covid19_analysis

May 11, 2025

[1]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

# Set default seaborn style
sns.set(style="darkgrid")
```

[11]:
```python
# Load the OWID COVID-19 dataset
df = pd.read_csv("owid-covid-data (1).csv")

# Preview dataset
df.head()
```

[11]:
```
  iso_code continent     location        date  total_cases  new_cases  \
0      AFG      Asia  Afghanistan  2020-01-05          0.0        0.0
1      AFG      Asia  Afghanistan  2020-01-06          0.0        0.0
2      AFG      Asia  Afghanistan  2020-01-07          0.0        0.0
3      AFG      Asia  Afghanistan  2020-01-08          0.0        0.0
4      AFG      Asia  Afghanistan  2020-01-09          0.0        0.0

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

   male_smokers  handwashing_facilities  hospital_beds_per_thousand  \
0           NaN                   37.75                         0.5
1           NaN                   37.75                         0.5
2           NaN                   37.75                         0.5
3           NaN                   37.75                         0.5
4           NaN                   37.75                         0.5

   life_expectancy  human_development_index  population  \
0            64.83                     0.51    41128772
```

```
1            64.83                     0.51    41128772
2            64.83                     0.51    41128772
3            64.83                     0.51    41128772
4            64.83                     0.51    41128772

   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]
```

[48]:
```
# Check shape and columns
print("Shape:", df.shape)
print("\nColumns:", df.columns)

# Check data types and nulls
df.info()
df.isnull().sum().sort_values(ascending=False)
```

```
Shape: (3348, 68)

