Covid-19 Global Data Tracker

May 12, 2025

[23]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

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

# Check column names
print("Columns:\n", df.columns)

# Preview the first 5 rows
print("\nHead:\n", df.head())

# Identify missing values
print("\nMissing values:\n", df.isnull().sum())
```

```
Columns:
 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')

Head:
   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]

Missing values:
 iso_code                                        0
continent                                    16665
location                                         0
date                                             0
total_cases                                  37997
                                             …
population                                       0
excess_mortality_cumulative_absolute        337901
excess_mortality_cumulative                 337901
excess_mortality                            337901
excess_mortality_cumulative_per_million     337901
Length: 67, dtype: int64
```

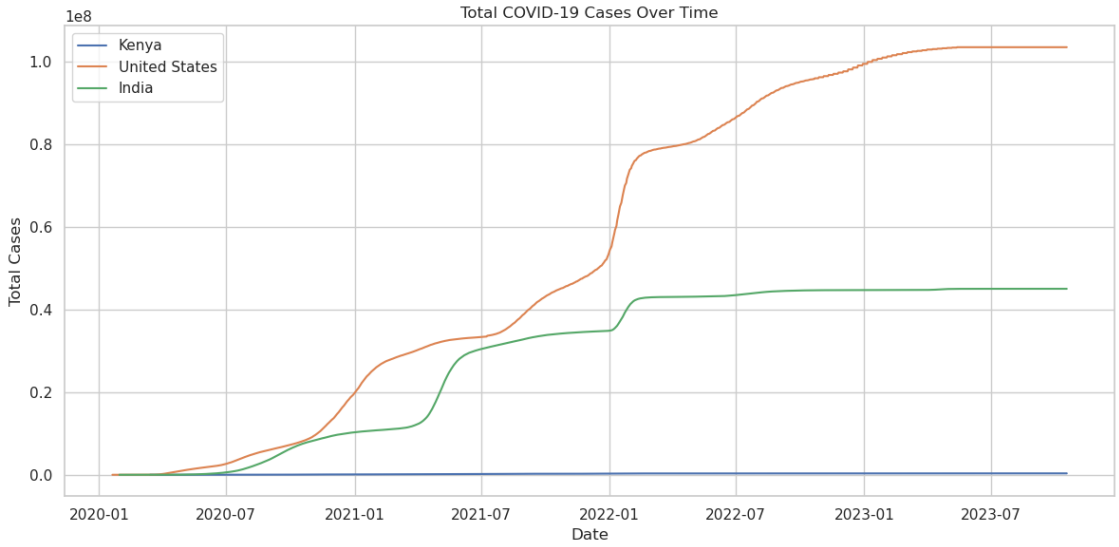
[21]:
```python
# Load the dataset
df = pd.read_csv("owid-covid-data.csv")

# Filter for countries of interest
countries = ["Kenya", "United States", "India"]
df = df[df["location"].isin(countries)]

# Drop rows with missing 'date' or other critical values (adjust as needed)
df = df.dropna(subset=["date"])

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

# Handle missing numeric values
# Fill with forward fill, backfill, or interpolation (example: interpolate)
numeric_cols = df.select_dtypes(include=["float64", "int64"]).columns
df[numeric_cols] = df[numeric_cols].interpolate()

# Preview cleaned data
print(df.head())
```

```
       iso_code continent location       date  total_cases  new_cases  \
139773      IND      Asia    India 2020-01-03          NaN        0.0
139774      IND      Asia    India 2020-01-04          NaN        0.0
139775      IND      Asia    India 2020-01-05          NaN        0.0
139776      IND      Asia    India 2020-01-06          NaN        0.0
139777      IND      Asia    India 2020-01-07          NaN        0.0

        new_cases_smoothed  total_deaths  new_deaths  new_deaths_smoothed  \
```

```
139773                NaN           NaN         0.0                   NaN
139774                NaN           NaN         0.0                   NaN
139775                NaN           NaN         0.0                   NaN
139776                NaN           NaN         0.0                   NaN
139777                NaN           NaN         0.0                   NaN

        …  male_smokers  handwashing_facilities  hospital_beds_per_thousand  \
139773  …          20.6                   59.55                        0.53
139774  …          20.6                   59.55                        0.53
139775  …          20.6                   59.55                        0.53
139776  …          20.6                   59.55                        0.53
139777  …          20.6                   59.55                        0.53

        life_expectancy  human_development_index    population  \
139773            69.66                    0.645  1.417173e+09
139774            69.66                    0.645  1.417173e+09
139775            69.66                    0.645  1.417173e+09
139776            69.66                    0.645  1.417173e+09
139777            69.66                    0.645  1.417173e+09

        excess_mortality_cumulative_absolute  excess_mortality_cumulative  \
139773                                   NaN                          NaN
139774                                   NaN                          NaN
139775                                   NaN                          NaN
139776                                   NaN                          NaN
139777                                   NaN                          NaN

        excess_mortality  excess_mortality_cumulative_per_million
139773               NaN                                     NaN
139774               NaN                                     NaN
139775               NaN                                     NaN
139776               NaN                                     NaN
139777               NaN                                     NaN

[5 rows x 67 columns]
```

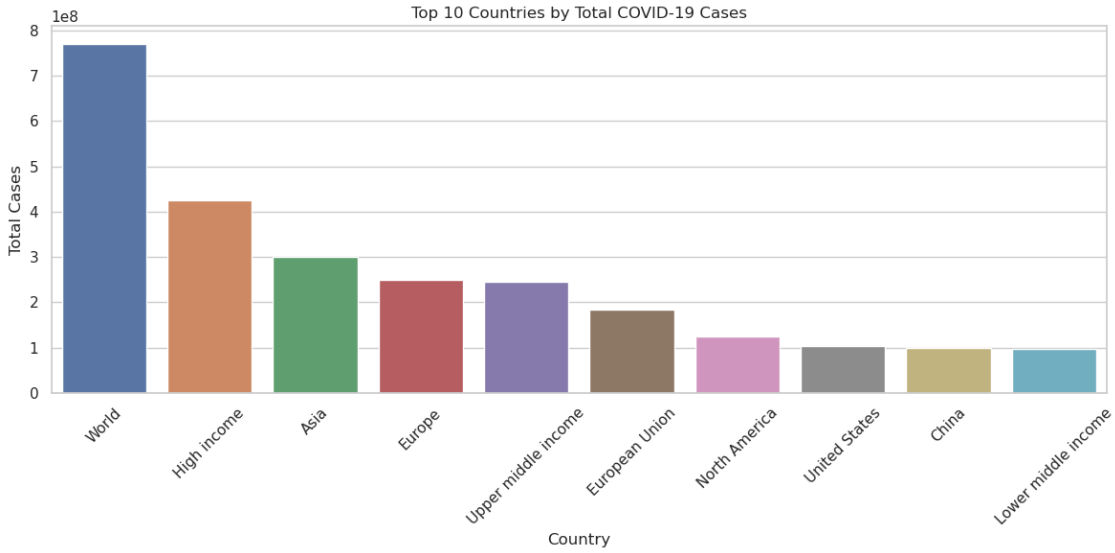
[17]:
```python
# # Exploratory Data Analysis (EDA)

# Load only the necessary columns
use_cols = [
    "date", "location", "total_cases", "total_deaths",
    "new_cases", "new_deaths", "total_vaccinations"
]
df = pd.read_csv("owid-covid-data.csv", usecols=use_cols)

# Filter for specific countries
countries = ["Kenya", "United States", "India"]
```

```
df = df[df["location"].isin(countries)]

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

# Calculate death rate
df["death_rate"] = df["total_deaths"] / df["total_cases"]

# Set style
sns.set(style="whitegrid")

# Plot total cases over time
plt.figure(figsize=(12, 6))
for country in countries:
    subset = df[df["location"] == country]
    plt.plot(subset["date"], subset["total_cases"], label=country)

