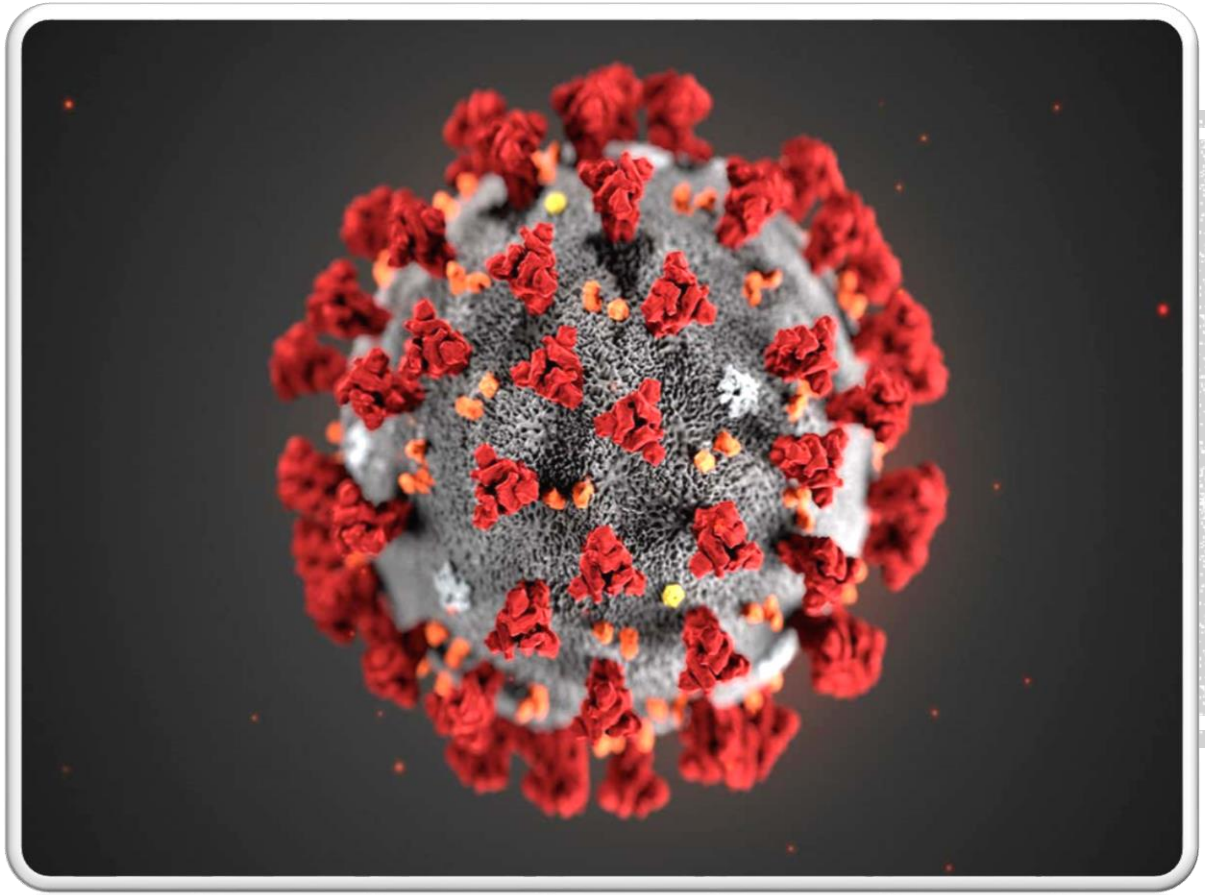


SUMMARIZING ANALAYSIS ON COVID-19 DATASET (INDIA)

A conclusive report

INTRODUCTION

- We recently built a project to understand the COVID-19 trend in India to analyse the future outcomes in terms of 30 parameters.
- It was aimed to analyse what factors help us understand the severity of rising cases and which factors do we need to look at the most.
- A dataset was provided and we were given specific targets we needed to satisfy in order to mark the project complete.
- The basic tasks were subsetting just India related values and dropping columns with more than 50% null values, otherwise replacing them with either median or mean where satisfied.
- We were told to perform exploratory data analysis by creating histograms,scatterplots and lineplots that provided us with the graphical understanding of trends and made it easier for us to derive conclusions.

