

```
In [1]: #Loading the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set(color_codes=True)
```

```
In [2]: #Loading the required dataset
covid = pd.read_csv('covid.csv')
```

```
In [3]: #Having a glance at some of the records
covid.head()
```

```
Out[3]:
```

	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	new_ca
0	ABW	Aruba	2020-03-13	2	2	0	0	18.733	
1	ABW	Aruba	2020-03-20	4	2	0	0	37.465	
2	ABW	Aruba	2020-03-24	12	8	0	0	112.395	
3	ABW	Aruba	2020-03-25	17	5	0	0	159.227	
4	ABW	Aruba	2020-03-26	19	2	0	0	177.959	

5 rows × 32 columns

```
In [4]: #Looking at the shape
covid.shape
```

```
Out[4]: (19496, 32)
```

```
In [5]: covid.columns
```

```
Out[5]: Index(['iso_code', 'location', 'date', 'total_cases', 'new_cases',
            'total_deaths', 'new_deaths', 'total_cases_per_million',
            'new_cases_per_million', 'total_deaths_per_million',
            'new_deaths_per_million', 'total_tests', 'new_tests',
            'total_tests_per_thousand', 'new_tests_per_thousand',
            'new_tests_smoothed', 'new_tests_smoothed_per_thousand', 'tests_units',
            'stringency_index', 'population', 'population_density', 'median_age',
            'aged_65_older', 'aged_70_older', 'gdp_per_capita', 'extreme_poverty',
            'cvd_death_rate', 'diabetes_prevalence', 'female_smokers',
            'male_smokers', 'handwashing_facilities', 'hospital_beds_per_100k'],
            dtype='object')
```

```
In [6]: #Looking at the different locations
covid["location"].value_counts()
```

```
Out[6]: Sweden      146
Canada      146
United States  146
Lithuania    146
Brazil       146
...
```

```

Yemen          45
Western Sahara  29
Tajikistan     24
Comoros        23
Lesotho        10
Name: location, Length: 212, dtype: int64

```

```

In [7]: #Checking if columns have null values
        covid.isna().any()

```

```

Out[7]: iso_code          True
        location         False
        date            False
        total_cases      False
        new_cases        False
        total_deaths     False
        new_deaths       False
        total_cases_per_million  True
        new_cases_per_million   True
        total_deaths_per_million True
        new_deaths_per_million  True
        total_tests       True
        new_tests         True
        total_tests_per_thousand True
        new_tests_per_thousand  True
        new_tests_smoothed  True
        new_tests_smoothed_per_thousand True
        tests_units       True
        stringency_index   True
        population        True
        population_density  True
        median_age        True
        aged_65_older     True
        aged_70_older     True
        gdp_per_capita    True
        extreme_poverty   True
        cvd_death_rate    True
        diabetes_prevalence True
        female_smokers     True
        male_smokers       True
        handwashing_facilities True
        hospital_beds_per_100k True
        dtype: bool

```

```

In [8]: #Getting the sum of null values across each column
        covid.isna().sum()

```

```

Out[8]: iso_code          64
        location          0
        date              0
        total_cases       0
        new_cases         0
        total_deaths      0
        new_deaths        0
        total_cases_per_million  377
        new_cases_per_million   377
        total_deaths_per_million 377
        new_deaths_per_million  377
        total_tests       14332
        new_tests         14904
        total_tests_per_thousand 14332
        new_tests_per_thousand  14904
        new_tests_smoothed  13866
        new_tests_smoothed_per_thousand 13866
        tests_units       13267
        stringency_index   4500
        population         64
        population_density   850

```

```

median_age      1743
aged_65_older   1980
aged_70_older   1832
gdp_per_capita  1982
extreme_poverty  7878
cvd_death_rate  1817
diabetes_prevalence 1174
female_smokers   5052
male_smokers     5206
handwashing_facilities 11822
hospital_beds_per_100k 3160
dtype: int64

```

```

In [9]: #Getting the cases in India
india_case=covid[covid["location"]=="India"]

```

```

In [10]: india_case.head()

```

```

Out[10]:

```

	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	new
8379	IND	India	2019-12-31	0	0	0	0	0.0	
8380	IND	India	2020-01-01	0	0	0	0	0.0	
8381	IND	India	2020-01-02	0	0	0	0	0.0	
8382	IND	India	2020-01-03	0	0	0	0	0.0	
8383	IND	India	2020-01-04	0	0	0	0	0.0	

5 rows × 32 columns

```

In [11]: india_case.tail()

```

```

Out[11]:

```

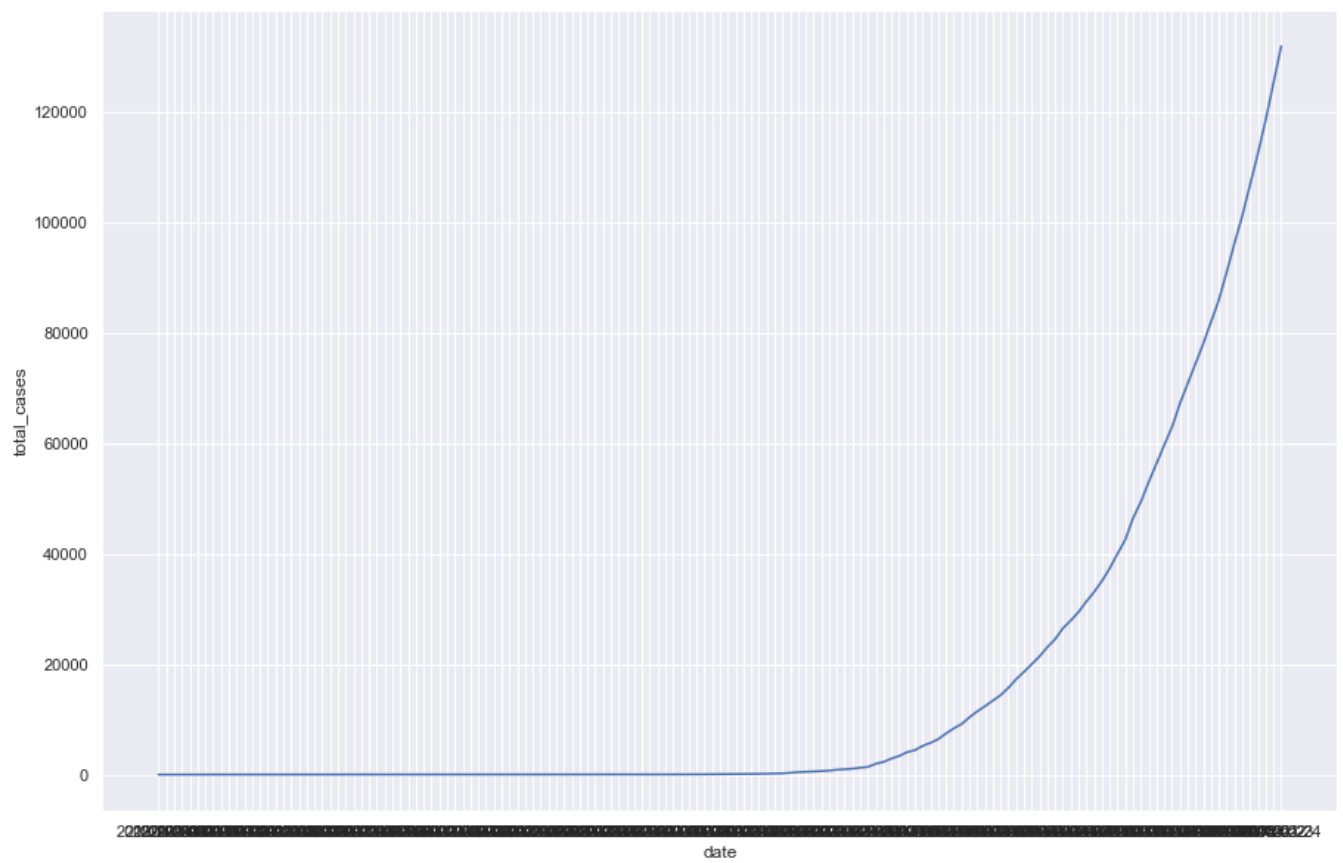
	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	new
8519	IND	India	2020-05-20	106750	5611	3303	140	77.355	
8520	IND	India	2020-05-21	112359	5609	3435	132	81.419	
8521	IND	India	2020-05-22	118447	6088	3583	148	85.831	
8522	IND	India	2020-05-23	125101	6654	3720	137	90.653	
8523	IND	India	2020-05-24	131868	6767	3867	147	95.556	

5 rows × 32 columns

```

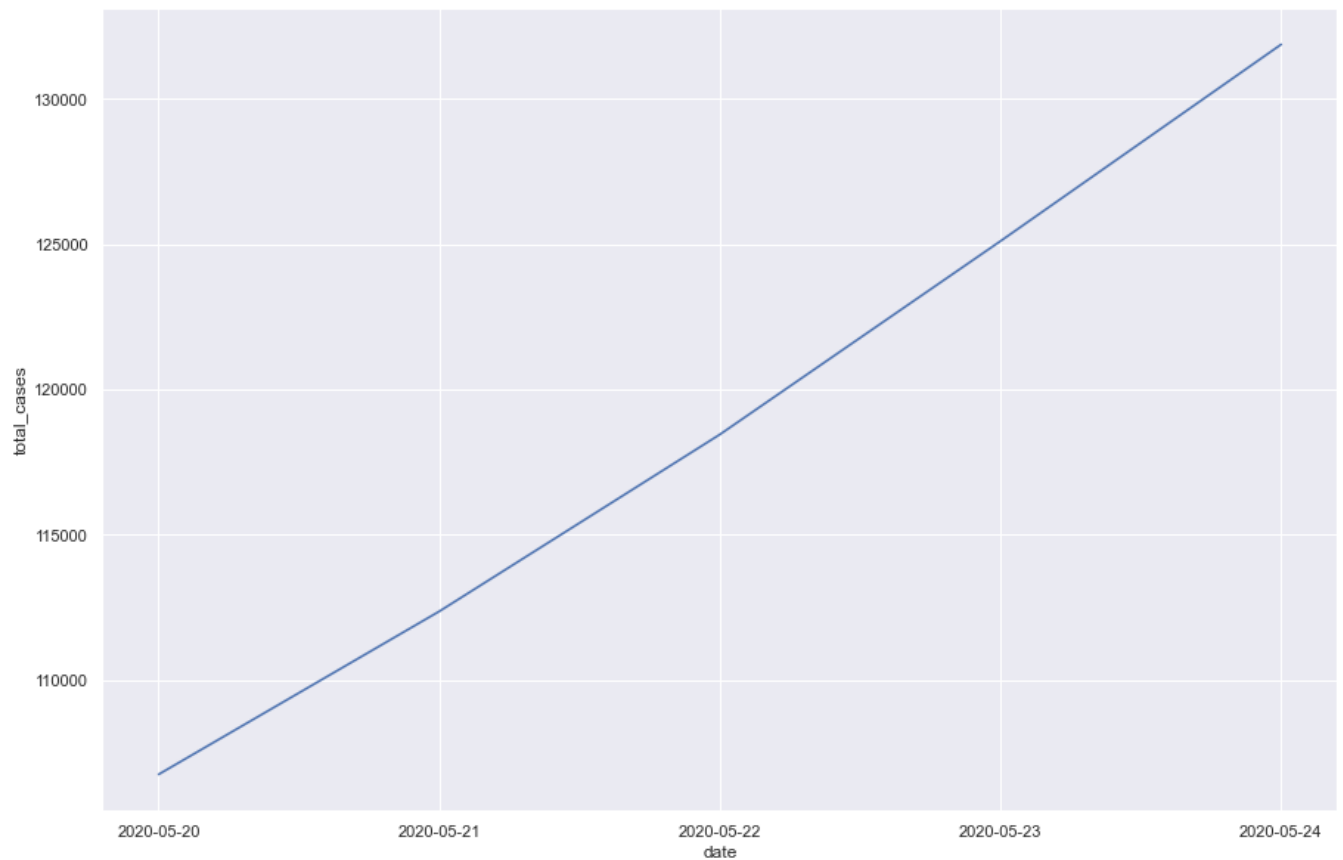
In [12]: #Total cases per day
sns.set(rc={'figure.figsize':(15,10)})
sns.lineplot(x="date",y="total_cases",data=india_case)
plt.show()

