Covid_proj_Data_analysis

May 5, 2025

[1]:
```
import pandas as pd

# Load dataset
df = pd.read_csv("owid-covid-data.csv")

# Inspect the dataset
print("Columns in dataset:", df.columns)
print("First few rows:", df.head())

# Check for missing values
print("Missing values count:", df.isnull().sum())
```

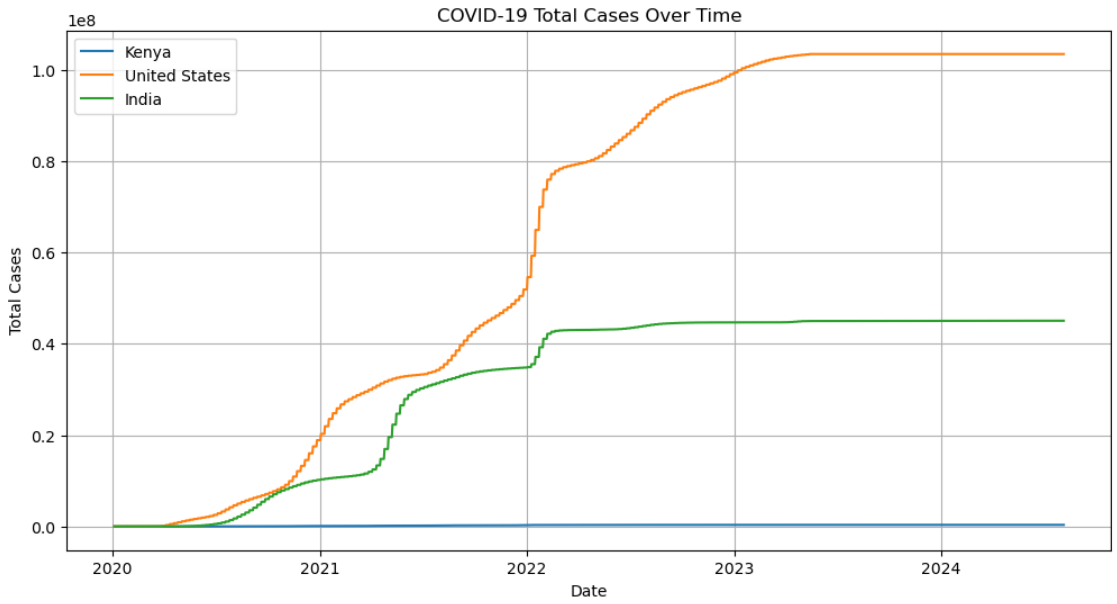
```
Columns in dataset: Index(['iso_code', 'continent', 'location', 'date',
'total_cases', 'new_cases',
       'new_cases_smoothed', 'total_deaths', 'new_deaths',
       'new_deaths_smoothed', 'total_cases_per_million',
       'new_cases_per_million', 'new_cases_smoothed_per_million',
       'total_deaths_per_million', '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')
First few rows:   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]
Missing values count: iso_code                                    0
```

