Earthquake Spatio-Temporal Relationship Network

CPSC 572: Fundamentals of Social Network Analysis and Data Mining

Rafael Flores Souza	30128094
Laura Timm	30064647
Zain Sheikh	30056015

1. Abstract

Earthquakes are one of many natural disasters that negatively affect human civilization, therefore we decided to analyze the occurrence of earthquake behavior in specific regions with high seismic activity, formulating an analysis which could be of use to city planners and governments in minimizing human and material losses. For the analysis, we created a directed network in which the latitude/longitude location of earthquake occurrences are the nodes and the edges connect two nodes if they occurred within a maximum distance D kilometers and happened at most T days apart. After this set up, we narrowed down what we wanted to gather and learn from our network in the form of our three research questions:

1. Does a higher magnitude mainshock tend to suggest the occurrence of more foreshocks in the “region”?
2. What is the probability of an earthquake occurring in the same “region” as a past mainshock?
3. Is there a domino effect of foreshock magnitudes near each mainshock?.

Through various techniques, we concluded that an earthquake tends to influence the occurrence of other earthquakes based on magnitude and its influence power (e.g. how many of the earthquakes it influences also influence each other). We also found a way to measure the probability of an earthquake occurring in the same region as past earthquakes; the higher the magnitude of earthquakes, and the greater the quantity of earthquakes within a region, the greater the probability of future earthquakes occurring within that region. We explored our earthquake network to see if we were able to find a domino effect of magnitudes from an earthquake with a high origin magnitude for our data. Finally, we found that there was a downward trend in magnitudes as we moved away from earthquakes with higher magnitudes to its neighboring levels.

2. Research Question Approaches

Earlier we mentioned our three research questions and would like to expand on what approaches we used to answer them. For the first question, we will compare our network against an ensemble of directed random-geometric and directed random null models. Specifically, we will look at their number of edges and plot out-degree centrality & clustering coefficient values versus earthquake magnitudes (each in a separate plot) to come to a conclusion. Next, for the second question, we will calculate the probabilities of earthquakes causing subsequent future earthquakes, based on their magnitudes, giving us a formal approach to calculate the probability of earthquakes occurring within the same region as past mainshocks. Lastly, for the third question we used recursion to find the averages of each level of neighbors from an origin node that had a magnitude of 5.5 or higher. We then plotted our findings to create a visual which we used to determine if there was a decline at each level as we moved away from the origin node. After we plotted our findings we compared our network's results with an Erdos-Renyi Model and a Random Geometric Network to see if we were able to notice a difference in our model vs the Null Models. Through answering all these questions, we also explored the existence of a hub-spoke motif within our network, with large-magnitude hubs and small/medium magnitude spokes.

3. Introduction

Appropriately frame your project in the existing literature. This section must be properly referenced.

Natural disasters such as earthquakes have affected human societies since the beginning of time. Through careful observation and data-collection, scientists have been able to take these occurrences and identify patterns to help us better understand our world. Since earthquakes cannot be predicted this info is useful to help scientists generate models to get a better understanding of the next possible earthquake that could hit. Earthquakes alone come with a wealth of data, including magnitude, geographic location, and time. This data, coupled with the technology which allows us to create, visualize, and analyze the network of earthquakes within an area not only provides us with a rich topic to study and understand real-world networks, but it also allows us to explore analyze and answer important questions regarding the nature of earthquake occurrences, as well as their predictability.

With this motivation, we created a network which allows us to explore the data of individual earthquakes, as well as the relationships between earthquakes. Through the analysis which will be outlined in this paper, we were able to identify patterns which exist in earthquake occurrences, based on their magnitude as well as their relative location. We also provided the beginning blueprints for future work on the topic such as earthquake probability calculations. This research paper will surely provide insight into the not-so-random nature of natural disasters, as well as exemplify a meaningful approach to data-network analysis.

4. Dataset Description

The raw data was retrieved from [USGS's website] which provides earthquake occurrences on (within) a region and time frame as a csv file on the right. The csv consists of various columns of which the

time	latitude	longitude	depth	mag	magType	nst	gap
2022-09-20T01:10:41.495Z	31.7553	-104.6248	11.266	2.9	ml	30	73
2022-09-19T23:39:53.260Z	44.782	-110.8131667	8.72	2.72	ml	25	48
2022-09-19T22:32:11.528Z	53.123	-166.3597	78.937	2.9	ml	25	247
2022-09-19T21:25:58.281Z	52.9151	-161.0151	10	4.1	mb	11	248
2022-09-19T21:18:14.697Z	36.6608	-113.4843	10	4.4	mb	54	44
2022-09-19T20:04:01.054Z	58.8803	-153.9924	25.6	2.6	ml		
2022-09-19T20:03:47.380Z	59.7247	-152.7972	89.2	3.5	ml		
2022-09-19T19:30:39.858Z	18.4679	-103.8357	10	5.3	mb	80	166
2022-09-19T18:05:06.826Z	18.3667	-103.2524	15.134	7.6	mww	123	150
2022-09-19T10:29:30.883Z	14.2105	-92.0351	67.575	4.2	mb	38	180
2022-09-19T09:41:15.288Z	63.5593	-150.8365	2.5	3.2	ml		
2022-09-19T08:26:57.519Z	62.6162	-151.368	3.2	3.6	ml		
2022-09-19T04:47:18.910Z	36.7296667	-121.3453333	7.74	3.26	ml	78	53
2022-09-19T02:07:55.511Z	46.9429	-112.5071	11.686	2.6	ml	19	70
2022-09-19T01:37:05.384Z	64.7813	-149.0895	12.5	2.5	ml		
2022-09-19T01:15:30.424Z	64.7449	-149.1255	16.8	4.8	ml		
2022-09-18T22:40:09.710Z	19.9531666666667	-155.3875	35.58	2.65	ml	42	201

one we used were *occurrence time, latitude, longitude, magnitude, and id* of earthquakes. For our project we chose to use the dataset of North American

dmin	rms	net	id	updated	place
0.137	0.73	us	us7000i9ez	2022-09-20T01:26:49.040Z	52 km SSW of Whites City, New Mexico
0.02733	0.21	uu	uu60516617	2022-09-20T00:06:09.040Z	23 km SSW of Mammoth, Wyoming
0.726	0.44	us	us7000i9ea	2022-09-19T23:39:31.040Z	84 km S of Unalaska, Alaska
2.097	0.5	us	us7000i9dc	2022-09-19T22:37:40.738Z	253 km SSE of King Cove, Alaska
0.639	0.88	us	us7000i9d9	2022-09-20T03:52:26.531Z	47 km ESE of Littlefield, Arizona
	1.13	ak	ak022c1mqd92	2022-09-19T20:22:13.797Z	76 km SE of Kokhanok, Alaska
	0.67	ak	ak022c1mqc69	2022-09-19T21:07:48.807Z	54 km W of Anchor Point, Alaska
2.732	0.77	us	us7000i9cl	2022-09-20T01:05:42.573Z	27 km WSW of La Placita de Morelos, Mexico
2.3	1.88	us	us7000i9bw	2022-09-20T04:00:02.207Z	37 km SE of Aquila, Mexico
2.062	1.14	us	us7000i98u	2022-09-19T13:55:23.040Z	16 km SW of Champerico, Guatemala
	0.81	ak	ak022c1genep	2022-09-19T11:04:14.300Z	44 km E of Denali National Park, Alaska
	0.77	ak	ak022c1fq39l	2022-09-19T17:30:09.343Z	33 km WNW of Petersville, Alaska
0.02218	0.19	nc	nc73783386	2022-09-20T01:38:30.094Z	7km SSW of Tres Pinos, CA
0.091	0.36	us	us7000i96k	2022-09-19T23:21:26.092Z	
	0.83	ak	ak022c1bmauy	2022-09-20T03:06:26.260Z	19 km N of Four Mile Road, Alaska
	0.99	ak	ak022c1bhn1i	2022-09-20T00:59:25.768Z	15 km N of Four Mile Road, Alaska
	0.14	hv	hv73143812	2022-09-19T21:42:48.745Z	9 km SSW of Pa'auilo, Hawaii

earthquakes along the Pacific Ocean over a 1 year span referred to as *data.csv* from now on. Before we discuss how the data was processed and the network was constructed we will present how nodes and edges will be defined. Consequently, Nodes in our network represent the location of earthquake occurrences, represented as longitude and latitude pairs. The directed edges or links will connect a node a with a node b if b occurred within a maximum distance D kilometers and happened at most T days apart from a. We calculated D (in km) and T (in hrs) using the Gutenberg-Richter (GR) relation described by *Wang et al.* as follows:

$D = 10^{a_1 * m + b_1}$ where m is magnitude of an earthquake and a_1, b_1 are constants.

$T = 10^{a_2 * m + b_2}$ where m is magnitude of an earthquake and a_2, b_2 are constants.