```



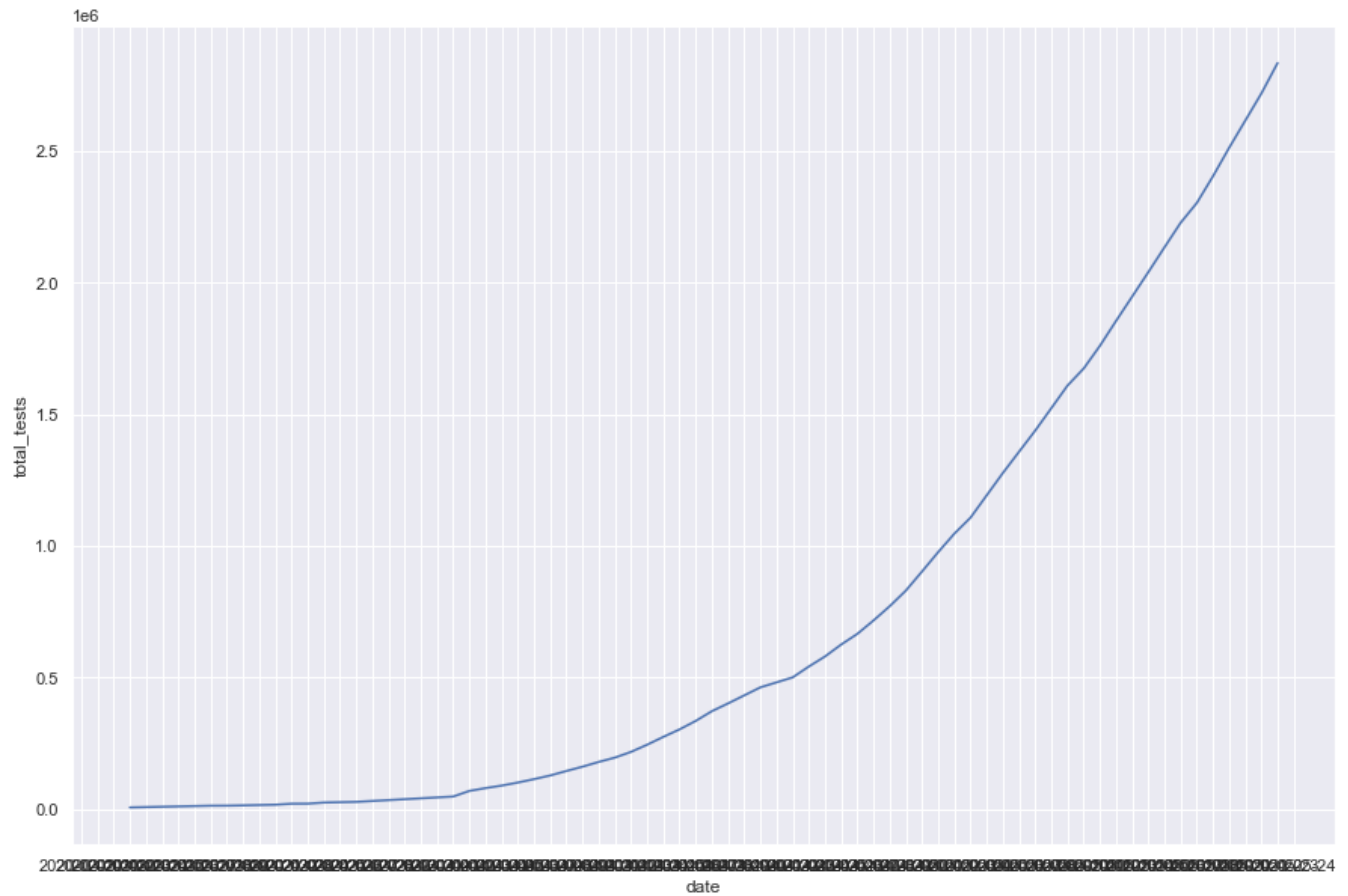
```
In [13]: #Making a dataframe for last 5 days
india_last_5_days=india_case.tail()
```

```
In [15]: #Total cases in last 5 days
sns.set(rc={'figure.figsize': (15,10)})
sns.lineplot(x="date",y="total_cases",data=india_last_5_days)
plt.show()
```