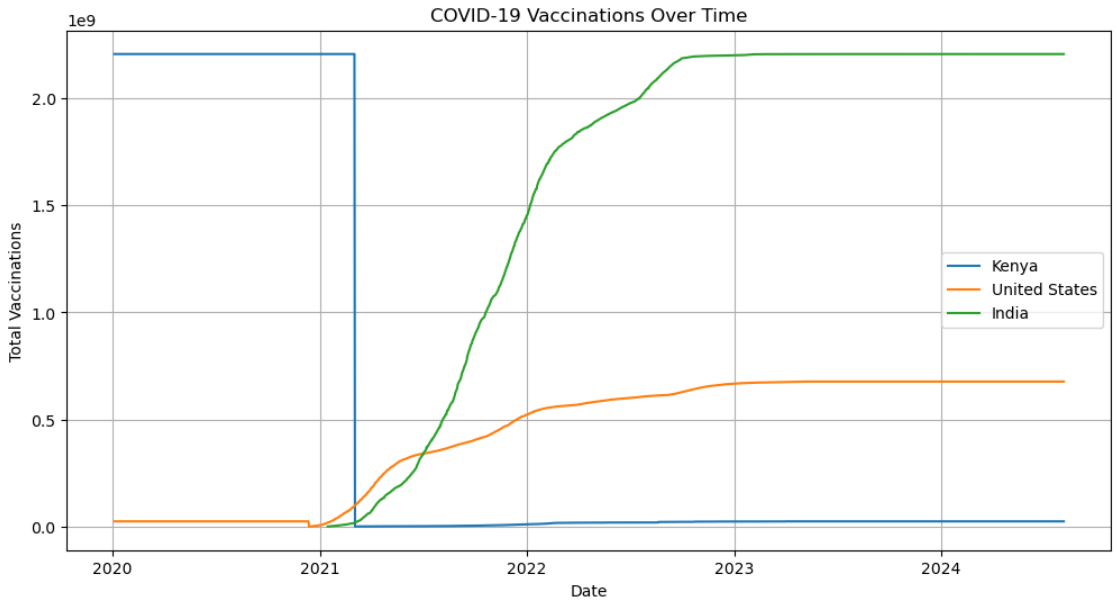
```
continent                                  26525
location                                       0
date                                           0
total_cases                                17631
                                             ...
population                                     0
excess_mortality_cumulative_absolute      416024
excess_mortality_cumulative               416024
excess_mortality                          416024
excess_mortality_cumulative_per_million   416024
Length: 67, dtype: int64
```

[4]:
```python
# Filtering for selected countries (Kenya, USA, India)
selected_countries = ['Kenya', 'United States', 'India']
df_filtered = df[df['location'].isin(selected_countries)]

# Dropping rows with missing critical values
df_filtered = df_filtered.dropna(subset=['date', 'total_cases', 'total_deaths'])

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

# Handling missing numeric values
df_filtered.ffill(inplace=True)  # Forward fill missing values
```

[5]:
```python
import matplotlib.pyplot as plt
import seaborn as sns

# Plot total cases over time
plt.figure(figsize=(12,6))
for country in selected_countries:
    country_data = df_filtered[df_filtered['location'] == country]
    plt.plot(country_data['date'], country_data['total_cases'], label=country)

plt.xlabel("Date")
plt.ylabel("Total Cases")
plt.title("COVID-19 Total Cases Over Time")
plt.legend()
plt.grid(True)
plt.show()
```



[6]:
```
# Plot cumulative vaccinations over time
plt.figure(figsize=(12,6))
for country in selected_countries:
    country_data = df_filtered[df_filtered['location'] == country]
    plt.plot(country_data['date'], country_data['total_vaccinations'],␣
 ↪label=country)

plt.xlabel("Date")
plt.ylabel("Total Vaccinations")
plt.title("COVID-19 Vaccinations Over Time")
plt.legend()
plt.grid(True)
plt.show()
```



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

# Preparing latest data for visualization
latest_data = df_filtered[df_filtered['date'] == df_filtered['date'].max()]
fig = px.choropleth(latest_data, locations="iso_code", color="total_cases",
                    hover_name="location", title="COVID-19 Cases by Country")

fig.show()
```

[ ]:
```

```

[10]:
```
import pandas as pd

# Load dataset
df = pd.read_csv("owid-covid-data.csv")
df['date'] = pd.to_datetime(df['date'])

# User inputs
user_country = input("Enter country name (e.g., Kenya, USA, India): ").strip()
start_date_str = input("Enter start date (YYYY-MM-DD): ").strip()
end_date_str = input("Enter end date (YYYY-MM-DD): ").strip()

# Convert inputs to datetime safely
try:
    start_date = pd.to_datetime(start_date_str, format="%Y-%m-%d",␣
 ↪errors="coerce")
    end_date = pd.to_datetime(end_date_str, format="%Y-%m-%d", errors="coerce")
```

```
    if pd.isnull(start_date) or pd.isnull(end_date):
        raise ValueError("Invalid date format. Please enter dates in YYYY-MM-DD␣
 ↪format.")
except ValueError as e:
    print(e)
    exit()

# Filter data based on user input
df_filtered = df[(df['location'] == user_country) & (df['date'] >= start_date)␣
 ↪& (df['date'] <= end_date)]

# Display filtered results
print(df_filtered.head())
```

```
Enter country name (e.g., Kenya, USA, India):  South Africa
Enter start date (YYYY-MM-DD):  2020-05-05
Enter end date (YYYY-MM-DD):  2021-05-05
       iso_code continent     location       date  total_cases  new_cases  \
356011      ZAF    Africa  South Africa 2020-05-05       6336.0        0.0
356012      ZAF    Africa  South Africa 2020-05-06       6336.0        0.0
356013      ZAF    Africa  South Africa 2020-05-07       6336.0        0.0
356014      ZAF    Africa  South Africa 2020-05-08       6336.0        0.0
356015      ZAF    Africa  South Africa 2020-05-09       6336.0        0.0

        new_cases_smoothed  total_deaths  new_deaths  new_deaths_smoothed  \
356011              282.14         123.0         0.0                 5.29
356012              282.14         123.0         0.0                 5.29
356013              282.14         123.0         0.0                 5.29
356014              282.14         123.0         0.0                 5.29
356015              282.14         123.0         0.0                 5.29

        …  male_smokers  handwashing_facilities  hospital_beds_per_thousand  \
356011  …          33.2                   43.99                        2.32
356012  …          33.2                   43.99                        2.32
356013  …          33.2                   43.99                        2.32
356014  …          33.2                   43.99                        2.32
356015  …          33.2                   43.99                        2.32

        life_expectancy  human_development_index  population  \
356011            64.13                     0.71    59893884
356012            64.13                     0.71    59893884
356013            64.13                     0.71    59893884
356014            64.13                     0.71    59893884
356015            64.13                     0.71    59893884

        excess_mortality_cumulative_absolute  excess_mortality_cumulative  \
```

```
356011                                  NaN                          NaN
356012                                  NaN                          NaN
356013                                  NaN                          NaN
356014                                  NaN                          NaN
356015                                  NaN                          NaN

        excess_mortality  excess_mortality_cumulative_per_million
356011               NaN                                     NaN
356012               NaN                                     NaN
356013               NaN                                     NaN
356014               NaN                                     NaN
356015               NaN                                     NaN

