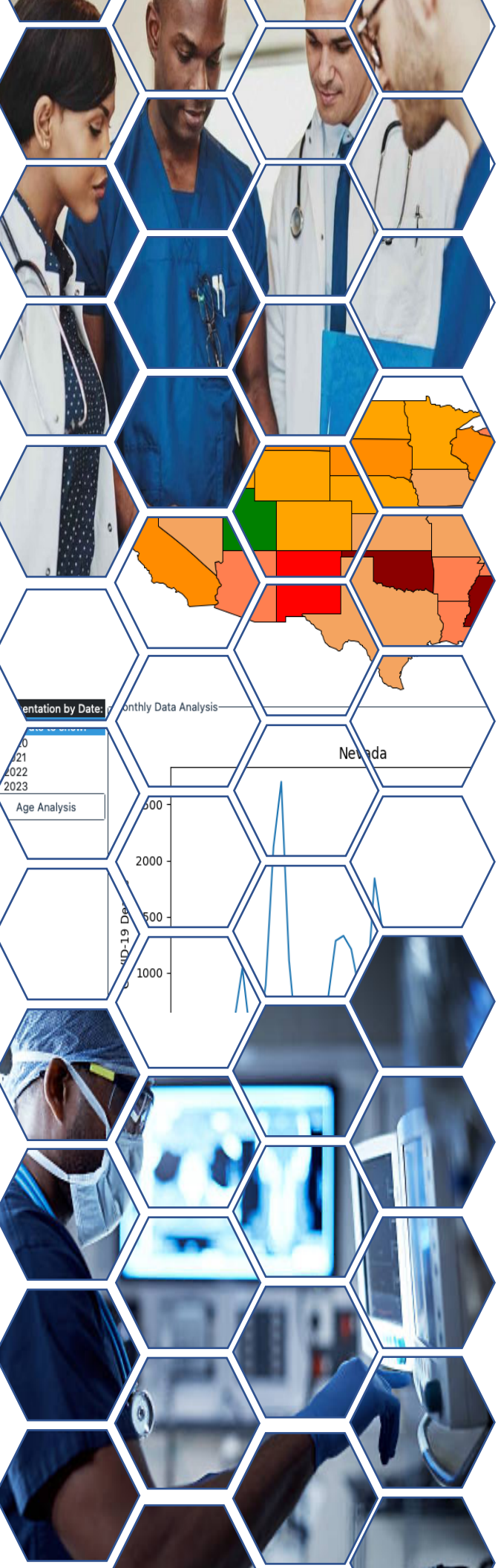


HealthMap

Data Analysis and Visualization Project

Ashkan Nikfarjam
Quynh Thach

Table of Contents

Background

Respiratory illnesses are a major concern for public health in the United States, due to their highly contagious nature. This has been especially highlighted in recent years with the rise of the new COVID-19 virus that caused a worldwide pandemic and cost many lives in the U.S. Due to varying levels of immune system functionality in different age groups, some populations may face higher mortality rates due to respiratory illnesses, or the rise of disease in a certain demographic may be earlier each year than in others. The ability to monitor diseases and track trends over time can clarify the preparedness of different states and enable timely public health interventions to be implemented.

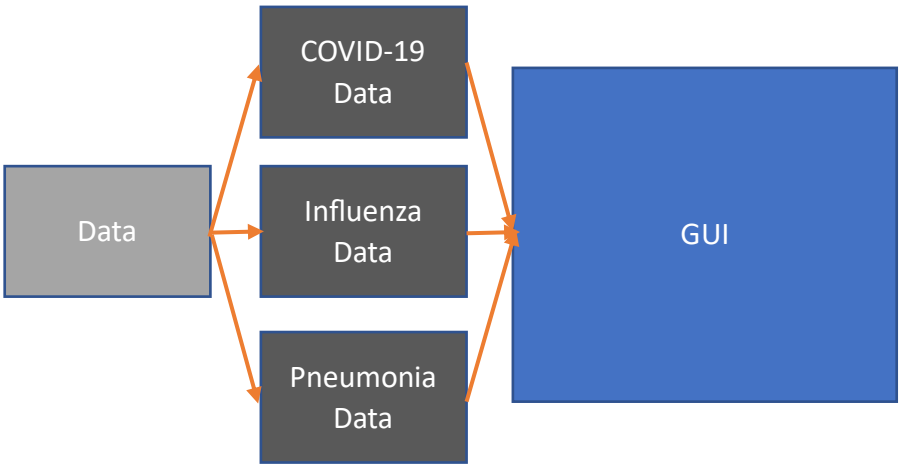
Problem

The US Department of Health and Human Services releases a dataset that gets updated biweekly (DATASET SOURCE). In this dataset, they record the number of deaths per week for three common types of respiratory diseases: COVID-19, influenza, and pneumonia, as well as how deaths from these respiratory diseases compare to total deaths in each period. The number of deaths in each state is further categorized by age into four groups: 0-17 years old, 18-64 years old, 65 and older, and all ages (for cases where age has not been disclosed.)

The goal of this project is thus to build a data analysis and visualization tool ("HealthMap") using Python to be able to visualize patterns in this dataset. The first dimension to analyze is geographic trends, in the form of a color-coded map of the U.S. We also want to view data over time, by being able to select different years to view data for or view monthly charts. Finally, we would like to see breakdowns by age to see how different age groups are affected.

Overall Organization

There are two major phases of the project. The first is data preprocessing, where the raw dataset is cleaned up to only contain the information we need and is split by disease for consumption by the graphical user interface (GUI). The GUI is built using TKinter and applies additional transformations on each of the disease-specific dataframes for each of the charts displayed. This process is illustrated in the figure below in Figure 1. Overall, the project employs a procedural programming style to perform both the data preprocessing as well as the GUI rendering.



Design of Program

Data Preprocessing

Data was preprocessed using the pandas library to read the raw data CSV files. For each disease, only information about the week that the data is for, state ("jurisdiction"), number of deaths for the disease, and age group were kept.

GUI

The user interface was built using TKinter, specifically the ttk and ttk-booootstrap packages due to their greater selection of widgets and styling options. Some charts and plots were made via matplotlib and seaborn. There were several components in the GUI, such as a map for geographic data visualization, line graph showing monthly deaths, and heatmap of deaths in each age-group by state, alongside data rendered in a tabular format. The layout of the interface is shown below. Each of the sections shown was created as a Frame element as a widget placeholder.

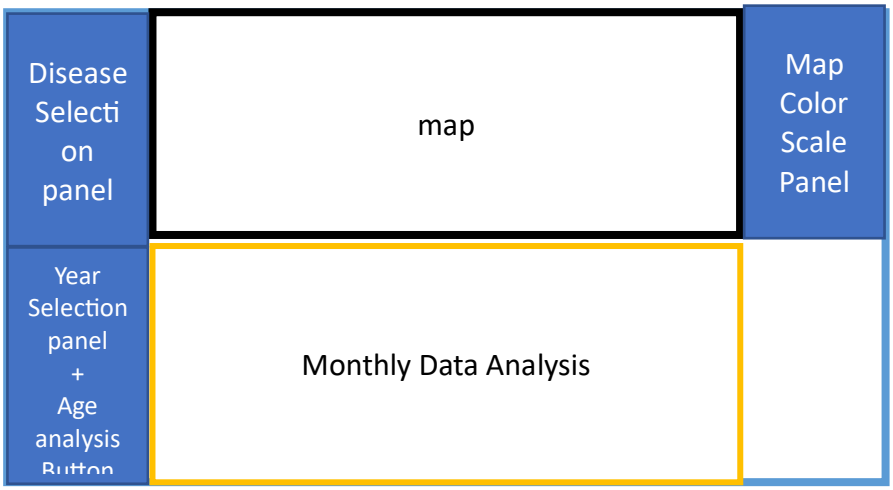


Figure 2

To create the map showing the geographic distribution of deaths, the TKinter Canvas element was used. We found a public GeoJSON file containing the coordinates and outlines for each U.S. state and used that to render each state as a Polygon element on the canvas. A dictionary was kept that tracked which polygon corresponded to each state. For simplicity of rendering, Alaska and Hawaii were excluded. When an option is selected to view data for a different disease or year, additional filtering is performed on the dataframe. The thresholds for each color in the color scale are calculated based on the maximum number of deaths per capita and stored as a calculated color dictionary. We can then use the state polygon dictionary and the color dictionary to efficiently re-color the map. The legend for the colors is displayed on the right side of the map based on the color dictionary.

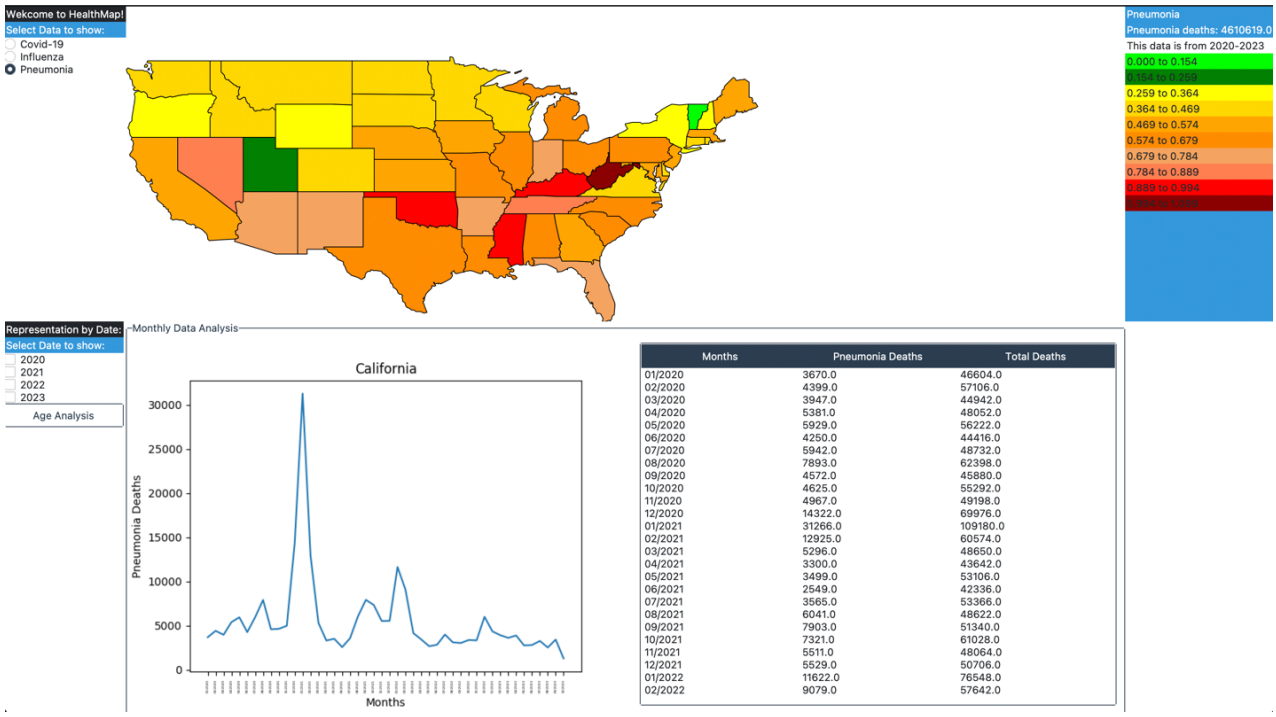
To show the time dimension of deaths, data from different years can be included using checkboxes on the side, or a monthly view for a state can be opened by clicking the state. Each of these options performs additional filtering in the GUI. To render the monthly view on a state, click, we use the TKinter click event to find the closest polygon to the click, and then use our state polygon dictionary to check which state the polygon belongs to. This allows us to render a line plot tracking the monthly deaths in the state for the disease. The filtered dataframe showing the month and deaths for the state is also presented as a table next to the plot.

Finally, the age analysis can be opened in a separate window by clicking on a button in the sidebar. The age analysis panel renders a heatmap of deaths categorized by age group and state. The table showing deaths by age group in each state is also shown below it.

Results

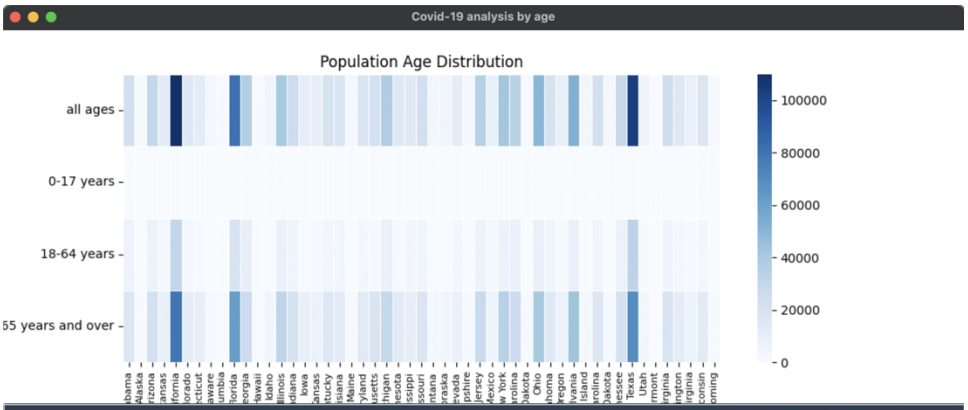
The following image in Figure 3 shows the main interface window for the HealthMap application. On the left side, there are options to select which disease to view data for, as well as for which years. The top half of the window shows the map of the U.S., color-coded by number of deaths per state according to the legend in the bar on the right side. For this example, we have clicked on California, which opens the bottom half of the panel with a detailed line plot of pneumonia deaths in California over time.

By clicking the “Age Analysis” button on the left side, we can open the age analysis window, as shown in Figure 4. Here we can see for COVID-19 that the 65+ year old population experienced more deaths in many states.



Months	Pneumonia Deaths	Total Deaths
01/2020	3670.0	46604.0
02/2020	4399.0	57106.0
03/2020	3947.0	44942.0
04/2020	5381.0	48052.0
05/2020	5929.0	56222.0
06/2020	4250.0	44416.0
07/2020	5942.0	48732.0
08/2020	7893.0	62398.0
09/2020	4572.0	45880.0
10/2020	4625.0	55292.0
11/2020	4967.0	49198.0
12/2020	14322.0	69976.0
01/2021	31266.0	109180.0
02/2021	12925.0	60574.0
03/2021	5296.0	48650.0
04/2021	3300.0	43642.0
05/2021	3499.0	53106.0
06/2021	2549.0	42336.0
07/2021	3565.0	53366.0
08/2021	6041.0	48622.0
09/2021	7903.0	51340.0
10/2021	7321.0	61028.0
11/2021	5511.0	48064.0
12/2021	5529.0	50706.0
01/2022	11622.0	76548.0
02/2022	9079.0	57642.0

Figure 3



POPESTIMATE2022	Jurisdiction	all ages	0-17 years	18-64 years	65 years and over
333287557	United States	1150115.0	1710.0	275983.0	872422.0
5074296	Alabama	22462.0	0.0	6148.0	15957.0
733583	Alaska	1079.0	0.0	251.0	632.0
7359197	Arizona	29839.0	0.0	7956.0	21484.0
3045637	Arkansas	12523.0	0.0	2787.0	9226.0
39029342	California	109736.0	0.0	30547.0	78952.0
5839926	Colorado	15015.0	0.0	3084.0	11531.0
3626205	Connecticut	12193.0	0.0	1614.0	10153.0
1018396	Delaware	3036.0	0.0	367.0	2221.0
671803	District of Columbia	1396.0	0.0	217.0	936.0

Figure 4

Instructions

- After downloading the zip file contain the codes and unzip in your desired directory.
- Open terminal and change to the directory that contains the files.
    - $ cd ~/path to the file/
- 2 ways to run this program:
    - Execute the run script
      $sh run.sh (exclude $)
      If this doesn't work first do this and try again:
      $ chmod u+x run.sh
    - Run DataFrames.py first.
      $python3 DataFrames.py
    - To run GUI: $ python3 GUI.py
- Instruction to how to navigate the GUI is included in readme.md as well as screenshot of some dataframes

Files Instructions:

- DataFrames.py is the script that prep that create data frames for each disease.
- analysisFunctions.py is the file containing functions that prep the data for visualization.
- Banner.py contains the function that create the monthly analysis frame.
- AgeBanner.py function creates the widgets for age analysis window.
- Table.py contains frame in places that is needed.
- Data.csv contains the data
    - Warning: please do not modify this file

- State-geojason.json contains the geodata dictionary that we used to create the map
- Run.sh is a bash script that run and execute HealthMap without use need to anything else.

Works Cited:

- "NST-EST2022." *Index of /Programs-Surveys/Popest/Datasets*, United States Census Bureau, 22 Dec. 2022, www2.census.gov/programs-surveys/popest/datasets/.

- Publisher Centers for Disease Control and Prevention. (2023, November 10). *Provisional Death Counts for Influenza, Pneumonia, and COVID-19*. Catalog. https://catalog.data.gov/dataset/provisional-death-counts-for-influenza-pneumonia-and-covid-19

References to python Libraries used:

Tkinter ttk: https://docs.python.org/3/library/tkinter.ttk.html

TtkBootstrap: https://ttkbootstrap.readthedocs.io/en/version-0.5/handbook.html

matpotlib.pyplot: https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html