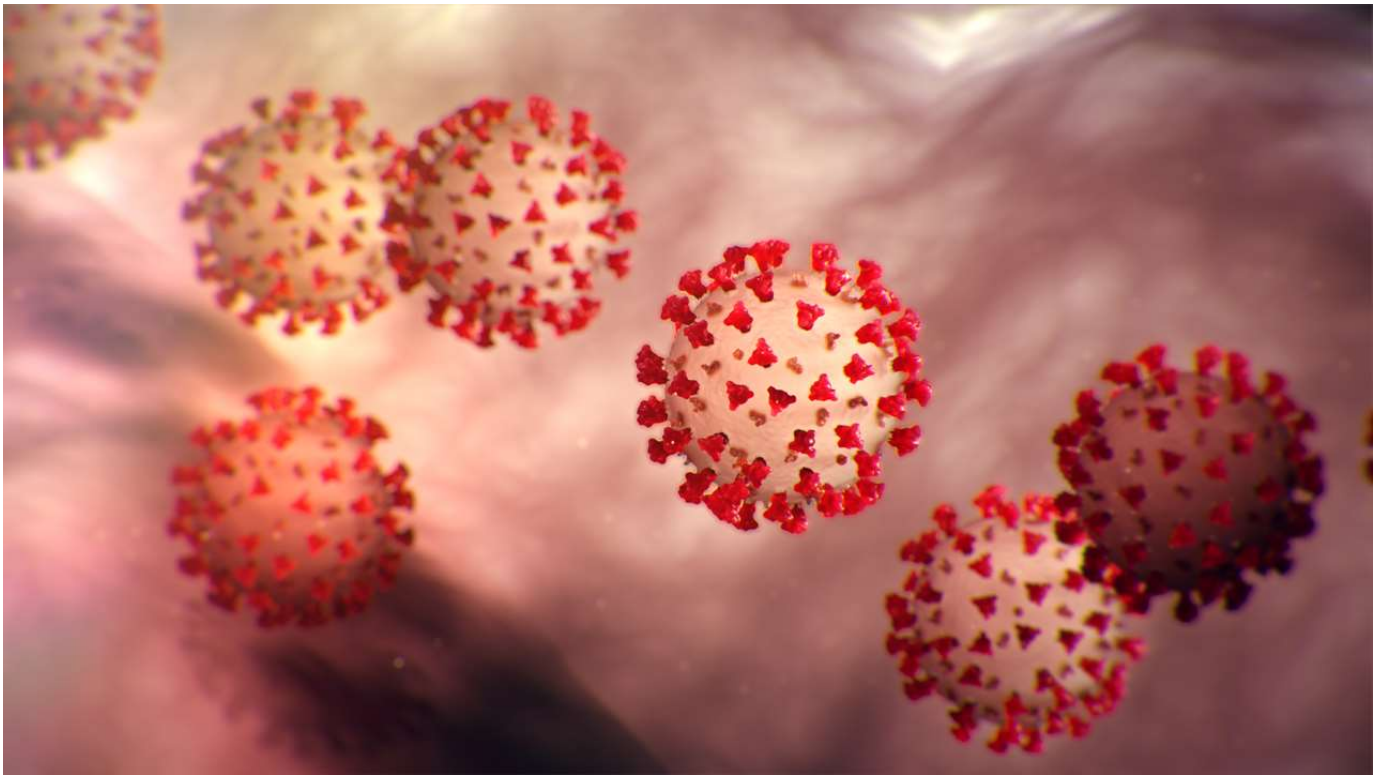
COVID-19 Interactive Analysis Dashboard

What is COVID-19?

Coronaviruses are a large family of viruses that may cause respiratory illnesses in humans ranging from common colds to more severe conditions such as Severe Acute Respiratory Syndrome (SARS) and Middle Eastern Respiratory Syndrome (MERS).'Novel coronavirus' is a new, previously unidentified strain of coronavirus. The novel coronavirus involved in the current outbreak has been named SARS-CoV-2 by the World Health Organization (WHO). The disease it causes has been named “coronavirus disease 2019” (or “COVID-19”).`



```
import pandas as pd
import numpy as np


# loading data right from the source:
death_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/c
confirmed_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/mast
recovered_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/mast
country_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/web-da


#confirmed_df.head()


#recovered_df.head()
```

```
#death_df.head()


#country_df.head()


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


#confirmed_df.head()


#recovered_df.head()


#death_df.head()


#country_df.head()


# total number of confirmed, death and recovered cases
confirmed_total = int(country_df['confirmed'].sum())
deaths_total = int(country_df['deaths'].sum())
recovered_total = int(country_df['recovered'].sum())
active_total = int(country_df['active'].sum())


from IPython.core.display import display, HTML


# displaying the total stats

display(HTML("<div style = 'background-color: #504e4e; padding: 30px '>" +
             "<span style='color: #fff; font-size:30px;'> Confirmed cases: "  + str(confir
             "<span style='color: red; font-size:30px;margin-left:20px;'> Deaths: " + str(
             "<span style='color: lightgreen; font-size:30px; margin-left:20px;'> Recovere
             "</div>")
       )
```

Confirmed cases: 264950926 Deaths: 5243482 Recovered cases: 0

Double-click (or enter) to edit

COVID-19 Confirmed/Death/Recovered cases by countries

Enter number of countries you want the data for:

```
import plotly.graph_objects as go


from ipywidgets import interact, interactive, fixed, interact_manual


import ipywidgets as widgets


# sorting the values by confirmed descednding order
# country_df.sort_values('confirmed', ascending= False).head(10).style.background_gradient
fig = go.FigureWidget( layout=go.Layout() )
def highlight_col(x):
    r = 'background-color: pink'
    y = 'background-color: lightyellow'
    g = 'background-color: lightgreen'
    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('confirmed', ascending= False).head(n).style.apply(highl

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 disappe
widgets.VBox([fig], layout=ipywLayout)
```

)

last_update	lat	long_	confirmed	deaths	recovered	active	incident_r
2021-12-04 06:22:14	40.000000	-100.000000	48990127	787695	nan	nan	14869.541
2021-12-04 06:22:14	20.593684	78.962880	34624360	470530	nan	nan	2509.003
2021-12-04 06:22:14	-14.235000	-51.925300	22129409	615400	nan	nan	10410.928
2021-12-04 06:22:14	55.000000	-3.000000	10438381	145874	nan	nan	15376.337
2021-12-04 06:22:14	61.524000	105.318800	9565909	273463	nan	nan	6554.935
2021-12-04							

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

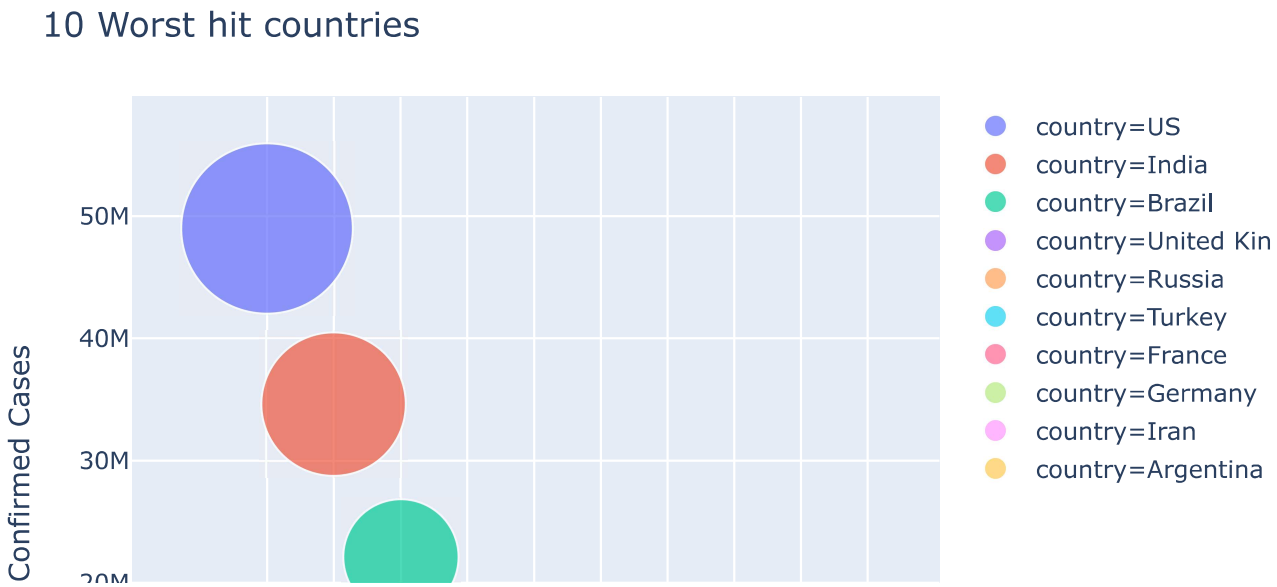
2021-12-04

```
import plotly.express as px
```

2021-12-04 51.165691 10.451526 6134492 102951 nan nan 7377.174

```
# # plotting the 20 worst hit countries

def bubble_chart(n):
    fig = px.scatter(sorted_country_df.head(n), x="country", y="confirmed", size="confirme
                hover_name="country", size_max=60)
    fig.update_layout(
    title=str(n) +" Worst hit countries",
    xaxis_title="Countries",
    yaxis_title="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



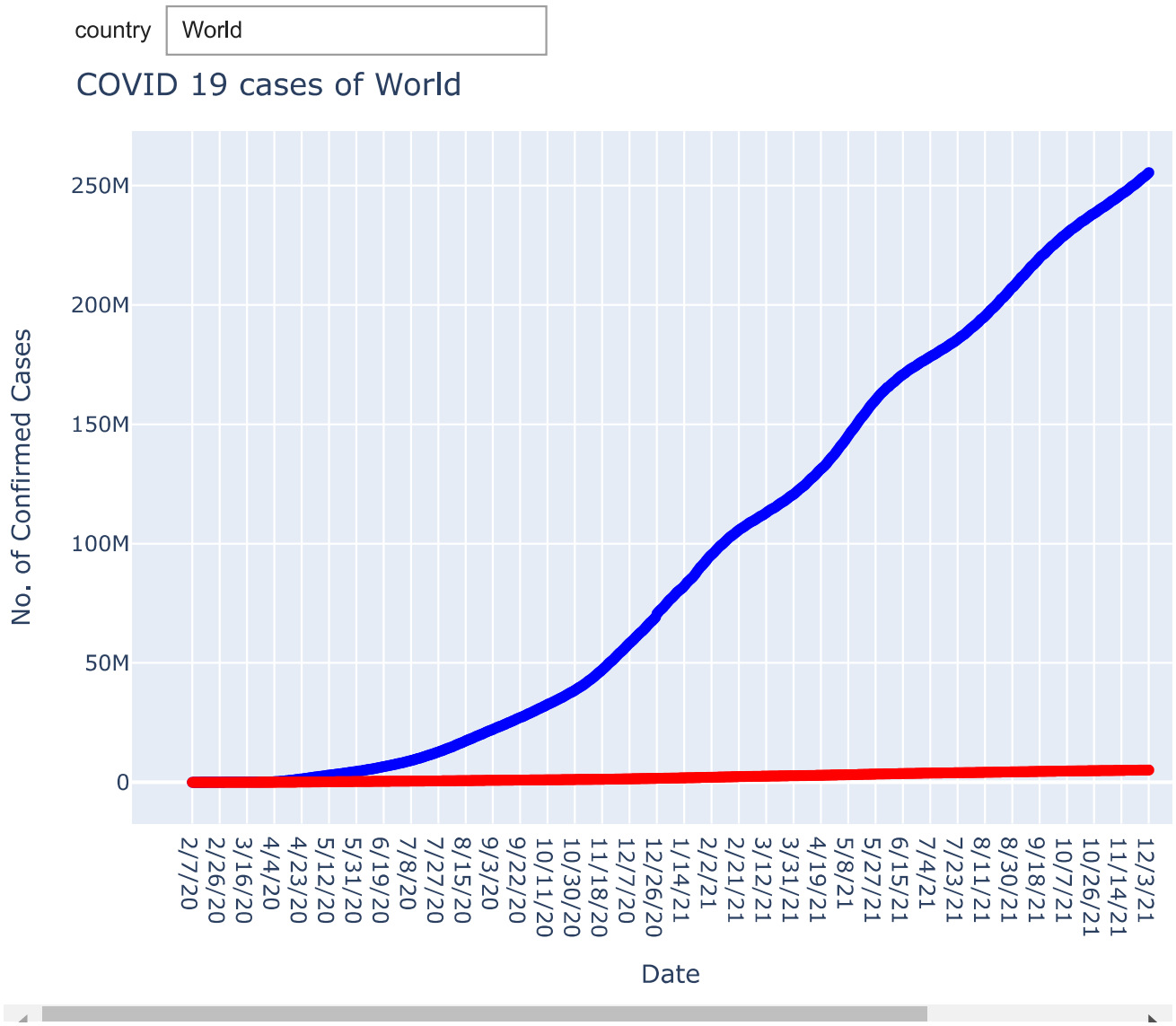

