Exploratory Data Analysis(EDA) of Covid 19 data in India

In [1]:
```
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import datetime as dt
import numpy as np
```

In [2]:
```
# importing main dataset
df = pd.read_csv('covid_19_india.csv',parse_dates=['Date'],dayfirst=True)
```

In [3]:
```
df.head()
```

Out[3]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
0	1	2020-01-30	6:00 PM	Kerala	1	0	0	0	1
1	2	2020-01-31	6:00 PM	Kerala	1	0	0	0	1
2	3	2020-02-01	6:00 PM	Kerala	2	0	0	0	2
3	4	2020-02-02	6:00 PM	Kerala	3	0	0	0	3
4	5	2020-02-03	6:00 PM	Kerala	3	0	0	0	3

In [4]:
```
# Keeping only required columns

df = df[['Date','State/UnionTerritory','Cured','Deaths','Confirmed']]

# Remaining column names

df.columns = ['date','state','cured','deaths','confirmed']
```

In [5]:
```
# Looking at the earlier dates

df.head()
```

Out[5]:

	date	state	cured	deaths	confirmed
0	2020-01-30	Kerala	0	0	1
1	2020-01-31	Kerala	0	0	1
2	2020-02-01	Kerala	0	0	2
3	2020-02-02	Kerala	0	0	3
4	2020-02-03	Kerala	0	0	3

In [6]:
```
# Looking for the latest dates

df.tail()
```

Out[6]:

	date	state	cured	deaths	confirmed
18105	2021-08-11	Telangana	638410	3831	650353
18106	2021-08-11	Tripura	77811	773	80660
18107	2021-08-11	Uttarakhand	334650	7368	342462
18108	2021-08-11	Uttar Pradesh	1685492	22775	1708812
18109	2021-08-11	West Bengal	1506532	18252	1534999

In [7]:
```
# Current date
today = df[df.date == '2021-08-11']
```

In [8]:
```
today.head()
```

Out[8]:

	date	state	cured	deaths	confirmed
18074	2021-08-11	Andaman and Nicobar Islands	7412	129	7548
18075	2021-08-11	Andhra Pradesh	1952736	13564	1985182
18076	2021-08-11	Arunachal Pradesh	47821	248	50605
18077	2021-08-11	Assam	559684	5420	576149
18078	2021-08-11	Bihar	715352	9646	725279

In [9]:
```
# Sorting data w.r.t. number of confirmed cases
max_confirmed_cases = today.sort_values(by='confirmed',ascending=False)
max_confirmed_cases.head()
```

Out[9]:

	date	state	cured	deaths	confirmed
18094	2021-08-11	Maharashtra	6159676	134201	6363442
18090	2021-08-11	Kerala	3396184	18004	3586693
18089	2021-08-11	Karnataka	2861499	36848	2921049
18104	2021-08-11	Tamil Nadu	2524400	34367	2579130
18075	2021-08-11	Andhra Pradesh	1952736	13564	1985182

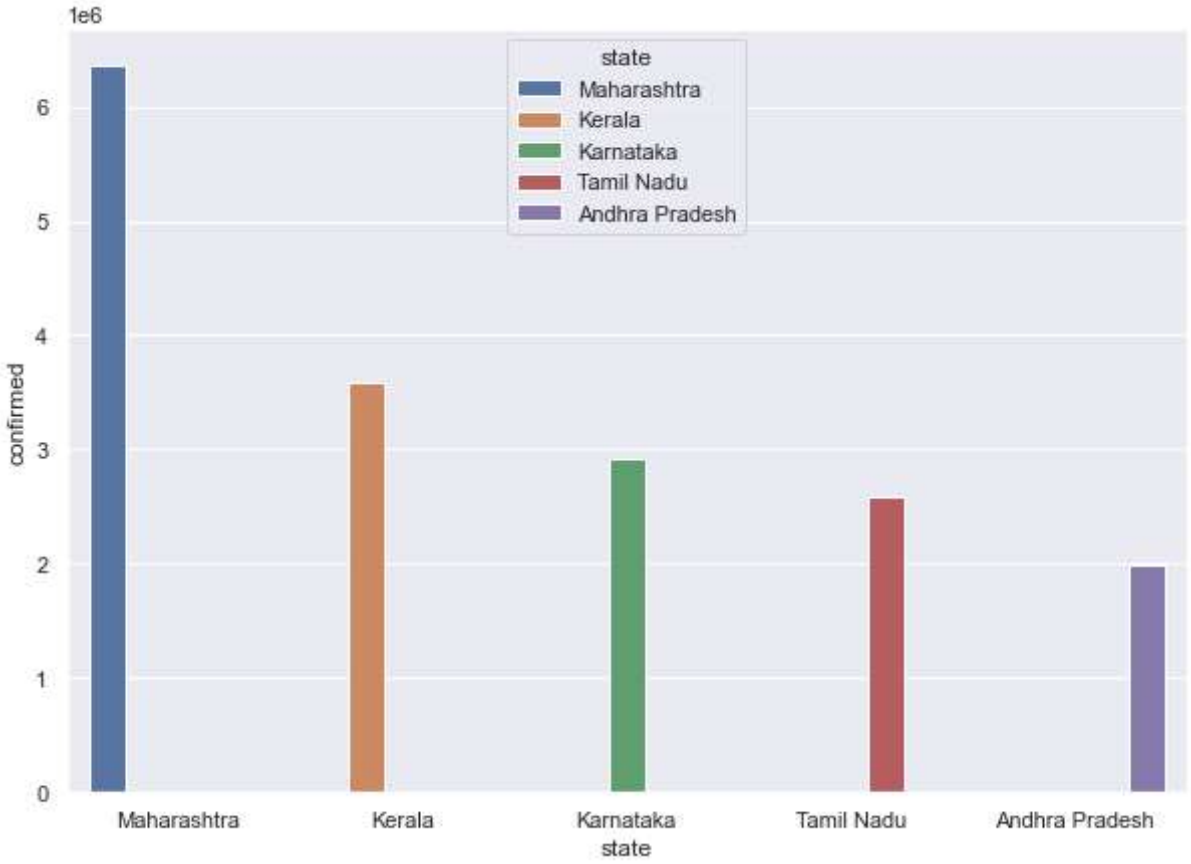
In [10]:
```
# Getting states with maximum number of confirmed cases
top_states_confirmed = max_confirmed_cases[0:5]
```

In [11]:
```
top_states_confirmed
```

Out[11]:

	date	state	cured	deaths	confirmed
18094	2021-08-11	Maharashtra	6159676	134201	6363442
18090	2021-08-11	Kerala	3396184	18004	3586693
18089	2021-08-11	Karnataka	2861499	36848	2921049
18104	2021-08-11	Tamil Nadu	2524400	34367	2579130
18075	2021-08-11	Andhra Pradesh	1952736	13564	1985182

