

1 CREACIÓN Y MANEJO DE VECTORES DE DATOS.

Este tipo de objetos se denominan estructuras atómicas ya que todos sus elementos son del mismo tipo o modo: character (carácter) o numeric (numérico) que puede ser integer (entero), double (real), complex (complejo), logical (lógico).

1.1 VECTORES NUMÉRICOS

FORMA 1 - Crear un vector numérico vacío y añadirle luego sus elementos.

- *Ejemplo 1:*

```
v = numeric(3);  
v  
  
## [1] 0 0 0  
  
# el vector tiene longitud 3 y sus componentes serán NA (Not  
#Available /"Missing" Values) que es la forma como R maneja  
#los datos omitidos o faltantes.
```

- *Ejemplo 2:*

```
v[3] <- 17;  
v # Asigna el valor de 17 en la tercera posicion del vector v.  
  
## [1] 0 0 17
```

FORMA 2 - Crear un vector numérico asignándole todos sus elementos o valores.

-*Ejemplo 1:*

```
x <- c(2,4,3.1,8,6)  
x #Mostrar el vector  
  
## [1] 2.0 4.0 3.1 8.0 6.0  
  
is.integer(x) #¿El vector contiene datos tipo enteros?  
  
## [1] FALSE
```

```
is.double(x) #¿ El vector contiene datos de tipo doubles?
## [1] TRUE

length(x) # Tamaño del vector
## [1] 5
```

- *Ejemplo 2:* Modifique el vector agregándole el valor 9 en la posición 3.

```
#x <- edit(x) #Este comando muestra una ventana para
#agregar el valor en la posición en donde se requiera
x
## [1] 2.0 4.0 3.1 8.0 6.0
```

FORMA 3 - Crear un vector numérico dando un rango de valores.

- *Ejemplo 1:*

```
y <- 1:4; # crea un vector de valores enteros en
#que su primer elemento es 1 su último es 4
y
## [1] 1 2 3 4
```

- *Ejemplo 2:* Modificación de los elementos de un vector, (para modificar un elemento de un vector se escribe su nombre (del vector) y entre corchetes el índice del elemento que se quiera modificar).

```
y[2] <- 5
y
## [1] 1 5 3 4
```

- *Ejemplo 3:* Crear un vector con elementos de otro; (vector de tamaño 5 con elementos de las posiciones pares de u)

```
u <- 1:12;
u
```

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

u1<- u[2*1:5]
u1

## [1] 2 4 6 8 10
```

FORMA 4 - Crear un vector numérico utilizando la función `assign()`.

- *Ejemplo 1:* `z` (crea un vector en dos copias de `x` con un cero entre ambas)

```
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
```

FORMA 5 - Crear un vector numérico generando una sucesión de valores.

- *Ejemplo 1:* (compárese a como fue generado el vector `y` y `u`)

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

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

- *Ejemplo 2:*

```
s2 <- seq(from=-1, to=5);
s2 #Crea un vector cuyo elemento inicial es -1 y su elemento

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

#final es 5, y cada dos elementos consecutivos del vector
#tienen una diferencia de una unidad.
```

- *Ejemplo 3:*

```
s3 <- seq(to=2, from=-2); # note que puede invertir el orden
#de \to" y de \from".
s3

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

- *Ejemplo 4:* Secuencia con incremento o decremento:

```
s4 <-seq(from=-3, to=3, by=0.2); # crea una secuencia que
#inicia en -3 y termina en 3 con incrementos de 0.2 en 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 -0.2
## [16]  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  2.6  2.8
## [31]  3.0
```

- *Ejemplo 5:* Repetición de una secuencia.

```
s5 <-rep(s3, times=3); #repite el vector s3, tres veces consecutivas
s5

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

1.1.1 OPERACIONES CON VECTORES NUMÉRICOS.

- *Ejemplo 1:* $1/x$ (observe que calcula el inverso de cada elemento del vector)

```
x

## [1] 2.0 4.0 3.1 8.0 6.0

1/x

## [1] 0.5000000 0.2500000 0.3225806 0.1250000 0.1666667
```

- *Ejemplo 2:* $v=2*x+z+1$; v (genera un nuevo vector, v , de longitud 11, construido sumando, elemento a elemento, el vector $2*x$ repetido 2.2 veces, el vector y , y el número 1 repetido 11 veces “Reciclado en R es repetir las veces necesarias un vector cuando en una operación intervienen vectores de distinta longitud”).

```
v=2*x+z+1;

## Warning in 2 * x + z: longitud de objeto mayor no es múltiplo de
la longitud de uno menor

v

## [1] 7.0 13.0 10.3 25.0 19.0 5.0 11.0 11.2 20.1 21.0 11.0
```

- *Ejemplo 3:* (calcula el producto interno entre dos vectores. Ambos deben tener el mismo número de elementos).

```
e1 <- c(1,2,3,4);  
e2 <- c(4,5,6,7);  
crossprod(e1,e2)  
  
##      [,1]  
## [1,]   60  
  
t(e1)%*%e2  
  
##      [,1]  
## [1,]   60
```

1.1.2 OPERACIONES DE FUNCIONES SOBRE VECTORES NUMÉRICOS.

- *Ejemplo 1:* Vector transpuesto del vector x: $x_t = t(x)$; x_t .

```
x  
  
## [1] 2.0 4.0 3.1 8.0 6.0  
  
xt=t(x);  
xt  
  
##      [,1] [,2] [,3] [,4] [,5]  
## [1,]    2    4  3.1    8    6
```

- *Ejemplo 2:* (crea un nuevo vector de la misma longitud que y, en el cual cada elemento es la exponencial elevando a su respectivo elemento en y).

```
u=exp(y);  
u  
  
## [1] 2.718282 148.413159 20.085537 54.598150
```

- options(digits=10); u # Permite visualizar un mínimo de 10 dígitos

```
options(digits = 10);  
u  
  
## [1] 2.718281828 148.413159103 20.085536923 54.598150033
```

OTRAS OPERACIONES:

- *Ejemplo 1:*

```
resun <- c(length(y), sum(y), prod(y), min(y), max(y));  
resun # tamaño, suma, producto, mínimo y máximo. todo del vector y  
## [1] 4 13 60 1 5
```

- *Ejemplo 2:* Ordenamiento de un vector.

```
yo <- sort(y);  
yo  
## [1] 1 3 4 5
```

1.2 VECTORES DE CARACTERES.

FORMA 1 - Crear un vector de caracteres vacío y añadirle luego sus elementos.

- *Ejemplo 1:*

```
s <-character()  
s  
## character(0)
```

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

- *Ejemplo 1:* Crear el vector de caracteres:

```
deptos <- c("Santa Ana", "Sonsonate", "san Salvador");  
deptos  
## [1] "Santa Ana" "Sonsonate" "san Salvador"
```

- *Ejemplo 2:* Agregue el elemento "Ahuachapán" en la cuarta posición. (R Permite incrementar el tamaño del vector en cualquier instante).

```
deptos[4]="Ahuachapan";  
deptos  
  
## [1] "Santa Ana"      "Sonsonate"      "san Salvador" "Ahuachapan"
```