```
def plot_cases_of_a_country(country):
    labels = ['confirmed', 'deaths']
    colors = ['blue', 'red']
    mode_size = [6, 8]
    line_size = [4, 5]

    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();
```

Check the details of your country or the World

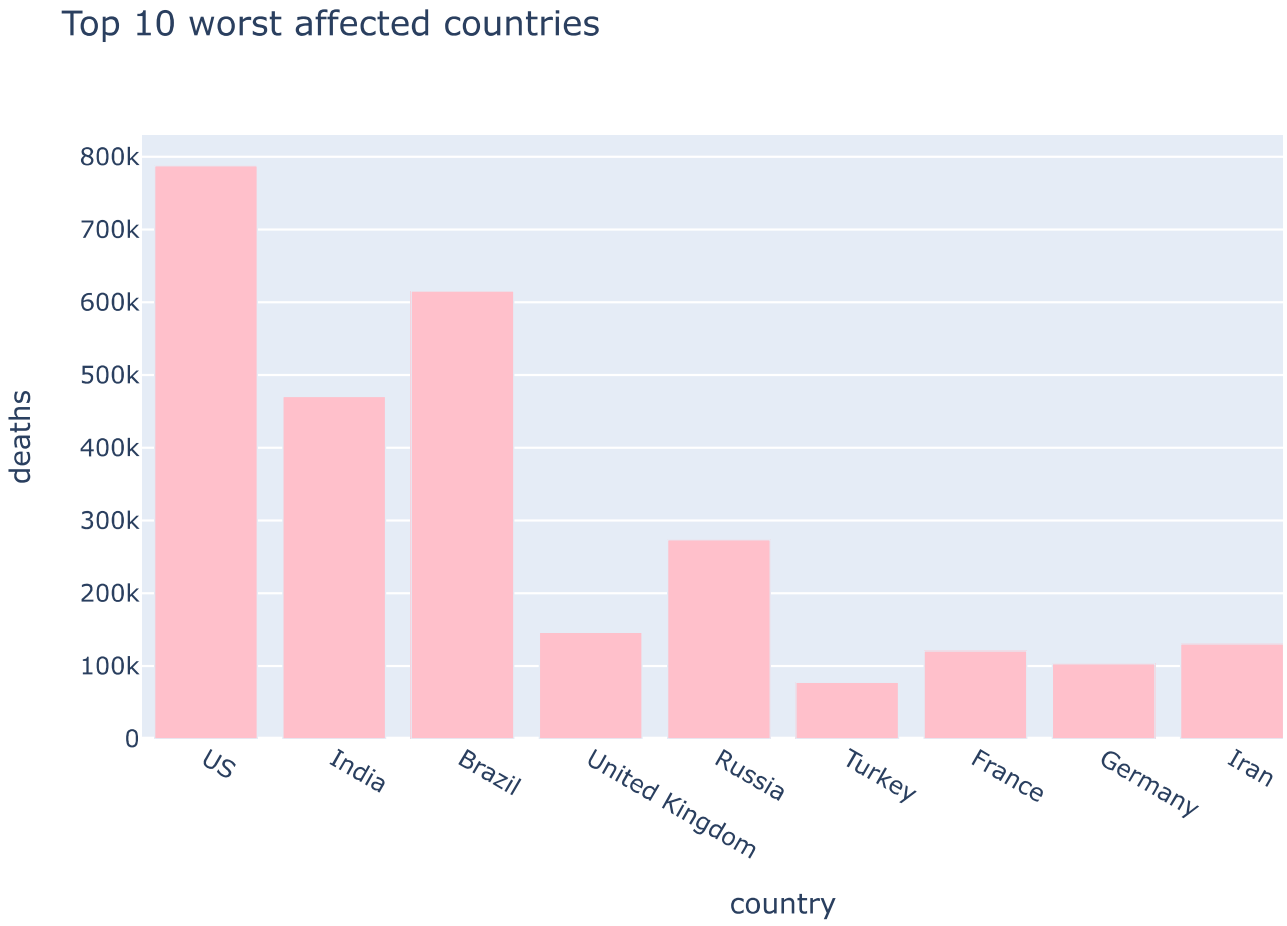
Enter the name of your country(in capitalized format(e.g. Italy)) and world for total cases

```
interact(plot_cases_of_a_country, country='World')

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

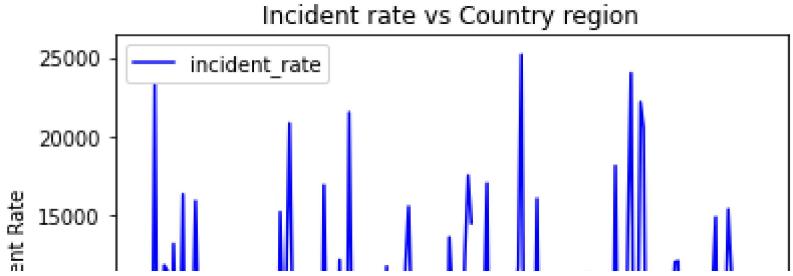


10 worst hit countries - Death cases

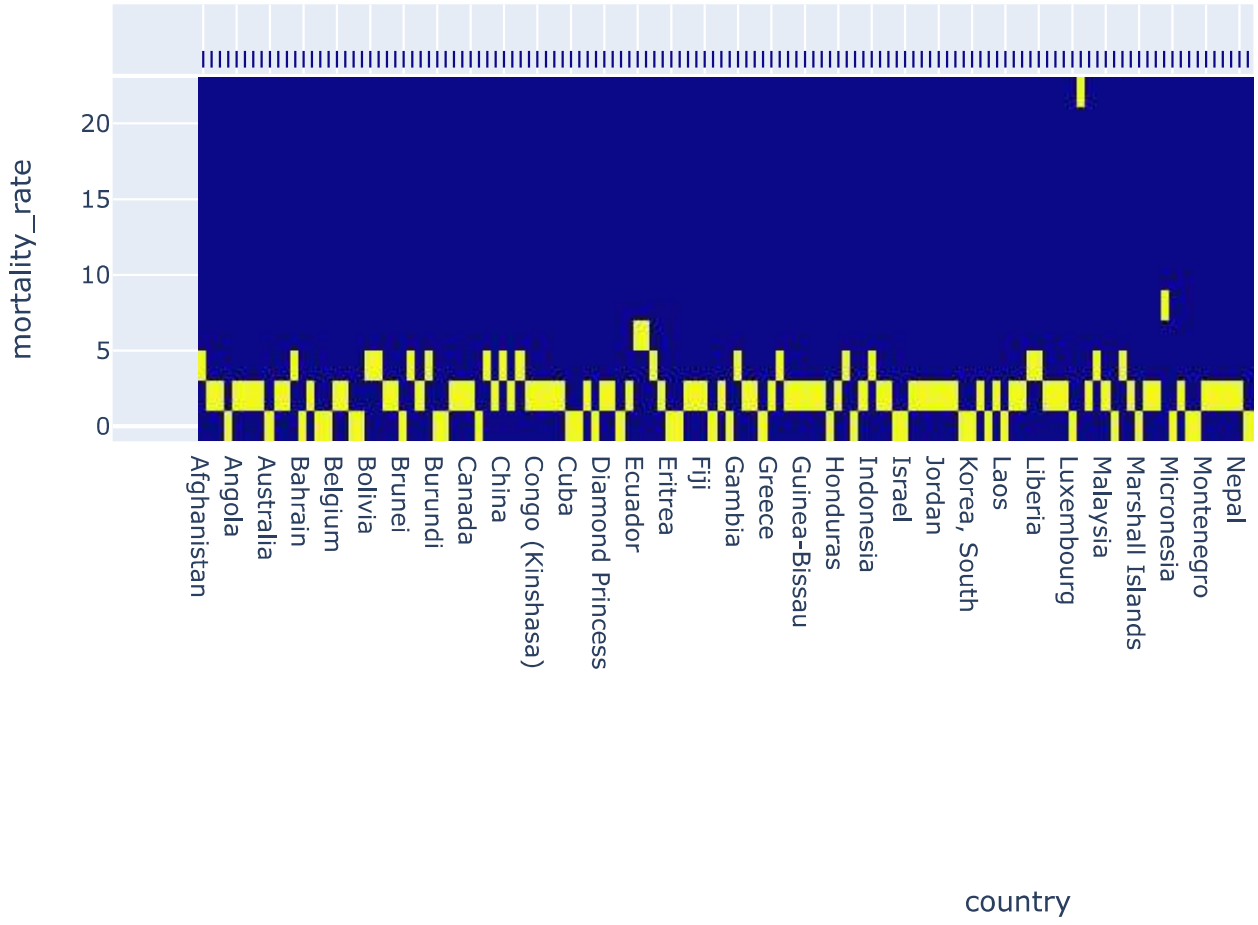
```
px.bar(
    sorted_country_df.head(10),
    x = "country",
    y = "deaths",
    title= "Top 10 worst affected countries", # the axis names
    color_discrete_sequence=["pink"],
    height=500,
    width=800
)
```



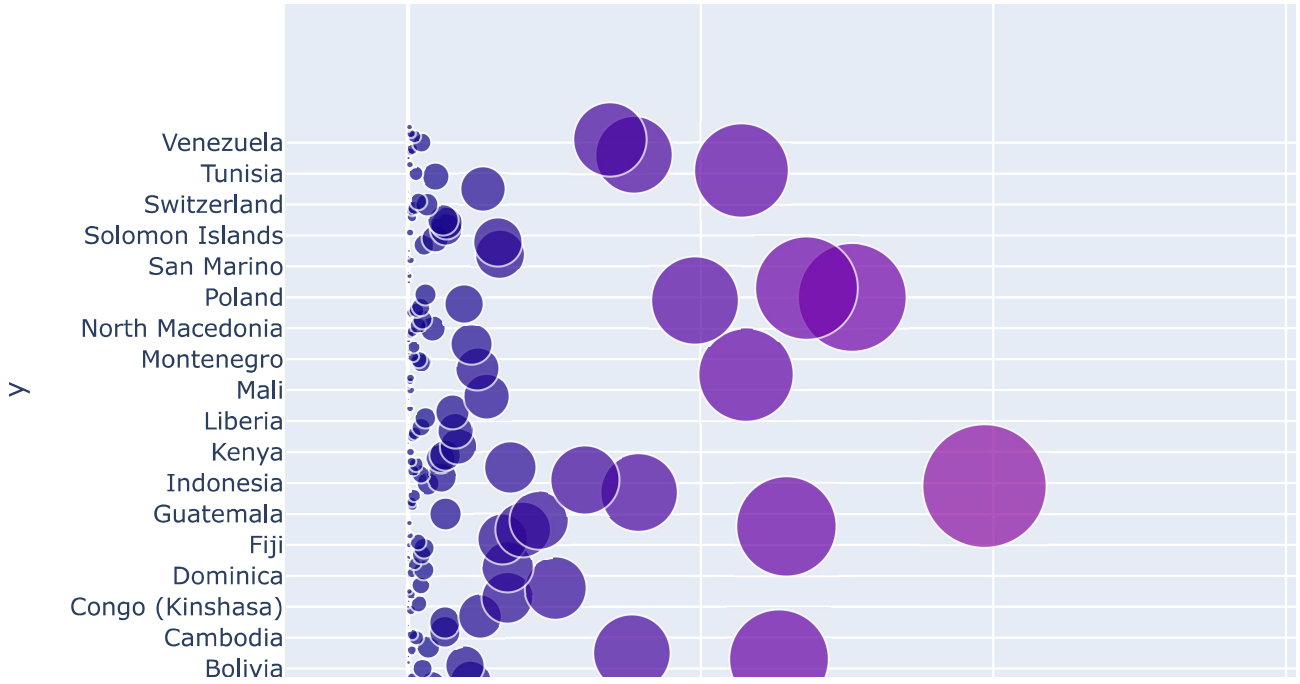

```
import matplotlib.pyplot as plt


country_df.plot(x='country', y='incident_rate',color='blue')
plt.xlabel("Country Region")
plt.ylabel("Incident Rate")
plt.title("Incident rate vs Country region")
plt.show()
```



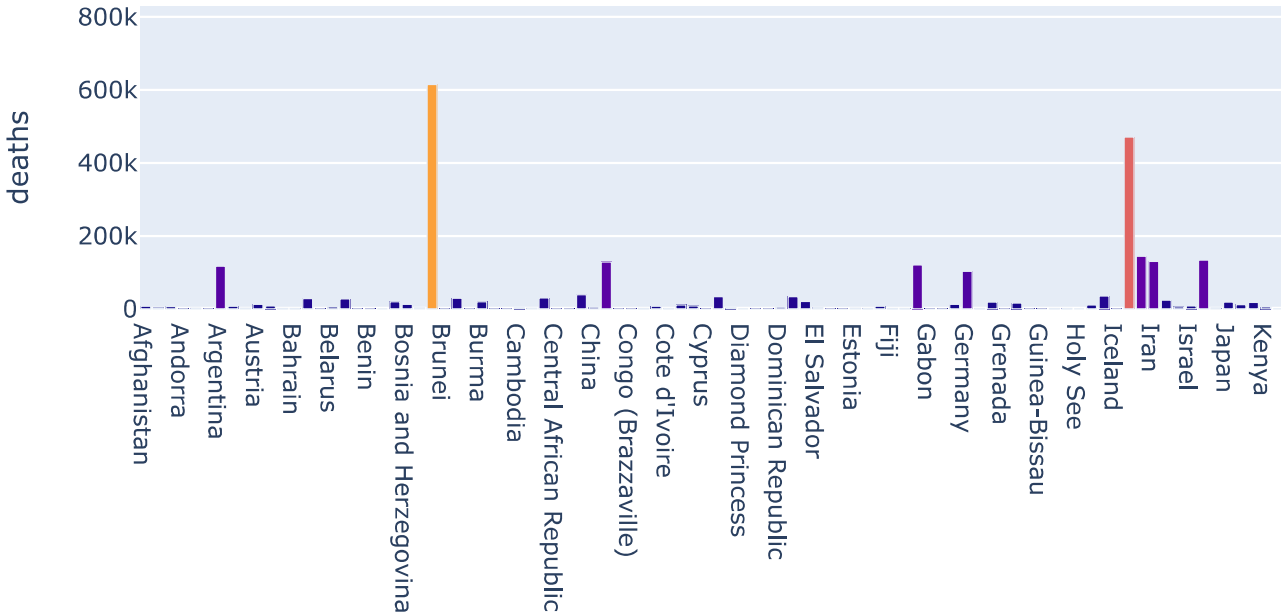

```
import plotly.express as px
fig = px.density_heatmap(country_df, x=country_df["country"], y=country_df["mortality_rate
fig.show()
```




```
px.scatter(country_df.head(50), x=country_df['deaths_28_days'],y=country_df['country'],
           hover_data=[country_df['country'],country_df['deaths_28_days'] ],
           color=country_df['deaths_28_days'] , size=country_df['deaths_28_days'] , size_m
#x axis: death rate 28 days
#y axis: Country name
```




```
px.bar(country_df, x=country_df["country"], y=country_df["deaths"], color=country_df["deat
```



Summary:

1. What is COVID-19?
2. Data loading from John Hopkins CSSE data repository
3. Data Cleaning and Preparation
4. Visualising N number of worst hit countries using plotly scatter plot

5. Plotting confirmed cases as a bubble chart
6. Plotting line chart
7. Plotting bar chart
8. Plotting line chart
9. Plotting Density chart
10. Plotting scatter plot
11. Plotting bar plot

Symptoms: People may be sick with the virus for 1 to 14 days before developing symptoms. The most common symptoms of coronavirus disease (COVID-19) are fever, tiredness, difficulty in breathing(severe cases) and dry cough. Most people (about 80%) recover from the disease without needing special treatment.

References:

More Info on COVID-19:

- https://www.who.int/health-topics/coronavirus
- https://www.who.int/emergencies/diseases/novel-coronavirus-2019
- https://www.nature.com/articles/s41597-020-0448-0

Refered youtube link:

https://www.youtube.com/watch?v=FngV4VdYrkA

✓ 3s completed at 2:59 PM