Group Project: POIMAGIC

an Early Warning Systems for Streaming Spatial Events

Jeffrey Levan
University of houston
Houston Tx U.S
jeffreylevan88@gmail.com

Nghia Luu
University of houston
Houston Tx U.S
nghialuu1475369@gmail.com

William Le
University of houston
Houston Tx U.S
lewilliam27@gmail.com

Abstract

In this article, we simulate an Early Warning System called POIMAGIC for streaming spatial events based on Earthquake hotspots and then to study if the developed system can be reused to detect Earthquakes in a different region, with different amounts of available data. The original idea of this project is to implement the use of the built-in Kernel Density Estimation (KDE) function to train a model that is able to detect earthquakes and mark them onto the real world map using major events that happened in the past. The goal is to facsimile the KDE model by our own “grid” method, which creates a 99x99 array where every element of the grid represents the number of earthquakes that occurred in a specific location (indicated by its longitude and latitude) mapped to the real world map.

Furthermore, we come up with two reasonable thresholds (in percentage) for the purpose of determining any “high” hotspots and “medium” hotspots within the given dataset. This, in fact, becomes a tough and time-consuming process due to the fact that different locations on the world map have different total number of earthquakes; hence, the thresholds used for this region definitely cannot be applied to other regions. In addition, we consider any location a hotspot if and only if they remain above the thresholds for a number of consecutive timeframes (we discuss more about this in the “methodology” portion). Eventually, this application turns out to not only be used for detecting any major earthquakes, but we also hope to use it for predicting those earthquakes that could possibly occur in the far future.

1. Introduction

We made use of two datasets, US and EU earthquakes. Each dataset is split into a number of batches where each batch contains two consecutive months worth of data. We will analyze these datasets using a sliding window of a defined size, three for the US dataset and (BLANK) for the EU dataset. Each dataset contains the time the earthquake was recorded, the coordinates (longitude and latitude), the depth of the earthquake, and the magnitude. Using this data, we are able to compile it into our grid system which would first compile a grid of the total number of occurrences at each coordinate. We will then map each coordinate to a cell in our 99x99 array. Once we have this grid we will then transform this grid using two user defined density thresholds, one for defining small high density areas and the other for defining large medium densities and plot the results over time in a series of pictures and a gif. The gif will allow us to visualize the change of hotspots over the whole dataset.
After creating this “grid” system, we are left with three user defined variables used for testing, the two threshold percentages and the sliding window size. The threshold percentages will vary depending on what data is being used due to different areas of the world having varying occurrences of earthquakes. The sliding window is how many batches our system will check for that same density. Only if the coordinate experiences the same density over X batches will that spot be considered a hotspot. This will help prevent outliers and helps ensure the accuracy of our plots.

2. Datasets

The dataset given was extracted from https://earthquake.usgs.gov/earthquakes/feed/v1.0/csv.php. In the original dataset, location was given by country, therefore the filtered dataset must have used latitude and longitude in order to extract EU. For the US, as it is a country in the original dataset US could be found in location. There are more features in the original dataset but for the purpose of this task only 5 which were time, latitude, longitude, depth and magnitude. Time is the date at which the earthquake occurred. Latitude and longitude are the coordinates to the location of the earthquake. Depth is how far down the event occurred in km. Mag is the magnitude of the earthquake which is a scale to know the strength of an earthquake.

For the US, we were given 10 contiguous batches from the end of 2020 to 2022 (3 files per batch). Each file had 2 months in them, giving us 24 months to work with.

For the EU, we were given 36 files total from the start of 2016 to the end of 2018. Each file contains 1 month giving us 36 months of data to work with.

3. Methodologies

a. **Kernel Density Estimation (KDE)**
Kernel Density Estimation (KDE) plot is used for visualizing the probability density of univariate or multiple variables together. It depicts the probability density function of the continuous or non-parametric data variables. To train the density model, we simply call the built-in function *KernelDensity()* in the *sklearn.neighbors* package and pass in the proper *bandwidth*, *metric, kernel,* and *algorithm.* Once we get the trained model, we fit it on the *xy* grid, which contains all the latitudes and longitude coordinates, by computing the log-likelihood of each sample under the model using *score_samples()* function and taking the exponential of that.

```
# run kernel density estimation

kde25 = sklearn.neighbors.KernelDensity(
    bandwidth=5,
    metric='minkowski',
    kernel='gaussian',
    algorithm='ball_tree'
)
kde25.fit(df_large.values)

# fit the trained model on the xy grid

log_density = kde25.score_samples(xy25)
density = numpy.exp(log_density)
density = density.reshape(x25.shape)
print("Shape of Density Values:\n{}\n".format(density.shape))
```

```
Shape of Density Values:
(1717, 1717)
```

The *density* value obtained is then used for plotting the contour map along with parameters such as list of longitudes, latitudes, and levels (see code below).

```
levels = numpy.linspace(0, density.max(), 25)
plt.contourf(x25, y25, density, levels=levels, cmap=plt.cm.Reds)
plt.show()
```

We obtained the plot of Density Estimation for locations of earthquakes within the US (see figure below). Note that in this example, we only care about those earthquakes with their magnitude above 2.5. We find an interesting fact that the highest frequent hotspots occur at places near Alaska and Hawaii (corresponding to North West and South West of the US).

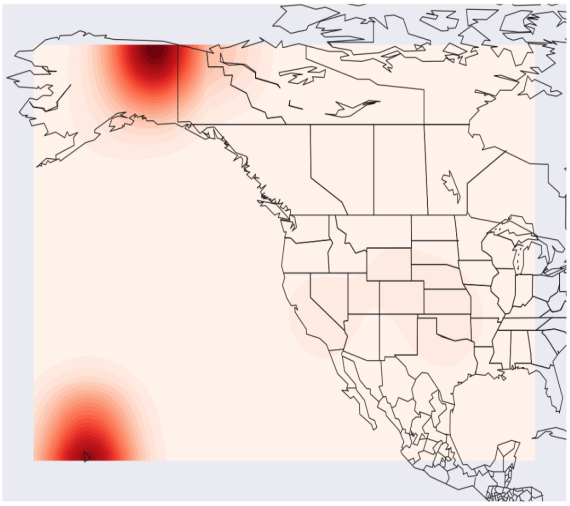

Figure 1. Density Estimation for locations of earthquakes within the US with magnitude >= 2.5

b. "Grid" method

In this article, we introduce the "grid" method as a new approach to simulate the built-in KDE function. The idea of this method is to:

- Generate a **99** by **99** grid across the map of a given region, each grid cell contains the number of earthquakes from corresponding locations.
- Map each grid cell to a range of the corresponding **latitudes** and **longitudes** on the real map.
- Merge all neighboring grid cells that are above the **thresholds**.
- Consider each grid cell a **"hotspot"** if and only if it remains above the threshold for ***x*** consecutive files (one batch).

Notice that the dimension of the grid we use does not necessarily have to be 99 by 99. This is just a *well-chosen* number that we use for our grid. Furthermore, the dimension of the grid we choose plays an important role towards our final result. This is because of the fact that each grid cell is mapped to a range of latitudes and longitudes. For example, if we think of using the grid of dimension 1 by 1, then that single grid cell corresponds to our whole regional map. As a result, every single coordinate *(lon, lat)* is mapped to that corresponding grid cell and hence, it would contain the total number of earthquakes at all times, which would not make any sense. Vice versa, if the dimension of the grid is too large, then it would take a very long time to run through our algorithm and the result we obtain would not be optimal.

To map coordinates (lon, lat) to its corresponding grid cell, we simply take the integer part of the numbers (round it down). Also note that for the longitude coordinate, since the minimum value is negative, we need to subtract that negative value before rounding it down. We do this because the grid cell's indices do not take negative values (0..98). For example, a location that has coordinates (-124.25, 35.27) with min_lon = -124 will be converted to grid index (0, 35). Since the min_lat value is not negative, we do not need to do so. The function *get_grid()* below will help us handle this task:

```
def get_grid (csv_pr, shape, min_lon):
    grid_lon = numpy.zeros(99)
    grid_lat = numpy.zeros(99)
    grid = numpy.zeros(99)

    xlist = []
    ylist = []
    zlist = numpy.zeros(shape)

    for i in range (shape-1):
        tmp = numpy.zeros(shape)
        zlist = numpy.vstack([zlist, tmp])

    for i in range (98):
        tmp = numpy.zeros(99)
        grid = numpy.vstack([grid,tmp])

    for files in csv_pr:
        df_tmp = pd.read_csv(files)
        lon_list = df_tmp['longitude']
        lat_list = df_tmp['latitude']
        xlist = xlist + list(lon_list)
        ylist = ylist + list(lat_list)
        for j in range (len(lon_list)):
            lon = lon_list[j]
            lat = lat_list[j]
            grid_lon[math.floor(lon-min_lon)] += 1
            grid_lat[math.floor(lat)] += 1
            grid[math.floor(lon-min_lon)][math.floor(lat)] += 1
    return [grid_lon, grid_lat, grid, xlist, ylist, zlist]
```

```
grid_lon, grid_lat, grid, xlist, ylist, zlist = get_grid(csv_pr, 5762, min_lon)
```

