제6주 : 데이터프레임

6.1. 데이터 프레임(data.frame)

- 테이블 형태의 자료 구조이다. 테이블의 각 열에 해당하는 위치에 길이가 고정된 벡터(또는 팩터) 형태의 자료가 배치된다. 즉, 길이가 동일한 벡터(또는 팩터)들의 모음이라고 할 수 있다.
- 구성: 각 열 벡터는 서로 다른 기본 자료형을 가질 수 있다. 하지만 하나의 열벡터를 구성하는 원소들은 동일한 기본 자료형을 가져야 한다.
- 거의 모든 정형화된 통계자료는 데이터 프레임 형태를 가지므로 많이 사용되는 자료구조입니다. 설명을 위해 데이터프레임을 직접 입력하여 생성하는 방법을 설명하겠습니다. 하지만 데이터프레임은 직접 입력하는 것 보다는 외부 데이터로부터 가져오는 경우가 많습니다.

- 생성방법

```
## 1: 테이블의 각 열에 해당하는 벡터를 직접 입력하는 방법
 x <- data.frame(벡터1, 벡터2, ...)

## 2. 테이블의 각 열에 해당하는 벡터와 그 이름을 입력하는 방법
 x <- data.frame(이름1=벡터1, 이름2=벡터2, ...)
```

- 사용예

```
> name <- c("철수", "영희", "길동")
> age <- c(21, 20, 31)
> gender <- factor(c("M", "F", "M"))
> character <- data.frame(name, age, gender)
➔ data frame을 구성할 속성은 name, age, gender
```

```
> character
  name age gender
1 철수   21      M
2 영희   20      F
3 길동   31      M
```
➔ 열 이름이 name, age, gender인 데이터프레임이 생성되었다. 반면에 리스트의 경우, list(name, age, gender)는 각 원소의 이름이 설정되지 않은 리스트를 생성한다.

```
> character$name
[1] 철수 영희 길동
Levels: 길동 영희 철수
```
➔name 속성의 값 가져오기

```
> character[1, ]
 name age gender
1 철수  21      M
```
➔첫 번째 행에 해당하는 값 가져오기(세 속성 모두 포함)

```
> character[, 2]
[1] 21 20 31
```
➔두 번째 열에 해당하는 값 가져오기 (모든 행의 두 번째 열의 값을 가져옴. 벡터)

```
> character[3, 1]
[1] 길동
Levels: 길동 영희 철수
```
➔세 번째 행의 첫 번째 열의 값 가져오기(벡터)

- 데이터 프레임 내의 자료값에 접근하는 방법:
  ① 행렬 내의 자료값에 접근하는 방법과 동일하게 자료값에 접근할 수 있다:
  예: x[3, 1] x[1,] x[, "age"]

② 각 열에 이름이 붙어있는 경우, 리스트 내의 자료값에 접근하는 방법과 동일하게 자료값에 접근할 수 있다:

예: x$name x$name[3] x$age

- 유용한 함수들(1):
  - 행렬에서 사용가능한 함수들 colnames(), rownames(), nrow(), ncol(), dim(), dimnames() 등을 사용할 수 있다.

    예: colnames(cars)
    dim(cars)
    dimnames(cars)
  - 리스트에서 사용가능한 함수들 length(), names() 등을 사용할 수 있다.

    예: length(cars)
    names(cars)

- 유용한 함수(2) with()

with(dat, expr, ...): 데이터 프레임 dat에 대해 명령문 expr을 수행합니다. 명령문 내에서 dat의 열 이름을 직접 사용할 수 있습니다.

- dat: 데이터 프레임 혹은 리스트
- expr: 수행할 명령문

사용예:

```
> head(cars)
 speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8  16
6     9   10
> mean(cars$speed)
[1] 15.4
```

```
> mean(cars$dist)
[1] 43
> with(cars, mean(speed))
[1] 15.4
> with(cars, mean(dist))
[1] 43
```

➔ 명령문 내에서 열 이름을 바로 사용할 수 있습니다. 열 이름을 지속적으로 사용하는 경우에는 attach()라는 함수를 이용하지만, 그렇지 않은 경우 with()를 추천합니다.

- 유용한 함수(3): subset()

subset(dat,
subset,
select, drop = FALSE, ...)

: 데이터 프레임 dat으로부터 조건에 맞는 부분집합을 추출합니다. 명령문 내에서 dat의 열 이름을 직접 사용할 수 있습니다.

- dat: 데이터 프레임 등 부분집합을 추출할 R 객체
- subset: 행의 부분집합을 선택할 조건.
  (예: 조건문을 이용할 경우, TRUE와 FALSE로 된 벡터 형태이다. 결측치가 있을 경우 FALSE로 간주한다.)
- select: 열의 부분집합을 선택할 조건.
- drop: "["로 전달될 슬라이싱 조건. 기본값은 FALSE

사용예:

```
> airquality
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
```

……

```
> subset(airquality, Temp > 80)
Ozone Solar.RWind Temp Month Day
29    45     252 14.9   81     5  29
35    NA     186  9.2   84     6   4
36    NA     220  8.6   85     6   5
```

……

➔airquality에서 Temp가 80보다 큰 자료를 가져옵니다.

➔비교: airquality[airquality$Temp > 80,]

```
> subset(airquality, Temp > 80, select = c(Ozone,
Temp))
   Ozone Temp
29    45   81
35    NA   84
36    NA   85
```

……

➔airquality에서 Temp가 80보다 큰 자료 중 Ozone 열과 Temp 열을 가져옵니다.

➔비교: airquality[airquality$Temp > 80, c("Ozone","Temp")]

```
> subset(airquality, Temp > 80, select = -c(Ozone,
Temp))
  Solar.R Wind Month Day
29     252 14.9     5  29
35     186  9.2     6   4
36     220  8.6     6   5
```

……

➔airquality에서 Temp가 80보다 큰 자료 중 Ozone 열과 Temp 열을 제외하고 가져옵니다.

- 유용한 함수(4): na.omit()

na.omit(dat)
데이터프레임 dat에서 어떤 값이 NA일 경우에 해당하는 행의 자료들을 제외하고 가져오는 함수입니다.

사용예:

```
> 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 ...
```

➔airqulity 자료는 153개의 관찰자료(행)를 갖는 자료구조입니다.

```
> str(na.omit(airquality))
'data.frame':   111 obs. of  6 variables:
$ Ozone  : int  41 36 12 18 23 19 8 16 11 14 ...
$ Solar.R: int  190 118 149 313 299 99 19 256 290 274
...
$ Wind   : num  7.4 8 12.6 11.5 8.6 13.8 20.1 9.7 9.2
10.9 ...
$ Temp   : int  67 72 74 62 65 59 61 69 66 68 ...
$ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
$ Day    : int  1 2 3 4 7 8 9 12 13 14 ...
-attr(*, "na.action")=Class 'omit'  Named int [1:42]
56 10 11 25 26 27 32 33 34 ...
.. ..-attr(*, "names")= chr [1:42] "5" "6" "10" "11"
...
```

➔na.omit을 수행하고 나면 NA를 포함하는 행을 제외하고 가져옵니다. na.action 속성(attribute)에 제외된 행번호를 기록합니다.

```
> newair <- na.omit(airquality)
> attr(newair, "na.action")
  5   6  10  11  25  26  27  32  33  34  35  36  37  39  42
  5   6  10  11  25  26  27  32  33  34  35  36  37  39  42
 43  45  46  52  53  54  55  56  57  58  59  60  61  65  72
 43  45  46  52  53  54  55  56  57  58  59  60  61  65  72
 75  83  84  96  97  98 102 103 107 115 119 150
 75  83  84  96  97  98 102 103 107 115 119 150
attr(,"class")
[1] "omit"
```

- 유용한 함수(5): merge(x,y)

```
merge(x, y,
      by = intersect(names(x), names(y)),
     by.x = by, by.y = by,
     all = FALSE, all.x = all, all.y = all,
     sort = TRUE,
     suffixes = c(".x",".y"),
     ...)
```

