[5]Data Processing

Data Processing

- 데이터 분리
- 그룹별 처리
- 조건검색
- NA 처리
- Sorting
- Sampling
- Replace

데이터 조작

- split() : 그룹화된 변수를 분리, 데이터프레임의 factor별로 데이터 분리
  - 형식: 'split(데이터, factor)'

```
>data(sleep )  #sleep 내장데이터 가져오기
> str(sleep)    #sleep 구조 확인
'data.frame':20 obs. of  3 variables:
 $ extra: num  0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0 2 ...
 $ group: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ ID   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
```

```
> #split
> g= split(sleep, sleep$group)   #group로 분리하여 리스트로 반환
> str(g)
List of 2
 $ 1:'data.frame':          10 obs. of  3 variables:
  ..$ extra: num [1:10] 0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0 2
  ..$ group: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1
  ..$ ID   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
 $ 2:'data.frame':          10 obs. of  3 variables:
  ..$ extra: num [1:10] 1.9 0.8 1.1 0.1 -0.1 4.4 5.5 1.6 4.6 3.4
  ..$ group: Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 2
  ..$ ID   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10
> g$'1'
   extra group ID
1    0.7     1  1
2   -1.6     1  2
...
9    0.0     1  9
10   2.0     1 10
> names(g) = c("G1","G2")    #리스트명 변경
```

그룹별 데이터 조작

- tapply() : 그룹화된 단일 변수에 함수적용
  형식: tapply(단일데이터, 그룹, 함수)
  - 그룹은 factor 형 데이터
  - 데이터가 속한 그룹별로 함수 적용

```
>tapply(sleep$extra, sleep$group,mean ) #그룹별 초과시간의 평균
   1    2
0.75 2.33

> d1 = read.table("data1.txt", header=T)
> str(d1)
'data.frame':6 obs. of  5 variables:
 $ name  : Factor w/ 6 levels "hong","jin","kim",..: 3 5 4 2 6 1
 $ btype : Factor w/ 4 levels "A","AB","B","O": 1 1 4 2 3 4
 $ height: num  161 183 155 176 170 ...
 $ weight: num  55.3 83.6 60.5 77.8 69.3 75.7
 $ age   : int  22 35 19 46 25 33

> tapply(d1$age, d1$btype, mean) #혈액형 별 평균 나이
   A   AB    B    O
28.5 46.0 25.0 26.0
```

그룹별 데이터 조작

- aggregate() : 그룹화된 다중 변수에 함수적용
  형식: aggregate(다중데이터, list, 함수)'
  - List : 그룹을 리스트로 생성
  - 각 데이터가 속한 그룹별로 함수 적용

```
>aggregate(d1[,c(3:5)], list(btype=d1$btype), mean) #혈액형별 키,몸무게,나이의 평균
  btype height weight  age
1     A 171.95  69.45 28.5
2    AB 176.30  77.80 46.0
3     B 169.50  69.30 25.0
4     O 167.35  68.10 26.0
```

name	btype	height	weight	age
kim	A	160.7	55.3	22
min	A	183.2	83.6	35
lee	O	155.4	60.5	19
jin	AB	176.3	77.8	46
park	B	169.5	69.3	25
hong	O	179.3	75.7	33

dataframe : d1

조건검색

- which : 벡터 또는 배열에서 주어진 조건을 만족하는 값의 색인을 검색
  - 형식 : which(조건식)
  - which.min : 최소값의 위치, which.max: 최대값의 위치

```
> which(d1$name =="park")   #조건을 만족하는 위치 값
[1] 5
> i1 = which(d1$btype =="B")
> d1[i1,]                          #혈액형이 B인 d1목록 확인
  name btype height weight age
5 park     B  169.5   69.3  25
> i3 = which(d1$btype =="B" & d1$name =="park")  # 여러 조건 만족 위치값
> d1[i3,]
  name btype height weight age
5 park     B  169.5   69.3  25
> i4 = which(d1$age >= 25 | d1$height >= 170)  # 나이가 25세 이상이거나  키가 170이상
> d1[i4,]
  name btype height weight age
2  min     A  183.2   83.6  35
4  jin    AB  176.3   77.8  46
5 park     B  169.5   69.3  25
6 hong     O  179.3   75.7  33
> maxa=which.max(d1$age)       #최대나이
> d1[maxa,]
  name btype height weight age
4  jin    AB  176.3   77.8  46
```

NA 처리

```
>  ad1 = c("park", NA, 177.8, 78.4, 23)
> ad2 = c("jung", "B", NA, NA, 27)
> d2 = rbind(d1, ad1, ad2)                #NA값을 가진 2개의 행을 d1에 추가하여 d2생성

> is.na(d2)                               #NA 여부 확인
   name btype height weight   age
1 FALSE FALSE  FALSE  FALSE FALSE
2 FALSE FALSE  FALSE  FALSE FALSE
3 FALSE FALSE  FALSE  FALSE FALSE
4 FALSE FALSE  FALSE  FALSE FALSE
5 FALSE FALSE  FALSE  FALSE FALSE
6 FALSE FALSE  FALSE  FALSE FALSE
7 FALSE  TRUE  FALSE  FALSE FALSE
8 FALSE FALSE   TRUE   TRUE FALSE

> i = which(is.na(d2$btype))              #혈액형이 NA인 경우
> d2[i,]
  name btype height weight age
7 park  <NA>  177.8   78.4  23

> na.omit(d2) > #na.omit, NA값이 있는 행 제거
  name btype height weight age
1  kim      A  160.7   55.3  22
2  min      A  183.2   83.6  35
3  lee      O  155.4   60.5  19
4  jin     AB  176.3   77.8  46
5 park      B  169.5   69.3  25
6 hong      O  179.3   75.7  33
```

데이터 정렬

- Sort() : 데이터를 정렬하여 반환
- Order(): 데이터 순서에 대한 인덱스 반환

```
> d1$age
[1] 22 25 19 46 25 33
> sort(d1$age)
[1] 19 22 25 25 33 46
>sort(d1$age, decreasing = TRUE) # 내림차순 정렬
[1] 46 35 33 25 22 19
> order(d1$age)   #ascending
[1] 3 1 5 6 2 4
> order(-d1$age)  #descending
[1] 4 2 6 5 1 3
> ageA = order(d1$age)
> d1[ageA,]
  name btype height weight age
3  lee     O  155.4   60.5  19
1  kim     A  160.7   55.3  22
2  min     A  183.2   83.6  25
5 park     B  169.5   69.3  25
6 hong     O  179.3   75.7  33
4  jin    AB  176.3   77.8  46
> ageHA = order(d1$age, d1$height)
> d1[ageHA,]
  name btype height weight age
3  lee     O  155.4   60.5  19
1  kim     A  160.7   55.3  22
5 park     B  169.5   69.3  25
2  min     A  183.2   83.6  25
6 hong     O  179.3   75.7  33
4     jin    AB  176.3   77.8  46
```

name	btype	height	weight	age
kim	A	160.7	55.3	22
min	A	183.2	83.6	25
lee	O	155.4	60.5	19
jin	AB	176.3	77.8	46
park	B	169.5	69.3	25
hong	O	179.3	75.7	33

dataframe : d1

Sampling

- 무작위(random)로 데이터 추출
  - sample(데이터, 개수, 복원추출여부)
  - 복원(replace) 추출(TRUE) / 비복원 추출 (FALSE)
  - 재현성을 위한 seed 값 설정

```
#sampling
>set.seed(1)
> x=1:10
> sample(x,size=10, replace=TRUE)
 [1]  3  4  6 10  3  9 10  7  7  1
> sample(x,size=10, replace=FALSE)
 [1]  3  2  6 10  5  7  8  4  1  9

> #dataframe에서의 sampling
> n = rownames(d1)
> n3 =sample(n,size=3, replace=FALSE)
> (d1[n3,])
  name btype height weight age
5 park     B  169.5   69.3  25
2  min     A  183.2   83.6  35
4  jin    AB  176.3   77.8  46
```

Replace

- 벡터의 값을 변경

  replace(x, list, values)
  - x : 대상 벡터
  - list : 변경할 값의 인덱스
  - values : 값

```
#replace
ad2 = c("jung", "B", NA, NA, 27)
d2 = rbind(d1, ad2)
replace(d2$name, which(d2$name=="hong"), "hong2")
replace(d2, which(is.na(d2)), 0)
```

dplyer를 이용한 데이터 처리

```
library(dplyr)
# 파이프 사용 : x %>% f(x)
iris %>% head    #head(iris)
iris %>% head(10)
#행 정렬
arrange(iris, -Sepal.Length)  #Sepal.Length 역순 정렬
iris %>% arrange(Sepal.Length, Sepal.Width)  #Sepal.Length 가 같다면  Sepal.Width순으로 정렬

#select : 열(속성) 선택
select (iris,Sepal.Length,Sepal.Width )
iris %>% select(Sepal.Length,Sepal.Width )

#filter : 조건에 따른 데이터 선택
d1 = filter(iris, Species == 'setosa')

#mutate : 속성값을 변환하거나 새로운 속성 설정
iris %>%
  mutate(SepalRatio = Sepal.Width/Sepal.Length,
         PetalRatio = Petal.Width/Petal.Length)

#sample_n, sample_frac : 랜덤샘플링
sample_n(iris, 10)
sample_frac(iris, 0.01)

#distinct : 고유행 검색
distinct(select(iris, Sepal.Width))


#group_by : 데이터를 그룹으로 나눈후 연산 적용,
#group_by + summarize
iris %>%
  group_by(Species) %>%
  summarize(median(Sepal.Width, Sepal.Length)),
                  median(Petal.Width, Petal.Length))
```

연습문제

- iris dataset을 이용하여 다음을 처리

  (1)Sepal.width가 평균보다 큰 것 확인

  (2)Petal.Length가 3~5범위 확인

  (3)Petal.width가 중간값보다 작은 것 확인

  (4) Species 별로 최대, 최소, 평균, 표준편차 계산

  (5) Sepal.width 의 order 확인

  (6) Petal.Length 로 정렬, 같으면 Petal.Width 로 정렬

  (7) Species(종)로 분리

  (8) Species(종) 별로 10개씩 추출

  (9)(8)에서 추출된 것을 iris2로 결합