Data Wrangling 2: Reshape and Combine Tables

Haohan Chen

Last update: October 05, 2023

Objective

This lecture continues our introduction to data wrangling with R. Using the *V-Dem* data as an example, we will learn how to *reshape* and *merge* datasets using a set of `tidyverse` functionality. Specifically, we will focus on functions...

1. ... to *reshape* a table (long <-> wide) with `pivot_longer` and `pivot_wider`
2. ... to *stack* tables by row or by column with `bind_rows` and `bind_cols` (or, alternatively, `cbind` and `rbind`)
3. ... to *merge* two tables with `inner_join`, `full_join`, `left_join`, `right_join`, `semi_join`, and `anti_join`

Further Reading

- R for Data Science (2e) Chapters 6 and 20: https://r4ds.hadley.nz/
- `dplyr` cheatsheet (the "Combine Tables" section on p. 2)
- `tidyr` cheatsheet (the "Reshape Data" section on p. 1): https://rstudio.github.io/cheatsheets/html/tidyr.html

Outline of In-Class Demo

For this in-class demonstration, we will continue working on the external parts of the V-Dem data from 1984 to 2022. The data are located here: `_DataPublic_/vdem/1984_2022/vdem_1984_2022_external`

1. Reshape the V-Dem dataset
    1. `pivot_longer`: Make it a long table where each variable gets its own row. That is, a row in the new dataset is a *country-year-observation*.
    2. `pivot_wider`: Widen the above long table so that each *Year* has its own column.
2. Stack multiple subsets of the V-Dem datasets by row and by columns
    1. `bind_cols`: Merge the following two subsets of the V-Dem data: `_DataPublic_/vdem/1984_2022/vdem_1984_2022` and `_DataPublic_/vdem/1984_2022/vdem_1984_2022_index`
    2. `bind_rows`: Merge the following two subsets of the V-Dem data: `_DataPublic_/vdem/1984_2022/vdem_1984_2022` and `_DataPublic_/vdem/1945_1983/vdem_1945_1983_external`
3. Join multiple regional subsets of the V-Dem datasets
    1. Make a new data frame that contains the following variables: `country_name`, `year`, `e_regionpol_6C`, `e_fh_status`, `e_gdppc`, and `e_gdp`

2. Create two separate subsets of the above data frames. Each subset include a subset of countries/ regions that are within the *region* (defined by `e_regiongeo` and `e_regionpol_6C` respectively) where *China* is located.
3. Explore the behavior of `inner_join`, `full_join`, `left_join`, `right_join`, `semi_join`, and `anti_join` with the two data frames.

4. Validate V-dem's GDP data with World Bank data

In-Class Demo

```
library(tidyverse)
```

1. Reshape the V-Dem dataset

```
# INSERT CODE
```

2. Stack multiple subsets of the V-Dem datasets

```
# INSERT CODE
```

3. Join multiple regional subsets of the V-Dem datasets

```
# INSERT CODE
```

4. Validate the GDP data in V-Dem with World Bank data

Task: There are many different "versions" of GDP data. I wonder whether the GDP data in the V-Dem dataset is reliable. So I would like to validate it with data from the World Bank.

Download World Bank Data We will start the adventure by downloading World Bank data.

```
# Install the WDI package that helps fetch data from the World Bank dataset
# See: https://github.com/vincentarelbundock/WDI

# install.packages("WDI")

# Note: Comment out the above "install.packages" command after you are done with installing the package
```

```
library(WDI)
```

```
# Search for GDP related data
wb_gdpdata_list <- WDIsearch("gdp")

str(wb_gdpdata_list)
```

```
## 'data.frame':    540 obs. of  2 variables:
##  $ indicator: chr  "5.51.01.10.gdp" "6.0.GDP_current" "6.0.GDP_growth" "6.0.GDP_usd" ...
##  $ name     : chr  "Per capita GDP growth" "GDP (current $)" "GDP growth (annual %)" "GDP (constant 2
```

```r
# Narrow down to indicators of GDP (I have done some pre-screening)
wb_gdpdata_list_s <- wb_gdpdata_list |> filter(str_detect(indicator, "^NY\\.GDP"))
```

```r
# Download GDP-related data
wb_gdpdata <- WDI(
  indicator = c("NY.GDP.MKTP.PP.KD", "NY.GDP.PCAP.PP.KD"),
  country = "all",
  start = 1984, end = 2022)
```

```r
# Remove the intermediate data we no longer need.
rm(wb_gdpdata_list, wb_gdpdata_list_s)
```

To match two datasets from two different sources, we should always check whether the "identifiers" are consistent. In our case, are names of countries specified in the same way in the V-Dem and the World Bank dataset?

```r
# Check the specification of country names.
```

Find Country Identifiers When it comes to matching countries, country codes are usually more reliable. The problem is that we do not have country codes in the V-Dem data. An R package named `countrycode` can help.

```r
# install.packages("countrycode")
# See how you may use the package: https://github.com/vincentarelbundock/countrycode
```

```r
# INSERT CODE: Use countrycode to make country code indicators
```

Join and Compare Now that we have cleaned the World Bank data, our final step is to join it with the World Bank data and compare the GDP and GDP per capita indicators from the two sources.

```r
# INSERT CODE: Join the two datasets
```

```r
# INSERT CODE: Compare the two datasets
```