04/21/21

Project 2 Write-Up

**Overview**

In this project, we were given the general task of data visualization through D3 or Plotly. I chose the former as I found it to be more compelling and had originally wanted to include a word cloud (more on why that did not work later). I decided to focus the work on attempting to collage work together in order to create an app which takes live tweets and gives the user interactivity as they are able to type in the key word or hashtag they would like to see the most popular tweets of from the past 6 days.

**Tools**

I used JavaScript, PHP, an API through Twitter and the Tweepy library, D3, and Bootstrap.

**Method and Limitations**

It was an incredibly involved process to piece together a way in which to funnel live tweets through PHP without prior exposure. I found a twitter sentiment analysis github page (Benson Ruan’s, license enclosed) which used PHP in order to query tweets and took that as a model. I then tweaked it until it suited my needs, having to learn a lot about PHP in the process. This took the majority of my time, as I had to grasp tweepy endpoints, PHP, navigate through the prior connections between JS and PHP, and debug as well as recreate my own.

Then, I decided to make a dynamic scatter plot with D3 to show the relationship between likes and followers of tweets and their users for the most popular tweets from the past 6 days including the key word (which I had designed the twitter query to do). I then included another button and event listener to do the same to add another layer of interactivity – that way, the user would be able to compare and contrast and examine the data more broadly. For example, one could see the broad popularity of tweets with “Pfizer” versus those with “Moderna”, and they could see how much of a role users play.

I also spent an enormous amount of time trying to include a wordblob. First, I mapped and reduced the tweets from the data to isolate the words. Then, I organized them by count. Then, I included an array of words not to be included in the wordblob. However, I ran into an issue when I needed to have a way to include npm on the app. I didn’t want to create a new page on the front-end of my app to allow this, so I tried to go through multiple layers of source code to manually add the wordblob to the document. This was in v3 instead of v4, the one I was using, so I then had to comb through each specific calling item. Some were impossible to replicate by hand with my skill level, so I then tried to add the entirety of the dispatch source code to my document. This was also a bit too hefty for my current ability, so I put it aside.

**Further work**

I would like to finish the wordblob and do some more work on correlation. Since I did this project myself, it was incredibly involved already, but I would have liked to have included more graphics. I also would like to be able to geolocate the tweets and explore them further.