Covid-19

August 3, 2025

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

# Ignore all warnings
warnings.filterwarnings('ignore')
```

[2]:
```
df = pd.read_csv("country_wise_latest.csv")
df.head()
```

[2]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	Confirmed last week	1 week change	1 week % increase	WHO Region
0	Afghanistan	36263	1269	25198	9796	106	10	18	3.50	69.49	5.04	35526	737	2.07	Eastern Mediterranean
1	Albania	4880	144	2745	1991	117	6	63	2.95	56.25	5.25	4171	709	17.00	Europe
2	Algeria	27973	1163	18837	7973	616	8	749	4.16	67.34	6.17	23691	4282	18.07	Africa
3	Andorra	907	52	803	52	10	0	0	5.73	88.53	6.48	884	23	2.60	Europe
4	Angola	950	41	242	667	18	1	0	4.32	25.47	16.94	749	201	26.84	Africa

[3]:
```
# Step 2: Data summary & missing value check
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 187 entries, 0 to 186 Data
columns (total 15 columns):
```

```
#   Column               Non-Null Count Dtype
--- ------               -------------- -----
0   Country/Region       187 non-null  object
1   Confirmed            187 non-null  int64
2   Deaths               187 non-null  int64
3   Recovered            187 non-null  int64
4   Active               187 non-null  int64
5   New cases            187 non-null  int64
6   New deaths           187 non-null  int64
7   New recovered        187 non-null  int64
8   Deaths / 100 Cases   187 non-null  float64
9   Recovered / 100 Cases187 non-null  float64
 10 Deaths / 100 Recovered 187 non-   float64
 null
11 Confirmed last week   187 non-null  int64
12 1 week change         187 non-null  int64
13 1 week % increase     187 non-null  float64
14 WHO Region            187 non-null  object
dtypes: float64(4), int64(9),
object(2)
memory usage: 22.0+
KB None
```

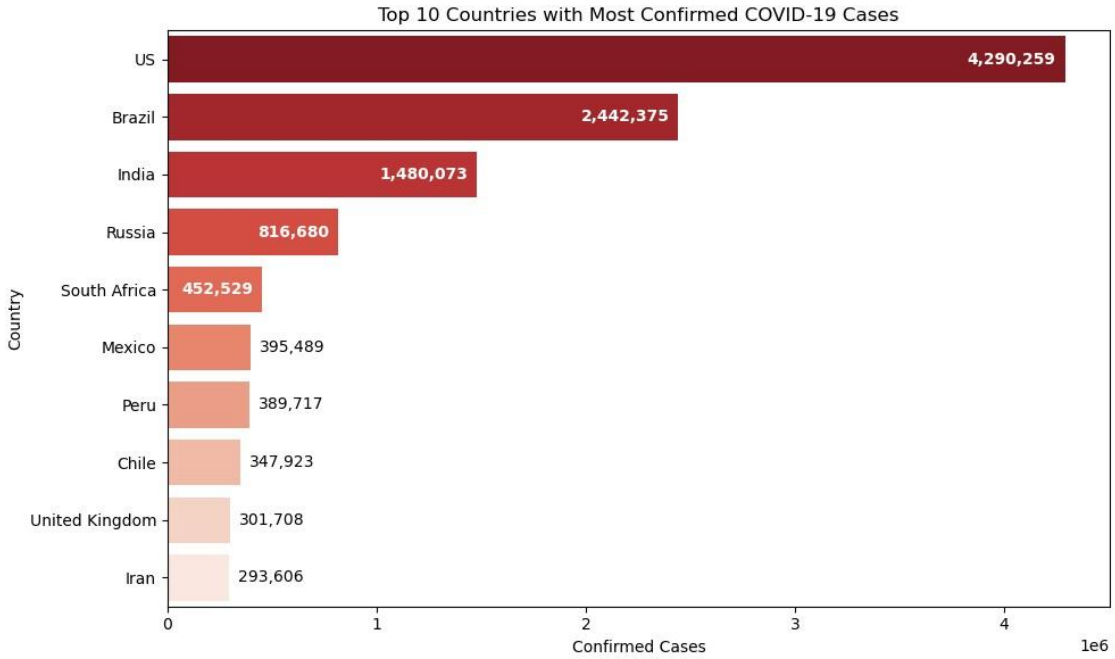
[4]:
```
print(df.isnull().sum())
```

