covid-19dataset

May 14, 2025

[1]:
```
import pandas as pd
```

[2]:
```
import matplotlib.pyplot as plt
```

```
Matplotlib is building the font cache; this may take a moment.
```

[3]:
```
import seaborn as sns
```

[4]:
```
import plotly.express as px
```

[6]:
```
plt.style.use('seaborn-v0_8')
```

[7]:
```
sns.set_theme()
```

[14]:
```
df = pd.read_csv('owid-covid-data.csv')
```

[15]:
```
print("\nColumns:\n", df.columns)
```

```
Columns:
 Index(['iso_code', 'continent', 'location', 'date', 'total_cases', 'new_cases',
       'new_cases_smoothed', 'total_deaths', 'new_deaths',
       'new_deaths_smoothed', 'total_cases_per_million',
       'new_cases_per_million', 'new_cases_smoothed_per_million',
       'total_deaths_per_million', 'new_deaths_per_million',
       'new_deaths_smoothed_per_million', 'reproduction_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_tests_smoothed_per_thousand',
       'positive_rate', 'tests_per_case', 'tests_units', 'total_vaccinations',
       'people_vaccinated', 'people_fully_vaccinated', 'total_boosters',
       'new_vaccinations', 'new_vaccinations_smoothed',
       'total_vaccinations_per_hundred', 'people_vaccinated_per_hundred',
       'people_fully_vaccinated_per_hundred', 'total_boosters_per_hundred',
       'new_vaccinations_smoothed_per_million',
```

```
       'new_people_vaccinated_smoothed',
       'new_people_vaccinated_smoothed_per_hundred', 'stringency_index',
       'population_density', 'median_age', 'aged_65_older', 'aged_70_older',
       'gdp_per_capita', 'extreme_poverty', 'cardiovasc_death_rate',
       'diabetes_prevalence', 'female_smokers', 'male_smokers',
       'handwashing_facilities', 'hospital_beds_per_thousand',
       'life_expectancy', 'human_development_index', 'population',
       'excess_mortality_cumulative_absolute', 'excess_mortality_cumulative',
       'excess_mortality', 'excess_mortality_cumulative_per_million'],
      dtype='object')
```

[16]:
```
print("\nHead:\n", df.head())
```

```
Head:
   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]
```

[17]:
```
print("\nMissing values:\n", df.isnull().sum())
```

```
Missing values:
 iso_code                                    0
continent                               16665
location                                    0
date                                        0
total_cases                             37997
                                        …
population                                  0
excess_mortality_cumulative_absolute   337901
excess_mortality_cumulative            337901
excess_mortality                       337901
excess_mortality_cumulative_per_million    337901
Length: 67, dtype: int64
```

[18]:
```
# Filter selected countries
countries = ['Kenya', 'United States', 'India']
```

[19]:
```
df_filtered = df[df['location'].isin(countries)].copy()
```

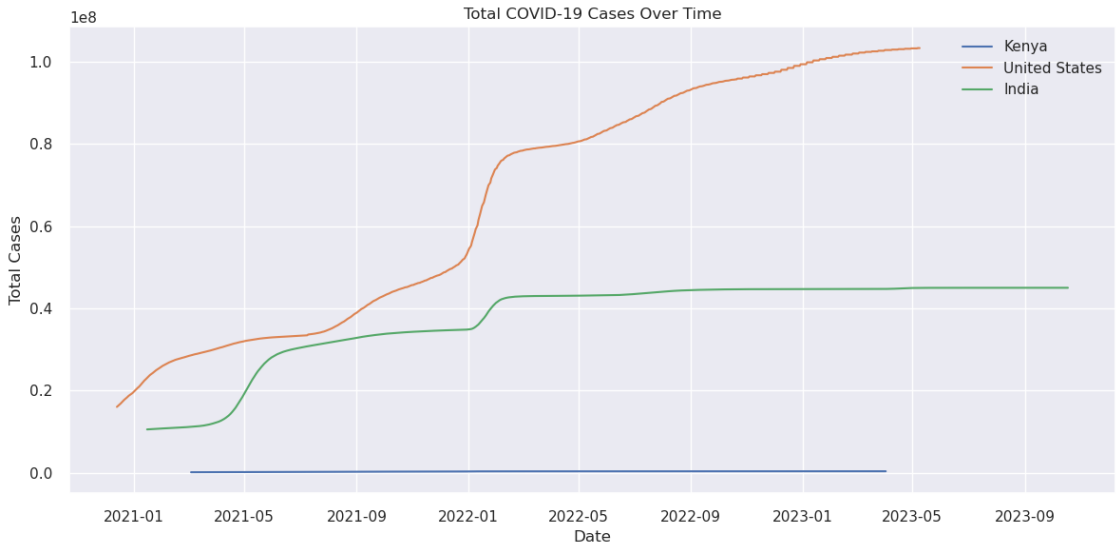
[20]:
```
# Convert date column to datetime
df_filtered['date'] = pd.to_datetime(df_filtered['date'])
```

