SPEC Lab R Resources: Reshaping Data To Make it Tidy – Homework

Jasmine Chu, Claudia Salas Gimenez, Anna Baidar and Ben Graham

Spring 2025

Welcome to the last assignment of Module 3: Data Management II. This homework assignment is designed to help you apply the data tidying and merging skills you've developed through the walkthroughs and group exercises. The goal is to further practice the content covered in this module and prepare you for handling datasets effectively in social science research. By working with real-world data from the World Bank Ed Stats and World Development Indicators databases, you'll gain hands-on experience transforming untidy data into analysis-ready formats, merging datasets, and assigning meaningful variable names.

Save your responses in your personal subfolder in the `412_413 shared` Google Drive folder. The R script should be titled `HW_DM2_[YOUR INITIALS]`. You can also save a copy of your R script to your own computer for future reference.

Initial Setup

Before starting, ensure you've set up your working directory and installed the necessary libraries. Also, download the `world_bank_education.csv` dataset from the World Bank Ed Stats datasets, which contains Barro and Lee's measures for estimates for the population between 15 and 19 year-old that have completed secondary school.

To gain a deeper understanding of the datasets and their variables, refer to the World Development Indicators data catalog to better understand the datasets and their variables. Keep in mind that since these datasets have been modified for this walkthrough, the variable names may differ slightly from those in the original codebooks. Barro-Lee website to view the list of variables, countries, data sources and brief description of the methodology, as well as access the latest updated version of the Barro-Lee dataset.

```
# Set working directory
#setwd("YourFolderPath")

# Load packages
library(tidyverse)
library(dplyr)
library(ggplot2)
library(ggrepel)

# Load the data
dt <- read_csv("world_bank_data_education.csv")
```

The goal of this assignment is to tidy the `world_bank_data_literacy_rates.csv` data into a country-year format.

Exercise 1: Reshaping Data

Exercise 1.1: Rename Variables

Before reshaping the dataset, start by renaming the columns referring to the years 1970 through 2020 to their respective year numbers.

```
# Rename the columns that stand for the years between 1970 and 2020
names(dt)[5:55] <- c(1970:2020)
```

Exercise 1.2: Reshape and Clean the Data

Next, transform the dataset into a tidy format where each row represents a unique combination of country and year.

Helpful Hint: Remember to use backticks (`` ` ` ``) around variable names that contain spaces to ensure they are properly recognized in your code.

```
# Tidy and clean the World Bank data
dt1 <- dt %>%
  mutate(across(`1970`:`2020`, as.numeric)) %>%
  ## Convert columns 1970:2020 to numeric
  rename(country = `Country Name`) %>%
  ## Rename 'Country Name'
  select(-`Series Code`) %>%
  ## Take out 'Series Code' variable
  pivot_longer(cols = `1970`:`2020`,
               names_to = "year",
               values_to = "value") %>%
               ## Transform data to long format
  mutate(year = as.numeric(year)) %>%
  ## Convert column 'year' to numeric
  pivot_wider(names_from = Series,
              values_from = value,
              id_cols = c(year, country))
              ## Spread column 'Series' into separate columns, with each column
              ## representing a different educational statistic
```

Exercise 1.3: Rename the New Variables

Lastly, assign concise, descriptive names to the variables.

```
# Option 1: rename variables using the rename() function in dplyr package
dt1 <- dt1 %>%
  rename(
    "secondary_total_BL" =
      "Barro-Lee: Average years of secondary schooling, age 15-19, total",
    "secondary_female_BL" =
      "Barro-Lee: Average years of secondary schooling, age 15-19, female")

# Option 2: rename variables using base R by specifying the column number
names(dt1)[3] <- "secondary_total_BL"
names(dt1)[4] <- "secondary_female_BL"
```

Exercise 2: Include Additional Variables

Now that you've reshaped and cleaned educational data, let's dive deeper by adding more dimensions to the analysis. In this exercise, we'll expand the dataset by including data on literacy rates for population between 15 and 24 from the World Development Indicators.

Exercise 2.1: Load and Rename Variables

Let's start by loading the `world_bank_data_literacy_rates.csv` data from the Training Data Spring 2024 folder, and rename the columns for the years 1970 through 2020 to their respective year numbers.

```
# Load the data
df <- read_csv("world_bank_data_literacy_rates.csv")

# Rename the columns that stand for the years between 1960 and 2020
names(df)[5:55] <- c(1970:2020)
```

Exercise 2.2: Reshape and Clean the Data

As done with the `world_bank_education.csv` dataset, transform the literacy rate dataset into tidy format, where each row represents a unique country-year combination.

Helpful Hint: Remember that if variable names consist of multiple words separated by spaces, you'll need to enclose them in backticks (` `) to ensure they are recognized correctly in your code.

```
# Tidy and clean the World Bank data
df1 <- df %>%
  mutate(across(`1970`:`2020`, as.numeric)) %>%
  ## Convert columns 1970:2020 to numeric
  rename(country = `Country Name`) %>%
  ## Rename 'Country Name'
  select(-`Series Code`) %>%
  ## Take out 'Series Code' variable
  pivot_longer(cols = `1970`:`2020`,
               names_to = "year",
               values_to = "value") %>%
  ## Transform data to long format
  mutate(year = as.numeric(year)) %>%
  ## Convert column 'year' to numeric
  pivot_wider(names_from = `Series Name`,
              values_from = value,
              id_cols = c(year, country))
              ## Spread column 'Series' into separate columns, with each column
              ## representing a different educational statistic
```

Exercise 2.3: Rename the New Variables

Also, provide clear and concise names for literacy rate-related variables.

```
# Option 1: rename variables using the 'rename()' function in dplyr package
df1 <- df1 %>%
  rename(
    "literacy_rate_total" = "Literacy rate, youth total (% of people ages 15-24)")

# Option 2: rename variables using base R by specifying the column number
names(df1)[3] <- "literacy_rate_total"
```

Exercise 3: Merging Datasets

Now that both datasets are clean and tidy, it's time to combine both datasets. Merge the education and literacy data, keeping all available data from both datasets.

Note: The standard practice in SPEC Lab is to merge datasets using Gleditsch-Ward Country Codes and year for precise matching. However, for this assignment, since both datasets originate from the World Bank, merging by country name and year is sufficient.

```
# Merge datasets by 'country' and 'year'
merged <- full_join(dt1, df1, by = c("year", "country"))
```

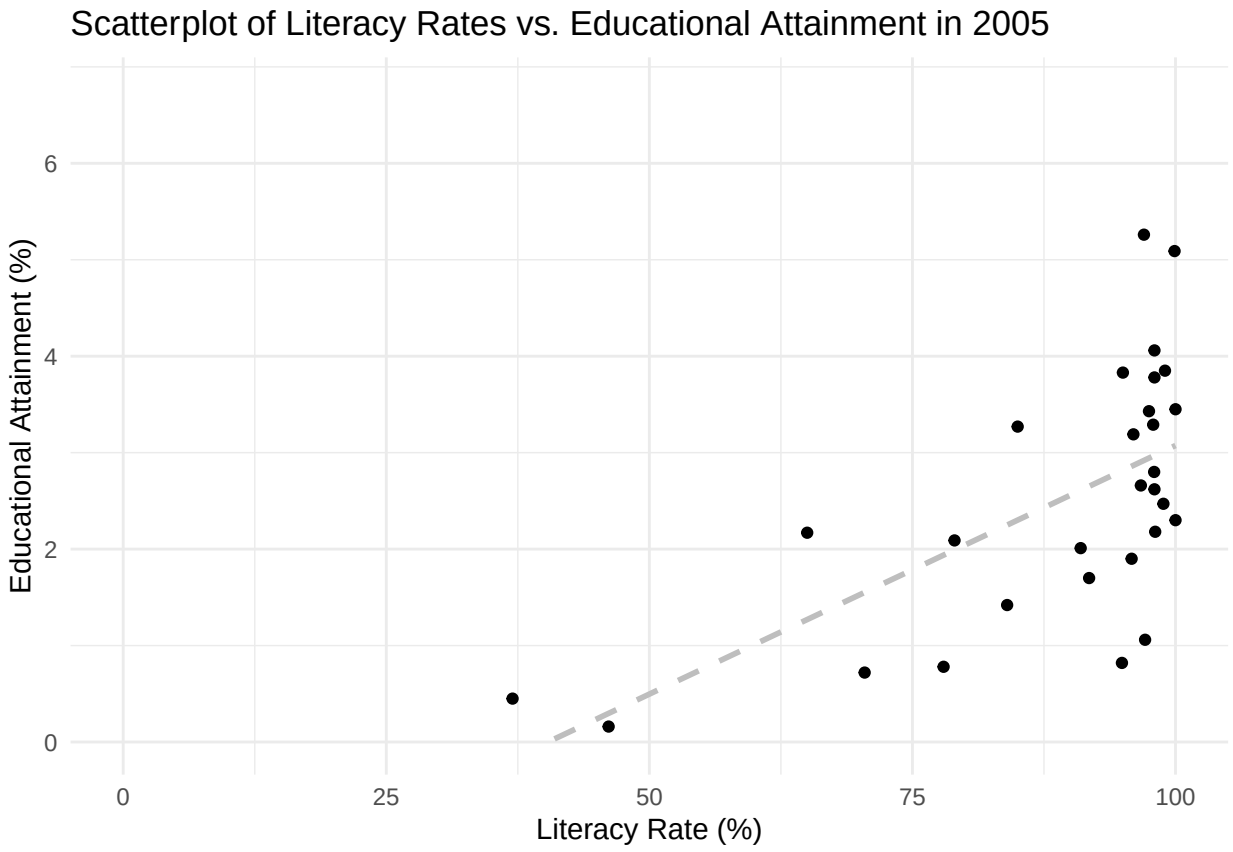
Exercise 4: Visualizing the Relationship Between Literacy Rates and Educational Attainment

Use the merged dataset to create a scatterplot of the relationship between the literacy rates and educational attainment in a single year.

```
# For this question, I have chosen the year 2005, so the answer key might look
# slightly different if you select other years

# Select data for 2005
merged_2005 <- merged %>%
  filter(year == 2005)

# Scatterplot of literacy rates vs. educational attainment by year
ggplot(merged_2005,
       aes(x = literacy_rate_total, y = secondary_total_BL)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE,
              color = "grey", linetype = "dashed") +
  labs(
    title = "Scatterplot of Literacy Rates vs. Educational Attainment in 2005",
    x = "Literacy Rate (%)",
    y = "Educational Attainment (%)") +
  scale_y_continuous(
    limits = c(0, NA)
  ) +
  scale_x_continuous(
    limits = c(0, NA)
  ) +
  theme_minimal()
```



Bonus Exercise 1

As bonus question, plot the over time trends in the global average of literacy rates and educational attainment.

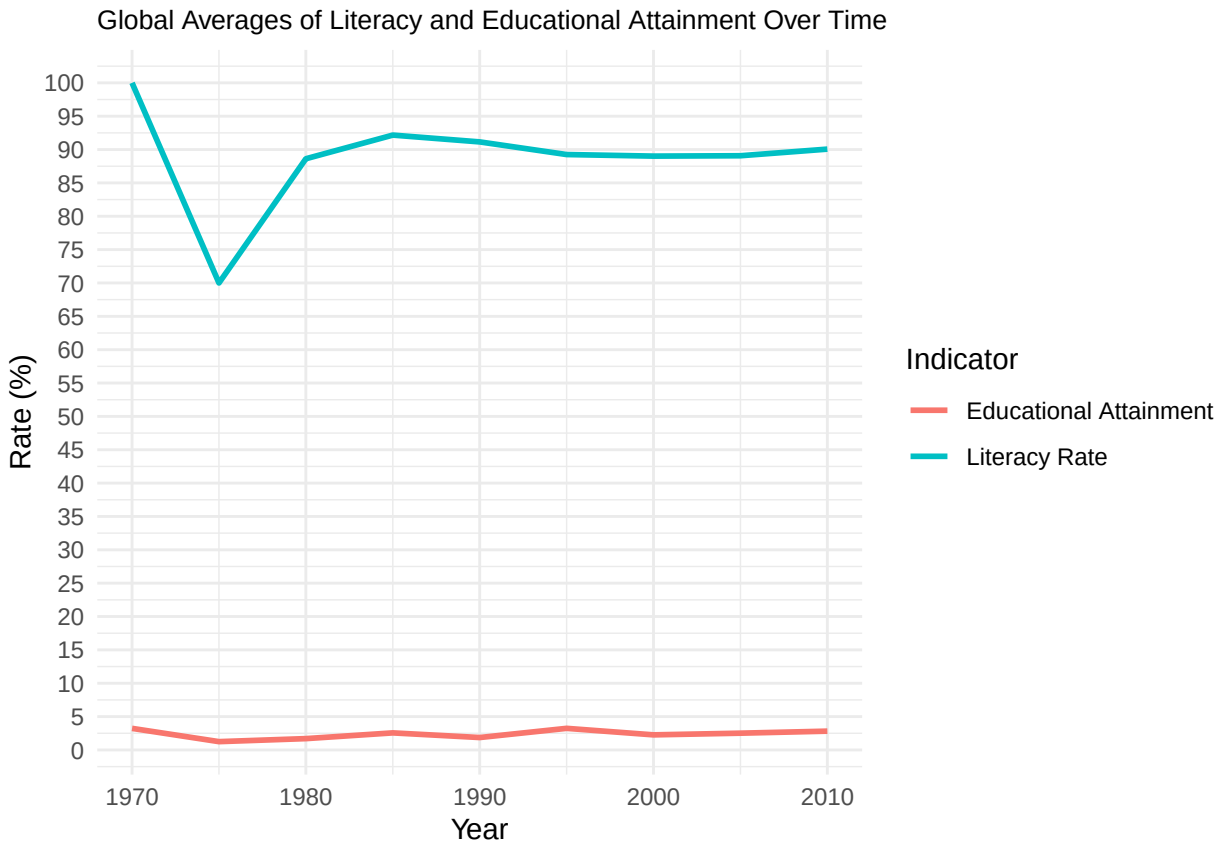
```
# First, calculate the global averages for all countries
global_avg <- merged %>%
  mutate(
    year = as.numeric(year),
    literacy_rate_total = as.numeric(literacy_rate_total),
    secondary_total_BL = as.numeric(secondary_total_BL)
  ) %>%
  filter(
    !is.na(year),
    !is.na(literacy_rate_total),
    !is.na(secondary_total_BL)
  ) %>%
  group_by(year) %>%
  summarize(
    global_avg_literacy = mean(literacy_rate_total, na.rm = TRUE),
    global_avg_edu_attain = mean(secondary_total_BL, na.rm = TRUE)
  )

# Reshape into long format
global_avg_long <- global_avg %>%
  pivot_longer(
    cols = c("global_avg_literacy", "global_avg_edu_attain"),
```

```
    names_to = "variable",
    values_to = "value"
  )

# Plot both lines on the same y-axis
## Determine maximum value for setting y-axis breaks
max_val <- max(global_avg_long$value, na.rm = TRUE)

ggplot(global_avg_long, aes(x = year, y = value, color = variable)) +
  geom_line(size = 1) +
  labs(
    title = "Global Averages of Literacy and Educational Attainment Over Time",
    x = "Year",
    y = "Rate (%)"
  ) +
  scale_color_discrete(
    name = "Indicator",
    labels = c(
      "global_avg_edu_attain" = "Educational Attainment",
      "global_avg_literacy"   = "Literacy Rate"
    )
  ) +
  scale_y_continuous(
    breaks = seq(0, ceiling(max_val), 5)    # Ticks every 5 units up to your max
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 10)
  )
```



Bonus Exercise 2

Although in Bonus Exercise 1 we plotted overall trends in literacy rates and educational attainment, our final dataset contains many missing values. This means the patterns we observe might primarily reflect only those countries that did report literacy rates and educational attainment. In this exercise, plot the over-time trends in the global average of literacy rates and educational attainment for countries with non-missing values. What types of countries are most likely to report their literacy rate and educational attainment data?

