제2강

변수와 벡터

Section 01

R의 기본 연산

Section 02

변수

Section 03

벡터의 이해

Section 04

벡터의 연산

Section 05

리스트와 팩터

5. 리스트와 팩터

1. 리스트

- 서로 다른 자료형의 값들을 1차원 배열에 저장하고 다룰 수 있도록 해주는 수단

코드 2-21

```
ds <- c(90, 85, 70, 84)
my.info <- list(name='Tom', age=60, status=TRUE, score=ds)
my.info                 # 리스트에 저장된 내용을 모두 출력
my.info[[1]]            # 리스트의 첫 번째 값을 출력
my.info$name            # 리스트에서 값의 이름이 name인 값을 출력
my.info[[4]]            # 리스트의 네 번째 값을 출력
```

```
> ds <- c(90, 85, 70, 84)
> my.info <- list(name='Tom', age=60, status=TRUE, score=ds)
> my.info                    # 리스트에 저장된 내용을 모두 출력
$name
[1] "Tom"


$age
[1] 60
```

5. 리스트와 팩터

```
$status
[1] TRUE


$score
[1] 90 85 70 84
> my.info[[1]]                # 리스트의 첫 번째 값을 출력
[1] "Tom"
> my.info$name                # 리스트에서 값의 이름이 name인 값을 출력
[1] "Tom"
> my.info[[4]]                # 리스트의 네 번째 값을 출력
[1] 90 85 70 84
```

리스트의 내용을 출력해보면, 벡터와 달리 세로로 하나씩 출력되는 것을 볼 수가 있다.
리스트에 저장된 값을 추출하는 방법은 벡터와 비슷하다. 인덱스를 이용할 수 있는데 벡터와 차이점은 인덱스 지정 부분이 [ ]가 아닌 [[ ]]을 사용한다는 것이다.
아울러, '리스트이름$값의이름 ' 형태로 값을 지정하면 된다.

하지만, 리스트는 벡터보다 조작의 자유도가 떨어진다. 예를 들면, 산술연산을 할 때 불편하다. 하여 unlist(리스트명)함수를 이용하여 벡터로 변환하여 사용하면 편리하다.

5. 리스트와 팩터

2. 팩터

- 문자형 데이터가 저장된 벡터의 일종
- 예를 들어 성별, 혈액형, 선호 정당 등과 같이 저장할 문자값들이 몇 종류로 정해져 있을 때 팩터를 사용

코드 2-22

```
bt <- c('A', 'B', 'B', 'O', 'AB', 'A')        # 문자형 벡터 bt 정의
bt.new <- factor(bt)                          # 팩터 bt.new 정의
bt                                            # 벡터 bt의 내용 출력
bt.new                                        # 팩터 bt.new의 내용 출력
bt[5]                                         # 벡터 bt의 5번째 값 출력
bt.new[5]                                     # 팩터 bt.new의 5번째 값 출력
levels(bt.new)                                # 팩터에 저장된 값의 종류를 출력
as.integer(bt.new)                  # 팩터의 문자값을 숫자로 바꾸어 출력
bt.new[7] <- 'B'                              # 팩터 bt.new의 7번째에 'B' 저장
bt.new[8] <- 'C'                              # 팩터 bt.new의 8번째에 'C' 저장
bt.new                                        # 팩터 bt.new의 내용 출력
```

5. 리스트와 팩터

```
> bt <- c('A', 'B', 'B', 'O', 'AB', 'A')      # 문자형 벡터 bt 정의
> bt.new <- factor(bt)                        # 팩터 bt.new 정의
> bt                                          # 벡터 bt의 내용 출력
[1] "A"  "B"  "B"  "O"  "AB" "A"
> bt.new                                      # 팩터 bt.new의 내용 출력
[1] A  B  B  O  AB A
Levels: A AB B O
> bt[5]                                       # 벡터 bt의 5번째 값 출력
[1] "AB"
> bt.new[5]                                   # 팩터 bt.new의 5번째 값 출력
[1] AB
Levels: A AB B O
> levels(bt.new)                              # 팩터에 저장된 값의 종류를 출력
[1] "A"  "AB" "B"  "O"
> as.integer(bt.new)                          # 팩터의 문자값을 숫자로 바꾸어 출력
[1] 1 3 3 4 2 1
> bt.new[7] <- 'B'                            # 팩터 bt.new의 7번째에 'B' 저장
> bt.new[8] <- 'C'                            # 팩터 bt.new의 8번째에 'C' 저장
```

벡터와 팩터의 차이점

1. 출력시 벡터는 " "가 붙지만, 팩터는 붙지 않는다.
2. 팩터는 Levels가 함께 출력되는데 이것은 팩터에 저장된 값의 종류를 알려주는 것이다. 팩터의 목적이 어떤 종류를 나타내는 문자값을 저장하는 것이기 때문에 종류의 정보를 나타내는 것을 잊지말자.

5. 리스트와 팩터

```
Warning message:
In `[<-.factor`(`*tmp*`, 8, value = "C") :
  invalid factor level, NA generated
> bt.new                                  # 팩터 bt.new의 내용 출력
[1] A    B    B    O    AB   A    B    <NA>
Levels: A AB B O
```

팩터는 이미 지정된 값의 종류 외에 다른 값이 들어오는 것을 막는다.
하여, 위의 코드에서 C라는 값은 팩터에 없기 때문에 경고메시지가 뜨고 아울러 출력해보면 NA가 뜬다.
(NA는 Not Available의 약자이다.)

리스트와 팩터는 벡터에 비해 사용 빈도가 적긴 하지만 중요한 자료구조이니 명확히 이해하도록 하자.

감사합니다.