

Week 4 DS Recitation: Tidy Data

SIBDS 2024 @ Columbia

17 June, 2024

Getting Started

Tasks:

1. Create a new R project and named it `week4_DS_recitation`
2. Put the `week4_DS_recitation_tidy_data.Rmd` file into the same folder of the R project you just created.

Pivot longer

Tasks:

1. The `billboard` dataset in `tidyr` package records the billboard rank of songs in the year 2000. Reshape the dataset by changing the `wk1` to `wk76` to a variable called `week`, and the values to a variable called `rank`. Also try to use `values_drop_na` to drop rows that correspond to missing values (Not every song stays in the charts for all 76 weeks).

```
# Your answer starts here
```

2. Redo the above reshaping process, convert the `week` variable to an integer this time.

```
# Your answer starts here
```

Pivot wider

Tasks:

1. The following `warpbreaks` dataset contains the warp break experiment results with nine replicates for every combination of `wool` (A and B) and `tension` (L, M, H):

Try to run the following code, what happens if we attempt to pivot the levels of `wool` into the columns?

```
warpbreaks %>%  
  pivot_wider(  
    names_from = wool,  
    values_from = breaks  
  )
```

Try to change the default setting of `values_fn` to get a wider dataframe summarizing the `mean` of those 9 experiment results (mean `breaks` for each combination of `wool` and `tension`).

```
# Try to modify this code
warpbreaks %>%
  pivot_wider(
    names_from = wool,
    values_from = breaks
  )
```

2. The `us_rent_income` dataset contains information about median income and rent for each state in the US for 2017. Here both `estimate` and `moe` ($1.645 \times \text{SE}$) are values columns. Try to provide a summary of the data for each state in a single row, outlining both income and rent estimates for each variable.

```
# Your answer starts here
```

Longer then wider

Tasks:

1. The `world_bank_pop` dataset contains data from the World Bank about population per country from 2000 to 2018.

Try to tidy the data, you can follow these steps:

- Firstly, `year` is spread across multiple columns, we can use `pivot_longer` to put them into a single column
- Next, focus on the `indicator`. Here `SP.POP.GROW` is population growth, `SP.POP.TOTL` is total population, and `SP.URB.*` are the same but only for urban areas. Let's split this up into two variables: `area` (total or urban) and the actual `variable` (population or growth), you may need this : `separate(indicator, c(NA, "area", "variable"))`
- Finally, we can complete the tidying by pivoting (using `pivot_wider`) `variable` and `value` to make `TOTL` and `GROW` columns

```
# Your answer starts here
```

Joining datasets

Tasks:

1. Install and load the `nycflights13` dataset use the following code.

```
if (!requireNamespace("nycflights13", quietly = TRUE)) {
  install.packages("nycflights13")
}

library(nycflights13)
```

2. Join the `weather` dataset to the `flights2` dataset (created by the following code) using `left_join`. Which variables are used in this joining process?

```
data(weather)
data(flights)
```

```
flights2 <- flights %>% select(year:day, hour, origin, dest, tailnum, carrier)
# Your answer starts here
```

3. Join the `airports` dataset to the `flights2` dataset using `left_join`. The variable we want to use in the `airports` dataset is `faa`. Noticing that each flight has an origin (`origin`) and destination (`dest`) airport, so we need to specify which one we want to join to.

```
data(airports)
```

```
# Your answer starts here
```