

A light blue dotted world map serves as the background for the central text.

CÓMO CREAR > **MATRICES** > EN RSTUDIO

En esta sección abordaremos los siguientes tópicos con relación a las matrices en R usando Rstudio:



- Crear matrices en un script en R,
- Realizar operaciones aritméticas con matrices utilizando RStudio y
- Seleccionar elementos en una matriz.

¿QUÉ SON LAS MATRICES?

Las matrices y arrays pueden ser descritas como vectores multidimensionales. Al igual que un vector, únicamente pueden contener datos de un sólo tipo, pero además de largo, tienen más dimensiones. En un sentido estricto, las matrices son un caso especial de un array, que se distingue por tener específicamente dos dimensiones, un “largo” y un “alto”. Las matrices son, por lo tanto, una estructura con forma rectangular, con renglones y columnas.

Arrays

Los arrays, por su parte, pueden tener un número arbitrario de dimensiones. Pueden ser cubos, hipercubos y otras formas. Su uso no es muy común en R, aunque a veces es deseable contar con objetos n-dimensionales para manipular datos. Como los arrays tienen la restricción de que todos sus datos deben ser del mismo tipo, no importando en cuántas dimensiones se encuentren, esto limita sus usos prácticos.

En general, es preferible usar listas en lugar de arrays.

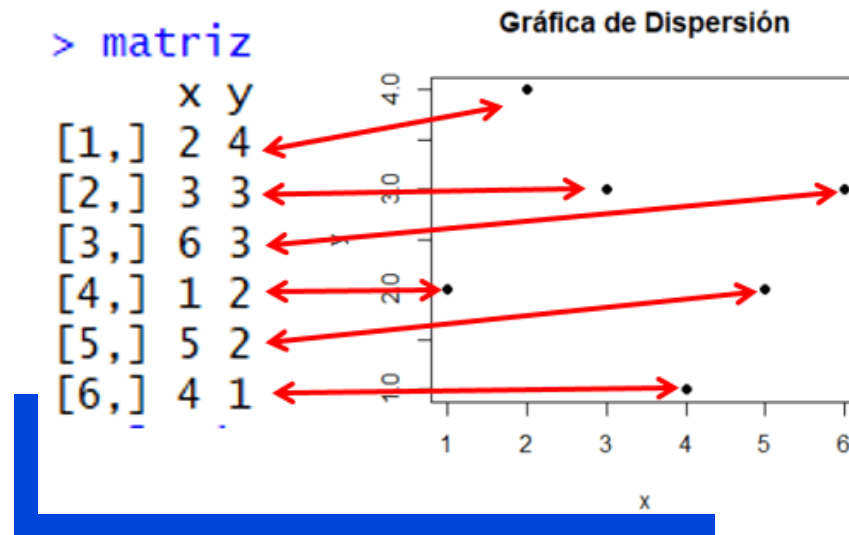
1	2						8	4
	3						7	
		4				6		
			2		3			
		5				9		
		6		9		5		
	7						2	
				5				

¿POR QUÉ SON IMPORTANTES EN LAS MATRICES R?

- Porque las matrices son usadas de manera regular en matemáticas y estadística, además es una estructura de datos de uso común en R



Ejemplo: Si observamos la grafica, cada punto tiene coordenadas en x e y. Los números que corresponden a cada coordenada están registrados en la matriz.



Entonces, es importante aprender matrices para poder aprender con mayor facilidad a hacer las graficas en R.

¿CÓMO CONSTRUIR MATRICES EN R STUDIO?

La función `matrix` permite una matriz en RStudio o R base, pasando como input un vector numérico, de caracteres o lógico. Es decir, lo que va adentro de esta función es el contenido de la matriz y cuántas filas y columnas tiene la matriz, una de las formas de construir una matriz es a partir de un vector, como sigue:

```
(m <- 11:30) # Un vector con 20 números

## [1] 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
## [19] 29 30

# Para convertirla en matriz simplemente se especifica el
# atributo dim
dim(m) <- c(4, 5) # 4 renglones y 5 columnas
m

##      [,1] [,2] [,3] [,4] [,5]
## [1,]  11  15  19  23  27
## [2,]  12  16  20  24  28
## [3,]  13  17  21  25  29
## [4,]  14  18  22  26  30
```

`matrix(data, ← Contenido
nrow = 1, ← Filas
ncol = 1) ← Columnas`

Además, podemos poner nombres a las columnas de la matriz con la función `colname()` y a las filas de la matriz con la función `rowname()`.

En R, guardamos los vectores en el entorno de variables

```
rownames(m) <- c("uno", "dos", "tres", "cuatro", "cinco")
colnames(m) <- c("UNO", "DOS", "TRES", "CUATRO")
m
```

##		UNO	DOS	TRES	CUATRO
##	uno	11	12	13	14
##	dos	15	16	17	18
##	tres	19	20	21	22

Las funciones `rbind()` y `cbind()`, son otras que se pueden utilizar para construir matrices, dando, ya sea los reglones individuales o las columnas individuales, respectivamente

```
m1 <- rbind(c(1.5, 3.2, -5.5), c(0, -1.1, 60))
m1
```

##		[,1]	[,2]	[,3]
##	[1,]	1.5	3.2	-5.5
##	[2,]	0.0	-1.1	60.0

Añadir y eliminar columnas de matrices en R

Como mostramos antes, la función `cbind` puede usarse para crear una matriz. Sin embargo, el uso principal de la función es agregar columnas a las estructuras de datos. Para eliminar columnas, utiliza el operador `-` indicando el índice de la columna en el segundo argumento entre corchetes, como en el ejemplo.

```
# Añadir columna
M<- cbind(M, c(6, 1, 7))

# Añadir dos columnas
m1 <- cbind(M, c(6, 1, 7), c(1, 6, 1))

# Eliminar la primera columna
M <- M[, -1]

# Borrar la primera y la tercera columna
M <- A[, -c(1, 3)]
```



Añadir y eliminar filas a una matriz en R

Equivalentemente a la función anterior, la función `rbind` se puede usar para agregar filas a objetos. También puedes eliminar filas de la misma manera que antes, pero indicando el índice en el primer argumento entre corchetes.

```
# Agregar fila  
A <- rbind(A, c(6, 1))  
  
# Agregar fila de cinco  
A <- rbind(A, 5)  
  
# Eliminar la segunda fila  
A <- A[-2, ]
```





Apilar matrices en R

La función `rbind` también se puede utilizar para apilar o combinar matrices:

```
x <- matrix(c(2, 7, 1, 3, 6, 1), ncol = 2, byrow = TRUE)
y <- matrix(c(3, 7, 6, 3, 5, 9), ncol = 2, byrow = TRUE)

# Apilando las matrices
rbind(x, y)
```

Si creamos una lista de matrices en R y no conocemos la longitud final que tendrá la lista, se puede usar la función `do.call` de la siguiente manera para unir las matrices:

```
lista_matrices <- list(x, y)
lista_matrices
do.call(rbind, lista_matrices)
rbind(lista_matrices[[1]], lista_matrices[[2]]) # Equivalente
```

El código anterior devolverá la misma salida que `rbind(x, y)`, la diferencia es que con `do.call` no necesitamos saber el número de matrices que se apilarán. Si la lista contiene más matrices en su interior, la función seguirá funcionando.



- **Añadir nombres a las filas y a las columnas**

Se pueden cambiar los nombres de las filas y columnas de una matriz con las funciones `rownames` y `colnames`.

- **Eliminar nombres de las filas y las columnas de las matrices**

En caso de que estemos trabajando con una matriz con nombres y deseamos eliminar, simplemente vamos a eliminar los nombres de las filas o columnas asignándoles el valor `NULL` o usando la función `unname` para borrar todos los nombres.

- **Acceder a los elementos de una matriz en R**

Acceder a los elementos de una matriz es similar a acceder a los elementos de un data frame en R. La principal manera para acceder a los datos es usar `matriz[filas, columnas]`. Sin embargo, existen ciertas diferencias entre las estructuras de datos.

- **Eliminar valores NA, NaN e Inf de una matriz**

Existen diferentes tipos como los valores `NA` (no disponible), `NaN` (no es un número) e `Inf` (infinito). Ten en cuenta que puedes eliminar las filas o columnas que contienen estos valores o reemplazarlos con otros valores.

¿CÓMO OPERAR CON MATRICES EN R STUDIO?

A veces, necesitamos modificar nuestras matrices con operaciones aritméticas básicas. Estamos hablando de sumas, restas, multiplicaciones y divisiones. Para ellos podemos ejecutar el siguiente código:

Definiendo la Matriz A:

```
A=matrix(1:9, nrow = 3, byrow = TRUE)
```

A

Definiendo la Matriz A:

```
a=c(5,5,1) # El primer vector.
```

```
b=c(2,2,7) # El segundo vector.
```

```
d=c(5,7,6) # El tercer vector.
```

```
B=rbind(a,b,d) # La matriz B.
```

B

```
Matriz1=A+B # Suma de matrices.
```

Matriz1

```
Matriz2=A-B # Sustracción de matrices.
```

Matriz2

```
Matriz3<-4*A # Multiplicación por un escalar.
```

Matriz3

Unas de las operaciones más importantes es la multiplicación de matrices (comando %*)

```
AB<-A%*%B # Multiplicación de matrices.
```

AB

```
A2<-A%*%A # Para calcular el cuadrado de la matriz A.
```

A2

```
A3<-A%*%A%*%A # Para calcular la matriz cubica de A.
```

A3

```
#=====
```

```
install.packages("Biodem")
```

```
library(Biodem)
```

```
AA<-mtx.exp(A,2) # La matriz A elevado al cuadrado.
```

```
# Comprobemos si es igual al hallado con el comando %*%.
```

AA==A2

```
A8<-mtx.exp(A,8) # La matriz A elevado a la 8.
```

A8

A light blue dotted world map serves as the background for the central text.

GRACIAS