Covid19

In [1]:

```
import requests
from pandas.io.json import json_normalize
URL = "https://api.covid19india.org/data.json"
data = requests.get(url=URL).json()
covid19_df = json_normalize(data['statewise'])
print("Total Confirmed Cases: "+str(covid19_df[covid19_df.state == "Total"]['confirmed'
]))
print("Total Active Cases: "+str(covid19_df[covid19_df.state == "Total"]['active']))
print("Total Recovered Cases: "+str(covid19_df[covid19_df.state == "Total"]['recovered'
]))
print("Total Deceased Cases: "+str(covid19_df[covid19_df.state == "Total"]['deaths']))
print(covid19_df[['state','confirmed','active','recovered','deaths']].sort_values(by="co
nfirmed", ascending=False))
```

```
Total Confirmed Cases: 0    17340
Name: confirmed, dtype: object
Total Active Cases: 0    13923
Name: active, dtype: object
Total Recovered Cases: 0    2858
Name: recovered, dtype: object
Total Deceased Cases: 0    559
Name: deaths, dtype: object
```

	state	confirmed	active	recovered	deaths
16	Bihar	96	52	42	2
8	Telangana	858	651	186	21
28	Puducherry	7	3	4	0
27	Goa	7	0	7	0
17	Odisha	68	43	24	1
9	Andhra Pradesh	647	565	65	17
18	Uttarakhand	44	33	11	0
1	Maharashtra	4200	3470	507	223
19	Jharkhand	41	39	0	2
10	Kerala	401	129	270	2
11	Karnataka	390	263	111	16
20	Himachal Pradesh	39	21	16	2
21	Chhattisgarh	36	11	25	0
12	Jammu and Kashmir	354	293	56	5
22	Assam	35	17	17	1
13	West Bengal	339	261	66	12
23	Chandigarh	26	13	13	0
14	Haryana	250	143	104	3
15	Punjab	244	191	37	16
2	Delhi	2003	1668	290	45
29	Manipur	2	1	1	0
30	Tripura	2	1	1	0
24	Ladakh	18	4	14	0
3	Gujarat	1743	1575	105	63
0	Total	17340	13923	2858	559
25	Andaman and Nicobar Islands	15	4	11	0
4	Rajasthan	1478	1250	205	23
5	Tamil Nadu	1477	1051	411	15
6	Madhya Pradesh	1407	1204	131	72
7	Uttar Pradesh	1100	956	127	17
26	Meghalaya	11	10	0	1
31	Mizoram	1	1	0	0
32	Arunachal Pradesh	1	0	1	0
33	Nagaland	0	0	0	0
34	Dadra and Nagar Haveli	0	0	0	0
35	Daman and Diu	0	0	0	0
36	Lakshadweep	0	0	0	0
37	Sikkim	0	0	0	0

In [2]:
```
covid19_df
```

Out[2]:

	active	confirmed	deaths	deltaconfirmed	deltadeaths	deltarecovered	lastupdatedtime	recovered	
0	13091	16125	528	402	7	40	19/04/2020 16:04:04	2506	
1	3072	3648	211	0	0	0	18/04/2020 21:44:05	365	Mah
2	1643	1893	43	0	0	0	18/04/2020 23:05:06	207	
3	992	1372	15	0	0	0	18/04/2020 18:35:10	365	Ta
4	1204	1431	22	80	1	5	19/04/2020 14:54:05	205	R
5	1206	1402	69	0	0	0	18/04/2020 23:05:08	127	
6	1452	1604	58	228	5	1	19/04/2020 12:14:06	94	
7	852	974	14	0	0	0	18/04/2020 20:11:19	108	Uttar
8	605	809	18	0	0	0	18/04/2020 21:15:08	186	Te
9	565	647	17	44	1	23	19/04/2020 15:35:05	65	
10	140	399	2	0	0	0	18/04/2020 18:07:06	257	
11	270	388	14	4	0	0	19/04/2020 12:39:06	104	K
12	285	341	5	0	0	0	18/04/2020 19:45:06	51	Jar
13	236	310	12	23	0	7	19/04/2020 12:14:07	62	Wes
14	139	246	3	14	0	4	19/04/2020 15:55:05	104	
15	187	234	16	0	0	0	18/04/2020 20:55:08	31	
16	45	89	2	3	0	0	19/04/2020 15:55:06	42	
17	36	61	1	0	0	0	18/04/2020 17:56:06	24	
18	33	42	0	0	0	0	18/04/2020 15:26:17	9	Utta
19	11	36	0	0	0	0	18/04/2020 19:36:06	25	Chh

	active	confirmed	deaths	deltaconfirmed	deltadeaths	deltarecovered	lastupdatedtime	recovered	
20	21	39	2	0	0	0	18/04/2020 19:15:06	16	H
21	21	34	1	0	0	0	18/04/2020 17:16:14	12	
22	36	38	2	5	0	0	19/04/2020 15:29:30	0	Jh
23	14	23	0	0	0	0	18/04/2020 17:25:07	9	Ch
24	4	18	0	0	0	0	16/04/2020 15:13:07	14	
25	4	15	0	1	0	0	19/04/2020 16:04:06	11	Anda
26	10	11	1	0	0	0	18/04/2020 12:38:07	0	Me
27	1	7	0	0	0	0	15/04/2020 20:10:36	6	
28	3	7	0	0	0	0	19/04/2020 08:25:07	4	Pu
29	1	2	0	0	0	0	06/04/2020 22:35:54	1	
30	1	2	0	0	0	0	10/04/2020 20:00:27	1	
31	1	1	0	0	0	0	26/03/2020 07:19:29	0	
32	0	1	0	0	0	0	16/04/2020 19:33:11	1	A
33	1	1	0	0	0	0	12/04/2020 23:35:29	0	N
34	0	0	0	0	0	0	17/04/2020 15:03:07	0	D Nag
35	0	0	0	0	0	0	26/03/2020 07:19:29	0	Da
36	0	0	0	0	0	0	26/03/2020 07:19:29	0	Laksl
37	0	0	0	0	0	0	26/03/2020 07:19:29	0	

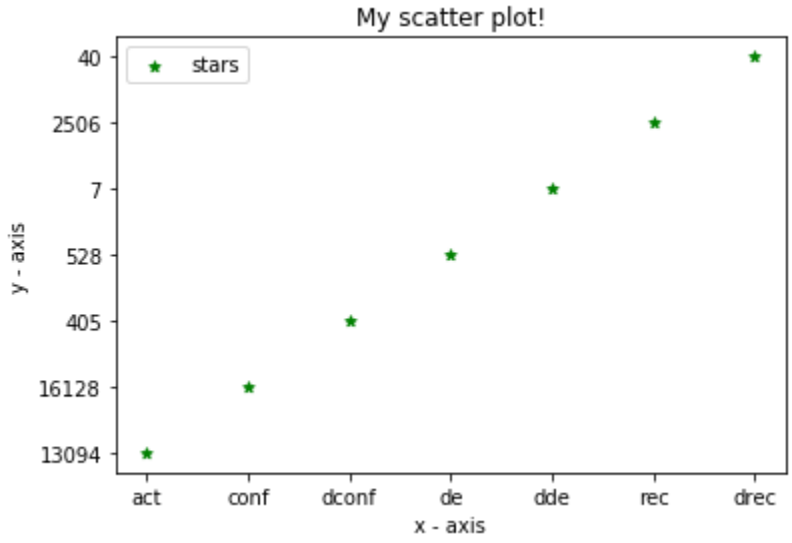
In [3]:
```
active=covid19_df['active'][0]
confirmed=covid19_df['confirmed'][0]
deltaconfirmed=covid19_df['deltaconfirmed'][0]
deaths=covid19_df['deaths'][0]
deltadeaths=covid19_df['deltadeaths'][0]
recovered=covid19_df['recovered'][0]
deltarecovered=covid19_df['deltarecovered'][0]
```

In [4]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
title=('active','confirmed','deltaconfirmed','deaths','deltadeaths','recovered','deltare
covered')
y_pos=np.arange(len(title))
performance=[active,confirmed,deltaconfirmed,deaths,deltadeaths,recovered,deltarecovered
]
#plt.plot([active,confirmed,deltaconfirmed,deaths,deltadeaths,recovered,deltarecovered])
```

ScatterPlot

In [6]:

```
import matplotlib.pyplot as plt
import requests
from pandas.io.json import json_normalize
URL = "https://api.covid19india.org/data.json"
data = requests.get(url=URL).json()
covid19_df = json_normalize(data['statewise'])
active=covid19_df['active'][0]
confirmed=covid19_df['confirmed'][0]
deltaconfirmed=covid19_df['deltaconfirmed'][0]
deaths=covid19_df['deaths'][0]
deltadeaths=covid19_df['deltadeaths'][0]
recovered=covid19_df['recovered'][0]
deltarecovered=covid19_df['deltarecovered'][0]

# x-axis values
x = ['act','conf','dconf','de','dde','rec','drec']
# y-axis values
y = [active,confirmed,deltaconfirmed,deaths,deltadeaths,recovered,deltarecovered]

# plotting points as a scatter plot
plt.scatter(x, y, label= "stars", color= "green",
            marker= "*", s=30)

# x-axis label
plt.xlabel('x - axis')
# frequency label
plt.ylabel('y - axis')
# plot title
plt.title('My scatter plot!')
# showing legend
plt.legend()

# function to show the plot
plt.show()
print('act = Total Active Cases')
print('cnf = Total Confirmed cases')
print('dconf = Total Deltaconfirmed cases')
print('de = Total Deaths')
print('dde = Total Deltadeaths')
print('rec = Total Recovered')
print('drec = Total Deltarecovered')
```



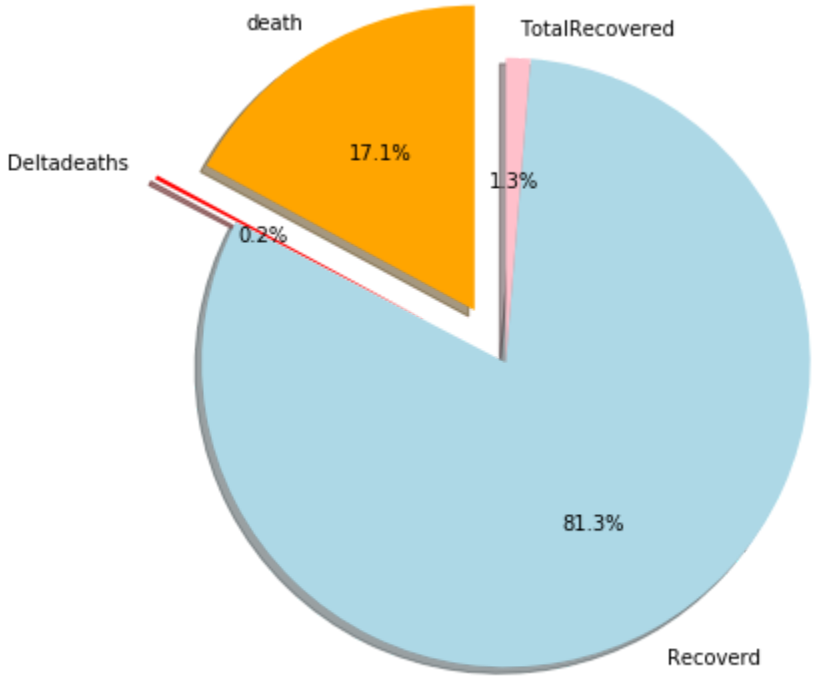

```
act = Total Active Cases
cnf = Total Confirmed cases
dconf = Total Deltaconfirmed cases
de = Total Deaths
dde = Total Deltadeaths
rec = Total Recovered
drec = Total Deltarecovered
```

pie chart

In [7]:

```
import matplotlib.pyplot as plt
import requests
from pandas.io.json import json_normalize
URL = "https://api.covid19india.org/data.json"
data = requests.get(url=URL).json()
covid19_df = json_normalize(data['statewise'])
active=covid19_df['active'][0]
confirmed=covid19_df['confirmed'][0]
deltaconfirmed=covid19_df['deltaconfirmed'][0]
deaths=covid19_df['deaths'][0]
deltadeaths=covid19_df['deltadeaths'][0]
recovered=covid19_df['recovered'][0]
deltarecovered=covid19_df['deltarecovered'][0]


labels=['death','Deltadeaths','Recoverd','TotalRecovered']
sizes=[deaths,deltadeaths,recovered,deltarecovered]
explode=[0.2,0.3,0,0]
colors = ['orange','red','lightblue','pink']
plt.figure(figsize = (10, 7))
plt.pie(sizes,labels=labels,colors=colors,shadow='true',autopct='%1.1f%%',explode=explod
e,startangle=90)
```

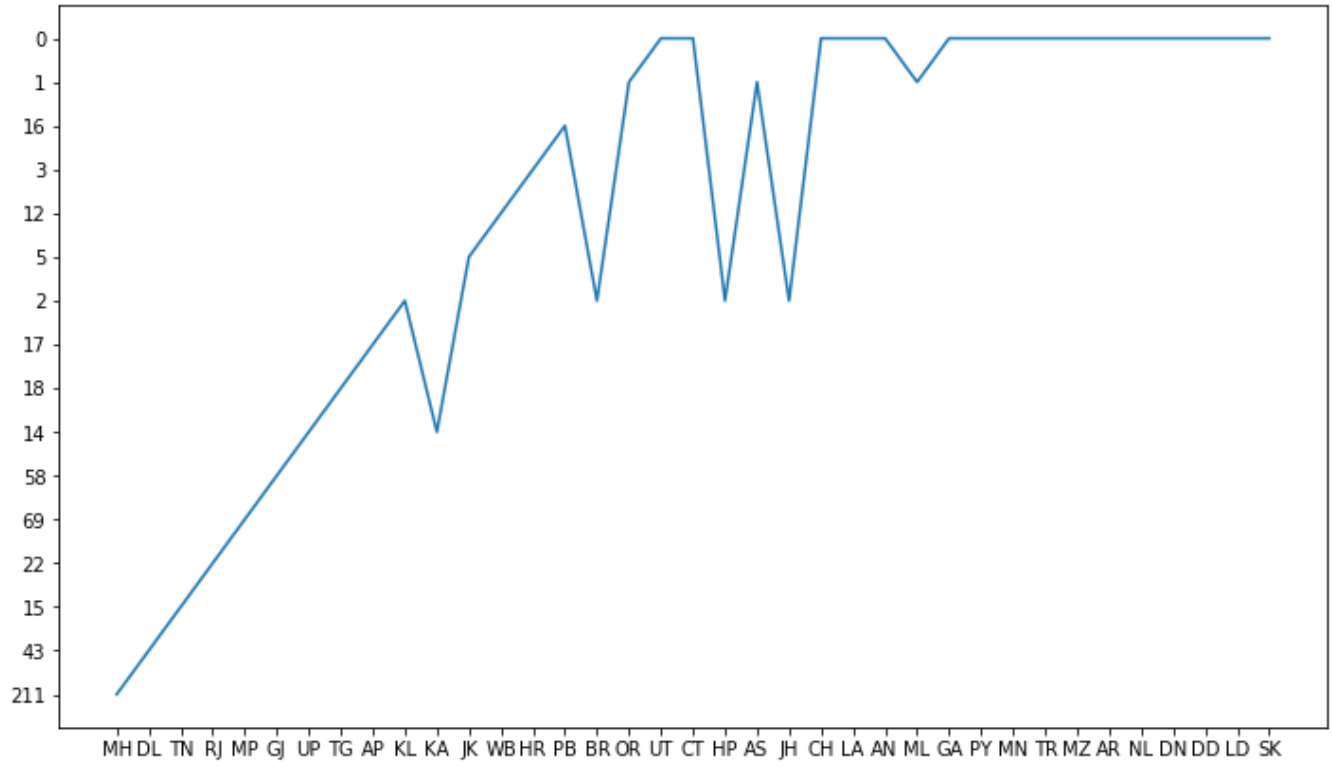
Out[7]:
```
([<matplotlib.patches.Wedge at 0x186524c2470>,
  <matplotlib.patches.Wedge at 0x186524c2eb8>,
  <matplotlib.patches.Wedge at 0x186524ca898>,
  <matplotlib.patches.Wedge at 0x186525d5278>],
 [Text(-0.6665739916695608, 1.1160999568272139, 'death'),
  Text(-1.2373086370653816, 0.6550323172511472, 'Deltadeaths'),
  Text(0.5319329419285788, -0.9628329789175314, 'Recoverd'),
  Text(0.04485304935574189, 1.0990851668380806, 'TotalRecovered')],
 [Text(-0.4101993794889604, 0.6868307426629009, '17.1%'),
  Text(-0.7954126952563167, 0.42109220394716607, '0.2%'),
  Text(0.290145241051952, -0.5251816248641079, '81.3%'),
  Text(0.024465299648586483, 0.5995010000934985, '1.3%')])
```



Num of death state wise : line plot

In [13]:

```
import matplotlib.pyplot as plt
import requests
import numpy as np
from pandas.io.json import json_normalize
URL = "https://api.covid19india.org/data.json"
data = requests.get(url=URL).json()
covid19_df = json_normalize(data['statewise'])
statename=[]
ideath=[]
for i in covid19_df['statecode'][1:]:
    statename.append(i)
for i in covid19_df['deaths'][1:]:
    ideath.append(i)

plt.figure(figsize = (12, 7))
plt.plot(statename, ideath)
plt.show()
import itertools
for (a,b) in zip(covid19_dfnew['statecode'], covid19_dfnew['state']):
    print(a , ' = ' , b)
```

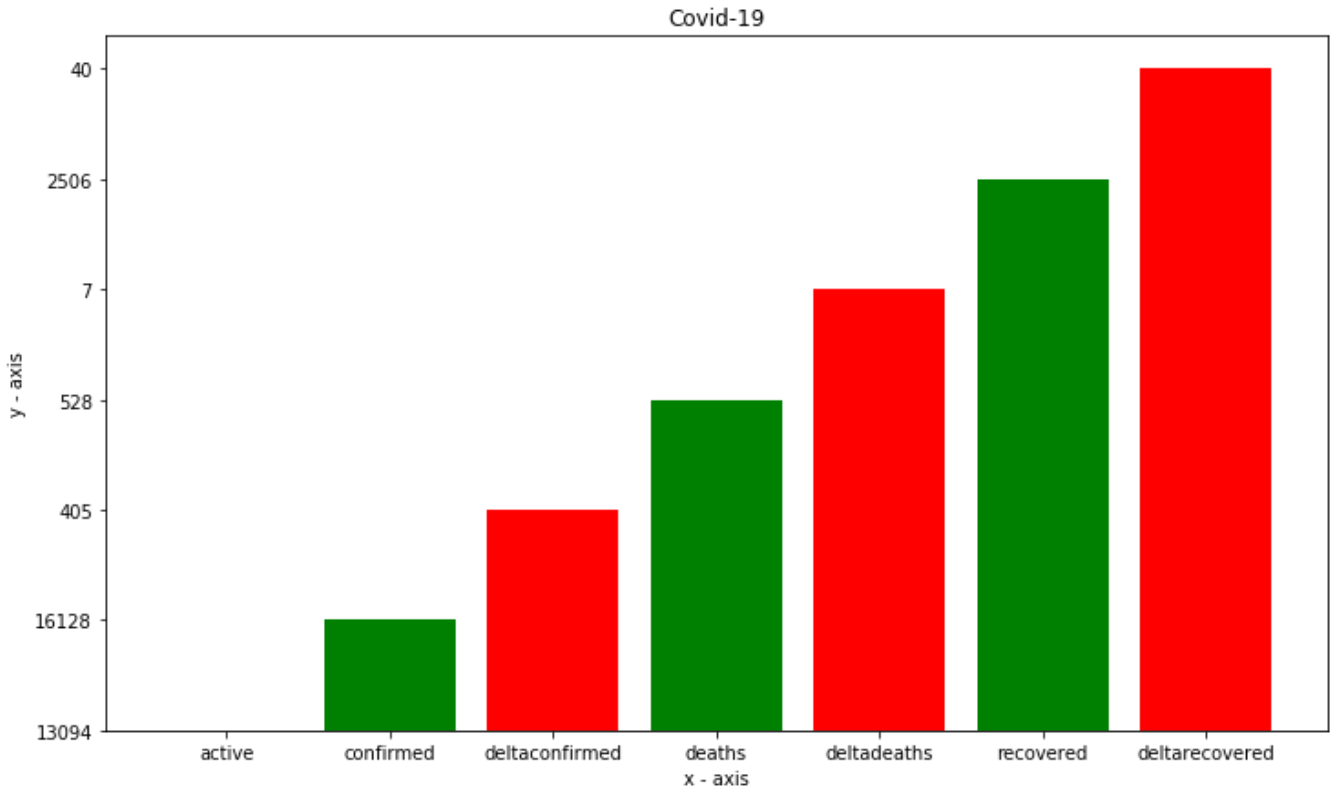

0
1
16
3
12
5
2
17
18
14
58
69
22
15
43
211
MH DL TN RJ MP GJ UP TG AP KL KA JK WB HR PB BR OR UT CT HP AS JH CH LA AN ML GA PY MN TR MZ AR NL DN DD LD SK

```
MH  =  Maharashtra
DL  =  Delhi
TN  =  Tamil Nadu
RJ  =  Rajasthan
MP  =  Madhya Pradesh
GJ  =  Gujarat
UP  =  Uttar Pradesh
TG  =  Telangana
AP  =  Andhra Pradesh
KL  =  Kerala
KA  =  Karnataka
JK  =  Jammu and Kashmir
WB  =  West Bengal
HR  =  Haryana
PB  =  Punjab
BR  =  Bihar
OR  =  Odisha
UT  =  Uttarakhand
CT  =  Chhattisgarh
HP  =  Himachal Pradesh
AS  =  Assam
JH  =  Jharkhand
CH  =  Chandigarh
LA  =  Ladakh
AN  =  Andaman and Nicobar Islands
ML  =  Meghalaya
GA  =  Goa
PY  =  Puducherry
MN  =  Manipur
TR  =  Tripura
MZ  =  Mizoram
AR  =  Arunachal Pradesh
NL  =  Nagaland
DN  =  Dadra and Nagar Haveli
DD  =  Daman and Diu
LD  =  Lakshadweep
SK  =  Sikkim
```

Bar Plot

In [10]:

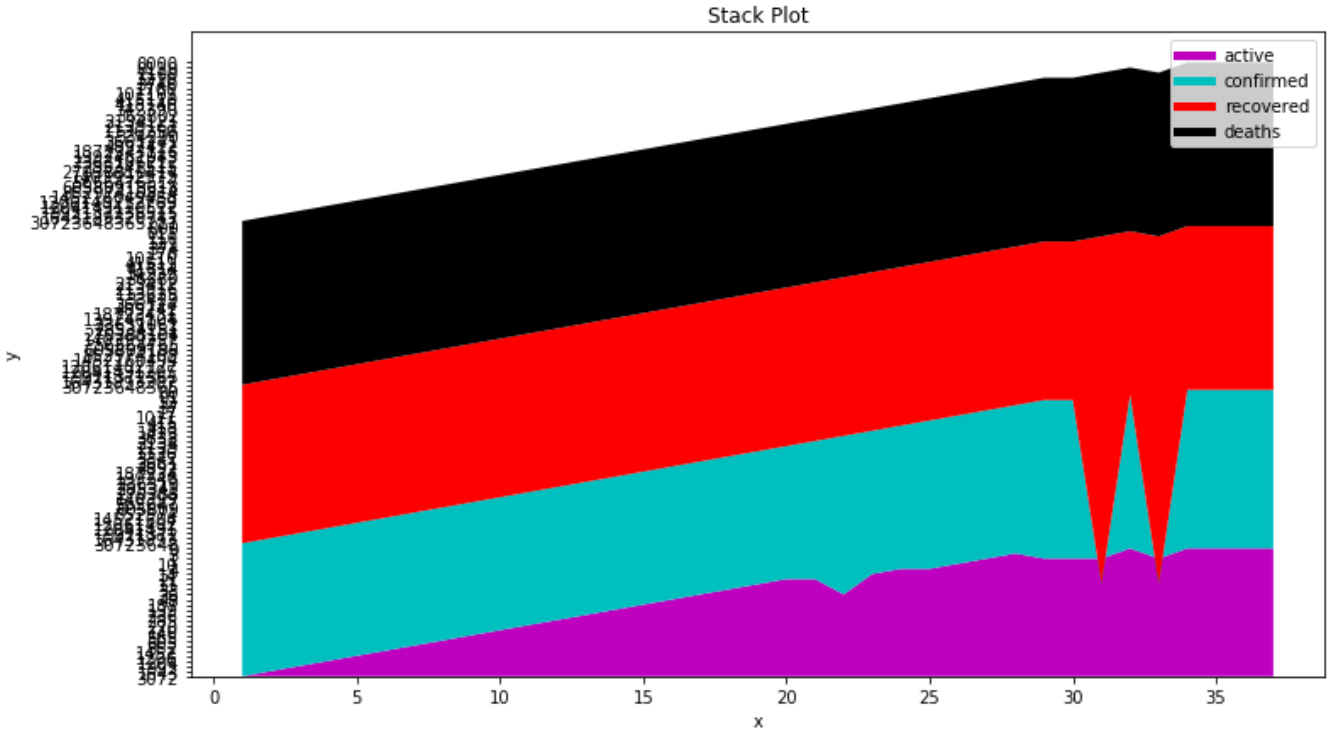
```
import matplotlib.pyplot as plt
import requests
from pandas.io.json import json_normalize
URL = "https://api.covid19india.org/data.json"
data = requests.get(url=URL).json()
covid19_df = json_normalize(data['statewise'])

active=covid19_df['active'][0]
confirmed=covid19_df['confirmed'][0]
deltaconfirmed=covid19_df['deltaconfirmed'][0]
deaths=covid19_df['deaths'][0]
deltadeaths=covid19_df['deltadeaths'][0]
recovered=covid19_df['recovered'][0]
deltarecovered=covid19_df['deltarecovered'][0]
time=covid19_df['lastupdatedtime'][0]
# x-coordinates of left sides of bars
left = [1,2,3,4,5,6,7]

# heights of bars
height = [active,confirmed,deltaconfirmed,deaths,deltadeaths,recovered,deltarecovered]

# labels for bars
tick_label = ['active','confirmed','deltaconfirmed','deaths','deltadeaths','recovered',
'deltarecovered']
plt.figure(figsize = (12, 7))
# plotting a bar chart
plt.bar(left, height, tick_label = tick_label,
        width = 0.8, color = ['red', 'green'])

# naming the x-axis
plt.xlabel('x - axis')
# naming the y-axis
plt.ylabel('y - axis')
# plot title
plt.title('Covid-19')

plt.show()
print('last updated time = ' , time)
```



last updated time = 19/04/2020 16:14:05

Area Plot:-

In [11]:
```
import matplotlib.pyplot as plt
import requests
from pandas.io.json import json_normalize
URL = "https://api.covid19india.org/data.json"
data = requests.get(url=URL).json()
covid19_df = json_normalize(data['statewise'])
covid19_dfnew=covid19_df.drop(covid19_df.index[[0]])
count=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
31,32,33,34,35,36,37]

active =covid19_dfnew['active']
confirmed=covid19_dfnew['confirmed']
recovered =covid19_dfnew['recovered']
deaths = covid19_dfnew['deaths']
plt.figure(figsize = (12, 7))
plt.plot([],[],color='m', label='active', linewidth=5)
plt.plot([],[],color='c', label='confirmed', linewidth=5)
plt.plot([],[],color='r', label='recovered', linewidth=5)
plt.plot([],[],color='k', label='deaths', linewidth=5)

plt.stackplot(count, active,confirmed,recovered,deaths, colors=['m','c','r','k'])
plt.xlabel('x')
plt.ylabel('y')
plt.title('Stack Plot')
plt.legend()
plt.show()
```

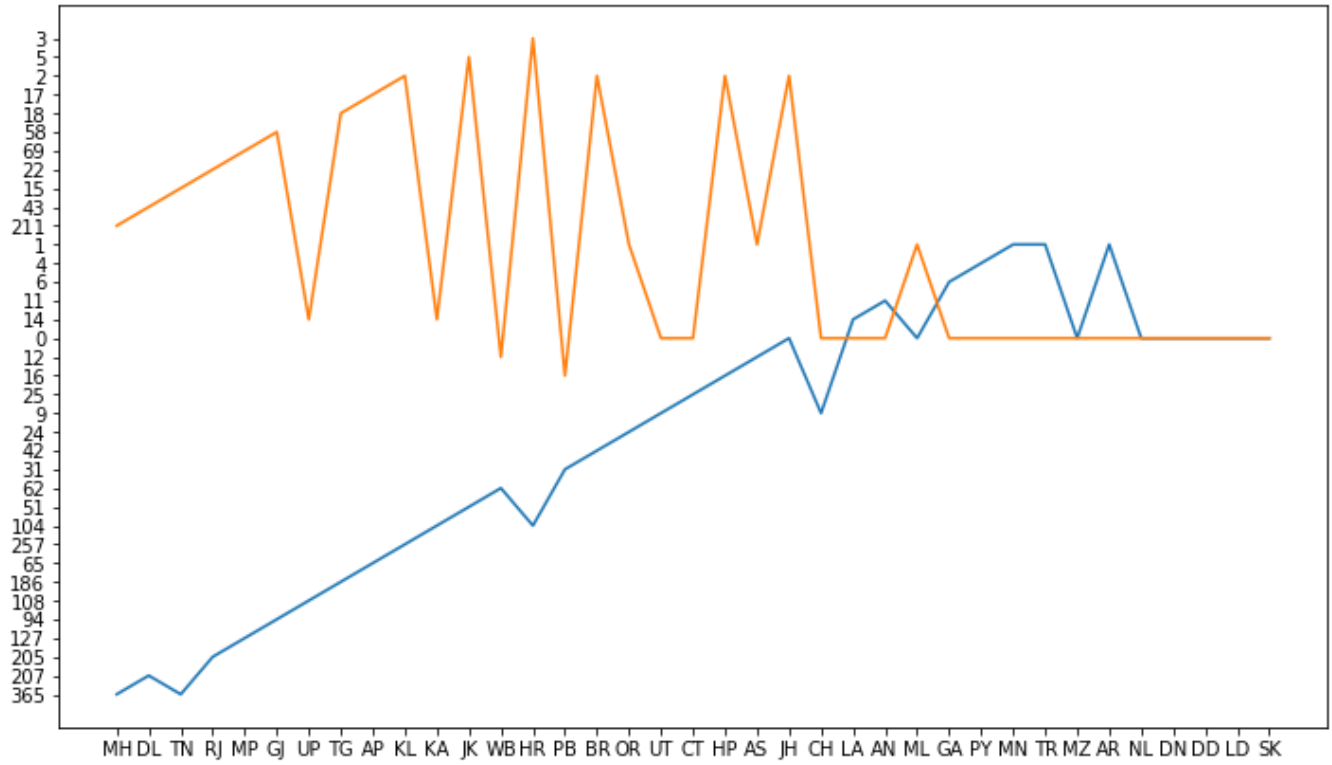


Line Plot - Deaths and Recovered

In [12]:

```
import matplotlib.pyplot as plt
import requests
from pandas.io.json import json_normalize
URL = "https://api.covid19india.org/data.json"
data = requests.get(url=URL).json()
covid19_df = json_normalize(data['statewise'])
covid19_dfnew=covid19_df.drop(covid19_df.index[[0]])
plt.figure(figsize = (12, 7))
statecode=covid19_dfnew['statecode']
recovered=covid19_dfnew['recovered']
deaths=covid19_dfnew['deaths']
plt.plot(statecode, recovered, label='covid-19')
plt.plot(statecode, deaths, label='covid-19')
import itertools
for (a,b) in zip(covid19_dfnew['statecode'], covid19_dfnew['state']):
    print(a , ' = ' , b)
```

MH = Maharashtra
DL = Delhi
TN = Tamil Nadu
RJ = Rajasthan
MP = Madhya Pradesh
GJ = Gujarat
UP = Uttar Pradesh
TG = Telangana
AP = Andhra Pradesh
KL = Kerala
KA = Karnataka
JK = Jammu and Kashmir
WB = West Bengal
HR = Haryana
PB = Punjab
BR = Bihar
OR = Odisha
UT = Uttarakhand
CT = Chhattisgarh
HP = Himachal Pradesh
AS = Assam
JH = Jharkhand
CH = Chandigarh
LA = Ladakh
AN = Andaman and Nicobar Islands
ML = Meghalaya
GA = Goa
PY = Puducherry
MN = Manipur
TR = Tripura
MZ = Mizoram
AR = Arunachal Pradesh
NL = Nagaland
DN = Dadra and Nagar Haveli
DD = Daman and Diu
LD = Lakshadweep
SK = Sikkim


3
5
2
17
18
58
69
22
15
43
211
1
4
6
11
14
0
12
16
25
9
24
42
31
62
51
104
257
65
186
108
94
127
205
207
365
MH DL TN RJ MP GJ UP TG AP KL KA JK WB HR PB BR OR UT CT HP AS JH CH LA AN ML GA PY MN TR MZ AR NL DN DD LD SK

In [ ]:

/