

데이터과학을 위한 **R**프로그래밍

2주차. 벡터, 행렬의 연산 및 함수



이혜선 교수

포항공과대학교 산업경영공학과



목차

2주차. 벡터, 행렬의 연산 및 함수

1차시

벡터 및 행렬 생성

2차시

벡터와 행렬의 연산

3차시

간단한 함수 생성 및 루프



2주차

2차시

벡터와 행렬의 연산

● 기본 연산

☑ 기본 연산 기호

Operator	Description
+	addition
-	subtraction
*	multiplication
/	division
^ or **	exponentiation
x %% y	modulus (x mod y) 5%%2 is 1
x %/ y	integer division 5%/2 is 2

Operator	Description
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	exactly equal to
!=	not equal to
!x	Not x
x y	x OR y
x & y	x AND y
isTRUE(x)	test if X is TRUE

● 기본 연산

☑ 더하기, 곱하기, 나누기 (몫, 나머지)

2의 3승 (2^3 , $2^{**}3$)

4의 3승 (4^3 , $4^{**}3$)

%% (7을 5로 나눴을 때 나머지)

%% (7을 5로 나눴을 때 몫)

```
# calculation
```

```
2^3
```

```
4**3
```

```
7%%5
```

```
7%/5
```

```
> 2^3
```

```
[1] 8
```

```
> 4**3
```

```
[1] 64
```

```
> 7%%5
```

```
[1] 2
```

```
> 7%/5
```

```
[1] 1
```

행렬의 연산

✓ 행렬의 연산

Operator or Function	Description
$A * B$	Element-wise multiplication
$A \%* \% B$	Matrix multiplication
$A \%o \% B$	Outer product. AB'
$\text{crossprod}(A,B)$ $\text{crossprod}(A)$	$A'B$ and $A'A$ respectively.
$t(A)$	Transpose
$\text{solve}(A, b)$	Returns vector x in the equation $b = Ax$ (i.e., $A^{-1}b$)
$\text{solve}(A)$	Inverse of A where A is a square matrix.
$\text{ginv}(A)$	Moore-Penrose Generalized Inverse of A . $\text{ginv}(A)$ requires loading the MASS package.
$y \leftarrow \text{eigen}(A)$	$y\$val$ are the eigenvalues of A $y\$vec$ are the eigenvectors of A

$y \leftarrow \text{svd}(A)$	Single value decomposition of A . $y\$d$ = vector containing the singular values of A . $y\$u$ = matrix with columns contain the left singular vectors of A . $y\$v$ = matrix with columns contain the right singular vectors of A .
$R \leftarrow \text{chol}(A)$	Choleski factorization of A . Returns the upper triangular factor, such that $R'R = A$.
$y \leftarrow \text{qr}(A)$	QR decomposition of A . $y\$qr$ has an upper triangle that contains the decomposition and a lower triangle that contains information on the Q decomposition. $y\$rank$ is the rank of A . $y\$qraux$ a vector which contains additional information on Q. $y\$pivot$ contains information on the pivoting strategy used.
$\text{cbind}(A,B,...)$	Combine matrices(vectors) horizontally. Returns a matrix.
$\text{rbind}(A,B,...)$	Combine matrices(vectors) vertically. Returns a matrix.
$\text{rowMeans}(A)$	Returns vector of row means.
$\text{rowSums}(A)$	Returns vector of row sums.
$\text{colMeans}(A)$	Returns vector of column means.
$\text{colSums}(A)$	Returns vector of column sums.

행렬의 연산

✓ 행렬 생성

```
# matrix example (2*5)  
m1<-matrix(1:10, nrow=2)  
m1
```



```
> m1<-matrix(1:10, nrow=2)  
> m1  
      [,1] [,2] [,3] [,4] [,5]  
[1,]    1    3    5    7    9  
[2,]    2    4    6    8   10
```

✓ 행렬의 차원 알아보기 : dim(..)

```
# dimension of m1  
dim(m1)
```



```
> dim(m1)  
[1] 2 5
```

행렬의 연산

전치 행렬(transpose) 구하기 (t)

```
m2<-matrix(1:6, ncol=3)
m2

# transpose of m2
tm2<-t(m2)
tm2
```



```
> m2<-matrix(1:6, ncol=3)
> m2
      [,1] [,2] [,3]
[1,]     1     3     5
[2,]     2     4     6
>
> # transpose of m2
> tm2<-t(m2)
> tm2
      [,1] [,2]
[1,]     1     2
[2,]     3     4
[3,]     5     6
```

전치행렬은 행과 열을 바꾼 행렬



m2는 (2*3)행렬,
tm2는 (3*2)행렬

행렬의 연산

☑ determinant 구하기 (det)

determinant 식

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$$

$$d1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

의 determinant는 -2

```
# determinant of matrix,  
d1<-matrix(1:4, nrow=2, byrow=T)  
d1  
det(d1)
```



```
> d1<-matrix(1:4, nrow=2, byrow=T)  
> d1  
      [,1] [,2]  
[1,]    1    2  
[2,]    3    4  
> det(d1)  
[1] -2
```

행렬의 연산

✓ 역행렬(inverse) 구하기 (solve)

$$d1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

의 역행렬 :

$$\text{inverse}(d1) = \begin{bmatrix} -2.0 & 1.0 \\ 1.5 & -0.5 \end{bmatrix}$$

```
#inverse of matrix
d1_inv<-solve(d1)
d1_inv

# d1*inv(d1)=identity matrix
d1%%d1_inv
```



```
> d1_inv<-solve(d1)
> d1_inv
      [,1] [,2]
[1,] -2.0  1.0
[2,]  1.5 -0.5
> d1%%d1_inv
      [,1] [,2]
[1,]  1 1.110223e-16
[2,]  0 1.000000e+00
```

$d1 * (d1 \text{의 역행렬}) = \text{단위행렬 (대각행렬이 1인 행렬)}$

행렬의 연산

✓ 역행렬을 이용한 방정식 해 구하기 (solve)

➤ $\text{solve}(A,b)$: $AX = b$ 의 해를 찾음

$$\begin{cases} 3x + 2y = 8 \\ x + y = 2 \end{cases}$$



$$A = \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}, X = \begin{bmatrix} x \\ y \end{bmatrix}, b = \begin{bmatrix} 8 \\ 2 \end{bmatrix}$$

방정식의 해를 구하기 위해
a(행렬)와 b(벡터)를 생성

solve함수를 이용해
x와 y의 해를 찾음
(답 : x=4, y=-2)

```
#solve equation
# 3x+2y=8, x+y=2

# matrix a, b
a <- matrix(c(3,1,2,1),nrow=2,ncol=2)
b <- matrix(c(8,2),nrow=2,ncol=1)
a
b
```



`solve(a,b)`



```
> solve(a,b)
      [,1]
[1,]     4
[2,]    -2
```

행렬의 연산

☑ 매뉴얼 보기 : `help(solve)`

The screenshot displays the R Studio interface. The top-left pane shows the R script editor with the following code:

```
44 # x=4, y=-2
45 solve(a,b)
46 # to see solve
47 help(solve)
48
49
```

The bottom-left pane shows the R console output:

```
> b <- matrix(c(8,2),nrow=2,ncol=1)
> a
      [,1] [,2]
[1,]    3    2
[2,]    1    1
> b
      [,1]
[1,]    8
[2,]    2
> solve(a,b)
      [,1]
[1,]    4
[2,]   -2
> help(solve)
> |
```

The right pane shows the R Documentation for the `solve` function, titled "Solve a System of Equations".

size chr [1:4] "S" "M" "L" "XL"

size factor Factor w/ 4 levels "S" "M" "L" "XL"

Files Plots Packages Help Viewer

R: Solve a System of Equations - Find on Topic

`solve` (base) R Documentation

Solve a System of Equations

Description

This generic function solves the equation $a \%*\% x = b$ for x , where b can be either a vector or a matrix.

Usage

```
solve(a, b, ...)
```

Default S3 method:

```
solve(a, b, tol, LINPACK = FALSE, ...)
```

Arguments

a	a square numeric or complex matrix containing the coefficients of the linear system. Logical matrices are coerced to numeric.
b	a numeric or complex vector or matrix giving the right-hand side(s) of the linear system. If missing, b is taken to be an identity matrix and <code>solve</code> will return the inverse of a .
tol	the tolerance for detecting linear dependencies in the columns of a . The default is <code>.Machine\$double.eps</code> . Not currently used with complex matrices a .
LINPACK	logical. Defunct and ignored.

행렬의 연산

☑ 고유치(eigenvalue)와 고유벡터(eigenvector)

```
# example for eigen value and eigen vector  
# already centered matrix  
x<-matrix(c(-3,-2,0, 1, 2, 2, -3, -3, 0, 2, 2, 2, 5,7,4,0,-5,-11), nrow =6, ncol=3)  
x  
dim(x)
```

(6*3)의 행렬 x,
행렬 x의 차원

```
> x  
      [,1] [,2] [,3]  
[1,]   -3   -3    5  
[2,]   -2   -3    7  
[3,]    0    0    4  
[4,]    1    2    0  
[5,]    2    2   -5  
[6,]    2    2  -11  
> dim(x)  
[1] 6 3
```

행렬의 연산

☑️ 고유치(eigenvalue)와 고유벡터(eigenvector)

```
# eigen value and eigen vector  
e1<-eigen(t(x)%*%x)  
e1
```

```
> e1<-eigen(t(x)%*%x)  
> e1  
eigen() decomposition  
$values  
[1] 273.546962 13.845220 0.607818  
  
$vectors  
      [,1]      [,2]      [,3]  
[1,] -0.2525343 0.5487321 0.79694382  
[2,] -0.2841664 0.7452586 -0.60319073  
[3,] 0.9249194 0.3787911 0.03227211
```

여기서 $t(x)\%*\%x$ 는 공분산 행렬이라고 할 수 있음