 en intervalos de 1 (por defecto)
```

```
## [1] 1 2 3 4 5 6
```

```
seq(1,2, by=0.1) # Genera valores entre 1 y 2 en incrementos de 0.1

## [1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0

seq(1,20, length=30) # Genera 30 valores entre 1 y 20

## [1] 1.000000 1.655172 2.310345 2.965517 3.620690 4.275862 4.931034
## [8] 5.586207 6.241379 6.896552 7.551724 8.206897 8.862069 9.517241
## [15] 10.172414 10.827586 11.482759 12.137931 12.793103 13.448276 14.103448
## [22] 14.758621 15.413793 16.068966 16.724138 17.379310 18.034483 18.689655
## [29] 19.344828 20.000000
```

Actividades

Generar las siguientes secuencias:

1. Una secuencia entre 0 y π con 10 valores.

```
seq(_,_, length=10)
```

2. Una secuencia de 10 a 8, con incrementos de 0.33

```
seq(10, _, by=_)
```

3. Una secuencia con valores entre 0 y $\log_2(8)$ con 20 valores.

```
# Inserten su código abajo
```

Graficando funciones con R

Una característica muy potente de Res en la realización de gráficos de funciones. Para lo anterior consideraremos que una función puede graficarse a partir de su relación $y = f(t)$ o $y = f(x)$ y los valores de x los podemos generar mediante `c()` o `seq()` y los valores de y aplicando una función a la variable x . Por ejemplo, graficaremos $y = \cos(2\pi t)$ de -2 a 2 .

```
library(basetheme)
basetheme("deepblue")
t <- seq(-2,2, length=200) # Generamos los valores del eje horiz.
y <- cos(2*pi*t)           # Calculamos la función.
plot(t,y, type="l", col="green")
grid()
```

