

An Analysis of Global Wealth Distribution Through Forbes 400 Wealthiest People

Github link: <https://github.com/jacklang0/SI206FinalProject>

1. The goals for your project (10 points)

The goals of this project were to analyze the global wealth distribution. Specifically, we wanted to look at the characteristics of the 400 wealthiest individuals. The data was pulled from the Forbes 400 API and wikipedia page on Distribution of Wealth. With this information, we planned to analyze how the wealthiest individuals compare to the wealth from their origin company, counts of countries on the list, and trends in industries, gender, income inequality, and age on the Forbes 400 List.

From the calculations and graphs comparing different variables, we hoped to analyze how inequalities exist throughout countries based on the gini-coefficient percentage, gender representation of wealthiest individuals, age, and count of people on the Forbes 400 list. We also hoped to calculate total wealth of the US, highest and lowest gini percentages, and country's counts on the Forbes 400 list.

2. The goals that were achieved (10 points)

We were able to analyze many global wealth trends in this project, comparing factors of inequality in countries and across the Forbes 400 List. The calculations of total US total wealth and top wealth of 100 US billionaires as shown in Figure 1 supports that there is a large wealth disparity. 100 people have about 2.3% of US income, compared to over 334 million other US citizens. Additionally, Figure 2 and the calculations of the five most represented Forbes 400 countries in CalculatedData.txt supports huge differences between representation. More developed countries including the United States, China, Russia, and India have the most people on the Forbes 400 list.

Another goal we were able to achieve in this project is measuring gender inequality on the Forbes 400 list. As shown in Figure 2 and Figure 3, representation of males is much larger in the wealthiest individuals and the top five industries from the Forbes 400 list.

Two goals that we couldn't achieve in this project are identifying inequalities from the Forbes 400 list based on age and the gini-coefficient of a country. A gini-coefficient measures income inequality in a country, with higher percentages meaning greater inequality. Despite these varying gini-coefficients in Figure 4, there does not seem to be an association with the number of people on the Forbes 400 list. Additionally, there does not seem to be an association between age and net worth from Figure 5.

3. The problems that you faced (10 points)

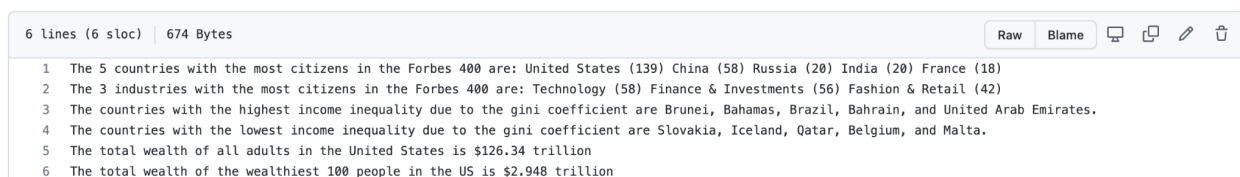
The Forbes 400 API contained a column of birth dates rather than age. Unfortunately, the birth dates were in the form of a 12 digit positive or negative number (i.e. -188438400000) and there was no documentation as to what that represented. Using problem-solving skills, we deduced that this number was 1000s of seconds after (or before if negative) 1970. Thus we were able to correctly calculate the age of the people.

When mapping the countries in the Forbes list to the wikipedia page, there were some countries that had different names in each source (i.e. Czechia vs Czech Republic). We had to rename one of the countries to map correctly. Also, there was one country (Monaco) which did not show up in the Wiki data source so we inserted it into the CountryWealth table with no wealth data.

The Wikipedia data source varied in units on the table. An example is the counts of adults represented as thousands but other entries were as percentages. In order to overcome this issue, we had to check unit conversions to make sure that the calculations were accurate.

4. Your file that contains the calculations from the data in the database (10 points)

<https://github.com/jacklang0/SI206FinalProject/blob/main/CalculatedData.txt>



```
6 lines (6 sloc) | 674 Bytes
1 The 5 countries with the most citizens in the Forbes 400 are: United States (139) China (58) Russia (20) India (20) France (18)
2 The 3 industries with the most citizens in the Forbes 400 are: Technology (58) Finance & Investments (56) Fashion & Retail (42)
3 The countries with the highest income inequality due to the gini coefficient are Brunei, Bahamas, Brazil, Bahrain, and United Arab Emirates.
4 The countries with the lowest income inequality due to the gini coefficient are Slovakia, Iceland, Qatar, Belgium, and Malta.
5 The total wealth of all adults in the United States is $126.34 trillion
6 The total wealth of the wealthiest 100 people in the US is $2.948 trillion
```

5. The visualization that you created (i.e. screenshot or image file) (10 points)

Our five visualizations that we created are shown below. Figure 1 is a pie chart comparing the percentage of total US wealth from the top 100 wealthiest individuals to the remaining 334 million Americans. Figure 2 is a segmented horizontal bar graph that shows the count of each country's representation in the Forbes 400 list, separated by gender. Figure 3 is a vertical side-by-side bar graph on the count of the top five industries in the Forbes 400 list, separated by gender of the industries on the list. Figure 4 is a scatterplot comparing the number of people on the Forbes 400 List and the gini-coefficient of countries. Figure 5 is a scatterplot that compares age vs. net worth from the Forbes 400 list.

