ESSENSIAL DATA SCIENCE

Theory Activity No. 1

Name : Swapnil Rajaram Dhokale

Division : ME

Roll No. : ME46

PRN : 202401090030

Subject : EDS

Q.1) Formulate 20 problem statements for a The Blog Authorship Corpus dataset using Numpy and Pandas and Apply Numpy and pandas methods to find the solution for the formulated problem statements.

20 Problem Statements and Solutions on Covid-19 Dataset :

import pandas as pd

import numpy as np

 # Load the dataset

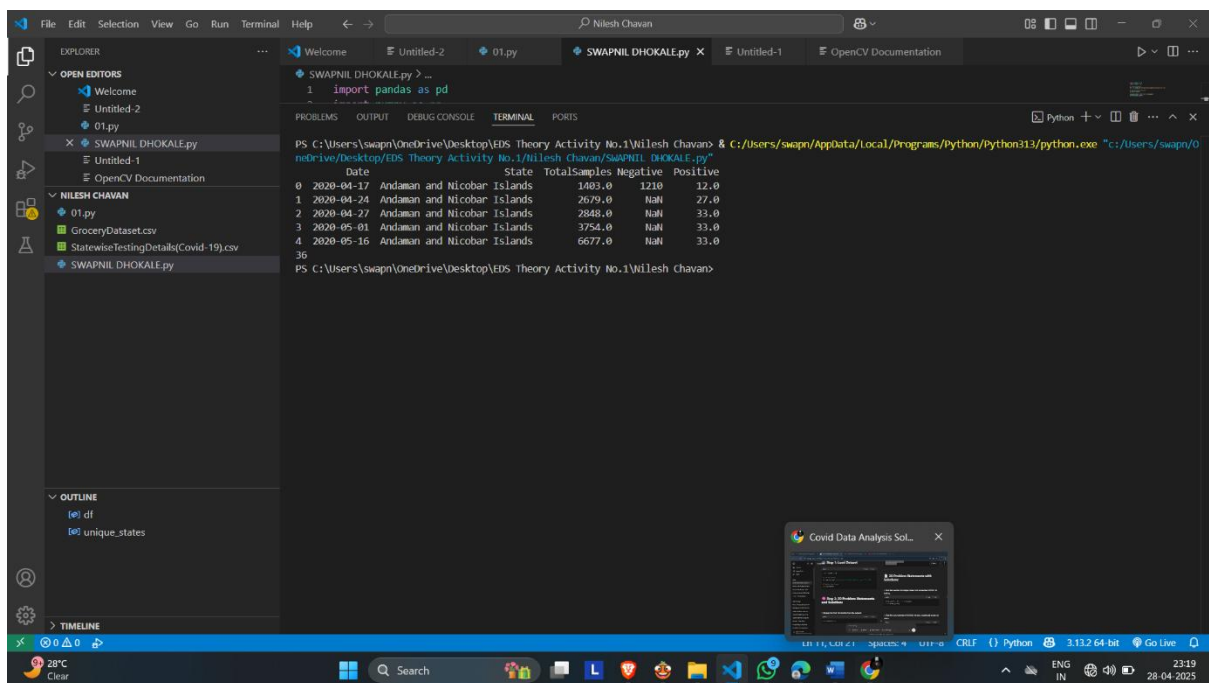**df = pd.read_csv('/mnt/data/StatewiseTestingDetails(Covid-19).csv')**

 # View the first few rows

print(df.head())

