# Introduction to R

## Exercise C

This exercise is, to use a favorite Britishism of mine, a bit of a dog's breakfast of topics.

### Part I: Simulations

The purpose of this part is to familiarize you further with (and give you practice) conducting simulations using R . There are many, many useful resources for learning this aspect of what we do; a few I like are here, here, here, and here.

Chances are that most of the empirical work you do as a social scientist will not involve a great deal of simulation. With that said, the value of having basic-to-moderate programming skills in an otherwise-useful language like R are several: you'll better understand how the packages and routines you use work; you'll probably write better, clearer code; and you'll be able to keep abreast of the fast-changing capabilities of R better than you otherwise would. Moreover, because it is 2023 and computers are pretty fast, most of the kinds of things you *will* be doing will not require the kind of speed-focused optimization that people who work in the tech industry obsess about;[1] that means that even a relatively basic understanding of programming is enough to do quite a lot.

For this part of the assignment, please:

1. Pick a number between 8 and 23; we'll call that $n_1$. Pick another number between 310 and 1476 (call that one $n_2$).

2. Write a loop to draw $N_2$ observations $X$ such that $X \sim N(n_1, 0.5 \times n_1)$ and save them to a data frame.

3. Write a nested loop to draw $N_2$ observations from each of a series of Normal distributions with $N(\mu, 0.7 \times \mu)$ where $\mu \in \{1, 2, ...n_1\}$ and save them to $n_1$ data frames.

4. Write R code to loop over the elements of $Z \in \{$`red`, `yellow`, `black`, `blue`, `seagreen`$\}$; for each value, draw 3144 observations from a $\chi^2_{n_1-k}$ distribution, where $k$ is the number of letters in the respective element of $Z$. Store those draws in an object named with the value of that element of $Z$ (so, name the first batch of draws "`red`," the second "`yellow`," etc.). Plot the density of each set of draws against [0,30], distinguishing each density by its respective color in $Z$. Do each of these steps by referencing elements of $Z$ in the loop(s).[2]

5. Repeat steps 2-4, this time using `apply` functions rather than loops.[3]

[1] It is for this reason that we won't dig deeply into topics like parallelization, running R on high-performance machines, building in calls to lower-level / faster languages like C++, and so forth.

[2] Hint: `get` will probably be helpful here.

[3] Help for the various – and sometimes confusing – variants of `apply` can be found (e.g.) here, here, here, and/or here.

**Part II: Maps**

This part of the exercise is about *maps*. Maps are to your readers and interlocutors as catnip is to cats: an irresistible treat, prone to causing irrational outbursts of sheer joy.[4] Moreover, they can be incredibly useful (and powerful) in many contexts. The goal, then, is to give you some basic experience in making maps using R , and in integrating data from other sources with geolocated data to create simple things like choropleths.

For this part of the exercise, your (relatively simple) task is to create two maps.

II.A

Your first map will be a county-level map of the U.S. that will use data from the New York Times COVID-19 Github repo. Specifically:

1. On the Github repo, get the county-level data *for the date of your most recent birthday*; this will be in either the `us-counties-2021.csv` or (more likely) `us-counties-2022.csv` data files.
2. Create a county-level choropleth of the number of recorded cases of COVID-19 in each county as of that date (that is, the `cases` variable). Note that this should be the raw numbers of cases, not normalized by population. Be sure that your map includes a key/legend.

II.B

Your second map will be a country-level map, using national-level data on every country in the international system. Specifically, we'll examine (and plot) data from the OECD's 2019 Gender, Institutions and Development Database, details of which can be found here. More specifically still, your task is to create a choropleth of the variable measuring the "percentage of girls aged 15-19 years ever married, divorced, widowed or in an informal union" (variable `DF_CM_PRACT`). As in part A above, be sure that your map includes a legend specifying the values associated with the shades on the map.

For both maps, here are a few hints:

- For U.S. counties and states, the Census Bureau's Federal Information Processing Standards (FIPS) codes are the gold standard for ensuring reliable identification of geographic entities (and thus are especially valuable for merging data from different sources).
- Country-level data are most often identified by ISO-3166-1 codes, conventionally their three-letter alphabetic code (so, for example, `TUV` for Tuvalu).
- As far as R code goes, this is a good general reference; see especially Chapter 9. For maps (like these) with widely-used, well-defined geolocated units, take a look at the `choroplethr`, `tmap`, `spplot`, and `GISTools` packages; see here for more.

[4] This statement has not been evaluated by the American Medical Association, the Humane Society of the United States, or the American Association of Geographers.