

# Introdução à Análise de Dados com Linguagem R

## Aula 3

### 1. Matriz

Matriz é uma estrutura de dados semelhante a vetor, exceto que na matriz temos 2 dimensões, uma para as linhas e outra para as colunas. O código a seguir mostra a criação de uma matriz 3x3.

```
In [113...] matriz <- matrix(1:9, nrow = 3, ncol = 3)
print(matriz)
```

```
1 4 7
2 5 8
3 6 9
```

### Somar linhas e Colunas de uma Matriz

A função `apply`, parte do pacote `base` do R, pode ser usada para aplicar uma determinada função a uma matriz, e recebe 3 argumentos como parâmetro: a matriz contendo os dados, a indicação do sentido de aplicação da função, representado pelos números 1 (linha) ou 2 (coluna) e a função a ser aplicada.

Somar as linhas de uma matriz:

```
In [64]: print(apply(matriz, 1, sum))
```

```
[1] 12 15 18
```

Somar os valores das colunas de uma matriz:

```
In [65]: print(apply(matriz, 2, sum))
```

```
[1] 6 15 24
```

### Somar os Elementos da Diagonal de uma Matriz

```
In [117...] m <- matrix(1:9, nrow = 3, ncol = 3)
print(m)
```

```
      [,1] [,2] [,3]
[1,]     1     4     7
[2,]     2     5     8
[3,]     3     6     9
```

```
In [115...] sum(diag(m))
```

```
15
```

### Sentido de Preenchimento dos Dados em uma Matriz

A função `matrix()` tem por padrão o preenchimento no sentido das colunas, porém, em alguns casos podemos necessitar preencher uma matriz no sentido das linhas, para isso devemos definir o valor do argumento `byrow = TRUE`

```
In [130... matrix <- matrix(1:9, nrow = 3, ncol = 3, byrow = TRUE)
print(matrix)
```

```
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

## Atribuir Nomes as Linhas e Colunas de uma Matriz

```
In [135... matrix <- matrix(1:9, nrow = 3, ncol = 3, byrow = TRUE)
print(matrix)
```

```
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

```
In [137... # Atribuir Nomes as Linhas da matriz
rownames(matrix) <- c('Linha 1', 'Linha 2', 'Linha 3')
```

```
In [139... print(matrix)
```

```
      [,1] [,2] [,3]
Linha 1    1    2    3
Linha 2    4    5    6
Linha 3    7    8    9
```

```
In [140... # Atribuir Nomes as colunas da matriz
colnames(matrix) <- c('Coluna 1', 'Coluna 2', 'Coluna 3')
```

```
In [141... matrix
```

	Coluna 1	Coluna 2	Coluna 3
Linha 1	1	2	3
Linha 2	4	5	6
Linha 3	7	8	9

## Obter os nomes das Linhas e Colunas de uma Matriz

### Somente os Nomes das Linhas

```
In [128... rownames(matrix)
```

1. 'Linha 1'
2. 'Linha 2'
3. 'Linha 3'

### Somente os Nomes das Colunas

```
In [129... colnames(matriz)
```

1. 'Coluna 1'
2. 'Coluna 2'
3. 'Coluna 3'

## Nomes das Linhas e Colunas

```
In [142... dimnames(matriz)
```

1. A. 'Linha 1'  
B. 'Linha 2'  
C. 'Linha 3'
2. A. 'Coluna 1'  
B. 'Coluna 2'  
C. 'Coluna 3'

## Acessar Linhas e Colunas da Matriz

```
In [81]: # mostrar a primeira linha da matriz  
print(matriz[1, ])
```

```
[1] 1 2 3
```

```
In [82]: # mostrar a segunda Coluna da matriz  
print(matriz[, 2])
```

```
[1] 2 5 8
```

## Acessar Elementos da Matriz

```
In [84]: # Mostrar o elemento pertencente a segunda linha e segunda coluna  
print(matriz[2, 2])
```

```
[1] 5
```

## Alterar os Elementos de uma Matriz

```
In [87]: # alterar o elemento da linha 2 coluna 2, número 5, para 0  
matriz[2, 2] <- 0  
print(matriz)
```

```
      [,1] [,2] [,3]  
[1,]     1     2     3  
[2,]     4     0     6  
[3,]     7     8     9
```

## Operações com Matrizes

- **Maior e menor valor entre os elementos da matriz**

```
In [104... # maior valor entre os elementos da matriz  
max(matriz)
```

9

```
In [105]: # menor valor entre os elementos da matriz  
min(matriz)
```

0

- **Maior e menor valor de uma linha ou coluna da matriz**

```
In [99]: # maior valor entre os elementos da primeira linha  
max(matriz[1,])
```

3

```
In [102]: # menor valor entre os elementos da terceira coluna  
min(matriz[:,3])
```

3

- **Média dos elementos da matriz**

```
In [103]: mean(matriz)
```

4.4444444444444444

- **Somar os valores das linhas e colunas**

- **Soma de elementos da matriz**

```
In [106]: # somar os valores da primeira linha  
sum(matriz[1, ])
```

6

```
In [107]: # somar os valores da terceira coluna  
sum(matriz[:, 3])
```

18

```
In [88]: # somar os elementos da segunda linha da matriz  
sum(matriz[2, ])
```

10

- **Diagonal da matriz**

```
In [89]: # Obter a diagonal da matriz  
diag(matriz)
```

1.1

2.0

3.9

```
In [90]: # Obter a soma entre os elementos da diagonal da matriz  
sum(diag(matriz))
```

- **Transposição de Matriz**

```
In [91]: # Transpor a matriz
t(matriz)
```

```
1  4  7
```

```
2  0  8
```

```
3  6  9
```

- **Soma entre matrizes**

```
In [95]: # Definição das matrizes "a" e "b"
a <- matrix(1:6, nrow = 3, byrow = TRUE)
b <- matrix(1:6, nrow = 3, byrow = TRUE)

print(a)
print(b)
```

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

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

```
In [97]: # soma das matrizes a e b
a + b
```

```
2  4
```

```
6  8
```

```
10 12
```

## Combinar Vetores em Matriz

Em R podemos combinar vetores para formar uma matriz em que cada vetor fará parte de uma coluna ou linha da matriz. Para combinar vetores em linhas matriciais usamos a função `rbind()`, e para combinar vetores em colunas da matriz usamos a função `cbind()`. O exemplo a seguir mostra como combinar três vetores com orientação nas linhas de uma matriz.

```
In [30]: # Vetor referente a uma amostra de valores de ações da Apple
apple <- c(109.49, 109.90, 109.11, 109.95, 111.03)

# Vetor referente a uma amostra de valores de ações da IBM
ibm <- c(159.82, 160.02, 159.84, 160.35, 164.79)

# Vetor referente a uma amostra de valores de ações da Microsoft
microsoft <- c(59.20, 59.25, 60.22, 59.95, 61.37)

# combinar os vetores em uma matriz onde cada linha receberá os valores dos vetores
rbind(apple, ibm, microsoft)
```

<b>apple</b>	109.49	109.90	109.11	109.95	111.03
<b>ibm</b>	159.82	160.02	159.84	160.35	164.79
<b>microsoft</b>	59.20	59.25	60.22	59.95	61.37

A seguir é demonstrado como combinar os elementos de vetores em colunas de uma matriz.

```
In [31]: # combinar os vetores em uma matriz onde cada coluna receberá os valores dos vetores
cbind(apple, ibm, microsoft)
```

<b>apple</b>	<b>ibm</b>	<b>microsoft</b>
109.49	159.82	59.20
109.90	160.02	59.25
109.11	159.84	60.22
109.95	160.35	59.95
111.03	164.79	61.37

## Matriz de Correlação

Como exemplo prático para demonstrar o uso de matriz para calcular a correlação entre variáveis, usaremos os dados referente a publicação:

Ramsey, F.L. and Schafer, D.W. (2013). *The Statistical Sleuth: A Course in Methods of Data Analysis* (3rd ed), Cengage Learning.

Os dados são os valores médios de peso cerebral (g), peso corporal (g), duração da gestação (dias) e tamanho da prole de 96 espécies de mamíferos.

```
In [1]: # Carregar os dados vetoriais
load('./data/dados_modulo_1_aula_3.rda')

# listar os objetos no ambiente R
ls()
```

1. 'cerebro'
2. 'corpo'
3. 'especies'
4. 'gestacao'
5. 'prole'

```
In [2]: # Combinar os vetores em uma matriz
m <- cbind(cerebro, corpo, gestacao, prole)

# Mostrar as primeiras 6 linhas da matriz
head(m)

# Mostrar as últimas 6 linhas da matriz
tail(m)
```

<b>cerebro</b>	<b>corpo</b>	<b>gestacao</b>	<b>prole</b>
9.6	2.20	31	5.0

9.9	0.78	98	1.2
4480.0	2800.00	655	1.0
20.3	2.80	104	1.3
219.0	89.00	218	1.0
53.0	6.00	60	2.2

	cerebro	corpo	gestacao	prole
[91,]	198	45.0	300	1.1
[92,]	550	400.0	310	1.0
[93,]	179	32.0	180	1.0
[94,]	102	5.5	210	1.0
[95,]	185	150.0	120	4.0
[96,]	334	250.0	255	1.0

## Atribuir um Atributo a uma Matriz

Para inserir um atributo a matriz utilizamos a função `attr()`, passando como argumentos a matriz e um rótulo para nomear o atributo. Como demonstração iremos inserir um atributo a nossa matriz definida anteriormente, este atributo será a referência bibliográfica dos dados.

```
In [3]: # Obter os atributos da matriz
attributes(m)
```

**\$dim**

1. 96
2. 4

**\$dimnames**

1. NULL
2.
  - A. 'cerebro'
  - B. 'corpo'
  - C. 'gestacao'
  - D. 'prole'

```
In [4]: # Inserir o atributo
attr(m, 'Fonte') <- 'Ramsey, F.L. and Schafer, D.W. (2013). The Statistical Sleuth: A Co
```

```
In [5]: # conferir os atributos da matriz
attributes(m)
```

**\$dim**

1. 96
2. 4

**\$dimnames**

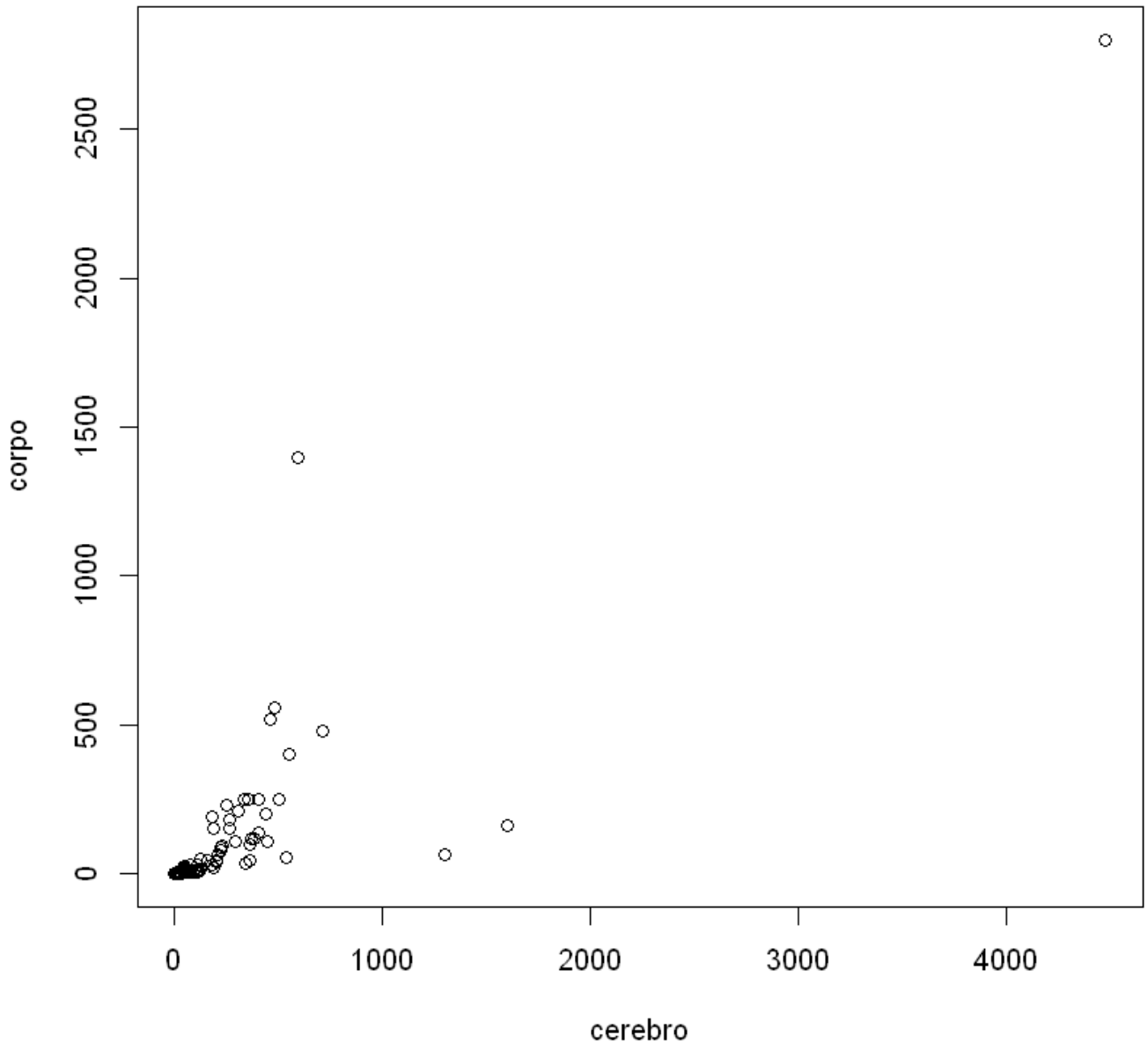
1. NULL
2.
  - A. 'cerebro'
  - B. 'corpo'
  - C. 'gestacao'

**\$Fonte**

'Ramsey, F.L. and Schafer, D.W. (2013). The Statistical Sleuth: A Course in Methods of Data Analysis (3rd ed), Cengage Learning.'

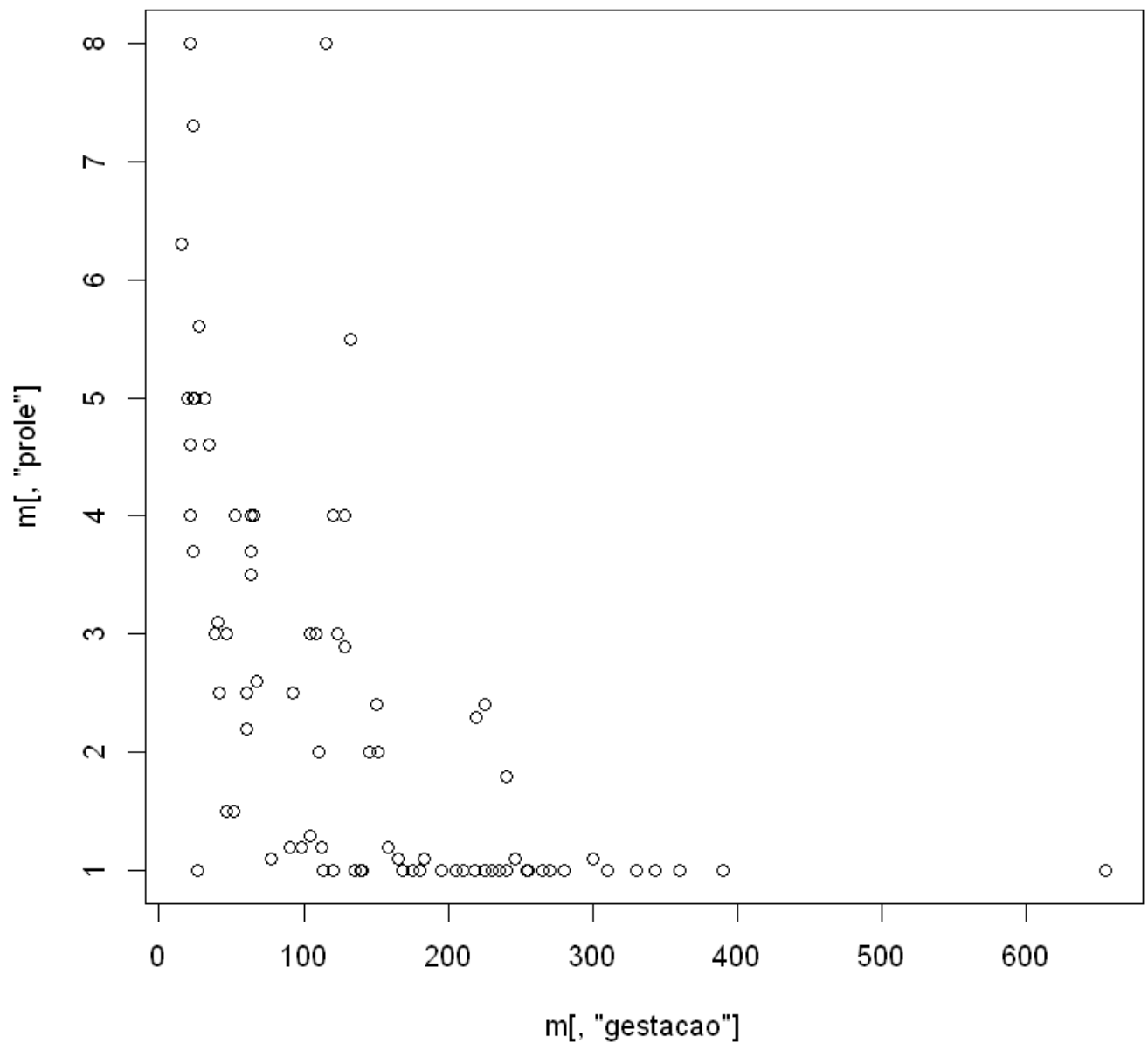
## Gerar Gráficos a partir dos Dados de uma Matriz

```
In [49]: # gráfico da relação entre as duas primeiras colunas (cerebro e corpo)
plot(m)
```



```
In [54]: # gráfico da relação entre as duas primeiras colunas (gestacao e prole)
# plot(m[, 3], m[, 4])
plot(m[, 'gestacao'], m[, 'prole'])
```





## 2. Array

Em R array é uma estrutura de dados tridimensional. Criamos um array através da função `array(x, dim)`, onde o parâmetro `x` é um vetor e `dim` são as dimensões do array.

```
In [58]: a <- array(c(1:24), dim = c(3, 3, 2))
```

```
In [59]: print(a)
```

```
, , 1
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

, , 2
```

```
[,1] [,2] [,3]  
[1,] 10 13 16  
[2,] 11 14 17  
[3,] 12 15 18
```

## Acessar Elementos do Array

```
In [60]: # Acessar a primeira tabela  
a[, , 1]
```

```
1 4 7
```

```
2 5 8
```

```
3 6 9
```

```
In [62]: # Acessar a primeira linha da tabela 1  
print(a[1, , 1])
```

```
[1] 1 4 7
```

```
In [69]: # Acessar a primeira coluna da segunda tabela  
print(a[, 1, 2])
```

```
[1] 10 11 12
```

## Operações com Arrays

```
In [70]: # Obter o maior valor da primeira tabela  
max(a[, , 1])
```

```
9
```

```
In [71]: # Obter a soma da primeira coluna da tabela 1  
sum(a[, 1, 1])
```

```
6
```

```
In [72]: # obter a média dos valores da segunda linha da segunda tabela  
mean(a[, 2, 2])
```

```
14
```

```
In [75]: # Obter a soma entre os valores da primeira coluna da table 1 com os da  
# primeira coluna da tabela 2  
sum(a[, 1, 1], a[, 1, 2])
```

```
39
```

```
In [74]: # obter a soma dos valores da diagonal da primeira tabelle  
sum(diag(a[, , 1]))
```

```
15
```

## Atribuir Nomes as Dimensões do Array

Assim como podemos atribuir nomes aos elementos de um vetor e as duas dimensões de uma matriz, também é possível o fazer para arrays. Para tal utilizamos a função `dimnames()`, passando como parâmetros três vetores com os nomes das linhas da matriz, nomes das colunas e nomes das matrizes.

```
In [76]: a <- array(c(1:24), # Vetor
                  dim = c(3, 3, 2), # Dimensões do array
                  dimnames = list(c('L1', 'L2', 'L3'), # Nome das linhas das matrizes
                                  c('C1', 'C2', 'C3'), # Nome das colunas das matrizes
                                  c('Matriz 1', 'Matriz 2'))) # Nomes das Matrizes

print(a)

, , Matriz 1

  C1 C2 C3
L1  1  4  7
L2  2  5  8
L3  3  6  9

, , Matriz 2

  C1 C2 C3
L1 10 13 16
L2 11 14 17
L3 12 15 18
```

## Inserir Atributo em um Array

```
In [77]: # inserir um atributo ao array "a"
attr(a, 'Observação') <- 'Meu primeiro array em R!!'
```

```
In [79]: # checar os atributos do array
print(attributes(a))

$dim
[1] 3 3 2

$dimnames
$dimnames[[1]]
[1] "L1" "L2" "L3"

$dimnames[[2]]
[1] "C1" "C2" "C3"

$dimnames[[3]]
[1] "Matriz 1" "Matriz 2"

$Observação
[1] "Meu primeiro array em R!!"
```

```
In [ ]:
```