Math 126 Problem 2: Wikipedia Articles

Spring 2027

Submission Instructions

Please upload your work to Gradescope by **11:59pm on Monday, February 1, 2027 (Pacific time).**

- 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 written legibly, or typeset in LaTeX or markdown.
- I recommend that you write your solutions in a Python notebook using Google Colab.
- You should submit your solutions as a *single PDF* via the assignment on Gradescope.

Grading: The point of the problem set is for *you* to learn. To this end, I hope to disincentivize the use of LLMs by *not* grading your work for correctness. Instead, you will grade your own work by comparing it to my solutions. This self-grade is due the Friday *after* the problem set is due, also on Gradescope.
Original notebook: Google Colab.

Wikipedia Articles

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 A

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 B

Implement your method from Part A 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?

Here's some code you can use:

Starter Code and Data

```
import requests
import math

# Increase samples to get enough collisions
m = 100000
batch_size = 500
titles = []

URL = "https://en.wikipedia.org/w/api.php"
headers = {"User-Agent": "WikiSizeEstimator/1.0"}

print(f"Fetching {m} random URLs (this will take a few minutes)...")

while len(titles) < m:
    params = {
        "action": "query",
        "format": "json",
        "list": "random",
        "rnlimit": batch_size,
        "rnnamespace": 0
    }

    try:
        response = requests.get(url=URL, params=params, headers=headers)
        data = response.json()

        if "query" in data:
            for page in data["query"]["random"]:
                titles.append(page['id']) # Use ID, it's safer/faster than title strings
        else:
            print("Retrying...")

    except Exception:
        pass
```