

Problem Set 1

ECON 480 — Fall 2022

Due by Class Wednesday September 21

The Popularity of Baby Names

Install and load the package `babynames`. Get help for `?babynames` to see what the data includes. **Also, don't forget to load `tidyverse`!**

1.
 - a. What are the top 5 boys names for 2017, and what *percent* (note *not* the proportion!) of overall names is each?

- b. What are the top 5 *girls* names for 2017, and what *percent* of overall names is each?

2. Make two barplots of these top 5 names, one for each sex. Map aesthetics `x` to `name` and `y` to `prop` [or `percent`, if you made that variable, as I did.] and use `geom_col` (since you are declaring a specific `y`, otherwise you could just use `geom_bar()` and just an `x`.)

3. Find your name. [If your name isn't in there, pick a random name.] `count` by `sex` how many babies since 1880 were named your name. [Hint: if you do only this, you'll get the number of *rows* (years) there are in the data. You want to add the number of babies in each row (`n`), so inside `count`, add `, wt = n` to weight the count by `n`.] Also create a variable for the percent of each sex.

4. Make a line graph of the number of babies with your name over time, colored by `sex`.

5.

- a. Find the most common name for boys by year between 1980-2017. [Hint: you'll want to first `group_by(year)`. Once you've got all the right conditions, you'll get a table with a lot of data. You only want to keep just the 1st row of each grouped-year's data, so add `%>% slice(1)`.]

- b. Now do the same for girls.

6. Now let's graph the evolution of the most common names since 1880.
 - a. First, find out what are the top 10 *overall* most popular names for boys and for girls in the data. [Hint: first `group_by(name)`.] You may want to create two objects, each with these top 5 names as character elements.
 - b. Now make two `linegraphs` of these 5 names over time, one for boys, and one for girls. [Hint: you'll first want to subset the data to use for your `data` layer in the plot. First `group_by(year)` and also make sure you only use the names you found in Part A. Try using the `%in%` command to do this.]

Political and Economic Freedom Around the World

For the remaining questions, we'll look at the relationship between Economic Freedom and Political Freedom in countries around the world today. Our data for economic freedom comes from the [Fraser Institute](#), and our data for political freedom comes from [Freedom House](#).

7. Download these two datasets that I've cleaned up a bit: [If you want a challenge, try downloading them from the websites and cleaning them up yourself!]

- [econ_freedom.csv](#)
- [pol_freedom.csv](#)

Below is a brief description of the variables I've put in each dataset:

Econ Freedom

Variable	Description
year	Year
ISO	Three-letter country code
country	Name of the country
ef_index	Total economic freedom index (0 - least to 100 - most)
rank	Rank of the country in terms of economic freedom
continent	Continent the country is in

Pol Freedom

Variable	Description
country	Name of the country
C/T	Whether the location is a country (C) or territory (T)
year	Year
status	Whether the location is Free (F), Partly Free (F) or Not Free (NF)
fh_score	Total political freedom index (0 - least to 100 - most)

Import and save them each as an object using `my_df_name <- read_csv("name_of_the_file.csv")`. I suggest one as `econ` and the other as `pol`, but it's up to you. Look at each object you've created.

8. Now let's join them together so that we can have a single dataset to work with. You can learn more about this [in the 1.4 slides](#). Since both datasets have both `country` and `year` (spelled exactly the same in both!), we can use these two variables as a `key` to combine observations. Run the following code (substituting whatever you want to name your objects):

```
freedom <- left_join(econ, pol, # join pol tibble to econ tibble
                    by = c("country", "year")) # keys to match variables between two tibbles
```

Take a look at `freedom` to make sure it appears to have worked.

9.
 - a. Make a barplot of the 10 countries with the highest Economic Freedom index score in 2018. You may want to find this first and save it as an object to use for your plot's `data` layer. Use `geom_col()` since we will map `ef_index` to `y`. If you want to order the bars, set `x = fct_reorder(ISO, desc(ef_index))` to reorder `ISO` (or `country`, if you prefer) by EF score in descending order.
 - b. Make a barplot of the 10 countries with the highest Freedom House index score in 2018, similar to what you did for Part A.

10. Now make a scatterplot of Political freedom (`fh_score` as `y`) on Economic Freedom (`ef_index` as `x`) in the year 2018, and color by continent.

11. Save your plot from Question 10 as an object, and add a new layer where we will highlight a few countries. Pick a few countries (I suggest using the ISO code) and create a new object `filtering` the data to only include these countries (again the `%in%` command will be most helpful here).

Additionally, *install* and *load* a package called "ggrepel", which will adjust labels so they do not overlap on a plot.

Then, add the following layer to your plot:

```
geom_label_repel(data = countries, # or whatever object name you created
                 aes(x = ef_index,
                    y = fh_score,
                    label = ISO, # show ISO as label (you could do country instead)
                    color = continent),
                 alpha = 0.5, # make it a bit transparent
                 box.padding = 0.75, # control how far labels are from points
                 show.legend = F) # don't want this to add to the legend
```

This should highlight these countries on your plot.

12. Let's just look only at the United States and see how it has fared in both measures of freedom over time. `filter()` the data to look only at the United States (its ISO is "USA"). Use both a `geom_point()` layer and a `geom_path()` layer, which will connect the dots over time. Let's also see this by labeling the years with an additional layer `geom_text_repel(aes(label = year))`.