Part 1. R 프로그래밍 (데이터 분석 전문가 양성과정)

04

벡터의 이해

경북대학교 배준현 교수
(joonion@knu.ac.kr)

04. 벡터의 이해

- R의 데이터 오브젝트:
  - 벡터: vector()
  - 팩터: factor()
  - 리스트: list()
  - 행렬: matrix()
  - 데이터 프레임: data.frame()

04. 벡터의 이해

- **벡터: *vector***
  - R에서 가장 기초적인 데이터 구조
  - R에서 모든 데이터는 벡터로 구성되어 있다.
  - **벡터의 생성 함수**: vector(), :, c(), seq(), rep()

04. 벡터의 이해

```
> v1 <- vector(length=2)
> v1
[1] FALSE FALSE

> v2 <- 1:10
> v2
 [1]  1  2  3  4  5  6  7  8  9 10

> v3 <- c(1, 2, 3, 5, 7)
> v3
[1] 1 2 3 5 7

> v4 <- seq(from=1, to=10, by=2)
> v4
[1] 1 3 5 7 9

> v5 <- rep(1:3, each=2, times=2)
> v5
 [1] 1 1 2 2 3 3 1 1 2 2 3 3
```

04. 벡터의 이해

- c() 함수: combine 함수는 여러 개의 벡터를 결합하여 하나의 벡터를 생성

```
> v1 <- c(1, 2, 3:5)
> v1
[1] 1 2 3 4 5

> v2 <- c(5:6, seq(7, 9, 2))
> v2
[1] 5 6 7 9

> v3 <- c(v1, v2)
> v3
[1] 1 2 3 4 5 5 6 7 9
```

- **벡터 원소의 자료형:** *type of elements*
  - 벡터의 원소: 논리형, 숫자형, 문자형
  - 벡터의 원소는 반드시 동일한 기본 자료형을 가진다.
    - 데이터 유형이 다르면 자동 변환: 논리형 < 숫자형 < 문자형

04. 벡터의 이해

```
> v1 <- c(T, T, F, F, T)
> v1
[1]  TRUE  TRUE FALSE FALSE  TRUE

> v2 <- c(T, F, 3, 3.14)
> v2
[1] 1.00 0.00 3.00 3.14

> v3 <- c(3, 3.14, "PI=3.14")
> v3
[1] "3"       "3.14"    "PI=3.14"

> v4 <- c(T, F, 3, "3.14")
> v4
[1] "TRUE"  "FALSE" "3"     "3.14"
```

- **벡터의 인덱싱:** *vector indexing*
  - 대괄호 [ ] 안에 원소 위치를 지정하여 원소를 선택
  - 원소의 위치도 벡터로 지정할 수 있음
  - 조건문으로 벡터를 필터링할 수 있음

```
> v <- c(10, 20, 30, 40, 50, 60, 70)

> v[1]
[1] 10
> v[7]
[1] 70

> v[1:3]
[1] 10 20 30
> v[c(1, 3, 5, 7)]
[1] 10 30 50 70

> v[c(T, T, F, F, F, F, T)]
[1] 10 20 70
```

```
> v <- c(10, 20, 30, 40, 50, 60, 70)

> v < 30
[1]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
> v[v < 30]
[1] 10 20

> v < 30 | v > 50
[1]  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE
> v[v < 30 | v > 50]
[1] 10 20 60 70

> v %% 3 == 0 & v %% 4 == 0
[1] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
> v[v %% 3 == 0 & v %% 4 == 0]
[1] 60
```

- 인덱스에 음수 벡터를 지정하여 제외(filtering)할 수도 있음

```
> v <- c(10, 20, 30, 40, 50, 60, 70)
> v[-1]
[1] 20 30 40 50 60 70

> v[-(1:3)]
[1] 40 50 60 70

> v[-c(1, 3, 5, 7)]
[1] 20 40 60

> v[!(v %% 20 == 0)]
[1] 10 30 50 70
```

- 인덱스로 추출한 벡터의 원소값에 다른 값을 할당할 수 있음

```
> v <- c(10, 20, 30, 40, 50, 60, 70)
> v[1] <- 11
> v
[1] 11 20 30 40 50 60 70

> v[2:3] <- c(22, 33)
> v
[1] 11 22 33 40 50 60 70

> v[v >= 50] <- v[v < 30]
> v
[1] 11 22 33 40 11 22 11

> v[8] <- 80
> v
[1] 11 22 33 40 11 22 11 80
```

- **피자나라치킨공주 벡터의 생성**

```
> v <- c()
> for (i in 1:15) {
+     if (i %% 3 == 0 && i %% 5 == 0) {
+         v <- c(v, 'PZ')
+     } else if (i %% 3 == 0) {
+         v <- c(v, 'P')
+     } else if (i %% 5 == 0) {
+         v <- c(v, 'C')
+     } else {
+         v <- c(v, 'D')
+     }
> }
> v
[1] "D"  "D"  "P"  "D"  "C"  "P"  "D"  "D"  "P"  "C"  "D"  "P"  "D"  "D"  "PZ“
> which(v == 'P')
> which(v == 'C')
> which(v == 'PC')
```

- **벡터의 연산**: *vectorized arithmetic*
  - 벡터와 벡터간의 연산은 원소 단위로 처리
  - 재사용 규칙: *recycling rule*
    - 벡터의 길이가 다르면 짧은 벡터를 반복(rep())하여 길이를 맞춤

```
> v1 <- c(1, 2, 3, 4, 5)
> v2 <- c(1, 2)

> v3 <- v1 + v2
Warning message:
In v1 * v2 :
  longer object length is not a multiple of shorter object length
> v3
[1] 2 4 4 6 6

> v4 <- 2 * v1
> v4
[1]  2  4  6  8 10

> v5 <- v1 * v2
> v5
[1] 1 4 3 8 5
```

- **논리값은 숫자로 취급할 수 있음**: TRUE는 1, FALSE는 0

```
> v1 <- c(T, T, F, F, T)
> sum(v1)
[1] 3

> v2 <- v1 + 2
> v2
[1] 3 3 2 2 3

> v3 <- 1:5 + c(T, F)
> v3
[1] 2 2 4 4 6

> sum(1:5 > c(2, 4))
[1] 2
```

- **팩터:** *factor*
  - R에서 범주형 데이터를 처리하기 위한 데이터 오브젝트
  - 레벨(level): 범주형 변수가 가질 수 있는 범주값
    - 팩터는 레벨로 지정된 범주값만 가질 수 있는 벡터

```
> sex <- c('M', 'F', 'M', 'F', 'F')
> sex
[1] "M" "F" "M" "F" "F“

> f.sex <- factor(sex)
> f.sex
[1] M F M F F
Levels: F M

> str(f.sex)
Factor w/ 2 levels "F","M": 2 1 2 1 1
> levels(f.sex)
[1] "F" "M"
> table(f.sex)
f.sex
F M
3 2
```

```
> blood <- c(1, 2, 3, 1, 4, 3, 2, 4)
> blood
[1] 1 2 3 1 4 3 2 4

> f.blood <- factor(blood,
+                   levels = c(1, 2, 3, 4),
+                   labels = c('A', 'B', 'AB', 'O'))
> f.blood
[1] A  B  AB A  O  AB B  O
Levels: A B AB O

> levels(f.blood)
[1] "A"  "B"  "AB" "O"
> table(f.blood)
f.blood
 A  B AB  O
 2  2  2  2
```

- 리스트: *list*
  - 벡터의 벡터: 서로 다른 유형의 원소를 가질 수 있는 벡터
  - 리스트의 인덱싱:
    - $ 기호를 이용해서 원소의 이름으로 인덱스 지정
    - 또는, 두 개의 대괄호 [[ ]]내에 인덱스 지정

```
> v1 <- 1:7
> v2 <- c('홍길동', '전우치', '주니온', '아사달', '아사녀', '연오랑', '세오녀')
> v3 <- factor(c('M', 'M', 'M', 'M', 'F', 'M', 'F'))
> lst <- list(no = v1, name = v2, sex = v3)
> str(lst)
List of 3
 $ no  : int [1:7] 1 2 3 4 5 6 7
 $ name: chr [1:7] "홍길동" "전우치" "주니온" "아사달" ...
 $ sex : Factor w/ 2 levels "F","M": 2 2 2 2 1 2 1
> lst
$no
[1] 1 2 3 4 5 6 7

$name
[1] "홍길동" "전우치" "주니온" "아사달" "아사녀" "연오랑" "세오녀"

$sex
[1] M M M M F M F
Levels: F M
```

```
> names(lst)
[1] "no"   "name" "sex"

> lst$no
[1] 1 2 3 4 5 6 7
> sum(lst$no)
[1] 28

> lst$name
[1] "홍길동" "전우치" "주니온" "아사달" "아사녀" "연오랑" "세오녀"
> lst$name[1:3]
[1] "홍길동" "전우치" "주니온"

> lst$sex
[1] M M M M F M F
Levels: F M
> table(lst$sex)
F M
2 5
```

- **행렬**: *matrix*
  - 행(row)과 열(col)의 2차원을 가진 벡터
    - 2차원 벡터이기 때문에 모든 원소는 동일한 원시 자료형을 가져야 함
  - 행렬의 인덱싱:
    - 대괄호 [] 안에 콤마로 구분하여 행과 열의 위치 지정: [*row, col*]

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

> dim(m)
[1] 4 3
> nrow(m)
[1] 4
> ncol(m)
[1] 3
```

```
> m[2, 3]
[1] 6
> m[3, 2]
[1] 8

> m[2:3, 1:2]
     [,1] [,2]
[1,]    4    5
[2,]    7    8
> m[1:2, ]
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> m[, 1:2]
     [,1] [,2]
[1,]    1    2
[2,]    4    5
[3,]    7    8
[4,]   10   11
```

- rbind(), cbind() 함수로 행렬을 합칠 수 있음

```
> rbind(m[1, ], m[4, ])
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]   10   11   12

> cbind(m[, 1], m[, 3])
     [,1] [,2]
[1,]    1    3
[2,]    4    6
[3,]    7    9
[4,]   10   12
```

- **연습문제 4.1:**
  - 피자나라치킨공주 벡터를 생성하는 코드를 참조하여
    - 1에서 15까지 각 인덱스 숫자의 약수의 개수 벡터 div를 생성하시오.
    - div: 1 2 2 3 2 4 2 4 3 4 2 6 2 4 4
  - div 벡터에서 벡터의 인덱싱과 벡터 연산을 이용하여 물음에 답하시오.
    - 약수의 개수가 2인 원소의 개수는 몇 개인가?
      - 반드시 sum() 함수를 이용할 것
    - 약수의 개수가 2인 원소의 인덱스를 모두 출력하시오.
      - 반드시 which() 함수를 이용할 것
  - 1에서 15까지 소수의 개수는 몇 개인가?

- **연습문제 4.2:**
  - **다섯 사람의 키와 몸무게를 조사한 표가 아래와 같이 제시되었다.**
    - **키, 몸무게의 벡터** height, weight**를 생성하시오.**
    - **혈액형의 팩터** blood**를 생성하시오.**
    - height, weight, blood**를 각각 원소의 이름으로 가진 리스트** lst**를 생성하시오.**
    - lst$height**와** lst$weight**의 평균을 계산하시오.**
    - lst$blood**의 빈도표를 출력하시오.**

변량	A	B	C	D	E
키	163	175	182	178	161
몸무게	65	87	74	63	51
혈액형	A	B	AB	O	A

Any Questions?