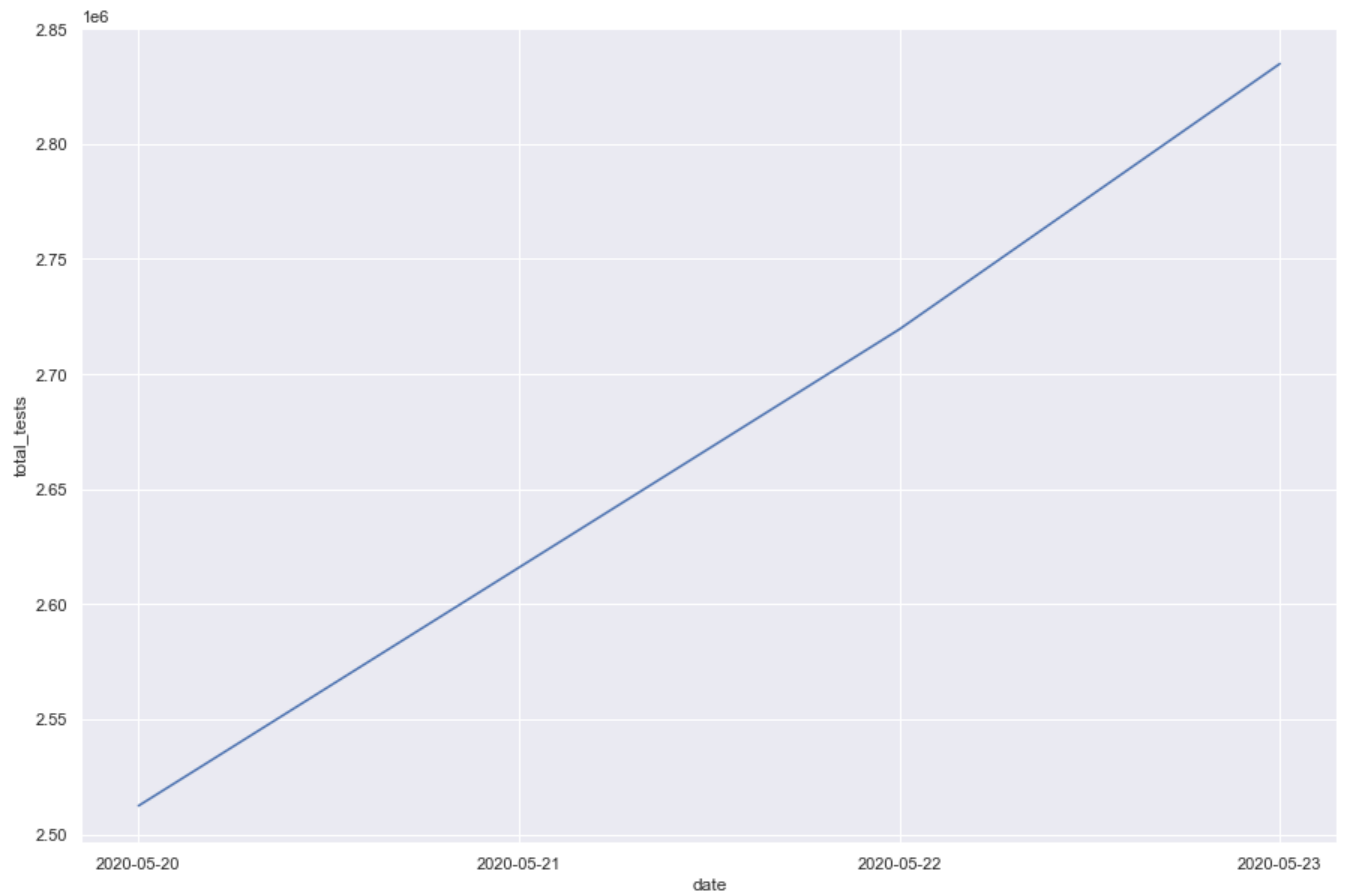
```
In [16]: #Total tests per day
sns.set(rc={'figure.figsize': (15,10)})
```

```
sns.lineplot(x="date", y="total_tests", data=india_case)
plt.show()
```



In [17]:

```
#Total tests in last 5 days
sns.set(rc={'figure.figsize': (15,10)})
sns.lineplot(x="date", y="total_tests", data=india_last_5_days)
plt.show()
```



In [18]:

```
#Brazil Case
```

```
brazil_case=covid[covid["location"]=="Brazil"]
brazil_case.head()
```

Out[18]:

	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	nev
2510	BRA	Brazil	2019-12-31	0	0	0	0	0.0	
2511	BRA	Brazil	2020-01-01	0	0	0	0	0.0	
2512	BRA	Brazil	2020-01-02	0	0	0	0	0.0	
2513	BRA	Brazil	2020-01-03	0	0	0	0	0.0	
2514	BRA	Brazil	2020-01-04	0	0	0	0	0.0	

5 rows × 32 columns

In [19]:

```
brazil_case.tail()
```

Out[19]:

	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	nev
2651	BRA	Brazil	2020-05-20	271628	17408	17971	1179	1277.892	
2652	BRA	Brazil	2020-05-21	291579	19951	18859	888	1371.753	
2653	BRA	Brazil	2020-05-22	310087	18508	20047	1188	1458.825	
2654	BRA	Brazil	2020-05-23	330890	20803	21048	1001	1556.694	
2655	BRA	Brazil	2020-05-24	347398	16508	22013	965	1634.357	

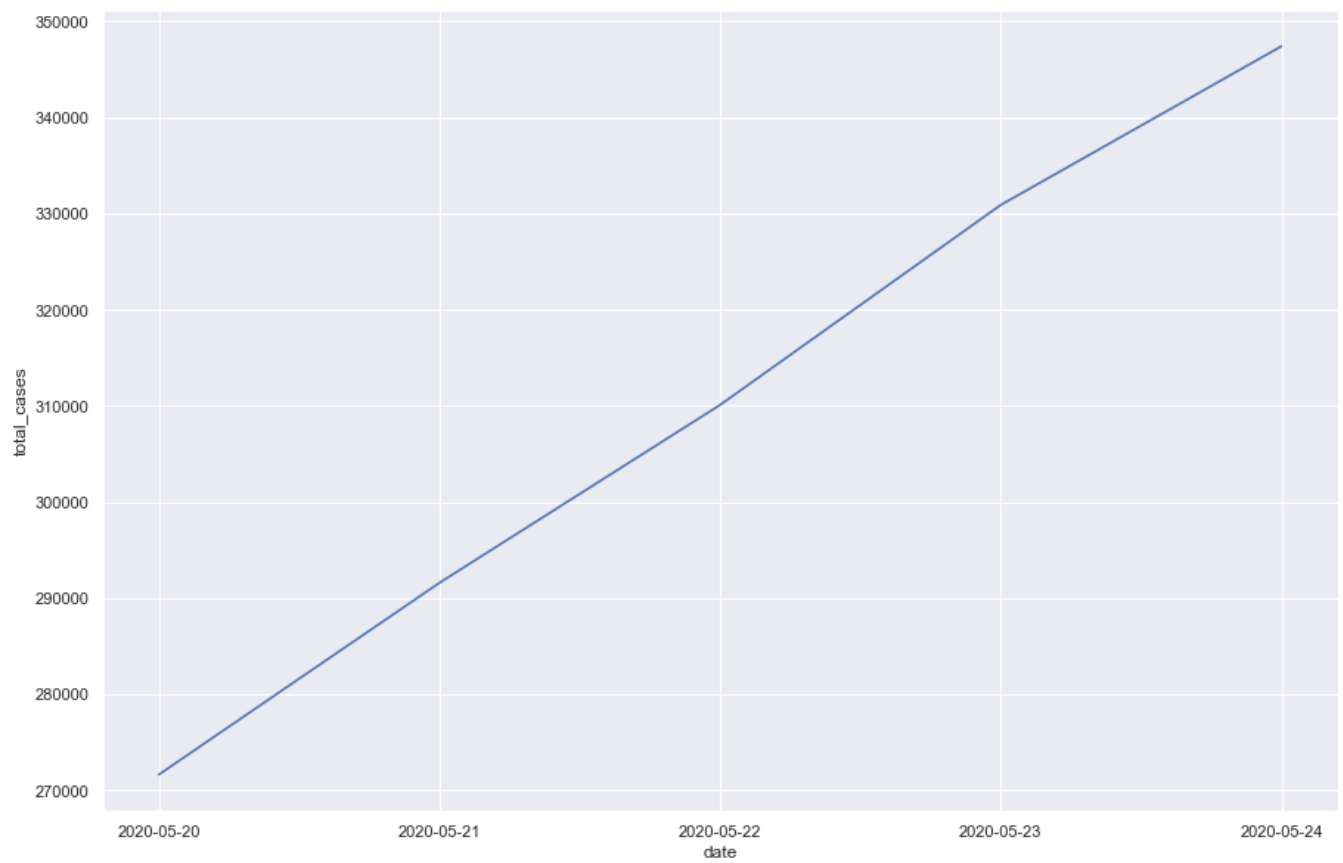
5 rows × 32 columns

In [20]:

```
#Making a dataframe for brazil for last 5 days
brazil_last_5_days=brazil_case.tail()
```

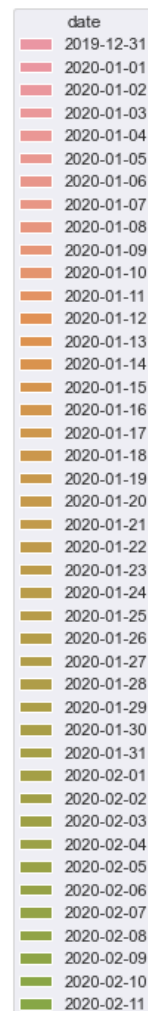
In [21]:

```
#Total cases in last 5 days
sns.set(rc={'figure.figsize':(15,10)})
sns.lineplot(x="date",y="total_cases",data=brazil_last_5_days)
plt.show()
```

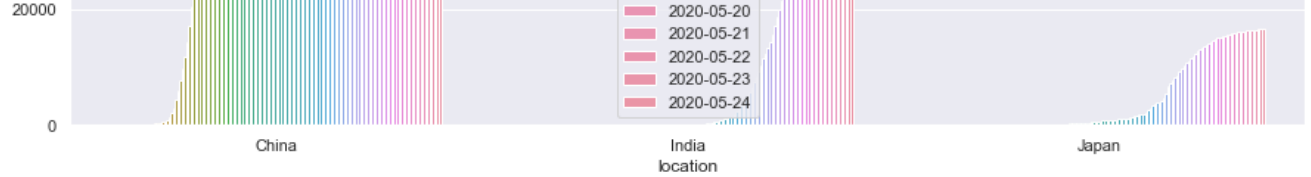


```
In [22]: #Understanding cases of India, China and Japan
india_japan_china=covid[(covid["location"] == "India") | (covid["location"] == "China")]
```

```
In [23]: #Plotting growth of cases across China, India and Japan
sns.set(rc={'figure.figsize': (15,10)})
sns.barplot(x="location",y="total_cases",data=india_japan_china,hue="date")
plt.show()
```

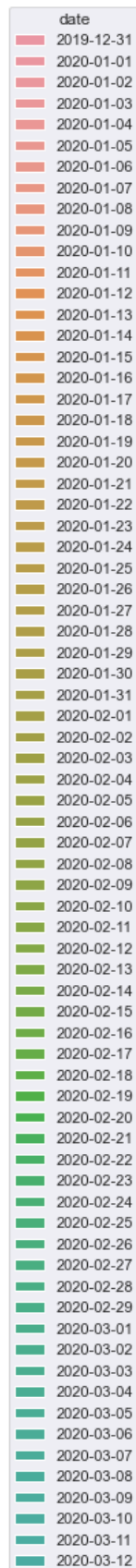


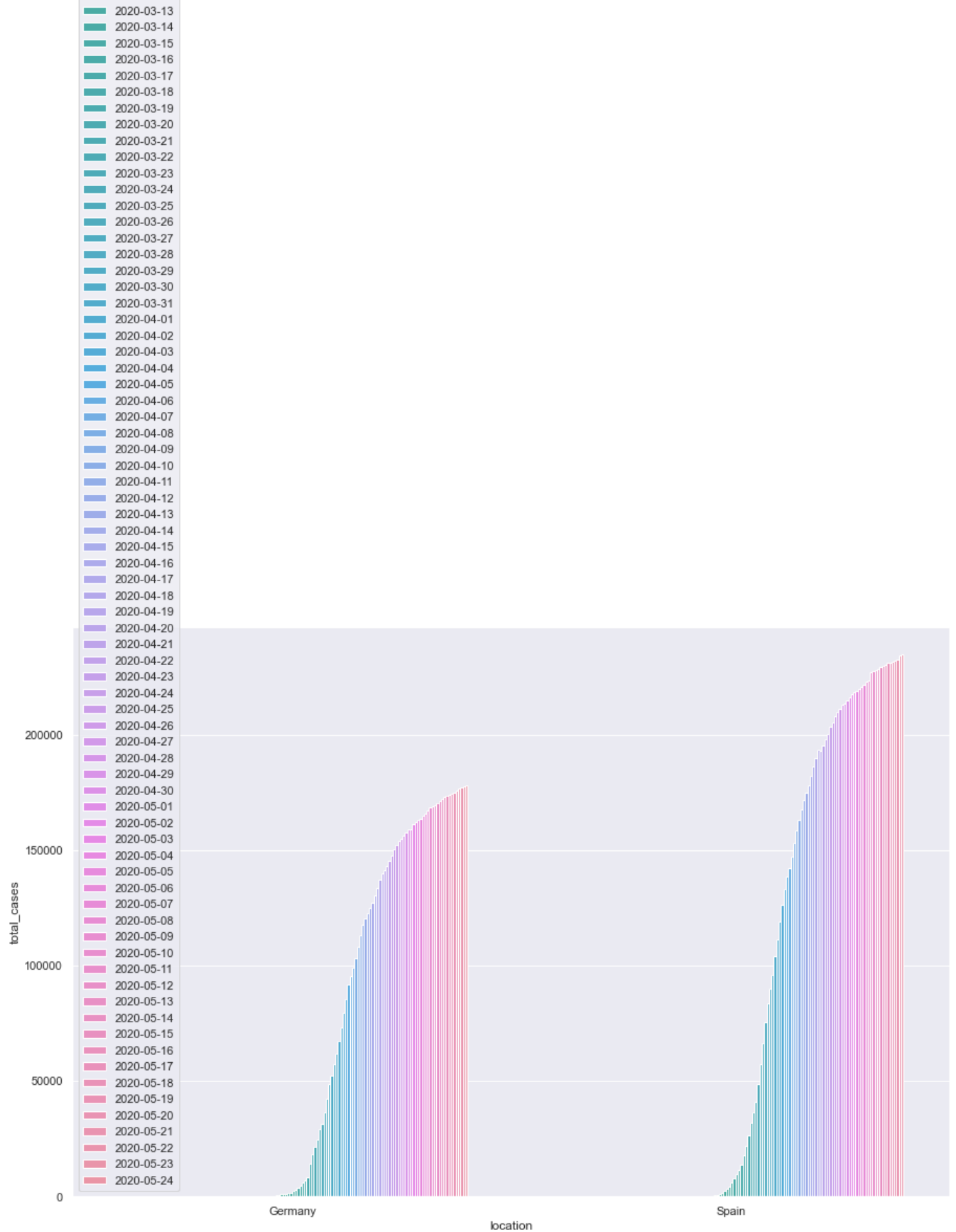




```
In [24]: #Understanding cases of germany and spain
germany_spain=covid[(covid["location"] == "Germany") | (covid["location"] == "Spain")]
```

```
In [25]: #Plotting growth of cases across Germany and Spain
sns.set(rc={'figure.figsize': (15,10)})
sns.barplot(x="location", y="total_cases", data=germany_spain, hue="date")
plt.show()
```