- x, y : 합칠 대상이 되는 데이터 프레임
- by : 합칠 때 기준이 되는 열로서, 기본값은 두 데이터 프레임 중 동일한 이름을 갖는 열
- by.x : 합칠 기준이 되는 x의 열 이름
- by.y : 합칠 기준이 되는 y의 열 이름
- all.x : TRUE로 지정시 모든 x의 행이 합쳐지고 이에 해당하는 y가 없을 경우 y 열에 해당하는 값은 NA
- all.y : TRUE로 지정시 모든 y의 행이 합쳐지고 이에 해당하는 x가 없을 경우 x 열에 해당하는 값은 NA
- all : all.x 와 all.y 가 동일한 TRUE 나 FALSE를 갖게 함
- sort : TRUE이면 합쳐질 열 이름 순으로 정렬
- suffixes : X와 Y의 이름이 서로 동일할 경우 두 개의 문자로 구성된 suffix 추가

사용예:

```
> ## authors 자료 만들기
> surname <- I(c("Tukey", "Venables", "Tierney", "Ripley",
"McNeil"))
>  nationality  <-  c("US",  "Australia",  "US",  "UK",
"Australia")
> deceased <- c("yes", rep("no", 4))
> authors <- data.frame(surname, nationality, deceased)
> authors
> ## books 자료 만들기
> name <- I(c("Tukey", "Venables", "Tierney",
            "Ripley", "Ripley", "McNeil", "R Core"))
> title <- c("Exploratory Data Analysis",
             "Modern Applied Statistics ...",
             "LISP-STAT",
             "Spatial Statistics",
             "Stochastic Simulation",
             "Interactive Data Analysis",
             "An Introduction to R")
> other.author <- c(NA, "Ripley", NA, NA, NA, NA,
                    "Venables & Smith")
> books <- data.frame(name, title, other.author)
> books
> ## author 자료와 books자료를 합치기
> ## 공통된 변수 surname변수와 name변수를 기준으로 합치기:
    전달인자 by.x = "surname"  및  by.y = "name"을 지정
> (m1 <- merge(authors, books, by.x = "surname", by.y =
"name"))
   surname  nationality  deceased                      title   other.author
1   McNeil    Australia       no      Interactive Data Analysis         <NA>
2   Ripley           UK       no            Spatial Statistics          <NA>
3   Ripley           UK       no         Stochastic Simulation          <NA>
4  Tierney           US       no                     LISP-STAT          <NA>
5    Tukey           US      yes      Exploratory Data Analysis          <NA>
6 Venables    Australia       no   Modern Applied Statistics ...       Ripley
```

➔authors와 books를 authors의 surname 값과 books의 name 값이 같은 자료를 한행으로 하여합친다.

```
> (m2 <- merge(books, authors, by.x = "name", by.y = "surname"))
     name                          title other.author  nationality deceased
1   McNeil     Interactive Data Analysis         <NA>    Australia       no
2   Ripley           Spatial Statistics         <NA>           UK       no
3   Ripley        Stochastic Simulation         <NA>           UK       no
4  Tierney                    LISP-STAT         <NA>           US       no
5    Tukey     Exploratory Data Analysis         <NA>           US      yes
6 Venables Modern Applied Statistics ...      Ripley   Australia       no
```

➔authors와 books를 authors의 name 값과 books의 surname 값이 같은 자료를 한 행으로 하여 합친다.

➔ 위 두 경우 모두 x와 y 자료 둘에 공통 값이 있는 행만 합쳐지고 그렇지 않은 행은 포함되지 않았다. all = FALSE가 기본값이므로.
➔ all = TRUE로 고쳐서 다시 명령을 수행하고 결과를 확인하시오.

➔ 위 두 경우 모두 x와 y 자료가 합쳐진 후 열의 값들 순서대로 정렬한다. sort = TRUE가 기본값이므로.
➔ sort = FALSE로 놓고 다시 명령을 수행하고 결과를 확인하시오.