1 Assignment 2

1) Create a variable phrase containing a list of words. Review the operations described in the previous chapter, including addition, multiplication, indexing, slicing, and sorting.

Addition:

```
phrase = ['sandwiches', 'are', 'very', 'tasty']
>>> phrase = phrase + ['!']
>>> phrase
['sandwiches', 'are', 'very', 'tasty', '!']
```

Indexing

```
>>> phrase[0]
'sandwiches'
```

Slicing

```
>>> phrase[2:4]
['very', 'tasty']
```

Sorting:

```
>>> words = sorted(phrase)
>>> words
['!', 'are', 'sandwiches', 'tasty', 'veryvery']
```

Multiplication:

```
>>> phrase[2] = phrase[2] * 2
>>> phrase
['sandwiches', 'are', 'veryvery', 'tasty', '!']
```

2) Use the corpus module to explore austen-persuasion.txt. How many word tokens does this book have? How many word types?

```
>>> persuasion =
    nltk.corpus.gutenberg.words('austen-persuasion.txt')
>>> persuasion
['[', 'Persuasion', 'by', 'Jane', 'Austen', '1818', ...]
>>> len(persuasion)
```

```
98171
>>> len(set(persuasion))
6132
```

3)Use the Brown corpus reader nltk.corpus.brown.words() or the Web text corpus reader nltk.corpus.webtext.words() to access some sample text in two different genres.

```
>>> nltk.corpus.brown.words()
['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...]
>>> nltk.corpus.webtext.words()
['Cookie', 'Manager', ':', '"', 'Don', "'", 't', ...]
```

4)Read in the texts of the State of the Union addresses, using the `state_union` corpus reader. Count occurrences of men, women, and people in each document. What has happened to the usage of these words over time?

```
>>> nltk.corpus.state_union.words()
['PRESIDENT', 'HARRY', 'S', '.', 'TRUMAN', "'", 'S', ...]
>>> sotu = nltk.corpus.state_union.words()
>>> sotu.count("men")
228
>>> sotu.count("women")
141
>>> sotu.count("people")
1291
```

5)Investigate the `holonym-meronym` relations for some nouns. Remember that there are three kinds of `holonym-meronym` relation, so you need to use: `member_meronyms()`, `part_meronyms()`, `substance_meronyms()`, `member_holonyms()`, `part_holonyms()`, and `substance_holonyms()`.

19)Pick a pair of texts and study the differences between them, in terms of vocabulary, vocabulary richness, genre, etc. Can you find pairs of words which have quite different meanings across the two texts, such as monstrous in Moby Dick and in Sense and Sensibility?

23)Zipf's Law: Let f(w) be the frequency of a word w in free text. Suppose that all the words of a text are ranked according to their frequency, with the most frequent word first. Zipf's law states that the frequency of a word type is inversely proportional to its rank (i.e. f r = k, for some constant k). For example, the 50th most common word type should occur three times as frequently as the 150th most common word type. Write a function to process a large text and plot word frequency against word rank using pylab.plot. Do you confirm Zipf's law? (Hint: it helps to use a logarithmic scale). What is going on at the extreme ends of the plotted line? Generate random text, e.g., using random.choice("abcdefg "), taking care to include the space character. You will need to import random first. Use the string concatenation operator to accumulate characters into a (very) long string. Then tokenize this string, and generate the Zipf plot as before, and compare the two plots. What do you make of Zipf's Law in the light of this?