CSCI 657 Spring 2023

HOMEWORK #1

Word frequencies, word length frequencies and Zipf’s Law  
(60 points)MERGEFORMATMERGEFORMAT

Purpose

1. Ensure that everyone can use basic Python constructs including dictionaries, list comprehensions, command line arguments, sorting, formatting, basic I/O and file encodings.

2. Ensure that everyone has a turing password and basic Linux competence. (There will only be 1‑2 assignments on turing, but I will also use to post data and sample code.)

3. Obtain some data on letter frequencies in English. Letter frequency is useful for many applications, including cryptography, data compression, and language identification. Language identification is useful in machine translation, speech recognition and other applications.

4. Obtain some empirical evidence for (or against) Zipf’s Law. Zipf’s Law says that frequency is inversely proportional to rank. We will look at frequency of word lengths.

5. Obtain some basic practice in data cleaning.

Specs

1. Use the UTF‑8 version of The Adventures of Tom Sawyer provided on turing at ~t90rkf1/d657/dhw/hw1-zipf/74-2021-0329.txt. You can tell that this file is UTF‑8 by reading the first few lines of the file. You could also make a guess (many programs do) by looking at the frequencies of bit patterns. However, character files on Unix do not contain metadata to provide an unambiguous identification of encodings.

This text comes from Project Gutenberg at https://www.gutenberg.org/ebooks/74, but please use the version provided on turing so that you can get the specific version and encoding desired.

Write a program to do the following.

2. Open the file. If a command line argument is provided, use it as the file name. Otherwise use the default file name  
/home/turing/t90rkf1/d657/dhw/hw1-zipf/74-2021-0329.txt.  
The default encoding is UTF‑8 on turing, but you may need to specify it on other platforms.

I will run your program as follows: ./hw1-xxxx.py > hw1out.txt, i.e. using the default filename and your FLID (not the constant xxxx!). That also means you need the #! line at the top of your program.

3. In Project Gutenberg files, you don’t have to worry about words being continued from one line to the next. Replace all characters other than letters by a space. Use the replace function of string, not a loop. (We will separately study the effect of dropping the hyphen and the apostrophe.

4. Convert to lower case.

5. Print a table showing the frequency of each word length, i.e., how many 1-letter, 2-letter, etc. words are found in the input, as shown in the sample output. To avoid division by 0, leave the last column of the first row blank. Print a total line showing the number of words represented in the table.

6. Print the summary statistics shown in the sample output. The purpose of the summary statistics is to verify that your program is working correctly. You may need to use another source, e.g., less, grep or wc, to verify that your output is correct.

7. Print a table showing the frequency of each word frequency as shown in the sample output.

8. Print a table showing the frequency of each word as shown in the sample output.

Test data:

You will find Unix utilities like less, grep and wc useful for making sure your counts are correct.

Your output should match mine in the homework directory. You can test that as follows:

./hw1-xxxx.py > myout.txt

diff -Bw myout.txt ~/d657/dhw/hw1-zipf/hw1out.txt

That directory also includes a bash script ./hw1ck.bash that you can use if you prefer to execute the diff command above.

Submission:

Follow one of the style conventions for this class.

Name your program hw1-xxxx.py, where xxxx is your FLID.

All tables must have aligned columns. Integers must be right-aligned. Real numbers must be decimal-aligned.

Every number you print must have a rubric, e.g., as a row or column heading.

Use list comprehensions or other high-level constructs wherever possible. You can use a loop for the main I/O, but otherwise only use loops if there is no other appropriate way to solve a problem.

Grading:

Three tables @ 15 points each, plus 15 points for the other miscellaneous outputs. There will be a penalty for violating any of the rules above. Also, there are penalties for late homework and