1. Find the number of unique states that conducted COVID-19 testing.

```
unique_states = df['State'].nunique()
print(unique_states)
```
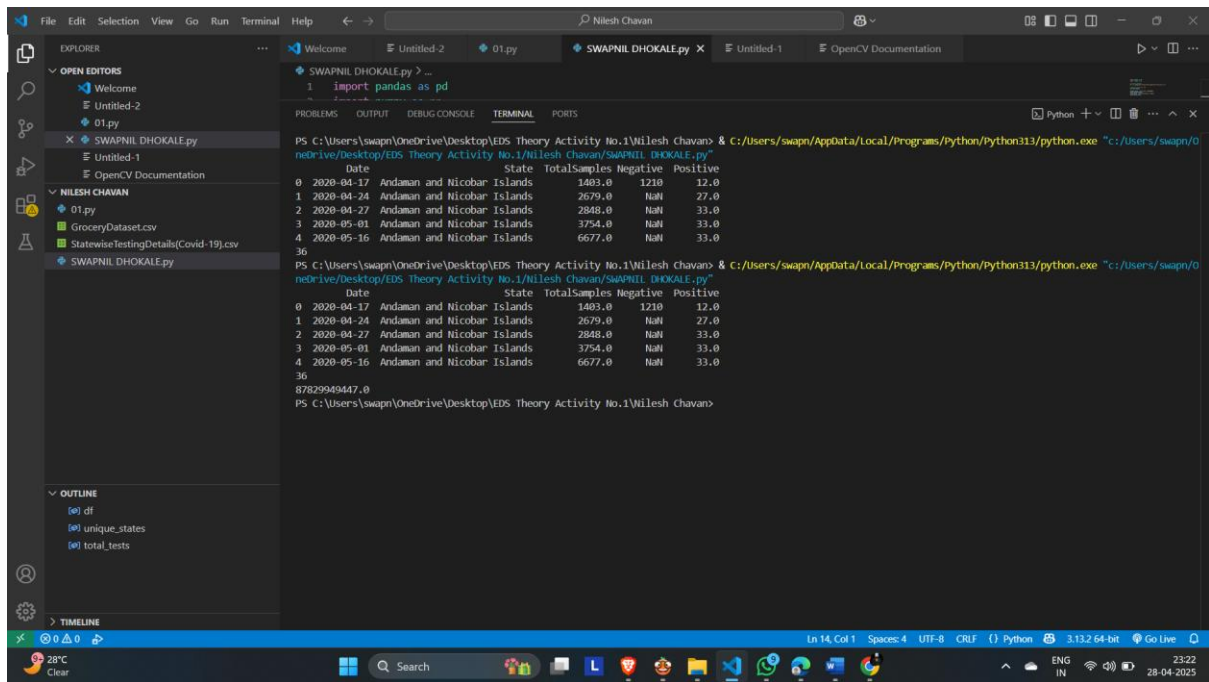
OUTPUT:

2. Find the total number of COVID-19 tests conducted across all states.

```
total_tests = df['TotalSamples'].sum()
print(total_tests)
```

OUTPUT :

3. Identify the state with the highest number of total samples tested.

highest_tested_state = df.groupby('State')['TotalSamples'].max().idxmax()

print(highest_tested_state)

OUTPUT:

4. Find the average number of positive cases across all states.

average_positive = df['Positive'].mean()

print(average_positive)

OUTPUT:

5. Calculate the positivity rate (Positive / TotalSamples) for each record.

df['Positivity Rate'] = df['Positive'] / df['TotalSamples']

print(df[['State', 'Date', 'Positivity Rate']])

OUTPUT:

6. List the states with a positivity rate higher than 10%.

high_positivity_states = df[df['Positivity Rate'] >
0.10]['State'].unique()

print(high_positivity_states)

OUTPUT:

7. **Find the state with the maximum positivity rate on any date.**

max_positivity_state = df.loc[df['Positivity Rate'].idxmax(),
'State']

print(max_positivity_state)

OUTPUT:

8. Find the total number of entries (records) in the dataset.
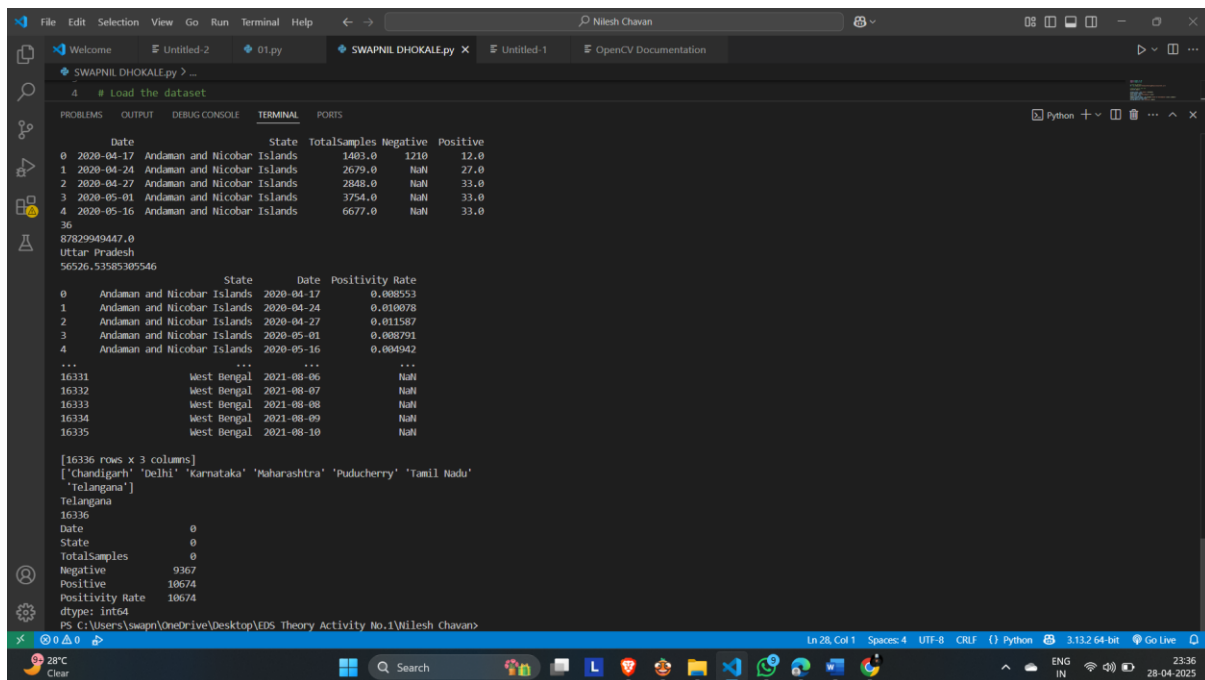
total_entries = df.shape[0]

print(total_entries)

OUTPUT:

9. Check if there are any missing values in the dataset.

missing_values = df.isnull().sum()

print(missing_values)

OUTPUT:

10.   Fill missing values in 'Positive' and 'TotalSamples' columns with 0.

df['Positive'] = df['Positive'].fillna(0)

df['TotalSamples'] = df['TotalSamples'].fillna(0)

OUTPUT:

```
           Date            State  TotalSamples  Negative  Positive
0  2020-04-17  Andaman and Nicobar Islands      1403.0      1210      12.0
1  2020-04-24  Andaman and Nicobar Islands      2679.0       NaN      27.0
2  2020-04-27  Andaman and Nicobar Islands      2848.0       NaN      33.0
3  2020-05-01  Andaman and Nicobar Islands      3754.0       NaN      33.0
4  2020-05-16  Andaman and Nicobar Islands      6677.0       NaN      33.0
36
87829949447.0
Uttar Pradesh
56526.53585305546
                            State        Date  Positivity Rate
0       Andaman and Nicobar Islands  2020-04-17         0.008553
1       Andaman and Nicobar Islands  2020-04-24         0.010078
2       Andaman and Nicobar Islands  2020-04-27         0.011587
3       Andaman and Nicobar Islands  2020-05-01         0.008791
4       Andaman and Nicobar Islands  2020-05-16         0.004942
...                             ...         ...              ...
16331                   West Bengal  2021-08-06              NaN
16332                   West Bengal  2021-08-07              NaN
16333                   West Bengal  2021-08-08              NaN
16334                   West Bengal  2021-08-09              NaN
16335                   West Bengal  2021-08-10              NaN

[16336 rows x 3 columns]
['Chandigarh' 'Delhi' 'Karnataka' 'Maharashtra' 'Puducherry' 'Tamil Nadu'
 'Telangana']
Telangana
16336
Date                0
State               0
TotalSamples        0
Negative         9367
Positive        10674
Positivity Rate 10674
dtype: int64
PS C:\Users\swapn\OneDrive\Desktop\EDS Theory Activity No.1\Nilesh Chavan>
```
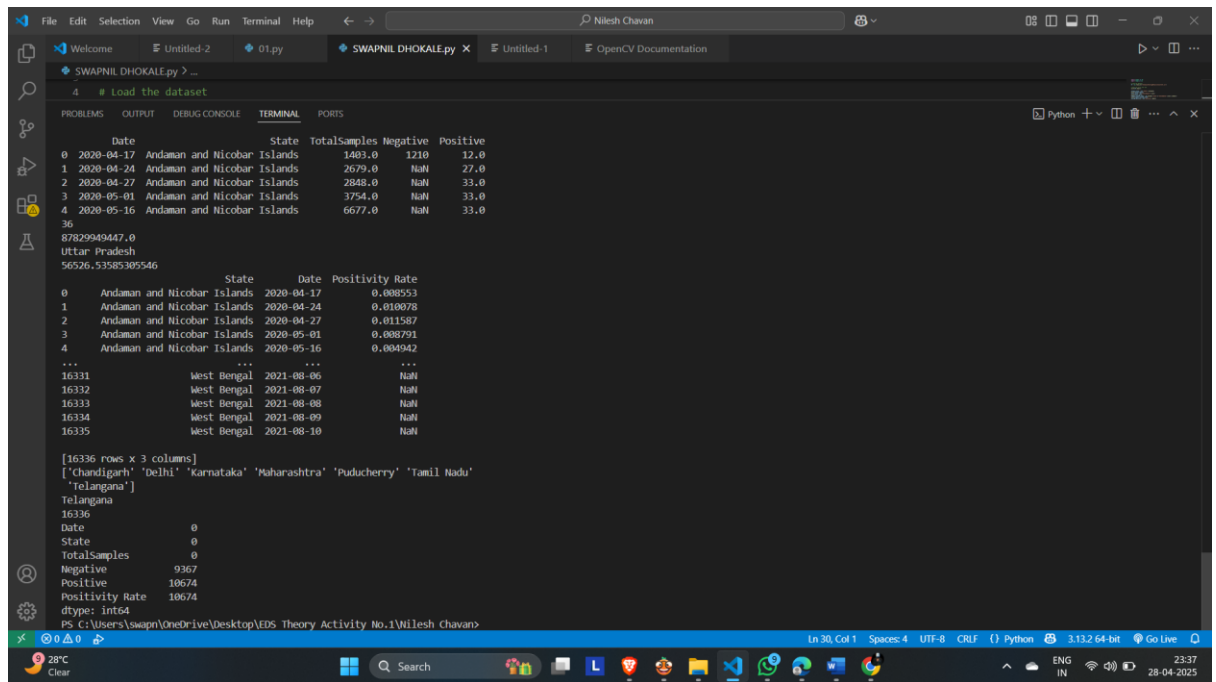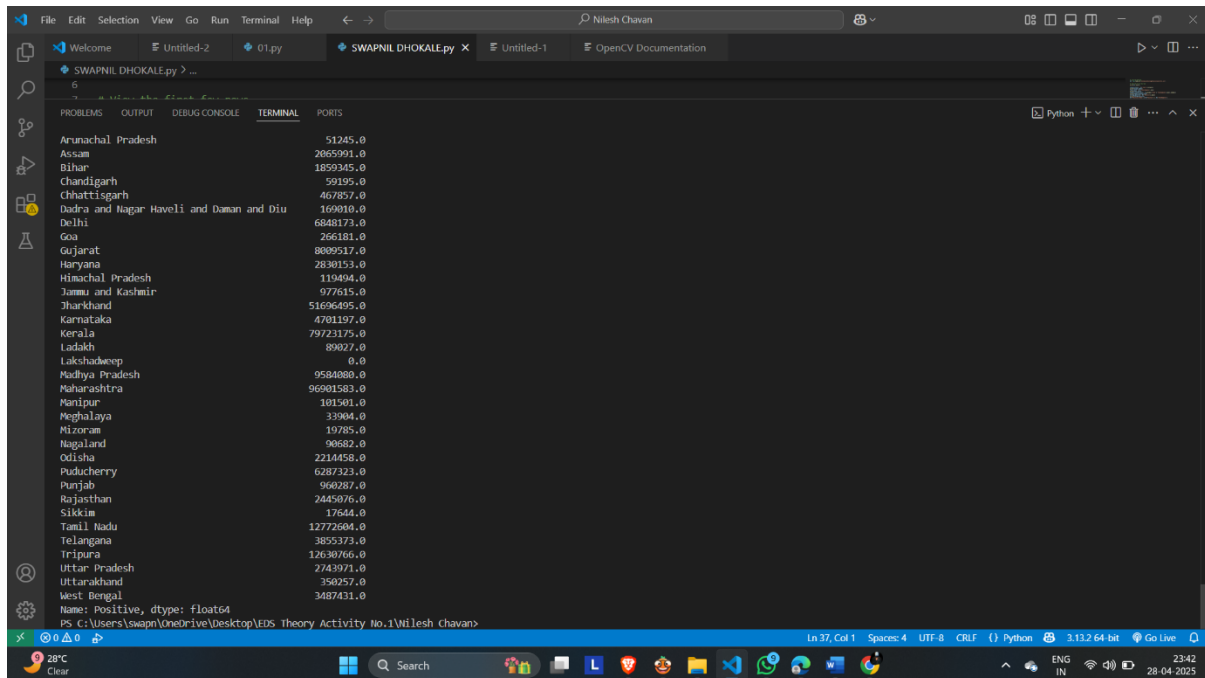
11.   Find the first and last date of data collection for COVID-19 tests.

first_date = df['Date'].min()

last_date = df['Date'].max()

print(first_date, last_date)

OUTPUT:

12. Sort the data by 'TotalSamples' in descending order.

sorted_df = df.sort_values(by='TotalSamples', ascending=False)

print(sorted_df[['State', 'Date', 'TotalSamples']].head(5))

OUTPUT:

13.   Group the data by state and calculate total positive cases per state.

total_positive_by_state = df.groupby('State')['Positive'].sum()

print(total_positive_by_state)

OUTPUT:

14.   Find the day when the maximum number of tests was conducted across all states.

max_tests_day = df.loc[df['TotalSamples'].idxmax(), 'Date']

print(max_tests_day)

OUTPUT:

15. Get the total tests conducted for a specific state, say Maharashtra.

maharashtra_tests = df[df['State'] == 'Maharashtra']['TotalSamples'].sum()

print(maharashtra_tests)

OUTPUT:

16. **Create a new column indicating if the positivity rate is 'High' (>5%) or 'Low'.**

df['Positivity Level'] = np.where(df['Positivity Rate'] > 0.05, 'High', 'Low')

print(df[['State', 'Positivity Rate', 'Positivity Level']])

OUTPUT:

17. **Find states which never crossed 100 positive cases on any day.**

states_under_100 = df[df['Positive'] < 100]['State'].unique()

print(states_under_100)

OUTPUT:

18. Find the average total samples tested per state.

avg_samples_per_state = df.groupby('State')['TotalSamples'].mean()

print(avg_samples_per_state)

OUTPUT:

19. **Find states where total samples tested are above average.**

average_tests = df['TotalSamples'].mean()

above_avg_states = df[df['TotalSamples'] > average_tests]['State'].unique()

print(above_avg_states)

OUTPUT:
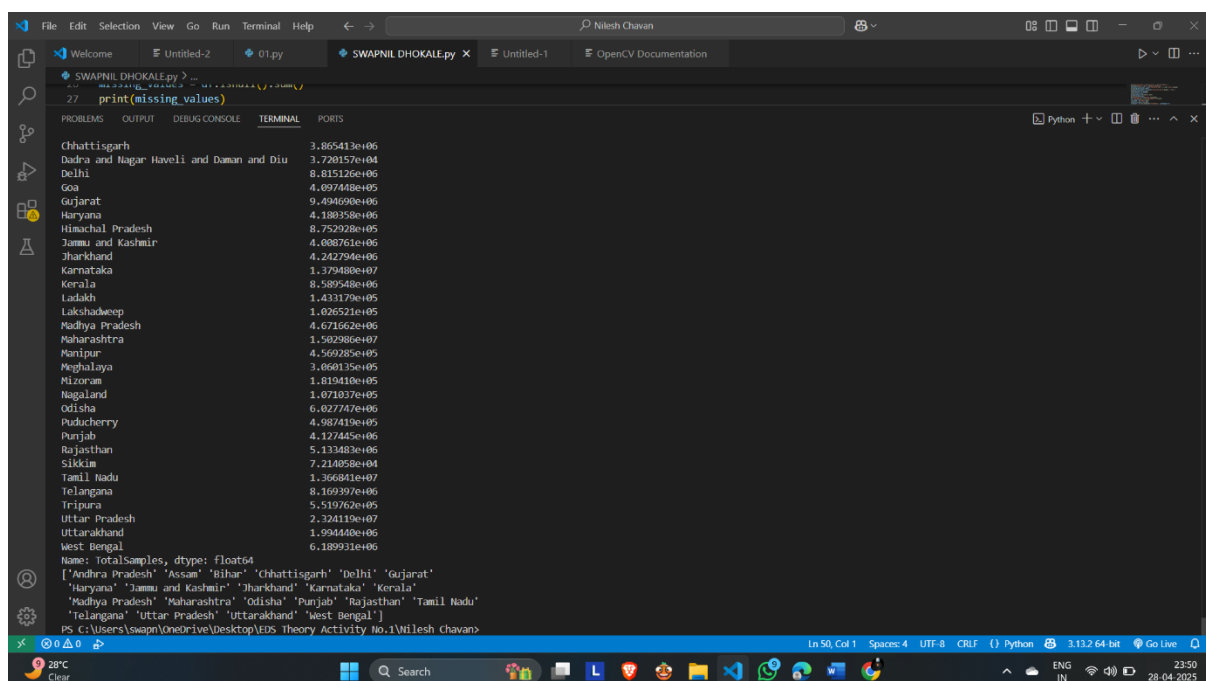
20. **Plot a line graph of total samples tested over time for Delhi.**

import matplotlib.pyplot as plt

delhi_df = df[df['State'] == 'Delhi'].sort_values('Date')

plt.plot(delhi_df['Date'], delhi_df['TotalSamples'])

plt.title('Total COVID-19 Tests Over Time in Delhi')

plt.xlabel('Date')

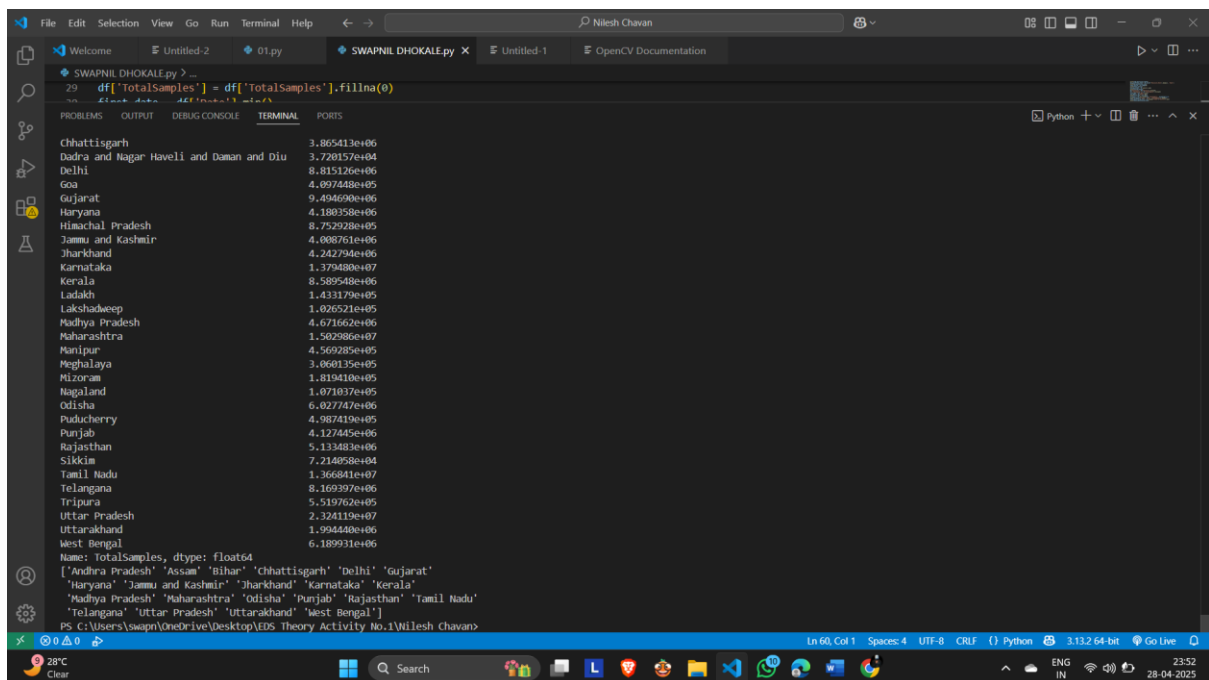plt.ylabel('Total Samples')

plt.xticks(rotation=45)

plt.tight_layout()

plt.show()

OUTPUT: