1. Write a python program (not a Jupyter notebook, but a py file you run from the command line) that accepts the cats\_txt.txt file as input and counts the frequency of all words and punctuation in that text file, ordered by frequency. Make sure to handle capital and lowercase versions of words and count them together.
2. Document how to run the program you created in question 1 in a readme.md file in your repo. Be as clear as possible. Use proper markdown, and consider using screenshots. Be sure to briefly discuss why this kind of exercise might be helpful for NLP in your markdown.

In a Jupyter notebook:

1. Load the tokenized Paradise Lost from the Gutenberg Corpus in NLTK. <https://www.nltk.org/book/ch02.html> . Stem or lemmatize the words and find counts. Select the top 20 words and create a histogram. Exclude stop words and make sure you are including words of all capitalizations in your count. If there are any meaningless “words” that are produced in your list or top words, alter your logic to exclude them. Specify why you chose stemming or lemmatization. \*
2. Perform Vader Sentiment Analysis on the book. Find the 5 most negative, 5 most positive, and 5 most neutral sentences in Paradise Lost. <http://www.nltk.org/howto/sentiment.html> . This may take a while to run, so you can always start with a small subset of the data (100 sentences) and then once your code works as expected, expand it to the whole book and let it run. \*
3. Explain your findings from the previous question. Are the sentences and their sentiment analysis scores correct? Explain why or why not.

\* If this is taking a long time to run on your computer, you can run only 10,000 sentences. But please try to do it with the whole text before you give up. One of my computers can handle it in a timely manner, the other not so much.