**Using Natural Language Processing (NLP) to**

**Analyze Technology Job Postings**

Final Project by: Hikaru Sugimori & Arvind Bhaskar

Date: 12/13/2019

Class: Data Analysis with R and Python (Bus 32100-85, Autumn 2019)

# Abstract

The central question asked by this project was: “How do you get a job in tech?” In order to answer this question, we developed various natural language processing (NLP) methods in Python to analyze over 22,000 U.S. technology job postings. Using our NLP methods, we were able to recognize patterns and derive meaningful conclusions from the jobs data.

# Introduction

NLP is an area of computer science and artificial intelligence (AI) that deals with the interactions between computers and human (natural) languages, in particular how computers process and analyze large amounts of natural language data. Successful implementations of NLP include search engines like Google, Facebook news feeds, Siri, and spam filters.

In our view, Python was the ideal language to do NLP as Python has powerful open source NLP platforms such as The Natural Language Toolkit (NLTK), which is a suite of libraries and programs for symbolic and statistical NLP for English. It contains text processing libraries for tokenization, parsing, classification, stemming, tagging, and semantic reasoning.

# Dice Dataset

As stated in the abstract, we wanted to figure out how one could get a job in tech. In order to answer this, the natural dataset for us to look at was a dataset of tech jobs.

Kaggle had a pre-crawled dataset, taken as a subset of a larger dataset of more than 4.6 million job listings that was created by extracting data from Dice.com, a prominent US-based technology job board. This dataset had columns for the following: advertiser url, company, employment type, job description, job location, job title, post date, shift, and skills.

We began our project by “playing with” this dataset to get an intuitive feeling for what it contained. The following were our findings:

* The typical job description column contained 307 words.
* The job description with the most words contained 3,455 words.
* The words with the highest term frequency in the job description column was “security”. Term frequency is defined as the number of times a term appears divided by the total number of terms.
* The max character count was 24,382, a job posting for Selenium engineers.

The column of primary interest to us was the “job description" column as that is where all the text and data was.

The first challenge that confronted us in trying to analyze this column was the fact that computers cannot “comprehend” language the way we can. Reading comprehension for literate people is second nature, and we do it without really knowing how we do it, but for computers, we have to clearly define what comprehension is!

The first thing we did was to convert the data into a format that would be amenable to analytics (in NLP this is sometimes called pre-processing). We then separated the noise from the signal. In practical terms, we had to clean the data.

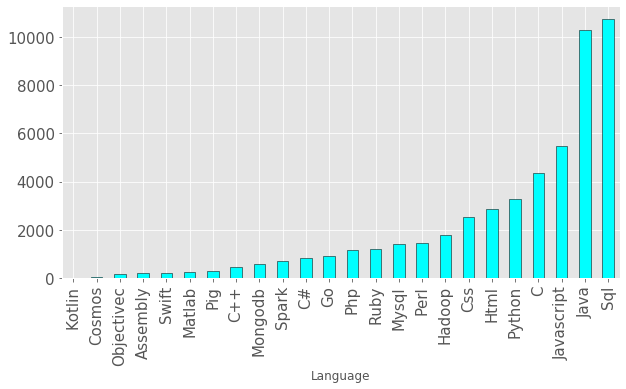
The end-goal we had in mind when cleaning the data was to get to a point where we can see the most frequent meaningful words that appeared in the data. Therefore, we had to clean the data in a way that we could best achieve this goal.

The following are the steps we took:

1. I noticed that some of the columns in our dataset were not strings so I used the astype(str) command to turn columns into data type string if they weren’t strings already so that out NLP methods could be applied.
2. Turned all the letters into lowercase because our goal was to identify the most frequent words we didn’t want to count “Programming” and “programming” as separate unique words.
3. Removed punctuation by replacing “[^\w\s]” with a blank space.
4. Removed stop words. In NLP, stop words are frequently occurring words that add very little meaning, such as "and", "the", "a", and "an". Creating our own list of stop words would have been impractical; fortunately, the NLTK platform already has a list of stopwords in 16 different languages we could use to remove the stop words.
5. Turned the sentences into a series of separate words and counted the most frequently occurring words.
6. We then compressed steps 1~5 into a single function that cleans entire columns of text and outputs the most frequent words. We called this function the cool function (special thanks to PhD student Andrew Borowick for writing the cool function).
7. Even after all this cleaning sometimes the first couple words are overly generic quasi stop words that don’t give us that much insight into the data. In the case of the job description column, these words include, “experience”, “business”, “work”, “strong”, etc. For this reason, we modified Andrew’s cool function so that we can specify the range of top words the cool function will output. We added parameters a and b, where a is the rank of the first top word the cool function will output, and b is the rank of the last top word the cool function will output. For example, if I gave the cool function a = 50 and b = 100, the cool function would return the top 50~100 frequently occurring words.

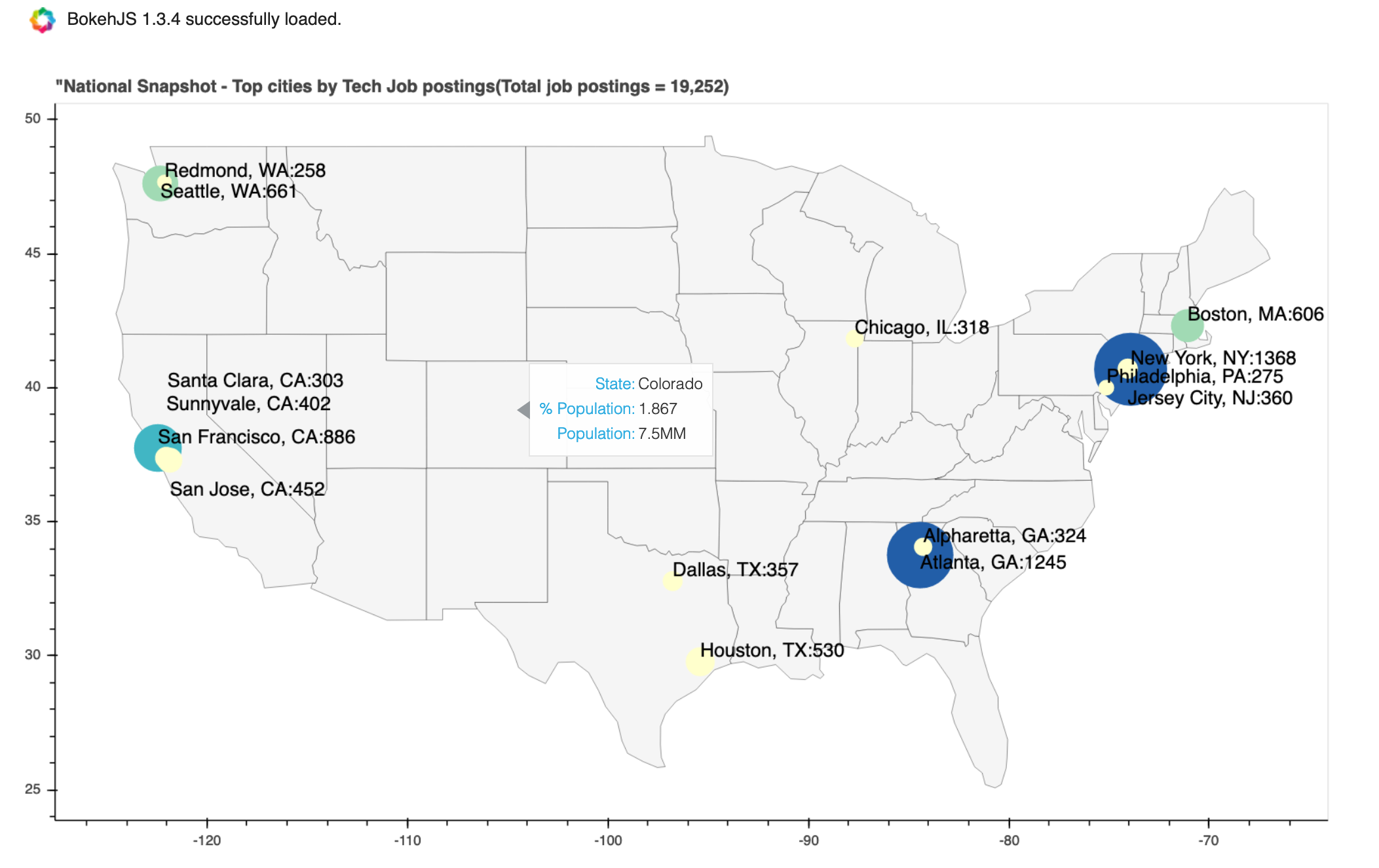
We soon realized that the cool function was a very powerful tool as we could easily comb through 22,000 rows of text and immediately recognize patterns. The following are some interesting insights that we gained from applying the function and other NLP methods across various columns in our dataset:

