

Objetos de datos en R

Técnicamente hablando, R es un lenguaje de expresiones con una sintaxis simple. La información que manipulamos en R se crea a partir de comandos y se estructura en forma de objetos. Por tal motivo, trabajar con R implica conocer los principales tipos de objetos y sus propiedades básicas.

o Vector

Un vector es el tipo de datos fundamental de R, es una colección ordenada de elementos de un mismo tipo. Se crea utilizando la función de combinación `c()`, pudiendo contener como argumento un número determinado de elementos vectoriales separados por una coma, el valor del vector se obtendrá mediante la concatenación de todos sus elementos, ej.

```
#Crear los siguientes vectores
x <- c(1, 10, 49)           #v es un vector numérico
x <- c(2L, 10L, 9L)         #x es un vector numérico
y <- c("a", "b", "c")       #y es un vector de caracteres
z <- c(TRUE, FALSE, TRUE)   #z es un vector booleano
a <- c(1+2i, 2+9i)          #a es un vector complejo
b <- 6                      #b es un vector numérico de un elemento
```

o Factor

Factor es un término que hace referencia a un tipo de datos estadísticos utilizado para almacenar variables nominales o categóricas. Los factores son valores de naturaleza no numérica, que corresponden a categorías como Alto, Medio y Bajo; sin embargo, en R, pueden codificarse usando números. Por ejemplo, "x" es un vector que tiene 5 observaciones de dos niveles Alto y Bajo.

```
#Dado el vector de caracteres "x"
```

```
> x <- c("Alto", "Bajo", "Bajo", "Bajo", "Alto")
> factor(x)
[1] Alto Bajo Bajo Bajo Alto
Levels: Alto Bajo
```

```
#Factor a partir de un vector numérico
> v = c(3,1,2,2,1) #Crea el vector "v"
> fv = factor(v)   #Crea el factor "fv" a partir de "v".
> fv
[1] 3 1 2 2 1
Levels: 1 2 3
```

```
#Asignar etiqueta a los niveles del factor
```

```
> lv = factor(v, labels = c("Alto", "Medio", "Bajo"))
> lv
[1] Bajo Alto Medio Medio Alto
Levels: Alto Medio Bajo
```

o Matriz

Las matrices son casos especiales de un tipo de objeto R más general llamados *arrays*. Los *arrays* pueden ser multidimensionales. Por ejemplo, una matriz tridimensional constituida por filas, columnas y capas. Nosotros abordaremos a una matriz como una colección de elementos del mismo tipo de datos (numéricos, de caracteres o lógicos) dispuestos en un número fijo de filas y columnas.

En R una matriz se construye usando la función `matrix()`. Replique los ejemplos mostrados a continuación y observe las diferencias.

```
> matrix(1: 9, byrow = TRUE, nrow = 3)
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

```
> matrix(1: 9, byrow = FALSE, nrow = 3)
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

El argumento `byrow` indica que la matriz se completa por las filas. Si deseamos que la matriz sea llenada por columnas, simplemente colocamos `byrow = FALSE` que es el argumento por defecto.

Como hemos podido observar, una matriz se construye a partir de un vector, la función `matrix()` recibe un vector como argumento y lo transforma en una matriz de acuerdo a las dimensiones especificadas, por tanto se puede construir de la siguiente manera:

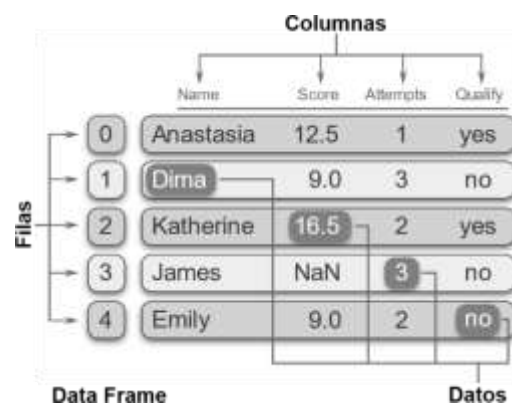
```
> v <- 1:15
> v
[1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
> matriz_v5x3 <- matrix(v, byrow = TRUE, nrow = 5)
> matriz_v5x3
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
[4,]   10   11   12
[5,]   13   14   15
> matriz_v3x5 <- matrix(v, byrow = TRUE, nrow = 3)
> matriz_v3x5
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    6    7    8    9   10
[3,]   11   12   13   14   15
```

La dimensión de una matriz se obtiene con el comando `dim()` cuyo resultado es un vector de dos elementos, el primero informa el número de filas y el segundo elemento se refiere al número de columnas.

```
> dim(matriz_v3x5)
[1] 3 5
```

○ Dataframe

Analizar datos implica manejar información cualitativa y cuantitativa. En R, un *data frame* (o cuadro de datos) es una estructura de datos que permite almacenar información de diferentes tipos. Un *data frame* es como una matriz, con una estructura bidimensional de filas y columnas, con la particularidad que cada columna puede almacenar diferentes tipos de datos.



Un *data frame* se crea con la función `data.frame()`, los argumentos de esta función lo conforman las columnas que constituirán el cuadro de datos

Para crear un *data frame* partiremos por la creación de vectores de la misma longitud.

```
> #vectores que formarán las columnas del data frame
> int <- c(1L, 2L, 3L, 4L)
> int
[1] 1 2 3 4
```

```
> texto <- c('cafe', 'maíz', 'soya', 'arroz')
> texto
[1] "cafe" "maíz" "soya" "arroz"
```

```
> num <- c(23.5, 40, 52.2, 15.72)
> num
[1] 23.50 40.00 52.20 15.72
> logico <- c(FALSE, T, F, TRUE)
> logico
[1] FALSE TRUE FALSE TRUE
```

```
> #Construcción del data frame
> df <- data.frame(int, texto, num, logico)
> df
  int texto  num logico
1   1  cafe 23.50  FALSE
2   2  maíz 40.00   TRUE
3   3  soya 52.20  FALSE
4   4 arroz 15.72   TRUE
```

En el ejemplo se ha creado el *data frame* `df` con cuatro columnas con diferentes tipos de datos.

Para consultar la estructura de datos de nuestro *data frame* se hace uso de la función `str()`, que muestra la estructura del conjunto de datos. Dicha función genera la siguiente información: el número total de observaciones (filas), el número total de variables (columnas), una lista completa de los nombres de las variables, el tipo de datos de cada variable y los valores de las primeras observaciones de cada variable.

```
> str(df)
'data.frame':    4 obs. of  4 variables:
 $ int      : int   1 2 3 4
 $ texto    : chr   "cafe" "maíz" "soya" "arroz"
 $ num      : num   23.5 40 52.2 15.7
 $ logico: logi   FALSE TRUE FALSE TRUE
```

Para explorar nuestro *data frame* también podemos hacer uso de la función `summary()` la cual permite conocer medidas estadísticas básicas de las variables que componen nuestro conjunto de datos.

```
> summary(df)
      int      texto      num
Min.   :1.00  Length:4  Min.   :15.72
1st Qu.:1.75  Class  :character 1st Qu.:21.55
Median :2.50  Mode   :character Median :31.75
Mean   :2.50                      Mean   :32.85
3rd Qu.:3.25                      3rd Qu.:43.05
Max.   :4.00                      Max.   :52.20
      logico
Mode :logical
FALSE:2
TRUE :2
```

○ Lista

Las listas, a diferencia de los otros tipos de objetos de R, pueden contener componentes de diferentes tipos. Una lista en R le permite almacenar una variedad de objetos bajo un mismo nombre de una manera ordenada. Estos objetos pueden ser vectores, matrices, *data frames*, incluso otras listas, etc.; en una lista se puede almacenar prácticamente cualquier información, por lo que son consideradas como contenedores generales de datos.

Para crear una lista a partir de un conjunto de objetos pre-existente utilizamos la función `list()`. Los argumentos esta función son los componentes de la lista.

```
> vector_num <- 1:10
> vector_pal <- c("universidad", "ciudad", "sector")
> matriz <- matrix(c(1:9), nrow = 3)
> sub_lista <- list(vector_num, vector_pal
> lista <- list(vector_num, vector_pal, matriz, sub_lista)
> lista
[[1]]
 [1]  1  2  3  4  5  6  7  8  9 10

[[2]]
[1] "universidad" "ciudad"      "sector"

[[3]]
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

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

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

```
[[4]][[2]]
```

```
[1] "universidad" "ciudad" "sector"
```

Si desea nombrar sus listas después de crearlas, puede utilizar la función `names()`. Veamos un ejemplo con una lista formada por dos vectores y una matriz del ejemplo precedente.

```
> lista <- list(vector_num, vector_pal, matriz)
> names(lista) <- c("numeros", "palabras", "matriz")
```

```
>
```

```
> lista
```

```
$numeros
```

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

```
$palabras
```

```
[1] "universidad" "ciudad" "sector"
```

```
$matriz
```

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

De esa forma será mucho más sencillo si por ejemplo desea acceder a los elementos de una lista de manera separada. Veamos.

```
> lista$matriz
```

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

Otra forma de seleccionar un componente de una lista es usando la posición numerada de ese componente. Por ejemplo, para acceder al tercer componente de “lista”, lo podemos hacer digitando el número 3 entre corchetes.

```
> lista[3]
```

```
$matriz
```

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