In [34]:

```python
import pandas as pd

try:
    df = pd.read_csv('owid-covid-data.csv')

    df.head()
    df.shape
    df.columns
    df.isnull().sum()
    df.info()
except Exception as e:
    print('Error loading dataset: {e}')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 302512 entries, 0 to 302511
Data columns (total 67 columns):
 #   Column                                      Non-Null Count   Dtype  
---  ------                                      --------------   -----  
 0   iso_code                                    302512 non-null  object 
 1   continent                                   288160 non-null  object 
 2   location                                    302512 non-null  object 
 3   date                                        302512 non-null  object 
 4   total_cases                                 266771 non-null  float64
 5   new_cases                                   294064 non-null  float64
 6   new_cases_smoothed                          292800 non-null  float64
 7   total_deaths                                246214 non-null  float64
 8   new_deaths                                  294139 non-null  float64
 9   new_deaths_smoothed                         292909 non-null  float64
 10  total_cases_per_million                     266771 non-null  float64
 11  new_cases_per_million                       294064 non-null  float64
 12  new_cases_smoothed_per_million              292800 non-null  float64
 13  total_deaths_per_million                    246214 non-null  float64
 14  new_deaths_per_million                      294139 non-null  float64
 15  new_deaths_smoothed_per_million             292909 non-null  float64
 16  reproduction_rate                           184817 non-null  float64
 17  icu_patients                                34764 non-null   float64
 18  icu_patients_per_million                    34764 non-null   float64
 19  hosp_patients                               35138 non-null   float64
 20  hosp_patients_per_million                   35138 non-null   float64
 21  weekly_icu_admissions                       9101 non-null    float64
 22  weekly_icu_admissions_per_million           9101 non-null    float64
 23  weekly_hosp_admissions                      21287 non-null   float64
 24  weekly_hosp_admissions_per_million          21287 non-null   float64
 25  total_tests                                 79387 non-null   float64
 26  new_tests                                   75403 non-null   float64
 27  total_tests_per_thousand                    79387 non-null   float64
 28  new_tests_per_thousand                      75403 non-null   float64
 29  new_tests_smoothed                          103965 non-null  float64
 30  new_tests_smoothed_per_thousand             103965 non-null  float64
 31  positive_rate                               95927 non-null   float64
 32  tests_per_case                              94348 non-null   float64
 33  tests_units                                 106788 non-null  object 
 34  total_vaccinations                          73561 non-null   float64
 35  people_vaccinated                           70411 non-null   float64
 36  people_fully_vaccinated                     68149 non-null   float64
 37  total_boosters                              42324 non-null   float64
 38  new_vaccinations                            60542 non-null   float64
 39  new_vaccinations_smoothed                   163536 non-null  float64
 40  total_vaccinations_per_hundred              73561 non-null   float64
 41  people_vaccinated_per_hundred               70411 non-null   float64
 42  people_fully_vaccinated_per_hundred         68149 non-null   float64
 43  total_boosters_per_hundred                  42324 non-null   float64
 44  new_vaccinations_smoothed_per_million       163536 non-null  float64
 45  new_people_vaccinated_smoothed              163587 non-null  float64
 46  new_people_vaccinated_smoothed_per_hundred  163587 non-null  float64
 47  stringency_index                            193194 non-null  float64
 48  population_density                          256703 non-null  float64
 49  median_age                                  238751 non-null  float64
 50  aged_65_older                               230391 non-null  float64
 51  aged_70_older                               236359 non-null  float64
 52  gdp_per_capita                              233979 non-null  float64
 53  extreme_poverty                             150700 non-null  float64
 54  cardiovasc_death_rate                       234406 non-null  float64
 55  diabetes_prevalence                         246348 non-null  float64
 56  female_smokers                              175815 non-null  float64
 57  male_smokers                                173423 non-null  float64
 58  handwashing_facilities                      114817 non-null  float64
 59  hospital_beds_per_thousand                  206911 non-null  float64
 60  life_expectancy                             278219 non-null  float64
 61  human_development_index                     227212 non-null  float64
 62  population                                  302512 non-null  float64
 63  excess_mortality_cumulative_absolute        10295 non-null   float64
 64  excess_mortality_cumulative                 10295 non-null   float64
 65  excess_mortality                            10295 non-null   float64
 66  excess_mortality_cumulative_per_million     10295 non-null   float64
dtypes: float64(62), object(5)
memory usage: 154.6+ MB
```

In [5]:

```
# filter by specific countries
countries_of_interest = ['Kenya', 'United States', 'Germany']
df_filtered = df[df['location'].isin(countries_of_interest)]
```

In [7]:

```
# drop missing dates or total_cases
df_filtered = df_filtered.dropna(subset=['date', 'total_cases'])
```

In [8]:

```
# convert date to datetime
df_filtered['date'] = pd.to_datetime(df_filtered['date'])
```

In [35]:

```
# fill columns with 0 where missing
df_filtered[['new_cases', 'new_deaths']] = df_filtered[['new_cases', 'new_deaths']].fillna(0)
```

In [24]:

```
# preview
df_filtered.info()
df_filtered.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3499 entries, 0 to 3498
Data columns (total 68 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   iso_code                                    3499 non-null   object
 1   continent                                   3499 non-null   object
 2   location                                    3499 non-null   object
 3   date                                        3499 non-null   datetime64[ns]
 4   total_cases                                 3499 non-null   float64
 5   new_cases                                   3499 non-null   float64
 6   new_cases_smoothed                          3495 non-null   float64
 7   total_deaths                                3398 non-null   float64
 8   new_deaths                                  3499 non-null   float64
 9   new_deaths_smoothed                         3495 non-null   float64
 10  total_cases_per_million                     3499 non-null   float64
 11  new_cases_per_million                       3499 non-null   float64
 12  new_cases_smoothed_per_million              3495 non-null   float64
 13  total_deaths_per_million                    3398 non-null   float64
 14  new_deaths_per_million                      3499 non-null   float64
 15  new_deaths_smoothed_per_million             3495 non-null   float64
 16  reproduction_rate                           3376 non-null   float64
 17  icu_patients                                2121 non-null   float64
 18  icu_patients_per_million                    2121 non-null   float64
 19  hosp_patients                               1002 non-null   float64
 20  hosp_patients_per_million                   1002 non-null   float64
 21  weekly_icu_admissions                       617 non-null    float64
 22  weekly_icu_admissions_per_million           617 non-null    float64
 23  weekly_hosp_admissions                      2134 non-null   float64
 24  weekly_hosp_admissions_per_million          2134 non-null   float64
 25  total_tests                                 3384 non-null   float64
 26  new_tests                                   2259 non-null   float64
 27  total_tests_per_thousand                    3384 non-null   float64
 28  new_tests_per_thousand                      2259 non-null   float64
 29  new_tests_smoothed                          3373 non-null   float64
 30  new_tests_smoothed_per_thousand             3373 non-null   float64
 31  positive_rate                               3377 non-null   float64
 32  tests_per_case                              3377 non-null   float64
 33  tests_units                                 2491 non-null   object
 34  total_vaccinations                          2458 non-null   float64
 35  people_vaccinated                           2458 non-null   float64
 36  people_fully_vaccinated                     2351 non-null   float64
 37  total_boosters                              1968 non-null   float64
 38  new_vaccinations                            2421 non-null   float64
 39  new_vaccinations_smoothed                   2455 non-null   float64
 40  total_vaccinations_per_hundred              2458 non-null   float64
 41  people_vaccinated_per_hundred               2458 non-null   float64
 42  people_fully_vaccinated_per_hundred         2351 non-null   float64
 43  total_boosters_per_hundred                  1968 non-null   float64
 44  new_vaccinations_smoothed_per_million       2455 non-null   float64
 45  new_people_vaccinated_smoothed              2455 non-null   float64
 46  new_people_vaccinated_smoothed_per_hundred  2455 non-null   float64
 47  stringency_index                            3479 non-null   float64
 48  population_density                          3499 non-null   float64
 49  median_age                                  3499 non-null   float64
 50  aged_65_older                               3499 non-null   float64
 51  aged_70_older                               3499 non-null   float64
 52  gdp_per_capita                              3499 non-null   float64
 53  extreme_poverty                             2304 non-null   float64
 54  cardiovasc_death_rate                       3499 non-null   float64
 55  diabetes_prevalence                         3499 non-null   float64
 56  female_smokers                              3499 non-null   float64
 57  male_smokers                                3499 non-null   float64
 58  handwashing_facilities                      1125 non-null   float64
 59  hospital_beds_per_thousand                  3499 non-null   float64
 60  life_expectancy                             3499 non-null   float64
 61  human_development_index                     3499 non-null   float64
 62  population                                  3499 non-null   float64
 63  excess_mortality_cumulative_absolute        2367 non-null   float64
 64  excess_mortality_cumulative                 2367 non-null   float64
 65  excess_mortality                            2367 non-null   float64
 66  excess_mortality_cumulative_per_million     2367 non-null   float64
 67  death_rate                                  3398 non-null   float64
dtypes: datetime64[ns](1), float64(63), object(4)
memory usage: 1.8+ MB
```

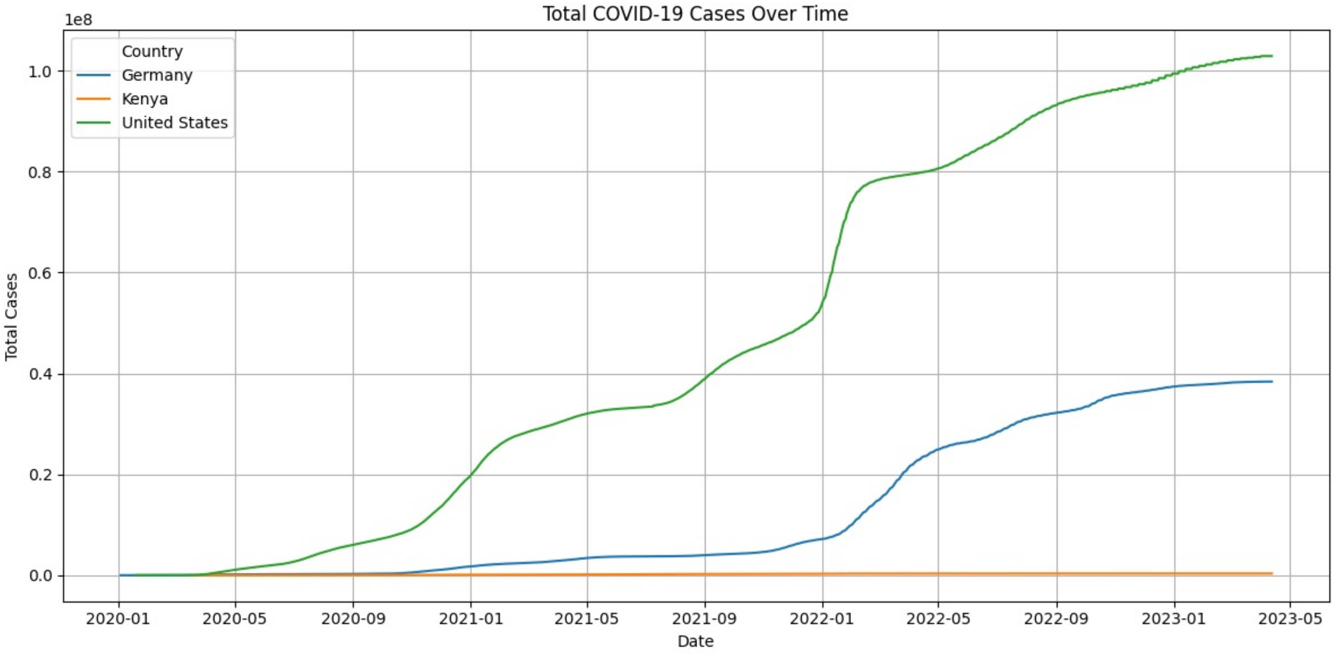
Out[24]:

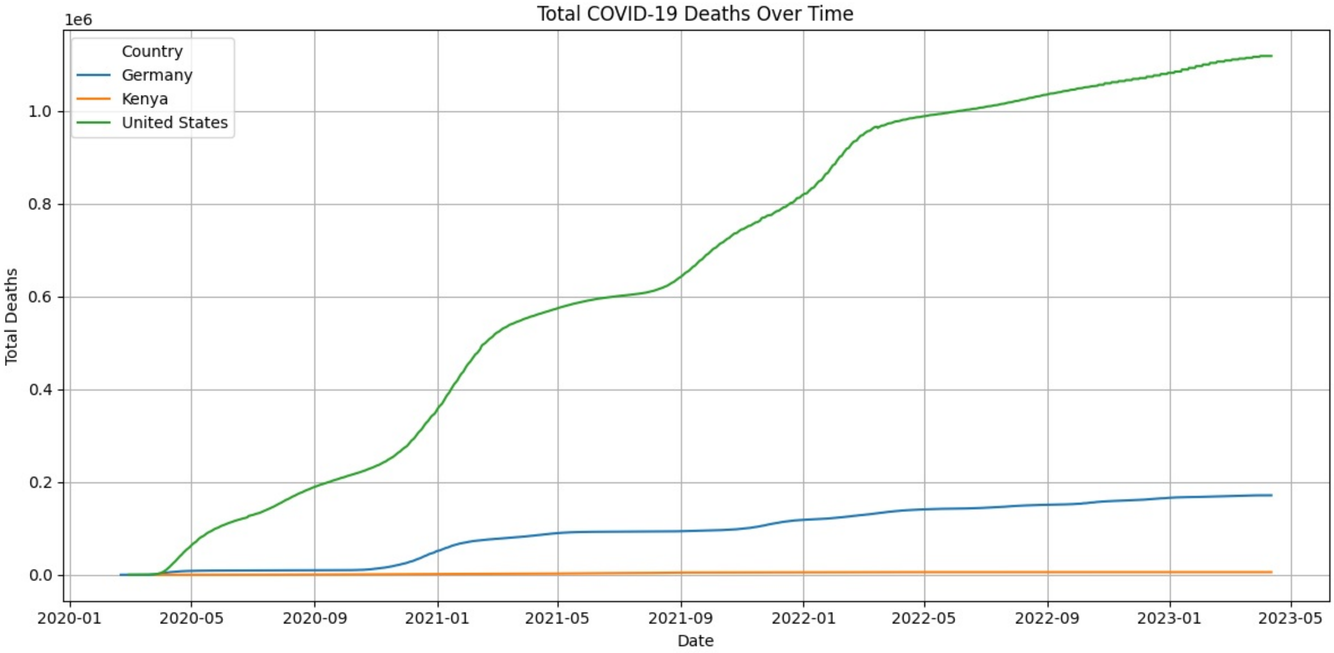
	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_death
0	DEU	Europe	Germany	2020-01-04	1.0	1.0	NaN	NaN	0.0	
1	DEU	Europe	Germany	2020-01-05	1.0	0.0	NaN	NaN	0.0	
2	DEU	Europe	Germany	2020-01-06	1.0	0.0	NaN	NaN	0.0	
3	DEU	Europe	Germany	2020-01-07	1.0	0.0	NaN	NaN	0.0	
4	DEU	Europe	Germany	2020-01-08	1.0	0.0	0.143	NaN	0.0	

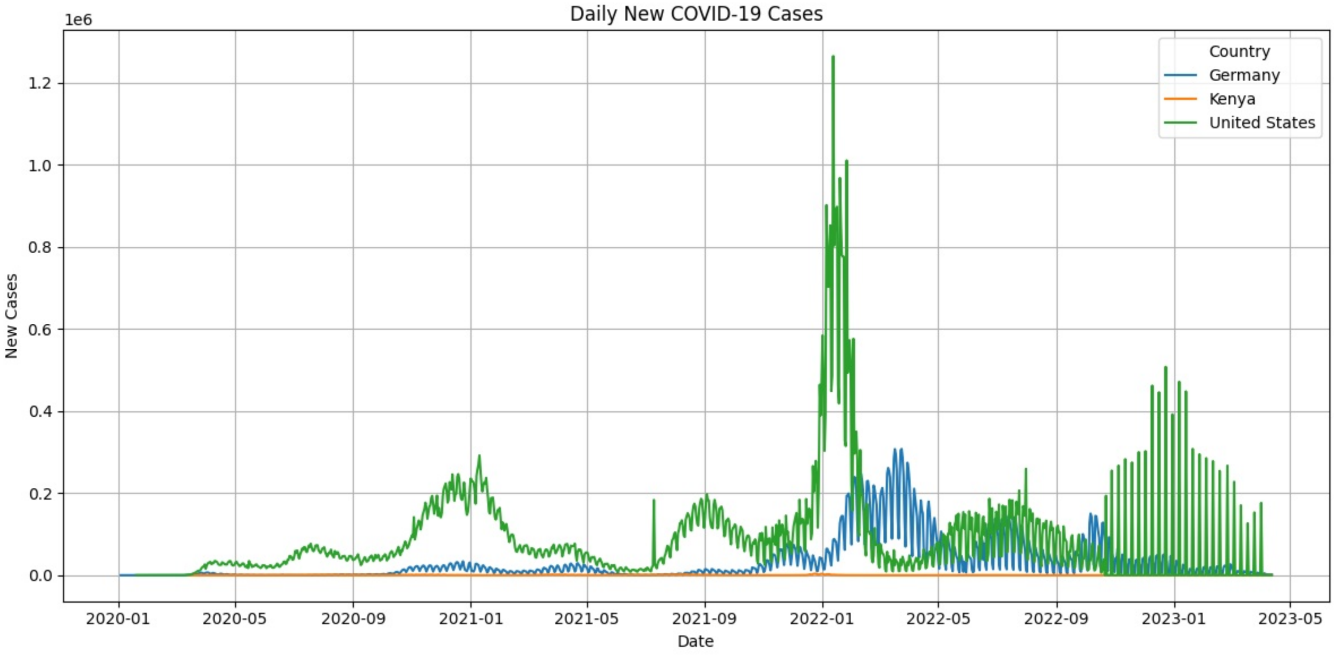
5 rows × 68 columns

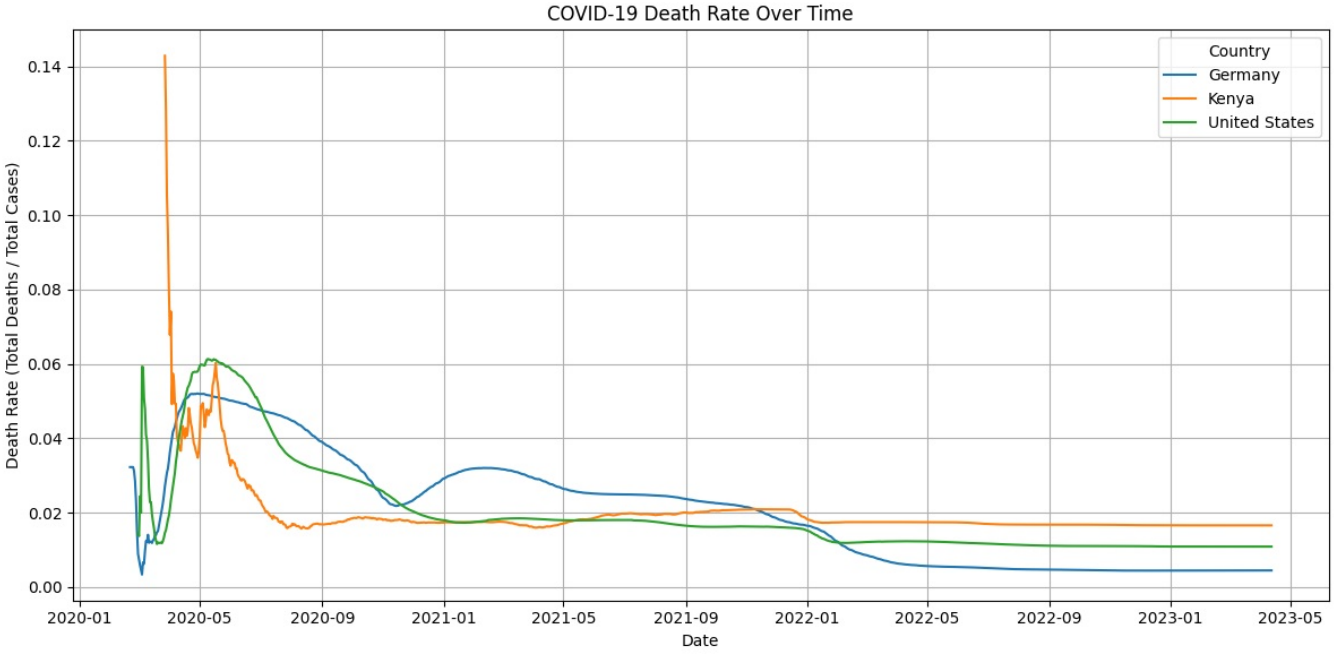
In [25]:

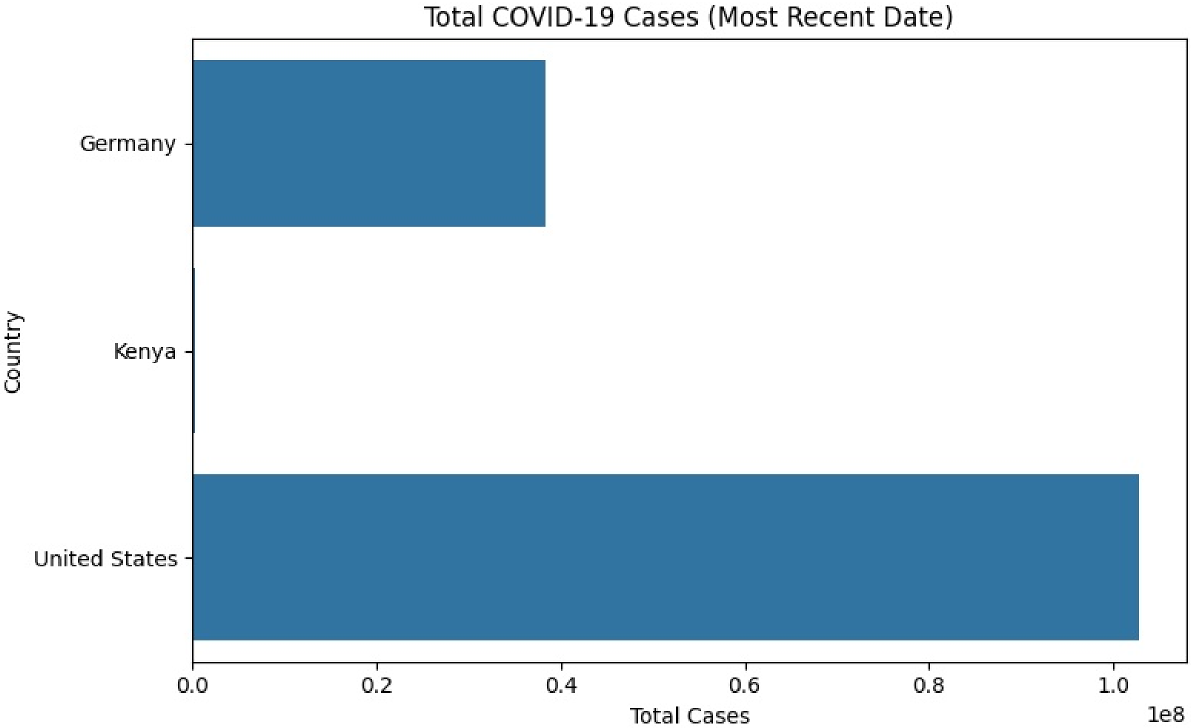
```python
# Exploratory Data Analysis (EDA)

import matplotlib.pyplot as plt
import seaborn as sns

# Line Chart (Cases Over Time)
try:
    plt.figure(figsize=(12, 6))
    sns.lineplot(data=df_filtered, x='date', y='total_cases', hue='location')
    plt.title('Total COVID-19 Cases Over Time')
    plt.xlabel('Date')
    plt.ylabel('Total Cases')
    plt.legend(title='Country')
    plt.grid(True)
    plt.tight_layout()
    plt.show()
except Exception as e:
    print('Error plotting line chart: {e}')

# Line Chart (Deaths Over Time)
try:
    plt.figure(figsize=(12, 6))
    sns.lineplot(data=df_filtered, x='date', y='total_deaths', hue='location')
    plt.title('Total COVID-19 Deaths Over Time')
    plt.xlabel('Date')
    plt.ylabel('Total Deaths')
    plt.legend(title='Country')
    plt.grid(True)
    plt.tight_layout()
    plt.show()
except Exception as e:
    print('Error plotting line chart: {e}')

# Line Chart (Daily New Cases)
try:
    plt.figure(figsize=(12, 6))
    sns.lineplot(data=df_filtered, x='date', y='new_cases', hue='location')
    plt.title('Daily New COVID-19 Cases')
    plt.xlabel('Date')
    plt.ylabel('New Cases')
    plt.legend(title='Country')
    plt.grid(True)
    plt.tight_layout()
    plt.show()
except Exception as e:
    print('Error plotting line chart: {e}')

# Line Chart (Death Rate)
try:
    df_filtered['death_rate'] = df_filtered['total_deaths'] / df_filtered['total_cases']
    plt.figure(figsize=(12, 6))
    sns.lineplot(data=df_filtered, x='date', y='death_rate', hue='location')
    plt.title('COVID-19 Death Rate Over Time')
    plt.xlabel('Date')
    plt.ylabel('Death Rate (Total Deaths / Total Cases)')
    plt.legend(title='Country')
    plt.grid(True)
    plt.tight_layout()
    plt.show()
except Exception as e:
    print('Error plotting line chart: {e}')

# Bar Chart (Top Country by Latest Total Cases)
try:
    latest_data = df_filtered.sort_values('date').groupby('location').tail(1)
    plt.figure(figsize=(8, 5))
    sns.barplot(data=latest_data, x='total_cases', y='location')
    plt.title('Total COVID-19 Cases (Most Recent Date)')
    plt.xlabel('Total Cases')
    plt.ylabel('Country')
    plt.tight_layout()
    plt.show()
except Exception as e:
    print('Error plotting bar chart: {e}')

# Heatmap (Correlation Analysis)
try:
    plt.figure(figsize=(10, 6))
    corr = df_filtered[['total_cases', 'new_cases', 'total_deaths', 'new_deaths', 'death_rate']].corr()
    sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f')
    plt.title('Correlation Matrix')
    plt.tight_layout()
    plt.show()
except Exception as e:
    print('Error plotting heatmap: {e}')
```

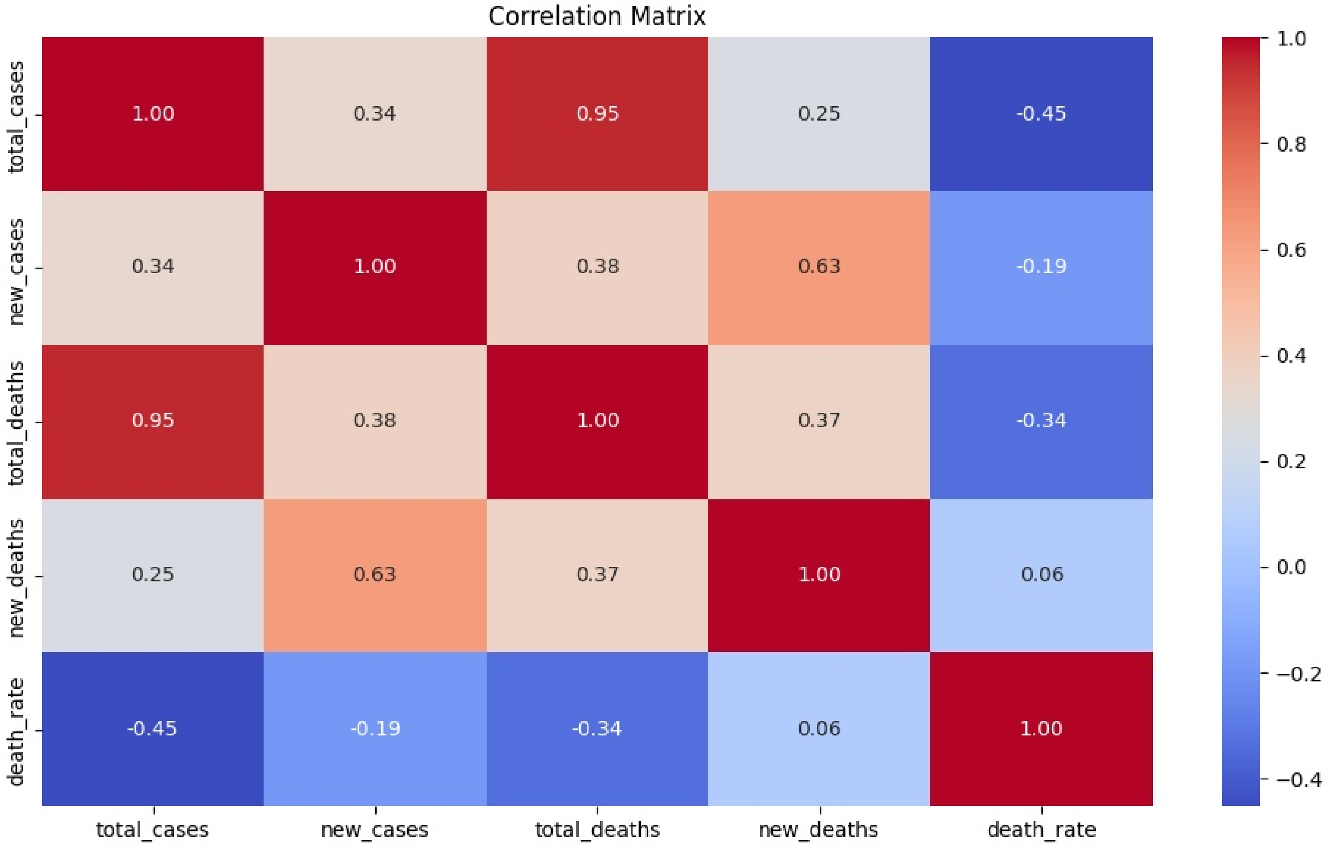





Daily New COVID-19 Cases
1e6
New Cases
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Date
2020-01
2020-05
2020-09
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
Country
Germany
Kenya
United States


COVID-19 Death Rate Over Time
Death Rate (Total Deaths / Total Cases)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
Date
2020-01
2020-05
2020-09
2021-01
2021-05
2021-09
2022-01
2022-05
2022-09
2023-01
2023-05
Country
Germany
Kenya
United States

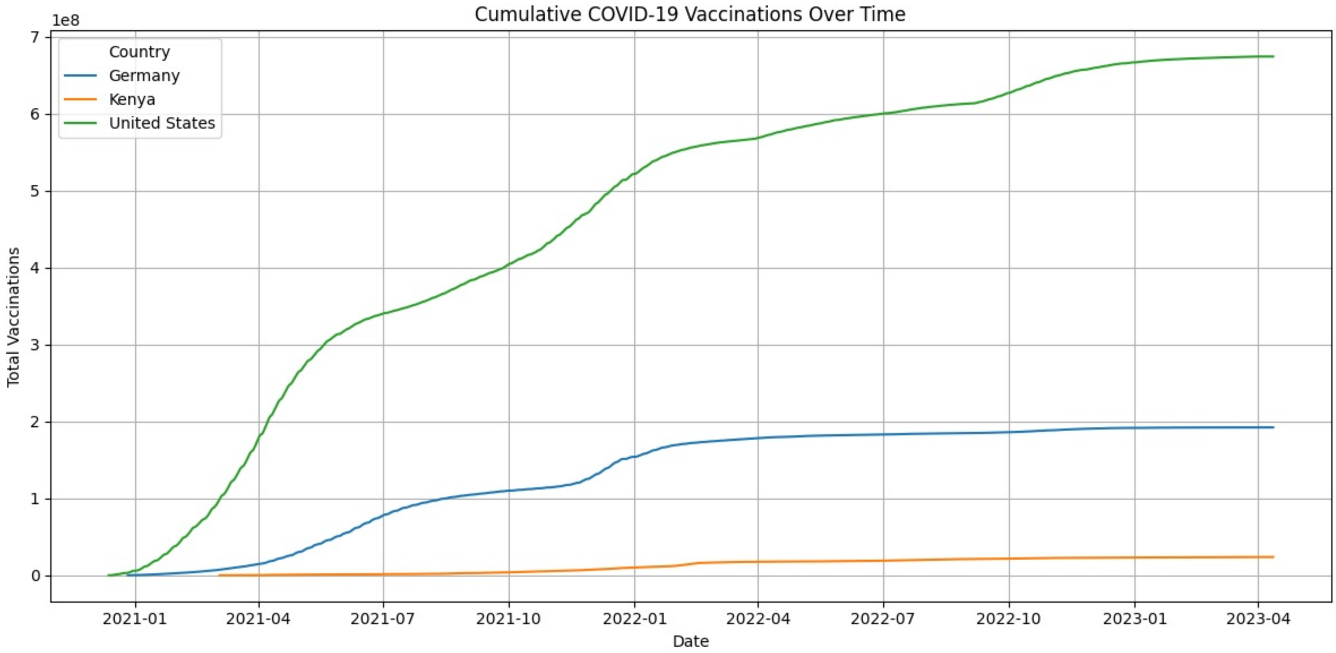


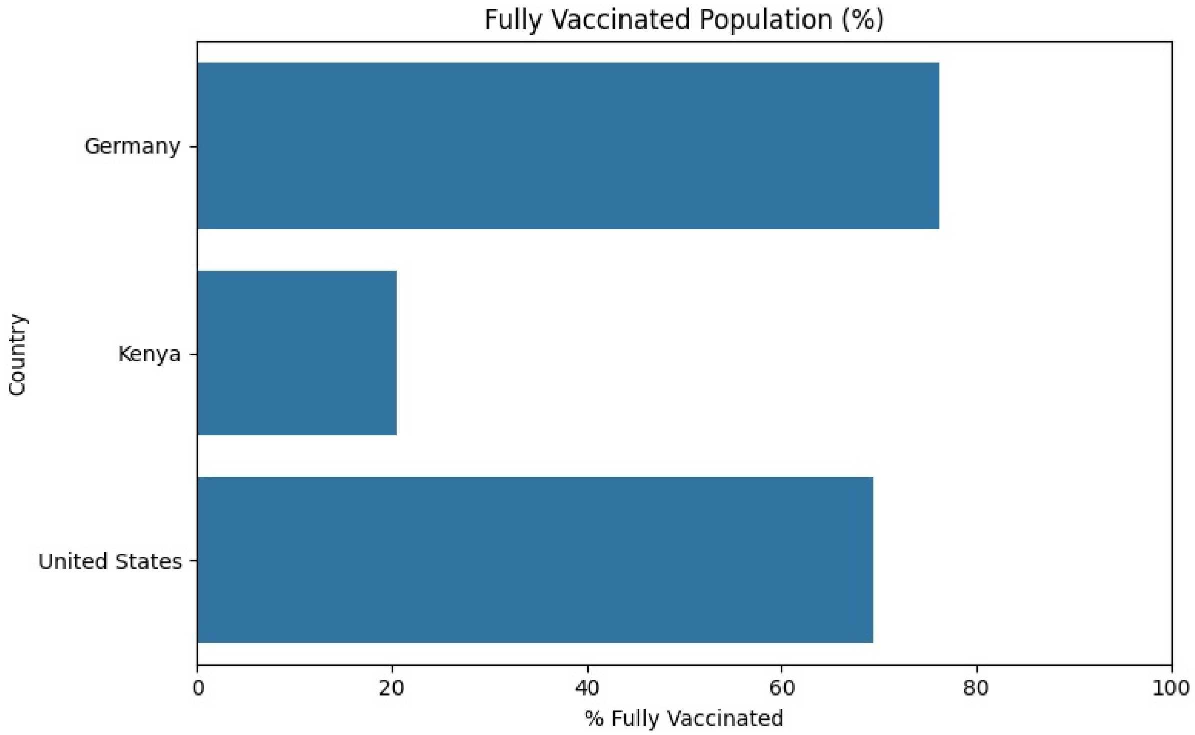


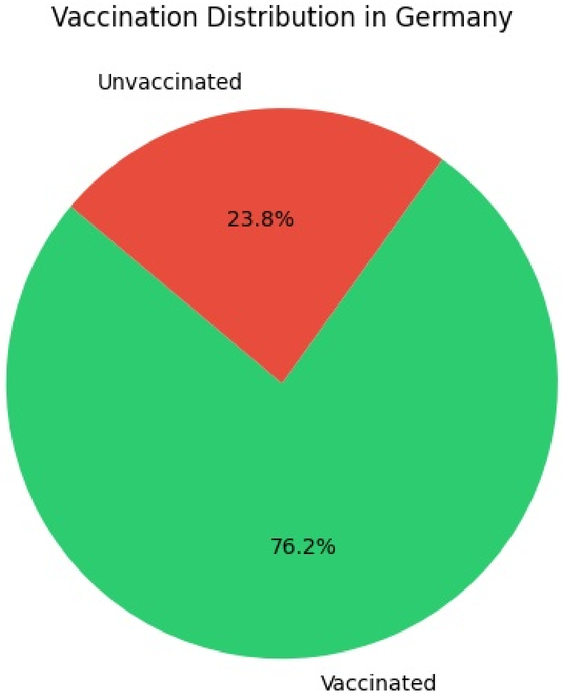
In [27]:

```
# Visualizing Vaccination Progress

import matplotlib.pyplot as plt
import seaborn as sns
```

```python
# Line Chart (Total Vaccinations)
try:
    plt.figure(figsize=(12, 6))
    sns.lineplot(data=df_filtered, x='date', y='total_vaccinations', hue='location')
    plt.title('Cumulative COVID-19 Vaccinations Over Time')
    plt.xlabel('Date')
    plt.ylabel('Total Vaccinations')
    plt.legend(title='Country')
    plt.grid(True)
    plt.tight_layout()
    plt.show()
except Exception as e:
    print('Error plotting line chart: {e}')

# Bar Chart (Vaccinated Population)
try:
    latest_vax = df_filtered.sort_values('date').groupby('location').tail(1)

    plt.figure(figsize=(8, 5))
    sns.barplot(data=latest_vax, x='people_fully_vaccinated_per_hundred', y='location')
    plt.title('Fully Vaccinated Population (%)')
    plt.xlabel('% Fully Vaccinated')
    plt.ylabel('Country')
    plt.xlim(0, 100)
    plt.tight_layout()
    plt.show()
except Exception as e:
    print('Error plotting bar chart: {e}')

# Pie Chart (Vaccinated vs Unvaccinated)
try:
    country = 'Germany'
    latest = df_filtered[df_filtered['location'] == country].sort_values('date').iloc[-1]
    vaccinated = latest['people_fully_vaccinated_per_hundred']
    unvaccinated = 100 - vaccinated

    plt.figure(figsize=(5, 5))
    plt.pie([vaccinated, unvaccinated],
            labels=['Vaccinated', 'Unvaccinated'],
            autopct='%1.1f%%',
            colors=['#2ecc71', '#e74c3c'],
            startangle=140)
    plt.title(f'Vaccination Distribution in {country}')
    plt.tight_layout()
    plt.show()
except Exception as e:
    print('Error plotting pie chart: {e}')
```







In [31]:

```
# Choropleth Map

import plotly.express as px

# latest data
latest_df = df.sort_values('date').groupby('location').tail(1)

# get only the neccessary column
latest_df = latest_df[latest_df['iso_code'].str.len() == 3]

# Choropleth
try:
    fig = px.choropleth(latest_df,
                    locations='iso_code',
                    color='total_cases',
                    hover_name='location',
                    color_continuous_scale='Reds',
                    title='Total COVID-19 Cases by Country',
                    labels={'total_cases': 'Total Cases'})
    fig.update_layout(geo=dict(showframe=False, showcoastlines=False))
    fig.show()
except Exception as e:
    print('Error drawing map: {e}')
```

Analysis

1. As of the latest available date, the United States reported the highest total number of COVID-19 cases among the selected countries, significantly surpassing Germany and Kenya.
2. While all countries experienced multiple waves of infections, the U.S. had the most pronounced peaks, whereas Germany showed more controlled fluctuations. Kenya showed lower overall case numbers but with sharp localized spikes.
3. Germany led in vaccination efforts, with over 70% of its population fully vaccinated, compared to 60% in the U.S. and 20% in Kenya. Kenya’s vaccination rollout lagged behind due to limited supply and distribution infrastructure.
4. Despite high case numbers, Germany maintained a relatively low death rate, possibly due to effective healthcare capacity. In contrast, the death rate in Kenya was more variable, particularly during surges when medical resources were strained.