SI 206 Final Project Documentation

Alex Chuang, Alex Mah, Kevin Song

Github Repository: https://github.com/konneech/SI206-FinalProject.git

The goals for your project (10 points)

For this project, we set goals related to creating significance out of the project and available resources. Our first goal was to establish connections between three separate databases. Another goal for this project was to make meaningful calculations using our database table. Finally, we made it a goal to analyze and consider takeaways from our calculations and visualizations.

The goals that were achieved (10 points)

For our first goal, we first looked for sources where we could pull databases. API Ninjas is a website that features many APIs, from which we found their countries, COVID-19, and air quality APIs to be of interest. Combining these three, we wanted to connect COVID-19 cases to carbon emissions, and organizing that information by countries alongside country statistics. This goal of finding connections between APIs was also met through conducting some outside research on related topics, such as researching lower carbon levels in recent years or global COVID-19 cases trends.

For our second goal, we wanted to make calculations that highlight the connection made in our first goal. By calculating air quality index to average population size, we created a system that rates countries into categories, ranging from good to very unhealthy. Additionally, comparing internet users to statistics such as percentage of population that had COVID-19 and the life expectancy gaps between genders gives additional insight to how different countries dealt with the global pandemic, how countries of different economic and gender culture statuses were impacted by COVID-19, among other takeaways. Through our calculations and visualizations, there are several interpretations that our audience can use to further their understanding of the global pandemic and its impact on worldwide air quality.

For our last goal, it was important for our presentation to analyze our visualizations, and think about takeaways, correlations, and conclusions. Thinking about how are data might have inconsistencies, margin of error, and the reasons for that were important to understanding our results. Factors like immigration and travel, population density and growth, access to medicak technology and technological development, are just a few factors that were not included in our

APIs and therefore our calculations and visualizations. As a result, our findings are not a completely accurate representation, but rather an insightful perspective of statistical correlations of the globalization of COVID-19 and global AQI.

The problems that you faced (10 points)

Initially, our original plan was to use different databases for this project. Our first plan was to use databases on different videogame playerbases to compare how each game determines player skill and rank. While we had a clear vision for a scope of this plan, upon implementation and execution, it became difficult to correlate different ranking systems, and how each game fundamentally trains player skill. For example, playing an online board game like chess is fundamentally different from playing a first-person shooter like Valorant. As a result, our project made a full pivot and we decided to utilize new databases: country statistics, COVID-19 cases, and air quality.

After redrafting our plan, some problems with the countries API were faced. Since the country API only outputs five countries for one the urls, it was necessary to change the data in order to access one hundred different countries. This was accomplished by manually adding a countries list with one hundred entries. Next, the runtime of the countries database was slow due to its size, so it was necessary to split it into four which resulted in a faster runtime.

Finally, revisiting lecture slides assisted the completion of creating charts with Matplotlib.

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

Gender gap in life expectancies

https://drive.google.com/file/d/1o4y4Hp4QZAXQQNma-oc3oGZ4X4Vbx5De/view?usp=sharing

Percentages of COVID-19 cases in population to percentage of internet users in population

https://drive.google.com/file/d/1jZUDsw7EHzeeAx8N16rnj7RDj7Qy5Jnv/view?usp=sharing

AQI to country population

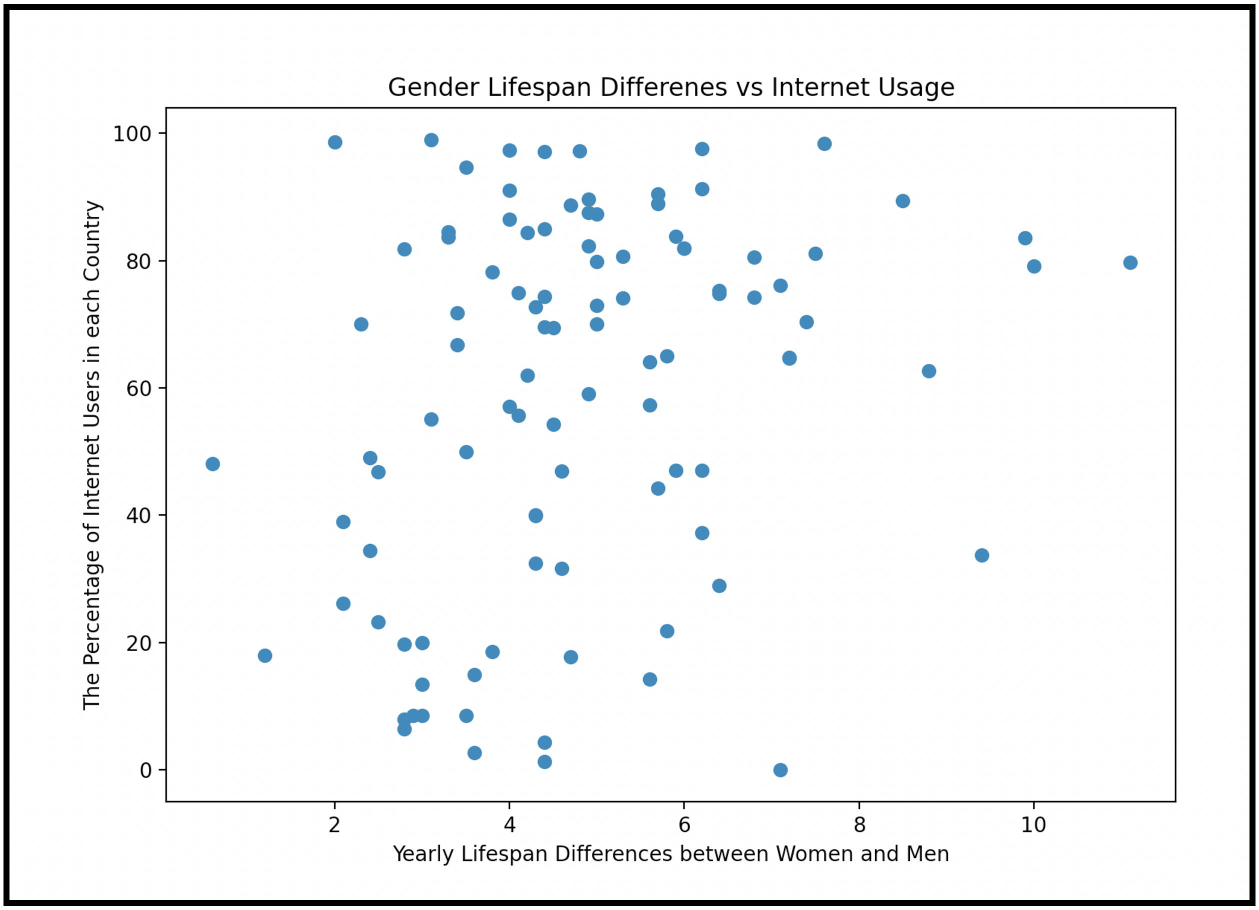
https://drive.google.com/file/d/159E47c1r1wUmR-CfC9smtrrfjLn2p2L0/view?usp=sharing

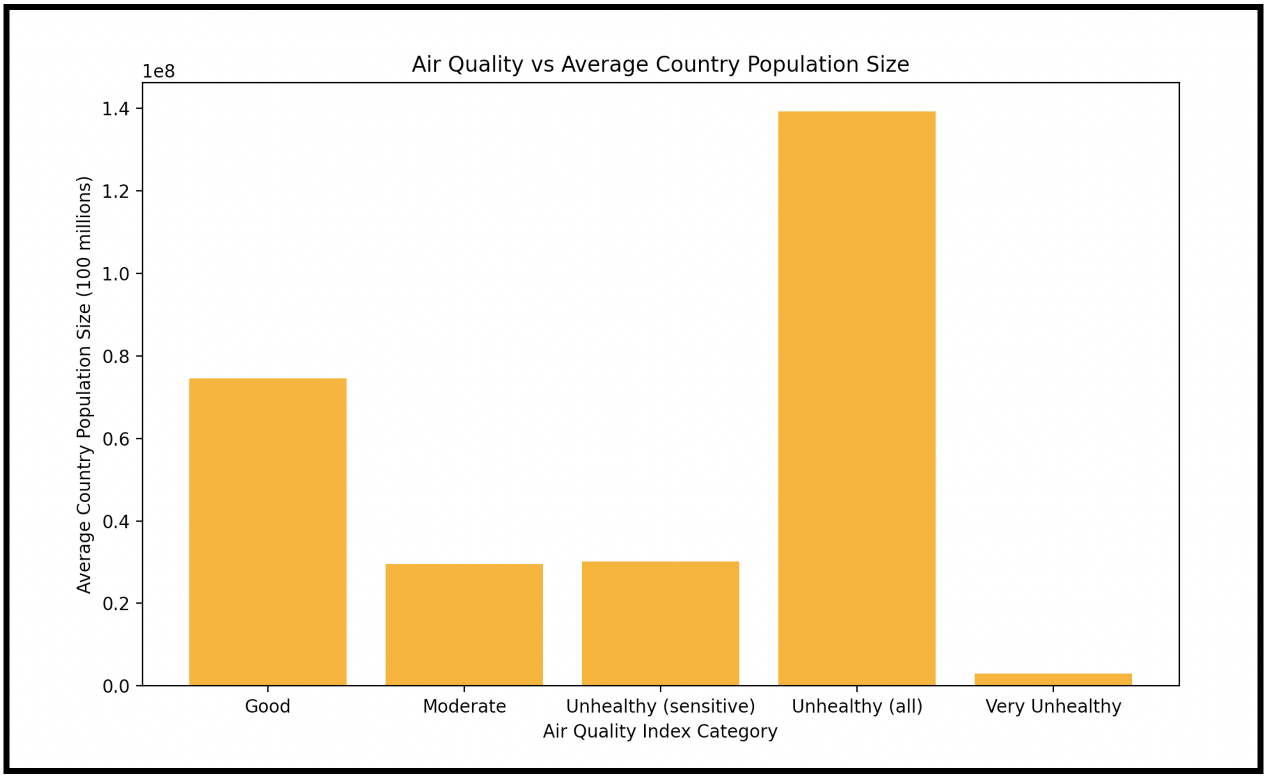
In this project, three main calculations were conducted. The first calculation compares the gap in life expectancy between genders to internet users. Since we already calculated the life expectancy gender gap, we then used a scatterplot to compare the two factors, the x-axis being “Yearly Lifespan Differences between Women and Men” and the y-axis “The Percentage of Internet Users in Each Country.”

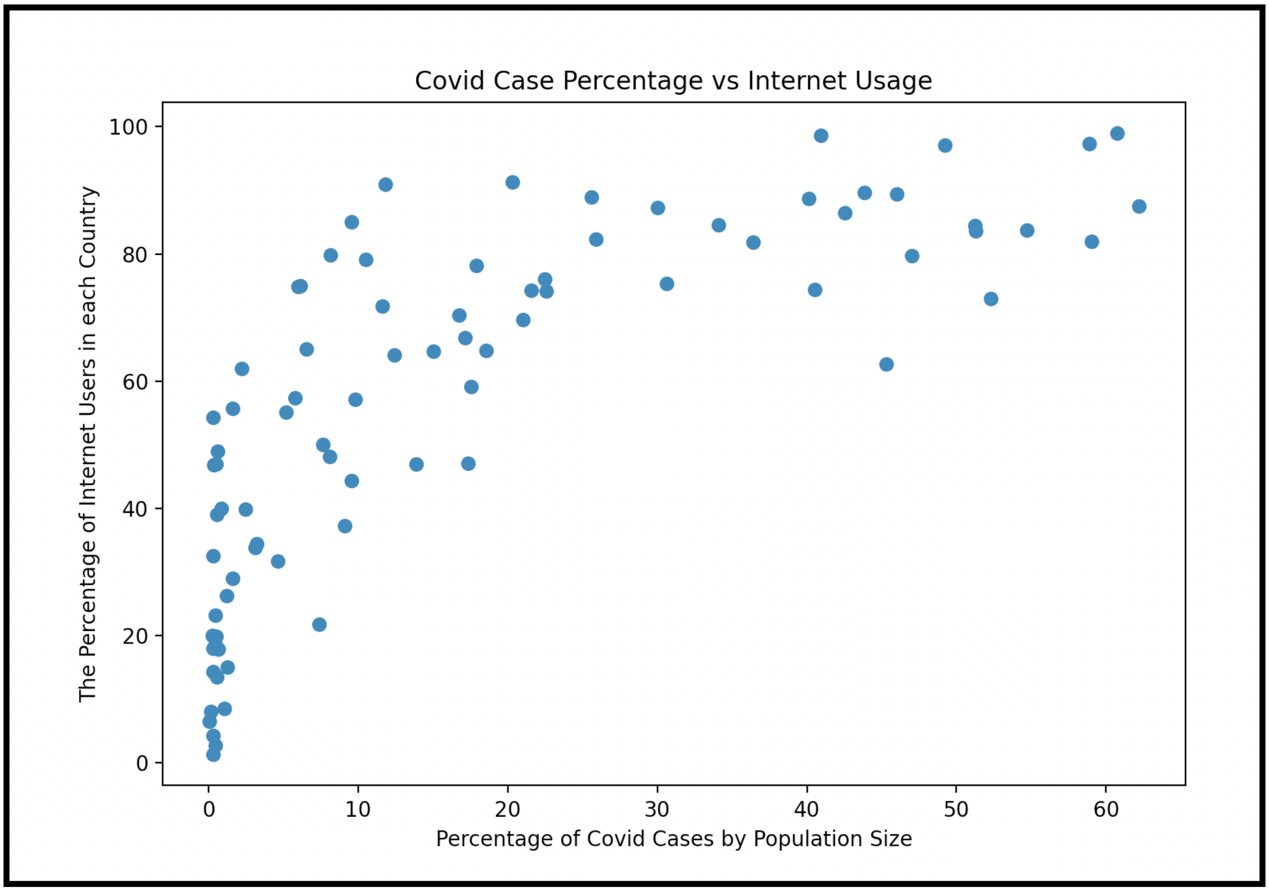
Our next calculation compares the percentage of the population that fits into the total number of COVID-19 cases to the percentage of the population that are internet users. Using the internet users statistic from our first calculation and plotting that in relation to each country’s COVID-19 cases population on a scatterplot, it was a logical choice to compare population percentages.

Finally, for our last main calculation, we used a bar graph to showcase different ratings for AQI (air quality index) based on the population size of a country. By categorizing this calculation on a scale of zero to three hundred, we could conclude if a country’s overall air quality is “good” to “very unhealthy,” with respect to the size of the country’s population. In this calculation, it was important to include population as opposed to using a country’s geographical size.

The visualization that you created (i.e. screen shot or image file) (10 points)


Gender Lifespan Differenes vs Internet Usage
The Percentage of Internet Users in each Country
0
20
40
60
80
100
2
4
6
8
10
Yearly Lifespan Differences between Women and Men


Air Quality vs Average Country Population Size
1e8
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Average Country Population Size (100 millions)
Good
Moderate
Unhealthy (sensitive)
Unhealthy (all)
Very Unhealthy
Air Quality Index Category


Covid Case Percentage vs Internet Usage
100
80
60
40
20
0
The Percentage of Internet Users in each Country
0
10
20
30
40
50
60
Percentage of Covid Cases by Population Size

Instructions for running your code (10 points)

1. Run country.py 4 times to add the country data table and the gap_expectancies table to the database.
    a. This also creates the gap expectancies txt file for the first calculation
2. Run airquality.py 4 times to add the air quality data table to the database.
3. Run covid.py 4 times to add the covid data table to the database.
4. Run visualizations.py to get the visualizations.
    a. This also creates the aqi vs population and covid percentages txt files for the 2nd and 3rd calculations.

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

country.py

1. get_data(country)
    a. Purpose: gathers all of the data from the Country API for one specific country. This data includes lots of quantitative data about the country. It then puts the data into a json format and returns it.
    b. Functionality: It takes in the request url for a specific country, loads the data into a response, converts the data into json form, and returns it. If getting the url fails, the function returns none.
    c. Input: country list
    d. Output: json country data from the API

2. create_poopulation_table(countries, cur, conn)
    a. Purpose: To create a table of quantitative data from the Country API.
    b. Functionality: It takes in a list of countries. It first creates a table called country_data with the appropriate columns. It then goes through each country from the country list, 25 at a time, and adds the country and its respective quantitative data to the table.
    c. Input: country, cur, conn
    d. Output: a new database table in country.db called country_data

3. create_gap_expectancies_table(cur, conn)
    a. Purpose: To create a table of the difference of lifespans between male and female for each country.
    b. Functionality: It takes the data from the country_data in the database and calculates the difference between female_life_expectancy and male_life_expectancy and adds it to a new table.
    c. Input: cur, conn
    d. Output: a new database table in country.db called lifespan_gaps

4. main()
    a. Purpose: Running the create_population_table function, and the create_gap_expectancies_table function
    b. Functionality: Sets up the connection to the country.db database and holds the list of countries. It then runs the create_population_table function with the list of 100 countries and prints out that it added 25 rows to the database.
    c. Input: none
    d. Output: 'added 25 rows of both tables to database'

airquality.py

5. get_data(country)
    a. Purpose: gathers all of the data from the Air Quality API for one specific country. This data includes quantitative data about a country's air quality, such as AQI. It then puts the data into a json format and returns it.
    b. Functionality: It takes in the request url for a specific country, loads the data into a response, converts the data into json form, and returns it. If getting the url fails, the function returns none.
    c. Input: country list
    d. Output: json air quality data from the API

6. create_air_quality_table(countries, cur, conn)
    a. Purpose: To create a table of quantitative data from the Air Quality API.
    b. Functionality: It takes in a list of countries. It first creates a table called air_quality with the appropriate columns. It then goes through each country from the country list, 25 at a time, and adds the country and its respective quantitative data to the table.

   c. Input: country, cur, conn
   d. Output: a new database table called air_quality in country.db

7. main()
   a. Purpose: Running the create_air_quality_table function
   b. Functionality: Sets up the connection to the country.db database and holds the list of countries. It then runs the create_air_quality_table function with the list of 100 countries and prints out that it added 25 rows to the database.
   c. Input: none
   d. Output: 'added 25 rows to database'

covid.py

8. get_data(country)
   a. Purpose: gathers all of the data from the Covid-19 API for one specific country. This data includes quantitative data about a country's covid data, such as total cases. It then puts the data into a json format and returns it.
   b. Functionality: It takes in the request url for a specific country, loads the data into a response, converts the data into json form, and returns it. If getting the url fails, the function returns none.
   c. Input: country list
   d. Output: json covid data from the API

9. create_covid_table(countries, cur, conn)
   a. Purpose: To create a table of quantitative data from the Covid-19 API.
   b. Functionality: It takes in a list of countries. It first creates a table called covid_data with the appropriate columns. It then goes through each country from the country list, 25 at a time, and adds the country and its respective quantitative data to the table.
   c. Input: country, cur, conn
   d. Output: a new database table called covid_data in country.db

10. main()
   a. Purpose: Running the create_covid_table function

b. Functionality: Sets up the connection to the country.db database and holds the list of countries. It then runs the create_covid_table function with the list of 100 countries and prints out that it added 25 rows to the database.

c. Input: none

d. Output: ‘added 25 rows to database’

visualizations.py

11. gender_gap_vs_internet_users(cur)

    a. Purpose: calculating the gap between lifespans of women and men. It then creates a scatter plot to see the correlation between country internet usage and gender lifespan gaps.

    b. Functionality: Pulls data from lifespan_gaps and country_data. Pulls the lifespan differences and the internet usage columns and adds them into lists. It then plots each country on a scatter plot.

    c. Input: cur

    d. Output: a scatter plot

12. covid_vs_internet_users(cur)

    a. Purpose: calculating the percentage of covid cases per population size for each country. It then creates a scatter plot to see the correlation between the covid-19 rate and internet usage for each country.

    b. Functionality: Pulls data from covid_data and country_data. Pulls the total_covid_cases, population and internet_users columns and adds them into lists. It removes all of the invalid data from each of the lists and calculates the percentage of covid per population size. It writes the calculations to a text file and then plots each country on a scatter plot.

    c. Input: cur

    d. Output: a scatter plot

13. population_per_aqi_category(cur)

    a. Purpose: calculating the average population size of all countries in a specific aqi category. It then creates a bar graph to show which aqi category has the highest average population size.

    b. Functionality: Pulls data from country_data and air_quality. Pulls air_quality_index column and adds them into lists, depending on what AQI category they fall under. It then loops through each of those lists and calculates the average population size for each category by pulling the population column from country_data. It then writes the calculation to a text file and creates a bar graph that illustrates the data.

c. Input: cur
d. Output: a bar graph

14. main()
    a. Purpose: Running all of the visualization functions
    b. Functionality: Sets up the connection to the country.db database and runs all of the visualization functions.
    c. Input: none
    d. Output: none

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**	**Result**
11/29	Had questions about the project and project expectations.	Lecture on example projects	Clearer understanding of project, better progress and began to reconsider our APIs
12/4	Change project scope, hard to find connections between APIs in original project plan.	Online websites, API list from class	Online research for APIs, we also had to rethink our plan for the project
12/6	Finding new source for APIs.	API Ninjas website	We found three APIs to use (countries, air quality, and Covid-19)
12/9	The runtime of inserting country_data into the database is too large.	Office Hours	I am going to keep my data the same, but I need to split up the amount that I put into the table each time I run the code.

12/9	Forgot how to create a chart through matplotlib.	Lecture Slides	Figured it out and successfully created a graph
12/10	Kept getting list assignment index out of range errors	Online websites	Figured out that I was deleting elements in succession, and then trying to access them later. Changed my code and fixed the error.
12/11	Drafting/preparing presentation of project	Online websites and articles, individual research on COVID-19, carbon emissions, world economy, etc.	We wanted to talk about takeaways from our project, which required us to look into external resources on relevant topics. We conducted our own research, and considered interpretations and connections to our project.