



# R 특강

R의 vector, matrix, array

# R에서의 array, vector, matrix



## array

- 동일한 데이터 타입을 갖는 데이터의 다차원 구조 (1차원 이상일 수 있음)
- 동일한 데이터 타입이란? integer, float, logical, character 등의 같은 데이터 타입으로 된 것
- 1차원 array 대신 vector 를 주로 사용
- 2차원 array 대신 matrix 을 주로 사용

## vector

- 동일한 데이터 타입을 갖는 데이터의 1차원 구조 (각 데이터 접근을 위치 번호로 함)

## matrix

- 동일한 데이터 타입을 갖는 데이터의 2차원 구조 (각 데이터 접근을 행, 열번호로 함)

# R에서의 1차원 array 구조



## 1차원 array

- 1개의 행 또는 열로 볼 수 있지만, 그 보다는 여러 개의 element를 갖는 1차원 구조로 이해하는 것이 좋음
- 1개 행 또는 1개 열을 갖는 2차원과 혼란이 있을 수 있기 때문에 조심해야 함

1	2	3	4
1번	2번	3번	4번

1차원 array  
(4개 element)

1	1번
2	2번
3	3번
4	4번

```
a1 <- array(1:4, c(4))
```

```
a1[3] # 3
```

array 요소 개수

접근할 요소의 위치 번호

vector

```
v <- 1:4
```

```
v[3] # 3
```

접근할 요소의 위치 번호

# R에서의 2차원 array 구조

## 2차원 array

1이상의 행, 열을 가지고 있으며 구조는 항상 (행, 열)의 순서로 표기함

1행	1	2	3	4
	1열	2열	3열	4열

2차원 array  
(1행 4열)

```
a21r <- array(1:4, c(1, 4))  
a21r[1, 3]
```

접근할 요소의 행,열 번호

	1열
1행	1
2행	2
3행	3
4행	4

2차원 array  
(4행 1열)

```
a21c <- array(1:4, c(4, 1))  
a21c[3, 1]
```

접근할 요소의 행,열 번호

1행	1	3	5	7
2행	2	4	6	8
	1열	2열	3열	4열

2차원 array  
(2행 4열)

```
a2 <- array(1:8, c(2, 4))  
a2[1, 3]
```

접근할 요소의 행,열 번호



# R에서의 matrix 구조



matrix(data, nrow=1, ncol=1, byrow=FALSE)

- 동일한 데이터 타입을 갖는 데이터의 2차원 구조로 기본은 열 우선 채우기이지만 행 우선 채우기 가능
- nrow, ncol을 생략가능 (생략 시 자동계산, 모두 생략 시 nrow=데이터 개수, ncol=1 의 2차원 구조)

1행	1	3	5	7
2행	2	4	6	8
	1열	2열	3열	4열

m2a<- matrix(1:8, nrow=2) nrow : matrix에서 행의 수 (열의 수 자동계산)  
m2a[1, 3] # 5

m2b <- matrix(1:8, ncol=4)  
m2b[1, 3] # 5 ncol : matrix에서 열의 수 (행의 수 자동계산)  
접근할 데이터의 [행, 열] 번호

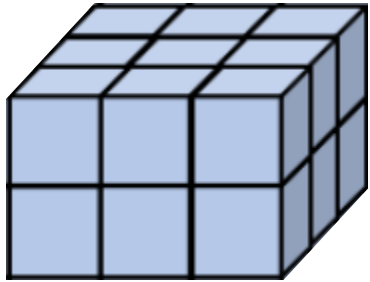
1행	1	2	3	4
2행	5	6	7	8
	1열	2열	3열	4열

m2c <- matrix(1:8, nrow=2, byrow=TRUE)  
m2c[1, 3] # 3 matrix 채우기를 row 우선으로 함  
접근할 데이터의 [행, 열] 번호

# R에서의 3차원 array 구조

## 3차원 array

- 1이상의 행, 열, 면을 가지고 있으며 구조는 항상 (행, 열, 면)의 순서로 표기함



3차원 array  
2행 3열 3면

1면	2면	3면
1행 2 3 4 5 6	7 8 9 10 11 12	13 14 15 16 17 18
1열 2열 3열		

```
a3 <- array(1:18, c(2,3,3))
```

```
a3[1, 3, 2] # 11
```

array에서 (행, 열, 면)의 수

접근할 데이터의 [행, 열, 면] 번호