Visualize current state - plot COVID data

In [1]:

```
countryToAnalyze = "India"
stateToAnalyze ="Karnataka"
```

In [2]:

```
#Download data from "https://github.com/CSSEGISandData/COVID-19.git"

confirmedCsv = "COVID-19/csse_covid_19_data/csse_covid_19_time_series/time_serie
s_covid19_confirmed_global.csv"
recoveredCsv = "COVID-19/csse_covid_19_data/csse_covid_19_time_series/time_serie
s_covid19_recovered_global.csv"
deathsCsv = "COVID-19/csse_covid_19_data/csse_covid_19_time_series/time_series_c
ovid19_deaths_global.csv"

try:
    f = open(confirmedCsv)
except IOError:
    print('Download data from "https://github.com/CSSEGISandData/COVID-19.git"')
    assert False
finally:
    f.close()
```

In [3]:

```
import pandas as pd

confirmedDf = pd.read_csv(confirmedCsv)
recoveredDf = pd.read_csv(recoveredCsv)
deathsDf = pd.read_csv(deathsCsv)
```

In [4]:

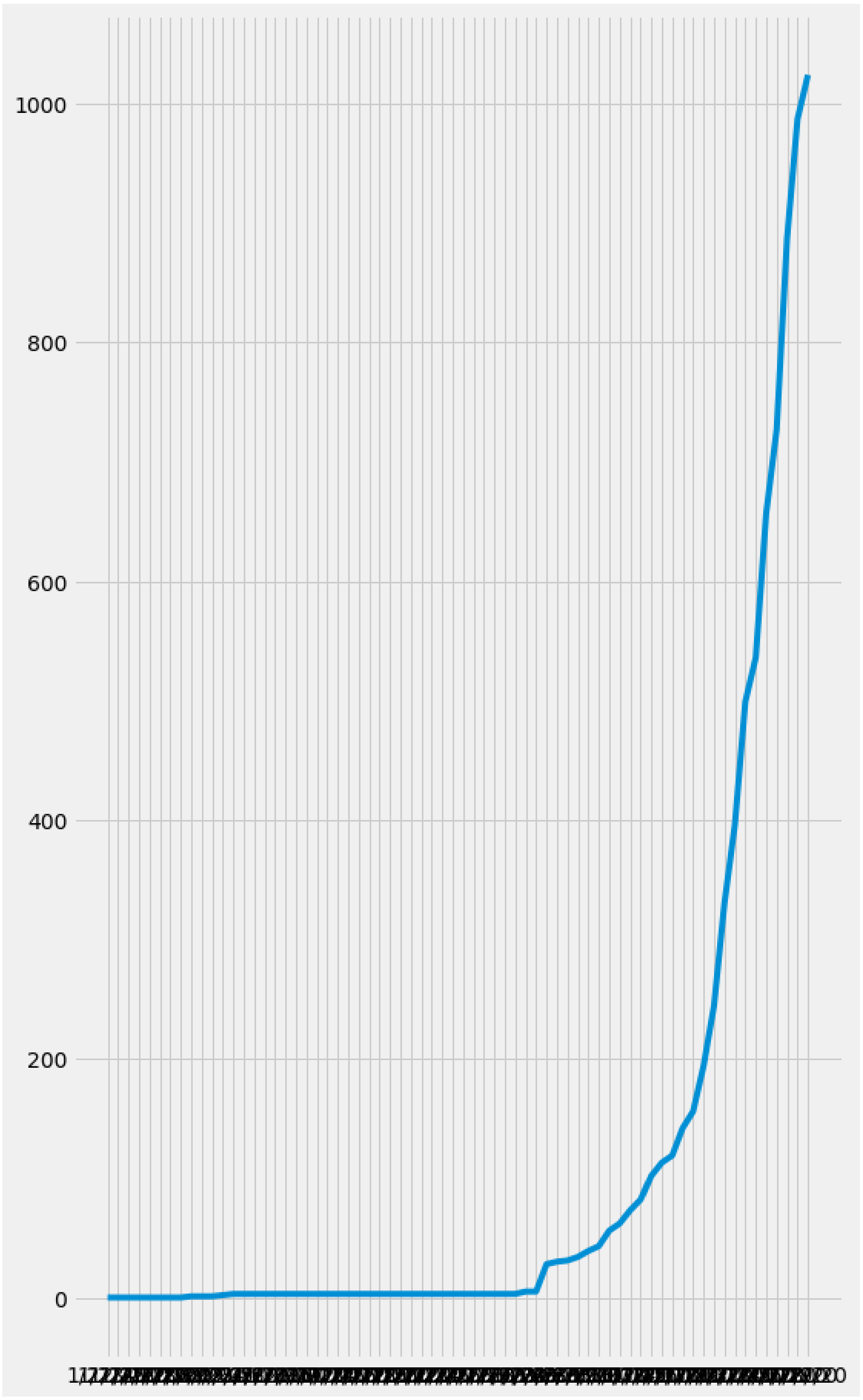
```
#from pandas_profiling import ProfileReport

##profile = ProfileReport(confirmedDf, title='Confirmed - Pandas Profiling Repor
t', html={'style':{'full_width':True}})
#profile = ProfileReport(confirmedDf)

#profile
##profile.to_widgets()
##profile.to_notebook_iframe()
```

In [5]:

```
from matplotlib import pyplot
%matplotlib inline

pyplot.style.use("fivethirtyeight")# for pretty graphs

# Increase the default plot size and set the color scheme
pyplot.rcParams['figure.figsize'] = 8, 15

confirmedTSDf = confirmedDf.loc[confirmedDf["Country/Region"] == countryToAnalyz
e].T[4:]
pyplot.figure(1)
pyplot.plot(confirmedTSDf)
```

Out[5]:

```
[<matplotlib.lines.Line2D at 0x117577710>]
```



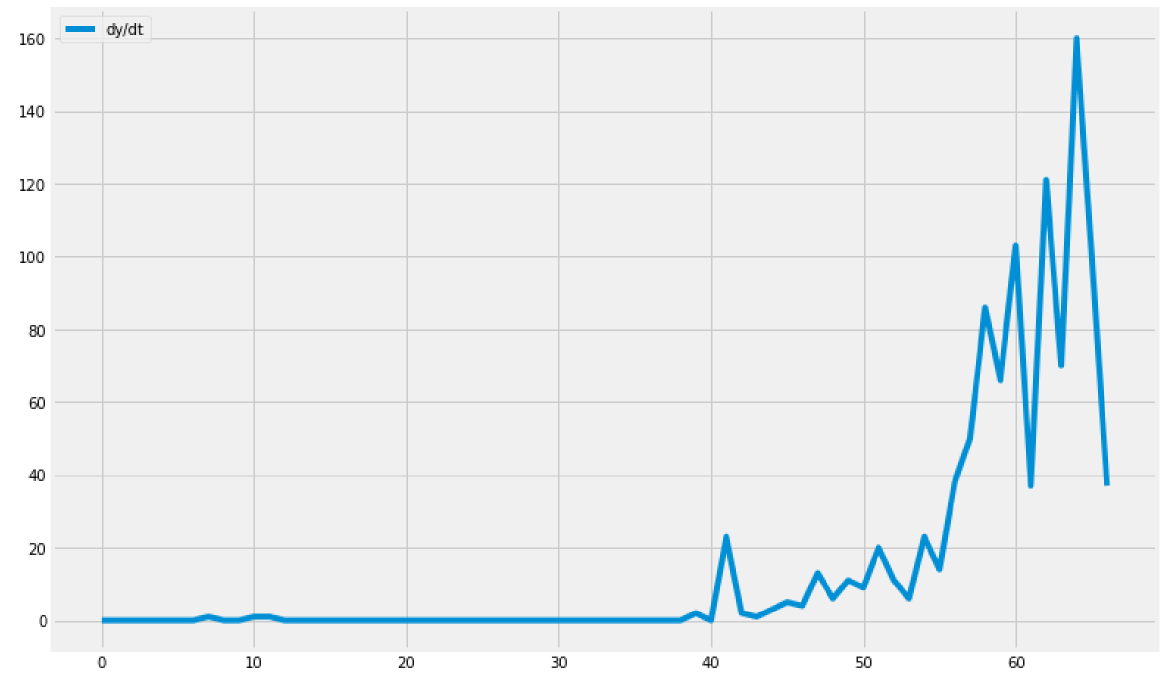
In [6]:

```
%matplotlib inline
import numpy as np
print(confirmedTSDf.values.flatten())
dy_dt= np.diff(confirmedTSDf.values.flatten())
print(dy_dt)
f, ax = pyplot.subplots(figsize=(12, 8))
pyplot.plot(dy_dt, label="dy/dt")
pyplot.legend()
```

```
[0  0  0  0  0  0  0  0  1  1  1  2  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3
3  3  3
  3  3  3  5  5  28  30  31  34  39  43  56  62  73  82  102  113  119  142  156  194  244
330
  396  499  536  657  727  887  987  1024]
[0  0  0  0  0  0  0  1  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0
  0  0  2  0  23  2  1  3  5  4  13  6  11  9  20  11  6  23  14  38  50  86  66  103  37  121
70
  160  100  37]
```

