## The Narrative

Our project goal was to analyze job postings to discover in-demand data science skills.

We grabbed data primarily from the Indeed API and supplemented it with data from Hacker News. We focused on the Denver and Boulder areas but included national listings from Hacker News too.

We used the web scraping tool Beautiful Soup to extract and group skill phrases.

Josh wrote a dictionary of skill phrases that were pulled from the data science job summaries. That list was pared down to about 300 terms and skills.

Then we saved approximately a total of approximately 4,200 html job summaries for data exploration. They were all posted in the past 30 days.

Next, we identified a handful of skills that we will have learned through this class to compare with the skills list in the job summaries. Those included Python, Javascript, SQL, and Excel.

We plotted scatter and bar charts that were less than successful in their revelations. We found that Excel’s column and hierarchy charts actually suited our small data results well.

The top 5 skills requested to work in data science from our results are Python, SQL, Access, Excel, and React, followed closely by Javascript and Java. And the following pairings were the most evident.

Python : javascript, java, sql, react, and matlab

SQL : python, javascript, java, excel, access

JavaScript : python, java, react, sql, css

Excel: powerpoint, access, sql, python, javascript

We were able to verify our conclusions to some degree by comparing our results to those found on a Hacker News posting where someone else shared the same goal as our research project. For a June job search on HN, they found the following apps the most frequently demanded: **Python**, Ruby on Rails, Docker, **React, and Javascript**.

Notable difference: Rails and Docker did not surface as high in our research. So we might look at why that might be the case.

Some hurdles of the process entailed:

* Saving the data out to an HTML files and parsing the text so that we didn’t have to run the API every time we were playing with the code. This was more complicated that it sounded and took a good part of a class session to learn from Jeff.
* Generally speaking, we experienced a learning curve with our workflow. For instance, one team member was the first out of the gate and raced to writing code all hours of the day and night, setting up the basis for our work while the rest tried to catch up and figure out what was done and how to contribute.
* One of the challenges included how to deal with similar terms. Some had typos and acronyms and different ways to say the same thing. Was there a way we could synthesize the data into one list? We considered creating a synonyms list.
* We first tried to hone in on the qualifications by looking at the HTML code and extracting just the particular job qualifications part, but it wasn’t consistent through all the postings. We eventually decided to just search the entire job posting text and let the repeated terms float to the top.
* Another challenge involved collaboration. While we had a good span of time to work on the project, it also was a popular time for a summer recharge. Consequently, you had members working out of sync and not totally understanding what everyone was working on. We found it helpful to use Slack to communicate and even send files and snippets to each other, but keeping track would be difficult if the project were to extend much longer.
* Github posed a challenge in organizing files and figuring when to upload files and how to deal with large data files. We hacked this together, but our process was not sustainable.

One question that remains unanswered is what were the salary ranges for each queried job title? The answer wouldn’t have been anything magical to create, but a supplemental graph could add value to the conclusions we uncovered.

Another aspect that we considered incorporating was to allow the user to input their skills and output which skills they needed to acquire to be more marketable. We determined that it was easier to just list the top skills instead.

And while we were interested in how demanded skills were related to geography, that undertaking involved a bigger data set than we were prepared to tackle. Focusing on the Denver region made a sensible start.

To further make a practical application, we would want to return a URL where a user could apply on the company website, rather than through the aggregate Indeed compiler (which is virtually useless).

More research questions that arose from the project included:

* How could we predict which companies would need to post a job listing before they do?
* What are primary traits of the companies that are hiring these types of jobs?
* What are the most popular industries?
* Which companies are posting the most opportunities?
* And damn it, how much are they actually paying people? (We found companies are shy to reveal salary figures in most job postings.)