

Data Science desde 0 - RStudio

Miguel Sencie para DSRP

27/06/2020

“R, una calculadora más”

Algunos operadores...

OPERADOR	DESCRIPCIÓN
+	Suma
-	Resta
*	Multiplicación
/	División
^	Potenciación
%/%	Cociente Entero
%%	Residuo o Módulo
==	Igual
!=	Diferente
>=, <, >, <=	Comparación
=, <-, ->	Asignación
!, &,	Negación, Y, O

y funciones antes de empezar...

FUNCIÓN	SIGNIFICADO
sqrt(x,y)	Raíz y de x
factorial(x)	Factorial de x
log(x)	Logaritmo en base e de x
log10(x)	Logaritmo en base 10 de x
exp(x)	Número e elevado a la x
pi	Número pi
gamma(x)	Función gamma de x
round(x,n)	x redondeado a n cifras decimales
abs(x)	Valor absoluto de x
cos(x), sin(x), tan(x)	Coseno de x, Seno de x, Tangente de x

Calcular las siguientes operaciones:

- a. $2 + 7 + 8$
- b. $5 - 2 - 10$
- c. $25 * 12$
- d. $124/3$

```
2 + 7 + 8
```

```
## [1] 17
```

```
5 - 2 - 10
```

```
## [1] -7
```

```
25 * 12
```

```
## [1] 300
```

```
124/3
```

```
## [1] 41.33333
```

Aumentamos el nivel:

- a. Logaritmo neperiano de 10
- b. 7^2
- c. Raíz cuadrada de 81
- d. Número e y pi

```
log(10)
```

```
## [1] 2.302585
```

```
7^2
```

```
## [1] 49
```

```
81^(1/2); sqrt(81)
```

```
## [1] 9
```

```
## [1] 9
```

```
exp(1); pi
```

```
## [1] 2.718282
```

```
## [1] 3.141593
```

Para casa:

1. Valor absoluto de -3.5
2. Factorial 12
3. Redondear pi a 5 cifras decimales

“Verdad o Falso”

Hacemos las siguientes comparaciones numéricas:

```
1254 > 1687
```

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

```
1487 == 1874
```

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

```
1887 <= 2020
```

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

```
2020 != 2021
```

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

```
round(pi,2) < round(pi, 5)
```

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

Ahora con texto, que sucedería?

```
"M" != "m"
```

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

```
"Texto" == "texto"
```

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

```
"a" > "A"
```

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

```
"A" < "Z"
```

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

“Y si les asignamos nombres?”

```
x = 15
x

## [1] 15

x <- 35
x

## [1] 35

#Tipo de datos más comunes
nombre = "Miguel"
class(nombre)

## [1] "character"

edad = 23
class(edad)

## [1] "numeric"

sexo = T #T: Masculino(1) F: Femenino(0)
class(sexo)

## [1] "logical"

estatura = 1.65
class(estatura)

## [1] "numeric"
```

VECTORES

- Función “c”

```
v1 <- c(1,2,3,4,5)
v1

## [1] 1 2 3 4 5

class(v1)

## [1] "numeric"

v2 <- c("Perú", "Bolivia", "Argentina")
v2

## [1] "Perú"      "Bolivia"   "Argentina"

class(v2)

## [1] "character"
```

```

v3 <- c(T, F, T, T)
v3

## [1] TRUE FALSE TRUE TRUE

class(v3)

## [1] "logical"

c(v1, v2, v3, "DSRP")

## [1] "1" "2" "3" "4" "5" "Perú"
## [7] "Bolivia" "Argentina" "TRUE" "FALSE" "TRUE" "TRUE"
## [13] "DSRP"

```

- Función “rep”

```

rep("a", 15)

## [1] "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a" "a"

rep(v1, times = 5)

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

rep(v1, each = 5)

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

rep(v1, times = c(5,2,7,10,3))

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

```

- Función “seq”

```

seq(0,10,by = 2)

## [1] 0 2 4 6 8 10

seq(30, 10,by = -2)

## [1] 30 28 26 24 22 20 18 16 14 12 10

0:50

## [1] 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
23 24
## [26] 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
48 49
## [51] 50

seq(2, by = 2, length.out = 10)

## [1] 2 4 6 8 10 12 14 16 18 20

```

Operaciones y funciones con vectores

```
x = seq(2, by = 2, length.out = 10)
y = sqrt(x)
```

```
3*x
```

```
## [1] 6 12 18 24 30 36 42 48 54 60
```

```
5+x
```

```
## [1] 7 9 11 13 15 17 19 21 23 25
```

```
x+y
```

```
## [1] 3.414214 6.000000 8.449490 10.828427 13.162278 15.464102 17.741657
## [8] 20.000000 22.242641 24.472136
```

```
x*y
```

```
## [1] 2.828427 8.000000 14.696938 22.627417 31.622777 41.569219 52.383203
## [8] 64.000000 76.367532 89.442719
```

```
length(x) #Longitud del vector x
```

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

```
max(x); min(x) #Máximo y mínimo del vector x
```

```
## [1] 20
```

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

```
sum(x) #Suma de las entradas del vector
```

```
## [1] 110
```

```
mean(x); sum(x)/length(x) #Media del vector x
```

```
## [1] 11
```

```
## [1] 11
```

```
sort(x); rev(sort(x)) #Ordena las entradas del vector x
```

```
## [1] 2 4 6 8 10 12 14 16 18 20
```

```
## [1] 20 18 16 14 12 10 8 6 4 2
```

```
cumsum(x) #Suma acumulada de las entradas del vector x
```

```
## [1] 2 6 12 20 30 42 56 72 90 110
```

MATRICES

- Usando la función “matrix”

```
x = 1:20
```

```
matrix(x,nrow = 5) #Entradas ordenadas por columnas
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    6   11   16
## [2,]    2    7   12   17
## [3,]    3    8   13   18
## [4,]    4    9   14   19
## [5,]    5   10   15   20
```

```
matrix(x, nrow = 5, byrow = TRUE) #Entradas ordenadas por filas
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    5    6    7    8
## [3,]    9   10   11   12
## [4,]   13   14   15   16
## [5,]   17   18   19   20
```

```
matrix(1, nrow = 3, ncol = 4)
```

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

- Usando la función “cbind” y “rbind”

```
a = 1:5
```

```
b = rep(1,5)
```

```
c = a^2
```

```
cbind(a,b,c) #Por columnas
```

```
##      a b  c
## [1,] 1 1  1
## [2,] 2 1  4
## [3,] 3 1  9
## [4,] 4 1 16
## [5,] 5 1 25
```

```
rbind(a,b,c) #Por filas
```

```
##      [,1] [,2] [,3] [,4] [,5]
## a      1    2    3    4    5
## b      1    1    1    1    1
## c      1    4    9   16   25
```

Operaciones y funciones con matrices

```
A = rbind(a,b,c)
B = cbind(a,b,c)
C = matrix(1:15, nrow = 3, byrow = TRUE)
```

```
7*B
```

```
##      a b  c
## [1,] 7 7  7
## [2,] 14 7 28
## [3,] 21 7 63
## [4,] 28 7 112
## [5,] 35 7 175
```

```
A*C #Multiplicación entrada por entrada de cada matriz
```

```
##  [,1] [,2] [,3] [,4] [,5]
## a    1    4    9   16   25
## b    6    7    8    9   10
## c   11   48  117  224  375
```

```
D = A%*%B #Multiplicación de matrices
D
```

```
##      a b  c
## a   55 15 225
## b   15  5  55
## c  225 55 979
```

```
t(B)
```

```
##  [,1] [,2] [,3] [,4] [,5]
## a    1    2    3    4    5
## b    1    1    1    1    1
## c    1    4    9   16   25
```

```
det(D) #Solo matriz cuadrada
```

```
## [1] 700
```

```
solve(D) #Solo matriz cuadrada e invertible
```

```
##      a      b      c
## a  2.6714286 -3.3 -0.42857143
## b -3.3000000  4.6  0.50000000
## c -0.4285714  0.5  0.07142857
```

```
qr(D)$rank
```

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



```

diag(D)

##      a      b      c
##  55      5  979

diag(1, nrow = 5)

##      [,1] [,2] [,3] [,4] [,5]
## [1,]      1      0      0      0      0
## [2,]      0      1      0      0      0
## [3,]      0      0      1      0      0
## [4,]      0      0      0      1      0
## [5,]      0      0      0      0      1

ncol(A) #Número de columnas de la matriz

## [1] 5

nrow(A) #Número de filas de la matriz

## [1] 3

dim(A) #Dimensiones de la matriz (fila, columna)

## [1] 3 5

colMeans(A) #Medias de las columnas de la matriz

## [1] 1.000000 2.333333 4.333333 7.000000 10.333333

rowSums(A) #Suma de las filas de la matriz

##      a      b      c
##  15      5  55

```