

R컴퓨팅

6강

데이터 구조 III

정보통계학과 장영재 교수

1 리스트

2 데이터프레임



1

리스트

1 리스트

➤ 서로 다른 형태(mode)의 데이터로 구성된 객체를 의미

행렬과 배열 등이 동일한 형태의 원소로 이루어진 객체인 반면 리스트를 구성하는 성분(component)은 서로 다른 형태의 원소를 가질 수 있고, 길이도 다를 수도 있음

※ 리스트의 주요 속성

속성	설명
length	자료의 개수
mode	자료의 형태
names	각 구성요소의 이름

1 리스트

- 보기 6-1: list() 함수를 이용하여 두 개의 벡터와 한 개의 문자열을 가진 리스트를 생성하고 속성을 출력하는 예제

```
> a <- 1:10  
> b <- 11:15  
> klist <- list(vec1=a, vec2=b, descrip="example") # 리스트를 생성  
> length(klist)    # 리스트 klist의 자료의 개수  
[1] 3  
> mode(klist)      # 리스트 klist의 자료의 형태  
[1] "list"  
> names(klist)      # 리스트 klist의 각 성분의 이름  
[1] "vec1" "vec2" "descrip"
```

1 리스트

1 리스트의 생성 및 추출

➤ 리스트를 생성하고 추출하는 방법은 행렬과 배열 등과 유사

특정한 성분을 추출하거나 성분 내의 특정 원소를 추출하려면
아래와 같은 연산자를 사용

- `[[]]` 를 사용하여 리스트에서의 성분을 추출
- 성분의 이름이 부여되었다면 `$` 연산자로 구분됨
- `[]` 연산자를 사용하여 성분 내의 특정 원소를 추출

1 리스트

1 리스트의 생성 및 추출

- 보기 6-2: list() 함수를 이용하여 "A"라는 문자와 1에서 8까지의 원소를 갖는 벡터를 성분으로 하는 list1이라는 리스트를 생성한 뒤, list1의 세 번째 성분으로 논리값 T, F를 갖는 벡터를 추가하고 두 번째 성분의 아홉 번째 원소로 9를 추가

```
> list1 <- list("A", 1:8)      #list1 리스트 생성
> list1
[[1]]
[1] "A"
[[2]]
[1] 1 2 3 4 5 6 7 8
> list1[[3]] <- list(c(T, F))  #세 번째 성분을 추가
> list1[[2]][9] <- 9          #두 번째 성분에 원소 추가
```

1 리스트

1 리스트의 생성 및 추출

```
> list1
```

```
[[1]]
```

```
[1] "A"
```

```
[[2]]
```

```
[1] 1 2 3 4 5 6 7 8 9
```

```
[[3]]
```

```
[[3]][[1]]
```

```
[1] TRUE FALSE
```

```
> list1[[3]] <- NULL #세 번째 성분 삭제
```

```
> list1[[2]] <- list1[[2]][-9] #두 번째 성분의 9번째 원소 삭제
```

1 리스트

1 리스트의 생성 및 추출

```
> list1
```

```
[[1]]
```

```
[1] "A"
```

```
[[2]]
```

```
[1] 1 2 3 4 5 6 7 8
```


1 리스트

1 리스트의 생성 및 추출

- 보기 6-3: 1에서 10까지의 원소를 갖는 벡터 a와 11에서 15까지의 원소를 갖는 벡터 b를 생성하고 “example”이라는 문자와 함께 nlist라는 리스트를 생성하기

```
> a <- 1:10  
> b <- 11:15      # 벡터 a, b 생성  
> nlist <- list(vec1=a, vec2=b, descrip="example")  
> nlist           # a, b벡터 및 descrip 변수에 example 문자 부여  
$vec1  
[1] 1 2 3 4 5 6 7 8 9 10  
$vec2  
[1] 11 12 13 14 15
```

1 리스트

1 리스트의 생성 및 추출

```
$descrip
```

```
[1] "example"
```

```
> nlist[[2]][5]      # 두 번째 성분 vec2의 다섯 번째 원소
```

```
[1] 15
```

```
> nlist$vec2[c(2,3)]  # vec2의 두 번째와 세 번째 원소
```

```
[1] 12 13
```



2 데이터프레임

2 데이터프레임

- 행렬은 차원으로 표시되며 같은 형태의 객체를 가지는 반면, 데이터 프레임은 각 열들이 서로 다른 형태의 객체를 가질 수 있음

- ① 데이터 프레임은 형태(mode)가 일반화된 행렬(matrix)
- ② 데이터 프레임이라는 하나의 객체에 여러 종류의 자료가 들어갈 수 있음
- ③ 데이터 프레임의 각 열은 각각 변수와 대응
- ④ 분석이나 모형 설정에 적합한 자료 객체

2 데이터프레임

1 데이터 프레임의 생성

```
read.table(file, header = FALSE, sep = "", ...)
```

```
data.frame(객체1, 객체2, ...)
```

```
as.data.frame(대상객체, ...)
```

2 데이터프레임

1 데이터 프레임의 생성

- 보기 6-4: 데이터가 다음과 같이 작성되어 "d:/story.txt"에 저장되어 있다고 가정하자. read.table() 함수를 이용하여 이 파일을 읽어보기

num	name	age	sex
1	Lee	55	M
2	Park	47	F
3	So	35	M
4	Kim	26	F
5	Yoon	29	M

```
> d2 <- read.table("d: /story.txt", row.names =  
+ 'num', header = T)
```

- row.names = 'num' 은 데이터의 num이라는 변수가 행 이름임을 나타내며, header = T 는 첫 번째 줄이 변수명임을 의미

2 데이터프레임

1 데이터 프레임의 생성

- 보기 6-5: 문자형 벡터 char1과 수치형 벡터 num1을 생성하고 결합하여 test1이라는 데이터 프레임을 만들어 보는 예제

```
> char1 <- rep(LETTERS[1:3],c(2,2,1)) # 문자형 벡터 char1
> char1
[1] "A" "A" "B" "B" "C"
> num1 <- rep(1:3,c(2,2,1)) # 수치형 벡터 num1
> num1
[1] 1 1 2 2 3
> test1 <- data.frame(char1, num1) # test1 데이터 프레임 생성
```

2 데이터프레임

1 데이터 프레임의 생성

```
> test1
```

```
char1 num1
```

```
1 A 1
```

```
2 A 1
```

```
3 B 2
```

```
4 B 2
```

```
5 C 3
```


2 데이터프레임

1 데이터 프레임의 생성

- 보기 6-6: 문자 a부터 o까지로 이루어진 벡터를 생성하고 이를 5행 3열의 행렬로 변환한 뒤 a1으로 저장하고 데이터 프레임으로 변환

```
> a1 <- c("a","b","c","d","e","f","g","h","i","j","k","l","m","n","o")
> dim(a1) <- c(5,3) # 5행 3열인 a1 행렬 생성
> a1
[1,] [2,] [3,]
[1,] "a" "f" "k"
[2,] "b" "g" "l"
[3,] "c" "h" "m"
[4,] "d" "i" "n"
[5,] "e" "j" "o"
```

2 데이터프레임

1 데이터 프레임의 생성

```
> test3 <- as.data.frame(a1) # a1을 데이터 프레임으로 변환
> test3
  V1 V2 V3
1 a f k
2 b g l
3 c h m
4 d i n
5 e j o
```

2 데이터프레임

2 데이터 프레임의 결합

- 서로 다른 데이터 프레임을 생성하였다면, 여러 데이터 프레임을 합하여 하나의 데이터 프레임으로 생성할 수 있음

데이터 프레임을 합치는 방법에는 아래와 같은 함수가 사용

- cbind() 함수를 이용하여 옆으로 합치기
- rbind() 함수를 이용하여 아래로 합치기
- merge() 함수를 이용하여 병합하기

2 데이터프레임

2 데이터 프레임의 결합

- 보기 6-7: ① 보기 3-22의 test1과 보기 3-23의 test3라는 데이터 프레임을 cbind() 함수를 이용하여 옆으로 합치기 ② test4라는 새로운 데이터 프레임을 생성하고 rbind() 함수를 이용하여 test1과 test4를 아래로 합치기

```
> cbind(test1,test3) #test1과 test2를 옆으로 합친 형태
char1 num1 V1 V2 V3
1 A 1 a f k
2 A 1 b g l
3 B 2 c h m
4 B 2 d i n
5 C 3 e j o
> char1 <- rep(LETTERS[1:3],c(1,2,2))
```

2 데이터프레임

2 데이터 프레임의 결합

```
> char1
[1] "A" "B" "B" "C" "C"
> num1 <- rep(1:3,c(1,1,3))
> num1
[1] 1 2 3 3 3
> test4 <- data.frame(char1, num1)
> test4      # test4 데이터 프레임 생성
  char1 num1
1 A      1
2 B      2
3 B      3
4 C      3
5 C      3
```

2 데이터프레임

2 데이터 프레임의 결합

```
> rbind(test1, test4) # 아래로 합친 형태
```

```
char1 num1
```

```
1 A 1
```

```
2 A 1
```

```
3 B 2
```

```
4 B 2
```

```
5 C 3
```

```
6 A 1
```

```
7 B 2
```

```
8 B 3
```

```
9 C 3
```

```
10 C 3
```

2 데이터프레임

2 데이터 프레임의 결합

- merge() 함수를 사용하면 같은 변수에 대해서 한번만 출력
 - merge() 함수를 이용하여 test1과 test4를 병합할 경우에는 처음의 변수에 대해 알파벳 순서로 정렬이 되고 나머지 변수들이 옆으로 합쳐지게 됨

```
> merge(test1, test4) # test1, test4를 병합한 형태
```

```
char1 num1
```

```
1 A 1
```

```
2 A 1
```

```
3 B 2
```

```
4 B 2
```

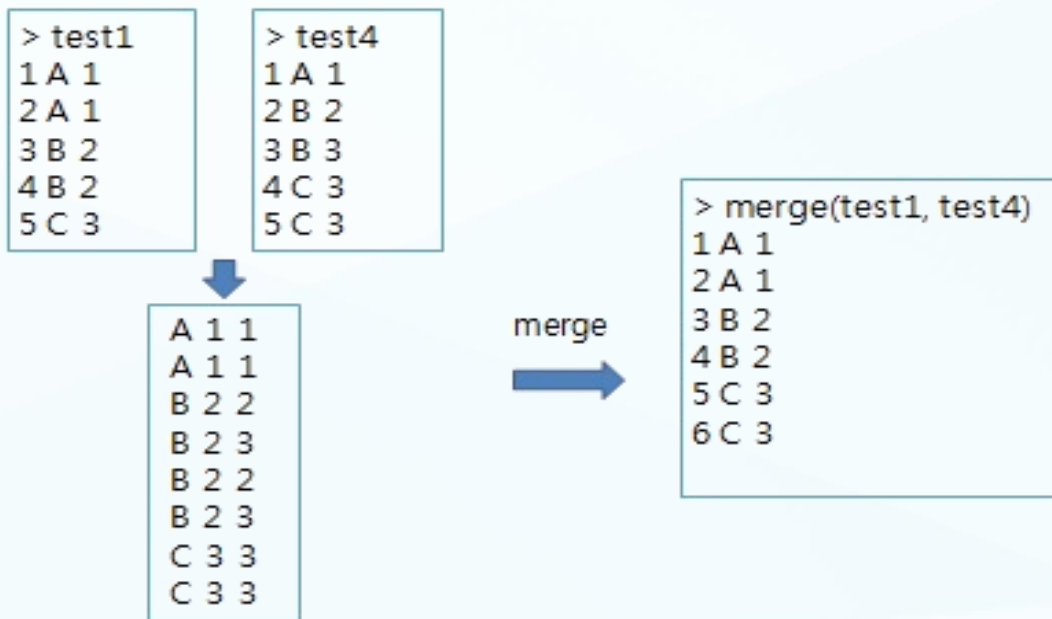
```
5 C 3
```

```
6 C 3
```

2 데이터프레임

2 데이터 프레임의 결합

※ merge() 함수의 병합원리



R컴퓨팅

수고하셨습니다