In [12]:
```
# Making bar plot or states with top confirmed cases
sns.set(rc={'figure.figsize':(10,7)})
sns.barplot(x="state",y="confirmed",data=top_states_confirmed,hue = "state")
plt.show()
```



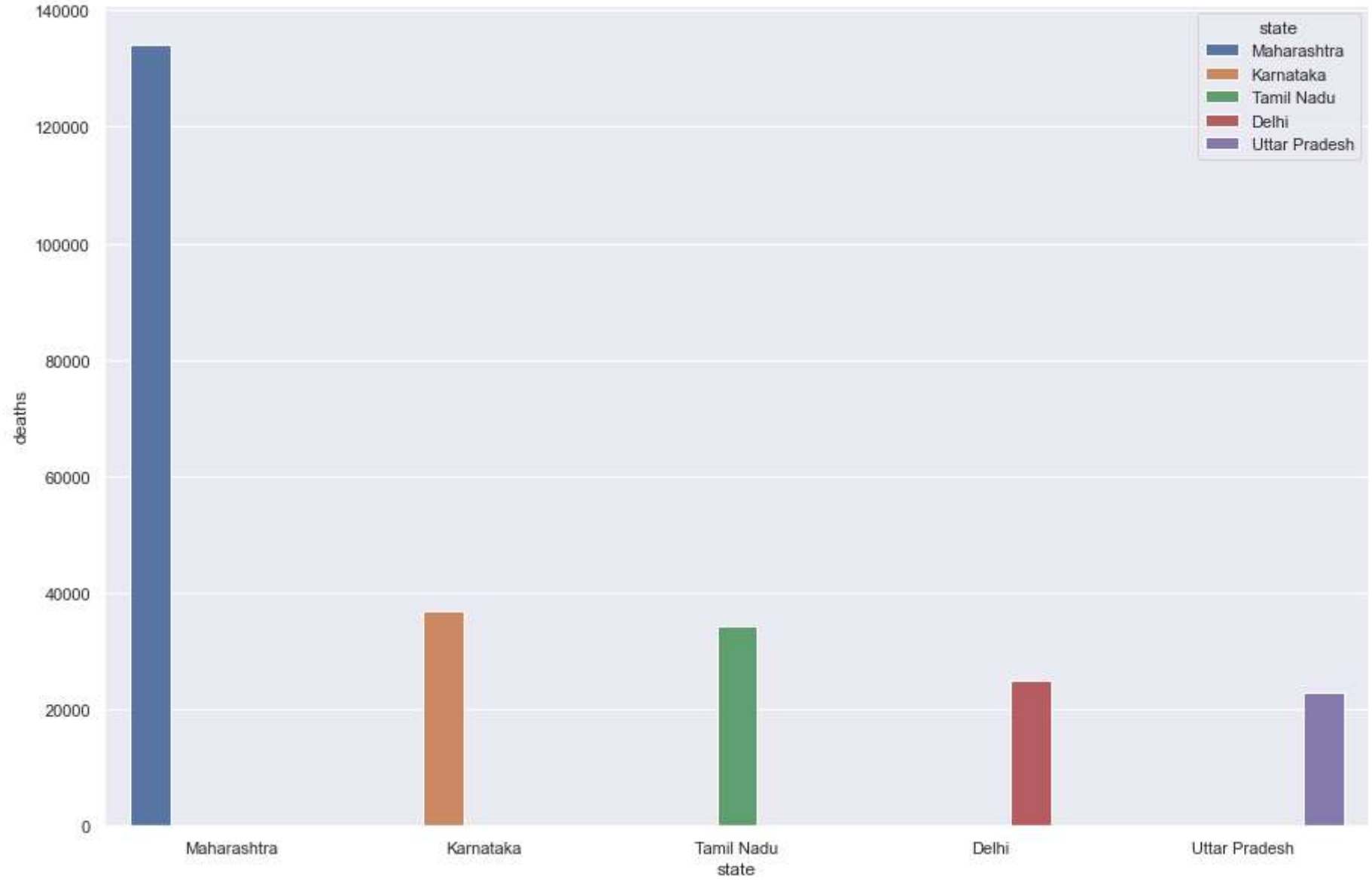
In [13]:
```
# Sorting data w.r.t. number of deaths cases
max_death_cases=today.sort_values(by="deaths",ascending=False)
max_death_cases.head()
```

Out[13]:

	date	state	cured	deaths	confirmed
18094	2021-08-11	Maharashtra	6159676	134201	6363442
18089	2021-08-11	Karnataka	2861499	36848	2921049
18104	2021-08-11	Tamil Nadu	2524400	34367	2579130
18082	2021-08-11	Delhi	1411280	25068	1436852
18108	2021-08-11	Uttar Pradesh	1685492	22775	1708812

In [14]:
```
# Getting states with maximum number of death cases
top_states_death = max_death_cases[0:5]
```

In [15]:
```
# Making bar plot for states with top death cases
sns.set(rc={'figure.figsize':(15,10)})
sns.barplot(x="state",y="deaths",data=top_states_death,hue="state")
plt.show()
```



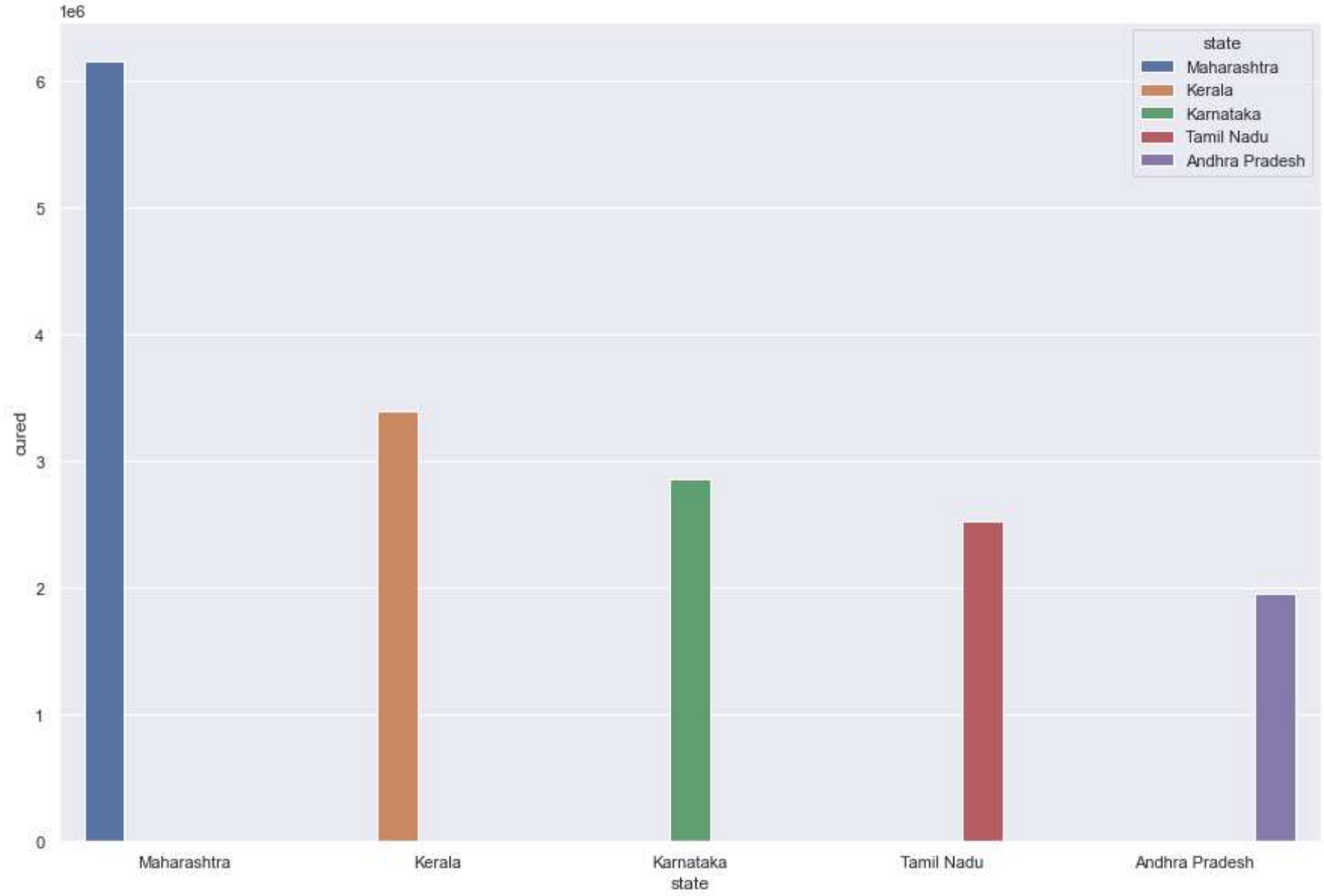
In [16]:
```
# Sorting data w.r.t. number of cured cases
max_cured_cases = today.sort_values(by="cured",ascending=False)
max_cured_cases.head()
```

Out[16]:

	date	state	cured	deaths	confirmed
18094	2021-08-11	Maharashtra	6159676	134201	6363442
18090	2021-08-11	Kerala	3396184	18004	3586693
18089	2021-08-11	Karnataka	2861499	36848	2921049
18104	2021-08-11	Tamil Nadu	2524400	34367	2579130
18075	2021-08-11	Andhra Pradesh	1952736	13564	1985182

In [17]:
```
# Getting states with maximum number of cases
top_states_cured = max_cured_cases[0:5]
```

In [18]:
```
# Making bar plot for states with top death cases
sns.set(rc={'figure.figsize':[15,10]})
sns.barplot(x="state",y="cured",data = top_states_cured,hue="state")
plt.show()
```



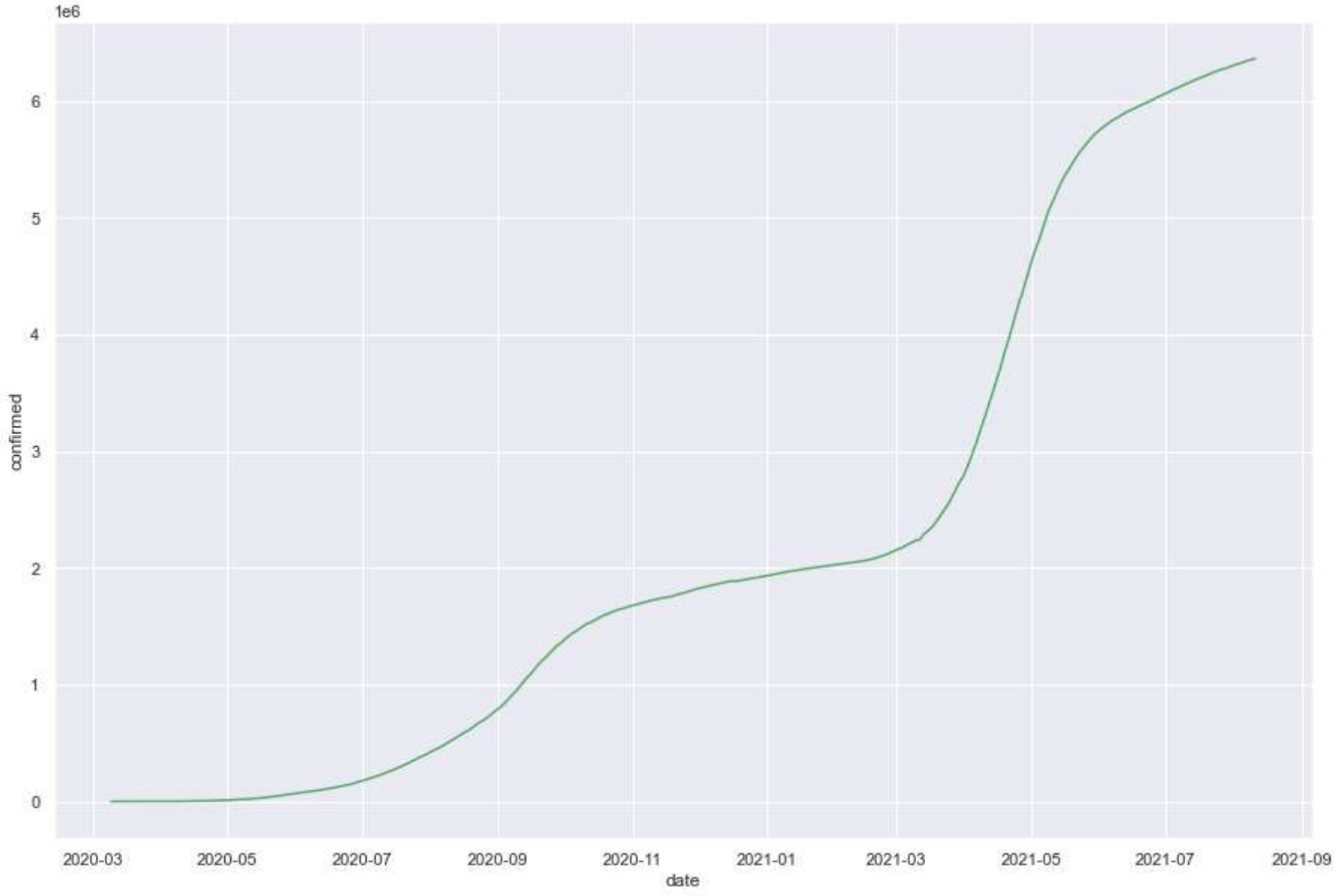
In [20]:
```
# Maharashtra
maha = df[df.state == 'Maharashtra']
maha
```

Out[20]:

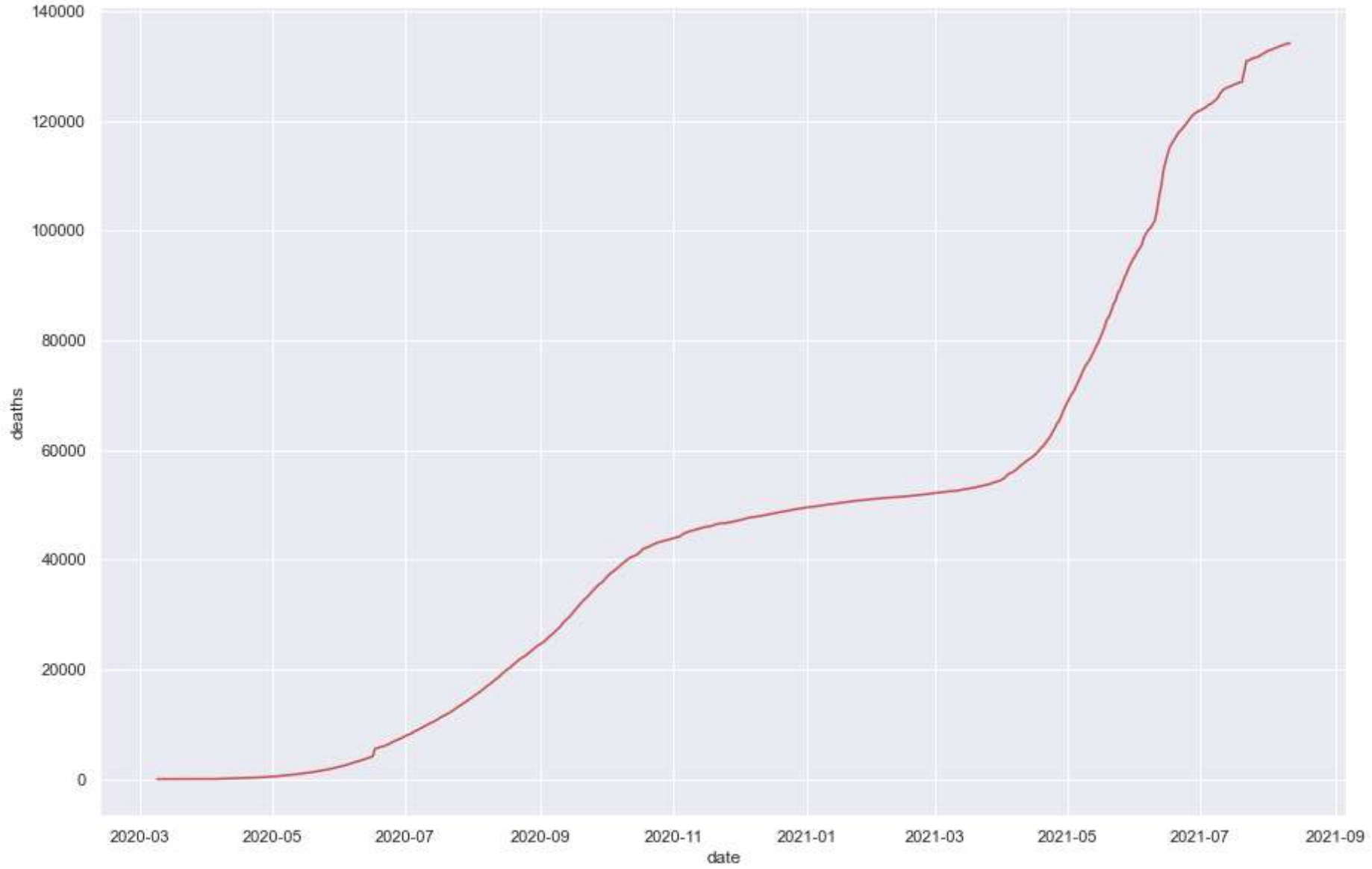
	date	state	cured	deaths	confirmed
76	2020-03-09	Maharashtra	0	0	2
91	2020-03-10	Maharashtra	0	0	5
97	2020-03-11	Maharashtra	0	0	2
120	2020-03-12	Maharashtra	0	0	11
133	2020-03-13	Maharashtra	0	0	14
...	...	...	...	...	...
17950	2021-08-07	Maharashtra	6130137	133717	6341759
17986	2021-08-08	Maharashtra	6139493	133845	6347820
18022	2021-08-09	Maharashtra	6144388	133996	6353328
18058	2021-08-10	Maharashtra	6151956	134064	6357833
18094	2021-08-11	Maharashtra	6159676	134201	6363442

520 rows × 5 columns

In [21]:
```
# Visualising confirmed cases in maharashtra
sns.set(rc={'figure.figsize':[15,10]})
sns.lineplot(x="date",y="confirmed",data=maha,color="g")
plt.show()
```



In [22]:
```
# Visualising death cases in maharashtra
sns.set(rc={'figure.figsize':[15,10]})
sns.lineplot(x="date",y="deaths",data=maha,color="r")
plt.show()
```



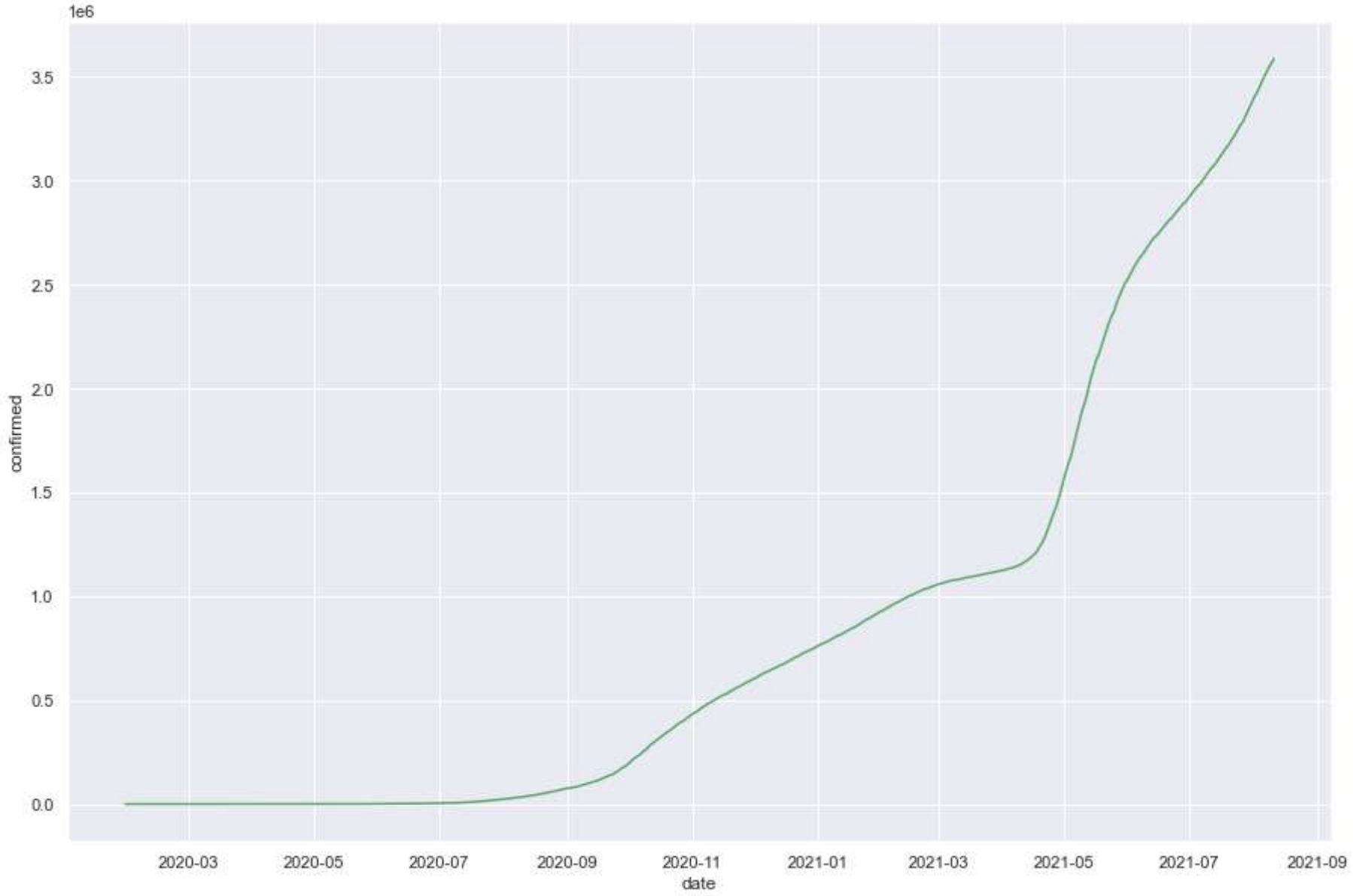
In [26]:
```
# Kerala
kerala = df[df.state == 'Kerala']
kerala
```

Out[26]:

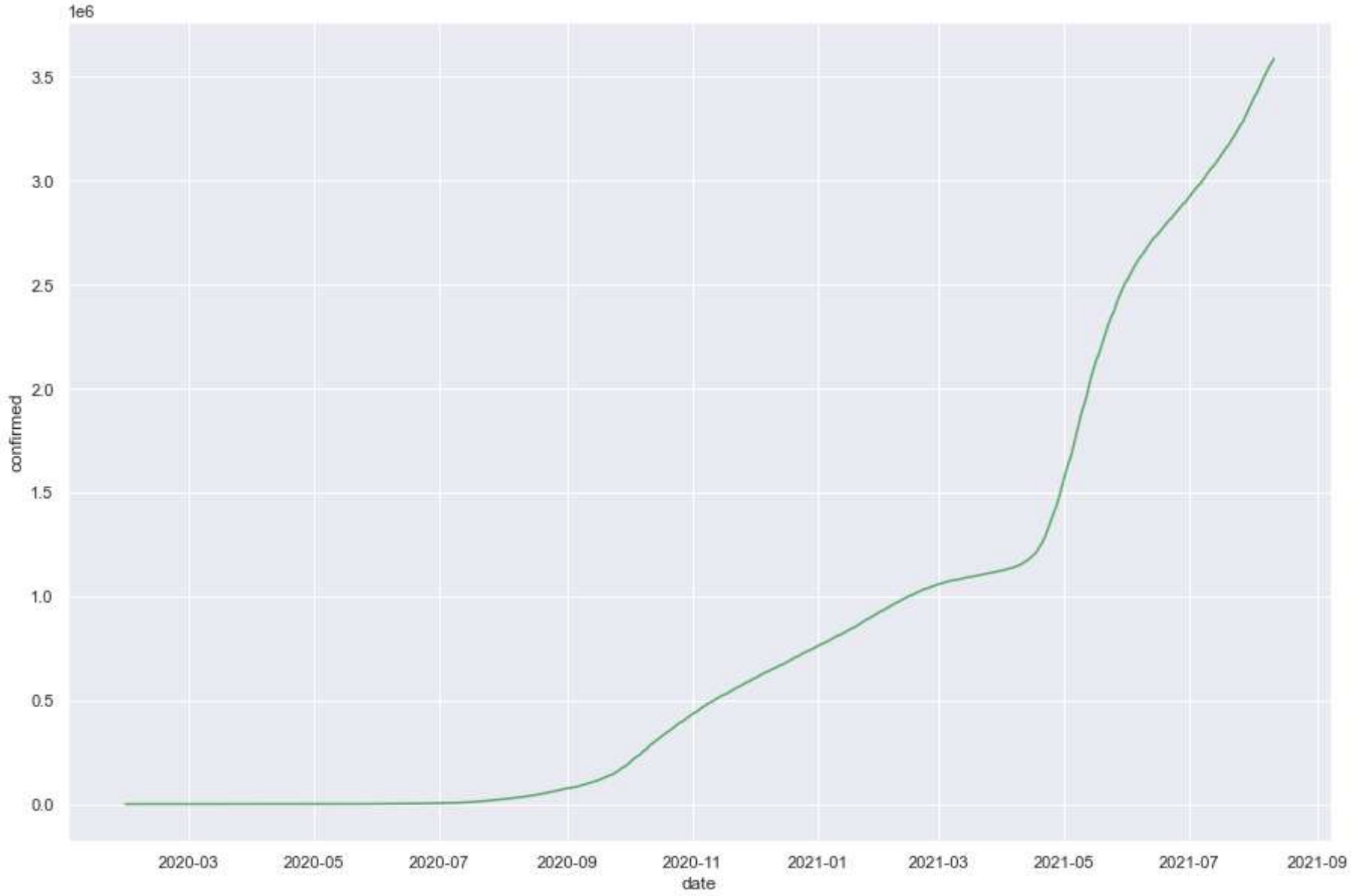
	date	state	cured	deaths	confirmed
0	2020-01-30	Kerala	0	0	1
1	2020-01-31	Kerala	0	0	1
2	2020-02-01	Kerala	0	0	2
3	2020-02-02	Kerala	0	0	3
4	2020-02-03	Kerala	0	0	3
...	...	...	...	...	...
17946	2021-08-07	Kerala	3317314	17515	3513551
17982	2021-08-08	Kerala	3337579	17654	3533918
18018	2021-08-09	Kerala	3357687	17747	3552525
18054	2021-08-10	Kerala	3377691	17852	3565574
18090	2021-08-11	Kerala	3396184	18004	3586693

560 rows × 5 columns

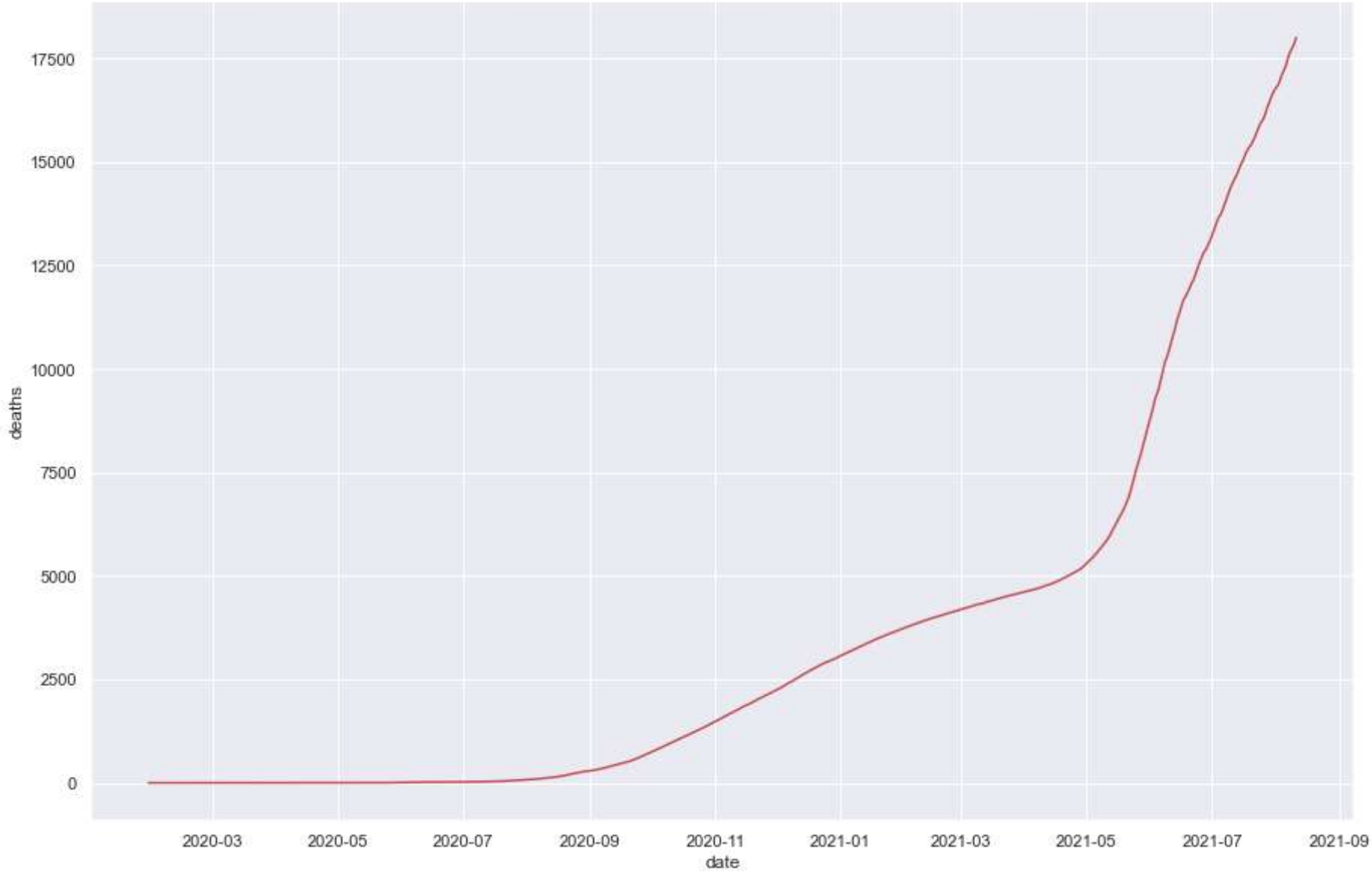
In [27]:
```
# Visualising confirmed cases in kerala
sns.set(rc={'figure.figsize':[15,10]})
sns.lineplot(x="date",y="confirmed",data=kerala,color="g")
plt.show()
```



In [29]:
```
# Visualising confirmed cases in kerala
sns.set(rc={'figure.figsize':[15,10]})
sns.lineplot(x="date",y="confirmed",data=kerala,color="g")
plt.show()
```



In [31]:
```
# Visualising death cases in Kerala
sns.set(rc={'figure.figsize':[15,10]})
sns.lineplot(x="date",y="deaths",data=kerala,color="r")
plt.show()
```



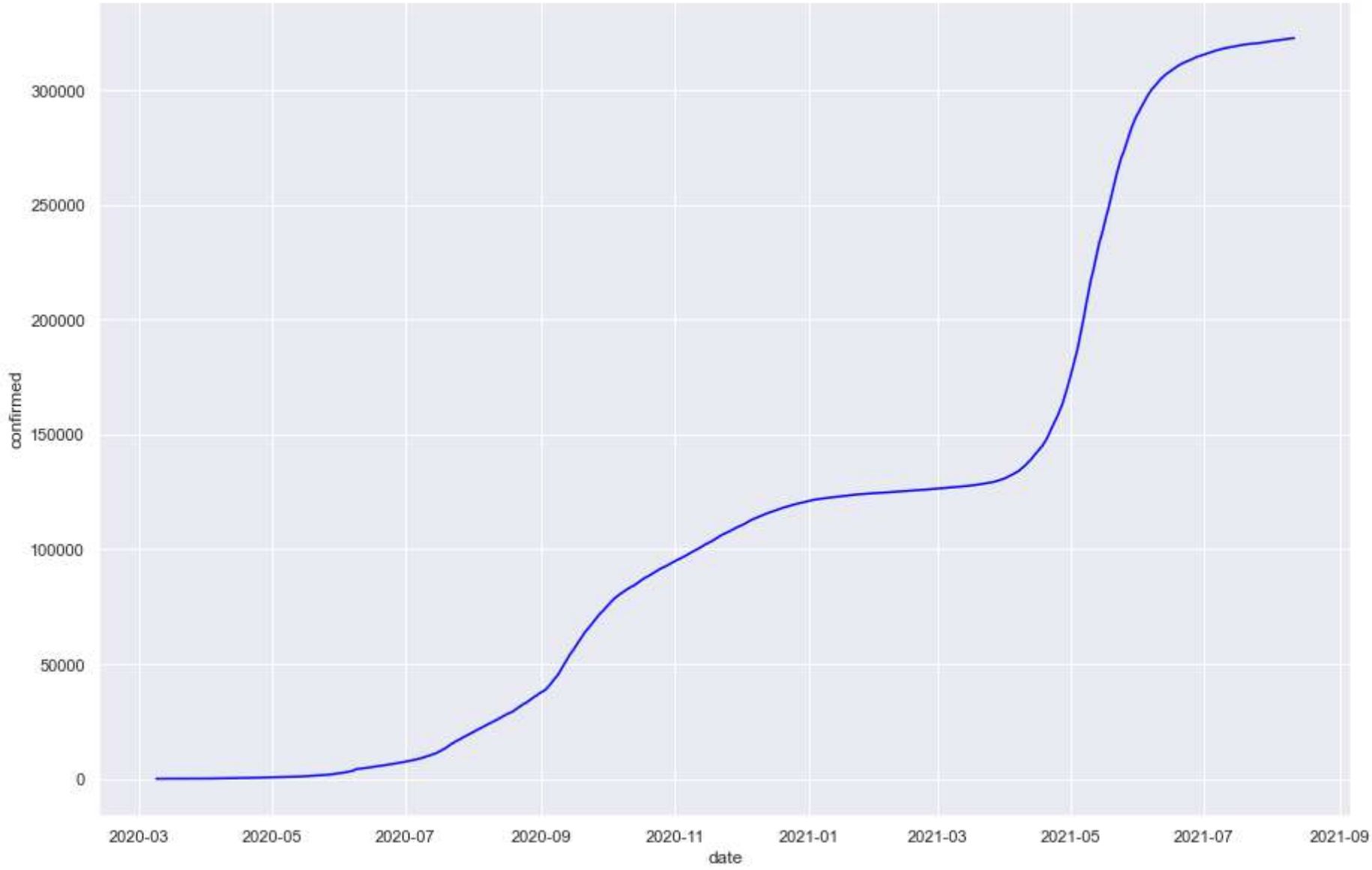
In [32]:
```
# Jammu and Kashmir
jk = df[df.state == 'Jammu and Kashmir']
jk
```

Out[32]:

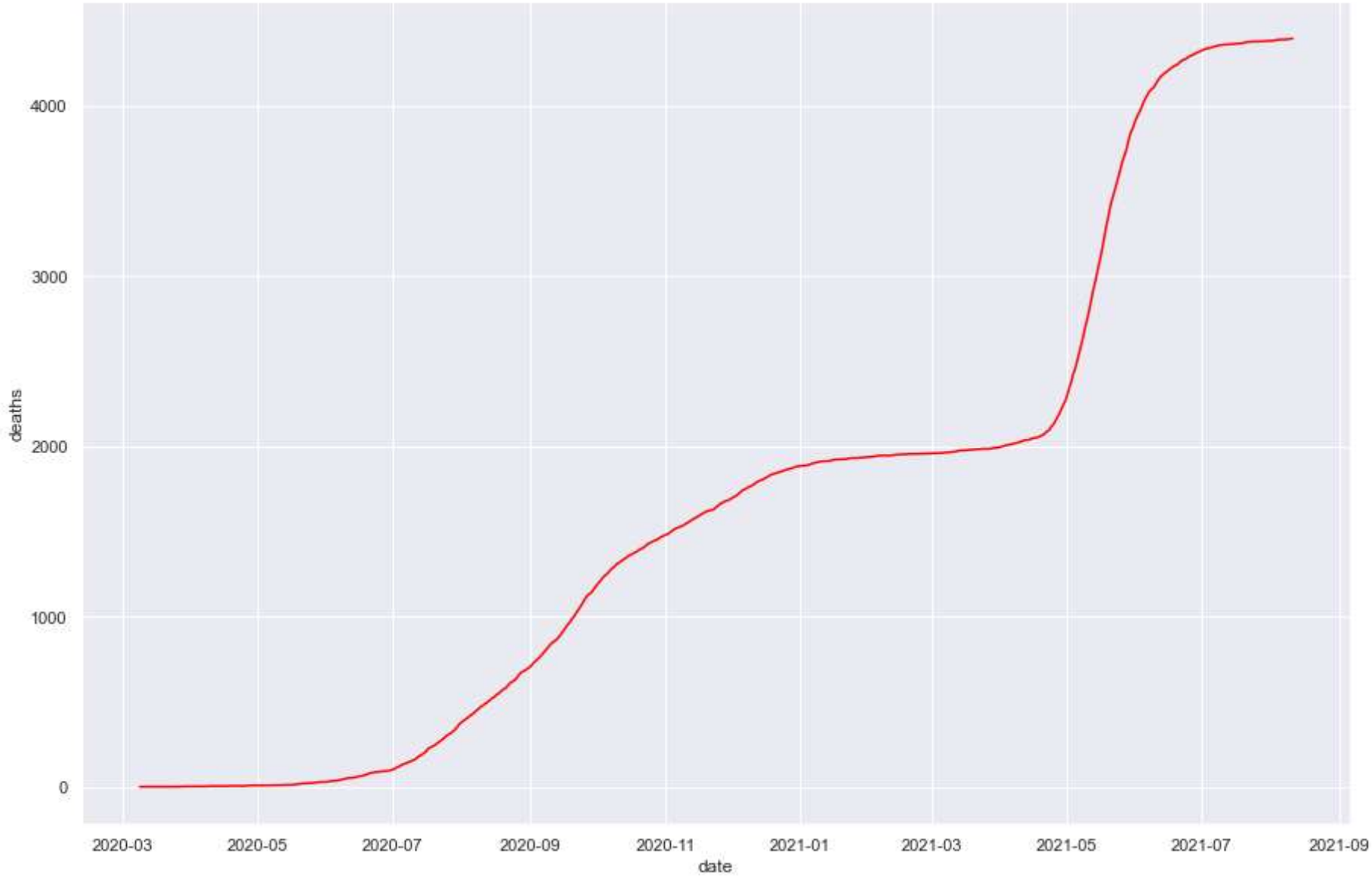
	date	state	cured	deaths	confirmed
81	2020-03-09	Jammu and Kashmir	0	0	1
96	2020-03-10	Jammu and Kashmir	0	0	1
106	2020-03-11	Jammu and Kashmir	0	0	1
117	2020-03-12	Jammu and Kashmir	0	0	1
130	2020-03-13	Jammu and Kashmir	0	0	1
...	...	...	...	...	...
17943	2021-08-07	Jammu and Kashmir	316496	4386	322286
17979	2021-08-08	Jammu and Kashmir	316632	4386	322428
18015	2021-08-09	Jammu and Kashmir	316761	4389	322550
18051	2021-08-10	Jammu and Kashmir	316957	4390	322658
18087	2021-08-11	Jammu and Kashmir	317081	4392	322771

521 rows × 5 columns

In [34]:
```
# Visualising confirmed cases in Jammu and kashmir
sns.set(rc={'figure.figsize':[15,10]})
sns.lineplot(x="date",y="confirmed",data=jk,color="blue")
plt.show()
```



In [35]:
```
# Visualising death cases in Jammu and Kashmir
sns.set(rc={'figure.figsize':[15,10]})
sns.lineplot(x="date",y="deaths",data=jk,color="red")
plt.show()
```



In [36]:
```
# Checking state wise testing details
tests = pd.read_csv('StatewiseTestingDetails.csv')
tests
```

Out[36]:

	Date	State	TotalSamples	Negative	Positive
0	2020-04-17	Andaman and Nicobar Islands	1403.0	1210	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	2020-05-16	Andaman and Nicobar Islands	6677.0	NaN	33.0
...	...	...	...	...	...
16331	2021-08-06	West Bengal	15999961.0	NaN	NaN
16332	2021-08-07	West Bengal	16045662.0	NaN	NaN
16333	2021-08-08	West Bengal	16092192.0	NaN	NaN
16334	2021-08-09	West Bengal	16122345.0	NaN	NaN
16335	2021-08-10	West Bengal	16162814.0	NaN	NaN

16336 rows × 5 columns

In [38]:
```
test_latest = tests[tests.Date == '2021-08-10']
test_latest
```

Out[38]:

	Date	State	TotalSamples	Negative	Positive
940	2021-08-10	Andhra Pradesh	25311733.0	23326551	NaN
1417	2021-08-10	Arunachal Pradesh	986281.0	NaN	NaN
1886	2021-08-10	Assam	19850867.0	NaN	NaN
2375	2021-08-10	Bihar	38820518.0	NaN	NaN
2854	2021-08-10	Chandigarh	629060.0	565758	NaN
3336	2021-08-10	Chhattisgarh	11762041.0	NaN	NaN
3995	2021-08-10	Delhi	24333906.0	NaN	NaN
4478	2021-08-10	Goa	1102474.0	NaN	NaN
4965	2021-08-10	Gujarat	26192626.0	NaN	NaN
5457	2021-08-10	Haryana	11135555.0	NaN	NaN
5945	2021-08-10	Himachal Pradesh	2961627.0	2752249	NaN
6434	2021-08-10	Jammu and Kashmir	12307566.0	11984795	NaN
6918	2021-08-10	Jharkhand	12184347.0	11836897	NaN
7409	2021-08-10	Karnataka	40104915.0	NaN	NaN
7906	2021-08-10	Kerala	28745545.0	NaN	NaN
8395	2021-08-10	Lakshadweep	226724.0	NaN	NaN
8887	2021-08-10	Madhya Pradesh	15144644.0	NaN	NaN
9375	2021-08-10	Maharashtra	49905065.0	NaN	NaN
9781	2021-08-10	Manipur	1136573.0	NaN	NaN
10190	2021-08-10	Meghalaya	894820.0	825051	NaN
10655	2021-08-10	Mizoram	688280.0	NaN	NaN
11139	2021-08-10	Nagaland	280777.0	NaN	NaN
11631	2021-08-10	Odisha	16683764.0	NaN	NaN
12109	2021-08-10	Puducherry	1557320.0	1326325	NaN

	Date	State	TotalSamples	Negative	Positive
12600	2021-08-10	Punjab	12475529.0	NaN	NaN
13091	2021-08-10	Rajasthan	13185136.0	NaN	NaN
13504	2021-08-10	Sikkim	213375.0	NaN	NaN
13995	2021-08-10	Tamil Nadu	39002757.0	NaN	NaN
14414	2021-08-10	Telangana	22991849.0	NaN	NaN
14861	2021-08-10	Tripura	1630572.0	1550159	80413.0
15351	2021-08-10	Uttar Pradesh	67897856.0	NaN	NaN
15842	2021-08-10	Uttarakhand	6526861.0	6184399	NaN
16335	2021-08-10	West Bengal	16162814.0	NaN	NaN

In [39]:
```
# Linear Regression
from sklearn.model_selection import train_test_split
```

In [40]:
```
maha
```

Out[40]:

	date	state	cured	deaths	confirmed
76	2020-03-09	Maharashtra	0	0	2
91	2020-03-10	Maharashtra	0	0	5
97	2020-03-11	Maharashtra	0	0	2
120	2020-03-12	Maharashtra	0	0	11
133	2020-03-13	Maharashtra	0	0	14
...	...	...	...	...	...
17950	2021-08-07	Maharashtra	6130137	133717	6341759
17986	2021-08-08	Maharashtra	6139493	133845	6347820
18022	2021-08-09	Maharashtra	6144388	133996	6353328
18058	2021-08-10	Maharashtra	6151956	134064	6357833
18094	2021-08-11	Maharashtra	6159676	134201	6363442

520 rows × 5 columns

In [41]:
```
# Converting date-time to ordinal
maha['date']=maha['date'].map(dt.datetime.toordinal)
maha.head()
```

```
C:\Users\MOHD. RAEES\AppData\Local\Temp\ipykernel_9156\3352117026.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  maha['date']=maha['date'].map(dt.datetime.toordinal)
```

Out[41]:

	date	state	cured	deaths	confirmed
76	737493	Maharashtra	0	0	2
91	737494	Maharashtra	0	0	5
97	737495	Maharashtra	0	0	2
120	737496	Maharashtra	0	0	11
133	737497	Maharashtra	0	0	14

In [42]:
```
# Getting dependent variable and independent variable
x = maha['date']
y = maha['confirmed']
```

In [44]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [50]:
```
from sklearn.linear_model import LinearRegression
```

In [51]:
```
lr = LinearRegression()
```

In [52]:
```
y_train
```

Out[52]:
```
10354    1958282
4550      375799
7280     1535315
16798    6104917
4340      318695
          ...
5180      560126
13918    3703584
6440     1167496
16978    6149264
1175        4669
Name: confirmed, Length: 364, dtype: int64
```

In [57]:
```
lr.fit(np.array(x_train).reshape(-1,1),np.array(y_train).reshape(-1,1))
```

Out[57]:
```
LinearRegression()
```

In [58]:
```
maha.tail()
```

Out[58]:

	date	state	cured	deaths	confirmed
17950	738009	Maharashtra	6130137	133717	6341759
17986	738010	Maharashtra	6139493	133845	6347820
18022	738011	Maharashtra	6144388	133996	6353328
18058	738012	Maharashtra	6151956	134064	6357833
18094	738013	Maharashtra	6159676	134201	6363442

In [55]:
```
lr.predict(np.array([[737625]]))
```

Out[55]:
```
array([[525312.36777115]])
```

Analyzed by

Md Raiesh, Enrollment number : **19UME116**, Registration number : **1911345**, B Tech,7**th** semester,Section : **A**, Mechanical Engineering Department, National Institute of Technology Agartala, Tripura 799046,

In [ ]: