**Project Overview**

I imported 5 text files authored by Charles Dickens from the Gutenberg Project. I analyzed them using DSU processes, dictionary histograms, and dictionary subtraction. Additionally, I tested using the NLP library package in order to test similarity of words. However, that library is not as well-developed as I had hoped so I ended up not being able to use it. I hoped to be able to create a program that would be able to find the main ideas behind the motifs of a text. The program works fairly well and is able to present the main ideas in a book. In the end, I plotted the words and their frequencies in a bar graph to better visualize them.

**Implementation**

The program analyzes the code initially to create a histogram for all the words in the text. However, some of those words can be easily disqualified based on their frequency. For example, the word ‘a’ should be disqualified as a motif and probably appears a large number of times. As a result, the program takes in lower and upper bound thresholds that remove completely implausible words from the list.

After this process occurs, the dictionary of words is more accurate to phrases related to the motifs. However, it is still not accurate enough. As a result, I need to compare the frequency of these words with the frequency of words in other books by the same author. This comparison will allow me to separate words used by the author commonly in writing from those words which are actually related to main ideas in the book. As a result, I created a list of dictionaries, with each dictionary containing words and their frequencies from each book. Afterwards, I simply iterate through each of the words in the dictionaries and compared their frequencies to the frequencies of those in the main book. If the fraction of occurrences of the words in their respective books is comparable, then it is not likely that the word belongs in the list of motifs.

Originally, I planned to pickle all of my text files. However, this proved to not be useful because I only wanted to compare the actual texts by Charles Dickens. In the text files downloaded from The Gutenberg Project, there were different headers and footers in each text file that would have interfered with my text analysis. As a result, I had to download the files individually and simply open them in python.

**Results**

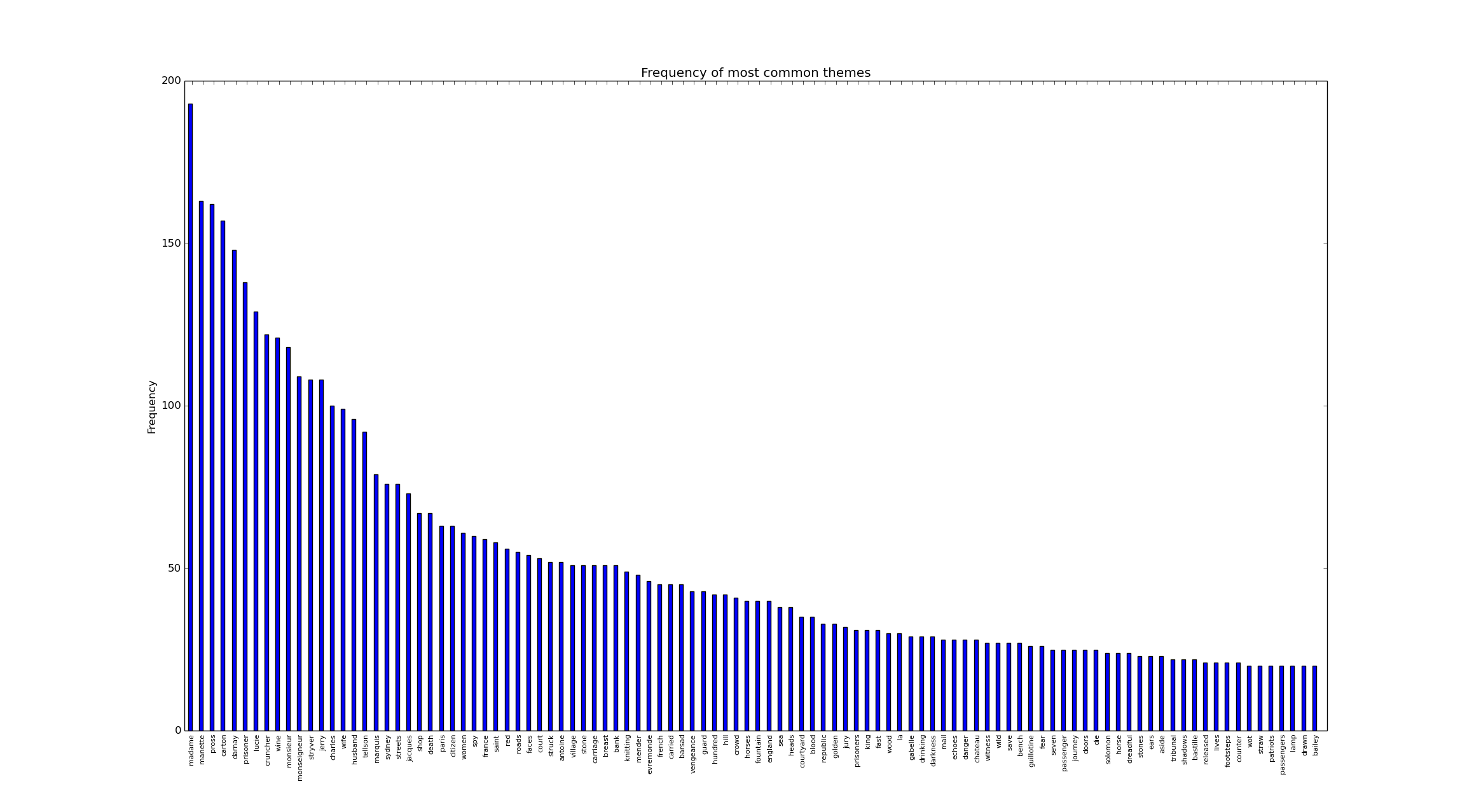
The following results are phrases associated with the motifs of *A Tale of Two Cities*:

