COVID-19: The Pandemic of 2020

Team Name: Pritesh&Gaurav

Team : Pritesh Jha
Gaurav Agarwal

Author Note: Our core interests belong to Machine Learning and Deep learning which can be utilized for the growth of humanity towards a better world.

E-mail: priteshjha27@gmail.com
agarwal.gaurav2704@gmail.com

LinkedIn: https://www.linkedin.com/in/priteshjha27/
https://www.linkedin.com/in/gauravagarwal2704/

Web App link: https://azure-dash-test.azurewebsites.net/

Video Demo: https://youtu.be/xxLmWRriQVU

Note – The Video is based on the demo running on local machine. Please refer to the web app link for Azure services.

Notebook Link :
https://covid19priteshjha27.notebooks.azure.com/j/notebooks/Analysis_Notebook.ipynb

Introduction:

An outbreak of “pneumonia of unknown etiology” in Wuhan, Hubei Province in China in early December 2019 has now reached a Pandemic state globally. The causative agent soon proved to be a new beta-coronavirus related to the Middle East Respiratory Syndrome virus (MERS-CoV) and the Severe Acute Respiratory Syndrome virus (SARS-CoV). The novel coronavirus SARS-CoV-2 disease has been named “COVID-19” by the World Health Organization (WHO) and on January 30.

This project is based on the analysis of the virus outbreak globally along with a predictive modelling on the outbreak in India. This project also offers the user to locate any nearby essential services and display it on the map using Foursquare API.

Problem to be Solved / Simplified?

As the virus outbreak has hit every country globally and every country has been in a state of quarantine, it has been a single and most debated question that can be framed in multiple ways such as –

- How long will the pandemic last globally?
- When will the quarantine end?
- When can the international travels resume?

This project does not give a date or time that can answer the above questions.

However, this project is focussed on the analytical means to be a decisive factor for the world governments to answer the above questions.

Who is this project for?

The targeted audience for the project is both the governments and the people. The analysis on the global scale have been categorized on the basis of multiple features of a nation such as Population of the country, Median age of the population and the number of hospital beds per 1000 patients in the country. Also, the hospitals, police stations and the pharmacy can be located on the map based on a user input provided address.

What should be the aim of the reader?

The aim of the reader should be to strategize their following weeks operations in relation to the analysis and the predictive modelling so that the daily routines could be carried out without any public exposure to the virus. Also, in case on emergencies, the essential services could be located within a given radius of 2 KM (default value, but can be variable).

Azure services Used for the project:

1. Azure Open Dataset
2. Azure ML Services
3. Azure Jupyter Notebook and Visual studio Code
4. Azure Docker Container registry
5. Azure Web App

Significance

The significance of this contribution is that it is mainly focussed on improvising the current health situations by providing a quantitative analysis of the dataset collected from various sources and to help the general public in these times of emergencies in locating essential services.

Table of Contents

1. Importing Libraries

The libraries required to run the code on a Jupyter notebook are as follows:

- NumPy
  Web – https://numpy.org/

- Pandas
  Web – https://pandas.pydata.org/

- SciPy
  Web – https://www.scipy.org/

- SkLearn
  Web – https://scikit-learn.org/stable/

- GeoPy
  Web – https://geopy.readthedocs.io/en/stable/

- Folium
  Web – https://python-visualization.github.io/folium/

- Matplotlib
  Web – https://matplotlib.org/index.html

- Plotly and Dash
  Web – https://plotly.com/

- Json

- os

- requests

- warnings

2. Data Collection:

The data for the Covid-19 cases have been adopted from 3 different sources for this project –

- https://azure.microsoft.com/en-in/services/open-datasets/catalog/covid-19-data-lake/
  The Oxford Covid-19 Government Response Tracker (OxCGRT) dataset contains systematic information on which governments have taken which measures, and when.

- https://github.com/CSSEGISandData/COVID-19
  This is provided by Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE)
- https://covid.ourworldindata.org/data/
  This is open source dataset provided by OurWorldInData organization.

  Both of the above-mentioned data sources have been primarily scraped from sources such as:
- https://covid19.who.int/
- https://www.worldometers.info/

The data for the essential services will be acquired though the Foursquare API. Here is the link to the webpage of the API:

- https://developer.foursquare.com/places

3. Data Manipulation:

Upon loading the dataset provided by Oxford and John Hopkins University, the data for Confirmed Cases/ Deaths/ Recovered cases are displayed as follows:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20
0	NaN	Afghanistan	33.0000	65.0000	0	0	0	0	0	0
1	NaN	Albania	41.1533	20.1683	0	0	0	0	0	0
2	NaN	Algeria	28.0339	1.6596	0	0	0	0	0	0
3	NaN	Andorra	42.5063	1.5218	0	0	0	0	0	0
4	NaN	Angola	-11.2027	17.8739	0	0	0	0	0	0

The dates are arranged as columns in the dataset. This is not how I intend to keep the dataset as it is going to be troublesome in the later stages of the project where the data needs to be plotted in a timeseries order. Hence, the data needs to be “melted” using pandas.dataframe.melt() method which gives us the output data as follows:

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered
0	NaN	Afghanistan	33.0000	65.0000	1/22/20	0	0	0.0
1	NaN	Albania	41.1533	20.1683	1/22/20	0	0	0.0
2	NaN	Algeria	28.0339	1.6596	1/22/20	0	0	0.0
3	NaN	Andorra	42.5063	1.5218	1/22/20	0	0	0.0
4	NaN	Angola	-11.2027	17.8739	1/22/20	0	0	0.0

The “Date” column is of the datatype String. I am converting it into a timeseries object with pandas method pandas.to_datetime().

The are a few Ships names in the data set as well which have been in quarantined state as they have been impacted with the virus as well.

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered
37	Grand Princess	Canada	37.6489	-122.6655	2020-01-22	0	0	NaN
88	NaN	Diamond Princess	0.0000	0.0000	2020-01-22	0	0	0.0
231	Diamond Princess	Canada	0.0000	0.0000	2020-01-22	0	0	NaN
251	NaN	MS Zaandam	0.0000	0.0000	2020-01-22	0	0	0.0
303	Grand Princess	Canada	37.6489	-122.6655	2020-01-23	0	0	NaN
...	...	...	...	...	...	...	...	...
38289	NaN	MS Zaandam	0.0000	0.0000	2020-06-13	9	2	0.0
38341	Grand Princess	Canada	37.6489	-122.6655	2020-06-14	13	0	NaN
38392	NaN	Diamond Princess	0.0000	0.0000	2020-06-14	712	13	651.0
38535	Diamond Princess	Canada	0.0000	0.0000	2020-06-14	0	1	NaN
38555	NaN	MS Zaandam	0.0000	0.0000	2020-06-14	9	2	0.0

We don't need these in the analysis. Therefore, dropped these from the dataset.

We can find the Active Cases from the dataset using the following formula:

- Active Case = confirmed - (deaths + recovered)

There are a few countries in the dataframe where the stats are split based on their individual states/provinces. Hence those states data have been merged with their country data to give an accurate numerical figure for each country.

We have the total number of confirmed cases/deaths/recovered cases for each day. With this data we can find New cases/ New Deaths/ New Recovered cases by subtracting total cases on nth day by (n-1) th day.

Now, adding the dataset from OurWorldInData organization.

Both the datasets are "Joined "using pandas.merge() method which is similar to SQL's JOIN methods.

Now, the GeoPy library will be used to add columns such as "Latitude" and "Longitude" in the dataframe.

The columns in the data frame are as follows:

```
df.columns
```

```
Index(['Date', 'Country/Region', 'Confirmed', 'Deaths', 'Recovered', 'Active',
       'New_Cases', 'New_Deaths', 'New_Recovered', 'continent', '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',
       'combined_smokers', 'Latitude', 'Longitude'],
      dtype='object')
```

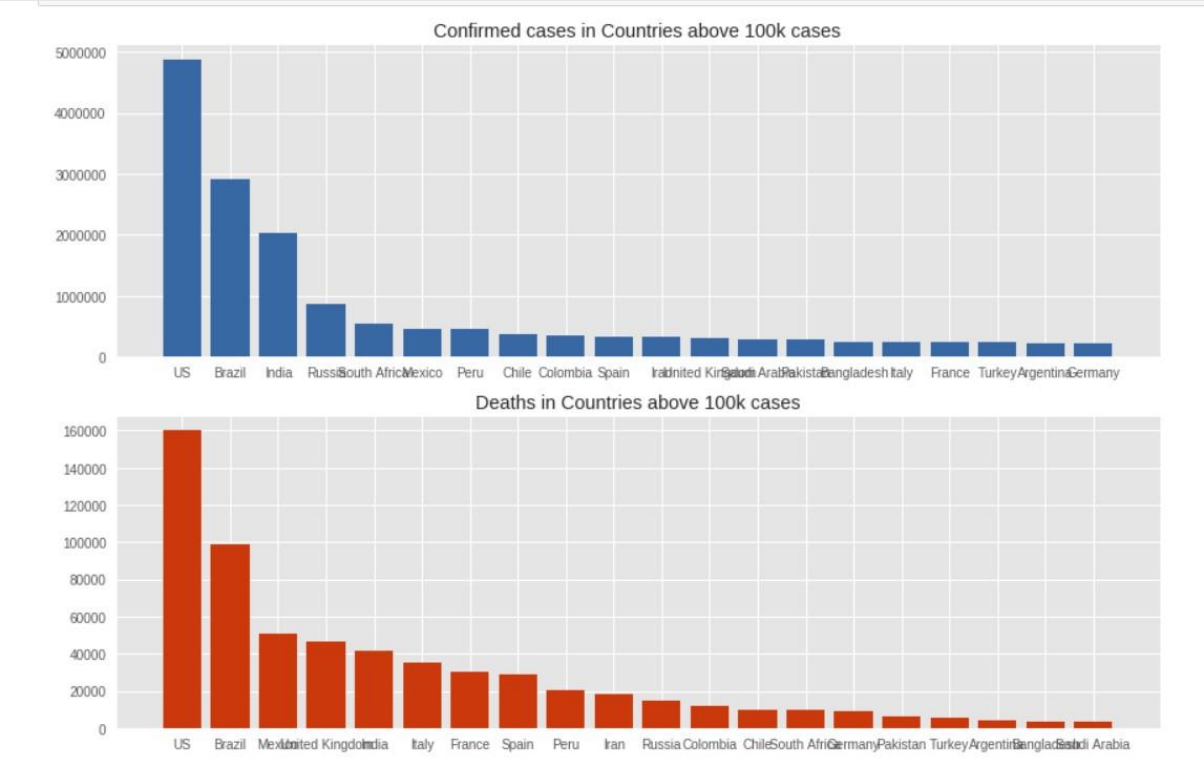
And thus, the dataset is complete for analysis.

4. Data Exploration and Visualization - Part 1

Here, we will analyse and summarize the data insights on a global scale.

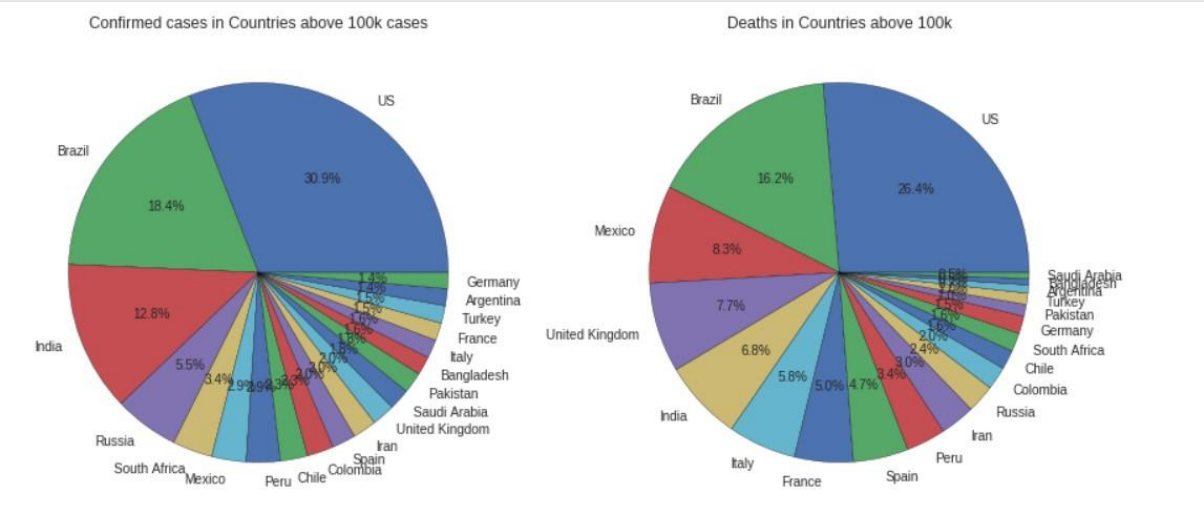
As displayed above, out of the first 10 countries to be affected by the virus, the countries that have been impacted the most are China, France and US as these nations have reach close to 100k cases.

So, let us review all the countries that have reached 100k confirmed cases.

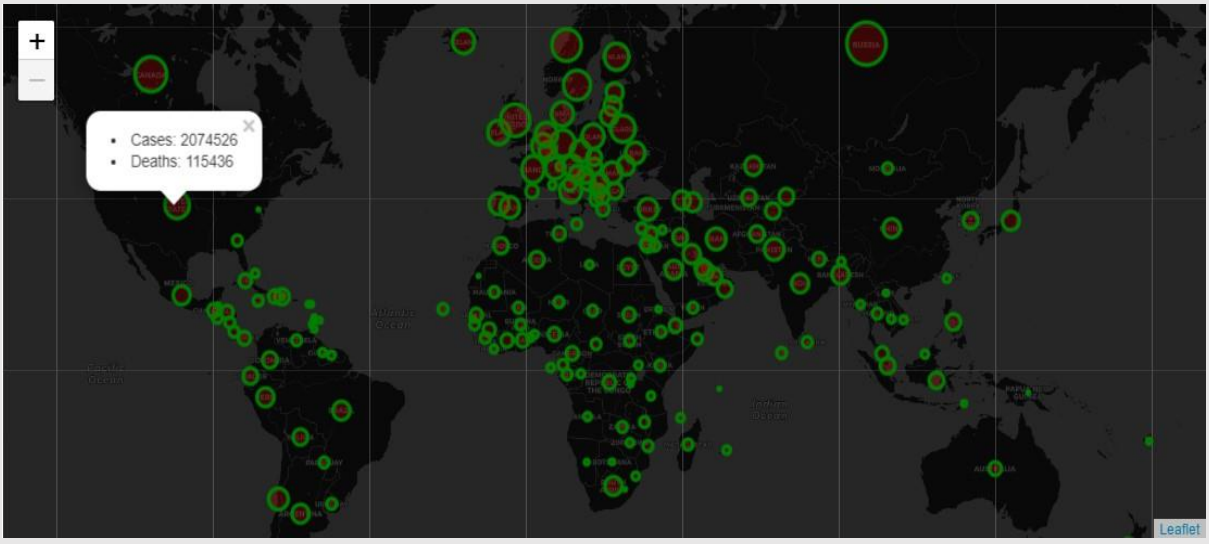


As the project is mainly focussed on the predictive model for India, we will be using these countries as a reference point.

