Introduction to Matplotlib library in Python

Lab report #08

Fall 2022

CSE-408L Data Analytics lab

Submitted by: **Ashfaq Ahmad**

Registration No: **19PWCSE1795**

Class Section: **B**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: _____

Submitted to:

Eng: Faizullah

Dec 22, 2022

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

lab08

December 22, 2022

1 Lab Tasks

[77]:
```
import matplotlib.pyplot as plt
```

[78]:
```
import pandas as pd
```

[79]:
```
import numpy as np
```

[80]:
```
read_df=pd.read_csv('italy-covid-daywise.csv')
```

[81]:
```
read_df
```

[81]:

	date	new_cases	new_deaths	new_tests
0	2019-12-31	0	0	NaN
1	2020-01-01	0	0	NaN
2	2020-01-02	0	0	NaN
3	2020-01-03	0	0	NaN
4	2020-01-04	0	0	NaN
..	…	…	…	…
243	2020-08-30	1444	1	53541.0
244	2020-08-31	1365	4	42583.0
245	2020-09-01	996	6	54395.0
246	2020-09-02	975	8	NaN
247	2020-09-03	1326	6	NaN

[248 rows x 4 columns]

1.1 Task01

Display the graph of death cases verses months.

[82]:
```
read_df['month'] = pd.to_datetime(read_df.date).dt.strftime('%m/%y')
read_df
```

[82]:

	date	new_cases	new_deaths	new_tests	month
0	2019-12-31	0	0	NaN	12/19
1	2020-01-01	0	0	NaN	01/20
2	2020-01-02	0	0	NaN	01/20

```
3    2020-01-03          0           0        NaN  01/20
4    2020-01-04          0           0        NaN  01/20
..          …        …          …        …    …
243  2020-08-30       1444           1    53541.0  08/20
244  2020-08-31       1365           4    42583.0  08/20
245  2020-09-01        996           6    54395.0  09/20
246  2020-09-02        975           8        NaN  09/20
247  2020-09-03       1326           6        NaN  09/20

[248 rows x 5 columns]
```

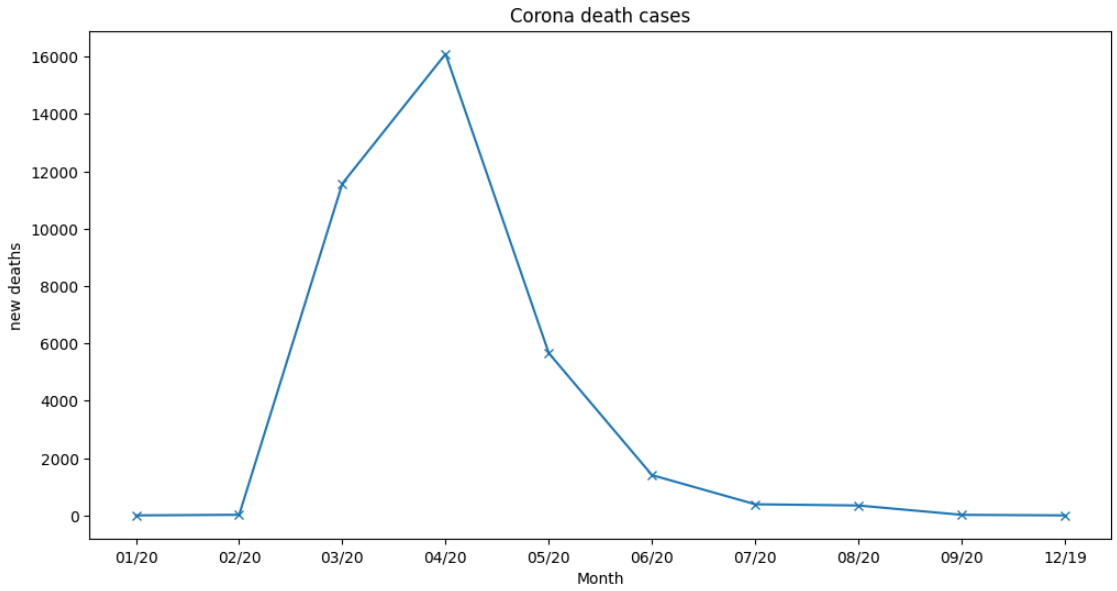
[89]:
```
month_df=read_df.groupby('month').sum(numeric_only=True)
month_df


#The numeric_only parameter determines whether the sum method should
#only include numeric columns when performing the summation.
#By default, this parameter is set to True, which means that non-numeric␣
 ↪columns
#will be excluded from the summation. However, in a future version of pandas,␣
 ↪the default value of numeric_only
#will change to False, which means that non-numeric columns will be included in␣
 ↪the summation unless explicitly excluded.
#if i remove this parameter it will give me warning because by default␣
 ↪numeric_only is True and it will remove 'date' column
# implicitly which will give me a warning. here i specified defualt option␣
 ↪explicitly so it is not giving me any warning.
```

