Earthquake Visualization Final Document

Team Only Visuals

Dataset Summary:

Our dataset, Mag6PlusEarthquakes_1900-2013.xlsx, is from the public tableau data website. This dataset includes all recorded global earthquakes with a magnitude of 6 or greater from 1900 - 2014. All attributes were derived from the United States Geological Survey (USGS). The 14 distinct attributes are time, date, latitude, longitude, depth, magnitude, magnitude type, nst, gap, dmin, rms, net, id, and place. However, we mainly used time, date, place, longitude, and latitude to build our visualization. Furthermore, we decided to filter the data based on decade, so we excluded earthquake events after 2009. As a result, we only included complete decades in our visualization. (dataset only includes earthquakes until 2014) The original dataset included 8314 rows (individual reported earthquakes), but we included 7632 rows from 1900 through 2009.

Since several earthquakes occurred in the sea and not necessarily on a landmass, we thought categorizing the longitude and latitude coordinates of land earthquakes along with oceanic earthquakes would be too inefficient. We first attempted to use python libraries (geopy and pycountry_convert), but that didn't work the best for categorizing points located in the ocean, which was half of the dataset, since the libraries were only for labeling continents. Instead, we defined ranges of longitude and latitude for each continent, including its surrounding water territories, and grouped all observations' coordinates based on these ranges. That was conducted using excel with custom formulas as well as R for classifying the points based on longitude and latitude. As a result, we created a new attribute: continent. With the continent classification, we broke down each grouping even further, dividing the earthquakes of each continent by the decade that they occurred. With this data, we were able to construct the line graph and its tooltip feature. Each line on the graph represents a continent, each point on the line represents a decade, and hovering on a point displays a tooltip with the top three locations with the highest earthquake frequency on that continent, in that decade.

Visuals

We used the dataset to render the points on the frequency map. For the system design, we mainly used D3.js, vanilla Javascript, HTML, and CSS. Within D3.js, we used the Geo Projection and Leaflet libraries to render the interactive map. To integrate both the line graph and map to be interactive together, we utilized the selection of various classes. We didn't need to host the data on a database, so it was sufficient to host the overall visualization on GitHub Pages (https://mmei-d.github.io/Earthquake-Visualization/).

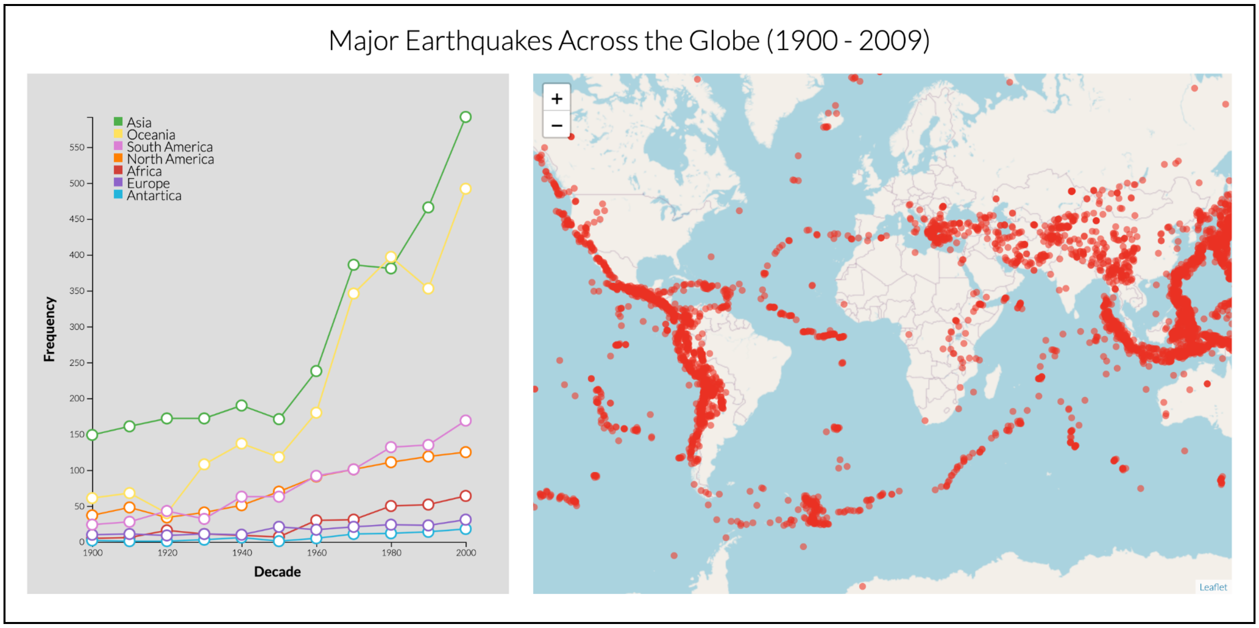

Photo 1. Default View: This shows what the visualization looks like upon first loading it. There is a static legend at the top left hand corner of the line graph. All line graph nodes are unselected.

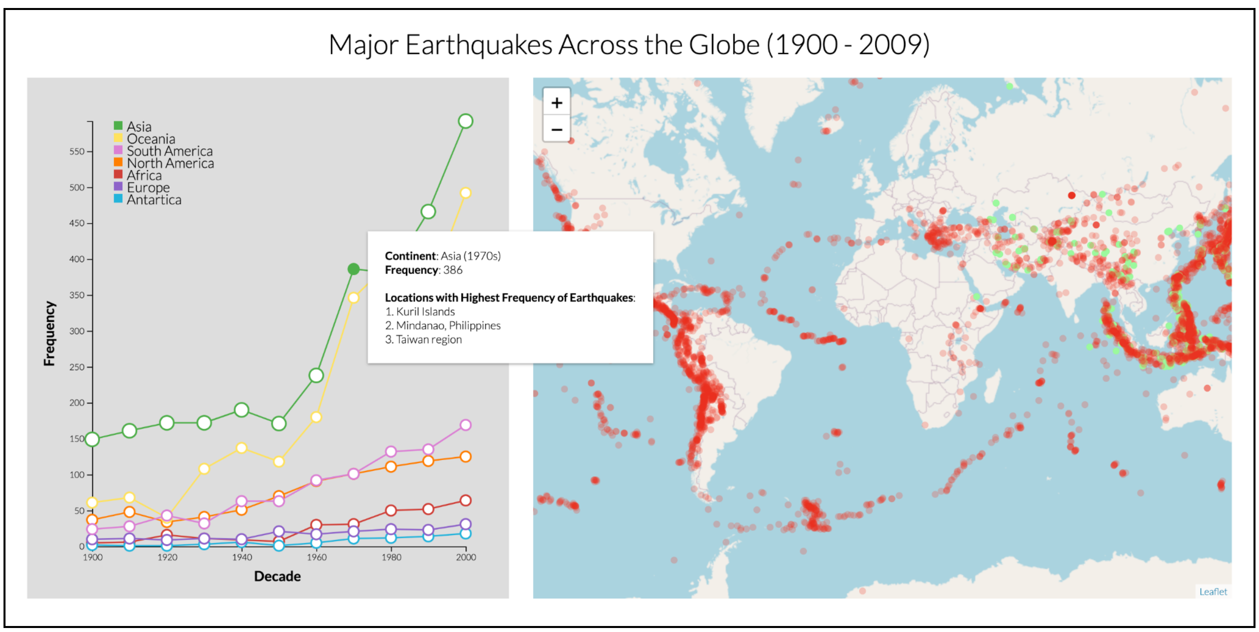

Photo 2. Selecting a Line Graph Node: When a node on the line graph is highlighted, it becomes filled in with the same color as the corresponding line. All nodes on that line also become larger to emphasize that continent. Each node on the line graph has a tooltip that appears when hovering, and it reveals more details about a decade's earthquakes. When a node is clicked, the map will change accordingly. All map points specific to that node's continent and decade become highlighted in neon green. All map points not

selected fade in opacity. All points on the map are red and have max opacity (like the default visualization) only when no nodes on the line graph are selected.

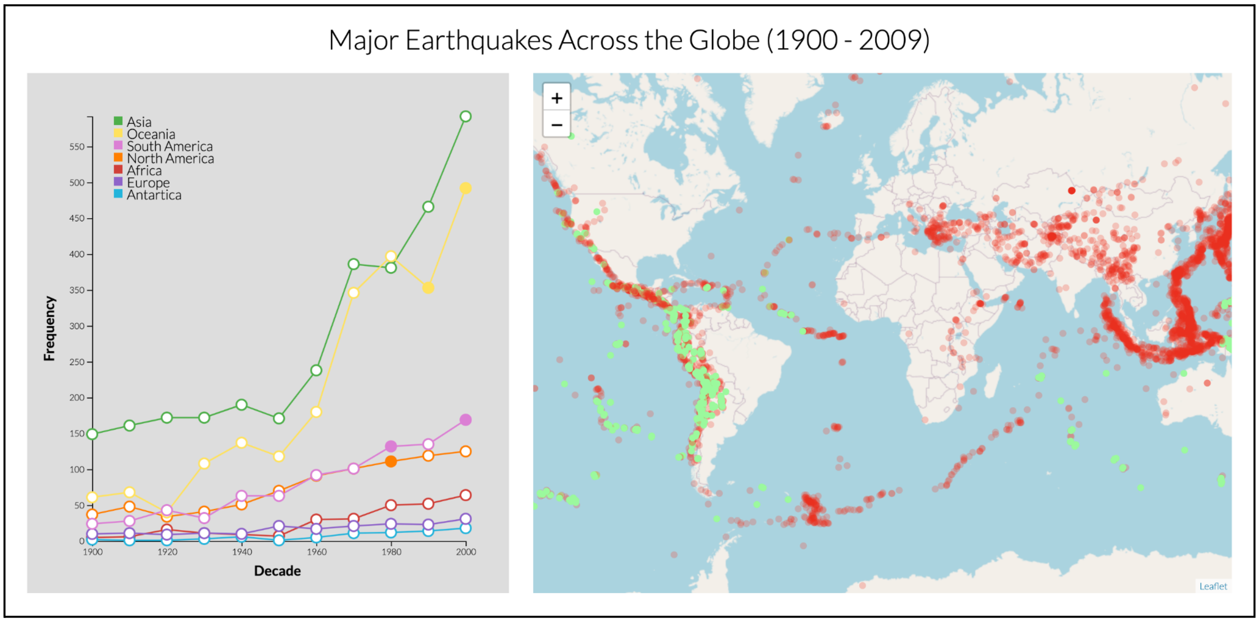

Photo 3. Selecting multiple Line Graph Nodes: Multiple line graph nodes can be selected across continents and decades, and all the points for those corresponding nodes will be highlighted on the map.

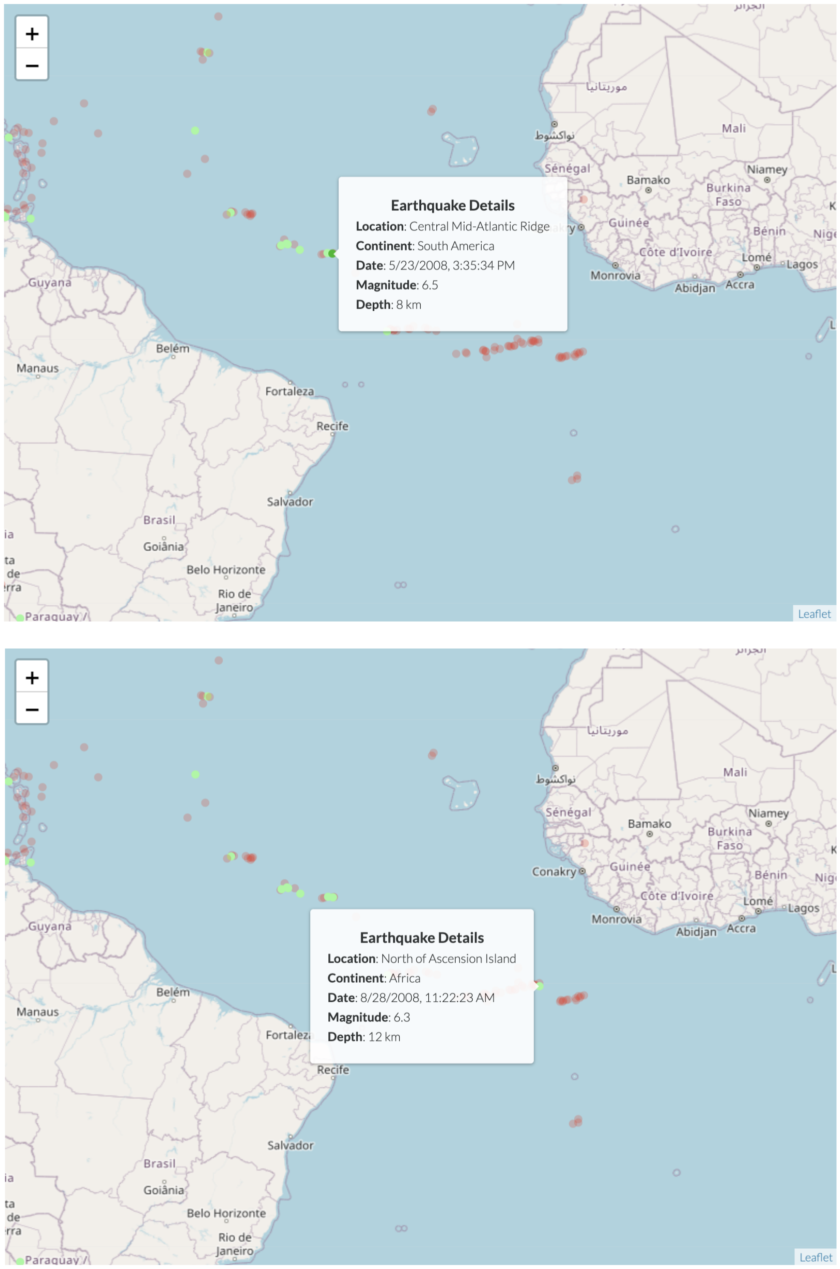

Photo 4. Map Zoom and Tooltip: The user and either use their mousepad or the buttons in the top left hand corner to zoom into the map and see the location of each earthquake more clearly. Hovering over a point on the map reveals a tooltip describing individual earthquake data. Hovering over a regular non-selected red point turns the point neon green. Hovering over a highlighted neon green point turns it dark green.

<u>Contributions</u>

- Planning and organizing: Everyone
- Data-wrangling: Chau, Emma, Mei

- Line Graph Code: Muhammad, Mei, Tye, Chau
- Map Code: Tye, Mei
- Line Graph Integration w/ Map: Muhammad, Mei, Tye
- Video Figure: Tye, Muhammad, Chau, Mei, Emma
- Document: Chau, Emma, Mei

Final video demo:
https://youtu.be/l_WDpUKonqk

Reflections

Planning the project and rendering a prototype was far easier than the implementation. We greatly underestimated the learning curve of D3, having spent most of our time researching libraries, previous implementation examples, and other trivial tasks.

Originally, our plan was to display frequency trends of earthquakes throughout the years through a mix between bubble and line plot stratified by continent. There would be a filter for each continent present on the plot as well as a hover feature in which the action would have a tooltip appear displaying information regarding specific frequency numbers and top countries for the continent. On top of this, there was a slider that would allow the user to view all trends throughout the century. Upon considering feedback from Professor Wall, we decided to include a map of the individual earthquakes, allowing users to pinpoint individual occurrences. From here, our design changed to interacting between the line plot and map, specifically by clicking on a node representing a given decade on the line plot, then those earthquakes within this selected decade would appear on the map. As we began implementing, we decided to also add a highlight feature where the continent hovered over on the line plot would highlight earthquakes in this region on the map. Constructing this visualization also came with some complications.

We managed to hit our first substantial roadblock at the very beginning of implementation when cleaning the dataset. There were some issues with creating a new attribute within our data, as our dataset provided longitude and latitude vs. identifying each continent or country each earthquake took place. The next roadblock regards javascript and D3. We have realized javascript is much more difficult than we anticipated. While we had smaller problems along the way with features such as the tooltip not displaying correct information or filtering the information, we managed to work through and brainstorm solutions. One of the biggest hurdles presented itself when trying to place interactivity between the two visualizations (nodes on the plot highlighted specific correlated earthquakes on the map). This took the longest time to work around; however, solutions were implemented.

In terms of implementation, we have learned there is much more to consider when designing products for a user. Empathy and product motivation is the center of designing interfaces for users, yet it is also important to practice empathy in considering the problems a developer might face when implementing a design. When things do not go smoothly, it is essential to communicate these issues or potential hesitations in order to work around and create effective solutions for them together. Due to time conflicts, we were not able to figure out how to filter the lines on the line graph so that one line at a time can be shown. So, for future directions, that would be a main priority since it would make selecting nodes much easier. Another idea would be to categorize the earthquake points by not only continents but also by oceans as well. That was something we couldn't do this time since it was difficult to find a python library that would convert latitude and longitude points to oceans. Our last idea we would love to implement in the future would be to highlight each continent's map points in different colors when various nodes are selected. That way, it's even easier for the user to differentiate between regions and time periods.

If there was anything that went smoothly, it would be the teamwork. All members were eager to help, listen to other members' ideas, and willing to compromise. Though our schedules were hectic, we managed to meet up in person this entire week, working long hours side-by-side to finish the assignment. The open mindedness and amiability of the team members made working on this extremely stressful project enjoyable.