In [26]:

```
#Getting latest data
last_day_cases=covid[covid["date"]=="2020-05-24"]
last_day_cases
```

Out[26]:

	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million
62	ABW	Aruba	2020-05-24	101	0	3	0	945.99
198	AFG	Afghanistan	2020-05-24	9998	782	216	11	256.83
262	AGO	Angola	2020-05-24	60	0	3	0	1.82

321	AIA	Anguilla	2020-05-24	3	0	0	0	199.97
398	ALB	Albania	2020-05-24	989	8	31	0	343.66
...
19045	YEM	Yemen	2020-05-24	212	7	39	6	7.10
19153	ZAF	South Africa	2020-05-24	21343	1218	407	10	359.86
19220	ZMB	Zambia	2020-05-24	920	0	7	0	50.04
19285	ZWE	Zimbabwe	2020-05-24	56	0	4	0	3.76
19431	OWID_WRL	World	2020-05-24	5273572	97636	341722	3633	676.55

207 rows × 32 columns

In [27]:

#Sorting data w.r.t total_cases
max_cases_country=last_day_cases.sort_values(by="total_cases",ascending=False)
max_cases_country

Out[27]:

	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million
19431	OWID_WRL	World	2020-05-24	5273572	97636	341722	3633	676.550
18391	USA	United States	2020-05-24	1622670	21236	97087	1080	4902.287
2655	BRA	Brazil	2020-05-24	347398	16508	22013	965	1634.357
15569	RUS	Russia	2020-05-24	335882	9434	3388	139	2301.595
9396	ITA	Italy	2020-05-24	229327	669	32735	119	3792.922
...
18723	VGB	British Virgin Islands	2020-05-24	8	0	1	0	264.577
1645	BES	Bonaire Sint Eustatius and Saba	2020-05-24	6	0	0	0	228.824
5543	ESH	Western Sahara	2020-05-24	6	0	0	0	10.045
321	AIA	Anguilla	2020-05-24	3	0	0	0	199.973
11086	LSO	Lesotho	2020-05-24	2	1	0	0	0.934

207 rows × 32 columns

In [28]:

```
#Top 5 countries with maximum cases
max_cases_country[1:6]
```

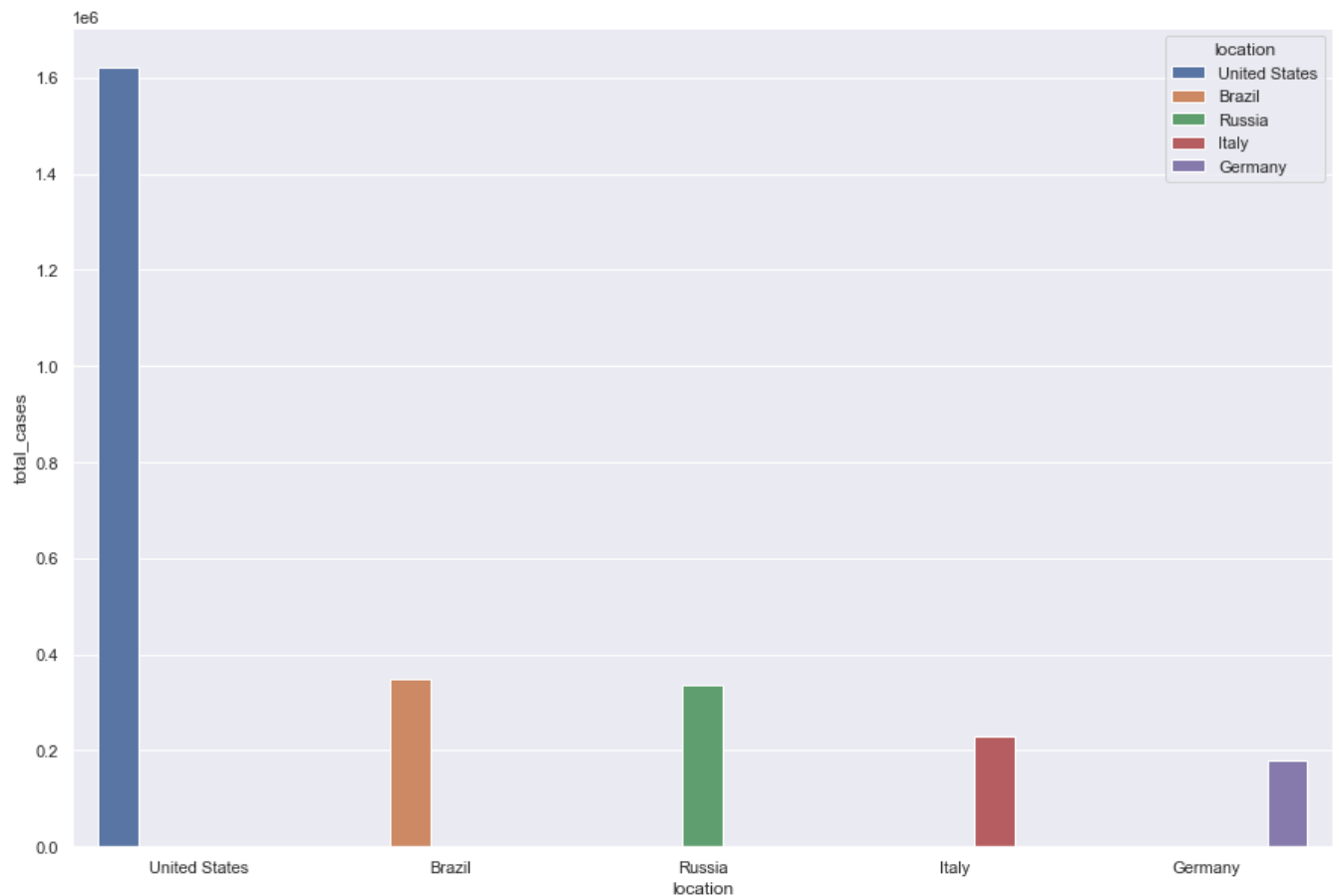
Out[28]:

	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	new_cases_per_million
18391	USA	United States	2020-05-24	1622670	21236	97087	1080	4902.287	0.006
2655	BRA	Brazil	2020-05-24	347398	16508	22013	965	1634.357	0.008
15569	RUS	Russia	2020-05-24	335882	9434	3388	139	2301.595	0.003
9396	ITA	Italy	2020-05-24	229327	669	32735	119	3792.922	0.000
4613	DEU	Germany	2020-05-24	178281	431	8247	31	2127.866	0.000

5 rows × 32 columns

In [29]:

```
#Making bar-plot for countries with top cases
sns.barplot(x="location",y="total_cases",data=max_cases_country[1:6],hue="location")
plt.show()
```



In [30]:

```
india_case.head()
```

Out[30]:

	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	new_cases_per_million
8379	IND	India	2019-12-31	0	0	0	0	0.0	0.0
8380	IND	India	2020-01-01	0	0	0	0	0.0	0.0
8381	IND	India	2020-01-02	0	0	0	0	0.0	0.0

8382	IND	India	2020-01-03	0	0	0	0	0.0
8383	IND	India	2020-01-04	0	0	0	0	0.0

5 rows × 32 columns

```
In [32]: #Linear regression
from sklearn.model_selection import train_test_split
```

```
In [33]: #converting string date to date-time
import datetime as dt
india_case['date'] = pd.to_datetime(india_case['date'])
india_case.head()
```

C:\Users\swara\AppData\Local\Temp\ipykernel_28228\838081926.py:3: 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
india_case['date'] = pd.to_datetime(india_case['date'])

```
Out[33]:
```

	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	nev
8379	IND	India	2019-12-31	0	0	0	0	0.0	
8380	IND	India	2020-01-01	0	0	0	0	0.0	
8381	IND	India	2020-01-02	0	0	0	0	0.0	
8382	IND	India	2020-01-03	0	0	0	0	0.0	
8383	IND	India	2020-01-04	0	0	0	0	0.0	

5 rows × 32 columns

```
In [34]: india_case.head()
```

```
Out[34]:
```

	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	nev
8379	IND	India	2019-12-31	0	0	0	0	0.0	
8380	IND	India	2020-01-01	0	0	0	0	0.0	
8381	IND	India	2020-01-02	0	0	0	0	0.0	
8382	IND	India	2020-01-03	0	0	0	0	0.0	
8383	IND	India	2020-01-04	0	0	0	0	0.0	

5 rows × 32 columns

```
In [35]: #converting date-time to ordinal
        india_case['date']=india_case['date'].map(dt.datetime.toordinal)
        india_case.head()
```

C:\Users\swara\AppData\Local\Temp\ipykernel_28228\1469438671.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

```
        india_case['date']=india_case['date'].map(dt.datetime.toordinal)
```

```
Out[35]:
```

	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	ne
8379	IND	India	737424	0	0	0	0	0.0	
8380	IND	India	737425	0	0	0	0	0.0	
8381	IND	India	737426	0	0	0	0	0.0	
8382	IND	India	737427	0	0	0	0	0.0	
8383	IND	India	737428	0	0	0	0	0.0	

5 rows × 32 columns

```
In [36]: #getting dependent variable and inpedent variable
        x=india_case['date']
        y=india_case['total_cases']
```

```
In [37]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```
In [38]: from sklearn.linear_model import LinearRegression
```

```
In [39]: lr = LinearRegression()
```

```
In [40]: lr.fit(np.array(x_train).reshape(-1,1),np.array(y_train).reshape(-1,1))
```

```
Out[40]: LinearRegression()
```

```
In [41]: india_case.tail()
```

```
Out[41]:
```

	iso_code	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	ne
8519	IND	India	737565	106750	5611	3303	140	77.355	
8520	IND	India	737566	112359	5609	3435	132	81.419	
8521	IND	India	737567	118447	6088	3583	148	85.831	
8522	IND	India	737568	125101	6654	3720	137	90.653	
8523	IND	India	737569	131868	6767	3867	147	95.556	

5 rows × 32 columns

```
In [42]: y_pred=lr.predict(np.array(x_test).reshape(-1,1))
```

```
In [43]:
```

```
from sklearn.metrics import mean_squared_error
```

```
In [44]: mean_squared_error(x_test,y_pred)
```

```
Out[44]: 525988157783.2488
```

```
In [45]: lr.predict(np.array([[737573]]))
```

```
Out[45]: array([[55032.59839666]])
```

```
In [ ]:
```