Tamil Nadu Covid-19 Analysis

Web Mining Project

By Akshay Kumar (17BCE1290)

Import packages

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime
import requests
import warnings
warnings.filterwarnings('ignore')
import plotly.express as px
import plotly.graph_objects as go
import seaborn as sn
```

Read CSV

In [2]:
```
df=pd.read_csv('covid19-in-india/covid_19_india.csv')
```

Preliminary Data Analysis

In [3]:
```
df
```

Out[3]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
0	1	30/01/20	6:00 PM	Kerala	1	0	0	0	1
1	2	31/01/20	6:00 PM	Kerala	1	0	0	0	1
2	3	01/02/20	6:00 PM	Kerala	2	0	0	0	2
3	4	02/02/20	6:00 PM	Kerala	3	0	0	0	3
4	5	03/02/20	6:00 PM	Kerala	3	0	0	0	3
...	...	...	...	...	...	...	...	...	...
1699	1700	07/05/20	8:00 AM	Telengana	-	-	628	29	1107
1700	1701	07/05/20	8:00 AM	Tripura	-	-	2	0	43
1701	1702	07/05/20	8:00 AM	Uttarakhand	-	-	39	1	61
1702	1703	07/05/20	8:00 AM	Uttar Pradesh	-	-	1130	60	2998
1703	1704	07/05/20	8:00 AM	West Bengal	-	-	364	144	1456

1704 rows × 9 columns

In [4]:
```
df['State/UnionTerritory'].unique()
```

Out[4]:
```
array(['Kerala', 'Telengana', 'Delhi', 'Rajasthan', 'Uttar Pradesh',
       'Haryana', 'Ladakh', 'Tamil Nadu', 'Karnataka', 'Maharashtra',
       'Punjab', 'Jammu and Kashmir', 'Andhra Pradesh', 'Uttarakhand',
       'Odisha', 'Puducherry', 'West Bengal', 'Chhattisgarh',
       'Chandigarh', 'Gujarat', 'Himachal Pradesh', 'Madhya Pradesh',
       'Bihar', 'Manipur', 'Mizoram', 'Andaman and Nicobar Islands',
       'Goa', 'Unassigned', 'Assam', 'Jharkhand', 'Arunachal Pradesh',
       'Tripura', 'Nagaland', 'Meghalaya', 'Nagaland#', 'Jharkhand#',
       'Dadar Nagar Haveli'], dtype=object)
```

In [5]:
```
df['State/UnionTerritory'].nunique()
```

Out[5]:
```
37
```

In [6]:
```
tn=df[df['State/UnionTerritory']=='Tamil Nadu']
```

In [7]:
```
tn.head(20)
```

Out[7]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
61	62	07/03/20	6:00 PM	Tamil Nadu	1	0	0	0	1
67	68	08/03/20	6:00 PM	Tamil Nadu	1	0	0	0	1
79	80	09/03/20	6:00 PM	Tamil Nadu	1	0	0	0	1
94	95	10/03/20	6:00 PM	Tamil Nadu	1	0	0	0	1
105	106	11/03/20	6:00 PM	Tamil Nadu	1	0	0	0	1
116	117	12/03/20	6:00 PM	Tamil Nadu	1	0	0	0	1
129	130	13/03/20	6:00 PM	Tamil Nadu	1	0	0	0	1
142	143	14/03/20	6:00 PM	Tamil Nadu	1	0	0	0	1
156	157	15/03/20	6:00 PM	Tamil Nadu	1	0	0	0	1
171	172	16/03/20	6:00 PM	Tamil Nadu	1	0	0	0	1
186	187	17/03/20	6:00 PM	Tamil Nadu	1	0	0	0	1
202	203	18/03/20	6:00 PM	Tamil Nadu	1	0	0	0	1
220	221	19/03/20	6:00 PM	Tamil Nadu	2	0	1	0	2
240	241	20/03/20	6:00 PM	Tamil Nadu	3	0	1	0	3
262	263	21/03/20	6:00 PM	Tamil Nadu	3	0	1	0	3
285	286	22/03/20	6:00 PM	Tamil Nadu	5	2	1	0	7
308	309	23/03/20	6:00 PM	Tamil Nadu	7	2	1	0	9
332	333	24/03/20	6:00 PM	Tamil Nadu	13	2	1	0	15
357	358	25/03/20	6:00 PM	Tamil Nadu	16	2	1	0	18
387	388	26/03/20	6:00 PM	Tamil Nadu	20	6	1	1	26

In [8]:
```
tn.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 62 entries, 61 to 1698
Data columns (total 9 columns):
Sno                         62 non-null int64
Date                        62 non-null object
Time                        62 non-null object
State/UnionTerritory        62 non-null object
ConfirmedIndianNational     62 non-null object
ConfirmedForeignNational    62 non-null object
Cured                       62 non-null int64
Deaths                      62 non-null int64
Confirmed                   62 non-null int64
dtypes: int64(4), object(5)
memory usage: 4.8+ KB
```

In [9]:
```
tn.isnull().sum()
```

Out[9]:
```
Sno                         0
Date                        0
Time                        0
State/UnionTerritory        0
ConfirmedIndianNational     0
ConfirmedForeignNational    0
Cured                       0
Deaths                      0
Confirmed                   0
dtype: int64
```

No NULL Values

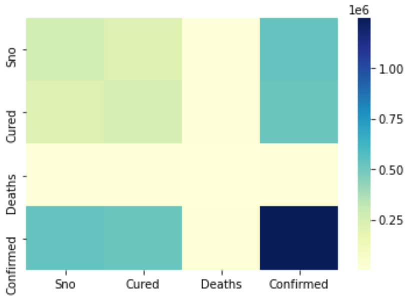
In [10]:
```
tn.describe()
```

Out[10]:

	Sno	Cured	Deaths	Confirmed
count	62.000000	62.000000	62.000000	62.000000
mean	769.322581	327.338710	9.741935	945.258065
std	513.685455	506.741825	11.011076	1117.872481
min	62.000000	0.000000	0.000000	1.000000
25%	291.750000	1.000000	0.000000	7.500000
50%	719.000000	8.000000	5.000000	596.000000
75%	1209.000000	590.500000	17.750000	1577.000000
max	1699.000000	1516.000000	35.000000	4829.000000

No peculiarities in data

```
In [11]: covMatrix = pd.DataFrame.cov(tn)
         sn.heatmap(covMatrix, annot=False, fmt='g',cmap="YlGnBu")
         plt.show()
```

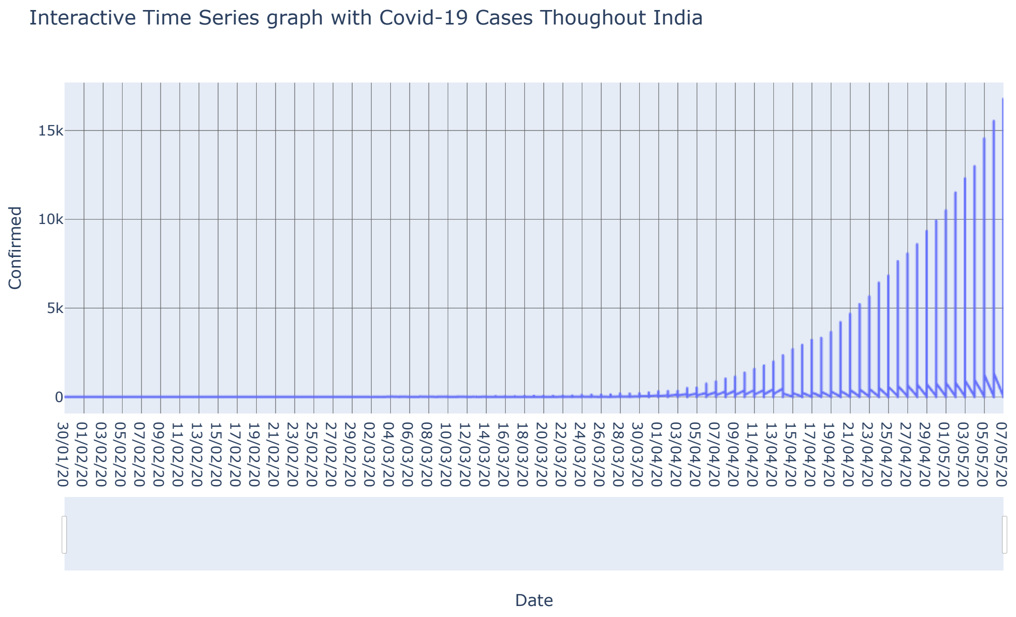


No Correlation between the attributes.

```
In [12]: df=pd.read_csv('covid19-in-india/covid_19_india.csv')
```

```
In [13]: fig = px.line(df, x='Date', y='Confirmed', title='Interactive Time Series graph with Covid-19 Cases Thoughout India',hover_name='State/UnionTe
         rritory')

         fig.update_xaxes(rangeslider_visible=True)
         fig.show()
```



Graph shows an increasing amount of Daily cases in India.

Dropping unwanted columns

```
In [14]: df=df.drop('Time',axis=1)
```

```
In [15]: df=df.drop('ConfirmedIndianNational',axis=1)
         df=df.drop('ConfirmedForeignNational',axis=1)
```

```
In [16]: df.columns
```

```
Out[16]: Index(['Sno', 'Date', 'State/UnionTerritory', 'Cured', 'Deaths', 'Confirmed'], dtype='object')
```

```
In [17]: df['State/UnionTerritory'].unique()
```

```
Out[17]: array(['Kerala', 'Telengana', 'Delhi', 'Rajasthan', 'Uttar Pradesh',
                'Haryana', 'Ladakh', 'Tamil Nadu', 'Karnataka', 'Maharashtra',
                'Punjab', 'Jammu and Kashmir', 'Andhra Pradesh', 'Uttarakhand',
                'Odisha', 'Puducherry', 'West Bengal', 'Chhattisgarh',
                'Chandigarh', 'Gujarat', 'Himachal Pradesh', 'Madhya Pradesh',
                'Bihar', 'Manipur', 'Mizoram', 'Andaman and Nicobar Islands',
                'Goa', 'Unassigned', 'Assam', 'Jharkhand', 'Arunachal Pradesh',
                'Tripura', 'Nagaland', 'Meghalaya', 'Nagaland#', 'Jharkhand#',
                'Dadar Nagar Haveli'], dtype=object)
```

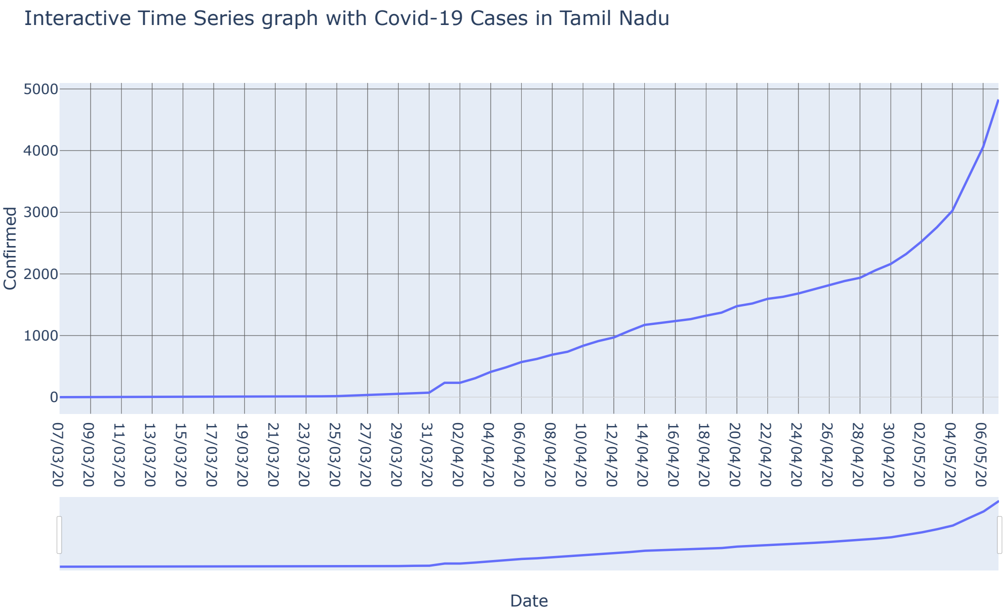
Selecting data from the state Tamil Nadu

```
In [18]: data=df[df['State/UnionTerritory']=='Tamil Nadu']
```

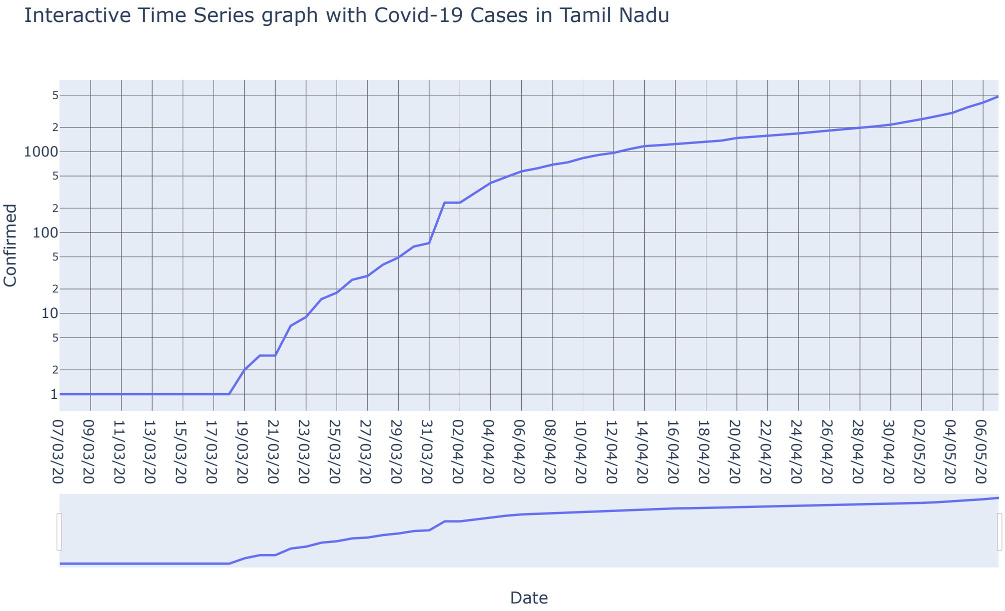
```
In [19]: data=data.drop('Sno',axis=1)
```

```
In [20]: fig = px.line(data, x='Date', y='Confirmed', title='Interactive Time Series graph with Covid-19 Cases in Tamil Nadu',hover_name='State/UnionTerritory')

         fig.update_xaxes(rangeslider_visible=True)
         fig.show()
```



```
In [21]: fig.update_layout( yaxis_type="log")
```



```
In [22]: data['State/UnionTerritory'].unique()
```

```
Out[22]: array(['Tamil Nadu'], dtype=object)
```

In [23]:
```
data.head()
```

Out[23]:

	Date	State/UnionTerritory	Cured	Deaths	Confirmed
61	07/03/20	Tamil Nadu	0	0	1
67	08/03/20	Tamil Nadu	0	0	1
79	09/03/20	Tamil Nadu	0	0	1
94	10/03/20	Tamil Nadu	0	0	1
105	11/03/20	Tamil Nadu	0	0	1

Creating new column such that the number of days from Mach 7th is displayed

In [24]:
```
data['Day']=np.arange(len(data))+1
```

In [25]:
```
data
```

Out[25]:

	Date	State/UnionTerritory	Cured	Deaths	Confirmed	Day
61	07/03/20	Tamil Nadu	0	0	1	1
67	08/03/20	Tamil Nadu	0	0	1	2
79	09/03/20	Tamil Nadu	0	0	1	3
94	10/03/20	Tamil Nadu	0	0	1	4
105	11/03/20	Tamil Nadu	0	0	1	5
...	...	...	...	...	...	...
1568	03/05/20	Tamil Nadu	1341	29	2757	58
1600	04/05/20	Tamil Nadu	1379	30	3023	59
1632	05/05/20	Tamil Nadu	1409	31	3550	60
1665	06/05/20	Tamil Nadu	1485	33	4058	61
1698	07/05/20	Tamil Nadu	1516	35	4829	62

62 rows × 6 columns

creating an separate column for Active Covid-19 cases

In [26]:
```
data.insert(6, "Active", 0, True)
```

In [27]:
```
data
```

Out[27]:

	Date	State/UnionTerritory	Cured	Deaths	Confirmed	Day	Active
61	07/03/20	Tamil Nadu	0	0	1	1	0
67	08/03/20	Tamil Nadu	0	0	1	2	0
79	09/03/20	Tamil Nadu	0	0	1	3	0
94	10/03/20	Tamil Nadu	0	0	1	4	0
105	11/03/20	Tamil Nadu	0	0	1	5	0
...	...	...	...	...	...	...	...
1568	03/05/20	Tamil Nadu	1341	29	2757	58	0
1600	04/05/20	Tamil Nadu	1379	30	3023	59	0
1632	05/05/20	Tamil Nadu	1409	31	3550	60	0
1665	06/05/20	Tamil Nadu	1485	33	4058	61	0
1698	07/05/20	Tamil Nadu	1516	35	4829	62	0

62 rows × 7 columns

In [28]:
```
count=0
lis=[]
data.reset_index(drop=True, inplace=True)
t1=data['Confirmed']
t2=data['Cured']
t3=data['Deaths']
```

In [29]:
```
for i in range(62):
    count=t1[i]-t2[i]-t3[i]
    lis.append(count)
```

In [30]:
```
lis
```

Out[30]:
```
[1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 2,
 2,
 6,
 8,
 14,
 17,
 24,
 27,
 37,
 44,
 62,
 69,
 227,
 227,
 302,
 403,
 476,
 558,
 608,
 664,
 709,
 805,
 859,
 915,
 1014,
 1104,
 1111,
 1110,
 1072,
 1025,
 992,
 1051,
 1046,
 943,
 949,
 911,
 867,
 838,
 841,
 812,
 865,
 925,
 1038,
 1186,
 1387,
 1614,
 2110,
 2540,
 3278]
```

In [31]:
```
data['Active'] = lis
```

In [32]:
```
data=data.drop('Date',axis=1)
```

In [33]:
```
data
```

Out[33]:

	State/UnionTerritory	Cured	Deaths	Confirmed	Day	Active
0	Tamil Nadu	0	0	1	1	1
1	Tamil Nadu	0	0	1	2	1
2	Tamil Nadu	0	0	1	3	1
3	Tamil Nadu	0	0	1	4	1
4	Tamil Nadu	0	0	1	5	1
...	...	...	...	...	...	...
57	Tamil Nadu	1341	29	2757	58	1387
58	Tamil Nadu	1379	30	3023	59	1614
59	Tamil Nadu	1409	31	3550	60	2110
60	Tamil Nadu	1485	33	4058	61	2540
61	Tamil Nadu	1516	35	4829	62	3278

62 rows × 6 columns

In [34]:
```
data.reset_index(drop=True, inplace=True)
```

In [35]:

