

Lab 11: Word Counts

due: Dec 7 or 8, 2017

The purpose of this lab is to count the frequency of unique words in a file. You are given a program that goes through a text file and finds unique words, storing each of these unique words in an array of strings. Your task is to count how many times each word occurs and visualize that frequency in some manner.

1. Read in the text file.
2. Use the provided code to loop through all the text and store each different word in the array.
3. Modify the code to count the frequency of each word.
4. Draw a boxplot with the words and their frequency.
5. **Challenge:** Use a different text file, consider using project gutenber¹.
6. **Challenge:** Create a visualization where the size of each word corresponds to its frequency.

Basic Boxplot Code:

```
int[] freq = { 40, 80, 30, 34, 78, 22, 30, 9, 10 };
for (int i = 0; i < freq.length; i++) {
    rect(0, i*10, freq[i], 9);
}
```

Learning Objectives

- Read text files.
- Practice using arrays.
- Create boxplots.

Deliverables

- Your program should start with a comment that includes your name, email, date, assignment description, collaboration statement, and reflection.
- Bring a hardcopy of your program (the source code, not the graphics) to your next lab period.
- Be prepared to run the Processing sketch and demonstrate your “[Theory of the Program](#)”.

```
void setup() {
    size(500, 500);
    WordCount wc = new WordCount();
    String[] data = loadStrings("gettysburg-address.txt");
    for (int i = 0; i < data.length; i++) {
        String[] row = splitTokens(data[i]);
        for (int j = 0; j < row.length; j++) {
            String s = row[j];
            wc.add(new Word(s));
        }
    }
    wc.display();
}
```

¹ <https://www.gutenberg.org>

```

class Word {
    String s;
    int freq;

    Word(String s) {
        freq = 1;
        this.s = s.toLowerCase().replaceAll("\\W", "");
    }

    String toString() {
        return s + ":" + freq;
    }
}

```

```

class WordCount {
    Word[] words = new Word[0];

    // Return the index of word in the array,
    // return -1 if it does not exist
    int find(String word) {
        for (int i = 0; i < words.length; i++) {
            if (words[i].s.equals(word)) {
                return i;
            }
        }
        return -1;
    }

    //Add s to the word count
    void add(Word w) {
        if (find(w.s) == -1) {
            words = (Word[])append(words, w);
        }
    }

    void display() {
        println(words);
    }
}

```