a place of mind
THE UNIVERSITY OF BRITISH COLUMBIA

COSC 329 – Job Scraper

Project Report

Mark Behnke
Talon Pratt

Table of Contents:

Github Link:

https://github.com/Markbehnke/indeedScraper

YouTube Video Link:

https://www.youtube.com/watch?v=kIY_0yoAINw&ab_channel=Spyrunite

Overview of Project Structure:

- **Main.py**
  - **Job scraper for indeed.ca**
- **languageCounter.py**
  - **Preprocesses and stems the text**
  - **Counts amount of jobs that require certain coding languages**
- **Ngram.py**
  - **Creates ngrams for each processed data file**

- **Logs folder**
  - **Contains all the raw data from our job scraper**
- **Medoids.py**
  - **Contains the clustering algorithm and the related dendrogram as a result. (See below for dendrogram)**
  - **This file uses TFIDFVectorizing to create the dendrogram**
  - **This file uses hierarchical clustering from scipy.cluster.hierarchy**
- **processesLogs folder**
  - **Contains all the processed data after stemming and preprocessing**
- **Ngram data folder**
  - **Contains all the ngram data**
- **ProjectCharts.xlsv**
  - **Data and excel charts that we took out of our data and found interesting enough to put in the report**

Details for each file, how they work, and problem we ran into:

Main.py:

This is the job scraper for Indeed.ca and it was quite frankly a nightmare to get working properly. Indeed was clearly not enjoying having us scrape their website for data and was constantly trying to stop us from collecting data. Using the python library

BeautifulSoup, we can easily scrape a website of all the contents of it. Getting the title, company, and salary for each job is very easy since it’s plainly printed on each search page, but the full details are not visible without going into another link for each job posting. The URL for each job posting has a unique html class id that changes for each job posting because indeed clearly does not want us to have an easy time scraping data, so we had to find something that each of the URLs for the job postings shared and the other URLs did not. We noticed that each job posting URL ended in “js=3” so we are looking for that in the URLs and making a separate HTTP request.

Now that we can get the full details for each job posting, collecting the data for each job posting should be pretty easy. Unfortunately indeed strongly dislikes bots and loves giving captchas. We somewhat got around this by setting a 10 second sleep between each page load, but they still gave us a captcha roughly every 50 pages so we had to load up a vpn and change locations every 50 pages or so. Because we had to slow down our data collection so much this step took a lot longer than expected, taking roughly 70 computing hours between the 2 of us.

languageCounter.py

This file does the preprocessing, stemming, and a count for programming languages. We made extensive use of the NLTK library in this file, using all 3 of the tokenize, stopwords, and PorterStemmer modules. With those modules it was fairly simple to take out stopwords, stem the remaining words, and tokenize them for easier use in data processing with the words. We were also interested in seeing what jobs required what programming languages, and since we already had all the data processed in this file we added in the counter to this file. This is separate from the ngram counter we made later for detecting skills. This just uses a simple bag of words to see if a job requires a language, but we also limited it to one occurrence per job posting since a job posting mentioning the same language twice doesn't mean that it's twice as required or that there are two jobs that require it. We also made sure to adjust the number of total jobs for the jobs that didn't scrape properly. These were almost entirely jobs that weren't hosted on indeed and instead just redirected to a different website. Since we can't scrape data from a website that doesn't have a consistent format to the rest, we just threw out these job postings from the dataset. The very last line on the processed file is the number of jobs properly scraped, which is read in by ngram.py so that it can properly calculate the % of job postings that contain a specific ngram.

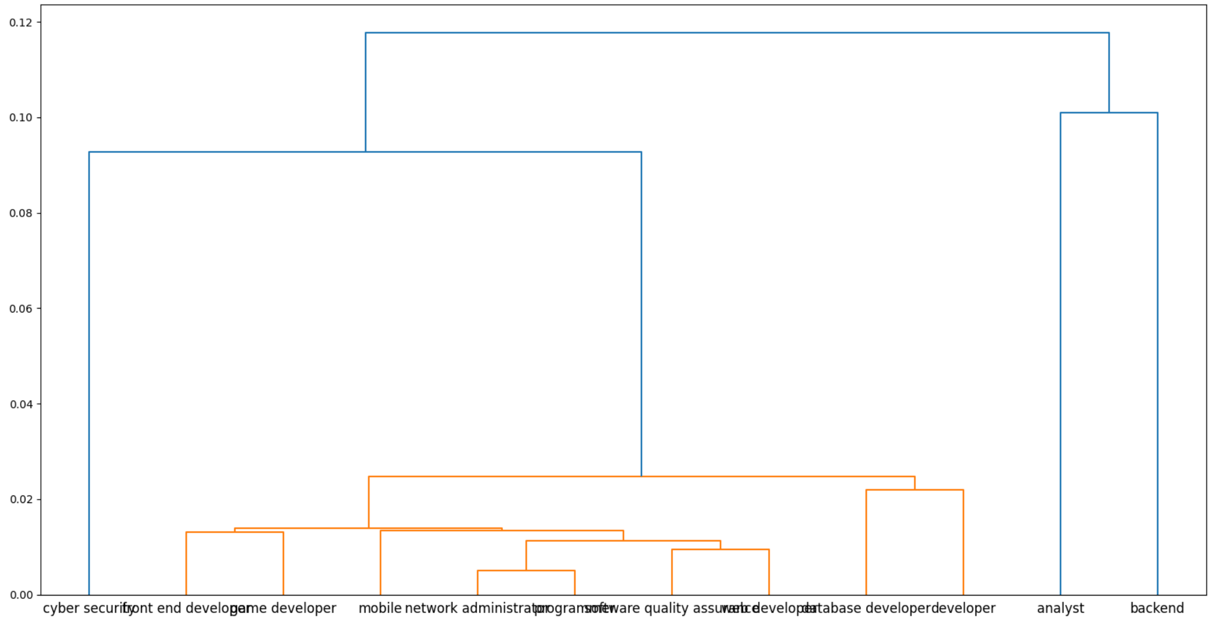
Ngram.py

This file creates all the ngrams for each processed data file. First we used the NLTK library to tokenize the data, and the regular expression library to take out all characters that weren't alphanumeric. This also removed all non-english postings. Ngrams are created again using the NLTK library's ngram module for all values of n between 1 and 6 and the top 100 ngrams for each job title are written to be used for visualization later.

Medoids.py

Similar to Assignment 2, we used clustering using hierarchical agloromitative clustering to cluster the job titles which share the most similar job postings. These are grouped off of which skills each job title shares with each other. To do this, we calculated the cosine similarity matrix between each document. We then took each cosine matrix and displayed the dendrogram of the result of this matrix.

Note: The x-axis titles have some overlap due to the length of the job titles. We tried to fix this by changing the figsize of the graph, but this did not work.



Jobs Scraped:

The following job titles got scraped for data collection for the project:

- Analyst
- Backend developer
- Cyber Security
- Database developer
- Developer
- Front end developer
- Game developer
- Mobile developer
  - Note, this job had the highest amount of French postings. This made the data for this particular dataset be skewed.
- Network Administrator
- Programmer
- Data Scientist
- Software Quality Assurance
- Web developer
- Cloud Engineer

Each of these job titles scraped the Indeed page limit of 100 pages, where each page had 15 jobs. This equates to around **~21,000** total jobs scraped.

Test Cases:

As this project was almost all web scraping and compiling the data, all the cases we tested were done manually.

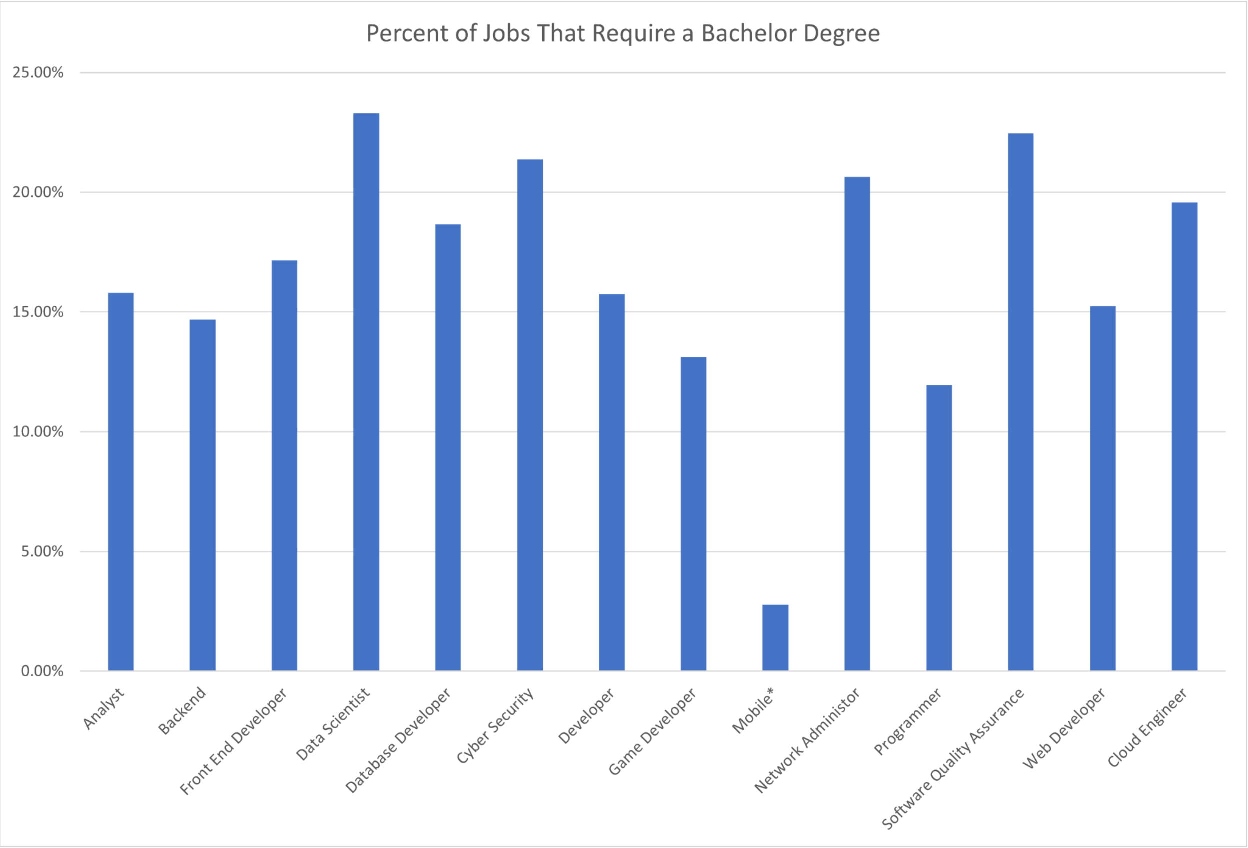
We tested the job scraper heavily before scraping the data we needed for the project. This included searching generic non-CS related job titles (Customer service, Clerk, etc.). We did this to make sure Beautiful Soup was scraping the correct HTML tags/classes. Once we knew BS was scraping the correct tags, we then continued to test it on CS related positions. From here, we scraped the job titles that were related to the project's job titles.

To test the languageCounter.py file, we first made a bag of words of programming languages to count the occurrences and percentages that these languages come up with in each job title. Once the correct stemming and processing algorithm was inplace, we tested it on data from previous assignments in the class. When it was all up and running, we used the scraped job dataset found in ./logs/* to perform the algorithm. These processed files can be found in the 'logs' folder.

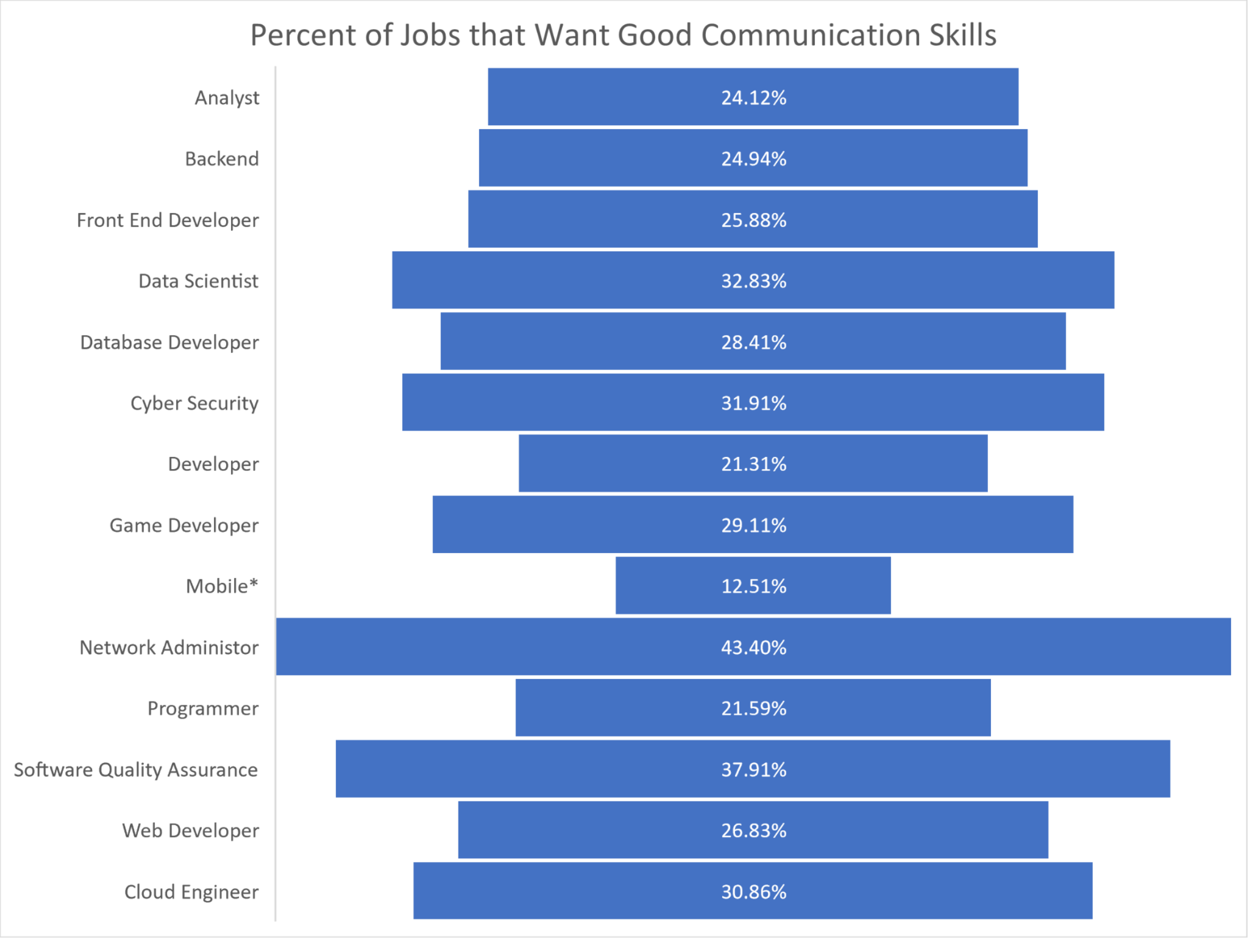
The Ngrams.py file was very similar to assignment 5, where we collected ngrams of the provided dataset. Testing ngrams was done lightly as we used some previous code from that assignment to create the ngrams from the processed data folder.

Medoids.py was tested on the processedLogs folder. We tested this on the data set from Assignment 5 as we knew that algorithm performed correctly. Once we saw it worked on a dataset we had prior knowledge of, we performed the algorithm on the preprocessed data, resulting in the dendrogram found above.

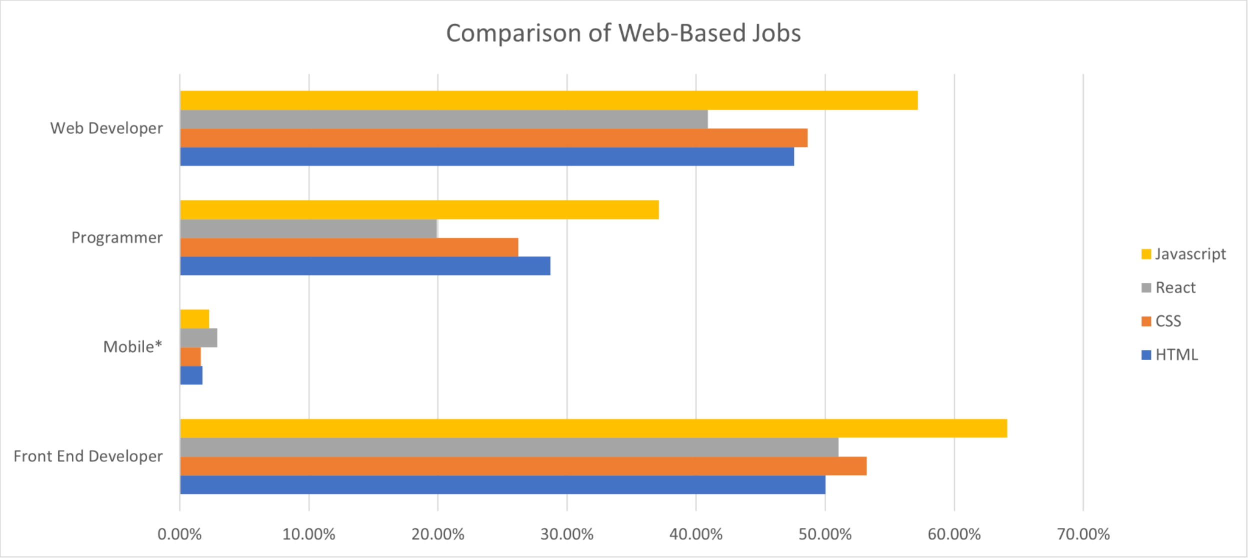
Data Visualization and Interpretation



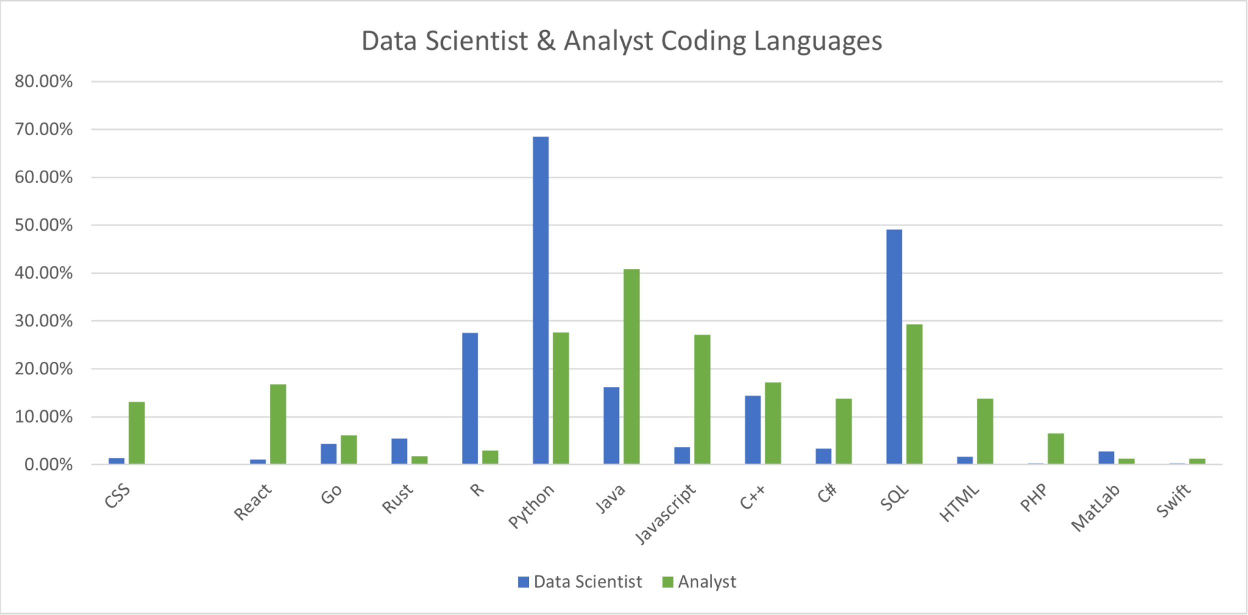
From this chart we can see that both Software Quality Assurance and Data Scientists jobs require the highest chance that candidates have a Bachelor's Degree over the other job titles. Shortly behind these two Cyber Security and Network Administrators have the highest percent chance that the job posting requires a Bachelor's Degree. Interestingly, Mobile developers have the lowest chance that the job requires a degree but this could be due to the dataset being skewed because of all of the french job postings.



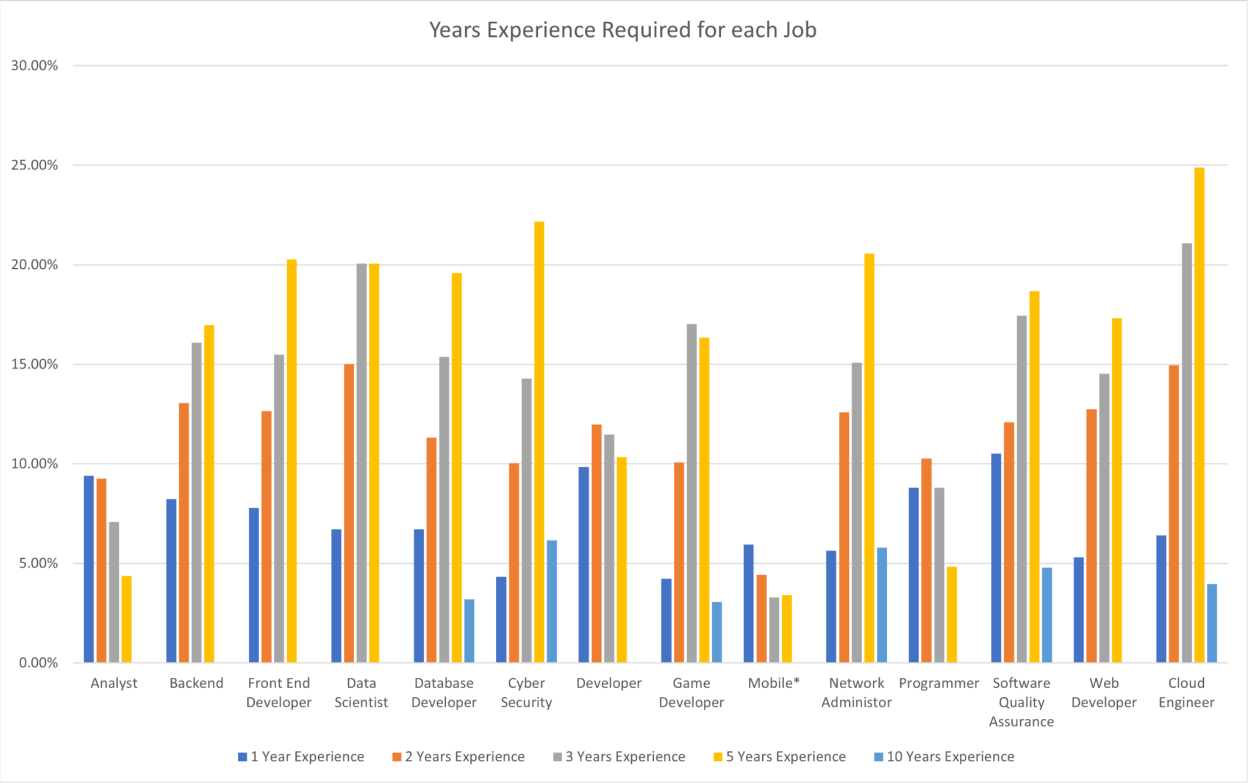
Communication skills are important across all fields of computer science. We can see that almost all the graphs have similar amounts of requested communication skills. Similar to before, Software Quality Assurance and Network Administrators have the highest chance that this skill is necessary. The rest of the job postings have a similar percent chance that communication skills are necessary.



Across all the frontend or web developer positions, JavaScript is at the top of the ladder, with HTML/CSS close behind. React is a very popular framework these days, where if a job asks for JavaScript, 10-15% of the time they also ask for React.



From here, we can see almost all the languages are used to some extent in these job postings. Python is largely ahead with SQL about 20% lower than Python. R comes in at 3rd among the most requested job postings. Interestingly, SQL is much higher than we originally thought for these postings.



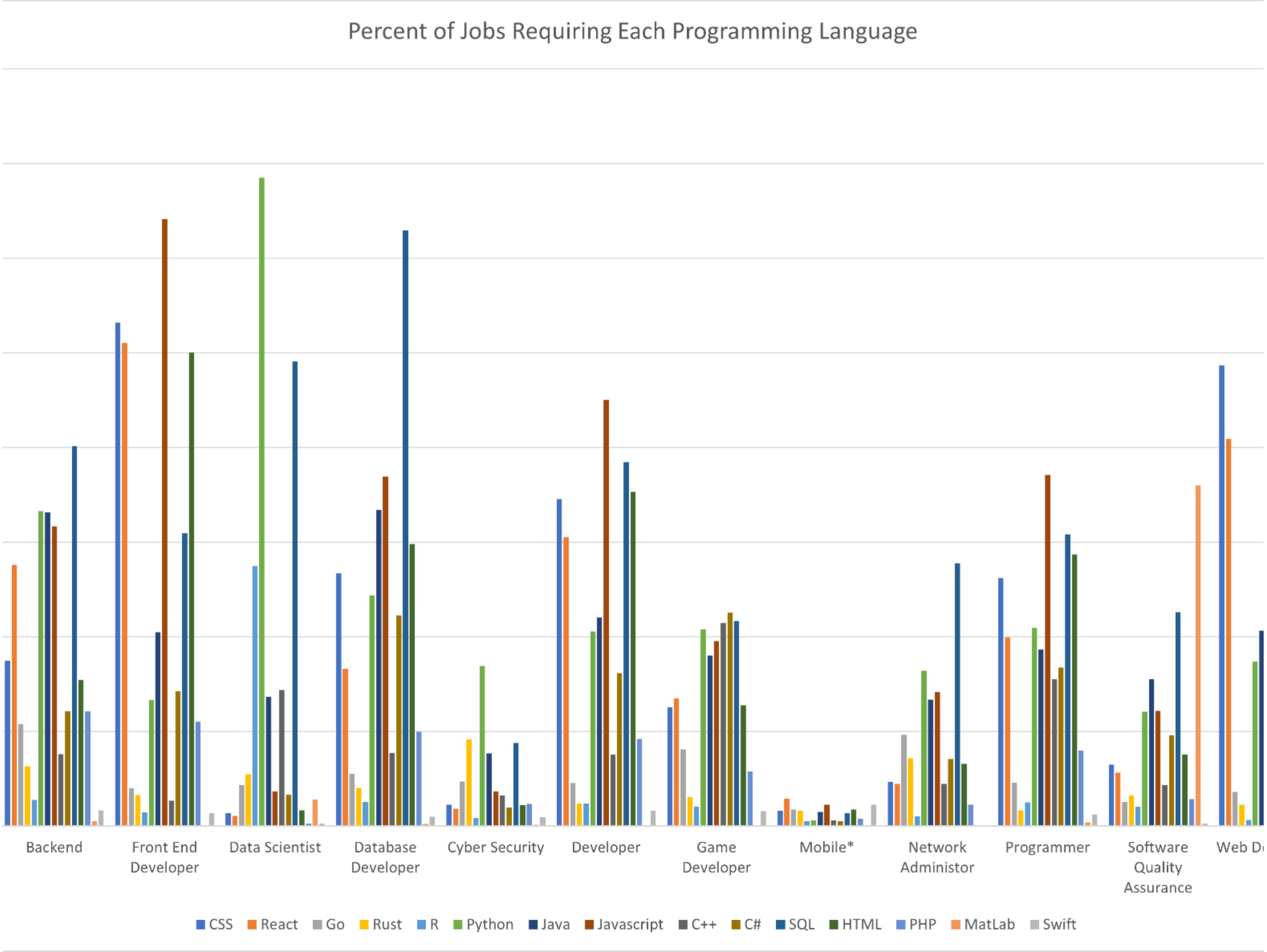
From this graph we see Web Developers have the highest chance for junior developers (1-year of experience). This means that if a person is interested in breaking into the programming field, they have the highest chance to land a job with a web development company. Closely behind Web Development is Game Developers.

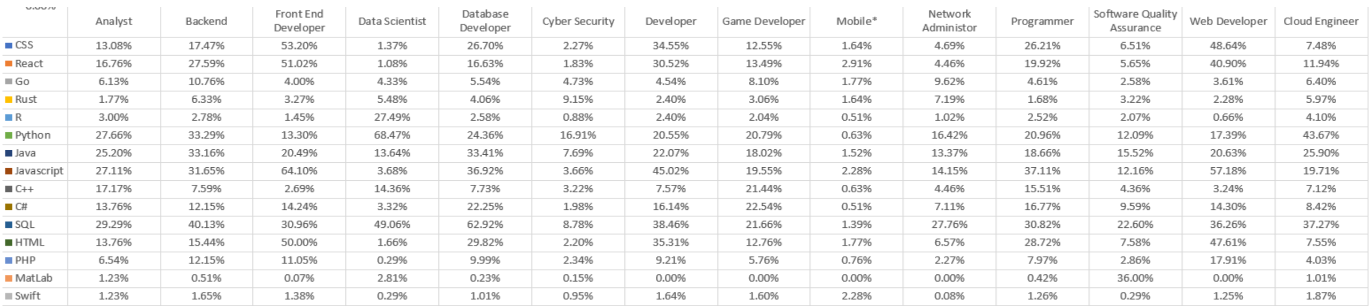
From the non-generic job titles (Programmer, Developer) we can see that Data Scientist has the greatest chance of requiring 5+ years of experience.

Database developers have the same amount of 3 years experience as 5 years. This could mean that the longer you are in this field, does not necessarily mean more opportunities.

The rest of the jobs are fairly equal with 1,2,3,5, years of experience increasing in that order.

We can see, apart from the generic job titles, Cyber Security and Web Developer titles are the only ones that request 10+ years of experience. This shows that even with over a decade of experience, your skills are still in great demand.





	Analyst	Backend	Front End Developer	Data Scientist	Database Developer	Cyber Security	Developer	Game Developer	Mobile*	Network Administor	Programmer	Software Quality Assurance	Web Developer	Cloud Engineer
■ CSS	13.08%	17.47%	53.20%	1.37%	26.70%	2.27%	34.55%	12.55%	1.64%	4.69%	26.21%	6.51%	48.64%	7.48%
■ React	16.76%	27.59%	51.02%	1.08%	16.63%	1.83%	30.52%	13.49%	2.91%	4.46%	19.92%	5.65%	40.90%	11.94%
■ Go	6.13%	10.76%	4.00%	4.33%	5.54%	4.73%	4.54%	8.10%	1.77%	9.62%	4.61%	2.58%	3.61%	6.40%
■ Rust	1.77%	6.33%	3.27%	5.48%	4.06%	9.15%	2.40%	3.06%	1.64%	7.19%	1.68%	3.22%	2.28%	5.97%
■ R	3.00%	2.78%	1.45%	27.49%	2.58%	0.88%	2.40%	2.04%	0.51%	1.02%	2.52%	2.07%	0.66%	4.10%
■ Python	27.66%	33.29%	13.30%	68.47%	24.36%	16.91%	20.55%	20.79%	0.63%	16.42%	20.96%	12.09%	17.39%	43.67%
■ Java	25.20%	33.16%	20.49%	13.64%	33.41%	7.69%	22.07%	18.02%	1.52%	13.37%	18.66%	15.52%	20.63%	25.90%
■ Javascript	27.11%	31.65%	64.10%	3.68%	36.92%	3.66%	45.02%	19.55%	2.28%	14.15%	37.11%	12.16%	57.18%	19.71%
■ C++	17.17%	7.59%	2.69%	14.36%	7.73%	3.22%	7.57%	21.44%	0.63%	4.46%	15.51%	4.36%	3.24%	7.12%
■ C#	13.76%	12.15%	14.24%	3.32%	22.25%	1.98%	16.14%	22.54%	0.51%	7.11%	16.77%	9.59%	14.30%	8.42%
■ SQL	29.29%	40.13%	30.96%	49.06%	62.92%	8.78%	38.46%	21.66%	1.39%	27.76%	30.82%	22.60%	36.26%	37.27%
■ HTML	13.76%	15.44%	50.00%	1.66%	29.82%	2.20%	35.31%	12.76%	1.77%	6.57%	28.72%	7.58%	47.61%	7.55%
■ PHP	6.54%	12.15%	11.05%	0.29%	9.99%	2.34%	9.21%	5.76%	0.76%	2.27%	7.97%	2.86%	17.91%	4.03%
■ MatLab	1.23%	0.51%	0.07%	2.81%	0.23%	0.15%	0.00%	0.00%	0.00%	0.00%	0.42%	36.00%	0.00%	1.01%
■ Swift	1.23%	1.65%	1.38%	0.29%	1.01%	0.95%	1.64%	1.60%	2.28%	0.08%	1.26%	0.29%	1.25%	1.87%

This graph is the most cumulative graph of them all. It shows our predetermined skillset (programming languages) and which jobs have the highest needs for those skills. Examining this dataset, across all job titles, Java is the most requested job skill. One common thing we came across when viewing job skills needed is that candidates have an OOP language knowledge. For example, this sentence comes across very often: "Knowledge of an OOP language. Example: Java". This may be why Java is very high across all roles.

We can see that SQL is highest in Database developers, and close behind in Data Scientist positions. Where in Data Scientist positions, Python is the most requested skill. Interestingly, SQL is quite high across all job fields,

Front End developers had the longest list of required skills with HTML, SQL, JavaScript, and Java all over 30% of postings. This means that candidates generally need a breadth of knowledge to fulfill the job posting's needs. These skills were heavily mirrored with Web Developers skills.

Game Developers also had a bunch of skills, though there was not a skill that was massively ahead of the others. This may be because we did not check for video game specific tools such as Unity and Unreal Engine.

Conclusions:

Overall, this project introduced us to a variety of skills such as web scraping with Beautiful Soup, data analysis, optimizing algorithms, and data visualization. From the data we gathered, users can see which skills are required to get a job in a specific field of interest. Using this knowledge, the user can learn those required skills, build some projects, and show it off on their resume to help their chances of landing a job. Some fields require much more depth of knowledge than others, while some fields require more skills than others.