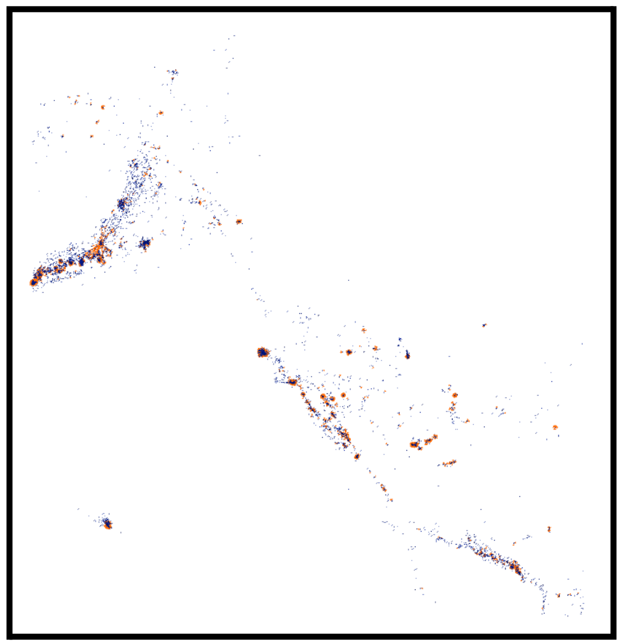
Note that we found constants a_1, b_1, a_2, b_2 by trial and error. If they are too large they result in a complete graph and if they are too small they result in an extremely disconnected graph. With the definition of nodes and edges established, we can proceed to discuss data processing and network creation. We reorganized and cleaned the data to make the creation of the network easier in the following way:

1. We read the raw *data.csv* file into a dataframe
2. We created a *nodes_df* where each row was a node consisting of *occurrence time*, *latitude, longitude, magnitude, and id*.
3. We created an *edges_df* were we Iterated over each row and applied the spatio-temporal relationship [described earlier] against all other rows (to determine edge connectivity)
4. Lastly, we saved both *nodes_df* and *edges_df* as csv files (ex: nodes.csv and edges.csv)

After having, these two output csv files, we created our network as follows:

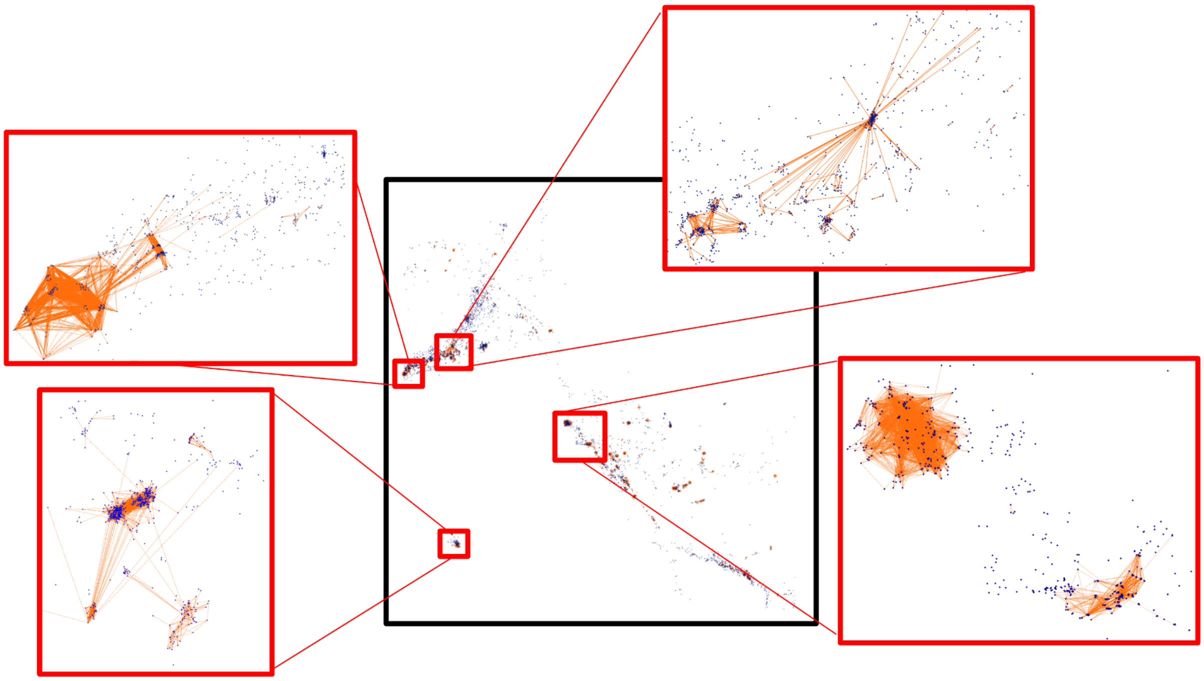
1. Opened the *nodes.csv* file and iteratively added each node to a new NetworkX Digraph.
2. Opened the *edges.csv* file and iteratively added each node to the previously created Digraph.

Fig 5.1: Entire Network Visualization



5. Network Visualization

For our network visualization, we used Gephi to plot the location of earthquakes (nodes in blue) and the influence between them (edges in red) in the figure 5.1. (See larger Image in Appendix A. In addition, we determined areas of interest (in the figure below) that indicate huge hubs of large magnitude that influence large regions and are the main reason for the topology, structure, and properties of our network.



6. Basic Statistics and Results

Some basic statistics of our network that were useful for answering our research questions are presented in *table 6.1* and also its degree distribution is shown in *Fig 6.1* that depicts that our network is scale-free because its degree distribution can be modeled by a power law function. In addition, we can observe the existence of hubs of up to 1000 nodes. Now that we have some basic info of our network, we can start by creating two types of null models: directed Erdos-Renyi and directed Random Geometric.

We used the NetworkX Erdos-Renyi graph function to create the ER Model. Thus, it had a fixed number of nodes equal to the number of nodes in our earthquake network and the presence of edges were created using `p = actual_L / max_L`. Once the ER Graph was created we randomly assigned the magnitudes to each node.

As for the random geometric model, we also used NetworkX to create it. Into this function, we passed the number of nodes `n`, a radius `r` which was computed by finding the average of the maximum distance between each node and its neighbors, and the `position` of each node in Universal Transverse Mercator (utm) coordinates in the original earthquake network. After running this function, we got an undirected random geometric network with no data (e.g. no magnitudes of earthquakes). To overcome this, we manually created a digraph and based its connectivity on the generated undirected geometric network. Lastly, we grabbed the node magnitudes of our earthquake network and assigned them to each corresponding node based on a unique id that was preserved.

Once we had a mechanism of creating these two models, we created ensembles of 1000 networks to find the average values of various metrics and plotted their degree distributions for comparison purposes (described in the table and image below). This gives us a good starting point to compare our Earthquake network to the null models and start answering our research questions.

Properties	Earthquake Network	Random Geometric Network	Random Network
Number of Nodes	7,766	7,766	7,766
Number of Edges	40,495.0	24,813.0	40,484.09
Avg Out-Degree	5.21	3.20	5.21
Avg In-Degree	5.21	3.20	5.21
Avg Max Out-Degree	941	128.43	15.87
Avg Min Out-Degree	0.0	0.0	0.0
Avg Max In-Degree	123.0	85.0	15.84
Avg Min In-Degree	0.0	0.0	0.0
Avg. Clustering Coefficient	0.24	0.17	0.000675
Avg. Shortest Path Length [largest component]	1.0	1.0	4.09
Avg. Num of Connected Components	7,765.0	7,765.0	88.05

Table 6.1: Earthquake, ER, and RG network stats

Fig 6.1: Earthquake Network Degree Distribution

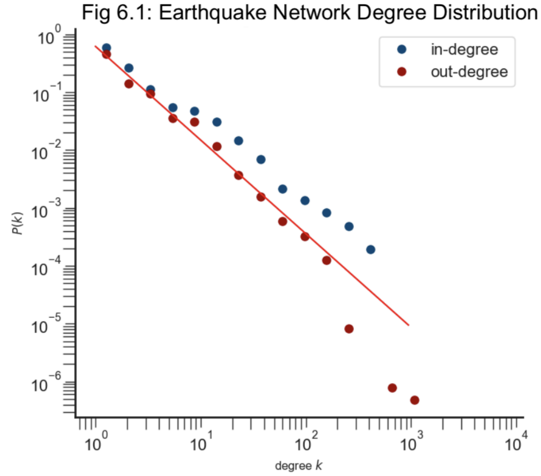


Fig 6.2: ER Network Degree Distribution

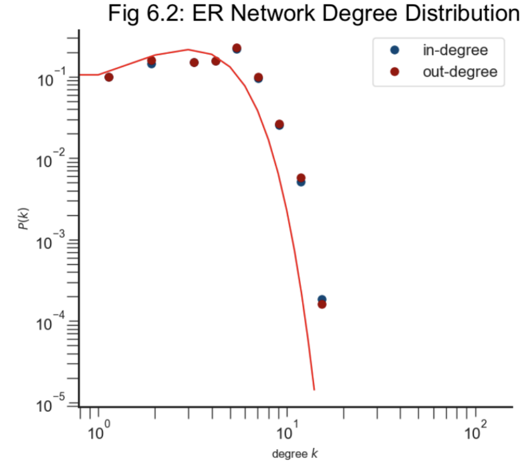
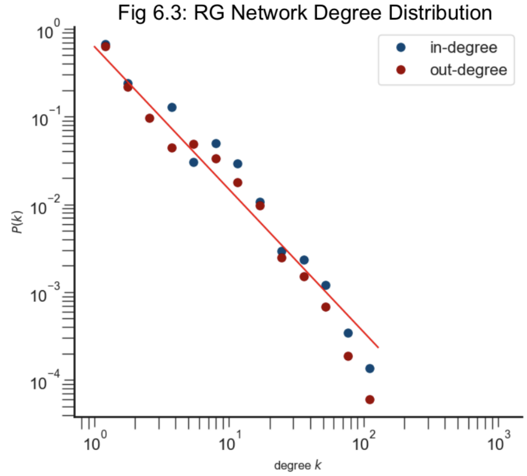


Fig 6.3: RG Network Degree Distribution



6.1 Does a higher magnitude mainshock tend to suggest the occurrence of more foreshocks in the "region"?

From the Erdos-Renyi network stats we saw that they do not present hubs because the maximum avg max out and in degrees are really tiny (also easily seen in the degree distribution). This leads to small clustering coefficient and lack of tightly knit groups. If every node is similar to every other node, it's impossible to determine influence. Therefore, this shows us that the existence of nodes with sufficiently larger magnitude (hubs) is primordial to see the presence and measure influence of an earthquake.

Hypothesis: An Earthquake's higher connectivity is always the most influential.

Method: First, we computed the ***average out-degree*** of earthquakes of Richter magnitudes between 0-1, 1-2, 2-3, 3-4,..., and 7-8. Then, we plotted a bar chart of earthquake magnitudes (x-axis) vs avg. out-degree (y-axis). [*Fig 7.1.1*]

Findings: The information depicted in the plot allows us to see that earthquakes with magnitudes between 6-7 tend to have the highest average out degree centrality or tend to connect/influence the most earthquakes after them. In this case, earthquakes of magnitude 7-8 seem to not have any out-degree however this is because the dataset contains exactly one earthquake of such magnitude. In addition, it is important to note that how the relationship between earthquakes (edge definitions) were set influences how the average out-degree is going to look. For instance, we defined that earthquakes with larger magnitude have a larger influence domain of space and time and therefore it would be expected they influence or are connected to more earthquakes. Furthermore, the edge definitions follow an exponential law (as defined above). This is the reason the average degree of earthquakes of magnitude 6-7 seems much larger than for previous magnitude values (as it grows exponentially).

From the random geometric network, we saw that it presented a similar degree distribution but had a smaller number of edges as compared to the Earthquake network. In addition, we saw that RG networks, in general, tend to have smaller degree metrics (avg deg, in-deg, out-deg) but hold a similar clustering coefficient. From this we concluded that an earthquake with higher connectivity might not always be the most influential. Furthermore, this was also supported by the idea that an earthquake is influenced by many other earthquakes around it. This means that a main-shock can influence some earthquakes and in turn those earthquakes and the main-shock can influence other earthquakes. To better describe this relationship, we decided to make use of the earthquake/node *average clustering coefficient* to find a metric that defines how much a main-shock actually influences other earthquakes to which it is connected. For instance, if the clustering coefficient for a given earthquake X is high and X connects to an earthquake Y, it would mean that a lot of earthquakes which X influences (apart from Y) also influence earthquake Y.

Hypothesis: An Earthquake's higher connectivity is not the only thing that causes how influential it is, but also the local clustering coefficient.

Method: To show how this works, we first computed the ***average clustering coefficient*** of earthquakes of Richter magnitudes between 0-1, 1-2, 2-3, 3-4,..., and 7-8 and then, we plotted a bar chart of earthquake magnitudes (x-axis) vs avg. clustering coefficient (y-axis) [*Fig 7.1.2*]

Conclusion: This plot shows us that even though high magnitude earthquakes tend to be more influential in connectivity (as in the previous plot) they tend to not be as solely influential in the occurrence of foreshocks in a region. This means other mainshocks around also influence the region which is something we would expect. Therefore, we concluded that a mainshock tends to influence the occurrence of more foreshocks in a region based on two factors (apart from magnitude): connectivity and most importantly influence power/clustering coefficient.

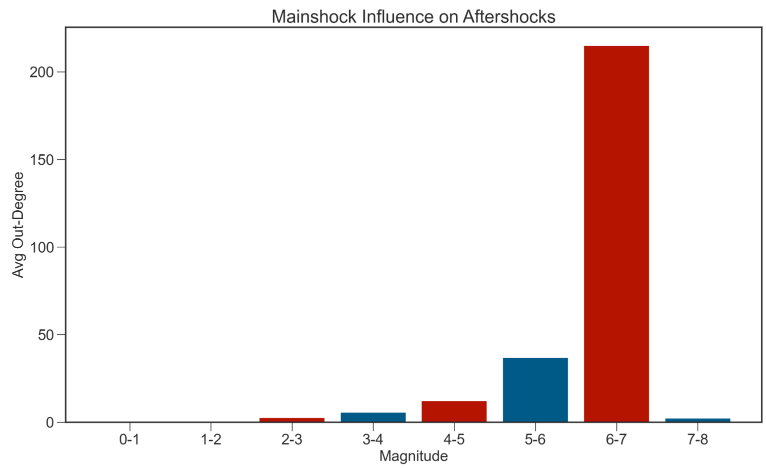


Fig 7.1.1: Mainshock influence on Aftershocks

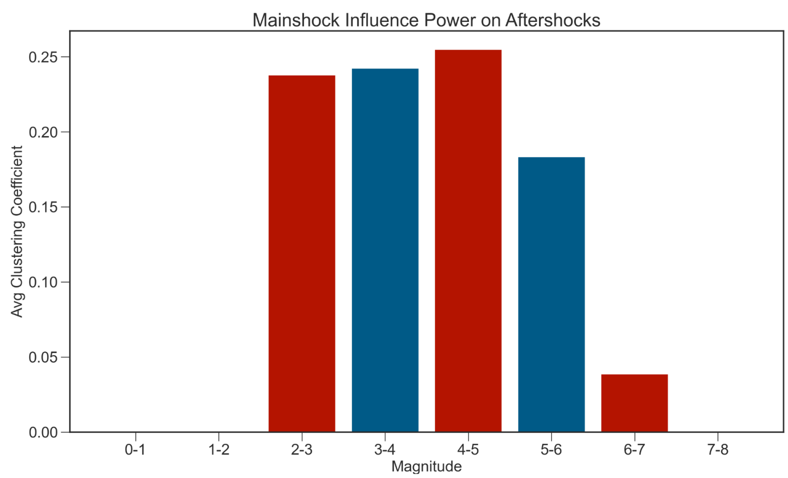


Fig 7.1.2 : Mainshock Influence Power on Aftershocks

6.2 What is the Probability of an Earthquake Occurring in the Same Region as a Past Mainshock?

<u>Formulating The Question:</u>

The obvious and essential concern on the forefront of any study of natural disasters lies in the attempt to predict the occurrence of said natural disaster. This led us to the formulation of our second question: what is the probability of an earthquake occurring in the same region as a past mainshock? To explore this question, we must first define some terminology.

Region: an earthquake *B* is in the same "region" as an earthquake *A* if it is within the sphere of influence of *A*. That is, earthquake *B* occurs within a specified distance and time after the occurrence of *A*, such that our network decidedly creates an edge from *A* to *B*. For a more detailed explanation of edge-creation in our model, see the section on nodes and edges.

Mainshock and Aftershock: An earthquake *A* is a mainshock in relation to an earthquake *B* (the aftershock) if there is an edge from *A* to *B*

Causation: In this report, we say that earthquake *A* caused earthquake *B* if there is an edge from *A* to *B*. In reality, the connection between the two earthquakes can most accurately only be described as a correlation as there is no evidence for direct causation.

Hypothesis 2:

To guide our exploration of this question, we hypothesized that mainshocks of larger magnitude would have a higher probability of causing aftershocks than mainshocks of lower magnitude. It follows, then, that the probability of an earthquake occurring within a region would be greater if there were more high-magnitude earthquakes that occurred within that region.

Method:

The logical process of our method followed the same logic as our hypothesis. We began by exploring the probabilities of mainshocks of different magnitudes causing subsequent aftershocks. We then took a step further to explore the probabilities of aftershock occurrence based on the magnitude of the aftershock. That is, we granulated on the original probability calculations to see, for example, what would be the probability of a magnitude-2 earthquake occurring within the same region as a magnitude-7 mainshock. The process of these calculations was a simple process of observing the patterns and running calculations on our existing network. To determine the probability of a mainshock of magnitude-*x* causing an aftershock, we would simply count up the number of magnitude-*x* earthquakes in our network, and use that to divide on the number of magnitude-*x* earthquakes that actually caused subsequent aftershocks. To then find the probability of that mainshock causing an aftershock of a specific magnitude, we simply counted up how many of those aftershocks were of a certain magnitude.

Probability of magnitude-*x* mainshock causing an aftershock = y/z, where

y = number of magnitude-*x* earthquakes that cause an aftershock

z = total number of magnitude-x earthquakes

Findings and Analysis:

The calculated probabilities of mainshocks of all magnitudes causing aftershocks is shown in Figure 7.2.1. It should be noted that since all our data on the magnitudes of earthquakes were decimal numbers between 1 and 8, the categorization of this data has been bucketed such that the magnitude of the earthquake has been rounded to the nearest whole number. In Figure 7.2.1, and the subsequent graph in Figure 7.2.2, we can see that there is a very clear positive relationship between the magnitude of the mainshock, and the probability of an aftershock occurring within the same region. To analyze the significance of this discovery, we then compared these results to the same calculations performed on the generated null models. In the ER model, we can see that mainshocks of all magnitudes have close to a 100% probability of causing aftershocks. This shows that there is a definite significance within the patterns exhibited in our network, as they are clearly not random. The probabilities graph shown in the random-geometric model in Figure 7.2.4, however, displays a much more similar pattern to our network. Though there are differences between the two graphs – i.e. an evident “flat” portion in the middle of the random-geometric graph, in comparison to the much more linear growth in our own network – the similarity within the generally positive correlation in both graphs was enough for us to explore this finding. Through analysis of our data, the similarity between these two models makes sense as there are clear biases that limit our calculations. The most obvious of these biases is that we have significantly more nodes for low-magnitude earthquakes than we do for high-magnitude earthquakes. This naturally exaggerates our data as the probability calculations of high-magnitude earthquakes will tend to be much more inflated since every single aftershock makes up a greater proportion of overall instances of mainshocks of that magnitude. With this bias in mind, we still believe that the positive correlation within our data is not completely random and therefore of proficient significance in supporting the first part of our hypothesis: larger magnitude mainshocks have a higher probability of causing subsequent aftershocks.

Magnitude of Mainshock	Probability of Mainshock Causing Aftershock
1	NO DATA
2	0.32230
3	0.4700
4	0.6497
5	0.8218
6	0.9688
7	1.0000
8	1.0000

Figure 7.2.1: Probability of Aftershock Occurrence VS Mainshock Magnitude, Table

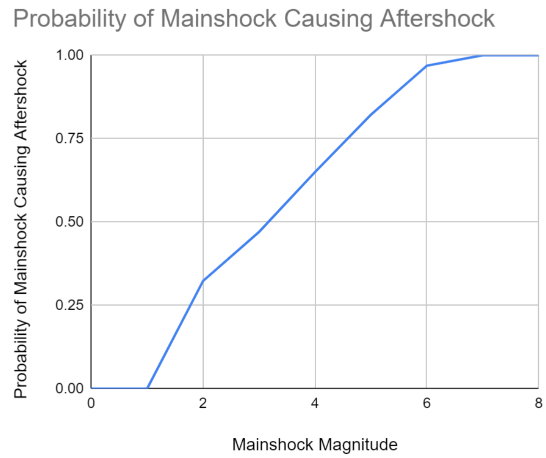


Figure 7.2.2: Probability of Aftershock Occurrence VS Mainshock Magnitude, Line Graph

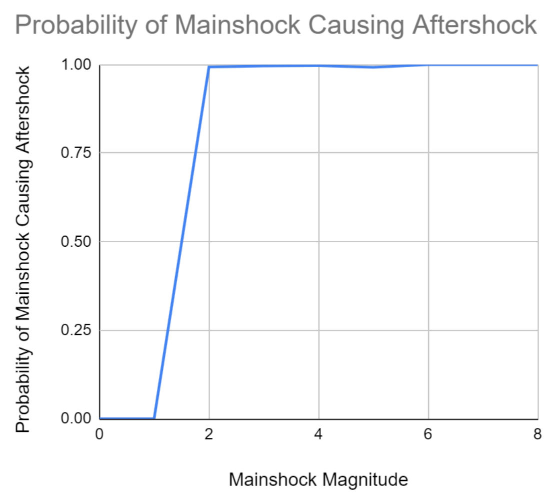


Figure 7.2.3: Probability of Aftershock Occurrence VS Mainshock Magnitude Within the Erdos-Renyi Model, Line Graph

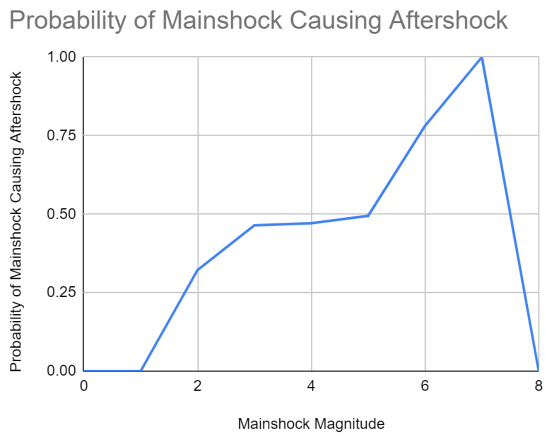


Figure 7.2.4: Probability of Aftershock Occurrence VS Mainshock Magnitude Within the Random-Geometric Model, Line Graph

Next, Figure 7.2.5 shows the data we calculated through further exploring the granularity of these probability calculations, by examining the data based on the magnitude of the aftershock. We can see, in this table, that the general trend is that mainshocks of low magnitudes almost only cause aftershocks of low magnitudes, whereas high-magnitude mainshocks cause earthquakes of a wide variety of magnitudes; the majority of aftershocks caused by magnitude-2 mainshocks have a magnitude of 2 or 3. On a higher magnitude mainshock, the magnitudes of the aftershocks are spread out. A magnitude 7 mainshock, for example, shows aftershocks of magnitudes 2, 3, 4, 5, 6, and 7. What this observation tells us is that magnitudes of higher earthquakes are not only more likely to cause subsequent earthquakes, the probability of the magnitude of those aftershocks is determined by, with a positive correlation, the magnitude of the mainshock.

Magnitude of Mainshock	Probability of Aftershock Being of Magnitude ___							
	1	2	3	4	5	6	7	8
1	No Data	No Data	No Data	No Data	No Data	No Data	No Data	No Data
2	No Data	0.1532	0.7381	0.1004	0.0082	0	0	0
3	No Data	0.0721	0.7949	0.1140	0.0178	0.0011	0	0
4	No Data	0.0354	0.3793	0.4673	0.1087	0.0092	0	0
5	No Data	0.0409	0.4015	0.4489	0.1003	0.0080	0.0004	0
6	No Data	0.0420	0.4631	0.3945	0.0880	0.0120	0.0004	0
7	No Data	0.10370	0.6848	0.1705	0.0375	0.0030	0.0006	0
8	No Data	0	0.5000	0	0.5000	0	0	0

Figure 7.2.5: Percentage of Aftershocks of Each Magnitude

Conclusion:

As demonstrated in the *Findings and Analysis* section, we have shown strong evidence to support our hypothesis that higher magnitude mainshocks have a higher likelihood of causing aftershocks. Thus, it logically follows that the probability of an earthquake occurring within a region will be higher as there are more magnitudes within that region, and that probability would only increase as the magnitude of those existing earthquakes increases. With these findings, we were then able to quantify our results with a simple probability calculation.

We had the probability of an earthquake occurring based on the magnitude of a mainshock within that region. Thus, in the simplest case, the probability of an earthquake occurring in a region, as a result of a mainshock, is the probability of the mainshock in the region causing an aftershock. Take, for example, a region where a magnitude-5 earthquake has occurred. The probability of an earthquake occurring in that region would be $P(5) = 0.8218$.

This calculation gets more complicated when we take into account that multiple mainshocks occur within the same region, any of which could contribute to the occurrence of an aftershock within that region. We must keep in mind that any of these mainshocks causing an aftershock are not mutually exclusive events and therefore the probability cannot be calculated with a simple summation of unions. The probability of an earthquake occurring as a result of a mainshock within a region is equal to the union of the probabilities of all the earthquakes that exist within the region, causing a subsequent aftershock.

Take an earthquake that exists in region R. Within R, there are earthquakes of magnitudes 4, 5, 3, and 6. The probability of an earthquake occurring in region R would thus be $P(4) \cup P(5) \cup P(3) \cup P(6)$. If these earthquakes causing a subsequent aftershock were mutually exclusive events, the probability of an aftershock occurring within this region would simply be $P(4) + P(5) + P(3) + P(6)$. Since, however, these probabilities are not mutually exclusive, we must amend this calculation to subtract any double-counts, as seen in the venn diagram in Figure 7.2.6. The new equation for the probability of an earthquake occurring within region R is thus

$P(4) \cup P(5) \cup P(4) \cup P(6) = P(4) + P(5) + P(3) + P(6) - P(4 \cap 5) - P(4 \cap 3) - P(4 \cap 6) - P(5 \cap 3) - P(5 \cap 6) - P(3 \cap 6) + P(4 \cap 5 \cap 3) + P(4 \cap 5 \cap 6) + P(4 \cap 3 \cap 6) + P(5 \cap 3 \cap 6) - P(4 \cap 5 \cap 3 \cap 6).$

This type of probability calculation can be made for any such scenario, by using the probabilities of mainshocks of different magnitudes causing subsequent earthquakes within their region. Furthermore, these probabilities can be substituted with our granulated data from figure 7.2.5, in order to predict the probability of an earthquake of any specific magnitude occurring within a region .

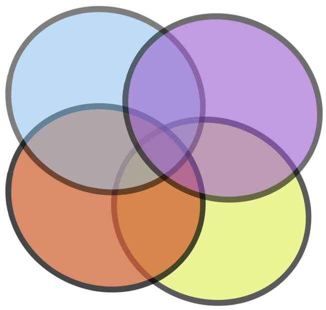


Figure 7.2.6: Venn Diagram of the Probabilities of Mainshocks Causing Aftershocks, Showing Overlaps of Mutually Non-Exclusive Events

.3 Is there a domino effect of foreshock magnitudes near each mainshock?

Hypothesis 3: We expect to see a domino effect from an earthquake of high magnitude decreasing at each neighbor level.

Method: We wanted to explore this idea in our earthquake network to see if we were able to find a domino effect of foreshock magnitudes near each mainshock for our data. We did this by traversing through the network starting at an origin node, which had magnitudes of 5.5 or higher. As we traversed from the origin node, we found the average magnitude of all of its first-neighbors, then all its second-neighbors and so on recursively calculating the averages (up to the fifth neighbors) to see if there was a decline in magnitude at each level.

Findings: Our overall findings were that there was always a decrease in magnitude from the origin node to the 1st neighbors. We observed that there was a ripple or cascade of earthquakes of varying magnitudes after the first neighbors as seen in Figure 1. The full results of each level's average magnitudes can be seen in Appendix Table A.