[5 rows x 67 columns]
```

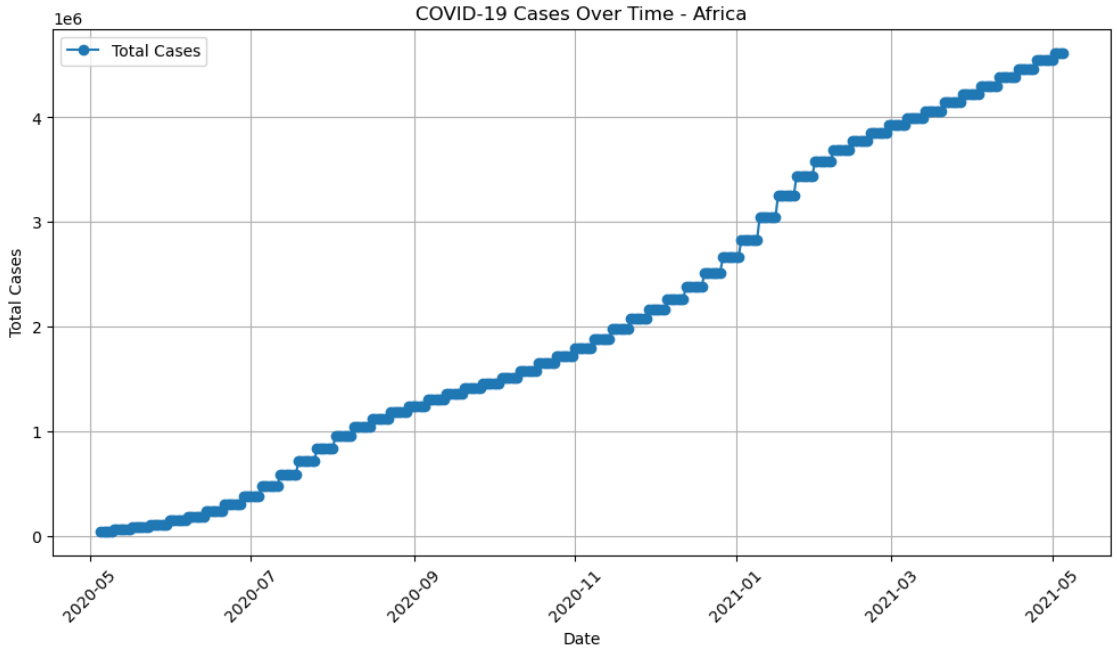
[ ]:

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

# Load dataset
df = pd.read_csv("owid-covid-data.csv")

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

# Allow all valid country names dynamically
valid_countries = df['location'].unique().tolist()

# User selects a country
user_country = input(f"Enter country name (Valid options: {', '.
 ↪join(valid_countries[:10])}...): ").strip()

# Validate input country
if user_country not in valid_countries:
    print("Invalid country! Please enter a valid country name.")
    exit()

# Get user date input
start_date_str = input("Enter start date (YYYY-MM-DD): ").strip()
end_date_str = input("Enter end date (YYYY-MM-DD): ").strip()

# Convert inputs to datetime safely
try:
    start_date = pd.to_datetime(start_date_str, format="%Y-%m-%d",␣
 ↪errors="coerce")
    end_date = pd.to_datetime(end_date_str, format="%Y-%m-%d", errors="coerce")
```

```
    if pd.isnull(start_date) or pd.isnull(end_date):
        raise ValueError("Invalid date format. Please enter dates in YYYY-MM-DD␣
 ↪format.")
except ValueError as e:
    print(e)
    exit()

# Filter data safely
df_filtered = df[df['location'] == user_country].copy()

# Handle missing values correctly
df_filtered.ffill(inplace=True)  # Forward fill missing values

# Apply date filters
df_user = df_filtered[(df_filtered['date'] >= start_date) &␣
 ↪(df_filtered['date'] <= end_date)]

# Check if filtered data exists
if df_user.empty:
    print(f"No data found for {user_country} in the selected date range.")
    exit()

# Display first few rows of filtered data
print(df_user.head())

# Plot total cases over time
plt.figure(figsize=(12, 6))
plt.plot(df_user['date'], df_user['total_cases'], marker='o', linestyle='-',␣
 ↪label="Total Cases")
plt.xlabel("Date")
plt.ylabel("Total Cases")
plt.title(f"COVID-19 Cases Over Time - {user_country}")
plt.xticks(rotation=45)
plt.grid(True)
plt.legend()
plt.show()
```

```
Enter country name (Valid options: Afghanistan, Africa, Albania, Algeria,
American Samoa, Andorra, Angola, Anguilla, Antigua and Barbuda, Argentina…):
Africa
Enter start date (YYYY-MM-DD):  2020-05-05
Enter end date (YYYY-MM-DD):  2021-05-05
      iso_code  continent location       date  total_cases  new_cases  \
1795  OWID_AFR        NaN   Africa 2020-05-05      44003.0        0.0
1796  OWID_AFR        NaN   Africa 2020-05-06      44003.0        0.0
1797  OWID_AFR        NaN   Africa 2020-05-07      44003.0        0.0
```

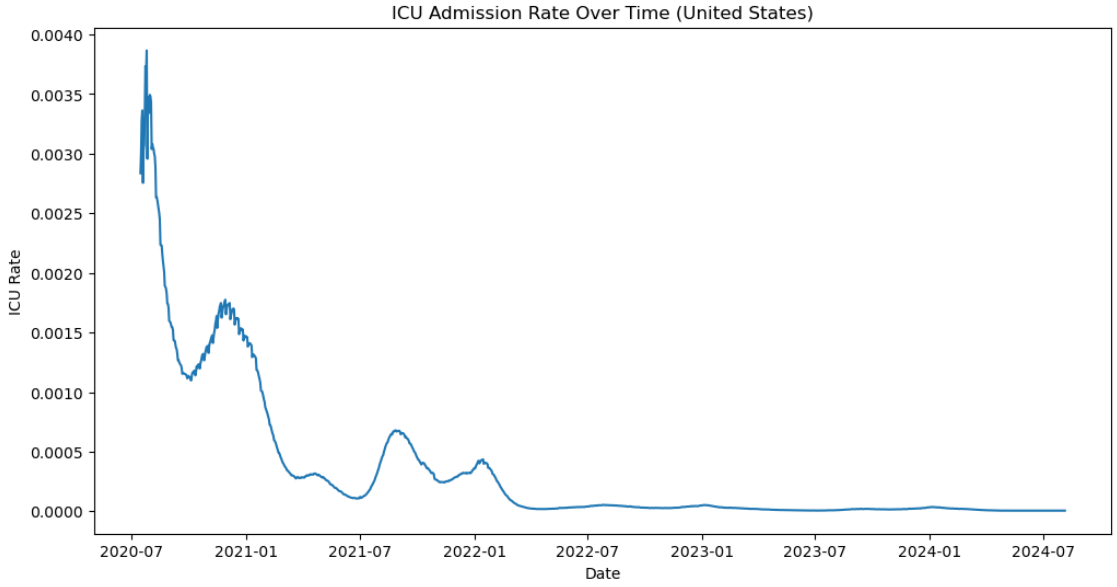
```
1798  OWID_AFR        NaN   Africa 2020-05-08      44003.0        0.0
1799  OWID_AFR        NaN   Africa 2020-05-09      44003.0        0.0

      new_cases_smoothed  total_deaths  new_deaths  new_deaths_smoothed  …  \
1795             1811.57        2151.0         0.0                55.14  …
1796             1811.57        2151.0         0.0                55.14  …
1797             1811.57        2151.0         0.0                55.14  …
1798             1811.57        2151.0         0.0                55.14  …
1799             1811.57        2151.0         0.0                55.14  …

      male_smokers  handwashing_facilities  hospital_beds_per_thousand  \
1795           NaN                     NaN                         NaN
1796           NaN                     NaN                         NaN
1797           NaN                     NaN                         NaN
1798           NaN                     NaN                         NaN
1799           NaN                     NaN                         NaN

      life_expectancy  human_development_index  population  \
1795              NaN                      NaN  1426736614
1796              NaN                      NaN  1426736614
1797              NaN                      NaN  1426736614
1798              NaN                      NaN  1426736614
1799              NaN                      NaN  1426736614

      excess_mortality_cumulative_absolute  excess_mortality_cumulative  \
1795                                   NaN                          NaN
1796                                   NaN                          NaN
1797                                   NaN                          NaN
1798                                   NaN                          NaN
1799                                   NaN                          NaN

      excess_mortality  excess_mortality_cumulative_per_million
1795               NaN                                     NaN
1796               NaN                                     NaN
1797               NaN                                     NaN
1798               NaN                                     NaN
1799               NaN                                     NaN

[5 rows x 67 columns]
```



[ ]:

[3]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load dataset
df = pd.read_csv("owid-covid-data.csv")
df['date'] = pd.to_datetime(df['date'])

# Define country (ensure 'user_country' is set)
user_country = "United States"  # Example

# Filter data
df_filtered = df[df['location'] == user_country].copy()
df_filtered.ffill(inplace=True)  # Handle missing values

# Check if ICU data exists before running calculations
if 'icu_patients' in df.columns:
    df_filtered['ICU Rate'] = df_filtered['icu_patients'] /␣
 ↪df_filtered['total_cases']

    # Plot ICU rate over time
    plt.figure(figsize=(12, 6))
    plt.plot(df_filtered['date'], df_filtered['ICU Rate'])
    plt.xlabel("Date")
```

```
    plt.ylabel("ICU Rate")
    plt.title(f"ICU Admission Rate Over Time ({user_country})")
    plt.show()
else:
    print("ICU data is not available in this dataset.")
```



[ ]: