Data Analysis Techniques:

Right now, we have a dataset of 900 days (dynamic dataset which grows each day) from 49 European countries in-total. The data is being read from a GitHub repository[1] of Our World in Data (a scientific online publication)

Terminology Definition:

Global (max/min) value: This value is the maximum (or minimum) value of a particular criterion (such as new cases) out of all the countries under consideration in a data frame.

Local (max/min) value: This value is the maximum (or minimum) value of a particular criterion (such as new cases) for each country separately.

Moving Average Method:

At first, when we were directly plotting the points from our dataset, it was becoming difficult for us to comprehend how the trend was emerging out to be as there were continuous high frequency of up and down patterns as shown in fig (1)— marked with red circle. Therefore, we thought of using the moving average method on our dataset before plotting.

In moving average method, we took a specified number of past days, averaged those days, updated our database with that new average point and then plotted it. For instance, in a 7-day moving average, we are taking average of the last 7 days' worth of data points and then plotting it. In our plot, our starting date is around 10th January. So, we took the average of data points from 10th Jan to 17th Jan and updated the 17th of Jan data point with that new average value. Then, we moved one day and took the average from 11th Jan to 18th Jan (called a window) and updated the 18th of Jan with the new average value. We repeated this until we reached the end of the database.
One thing to notice is that in this newly updated database, the first 7 values would not be updated with an average value. Therefore, we need to drop those values from our updated database. The resulting plot for 7 days moving average is shown in fig (2).

Moreover, if we increase the number of moving days to let's say 14, then we can observe that our plot will become smoother. See, fig (3), for 14 days moving average plot. In this, we are taking average of past 14 days in each window and in the end dropping the first 14 days.

It is strongly recommended to take the number of moving days to be in the multiple of 7 as this ensures that the frequency of the number of each day in each window is coming out to be the same, i.e., if your moving days are 14 then it ensures that you will always get 2 Mondays, 2 Tuesdays or 2 any other day in your moving day window of 14 days. As Covid data is being logged in by humans, and laboratories are closed on Sundays.[2]

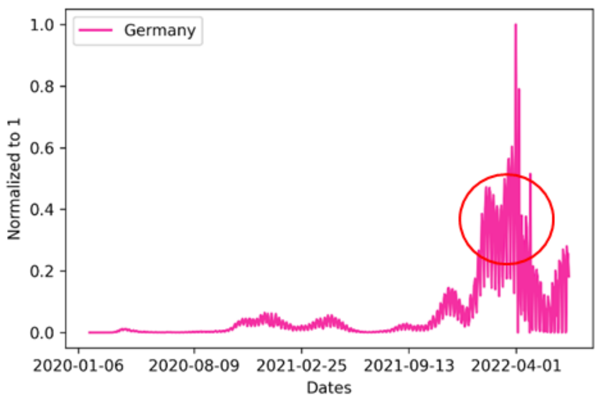


Figure 1: Plot without taking moving average

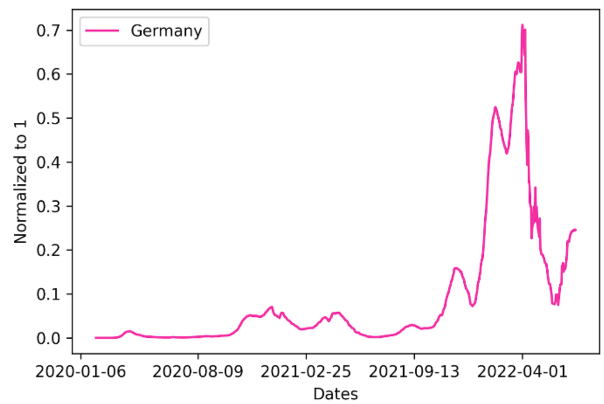


Figure 2: Plot after taking 7 days of moving average

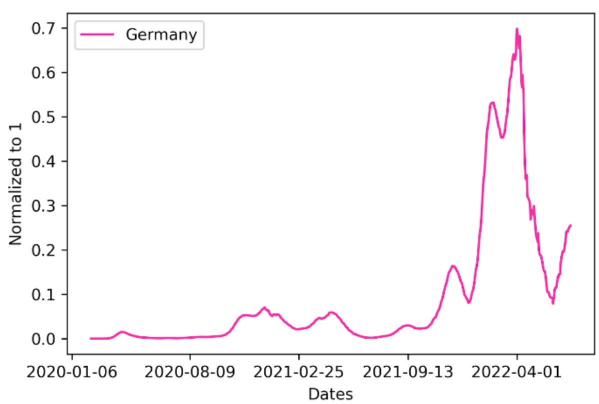


Figure 3: Plot after taking 14 days of moving average

The Analysis Criteria:

We are currently employing four criterions for analyzing the European covid data set:

1. **New Cases**: The data point in this criterion shows the number of new covid cases each day.
2. **New Deaths:** The data point in this criterion shows the number of deaths resulting from covid each day.
3. **Number of Hospitalizations:** The data point in this criterion shows the total number of hospitalizations on each day.
4. **Number of ICU patients:** The data point in this criterion shows the total number of ICU patients each day.

Data Cleaning:

- **Removing Countries having no data recorded value of specific criterion (such as new cases):** Out of the 49 countries many countries have not recorded for all criteria. For example, Germany does not have recorded data for the criterion of 'New Cases.' Given Below is the list of countries which have recorded data in respective criteria.

    - **New Cases:**

      Albania, Andorra, Austria, Belarus, Belgium, Bosnia and Herzegovina, Bulgaria, Croatia, Cyprus, Czechia, Denmark, Estonia, Faeroe Islands, Finland, France, Germany, Gibraltar, Greece, Hungary, Iceland, Ireland, Isle of Man, Italy, Kosovo, Latvia, Liechtenstein, Lithuania, Luxembourg, Malta, Moldova, Monaco, Montenegro, Netherlands, North Macedonia, Norway, Poland, Portugal, Romania, Russia, San Marino, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Ukraine, United Kingdom, Vatican,

    - **New Deaths:**

      Albania, Andorra, Austria, Belarus, Belgium, Bosnia and Herzegovina, Bulgaria, Croatia, Cyprus, Czechia, Denmark, Estonia, Faeroe Islands, Finland, France, Germany, Gibraltar, Greece, Hungary, Iceland, Ireland, Isle of Man, Italy, Kosovo, Latvia, Liechtenstein, Lithuania, Luxembourg, Malta, Moldova, Monaco, Montenegro, Netherlands, North Macedonia, Norway, Poland, Portugal, Romania, Russia, Sano, Se, San Marino, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Ukraine, United Kingdom,

    - **Number of Hospitalizations:**

      Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Denmark, Estonia, Finland, France, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, United Kingdom,

    - **Number of ICU Patients:**

Austria, Belgium, Bulgaria, Cyprus, Czechia, Denmark, Estonia, Finland, France, Germany, Iceland, Ireland, Italy, Latvia, Luxembourg, Malta, Netherlands, Portugal, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, United Kingdom.

- **Using interpolation to bridge the gaps in the data:** Due to missing data points from many countries, we performed interpolation to make the data continuous. Currently, we are employing linear interpolation. The plot of the countries [‘Germany’, ‘France’ and ‘Italy’] through the technique of local dynamic mean maximum before and after interpolation depicts the results.
  Linear interpolation in Pandas fills the spaces with the values that are equally spaced and does not give weightage to the index values.

```
In [36]:  df = second_df.iloc[:,4]

Out[36]: 0     25.0
         1     50.0
         2      NaN
         3      NaN
         4      NaN
         5    100.0
         6     75.0
         7     25.0
         8      NaN
```

Figure 4. Data Frame without interpolation

```
In [37]:  df = second_df.iloc[:,4]
          df.interpolate(limit_area = "inside")

Out[37]: 0     25.0
         1     50.0
         2     62.5
         3     75.0
         4     87.5
         5    100.0
         6     75.0
         7     25.0
         8      NaN
```

Figure 5. Data Frame with linear interpolation

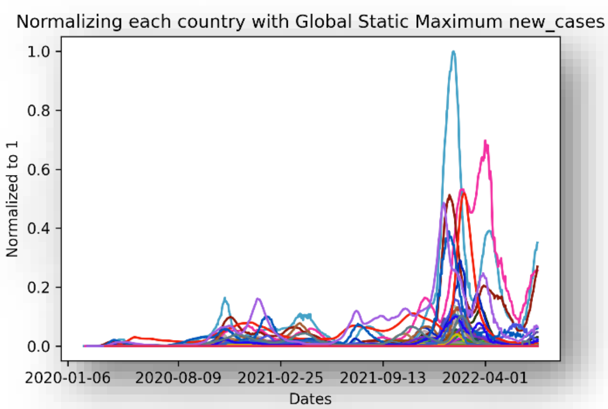
Normalization Techniques:

**(All these normalization techniques have been applied on the dataset after taking the rolling moving average). See footnote for more info.*

Techniques using Global Normalizations:

- **Technique 1**: Global static maximum: In this technique, each country's data point for each day in the dataset was divided by the global max value. (In our dataset, France had the global max value*)

Below is the plot of all the 49 countries, for the parameter **new cases.**



To get more clarity from the above plot, we have divided it into three sub plots based on the maximum normalized value for each country.

```
if x <= 0.05:
    small_data_country_index.append(colname)

elif x > 0.05 and x <= 0.2:
    medium_data_country_index.append(colname)

else:
    large_data_country_index.append(colname)
```

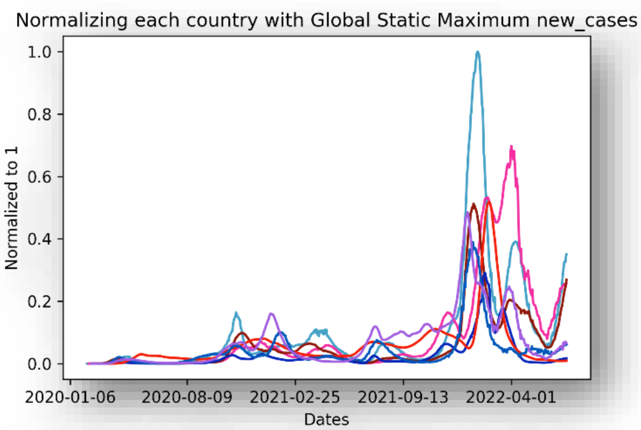
For instance, Austria has the maximum normalization value of 0.127403 out of all data points in its column from the data frame. As this value comes between 0.05 and 0.2, we have put this country in the medium_data_country_index list. Similarly, we have segregated the rest of the 48 countries. The thresholds were decided after considering different data visualizations of this dataset.

*(All these normalization techniques have been applied on the dataset after taking the rolling moving average). In the original data set, Germany had the highest peak but after taking the rolling moving average France had the highest value because it sustained a higher value than Germany over a longer period.

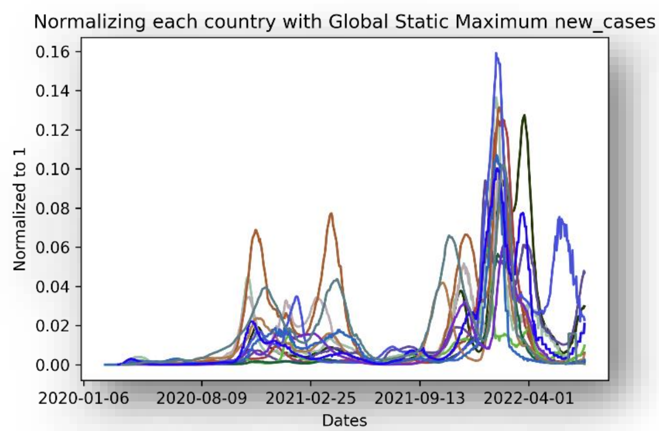
Large Max Value Country Dataset: ['France', 'Germany', 'Italy', 'Netherlands', 'Russia', 'Spain', 'United Kingdom']

Medium Max Value Country Dataset: ['Austria', 'Belgium', 'Czechia', 'Denmark', 'Greece', 'Ireland', 'Norway', 'Poland', 'Portugal', 'Romania', 'Slovakia', 'Sweden', 'Switzerland', 'Ukraine']

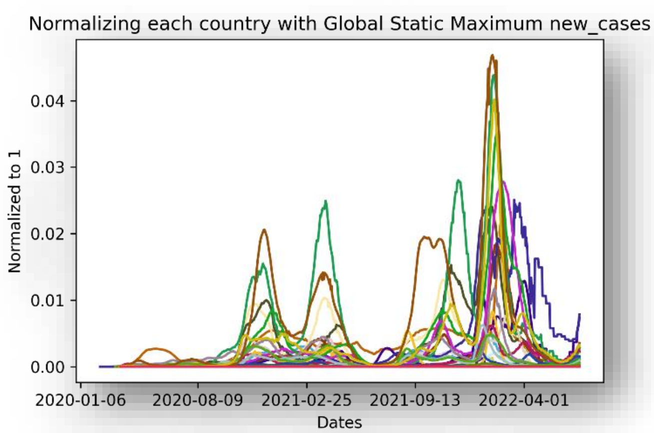
Small Max Value Country Dataset: ['Albania', 'Andorra', 'Belarus', 'Bosnia and Herzegovina', 'Bulgaria', 'Croatia', 'Cyprus', 'Estonia', 'Faeroe Islands', 'Finland', 'Gibraltar', 'Hungary', 'Iceland', 'Isle of Man', 'Kosovo', 'Latvia', 'Liechtenstein', 'Lithuania', 'Luxembourg', 'Malta', 'Moldova', 'Monaco', 'Montenegro', 'North Macedonia', 'San Marino', 'Serbia', 'Slovenia', 'Vatican']



Large Max Value Country Dataset

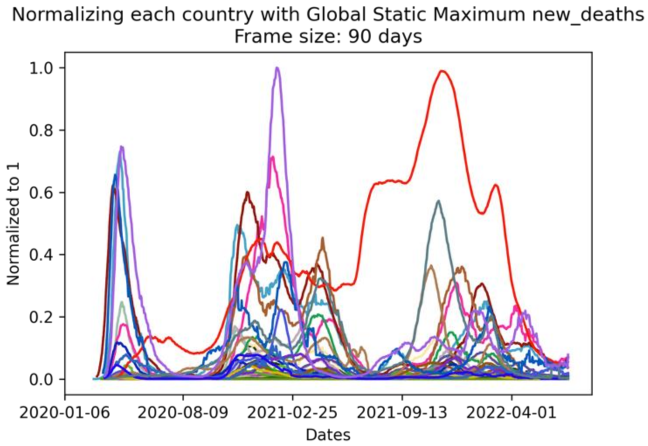


Medium Max Value Country Dataset



Small Max Value Country Dataset

Below is the plot of all the <u>countries</u>, for the parameter **New Deaths**

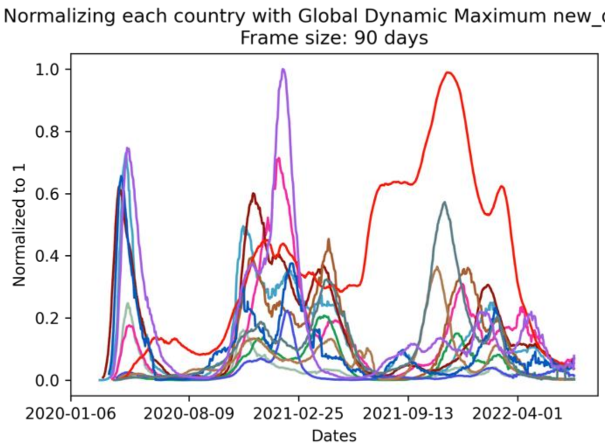


Large Max Value Country Dataset: ['Belgium', 'France', 'Germany', 'Hungary', 'Italy', 'Poland', 'Portugal', 'Romania', 'Russia', 'Spain', 'Ukraine', 'United Kingdom']

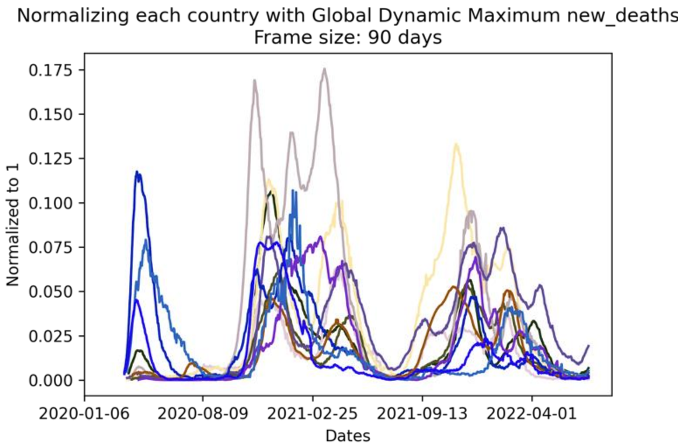
Medium Max Value Country Dataset: ['Austria', 'Bosnia and Herzegovina', 'Bulgaria', 'Croatia', 'Czechia', 'Greece', 'Netherlands', 'Serbia', 'Slovakia', 'Sweden', 'Switzerland']

Small Max Value Country Dataset: ['Albania', 'Andorra', 'Belarus', 'Cyprus', 'Denmark', 'Estonia', 'Faeroe Islands', 'Finland', 'Gibraltar', 'Iceland', 'Ireland', 'Isle of Man', 'Kosovo', 'Latvia', 'Liechtenstein', 'Lithuania', 'Luxembourg', 'Malta', 'Moldova', 'Monaco', 'Montenegro', 'North Macedonia', 'Norway', 'San Marino', 'Slovenia']

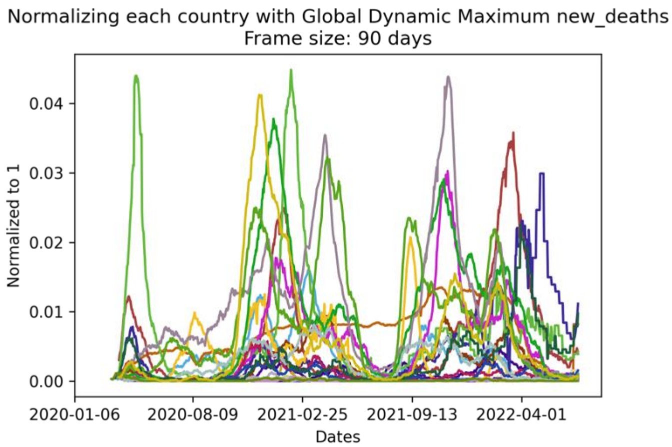
Three subplots:



Large Max Value Country Dataset



Medium Max Value Country Dataset



Small Max Value Country Dataset

Technique 2: Global dynamic maximum

Definition and important points:

- **Max Global Val** is the maximum global value out of all the max global values that we got from each frame for each day.
- **Max array** is an array of size 900 by 1 (rows = length of the entire date series and column = 1). It stores the Max Global Val for each day of our data set.
- **Span size**: The number of data elements under consideration while finding the Max Global Value for a particular date
- The value of the window frame can be changed to suit our data visualization purposes.

Method:
In this technique, instead of taking the global max value from the entire data series of 900 days (about 2 and a half years), we are sliding (by shifting one day) a window frame of 90 days (about 3 months) through the entire dataset [top to down as in our database on GitHub dates are rows and countries are columns]. We are doing this because the technology and infrastructure against Covid 19 has increased manifold compared to the starting point of this pandemic which was 2 years ago. Therefore, we need to look for the global max value for each data point in its vicinity.

For instance, consider the 90th data point (let us say x) in our data set which is also the end point of our first sliding frame of 90 days (see the above figure for reference).
For this frame, we will:

- **first** find the global max value (let us say G1),
- **update** each row in the max array
- then **slide** our window by 1 day to move forward to the next frame.

In the next frame, we will again:

- **find** the global max value (let us say G2),
- **update** each row in the max array (if and only if G2 > G1) and
- then **slide** our window again by one day.

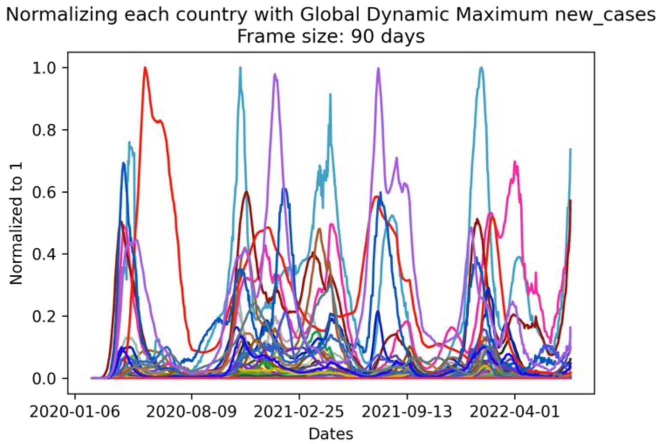
We will repeat this till we reach the starting point of the last sliding frame.

Notice, that for last frame, x was its 89^{th} data point. By the time x becomes the 1^{st} data point of some frame in one of the sliding windows, its corresponding value in the max array would be out of approximately 180 days (about 6 months): 90 days before in the starting window frame plus 90 days after in the ending window frame.

Therefore, every data point's corresponding final value in the max array is found after at most 180 turns. In short, for each data point, we are finding the Max Global Val after evaluation of at most 180 global max values for that data point.

After finalizing the max array, we just normalize by dividing each row (date) for all the countries in the data set with the corresponding Max Global Val in the max array.

The resulting normalized data set has been plotted below.
Below is the plot of all the countries, for the parameter **New cases.**

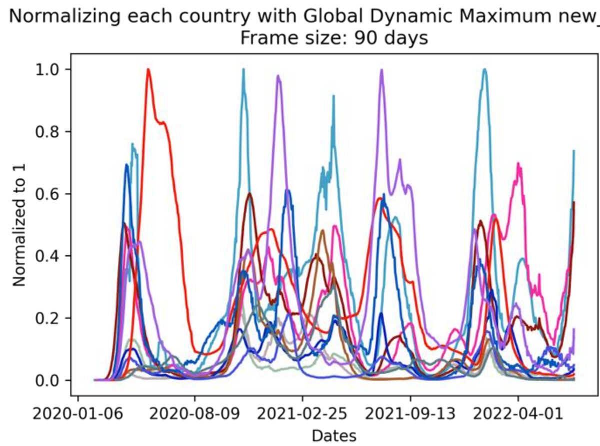


To get more clarity from the above plot, we have divided it into three sub plots based on the maximum normalized value for each country— like the way we did in the last technique. (The threshold value has also not been changed)

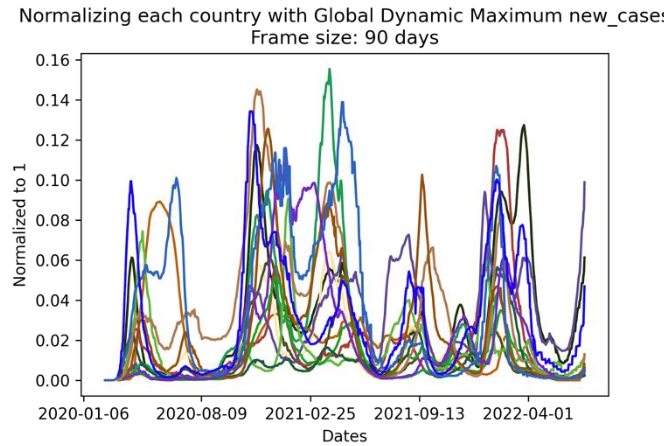
Large Max Value Country Dataset: ['Belgium', 'Czechia', 'France', 'Germany', 'Italy', 'Netherlands', 'Poland', 'Portugal', 'Russia', 'Spain', 'Ukraine', 'United Kingdom']

Medium Max Value Country Dataset: ['Austria', 'Belarus', 'Bulgaria', 'Croatia', 'Denmark', 'Greece', 'Hungary', 'Ireland', 'Lithuania', 'Norway', 'Romania', 'Serbia', 'Slovakia', 'Sweden', 'Switzerland']

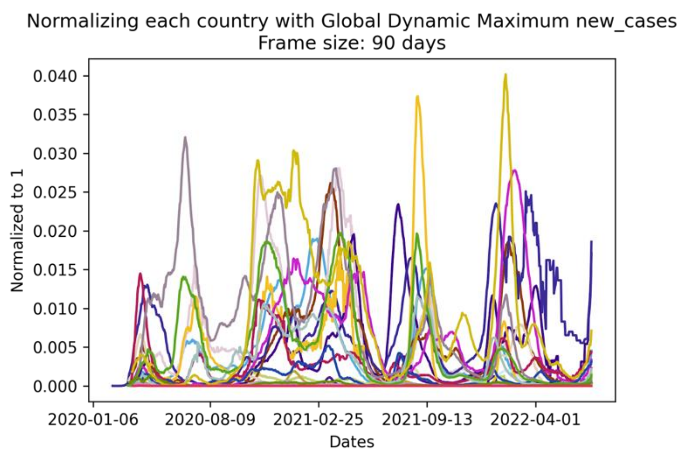
Small Max Value Country Dataset: ['Albania', 'Andorra', 'Bosnia and Herzegovina', 'Cyprus', 'Estonia', 'Faeroe Islands', 'Finland', 'Gibraltar', 'Iceland', 'Isle of Man', 'Kosovo', 'Latvia', 'Liechtenstein', 'Luxembourg', 'Malta', 'Moldova', 'Monaco', 'Montenegro', 'North Macedonia', 'San Marino', 'Slovenia', 'Vatican']



