Project by : KRITIKA RAWAT.

project title : Analysis of COVID-19 data .

course : bechlor's in computer science.

Project Title: Analysis of Global COVID-19 Data

Objective:
The primary goal of this project was to analyze a comprehensive dataset of global COVID-19 cases to uncover trends and patterns in the spread of the virus. By visualizing the data, the project aimed to highlight the impact of the pandemic across different countries and provide actionable insights that could aid in public health planning and resource allocation.

Approach:

1. **Data Preparation:**
    - **Data Import:** The dataset, which contained information on COVID-19 cases from various countries, was imported into a Pandas DataFrame.
    - **Data Cleaning:** The dataset was cleaned to ensure accuracy and consistency. This included:
        - Handling missing values by either imputing them with appropriate values or removing incomplete records.
        - Converting data types where necessary, such as changing date columns to datetime objects for better manipulation.
        - Dropping irrelevant columns that did not contribute to the analysis, thus focusing only on the necessary data.
    - **Preprocessing:** Ensured that all data was in a format suitable for analysis, including standardizing country names and verifying data integrity.
2. **Exploratory Data Analysis (EDA):**
    - **Pattern Identification:** Conducted initial explorations to identify key patterns and trends in the data.
    - **Top Countries Analysis:** Focused on visualizing the distribution of confirmed COVID-19 cases. Created an interactive bar chart to highlight the top 10 countries with the

highest number of confirmed cases, making it easier to see which countries were most affected.

3. **Visualization:**
    - **Plotly Express Library:** Used Plotly Express to create interactive and dynamic visualizations. This library allows users to explore the data visually and interact with the charts for a more comprehensive understanding.
    - **Pie Chart:** Developed an interactive pie chart to show the distribution of critical COVID-19 cases by country, providing a clear view of which countries had the highest number of critical cases.

Key Insights:

- **High Impact Countries:** The analysis identified the USA, India, and France as having the highest numbers of confirmed COVID-19 cases. This finding underscored the severe impact of the pandemic in these regions, highlighting the need for targeted public health interventions and resource allocation.

Tools & Technologies:

- **Python:** The primary programming language used for data analysis and visualization.
- **Pandas:** Utilized for data manipulation, including cleaning, preprocessing, and aggregation.
- **Plotly Express:** Employed to create interactive and engaging visualizations, making it easier to understand complex data and trends.

Impact:
The project provides a detailed, data-driven perspective on the global impact of COVID-19. By visualizing the data effectively, it enables stakeholders, including public health officials and policymakers, to better comprehend the spread of the virus. The insights gained from this analysis can help prioritize resources, plan interventions, and improve strategies for managing public health crises.

ANALIZATION OF COVID-19 DATA .

In [7]:
```python
import pandas as pd

# Specify the full path to your CSV file
file_path = "C:/Users/KRITIKA RAWAT/Downloads/covid_dataset.csv"

# Read the CSV file into a DataFrame
df = pd.read_csv(file_path, encoding='unicode_escape')
```

In [8]:
```python
df.shape
```

Out[8]: (225, 9)

In [10]:
```python
df.head(225)
```

Out[10]:

	Unnamed: 0	country	code	confirmed	recovered	critical	deaths	lastChange	lastUpdate
0	0	Afghanistan	AF	234174	211080	0	7996	2024-06-04T00:18:51+00:00	2024-08-09T09:13:41+00:00
1	2	Albania	AL	334863	330233	0	3605	2024-06-04T00:27:05+00:00	2024-08-09T09:13:41+00:00
2	3	Algeria	DZ	272010	183061	0	6881	2024-06-04T00:19:02+00:00	2024-08-09T09:13:41+00:00
3	5	Andorra	AD	48015	47563	0	165	2024-02-11T22:05:54+00:00	2024-08-09T09:13:41+00:00
4	6	Angola	AO	107327	103419	0	1937	2024-06-04T00:20:45+00:00	2024-08-09T09:13:41+00:00
...	...	...	...	...	...	...	...	...	...
220	243	Wallis and Futuna	WF	3550	438	0	8	2024-02-11T22:03:31+00:00	2024-08-09T09:13:41+00:00
221	244	Western Sahara	EH	10	9	0	1	2024-02-11T22:11:46+00:00	2024-08-09T09:13:41+00:00
222	245	Yemen	YE	11945	9124	0	2159	2024-02-11T22:02:34+00:00	2024-08-09T09:13:41+00:00
223	246	Zambia	ZM	349304	341316	0	4069	2024-06-04T00:26:18+00:00	2024-08-09T09:13:41+00:00
224	247	Zimbabwe	ZW	266359	258888	12	5740	2024-06-04T00:28:51+00:00	2024-08-09T09:13:41+00:00

225 rows × 9 columns

In [11]:
```python
df.info()
```

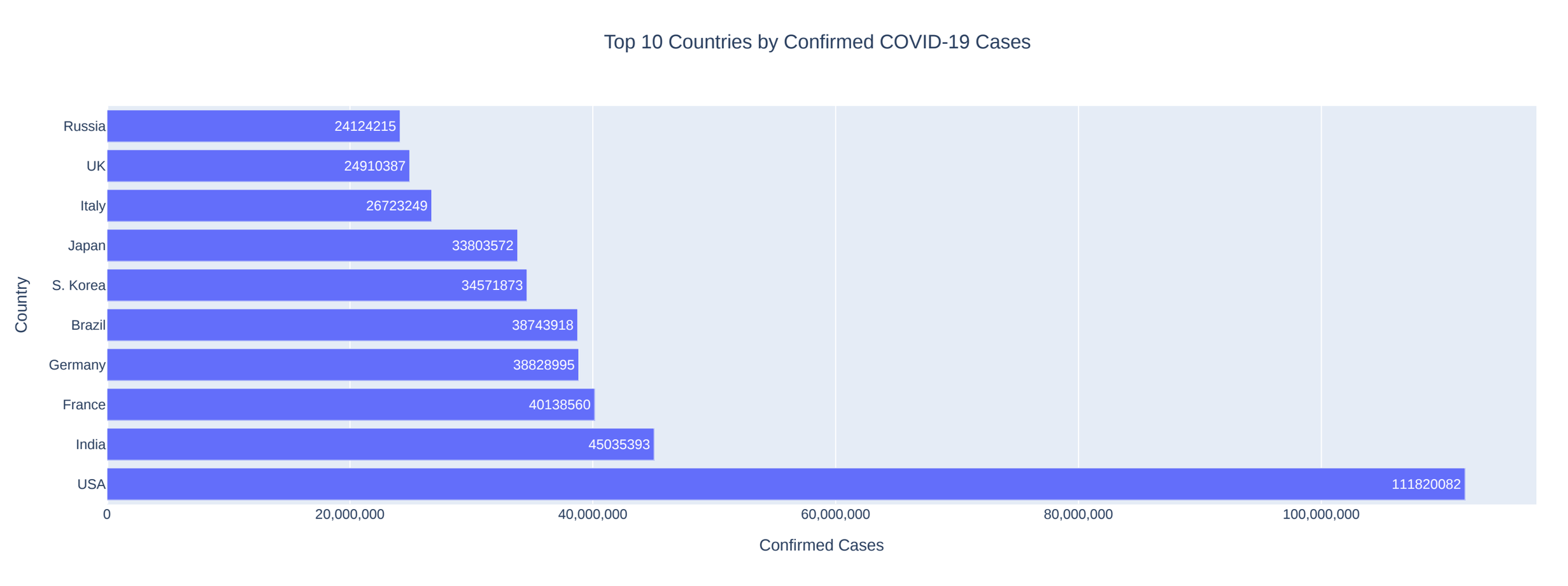
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 225 entries, 0 to 224
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   Unnamed: 0  225 non-null    int64 
 1   country     225 non-null    object
 2   code        224 non-null    object
 3   confirmed   225 non-null    int64 
 4   recovered   225 non-null    int64 
 5   critical    225 non-null    int64 
 6   deaths      225 non-null    int64 
 7   lastChange  225 non-null    object
 8   lastUpdate  225 non-null    object
dtypes: int64(5), object(4)
memory usage: 15.9+ KB
```

In [12]:
```python
df = df.drop(columns=['Unnamed: 0'])
```

In [13]:
```python
df['lastChange'] = pd.to_datetime(df['lastChange'])
df['lastUpdate'] = pd.to_datetime(df['lastUpdate'])
```

this graph shows Top 10 Countries by Confirmed COVID-19 Cases

In [45]:
```python
import plotly.express as px

# Create a DataFrame with the top N countries
top_countries = df.sort_values(by='confirmed', ascending=False).head(10)

# Create an interactive bar chart
fig = px.bar(top_countries, x='confirmed', y='country',
             title='Top 10 Countries by Confirmed COVID-19 Cases',
             labels={'confirmed': 'Confirmed Cases', 'country': 'Country'},
             text='confirmed')

# Update layout for better readability
fig.update_layout(xaxis_title='Confirmed Cases',
                  yaxis_title='Country',
                  xaxis=dict(tickformat=','),
                  title={'x': 0.5, 'xanchor': 'center'})

# Show the plot
fig.show()
```

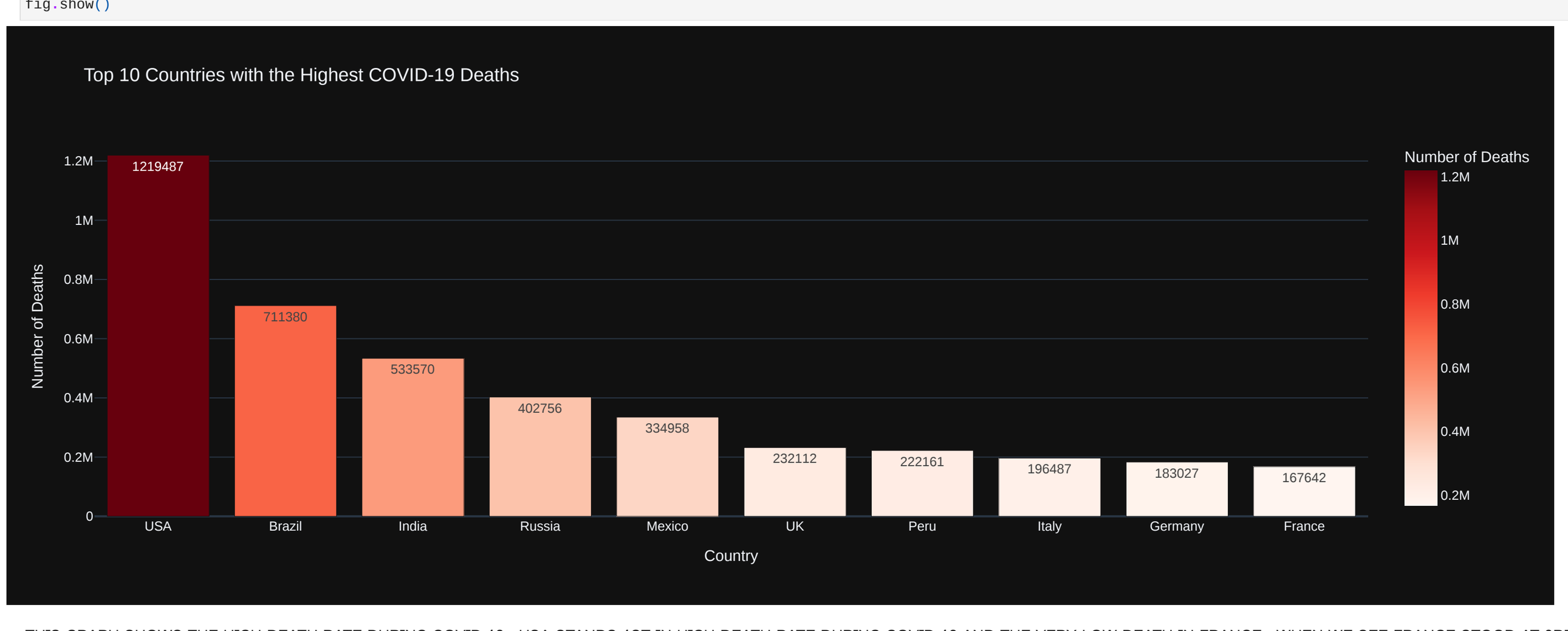


THE ABOVE SHOWS THE TOP 10 COUNTRIES BY HIGHLY CONFIRMED COVID-19 CASES . USA , INDIA , FRANCE ARE IN TOP 3 . MOST HIGH CINFIRMED CASES CAME FROM USA.

In [14]:
```python
# Display the first few rows of the DataFrame
print(df.head())

# Get summary statistics
print(df.describe())

# Check for any remaining missing values
print(df.isnull().sum())
```

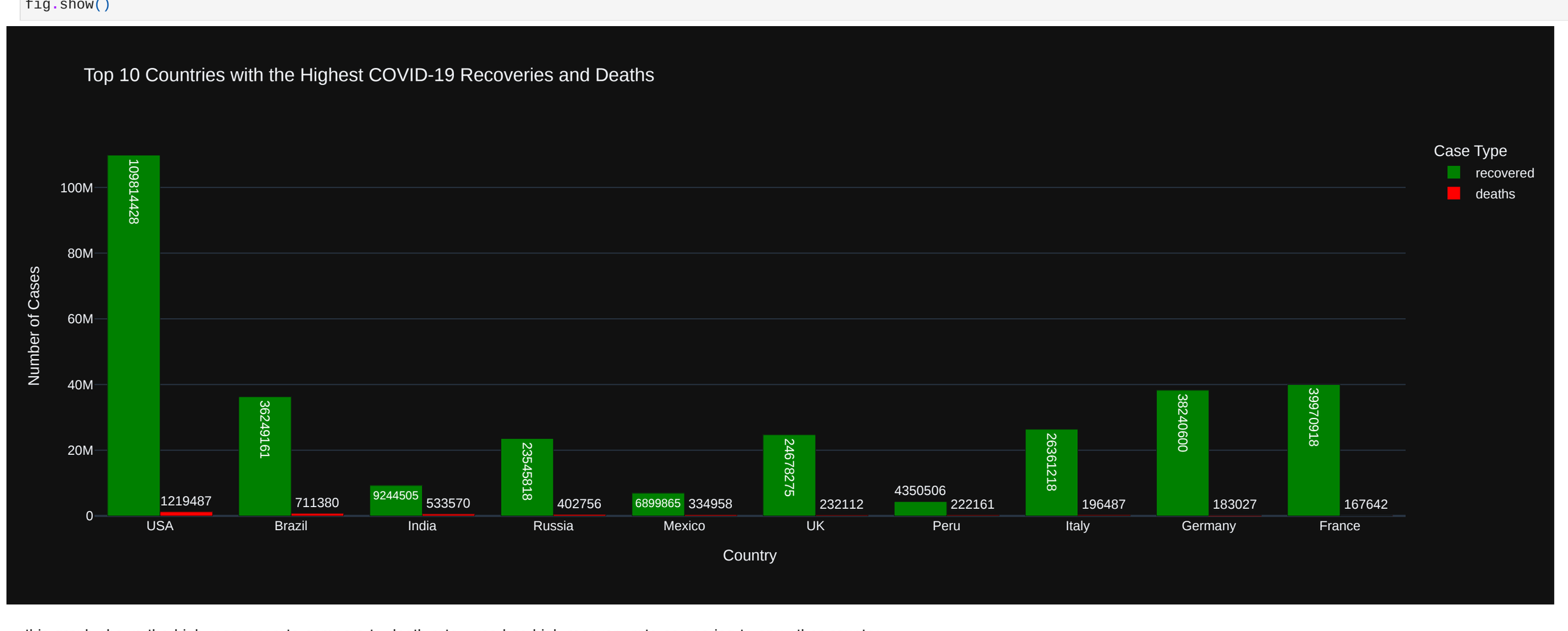
```
       country code  confirmed  recovered  critical  deaths  \
0  Afghanistan   AF     234174     211080         0    7996   
1      Albania   AL     334863     330233         0    3605   
2      Algeria   DZ     272010     183061         0    6881   
3      Andorra   AD      48015      47563         0     165   
4       Angola   AO     107327     103419         0    1937   

                 lastChange                lastUpdate  
0 2024-06-04 00:18:51+00:00 2024-08-09 09:13:41+00:00  
1 2024-06-04 00:27:05+00:00 2024-08-09 09:13:41+00:00  
2 2024-06-04 00:19:02+00:00 2024-08-09 09:13:41+00:00  
3 2024-02-11 22:05:54+00:00 2024-08-09 09:13:41+00:00  
4 2024-06-04 00:20:45+00:00 2024-08-09 09:13:41+00:00  
          confirmed     recovered    critical        deaths
count  2.250000e+02  2.250000e+02  225.000000  2.250000e+02
mean   3.116295e+06  2.624231e+06   12.786667  3.117617e+04
std    1.014304e+07  9.313309e+06   72.089718  1.105688e+05
min    1.000000e+01  0.000000e+00    0.000000  0.000000e+00
25%    2.733400e+04  1.542700e+04    0.000000  1.940000e+02
50%    2.303540e+05  1.406220e+05    0.000000  2.250000e+03
75%    1.356546e+06  1.258432e+06    0.000000  1.630300e+04
max    1.118201e+08  1.098144e+08  940.000000  1.219487e+06
country       0
code          1
confirmed     0
recovered     0
critical      0
deaths        0
lastChange    0
lastUpdate    0
dtype: int64
```

In [4]:
```python
import pandas as pd
import plotly.express as px

# Load the dataset
file_path = 'C:/Users/KRITIKA RAWAT/Downloads/covid_dataset.csv'
covid_data = pd.read_csv(file_path)

# Clean the dataset by removing the unnecessary 'Unnamed: 0' column
covid_data = covid_data.drop(columns=['Unnamed: 0'])

# Sort the data by deaths and select the top 10 countries
top_10_deaths = covid_data.sort_values(by='deaths', ascending=False).head(10)

# Create an interactive bar chart using Plotly
fig = px.bar(
    top_10_deaths,
    x='country',  # Countries on the x-axis
    y='deaths',   # Number of deaths on the y-axis
    text='deaths',  # Display the death count on the bars
    title='Top 10 Countries with the Highest COVID-19 Deaths',
    labels={'deaths': 'Number of Deaths', 'country': 'Country'},
    color='deaths',  # Color bars based on the number of deaths
    color_continuous_scale='Reds'
)

# Update the layout for better visualization
fig.update_layout(
    xaxis_title='Country',
    yaxis_title='Number of Deaths',
    template='plotly_dark'
)

# Show the interactive bar chart
fig.show()
```



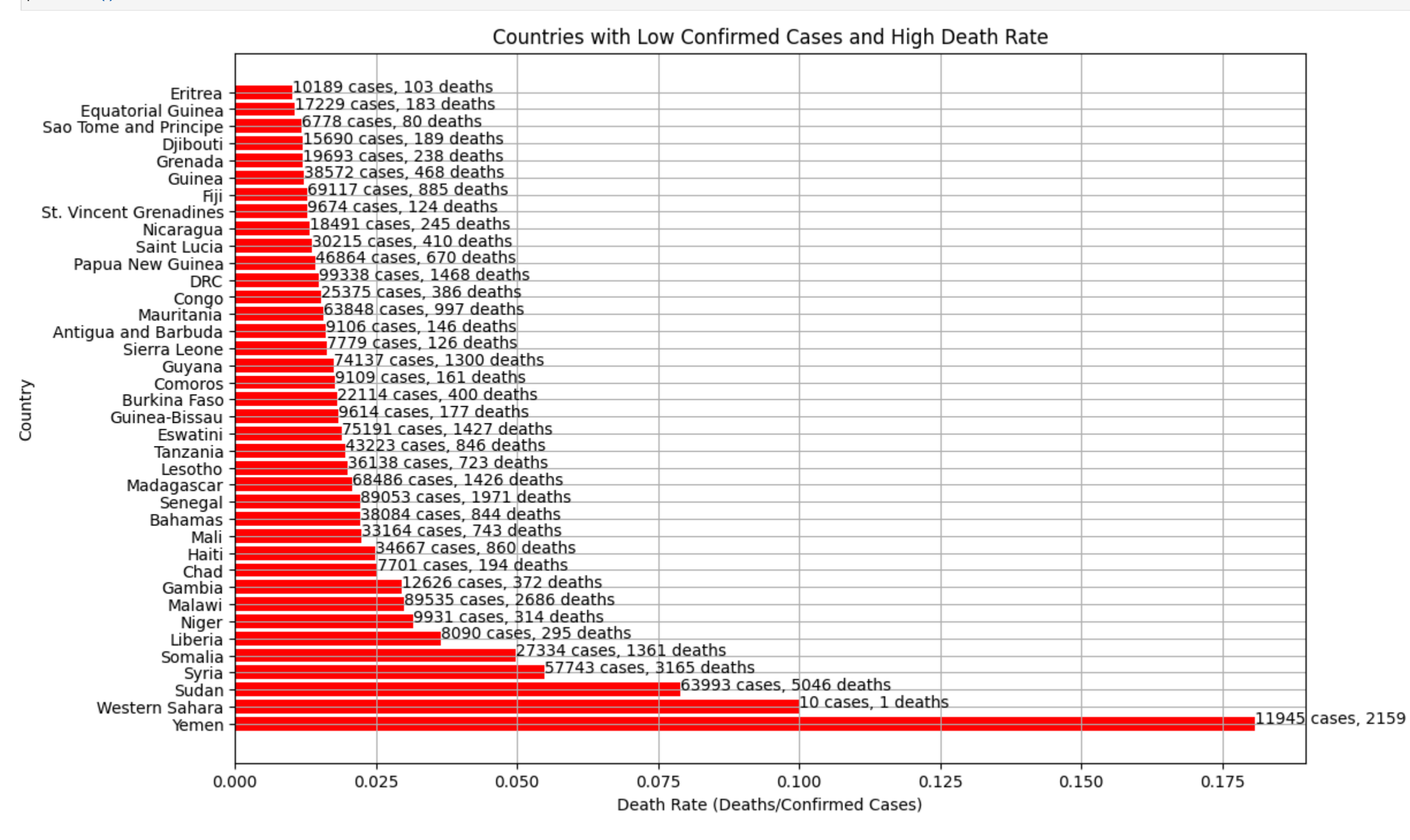
THIS GRAPH SHOWS THE HIGH DEATH RATE DURING COVID-19 . USA STANDS 1ST IN HIGH DEATH RATE DURING COVID-19 AND THE VERY LOW DEATH IN FRANCE . WHEN WE SEE FRANCE STOOD AT 3RD MOST HIGHLY CONFIRMED CASES BUT DUE TO THEY HAVE VERY GOOD MEDICAL TEAM THAT HLEPS TO SAVE LIFE IN THIER COUNTRY.

In [2]:
```python
import pandas as pd
import plotly.express as px

# Load the dataset
file_path = 'C:/Users/KRITIKA RAWAT/Downloads/covid_dataset.csv'
covid_data = pd.read_csv(file_path)

# Clean the dataset by removing the unnecessary 'Unnamed: 0' column
covid_data = covid_data.drop(columns=['Unnamed: 0'])

# Sort the data by deaths and select the top 10 countries
top_10_deaths = covid_data.sort_values(by='deaths', ascending=False).head(10)

# Melt the dataset to have a 'variable' column for recovered and deaths
melted_data = top_10_deaths.melt(id_vars=['country'], value_vars=['recovered', 'deaths'],
                                 var_name='Case Type', value_name='Count')

# Create an interactive grouped bar chart using Plotly
fig = px.bar(
    melted_data,
    x='country',  # Countries on the x-axis
    y='Count',    # Number of cases on the y-axis
    color='Case Type',  # Differentiate between recovered and deaths
    barmode='group',  # Group bars side-by-side
    text='Count',  # Display the count on the bars
    title='Top 10 Countries with the Highest COVID-19 Recoveries and Deaths',
    labels={'Count': 'Number of Cases', 'country': 'Country', 'Case Type': 'Case Type'},
    color_discrete_map={'recovered': 'green', 'deaths': 'red'}  # Set specific colors for recovered and deaths
)

# Update the layout for better visualization
fig.update_layout(
    xaxis_title='Country',
    yaxis_title='Number of Cases',
    template='plotly_dark'
)

# Show the interactive bar chart
fig.show()
```



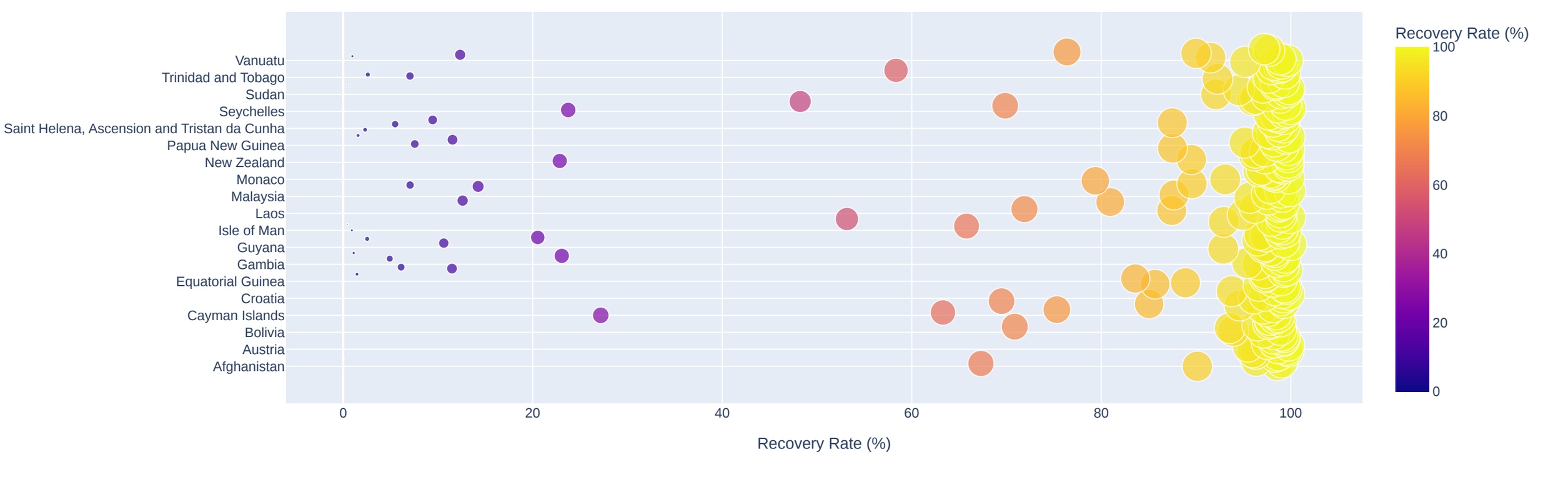
this graph shows the high recovery rate campare to death rate . usa has high recovery rate comparing to any other country .

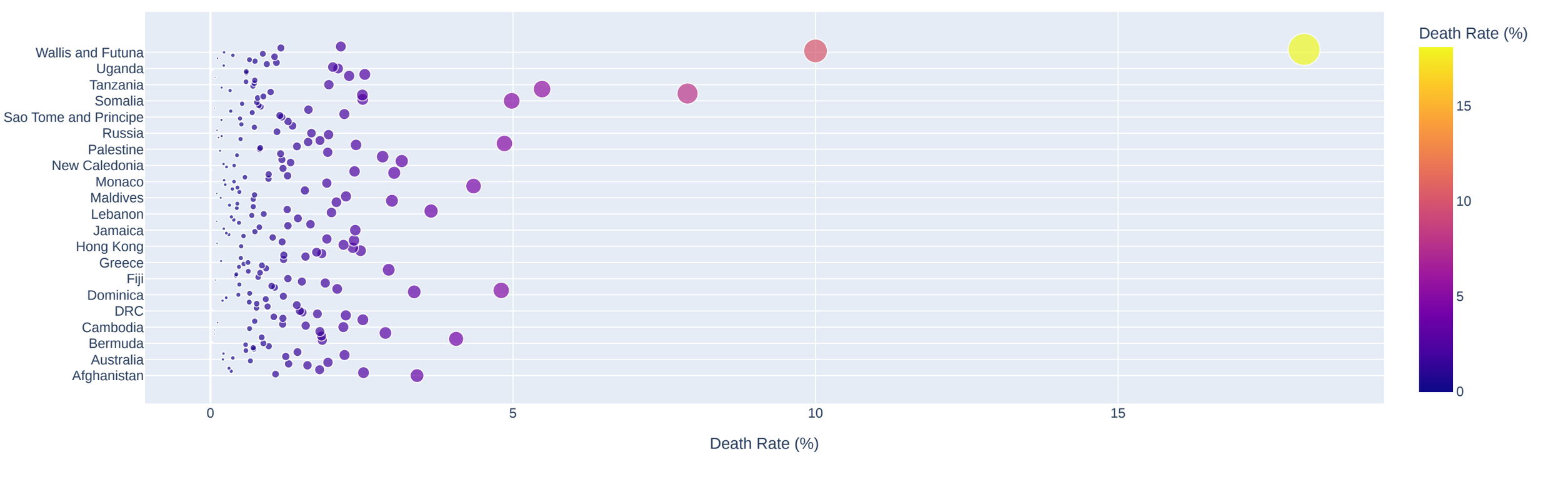
In [10]:
```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the CSV file
file_path = 'C:/Users/KRITIKA RAWAT/Downloads/covid_dataset.csv'  # Replace with the actual path to your CSV file
covid_data = pd.read_csv(file_path)

# Calculate the death rate per confirmed case
covid_data['death_rate'] = covid_data['deaths'] / covid_data['confirmed']

# Filter the dataset for countries with low cases and high deaths
# Adjust the thresholds as needed (example: less than 100,000 cases and death rate higher than 0.01)
low_cases_high_deaths = covid_data[(covid_data['confirmed'] < 100000) & (covid_data['death_rate'] > 0.01)]

# Sort the data by death rate in descending order for better visualization
sorted_data = low_cases_high_deaths.sort_values(by='death_rate', ascending=False)

# Create a bar graph
plt.figure(figsize=(12, 8))
plt.barh(sorted_data['country'], sorted_data['death_rate'], color='red')
plt.xlabel('Death Rate (Deaths/Confirmed Cases)')
plt.ylabel('Country')
plt.title('Countries with Low Confirmed Cases and High Death Rate')
plt.grid(True)

# Annotate the bars with actual numbers of confirmed cases and deaths
for index, value in enumerate(sorted_data['death_rate']):
    plt.text(value, index, f'{sorted_data["confirmed"].iloc[index]} cases, {sorted_data["deaths"].iloc[index]} deaths')

plt.show()
```

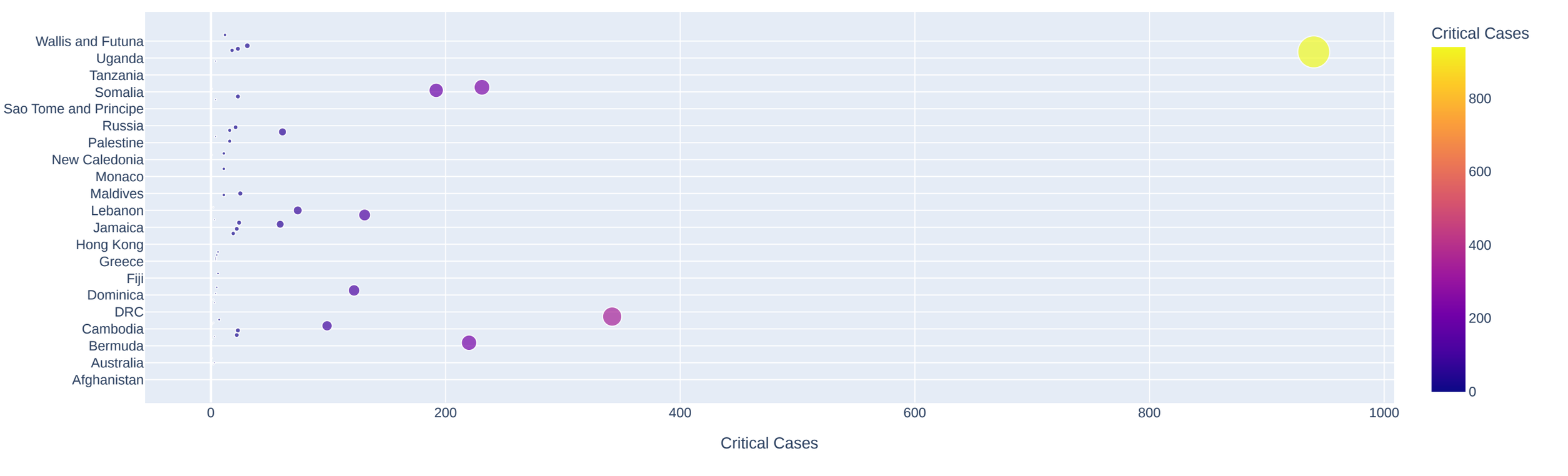


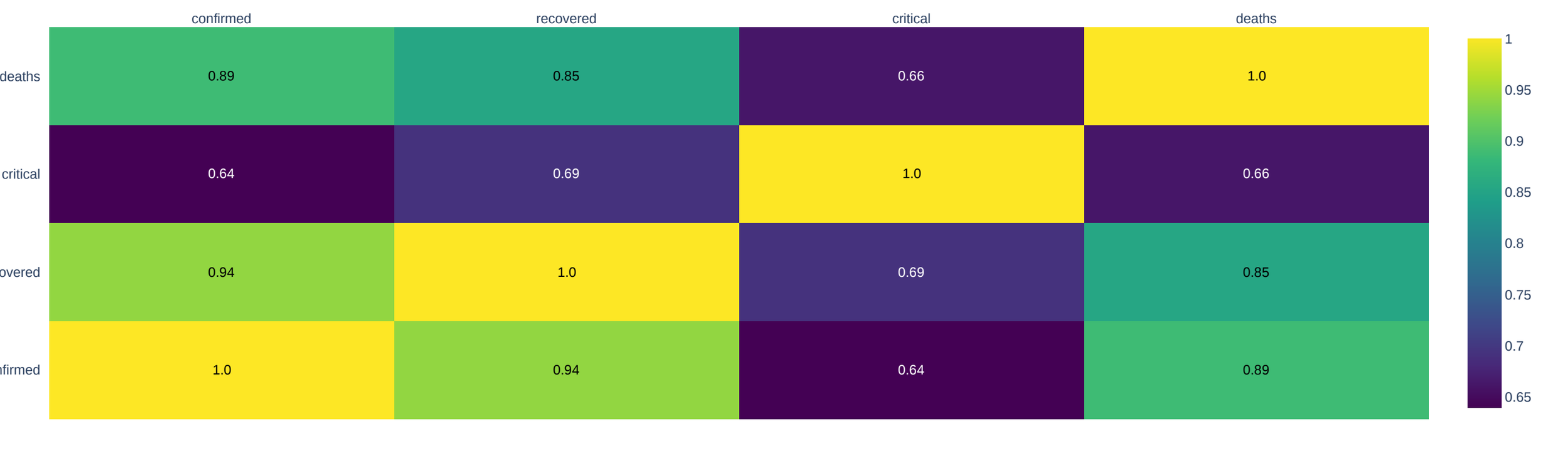
this graph shows the country with cases are low but death is high . yemen is 11945 cases but 2159 , in yemen death rate is high.

In [17]:
```python
import pandas as pd
import plotly.express as px
import plotly.figure_factory as ff

# Load your dataset
covid_data = pd.read_csv('C:/Users/KRITIKA RAWAT/Downloads/covid_dataset.csv')

# Calculate recovery and death rates
covid_data['recovery_rate'] = (covid_data['recovered'] / covid_data['confirmed']) * 100
covid_data['death_rate'] = (covid_data['deaths'] / covid_data['confirmed']) * 100

# 1. Scatter plot for Confirmed Cases with Country Names
fig1 = px.scatter(covid_data, x='confirmed', y='country', size='confirmed', color='confirmed',
                  hover_name='country', title='Distribution of Confirmed COVID-19 Cases by Country',
                  labels={'confirmed': 'Confirmed Cases'})
fig1.show()

# 2. Scatter plot for Recovery Rate with Country Names
fig2 = px.scatter(covid_data, x='recovery_rate', y='country', size='recovery_rate', color='recovery_rate',
                  hover_name='country', title='Recovery Rate by Country',
                  labels={'recovery_rate': 'Recovery Rate (%)'})
fig2.show()

# 3. Scatter plot for Death Rate with Country Names
fig3 = px.scatter(covid_data, x='death_rate', y='country', size='death_rate', color='death_rate',
                  hover_name='country', title='Death Rate by Country',
                  labels={'death_rate': 'Death Rate (%)'})
fig3.show()

# 4. Scatter plot for Critical Cases with Country Names
fig4 = px.scatter(covid_data, x='critical', y='country', size='critical', color='critical',
                  hover_name='country', title='Distribution of Critical COVID-19 Cases by Country',
                  labels={'critical': 'Critical Cases'})
fig4.show()

# 5. Correlation Matrix
corr_matrix = covid_data[['confirmed', 'recovered', 'critical', 'deaths']].corr()
fig5 = ff.create_annotated_heatmap(z=corr_matrix.values,
                                   x=list(corr_matrix.columns),
                                   y=list(corr_matrix.index),
                                   annotation_text=corr_matrix.round(2).values,
                                   colorscale='Viridis', showscale=True)
fig5.update_layout(title='Correlation Matrix of COVID-19 Statistics')
fig5.show()
```









all the five graphs describe below things

1.Scatter plot for Confirmed Cases with Country Names: This graph shows the distribution of confirmed COVID-19 cases by country. Each bubble represents a country, and the size of the bubble corresponds to the number of confirmed cases. The x-axis shows the number of confirmed cases, and the y-axis displays the country names.

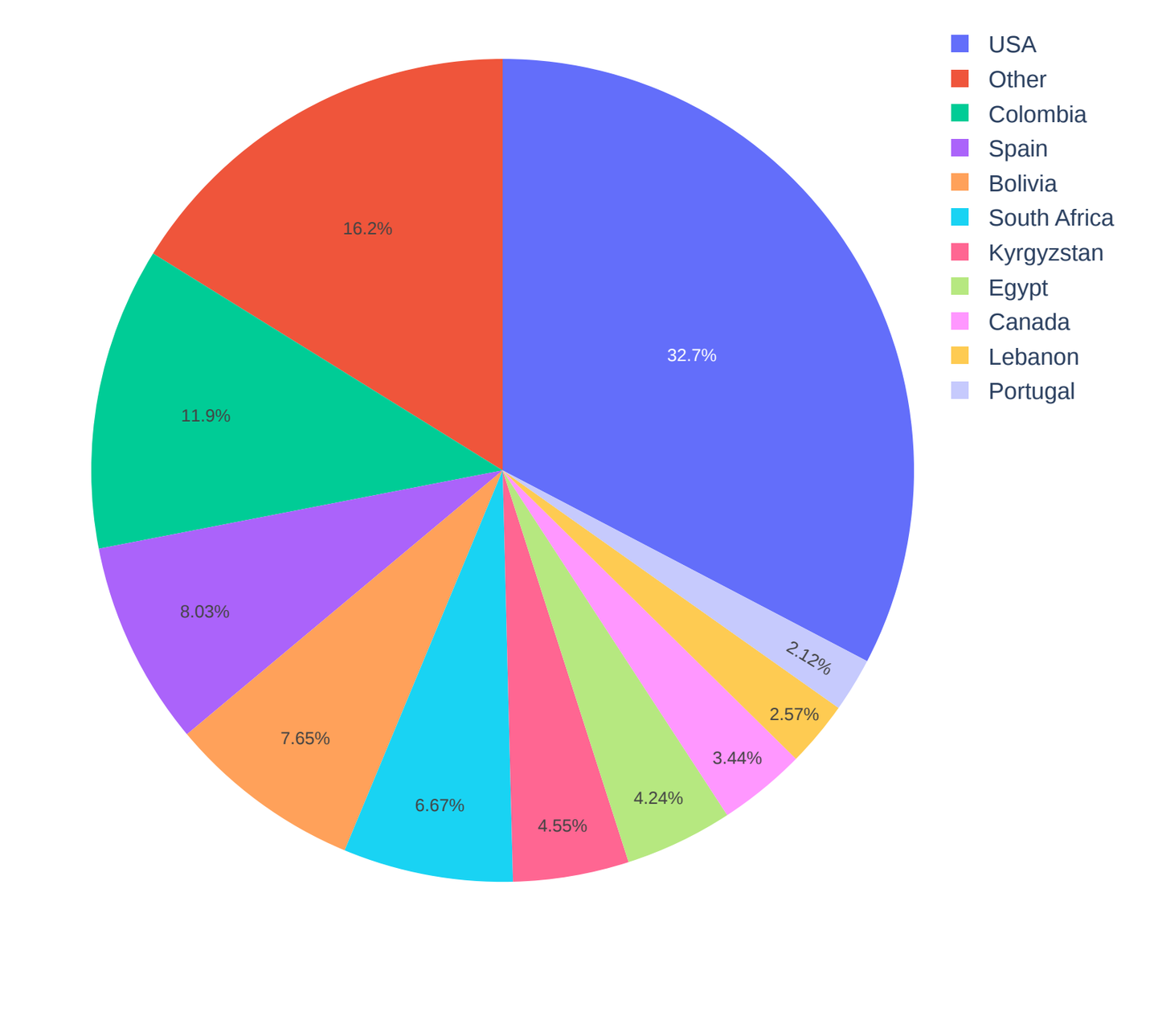
2.Scatter plot for Recovery Rate with Country Names: This graph shows the recovery rate by country. Each bubble represents a country, and the size of the bubble corresponds to the recovery rate. The x-axis displays the recovery rate (in percentage), and the y-axis shows the country names.

3.Scatter plot for Death Rate with Country Names: This graph shows the death rate by country. Each bubble represents a country, and the size of the bubble corresponds to the death rate. The x-axis displays the death rate (in percentage), and the y-axis shows the country names.

4.Scatter plot for Critical Cases with Country Names: This graph shows the distribution of critical COVID-19 cases by country. Each bubble represents a country, and the size of the bubble corresponds to the number of critical cases. The x-axis shows the number of critical cases, and the y-axis displays the country names.

5.Correlation Matrix: This graph displays the correlation matrix of confirmed, recovered, critical, and death cases. The matrix shows the correlation coefficients between these variables. The annotated heatmap helps visualize the strength and direction of the correlations.

In [31]:
```python
import pandas as pd
import plotly.express as px

# Load the CSV file
file_path = 'C:/Users/KRITIKA RAWAT/Downloads/covid_dataset.csv'  # Replace with your file path
covid_data = pd.read_csv(file_path)

# Replace 'country' and 'critical' with the actual column names
country_data = covid_data[['country', 'critical']]

# Group by country and sum the critical cases
country_critical_cases = country_data.groupby('country').sum().reset_index()

# Sort by critical cases and select the top 10 countries
top_10_countries = country_critical_cases.sort_values(by='critical', ascending=False).head(10)

# Calculate the sum of critical cases for other countries
other_cases = country_critical_cases['critical'].sum() - top_10_countries['critical'].sum()

# Add an "Other" category for the remaining countries
other_row = pd.DataFrame({'country': ['Other'], 'critical': [other_cases]})
top_10_countries = pd.concat([top_10_countries, other_row], ignore_index=True)

# Create the pie chart using Plotly
fig = px.pie(top_10_countries,
             names='country',
             values='critical',
             title='TOP 10 HIGHLY RECORDED CRITICAL CASES',
             labels={'country': 'Country', 'critical': 'Critical Cases'})

# Update layout to make the graph bigger and less messy
fig.update_layout(width=800, height=800, title_font_size=24, legend_title_font_size=15, legend_font_size=16)

# Show the pie chart
fig.show()
```



this pie chart shows top 10 countries recorded critical cases