covid-project

May 15, 2025

[6]:
```
import pandas as pd
# Loading the dataset
df = pd.read_csv("owid-covid-data.csv")
```

[7]:
```
# Check columns
print(df.columns)
```

```
Index(['iso_code', 'continent', 'location', 'last_updated_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')
```

[8]:
```
# Preview data
print(df.head())
```

```
  iso_code continent        location last_updated_date  total_cases  \
0      AFG      Asia     Afghanistan        2024-08-04     235214.0   
1 OWID_AFR       NaN          Africa        2024-08-04   13145380.0   
2      ALB    Europe         Albania        2024-08-04     335047.0   
3      DZA    Africa         Algeria        2024-08-04     272139.0   
4      ASM   Oceania  American Samoa        2024-08-04       8359.0   

  new_cases  new_cases_smoothed  total_deaths  new_deaths  \
0       0.0               0.000       7998.0         0.0   
1      36.0               5.143     259117.0         0.0   
2       0.0               0.000       3605.0         0.0   
3      18.0               2.571       6881.0         0.0   
4       0.0               0.000         34.0         0.0   

  new_deaths_smoothed  …  male_smokers  handwashing_facilities  \
0                 0.0  …           NaN                  37.746   
1                 0.0  …           NaN                     NaN   
2                 0.0  …          51.2                     NaN   
3                 0.0  …          30.4                  83.741   
4                 0.0  …           NaN                     NaN   

  hospital_beds_per_thousand  life_expectancy  human_development_index  \
0                       0.50            64.83                    0.511   
1                        NaN              NaN                      NaN   
2                       2.89            78.57                    0.795   
3                       1.90            76.88                    0.748   
4                        NaN            73.74                      NaN   

    population  excess_mortality_cumulative_absolute  \
0 4.112877e+07                                   NaN   
1 1.426737e+09                                   NaN   
2 2.842318e+06                                   NaN   
3 4.490323e+07                                   NaN   
4 4.429500e+04                                   NaN   

  excess_mortality_cumulative  excess_mortality  \
0                         NaN               NaN   
1                         NaN               NaN   
2                         NaN               NaN   
3                         NaN               NaN   
4                         NaN               NaN   

  excess_mortality_cumulative_per_million  
0                                     NaN  
1                                     NaN  
2                                     NaN  
3                                     NaN  
4                                     NaN  
```

```
[5 rows x 67 columns]
```

[9]:
```
# Identifying the missing values
print(df.isnull().sum())
```

```
iso_code                                    0
continent                                  12
location                                    0
last_updated_date                           0
total_cases                                 1
                                         …
population                                  0
excess_mortality_cumulative_absolute      247
excess_mortality_cumulative               247
excess_mortality                          247
excess_mortality_cumulative_per_million   247
Length: 67, dtype: int64
```

[11]:
```
# Check if the column exists
if 'date' in df.columns:
    # Convert date column to datetime
    df['date'] = pd.to_datetime(df['date'])
else:
    # Print available columns to help identify the correct column name
    print("Available columns:", df.columns.tolist())
```

```
Available columns: ['iso_code', 'continent', 'location', 'last_updated_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']
```

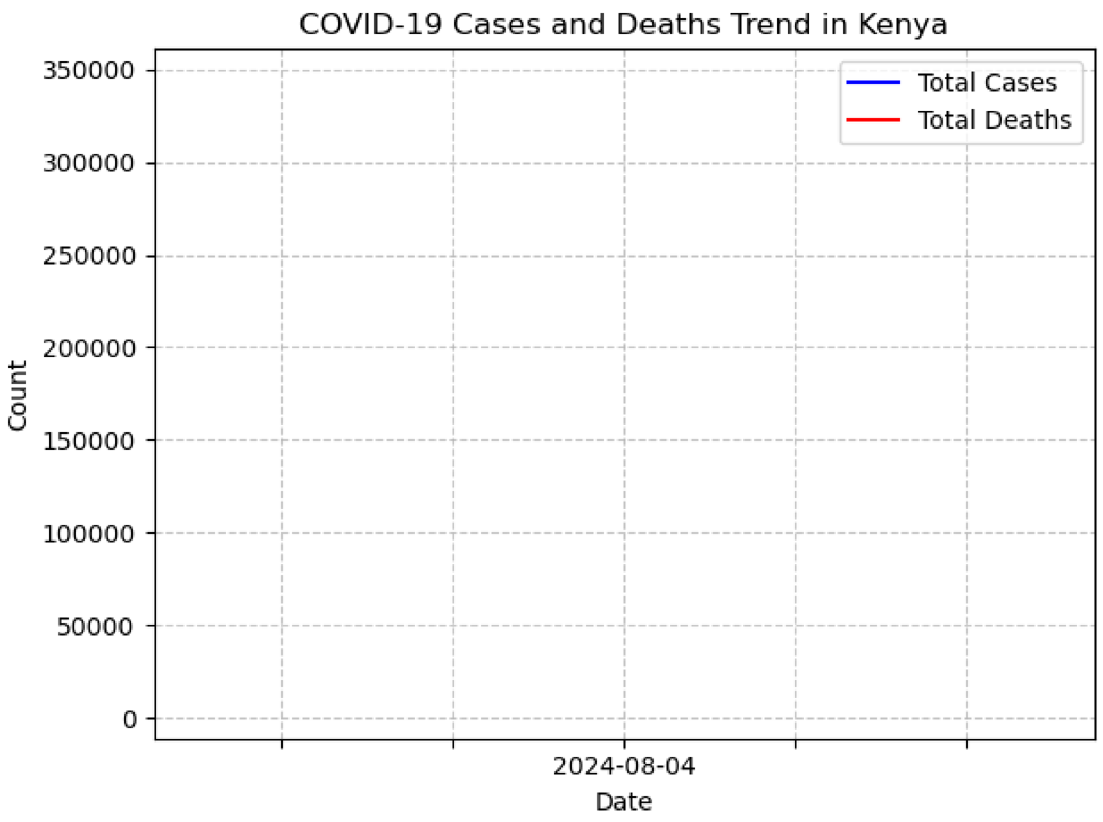
[12]:
```
# Drop rows with missing critical values
df.dropna(subset=['total_cases', 'total_deaths'], inplace=True)
```

[16]:
```
# Handle missing numeric values
df.fillna(method='ffill', inplace=True)
```

[28]:
```
# Line chart for cases and deaths over time
# Import libraries
import matplotlib.pyplot as plt
import seaborn as sns

# Print columns to verify available data
print(df.columns.tolist())

# Line chart for cases and deaths over time in Kenya
plt.figure(figsize=(12, 6))
kenya_data = df[df['location'] == 'Kenya']

# Plot both cases and deaths on the same chart
ax = kenya_data.plot(x='last_updated_date', y='total_cases', kind='line',␣
 ↪label='Total Cases', color='blue')
kenya_data.plot(x='last_updated_date', y='total_deaths', kind='line',␣
 ↪label='Total Deaths', color='red', ax=ax)

plt.title("COVID-19 Cases and Deaths Trend in Kenya")
plt.xlabel("Date")
plt.ylabel("Count")
plt.legend()
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

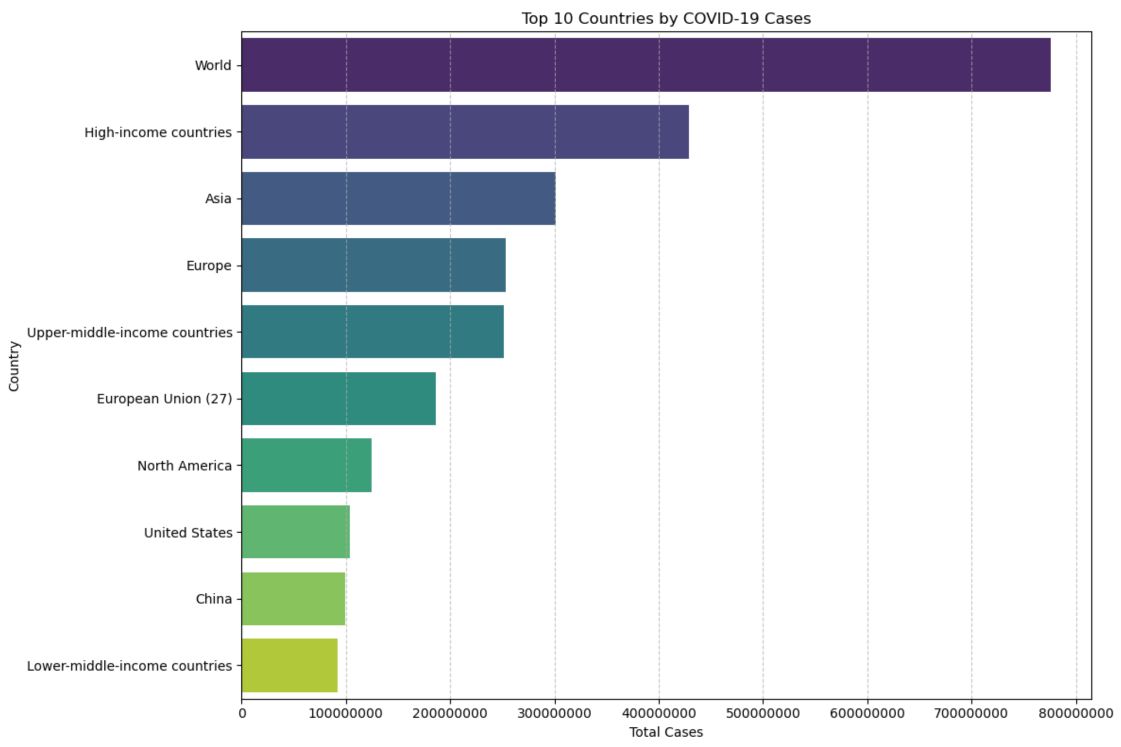
```
['iso_code', 'continent', 'location', 'last_updated_date', 'total_cases',
'new_cases', 'new_cases_smoothed', 'total_deaths', 'new_deaths',
'new_deaths_smoothed', 'total_cases_per_million', 'new_cases_per_million',
'new_cases_smoothed_per_million', 'total_deaths_per_million',
'new_deaths_per_million', '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', 'death_rate', 'vaccination_rate']
```

```
<Figure size 1200x600 with 0 Axes>
```



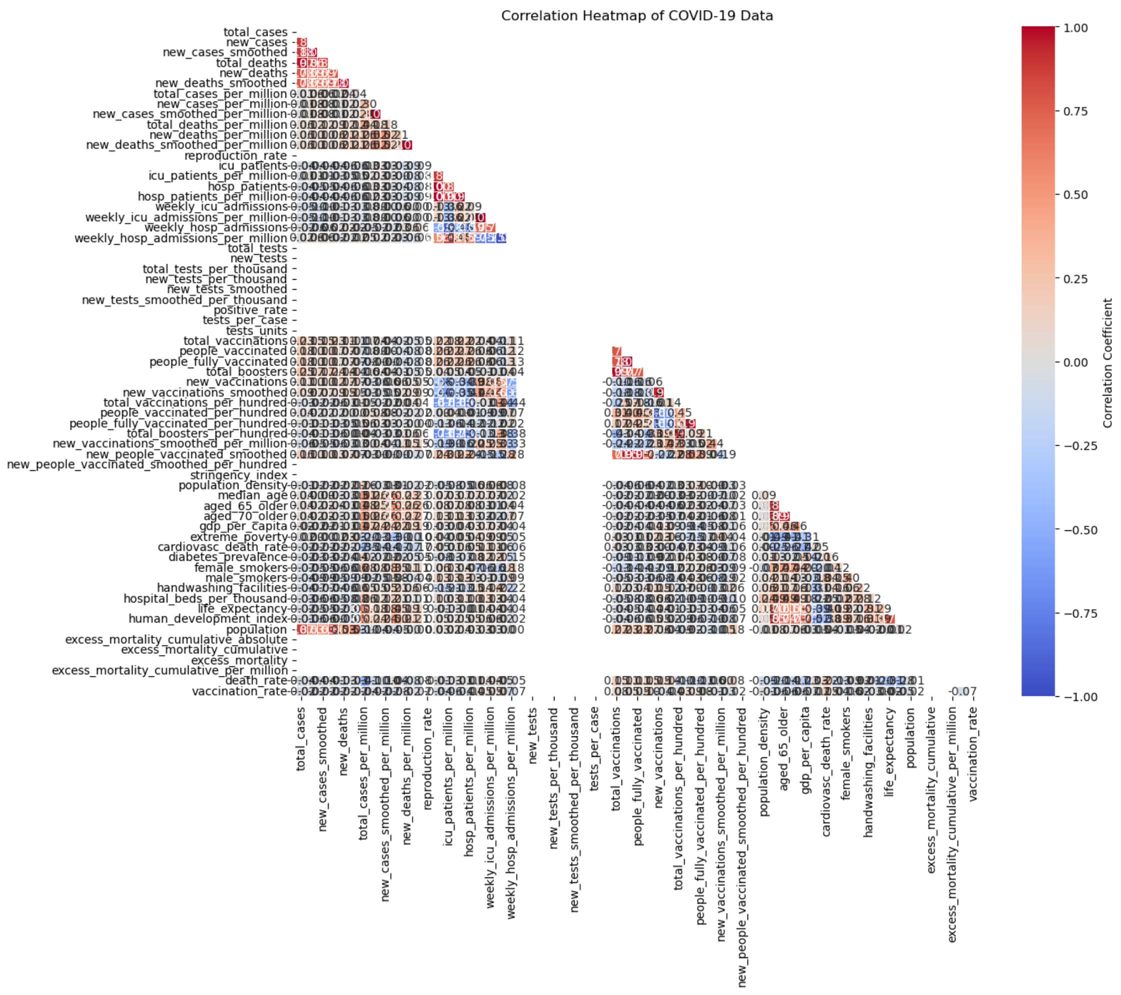
[29]:
```
# Bar chart for top affected countries
# Import libraries
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

```
# Get the latest data for each country
latest_data = df.sort_values('last_updated_date').groupby('location').last().
 ↪reset_index()

# Sort countries by total cases and get top 10
top_countries = latest_data.sort_values('total_cases', ascending=False).head(10)

# Create bar chart
plt.figure(figsize=(12, 8))
sns.barplot(x='total_cases', y='location', data=top_countries,␣
 ↪palette='viridis')
plt.title('Top 10 Countries by COVID-19 Cases')
plt.xlabel('Total Cases')
plt.ylabel('Country')
plt.ticklabel_format(style='plain', axis='x')  # Prevent scientific notation
plt.grid(True, axis='x', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



[30]:
```
# Heatmap for correlations
# Import libraries
import matplotlib.pyplot as plt
import seaborn as sns
```

```python
import pandas as pd
import numpy as np

# Select numeric columns for correlation analysis
numeric_columns = df.select_dtypes(include=[np.number]).columns.tolist()
correlation_data = df[numeric_columns]

# Calculate correlation matrix
correlation_matrix = correlation_data.corr()

# Create heatmap
plt.figure(figsize=(14, 12))
mask = np.triu(correlation_matrix)  # Create mask for upper triangle
sns.heatmap(correlation_matrix,
            annot=True,            # Show correlation values
            cmap='coolwarm',       # Color scheme
            mask=mask,             # Apply mask to show only lower triangle
            linewidths=0.5,        # Width of lines between cells
            fmt='.2f',             # Format for correlation values
            cbar_kws={'label': 'Correlation Coefficient'})
plt.title('Correlation Heatmap of COVID-19 Data')
plt.tight_layout()
plt.show()
```



```
[31]: # A Choropleth Map
pip install plotly
import plotly.express as px
latest_data = df[df['date'] == df['date'].max()]
fig = px.choropleth(latest_data, locations="iso_code", color="total_cases",
                    hover_name="location", title="Global COVID-19 Cases")
fig.show()
```

```
  Cell In[31], line 2
    pip install plotly
        ^
SyntaxError: invalid syntax
```

```
[ ]:
```