

# Matrices

Victor Lopez

2023-01-12

## Crear matriz

```
# matrix(vector, nrow or ncol = n, byrow = TRUE or FALSE). Por defecto byrow es FALSE y se  
#construye por columnas  
  
m = matrix(1:12, nrow = 4, ncol = 5, byrow = TRUE) # En caso de que las columnas y filas tengas  
#espacios sobrantes, volveran a repetir los elementos. En lugar de un vector podria ser un num  
  
m = rbind(c(1, 2, 3), c(4, 5, 6), c(7, 8, 9)) # Construir matriz por filas  
m = cbind(c(1, 2, 3), c(4, 5, 6), c(7, 8, 9)) # Construir matriz por columnas  
# En lugar de vectores podemos pasar matrices y agregarle filas o columnas con la misma longitud  
  
m = diag(c(1, 2, 3)) # Vector identidad. Diagonal con el vector y los demas elementos son ceros  
m[2, 4] # [Fila, columna]. Nos devuelve un dato en concreto de la matriz  
m[4,] # Devuelve un vector con los datos de la cuarta fila  
m[,3] # Devuelve un vector con los datos de la tercera columna  
  
m[c(2, 3, 5), 1:2]
```

## Metodos

```
diag(m) # Vector que es la diagonal de la matriz  
  
nrow(m) # Numero de filas de la matriz  
ncol(m)  
  
dim(m) # Dimension filas y columnas  
  
sum(m) # Suma de los elementos  
  
prod(m) # Producto de los elementos  
  
mean(m) # Promedio de los elementos  
  
colSums(m) # Devuelve vector de las sumas de cada columna  
  
rowSums(m)
```

```
colMeans(m)

rowMeans(m)

apply(m, MARGIN = 1, FUN = function(x){sqrt(sum(x^2))}) # x en function es un vector de los
#valores de cada columna y lo que retorne, lo retornara para cada columna. MARGIN = 1 indica
#filas, MARGIN = 2 indica columnas, MARGIN = c(1, 2) indica que x sera cada elemento
```

## Operaciones

```
t(m) # Transpuesta

+

* # Producto de cada coordenada por cada coordenada

%*% # Para multiplicar matrices

# Con el paquete Biodem
mtx.exp(m, n) # Potencia. Aproxima, no son exactas

# Con el paquete expm
%% # Potencia

det(m) # Determinante

qr(m)$rank # Calcular rango

solve(m) # Inversa de una matriz invertible
solve(m, b) # Tambien sirve para resolver sistemas de ecuaciones lineales donde b es el vector
#de terminos independientes

eigen(m)$values # Devuelve un vector con los valores propios en orden decreciente de su valor
#absoluto y repetidos tantas veces como su multiplicidad algebraica
eigen(m)$vectors # Nos da una matriz cuyas columnas son los vectores propios de la matriz
# Si hay algún vap con multiplicidad algebraica mayor que 1 (es decir, que aparece más de una
#vez), la función eigen() da tantos valores de este vap como su multiplicidad algebraica indica.
#Además, en este caso, R intenta que los veps asociados a cada uno de estos vaps sean
#linealmente independientes. Por tanto, cuando como resultado obtenemos veps repetidos asociados
#a un vap de multiplicidad algebraica mayor que 1, es porque para este vap no existen tantos
#veps linealmente independientes como su multiplicidad algebraica y, por consiguiente, la matriz
#no es diagonalizable.
```