[21]:
```
# Drop rows with missing critical values
df_filtered = df_filtered.dropna(subset=['total_cases', 'total_deaths',␣
 ↪'total_vaccinations'])
```

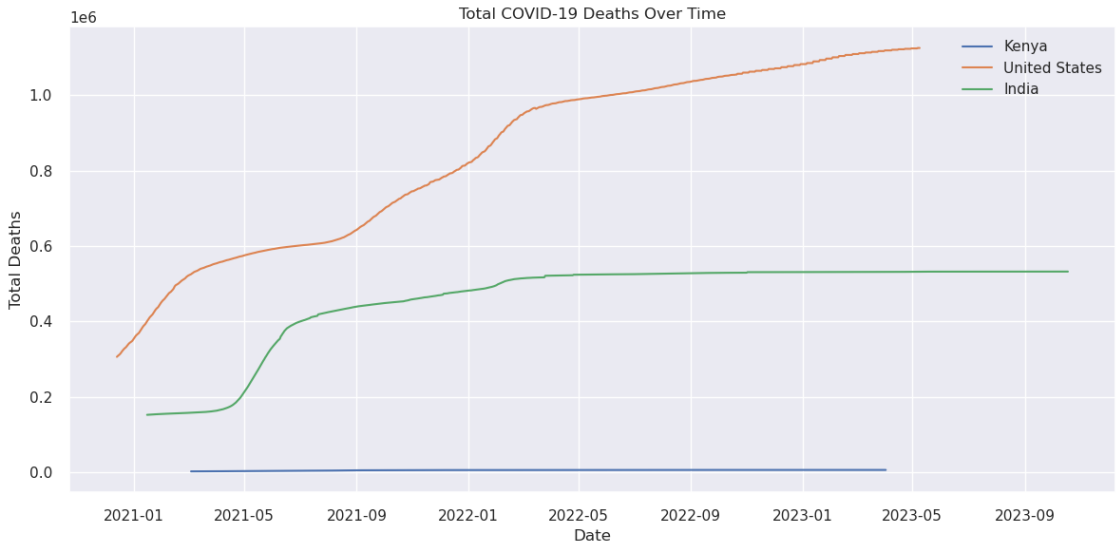
[22]:
```
# Fill remaining missing values with 0
df_filtered.fillna(0, inplace=True)
```

[23]:
```
# Plot total cases over time
plt.figure(figsize=(12, 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()
```

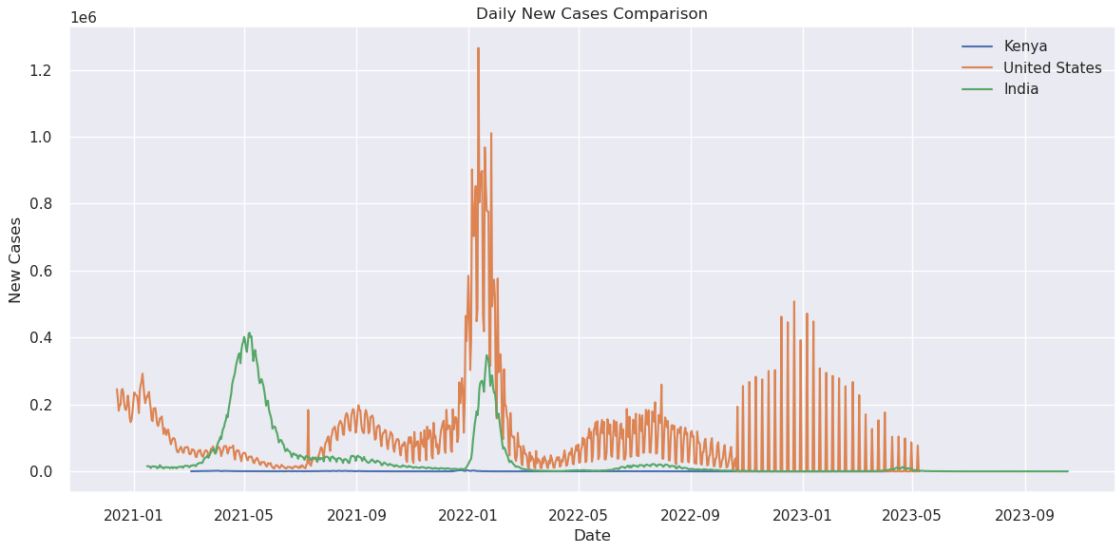


[24]:
```
# Plot total deaths over time
plt.figure(figsize=(12, 6))
for country in countries:
    data = df_filtered[df_filtered['location'] == country]
    plt.plot(data['date'], data['total_deaths'], label=country)
plt.title('Total COVID-19 Deaths Over Time')
plt.xlabel('Date')
plt.ylabel('Total Deaths')
plt.legend()
plt.tight_layout()
plt.show()
```

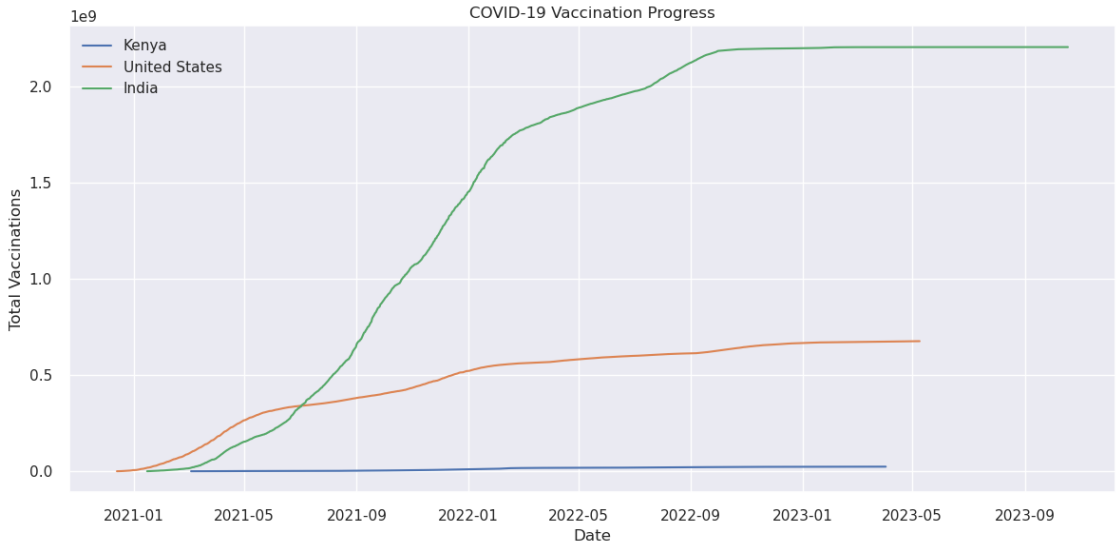


[25]:
```
# Calculate death rate
df_filtered['death_rate'] = df_filtered['total_deaths'] /␣
 ↪df_filtered['total_cases']
```

