Data Wrangling for EDSDers

Module outline

13-16 Nov, 2023

Tim Riffe
Universidad del País Vasco & Ikerbasque (Basque Foundation for Science)

13 Nov, 2023

Aim

The aim of this module is to show a variety of data wrangling operations in a variety of source and target data situations.

A day will consist in interactive demonstration lasting around 3-4 hours from 9:00-12:00 or so. This will include worked examples to demonstrate concepts as well as problems to solve individually or in groups. Troubleshooting will take place throughout. Participants should have the most recent versions of `Rstudio` and `R` installed.

Schedule

This index is to be updated as materials are finalized. It's more for posterity than a prospective syllabus.

Session 1 (Monday, 13 Nov)

In this session we cover the basics of Rmarkdown, and we practice `tidyverse` data handling functions on the `gapminder` dataset, including `mutate()`, `summarize()`, and `group_by()`. We also get in some good `ggplot2` practice.

Session 2 (Tuesday, 14 Nov)

Lots more practice with `group_by()`, `mutate()`, `summarize()`. We also see how to read in and tabulate from fixed-width files `read_fwf()` (USA natality file), and we make semi-sophisticated `ggplot()` figures from it.

Session 3 (Wednesday, 15 Nov)

We cover data reshaping on the example of panel data (student request). First we practice functions such as `pivot_longer()` and `pivot_wider()` on the `gapminder` data, then we move to a more complex HRS data example. We end up calculating transition probabilities for a multistate health model, then smooth them, then we compare mortality estimates by health status with the HMD pattern.

[Session 4] (Thursday, 16 Nov) In this session we improvised code for deterministic data merging, there is not yet a handout for this. We harmonized and merged health prevalence data from Eurostat with population counts from the World Population Prospects. The lesson is that most of the work of merging is in the harmonization stage. The merge part is straightforward, just figure out if you need a `left_join()`, `right_join()`, `inner_join()`, `full_join()`, `anti_join()`, or `cross_join` :-)