Covid plots

August 21, 2020

[2]:

```
import os
os.getcwd()

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import csv
```

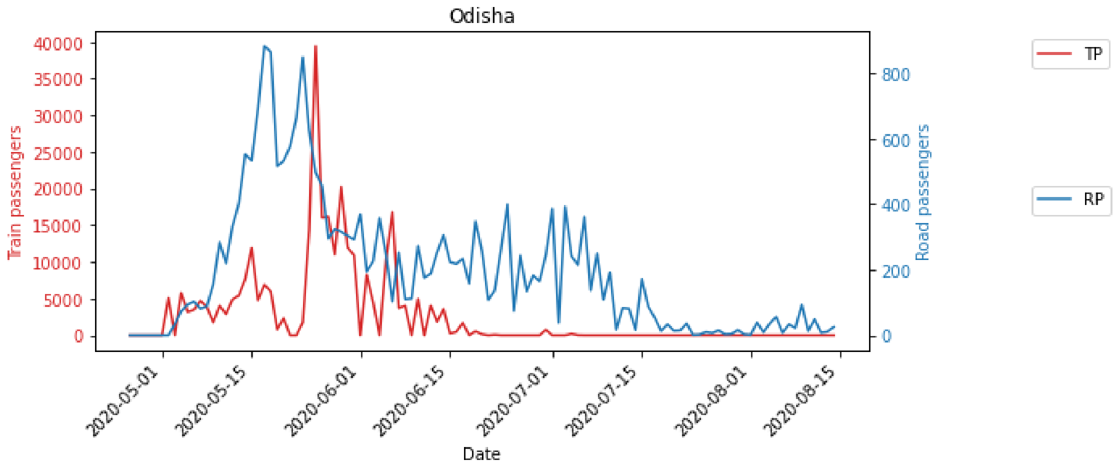
[134]:

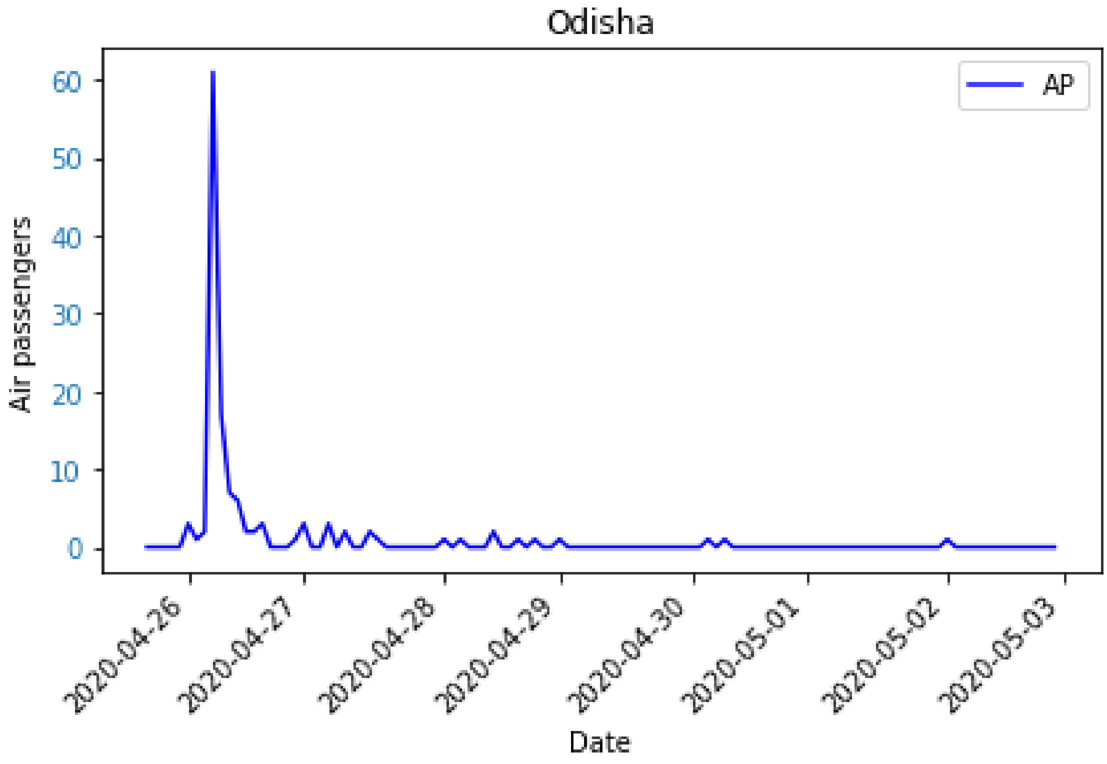
```
def␣
 ↪my_single_axis_plot(xdata,yseries,colorseries,xlabel,ylabel,linelabels,xlabelcolor,ylabelco
 ↪
    fig, ax1 = plt.subplots(figsize=(6, 4))
    n = len(yseries)
    ax1.set_xlabel(xlabel)
    ax1.set_ylabel(ylabel)
    ax1.tick_params(axis='x', labelcolor=xlabelcolor)
    ax1.tick_params(axis='y', labelcolor=ylabelcolor)

    for i in range(n):
        ax1.plot(xdata, yseries[i], color=colorseries[i], label=linelabels[i])
        ax1.legend()

    ax1.set_xticklabels(xdata, rotation = 45, ha="right")

    fig.tight_layout()  # otherwise the right y-label is slightly clipped
    plt.title(areaname)
    plt.show()

def my_double_axes_plot(xdata,
                        yseries1,colorseries1,
                        yseries2,colorseries2,
                        xlabel,ylabel1,ylabel2,
                        linelabels1,linelabels2,
                        xlabelcolor,ylabelcolor1,ylabelcolor2,
                       areaname):
    fig, ax1 = plt.subplots(figsize=(10, 4))
    ax1.set_xlabel(xlabel)
    ax1.set_ylabel(ylabel1, color=ylabelcolor1)
    p = []
    for i in range(len(yseries1)):
       ax1.plot(xdata, yseries1[i], color=colorseries1[i], label=linelabels1[i])
        ax1.legend(bbox_to_anchor=(1.2, 1), loc='upper left')
    ax1.tick_params(axis='y', labelcolor=color)
    ax1.set_xticklabels(xdata, rotation = 45, ha="right")
    ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis

    ax2.set_ylabel(ylabel2, color=ylabelcolor2)  # we already handled the␣
↪x-label with ax1
    for i in range(len(yseries2)):
       ax2.plot(xdata, yseries2[i], color=colorseries2[i], label=linelabels2[i])
        ax2.legend(bbox_to_anchor=(1.2, 0.4), loc='lower left')
    ax2.tick_params(axis='y', labelcolor=ylabelcolor2)

    fig.tight_layout()  # otherwise the right y-label is slightly clipped
    plt.title(areaname)
    plt.show()
```

[152]:
```
def read_dataset(areaname):
    od = pd.read_excel('covid_processed.xlsx',sheet_name=areaname, skiprows=0)
    ndates = od['Date']
    dates = []
    for i in range(len(ndates)):
        dates.append(ndates[i].date())

    return (od, dates)

od, dates = read_dataset('Odisha')
```

0.0.1 1.a The following plot shows the daily passengers from train, road, and air.

