

Coding Challenge 4

Public Health 460

Due: Friday March 19th @ 5:00 PM EST

This assignment is all about working with data and making sure it is tidy. We will be working with joins and merges of datasets.

- There are 14 questions for credit, for a total of 20 points.
- If you are asked to use a specific function for a question, you must use it.
- Show all your code to receive credit.
- Read the questions all the way through, some ask for typed out answers.
- Include your `.rmd` and your `html` when submitting.
- Download the `mmr_data.csv` and `iso.csv` from the Google Drive.

This assignment requires you to work with two datasets. The first (`mmr_data.csv`) contains data from the United Nations maternal mortality estimation group. It provides measures of annual maternal mortality rates (MMR), measured in deaths per 100,000 live-births, for a set of countries around the world. Many countries are not able to provide accurate data on maternal mortality for some years. These data are a realistic example of what analysts have to work with when trying to estimate trends in maternal mortality across many years. Note that this dataset only contains a unique three-letter country code from the International Standards Organization (ISO). While it may be clear which countries some of these abbreviations refer to, others are not as clear.

The second dataset (`iso.csv`) contains information on countries, linking country names and continents with their ISO code.

- 1) Load the `tidyverse` and read in both datasets using relative path files (1 pt)
- 2) If you are interested in analyzing the change in maternal mortality over time using these datasets, what would the natural unit of analysis for these data be? (1 pt)
- 3) Is the `mmr_data` data frame in a “tidy” format based on the unit of analysis identified in question 2? Justify your answer. Use `head()` or `print()` to print out the first 10 rows of the dataset to justify your answer. (1 pt)
- 4) Use the function `pivot_longer()` to transform the MMR data into a long data format. Name the new column of observed MMR values as `mmr`. Remove rows with NA values for `mmr` from the transformed dataset. (3 pts)
- 5) Check to make sure you do not have any NAs in your new transformed dataset. (Hint: there are lots of ways to do this, but you could use either the `is.na()` or `anyNA()` function.) (1 pt)
- 6) How many observations are in the new tidy dataset? Run a line of code that gives you the answer. (1 pt)
- 7) What variable type is the `year` column in your newly tidy dataset. Print out a line of code that supports your written answer. (1 pt)
- 8) Change the type of the `year` variable to something different so that R can recognize it as a number. Use the `mutate()` function to do this (1 pt)

- 9) Perform a left join using the `left_join()` function to add information about the country name to the newly tidy dataset. Look at the joined table using `head()` or `print(...,n=5)` to view the first 5 rows. (1 pt)
 - 10) Join the same datasets in the same order using the function `full_join()`. (1 pt)
 - 11) What is the difference between a `full_join` and a `left_join`? Answer this question in the context of explaining the difference between the outputs for question 9) and 10). (2 pts)
- ** For the final three questions, use the complete dataset from the `left_join` completed above.****
- 12) Create an interactive plot that shows the data in a geom or geoms of your choice, faceted by continent and colored by country name. (2 pts)
 - 13) Compute the following summary values by continent: the total number of countries with data, the total number of observations, and the maximum and minimum observed MMR values. Print out the results from your computation (no need for fancy output, R console output is ok). (2 pts)
 - 14) Take the complete dataset, with MMR values and full country information (from the `left_join`), and transform it back to a non-tidy dataset that is similar to the original dataset but contains the additional rows with country name and continent. Use `pivot_wider()` to do this, the final dataset will have countries as rows and the years as columns. Additionally, add the prefix `yr` to the front of every column name referring to a year and make sure the columns are sorted chronologically (hint: read the help file for `pivot_wider()`). Print out the first five rows of your new dataset. (2 pts)