L&S 39F: Data science and the mind (Fall 2005)

Homework 1

September 22 (Due *beginning* of session on Tue, Oct 6)

*This instruction sheet is best accessed electronically - the cyan boxes below provide clickable links. All coding should be completed in a Python Jupyter Notebook (or Python). Please type your solutions into a *single* PDF that includes both figures and written explanations, with your own Python code enclosed in Appendix. Email this PDF to the course instructor with the title "L&S39F HW1 (YOUR NAME)". You should turn in your individual solutions, although collaborative discussions are encouraged (acknowledge your collaborator(s) in the write-up). Late work receives a deduction of 1 point per delayed hour.*

In this assignment, you will explore the principle of word frequencies. The purpose is to illustrate how appropriate data manipulation can help visualize principles such as Zipf's law. Specifically, you will be able to reconstruct Fig. 2-1 (A) on p25 of Zipf's *Human behavior and the principle of least effort* (i.e. session reading material) with modern computing tools - the computer. To do so, you will first need to take the following steps:

1) Download *Data* and *Code* under *Hw1* in the NB column from the bcourses syllabus.
2) Upload "ulysses.txt" to your Python Jupyter folder.
3) Copy and paste the content of "hw1_starter_code.txt" into a new Python notebook. You will now be in a position to explore the following questions.

Q1. Running the starter code, you should obtain "word_dict" which is a dictionary [practice] containing (word,frequency) pairs for all unique words in the novel Ulysses (Joyce, 1922). Your job is to 1) rank these words by their frequencies in a descending order, such that the word with the highest frequency has a rank of 1, 2) generate Figure 1 that plots frequency (y-axis) against rank (x-axis) for the 100 most frequent words. *Note: Use dots to represent data points and label both axes for this and all following figures. Points will be deducted for figures that contain unannotated axes.* [2 points]

Q2. Generate Figure 2 that plots frequency against rank for the 10,000 most frequent words instead. [1 point] Why do you think plotting the raw frequencies and rank values would be unideal in this case? Explain briefly. [1 point]

Q3. Generate Figure 3 that plots log-transformed frequency against log-transformed rank for the 10,000 most frequent words from Q2 - did your figure replicate that in Zipf's book? [1 point]

Q4. For each of the 10,000 words you have examined, calculate the their word lengths (e.g. "of" has length 2, and "the" has length 3). Generate Figure 4 that plots word length against log-transformed frequency for these words. [2 points] What trend do you observe from these data? State briefly. [1 point] How would you explain this observation? [1 point]

Q5. Tabulate the most frequent 20 words and their frequency counts obtained from your Python program. [1 point]

Q-fun. Try your code on different novels from Gutenberg (for fun)! You can do so by downloading the plain text file for a given novel, uploading it onto your Jupyter folder, and assigning variable "novelfile" to whichever name you have assigned to the uploaded .txt file. Does Zipf's law apply equally well to novels other than Ulysses?