['antoine', 'aside', 'bailey', 'bank', 'barsad', 'bastille', 'bench', 'blood', 'breast', 'carriage', 'carried', 'carton', 'charles', 'chateau', 'citizen', 'counter', 'court', 'courtyard', 'crowd', 'cruncher', 'danger', 'darkness', 'darnay', 'death', 'die', 'doors', 'drawn', 'dreadful', 'drinking', 'ears', 'echoes', 'england', 'evremonde', 'faces', 'fast', 'fear', 'footsteps', 'fountain', 'france', 'french', 'gabelle', 'golden', 'guard', 'guillotine', 'heads', 'hill', 'horse', 'horses', 'hundred', 'husband', 'jacques', 'jerry', 'journey', 'jury', 'king', 'knitting', 'la', 'lamp', 'lives', 'lucie', 'madame', 'mail', 'manette', 'marquis', 'mender', 'monseigneur', 'monsieur', 'paris', 'passenger', 'passengers', 'patriots', 'prisoner', 'prisoners', 'pross', 'red', 'released', 'republic', 'roads', 'saint', 'save', 'sea', 'seven', 'shadows', 'shop', 'solomon', 'spy', 'stone', 'stones', 'straw', 'streets', 'struck', 'stryver', 'sydney', 'tellson', 'tribunal', 'vengeance', 'village', 'wife', 'wild', 'wine', 'witness', 'women', 'wood', 'wot']

Their frequencies are shown in this output:

{'golden': 33, 'wine': 121, 'doors': 25, 'evremonde': 46, 'echoes': 28, 'crowd': 41, 'passenger': 25, 'republic': 33, 'manette': 163, 'jury': 32, 'passengers': 20, 'aside': 23, 'mender': 48, 'heads': 38, 'saint': 58, 'sea': 38, 'struck': 52, 'knitting': 49, 'sydney': 76, 'lucie': 129, 'counter': 21, 'witness': 27, 'courtyard': 35, 'roads': 55, 'marquis': 79, 'guillotine': 26, 'carton': 157, 'bank': 51, 'footsteps': 21, 'cruncher': 122, 'drawn': 20, 'jerry': 108, 'jacques': 73, 'prisoners': 31, 'darnay': 148, 'spy': 60, 'husband': 96, 'monsieur': 118, 'death': 67, 'dreadful': 24, 'streets': 76, 'darkness': 29, 'horse': 24, 'hundred': 42, 'vengeance': 43, 'wot': 20, 'fountain': 40, 'madame': 193, 'carriage': 51, 'lives': 21, 'gabelle': 29, 'solomon': 24, 'ears': 23, 'seven': 25, 'save': 27, 'journey': 25, 'shop': 67, 'die': 25, 'french': 45, 'patriots': 20, 'france': 59, 'mail': 28, 'fast': 31, 'bailey': 20, 'stryver': 108, 'bastille': 22, 'breast': 51, 'danger': 28, 'stone': 51, 'monseigneur': 109, 'lamp': 20, 'tellson': 92, 'la': 30, 'red': 56, 'prisoner': 138, 'charles': 100, 'pross': 162, 'bench': 27, 'citizen': 63, 'guard': 43, 'paris': 63, 'chateau': 28, 'straw': 20, 'shadows': 22, 'faces': 54, 'tribunal': 22, 'antoine': 52, 'blood': 35, 'released': 21, 'wood': 30, 'england': 40, 'wife': 99, 'horses': 40, 'court': 53, 'hill': 42, 'king': 31, 'stones': 23, 'village': 51, 'barsad': 45, 'drinking': 29, 'women': 61, 'fear': 26, 'wild': 27, 'carried': 45}

Below is a bar graph of the text I had generated. This bar graph is sorted by frequency in order to allow for a better understanding of which texts are most important. Because of the number of words, it is difficult to read the graph in detail unless it is full-screened.

Although it may appear that the texts results don’t give the results I want, there are actually many similarities in the results with the motif. For example, “blood,” “wine,” “red,” “drinking,” “crowd,” etc are all words which relate to the motif of drinking and being crazed by the mob effect to cause dangerous activities. In addition, it adds to other themes such as female and male relations, and the poor and the rich. 

It was difficult to remove some items such as proper names from the file since it isn’t as simple as removing all words with a capital letter. The best way I would implement this would be using a natural language processor or a dictionary of all words in the english dictionary. Some words such as “seven” have a signficance in the book but don’t necessarily relate to any motifs. However, overall, the program was very successful at deciphering those words which are significant to major themes in the book.

**Reflection**

This project went well in the sense that the program itself did what it was supposed to. However, I would like to implement more natural language processing algorithms in order to categorize the results in a more succinct way that will allow users to be able to form a thesis about motifs in the book based on the groups. After grouping the words, I would plan on using association in the text to detect areas where words in the group are in close proximity. This will allow me to highlight main areas where the motif is elaborated upon in more detail. I believe my project would have been able to go further if the natural language processing library had a wider variety of associations than it did.