Charles Yu
Michelle Yang
Jonathan Lee
INFO 4310 - HW4 Final Report

Describe the dataset you chose (including any potential weaknesses or post-processing necessary) and identify specific needs of consumers who are exploring that data to make a more informed decision. Your needs should match up with the affordances and visual metaphors you later implement.

Of the datasets made available to us, our team chose the dataset of Yelp restaurants in the Boston area. In our initial data exploration, we were deciding between the Yelp datasets for Pittsburgh and Boston but ultimately went with Boston because we thought it had greater variety of restaurant categories. In terms of the weaknesses and necessary post-processing of the data, we originally found that there were a number of restaurants that were not actually in Boston, but right outside of Boston. These points appeared to be outside of our map which we decided to filter out to be accurate that the points only cover Boston. This moved our data points from 343 to 245. We then also filtered out duplicate entries based not only on name but also latitude and longitude because we realized chains may share the same name. This further decreased our data points to a count of 212.

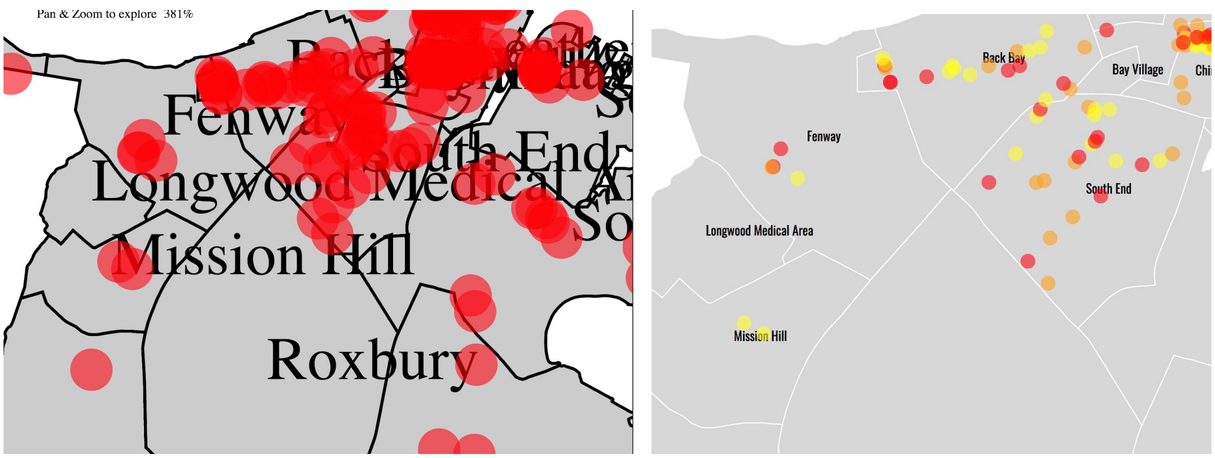
We identified two primary user types that would utilize the visualization tool we built. The first type of user is someone who is not familiar with the area (i.e. a tourist or someone traveling/moving to the area for the first time) who is in need of discovering places to eat either near their current location or a location they plan to be at (if they plan to visit a tourist spot for example). The second type of user is someone who is already familiar with the Boston area, but also is interested in finding a place to eat near their location.

These users have similarities and differences in their needs which lead us to the affordances and visual metaphors we implement to address both sets of user needs. Both users have the common goal of finding an ideal place to eat in the Boston area. In order to accomplish this, both users will need to be able to indicate some of their preferences given that the definition of “ideal” is subjective. As such, we afford the user the ability to filter their options based on attributes the dataset allow (star rating, restaurant type, and location).

Where the users differ is in their familiarity of the area. The first user who does not know the area well will need to be able to explore the area more than that of the familiar user. We afford this by providing a map view for the user to navigate around, panning in and out to better pinpoint where he/she wishes to explore. The second user, who is more familiar with the area, would already know the names of different neighborhoods/locations. To utilize this additional knowledge by this user, we include the ability to filter by neighborhood as they would be able to better know which neighborhood corresponds to what area, thus saving them time during the narrowing phase.

Identify potential challenges for users in seeing or interacting with the data. For instance, high densities of points in certain geographic areas might make it challenging for users to find individual points on a static map.

There are several potential challenges for users in seeing or interacting with the data, especially given the large number of data points. One such challenge is that users would run into the issue of navigating areas of high quantity and density of points. This leads to information overload and makes it harder to click/hover over specific points. We address this issue in 2 ways. The first thing we implemented was to set a constant size for data points that is unaffected by d3.zoom transformations (by default d3 enlarges every element by a constant scaling factor as the user zooms in), This made it so zooming in wouldn't make the data points larger and helps solve much of the overlapping issue that comes with high density. On top of that, users can then always be able to zoom to a point in which there are little to no overlaps of data points and everything becomes easily selectable.



Before corrective scaling implementation → After corrective scaling implementation

The second thing we implemented was the location filter. This helps reduce the number of data points showing which decreases the chance for high density overlaps.

Another challenge users might face when interacting with the data is the user need of being able to make comparisons when exploring available restaurants. After all, people usually want to find the best place to eat based on their needs and people's memory can only hold so much information at a time. We address this issue by implementing a pinning feature where you can pin the restaurants you might consider going to to make easier comparisons.

Outline the interaction affordances you are implementing to help support the data exploration process. Link the affordances to particular user needs and explain how the interactions will help in the exploration. Describe how your interactions help overcome the challenges you identify for *data explorers.* (already mentioned in other sections -- may revisit to clarify)

Thus, zooming in/out, panning, the size scaling of data points, filtering systems, and pinning features all afford the users different ways to help support their data exploration process for the reasons already described above. To reiterate, all of the features offer affordances that can help both user types with specific features helping more than others depending on the user type (also already earlier described above). For example, the user with that is more familiar with the area will benefit more from the filtering by area functionality given they will already have mental models of what neighborhood name corresponds to what area.

Briefly describe your final interactive visualization application.

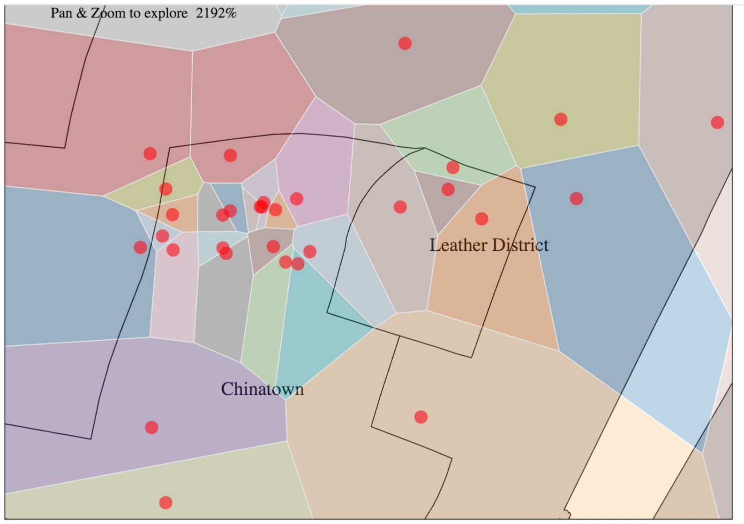
Briefly describing our final interactive visualization application, we have a map of Boston fully populated with restaurants as different colored points. The different colors reflect each restaurants star rating. Hovering over points will bring up a tooltip of information of the restaurant. Users can navigate the map by panning/dragging around and zooming in and out which scales the points and makes them easier to click as you focus your search. They can also narrow their search by using the filtering tools provided. Finally, users can pin restaurants of interest to make easier comparisons later on and lead to more efficient decision making.

Step back and think about issues or trade-offs associated with the interactions you developed, and how you might alleviate those (or whether they are unavoidable). Because you aren't creating an exhaustive data exploration system, this is your chance to talk about what you would do if you had more time to develop the system.

Briefly outline the development process of your tool. Explain how your visualization/interactions changed between design and final implementation. Comment on any trade-offs or design choices you had to make while developing.

Our development process and design decisions evolved over time. Our team began by doing some brief market research by exploring different websites and apps that existed that solved similar problems (Yelp, GrubHub, Foursquare, etc). From there, we were able to note some key (and obvious) functionality we would need including a way to narrow a search as well as displaying data points. We began our development process imagining the final visualization to be a collection of data points for the restaurants with that ability for users to select them to view more information. However, even with such a seemingly straightforward idea, our design went through many iterations as the development process progressed. With each iteration, design decisions were made with trade-offs considered as will be described below.

First, we went through a number of explorations in how we wanted to display the map and the data points. When we saw that plotting the points as is on the map was very overwhelming, we played around with different ways to organize the data points in a more digestible, interactable way. One such experimentation was implementing what's called a voronoi tessellation where each point would produce a geometric area around it to increase the clickability of small points close to each other.

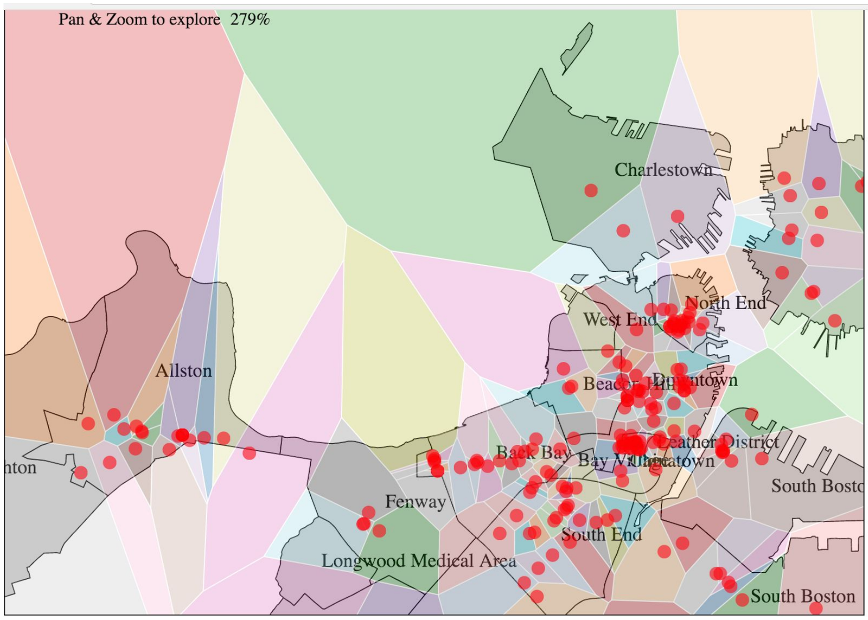


Voronoi tessellation implementation

This works great for points that are really close to each other by increasing the clickable surface area. However, we later decided to scrap this implementation because it became very excessive and overwhelming when points were not super close to each other, especially those that were on the border radius (see image below). Furthermore, the tessellations were visually overwhelming and we did not think the areas defined by them would be that intuitive to most users.

Instead, we went with the scaling solution described earlier as it removes the overwhelming factor and is able to remove overlaps when the user zooms in enough. The tradeoff of this final design, though, is that it does require the user to zoom in at times to address the overlapping issue. We tried to alleviate this at one point by implementing a “jiggle” button that randomly shifts some of the points very slightly to reveal points that were overlapped. However, we found that this was not very effective and was probably also not very intuitive.

One possible way to alleviate the overlapping issue moving forward is to produce a function that makes sure data points do not overlap (or only overlap partially) by detecting how much they touch and setting a threshold on that. That too has its own trade-off of what happens when there are just too many overlapping points and things just spread out too much (though we predict this to rarely ever be the case). Had we more time, we would have explored this solution further.



Voronoi tessellation implementation becomes overwhelming

One other large change between initial design to final implementation was the searching functionality. Originally, we imagined we would implement a search bar in which users could type in keywords such as the partial name, location/neighborhood, or types of food they were craving. However, after experimenting with that, we realized that the dataset was fairly limited in what could be parsed for the search bar implementation. Additionally, searching requires the user to come in with a keyword in mind, and one can only search one keyword at a time. This required us to think of alternative ways for users to narrow down their search while considering our limited data fields. Thus, our search bar eventually evolved into a filtering system in which users can filter the points based on star rating, type of food (cuisine), and location based on the data available. By providing what the user can input, we face the tradeoff of giving a user the most relevant search result, but ultimately we avoid the issue of a user inputting a search inquiry that simply cannot be matched to any of the limited data fields and inputs we had.

With filtering came a lot of design decisions we had to make as well. For one, how would we visually express star ratings? Initially we had the idea to make the data points numbers based on the star rating (1, 2, 3, 4, 5) but we quickly discarded this idea because it was ambiguous, overwhelming, and made the points even less clickable. Instead we decided to express star rating with color. This is much easier for the brain to process and didn't reduce the clickability of the points as did the numbered method. There does exist a tradeoff of this color-coding solution, though, which is how to let the user know what each color means. However, we alleviate this problem by providing a legend and use colors that follow a gradient where yellow is low ratings, orange medium, and red high.

Another design decision around filtering was whether a user would be able to select multiple location and cuisine filters. We ultimately decided to restrict the user to only picking one neighborhood/location because we predict the user's use case to be searching within the specific area they are currently at or will be rather than the other way around. Rarely do we find people choosing a far away place to go based solely on food when they are exploring generally. Instead, it is much more common for users to explore food options near their current location or future location. On the other hand, we did not limit the number of cuisine options the user could filter for because users exploring food options are likely to be more open to different types of food and are not always craving just one specific type of cuisine.

Lastly, the default state of our visualization changed throughout the development process as well. Initially, we imagined the map to not be fully populated upon opening the webpage. We thought this was best at the time because it reduced the information overload that comes with fully populating the map due to the high quantity and density of points. However, we later decided to populate the map from the start to encourage user interaction rather than leaving the user confused of what the point of the visualization was for. It does this by reducing the barrier of action by not needing the user to enter their filter options first to interact right away.

Identify how work was broken down in the group and explain each group member's contributions to the project. Give a rough breakdown of how much time you spent developing and which parts of the project took the most time.

Work Breakdown:

- Charles
    - Contributed in early market research & data exploration
    - Experimented with certain functionalities
    - Built final visualization functionality
    - Ideated tool features and functions
    - Analyzed tradeoffs and influenced design decisions throughout development process
- Michelle
    - Contributed in early market research
    - Ideated tool features and functions
    - Analyzed tradeoffs and influenced design decisions throughout development process
    - Stylized the UI, features, and filters with CSS/HTML/d3
- Jonathan
    - Contributed in early market research
    - Produced early mocks and idea generation
    - Analyzed tradeoffs and influenced design decisions throughout development process
    - Completed the full final report