Lesson 6 Chapter 12 & 13

Hao Wang

3/9/2022

0. Load libraries

```
knitr::opts_chunk$set(echo = TRUE)

library(tidyverse)
library(rlang)

# Load NYC flight dataset
library(nycflights13)
```

$12.3 Pivoting

- pivot_longer(data, cols, names_to = "name", values_to = "value")

```
relig_income
```

```
## # A tibble: 18 x 11
##    religion ‘<$10k‘ ‘$10-20k‘ ‘$20-30k‘ ‘$30-40k‘ ‘$40-50k‘ ‘$50-75k‘ ‘$75-100k‘
##    <chr>      <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>      <dbl>
##  1 Agnostic      27        34        60        81        76       137        122
##  2 Atheist       12        27        37        52        35        70         73
##  3 Buddhist      27        21        30        34        33        58         62
##  4 Catholic     418       617       732       670       638      1116        949
##  5 Don’t k~      15        14        15        11        10        35         21
##  6 Evangel~     575       869      1064       982       881      1486        949
##  7 Hindu          1         9         7         9        11        34         47
##  8 Histori~     228       244       236       238       197       223        131
##  9 Jehovah~      20        27        24        24        21        30         15
## 10 Jewish        19        19        25        25        30        95         69
## 11 Mainlin~     289       495       619       655       651      1107        939
## 12 Mormon        29        40        48        51        56       112         85
## 13 Muslim         6         7         9        10         9        23         16
## 14 Orthodox      13        17        23        32        32        47         38
## 15 Other C~       9         7        11        13        13        14         18
## 16 Other F~      20        33        40        46        49        63         46
## 17 Other W~       5         2         3         4         2         7          3
## 18 Unaffil~     217       299       374       365       341       528        407
## # ... with 3 more variables: $100-150k <dbl>, >150k <dbl>,
## #   Don't know/refused <dbl>
```

```
relig_income %>%
  pivot_longer(cols = !religion, names_to = "income", values_to = "count")
```

```
## # A tibble: 180 x 3
##    religion income             count
##    <chr>    <chr>              <dbl>
##  1 Agnostic <$10k                 27
##  2 Agnostic $10-20k               34
##  3 Agnostic $20-30k               60
##  4 Agnostic $30-40k               81
##  5 Agnostic $40-50k               76
##  6 Agnostic $50-75k              137
##  7 Agnostic $75-100k             122
##  8 Agnostic $100-150k            109
##  9 Agnostic >150k                 84
## 10 Agnostic Don't know/refused    96
## # ... with 170 more rows
```

- pivot_wider(data, names_from, values_from)

```
fish_encounters
```

```
## # A tibble: 114 x 3
##    fish  station  seen
##    <fct> <fct>   <int>
##  1 4842  Release     1
##  2 4842  I80_1       1
##  3 4842  Lisbon      1
##  4 4842  Rstr        1
##  5 4842  Base_TD     1
##  6 4842  BCE         1
##  7 4842  BCW         1
##  8 4842  BCE2        1
##  9 4842  BCW2        1
## 10 4842  MAE         1
## # ... with 104 more rows
```

```
fish_encounters %>%
  pivot_wider(names_from = station, values_from = seen)
```

```
## # A tibble: 19 x 12
##    fish  Release I80_1 Lisbon  Rstr Base_TD   BCE   BCW  BCE2  BCW2   MAE   MAW
##    <fct>   <int> <int>  <int> <int>   <int> <int> <int> <int> <int> <int> <int>
##  1 4842        1     1      1     1       1     1     1     1     1     1     1
##  2 4843        1     1      1     1       1     1     1     1     1     1     1
##  3 4844        1     1      1     1       1     1     1     1     1     1     1
##  4 4845        1     1      1     1       1    NA    NA    NA    NA    NA    NA
##  5 4847        1     1      1    NA      NA    NA    NA    NA    NA    NA    NA
##  6 4848        1     1      1     1      NA    NA    NA    NA    NA    NA    NA
##  7 4849        1     1     NA    NA      NA    NA    NA    NA    NA    NA    NA
##  8 4850        1     1     NA     1       1     1     1    NA    NA    NA    NA
##  9 4851        1     1     NA    NA      NA    NA    NA    NA    NA    NA    NA
## 10 4854        1     1     NA    NA      NA    NA    NA    NA    NA    NA    NA
## 11 4855        1     1      1     1       1    NA    NA    NA    NA    NA    NA
## 12 4857        1     1      1     1       1     1     1     1     1    NA    NA
## 13 4858        1     1      1     1       1     1     1     1     1     1     1
## 14 4859        1     1      1     1       1    NA    NA    NA    NA    NA    NA
## 15 4861        1     1      1     1       1     1     1     1     1     1     1
## 16 4862        1     1      1     1       1     1     1     1     1    NA    NA
## 17 4863        1     1     NA    NA      NA    NA    NA    NA    NA    NA    NA
## 18 4864        1     1     NA    NA      NA    NA    NA    NA    NA    NA    NA
## 19 4865        1     1      1    NA      NA    NA    NA    NA    NA    NA    NA
```

- separate(data, col, into, sep)

```
df <- data.frame(x = c(NA, "x.y", "x.z", "y.z"))
df
```

```
##       x
## 1 <NA>
## 2  x.y
## 3  x.z
## 4  y.z
```

```
df %>% separate(x, c("A", "B"))
```

```
##       A    B
## 1 <NA> <NA>
## 2    x    y
## 3    x    z
## 4    y    z
```

```
# use regular expression in separator.
df %>% separate(x, c("A", "B"), sep = '\\.')
```

```
##       A    B
## 1 <NA> <NA>
## 2    x    y
## 3    x    z
## 4    y    z
```

- unite()

```
df_sep <- df %>% separate(x, c("A", "B"))

# Different separators.
df_sep %>% unite(x, A, B)
```

```
##       x
## 1 NA_NA
## 2   x_y
## 3   x_z
## 4   y_z
```

```
# don't need to use regular express for the separator
df_sep %>% unite(x, A, B, sep = ".")
```

```
##       x
## 1 NA.NA
## 2   x.y
## 3   x.z
## 4   y.z
```

$13.4 Mutating joins

Compare the joins in dplyr and SQL.

dplyr	SQL
inner_join(x, y, by = "z")	SELECT * FROM x INNER JOIN y USING (z)
left_join(x, y, by = "z")	SELECT * FROM x LEFT OUTER JOIN y USING (z)
right_join(x, y, by = "z")	SELECT * FROM x RIGHT OUTER JOIN y USING (z)
full_join(x, y, by = "z")	SELECT * FROM x FULL OUTER JOIN y USING (z)

$13.7 Set operations

- intersect(x, y): return only obs. in both x and y.
- union(x, y): return unique obs. in x and y.
- setdiff(x, y): return obs. in x, but not in y.

Exercise:

Compare the sheet **PBI New** and **SAS New** of Excel file: **Test PBI and SAS new and endorsement 20220214.xlsx**

- Policy number is the primary key for each sheet. Find out the difference records between the two sheets.
- use join function and setdiff function to see if you can get the same results.