Lab – Zipf's Law

This lab concerns **Zipf's law**: the empirical statistical observation that word frequency is a power law of word rank (straight line in a log-log plot) with exponent -1. Much is known about Zipf's law but many questions remain open, including:

- What explains which documents fit Zipf's law and which don't?
- Why is the exponent -1? What deviations from this exponent are there, and what explains them?
- What other documents (in addition to Wikipedia and the patent record) empirically show a bend in slope from -1 to -2? What explains where it bends?

In this lab you try to answer these questions.

This lab may be done individually or in a small group (up to 4 people). Individuals and groups will briefly present their results, and the class will discuss whether they suggest answers to the questions above. Presentations typically consist of slides (Keynote, Powerpoint, etc.)

The **lab report** from each individual and group consists of the following slides:

1. title slide with names of students, description of the question investigated, documents studied, and counts of word types and tokens in all documents
2. slides (one or more) with Zipf law plots for all documents studied
3. a slide summarizing the upshot of your results

This lab analyses individual documents for evidence of Zipf's law using a few simple **Unix commands** in the **terminal window**. (UNIX is a very flexible and powerful programming environment.) This lab also uses two little programs or scripts (in R and Python), but the scripts are treated as black boxes and not modified in this lab.
The scripts can be run on any standard digital text file (best is a *.tex document devoid of special formatting characters). The lab illustrates these scripts by applying them to a data file consisting of about 1000 abstracts from patents published on the USPTO web site in January 2005.

To start this lab, create a working directory with the following 3 files:

- patentsample.txt – English text file with 1000 patent abstracts (Jan 2005)
- doc2freq.py – Python script, makes frequency table of words from text document
- zipf.R – R script, makes Zipf plot from a frequency table

Get all the files from the Phil 412 moodle. In a terminal window, change your working directory to your working directory for this lab:

```
cd [drag directory folder from Finder into terminal window]
```

Preliminary tasks - Examine the electronic document patentsample.txt, using TextEdit.

- Record the size of the document
- Record the size of the dictionary for the document
- Briefly describe the contents of the document
- Try to predict word frequencies in this document

In your lab report record the document size (number of word tokens) and dictionary size (number of word types) of each of your documents, so that the class can aggregate those results into a type vs. token scatter plot.

To calculate the document size, enter the following command in a terminal window:

```
wc   patentsample.txt
```

This command produces three numbers: 993 107088 729088 patentsample.txt
The number of words (tokens) in the file is the middle number. (The first number is the number of lines in the file, and the third is the number of characters.)

To make the Python script into an executable command, enter the following command in the working directory:

```
chmod 755 doc2freq.py
```

The Python script doc2freq.py creates a word frequency table (in a *.tsv file) for any text document you supply.

- Examine this file in TextEdit, and briefly describe it.
- What in this file do you understand and what is a total mystery?

To run this Python script on the text document containing the patent abstracts, enter this command in your working directory:

```
python    doc2freq.py    patentsample.txt
```

This creates a new file in your working directory (named *patentsample.tsv*) with the word frequency table for the document.

- How many words are in the document's dictionary?
- Describe any "dirt" you see in the data and describe how you could clean it.

Use the zipf.r to plot the resulting frequncy data, in the terminal window:

```
Rscript     zipf.r     patentsample.tsv
```

This script creates a pdf file in your working directory, called *patentsample.tsv.pdf*, containing a Zipf plot of your original document, a log-log plot of the frequency of each word in the document as a function of its rank among all words in the document's dictionary.

- In a Finder window, open the Zipf plot and describe its notable features.

The main task: Devise and complete a project investigating some aspect of Zipf's law in various kinds of documents. Describe what you see in your project results, and explain what you can. Can you make any interesting testable predictions? Here are examples of possible project subjects:

1. Make a Zipf plot of some **real world text corpus**, using (and perhaps modifying) the code you were given.
2. Make Zipf law plots for **random languages** of different sizes. Put all the Zipf plots on the same graph, with the same axes.
3. Make a log-log scatter plot of **dictionary size vs. corpus size** (# word types vs. # word tokens) for the data sets analyzed by other students.
4. Add useful features to **Zipf plotting software**
    a. display the dictionary size and corpus size
    b. fit a line to the data (or a subset), plot the line over a scatter plot
    c. display the line's slope (exponent)
    d. display error bars for the slope (exponent)