# MATRICES CON OCTAVE

Ramon Ceballos

21/2/2021

## 1. Definir Matrices en Octave

Para crear una matriz fila, se hace:

```
row = [1 2 3]
```

```
## row =
##
##    1   2   3
```

Para crear una matriz columna, se hace:

```
col = [1;2;3]
```

```
## col =
##
##    1
##    2
##    3
```

Entonces, para crear una matriz:

```
M = [1 2 3; 4 5 6; 7 8 9]
```

```
## M =
##
##    1   2   3
##    4   5   6
##    7   8   9
```

Para llamar a un elemento, utilizamos la sintaxis siguiente:

```
M = [1 2 3; 4 5 6; 7 8 9];
M(1,1)
```

```
## ans = 1
```

Para llamar a una fila, utilizamos la sintaxis siguiente:

```
M = [1 2 3; 4 5 6; 7 8 9];
M(2,:)
```

```
## ans =
##
##    4   5   6
```

Para llamar a una columna, utilizamos la sintaxis siguiente:

```
M = [1 2 3; 4 5 6; 7 8 9];
M(:,3)
```

```
## ans =
##
##    3
##    6
##    9
```

## 1.1 Comprobar tipo de matriz

Para saber el tipo de una matriz, lo que hacemos es utilizar la función **`matrix_type()`**.

Matriz completa (**Full**) con todo lleno de números.

```
M = [1 2 3; 4 5 6; 7 8 9];
matrix_type(M)
```

```
## ans = Full
```

**Upper** refiere a una matriz triangular superior, aunque sea diagonal.

```
N = [1 0 0; 0 1 0; 0 0 9];
matrix_type(N)
```

```
## ans = Upper
```

## 1.2 Crear diversos tipos de matrices

Para crear una **matriz de ceros**, utilizamos la función **`repmat(0,m,n)`** (repetir matriz) donde $m$ son el número de filas y $n$ el de columnas:

```
O = repmat(0,5,8)
```

```
## O =
##
##    0   0   0   0   0   0   0   0
##    0   0   0   0   0   0   0   0
##    0   0   0   0   0   0   0   0
##    0   0   0   0   0   0   0   0
##    0   0   0   0   0   0   0   0
```

Para crear una **matriz de unos**, utilizamos una sintaxis similar a la anterior:

```
ones = repmat(1,3,7)
```

```
## ones =
##
##    1   1   1   1   1   1   1
##    1   1   1   1   1   1   1
##    1   1   1   1   1   1   1
```

Para crear una **matriz diagonal**, se emplea:

```
N = diag([1 2 3 4 5])
```

```
## N =
##
## Diagonal Matrix
##
##    1   0   0   0   0
##    0   2   0   0   0
##    0   0   3   0   0
##    0   0   0   4   0
##    0   0   0   0   5
```

## 1.3 Diagonal y dimensión de una matriz

Para obtener la **diagonal principal de una matriz** se utiliza:

```
N = diag([1 2 3 4 5]);
diagonal = diag(N)
```

```
## diagonal =
##
##    1
##    2
##    3
##    4
##    5
```

Para saber la **dimensión de una matriz**, hacemos uso de la función **`size()`** se utiliza:

```
M = [1 2 3; 4 5 6; 7 8 9];
size(M)
```

```
## ans =
##
##    3   3
```

## 2. Manipulación de matrices con Octave

Para **sumar todos los elementos de una matriz**, se emplea un doble sum() ya que el primero solo sumaría columnas y el segundo haría la suma ya total.

```
M = [1 2 3; 4 5 6; 7 8 9];
suma = sum(sum(M))
```

```
## suma = 45
```

Para **sumar los elementos de una matriz por filas**:

```
M = [1 2 3; 4 5 6; 7 8 9];
sumaFil = sum(M,2)
```

```
## sumaFil =
##
##     6
##    15
##    24
```

Para **sumar los elementos de una matriz por columnas**, basta simplemente con hacer lo siguiente:

```
M = [1 2 3; 4 5 6; 7 8 9];
sumaCol = sum(M)
```

```
## sumaCol =
##
##    12   15   18
```

O bien:

```
M = [1 2 3; 4 5 6; 7 8 9];
sumaCol = sum(M, 1)
```

```
## sumaCol =
##
##    12   15   18
```

Para obtener el **productorio** de los eleemntos de una matriz se emplea:

```
M = [1 2 3; 4 5 6; 7 8 9];
producto = prod(prod(M))
```

```
## producto = 362880
```

Para calcular la **media los elementos de una matriz** se usa:

```
M = [1 2 3; 4 5 6; 7 8 9];
media = mean(mean(M))
```

```
## media = 5
```

Para calcular la **media los elementos de una matriz por fila*s se hace lo siguiente:

```
M = [1 2 3; 4 5 6; 7 8 9];
mediaFil = mean(M,2)
```

```
## mediaFil =
##
##    2
##    5
##    8
```

Para calcular la **media los elementos de una matriz por columnas** basta simplemente con hacer lo siguiente:

```
M = [1 2 3; 4 5 6; 7 8 9];
mediaCol = mean(M)
```

```
## mediaCol =
##
##    4   5   6
```

O bien:

```
M = [1 2 3; 4 5 6; 7 8 9];
mediaCol = mean(M,1)
```

```
## mediaCol =
##
##    4   5   6
```

# 3. Operaciones con matrices en Octave

## 3.1 Transpuesta de una matriz

Para calcular la transpuesta de una matriz, utilizamos el apostrofe, `'`.

```
M = [1 2 3; 4 5 6; 7 8 9];
M'
```

```
## ans =
##
##    1   4   7
##    2   5   8
##    3   6   9
```

### 3.2 Traza de una matriz

Para calcular la traza de una matriz, utilizamos la función `trace()`:

```
A = [1 -1; 0 3];
B = [2 1; -1 0];
trace(A)
trace(B)
```

```
## ans = 4
## ans = 2
```

### 3.3 Suma de matrices

Suma de matrices:

```
A = [1 -1; 0 3];
B = [2 1; -1 0];
A+B
```

```
## ans =
##
##    3   0
##   -1   3
```

### 3.4 Producto de un escalar por una matriz

Producto de un escalar por una matriz:

```
A = [1 -1; 0 3];
2*A
```

```
## ans =
##
##    2  -2
##    0   6
```

### 3.5 Producto de matrices

Producto de matrices:

```
A = [1 -1; 0 3];
B = [2 1; -1 0];
A*B
```

```
## ans =
##
##    3   1
##   -3   0
```

### 3.6 Potencia de matrices

Potencia de matrices:

```
A = [1 -1; 0 3];
B = [2 1; -1 0];
A^3
B^4
```

```
## ans =
##
##     1  -13
##     0   27
##
## ans =
##
##    5   4
##   -4  -3
```

## 4. Rango e inversa de una matriz con Octave

### 4.1 Rango de una matriz

Para calcular el rango de una matriz, utilizamos la función **`rank()`**.

```
A = [1 -1; 0 3];
B = [2 1; -1 0];
rank(A)
rank(B)
```

```
## ans = 2
## ans = 2
```

### 4.2 Inversa de una matriz

Para calcular la inversa de una matriz, hacemos uso de la función **`inv()`**.

```
A = [1 -1; 0 3];
B = [2 1; -1 0];
inv(A)
inv(A)*A
```

```
## ans =
##
##    1.0000   0.3333
##         0   0.3333
##
## ans =
##
##    1.0000  -0.0000
##         0   1.0000
```

Comprobamos si el producto da la identidad.

```
A = [1 -1; 0 3];
inv(A)*A
```

```
## ans =
##
##    1.0000  -0.0000
##         0   1.0000
```