```python
import pandas as pd

# Define file paths
base_path = '/kaggle/input/corona-virus-report/'

# Load each CSV file
worldometer_df = pd.read_csv(base_path + 'worldometer_data.csv')
usa_county_df = pd.read_csv(base_path + 'usa_county_wise.csv')
full_grouped_df = pd.read_csv(base_path + 'full_grouped.csv')
clean_complete_df = pd.read_csv(base_path +
'covid_19_clean_complete.csv')
country_latest_df = pd.read_csv(base_path + 'country_wise_latest.csv')
day_wise_df = pd.read_csv(base_path + 'day_wise.csv')

datasets = {
    'Worldometer': worldometer_df,
    'USA County-wise': usa_county_df,
    'Full Grouped': full_grouped_df,
    'Clean Complete': clean_complete_df,
    'Country-wise Latest': country_latest_df,
    'Day-wise': day_wise_df,
}

for name, df in datasets.items():
    print(f"\n□ {name} Dataset ----------------------------")
    print("□ Shape:", df.shape)
    print("□ Columns:", df.columns.tolist())
    print("□ Head:\n", df.head(2))
```

```
□ Worldometer Dataset ----------------------------
□ Shape: (209, 16)
□ Columns: ['Country/Region', 'Continent', 'Population', 'TotalCases',
'NewCases', 'TotalDeaths', 'NewDeaths', 'TotalRecovered',
'NewRecovered', 'ActiveCases', 'Serious,Critical', 'Tot Cases/1M pop',
'Deaths/1M pop', 'TotalTests', 'Tests/1M pop', 'WHO Region']
□ Head:
   Country/Region      Continent   Population  TotalCases  NewCases  \
0            USA  North America  331198130.0     5032179       NaN
1         Brazil  South America  212710692.0     2917562       NaN

   TotalDeaths  NewDeaths  TotalRecovered  NewRecovered
ActiveCases  \
0     162804.0        NaN       2576668.0           NaN    2292707.0

1      98644.0        NaN       2047660.0           NaN     771258.0


   Serious,Critical  Tot Cases/1M pop  Deaths/1M pop  TotalTests  \
0           18296.0           15194.0          492.0  63139605.0
```

```
1            8318.0           13716.0          464.0  13206188.0

   Tests/1M pop WHO Region
0      190640.0   Americas
1       62085.0   Americas

□ USA County-wise Dataset -----------------------------
□ Shape: (627920, 14)
□ Columns: ['UID', 'iso2', 'iso3', 'code3', 'FIPS', 'Admin2',
'Province_State', 'Country_Region', 'Lat', 'Long_', 'Combined_Key',
'Date', 'Confirmed', 'Deaths']
□ Head:
    UID iso2 iso3  code3  FIPS Admin2  Province_State Country_Region
Lat  \
0   16   AS  ASM     16  60.0    NaN  American Samoa             US -
14.2710
1  316   GU  GUM    316  66.0    NaN            Guam             US
13.4443

      Long_        Combined_Key     Date  Confirmed  Deaths
0 -170.1320  American Samoa, US  1/22/20          0       0
1  144.7937            Guam, US  1/22/20          0       0

□ Full Grouped Dataset -----------------------------
□ Shape: (35156, 10)
□ Columns: ['Date', 'Country/Region', 'Confirmed', 'Deaths',
'Recovered', 'Active', 'New cases', 'New deaths', 'New recovered',
'WHO Region']
□ Head:
          Date Country/Region  Confirmed  Deaths  Recovered  Active
New cases  \
0  2020-01-22    Afghanistan          0       0          0       0
0
1  2020-01-22        Albania          0       0          0       0
0

   New deaths  New recovered             WHO Region
0           0              0  Eastern Mediterranean
1           0              0                 Europe

□ Clean Complete Dataset -----------------------------
□ Shape: (49068, 10)
□ Columns: ['Province/State', 'Country/Region', 'Lat', 'Long', 'Date',
'Confirmed', 'Deaths', 'Recovered', 'Active', 'WHO Region']
□ Head:
   Province/State Country/Region       Lat       Long        Date
Confirmed  \
0            NaN    Afghanistan  33.93911  67.709953  2020-01-22
0
1            NaN        Albania  41.15330  20.168300  2020-01-22
```

```
0

   Deaths  Recovered  Active             WHO Region
0       0          0       0  Eastern Mediterranean
1       0          0       0                 Europe

□ Country-wise Latest Dataset -----------------------------
□ Shape: (187, 15)
□ Columns: ['Country/Region', 'Confirmed', 'Deaths', 'Recovered',
'Active', 'New cases', 'New deaths', 'New recovered', 'Deaths / 100
Cases', 'Recovered / 100 Cases', 'Deaths / 100 Recovered', 'Confirmed
last week', '1 week change', '1 week % increase', 'WHO Region']
□ Head:
   Country/Region  Confirmed  Deaths  Recovered  Active  New cases
New deaths  \
0    Afghanistan      36263    1269      25198    9796        106
10
1        Albania       4880     144       2745    1991        117
6

   New recovered  Deaths / 100 Cases  Recovered / 100 Cases  \
0             18                3.50                  69.49
1             63                2.95                  56.25

   Deaths / 100 Recovered  Confirmed last week  1 week change  \
0                    5.04                35526            737
1                    5.25                 4171            709

   1 week % increase             WHO Region
0               2.07  Eastern Mediterranean
1              17.00                 Europe

□ Day-wise Dataset -----------------------------
□ Shape: (188, 12)
□ Columns: ['Date', 'Confirmed', 'Deaths', 'Recovered', 'Active', 'New
cases', 'New deaths', 'New recovered', 'Deaths / 100 Cases',
'Recovered / 100 Cases', 'Deaths / 100 Recovered', 'No. of countries']
□ Head:
          Date  Confirmed  Deaths  Recovered  Active  New cases  New
deaths  \
0  2020-01-22        555      17         28     510          0
0
1  2020-01-23        654      18         30     606         99
1

   New recovered  Deaths / 100 Cases  Recovered / 100 Cases  \
0              0                3.06                   5.05
1              2                2.75                   4.59

   Deaths / 100 Recovered  No. of countries
```

```
0                   60.71                 6
1                   60.00                 8

/usr/local/lib/python3.11/dist-packages/pandas/io/formats/
format.py:1458: RuntimeWarning: invalid value encountered in greater
  has_large_values = (abs_vals > 1e6).any()
/usr/local/lib/python3.11/dist-packages/pandas/io/formats/format.py:14
59: RuntimeWarning: invalid value encountered in less
  has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals >
0)).any()
/usr/local/lib/python3.11/dist-packages/pandas/io/formats/format.py:14
59: RuntimeWarning: invalid value encountered in greater
  has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals >
0)).any()
```

I Choose Worldometer dataset from above

```
print(" Head:\n", worldometer_df.head())
```

```
 Head:
   country/region      continent    population  totalcases
newcases  \
0            USA  North America  3.311981e+08     5032179       NaN
1         Brazil  South America  2.127107e+08     2917562       NaN
2          India           Asia  1.381345e+09     2025409       NaN
3         Russia         Europe  1.459409e+08      871894       NaN
4   South Africa         Africa  5.938157e+07      538184       NaN

   totaldeaths  newdeaths  totalrecovered  newrecovered
activecases  \
0     162804.0        NaN       2576668.0           NaN    2292707.0

1      98644.0        NaN       2047660.0           NaN     771258.0

2      41638.0        NaN       1377384.0           NaN     606387.0

3      14606.0        NaN        676357.0           NaN     180931.0

4       9604.0        NaN        387316.0           NaN     141264.0


   serious,critical  tot_cases/1m_pop  deaths/1m_pop  totaltests  \
0           18296.0           15194.0          492.0  63139605.0
1            8318.0           13716.0          464.0  13206188.0
2            8944.0            1466.0           30.0  22149351.0
3            2300.0            5974.0          100.0  29716907.0
4             539.0            9063.0          162.0   3149807.0

   tests/1m_pop      who_region
```

```
0      190640.0        Americas
1       62085.0        Americas
2       16035.0  South-EastAsia
3      203623.0          Europe
4       53044.0          Africa
```

```
/usr/local/lib/python3.11/dist-packages/pandas/io/formats/
format.py:1458: RuntimeWarning: invalid value encountered in greater
  has_large_values = (abs_vals > 1e6).any()
/usr/local/lib/python3.11/dist-packages/pandas/io/formats/format.py:14
59: RuntimeWarning: invalid value encountered in less
  has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals >
0)).any()
/usr/local/lib/python3.11/dist-packages/pandas/io/formats/format.py:14
59: RuntimeWarning: invalid value encountered in greater
  has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals >
0)).any()
```

```python
print("\n Info:")
worldometer_df.info()
```

```
 Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209 entries, 0 to 208
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   country/region    209 non-null    object
 1   continent         208 non-null    object
 2   population        208 non-null    float64
 3   totalcases        209 non-null    int64
 4   newcases          4 non-null      float64
 5   totaldeaths       188 non-null    float64
 6   newdeaths         3 non-null      float64
 7   totalrecovered    205 non-null    float64
 8   newrecovered      3 non-null      float64
 9   activecases       205 non-null    float64
 10  serious,critical  122 non-null    float64
 11  tot_cases/1m_pop  208 non-null    float64
 12  deaths/1m_pop     187 non-null    float64
 13  totaltests        191 non-null    float64
 14  tests/1m_pop      191 non-null    float64
 15  who_region        184 non-null    object
dtypes: float64(12), int64(1), object(3)
memory usage: 26.3+ KB
```

```python
print("\n Describe:")
print(worldometer_df.describe(include='all'))
```

```
 Describe:
       country/region continent    population    totalcases
newcases  \
count             209       208  2.080000e+02  2.090000e+02
4.000000
unique            209         6           NaN           NaN
NaN
top               USA    Africa           NaN           NaN
NaN
freq                1        57           NaN           NaN
NaN
mean              NaN       NaN  3.041549e+07  9.171850e+04
1980.500000
std               NaN       NaN  1.047661e+08  4.325867e+05
3129.611424
min               NaN       NaN  8.010000e+02  1.000000e+01
20.000000
25%               NaN       NaN  9.663140e+05  7.120000e+02
27.500000
50%               NaN       NaN  7.041972e+06  4.491000e+03
656.000000
75%               NaN       NaN  2.575614e+07  3.689600e+04
2609.000000
max               NaN       NaN  1.381345e+09  5.032179e+06
6590.000000

          totaldeaths   newdeaths  totalrecovered  newrecovered
activecases  \
count      188.000000    3.000000    2.050000e+02      3.000000
2.050000e+02
unique            NaN         NaN             NaN           NaN
NaN
top               NaN         NaN             NaN           NaN
NaN
freq              NaN         NaN             NaN           NaN
NaN
mean      3792.590426  300.000000    5.887898e+04   1706.000000
2.766433e+04
std      15487.184877  451.199512    2.566984e+05   2154.779803
1.746327e+05
min          1.000000    1.000000    7.000000e+00     42.000000
0.000000e+00
25%         22.000000   40.500000    3.340000e+02    489.000000
8.600000e+01
50%        113.000000   80.000000    2.178000e+03    936.000000
8.990000e+02
75%        786.000000  449.500000    2.055300e+04   2538.000000
7.124000e+03
max     162804.000000  819.000000    2.576668e+06   4140.000000
```

```
2.292707e+06

        serious,critical  tot_cases/1m_pop  deaths/1m_pop
totaltests  \
count         122.000000        208.000000     187.000000
1.910000e+02
unique               NaN               NaN            NaN
NaN
top                  NaN               NaN            NaN
NaN
freq                 NaN               NaN            NaN
NaN
mean          534.393443       3196.024038      98.681176
1.402405e+06
std          2047.518613       5191.986457     174.956862
5.553367e+06
min             1.000000          3.000000       0.080000
6.100000e+01
25%             3.250000        282.000000       6.000000
2.575200e+04
50%            27.500000       1015.000000      29.000000
1.357020e+05
75%           160.250000       3841.750000      98.000000
7.576960e+05
max         18296.000000      39922.000000    1238.000000
6.313960e+07

         tests/1m_pop who_region
count      191.000000        184
unique            NaN          6
top               NaN     Europe
freq              NaN         55
mean     83959.366492        NaN
std     152730.591240        NaN
min          4.000000        NaN
25%       8956.500000        NaN
50%      32585.000000        NaN
75%      92154.500000        NaN
max     995282.000000        NaN

/usr/local/lib/python3.11/dist-packages/pandas/io/formats/
format.py:1458: RuntimeWarning: invalid value encountered in greater
  has_large_values = (abs_vals > 1e6).any()
/usr/local/lib/python3.11/dist-packages/pandas/io/formats/format.py:14
59: RuntimeWarning: invalid value encountered in less
  has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals >
0)).any()
/usr/local/lib/python3.11/dist-packages/pandas/io/formats/format.py:14
59: RuntimeWarning: invalid value encountered in greater
```

```
  has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals >
0)).any()
```

```
print("\n Missing Values:")
print(worldometer_df.isnull().sum())

worldometer_df = worldometer_df.dropna(thresh=3)

print("\n Duplicates:", worldometer_df.duplicated().sum())
worldometer_df = worldometer_df.drop_duplicates()
```

```
 Missing Values:
country/region        0
continent             1
population            1
totalcases            0
newcases            205
totaldeaths          21
newdeaths           206
totalrecovered        4
newrecovered        206
activecases           4
serious,critical     87
tot_cases/1m_pop      1
deaths/1m_pop        22
totaltests           18
tests/1m_pop         18
who_region           25
dtype: int64

 Duplicates: 0
```

```
worldometer_df.columns =
worldometer_df.columns.str.strip().str.lower().str.replace(' ', '_')
```

Handle Missing Values

```
worldometer_df['continent'] =
worldometer_df['continent'].fillna('Unknown')
worldometer_df['who_region'] =
worldometer_df['who_region'].fillna('Unknown')


worldometer_df = worldometer_df.dropna(subset=['population'])

fill_zeros = [
    'newcases', 'totaldeaths', 'newdeaths', 'totalrecovered',
'newrecovered',
    'activecases', 'serious,critical', 'tot_cases/1m_pop',
    'deaths/1m_pop', 'tests/1m_pop'
]
worldometer_df[fill_zeros] = worldometer_df[fill_zeros].fillna(0)


if worldometer_df['totaltests'].isnull().sum() > 0:
    median_tests = worldometer_df['totaltests'].median()
    worldometer_df['totaltests'] =
worldometer_df['totaltests'].fillna(median_tests)

print(" All missing values handled:")
print(worldometer_df.isnull().sum().sort_values(ascending=False))
```

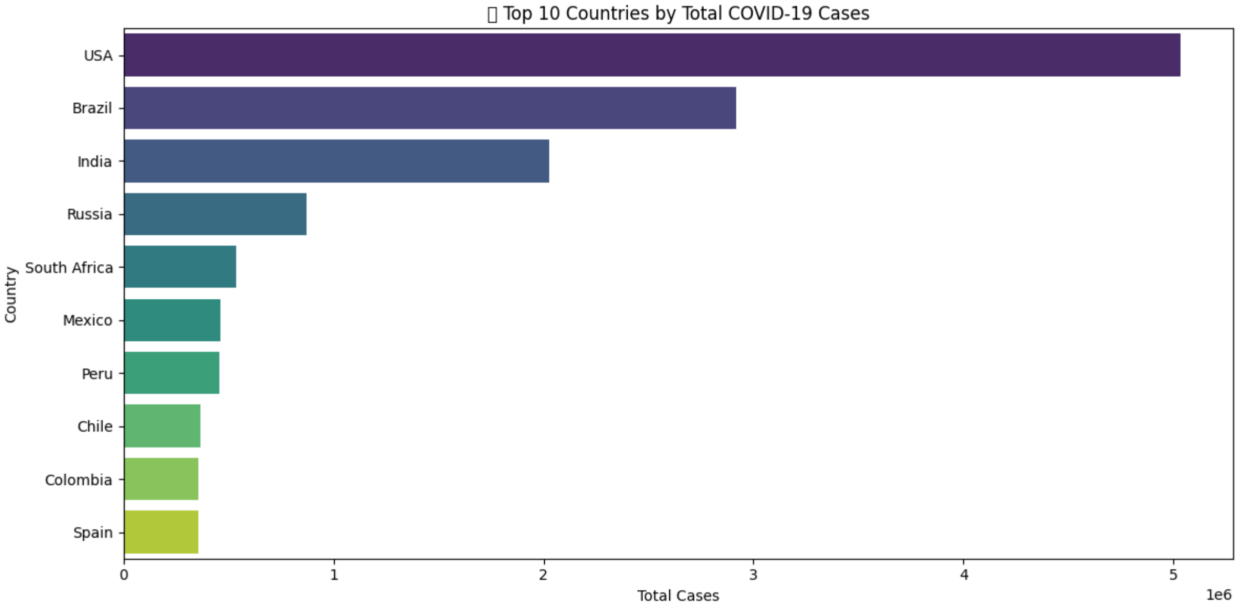
```
 All missing values handled:
country/region      0
continent           0
population          0
totalcases          0
newcases            0
totaldeaths         0
newdeaths           0
totalrecovered      0
newrecovered        0
activecases         0
serious,critical    0
tot_cases/1m_pop    0
deaths/1m_pop       0
totaltests          0
tests/1m_pop        0
who_region          0
dtype: int64
```

```
import matplotlib.pyplot as plt
import seaborn as sns
top_10_cases = worldometer_df.sort_values(by='totalcases',
ascending=False).head(10)

plt.figure(figsize=(12, 6))
sns.barplot(x='totalcases', y='country/region', data=top_10_cases,
palette='viridis')
plt.title('□ Top 10 Countries by Total COVID-19 Cases')
plt.xlabel('Total Cases')
plt.ylabel('Country')
plt.tight_layout()
plt.show()
```

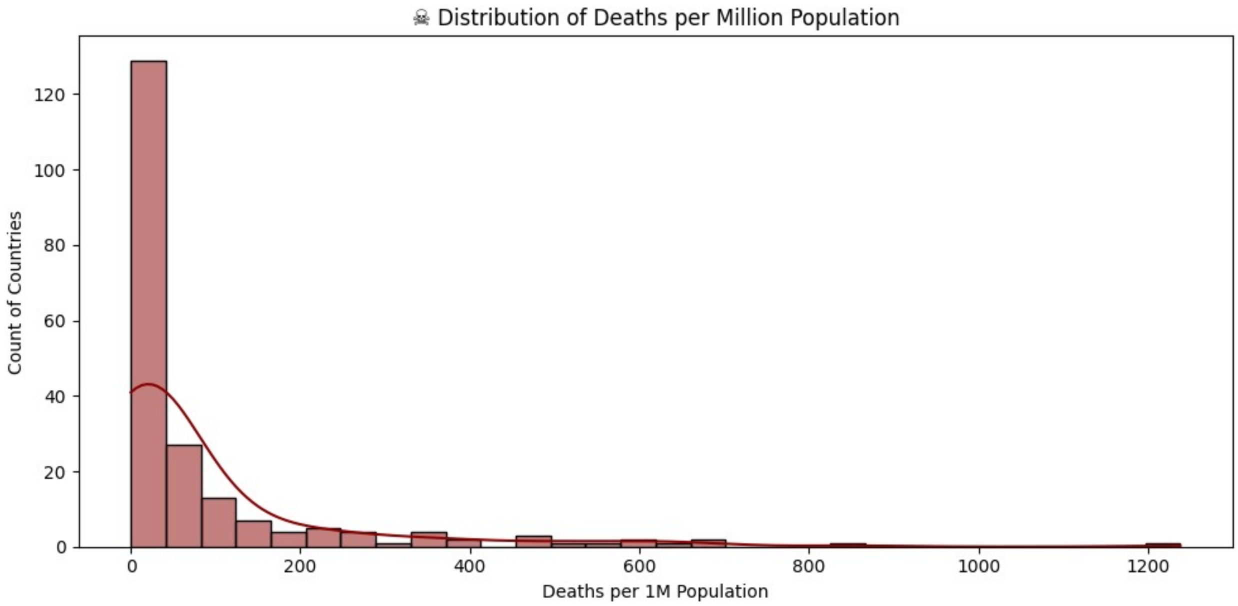
```
/tmp/ipykernel_36/3094867491.py:10: UserWarning: Glyph 128285 (\N{TOP
WITH UPWARDS ARROW ABOVE}) missing from current font.
```

```
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151
: UserWarning: Glyph 128285 (\N{TOP WITH UPWARDS ARROW ABOVE}) missing
from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```



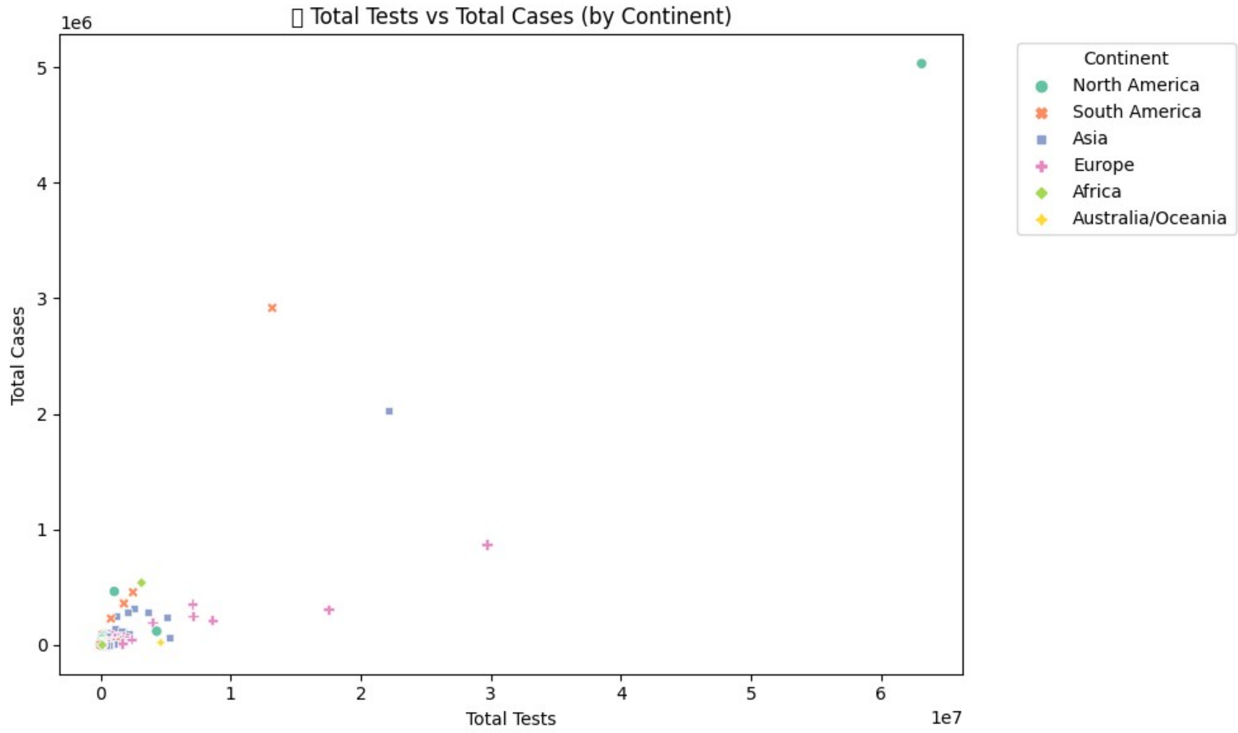
```
plt.figure(figsize=(10, 5))
sns.histplot(worldometer_df['deaths/1m_pop'], bins=30, kde=True,
color='darkred')
plt.title('☠ Distribution of Deaths per Million Population')
plt.xlabel('Deaths per 1M Population')
plt.ylabel('Count of Countries')
plt.tight_layout()
plt.show()
```

```
/usr/local/lib/python3.11/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



```python
plt.figure(figsize=(10, 6))
sns.scatterplot(
    x='totaltests', y='totalcases',
    data=worldometer_df,
    hue='continent', style='continent', palette='Set2'
)
plt.title('□ Total Tests vs Total Cases (by Continent)')
plt.xlabel('Total Tests')
plt.ylabel('Total Cases')
plt.legend(title='Continent', bbox_to_anchor=(1.05, 1), loc='upper
left')
plt.tight_layout()
plt.show()
```

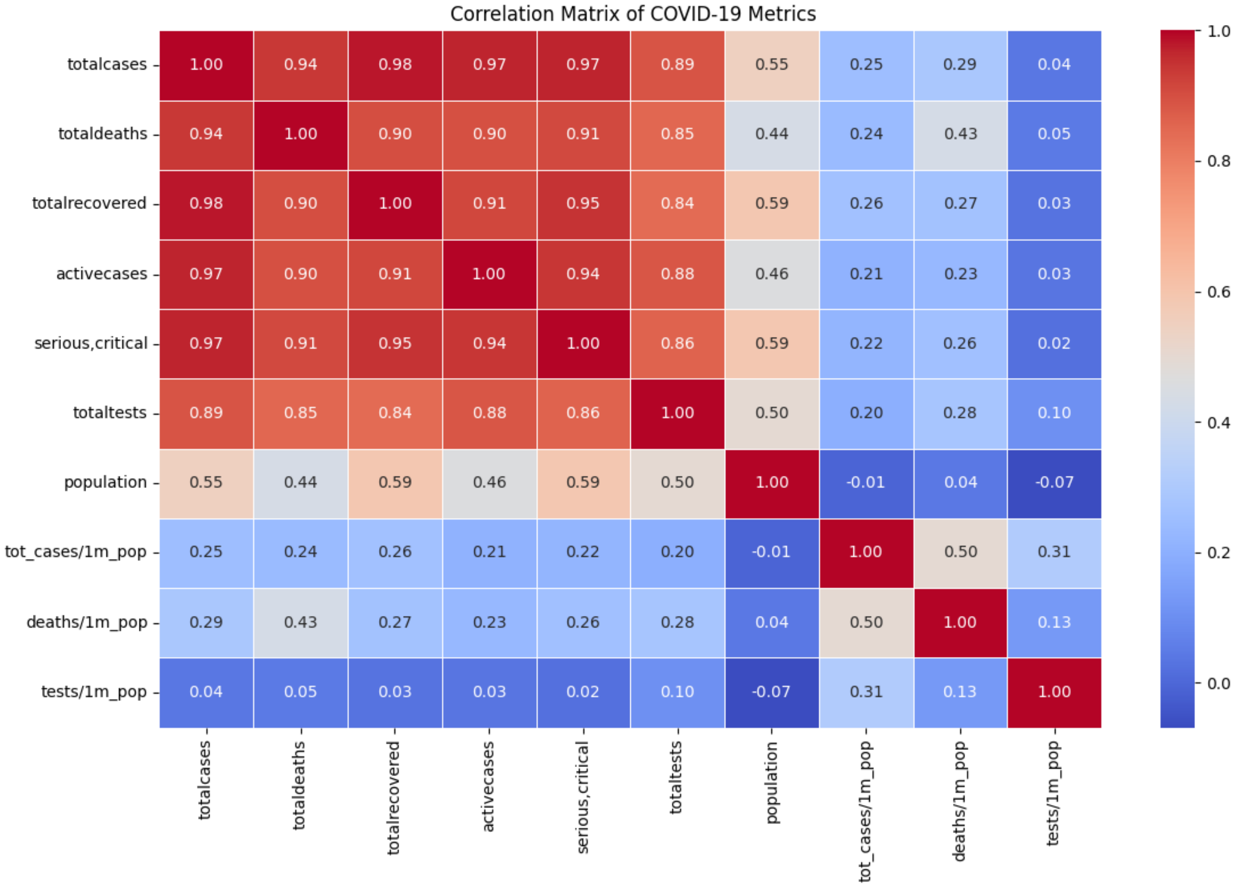
```
/tmp/ipykernel_36/3117219852.py:11: UserWarning: Glyph 129514 (\N{TEST
TUBE}) missing from current font.
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151
: UserWarning: Glyph 129514 (\N{TEST TUBE}) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```



```
numeric_cols = [
    'totalcases', 'totaldeaths', 'totalrecovered', 'activecases',
    'serious,critical', 'totaltests', 'population',
    'tot_cases/1m_pop', 'deaths/1m_pop', 'tests/1m_pop'
]

corr_matrix = worldometer_df[numeric_cols].corr()

plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f",
linewidths=0.5)
plt.title(' Correlation Matrix of COVID-19 Metrics')
plt.tight_layout()
plt.show()
```


Correlation Matrix of COVID-19 Metrics
totalcases 1.00 0.94 0.98 0.97 0.97 0.89 0.55 0.25 0.29 0.04
totaldeaths 0.94 1.00 0.90 0.90 0.91 0.85 0.44 0.24 0.43 0.05
totalrecovered 0.98 0.90 1.00 0.91 0.95 0.84 0.59 0.26 0.27 0.03
activecases 0.97 0.90 0.91 1.00 0.94 0.88 0.46 0.21 0.23 0.03
serious,critical 0.97 0.91 0.95 0.94 1.00 0.86 0.59 0.22 0.26 0.02
totaltests 0.89 0.85 0.84 0.88 0.86 1.00 0.50 0.20 0.28 0.10
population 0.55 0.44 0.59 0.46 0.59 0.50 1.00 -0.01 0.04 -0.07
tot_cases/1m_pop 0.25 0.24 0.26 0.21 0.22 0.20 -0.01 1.00 0.50 0.31
deaths/1m_pop 0.29 0.43 0.27 0.23 0.26 0.28 0.04 0.50 1.00 0.13
tests/1m_pop 0.04 0.05 0.03 0.03 0.02 0.10 -0.07 0.31 0.13 1.00
totalcases
totaldeaths
totalrecovered
activecases
serious,critical
totaltests
population
tot_cases/1m_pop
deaths/1m_pop
tests/1m_pop
1.0
0.8
0.6
0.4
0.2
0.0