[137]:
```
def plot_train_road_air_passengers(dataset, dates, areaname):
    redcolor = 'tab:red'
    n_daily_train = od['Train passengers']
    n_daily_road = od['Road passengers']
    n_daily_air = od['Air passengers']

    my_double_axes_plot(dates,
                            [n_daily_train],['tab:red'],
                            [n_daily_road],['tab:blue'],
                            'Date','Train passengers','Road passengers',
                            ['TP'],['RP'],
                            'black','tab:red','tab:blue',areaname)
    my_single_axis_plot(dates,[n_daily_air],['blue'],
                        'Date','Air passengers',['AP'],'black','tab:blue',areaname)

plot_train_road_air_passengers(od, dates, 'Odisha')
```

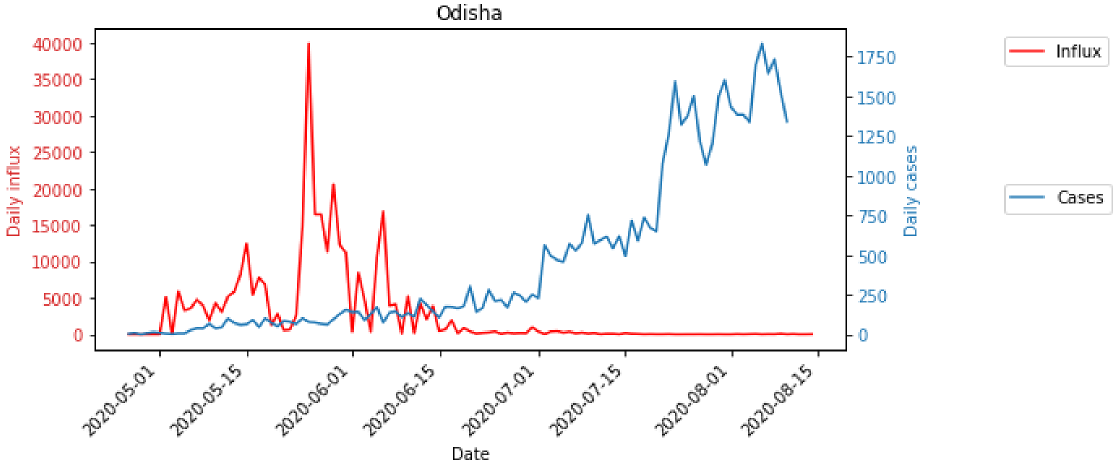




0.0.2 1.b The following plot shows the daily influx and daily cases.

[139]:

```
def plot_influx_cases(dataset,dates,areaname):
    n_daily_influx = od['Train passengers'] + od['Road passengers'] + od['Air␣
 ↪passengers']
    len(n_daily_influx)
    n_daily_cases = od['Positive cases']
    my_double_axes_plot(dates,
                        [n_daily_influx],['red'],
                        [n_daily_cases],['tab:blue'],
                        'Date','Daily influx','Daily cases',
                        ['Influx'],['Cases'],
                        'black','tab:red','tab:blue',areaname)
plot_influx_cases(od,dates,'Odisha')
```

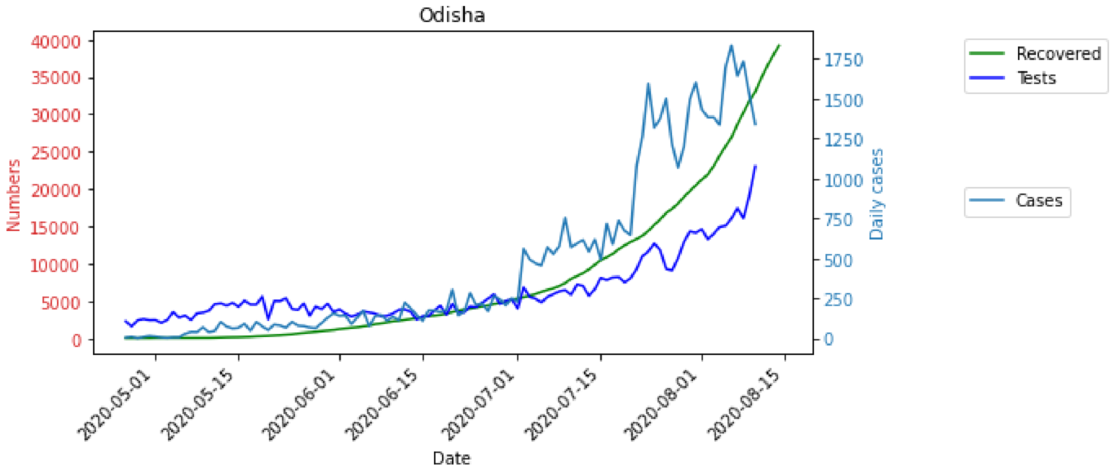


1 1.c The following plot shows the daily cases, recovered, and deaths.

[153]:

```
def plot_cases_recovered_tests(dataset, dates,areaname):
    n_daily_cases = od['Positive cases']
    n_daily_recovered = od['Recovered']
    daily_tests = od['Total Tested']
    my_double_axes_plot(dates,
                        [n_daily_recovered,daily_tests],['green','blue'],
                        [n_daily_cases],['tab:blue'],
                        'Date','Numbers','Daily cases',
                        ['Recovered','Tests'],['Cases'],
                        'black','tab:red','tab:blue',areaname)
```

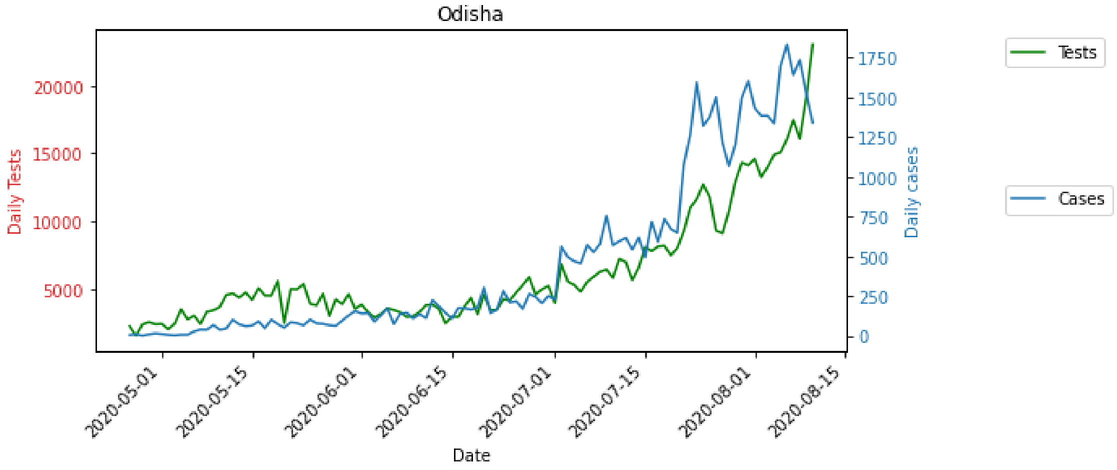
```
def plot_cases_recovered(dataset, dates,areaname):
    n_daily_cases = od['Positive cases']
    n_daily_recovered = od['Recovered']
    my_double_axes_plot(dates,
                            [n_daily_recovered],['green'],
                            [n_daily_cases],['tab:blue'],
                            'Date','Numbers','Daily cases',
                            ['Recovered'],['Cases'],
                            'black','tab:red','tab:blue',areaname)
plot_cases_recovered_tests(od, dates, 'Odisha')
```



1.0.1 1.d The following plot shows the daily cases and tests.

