



문공 A0015

# R 프로그래밍

김 태 완

kimtwan21@dongduk.ac.kr

# 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임
  - 1차원 데이터와 2차원 데이터

몸무게
62.4
75.3
59.8
72.1

1차원 데이터

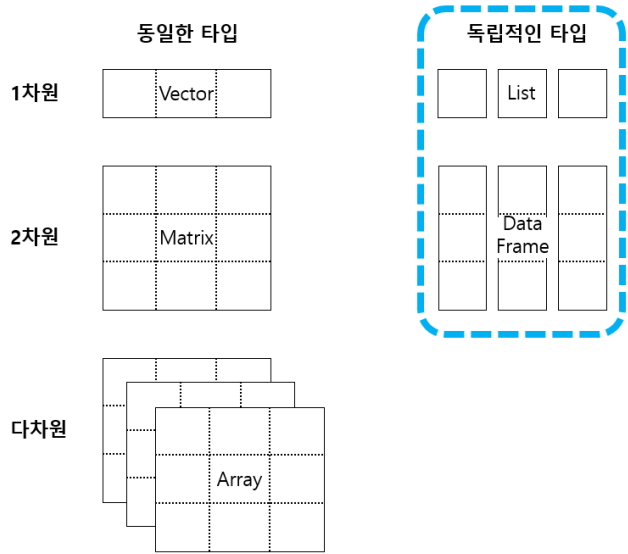
행/관측값	열/변수			
	키	몸무게	나이	
	168.4	62.4	29	
	174.6	75.3	31	
	169.5	59.8	25	
	181.4	72.1	36	
셀				

2차원 데이터

- 행(row)/관측 값(observation) : 가로줄 방향
- 열(column)/변수(variable) : 세로줄 방향
- 셀(cell) : 사각형 영역

# 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임
  - 1차원 데이터 : 벡터, 리스트, 팩터
  - 2차원 데이터 : 매트릭스, 데이터프레임



	벡터	리스트	매트릭스	데이터프레임
데이터의 형태	1차원 데이터		2차원 데이터	
데이터의 종류	동일한 자료형	서로 다른 자료형 가능	동일한 자료형	서로 다른 자료형 가능
R에서 제공하는 데이터셋			<u>state.x77</u> ↓ 미국 50개 주에 대한 통계	<u>iris</u> ↓ 150그루의 붓꽃에 대한 통계

# 매트릭스와 데이터프레임

- 매트릭스

- 매트릭스 만들기 : 매트릭스의 모든 셀에 저장되는 값은 동일한 자료형 이어야 함
  - 기본적인 매트릭스 만들기 : `matrix()` 함수 이용

1에서 20까지  
매트릭스에 저장

열의 수 = 5

```
z <- matrix(1:20, nrow = 4, ncol = 5)
```

z

행의 수 = 4

	[ ,1]	[ ,2]	[ ,3]	[ ,4]	[ ,5]
[1, ]	1	5	9	13	17
[2, ]	2	6	10	14	18
[3, ]	3	7	11	15	19
[4, ]	4	8	12	16	20

열 방향으로 채워짐

```
z <- matrix(1:20, nrow = 4, ncol = 5, byrow = TRUE)
```

z

매트릭스에 저장될 값들을  
행 방향으로 채움

	[ ,1]	[ ,2]	[ ,3]	[ ,4]	[ ,5]
[1, ]	1	2	3	4	5
[2, ]	6	7	8	9	10
[3, ]	11	12	13	14	15
[4, ]	16	17	18	19	20

행 방향으로 채워짐

# 매트릭스와 데이터프레임

- 매트릭스

- 매트릭스 만들기 : 매트릭스의 모든 셀에 저장되는 값은 동일한 자료형 이어야 함
  - 기본적인 매트릭스 만들기 : `matrix()` 함수 이용

```
z <- matrix(0, 3, 4)
```

```
z2 <- matrix(NA, 2, 5)
```

```
v <- 1:12
```

```
matrix(data=v, ncol=4)
```

```
mat <- matrix(data=v, ncol=4)
```

```
str(mat)
```

```
dim(mat)
```

```
nrow(mat) = dim(mat)[1] # 행의 개수
```

```
ncol(mat) = dim(mat)[2] # 열의 개수
```

```
length(mat) # 셀의 개수
```

# 매트릭스와 데이터프레임

- 매트릭스

- 매트릭스 만들기 : 매트릭스의 모든 셀에 저장되는 값은 동일한 자료형 이어야 함
  - 기존 매트릭스에 벡터를 추가하여 새로운 매트릭스 만들기 : `cbind()` 함수, `rbind()` 함수 이용

```
1 x <- 1:3
2 y <- 4:6
3 m1 <- cbind(x, y)
# cbind() 함수 :
  두 개의 벡터를 열 방향으로 bind
4 cbind(1:3, 4:6, matrix(7:12, 3,2))
5 m2 <- rbind(x, y)
# rbind() 함수 :
  두 개의 벡터를 행 방향으로 bind
6 rbind(matrix(1:6, 2, 3), matrix(7:12, 2, 3))
```



```
> x <- 1:3
> y <- 4:6
> m1 <- cbind(x, y)
> m1
      x y
[1, ] 1 4
[2, ] 2 5
[3, ] 3 6
> m2 <- rbind(x, y)
> m2
      [,1] [,2] [,3]
x      1    2    3
y      4    5    6
```

# 매트릭스와 데이터프레임

- 매트릭스

- 매트릭스 만들기 : 매트릭스의 모든 셀에 저장되는 값은 동일한 자료형 이어야 함
  - 기존 매트릭스에 벡터를 추가하여 새로운 매트릭스 만들기 : `cbind()` 함수, `rbind()` 함수 이용

```
1 x <- 1:3
2 y <- 4:6
3 m1 <- cbind(x, y)
4 m2 <- rbind(x, y)
5 m3 <- cbind(m1, x)
6 m3
# [벡터] + [매트릭스]
```



```
> x <- 1:3
> y <- 4:6
> m1 <- cbind(x, y)
> m2 <- rbind(x, y)
> m3 <- cbind(m1, x)
> m3
      x y x
[1, ] 1 4 1
[2, ] 2 5 2
[3, ] 3 6 3
           
      m1  x
```

# 매트릭스와 데이터프레임

- 매트릭스

- 매트릭스 만들기 : 매트릭스의 모든 셀에 저장되는 값은 동일한 자료형 이어야 함
  - 기존 매트릭스에 벡터를 추가하여 새로운 매트릭스 만들기 : `cbind()` 함수, `rbind()` 함수 이용

```
1 x <- 1:3
2 y <- 4:6
3 m2 <- rbind(x, y)
4 m3 <- cbind(m1, x)
5 m4 <- rbind(m2, m3)
6 m4
# [매트릭스] + [매트릭스]
```



```
> x <- 1:3
> y <- 4:6
> m2 <- rbind(x, y)
> m3 <- cbind(m1, x)
> m4 <- rbind(m2, m3)
> m4
```

	x	y	x
x	1	2	3
y	4	5	6
	1	4	1
	2	5	2
	3	6	3

Blue brackets on the right side of the table indicate that the first two rows (x and y) correspond to matrix m2, and the next three rows correspond to matrix m3.



# 매트릭스와 데이터프레임

- 매트릭스

- 매트릭스에서의 값 추출

- 인덱스 값을 이용하여 매트릭스에서의 값 추출하기 : 매트릭스[행,열]

```
1 z <- matrix(1:15, nrow = 3, ncol = 5)
2 z행
3 z[2, 3]열
# z의 2행 3열에 있는 값
4 z[1, 4]
# z의 1행 4열에 있는 값
5 z[2, ]
# z의 2행에 있는 모든 값
6 z[, 4]
# z의 4열에 있는 모든 값
```



```
> z <- matrix(1:15, nrow = 3, ncol = 5)
> z
      [,1] [,2] [,3] [,4] [,5]
[1,]  1   4   7  10  13
[2,]  2   5   8  11  14
[3,]  3   6   9  12  15
> z[2, 3]
[1] 8
> z[1, 4]
[1] 10
> z[2, ]
[1] 2 5 8 11 14
> z[, 4]
[1] 10 11 12
```

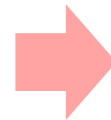
# 매트릭스와 데이터프레임

- 매트릭스

- 매트릭스에서의 값 추출

- 매트릭스에서 여러 개의 값을 동시에 추출하기 : 매트릭스[행,열]

```
1  z <- matrix(1:15, nrow = 3, ncol = 5)
2  z
3  z [2, 1:3]
# 2행의 값 중 1~3열에 있는 값
4  z [1, c(1, 2, 4)]
# 1행의 값 중 1, 2, 4열에 있는 값
5  z [c(1,3), ]
# 1, 3행에 있는 모든 값
```



```
> z <- matrix(1:15, nrow = 3, ncol = 5)
> z
      [,1] [,2] [,3] [,4] [,5]
[1,]  1   4   7  10  13
[2,]  2   5   8  11  14
[3,]  3   6   9  12  15
> z[2, 1:3]
[1]  2  5  8
> z[1, c(1, 2, 4)]
[1]  1  4 10
> z[c(1,3), ]
      [,1] [,2] [,3] [,4] [,5]
[1,]  1   4   7  10  13
[2,]  3   6   9  12  15
```

# 매트릭스와 데이터프레임

- 매트릭스

- 매트릭스의 행과 열에 이름 지정 : rownames( ), colnames( ) 함수 이용

```
1 score <- matrix(c(90, 85, 69,  
                    85, 96, 49),  
                  nrow = 3, ncol = 2)  
2 score  
3 rownames(score) <- c("A", "B", "C")  
# 행에 이름을 지정  
4 colnames(score) <- c("English", "Math")  
# 열에 이름을 지정  
5 score
```



```
> score <- matrix(c(90, 85, 69,  
                    85, 96, 49),  
                  nrow = 3, ncol = 2)  
  
> score  
      [,1] [,2]  
[1, ] 90   85  
[2, ] 85   96  
[3, ] 69   49  
  
> rownames(score) <- c("A", "B", "C")  
> colnames(score) <- c("English", "Math")  
> score  
      English Math  
A         90    85  
B         85    96  
C         69    49
```

# 매트릭스와 데이터프레임

- 매트릭스

- 매트릭스의 행과 열에 이름 지정 : `rownames( )`, `colnames( )` 함수 이용
  - 이름을 이용하여 매트릭스 값 추출하기

```
1 score
2 score['A', 'Math']
# A의 수학성적
3 score['B', ]
# B의 모든 과목 성적
4 rownames(score)
# score 행의 이름
5 colnames(score)
# score 열의 이름
6 rownames(score)[2]
# score 행의 이름 중 두 번째 값
```



```
> score
  English Math
A     90     85
B     85     96
C     69     49
> score['A', 'Math']
[1] 85
> score['B', ]
  English Math
      85     96
> rownames(score)
[1] 'A' 'B' 'C'
> colnames(score)
[1] 'English' 'Math'
> rownames(score)[2]
[1] 'B'
```

# 매트릭스와 데이터프레임

## • 데이터프레임

- 데이터프레임 만들기 : 서로 다른 자료형의 값들이 함께 저장될 수 있음
  - 단, 같은 열의 값들의 자료형은 동일해야 함
  - `data.frame()` 함수 이용

```
1 city <- c('Seoul', 'Tokyo', 'Washington')  
# 문자형 벡터  
2 rank <- c(1, 3, 2)  
# 숫자형 벡터  
3 city.info <- data.frame(city, rank)  
4 city.info
```



```
> city <- c('Seoul', 'Tokyo', 'Washington')  
> rank <- c(1, 3, 2)  
> city.info <- data.frame(city, rank)  
> city.info
```

