

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly
import plotly.express as px

In [2]: df = pd.read_csv('covid_19_clean_complete.csv', parse_dates=['date'])
df
Out[2]:
   Province/State   Country/Region   Lat   Long   Date   Confirmed   Deaths   Recovered   Active   WHO Region
0   NaN             Afghanistan     33.939110  67.709953  2020-01-22   0         0         0         0   Eastern Mediterranean
1   NaN             Albania         41.515390  20.169300  2020-01-22   0         0         0         0   Europe
2   NaN             Algeria         28.033900  1.659600  2020-01-22   0         0         0         0   Africa
3   NaN             Andorra         42.506300  1.521800  2020-01-22   0         0         0         0   Europe
4   NaN             Angola          -11.202700  17.873900  2020-01-22   0         0         0         0   Africa
...  ...
49063  NaN   Sao Tome and Principe   0.186400  6.613100  2020-07-27   865       14       734       117   Africa
49064  NaN             Yemen        15.552727  48.516388  2020-07-27   1691      483      833      375   Eastern Mediterranean
49065  NaN             Comoros     -11.645000  43.333300  2020-07-27   354         7       328       19   Africa
49066  NaN             Tajikistan   38.861000  71.276100  2020-07-27   7235        60      6028      1147   Europe
49067  NaN             Lesotho     -29.610000  28.233600  2020-07-27   505        12       128       365   Africa
49068 rows x 10 columns

In [3]: df.rename(columns={'Province/State':'State','Country/Region':'Country'}, inplace=True)
df
Out[4]:
   State   Country   Lat   Long   Date   Confirmed   Deaths   Recovered   Active   WHO Region
0   NaN   Afghanistan  33.939110  67.709953  2020-01-22   0         0         0         0   Eastern Mediterranean
1   NaN   Albania      41.515390  20.169300  2020-01-22   0         0         0         0   Europe
2   NaN   Algeria      28.033900  1.659600  2020-01-22   0         0         0         0   Africa
3   NaN   Andorra      42.506300  1.521800  2020-01-22   0         0         0         0   Europe
4   NaN   Angola       -11.202700  17.873900  2020-01-22   0         0         0         0   Africa
...  ...
49063  NaN   Sao Tome and Principe  0.186400  6.613100  2020-07-27   865       14       734       117   Africa
49064  NaN   Yemen       15.552727  48.516388  2020-07-27   1691      483      833      375   Eastern Mediterranean
49065  NaN   Comoros     -11.645000  43.333300  2020-07-27   354         7       328       19   Africa
49066  NaN   Tajikistan  38.861000  71.276100  2020-07-27   7235        60      6028      1147   Europe
49067  NaN   Lesotho    -29.610000  28.233600  2020-07-27   505        12       128       365   Africa
49068 rows x 10 columns

In [5]: df['Country'].nunique()
Out[5]:
187

In [6]: df['Active'] = df['Confirmed'] - df['Deaths'] - df['Recovered']

In [7]: #For plotting the cases on last value in date
top = df[df['date'] == df['date'].max()][0]
top
Out[7]:
   State   Country   Lat   Long   Date   Confirmed   Deaths   Recovered   Active   WHO Region
0   NaN   Afghanistan  33.939110  67.709953  2020-07-27   865      1396      2536      976   Eastern Mediterranean
1   NaN   Albania      41.515390  20.169300  2020-07-27   4880      164      3746      196   Europe
2   NaN   Algeria      28.033900  1.659600  2020-07-27   27973      1163      18837      7973   Africa
48810  NaN   Andorra      42.506300  1.521800  2020-07-27   907        52      803        52   Europe
48811  NaN   Angola       -11.202700  17.873900  2020-07-27   950         41      242      667   Africa
...  ...
49063  NaN   Sao Tome and Principe  0.186400  6.613100  2020-07-27   865       14       734       117   Africa
49064  NaN   Yemen       15.552727  48.516388  2020-07-27   1691      483      833      375   Eastern Mediterranean
49065  NaN   Comoros     -11.645000  43.333300  2020-07-27   354         7       328       19   Africa
49066  NaN   Tajikistan  38.861000  71.276100  2020-07-27   7235        60      6028      1147   Europe
49067  NaN   Lesotho    -29.610000  28.233600  2020-07-27   505        12       128       365   Africa
49068 rows x 10 columns

In [8]: w = top.groupby('Country')[['Confirmed', 'Active', 'Deaths']].sum().reset_index()
w
Out[8]:
   Country   Confirmed   Active   Deaths
0   Afghanistan      36263      9796      1269
1   Albania          4880      1991      144
2   Algeria          27973      7973      1163
3   Andorra           907         52         62
4   Angola           950         667         41
...  ...
182  West Bank and Gaza  10621      6791         78
183  Western Sahara         10         1         1
184   Yemen          1691      375      483
185   Zambia          4592      1597      140
186   Zimbabwe         2704      2126         36
187 rows x 4 columns

In [9]: a = df.groupby('Country')[['Confirmed', 'Active', 'Deaths', 'Recovered']].sum().reset_index()
a
Out[9]:
   Country   Confirmed   Active   Deaths   Recovered
0   Afghanistan      1936380      1089052      49098      790240
1   Albania          196702      72117      5708      118877
2   Algeria          1179755      345886      77972      755897
3   Andorra          94404      19907      5423      69074
4   Angola           22662      15011      1078      6573
...  ...
182  West Bank and Gaza      233461      170967      1370      61134
183  Western Sahara           901         190         63      648
184   Yemen          67180      2594      17707      23779
185   Zambia          129421      43167      2643      83611
186   Zimbabwe          50794      37706      881      12207
187 rows x 5 columns

In [10]: fig = choropleth(a, locations = 'Country', locationmode = 'country names', color = 'Active', hover_name = 'Country',
                        range_color = [1, 500], color_continuous_scale = 'magma', title = 'Active cases Countries')
fig.show()

Active cases Countries



In [11]: #plot for confirmed cases
plt.figure(figsize=(15,10))
#t_cases = df.groupby('date')[['Confirmed']].sum().reset_index()
#t_cases['date'] = pd.to_datetime(t_cases['date'])
e = sns.pointplot(x = t_cases.Date, y = t_cases.Confirmed, color = 'b')
e.set(xlabel = 'Dates', ylabel = 'Cases total')
plt.xticks(rotation=90, fontsize=18)
plt.yticks(fontsize=15)
plt.xlabel('Dates', fontsize=18)
plt.ylabel('Total Cases', fontsize=38)
Text(0, 8.5, 'Total Cases')

Out[11]:


In [98]: #Top 20 countries having most active cases
t_active = a.groupby('Country')[['Active']].sum().sort_values(ascending=False).head(20).reset_index()
t_active
Out[98]:
   Country   Active
0   US      156981121
1   Brazil      210940009
2   United Kingdom      22624595
3   Russia      18698578
4   India      1587913
5   France      10995287
6   Spain      8771420
7   Canada      8565805
8   Peru      7748987
9   Italy      7363518
10  Pakistan      5633262
11  South Africa      5156341
12  Bangladesh      4924394
13  Netherlands      4528235
14  Sweden      4524247
15  Turkey      4092900
16  Colombia      3923706
17  Saudi Arabia      3783704
18  Belgium      3699945
19  Chile      3202581

In [113]: plt.figure(figsize=(15,10))
plt.title('Top 20 countries having most active cases', fontsize=30)
f = sns.barplot(x = t_active.Active, y = t_active.Country)
f.set(xlabel = 'Total Cases', ylabel = 'Countries')
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.xlabel('Total Cases', fontsize=20)
plt.ylabel('Country', fontsize=20)
Text(0, 8.5, 'Country')

Top 20 countries having most active cases



In [109]: #Top 20 countries having most deaths
t_deaths = a.groupby('Country')[['Deaths']].sum().sort_values(ascending=False).head(20).reset_index()
t_deaths
Out[109]:
   Country   Deaths
0   US      1031141
1   United Kingdom      399775
2   Brazil      259844
3   Spain      270717
4   France      240424
5   Spain      240424
6   Mexico      238277
7   India      1111831
8   Iran      1024136
9   Belgium      963679
10  Germany      871322
11  Canada      699566
12  China      672413
13  Peru      625113
14  Netherlands      622314
15  Russia      615385
16  Turkey      466056
17  Sweden      448913
18  Ecuador      346018
19  Chile      322840

In [113]: plt.figure(figsize=(15,10))
plt.title('Top 20 countries having most death', fontsize=30)
g = sns.barplot(x=t_deaths.Deaths, y=t_deaths.Country)
for i, (value, name) in enumerate(zip(t_deaths.Deaths, t_deaths.Country)):
    # f = f(value, i, 0.5, f'(value: {v}), size=10, ha='left', va='center')
    f = f(value, i, 0.5, f'(value: {v}), size=10, ha='left', va='center')
f.set(xlabel='Total Cases', ylabel='Country')
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.xlabel('Total Cases', fontsize=20)
plt.ylabel('Country', fontsize=20)
Text(0, 8.5, 'Country')

Top 20 countries having most death



In [116]: recover = a.groupby('Country')[['Recovered']].sum().sort_values(ascending=False).head(20).reset_index()
recover
Out[116]:
   Country   Recovered
0   US      9503445
1   Brazil      54402073
2   Russia      25120448
3   India      23783720
4   Germany      1710739
5   Italy      15678910
6   Iran      15200895
7   Spain      15085883
8   Turkey      12340389
9   China      12020983
10  China      11468656
11  Mexico      11141225
12  Peru      1062846
13  Saudi Arabia      8474207
14  France      7182115
15  Pakistan      6938003
16  South Africa      5836423
17  Qatar      4676443
18  Bangladesh      2917402
19  Switzerland      2675862

In [117]: plt.figure(figsize=(15,10))
plt.title('Top 20 countries having most recovered', fontsize=30)
f = sns.barplot(x=recover.Recovered, y=recover.Country)
for i, (value, name) in enumerate(zip(recover.Recovered, recover.Country)):
    # f = f(value, i, 0.5, f'(value: {v}), size=10, ha='left', va='center')
    # f = f(value, i, 0.5, f'(value: {v}), size=10, ha='left', va='center')
f.set(xlabel='Total Cases', ylabel='Country')
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.xlabel('Total Cases', fontsize=20)
plt.ylabel('Country', fontsize=20)
Text(0, 8.5, 'Country')

Top 20 countries having most recovered



In [118]: Brazil = df[df['Country'] == 'Brazil']
Brazil = df.groupby('date')[['Recovered', 'Deaths', 'Confirmed', 'Active']].sum().reset_index()
Brazil = Brazil.iloc[9].reset_index().drop('index', axis=1)
C:\Users\NVP\AppData\Local\Temp\ipykernel_11840\3425565344.py:2: FutureWarning:
Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

In [119]: India = df[df['Country'] == 'India']
India = India.groupby('date')[['Recovered', 'Deaths', 'Confirmed', 'Active']].sum().reset_index()
India = India.iloc[9].reset_index().drop('index', axis=1)
C:\Users\NVP\AppData\Local\Temp\ipykernel_11840\3860180568.py:2: FutureWarning:
Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

In [120]: US = df[df['Country'] == 'US']
US = US.groupby('date')[['Recovered', 'Deaths', 'Confirmed', 'Active']].sum().reset_index()
US = US.iloc[9].reset_index().drop('index', axis=1)
C:\Users\NVP\AppData\Local\Temp\ipykernel_11840\3864338498.py:2: FutureWarning:
Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

In [122]: plt.figure(figsize=(15,10))
sns.pointplot(x=Brazil.index, y=Brazil.Confirmed, color='blue')
sns.pointplot(x=US.index, y=US.Confirmed, color='pink')
sns.pointplot(x=India.index, y=India.Confirmed, color='red')
plt.xlabel('No. of days', fontsize=20)
plt.ylabel('Confirmed cases', fontsize=20)
plt.title('Confirmed cases over time', fontsize=38)
plt.show()

Confirmed cases over time



In [123]: #Pip Prophet
!pip install prophet
Collecting prophet
  Downloading prophet-1.1.5-py3-none-win_and4.whl (13.3 MB)
Requirement already satisfied: python-dateutil<2.8.1 in c:\users\NVP\AppData\Local\Temp\ipykernel_123\site-packages (from prophet) (1.4.2)
Collecting cymath==0.4
  Downloading cymath-0.4.0-py3-none-any.whl (90 kB)
Requirement already satisfied: numpy>=1.15.4 in c:\users\NVP\AppData\Local\Temp\ipykernel_123\site-packages (from prophet) (1.24.4)
Collecting importlib-resources
  Downloading importlib-resources-4.1.3-py3-none-any.whl (34 kB)
Requirement already satisfied: pandas<=1.0.4 in c:\users\NVP\AppData\Local\Temp\ipykernel_123\site-packages (from prophet) (1.64.1)
Requirement already satisfied: packaging>=20.0 in c:\users\NVP\AppData\Local\Temp\ipykernel_123\site-packages (from prophet) (22.1)
Requirement already satisfied: kaldiio==0.2.0 in c:\users\NVP\AppData\Local\Temp\ipykernel_123\site-packages (from prophet) (0.4.3)
Requirement already satisfied: colorama in c:\users\NVP\AppData\Local\Temp\ipykernel_123\site-packages (from prophet) (0.4.5)
Requirement already satisfied: six>=1.5 in c:\users\NVP\AppData\Local\Temp\ipykernel_123\site-packages (from python-dateutil<2.8.1->prophet) (1.16.0)
Installing collected packages: six, importlib-resources, holidays, cymath, prophet
Successfully installed cymath-0.4.0 holidays-0.4.0 importlib-resources-4.1.3 prophet-1.1.5 six-1.16.0
In [124]: from prophet import Prophet

In [125]: df.head()
Out[125]:
   State   Country   Lat   Long   Date   Confirmed   Deaths   Recovered   Active   WHO Region
1   NaN   Albania      41.515390  20.169300  2020-01-22   0         0         0         0   Europe
2   NaN   Algeria      28.033900  1.659600  2020-01-22   0         0         0         0   Africa
3   NaN   Andorra      42.506300  1.521800  2020-01-22   0         0         0         0   Europe
4   NaN   Angola       -11.202700  17.873900  2020-01-22   0         0         0         0   Africa

In [127]: df.groupby('date').sum().head()
Out[127]:
   Date
2020-01-23  5594.20365  6140.869714   955      17      28      510
2020-01-23  5594.20365  6140.869714   654      18      30      606
2020-01-24  5594.20365  6140.869714   941      26      36      879
2020-01-25  5594.20365  6140.869714   1434      42      39      1353
2020-01-26  5594.20365  6140.869714   2118      56      52      2010

In [128]: total_active = df['Active'].sum()
print('Total no. of active COVID 19 CASES', total_active)
Total no. of active COVID 19 CASES 396753590

In [138]: confirmed = df.groupby('date').sum()[['Confirmed']].reset_index()
deaths = df.groupby('date').sum()[['Deaths']].reset_index()
recovered = df.groupby('date').sum()[['Recovered']].reset_index()
confirmed.tail()
Out[138]:
   Date   Confirmed
183  2020-07-23  1550481
184  2020-07-24  15704545
185  2020-07-25  16047190
186  2020-07-26  16251796
187  2020-07-27  16480485

In [133]: deaths.tail()
Out[133]:
   Date   Deaths
183  2020-07-23  633506
184  2020-07-24  638660
185  2020-07-25  644517
186  2020-07-26  648621
187  2020-07-27  654038

In [132]: recovered.tail()
Out[132]:
   Date   Recovered
183  2020-07-23  8710669
184  2020-07-24  8929705
185  2020-07-25  9158743
186  2020-07-26  9293464
187  2020-07-27  9468087

In [133]: #Forecasting confirmed cases
confirmed.columns = ['ds', 'y']
confirmed['ds'] = pd.to_datetime(confirmed['ds'])
confirmed.tail()
Out[133]:
   ds   y
183  2020-07-23  1550481
184  2020-07-24  15704545
185  2020-07-25  16047190
186  2020-07-26  16251796
187  2020-07-27  16480485

In [134]: m = Prophet(interval_width=0.95)
m.fit(confirmed)
02:39:17 c:\python - INFO Chain [1] start processing
02:39:20 c:\python - INFO Chain [1] done processing
<prophet.forecaster.Prophet at 0x228f6521ed0>
Out[134]:
future = m.make_future_dataframe(periods=7, freq='D')
future.tail()
Out[135]:
   ds
190  2020-07-30
191  2020-07-31
192  2020-08-01
193  2020-08-02
194  2020-08-03

In [138]: forecast = m.predict(future)
forecast['ds', 'yhat', 'yhat_lower', 'yhat_upper'].tail()
Out[138]:
   ds   yhat   yhat_lower   yhat_upper
190  2020-07-31  1.674820e+07  1.658910e+07  1.691050e+07
191  2020-07-31  1.688680e+07  1.669680e+07  1.708680e+07
192  2020-07-31  1.717125e+07  1.700440e+07  1.734010e+07
193  2020-08-02  1.756410e+07  1.738440e+07  1.774380e+07
194  2020-08-03  1.776400e+07  1.758470e+07  1.794330e+07

In [139]: confirmed_forecast_plot = m.plot(forecast)


In [140]: confirmed_forecast_plot = m.plot_components(forecast)


In [1]:
```