Project Overview: For my project, I downloaded my own tweets and analyzed them. In my analysis, I found and printed the number of tweets I've liked, a histogram of the words in my most recent 199 tweets, and the general sentiment of my most recent 199 tweets, found with NLTK's VADER Sentiment Intensity Analyzer[1]. Using this data, I hoped to infer the topics I tweet about most and whether my tweets tend to be positive or negative. Finding the number of tweets I've liked was just for fun, rather than for real analytical purposes, since I wanted to be able to calculate how many tweets I like on average per day.

Implementation: First, the script downloads a given user's (in this case, my) most recent 199 tweets into a text file. It also puts the current date (the date the tweets were retrieved) and the user's username at the top of the file. However, before doing any of this, the existing text file is analyzed. If the date at the top of the file is within two days of the current date and the username at the top of the file matches the username that the person running the script specified, the file is not updated because it is assumed to be up to date. To be clear, if the date in the file is more than two days ago or if the username does not match the inputted username, the text file is updated with the specified user's newest tweets.

After downloading the user's tweets, they are analyzed in a few different ways. Using Twitter's API, I was able to easily retrieve the number of tweets the user has liked in their history on Twitter. Using a few different functions more extensively described in the script's docstrings, I also create a histogram of the words in the text file containing the user's tweets. I also added an argument that limits the amount of times a word can exist in the text file in order to be "counted" when making the histogram. In practice, this means that words that only appear between one and nine times are not accounted for, making the histogram take up significantly less space in the script's output. Last, the script outputs the general sentiment of the words in the user's tweets based on a default VADER analysis.

One design decision I made was to use the python-twitter package instead of another package, like Tweepy, to simplify Twitter API calls in Python. I chose python-twitter because it was the recommended package on the assignment page, and even with some difficulty (specifically with using the Twitter API's User object), I decided to stick with it. Ultimately, I'm glad I made this decision, because python-twitter made the process of downloading tweets relatively simple and reliable.

Results: When I analyzed my own tweets, I made a number of interesting observations:

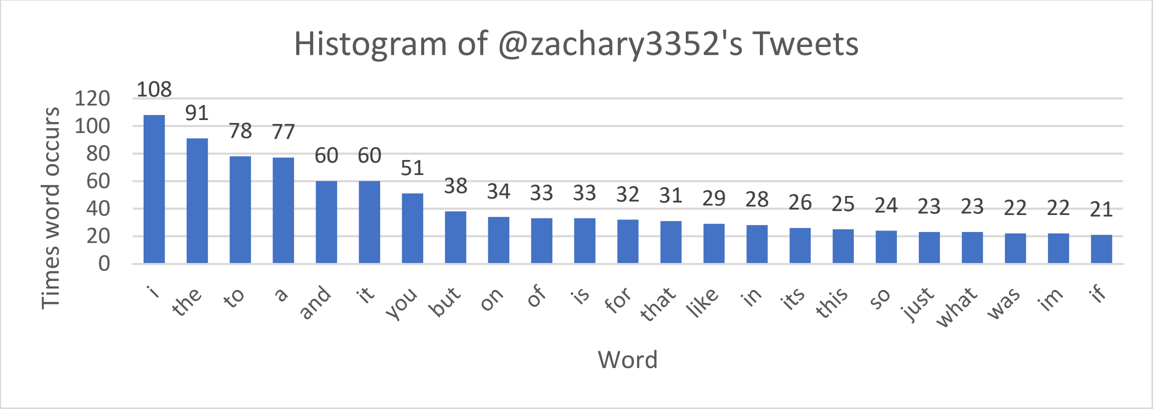
- Since joining Twitter in October 2015, I've liked 45,621 tweets. October 2015 was about 4.5 years ago, or about 1600 days ago. Using this data, I found that I like approximately 29 tweets on average per day. Maybe I should get out into the world occasionally…

- Here is a histogram of the words in my tweets:

```
[('i', 108), ('the', 91), ('to', 78), ('a', 77), ('and', 60), ('it', 60), ('you', 51), ('but', 38), ('on', 34), ('of',
33), ('is', 33), ('for', 32), ('that', 31), ('like', 29), ('in', 28), ('its', 26), ('this', 25), ('so', 24), ('just', 23),
('what', 23), ('was', 22), ('im', 22), ('if', 21), ('my', 19), ('have', 19), ('cool', 18), ('about', 17),
('your', 15), ('youre', 15), ('with', 15), ('be', 15), ('are', 14), ('they', 14), ('at', 13), ('an', 13), ('out',
13), ('not', 13), ('bashertech', 13), ('think', 12), ('or', 12), ('do', 12), ('really', 12), ('one', 11), ('me',
11), ('when', 11), ('as', 11)]
```

  A graph of this histogram, for words occurring more than 20 times, is shown below:

[1] https://www.nltk.org/_modules/nltk/sentiment/vader.html



- According to NLTK VADER, 23.9% of the words in my tweets are considered “positive”, while 5.2% are considered “negative” and 70.9% are considered “neutral”.

Alignment: I began this assignment with several questions: Which words do I tweet most? Which topics do I most often tweet about? How “happy” are my tweets, in general? How many tweets have I liked since joining Twitter, and about how many tweets do I like in an average day?

When visualizing this project, I imagined an answer to these questions might include a number of tweets I’ve liked, a list of topics I tweet about most frequently, and a percentage of “positive” and “negative” words in my tweets. Ultimately, my project contains some of these results, as well as a graphical histogram I didn’t expect to make. I found that the tools provided by the assignment page aligned very well with the analysis I wanted to perform on my tweets, which made finding answers to some of my questions easier than I expected. Overall, these tools worked well, and I feel confident in the results they rendered, assuming they themselves were correctly and reliably written. I have no reason to believe they weren’t!

Interestingly, the hypotheses I made internally don’t seem to align too closely with the results I actually found. For example, while I was hoping to use a histogram to figure out which subjects I tweet about most frequently, it turns out I can’t just use words to determine this. In reality, the words I tweet most often are (upon reflection, unsurprisingly) filler/helper words, such as “I”, “the”, and “a”. Obviously, these aren’t the “topics” I talk about most on Twitter! I believe my tweets would need to be analyzed much further to determine the topics I tweet about most. In addition, I expected a much larger percentage of the words in my tweets to be considered “positive” (maybe 40-50% compared to the 23.9% I actually found). I believe this is due to the relatively high number of words VADER doesn’t classify—70.9% of the words in my tweets!

Reflection: For the most part, this project went very well. I was able to complete my code a couple days before the project was due, and I didn’t have to spend hours and hours debugging. That said, I certainly encountered issues along the way (including figuring out how to use Twitter’s User object and reading/opening text files in Python), but NINJA hours were very helpful (thanks, Hyegi!). Overall, I really enjoyed learning more about my Twitter usage.

In the future, I could definitely improve my code by implementing docstrings and doctests earlier in the programming process. For this project, I implemented these after my code was written, which didn’t really help me (but certainly made my code clearer). I believe this project was appropriately scoped, because I was able to complete it in a reasonable amount of time and because it provided a good balance of fun and challenge. In the future, I’m excited to use the text opening/editing skills I learned from this project!