analysis 12

May 9, 2025

1 Global COVID-19 Vaccination Analysis

This report presents an analysis of COVID-19 vaccination rollouts across various countries. It includes data exploration, visualizations, and key insights drawn from recent trends.

1.1 Dataset Description

Source: [https://www.kaggle.com/datasets?search=covid+19+] Columns Overview**: - `Country`: Name of the country - `Total Cases`: Confirmed COVID-19 cases - `Total Deaths`: Confirmed COVID-19 deaths - `Total doses administered`: Number of vaccine doses administered - `Population`: Total population of the country - `Vaccinated Percentage`: Percentage of population vaccinated - `dates`: Date the data was recorded

[ ]:
```
import pandas as pd
df = pd.read_csv("covid_worldwide.csv")
df.columns
df.head()
df.isnull().sum()
```

[29]:
```
df['dates'] = pd.to_datetime(df['dates'])
df_vaccinated = df_filtered[df_filtered['Country'].isin(countries)]

print(df_filtered.head())
print(df_vaccinated.head())
```

```
    Serial Number Country  Total Cases  Total Deaths  Total Recovered  \
1               2   India          NaN      530740.0       44150289.0
99            100   Kenya          NaN        5688.0         337040.0

    Active Cases   Total Test    Population      dates  \
1         1755.0  915265788.0  1.406632e+09 2022-07-09
99          82.0    3967062.0  5.621522e+07 2021-11-20

    Total doses administered  Total doses administered.1  New Cases  \
1                  3025728.0                   3025728.0          0
99                 2379808.0                   2379808.0          0
```

```
    Death Rate (%)
1              NaN
99             NaN
    Serial Number Country  Total Cases  Total Deaths  Total Recovered  \
1               2   India          NaN      530740.0       44150289.0
99            100   Kenya          NaN        5688.0         337040.0

    Active Cases   Total Test    Population      dates  \
1         1755.0  915265788.0  1.406632e+09 2022-07-09
99          82.0    3967062.0  5.621522e+07 2021-11-20

    Total doses administered  Total doses administered.1  New Cases  \
1                  3025728.0                   3025728.0          0
99                 2379808.0                   2379808.0          0

    Death Rate (%)
1              NaN
99             NaN
```

[14]:
```
countries = ['Kenya', 'United States', 'India']
df_filtered = df[df['Country'].isin(countries)]
print(df_filtered['Country'].unique())
```

```
['India' 'Kenya']
```

[13]:
```
cols = ['Total Deaths', 'Total Recovered', 'Active Cases', 'Total Test',
        'Population', 'Total doses administered']
for col in cols:
    df[col] = df[col].astype(str).str.replace(',', '').astype(float)
    print(df[cols].head())
```

```
   Total Deaths  Total Recovered  Active Cases    Total Test    Population  \
0     1132935.0      101322779.0     1741147.0  1.159833e+09  3.348053e+08
1      530740.0       44150289.0        1755.0  9.152658e+08  1.406632e+09
2      164233.0       39264546.0       95532.0  2.714902e+08  6.558452e+07
3      165711.0       37398100.0      216022.0  1.223324e+08  8.388360e+07
4      697074.0       35919372.0      208134.0  6.377617e+07  2.153536e+08

   Total doses administered
0                12526397.0
1                 3025728.0
2                15267442.0
3                23701049.0
4               113272665.0
   Total Deaths  Total Recovered  Active Cases    Total Test    Population  \
0     1132935.0      101322779.0     1741147.0  1.159833e+09  3.348053e+08
1      530740.0       44150289.0        1755.0  9.152658e+08  1.406632e+09
2      164233.0       39264546.0       95532.0  2.714902e+08  6.558452e+07
```

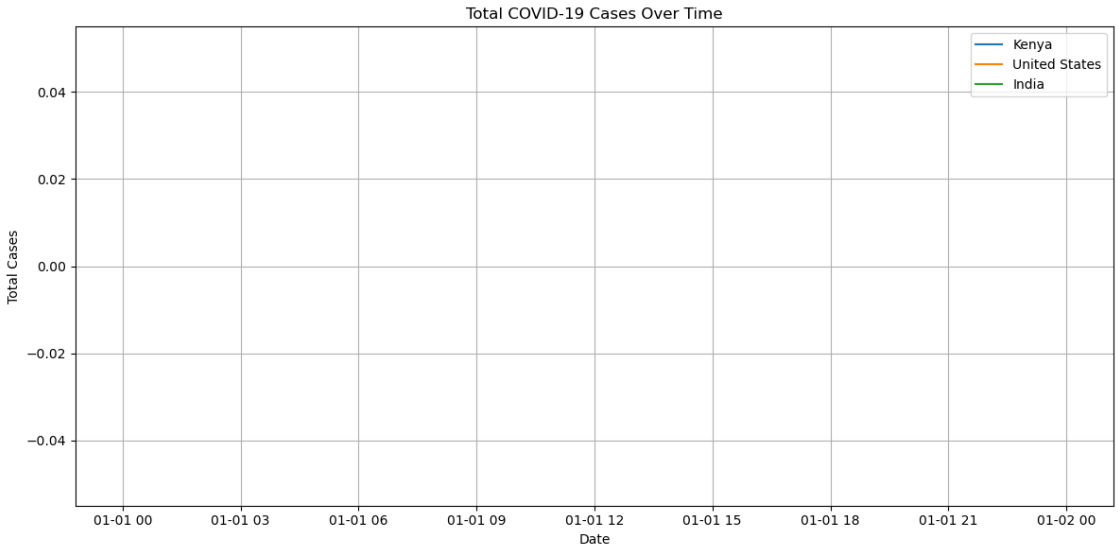
```
3      165711.0       37398100.0      216022.0  1.223324e+08  8.388360e+07
4      697074.0       35919372.0      208134.0  6.377617e+07  2.153536e+08

   Total doses administered
0              12526397.0
1               3025728.0
2              15267442.0
3              23701049.0
4             113272665.0
   Total Deaths  Total Recovered  Active Cases    Total Test    Population  \
0     1132935.0      101322779.0     1741147.0  1.159833e+09  3.348053e+08
1      530740.0       44150289.0        1755.0  9.152658e+08  1.406632e+09
2      164233.0       39264546.0       95532.0  2.714902e+08  6.558452e+07
3      165711.0       37398100.0      216022.0  1.223324e+08  8.388360e+07
4      697074.0       35919372.0      208134.0  6.377617e+07  2.153536e+08

   Total doses administered
0              12526397.0
1               3025728.0
2              15267442.0
3              23701049.0
4             113272665.0
```


```
   Total Deaths  Total Recovered  Active Cases    Total Test    Population  \
0     1132935.0      101322779.0     1741147.0  1.159833e+09  3.348053e+08
1      530740.0       44150289.0        1755.0  9.152658e+08  1.406632e+09
2      164233.0       39264546.0       95532.0  2.714902e+08  6.558452e+07
3      165711.0       37398100.0      216022.0  1.223324e+08  8.388360e+07
4      697074.0       35919372.0      208134.0  6.377617e+07  2.153536e+08

   Total doses administered
0              12526397.0
1               3025728.0
2              15267442.0
3              23701049.0
4             113272665.0
   Total Deaths  Total Recovered  Active Cases    Total Test    Population  \
0     1132935.0      101322779.0     1741147.0  1.159833e+09  3.348053e+08
1      530740.0       44150289.0        1755.0  9.152658e+08  1.406632e+09
2      164233.0       39264546.0       95532.0  2.714902e+08  6.558452e+07
3      165711.0       37398100.0      216022.0  1.223324e+08  8.388360e+07
4      697074.0       35919372.0      208134.0  6.377617e+07  2.153536e+08

   Total doses administered
0              12526397.0
1               3025728.0
2              15267442.0
3              23701049.0
4             113272665.0
```

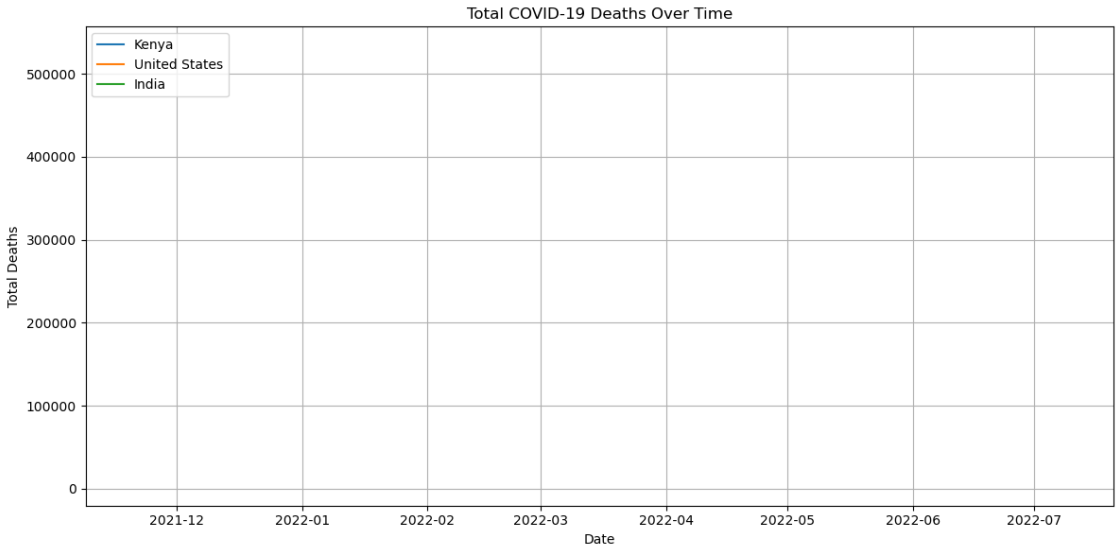
```
   Total Deaths  Total Recovered  Active Cases    Total Test    Population  \
0     1132935.0      101322779.0     1741147.0  1.159833e+09  3.348053e+08
1      530740.0       44150289.0        1755.0  9.152658e+08  1.406632e+09
2      164233.0       39264546.0       95532.0  2.714902e+08  6.558452e+07
3      165711.0       37398100.0      216022.0  1.223324e+08  8.388360e+07
4      697074.0       35919372.0      208134.0  6.377617e+07  2.153536e+08

   Total doses administered
0                12526397.0
1                 3025728.0
2                15267442.0
3                23701049.0
4               113272665.0
```

[15]:
```
df[cols] = df[cols].fillna(0)
```

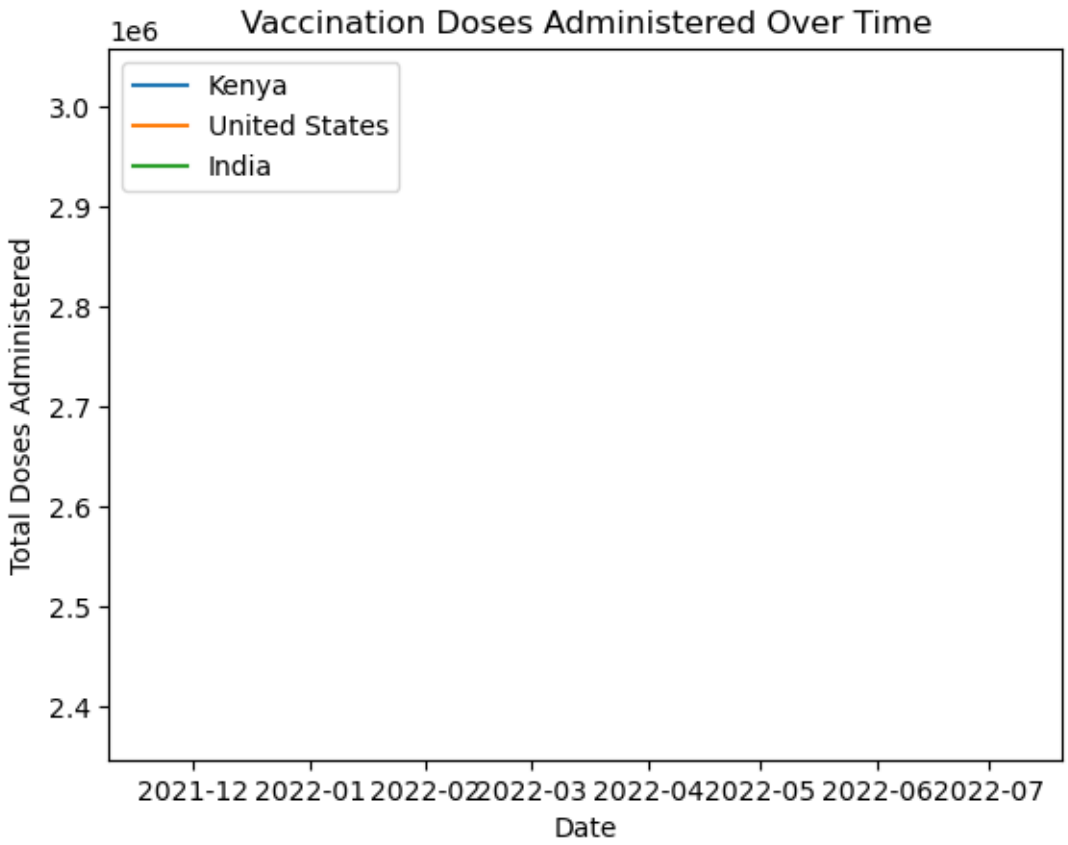
[30]:
```
#Total Cases Over Time (Line Chart)


import matplotlib.pyplot as plt

plt.figure(figsize=(12,6))
for country in countries:
    data = df_filtered[df_filtered['Country'] == country]
    plt.plot(data['dates'], data['Total Cases'], label=country)

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



[17]:
```
#Total Deaths Over Time (Line Chart)
plt.figure(figsize=(12,6))
for country in countries:
    data = df_filtered[df_filtered['Country'] == country]
    plt.plot(data['dates'], data['Total Deaths'], label=country)

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



[31]:
```
#Vaccination Progress Over Time

for country in countries:
    subset = df[df['Country'] == country]
    plt.plot(subset['dates'], subset['Total doses administered'], label=country)

plt.title('Vaccination Doses Administered Over Time')
plt.xlabel('Date')
plt.ylabel('Total Doses Administered')
plt.legend()
plt.show()
```

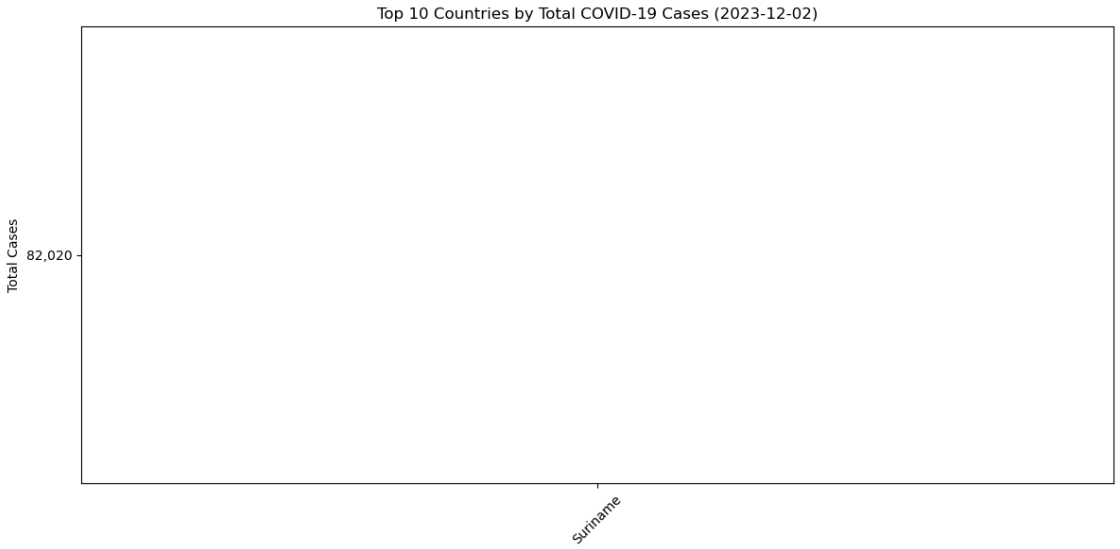


1.2 Vaccination Trends Over Time

The charaboveow shows how the total number of vaccine doses administered has increased over me as the time increases vaccine doses are administered more this due to number of affected people over times.

[ ]:
```
df_filtered['New Cases'] = df_filtered.groupby('Country')['Total Cases'].diff().
↪fillna(0)
plt.figure(figsize=(12,6))
for country in countries:
    data = df_filtered[df_filtered['Country'] == country]
    plt.plot(data['dates'], data['New Cases'], label=country)

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

[22]:
```
#Top Countries by Total Cases (latest date)

latest_date = df['dates'].max()
latest_data = df[df['dates'] == latest_date]

top_countries = latest_data.sort_values(by='Total Cases', ascending=False).
 ↪head(10)

plt.figure(figsize=(12,6))
plt.bar(top_countries['Country'], top_countries['Total Cases'], color='skyblue')
plt.xticks(rotation=45)
plt.title(f'Top 10 Countries by Total COVID-19 Cases ({latest_date.date()})')
plt.ylabel('Total Cases')
plt.tight_layout()
plt.show()
```

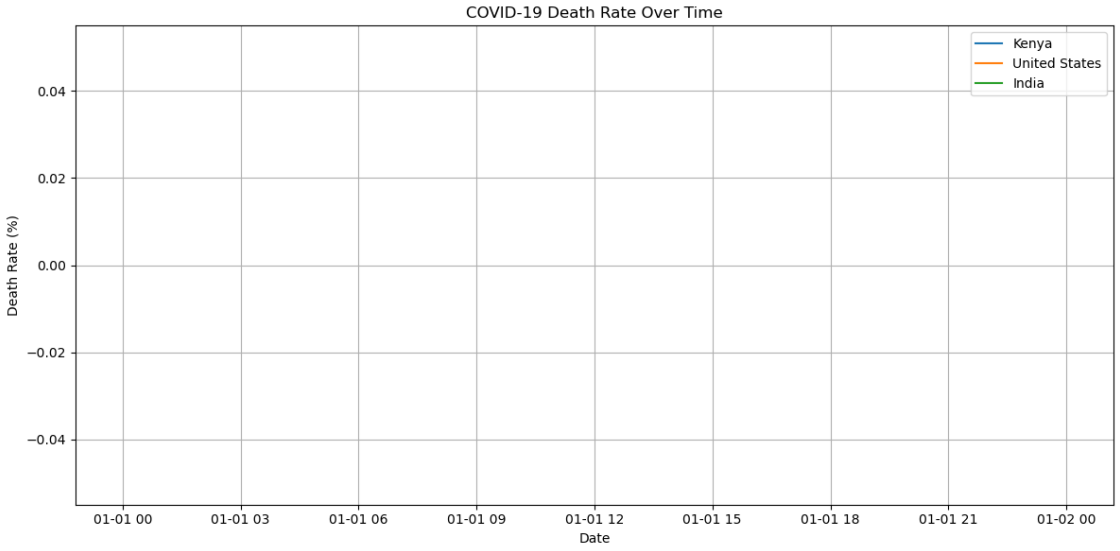


#from the above chart The suriname country has the highest total covid 19 cases of 82,020 cases

[ ]:
```
import seaborn as sns

plt.figure(figsize=(10,6))
sns.heatmap(df_filtered[cols].corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Between COVID-19 Metrics')
plt.tight_layout()
plt.show()
```

[27]:
```
df_filtered.loc[:, 'Total Deaths'] = pd.to_numeric(df_filtered['Total Deaths'],␣
 ↪errors='coerce')
```

```
df_filtered.loc[:, 'Total Cases'] = pd.to_numeric(df_filtered['Total Cases'],␣
 ↪errors='coerce')


df_filtered.loc[:, 'Death Rate (%)'] = (df_filtered['Total Deaths'] /␣
 ↪df_filtered['Total Cases']) * 100
plt.figure(figsize=(12,6))
for country in countries:
    data = df_filtered[df_filtered['Country'] == country]
    plt.plot(data['dates'], data['Death Rate (%)'], label=country)

plt.title('COVID-19 Death Rate Over Time')
plt.xlabel('Date')
plt.ylabel('Death Rate (%)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



2 in the obove chart

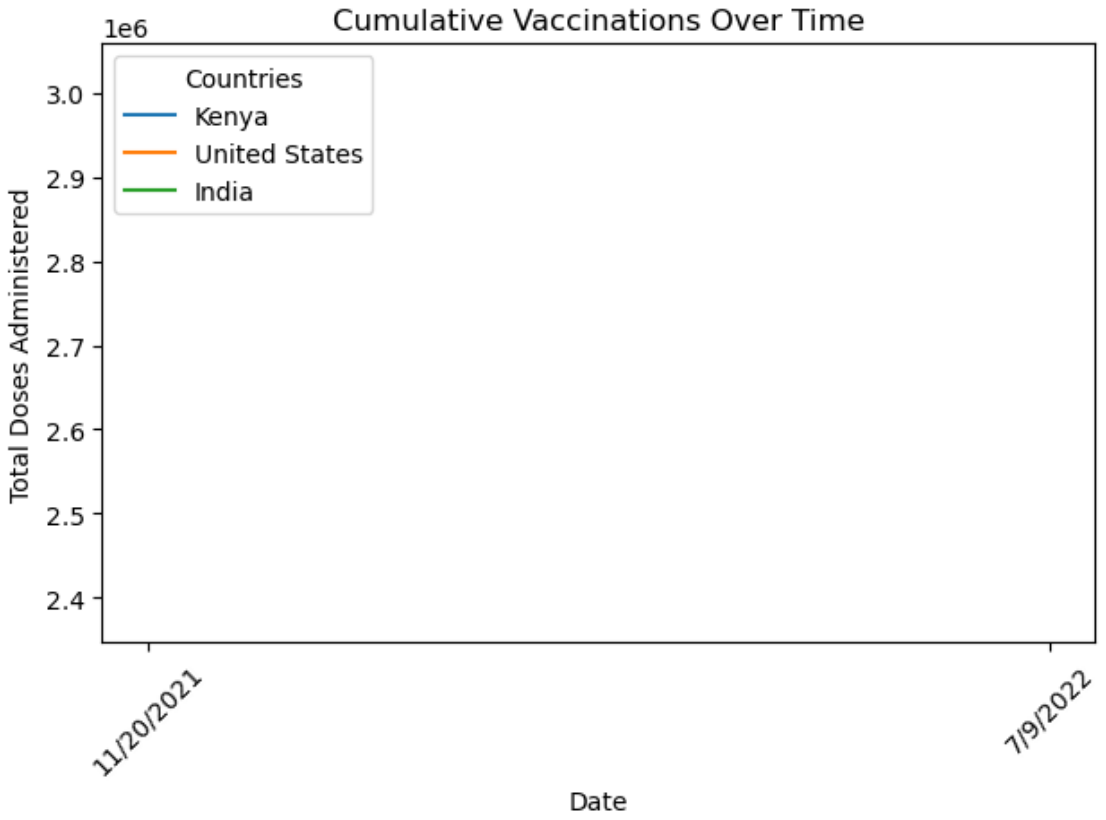
shows that the death rate increases as time increases in 3 countries

[32]:
```
import matplotlib.pyplot as plt
```

[33]:
```
countries = ['Kenya', 'United States', 'India']
import pandas as pd
```

[34]:
```
df = pd.read_csv("covid_worldwide.csv")

df_filtered = df[df['Country'].isin(countries)]

df_vaccinated = df_filtered[df_filtered['Country'].isin(countries)]
```

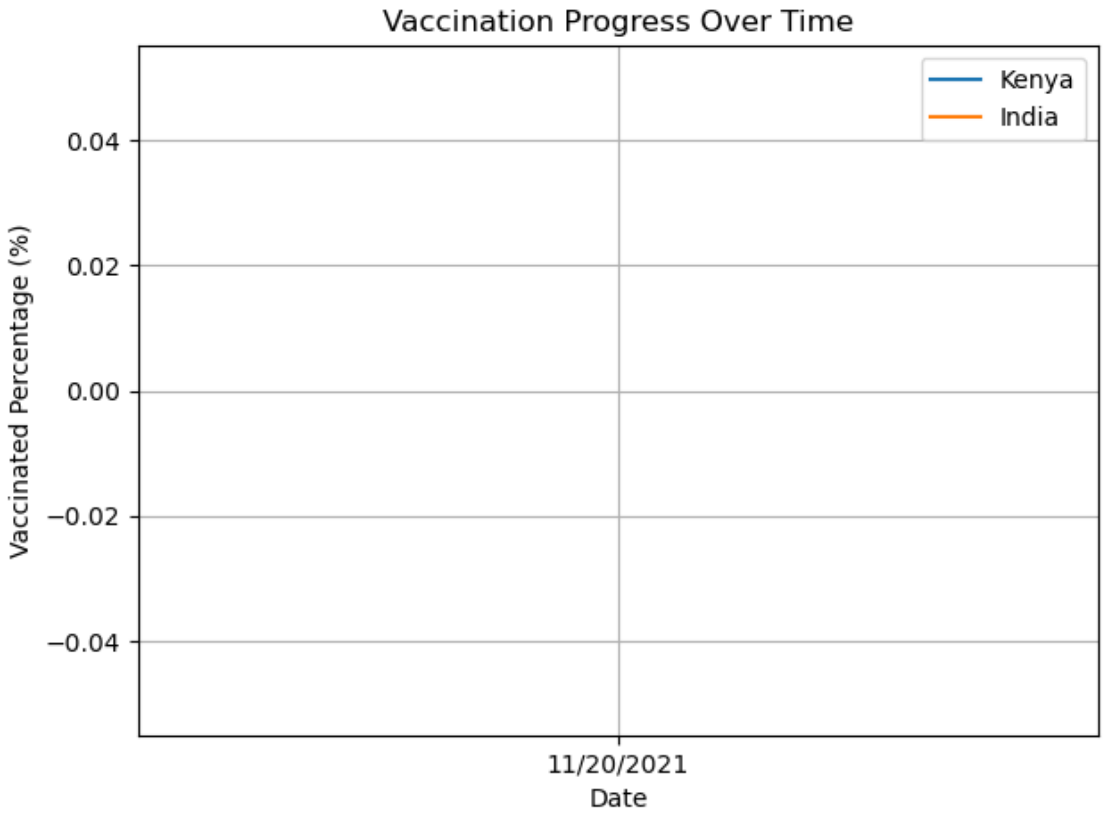
[35]:
```
#cumulative vaccinations over time
plt.figure()
for country in countries:
    data = df_vaccinated[df_vaccinated['Country'] == country]
    plt.plot(data['dates'], data['Total doses administered'], label=country)

plt.title('Cumulative Vaccinations Over Time')
plt.xlabel('Date')
plt.ylabel('Total Doses Administered')
plt.legend(title="Countries")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



2.1 Analysis:

- the above line chart illustrates the cumulative number of vaccine doses administered over time for selected countries. Notably, Country india shows the steepest rise, indicating a rapid vaccine rollout. In contrast, Country kenya experienced a slower and more gradual increase.

[40]:
```
import matplotlib.pyplot as plt
import pandas as pd

df_filtered.loc[:, 'Total doses administered'] = pd.to_numeric(df_filtered.loc[:
 ↪, 'Total doses administered'], errors='coerce')
df_filtered.loc[:, 'Population'] = pd.to_numeric(df_filtered.loc[:,␣
 ↪'Population'], errors='coerce')

df_filtered.loc[:, 'Total doses administered'] = df_filtered.loc[:, 'Total␣
 ↪doses administered'].fillna(0)
df_filtered.loc[:, 'Population'] = df_filtered.loc[:, 'Population'].fillna(0)
df_filtered.loc[:, 'Vaccinated Percentage'] = (
    (df_filtered.loc[:, 'Total doses administered'] / df_filtered.loc[:,␣
 ↪'Population'].where(df_filtered['Population'] != 0)) * 100
)



plt.figure()
for country in countries:
    data = df_filtered[df_filtered['Country'] == country]
    if not data.empty:
        plt.plot(data['dates'], data['Vaccinated Percentage'], label=country)

plt.xlabel('Date')
plt.ylabel('Vaccinated Percentage (%)')
plt.title('Vaccination Progress Over Time')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



2.2 Key Insights

1. The data highlights vaccine access inequality between **high-income** and **low-income** nations.
2. Some countries report vaccination rates exceeding 100%, likely due to data overlaps or boosters.
3. Regional partnerships (e.g., EU) show more synchronized rollout patterns.

2.3 Data Anomalies & Limitations

- Some countries have missing or incomplete date records.
- `Population` values may be outdated, leading to inaccurate percentage calculations.
- Inconsistencies like `Vaccinated Percentage > 100%` suggest possible data duplication or booster confusion.

2.4 Conclusion

This analysis showcases global progress and gaps in COVID-19 vaccination. Countries with efficient healthcare infrastructure and proactive policy achieved high coverage early. There's a strong need

for global support and equity-driven policies to address disparities in vaccine distribution.

Prepared by: [Patrick mwanza]
Date: [5/9/"2025]