

Complementos

Sesión 2

Natalie Julian - www.nataliejulian.com

Estadística UC y Data Scientist en Zippedi Inc.

¿Por qué es importante aprender sobre vectores en R?

Los vectores son importantes pues corresponden a la unidad elemental de las matrices, a partir de vectores podremos crear matrices y otras estructuras más complejas.

Estructura de un vector

La estructura de un vector en R es bastante particular. Debemos anteponer una letra `c` y luego abrir paréntesis. Dentro de los paréntesis debemos escribir los elementos del vector separados por comas:

```
c(elemento1, elemento2, elemento3, ....)
```

Podemos crear vectores de números o vectores de texto/caracteres. Por ejemplo:

```
c(2,4,-1,1/6,9,10,5, log(10)) #vector de números o vector numerico
```

```
c("Hola", "Que tal", "Mis saludos") #vector de texto o caracteres
```

Tipos de vectores

Curiosamente, es posible escribir un vector de números pero en formato de texto o caracteres, por ejemplo:

```
c("2", "4", "-1", "1/6", "9", "10", "5")    #Vector en formato character
```

Es un vector que contiene números pero será leído en formato de texto.

¿Por qué es importante saber en qué formato se lee un vector?

Es importante conocer el tipo o formato del vector porque R posee funciones que solo pueden aplicarse a vectores de tipo numérico y otras a vectores de tipo carácter. Por ejemplo, si estuviéramos interesados en calcular un promedio note que a un vector numérico se le puede calcular de manera sencilla pero a un vector de caracteres no:

```
numeros<-c(22.5, 87.3, 43.2, 95/2) #Se le asigna al objeto numeros el vector
```

```
mean(numeros)
[1] 50.125
```

```
numeros2<-c("22.5", "87.3", "43.2", "95/2") #vector de caracteres
```

```
mean(numeros2)
[1] NA
```

Warning message:

In mean.default(numeros2) :

argument is not numeric or logical: returning NA

¿Cómo saber el formato de un vector en R?

Si queremos saber el formato de un vector, podemos utilizar la función `class()`, por ejemplo, note que:

```
numeros #Muestra el vector asociado al objeto numeros  
[1] 22.5 87.3 43.2 47.5
```

```
class(numeros) #Muestra el tipo de vector que es numeros  
[1] "numeric"
```

El vector `numeros` que recién creamos es de tipo `numeric`, es decir, sus elementos son números. ¿Qué pasa con el vector `numeros2`?:

```
numeros2 #Muestra el vector asociado al objeto numeros2  
[1] "22.5" "87.3" "43.2" "95/2"
```

```
class(numeros2) #Muestra el tipo de vector que es numeros2  
[1] "character"
```

Es de tipo `character`, sus elementos se leen en formato de caracteres.

Observación

Además, note que cuando se crearon los objetos `numeros` y `numeros2` aparecen aquí:

The screenshot shows the RStudio interface. The script editor on the left contains the following R code:

```
15  
16  
17 numeros #Muestra el vector asociado al objeto numeros  
18 class(numeros) #Muestra el tipo de vector que es numeros  
19  
20 numeros2 #Muestra el vector asociado al objeto numeros2  
21 class(numeros2) #Muestra el tipo de vector que es numeros2  
22
```

The console at the bottom shows the output of the code:

```
> class(numeros)  
[1] "numeric"  
> numeros  
[1] 22.5 87.3 43.2 47.5  
> class(numeros)  
[1] "numeric"  
> numeros #Muestra el vector asociado al objeto numeros  
[1] 22.5 87.3 43.2 47.5  
> class(numeros) #Muestra el tipo de vector que es numeros  
[1] "numeric"  
> numeros2 #Muestra el vector asociado al objeto numeros2  
[1] "22.5" "87.3" "43.2" "95/2"  
> class(numeros2) #Muestra el tipo de vector que es numeros2  
[1] "character"
```

The Environment pane on the right shows the objects created in the Global Environment:

Values	
numeros	num [1:4] 22.5 87.3 43.2 47.5
numeros2	chr [1:4] "22.5" "87.3" "43.2" "95/2"

Two orange arrows point from the text annotations to the Environment pane. The first arrow points to the header area of the Environment pane, and the second arrow points to the data rows.

Cuando creamos objetos, siempre aparecerán aquí

Aquí aparece el formato de cada objeto

- Cree objetos con los siguientes vectores en R (es decir, asígneles un nombre a los vectores). ¿De qué tipo es cada uno?

```
seq(-1,20, len=20)
```

```
rep(c("Hola", "Bienvenida"), c(3,2))
```

```
rep(c("2", "-1"), 3)
```

```
c(2,-1, 2, -1, 2, -1)
```

- ¿De qué tipo crees que será el siguiente vector? ¿Por qué?

```
c("Universidad", "Catolica", 2020)
```


PRACTICA EN R

Cambiar el formato de un vector

Supongamos que tenemos el siguiente vector:

```
(cantidades<-c("2", "45", "63.3", "31", "90"))  
[1] "2"      "45"     "63.3"   "31"     "90"
```

Y, evidentemente, su formato es de tipo `character` pues utilizamos comillas al enumerar cada elemento:

```
class(cantidades)  
[1] "character"
```

¿Qué pasa si quiero que dicho vector se lea en formato numérico? Podemos utilizar la función `as.numeric()`:

```
as.numeric(cantidades)  
[1]  2.0 45.0 63.3 31.0 90.0
```

Esto modifica el formato de lectura (cuando es posible) a numérico. Para modificar el vector `cantidades` basta con realizar la asignación nuevamente:

```
(cantidades<-as.numeric(cantidades))  
[1]  2.0 45.0 63.3 31.0 90.0
```

¿QUÉ PUEDO HACER CON VECTORES DE TIPO
NUMERIC EN R?

Operaciones matemáticas

¡Todas las operaciones matemáticas de R son aplicables a vectores numéricos!. Por ejemplo:

```
exp(1:10)
```

```
[1]      2.718282      7.389056     20.085537     54.598150    148.413159  
[6]   403.428793  1096.633158  2980.957987  8103.083928 22026.465795
```

```
tan(seq(pi, 2*pi, by=0.5))
```

```
[1] -1.224647e-16  5.463025e-01  1.557408e+00  1.410142e+01 -2.185040e+00  
[6] -7.470223e-01 -1.425465e-01
```

```
prod(1:5)
```

```
[1] 120
```

```
c(3,-3)**2
```

```
[1] 9 9
```

A cada elemento del vector se le aplica la operación.

Practica 2

Cree un vector llamado `distancias` que contenga las siguientes distancias:

24.3 29.67 30.1 34.9 39.1 40.45 12.8 27.0 89.6 45.003 90.6

Copia las siguientes líneas de código y comenta qué realiza cada una:

```
length(distancias)
class(distancias)
which.max(distancias)
distancias[which.max(distancias)]
which.min(distancias)
distancias[which.min(distancias)]
distancias[c(4,5,6)]
order(distancias)
distancias[order(distancias)]
round(distancias, 1)
```

PRACTICA EN R

RESPUESTAS PRÁCTICA

Respuestas práctica 1

```
(vector1<-seq(-1,20, len=20)) #Si encerramos en paréntesis una asignación, muestra inmediatamente el objeto creado
```

```
[1] -1.0000000 0.1052632 1.2105263 2.3157895 3.4210526 4.5263158  
[7] 5.6315789 6.7368421 7.8421053 8.9473684 10.0526316 11.1578947  
[13] 12.2631579 13.3684211 14.4736842 15.5789474 16.6842105 17.7894737  
[19] 18.8947368 20.0000000
```

```
class(vector1)
```

```
[1] "numeric"
```

```
(vector2<-rep(c("Hola", "Bienvenida"), c(3,2)))
```

```
[1] "Hola"      "Hola"      "Hola"      "Bienvenida" "Bienvenida"
```

```
class(vector2)
```

```
[1] "character"
```

```
(vector3<-rep(c("2", "-1"), 3))
```

```
[1] "2"  "-1" "2"  "-1" "2"  "-1"
```

```
class(vector3)
```

```
[1] "character"
```

```
(vector4<-c(2,-1, 2, -1, 2, -1))
```

```
[1] 2 -1 2 -1 2 -1
```

```
class(vector4)
```

```
[1] "numeric"
```


Respuestas práctica 1

El vector `c("Universidad", "Catolica", 2020)` posee caracteres y números. ¿De qué tipo será? ¿numeric o character?

Utilicemos la función `class()` en el vector (note que no es necesario definir el vector, podemos evaluarlo directamente. Se recomienda definir objetos cuando se utilicen más de una vez).

```
class(c("Universidad", "Catolica", 2020))  
[1] "character"
```

El vector es de tipo `character`. Un vector siempre tiene elementos del mismo tipo, por lo que, R trata de que todos los elementos del vector se lean con el mismo formato, por lo que lo más natural es que sea todo en formato carácter (ya que es imposible que una palabra se lea en formato numérico). Cada vez que tengamos vectores mixtos el formato será de tipo `character`.

Respuestas práctica 2

```
length(distancias) #Entrega la cantidad de distancias que contiene el vector distancias
class(distancias) #Indica el tipo de vector que es el vector distancias
which.max(distancias) #Indica la posición del vector distancias en la que se tiene la mayor distancia
distancias[which.max(distancias)] #Extrae la máxima distancia del vector distancias
which.min(distancias) #Indica la posición del vector distancias en la que se tiene la menor distancia
distancias[which.min(distancias)] #Extrae la mínima distancia del vector distancias
distancias[c(4,5,6)] #Extrae los elementos 4, 5 y 6 del vector distancias
order(distancias) #Entrega un vector con las posiciones para ordenar el vector distancias
distancias[order(distancias)] #Entrega el vector distancias ordenado
round(distancias, 1) #Redondea el vector distancias al primer decimal
```