```
data
```

Out[35]:

	State/UnionTerritory	Cured	Deaths	Confirmed	Day	Active
0	Tamil Nadu	0	0	1	1	1
1	Tamil Nadu	0	0	1	2	1
2	Tamil Nadu	0	0	1	3	1
3	Tamil Nadu	0	0	1	4	1
4	Tamil Nadu	0	0	1	5	1
...	...	...	...	...	...	...
57	Tamil Nadu	1341	29	2757	58	1387
58	Tamil Nadu	1379	30	3023	59	1614
59	Tamil Nadu	1409	31	3550	60	2110
60	Tamil Nadu	1485	33	4058	61	2540
61	Tamil Nadu	1516	35	4829	62	3278

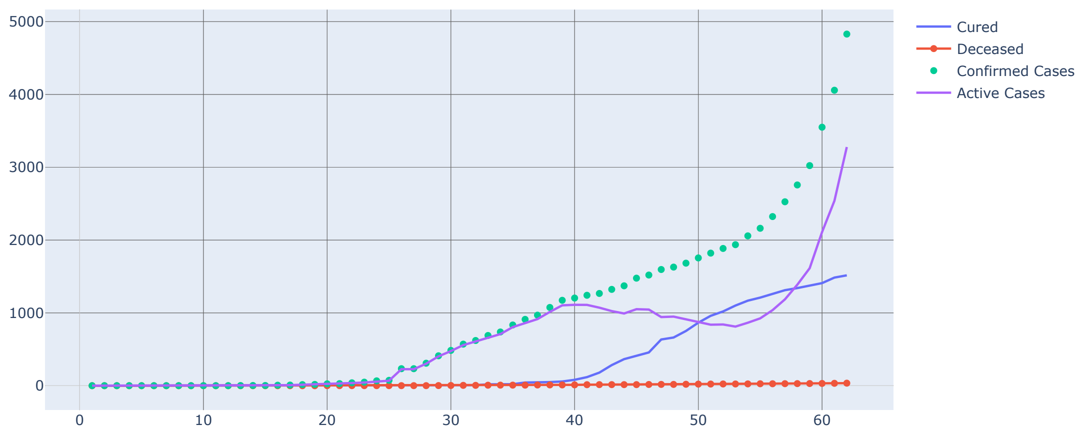
62 rows × 6 columns

Plotting the Cured, Deceased, Confirmed Covid-19 cases and Active Covid-19 cases

In [36]:

```
fig = go.Figure()
fig.add_trace(go.Scatter(x=data['Day'], y=data['Cured'],mode='lines',name='Cured'))
fig.add_trace(go.Scatter(x=data['Day'], y=data['Deaths'],mode='lines+markers',name='Deceased'))
fig.add_trace(go.Scatter(x=data['Day'], y=data['Confirmed'],mode='markers', name='Confirmed Cases'))
fig.add_trace(go.Scatter(x=data['Day'], y=data['Active'],mode='lines', name='Active Cases'))

fig.show()
```



Prediction of Confirmed Cases

In [37]:

```
dff=data.drop('Cured',axis=1)
dff=dff.drop('Deaths',axis=1)
dff=dff.drop('Active',axis=1)
```

In [38]:

```
f=dff['Day'].values
y=dff['Confirmed'].values
```

In [39]:

```
f1=np.array([i for i in f])
y1=np.array([i for i in y])
```

In [40]:

```
theta=0.4
n=len(f)
s=np.empty(n)
def yp(f,tau,theta,gam,b):
    z=tau/(1+np.exp(-gam*(f-theta)))+b

    return(z)
p0=[max(y1), np.median(f1),1,min(y1)]
print(yp(f1,max(y1), np.median(f1),1,min(y1)))
```

```
[1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000001e+00
 1.00000001e+00 1.00000004e+00 1.00000011e+00 1.00000030e+00
 1.00000082e+00 1.00000222e+00 1.00000604e+00 1.00001641e+00
 1.00004461e+00 1.00012126e+00 1.00032961e+00 1.00089597e+00
 1.00243549e+00 1.00662035e+00 1.01799594e+00 1.04891773e+00
 1.13296985e+00 1.36143243e+00 1.98234888e+00 3.66936804e+00
 8.24920912e+00 2.06546950e+01 5.40559460e+01 1.42548762e+02
 3.67319151e+02 8.81932854e+02 1.82414389e+03 3.00685611e+03
 3.94906715e+03 4.46368085e+03 4.68845124e+03 4.77694405e+03
 4.81034530e+03 4.82275079e+03 4.82733063e+03 4.82901765e+03
 4.82963857e+03 4.82986703e+03 4.82995108e+03 4.82998200e+03
 4.82999338e+03 4.82999756e+03 4.82999910e+03 4.82999967e+03
 4.82999988e+03 4.82999996e+03 4.82999998e+03 4.82999999e+03
 4.83000000e+03 4.83000000e+03 4.83000000e+03 4.83000000e+03
 4.83000000e+03 4.83000000e+03 4.83000000e+03 4.83000000e+03
 4.83000000e+03 4.83000000e+03]
```

In [41]:

```
from scipy.optimize import curve_fit
c,cov=curve_fit(yp,f1,y1,p0,method='dogbox', maxfev=6000)
```

In [42]:

```
yo=yp(f1,c[0],c[1],c[2],c[3])
```

In [43]:

```
from sklearn.metrics import r2_score
print(r2_score(yo,f1))
```

```
-0.9508575186522852
```

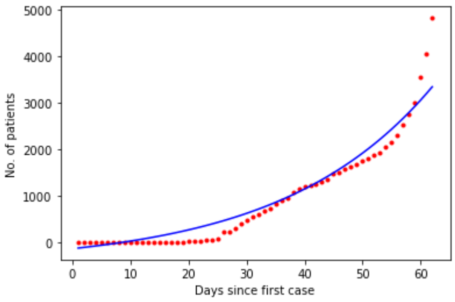
Plotting Result : Prediction of Confirmed Cases

In [44]:

```
plt.plot(f1,y1,'r.')
plt.plot(f1,yo,'b-')
plt.xlabel("Days since first case")
plt.ylabel("No. of patients")
```

Out[44]:

```
Text(0, 0.5, 'No. of patients')
```



Red dotted line denotes the actual data
Blue Line dentotes the predicted value

Prediction of the Deceased

In [45]:

```
df3=data.drop('Cured',axis=1)
df3=df3.drop('Deaths',axis=1)
df3=df3.drop('Confirmed',axis=1)
```

In [46]:

```
f2=df3['Day'].values
y2=df3['Active'].values
f21=np.array([i for i in f2])
y21=np.array([i for i in y2])
```

```
In [47]: theta=0.4
         n=len(f2)
         s2=np.empty(n)
         def yp2(f,tau,theta,gam,b):
             z=tau/(1+np.exp(-gam*(f-theta)))+b
             return(z)
         p0=[max(y21), np.median(f21),1,min(y21)]
         print(yp2(f21,max(y21), np.median(f21),1,min(y21)))
```

```
[1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00
 1.00000001e+00 1.00000003e+00 1.00000008e+00 1.00000020e+00
 1.00000055e+00 1.00000151e+00 1.00000410e+00 1.00001114e+00
 1.00003028e+00 1.00008231e+00 1.00022374e+00 1.00060820e+00
 1.00165325e+00 1.00449400e+00 1.01221592e+00 1.03320611e+00
 1.09026200e+00 1.24534593e+00 1.66683363e+00 2.81200837e+00
 5.92087544e+00 1.43419114e+01 3.70151979e+01 9.70854924e+01
 2.49663114e+02 5.98990867e+02 1.23857831e+03 2.04142169e+03
 2.68100913e+03 3.03033689e+03 3.18291451e+03 3.24298480e+03
 3.26565809e+03 3.27407912e+03 3.27718799e+03 3.27833317e+03
 3.27875465e+03 3.27890974e+03 3.27896679e+03 3.27898778e+03
 3.27899551e+03 3.27899835e+03 3.27899939e+03 3.27899978e+03
 3.27899992e+03 3.27899997e+03 3.27899999e+03 3.27900000e+03
 3.27900000e+03 3.27900000e+03 3.27900000e+03 3.27900000e+03
 3.27900000e+03 3.27900000e+03 3.27900000e+03 3.27900000e+03
 3.27900000e+03 3.27900000e+03]
```

```
In [48]: from scipy.optimize import curve_fit
         c2,cov2=curve_fit(yp2,f21,y21,p0,method='lm', maxfev=5000)
```

```
In [49]: yo2=yp2(f21,c2[0],c2[1],c2[2],c[3])
```

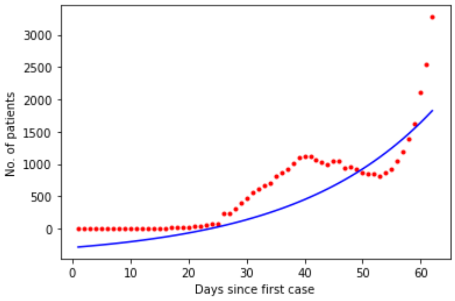
```
In [50]: from sklearn.metrics import r2_score
         print(r2_score(yo2,y21))
```

```
0.5878413765131831
```

Plotting Result : Prediction of the Deceased

```
In [51]: plt.plot(f21,y21,'r.')
         plt.plot(f21,yo2,'b-')
         plt.xlabel("Days since first case")
         plt.ylabel("No. of patients")
```

```
Out[51]: Text(0, 0.5, 'No. of patients')
```



The predicted of deaths is low due to the data being ununiform (no pattern) and high variation in the death data.

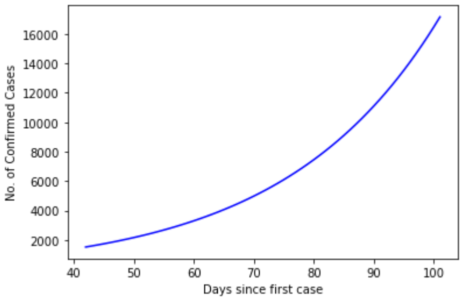
Forecasting Covid-19 Cases in Tamil Nadu

```
In [52]: fut=np.arange(42, 102)
```

```
In [53]: yo2_pred=yp(fut,c[0],c[1],c[2],c2[3])
```

```
In [54]: plt.plot(fut,yo2_pred,'b-')
         plt.xlabel("Days since first case")
         plt.ylabel("No. of Confirmed Cases")
```

```
Out[54]: Text(0, 0.5, 'No. of Confirmed Cases')
```



The forecasting shows that the number of confirmed cases in Tamil Nadu will increase to about 7,750 Covid-19 cased by June 2020

Tamil Nadu Testing analysis

```
In [55]: test=pd.read_csv('covid19-in-india/StatewiseTestingDetails.csv')
```

```
In [56]: test
```

Out[56]:

	Date	State	TotalSamples	Negative	Positive
0	2020-04-17	Andaman and Nicobar Islands	1403.0	1210.0	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	2020-04-02	Andhra Pradesh	1800.0	1175.0	132.0
...	...	...	...	...	...
754	2020-04-30	West Bengal	16525.0	NaN	758.0
755	2020-05-01	West Bengal	18566.0	NaN	NaN
756	2020-05-02	West Bengal	20976.0	NaN	795.0
757	2020-05-03	West Bengal	22915.0	NaN	922.0
758	2020-05-04	West Bengal	25116.0	NaN	1259.0

759 rows × 5 columns

```
In [57]: tn=test[test['State']=='Tamil Nadu']
```

```
In [58]: tn.reset_index(drop=True, inplace=True)
```

```
In [59]: tn
```

Out[59]:

	Date	State	TotalSamples	Negative	Positive
0	2020-04-03	Tamil Nadu	3684.0	2789.0	411.0
1	2020-04-08	Tamil Nadu	5305.0	4414.0	690.0
2	2020-04-09	Tamil Nadu	7267.0	5824.0	834.0
3	2020-04-10	Tamil Nadu	8410.0	6838.0	911.0
4	2020-04-11	Tamil Nadu	9842.0	7779.0	969.0
5	2020-04-12	Tamil Nadu	10655.0	NaN	1075.0
6	2020-04-13	Tamil Nadu	12746.0	NaN	1173.0
7	2020-04-14	Tamil Nadu	19255.0	13234.0	1204.0
8	2020-04-15	Tamil Nadu	21994.0	15210.0	1242.0
9	2020-04-16	Tamil Nadu	26005.0	18743.0	1267.0
10	2020-04-17	Tamil Nadu	29673.0	21628.0	1323.0
11	2020-04-18	Tamil Nadu	35036.0	27192.0	1372.0
12	2020-04-19	Tamil Nadu	40876.0	31853.0	1477.0
13	2020-04-20	Tamil Nadu	46985.0	38082.0	1520.0
14	2020-04-21	Tamil Nadu	53045.0	43582.0	1596.0
15	2020-04-22	Tamil Nadu	59023.0	49506.0	1629.0
16	2020-04-23	Tamil Nadu	65977.0	56836.0	1683.0
17	2020-04-24	Tamil Nadu	72403.0	62596.0	1755.0
18	2020-04-25	Tamil Nadu	80110.0	69390.0	1821.0
19	2020-04-26	Tamil Nadu	87605.0	77133.0	1885.0
20	2020-04-27	Tamil Nadu	94781.0	83021.0	1937.0
21	2020-04-28	Tamil Nadu	101874.0	97908.0	2058.0
22	2020-04-29	Tamil Nadu	109961.0	105864.0	2162.0
23	2020-04-30	Tamil Nadu	119748.0	115761.0	2323.0
24	2020-05-01	Tamil Nadu	129363.0	124852.0	2526.0
25	2020-05-02	Tamil Nadu	139490.0	135698.0	2757.0
26	2020-05-03	Tamil Nadu	150107.0	145520.0	3023.0
27	2020-05-04	Tamil Nadu	162970.0	158558.0	3550.0

```
In [60]: tn.describe()
```

Out[60]:

	TotalSamples	Negative	Positive
count	28.000000	26.000000	28.000000
mean	60863.928571	58454.269231	1649.035714
std	49213.379674	48755.330025	719.709174
min	3684.000000	2789.000000	411.000000
25%	17627.750000	16093.250000	1196.250000
50%	50015.000000	46544.000000	1558.000000
75%	96554.250000	94186.250000	1967.250000
max	162970.000000	158558.000000	3550.000000

In [61]:
```
tn.isna().sum()
```
Out[61]:
```
Date            0
State           0
TotalSamples    0
Negative        2
Positive        0
dtype: int64
```

In [62]:
```
tn=tn.fillna(tn.mean())
```

In [63]:
```
tn
```
Out[63]:

	Date	State	TotalSamples	Negative	Positive
0	2020-04-03	Tamil Nadu	3684.0	2789.000000	411.0
1	2020-04-08	Tamil Nadu	5305.0	4414.000000	690.0
2	2020-04-09	Tamil Nadu	7267.0	5824.000000	834.0
3	2020-04-10	Tamil Nadu	8410.0	6838.000000	911.0
4	2020-04-11	Tamil Nadu	9842.0	7779.000000	969.0
5	2020-04-12	Tamil Nadu	10655.0	58454.269231	1075.0
6	2020-04-13	Tamil Nadu	12746.0	58454.269231	1173.0
7	2020-04-14	Tamil Nadu	19255.0	13234.000000	1204.0
8	2020-04-15	Tamil Nadu	21994.0	15210.000000	1242.0
9	2020-04-16	Tamil Nadu	26005.0	18743.000000	1267.0
10	2020-04-17	Tamil Nadu	29673.0	21628.000000	1323.0
11	2020-04-18	Tamil Nadu	35036.0	27192.000000	1372.0
12	2020-04-19	Tamil Nadu	40876.0	31853.000000	1477.0
13	2020-04-20	Tamil Nadu	46985.0	38082.000000	1520.0
14	2020-04-21	Tamil Nadu	53045.0	43582.000000	1596.0
15	2020-04-22	Tamil Nadu	59023.0	49506.000000	1629.0
16	2020-04-23	Tamil Nadu	65977.0	56836.000000	1683.0
17	2020-04-24	Tamil Nadu	72403.0	62596.000000	1755.0
18	2020-04-25	Tamil Nadu	80110.0	69390.000000	1821.0
19	2020-04-26	Tamil Nadu	87605.0	77133.000000	1885.0
20	2020-04-27	Tamil Nadu	94781.0	83021.000000	1937.0
21	2020-04-28	Tamil Nadu	101874.0	97908.000000	2058.0
22	2020-04-29	Tamil Nadu	109961.0	105864.000000	2162.0
23	2020-04-30	Tamil Nadu	119748.0	115761.000000	2323.0
24	2020-05-01	Tamil Nadu	129363.0	124852.000000	2526.0
25	2020-05-02	Tamil Nadu	139490.0	135698.000000	2757.0
26	2020-05-03	Tamil Nadu	150107.0	145520.000000	3023.0
27	2020-05-04	Tamil Nadu	162970.0	158558.000000	3550.0

Total number of tests done in Tamil Nadu

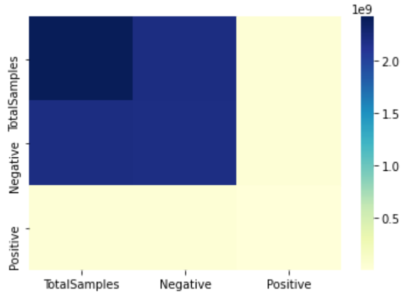
In [64]:
```
tn_testing=tn.iloc[27, 2]
tn_testing
```
Out[64]:
```
162970.0
```

Total number positive test in Tamil Nadu

In [65]:
```
tn_positive = tn.iloc[27,4]
tn_positive
```
Out[65]:
```
3550.0
```

Finding Relation between the attributes

In [66]:
```
covMatrix = pd.DataFrame.cov(tn)
sn.heatmap(covMatrix, annot=False, fmt='g',cmap="YlGnBu")
plt.show()
```

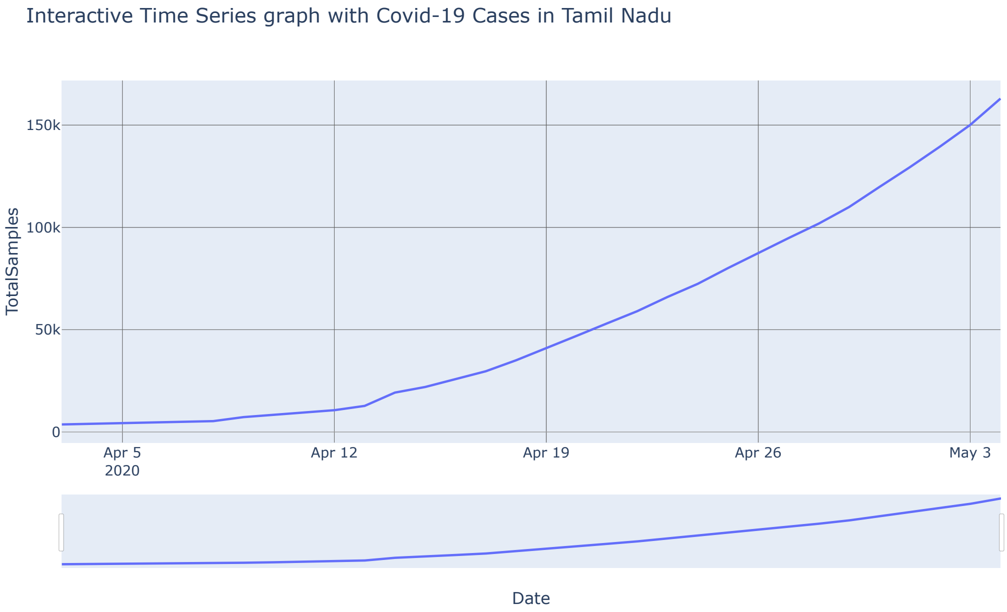


Cumulative Time Series graph on the testing done in Tamil Nadu

In [67]:

```
fig = px.line(tn, x='Date', y='TotalSamples', title='Interactive Time Series graph with Covid-19 Cases in Tamil Nadu',hover_name='TotalSamples')

fig.update_xaxes(rangeslider_visible=True)
fig.show()
```

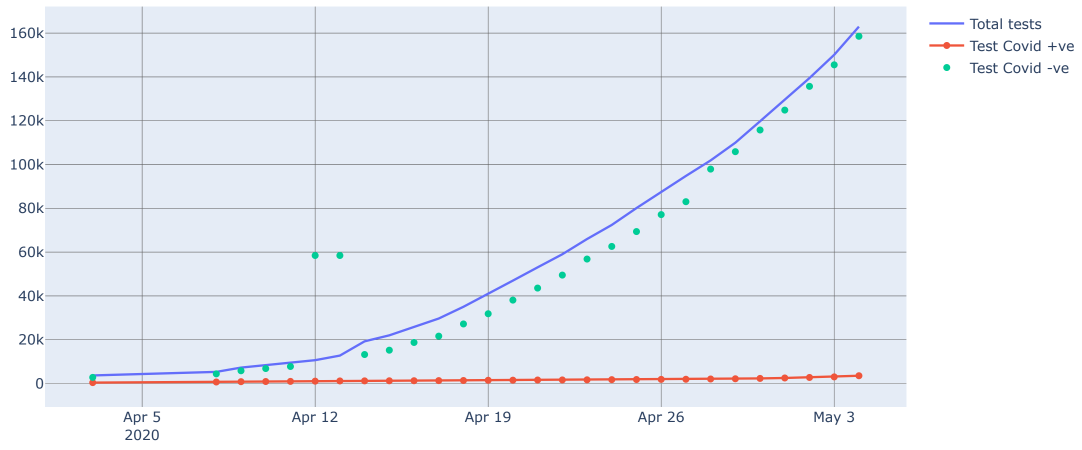


Plotting of the Total cases, +ve covid-19 cases and -ve covid-19 cases

In [68]:

```
fig = go.Figure()
fig.add_trace(go.Scatter(x=tn['Date'], y=tn['TotalSamples'],mode='lines',name='Total tests'))
fig.add_trace(go.Scatter(x=tn['Date'], y=tn['Positive'],mode='lines+markers',name='Test Covid +ve'))
fig.add_trace(go.Scatter(x=tn['Date'], y=tn['Negative'],mode='markers', name='Test Covid -ve'))


fig.show()
```



This tells us that more number of samples, more the chances of the test being negative.

In [69]:

```
pop=pd.read_csv('covid19-in-india/population_india_census2011.csv')
```

```
In [70]: pop
```

Out[70]:

	Sno	State / Union Territory	Population	Rural population	Urban population	Area	Density	Gender Ratio
0	1	Uttar Pradesh	199812341	155317278	44495063	240,928 km2 (93,023 sq mi)	828/km2 (2,140/sq mi)	912
1	2	Maharashtra	112374333	61556074	50818259	307,713 km2 (118,809 sq mi)	365/km2 (950/sq mi)	929
2	3	Bihar	104099452	92341436	11758016	94,163 km2 (36,357 sq mi)	1,102/km2 (2,850/sq mi)	918
3	4	West Bengal	91276115	62183113	29093002	88,752 km2 (34,267 sq mi)	1,029/km2 (2,670/sq mi)	953
4	5	Madhya Pradesh	72626809	52557404	20069405	308,245 km2 (119,014 sq mi)	236/km2 (610/sq mi)	931
5	6	Tamil Nadu	72147030	37229590	34917440	130,058 km2 (50,216 sq mi)	555/km2 (1,440/sq mi)	996
6	7	Rajasthan	68548437	51500352	17048085	342,239 km2 (132,139 sq mi)	201/km2 (520/sq mi)	928
7	8	Karnataka	61095297	37469335	23625962	191,791 km2 (74,051 sq mi)	319/km2 (830/sq mi)	973
8	9	Gujarat	60439692	34694609	25745083	196,024 km2 (75,685 sq mi)	308/km2 (800/sq mi)	919
9	10	Andhra Pradesh	49577103	34966693	14610410	162,968 km2 (62,922 sq mi)	303/km2 (780/sq mi)	993
10	11	Odisha	41974218	34970562	7003656	155,707 km2 (60,119 sq mi)	269/km2 (700/sq mi)	979
11	12	Telengana	35003674	21395009	13608665	112,077 km2 (43,273 sq mi)	312/km2 (810/sq mi)	988
12	13	Kerala	33406061	17471135	15934926	38,863 km2 (15,005 sq mi)	859/km2 (2,220/sq mi)	1084
13	14	Jharkhand	32988134	25055073	7933061	79,714 km2 (30,778 sq mi)	414/km2 (1,070/sq mi)	948
14	15	Assam	31205576	26807034	4398542	78,438 km2 (30,285 sq mi)	397/km2 (1,030/sq mi)	954
15	16	Punjab	27743338	17344192	10399146	50,362 km2 (19,445 sq mi)	550/km2 (1,400/sq mi)	895
16	17	Chhattisgarh	25545198	19607961	5937237	135,191 km2 (52,198 sq mi)	189/km2 (490/sq mi)	991
17	18	Haryana	25351462	16509359	8842103	44,212 km2 (17,070 sq mi)	573/km2 (1,480/sq mi)	879
18	19	Uttarakhand	10086292	7036954	3049338	53,483 km2 (20,650 sq mi)	189/km2 (490/sq mi)	963
19	20	Himachal Pradesh	6864602	6176050	688552	55,673 km2 (21,495 sq mi)	123/km2 (320/sq mi)	972
20	21	Tripura	3673917	2712464	961453	10,486 km2 (4,049 sq mi)	350/km2 (910/sq mi)	960
21	22	Meghalaya	2966889	2371439	595450	22,429 km2 (8,660 sq mi)	132/km2 (340/sq mi)	989
22	23	Manipur	2570390	1793875	776515	22,327 km2 (8,621 sq mi)	122/km2 (320/sq mi)	992
23	24	Nagaland	1978502	1407536	570966	16,579 km2 (6,401 sq mi)	119/km2 (310/sq mi)	931
24	25	Goa	1458545	551731	906814	3,702 km2 (1,429 sq mi)	394/km2 (1,020/sq mi)	973
25	26	Arunachal Pradesh	1383727	1066358	317369	83,743 km2 (32,333 sq mi)	17/km2 (44/sq mi)	938
26	27	Mizoram	1097206	525435	571771	21,081 km2 (8,139 sq mi)	52/km2 (130/sq mi)	976
27	28	Sikkim	610577	456999	153578	7,096 km2 (2,740 sq mi)	86/km2 (220/sq mi)	890
28	29	Delhi	16787941	419042	16368899	1,484 km2 (573 sq mi)	11,297/km2 (29,260/sq mi)	868
29	30	Jammu and Kashmir	12267032	9064220	3202812	125,535 km2 (48,469 sq mi)	98/km2 (250/sq mi)	890
30	31	Puducherry	1247953	395200	852753	479 km2 (185 sq mi)	2,598/km2 (6,730/sq mi)	1037
31	32	Chandigarh	1055450	28991	1026459	114 km2 (44 sq mi)	9,252/km2 (23,960/sq mi)	818
32	33	Dadra and Nagar Haveli and Daman and Diu	585764	243510	342254	603 km2 (233 sq mi)	970/km2 (2,500/sq mi)	711
33	34	Andaman and Nicobar Islands	380581	237093	143488	8,249 km2 (3,185 sq mi)	46/km2 (120/sq mi)	876
34	35	Ladakh	274000	43840	230160	96,701 km2 (37,336 sq mi)	2.8/km2 (7.3/sq mi)	853
35	36	Lakshadweep	64473	14141	50332	32 km2 (12 sq mi)	2,013/km2 (5,210/sq mi)	946

```
In [71]: pop.describe()
```

Out[71]:

	Sno	Population	Rural population	Urban population	Gender Ratio
count	36.000000	3.600000e+01	3.600000e+01	3.600000e+01	36.000000
mean	18.500000	3.362689e+07	2.315336e+07	1.047353e+07	937.583333
std	10.535654	4.305758e+07	3.212429e+07	1.312631e+07	65.544478
min	1.000000	6.447300e+04	1.414100e+04	5.033200e+04	711.000000
25%	9.750000	1.439840e+06	5.451570e+05	6.652765e+05	907.750000
50%	18.500000	2.106970e+07	1.278679e+07	5.167890e+06	947.000000
75%	27.250000	5.229275e+07	3.496766e+07	1.604342e+07	976.750000
max	36.000000	1.998123e+08	1.553173e+08	5.081826e+07	1084.000000

```
In [72]: pop=pop[pop['State / Union Territory']=='Tamil Nadu']
```

```
In [73]: pop.reset_index(drop=True, inplace=True)
```

```
In [74]: pop
```

Out[74]:

	Sno	State / Union Territory	Population	Rural population	Urban population	Area	Density	Gender Ratio
0	6	Tamil Nadu	72147030	37229590	34917440	130,058 km2 (50,216 sq mi)	555/km2 (1,440/sq mi)	996

```
In [75]: tn_population=pop.iloc[0, 2]
```

```
In [76]: tn_population
```

```
Out[76]: 72147030
```

```
In [77]: tn_testing
```

```
Out[77]: 162970.0
```

Population of Tamil Nadu : 72,147,030

Total testing by May 4th : 162,970

```
In [78]: test_rate = (tn_testing/tn_population)
         test_rate_per_million = (tn_testing/tn_population)*1000000
         test_rate_per_thousand= (tn_testing/tn_population)*1000
```

```
In [79]: test_rate
```

```
Out[79]: 0.002258859443001327
```

```
In [80]: test_rate_per_million
```

```
Out[80]: 2258.859443001327
```

```
In [81]: test_rate_per_thousand
```

```
Out[81]: 2.258859443001327
```

```
In [82]: Test_positive_rate = (tn_positive/tn_testing)*100
```

```
In [83]: Test_positive_rate
```

```
Out[83]: 2.178315027305639
```

Only 2.18 % of the tests are turned out to be positive

Tamil Nadu Testing Prediction

```
In [84]: tn.columns
```

```
Out[84]: Index(['Date', 'State', 'TotalSamples', 'Negative', 'Positive'], dtype='object')
```

```
In [85]: tn['Day']=np.arange(len(tn))+1
```

```
In [86]: dff=tn.drop('State',axis=1)
         dff=dff.drop('Negative',axis=1)
         dff=dff.drop('Positive',axis=1)
         dff=dff.drop('Date',axis=1)
```

```
In [87]: f2=dff['Day'].values
         y2=dff['TotalSamples'].values
         f21=np.array([i for i in f2])
         y21=np.array([i for i in y2])
```

```
In [88]: theta=0.4
         n=len(f2)
         s2=np.empty(n)
         def yp2(f,tau,theta,gam,b):
             z=tau/(1+np.exp(-gam*(f-theta)))+b
             return(z)
         p0=[max(y21), np.median(f21),1,min(y21)]
         print(yp2(f21,max(y21), np.median(f21),1,min(y21)))
```

```
[  3684.2234249    3684.6073304    3685.65088463   3688.48749158
   3696.1976897    3717.15249461   3774.08633446   3928.64767238
   4347.31034356   5474.54204051   8461.01424555  16046.60759806
  33413.88761472  65211.80279403 105126.19720597 136924.11238528
 154291.39240194 161876.98575445 164863.45795949 165990.68965644
 166409.35232762 166563.91366554 166620.84750539 166641.8023103
 166649.51250842 166652.34911537 166653.3926696  166653.7765751 ]
```

```
In [89]: from scipy.optimize import curve_fit
         c2,cov2=curve_fit(yp2,f21,y21,p0,method='dogbox', maxfev=2000)
```

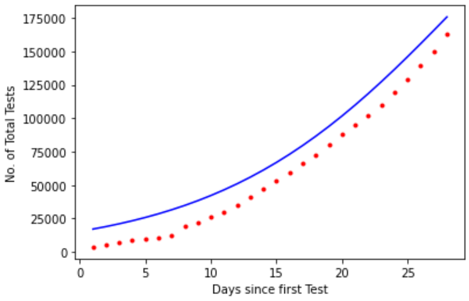
```
In [90]: yo2=yp2(f21,c2[0],c2[1],c2[2],c[3])
```

```
In [91]: from sklearn.metrics import r2_score
         print(r2_score(yo2,y21))
```

```
0.8983137329480259
```

```
In [92]: plt.plot(f21,y21,'r.')
         plt.plot(f21,yo2,'b-')
         plt.xlabel("Days since first Test")
         plt.ylabel("No. of Total Tests")
```

```
Out[92]: Text(0, 0.5, 'No. of Total Tests')
```



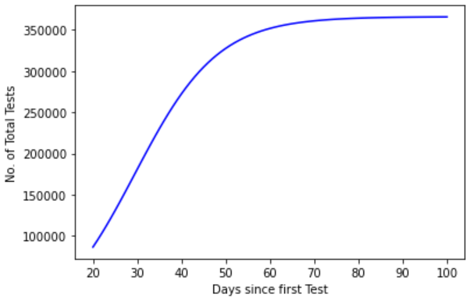
Tamil Nadu Testing Forecasting

```
In [93]: fut=np.arange(20, 101)
```

```
In [94]: yo_pred=yp(fut,c2[0],c2[1],c2[2],c2[3])
```

```
In [95]: plt.plot(fut,yo_pred,'b-')
         plt.xlabel("Days since first Test")
         plt.ylabel("No. of Total Tests")
```

```
Out[95]: Text(0, 0.5, 'No. of Total Tests')
```



From the above forecasting, a total number of 3 lakh tests are expected to be performed by the end of May-2020

```
In [96]: indi=pd.read_csv('covid19-in-india/IndividualDetails.csv')
```

```
In [97]: indi
```

Out[97]:

	id	government_id	diagnosed_date	age	gender	detected_city	detected_district	detected_state	nationality	current_status	status_change_date	notes
0	0	KL-TS-P1	30/01/2020	20	F	Thrissur	Thrissur	Kerala	India	Recovered	14/02/2020	Travelled from Wuhan
1	1	KL-AL-P1	02/02/2020	NaN	NaN	Alappuzha	Alappuzha	Kerala	India	Recovered	14/02/2020	Travelled from Wuhan
2	2	KL-KS-P1	03/02/2020	NaN	NaN	Kasaragod	Kasaragod	Kerala	India	Recovered	14/02/2020	Travelled from Wuhan
3	3	DL-P1	02/03/2020	45	M	East Delhi (Mayur Vihar)	East Delhi	Delhi	India	Recovered	15/03/2020	Travelled from Austria, Italy
4	4	TS-P1	02/03/2020	24	M	Hyderabad	Hyderabad	Telangana	India	Recovered	02/03/2020	Travelled from Dubai to Bangalore on 20th Feb,...
...	...	...	...	...	...	...	...	...	...	...	...	...
27885	27886	NaN	26/04/2020	NaN	NaN	NaN	NaN	Delhi	NaN	Hospitalized	26/04/2020	Details awaited
27886	27887	NaN	26/04/2020	NaN	NaN	NaN	NaN	Uttarakhand	NaN	Hospitalized	26/04/2020	Details awaited
27887	27888	NaN	26/04/2020	12	F	NaN	Munger	Bihar	NaN	Hospitalized	26/04/2020	Details awaited
27888	27889	NaN	26/04/2020	13	F	NaN	Munger	Bihar	NaN	Hospitalized	26/04/2020	Details awaited
27889	27890	NaN	26/04/2020	17	F	NaN	Munger	Bihar	NaN	Hospitalized	26/04/2020	Details awaited

27890 rows × 12 columns

```
In [98]: ind=indi[indi['detected_state']=='Tamil Nadu']
```

In [99]:

```
ind.describe(include="all")
```

Out[99]:

	id	government_id	diagnosed_date	age	gender	detected_city	detected_district	detected_state	nationality	current_status	status_change_date	notes
count	1885.000000	1885	1885	420	1013	103	1885	1885	60	1885	1885	1882
unique	NaN	1885	40	80	2	35	36	1	4	3	39	56
top	NaN	TN-P1476	01/04/2020	25	M	Mettupalayam	Chennai	Tamil Nadu	India	Hospitalized	01/04/2020	Details awaited
freq	NaN	1	110	16	714	20	528	1885	52	1878	110	1265
mean	10133.529973	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
std	7701.043559	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
min	33.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
25%	3439.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
50%	7968.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
75%	16189.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
max	26593.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [100]:

```
ind['detected_district'].unique()
```

Out[100]:

```
array(['Kancheepuram', 'Chennai', 'Erode', 'Coimbatore', 'Tirunelveli',
       'Tiruppur', 'Madurai', 'Chengalpattu', 'Salem', 'Ranipet',
       'Thanjavur', 'Vellore', 'Virudhunagar', 'Karur', 'Tiruvannamalai',
       'Viluppuram', 'Namakkal', 'Kanyakumari', 'Thoothukkudi', 'Theni',
       'Dindigul', 'Sivaganga', 'Tirupathur', 'Thiruvarur',
       'Ramanathapuram', 'Thiruvallur', 'Nagapattinam', 'Cuddalore',
       'Kallakurichi', 'Perambalur', 'Tiruchirappalli', 'Nilgiris',
       'Ariyalur', 'Tenkasi', 'Pudukkottai', 'Dharmapuri'], dtype=object)
```

In [101]:

```
ind.isnull().sum()
```

Out[101]:

```
id                       0
government_id            0
diagnosed_date           0
age                   1465
gender                 872
detected_city         1782
detected_district        0
detected_state           0
nationality           1825
current_status           0
status_change_date       0
notes                    3
dtype: int64
```

There are missing values in of age, gender, city, and nationality

But all the districts are recorded

In [102]:

```
df=ind['detected_district'].value_counts()
```

In [103]:

```
type(df)
```

Out[103]:

```
pandas.core.series.Series
```

In [104]:

```
df.unique()
```

Out[104]:

```
array([528, 141, 112,  79,  75,  70,  65,  59,  55,  54,  51,  44,  41,
        38,  35,  32,  31,  30,  27,  26,  23,  18,  16,  15,  12,  11,
         9,   7,   6,   5,   1], dtype=int64)
```

In [105]:

```
count=0
lis=[]
for i in range(36):
    count=df[i]
    lis.append(count)
```

In [106]:
```
lis.append(0)
lis
```

Out[106]:
```
[528,
 141,
 112,
 79,
 75,
 70,
 65,
 59,
 55,
 55,
 54,
 51,
 51,
 44,
 44,
 41,
 38,
 35,
 32,
 31,
 30,
 27,
 26,
 23,
 18,
 18,
 16,
 15,
 12,
 11,
 9,
 7,
 6,
 5,
 1,
 1,
 0]
```

In [107]:
```
ll=['Chennai','Coimbatore','Tiruppur','Dindigul','Madurai','Erode','Tirunelveli','Namakkal','Thanjavur','Chengalpattu','Thiruvallur','Viluppur
am','Tiruchirappalli','Nagappattinam','Theni','Karur','Ranipet','Tenkasi','Virudunagar','Salem','Thiruvarur','Thoothukkudi','Cuddalore','Vello
re','Kancheepuram','Tirupathur','Kanniyakumari','Ramanathapuram','Sivaganga','Tiruvannamalai','The Nilgiris','Perambalur','Kallakurichi','Ariy
alur','Pudukkottai','Dharmapuri','Krishnagiri']
```

In [108]:
```
df = pd.DataFrame(list(zip(ll, lis)),columns =['Districts', 'Count'])
df
```

Out[108]:

	Districts	Count
0	Chennai	528
1	Coimbatore	141
2	Tiruppur	112
3	Dindigul	79
4	Madurai	75
5	Erode	70
6	Tirunelveli	65
7	Namakkal	59
8	Thanjavur	55
9	Chengalpattu	55
10	Thiruvallur	54
11	Viluppuram	51
12	Tiruchirappalli	51
13	Nagappattinam	44
14	Theni	44
15	Karur	41
16	Ranipet	38
17	Tenkasi	35
18	Virudunagar	32
19	Salem	31
20	Thiruvarur	30
21	Thoothukkudi	27
22	Cuddalore	26
23	Vellore	23
24	Kancheepuram	18
25	Tirupathur	18
26	Kanniyakumari	16
27	Ramanathapuram	15
28	Sivaganga	12
29	Tiruvannamalai	11
30	The Nilgiris	9
31	Perambalur	7
32	Kallakurichi	6
33	Ariyalur	5
34	Pudukkottai	1
35	Dharmapuri	1
36	Krishnagiri	0

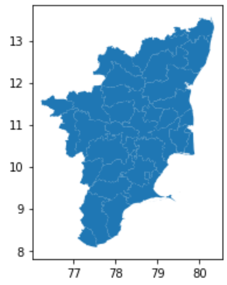
In [109]:
```
import geopandas as gpd
fp = "covid19-in-india/gadm36_IND_2.shp"
map_df = gpd.read_file(fp)

map_df = map_df[['NAME_1', 'NAME_2', 'geometry']]
map_df = map_df[map_df['NAME_1']=='Tamil Nadu']
map_df.plot()


map_df.head()
```

Out[109]:

	NAME_1	NAME_2	geometry
508	Tamil Nadu	Ariyalur	POLYGON ((79.25848 10.97425, 79.25031 10.96728...
509	Tamil Nadu	Chennai	POLYGON ((80.23345 12.96909, 80.22396 12.95902...
510	Tamil Nadu	Coimbatore	POLYGON ((77.15656 10.78965, 77.15119 10.78552...
511	Tamil Nadu	Cuddalore	POLYGON ((79.63276 11.23497, 79.62713 11.23186...
512	Tamil Nadu	Dharmapuri	POLYGON ((78.70405 11.96791, 78.69788 11.94817...

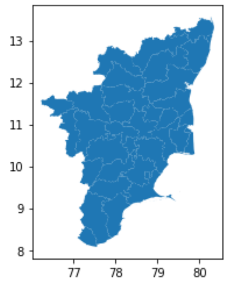


In [110]:

```
map_df = map_df[['NAME_2', 'geometry']]
map_df.plot()
```

Out[110]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x28ca66f4f98>
```



In [111]:

```
district_wise = df[['Districts', 'Count']]
district_wise
```

Out[111]:

	Districts	Count
0	Chennai	528
1	Coimbatore	141
2	Tiruppur	112
3	Dindigul	79
4	Madurai	75
5	Erode	70
6	Tirunelveli	65
7	Namakkal	59
8	Thanjavur	55
9	Chengalpattu	55
10	Thiruvallur	54
11	Viluppuram	51
12	Tiruchirappalli	51
13	Nagappattinam	44
14	Theni	44
15	Karur	41
16	Ranipet	38
17	Tenkasi	35
18	Virudunagar	32
19	Salem	31
20	Thiruvarur	30
21	Thoothukkudi	27
22	Cuddalore	26
23	Vellore	23
24	Kancheepuram	18
25	Tirupathur	18
26	Kanniyakumari	16
27	Ramanathapuram	15
28	Sivaganga	12
29	Tiruvannamalai	11
30	The Nilgiris	9
31	Perambalur	7
32	Kallakurichi	6
33	Ariyalur	5
34	Pudukkottai	1
35	Dharmapuri	1
36	Krishnagiri	0

In [112]:
```
district_wise = df[['Districts', 'Count']]
merged = map_df.set_index('NAME_2').join(district_wise.set_index('Districts'))

merged
```

Out[112]:

NAME_2	geometry	Count
Ariyalur	POLYGON ((79.25848 10.97425, 79.25031 10.96728...	5
Chennai	POLYGON ((80.23345 12.96909, 80.22396 12.95902...	528
Coimbatore	POLYGON ((77.15656 10.78965, 77.15119 10.78552...	141
Cuddalore	POLYGON ((79.63276 11.23497, 79.62713 11.23186...	26
Dharmapuri	POLYGON ((78.70405 11.96791, 78.69788 11.94817...	1
Dindigul	POLYGON ((78.15361 10.08057, 78.15300 10.08310...	79
Erode	POLYGON ((77.84909 11.23407, 77.85358 11.22551...	70
Kancheepuram	MULTIPOLYGON (((79.96208 12.22216, 79.96208 12...	18
Kanniyakumari	MULTIPOLYGON (((77.55596 8.07903, 77.55596 8.0...	16
Karur	POLYGON ((78.49814 10.68002, 78.48006 10.68318...	41
Krishnagiri	POLYGON ((78.63942 12.24634, 78.64082 12.23947...	0
Madurai	POLYGON ((78.06779 9.74771, 78.06450 9.73725, ...	75
Nagappattinam	MULTIPOLYGON (((79.74597 10.27736, 79.74541 10...	44
Namakkal	POLYGON ((78.43343 11.33039, 78.43341 11.32425...	59
Perambalur	POLYGON ((78.99304 11.04352, 78.98911 11.04572...	7
Pudukkottai	POLYGON ((78.95749 9.98304, 78.95445 9.98641, ...	1
Ramanathapuram	MULTIPOLYGON (((79.53009 9.38596, 79.53014 9.3...	15
Salem	POLYGON ((78.22656 11.90686, 78.22643 11.90152...	31
Sivaganga	POLYGON ((78.70577 9.69627, 78.72670 9.69614, ...	12
Thanjavur	MULTIPOLYGON (((79.51510 10.32097, 79.51486 10...	55
The Nilgiris	POLYGON ((76.85727 11.34142, 76.85609 11.33792...	9
Theni	POLYGON ((77.34090 9.59505, 77.32851 9.57323, ...	44
Thiruvallur	MULTIPOLYGON (((80.02853 13.12436, 80.02964 13...	54
Thiruvarur	MULTIPOLYGON (((79.62569 10.34375, 79.62569 10...	30
Thoothukkudi	MULTIPOLYGON (((78.11542 8.61681, 78.11542 8.6...	27
Tiruchirappalli	POLYGON ((78.36396 10.29705, 78.35506 10.29553...	51
Tirunelveli	MULTIPOLYGON (((77.96347 8.32891, 77.96347 8.3...	65
Tiruppur	POLYGON ((77.43851 11.11145, 77.44918 11.10544...	112
Tiruvannamalai	POLYGON ((79.50778 12.34159, 79.50247 12.33965...	11
Vellore	POLYGON ((78.84798 12.62251, 78.84678 12.61580...	23
Viluppuram	MULTIPOLYGON (((79.94625 12.21915, 79.94625 12...	51
Virudunagar	POLYGON ((77.96670 9.24325, 77.95448 9.23935, ...	32

In [113]:
```
merged.isna().sum()
```

Out[113]:
```
geometry    0
Count       0
dtype: int64
```

In [114]:
```
with pd.option_context('display.max_rows', None, 'display.max_columns', None):  # more options can be specified also
    print(merged)
```

```
                                                      geometry  Count
NAME_2
Ariyalur         POLYGON ((79.25848 10.97425, 79.25031 10.96728...      5
Chennai          POLYGON ((80.23345 12.96909, 80.22396 12.95902...    528
Coimbatore       POLYGON ((77.15656 10.78965, 77.15119 10.78552...    141
Cuddalore        POLYGON ((79.63276 11.23497, 79.62713 11.23186...     26
Dharmapuri       POLYGON ((78.70405 11.96791, 78.69788 11.94817...      1
Dindigul         POLYGON ((78.15361 10.08057, 78.15300 10.08310...     79
Erode            POLYGON ((77.84909 11.23407, 77.85358 11.22551...     70
Kancheepuram     MULTIPOLYGON (((79.96208 12.22216, 79.96208 12...     18
Kanniyakumari    MULTIPOLYGON (((77.55596 8.07903, 77.55596 8.0...     16
Karur            POLYGON ((78.49814 10.68002, 78.48006 10.68318...     41
Krishnagiri      POLYGON ((78.63942 12.24634, 78.64082 12.23947...      0
Madurai          POLYGON ((78.06779 9.74771, 78.06450 9.73725, ...     75
Nagappattinam    MULTIPOLYGON (((79.74597 10.27736, 79.74541 10...     44
Namakkal         POLYGON ((78.43343 11.33039, 78.43341 11.32425...     59
Perambalur       POLYGON ((78.99304 11.04352, 78.98911 11.04572...      7
Pudukkottai      POLYGON ((78.95749 9.98304, 78.95445 9.98641, ...      1
Ramanathapuram   MULTIPOLYGON (((79.53009 9.38596, 79.53014 9.3...     15
Salem            POLYGON ((78.22656 11.90686, 78.22643 11.90152...     31
Sivaganga        POLYGON ((78.70577 9.69627, 78.72670 9.69614, ...     12
Thanjavur        MULTIPOLYGON (((79.51510 10.32097, 79.51486 10...     55
The Nilgiris     POLYGON ((76.85727 11.34142, 76.85609 11.33792...      9
Theni            POLYGON ((77.34090 9.59505, 77.32851 9.57323, ...     44
Thiruvallur      MULTIPOLYGON (((80.02853 13.12436, 80.02964 13...     54
Thiruvarur       MULTIPOLYGON (((79.62569 10.34375, 79.62569 10...     30
Thoothukkudi     MULTIPOLYGON (((78.11542 8.61681, 78.11542 8.6...     27
Tiruchirappalli  POLYGON ((78.36396 10.29705, 78.35506 10.29553...     51
Tirunelveli      MULTIPOLYGON (((77.96347 8.32891, 77.96347 8.3...     65
Tiruppur         POLYGON ((77.43851 11.11145, 77.44918 11.10544...    112
Tiruvannamalai   POLYGON ((79.50778 12.34159, 79.50247 12.33965...     11
Vellore          POLYGON ((78.84798 12.62251, 78.84678 12.61580...     23
Viluppuram       MULTIPOLYGON (((79.94625 12.21915, 79.94625 12...     51
Virudunagar      POLYGON ((77.96670 9.24325, 77.95448 9.23935, ...     32
```

In [115]:

```
merged.describe()
```

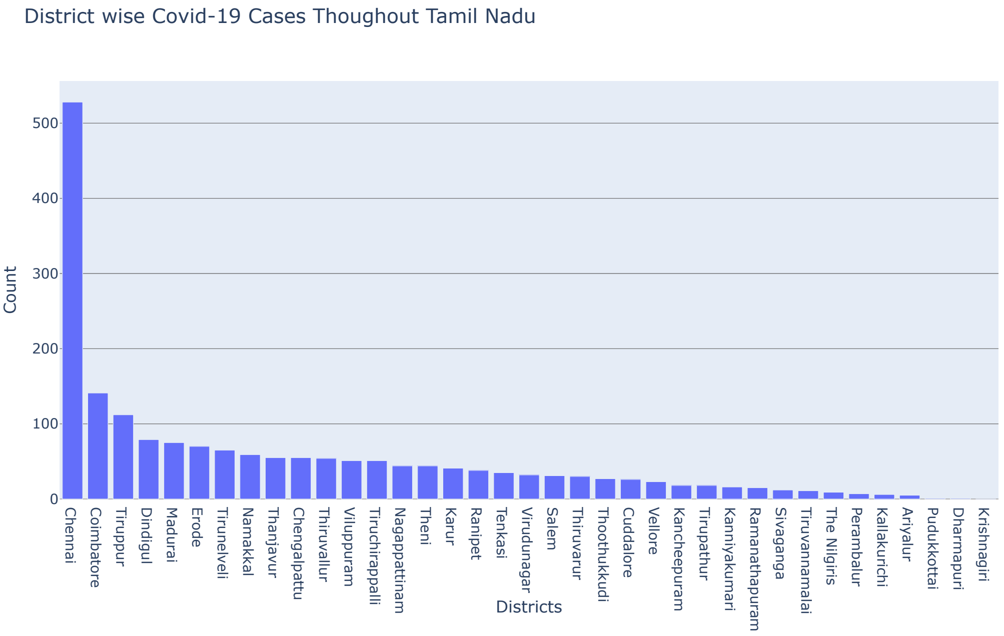
Out[115]:

	Count
count	32.000000
mean	54.156250
std	92.382483
min	0.000000
25%	14.250000
50%	31.500000
75%	56.000000
max	528.000000

In [116]:

```
fig = px.bar(district_wise, x='Districts', y='Count', title='District wise Covid-19 Cases Thoughout Tamil Nadu',hover_name='Count')

fig.show()
```



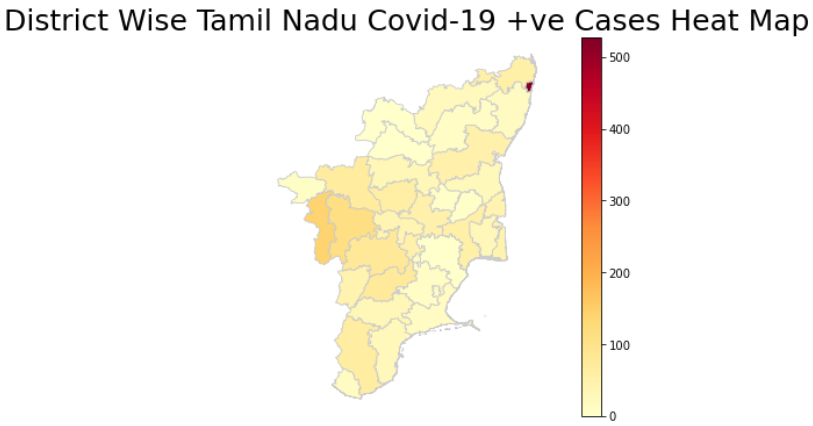
In [117]:

```
fig, ax = plt.subplots(1, figsize=(10, 6))
ax.axis('off')
ax.set_title('District Wise Tamil Nadu Covid-19 +ve Cases Heat Map', fontdict={'fontsize': '25', 'fontweight' : '3'})

merged.plot(column='Count', cmap='YlOrRd', linewidth=0.8, ax=ax, edgecolor='0.8', legend=True)
```

Out[117]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x28ca629e550>
```


District Wise Tamil Nadu Covid-19 +ve Cases Heat Map
500
400
300
200
100
0