Combining Data Sets

ECON 490

Taylor Mackay || Email: tmackay@fullerton.edu

Slides Overview

In these slides, we'll discuss:

- Combining or merging data sets
- How to prepare your data for merging and `join()` functions in R

Combining Data Sets

Why merge or combine data?

- For your projects (and most data analysis), you'll get data from ***multiple*** sources
- You combine data sets by ***merging or joining*** them together

Examples include combining:

- State-level data with individual-level survey responses
- Location-based crime data with local area demographic data
- NBA player performance data with salary and contract records

The Shape of Data Sets

When you combine two data sets, the result is ***wider*** data

- You started with some set of columns / variables, then added new columns
- Different variables across data sets

Sometimes, you have data that you want to ***stack or append***

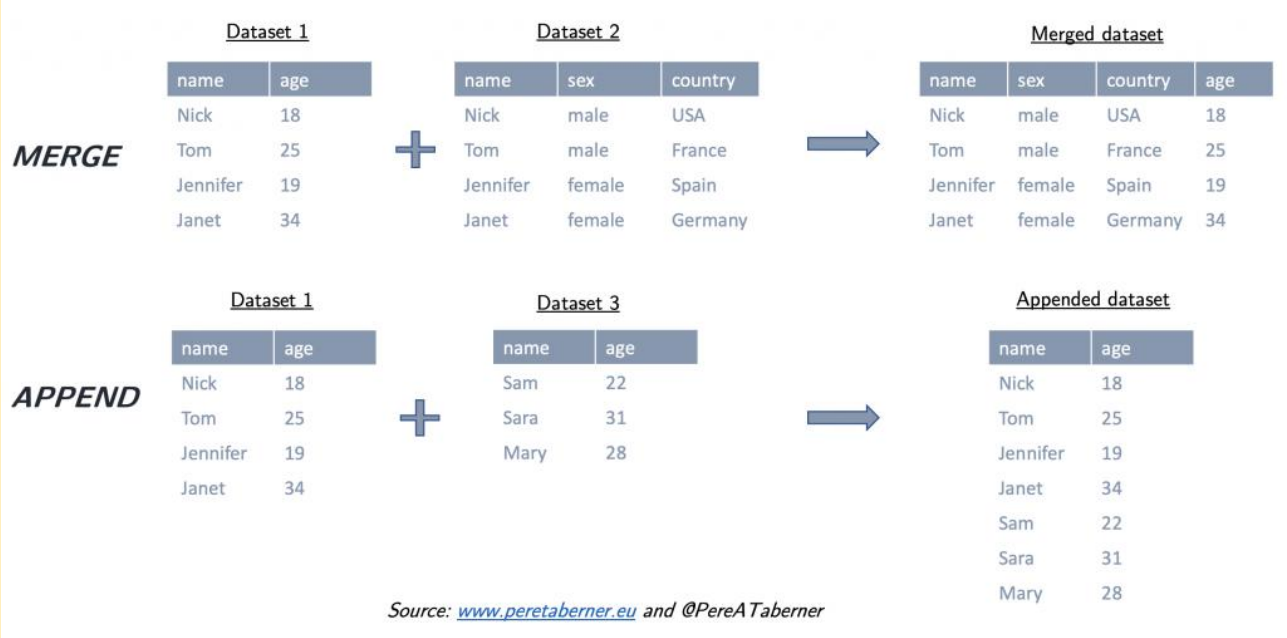
- Example: county-level pollution data for CA in 2020 and 2021
- ***Same*** variables in ***both*** data sets → append data to get panel with both years

Appending data results in ***longer*** data

- Depending on context, merges might also, but main goal = new columns

Appending Data in R

In screenshot below, use `bind_rows()` function to ***append*** data sets 1 & 3



Before You Combine Data

Before you combine data sets, you need to understand ***data structure***

- Within both data sets, what variables uniquely identify each row?

Across both data sets, what variables will you use to link data?

- These are your “key” or “by” variables
- What is the shared structure of both data sets? Defined by ***less*** granular data

Example: Combining individual-level CPS with state-level home price data

- While CPS has individual identifiers, the ***shared structure*** is state and year, so these are the key or by variables (state-level data is less granular)

Preparing Data for Merging

Once you've identified your key variables, make sure they're ***consistent***

- Across both data sets, is each variable ***same format*** (i.e., factor, string, etc.?)
- Are spellings consistent? I.e., check "California" vs. "california" vs. "CA"

If necessary, create new, consistent versions of key variables using `mutate()`

Check you don't have duplicates across key variables

- E.g., in state-by-year ACS data, suppose you found 2 Arizona in 2010 rows
- Drop extra row using `filter()` or combine via `group_by() + summarize()`

Join Functions in R

Given two data sets, you'll have one of the following types of merges or joins:

- ***1-to-1:*** Each observation in 1st dataset matches exactly one row in 2nd
- ***Many-to-1:*** Multiple observations in 1st dataset match to same row in 2nd
- ***Many-to-Many:*** Multiple rows match to multiple rows → avoid doing this!

Most useful join functions in R for your projects:

- `inner_join()`: Keep ***only*** matched rows in ***both*** data sets
- `left_join()`: Keep all rows from 1st data set and all matches from 2nd
- `full_join()`: Keep all rows from ***both*** data sets

After Merging Data

The most important thing to do is “sanity check” your new merged data set

- Check the number of rows in the matched data set using `nrows()`
- Does it match what you expected?

Check NAs and summary statistics for important variables using `summary()`

Example: Combine state-level crime data with state-level ACS via `inner_join()`

- This keeps matched rows, so you should have rows for 51 states (counting DC) multiplied by 5 years; if you don’t, something went wrong!

Join Examples

Inputs

DF1

ID	Value
A	123
B	769
C	475
D	978

DF2

ID	Level
A	Red
B	Blue
E	Green
F	Yellow

Join Functions

`inner_join(DF1, DF2)`

`left_join(DF1, DF2)`

Outputs

ID	Value	Level
A	123	Red
B	769	Blue

ID	Value	Level
A	123	Red
B	769	Blue
C	475	*NA*
D	978	*NA*