Coding Challenge 3

Telling Stories with Data

Due: March 5, 5:00 pm

This assignment is about grouping variables and merging datasets together. Please turn in an HTML file and your .Rmd file.

- Inlcude all your code in the output unless specified.
- The `us_locations.csv` dataset is on Google Drive. This has data about states and counties in the US, including the standard FIPS code identifiers of locations, names of locations, and population sizes.
- If a question asks you to use a specific function, you must use it.

This assignment uses public health surveillance data on COVID-19. We will use the `covidcast` R package to retrieve data and conduct our analysis. This package is an incredibly powerful tool and opens up access to numerous datasets related to COVID that are part of the COVIDcast Epidata API created by the Delphi Group at Carnegie Mellon University. Check out the websites for more information about the numerous data signals available.

1) Install the `covidcast` package using the following code. Note that you should run the following code locally on your machine but you only need to install the package once, not every time you knit your assignment. Therefore, the following code should NOT be included in your knitted file. (0 points)

```
devtools::install_github("cmu-delphi/covidcast",
                         ref = "main",
                         subdir = "R-packages/covidcast")
```

2) Read in `us_locations.csv` and load the `tidyverse` package and `covidcast` package. (1 pt)

3) The `covidcast` package allows you to query over 1b rows of surveillance data related to COVID-19 that are stored centrally in the Epidata API. Run the following query to download confirmed COVID-19 case data as reported by the JHU Center for Systems Science and Engineering (CSSE) dashboard. Note that the specified `geo_values` are FIPS codes that correspond to Hampshire, Franklin and Middlesex counties in Massachusetts. (1 pt)

```
covid_cases <- covidcast_signal(data_source = "jhu-csse",
                                signal = "confirmed_incidence_num",
                                start_day = "2020-03-01", end_day = "2021-02-20",
                                geo_values = c("25015", "25011", "25017"))
```

4) Look at the resulting dataset in the local data viewer. What is the unit of analysis of this dataset? In other words, what unit of observation does each row represent? (2 pts)

5) Use the `group_by()` and `summarize()` functions to compute the total number of confirmed COVID-19 cases reported in each of the three counties between March 1, 2020 and February 20, 2021. Print out the results in your report (no need to do any fancy formatting, standard R console output is ok, as long as it is clear which cumulative case count corresponds to each county). (4 pts)

6) Adapt the code you used in the previous question to print out the number of cases each county had in all of 2020. (2 pts)

7) To make the results in the previous question a bit more detailed and human-readable, perform the following analyses (ideally, although no required to be, in one chained `dplyr` command) (4 pts):

   a) use `left_join()` to join the `us_locations` dataset with the results above. (Tip: order matters when using left join - you only need to merge in the data from `us_locations` for the three counties for which you you have data.)
   b) create a new variable called `fraction_infected` that computes the number of confirmed cases as a fraction of the total population of the county
   c) print out the results in four columns (in this order): county name, total confirmed infections, population, fraction infected.

8) Query the Epidata API for confirmed cases for all counties in the month of December 2020 and save this data as a new object in your workspace. (Tip: you can leave the `geo_values` argument unspecified and it will return all counties.) (1 pt)

9) Using similar computations as in previous questions, compute and print out the 10 counties with the highest rate of confirmed COVID-19 cases in December 2020 relative to their population size. (3 pts)

10) Re-run the same analysis as above but use `group_by()` and `filter()` so that only the county with the highest fraction confirmed cases in December 2020 is shown. The resulting output should show the top 10 counties, with only one line for each state. (2 pts)