

# SPEC Lab REU R Resources: Data Management III- Merging & Reshaping Data

Alix Ziff, Benjamin A.T. Graham, and Jasmine Chu

January 2023

In this assignment, you will continue to work with the IDC powersharing data. In our last assignment, we used data that was already in country-year format. The goal for this assignment is to walk through some of the steps we used to get from individual coder sheets which each covered a single country, to a merged country-year dataset.

You will need the following files: 1. LATVIA\_NJ\_fortraining.csv 2. FINLAND\_JC\_fortraining.csv

**Preparation:** Write a detailed header for your new R script, clear old objects out of your environment, set your working directory, load the appropriate packages (e.g. tidyverse), and then load each spreadsheet.

```
# In this assignment, I will be working with the tidy and dplyr packages, both  
# from tidyverse. Throughout this group work, we will learn how to tidy up  
# the data by going from individual coder sheets to the finalized merged country  
# year dataset
```

```
rm(list=ls()) #clear everything  
library(tidyverse) #load the correct library
```

```
# setwd("/Users/JasmineChu/Desktop") # your filepath may vary  
# there are actually two methods when reading csv files. The first method is the  
# read_csv() function, which loads data in as a tibble and allows nonstandard  
# variable names. Essentially, this makes our lives easier in our instance.
```

```
##### using read_csv() #####  
latvia <- read_csv("LATVIA_NJ_fortraining.csv") #load the first dataset  
finland <- read_csv("FINLAND_JC_fortraining.csv") # load the second spreadsheet
```

```
# Now, the second method is to use the read.csv() function, which loads the data in  
# as a dataframe and thus, does not allow variable names that have multiple  
# words, capital letters, or that start with numbers.
```

```
##### using read.csv() #####  
latvia <- read.csv("LATVIA_NJ_fortraining.csv") #load the first dataset  
finland <- read.csv("FINLAND_JC_fortraining.csv") # load the second spreadsheet
```

```
# Notice how when we use the read.csv() function, we see capital Xs  
# stick out on the front of the column names. When we tidy up our data in the  
# next exercise, we want to make sure to get rid of these Xs, which can be  
# shown in the next few steps.
```

*Exercise 1:* Pivot the data in each spreadsheet to make it tidy – i.e. variables as columns and years as rows.

*Hint:* You will need to create a variable to identify the country.

```

# reshape and clean Finland data
finland <- mutate_at(finland, 2:11, as.numeric)%>% #make values numeric
  filter(!COUNTRY=="") %>% #filter out any empty values
  pivot_longer(names_to = "year",
               values_to = "value",
               c(X2010:X2019))%>% #pivot long (creates year variable)
  pivot_wider (names_from = COUNTRY,
              values_from = value, id_cols = year)%>% #variables as columns
  separate(col = year, into = c(NA,"year"), sep = "X") %>% #rename year variable
  # and get rid of the leading X's in front of the years
  mutate(year = as.numeric(year))%>%
  mutate("country" = "Finland")

# next, reshape and clean Latvia data
latvia <- mutate_at(latvia, 2:11, as.numeric)%>% #make values numeric
  pivot_longer(names_to = "year",
               values_to = "value",
               c(X2010:X2019))%>% #pivot long (creates year variable)
  pivot_wider (names_from = COUNTRY,
              values_from = value, id_cols = year) %>% #variables as columns
  separate(col = year, into = c(NA,"year"), sep = "X") %>% # #rename year variable
  # and get rid of the leading X's in front of the years
  mutate(year = as.numeric(year))%>%
  mutate("country"="Latvia")

```

*Exercise 2:* Use `as.numeric` to make the values that should be numeric (like years) actually numeric.

```

finland<- finland %>%
  mutate_at(vars(year:jrevman), as.numeric)

latvia<- latvia %>%
  mutate_at(vars(year:jrevman), as.numeric)

```

*Exercise 3:* Merge these two spreadsheets together.

```

full_data <- full_join(latvia, finland) #merge spreadsheets into one

## Joining, by = c("year", "constsus", "treaty", "martiallaw_binary", "gcman",
## "gchimp", "unity", "partynoethnic", "mveto", "resman", "resimp", "resseats",
## "resseatsimp", "reestablish", "relrestrict", "relconstp", "relconstd",
## "stconst", "state", "muni", "subtax", "subed", "subpolice", "fedunits",
## "auton", "milleg", "miman", "milvote", "milparty", "jtenure", "jcause",
## "japptbr", "jconst", "jrevman", "country")

```

*Exercise 4:* Identify a subset of six variables you find most interesting and rename those variables to simple, descriptive names. Remember to keep everything () else! We don't want to subset the data in this exercise. The IDC powersharing codebook is in the Training Data folder.

```

full_data <- full_data%>% #use the piping command to rename multiple variables
  rename(c("police" = "subpolice"))%>% #first variable
  rename(c("rel_est" = "reestablish"))%>% #second variable
  rename(c("seats" = "resseats"))%>% #third variable
  rename(c("martiallaw" = "martiallaw_binary"))%>% #fourth variable
  rename(c("religion" = "relrestrict"))%>% #fifth variable
  rename(c("party" = "partynoethnic")) #sixth variable

```

*Exercise 5:* Save your merged dataset as a .rds file in your homework submission folder. Also save a copy of your R script in there. Don't forget to annotate the heck out of your R script for future you!

```
saveRDS(full_data, file = "DM3GroupWork_JASMINECHU.rds") #save as rds file
```