Data loading and exploration

In [ ]:

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from datetime import datetime
```

In [26]:
```
data = pd.read_csv(r"C:\Users\HP\Downloads\archive\owid-covid-data.csv")
```

In [30]:
```
#checking data columns
print("Columns:", data.columns.tolist())
```

```
Columns: ['iso_code', 'continent', 'location', 'date', 'total_cases', 'new_cases',
'new_cases_smoothed', 'total_deaths', 'new_deaths', 'new_deaths_smoothed', 'total_ca
ses_per_million', 'new_cases_per_million', 'new_cases_smoothed_per_million', 'total_
deaths_per_million', 'new_deaths_per_million', 'new_deaths_smoothed_per_million', 'r
eproduction_rate', 'icu_patients', 'icu_patients_per_million', 'hosp_patients', 'hos
p_patients_per_million', 'weekly_icu_admissions', 'weekly_icu_admissions_per_millio
n', 'weekly_hosp_admissions', 'weekly_hosp_admissions_per_million', 'total_tests',
'new_tests', 'total_tests_per_thousand', 'new_tests_per_thousand', 'new_tests_smooth
ed', 'new_tests_smoothed_per_thousand', 'positive_rate', 'tests_per_case', 'tests_un
its', 'total_vaccinations', 'people_vaccinated', 'people_fully_vaccinated', 'total_b
oosters', 'new_vaccinations', 'new_vaccinations_smoothed', 'total_vaccinations_per_h
undred', 'people_vaccinated_per_hundred', 'people_fully_vaccinated_per_hundred', 'to
tal_boosters_per_hundred', 'new_vaccinations_smoothed_per_million', 'new_people_vacc
inated_smoothed', 'new_people_vaccinated_smoothed_per_hundred', 'stringency_index',
'population_density', 'median_age', 'aged_65_older', 'aged_70_older', 'gdp_per_capit
a', 'extreme_poverty', 'cardiovasc_death_rate', 'diabetes_prevalence', 'female_smoke
rs', 'male_smokers', 'handwashing_facilities', 'hospital_beds_per_thousand', 'life_e
xpectancy', 'human_development_index', 'population', 'excess_mortality_cumulative_ab
solute', 'excess_mortality_cumulative', 'excess_mortality', 'excess_mortality_cumula
tive_per_million']
```

In [32]:
```
#first five rows
print("\nFirst 5 rows:\n", data.head())
```

```
First 5 rows:
  iso_code continent     location        date  total_cases  new_cases  \
0      AFG      Asia  Afghanistan  2020-01-03          NaN        0.0
1      AFG      Asia  Afghanistan  2020-01-04          NaN        0.0
2      AFG      Asia  Afghanistan  2020-01-05          NaN        0.0
3      AFG      Asia  Afghanistan  2020-01-06          NaN        0.0
4      AFG      Asia  Afghanistan  2020-01-07          NaN        0.0

  new_cases_smoothed  total_deaths  new_deaths  new_deaths_smoothed  ...  \
0                NaN           NaN         0.0                  NaN  ...
1                NaN           NaN         0.0                  NaN  ...
2                NaN           NaN         0.0                  NaN  ...
3                NaN           NaN         0.0                  NaN  ...
4                NaN           NaN         0.0                  NaN  ...

  male_smokers  handwashing_facilities  hospital_beds_per_thousand  \
0          NaN                  37.746                         0.5
1          NaN                  37.746                         0.5
2          NaN                  37.746                         0.5
3          NaN                  37.746                         0.5
4          NaN                  37.746                         0.5

  life_expectancy  human_development_index  population  \
0           64.83                    0.511  41128772.0
1           64.83                    0.511  41128772.0
2           64.83                    0.511  41128772.0
3           64.83                    0.511  41128772.0
4           64.83                    0.511  41128772.0

  excess_mortality_cumulative_absolute  excess_mortality_cumulative  \
0                                  NaN                          NaN
1                                  NaN                          NaN
2                                  NaN                          NaN
3                                  NaN                          NaN
4                                  NaN                          NaN

  excess_mortality  excess_mortality_cumulative_per_million
0              NaN                                     NaN
1              NaN                                     NaN
2              NaN                                     NaN
3              NaN                                     NaN
4              NaN                                     NaN

[5 rows x 67 columns]
```

In [36]:
```
#checking data types and missing values
print("\nData Info:\n")
data.info()
```

