

# Matrices

Téllez Gerardo Rubén

4/4/2021

## Cómo se definen

- **matrix(vector, nrow=n, byrow=valor\_lógico):** para definir una matriz de  $n$  filas formada por las entradas del vector
  - **nrow:** número de filas
  - **burow:** TRUE la matriz se constituye por filas, FALSE (por defecto) la matriz se constituye por columnas
  - **ncol:** número de columnas (puede usarse en lugar de *ncol:*)
- R muestra las matrices indicando como [i,] la fila i-esima y [,j] la columna j-esima
- Todas las entradas de una matriz han de ser del mismo tipo de datos

Ejemplo:

```
#Para construir una matriz con los números del 1 al 12 definido en columnas
M <- matrix(1:12, nrow= 4 )
M
```

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

```
#Definido en filas
N <- matrix(1:12, nrow = 4, byrow = TRUE)
N
```

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

```
#En tres filas
M1 <- matrix(1:12, nrow= 3 )
M1
```

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

## Advertencias

Querer un número  $n$  de filas que no sean múltiplo o submúltiplo de la cantidad de datos despliega una advertencia **REPITIENDO LOS DATOS EN LA MATRIZ GENERADA**

```
M2 <- matrix(1:7, nrow = 5)
```

```
## Warning in matrix(1:7, nrow = 5): la longitud de los datos [7] no es un
## submúltiplo o múltiplo del número de filas [5] en la matriz
```

```
M2
```

```
##      [,1] [,2]
## [1,]    1    6
## [2,]    2    7
## [3,]    3    1
## [4,]    4    2
## [5,]    5    3
```

De crear una matriz con un solo número en lugar de un vector, se arroja la matriz con todas las entradas del mismo número

```
matrix(7, nrow = 5, ncol = 5) #De usar sólo un nx, otorga una sola columna o fila
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    7    7    7    7    7
## [2,]    7    7    7    7    7
## [3,]    7    7    7    7    7
## [4,]    7    7    7    7    7
## [5,]    7    7    7    7    7
```

## Ejercicio

- ¿Cómo definirías una matriz constante? Es decir, ¿Cómo definirías una matriz  $A$  tal que  $\forall i = 1, \dots, n; j = 1, \dots, m; a_{i,j} = k$  siendo  $k \in \mathbb{R}$ ? Como R no admite incógnitas, prueba para el caso específico  $n = 3, m = 5, k = 0$
- Con el vector  $\text{vec} = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$  crea la matriz

$$\begin{pmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{pmatrix}$$

```
#Para el caso específico
matrix(0, nrow = 3, ncol = 5)
```

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

```
#Con el vector
vec <- 1:12
m1 =matrix(vec, nrow = 3)
m1
```

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

## Cómo construirlas

Más que usar la función `matrix()` se suelen usar las funciones `rbind(vectores...)` y `cbind(vectores...)`

- **`rbind(vector1, vector2, ...)`**: construye la matriz de filas *vector1*, *vector2*, ....
- **`cbind(vector1, vector2, ...)`**: construye la matriz de columnas *vector1*, *vector2*, ....
  - Los vectores han de tener la misma longitud
  - También sirve para añadir columnas o filas a una matriz o concatenar por columnas o filas, matrices con el mismo número de filas o columnas.
- **`diag(vector)`**: para construir una matriz diagonal con un vector dado
  - Si aplicamos un *diag()* a un número *n*, produce una matriz identidad del orden *n*

Ejemplo c y r bind:

```
rbind(m1, c(10, 20, 30, 40), c(100, 200, 300, 400))
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
## [4,]   10   20   30   40
## [5,]  100  200  300  400
```

```
cbind(m1, c(10, 20, 30), c(100, 200, 300))
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    4    7   10   10  100
## [2,]    2    5    8   11   20  200
## [3,]    3    6    9   12   30  300
```

Una matriz diagonal es aquella que sólo tiene números en la diagonal, como las matrices solución de Gauss-Jordan.

```
diag(1:4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    2    0    0
## [3,]    0    0    3    0
## [4,]    0    0    0    4
```

```
diag(5, nrow = 3)
```

```
##      [,1] [,2] [,3]
## [1,]    5    0    0
## [2,]    0    5    0
## [3,]    0    0    5
```

## Submatrices

Similar a subvectores, permite acceder a sitios específicos de matrices.

- **matriz[i, j]:** indica la entrada (i, j) de la matriz, siendo  $i, j \in \mathbb{N}$ . Si  $i$  y  $j$  son vectores de índices, estaremos definiendo la submatriz con las filas pertenecientes al vectores  $i$  y las columnas pertenecientes al vector  $j$ .
- **matriz[i,]:** indica la fila  $i$ -ésima de la matriz, siendo  $i \in \mathbb{N}$
- **matriz[,j]:** indica la columna  $j$ -ésima de la matriz, siendo  $j \in \mathbb{N}$  -Si  $i$  o  $j$  es un vector de índices, estaremos definiendo la submatriz con las filas o columnas pertenecientes al vector  $i$  o  $j$ .

Ejemplo:

```
m1
```

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

```
m1[2,]
```

```
## [1]  2  5  8 11
```

```
m1[,4]
```

```
## [1] 10 11 12
```

```
m1[2, 4]
```

```
## [1] 11
```

```
Va <- 1:2
Vb <- 3:4
m1[Va, Vb]
```

```
##      [,1] [,2]
## [1,]    7   10
## [2,]    8   11
```

## Funciones de las matrices

- **diag(matriz):** para obtener la diagonal de la matriz
- **nrow(matriz):** devuelve el número de filas de la matriz
- **ncol(matriz):** devuelve el número de columnas de la matriz
- **dim(matriz):** devuelve las dimensiones de la matriz
- **sum(matriz):** la suma de todas las entradas de la matriz
- **prod(matriz):** la suma de todas las entradas de la matriz
- **mean(matriz):** la media aritmética de todas las entradas de la matriz

```
diag(m1)
```

```
## [1] 1 5 9
```

```
nrow(m1)
```

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

```
ncol(m1)
```

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

```
dim(m1)
```

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

```
sum(m1)
```

```
## [1] 78
```

```
prod(m1)
```

```
## [1] 479001600
```

```
mean(m1)
```

```
## [1] 6.5
```

Funciones por filas o columnas

- **colSums(matriz):** sumas por entradas de la matriz
- **rowSums(matriz):** sumas por filas de la matriz
- **colMeans(matriz):** medias por columnas de la matriz
- **rowMeans(matriz):** medias por filas de la matriz

```
colSums(m1)
```

```
## [1] 6 15 24 33
```

```
rowSums(m1)
```

```
## [1] 22 26 30
```

```
colMeans(m1)
```

```
## [1] 2 5 8 11
```

```
rowMeans(m1)
```

```
## [1] 5.5 6.5 7.5
```

## Funciones por filas o columnas, función apply

\_\_\_apply(matriz, MARGIN = X, FUN = función) + X = 1: por filas + X = 2: por columnas + X = c

```
apply(m1, MARGIN = 1, FUN = function(x){sqrt(sum(x^2))})
```

```
## [1] 12.88410 14.62874 16.43168
```

```
apply(m1, MARGIN = 2, FUN = function(x){sqrt(sum(x^2))})
```

```
## [1] 3.741657 8.774964 13.928388 19.104973
```

```
apply(m1, MARGIN = c(1, 2), FUN = function(x){x^2})
```

```
##      [,1] [,2] [,3] [,4]  
## [1,]    1   16   49  100  
## [2,]    4   25   64  121  
## [3,]    9   36   81  144
```

Ejemplo Dada la matriz

$$\begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

```
A <- matrix(1:9, ncol = 3)  
dim(A)
```

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

```
diag(A)
```

```
## [1] 1 5 9
```