COVID-19_Global data Analysis

May 16, 2025

[ ]:
```
COVID-19 Global Data Tracker
A Comprehensive Analysis of Cases, Deaths, and Vaccinations Worldwide

INTRODUCTION
The COVID-19 pandemic has had a profound impact on global health, economies,␣
 ↪and societies. Understanding its spread, vaccination progress, and mortality␣
 ↪trends is crucial for policymakers, researchers, and the public.
This project, COVID-19 Global Data Tracker, leverages Python and data␣
 ↪visualization tools to analyze worldwide COVID-19 data. We explore infection␣
 ↪rates, fatalities, vaccination trends, and regional comparisons to uncover␣
 ↪key insights.

By the end of this project, we aim to:
▯ Identify high-risk regions and trends in cases/deaths.
▯ Compare vaccination progress across countries.
▯ Visualize data interactively for better comprehension.

Project Description
The COVID-19 Global Data Tracker is a data-driven Python project that involves␣
 ↪importing, cleaning, analyzing, and visualizing COVID-19 data on cases,␣
 ↪deaths, and vaccinations globally. Using tools like pandas, matplotlib,␣
 ↪seaborn, and plotly, this project enables us to explore trends, compare␣
 ↪statistics among countries, and communicate key findings through␣
 ↪visualizations and narrative insights.
The final output is presented in a comprehensive Jupyter Notebook, combining␣
 ↪code, plots, and textual explanations.
```

[25]:
```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
```

[10]:
```
# Load dataset
df = pd.read_csv("owid-covid-data.csv")
```

```
df.columns
```

```
[10]: 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')
```

```
[26]: # Preview data
      df.head()
```

```
[26]:     iso_code continent     location       date  total_cases  new_cases  \
      416      AFG      Asia  Afghanistan 2021-02-22      55617.0       13.0
      422      AFG      Asia  Afghanistan 2021-02-28      55714.0        7.0
      438      AFG      Asia  Afghanistan 2021-03-16      55995.0       10.0
      460      AFG      Asia  Afghanistan 2021-04-07      56873.0       94.0
      475      AFG      Asia  Afghanistan 2021-04-22      58312.0       98.0

           new_cases_smoothed  total_deaths  new_deaths  new_deaths_smoothed  …  \
      416              14.714        2433.0         1.0                0.857  …
      422              15.714        2443.0         0.0                1.571  …
      438              17.000        2460.0         1.0                1.286  …
      460              59.857        2512.0         0.0                4.000  …
      475             111.143        2561.0         4.0                4.000  …
```

```
     handwashing_facilities  hospital_beds_per_thousand  life_expectancy  \
416                  37.746                         0.5            64.83
422                  37.746                         0.5            64.83
438                  37.746                         0.5            64.83
460                  37.746                         0.5            64.83
475                  37.746                         0.5            64.83

     human_development_index  population  \
416                    0.511  41128772.0
422                    0.511  41128772.0
438                    0.511  41128772.0
460                    0.511  41128772.0
475                    0.511  41128772.0

     excess_mortality_cumulative_absolute  excess_mortality_cumulative  \
416                                   NaN                          NaN
422                                   NaN                          NaN
438                                   NaN                          NaN
460                                   NaN                          NaN
475                                   NaN                          NaN

     excess_mortality  excess_mortality_cumulative_per_million    month
416               NaN                                      NaN  2021-02
422               NaN                                      NaN  2021-02
438               NaN                                      NaN  2021-03
460               NaN                                      NaN  2021-04
475               NaN                                      NaN  2021-04

[5 rows x 68 columns]
```

[27]:
```
df.tail(10)
```

[27]:
```
       iso_code continent  location        date  total_cases  new_cases  \
247190      SVN    Europe  Slovenia  2022-02-22     882805.0     2337.0
247191      SVN    Europe  Slovenia  2022-02-23     885208.0     2403.0
247192      SVN    Europe  Slovenia  2022-02-24     887581.0     2373.0
247193      SVN    Europe  Slovenia  2022-02-25     889518.0     1937.0
247194      SVN    Europe  Slovenia  2022-02-26     891346.0     1828.0
247195      SVN    Europe  Slovenia  2022-02-27     892275.0      929.0
247196      SVN    Europe  Slovenia  2022-02-28     892955.0      680.0
247197      SVN    Europe  Slovenia  2022-03-01     895514.0     2559.0
247198      SVN    Europe  Slovenia  2022-03-02     897383.0     1869.0
247199      SVN    Europe  Slovenia  2022-03-03     899246.0     1863.0

        new_cases_smoothed  total_deaths  new_deaths  new_deaths_smoothed  \
247190            3373.000        7003.0        25.0               25.571
```

```
247191           3037.714       7028.0       25.0               25.143
247192           2722.000       7043.0       15.0               23.857
247193           2449.857       7062.0       19.0               22.571
247194           2173.143       7087.0       25.0               23.000
247195           1977.714       7110.0       23.0               22.714
247196           1783.857       7129.0       19.0               21.571
247197           1815.571       7146.0       17.0               20.429
247198           1739.286       7159.0       13.0               18.714
247199           1666.429       7171.0       12.0               18.286

        …  handwashing_facilities  hospital_beds_per_thousand  \
247190  …                     NaN                         4.5
247191  …                     NaN                         4.5
247192  …                     NaN                         4.5
247193  …                     NaN                         4.5
247194  …                     NaN                         4.5
247195  …                     NaN                         4.5
247196  …                     NaN                         4.5
247197  …                     NaN                         4.5
247198  …                     NaN                         4.5
247199  …                     NaN                         4.5

        life_expectancy  human_development_index  population  \
247190            81.32                    0.917   2119843.0
247191            81.32                    0.917   2119843.0
247192            81.32                    0.917   2119843.0
247193            81.32                    0.917   2119843.0
247194            81.32                    0.917   2119843.0
247195            81.32                    0.917   2119843.0
247196            81.32                    0.917   2119843.0
247197            81.32                    0.917   2119843.0
247198            81.32                    0.917   2119843.0
247199            81.32                    0.917   2119843.0

        excess_mortality_cumulative_absolute  excess_mortality_cumulative  \
247190                                   NaN                          NaN
247191                                   NaN                          NaN
247192                                   NaN                          NaN
247193                                   NaN                          NaN
247194                                   NaN                          NaN
247195                             5287.5005                        11.42
247196                                   NaN                          NaN
247197                                   NaN                          NaN
247198                                   NaN                          NaN
247199                                   NaN                          NaN

        excess_mortality  excess_mortality_cumulative_per_million    month
```

```
247190            NaN                              NaN  2022-02
247191            NaN                              NaN  2022-02
247192            NaN                              NaN  2022-02
247193            NaN                              NaN  2022-02
247194            NaN                              NaN  2022-02
247195           2.37                        2494.2888  2022-02
247196            NaN                              NaN  2022-02
247197            NaN                              NaN  2022-03
247198            NaN                              NaN  2022-03
247199            NaN                              NaN  2022-03

[10 rows x 68 columns]
```

[28]:
```
# Check missing values
missing_values = df.isnull().sum()
print(df.isnull().sum()[missing_values > 0])
```

```
continent                                   7509
reproduction_rate                          12653
icu_patients                               43739
icu_patients_per_million                   43739
hosp_patients                              43098
hosp_patients_per_million                  43098
weekly_icu_admissions                      49905
weekly_icu_admissions_per_million          49905
weekly_hosp_admissions                     47859
weekly_hosp_admissions_per_million         47859
total_tests                                32775
new_tests                                  34194
total_tests_per_thousand                   32775
new_tests_per_thousand                     34194
new_tests_smoothed                         25891
new_tests_smoothed_per_thousand            25891
positive_rate                              28239
tests_per_case                             28300
tests_units                                25556
people_vaccinated                           3302
people_fully_vaccinated                     5044
total_boosters                             24899
new_vaccinations                           10573
new_vaccinations_smoothed                    207
people_vaccinated_per_hundred               3302
people_fully_vaccinated_per_hundred         5044
total_boosters_per_hundred                 24899
new_vaccinations_smoothed_per_million        207
new_people_vaccinated_smoothed               519
new_people_vaccinated_smoothed_per_hundred   519
```

```
stringency_index                           13885
population_density                          7859
median_age                                 10088
aged_65_older                              10088
aged_70_older                              10325
gdp_per_capita                             10329
extreme_poverty                            24198
cardiovasc_death_rate                      10995
diabetes_prevalence                         9057
female_smokers                             15286
male_smokers                               16169
handwashing_facilities                     40407
hospital_beds_per_thousand                 12332
life_expectancy                             8221
human_development_index                    10581
excess_mortality_cumulative_absolute       52284
excess_mortality_cumulative                52284
excess_mortality                           52284
excess_mortality_cumulative_per_million    52284
dtype: int64
```

[16]:
```
# Define key columns
key_columns = ["date", "location", "total_cases", "total_deaths", "new_cases",␣
↪"new_deaths", "total_vaccinations"]

# Check for missing columns
missing_cols = [col for col in key_columns if col not in df.columns]

if missing_cols:
    print(f"Warning: These columns are missing: {missing_cols}")
else:
    print("All key columns are present!")
```

```
All key columns are present!
```

[14]:
```
# Filter the DataFrame for the Some of East African Countries
east_africa = ['Burundi', 'Democratic Republic of Congo', 'Kenya', 'Rwanda']

df_ea = df[df['location'].isin(east_africa)]

# Generate the statistical summary
summary = df_ea[['location', 'total_cases', 'total_deaths']].
↪groupby('location').describe()

# Display the summary
print(summary)
```

```
                             total_cases                                 \
                                count           mean            std  min
location
Burundi                        1107.0   22694.196929   21218.369972  2.0
Democratic Republic of Congo   1128.0   53444.719858   35751.624716  1.0
Kenya                          1125.0  206783.684444  127844.022215  1.0
Rwanda                         1124.0   73045.148577   56697.583861  1.0

                                                                    \
                                   25%       50%       75%       max
location
Burundi                         829.50   18972.0   42963.0   53719.0
Democratic Republic of Congo  15179.25   56915.0   91740.5   95944.0
Kenya                         94151.00  248461.0  334551.0  342992.0
Rwanda                         7277.75   96910.5  131308.0  133194.0

                             total_deaths                                \
                                    count         mean          std    min
location
Burundi                            1095.0     9.452055     6.020018    1.0
Democratic Republic of Congo       1093.0   995.881061   413.077670  209.0
Kenya                              1112.0  3700.362410  2185.599406    4.0
Rwanda                             1047.0   901.996180   621.448193    1.0


                                  25%     50%     75%     max
location
Burundi                          2.00    14.0    15.0    15.0
Democratic Republic of Congo   627.00  1091.0  1390.0  1464.0
Kenya                         1664.75  5135.5  5660.0  5688.0
Rwanda                         239.50  1332.0  1466.0  1468.0
```

[15]:
```python
# Drop rows with missing dates or important fields
df = df.dropna(subset=["date", "location", "total_cases", "total_deaths",␣
 ↪"new_cases", "new_deaths", "total_vaccinations"]
)
```

[18]:
```python
# Define numeric_columns first by selecting numeric columns from the DataFrame
numeric_columns = df.select_dtypes(include=['number']).columns

# Check for minimum in numeric columns
min_values = df[numeric_columns].min().sort_values(ascending=True)

print(f"Minimum values:\n{min_values.head(15)}")
```

```
Minimum values:
excess_mortality_cumulative_absolute      -37726.0980
excess_mortality_cumulative_per_million    -1693.2815
excess_mortality                             -66.4000
```

```
excess_mortality_cumulative              -12.9900
reproduction_rate                         -0.0200
new_tests_smoothed_per_thousand            0.0000
positive_rate                              0.0000
total_vaccinations                         0.0000
new_vaccinations                           0.0000
new_vaccinations_smoothed                  0.0000
people_vaccinated_per_hundred              0.0000
new_tests_smoothed                         0.0000
people_fully_vaccinated_per_hundred        0.0000
total_boosters_per_hundred                 0.0000
new_vaccinations_smoothed_per_million      0.0000
dtype: float64
```

[ ]:
```
The dataset such as new_tests, new_cases, new_cases_smoothed, and new_deaths,␣
↪that shows negative minimum values.
Given the nature of COVID-19 data, negative entries in these fields are␣
↪logically implausible (e.g., newly reported cases or deaths cannot␣
↪meaningfully be negative).
It is resolved by replacing negative entries with their absolute values to␣
↪ensure data consistency.
```

[19]:
```
# Replace negative values with their absolute values in all numeric columns
df[numeric_columns] = df[numeric_columns].abs()
```

[20]:
```
new_min_values = df[numeric_columns].min().sort_values(ascending=True)

print(f"New minimum values:\n{new_min_values.head(15)}")
```

```
New minimum values:
people_vaccinated                          0.0
weekly_hosp_admissions_per_million         0.0
stringency_index                           0.0
new_people_vaccinated_smoothed_per_hundred    0.0
new_tests_smoothed                         0.0
new_tests_smoothed_per_thousand            0.0
positive_rate                              0.0
weekly_hosp_admissions                     0.0
new_people_vaccinated_smoothed             0.0
excess_mortality                           0.0
new_vaccinations_smoothed_per_million      0.0
total_boosters_per_hundred                 0.0
new_vaccinations                           0.0
new_vaccinations_smoothed                  0.0
total_vaccinations_per_hundred             0.0
dtype: float64
```

```
[ ]: # Relace NaN values with 0 in all numeric columns
     df[numeric_columns] = df[numeric_columns].fillna(0)
```

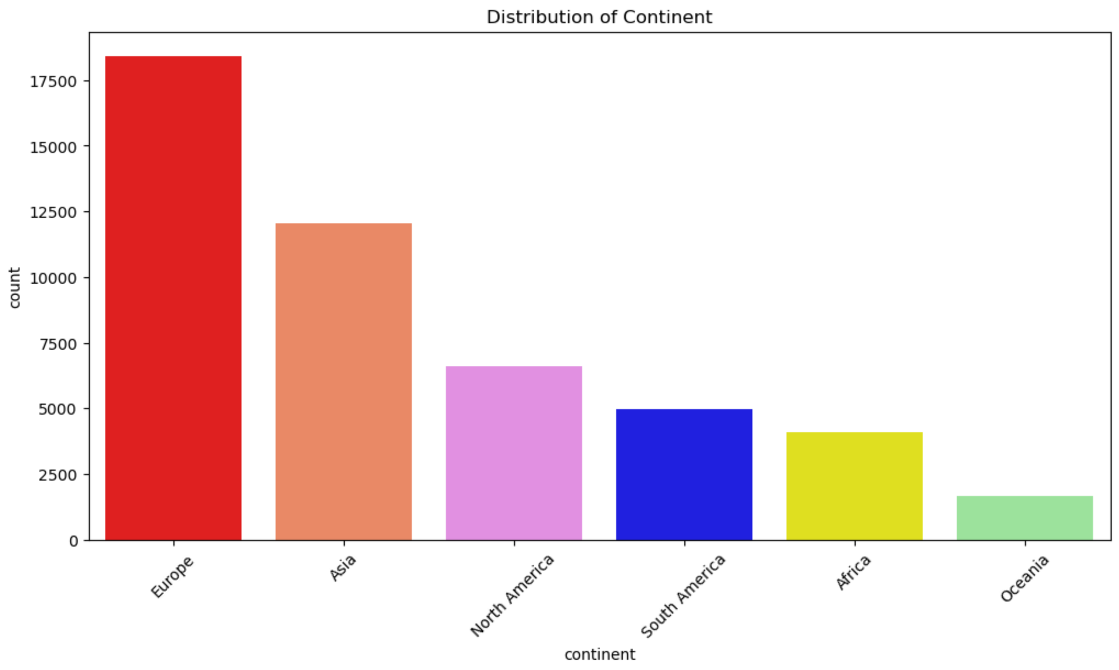
```
[21]: # Check for null values
      numeric_columns = df.select_dtypes(include=["number"]).columns.to_list()
      df[numeric_columns].isnull().sum()
```

```
[21]: total_cases                                0
      new_cases                                  0
      new_cases_smoothed                         0
      total_deaths                               0
      new_deaths                                 0
                                              …
      population                                 0
      excess_mortality_cumulative_absolute   52284
      excess_mortality_cumulative            52284
      excess_mortality                       52284
      excess_mortality_cumulative_per_million 52284
      Length: 62, dtype: int64
```

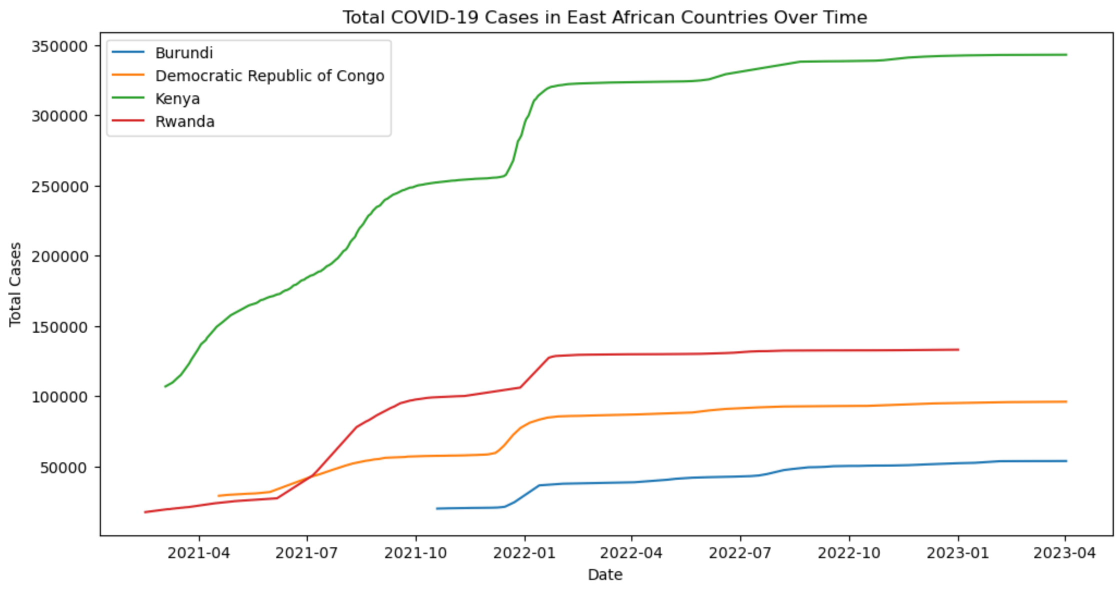
```
[23]: # Check for unique values in the 'continent' column
      df['continent'].value_counts()
```

```
[23]: continent
      Europe           18423
      Asia             12061
      North America     6623
      South America     4987
      Africa            4086
      Oceania           1677
      Name: count, dtype: int64
```

```
[29]: # Plotting the distribution of the continent column
      plt.figure(figsize=(12, 6))
      sns.countplot(data=df, x='continent', order=df['continent'].value_counts().
        ↪index,
                    palette=['red','coral','violet','blue','yellow','lightgreen'])
      plt.title('Distribution of Continent')
      plt.xticks(rotation=45)
      plt.show()
```

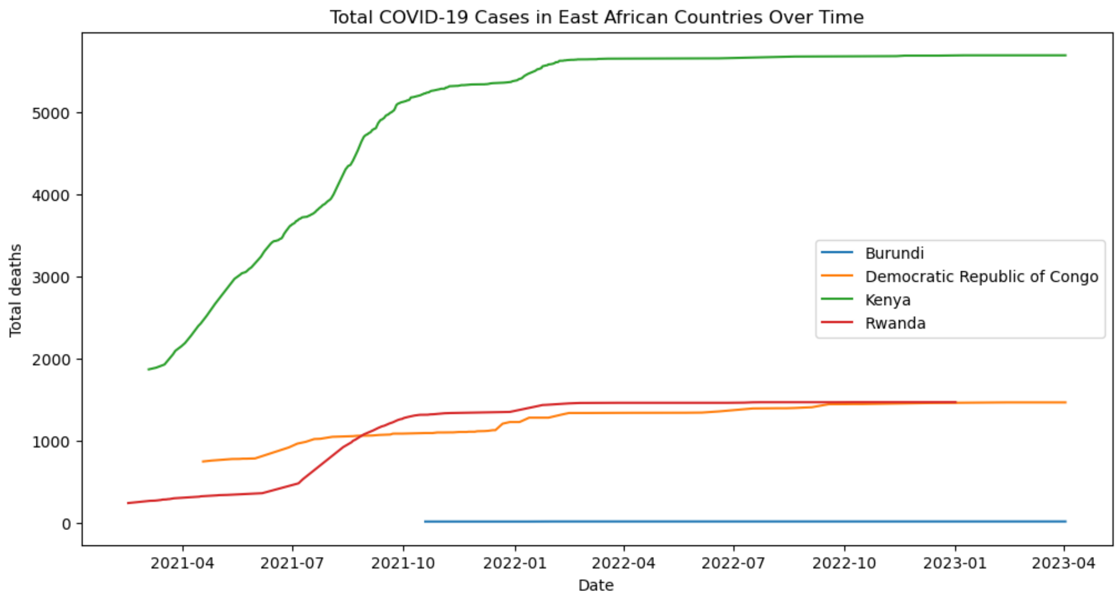


```
[30]: # Total cases in Some of East African Countries over time
east_africa = ['Burundi', 'Democratic Republic of Congo', 'Kenya', 'Rwanda']
plt.figure(figsize=(12, 6))
for country in east_africa:
    east_africa_data = df[df['location'] == country]  # Filter data for each␣
 ↪country
    plt.plot(east_africa_data['date'], east_africa_data['total_cases'],␣
 ↪label=country)  # This line needed indentation

plt.legend(title="Country")
plt.legend()
plt.title("Total COVID-19 Cases in East African Countries Over Time")
plt.xlabel("Date")
plt.ylabel("Total Cases")
plt.show()
```



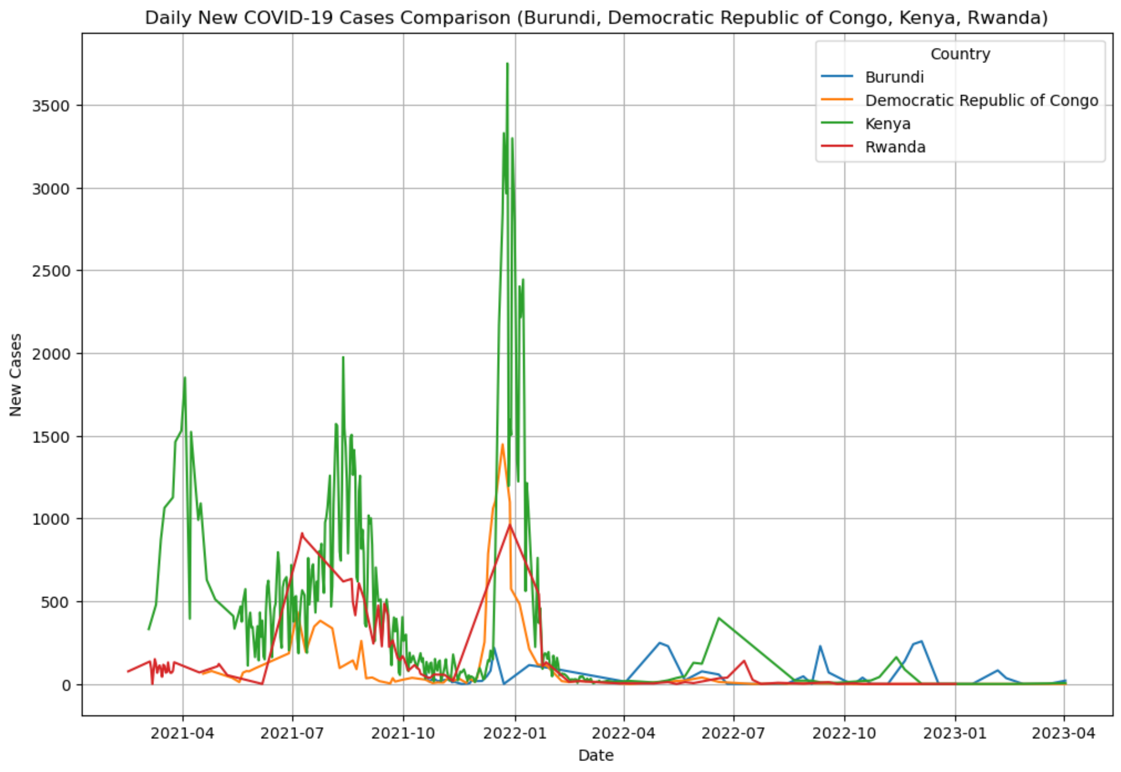
```
[ ]: TOTAL DEATHS OVER TIME (SOME OF EAST AFRICA COUNTRIES)
```

```
[31]: # Total deaths in Some of East African Countries over time
east_africa = ['Burundi', 'Democratic Republic of Congo', 'Kenya', 'Rwanda']
plt.figure(figsize=(12, 6))
for country in east_africa:
    east_africa_data = df[df['location'] == country]  # Filter data for each␣
 ↪country
    plt.plot(east_africa_data['date'], east_africa_data['total_deaths'],␣
 ↪label=country)  # This line needed indentation

plt.legend(title="Country")
plt.legend()
plt.title("Total COVID-19 Cases in East African Countries Over Time")
plt.xlabel("Date")
plt.ylabel("Total deaths")
plt.show()
```



```
[ ]: DAILY NEW COVID 19 CASES COMPARISON-EAST AFRICA
```

```
[32]: plt.figure(figsize=(12, 8))
      for country in east_africa:
          east_africa_data = df[df['location'] == country]  # Filter data for each␣
       ↪country
          plt.plot(east_africa_data['date'], east_africa_data['new_cases'],␣
       ↪label=country)  # Fixed indentation here

      # Customize the plot
      plt.title("Daily New COVID-19 Cases Comparison (Burundi, Democratic Republic of␣
       ↪Congo, Kenya, Rwanda)")
      plt.xlabel("Date")
      plt.ylabel("New Cases")
      plt.legend(title="Country")
      plt.grid(True)

      # Show the plot
      plt.show()
```



```
[ ]: ANALYSING COVID PREVELANCE, WORLDWIDE AND EAST AFRICA
```

```
[33]: east_africa = ['Burundi', 'Democratic Republic of Congo', 'Kenya', 'Rwanda']

      df_ea = df[df['location'].isin(east_africa)].copy()

      # Calculate death rate
      df_ea["death_rate"] = df_ea["total_deaths"] / df_ea["total_cases"]
      print(df_ea.pivot_table(index="date", columns="location", values="death_rate"))
```

```
location     Burundi  Democratic Republic of Congo     Kenya    Rwanda
date
2021-02-15       NaN                           NaN       NaN  0.013781
2021-03-04       NaN                           NaN  0.017472       NaN
2021-03-05       NaN                           NaN       NaN  0.013758
2021-03-06       NaN                           NaN       NaN  0.013744
2021-03-07       NaN                           NaN       NaN  0.013744
…              …                           …       …       …
2023-02-26  0.000280                           NaN       NaN       NaN
2023-03-05  0.000280                      0.015285       NaN       NaN
2023-03-19  0.000279                      0.015272       NaN       NaN
2023-03-26       NaN                      0.015267       NaN       NaN
2023-04-02  0.000279                      0.015259  0.016584       NaN
```

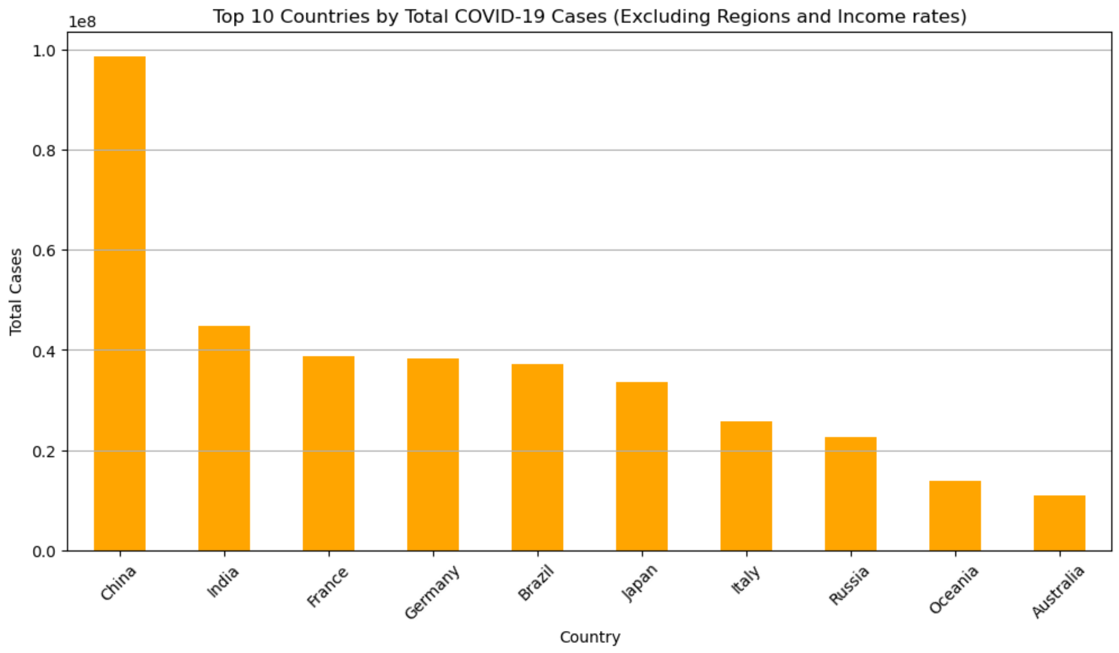
```
[357 rows x 4 columns]
```

[34]:
```
# Get all unique location names
unique_locations = df["location"].unique()
print(unique_locations)  # Displays ALL entries
```

```
['Afghanistan' 'Africa' 'Albania' 'Algeria' 'Andorra' 'Angola' 'Anguilla'
 'Antigua and Barbuda' 'Argentina' 'Armenia' 'Aruba' 'Asia' 'Australia'
 'Austria' 'Azerbaijan' 'Bahamas' 'Bahrain' 'Bangladesh' 'Barbados'
 'Belarus' 'Belgium' 'Belize' 'Benin' 'Bermuda' 'Bhutan' 'Bolivia'
 'Bonaire Sint Eustatius and Saba' 'Bosnia and Herzegovina' 'Botswana'
 'Brazil' 'British Virgin Islands' 'Brunei' 'Bulgaria' 'Burkina Faso'
 'Burundi' 'Cambodia' 'Cameroon' 'Canada' 'Cape Verde' 'Cayman Islands'
 'Central African Republic' 'Chad' 'Chile' 'China' 'Colombia' 'Comoros'
 'Congo' 'Cook Islands' 'Costa Rica' "Cote d'Ivoire" 'Croatia' 'Cuba'
 'Curacao' 'Cyprus' 'Czechia' 'Democratic Republic of Congo' 'Denmark'
 'Djibouti' 'Dominica' 'Dominican Republic' 'Ecuador' 'Egypt'
 'El Salvador' 'Equatorial Guinea' 'Estonia' 'Eswatini' 'Ethiopia'
 'Europe' 'European Union' 'Faeroe Islands' 'Fiji' 'Finland' 'France'
 'French Polynesia' 'Gabon' 'Gambia' 'Georgia' 'Germany' 'Ghana'
 'Gibraltar' 'Greece' 'Greenland' 'Grenada' 'Guatemala' 'Guernsey'
 'Guinea' 'Guinea-Bissau' 'Guyana' 'Haiti' 'High income' 'Honduras'
 'Hungary' 'Iceland' 'India' 'Indonesia' 'Iran' 'Iraq' 'Ireland'
 'Isle of Man' 'Israel' 'Italy' 'Jamaica' 'Japan' 'Jersey' 'Jordan'
 'Kazakhstan' 'Kenya' 'Kiribati' 'Kosovo' 'Kuwait' 'Kyrgyzstan' 'Laos'
 'Latvia' 'Lebanon' 'Lesotho' 'Liberia' 'Libya' 'Liechtenstein'
 'Lithuania' 'Low income' 'Lower middle income' 'Luxembourg' 'Madagascar'
 'Malawi' 'Malaysia' 'Maldives' 'Mali' 'Malta' 'Mauritania' 'Mauritius'
 'Mexico' 'Moldova' 'Monaco' 'Mongolia' 'Montenegro' 'Montserrat'
 'Morocco' 'Mozambique' 'Myanmar' 'Namibia' 'Nauru' 'Nepal' 'Netherlands'
 'New Caledonia' 'New Zealand' 'Nicaragua' 'Niger' 'Nigeria'
 'North America' 'North Macedonia' 'Norway' 'Oceania' 'Oman' 'Pakistan'
 'Palestine' 'Panama' 'Papua New Guinea' 'Paraguay' 'Peru' 'Philippines'
 'Poland' 'Portugal' 'Qatar' 'Romania' 'Russia' 'Rwanda'
 'Saint Kitts and Nevis' 'Saint Lucia' 'Saint Vincent and the Grenadines'
 'Samoa' 'San Marino' 'Sao Tome and Principe' 'Saudi Arabia' 'Senegal'
 'Serbia' 'Seychelles' 'Sierra Leone' 'Singapore'
 'Sint Maarten (Dutch part)' 'Slovakia' 'Slovenia']
```

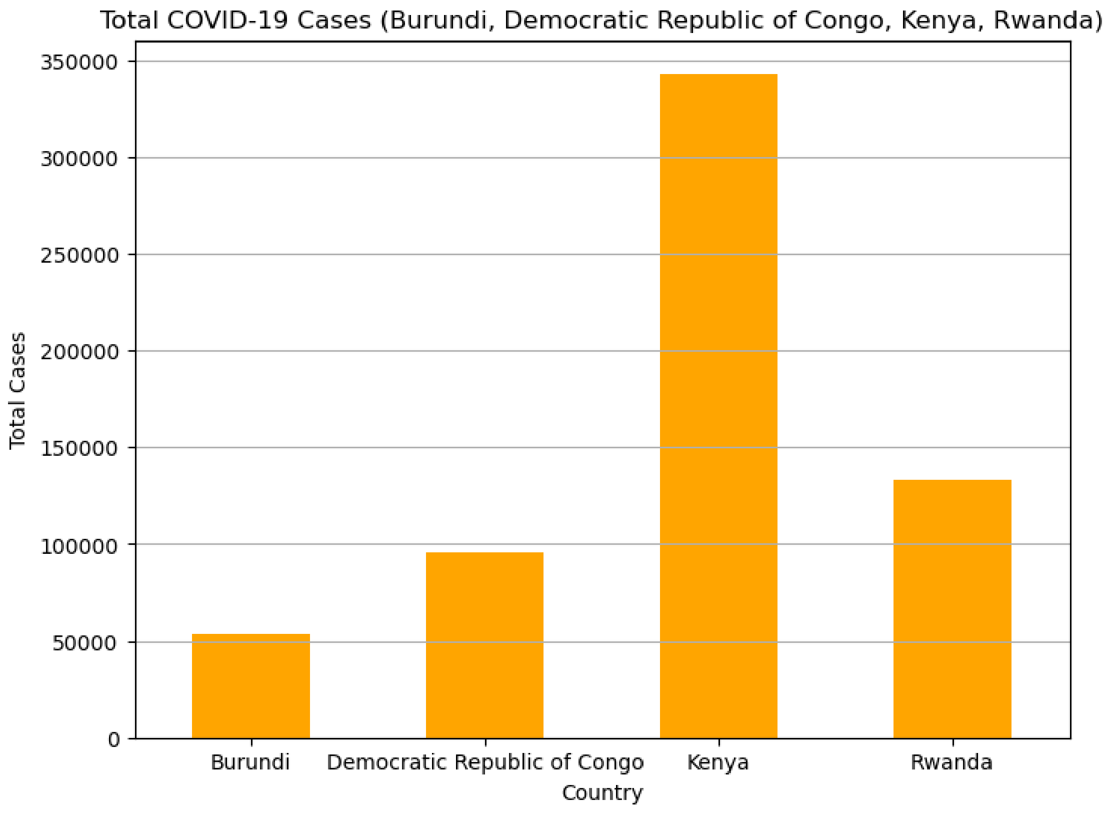
[35]:
```
# Exclude non-country entities
regions_to_exclude = ["World", "European Union", "Asia", "Europe", "Africa",␣
 ↪"North America", "South America",'High income','Upper middle income','Lower␣
 ↪middle income']
df_filtered = df[~df["location"].isin(regions_to_exclude)]

# Get top 10 countries by total cases
```

```
top_countries = df_filtered.groupby("location")["total_cases"].max().
 ↪nlargest(10)

# Plot bar chart
plt.figure(figsize=(12, 6))
top_countries.plot(kind="bar", color="Orange")

# Customize plot
plt.title("Top 10 Countries by Total COVID-19 Cases (Excluding Regions and␣
 ↪Income rates)")
plt.xlabel("Country")
plt.ylabel("Total Cases")
plt.xticks(rotation=45)
plt.grid(axis="y")

# Show plot
plt.show()
```



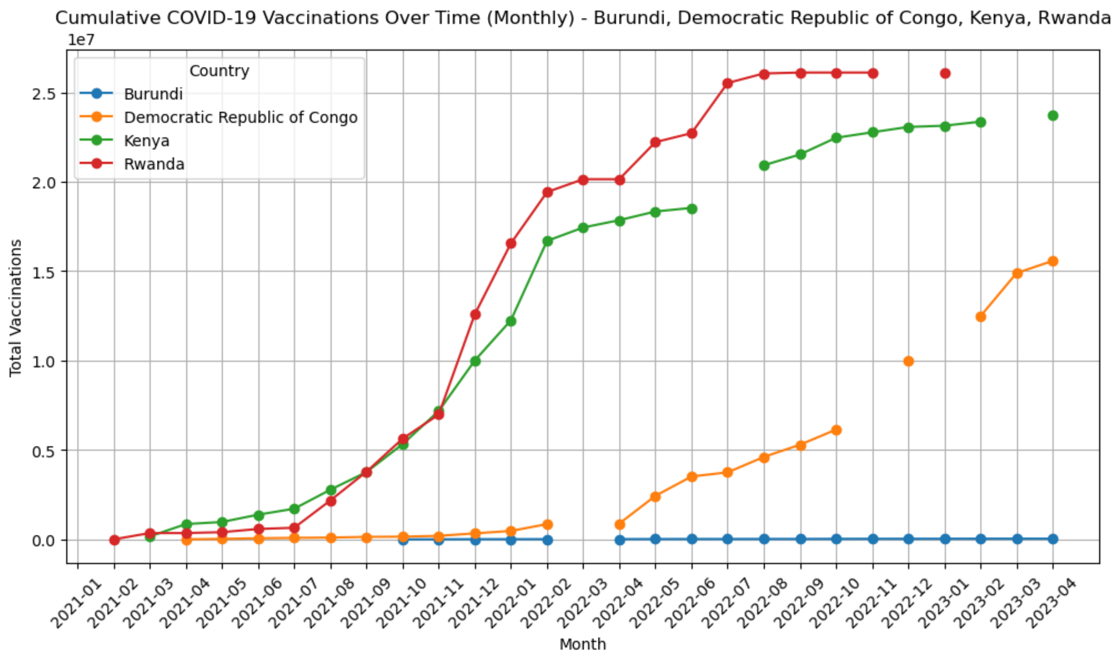
[36]:
```
# Filter for Some of East Africa Countries
east_africa = ['Burundi', 'Democratic Republic of Congo', 'Kenya', 'Rwanda']
df_ea = df[df['location'].isin(east_africa)].copy()

# Select latest available data
df_latest = df_ea.sort_values("date").groupby("location").last()
```

```
# Extract total cases
total_cases = df_latest["total_cases"]

# Plot bar chart
plt.figure(figsize=(8, 6))
total_cases.plot(kind="bar", color=["orange"])
# Customize plot
plt.title("Total COVID-19 Cases (Burundi, Democratic Republic of Congo, Kenya,␣
  ↪Rwanda)")
plt.xlabel("Country")
plt.ylabel("Total Cases")
plt.xticks(rotation=0)
plt.grid(axis="y")

# Show plot
plt.show()
```



[37]:
```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("owid-covid-data.csv")
# Ensure 'date' column is in datetime format
```

```
df["date"] = pd.to_datetime(df["date"], errors="coerce")

# Drop invalid dates
df = df.dropna(subset=["date"])

# Extract year-month for grouping
df["month"] = df["date"].dt.to_period("M")

# Filter for East Africa Countries
east_africa = ['Burundi', 'Democratic Republic of Congo', 'Kenya', 'Rwanda']
df_ea = df[df['location'].isin(east_africa)].copy()

# Group by month and country, taking the max vaccinations per month
df_grouped = df_ea.groupby(["month", "location"])["total_vaccinations"].max().
 ↪reset_index()

# Convert period to string for easy plotting
df_grouped["month"] = df_grouped["month"].astype(str)

# Plot cumulative vaccinations over months
plt.figure(figsize=(12, 6))
for country in east_africa:
    east_africa_data = df_grouped[df_grouped["location"] == country]
    plt.plot(east_africa_data["month"], east_africa_data["total_vaccinations"],␣
 ↪marker="o", linestyle="-", label=country)

# Customize plot
plt.title("Cumulative COVID-19 Vaccinations Over Time (Monthly) - Burundi,␣
 ↪Democratic Republic of Congo, Kenya, Rwanda")
plt.xlabel("Month")
plt.ylabel("Total Vaccinations")
plt.xticks(rotation=45)
plt.legend(title="Country")
plt.grid(True)

# Show plot
plt.show()
```

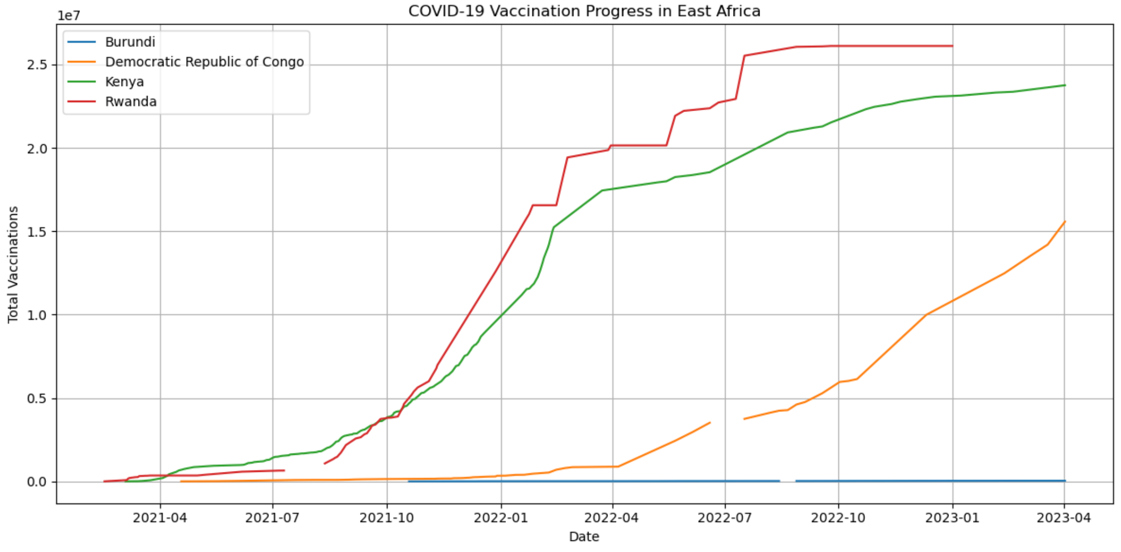


```
[38]: # Filter for East Africa Countries
      east_africa = ['Burundi', 'Democratic Republic of Congo', 'Kenya', 'Rwanda']
      df_ea = df[df['location'].isin(east_africa)]

      # Drop rows with missing vaccination or population data
      df_ea = df_ea.dropna(subset=['people_vaccinated', 'population'])

      # Get the latest data per country (assuming 'date' column is in datetime format)
      df_ea['date'] = pd.to_datetime(df_ea['date'])
      latest_vax = df_ea.sort_values('date').groupby('location').last()

      # Compute vaccinated individuals per population (as a percentage)
      latest_vax['vaccinated_per_population'] = (latest_vax['people_vaccinated'] /␣
       ↪latest_vax['population']) * 100

      # Select relevant columns for summary
      result = latest_vax[['people_vaccinated', 'population',␣
       ↪'vaccinated_per_population']]
      print(result)
```

```
                              people_vaccinated  population  \
location
Burundi                                 34323.0  12889583.0
Democratic Republic of Congo         14629322.0  99010216.0
Kenya                                14494372.0  54027484.0
Rwanda                               10572981.0  13776702.0
```

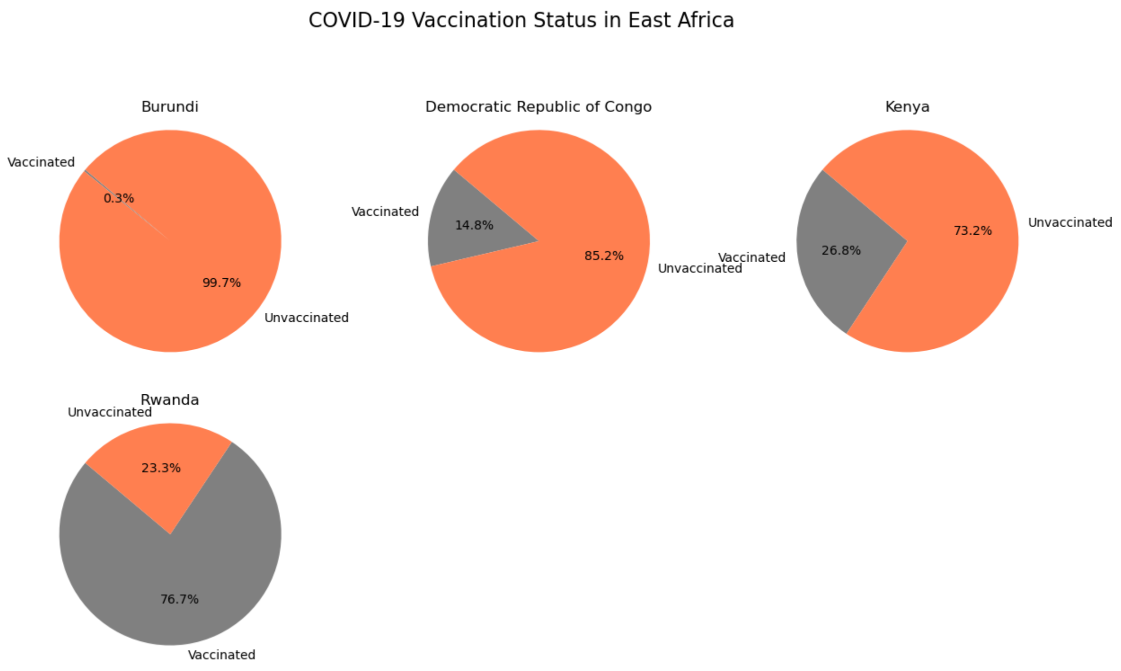
```
                                    vaccinated_per_population
location
Burundi                                              0.266285
Democratic Republic of Congo                        14.775568
Kenya                                               26.827775
Rwanda                                              76.745371
```

[49]:
```python
# Line chart for cumulative vaccinations over time
plt.figure(figsize=(12, 6))
for country in east_africa:
    country_data = df_ea[df_ea['location'] == country]
    plt.plot(country_data['date'], country_data['total_vaccinations'],␣
 ↪label=country)

plt.title('COVID-19 Vaccination Progress in East Africa')
plt.xlabel('Date')
plt.ylabel('Total Vaccinations')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

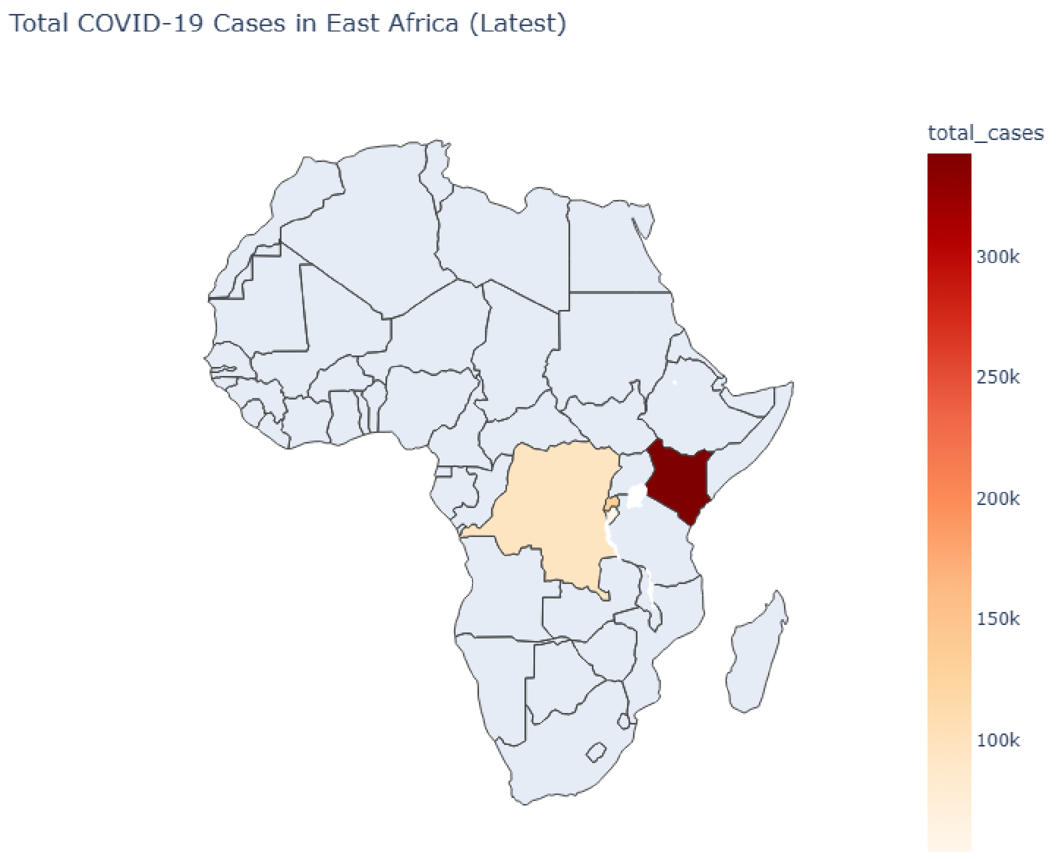


[48]:
```python
import pandas as pd
import matplotlib.pyplot as plt

# Filter for Some of East Africa Countries
east_africa = ['Burundi', 'Democratic Republic of Congo', 'Kenya', 'Rwanda']
```

```python
# Filter and clean data
df_ea = df[df['location'].isin(east_africa)].
↪dropna(subset=['people_vaccinated', 'population'])

# Ensure date column is datetime
df_ea['date'] = pd.to_datetime(df_ea['date'])

# Get the latest data per country
latest = df_ea.sort_values('date').groupby('location').last()

# Setup subplots grid (3 rows, 3 cols works for 8 countries)
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(15, 12))
axes = axes.flatten()  # Flatten the 2D array to make indexing easier

colors = ['grey', 'coral']  # grey for vaccinated, coral for unvaccinated

# Plot pie charts
for i, country in enumerate(east_africa):
    vaccinated = latest.loc[country, 'people_vaccinated']
    population = latest.loc[country, 'population']
    unvaccinated = population - vaccinated

    sizes = [vaccinated, unvaccinated]
    labels = ['Vaccinated', 'Unvaccinated']

    axes[i].pie(sizes, labels=labels, colors=colors, autopct='%.1f%%',␣
↪startangle=140)
    axes[i].set_title(f'{country}')
    axes[i].axis('equal')

# Hide any unused subplot (in this case, 9th subplot)
if len(east_africa) < len(axes):
    for j in range(len(east_africa), len(axes)):
        axes[j].axis('off')

# Title and layout
plt.suptitle('COVID-19 Vaccination Status in East Africa', fontsize=16)
plt.tight_layout
plt.show()
```



[47]:
```
# Filter for East African countries and get the latest data
df = pd.read_csv("owid-covid-data.csv")
latest_ea = df[df['location'].isin(east_africa)]
latest_ea = latest_ea.sort_values('date').groupby('location').last().
  ↪reset_index()

# Prepare choropleth map data
map_data = latest_ea.reset_index()[['iso_code', 'location', 'total_cases']].
  ↪dropna()

fig = px.choropleth(
    map_data,
    locations='iso_code',
    color='total_cases',
    hover_name='location',
    color_continuous_scale='OrRd',
    title='Total COVID-19 Cases in East Africa (Latest)'
)

fig.update_layout(geo_scope='africa', width=1000, height=700 )  # Focuses map␣
  ↪on Africa
fig.show()
```



[ ]:
```
## Final Insights: COVID-19 Situation in Some of East African Countries

This analysis provides analyisi of the COVID-19 pandemic and vaccination trends␣
  ↪in some of East African countries using data from the global dataset. The␣
  ↪following key insights were derived:

---

### 1. Kenya and Uganda Lead in Total Reported Cases
```

```
- Among the eight East African countries analyzed, Kenya and Uganda␣
 ↪consistently reported the highest number of total COVID-19 cases.
- This is likely due to relatively better testing capacity, urban population␣
 ↪density, and improved health reporting systems compared to neighbors.
- On the other hand, Burundi, DR Congo, and the rest reported fewer cases –␣
 ↪though this may reflect limited testing and underreporting rather than lower␣
 ↪transmission.

---

### 2. Vaccination Uptake Remains Low in Most Countries
- None of the East African countries had vaccinated more than 60% of their␣
 ↪population as of the latest data.
- Countries like Rwanda and Kenya showed notable progress in vaccination, while␣
 ↪DR Congo and Burundi remained significantly behind.
- This underscores gaps in vaccine accessibility, distribution infrastructure,␣
 ↪and public health outreach.

---

### 3. Death Rates Are Low – But Interpret with Caution
- Most countries reported low death rates relative to total cases, often below␣
 ↪2%.
- While this may seem encouraging, it is important to consider that limited␣
 ↪testing and reporting could distort the true impact.
- Additionally, many countries have a younger population, which may have␣
 ↪contributed to lower mortality, even with higher transmission rates.

---

### 4. Daily Case Trends Follow Global Waves
- Spikes in daily new cases were observed during mid-2021 and early 2022, which␣
 ↪corresponds to the Delta and Omicron waves globally.
- This confirms that East Africa was not isolated from global transmission␣
 ↪dynamics, reinforcing the importance of international coordination during␣
 ↪pandemics.

---

### 5. Data Gaps and Regional Disparities Still Exist
- DR Congo, and Burundi had large data gaps or inconsistent reporting over time.
- This limits the ability to track real-time progress and design effective␣
 ↪policies.
- Greater support is needed to strengthen health data systems, especially in␣
 ↪conflict-affected or low-resource areas.
```

```
---

## Final Thoughts

This East African COVID-19 analysis highlights both progress and challenges in␣
  ↪managing the pandemic. While some countries have made strides in vaccination␣
  ↪and surveillance, others remain behind due to structural and logistical␣
  ↪constraints. The findings emphasize the importance of data transparency,␣
  ↪regional cooperation, and health system investment for future preparedness.

##  Conclusion

This project provided a focused analysis of the COVID-19 pandemic in East␣
  ↪Africa using real-world global data. By exploring case trends, vaccination␣
  ↪progress, and comparing key metrics across countries, we gained valuable␣
  ↪insights into the regional impact and response strategies. The analysis␣
  ↪revealed disparities in vaccination rates, underreporting challenges, and␣
  ↪the importance of robust health systems.

Overall, this project demonstrates the power of data analytics in informing␣
  ↪public health decisions and highlights the need for continued investment in␣
  ↪data transparency and pandemic preparedness across East African nations.
```

```
[ ]:
```