ANALYZE TRENDS IN INCIDENCE AND MORTALITY OF COVID-19 USING PYTHON

Author - Kimaya Havle

Introduction

COVID-19 has put the world on edge. Its high infectious rate has put patients at risk of critical condition and death. There has been an unprecedented influx of patients in hospitals, especially ICUs. This multitude of admissions exceeded the ICU bed capacities. Shortage of oxygen masks, Personal Protective Equipment (PPEs), hospital staff, and other hospital supplies led to numerous deaths. The rapid surge has led to a potential compromise and inconsistency in patient care quality.

GOAL

The goal is use python to obtain trends in incidence and deaths due to COVID-19 which hospitals can use to be well-equipped and better prepare for treating patients and deliver quality care to every patient.

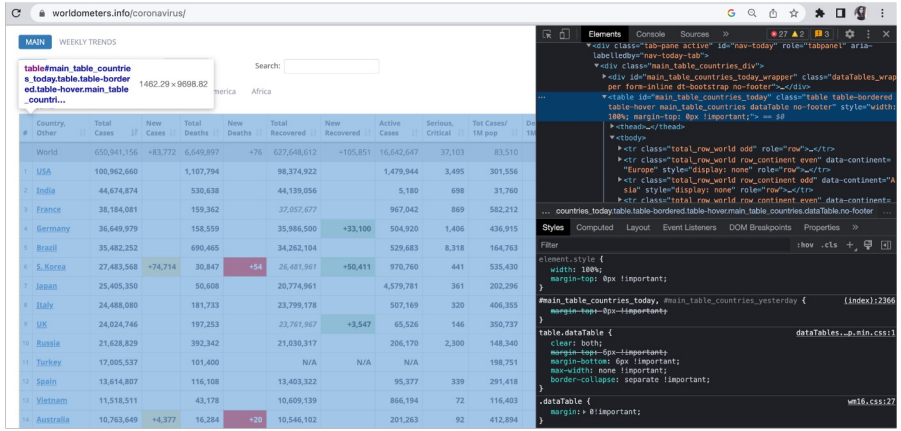
STRETCH GOAL

Perform predictions based on the trends
using the “scikit learn” module

Overview of the project

- Finding websites that provide COVID-19 data
- Understanding the web page layout and finding the relevant table
- Using BeautifulSoup module to scrape the data
- Running the scraper every day for 7 days to build dataset
- Storing that in CSV file
- Reading the CSV data , cleaning and filtering it and creating a Pandas Dataframe
- Re-modeling the Pandas Dataframe for visualization
- Performing visualization using Seaborn module

Inspecting world-o-meter HTML table



DataFrame Built

	Country,Other	TotalCases	NewCases	TotalDeaths	NewDeaths	TotalRecovered	ActiveCases	Population	Continent
China	China	297516	4010.0	5232.0	1.0	265471.0	26813.0	1.448471e+09	Asia
USA	USA	100425201	13467.0	1104602.0	88.0	97929295.0	1391304.0	3.348053e+08	North America
India	India	44671806	268.0	530601.0	0.0	44134001.0	7204.0	1.406632e+09	Asia
France	France	37492134	0.0	158511.0	0.0	36732115.0	601508.0	6.558452e+07	Europe
Germany	Germany	36348477	30016.0	157388.0	274.0	35600900.0	590189.0	8.388360e+07	Europe
...	...	...	...	...	...	...	...	...	...
Total:	Total:	64912852	34821.0	1335138.0	142.0	63095452.0	482262.0	0.000000e+00	South America
Total:	Total:	12886154	3908.0	22083.0	12.0	12555016.0	309055.0	0.000000e+00	Australia/Oceania
Total:	Total:	12702860	1766.0	258039.0	6.0	12014471.0	430350.0	0.000000e+00	Africa
Total:	Total:	721	0.0	15.0	0.0	706.0	0.0	0.000000e+00	0
Total:	Total:	644755103	263851.0	6632574.0	949.0	623519963.0	14602566.0	0.000000e+00	All

238 rows × 9 columns

Challenging Part of the Project

```
url = "https://www.worldometers.info/coronavirus/"
data = requests.get(url)
```

```
parsed = BeautifulSoup(data.content, "html.parser")
raw_table = parsed.find_all('table', id = "main_table_countries_yesterday")[0]
table_df = pd.read_html(str(raw_table), displayed_only = False)[0]
table_df = table_df[8:]
```

```
from datetime import datetime
today = datetime.today().strftime('%Y-%m-%d')
table_df.to_csv('worldometer-'+ today +'.csv')
```

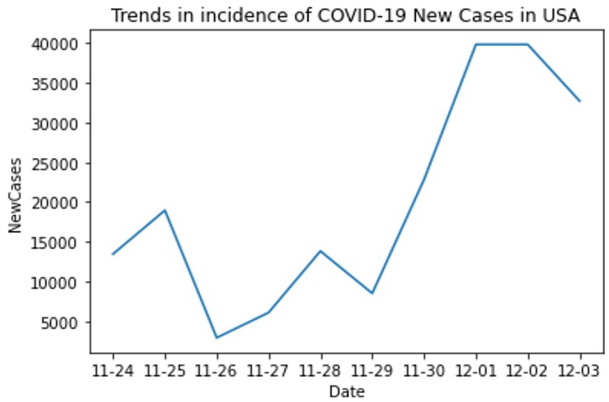
```
def obtain_data(file_name):
    # the csv file was converted in panda dataframe
    df_table = pd.read_csv(file_name)
    # only relevant data from the csv was obtained to filter the noise
    relevant_data = df_table[['Country,Other', 'TotalCases', 'NewCases', 'TotalDeaths', 'NewDeaths', 'TotalRecovered
    # data not available was replaced by 0
    relevant_data = relevant_data.fillna(0)
    # Inorder to obtain data based on the country, the data was remodelled to use country as the row index
    country_names = relevant_data['Country,Other']
    index_country_mapping = {}
    for i in range(len(country_names)):
        index_country_mapping[i] = country_names[i]
    relevant_data = relevant_data.rename(index = index_country_mapping)
    return relevant_data
```

Modules Used

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

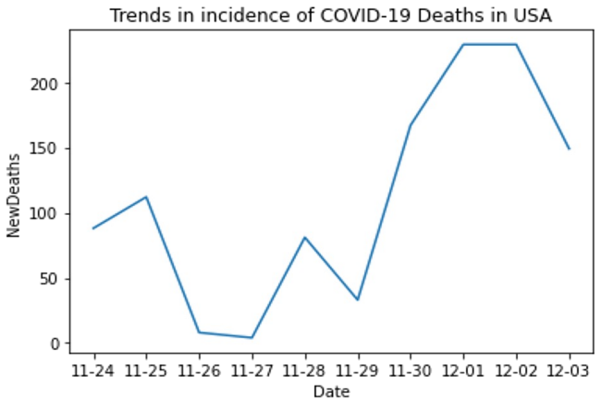
Trends in Incidence

```
usa_data_by_date = []
for filename in eleven_days_data_files:
    date = filename[12:22]
    data = obtain_data('COVID-19_Project/'+filename)
    five_countries_data = get_five_countries_data(data)
    usa_data = five_countries_data.loc[['USA']]
    dict_data = [date[5:], usa_data['NewCases'][0]]
    usa_data_by_date.append(dict_data)

usa_data_for_lineplot = pd.DataFrame(usa_data_by_date, columns=["Date", "NewCases"])
usa_data_for_lineplot
sns.lineplot(data=usa_data_for_lineplot, x="Date", y="NewCases").set(title="Trends in incidence of \
COVID-19 New Cases in USA")
```



Trends in Deaths

```
usa_data_by_date = []
for filename in eleven_days_data_files:
    date = filename[12:22]
    data = obtain_data('COVID-19_Project/'+filename)
    five_countries_data = get_five_countries_data(data)
    usa_data = five_countries_data.loc[['USA']]
    dict_data = [date[5:], usa_data['NewDeaths'][0]]
    usa_data_by_date.append(dict_data)

usa_data_for_lineplot = pd.DataFrame(usa_data_by_date, columns=["Date", "NewDeaths"])
usa_data_for_lineplot
sns.lineplot(data=usa_data_for_lineplot, x="Date", y="NewDeaths").set(title="Trends in incidence of COVID-19\
Deaths in USA")
```



Trends in Incidence and Deaths for Five Countries

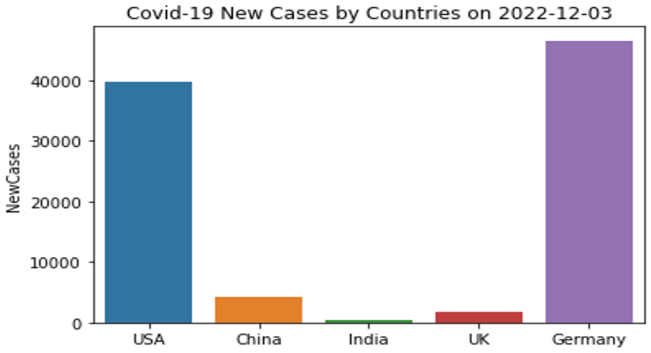
```
def get_five_countries_data(data):
    # For this project, five countries are chosen
    five_countries_data = data.loc[['USA','China','India','UK','Germany']]
    return five_countries_data
```

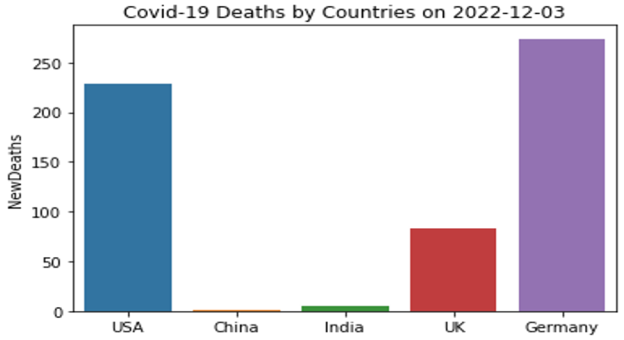
†

```
for filename in eleven_days_data_files:
    date = filename[12:22]
    data = obtain_data('COVID-19_Project/'+filename)
    five_countries_data = get_five_countries_data(data)
    new_cases_plot = barplot_new_cases(five_countries_data, date)
    barplot_new_cases_fig = new_cases_plot.get_figure()
    barplot_new_cases_fig.savefig('COVID-19_Project/Images/new_cases_plot_' + date + '.jpg')
```

```
def barplot_new_cases(data, date):
    new_cases_five_countries = sns.barplot(x=data.index, y=data['NewCases']).set_title('Covid-19 New Cases by Countr
    return new_cases_five_countries
```

```
def barplot_new_deaths(data, date):
    new_deaths_five_countries = sns.barplot(x=five_countries_data.index, y=five_countries_data['NewDeaths']).set_tit
    return new_deaths_five_countries
```





References:

- https://link.springer.com/article/10.1186/s13613-020-00702-7
- https://link.springer.com/article/10.1186/s13613-020-00702-7

THANK YOU