Out[6]:

```
<matplotlib.legend.Legend at 0x118301b50>
```

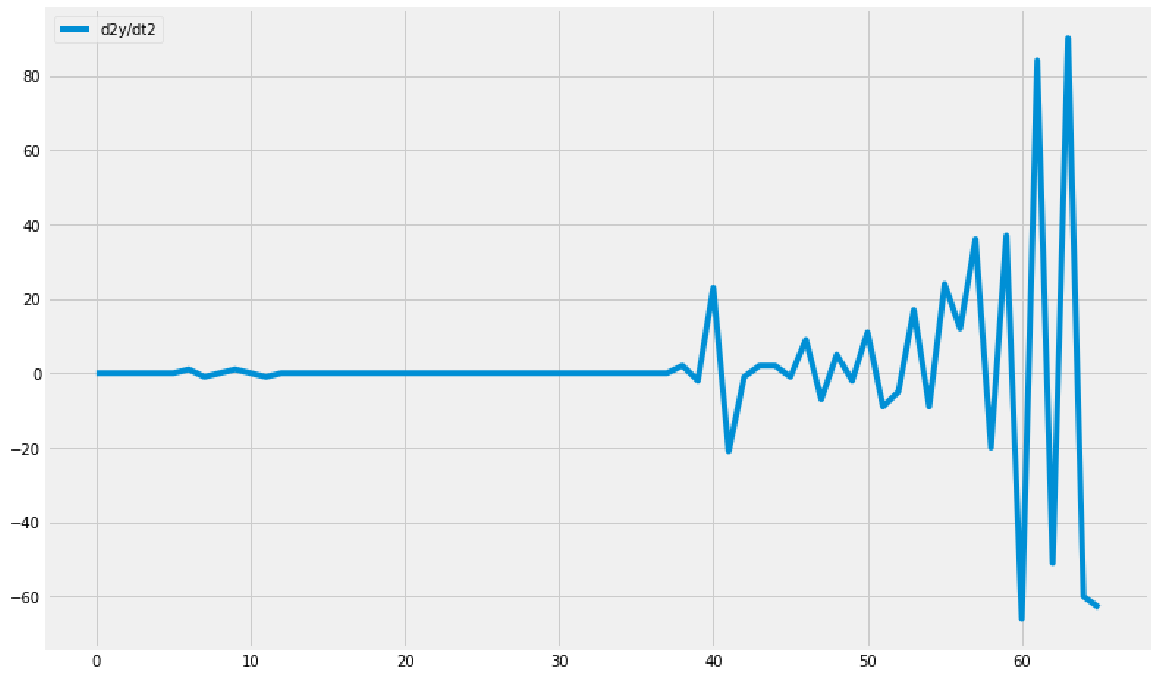


In [7]:

```
d2y_dt2= np.diff(dy_dt)
f, ax = pyplot.subplots(figsize=(12, 8))
pyplot.plot(d2y_dt2, label="d2y/dt2")
pyplot.legend()
```

Out[7]:

```
<matplotlib.legend.Legend at 0x11875c810>
```



Compute SEIR parametes - alpha, beta, gamma

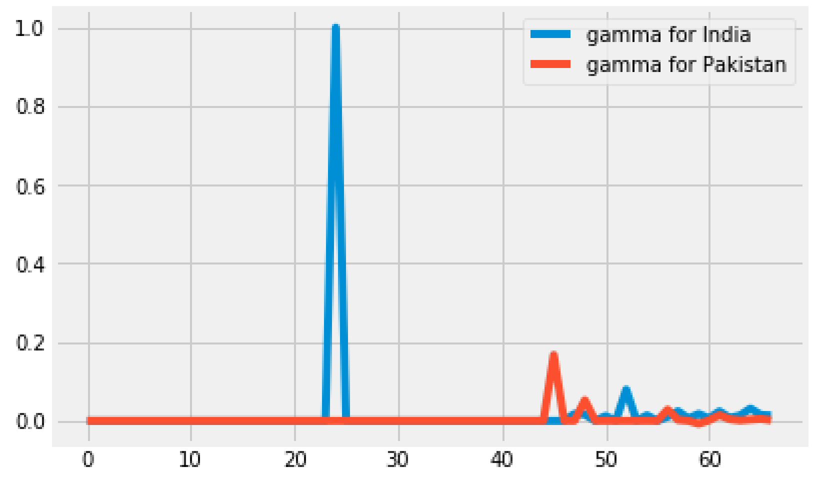
In [8]:

```
countries = list([countryToAnalyze, "Pakistan"])
pyplot.figure(1)

#This is incomplete - WIP
for country in countries:
    print(country)
    confirmedTSDf = confirmedDf.loc[confirmedDf["Country/Region"] == country].T[
4:]
    recoveredTSDf = recoveredDf.loc[recoveredDf["Country/Region"] == country].T[
4:]
    deathsTSDf = deathsDf.loc[deathsDf["Country/Region"] == country].T[4:]
    removedArr = recoveredTSDf.values + deathsTSDf.values
    dR_dt= np.diff(removedArr.flatten())
    confirmedArr = confirmedTSDf[1:].values.flatten()
    confirmedArr[confirmedArr == 0] = 0.0001 # to prevent divide by zero
    gamma = dR_dt / confirmedArr
    #print(confirmedTSDf)
    #print(recoveredTSDf)
    #print(removedArr)
    #print("dr_dt",dR_dt)
    #print(gamma)
    pyplot.plot(gamma, label="gamma for "+country)

pyplot.legend()
```

```
India
Pakistan
```

Out[8]:

```
<matplotlib.legend.Legend at 0x118789a10>
```



Analyze statewise for India

Download data from - [https://www.kaggle.com/sudalairajkumar/covid19-in-india (https://www.kaggle.com/sudalairajkumar/covid19-in-india)](https://www.kaggle.com/sudalairajkumar/covid19-in-india)

In [9]:

```
from datetime import datetime

covidDataFile = "covid19-in-india/covid_19_india.csv"
populationFile = "covid19-in-india/population_india_census2011.csv"
hospitalBedsFile = "covid19-in-india/HospitalBedsIndia.csv"
icmrTestingFile = "covid19-in-india/ICMRTestingDetails.csv"

try:
    f = open(covidDataFile)
except IOError:
    print('Download data from "https://www.kaggle.com/sudalairajkumar/covid19-in
-india"')
    assert False
finally:
    f.close()


def parser(x):
    return datetime.strptime(x, '%d/%m/%y')

def icmrDateParser(x):
    return datetime.strptime(x, '%d/%m/%y %H:%M')

covidIndiaDataDf = pd.read_csv(covidDataFile, parse_dates=[1], index_col=1, sque
eze=True, date_parser=parser)
populationDf = pd.read_csv(populationFile)
hospitalBedsDf = pd.read_csv(hospitalBedsFile)
icmrTestingDf = pd.read_csv(icmrTestingFile, parse_dates=[1], date_parser=icmrDa
teParser)
```

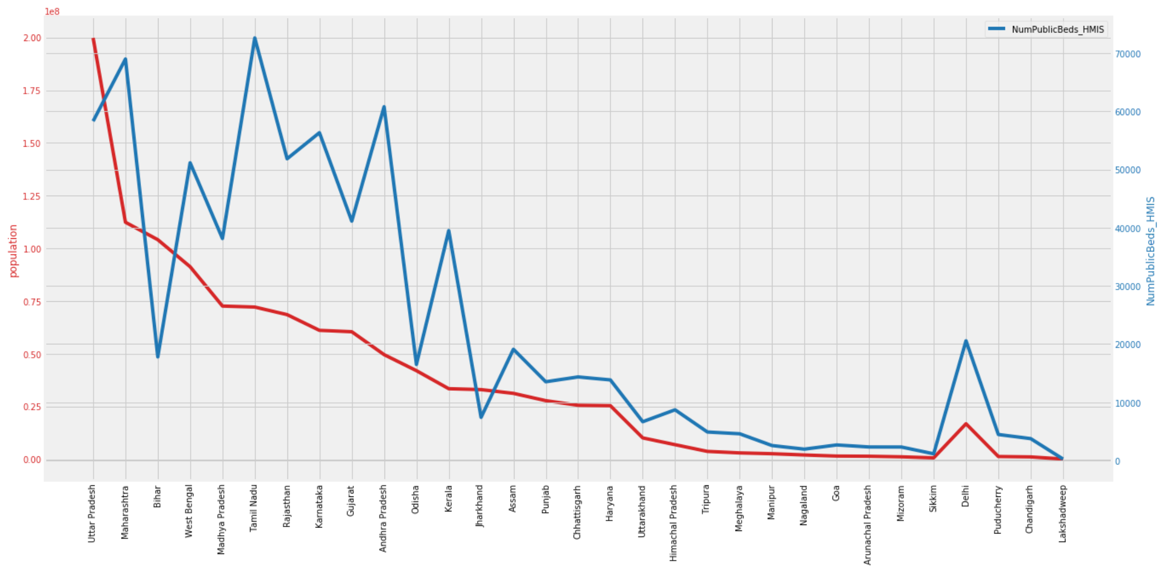
Capacity for maximum

In [10]:

```
%matplotlib inline


hospitalBedsDf.dropna(axis=0, how='all', inplace=True)
hospitalBedsDf = hospitalBedsDf.fillna(-1)
hospitalBedsDf = hospitalBedsDf.loc[hospitalBedsDf["State/UT"] != "All India"]
hospitalBedsDf["State / Union Territory"] = hospitalBedsDf["State/UT"]

populationHospitalBedsdf = pd.merge(populationDf, hospitalBedsDf, on=['State / U
nion Territory'])
states = populationHospitalBedsdf["State / Union Territory"]

fig, ax1 = pyplot.subplots(figsize=(20,10))

color = 'tab:red'
ax1.set_ylabel('population', color=color)
pyplot.xticks(rotation=90)
ax1.plot(states, populationHospitalBedsdf["Population"], label ="population", co
lor=color)
ax1.tick_params(axis='y', labelcolor=color)

color = 'tab:blue'
ax2 = ax1.twinx()
ax2.set_ylabel('NumPublicBeds_HMIS', color=color)

ax2.plot(states, populationHospitalBedsdf["NumPublicBeds_HMIS"].astype(int), lab
el="NumPublicBeds_HMIS", color=color)
ax2.tick_params(axis='y', labelcolor=color)
pyplot.legend()
```

Out[10]:

```
<matplotlib.legend.Legend at 0x105ce5950>
```



State-wise numbers

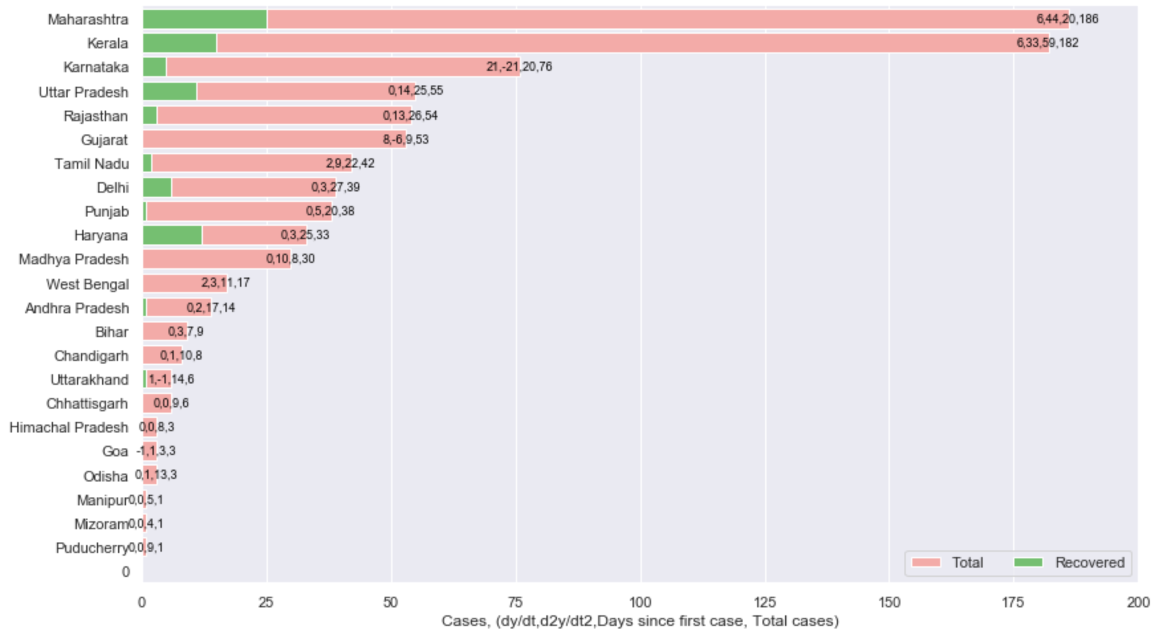
In [11]:

```
covidIndiaLastDayDataDf = pd.DataFrame(columns=covidIndiaDataDf.columns.values)

covidIndiaLastDayDataDf.insert(len(covidIndiaDataDf.columns), "dy_dt", [], True)
covidIndiaLastDayDataDf.insert(len(covidIndiaDataDf.columns), "d2y_dt2", [], Tru
e)
covidIndiaLastDayDataDf.insert(len(covidIndiaDataDf.columns), "days", [], True)


for state in states:
    stateDataDf = covidIndiaDataDf.loc[covidIndiaDataDf["State/UnionTerritory"]
==state]
    stateDataDf.sort_values('Date',ascending=False,inplace=True)
    if stateDataDf.shape[0] != 0:
        covidIndiaLastDayDataDf = covidIndiaLastDayDataDf.append(stateDataDf.ilo
c[0])
        dy_dt= np.diff(stateDataDf["ConfirmedIndianNational"].values)
        d2y_dt2= np.diff(dy_dt)
        days = 0
        if len(stateDataDf.index) > 0:
            date = stateDataDf.index[-1]
            days = (stateDataDf.index[0] - stateDataDf.index[-1]).days
            #print(days)
        else:
            date=datetime.strptime("1/1/1970", '%d/%m/%y')
        if date == 0:
            date=datetime.strptime("1/1/1970", '%d/%m/%y')
        last_dy_dt = 0
        if dy_dt.shape[0] != 0:
            last_dy_dt = dy_dt[0]
        last_d2y_dt2 = 0
        if d2y_dt2.shape[0] != 0:
            last_d2y_dt2 = d2y_dt2[0]
        covidIndiaLastDayDataDf.iloc[-1,-1] = -1*last_dy_dt
        covidIndiaLastDayDataDf.iloc[-1, -2] = -1*last_d2y_dt2
        covidIndiaLastDayDataDf.iloc[-1, -3] = days
    else:
        covidIndiaLastDayDataDf = covidIndiaLastDayDataDf.append(pd.Series(), ig
nore_index=True)
        covidIndiaLastDayDataDf.iloc[-1, 1] = state


covidIndiaLastDayDataDf.fillna(0, inplace=True)
```

```
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:10:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/panda
s-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cop
y
  # Remove the CWD from sys.path while we load stuff.
```

In [12]:

```
import seaborn as sns

# Disable warnings
import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
sns.set()

f, ax = pyplot.subplots(figsize=(12, 8))
covidIndiaLastDayDataDf['Name of State / UT']=covidIndiaLastDayDataDf['State/Uni
onTerritory']
covidIndiaLastDayDataDf['Total cases']=covidIndiaLastDayDataDf['ConfirmedIndianN
ational']+covidIndiaLastDayDataDf['ConfirmedForeignNational']
covidIndiaLastDayDataDf['Cured/Discharged/Migrated']=covidIndiaLastDayDataDf['Cu
red']
data = covidIndiaLastDayDataDf[['Name of State / UT','Total cases','Cured/Discha
rged/Migrated','Deaths', 'dy_dt', 'd2y_dt2', 'days']]

data.sort_values('Total cases',ascending=False,inplace=True)
sns.set_color_codes("pastel")
sns.barplot(x="Total cases", y="Name of State / UT", data=data,
            label="Total", color="r", ci=None)

sns.set_color_codes("muted")
g =sns.barplot(x="Cured/Discharged/Migrated", y="Name of State / UT", data=data,
            label="Recovered", color="g", ci=None)


# Add a legend and informative axis label
ax.legend(ncol=2, loc="lower right", frameon=True)
ax.set(xlim=(0, 200), ylabel="",
       xlabel="Cases, (dy/dt,d2y/dt2,Days since first case, Total cases)")
sns.despine(left=True, bottom=True)


order = 0
for index, row in data.iterrows():
    val = str(int(row['dy_dt'])) + "," + str(int(row['d2y_dt2'])) + "," + str(in
t(row['days'])) + ","+str(row["Total cases"])
    if val != "0,0,0,0":
        g.text(row["Total cases"], order, val, color='black', ha="center", fonts
ize=9, horizontalalignment='left',  verticalalignment='center')
    order += 1
```



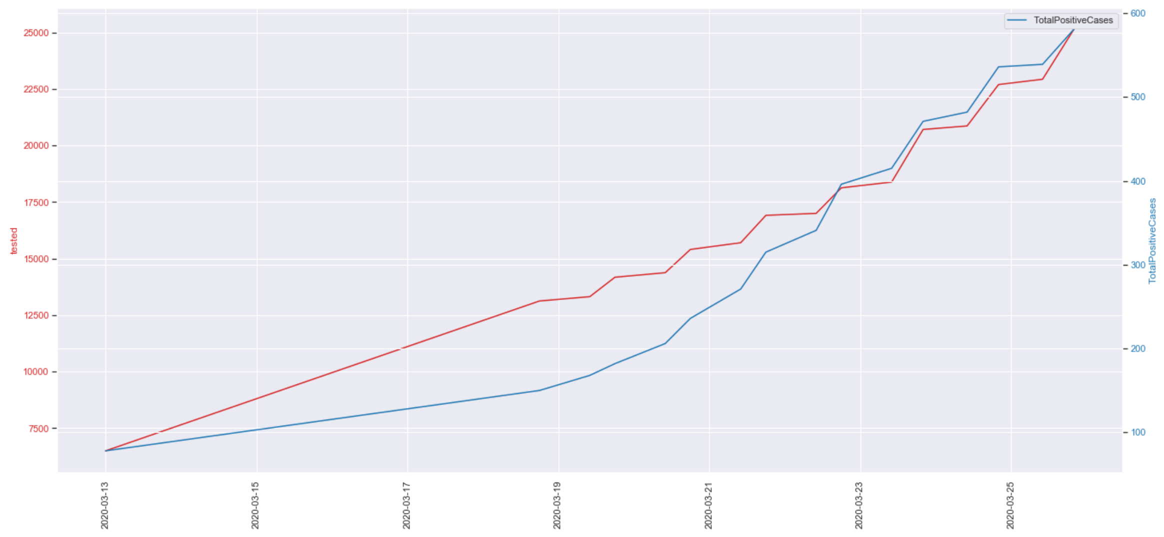
Testing to positive - trend and ratio

In [13]:

```
fig, ax1 = pyplot.subplots(figsize=(20,10))

color = 'tab:red'
ax1.set_ylabel('tested', color=color)
pyplot.xticks(rotation=90)
ax1.plot(icmrTestingDf["DateTime"], icmrTestingDf["TotalSamplesTested"], label =
"TotalSamplesTested", color=color)
ax1.tick_params(axis='y', labelcolor=color)

color = 'tab:blue'
ax2 = ax1.twinx()
ax2.set_ylabel('TotalPositiveCases', color=color)

ax2.plot(icmrTestingDf["DateTime"], icmrTestingDf["TotalPositiveCases"], label=
"TotalPositiveCases", color=color)
ax2.tick_params(axis='y', labelcolor=color)
pyplot.legend()
```

Out[13]:

```
<matplotlib.legend.Legend at 0x1a1b4cd110>
```



Giving an Index to states based on their handling - where to focus?

Lower the better

In [14]:

```
#Penalty for delay since first case
data["responseIndex"] = (0.001+data["days"])
#Penalty for rate of growth of cases
data["responseIndex"] = data["responseIndex"] + data["responseIndex"] * 1000*(0.
001+data["dy_dt"])
#Penalty if the rate of rate itself is up
data["responseIndex"] = data["responseIndex"] + data["responseIndex"] * 1000*(0.
001+data["d2y_dt2"])

data["responseIndex"] = data["responseIndex"] + data["responseIndex"] *100*(0.00
1+data["Total cases"])/(data["Total cases"].sum())

data.sort_values('responseIndex',ascending=False,inplace=True)
data.reset_index()
indexData = data[['Name of State / UT', 'responseIndex']]
indexData.drop(index=0, inplace=True)
display(indexData)
```

	Name of State / UT	responseIndex
11	Kerala	2.590135e+11
1	Maharashtra	1.195272e+11
5	Tamil Nadu	2.332954e+09
3	West Bengal	1.968183e+08
6	Rajasthan	4.921676e+06
14	Punjab	1.084232e+06
27	Delhi	8.973010e+05
16	Haryana	7.261116e+05
4	Madhya Pradesh	7.183916e+05
9	Andhra Pradesh	1.788948e+05
2	Bihar	8.602797e+04
29	Chandigarh	3.868806e+04
10	Odisha	3.514564e+04
15	Chhattisgarh	6.112726e+01
18	Himachal Pradesh	4.317191e+01
28	Puducherry	4.019470e+01
21	Manipur	2.233237e+01
25	Mizoram	1.786679e+01
26	0	4.000465e-03
20	0	4.000465e-03
12	0	4.000465e-03
24	0	4.000465e-03
22	0	4.000465e-03
30	0	4.000465e-03
19	0	4.000465e-03
13	0	4.000465e-03
23	Goa	-4.048193e+06
17	Uttarakhand	-2.377067e+07
8	Gujarat	-3.094461e+09
7	Karnataka	-8.676955e+10

Analyze a particular State

In [15]:

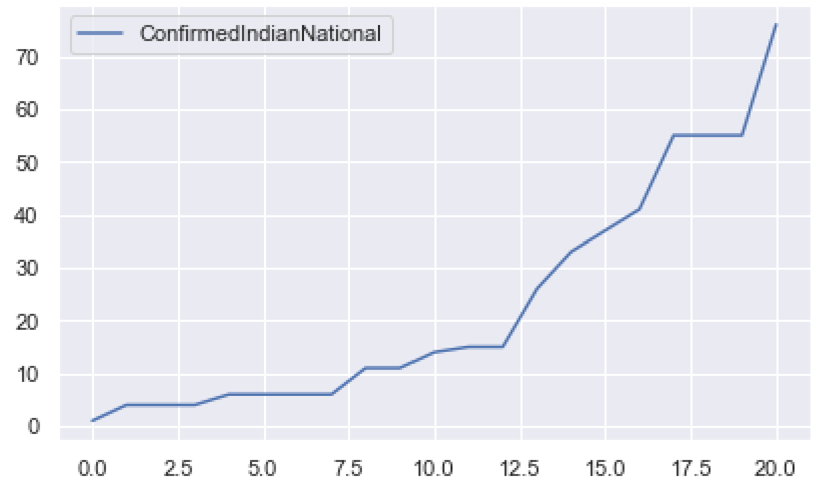
```
covidStateDataDf = covidIndiaDataDf.loc[covidIndiaDataDf["State/UnionTerritory"]
==stateToAnalyze]
```

In [16]:

```
%matplotlib inline
pyplot.figure(1)
pyplot.plot(covidStateDataDf["ConfirmedIndianNational"].values, label="Confirmed
IndianNational")
pyplot.legend()
```

Out[16]:

```
<matplotlib.legend.Legend at 0x1a1b581310>
```

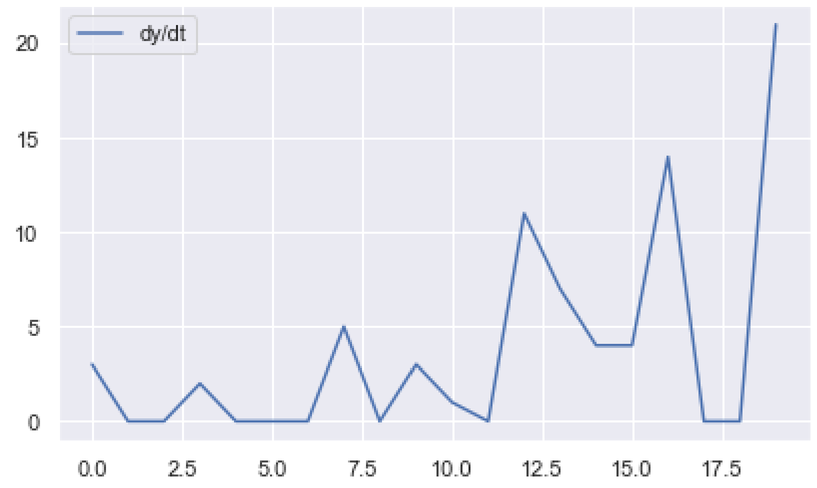


In [17]:

```
dy_dt= np.diff(covidStateDataDf["ConfirmedIndianNational"].values)
pyplot.figure(1)
pyplot.plot(dy_dt, label="dy/dt")
pyplot.legend()
```

Out[17]:

```
<matplotlib.legend.Legend at 0x1a1b9e0f90>
```

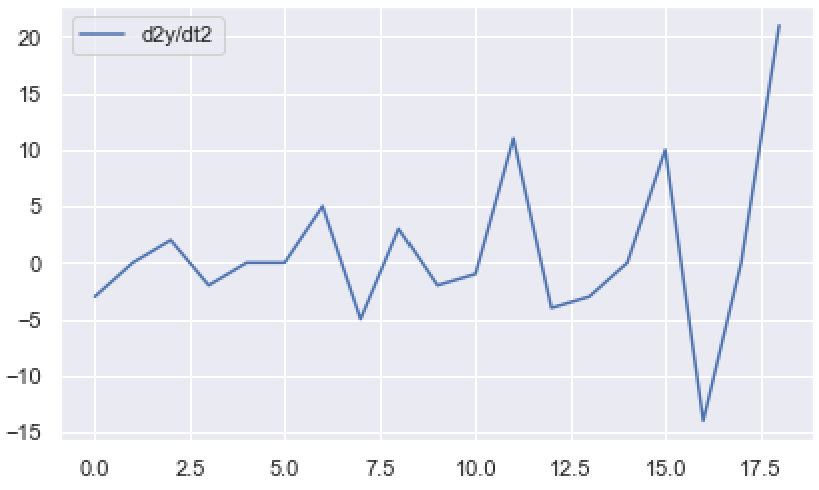


In [18]:

```
d2y_dt2= np.diff(dy_dt)
pyplot.figure(1)
pyplot.plot(d2y_dt2, label="d2y/dt2")
pyplot.legend()
```

Out[18]:

```
<matplotlib.legend.Legend at 0x1a1ba1c950>
```


d2y/dt2