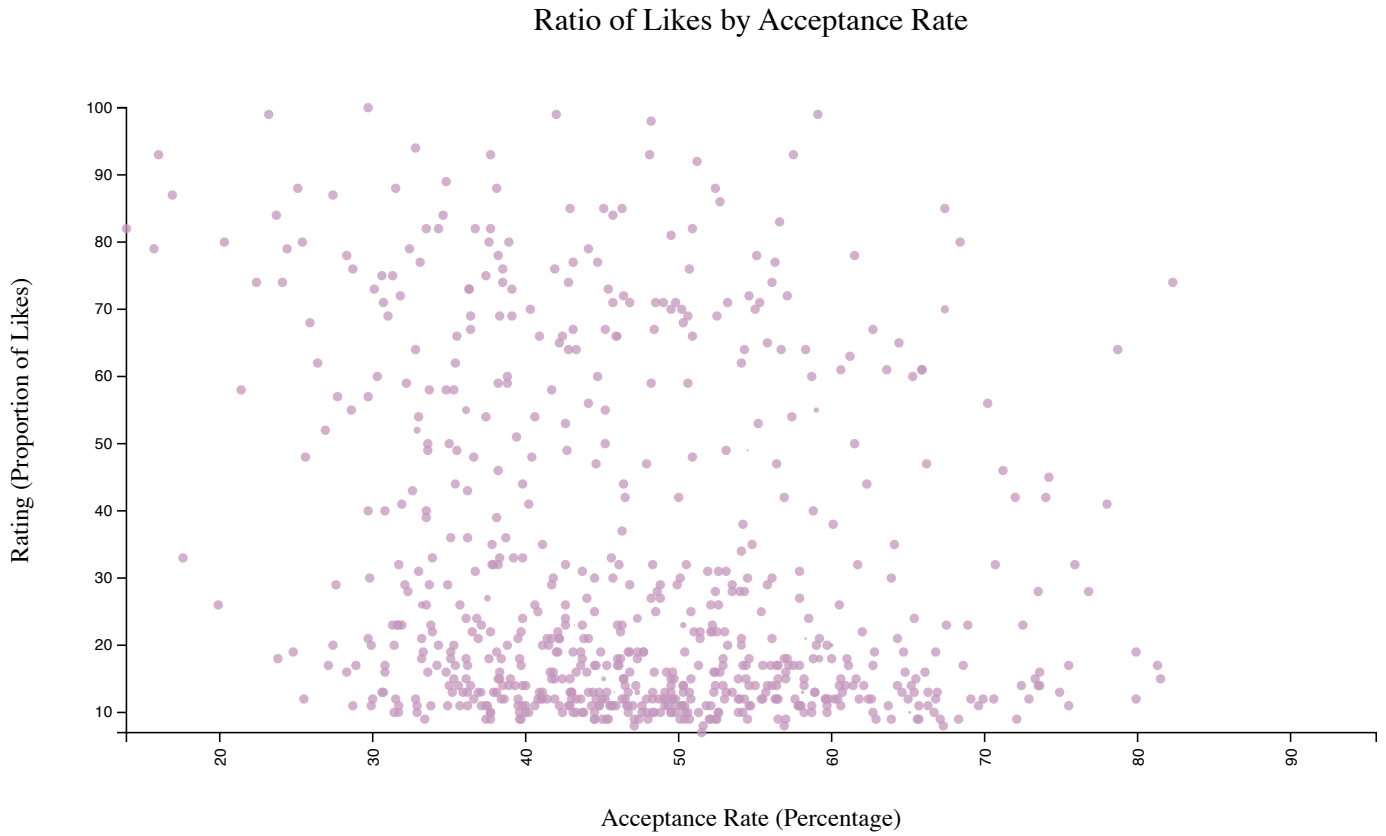
Final Project

Technology is a very lucrative field, and many people today want to work in industry. Some people start preparing years and years ahead of time, from participating in coding competitions to just practicing on Leetcode. The hope is to one day land a six figure software position. This project looks to analyze careers in tech, from the preparation step to what the future may hold. To do this, two datasets will be utilized: a dataset on Leetcode problems and a dataset on US software developer salaries.

Visualization 1:

Let's first look at the Leetcode dataset. A lot of people do practice problems on Leetcode, but not all solutions that people submit are accepted. On a (perhaps) unrelated note, each question can be liked or disliked by users. In this graph, we want to see if there is any relationship between these two factors for each question. Is there a correlation between the (relative) number of likes and the acceptance rate of a problem? Are questions with more accepted solutions more well-liked? We will use the ratio of likes over total likes plus dislikes rather than the total number of likes to standardize the number of likes across all questions. Watch as this graph plots each Leetcode question (as a coordinate of its rating and acceptance rate) on the plot as an animation.

Plot:



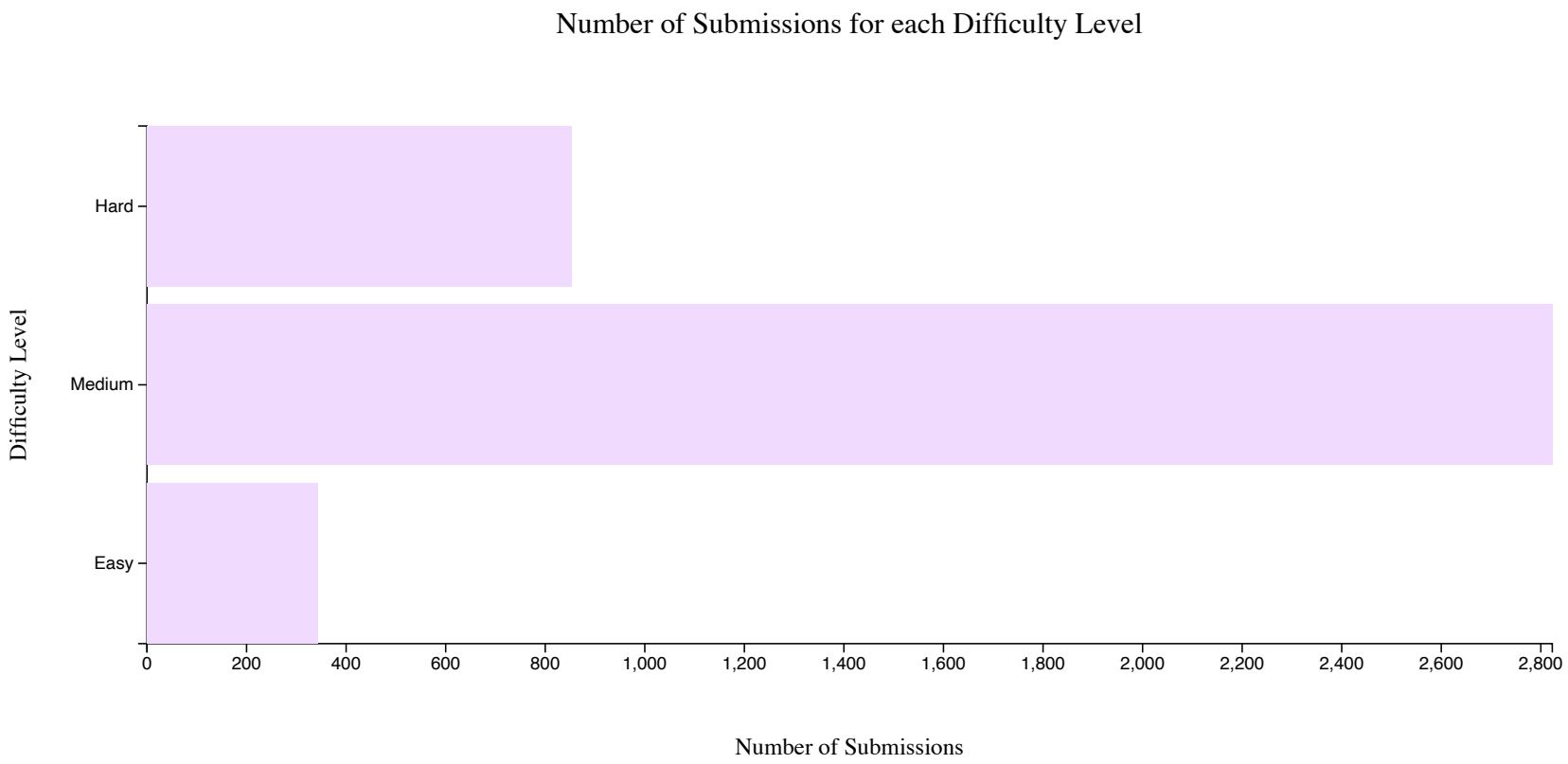
It seems like there isn't much correlation between likes and acceptance rate. Perhaps people like challenges, and whether or not they give a question a "like" is not based on if they could solve it or not, but rather what they thought of the problem overall.

Visualization 2:

While the number of accepted submissions did not affect the relative enjoyment of each Leetcode problem, we might wonder if more difficult problems are submitted less often. Maybe harder problems have less submissions because less people can solve them, and maybe easier problems have more submissions for the opposite reason. In this graph, we will graph the number of submissions for each level of difficulty, famously known as the "Leetcode Easy," "Leetcode Medium," and "Leetcode Hard." A "sort me!" button has been added so that we can sort the bars in descending order after plotting to see which category of difficulty is most submitted, and which is least.

Plot:

Sort me!

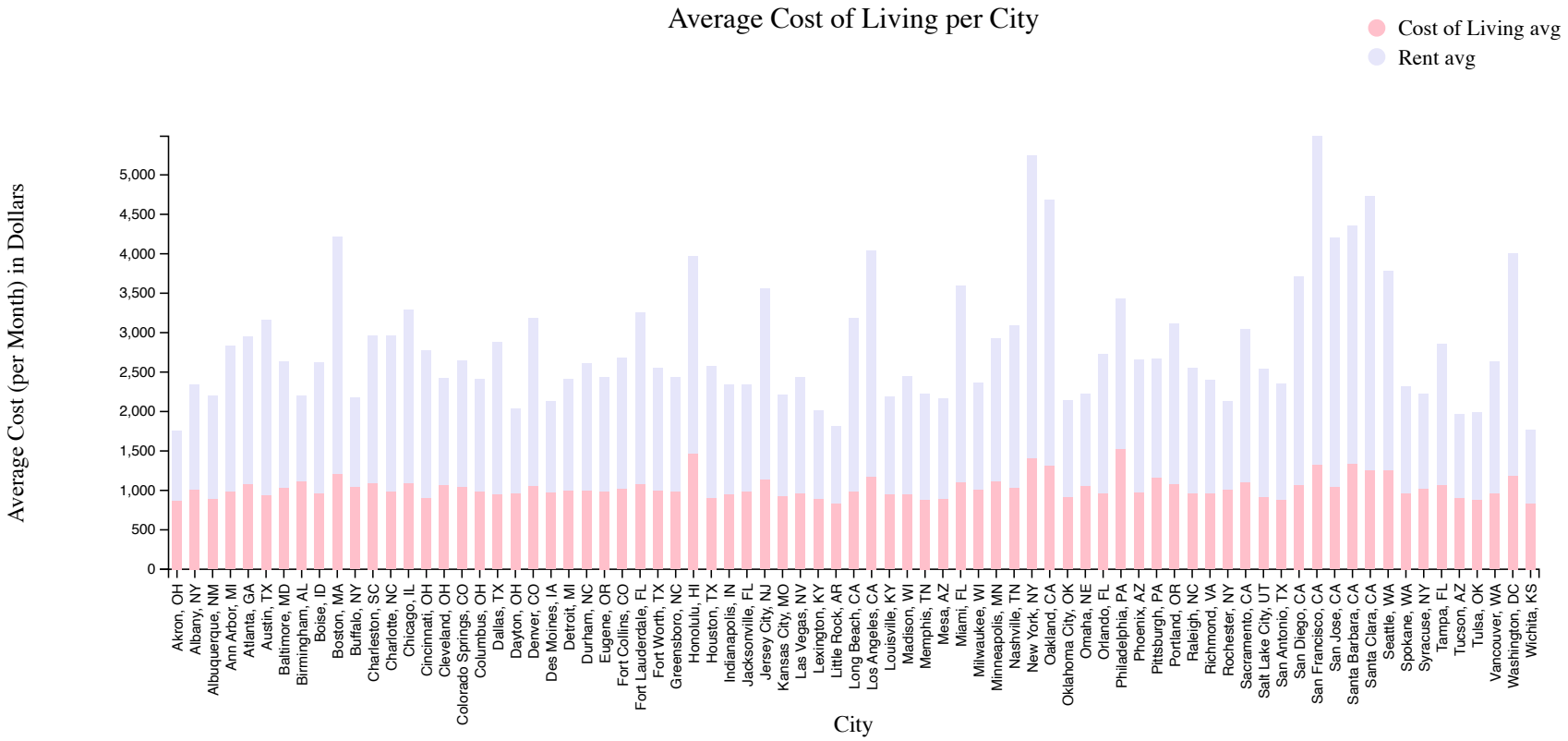


It's very interesting that "Leetcode Easy" problems are actually submitted the least. If we think about it, it could be that there aren't many questions that qualify as easy. The same could be said for difficult problems. It makes sense that the middle tier of difficulty is submitted the most then. Leetcode users are probably using Leetcode to prepare for interviews, and want to attempt more challenging problems, so easy problems have less submissions. However, our original hypothesis that harder problems are less often submitted due to difficulty in solving the problem could still be true for the medium vs hard problems, since more medium problems were submitted than hard problems.

Visualization 3:

As we just mentioned, many Leetcode users are using Leetcode to prepare for coding interviews. Many of them are looking for software development or software engineering internships. Many of these internships pay very well. Let's jump into their future and see what it might look like. Will their future high-paying jobs be an advantage? Software developers live all over the US, so we want to see the average cost of living in different cities all throughout the country. Which places could a software developer live comfortably? For this graph, we want to ask: What are the average costs of living across different US cities, and what proportion is made up of everyday costs vs rent? Since there are many cities that we are plotting, a tooltip has been helpfully added. When you hover over a bar, it will tell you the city and the two average costs we plotted.

Plot:



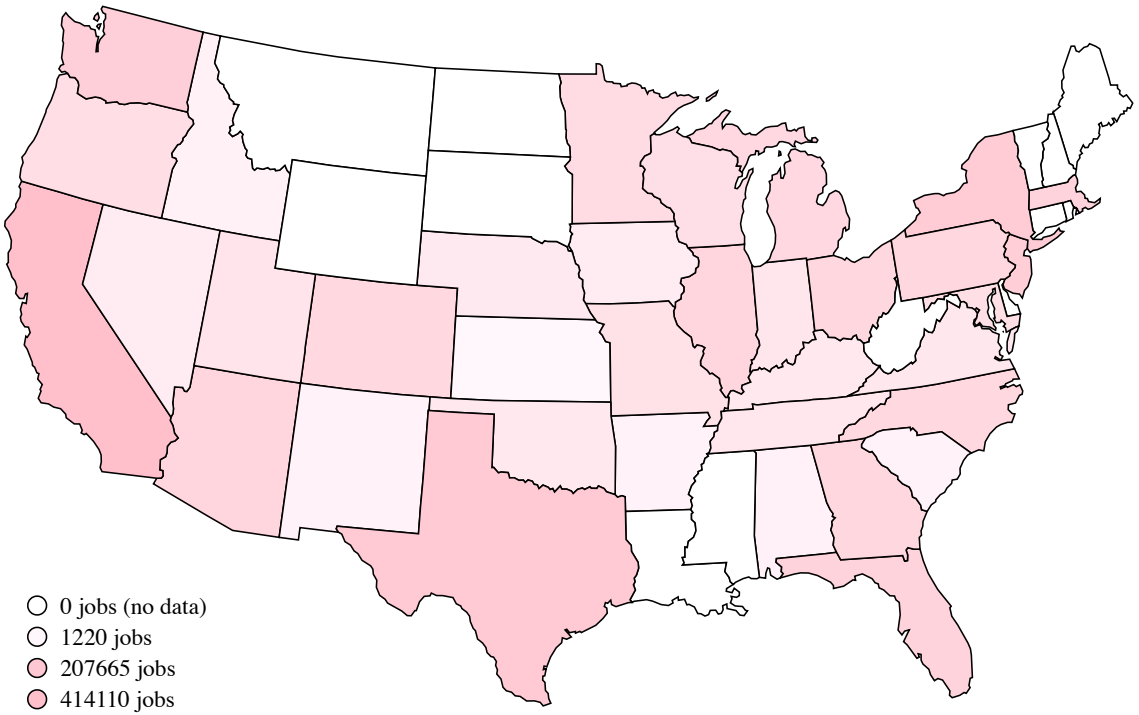
As expected, more popular cities in large metro areas (like San Francisco or LA) have a higher cost of living. It could be very expensive to live here as a software developer. However, these areas probably have a higher demand for software developers and perhaps pay better too, so it's a trade-off.

Visualization 4:

Knowing the average cost of living in different cities in the US, we might be curious as to how many software developers actually live in these areas. How does the number of software developer jobs differ across states? Are there large differences between certain states? To visualize this, we will plot the number of software developer jobs in each US state as a choropleth map of the US. To zoom in on the graph to see color differences better, use your mouse to select an area. The map will then zoom in on this area. Double click to zoom back out.

Plot:

Distribution of Software Developer Jobs Across the US

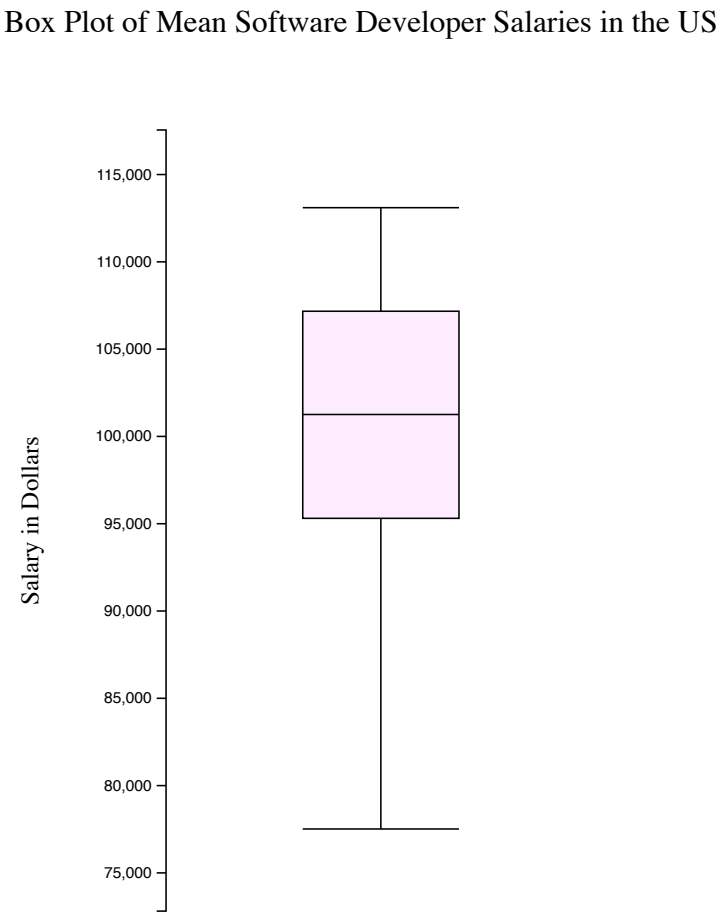


Unfortunately, our dataset did not include data on some states, so we have no definite answer for how many jobs are in those states. But as we can see from the rest of our data, California and Texas have the most software developer jobs. This makes sense with California's Silicon Valley and the recent rise in tech jobs in Texas. We also see the large discrepancy these states have with the other US states. Clearly tech companies favor certain areas.

Visualization 5:

Seeing all the software developer jobs in the US and the cost of living in different US cities brings us to the question of software developer salaries. What is the distribution of software developer salaries in the US? We will create a box plot using the mean sde salaries in different cities throughout the US.

Plot:



As we can see, the median software developer salary is a little more than $100k. Most salaries are around this point, with 50% of the values ranging from $95k to $107k. Maybe this means the cost of living is manageable for software developers. Perhaps all that Leetcode practice will pay off some day!

Visualization Descriptions:

Below, we will provide a brief summary and description for each visualization

Visualization 1

graph type: scatter plot
marks: points
channels: horizontal position (encoded by acceptance rate), vertical position (encoded by rating)
choice of color scheme: color chosen to match the overall theme, slight opacity used to show overlapping points

Visualization 2

graph type: bar chart
marks: lines
channels: horizontal position (encoded by number of submissions), vertical position (encoded by difficulty level)
choice of color scheme: color chosen to match the overall theme

Visualization 3

graph type: stacked bar chart
marks: lines
channels: vertical position (encoded by average costs in dollars), horizontal position (encoded by city), color (encoded by cateogry the cost falls in)
choice of color scheme: colors chosen to match the overall theme, two separate colors to denote the two subcategories we are graphing for each city (two different color hues)

Visualization 4

graph type: geomap
marks: areas
channels: horizontal and vertical position (encoded by location of state in the US that the data point corresponds to), color (encoded by number of software developer jobs)
choice of color scheme: color saturation based on a log scale of the number of software developer jobs that are in that state (more jobs means color is more saturated, and less jobs denoted by less saturated fill for the state)

Visualization 5

graph type: box plot
marks: lines (median line, whiskers), areas (the box)
channels: vertical position (encoded by average salary), length (of the whiskers, also encoded by average salary)
choice of color scheme: color chosen to match the overall theme