Large Max Value Country Dataset

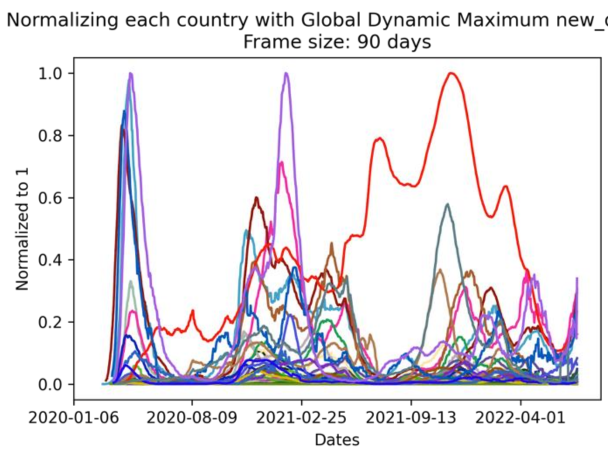


Medium Max Value Country Dataset



Small Max Value Country Dataset

Below is the plot of all the countries, for the parameter **New Deaths**

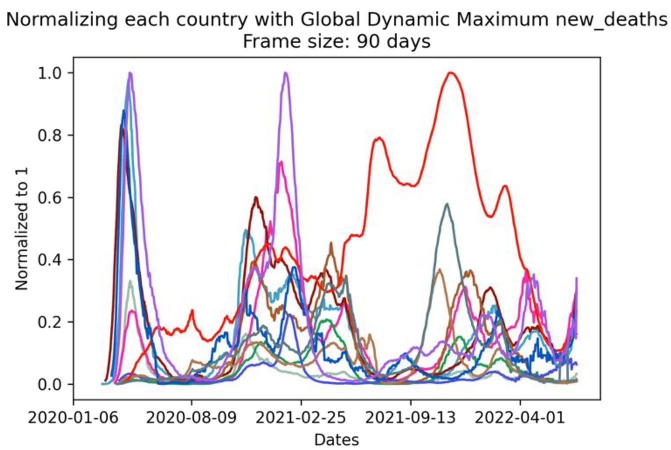


Large Max Value Country Dataset: ['Belgium', 'France', 'Germany', 'Hungary', 'Italy', 'Poland', 'Portugal', 'Romania', 'Russia', 'Spain', 'Ukraine', 'United Kingdom']

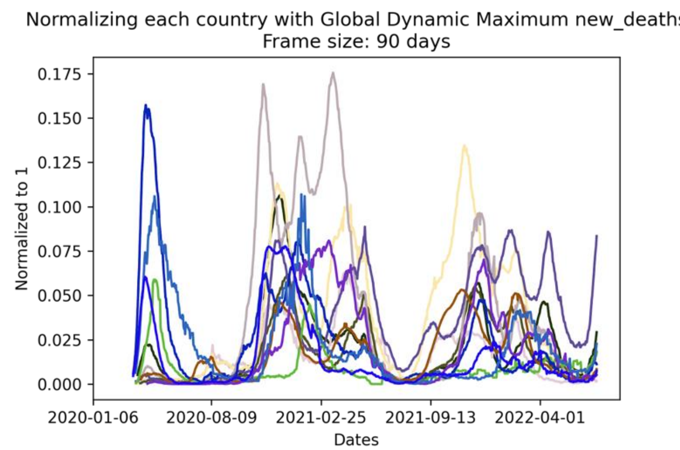
Medium Max Value Country Dataset: ['Austria', 'Bosnia and Herzegovina', 'Bulgaria', 'Croatia', 'Czechia', 'Greece', 'Ireland', 'Netherlands', 'Serbia', 'Slovakia', 'Sweden', 'Switzerland']

Small Max Value Country Dataset: ['Albania', 'Andorra', 'Belarus', 'Cyprus', 'Denmark', 'Estonia', 'Faeroe Islands', 'Finland', 'Gibraltar', 'Iceland', 'Isle of Man', 'Kosovo', 'Latvia', 'Liechtenstein', 'Lithuania', 'Luxembourg', 'Malta', 'Moldova', 'Monaco', 'Montenegro', 'North Macedonia', 'Norway', 'San Marino', 'Slovenia']

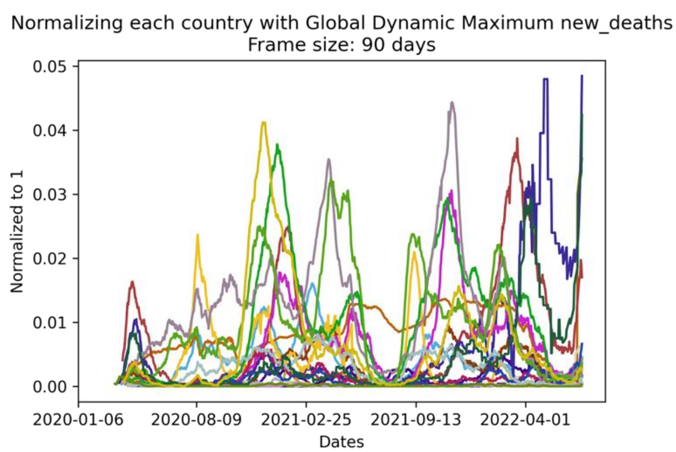
Three sub plots



Large Max Value Country Dataset

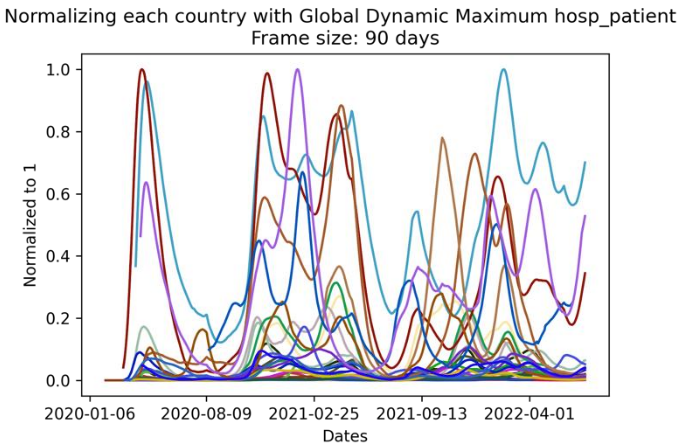


Medium Max Value Country Dataset



Small Max Value Country Dataset

Below is the plot of all the <u>countries</u>, for the parameter **New Hospital Patients:**

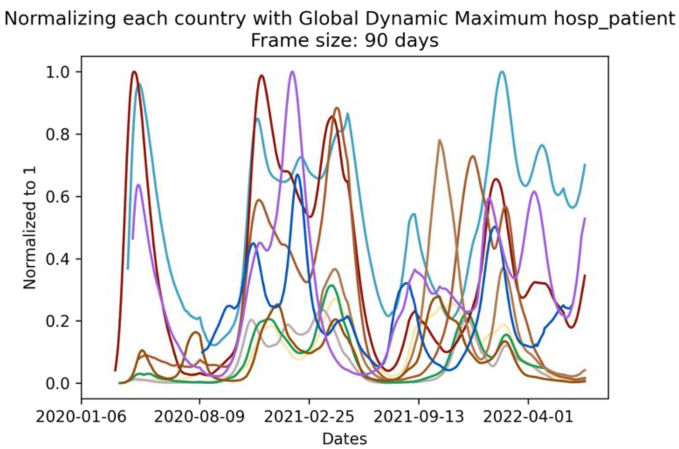


Large Max Value Country Dataset: ['Bulgaria', 'Czechia', 'France', 'Hungary', 'Italy', 'Poland', 'Romania', 'Serbia', 'Spain', 'United Kingdom']

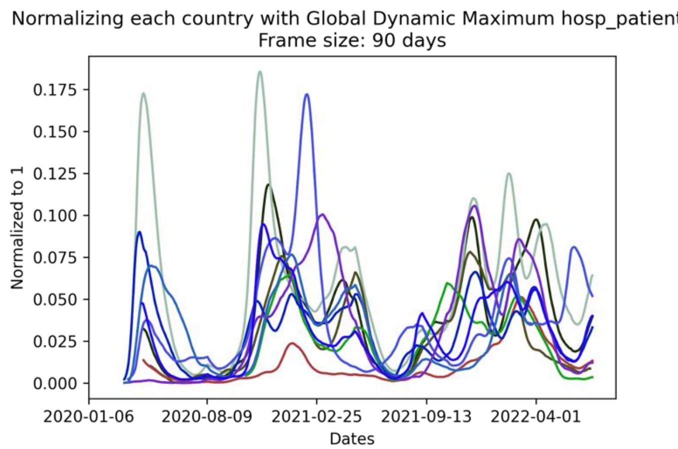
Medium Max Value Country Dataset: ['Austria', 'Belgium', 'Croatia', 'Denmark', 'Lithuania', 'Netherlands', 'Portugal', 'Slovakia', 'Sweden', 'Switzerland']

Small Max Value Country Dataset: ['Cyprus', 'Estonia', 'Finland', 'Iceland', 'Ireland', 'Latvia', 'Luxembourg', 'Malta', 'Norway', 'Slovenia']

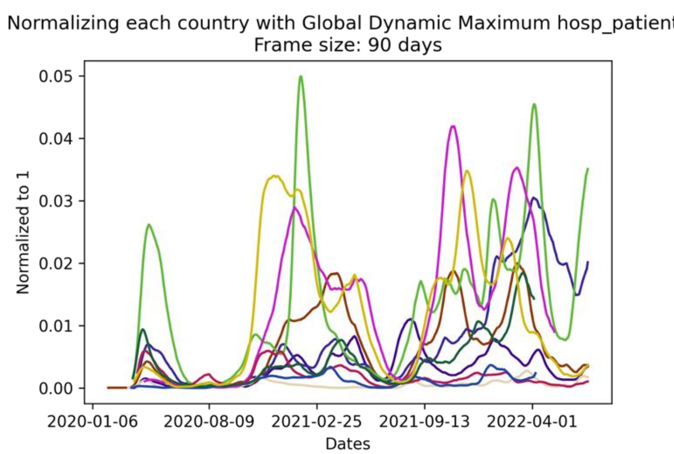
Three sub plots:



Large Max Value Country Dataset

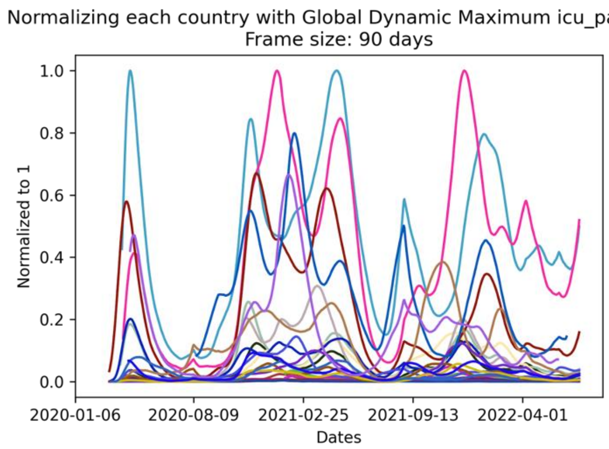


Medium Max Value Country Dataset



Small Max Value Country Dataset

Below is the plot of all the <u>countries</u>, for the parameter **ICU Patients:**

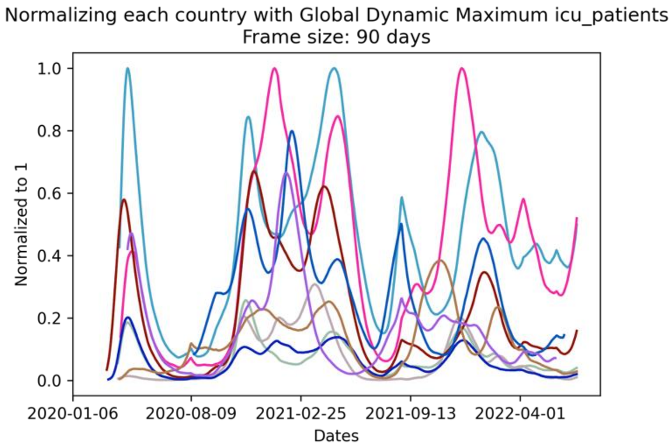


Large Max Value Country Dataset: ['Belgium', 'Czechia', 'France', 'Germany', 'Italy', 'Netherlands', 'Romania', 'Spain', 'United Kingdom']

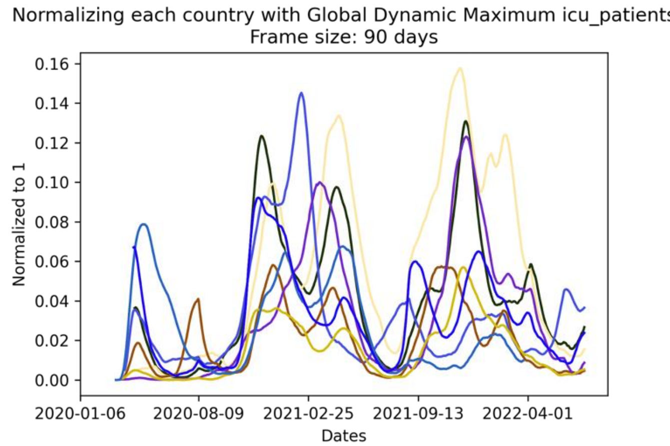
Medium Max Value Country Dataset: ['Austria', 'Bulgaria', 'Portugal', 'Serbia', 'Slovakia', 'Slovenia', 'Sweden', 'Switzerland']

Small Max Value Country Dataset: ['Cyprus', 'Denmark', 'Estonia', 'Finland', 'Iceland', 'Ireland', 'Latvia', 'Luxembourg', 'Malta']

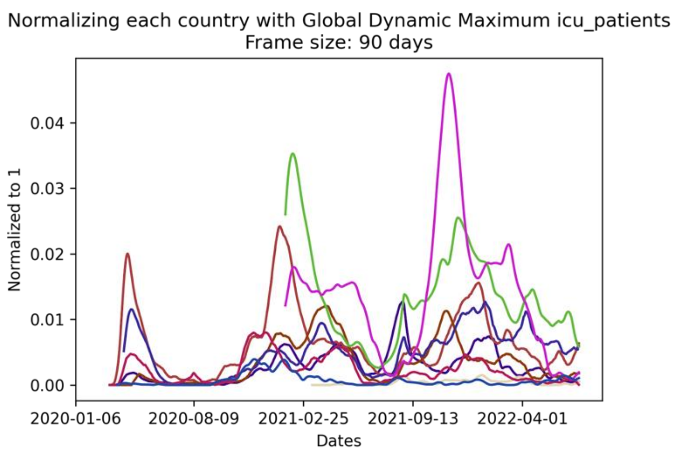
Three sub plots:



Large Max Value Country Dataset



Medium Max Value Country Dataset



Small Max Value Country Dataset

- **Technique 3:** Global dynamic mean maximum

Definition and important points:

Like in the previous technique, in this technique also we will use the concept of sliding window. One key difference here is that we have two data frames.

- One frame in which all the values for each date and for each country is stored (let us name it D1).
- Second frame in which the size of the data frame is same as the first one, but it is empty (let us name it D2): ready to store new values.

Here, we are taking sliding window size to be 120 days. So, for the first sliding frame of 120 days in D1, we will: (see the above figure for reference)

- **find** the global max value within that frame (let us say G1),
- **divide** each row in the window frame with that value G1
- **add** each row of D1 in the window frame with the corresponding empty rows of D2 and **update** those rows' values
- then **slide** our window by 1 day to move forward to the next frame.

In the next frame of D1, we will again:

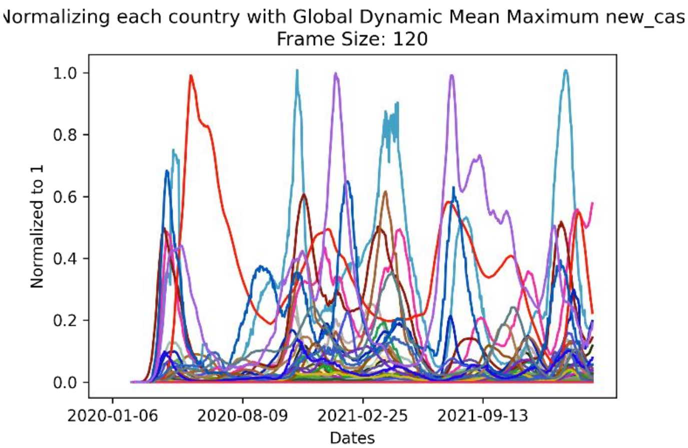
- **find** the global max value within that frame (let us say G2),
- **divide** each row in the window frame with that value G2
- **add** each row of D1 in the window frame with the corresponding rows of D2 and **update** those rows' values
- then **slide** our window again by one day.

Apart from the elements in the first and last sliding frame of D2, each element (let us say x) in the rest of the frames will be the summation of 120 different or same normalised values (equivalent to the window size). For instance, the summation could be like this, $\frac{x}{G_1} + \frac{x}{G_2} + \cdots + \frac{x}{G_{119}} + \frac{x}{G_{120}}$ where each value of G_s ($S \in \{n: 1 \le n \le 120\ and\ n\ \in N\}$) can be same or different.

Our last step is to **divide** each row of D2 by the window size (except the element of the first and last sliding frames) *

At the end, each element (x) of D2 data frame will contain the mean value of all the normalised values for that point. One example of that value could be $\frac{\frac{x}{G_1}+\frac{x}{G_2}+\cdots+\frac{x}{G_{119}}+\frac{x}{G_{120}}}{120}$.

The resulting normalized data set has been plotted below.
Below is the plot of all the countries, for the parameter **New cases.**



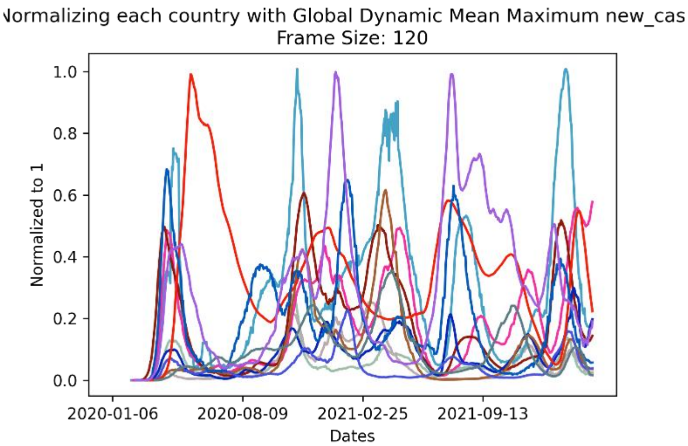
To get more clarity from the above plot, we have divided it into three sub plots based on the maximum normalized value for each country— like the way we did in the last technique. (The threshold value has also not been changed)

Large Max Value Country Dataset: ['Belgium', 'Czechia', 'France', 'Germany', 'Italy', 'Netherlands', 'Poland', 'Portugal', 'Russia', 'Spain', 'Ukraine', 'United Kingdom']

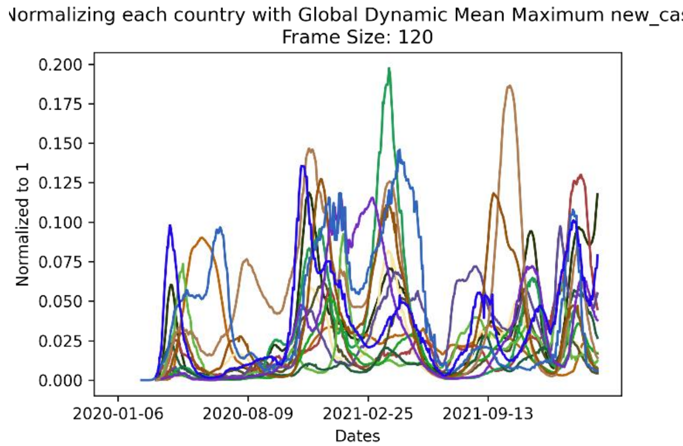
Medium Max Value Country Dataset: ['Austria', 'Belarus', 'Bulgaria', 'Croatia', 'Denmark', 'Greece', 'Hungary', 'Ireland', 'Lithuania', 'Norway', 'Romania', 'Serbia', 'Slovakia', 'Sweden', 'Switzerland']

Small Max Value Country Dataset: ['Albania', 'Andorra', 'Bosnia and Herzegovina', 'Cyprus', 'Estonia', 'Faeroe Islands', 'Finland', 'Gibraltar', 'Iceland', 'Isle of Man', 'Kosovo', 'Latvia', 'Liechtenstein', 'Luxembourg', 'Malta', 'Moldova', 'Monaco', 'Montenegro', 'North Macedonia', 'San Marino', 'Slovenia', 'Vatican']

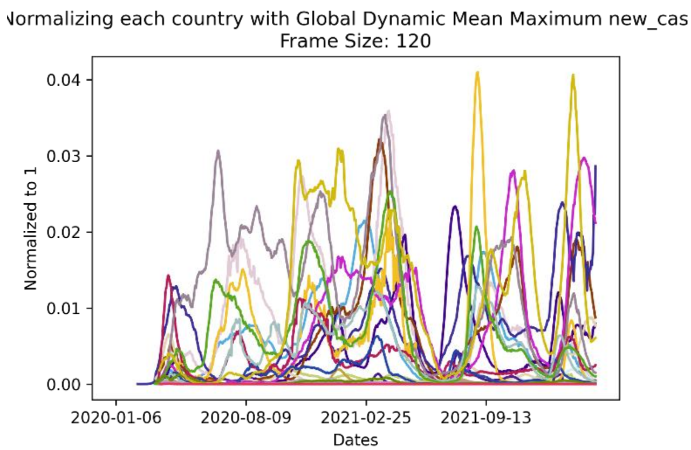
*For the first sliding window's rows, we need to appropriately consider the number of elements that have been summed and then divide. For the last sliding frame, we must drop those rows because the window frame size starts becoming smaller as we start approaching to the end of the data frame.



Large Max Value Country Dataset

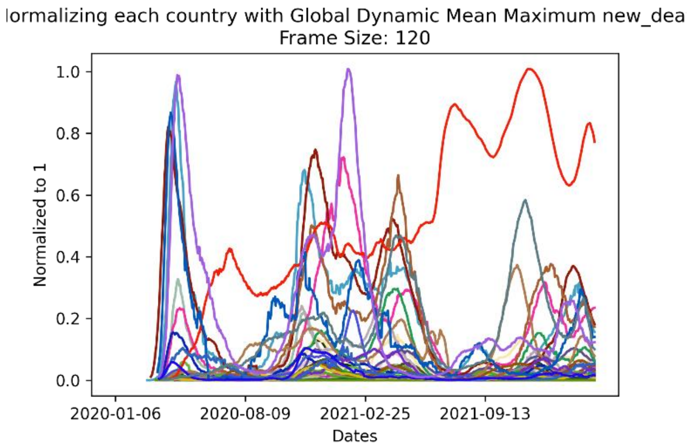


Medium Max Value Country Dataset



Small Max Value Country Dataset

Below is the plot of all the countries, for the parameter **New Deaths.**

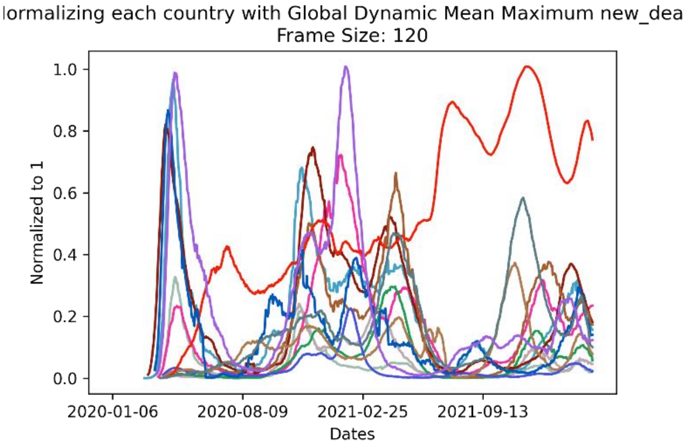


Large Max Value Country Dataset: ['Belgium', 'Czechia', 'France', 'Germany', 'Hungary', 'Italy', 'Poland', 'Portugal', 'Romania', 'Russia', 'Spain', 'Ukraine', 'United Kingdom']

