M51-tidyr

LearningSpoonsR

2019-01-12

Part 0. Setup

Part I. `join`: 두 개의 데이터 프레임을 합하는 법 (a.k.a. `merge`)

Part II. Workding with “tidy” data

Part 0. Setup

```r
source("infile-tidyr.R")
library(tidyverse) # Wickham's
library(sqldf)
```

1. `source("infile-tidyr.R")`
   - 해당 R 소스코드를 실행한 효과가 나옴
   - 긴 코드를 보이지 않게 숨기게 하는데에 유용함
   - 이 강의노트에서 사용할 데이터프레임들을 정의하는 코드
2. `tidyverse`
   - Wickham이 만든 packages들을 다 모아놓은 패키지
3. `sqldf`
   - R에서 SQL 명령어를 사용할 수 있게 해주는 패키지
   - SQL은 대용량의 복잡한 데이터를 다루는 데에 적합한 언어
   - 이런 Cross-Language 패키지들은 새로운 환경에서의 연착률을 도와줌

Part I. `join`: 두 개의 데이터 프레임을 합하는 법 (a.k.a. `merge`)

0. df1과 df2를 어떻게 합해야 할까요?

```
df1
```

```
##   CustomerId Product
## 1          1 Toaster
## 2          2 Toaster
## 3          3 Toaster
## 4          4   Radio
## 5          5   Radio
```

```
df2
```

```
##   CustomerId State
## 1          2 Seoul
## 2          4 Seoul
## 3          6 Busan
```

- join에는 4가지 방법이 있습니다.

1. Inner Join

```
inner_join(df1, df2)
merge(x = df1, y = df2, by = "CustomerId")
sqldf("SELECT CustomerId, Product, State
      FROM df1 JOIN df2 USING(CustomerID)")
```

```
## Joining, by = "CustomerId"
##   CustomerId Product State
## 1          2 Toaster Seoul
## 2          4   Radio Seoul
```

2. Left Join

```
left_join(df1, df2)
merge(x = df1, y = df2, by = "CustomerId", all.x = TRUE)
sqldf("SELECT CustomerId, Product, State
       FROM df1 LEFT JOIN df2 USING(CustomerID)")
```

```
## Joining, by = "CustomerId"
##   CustomerId Product State
## 1          1 Toaster  <NA>
## 2          2 Toaster Seoul
## 3          3 Toaster  <NA>
## 4          4   Radio Seoul
## 5          5   Radio  <NA>
```

3. Outer Join (full)

```r
full_join(df1, df2)
merge(x = df1, y = df2, by = "CustomerId", all = TRUE)
```

```
## Joining, by = "CustomerId"
##   CustomerId Product State
## 1          1 Toaster  <NA>
## 2          2 Toaster Seoul
## 3          3 Toaster  <NA>
## 4          4   Radio Seoul
## 5          5   Radio  <NA>
## 6          6    <NA> Busan
```

4. Right Join

```r
right_join(df1, df2)
merge(x = df1, y = df2, by = "CustomerId", all.y = TRUE)
```

```
## Joining, by = "CustomerId"
##   CustomerId Product State
## 1          2 Toaster Seoul
## 2          4   Radio Seoul
## 3          6    <NA> Busan
```

Summary

- Summary

```
inner_join(df1, df2)
left_join(df1, df2)
full_join(df1, df2)
right_join(df1, df2)
```

- Variations (join할때 사용할 key변수를 구체화)

```
inner_join(df1, df2)
inner_join(x=df1, y=df2)
inner_join(x=df1, y=df2, by = "CustomerId")
inner_join(x=df1, y=df2, by = c("CustomerId"))
inner_join(x=df1, y=df2, by = c("CustomerId"="CustomerId"))
```

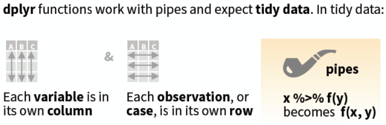
- `vlookup`이나 `index-match`함수를 이용해서 엑셀 파일 합해본 경험있으세요?
- R에서는 이게 정말 끝입니다.

blank

Part II. Workding with “tidy” data

0. 단정한 데이터?

- M21 p.17
- tidy data.frame!
    1. 개체 타입은 `data.frame`
    2. 각각의 row는 관찰값을 의미
    3. 각각의 column은 변수를 의미



dplyr functions work with pipes and expect **tidy data**. In tidy data:

Each **variable** is in its own **column** & Each **observation**, or **case**, is in its own **row**

pipes

x %>% f(y) becomes **f(x, y)**

Figure 1: from `dplyr` Cheatsheet

```
table1
```

```
##    ISO3 year  cases      popul
## 1   AFG 1999    745   19987071
## 2   AFG 2000   2666  201595360
## 3   BRA 1999  37737  172006362
## 4   BRA 2000  80488  174504898
## 5   CHN 1999 212258 1272915272
## 6   CHN 2000 213766 1280428583
```

- `table1`과 같은 정보를 담고 있지만, tidy하게 되있지 않은 데이터 구조가 있습니다.
- 이들을 tidy하게 `table1` 모양으로 바꿉니다.
- `pivot_table` in Excel

0. 목 적

- Before

```
table4a
```

```
##     ISO3    1999    2000
## 1    AFG     745    2666
## 2    BRA   37737   80488
## 3    CHN  212258  213766
```

```
table2
```

```
##     ISO3 year  type      count
## 1    AFG 1999 cases        745
## 2    AFG 1999 popul   19987071
## 3    AFG 2000 cases       2666
## 4    AFG 2000 popul  201595360
## 5    BRA 1999 cases      37737
## 6    BRA 1999 popul  172006362
```

```
table3
```

```
##     ISO3 year                rate
## 1    AFG 1999      745/19987071
## 2    AFG 2000     2666/201595360
## 3    BRA 1999    37737/172006362
## 4    BRA 2000    80488/174504898
## 5    CHN 1999 212258/1272915272
## 6    CHN 2000 213766/1280428583
```

- After

```
table1
```

```
##     ISO3 year   cases       popul
## 1    AFG 1999     745    19987071
## 2    AFG 2000    2666   201595360
## 3    BRA 1999   37737   172006362
## 4    BRA 2000   80488   174504898
## 5    CHN 1999  212258  1272915272
## 6    CHN 2000  213766  1280428583
```

1. Review (mutate)

```
table1
```

```
##   ISO3 year  cases      popul
## 1  AFG 1999    745   19987071
## 2  AFG 2000   2666  201595360
## 3  BRA 1999  37737  172006362
## 4  BRA 2000  80488  174504898
## 5  CHN 1999 212258 1272915272
## 6  CHN 2000 213766 1280428583
```

```
table1 %>% mutate(rate = cases / popul * 100)
```

```
##   ISO3 year  cases      popul        rate
## 1  AFG 1999    745   19987071 0.003727410
## 2  AFG 2000   2666  201595360 0.001322451
## 3  BRA 1999  37737  172006362 0.021939305
## 4  BRA 2000  80488  174504898 0.046123634
## 5  CHN 1999 212258 1272915272 0.016674951
## 6  CHN 2000 213766 1280428583 0.016694879
```

1. Review (group_by & summarise)

```
table1
```

```
##   ISO3 year  cases      popul
## 1  AFG 1999    745   19987071
## 2  AFG 2000   2666  201595360
## 3  BRA 1999  37737  172006362
## 4  BRA 2000  80488  174504898
## 5  CHN 1999 212258 1272915272
## 6  CHN 2000 213766 1280428583
```

```
table1 %>% group_by(year) %>% summarise(n = sum(cases))
table1 %>% count(year, wt = cases) # equivalent to above
```

```
## # A tibble: 2 x 2
##    year      n
##   <dbl>  <dbl>
## 1  1999 250740
## 2  2000 296920
```

