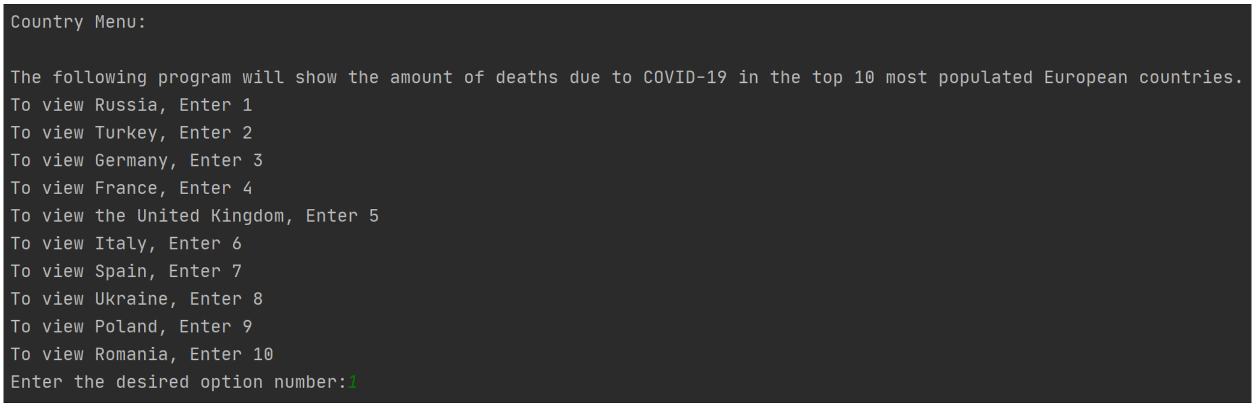
Stacy Shang
Professor Ryan
CS 110
13 December 2020

CS 110 Individual Project: COVID-19 Cases in Europe

Section 1:

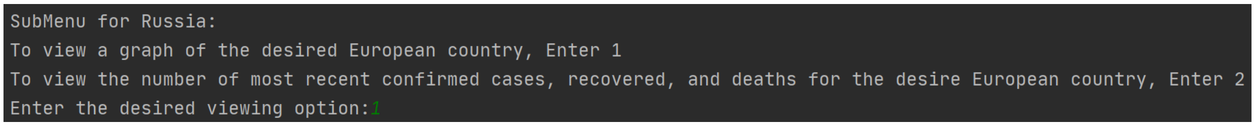
For this program, I decided to write a program that would show a graph that tracks the deaths due to COVID-19 within one of the top ten most populated European countries. These countries, from most populated to least populated, were Russia, Turkey, Germany, France, the United Kingdom, Italy, Spain, Ukraine, Poland, and Romania. The question that I wanted to answer for this project was: How do the most populated European nations compare in terms of COVD-19 deaths over the past year? The user would be provided a menu-based interface of options, with the main menu asking the user to enter a number from 1 to 10 corresponding with the respective countries. Then, they would be prompted to choose either "1" to view a graph of the COVID-19 related deaths over the past year for the country of their choosing, or "2" to view the most recent statistics relating to COVID-19 for that country: the number of confirmed cases, the number of people recovered, and the number of deaths.

An example of input and output of the program would be as follows: The user runs the program and is greeted with an explanation of what the program does and the menu of European nations. They then choose Russia, so they enter "1".



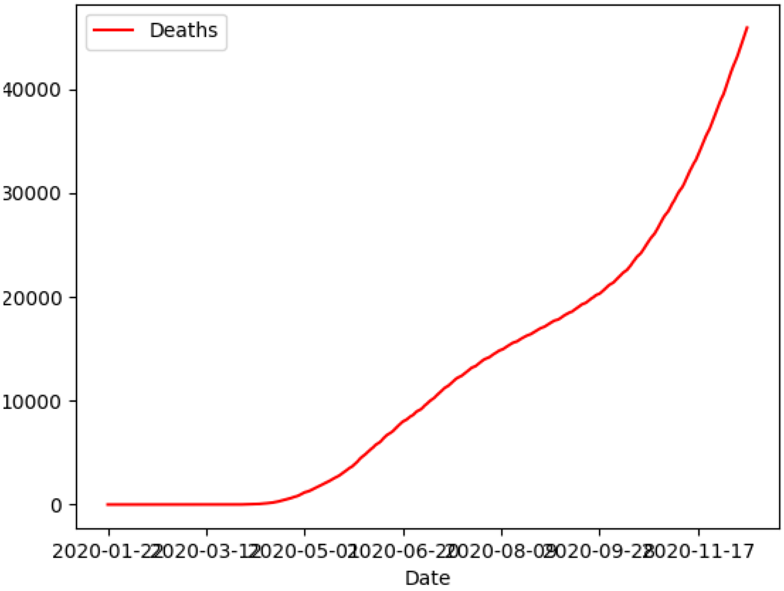
```
Country Menu:

The following program will show the amount of deaths due to COVID-19 in the top 10 most populated European countries.
To view Russia, Enter 1
To view Turkey, Enter 2
To view Germany, Enter 3
To view France, Enter 4
To view the United Kingdom, Enter 5
To view Italy, Enter 6
To view Spain, Enter 7
To view Ukraine, Enter 8
To view Poland, Enter 9
To view Romania, Enter 10
Enter the desired option number:1
```

The user wants to see the graph of the COVID-19 deaths over time, so they enter "1".

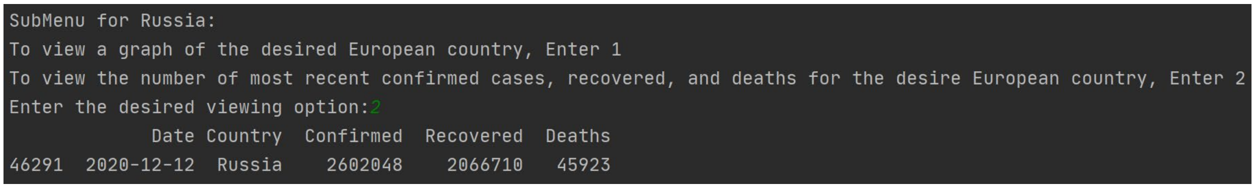


```
SubMenu for Russia:
To view a graph of the desired European country, Enter 1
To view the number of most recent confirmed cases, recovered, and deaths for the desire European country, Enter 2
Enter the desired viewing option:1
```

They then see a line graph of the cases over time.

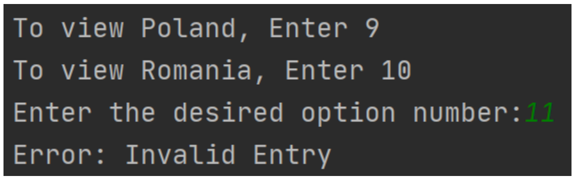


If the user wanted to view the most recent COVID-19 statistics for Russia, they would repeat the first step of choosing "1" for the country, then they would enter "2" and see the output.

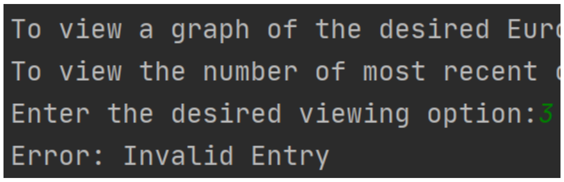


```
SubMenu for Russia:
To view a graph of the desired European country, Enter 1
To view the number of most recent confirmed cases, recovered, and deaths for the desire European country, Enter 2
Enter the desired viewing option:2
             Date Country  Confirmed  Recovered  Deaths
46291  2020-12-12  Russia    2602048    2066710   45923
```

If the user entered a different number for either of the menus, they would see an error message.



```
To view Poland, Enter 9
To view Romania, Enter 10
Enter the desired option number:11
Error: Invalid Entry
```



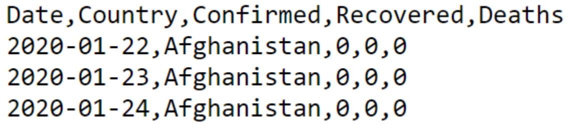
```
To view a graph of the desired Euro
To view the number of most recent c
Enter the desired viewing option:3
Error: Invalid Entry
```

Section 2:

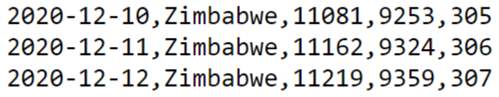
The targeted audience of my program would be individuals who wanted to see the trajectory of COVID-19 related deaths within the most populated European nations. This could be individuals who are from those European nations and wish to compare their nation's death count to their neighboring countries, or individuals who were simply interested in how European nations compared to one another. I became interested in this topic because of the closeness in proximity most European nations have to one another, as Spain, Italy, France, Germany, Poland, Ukraine, Romania, and Russia, eight out of the ten most populated European nations, are all connected to one another, potentially leading to similar COVID-19 death counts. In addition, I was interested in the relationship between the population of people within those countries and the amount of deaths they had from COVID-19.

Section 3:

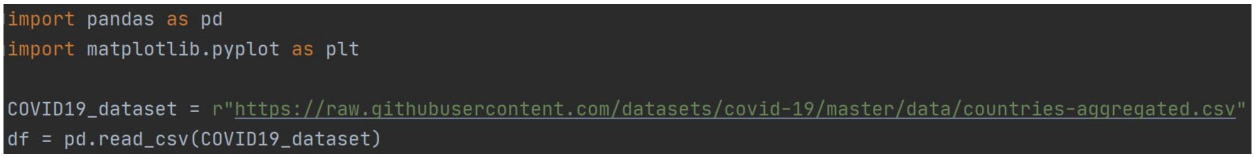
I collected my data by downloading a CSV dataset file that contained the day-by-day tracking of COVID-19 confirmed cases, persons recovered, and deaths for every country in the world from January 22, 2020 to present day (December 12, 2020). Pictured below is the first three entries of this dataset, the last three entries of this dataset, and the code I wrote in the program that imports the necessary libraries and csv file.



```
Date,Country,Confirmed,Recovered,Deaths
2020-01-22,Afghanistan,0,0,0
2020-01-23,Afghanistan,0,0,0
2020-01-24,Afghanistan,0,0,0
```

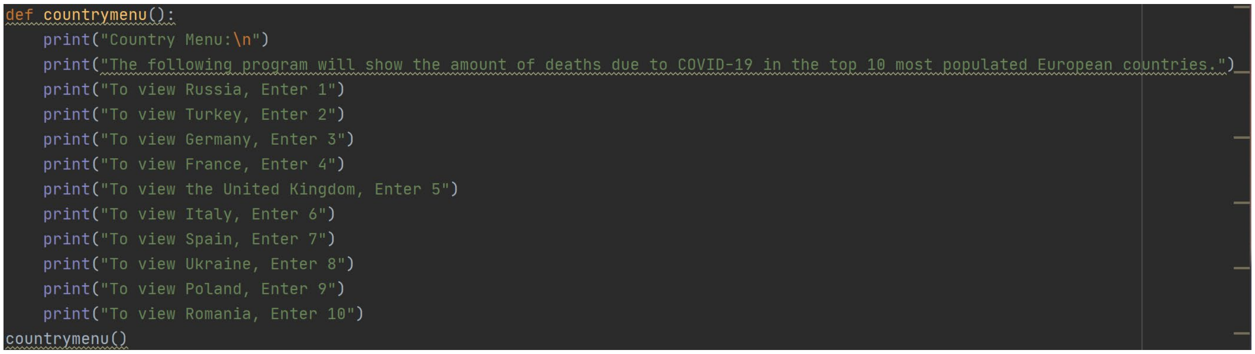


```
2020-12-10,Zimbabwe,11081,9253,305
2020-12-11,Zimbabwe,11162,9324,306
2020-12-12,Zimbabwe,11219,9359,307
```



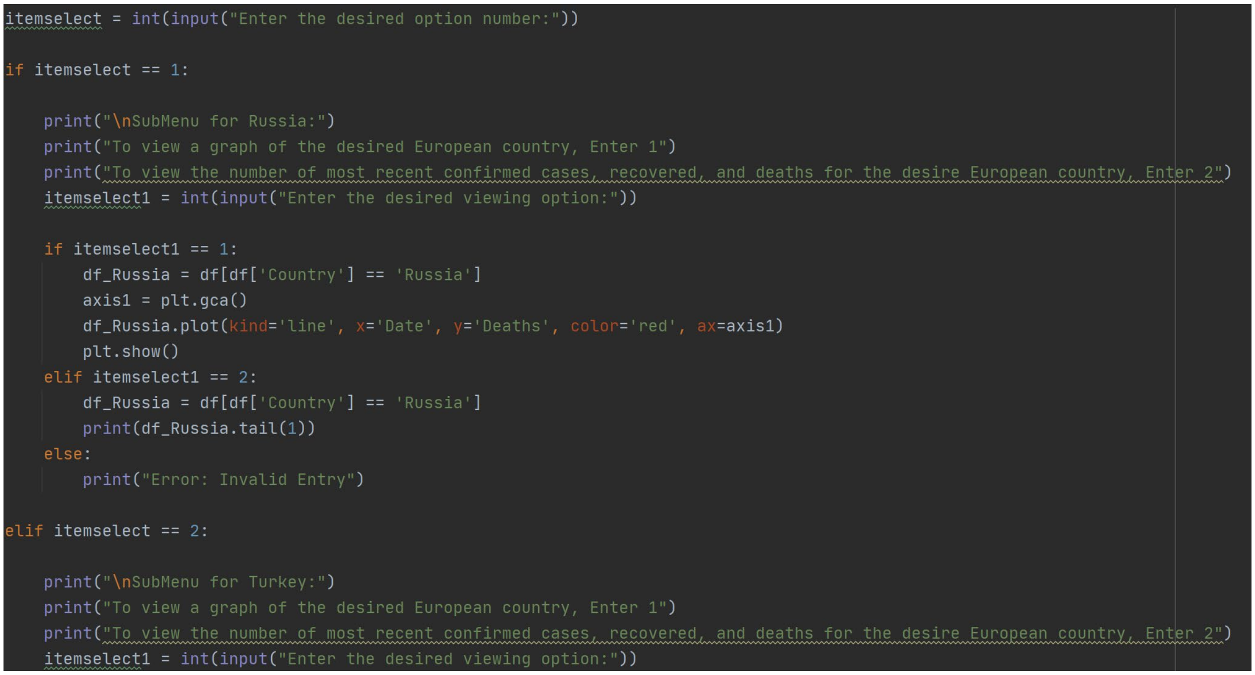
```
import pandas as pd
import matplotlib.pyplot as plt

COVID19_dataset = r"https://raw.githubusercontent.com/datasets/covid-19/master/data/countries-aggregated.csv"
df = pd.read_csv(COVID19_dataset)
```

I then created a menu that specified each of the 10 countries and allowed the user to enter a number that would correspond the desired country. The numbers were assigned from 1 to 10 from most populous to least populous out of the 10 most populated European nations.



```
def countrymenu():
    print("Country Menu:\n")
    print("The following program will show the amount of deaths due to COVID-19 in the top 10 most populated European countries.")
    print("To view Russia, Enter 1")
    print("To view Turkey, Enter 2")
    print("To view Germany, Enter 3")
    print("To view France, Enter 4")
    print("To view the United Kingdom, Enter 5")
    print("To view Italy, Enter 6")
    print("To view Spain, Enter 7")
    print("To view Ukraine, Enter 8")
    print("To view Poland, Enter 9")
    print("To view Romania, Enter 10")
countrymenu()
```

After the user entered in the country number for the Country Menu, an if-statement would appear corresponding to the country chosen. Within that if statement contains a Submenu corresponding to the country, asking the user to choose whether they wanted to see a graph of the COVID-19 deaths for the country, or to view the most recent statistics about that country (the confirmed cases, recovered numbers, and deaths). The user would enter a "1" or a "2" for the respective options, or be greeted with an error statement if they entered a number besides one of those.



```
itemselect = int(input("Enter the desired option number:"))

if itemselect == 1:

    print("\nSubMenu for Russia:")
    print("To view a graph of the desired European country, Enter 1")
    print("To view the number of most recent confirmed cases, recovered, and deaths for the desire European country, Enter 2")
    itemselect1 = int(input("Enter the desired viewing option:"))

    if itemselect1 == 1:
        df_Russia = df[df['Country'] == 'Russia']
        axis1 = plt.gca()
        df_Russia.plot(kind='line', x='Date', y='Deaths', color='red', ax=axis1)
        plt.show()
    elif itemselect1 == 2:
        df_Russia = df[df['Country'] == 'Russia']
        print(df_Russia.tail(1))
    else:
        print("Error: Invalid Entry")

elif itemselect == 2:

    print("\nSubMenu for Turkey:")
    print("To view a graph of the desired European country, Enter 1")
    print("To view the number of most recent confirmed cases, recovered, and deaths for the desire European country, Enter 2")
    itemselect1 = int(input("Enter the desired viewing option:"))
```

This internal if-statement was then repeated for each of the 10 countries, with the larger if-statement directing the user to the desired European nation they wished to view.

The programming technique and structure that I used to write this program was a menu-based structure that had a main menu that allowed the user to specify which European country they wanted to view, then corresponding submenus that would allow the user to choose to view a graph of COVID-19 deaths in that country or recent statistics about COVID-19 in that country.

Section 4:

Some difficulties I faced while programming this were finding the dataset that I wanted to use for the project and importing that dataset into the program. Other datasets I came across had the number of cases go from most recent to least recent, which would reverse the graph, and I was not sure how to reverse the dataset within the program. I found it was easiest to assign the numbers to the country within the menu to make it easier for the user, as I initially had the user type in the country name. However, with the previous method, I found that if there were spelling mistakes, spacing differences, or captilization errors (without writing a lowercase function into the code), that it was easier to have errors occur.

Section 5:

Potential extensions or improvements I could make to this code would to have it display the confirmed cases and recovered cases for the desired country on the graph, in addition to the tracking of the deaths. Other improvements I could make would be to add more countries to the list or to allow the user to open multiple countries' graphs or statistics at the same time so the user could compare them side-by-side.