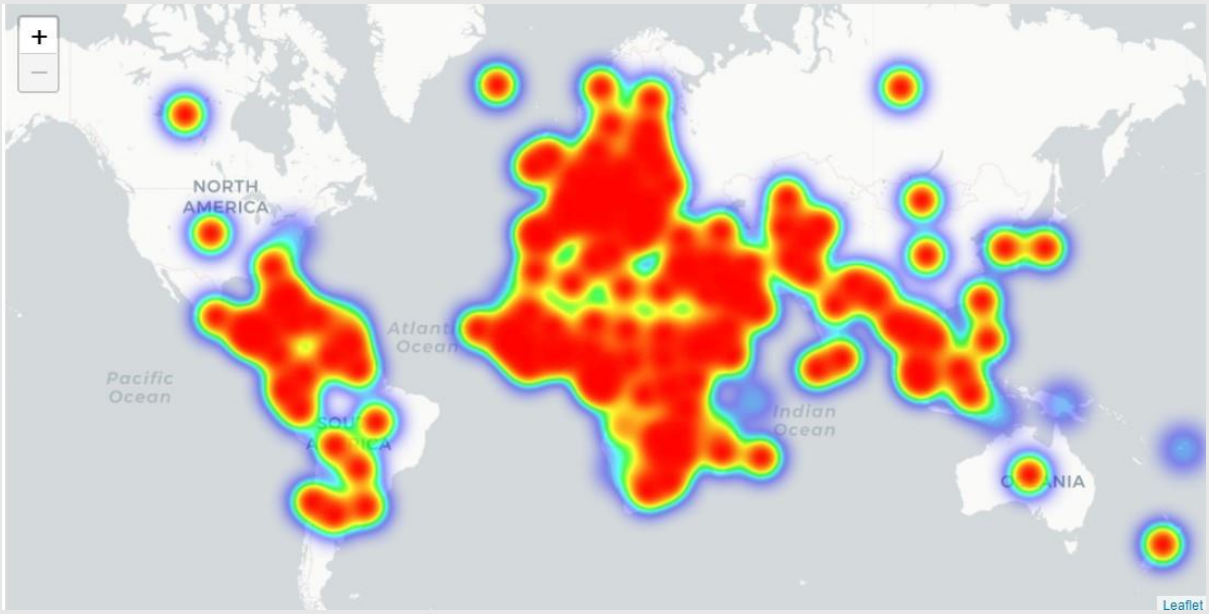
Out of these countries, let us have look at the ratios that each country contributes to the Confirmed cases and Deaths.



To visualize the impact of COVID-19 virus across globe, an interactive plot has been created on the Jupyter notebook that displays a popup with Confirmed cases and the Deaths count of that particular country. Here is a screenshot of the interactive chart.



Also, the highly impacted continents can be viewed on a Heatmap displayed below.



As we can conclude from the heatmap, the continents that are hugely affected by the virus are in the order Europe, Africa, Southern and Northern America and Southern Regions of Asia.

We have reviewed the global scale and have narrowed it to the regions where the virus has impacted the most.

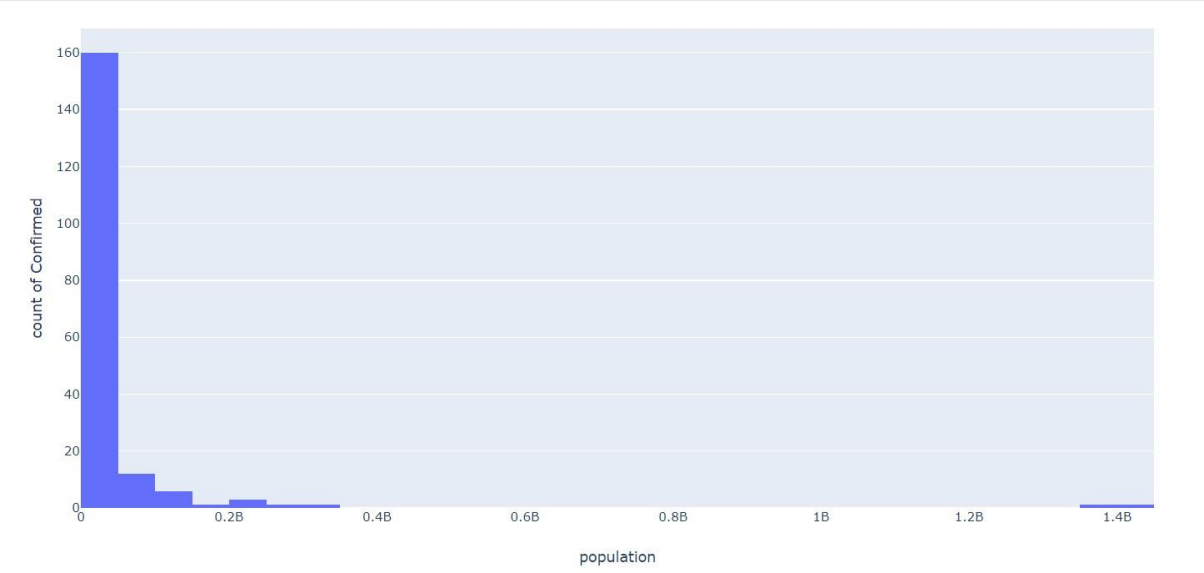
Now let us review the different correlative and distributive categories on which these countries have performed.

5. Data Analysis – Population

To analyse the Confirmed cases/ Deaths in all affected countries, we will be using histograms.

A histogram is an approximate representation of the distribution of numerical or categorical data. It was first introduced by Karl Pearson.

To construct a histogram, the first step is to "bin" the range of values—that is, divide the entire range of values into a series of intervals—and then count how many values fall into each interval. The bins are usually specified as consecutive, non-overlapping intervals of a variable.



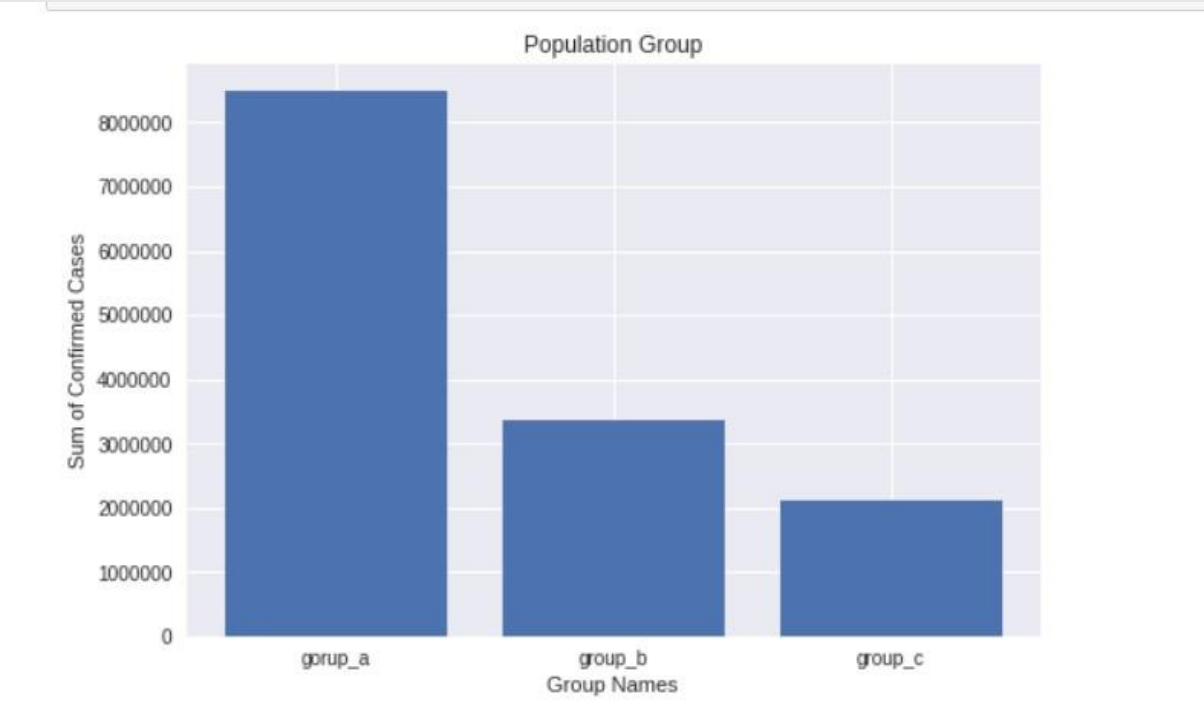
As per the histograms above, we can deduce that Histogram is Bimodal between 0-350M population i.e. there are 2 different groups of 0-150M and 200 to 350M.

3rd group is 1 Billion+ population which includes only India and China.

Let us split the countries in their respective groups and plot the summation of the Confirmed cases with respect to their groups.

The groups based on their population are as follows:

- Group A = 0 – 150M
- Group B = 200 – 350M
- Group C = 1B and above



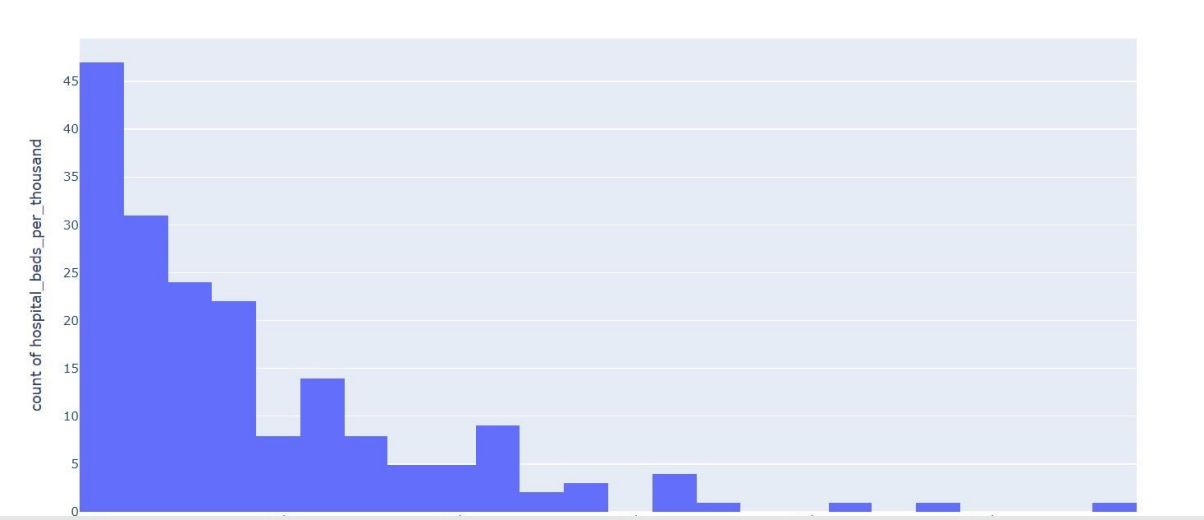
There are majority of countries that are affected by the virus in Group A. Hence, there is no conclusive finding when the countries were grouped based on their population.

Let us review the Trend based on the number of hospital beds per 1000 patients in each country.

6. Data Analysis - Hospital Beds per 1000 patients

For this case, we will be plotting histogram using GDP, Extreme Poverty index along with the number of hospital beds per 1000 patients in each country.

- Analysis of Poverty Ratio vs Hospital_Beds_per_1000:

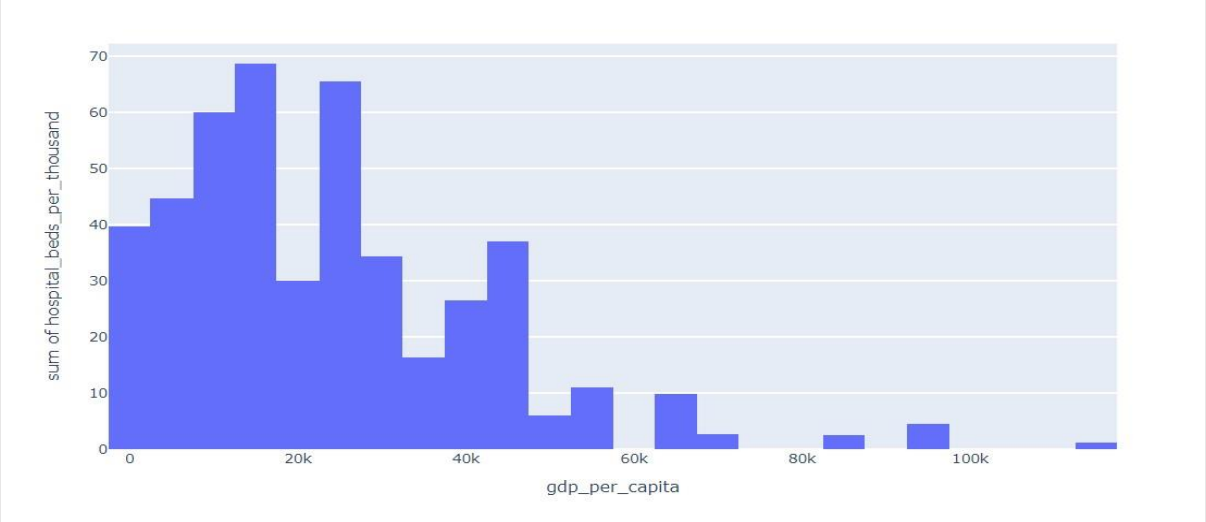


Histogram is Skewed Right. i.e., the Countries having least poverty ratio (comparatively better GDP per capita) have been impacted the most by the virus outbreak.

The most prominent reason is the International travels are at a higher scale in these countries.

The Countries having least poverty ratio have arranged and utilized majority of the hospital beds. However, Countries within range of 2.5 to 7.4 could only arrange an average of 34 beds per 1000 patients.

- Analysis of GDP_per_capita vs Hospital_Beds_per_1000



Countries with GDP per capita less have better resourcefulness towards hospital beds.

Key Finding:

In the above distribution plots, we could see that countries within range of extreme_poverty index 2.5 to 7.4 could only arrange an average of 34 beds per 1000 patients.
And also, the countries where GDP is above 47.5k have below 10 beds per 1000 patients where the average GDP of the countries around the world in 17.5k.

Filtering the countries that have either extreme poverty index between 2.5 to 7.4 or GDP more than 47.5k.
After applying the filters, we find that,
The total number of Countries with Higher GDP and Low Hospital_Beds_per_1000 is: 26

However, on adding Mortality rate due to COVID-19, The following countries have Mortality rate below 1 percent and the still active cases have reduced at a significant margin.
Hence, these countries won't have further huge virus impacts.

These countries include:

]:

	Country/Region	gdp_per_capita	Mortality_Rate_Covid19	Active_Cases
12	Tajikistan	2896.913	0.808871	24.722766
15	Kuwait	65530.537	0.669570	22.601185
4	Gabon	16562.413	0.654938	40.362142
25	United Arab Emirates	67293.483	0.572399	27.767807
19	Qatar	116935.600	0.158798	31.789958
22	Singapore	85535.383	0.049491	38.124828

Now, let us move to the countries where the number of beds per 1000 are less and also the mortality rate is above 1%.

These countries are as follows:

	Country/Region	gdp_per_capita	Mortality_Rate_Covid19	Active_Cases
3	Ecuador	10581.936	6.491269	48.911495
23	Switzerland	57410.166	5.497397	39.739116
6	Indonesia	11188.744	4.649146	31.677516
10	Peru	12236.706	4.484760	27.370561
11	Romania	23313.199	4.432162	45.490975
1	Brazil	14103.452	3.382068	20.025225
2	Colombia	13254.949	3.337620	42.888373
8	Nicaragua	5321.444	3.152230	30.702204
7	Mauritania	3597.633	2.436375	46.058349
9	Pakistan	5034.708	2.141111	38.544257
16	Luxembourg	94277.965	1.682454	40.576842
5	Georgia	9745.079	1.409619	30.431177
21	Saudi Arabia	49045.411	1.074849	22.174608

The Above Countries have the following attributes:

- Low Extreme Poverty index or Higher GDP per capita
- Low Hospital Beds per 1000 Patients
- Higher Mortality rate due to COVID-19
- But the number of active cases has reduced significantly

Hence, the above countries need to follow their current containment measures so that they do not get more impacted.

The remaining list of countries out of the filtered are as follows:

	Country/Region	gdp_per_capita	Mortality_Rate_Covid19	Active_Cases
17	Netherlands	48472.545	10.750892	88.848865
14	Ireland	67335.293	6.704080	56.453815
20	San Marino	56861.470	6.008584	67.954220
0	Bolivia	6885.829	4.009349	64.317369
24	US	54225.446	3.278413	63.986926
18	Norway	64800.057	2.703845	84.938741
13	Brunei	71809.251	2.127660	77.304965

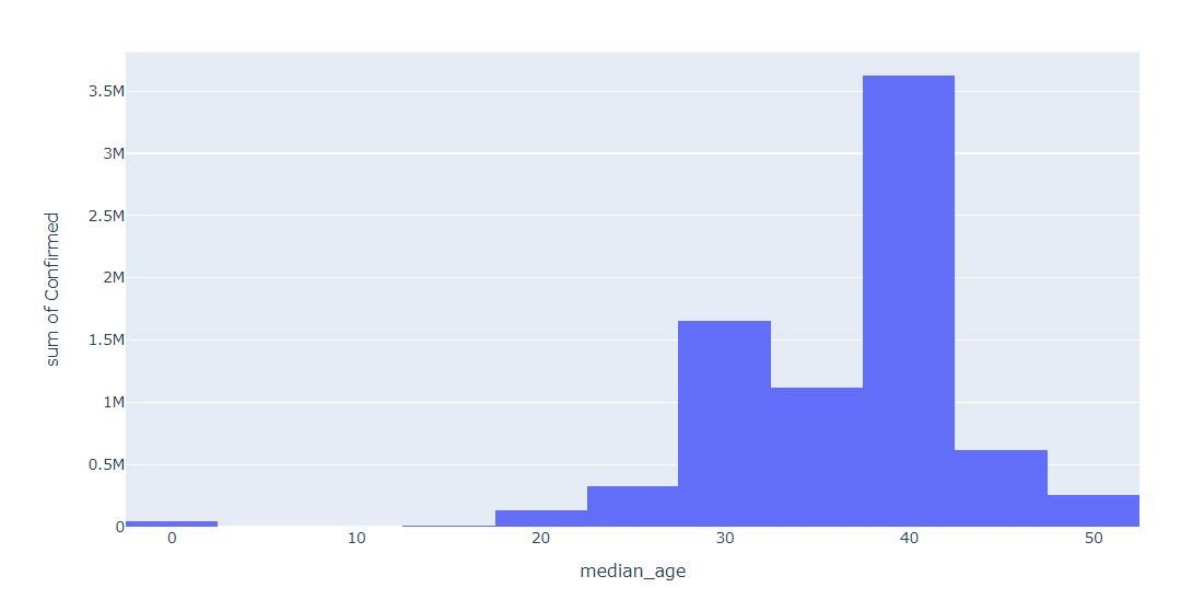
The Above Countries have the following attributes:

- Low Extreme Poverty index or Higher GDP per capita
- Low Hospital Beds per 1000 Patients
- High Mortality rate due to COVID-19
- High number of still active cases

Hence, the above countries require better containment measures to flatten the growth rate of the Virus.

7. Data Analysis - Median age of the population

For this case, we will be plotting histogram using Median age with the Confirmed cases and Deaths in each country.



Analysis of Median Age vs Confirmed Cases / Death cases:

- Countries with Median Age between 27 to 42 are highly affected
- It is safe to assume that the working-class age group is at high risk than rest of the population

Hence, until the growth rate curve of the virus flattens, Minimum work force needs to be deployed to get the daily business operations and the entrepreneurs need to come up with different solutions. For example, World-wide work from home have been implemented for IT jobs, etc.

8. Data Exploration and Visualization - Part 2

In this second stage of the Data exploration and visualization, we will focus on the spread of the COVID-19 in India.

The first case in India was reported on Jan 30th, 2020.

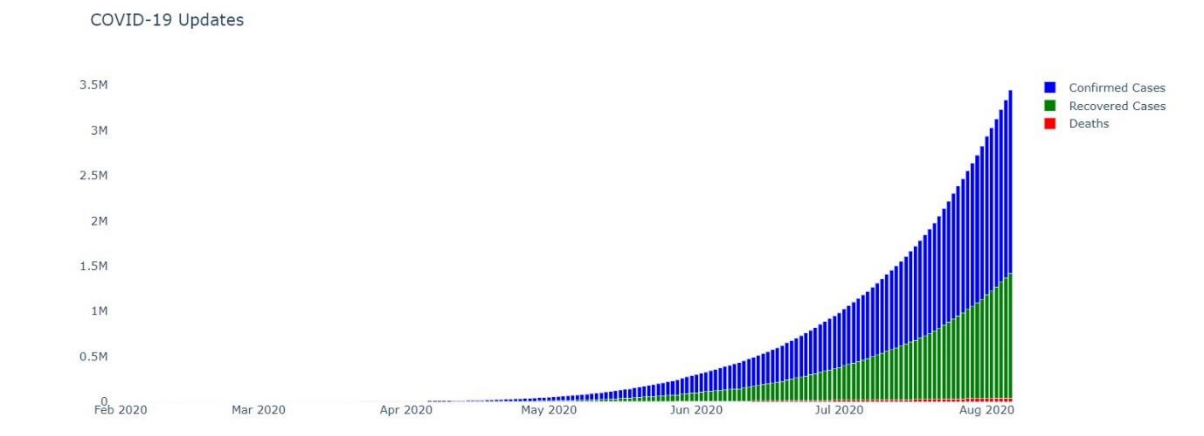
Let us view the descriptive statistics of the dataset for India.

	Confirmed	Deaths	Recovered	Active	New_Cases	New_Deaths	New_Recovered	total_tests	new_tests
count	137.000000	137.000000	137.000000	137.000000	137.000000	137.000000	137.000000	1.370000e+02	137.000000
mean	52478.051095	1549.510949	22467.379562	28461.160584	2342.496350	67.116788	1185.248175	1.006458e+06	39213.824818
std	84524.427505	2417.990491	40887.121884	41638.476189	3324.468523	105.086244	2135.665762	1.539870e+06	49993.269165
min	1.000000	0.000000	0.000000	0.000000	0.000000	-1.000000	0.000000	0.000000e+00	0.000000
25%	28.000000	0.000000	3.000000	25.000000	1.000000	0.000000	0.000000	0.000000e+00	0.000000
50%	5311.000000	150.000000	421.000000	4740.000000	601.000000	14.000000	46.000000	7.995000e+04	4280.000000
75%	70768.000000	2294.000000	22549.000000	45925.000000	3607.000000	104.000000	1445.000000	1.609037e+06	84835.000000
max	320922.000000	9195.000000	162379.000000	149348.000000	11929.000000	748.000000	11989.000000	5.507182e+06	151808.000000

Here is the correlation matrix between the columns.

	Confirmed	Deaths	Recovered	Active	New_Cases	New_Deaths	New_Recovered	total_tests	new_tests
Confirmed	1.000000	0.999023	0.994749	0.995147	0.936301	0.833373	0.843770	0.939299	0.868724
Deaths	0.999023	1.000000	0.990054	0.997716	0.940074	0.841668	0.843317	0.942033	0.880863
Recovered	0.994749	0.990054	1.000000	0.979851	0.914173	0.810335	0.837141	0.918079	0.824567
Active	0.995147	0.997716	0.979851	1.000000	0.948386	0.847125	0.841812	0.950521	0.902635
New_Cases	0.936301	0.940074	0.914173	0.948386	1.000000	0.880451	0.884966	0.900670	0.882704
New_Deaths	0.833373	0.841668	0.810335	0.847125	0.880451	1.000000	0.655469	0.803952	0.785452
New_Recovered	0.843770	0.843317	0.837141	0.841812	0.884966	0.655469	1.000000	0.789608	0.753590
total_tests	0.939299	0.942033	0.918079	0.950521	0.900670	0.803952	0.789608	1.000000	0.953921
new_tests	0.868724	0.880863	0.824567	0.902635	0.882704	0.785452	0.753590	0.953921	1.000000

Let us plot the growth of COVID-19 in India.



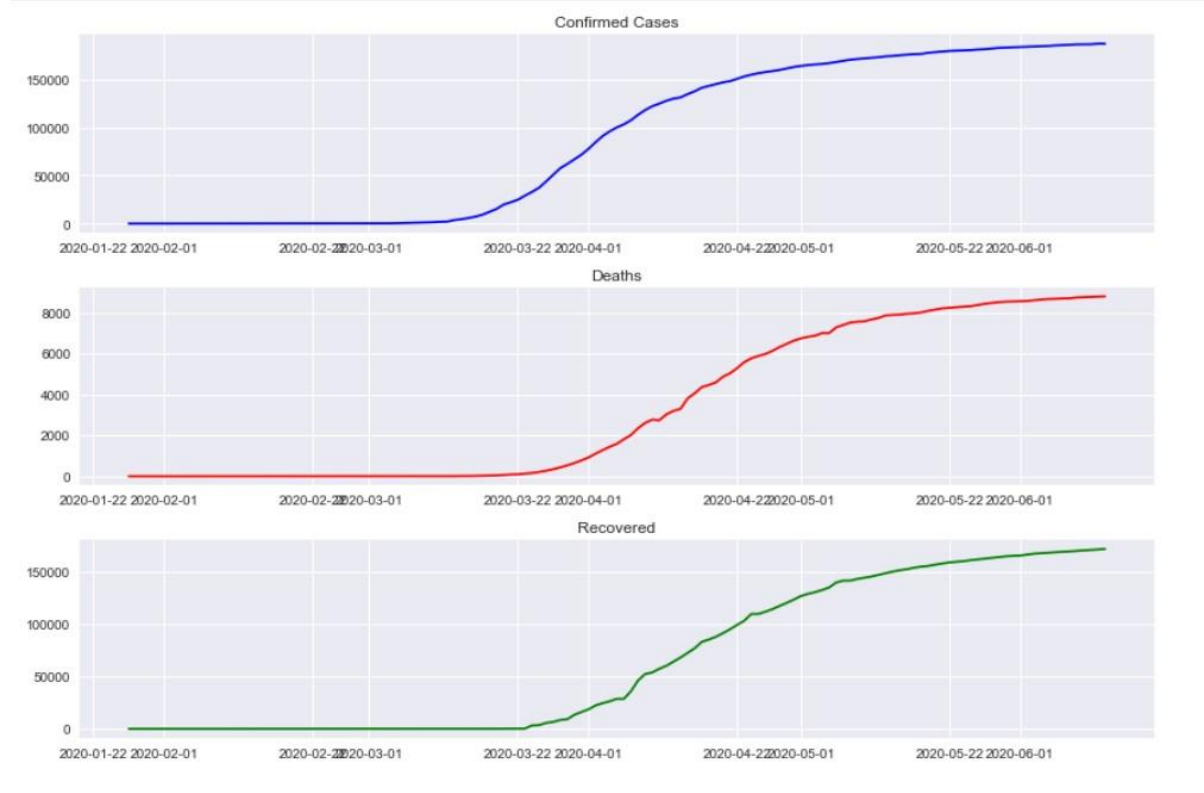
8. Predictive Modelling

In this section, we are going to perform a Non-linear regression on the growth rate of the Confirmed cases in India.
By definition, Regression analysis is a powerful statistical method that allows you to examine the relationship between two or more variables of interest.

While there are many types of regression analysis, at their core they all examine the influence of one or more independent variables on a dependent variable.

To perform a Non-linear regression, we must first figure out a curve that will be used to fit the model.
In the section 4. Data visualization, we had plotted the countries where the confirmed cases have crossed 100k cases. India was also included in that group. So it is logical to plot the graphs of the same group and find out which countries have started to show a flattened curve.

I have already plotted all the cases, and found out that Germany shows a good “sigmoid” curve as shown below.



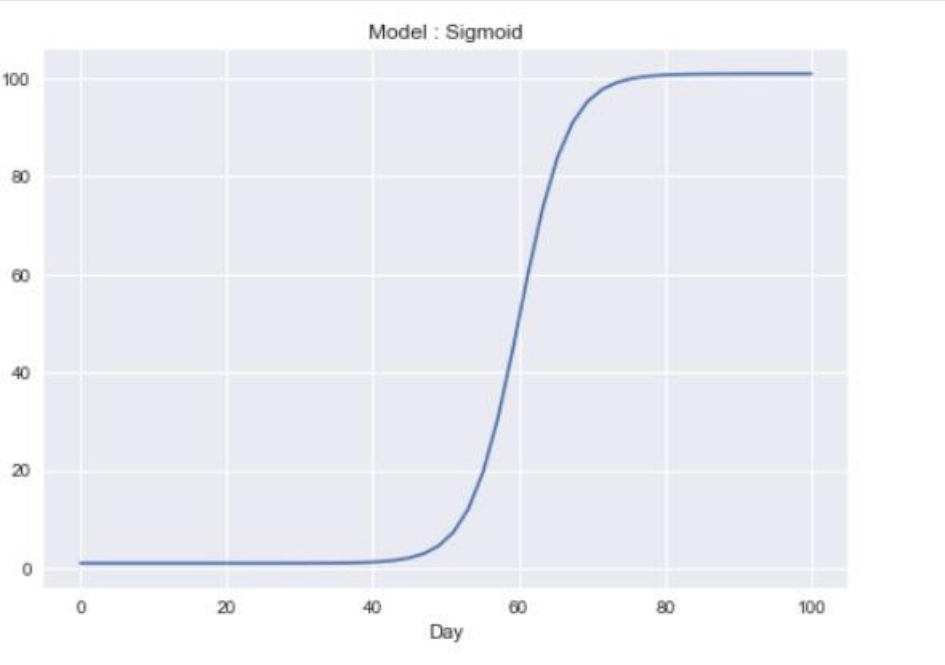
Hence, we will be using a sigmoid curve to fit the model.

Sigmoid Curve:

A sigmoid function is a mathematical function having a characteristic "S"-shaped curve or sigmoid curve.
A common example of a sigmoid function is the logistic function shown in the first figure and defined by the formula:

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$

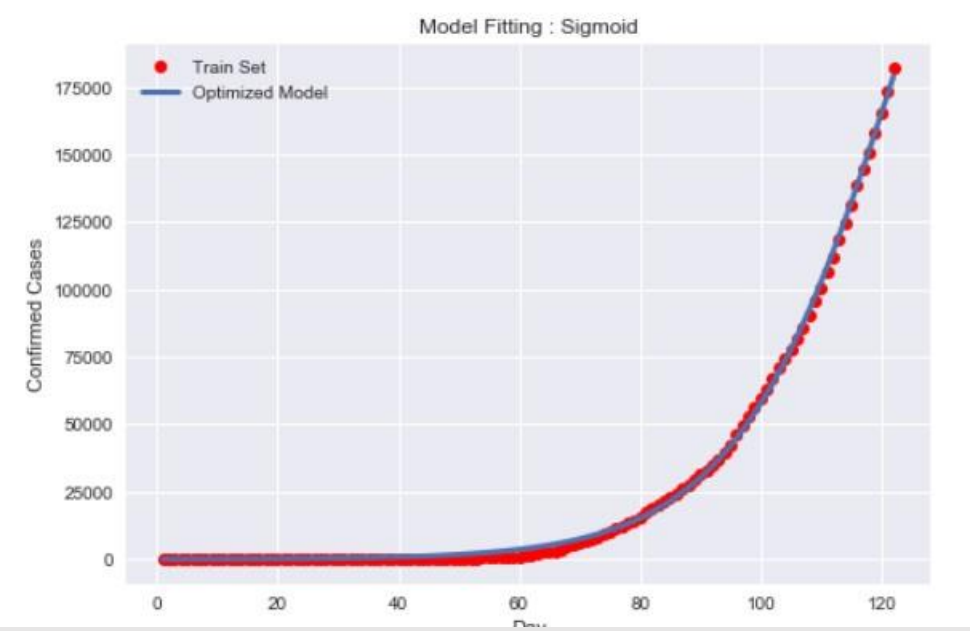
Plotting a sigmoid curve:



Model Train – Test – Split:

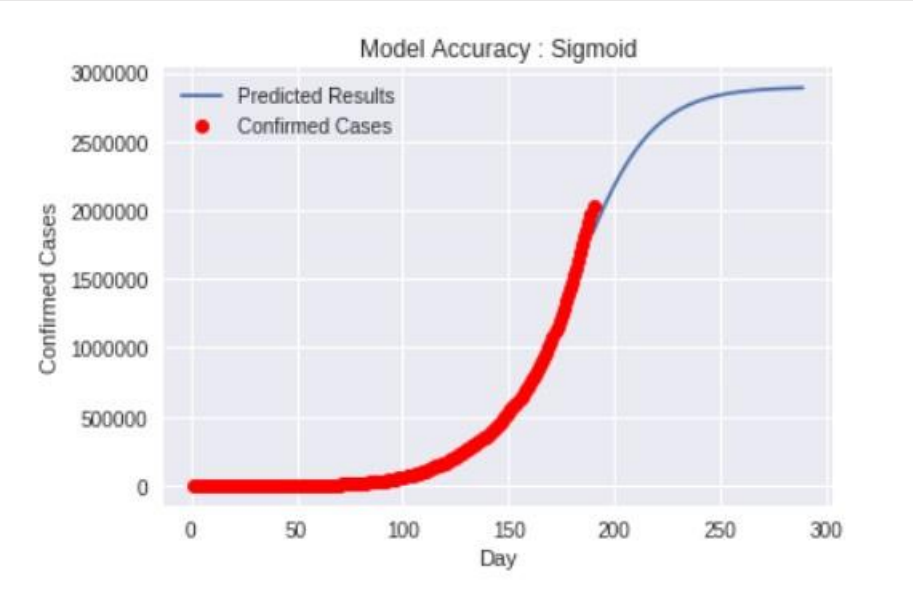
We will be carrying out our regression analysis with a 90/10 split between training set and testing set of the data.
SkLearn library function train_test_split() will be used to carry out the split and the SciPy library function scipy.optimize.curve_fit() function will be used to fit the model on the dataset.

The fitted model looks as shown below:



9. Evaluation:

The predicted model is as follows:

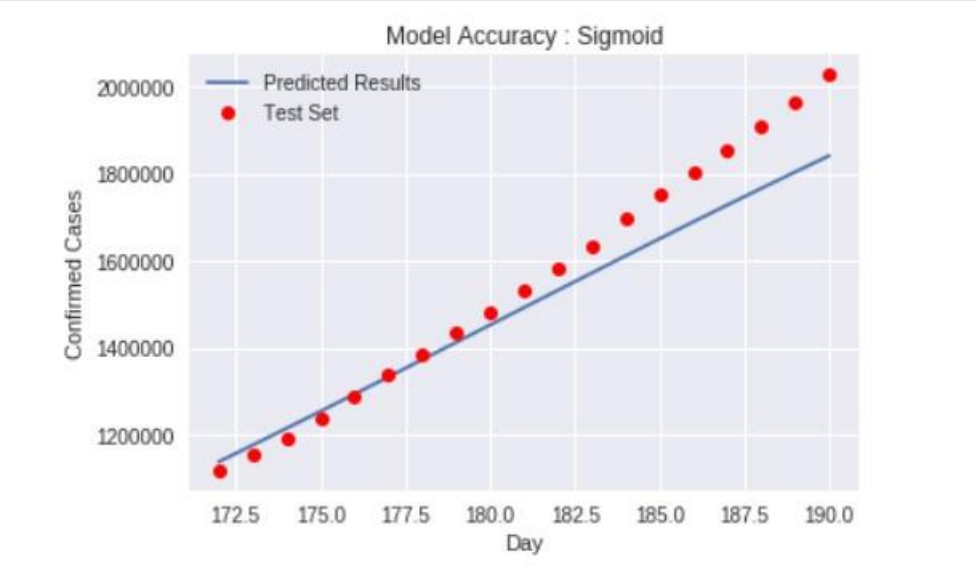


The evaluation metrics used in this case is R-Squared method:

The evaluation of the model is as shown below:

```
Mean absolute error: 64225.54
Residual sum of squares (MSE): 7287114736.08
R2-score: 0.91
```

The Model accuracy of the test set is **91%** close to the actual values.



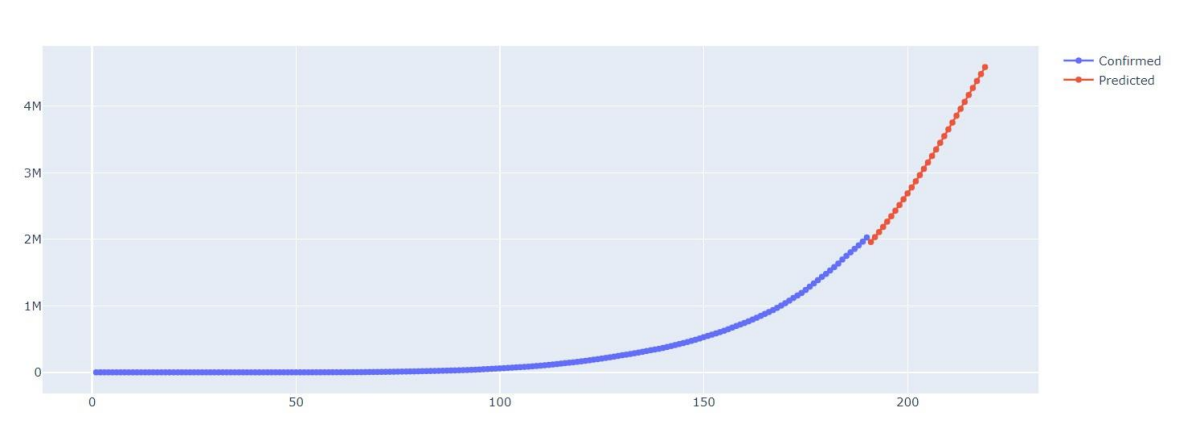
However, it is important to know the daily errors that have been occurring.

Limitations:

- The Train and Test split was done at 90%/10% splits. For every 5% increase in the test size, it is observed that the R-squared metrics fall to 0.85 and so forth.
  Hence, it is certain that the prediction model cannot be very accurate (like 80 and above) post 30 days of test size.
- If the very same analysis were to be conducted for any other country, scipy.optimize.curve_fit() function would need to be calibrated again.

10. Prediction for next 30 days:

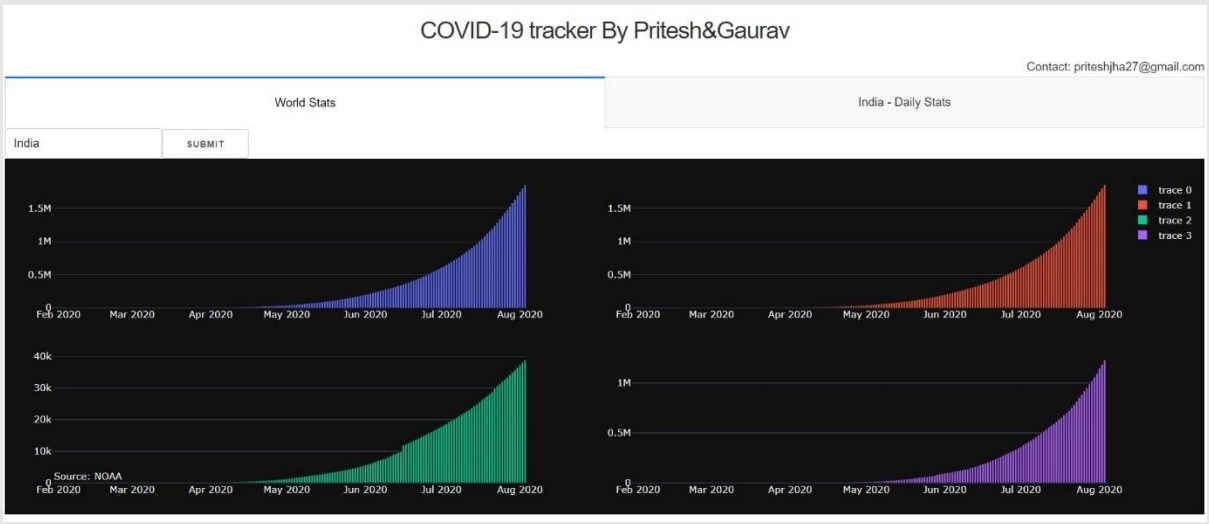
Now let us use the 100% dataset as the training set and plot the data according to our model for the next 30 days.
This can also be reviewed using the Web app provided.

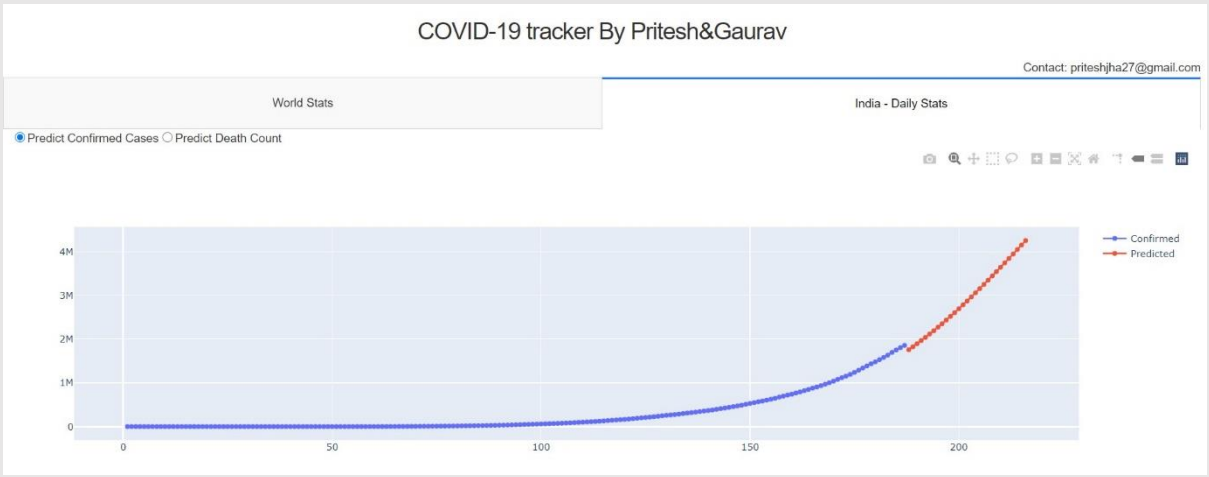


11. Web app deployment:

The dash app has been deployed via Docker Container registry using Web app services from Azure.
The application allows the user to Find the cumulative stats for any country and also tracks the Prediction model based on the previous model.

Here is the layout of the Web Dashboard:





12. Finding Essential services nearby:

This section has been added in the project for the users to provide an address as input and find any nearby essential services like:

1. Hospitals
2. Pharmacy
3. Police stations

The project uses Foursquare API to locate the essential services and displays the locations on the folium maps.
Here is the link to Foursquare webpage : https://developer.foursquare.com/places

Markers shown in the map are as follows:

- User Location – Blue Cloud
- Hospitals & Pharmacy – Red info icons
- Police Stations – Green info icons

The Notebook presentation is as follows:

Finding Essentials Near Your Place :

```
address = str(input('Address : '))
```

```
Address : Sector 20, Kharghar, Navi Mumbai, India
```