COVID19

May 16, 2025

[1]:
```
!pip install pandas matplotlib seaborn
```

```
Defaulting to user installation because normal site-packages is not writeable
Looking in links: /usr/share/pip-wheels
Requirement already satisfied: pandas in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (2.1.4)
Requirement already satisfied: matplotlib in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (3.8.0)
Requirement already satisfied: seaborn in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (0.12.2)
Requirement already satisfied: numpy<2,>=1.22.4 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
pandas) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
matplotlib) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
matplotlib) (23.2)
Requirement already satisfied: pillow>=6.2.0 in
```

```
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
matplotlib) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
matplotlib) (3.0.9)
Requirement already satisfied: six>=1.5 in
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages (from
python-dateutil>=2.8.2->pandas) (1.16.0)
```

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

# Set Seaborn style for visuals
sns.set(style="whitegrid")
```

[3]:
```
# Load the dataset
df = pd.read_csv("owid-covid-data.csv")

# Check shape and preview
print("Shape:", df.shape)
df.head()
```

```
Shape: (350085, 67)
```

[3]:

	iso_code	continent	location	date	total_cases	new_cases	\
0	AFG	Asia	Afghanistan	2020-01-03	NaN	0.0	
1	AFG	Asia	Afghanistan	2020-01-04	NaN	0.0	
2	AFG	Asia	Afghanistan	2020-01-05	NaN	0.0	
3	AFG	Asia	Afghanistan	2020-01-06	NaN	0.0	
4	AFG	Asia	Afghanistan	2020-01-07	NaN	0.0	

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

	male_smokers	handwashing_facilities	hospital_beds_per_thousand	\
0	NaN	37.746	0.5	
1	NaN	37.746	0.5	
2	NaN	37.746	0.5	
3	NaN	37.746	0.5	
4	NaN	37.746	0.5	

	life_expectancy	human_development_index	population	\

```
0            64.83                    0.511  41128772.0
1            64.83                    0.511  41128772.0
2            64.83                    0.511  41128772.0
3            64.83                    0.511  41128772.0
4            64.83                    0.511  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]:
```
# View columns
print(df.columns)

# Check missing values
df.isnull().sum().sort_values(ascending=False).head(20)
```

```
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')
```

[4]:
```
weekly_icu_admissions_per_million          339880
weekly_icu_admissions                      339880
excess_mortality_cumulative_per_million    337901
excess_mortality_cumulative_absolute       337901
excess_mortality_cumulative                337901
excess_mortality                           337901
weekly_hosp_admissions                     326832
weekly_hosp_admissions_per_million         326832
icu_patients                               312470
icu_patients_per_million                   312470
hosp_patients                              311183
hosp_patients_per_million                  311183
total_boosters                             302523
total_boosters_per_hundred                 302523
new_vaccinations                           284739
people_fully_vaccinated_per_hundred        277510
people_fully_vaccinated                    277510
new_tests_per_thousand                     274682
new_tests                                  274682
people_vaccinated_per_hundred              274174
dtype: int64
```

[5]:
```
df = df[['location', 'date', 'total_cases', 'total_deaths', 'new_cases',
         'new_deaths', 'total_vaccinations', 'people_fully_vaccinated',␣
 ↪'population']]
df.head()
```

[5]:

	location	date	total_cases	total_deaths	new_cases	new_deaths	\
0	Afghanistan	2020-01-03	NaN	NaN	0.0	0.0	
1	Afghanistan	2020-01-04	NaN	NaN	0.0	0.0	
2	Afghanistan	2020-01-05	NaN	NaN	0.0	0.0	
3	Afghanistan	2020-01-06	NaN	NaN	0.0	0.0	
4	Afghanistan	2020-01-07	NaN	NaN	0.0	0.0	

	total_vaccinations	people_fully_vaccinated	population
0	NaN	NaN	41128772.0
1	NaN	NaN	41128772.0

```
2                NaN                      NaN  41128772.0
3                NaN                      NaN  41128772.0
4                NaN                      NaN  41128772.0
```

[6]:
```
# Convert date column to datetime format
df['date'] = pd.to_datetime(df['date'])

# Filter for Kenya, USA, and India
countries = ['Kenya', 'United States', 'India']
df_countries = df[df['location'].isin(countries)].copy()

# Check result
df_countries.head()
```

[6]:
```
        location       date  total_cases  total_deaths  new_cases  new_deaths  \
139773     India 2020-01-03          NaN           NaN        0.0         0.0
139774     India 2020-01-04          NaN           NaN        0.0         0.0
139775     India 2020-01-05          NaN           NaN        0.0         0.0
139776     India 2020-01-06          NaN           NaN        0.0         0.0
139777     India 2020-01-07          NaN           NaN        0.0         0.0

        total_vaccinations  people_fully_vaccinated    population
139773                 NaN                      NaN  1.417173e+09
139774                 NaN                      NaN  1.417173e+09
139775                 NaN                      NaN  1.417173e+09
139776                 NaN                      NaN  1.417173e+09
139777                 NaN                      NaN  1.417173e+09
```

[7]:
```
# Fill missing values forward (group by country)
df_countries.sort_values(by=['location', 'date'], inplace=True)
df_countries.fillna(method='ffill', inplace=True)

# Drop remaining rows with critical missing values
df_countries.dropna(subset=['total_cases', 'total_deaths'], inplace=True)

# Check again
df_countries.isnull().sum()
```

```
/tmp/ipykernel_416/508114291.py:3: FutureWarning: DataFrame.fillna with 'method'
is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill()
instead.
  df_countries.fillna(method='ffill', inplace=True)
```

[7]:
```
location                0
date                    0
total_cases             0
total_deaths            0
```

```
new_cases                     0
new_deaths                    0
total_vaccinations          308
people_fully_vaccinated     337
population                    0
dtype: int64
```

[8]:
```
# Confirm no date conversion issues
print(df_countries.dtypes)

# Preview filtered dataset
df_countries.tail()
```

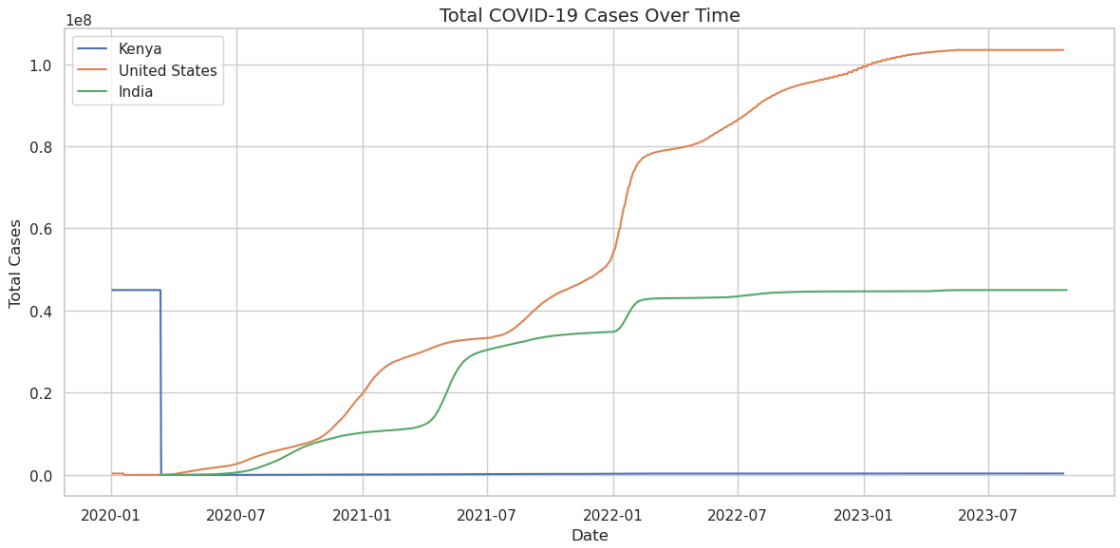
```
location                            object
date                        datetime64[ns]
total_cases                        float64
total_deaths                       float64
new_cases                          float64
new_deaths                         float64
total_vaccinations                 float64
people_fully_vaccinated            float64
population                         float64
dtype: object
```

[8]:

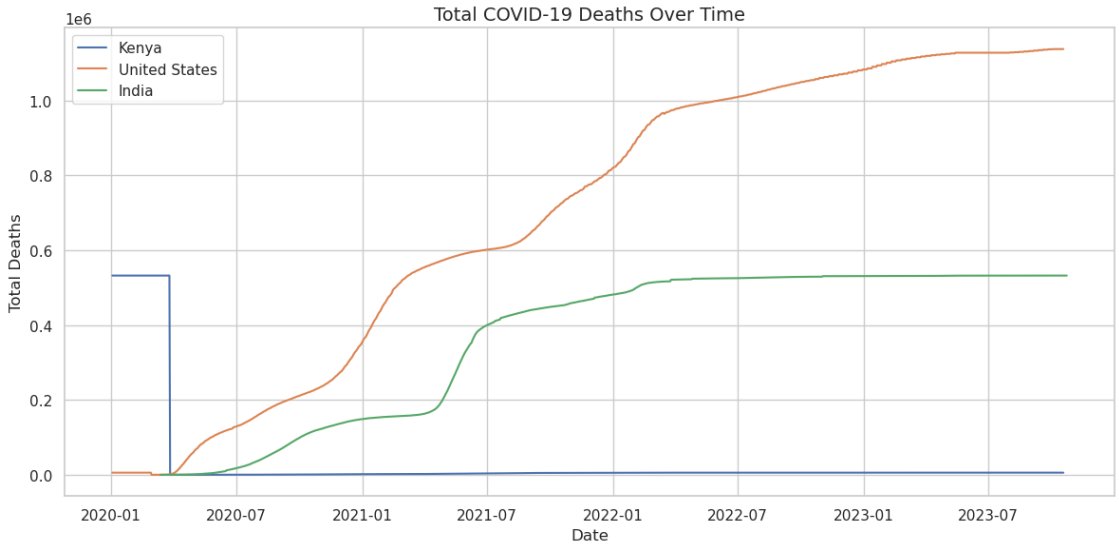
	location	date	total_cases	total_deaths	new_cases	new_deaths	total_vaccinations	people_fully_vaccinated	population
330861	United States	2023-10-14	103436829.0	1136920.0	0.0	0.0	676728782.0	230637348.0	338289856.0
330862	United States	2023-10-15	103436829.0	1136920.0	0.0	0.0	676728782.0	230637348.0	338289856.0
330863	United States	2023-10-16	103436829.0	1136920.0	0.0	0.0	676728782.0	230637348.0	338289856.0
330864	United States	2023-10-17	103436829.0	1136920.0	0.0	0.0	676728782.0	230637348.0	338289856.0
330865	United States	2023-10-18	103436829.0	1136920.0	0.0	0.0	676728782.0	230637348.0	338289856.0

[9]:
```
plt.figure(figsize=(12, 6))

# Plot total cases over time
for country in ['Kenya', 'United States', 'India']:
    country_data = df_countries[df_countries['location'] == country]
    plt.plot(country_data['date'], country_data['total_cases'], label=country)

plt.title("Total COVID-19 Cases Over Time", fontsize=14)
plt.xlabel("Date")
```

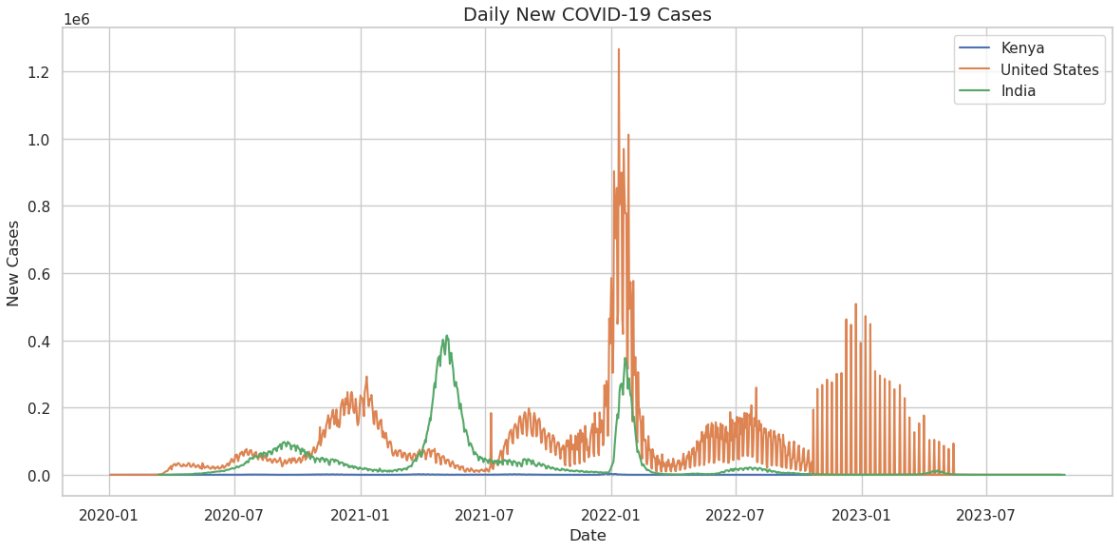
```
plt.ylabel("Total Cases")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



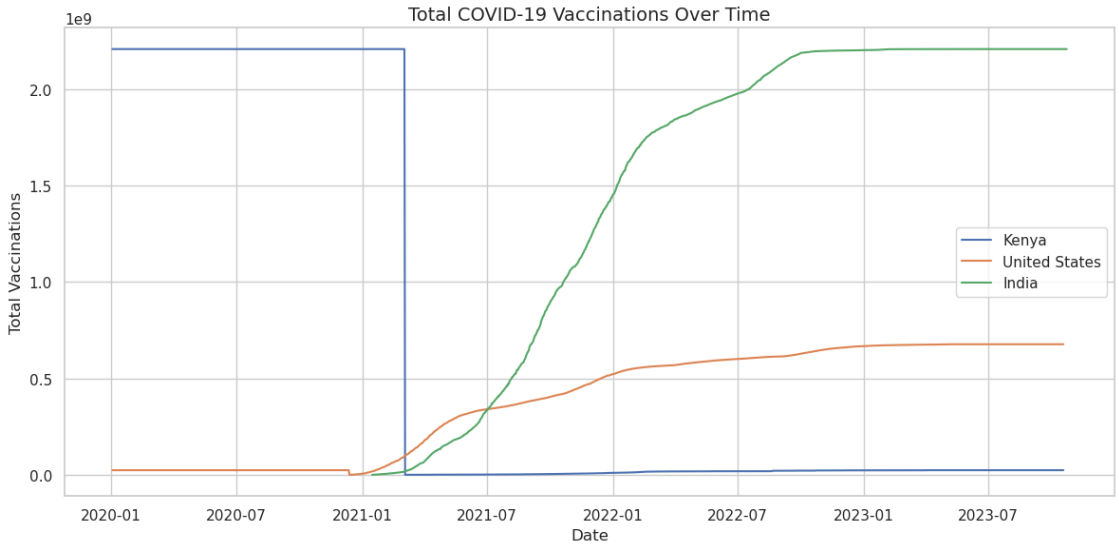
[10]:
```
plt.figure(figsize=(12, 6))

# Plot total deaths over time
for country in ['Kenya', 'United States', 'India']:
    country_data = df_countries[df_countries['location'] == country]
    plt.plot(country_data['date'], country_data['total_deaths'], label=country)

plt.title("Total COVID-19 Deaths Over Time", fontsize=14)
plt.xlabel("Date")
plt.ylabel("Total Deaths")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



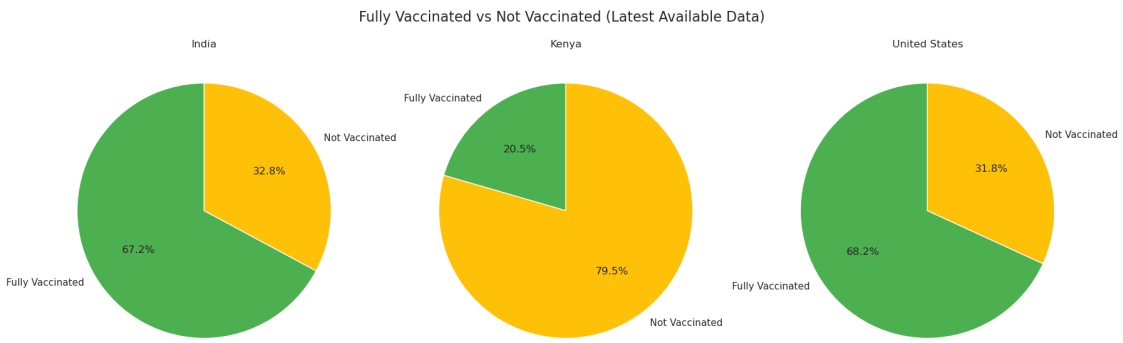
[12]:
```
plt.figure(figsize=(12, 6))

# Plot new daily cases over time
for country in ['Kenya', 'United States', 'India']:
    country_data = df_countries[df_countries['location'] == country]
    plt.plot(country_data['date'], country_data['new_cases'], label=country)

plt.title("Daily New COVID-19 Cases", fontsize=14)
plt.xlabel("Date")
plt.ylabel("New Cases")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



[13]:
```
plt.figure(figsize=(12, 6))

# Plot total vaccinations over time
for country in ['Kenya', 'United States', 'India']:
    country_data = df_countries[df_countries['location'] == country]
    plt.plot(country_data['date'], country_data['total_vaccinations'],␣
 ↪label=country)

plt.title("Total COVID-19 Vaccinations Over Time", fontsize=14)
plt.xlabel("Date")
plt.ylabel("Total Vaccinations")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



[14]:
```
# Group by country and get latest data point
latest_vax = df_countries.groupby('location').tail(1).copy()

# Calculate unvaccinated = population - fully vaccinated
latest_vax['unvaccinated'] = latest_vax['population'] -␣
 ↪latest_vax['people_fully_vaccinated']

# Plot pie charts for each country
fig, axs = plt.subplots(1, 3, figsize=(18, 6))

for ax, (_, row) in zip(axs, latest_vax.iterrows()):
    sizes = [row['people_fully_vaccinated'], row['unvaccinated']]
    labels = ['Fully Vaccinated', 'Not Vaccinated']
    colors = ['#4CAF50', '#FFC107']

    ax.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90,␣
 ↪colors=colors)
    ax.axis('equal')
    ax.set_title(f"{row['location']}")

plt.suptitle('Fully Vaccinated vs Not Vaccinated (Latest Available Data)',␣
 ↪fontsize=16)
plt.tight_layout()
plt.show()
```



[ ]:

1 COVID-19 Global Data Tracker – Summary of Insights

#Countries Analysised - Kenya - United States - India

-

1.1 Key Insights

1. **Un**ed States** had the highest total number of cases and deaths, followed by India. Kenya had significantly lower numbers but similar trends.
2. **Daily new cases** peaked around mid-2021 for all three countries, with varying waves of resurgence.
3. **Vaccination rates**:
   - The US leads in percentage of fully vaccinated individuals.
   - India follows with steady progress.
   - Kenya has the lowest full vaccination rate among the three.
4. Despite lower total cases, **Kenya showed similar mortality ratios**, highlighting the importance of medical infrastructure and early vaccination.
5. **Vaccination rollout** in the US was rapid, while India's picked up after mid-2021. Kenya's rollout has been slower due to access and distributionsues.

1.2 Patterns and Anomalies

- India had an extreme spike in early 2021 (Delta variant wave).
- The US had a more prolonged peak due to multiple variants.
- Kenya had smaller, delayed waves but with notable death spikes relative case numbers.

1.3 Conclusion

This analysis highlights the global disparity in COVID-19 case numbers, healthcare responses, and vaccine distribution. It underscores the need for **equitable vaccine access** and continued public health efforts, especially mStreamlit - Expand to more countries or regions

[ ]: