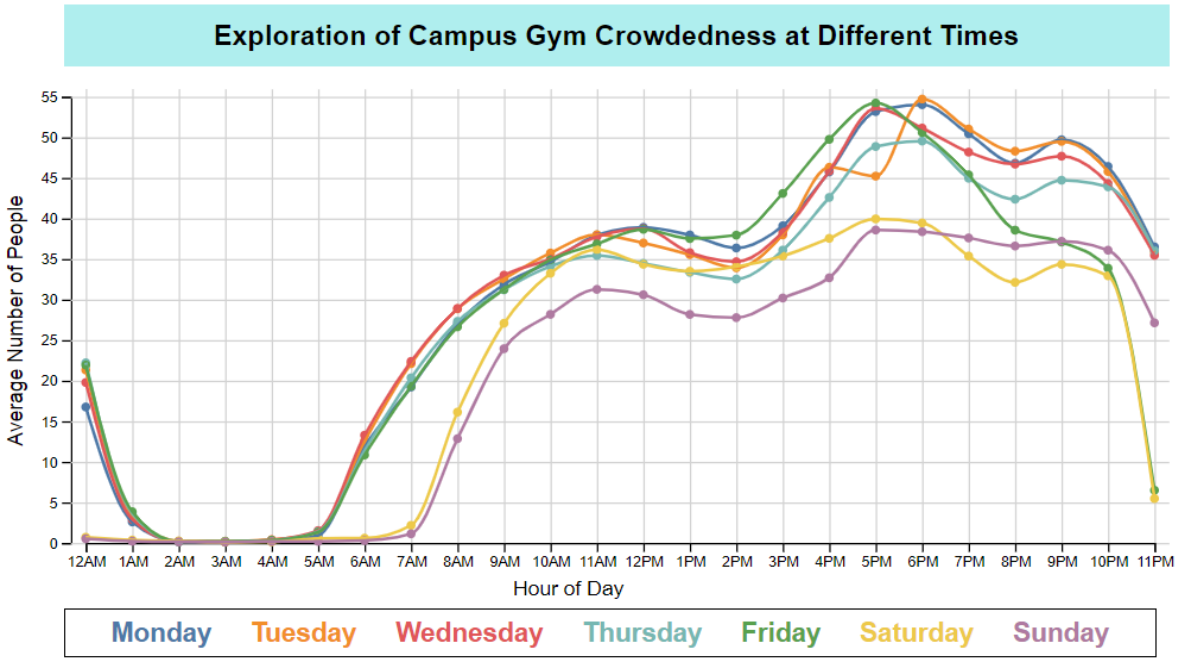
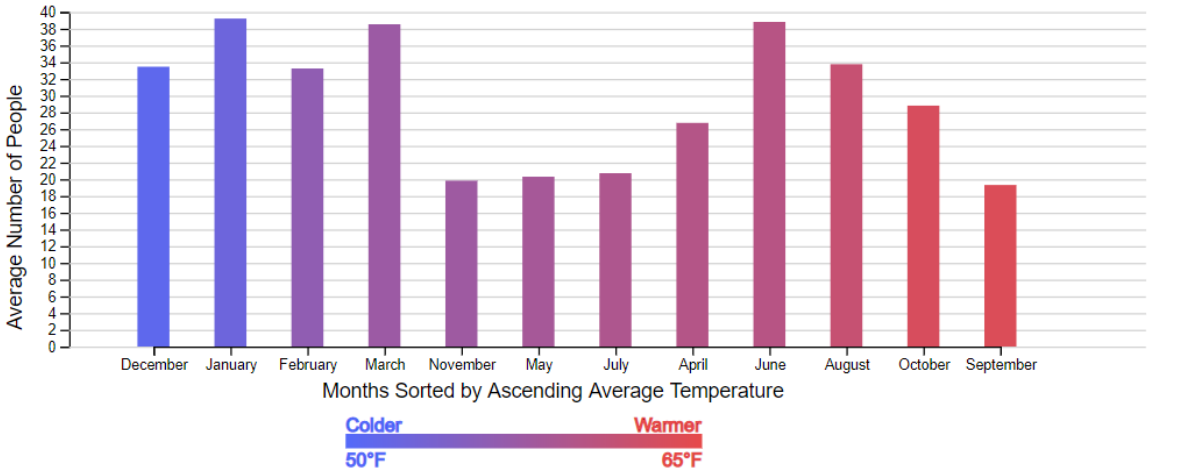
Project 1 Write-up

CS/INFO 3300

Thomas Bastis, Mohammed Hussien, Jason Zheng





Description of Data

The dataset we used for this project was sourced from Kaggle; it consists of 26,000 people counts (about every 10 minutes) from late 2015 to early 2017 of the number of people in a particular gym at UC Berkeley, as well as extra info including temperature and semester-specific information that might affect how crowded it is. In our visualizations, we made use of 5 variables in total: **number_people** (int), **day_of_week** (int; 0 [monday] - 6 [sunday]), **temperature** (float; degrees fahrenheit), **month** (int; 1 [jan] - 12 [dec]), and **hour** (int; 0 - 23).

In terms of pre-formatting the data, d3.autotype was used. And in order to more easily plot the data (for our first visualization), the average number of people in the gym, for each hour, for each day, was precalculated and stored in a new data structure. No points were filtered out, because there were no noticeable irregularities in the data related to the areas we were using.

Design Rationale

Our primary objective is to find the best times to go to the gym on college campuses.
We achieved this by visualizing the relationship between different times (months, days, hours) and the average number of people that are at the gym during those times.

We initially chose to create three visualizations for each unit of time to see if there were consistencies, any observations that stand out or any patterns that we could discern. The natural thing to do felt like bar charts with units of time on the x axes and average number of people on the y axes. The marks being the lines and the channels being vertical lengths and horizontal position.

After looking at those observations, we wanted to merge some of the visualizations to better grasp any pattern or relationship, especially the visualizations for days and hours.
For the visualization showing months and average number of people, we wanted to incorporate temperature and see if that could add value to the bar chart we had.

Hour/Day Visualization
In order to best combine the data for hours and for days, we decided to plot lines that would be color-coded depending on the day. Essentially, the x-axis would have hours (24 hours) and the y-axis would be the average number of people at a particular time on a particular day. So, we would plot seven line plots for the seven days of the week. The marks would be lines and points while the channels would be horizontal and vertical position as well as colors. In order to help viewers better visualize times, we've adjusted the formatting of the x-axis making sure it uses standard time instead of military time. In addition, we made sure to distinguish between morning and afternoon times by adding 'AM' and 'PM' to the hour labels.

We used a linear scale for both the hours and the average number of people and used an ordinal scale for the days. For the domain of the scale for the average number of people, we

decided upon a range from 0 to 50 because the averages are less than the extent of the full data. For the color scale, we used a d3.schemeTableau10 since we felt it was the best scale that allowed us to see through different overlaps.

Since we had lots of data to work with on any particular time and day, and since our line graphs needed to pass through one data point per hour, we had to calculate the average number of people at a particular time on each particular day. This gave us seven data points that a line plot could go through.

The way we did this was to create an array that held seven dictionaries. Each dictionary was for a particular day of the week, and had a key for day, hour, and the average number of people at the gym at that day and hour. From there, it was simple to define a line generator for our dictionaries, and then iterate through the array plotting the seven lines.

We also added axis labels so viewers could more easily understand what the variables on the chart were, as well as a legend at the bottom of the chart to inform viewers which colors correspond to which days. The dataset used numbers ranging from 0-6 to categorize days so we had to convert them to their respective string formats.

Month/Temperature Visualization

In order to best supplement our first visualization, we decided to create a bar chart of months vs. gym crowdedness, with average monthly temperature encoded in the bar stroke color.

Similar to the y-axis of the first visualization, we took the average number of people for a particular month because of the massive amount of points in our dataset. For the scale, we decided upon a linear scale since the bars filled the scale nicely. For its domain, we decided upon a range from 0 to 40, once again because the averages are less than the extent of the full data.

On the x-axis, months were first sorted by their average temperatures in ascending order, and then converted into their English equivalents (1 = January). These months were then plugged into the domain of our x-axis ordinal scale. To supply the ordinal scale with a range, we manually computed breakpoints given the start and end x-axis pixel coordinates.

For the color scheme of the bars, we decided on a custom-defined gradient from a slightly light blue to a slightly light red, because people associate cool temperatures with cool colors like blue, and warm temperatures with warm colors like red.

Finally, we added axis labels to help the viewer understand the month ordering on the x-axis and enhance overall clarity, as well as a legend to explain the exact gradient scale.

The Story

Our main visualization shows us that there are two peaks in gym crowdedness over the course of a day. The first peak is around 11AM-12PM, and the second around 5PM-6PM. These are consistent across days of the week, and may be explained by people being more likely to go to the gym just before eating a meal (lunch or dinner). It also shows that the weekend in general is a less popular time to go to the gym, at least in the later hours. Sunday in particular is noticeably less crowded than other days of the week, for most hours of the day. This is not particularly surprising, as it could be explained by many gym goers having Sunday as a rest day. Another noticeable phenomenon is that some days of the week have more drastic drops in crowdedness after the dinnertime peak. In particular, Friday has a huge drop-off from almost 55 to less than 40 in the span of three hours. This may be explained by the higher likelihood for Friday to be the day for social gatherings at night, which would be a significant factor given that this dataset is describing the habits of college students. Perhaps surprisingly, Saturday does not have as dramatic of a drop off, but this can likely be explained by the relatively low peak attendance, therefore not allowing the opportunity for a huge drop-off. Additionally, it seems as though all of the weekdays have a significant number of people going to the gym early morning (5AM +), as opposed to Saturday and Sunday, in which students seemingly have a much greater likelihood of sleeping in later. However, such a conclusion should be made carefully, as we are not certain if the gym's opening hours varied for the weekend vs. weekday.

The purpose of our secondary visualization is to explore how gym crowdedness varies per month, and see how temperature affects gym crowdedness. We see that the most popular month for gym goers is January, followed by June and March. One explanation for this interesting phenomenon could be that many students make New Year's resolutions in January to get back into shape, and thus visit the gym more often. We also see that gym crowdedness begins to decrease consistently after June and then bottoms-out in September, which can perhaps be explained by the fact that August to September are the warmest months in Berkeley, California. Besides the possibility that people are less likely to go to the gym when it is very hot outside, this phenomenon might be explained by summer break (when many students leave campus) and the start of Fall semester, when students are busy getting settled in. Surprisingly, gym crowdedness is very low during May, November, and July, the three middle months in terms of average temperature. This may be explained by Berkeley's academic calendar, in which finals take place during May and Thanksgiving break happens in November.

We were initially surprised by September having a higher average recorded temperature than August, but after reviewing historical temperature data from Southern California, we are less concerned since we may have underestimated how different Berkeley is compared to the Northeast. Another surprising thing we realized was that the number of data points available for each month was not consistent and that there were significant differences between months. This is especially apparent between the early half of the year (Jan-July) which has less data then the last part of the year (Aug-Dec). This could certainly have had an effect on the seemingly irregular temperature patterns of UC Berkeley.

In summary, we'd like to convey that the best times to go to the campus gym (assuming you want the minimum number of people there, and don't want to go in the middle of the night) is on the weekend, around 1-2PM, and when it's a modest temperature month out, around exam season or student breaks.

Team contribution

Jason: Coded most of visualization 2, wrote most of the design rationale for visualization 2, and wrote most of the story for visualization 2. Came up with different ways to combine visualizations.

Mohammed: Found ways to combine visualizations, coded the initial version of visualization 1 and edited design aspects (colors, labels, legend) , helped with parts of visualization 2, coded the legend for visualization 2, wrote most of the design rationale for visualization 1 and parts of the story section.

Thomas: Did most of the coding for the first visualization. Helped code the gradient for the second visualization. Typed most of the story for visualization 1, and edited all of this document. Also helped to brainstorm visualization ideas.

Time Taken:

~3 hours: Design and Brainstorming
~6 hours: Coding and Implementation
~4 hours: Finalizing, Polishing and write-up.