데이터 가공 패키지

데이터 처리 및 가공 패키지

- data.table
  - R의 데이터 프레임을 대신할 수 있는 더 빠르고 편리한 데이터 타입
- plyr
  - 데이터 분할하고(split), 분할된 결과에 함수를 적용한 뒤(apply), 그 결과를 재조합(combine)함
- sqldf
  - SQL을 사용한 데이터 처리

SQL을 사용한 데이터 처리

- R에는 많은 데이터 처리 함수가 있어 데이터를 편리하게 조작할 수 있다는 장점이 있으나 한편으로는 원하는 형태로 데이터를 만들기 위해서 여러 가지 함수를 알아야 하는 점이 부담됨.
- sqldf는 이런 부담을 줄이기 위해 큰 도움이 되는 패키지로, SQL 문을 사용할 줄 아는 사용자가 더욱 쉽게 데이터를 접근할 수 있게 해줌

- sqldf는 SQL 명령이 주어지면 자동으로 스키마를 생성하고 데이터를 테이블로 로드한 뒤 SQL 문을 수행함
- SQL의 실행 결과는 다시 R로 적재됨.
- 작업은 자동으로 이루어지기 때문에 사용자가 힘들여 데이터베이스를 설치하고 환경을 설정하는 작업이 필요 없음.
- 성능 최적화가 최대한 이루어진 데이터베이스 기술을 활용하게 되어 데이터 처리 성능도 상당히 우수함.

SQL을 사용한 데이터 처리

- iris에는 몇가지 종류의 꽃이 있는지 출력해줌
  - sqldf("select distinct Species from iris")
- setosa에 속하는 데이터에서 Sepal.Length의 평균
  - sqldf("select avg(Sepal_Length) from iris where Species='setosa'")
- R의 기본 함수들로 수행한다면 subset( )으로 원하는 종의 데이터를 얻은 뒤 mean( )을 적용
  - mean(subset(iris, Species == "setosa")$Sepal.Length)

- 종별 Sepal.Length의 평균은 SQL의 group by를 사용해 손쉽게 처리할 수 있음.
  - sqldf("select species, avg(sepal_length) from iris group by species")

분할, 적용, 재조합을 통한 데이터 분석(plyr패키지)

- plyr 패키는 데이터를 분할하고split, 분할된 데이터에 특정 함수를 적용한 뒤apply, 그 결과를 재조합combine하는 세 단계로 데이터를 처리하는 함수들을 제공함.
- plyr의 입력은 배열, 데이터 프레임, 리스트가 될 수 있음.
- 출력 역시 배열, 데이터 프레임, 리스트가 될 수 있으며 아무런 결과도 출력하지 않을 수도 있음.
- plyr은 데이터의 분할, 계산, 조합을 한 번에 처리해주어 여러 함수로 처리해야 할 일을 짧은 코드로 대신해줌.
- 입력과 출력에서 다양한 데이터 타입을 지원해주어 데이터 변환의 부담을 크게 덜어줌.

분할, 적용, 재조합을 통한 데이터 분석(plyr패키지)

- plyr의 데이터 처리 함수들은 {adl}{adl_}ply 형태의 5글자 함수명을 사용함.
- 첫 번째 글자는 입력 데이터 타입에 따라 각각 배열(a), 데이터 프레임(d), 리스트(l)로 정해짐.
- 두 번째 글자는 출력 데이터 타입으로 a, d, l 또는 _로 정해지는데 이 중 _는 아무런 출력도 내보내지 않음을 뜻함.
- 예)
  - adply( )는 입력이 배열, 출력이 데이터 프레임임.
  - llply( )는 입력과 출력이 리스트임.

분할, 적용, 재조합을 통한 데이터 분석(plyr패키지)

문자	용도	의미
a	입력, 출력	배열
d	입력, 출력	데이터 프레임
l	입력, 출력	리스트
_	출력	아무런 출력도 없음

`{adl}`

입력 데이터 타입

`{adl_}`

출력 데이터 타입

`ply`

분할, 적용, 재조합을 통한 데이터 분석(plyr패키지)

- plyr에는 이외에도 데이터 프레임 또는 배열을 입력으로 받고 a, d, p, _ 유형의 출력을 지원하는 {adp_}ply( ) 형태의 특별한 함수들이 있음.
- apply 계열 함수의 mapply( )와 유사하게 다수의 인자를 함수에 넘겨 처리하는 함수지만 출력을 좀 더 유연하게 지정할 수 있는 점이 다름.

  - > install.packages("plyr")
  - > library(plyr)

분할, 적용, 재조합을 통한 데이터 분석(plyr패키지)

- adply()
  - adply( )는 배열(a)을 받아 데이터 프레임(d)을 반환하는 함수임.
  - 입력이 반드시 배열일 필요는 없음.
  - 주어진 입력을 숫자 색인으로 읽을 수 있는지(즉, 행렬처럼 다룰 수 있는 형태의 데이터인가) 하는 점이 중요함.
  - 데이터 프레임도 숫자 색인으로 각 행이나 열을 접근할 수 있어 adply( )를 적용할 수 있음.
  - 배열을 분할하고 함수를 적용한 뒤 결과를 데이터 프레임으로 반환함.

```
plyr::adply(
  .data,        # 행렬, 배열, 또는 데이터 프레임
                # 함수를 적용할 방향. 1(행 방향), 2(컬럼 방향) 또는
                c(1, 2)(행과 컬럼 모두의 방향)을 지정할 수 있음.
  .margins,
  .fun=NULL  # .margin 방향으로 잘려진 데이터에 적용할 함수
)
```

반환 값은 데이터 프레임임.

분할, 적용, 재조합을 통한 데이터 분석(plyr패키지)

- ddply()
  - ddply( )는 데이터 프레임(d)을 입력으로 받아 데이터 프레임(d)을 내보내는 함수임.
  - 데이터 프레임을 분할하고 함수를 적용한 뒤 결과를 데이터 프레임으로 반환함.

```
plyr::ddply(
  .data,
  .variables,  # 데이터를 그룹 지을 변수명
  .fun=NULL
)
```

반환 값은 데이터 프레임임.

분할, 적용, 재조합을 통한 데이터 분석(plyr패키지)

- 그룹마다 연산을 쉽게 수행하기
- plyr의 예에서는 adply( ) 또는 ddply( )에 임의의 사용자 정의 함수를 넘겨주어 분석을 수행함.
- 공통적으로 자주 사용하는 유형의 계산은 transform( ), mutate( ), summarise( ), subset( )을 사용하면 더 간단히 표현할 수 있음.

분할, 적용, 재조합을 통한 데이터 분석(plyr패키지)

- base::transform : 객체(예를 들면, 데이터 프레임)를 변환함.

```
base::transform(
  _data # 변환할 객체
  ...   # 태그=값 형태의 인자들
)
```

- plyr::mutate : 데이터 프레임에 새로운 컬럼을 추가하거나 기존 컬럼을 수정함.

```
plyr::mutate(
  .data, # 변환할 데이터 프레임
  ...    # 새로운 컬럼 정의. 컬럼명=값 형식
)
```

분할, 적용, 재조합을 통한 데이터 분석(plyr패키지)

- plyr::summarise : 데이터 프레임을 요약함.

```
plyr::summarise(
  .data, # 요약할 데이터 프레임
  ...    # 변수=값 형태의 인자들
)
```

- base::subset : 벡터, 행렬, 데이터 프레임의 일부를 반환함.

```
subset(
  x,     # 일부를 취할 데이터
  subset # 데이터를 선택할지 여부를 지정한 논릿값 벡터
)
subset(
  x,
  subset,
  select # 선택할 컬럼의 벡터. 제외할 컬럼은 -를 붙여 표시함
)
```