



# 3주차: R의 데이터형과 연산

**ChulSoo Park**

School of Computer Engineering & Information Technology  
Korea National University of Transportation

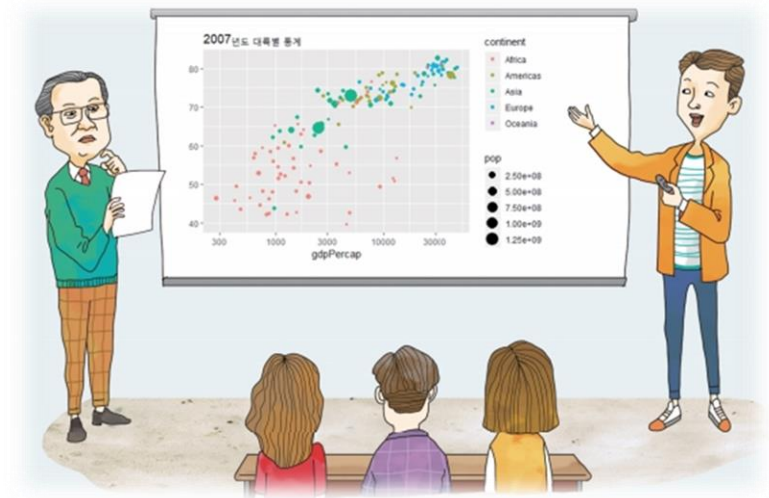
# 학습목표 (3주차)

- ❖ R에서 데이터의 처리와 저장
- ❖ 변수, 연산자, 벡터, 행렬, 리스트 등 Data 형태에 대한 이해
- ❖ R활용 문법 숙지

# 03

## CHAPTER

# R의 데이터형과 연산



## CONTENTS

- 3.1 데이터 저장과 처리
- 3.2 변수
- 3.3 데이터형
- 3.4 연산자
- 3.5 벡터
- 3.6 배열(행렬)
- 3.7 데이터 프레임
- 3.8 리스트
- 요약

- 데이터 과학의 문법에 대해 학습해 보자.
  - 변수: 데이터 저장 공간
  - 데이터형: 숫자형, 문자형, 범주형, 논리형, 특수 상수 등
  - 연산자: 산술, 비교, 논리 연산자
  - 벡터: 단일값들의 모임
  - 배열: 열과 행을 가지는 데이터 집합. 벡터의 요소들이 다시 벡터로 구성된 형태.
  - 데이터 프레임: 서로 다른 데이터 형이 표 형태로 정리된 구조. 각 속성의 크기가 같음.
  - 리스트: 데이터 프레임과 유사한 표 형태의 구조. 각 속성의 크기가 달라도 됨.

## 3.4 연산자

### ■ 연산자의 종류

#### ■ 산술연산자

연산자	설명	예
+	덧셈	
-	뺄셈	
*	곱셈	
/	나눗셈	
$\wedge$ 또는 **	지수승	
$x \% \% y$	x를y로 나눈 나머지 (정수 나눗셈 나머지)	$5 \% \% 2 \rightarrow 1$
$x \% / \% y$	X를 y로 나눈 몫 (정수 나눗셈 몫)	$5 \% / \% \rightarrow 2$

RGui (64-bit)

파일 편집 보기 기타 패키지들 윈도우즈 도움말

R Console

```

> x=2+5
> x
[1] 7
> y=5-2
> y
[1] 3
> x=2*5
> x
[1] 10
> y=5/2
> y
[1] 2.5
> x=5^2
> x
[1] 25
> y=5%%2
> y
[1] 1
> x=5%%2
> x
[1] 2
> y=2.5
> y <- as.integer(y)
> y
[1] 2
  
```

## 3.4 연산자

### ■ 연산자 우선순위

표 3-6 연산자 우선순위

연산자	설명	우선순위
$\wedge$ , **	지수승	높음 ↑
+, -	단항 플러스와 마이너스	
%any%	%%, %/% 등 연산자	
*, /	곱셈, 나눗셈	
+, -	덧셈, 뺄셈	낮음 ↓
==, !=, <, >, <=, >=	비교 연산자	
!	논리 부정(not)	
&, &&	논리 and	
,	논리 or	

RGui (64-bit)

