**DAC\_PHASE – 4-PROJECT SUBMISSION ON COVID -19 CASE ANALYSIS – DEVELOPMENT PART-02**

**TEAM MEMBER – JEFFRIN NELSON J ( 411521104047)**

****

**INTRODUCTION :**

COVID-19, also known as the coronavirus, is a highly contagious respiratory illness caused by the SARS-CoV-2 virus. It first emerged in late 2019 and has since spread globally, leading to a pandemic. It’s important to stay informed about preventive measures and follow guidelines from health authorities to stay safe. If you have any specific questions about COVID-19.When it comes to analyzing COVID-19 cases, experts look at various factors like the number of confirmed cases, testing rates, hospitalizations, and mortality rates. They analyze trends and patterns to understand the spread of the virus and make informed decisions to control it. It’s important to rely on credible sources like health departments and organizations for accurate and up-to-date information. If you have any specific questions about COVID-19 case analysis.

**ANALYSIS OBJECTIVE:**

The pandemic of Coronavirus Disease 2019 (COVID-19) is a timely reminder of the nature and impact andof Public Health Emergencies of International Concern. As of 12 January 2022, there were over 314 million cases and over 5.5 million deaths notifed since the start of the pandemic. The COVID-19 pandemic takes variable shapes and forms, in terms of cases and deaths, in diferent regions and countries of the world. The objective of this study is to analyse the variable expression of COVID-19 pandemic so that lessons can be learned towards an efective public health emergency response.

**METHODS**:

We conducted a mixed-methods study to understand the heterogeneity of cases and deaths due to the COVID-19 pandemic. Correlation analysis and scatter plot were employed for the quantitative data. We used Spear-Man’s correlation analysis to determine relationship strength between cases and deaths and koonsocio-economic and health systems. We organized qualitative information from the literature and conducted a thematic analysis to recog-nize patterns of cases and deaths and explain the fndings from the quantitative data.

**RESULTS**:

We have found that regions and countries with high human development index have higher cases and Deaths per million population due to COVID-19. This is due to International connectedness and mobility of their Population related to trade and tourism, and their vulnerability related to older populations and higher rates of non-communicable diseases. We have also identifed that the burden of the pandemic is also variable among high- and middle-income countries due to differences in the governance of the pandemic, fragmentation of health systems, and socio-economic inequities.

**KEYWORDS**:

Pandemics, Epidemics, COVID-19, Heterogeneity, Governance, Equity.

**DATA COLLECTION :**

The file contains information on newly reported COVID-19 cases and deaths in EU/EEA countries. Each row contains the corresponding data for a certain day and per country. The File is updated daily. You may use the data in line with ECDC’s copyright policy.Source ECDC uses multiple information sources per country. The information sources are Ministries of Health or National Public Health Institutes (websites, twitter official accounts or Facebook official Accounts). More information is available at <https://www.ecdc.europa.eu/en/covid-19/data-Collection>.

Interpretation of COVID-19 data.The data included in this file is collected by the ECDC Epidemic Intelligence from various sources and is affected by the local testing strategy, laboratory capacity and the effectiveness of surveillance systems. Comparing the epidemiological situation regarding COVID-19 between countries should therefore not be based on these rates alone. However, at the individual country level, this indicator may be useful for monitoring the national situation over time.Testing policies and the number of tests performed per 100 000 persons, vary markedly across the EU/EEA. More extensive testing will inevitably lead to more cases being detected.

The daily reported COVID-19 cases and deaths number should be used in combination with other factors including testing policies, number of tests performed, test positivity, excess mortality and rates of hospital and Intensive Care Unit (ICU) admissions, when analysing the epidemiological situation in a country. Most of these indicators are presented for EU/EEA Member States in the country Overview report.Even when using several indicators in combination, comparisons between countries should be done with caution and relevant epidemiological expertise.

**Variable Definition Code**

dateRep Date of reporting

“dd/mm/yyyy”

String

Day unit8

Month unit8

Year unit16

Cases Number of newly reported cases int64

Deaths Number of newly reported deaths int64

countriesAndterritories Name of the country or territory string

geoId 2-letter code string

countriesAndterritoryCode 3-letter ISO code string

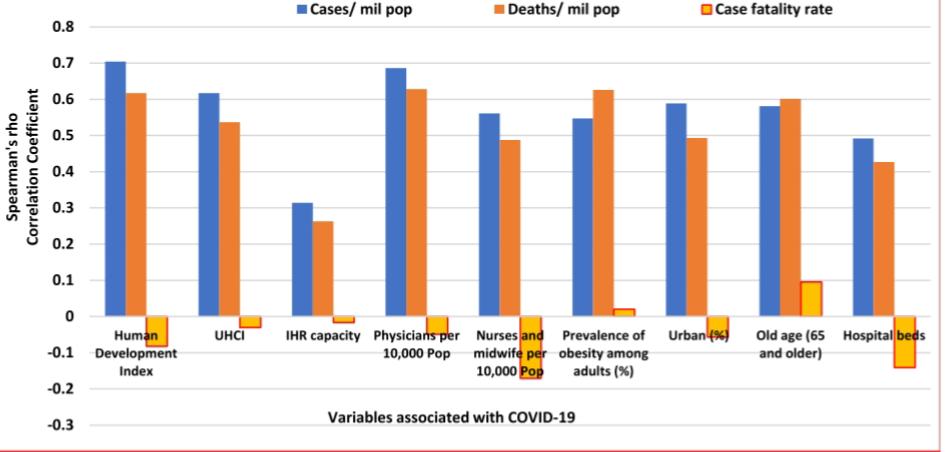
popData2020 Eurostat 2020 data int64

continentExp Name of the continent reporting string

**DATA SOURCE :**

Dataset: <https://www.kaggle.com/code/surajkumar88/covid19-case-study-analysis-viz-predictions>

**DATA VISUALIZATION :**

****

**PROGRAM :**

Pip install plotly –upgrade

Requirement already up-to-date: plotly in /opt/conda/envs/Python36/lib/python3.6/site-packages (4.10.0)

Requirement already satisfied, skipping upgrade: six in /opt/conda/envs/Python36/lib/python3.6/site-packages (from plotly) (1.12.0)

Requirement already satisfied, skipping upgrade: retrying>=1.3.3 in /opt/conda/envs/Python36/lib/python3.6/site-packages (from plotly) (1.3.3)

In [41]:

Import numpy as np # linear algebra

Import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

#Plotly Libraris

Import plotly.express as px

Import plotly.graph\_objects as go

#import plotly.figure\_factory as ff

#from plotly.colors import n\_colors

From plotly.subplots import make\_subplots

# Minmax scaler

From sklearn.preprocessing import MinMaxScaler

#itertools

Import itertools

#dataframe display settings

Pd.set\_option(‘display.max\_columns’, 5000000)

Pd.set\_option(‘display.max\_rows’, 50000000)

#to suppress un-necessary warnings

Import warnings

Warnings.filterwarnings(‘ignore’)

Importing Data

All datasets

In [42]:

Confirmed\_df = pd.read\_csv(‘https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/time\_series\_covid19\_confirmed\_global.csv’)

Deaths\_df = pd.read\_csv(‘https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/time\_series\_covid19\_deaths\_global.csv’)

Recoveries\_df = pd.read\_csv(‘https://raw.githubusercontent.com/CSSEG

Time Series of Basic Stats

In [46]:

Base\_stats = pd.DataFrame(columns=[‘Dates’,’Confirmed’,’Deaths’,’Recovered’,’Active’])

Base\_stats[‘Dates’] = confirmed\_df.columns[4:]

Base\_stats[‘Confirmed’] = base\_stats[‘Dates’].apply(lambda x: confirmed\_df[x].sum())

Base\_stats[‘Deaths’] = base\_stats[‘Dates’].apply(lambda x: deaths\_df[x].sum())

Base\_stats[‘Recovered’] = base\_stats[‘Dates’].apply(lambda x: recoveries\_df[x].sum())

Base\_stats.reset\_index(drop=False, inplace=True)

Base\_stats[‘Active’] = base\_stats[‘index’].apply(lambda x: (base\_stats[‘Confirmed’][x]-(base\_stats[‘Deaths’][x]+base\_stats[‘Recovered’][x])))

Base\_stats.head()

Out[46]:

Index Dates Confirmed Deaths Recovered Active

0 0 1/22/20 555 17 28 510

1 1 1/23/20 654 18 30 606

2 2 1/24/20 941 26 36 879

3 3 1/25/20 1434 42 39 1353

ISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/time\_series\_covid19\_recovered\_global.csv’)

Latest\_data = pd.read\_csv(‘https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_daily\_reports/08-22-2020.csv’)

Us\_medical\_data = pd.read\_csv(‘https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_daily\_reports\_us/08-22-2020.csv’)

#apple\_mobility = pd.read\_csv(‘https://covid19-static.cdn-apple.com/covid19-mobility-data/2015HotfixDev7/v3/en-us/applemobilitytrends-2020-08-21.csv’)

Preprocessing

In [43]:

Confirmed\_df.head()

Out[43]:

In [44]:

Confirmed\_group\_df = confirmed\_df.groupby(by=’Country/Region’,as\_index=False).sum()

Deaths\_group\_df = deaths\_df.groupby(by=’Country/Region’,as\_index=False).sum()

Recoveries\_group\_df = recoveries\_df.groupby(by=’Country/Region’,as\_index=False).sum()

Active\_group\_df = pd.DataFrame(columns=[confirmed\_group\_df.columns])

Active\_group\_df = deaths\_group\_df.copy()

For I in range(confirmed\_group\_df.shape[0]):

For j in range(3, confirmed\_group\_df.shape[1]):

Active\_group\_df.iloc[I,j] = confirmed\_group\_df.iloc[I,j]-(recoveries\_group\_df.iloc[I,j]+deaths\_group\_df.iloc[I,j])

In [45]:

Confirmed\_df.describe()

Deaths vs confirmed Cases for Top 20 Countries

Scatter\_ani\_df = pd.DataFrame(columns=[‘Dates’, ‘Country’,’Confirmed’,’Recovered’,’Deaths’])

Dates, Country, Confirmed, Deaths, Recovered = [],[],[],[],[]

For I in range(20):

Temp1 = [] Confirmed.extend(confirmed\_group\_sorted\_df[confirmed\_group\_sorted\_df.columns[3:]][i:i+1].T.values.tolist())

Dates.extend(confirmed\_group\_sorted\_df.columns[3:])

Temp1.append(confirmed\_group\_sorted\_df.iloc[I,0])

Temp = temp1\*(confirmed\_group\_sorted\_df.shape[1]-3)

Country.extend(temp)

Recovered.extend(recoveries\_group\_df.set\_index(recoveries\_group\_df[“Country/Region”], drop=True)[confirmed\_group\_sorted\_df.columns[3:]].loc[temp1].values.tolist()[0])

Deaths.extend(deaths\_group\_df.set\_index(deaths\_group\_df[“Country/Region”], drop=True)[confirmed\_group\_sorted\_df.columns[3:]].loc[temp1].values.tolist()[0])

Scatter\_ani\_df[‘Confirmed’] = pd.DataFrame(Confirmed)[0]

Scatter\_ani\_df[‘Dates’] = pd.DataFrame(Dates)[0]

Scatter\_ani\_df[‘Country’] = pd.DataFrame(Country)[0]

Scatter\_ani\_df[‘Recovered’] = pd.DataFrame(Recovered)[0]

Scatter\_ani\_df[‘Deaths’] = pd.DataFrame(Deaths)[0]

Fig = px.scatter(scatter\_ani\_df, x=”Confirmed”, y=”Deaths”, animation\_frame=”Dates”, animation\_group=”Country”,

Size=”Confirmed”, color=”Country”, hover\_name=”Country”,

#log\_x=True,

Size\_max=50, range\_x=[-10000,8000000], range\_y=[-10000,400000])

Fig.layout.updatemenus[0].buttons[0].args[1][“frame”][“duration”] = 50

Fig.show()

Number of countries affected over Time

Affected\_countries\_df = confirmed\_df.groupby(“Country/Region”).sum().drop([‘Lat’,’Long’],axis =1).apply(lambda x: x[x > 0].count(), axis =0)

Affected\_countries\_fig = go.Figure()

Affected\_countries\_fig.add\_trace(go.Scatter(x = base\_stats[‘Dates’],

Y = affected\_countries\_df,

Name = ‘Affected Countries’,

Mode=’lines’,

Line = dict(color=’#118ab2’),

Hovertemplate =’<br><b>Date</b>: %{x}’+’<br><i>No. of Countries </i>:’+’%{y}’,

))

Affected\_countries\_fig.update\_xaxes(showticklabels=False)

Affected\_countries\_fig.update\_layout(

#height=500, width=1100,

Title\_text=”Number of Countries Affected With COVID19”,

Title\_x=0.5, title\_font\_size=20,legend=dict(orientation=’h’,yanchor=’top’,y=1.12,xanchor=’right’,x=1)

Paper\_bgcolor=”mintcream”,

Xaxis\_title=”Date”, yaxis\_title=”Number of Countries”)

Affected\_countries\_fig.show()

Breakdown of Cases for Top 20 countries

Confirmed\_group\_melted\_df = pd.melt(confirmed\_group\_df, id\_vars=[‘Country/Region’], value\_vars=confirmed\_group\_df.columns[3:])

Confirmed\_group\_melted\_df.rename(columns={“variable”: “Dates”, “value”: “Confirmed”}, inplace=True)

Active\_group\_melted\_df = pd.melt(active\_group\_df, id\_vars=[‘Country/Region’], value\_vars=active\_group\_df.columns[3:])

Active\_group\_melted\_df.rename(columns={“variable”: “Dates”, “value”: “Count”}, inplace=True)

Recovered\_group\_melted\_df = pd.melt(recoveries\_group\_df, id\_vars=[‘Country/Region’], value\_vars=recoveries\_group\_df.columns[3:])

Recovered\_group\_melted\_df.rename(columns={“variable”: “Dates”, “value”: “Count”}, inplace=TrueTrue)

Deaths\_group\_melted\_df = pd.melt(deaths\_group\_df, id\_vars=[‘Country/Region’], value\_vars=deaths\_group\_df.columns[3:])

Deaths\_group\_melted\_df.rename(columns={“variable”: “Dates”, “value”: “Count”}, inplace=True)

Country\_specific\_fig = make\_subplots(specs=[[{“secondary\_y”: True}]])

Df\_dict={

“Confirmed”: [confirmed\_group\_melted\_df ,”#118ab2”],

“Active”: [active\_group\_melted\_df ,”#ef476f”],

“Recovered”: [recovered\_group\_melted\_df ,”#06d6a0”],

“Deaths”: [deaths\_group\_melted\_df ,”#073b4c”]

Forcountry in confirmed\_group\_df.sort\_values(by=confirmed\_group\_df.columns[-1], ascending=False)[‘Country/Region’].values.tolist()[:20]:

Country\_specific\_fig.add\_trace(go.Scatter(y=confirmed\_group\_df[confirmed\_group\_df[‘Country/Region’]==country][confirmed\_group\_df.columns[4:]].values.tolist()[0], X=confirmed\_group\_df.columns[4:],

Mode=’lines’, visible=(lambda x: True if x==”US” else False)(country),

Name=”Confirmed”, showlegend=True,

Line = dict(dash=”solid”, color=df\_dict[‘Confirmed’][1])

))

Country\_specific\_fig.add\_trace(go.Bar(y=confirmed\_group\_df[confirmed\_group\_df[‘Country/Region’]==country][confirmed\_group\_df.columns[4:]].T.reset\_index().reset\_index().rename(columns={confirmed\_group\_df[confirmed\_group\_df[‘Country/Region’]==country][confirmed\_group\_df.columns[4:]].T.reset\_index().reset\_index().columns[-1]:’count’,’index’:’dates’,’level\_0’:’index’})[‘index’].apply(lambda x: confirmed\_group\_df[confirmed\_group\_df[‘Country/Region’]==country][confirmed\_group\_df.columns[4:]].T.reset\_index().reset\_index().rename(columns={confirmed\_group\_df[confirmed\_group\_df[‘Country/Region’]==country][confirmed\_group\_df.columns[4:]].T.reset\_index().reset\_index().columns[-1]:’count’,’index’:’dates’,’level\_0’:’index’})[‘count’][x]-confirmed\_group\_df[confirmed\_group\_df[‘Country/Region’]==country][confirmed\_group\_df.columns[4:]].T.reset\_index().reset\_index().rename(columns={confirmed\_group\_df[confirmed\_group\_df[‘Country/Region’]==country][confirmed\_group\_df.columns[4:]].T.reset\_index().reset\_index().columns[-1]:’count’,’index’:’dates’,’level\_0’:’index’})[‘count’][x-1:x].sum()),

X=confirmed\_group\_df.columns[4:],

Name=”Daily Confirmed”, showlegend=True,

Visible=(lambda x: True if x==”US” else False)(country), Yaxis=’y2’, opacity=0.2))

# Trace for average moving

# country\_specific\_fig.add\_trace(go.Scatter(y=confirmed\_group\_melted\_df[confirmed\_group\_melted\_df[‘Country/Region’]==country].reset\_index(drop=True).reset\_index()[‘index’].apply(lambda x: (confirmed\_group\_melted\_df[confirmed\_group\_melted\_df[‘Country/Region’]==country].reset\_index(drop=True).reset\_index()[‘Confirmed’][x-7:x].sum())/7 if x>7 else (confirmed\_group\_melted\_df[confirmed\_group\_melted\_df[‘Country/Region’]==country].reset\_index(drop=True).reset\_index()[‘Confirmed’][0:x].sum())/7),#

X=confirmed\_group\_df.columns[4:],

#mode=’lines’, visible=False, name=country, showlegend=False,

#line = dict(dash=”dash”),

#hovertemplate = ‘<br><b>Date</b>: %{x}’+’<br><i>7-day moving avg.</i>: %{y}’,

# ))

For I in [“Active”, “Recovered”, “Deaths”]: Country\_specific\_fig.add\_trace(go.Scatter(y=df\_dict[i][0][df\_dict[i][0][‘Country/Region’]==country].reset\_index(drop=True).reset\_index()[‘index’].apply(lambda x: (df\_dict[i][0][df\_dict[i][0][‘Country/Region’]==country].reset\_index(drop=True).reset\_index()[‘Count’][x-7:x].sum())/7 if x>7 else (df\_dict[i][0][df\_dict[i][0][‘Country/Region’]==country].reset\_index(drop=True).reset\_index()[‘Count’][0:x].sum())/7), X=confirmed\_group\_df.columns[4:],

Mode=’lines’, visible=(lambda x: True if x==”US” else False)(country),

Name=I, showlegend=True,

Line = dict(dash=”solid”, color=df\_dict[i][1]),

#hovertemplate = ‘<br><i>’+i+’</i>: %{y:.2f}’,

))Country\_specific\_fig.update\_layout(

Updatemenus=[ Dict(

Buttons=list(

[dict(label = country,

Method = ‘update’,

Args = [{‘visible’: list(map(lambda x: True if 5\*index<=x<=5\*index+4 else False, list(range(100))))},

{‘title’: “Country :”+country+”<br>Position :”+str(index

‘showlegend’:True}]) for index, country in enumerate(confirmed\_group\_df.sort\_values(by=confirmed\_group\_df.columns[-1], ascending=False)[‘Country/Region’].values.tolist()[:20])

]),

Type = “dropdown”

Direction=”down”,

Pad={“r”: 0, “t”: 0},

Show active=True,

X=0,

Xanchor=”left”,

Y=1.2,

Yanchor=”top”

)

])

Country\_specific\_fig.update\_xaxes(showticklabels=False)

Country\_specific\_fig.update\_layout(

#height=500, width=1100,

Title\_text=”Number of Cases in top 50 Countries”,

Title\_x=0.5, title\_font\_size=15, paper\_bgcolor=”mintcream”, Legend=dict(orientation=’h’,yanchor=’top’,y=1.12,xanchor=’right’,x=1),

Yaxis\_title=”Number of Cases”, hovermode=’x unified’,

Xaxis=dict(title=’Dates <br> The Position of countries is solely based on No. of Confirmed Cases<br>Please use the dropdown to select the country of choice’))

Country\_specific\_fig.show()

CORRELATION MATRIX:

In [ ]:

Corr\_mat\_fig = go.Figure()

Corr\_mat\_fig = go.Figure(data=go.Splom(

Dimensions=[dict(label=’Confirmed’,

Values=base\_stats[‘Confirmed’]),

Dict(label=’Active’,

Values=base\_stats[‘Active’]),

Dict(label=’Recovered’

Values=base\_stats[‘Recovered’]),

Dict(label=’Deaths’,

Values=base\_stats[‘Deaths’])],

Text=base\_stats[‘Dates’],

Diagonal\_visible=False,

Marker=dict(color=’red’,

Showscale=False, # colors encode categorical variables

Line\_color=’white’, line\_width=0.5))

Corr\_mat\_fig.update\_layout(

#height=600, width=600,

Title\_text=”Correlation Matrix for types of Cases”,

Title\_x=0.5, title\_font\_size=15, paper\_bgcolor=”mintcream”, Legend=dict(orientation=’h’,yanchor=’top’,y=1.12,xanchor=’right’,x=1))

Corr\_mat\_fig.show()

Out[45]:

**PLOTTING SIMPLE PLOT GRAPH:**

‘*Y’*variable stores the ‘Daily Confirmed’ corona virus cases

‘*R’*variable stores the ‘Daily Recovered’ corona virus cases

‘D*’*variable stores the ‘Daily Deceased’ corona virus cases

And ‘X’ variable stores the ‘Date’

**PROGRAM :**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

data = pd.read\_csv('case\_time\_series.csv')

Y = data.iloc[61:,1].values

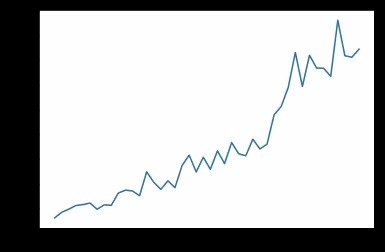
R = data.iloc[61:,3].values

D = data.iloc[61:,5].values

X = data.iloc[61:,0]

plt.plot(X,Y)

**OUTPUT :**



**COMPLETE CODE FOR ANALYSIS GRAPH:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

data = pd.read\_csv('case\_time\_series.csv')

Y = data.iloc[61:,1].values

R = data.iloc[61:,3].values

D = data.iloc[61:,5].values

X = data.iloc[61:,0]

plt.figure(figsize=(25,8))

ax = plt.axes()

ax.grid(linewidth=0.4, color='#8f8f8f')

ax.set\_facecolor("black")

ax.set\_xlabel('\nDate',size=25,color='#4bb4f2')

ax.set\_ylabel('Number of Confirmed Cases\n',

size=25,color='#4bb4f2')

ax.plot(X,Y,

color='#1F77B4',

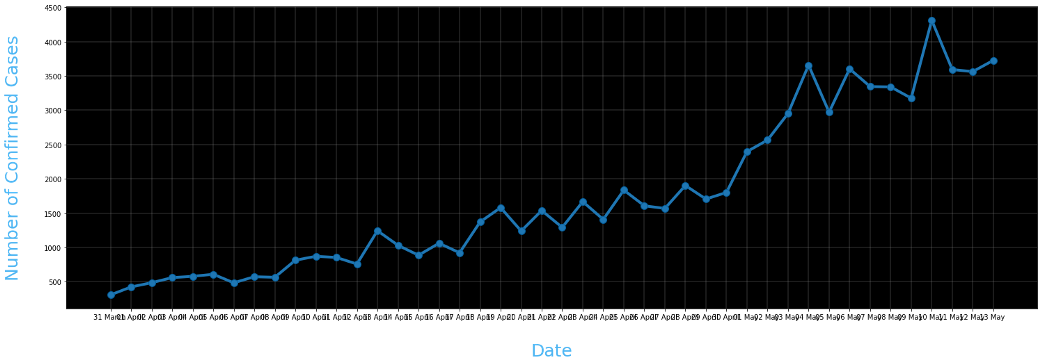
marker='o',

linewidth=4,

markersize=15,

markeredgecolor='#035E9B')

**OUTPUT :**



**Conclusion:**

The COVID-19 pandemic demonstrates that every country remains vulnerable to public health emer-Genies. The aspiration towards a healthier and safer society requires that countries develop and implement Coherent and context-specifc national strategy, improve governance of public health emergencies, build the capacity of their (public) health systems, minimize fragmentation, and tackle upstream structural issues, including socio-economic inequities. This is possible through a primary health care approach, which ensures provision of universal and equitable promotive, preventive and curative services, through whole-of-government and whole-of-society approaches.