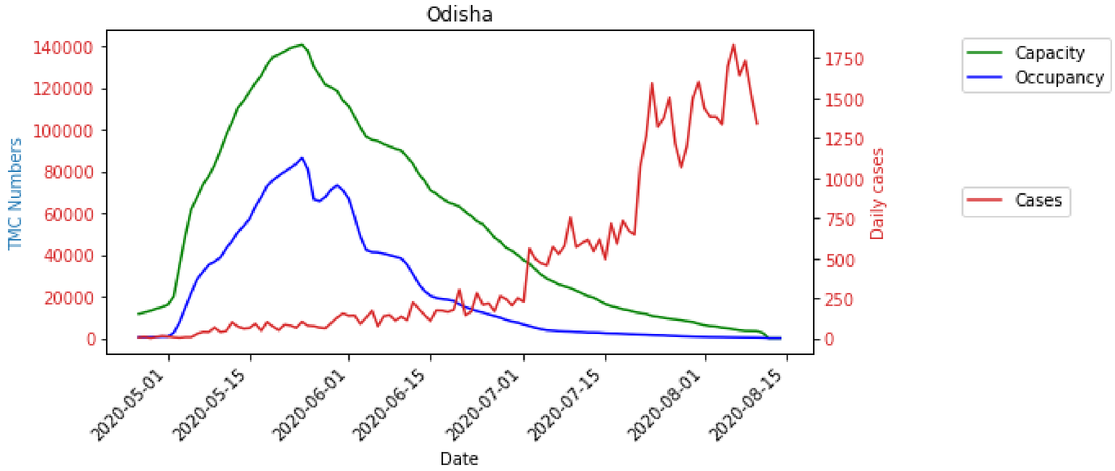
[142]:

```
def plot_cases_tests(dataset, dates,areaname):
    n_daily_cases = od['Positive cases']
    daily_tests = od['Total Tested']

    my_double_axes_plot(dates,
                            [daily_tests],['green'],
                            [n_daily_cases],['tab:blue'],
                            'Date','Daily Tests','Daily cases',
                            ['Tests'],['Cases'],
                            'black','tab:red','tab:blue',areaname)
plot_cases_tests(od, dates,'Odisha')
```



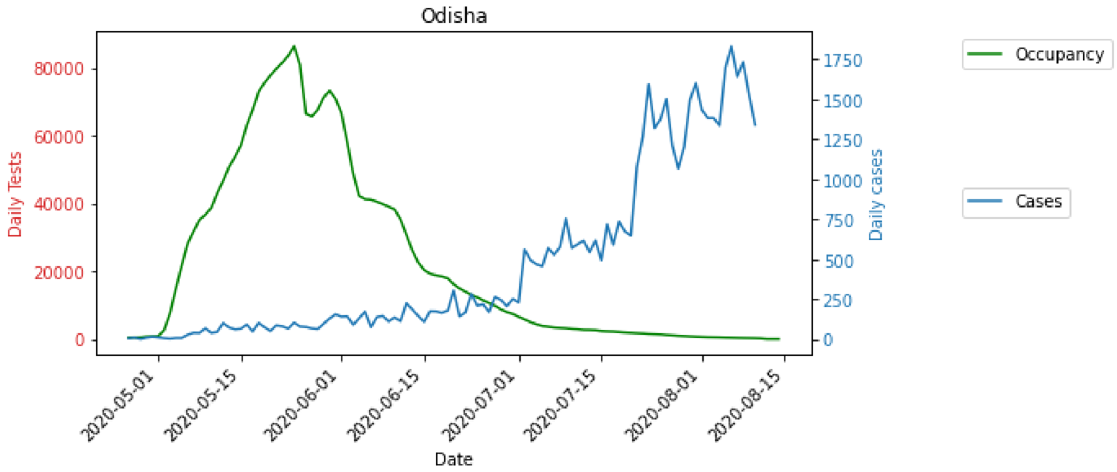
1.0.2 1.e The following plot shows the daily TMC capacity, occupancy, and daily cases.

```
[143]: def plot_tmc_capacity_occupancy_cases(dataset,dates,areaname):
           tmc_capacity = od['Total_TMC_capacity']
           tmc_occupancy = od['Total_TMC_occupants']
           my_double_axes_plot(dates,
                               [tmc_capacity,tmc_occupancy],['green','blue'],
                               [n_daily_cases],['tab:red'],
                               'Date','TMC Numbers','Daily cases',
                               ['Capacity','Occupancy'],['Cases'],
                               'black','tab:blue','tab:red',areaname)

       plot_tmc_capacity_occupancy_cases(od,dates,'Odisha')
```



1.0.3 1.f TMC occupancy and total tests

```
[144]: def plot_occupancy_tests(dataset, dates,areaname):
           tmc_occupancy = od['Total_TMC_occupants']
           daily_tests = od['Total Tested']

           my_double_axes_plot(dates,
                               [tmc_occupancy],['green'],
                               [n_daily_cases],['tab:blue'],
                               'Date','Daily Tests','Daily cases',
                               ['Occupancy'],['Cases'],
                               'black','tab:red','tab:blue',areaname)

       plot_occupancy_tests(od, dates,'Odisha')
```

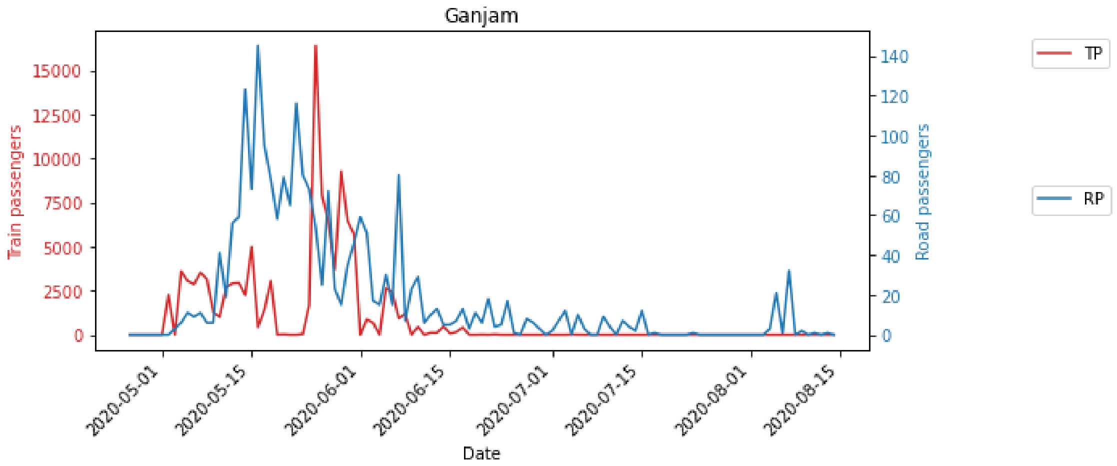


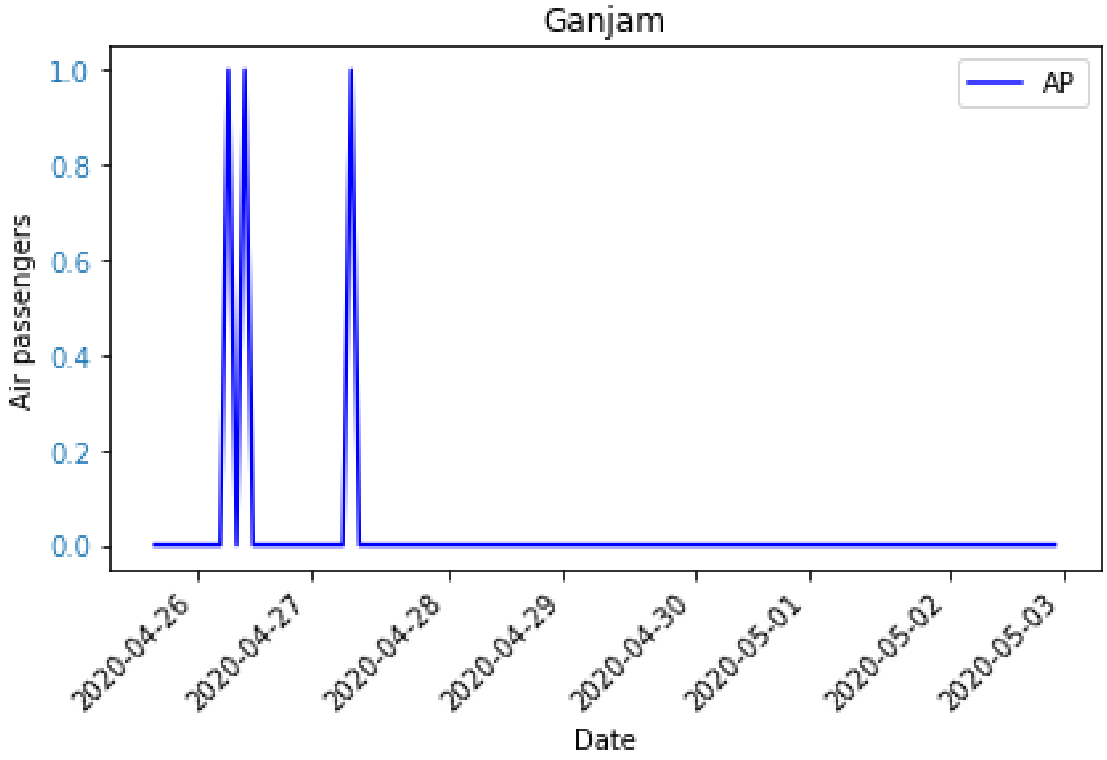
1.0.4 A few important districts

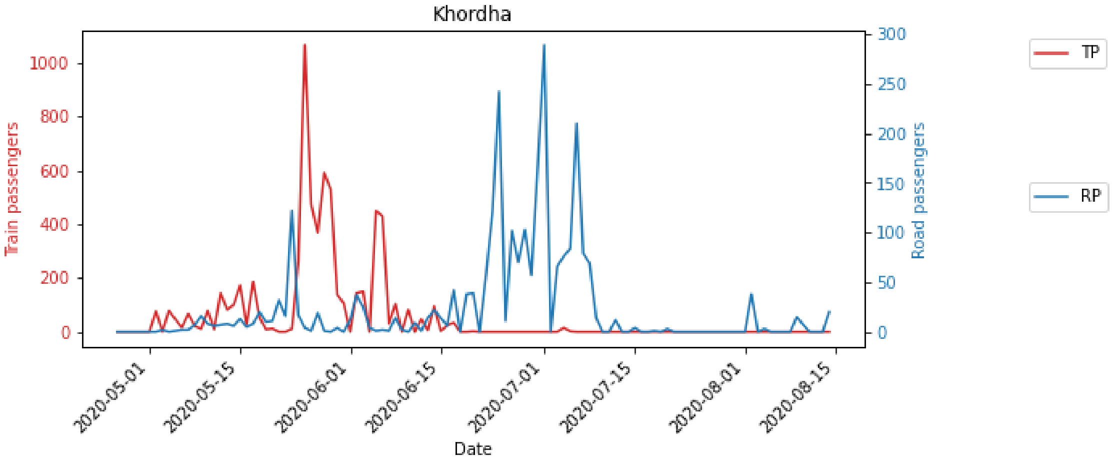
```
[132]: # imp_districts = ['Ganjam', 'Khordha', 'Cuttack', 'Sundargarh', 'Rayagada']
       imp_districts = ['Ganjam', 'Khordha', 'Cuttack', 'Sundargarh']
```

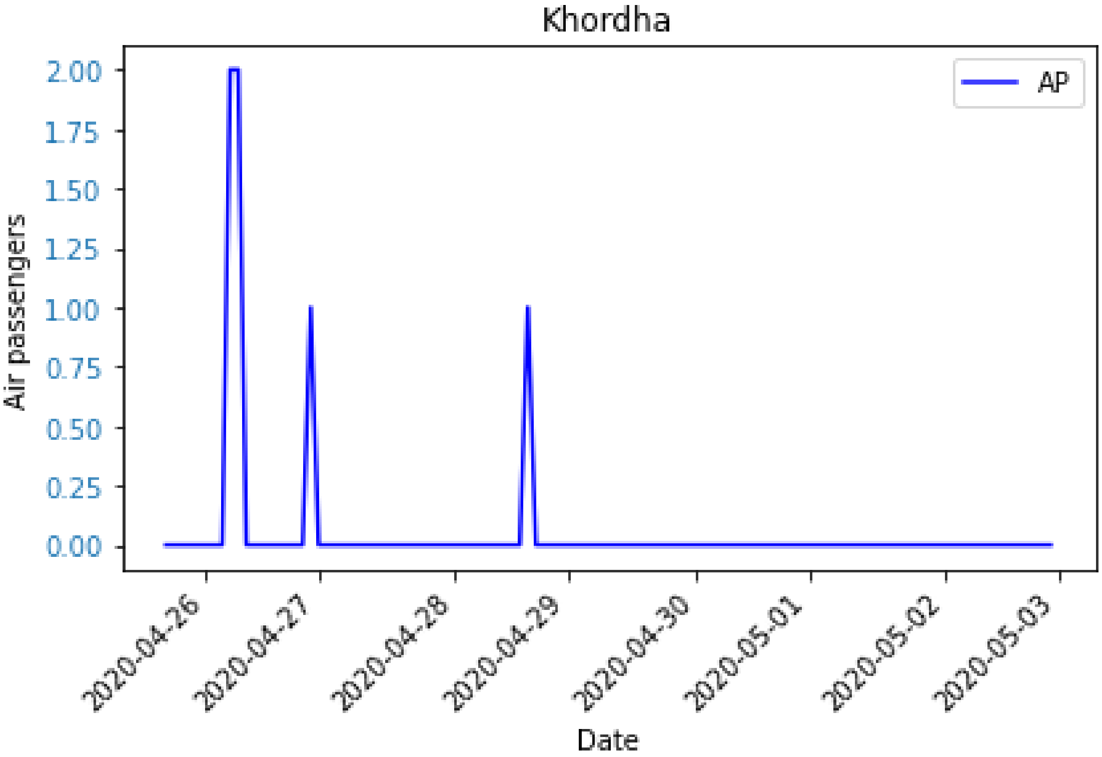
```
[146]: def plot_1a_1f(areaname):
           od, dates = read_dataset(areaname)
           plot_train_road_air_passengers(od, dates)
           plot_influx_cases(od,dates)
           plot_cases_recovered_tests(od, dates)
           plot_cases_tests(od, dates)
```

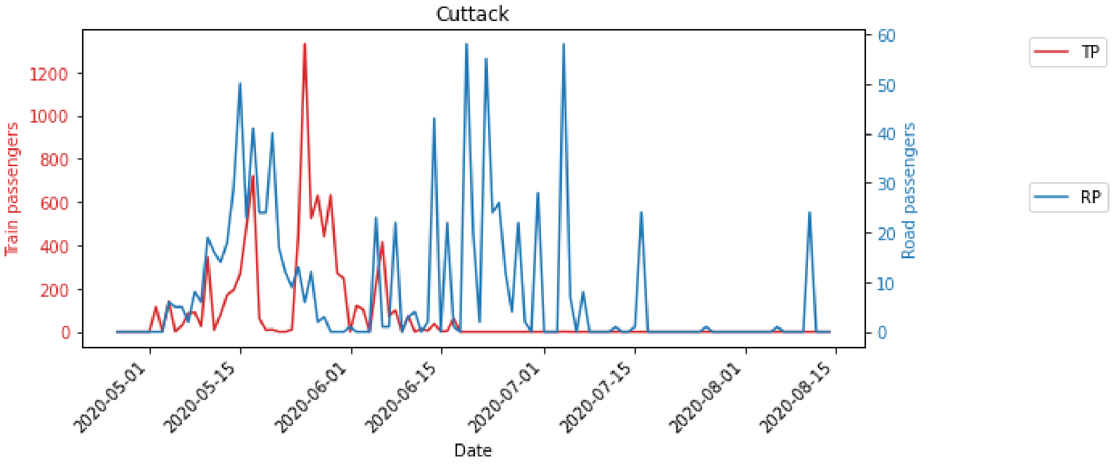
```
    plot_tmc_capacity_occupancy_cases(od,dates)
    plot_occupancy_tests(od, dates)

for areaname in imp_districts:
    od, dates = read_dataset(areaname)
    plot_train_road_air_passengers(od, dates, areaname)
```

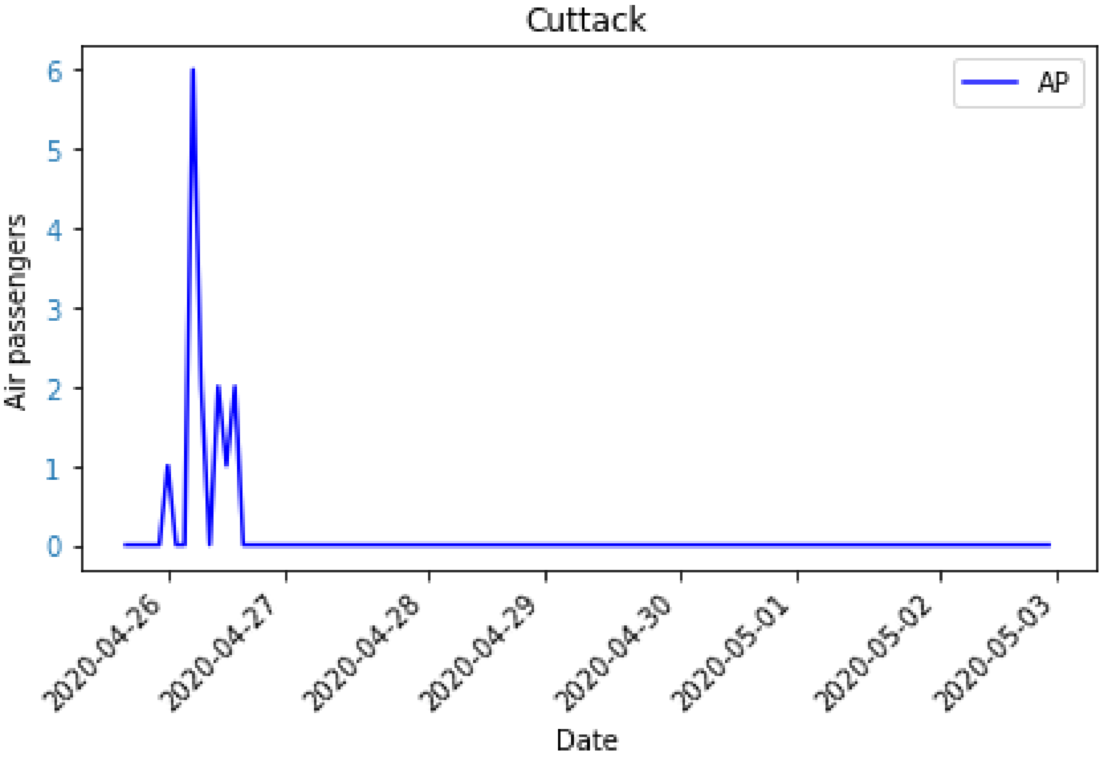


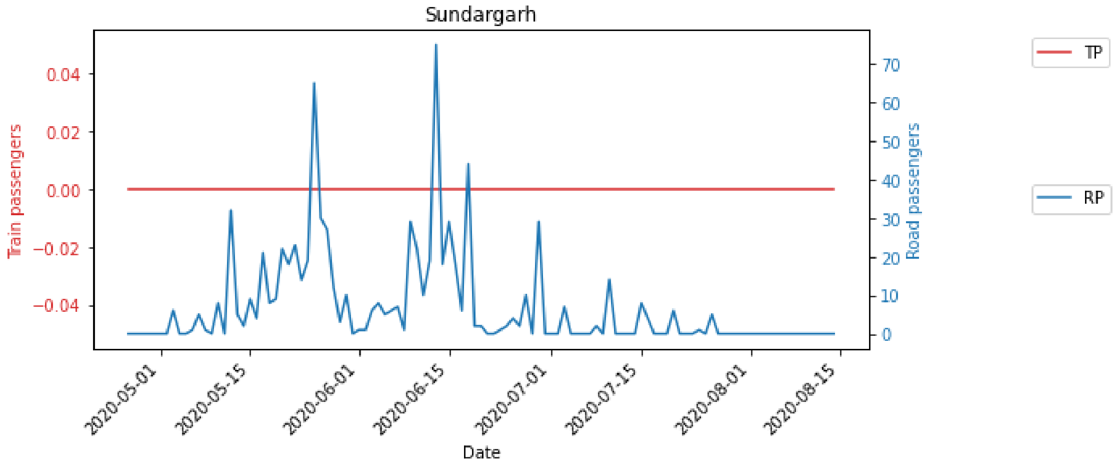


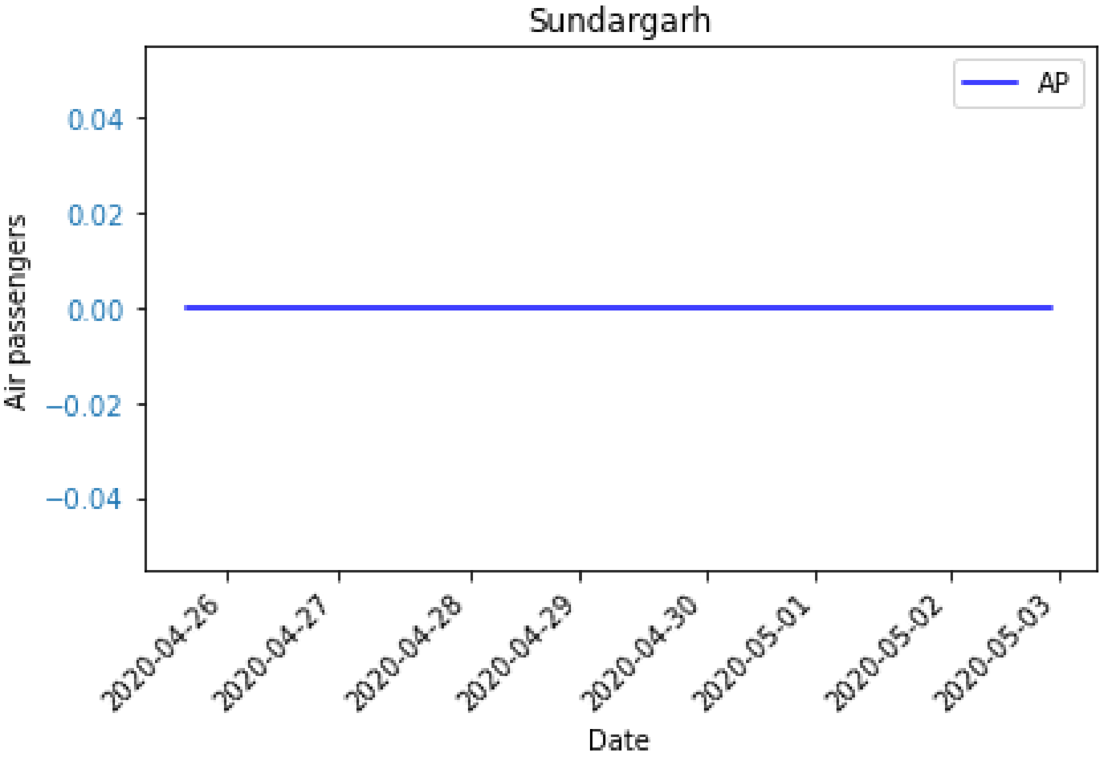

Khordha
Train passengers
0
200
400
600
800
1000
Road passengers
0
50
100
150
200
250
300
TP
RP
2020-05-01
2020-05-15
2020-06-01
2020-06-15
2020-07-01
2020-07-15
2020-08-01
2020-08-15
Date


Khordha
Air passengers
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
AP
2020-04-26
2020-04-27
2020-04-28
2020-04-29
2020-04-30
2020-05-01
2020-05-02
2020-05-03
Date


Cuttack
Train passengers
0
200
400
600
800
1000
1200
Road passengers
0
10
20
30
40
50
60
TP
RP
2020-05-01
2020-05-15
2020-06-01
2020-06-15
2020-07-01
2020-07-15
2020-08-01
2020-08-15
Date

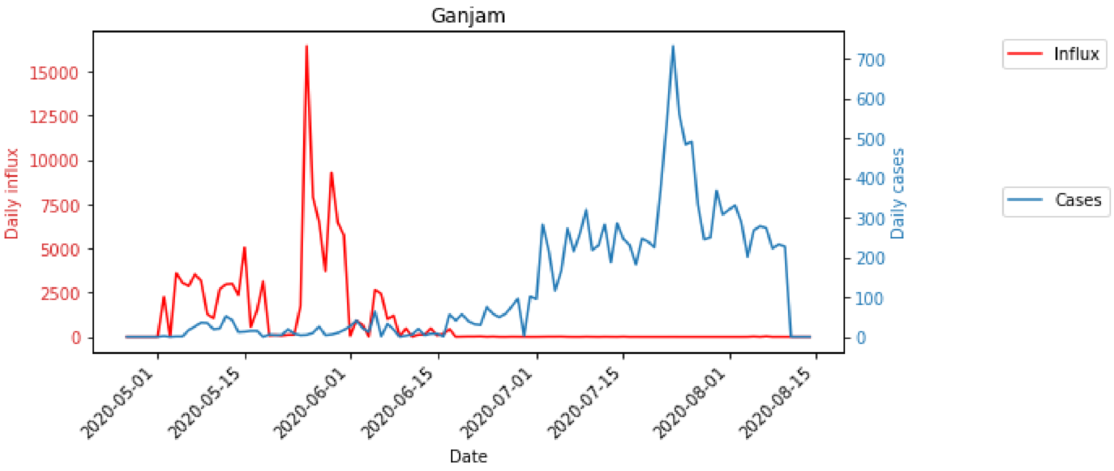

Cuttack
Air passengers
0
1
2
3
4
5
6
AP
2020-04-26
2020-04-27
2020-04-28
2020-04-29
2020-04-30
2020-05-01
2020-05-02
2020-05-03
Date

