Matrizes em R

Matrizes, são tabelas retangulares de números, expressões matemáticas ou símbolos, cujo elementos são arranjados em n linhas e p colunas. A matrix **A** abaixo, por exemplo, organiza cada elemento a_{ij} em 2 linhas e 3 colunas.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

Em R, a matriz é uma estrutura de dados homogênea (deve conter mesmo tipo de dados, e.i. numérico, caracteres, lógico) de duas dimensões. Sua sintaxe básica é realizada por meio da função `matrix()`, tendo como argumentos os elementos que compõem a matriz, na forma de um vetor, e o número de linhas e colunas.

```
M = matrix(data = 1:9, nrow = 3, ncol = 3)


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

Por *default* os elementos são organizados por coluna na matriz, caso se deseje que estes sejam organizados por linha usa-se o argumento `byrow = TRUE`.

```
M2 = matrix(data = 1:9, nrow = 3, ncol = 3, byrow = TRUE)


print(M2)
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

A função `matrix()`, na realidade, organiza as dimensões de um vetor qualquer. Na prática, não é necessário informar o número de linhas e de colunas se o vetor de dados é informado, contudo, tal prática facilita a leitura do código.

```
V3 = c(12, 23, 34, 45, 56, 67)


M3 = matrix(data = V3, nrow = 3)
```

```r
print(M3)
```

```
##      [,1] [,2]
## [1,]   12   45
## [2,]   23   56
## [3,]   34   67
```

Para criar uma matriz contendo um único valor, e.g. matriz nula, informa-se este valor no argumento data em conjunto com o número de linhas e colunas.

```r
M4 = matrix(data = 0, nrow = 3, ncol = 5)


print(M4)
```

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

Caso seja de interesse alocar uma matriz vazia, basta informar os argumentos ncol e nrow. Esta será completada com o valor lógico NA (do inglês, not avaliable).

```r
M5 = matrix(nrow = 4, ncol = 5)


print(M5)
```

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

Operações matriciais

Para ilustrar as operações entre matrizes usaremos as seguintes matrizes para fins de cálculo.

$$M_6 = \begin{pmatrix} 2 & 5 \\ 3 & 6 \end{pmatrix} \qquad M_7 = \begin{pmatrix} 5 & 8 \\ 3 & 2 \end{pmatrix}$$

$$M_8 = \begin{pmatrix} 2 & 6 & 0 \\ 4 & 2 & 1 \end{pmatrix} \qquad M_9 = \begin{pmatrix} 1 & 0.5 & 0.3 \\ 0.5 & 1 & 0.9 \\ 0.3 & 0.9 & 1 \end{pmatrix}$$

```r
M6 = matrix(c(2, 3, 5, 6), nrow = 2)

M7 = matrix(c(5, 3, 8, 2), nrow = 2)

M8 = matrix(c(2, 4, 6, 2, 0,1), nrow = 2, ncol = 3)

M9 = matrix(c(1, 0.5, 0.3, 0.5, 1, 0.9, 0.3, 0.9, 1), nrow = 3, ncol = 3)
```

Soma e subtração

Dada duas matrizes de mesma dimensionalidade, a matriz resultante da soma / subtração destas matrizes corresponde à matriz cujos elementos são a soma / subtração dos elementos das matrizes originais.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a+e & b+f \\ c+g & d+h \end{pmatrix}$$

Em R, para a soma e subtração de matrizes usa-se a mesma sintaxe que na operação entre escalares.

```r
M_soma = M7 + M6

print(M_soma)
##      [,1] [,2]
## [1,]    7   13
## [2,]    6    8
```

```r
M_subt = M7 - M6

print(M_subt)
##      [,1] [,2]
## [1,]    3    3
## [2,]    0   -4
```

Multiplicação por escalar

Uma matriz de dimensionalidade qualquer quando multiplicada por um escalar **k**, tal que k∈R, resulta em uma matriz de mesmas dimensões cujos elementos são o produto do escalar k por cada um dos elementos da matriz original.

$$k \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} k \cdot a & k \cdot b \\ k \cdot c & k \cdot d \end{pmatrix}$$

Em R, utiliza-se a mesma sintaxe que na multiplicação entre escalares.

```
M_prod_escalar = 42 * M9


print(M_prod_escalar)
##      [,1] [,2] [,3]
## [1,] 42.0 21.0 12.6
## [2,] 21.0 42.0 37.8
## [3,] 12.6 37.8 42.0
```

Multiplicação elemento a elemento

Os elementos da matriz resultante de uma multiplicação elemento a elemento entre duas matrizes de mesmas dimensões correspondem ao produto dos elementos equivalentes entre as matrizes originais.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \circ \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a \cdot e & b \cdot f \\ c \cdot g & d \cdot h \end{pmatrix}$$

Em R, a multiplicação elemento a elemento é realizada por meio do operador * desde que as matrizes tenham as mesmas dimensões.

```
M_prod_elemento = M6 * M7


print(M_prod_elemento)
##      [,1] [,2]
## [1,]   10   40
## [2,]    9   12
```

Multiplicação matricial

A multiplicação entre uma matriz (n x m) por uma matriz (m x p), em que **m,n,p** $\in \mathbb{N}^+$, resulta em uma matriz de dimensões (n x p), cujos elementos são a somatória do produto entre os elementos em linha da matriz que pré-multiplica pelos elementos em coluna da matriz que pós-multiplica.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a \cdot e + b \cdot g & a \cdot f + b \cdot h \\ c \cdot e + d \cdot g & c \cdot f + d \cdot h \end{pmatrix}$$

Em R, a multiplicação matricial clássica é desenvolvida pelo uso do operador %*%, respeitando a equidade entre o número de colunas da matriz que pré-multiplica e o número de linhas da matriz que pós-multiplica.

```
M_prod = M6 %*% M8
```

```
print(M_prod)
##      [,1] [,2] [,3]
## [1,]   24   22    5
## [2,]   30   30    6
```

Transposta de uma matriz

A transposta de uma matriz $\mathbf{A_{nxm}}$ é uma matriz $\mathbf{M=A^T_{mxn}}$, em que os elementos em coluna correspondem aos elementos em linha da matriz original.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^T = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$$

Em R, a transposta de uma matriz é determinada pela função `t()`.

```
M_transp = t(M8)


print(M_transp)
##      [,1] [,2]
## [1,]    2    4
## [2,]    6    2
## [3,]    0    1
```

Determinante de uma matriz

O determinante de uma matriz é uma função matricial que converte uma matriz quadrada em um escalar. Para uma matriz quadrada de ordem 2, o determinante é definido como:

$$det \begin{pmatrix} a & b \\ c & d \end{pmatrix} = a \cdot d - c \cdot b$$

Para uma matriz de ordem 3 ou superior, o determinante pode ser estimado pela seguinte equação:

$$det(A) = \sum_{j=1}^{n} (-1)^{i+j}. a_{ij}. det(A_{-i-j})$$

Em que $\mathbf{A_{-i-j}}$ corresponde à matriz **A** excluindo-se a linha i e a coluna j. Por esta equação uma matriz de ordem n > 2 tem seu cálculo simplificado à soma de n^2 determinantes originados da matriz **A**.

Em R, o determinante de uma matriz é calculado pela função `det()`

```
det(M7)
## [1] -14
```

Inversa de uma matriz

A inversa de uma matriz quadrada qualquer cujo determinante seja não nulo, $A_{n,n}$ é uma matriz de mesmas dimensões que atenda a seguinte condição:

$$A^{-1} \cdot A = I$$

Onde, I é a matriz identidade de mesma dimensão que a matriz A.

A inversa de uma matriz é estimada pela função `solve()`, mesma função utilizada para resolução de sistemas de equações em R.

```
M_inversa = solve(M6)


print(M_inversa)
##      [,1]       [,2]
## [1,]   -2  1.6666667
## [2,]    1 -0.6666667
```

Facilmente, verifica-se a condição de existência da matriz inversa:

```
solve(M6)%*%M6
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

Traço de uma matriz

O traço de uma matriz é determinado extraindo os elementos da diagonal da matriz quadrada e então efetuando a soma dos seus elementos. Assim, para uma matriz quadrada de ordem 2, o traço é determinado por:

$$tr\begin{pmatrix} a & b \\ c & d \end{pmatrix} = a + d$$

Em R, a obtenção do traço passa pela extração dos elementos da diagonal da matriz, por meio da função `diag()`, e a soma destes elementos pela função `sum()`.

```
sum(diag(M6))
## [1] 8
```