Note that the function also returns *grid_lon, grid_lat, xlist, ylist,* and *zlist.* The values of *xlist, ylist,* and *zlist* will be used for plotting a contour map later. On the other hand, *grid_lon* contains the total number of earthquakes at each range of longitudes. And similarly, *grid_lat* contains the total number of earthquakes at each range of latitudes. For example, *grid_lon[0]* contains the total number of earthquakes for every point that has its longitude between [-24, -23], and *grid_lon[1]* is the total number of earthquakes for every point that has its longitude between [-23, -22]. If we plot these two arrays for the US dataset, we obtain the graph below.

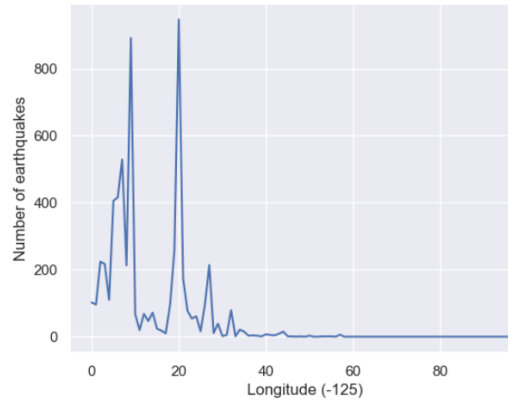

Figure 2. Total number of earthquakes per longitude range

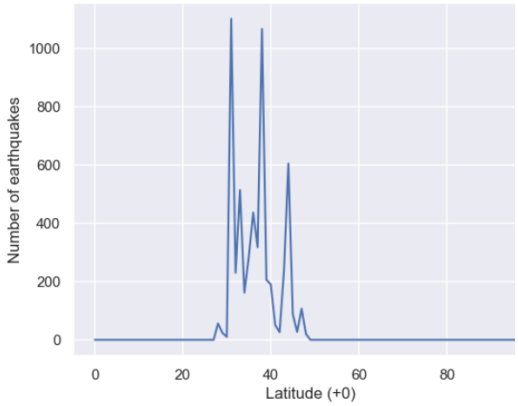

Figure 3. Total number of earthquakes per latitude range

Plotting *grid* using *seaborn.jointplot()* gives us:

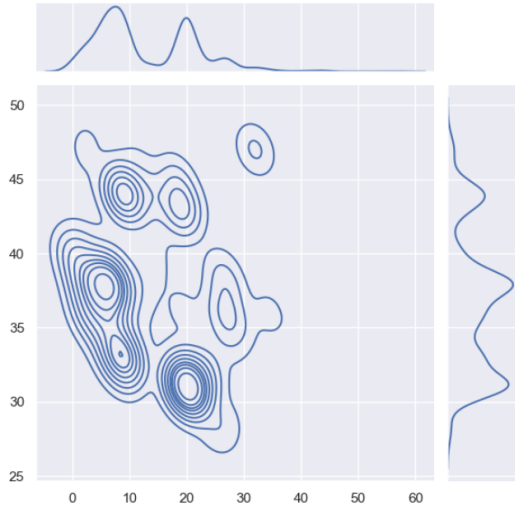

Figure 4. Density Estimation graph using seaborn.jointplot()

c. High and medium thresholds

Two values that will determine the threshold for the classification of the two types of hotspots. This value is given as a percentage of the total number of earthquakes per batch and will change per batch. This will be a user defined value as each dataset will have a different set of requirements. This will be used in the transformation from the grid to the merged grid. To calculate the threshold, we find the maximum number of earthquakes that occurred in the batch and multiply that by the percentage that the user set.

```
m = 0
for i in range(1, len(grids)-1):
    for j in range(1, len(grids[0])-1):
        if (grids[i][j] > m):
            m = grids[i][j]

threshold1 = math.floor(m/100*d1)
threshold2 = math.floor(m/100*d2)
print(m,threshold1,threshold2)
```

Figure 5. Image of raw code that depicts how the threshold is calculated

Where d1 is the small hot spot and d2 is the large hotspot.

This was one of the pain points of our experiment as the threshold values are very important as they directly affect our results. If you choose values that are too low almost anything could be considered as a hotspot making the data null. We encountered this while using our EU dataset.

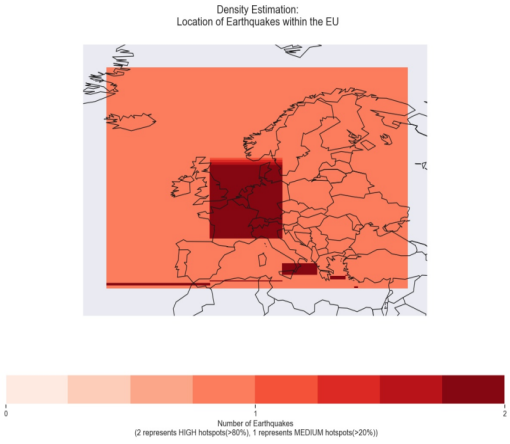

Figure 6. Depiction of the European Plot with threshold values 80% and 20%

From this image we can see that the upper threshold is set to 80% and the lower threshold is 20%. We can see from the results that the lower threshold was set far too low for our dataset, almost the entirety of the plot is considered the low density hotspot. By raising the value of the lower threshold density, we are left with the following image.

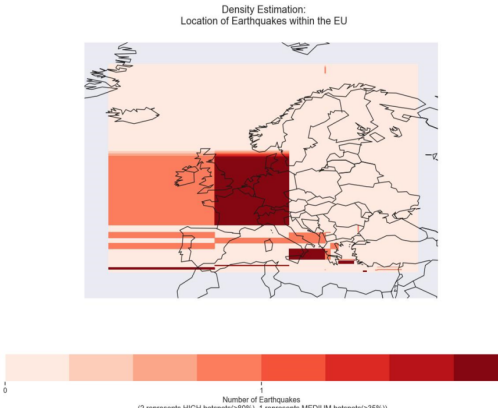

Figure 7. Depiction of the European plot with threshold values of 80% and 35%

From this image, we can distinctly see the various hotspots and the regions which they encompass. If the values set are too large, we will not get enough hotspots to be able to interpret the data properly. This is shown below:

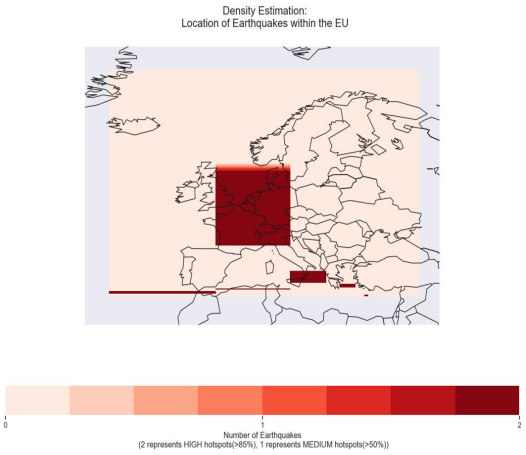

Figure 8. Depiction of the European plot with threshold values of 80% and 60%

We only see the dense hotspot regions and have no information on the less dense hotspot as that threshold is set to high.

For the US we opted with 30% for the upper threshold and 5% for the lower threshold as we can distinctly see the difference between hotspots.

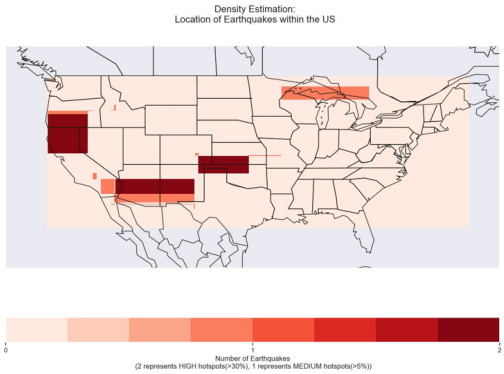

Figure 9. Depiction of the USA plot with threshold values of 30% and 5%

In comparison, switching the threshold b 50% for the upper bound and 20% for lower, does not give us a lot of information except that there are 2 hotspots with above 50%. It therefore makes sense why the 30% and 5% threshold were used, for a clearer map.

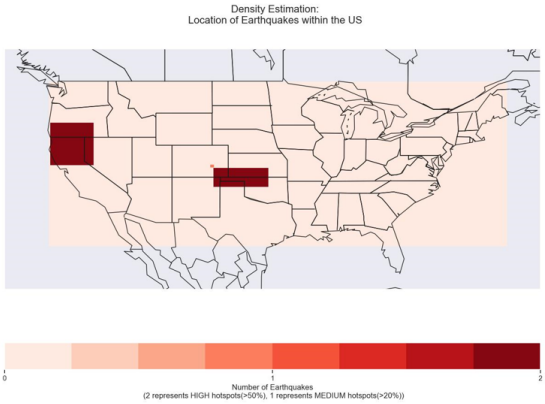

Figure 10. Depiction of the USA plot with threshold values of 50% and 20%

d. Merged grid cells

The merged grid cells are created by our function *get_all_merged()* which will transform the grids that have the number of occurrences into a grid that classifies that density into a hotspot type. This end result is what will be used to determine the final plot and where the “real” hotspots are.

Below is an excerpt of some rows in the original grid. Each different value corresponds to how many earthquakes occurred at that coordinate.

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 7
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0
 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 2. 0. 0. 2. 0. 0. 0. 1. 0. 0. 0. 1. 1
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 4
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0
 0. 0. 0.]
```

Figure 11. Image of the "grid" the system makes counting all occurrences of earthquakes in a particular area

Once we run the *get_all_merged()* function we return another array with the hotspot values, pictured below

```
[0. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.
 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.
 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.
 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.
 2. 2. 2.]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0.]
```

Figure 12. Image of the "grid" after a transformation of the occurrences into the two density classifications

Our returned array contains k number of 99x99 arrays, where k is the number of batches that are in our dataset. Each cell in these 99x99 arrays will correspond to its respective coordinate and will have a value of 0, 1, or 2 which will determine the density and what type of hotspot that region is. This set will then be fed into the final algorithm for our system, determining the "real" hotspots according to the sliding window

e. Hotspots determination

To determine the hotspots in our final plot, we need to have that region maintain that same density for multiple batches, as many as the user defined. If the user defines the sliding window as 3, that region will need to have that same density maintained over 3 batches (or 6 months) to be actually considered a hotspot in the final plot.

The sliding window is the other user-controlled variable in our system. This is another important variable because if this number is too low, there would be too many hotspots that are really just outliers. They do not recur enough to be considered a real hotspot. Below is a plot with the sliding window set to 2.

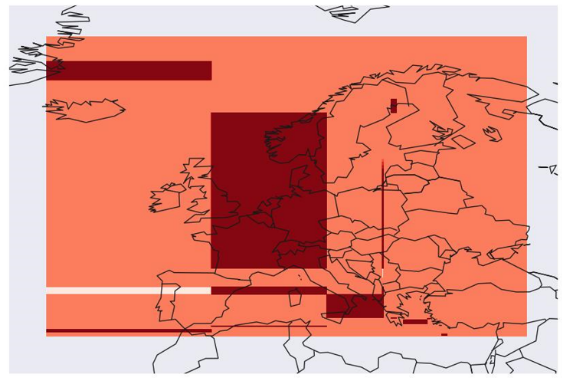

Figure 13. Depiction of the European plot with the sliding window value set to 2

From this image, we can see similar results as if the threshold was set too small. There are too many hotspots and the data is not legible. By allowing the system to ensure that the density is mainly consistent over time, we can get the plot of the "real" hotspots that we know earthquakes are concentrated around.

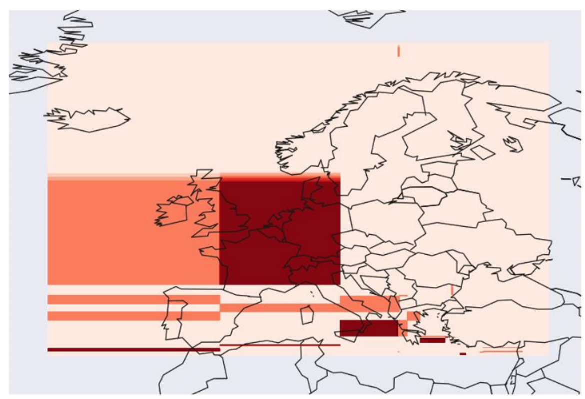

Figure 14. Depiction of the European plot with the sliding window value set to 3

This is the plot with the sliding window set to 3. As we can see the majority of the hotspot regions are now removed and we are left with the ones that remained consistent over 3 batches. This improves the data readability and allows for better interpretation of the data.

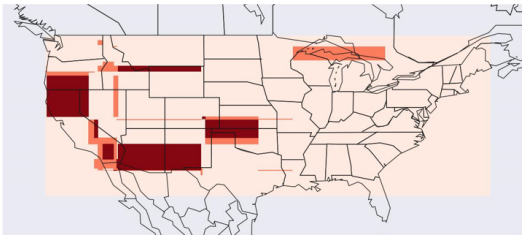

Figure 15. Depiction of the USA plot with the sliding window value set to 1

For the US, when using a single batch we get more hotspots, and the presence of high hotspots are more common. The reasoning of having a bigger window is so that one month does not dictate the location being a hotspot. Compared to the window of size 3 shown below, between Montana and Wyoming we can see that it does not appear. Meaning that the hotspot only appeared for one month. Therefore it cannot be called a hotspot.

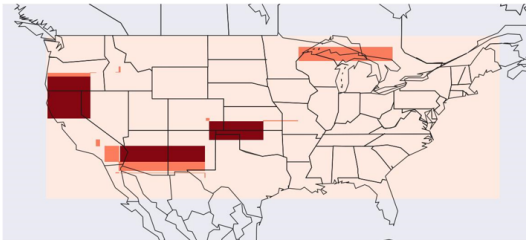

Figure 16. Depiction of the USA plot with the sliding window value set to 3

4. Analysis of result

Our grid system will analyze the data sets then returns a series of images that represent the change in hotspots over time and the corresponding gif animation. This method of visualization will allow us to better see where these hotspots are appearing and how they react to the new data that is coming in. Each image that the system generated will correspond to one batch of the dataset. In our US dataset we have twelve files with a window size of three. This will correspond to a total number of batches and images of ten.

Figure 17 represents the first image in our US dataset. From the image, we can see that we have a couple regional hotspots along the north border of California and another one stretched across southern California, New Mexico, and Arizona. We can also see two dense hotspots, one across the top of Texas and Oklahoma, and another near the border of California and Nevada. In figure 18, we see the change that happened in the next batch. The regional density located across New Mexico and Arizona has converted from a regional hotspot to a dense hotspot. We also experience many of the old hotspots expanding their region and the emergence of a new regional hotspot along the Great Lakes region.

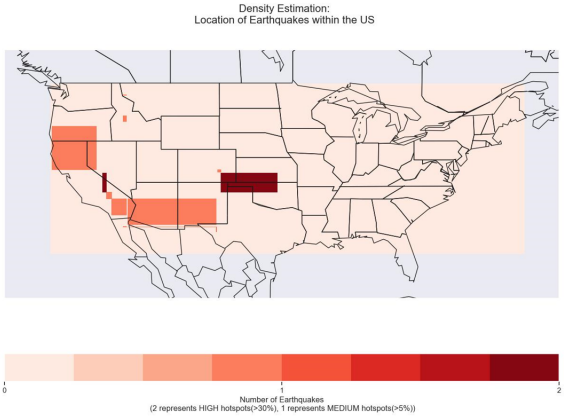

Figure 17. First image depicting the US plot with threshold values of 30% and 5% and the sliding window of 3

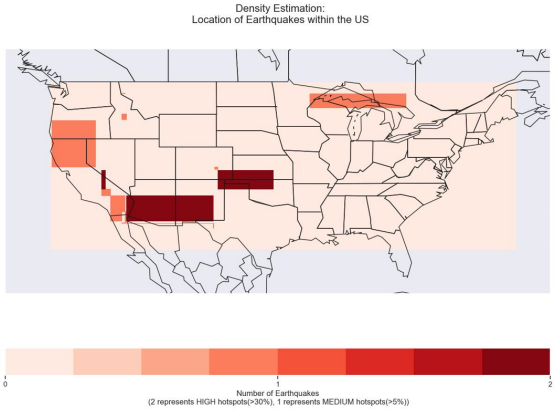

Figure 18. Second batch image of the US plot with threshold values of 30% and 5% and a sliding window value of 3

As we move further down the series of images, we can see most of the changes in hotspots occur near California. In figure 19, we see the depiction of the same plot but it is the seventh image in the series. From this figure, we can see that many of the hotspots are centered around

California with the majority of its area being considered a hotspot.

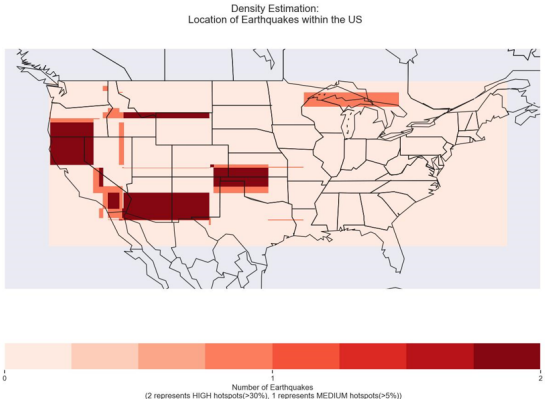

Figure 19. Seventh image of the series of US plots with threshold values of 30% and 5% and a sliding window value of 3.

From these results we can interpret that in the US, California is the most earthquake prone in the region. This can be backed up from the figure below. California lays along the Circum-Pacific Belt, which is a path along the Pacific Ocean that is often characterized by a large amount of active volcanoes and frequent earthquakes.

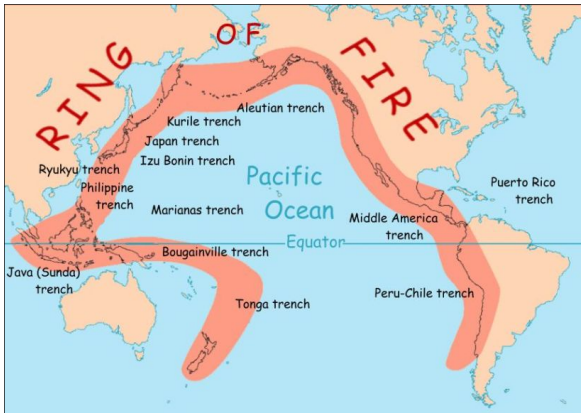

Figure 20. Image depicting the path of the Circum-Pacific Belt (the Ring of Fire) across the Pacific Ocean

For the EU in the first 7 images(figure 21), we can see that there is some middle density and high density at the mediterranean sea, the image then stays the same until image 20. In figure 22, we can see that France, Germany, Belgium, UK, Holland, Switzerland and Italy are in the high density of earthquakes. Going to the last image, we can see that the only change happens to be Italy changing from a medium hotspot to a high hotspot. We can see that being close to the mediterranean sea seems to affect earthquake, as the eastern of eu does not seems to be affected by earthquake.

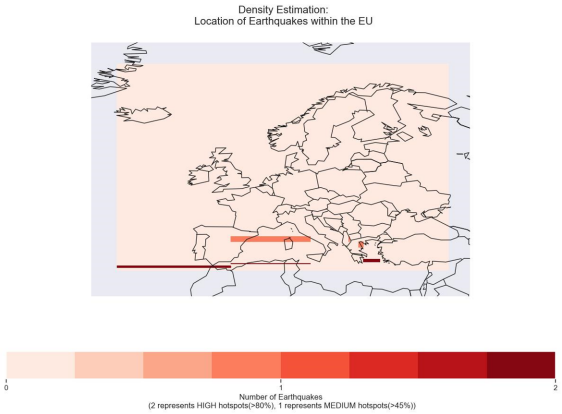

Figure 21. seventh image depicting the EU plot with threshold values of 80% and 45% and the sliding window of 3

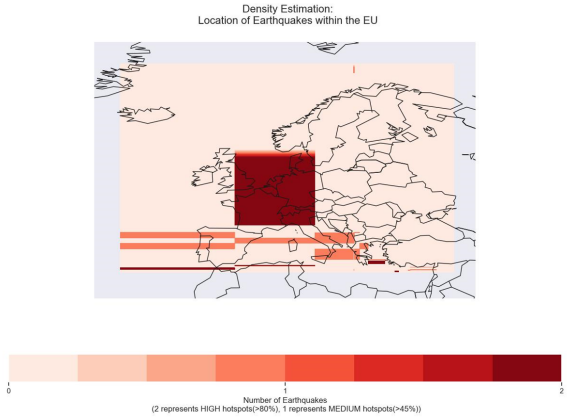

Figure 22. 22nd image depicting the EU plot with threshold values of 80% and 45% and the sliding window of 3

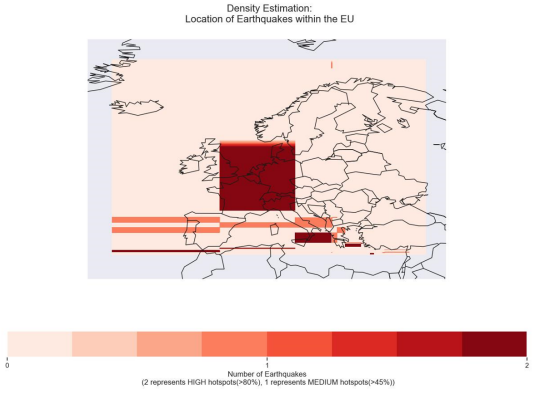

Figure 23. 3rd image depicting the EU plot with threshold values of 80% and 45% and the sliding window of 3

5. Comparison between Kernel Density Estimation and Grid method

For the map of the US, it seems that the grid method was able to capture correctly the information of the earthquake. The difference when looking both ways, Kernel Density Estimation is more pleasant and gives more precise information than the grid method. On the other hand, the Grid method allowed us to have a better control in knowing what information was put in.

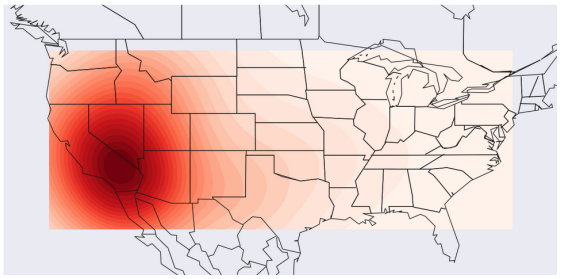

Figure 24. Density Estimation for locations of earthquakes within the US with magnitude >= 2.5 09/2020 - 10/2020

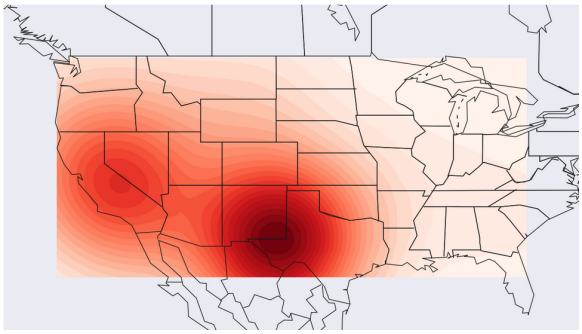

Figure 25. Density Estimation for locations of earthquakes within the US with magnitude >= 2.5 05/2022 - 06/2022

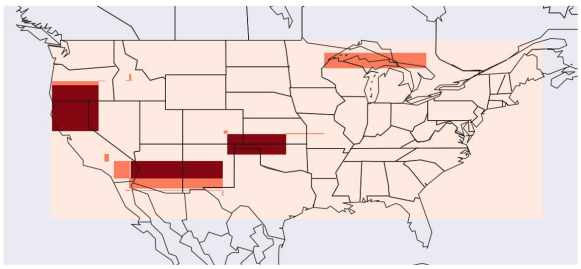

Figure 26. Depiction of the USA plot with the sliding window value set to 3

For the map of the EU, our grid method seems to have an issue (figure 28) compared to figure 27 which has a lot of earthquakes in Greece instead of France. What can be said is that the centering of our grid seems to be off.

Figure 27. Location of all earthquake in the EU between 10/2018 - 12/2018

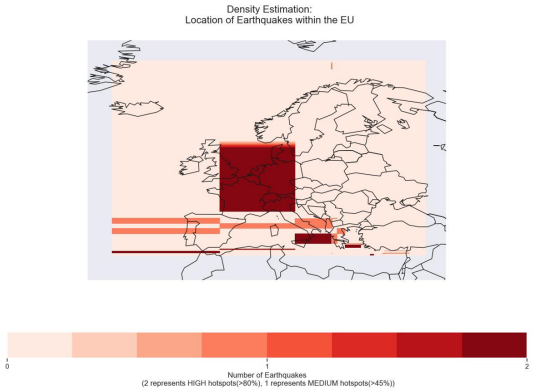

Figure 28. Image depicting the EU plot with threshold values of 80% and 45% and the sliding window of 3

However, this can be further improved by playing with those important parameters such as the grid's dimensions, number of sliding windows, and thresholds as we mentioned.

6. Conclusion

It turned out that figuring out the best parameters in order to obtain the best result we want becomes one of the biggest challenges overall in this project. The results we obtained might not be the best, however, this task really helped us understand how to do spatial data analysis related to density estimation and hotspot discovery. This task also gave us a good opportunity to deal with large data files as it is one of the daily jobs of data scientists.

To sum up, we observe that it is not very hard to reuse our system for the Europe region for the period of 2016 ~ 2018. All we need is the dataset recorded for Europe during this period, and the range of longitudes and latitudes of the region. However, since the US region doesn't have the same total number of earthquakes as Europe, the scale of thresholds for these two regions are completely different.

We created two animations of the images depicting hotspots for different thresholds over the two datasets US and EU below

Figure 29. Hotspots depiction over the US region

Figure 29. Hotspots depiction over the EU region

7. Team Contribution

- Nghia Luu - Python codes, Abstract, methodologies (a,b), Conclusion
- Jeffrey Levan - Introduction, Methodologies (c,d,e), Analysis (US)
- William Le - dataset, Methodologies (c,e) (For US), Analysis (EU), Comparison Between Kernel Density Estimation and Grid method