

Midterm Project

The program uses two packages: requests and matplotlib.

```
#This program looks at a COVID tracker API and visually presents the data using matplotlib
import requests
import matplotlib.pyplot as plt
```

The main focus for this project was matplotlib. Matplotlib is a data visualization and graphical plotting library in python. This package was used to create visualization of the data in the form of bar graphs, scatter plots, and pie charts. Throughout the program, matplotlib has been referred to using the alias *plt*.

Because I wanted to examine real-world COVID 19 data, I used the requests package to make HTTP requests to the COVID tracking API to retrieve COVID-19 data for each state. I used requests to get a JSON file containing the current COVID-19 data in the United States.

```
#Retrives data from a COVID tracker API
r = requests.get("https://api.covidtracking.com/v1/states/current.json")
COVID_data = r.json()
```

The requests package was chosen because it enables the program to send HTTP requests to the COVID tracking API in order to retrieve the data required to create the visualizations. The matplotlib package was chosen due to its sophisticated options for visualizing the data I acquired from the API.

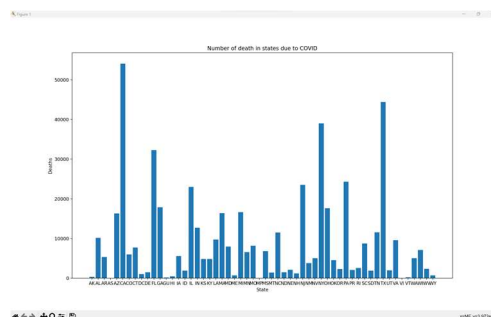
The program utilizes three matplotlib methods: `bar(x, height)`, `scatter(x, y)`, and `pie(x)`.

Method 1:

```
#Method 1: Bar graph showing number of death in states within the United States
xAxis = []
yAxis = []
for data in COVID_data:
    xAxis.append(data["state"])
    yAxis.append(data["death"])

plt.bar(xAxis, yAxis)
plt.xlabel("State")
plt.ylabel("Deaths")
plt.title("Number of death in states due to COVID")
plt.show()
```

The first method creates a bar graph showing the number of deaths in each state in the United States. I created two lists (`xAxis` and `yAxis`) to iterate through the data provided by the API to retrieve the state names and the number of deaths in the respective state. Then, using matplotlib's "bar" function, I created a graph with the names of the States on the x-axis and the number of deaths on the y-axis.

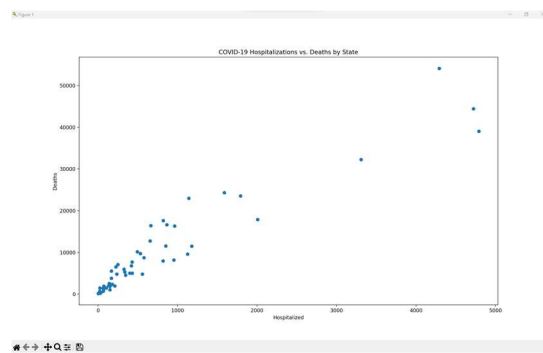


Method 2:

```
#Method 2: Scatter plot showing the correlation between people in each state being hospitalized compared to the number of death in the state
xAxis = []
for data in COVID_data:
    xAxis.append(data["hospitalizedCurrently"])

plt.scatter(xAxis, yAxis)
plt.xlabel("Hospitalized")
plt.ylabel("Deaths")
plt.title("COVID-19 Hospitalizations vs. Deaths by State")
plt.show()
```

The second method creates a scatter plot showing the correlation between the number of people hospitalized in each state and the number of deaths due to COVID-19. Similar to the first method, I created a list (xAxis) to iterate through the data in the JSON file and retrieve all the number of people hospitalized. I used the already populated list containing the number of deaths from the previous method. Combining the two lists, I used matplotlib's "scatter" function to create a scatter plot with the number of hospitalization on the x-axis and the number of death on the y-axis. This showed a positive correlation between the data fields.



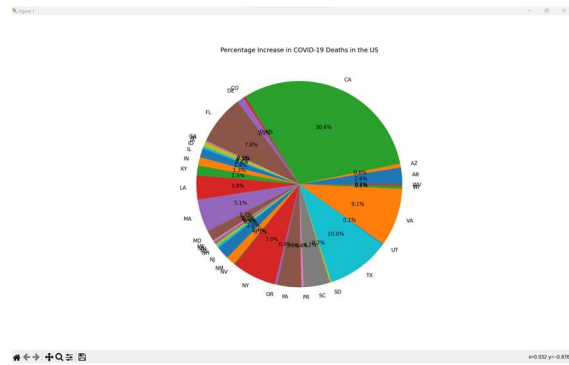
Method 3:

```
#Method 3: Pie chart showing the increase in death faced by States
state = []
size = []

for data in COVID_data:
    if data["deathIncrease"] > 0:
        state.append(data["state"])
        size.append(data["deathIncrease"])

fig1, ax1 = plt.subplots()
ax1.pie(size, labels=state, autopct="%1.1f%%")
plt.title("Percentage Increase in COVID-19 Deaths in the US")
plt.show()
```

For this method, I wanted to try a slightly harder function provided by matplotlib. I decided to create a pie chart showing the percentage increase in COVID-19 deaths in each state. Here I had two lists (state and size), although I set the condition for populating the list only if the increase in death is greater than 0. Lastly, I used matplotlib's "pie" function to create a pie chart with the state names as the labels and the percentage increase in deaths as the size of each slice.



In real-world settings, the requests package could be used to retrieve data from various APIs, such as financial market data or social media data. The matplotlib package could be used to create visualizations of various types of data, such as financial data or customer behavior data. These visualizations can help businesses and organizations to gain insights and make informed decisions. In retrospect, my code also simulates a real-world scenario and provides users with information about current COVID-19 data.