Visualization of the Yelp Open Dataset

Group 24

Eric Robert Timothy Huber Md Sadik Hasan Khan Benno Kossatz
Andreas William Randrup Madsen

February 27, 2025

1 Introduction

Data visualization permits us to cross the divide between raw data and understanding, allowing us to tell stories and uncover connections, patterns or causes obscured in the individual data points. Through applying concepts and theory from our "Data Visualization" class, we were able to create a strong visualization capable of aiding business owners to make more informed decisions about their opening hours, competition and location while conforming to the field's best practices. In order to support this goal, our group decided to explore the "Yelp Dataset" helpfully provided by Yelp. Ultimately, we chose this dataset after deliberating and presenting 8 other datasets, in part due to the fact that it was easy to access and extensive. The data was accessible online [2] via a direct download, meaning we were able to avoid the extra work of data scraping or API rate limitations. The breadth of the data allowed us to formulate several questions and explorations before honing in on our eventual task(s). With our new visualization, restaurant owners in Philadelphia, Tucson and Tampa will be able to gain an overview of key information required to start a business.

2 Data and Task Abstraction

The Yelp dataset is a selection of Yelp's businesses, reviews, and user data made available for academic use. It contains approximately nine million individual reviews from over 150.000 different businesses totaling almost 16 gigabytes of (uncompressed) data. This is split into six separate JSON files. These contain business, review, user, checkin, tip and photo data. The ones that are the most relevant to us are the business and review data which are stored in the `business.json` and `review.json` data respectively.

`business.json`

This contains business data including location data, attributes and categories, totaling roughly 900 megabytes. These include 14 categories along with various subcategories. For the sake of brevity, we will outline the attributes that are most relevant to our visualization.

- `business_id`: A unique 22-character string that functions as a key to identify business and connect them to the other data files including `review.json`.
- `name`: The name of the business as a string.
- `address`: The address of the business as a string.
- `city`: The city where the business is located as a string.
- `postal_code`: The postal code of the business.
- `state`: The state where the business is located as a string.
- `latitude`: A float giving the latitude of the business.
- `longitude`: A float giving the longitude of the business.
- `stars`: A float star rating, rounded to half-stars.
- `categories`: An array of strings for business categories.
- `hours`: An object containing the opening hours from Sunday to Saturday using a 24-hour clock.

`reviews.json`

This contains full review text and star review data for specific businesses including the `user_id` that wrote the review, the `business_id` the review is written for and some sentiment information. This file is roughly five gigabytes with nine categories. Again, the relevant ones to us are

- `review_id`: A unique 22-character string used to identify reviews.
- `business_id`: A unique 22-character string used to map to businesses in `business.json`.
- `stars`: An integer star rating.
- `date`: A date string formatted as `YYYY-MM-DD`.

This visualization is made to support an overview task and is aimed at restaurant businesses allowing them to gain an overview of temporal trends in ratings and how opening times affect ratings. To further support this, various filtering and navigation methods are implemented, allowing the user to gain further view for specific restaurant types and locations. Our visualization is designed to be viewed on a computer screen by a single user.

3 Related Work and Existing Solutions

The primary existing solution is naturally the program we drew the information from: Yelp. A business owner could undertake this process themselves by searching for his restaurant type and clicking through individual restaurants to get an idea. However, this is less efficient than our solution and misses out on some of the temporal trends that may or may not be present in this user's search. This would, however, potentially be more efficient if the task is to compare their business directly to one of their competitors. While our visualization does support this functionality, by allowing users to select single data points, it is more cumbersome for the user to locate a single point geospatially, rather than by searching in by name. However, since our visualization task is an overview rather than a look up or comparison, this isn't within our scope.

Other more bespoke solutions exist as well, like Chatmeter [1]. This is a tool to help aggregate reviews, survey results and other internal data and run analysis on them using their AI model. This analysis results in some visualization and other statistical factors, some of which target review score trends and how to increase them. These, however, are often geared more towards existing businesses, often with multiple locations. Franchises such as these often have mechanisms in place to collect data anyhow, so a tool like Chatmeter can be slotted into the corporate level quite expediently, allowing them to give top down directives without increasing the workload of individual franchises. Our tool, on the other hand, has a much greater value proposition for a new or prospective business owner about to embark on a new business. Additionally, Chatmeter also purports to provide actionable advice, whereas our goal is to allow the user to draw their own conclusions based on what our visualization shows rather than prescribe what is to be done.

4 Description of the Interactive Visualization Solution

Overall, our visualization aims to offer an accessible experience to users who may not be familiar with advanced visualizations or custom software. To achieve this, we adopted a guiding philosophy for our visualization types, dashboard design and interactions to be familiar (Line Charts, Scatterplots, Drop Down Menus, Zoom and Panning) or intuitive to the user (Hexbin, Contour Toggle, Lasso, Brushing and Linking).

4.1 Dashboard

We have implemented filters and options such as selecting cities, categories, and ratings, which enable users to specify their analytical focus, aligning well with Interaction Intents and Action Patterns. This supports intents like "Show me something conditionally" and "Show me a different representation" by dynamically adjusting the visualizations based on user inputs.

The dashboard also minimizes conceptual, spatial, and temporal separations by presenting intuitive controls alongside real-time updates on the visualizations. For example, category filters and toggle buttons provide immediate feedback, reducing the user's cognitive load and interaction costs. The design also adheres to Directness of Interaction, as users manipulate the data representations (e.g., selecting ratings or enabling kernel smoothing) directly through the interface. For

establishing Model-view-controller patterns, we incorporate the refresh and update functionality (e.g., Refresh Kernel and Refresh Plots button) to cater to the users with visual feedback. To recapitulate, the dashboard design adheres to Shneiderman's design principles, such as consistency, reversibility of actions, and user control, fostering a fluid and engaging analytic experience.

4.2 Hexbin

We chose a hexbin plot overlayed on a map because using a map to represent geospatial data is intuitive and provides context such as street names and districts to our data. The hexbin overlay mitigates overplotting (especially compared to a scatterplot) and allows users to identify high or low restaurant density at a glance. The markers for the restaurants were kept to allow precise brushing and linking. A simple scatterplot instead would obscure data context due to overplotting provided by the geographical map.

The viridis colour scheme was chosen for its increasing luminance, colour blind safety and multiple hues, adept at showcasing subtle changes compared to single-hue color schemes. Individual restaurant dots are highlighted in red to signify selection, providing a contrast against the map and hexbin colors. Familiar interactions like panning and zooming allow users to hone in on areas of interest. The Lasso Select and Box Select tools are intuitive ways to allow users to select restaurants in certain regions, highlighting the corresponding restaurants in the scatterplot and facilitating continuity across plots.

Overall, this allows (future) business owners to make out neighbourhoods with a low density of restaurants where the competition is sparse. The selection allows them to explore these neighbourhoods further by gaining additional knowledge from the opening-time scatterplot.

4.3 Scatter-Contour Plot

Scatterplots efficiently map two numerical values. For our data, we mapped opening hour (x-axis) and opening duration (y-axis) in order to gain an overview of business operating times. Instead of implementing a Triangular Model for this task, we decided on the scatterplot as it was quicker and simpler to implement and could solve the task almost equally well.

To combat overplotting we implemented toggleable contour lines and set marker alpha value low. We superimposed the contours instead of juxtaposed since it directly puts the contours into context with the markers in addition to simplifying the overall visualization. When toggling on the contours the alpha level of the markers decreases further, accentuating the contours. Toggling the contours off will restore the previous alpha level.

Although jittering was attempted, it did not help with the overplotting issue. We chose our scatter-contour plot hybrid over other solutions like the Gantt Chart due to its superior scalability. We decided against using a Splatterplot, as its implementation was non-trivial, and we could not find a library for it.

A color-blind-safe color scheme was chosen for accessibility. Coloring was done by rating class since we deemed it more meaningful to compare across that than across weekdays. Checkboxes were implemented to toggle marker and contour visibility by rating class or weekday (contours are recomputed when reselecting weekdays). This enables the user to identify trends such as low-competition time slots and gain an overview of opening hour patterns of businesses with low, medium or high ratings, depending on user strategy.

As with the Hexbin, markers can be brushed on the scatterplot to highlight the corresponding businesses on the Hexbin map, allowing the user to discover their locations.

4.4 Line Chart

Line charts provide an intuitive way to make out trends in temporal data as the slope can easily be read off and gives additional insight to the user. The Line chart figure creates a 365 day rolling average of the star ratings of the highlighted businesses grouped by category. Additionally, comparison between different restaurant types is possible. The user can see at a glance how ratings for a specific type of restaurant have developed over time and compare their developments to each other. The colour scheme chosen is the same as that of the scatter plot and maintains its reasoning. Users can see how restaurants of different types have performed in the last years. This can help to make out threats and opportunities. Opening an Asian restaurant when most Asian restaurants in the city are highly rated might indicate a highly competitive market and no need for additional restaurants in that category.

5 Example Use Case

Let's pretend I am a young entrepreneur living in Philadelphia looking to start a new Chinese restaurant. I want to use the visualization as a tool to figure out the best locations and optimal hours for my restaurant.

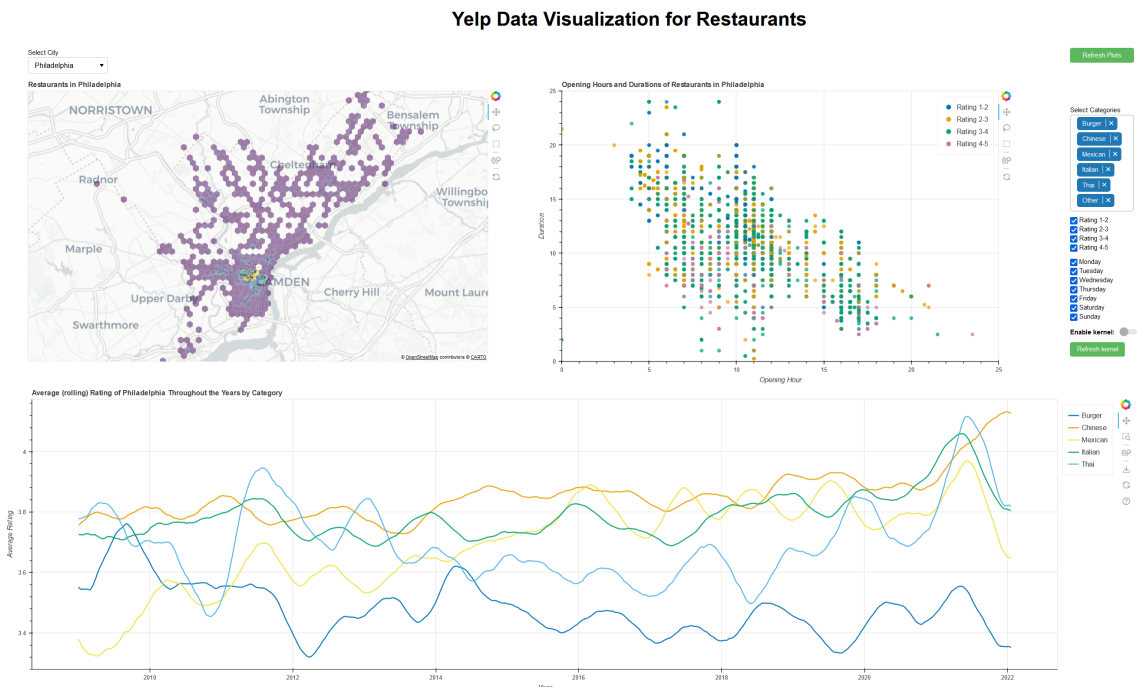

Figure 1: Upon opening the visualization, I can get an overview via the hexbin plot on the left. Being familiar with Philadelphia, it's pretty clear to me that the areas around City Center, Callow Hill and Queen Village have the highest concentration of restaurants. This signals both high competition and foot traffic, making these lucrative but competitive locations.

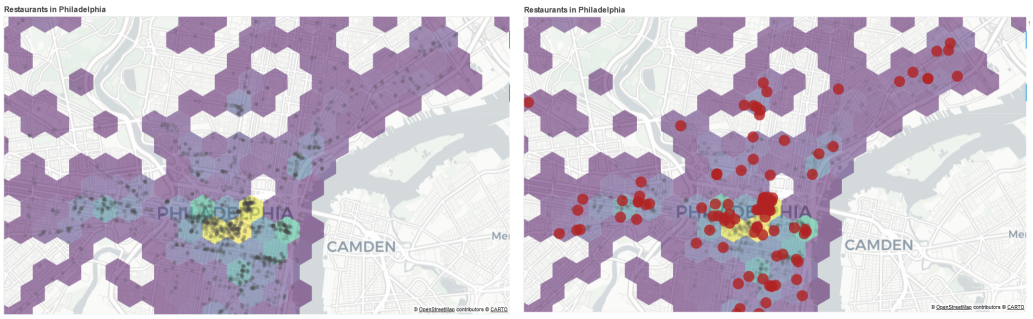

Figure 2: To refine my search further, I select "Chinese" in the multichoice menu, as my primary competition will be other Chinese restaurants. I can notice a slight shift in concentration northwest, identifying Chinatown. However, it's also evident that Chinese restaurants are all across the city, suggesting opportunities in various neighborhoods beyond Chinatown. Let's select the main traffic area (including Chinatown) next to get an idea of local trends.

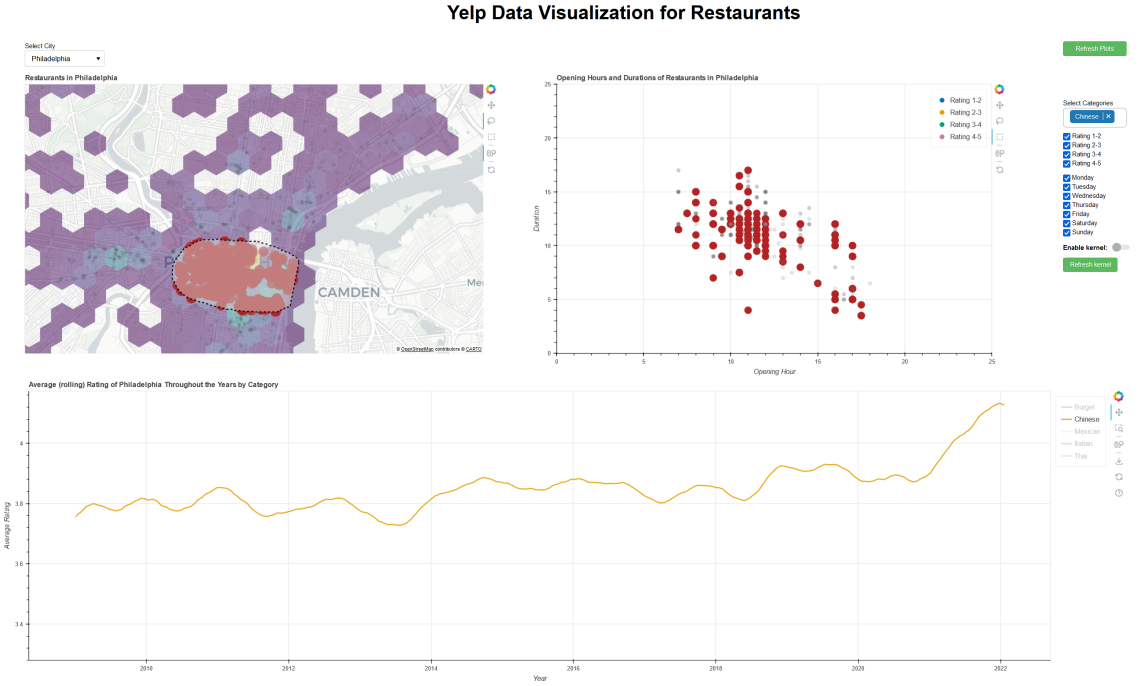

Figure 3: Despite the high competition, Chinese restaurants seem to be going up in ratings recently and indicating that our category will face significant competition. Now, let's filter reviews based on 1-2 stars and 4-5 stars using the scatterplot and enable the contour view to optimize operating hours for success.

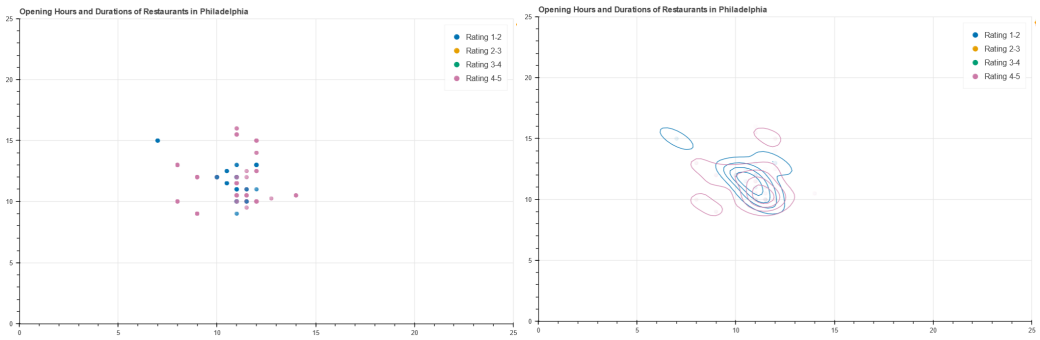

Figure 4: Unfortunately, there isn't a clear trend between the two sets of reviews. However, both plots reveal that many restaurants tend to open between 10 AM and 12 PM. Interestingly, starting earlier doesn't seem to correlate with a reduction in review quality. Potentially opening earlier could be a strategic advantage.

6 Discussion of Limitations

There are a couple nice-to-have features and performance improvements we would have liked to implement given more time. Primarily, this concerns runtime. When switching cities or toggling the contour plot, the delay is a little bit long. There are various smart ways to reduce this lagging time but most either involve major fundamental changes in how the program is structured or are tricks that are time consuming to implement. Furthermore, some of the runtime issues can be attributed to Bokeh as well as some issues regarding aesthetic considerations.

For example, the visibility of the contours could only be toggled on and off by removing them (as in deleting the rendered object of the contour) and recomputing them. We tried the basic solutions like turning down opacity, alpha and loading in the background - none of which were successful. There is likely a more elegant solution but we lack the technical familiarity with Bokeh and the time to invest into perfecting an already working feature. Another limitation associated with Bokeh is seen when panning on the Philadelphia map: some data points disappear until panning stops. Yelp provides a "category" attribute to each business consisting of a list of comma separated strings containing cuisine, food and business type. Theoretically, one 'categories' attribute could contain multiple different categories, such as 'Mexican' and 'Italian.' This could in theory present problems within our visualization, however, we did an analysis of our 10% random sample and found no shared categories among businesses. With this, coupled with our explicit category choices (mirroring Yelp) and the high implementation cost, the informed decision was made to leave it, but address it in the report.

7 Work Description

During the beginning of the project, each member of the group researched two datasets and presented them to the group, highlighting opportunities, potential shortcomings and potential visualizations that could answer questions not apparent in the data. After a group vote, the current dataset was chosen. The big picture design ideas were laid out jointly as a group. During this process, we discussed chart types, layout and user experience. Once those big picture ideas were decided upon, the finer details were nailed out by specific group members who also brought the visualizations to life.
Broadly speaking, these were split into:

- Hexbin & Line Chart, taken on by Benno
- Scatter & Contour Plots, taken on by Andreas
- Dashboard, taken on jointly but primarily by Sadik

All three presented hand drawn prototypes as well as lo-fi mockups to the group, where design discussions were had and decisions were arrived on. Finally, interaction was added and the charts were finalized.

Simultaneously to this process, Eric worked on the data preprocessing and overall ensuring that the data were provided for the visualizations to run without a hitch. This included general data cleaning, deciding what to do with missing values and how to connect the disparate JSON-files. Later, Eric also embarked upon adding the functionality of switching cities and the backend logic backing it up.

The discussion posts were initially started by Benno but were later taken on by Eric after a few weeks. Benno also set up GitHub. While we all tried to keep our code clean and documented, Eric refactored all of the backend code to ensure it was structured and in line with best practices. Benno later undertook a similar endeavor with the actual plot generating code.

The report rough draft was primarily written by Eric, with members contributing their insider information to their portions of the project, guided by questions provided by Eric. The report editing was also mainly Eric's task, again with joint group participation. Finally, the video was discussed and planned by the group before finally being recorded by Andreas.

References

[1] Chatmeter. https://www.chatmeter.com/. Accessed: 2024-12-10.

[2] Yelp. Yelp open dataset. https://www.yelp.com/dataset. Accessed: 2024-12-10.