R컴퓨팅

4강 데이터 구조 I

정보통계학과 장영재 교수

1 데이터의 기본 형태

2 벡터

1 데이터의 기본 형태

1 데이터의 기본 형태

- R의 특징 중 하나는 다양한 형태의 데이터 입수와 가공이 용이하며 통계분석에 최적화된 형태의 데이터 구조를 형성한다는 점

R에서 다루는 데이터의 기본적인 형태는 크게 벡터, 행렬, 배열과 리스트로 나눌 수 있음

2 벡터

2 벡터

1 벡터의 생성

- 벡터는 한 개 이상의 원소로 구성된 자료구조로서 R의 자료 객체 중에서 가장 기본이 되는 자료 객체를 의미

> 하나의 벡터의 원소는 한 가지 형태(mode)만이 가능하다는 점에 유의

2 벡터

1 벡터의 생성

보기 4-1: scan()함수를 이용하여 1에서 6까지 자연수로 이루어진 벡터를 생성하기

```
> scan() # 벡터 생성을 위한 scan()함수 입력
1: 1 2 3 # 벡터의 첫 번째부터 세 번째 원소들을 입력
4: 4 5 6 # 벡터의 네 번째부터 여섯 번째 원소들을 입력
7: # 공백
Read 6 items
[1] 1 2 3 4 5 6
```

2 벡터

1 벡터의 생성

```
〉scan(sep=“,”) # 벡터 생성을 위한 scan()함수 입력
1: 1 # 벡터의 첫 번째 원소를 입력
2: 2,3 # 벡터의 두 번째와 세 번째 원소를 입력
4: 4,5,6 # 벡터의 네 번째부터 여섯 번째 원서를 입력
7: # 공백
Read 6 items
[1] 1 2 3 4 5 6
```

2 벡터

1 벡터의 생성

보기 4-2: seq() 함수를 이용하여 1에서 6까지 자연수로 이루어진 벡터를 생성하기

```
> seq(from=1, to=6, by=1) # from=시작값, to=종료값, by=증가분(기본값)
[1] 1 2 3 4 5 6
> seq(to=6, from=1, by=1) # to=종료값, from=시작값, by=증가분
[1] 1 2 3 4 5 6
> seq(by=1, from=1, to=6) # by=증가분, from=시작값, to=종료값
[1] 1 2 3 4 5 6
```

2 벡터

1 벡터의 생성

```
〉seq(1, 6, 1) # 생략시 from=시작값, to=종료값, by=증가분 순서로 인식
[1] 1 2 3 4 5 6
```

벡터를 구성하는 수열의 시작과 끝을 나타내는 from, to, 그리고 간격을 의미하는 by 값이 지정되면 그 순서에 상관없이 동일한 벡터를 생성

2 벡터

1 벡터의 생성

보기 4-3: seq() 함수의 조건을 지정하여 벡터를 생성

```
> seq(from=1, to=7, length=3)  # from=시작값, to=종료값, 조건지정
 [1] 1 4 7
> seq(from=1, by=0.05, along=1:5)  # from=시작값, by=증가분, 조건지정
 [1] 1.00 1.05 1.10 1.15 1.20
> seq(from=1, to= 5, along=1:6) # from=시작값, to=종료값, 조건지정
 [1] 1.0 1.8 2.6 3.4 4.2 5.0
```

2 벡터

1 벡터의 생성

- length로서 조건을 지정할 경우에는 벡터의 크기, 즉 원소의 개수를 지정하여 시작값과 종료값 사이를 등간격으로 나누어 벡터를 생성

- along으로 수열을 지정하게 되면 해당 수열에 따라 동일한 크기의 벡터를 생성

2 벡터

1 벡터의 생성

- **보기 4-4: c()함수를 이용하여 기본벡터를 생성한 뒤 rep() 함수를 이용하여 생성된 벡터의 일정 원소가 반복되는 벡터를 생성**

```
> rep(c(1,2), times=2)  # 1과 2로 구성된 벡터를 2회 반복
 [1] 1 2 1 2
> rep(1:2, times=2)  # 1:2 1에서 2까지로 구성된 벡터를 2회 반복
 [1] 1 2 1 2
> rep(c(2,4), times=c(2,1))  # 앞의 원소는 2회, 위의 원소는 1회 반복
 [1] 2 2 4
```

2 벡터

1 벡터의 생성

```
〉 rep(c(2,4), each=2)  # 앞, 뒤 모든 원소를 2회 반복
   [1] 2 2 4 4
〉 rep(c(2,4,8), length=5)  # 2, 4, 8로 구성된 벡터를 반복하되 생성되는
벡터의 크기를 5로 한정
   [1] 2 4 8 2 4
```

2 벡터

2 벡터의 기본함수

› 보기 4-5: c()함수를 이용하여 1에서 6까지 자연수로 이루어진 벡터 생성

```
〉 c(1,2,3,4,5,6)  # c() 함수 내에 입력된 값을 조합하여 벡터 생성
  [1] 1 2 3 4 5 6
〉 c(1:6) # c() 함수 내 수열 값을 입력하여 벡터생성
  [1] 1 2 3 4 5 6
〉 c(1,3,2,4,6) # 순서 임의로 지정 가능
  [1] 1 3 2 4 6
```