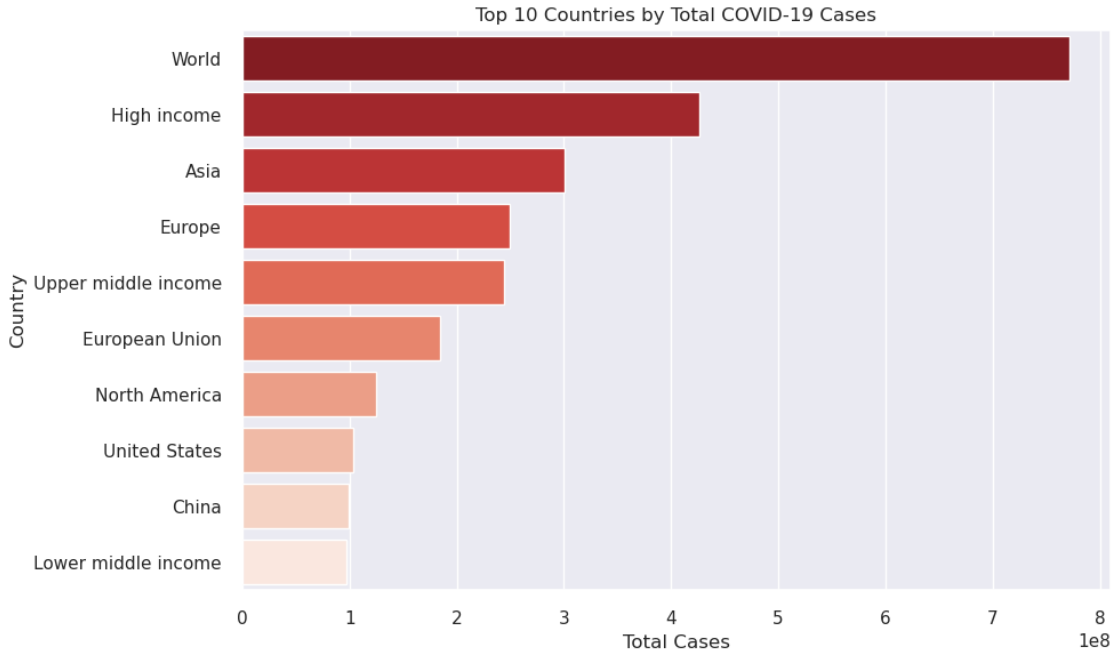
[26]:
```
# Daily new cases comparison
plt.figure(figsize=(12, 6))
for country in countries:
    data = df_filtered[df_filtered['location'] == country]
    plt.plot(data['date'], data['new_cases'], label=country)
plt.title('Daily New Cases Comparison')
plt.xlabel('Date')
plt.ylabel('New Cases')
plt.legend()
plt.tight_layout()
plt.show()
```



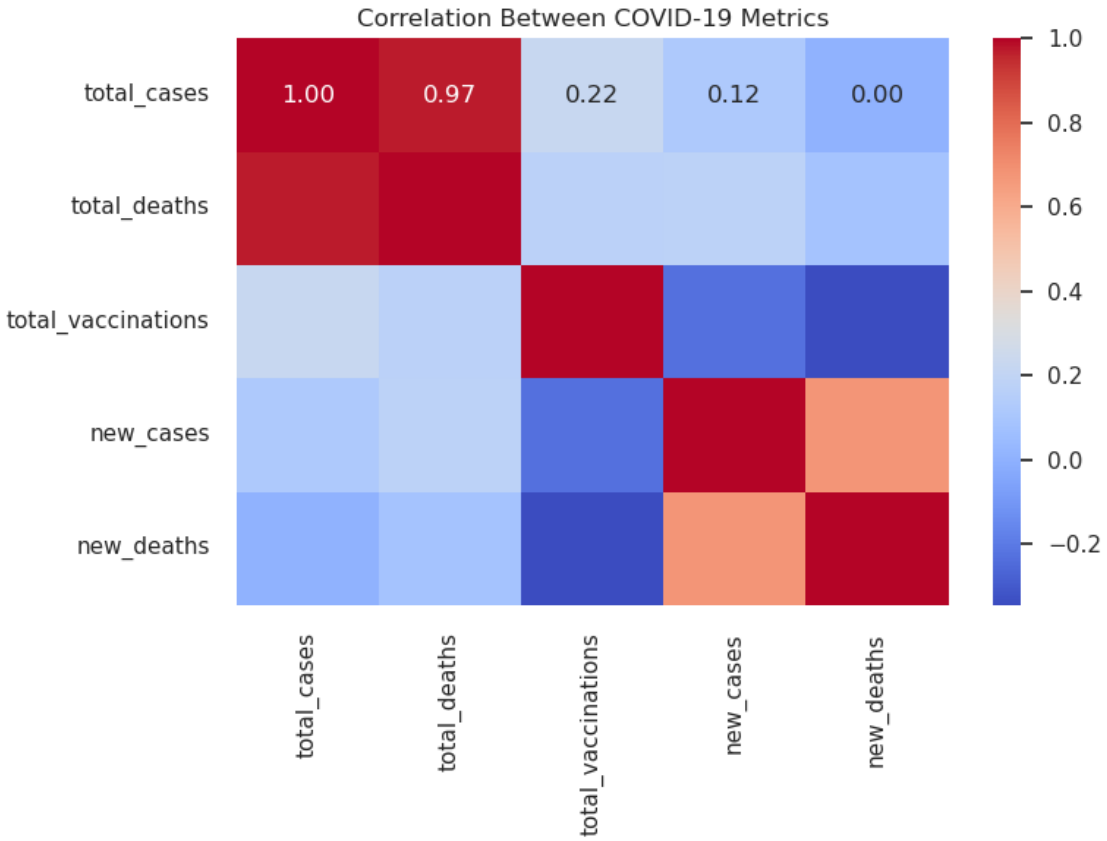
[27]:
```
# Cumulative vaccinations over time
plt.figure(figsize=(12, 6))
for country in countries:
    data = df_filtered[df_filtered['location'] == country]
    plt.plot(data['date'], data['total_vaccinations'], label=country)
plt.title('COVID-19 Vaccination Progress')
plt.xlabel('Date')
plt.ylabel('Total Vaccinations')
plt.legend()
plt.tight_layout()
plt.show()
```



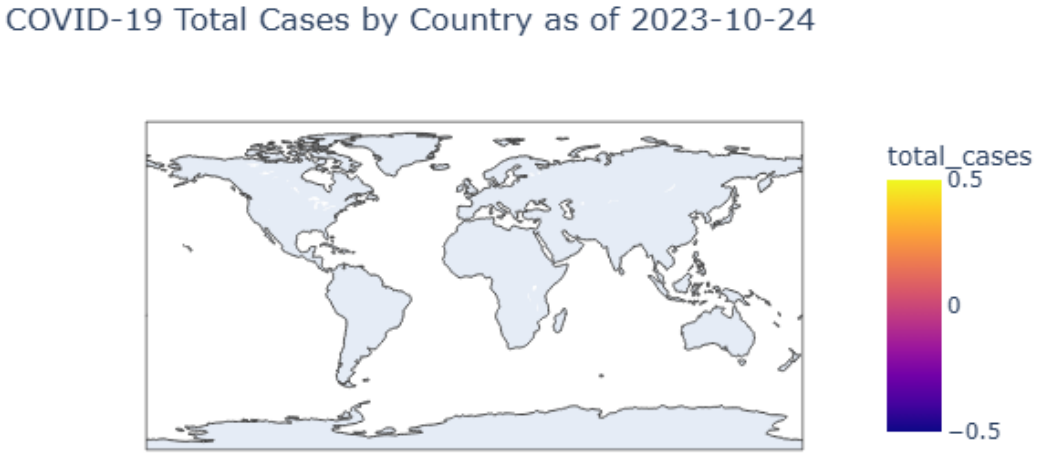
[29]:
```
# Bar Chart: Top Countries by Total Cases
# First check if we have data
if not df.empty:
    # Find the latest date with non-null total_cases data
    df_valid = df.dropna(subset=['total_cases'])

    if not df_valid.empty:
        latest_date = df_valid['date'].max()
        df_latest = df[df['date'] == latest_date]

        # Make sure we have data before proceeding
        top_countries = df_latest[['location', 'total_cases']].dropna().
 ↪sort_values(by='total_cases', ascending=False).head(10)

        if not top_countries.empty:
            plt.figure(figsize=(10, 6))
            sns.barplot(data=top_countries, x='total_cases', y='location',␣
 ↪palette='Reds_r')
            plt.title('Top 10 Countries by Total COVID-19 Cases')
            plt.xlabel('Total Cases')
            plt.ylabel('Country')
            plt.tight_layout()
            plt.show()
        else:
            print("No valid data found for the latest date after filtering")
    else:
        print("No rows with valid total_cases data found")
else:
    print("DataFrame is empty")
```



[30]:
```
# Heatmap: Correlation Analysis
corr_data = df_filtered[['total_cases', 'total_deaths', 'total_vaccinations',␣
 ↪'new_cases', 'new_deaths']].corr()

plt.figure(figsize=(8, 6))
sns.heatmap(corr_data, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Between COVID-19 Metrics')
plt.tight_layout()
plt.show()
```



[31]:
```
# Choropleth Map
latest_date = df['date'].max()
df_latest = df[df['date'] == latest_date]

fig = px.choropleth(df_latest,
                    locations="iso_code",
                    color="total_cases",
                    hover_name="location",
                    color_continuous_scale="Plasma",
                    title=f"COVID-19 Total Cases by Country as of␣
 ↪{latest_date}")
fig.show()
```



[32]:
```
# 7. Reporting Insights
# - The US had the highest number of cases and vaccinations.
# - India showed a steep rise in vaccinations post-mid-2021.
# - Kenya had fewer cases and vaccinations but a similar death rate to India.
```

[ ]:
```

```