



국립공주대학교
KONG JU NATIONAL UNIVERSITY


03-메트릭스와 데이터 프레임

R프로그래밍

소프트웨어학과
김정은



목차

- **행렬** 
- 데이터 프레임
- 행렬과 데이터 프레임 다루기
- 파일 데이터 읽기/쓰기

데이터의 개념

- 1차원 데이터 : '몸무게' 데이터와 같은 단일 주제의 데이터 → 벡터
- 2차원 데이터 : '키', '몸무게', '나이' 와 같은 여러 주제의 데이터
→ 행렬, 데이터 프레임

| 몸무게 | 키 | 몸무게 | 나이 |
|------|-------|------|----|
| 62.4 | 168.4 | 62.4 | 29 |
| 65.3 | 169.5 | 65.3 | 27 |
| 59.8 | 172.1 | 59.8 | 26 |
| 46.5 | 185.2 | 46.5 | 25 |
| 49.8 | 173.7 | 49.8 | 26 |
| 58.7 | 175.2 | 58.7 | 28 |

(a) 1차원 데이터 (b) 2차원 데이터

그림 3-1 1차원 데이터와 2차원
데이터

- 행렬(matrix): 데이터 테이블의 모든 셀의 값들이 동일한 자료형
- 데이터 프레임(data frame): 자료형이 다른 컬럼들로 구성

데이터의 개념(계속)

열(column), 컬럼,
변수(variable)

| name | age | job | office | M_F |
|------|-----|-----|--------|-----|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

셀(cell)

행(row)
관측값(observation)

그림 3-2 데이터 테이블에 사용하는 용어

기본 행렬 만들기

- 2차원 테이블 형태의 자료구조로, 행렬의 모든 셀에 저장되는 값은 동일한 자료형이어야 함

코드 3-1

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

매트릭스 z의 내용을 출력

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

매트릭스 z의 내용을 출력

| | [,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 |

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

매트릭스에
저장될 값

행의 수

열의 수

행렬에 저장될 값들을 행 방향으로 채우기

코드 3-2

```
z2 <- matrix(1:20, nrow=4, ncol=5, byrow=T)
z2                                     # 매트릭스 z2의 내용을 출력
```

```
> z2 <- matrix(1:20, nrow=4, ncol=5, byrow=T)
> z2                                     # 매트릭스 z2의 내용을 출력

     [,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
```

기존 행렬에 벡터를 추가하여 새로운 행렬 만들기

코드 3-3

```
x <- 1:4          # 벡터 x 생성
y <- 5:8          # 벡터 y 생성
z <- matrix(1:20, nrow=4, ncol=5) # 매트릭스 z 생성

m1 <- cbind(x,y)   # x와 y를 열 방향으로 결합하여 매트릭스 생성
m1                # 매트릭스 m1의 내용을 출력
m2 <- rbind(x,y)   # x와 y를 행 방향으로 결합하여 매트릭스 생성
m2                # 매트릭스 m2의 내용을 출력
m3 <- rbind(m2,x)   # 매트릭스 m2와 벡터 x를 행 방향으로 결합
m3                # 매트릭스 m3의 내용을 출력
m4 <- cbind(z,x)    # 매트릭스 z와 벡터 x를 열 방향으로 결합
m4                # 매트릭스 m4의 내용을 출력
```

인덱스값을 이용하여 행렬에서의 값 추출

- 행렬에서 특정 위치에 있는 값을 추출하는 방법은 벡터와 유사
- 값들의 위치를 나타내는 인덱스를 사용하는데, 2차원상에서 위치를 지정하려면 2개 필요

코드 3-4

```
z <- matrix(1:20, nrow=4, ncol=5) # 매트릭스 z 생성
z                                  # 매트릭스 z의 내용 출력

z[2,3]                            # 2행 3열에 있는 값
z[1,4]                            # 1행 4열에 있는 값
z[2,]                             # 2행에 있는 모든 값
z[4,]                             # 4열에 있는 모든 값
```

```
> z <- matrix(1:20, nrow=4, ncol=5) # 매트릭스 z 생성
> z                                  # 매트릭스 z의 내용 출력

  [,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
```


인덱스값을 이용하여 행렬에서의 값 추출(계속)

```
> z[2,3]           # 2행 3열에 있는 값
[1] 10
> z[1,4]           # 1행 4열에 있는 값
[1] 13
> z[2,]            # 2행에 있는 모든 값
[1]  2  6 10 14 18
> z[,4]            # 4열에 있는 모든 값
[1] 13 14 15 16
```

| | [,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 |

그림 3-3 매트릭스에서 값의 위치 지정

행렬에서 여러 개의 값을 동시에 추출

코드 3-5

```
z <- matrix(1:20, nrow=4, ncol=5)      # 매트릭스 z 생성
z                                         # 매트릭스 z의 내용 출력

z[2,1:3]                                # 2행의 값 중 1~3열에 있는 값
z[1,c(1,2,4)]                           # 1행의 값 중 1, 2, 4열에 있는 값
z[1:2,]                                  # 1, 2행에 있는 모든 값
z[,c(1,4)]                              # 1, 4열에 있는 모든 값
```

```
> z <- matrix(1:20, nrow=4, ncol=5)      # 매트릭스 z 생성
> z                                         # 매트릭스 z의 내용 출력

      [,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
>
> z[2,1:3]                                # 2행의 값 중 1~3열에 있는 값
[1]  2  6 10
> z[1,c(1,2,4)]                           # 1행의 값 중 1, 2, 4열에 있는 값
[1]  1  5 13
> z[1:2,]                                  # 1, 2행에 있는 모든 값

      [,1] [,2] [,3] [,4] [,5]
[1,]   1   5   9  13  17
[2,]   2   6  10  14  18
> z[,c(1,4)]                              # 1, 4열에 있는 모든 값

      [,1] [,2]
[1,]   1  13
[2,]   2  14
[3,]   3  15
[4,]   4  16
```

행렬의 행과 열에 이름을 지정하는 방법

코드 3-6

```
score <- matrix(c(90,85,69,78,  
                 85,96,49,95,  
                 90,80,70,60),  
               nrow=4, ncol=3)  
  
score  
rownames(score) <- c('John','Tom','Mark','Jane')  
colnames(score) <- c('English','Math','Science')  
score
```

```
> score <- matrix(c(90,85,69,78,  
+                 85,96,49,95,  
+                 90,80,70,60),  
+                 nrow=4, ncol=3)
```

```
> score  
      [,1] [,2] [,3]  
[1,]   90   85   90  
[2,]   85   96   80  
[3,]   69   49   70  
[4,]   78   95   60
```

```
> rownames(score) <- c('John','Tom','Mark','Jane')  
> colnames(score) <- c('English','Math','Science')
```

```
> score  
      English Math Science  
John      90   85     90  
Tom       85   96     80  
Mark      69   49     70  
Jane      78   95     60
```

행과 열에 지정한 이름을 이용하여 행렬 값 추출

코드 3-7

```
score['John','Math']          # John의 수학 성적
score['Tom',c('Math','Science')] # Tom의 수학, 과학 성적
score['Mark',]                # Mark의 모든 과목 성적
score[, 'English']            # 모든 학생의 영어 성적
rownames(score)                # score의 행의 이름
colnames(score)                # score의 열의 이름
colnames(score)[2]             # score의 열의 이름 중 두 번째 값
```

```
> score['John','Math']          # John의 수학 성적
[1] 85

> score['Tom',c('Math','Science')] # Tom의 수학, 과학 성적
   Math Science
   96      80

> score['Mark',]                # Mark의 모든 과목 성적
English  Math Science
   69    49    70


> score[, 'English']            # 모든 학생의 영어 성적
John Tom Mark Jane
   90  85  69  78

> rownames(score)                # score의 행의 이름
[1] "John" "Tom" "Mark" "Jane"

> colnames(score)                # score의 열의 이름
[1] "English" "Math" "Science"

> colnames(score)[2]             # score의 열의 이름 중 두 번째 값
[1] "Math"
```

목차

- 행렬
- 데이터 프레임 ← 
- 행렬과 데이터 프레임 다루기
- 파일 데이터 읽기/쓰기

데이터 프레임의 개념

- 숫자형 벡터, 문자형 벡터 등 서로 다른 형태의 데이터를 2차원 데이터 테이블 형태로 묶을 수 있는 자료구조
- 외관상으로는 행렬과 차이가 없지만 행렬에 저장되는 모든 값들이 동일한 자료형인 것과는 달리 데이터 프레임에는 서로 다른 자료형의 값들이 함께 저장

| 키 | 몸무게 |
|-------|------|
| 168.4 | 62.4 |
| 169.5 | 65.3 |
| 172.1 | 59.8 |
| 185.2 | 46.5 |
| 173.7 | 49.8 |
| 175.2 | 58.7 |

| 키 | 몸무게 | 성별 |
|-------|------|----|
| 168.4 | 62.4 | M |
| 169.5 | 65.3 | F |
| 172.1 | 59.8 | F |
| 185.2 | 46.5 | M |
| 173.7 | 49.8 | M |
| 175.2 | 58.7 | F |

(a) 매트릭스의 예

(b) 데이터프레임의 예

그림 3-4 매트릭스와 데이터프레임의 예

데이터 프레임 만들기

코드 3-8

```
city <- c("Seoul","Tokyo","Washington")    # 문자로 이루어진 벡터
rank <- c(1,3,2)                            # 숫자로 이루어진 벡터
city.info <- data.frame(city, rank)          # 데이터프레임 생성
city.info                                    # city.info의 내용 출력
```

```
> city <- c("Seoul","Tokyo","Washington")    # 문자로 이루어진 벡터
> rank <- c(1,3,2)                            # 숫자로 이루어진 벡터
> city.info <- data.frame(city, rank)          # 데이터프레임 생성
> city.info                                    # city.info의 내용 출력

  city rank
1  Seoul  1
2  Tokyo  3
3 Washington 2
```

iris 데이터 셋

- R에서 제공하는 실습용 데이터 셋 중의 하나로 데이터 프레임으로 되어 있음
- 150 종류의 붓꽃에 대해 4개 분야의 측정 데이터와 품종 정보를 결합하여 만든 데이터 셋

```
> iris
```

| | 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 |
| ... | (이하생략) | | | | |

iris 데이터 셋(계속)

| 열 이름 | 의미 | 자료형 |
|--------------|---------|---------|
| Sepal.Length | 꽃받침의 길이 | 숫자형 |
| Sepal.Width | 꽃받침의 폭 | 숫자형 |
| Petal.Length | 꽃잎의 길이 | 숫자형 |
| Petal.Width | 꽃잎의 폭 | 숫자형 |
| Species | 붓꽃의 품종 | 문자형(팩터) |

코드 3-9

표 3-1 iris

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

iris 데이터 셋(계속)

```
> iris[,c(1:2)]
```

```
# 1, 2열의 모든 데이터
```

```
  Sepal.Length Sepal.Width  
1           5.1         3.5  
2           4.9         3.0  
3           4.7         3.2  
...(생략)
```

```
> iris[,c(1,3,5)]
```

```
# 1, 3, 5열의 모든 데이터
```

```
  Sepal.Length Petal.Length  Species  
1           5.1           1.4   setosa  
2           4.9           1.4   setosa  
3           4.7           1.3   setosa  
...(생략)
```

```
> iris[,c("Sepal.Length", "Species")]
```

```
# 1, 5열의 모든 데이터
```

```
  Sepal.Length  Species  
1           5.1   setosa  
2           4.9   setosa  
3           4.7   setosa  
...(생략)
```

```
> iris[1:5,]
```

```
# 1~5행의 모든 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
```

```
> iris[1:5,c(1,3)]
```

```
# 1~5행의 데이터 중 1, 3열의 데이터
```

```
  Sepal.Length Petal.Length  
1           5.1           1.4  
2           4.9           1.4  
3           4.7           1.3  
4           4.6           1.5  
5           5.0           1.4
```

목차

- 행렬
- 데이터 프레임
- **행렬과 데이터 프레임 다루기**
- 파일 데이터 읽기/쓰기



iris 데이터 셋의 기본 내용 확인하기

- 행렬과 데이터 프레임은 모두 2차원 형태의 데이터를 저장하는 자료구조이기 때문에 다루는 방법이 대부분 동일
- 데이터 프레임인 iris 데이터 셋을 대상으로 학습을 하지만 여기서 배우는 내용은 행렬에도 동일하게 적용

코드 3-10

```
dim(iris)      # 행과 열의 개수 출력
nrow(iris)     # 행의 개수 출력
ncol(iris)     # 열의 개수 출력
colnames(iris) # 열 이름 출력, names()와 결과 동일
head(iris)     # 데이터셋의 앞부분 일부 출력
tail(iris)     # 데이터셋의 뒷부분 일부 출력
```

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

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

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

iris 데이터 셋의 기본 내용 확인하기(계속)

```
> colnames(iris)           # 열 이름 출력, names()와 결과 동일
[1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
[5] "Species"

> head(iris)               # 데이터셋의 앞부분 일부 출력
  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)               # 데이터셋의 뒷부분 일부 출력
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
145          6.7         3.3          5.7          2.5 virginica
146          6.7         3.0          5.2          2.3 virginica
147          6.3         2.5          5.0          1.9 virginica
148          6.5         3.0          5.2          2.0 virginica
149          6.2         3.4          5.4          2.3 virginica
150          5.9         3.0          5.1          1.8 virginica
```

iris 데이터 셋의 추가적인 내용 확인하기

코드 3-11

```
str(iris)           # 데이터셋 요약 정보 보기
iris[,5]            # 품종 데이터 보기
unique(iris[,5])    # 품종의 종류 보기(중복 제거)
table(iris[, "Species"]) # 품종의 종류별 행의 개수 세기
```

```
> str(iris)           # 데이터셋 요약정보 보기
'data.frame':150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
1 1 ...
```

iris 데이터 셋의 추가적인 내용 확인하기(계속)

```
> iris[,5] # 품종 데이터 보기
[1] setosa setosa setosa setosa setosa setosa
[7] setosa setosa setosa setosa setosa setosa
...(중간 생략)
[49] setosa setosa versicolor versicolor versicolor versicolor
[55] versicolor versicolor versicolor versicolor versicolor versicolor
...(중간 생략)
[97] versicolor versicolor versicolor versicolor virginica virginica
...(중간 생략)
[145] virginica virginica virginica virginica virginica virginica
Levels: setosa versicolor virginica
> unique(iris[,5]) # 품종의 종류 보기(중복 제거)
[1] setosa versicolor virginica
Levels: setosa versicolor virginica
> unique(iris[,5]) # 품종의 종류 보기(중복 제거)
[1] setosa versicolor virginica
Levels: setosa versicolor virginica
> table(iris[, "Species"]) # 품종의 종류별 행의 개수 세기

setosa versicolor virginica
50 50 50
```

행렬과 데이터 프레임에서 사용하는 함수

- 행별, 열별 합계와 평균 계산

코드 3-12

```
colSums(iris[,-5])      # 열별 합계
colMeans(iris[,-5])     # 열별 평균
rowSums(iris[,-5])      # 행별 합계
rowMeans(iris[,-5])     # 행별 평균
```

```
> colSums(iris[,-5])      # 열별 합계
Sepal.Length Sepal.Width Petal.Length Petal.Width
           876.5          458.6          563.7          179.9

> colMeans(iris[,-5])     # 열별 평균
Sepal.Length Sepal.Width Petal.Length Petal.Width
      5.843333      3.057333      3.758000      1.199333

> rowSums(iris[,-5])      # 행별 합계
[1] 10.2  9.5  9.4  9.4 10.2 11.4  9.7 10.1  8.9  9.6 10.8 10.0
[13]  9.3  8.5 11.2 12.0 11.0 10.3 11.5 10.7 10.7 10.7  9.4 10.6
...(중간 생략)
[133] 17.0 15.7 15.7 19.1 17.7 16.8 15.6 17.5 17.8 17.4 15.5 18.2
[145] 18.2 17.2 15.7 16.7 17.3 15.8

> rowMeans(iris[,-5])     # 행별 평균
[1] 2.550 2.375 2.350 2.350 2.550 2.850 2.425 2.525 2.225 2.400
[11] 2.700 2.500 2.325 2.125 2.800 3.000 2.750 2.575 2.875 2.675
...(중간 생략)
[131] 4.550 5.025 4.250 3.925 3.925 4.775 4.425 4.200 3.900 4.375
[141] 4.450 4.350 3.875 4.550 4.550 4.300 3.925 4.175 4.325 3.950
```


