SI 206 Final Project Report
Team Name: Indo
Team Members: Reinhard Wilmer (rwilmer) and Vivian Qian Yu Wong (wvivian)
Github Repo: https://github.com/wong-vivian/Final-Project
Project Goals:

- Research Question: How does a country's population and economic power correlate with its Covid-19 efforts?
- We measure economic power through Gross Domestic Product (GDP) in US Dollars (USD) and Covid-19 efforts through death and recovery rate per country.
- We aim to produce graphs that relate (1) death rate and population, (2) recovery rate and population, (3) recovery rate and GDP, (4) death rate and recovery rate.
- We aim to produce a file that details (1) Total confirmed Covid cases, (2) Total Covid related deaths, (3) Total recovery from Covid cases, (4) World Covid related Death Rate, (5) World Covid Recovery Rate, (6) World Average Covid Death Rate, (7) World Average Covid Recovery Rate, (8) World Average Covid Infection Rate

Achieved Goals:

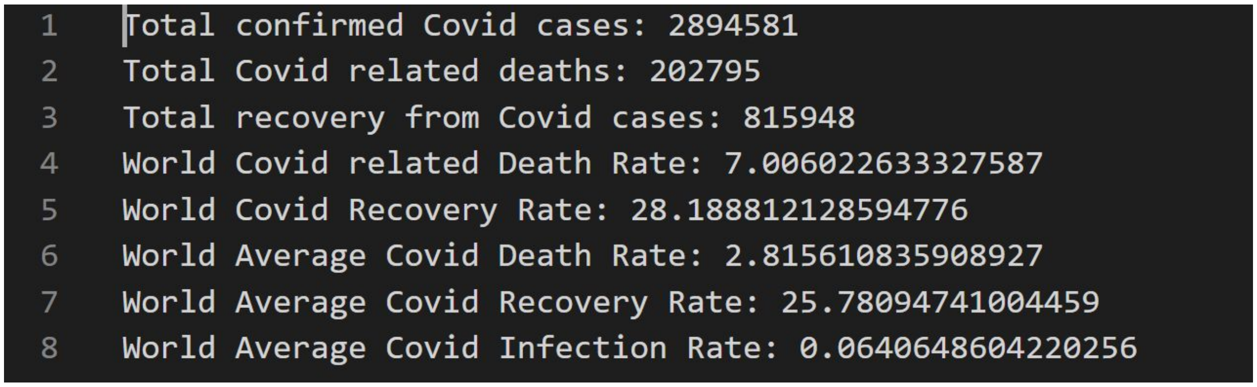
- We were able to produce 4 graphs that were able to show relation in all four aimed categories: (1) death rate and population, (2) recovery rate and population, (3) recovery rate and GDP, (4) death rate and recovery rate.
- We were able to produce a text file that shows our findings from both the Covid-19 and World Bank API's that details the following information: (1) Total confirmed Covid cases, (2) Total Covid related deaths, (3) Total recovery from Covid cases, (4) World Covid related Death Rate, (5) World Covid Recovery Rate, (6) World Average Covid Death Rate, (7) World Average Covid Recovery Rate, (8) World Average Covid Infection Rate.
- We were able to answer the research question through the results of our findings and we find some correlation between a country's population and economic power to its Covid-19 efforts/data.

Problems Faced:

- Different country names from API leads to missing countries when joining the Covid and Country_Indicator tables produced. For example, in Covid-19 API, the United States is listed as "United States of America" while the World Bank API lists them as "United States", hence, in the resulting joined table, we lost the data point and thus, the produced graph lost a key contributor to one of our goals.
- When pulling data from the World Bank API, we couldn't pull both population and GDP at the same time because the request url did not allow for both indicators to be pulled together. Hence, we had to make an adjustment by making another function.

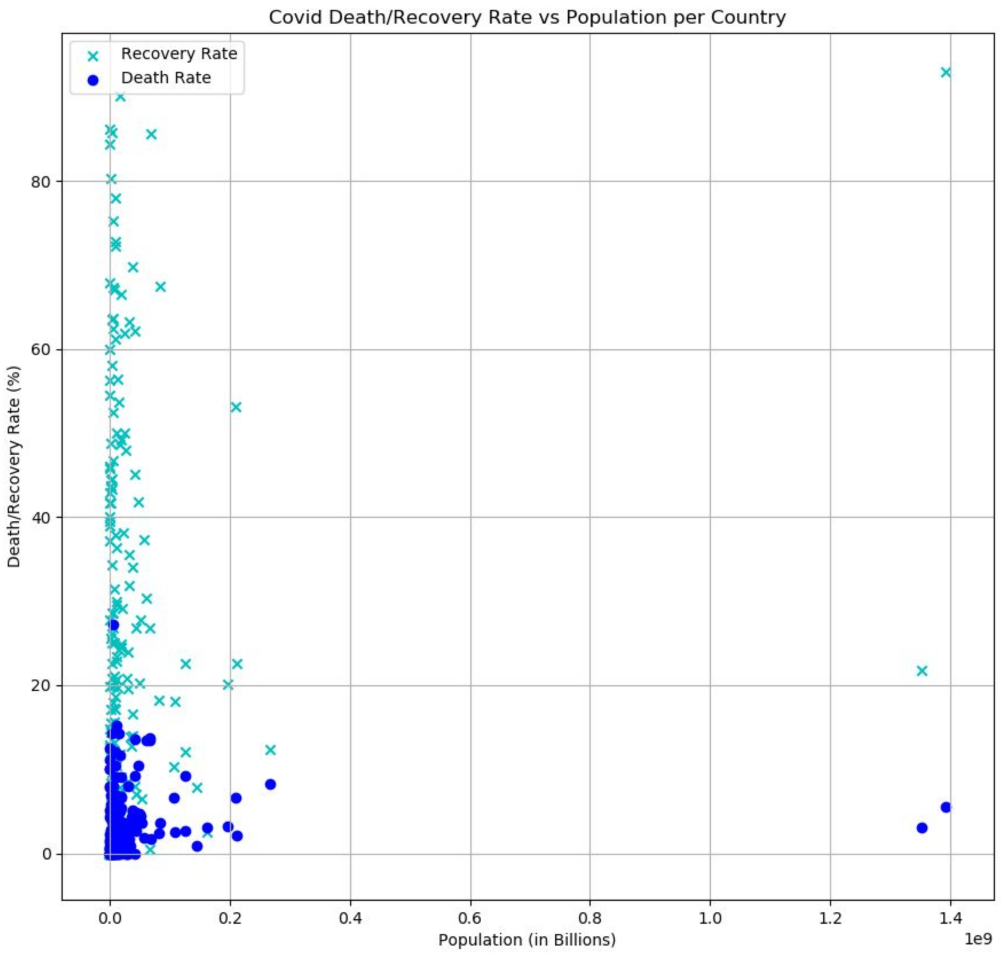
- The World Bank API includes different groupings such as “Arab World”, “World”, etc. that already aggregates the data for the different groupings. We resolved this when we joined the table with the Covid table since it only includes countries not groupings.

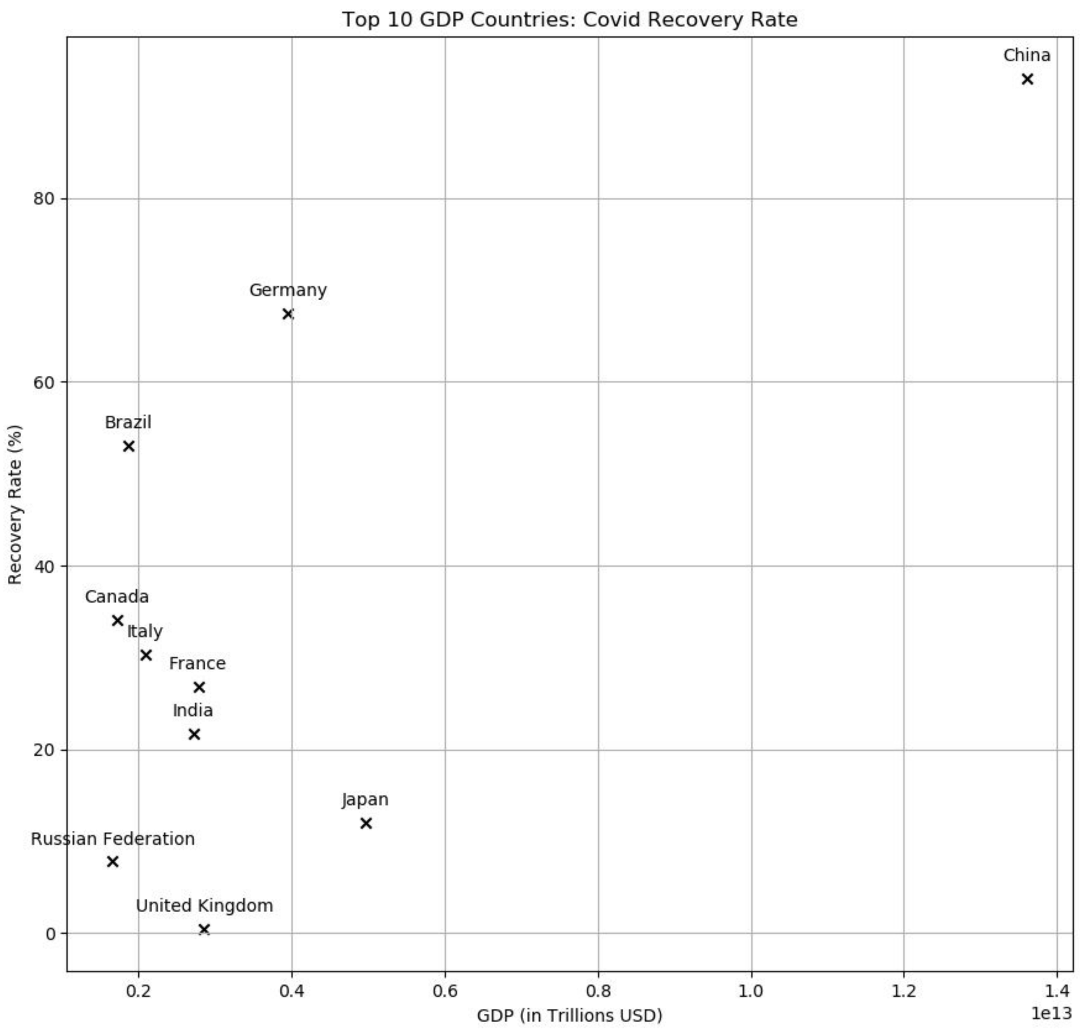
File (Calculation)

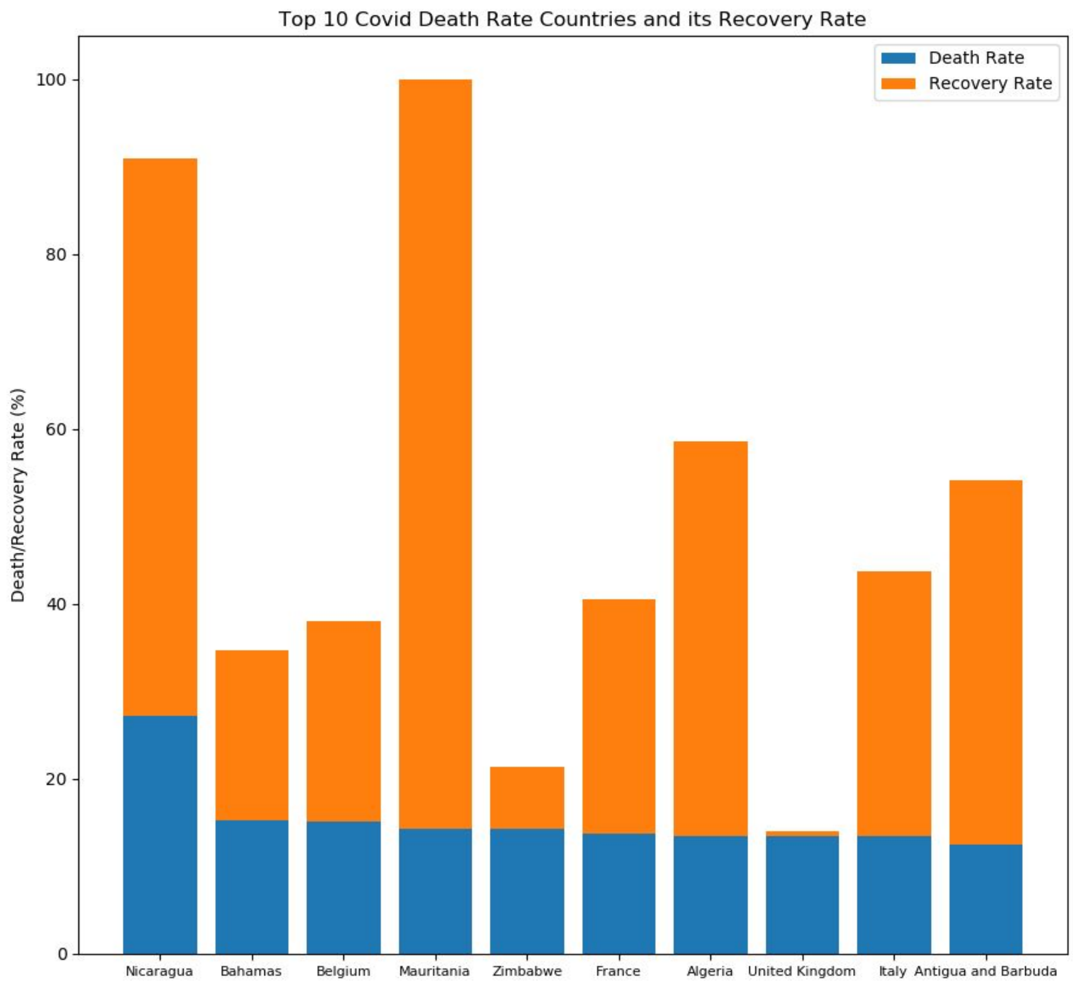


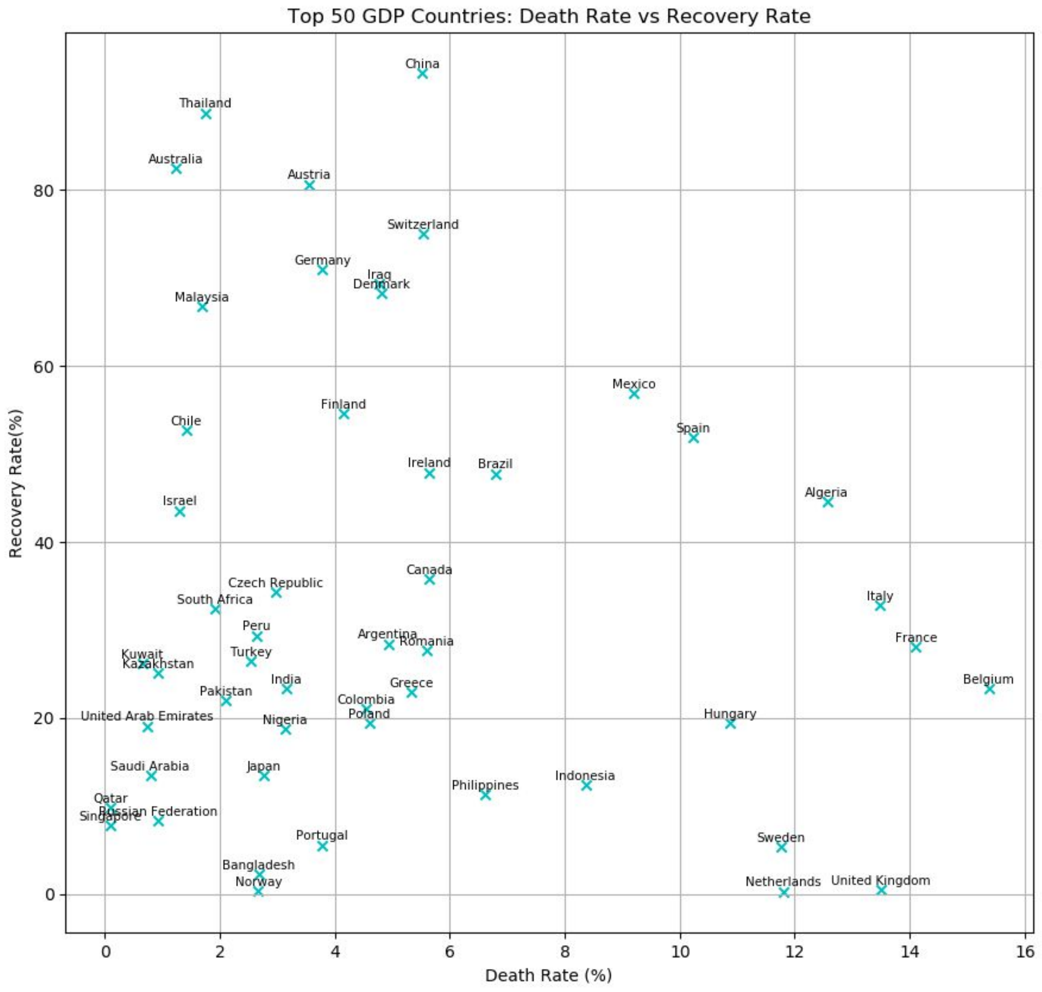
```
Total confirmed Covid cases: 2894581
Total Covid related deaths: 202795
Total recovery from Covid cases: 815948
World Covid related Death Rate: 7.006022633327587
World Covid Recovery Rate: 28.188812128594776
World Average Covid Death Rate: 2.815610835908927
World Average Covid Recovery Rate: 25.78094741004459
World Average Covid Infection Rate: 0.0640648604220256
```

Visualizations:


Covid Death/Recovery Rate vs Population per Country
Recovery Rate
Death Rate
Death/Recovery Rate (%)
Population (in Billions)
1e9
0
20
40
60
80
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4


Top 10 GDP Countries: Covid Recovery Rate
China
Germany
Brazil
Canada
Italy
France
India
Japan
Russian Federation
United Kingdom
Recovery Rate (%)
0
20
40
60
80
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e13
GDP (in Trillions USD)


Top 10 Covid Death Rate Countries and its Recovery Rate
Death Rate
Recovery Rate
Death/Recovery Rate (%)
0
20
40
60
80
100
Nicaragua
Bahamas
Belgium
Mauritania
Zimbabwe
France
Algeria
United Kingdom
Italy
Antigua and Barbuda



Top 50 GDP Countries: Death Rate vs Recovery Rate
Recovery Rate(%)
Death Rate (%)
China
Thailand
Australia
Austria
Switzerland
Germany
Iraq
Denmark
Malaysia
Mexico
Finland
Chile
Spain
Ireland
Brazil
Israel
Algeria
Canada
Czech Republic
South Africa
Italy
Peru
Argentina
Romania
France
Kuwait
Turkey
Kazakhstan
India
Greece
Belgium
Pakistan
Colombia
Poland
United Arab Emirates
Nigeria
Hungary
Saudi Arabia
Japan
Indonesia
Philippines
Qatar
Russian Federation
Singapore
Portugal
Sweden
Bangladesh
Norway
Netherlands
United Kingdom

Instructions:

Getting the data:

- Run **get_covid_data.py** at least 13 times to get the whole data for the Covid table in the database as there are 246 rows in total for 246 countries in the Covid-19 API. On the 13th time of running the data, it should print 'You have reached the end of data', indicating that all of the data has been inserted to the table.
- Run **get_country_indicator.py** at least 14 times to get the whole data for the Country_Indicator table in the database as there are 264 rows in total for 264 data from the Worldbank API. Just like **get_covid_data.py**, it will print on the 14th time of running the data, it should print 'You have reached the end of data', indicating that all of the data has been inserted to the table.
- Run **calculation.py** to get a text file **covid_calculation.txt** with all the calculated values.
- Run **graph.py** to view the graphs.

Documentation of Functions:

get_covid_data.py

1. setUpDatabase
    - Input: database_name (name of the database)
    - Output: No printed output, return cur, conn of the inputted database
    - Creates and set up database, returning cursor and connection to that database, needed in other functions

2. get_data_covid_countries
    - Input: cur, conn of a database
    - Output: No printed output until there is no more data to insert to the table, print 'You have reached the end of data' as an indicator that we have all the data we need from the API.
    - Nothing returned
    - Makes API call from the covid-19 API (url: "https://api.covid19api.com/summary"), decodes the JSON file to a dictionary, insert the data from the dictionary (total confirmed case, total deaths and total recovered of each country). Calculates death rate (100*total deaths/total confirmed) and recovered rate (100*total recovered/total confirmed). Inserts the data from the dictionary and the calculated values for each country to the Covid table in the database, 20 countries at a time (20 data at a time).

3. make_table_covid
    - Input: cur, conn of a database

- Output: If Covid table already exists, print ‘table exists, proceed to gathering data’, nothing returned
- Try creating a Covid table in the database. If an error happens (as what will happen if there already exists a Covid table), it will print ‘table exists, proceed to gathering data’.

4. main
   - Input: No input
   - Output: Printed output from respective called functions
   - Set cur, conn by calling setUpDatabase function with input ‘covid_worldbank.db’, the name of our database. Call make_table_covid with input cur, conn that has been set up before. Call get_data_covid_countries with input cur, conn set up before.

get_country_indicator.py

1. setUpDatabase
   - Input: database_name (name of the database)
   - Output: No printed output, return cur, conn of the inputted database
   - Creates and set up database, returning cursor and connection to that database, needed in other functions

2. get_data_population
   - Input: cur, conn of a database
   - Output: Returns dict_list (list of dictionaries of result) if request url works. If url doesn’t work, prints “Error when reading from url”
   - Makes API call from the World Bank API (url: http://api.worldbank.org/v2/country/all/indicator/SP.POP.TOTL?format=json&date=2018&per_page=500), decodes the JSON file to a dictionary and returns a list of dictionaries of the result.

3. get_data_gdp
   - Input: cur, conn of a database
   - Output: Returns dict_list (list of dictionaries of result) if request url works. If url doesn’t work, prints “Error when reading from url”
   - Makes API call from the World Bank API (url: http://api.worldbank.org/v2/country/all/indicator/NY.GDP.MKTP.CD?format=json&date=2018&per_page=500), decodes the JSON file to a dictionary and returns a list of dictionaries of the result.

4. make_table
    - Input: cur, conn of a database
    - Output: If Country_Indicator table already exists, print ‘table exists, proceed to gathering data’, nothing returned
    - Try creating a Country_Indicator table in the database with 3 columns: country(text), population(int), GDP(int). If an error happens (as what will happen if there already exists a Country_Indicator table), it will print ‘table exists, proceed to gathering data’.

5. insert_data
    - Input: cur, conn of a database
    - Output: No printed output until there is no more data to insert to the table, print ‘You have reached the end of data’ as an indicator that we have all the data we need from the API.
    - No return output
    - Calls get_data_population and get_data_gdp functions and inserts population and gdp data from the dictionary for each country to the Country_Indicator table in the database, 20 countries at a time (20 data at a time).

6. main
    - Input: No input
    - Output: Printed output from respective called functions
    - Set cur, conn by calling setUpDatabase function with input ‘covid_worldbank.db’, the name of our database. Call make_table with input cur, conn that has been set up before. Call insert_data with input cur, conn set up before.

calculation.py

1. setUpDatabase
    - Input: database_name (name of the database)
    - Output: No printed output, return cur, conn of the inputted database
    - Creates and set up database, returning cursor and connection to that database, needed in other functions

2. calculations
    - Input: cur, conn of a database
    - Output: No printed output, return total_dict (a dictionary with all the calculated values)
    - Creates a dictionary to store the calculations that will be made. Select and fetch sum and count of different rows in the Covid table. Calculated world’s total

confirmed covid-19 cases, total deaths, total recovered, world's death rate, world's recovery rate, average death rate and recovery rate of countries in the world. These values are then inserted to the dictionary created earlier, with their respective keywords. Return the dictionary with the calculations.

3. average_infection_rate_world
   - Input: cur, conn of a database
   - Output: No printed output, return average (float)
   - Create a list to store data for future use. Use JOIN and Select some columns from Covid and Country_Indicator table. Fetch all the data selected. Use for loop to convert the string numbers to float, and calculate infection rate (confirmed case*100/population) for each country, and append them to the created list. Calculate average of infection rate of the countries selected from the joined table. Return that average.

4. main
   - Input: No input
   - Output: No printed output
   - Set cur, conn by calling setUpDatabase function with input 'covid_worldbank.db', the name of our database. Called calculations function with input cur, conn that was set up, and assigning the returned dictionary to a variable. Called average_infection_rate_world function with input cur, conn set up before, and assigning the returned float to a variable. Writes the calculations to a text file.

graph.py

1. setUpDatabase
   - Input: database_name (name of the database)
   - Output: No printed output, return cur, conn of the inputted database
   - Creates and set up database, returning cursor and connection to that database, needed in other functions

2. top_50_gdp_dr_rr
   - Input: cur, conn of a database
   - Output: png file of the graph produced
   - Selects top 50 country, death_rate and recoverd_rate columns ranked by country's GDP from the JOIN Covid and Country_Indicator tables and appends them into 3 lists: country, death_rate and recovered_rate. Then, use the data and create a scatter plot with

death rate and recovery rate lists. Then, annotate the data points with the country lists, set labels, titles and screen size. Save the figure and show the plot.

3. top_10_rec_gdp
    - Input: cur, conn of a database
    - Output: png file of the graph produced
    - Selects top 10 country, recovered_rate and GDP columns ranked by GDP from the JOIN Covid and Country_Indicator tables and appends them into 3 lists: country, gdp and recovered_rate. Then, use the data and create a scatter plot with gdp and recovery rate lists. Then, annotate the data points with the country lists, set labels, titles and screen size. Save the figure and show the plot.

4. top_10_death_rec_rate
    - Input: cur, conn of a database
    - Output: png file of the graph produced
    - Selects top 10 country, recovered_rate and death_rate columns ranked by death rate from the Covid table and appends them into 3 lists: country, death_rate and recovered_rate. Then, use the data and create a bar graph with country and death rate lists. Repeat and create another bar graph with country and recovered rate lists with the death rate bar graph in the bottom and stack them. Then, set ylabel, titles, legend and screen size. Save the figure and show the plot.

5. tr_rr_vs_population
    - Input: cur, conn of a database
    - Output: png file of the graph produced
    - Selects country, recovered_rate, death_rate and population columns from the JOIN Covid and Country_Indicator tables and appends them into 3 lists: population, death_rate and recoverd_rate. Then, use the data and create a scatter plot with population and recovery rate lists. Repeat and create another scatter plot with population and death rate lists. Then, set labels, titles, legend and screen size. Save the figure and show the plot.

6. main
    - Input: No input
    - Output: png files from each respective called functions
    - Set cur, conn by calling setUpDatabase function with input ‘covid_worldbank.db’, the name of our database. Call top_50_gdp_dr_rr with input cur, conn that has been set up before. Call top_10_rec_gdp with input cur,

conn set up before. Call top_10_death_rec_rate with input cur, conn set up before. Call dr_rr_vs_population with input cur, conn set up before.

Resource Documentation:

Date	Issue Description	Location of Resource	Result
4/10/2020	We need to pull Covid-19 data per country to answer our research question.	https://api.covid19api.com/summary	We were able to pull the necessary data and use it to create tables and produce calculations and graphs.
4/10/2020	We need a request url to pull population and GDP data per country to answer our research question.	https://datahelpdesk.worldbank.org/knowledgebase/articles/889392-about-the-indicators-api-documentation	Used this resource to read the documentation of the API and the format for the request url to the World Bank API.
4/25/2020	Finding a starter code to make a stacked bar chart for top 10 death rate countries.	https://matplotlib.org/3.1.1/gallery/index.html	Were able to use the starter code and modify it to produce a stacked bar chart with death/recovery rate stacked.
4/25/2020	Annotating the data points with a country name.	http://queirozf.com/entries/add-labels-and-text-to-matplotlib-plots-annotation-examples	Were able to annotate each data point in the scatter plot with a country name using a loop that zips all three lists and labels each point.
4/25/2020	Getting the top number of rows for an indicator (for example, top 50 GDP countries) in Sqlite since using "TOP" doesn't seem to work. Need this because there are too many data points to make the graph any useful.	https://www.sqlitetutorial.net/sqlite-limit/	Used the "LIMIT" clause to select the top "X" number of rows to be used for the visualization task. Were able to produce a graph that answers the question more clearly with less "cluster" in one particular area.