

# R 프로그래밍

## (8주차)

2016. 04. 23(토)

장운호

(ADP 002-0004)

# 목차

## ※ 지난 주 복습

### I. 리스트(list)

### II. 데이터프레임(dataframe)

### III. 요인(factor)

### IV. apply 함수群

### V. R 데이터구조 요약

## ※ 행렬 Indexing Rule

벡터 Indexing Rule의 “원소”를 “행 과/또는 열”로 바꾸면 정확히 일치한다.

- 1) 양의 정수가 사용되면, 해당 위치의 **행과 열**을 의미한다.
- 2) 빈칸으로 둔 경우는 모든 **행 또는 열**이 된다.
- 3) 음의 정수가 사용되는 해당 위치의 **행 또는 열**은 제외한다는 의미다.
- 4) 조건식을 넣으면 조건식의 참(TRUE)인 **행과 열**이 선택된다.
- 5) 정수로 이뤄진 벡터를 넣으면,  
해당 벡터의 위치에 있는 **행 또는 열**을 선택한다.

※ 행이나 열의 이름이 지정되어 있으면, 이름 지정 時 지정된 이름을 가진 행 or 열이 선택됨.

참고자료) R과 Knitr를 활용한 데이터 연동형 문서 만들기 (고석범 저, 일부 수정 반영)



# I . 리스트(list)

# 1. 리스트(list)

리스트는 벡터의 한 종류로,  
원소를 한가지 타입이 아니라 여러가지 데이터 타입으로 구성할 수 있으며,  
심지어는 리스트내에 리스트를 원소로 가질 수도 있음.

- list함수를 활용하여 리스트를 생성함.

```
> x <- list(1:3, "a", c(TRUE, FALSE, TRUE), c(2.3, 5.9))
```

```
> str(x)
```

List of 4

```
$ : int [1:3] 1 2 3
```

```
$ : chr "a"
```

```
$ : logi [1:3] TRUE FALSE TRUE
```

```
$ : num [1:2] 2.3 5.9
```

## 2. 리스트(list) : 꾸러미 데이터 타입

리스트는 관련된 주제에 대한 데이터를 하나로 묶어주는 꾸러미라고 생각하시면 편함.

```
> c1Name <- "장운호"
> c1Age <- 46
> c1hobby <- c("등산","볼링")
> c1Visit <- c("관악산","북한산","청계산")
> customerDatabase <- list(name=c1Name, age=c1Age,
                           hobby=c1hobby, visit=c1Visit)
```

```
> customerDatabase
```

```
$name
```

```
[1] " 홍길동"
```

```
$age
```

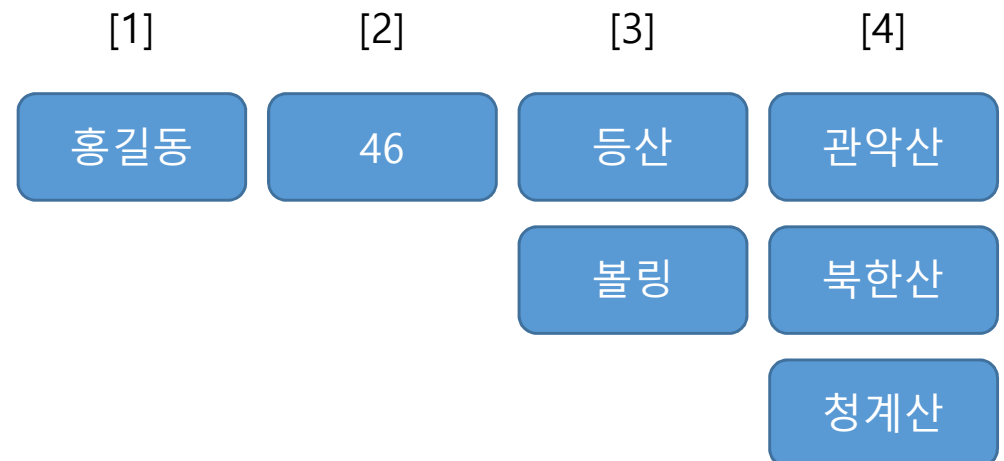
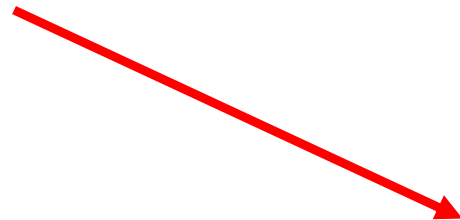
```
[1] 46
```

```
$hobby
```

```
[1] "등산" "볼링"
```

```
$visit
```

```
[1] "관악산" "북한산" "청계산"
```



### 3. 리스트(list) : 메모리 구조

겉보기는 벡터와 완전히 다르지만, 메모리에 저장되는 구조는 벡터와 동일하기 때문에, 벡터의 성질을 공유할 수 있는 것이라고 본 강사는 주장함.

```
x <- list(1:3, "a", c(TRUE, FALSE, TRUE), c(2.3, 5.9))
```

```
#list
```

```
mem_view(x)
```

```
#integer vector
```

```
intVec <- 1:4 ; mem_view(c)
```

```
#character vector
```

```
charVec <- LETTERS[1:4] ; mem_view(charVec)
```

```
#numeric vector
```

```
numVec <- c(1.1, 2.3579, 3.33333, 4.4321) ; mem_view(numVec)
```

```
#boolean vector
```

```
boolVec <- c(TRUE, FALSE, FALSE, TRUE) ; mem_view(boolVec)
```

### 3. 리스트 인덱싱/subsetting

리스트는 벡터이기 때문에 벡터와 동일한 방식으로 인덱싱이 가능하나, 이 경우, 리스트를 결과값으로 반환함.

- 리스트안의 내용만을 추출하기 위해서는 겹대괄호("[[") 연산자를 활용해야 함.  
※ 겹대괄호 연산자는 열고/닫아주어야 해서 불편한 바, 리스트의 원소에 이름이 지정되어 있는 경우는 줄여서 "\$"연산자만 써도 동일한 결과가 나옴.

```
x <- list(1:3, "a", c(TRUE, FALSE, TRUE), c(2.3, 5.9))
```

```
y <- list(int=1:3, char="a", bool=c(TRUE, FALSE, TRUE), num=c(2.3, 5.9))
```

```
> x[1]
[[1]]
[1] 1 2 3

> x[3]
[[1]]
[1] TRUE FALSE TRUE
```

```
> y[2]
$char
[1] "a"

> y[4]
$num
[1] 2.3 5.9
```

```
> x[[1]]
[1] 1 2 3

> x[[3]]
[1] TRUE FALSE TRUE
```

```
> y[[2]]
[1] "a"

> y[[4]]
[1] 2.3 5.9
```

```
> y[["char"]]
[1] "a"

> y[["num"]]
[1] 2.3 5.9
```

```
> y$int
[1] 1 2 3

> y$bool
[1] TRUE FALSE TRUE
```





## II. 데이터프레임 (dataframe)

# 1. 데이터프레임 (dataframe)

길이가 모두 같은 벡터로 만들어진 리스트를 특별히 구분하여, 데이터프레임(dataframe)이라 칭함.

```
> myFamilyNames <- c("Dad","Mom","Sis","Bro","Dog")
> myFamilyAges <- c(43,42,12,8,5)
> myFamilyGenders <- c("Male","Female","Female","Male","Female")
> myFamilyWeights <- c(188, 136, 83, 61, 44)
> myFamily <- data.frame(Name=myFamilyNames, Age=myFamilyAges,
                          Gender=myFamilyGenders, Weight=myFamilyWeights)
```

```
> myFamily
```

	Name	Age	Gender	Weight
1	Dad	43	Male	188
2	Mom	42	Female	136
3	Sis	12	Female	83
4	Bro	8	Male	61
5	Dog	5	Female	44

## 2. 데이터프레임의 특징

데이터 프레임은 matrix의 특성과 list의 특성을 공유하는 데이터 구조로, R에서 가장 광범위하게 사용되는 데이터 구조임.

열 : 변수, 속성

Name	Age	Gender	Weight
Dad	43	Male	188
Mom	42	Female	136
Sis	12	Female	83
Bro	8	Male	61
Dog	5	Female	44


행 : Case, 사례(instance)

### 3. 데이터 형태별 특성 비교

```
start.time <- Sys.time()
N <- 12
matrixdata <- matrix(rep(0,81), ncol=9)
for (i in 1:N) {
  for(j in 1:N) {
    matrixdata[i,j] <- i*j
  }
}
Sys.time() - start.time
```

```
start.time <- Sys.time()
N <- 9
matrixdata <- matrix(rep(0,81), ncol=9)
for (i in 1:N) {
  for(j in 1:N) {
    matrixdata[i,j] <- i*j
  }
}
Sys.time() - start.time
```

```
start.time <- Sys.time()
N <- 12
dfdata <- data.frame(matrix(rep(0,81), ncol=9))
for (i in 1:N) {
  for(j in 1:N) {
    dfdata[i,j] <- i*j
  }
}
end.time <- Sys.time()
```



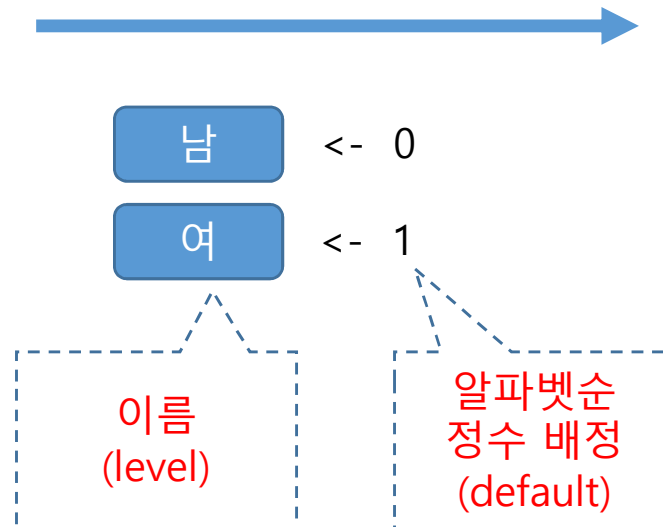
### Ⅲ. 요인 (factor)

# 1. 요인 (factor)

범주형(Categorical) 데이터를 메모리 효율적으로 처리하기 위한 데이터 타입으로 벡터 내에서 동일한 카테고리로 구분 가능한 원소를 묶어서 하나의 이름(level)으로 구분하고, 이 각각의 이름에 일련 번호를 부여하는 형태로 만들어짐.

- 외부적으로 이름이 보여지지만, R 내부적으로는 정수(Integer)로 인식되고, 처리됨.

이름	나이	성별
홍길동	25	남
이영희	60	여
김철수	45	남
한채아	19	여
김보성	48	남
한송이	25	여



이름	나이	성별
홍길동	25	0
이영희	60	1
김철수	45	0
한채아	19	1
김보성	48	0
한송이	25	1

## 2. 요인(factor) 설정 방법


각 카테고리의 이름을 levels 키워드로 지정가능하며,  
ordered 키워드를 활용하여 각 level값들의 순서를 지정해 줄 수 있음.

```
fac0 <- c("Male","Female","Female","Male")  
fac1 <- factor(c("Male","Female","Female","Male"))  
fac0;fac1  
fac2 <- factor(c(1,2,1,1,2,3))  
levels(fac1);levels(fac2)
```

```
fac3 <- factor(sample(c("high", "middle", "low"), 20, replace=TRUE))  
table(fac3)  
barplot(table(fac3))
```

```
fac4 <- factor(fac3, levels=c("low","middle","high"),  
               ordered=TRUE)  
barplot(table(fac4))
```

```
fac5 <- factor(sample(0:1, 20, replace=TRUE))  
fac6 <- factor(fac5, labels=c("abs","pre"))
```



## IV. apply 함수群



# 1. apply() 함수

행 또는 열 전체로 분석가가 지정한 함수를 적용하고,  
그 결과값을 벡터로 반환하는 함수.

사용 방법

```
apply( X=필요한 정보를 찾아 붙이고자 하는 대상데이터,  
       margin= 1 또는 2 , # 1는 행 방향, 2는 열방향 반복을 의미  
       FUN= 적용할 함수명 #익명함수 적용가능  
)
```

## 2. lapply() 함수

리스트 데이터 형식의 각 원소에 분석가가 지정한 함수를 반복 적용하고, 그 결과값을 리스트 형태로 반환해주는 함수임.

사용 방법

**lapply(** **X=**필요한 정보를 찾아 붙이고자 하는 대상데이터(리스트 타입 필수),  
          **FUN=** 적용할 함수명 **#익명함수 적용가능**  
**)**

### 3. sapply() 함수

lapply와 동일한 기능을 수행하지만, 결과값 반환이 리스트가 아닌 데이터프레임 형태로 반환함.

사용 방법

```
sapply( X=필요한 정보를 찾아 붙이고자 하는 대상데이터,  
        FUN= 적용할 함수명 #익명함수 적용가능  
)
```

## 4. tapply() 함수

요인(factor) 데이터 타입의 column(변수)를 포함하고 있는 데이터에서 팩터변수를 기준으로 subset을 나눠서 집계를 해주는 함수

### 사용 방법

```
tapply( X= subset으로 나누어서 집계하고자 하는 데이터,  
        # 벡터 or 데이터프레임의 한 열  
        INDEX= subset의 기준이 될 요인(factor) 변수로 이루어진 벡터 ,  
        FUN= 적용할 함수명 #익명함수 적용가능  
        )
```



# V. R 데이터 구조 요약

# 1. R 데이터 구조 요약

같은 종류의  
데이터 타입을 가진  
벡터를  
수용하는  
Data Type

배열(array)  
교재 3장

매트릭스(matrix)  
교재 3장

N차원데이터  
수용가능

행(row)과  
열(column)로  
이루어진  
2차원 데이터

다른 종류의  
데이터 타입을 가진  
벡터들을  
결합시킬 수 있는  
Data Type

리스트(list)  
교재 4장

데이터프레임(dataframe)  
교재 5장

[1] [[1]]	[2] [[2]]	[3] [[3]]	[4] [[4]]
장운호	46	등산	관악산
		볼링	북한산
			청계산

이름	나이	취미	비고
장운호	25	등산	관악산
홍길동	60	볼링	A클럽
김철수	45	볼링	B클럽

범주형 데이터  
벡터들을  
효율적으로 표현하는  
Data Type

요인(factor)  
교재 6장

남	0
여	1

이름  
(level)

알파벳순 정수  
배정 (default)

End of Document.

감사합니다.