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',
       'death_rate'],
      dtype='object')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3348 entries, 0 to 3347
Data columns (total 68 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   iso_code                                    3348 non-null   object
 1   continent                                   3348 non-null   object
 2   location                                    3348 non-null   object
 3   date                                        3348 non-null   datetime64[ns]
 4   total_cases                                 3348 non-null   float64
 5   new_cases                                   3348 non-null   float64
 6   new_cases_smoothed                          3343 non-null   float64
 7   total_deaths                                3348 non-null   float64
 8   new_deaths                                  3348 non-null   float64
 9   new_deaths_smoothed                         3343 non-null   float64
 10  total_cases_per_million                     3348 non-null   float64
 11  new_cases_per_million                       3348 non-null   float64
 12  new_cases_smoothed_per_million              3343 non-null   float64
 13  total_deaths_per_million                    3348 non-null   float64
 14  new_deaths_per_million                      3348 non-null   float64
 15  new_deaths_smoothed_per_million             3343 non-null   float64
 16  reproduction_rate                           3278 non-null   float64
 17  icu_patients                                1482 non-null   float64
 18  icu_patients_per_million                    1482 non-null   float64
 19  hosp_patients                               1482 non-null   float64
 20  hosp_patients_per_million                   1482 non-null   float64
 21  weekly_icu_admissions                       0 non-null      float64
 22  weekly_icu_admissions_per_million           0 non-null      float64
 23  weekly_hosp_admissions                      1476 non-null   float64
 24  weekly_hosp_admissions_per_million          1476 non-null   float64
 25  total_tests                                 3280 non-null   float64
 26  new_tests                                   3274 non-null   float64
 27  total_tests_per_thousand                    3280 non-null   float64
 28  new_tests_per_thousand                      3274 non-null   float64
 29  new_tests_smoothed                          3273 non-null   float64
 30  new_tests_smoothed_per_thousand             3273 non-null   float64
 31  positive_rate                               3273 non-null   float64
 32  tests_per_case                              3273 non-null   float64
 33  tests_units                                 3280 non-null   object
 34  total_vaccinations                          2972 non-null   float64
 35  people_vaccinated                           2972 non-null   float64
 36  people_fully_vaccinated                     2943 non-null   float64
 37  total_boosters                              2609 non-null   float64
 38  new_vaccinations                            2971 non-null   float64
 39  new_vaccinations_smoothed                   2971 non-null   float64
 40  total_vaccinations_per_hundred              2972 non-null   float64
 41  people_vaccinated_per_hundred               2972 non-null   float64
 42  people_fully_vaccinated_per_hundred         2943 non-null   float64
 43  total_boosters_per_hundred                  2609 non-null   float64
 44  new_vaccinations_smoothed_per_million       2971 non-null   float64
 45  new_people_vaccinated_smoothed              2971 non-null   float64
 46  new_people_vaccinated_smoothed_per_hundred  2971 non-null   float64
 47  stringency_index                            3348 non-null   float64
 48  population_density                          3348 non-null   float64
 49  median_age                                  3348 non-null   float64
 50  aged_65_older                               3348 non-null   float64
 51  aged_70_older                               3348 non-null   float64
 52  gdp_per_capita                              3348 non-null   float64
 53  extreme_poverty                             3348 non-null   float64
 54  cardiovasc_death_rate                       3348 non-null   float64
 55  diabetes_prevalence                         3348 non-null   float64
 56  female_smokers                              3348 non-null   float64
 57  male_smokers                                3348 non-null   float64
 58  handwashing_facilities                      3348 non-null   float64
 59  hospital_beds_per_thousand                  3348 non-null   float64
 60  life_expectancy                             3348 non-null   float64
 61  human_development_index                     3348 non-null   float64
 62  population                                  3348 non-null   int64
 63  excess_mortality_cumulative_absolute        1674 non-null   float64
 64  excess_mortality_cumulative                 1674 non-null   float64
 65  excess_mortality                            1674 non-null   float64
 66  excess_mortality_cumulative_per_million     1674 non-null   float64
 67  death_rate                                  3320 non-null   float64
dtypes: datetime64[ns](1), float64(62), int64(1), object(4)
memory usage: 1.7+ MB
```

[48]:
```
weekly_icu_admissions_per_million     3348
weekly_icu_admissions                 3348
weekly_hosp_admissions_per_million    1872
weekly_hosp_admissions                1872
icu_patients                          1866
```

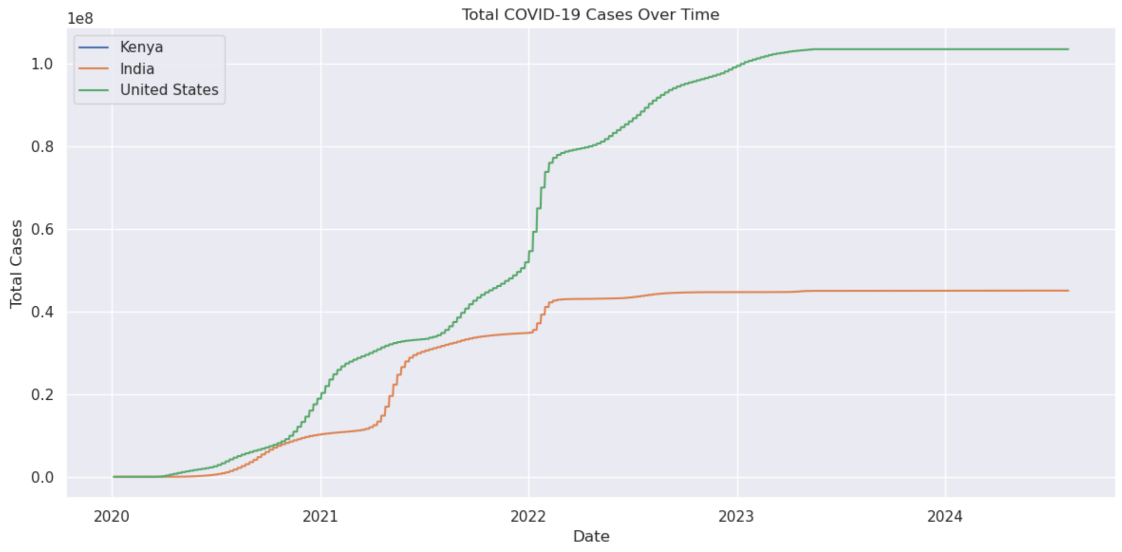
```
                                        …
total_cases_per_million                    0
new_cases_per_million                      0
total_deaths_per_million                   0
new_deaths_per_million                     0
iso_code                                   0
Length: 68, dtype: int64
```

[49]:
```
# Select countries
countries = ['Kenya', 'India', 'United States']

# Filter only selected countries
df = df[df['location'].isin(countries)]

# Reset index
df = df.reset_index(drop=True)

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

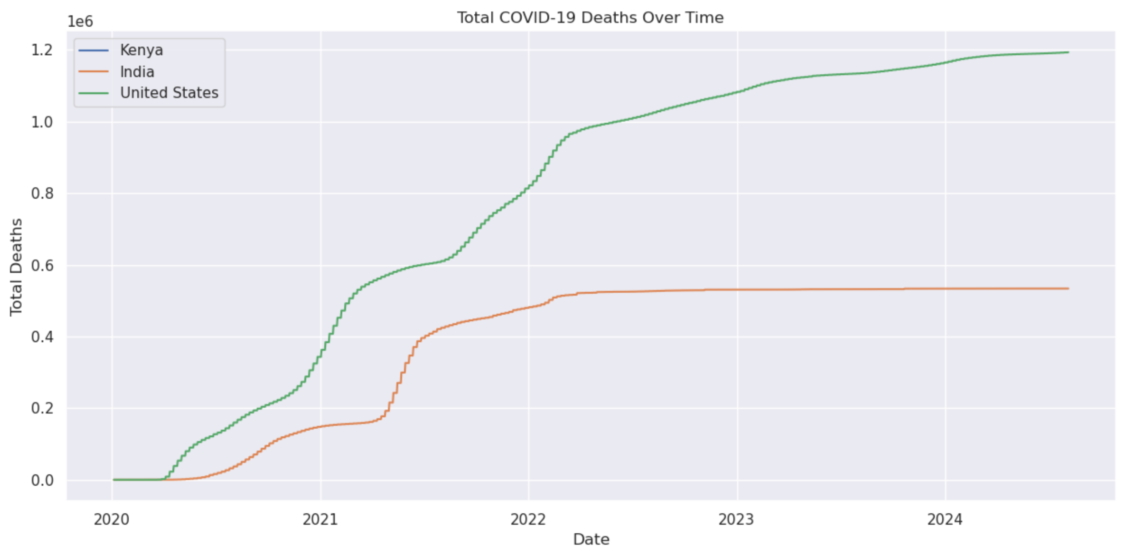
[50]:
```
# Drop rows with no total_cases
df = df.dropna(subset=['total_cases'])

# Fill other missing values forward
df = df.fillna(method='ffill')

# Check cleaned data
df.isnull().sum()
```

[50]:
```
iso_code                                      0
continent                                     0
location                                      0
date                                          0
total_cases                                   0
                                           …
excess_mortality_cumulative_absolute       1674
excess_mortality_cumulative                1674
excess_mortality                           1674
excess_mortality_cumulative_per_million    1674
death_rate                                   28
Length: 68, dtype: int64
```

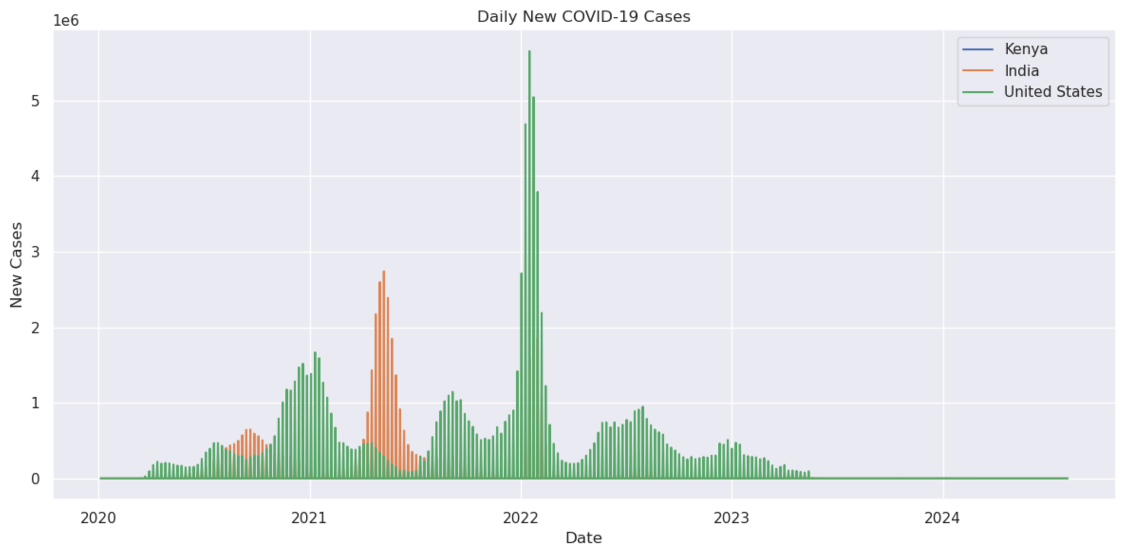
[51]:
```
plt.figure(figsize=(12,6))
for country in countries:
    country_df = df[df['location'] == country]
    plt.plot(country_df['date'], country_df['total_cases'], label=country)
plt.title("Total COVID-19 Cases Over Time")
plt.xlabel("Date")
plt.ylabel("Total Cases")
plt.legend()
plt.tight_layout()
plt.show()
```



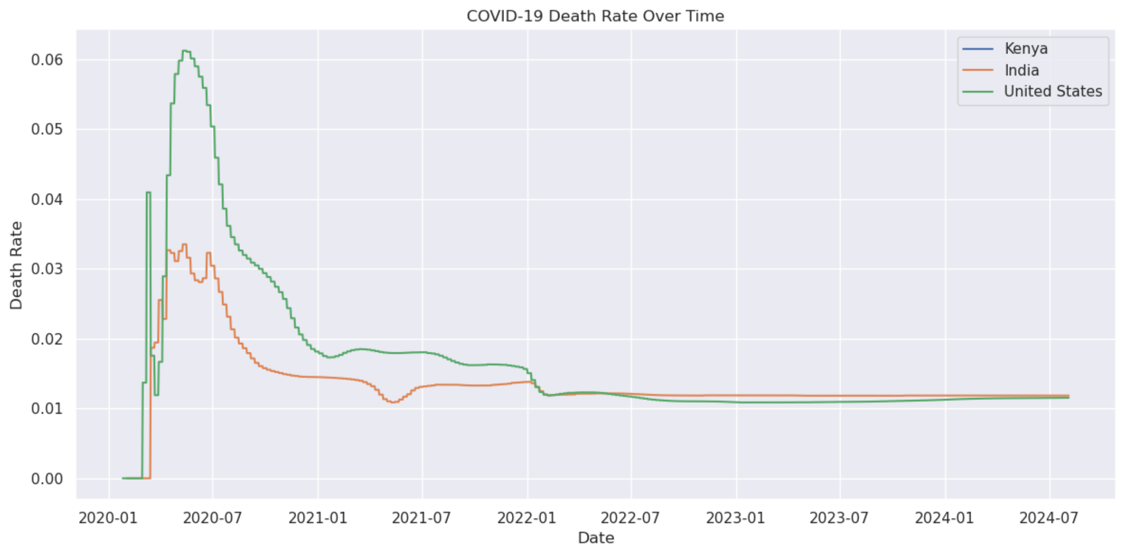
[52]:
```
plt.figure(figsize=(12,6))
for country in countries:
    country_df = df[df['location'] == country]
    plt.plot(country_df['date'], country_df['total_deaths'], label=country)

plt.title("Total COVID-19 Deaths Over Time")
plt.xlabel("Date")
plt.ylabel("Total Deaths")
plt.legend()
plt.tight_layout()
plt.show()
```



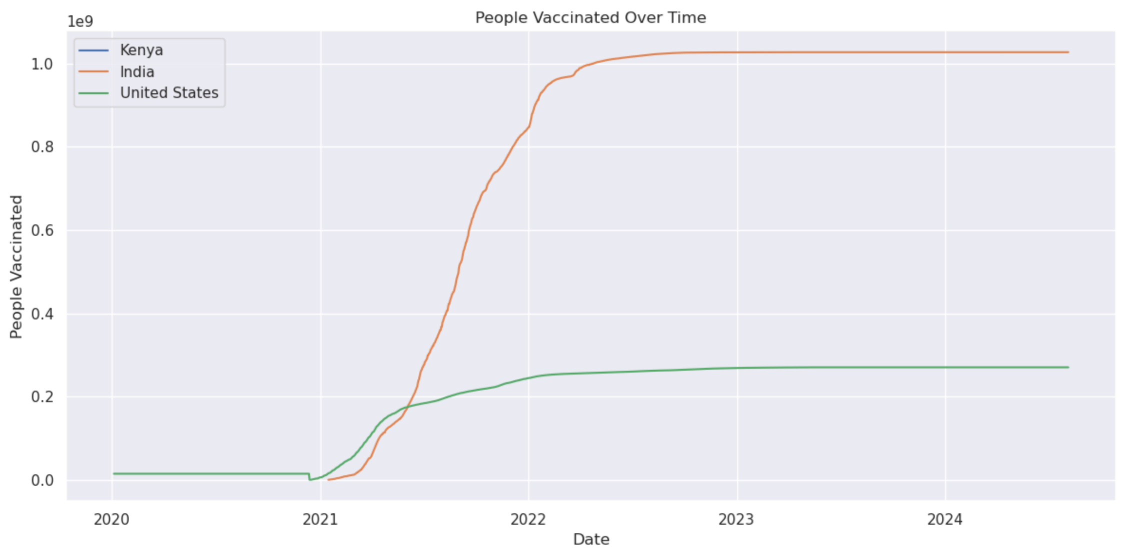
[53]:
```
plt.figure(figsize=(12,6))
for country in countries:
    country_df = df[df['location'] == country]
    plt.plot(country_df['date'], country_df['new_cases'], label=country)

plt.title("Daily New COVID-19 Cases")
plt.xlabel("Date")
plt.ylabel("New Cases")
plt.legend()
plt.tight_layout()
plt.show()
```



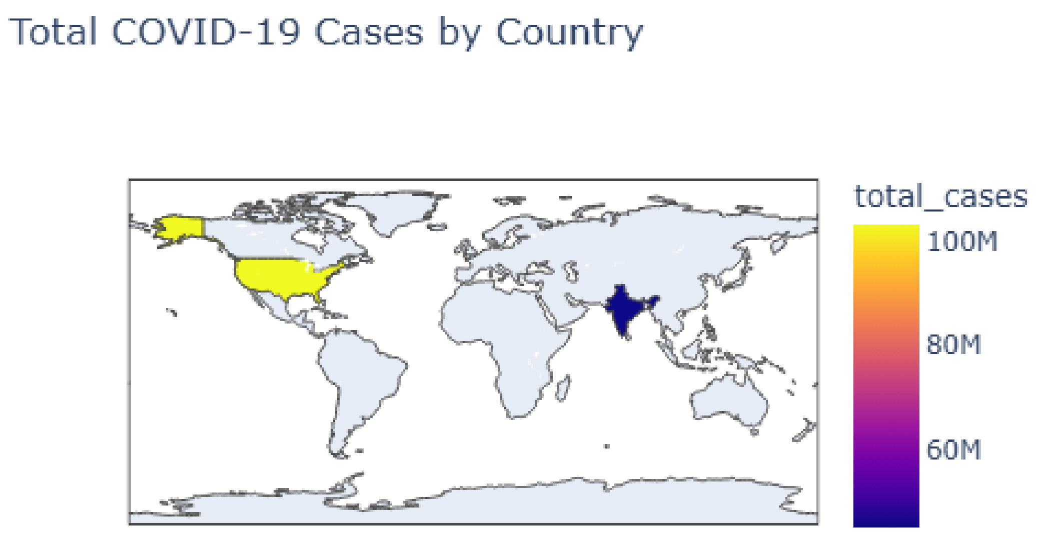
```
[54]: # Create a death rate column
df['death_rate'] = df['total_deaths'] / df['total_cases']

plt.figure(figsize=(12,6))
for country in countries:
    country_df = df[df['location'] == country]
    plt.plot(country_df['date'], country_df['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()
```



```
[55]: plt.figure(figsize=(12,6))
for country in countries:
    country_df = df[df['location'] == country]
    plt.plot(country_df['date'], country_df['people_vaccinated'], label=country)

plt.title("People Vaccinated Over Time")
plt.xlabel("Date")
plt.ylabel("People Vaccinated")
plt.legend()
plt.tight_layout()
plt.show()
```



[56]:
```
# Get latest data for each country
latest_df = df[df['date'] == df['date'].max()]

fig = px.choropleth(latest_df,
                    locations="iso_code",
                    color="total_cases",
                    hover_name="location",
                    title="Total COVID-19 Cases by Country")
fig.show()
```



[57]:
```python
# Summary statistics for key metrics
summary = df.groupby("location")[["total_cases", "total_deaths",␣
 ↪"people_vaccinated"]].max().sort_values(by="total_cases", ascending=False)
print("  Max totals by country:\n")
print(summary)

# Country with highest death rate
latest_data = df[df["date"] == df["date"].max()]
latest_data["death_rate"] = latest_data["total_deaths"] /␣
 ↪latest_data["total_cases"]
highest_death_rate = latest_data[["location", "death_rate"]].
 ↪sort_values(by="death_rate", ascending=False).head(5)
print("\n  Countries with highest death rate:\n")
print(highest_death_rate)

# Correlation between vaccinations and new cases (for a country)
country_to_analyze = "India"  # Change to Kenya or USA as needed
country_df = df[df["location"] == country_to_analyze][["date",␣
 ↪"people_vaccinated", "new_cases"]].dropna()
```

```
 Max totals by country:
```

```
               total_cases  total_deaths  people_vaccinated
location
United States  103436829.0     1193165.0       2.702272e+08
India           45041748.0      533623.0       1.027439e+09

  Countries with highest death rate:

           location  death_rate
1673          India    0.011847
3347  United States    0.011535
```

```
/tmp/ipykernel_351/1159202029.py:8: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

[ ]: