Yelp Distribution Map Final Report

Abstract

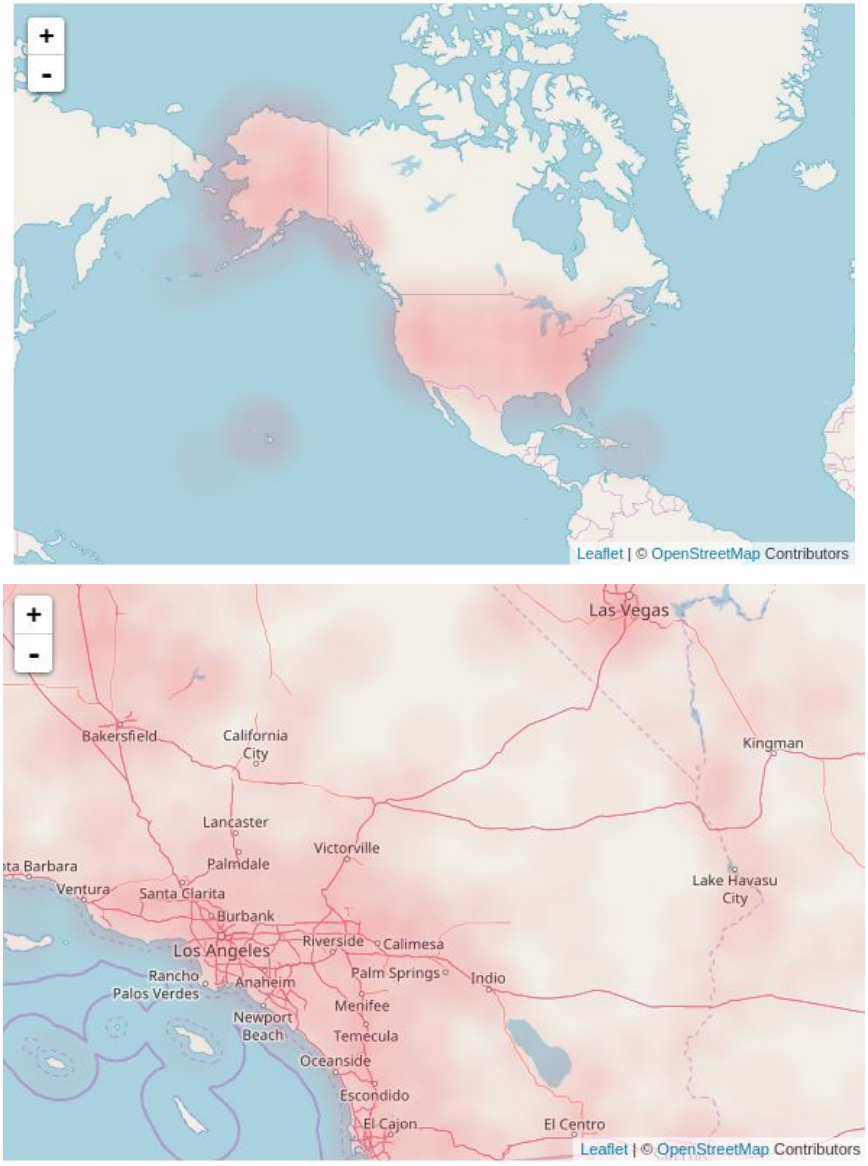
We were inspired to do a visualization of all the restaurants across the United States because we are both self proclaimed “foodies” and were interested in seeing their distribution. Henry has been planning a food trip to New Orleans, and Melissa has been planning a cross country road trip where she gets to try regional foods. We wanted this visualization to help us figure out which places in the US had the best food communities. Our target audience are people who are Yelp users, or anyone who is interested in the seeing the distribution of restaurants around the US. Our research question was how can we assist users in understanding the distribution of restaurants and their ratings across the United States. We hypothesized that both coasts would have higher densities than the middle of the US where the distribution would be more sparse. We also estimated that places with higher population, such as large cities would have higher numbers of reviews as well as higher average ratings. Upon further research we discovered that there already had existed a yelp heat map that was created about five years ago, but all we could find out about it was an online article and not even the actual site, so we decided to continue with our idea.

Yelp’s Fusion API was our main data source [3]. Given city and state name we were able to make get requests that provided us with information about all of the businesses in that region. The tricky aspect of this was not crossing the API limit of 5000 requests a day. Our work around that was simply to use a variety of emails to collect as many keys as possible in order to not be limited. In addition to using Yelp data, we retrieved the US city information from a corporation called Simple Maps Geographic Data Products [3], which is a trusted company that has been providing data and visualizations to many Fortune 500 companies and international organizations.

Visualization Design Evolution

Our initial idea for our project was a music visualizer. We planned to create a live as well as rendered visual to allow the user to better understand the physical representation of the sound that they generate, which would have used continuous data collection. We modified our project because we felt that the user would not leave with a big enough takeaway. We decided to go with a data heavy approach and visualize the distribution of both number of ratings as well as average rating across the country.

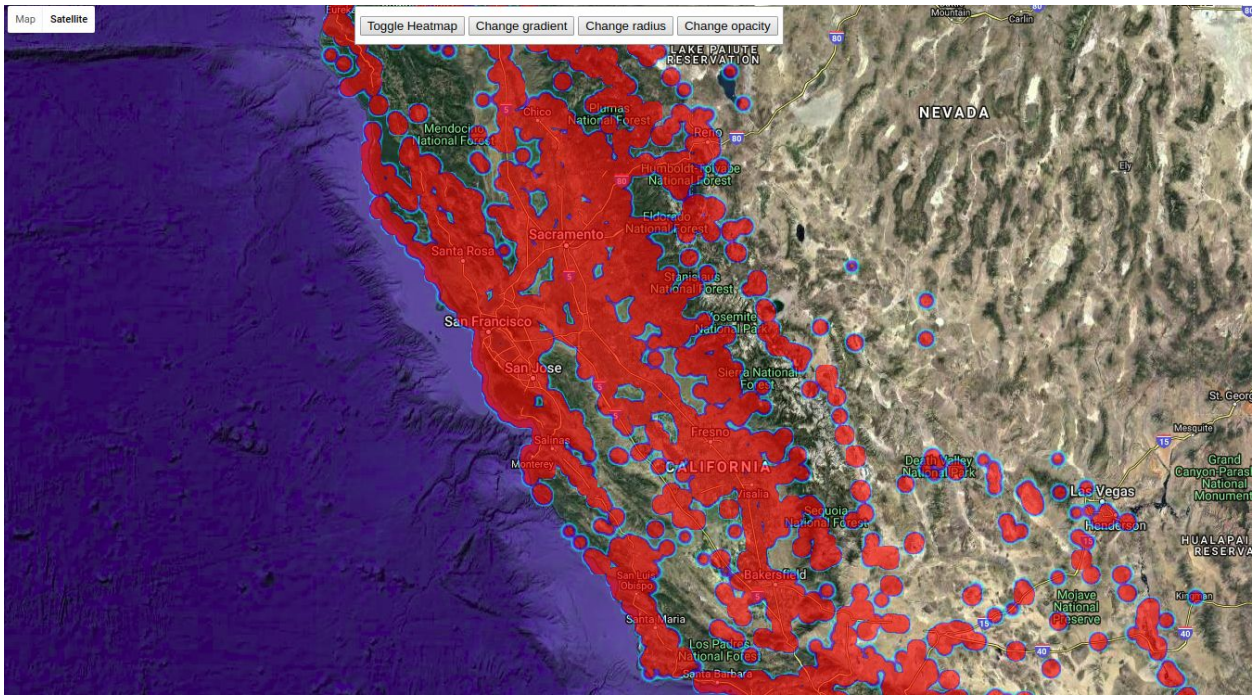
Our initial ideas centered around d3js, by finding a similar project or compatible library to adapt to our unique specifications. We explored the use of leaflet.js which ended in more difficulty that it was worth. The base implementation for a heatmap wasn't developed up to the point where it would be feasible for us to adapt it all the way to be compatible to our data.

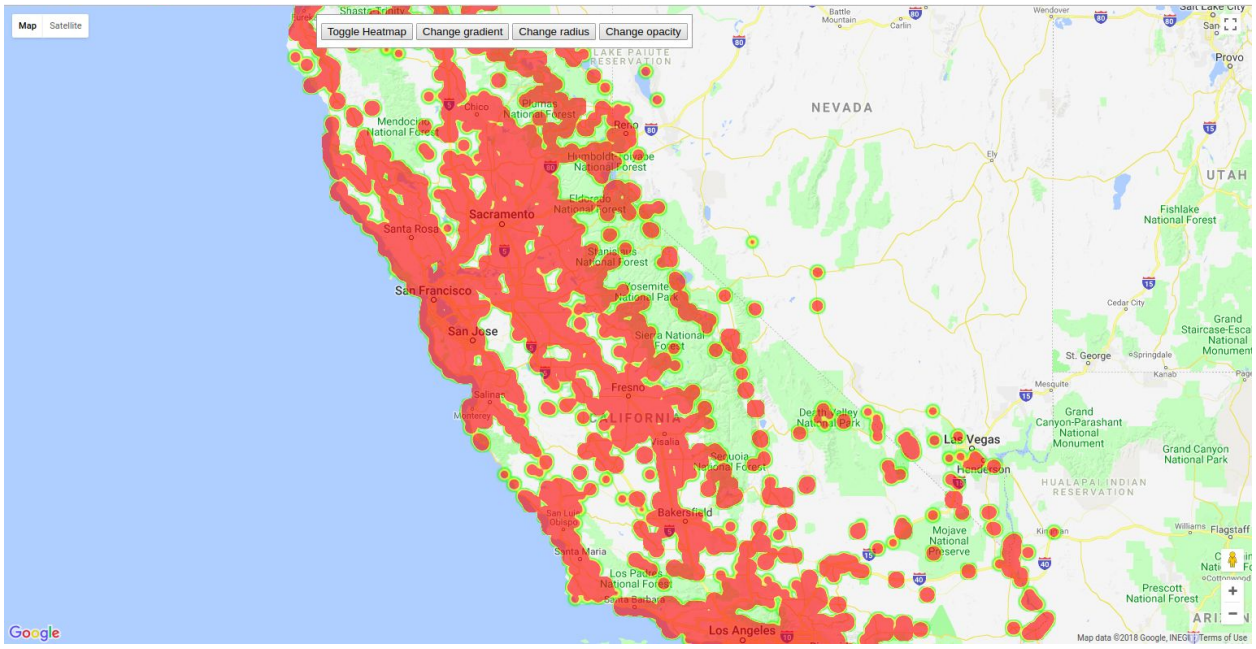


[Leaflet.js]

With this library we played around extensively with the color gradient functionality, and explored numerous solutions to get it to look more like the type of heat map an ordinary user would expect to see. In the end, we could not get the clarity to improve enough for us to begin experimenting with interactivity and sliders. We then moved on to the google

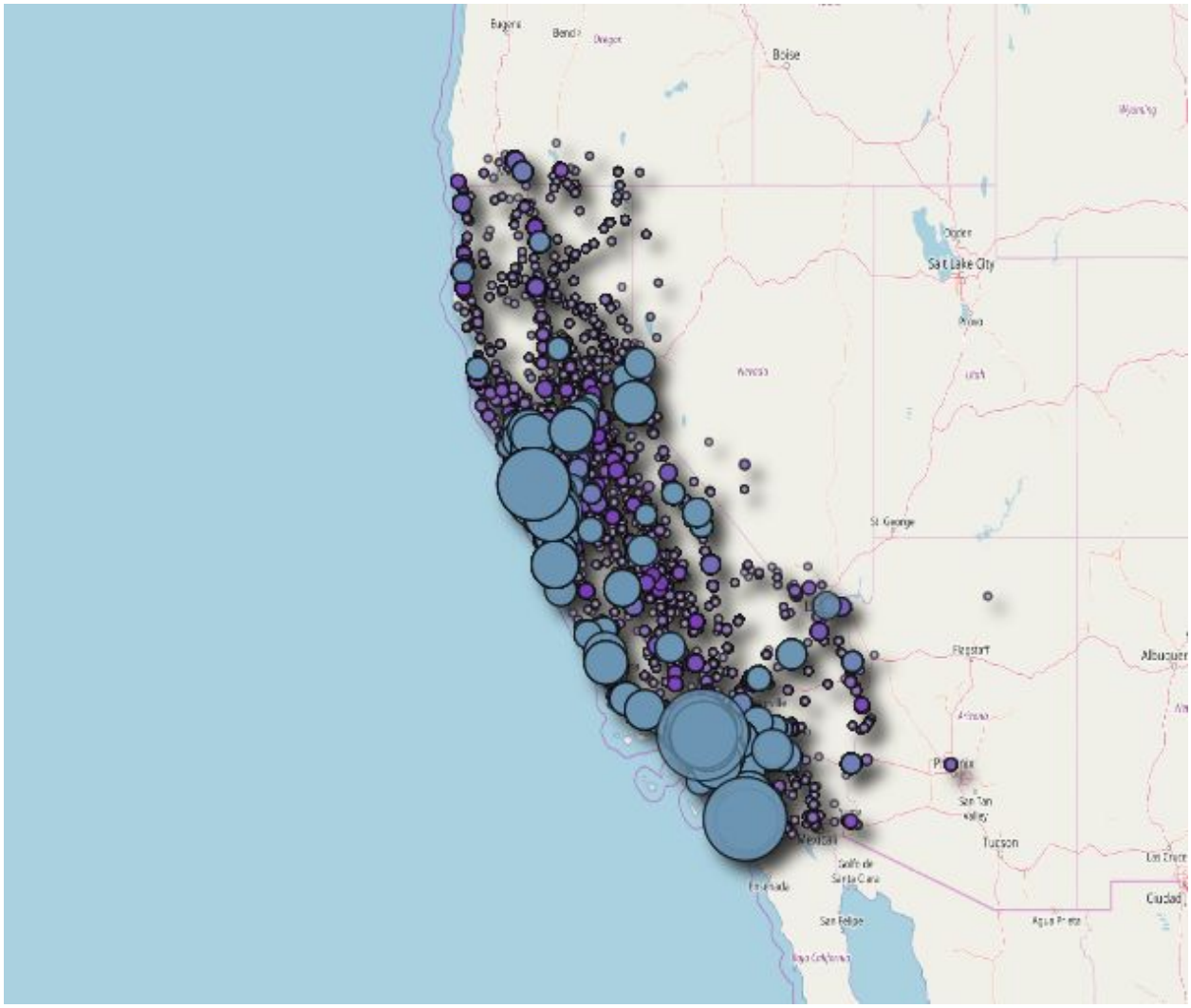
maps api, which allows for a layer on top of ordinary google maps to represent the heat map. This looked promising at first, so we added interactivity in the form of toggle buttons where users could modify the color gradient, radius of points, and opacity. This implementation ran well including during our pilot study, however when we were finally able to obtain all of our data, it struggled to keep up with the demand of over a million points. It still has potential to be a good visualization of our data, but it would have to be a state by state visualization which is little to compromise considering its level of customizability.





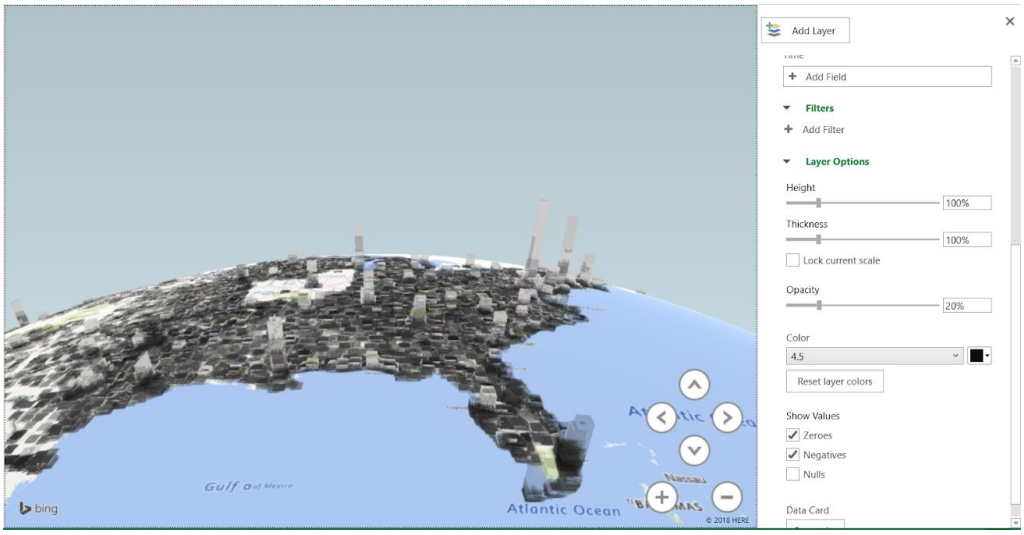
[Google Maps API]

The benefit of using the google maps api is that we can use both satellite and map layouts for our background, whereas in the other technologies we used we were limited to solely a standard map without satellite imaging. It is also easily customizable because it is run in a browser, so we were able to change the properties of the heat map layer. Our initial goal, however, was to show the country wide distribution, so we continued to search for tools that would help us complete our goal
We explored a lesser known tool called openheatmap. The documentation is rather lacking, but we ran it against our set of data and it came out more visually interesting than beneficial to a user's understanding. It is not customizable other than a few aesthetic options, and is rather confusing at a country wide scale. This is an example of the outcome using only data from california. The nature of the project overlays larger radii over smaller, so it makes data harder to understand when trying to analyze the distribution of ratings.

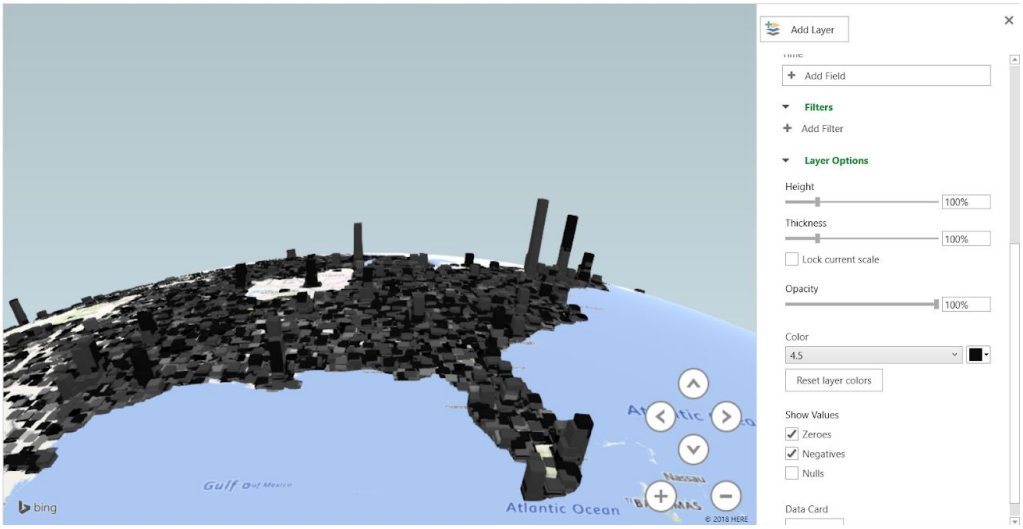


[openheatmap]

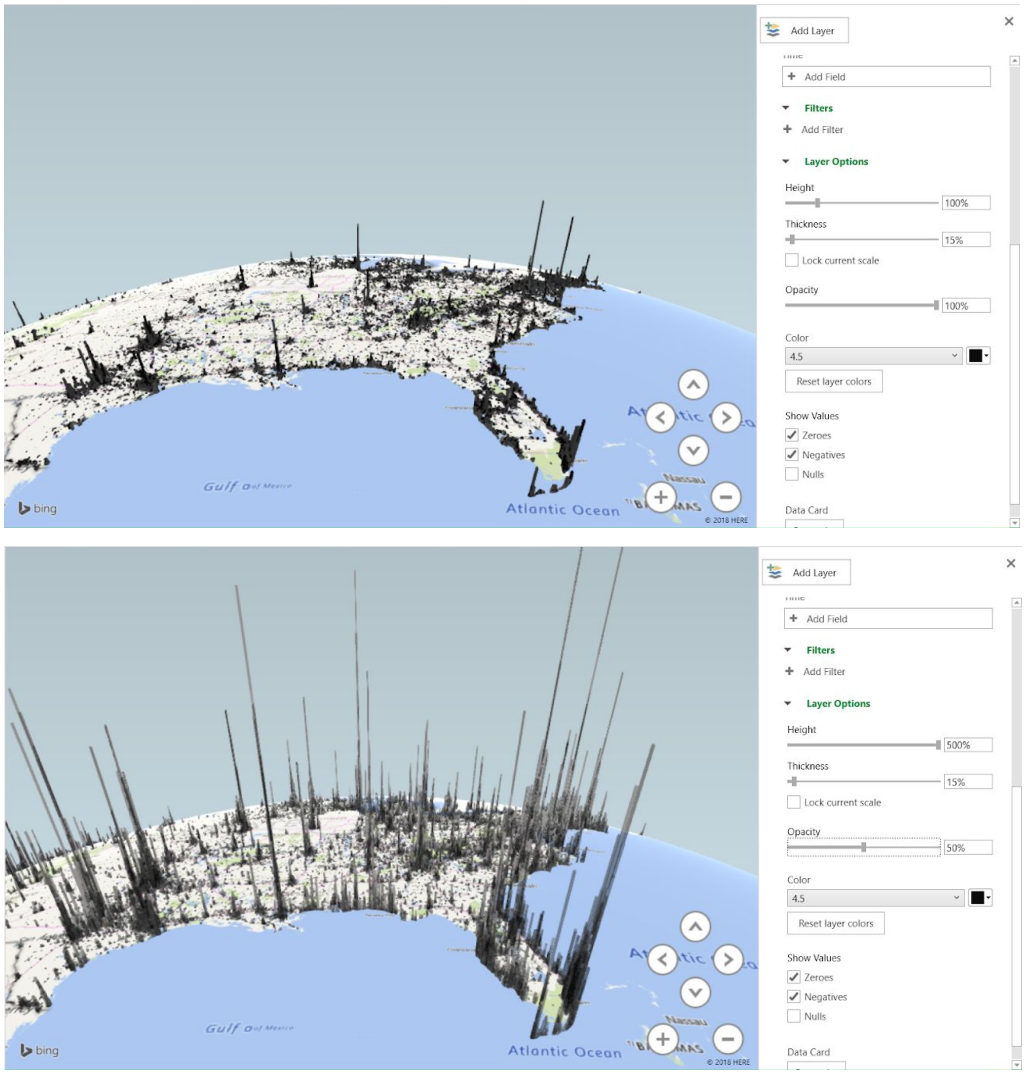
The lack of customizability of openheatmaps turned us away from this implementation. We then explored a microsoft technology called Powermaps. This is our final visualization evolution as it is able to render most of the country at a relatively usable speed. The reason why it cannot render the entire set of data is because we ran out of lines in the excel document that we set it up to read from. From our research, connecting it to a remote data source that could hold more lines of data than excel can would require purchasing a higher tier of microsoft office. With 90% of the data displayed, we believe that it provides an accurate base for the user to gain an understanding of the distribution of ratings over the country.



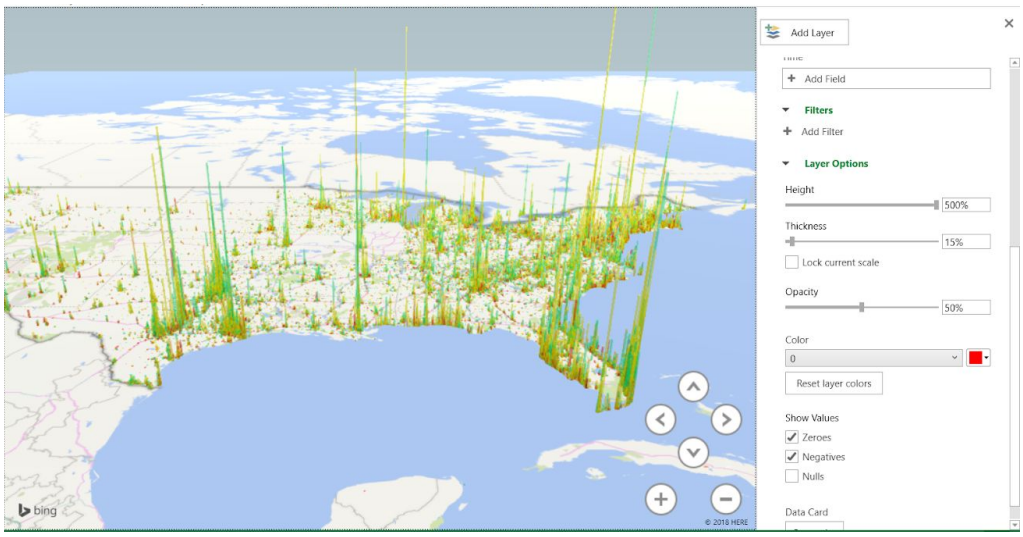
Here we have the height of each point being the number of ratings, and the color gradient corresponds to the average rating. This implementation allows the user to manipulate and interact with numerous properties of the data. Here we have set the color gradient to black and white, which sets a grayscale to each rating level 0-5 with intervals of 0.5. It is simple and easy to set the color gradient in general or specific colors for each rating level.



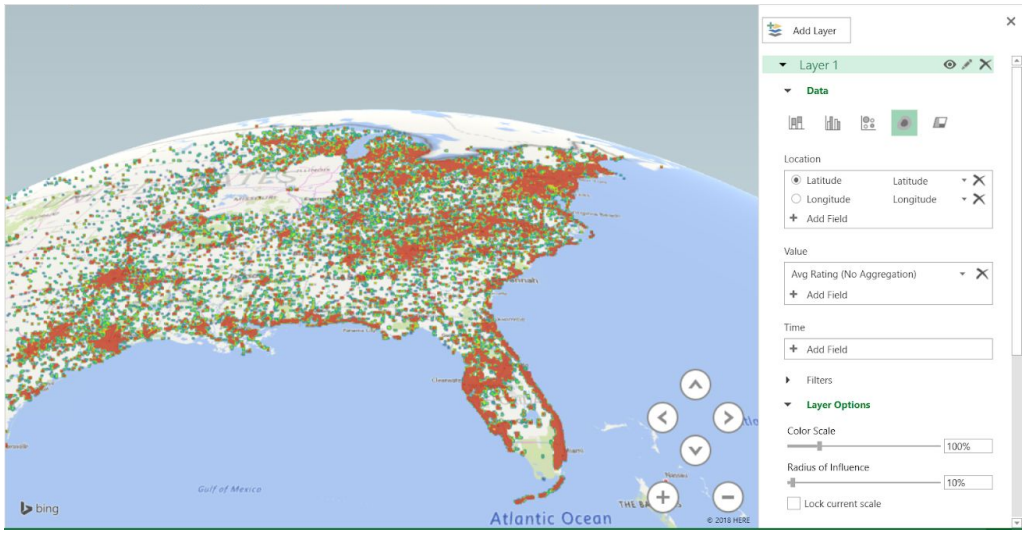
We can modify the opacity of the points, as well as the size, shape, and height of the polygon overlayed on top of each location. The height is described as a percentage in relation to the original value for rating.

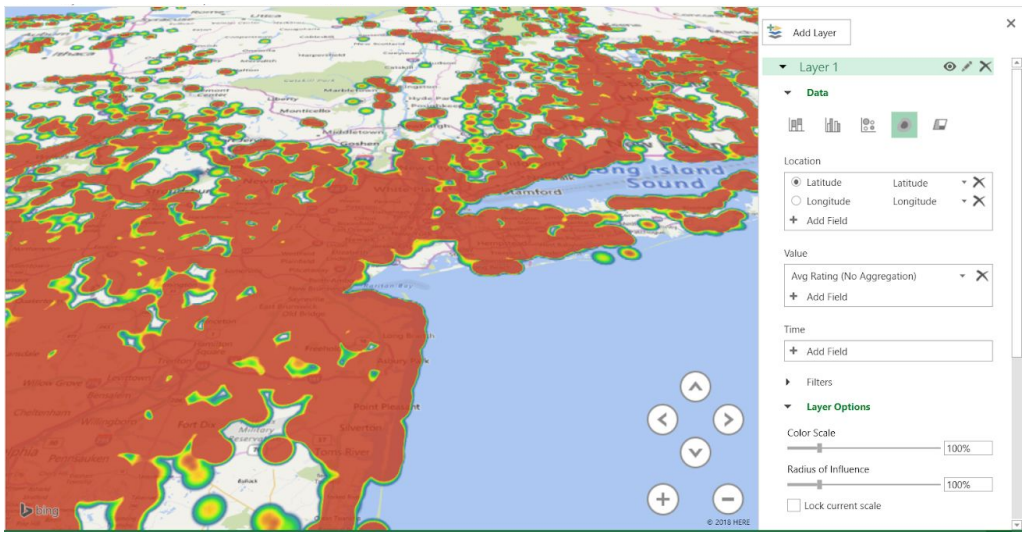


Another interesting feature is the ability to change the shape of the underlying map, whether a flat surface or a spherical globe. Below the shape has been changed to a flat surface, and the color has been modified to a green-yellow-red distribution, where green represents better average rating and red represents worse average rating.



This implementation also supports the typical heat map that we attempted to integrate into the leaflet library. The interactivity options on this implementation as well as speed of transition between types of visuals sets this implementation far ahead of our previous technologies.





[Microsoft Powermap]

Pilot Study Summary

Overall, we had somewhat similar feedback by each user during our pilot study. There were similar points of contention between our visualization and its users, even when we did not ask a question in that area. These popular issues were our first priority of fixing, and they consisted of points like adding a slider for radius and opacity that would replace our on/off toggle button, as well as adding a relationship between radius of points and zoom level. There were other interesting suggestions that were given that we have not had the chance of implementing yet. One point is how to distinguish between a city with a lot of bad ratings and a small town with a few very good ratings. While this is not be a very likely situation given the clarity of our overall visualizations, there exists the possibility that the frequency of poor ratings will overpower the small town, or that the small town will make the entire city appear to have better reviews than it actually does. The zoom level is a large factor in this issue as well as others, and it still presents a problem in our first visualization but has been solved within others. Other points of feedback we received was the priority of number of ratings versus average rating. While average rating might be more valuable to the visualization, number of reviews was easier to obtain and therefore we collected that first. We have since collected both and have incorporated them into our newer visualization. We got positive feedback on whether we should create a legend on the visualization, and suggestions ranged from a simple color definition to a complex adaptive legend that defined color based on zoom level, because the areas of color does change with zoom. What we ended up doing was creating a simple legend within our more recent visualization because the zoom level

has been accounted for and it doesnt change the area consumed by the color too much. Lastly, another interested suggestion was a hover functionality that defined an area on the map that was constrained by zoom. We implemented a clickable functionality where users can click on a particular point and it will show the location as well as the name of the institution. We were informed from our questions that our color options should suffice, and that always including town/city names would create unneeded clutter. We have the option of doing this anywhere however, because in an area familiar to the user this information could help them become more familiar with the data by making connections to personal experiences in this place. Overall participants of the pilot study responded to our first question quite positively and said that they thought that with our current level of interactivity they gained enough of a perspective of what we were aiming to display. One issue that we outlined to address in our user study is if our users actually gain a valuable takeaway or if they simply claimed that they gained a valuable takeaway after briefly looking at a few views of our visualization.

During our pilot study, we interacted with people of similar interests but different background. They line up well with our targeted audience, however this spread is not very wide. Our visualization is aimed towards anyone and all of experience are welcome. A more comprehensive list of participants will be asked to test our visualization in our user study, however the participants from our pilot study do not exceed the boundaries of our desired users. Our visualization changed dramatically as a result of the pilot study. Minor changes were mentioned in relation to our original visualization tested in the pilot study, but we also changed technologies after this feedback in an attempt to display our tremendous amount of data as clearly and intuitively as possible. The feedback we received is what motivated us to change our technology and hunt for the clearest visualization, as described in the Visualization Design Evolution section.

Beyond this course project, potential evolutions to our visualization could include integrating it into either Yelp's page as well as any food service, such as GrubHub. Motivations for this are that Yelp can gain ad traffic if there is a unique and exciting way to browse for popular food, and GrubHub could gain more customers if the interface was more enjoyable to explore with. Other than commercial extensions, simple additions that we could easily explore would be collecting world data and including this in our visualization. Even farther than this, we could add smaller towns to our data because as it stands we only include more prominent cities around the country. This would truly be a feat, however, because the amount of data that this would entail would stress the scalability of our implementation.

Our end result is based on microsoft's built in technology that draws from excel as it's data source.

Technical Implementation:

The core contribution to this visualization was the gathering of all of the data into one all encompassing file. This data took almost a full week to collect because of the API request limit constraints as well as cleaning the data. We wrote multiple python scripts that extracted and cleaned the data and exported it all into one CSV file that could then be used for multiple types of visualization. One of the main challenges after obtaining the data was finding a way to visualize it all without our visualization crashing. We made multiple attempts as described in our visualization evolution section. Our preliminary research included exploring previous projects such as that conducted at Berkeley [8] to be used for business insights, however their implementation was single location markers and their analysis was competition based. The initial route we took was to implement our heatmap with d3js following a recent experiment done in 2014 [9], so we explored the leaflet.js library. This library was difficult to manipulate and the changes we made to adapt it to our data resulted in a very unappealing visualization that contained little information that the user would find valuable. We experimented with scaling of of the rating value and a corresponding color gradient that flowed with our scale, however we could never find a happy relationship with the radius sizing as the user zooms in and out. We explored a solution we found called openheatmap that consisted of uploading a csv file The next attempt was to use Google Maps Heatmap JavaScript library [4]. We created an HTML file that used JavaScript to dynamically allocate the points on the Google Map. In order to dynamically allocate the points as opposed to hard coding them, we needed to run a local server and use an XMLHTTPRequest to get access to the file. Then we used a tool called Papaparse [5], which is a powerful csv parser, to extract the data and generate the points. This process worked really well at first, we could use toggle buttons, change the gradient and background, but as our data grew it got slower and slower. Google’s API couldn’t handle the million plus points that we were throwing at it. We turned to a technology called Microsoft Powermap. which involved little technical implementation, however it could render 90% of our data set while maintaining interactivity. This tool provides the greatest opportunity for users to gain an understanding of our data.

Who Did What?

Melissa took the lead on data collection, and Henry worked on adapting visuals to handle our data correctly as well as adding interactions within visualization tools that supported it.

References

[1] Warden, Pete. “Petewarden/Openheatmap.” *GitHub*, 9 June 2015, github.com/petewarden/openheatmap/wiki.

[2] Yelp. “Yelp/Yelp-Fusion.” *GitHub*, github.com/Yelp/yelp-fusion.

[3] “United States Cities Database.” *Simplemaps*, simplemaps.com/data/us-cities.

[4] Google, Google, developers.google.com/maps/documentation/javascript/examples/layer-heatmap.

[5] “Papa Parse.” *Papa Parse - Powerful CSV Parser for JavaScript*, www.papaparse.com/.

[6] “Leaflet.” Leaflet.heat, http://leafletjs.com/

[7] “Powermap.” Excel, 2016.

[8] “Yelp Data Visualization”, http://people.ischool.berkeley.edu/~sayantan.satpati/yelp/#portfolioModal3

[9] Raju, Akhila, Basnage, Cecile, and Yin, Jimmy. “Visualizing Yelp Ratings: Interactive Analysis and Comparison of Businesses.” http://vis.berkeley.edu/courses/cs294-10-fa14/wiki/images/f/f4/Datavis.pdf

[10] Babicki, Sasha, et al. “Heatmapper: Web-Enabled Heat Mapping for All.” *Nucleic Acids Research*, Oxford University Press, 8 July 2016, www.ncbi.nlm.nih.gov/pmc/articles/PMC4987948/.