Programming for Data Science

Matrices and Arrays in R language

Marco Beccuti

Università degli Studi di Torino

Dipartimento di Informatica

Matrix in R

- Matrix is an extension of vector to 2 dimensions.
  - it has rows and columns;
  - it is used for many purposes in statistics.

- To create a matrix 3×2 we can use function matrix:

$> x = rnorm(6)$ **random generation for the normal distribution**

$> x$

$[1] -0.1767643\ 1.5306045\ -0.3806768\ -0.1950407\ 0.3307676\ 0.8992097$

$> m = matrix(x, nrow = 3, ncol = 2)$

	[, 1]	[, 2]
[1,]	−0.1767643	−0.1950407
[2,]	1.5306045	0.3307676
[3,]	−0.3806768	0.8992097

Matrix in R

- input parameter byrow=TRUE means that the matrix is filled row by row rather than column by column.

$> x$

$[1] -0.1767643\ 1.5306045\ -0.3806768\ -0.1950407\ 0.3307676\ 0.8992097$

$> m = matrix(x, nrow = 3, ncol = 2)$

	[,1]	[,2]
[1,]	−0.1767643	−0.1950407
[2,]	1.5306045	0.3307676
[3,]	−0.3806768	0.8992097

$> m = matrix(x, nrow = 3, byrow = TRUE)$

	[,1]	[,2]
[1,]	−0.1767643	1.5306045
[2,]	−0.3806768	−0.1950407
[3,]	0.3307676	0.8992097

Matrix in R

- Useful functions for matrices include:
  nrow(), ncol(), t(), rownames(), colnames(),...

$> ncol(m)$
[1]2

$> nrow(m)$
[1]3

$> t(m)$ **transposition function: rows become columns and vice versa).**

	[,1]	[,2]	[,3]
[1,]	−0.1767643	−0.3806768	0.3307676
[2,]	1.5306045	−0.1950407	0.8992097

Matrix in R

- Useful functions for matrices include:
  nrow(), ncol(), t(), rownames(), colnames(),...

$> rownames(m) = c(\text{“}R1\text{”}, \text{“}R2\text{”}, \text{“}R3\text{”})$
$> m$

	[,1]	[,2]
R1	−0.1767643	1.5306045
R2	−0.3806768	−0.1950407
R3	0.3307676	0.8992097

$> colnames(m) = c(\text{“}C1\text{”}, \text{“}C2\text{”})$
$> m$

	C1	C2
R1	−0.1767643	1.5306045
R2	−0.3806768	−0.1950407
R3	0.3307676	0.8992097

Matrix in R

- Vector/Matrix and Vector/Matrix can be merged together by operators:
  cbind(), rbind

```
> x = 1:3
> y = 4:6
> cbind(x, y)
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

```
> x = 1:3
> y = 4:6
> rbind(x, y)
  [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

```
> cbind(y, x)
     [,1] [,2]
[1,]    4    1
[2,]    5    2
[3,]    6    3
```

```
> rbind(y, x)
  [,1] [,2] [,3]
[1,]    4    5    6
[2,]    1    2    3
```

Matrix in R

- Vector/Matrix and Vector/Matrix can be merged together by operators:
  cbind(), rbind
- Using cbind() It is not possible combine matrix with different number of rows;
- Using rbind() It is not possible combine matrix with different number of columns;
- Combining vectors with other vectors or matrices, short vectors are "recycled" to match long ones :

```
> x
[1]2.1 1.5 0.4 4.6
> y = 1 : 2
> cbind(y, x)
```

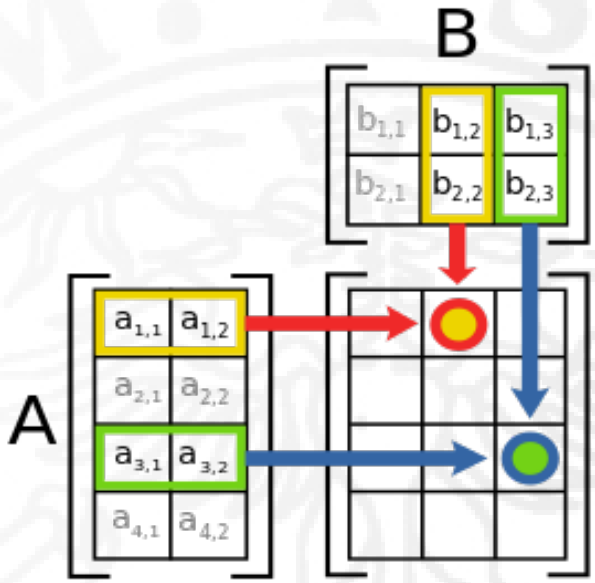
	[, 1]	[, 2]
[1,]	1	2.1
[2,]	2	1.5
[3,]	1	0.4
[4,]	2	4.6

```
> x
[1]2.1 1.5 0.4 4.6 7
> y = 1 : 2
> cbind(y, x)
```

Warning: number of rows of result is not a multiple of vector length

Matrix in R

Other functions for matrices include:



- $t(A)\%*\%B$ computes matrix multiplication;
- diag(n) creates a diagonal matrix with the values in the vector n on the diagonal;
- solve(A,B) solves the equation $a * x = b'$ for x vector;
- eigen(C) computes the eigenvalues and eigenvectors of C (e.g. stability analysis...)

Indexing matrix in R

Given a matrix m as follows:

	$C1$	$C2$
$R1$	-0.1767643	1.5306045
$R2$	-0.3806768	-0.1950407
$R3$	0.3307676	0.8992097

- then we can access the value in row 3, column 2 using:

  $> m[3, 2]$
  [1]0.8992097

  $> m[\text{“}R3\text{”}, \text{“}C2\text{”}]$
  [1]0.8992097
- to access multiple elements is possible as follows:

  $> m[, 2]$ **all elements in column 2.**
  [1]1.5306045 $-$ 0.1950407 0.8992097

  $> m[3,]$ **all elements in row 3.**
  [1]0.3307676 0.8992097

  $> m[c(1, 3), 2]$ **elements in column 2 and rows 1 and 3.**
  [1]1.5306045 0.8992097

Array in R

- An array is an extension of a matrix to more than 2 dimensions;
- Function array() can be used to create arrays:

  $> A1 = array(0, c(2, 2, 3))$ **create a 3d-array (dim. 2×2×3) with all elements 0.**

  $> a = rnorm(50)$
  $> A1 = array(a, c(2, 2, 3))$ **create a 3d-array from vector *a***

- Elements of multi-dimensional array can be indexed as those of a matrix:

  $> A1[2, ,]$ **Extracts the data in row 2 of the 3 matrices.**

  $> A1[, 3,]$ **Extracts the data in column 3 of the 3 matrices.**

  $> A1[, , 1]$ **Extracts the first matrix.**

  $> A1[1, 2, 3]$ **Extract element in row 1, column 2 and third matrix.**

Exercises on Matrices and Arrays

- Create a matrix A with values 10, 20, 30, 50, 4, 4 in column 1, values 1, 4, 2, 3, 2, 3 in column 2 and values 15, 11, 19, 5, 3 ,4 in column 3;
- Create a vector B with values 2.5, 3.5, 1.75, and combine A and B into a new matrix C using cbind();
- Combine A and B into a new matrix H using rbind();
- Determine the dimensions of C and H using dim() function;
- Compute the following matrix multiplication:

$$\begin{pmatrix} 1 & 4 & 3 \\ 0 & -2 & 8 \end{pmatrix} \times \begin{pmatrix} 1 & 9 \\ 2 & 17 \\ -6 & 3 \end{pmatrix}$$

Exercises on Matrices and Arrays

- Create a matrix A with values 10, 20, 30, 50, 4, 4 in column 1, values 1, 4, 2, 3, 2, 3 in column 2 and values 15, 11, 19, 5, 3 ,4 in column 3;

$> x = c(10, 20, 30, 50, 4, 4, 1, 4, 2, 3, 2, 3, 15, 11, 19, 5, 3, 4)$
$> A = matrix(x, ncol = 3)$

Exercises on Matrices and Arrays

- Create a vector B with values 2.5, 3.5, 1.75, and combine A and B into a new matrix C using cbind();

  $> B = c(2.5, 3.5, 1.75)$
  $> C = cbind(A, B)$

Exercises on Matrices and Arrays

- Combine A and B into a new matrix H using rbind();

$> B = c(2.5, 3.5, 1.75)$
$> C = rbind(A, B)$

Exercises on Matrices and Arrays

- Determine the dimensions of C and H using dim() function;

$> dim(C)$
[1]6 4

$> dim(H)$
[1]7 3

Exercises on Matrices and Arrays

- Compute the following matrix multiplication:

$$\begin{pmatrix} 1 & 4 & 3 \\ 0 & -2 & 8 \end{pmatrix} \times \begin{pmatrix} 1 & 9 \\ 2 & 17 \\ -6 & 3 \end{pmatrix}$$

$> x = c(1, 0, 4, -2, 3, 8)$
$> A = matrix(x, nrow = 2)$
$> y = c(1, 9, 2, 17, -6, 3)$
$> B = matrix(y, ncol = 2, byrow = T)$
$> A\%*\%B$

Exercises on Matrices and Arrays

- Solve the following equation system:

$$S = \begin{cases} x_1 + 2x_2 - 2x_3 = 1 \\ 2x_1 - x_2 + x_3 = 3 \\ x_1 + 3x_2 + x_3 = 1 \end{cases}$$

Exercises on Matrices and Arrays

- Solve the following equation system:

$$S = \begin{cases} x_1 + 2x_2 - 2x_3 = 1 \\ 2x_1 - x_2 + x_3 = 3 \\ x_1 + 3x_2 + x_3 = 1 \end{cases}$$

$> x = c(1, 2, -2, 2, -1, 1, 1, 3, 1)$
$> A = matrix(x, nrow = 3, byrow = T)$
$> B = c(1, 3, 1)$
$> solve(A, B)$