COVID-19 Data Visualization and Cases Prediction with Python

Importing modules

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px

from prophet import Prophet
from sklearn.metrics import r2_score
print('modules are imported')
```

```
modules are imported
```

Loading the Dataset

In [2]:
```
plt.style.use("ggplot")

df0 = pd.read_csv("H:/UTD/COVID-19_Project/Dataset/CONVENIENT_global_con
df1 = pd.read_csv("H:/UTD/COVID-19_Project/Dataset/CONVENIENT_global_dea

print('data set loaded')
```

```
data set loaded
```

Let's check the dataframe

In [3]:
```
df0.head()
```

Out[3]:

	Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antarctica	Antigua and Barbuda
0	Province/State	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1/23/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1/24/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	1/25/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	1/26/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 290 columns

In [4]:
```
df1.head()
```

Out[4]:

	Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antarctica	Antigua and Barbuda
0	Province/State	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1/23/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1/24/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	1/25/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	1/26/20	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 290 columns

In [5]:
```
df0.tail()
```

Out[5]:

	Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antarctica	Antig a Barbu
1138	3/5/23	21.0	0.0	8.0	0.0	0.0	0.0	0
1139	3/6/23	16.0	0.0	0.0	0.0	0.0	0.0	0
1140	3/7/23	30.0	0.0	13.0	0.0	0.0	0.0	0
1141	3/8/23	15.0	16.0	4.0	15.0	11.0	0.0	0
1142	3/9/23	0.0	14.0	2.0	0.0	0.0	0.0	0

5 rows × 290 columns

In [6]:
```
df1.tail()
```

Out[6]:

	Country/Region	Afghanistan	Albania	Algeria	Andorra	Angola	Antarctica	Antig a Barbu
1138	3/5/23	0.0	0.0	0.0	0.0	0.0	0.0	0
1139	3/6/23	0.0	0.0	0.0	0.0	0.0	0.0	0
1140	3/7/23	0.0	0.0	0.0	0.0	0.0	0.0	0
1141	3/8/23	0.0	0.0	0.0	0.0	0.0	0.0	0
1142	3/9/23	0.0	0.0	0.0	0.0	0.0	0.0	0

5 rows × 290 columns

Let's check the shape of the dataframe

In [7]:
```
df0.shape
```

Out[7]:
```
(1143, 290)
```

In [8]:
```
df1.shape
```

Out[8]:
```
(1143, 290)
```

Let's do some preprocessing

Let's prepare new data by combining the above datasets and then we will visualize a geographical plot of the data to see what we are going to work with:

In [11]:
```
world = pd.DataFrame({"Country":[],"Cases":[]})
world["Country"] = df0.iloc[:,1:].columns
cases = []
for i in world["Country"]:
    cases.append(pd.to_numeric(df0[i][1:]).sum())
world["Cases"]=cases

country_list=list(world["Country"].values)
idx = 0
for i in country_list:
    sayac = 0
    for j in i:
        if j==".":
            i = i[:sayac]
            country_list[idx]=i
        elif j=="(":
            i = i[:sayac-1]
            country_list[idx]=i
        else:
            sayac += 1
    idx += 1
world["Country"]=country_list
world = world.groupby("Country")["Cases"].sum().reset_index()
continent=pd.read_csv("H:/UTD/COVID-19_Project/Dataset/continents2.csv")
continent["name"]=continent["name"].str.upper()
```

Let's check the dataframe

In [13]:
```
world.head()
```

Out[13]:

	Country	Cases
0	Afghanistan	209451.0
1	Albania	334457.0
2	Algeria	271496.0
3	Andorra	47890.0
4	Angola	105288.0

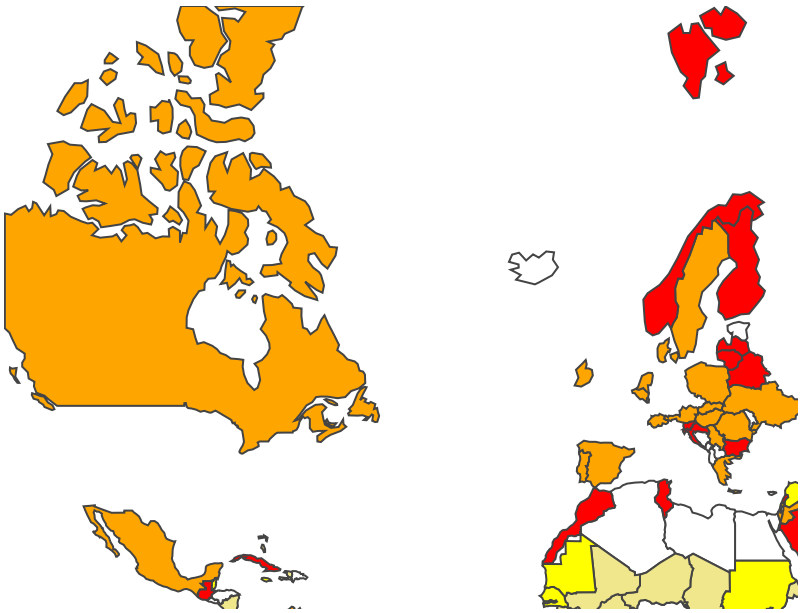
Let's visualize the data

Now here I will prepare three visualizations. One will be a geographical visualization to visualize the worldwide spread of Covid-19. Then the next visualization will be to have a look at the daily cases of Covid-19 in the world. Then the last visualization will be to have a look at the daily death cases of Covid-19 in the world.

Now let’s start data visualization by looking at the worldwide spread of Covid-19:

In [14]:

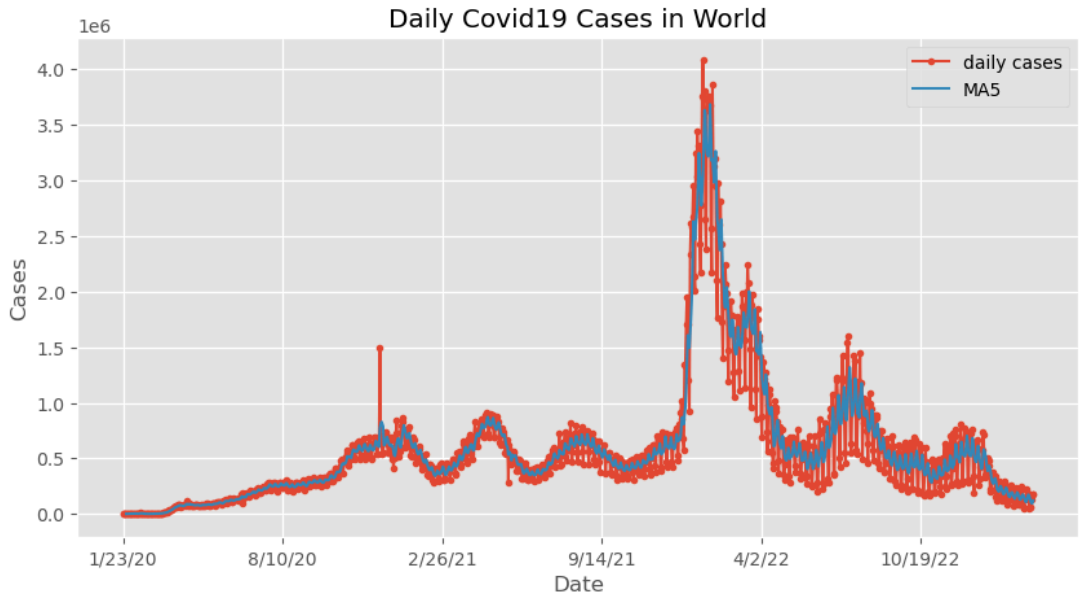
```
world["Cases Range"]=pd.cut(world["Cases"],[-150000,50000,200000,800000,
alpha =[]
for i in world["Country"].str.upper().values:
    if i == "BRUNEI":
        i="BRUNEI DARUSSALAM"
    elif  i=="US":
        i="UNITED STATES"
    if len(continent[continent["name"]==i]["alpha-3"].values)==0:
        alpha.append(np.nan)
    else:
        alpha.append(continent[continent["name"]==i]["alpha-3"].values[0
world["Alpha3"]=alpha

fig = px.choropleth(world.dropna(),
                    locations="Alpha3",
                    color="Cases Range",
                     projection="mercator",
                     color_discrete_sequence=["white","khaki","yellow","o
fig.update_geos(fitbounds="locations", visible=False)
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```



Now let's have a look at the daily cases all around the world:

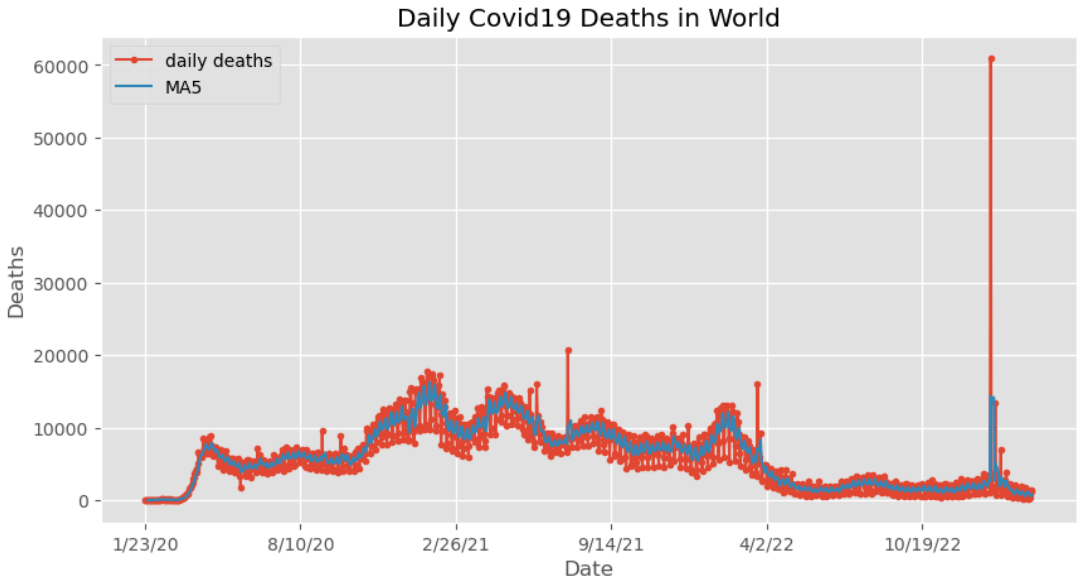
In [15]:

```
count = []
for i in range(1,len(df0)):
    count.append(sum(pd.to_numeric(df0.iloc[i,1:].values)))

df = pd.DataFrame()
df["Date"] = df0["Country/Region"][1:]
df["Cases"] = count
df=df.set_index("Date")

count = []
for i in range(1,len(df1)):
    count.append(sum(pd.to_numeric(df1.iloc[i,1:].values)))

df["Deaths"] = count

df.Cases.plot(title="Daily Covid19 Cases in World",marker=".",figsize=(1
df.Cases.rolling(window=5).mean().plot(figsize=(10,5),label="MA5")
plt.ylabel("Cases")
plt.legend()
plt.show()
```



Now let's have a look at the daily death cases of Covid-19:

In [16]:
```
df.Deaths.plot(title="Daily Covid19 Deaths in World",marker=".",figsize=
df.Deaths.rolling(window=5).mean().plot(figsize=(10,5),label="MA5")
plt.ylabel("Deaths")
plt.legend()
plt.show()
```



Covid-19 Cases Prediction with Python for Next 30 Days

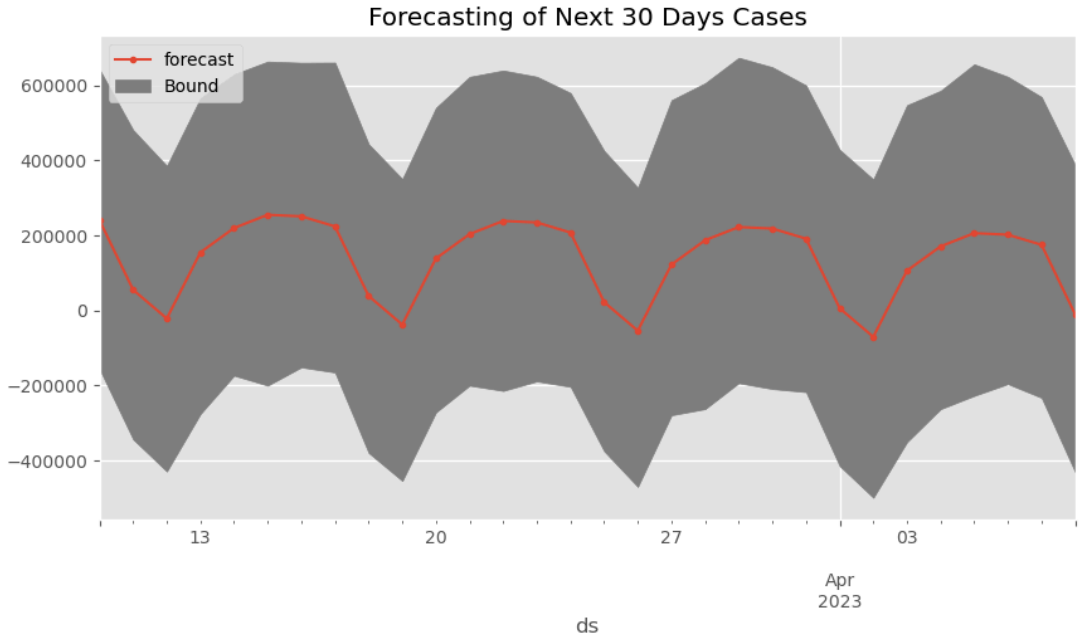
Now, I will use the Facebook prophet model for the task of Covid-19 cases prediction with Python for the next 30 days. Facebook prophet model uses time series method for forecasting.

Let's see how we can use the Facebook prophet model for Covid-19 cases prediction with Python for the next 30 days:

In [17]:

```
class Fbprophet(object):
    def fit(self,data):

        self.data  = data
        self.model = Prophet(weekly_seasonality=True,daily_seasonality=F
        self.model.fit(self.data)

    def forecast(self,periods,freq):

        self.future = self.model.make_future_dataframe(periods=periods,f
        self.df_forecast = self.model.predict(self.future)

    def plot(self,xlabel="Years",ylabel="Values"):

        self.model.plot(self.df_forecast,xlabel=xlabel,ylabel=ylabel,fig
        self.model.plot_components(self.df_forecast,figsize=(9,6))

    def R2(self):
        return r2_score(self.data.y, self.df_forecast.yhat[:len(df)])

df_fb  = pd.DataFrame({"ds":[],"y":[]})
df_fb["ds"] = pd.to_datetime(df.index)
df_fb["y"]  = df.iloc[:,0].values

model = Fbprophet()
model.fit(df_fb)
model.forecast(30,"D")
model.R2()

forecast = model.df_forecast[["ds","yhat_lower","yhat_upper","yhat"]].ta
forecast["yhat"].plot(marker=".",figsize=(10,5))
plt.fill_between(x=forecast.index, y1=forecast["yhat_lower"], y2=forecas
plt.legend(["forecast","Bound"],loc="upper left")
plt.title("Forecasting of Next 30 Days Cases")
plt.show()
```

```
C:\Users\Xploit\AppData\Local\Temp\ipykernel_3152\738480942.py:22: User
Warning:

Could not infer format, so each element will be parsed individually, fa
lling back to `dateutil`. To ensure parsing is consistent and as-expect
ed, please specify a format.

11:26:46 - cmdstanpy - INFO - Chain [1] start processing
11:26:46 - cmdstanpy - INFO - Chain [1] done processing
```


Forecasting of Next 30 Days Cases
forecast
Bound
600000
400000
200000
0
−200000
−400000
13
20
27
03
Apr
2023
ds