```
Country/Region         0
Confirmed              0
Deaths                 0
Recovered              0
Active                 0
New cases              0
New deaths             0
New recovered          0
Deaths / 100 Cases     0
Recovered / 100 Cases  0
Deaths / 100 Recovered 0
Confirmed last week    0
1 week change          0
1 week % increase      0
WHO Region             0
dtype: int64
```

[6]:
```
# Step 3: Visualize top 10 countries by confirmed COVID-19 cases
import matplotlib.pyplot as plt
import seaborn as sns
```

[7]:
```
# Prepare data
top_confirmed = df.sort_values(by='Confirmed', ascending=False).head(10)
```

[11]:
```
# Plot
plt.figure(figsize=(10,6))
sns.barplot(x='Confirmed', y='Country/Region', data=top_confirmed,␣
 ↪palette='Reds_r')
plt.title("Top 10 Countries with Most Confirmed COVID-19 Cases")
plt.xlabel("Confirmed Cases")
plt.ylabel("Country")
plt.tight_layout()
# Add numbers to bars:
for index, value in enumerate(top_confirmed['Confirmed']):
    if index < 5:
        # Top 5: label inside the bar
        plt.text(value - (max(top_confirmed['Confirmed']) * 0.01), index,␣
 ↪f'{value:,}',
                 va='center', ha='right', color='white', fontsize=10,␣
 ↪fontweight='bold')
    else:
        # Bottom 5: label outside the bar
        plt.text(value + (max(top_confirmed['Confirmed']) * 0.01), index,␣
 ↪f'{value:,}',
                 va='center', ha='left', color='black', fontsize=10)

plt.show()
```



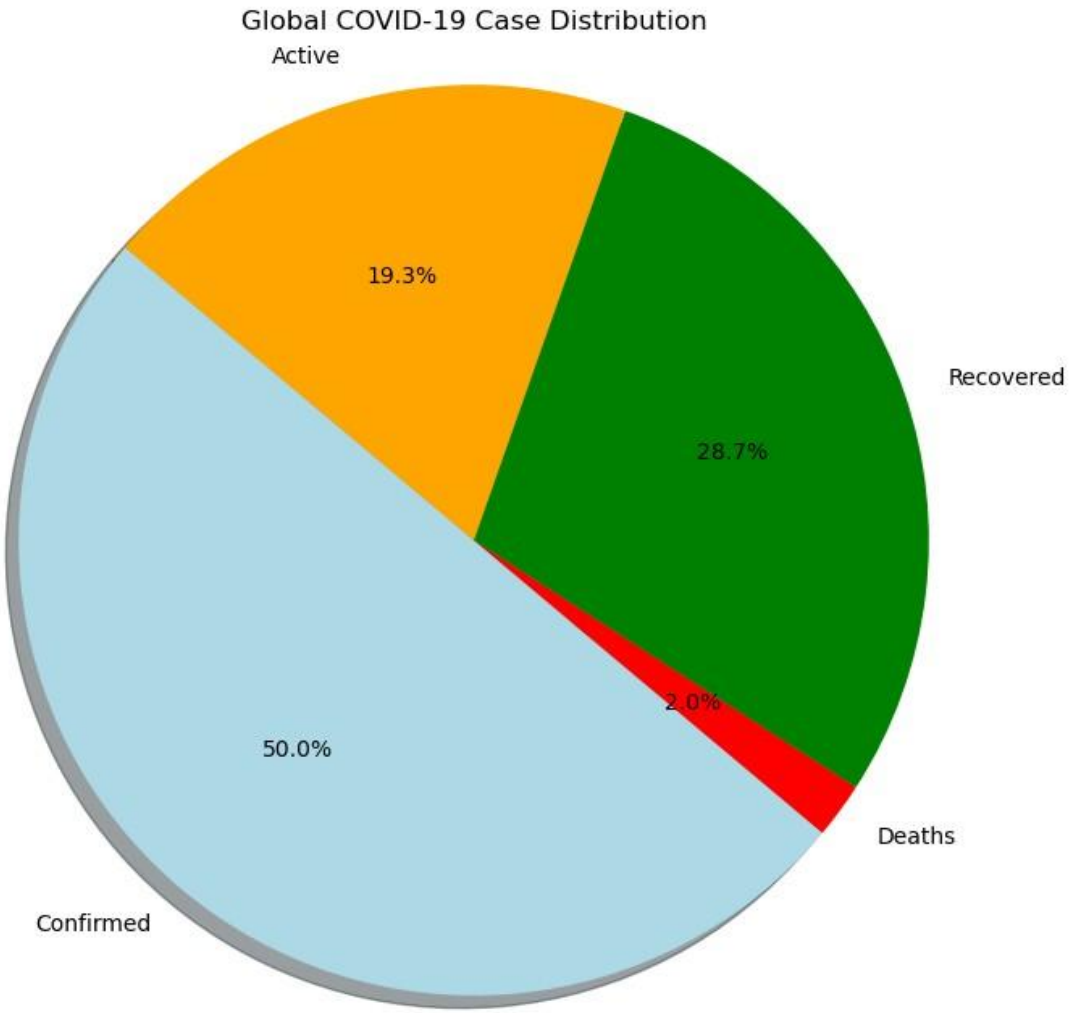
[12]:
```
print(top_confirmed[['Country/Region', 'Confirmed']])
```

```
    Country/Region Confirmed
173             US   4290259
23          Brazil   2442375
79           India   1480073
138         Russia    816680
154   South Africa    452529
111         Mexico    395489
132           Peru    389717
35           Chile    347923
177 United Kingdom   301708
81             Iran    293606
```

[14]:
```
# Step 7: Pie chart of global COVID-19 case
distribution total_confirmed = df['Confirmed'].sum()
total_deaths = df['Deaths'].sum() total_recovered =
df['Recovered'].sum() total_active = df['Active'].sum()
```

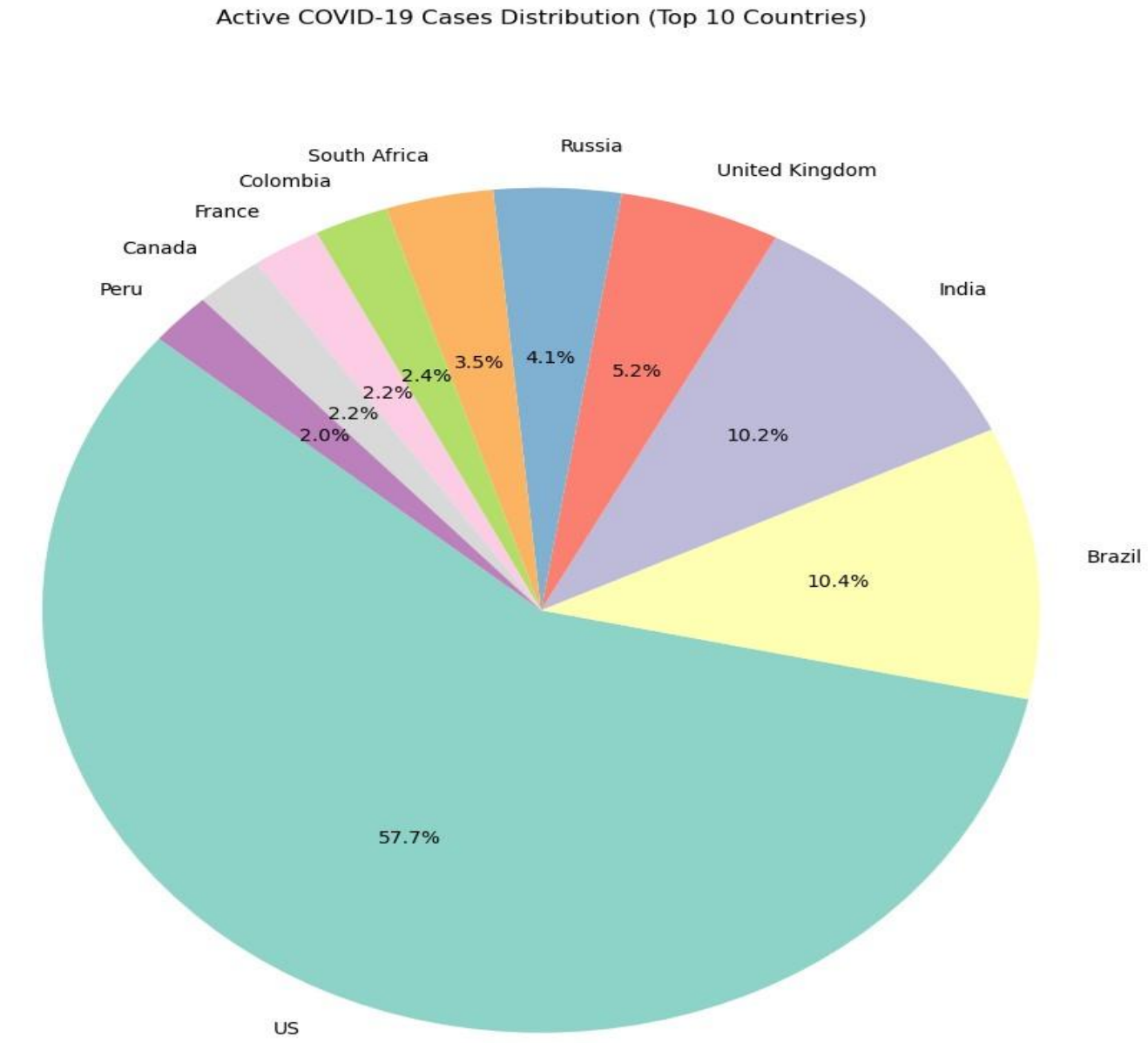
[15]:
```
# Prepare data labels = ['Confirmed', 'Deaths', 'Recovered',
'Active'] sizes = [total_confirmed, total_deaths, total_recovered,
total_active] colors = ['lightblue', 'red', 'green', 'orange']
```

[16]:
```
# Plot
plt.figure(figsize=(8, 8))
plt.pie(sizes, labels=labels, colors=colors, startangle=140, autopct='%1.1f%%',
  ↪shadow=True)
plt.title('Global COVID-19 Case Distribution ')
plt.axis('equal')  # Equal aspect ratio ensures the pie chart is circular
plt.show()
```



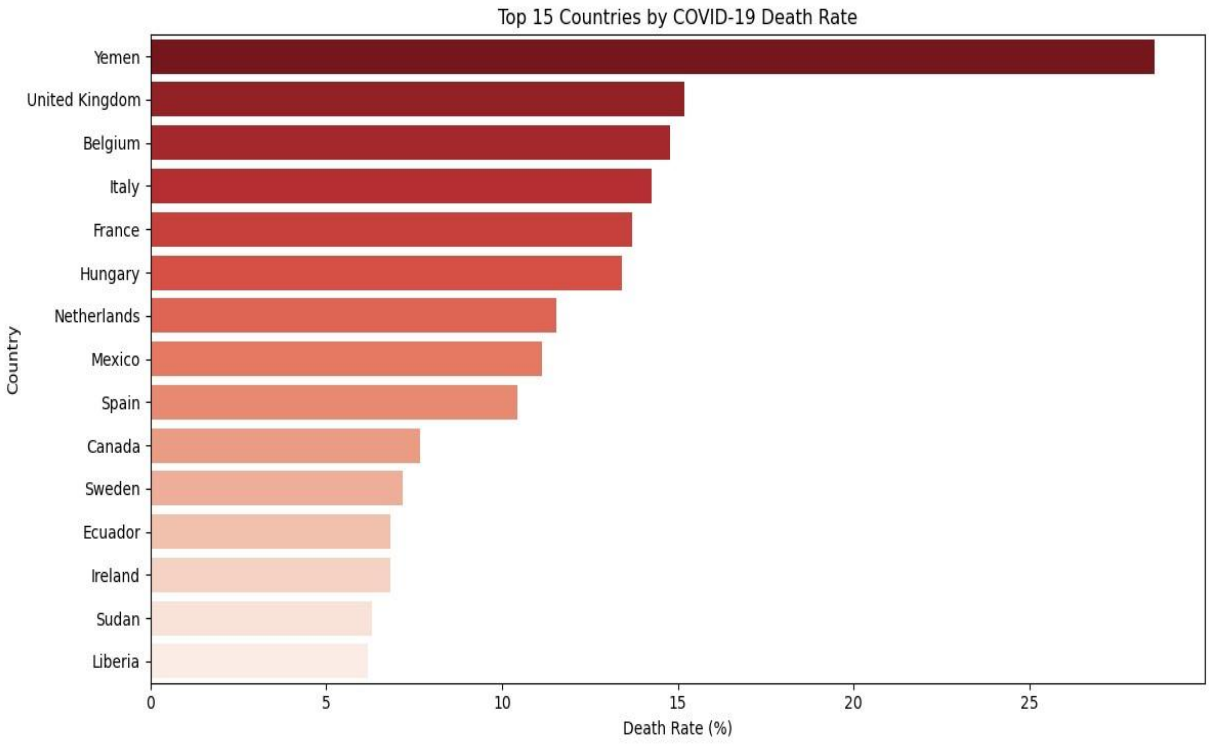
[19]:
```
# Step 11: Pie chart of Active Cases in Top 10 Countries
top_active = df.sort_values(by='Active', ascending=False).head(10)

plt.figure(figsize=(8,10))
plt.pie(
    top_active['Active'],
```

```
    labels=top_active['Country/Region'],
    autopct='%1.1f%%',
    startangle=140,
    colors=sns.color_palette('Set3')
)
plt.title('Active COVID-19 Cases Distribution (Top 10 Countries) ')
plt.axis('equal')  # Equal aspect ratio ensures pie is drawn as a circle.
plt.tight_layout()
plt.show()
```



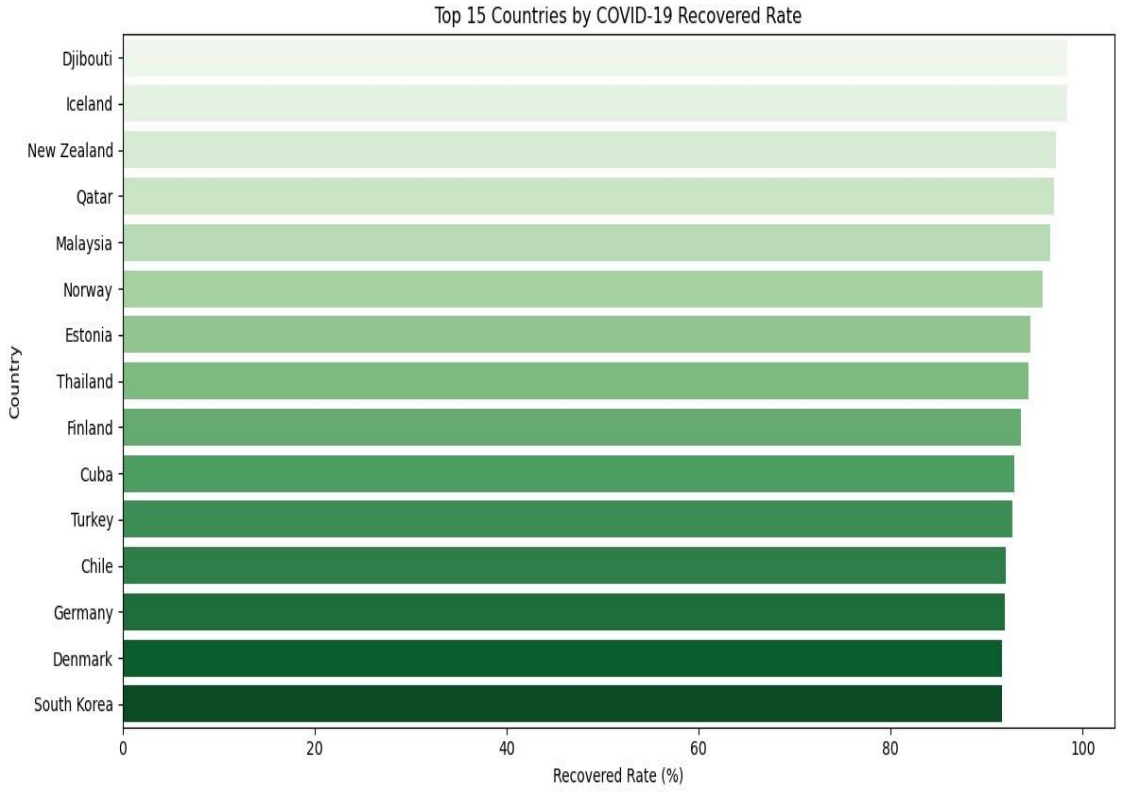
[20]:
```
# Step 14: Bar plot - Top 15 Countries by Death Rate df['Death Rate (%)'] =
(df['Deaths'] / df['Confirmed']) * 100 top15_death_rate =
df[df['Confirmed'] > 1000].sort_values(by='Death Rate (%)',␣
↪ascending=False).head(15) plt.figure(figsize=(12, 6))
```

```
sns.barplot(
    data=top15_death_rate,
    x='Death Rate ( %)',
    y='Country/Region ',
    palette='Reds_r'
)
plt.title('Top 15 Countries by COVID-19 Death Rate    ')
plt.xlabel('Death Rate ( %)')
plt.ylabel('Country')
plt.tight_layout()
plt.show()
```

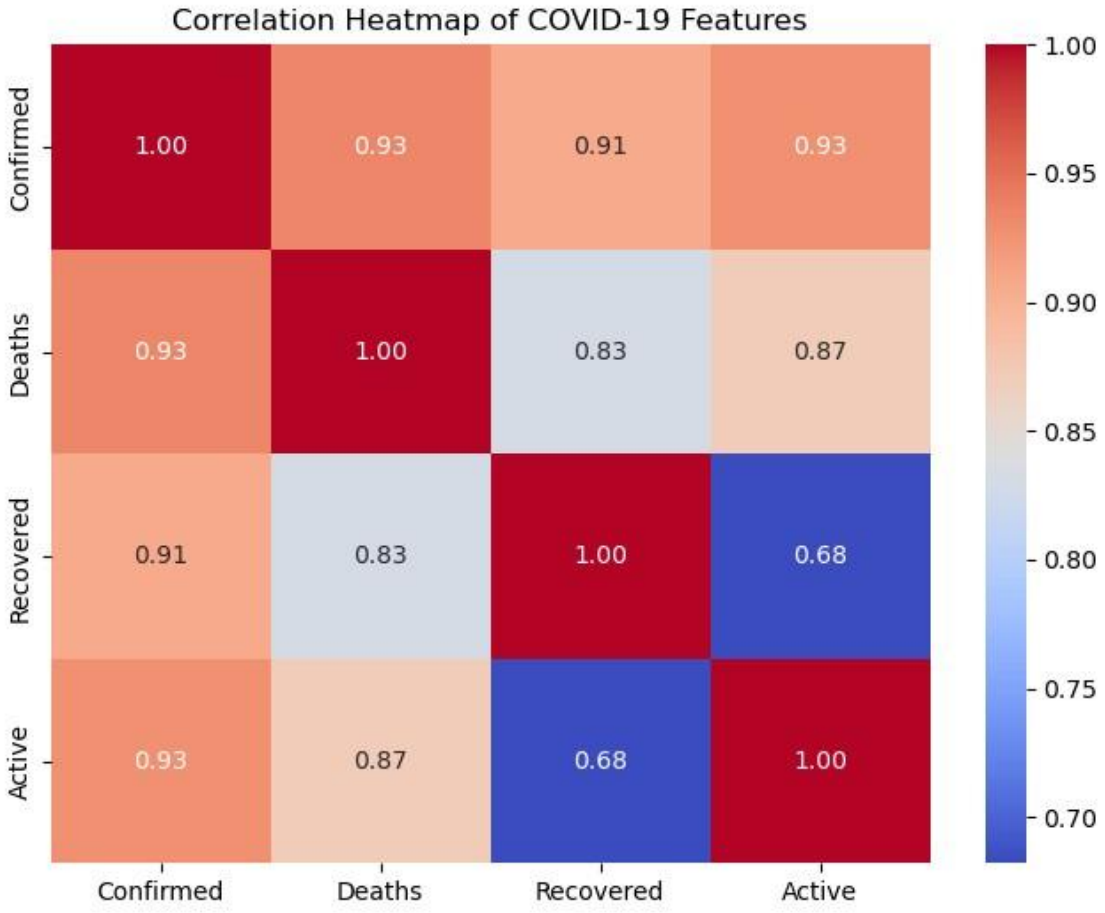


[21]:
```
# Step 15: Bar plot - Top 15 Countries by Recovered Rate
df['Recovered Rate (%)'] = (df['Recovered'] / df['Confirmed']) * 100
top15_recovery_rate = df[df['Confirmed'] > 1000].sort_values(by='Recovered Rate
 ↪(%)', ascending=False).head(15)

plt.figure(figsize=(12, 6))
sns.barplot(
    data=top15_recovery_rate,
    x='Recovered Rate (%)',
    y='Country/Region',
    palette='Greens'
)
plt.title('Top 15 Countries by COVID-19 Recovered Rate ')
plt.xlabel('Recovered Rate (%)')
plt.ylabel('Country')
plt.tight_layout()
```

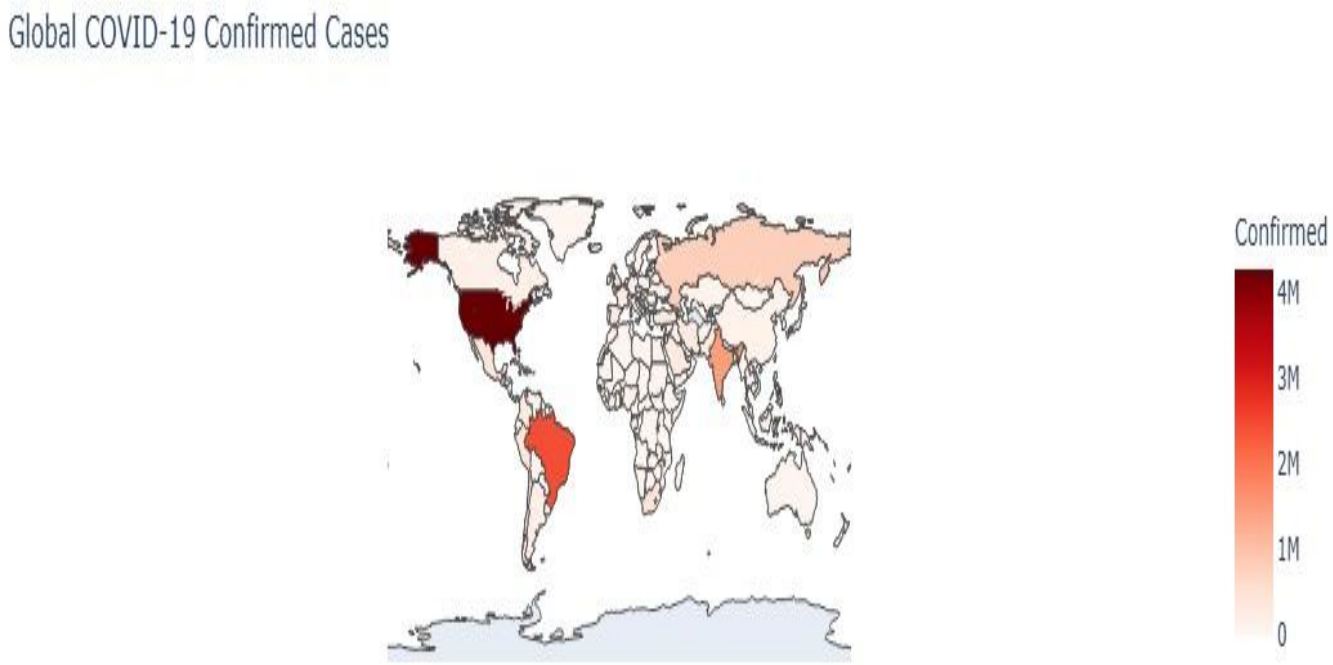
```
plt.show()
```



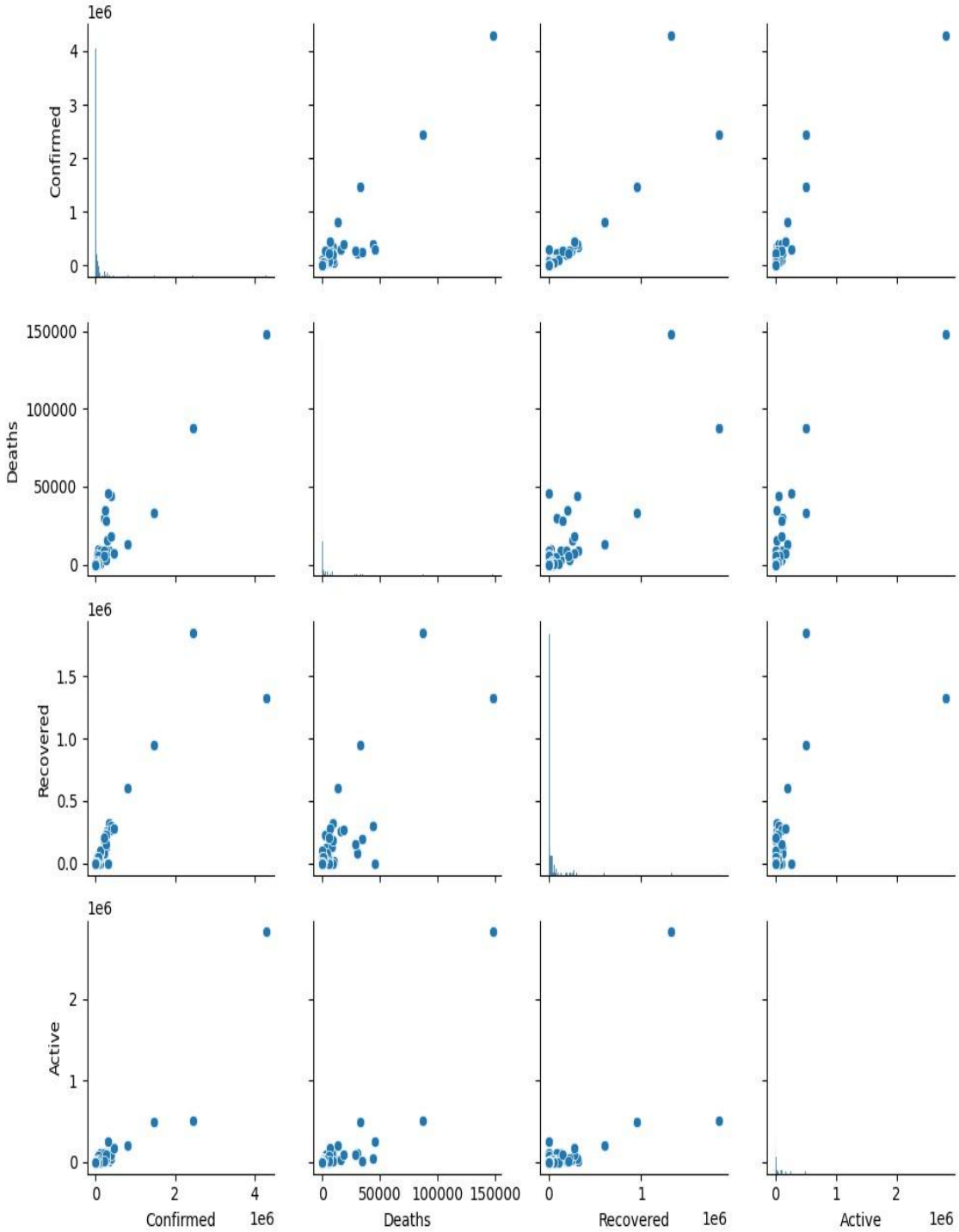
[17]:
```
 # Step 8: Correlation heatmap between COVID-19 features
plt.figure(figsize=(8,6))
sns.heatmap(df[['Confirmed', 'Deaths', 'Recovered',
           'Active']].corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap of COVID-19 Features')
plt.show()
```



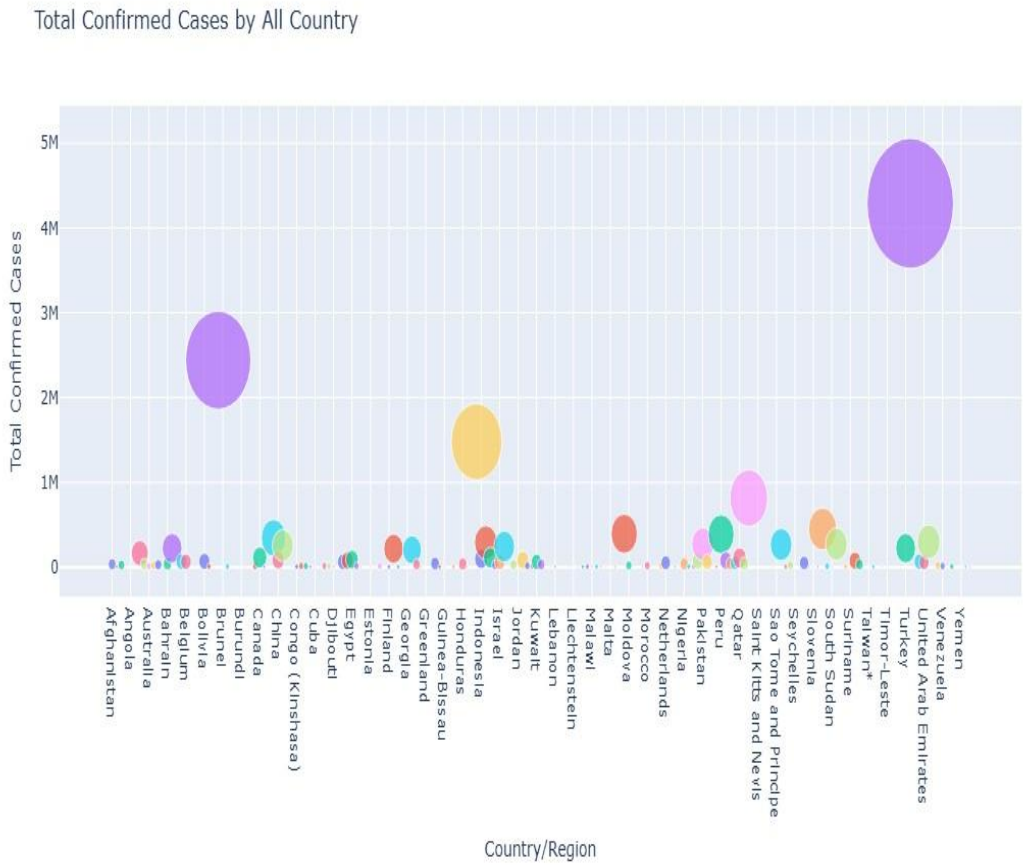
[18]:
```
import plotly.express as px

# Step 9: Choropleth map for Confirmed COVID-19 cases
fig = px.choropleth(
    df,
    locations="Country/Region",
    locationmode="country names",
    color="Confirmed",
    hover_name="Country/Region",
    color_continuous_scale="Reds",
    title="Global COVID-19 Confirmed Cases"
)

fig.update_layout(geo=dict(showframe=False, showcoastlines=True))
fig.show()
```



[23]:
```
# Step 17: Pair Plot for numerical features
sns.pairplot(df[['Confirmed', 'Deaths', 'Recovered', 'Active']])
plt.suptitle('Pairwise Relationships Between COVID-19 Metrics',
y=1.02) plt.tight_layout() plt.show()
```


Pairwise Relationships Between COVID-19 Metrics
Confirmed
Deaths
Recovered
Active
1e6
0
1
2
3
4
50000
100000
150000
0.0
0.5
1.0
1.5

[25]:
```
import plotly.express as px

# Step 18: Interactive bubble chart
fig = px.scatter(df,
                 x="Country/Region",
                 y="Confirmed",
                 size="Confirmed",
                 color="Country/Region",
                 hover_name="Country/Region",
                 size_max=60,
                 title="Total Confirmed Cases by All Country ",
                 labels={"Confirmed": "Total Confirmed Cases"},
                 height=600)

fig.update_layout(showlegend=False)

fig.show()
```



[ ]: