

2. 행렬(matrix)

행렬은 서로 다른 자료형을 가진 스칼라 값들을 2차원으로 나열한 자료구조이다. 즉, 행렬은 표 형태의 행(Row)과 열(Column)의 구조를 가진 데이터 유형이다. 벡터는 1차원 배열구조를 가지고 있지만 행렬은 2차원 구조를 가진다. 즉, 1차원 배열구조를 가진 벡터들의 집합이라고 할 수 있다.

행렬을 입력하기 위하여 **matrix()** 함수를 사용한다.

(1) 행렬 만들기

변수명 <- matrix(벡터 값, 행의 크기, 열의 크기)

변수명 <- matrix(벡터 값, nrow=3, ncol=4)

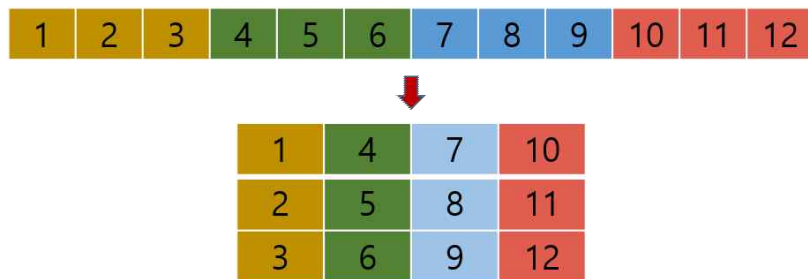
① 벡터를 열(column) 기준으로 행렬로 변환하기:

matrix(vector, byrow=FALSE, ncol=n)

② 벡터를 행(row) 기준으로 행렬로 변환하기:

matrix(vector, byrow=TRUE, ncol=n)

```
x <- matrix(1:12)
x
x <- matrix(1:12, 3, 4)      # a부터 b까지의 수(a:b)1)를 가지고 3×4 행렬
x                             # 기본값인 열부터 원소를 채운다.
matrix(1:12, 3, 4)
      [,1] [,2] [,3] [,4]
[1, ]    1    4    7   10
[2, ]    2    5    8   11
[3, ]    3    6    9   12
```



`x <- matrix(1:12, 3, 4, byrow=T)` # `byrow`²⁾ 옵션은 행부터 원소를 채우느냐에 대한 옵션이다. `byrow=F` 가 기본값(default)이다.

`x`

```
[ ,1] [ ,2] [ ,3] [ ,4]
[1, ] 1  2  3  4
[2, ] 5  6  7  8
[3, ] 9 10 11 12
```

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----



1	2	3	4
5	6	7	8
9	10	11	12

(2) 행렬의 행과 열에 명칭을 부여하기

`rownames(x) ; colnames(x)` # 행과 열 이름(명칭)을 설정하지 않으면, 값이 없음을 뜻하는 `NULL`이 출력됨

`NULL`

`NULL`

`x <- matrix(1:12, 3, 4, byrow=T); x`

`rownames(x) <- c("1행", "2행", "3행")` # 이름이 없는 행렬에 `colnames()`, `rownames()`를 활용하여 이름을 만들어 주자.

`colnames(x) <- c("1열", "2열", "3열", "4열")`

`x`

```
      1열  2열  3열  4열
1행   1   2   3   4
2행   5   6   7   8
3행   9  10  11  12
```

`rownames(x) <- c(1, 2, 3); x`

`rownames(x) <- NULL` # 행의 명칭 삭제하기

`t(x)`

- 1) 시작 값:종료 값의 형태, 이 경우 벡터(행렬)는 시작 값에서부터 종료 값에 이르기 전까지 1씩 증가한 실수들로 채워진다.
- 2) (연속) ...씩, (조금)씩, (정도·비율·차이) ...만큼, ...정도만큼, ...의 차로

```
x <- matrix(1:12, 3, 4, byrow=T, dimnames=list(c("1행","2행","3행"), c("1열","2열","3열","4열")))
```

dimnames는 list로 감싸주어야 한다. list의 첫 번째 요소 c("1행", ...)는 행의 이름, 두 번째 요소는 열의 이름을 뜻한다.

x

행렬에서 dimnames를 이용하여 행과 열의 이름을 지정하는 것보다는 rownames(), colnames()를 통해 행렬의 이름을 지정하는 방법을 자주 사용한다.

rownames(), colnames()를 통해 특정 행 또는 열의 이름만 바꿀 수 있다. 왜? 벡터이기 때문에

```
rownames(x)[2] <- "row2"
```

x

[실습] 다음 표를 R-코드로 작성하여라.

구분	height	weight
A	170	68.5
B	182	77.8
C	176	71.2
D	168	65.5

구분	A	B	C	D
height	170	182	176	168
weight	68.5	77.8	71.2	65.5

(3) 행렬을 벡터로 변환하기(vectorization)

① 행렬을 열(column) 기준으로 벡터로 변환하기: `as.vector(matrix)`

② 행렬을 행(row) 기준으로 벡터로 변환하기: `as.vector(t(matrix))`

```
# matrix into vector(vectorization) by column : as.vector( )
```

```
x <- matrix(1:15, 5, 3) # byrow 옵션이 없으므로 기본값인 열부터 원소를 채운다.
```

x

```
      [,1] [,2] [,3]
[1,]    1    6   11
[2,]    2    7   12
[3,]    3    8   13
[4,]    4    9   14
[5,]    5   10   15
```

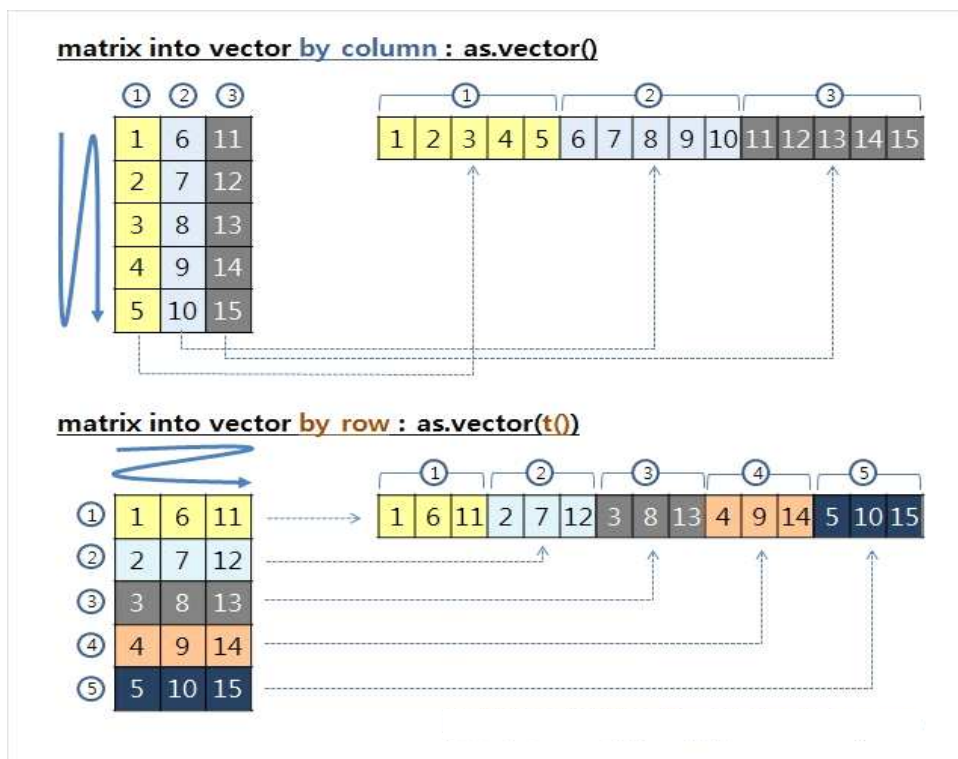
```
t(x) # t는 전치행렬(transpose matrix)
```

```

      [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    6    7    8    9   10
[3,]   11   12   13   14   15
y <- as.vector(x) ; y
[1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15

z <- as.vector(t(x)) ; z           # x를 행 기준으로 벡터로 전환하기는
                                   # t(x)를 열 기준으로 벡터로 변환하기
[1]  1  6 11  2  7 12  3  8 13  4  9 14  5 10 15

```



(4) 행렬의 연산

```

x <- matrix(1:6, 2, 3) ; x
y <- matrix(c(2, 0, -1, 3, 4, -2), 3, 2) ; y
x %*% y           # 행렬 x와 y의 곱 (행렬이 곱이 정의되어야 함)
x %*% t(x)        # 행렬 x와 전치행렬 t(x)의 곱. %*% (반드시 붙여서 입력!)
3*y              # 행렬의 스칼라 배, 모든 원소에 3배
x+y              # 행렬 x와 y의 덧셈 (두 행렬의 크기가 같아야 함)
y*y              # 행렬 y의 각 원소 제곱

```

[실습] 어떤 스포츠 입장권은 두 판매처 A 와 B 에서 판매되었다. 예선전과 결승전의 입장권은 좌석의 위치에 따라 1 등급, 2 등급, 3 등급으로 구분되어 있으며 가격은 다음과 같다. (단위: 달러)

구분	1 등급	2 등급	3 등급
결승전	500	250	150
예선전	150	100	60

구분	A	B
1 등급	5	8
2 등급	10	12
3 등급	15	15

(1) 위의 행렬을 R-코드로 작성하여라.

(2) 이때, 전체 가격표를 계산하기 위한 R-코드로 작성하여라.

구분	A	B
결승전	7,250	9,250
예선전	2,650	3,300

(5) 행렬 데이터의 접근

행렬은 색인 또는 행과 열의 이름을 통해 접근할 수 있다.

$A[m, n]$: 행렬 A 의 m 행, n 열에 저장된 값. 이때 m 이나 n 에 벡터를 사용해 여러 값을 지정할 수 있음. m 이나 n 중 하나를 생략하면 전체 행 또는 전체 열을 의미함.

```
z <- matrix(c(1,-1,5,2,1,0,-3,0,1,4,-3,0), nrow=4, byrow=T) ; z
```

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

z[3,2] # 3행 2열 성분을 출력 [행의 위치, 열의 위치]

z[1,] ; z[,3] # 1행을 모두 원소를 출력/ 3열을 모두 원소를 출력

z[,2, drop=F] # 행과 열을 출력할 때 행렬의 형태를 보존해서 출력하고 싶은 경우 **drop=F(FALSE)**를 해 준다.

z[-2,] ; z[, -3] # 2행을 제외한 모든 행을 출력 / 3열을 제외한 모두 열을 출력

z[,2:3] # 모든 행의 2열과 3열을 모두 출력

```

dim(z) # dim은 dimension의 약자이며, 행렬 z의 사이즈 행과 열의 크기를 출력
dim(z)[1] # 행렬 z의 행의 크기 출력
dim(z)[2] # 행렬 z의 열의 크기 출력
dim(z)[3] # NA를 출력
rowSums(z) # 각각의 행의 합을 출력
colSums(z) # 각각의 열의 합을 출력
rowMeans(z) # 각각의 행의 평균을 출력
colMeans(z) # 각각의 열의 평균을 출력
apply(z, 1, sum) # 행렬 z의 모든 행들의 합을 출력
apply(z, 2, sum) # 행렬 z의 모든 열들의 합을 출력
apply(z, 1, mean) # 행렬 z의 모든 행들의 평균을 출력
apply(z, 2, mean) # 행렬 z의 모든 열들의 평균을 출력

```



apply() 안에 첫 번째는 행렬인수, 두 번째 값이 1일 때는 행 기준, 2일 때는 열 기준이고, 세 번째 인수는 함수가 들어감(사용자 정의 함수도 가능)



apply 함수 외에도 비슷한 함수가 **tapply**, **sapply**, **lapply** 등등 여러 가지가 있다.

[실습] 어떤 스포츠 입장권은 두 판매처 A와 B에서 판매되었다. 예선전과 결승전의 입장권은 좌석의 위치에 따라 1등급, 2등급, 3등급으로 구분되어 있으며 가격은 다음과 같다. (단위: 달러)

구분	1 등급	2 등급	3 등급
결승전	500	250	150
예선전	150	100	60

구분	A	B
1 등급	5	8
2 등급	10	12
3 등급	15	15

- (1) 판매처 B가 입장권을 판매한 총금액을 구하는 R-코드를 작성하여라.
- (2) 판매처 A가 결승전 경기에 판매한 총금액을 구하는 R-코드를 작성하여라.

구분	A	B
결승전	7,250	9,250
예선전	2,650	3,300

(6) 벡터의 결합으로 행렬 만들기

`rbind()`와 `cbind()`를 사용한다.

```
x1 <- c(1,-1,3) ; y1 <- c(0,4,2)
m1 <- rbind(x1, y1) ; m1      # 각각의 벡터들이 각각의 행으로 간주됨

m1 <- rbind(c(1,-1,3), c(0,4,2)) ; m1    # 2×3 행렬
      [,1] [,2] [,3]
[1, ]    1  -1    3
[2, ]    0   4    2
x2 <- c(1,-1,3) ; y2 <- c(0,4,2)
m2 <- cbind(x2, y2) ; m2      # 각각의 벡터들이 각각의 열로 간주됨

m2 <- cbind(c(1,-1,3), c(0,4,2)) ; m2    # 3×2 행렬
      [,1] [,2]
[1, ]    1    0
[2, ]   -1    4
[3, ]    3    2
```

`rbind`라는 것은 row+bind의 약자. 행으로 묶으라는 뜻. 즉 벡터 (1,-1,3), 벡터 (0,4,2)를 행으로 묶어라! 따라서 위의 결과처럼 2×3 matrix의 형태로 출력.

□ 벡터의 결합에 의한 행렬

```
v1 <- c(1, 2, 3, 4)
v2 <- c(5, 6, 7, 8)
v3 <- c(9, 10, 11, 12)
```

```
x <- cbind(v1, v2, v3)
x
```

벡터 v1, v2, v3을 열 단위로 합침
⇒ cbind는 column bind를 의미

출력 결과

	v1	v2	v3
[1,]	1	5	9
[2,]	2	6	10
[3,]	3	7	11
[4,]	4	8	12

```
y <- rbind(v1, v2, v3)
y
```

벡터 v1, v2, v3을 행 단위로 합침
⇒ rbind는 row bind를 의미

출력 결과

	[,1]	[,2]	[,3]	[,4]
v1	1	2	3	4
v2	5	6	7	8
v3	9	10	11	12

```
rownames(x) <- c("1행", "2행", "3행", "4행") # x 행렬에 행의 이름 추가
colnames(x) <- c("1열", "2열", "3열")        # x 행렬에 열의 이름 추가
```