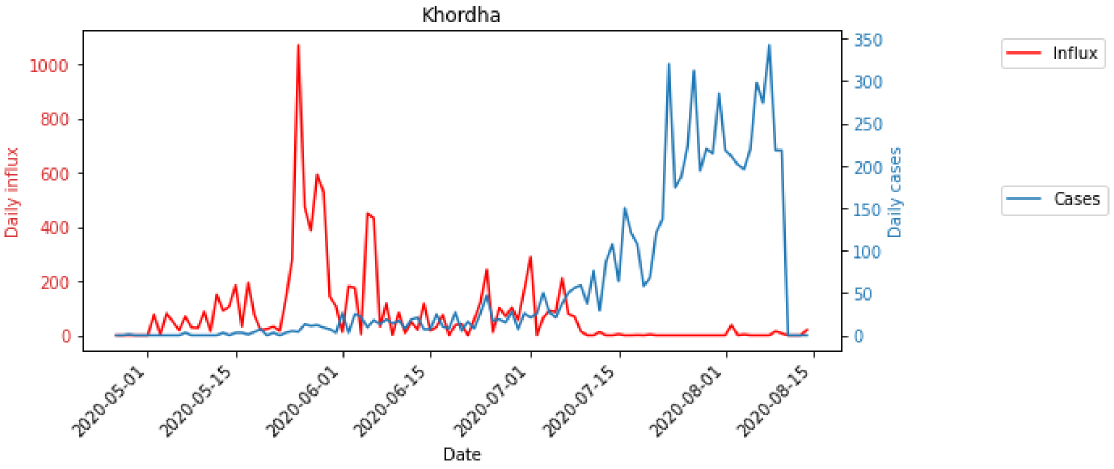


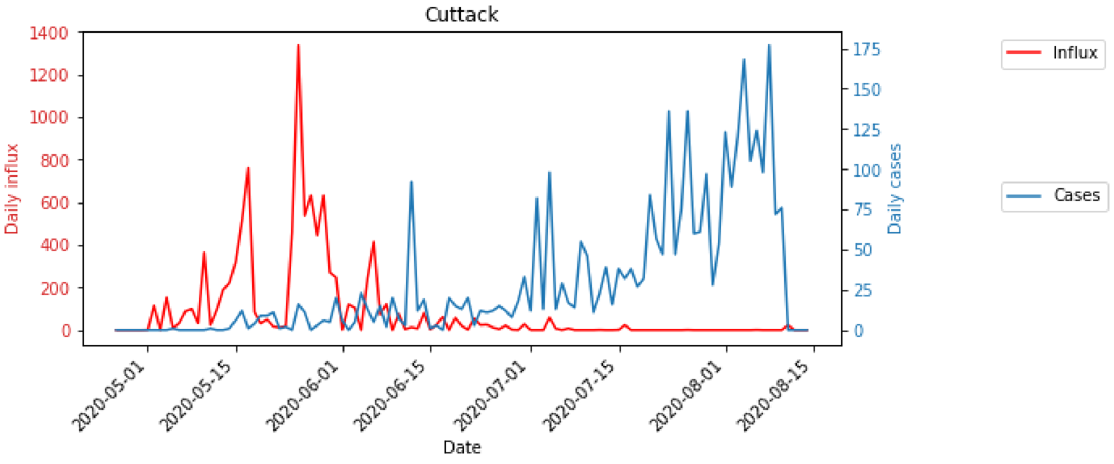


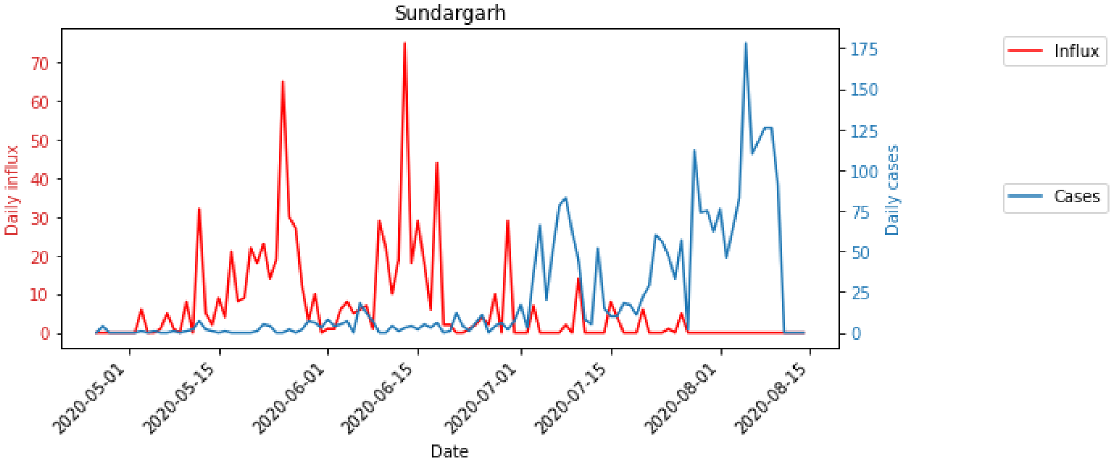
[147]:

```
for areaname in imp_districts:
    od, dates = read_dataset(areaname)
    plot_influx_cases(od,dates,areaname)
```

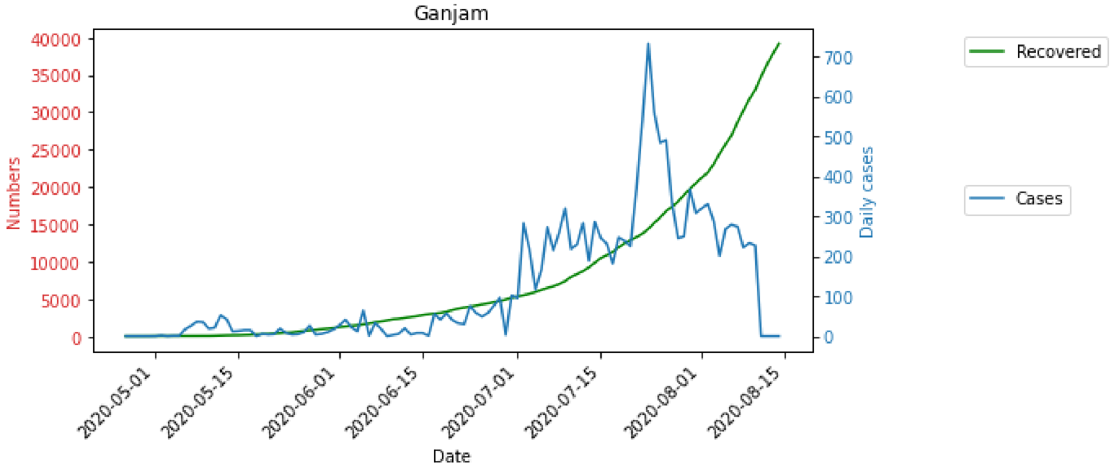

Ganjam
Daily influx
15000
12500
10000
7500
5000
2500
0
Daily cases
700
600
500
400
300
200
100
0
2020-05-01
2020-05-15
2020-06-01
2020-06-15
2020-07-01
2020-07-15
2020-08-01
2020-08-15
Date
Influx
Cases

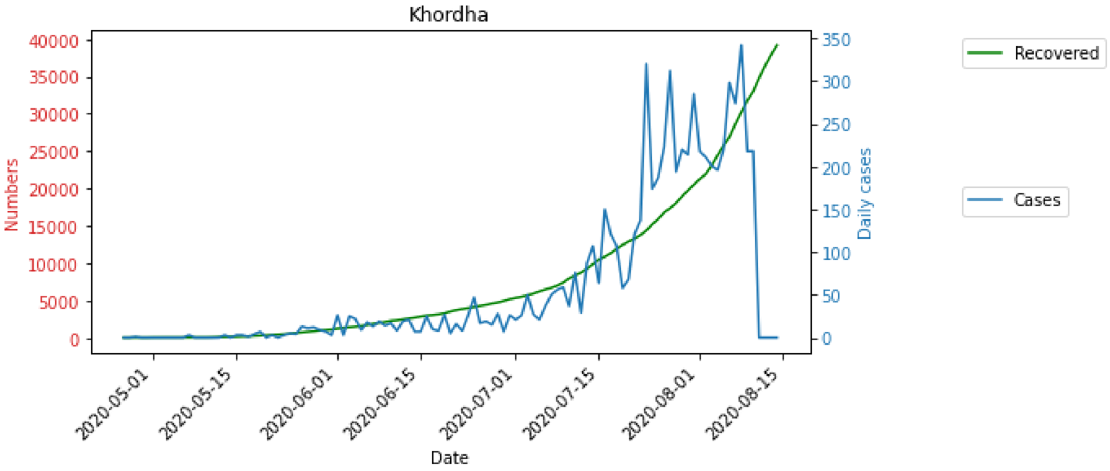

Khordha
Daily influx
1000
800
600
400
200
0
Daily cases
350
300
250
200
150
100
50
0
2020-05-01
2020-05-15
2020-06-01
2020-06-15
2020-07-01
2020-07-15
2020-08-01
2020-08-15
Date
Influx
Cases

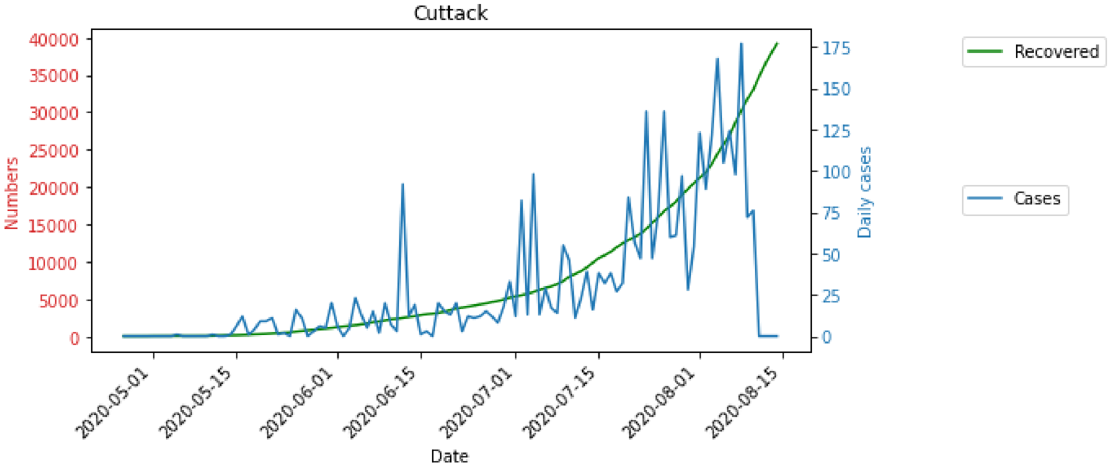

Cuttack
Daily influx
1400
1200
1000
800
600
400
200
0
Daily cases
175
150
125
100
75
50
25
0
2020-05-01
2020-05-15
2020-06-01
2020-06-15
2020-07-01
2020-07-15
2020-08-01
2020-08-15
Date
Influx
Cases

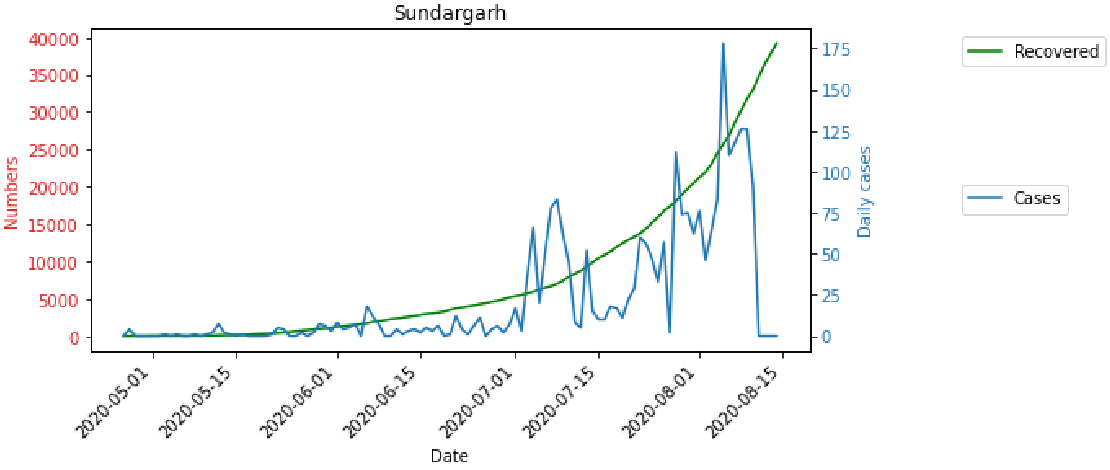


```
[154]: for areaname in imp_districts:
           od, dates = read_dataset(areaname)
           plot_cases_recovered(od,dates,areaname)
```




Khordha
Numbers
40000
35000
30000
25000
20000
15000
10000
5000
0
Daily cases
350
300
250
200
150
100
50
0
2020-05-01
2020-05-15
2020-06-01
2020-06-15
2020-07-01
2020-07-15
2020-08-01
2020-08-15
Date
Recovered
Cases


Cuttack
Numbers
40000
35000
30000
25000
20000
15000
10000
5000
0
Daily cases
175
150
125
100
75
50
25
0
2020-05-01
2020-05-15
2020-06-01
2020-06-15
2020-07-01
2020-07-15
2020-08-01
2020-08-15
Date
Recovered
Cases


Sundargarh
Numbers
40000
35000
30000
25000
20000
15000
10000
5000
0
Daily cases
175
150
125
100
75
50
25
0
2020-05-01
2020-05-15
2020-06-01
2020-06-15
2020-07-01
2020-07-15
2020-08-01
2020-08-15
Date
Recovered
Cases

[ ]:

```
for areaname in imp_districts:
    od, dates = read_dataset(areaname)
    plot_cases_tests(od,dates,areaname)
```

[ ]:

```
for areaname in imp_districts:
    od, dates = read_dataset(areaname)
    plot_occupancy_tests(od,dates,areaname)
```

[ ]:

```
```