1. 벡터: 객체(자료의 저장소)중의 하나 ~ 변수와 비슷하다고 이해

2. 벡터생성: 자료type

3. 속성(attributes):

객체의 정보를 나타내는 메타데이터(metadata)(다른 데이터를 설명해 주는 데이터)

4. 벡터의 구성요소 선택(서브셋팅):

5. 벡터연산: 구성요소(element-wise) 연산, 벡터연산

6. 벡터 생성 함수: c(), 콜론이용, seq(), rep()

7. NA와 NULL

NA, NULL, NaN, Inf

## 1.벡터

벡터이름: 첫글자로 숫자를 사용, 글자사이에 "-" 사용, "." 뒤에 숫자 ~ NO

벡터에 데이터 할당(대입): "<-"와 "="는 대부분의 경우에 구분없이 사용, but "<-"가 우선순위

## 2. 벡터생성

1) 스칼라(scalar): 하나의 값, 객체, 원소가 하나인 벡터

2) 실수형 벡터:

mode(), typeof(): 객체의 자료 속성

str(): 객체의 자료 속성, 구조, 데이터 미리보기

| | mode() | typeof() | str() |
|---|---|---|---|
| 실수형 | numeric | double | num, 구조, 미리보기 |
| 정수형 | numeric | integer | int, 구조, 미리보기 |
| 문자형 | character | character | chr, 구조, 미리보기 |
| 논리형 | logical | logical | logi, 구조, 미리보기 |
| 요인형 | numeric | integer | Factor, 구조, 미리보기 |
| 복소수형 | complex | complex | cplx, 구조, 미리보기 |
| 원시형 | raw | raw | raw, 구조, 미리보기 |
| 함수 | function | closure, builtin, special | function, 함수구조 |

```
> a<-c(1,2,3); mode(a); typeof(a); str(a)
> b<-c(TRUE, FALSE); mode(b); typeof(b); str(b)
> c<-c('Tommy', 'Sally'); mode(c); typeof(c); str(c)
```

> d<-c(1+2i,2+3i) ; mode(d); typeof(d); str(d)

> e<-c(1,2,"A"); mode(e); typeof(e); str(e)

> m <- matrix(1:10, 2) # 행렬

> mode(m); typeof(m); str(m)

> sum12=function(x,y){x+2*y} # x,y 입력하면 결과가 x+2*y 되는 함수

> mode(sum12); typeof(sum12); str(sum12)

> mode(max); typeof(max) ; str(max) # 기본내장함수(builtin)

> mode(log); typeof(log); str(log)  # 기본내장함수(special)

If a function calls C-code, builtin/special refers to whether or not its arguments are evaluated before being passed to the C-code.

**표 2.1 벡터생성과 정보에 관한 함수**

> x<-c(1,3,5,7)
> names(x)<-c('x1','x2','x3','x4')
> x
> nemes(x); length(x); is,vector(x)

append() 함수: append(x,y,after=m)

> rm(x)
> x<-c(90,80,95)
> z0<-append(x, 70,  after=0) ; z0
> z1<-append(x, 70,  after=1) ; z1
> z2<-append(x, 70,  after=2) ; z2
> z3<-append(x, 70,  after=3) ; z3
> y<-c(1,2)
> v0<-append(x, y,  after=0) ; v0
> v2<-append(x, y,  after=2) ; v2

3) 정수형 벡터:

> x <- c(5, 6, 3); y <- c(5L, 6L, 3L)
> typeof(x); typeof(y)

4) 문자형(character) 벡터:

벡터는 문자열(string)을 저장, 문자열은 쌍따옴표 " " 혹은 홑따옴표 '  '사용

> y <- c("fruits", "berry")
> str(y)

5) 논리형 벡터:
구성요소가 TRUE, FALSE

```
> b <- c(TRUE, FALSE, TRUE)
> str(b)
> b1=as.integer(b); b2=as.numeric(b) # b를 정수형, 실수형 벡터로
> typeof(b1); typeof(b2)
> b11=as.logical(b1); b22=as.logical(b2) # b1, b2를 다시 논리형 벡터로
> typeof(b11); typeof(b22)
```

6) 요인형(factor) 벡터:
범주형(categorical) 자료를 표현하는 벡터.
factor() 함수를 이용해서 요인형 벡터로 변환.

```
> x <- c("O","A","B","O","A")
> xf <- factor(x)
> xf
> attributes(xf)
> str(x); str(xf); levels(xf)
> typeof(xf)
```

7) 복소수형 벡터:

```
> x<-c(3+8i, 1+2i)
> x
> str(x)
```

8) 원시형(raw) 벡터: 16진수

```
> x<-c(1,20,30)
> x16<-as.raw(x) # (1, 20, 30)을 16진수로
> x; x16
> xi=as.numeric(x16) # (1,14,1e)를 (1,20,30)으로
> str(x); str(x16); str(xi)
```

3. 속성(attributes): 객체의 정보를 나타내는 메타데이터(metadata)(다른 데이터를 설명해 주는 데이터),
attributes(), class(), dim(), dimnames(), names(), row.names(), colnames()함수...
클래스(class), 차원(dim)
차원이름(dimnames), 이름(names), 행이름(rowames), 열이름(colnames)

```
> x <-c(1,2,3)
> names(x); class(x)
> str(x); attributes(x)
> names(x) <-  c('x1','x2','x3')
```

```
> names(x); class(x)
> str(x)
> attributes(x)

> m <- matrix(1:6, 2) # 2*3 행렬
> class(m); attributes(m)
> row.names(m)<-c("row1","row2")
> colnames(m)<-c("col1","col2","col3")
> class(m); attributes(m)
> dim(m)
```

## 4. 벡터의 구성요소 선택(서브셋팅):

1) 벡터 서브셋팅: 벡터에서 구성요소를 선택, 즉 벡터의 일부분을 추출

**표 2.2 벡터 서브셋팅**

```
> x<-c(5,7,1,3,9,2)
> x[2]; x[-5]
> x[c(2,5)]; x[-c(2,5)]; x[c(-2,-5)]
> x[c(2,-5)]
> x[x>2]; x[x<=3];  x[x>1 & x<6]
> x[c(T,F,F,F,F,T)] # TRUE에 대응하는 원소만 추출
> x[c(T,F)] # (T,F,T,F,T,F)중 TRUE에 대응하는 원소만 추출
> x[c(F,T)] # (F,T,F,T,F,T)중 TRUE에 대응하는 원소만 추출
> names(x) <- c('x1','x2','x3','x4','x5','x6')
> x[c('x1', 'x4')]
```

## 5. 벡터연산

1) 구성 요소단위 연산

```
> x<-c(1,2); y<-c(3,4)
> x+y; x*y; x/y
```

2) 벡터 재활용

```
# 두 벡터 길이가 약수/배수 관계
> x<-c(1,2,3,4); y<-c(10,20)
> x+y; x*y; x/y
# 두 벡터길이가 약수/배수 관계가 되지 않으므로 경고 메시지
> x<-c(1,2,3); y<-c(10,20)
> x+y; x*y; x/y
```

3) 벡터화 연산

> x^2; sqrt(x)

4) 비교 연산

**표2.3 비교연산자**

> x<-c(5,7,1,3,9,2)
> x>2; x!=1
> sum(x>2) # 2보다 큰 원소의 합
> y<-c(1,2,3)
> x %in% y # x의 원소가 y의 구성요소인가?

**표2.4 논리연산자**

> x<-c(1,2,3);

> x==c(1,3,3) & x==c(1,2,2); x==c(1,3,3) && x==c(1,2,2)

> x==c(1,3,3) | x==c(1,2,2); x==c(1,3,3) || x==c(1,2,2)

which() 함수: 조건 만족하는 원소(구성요소)의 위치 찾기

> v= which(x>1)

> v

> x[v]; x[x>1]

all(), any() 함수 ~ 모두 TRUE, 하나라도 TRUE이면 TRUE

> all(x>1); any(x>1)

## 6. 벡터 생성 함수

**표 2.5 벡터생성함수**

## 7. NA와 NULL

NA: not available, 값이 존재하지 않음(값을 모름)

NULL: empty, 값이 없는 상태

NaN: not a number, 수학적으로 정의가 되지 않는 값

Inf: infinity, 무한대

> x1<-c(10,20,NA)

> x1; length(x1); sum(x1); sum(x1, na.rm=TRUE)

```
> x2<-c(30,40,NULL)
> x2;length(x2); sum(x2)
> x3<- 1/0; x4<- -1/0
> x3; x4
> x5<-0/0
> x5
```