PARAMETERS

- 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 thousand, life expectancy

- *the parameters are both numerical and categorical in nature.

A WORD OF ADVICE

- Whenever we explore a dataset it is natural that all parameters don't play a decisive role in the prediction model and thus add little to no value in the development of the project but are still explored to keep for future study.
- Hence the following slides will only cover major concepts and parameters rather than covering all of them for keeping the presentation to the point and not redundant.

CONVERSION AND SUBSETTING

- The dates were first converted into ordinal form in python to ease understanding.

```
import datetime as dt
df["date"]=pd.to_datetime(df["date"])
df["date"]=df["date"].map(dt.datetime.toordinal)
```

- This made our first column after dropping location and iso codes cause it was just done on India.

```
df.drop(['iso_code','continent','location'],axis=1,inplace=True)
```

```
df=covid.loc[covid['location'] == 'India']
```

HANDLING THE MISSING VALUES

- After reding the data we found out that several columns were having null values so for our model to work either these columns were to be dropped but only on one condition that if the null values were more than 50% of the total values but in our case it was not true hence we replaced them with the mean, median and mode.

```
df['total_tests']=df['total_tests'].fillna(df['total_tests'].median())

df['new_tests']=df['new_tests'].fillna(df['new_tests'].mean())

df['total_tests_per_thousand']=df['total_tests_per_thousand'].fillna(df['total_tests_per_thousand'].med

df['new_tests_per_thousand']=df['new_tests_per_thousand'].fillna(df['new_tests_per_thousand'].mean())

df['new_tests_smoothed']=df['new_tests_smoothed'].fillna(df['new_tests_smoothed'].mean())

df['new_tests_smoothed_per_thousand']=df['new_tests_smoothed_per_thousand'].fillna(df['new_tests_smooth

df['stringency_index']=df['stringency_index'].fillna(df['stringency_index'].median())
```

EXPLORATORY DATA ANALYSIS

- In order to analyse our data we need to perform univariate as well as bivariate analysis as well as find the mean median and mode.
- From this we found out that a lot of columns had the same value throughout, so in further analysis we avoid those in order to have a to the point presentation

MEAN AND MEDIAN

```
df.mean()
```

```
date                                 7.375211e+05
total_cases                          1.157136e+05
new_cases                            4.379139e+03
total_deaths                         3.412995e+03
new_deaths                           1.168763e+02
total_cases_per_million              8.385014e+01
new_cases_per_million                3.173284e+00
total_deaths_per_million             2.473186e+00
new_deaths_per_million               8.468041e-02
total_tests                          2.904157e+06
new_tests                            1.084671e+05
total_tests_per_thousand             2.104613e+00
new_tests_per_thousand               7.859223e-02
new_tests_smoothed                   9.203472e+04
new_tests_smoothed_per_thousand      6.669298e-02
tests_units                          6.237113e-01
stringency_index                     5.562062e+01
population                           1.380004e+09
population_density                   4.504190e+02
median_age                           2.820000e+01
aged_65_older                        5.989000e+00
aged_70_older                        3.414000e+00
gdp_per_capita                       6.426674e+03
extreme_poverty                      2.120000e+01
cvd_death_rate                       2.822800e+02
diabetes_prevalence                  1.039000e+01
female_smokers                       1.900000e+00
male_smokers                         2.060000e+01
handwashing_facilities               5.955000e+01
hospital_beds_per_thousand           5.300000e-01
life_expectancy                      6.966000e+01
dtype: float64
```

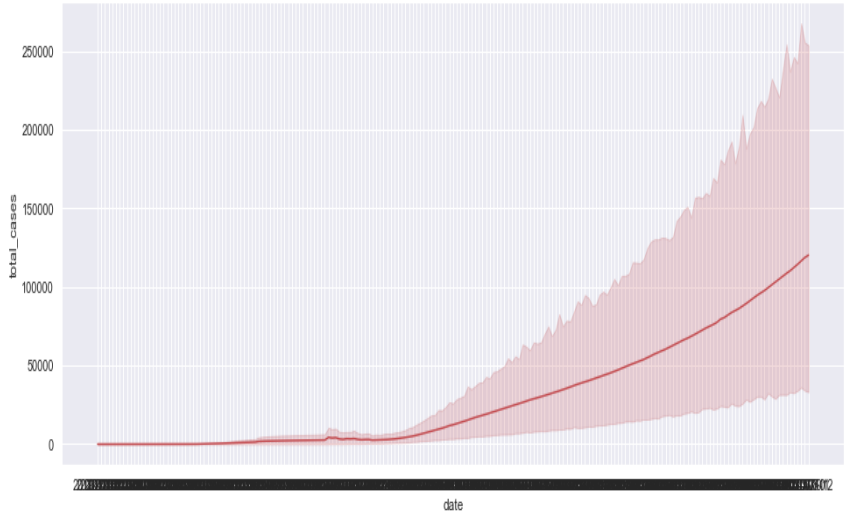
```
df.median()
```

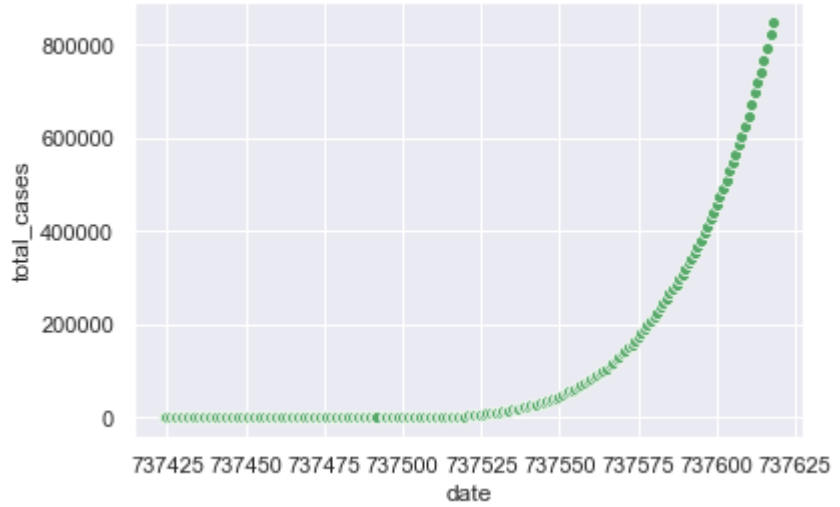
```
date                                 7.375215e+05
total_cases                          4.244000e+03
new_cases                            5.845000e+02
total_deaths                         1.115000e+02
new_deaths                           1.600000e+01
total_cases_per_million              3.075500e+00
new_cases_per_million                4.240000e-01
total_deaths_per_million             8.100000e-02
new_deaths_per_million               1.150000e-02
total_tests                          2.302792e+06
new_tests                            1.084671e+05
total_tests_per_thousand             1.669000e+00
new_tests_per_thousand               7.859223e-02
new_tests_smoothed                   9.203472e+04
new_tests_smoothed_per_thousand      6.669298e-02
tests_units                          1.000000e+00
stringency_index                     7.546000e+01
population                           1.380004e+09
population_density                   4.504190e+02
median_age                           2.820000e+01
aged_65_older                        5.989000e+00
aged_70_older                        3.414000e+00
gdp_per_capita                       6.426674e+03
extreme_poverty                      2.120000e+01
cvd_death_rate                       2.822800e+02
diabetes_prevalence                  1.039000e+01
female_smokers                       1.900000e+00
male_smokers                         2.060000e+01
handwashing_facilities               5.955000e+01
hospital_beds_per_thousand           5.300000e-01
life_expectancy                      6.966000e+01
dtype: float64
```

BIVARIATE ANALYSIS

TOTAL CASES

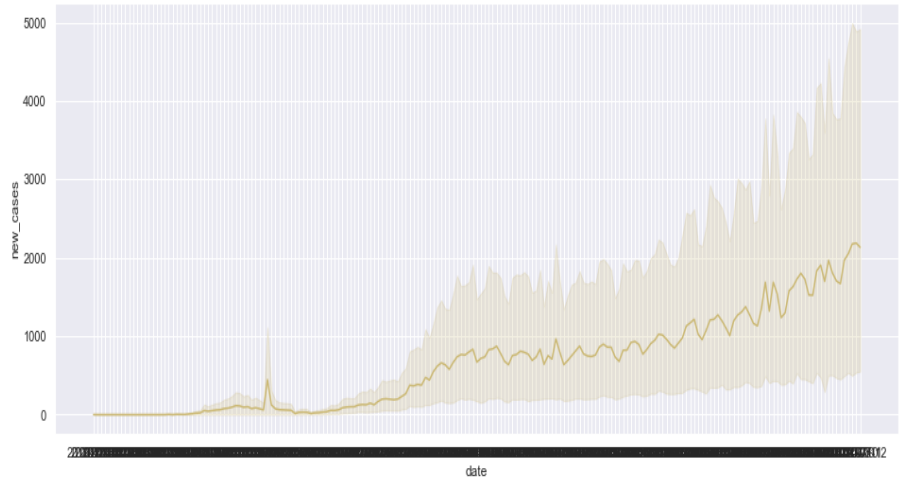
- This was the parameter selected to predict in the model as our target was to find the total cases at a particular date .We notice a steady increase in cases by dates.

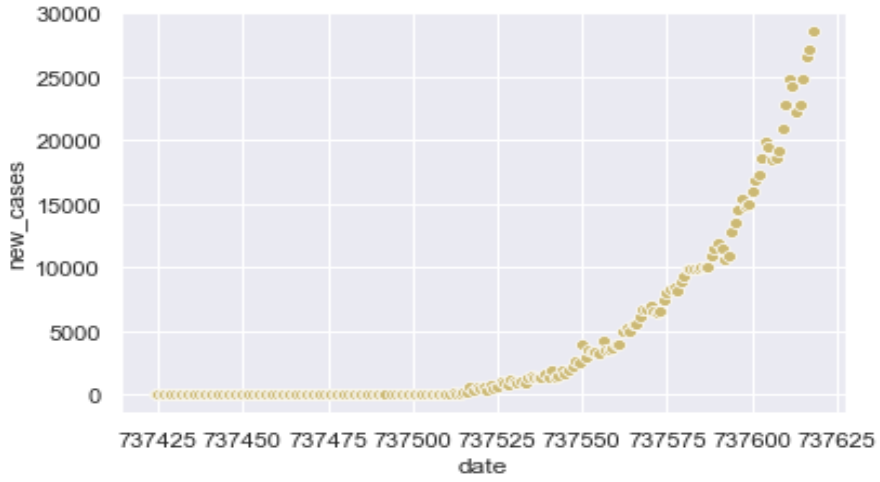




NEW CASES

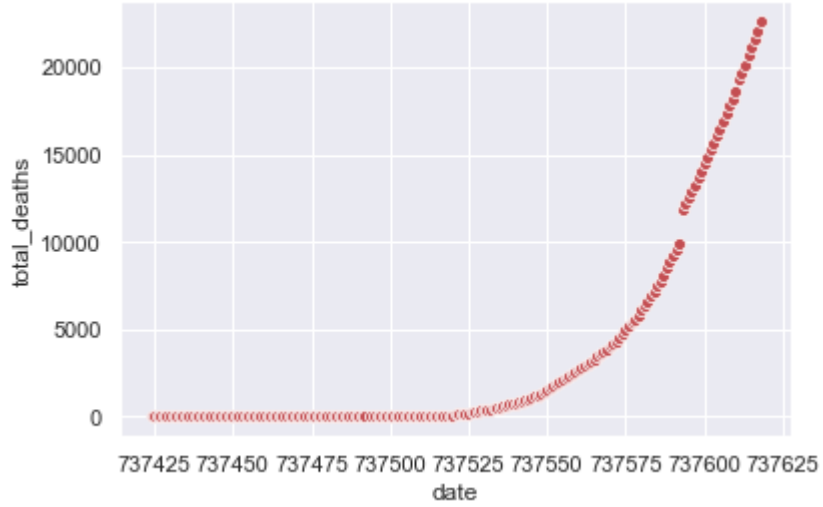
- Follows a similar pattern and thus increases steadily.

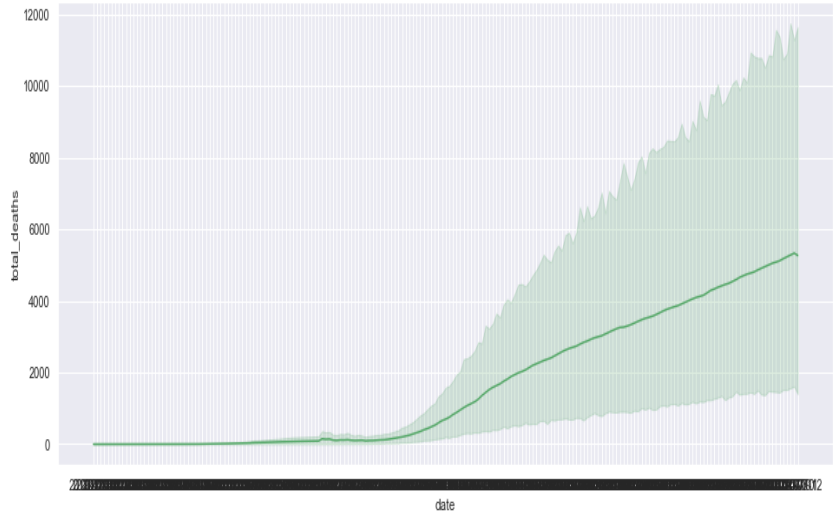




TOTAL DEATHS

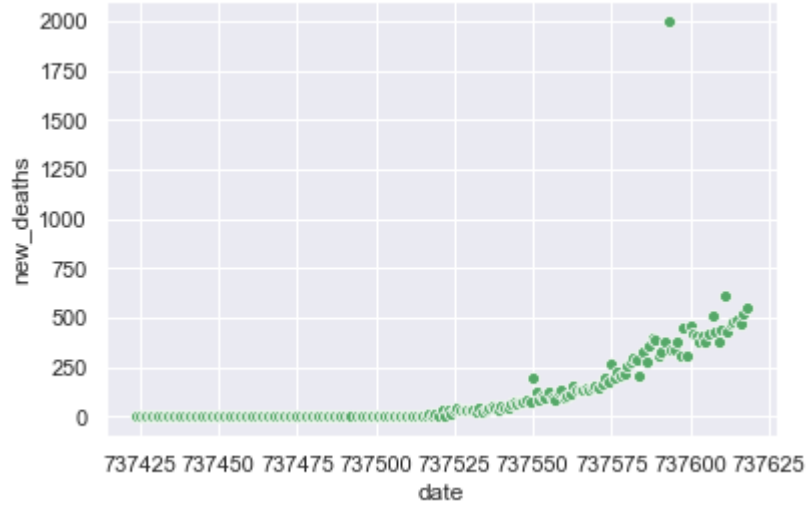
- This parameter is also one of the important ones as we have to predict the mortality rate to prepare resources and conditions for the future.

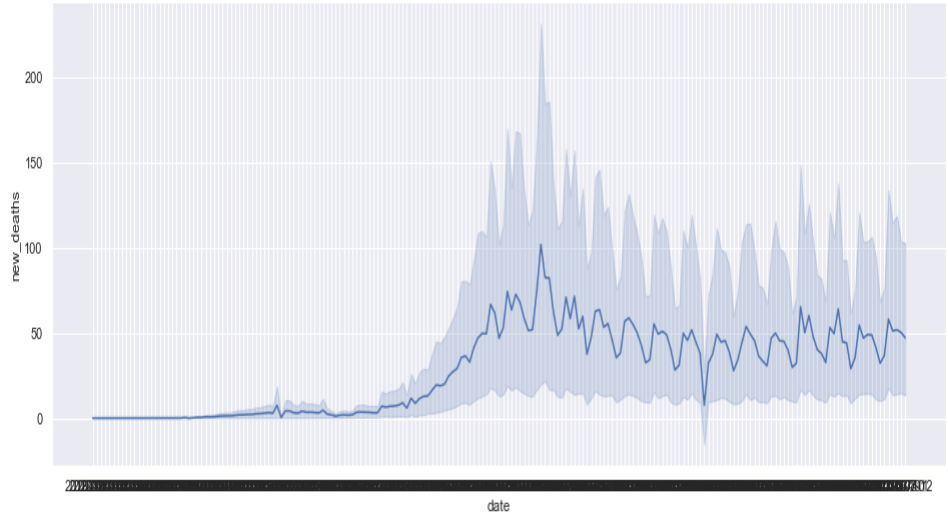




NEW DEATHS

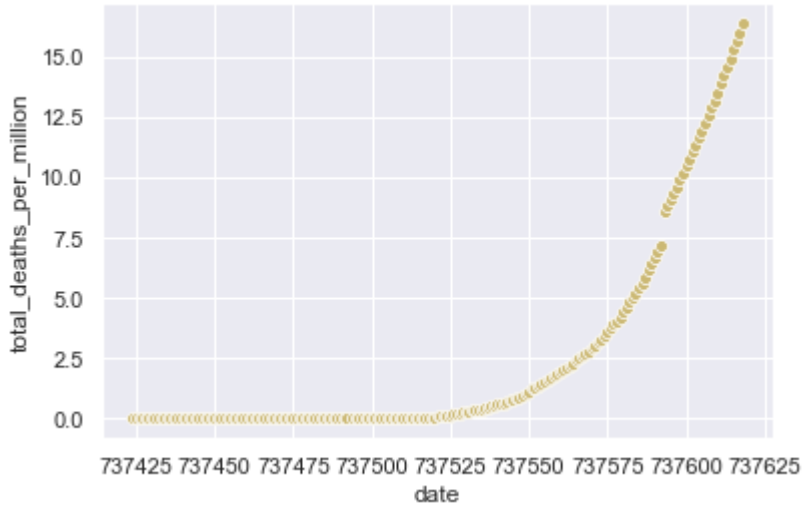
- Similar to the total deaths parameter this trend is bound to follow a similar pattern. But. if we observe the new deaths count is rather erratic and not a steady increase.

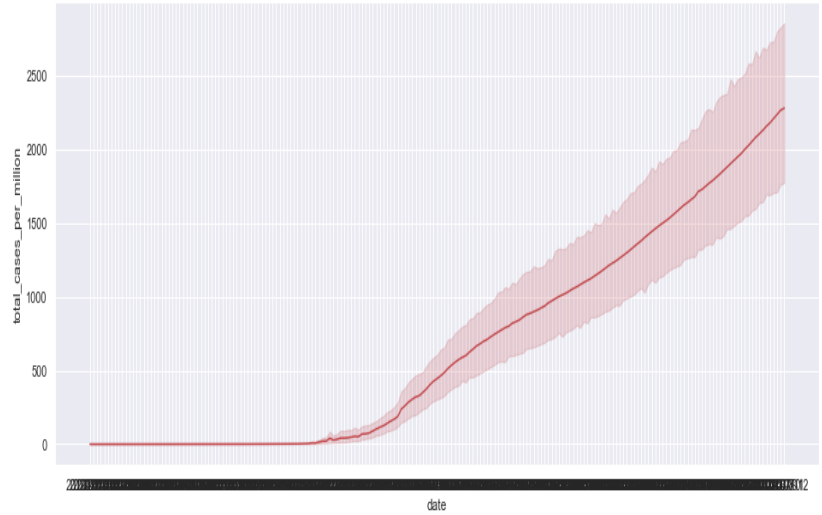




TOTAL CASES PER MILLION

- In terms of finding the capabilities of the virus to spread. This parameter helps us understand the rate of infection and helps us make guidelines to follow.

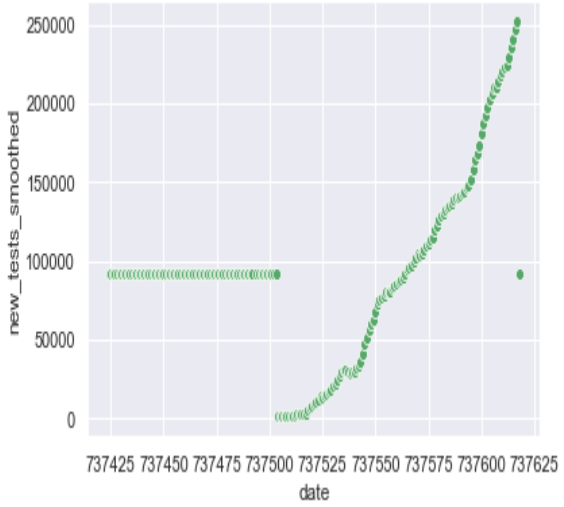


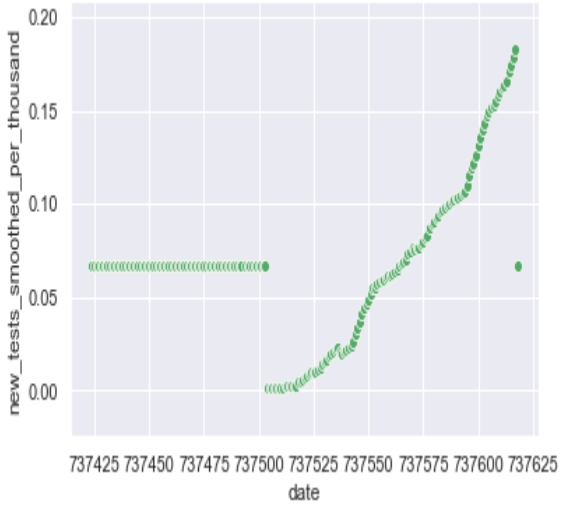


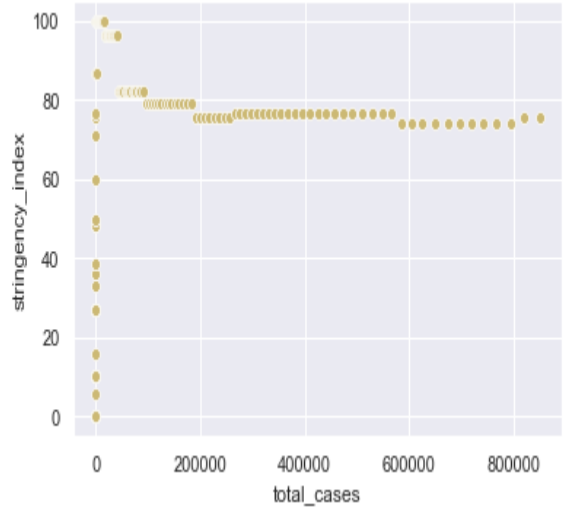
OTHER IMPORTANT PARAMETERS WORTH A MENTION.

- New Cases smoothed
- stringency index

new cases smoothed per 1k







PREDICTION MODELS

- We were given two models i.e. linear regression and random forest classifier to build a prediction model for predicting total cases in the future.
- First step was to perform a train and test split in order to divide our data into training and testing data

SELECTING THE TARGET COLUMN AS TOTAL_CASES AND REST OF THE COLUMNS AS FEATURES

```
X = df.drop('total_cases',axis=1)
y = df['total_cases']
```

DIVIDING THE DATA INTO TRAIN AND TEST DOING TRAIN TEST SPLIT

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

SCORES OF OUR RESPECTIVE MODELS:

In [82]:
```
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
print('Variance score: %.2f' % lm.score(X_test, y_test))
```

```
MAE: 0.2086407987878654
MSE: 0.07013187982105064
RMSE: 0.2648242432653224
Variance score: 1.00
```

In [83]:
```
lm.score(X_test,y_test)
```

```
0.9999999999987547
```

RANDOM FOREST

RANDOM FOREST

In [84]:
```
from sklearn.ensemble import RandomForestRegressor

rfc=RandomForestRegressor()
rfc.fit(X_train,y_train)
y_pred = rfc.predict(X_test)
```

In [85]:
```
rfc.score(X_test,y_test)
```

```
0.9141818412937843
```

After coming up with unique solutions to increase the accuracy of the models and achieving satisfactory results
It was an insightful project targeting the crux of a genuine problem we face as of now in terms of a deadly virus .

CONCLUSION

- In this project we learnt the following:
- 1)To import a dataset and subset needed values
- 2) To perform exploratory data analysis on parameters
- 3) To build prediction models with the various tools and techniques available
- 4) To perform prediction on a parameter and concluding it with evidence.

A VOTE OF THANKS

- I would like to thank the whole team present at verzeo to come up with such a memorable course.
- Providing a platform of knowledge in these testing times is a deed well done.
- I further thank al the members to make the course as detailed and entertaining as they could.
- Looking forward to more such courses.