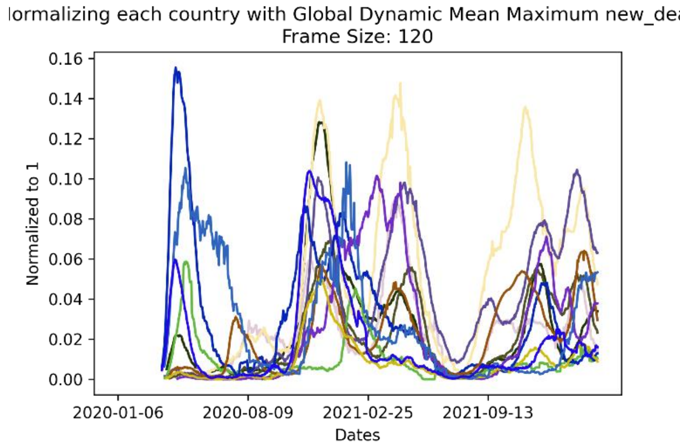
Medium Max Value Country Dataset: ['Austria', 'Bosnia and Herzegovina', 'Bulgaria', 'Croatia', 'Greece', 'Ireland', 'Netherlands', 'Serbia', 'Slovakia', 'Slovenia', 'Sweden', 'Switzerland']

Small Max Value Country Dataset: ['Albania', 'Andorra', 'Belarus', 'Cyprus', 'Denmark', 'Estonia', 'Faeroe Islands', 'Finland', 'Gibraltar', 'Iceland', 'Isle of Man', 'Kosovo', 'Latvia', 'Liechtenstein', 'Lithuania', 'Luxembourg', 'Malta', 'Moldova', 'Monaco', 'Montenegro', 'North Macedonia', 'Norway', 'San Marino']

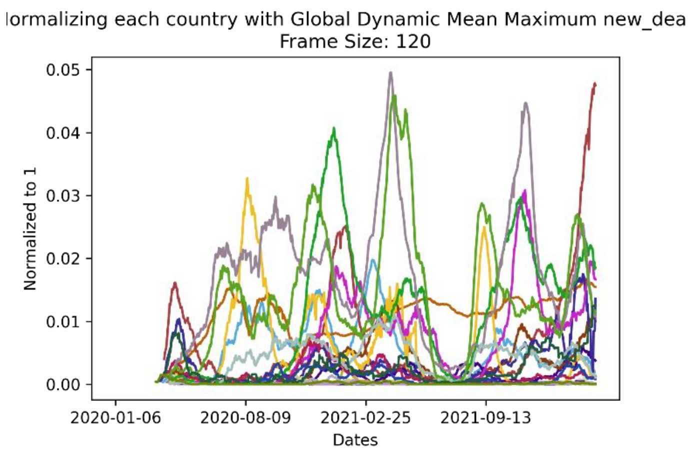
Three subplots:



Large Max Value Country Dataset

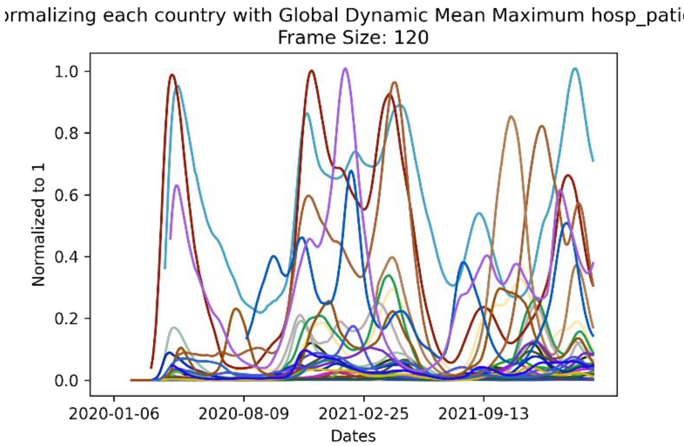


Medium Max Value Country Dataset



Small Max Value Country Dataset

Below is the plot of all the <u>countries</u>, for the parameter **Hospital Patients**

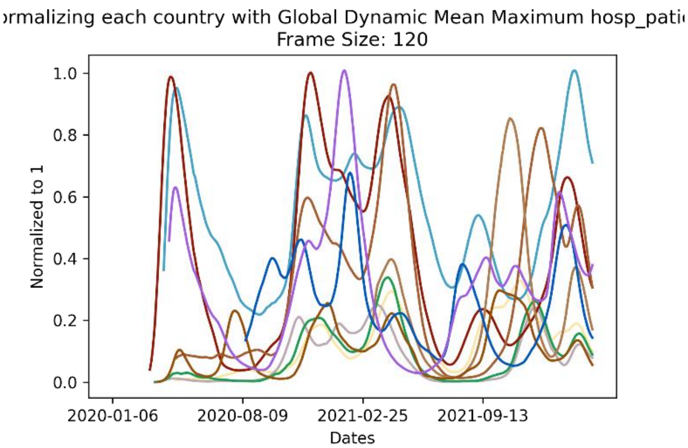


Large Max Value Country Dataset: `['Bulgaria', 'Czechia', 'France', 'Hungary', 'Italy', 'Poland', 'Romania', 'Serbia', 'Spain', 'United Kingdom']`

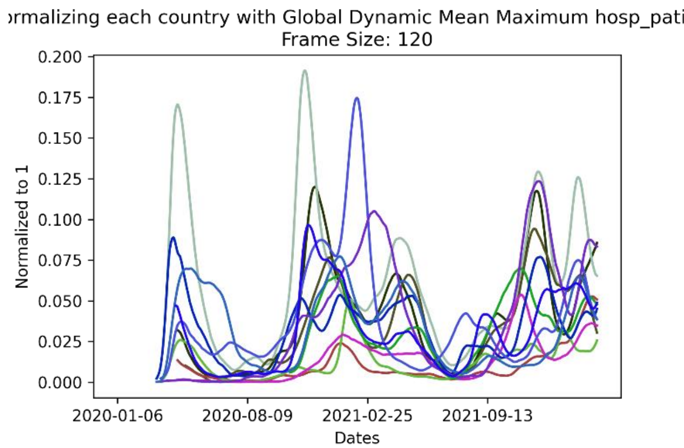
Medium Max Value Country Dataset: `['Austria', 'Belgium', 'Croatia', 'Denmark', 'Ireland', 'Latvia', 'Lithuania', 'Netherlands', 'Portugal', 'Slovakia', 'Sweden', 'Switzerland']`

Small Max Value Country Dataset: `['Cyprus', 'Estonia', 'Finland', 'Iceland', 'Luxembourg', 'Malta', 'Norway', 'Slovenia']`

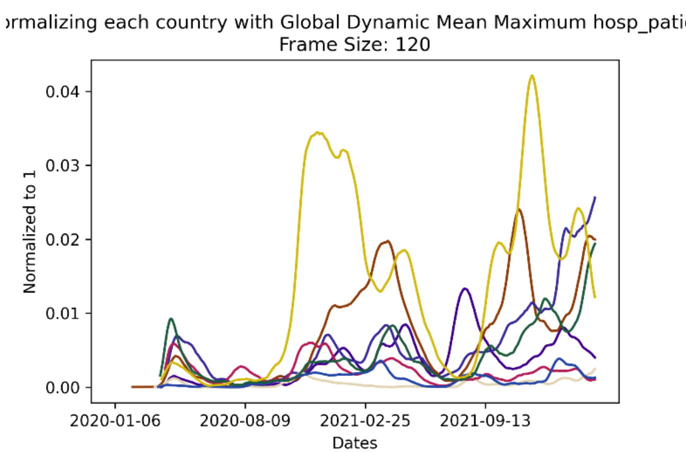
Three subplots:



Large Max Value Country Dataset

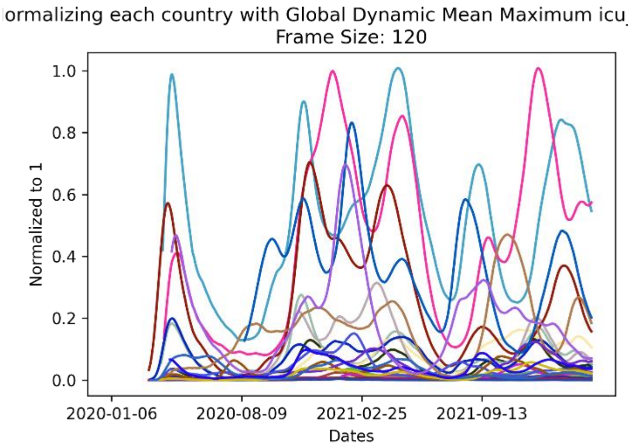


Medium Max Value Country Dataset



Small Max Value Country Dataset

Below is the plot of all the <u>countries</u>, for the parameter **ICU Patients**

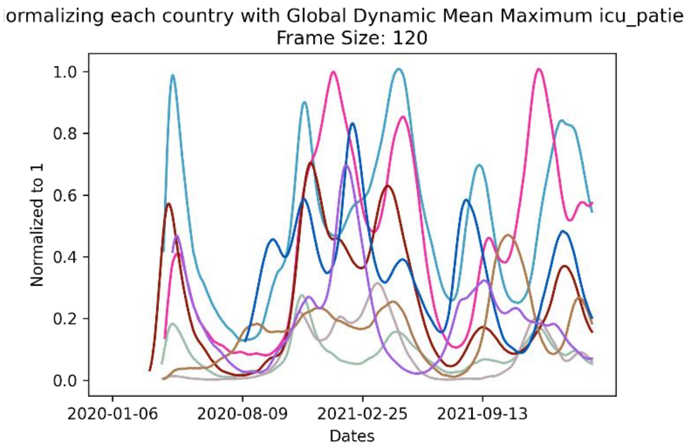


Three sub plots:

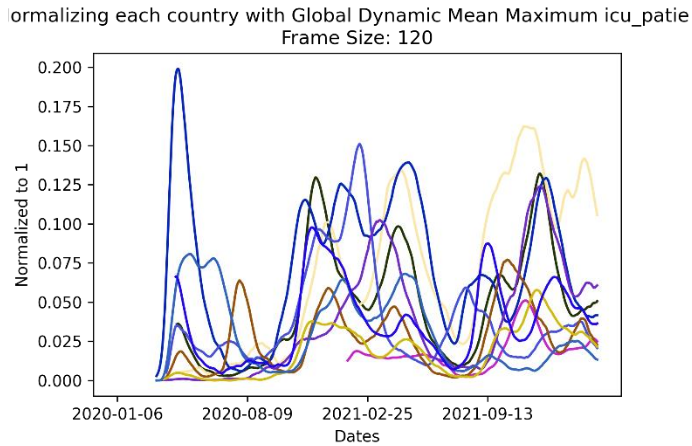
`Large Max Value Country Dataset:` `['Belgium', 'Czechia', 'France', 'Germany', 'Italy', 'Romania', 'Spain', 'United Kingdom']`

`Medium Max Value Country Dataset:` `['Austria', 'Bulgaria', 'Latvia', 'Netherlands', 'Portugal', 'Serbia', 'Slovakia', 'Slovenia', 'Sweden', 'Switzerland']`

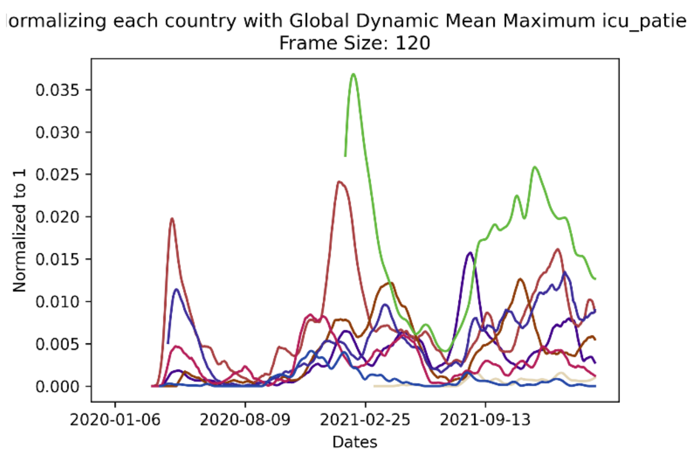
`Small Max Value Country Dataset:` `['Cyprus', 'Denmark', 'Estonia', 'Finland', 'Iceland', 'Ireland', 'Luxembourg', 'Malta']`



Large Max Value Country Dataset



Medium Max Value Country Dataset



Small Max Value Country Dataset

Techniques using Local Normalization:

The plots in global techniques have been separated into three:
1. Large Country Dataset
2. Medium Country Dataset
3. Small Country Dataset

The logic behind these separations is that the globally normalized data points of countries lie in different sets such as between (0,0.05) or between (0,0.2) depending on the population of the country. Such logic cannot guide the separation of graphs in local techniques as each country's data will be normalized to its own maximum. However, here we are experimenting to separate the graph into four sections by the three techniques:
1. Actual population of European countries.
2. Density of European Countries
3. Segregating countries by giving half weightage to populations and half weightage to density.

This might show us whether countries with smaller populations follow the same trend as with the larger countries or not.

Following is a detailed view of the techniques of segregation.

1. **Segregating by Actual Population of European Countries:** Four Groups were created in ascending order (smallest to largest) of populations
    a. **Group 1:**
        Vatican City, Gibraltar, San Marino, Liechtenstein, Monaco, Faroe Islands, Andorra, Isle of Man, Iceland, Malta, Montenegro, Luxembourg
    b. **Group 2:**
        Cyprus, Estonia, Latvia, Slovenia, North Macedonia, Lithuania, Albania, Bosnia and Herzegovina, Moldova, Croatia, Ireland, Slovakia
    c. **Group 3:**
        Norway, Finland, Denmark, Bulgaria, Serbia, Switzerland, Austria, Belarus, Hungary, Portugal, Sweden, Greece
    d. **Group 4:**
        Czech Republic, Belgium, Netherlands, Romania, Poland, Ukraine, Spain, Italy, France, United Kingdom, Germany, Russia
2. **Segregating by Density of European Countries:** Four Groups were created in ascending order (smallest to largest) of density of populations
    a. **Group 1:**
        Iceland, Russia, Finland, Norway, Sweden, Latvia, Estonia, Faroe Islands, Lithuania, Belarus, Montenegro, Bulgaria
    b. **Group 2:**
        Bosnia and Herzegovina, Ireland, Ukraine, Croatia, Greece, Romania, North Macedonia, Spain, Serbia, Albania, Slovenia, Hungary
    c. **Group 3:**

Austria, Portugal, Slovakia, Moldova, France, Poland, Cyprus, Denmark, Czech Republic, Isle of Man, Andorra, Italy

d. **Group 4:**
   Switzerland, Germany, Liechtenstein, Luxembourg, United Kingdom, Belgium, Netherlands, San Marino, Malta, Vatican City, Gibraltar, Monaco

3. **Segregating countries by giving half weightage to populations and half weightage to density:** Four Groups were created in ascending order (smallest to largest). Giving half weightage to both population and population density countries were sorted. The resultant are the four groups.
   a. **Group 1:**
      Faroe Islands, Iceland, Montenegro, Isle of Man, Andorra, Estonia, Liechtenstein, Latvia, Cyprus, Luxembourg, North Macedonia, Slovenia
   b. **Group 2:**
      Lithuania, Albania, Bosnia and Herzegovina, San Marino, Croatia, Moldova, Ireland, Norway, Finland, Slovakia, Denmark, Bulgaria
   c. **Group 3:**
      Serbia, Belarus, Austria, Switzerland, Hungary, Sweden, Malta, Greece, Portugal, Czech Republic, Vatican City, Belgium
   d. **Group 4:**
      Romania, Netherlands, Poland, Gibraltar, Ukraine, Spain, Italy, France, United Kingdom, Germany, Monaco, Russia

- **Technique 4**: Local static maximum: In this technique for each specific criterion the data for each country was divided by its own maximum (of the duration of the recorded data) (approximately 900 days (about 2 and a half years)).

  The plots by the three segregating techniques are given below in the form of OneDrive links.

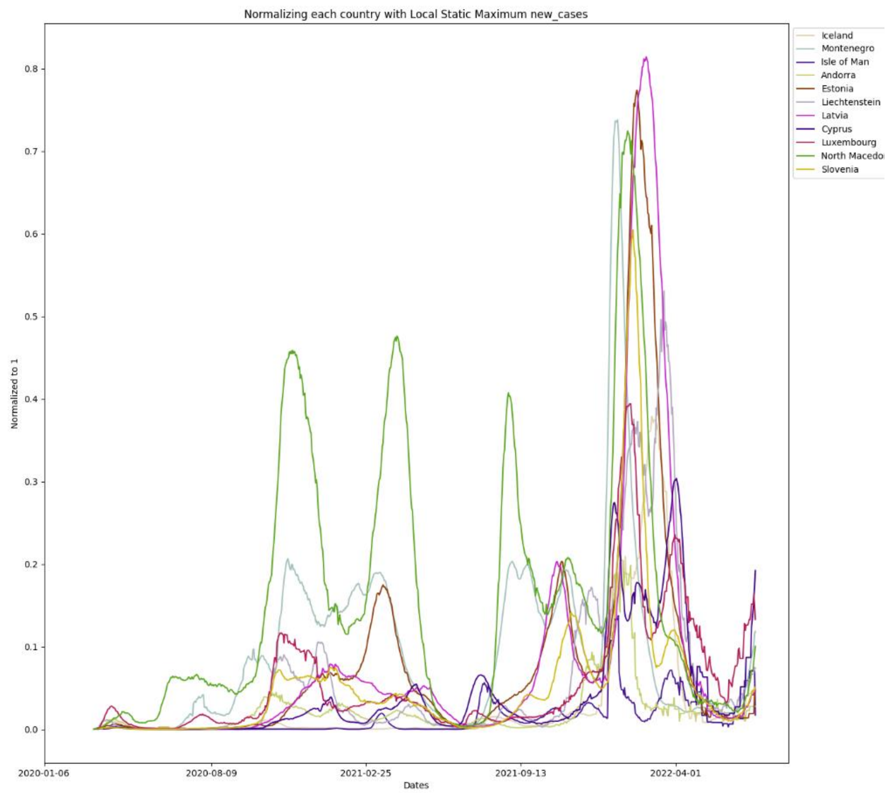
  a. Segregation by Population of European Countries:
  https://plakshauniversity1-my.sharepoint.com/:f:/g/personal/abhigyan_mehrotra_plaksha_edu_in/Euh5VBOPx9pJsy6g3pqZY7cBbpDT5wTvZxrVlrvs9WJdfQ?e=Qe7ooM

  b. Segregating by Population Density of European Countries:
  https://plakshauniversity1-my.sharepoint.com/:f:/g/personal/abhigyan_mehrotra_plaksha_edu_in/EmC8C80GWCJBsVHfiuGkytMBLrwr79ySCPp6zsZQJ06qzA?e=vaWDbe

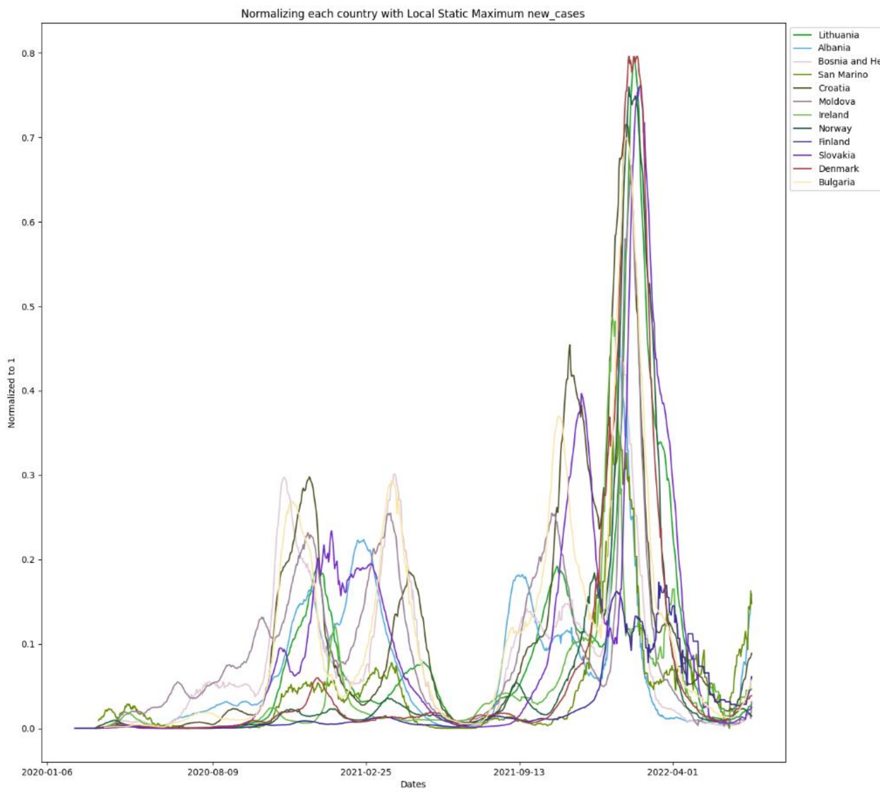
  c. Segregating by giving half weightage to population and half weightage to density:
  https://plakshauniversity1-my.sharepoint.com/:f:/g/personal/abhigyan_mehrotra_plaksha_edu_in/EjW4dy8Uc2hFgbf3BGjXFIMByYOszryH42vUXWvBzKci4w?e=7FWpfw

Pasting Plots for Local Static New Cases in the third type of segregation (weighted population and density)

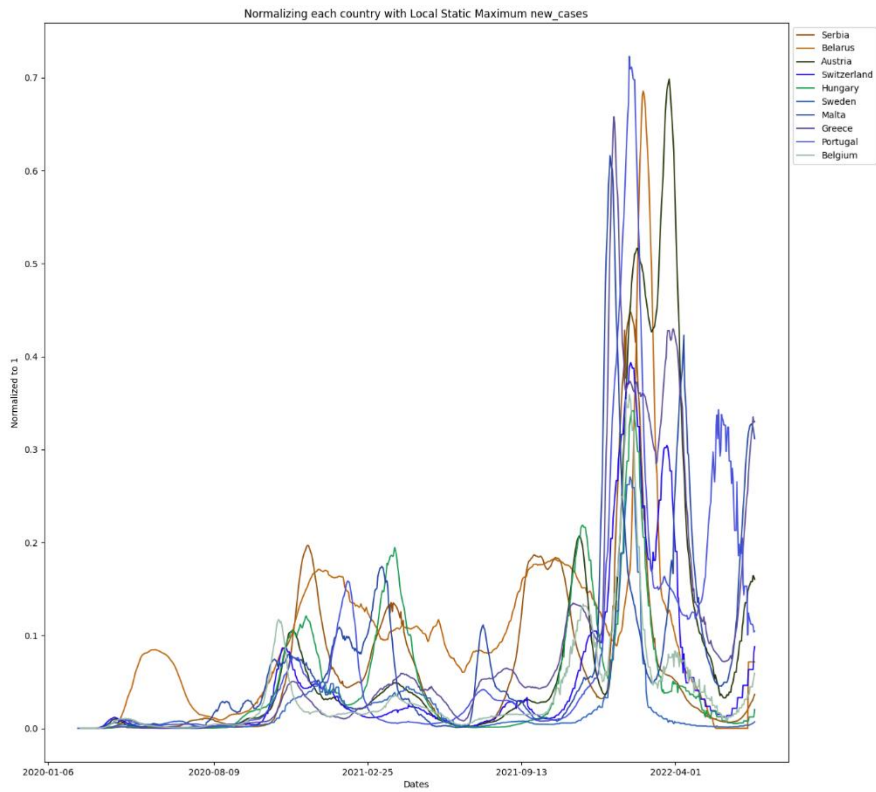
1. Group1's plot of new cases:



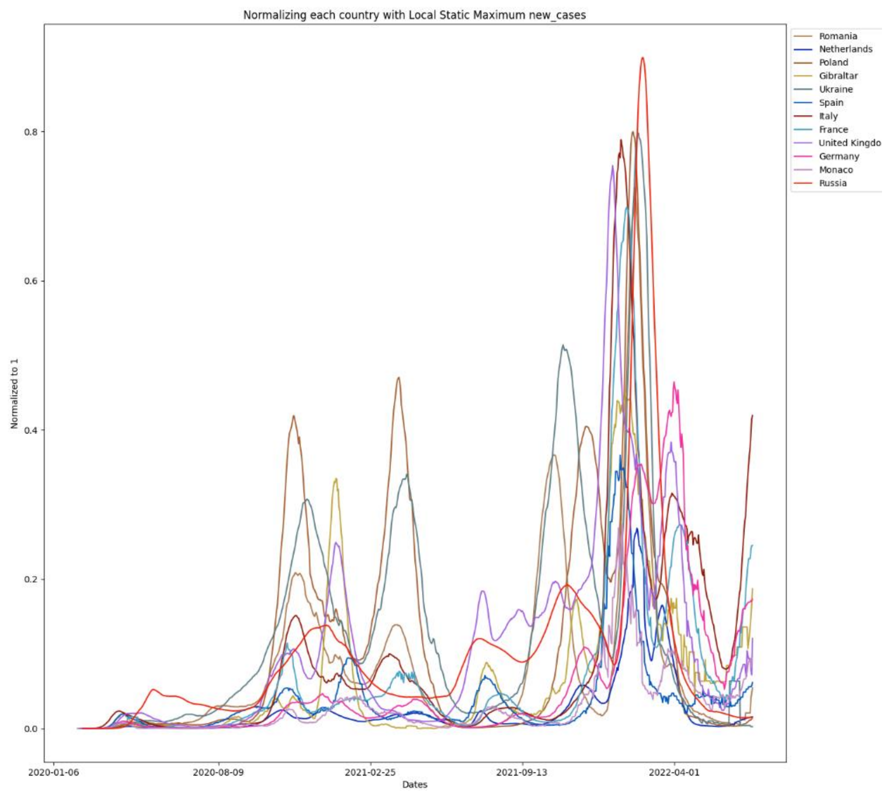
2. Group 2's plot of new cases



3. Group 3's plot of new cases



4. Group 4 ’s plot of new cases



- **Technique 5**: Local dynamic maximum: In this technique each country's data was normalized to a dynamic time frame maximums. The following steps were taken for each country;

    1. **Decide a timeframe:** The timeframe for each country was common. Usually, the time frame chosen was 120 days.

    2. **Obtain the maximum value for each date:** Suppose the time frame is 120. The time frame was shifted by one day and till the end of the duration. For each shift the time frame might have a different maximum. Each date could be said to be under multiple time frames (in this case 120 which would encompass a total of 240 days). Hence for each day there could be 120 maximums present. The maximum of all these 120 maximums was assigned as the maximum value for each day.

    3. **Divide each data point by the maximum obtained for the day:** In this step the data point for each data was normalized by its maximum obtained for its date.

The plots by the three segregation techniques are given below in the form of share points links:

a. Segregation by Population of European Countries:

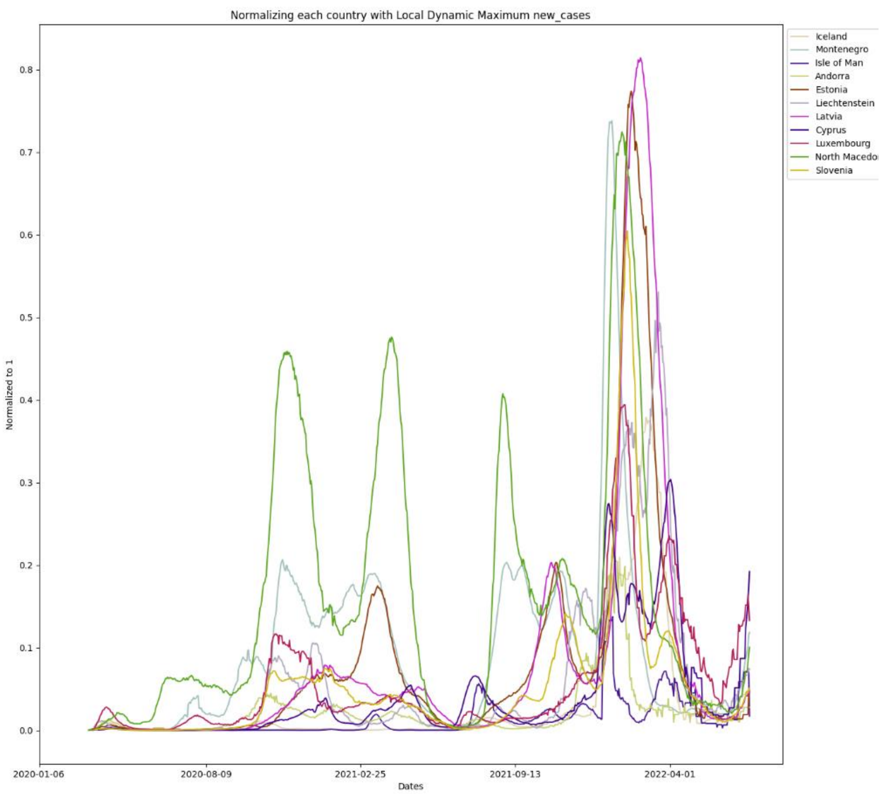
   https://plakshauniversity1-my.sharepoint.com/:f:/g/personal/abhigyan_mehrotra_plaksha_edu_in/EtYIfHnOXJBAsoYTuloXlpYBSH_4i5nW45qff0oNZ7AkBQ?e=41yhpQ

b. Segregating by Population Density of European Countries
c. Segregating by giving half weightage to population and half weightage to density:

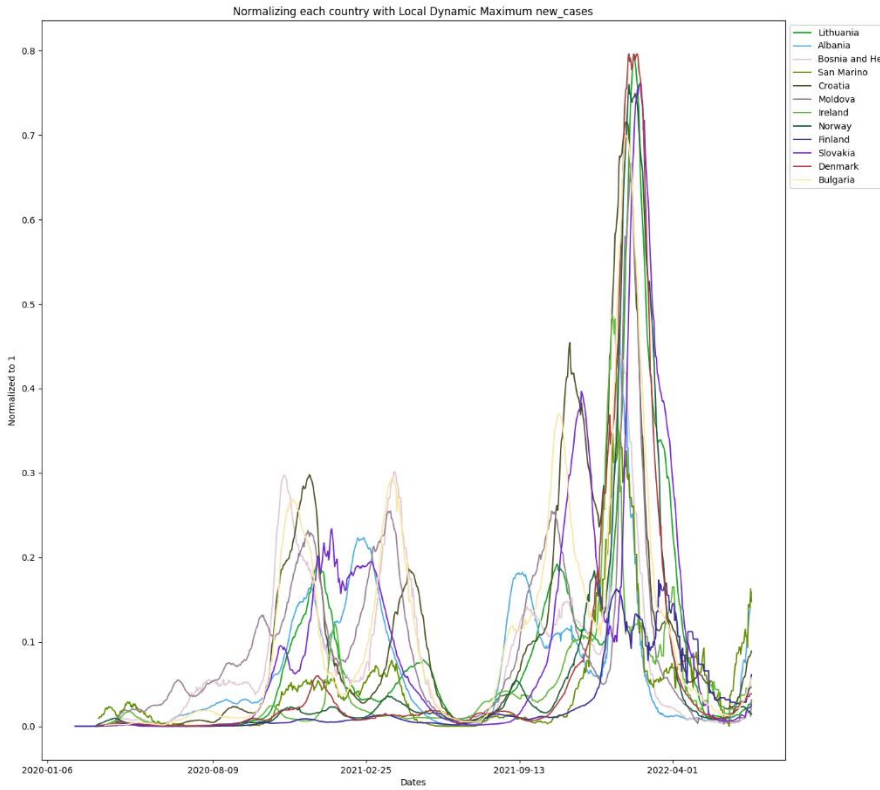
   https://plakshauniversity1-my.sharepoint.com/:f:/g/personal/abhigyan_mehrotra_plaksha_edu_in/Esb2VCJ0QPtIlNpTznXMHDMB19s3i_DctujxYCnDaYfpWQ?e=4G9irb

Pasting Plots for Local Dynamic Mean Maximum New Cases for the third type of segregation (Weighted Population and Density)

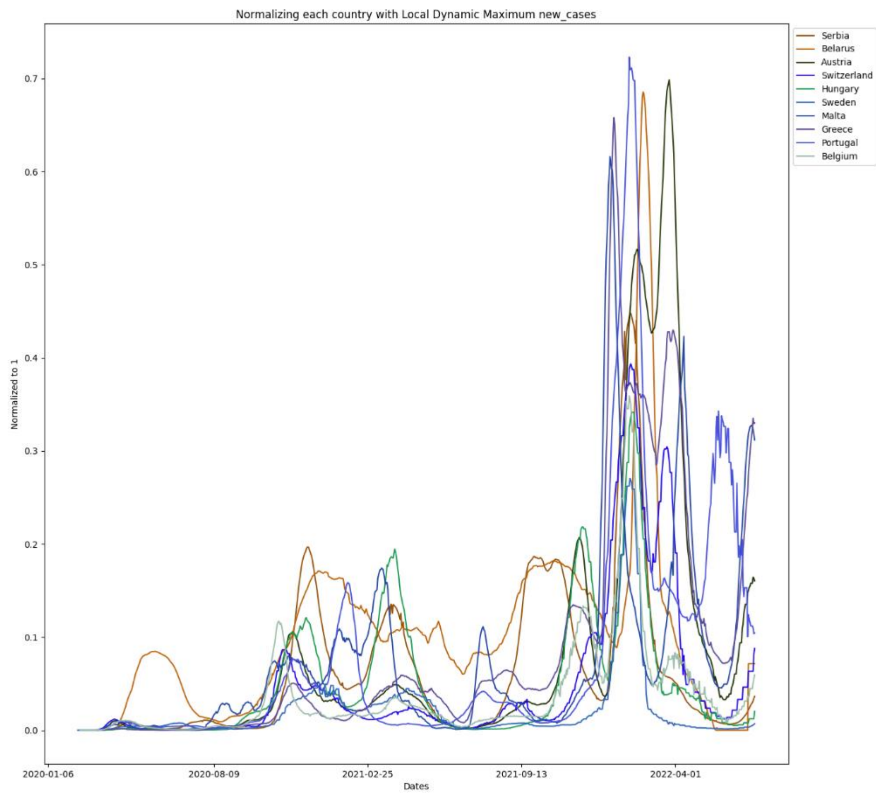
1. Group 1’s New Cases Plot:



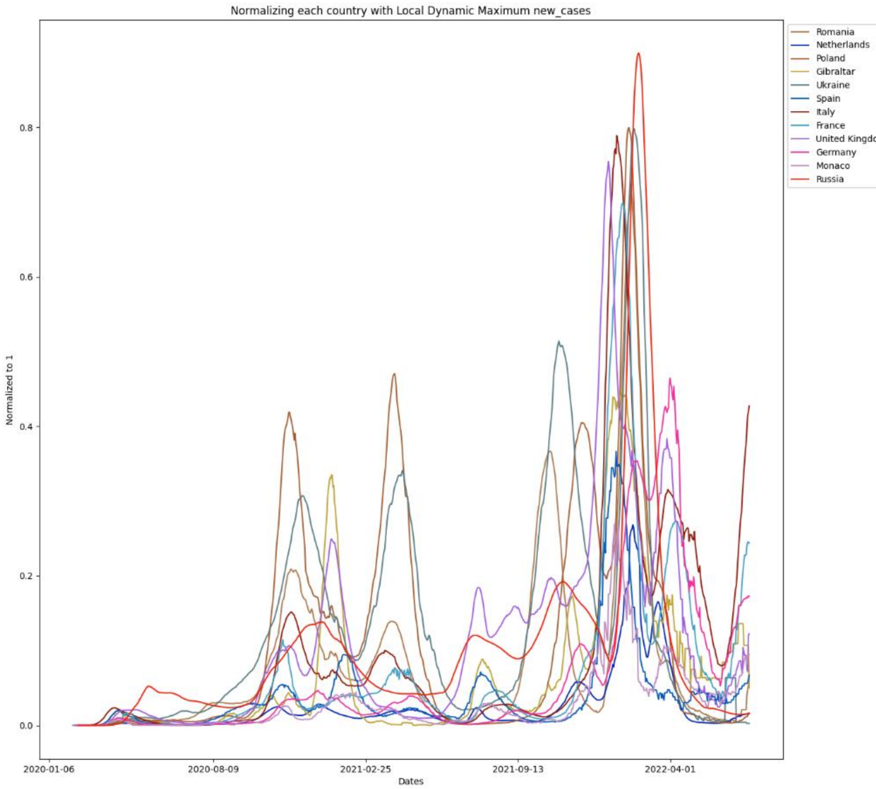
2. Group 2's New Cases Plot:



3. Group 3's New Cases Plot



4. Group 4's New Cases plot:



Normalizing each country with Local Dynamic Maximum new_cases
Normalized to 1
0.0
0.2
0.4
0.6
0.8
2020-01-06
2020-08-09
2021-02-25
2021-09-13
2022-04-01
Dates
Romania
Netherlands
Poland
Gibraltar
Ukraine
Spain
Italy
France
United Kingdo
Germany
Monaco
Russia

- **Technique 6**: Local dynamic mean maximum: In this technique each data point was normalized by multiple timeframe Local maximums and stored in an array, The mean of this array was taken, and the final value was assigned to the date. Currently we have taken the duration of one timeframe to be 120 days. Hence the data on a single date may be normalized 120 times, added and then averaged to give the final value. The following steps were taken.

    - **Decide the timeframe:** Before running the program a common dynamic timeframe such as 120 days is decided for all countries

    - **Obtain Maximum values for each shift in the time frame:** The dynamic timeframe was shifted by one date at a time. This shift gives a specific local maximum for each country.

    - **Obtain Normalized values:** After obtaining a maximum all the data points in the timeframe were normalized by the maximum and stored in an array.

    - **Averaging the Normalized values:** Since the shifts in the timeframe create normalized value arrays of overlapping dates, they can be added. After adding these values are averaged to obtain a final data Frame. This data Frame was then plotted.

The plots by the three segregating techniques are given below in the form of SharePoint links.

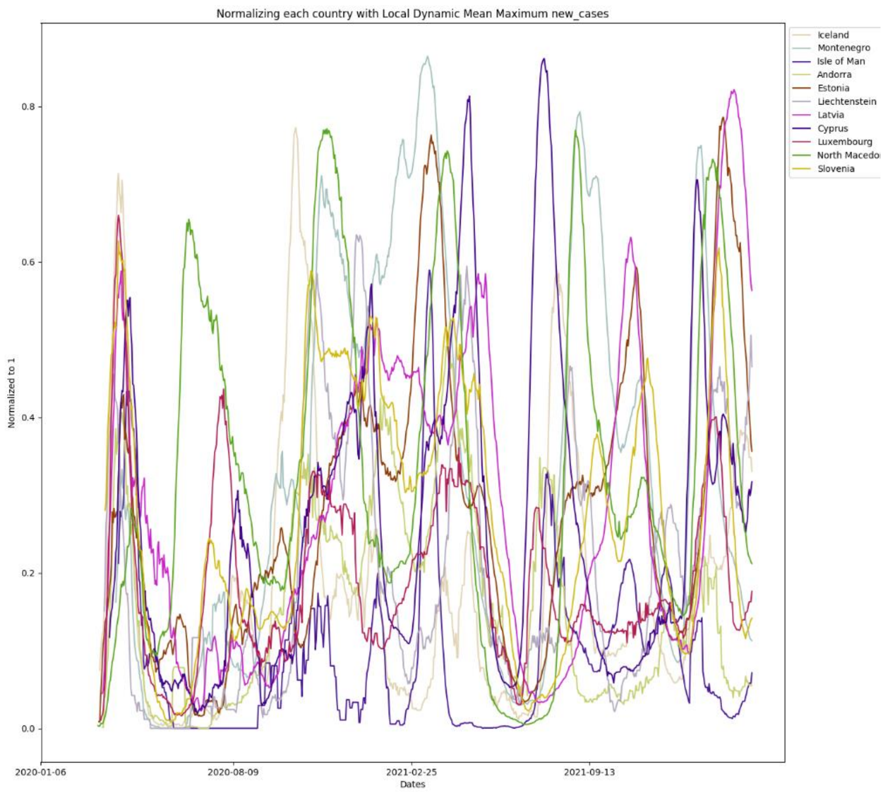
a. Segregation by Population of European Countries:
https://plakshauniversity1-my.sharepoint.com/:f:/g/personal/abhigyan_mehrotra_plaksha_edu_in/EuwH29vwfgNKqthr5uv_WAQBwPzh4BZsfBTzHTyMFWNEug?e=HoDK8T

b. Segregating by Population Density of European Countries:
https://plakshauniversity1-my.sharepoint.com/:f:/g/personal/abhigyan_mehrotra_plaksha_edu_in/EnHlLuMb8LxMrqZgDAR2YuIBT0Zq05-0JTeTkpoTV8xXfA?e=VqcY1s

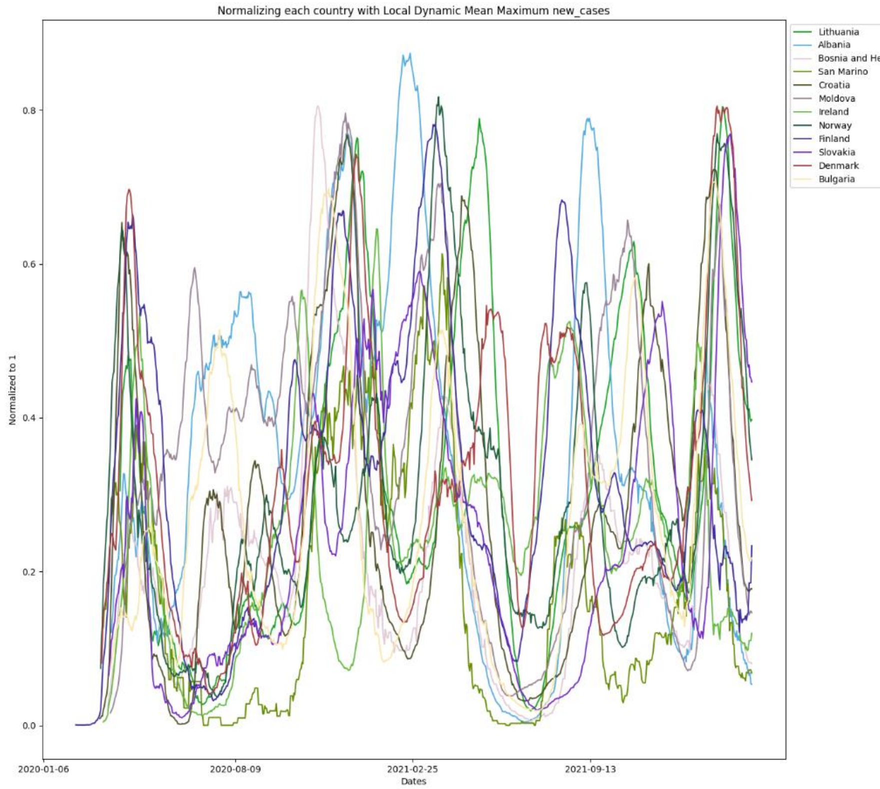
c. Segregating by giving half weightage to population and half weightage to density
https://plakshauniversity1-my.sharepoint.com/:f:/g/personal/abhigyan_mehrotra_plaksha_edu_in/EmIZZeACqTxEiRahhfuCTnwBjlGJYslktE1F_4kXpdZSig?e=IgYXVu

Pasting plots of Local Dynamic New Cases in the third type of segregation (weighted population and density)

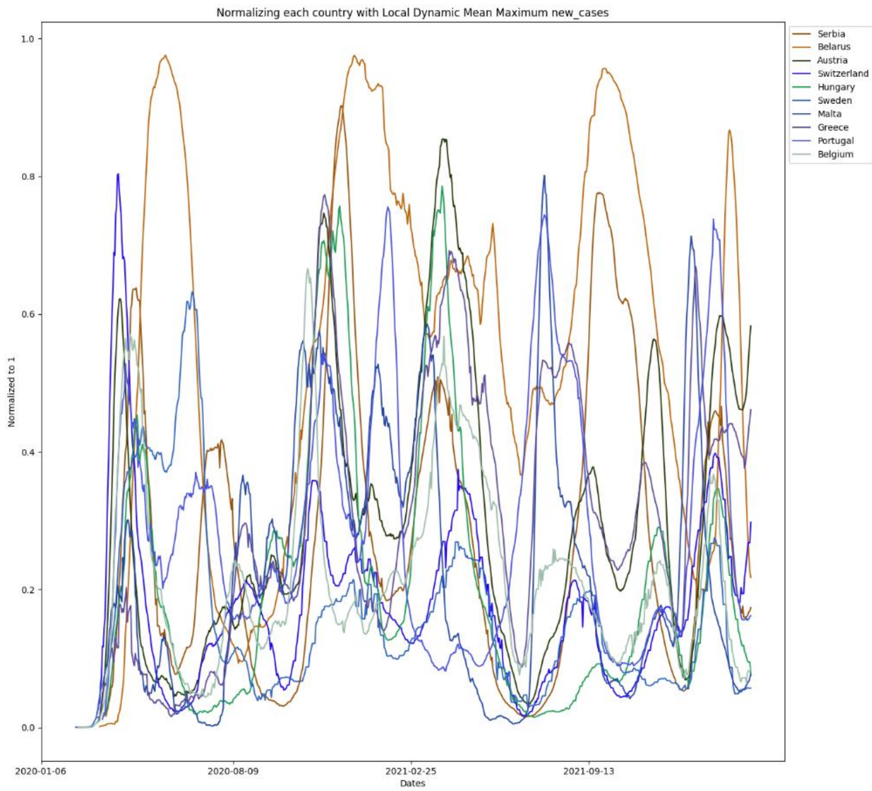
1. Group 1's plot of new cases



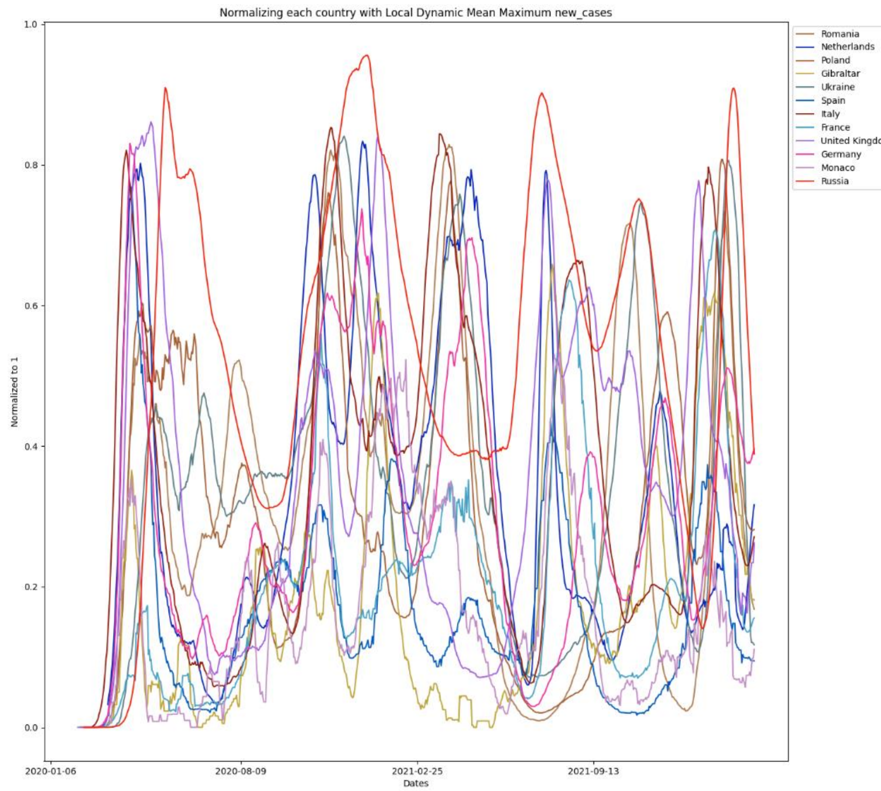
2. Group 2's plot of new cases



Normalizing each country with Local Dynamic Mean Maximum new_cases
Normalized to 1
Dates
0.0
0.2
0.4
0.6
0.8
2020-01-06
2020-08-09
2021-02-25
2021-09-13
Lithuania
Albania
Bosnia and He
San Marino
Croatia
Moldova
Ireland
Norway
Finland
Slovakia
Denmark
Bulgaria



3. Group 3’s plot of new cases



4. Group 4’s plot of new cases

Data Visualization:

For visualizing the various data frames obtained by the above techniques the python library Matlab was used. Additionally, each country was assigned a unique colour obtained by random hex values. For reference following are the hex values: along with the countries.

1. France: #44a2c7
2. Germany: #f42fa2
3. Finland: #3b2b99
4. Russia: #f41f09
5. United Kingdom: #a25ee1
6. Italy: #91190a
7. Spain: #0258bb
8. Sweden: #2f65bf
9. Slovenia: #d2ba0d
10. Denmark: #a83e40
11. Estonia: #8d3b08
12. Belgium: #9dbeaa
13. Greece: #5e4b98
14. Luxembourg: #b51c57
15. Norway: #1e5c3e
16. Switzerland: #2106f2
17. Albania: #53ace0
18. Austria: #223406
19. Croatia: #4f5026
20. Latvia: #c923cc
21. Romania: #ae7a50
22. North Macedonia: #59a61f
23. Serbia: #96510b
24. Netherlands: #0525ba
25. Belarus: #bc6309
26. Iceland: #e2d4b3
27. Monaco: #c28cc5
28. Ireland: #62b93e
29. San Marino: #668d02
30. Czechia: #bcaab2
31. Portugal: #4851de
32. Andorra: #c6d06f
33. Ukraine: #5a7c84
34. Hungary: #1d9d53
35. Liechtenstein: #b3a9c0
36. Faeroe Islands: #d2eef6
37. Poland: #a55d32
38. Gibraltar: #bfa438
39. Bosnia and Herzegovina: #e3cad7
40. Malta: #2346a6

41. Slovakia: #732bc8
42. Vatican: #e52e53
43. Moldova: #998396
44. Cyprus: #400089
45. Bulgaria: #fbe7a8
46. Kosovo: #f2c023
47. Montenegro: #a2c1bd
48. Lithuania: #18a424
49. Isle of Man: #4a1793
50. Guernsey: #171d71
51. Jersey: #256586

Appendix:
Legends

Reference:

1. https://github.com/owid/covid-19-data
2. https://www.georgiaruralhealth.org/blog/what-is-a-moving-average-and-why-is-it-useful/