행렬과 데이터 프레임에서 사용하는 함수(계속)

- 행과 열의 방향 전환

코드 3-13

```
z <- matrix(1:20, nrow=4, ncol=5)
z
t(z)                # 행과열 방향 전환
```

```
> z <- matrix(1:20, nrow=4, ncol=5)
> z
      [,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
> t(z)                # 행과열 방향 전환
      [,1] [,2] [,3] [,4]
[1,]   1    2    3    4
[2,]   5    6    7    8
[3,]   9   10   11   12
[4,]  13   14   15   16
[5,]  17   18   19   20
```

행렬과 데이터 프레임에서 사용하는 함수(계속)

- 조건에 맞는 행과 열의 값 추출

코드 3-14

```
IR.1 <- subset(iris, Species=="setosa")  
IR.1  
IR.2 <- subset(iris, Sepal.Length>5.0 &  
              Sepal.Width>4.0)  
IR.2  
IR.2[, c(2,4)]      # 2, 4열의 값만 추출
```

- **iris**
데이터를 추출하는 대상이 `iris` 데이터셋이다.
- **Species=="setosa"**
데이터를 추출할 조건을 지정하는 부분으로, 품종 열의 값이 "setosa"인 열만 추출하라는 의미이다.

행렬과 데이터 프레임에서 사용하는 함수(계속)

```
> IR.1 <- subset(iris, Species=="setosa")
```

```
> IR.1
```

| | 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 |
| ...(중간 생략) | | | | | |
| 49 | 5.3 | 3.7 | 1.5 | 0.2 | setosa |
| 50 | 5.0 | 3.3 | 1.4 | 0.2 | setosa |

```
> IR.2 <- subset(iris, Sepal.Length>5.0 &  
+               Sepal.Width>4.0)
```

```
> IR.2
```

| | 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 |

```
> IR.2[, c(2,4)] # 2, 4열의 값만 추출
```

| | Sepal.Width | Petal.Width |
|----|-------------|-------------|
| 16 | 4.4 | 0.4 |
| 33 | 4.1 | 0.1 |
| 34 | 4.2 | 0.2 |

행렬과 데이터 프레임에서 사용하는 함수(계속)

- 행렬과 데이터 프레임에서 산술연산

코드 3-15

```
a <- matrix(1:20,4,5)
b <- matrix(21:40,4,5)
a
b

2*a           # 매트릭스 a에 저장된 값들에 2를 곱하기
b-5
2*a + 3*b

a+b
b-a
b/a
a*b

a <- a*3
b <- b-5
```

행렬과 데이터 프레임의 자료구조 확인

코드 3-16

```
class(iris)           # iris 데이터셋의 자료구조 확인
class(state.x77)      # state.x77 데이터셋의 자료구조 확인
is.matrix(iris)       # 데이터셋이 매트릭스인지를 확인하는 함수
is.data.frame(iris)   # 데이터셋이 데이터프레임인지를 확인하는 함수
is.matrix(state.x77)
is.data.frame(state.x77)
```

```
> class(iris)           # iris 데이터셋의 자료구조 확인
[1] "data.frame"
> class(state.x77)      # state.x77 데이터셋의 자료구조 확인
[1] "matrix"
> is.matrix(iris)       # 데이터셋이 매트릭스인지를 확인하는 함수
[1] FALSE
> is.data.frame(iris)   # 데이터셋이 데이터프레임인지를 확인하는 함수
[1] TRUE
> is.matrix(state.x77)
[1] TRUE
> is.data.frame(state.x77)
[1] FALSE
```

행렬과 데이터 프레임의 자료구조 반환

코드 3-17

매트릭스를 데이터프레임으로 변환

```
st <- data.frame(state.x77)
```

```
head(st)
```

```
class(st)
```

데이터프레임을 매트릭스로 변환

```
iris.m <- as.matrix(iris[,1:4])
```

```
head(iris.m)
```

```
class(iris.m)
```

```
> # 매트릭스를 데이터프레임으로 변환
```

```
> st <- data.frame(state.x77)
```

```
> head(st)
```

| | Population | Income | Illiteracy | Life.Exp | Murder | HS.Grad | Frost | Area |
|----------|------------|--------|------------|----------|--------|---------|-------|--------|
| Alabama | 3615 | 3624 | 2.1 | 69.05 | 15.1 | 41.3 | 20 | 50708 |
| Alaska | 365 | 6315 | 1.5 | 69.31 | 11.3 | 66.7 | 152 | 566432 |
| Arizona | 2212 | 4530 | 1.8 | 70.55 | 7.8 | 58.1 | 15 | 113417 |
| Arkansas | 2110 | 3378 | 1.9 | 70.66 | 10.1 | 39.9 | 65 | 51945 |

| | | | | | | | | |
|------------|-------|------|-----|-------|------|------|-----|--------|
| California | 21198 | 5114 | 1.1 | 71.71 | 10.3 | 62.6 | 20 | 156361 |
| Colorado | 2541 | 4884 | 0.7 | 72.06 | 6.8 | 63.9 | 166 | 103766 |

```
> class(st)
```

```
[1] "data.frame"
```

```
> # 데이터프레임을 매트릭스로 변환
```

```
> iris.m <- as.matrix(iris[,1:4])
```

```
> head(iris.m)
```

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

```
> class(iris.m)
```

```
[1] "matrix"
```

데이터 프레임의 열 추출

코드 3-18

```
iris[, "Species"] # 결과=벡터. 매트릭스와 데이터프레임 모두 가능
iris[, 5]         # 결과=벡터. 매트릭스와 데이터프레임 모두 가능
iris["Species"]   # 결과=데이터프레임. 데이터프레임만 가능
iris[5]           # 결과=데이터프레임. 데이터프레임만 가능
iris$Species      # 결과=벡터. 데이터프레임만 가능
```

데이터 프레임의 열 추출(계속)

```
> iris[, "Species"]
[1] setosa    setosa    setosa    setosa    setosa    setosa    setosa
[8] setosa    setosa    setosa    setosa    setosa    setosa    setosa
...(중간 생략)
[141] virginica virginica virginica virginica virginica virginica virginica
[148] virginica virginica virginica
Levels: setosa versicolor virginica

> iris[, 5]
[1] setosa    setosa    setosa    setosa    setosa    setosa    setosa
[8] setosa    setosa    setosa    setosa    setosa    setosa    setosa
...(중간 생략)
[141] virginica virginica virginica virginica virginica virginica virginica
[148] virginica virginica virginica
Levels: setosa versicolor virginica

> iris["Species"]
      Species
1      setosa
2      setosa
3      setosa
...(이하 생략)

> iris[5]
```

```
      Species
1      setosa
2      setosa
3      setosa
...(이하 생략)

> iris$Species
[1] setosa    setosa    setosa    setosa    setosa    setosa    setosa
[8] setosa    setosa    setosa    setosa    setosa    setosa    setosa
...(이하 생략)
[134] virginica virginica virginica virginica virginica virginica virginica
[141] virginica virginica virginica virginica virginica virginica virginica
[148] virginica virginica virginica
Levels: setosa versicolor virginica
```


목차

- 행렬
- 데이터 프레임
- 행렬과 데이터 프레임 다루기
- **파일 데이터 읽기/쓰기**



파일 형식 변환

- 엑셀 파일에 테이블 형태의 데이터가 저장되어 있는 경우를 가정
- 엑셀 파일을 .csv 형태로 변환하여 저장 후 R에서 .csv 파일을 읽음
- 읽어온 파일은 데이터 프레임 형태로 저장됨

| | A | B | C | D | E | F |
|---|-------|---------|------|------|-------|-----|
| 1 | Ozone | Solar.R | Wind | Temp | Month | Day |
| 2 | 41 | 190 | 7.4 | 67 | 5 | 1 |
| 3 | 36 | 118 | 8 | 72 | 5 | 2 |
| 4 | 12 | 149 | 12.6 | 74 | 5 | 3 |
| 5 | 18 | 313 | 11.5 | 62 | 5 | 4 |
| 6 | NA | NA | 14.3 | 56 | 5 | 5 |
| 7 | 28 | NA | 14.9 | 66 | 5 | 6 |
| 8 | 23 | 299 | 8.6 | 65 | 5 | 7 |
| 9 | 19 | 99 | 13.8 | 59 | 5 | 8 |

그림 3-5 대기의 질 데이터 파일

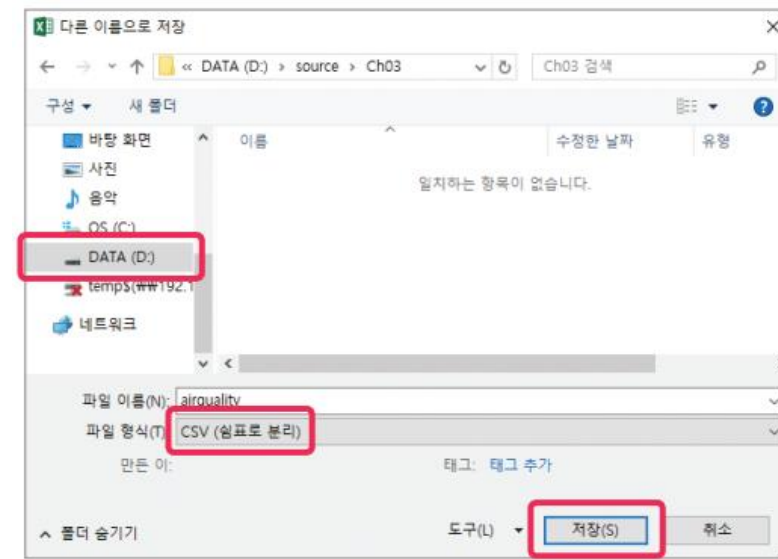


그림 3-6 CSV 형식으로 데이터셋 저장

파일 데이터 읽기

코드 3-19

```
setwd("D:/source")          # 작업 폴더 지정
air <- read.csv("airquality.csv", header=T)  # .csv 파일 읽기
head(air)
```

```
> setwd("D:/source")          # 작업 폴더 지정
> air <- read.csv("airquality.csv", header=T)  # .csv 파일 읽기
> head(air)
```

| | Ozone | Solar.R | Wind | Temp | Month | Day |
|---|-------|---------|------|------|-------|-----|
| 1 | 41 | 190 | 7.4 | 67 | 5 | 1 |
| 2 | 36 | 118 | 8.0 | 72 | 5 | 2 |
| 3 | 12 | 149 | 12.6 | 74 | 5 | 3 |
| 4 | 18 | 313 | 11.5 | 62 | 5 | 4 |
| 5 | NA | NA | 14.3 | 56 | 5 | 5 |
| 6 | 28 | NA | 14.9 | 66 | 5 | 6 |

파일 데이터 쓰기

코드 3-20

```
setwd("D:/source") # 작업 폴더 지정
my.iris <- subset(iris, Species='Setosa') # Setosa 품종 데이터만 추출
write.csv(my.iris, "my_iris.csv", row.names=F) # .csv 파일에 저장하기
```

```
> setwd("D:/source") # 작업 폴더 지정
> my.iris <- subset(iris, Species='Setosa') # Setosa 품종 데이터만 추출
> write.csv(my.iris, "my_iris.csv", row.names=F) # .csv 파일에 저장하기
```

- **my.iris**
저장할 데이터가 들어 있는 곳이 my.iris이다.
- **"my_iris.csv"**
저장할 파일의 이름을 지정한다.
- **row.names=F**
my.iris를 저장할 때 행 번호를 붙이지 말라는 의미이다.

Thank You
Any Questions?