COVID-19 Global Data Tracker

Welcome to your COVID-19 data analysis project. In this notebook, we'll analyze trends in global COVID-19 data including cases, deaths, and vaccinations.

In [10]:
```
# Step 1: Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

# Configure plots
sns.set(style='whitegrid')
plt.rcParams['figure.figsize'] = (14, 7)
```

In [2]:
```
# Step 2: Load the COVID-19 dataset
df = pd.read_csv("owid-covid-data/owid-covid-data.csv")
df.head()
```

Out[2]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	tc
0	AFG	Asia	Afghanistan	2020-01-03	NaN	0.0	NaN	
1	AFG	Asia	Afghanistan	2020-01-04	NaN	0.0	NaN	
2	AFG	Asia	Afghanistan	2020-01-05	NaN	0.0	NaN	
3	AFG	Asia	Afghanistan	2020-01-06	NaN	0.0	NaN	
4	AFG	Asia	Afghanistan	2020-01-07	NaN	0.0	NaN	

5 rows × 67 columns

In [3]:
```
# 🔍 Step 3: Explore the dataset
print("\nColumns in the dataset:")
print(df.columns.tolist())

print("\nMissing values per column:")
print(df.isnull().sum())
```

```
Columns in the dataset:
['iso_code', 'continent', 'location', 'date', 'total_cases', 'new_cases', 'new_cases
_smoothed', 'total_deaths', 'new_deaths', 'new_deaths_smoothed', 'total_cases_per_mi
llion', 'new_cases_per_million', 'new_cases_smoothed_per_million', 'total_deaths_per
_million', 'new_deaths_per_million', 'new_deaths_smoothed_per_million', 'reproductio
n_rate', 'icu_patients', 'icu_patients_per_million', 'hosp_patients', 'hosp_patients
_per_million', 'weekly_icu_admissions', 'weekly_icu_admissions_per_million', 'weekly
_hosp_admissions', 'weekly_hosp_admissions_per_million', 'total_tests', 'new_tests',
'total_tests_per_thousand', 'new_tests_per_thousand', 'new_tests_smoothed', 'new_tes
ts_smoothed_per_thousand', 'positive_rate', 'tests_per_case', 'tests_units', 'total_
vaccinations', 'people_vaccinated', 'people_fully_vaccinated', 'total_boosters', 'ne
w_vaccinations', 'new_vaccinations_smoothed', 'total_vaccinations_per_hundred', 'peo
ple_vaccinated_per_hundred', 'people_fully_vaccinated_per_hundred', 'total_boosters_
per_hundred', 'new_vaccinations_smoothed_per_million', 'new_people_vaccinated_smooth
ed', 'new_people_vaccinated_smoothed_per_hundred', 'stringency_index', 'population_d
ensity', 'median_age', 'aged_65_older', 'aged_70_older', 'gdp_per_capita', 'extreme_
poverty', 'cardiovasc_death_rate', 'diabetes_prevalence', 'female_smokers', 'male_sm
okers', 'handwashing_facilities', 'hospital_beds_per_thousand', 'life_expectancy',
'human_development_index', 'population', 'excess_mortality_cumulative_absolute', 'ex
cess_mortality_cumulative', 'excess_mortality', 'excess_mortality_cumulative_per_mil
lion']

Missing values per column:
iso_code                                      0
continent                                 14352
location                                      0
date                                          0
total_cases                               35741
                                          ...
population                                    0
excess_mortality_cumulative_absolute     292217
excess_mortality_cumulative              292217
excess_mortality                         292217
excess_mortality_cumulative_per_million  292217
Length: 67, dtype: int64
```

In [3]:

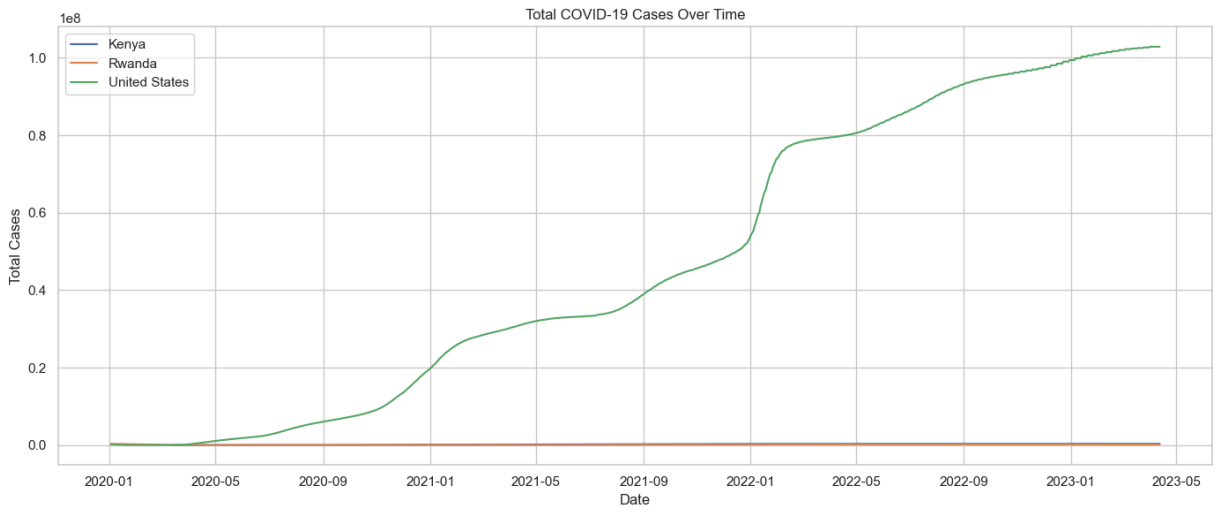
```python
# Step 4: Data Cleaning - Filter and Prepare

# Filter the data for Kenya, Rwanda, and USA
countries = ['Kenya', 'Rwanda', 'United States']
df_filtered = df[df['location'].isin(countries)].copy()

# Convert date column to datetime
df_filtered['date'] = pd.to_datetime(df_filtered['date'])

# Sort by location and date
df_filtered = df_filtered.sort_values(by=['location', 'date'])

# Interpolate missing values for key numeric columns
cols_to_interp = ['total_cases', 'total_deaths', 'total_vaccinations']
df_filtered[cols_to_interp] = df_filtered[cols_to_interp].interpolate(method='linea

# Preview cleaned data
df_filtered.head()
```

Out[3]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed
136231	KEN	Africa	Kenya	2020-01-03	NaN	0.0	NaN
136232	KEN	Africa	Kenya	2020-01-04	NaN	0.0	NaN
136233	KEN	Africa	Kenya	2020-01-05	NaN	0.0	NaN
136234	KEN	Africa	Kenya	2020-01-06	NaN	0.0	NaN
136235	KEN	Africa	Kenya	2020-01-07	NaN	0.0	NaN

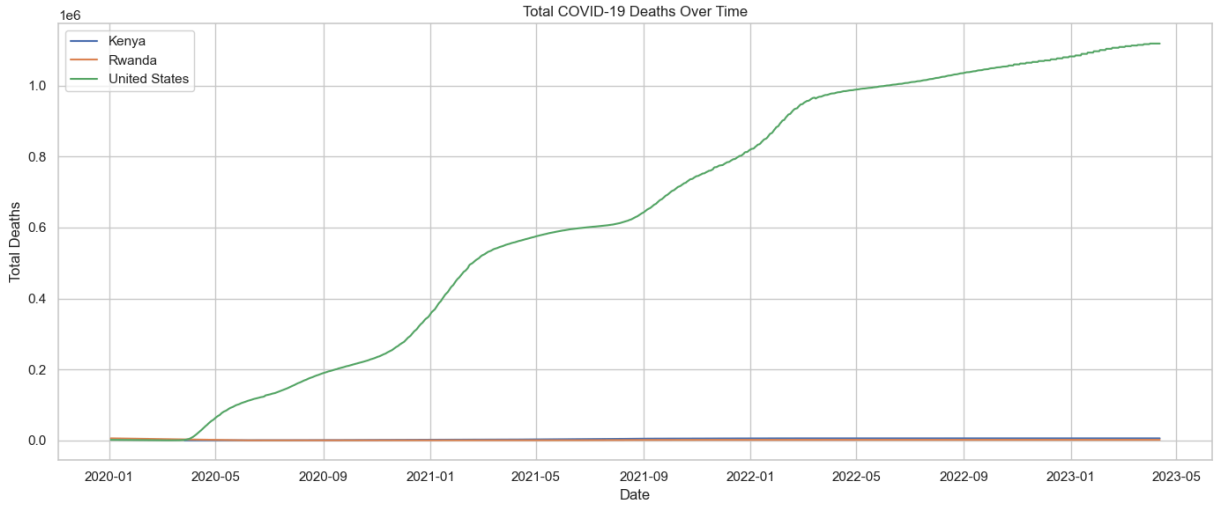
5 rows × 67 columns

In [4]:
```
# Total cases over time
plt.figure(figsize=(14, 6))
for country in countries:
    data = df_filtered[df_filtered['location'] == country]
    plt.plot(data['date'], data['total_cases'], label=country)
plt.title('Total COVID-19 Cases Over Time')
plt.xlabel('Date')
plt.ylabel('Total Cases')
plt.legend()
plt.tight_layout()
plt.show()
```

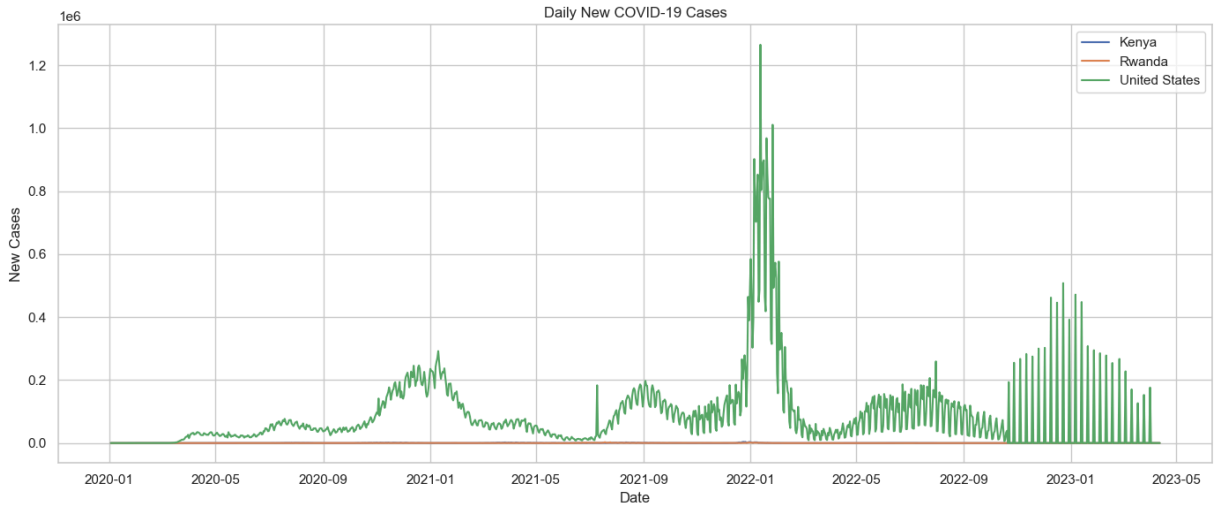


In [5]:
```
# Total deaths over time
plt.figure(figsize=(14, 6))
for country in countries:
    data = df_filtered[df_filtered['location'] == country]
    plt.plot(data['date'], data['total_deaths'], label=country)
plt.title('Total COVID-19 Deaths Over Time')
plt.xlabel('Date')
plt.ylabel('Total Deaths')
```

```
plt.legend()
plt.tight_layout()
plt.show()
```

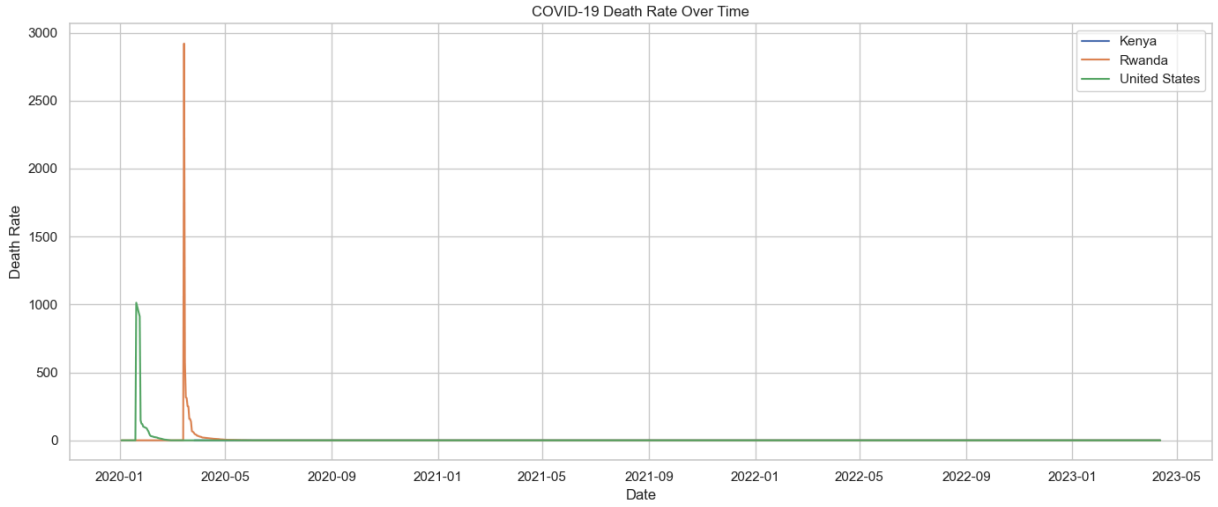


In [6]:
```
# Daily new cases comparison
plt.figure(figsize=(14, 6))
for country in countries:
    data = df_filtered[df_filtered['location'] == country]
    plt.plot(data['date'], data['new_cases'], label=country)
plt.title('Daily New COVID-19 Cases')
plt.xlabel('Date')
plt.ylabel('New Cases')
plt.legend()
plt.tight_layout()
plt.show()
```

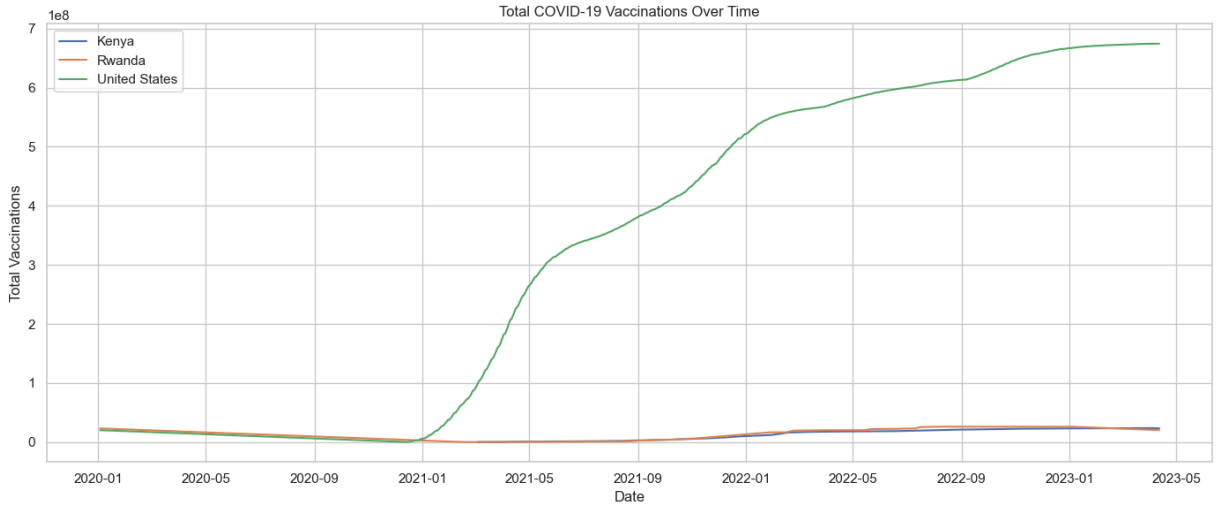


In [7]:
```
# Death rate: total_deaths / total_cases
df_filtered['death_rate'] = df_filtered['total_deaths'] / df_filtered['total_cases'

plt.figure(figsize=(14, 6))
for country in countries:
    data = df_filtered[df_filtered['location'] == country]
    plt.plot(data['date'], data['death_rate'], label=country)
plt.title('COVID-19 Death Rate Over Time')
```

```
plt.xlabel('Date')
plt.ylabel('Death Rate')
plt.legend()
plt.tight_layout()
plt.show()
```



In [8]:
```
# Total vaccinations over time
plt.figure(figsize=(14, 6))
for country in countries:
    data = df_filtered[df_filtered['location'] == country]
    plt.plot(data['date'], data['total_vaccinations'], label=country)
plt.title('Total COVID-19 Vaccinations Over Time')
plt.xlabel('Date')
plt.ylabel('Total Vaccinations')
plt.legend()
plt.tight_layout()
plt.show()
```



Choropleth Map – Total Cases by Country (Latest Date)

In [19]:
```
# 📌 Step 1: Get the latest available date in dataset
latest_date = df['date'].max()

# 📌 Step 2: Filter dataset for latest date
latest_df = df[df['date'] == latest_date]

# 📌 Step 3: Keep only necessary columns and drop NaNs
choropleth_df = latest_df[['iso_code', 'location', 'total_cases']].dropna()

# 📌 Step 4: Plot Choropleth Map
fig = px.choropleth(
    choropleth_df,
    locations='iso_code',
    color='total_cases',
    hover_name='location',
    color_continuous_scale='OrRd',
    title=f'Total COVID-19 Cases by Country on {latest_date}',
    labels={'total_cases': 'Total Cases'}
)
fig.update_layout(geo=dict(showframe=False, showcoastlines=False))
fig.show()
```

Insights Summary

1. Vaccine Rollout

- The United States shows the fastest and earliest vaccine rollout among the selected countries.
- Kenya and Rwanda started vaccinations later and at a slower pace, though Rwanda shows a consistent upward trend.

2. Case Trends

- The U.S. experienced multiple waves with very high daily new cases compared to Kenya and Rwanda.
- Rwanda managed to keep total cases significantly lower despite proximity to global hotspots.

3. Death Rate Patterns

- Kenya and Rwanda have lower death rates compared to the U.S.
- Possible factors: younger population, underreporting, or different testing policies.

4. Anomalies

- Data gaps or unexpected drops in daily cases or vaccinations were seen for Rwanda during certain periods — likely due to reporting delays.

- Some countries report "0" new cases on weekends — indicative of batch reporting.

Recommendations for Further Analysis

- Explore population-adjusted metrics (cases per million, vaccinations per 100 people).
- Compare testing rates and positivity ratios.
- Include socioeconomic or healthcare infrastructure data to interpret the results better.