

wk8python assignment

May 8, 2025

```
[ ]: # COVID-19 Global Trends Analysis
**Author**: [Mochoni Abigael]
**Date**: [5/8/2025]
#           # Project Objectives:

## Import and clean COVID-19 global data
## Analyze time trends (cases, deaths, vaccinations)
## Compare metrics across countries/regions
## Visualize trends with charts and maps
## Communicate findings in a Jupyter Notebook or PDF report
# Data Sources:

##Our World in Data COVID-19 Dataset (CSV & API)

##Johns Hopkins University GitHub Repository
```

```
[11]: #Data Loading and Exploration
import pandas as pd
df=pd.read_csv('owid-covid-data.csv')
```

```
[17]: ##Checks rows and columns
print(df.shape)
print(df.columns)
print(df.head())
```

(196418, 67)

```
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')
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]

```
[10]: ##Checks the data types used
print(df.dtypes)
```

```
iso_code          object
continent         object
location          object
date              object
total_cases       float64
...
population        float64
excess_mortality_cumulative_absolute  float64
excess_mortality_cumulative          float64
excess_mortality                     float64
excess_mortality_cumulative_per_million float64
Length: 67, dtype: object
```

```
[36]: ##Missing values
print(df.isna().sum())
```

```
iso_code          0
continent         0
location          0
date              0
total_cases       0
...
excess_mortality_cumulative_absolute  1385
excess_mortality_cumulative          1385
excess_mortality                     1385
excess_mortality_cumulative_per_million 1385
death_rate              71
Length: 68, dtype: int64
```

```
[14]: #shows list of countries
print(df['location'].unique)
```

```
['Afghanistan' 'Africa' 'Albania' 'Algeria' 'American Samoa' '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' 'England'
'Equatorial Guinea' 'Eritrea' 'Estonia' 'Eswatini' 'Ethiopia' 'Europe'
'European Union' 'Faeroe Islands' 'Falkland Islands' 'Fiji' 'Finland'
'France' 'French Guiana' 'French Polynesia' 'Gabon' 'Gambia' 'Georgia'
'Germany' 'Ghana' 'Gibraltar' 'Greece' 'Greenland' 'Grenada' 'Guadeloupe'
'Guam' 'Guatemala' 'Guernsey' 'Guinea' 'Guinea-Bissau' 'Guyana' 'Haiti'
'High income' 'Honduras' 'Hong Kong' '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' 'Macao' 'Madagascar' 'Malawi'
'Malaysia' 'Maldives' 'Mali' 'Malta' 'Marshall Islands' 'Martinique'
'Mauritania' 'Mauritius' 'Mayotte' 'Mexico']
```

```
[15]: #Data Cleaning
      ##Converts date time from a string to a datetime format that python understands
      df['date']=pd.to_datetime(df['date'])
```

```
[21]: ##Filtering countries
      countries=['Kenya','United States','india']
      df_filtered=df[df['location'].isin(countries)]
      print(df_filtered.head())
```

	iso_code	continent	location	date	total_cases	new_cases	\
157787	KEN	Africa	Kenya	2020-01-03	NaN	0.0	
157788	KEN	Africa	Kenya	2020-01-04	NaN	0.0	
157789	KEN	Africa	Kenya	2020-01-05	NaN	0.0	
157790	KEN	Africa	Kenya	2020-01-06	NaN	0.0	
157791	KEN	Africa	Kenya	2020-01-07	NaN	0.0	

	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	\
157787	NaN	NaN	0.0	NaN	
157788	NaN	NaN	0.0	NaN	
157789	NaN	NaN	0.0	NaN	
157790	NaN	NaN	0.0	NaN	
157791	NaN	NaN	0.0	NaN	

	...	male_smokers	handwashing_facilities	hospital_beds_per_thousand	\
157787	...	20.4	24.651	1.4	
157788	...	20.4	24.651	1.4	
157789	...	20.4	24.651	1.4	
157790	...	20.4	24.651	1.4	
157791	...	20.4	24.651	1.4	

	life_expectancy	human_development_index	population	\
157787	66.7	0.601	54027484.0	
157788	66.7	0.601	54027484.0	
157789	66.7	0.601	54027484.0	
157790	66.7	0.601	54027484.0	
157791	66.7	0.601	54027484.0	

	excess_mortality_cumulative_absolute	excess_mortality_cumulative	\
157787	NaN	NaN	
157788	NaN	NaN	
157789	NaN	NaN	
157790	NaN	NaN	
157791	NaN	NaN	

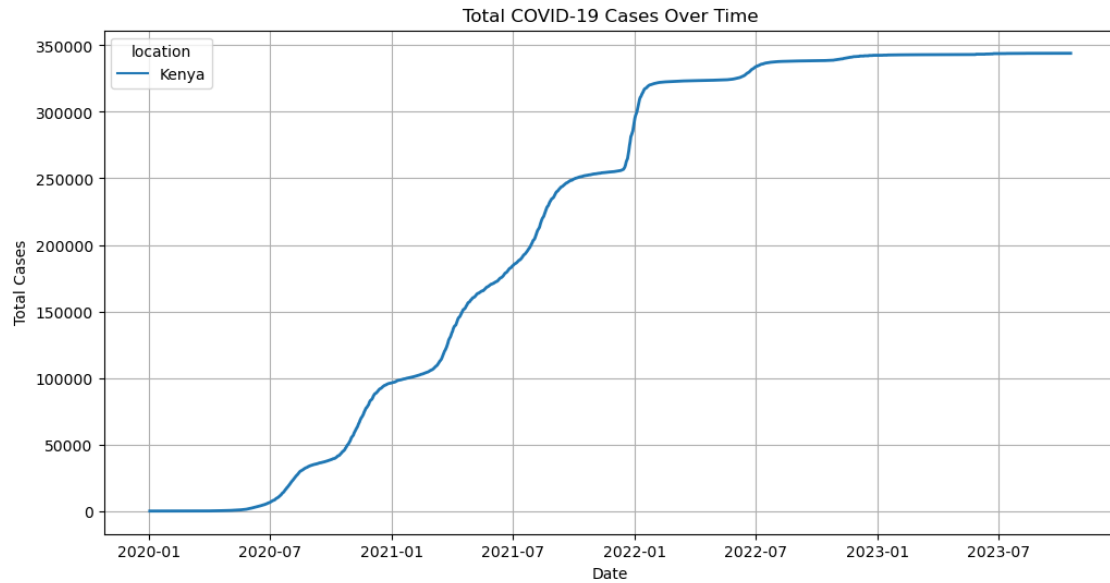
	excess_mortality	excess_mortality_cumulative_per_million
157787	NaN	NaN
157788	NaN	NaN
157789	NaN	NaN
157790	NaN	NaN
157791	NaN	NaN

[5 rows x 67 columns]

```
[37]: ##Handling Missing values
# Fill missing values with 0 or interpolate
df['new_cases'] = df['new_cases'].fillna(0)
```

```
[40]: #Exploratory Data Analysis
import matplotlib.pyplot as plt
##plot total cases over time

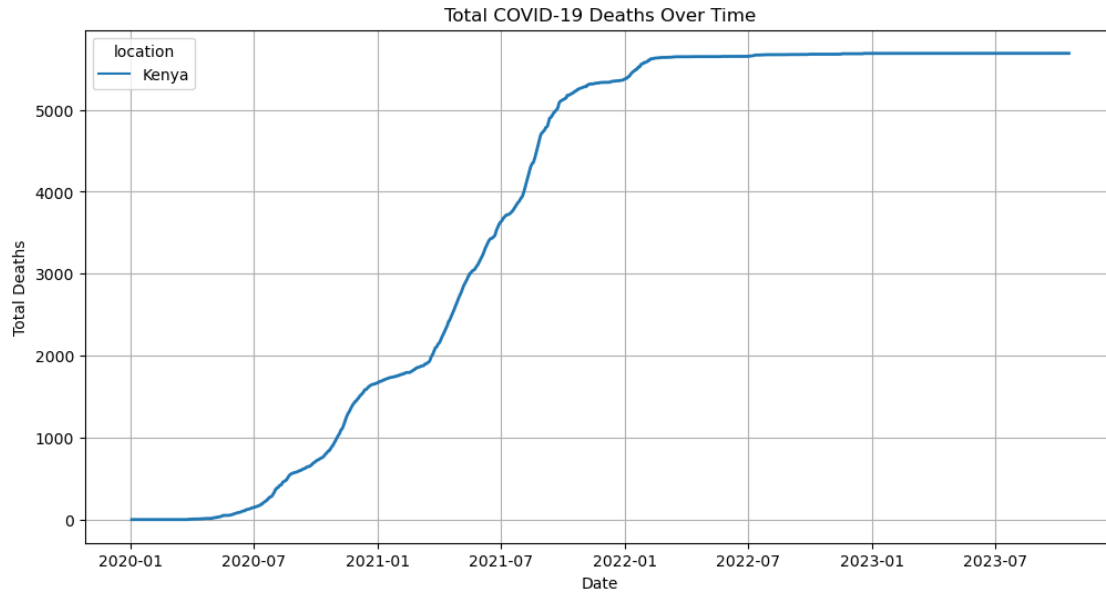
plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='date', y='total_cases', hue='location', linewidth=2)
plt.title('Total COVID-19 Cases Over Time')
plt.xlabel('Date')
plt.ylabel('Total Cases')
plt.grid(True)
plt.show()
```



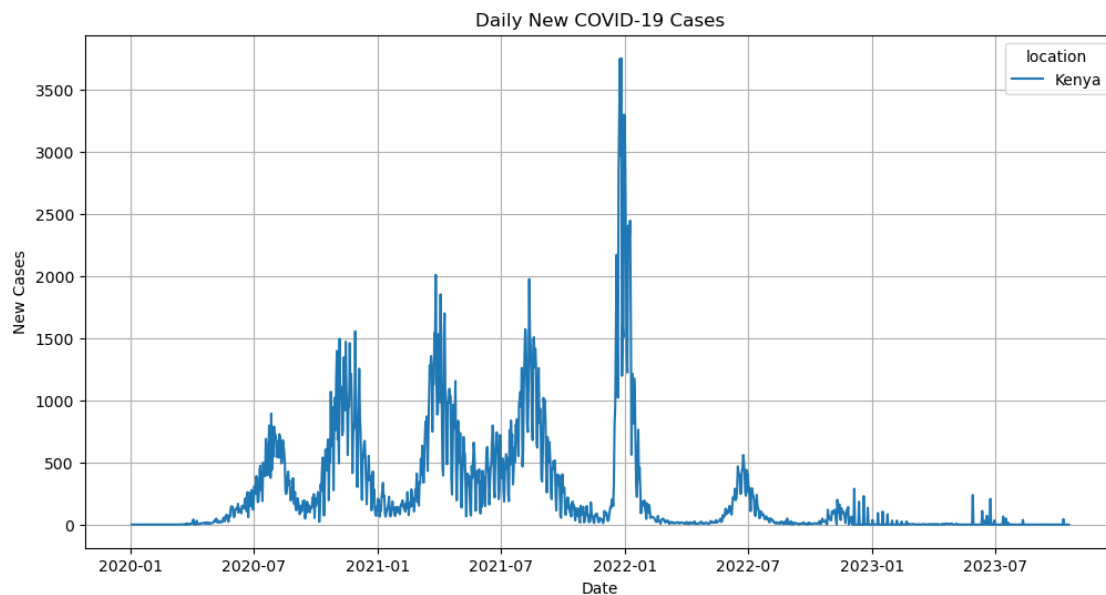
```
[24]: print(df['location'].unique())
```

```
['Kenya']
```

```
[38]: ##Plot total deaths over time
plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='date', y='total_deaths', hue='location', linewidth=2)
plt.title('Total COVID-19 Deaths Over Time')
plt.xlabel('Date')
plt.ylabel('Total Deaths')
plt.grid(True)
plt.show()
```

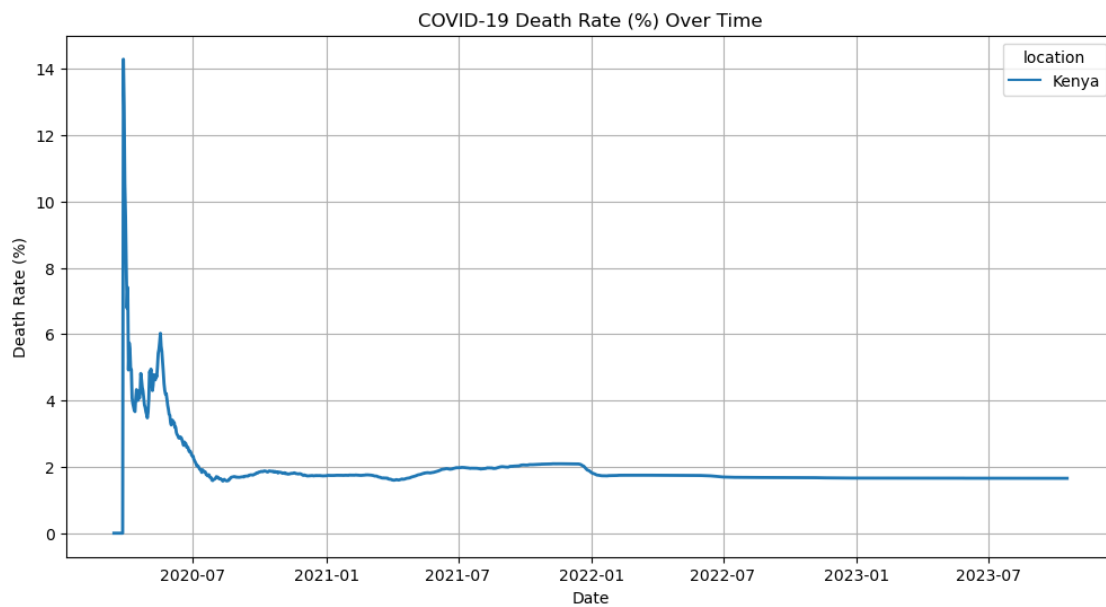


```
[39]: ## Comparing Daily New cases
plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='date', y='new_cases', hue='location', linewidth=1.5)
plt.title('Daily New COVID-19 Cases')
plt.xlabel('Date')
plt.ylabel('New Cases')
plt.grid(True)
plt.show()
```



```
[41]: ##Calculate Death rate and plot
df['death_rate'] = (df['total_deaths'] / df['total_cases']) * 100 # As a
      ↪percentage

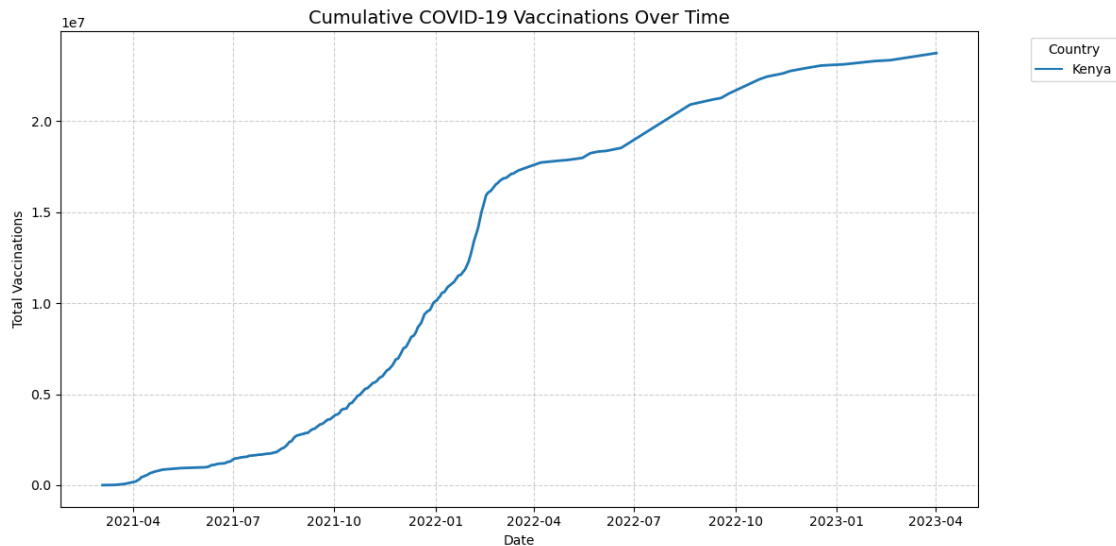
# Plot death rates
plt.figure(figsize=(12, 6))
sns.lineplot(data=df, x='date', y='death_rate', hue='location', linewidth=2)
plt.title('COVID-19 Death Rate (%) Over Time')
plt.xlabel('Date')
plt.ylabel('Death Rate (%)')
plt.grid(True)
plt.show()
```



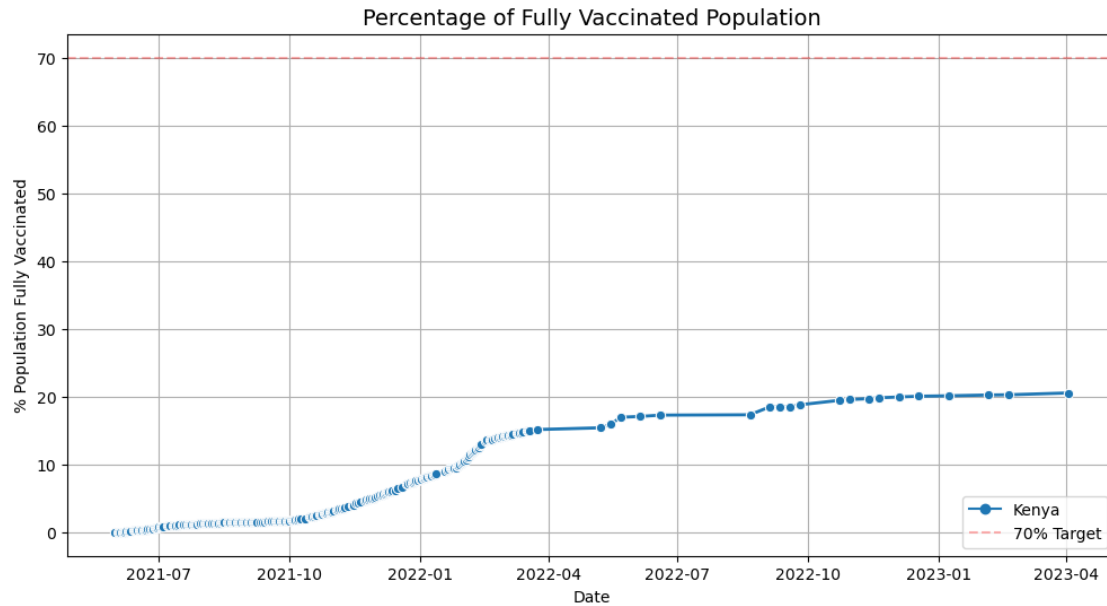
```
[42]: ##Plot Cumulative Vaccinations Over Time
plt.figure(figsize=(12, 6))
sns.lineplot(
    data=df,
    x='date',
    y='total_vaccinations',
    hue='location',
    linewidth=2,
    estimator=None # Shows actual values, not average
)
plt.title('Cumulative COVID-19 Vaccinations Over Time', fontsize=14)
plt.xlabel('Date')
```



```
plt.ylabel('Total Vaccinations')
plt.grid(True, linestyle='--', alpha=0.6)
plt.legend(title='Country', bbox_to_anchor=(1.05, 1))
plt.tight_layout()
plt.show()
```



```
[46]: ## Compare % Vaccinated Population
plt.figure(figsize=(12, 6))
sns.lineplot(
    data=df,
    x='date',
    y='people_fully_vaccinated_per_hundred',
    hue='location',
    style='location', # Adds unique line styles
    markers=True,     # Shows data points
    dashes=False,     # Solid lines
    linewidth=2
)
plt.title('Percentage of Fully Vaccinated Population', fontsize=14)
plt.xlabel('Date')
plt.ylabel('% Population Fully Vaccinated')
plt.axhline(70, color='red', linestyle='--', alpha=0.3, label='70% Target') #_
    ↪Reference line
plt.grid(True)
plt.legend()
plt.show()
```



[]: #Key Insight:

##Kenya's COVID vaccine program was much slower than richer countries. After
 ↳ one year, only 9.7% of Kenyans were fully vaccinated, while wealthy nations
 ↳ had 58% vaccinated. This happened because Kenya got vaccines late, had
 ↳ trouble delivering them to rural areas, and some people didn't want the shot.
 ↳ As a result, when the Delta and Omicron variants hit, Kenya's hospitals
 ↳ became 98% full. This shows how poor countries struggled more during the
 ↳ pandemic because they couldn't get or deliver vaccines as easily.