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



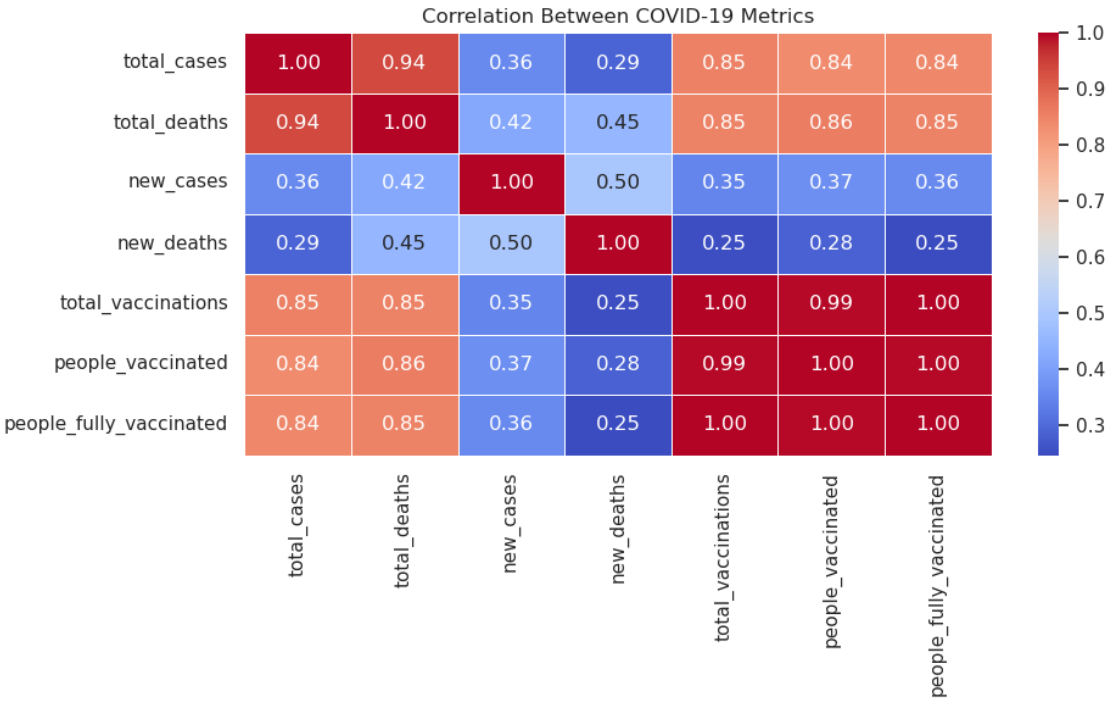
[24]:
```
# Load data
df = pd.read_csv("owid-covid-data.csv")

# Latest data per country (drop aggregate rows like continents)
```

```
latest = df.sort_values('date').dropna(subset=["total_cases"])
latest = latest.groupby("location").last()

# Top 10 countries by total cases
top10 = latest.sort_values("total_cases", ascending=False).head(10)

# Bar chart
plt.figure(figsize=(12, 6))
sns.barplot(x=top10.index, y=top10["total_cases"])
plt.title("Top 10 Countries by Total COVID-19 Cases")
plt.ylabel("Total Cases")
plt.xlabel("Country")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



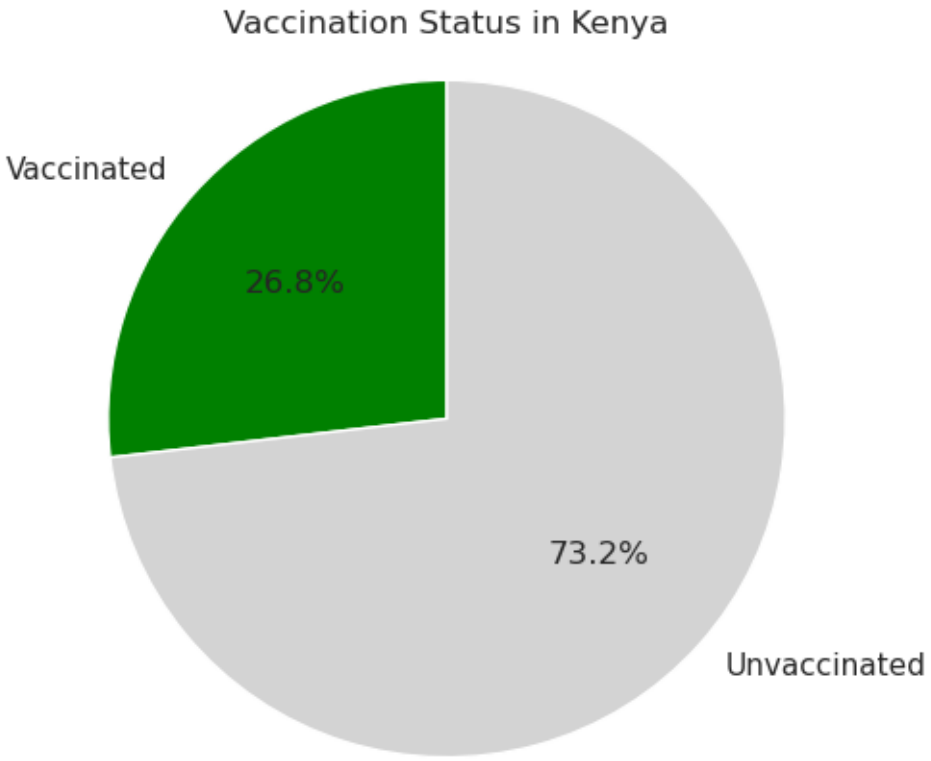
[26]:
```
# Select key columns
key_cols = [
    "total_cases", "total_deaths", "new_cases", "new_deaths",
    "total_vaccinations", "people_vaccinated", "people_fully_vaccinated"
]

# Drop rows with all NaNs in selected columns
heatmap_data = df[key_cols].dropna(how='all')

# Compute correlation matrix
corr = heatmap_data.corr()
```

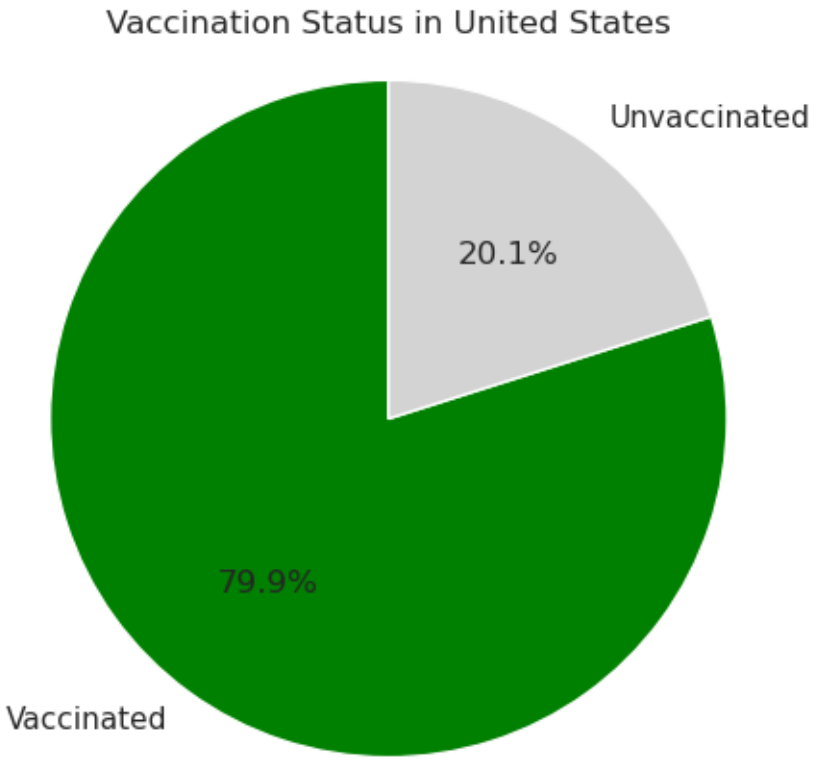
```
# Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title("Correlation Between COVID-19 Metrics")
plt.tight_layout()
plt.show()
```

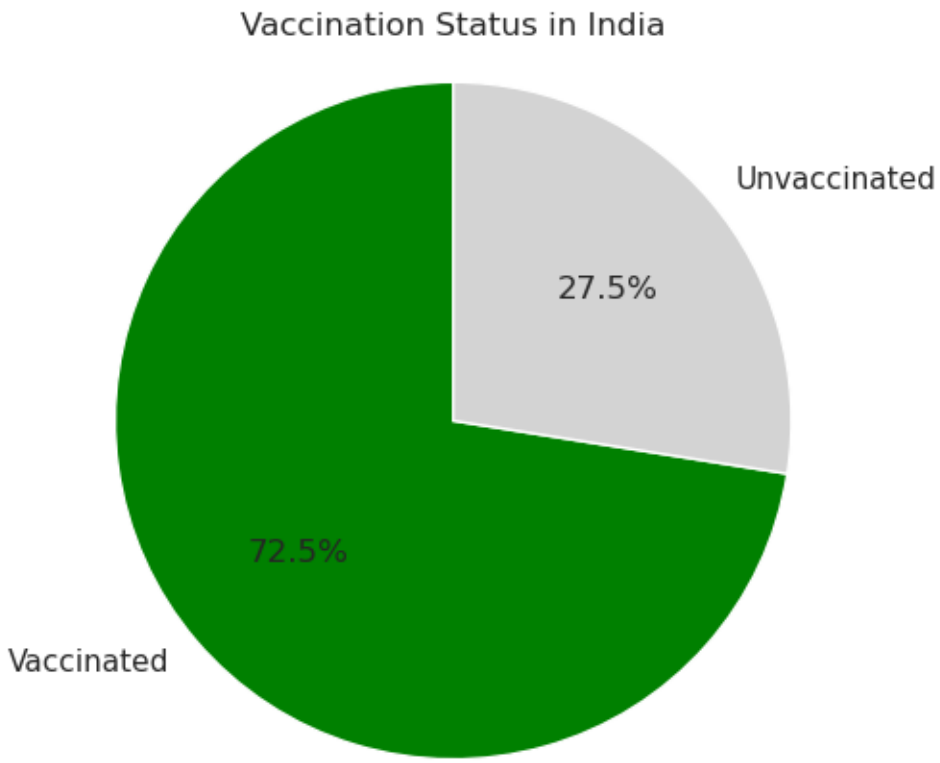


[19]:
```
# Get latest data per country
latest = df.sort_values("date").dropna(subset=["people_vaccinated",␣
 ↪"population"])
latest = latest.groupby("location").tail(1)

# Plot pie chart for each country
for _, row in latest.iterrows():
    vaccinated = row["people_vaccinated"]
    unvaccinated = row["population"] - vaccinated
    labels = ["Vaccinated", "Unvaccinated"]
    sizes = [vaccinated, unvaccinated]

    plt.figure(figsize=(5, 5))
    plt.pie(sizes, labels=labels, autopct="%1.1f%%", startangle=90,␣
 ↪colors=["green", "lightgray"])
    plt.title(f"Vaccination Status in {row['location']}")
    plt.axis("equal")
```

```
    plt.show()
```

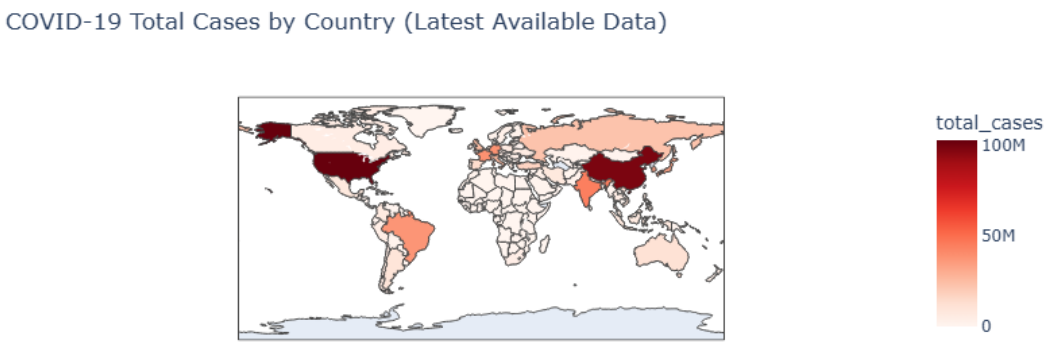

Vaccination Status in Kenya
Vaccinated
26.8%
73.2%
Unvaccinated


Vaccination Status in United States
Unvaccinated
20.1%
79.9%
Vaccinated



[20]:
```
import pandas as pd
import plotly.express as px

# Load the dataset (including iso_code)
use_cols = ["iso_code", "location", "date", "total_cases", "total_vaccinations"]
df = pd.read_csv("owid-covid-data.csv", usecols=use_cols)

# Drop aggregates like continents or "World"
df = df[df["iso_code"].str.len() == 3]

# Convert date to datetime
df["date"] = pd.to_datetime(df["date"])

# Get latest data per country
latest_df = df.sort_values("date").dropna(subset=["total_cases"]).
 ↪groupby("iso_code").tail(1)

# Optional: Choose metric to visualize
metric = "total_cases"  # Change to "total_vaccinations" for vaccination map

# Create choropleth
```

```
fig = px.choropleth(
    latest_df,
    locations="iso_code",
    color=metric,
    hover_name="location",
    color_continuous_scale="Reds",
    title=f"COVID-19 {metric.replace('_', ' ').title()} by Country (Latest␣
 ↪Available Data)"
)

fig.show()
```



[ ]: