COVID VACCINES ANALYSIS

NAME: GANGAPATNAM SASIDEEKSHATH
REG NO:au723921104015

dac-phase3

October 17, 2023

```python
[1]: # This Python 3 environment comes with many helpful analytics libraries␣
     ↪installed
     # It is defined by the kaggle/python Docker image: https://github.com/kaggle/
     ↪docker-python
     # For example, here's several helpful packages to load

     import numpy as np # linear algebra
     import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

     # Input data files are available in the read-only "../input/" directory
     # For example, running this (by clicking run or pressing Shift+Enter) will list␣
     ↪all files under the input directory

     import os
     for dirname, _, filenames in os.walk('/kaggle/input'):
         for filename in filenames:
             print(os.path.join(dirname, filename))

     # You can write up to 20GB to the current directory (/kaggle/working/) that␣
     ↪gets preserved as output when you create a version using "Save & Run All"
     # You can also write temporary files to /kaggle/temp/, but they won't be saved␣
     ↪outside of the current session
```

```
/kaggle/input/corona-virus-report/covid_19_clean_complete.csv
/kaggle/input/corona-virus-report/country_wise_latest.csv
/kaggle/input/corona-virus-report/day_wise.csv
/kaggle/input/corona-virus-report/usa_county_wise.csv
/kaggle/input/corona-virus-report/worldometer_data.csv
/kaggle/input/corona-virus-report/full_grouped.csv
```

0.1 Import Liabraries

```python
[2]: import seaborn as sns
     import matplotlib.pyplot as plt
     import plotly.express as px
```

1

```
[3]: df=pd.read_csv("/kaggle/input/corona-virus-report/covid_19_clean_complete.csv")
     df.head()
```

```
[3]:   Province/State Country/Region       Lat       Long       Date  Confirmed  \
     0            NaN    Afghanistan  33.93911  67.709953  2020-01-22          0
     1            NaN        Albania  41.15330  20.168300  2020-01-22          0
     2            NaN        Algeria  28.03390   1.659600  2020-01-22          0
     3            NaN        Andorra  42.50630   1.521800  2020-01-22          0
     4            NaN         Angola -11.20270  17.873900  2020-01-22          0

        Deaths  Recovered  Active            WHO Region
     0       0          0       0  Eastern Mediterranean
     1       0          0       0                 Europe
     2       0          0       0                 Africa
     3       0          0       0                 Europe
     4       0          0       0                 Africa
```

```
[4]: df.shape
```

```
[4]: (49068, 10)
```

```
[5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49068 entries, 0 to 49067
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Province/State  14664 non-null  object
 1   Country/Region  49068 non-null  object
 2   Lat             49068 non-null  float64
 3   Long            49068 non-null  float64
 4   Date            49068 non-null  object
 5   Confirmed       49068 non-null  int64
 6   Deaths          49068 non-null  int64
 7   Recovered       49068 non-null  int64
 8   Active          49068 non-null  int64
 9   WHO Region      49068 non-null  object
dtypes: float64(2), int64(4), object(4)
memory usage: 3.7+ MB
```

```
[6]: df['Month'] = pd.to_datetime(df['Date']).dt.month
     df['Year'] = pd.to_datetime(df['Date']).dt.year
```

```
[7]: df.isnull().sum()
```

```
[7]: Province/State    34404
     Country/Region        0
     Lat                   0
     Long                  0
     Date                  0
     Confirmed             0
     Deaths                0
     Recovered             0
     Active                0
     WHO Region            0
     Month                 0
     Year                  0
     dtype: int64
```

```
[8]: total_confirmed = df.groupby("Country/Region")["Confirmed"].sum().
     ↪sort_values(ascending=False).head(20).reset_index()
```
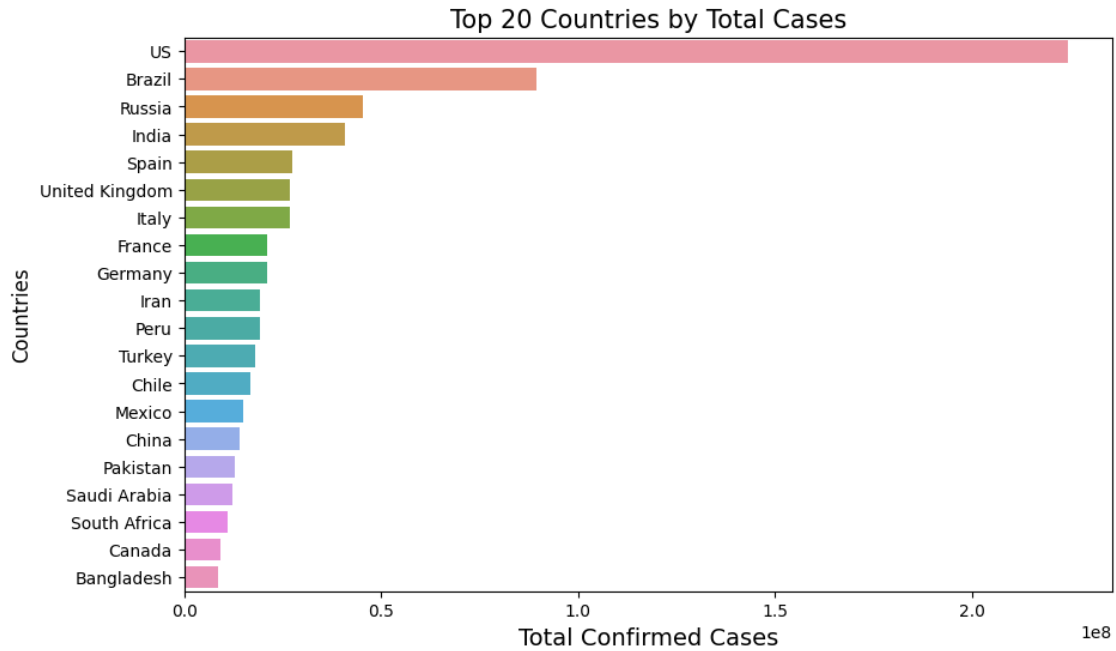
```
[9]: x_column = 'Confirmed'
     y_column = 'Country/Region'

     plt.figure(figsize=(10, 6))

     sns.barplot(x=x_column, y=y_column, data=total_confirmed)

     plt.xlabel('Total Confirmed Cases', fontsize=14)
     plt.ylabel('Countries', fontsize=12)
     plt.title('Top 20 Countries by Total Cases', fontsize=15)

     plt.show()
```

Top 20 Countries by Total Cases

[10]: `total_confirmed`

[10]:
```
    Country/Region  Confirmed
0               US  224345948
1           Brazil   89524967
2           Russia   45408411
3            India   40883464
4            Spain   27404045
5   United Kingdom   26748587
6            Italy   26745145
7           France   21210926
8          Germany   21059152
9             Iran   19339267
10            Peru   19263916
11          Turkey   17903345
12           Chile   16935654
13          Mexico   14946202
14           China   14132002
15        Pakistan   12833994
16    Saudi Arabia   12362961
17    South Africa   11168743
18          Canada    9356551
19      Bangladesh    8754729
```

[11]: ```python
total_death = df.groupby("Country/Region")["Deaths"].sum().
    sort_values(ascending=False).head(20).reset_index()
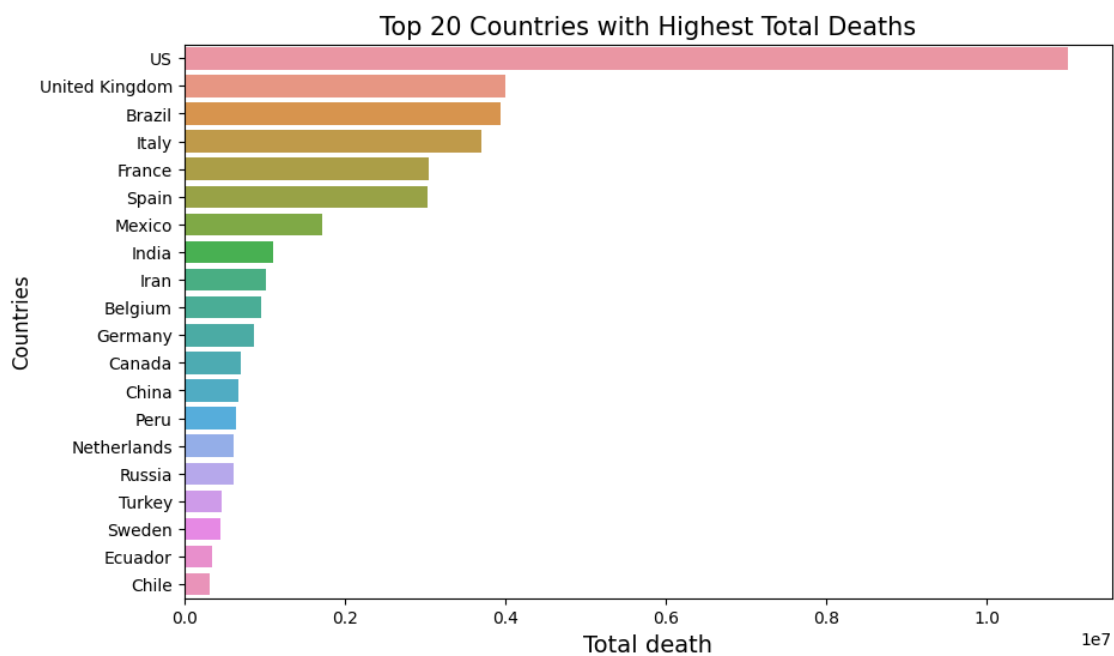```

```
[12]: x_column = 'Deaths'
      y_column = 'Country/Region'

      plt.figure(figsize=(10, 6))

      sns.barplot(x=x_column, y=y_column, data=total_death)

      plt.xlabel('Total death', fontsize=14)
      plt.ylabel('Countries', fontsize=12)
      plt.title('Top 20 Countries with Highest Total Deaths', fontsize=15)

      plt.show()
```



```
[13]: total_death
```

```
[13]:      Country/Region     Deaths
      0                US   11011411
      1    United Kingdom    3997775
      2            Brazil    3938034
      3             Italy    3707717
      4            France    3048524
      5             Spain    3033030
      6            Mexico    1728277
      7             India    1111831
      8              Iran    1024136
      9           Belgium     963679
```

```
10        Germany     871322
11         Canada     699566
12          China     672413
13           Peru     652113
14     Netherlands    622314
15         Russia     619385
16         Turkey     466056
17         Sweden     448913
18        Ecuador     346618
19          Chile     322480
```

[14]: ```python
total_recover = df.groupby("Country/Region")["Recovered"].sum().
↪sort_values(ascending=False).head(20).reset_index()
```
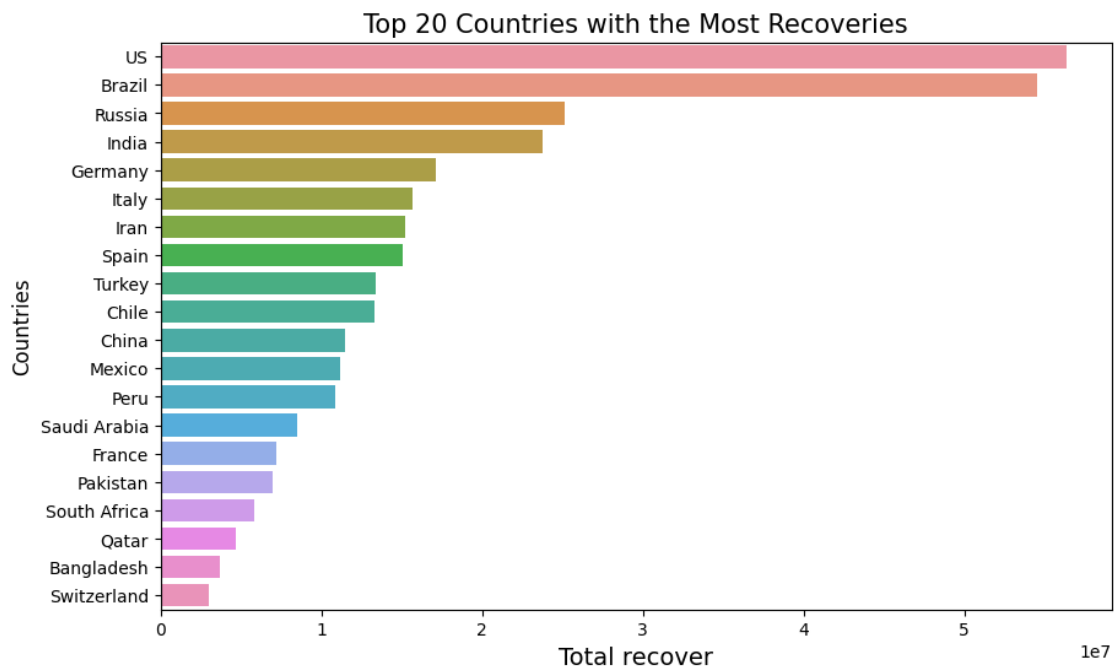
[15]: ```python
x_column = 'Recovered'
y_column = 'Country/Region'

plt.figure(figsize=(10, 6))

sns.barplot(x=x_column, y=y_column, data=total_recover)

plt.xlabel('Total recover', fontsize=14)
plt.ylabel('Countries', fontsize=12)
plt.title('Top 20 Countries with the Most Recoveries', fontsize=15)
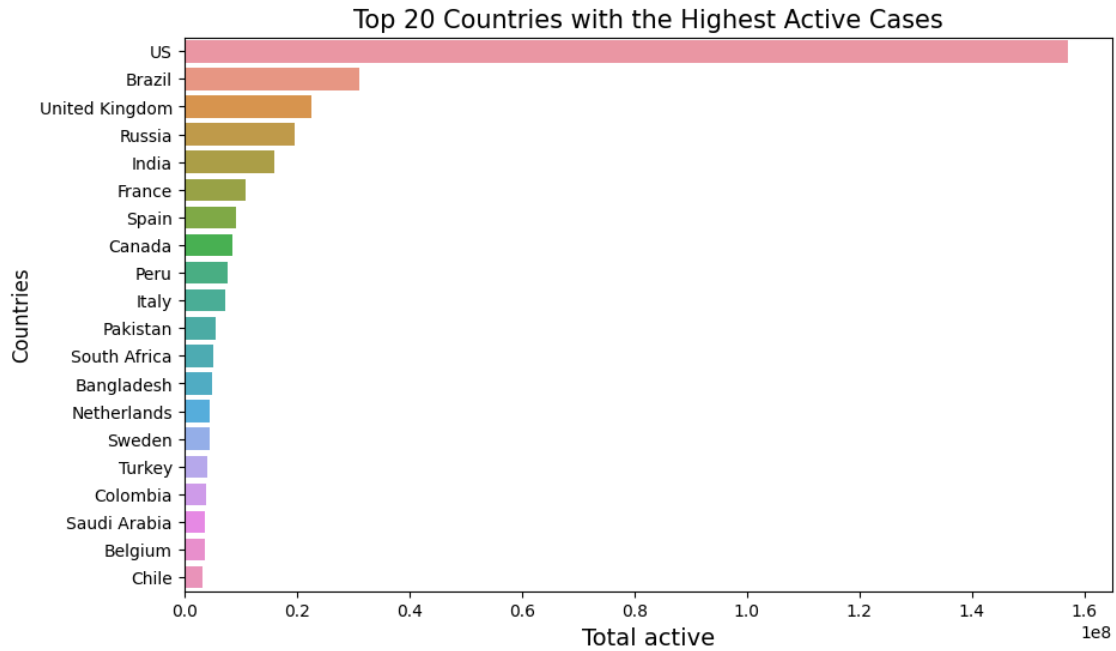
plt.show()
```

```
[16]: total_recover
```

```
[16]:     Country/Region  Recovered
      0               US   56353416
      1           Brazil   54492873
      2           Russia   25120448
      3            India   23783720
      4          Germany   17107839
      5            Italy   15673910
      6             Iran   15200895
      7            Spain   15093583
      8           Turkey   13345389
      9            Chile   13292593
      10           China   11466866
      11          Mexico   11141225
      12            Peru   10862846
      13    Saudi Arabia    8474107
      14          France    7182115
      15        Pakistan    6936003
      16    South Africa    5836423
      17           Qatar    4676443
      18      Bangladesh    3714702
      19     Switzerland    2957883
```

```
[17]: total_active = df.groupby("Country/Region")["Active"].sum().
       ↪sort_values(ascending=False).head(20).reset_index()
```

```
[18]: x_column = 'Active'
      y_column = 'Country/Region'

      plt.figure(figsize=(10, 6))   # Set the figure size as per your preference

      sns.barplot(x=x_column, y=y_column, data=total_active)

      plt.xlabel('Total active', fontsize=14)
      plt.ylabel('Countries', fontsize=12)
      plt.title('Top 20 Countries with the Highest Active Cases', fontsize=15)
      plt.show()
```

Top 20 Countries with the Highest Active Cases


```
[19]: total_active
```

```
[19]:    Country/Region      Active
      0              US  156981121
      1           Brazil   31094060
      2   United Kingdom   22624595
      3           Russia   19668578
      4            India   15987913
      5           France   10980287
      6            Spain    9277432
      7           Canada    8656985
      8             Peru    7748957
      9            Italy    7363518
      10        Pakistan    5633262
      11    South Africa    5150341
      12      Bangladesh    4924394
      13     Netherlands    4528235
      14          Sweden    4524247
      15          Turkey    4091900
      16        Colombia    3832786
      17    Saudi Arabia    3783704
      18         Belgium    3689945
      19           Chile    3320581
```

```
[20]: fig = px.sunburst(df, path=['WHO Region', 'Country/Region'], values='Confirmed')

      fig.update_layout(width = 700,
                        height = 600,
                        title = 'Total Confirmed',
                        title_x=0.5)

      fig.show();
```

```
[21]: fig = px.sunburst(df, path=['WHO Region', 'Country/Region'], values='Deaths')

      fig.update_layout(width = 700,
                        height = 600,
                        title = 'Total Deaths',
                        title_x=0.5)

      fig.show();
```

```
[22]: fig = px.sunburst(df, path=['WHO Region', 'Country/Region'], values='Recovered')

      fig.update_layout(width = 700,
                        height = 600,
                        title = 'Total Recovered',
                        title_x=0.5)

      fig.show();
```

```
[23]: fig = px.sunburst(df, path=['WHO Region', 'Country/Region'], values='Active')

      fig.update_layout(width = 700,
                        height = 600,
                        title = 'Total Active',
                        title_x=0.5)

      fig.show();
```

```
[24]: x_column = 'Month'
      y_column = 'Deaths'

      custom_palette = sns.color_palette("Set2")
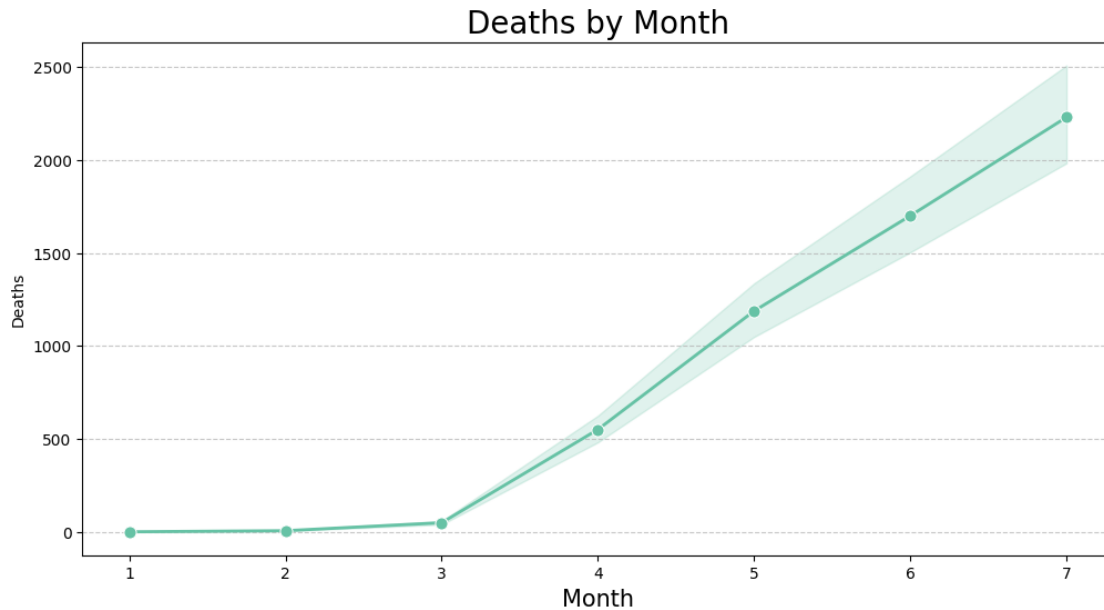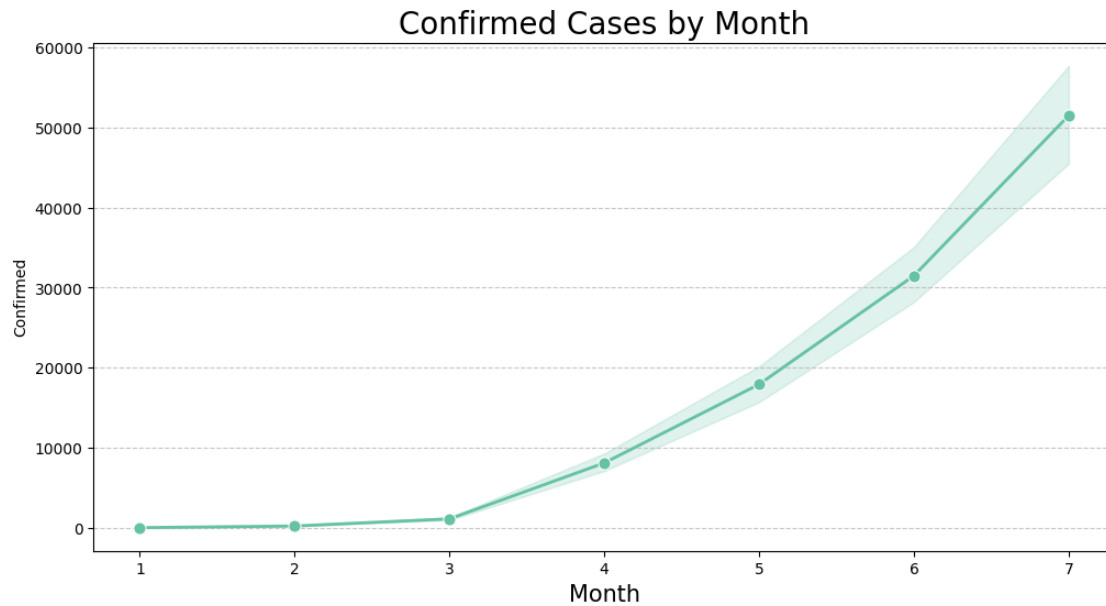
      plt.figure(figsize=(12, 6))

      sns.lineplot(x=x_column, y=y_column, data=df, linewidth=2,
       ↪color=custom_palette[0], marker='o', markersize=8)

      plt.xlabel('Month', fontsize=15)
```

```
plt.title('Deaths by Month', fontsize=20)

plt.grid(axis='y', linestyle='--', alpha=0.7)

# Customize the legend
plt.show()
```

Deaths by Month


```
x_column = 'Month'
y_column = 'Confirmed'
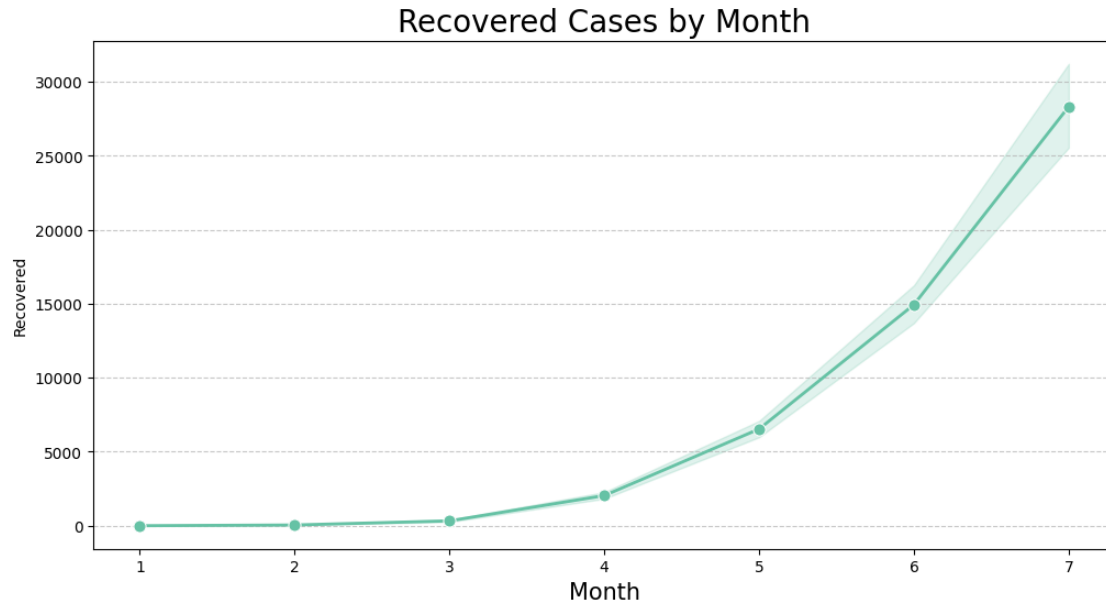
custom_palette = sns.color_palette("Set2")

plt.figure(figsize=(12, 6))

sns.lineplot(x=x_column, y=y_column, data=df, linewidth=2,␣
 ↪color=custom_palette[0], marker='o', markersize=8)

plt.xlabel('Month', fontsize=15)
plt.title('Confirmed Cases by Month', fontsize=20)

plt.grid(axis='y', linestyle='--', alpha=0.7)


plt.show()
```

Confirmed Cases by Month


```
[26]: x_column = 'Month'
      y_column = 'Recovered'

      custom_palette = sns.color_palette("Set2")

      plt.figure(figsize=(12, 6))

      sns.lineplot(x=x_column, y=y_column, data=df, linewidth=2,␣
        ↪color=custom_palette[0], marker='o', markersize=8)

      plt.xlabel('Month', fontsize=15)
      plt.title('Recovered Cases by Month', fontsize=20)

      plt.grid(axis='y', linestyle='--', alpha=0.7)


      plt.show()
```

Recovered Cases by Month


```
[27]: x_column = 'Month'
      y_column = 'Active'

      custom_palette = sns.color_palette("Set2")

      plt.figure(figsize=(12, 6))

      sns.lineplot(x=x_column, y=y_column, data=df, linewidth=2,␣
       ↪color=custom_palette[0], marker='o', markersize=8)

      plt.xlabel('Month', fontsize=15)
      plt.title('Active Cases by Month', fontsize=20)

      plt.grid(axis='y', linestyle='--', alpha=0.7)
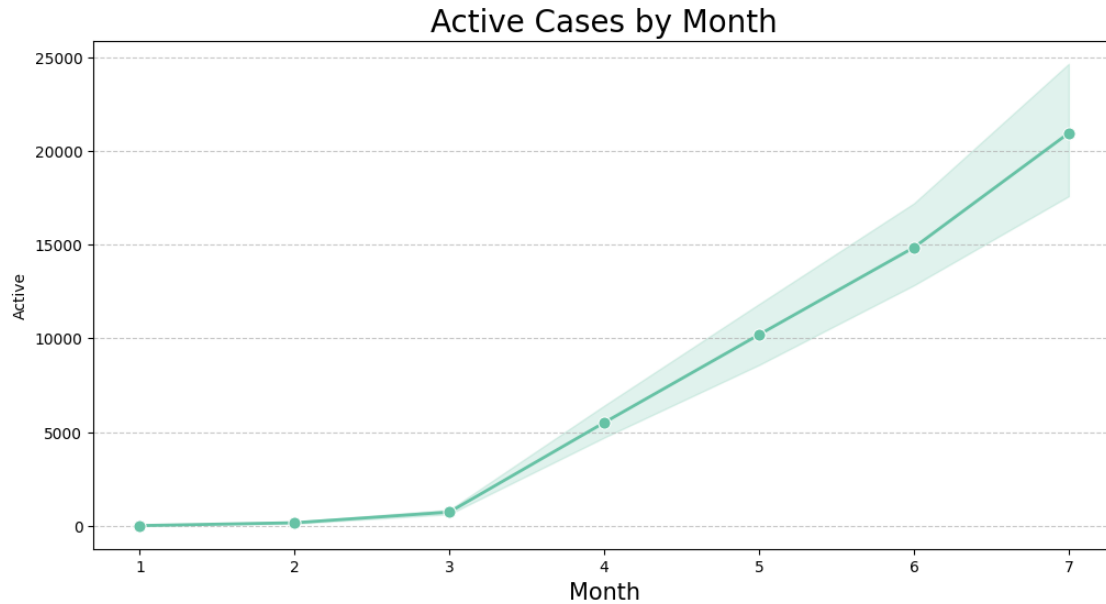

      plt.show()
```

Active Cases by Month


```
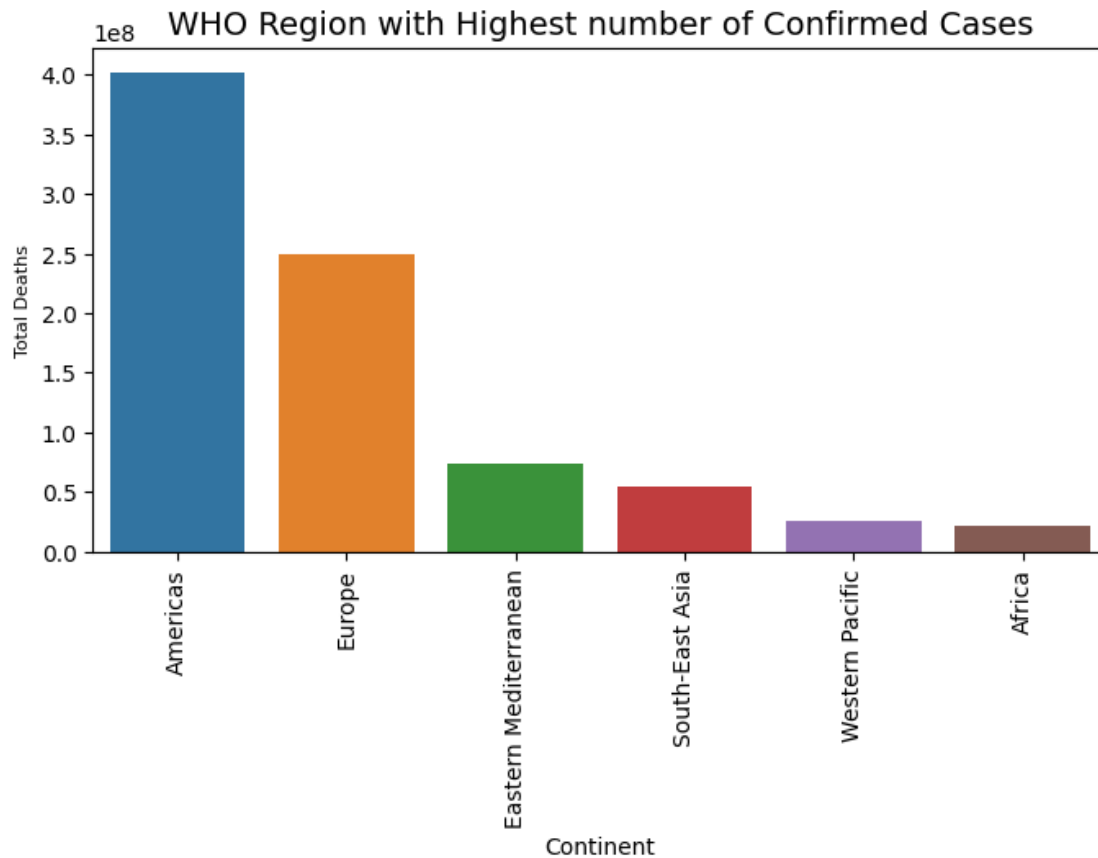[28]: df_death = df.groupby("WHO Region")["Deaths"].sum().
      ↪sort_values(ascending=False).head(10).reset_index()
```

```
[29]: x_column = 'WHO Region'
      y_column = 'Deaths'

      plt.figure(figsize=(8, 4))

      sns.barplot(x=x_column, y=y_column, data=df_death)

      plt.xlabel('Continent', fontsize=4)
      plt.ylabel('Total Deaths', fontsize=8)
      plt.title('WHO Region with Highest number of Deaths', fontsize=14)

      plt.xticks(rotation=90)
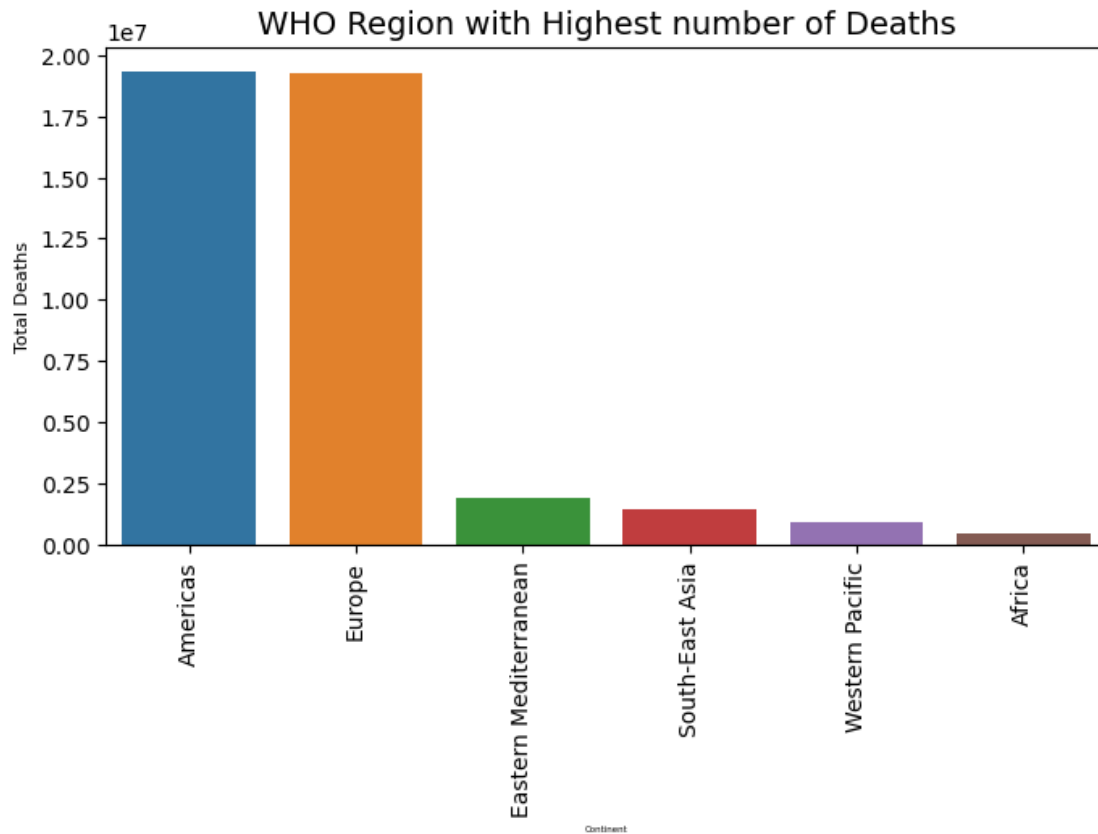
      plt.show()
```

WHO Region with Highest number of Deaths

```
[30]: df_conferm = df.groupby("WHO Region")["Confirmed"].sum().
      ↪sort_values(ascending=False).head(10).reset_index()
```

```
[31]: x_column = 'WHO Region'
      y_column = 'Confirmed'

      plt.figure(figsize=(8, 4))

      sns.barplot(x=x_column, y=y_column, data=df_conferm)

      plt.xlabel('Continent', fontsize=10)
      plt.ylabel('Total Deaths', fontsize=8)
      plt.title('WHO Region with Highest number of Confirmed Cases', fontsize=14)

      plt.xticks(rotation=90)

      plt.show()
```

WHO Region with Highest number of Confirmed Cases


```
[32]: df_recover = df.groupby("WHO Region")["Recovered"].sum().
       ↪sort_values(ascending=False).head(10).reset_index()
```

```
[33]: x_column = 'WHO Region'
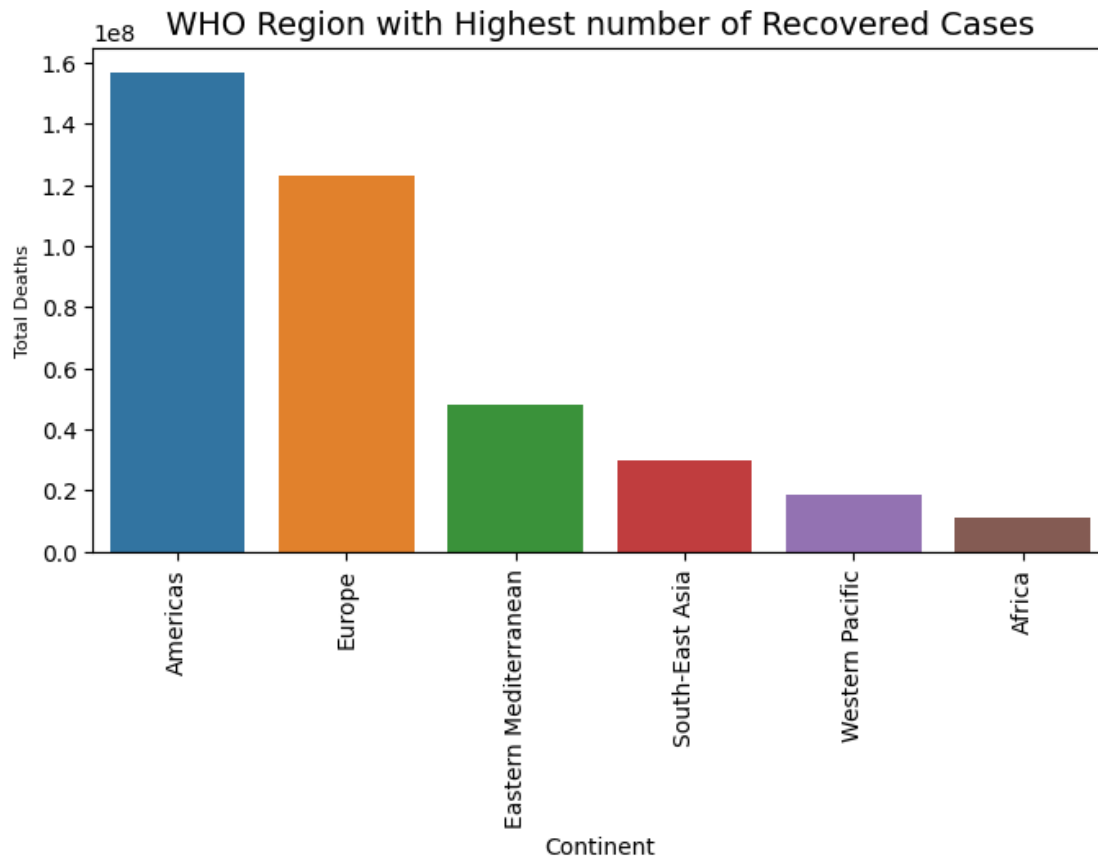      y_column = 'Recovered'

      plt.figure(figsize=(8, 4))

      sns.barplot(x=x_column, y=y_column, data=df_recover)

      plt.xlabel('Continent', fontsize=10)
      plt.ylabel('Total Deaths', fontsize=8)
      plt.title('WHO Region with Highest number of Recovered Cases', fontsize=14)

      plt.xticks(rotation=90)

      plt.show()
```

WHO Region with Highest number of Recovered Cases

```
[34]: df_active = df.groupby("WHO Region")["Active"].sum().
      ↪sort_values(ascending=False).head(10).reset_index()
```

```
[35]: x_column = 'WHO Region'
      y_column = 'Active'

      plt.figure(figsize=(8, 4))

      sns.barplot(x=x_column, y=y_column, data=df_active)

      plt.xlabel('Continent', fontsize=10)
      plt.ylabel('Total Deaths', fontsize=8)
      plt.title('WHO Region with Highest number of Active Cases', fontsize=14)

      plt.xticks(rotation=90)

      plt.show()
```

WHO Region with Highest number of Active Cases