2 벡터

2 벡터의 기본함수

- 보기 4-6: c()함수를 이용하여 “1”, “KNOU”, “한국방송통신대학교”로 이루어진 문자형 벡터를 생성

```
> c("1","KNOU","한국방송통신대학교")
[1] "1"          "KNOU"          "한국방송통신대학교"
```

문자형 벡터 생성시에는 따옴표(“”)가 필수적으로 추가

2 벡터

2 벡터의 기본함수

› 보기 4-7: c()함수를 이용하여 논리형 벡터를 생성

```
〉 c(T,F,T)
[1]  TRUE FALSE  TRUE
〉 c(TRUE,FALSE,TRUE)
[1]  TRUE FALSE  TRUE
```

2 벡터

3 벡터자료의 편집

- 벡터와 관련된 기본함수를 통해 벡터를 생성하였다면, 때에 따라서는 이를 적당히 편집하고 가공할 필요

자료의 삽입, 삭제 등과 관련된 방법들의 예
- [ ]를 이용하여 벡터의 일부 원소를 추출
- replace() 함수를 이용하여 일부 자료를 대체
- append() 함수를 이용하여 조건에 따른 위치에 자료를 삽입
- sort() 함수를 이용하여 자료를 정렬
- rank() 함수를 이용하여 자료의 순위를 출력
- order() 함수를 이용하여 오름차순에 의한 자료의 위치 값 출력

2 벡터

3 벡터자료의 편집

벡터명[원소번호 또는 조건문, …]

replace(벡터, 원소번호, 교체자료)

append(벡터, 삽입자료, after=원소번호)

sort(벡터, decreasing=FALSE, …)

rank(벡터, na.last = TRUE, ties.method =
c("average", "first", "random", "max", "min"))

order(벡터, na.last = TRUE, decreasing=FALSE)

2 벡터

3 벡터자료의 편집

- "na.last = TRUE"라고 기본값으로 지정된 옵션의 의미는 만약 결측치가 존재한다면 이러한 결측치 값들을 자료의 맨 끝에 위치하는 것으로 간주한다는 의미

- "decreasing = FALSE"라는 기본 옵션은 내림차순이 아니라는, 즉 오름차순을 기본값으로 한다는 의미

2 벡터

3 벡터자료의 편집

- 보기 4-8: ① 11에서 20까지의 값을 갖는 벡터 v1을 생성하고 세 번째와 다섯 번째 값을 추출하기 ② v1의 원소 중 15보다 큰 값만을 출력하고 v1중 두 번째와 네 번째 값만을 삭제하고 출력해 보기

```
> v1<-c(11:20) # 11에서 20까지의 값을 갖는 벡터 v1 생성
> v1
 [1] 11 12 13 14 15 16 17 18 19 20
> v1[c(3,5)]  # v1의 세 번째와 다섯 번째 값을 출력
 [1] 13 15
> v1[v1>15]  # v1의 원소 중 15보다 큰 값을 출력
 [1] 16 17 18 19 20
```

2 벡터

3 벡터자료의 편집

```
> v2<-c(1:5)
> v2
  [1] 1 2 3 4 5
> replace(v2,2,6) # v2의 두 번째 원소의 값을 6으로 변경
  [1] 1 6 3 4 5
> append(v2,8,after=5) # v2의 다섯 번째 값 다음에 8을 추가
  [1] 1 2 3 4 5 8
> v3 <- append(v2,8,after=5)
> v3
  [1] 1 2 3 4 5 8
```

2 벡터

3 벡터자료의 편집

- 보기 4-10: 아래의 규칙에 따라 생성된 벡터 x를 ① sort() 함수를 이용하여 오름차순으로 정렬하고, rank() 함수를 통해 자료의 순위를 출력 ② order() 함수를 이용하여 오름차순에 의한 자료의 위치 값을 출력

- (규칙) : 1을 세 번 반복한 뒤 1에서 5까지 2만큼의 증가분으로 수열을 생성하여 덧붙임. 이어서 rev() 함수를 이용해 1에서 5까지 이르는 크기 3인 벡터의 역순을 구해 연결하고 2를 세 번 반복한 벡터를 덧붙임

2 벡터

3 벡터자료의 편집

```
> x<-c(rep(1,3),seq(1,5,by=2),rev(seq(1,5,length=3)),rep(2,3))
> x
[1] 1 1 1 1 3 5 5 3 1 2 2 2
> sort(x) # 벡터 x를 오름차순으로 정렬
[1] 1 1 1 1 1 2 2 2 3 3 5 5
> rank(x) # 자료의 순위를 출력
[1] 3.0 3.0 3.0 3.0 9.5 11.5 11.5 9.5 3.0 7.0 7.0 7.0
> order(x) # 오름차순에 의한 자료의 위치값 출력
[1] 1 2 3 4 9 10 11 12 5 8 6 7
```

R컴퓨팅