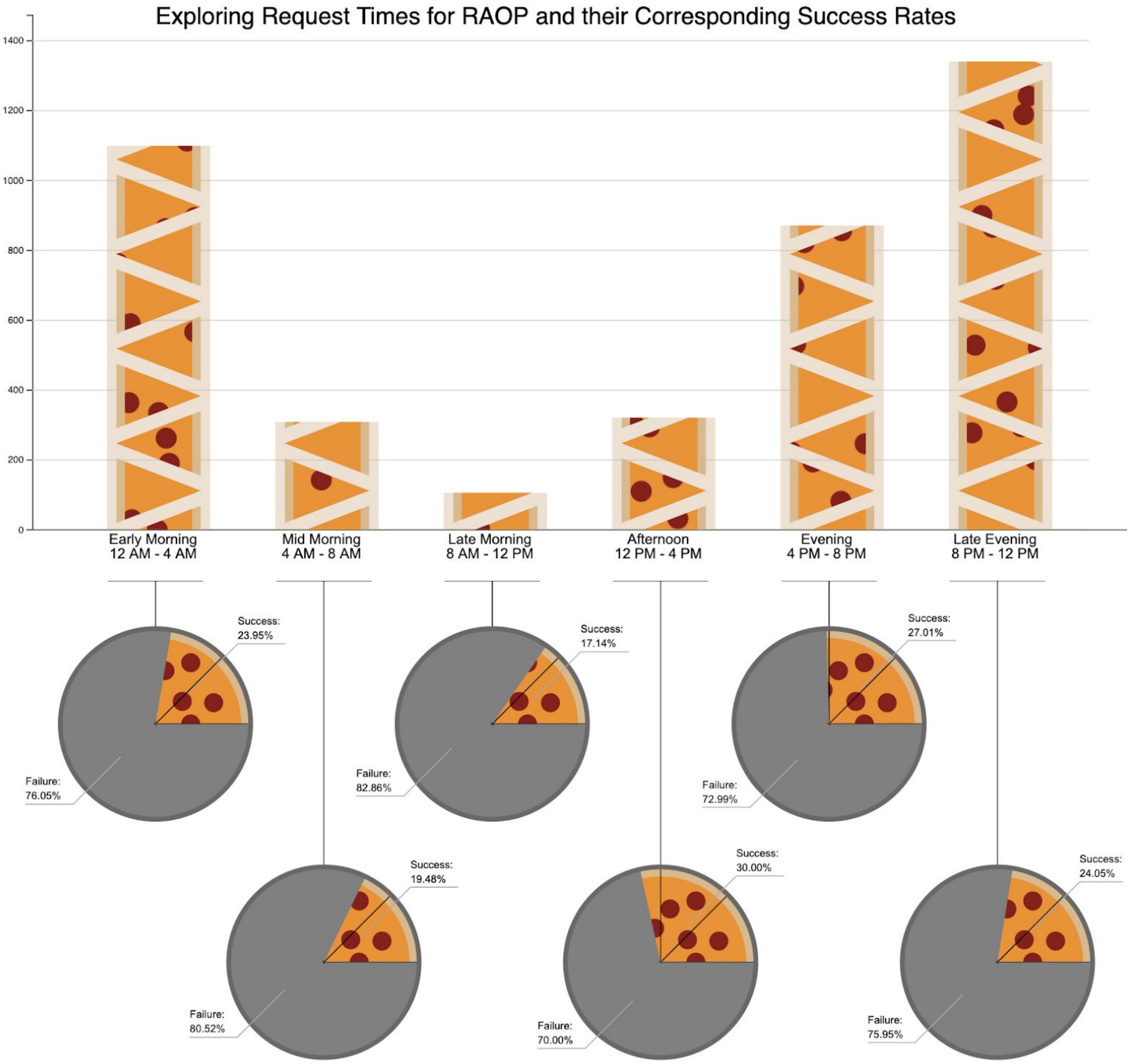
INFO 3300 Project One Written Description

By Nikki Bregman, Dylan Magsarili, & Divya Ramakrishnan

GitHub Repository: https://github.coecis.cornell.edu/nb487/info3300-project1.git

Final Visualization:



Description of Data

We obtained our data from the Kaggle dataset containing 5671 requests collected from the Reddit community ‘Random Acts of Pizza’ (RAOP) between December 8, 2010 and September 29, 2013. The data is stored in JSON format, where each JSON entry corresponds to one of the 5671 collected requests made by Reddit users. The fields included in the dataset are comprised of user information such as username, number of upvotes/downvotes at the time the request was collected, how old the requester’s account is at time of collection, whether or not the requestor edited their comment, to name a few.

We reformatted the data from JSON structs into JavaScript arrays. Then, we decided that we wanted to use a subset of the data to improve usability. The field we chose to display the information in the bar chart was the "unix_timestamp_of_request_utc". This field stores the time the request was posted in Coordinated Universal Time. There is another field “unixtimestampof_request,” which supposedly stores the time the request was posted, in the timezone of the poster, but the dataset authors noted that it likely isn’t very accurate, so we used the UTC field. From the unix timestamp, we created a JavaScript Date object, and then used the Date function ‘getUTCHours()’ to obtain the specific hours listed in each time stamp. Then, we used the hour of each individual request to group requests into different 4 hour buckets, i.e. 4am-7:59am.

Additionally, we filtered the array field [‘requester_received_pizza’] into two variables denoted ‘yesPizza’ and ‘noPizza’ to be able to break down whether or not the requestors grouped into each specific time frame successfully received their pizza.

Design Rationale

We came up with a two-part narrative that we wanted our visualization to convey. Firstly, we wanted the bar chart to display to the user, “Do people make more requests for random acts of pizza at certain hours of the day more than others?”. Secondly, if it indeed appeared true that more people make more requests for pizza at certain hours of the day than others, we wanted to examine whether or not Reddit users who made requests during these more popular request times were more successful in receiving pizza. Since our narrative had two distinct parts, we decided to use two charts.

For the first part of the narrative, we decided it would be most feasible to display these results using a bar chart. There were a multitude of reasons why we determined this would provide the most effective visual mapping of the data. The general use case for a bar graph is to make comparisons between different groups, and track changes over time. We used the various heights of the bars in the graph to denote how many requests were made by users in the various segments of time. Using a bar graph allowed us to effectively see the changes over a period of time and easily identify differences in request amount per segment.

The second form of visualization we used was a pie chart. Firstly, we decided that if there was a way to integrate a pie chart into our visualization, we would do it, because it would be a great opportunity to connect the method of visualization to actual subject matter of the data we were working with - pizza pies. Secondly, using a pie chart would simply and effectively convey the second portion of our narrative: whether or not requesters in each time frame bucket were successful. Pie charts are generally used to display percentages, and are easiest to decipher by the viewer when there are not many things being compared. Since we were only comparing two things: % of requests in the time frame of choice that were successful, and % of requests that were not, we concluded that using a pie chart to convey this information was the most rational design choice.

Once we had established the types of charts we were going to use in our visualization, we then explored the various marks and channels that we could use. The marks for the bar graph were the bars. In terms of channels, we included both magnitude and identity channels. Magnitude channels that were used were the positioning of the bars, the length of each bar, and the saturation/luminance of the bar background. In terms of saturation of the background of the bars, we wanted the slices of pizza to stick out against the white background, so settled on a light pink. We utilized triangles as a categorical physical channel to display various slices of pizza. These triangles were tilted to be horizontal facing, and were positioned spaced apart and alternating in the direction they faced, to essentially preserve the integrity of the rectangular shape of the individual bars. We varied the color hue of the different segments of the specific 'pizza slice' triangles until we were confident it bore an identifiable resemblance to actual pizza slices. We chose to incorporate the triangular pizza slices into the design to attract the eye of the viewer and allow them to strongly identify the subject of the data being displayed.

The bar chart has bars filled with pizza slices, which relates to the visualization theme. The pizza slices within the bars in the bar graph were created by making functions that built individual pizza elements, and then stacked them up to the necessary height. The slices are generated by drawASlice() function, which draws the slice based on given coordinates and adds a pattern fill with randomized circles that represent pepperoni. The slices are added to a bar group, so each bar has its own group.

The pizza slices are polygons with pattern fills. The functions pizzaSlicePoints() and reversepizzaSlicePoints() return 2 strings in an array. Those strings are the x,y coordinates of the vertices that, when passed as points to 2 polygon objects, creates 2 triangles that fit on top of each other, creating the appearance of a pizza slice and crust.

Then, the shape of the bar is used to make a clip path that is applied to the bar group, removing any pizza slices that fall outside of the bar. At first, we used white rectangles on the top and bottom of the chart to cut off excess slices, but this required the slices to be below the white rectangles, which needed to be below the annotations. This meant that the gridlines, albeit a faint light grey, were drawn over the chart. The clip path method solves this issue.

Once we observed the data visualized through the bars of the graph, we determined that we were not going to apply any transformations to it, since we were satisfied with the way it was displayed and linear scaling.

For our pie charts, we used spatial region marks with varied 2D areas as channels, as is standard for pie charts. In addition, since our pie charts are only displaying binary values of yes and no, we split our identity channels into two patterns. One is a pattern resembling a slice of pizza, and the other a solid gray color similar to a tray that pizza might be served on. Each pie chart shows the percentages of successful and unsuccessful pizza retrievals for a given time period, with the pizza section of the chart representing the percentage of posts that resulted in a successful retrieval, and the tray section of the chart representing the percentage of posts that did not result in a successful retrieval.

The pizza sections of the charts were created by using a number of layered circle elements to represent the pizza crust, cheese, and pepperoni, along with line elements on top to represent slices and provide a visual indicator of 12.5% of the spatial area of a chart. The tray section of the chart was made by using the stroke-dasharray attribute to create a circle with a very wide stroke that covered a percentage of the circle's circumference equal to the percentage indicated by the data. The code for the creation of these sections can be found in the cutOut() function. Additionally, we added labels to the graphs that indicate which section correlates to which type of data, as well as a numerical indicator of the percentage to supplement the area channels. The code for the labels can be found in createLabel().

Though there were other options for visualizing this aspect of the data, we decided to utilize pie charts because they struck the right balance of displaying the data and thematic resonance. As mentioned earlier, we felt that pie charts were an excellent fit for our dataset given the pizza thematic and the visual similarity between pie charts and pizza. We feel that this thematic resonance makes the visualization more engaging to the user than a more abstract visualization would.

The Story

We intended for the visualization to provide some insight into *how* random 'Random Acts of Pizza' truly are. There were a multitude of fields to choose from in the initial dataset. However, we wanted to choose one of the fields and analyze it in detail, whilst preventing our visualization from becoming cluttered. We thought that if we looked at a singular attribute held by each of the 5,761 users that this dataset was collected on, and further observed the success rates of the various groups within that attribute, we could see if any underlying data trends or similarities between the users for this attribute affected the outcome of obtaining pizza.

We were surprised at how clear of a difference is displayed by the heights of the various bars in the chart. When we decided to use 'time request was made' as the field to observe data from, we hypothesized that certain time frames would indeed contain more requests made, and display larger success rates. In terms of general cultural trends, we assumed people would probably not be craving pizza, or surfing Reddit threads during the working hours of the day. And we indeed did see this in the display, as the heights of the bars in the buckets 4-8 pm, 8pm-12am, 12-4 am were significantly higher than for the buckets 4-8 am, 8am-12pm, 12-4 pm. It is important, however, to note that the times are in UTC, not EST, and the majority of the Reddit user base is in America. As discussed in the beginning, we chose to not to use the "local" time field because the dataset authors said it was unreliable, so we used the data we knew was correctly standardized.

We wanted our visualization to give the user insight into times where making requests for pizza appeared more successful than others. Ultimately, the bars in the buckets 4-8 pm, 8pm-12am, and 12-4 am did end up displaying a slightly higher success rate than the other bars, as shown by the percentages of the pie chart. Therefore, our chart does support the idea that perhaps requests made during those times are met with slightly more success than those made during earlier hours of the day. Although our visualization, as well as the information provided by this single dataset is not enough to draw concrete conclusions about Random Acts of Pizza, it does allow the viewer to see the success rate of a large group, and decide whether or not to make note of this information when perhaps requesting or making a random act of pizza themselves.

Outline of Team Contribution:

Nikki Bregman: Created the bar chart for the visualization, wrote code to convert data from JSON & filter it, contributed to report

Divya Ramakrishnan: Worked on pie charts for the visualization, wrote most of report

Dylan Magsarili: Worked on pie charts for visualization, wrote code to convert data from JSON & filter it, contributed to report

Hours spent developing : in total roughly 38 cumulative hours were spent developing.

Breakdown:

3 Hours: planning visualization
Comments: This is the sum of the time spent meeting in person to discuss our ideas, meeting with the TA for further advice, sketching out and debating various chart/layout options, exploring which field from the dataset we wanted to work with, and deciding on the Kaggle Random Acts of Pizza Dataset dataset

20 hours: coding the data visualization
Comments: Most of this time was spent understanding how to effectively visualize the data. We decided on which charts to use fairly early, but it took a significant amount of time to decide which marks and visual channels would tell our data's story in the best way. Many rudimentary pie chart options were tested, because a multitude of pie chart implementations made it difficult to utilize the existing Pizza pie implementation Dylan created in the initial stages of the project. The bar chart was very intensive in coding the slice style, because it involved a lot of calculations to be scalable.

5 hours: loading, filtering, and sorting the data
Comments: The Kaggle dataset was relatively straightforward and easy to work with, so this step did not take too long. More time was spent deciding how to filter the data.

5 hours: combining each group member's individual work and finalizing our ultimate visualization
Comments: In this step, we made sure the different charts fit cohesively in the visualization and combined the code for the individual sections we were working on, as well as added finishing touches to stylistic elements of the graphs.

5 hours: Writing Report
Comments: it took less time to describe the story we were trying to tell with the visualization and give a description of the data, since both of those were clear from the start of the project. The design rationale took the most time because we had made many distinct decisions about which particular aspect of the dataset to work with, and how exactly to best visualize this aspect.