	city	rank
1	Seoul	1
2	Tokyo	3
3	Washington	2

열 방향으로 결합

문자형

숫자형

같은 열의 자료형은 동일

# 매트릭스와 데이터프레임

- 데이터프레임

- 데이터프레임 만들기 : data.frame( ) 함수 이용

```
1 v1 <- c('Seoul', 'Tokyo', 'Washington')
2 v2 <- c(1, 3, 2)
3 v3 <- c("KOR", "JPN", "USA")
4 city.info <- data.frame(city=v1, rank=v2, nation=v3)
5 str(city.info)

6 mat <- matrix(c(1,3,5,7,9,2,4,6,8,10,2,3,5,7,11), ncol=3)
7 as.data.frame(mat) # 행렬을 데이터프레임으로 변환
8 mat2 <- as.data.frame(mat)
9 colnames(mat2) <- c("first", "second", "third")
10 mat2
```

# 매트릭스와 데이터프레임

- 데이터프레임

- matrix와 동일하게 cbind( )함수, rbind( )함수 이용 가능

```
1 code <- c("ds001", "ds002", "ds003")
2 name <- c("math", "r", "statistics")
3 students <- c(50,40,30)

4 ds <- data.frame(code, name, students)
5 ds <- rbind(ds, c("ds004", "seminar", 100))

6 new <- data.frame(code=c("ds005","ds006"), name=c("coding"), students=c(56,77))
7 ds <- rbind(ds, new)
8 ds$professor = c("taewan","taewan2","taewan3","taewan4","taewan5","taewan6") #cbind 동일
```

# 매트릭스와 데이터프레임

## • 데이터프레임

- Iris 데이터셋 : R에 저장된 데이터 셋으로, 대표적인 데이터프레임

→ 외부에서 불러올 필요 없이 'iris'라고 입력하면 내용이 출력됨.

> iris

	꽃받침의 길이	꽃받침의 폭	꽃잎의 길이	꽃잎의 폭	붓꽃의 품종
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa

숫자형

문자형

...(이하생략)



# 매트릭스와 데이터프레임

- 데이터프레임

- 데이터프레임의 값 추출 : 매트릭스와 동일

```
1 iris[,c(1:2)]  
# 1, 2열의 모든 데이터  
2 iris[,c("Sepal.Length", "Species")]  
# 1, 5열의 모든 데이터  
3 iris[1:3,c(2,4)]  
# 1~3행의 데이터 중 2, 4열의 데이터
```



```
> iris[,c(1:2)]  
  Sepal.Length Sepal.Width  
1          5.1          3.5  
2          4.9          3.0  
3          4.7          3.2  
...  
> iris[,c("Sepal.Length", "Species")]  
  Sepal.Length Species  
1          5.1    setosa  
2          4.9    setosa  
3          4.7    setosa  
...  
> iris[1:3,c(2,4)]  
  Sepal.Width Petal.Width  
1          3.5          0.2  
2          3.0          0.2  
3          3.2          0.2
```

# 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기

- 매트릭스와 데이터프레임에 대한 산술연산

- 숫자로 구성된 매트릭스나 데이터프레임에 대해서 산술연산 적용 가능
- 자료구조 상에서 동일 위치에 있는 값들끼리 계산
- 자료구조의 크기(행과 열의 개수)가 동일해야 함.

```
> a <- matrix(1:6, 2, 3)
> b <- matrix(7:12, 2, 3)
> a
     [,1] [,2] [,3]
[1,]  1   3   5
[2,]  2   4   6
> b
     [,1] [,2] [,3]
[1,]  7   9  11
[2,]  8  10  12
```

```
> a + b
     [,1] [,2] [,3]
[1,]  8  12  16
[2,] 10  14  18
> b - a
     [,1] [,2] [,3]
[1,]  6   6   6
[2,]  6   6   6
```

```
> a * 2      # a값은 변경되지 않음
     [,1] [,2] [,3]
[1,]  2   6  10
[2,]  4   8  12
> a <- a * 2  # a값이 변경됨
> a
     [,1] [,2] [,3]
[1,]  2   6  10
[2,]  4   8  12
```

## 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기
  - 매트릭스와 데이터프레임에서 사용하는 함수
    - 데이터셋의 기본 정보 확인

함수명	설명
dim( )	데이터셋의 행과 열의 개수 출력
nrow( )	데이터셋의 행의 개수 출력
ncol( )	데이터셋의 열의 개수 출력
rownames( )	데이터셋의 행의 이름 출력
colnames( )	데이터셋의 열의 이름 출력
head( )	데이터셋의 시작 부분의 일부 데이터(1 ~ 6행) 출력
tail( )	데이터셋의 끝부분의 일부 데이터(n-5 ~ n행) 출력

# 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기
  - 매트릭스와 데이터프레임에서 사용하는 함수
    - 데이터셋의 기본 정보 확인 예시

```
> dim(iris)                                     # iris의 행과 열의 개수
[1] 150 5

> nrow(iris)                                    # iris의 행의 개수
[1] 150

> ncol(iris)                                    # iris의 열의 개수
[1] 5

> rownames(iris)                               #iris의 행의 이름
[1] 1 2 3 4 5 6 7 8 9 (...) 150

> colnames(iris)                              #iris의 열의 이름
[1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"
```

# 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기
  - 매트릭스와 데이터프레임에서 사용하는 함수
    - 데이터셋의 기본 정보 확인 예시

> head(iris) # iris의 시작부분 6개 데이터

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

> tail(iris) # iris의 끝부분 6개 데이터

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	6.7	3.3	5.7	2.5	virginica
2	6.7	3.0	5.2	2.3	virginica
3	6.3	2.5	5.0	1.9	virginica
4	6.5	3.0	5.2	2.0	virginica
5	6.2	3.4	5.4	2.3	virginica
6	5.9	3.0	5.1	1.8	virginica

## 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기
  - 매트릭스와 데이터프레임에서 사용하는 함수
    - 데이터셋의 기본 정보 확인

함수명	설명
str( )	데이터셋에 대한 전반적인 정보를 요약하여 출력
unique( )	어떤 그룹을 나타내는 자료에서 중복된 값을 제거하여 출력
table( )	자료에서 각 그룹별로 몇 개의 관측치가 존재하는지 출력

# 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기
  - 매트릭스와 데이터프레임에서 사용하는 함수
    - 데이터셋의 기본 정보 확인 예시

```
> str(iris) 150개의 행과 5개의 열로 구성
'data.frame' : 150 obs. of 5 variables :
iris의 자료구조 : 데이터프레임
              자료형 : 숫자형
$ Sepal.Length : num [5.1 4.9 4.7 ...] → 저장된 값
첫 번째 열의 이름 : Sepal.Length
$ Sepal.Width : num 3.5 3 3.2 ...
$ Petal.Length : num 1.4 1.4 1.3 ...
$ Petla.Width : num 0.2 0.2 0.2 ...
              자료형 : 문자형(팩터)
$ Species : Factor w/ 3 levels "setosa", "versicolor", "virginica" : 1 1 1...
              with 3 levels : 3가지 종류
```

## 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기
  - 매트릭스와 데이터프레임에서 사용하는 함수
    - 데이터셋의 기본 정보 확인 예시

```
> unique(iris[,5])  
[1] setosa versicolor virginica  
Levels : setosa versicolor virginica  
> table(iris[, "Species"])  
setosa versicolor virginica  
  50      50      50
```

# iris의 5번째 열(품종)의 종류 보기(중복 제거)

# iris의 5번째 열(품종)의 종류별 행의 개수 세기



## 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기
  - 매트릭스와 데이터프레임에서 사용하는 함수

함수명	설명
colSums( )	자료의 열별 합계 출력
colMeans( )	자료의 열별 평균 출력
rowSums( )	자료의 행별 합계 출력
rowMeans( )	자료의 행별 평균 출력
t( )	데이터의 행과 열의 방향을 바꾸어 출력(전치 행렬)
subset( )	전체 데이터에서 조건에 맞는 행들만 추출

## 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기
  - 매트릭스와 데이터프레임에서 사용하는 함수

```
> colSums(iris[ ,-5])                                     # 끝부분 5열의 합
Sepal.Length Sepal.Width Petal.Length Petal.Width Species
      876.5         458.6       563.7         179.9
> colMeans(iris[ ,-5])                                     # 끝부분 5열의 평균
Sepal.Length Sepal.Width Petal.Length Petal.Width Species
    5.843333    3.057333    3.758000    1.199333
> rowSums(iris[ ,-5])                                     # 끝부분 5행의 합
[1] 10.2 9.5 9.4 10.2 ...
> rowMeans(iris[ ,-5])                                    # 끝부분 5행의 평균
[1] 2.550 2.375 2.350 2.550 ...
```

## 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기
  - 매트릭스와 데이터프레임에서 사용하는 함수

```
> z <- matrix(1:15, nrow = 3, ncol = 5)
```

```
> z
```

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

```
> t(z)
```

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

# z의 행과 열의 방향 변환(전치 행렬)

## 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기
  - 매트릭스와 데이터프레임에서 사용하는 함수

데이터를 추출할 조건 : (1) Sepal.Length의 값이 5보다 큼  
(2) Sepal.Width의 값이 4보다 큼 ] 2개의 조건을 &(and)로 연결

```
> IR_1 <- subset(iris, Sepal.Length > 5.0 & Sepal.Width > 4.0)
```

데이터를 추출할 대상 : iris

```
> IR_1 # 조건을 만족하는 행을 출력
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
16	5.7	4.4	1.5	0.4	setosa
33	5.2	4.1	1.5	0.1	setosa
34	5.5	4.2	1.4	0.2	setosa

# subset( )함수의 경우 매트릭스에서는 작동이 잘 되지 않는 경우 발생  
데이터 프레임으로 변환 필요할 수 있음

## 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기
  - 매트릭스와 데이터프레임에서 사용하는 함수
    - 매트릭스와 데이터프레임의 자료구조 확인

함수명	설명
<code>class( )</code>	데이터셋의 자료구조의 종류를 출력
<code>is.matrix( )</code>	데이터셋이 매트릭스인지 확인
<code>is.data.frame( )</code>	데이터셋이 데이터프레임인지 확인

# 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기
  - 매트릭스와 데이터프레임에서 사용하는 함수
    - 매트릭스와 데이터프레임의 자료구조 확인

```
> class(iris)                                # iris의 자료구조 확인
[1] "data.frame"
> class(state.x77)                            # state.x77의 자료구조 확인
[1] "matrix"
> is.matrix(iris)                             # iris가 매트릭스인지 확인
[1] FALSE
> is.data.frame(iris)                         # iris가 데이터프레임인지 확인
[1] TRUE
> is.matrix(state.x77)                       # state.x77이 매트릭스인지 확인
[1] TRUE
> is.data.frame(state.x77)                   # state.x77이 데이터프레임인지 확인
[1] FALSE
```

# 매트릭스와 데이터프레임

- 매트릭스와 데이터프레임 다루기
  - 매트릭스와 데이터프레임의 자료구조 변환
    - 매트릭스를 데이터프레임으로 변환 : `data.frame( )` 함수 이용
    - 데이터프레임을 매트릭스로 변환 : `as.matrix( )` 함수 이용
    - 단, 데이터프레임의 모든 값의 자료형이 동일해야 매트릭스로 변환 가능

```
> st <- data.frame(state.x77)
# state.x77 매트릭스를 데이터프레임으로 변환
> class(st)
[1] "data.frame"
> iris_m <- as.matrix(iris[,1:4])
# iris 데이터프레임에서 숫자형인 1~4열만 매트릭스로 변환
> class(iris_m)
[1] "matrix"
```

## 예시

- 예시 1 : 매트릭스 만들기

```
1 d <- matrix(  
2 d
```

- 실행 결과

```
> d <- matrix(  
> d  
      [,1] [,2] [,3] [,4]  
[1, ]  5   20   35   50  
[2, ] 10   25   40   55  
[3, ] 15   30   45   60
```



## 예시

- 예시 2 : 매트릭스 d 다루기

```
1 d <- matrix((c(seq(5,60,5))), nrow = 3, ncol = 4)
2 d_t <- matrix(
3 d_t
```

- 실행 결과

```
> d <- matrix((c(seq(5,60,5))), nrow = 3, ncol = 4)
> d_t <- matrix(
> d_t                                     # d의 행과 열의 방향이 바뀜
      [,1] [,2] [,3] [,4]
[1, ]  5  10  15  20
[2, ] 25  30  35  40
[3, ] 45  50  55  60
```

## 예시

- 예시 3 : 새로운 매트릭스 만들기

```
1  b1 <- c(4,5,7,2)
2  b2 <- c(19,15,12,17)
3  b3 <- c(20,24,28,23)
# cording here #
```

- 실행 결과

```
> b1 <- c(4,5,7,2)
> b2 <- c(19,15,12,17)
> b3 <- c(20,24,28,23)
> m1 <- 
> m1                                     # b1, b2, b3 벡터 결합
      [,1] [,2] [,3]
[1, ]  4   19  20
[2, ]  5   15  24
[3, ]  7   12  28
[4, ]  2   17  23
```

## 예시

- 예시 4 : 매트릭스 다루기

```
1 rich_state.x77 <-   
2 rich_state.x77
```

- 실행 결과

```
> rich_state.x77 <-   
> rich_state.x77 # state.x77 매트릭스에서 income 값이 5000 이상인 데이터 출력
```

	Population	Income	Illiteracy	Life Exp
Alabama	3615	3624	2.1	69.05
Alaska	365	6315	1.5	69.31
Arizona	2212	4530	1.8	70.55

## 예시

- 예시 5 : 데이터프레임 생성하기

```
1 x <-  
2 y <-  
3 z <-  
4 df <-  
5 df
```

- 실행 결과

```
> x <-  
> y <-  
> z <-  
> df <-  
> df
```

	x	y	z
1	1	white	TRUE
2	2	blue	FALSE
3	3	yellow	TRUE

# x는 숫자형 벡터  
# y는 문자형 벡터  
# z는 논리형 벡터  
# x, y, z 벡터 결합하기

## 예시

- 예시 6 : 데이터프레임 다루기

```
1 class(airquality)
2 airquality_m <- 
3 class(airquality_m)
```

- 실행 결과

```
> class(airquality)           # airquailty : R에서 제공하는 일별 대기의 질을 측정한 데이터셋
[1] "data.frame"                 # 자료구조는 데이터프레임
> airquality_m <- 
> class(airquality_m)         # airquailty의 자료구조를 매트릭스로 변환
[1] "matrix" "array"
```

## 예시

- 예시 7 : 데이터프레임 다루기

```
1 nrow
```

```
2 subset
```

- 실행 결과

```
> nrow # 오존(Ozone)농도가 120을 넘는 날이 며칠인지 출력
```

```
[1] 3
```

```
> subset
```

```
  Ozone Solar.R Wind Temp Month Day
```

```
120   76   203   9.7   97    8   28
```

```
# 기온이 가장 높은 날의 데이터를 출력
```

## 예시

- 예시 8 : 데이터프레임 다루기

```
1 airquality  
2 airquality
```

- 실행 결과

```
> airquality  
      Wind  
1    7.4  
2    8.0  
3   12.6  
...
```

```
> airquality  
[1] 7.4 8.0 12.6 ...
```

# airquality의 "Wind" 열 추출  
# 결과 = 데이터프레임

# airquality의 "Wind" 열 추출  
# 결과 = 벡터

감사합니다

[kimtwan21@dongduk.ac.kr](mailto:kimtwan21@dongduk.ac.kr)

김 태 완