**Project Milestone III: An Application for Infectious Disease Analysis Final Report**

**Isabelle Katz**

**HCIP 5376 Dr Stukes**

**08Dec2023**

**INTRODUCTION**

**Project Goal**

The goal of this project was to develop a Python application that can predict medical staffing needs for influenza patients based on the time of year and population demographics. This application can assist healthcare providers and public health agencies in efficiently allocating resources and medical personnel to effectively manage influenza outbreaks, thereby improving patient care and reducing the burden on healthcare facilities.

**Vested Parties**

- Healthcare providers, including hospitals, clinics, and urgent care centers.

- Public health agencies responsible for managing and responding to influenza outbreaks.

- Medical staff who need to plan their schedules and resources for influenza patient care.

- The general population, as a more efficient healthcare response can improve the overall public health (CDC.)

**Problem Statement**

Managing medical staffing during influenza outbreaks is challenging due to the dynamic nature of the disease. The staffing requirements can vary significantly based on factors such as the time of year, the severity of the influenza strain, and the demographics of the affected population (OSHA.) Existing approaches often lack the accuracy needed for efficient resource allocation, leading to potential shortages or overstaffing, which can impact patient care and healthcare costs (Drummond 2023.)

**METHODS**

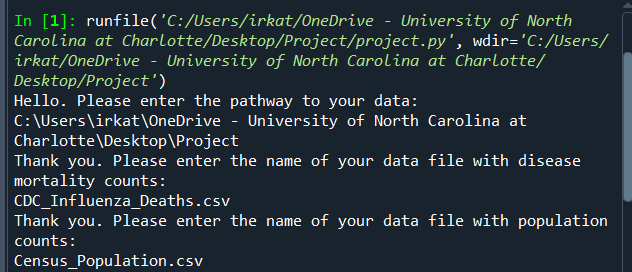
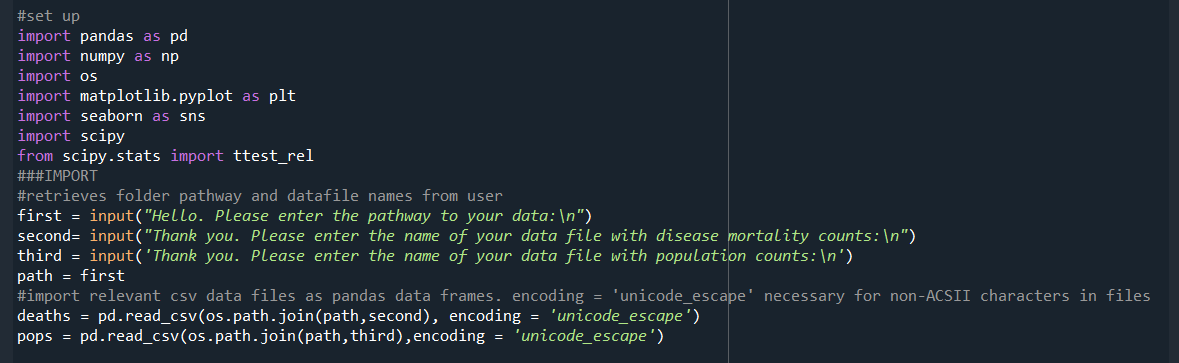
This application uses historical mortality data from the CDC to determine the temporal nature of yearly influenza outbreaks. Using CDC influenza mortality and vaccination data, as well as the US government census, it identifies vulnerable populations (age related, vaccination status) and supports the necessity for additional staffing based on the geographic distribution of these populations. Finally, it determines states’ relative staffing needs based on demographic makeup.

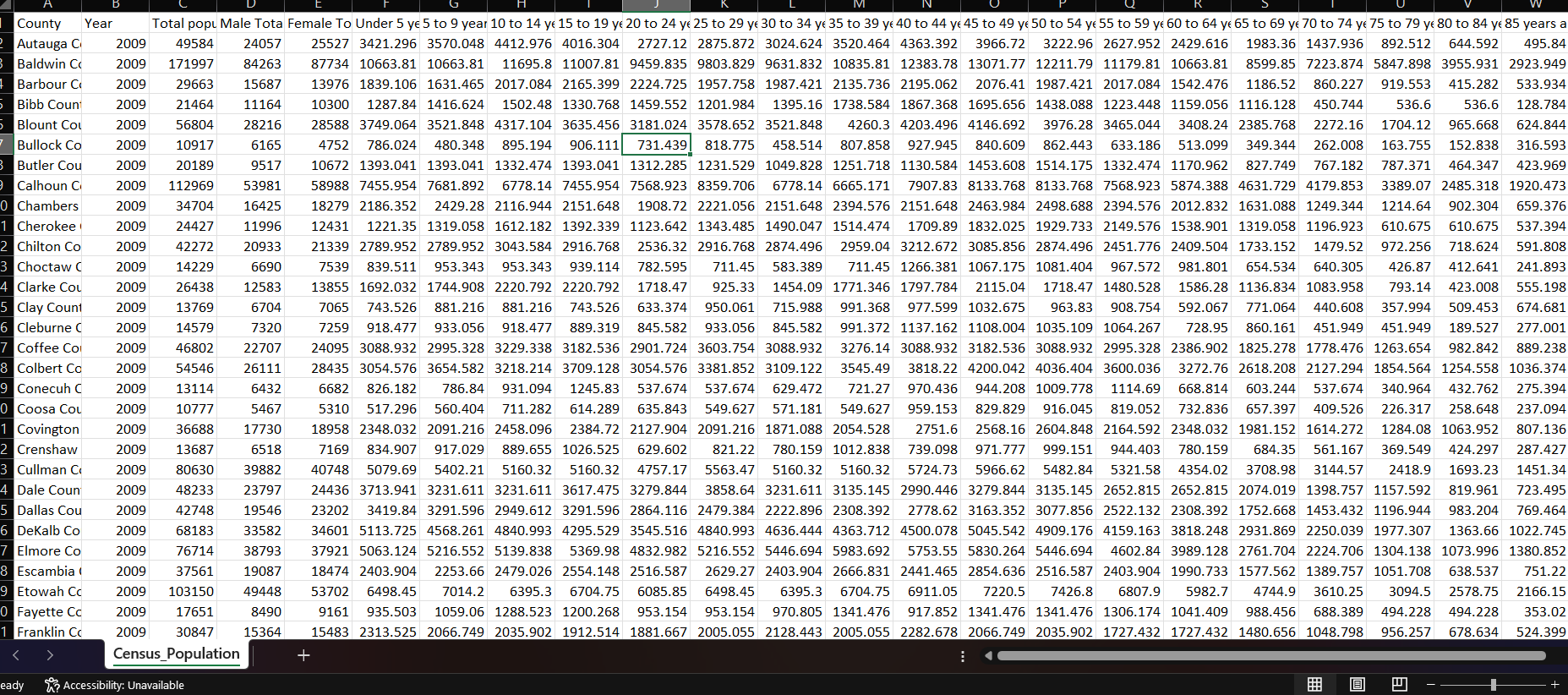
Creating the application required the following steps:

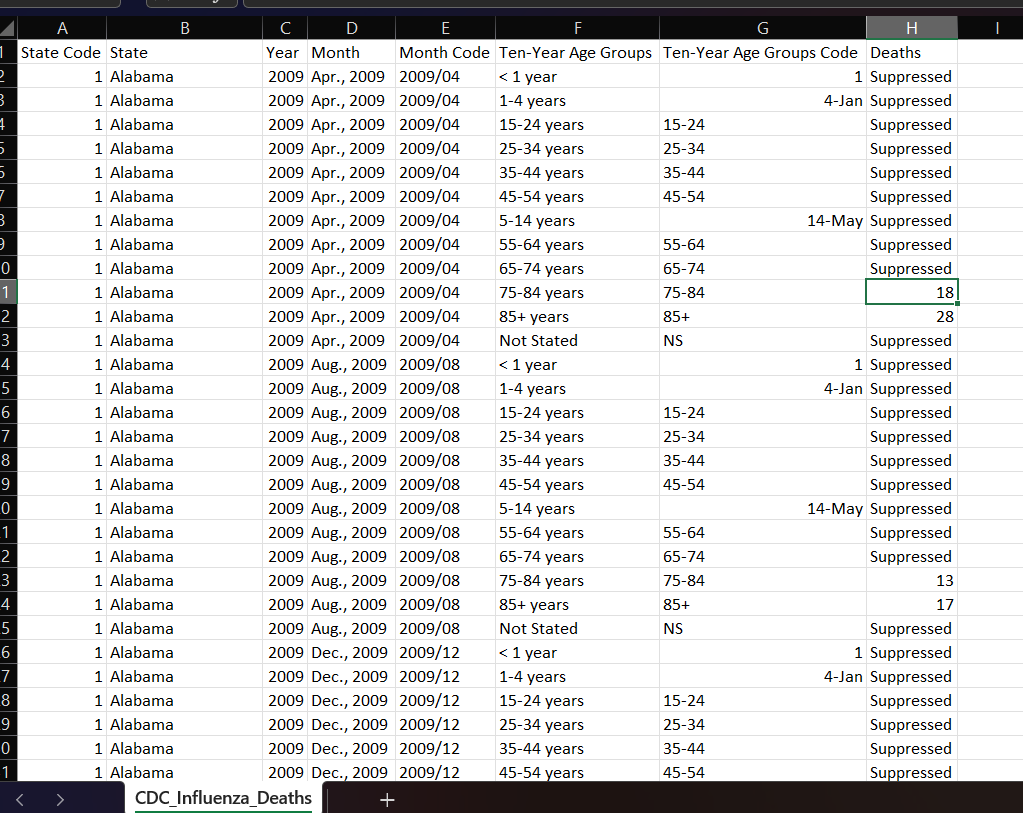
**1. Data Collection and Application Set Up:**

The initial set up of this program involves importing necessary libraries and dataframes. The user inputs the pathway to the location where the original data is and exports will be saved. They also provide two csv files containing the necessary information for analysis.

Historical influenza mortality data (CDC WONDER) and historic census data (US Census) was gathered. I chose these data sets because they were easily accessible, required a reasonable amount of memory to work with, and they were similarly structured. Although I originally planned to use data from the 2018-2019 flu season, the information was reported slightly differently, and 2012-2017 felt like enough for this project.

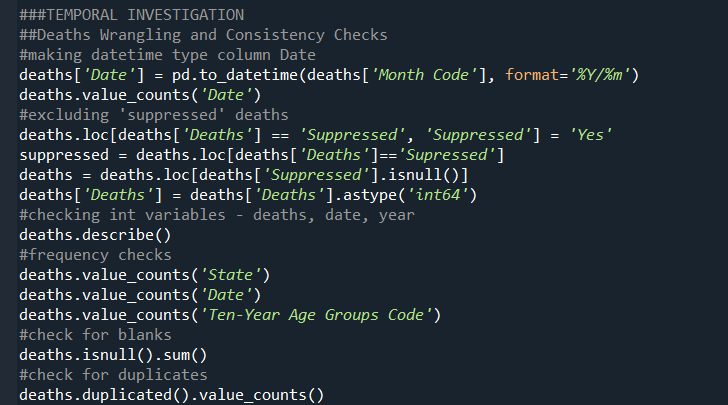






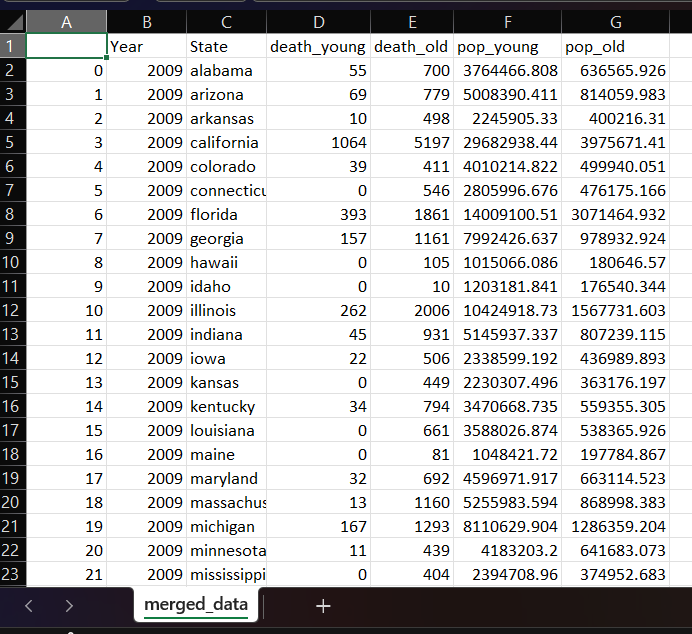
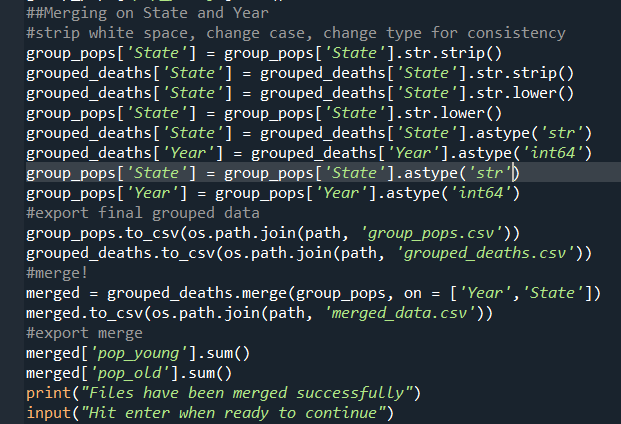
**2. Data Preprocessing:**

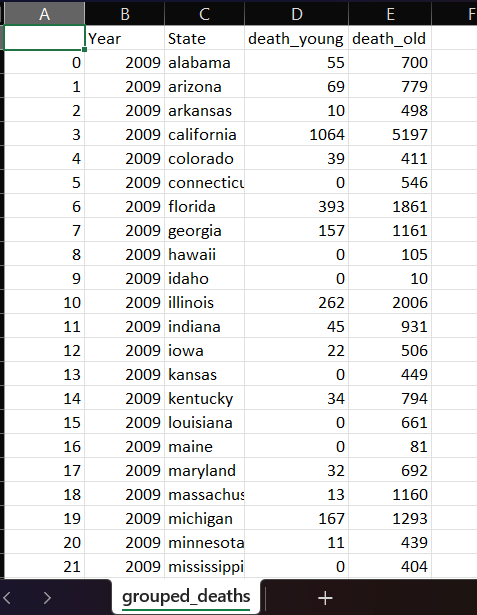
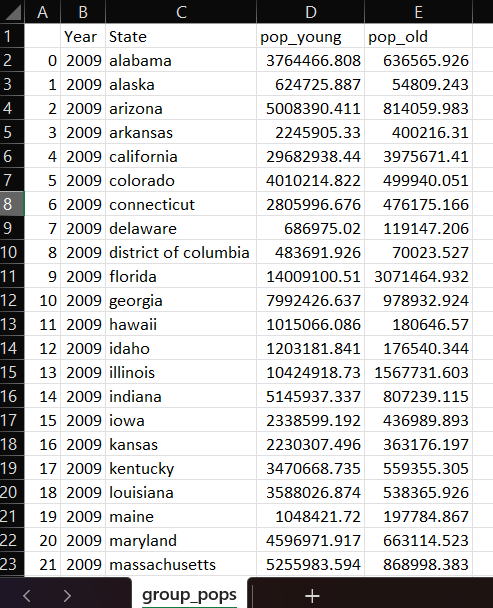
The program runs consistency checks and wrangles the data appropriately for a time based analysis. The biggest step here is removing the rows with insufficient data - in this case where deaths are ‘suppressed’.



Most of my time developing this application went to wrangling the data sets in preparation to be merged together and analyzed.

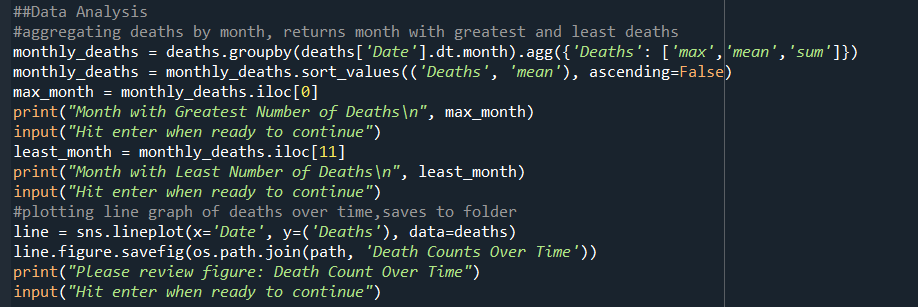






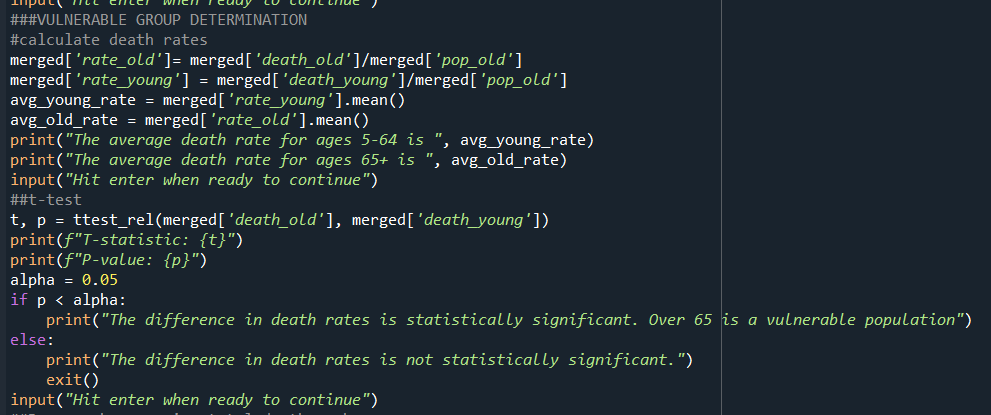
**3. Temporal Analysis**:

To investigate the annual pattern of flu epidemics, the program provides the months with the greatest and least number of deaths. This determines the ‘peak’ of flu season when the most help is needed. To confirm and visualize the annual pattern a graph of deaths over time is provided. The datetime type pandas has was very helpful in this section.

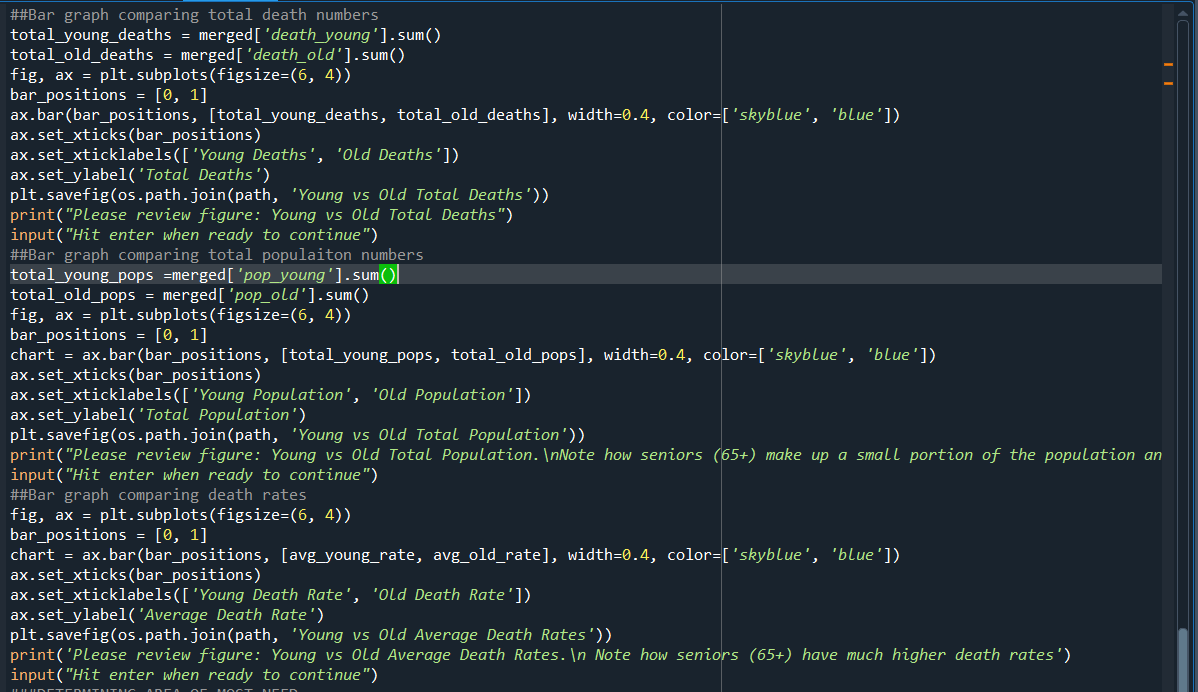


**4. Vulnerable Population Determination:**

With the merged data, the program calculates and performs a t-test on the young and old population death rates. I chose a t-test because I am comparing averages of two groups that have a fairly normal distribution. If the null hypothesis that there is no difference between the groups is supported the program will end since further analysis depends on 65+ being a vulnerable population.

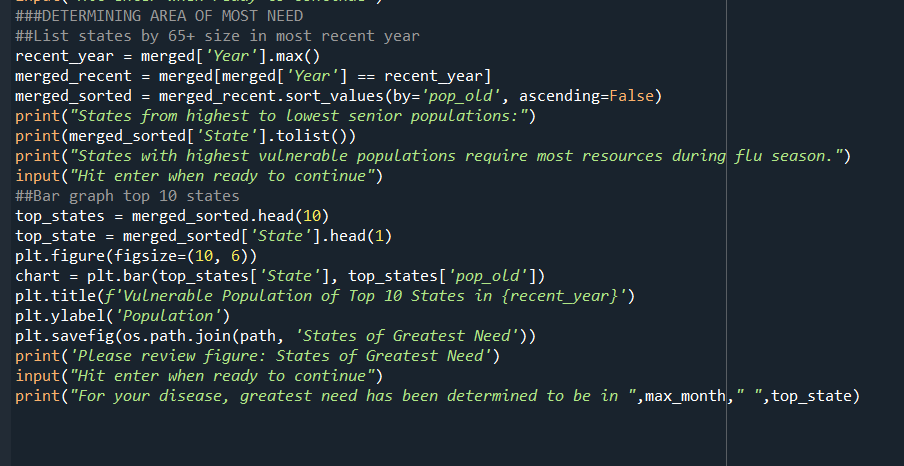


It then creates three bar graphs that illustrate the differences in groups and supports seniors as a vulnerable population.



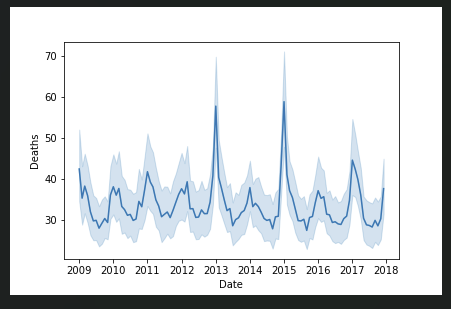
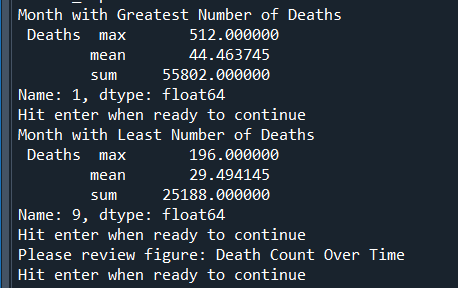
**5. Spatial Modeling:**

Finally, the application determines areas of greatest need based on vulnerable population size. It reports states in order of 65+ population size and creates a graph to demonstrate the top ten.

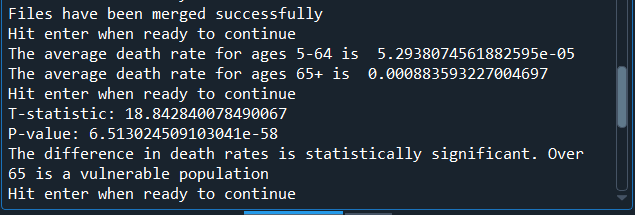


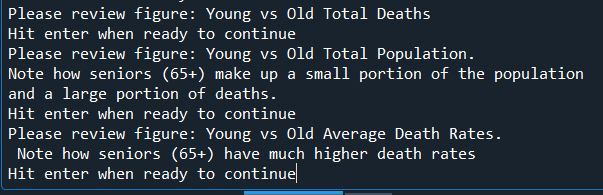
**RESULTS**

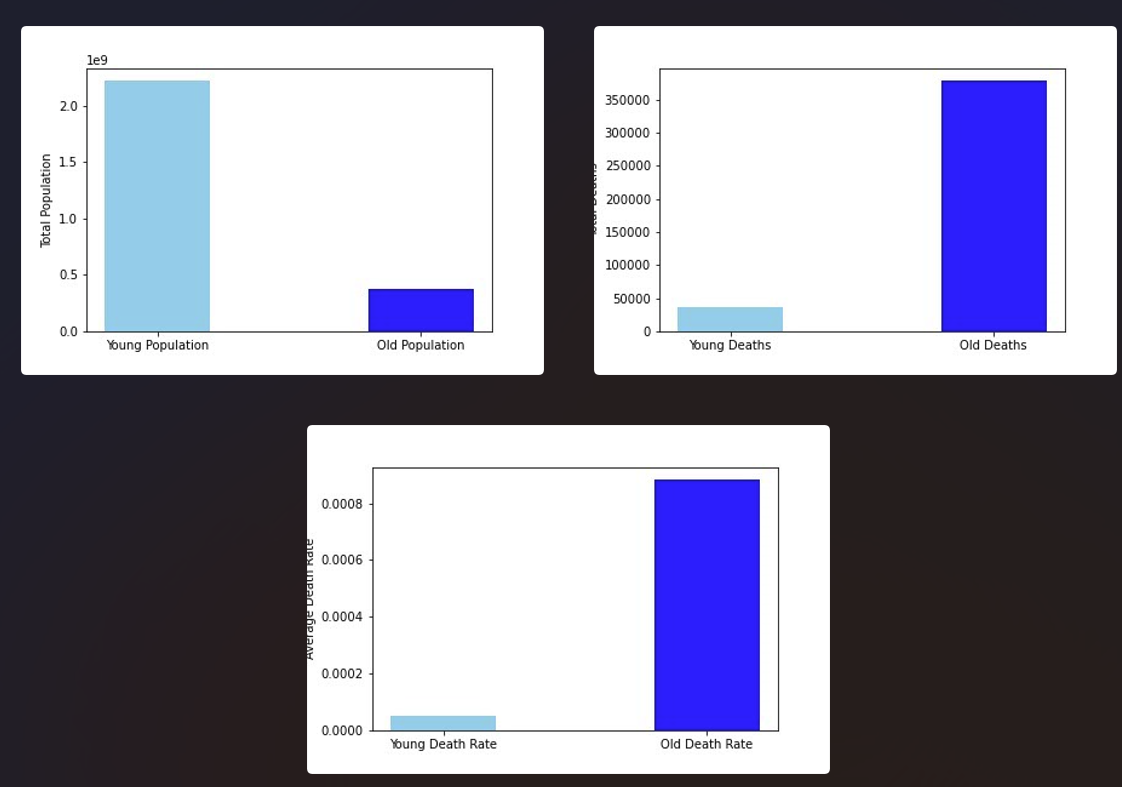
For the temporal aspect, the program outputs a description of the months with the greatest (January) and least (September) number of deaths, including the max, mean, and totals from 2009-2018. It also creates and saves a figure charting death counts over time, so the seasonality and annual pattern can be confirmed by the user.



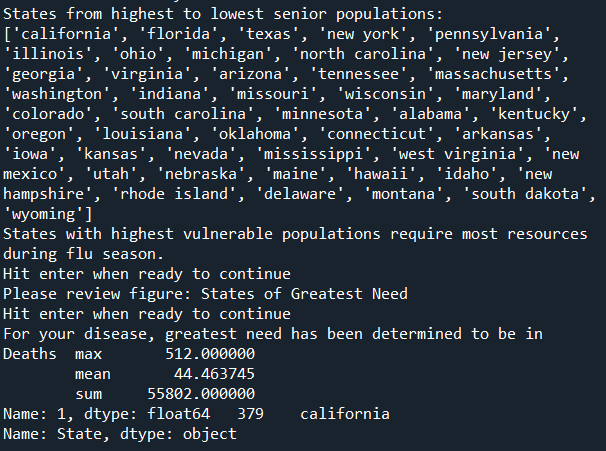
The rest of the analysis assumes that people over 65 are more likely to die of the flu than people 5-64. To support this, the program prints the average death rates of the two groups and the p-value from a t-test, as well as a statement confirming the difference is statistically significant. To visually add to the argument, bar graphs are created and saved that emphasize how people over 65 make up a small portion of the population, but a larger percentage of deaths from influenza.

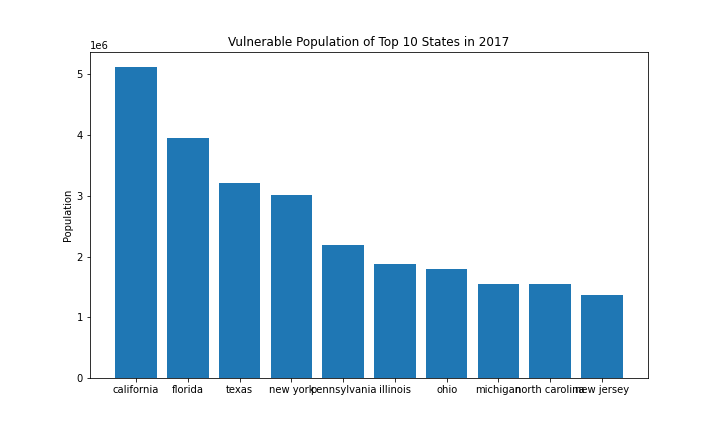






Concluding the analysis by applying the previously determined vulnerable population characteristics, the program prints a list of states in order of their 65+ population size. This corresponds to the greatest need during influenza season, as those over 65 are more likely to become severely ill, require medical attention, and die. A bar graph depicts the top ten greatest need states and the sizes of their 65+ population.



****

**DISCUSSION**

To determine the accuracy of the program in determining time and location of greatest need, I compared the top 10 states predicted to have the most flu deaths, and the month predicted to have the most flu deaths, with actual data from 2018.

| **Predicted Most Deaths** | **Actual Most Deaths** |
| --- | --- |
| January | January |
| California | California |
| Florida | New York |
| Texas | Texas |
| New York | Florida |
| Pennsylvania | Pennsylvania |
| Illinois | Illinois |
| Ohio | Ohio |
| Michigan | North Carolina |
| North Carolina | Michigan |
| New Jersey | Tennessee |

Excitingly, the program was fairly accurate in picking the states and month with the greatest influenza deaths. The month is correct, 50% of the states are in the right order, and only one was not in the top ten.

There are many other factors determining flu death rates in an area, such as population densities, additional vulnerable groups, politics, and socioeconomics - so I was surprised the state results matched up so well.

The seasonality of flu is very predictable, so I was more confident in that answer.

**CONCLUSIONS**

My initial plan was to create multiple functions that would be called with user interaction. However I ran into some challenges with using pandas dataframes with the functions. Although functions can act on and change the original dataframe, any new ones created within the function would have to be returned to be referenced again in the main script or future functions. This added a level of confusion, repetition, and difficulty debugging that I was unable to overcome, so I ended up with one continuous script.

**Data Restrictions and Limitations:**

All data has limitations. Bias and error can be found in automatic and manual collection, and can be done intentionally or unintentionally. The large amount of ‘suppressed’ death counts eliminated a potential vulnerable group, under 5 year olds, and also could have conferred additional bias. Particularly for health care data, bias can have great impacts on data analysis results and who those results can be applied to. For example, high income people are more likely to have access to healthcare, and are therefore more likely to go to the hospital to be treated. Their death would be recorded as influenza related and counted, whereas a lower income person may die at home, without a positive test result, and not be included. For census data, undocumented immigrants may report inaccurately to protect themselves and their families.

These two small examples are multiplied as data sets grow, and often reflect systematic disparities in health care. If these biases are not carefully considered and investigated, incorrect conclusions with potentially harmful consequences could be drawn.

**Programming Restrictions and Limitations:**

As I only started learning python and pandas a few months ago, I was restricted by the amount of time it took to complete simple tasks. Basic data manipulation required research, and I struggled with the syntax of the graphing features. Therefore the application is not as pretty or robust as I would have liked.

My original intention to use functions for more flexibility didn’t work out, making the program much less versatile. I think if I had planned and investigated how to use data frames with functions I could have created something neater. I definitely need to work on remembering to create new databases as opposed to overriding old ones and better naming schemes. This would have assisted me in creating the functions I wanted.

The program is not very user friendly. Typing in the path and file names must be exact. The files also have to be set up the same way the ones I used were - with the same columns and data structure. The program could be adapted to different data sets through more wrangling and user interactions. For example it would be good to provide reports of the data manipulation steps - such as “There are [3] duplicated rows. Would you like to delete them? Y/N”. There were no duplicates in this data, but this could similarly be done for missing values, which were present in the deaths data and were removed. It would also be smart to report the percentage of duplicates and missing values so the user is aware of limitations.

I would like to have more justification for the annual pattern, such as comparing each month year to year to check for consistency, and maybe having a threshold of standard deviation that would return “No pattern found”. For full functionality it would also have to test for and analyze different patterns of diseases. The output of the greatest and least months is also not very intuitive, but I ran into some issues with the datetime and aggregations.

Investigating more potential vulnerable populations would be interesting. Although far beyond my capabilities, machine learning algorithms are able to pick up on patterns unnoticed by people, potentially revealing additional vulnerable groups to assess. I planned on looking at both under 5 and over 65 age groups as vulnerable, but unfortunately did not have the death counts for the youngest age group.

A map would be the most effective way of presenting the area data. I also might have looked at different population densities as opposed to population size. The program also does not output any hard numbers, which would be the most helpful in making difficult decisions about determining resources to minimize deaths during flu season.

**REFLECTION**

I really enjoyed the structure of this class. I think the videos and text book were clear and the exercises effective in learning and practicing the skills we were taught. The opportunity to create and develop an independent project was really fun and exciting to end the semester with a relevant and working deliverable.

I do have prior coding experience in Java, C++, Matlab, R, and SQL, and had previously worked with Pandas dataframes. This definitely helped me understand the syntax in the context of what it could be used for. Although pandas was extremely useful in my project, I would be interested to see if I could create something similar using only the methods we learned this semester.

I enjoy coding projects and data analysis and will continue to improve my skills and knowledge in both areas. I plan on continuing with Python as it is freshest in my memory and is versatile and readable.

Beyond investigating data myself, I think a basic understanding of programming language and methods is necessary to fully use and apply analysis and resources other people create. Knowing where the data comes from, how it has been manipulated, and what the results are actually showing is key in preventing bias and using analysis is a positive way for the people it represents. I am interested in continuing to learn about data ethics, public health collection methods and standards of practice, and data communication in the future of my Health Informatics and Analytics Masters.

**REFERENCES**

Bureau, U. S. C. (2023). *Explore census data.* https://data.census.gov/table/ACSST1Y2022.S0103?q=demographics%2Bby%2Bstate

Centers for Disease Control and Prevention. (2023). *CDC WONDER*. https://wonder.cdc.gov/controller/datarequest/D158

Centers for Disease Control and Prevention. (2022, November 21). *Interim guidance for influenza outbreak management in long-term care and post-acute care facilities*. Centers for Disease Control and Prevention. https://www.cdc.gov/flu/professionals/infectioncontrol/ltc-facility-guidance.htm

Drummond, H. (2023, September 6). *Staffing for survival: The critical importance of hospital preparedness during the 2023 flu season*. LinkedIn. https://www.linkedin.com/pulse/staffing-survival-critical-importance-hospital-during-hank-drummond/

OSHA. (2023). *Seasonal flu - employer guidance reducing healthcare workers’ exposures to seasonal flu virus.* Occupational Safety and Health Administration. https://www.osha.gov/seasonal-flu/healthcare-employers#