Problem Set 5 (due Wednesday, November 3, 11:59pm)

Directions:

Answers: For each problem, remember you must briefly explain/justify how you obtained your answer, as correct answers without an explanation will receive no credit. Moreover, in the event of an incorrect answer, we can still try to give you partial credit based on the explanation you provide. It is fine for your answers to include summations, products, factorials, exponentials, or combinations; you don’t need to calculate those all out to get a single numeric answer, for instance $26^7$ or $26!/7!$ or $26 \cdot \binom{26}{7}$.

Your solutions need to be concise, clear and mathematically rigorous. We will take off points for lack of clarity or for excess verbosity. Please see section worksheet solutions (posted on the course website) to gauge the level of detail we are expecting.

Please clearly indicate your final answer, in such a way as to distinguish it from the rest of your explanation.

This pset may be done with a single partner. In this case, only one person will submit on Gradescope and add their partner as a collaborator. This time this DOES NOT apply to the coding. The coding in this assignment will be done solo, and uploaded as such. Individuals and pairs are still encouraged to discuss problem-solving strategies with other classmates as well as the course staff, but each pair must write up their own solutions independently and without looking at anything anybody else wrote up.

Submission: You must upload a pdf of your written solutions to Gradescope under “PSet 5 [Written]”. Problem 6 is a coding problem, so under “Pset 5 [Coding]” you will be individually uploading a .py file called cse312_pset5_gen_rvs.py. If you do the extra credit, that will be submitted separately under “Pset5 [Extra]”. (Instructions as to how to upload your solutions to Gradescope are on the course web page.) The use of LaTeX is highly recommended.

Note that if you want to hand-write your solutions, you’ll need to scan them. We will take off points for hand-written solutions that are difficult to read due to poor handwriting and neatness.

Please cite any collaboration at the top of your submission (beyond your partner, if you have one, who should already be listed).

1. Bloom Filter Analysis (12 points)

Suppose you are using a Bloom filter with $k$ hash functions, where each bit array has length $m$ and suppose that you’ve added a set $S$ of $n$ items from universe $U$ into the Bloom filter. For concreteness suppose that $S = \{x_1, \ldots, x_n\}$. Suppose that subsequently there is a lookup (“contains”) of item $y \not\in S$. We want to compute the probability of a false positive on $y$.

We make the following heuristic assumptions for the purposes of this analysis:

- For every element $u \in U$, and for every hash function $h_i$, $h_i(u)$ is uniformly distributed. (i.e., each $h_i(u)$ is equally likely to be any index in $\{0, 1, \ldots, m - 1\}$).

- All of the $h_i(u)$ values, as we vary both $i$ (between 1 and $k$) and $u \in U$ are mutually independent.

For concreteness let $X_{ij} := h_i(x_j)$ and let $Y_i := h_i(y)$. The assumptions imply that all of the $X_{ij}$’s and $Y_i$’s are mutually independent (for all $i$ and $j$).
(a) [2 Points] What is 
\[ \Pr(X_{ij} = Y_i) \]?

(b) [5 Points] What is 
\[ \Pr(\text{there is some } j \in \{1, \ldots, n\} \text{ such that } X_{ij} = Y_i) \]?

(c) [5 Points] What is the probability of a false positive on item \( y \)?

2. Random Grades (15 points)
Every week, 20,000 students flip a 10,000-sided fair dice, numbered 1 to 10,000, to see if they can get their GPA changed to a 4.0. If they roll a 1, they win (they get their GPA changed). You may assume each student’s roll is independent. Let \( X \) be the number of students who win.

(a) [5 Points] For any given week, give the appropriate probability distribution (including parameter(s)), and find the expected number of students who win.

(b) [5 Points] For any given week, find the exact probability that at least 2 students win. Give your answer to 5 decimal places.

(c) [5 Points] For any given week, estimate the probability that at least 2 students win, using the Poisson approximation. Give your answer to 5 decimal places.

3. Instagram (15 points)
A photo-sharing startup offers the following service. A client may upload any number \( N \) of photos and the server will compare each of the \( \binom{N}{2} \) pairs of photos with their proprietary image matching algorithms to see if there is any person that is in both pictures. Testing shows that the matching algorithm is the slowest part of the service, taking about 100 milliseconds of CPU time per photo pair. Hence, estimating the number of photos uploaded by each client is a key part of sizing their data center. The people in charge say that their gut feeling is that \( N = 10 \). You (the chief technical officer) say, “but \( N \) is a random variable”. What will the expected time (in milliseconds) for CPU demand per client be (as a function of \( N \), \( p \) or \( \lambda \)) if \( N \) follows

(a) [3 Points] the “distribution” where \( N \) is the same fixed number with probability 1?

(b) [4 Points] the Poisson distribution with parameter \( \lambda \)?

(c) [4 Points] the geometric distribution with parameter \( p \)?

