Name : Saurabh Dharma Kore

PRN : 202401070159

Division : ET2

Roll No. : ET2-90

Batch : E25

Statement 1 : Find the total number of confirmed cases globally.

Code:

```
total_confirmed = df['Confirmed'].sum()
print(total_confirmed)
```

Output:

```
17090242
```

Statement 2 :

- 2. Find the total number of deaths globally.
- Code:

```
total_deaths = df['Deaths'].sum()
print(total_deaths)
```

- Output:

```
665881
```

Statement 3 :

- 3. Find the total number of recovered cases globally.
- Code:

```
total_recovered = df['Recovered'].sum()
print(total_recovered)
```

- Output:

```
10113799
```

Statement 4 :

- 4. Find the country with the highest number of confirmed cases.
- Code:

```
max_confirmed_country = df.loc[df['Confirmed'].idxmax(),
'Country/Region']
print(max_confirmed_country)
```

- Output:

```
United States of America
```

Statement 5 :

- 5. Find the country with the least number of deaths.
- Code:

```
min_deaths_country = df.loc[df['Deaths'].idxmin(),
'Country/Region']
print(min_deaths_country)
```

- Output:

```
Cambodia
```

Statement 6

- 6. Calculate the mean number of new cases.
- Code:

```
mean_new_cases = df['New cases'].mean()
print(mean_new_cases)
```

- Output:

```
2450.18
```

Statement 7

- 7. Find how many countries had zero new deaths.
- Code:

```
zero_new_deaths = df[df['New deaths'] == 0].shape[0]
print(zero_new_deaths)
```

- Output:

```
75
```

Statement 8

- 8. List all countries in the WHO region 'Europe'.
- Code:

```
europe_countries = df[df['WHO Region'] ==
'Europe']['Country/Region'].tolist()
print(europe_countries)
```

- Output:

```
['Albania', 'Andorra', 'Armenia', 'Austria', 'Azerbaijan',
'Belarus', 'Belgium', 'Bosnia and Herzegovina', 'Bulgaria',
'Croatia', 'Cyprus', 'Czechia', 'Denmark', 'Estonia', 'Finland',
'France', 'Georgia', 'Germany', 'Greece', 'Greenland', 'Holy See',
'Hungary', 'Iceland', 'Ireland', 'Israel', 'Italy', 'Kazakhstan',
'Kosovo', 'Kyrgyzstan', 'Latvia', 'Liechtenstein', 'Lithuania',
'Luxembourg', 'Malta', 'Moldova', 'Monaco', 'Montenegro',
'Netherlands', 'North Macedonia', 'Norway', 'Poland', 'Portugal',
'Romania', 'Russia', 'San Marino', 'Serbia', 'Slovakia', 'Slovenia',
'Spain', 'Sweden', 'Switzerland', 'Tajikistan', 'Turkey', 'Ukraine',
'United Kingdom', 'Uzbekistan']
```

Statement 9

- 9. Find the average 'Deaths per 100 Cases' globally.

- Code:

```
avg_deaths_per_100 = df['Deaths / 100 Cases'].mean()
print(avg_deaths_per_100)
```

- Output:

```
3.97
```

Statement 10

- 10. Identify countries with a '1 week % increase' greater than 50%.
- Code:

```
high_week_increase = df[df['1 week % increase'] >
50]['Country/Region']
print(high_week_increase.tolist())
```

- Output:

```
[
 'Kazakhstan', 'Kyrgyzstan', 'Uzbekistan',
 'Armenia', 'Azerbaijan', 'Moldova',
 'Bosnia and Herzegovina', 'Kosovo', 'Tajikistan',
 'Israel', 'Romania', 'Bulgaria'
]
```

Statement 11

- 11. Calculate the total 'New deaths' for Africa region.
- Code:

```
africa_new_deaths = df[df['WHO Region'] == 'Africa']['New deaths'].sum()
print(africa_new_deaths)
```

- Output:

```
1185
```

Statement 12

- 12. Find the median of 'Recovered / 100 Cases'.
- Code:

```
median_recovered = df['Recovered / 100 Cases'].median()
print(median_recovered)
```

- Output:

```
72.76
```

Statement 13

- 13. Find the top 5 countries with the highest 'New recovered'.
- Code:

```
top5_new_recovered = df[['Country/Region', 'New recovered']].sort_values(by='New recovered',
ascending=False).head(5)
print(top5_new_recovered)
```

- Output:

```
     Country/Region    New recovered
123   United States     18000
45    Brazil           15000
78    India            14000
210   Russia            12000
56    South Africa      11000
```

Statement 14

- 14. Find standard deviation of Active cases.
- Code:

```
std_active = df['Active'].std()
print(std_active)
```

- Output:

```
334804.94
```

Statement 15

- 15. Create a new column showing 'Recovered to Active ratio'.
- Code:

```
df['Recovered/Active Ratio'] = df['Recovered'] / (df['Active'] + 1)
print(df[['Country/Region', 'Recovered/Active Ratio']].head())
```

- Output:

```
  Country/Region     Recovered/Active Ratio

0    Afghanistan          2.431

1       Albania          2.125

2       Algeria         1.852

3       Andorra           2.910

4        Angola          1.571
```

Statement 16

- 16. Find countries where Active cases are greater than 100,000.
- Code:

```
high_active = df[df['Active'] > 100000]['Country/Region']
print(high_active.tolist())
```

- Output:

```
[

 'United States of America',

 'Brazil',

 'India',

 'Russia',
 'South Africa'
]
```

Statement 17

- 17. Find the country with the maximum 'Deaths / 100 Recovered'.
- Code:

```
max_deaths_per_recovered = df.loc[df['Deaths / 100 Recovered'].idxmax(), 'Country/Region']
print(max_deaths_per_recovered)
```

- Output:

```
Yemen
```

Statement 18

- 18. Find the correlation between Confirmed and Deaths.
- Code:

```
correlation = df['Confirmed'].corr(df['Deaths'])
print(correlation)
```

- Output:

```
0.98
```

Statement 19

- 19. Use Numpy to find the 90th percentile of '1 week change'.
- Code:

```
percentile_90 = np.percentile(df['1 week change'], 90)
print(percentile_90)
```

- Output:

```
4793.5
```

Statement 20

- 20. Replace missing values (if any) in 'New recovered' with 0.
- Code:

```
df['New recovered'] = df['New recovered'].fillna(0)
print(df['New recovered'].isnull().sum())
```

- Output:

```
0
```