covid_analysis

May 14, 2025

[1]:

```
import pandas as pd
```

[2]:

```
# loading data
df = pd.read_csv('owid-covid-data.csv')
```

[3]:

```
# shows preview of the data
df.head()
```

[3]:

```
  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  41128772.0
1            64.83                     0.51  41128772.0
2            64.83                     0.51  41128772.0
3            64.83                     0.51  41128772.0
4            64.83                     0.51  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]
```

[4]:
```
# Checking columns
df.columns
```

[4]:
```
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')
```

[5]:

```
# Check how many missing values
df.isnull().sum()
```

[5]:

```
iso_code                                        0
continent                                   16799
location                                        0
date                                            0
total_cases                                  9338
                                            …
population                                      1
excess_mortality_cumulative_absolute       259010
excess_mortality_cumulative                259010
excess_mortality                           259010
excess_mortality_cumulative_per_million    259010
Length: 67, dtype: int64
```

[6]:

```
# DATA CLEANING
# Goal: Prepare data for analysis
#  Filter countries of interest
countries = ['Kenya', 'United States', 'India', 'Brazil', 'South Africa',␣
 ↪'China']
df_filtered = df[df['location'].isin(countries)]
```

[7]:

```
# DROPPING ROWS WITH CRITICAL VALUES
df_filtered = df_filtered.dropna(subset=['date', 'total_cases', 'total_deaths'])
```

[8]:

```
# Converting the date column to datetime
df_filtered['date'] = pd.to_datetime(df_filtered['date'])
```

[9]:

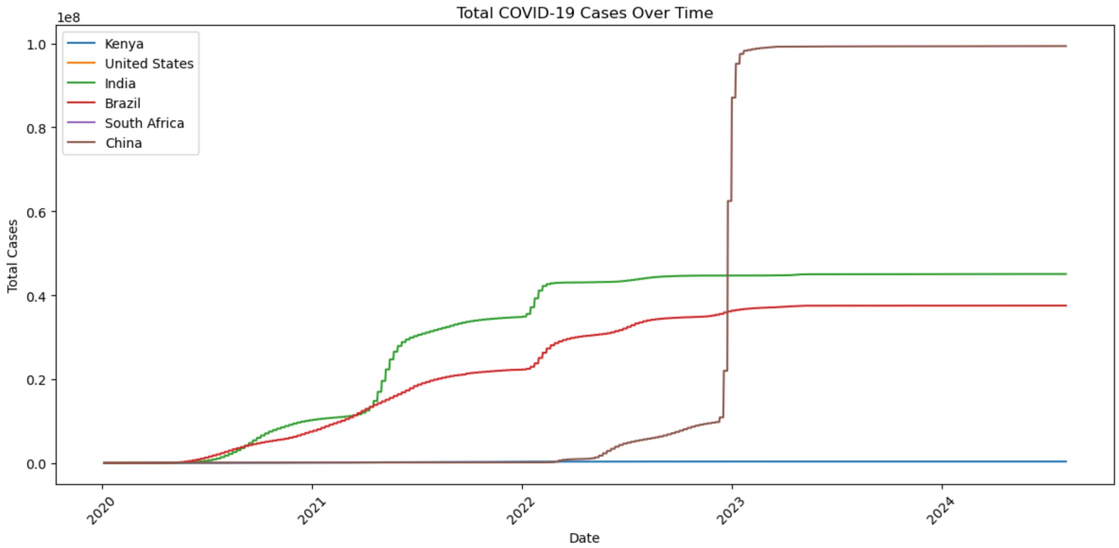
```
# Handling missing numeric values with  Interpolate
df_filtered[['new_cases', 'new_deaths', 'total_vaccinations']] =␣
 ↪df_filtered[['new_cases', 'new_deaths', 'total_vaccinations']].interpolate()
```

[11]:

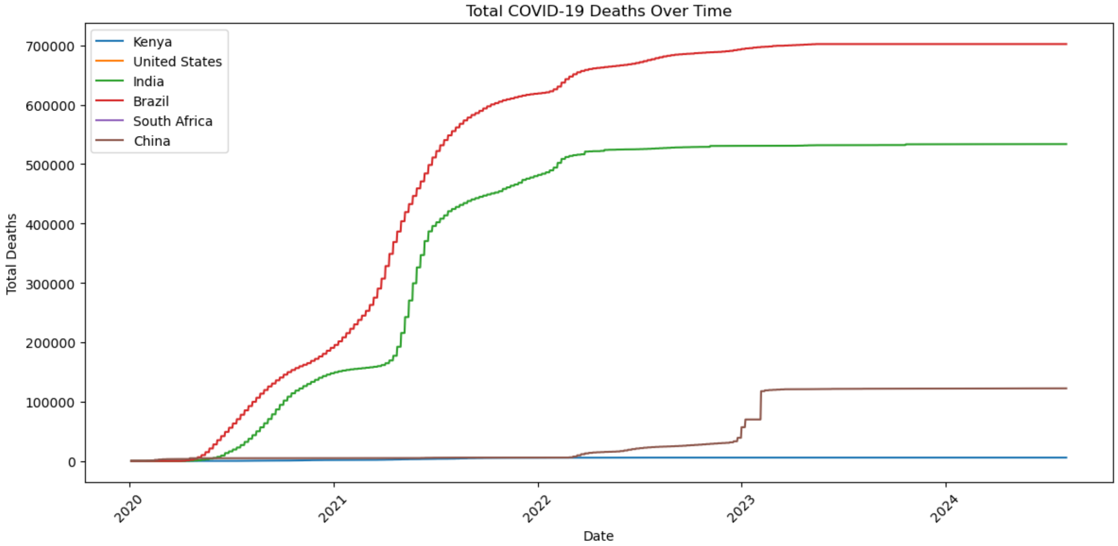
```
# Exploratory Data Analysis
# Goal: Generate descriptive statistics & explore trends
import matplotlib.pyplot as plt
import seaborn as sns
```

[12]:

```
#  Plot total cases over time
plt.figure(figsize=(12, 6))

for country in countries:
    country_data = df_filtered[df_filtered['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()
```

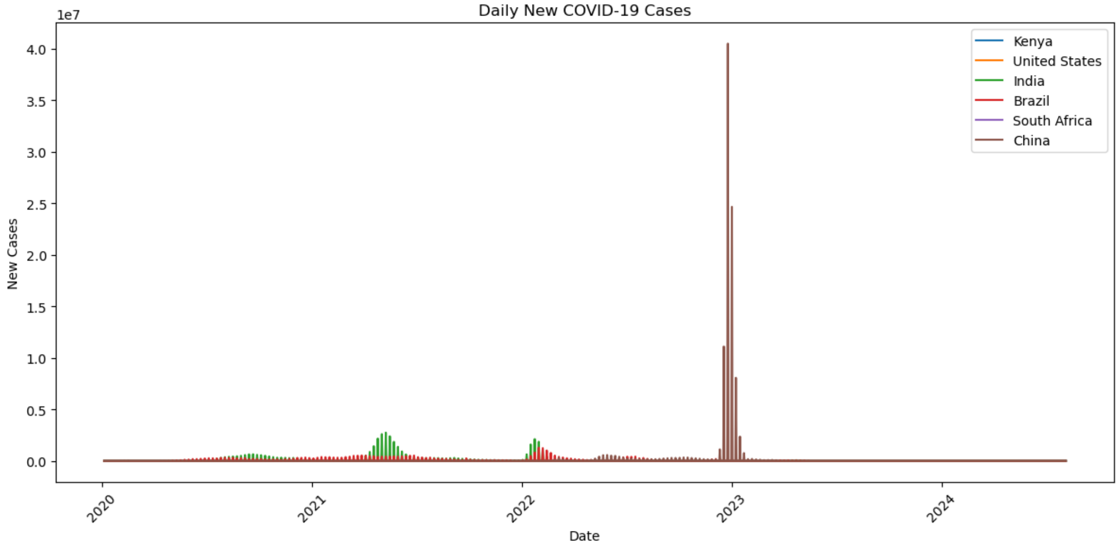


[13]:
```
# Plot total deaths over time
plt.figure(figsize=(12, 6))

for country in countries:
    country_data = df_filtered[df_filtered['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()
```