[89]:
```
       new_cases  new_deaths  new_tests
month
01/20          3           0        0.0
02/20        885          21        0.0
03/20     100851       11570        0.0
04/20     101852       16091   419591.0
05/20      29073        5658  1078720.0
06/20       7772        1404   830354.0
07/20       6722         388   797692.0
08/20      21060         345  1098704.0
09/20       3297          20    54395.0
12/19          0           0        0.0
```

[84]:
```
plt.figure(figsize=(12,6))
plt.plot(month_df['new_deaths'],marker='x')
plt.xlabel('Month')
plt.ylabel('new deaths')
plt.title('Corona death cases')
```

[84]: Text(0.5, 1.0, 'Corona death cases')

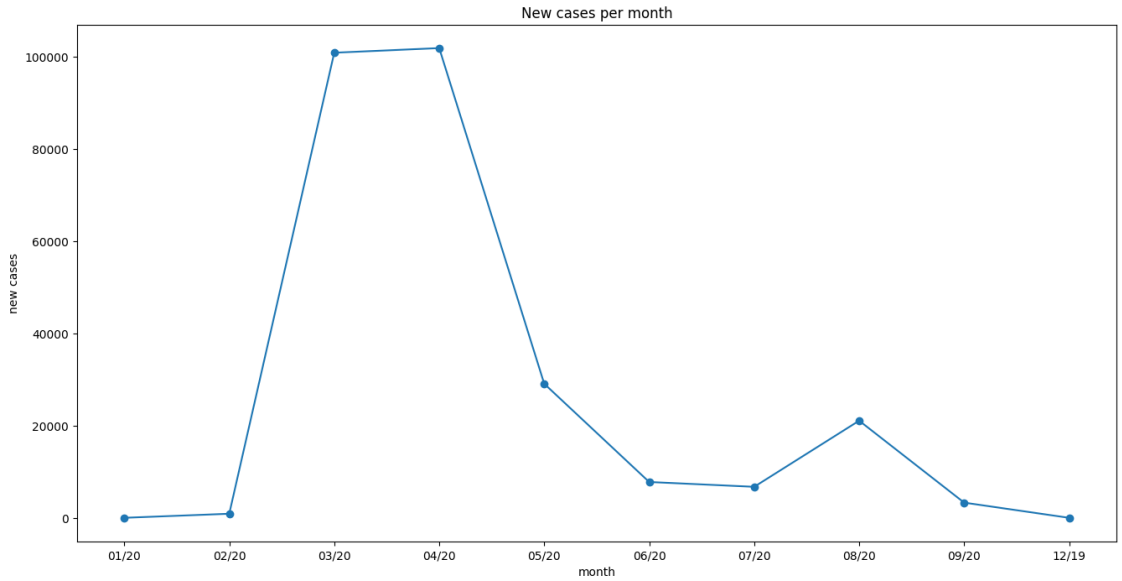


1.2 Task02

Display the graph of new cases verses months

[85]:
```
plt.figure(figsize=(16,8))
plt.plot(month_df['new_cases'],marker='o')
plt.xlabel('month')
plt.ylabel('new cases')
plt.title('New cases per month')
```

[85]: Text(0.5, 1.0, 'New cases per month')

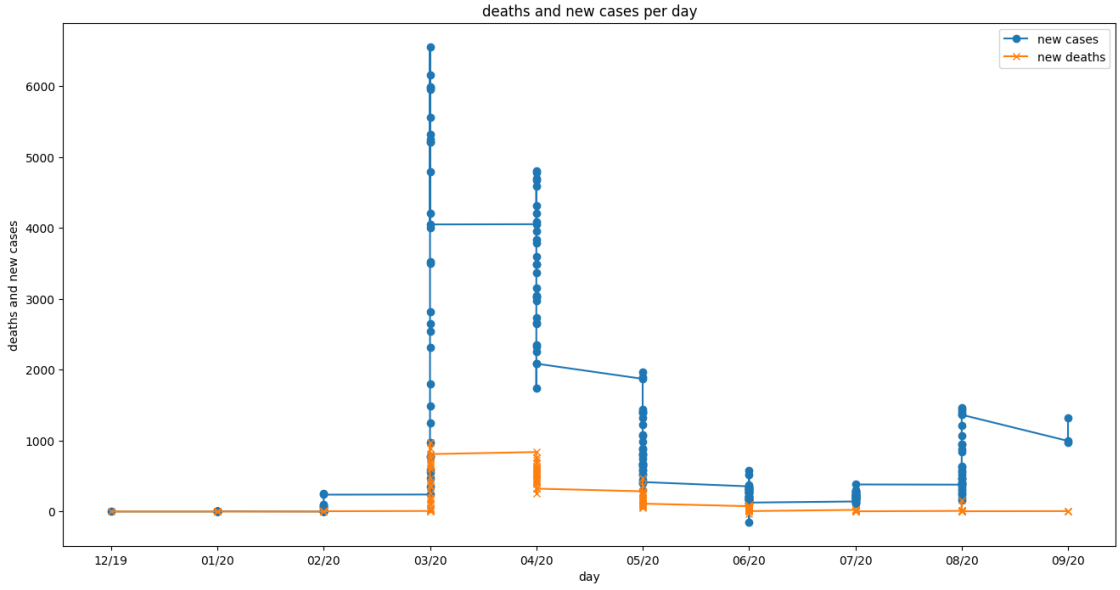


1.3 Task03

Compare the new cases and death cases day-wise on multi-line graph, marks the legends and properly label and title the graph.

[86]:
```
plt.figure(figsize=(16,8))
plt.plot(read_df['month'],read_df['new_cases'],marker='o')
plt.plot(read_df['month'],read_df['new_deaths'],marker='x')
plt.legend(['new cases','new deaths'])
plt.xlabel('day')
plt.ylabel('deaths and new cases')
plt.title('deaths and new cases per day')
```

[86]:
```
Text(0.5, 1.0, 'deaths and new cases per day')
```

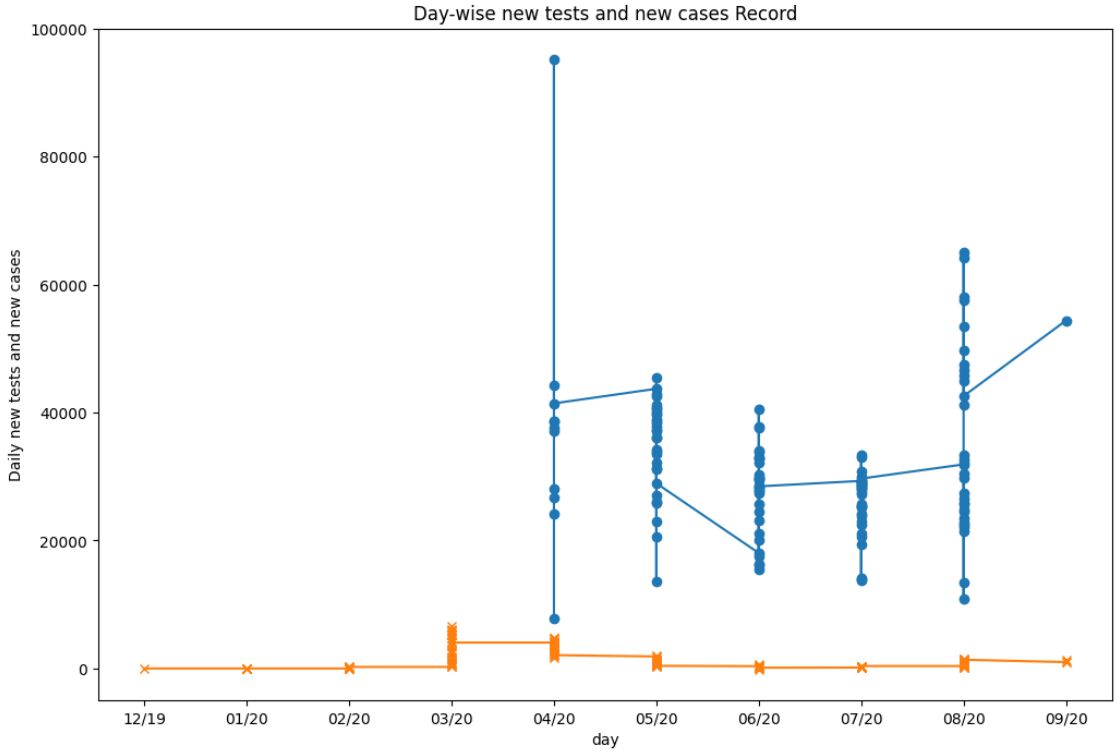


1.4 Task04

Display how the new cases and the new tests are related day-wise on multi-line graph, mark the legends and properly label and title the graph.

[87]:
```
plt.figure(figsize=(12,8))
plt.plot(read_df['month'],read_df.iloc[:,-2],marker='o')
plt.plot(read_df.loc[:,'month'],read_df.iloc[:,1],marker='x')
plt.xlabel('day')
plt.ylabel('Daily new tests and new cases')
plt.title('Day-wise new tests and new cases Record')
```

[87]:
```
Text(0.5, 1.0, 'Day-wise new tests and new cases Record')
```


Day-wise new tests and new cases Record
Daily new tests and new cases
day
100000
80000
60000
40000
20000
0
12/19
01/20
02/20
03/20
04/20
05/20
06/20
07/20
08/20
09/20

[ ]: