Q1)

1)

Assumptions:

We have used ‘tt’ to calculate cumulative sum of confirmed cases, recovered cases and deceased cases within a given range of date.

As we were manipulating on the same dataframe, please ignore the warning(SettingWithCopyWarning).

Methodology:

We have used cumsum() to calculate total cases and grouped by their status then extracted last 3 rows of dataframe to print confirmed count, recovered count and deceased count respectively.

Output:

confirmed_count: 4110211 recovered_count: 3177666 deceased_count: 70094

2)

Methodology:

We have separated the data of Delhi state and stored it in a different dataframe and then used cumsum() to calculate total cases and grouped by their status then extracted last 3 rows of dataframe to print confirmed count, recovered count and deceased count respectively.

Output:

confirmed_count: 188193 recovered_count: 163785 deceased_count: 4538

3)

Methodology:

We have separated the data of Delhi state and Maharastra state and stored them in two different dataframes and then calculated total confirmed, recovered and deceased cases for them separately, and then added them to get the total confirmed, recovered, deceased cases of both states combined.

Output:

confirmed_count: 1072055 recovered_count: 800359 deceased_count: 30813

4)

Assumptions:

We have dropped ‘tt’,’un’,’dd’,’ld’.

Methodology:

We have separated the dataframe according to status(Confirmed, recovered, deceased) into three different dataframes. Then we calculated sum to get total cases of each state, and used max() to retrieve the state with maximum confirmed, recovered and deceased counts respectively.

Output:

Confirmed

Highest affected State is: mh

Highest affected State count is: 883862

Recovered

Highest affected State is: mh

Highest affected State count is: 636574

Deceased

Highest affected State is: mh

Highest affected State count is: 26275

5)

Assumptions:

We have dropped ‘tt’,’un’,’dd’,’ld’.

Methodology:

We have separated the dataframe according to status (Confirmed, recovered, deceased) into three different dataframes. Then we calculated sum to get total cases of each state, and used min() to retrieve the state with minimum confirmed, recovered and deceased counts respectively.

Output:

Confirmed

Lowest affected State is: mz

Lowest affected State count is: 1062

Recovered

Lowest affected State is: mz

Lowest affected State count is: 713

Deceased

Lowest affected State is: mz

Lowest affected State count is: 0

6)

Methodology:

We separated the data of Delhi state into a new dataframe, then used groupby() to group data based on status (confirmed, recovered, deceased) and then used max() to get the day with maximum confirmed, recovered and deceased count spike.

Output:

Confirmed

Day: 2020-06-23

Count: 3947

Recovered

Day: 2020-06-20

Count: 7725

Deceased

Day: 2020-06-16

Count: 437

7)

Assumptions:

We have dropped ‘tt’,’un’.

Methodology:

We have separated the dataframe according to status (Confirmed, recovered, deceased) into three different dataframes. Then set the index of each dataframe as ‘date’. Next we have subtracted (recovered dataframe + deceased dataframe) from confirmed dataframe and stored it into a new dataframe which contains the active cases. Then we have used sum() to get the total number of active cases per state.

Output:

an 343

ap 100880

ar 1525

as 28404

br 16735

ch 2143

ct 22320

dd 0

dl 19870

dn 301

ga 4945

gj 16266

hp 2023

hr 14912

jh 14980

jk 9547

ka 100224

kl 21867

la 834

ld 0

mh 221013

ml 1374

mn 1872

mp 15687

mz 349

nl 701

or 25856

pb 15870

py 5163

rj 14996

sk 561

tg 32405

tn 51580

tr 5905

up 59963

ut 7649

wb 23390

dtype: int64

Q2

1)

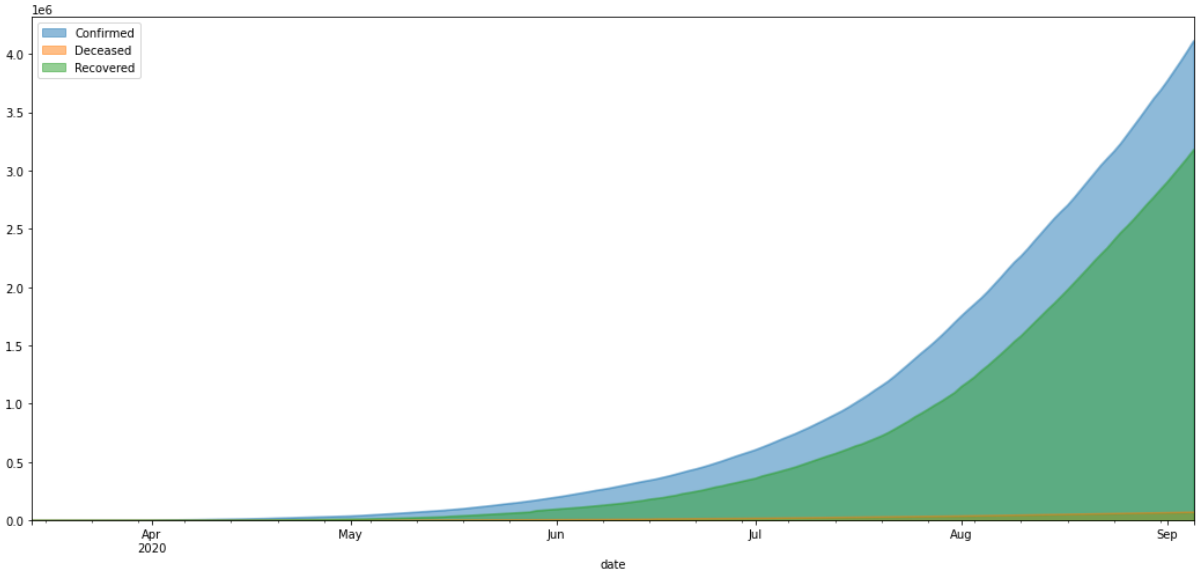
Assumptions:

As we were manipulating on the same dataframe, please ignore the warning (SettingWithCopyWarning).

Methodology:

We have used cumsum() to calculate total cases and grouped by their status then extracted last 3 rows of dataframe to print confirmed count, recovered count and deceased count respectively. Then we have used matplotlib to show the area plot.

Output:

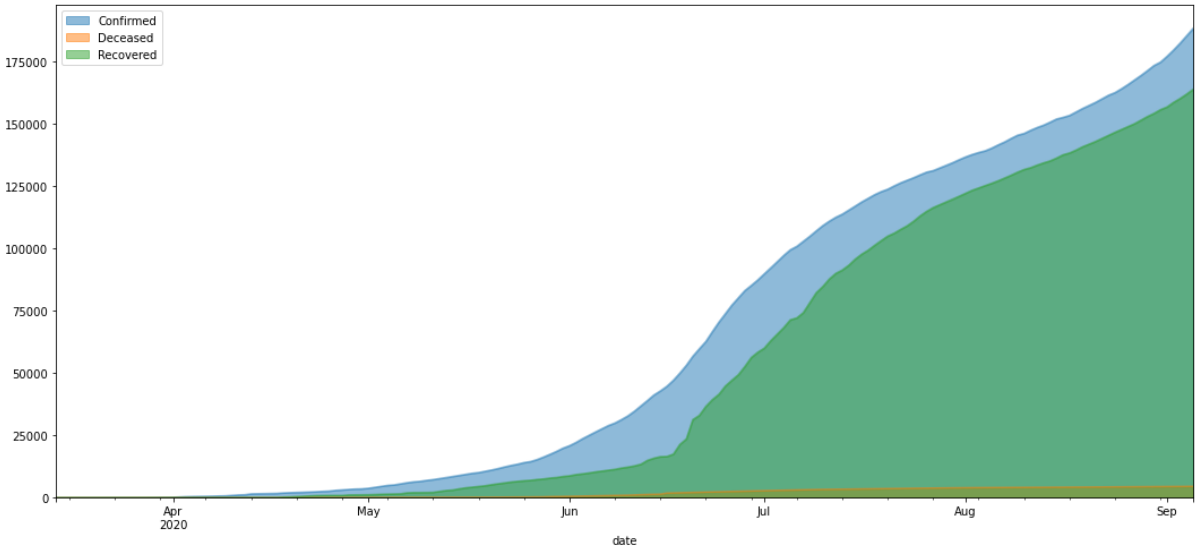


2)

Methodology:

We have separated the data of Delhi state and stored it in a different dataframe and then used cumsum() to calculate total cases and grouped by their status then extracted last 3 rows of dataframe to print confirmed count, recovered count and deceased count respectively. Then we have used matplotlib to show the area plot.

Output:



3)

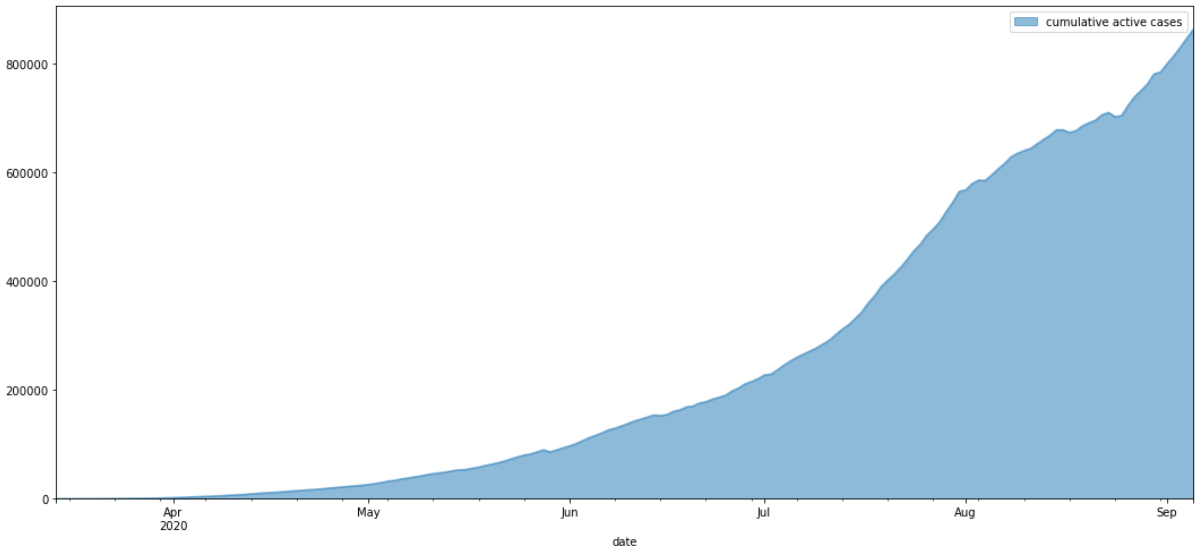
Assumptions:

As we were manipulating on the same dataframe, please ignore the warning (SettingWithCopyWarning).

Methodology:

We have separated the dataframe according to status (Confirmed, recovered, deceased) into three different dataframes. Then set the index of each dataframe as ‘date’. Next we have subtracted (recovered dataframe + deceased dataframe) from confirmed dataframe and stored it into a new dataframe which contains the active cases. Then we have used sum() to get the total number of active cases per state. Then we have used matplotlib to show the area plot.

Output:



Q3)

Assumptions:

We have considered date to start from 1 for the ease of calculation of Linear Regression intercept and slope.

Methodology:

We have separated the data of Delhi state based on status and stored them into three different dataframes. We have created a function Linear Regression, which calculates the slope and intercept of Linear Regression line using Regression formula. Now we call this Linear Regression function to calculate the slope and intercept of each of the three dataframes.

Output

Confirmed Slope and intercept

12.2142692053709

-11.684415584415774

Recovered Slope and intercept

12.305528285274049

-158.44266233766245

Deceased Slope and intercept

0.19023332599603787

8.948441558441559