COVID-19 Interactive Analysis Dashboard

```
# importing libraries

from __future__ import print_function
from ipywidgets import interact, interactive, fixed, interact_manual
from IPython.core.display import display, HTML

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import folium
import plotly.graph_objects as go
import seaborn as sns
import ipywidgets as widgets
```

```
# loading data right from the source:
death_df = pd.read_csv("/content/time_series_covid19_deaths_global.csv")
confirmed_df = pd.read_csv('/content/time_series_covid19_confirmed_global.csv')
recovered_df = pd.read_csv('/content/time_series_covid19_recovered_global.csv')
country_df = pd.read_csv('/content/worldometer_coronavirus_summary_data.csv')
```

```
confirmed_df.head()
```

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	
1	NaN	Albania	41.15330	20.168300	0	0	0	
2	NaN	Algeria	28.03390	1.659600	0	0	0	
3	NaN	Andorra	42.50630	1.521800	0	0	0	
4	NaN	Angola	-11.20270	17.873900	0	0	0	

5 rows × 1147 columns

```
recovered_df.head()
```

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	
1	NaN	Albania	41.15330	20.168300	0	0	0	
2	NaN	Algeria	28.03390	1.659600	0	0	0	
3	NaN	Andorra	42.50630	1.521800	0	0	0	
4	NaN	Angola	-11.20270	17.873900	0	0	0	

5 rows × 1147 columns

```
death_df.head()
```

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	
1	NaN	Albania	41.15330	20.168300	0	0	0	
2	NaN	Algeria	28.03390	1.659600	0	0	0	
3	NaN	Andorra	42.50630	1.521800	0	0	0	
4	NaN	Angola	-11.20270	17.873900	0	0	0	

5 rows × 1147 columns

```
country_df.head()
```

	country	continent	total_confirmed	total_deaths	total_recovered	active_case
0	Afghanistan	Asia	179267	7690.0	162202.0	9375
1	Albania	Europe	275574	3497.0	271826.0	251
2	Algeria	Africa	265816	6875.0	178371.0	80570
3	Andorra	Europe	42156	153.0	41021.0	982
4	Angola	Africa	99194	1900.0	97149.0	145

```
print(country_df.columns)
```

```
Index(['country', 'continent', 'total_confirmed', 'total_deaths',
       'total_recovered', 'active_cases', 'serious_or_critical',
       'total_cases_per_1m_population', 'total_deaths_per_1m_population',
       'total_tests', 'total_tests_per_1m_population', 'population'],
      dtype='object')
```

```
# data cleaning

# renaming the df column names to lowercase
country_df.columns = map(str.lower, country_df.columns)
confirmed_df.columns = map(str.lower, confirmed_df.columns)
death_df.columns = map(str.lower, death_df.columns)
recovered_df.columns = map(str.lower, recovered_df.columns)

# changing province/state to state and country/region to country
confirmed_df = confirmed_df.rename(columns={'province/state': 'state', 'country/region': 'country'})
recovered_df = confirmed_df.rename(columns={'province/state': 'state', 'country/region': 'country'})
death_df = death_df.rename(columns={'province/state': 'state', 'country/region': 'country'})
country_df = country_df.rename(columns={'country_region': 'country'})
# country_df.head()
```

```
print(country_df.columns)
```

```
Index(['country', 'continent', 'total_confirmed', 'total_deaths',
       'total_recovered', 'active_cases', 'serious_or_critical',
       'total_cases_per_1m_population', 'total_deaths_per_1m_population',
       'total_tests', 'total_tests_per_1m_population', 'population'],
      dtype='object')
```

```
print(confirmed_df.columns)
```

```
Index(['state', 'country', 'lat', 'long', '1/22/20', '1/23/20', '1/24/20',
       '1/25/20', '1/26/20', '1/27/20',
       ...
```

```
           '2/28/23', '3/1/23', '3/2/23', '3/3/23', '3/4/23', '3/5/23', '3/6/23',
           '3/7/23', '3/8/23', '3/9/23'],
          dtype='object', length=1147)
```

```
# total number of confirmed, death and recovered cases
total_confirmed= int(country_df["total_confirmed"].sum())
total_deaths = int(country_df['total_deaths'].sum())
total_recovered = int(country_df['total_recovered'].sum())



# displaying the total stats

display(HTML("<div style = 'background-color: #090C02; padding: 30px '>" +
             "<span style='color: #FFE5D9; font-size:30px;'> Confirmed: "  + str(total_confirmed) +"</span>
             "<span style='color: #ED1C24; font-size:30px;margin-left:20px;'> Deaths: " + str(total_deaths)
             "<span style='color: #D8E2DC; font-size:30px; margin-left:20px;'> Recovered: " + str(total_rec
             "</div>")
       )
```

Confirmed: 521077152 Deaths: 6288083
Recovered: 460397633

```
# sorting the values by confirmed descednding order
# country_df.sort_values('confirmed', ascending= False).head(10).style.background_gradient(cmap='copper')
fig = go.FigureWidget( layout=go.Layout() )
def highlight_col(x):
    r = 'background-color: red'
    y = 'background-color: purple'
    g = 'background-color: grey'
    df1 = pd.DataFrame('', index=x.index, columns=x.columns)
    df1.iloc[:, 4] = y
    df1.iloc[:, 5] = r
    df1.iloc[:, 6] = g

    return df1

def show_latest_cases(n):
    n = int(n)
    return country_df.sort_values('total_confirmed', ascending= False).head(n).style.apply(highlight_col, a

interact(show_latest_cases, n='10')

ipywLayout = widgets.Layout(border='solid 2px green')
ipywLayout.display='none' # uncomment this, run cell again - then the graph/figure disappears
widgets.VBox([fig], layout=ipywLayout)
```

n 10

	country	continent	total_confirmed	total_deaths	total_recovered	active_c
216	USA	North America	84209473	1026646.000000	81244260.000000	1938567.00
94	India	Asia	43121599	524214.000000	42579693.000000	17692.00
26	Brazil	South America	30682094	664920.000000	29718402.000000	298772.00
72	France	Europe	29160802	147257.000000	28156674.000000	856871.00
78	Germany	Europe	25780226	137919.000000	23956700.000000	1685607.00
212	UK	Europe	22159805	176708.000000	21677896.000000	305201.00
165	Russia	Europe	18260293	377670.000000	17647179.000000	235444.00
189	South Korea	Asia	17782061	23709.000000	nan	
101	Italy	Europe	17057873	165244.000000	15894511.000000	998118.00
209	Turkey	Asia	15053168	98890.000000	14951238.000000	3040.00

```
sorted_country_df = country_df.sort_values('total_confirmed', ascending= False)
```

Slide to check for the worst hit countries

```
 # plotting the 20 worst hit countries

def bubble_chart(n):
    fig = px.scatter(sorted_country_df.head(n), x="country", y="total_confirmed", size="total_confirmed", c
               hover_name="country", size_max=60)
    fig.update_layout(
    title=str(n) +" Worst hit countries",
    xaxis_title="Countries",
    yaxis_title="total_Confirmed Cases",
    width = 700
    )
    fig.show();

interact(bubble_chart, n=10)

ipywLayout = widgets.Layout(border='solid 2px green')
ipywLayout.display='none'
widgets.VBox([fig], layout=ipywLayout)
```

n 10

```
from google.colab import output
output.enable_custom_widget_manager()
```

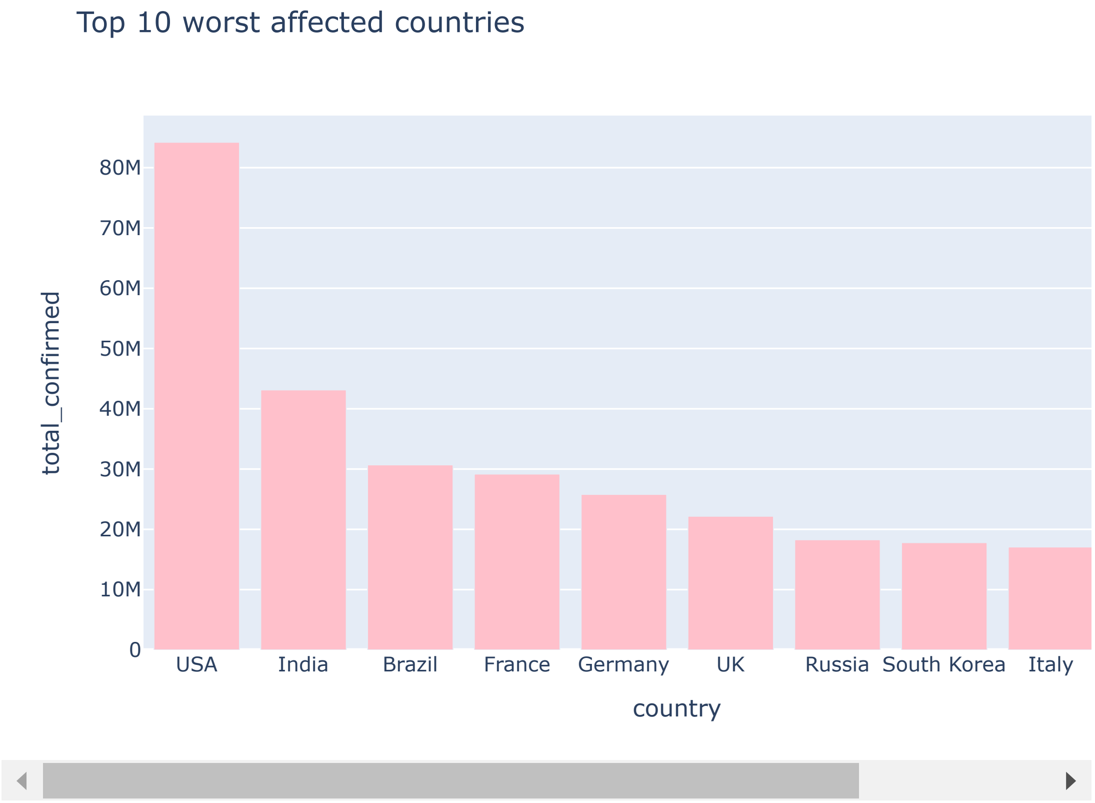
Support for third party widgets will remain active for the duration of the session. To disable support:

```
from google.colab import output
output.disable_custom_widget_manager()
```

```
def plot_cases_of_a_country(country):
    labels = ['tota_confirmed', 'total_deaths']
    colors = ['blue', 'red']
    mode_size = [6, 8]
    line_size = [4, 5]

    df_list = [confirmed_df, death_df]

    fig = go.Figure();

    for i, df in enumerate(df_list):
        if country == 'World' or country == 'world':
            x_data = np.array(list(df.iloc[:, 20:].columns))
            y_data = np.sum(np.asarray(df.iloc[:,4:]),axis = 0)

        else:
            x_data = np.array(list(df.iloc[:, 20:].columns))
            y_data = np.sum(np.asarray(df[df['country'] == country].iloc[:,20:]),axis = 0)

        fig.add_trace(go.Scatter(x=x_data, y=y_data, mode='lines+markers',
        name=labels[i],
        line=dict(color=colors[i], width=line_size[i]),
        connectgaps=True,
        text = "Total " + str(labels[i]) +": "+ str(y_data[-1])
        ));

    fig.update_layout(
        title="COVID 19 cases of " + country,
        xaxis_title='Date',
        yaxis_title='No. of Confirmed Cases',
        margin=dict(l=20, r=20, t=40, b=20),
        paper_bgcolor="lightgrey",
        width = 800,

    );

    fig.update_yaxes(type="linear")
    fig.show();
```

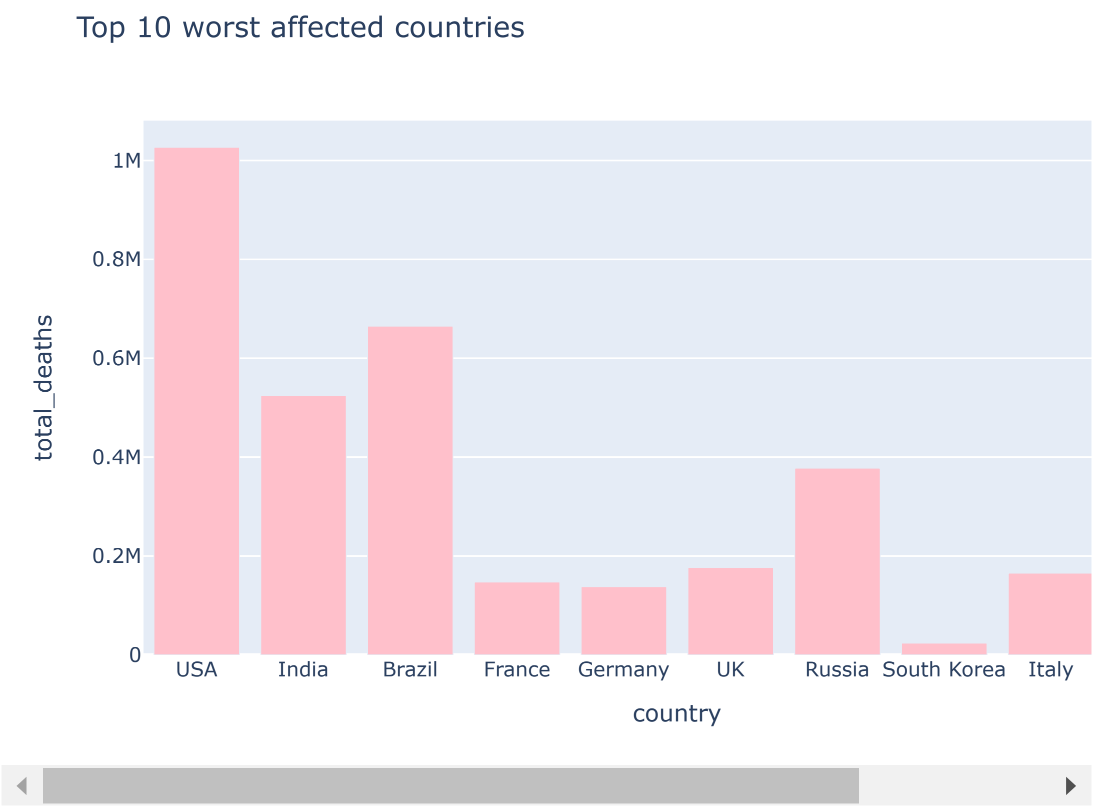
10 worst hit countries - Confirmed cases

```
px.bar(
    sorted_country_df.head(10),
    x = "country",
    y = "total_confirmed",
    title= "Top 10 worst affected countries", # the axis names
    color_discrete_sequence=["pink"],
    height=500,
    width=800
)
```



10 worst hit countries - Death cases

```
px.bar(
    sorted_country_df.head(10),
    x = "country",
    y = "total_deaths",
    title= "Top 10 worst affected countries", # the axis names
    color_discrete_sequence=["pink"],
    height=500,
    width=800
)
```



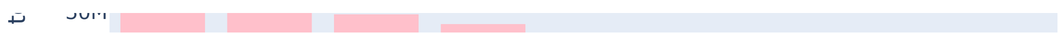
Worst hit countries - Recovering cases

```
px.bar(
    sorted_country_df.head(10),
    x = "country",
    y = "total_recovered",
    title= "Top 10 worst affected countries", # the axis names
    color_discrete_sequence=["pink"],
    height=500,
    width=800
)
```

Top 10 worst affected countries

```
confirmed_df.isnull().sum()
```

```
Province/State    198
Country/Region      0
Lat                 2
Long                2
1/22/20             0
                  ...
3/5/23              0
3/6/23              0
3/7/23              0
3/8/23              0
3/9/23              0
Length: 1147, dtype: int64
```



```
confirmed_df[confirmed_df.isnull().any(axis=1)]
```

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20
0	NaN	Afghanistan	33.939110	67.709953	0	0	0
1	NaN	Albania	41.153300	20.168300	0	0	0
2	NaN	Algeria	28.033900	1.659600	0	0	0
3	NaN	Andorra	42.506300	1.521800	0	0	0
4	NaN	Angola	-11.202700	17.873900	0	0	0
...	...	...	...	...	...	...	...
284	NaN	West Bank and Gaza	31.952200	35.233200	0	0	0
285	NaN	Winter Olympics 2022	39.904200	116.407400	0	0	0
286	NaN	Yemen	15.552727	48.516388	0	0	0
287	NaN	Zambia	-13.133897	27.849332	0	0	0
288	NaN	Zimbabwe	-19.015438	29.154857	0	0	0