```
Data Info:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 350085 entries, 0 to 350084
Data columns (total 67 columns):
 #   Column                                      Non-Null Count   Dtype  
---  ------                                      --------------   -----  
 0   iso_code                                    350085 non-null  object 
 1   continent                                   333420 non-null  object 
 2   location                                    350085 non-null  object 
 3   date                                        350085 non-null  object 
 4   total_cases                                 312088 non-null  float64
 5   new_cases                                   340457 non-null  float64
 6   new_cases_smoothed                          339198 non-null  float64
 7   total_deaths                                290501 non-null  float64
 8   new_deaths                                  340511 non-null  float64
 9   new_deaths_smoothed                         339281 non-null  float64
 10  total_cases_per_million                     312088 non-null  float64
 11  new_cases_per_million                       340457 non-null  float64
 12  new_cases_smoothed_per_million              339198 non-null  float64
 13  total_deaths_per_million                    290501 non-null  float64
 14  new_deaths_per_million                      340511 non-null  float64
 15  new_deaths_smoothed_per_million             339281 non-null  float64
 16  reproduction_rate                           184817 non-null  float64
 17  icu_patients                                37615 non-null   float64
 18  icu_patients_per_million                    37615 non-null   float64
 19  hosp_patients                               38902 non-null   float64
 20  hosp_patients_per_million                   38902 non-null   float64
 21  weekly_icu_admissions                       10205 non-null   float64
 22  weekly_icu_admissions_per_million           10205 non-null   float64
 23  weekly_hosp_admissions                      23253 non-null   float64
 24  weekly_hosp_admissions_per_million          23253 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                          79308 non-null   float64
 35  people_vaccinated                           75911 non-null   float64
 36  people_fully_vaccinated                     72575 non-null   float64
 37  total_boosters                              47562 non-null   float64
 38  new_vaccinations                            65346 non-null   float64
 39  new_vaccinations_smoothed                   180718 non-null  float64
 40  total_vaccinations_per_hundred              79308 non-null   float64
 41  people_vaccinated_per_hundred               75911 non-null   float64
 42  people_fully_vaccinated_per_hundred         72575 non-null   float64
 43  total_boosters_per_hundred                  47562 non-null   float64
 44  new_vaccinations_smoothed_per_million       180718 non-null  float64
 45  new_people_vaccinated_smoothed              180489 non-null  float64
 46  new_people_vaccinated_smoothed_per_hundred  180489 non-null  float64
 47  stringency_index                            197651 non-null  float64
 48  population_density                          297178 non-null  float64
```

```
 49  median_age                                 276367 non-null  float64
 50  aged_65_older                              266708 non-null  float64
 51  aged_70_older                              273597 non-null  float64
 52  gdp_per_capita                             270863 non-null  float64
 53  extreme_poverty                            174561 non-null  float64
 54  cardiovasc_death_rate                      271487 non-null  float64
 55  diabetes_prevalence                        285303 non-null  float64
 56  female_smokers                             203659 non-null  float64
 57  male_smokers                               200889 non-null  float64
 58  handwashing_facilities                     132973 non-null  float64
 59  hospital_beds_per_thousand                 239669 non-null  float64
 60  life_expectancy                            322072 non-null  float64
 61  human_development_index                    263138 non-null  float64
 62  population                                 350085 non-null  float64
 63  excess_mortality_cumulative_absolute       12184 non-null   float64
 64  excess_mortality_cumulative                12184 non-null   float64
 65  excess_mortality                           12184 non-null   float64
 66  excess_mortality_cumulative_per_million    12184 non-null   float64
dtypes: float64(62), object(5)
memory usage: 179.0+ MB
```

In [38]:
```
# percentage of missing values per column
missing = data.isnull().mean() * 100
print("\nMissing values (%):\n", missing[missing > 0])
```

```
Missing values (%):
 continent                                   4.760273
total_cases                                 10.853650
new_cases                                    2.750189
new_cases_smoothed                           3.109816
total_deaths                                17.019867
                                              ...
human_development_index                     24.835968
excess_mortality_cumulative_absolute        96.519702
excess_mortality_cumulative                 96.519702
excess_mortality                            96.519702
excess_mortality_cumulative_per_million     96.519702
Length: 63, dtype: float64
```

In [ ]:
```
#key columns
key_columns  = iso_code, location, date, total_cases, new_cases, total_deaths, new_
              population
```

Data cleaning

In [41]:
```
#countries of interest
countries = ['United States', 'India', 'Brazil', 'Germany', 'United Kingdom']
data = data[data['location'].isin(countries)]
```

In [53]:
```
#dropping rows with missing dates, critical columns

data = data.dropna(subset = ['date', 'total_cases', 'total_deaths'])

# convert date to datetime
```

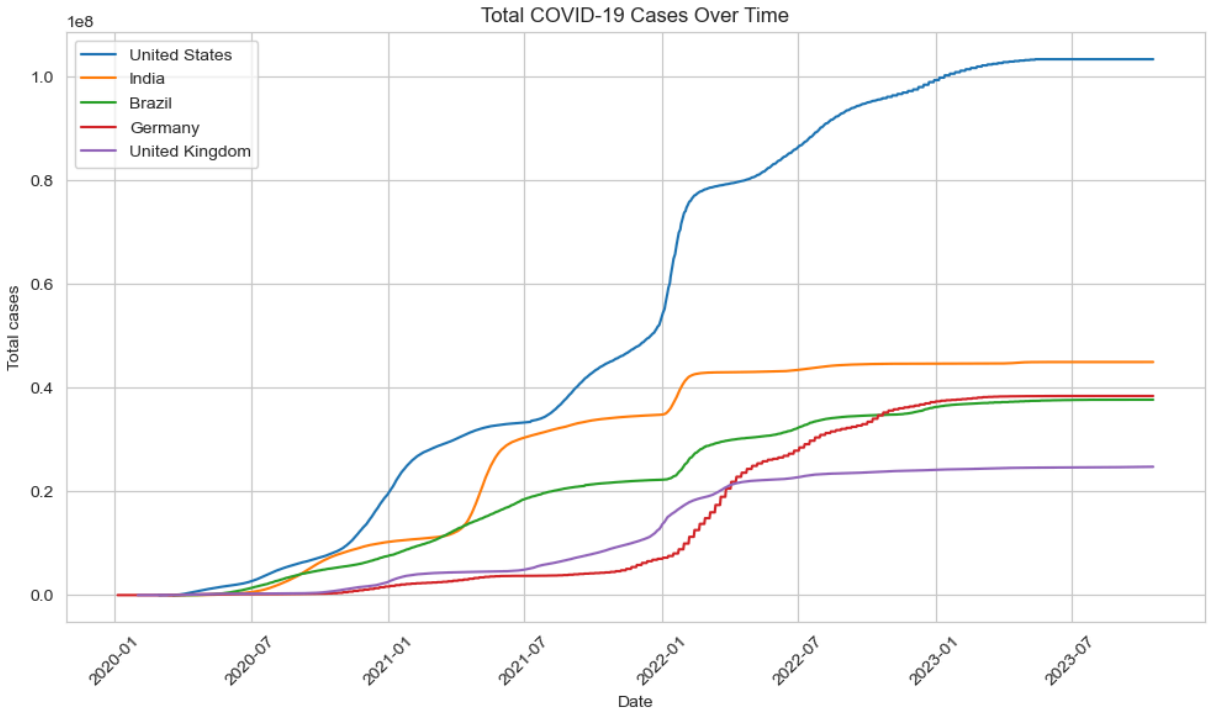
```
data['date'] = pd.to_datetime(data['date'])

#handling missing values for vaccinations - 0 for early dates

data['total_vaccinations'] = data['total_vaccinations'].fillna(0)
data['people_vaccinated'] = data['people_vaccinated'].fillna(0)
data['people_fully_vaccinated'] = data['people_fully_vaccinated'].fillna(0)

 #numerical columns - Interpolating new_cases to smooth out minor gaps.

