Guías prácticas

Erika Martínez

November 27, 2015

1.1VECTORES NUMERICOS Crear un vector numerico vacio y añadirle luego sus elementos

```
v <- numeric(3);v

## [1] 0 0 0

v[3] <- 17; v

## [1]  0  0 17
```

Crear un vector numérico asignándole todos sus elementos o valores

```
x <- c(2, 4, 3.1, 8, 6)
x<- edit(x)
x

## [1] 2.0 4.0 3.1 8.0 6.0

is.integer(x)

## [1] FALSE

is.double(x)

## [1] TRUE

length(x)

## [1] 5
```

Crear un vector numérico dando un rango de valores

```
y = 1:4; y

## [1] 1 2 3 4

y[2] <- 5
u <- 1:12; u1=u[2 * 1:5]
```

Crear un vector numérico utilizando la función assign()

```
assign("z", c(x, 0, x)); z

##  [1] 2.0 4.0 3.1 8.0 6.0 0.0 2.0 4.0 3.1 8.0 6.0
```

Crear un vector numérico generando una sucesión de valores

```
s1 <- seq(2, 10); s1

## [1]  2  3  4  5  6  7  8  9 10

s2 = seq(from=-1, to=5); s2
```

```
## [1] -1  0  1  2  3  4  5

s3<-seq(to=2, from=-2); s3

## [1] -2 -1  0  1  2

s4=seq(from=-3, to=3, by=0.2); s4

##  [1] -3.0 -2.8 -2.6 -2.4 -2.2 -2.0 -1.8 -1.6 -1.4 -1.2 -1.0 -0.8 -0.6 -0.4
## [15] -0.2  0.0  0.2  0.4  0.6  0.8  1.0  1.2  1.4  1.6  1.8  2.0  2.2  2.4
## [29]  2.6  2.8  3.0

s5 <- rep(s3, times=3); s5

##  [1] -2 -1  0  1  2 -2 -1  0  1  2 -2 -1  0  1  2
```

1.1.1OPERACIONES CON VECTORES NUMERICOS

```
x=8
1/x

## [1] 0.125

v=2*x+z+1; v

##  [1] 19.0 21.0 20.1 25.0 23.0 17.0 19.0 21.0 20.1 25.0 23.0

e1 <- c(1, 2, 3, 4); e2<-c(4, 5, 6, 7); crossprod(e1, e2)

##      [,1]
## [1,]   60

t(e1)%*%e2

##      [,1]
## [1,]   60
```

OPERACIONES DE FUNCIONES SOBRE VECTORES NUMERICOS

```
xt<- t(x); xt

##      [,1]
## [1,]    8

u = exp(y);u

## [1]   2.718282 148.413159  20.085537  54.598150

options(digits=10); u

## [1]   2.718281828 148.413159103  20.085536923  54.598150033
```

OTRAS OPERACIONES

```
resum <- c(length(y), sum(y), prod(y), min(y), max(y)); resum

## [1]  4 13 60  1  5

yo <- sort(y); yo

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

1.2 VECTORES DE CARACTERES

```
S<-character()
S

## character(0)
```

FORMA 4-Crear una matriz a partir de la unión de vectores

```
x1 <- seq(0, 10, 2); x1

## [1]  0  2  4  6  8 10

x2 <- seq(1, 11, 2); x2

## [1]  1  3  5  7  9 11

x3 <- runif(6); x3

## [1] 0.49822225748 0.06834137882 0.01149970759 0.48312492459 0.46736010094
## [6] 0.49631722132

Xcol <- cbind(x1, x2, x3); Xcol

##      x1 x2           x3
## [1,]  0  1 0.49822225748
## [2,]  2  3 0.06834137882
## [3,]  4  5 0.01149970759
## [4,]  6  7 0.48312492459
## [5,]  8  9 0.46736010094
## [6,] 10 11 0.49631722132

Xfil <- rbind(x1, x2, x3); Xfil

##            [,1]          [,2]          [,3]         [,4]         [,5]
## x1 0.0000000000 2.00000000000 4.00000000000 6.0000000000 8.0000000000
## x2 1.0000000000 3.00000000000 5.00000000000 7.0000000000 9.0000000000
## x3 0.4982222575 0.06834137882 0.01149970759 0.4831249246 0.4673601009
##             [,6]
## x1 10.0000000000
## x2 11.0000000000
## x3  0.4963172213

X <- Xfil[1:3, c(2, 3)]; X

##             [,1]          [,2]
## x1 2.00000000000 4.00000000000
## x2 3.00000000000 5.00000000000
## x3 0.06834137882 0.01149970759
```

FORMA 2-Crear un vector de caracteres asignándole todos sus elementos

```
deptos <- c("Santa Ana", "Sonsonate", "San Salvador"); deptos

## [1] "Santa Ana"    "Sonsonate"    "San Salvador"

deptos[4]="AhuachapÃ¡n"; deptos

## [1] "Santa Ana"    "Sonsonate"    "San Salvador" "AhuachapÃ¡n"
```

2.CREACIÓN Y MANEJO DE MATRICES. 2.1CREACIÓN DE MATRICES NUMÉRICAS.

FORMA 1-Crear una matriz numérica vacía y añadirle luego sus elementos.

```
M <- matrix(numeric(), nrow = 3, ncol=4);M

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

M[2,3] <- 6; M

##      [,1] [,2] [,3] [,4]
## [1,]   NA   NA   NA   NA
## [2,]   NA   NA    6   NA
## [3,]   NA   NA   NA   NA
```

FORMA 2-Crear una matriz numérica asignándole todos sus elementos o valores.

```
A <- matrix(c(2, 4, 6, 8, 10, 12), nrow=2, ncol=3); A

##      [,1] [,2] [,3]
## [1,]    2    6   10
## [2,]    4    8   12

mode(A)

## [1] "numeric"

dim(A)

## [1] 2 3

attributes(A)

## $dim
## [1] 2 3

is.matrix(A)

## [1] TRUE

is.array(A)

## [1] TRUE
```

FORMA 3-Crear una matriz numérica dando un rango de valores

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

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

2.2OPERACIONES CON MATRICES NUMÉRICAS. MULTIPLICACION DE MATRICES MATRICES NUMERICAS:

```
v<-c(1, 2); v %*%A
```

```
##      [,1] [,2] [,3]
## [1,]   10   22   34
```

```
P<- A %*% B; P
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   44   98  152  206
## [2,]   56  128  200  272
```

```
2*A
```

```
##      [,1] [,2] [,3]
## [1,]    4   12   20
## [2,]    8   16   24
```

OPERACIONES DE FUNCIONES SOBRE MATRICES NUMÉRICAS:

```
length(A)
```

```
## [1] 6
```

```
T=sqrt(B); T
```

```
##             [,1]        [,2]        [,3]        [,4]
## [1,] 1.000000000 2.000000000 2.645751311 3.162277660
## [2,] 1.414213562 2.236067977 2.828427125 3.316624790
## [3,] 1.732050808 2.449489743 3.000000000 3.464101615
```

```
t(A)
```

```
##      [,1] [,2]
## [1,]    2    4
## [2,]    6    8
## [3,]   10   12
```

```
C <- matrix(c(2, 1, 10, 12), nrow=2, ncol=2); C
```

```
##      [,1] [,2]
## [1,]    2   10
## [2,]    1   12
```

```r
det(C)
```

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

```r
InvC <- solve(C) ; InvC
```

```
##                [,1]          [,2]
## [1,]  0.85714285714 -0.7142857143
## [2,] -0.07142857143  0.1428571429
```

```r
b=diag(2); InvC<-solve(C, b); InvC
```

```
##                [,1]          [,2]
## [1,]  0.85714285714 -0.7142857143
## [2,] -0.07142857143  0.1428571429
```

```r
eigen(C)
```

```
## $values
## [1] 12.916079783  1.083920217
##
## $vectors
##                [,1]           [,2]
## [1,] -0.6754894393 -0.99583021557
## [2,] -0.7373696613  0.09122599279
```

```r
diag(C)
```

```
## [1]  2 12
```

```r
diag(u1)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    2    0    0    0    0
## [2,]    0    4    0    0    0
## [3,]    0    0    6    0    0
## [4,]    0    0    0    8    0
## [5,]    0    0    0    0   10
```

```r
diag(3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

OTRAS OPERACIONES:

```r
c(length(A), sum(A), prod(A), min(A), max(A))
```

```
## [1]     6    42 46080     2    12
```

```r
O <- matrix(sort(C), nrow=2, ncol=2); O
```

```
##      [,1] [,2]
## [1,]    1   10
## [2,]    2   12
```

2.3 CREACIÓN DE UNA MATRIZ DE CADENAS

```
nombres <- matrix(c("Carlos", "José", "Ana", "Rení", "Maria", "Mario"), nrow=3, ncol=2)
nombres
```

```
##      [,1]     [,2]
## [1,] "Carlos" "Rení"
## [2,] "José"   "Maria"
## [3,] "Ana"    "Mario"
```

3.CREACIÓN Y MANEJO DE MATRICES INDEXADAS (ARRAY)

```
X <- array(c(1, 3, 5, 7, 9, 11), dim=c(2, 3)); X
```

```
##      [,1] [,2] [,3]
## [1,]    1    5    9
## [2,]    3    7   11
```

```
Z <- array(1, c(3, 3)); Z
```

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

```
W <- 2*Z+1; W
```

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

```
TX <- t(X); TX
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    5    7
## [3,]    9   11
```

```
a <- c(2, 4, 6); a
```

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

```
b <- 1:3;b
```

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

```
M <- a %o% b; M
```

```
##      [,1] [,2] [,3]
## [1,]    2    4    6
## [2,]    4    8   12
## [3,]    6   12   18
```

```
c <- a * b; c
```

```
## [1]  2  8 18
```

```
Arreglo3 <- array(c(1:8, 11:18, 111:118), dim = c(2, 4, 3))
Arreglo3
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,]    1    3    5    7
## [2,]    2    4    6    8
##
## , , 2
##
##      [,1] [,2] [,3] [,4]
## [1,]   11   13   15   17
## [2,]   12   14   16   18
##
## , , 3
##
##      [,1] [,2] [,3] [,4]
## [1,]  111  113  115  117
## [2,]  112  114  116  118
```