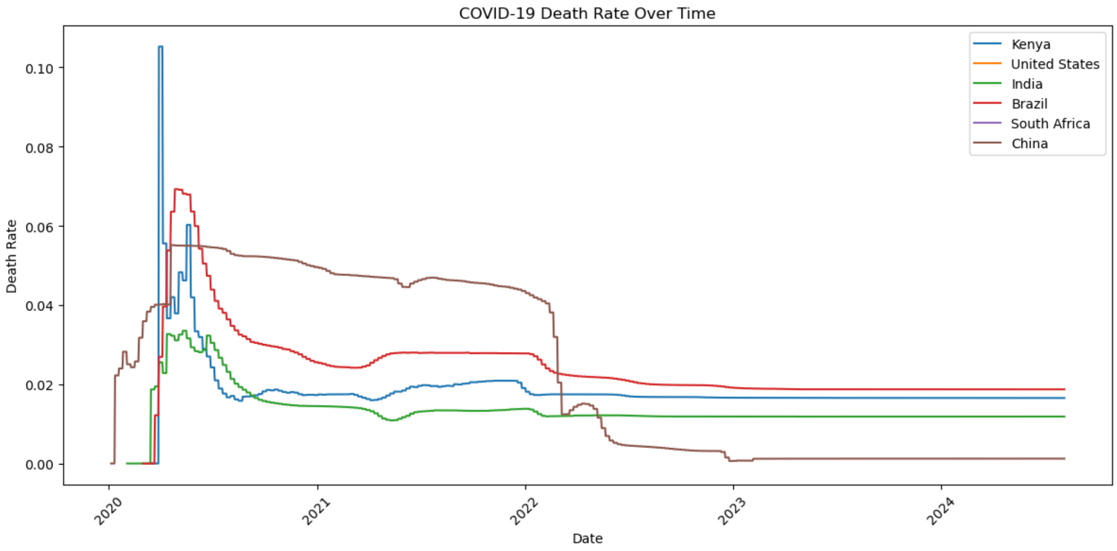
[15]:

```
# Compare daily new cases between the filtered countries
plt.figure(figsize=(12, 6))

for country in countries:
    country_data = df_filtered[df_filtered['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.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()
```

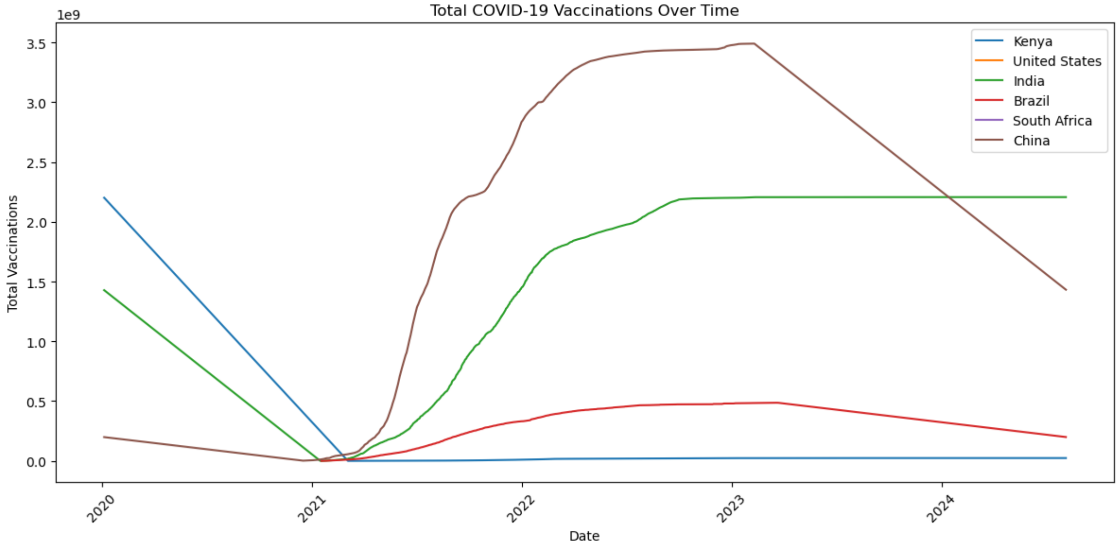


[16]:

```
# Calculating the death rate: total_deaths / total_cases
# Creating a new column for death rate
df_filtered['death_rate'] = df_filtered['total_deaths'] /␣
  ↪df_filtered['total_cases']

# Plot death rate over time
plt.figure(figsize=(12, 6))

for country in countries:
    country_data = df_filtered[df_filtered['location'] == country]
    plt.plot(country_data['date'], country_data['death_rate'], label=country)

plt.title('COVID-19 Death Rate Over Time')
plt.xlabel('Date')
plt.ylabel('Death Rate')
plt.xticks(rotation=45)
plt.legend()
plt.tight_layout()
plt.show()
```

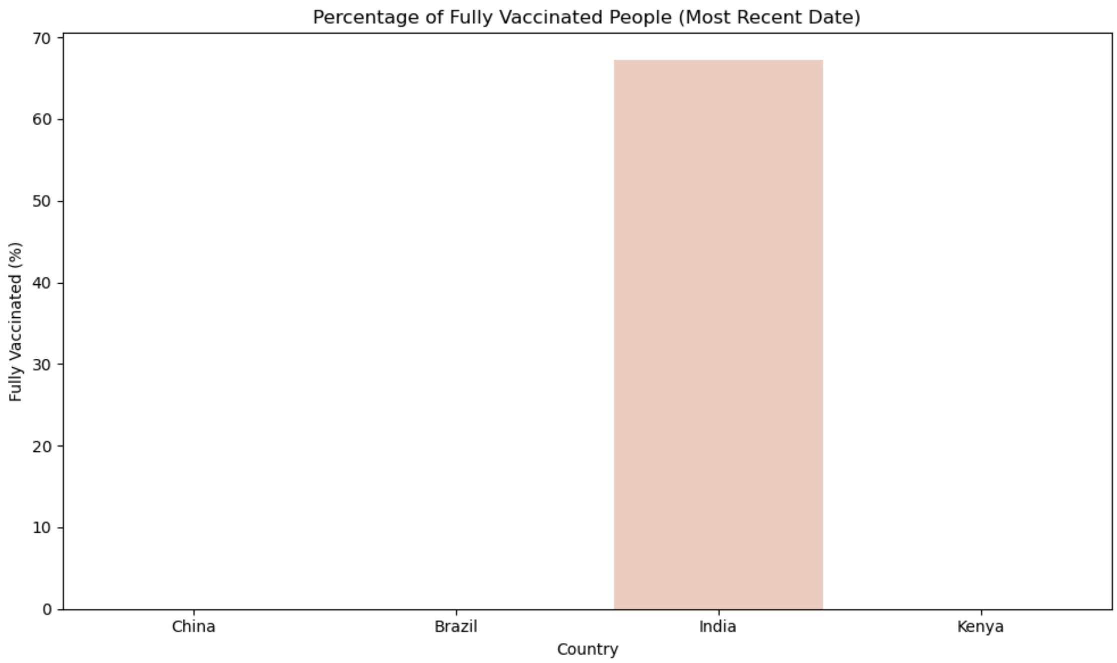


```
[17]: # Visualizing Vaccination Progress
      # cumulative vaccinations over time for selected countries
      plt.figure(figsize=(12, 6))

      for country in countries:
          country_data = df_filtered[df_filtered['location'] == country]
          plt.plot(country_data['date'], country_data['total_vaccinations'],␣
       ↪label=country)

      plt.title('Total COVID-19 Vaccinations Over Time')
      plt.xlabel('Date')
      plt.ylabel('Total Vaccinations')
      plt.xticks(rotation=45)
      plt.legend()
      plt.tight_layout()
      plt.show()
```