2. gather from table4a & table4b

```
table4a
```

```
##   ISO3   1999   2000
## 1  AFG    745   2666
## 2  BRA  37737  80488
## 3  CHN 212258 213766
```

```
tidy4a <- table4a %>%
  gather(colnames(table4a)[-1],
         key = "year",
         value = "cases")
tidy4a
```

```
##   ISO3 year  cases
## 1  AFG 1999    745
## 2  BRA 1999  37737
## 3  CHN 1999 212258
## 4  AFG 2000   2666
## 5  BRA 2000  80488
## 6  CHN 2000 213766
```

```
table4b
```

```
##   ISO3       1999       2000
## 1  AFG   19987071  201595360
## 2  BRA  172006362  174504898
## 3  CHN 1272915272 1280428583
```

```
tidy4b <- table4b %>%
  gather(colnames(table4b)[-1],
         key = "year",
         value = "popul")
tidy4b
```

```
##   ISO3 year      popul
## 1  AFG 1999   19987071
## 2  BRA 1999  172006362
## 3  CHN 1999 1272915272
## 4  AFG 2000  201595360
## 5  BRA 2000  174504898
## 6  CHN 2000 1280428583
```

```r
left_join(tidy4a, tidy4b)
left_join(tidy4a, tidy4b, by = c("ISO3", "year"))
left_join(tidy4a, tidy4b, by = c("ISO3"="ISO3", "year"="year"))
```

```
## Joining, by = c("ISO3", "year")
##   ISO3 year  cases      popul
## 1  AFG 1999    745   19987071
## 2  BRA 1999  37737  172006362
## 3  CHN 1999 212258 1272915272
## 4  AFG 2000   2666  201595360
## 5  BRA 2000  80488  174504898
## 6  CHN 2000 213766 1280428583
```

3. spread from table2

```
table2
```

```
##   ISO3 year  type      count
## 1  AFG 1999 cases        745
## 2  AFG 1999 popul   19987071
## 3  AFG 2000 cases       2666
## 4  AFG 2000 popul 201595360
## 5  BRA 1999 cases      37737
## 6  BRA 1999 popul 172006362
```

```
table2 %>% spread(key = "type", value = "count")
```

```
##   ISO3 year cases     popul
## 1  AFG 1999   745  19987071
## 2  AFG 2000  2666 201595360
## 3  BRA 1999 37737 172006362
```

4. separate from table3

```
table3
```

```
##   ISO3 year              rate
## 1  AFG 1999      745/19987071
## 2  AFG 2000     2666/201595360
## 3  BRA 1999   37737/172006362
## 4  BRA 2000   80488/174504898
## 5  CHN 1999 212258/1272915272
## 6  CHN 2000 213766/1280428583
```

```
table3 %>% separate(rate, into = c("cases", "popul"), sep = "/")
```

```
##   ISO3 year  cases      popul
## 1  AFG 1999    745   19987071
## 2  AFG 2000   2666  201595360
## 3  BRA 1999  37737  172006362
## 4  BRA 2000  80488  174504898
## 5  CHN 1999 212258 1272915272
## 6  CHN 2000 213766 1280428583
```

참고: Classical method

```
table3$cases <-
  sapply(strsplit(table3$rate, split = "/"), function(x) x[1])
table3$popul <-
  sapply(strsplit(table3$rate, split = "/"), function(x) x[2])
```

Summary

- Before

```
table4a
```

```
##   ISO3   1999   2000
## 1  AFG    745   2666
## 2  BRA  37737  80488
## 3  CHN 212258 213766
```

```
table2
```

```
##   ISO3 year  type     count
## 1  AFG 1999 cases       745
## 2  AFG 1999 popul  19987071
## 3  AFG 2000 cases      2666
## 4  AFG 2000 popul 201595360
## 5  BRA 1999 cases     37737
## 6  BRA 1999 popul 172006362
```

```
table3
```

```
##   ISO3 year              rate  cases      popul
## 1  AFG 1999      745/19987071    745   19987071
## 2  AFG 2000    2666/201595360   2666  201595360
## 3  BRA 1999   37737/172006362  37737  172006362
## 4  BRA 2000   80488/174504898  80488  174504898
## 5  CHN 1999 212258/1272915272 212258 1272915272
## 6  CHN 2000 213766/1280428583 213766 1280428583
```

- After

```
table1
```

```
##   ISO3 year  cases      popul
## 1  AFG 1999    745   19987071
## 2  AFG 2000   2666  201595360
## 3  BRA 1999  37737  172006362
## 4  BRA 2000  80488  174504898
## 5  CHN 1999 212258 1272915272
## 6  CHN 2000 213766 1280428583
```