Sentiment Analysis:

1. First I ran the tweetsentiment.py with the inputs of AFINN-111.txt and output.txt, requesting it to output to the file newoutpt.txt My output was:

-5 sentiments 24

-4 sentiments 155

-3 sentiments 223

-2 sentiments 344

-1 sentiments 522

0 sentiments 0

1 sentiments 322

2 sentiments 532

3 sentiments 380

4 sentiments 101

5 sentiments 3

Weighted Mean Calculation:

5\*3+101\*4+380\*3+532\*2+322\*1+0\*0+522\*-1+344\*-2+223\*-3+155\*-4+24\*-5 = 317 / (24+155+223+344+544+322+532+380+10+3)

Mean = 0.12509593246354567

The mean of sentiment analysis of my Twitter data is 0.1259 which means that the data I pulled is slightly more positive than neutral, essentially saying that I pulled as many positive tweets as negative. This type of analysis seems to be kind subjective because I randomly pulled data at no real given time interval. I guess if you timed the pulling of the data, you could take random samples of twitter streams to get the overall sentiment from the internet? But then, I don’t understand where the actual tweets came from, some many these would not be representative of the twitter world. Where these just the tweets at the time of me pulling the data?

If the AFINN-111 text had a different number of words then my analysis may be different but hard to say how because it would depend on the words added and the words pulled from the Twitter data. For example, if I added more negative words and those came up in my Twitter data, then my sentiment could be more negative.

A way to compensate with the AFINN data is to make sure you consistently use the same file to do your analysis. You wouldn’t want to use AFINN-111 this time and compare it to a previous version for another set of tweets because they you wouldn’t be comparing apples to apples.

Sarcasm is a problem because the implied tone of the message is not what that straight AFINN meaning would be. For exactly if you said, “Well *that* was great” in a sarcastic tone, you are clearly not conveying a positive emotion but the analysis of great in AFINN would give it a positive score swaying the analysis towards a wrong sentiment. I don’t know how text and sentiment analysis accounts for sarcasm. I imagine this is hard to do.

Further Analysis of Tweets:

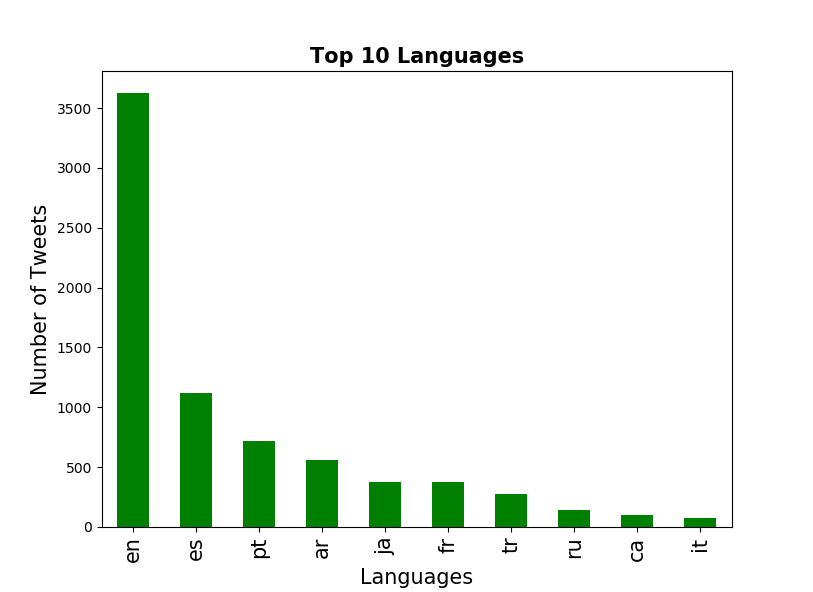
1. First I installed the following on my command prompt

pip install -U pip setuptools

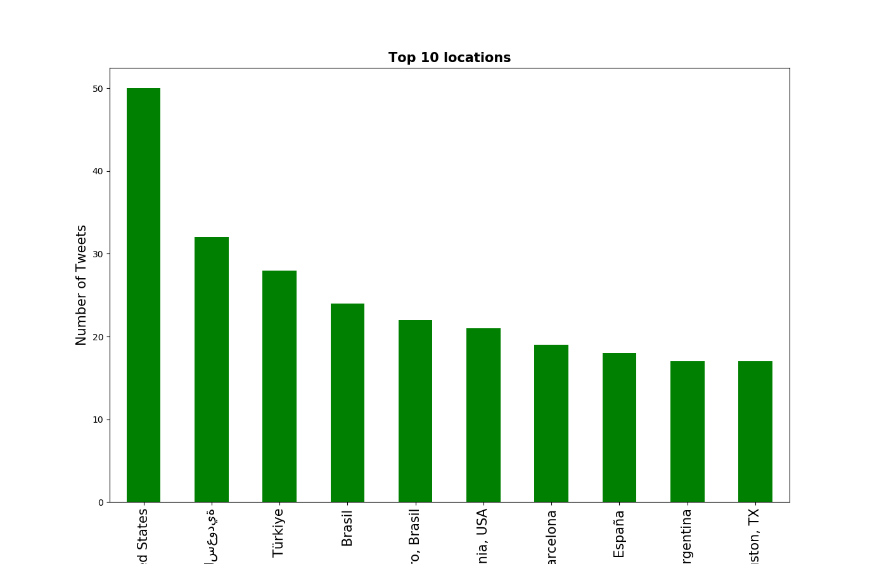
pip install matplotlib

pip install pandas

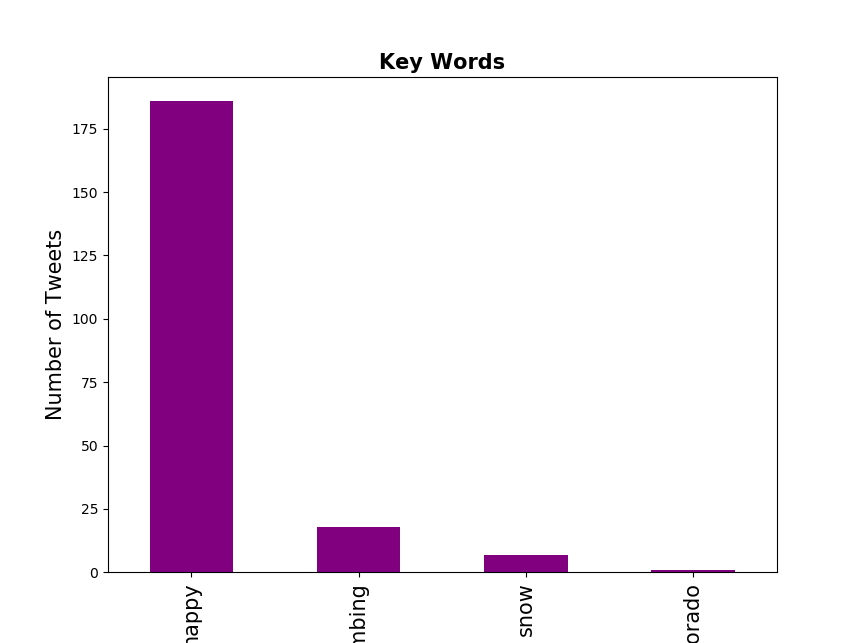
1. Next, I ran twitterstream\_output\_languages3.py which uses my req\_output.json file that I created last week. The result of this was the following chart.



1. Analysis: The top language by far for the tweets I pulled was English. From there is Spanish, Portuguese, Arabic and Japanese.
2. Next, I ran twitterstream\_output\_locations.py in my console. The output was the following:



1. The top location was the United States and the second was an Arabic country. What is confusing is that the top language above was English which makes sense that the top location would be the United States, but the next most tweets were Spanish and Portuguese, but the top location is not Spain or Portugal. Does this mean that people in Arabic countries are tweeting in Spanish and Portuguese or did I miss something?
2. Lastly I ran the key\_words.py after changing them to things like snow, Colorado, and rock climbing. My output is below:



1. If I included the words that interested me in my AFINN sentiment analysis, then depending on the words I picked it could have shifted the sentiment. I’m not 100% sure how picking keywords here would truly affect the sentiment analysis since I was just reading ALL of the words in the Tweets for that, while in this example I am just displaying the keywords that are most interesting to me. My keywords are pretty generic, but maybe if it was a snowy week in the United States, that could have skewed the number of people tweeting about that. I imagine snow is a neutral word, so it may have evened out my analysis. I used the word happy which is positive but generic. It would be interesting to do this after a terrible event and see if less people are tweeting about being happy, but the issue I see with this is that this last chart produces a NUMBER of tweets, which would depend on how long I ran my initial python file. In order for this analysis to be a little more rigorous, or if I wanted to actually use this last chart, I would have to figure out how to work with the same amount of data (tweets).