* Although the first 50 words of the job description column were mostly generic quasi stop words, there were somewhat notable words such as "data", "software", "security", "web", and "application".
* SQL and Java are in the top 100 words in job description. In the skills column, these two languages are #2 and #3 respectively. This speaks to the popularity of these languages. It is worth noting however, that this dataset is from two years ago and the landscape has probably changed now with the ascent of Python in recent years.

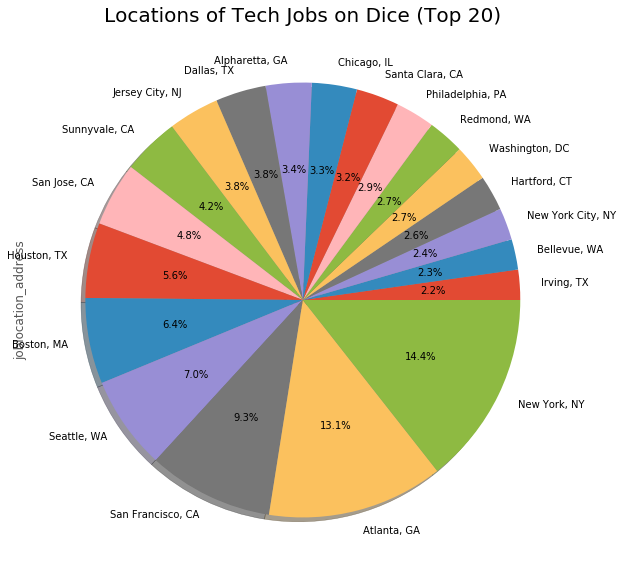


Top Programming Lang. in Dice Dataset 1

* Looking at the job location column, California is by far the state with the most tech jobs, followed by New York. New York City was the city with the most job postings, followed by Atlanta, and San Francisco. The top fifteen cities account for nearly 43% (8,345 jobs) of total jobs postings (total 19,252 jobs postings) on Dice for tech workers, or about 1 in 2 jobs listings in the nation. Out of Dice dataset of 22,000 job postings, 19,252 had a valid location specified. In fact, 43% of job postings are in 13 metropolitan areas, if Redmond, WA & Alpharetta, GA are considered part of Seattle, WA and Atlanta, GA respectively. The Atlanta area (1,569 postings) followed by New York City was the largest metropolitan area for tech. Silicon Valley continues to be a critically important hub for tech companies. San Francisco, San Jose, Santa Carla & Sunnyvale account for 2,043 job postings (>10%). The discussion doesn’t end there, however, as technology increasingly has a significant presence across the nation. Cities such as Boston, Seattle, New York, Dallas, Atlanta and Chicago, also have strong demand for tech talent.

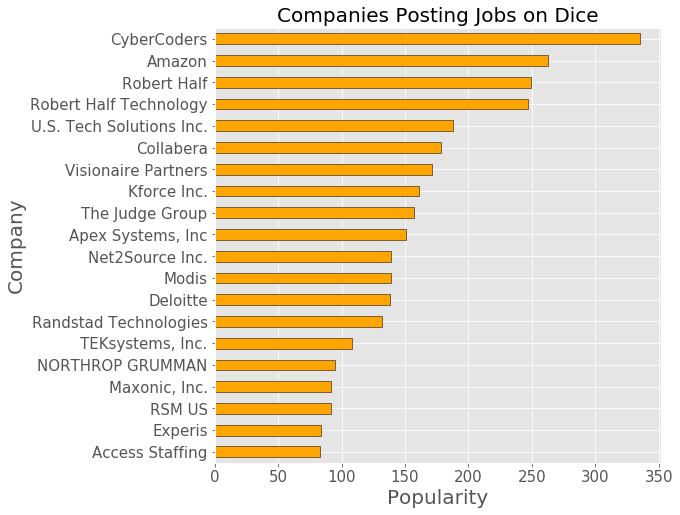


Top Cities by Tech Job Postings 1

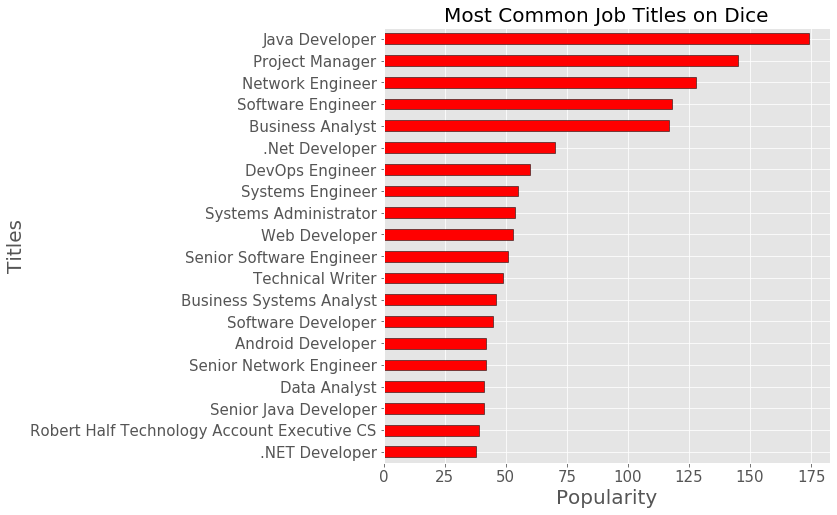


Top Cities by Tech Job Postings 2

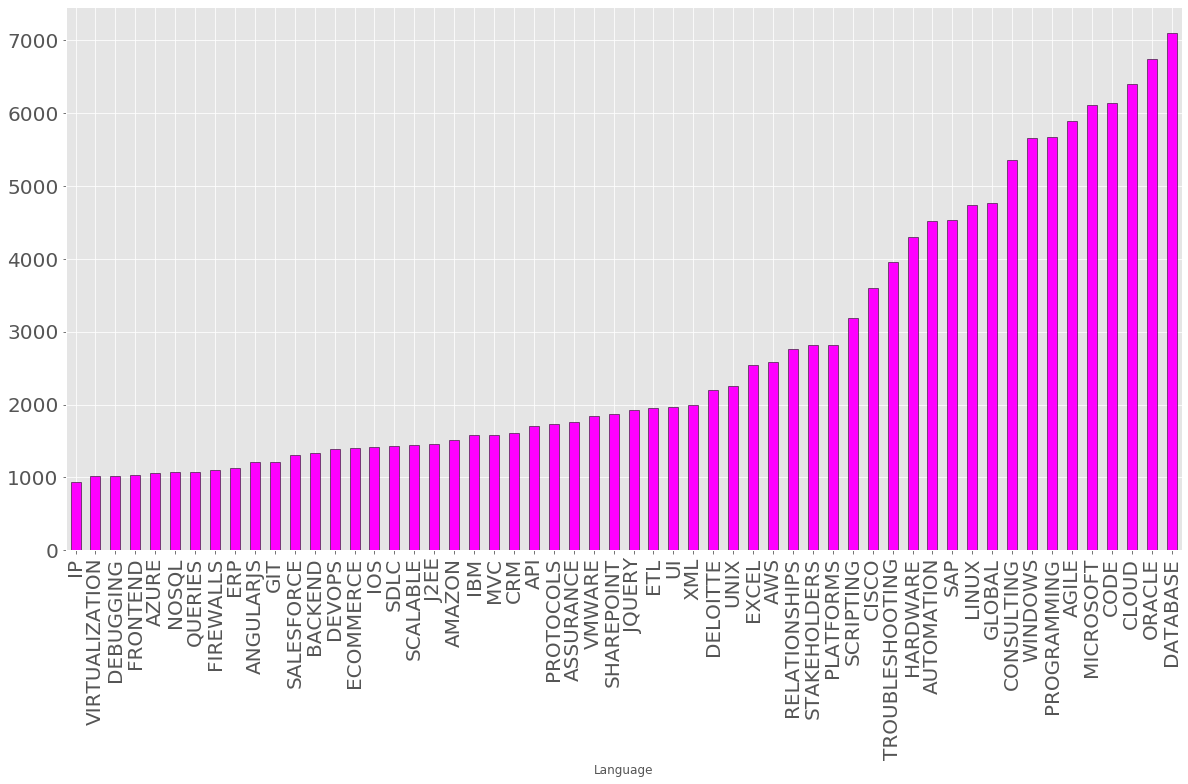
* Amazon seems to be the biggest tech employer on Dice; it's the first tech company that comes up. There are also staffing companies like CyberCoders, Robert Half and Kforce sourcing talent on behalf of other companies. Deloitte is also a top hit which is surprising because many people think of it more as an accounting/consulting company. For Amazon, notable top words include, “aws”, “customer”, “audible”, and “cloud”. For Deloitte, notable words include, “sap” , and “consulting”.



* The top five job titles were Java Developer, Project Manager, Network Engineer, Software Engineer, and Business Analyst.



* The most common employment type was full time.
* These jobs were all posted within the same 4-week period. This was somewhat unfortunate because we will not be able to do a time series showing the shift in language patterns over time.
* Other notable words from the Dice dataset include the following:



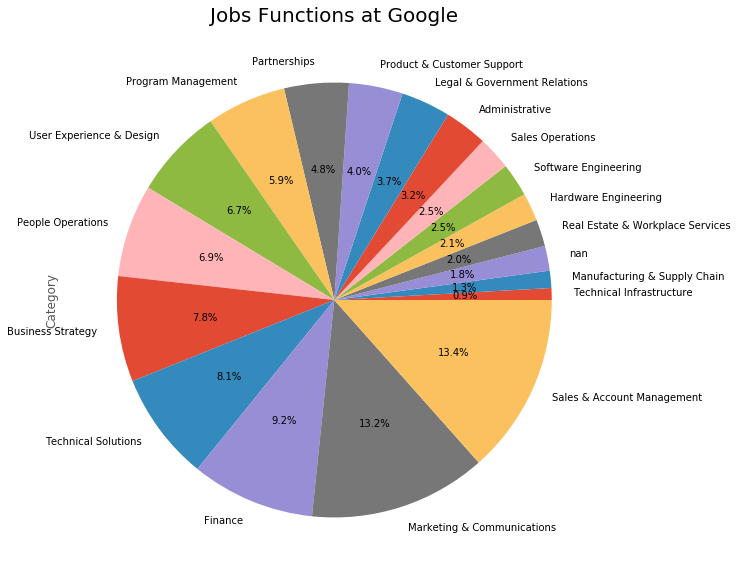
Sometimes merely using the value\_counts() function was more appropriate than blindly applying the cool function everywhere. For example, if you wanted to find the cities that are the most popular using the job location column, “New York, NY” should be identified as a single unit; we do not want to separate these into “New”, “York” and “NY” for obvious reasons (note: the cool function was appropriate for identifying the top state). In this case, we want “phrase frequency” rather than “word frequency”. For the same reason, we just used the value\_counts() function to analyze the company and job title columns.

**Google Dataset**

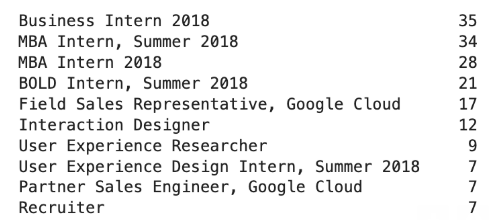
Although we were expected to just analyze one dataset for this project, once we finished applying our NLP methods to the Dice dataset, we were very eager to apply our methods beyond a single dataset, and this is exactly what we did. On Kaggle, we found a dataset of jobs postings at Google. The columns for this dataset were very similar to the Dice dataset (job title, job location, job responsibilities, minimum and preferred qualifications). This dataset seemed interesting as Google is at the cutting edge of the tech industry. Google dataset we analyzed contains 1,250 rows of data.

The following are some of our findings from the Google dataset:

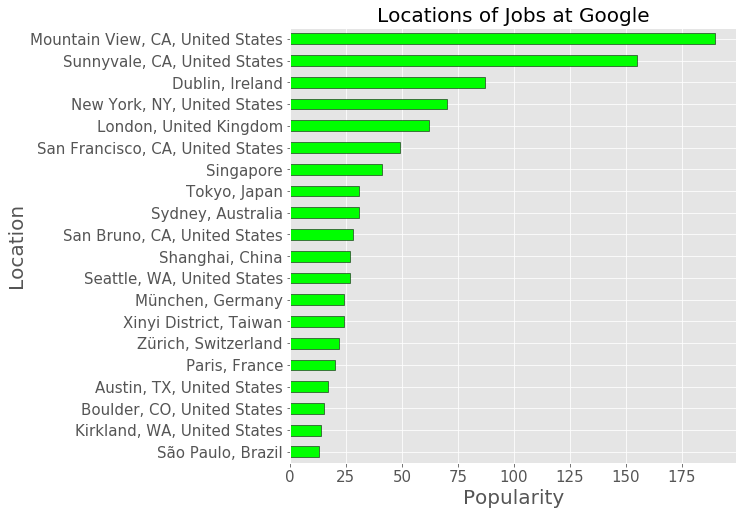
* Google has lots of non-technical jobs (or at least they do not sound technical). In the job category column, the top 6 categories included: Sales & Account Management, Marketing & Communications, Finance, Business Strategy, and People Operations.

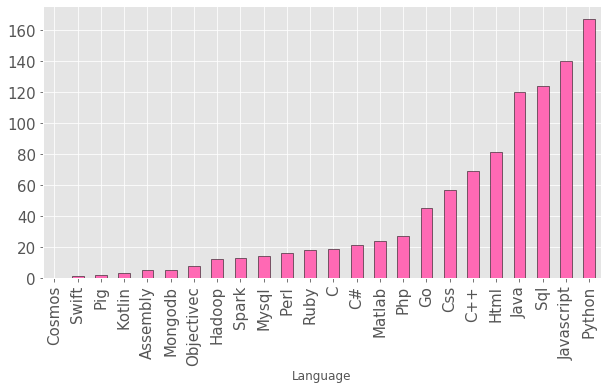


* There are lots of opportunities for business school grads! The following are among the top four job titles: “Business Intern 2018”, “MBA Intern Summer 2018”, and “MBA Intern 2018”. *See below.*



* There are lots of words relating to interpersonal skills, such as “teams”, “relationships”, “communicate”, “speak”, “idiomatically”, and “verbal”.
* The Google jobs were in California, but they also had a sizable portion of jobs in Dublin, NYC, London, Tokyo, and Sydney.



* Python is the most popular programming language at Google! What we learned in class is relevant! 

In order to arrive at this conclusion that Python is the most popular programming language at Google, we introduced another NLP method. Previously, with the cool function, we were “passive readers” of the text, whatever the most frequent meaningful word was, the cool function counted and ranked them. This time we wanted to be active readers. We defined what the programming language words were and then did a search of three different columns to count those words.

# Suggestions for Future Research

There were many more things we were interested in doing but we were unable to get to due to time constraints. These include the following:

* Add an accurate compensation column to the dataset and do a regression to see what words are predictive of higher compensation. We attempted to do this, but we did not have enough time.
* Do a time series spanning several years to see the changes in language over time. As aforementioned, this was not possible for us as our datasets only has jobs from the same year.
* Subset datasets by company (beyond just Amazon and Deloitte), geography, function, etc. and see how these impact the frequently occurring words.
* Create other visualizations of our data such as word clouds. We attempted to do this as well but had issues downloading the word cloud library.

All of these are very interesting topics and we encourage anyone interested in NLP to look into these topics.

# Conclusion

NLP is only beginning to come into its own. Despite only being beginners in Python, let alone NLP, we have already palpably felt its benefits as well as its potential. Through NLP we were able to gain valuable insights into how one could get a job in tech by rapidly combing through 22,000 job descriptions, allowing us to easily recognize patterns in the data. Needless to say, we are very excited about the future of NLP.

# 