

PYTHON THEORY ACTIVITY 1

Details:

Name: Sakshi Ramesh Mogali

Div: CS5 Roll No: CS5-03

PRN: 202401100007

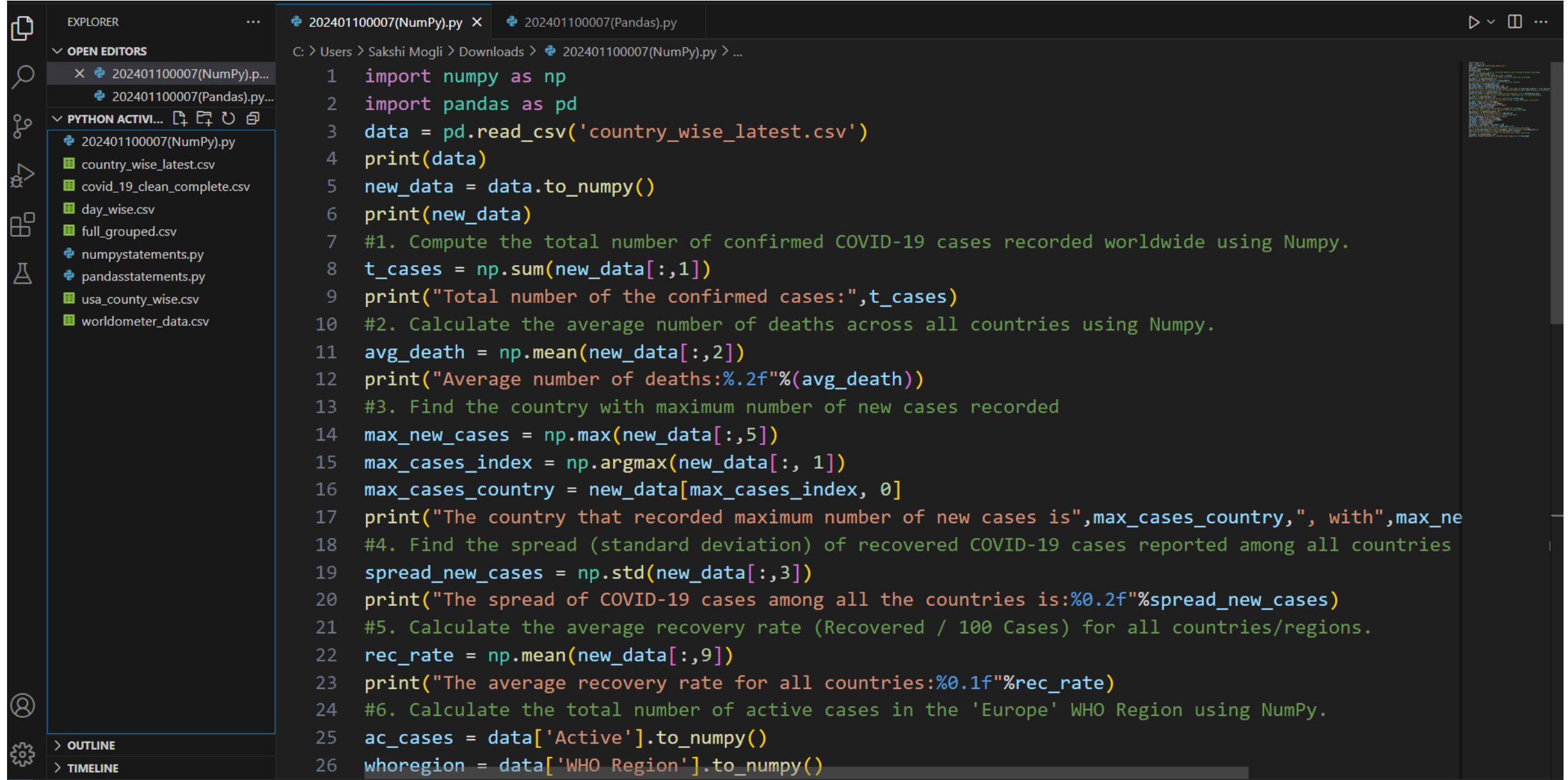
10 PROBLEM STATEMENTS ON NUMPY

1. Compute the total number of confirmed COVID-19 cases recorded worldwide using NumPy.
2. Calculate the average number of deaths across all countries using NumPy.
3. Find the country with maximum number of new cases recorded
4. Find the spread (standard deviation) of recovered COVID-19 cases reported among all countries using NumPy.
5. Calculate the average recovery rate ($\text{Recovered} / 100 \text{ Cases}$) for all countries/regions.
6. Calculate the total number of active cases in the 'Europe' WHO Region using NumPy.
7. Find the median number of new recovered COVID-19 cases using NumPy.
8. Calculate the COVID-19 death rate for India using NumPy
9. Calculate the correlation between new cases and new deaths using NumPy.
10. Find the std deviation in number of confirmed cases weekly using NumPy.

10 PROBLEM STATEMENTS ON PANDAS

1. Calculate the daily increase in the number of confirmed cases.
2. Find the day with the highest number of recoveries.
3. Find the day with the highest number of new cases
4. Calculate the death rate and recovery rate day-wise.
5. Find the day with maximum deaths reported.
6. Identify periods where active cases were consistently decreasing for at least 5 consecutive days.
7. What is the overall mortality rate at the end of the dataset?
8. Find the first day when more than 10,000 new cases were reported.
9. Find the average number of new cases reported daily.
10. Find the total number of confirmed cases, deaths, and recovered cases at the end.

CODE FOR NUMPY STATEMENTS

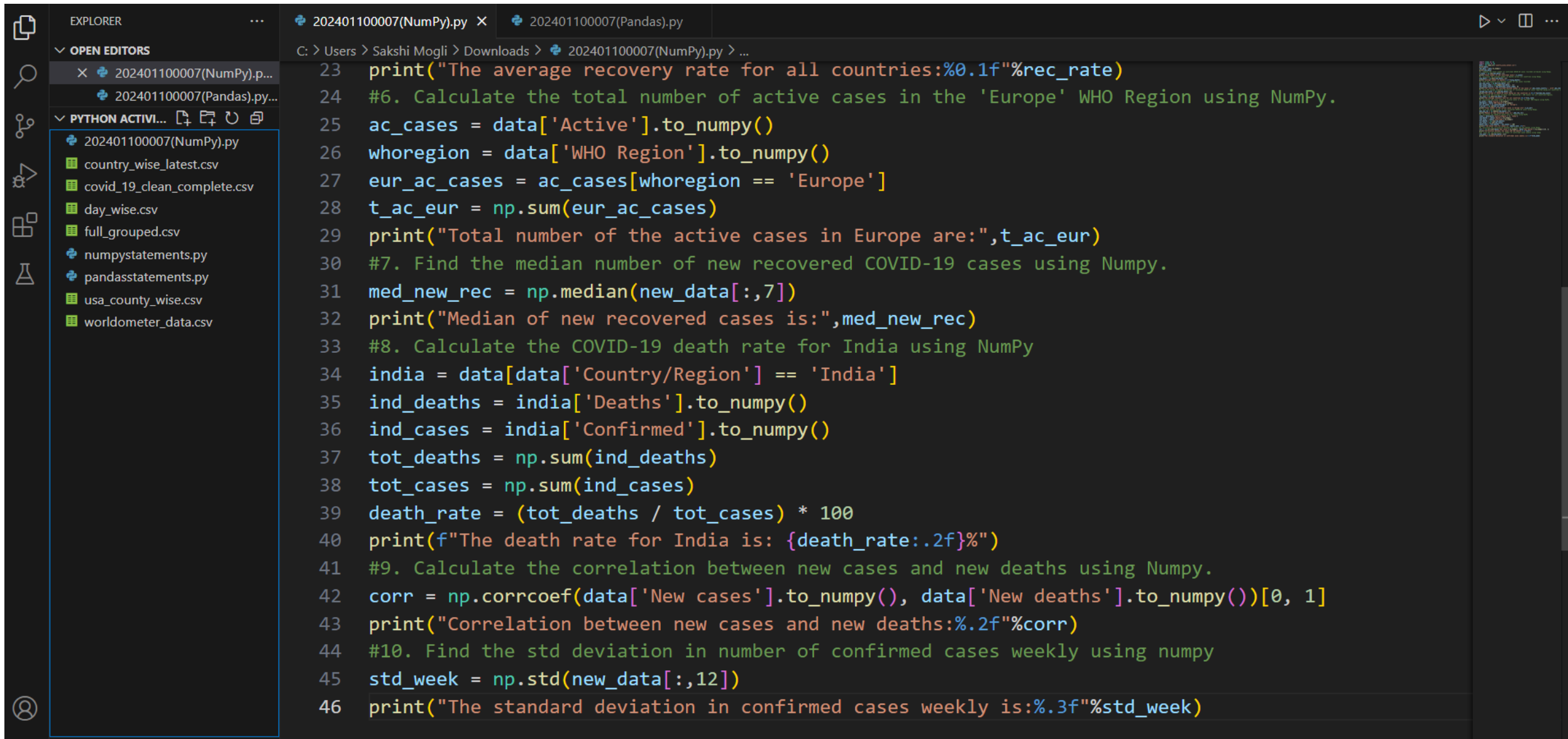


The image shows a Visual Studio Code editor window with two tabs open: `202401100007(NumPy).py` and `202401100007(Pandas).py`. The active tab is `202401100007(NumPy).py`, which contains the following Python code:

```
1 import numpy as np
2 import pandas as pd
3 data = pd.read_csv('country_wise_latest.csv')
4 print(data)
5 new_data = data.to_numpy()
6 print(new_data)
7 #1. Compute the total number of confirmed COVID-19 cases recorded worldwide using Numpy.
8 t_cases = np.sum(new_data[:,1])
9 print("Total number of the confirmed cases:",t_cases)
10 #2. Calculate the average number of deaths across all countries using Numpy.
11 avg_death = np.mean(new_data[:,2])
12 print("Average number of deaths:%.2f"%(avg_death))
13 #3. Find the country with maximum number of new cases recorded
14 max_new_cases = np.max(new_data[:,5])
15 max_cases_index = np.argmax(new_data[:, 1])
16 max_cases_country = new_data[max_cases_index, 0]
17 print("The country that recorded maximum number of new cases is",max_cases_country,", with",max_ne
18 #4. Find the spread (standard deviation) of recovered COVID-19 cases reported among all countries
19 spread_new_cases = np.std(new_data[:,3])
20 print("The spread of COVID-19 cases among all the countries is:%0.2f"%spread_new_cases)
21 #5. Calculate the average recovery rate (Recovered / 100 Cases) for all countries/regions.
22 rec_rate = np.mean(new_data[:,9])
23 print("The average recovery rate for all countries:%0.1f"%rec_rate)
24 #6. Calculate the total number of active cases in the 'Europe' WHO Region using NumPy.
25 ac_cases = data['Active'].to_numpy()
26 whoregion = data['WHO Region'].to_numpy()
```

The Explorer sidebar on the left shows the project structure with files like `country_wise_latest.csv`, `covid_19_clean_complete.csv`, `day_wise.csv`, `full_grouped.csv`, `numpystatements.py`, `pandasstatements.py`, `usa_county_wise.csv`, and `worldometer_data.csv`. The Python Activity bar is also visible, showing the active file `202401100007(NumPy).py`.

CODE FOR NUMPY STATEMENTS



The image shows a code editor interface with a dark theme. On the left, there is an 'EXPLORER' sidebar showing a file tree. The 'PYTHON ACTIVITY...' section is expanded, showing a list of files: '202401100007(NumPy).py', 'country_wise_latest.csv', 'covid_19_clean_complete.csv', 'day_wise.csv', 'full_grouped.csv', 'numpystatements.py', 'pandasstatements.py', 'usa_county_wise.csv', and 'worldometer_data.csv'. The main editor area displays the code for '202401100007(NumPy).py'. The code is a Python script that uses NumPy to analyze COVID-19 data. It includes comments for each step, such as 'Calculate the total number of active cases in the 'Europe' WHO Region using NumPy.' and 'Find the median number of new recovered COVID-19 cases using Numpy.' The code uses various NumPy functions like `np.sum()`, `np.median()`, and `np.std()` to perform calculations on the data. The code is as follows:

```
23 print("The average recovery rate for all countries:%0.1f"%rec_rate)
24 #6. Calculate the total number of active cases in the 'Europe' WHO Region using NumPy.
25 ac_cases = data['Active'].to_numpy()
26 whoregion = data['WHO Region'].to_numpy()
27 eur_ac_cases = ac_cases[whoregion == 'Europe']
28 t_ac_eur = np.sum(eur_ac_cases)
29 print("Total number of the active cases in Europe are:",t_ac_eur)
30 #7. Find the median number of new recovered COVID-19 cases using Numpy.
31 med_new_rec = np.median(new_data[:,7])
32 print("Median of new recovered cases is:",med_new_rec)
33 #8. Calculate the COVID-19 death rate for India using NumPy
34 india = data[data['Country/Region'] == 'India']
35 ind_deaths = india['Deaths'].to_numpy()
36 ind_cases = india['Confirmed'].to_numpy()
37 tot_deaths = np.sum(ind_deaths)
38 tot_cases = np.sum(ind_cases)
39 death_rate = (tot_deaths / tot_cases) * 100
40 print(f"The death rate for India is: {death_rate:.2f}%")
41 #9. Calculate the correlation between new cases and new deaths using Numpy.
42 corr = np.corrcoef(data['New cases'].to_numpy(), data['New deaths'].to_numpy())[0, 1]
43 print("Correlation between new cases and new deaths:%.2f"%corr)
44 #10. Find the std deviation in number of confirmed cases weekly using numpy
45 std_week = np.std(new_data[:,12])
46 print("The standard deviation in confirmed cases weekly is:%.3f"%std_week)
```

OUTPUT FOR NUMPY STATEMENTS

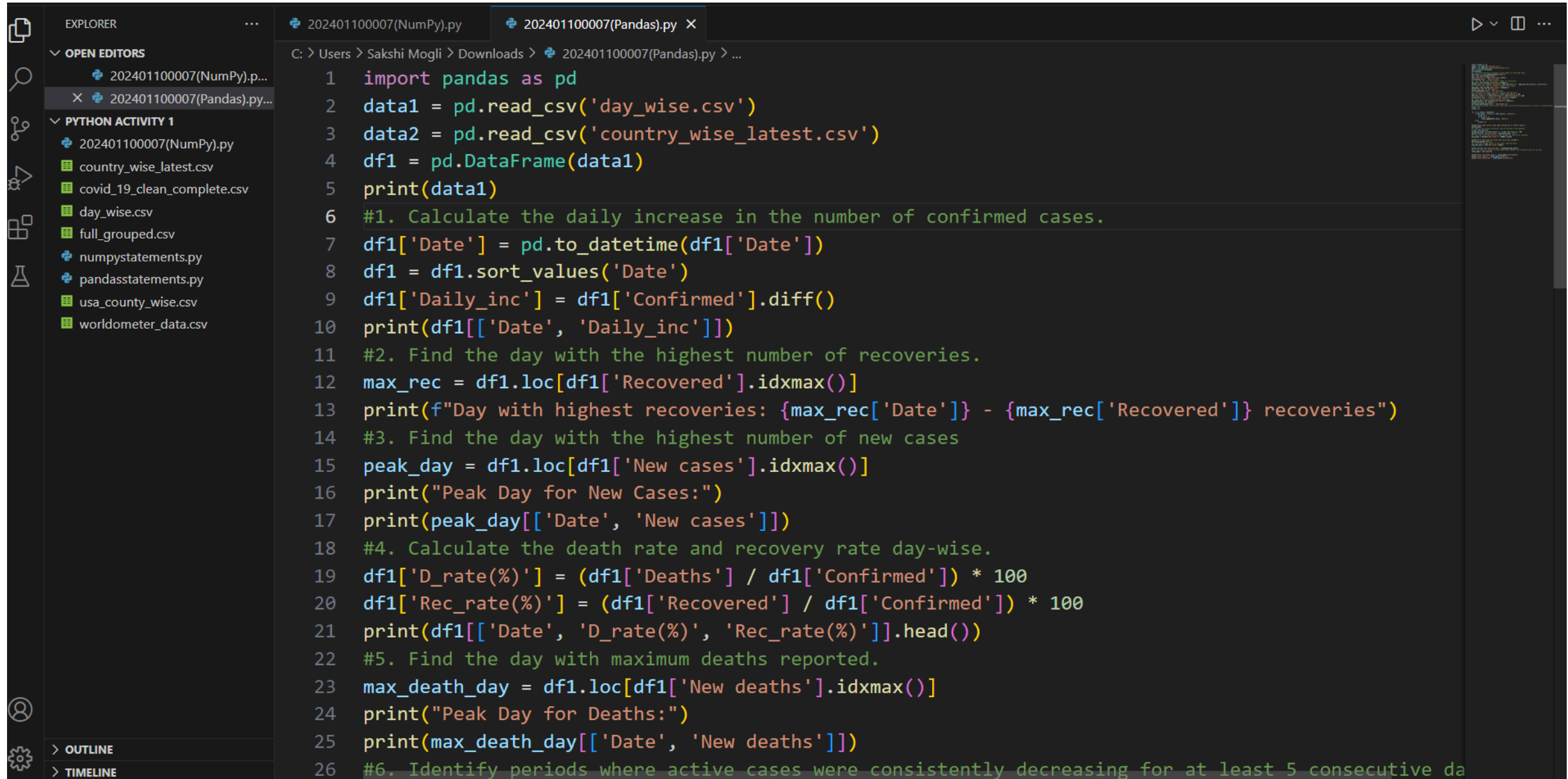
The screenshot shows a code editor with a file explorer on the left and a terminal window at the bottom. The file explorer lists several files, including `numpystatements.py` and `pandasstatements.py`. The terminal window displays the output of a Python script, which includes a table of COVID-19 statistics for various countries and regions, followed by summary statistics.

```
23 print("The average recovery rate for all countries:%0.1f"%rec_rate)
```

	Country/Region	Confirmed	Deaths	Recovered	Active	Deaths / 100 Recovered	Confirmed last week	1 week change	1 week % increase	WHO Region
0	Afghanistan	36263	1269	25198	9796	5.04	35526	737	2.07	Eastern Mediterranean
1	Albania	4880	144	2745	1991	5.25	4171	709	17.00	Europe
2	Algeria	27973	1163	18837	7973	6.17	23691	4282	18.07	Africa
3	Andorra	907	52	803	52	6.48	884	23	2.60	Europe
4	Angola	950	41	242	667	16.94	749	201	26.84	Africa
..
182	West Bank and Gaza	10621	78	3752	6791	2.08	8916	1705	19.12	Eastern Mediterranean
183	Western Sahara	10	1	8	1	12.50	10	0	0.00	Africa
184	Yemen	1691	483	833	375	57.98	1619	72	4.45	Eastern Mediterranean
185	Zambia	4552	140	2815	1597	4.97	3326	1226	36.86	Africa
186	Zimbabwe	2704	36	542	2126	6.64	1713	991	57.85	Africa

[187 rows x 15 columns]
[['Afghanistan' 36263 1269 ... 737 2.07 'Eastern Mediterranean']
['Albania' 4880 144 ... 709 17.0 'Europe']
['Algeria' 27973 1163 ... 4282 18.07 'Africa']
...
['Yemen' 1691 483 ... 72 4.45 'Eastern Mediterranean']
['Zambia' 4552 140 ... 1226 36.86 'Africa']
['Zimbabwe' 2704 36 ... 991 57.85 'Africa']]
Total number of the confirmed cases: 16480485
Average number of deaths:3497.52
The country that recorded maximum number of new cases is US , with 56336 new cases
The spread of COVID-19 cases among all the countries is:189678.98
The average recovery rate for all countries:64.8
Total number of the active cases in Europe are: 1094656
Median of new recovered cases is: 22.0
The death rate for India is: 2.26%
Correlation between new cases and new deaths:0.94
The standard deviation in confirmed cases weekly is:47363.976
PS D:\codeing\python\Python activity 1>

CODE FOR PANDAS STATEMENTS



The image shows a code editor interface with a dark theme. On the left, there is a sidebar with an 'EXPLORER' pane showing a file tree. The tree includes a folder 'PYTHON ACTIVITY 1' containing several files: '202401100007(NumPy).py', 'country_wise_latest.csv', 'covid_19_clean_complete.csv', 'day_wise.csv', 'full_grouped.csv', 'numpystatements.py', 'pandasstatements.py', 'usa_county_wise.csv', and 'worldometer_data.csv'. The main editor area displays a Python script named '202401100007(Pandas).py'. The script performs several data analysis tasks using Pandas, including reading CSV files, calculating daily increases, finding the day with the highest recoveries, finding the peak day for new cases, calculating death and recovery rates, finding the peak day for deaths, and identifying periods of decreasing active cases.

```
1 import pandas as pd
2 data1 = pd.read_csv('day_wise.csv')
3 data2 = pd.read_csv('country_wise_latest.csv')
4 df1 = pd.DataFrame(data1)
5 print(data1)
6 #1. Calculate the daily increase in the number of confirmed cases.
7 df1['Date'] = pd.to_datetime(df1['Date'])
8 df1 = df1.sort_values('Date')
9 df1['Daily_inc'] = df1['Confirmed'].diff()
10 print(df1[['Date', 'Daily_inc']])
11 #2. Find the day with the highest number of recoveries.
12 max_rec = df1.loc[df1['Recovered'].idxmax()]
13 print(f"Day with highest recoveries: {max_rec['Date']} - {max_rec['Recovered']} recoveries")
14 #3. Find the day with the highest number of new cases
15 peak_day = df1.loc[df1['New cases'].idxmax()]
16 print("Peak Day for New Cases:")
17 print(peak_day[['Date', 'New cases']])
18 #4. Calculate the death rate and recovery rate day-wise.
19 df1['D_rate(%)'] = (df1['Deaths'] / df1['Confirmed']) * 100
20 df1['Rec_rate(%)'] = (df1['Recovered'] / df1['Confirmed']) * 100
21 print(df1[['Date', 'D_rate(%)', 'Rec_rate(%)']].head())
22 #5. Find the day with maximum deaths reported.
23 max_death_day = df1.loc[df1['New deaths'].idxmax()]
24 print("Peak Day for Deaths:")
25 print(max_death_day[['Date', 'New deaths']])
26 #6. Identify periods where active cases were consistently decreasing for at least 5 consecutive da
```

CODE FOR PANDAS STATEMENTS

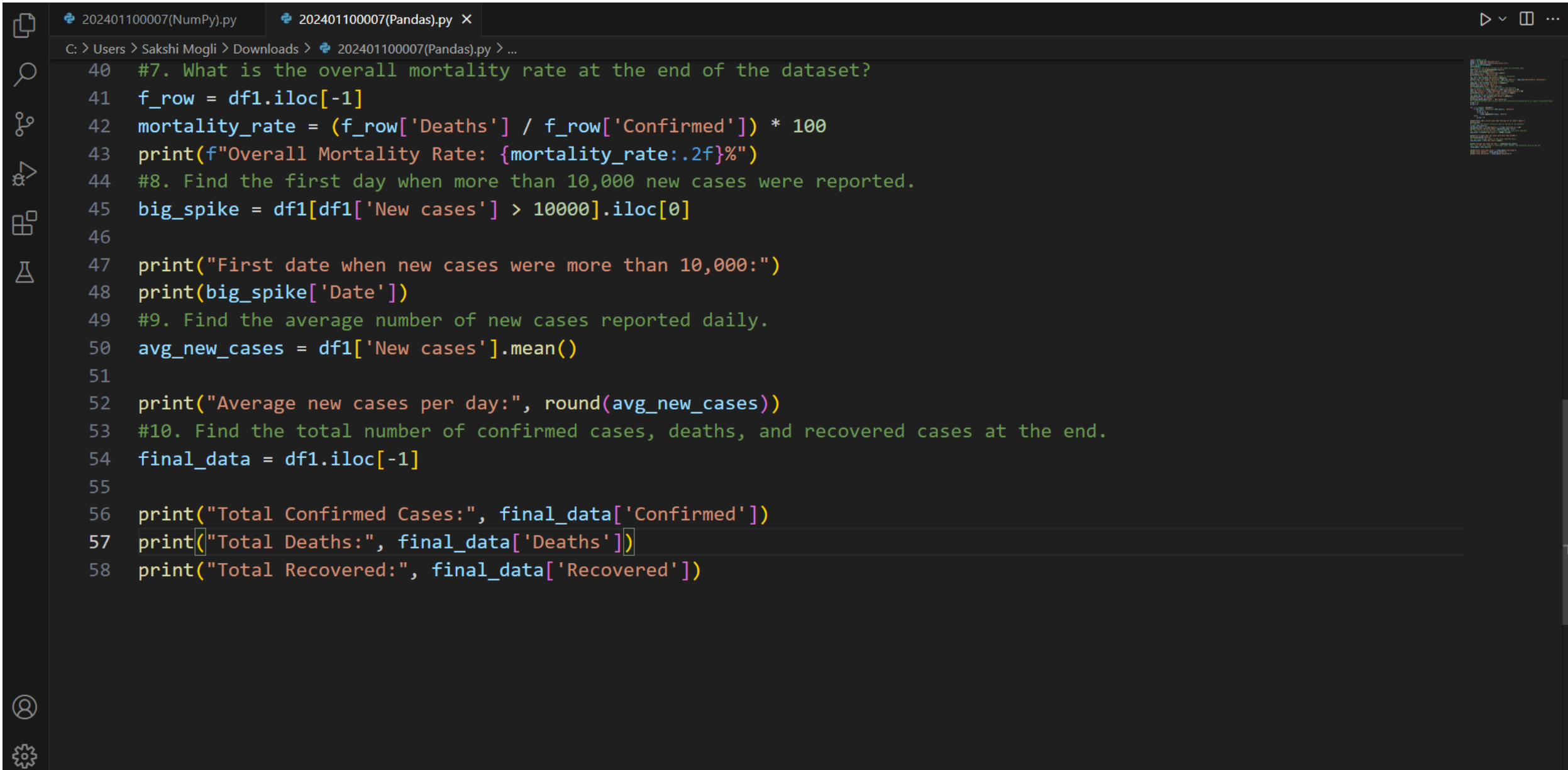
202401100007(NumPy).py

202401100007(Pandas).py X

C: > Users > Sakshi Mogli > Downloads > 202401100007(Pandas).py > ...

```
23 max_death_day = df1.loc[df1['New deaths'].idxmax()]
24 print("Peak Day for Deaths:")
25 print(max_death_day[['Date', 'New deaths']])
26 #6. Identify periods where active cases were consistently decreasing for at least 5 consecutive days.
27 d_dec = []
28 d_row = 0
29
30 for i in range(1, len(df1)):
31     if df1.loc[i, 'Active'] < df1.loc[i-1, 'Active']:
32         d_row += 1
33         if d_row >= 5:
34             d_dec.append(df1.loc[i, 'Date'])
35     else:
36         d_row = 0
37
38 print("Dates when active cases kept falling for at least 5 days:")
39 print(d_dec)
40 #7. What is the overall mortality rate at the end of the dataset?
41 f_row = df1.iloc[-1]
42 mortality_rate = (f_row['Deaths'] / f_row['Confirmed']) * 100
43 print(f"Overall Mortality Rate: {mortality_rate:.2f}%")
44 #8. Find the first day when more than 10,000 new cases were reported.
45 big_spike = df1[df1['New cases'] > 10000].iloc[0]
46
47 print("First date when new cases were more than 10,000:")
48 print(big_spike['Date'])
```


CODE FOR PANDAS STATEMENTS



The image shows a code editor with two tabs: '202401100007(NumPy).py' and '202401100007(Pandas).py'. The active tab is '202401100007(Pandas).py'. The code is written in Python and uses Pandas for data analysis. It includes comments for each step, variable assignments, and print statements to display results. The code is as follows:

```
40 #7. What is the overall mortality rate at the end of the dataset?
41 f_row = df1.iloc[-1]
42 mortality_rate = (f_row['Deaths'] / f_row['Confirmed']) * 100
43 print(f"Overall Mortality Rate: {mortality_rate:.2f}%")
44 #8. Find the first day when more than 10,000 new cases were reported.
45 big_spike = df1[df1['New cases'] > 10000].iloc[0]
46
47 print("First date when new cases were more than 10,000:")
48 print(big_spike['Date'])
49 #9. Find the average number of new cases reported daily.
50 avg_new_cases = df1['New cases'].mean()
51
52 print("Average new cases per day:", round(avg_new_cases))
53 #10. Find the total number of confirmed cases, deaths, and recovered cases at the end.
54 final_data = df1.iloc[-1]
55
56 print("Total Confirmed Cases:", final_data['Confirmed'])
57 print("Total Deaths:", final_data['Deaths'])
58 print("Total Recovered:", final_data['Recovered'])
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\codeing\python\Python activity 1> python -u "c:\Users\Sakshi Mogli\Downloads\202401100007(Pandas).py"

    Date Confirmed Deaths Recovered Active New cases New deaths New recovered Deaths / 100 Cases Recovered / 100 Cases Deaths / 100 Recovered No. of countries
0 2020-01-22 555 17 28 510 0 0 0 3.06 5.05 60.71 6
1 2020-01-23 654 18 30 606 99 1 2 2.75 4.59 60.00 8
2 2020-01-24 941 26 36 879 287 8 6 2.76 3.83 72.22 9
3 2020-01-25 1434 42 39 1353 493 16 3 2.93 2.72 107.69 11
4 2020-01-26 2118 56 52 2010 684 14 13 2.64 2.46 107.69 13
.. ... ..
183 2020-07-23 15510481 633506 8710969 6166006 282756 9966 169714 4.08 56.16 7.27 187
184 2020-07-24 15791645 639650 8939705 6212290 281164 6144 228736 4.05 56.61 7.16 187
185 2020-07-25 16047190 644517 9158743 6243930 255545 4867 219038 4.02 57.07 7.04 187
186 2020-07-26 16251796 648621 9293464 6309711 204606 4104 134721 3.99 57.18 6.98 187
187 2020-07-27 16480485 654036 9468087 6358362 228693 5415 174623 3.97 57.45 6.91 187

[188 rows x 12 columns]
    Date Daily_inc
0 2020-01-22 NaN
1 2020-01-23 99.0
2 2020-01-24 287.0
3 2020-01-25 493.0
4 2020-01-26 684.0
.. ...
183 2020-07-23 282756.0
184 2020-07-24 281164.0
185 2020-07-25 255545.0
186 2020-07-26 204606.0
187 2020-07-27 228689.0

[188 rows x 2 columns]
Day with highest recoveries: 2020-07-27 00:00:00 - 9468087 recoveries
Peak Day for New Cases:
Date 2020-07-23 00:00:00
New cases 282756
Name: 183, dtype: object
    Date D_rate(%) Rec_rate(%)
0 2020-01-22 3.063063 5.045045
1 2020-01-23 2.752294 4.587156
2 2020-01-24 2.763018 3.825717
3 2020-01-25 2.928870 2.719665
4 2020-01-26 2.644004 2.455146
Peak Day for Deaths:
```

OUTPUT FOR PANDAS STATEMENTS

```
4 2020-01-26    2.644004    2.455146
Peak Day for Deaths:
Date          2020-07-23 00:00:00
New deaths          9966
Name: 183, dtype: object
Dates when active cases kept falling for at least 5 days:
[Timestamp('2020-02-26 00:00:00'), Timestamp('2020-02-27 00:00:00'), Timestamp('2020-02-28 00:00:00'), Timestamp('2020-02-29 00:00:00'), Timestamp('2020-03-01 00:00:00'), Timestamp('2020-03-02 00:00:00'), Timestamp('2020-03-03 00:00:00'), Timestamp('2020-03-04 00:00:00')]
Overall Mortality Rate: 3.97%
First date when new cases were more than 10,000:
2020-02-13 00:00:00
Average new cases per day: 87771
Total Confirmed Cases: 16480485
Total Deaths: 654036
Total Recovered: 9468087
PS D:\codeing\python\Python activity 1> █
```

GOOGLE DRIVE LINK FOR ACCESSING CODES, OUTPUTS AND DATASETS IMPORTED

<https://drive.google.com/drive/folders/1H0tGh5qO5J1ecVIIfNf-B5xODCuqPSKx?usp=sharing>