

Matrices

Oscar Gerardo Hernández Martínez

8/7/2019

¿Cómo se construye una matriz?

- **matrix(vector, nrow=n, byrow=valor_lógico)**: para definir una matriz de n filas formada por las entradas del vector
 - **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.
 - **ncol**: número de columnas (puede usarse en lugar de *nrow*)
 - **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 (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
 - * Si aplicamos *diag* a un número n , produce una matriz identidad de orden n

Submatrices

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

R muestra las matrices indicando 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.

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 media 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 medias aritméticas por columnas de la matriz.
- **rowMeans(matriz)**: obtenemos las medias aritméticas por filas de la matriz.
- **t(matriz)**: para obtener la transpuesta de la matriz.
- **+**: para sumar matrices.
- *****: para el producto de un escalar por una matriz.
- **%*%**: para multiplicar matrices.
- **mtx.exp(matriz,n)**: para elevar la matriz a n .
 - Del paquete **Biodem**

- * No calcula las potencias exactas, las aproxima.
- **%%**: para elevar matrices.
 - Del paquete **expm**
 - * No calcula las potencias exactas, las aproxima.
- **det(matriz)**: para calcular el determinante de la matriz.
- **qr(matriz)\$rank**: para calcular el rango de la matriz.
- **solve(matriz)**: para calcular la inversa de una matriz invertible.
 - También sirve para resolver sistemas de ecuaciones lineales. Para ello introducimos **solve(matriz,b)**, donde *b* es el vector de términos independientes.
- **eigen(matriz)**: para calcular los valores propios (vaps) y vectores propios (veps).
 - **eigen(matriz)\$values**: nos da el vector con los vaps de la matriz en orden decreciente de su valor absoluto y repetidos tantas veces como su multiplicidad algebraica.
 - **eigen(matriz)\$vectors**: nos da una matriz cuyas columnas son los veps de la matriz.

Función **apply()**

- **apply(matriz, MARGIN=..., FUN=función)**: para aplicar otras funciones a las filas o las columnas de una matriz.
 - *MARGIN*: ha de ser 1 si queremos aplicar la función por filas; 2 si queremos aplicarla por columnas; o c(1,2) si la queremos aplicar a cada entrada.