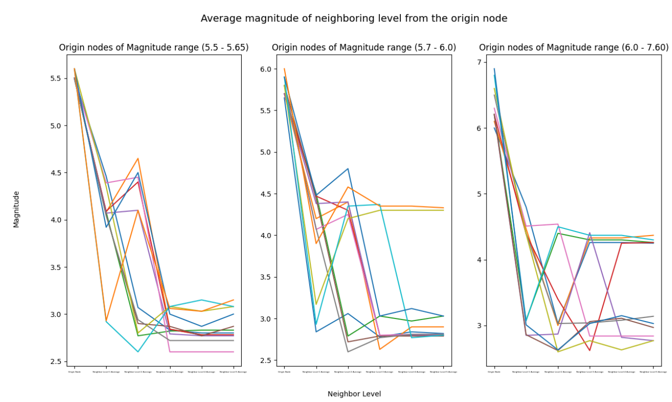


Figure 7.3.1: Average magnitude of neighboring levels from the origin node, based off [Appendix Table 1](#) and separated into 3 plots to show Origin Nodes starting at varying Magnitudes

We initially expected all the neighboring nodes to decrease at each level. Our actual findings did show a downward trend as you move away from the origin node, but there were still lower levels that had higher average magnitudes than their predecessor. As the magnitude of the original node was increased the downward trend became less prominent, and more unpredictable. Shebalin's journal article mention's multiple "earthquake chains" which are a propagation of the chain preceding another earthquake, and mentions other factors that are also correlated to the chaining effect of earthquakes, such as fault lines. In our model we only looked directly at average magnitude, thus this could have impacted our findings.

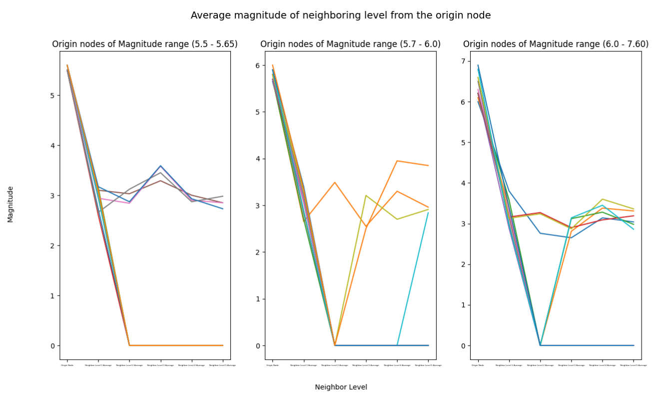


Figure 7.3.2: Average magnitude of neighboring levels from the origin node of our ***Erdos-Renyi Model***

When we compared our model with the Erdos-Renyi Random model we noticed that there were far less "Earthquake chains", or the chains would break around the 2nd neighbor resulting in an average of 0 magnitude. We found that our Earthquake Model had more of a domino effect on magnitude showing a step type pattern whereas the random model either dropped to an average of 0 magnitude indicating that level had no neighbors or it would cluster around a magnitude of 3, which was around the same endpoint that our network had. Overall, we were able to see that the domino effect that we hypothesized was in fact not random when compared to the Erdos-Renyi Random model.

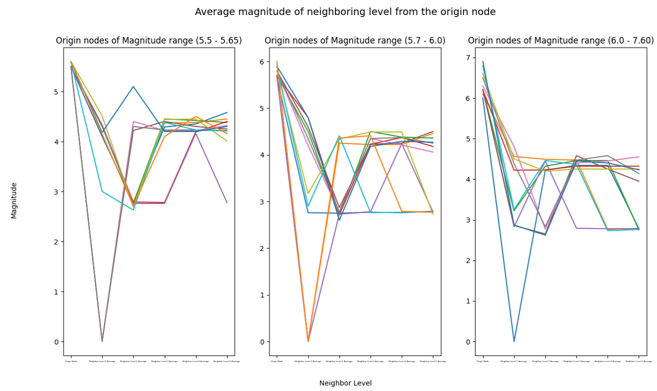


Figure 7.3.3: Average magnitude of neighboring levels from the origin node of our ***Random Geometric Model***

When we compared our model with a Random Geometric model (RG Model) which kept the magnitudes and locations the same, but changed the edges, and noticed that they had similar results but noticed there was not as prominent of a downward trend line. The difference was most noticeable in the 5.5 - 5.65 range magnitudes as our network trended down and the Random Geometric model had a flatter trendline that plateaued around an average magnitude of 4. For the magnitudes with ranges 5.7 - 6.0 our earthquake network only had a couple outliers from the downward trend whereas the RG Model shows the transpose of that. Lastly, for the magnitudes with ranges 6.0 -7.60, our Network results for this were noisy, but we were able to see a downward trend in just over half of the earthquake chains where the RG model only had 3 chains that trended downward. Overall, we were able to see a difference in the RG Model vs our model that showed our findings were significant. There were not as prominent as the comparison to the Erdos-Renyi Random model since the magnitudes and locations remained the same with new connections vs the

Erdos-Renyi Random model which was completely random connections and locations then also randomizing the magnitudes from the original graph to each node.

Conclusion: Overall, on average we did see the domino effect that we expected. Suggesting that an earthquake of high magnitude does have an influence in the occurrence of its connected neighbors. We noticed at higher magnitudes that there were still lower levels that had higher average magnitudes than their predecessors. We hypothesize that this is due to external factors such as fault lines, or other environmental factors. As our model only looked directly at the average magnitude, distance and time this may have impacted the noisiness seen in the graphs for the origin nodes with higher magnitudes. For future work we plan on exploring the environmental factors to see what kind of effect they have on magnitude as well.

8. Discussion

We were able to frame our work in the current literature and satisfactorily answer all three of our research questions with sophisticated approaches. Overall, we achieved all our research goals for this research project.

Furthermore, We learned that modeling time-space influence accurately is a complex task. This was especially more noticeable in the generation of null models. When we attempted to generate a random geometric model, we were able to get close (in terms of topology and stats) to our Earthquake network. However, due to the complexity of the time-space influence metric we could not fully model our network. Another thing we learned is that areas that contain huge hubs with large magnitudes tend to be hard to analyze and visualize properly because they become too cluttered. There are other approaches in the literature that suggest other ways of defining nodes that get rid of this problem entirely. One example is the one presented by *Xuan He et al.* where they partition the surface of the earth into cells of x meters by x meters . Then, they make the nodes of their network each of these cells and connect two cells if they have earthquakes that influence each other within them.

We think that our second research question could be expanded further. Specifically, more complex statistics could be employed to determine more accurate probabilities. We partially came up with one approach that we explored but could not fully develop due to time and knowledge constraints (described in the methods section) Another area in our research project that could be further expanded is improving the measure of influence of an earthquake that we explored in our first research question. For example, ideally we should have a single metric that combines connectivity (out-degree centrality) and clustering coefficient. Furthermore, adding certainty/error into this measurement can make our analysis stronger and allow us to answer the question in a simple and more concise way with a good metric to support it.

Lastly, we believe that the creation and selection of null models could be researched further and improved. Such a null model should consider directed graphs, preserve degree distribution, and consider the spatio-temporal implications of our network. We observed that a random geometric null model closely describes our network. However, some tweaks need to be done in the areas mentioned to make it a perfect model of our network.

In terms of future work, a better measure of influence power could be found as related to question one. For instance, such influence could be described as a combination of more than just the magnitude of an earthquake and could include certainty/error. Furthermore, future work can be performed in better answering our second research question by using more complex statistics to determine more accurate probabilities. Also, there is room for finding a simplified network (e.g. null model) that has all key topological and structural properties that make our network have its intrinsic properties and models it more closely.

9. Methods

9.1 Network Creation Approach

To create our network, we first read the raw csv data and created two data frames (that later on we store as nodes.csv and edges.csv). For the nodes dataframe, we just go through each node in the input data and add them to the nodes dataframe. As for the edges, we compute the edge connectivity by using the spatio-temporal relationship mentioned earlier. The code looks like:

```
    @classmethod
    def df_from_csv(cls, input_raw_csv : str, output_nodes_csv : str, output_edges_csv :
str):

        start_time = time.time()

        # Some Variables
        input_df_columns = ['id','time','latitude', 'longitude', 'mag']
        nodes_df_columns = ['id', 'lat', 'long', 'mag', 'time']
        edges_df_columns = ["Source", "Target"]

        # Prepare temporary data frame
        input_df = pd.read_csv(input_raw_csv, usecols = input_df_columns)
        input_df.rename(columns={"latitude":"lat", "longitude":"long"}, inplace = True)
        input_df['time'] = pd.to_datetime(input_df['time'])

        # Create new dataframes for nodes and edges
        nodes_df = None
        if exists(output_nodes_csv):
            nodes_df = pd.read_csv(output_nodes_csv, usecols = nodes_df_columns)
        else:
            nodes_df = pd.DataFrame(data=input_df[nodes_df_columns])
            nodes_df.to_csv(output_nodes_csv, index=True)

        edges_df = pd.DataFrame(columns=edges_df_columns)
        if exists(output_edges_csv):
            edges_df = pd.read_csv(output_edges_csv, usecols = edges_df_columns)
            return nodes_df, edges_df

        # Vars used to hold source and destination earthquakes
        source : EarthQuake = None
        dest : EarthQuake = None

        # Iterate over the data to check if dest earthquake is within influence domain of
source earthquake
        # and include them in edges_df if necessary.
        for index, row in input_df.iterrows():
           source = EarthQuake(row['id'], row['time'], row['lat'], row['long'], row['mag'])
           for index2, row2 in input_df.iterrows():
```

```
            if(index == index2): continue

            dest = EarthQuake(row2['id'], row2['time'], row2['lat'], row2['long'],
row2['mag'])
            is_in_domain = source.is_in_influence_domain(dest)

            if is_in_domain:
                new_row = {'Source':source.id, 'Target':dest.id}
                edges_df = edges_df.append(new_row, ignore_index=True)

        # Writing it to csv to avoid doing all this computation (unless we have to)
        edges_df.to_csv(output_edges_csv, index=False)

        end_time = time.time()
        print(f"\nProcessing Time: {timedelta(seconds=floor(end_time-start_time))}\n")

        return nodes_df, edges_df
```

Once we have these two dataframes, we open them and create a networkX Digraph to be able to perform calculations. This looks like the following:

```
    @classmethod
    def create_graph(cls, nodes_df=[], edges_df=[], is_undirected=False):
        G = nx.DiGraph()
        for idx, node in nodes_df.iterrows():
            lat = float(node['lat'])
            long = float(node['long'])
            coords = (lat, long)
            G.add_node(str(node['id']), pos=coords, magnitude=node['mag'], id=idx)

        for idx, edge in edges_df.iterrows():
            G.add_edge(edge['Source'], edge['Target'])

        if is_undirected:
            return G.to_undirected()
        else:
            return G
```

9. 2 Research Question Approaches

First Research Question

To be able to plot earthquake-magnitudes versus average out-degree centrality, we first computed a list of pairs of (out-deg, magnitude) as described in the following piece of code:

```
    @classmethod
    def compute_outdeg_and_maginute(cls, G):
        results = []
        for node in G.nodes(data=True):
            results.append((G.out_degree(node[0]), node[1]['magnitude']))
```

```
        return results
```

After this, we used this information to compute the average out-degree of earthquakes of Richter magnitudes between 0-1, 1-2, 2-3, 3-4,..., and 7-8. This is shown in the code below

```
# Computing average out degree of earthquakes
        # data = list of pairs (out-deg, magnitude)
        for d in data:
            out_degrees_sum[math.floor(d[1])] += d[0]
            num_of_outdegrees[math.floor(d[1])] += 1

        for i in range(0, len(x)):
            y.append(out_degrees_sum[i] / (1 if num_of_outdegrees[i] == 0 else
num_of_outdegrees[i]))
```

Furthermore, to be able to plot earthquake-magnitudes versus average clustering coefficient, we first computed a list of pairs of (clustering coefficient, magnitude) as described in the following piece of code:

```
    @classmethod
    def compute_outdeg_and_avg_clustering_coeff(cls, G):
        results = []
        for node in G.nodes(data=True):
            results.append((nx.clustering(G, node[0]), node[1]['magnitude']))
        return results
```

After this, we used this information to compute the average clustering coefficient of earthquakes of Richter magnitudes in the ranges described earlier as follows:

```
        # Computing average clustering coefficient of earthquakes in specific ranges
        # data = list of pairs (avg clustering coeff, magnitude)
        for d in data:
            clustering_coeff_sum[math.floor(d[1])] += d[0]
            num_of_clustering_coeffs[math.floor(d[1])] += 1

        for i in range(0, len(x)):
            y.append(clustering_coeff_sum[i] / (1 if num_of_clustering_coeffs[i] == 0 else
num_of_clustering_coeffs[i]))
```

Second Research Question

In order to calculate the probability of an earthquake occurring within the same region as past mainshocks, we can define a region as the area of influence around an earthquake. For example, earthquakes A and B are in the same region if they are connected by a node. That is, A occurs within a certain distance and time from B such that B has an edge connecting to A or vice versa. In order to calculate the probability of an earthquake occurring in the same region as existing earthquakes, we first calculated the probability of earthquakes of all magnitudes causing subsequent earthquakes. Here, this “causation” refers to a connection from the mainshock to a subsequent aftershock, despite this being more of a correlation than a provable causation.

Thus our objective is to calculate the probability of an earthquake occurring within a specific region by determining the probability of those existing earthquakes causing subsequent earthquakes. We thus calculated P(X),

where *P(X)* is the probability that earthquake *X* causes a subsequent aftershock.

First, to calculate P(X), we needed to calculate the probability of an earthquake causing subsequent earthquakes based on its magnitude. We thus calculated P(X) such that x is the magnitude of earthquake X, for all x = 1, 2, 3, 4, 5, 6, 7, 8. To do this, we counted all occurrences of earthquakes X, and used that to divide on all earthquakes X such that X caused an aftershock. This gave us P(X) for mainshocks of all magnitudes. The following code was used for this calculation:

```
'''
   Method: calculate_probabilities
   Param:  x: desired magnitude (Integer) of mainshock
           G: Graph of earthquakes
   Description: Calculates the probabilities of subsequent earthquakes occurring as a result
of mainshocks of magnitude x
'''

@classmethod
def calculate_probabilities(cls, x, G):

   neighbor_list = []
   # get all nodes with magnitude x
   nodes_with_spec_mag = []
   nodes_that_cause_subsequent_earthquakes = 0

   subsequent_earthquakes = {1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0, 7: 0, 8: 0, 9: 0, 10: 0}

   for node in G.nodes(data=True):
       if round(node[1]['mag']) == x:
           nodes_with_spec_mag.append(node)
           neighbor_list = [nbr for nbr in G.neighbors(node[0])]
           if len(neighbor_list) > 0:
               nodes_that_cause_subsequent_earthquakes =
nodes_that_cause_subsequent_earthquakes + 1
               # print( G.nodes[neighbor_list[0]]['mag'] )
           for sub_node in neighbor_list:
               mag = round(G.nodes[sub_node]['mag'])
               subsequent_earthquakes[mag] = subsequent_earthquakes[mag] + 1
           # print(f"Neighbour({node}) = {neighbor_list}")

   total_nodes = len(nodes_with_spec_mag)
   total_subsequent_earthquakes = subsequent_earthquakes[1] + subsequent_earthquakes[2] +
subsequent_earthquakes[3] + subsequent_earthquakes[4] + subsequent_earthquakes[5] +
subsequent_earthquakes[6] + subsequent_earthquakes[7] + subsequent_earthquakes[8] +
subsequent_earthquakes[9] + subsequent_earthquakes[10]

   prob_sub_earthquake_caused =
ProbabilityCalculations.divide(nodes_that_cause_subsequent_earthquakes, total_nodes)
```

```
 average_subsequent_earthquakes_caused_per_mainshock =
ProbabilityCalculations.divide(total_subsequent_earthquakes,total_nodes)

   ProbabilityCalculations.probabilities_df.loc[x, ['Mainshock Magnitude']] = x

   # print("total number of nodes with magnitude ", x, ": \t\t\t\t\t\t\t\t\t\t\t",
total_nodes)
   ProbabilityCalculations.probabilities_df.loc[x, ['Number of Mainshock']] = total_nodes

   # print("total number of nodes that cause subsequent nodes: \t\t\t\t\t\t\t\t\t",
nodes_that_cause_subsequent_earthquakes)
   ProbabilityCalculations.probabilities_df.loc[
       x, ['Mainshocks Which Cause Subshocks']] = nodes_that_cause_subsequent_earthquakes

   # print("total number of subsequent earthquakes: \t\t\t\t\t\t\t\t\t\t\t",
total_subsequent_earthquakes)
   ProbabilityCalculations.probabilities_df.loc[x, ['Number of Subshocks']] =
total_subsequent_earthquakes

   # print("probability of node with magnitude ", x, " causing subsequent earthquakes:
\t\t\t\t", prob_sub_earthquake_caused)
   ProbabilityCalculations.probabilities_df.loc[
       x, ['Probability of Mainshock Causing Subshock']] = prob_sub_earthquake_caused

   # print("average number of subsequent earthquakes caused by earthquakes of magnitude ",
x, " : \t", average_subsequent_earthquakes_caused_per_mainshock)
   ProbabilityCalculations.probabilities_df.loc[
       x, ['Average Number of Subshocks per Mainshock']] =
average_subsequent_earthquakes_caused_per_mainshock

   for i in range(1, 11):
       current_column = 'Subshock Magnitude ' + str(i) + ' Occurence Probability'
       ProbabilityCalculations.probabilities_df.loc[x, [current_column]] =
ProbabilityCalculations.divide(
           subsequent_earthquakes[i], total_subsequent_earthquakes)


'''
   Method: get_all_probabilities
   Param: G: Graph of earthquakes
   Description: Calculates probabilities of subsequent shocks of earthquakes of all
magnitudes (0-10)
'''
@classmethod
def get_all_probabilities(cls, G, path):
   for i in range(11):
       print("-------------------MAGNITUDE = ", i, "-------------------")
       ProbabilityCalculations.calculate_probabilities(i, G)
       print("\n\n\n")
   print(ProbabilityCalculations.probabilities_df.to_string())
   ProbabilityCalculations.probabilities_df.to_csv(path, index=False)
```

```python
# custom divide function to take care of division by 0
@classmethod
def divide(cls, a, b):
   if (b == 0):
       return 0
   else:
       return a / b
```

Now that we had P(X), we could calculate the probability of an earthquake occurring within a region based on the earthquakes that exist within that region. If a region contained earthquakes of magnitudes 5 and 6, for example, the probability of an earthquake occurring within that region would be:

$$P(5 \cup 6) = P(5) + P(6) - P(5 \cap 6),$$

Retrieved from: https://www.thoughtco.com/probability-union-of-three-sets-more-3126263

This concludes our formula for the calculation for the probability of an earthquake occurring within a region as past mainshocks:

P(B|A)=P(A) + P(B) - P(A and B),

This formula is expandable for more mainshocks by simply adding up the probabilities of each mainshock and subtracting the overlap since these probabilities are not mutually exclusive.

This tactic gets tricky, however, when we try to get the exact value for the intersection operations, such as $P(5 \cap 6)$. Trying to calculate the intersection between the two probabilities would take further analysis of the events and the exact mathematics in determining those values. Given more time and resources, this would be the next logical step for us to explore.

Furthermore, to make our probability calculations more sophisticated, we would have to look at more than just magnitude in determining the probability of a mainshock causing aftershocks. We could look at the longitude/latitude values, as well as the time of occurrence as possible factors. Again, with additional time, these factors could be explored to help make our probability calculations as sophisticated and inclusive as possible.

Third Research Question

First we found all of the origin nodes that had a magnitude of 5.5 or higher and added them to an array. We also create a list for the `array_avg_mag_per_level` adding the origin node at position. This will be the list of nodes that we will use as a starting point for our recursion.

```python
def nodes_with_specific_magnitude(cls, G, csv_path, img_path):

        #check for mag >= 5.5
        for node in G.nodes(data=True):
            if node[1]['mag'] >= spec_magnitude:
                nodes_with_spec_mag.append(node)

        original_node_list =len(nodes_with_spec_mag)
        rows, cols = (original_node_list+1, 7)
        array_ave_mag_per_level = [[0 for i in range(cols)] for j in range(rows)]

        for quakes in nodes_with_spec_mag:
            array_ave_mag_per_level[count][neighbour_level] = quakes[1]['mag']
            count = count + 1
```

Then using that array we get the neighbors at each level calculating the average of each neighbor. First it sums all the magnitudes of the nodes at the first neighboring level, then it divides by the number of neighbors. If the neighbor level is <6 it recursively calls the function to add the next level of neighbors, until we have 5 levels of neighbors' average magnitudes calculated. Once we gathered all of our data we plotted it on 3 separate plots to reduce the noise. This plot was then used to compare the starting point magnitudes and its effects on its neighbors. As well as comparing our Network against the null networks.

```python
def calc_average_magnitude(cls, nodes_with_spec_mag, spec_magnitude, G,
neighbour_level_original, array_ave_mag_per_level, original_node_list):

        for n in range(len(nodes_with_spec_mag)):
            if n > original_node_list: break;
            neighbour_level = neighbour_level_original
            neighbors = []
            neighbors = [nbr for nbr in G.neighbors(nodes_with_spec_mag[n][0]) if
G.nodes[nbr]['mag'] < spec_magnitude]

            # Calculate the average magnitude for all the neighbors
            average_mag = 0
            out_node_list = []
            for element in neighbors:
                average_mag += G.nodes[element]['mag']

            # if more than 1 element divide by elements or if 1 element leave the same
            if len(neighbors) > 1:
                average_mag = round(average_mag / len(neighbors), 2)
            elif len(neighbors) == 1:
                average_mag = average_mag

            if neighbour_level == 6: break;
            if len(out_node_list) > 0:
                cls.calc_average_magnitude(out_node_list, spec_magnitude, G,
(neighbour_level + 1),    array_ave_mag_per_level, original_node_list)
```

9.3 Approach to Generating Null Models

Erdos-Renyi Random Model

We used the NetworkX Erdos-Renyi graph function and the code that was provided from the Real World Networks - Null Models Exercise in CPSC 572 to create our ER Null models. We used the same amount of nodes as our original graph `GN = len(G.nodes()`. Then used that to find the probability of possible edges, `p = actual_L / max_L`. Once we had those parameters we used the networkX function to find 1000 instances which we then used to find the statistics for our ER Random model. The loading bar was found from https: // www.geeksforgeeks.org / progress - bars - in -python.

```
def er_network(cls, G, csv_path, one_network=False,num_networks=1000):

        GN = len(G.nodes())
        max_L = GN * (GN - 1) / 2
        actual_L = len(G.edges())
        p = actual_L / max_L

        nx.erdos_renyi_graph(GN, p, directed=False)

        resultStats = Stats()
        for _ in tqdm(range(num_networks, ascii=False, ncols=75):
            network = nx.erdos_renyi_graph(GN, p, directed=False)
            print(network);
            stats = Stats.from_undirected_network(network)
            resultStats.add(stats)

        resultStats.avg(num_networks)
        resultStats.to_csv(csv_path)
```

Random Geometric Model

We used NetworX's random geometric network function to create a null model. This function accepts the number of nodes `n`, `radius`, and `position`. For computing the radius, we found the average of the maximum distance of a each node to its neighbors as described in the code below:

```
    @classmethod
    def avg_euclidean_dist(cls, G, umt_dict, avg_dist_csv):

        # Check if csv already exists and return that as a dict
        if exists(avg_dist_csv):
            return pd.read_csv(avg_dist_csv, usecols =
['avg_euclidean_dist']).iloc[0]['avg_euclidean_dist']

        # Used to store the result
        avg_dist_df = pd.DataFrame(columns=['avg_euclidean_dist'])
```

```
        avg_dist_tot = 0

        # For each node find the maximum distance between itself and its neighbors
        # Then find the average of that
        for node in umt_dict.items():
            avg_dist_tmp = 0
            for neighbor in list(G.successors(node[0])):
                neighbor_utm_coords = utm.from_latlon(G.nodes[neighbor]['pos'][0],
G.nodes[neighbor]['pos'][1])
                avg_dist_tmp = max(avg_dist_tmp, math.dist([node[1][0], node[1][1]],
[neighbor_utm_coords[0], neighbor_utm_coords[1]]))
            avg_dist_tot += avg_dist_tmp

        avg_dist_tot /= len(umt_dict)
        avg_dist_df = avg_dist_df.append({'avg_euclidean_dist':avg_dist_tot},
ignore_index=True)
        avg_dist_df.to_csv(avg_dist_csv, index=False)

        return avg_dist_tot
```

For the position, we created a dictionary of node ids (simple integers) and the coordinates of each node in universal transverse mercator (utm) coordinates. To create this dictionary we did the following:

```
    @classmethod
    def to_umt(cls, G, csv_path):

        # Check if csv already exists and return that as a dict
        if exists(csv_path):
            utm_df = pd.read_csv(csv_path, usecols = ['id','utm_easting', 'utm_northing'])
            utm_dict_no_id = {i[0]: (i[1], i[2]) for i in list(utm_df[['utm_easting',
'utm_northing']].itertuples(index=True, name=None))}
            utm_dict_with_id = {i[1]: (i[2], i[3]) for i in list(utm_df[['id', 'utm_easting',
'utm_northing']].itertuples(index=True, name=None))}
            return utm_dict_no_id, utm_dict_with_id

        # Create the data frame because the file does not exist
        utm_df = pd.DataFrame(columns=['utm_easting', 'utm_northing', 'id'])

        # Iterate over each node in the network
        for node in G.nodes(data=True):
            # NOTE: We ignore the UTM Zone because earthquakes are close to each other only
in the pacific plate.
            utm_coords = utm.from_latlon(node[1]['pos'][0], node[1]['pos'][1])
            new_row = {'id': node[0], 'utm_easting': utm_coords[0],
'utm_northing':utm_coords[1]}
            utm_df = utm_df.append(new_row, ignore_index=True)

        # Write the dataframe into csv
        utm_df.to_csv(csv_path)

        # Convert the dataframe to a dict and return it
        utm_dict_no_id = {i[0]: (i[1], i[2]) for i in list(utm_df[['utm_easting',
```

```
'utm_northing']].itertuples(index=True, name=None))}
        utm_dict_with_id = {i[1]: (i[2], i[3]) for i in list(utm_df[['id', 'utm_easting',
'utm_northing']].itertuples(index=True, name=None))}
        return utm_dict_no_id, utm_dict_with_id
```

Once we had these three parameters and the positions, we realized that networkX's function was going to produce an undirected network which was not going to be useful for our comparison. To solve this issue, we created a dictionary of node ids (simple integers) and the magnitudes of earthquakes coming directly from our network as follows:

```
@classmethod
   def get_node_info(cls, csv_path):
       nodes_df = pd.read_csv(csv_path)
       return [(i[0], i[1], i[2]) for i in list(nodes_df[['mag',
'time']].itertuples(index=True, name=None))]
```

With all this info, we ran the function described below:

```
   @classmethod
   def rg_network_stats(cls, n, radius, pos, csv_path, node_info, edge_list,
num_networks=1000): # Random Geometric Network

       # if exists(csv_path): return

       resultStats = Stats()
       for _ in tqdm(range(num_networks), desc="Loading...", ascii=False, ncols=75):
           network = cls.rg_network(n, radius, pos, node_info, edge_list=edge_list)
           stats = Stats.from_directed_network(network)
           resultStats.add(stats)

       resultStats.avg(num_networks)
       resultStats.to_csv(csv_path, is_directed=True)


   @classmethod
   def rg_network(cls, n, radius, pos, node_info, edge_list):  # Return a single Random
Geometric DIRECTED network

       # Undirected graph
       G = cls._add_node_info(nx.random_geometric_graph(n, radius, pos=pos),
node_info=node_info)
       D = nx.DiGraph()

       # Add all nodes to Digraph
       for node in G.nodes(data=True):
           D.add_node(node[0], pos=node[1]['pos'], mag=node[1]['mag'], time=node[1]['time'])

       # Add connectivity to Digraph
       for node in G.nodes(data=True):
           for neighbor in G.neighbors(node[0]):
```

```
            if (node[0], neighbor) in edge_list:
                D.add_edge(node[0], neighbor)
    return D
```

This function first calls networkX's random geometric function and passes the three parameters described earlier. The output is an undirected network as expected. From here, we manually convert the network to a directed network by looking at the edge list of the earthquake network to determine direction. This edge list is created in the following way:

```
    @classmethod
    def get_edge_list(cls, G): # G must be a directed network
        edge_list = []
        for node in G.nodes(data=True):
            for neighbor in list(G.successors(node[0])):
                edge_list.append((node[1]['id'], G.nodes[neighbor]['id']))  # Means there is
an outgoing edge from node to its neighbor
        return edge_list
```

After that, we added magnitude information (from the dictionary mentioned earlier) to each node. This was possible because there existed a one-to-one mapping between node id's. After all this, we end up with a directed random geometric null model that contains magnitude information.

9.4 Approach for Detection of Motifs

To find the hub-spoke motifs within our graph, we simply divided up our nodes into high, medium, and low magnitude earthquakes. We then looked for patterns of hubs and spokes, and saved the data into a csv file, using the following code:

```
'''
   Method: find_hub_spokes
   Param: G: Graph of earthquakes
   Description: Find existence of hub-and-spoke structures in the graph
'''
@classmethod
def find_hub_spokes(cls, G, spokes, path):

   high_mag = [7, 8, 9, 10]
   med_mag = [4, 5, 6]
   low_mag = [0, 1, 2, 3]
```

We did the following for the high, med and low magnitudes:

```
   for node in G.nodes(data=True):
       # hubs = high magnitude nodes
       if round(node[1]['mag']) in mag:
           number_of_high_mag_nodes = number_of_high_mag_nodes + 1
           neighbor_list = []
```

```
            high_mag_nodes.append(node)
            neighbor_list = [nbr for nbr in G.neighbors(node[0])]
            count = 0
            for sub_node in neighbor_list:
                if count >= spokes:
                    number_of_high_mag_motifs = number_of_high_mag_motifs + 1
                    break
                mag = round(G.nodes[sub_node]['mag'])
                if mag not in high_mag:
                    count = count + 1


    #save data into df
    # magnitude earthquakes
    ProbabilityCalculations.hub_spoke_df.loc[0, ['Number of Hubs']] =
number_of_high_mag_nodes
    field = 'Number of Hubs with ' + str(spokes) + ' Spokes'
    ProbabilityCalculations.hub_spoke_df.loc[0, [field]] = number_of_high_mag_motifs
    field = 'Percentage of Hubs with ' + str(spokes) + ' Spokes'
    ProbabilityCalculations.hub_spoke_df.loc[0, ['Percentage of Hubs with Spokes']] =
ProbabilityCalculations.divide(number_of_high_mag_motifs, number_of_high_mag_nodes)



    #save dataframe
    ProbabilityCalculations.hub_spoke_df.to_csv(path, index=False)
```

10. Visualization

We wanted the visualization to be able to clearly show us some of the main patterns that could exist within our network. In order to do this, we bucketed nodes into “low”, “medium”, and “high” magnitude earthquakes. The low-magnitude earthquakes were coloured green, the medium-magnitude nodes were coloured orange, and red was chosen as the colour for the high-magnitude earthquakes. This gives us a clear visual distinction to analyze how earthquakes of varying degrees act. We also increased the size of nodes such that nodes with a higher out-degree (hubs) would be of larger size.

With these visualization techniques implemented, we can see, by zooming into specific communities, how earthquakes of varying magnitudes tend to act. Figure 10.0 shows a community of earthquakes that are heavily connected. We can see that the red high-magnitude earthquake is the biggest in the community, showing that it is a major hub. This coincides with our findings that high-magnitude earthquakes correlate to the existence of other earthquakes around it. The orange earthquakes within that area also tend to be bigger hubs than the low-magnitude green nodes. This visualization makes all these key findings very apparent.

Figure 10.1 shows a zoomed-out version of our visualization, presenting the fact that our data has very clear clusters and separated graphs.

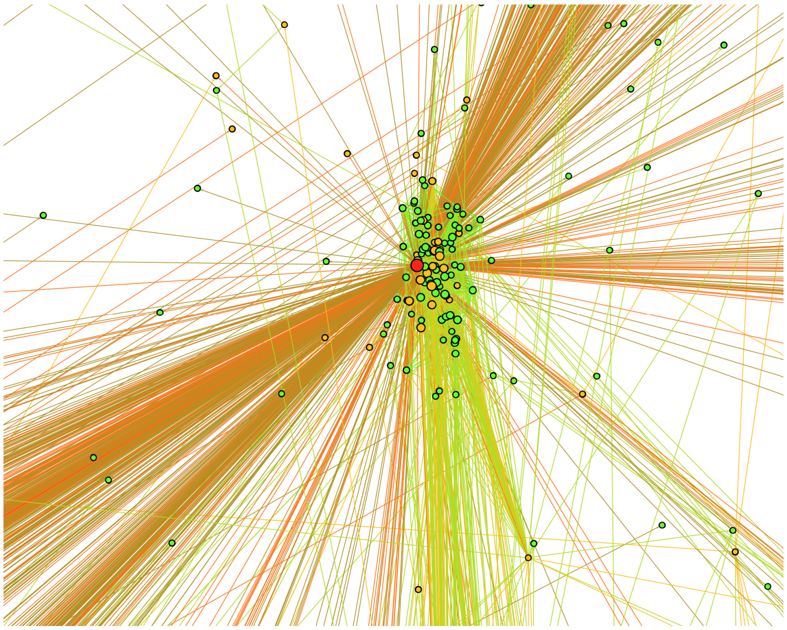


Figure 10.0

Figure 10.1

Intro:

According to Chakraborty et al. "Earthquakes are probably one of the most important natural phenomena affecting human life and property" . Through careful observation and data-collection, scientists have been able to take these occurrences and identify patterns to help us better understand our world. "Many scientists have proven that complex interactions and correlations exist in the seismic system in the past decades. The concept of earthquake network was introduced to seismology by Abe and Suzuki in 2004" (He et al.) Since earthquakes cannot be predicted this info is useful to help scientists generate models to get a better understanding of the next possible earthquake that could hit. Earthquakes alone come with a wealth of data, including magnitude, geographic location, and time. "Recently the time sequence of occurrence of different tremors and the positions of their epicenters in different earthquake catalogs have been studied using the tools of complex network theory." (Chakraborty et al.). This data, coupled with the technology which allows us to create, visualize, and analyze the network of earthquakes within an area not only provides us with a rich topic to study and understand real-world networks, but it also allows us to explore analyze and answer important questions regarding the nature of earthquake occurrences, as well as their predictability.

With this motivation, we created a network which allows us to explore the data of individual earthquakes, as well as the relationships between earthquakes. Through the analysis which will be outlined in this paper, we were able to identify patterns which exist in earthquake occurrences, based on their magnitude as well as their relative location. We also provided the beginning blueprints for future work on the topic such as earthquake probability calculations. This research paper will surely provide insight into the not-so-random nature of natural disasters, as well as exemplify a meaningful approach to data-network analysis.

References

Chakraborty, Mukherjee, G., & Manna, S. S. (2015). Weighted network analysis of earthquake seismic data. Physica A, 433, 336–343. https://doi.org/10.1016/j.physa.2015.03.037

He, Wang, L., Zhu, H., & Liu, Z. (2021). Statistical analysis of complex weighted networks for seismicity. Physica A, 563, 125468–. https://doi.org/10.1016/j.physa.2020.125468

Noora, S. (2019) Estimating the Probability of Earthquake Occurrence and Return Period Using Generalized Linear Models. *Journal of Geoscience and Environment Protection*, 7, 11-24. doi: 10.4236/gep.2019.79002.

Shebalin, P. (2006). Increased correlation range of seismicity before large events manifested by earthquake chains. *Tectonophysics*, *424*(3-4), 335–349. https://doi.org/10.1016/j.tecto.2006.03.040

Taylor, C. (2020). Probability of the Union of 3 or More Sets. *ThoughtCo*, thoughtco.com/probability-union-of-three-sets-more-3126263.

Appendix Table 1:

Average magnitude of neighboring level from the origin node

Origin Node Magnitude	Neighbor Level 1 Average	Neighbor Level 2 Average	Neighbor Level 3 Average	Neighbor Level 4 Average	Neighbor Level 5 Average
5.5	4.46	3.07	2.83	2.8	2.8
5.5	4.08	4.65	2.84	2.78	2.78
5.5	4.08	2.77	2.82	2.83	2.83
5.5	4.09	4.4	2.84	2.78	2.78
5.5	4.07	4.1	2.79	2.77	2.77
5.5	4.05	2.9	2.87	2.77	2.87
5.5	4.39	4.45	2.6	2.6	2.6
5.5	4.04	2.94	2.72	2.72	2.72
5.6	4.35	2.8	3.08	3.03	3.08
5.6	2.92	2.6	3.08	3.15	3.08
5.6	3.92	4.5	3	2.87	3
5.6	2.93	4.1	3.06	3.03	3.15
5.65	2.84	3.06	2.78	2.84	2.82
5.7	4.2	4.4	2.63	2.9	2.9
5.7	4.5	2.79	3.03	2.97	3.03
5.7	4.47	4.3	2.8	2.8	2.8
5.7	4.38	4.4	2.8	2.8	2.8
5.8	4.44	2.72	2.79	2.79	2.79
5.8	4.07	4.25	2.8	2.81	2.81
5.8	4.08	2.6	2.77	2.81	2.81
5.8	3.17	4.2	4.3	4.3	4.3
5.9	2.93	4.35	4.37	2.77	2.8
5.9	4.48	4.8	3.03	3.12	3.03

Average magnitude of neighboring level from the origin node

6	3.9	4.58	4.35	4.35	4.33
6	4.8	3.03	4.26	4.26	4.25
6.1	4.47	3	4.33	4.33	4.37
6.2	3.08	4.4	4.3	4.3	4.26
6.2	4.39	3.4	2.62	4.25	4.26
6.2	2.85	2.87	4.41	2.82	2.77
6.21	2.86	2.62	3.06	3.11	2.97
6.3	4.51	4.54	2.84	2.84	2.84
6.5	4.4	3.03	3.04	3.08	3.14
6.6	4.39	2.6	2.77	2.63	2.77
6.8	3.07	4.5	4.37	4.37	4.3
6.9	3.01	2.63	3.03	3.15	3.03
7.6	4.1	4.37	3.4	3	

Appendix A:

A Full view of out Earthquake Network