

Matrices

Alexis Frías Domínguez

8/8/2020

Matrices

Cómo definirlas

- `matrix(vector, nrow=n, byrow=valor_lógico)` para definir una matriz de n filas formadas por las entradas del vector

```
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
```

- **nrow** número de filas
- **byrow** si se iguala a TRUE, la matriz se construye por filas; si se iguala a FALSE (valor por defecto), se construye por columnas.

```
ma = matrix(1:12, nrow = 4, byrow = TRUE)
ma
```

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

- **ncol** número de columnas (puede usarse en lugar de nrow)
- R muestra las matrices como $[i]$ la fila i -ésima y $[j]$ la columna j -ésima
- Todas las entradas de una matriz han de ser del mismo tipo de datos

Cómo construirlas

- `rbind(vector1, vector2, ...)` construye la matriz de filas vector1, vector2,...

```
a = c(5,3,6)
b = c(8,5,3)
rbind(a,b)
```

```
##      [,1] [,2] [,3]
## a      5      3      6
## b      8      5      3
```

- `cbind(vector1, vector2, ...)` construye la matriz de columnas `vector1`, `vector2`,...

```
cbind(a,b)
```

```
##      a b
## [1,] 5 8
## [2,] 3 5
## [3,] 6 3
```

- Los vectores han de tener la misma longitud
- También sirve para añadir columnas (filas) a una matriz o concatenar por columnas (filas) matrices con el mismo número de filas (columnas)
- `diag(vector)` para construir una matriz diagonal con un vector dado

```
diag(a)
```

```
##      [,1] [,2] [,3]
## [1,]    5    0    0
## [2,]    0    3    0
## [3,]    0    0    6
```

- Si aplicamos `diag` a un número n , produce una matriz identidad de orden n

```
diag(6)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    0    0    0    0    0
## [2,]    0    1    0    0    0    0
## [3,]    0    0    1    0    0    0
## [4,]    0    0    0    1    0    0
## [5,]    0    0    0    0    1    0
## [6,]    0    0    0    0    0    1
```

Submatrices

- `matriz[i,j]` indica la entrada (i,j) de la matriz. Si i,j son vectores de índices, estaremos definiendo la submatriz con las filas pertenecientes al vector i y columnas pertenecientes al vector j

```
m
```

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

```
m[2,3]
```

```
## [1] 10
```

- **matriz[i,]** indica la fila i -ésima de la matriz

```
m[1,]
```

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

- **matriz[,j]** indica la columna j -ésima

```
m[,1]
```

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

- Si $i(j)$ es un vector de índices, estaremos definiendo la submatriz con las filas (columnas) pertenecientes al vector $i(j)$

Funciones

- **diag(matriz)** para obtener la diagonal de la matriz
- **nrow(matriz)** nos devuelve el número de filas de la matriz
- **ncol(matriz)** nos devuelve el número de columnas de la matriz
- **dim(matriz)** nos devuelve las dimensiones de la matriz
- **sum(matriz)** obtenemos la suma de todas las entradas de la matriz
- **prod(matriz)** obtenemos el producto de todas las entradas de la matriz
- **mean(matriz)** obtenemos la medida aritmética de todas las entradas de la matriz
- **colSums(matriz)** obtenemos las sumas por columnas de la matriz
- **rowSums(matriz)** obtenemos las sumas por filas de la matriz
- **colMeans(matriz)** obtenemos las medidas aritméticas por columnas de la matriz
- **rowMeans(matriz)** obtenemos las medidas aritméticas por filas de la matriz