그룹별 데이터 조작

그룹별 데이터 조작

- 데이터 조작은 전체 데이터 분석 시간의 80%를 차지함
- 이 과정에서 데이터를 세분하고 세분화된 데이터에 대해 반복적인 연산이 필요하게됨.
- Split-Apply-Combine이라고도 함
  - 빅데이터 처리를 위해 하둡을 사용한 경우, 맵리듀스와 비슷한 과정임

data.frame의 행과 컬럼 합치기

- rbind()와 cbind()는 각각 행 도는 컬럼 행태로 주어진 벡터, 행렬, 데이터 프레임을 합쳐고 결과를 행렬 또는 데이터 프레임으로 출력함.

  - rbind() : 지정한 데이터를 행으로 합침
  - cbind(): 지정한 데이터를 컬럼으로 합침

    - rbind(c(1, 2, 3), c(4, 5, 6))
    - cbind(c(1,2,3), c(4, 5, 6))

apply계열 함수

- R에는 벡터, 행렬 또는 데이터 프레임에 임의의 함수를 적용한 결과를 얻기 위한 apply계열 함수가 있음.
- 이 함수들은 데이터 전체에 함수를 한번에 적용하는 벡터 연산을 수행하므로 속도가 빠름

  - apply()
    - 배열 또는 행렬에 주어진 함수를 적용한 뒤 그 결과를 벡터, 배열, 리스트로 반환함
  - lapply()
    - 벡터, 리스트 또는 표현식에 함수를 적용하여 그 결과를 리스트로 반환함
  - sapply()
    - lappy와 유사하지만 결과를 벡터, 행렬, 배열로 반환
  - tapply()
    - 벡터에 있는 데이터를 특정 기준에 따라 그룹으로 묶은 뒤 각 그룹마다 주어진 함수를 적용하고 그결과를 반환함
  - mapply()
    - sapply의 확장된 버전으로 , 여러 개의 벡터 도는 리스트를 인자로 받아 함수에 각데이터의 첫째 요소를 적용한 결과, 둘째 요소를 적용한 결과, 셋째 요소를 적용한 결과등을 반환함

apply() 함수

- 행렬의 행 또는 열 바향으로 특정 함수를 적용하는데 사용함.

apply(
    X, 배열 또는 행렬
    MARGIN, 함수를 적용하는 방향, 1은 행 방향, 2는 열방향
      c(1,2)는 행과 열 발향 모두를 의미함
    FUNC 적용할 함수
)

- rowSums(), rowMeans(), colSums(), colMeans()

rowSum(
    X, 배열 또는 숫자를 지정한 데이터 프레임
    Na.rm=FALSE   NA를 제외할지 여부
)

lapply() 함수

- 리스트를 반환하는 특징이 있는 apply계열 함수

```
lapply(
    X           벡터, 리스트, 표현식 또는 데이터 프레임
    FUNC        적용할 함수
    ...         추가 인자, 이 인자들은 FUNC에 전달됨
)
```

- 리스트보다는 벡터 도는 데이터 프레임을 사용하기 때문에 직관적인 면이 있으므로 lapply()의 결과를 벡터 또는 데이터 프레임으로 변환할 필요가 있음 이경우 unlist()를 사용함

```
unlist(
    X,              R객체, 보통 리스트나 벡터
    Recursive=FALSE,    X에 포함된 리스트 역시 재귀적으로 변환할지 여부
    Un.name=TRUE        리스트 내 값의 이름을 보존할지 여부
)
```

do.call함수

- 함수를 리스트로 주어진 인자에 적용하여 결과를 반환함

```
do.call(
    what,          호출할 함수
    args           함수에 전달할 인자 리스트
)
```

sapply()함수

- sapply()는 lapply()함수와 유사하지만 리스트 대신 행렬, 베거 등의 데이터 타입으로 결과를 반환하는 특징을 가짐

```
sapply(
    X,          벡터,리스트, 표현식, 데이터 프레임
    FUNC,       적용함수
    ...
)
```

  - 반환 값이 FUNC의 결과가 길이가 1인 벡터들이면 벡터, 길이가 1보다 큰 벡터들이면 행렬임

tapply()함수

- tapply()는 그룹별로 함수를 적용하기 위한 apply함수임

tapply(

X, 벡터

INDEX, 데이터를 그룹으로 묶을 색인, 팩터를 지정해야 하고, 팩터가 아닌 타입이 지정되면 팩터로 형 변환됨

FUNC, 각 그룹마다 적용할 함수

...

)

mapply()함수

- mapply()와 sapply()함수와 유사하지만 다수의 인자를 함수에 넘긴다는 점에서 차이가 있음.
- 주요 사용 목적은 다수의 인자를 받는 함수 FUNC()이 있고 FUNC()에 넘겨줄 인자들이 데이터로 저장되어 있을때, 데이터에 저장된 값들을 인자로 하여 함수를 호출함

```
mapply(
    FUNC        실행할 함수
    ... 적용할 인자
)
```

doBy패키지

- 데이터 분석에서는 데이터 전체에 대해 함수를 호출하기 보다는 데이터를 그룹별로 나눈후 각 그룹별로 함수를 호출함.
- tapply()외에도 이런 목적에 특화된 doBy패키지가 있음.

  - doBy::summaryBy()
    - 데이터 프레임을 컬럼 값에 따라 그룹으로 묶은후 요약 값을 계산함.
  - doBy::orderBy()
    - 지정된 컬럼 값에 따라 데이터 프레임을 정렬
  - doBy::sampleBy()
    - 데이터 프레임을 특정 컬럼 값에 따라 그룹으로 묶은 후 각 그룹에서 샘플(sample) 추출함

데이터 분리 및 병합

- 데이터를 조건에 따라 분리하는 split(), subset()함수
- 분리되어있는 데이터를 공통된 값에 따라 병합하는 merge()함수

  - split()함수
    - 주어진 조건에 따라 데이터를 분리함.
  - subset()함수
    - 주어진 조건을 만족한느 데이터를 선택함
  - merge()함수
    - 데이터를 공통된 값을 기준으로 병함함.

데이터 정렬

- 데이터를 정렬하는 함수인 sort()와 order()함수가 있음
- sort()는 주어진 데이터를 직접 정렬해주는 함수
- order()는 데이터를 정렬했을때의 순서를 반환함.

데이터 프레임 컬럼 접근

- 데이터 프레임에 지정된 컬럼을 매번 df$colname과 같은 형식으로 접근하면 매번 데이터 프레임이름과 df와 $를 반복하게 되어 코드가 가독성이 떨어짐. 리스트인 경우도 비슷함.
  - with()
    - 코드 블록 안에서 필드 이름만으로 데이터를 곧바로 접근할 수 있음
  - within()
    - with()와 동일한 기능을 제공하지만 데이터에 저장된 값을 손쉽게 변경하는 기능을 제공함
  - attach()
    - Attach()이후 코드에서는 빌드 이름만으로 데이터를 곧바로 접근할 수 있음.
  - detach()
    - Attach()의 반대 역살을 함

그룹별 연산

- doBy가 데이터를 그룹별로 나눈 후 특정 계산을 적용하기 위한 함수들의 패키지인 반면, aggregate()는 일반적인 그룹별 연산을 위한 함수임.

MySQL 연동

- 데이터가 MySQL과 같은 데이터베이스에 저장되어있는 경우, R과 병행하고 싶다면 R과 MySQL간의 데이터 입출력 방법을 고려해야함

- 이러한 목적으로 RMySQL 패키지가 있음.

- Building R for Windows에서 RTools35.exe를 받음
  - https://cran.r-project.org/bin/windows/Rtools/
  - 설치시 Select Additional Tasks의 모든 항목을 체크함
  - 환경변수 세팅 MYSQL_HOME => C:\Program Files\MySQL\MySQL Server 5.7
  - C:\Program Files\MySQL\MySQL Server 5.7\lib\libmysql.lib를 C:\Program Files\MySQL\MySQL Server 5.7\lib\opt\로 복사함

  - install.packages("RMySQL", type = "source")
  - library(RMySQL)

MySQL 연동

```
RMySQL::dbConnect(
    drv,        데이터베이스 드라이버
    user,   사용자 이름
    password,       비밀번호
    dbname,         데이터베이스 이름
    host    호스트
)

con <- dbConnect(MySQL(), user="test", password="1234",
            dbname="test", host="127.0.0.1)
dbListTables(con)
```