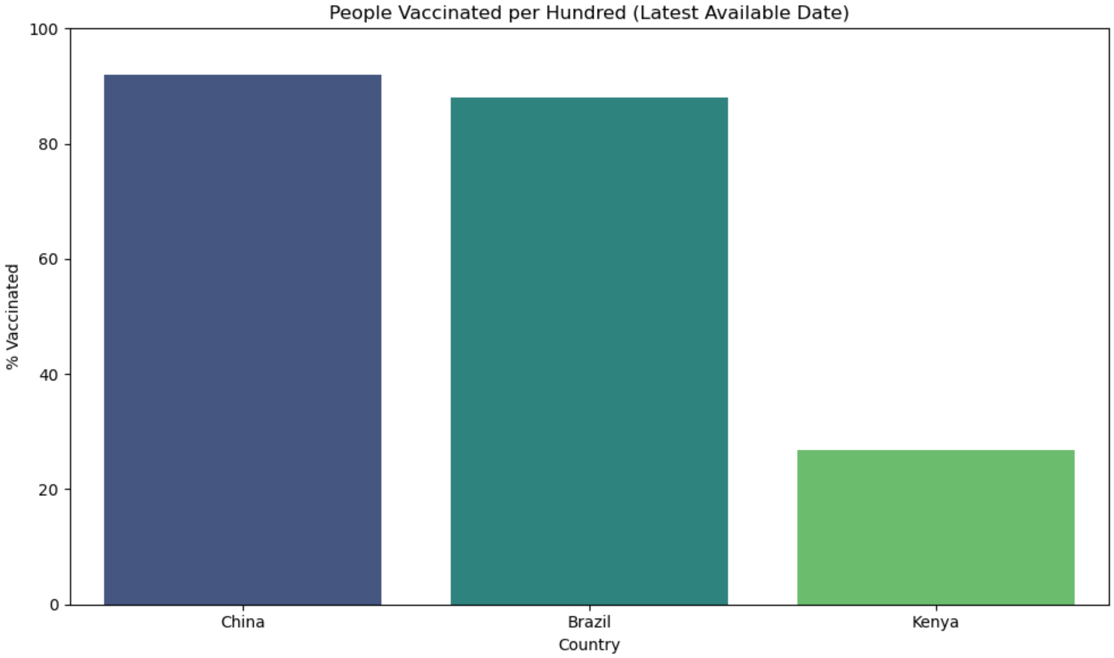
[18]:

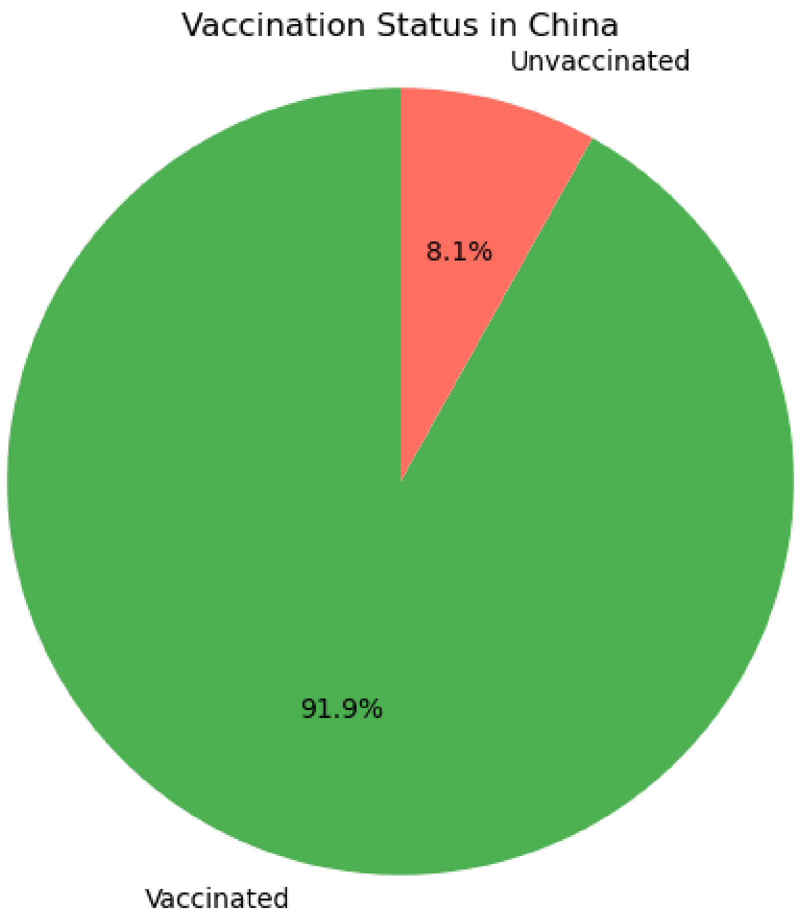
```
# Get the most recent data for each country
latest_vax = df_filtered.sort_values('date').groupby('location').tail(1)

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

# Plot a bar chart
plt.figure(figsize=(10, 6))
sns.barplot(x='location', y='people_fully_vaccinated_per_hundred',␣
  ↪data=latest_vax, palette='coolwarm')

plt.title('Percentage of Fully Vaccinated People (Most Recent Date)')
plt.ylabel('Fully Vaccinated (%)')
plt.xlabel('Country')
plt.tight_layout()
plt.show()
```

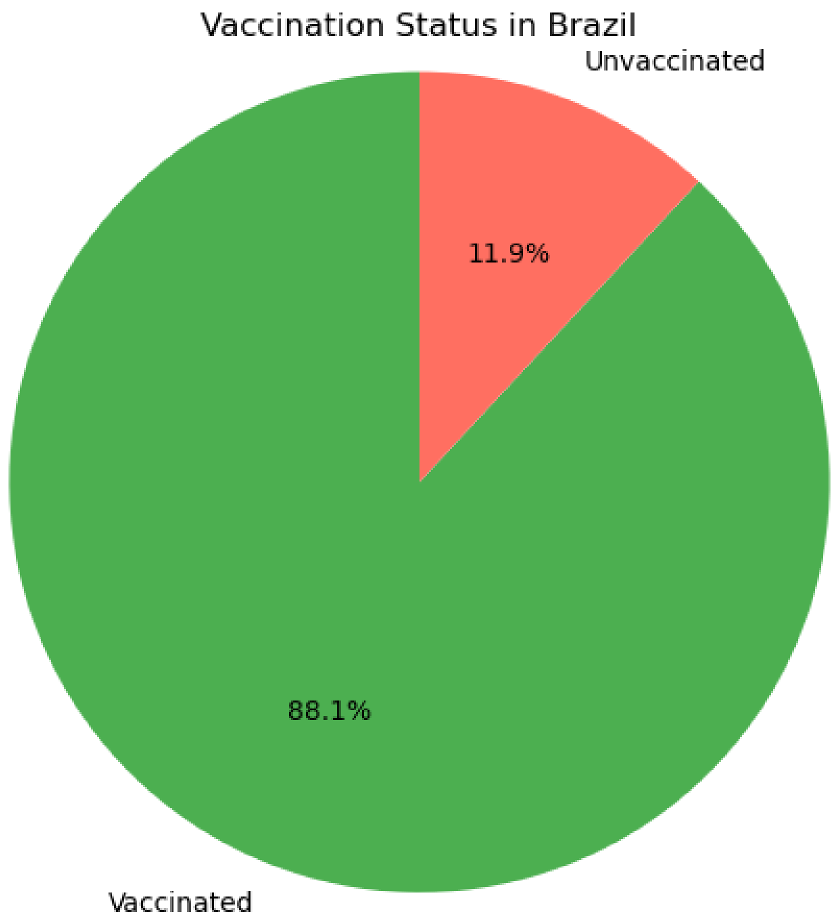


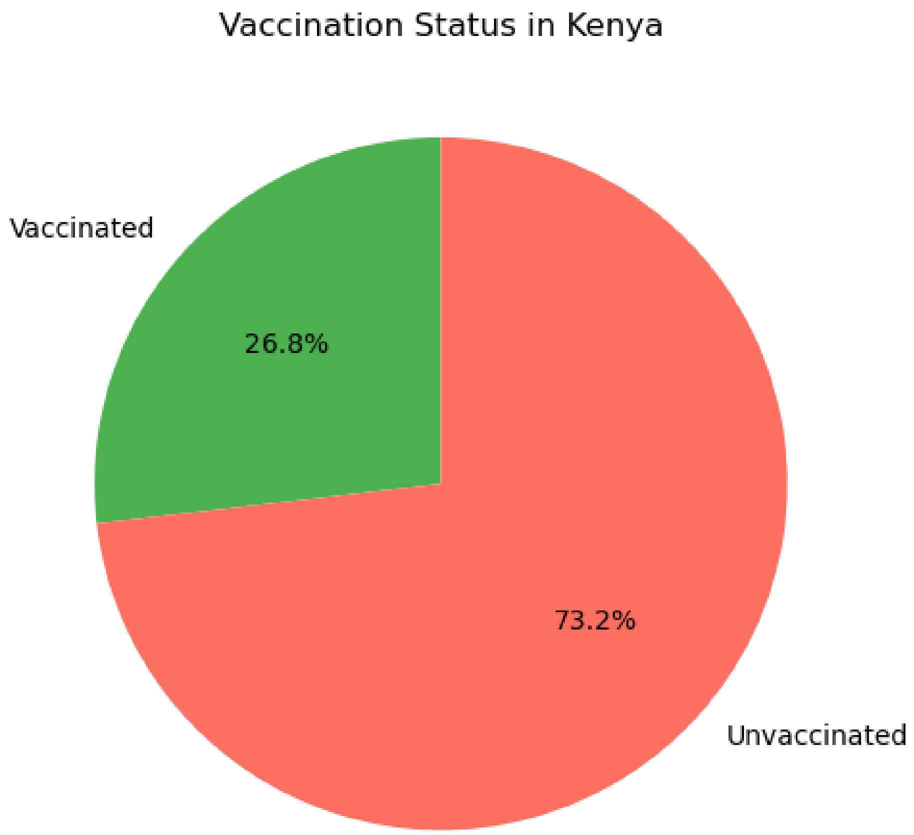
[21]:
```
# Selected countries
countries = ['Kenya', 'United States', 'Brazil', 'South Africa', 'China']

# Filter dataset to include only selected countries
vaccination_data = df_filtered[df_filtered['location'].isin(countries)]

# Drop rows where vaccination % is missing
vaccination_data = vaccination_data.
 ↪dropna(subset=['people_vaccinated_per_hundred'])

# Get the latest available record with vaccination data for each country
latest_vax = vaccination_data.sort_values('date').groupby('location').tail(1)

# ==== 1. BAR CHART: % Vaccinated per Country ====
plt.figure(figsize=(10, 6))
sns.barplot(x='location', y='people_vaccinated_per_hundred', data=latest_vax,␣
 ↪palette='viridis')

plt.title('People Vaccinated per Hundred (Latest Available Date)')
plt.ylabel('% Vaccinated')
plt.xlabel('Country')
plt.ylim(0, 100)
plt.tight_layout()
plt.show()
```

```python
# ==== 2. PIE CHARTS: Vaccinated vs Unvaccinated ====
for index, row in latest_vax.iterrows():
    vaccinated = row['people_vaccinated_per_hundred']
    unvaccinated = 100 - vaccinated
    labels = ['Vaccinated', 'Unvaccinated']
    sizes = [vaccinated, unvaccinated]
    colors = ['#4CAF50', '#FF6F61']

    plt.figure(figsize=(5, 5))
    plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)
    plt.title(f'Vaccination Status in {row["location"]}')
    plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
    plt.tight_layout()
    plt.show()
```




Vaccination Status in China
Unvaccinated
8.1%
91.9%
Vaccinated


Vaccination Status in Brazil
Unvaccinated
11.9%
88.1%
Vaccinated



[6]:

```python
import pandas as pd

# Loading the dataset
df = pd.read_csv('owid-covid-data.csv', parse_dates=['date'])
```

[17]:

```python
import plotly.express as px
import plotly.io as pio

# Use a safe renderer for Jupyter
pio.renderers.default = 'iframe'

# Plot the choropleth map
fig = px.choropleth(
    latest_vax,
    locations='iso_code',
    color='people_fully_vaccinated_per_hundred',
    hover_name='location',
    color_continuous_scale='Viridis',
    title='Vaccination Rates (% Fully Vaccinated) in Selected Countries',
    labels={'people_fully_vaccinated_per_hundred': '% Fully Vaccinated'},
    projection='natural earth'
)

fig.show()
```

1 COVID-19 Global Data Tracker

Analyzing COVID-19 cases, deaths, and vaccinations across six countries (2020–2025)

1.1 Data Source

- Dataset: Our World in Data – COVID-19 Dataset
- Format: CSV (`owid-covid-data.csv`)

1.2 Selected Countries for Analysis

- Kenya
- United States
- Brazil
- South Africa
- China
- India

1.3 Insights & Summary

1.3.1 1. COVID-19 Case Trends

- The United States and India recorded the highest number of total COVID-19 cases.
- Kenya, South Africa, and China reported significantly fewer total cases over time.