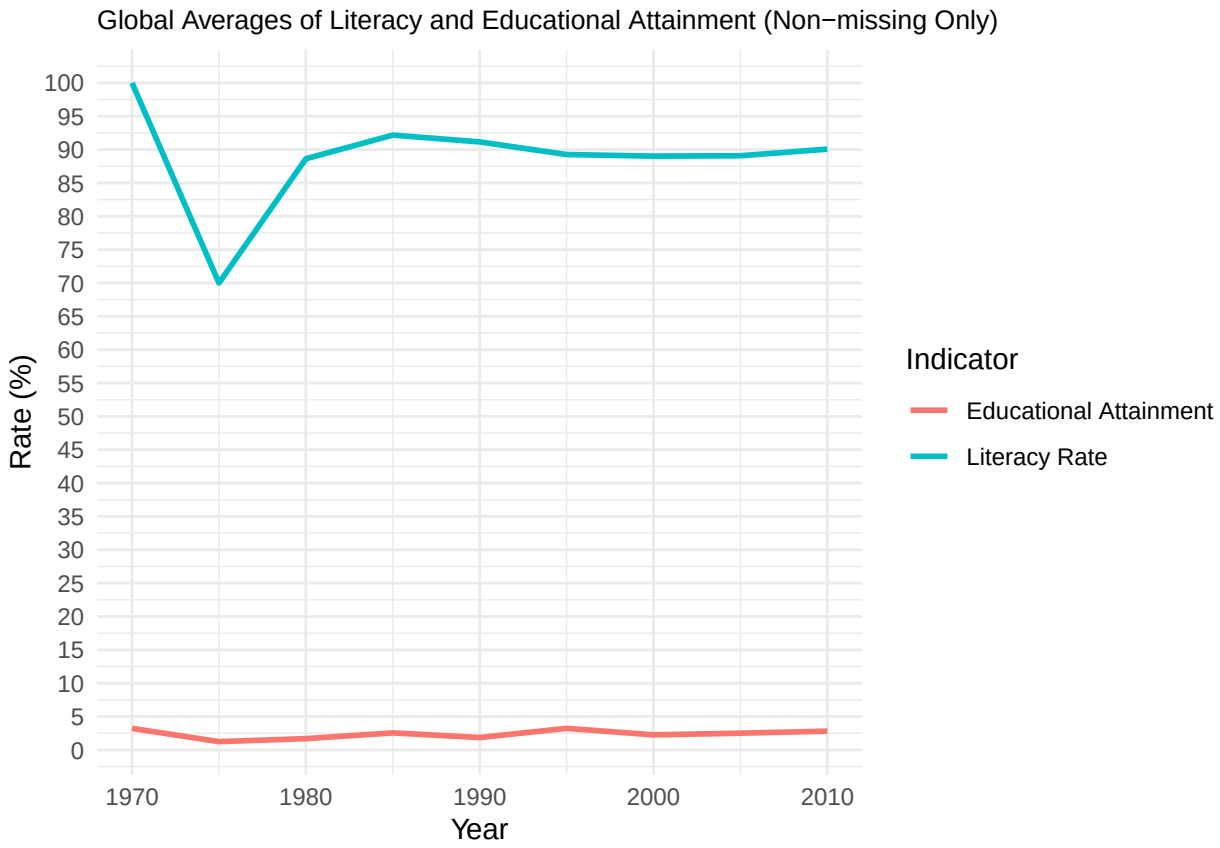
```
# Filter out rows where either literacy_rate_total or secondary_total_BL is missing
global_avg_nomiss <- merged %>%
  mutate(
    year = as.numeric(year),
    literacy_rate_total = as.numeric(literacy_rate_total),
    secondary_total_BL  = as.numeric(secondary_total_BL)
  ) %>%
  filter(
    !is.na(year),
    !is.na(literacy_rate_total),
    !is.na(secondary_total_BL)
  ) %>%
  group_by(year) %>%
  summarize(
    global_avg_literacy    = mean(literacy_rate_total, na.rm = TRUE),
    global_avg_edu_attain  = mean(secondary_total_BL, na.rm = TRUE)
  )

# Reshape into long format
```

```r
global_avg_nomiss_long <- global_avg_nomiss %>%
  pivot_longer(
    cols      = c("global_avg_literacy", "global_avg_edu_attain"),
    names_to  = "variable",
    values_to = "value"
  )

# Plot both lines on the same y-axis
## Determine maximum value for setting y-axis breaks
max_val_nomiss <- max(global_avg_nomiss_long$value, na.rm = TRUE)

ggplot(global_avg_nomiss_long, aes(x = year, y = value, color = variable)) +
  geom_line(size = 1) +
  labs(
    title = "Global Averages of Literacy and Educational Attainment (Non-missing Only)",
    x = "Year",
    y = "Rate (%)"
  ) +
  scale_color_discrete(
    name = "Indicator",
    labels = c(
      "global_avg_edu_attain" = "Educational Attainment",
      "global_avg_literacy"   = "Literacy Rate"
    )
  ) +
  scale_y_continuous(
    breaks = seq(0, ceiling(max_val_nomiss), 5)  # Ticks every 5 units up to the max data point
  ) +
  theme_minimal()  +
  theme(
    plot.title = element_text(size = 10)
  )
```


Global Averages of Literacy and Educational Attainment (Non−missing Only)
Rate (%)
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
1970
1980
1990
2000
2010
Year
Indicator
Educational Attainment
Literacy Rate