```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import skew, kurtosis, shapiro
import statsmodels.api as sm

# Load datasets
covidtotals = pd.read_csv('covidtotals.csv')
nls97_dummy_800 = pd.read_csv('nls97_dummy_800.csv')
```

```
covidtotals
```

	iso_code	lastdate	location	total_cases	total_deaths	total_cases_pm	total_deaths_pm	population	pop_density	median_age
0	AFG	2024-02-04	Afghanistan	231539.0	7982.0	5629.611	194.073	41128772	54.422	18.6
1	ALB	2024-01-28	Albania	334863.0	3605.0	117813.348	1268.331	2842318	104.871	38.0
2	DZA	2023-12-03	Algeria	272010.0	6881.0	6057.694	153.241	44903228	17.348	29.1
3	ASM	2023-09-17	American Samoa	8359.0	34.0	188712.044	767.581	44295	278.205	NaN
4	AND	2023-05-07	Andorra	48015.0	159.0	601367.684	1991.408	79843	163.755	NaN
...	...	...	...	...	...	...	...	...	...	...
226	VNM	2023-10-22	Vietnam	11624000.0	43206.0	118386.518	440.039	98186856	308.127	32.6
227	WLF	2023-06-04	Wallis and Futuna	3550.0	8.0	306140.048	689.893	11596	NaN	NaN
228	YEM	2022-11-06	Yemen	11945.0	2159.0	354.487	64.072	33696612	53.508	20.3
229	ZMB	2023-12-03	Zambia	349304.0	4069.0	17449.783	203.270	20017670	22.995	17.7
230	ZWE	2024-01-28	Zimbabwe	266265.0	5737.0	16314.719	351.520	16320539	42.729	19.6

231 rows × 17 columns

Next steps: Generate code with covidtotals | View recommended plots | New interactive sheet

```
covidtotals.set_index('iso_code', inplace=True)
nls97_dummy_800.set_index('personid', inplace=True)
```

```
print(covidtotals[['total_cases', 'total_deaths']].describe())
```

```
        total_cases  total_deaths
count  2.310000e+02  2.310000e+02
mean   3.351599e+06  3.021420e+04
std    1.148321e+07  1.047789e+05
min    4.000000e+00  0.000000e+00
25%    2.567150e+04  1.775000e+02
50%    1.914960e+05  1.937000e+03
75%    1.294286e+06  1.415000e+04
max    1.034368e+08  1.127152e+06
```

```
print("Quantiles:\n", covidtotals[['total_cases', 'total_deaths', 'total_cases_pm', 'total_deaths_pm']].quantile([0.25, 0.5, 0.75]))
```

```
Quantiles:
       total_cases  total_deaths  total_cases_pm  total_deaths_pm
0.25      25671.5         177.5       21821.863          141.177
0.50     191496.0        1937.0      133946.251          827.046
0.75    1294286.0       14150.0      345689.831         1997.513
```

```
print("Skewness:\n", covidtotals[['total_cases', 'total_deaths']].skew())
print("Kurtosis:\n", covidtotals[['total_cases', 'total_deaths']].kurt())
```

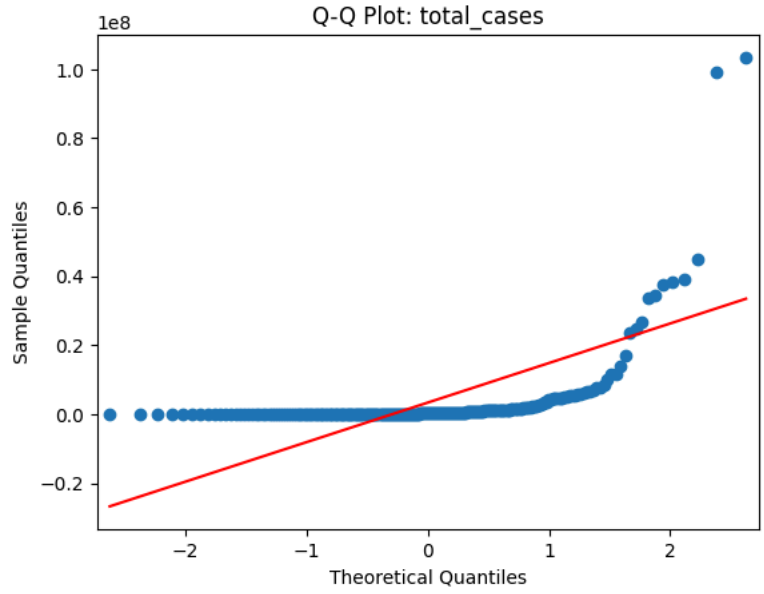
```
Skewness:
 total_cases     6.307179
total_deaths    7.098945
dtype: float64
Kurtosis:
```

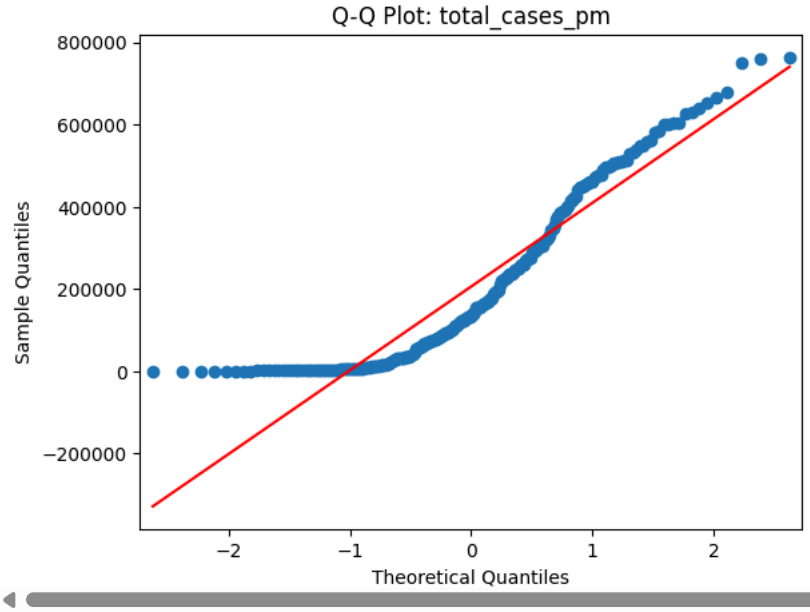
```
total_cases     47.080248
total_deaths    61.727944
dtype: float64
```

```
for col in ['total_cases', 'total_deaths']:
    stat, p = shapiro(covidtotals[col].dropna())
    print(f"{col}: Shapiro-Wilk stat={stat:.4f}, p-value={p:.4f}")
```

```
total_cases: Shapiro-Wilk stat=0.3031, p-value=0.0000
total_deaths: Shapiro-Wilk stat=0.2958, p-value=0.0000
```

```
for col in ['total_cases', 'total_cases_pm']:
    sm.qqplot(covidtotals[col].dropna(), line='s')
    plt.title(f'Q-Q Plot: {col}')
    plt.show()
```






```
def iqr_outliers(series):
    Q1 = series.quantile(0.25)
    Q3 = series.quantile(0.75)
    IQR = Q3 - Q1
    outliers = series[(series < Q1 - 1.5*IQR) | (series > Q3 + 1.5*IQR)]
    return outliers

outliers_cases = iqr_outliers(covidtotals['total_cases'])
print("Outliers in total_cases:\n", outliers_cases)
```

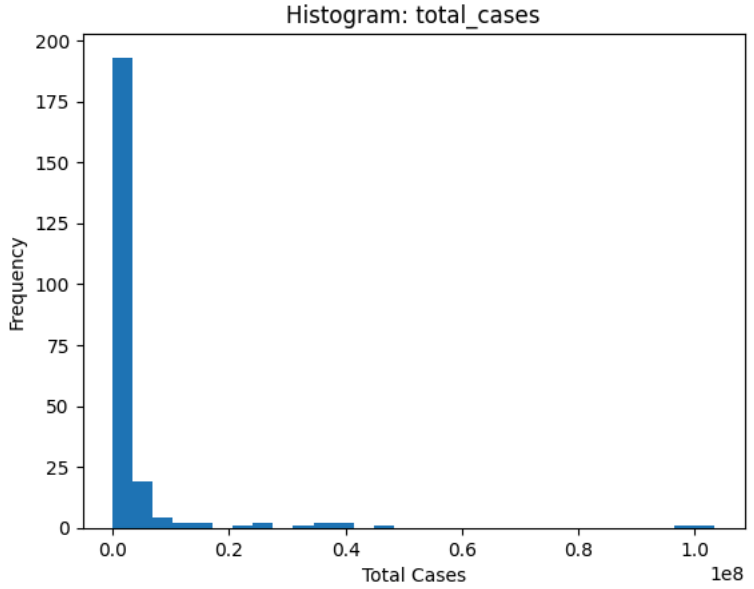
```
Outliers in total_cases:
 iso_code
ARG     10094643.0
AUS     11769858.0
AUT      6081287.0
```

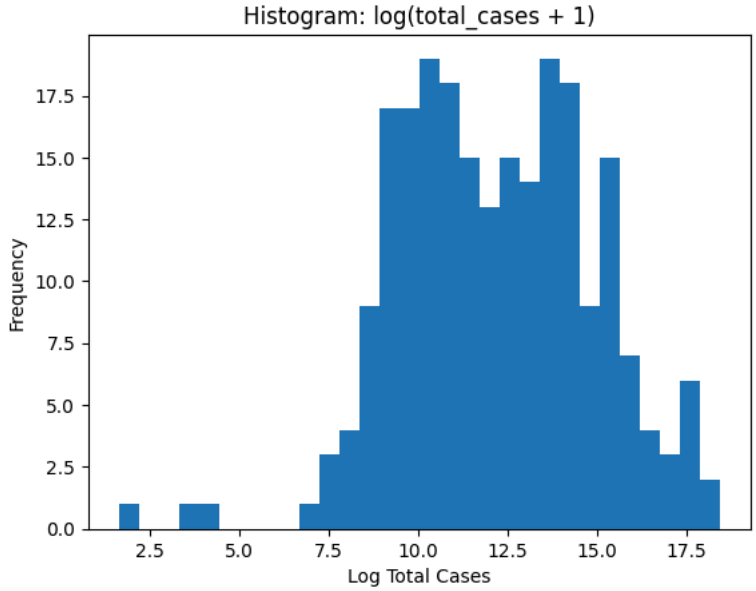
```
BEL      4855952.0
BRA     37519960.0
CAN      4774222.0
CHL      5345008.0
CHN     99329249.0
COL      6393550.0
CZE      4756085.0
DNK      3433033.0
FRA     38997490.0
DEU     38437756.0
GRC      5611832.0
IND     45026139.0
IDN      6828268.0
IRN      7626527.0
ISR      4841558.0
ITA     26699442.0
JPN     33803572.0
MYS      5269967.0
MEX      7702809.0
NLD      8633769.0
PER      4536733.0
PHL      4140383.0
POL      6653365.0
PRT      5641992.0
ROU      3519108.0
RUS     23774451.0
ZAF      4072636.0
KOR     34571873.0
ESP     13980340.0
CHE      4450977.0
THA      4765718.0
TUR     17004677.0
UKR      5521032.0
GBR     24892903.0
USA    103436829.0
VNM     11624000.0
Name: total_cases, dtype: float64
```

```
def detect_outliers_iqr(df, columns):
    outliers_dict = {}
    for col in columns:
        outliers_dict[col] = iqr_outliers(df[col])
    return outliers_dict

outliers = detect_outliers_iqr(covidtotals, ['total_cases', 'total_deaths', 'total_cases_pm', 'total_deaths_pm'])


outliers_deaths_pm = iqr_outliers(covidtotals['total_deaths_pm'])
print(covidtotals.loc[outliers_deaths_pm.index, ['total_deaths_pm', 'population', 'gdp_per_capita']])
```

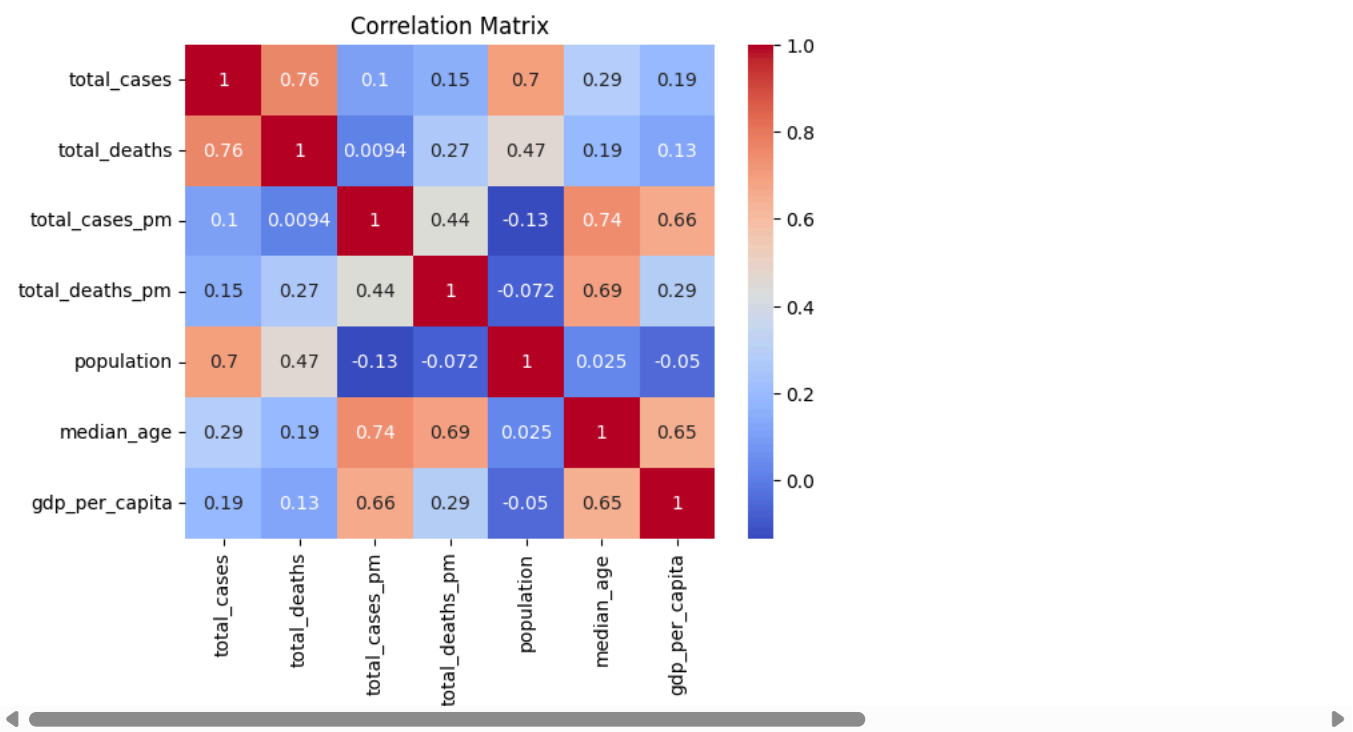
```
           total_deaths_pm  population  gdp_per_capita
iso_code
BIH              5066.290     3233530       11713.895
BGR              5703.518     6781955       18563.307
HUN              4918.281     9967304       26777.561
PER              6507.656    34049588       12236.706
```

```
plt.hist(covidtotals['total_cases'].dropna(), bins=30)
plt.title('Histogram: total_cases')
plt.xlabel('Total Cases')
plt.ylabel('Frequency')
plt.show()

plt.hist(np.log1p(covidtotals['total_cases'].dropna()), bins=30)
plt.title('Histogram: log(total_cases + 1)')
plt.xlabel('Log Total Cases')
plt.ylabel('Frequency')
plt.show()
```






```
cols = ['total_cases', 'total_deaths', 'total_cases_pm', 'total_deaths_pm', 'population', 'median_age', 'gdp_per_capita']
cor_matrix = covidtotals[cols].corr()
sns.heatmap(cor_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```



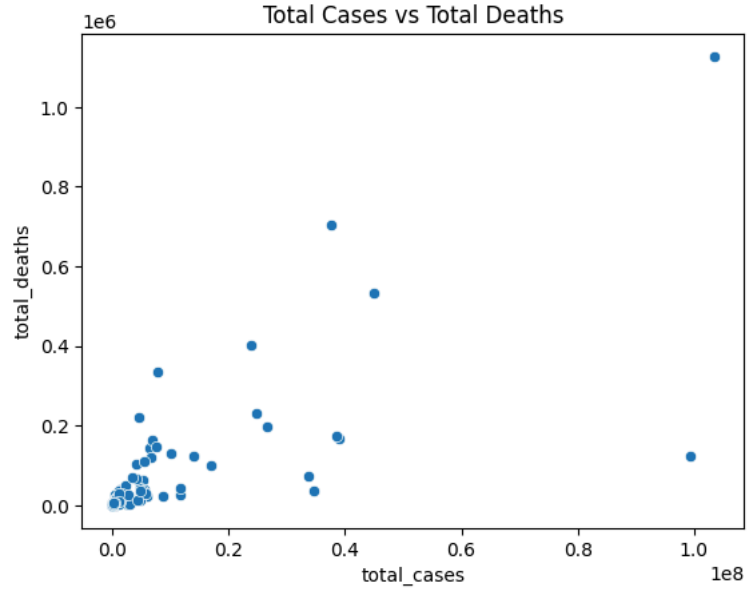

```
covidtotals['cases_q'] = pd.qcut(covidtotals['total_cases'], q=4, labels=False)
covidtotals['deaths_q'] = pd.qcut(covidtotals['total_deaths'], q=4, labels=False)

crosstab = pd.crosstab(covidtotals['cases_q'], covidtotals['deaths_q'])
print("Cases vs Deaths Quantile Crosstab:\n", crosstab)
```

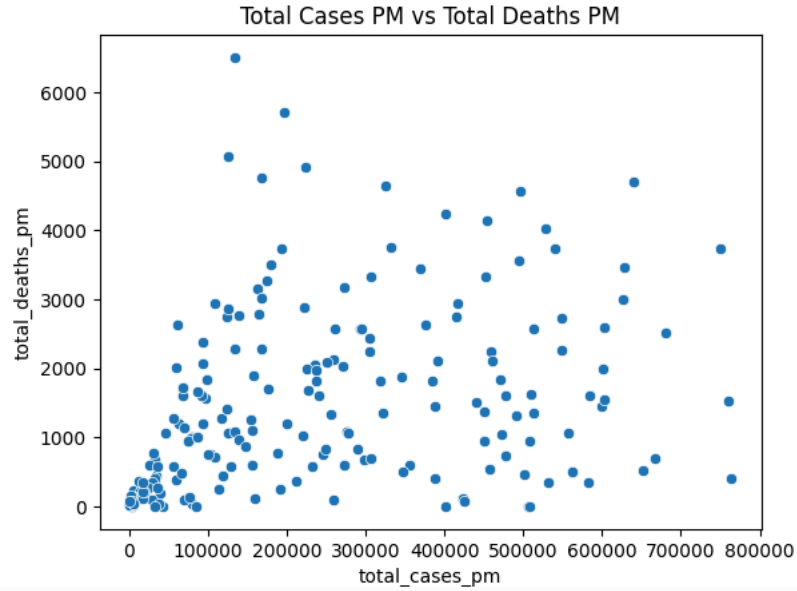
```
Cases vs Deaths Quantile Crosstab:
 deaths_q   0   1   2   3
cases_q
0         47  10   1   0
1         11  37  10   0
2          0  11  36  10
3          0   0  10  48
```

```
unexpected = covidtotals[(covidtotals['cases_q'] == 3) & (covidtotals['deaths_q'] <= 1)]
print("Countries with high cases and low deaths:\n", unexpected[['total_cases', 'total_deaths']])
```

```
Countries with high cases and low deaths:
 Empty DataFrame
Columns: [total_cases, total_deaths]
Index: []
```

```
sns.scatterplot(data=covidtotals, x='total_cases', y='total_deaths')
plt.title('Total Cases vs Total Deaths')
plt.show()

sns.scatterplot(data=covidtotals, x='total_cases_pm', y='total_deaths_pm')
plt.title('Total Cases PM vs Total Deaths PM')
plt.show()
```






```
filter1 = nls97_dummy_800[(nls97_dummy_800['wageincome'] > 0) & (nls97_dummy_800['weeksworked_2020'] == 0)]
print("Wage income but no work weeks:\n", filter1[['wageincome', 'weeksworked_2020']])
```

```
Wage income but no work weeks:
             wageincome  weeksworked_2020
personid
17         31748.27076                 0
98         69027.31060                 0
112        31995.99540                 0
116        13785.45235                 0
171        43365.09485                 0
434        32075.54430                 0
440        45696.51454                 0
482        37055.79023                 0
571        70229.12666                 0
593        14662.25443                 0
594        32931.04389                 0
596        51266.49182                 0
612        49499.90239                 0
629        28282.65433                 0
670        44458.66410                 0
678        41697.10686                 0
```

```
four_year_college = nls97_dummy_800[nls97_dummy_800['college_type'].str.contains('4-year', na=False)]
print("Individuals ever enrolled in a 4-year college:\n", four_year_college[['college_type']])
```

```
Individuals ever enrolled in a 4-year college:
           college_type
personid
4              4-year
5              4-year
8              4-year
10             4-year
14             4-year
```

```
...              ...
785           4-year
791           4-year
793           4-year
794           4-year
798           4-year

[329 rows x 1 columns]
```

```
grad_no_bachelor =nls97_dummy_800[(nls97_dummy_800['enrolled_grad'] == 1) & (nls97_dummy_800['enrolled_bachelors'] != 1)]
print("Graduate enrollment without bachelor's enrollment:\n", grad_no_bachelor[['enrolled_grad', 'enrolled_bachelors']])
```

```
Graduate enrollment without bachelor's enrollment:
            enrolled_grad  enrolled_bachelors
personid
2                       1                   0
4                       1                   0
14                      1                   0
27                      1                   0
38                      1                   0
...                   ...                 ...
767                     1                   0
771                     1                   0
772                     1                   0
777                     1                   0
785                     1                   0

[112 rows x 2 columns]
```

```
bachelor_or_higher = nls97_dummy_800[nls97_dummy_800['highestdegree'].isin(['Bachelor', 'Master', 'PhD'])]
no_four_year_enroll = bachelor_or_higher[~bachelor_or_higher['college_type'].str.contains('4-year', na=False)]
print("Bachelor+ degree holders with no 4-year college enrollment:\n", no_four_year_enroll[['highestdegree', 'college_type']])
```

```
Bachelor+ degree holders with no 4-year college enrollment:
           highestdegree college_type
personid
3               Master          NaN
7             Bachelor          NaN
11              Master        Other
12            Bachelor       2-year
22              Master          NaN
...                ...          ...
786                PhD       2-year
789           Bachelor       2-year
790           Bachelor       2-year
795           Bachelor       2-year
796             Master       2-year

[248 rows x 2 columns]
```

```
wage_mean = nls97_dummy_800['wageincome'].mean()
wage_std = nls97_dummy_800['wageincome'].std()
high_wage = nls97_dummy_800[nls97_dummy_800['wageincome'] > wage_mean + 3 * wage_std]
print("High wage earners (3σ above mean):\n", high_wage[['wageincome']])
```

```
High wage earners (3σ above mean):
             wageincome
personid
319       88646.39454
585       87280.85102
```

```
nls97_dummy_800['week_change'] = nls97_dummy_800['weeksworked_2021'] - nls97_dummy_800['weeksworked_2020']
significant_change = nls97_dummy_800[abs(nls97_dummy_800['week_change']) > 20]
print("Significant change in weeks worked (>|20|):\n", significant_change[['weeksworked_2020', 'weeksworked_2021', 'week_change']])
```

```
Significant change in weeks worked (>|20|):
            weeksworked_2020  weeksworked_2021  week_change
personid
4                         51                 4          -47
9                          4                31           27
13                         5                49           44
16                        32                 2          -30
17                         0                25           25
...                      ...               ...          ...
785                       44                20          -24
797                       51                17          -34
798                       36                10          -26
799                        7                47           40
800                       43                 1          -42

[305 rows x 3 columns]
```

```
crosstab_grade_degree = pd.crosstab(nls97_dummy_800['highestgradecompleted'], nls97_dummy_800['highestdegree'])
print("Crosstab: Highest Grade vs. Highest Degree:\n", crosstab_grade_degree)
```

```
Crosstab: Highest Grade vs. Highest Degree:
 highestdegree          Associate  Bachelor  High School  Master  PhD
highestgradecompleted
10th                          24        39           29      28   37
```