Figure 1: Top100USBillionaires.png

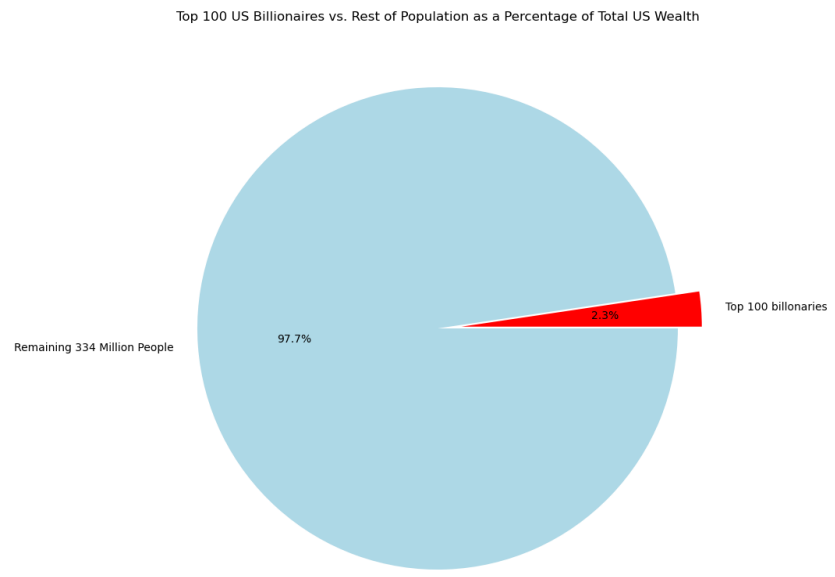


Figure 2: CountryCounts.png

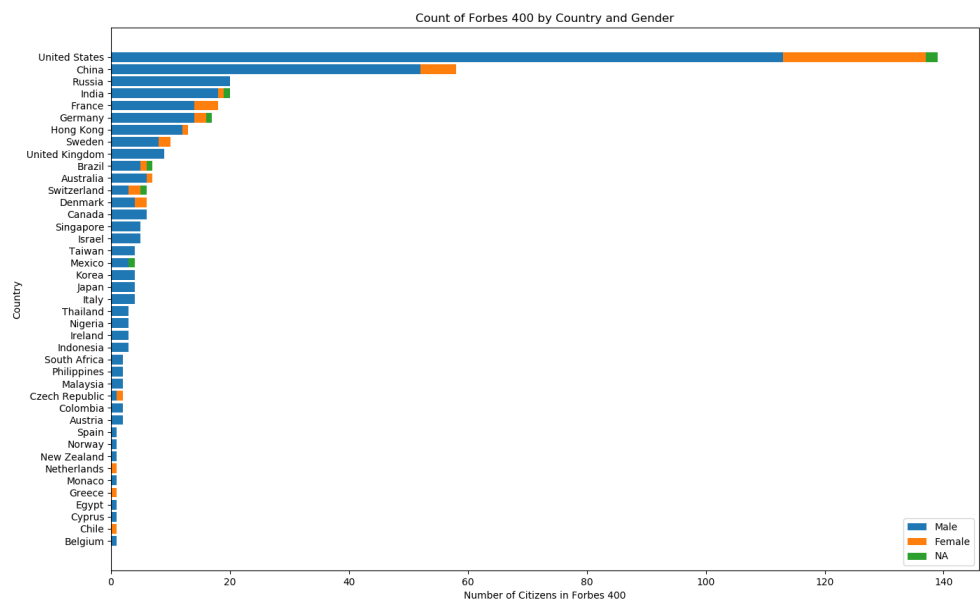


Figure 3: IndustryCounts.png

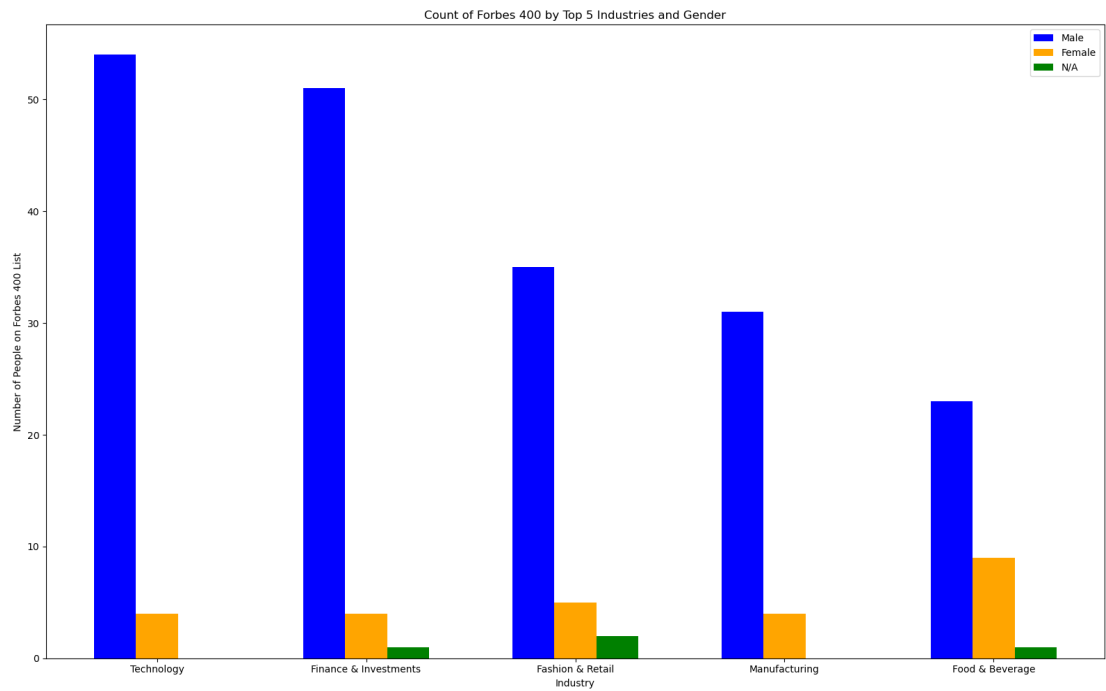


Figure 4: GiniCountGraph.png

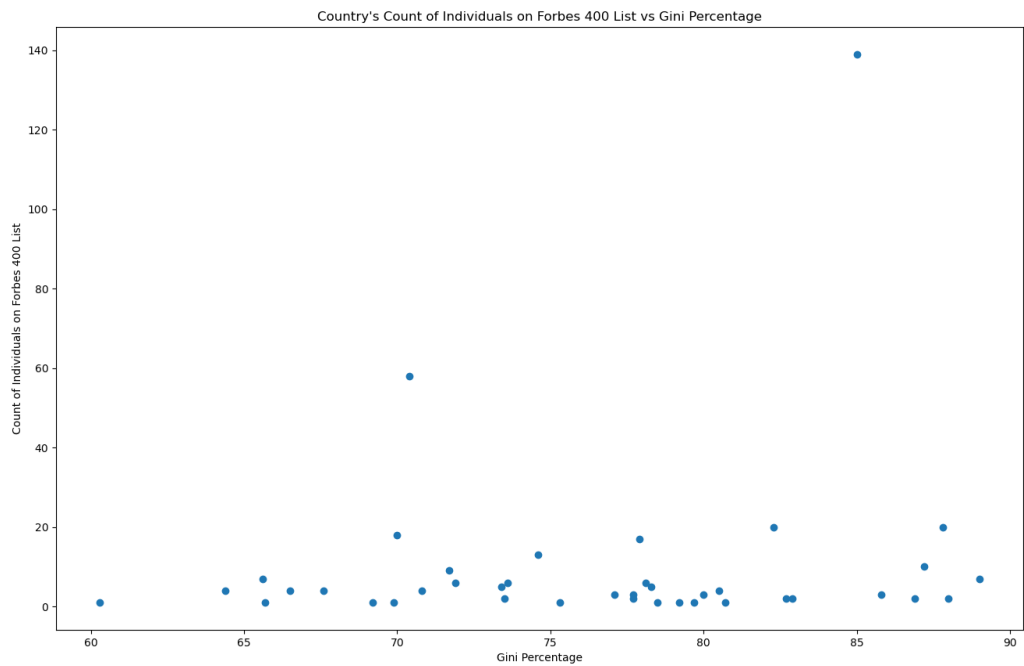
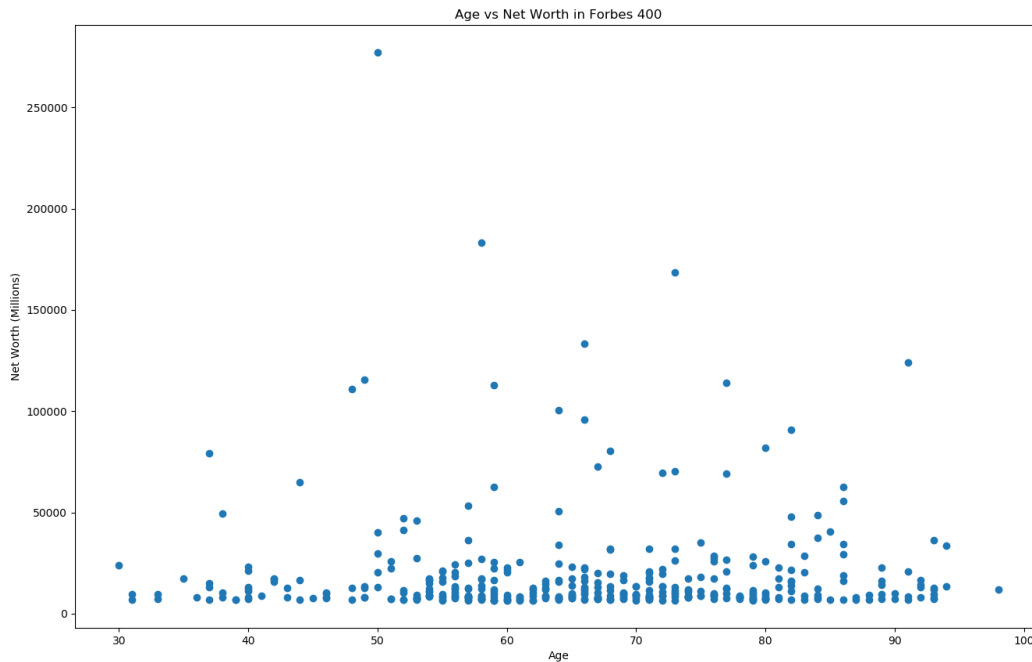


Figure 5: AgeVsNetworth.png



6. Instructions for running your code (10 points)

- Delete database Wealth.db from folder if testing how elements are added into database.
- Run file CountryWealthDistribution.py first. It creates the Beautiful Soup object for the Wikipedia website page and the starting table in Wealth.db called CountryWealth, which the other tables created use the country key from.
- Next, run the file ForbesAPIPull.py. This grabs information from the Forbes 400 list and creates two more tables in Wealth.db: ForbesPeople and Industries. First in def main(), create local cache by uncommenting “data = call_api()” and “write_cache(data, filename)” and commenting out the calls to “insert_into_industries()” and “insert_into_people”. Run the program. Then, comment out “data = call_api()” and “write_cache(data, filename)” and uncomment “insert_into_industries()” Repeatedly run the program until the Industries table stops getting rows added. Then, uncomment “insert_into_people()” and repeatedly run the program until the ForbesPeople table stops getting rows added.
- Run Calculations.py file based on information stored in database
- Check how information from calculations is stored in CalculatedData.txt
- Run Visualization.py file. To check how visualizations are saved, please visit Top100USBillionaires.png, CountryCounts.png, IndustryCounts.png, GiniCountGraph.png, and AgeVsNetworth.png.

7. Documentation for each function that you wrote. This includes the input and output for each function (20 points)

Documentations for functions in ForbesAPIPull.py:

- `def read_cache(filename):`
 - Function Description: Reads json from cache
 - Input: cache file name
 - Output: json variable
- `def write_cache(data, filename)`
 - Function Description: Saves json in a local cache
 - Input: json variable, name of cache file
 - Output: None
- `def call_api()`
 - Function Description: Calls Forbes400 API to get json of Forbes list
 - Input: None
 - Output: json file of Forbes list
- `def set_up_database(db_name)`
 - Function Description: Sets up connection to the Wealth.db
 - Input: name of database
 - Output: SQLite3 cursor and connection variables
- `def create_tables(cur, conn)`
 - Function Description: Creates ForbesPeople and Industries tables in db
 - Input: SQLite3 cursor and connection to db
 - Output: None
- `def get_key_counter(cur, table_name)`
 - Function Description: Returns the id value of the next id to be inserted into table
 - Input: SQLite3 cursor and connection to db and table name
 - Output: id value of the next id to be inserted into table
- `def insert_into_industries(cur, conn, data, n=25)`
 - Function Description: Inserts n new/unique entries into Industries
 - Input: SQLite3 cursor and connection to db, json variable, number of rows (25)
 - Output: None
- `def insert_into_people(cur, conn, data, n=25)`
 - Function Description: Inserts n new/unique entries into ForbesPeople
 - Input: SQLite3 cursor and connection to db, json variable, number of rows (25)
 - Output: None

Documentation for functions in CountryWealthDistribution.py

- `def get_website_info()`
 - Function Description: This function creates a Beautiful Soup object of the Wikipedia page on the global distribution of wealth From this object the function navigates to a table and creates a list of lists with that contains the country, number of adults, mean wealth per adult, media wealth per adult, percent of people with under \$10,000, percent of people with between \$10,000 and \$100,000, percent of people with between \$100,000 and \$1 Million, and the gini-percentage of the country.
 - Input: No parameters. Wikipedia url = https://en.wikipedia.org/wiki/Distribution_of_wealth
 - Output: Returns a list of each country's information on table

- `def setUpDatabase(db_name)`
 - Function Description: This function sets a path to set up the database and creates a cursor and connection for the SQL database.
 - Input: Database name. In the main function, the name is “Wealth.db”
 - Output: The output is the cursor (cur) and connection (conn)
- `def get_key_counter(cur)`
 - Function Description: This function selects the max key from the table CountryWealth created, in order to figure out what key the cursor should insert into if the program is run. This helps meet the requirement of adding 25 or fewer items at a time into a table, as it inserts 25 consecutive items from the max key added in the table last.
 - Input: Cursor for SQL database (cur)
 - Output: Key of last entry added into table. If the table is blank, the function returns 0, telling the program to start the database key as 0.
- `def create_website_database(wealth_info, cur, conn, counter)`
 - Function Description: This function creates a table for each country from the information retrieved from `get_website_info()`, starting at the max key returned from `get_key_counter()`. The table created if it doesn’t already exist is CountryWealth. Information from each country is inserted into the table up to 25 times, with a counter to check that only 25 or fewer items are added at a time.
 - Input: List of information for each country (wealth info), the cursor (cur), the connection (conn), and a counter which is equal to the max key return from `get_key_counter()`.
 - Output: Function does not return anything, but does commit a table called CountryWealth

Documentation for functions in Calculations.py

- `def total_wealth_US(US_key, cur, conn)`
 - Function Description: This function selects the number of adults and mean wealth per adult from the United States entry CountryWealth table in order to return an approximation of US total wealth.
 - Input: The US key to get the correct table row (161), cursor (cur), and connection for database (con).
 - Output: Total wealth of the United States calculation, approximately \$126.34 trillion.
- `def get_gini(cur, conn)`
 - Function Description: This function selects the country name and gini percentage from the CountryWealth table, sorts the results by the gini percentage, and returns a string that states the five countries with highest income inequality and the five countries with the lowest income inequality.
 - Input: The cursor (cur) and connection (conn) for SQL database
 - Output: Returns a string of results stating the five countries with the highest income inequality and the five countries with the lowest income inequality.
- `def get_countries_with_most_forbes400(cur, conn)`

- Function Description: This function queries the ForbesPeople and CountryWealth tables for the number of people on the Forbes list in each country and then outputs a returns a message string with the top 5 countries
- Input: SQLite3 cursor and connection to db
- Output: String message of top 5 countries with most people on Forbes list
- def get_industries_with_most_forbes400(cur, conn)
 - Function Description: This function queries the ForbesPeople and Industries tables for the number of people on the Forbes list in each country and then outputs a returns a message string with the top 5 industries
 - Input: SQLite3 cursor and connection to db
 - Output: String message of top 5 industries with most people on Forbes list
- def get_wealth_of_top_N_US_forbes400(cur, conn, N)
 - Function Description: This function queries the ForbesPeople and Industries tables to get the net worth of every US person on Forbes list. Then sums the total and returns dollar amount total
 - Input: SQLite3 cursor and connection to db
 - Output: Sum of the net worth of top wealthiest people in US in dollars
- def write_calcs_to_file(cur, conn)
 - Function Description: Calls the functions which calculate information and write them to "CalculatedData.txt"
 - Input: SQLite3 cursor and connection to db
 - Output: None

Documentation for functions in Visualization.py

- def graph_top_100_vs_total_US_wealth(cur, conn)
 - Function Description: This function calculates the total wealth of the US at the key of 161 on the CountryWealth table. It then selects the net worth of the 100 richest US citizens on the Forbes 400 list from the ForbesPeople table, and sums the total hundred people's wealth. From this information the function creates a pie chart comparing the wealth of the top US billionaires vs the rest of US citizens in trillions of dollars, using the sizes as a fraction of the US total wealth.
 - Input: Cursor (cur) and connection (conn) to database
 - Output: Nothing is returned in the function, but the pie chart is saved as the figure Top100USBillionaires.png
- def graph_count_by_country(cur, conn)
 - Function Description: This function queries the ForbesPeople and CountryWealth tables for the total number of people on the Forbes list and the counts of each gender for each country. Then it creates a horizontal bar graph of this data.
 - Input: SQLite3 cursor and connection to db
 - Output: None
- def graph_count_of_forbes_by_industry(cur, conn)
 - Function Description: This function gets the count of people in the top five industries on the Forbes 400 list in ForbesPeople and Industries table, separated by gender. From this information, a vertical side-by-side graph is created of the number of people in each of the top five industries, separated by gender.
 - Input: Cursor (cur) and connection (conn) to database

- Output: Nothing is returned in the function, but the bar graph is saved as the figure IndustryCounts.png
- def graph_gini_vs_number_billionaires(cur, conn)
 - Function Description: This function selects the gini percent from the CountryWealth table and number of people in each country from the ForbesPeople tables. With this information a scatterplot is created to compare the gini percentage and number of people on the Forbes 400 list for each country.
 - Input: Cursor (cur) and connection (conn) to database
 - Output: Nothing is returned in the function, but the scatterplot is saved as the figure GiniCountGraph.png
- def graph_age_vs_net_worth(cur, conn)
 - Function Description: This function queries the ForbesPeople tables for age and net worth of each individual in the Forbes 400 list. Then it creates a horizontal bar graph of this data.
 - Input: SQLite3 cursor and connection to db
 - Output: None
- def set_up_database(db_name)
 - Function Description: Sets up connection to the Wealth.db
 - Input: name of database
 - Output: SQLite3 cursor and connection variables

8. You must also clearly document all resources you used. The documentation should be of the following form (20 points)

Date	Issue Description	Location of Resource	Issue Resolved?
4/5/22	How to create retrieve information from Forbes API	Forbes 400 Documentation: https://github.com/jesseok/eya/Forbes400	Yes, used resource to learn how to as set parameters and retrieve API information
4/5/22	Adding only 25 items to database instead of	Talked to Professor Ericson, she advised that we used MAX key in order to add to the right location of the database	Issue was resolved. Able to add 25 or fewer items at a time
4/6/22	Needed to convert integer representing seconds after 1970 to age	https://www.w3schools.com/python/python_datetime.asp	Yes, looked at documentation of datetime variable and timedelta function
4/12/22	Did not remember file writing syntax	https://www.w3schools.com/python/python_file_writing.asp	Yes, just needed a quick refresher

4/16/22	Needing documentation of Matplotlib stacked bar graph	https://pythonguides.com/stacked-bar-chart-matplotlib/	Yes, site gave a good example
4/16/22	Bar graph was in ascending order for country counts, descending made more sense	https://www.codegrepper.com/code-examples/python/barh+reverse+order	Yes, just needed ax.invert_yaxis()
4/25/22	Making the bar graph side by side	Internet: https://www.educative.io/edpresso/how-to-create-a-bar-chart-using-matplotlib	Yes - was able to make bar graph side-by-side instead of segmented