PNU STAT - dplyr & data.table & dtplyr tutorial (12.26)

시작하기에 앞서 유의사항

1. 처음 보는 내용 → 집중, 아는 내용 → Remind
2. 다운받아 주세요(github.com/hotorch)

> hotorch/pnustat_lec_201912
> https://github.com/hotorch/pnustat_lec_201912

3. 짧은 시간 내에 많은 내용들이 들어가 있기 때문에 핵심만 다룰 예정입니다. 집중해서 잘 들어주세요!

4. 여유가 되면 실습할 시간을 따로 줄 생각입니다.

5. 이론의 비중이 상당히 작기 때문에 따로 공부하시는 것을 추천합니다.

6. 실습때 다루는 코드도 상당히 쉬운 수준이기 때문에 따로 공부하시면서 응용하셔야 합니다!

7. 막히거나 에러뜨거나 질문 → 손

전처리?

여러분이 분석 시간의 7~80%를 투자하는 시간입니다.

모델에 넣어 좋은 결과를 얻고 싶으시다면 많은 시간을 할애하셔야 합니다.**(쓰레기를 넣으면 쓰레기가 나오듯이!)**

이론의 탄탄함도 중요하지만 분석 Tool을 다루는데 있어서 **경쟁력은 전처리 실력**이라고 생각합니다.

dplyr

pipe operator %>% 는 이런 느낌이에요(in dplyr)

```
g(f(y)) == y %>% f() %>% g()
g(f(x,y,z)) == y %>% f(x,.,z) %>% g
```

장점은 코드를 보았을 때 **직관적인 해석**이 용이함

SQL vs dplyr

sql 구문 작동 순서(출처)

dplyr 직관적인 해석 순서(출처)

⑤ SELECT 컬럼명
① FROM 테이블명
② WHERE 조건식
③ GROUP BY 컬럼이나 (표현식)
④ HAVING 조건식(집계함수 조건)
⑥ ORDER BY 컬럼명(표현식)

```
filter(
  summarise(
    select(
      group_by(hflights, Year, Month, DayofMonth),
      Year:DayofMonth, ArrDelay, DepDelay
    ),
    arr = mean(ArrDelay, na.rm = TRUE),
    dep = mean(DepDelay, na.rm = TRUE)
  ),
  arr > 30 | dep > 30
)
```

순서 설명

1. 조회 대상 테이블을 확인 (FROM 절)
2. 조회하고자 하는 데이터를 추출하는 조건 확인 (WHERE 절)
3. 행들을 명시된 컬럼을 기준으로 그룹화 (GROUP BY 절)
4. 그룹화 된 데이터 중 조건에 맞는 데이터만 출력 (HAVING 절)
5. 명시된 데이터만 화면에 출력 (SELECT 절)
6. 원하는 순서대로 정렬 (ORDER BY 절)

---

dplyr의 주요 함수

- **filter** : 행 조건을 줘서 불러옴, sql에서 where
- **select** : 필요한 열만 선택
- **mutate** : 새로운 컬럼을 계산해서 생성, 파생변수를 만듬
- **arrange** : 조건에 따라 재정렬
- **group_by** : 그룹을 조건으로 사용
- **summarise** : 요약형 계산을 진행, 위와 같이 병행해서 사용함
- left join

code 연습!!

---

tidyr(따로 공부 해보세요!)

▼ spread : long폼 → wide폼

spread () Function : spread(data, key, value)

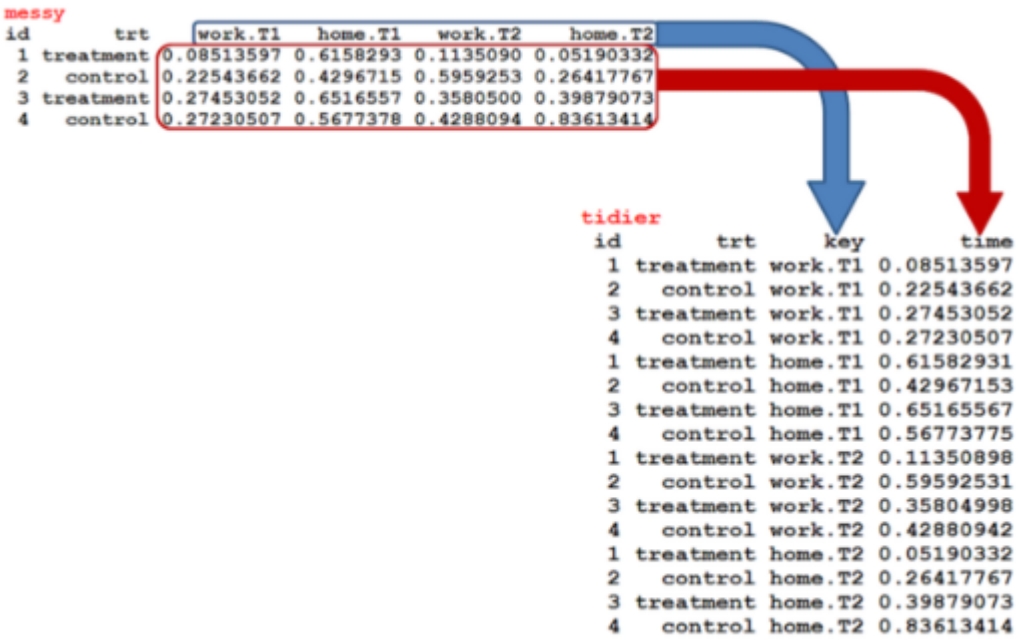
- data : data frame

- key : wide하게 변수명들이 될 column name
- value : key값들의 value

▼ gather : wide폼 → long폼

gather() Function : gather(data, key, value, ...)

- data : data frame
- key : column name들의 새로운 variable을 생성
- value : variable value값들의 variable을 생성
- ... : 변환할 column 지정



▼ seperate : 한 열에서 데이터를 지정 조건으로 분리

separate() Function : separate(data, col, into, sep = " “)

- data : data frame
- col : separate 하려는 변수
- into : 새롭게 넣게되는 변수이름
- sep : separte의 구분자(char, num, or symbol)

▼ unite : 여러 열의 데이터를 한 열로 합침

unite() Function : unite(data, col, ..., sep = " ")

- data : data frame
- col : Merge 한 변수이름
- ... : Merge하려는 변수들
- sep : Merge의 구분자(char, num, or symbol)

- extract : 데이터를 분리하는 폼을 지정하여 분리

---

data.table

C++ 기반의 **데이터 처리 패키지**

Basic

- dt[**i** , **j** , **by = 'colname', ...**]
  - dt : data.table 클래스를 가지는 데이터
  - i, j : i는 행을 선택하는 문법, j에는 열을 선택하는 문법
  - by : group_by의 대상이 되는 변수 이름의 list or vector

Index

- **data.table의 형식은 [행, 표현식, 옵션]** , data.frame은 [행, 열, 옵션]으로 되어있기 때문에 서로 조금씩 다른 인덱싱방법을 사용함
- 코드를 보자

Fast Grouping & Fast Selecting

data.table의 by에 의한 Grouping

- 코드를 보자

data.table의 J()에 의한 selecting

- **조건**으로 데이터 선택
  - 반드시 setkey를 이용하여 **KEY변수가 지정된 상태**여야함
  - **DT[J('제약조건')]**의 형식으로 작성함

처리 속도차이

- **data.frame보다 20배 빠름, pandas 보다 빠름**
- 그 이유는 key로 index 지정하고 연산하기 때문

조건을 이용한 데이터 선택

- data.table에서는 J표현식이 존재하며, **J를 사용하기 위해서는 key설정**이 반드시 필요
  - key는 행을 sort해주기 위한 기준 변수
- **setkeyv**의 경우는 key를 두개이상 설정할 수 있음
- **tables()**를 통해 key 지정된 것을 확인할 수 있음

Grouping 연산

- DT[, 연산식, by = 'variable']
- 여기서, **reserved keyword(종류가 엄청 많음)**는
  - **.N**
    - by에 의해 grouping이 될 땐 매칭된 변수의 행의 개수를 나타냄
    - 매칭이 되지 않는 다면 NA나 0으로 출력
  - **.SD**

    - Subset of x's Data for each group, excluding any columns used in by(or keyby)
    - 예를 들어, nrow(.SD)는 J와 by로 분류된 변수를 포함하는 행의 개수 단,(예를 들어) by로 나뉜 "male"과 "female"는 분리되어 계산
- 여기서, .SDcols는 .SD의 특정한 변수이름을 지정하여 해당 변수로만 이루어진 새 .SD를 구성함

Merge 연산

- **data.table merge연산이 빛을 보게 되는데 data.frame보다 400배 이상 속도를 보인다.**

Data 수정 및 삭제

- DT[i, 파생변수명 := '값']
- DT[i, `:=` (파생변수명 = '값')]

여러분들한테 당부의 말씀

- dplyr에서 알려준 함수 이외에 응용 함수가 되게 많습니다. 찾아서 공부하는 것 추천! (_all, _if, _each, ... etc)
- data-wrangling-cheatsheet 참고하는 것 추천**(R studio cheat sheets 모음)**
- dplyr 보다 data.table이 빠르지만 직관적이진 못합니다. 연습을 특히 많이 하시면서 찾아서 공부하는 것 추천합니다.
- 전처리가 60%이기 때문에 사람들 실력은 여기서 많이 갈립니다! 화이팅~

dtplyr : Data Table Back-End for dplyr

dplyr의 직관적인 풍부한 표현력 + data.table의 속도 반영

최근에 나온거니까 코드만 점검해봐요

Ref

- data.table cheat sheet

https://s3.amazonaws.com/assets.datacamp.com/img/blog/data+table+cheat+sheet.pdf

sql과 dplyr, tidyr
https://mrchypark.github.io/dabrp_classnote2/class3#1

RPubs
https://rpubs.com/bradleyboehmke/data_wrangling

Rdatatable/data.table
https://github.com/Rdatatable/data.table/wiki

R data.table
https://using.tistory.com/81

dtplyr: dplyr의 편리함과 data.table의 속도를 그대로!
http://henryquant.blogspot.com/2019/11/dtplyr-dplyr-datatable.html