Introduction to R

Exercise B

The purpose of this exercise is to begin to familiarize you with reshaping, aggregating, and merging data frames. Footnotes below contain hints for completing each part.

For this exercise, we'll be using data on daily COVID-19 vaccination statistics from each of the 67 counties in the state of Pennsylvania. The data file, `ExerciseB.csv`, is available on the course Github repo in the "Homework" folder. It contains data from December 14, 2020 to January 3, 2023 on six variables:

- `Date`: the day on which that line of data were collected;
- `County.Name`: a character variable containing the name of the county in Pennsylvania in which those data were collected;
- `Partially.Vaccinated`: The number of individuals in that county receiving their first (of two) COVID-19 vaccinations on that date;
- `Fully.Vaccinated`: The number of individuals in that county receiving their second (of two) COVID-19 vaccinations on that date;
- `First.Booster.Dose`: The number of individuals in that county receiving their first COVID-19 booster on that date;
- `Second.Booster.Dose`: The number of individuals in that county receiving their second COVID-19 booster on that date.

Note that for all these variables, values of `NA` (missing values) imply zero (0) vaccinations / boosters of that type in that county on that date.

I. "Reshaping": Wide / Long Conversion

As stored in the Github repo, the COVID-19 vaccination data are in "long" format, with one line of data per county per day. The purpose of Part I of this exercise is to familiarize you with "reshaping" data from "long" format to "wide" format, and back again. Your assignment is as follows:

1. Begin by reading in the data from the Github repo.[1]
2. Replace the missing / `NA` values in the data with zeros, and convert the `Date` variable to a date format.[2]
3. Reshape the existing data into "wide" format, where each row is a single day (/ date) and each column is a county-measure.[3]
4. Reshape the resulting "wide" data again, this time back to its original "long" format.

Note that there are many ways to reshape data in R. Some more detailed explanations of what reshaping is and how it can be accomplished (using commands like `reshape`, or `pivot_longer` / `pivot_wider`) are available (e.g.) here, here, and here; a web search will turn up many, many others.

[1] The `read_csv` command in the `readr` package is one option for doing this.

[2] The `lubridate` package, and the `mdy()` command in particular, is your friend here.

[3] Because there are four variables measuring vaccinations / boosters, you will have four columns (first vaccination, second vaccination, first booster, second booster) for each county, along with the column for the date, yielding $(4 \times 67) + 1 = 269$ columns in your "wide format" data frame.

II. Aggregating

Next, we'll have you *aggregate* some data: combining information across different units (here, counties or dates) to create aggregate data.

1. Begin with the original ("long" format) COVID-19 data.
2. Create a new, aggregated time series data frame that has one observation for each day in the data, where the variables are now the statewide sums (that is, the aggregated / summed numbers across all 67 counties) for each of the four vaccination / booster variables.[4]
3. Create a second, similarly-aggregated time series data frame where the variables are now cross-county averages (arithmetic means) of the four COVID variables for each date.
4. Finally, create a temporally-aggregated data frame (that is, with one observation per county, so $N = 67$) containing the sums of each variable for each county over the entire period measured.

III. Merging

Finally, we're going to merge our COVID-19 data with other county-level data from Pennsylvania.

1. Read in the data from Wikipedia's page on counties in Pennsylvania.[5]
2. Match-merge these data with the last ($N = 67$) data frame you created in Part II.[6]
3. Create a population-adjusted / per capita variant of all four COVID statistics (that is, express each county's total vaccination / booster numbers as a proportion or percentage of their total population).
4. Finally, plot each of the population-adjusted COVID variables against the year in which the county was established, and arrange these four scatterplots in a single graph.[7]

[4] The `aggregate` command in base R works great for this, and is a good tool to learn. You can also / alternatively use the various "tidyverse" alternatives (e.g., `dplyr`).

[5] Here, the `htmltab` command in the package of the same name will be of potential use.

[6] The command is, unsurprisingly, called `merge`. Note that the county names need to be harmonized across the two data frames prior to merging; the `substr` (for "substring") command in base R will do this, as will various commands in the `stringr` package, and/or regular expressions.

[7] If you're using base R graphics, you can get a 2 × 2 matrix of four plots by prefacing the four `plot` commands with `par(mfrow=c(2,2))`.