CSCI 1052 Problem Set 1

January 3, 2024

Submission Instructions

Please upload your solutions by **5pm Friday January 12, 2023.**

- You are encouraged to discuss ideas and work with your classmates. However, you **must acknowledge** your collaborators at the top of each solution on which you collaborated with others and you **must write** your solutions independently.
- Your solutions to theory questions must be typeset in LaTeX or markdown. I strongly recommend uploading the source LaTeX (found here) to Overleaf for editing.
- I recommend that you write your solutions to coding question in a Jupyter notebook using Google Colab.
- You should submit your solutions as a **single PDF** via the assignment on Canvas.

Problem 1 (from January 4)

In class, we calculated the number of duplicates in a sample of size m as

$$D = \sum_{i=1}^{m} \sum_{j=i+1}^{m} D_{i,j}$$

where $D_{i,j}$ is an indicator random variable that the ith and jth sampled items are the same. In expectation when the samples are drawn uniformly at random, we saw that

$$\mathbb{E}[D] = \frac{m(m-1)}{2n}.$$

Part 1

In practice, we know m the number of samples we've taken and D the number of duplicates in the sample. Using these quantities, suggest a method for estimating n the set size inspired by our expression for $\mathbb{E}[D]$.

Part 2

Implement your method from Part 1 to estimate the number of unique articles in the English Wikipedia. You can access "random" articles (see the discussion here) by visiting the link: `https://en.wikipedia.org/wiki/Special:Random`.

According to the article here, English Wikipedia has 6.7 million articles. How does that compare to your estimate? How does the fact that the random article feature doesn't perfectly return random articles bias your estimate?

In Python, you can get a random URL by running the following code:

```
import requests
response = requests.get("https://en.wikipedia.org/wiki/Special:Random")
random_url = response.url
```

In my experiments, it took 30 minutes to get 5000 random articles. At this rate, it would take 28 days to get 6.7 million articles. How long did your code take to run?

In your solution, include all relevant results and calculations in addition to a discussion of how accurate you think your estimate is.