파일 편집 보기 기타 패키지들 윈도우즈 도움말

R Console

```

> x=5+2^3/2
> x
[1] 9
> y=-5^2*3
> y
[1] 2
> y=-5^2
> y
[1] -25
> y=5^2*3
> y
[1] 1
> y=2+5*3^2
> y
[1] 7
> |
  
```

- 여러 단일 값을 하나의 변수 명으로 저장 가능
  - 지금까지 단일 값을 단일 변수에 저장하는 방법을 배웠다.
  - 단일 값을 하나의 변수로 저장하면 값이 많은 경우 변수의 수도 증가하게 됨.
  - 하나의 벡터 변수로 여러 단일값을 저장할 수 있음.

Console C:/RSources/ ↗

```
> x=c(1:3)
> y=c(3:6)
> z=c(3,1,2)
> union(x,y)    # 합집합
[1] 1 2 3 4 5 6
> intersect(x,y) #교집합
[1] 3
> setdiff(x,y)   # 차집합(x에서 y와 동일한 요소 제외)
[1] 1 2
> setdiff(y,x)   # 차집합(y에서 x와 동일한 요소 제외)
[1] 4 5 6
> setequal(x,y)
[1] FALSE
> setequal(z,x)  # x와 z가 동일한지 비교
[1] TRUE
```

## 3.6 배열(행렬)

- 배열: 열과 행으로 구성된 데이터
- 학생 100명의 DB, OS, Data Science 성적을 한꺼번에 저장하려면  
과목은 열(column)로, 학생 별 성적은 행(row)으로 100X3 배열로 생성

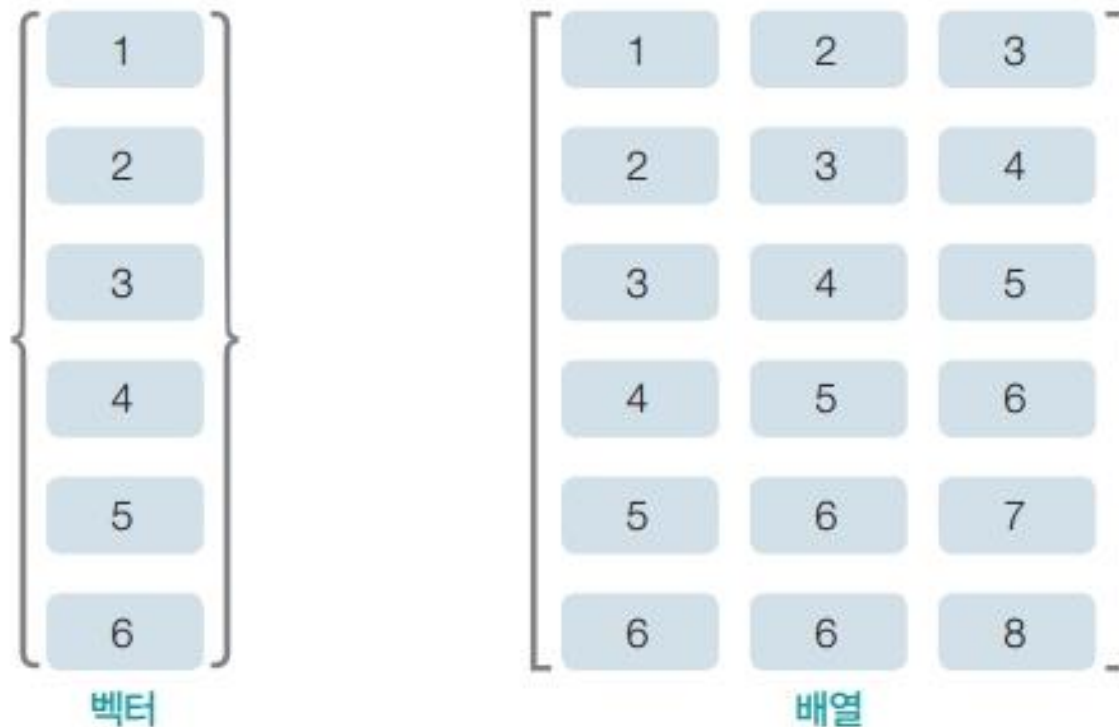


그림 3-4 벡터와 배열의 구성 형태



## 3.6 배열(행렬)

### ① 배열 생성 함수

- array 함수: N차원 배열 생성

벡터 Data

차원을 정의하는 벡터

Console C:/RSources/

```
> x=array(1:12,c(3,4))
```

```
> x
```

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

행렬로 구성 할 벡터 Data

행과 열 중 하나 결정

```
> y=1:20
```

```
> matrix(y,nrow=4,byrow=T)
```

	[,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

데이터를 행단위로 배치할지 여부(T/F)

```
> matrix(y,ncol=4,byrow=F)
```

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

## 3.7 데이터 프레임

### ■ 데이터 프레임

- 가장 흔히 쓰이는 표 형태의 데이터 구조를 가짐.
- 행렬과 달리 여러 데이터형을 혼합하여 저장할 수 있음.
- 리스트와 달리 행의 수를 일치시켜서 저장해야 함.

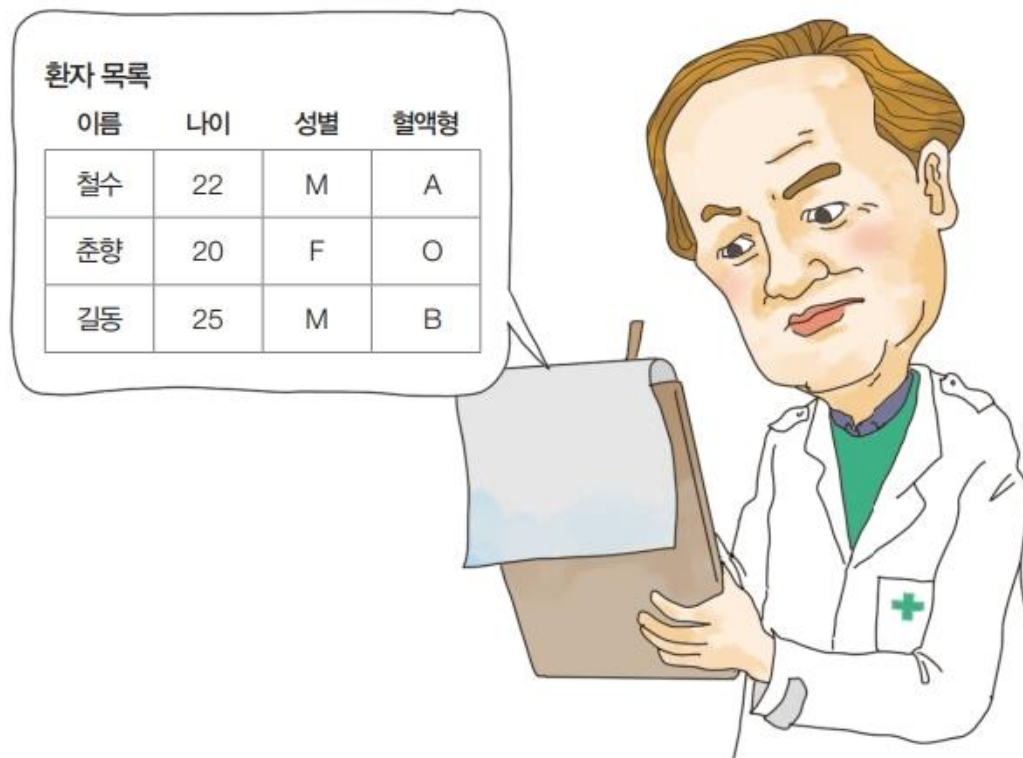


그림 3-5 데이터 프레임의 구성 형태

## 3.7 데이터 프레임

### ① 데이터 프레임 생성: data.frame 함수 이용

환자 데이터

name	age	gender	blood.type
철수	22	M	A
준향	20	F	O
길동	25	M	B

```

Console C:/RSources/
> name=c("철수","준향","길동")
> age=c(22,20,25)
> gender=factor(c("M","F","M"))
> blood.type=factor(c("A","O","B"))
> patients1=data.frame(name,age,gender,blood.type)
> patients1
  name age gender blood.type
1 철수  22      M          A
2 준향  20      F          O
3 길동  25      M          B
> patients2=data.frame(name=c("철수","준향","길동"),
+ age=c(22,20,25),gender=factor(c("M","F","M")),blood.type=factor(c("A","O","B")))
> patients2
  name age gender blood.type
1 철수  22      M          A
2 준향  20      F          O
3 길동  25      M          B
  
```

속성

요소

## 3.7 데이터 프레임

### ① 데이터 프레임 생성: data.frame 함수 이용

```
patients=data.frame()  
test=patients1  
patients=edit(test)
```

Console C:/RSources/ ➔

```
> patients1  
  name age gender blood.type  
1 철수  22     M         A  
2 춘향  20     F         O  
3 길동  25     M         B  
> patients=data.frame()  
> test=patients1  
> patients=edit(test)
```

데이터 편집기

파일 편집 도움말

	name	age	gender	blood.type	var5	var6	var7
1	철수	22	M	A			
2	춘향	20	F	O			
3	길동	25	M	B			
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							

변수 편집기

변수명 blood.type

유형 ☐ numeric ☒ character

## 3.7 데이터 프레임

### ① 데이터 프레임 생성: data.frame 함수 이용

```
patients=data.frame()
test=patients1
patients=edit(test)
```

Console C:/RSources/ ↗

```
> patients1
  name age gender blood.type
1 철수  22      M          A
2 춘향  20      F          O
3 길동  25      M          B

> patients
  name age1 gender blood.type
1 철수  22      M          A
2 춘향  20      F          O
3 길동1  35      F          A
```

데이터 편집기

파일 편집 도움말

	name	age	gender	blood.type	var5	var6	var7
1	철수	22	M	A			
2	춘향	20	F	O			
3	길동	25	M	B			
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							

데이터 편집기

파일 편집 도움말

	name	age	gender	blood.type	var5	var6	var7
1	철수	22	M	A			
2	춘향	20	F	O			
3	길동1	35	F	A			
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							





변수 편집기

변수명

유형 ☒ numeric ☐ character

## 3.7 데이터 프레임

### ② 데이터 프레임 요소에 접근(행과 열로 찾기와 \$사용)

```
Console C:/RSources/      
> patients$name # name 속성 값 출력  
[1] "철수" "준향" "길동"  
> patients[1,] # 1행의 값 출력  
name age gender blood.type  
1 철수 22 M A  
> patients[,2] # 2열의 값 출력  
[1] 22 20 25  
> patients[1,2] # 1행 2열의 값 출력  
[1] 22  
> patients[patients$name=="철수",] # 철수 환자 정보 출력  
name age gender blood.type  
1 철수 22 M A  
> patients[patients$name=="철수",c("name","age")]  
name age  
1 철수 22
```

## 3.7 데이터 프레임

### ③ 데이터 프레임에 유용한 함수

Attach, detach 함수 : 데이터 프레임의 속성명을 변수명으로 변경

```
Console C:/RSources/
> str(cars)
'data.frame':   50 obs. of  2 variables:
 $ speed: num  4 4 7 7 8 9 10 10 10 11 ...
 $ dist : num  2 10 4 22 16 10 18 26 34 17 ...
> speed
에러: 객체 'speed'를 찾을 수 없습니다
> attach(cars)
> speed
 [1]  4  4  7  7  8  9 10 10 10 11 11 12 12 12 12 13 13
[18] 13 13 14 14 14 14 15 15 15 16 16 17 17 17 18 18 18
[35] 18 19 19 19 20 20 20 20 20 22 23 24 24 24 24 25
> detach(cars)
> speed
에러: 객체 'speed'를 찾을 수 없습니다
```




## 3.7 데이터 프레임

### ③ 데이터 프레임에 유용한 함수

with함수 : 데이터 프레임에 다양한 함수 적용

(attach를 사용하지 않고 열 이름을 바로 사용 가능

attach는 지속적으로 사용할 시, with는 임시로 사용할 때 유용)





```
Console C:/RSources/     
> mean(speed)  
[1] 15.4  
> min(speed)  
[1] 4  
> with(cars, mean(speed))  
[1] 15.4  
> with(cars, min(speed))  
[1] 4  
> mean(cars$speed)  
[1] 15.4  
> min(cars$speed)  
[1] 4
```



## 3.7 데이터 프레임

### ③ 데이터 프레임에 유용한 함수

Subset 함수 : 데이터 프레임에서 일부 데이터만 추출

```
Console C:/Rsources/    
```

```
> subset(cars, speed>20)
  speed dist
44    22   66
45    23   54
46    24   70
47    24   92
48    24   93
49    24  120
50    25   85
```



```
> subset(cars, speed>20, select=c(dist))
  dist
44   66
45   54
46   70
47   92
48   93
49  120
50   85
```

```
> subset(cars, speed>20, select=-c(dist))
  speed
44    22
45    23
46    24
47    24
48    24
49    24
50    25
```

## 3.7 데이터 프레임

### ③ 데이터 프레임에 유용한 함수

na.omit 함수: 데이터 중 na를 제외하고 추출

```
Console C:/RSources/    
> str(airquality)  
'data.frame': 153 obs. of 6 variables:  
 $ Ozone : int 41 36 12 18 NA 28 23 19 8 NA ...  
 $ Solar.R: int 190 118 149 313 NA NA 299 99 19 194 ...  
 $ Wind : num 7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6  
 ...  
 $ Temp : int 67 72 74 62 56 66 65 59 61 69 ...  
 $ Month : int 5 5 5 5 5 5 5 5 5 5 ...  
 $ Day : int 1 2 3 4 5 6 7 8 9 10 ...  
> mean(Temp)  
[1] 77.88235  
> mean(Ozone)  
[1] NA  
> mean(na.omit(Ozone))  
[1] 42.12931
```

## 3.7 데이터 프레임

### ③ 데이터 프레임에 유용한 함수

merge 함수 : 여러 데이터 프레임 병합(> ?merge )

R: Merge Two Data Frames ▾

Find in Topic

Merge two data frames by common columns or row names, or do other versions of database *join* operations.

#### Usage

```
merge(x, y, ...)
```

```
## Default S3 method:
```

```
merge(x, y, ...)
```





```
## S3 method for class 'data.frame'
```

```
merge(x, y, by = intersect(names(x), names(y)),  
      by.x = by, by.y = by, all = FALSE, all.x = all,  
      sort = TRUE, suffixes = c(".x", ".y"), no.dups  
      incomparables = NULL, ...)
```

## 3.7 데이터 프레임

### ③ 데이터 프레임에 유용한 함수

merge 함수 : 여러 데이터 프레임 병합

```
Console C:/RSources/      
> name=c("철수","영희","길동")  
> age=c(22,20,25)  
> gender=factor(c("M","F","M"))  
> blood.type=factor(c("A","O","B"))  
> check=c("Y","N","N")  
> patients1=data.frame(name,age,gender,blood.type)  
> patients2=data.frame(name,check)  
> patients2  
  name check  
1 철수     Y  
2 영희     N  
3 길동     N  
> patients=merge(patients1,patients2,by="name")  
> patients  
  name age gender blood.type check  
1 길동  25     M         B      N  
2 영희  20     F         O      N  
3 철수  22     M         A      Y
```

## 3.7 데이터 프레임

### ③ 데이터 프레임에 유용한 함수


merge 함수 : 여러 데이터 프레임 병합

```
Console C:/RSources/
> name=c("철수","영희","길동")
> name1=c("철수","춘향","길동")
> age=c(22,20,25)
> gender=factor(c("M","F","M"))
> blood.type=factor(c("A","O","B"))
> check=c("Y","N","N")
> patients1=data.frame(name,age,gender,blood.type)
> patients2=data.frame(name1,check)
> patients=merge(patients1,patients2,by.x="name",by.y="name1")
> patients
  name age gender blood.type check
1 길동  25      M          B      N
2 철수  22      M          A      Y
> patients=merge(patients1,patients2,by.x="name",by.y="name1",all
=TRUE)
> patients
  name age gender blood.type check
1 길동  25      M          B      N
2 영희  20      F          O    <NA>
3 철수  22      M          A      Y
4 춘향  NA    <NA>        <NA>      N
```

## 3.7 데이터 프레임

### ③ 데이터 프레임에 유용한 함수

is.dataframe, as.data.frame 함수 : 데이터 프레임 형식 확인 및 변환

```
Console C:/RSources/   
> x=array(1:12, c(3,4))  
> str(x)  
int [1:3, 1:4] 1 2 3 4 5 6 7 8 9 10 ...  
> x  
      [,1] [,2] [,3] [,4]  
[1,]    1    4    7   10  
[2,]    2    5    8   11  
[3,]    3    6    9   12  
> is.data.frame(x)  
[1] FALSE  
> names(x)=c("국어", "수학", "영어", "sw")  
> x  
      [,1] [,2] [,3] [,4]  
[1,]    1    4    7   10  
[2,]    2    5    8   11  
[3,]    3    6    9   12  
attr(,"names")  
[1] "국어" "수학" "영어" "sw"  NA     NA     NA     NA     NA     NA     NA     NA
```

## 3.7 데이터 프레임

### ③ 데이터 프레임에 유용한 함수

is.dataframe, as.data.frame 함수 : 데이터 프레임 형식 확인 및 변환

```
Console C:/RSources/ ↗
> x=array(1:12, c(3,4))
> str(x)
int [1:3, 1:4] 1 2 3 4 5 6 7 8 9 10 ...
> x
      [,1] [,2] [,3] [,4]
[1,]     1     4     7    10
[2,]     2     5     8    11
[3,]     3     6     9    12
> is.data.frame(x)
[1] FALSE
> x=as.data.frame(x)
> is.data.frame(x)
[1] TRUE
> names(x)=c("국어", "수학", "영어", "sw")
> x
  국어 수학 영어 sw
1     1     4     7 10
2     2     5     8 11
3     3     6     9 12
```

## 3.8 리스트

### ■ 리스트

- 서로 다른 기본 데이터형을 갖는 자료 구조를 포함할 수 있음.
- 데이터 프레임보다 넓은 의미의 데이터 모임.
- 데이터 프레임과 달리 모든 속성의 크기가 같을 필요가 없음.

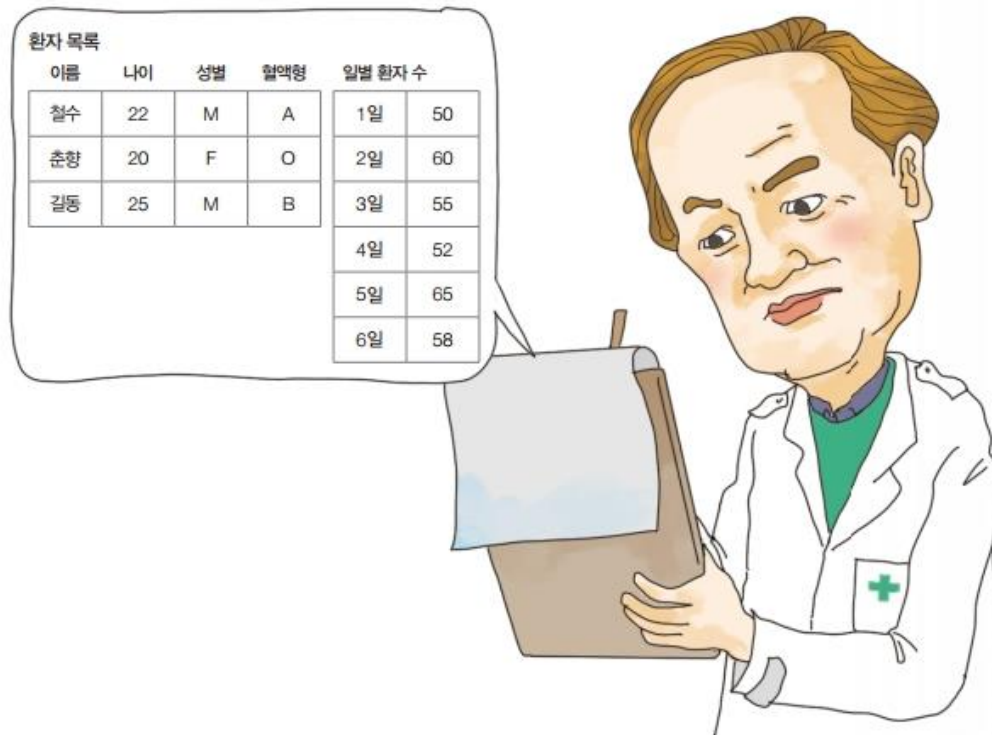


그림 3-6 리스트의 구성 형태



## 3.8 리스트

### ① 리스트 생성 및 데이터에 이름 부여

Console C:/RSources/ ↗

```
> name=c("철수","춘향","길동")
> age=c(22,20,25)
> gender=factor(c("M","F","M"))
> blood.type=factor(c("A","O","B"))
> patients=data.frame(name,age,gender,blood.type)
> patients
  name age gender blood.type
1 철수  22      M          A
2 춘향  20      F          O
3 길동  25      M          B
>
> no.patients=data.frame(day=c(1:6),no=c(50,60,70,33,44,55))
```

```
> # data 단순 추가
> listPatients=list(patients,no.patients)
> listPatients
```

```
[[1]]
  name age gender blood.type
1 철수  22      M          A
2 춘향  20      F          O
3 길동  25      M          B
```

```
[[2]]
  day no
1  1 50
2  2 60
3  3 70
4  4 33
5  5 44
6  6 55
```

Console C:/RSources/ ↗

```
> # 각 list data 이름 부여 추가
> listPatients=list(patients=patients,no.patients=no.patients)
> listPatients
```

```
$patients
  name age gender blood.type
1 철수  22      M          A
2 춘향  20      F          O
3 길동  25      M          B
```

```
$no.patients
  day no
1  1 50
2  2 60
3  3 70
4  4 33
5  5 44
6  6 55
```

## 3.8 리스트

### ② 리스트 생성 [리스트 요소에 접근 방법]

Console C:/RSources/ ↗

```
> listPatients$patients # 요소명 입력
```

	name	age	gender	blood.type
1	철수	22	M	A
2	춘향	20	F	O
3	길동	25	M	B

```
> listPatients[2] # 인덱스 입력
```

```
$no.patients
```




	day	no
1	1	50
2	2	60
3	3	70
4	4	33
5	5	44
6	6	55

```
> listPatients[[2]] # 인덱스 입력
```

	day	no
1	1	50
2	2	60
3	3	70
4	4	33
5	5	44
6	6	55

## 3.8 리스트

③ 리스트에 유용한 함수 : `lapply(list+apply)`, `sapply(simple+apply)` 함수

```
Console C:/RSources/     
> lapply(ListPatients$no.patients, mean)  
$day  
[1] 3.5  
  
$no  
[1] 53.5  
  
> lapply(ListPatients$patients1, mean)  
$name  
[1] NA  
  
$age  
[1] 22.33333  
  
$gender  
[1] NA  
  
$blood.type  
[1] NA
```

경고메시지(들):

```
1: In mean.default(X[[i]], ...) :  
   인자가 수치형 또는 논리형이 아니므로 NA를 반환합니다  
2: In mean.default(X[[i]], ...) :  
   인자가 수치형 또는 논리형이 아니므로 NA를 반환합니다
```

# 연습문제 (p.109) 1번)

# 연습 문제1

```
room=data.frame(room=c(30))
```

```
listPatients=list(patients=patients,no.patients=no.patients,room=room)
```

```
listPatients
```

Console C:/RSources/ ↗

```
> # 연습 문제1
> room=data.frame(room=c(30))
> listPatients=list(patients=patients,no.patients=no.patients,room=room)
> listPatients
```

```
$patients
  name age gender blood.type
1 철수  22      M          A
2 춘향  20      F          O
3 길동  25      M          B
```

```
$no.patients
  day no
1   1 50
2   2 60
3   3 70
4   4 33
5   5 44
6   6 55
```

```
$room
  room
1   30
```

Console C:/RSources/ ↗

```
> listPatients$room=NULL
> listPatients
```

```
$patients
  name age gender blood.type
1 철수  22      M          A
2 춘향  20      F          O
3 길동  25      M          B
```

```
$no.patients
  day no
1   1 50
2   2 60
3   3 70
4   4 33
5   5 44
6   6 55
```

## 요약

1. R 데이터 읽기
2. R 데이터 쓰기
3. 변수, 연산자, 벡터, 행렬, 리스트 등 데이터 형태 이해
4. R 사용 방법 숙지

# Thank you