(d) [4 Points] \( N = 80X + 5 \), where \( X \) is a Bernoulli random variable with parameter \( p \)?

In each case, include as part of your answer the expected value of \( N \) and the variance of \( N \). Make sure your answer is not in the form of a summation for this problem.

4. Binomial From Nowhere (13 points)
Consider repeatedly rolling a fair 6-sided die, each roll being independent of the others. Define the random variable \( Y \) to be the number of rolls until (and including) the first roll of a 6, and define the random variable \( X \) to be the number of 1’s rolled before the first 6 is rolled. Show that \( \Pr(X = j \mid Y = i) \), as \( j \) ranges over its possible values, is the probability mass function of a binomially distributed random variable and determine its parameters \( n \) and \( p \).

5. Sample Sampling Algorithm (15 points)
Consider the following algorithm for generating a random sample of size \( n \) from the set of integers \( \{1, 2, \ldots, N\} \), where \( 0 < n < N \).
Sample($N, n$):

1. $I = 0$
2. chosen = {} // chosen is a set of distinct integers, initially an empty set
3. while $|\text{chosen}| < n$:
   1. $I += 1$ // I is counting the total number of rolls of the die
   2. chosen.add(RollDie($N$)) // if the roll of the die (which is random in 1 ... $N$)
   3. // is not in chosen, then add it to chosen.
4. return chosen

(a) [9 Points] Let the random variable $I_i$ be the number of rolls it takes from the time the chosen set has $i-1$ values to the first time a new value is added (i.e., the chosen set has $i$ values). What type of random variable from our zoo is $I_i$ and what is/are the relevant parameter(s)? What is $I$ in terms of the random variables $I_i$? Calculate $E[I]$ in terms of $N$ and the harmonic numbers $H_m$ ($m \geq 1$). Recall that

$$H_m = \sum_{i=1}^{m} \frac{1}{i}$$

and for $k > r \geq 1$:

$$H_k - H_r = \frac{1}{r+1} + \frac{1}{r+2} + ... + \frac{1}{k} = \sum_{i=r+1}^{k} \frac{1}{i}$$

(b) [6 Points] What is $\text{Var}(I)$? You can leave your answer in summation form.

6. Explore the zoo! (10 points)

[Coding] Understanding the process that leads to different random variables is a great way to gain familiarity for what they mean. For each random variable, write a function that simulates its generation process. Your function should return a random sample of that rv, with the appropriate probability. The only function you can and should use to generate randomness is np.random.rand(): a function that returns a uniform random float in the range $[0, 1]$. Note that a function from one part may call a function from a previous part if you wish. For more clarity, we are asking you to generate a random sample from a particular distribution; multiple calls to your function can and should return different values in its range, approximately matching that variable's probability mass function.

Write your code for the following parts in the provided file: cse312_pset5_gen_rvs.py

(a) $X \sim \text{Ber}(p)$: 1 with probability $p$ and 0 with probability $1 - p$.

(b) $X \sim \text{Bin}(n, p)$: the number of heads in $n$ independent flips of a coin with probability of heads $p$.
   Implement the function gen_bin.

(c) $X \sim \text{Geo}(p)$: the number of flips up to and including the first head, when the probability of heads is $p$.
   Implement the function gen_geo.

(d) $X \sim \text{NegBin}(r, p)$: the number of flips up to and including the $r$-th head in independent coin tosses, when the probability of heads is $p$. Implement the function gen_negbin.

(e) $X \sim \text{HypGeo}(N, K, n)$: the number of kit kats you get when you grab $n$ random candies from a bag consisting of $N$ total candies, only $K$ of which are kit kats. Implement the function gen_hypgeo.

(f) $X \sim \text{Poi}(\lambda)$: the number of events in a minute, where the historical rate is $\lambda$ events per minute. Implement the function gen_poi.

(g) Given an arbitrary list (or numpy array) of probabilities, like $\vec{p} = [0.1, 0.3, 0.4, 0.2]$, sample an index with the appropriate probability. That is, return 0 with probability 0.1, 1 with probability 0.3, 2 with probability 0.4, and 3 with probability 0.2. Implement the function gen_arb.
7. Extra credit: Which one is real? (10 points)

Below are two sequences of Heads and Tails, each (supposedly) representing 300 independent flips of a fair coin. One of these sequences was truly randomly created, and one was typed by a human. Both sequences have exactly 149 heads. Which one is more likely to be the “real” random sequence? In your write-up, you should justify your reasoning with evidence and valid results, e.g. from running your code on the two sequences. There are multiple valid and correct approaches. To be eligible for full credit, you are required to turn in your detailed analysis along with your Python code.

(a) (Sequence 1)

```
TTHHTHTHTTHTHTTTTTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHT
THHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHT
THHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHT
HTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHT
```

(b) (Sequence 2)

```
HHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHT
HTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHT
HTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHTHT
```
