2019-2-4

Lab 04 - Due at end of class

Make a new branch in your local assignments repo called lab04, and be sure your local repo has that branch checked out. (do a git status in the command line and it will tell you the branch you are on) Do all work on that branch. When your changes done and pushed to github, issue a pull request and add me as a reviewer.

1. 30 pts. Word histograms part 2. Do the word histogram of emma.txt as you did last lab. Also read in words.txt as a dictionary of valid words. Have the script only print out words in the histogram that are also in words.txt.
2. Print a final total # of words that are in the words.txt “dictionary” and using matplotlib.pyplot, plot the 10 words that occur the most.
3. How many words are in your found words dictionary that are NOT in the reference dictionary?
4. Examine the words that are not in the reference dictionary, write a few suggestions on what you could do to improve the accuracy of your program (ie. How to change/filter the incoming data to produce more matches).
5. Birthdays

This exercise pertains to the so-called Birthday Paradox, which you can read about

at http://en.wikipedia.org/wiki/Birthday\_ paradox

A ) 35 pts. The theory is that with 23 random people in a room, there is a 50% chance that two people will have the same birthday. Write a function that generates N random birthdates, then returns True if any of them match, and False otherwise. It should accept integers from 1-366 and return a boolean value.

I recommend using numpy arrays rather than python lists. They have more useful functions like mean builtin.

The analogy to creating an empty list and appending a result in a loop is as follows:

Python list

X = []

For i in range(50):

X.append(some\_result())

Numpy array

X = np.empty(0)

For i in range(50):

X = np.append(X,some\_result())

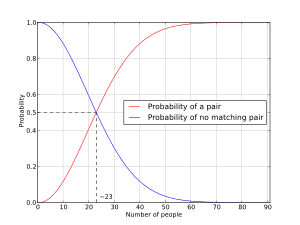
Then it’s useful to use X.mean(), max(), min(), sort() etc to see meaningful results.

1. 35 pts. Write a function that uses randint to generate random birthdates, returning the # of birthdates that were generated when the first duplicate occurs.

Call this function many times and compute the average # of people/birthdates it took to find a pair that matched.

Play with the # of trials run. How many trials are required before your simulated results are within 1% of the mathmatical projections?

1. OPTIONAL/BONUS - 10 pts. Generate your own plot of the probability of a pair (matching birthdates) for 1-80 people as on the wikipedia page. How many trials must you run to get plot that is within 1% of the data points in the probability table on the wikipedia page?



1. I like jokes. Tell me a joke if you know a good one. Here is a recent one I liked

------------------------[ Original Message ]--------------------

Letter of Recommendation -

While working with Mr. Xxxxxx, I have always found him

working studiously and sincerely at his table without idling or

gossiping with colleagues in the office. He seldom

wastes his time on useless things. Given a job, he always

finishes the given assignment in time. He is always

deeply engrossed in his official work, and can never be

found chitchatting in the canteen. He has absolutely no

vanity in spite of his high accomplishment and profound

knowledge of his field. I think he can easily be

classed as outstanding, and should on no account be

dispensed with. I strongly feel that Mr. Xxxxxx should be

pushed to accept promotion, and a proposal to administration be

sent away as soon as possible.

Jim YYYYY

Branch Manager

-----------------------------------------------------------------

A second note following the report:

XXXXXX WAS PRESENT WHEN I WAS WRITING THE REPORT MAILED TO YOU TODAY. KINDLY

READ ONLY THE ALTERNATIVE LINES 1,3,5,7,... FOR MY TRUE ASSESSMENT OF HIM.

REGARDS,

Jim YYYYY

Branch Manager

For a bonus 5pts, can you write a script that will print out the intended message in the joke?