CSci 343 Fundamentals of Data Science
Challenge 2

Submission Window Opens:
Friday, February 28

Points Available:
200 XP for a working demonstration
50 XP for correct submission & understandable code

Objectives:

- Implement basic sentiment analysis
- Familiarize yourself with MatPlotLib by visualizing your results

Assignment:

You have been contracted by a television network's piloting department. They are interested in getting in on the cutting edge of television by using data science to determine what makes a TV series a hit or a miss. They hypothesize that that happier, more positively written scripts attract more viewers. Being a data scientist, you are skeptical and decided to test their hypothesis by comparing the average sentiment scores of scripts from two TV series. One TV series was somewhat popular (series A) while the other was wildly popular (series B). Determine if these data supports their hypothesis.

You have been provided with a corpus of data. Specifically, the network has given you transcripts of each episode from the first season of each TV series. These are given to you as text files with names of the form a###script.txt for Series A and b###script.txt for Series B. You are to analyze the sentiment of the wording used in these scripts using the provided sentiment lexicon.

On the class data website, I have provided you with a sentiment lexicon based on the efforts of the folks over at the Natural Language Processing group at Stanford University. This lexicon is a CSV file with words in the first column and their sentiment scores in the second column. The sentiment score ranges from -1.0 to +1.0 (negative sentiment to positive sentiment).

Your task is to analyze the Challenge 2 data files using sentiment analysis. You will need to first make a list of all unique words in each script and count how many times each appears. Using the sentiment lexicon, you

will need to determine how many of the words have a positive or negative sentiment. Words that are more neutral typically appear much more often than very polarized words. As such, it is best to visualize this data on a $\log_{10}$ Y-axis scale. You will then need to display a histogram using MatPlotLib showing the count of the words that are *Negative* [-1.0, -0.6), *Weakly Negative* [-0.6, -0.2), *Neutral* [-0.2, 0.2], *Weakly Positive* (0.2, 0.6], and *Positive* (0.6, 1.0].*

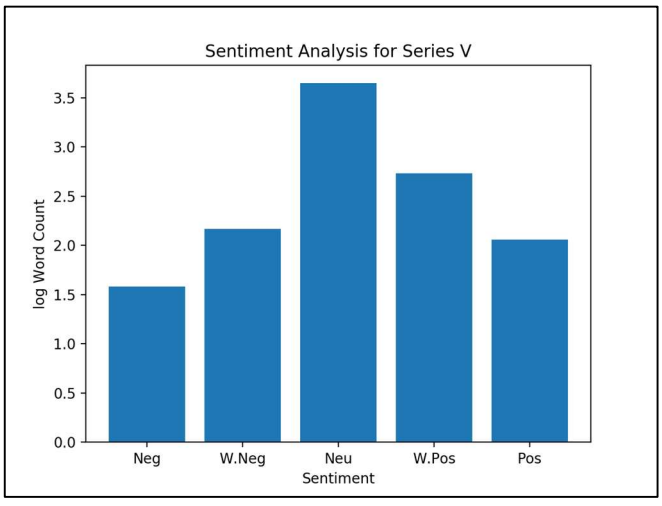
Make sure that the plot format matches the included sample (including axis labels and title with series name included). When you run your program, it should prompt you to enter the name of the series you want to plot (see example below).

Deliverables

1. Demo your working code to the class TA before uploading it to Blackboard. You cannot proceed to step 2 before doing this.
2. Once your code is working and you've demoed it to the TA, upload all your code to Blackboard as a single ZIP file. Name your ZIP file spiritAnimal.zip, where spiritAnimal is your class user ID (not your webID or ID number). Be sure to name your main source file "SpiritAnimal.py". In a comment at the top of the file, include the following information. Spirit Animal User ID, Date the file was last edited, Challenge Number, and cite any sources that you used as a reference for code, data, and content (including title and URL).

Sample Execution for Series V

```
[jones@Computer CSCI343]$ python main.py
Enter the series name: v
```



* These ranges are being presented using Interval Notion (https://en.wikipedia.org/wiki/Interval_(mathematics)#Notations_for_intervals)