1.3.2 2. Death Rates

- Brazil exhibited a higher death rate, especially in the early stages of the pandemic.
- India and China maintained relatively lower death rates.

1.3.3 3. Vaccination Progress

- Brazil led the group in vaccination rollout based on available data.
- Kenya and South Africa had slower vaccine uptake, and some data were incomplete or unavailable.

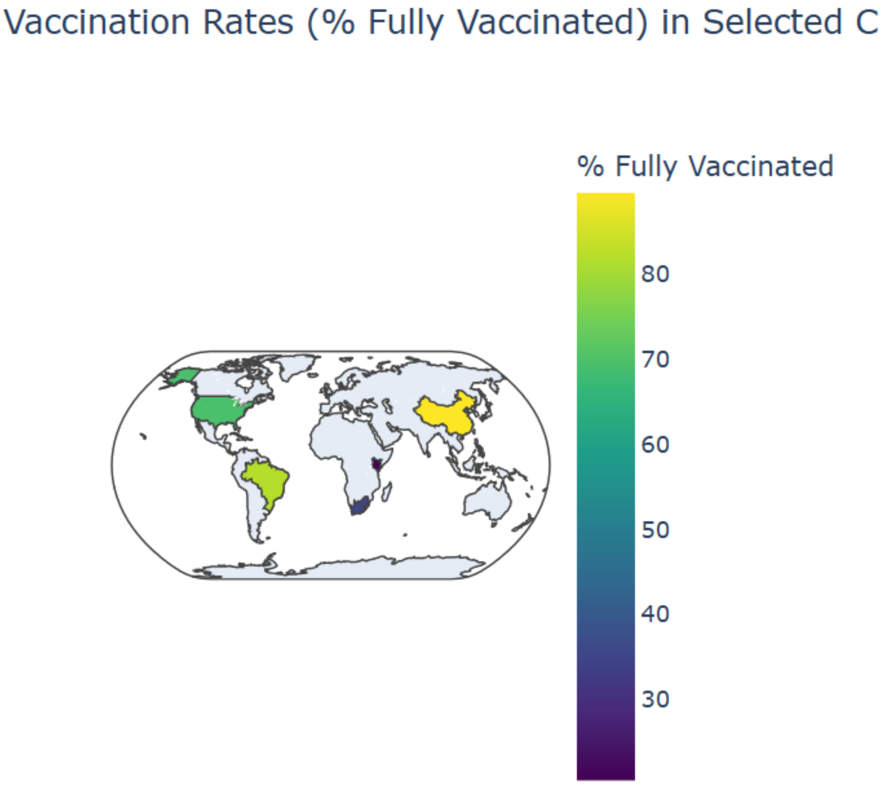
1.3.4 4. Daily New Cases

- All countries experienced waves of infections at different times.
- The United States showed the most dramatic and frequent spikes.

1.3.5 5. Data Limitations & Gaps

- Vaccination data were missing or inconsistent for some countries, like Kenya and South Africa.
- Death rate metrics may be affected by delayed or incomplete reporting.
- Choropleth maps were created but excluded from PDF export due to technical limitations.

1.4 Choropleth Map (Static Image)



1.5 Final Deliverables Included

- Filtered and cleaned dataset for six countries
- Visualizations: total cases, deaths, new cases, and death rate
- Vaccination progress: line chart, bar chart, and pie charts
- Choropleth map (static image embedded)
- Written narrative and insights using Markdown

1.6 Tools Used

- Jupyter Notebook
- Python (pandas, matplotlib, seaborn, plotly)
- Plotly Express for interactive mapping
- Markdown for documentation and reporting

[ ]: