

```

import pandas as pd

import numpy as np

import re

from collections import Counter

# Load the dataset

df=pd.read_csv('/content/drive/MyDrive/synthetic_covid_data.csv')

from google.colab import drive
drive.mount('/content/drive')

↗ Mounted at /content/drive

print(df.columns)

↗ Index(['Country', 'Date', 'Confirmed', 'Deaths', 'Recovered', 'Active',
        'Tests', 'Population'],
        dtype='object')

```

#1. Problem: Find total confirmed cases across all countries.

```

total_confirmed = df['Confirmed'].sum()
print("Total Confirmed Cases:", total_confirmed)

```

↗ Total Confirmed Cases: 1705721772

#2. Problem: Find the country with the highest number of confirmed cases.

```

max_confirmed_country = df.loc[df['Confirmed'].idxmax(), 'Country']
print("Country with max confirmed cases:", max_confirmed_country)

```

↗ Country with max confirmed cases: India

#3. Problem: Find the average number of deaths.

```

avg_deaths = df['Deaths'].mean()
print("Average Deaths:", avg_deaths)

```

↗ Average Deaths: 9052.7

#4. Problem: Calculate death rate (Deaths / Confirmed) for each country.

```

df['Death Rate'] = (df['Deaths'] / df['Confirmed']) * 100
print(df[['Country', 'Death Rate']])

```

↗

	Country	Death Rate
0	USA	7.209302
1	USA	1.727116
2	USA	2.213879
3	USA	2.783883
4	USA	2.235257
...	...	...
3655	South Korea	1.932634
3656	South Korea	1.935434
3657	South Korea	1.931978
3658	South Korea	1.933057
3659	South Korea	1.931985

[3660 rows x 2 columns]

#5. Problem: Find the country with the lowest active cases.

```

min_active_country = df.loc[df['Active'].idxmin(), 'Country']
print("Country with min active cases:", min_active_country)

```

↗ Country with min active cases: Germany

#6. Problem: Find the total number of tests conducted.

```
total_tests = df['Tests'].sum()
print("Total Tests:", total_tests)
```

↗ Total Tests: 2791709430

#7. Problem: Find the country with the highest testing per capita.

```
df['Tests per 1000'] = (df['Tests'] / df['Population']) * 1000
top_test_country = df.loc[df['Tests per 1000'].idxmax(), 'Country']
print("Country with highest tests per 1000 people:", top_test_country)
```

↗ Country with highest tests per 1000 people: South Korea

#8. Problem: What is the overall recovery rate?

```
overall_recovery_rate = (df['Recovered'].sum() / df['Confirmed'].sum()) * 100
print("Overall Recovery Rate:", overall_recovery_rate)
```

↗ Overall Recovery Rate: 79.06414083105226

#9. Problem: Find the median number of confirmed cases.

```
median_confirmed = df['Confirmed'].median()
print("Median Confirmed Cases:", median_confirmed)
```

↗ Median Confirmed Cases: 466815.5

#10. Problem: Find standard deviation of deaths.

```
std_deaths = df['Deaths'].std()
print("Standard Deviation of Deaths:", std_deaths)
```

↗ Standard Deviation of Deaths: 5215.3765947469

#11. Problem: Find correlation between confirmed cases and deaths.

```
correlation = df['Confirmed'].corr(df['Deaths'])
print("Correlation between Confirmed and Deaths:", correlation)
```

↗ Correlation between Confirmed and Deaths: 0.9954081851108539

#12. Problem: Add a column for cases per million population.

```
df['Cases per Million'] = (df['Confirmed'] / df['Population']) * 1_000_000
print(df[['Country', 'Cases per Million']])
```

↗

	Country	Cases per Million
0	USA	3.014242
1	USA	16.234850
2	USA	27.072103
3	USA	28.705402
4	USA	44.218236
...	...	...
3655	South Korea	36339.217957
3656	South Korea	36405.123215
3657	South Korea	36568.482857
3658	South Korea	36674.872207
3659	South Korea	36820.045636

[3660 rows x 2 columns]

#13. Problem: List all countries with active cases more than 100,000.

```
high_active = df[df['Active'] > 100000]
print(high_active[['Country', 'Active']])
```

↗

	Country	Active
190	USA	101239
191	USA	100852
192	USA	102069

```

193      USA  100528
200      USA  100142
...      ...    ...
3655  South Korea  184654
3656  South Korea  184760
3657  South Korea  184884
3658  South Korea  186327
3659  South Korea  188132

```

[1609 rows x 2 columns]

#14. Problem: Find total deaths for countries with less than 500,000 confirmed cases.

```

deaths_small_countries = df[df['Confirmed'] < 500000]['Deaths'].sum()
print("Deaths in countries with <500k confirmed:", deaths_small_countries)

```

➡ Deaths in countries with <500k confirmed: 9574335

#15. Problem: Rank countries by recovery rate.

```

df['Recovery Rate'] = (df['Recovered'] / df['Confirmed']) * 100
ranked_countries = df[['Country', 'Recovery Rate']].sort_values(by='Recovery Rate', ascending=False)
print(ranked_countries)

```

➡

	Country	Recovery Rate
366	India	234.929577
0	USA	183.023256
367	India	182.630428
368	India	173.646018
2928	Spain	170.555556
...	...	...
739	Brazil	36.931863
733	Brazil	34.991843
732	Brazil	26.951431
734	Brazil	22.374164
2562	Italy	12.486163

[3660 rows x 2 columns]

#16. Problem: What is the maximum number of recovered cases?

```

max_recovered = df['Recovered'].max()
print("Max Recovered Cases:", max_recovered)

```

➡ Max Recovered Cases: 774673

#17. Problem: How many countries have recovery rate greater than 75%?

```

high_recovery = df[df['Recovery Rate'] > 75].shape[0]
print("Countries with recovery rate >75%:", high_recovery)

```

➡ Countries with recovery rate >75%: 2992

#18. Problem: Find the country with the least number of tests.

```

min_tests_country = df.loc[df['Tests'].idxmin(), 'Country']
print("Country with least tests:", min_tests_country)

```

➡ Country with least tests: USA

#19. Problem: Calculate active cases as a percentage of confirmed cases for each country.

```

df['Active %'] = (df['Active'] / df['Confirmed']) * 100
print(df[['Country', 'Active %']])

```

➡

	Country	Active %
0	USA	-90.232558
1	USA	16.774611
2	USA	45.688762
3	USA	39.865690
4	USA	57.070387
...	...	...
3655	South Korea	20.089429
3656	South Korea	20.064572

```
3657 South Korea 19.988345
3658 South Korea 20.085916
3659 South Korea 20.200532
```

```
[3660 rows x 2 columns]
```

```
#20. Problem: Find mean cases per million population.
```

```
mean_cases_per_million = df['Cases per Million'].mean()
print("Mean Cases per Million:", mean_cases_per_million)
```

```
↗ Mean Cases per Million: 4224.970072124432
```