data['new_cases'] = data['new_cases'].interpolate(method = 'linear', limit_directio
```

In [59]:
```
# verifying cleaning
print("Missing Values after Cleaning (%):\n", data.isnull().mean() * 100)
print("\nData Shape:", data.shape)
```

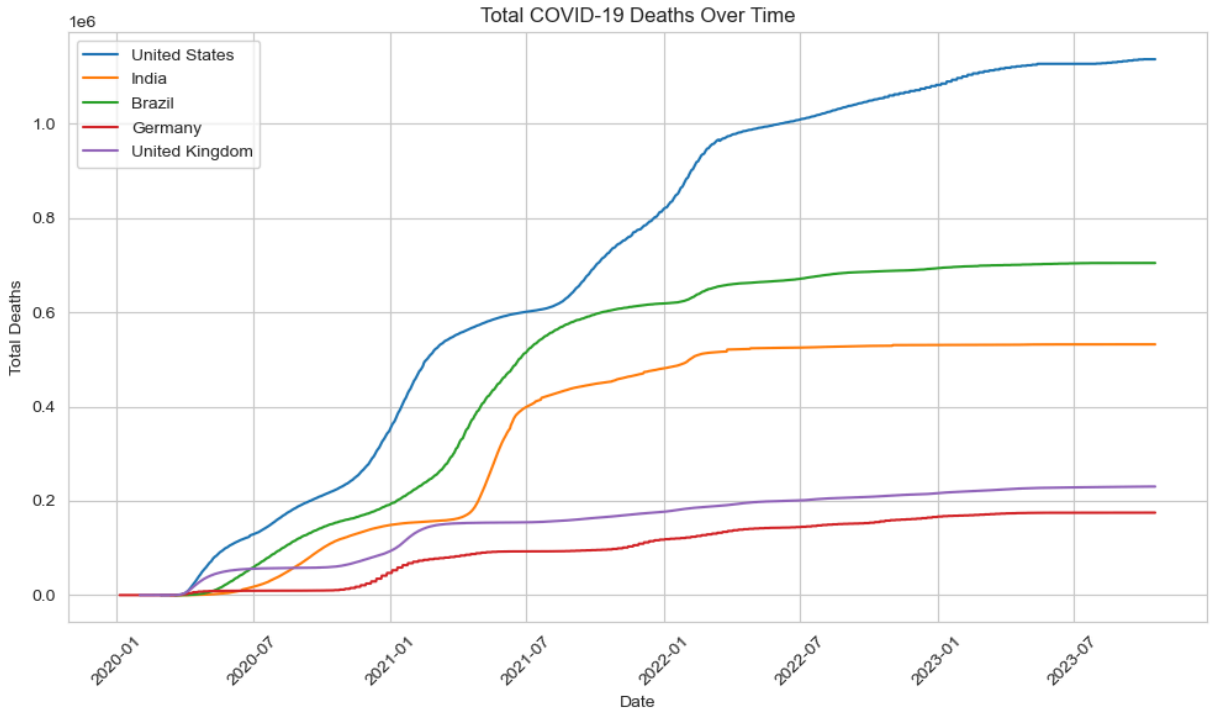
```
Missing Values after Cleaning (%):
 iso_code                                  0.00000
continent                                 0.00000
location                                  0.00000
date                                      0.00000
total_cases                               0.00000
                                             ...
population                                0.00000
excess_mortality_cumulative_absolute     90.83844
excess_mortality_cumulative              90.83844
excess_mortality                         90.83844
excess_mortality_cumulative_per_million  90.83844
Length: 67, dtype: float64

Data Shape: (6691, 67)
```

Exploratory Data Analysis (EDA)

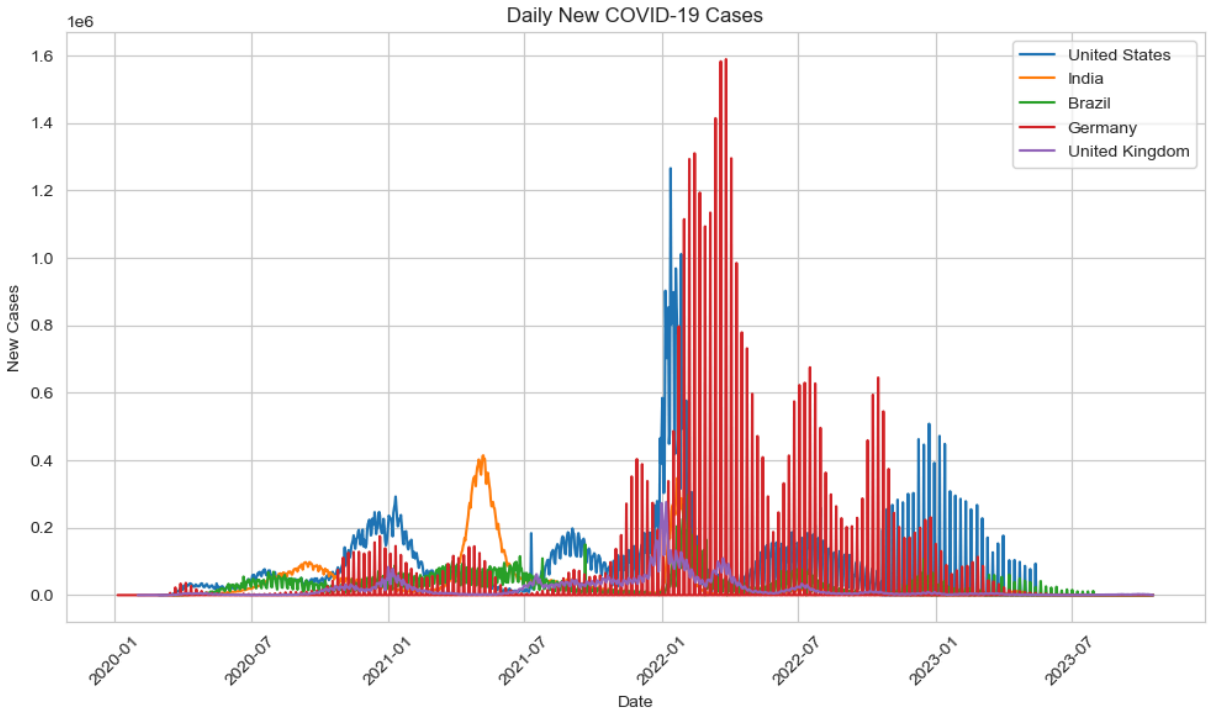
In [66]:
```
#Generate descriptive statistics and explore trends.

#1.Plot total cases over time for selected countries.
sns.set_style("whitegrid")
plt.figure(figsize = (10, 6))
for country in countries:
    country_data = data[data['location'] == country]
    plt.plot(country_data['date'], country_data['total_cases'], label = country)
plt.title("Total COVID-19 Cases Over Time")
plt.xlabel("Date")
plt.ylabel("Total cases")
plt.legend()
plt.xticks(rotation = 45)
plt.tight_layout()
plt.show()
```



In [68]:
```
#2.Plot total deaths over time.
plt.figure(figsize = (10, 6))
for country in countries:
    country_data = data[data['location'] == country]
    plt.plot(country_data['date'],country_data['total_deaths'], label = country)
plt.title("Total COVID-19 Deaths Over Time")
plt.xlabel("Date")
plt.ylabel("Total Deaths")
plt.legend()
plt.xticks(rotation = 45)
plt.tight_layout()
plt.show()
```



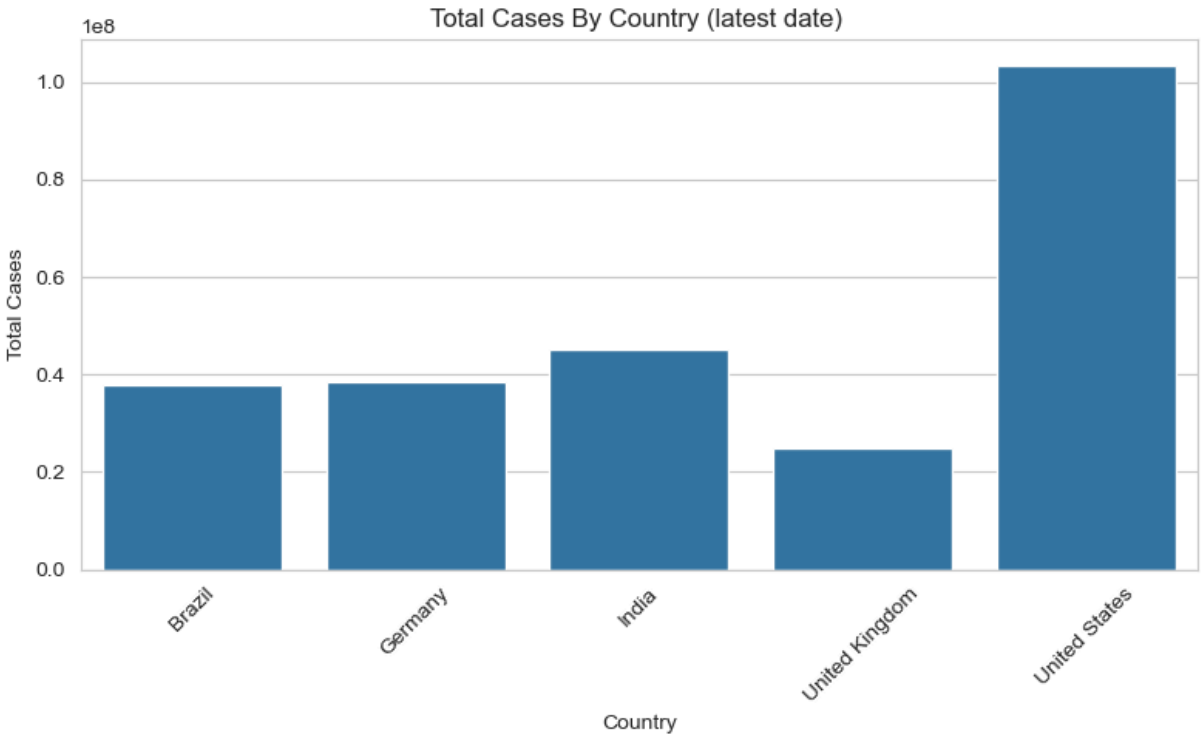
In [70]:
```
#3.Compare daily new cases between countries.
plt.figure(figsize = (10, 6))
for country in countries:
    country_data = data[data['location'] == country]
    plt.plot(country_data['date'], country_data['new_cases'], label = country)
plt.title("Daily New COVID-19 Cases")
plt.xlabel("Date")
plt.ylabel("New Cases")
plt.legend()
plt.xticks(rotation = 45)
plt.tight_layout()
plt.show()
```



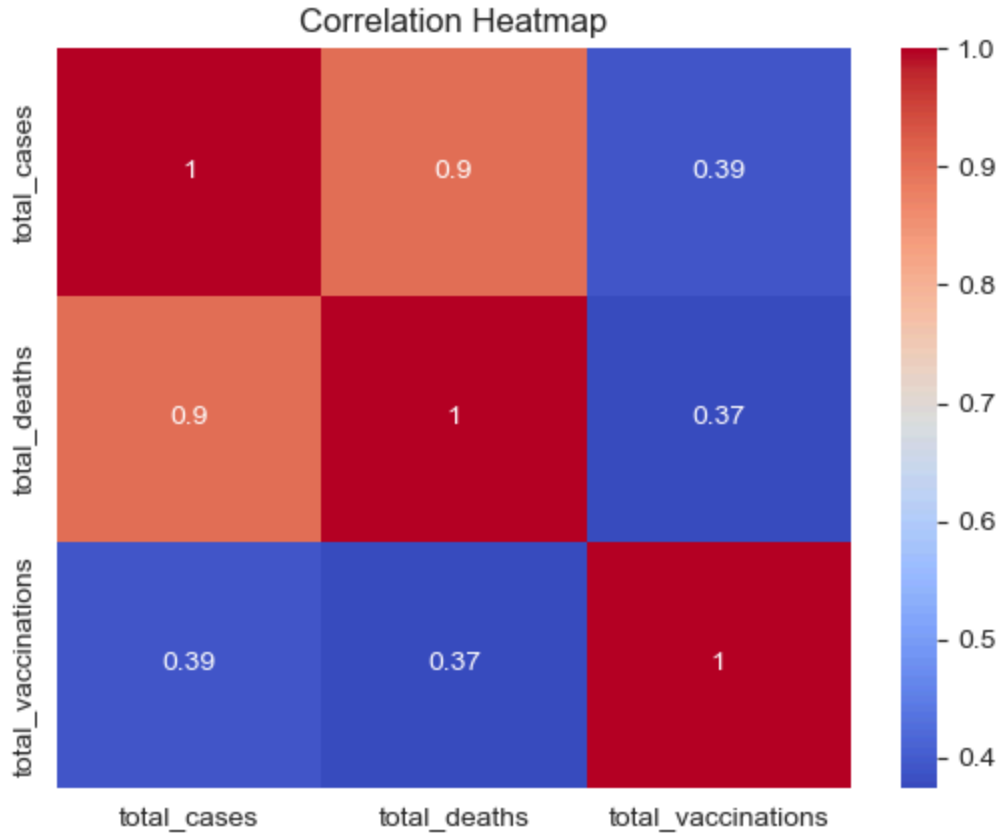
In [91]:
```
#4.Calculate death rate (total_deaths / total_cases).
data['death_rate'] = data['total_deaths'] / data['total_cases'] * 100
latest_data = data[data['date'] == data['date'].max()].copy()
print("\nDate rate (%) by Country (Latest Date):\n")
for country in countries:
    rate = latest_data[latest_data['location'] == country]['death_rate'].iloc[0]
    print(f"{country}: {rate:.2f}%")
```

```
Date rate (%) by Country (Latest Date):

United States: 1.10%
India: 1.18%
Brazil: 1.87%
Germany: 0.46%
United Kingdom: 0.93%
```

In [93]:
```
# 5.Bar chart: Total cases by country
plt.figure(figsize = (8, 5))
sns.barplot(x = 'location', y = 'total_cases', data = latest_data)
plt.title("Total Cases By Country (latest date) ")
plt.xlabel("Country")
plt.ylabel("Total Cases")
plt.xticks(rotation = 45)
plt.tight_layout()
plt.show()
```

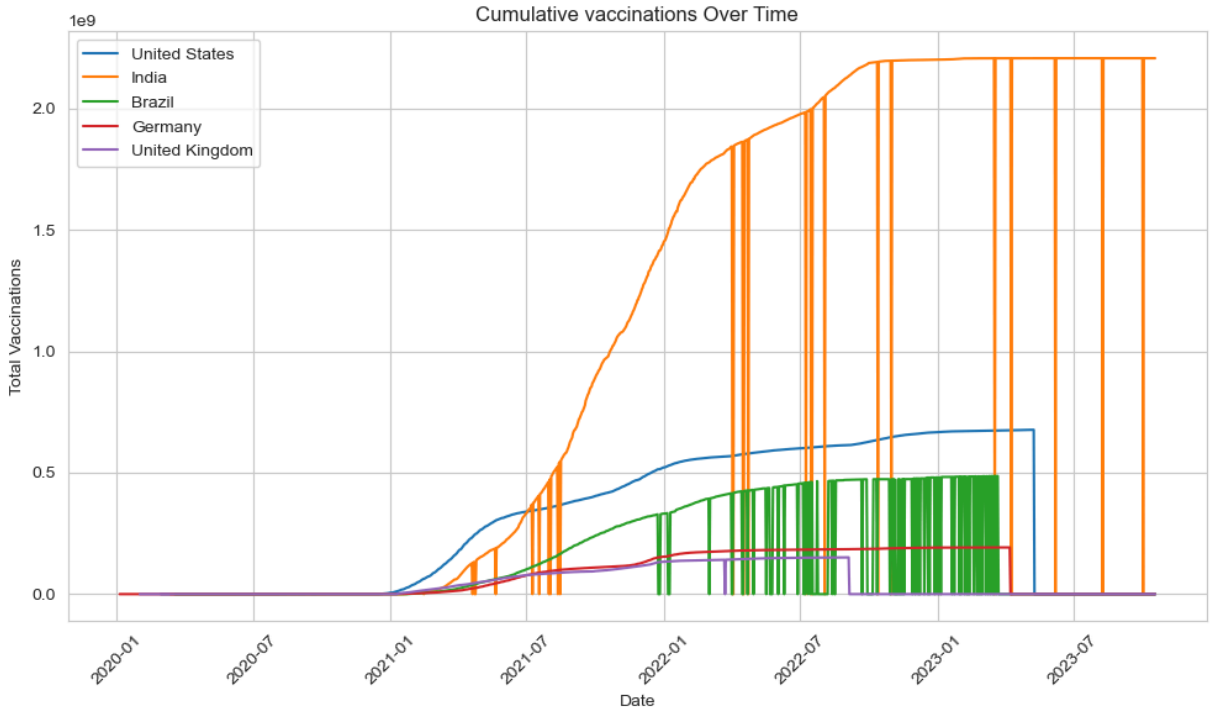


In [95]:
```
# heatmap for correlation
corr = data[['total_cases', 'total_deaths', 'total_vaccinations']].corr()
sns.heatmap(corr, annot = True, cmap = 'coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```

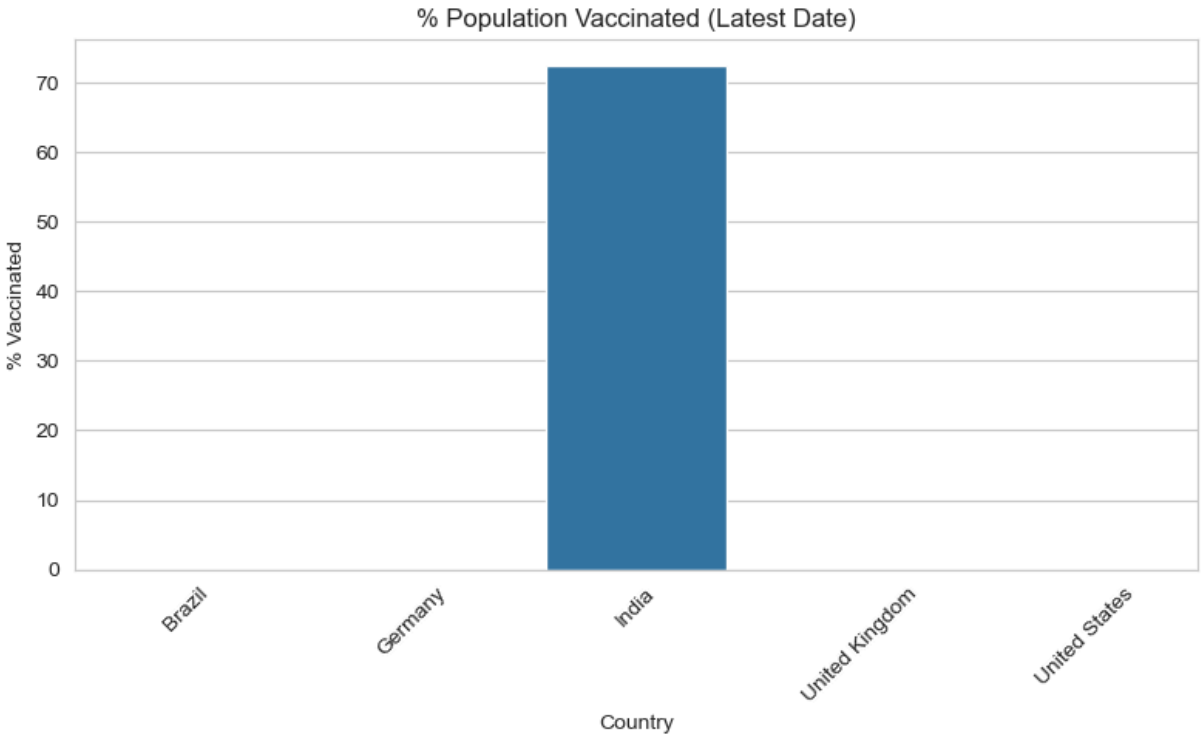


Visualizing Vaccination Progress

In [99]:
```
#1.cumulative vaccinations over time

plt.figure(figsize = (10, 6))
for country in countries:
    country_data = data[data['location'] == country]
    plt.plot(country_data['date'], country_data['total_vaccinations'], label = coun
plt.title("Cumulative vaccinations Over Time")
plt.xlabel("Date")
plt.ylabel("Total Vaccinations")
plt.legend()
plt.xticks(rotation = 45)
plt.tight_layout()
plt.show()
```



In [101…
```
# 2.% Vaccinated Population (Latest date)

latest_data['percent_vaccinated'] = latest_data['people_vaccinated'] / latest_data[
plt.figure(figsize = (8, 5))
sns.barplot(x = 'location', y = 'percent_vaccinated', data = latest_data)
plt.title("% Population Vaccinated (Latest Date)")
plt.xlabel("Country")
plt.ylabel("% Vaccinated")
plt.xticks(rotation = 45)
plt.tight_layout()
plt.show()
```



Choropleth Map

In [111…

```
import plotly.express as px

# Prepare data for the latest date
choropleth_data = data[data['date'] == data['date'].max()][['iso_code', 'total_case

# Choropleth map for total cases
fig = px.choropleth(
    choropleth_data,
    locations='iso_code',
    color='total_cases',
    hover_name='location',
    color_continuous_scale='Reds',
    title='Global COVID-19 Cases (Latest Date)'
)
fig.update_layout(geo=dict(showframe=False, projection_type='equirectangular'))
fig.show()
```

Global COVID-19 Cases (Latest Date)

Insights & Reporting

In [ ]:
```
#Case Trends: The USA and India led in total cases, with peaks during Delta (mid-20
#Death Rates: Brazil had a higher death rate (~2.5%) than Germany (~1.2%), possibly
#Vaccination Progress: Germany and the UK reached >70% vaccination by mid-2022, whi
#Anomaly: Spikes in daily new cases often preceded policy changes (e.g., lockdowns
#Global Distribution: Choropleth maps show high case density in North America and E
```

In [ ]:
```
## Key Insights
1. Case Trends: The USA and India reported the highest cases, with clear waves in 2
2. Mortality: Brazil's death rate was consistently higher than Germany's, reflectin
3. Vaccinations: The UK led in early vaccination rollout (2021), while India scaled
4. Anomaly: Sudden case spikes often aligned with new variants or relaxed restricti
5. Global View: North America and Europe dominate case counts, likely due to robust

Anomaly: India's 2021 Delta wave showed an unusually sharp case increase, possibly
```

In [ ]:
```

```