Lisa Yan
CS 109

Problem Set #3
Updated October 17, 2019

# Problem Set #3
# Due: 1:00pm on Wednesday, October 23rd

With problems by Mehran Sahami, Chris Piech, and David Varodayan

**Errata (updated October 17)**: Problem 10 has been updated to fix question numbering. The text of the problem has not changed.

**For each problem, briefly explain/justify how you obtained your answer.** In fact, most of the credit for each problem will be given for the derivation/model used as opposed to the final answer. Make sure to describe the distribution and parameter values you used (e.g., Bin(10, 0.3)) where appropriate. It is fine for your answers to include summations, products, factorials, exponentials, or combinations, unless you are specifically asked for a computed numerical answer.

## Warm-up

1. 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 number. The only probability function that you may use when coding your solution is `random()`: a function that returns a uniform random in the range [0, 1]. Submit your code (either Python or psuedocode). We include a solution to (a):

   a. $X \sim \text{Ber}(p = 0.4)$
   1 or 0 to indicate whether or not an underlying event was "successful."

```
def simulateBernoulli(p = 0.4):
    if random() < p:
        return 1
    return 0
```

   b. $X \sim \text{Bin}(n = 20, p = 0.4)$
   The number of successes after 20 independent experiments.

   c. $X \sim \text{Geo}(p = 0.03)$
   The number of trials until the first success.

   d. $X \sim \text{NegBin}(r = 5, p = 0.03)$
   The number of trials until 5 successes.

   e. $X \sim \text{Poi}(\lambda = 3.1)$ *approximate*
   The number of events in a minute, where the historical rate is 3.1 events per min.
   *Hint:* Break the minute down into 60,000 ms events like we did in lecture.

   f. $X \sim \text{Exp}(\lambda = 3.1)$ *approximate*
   The amount of time until the next event, where the historical rate is 3.1 events per min.
   *Hint:* Like part (e), think of an event for each millisecond.

If you are trying to understand probability mass functions, you may optionally try to visualize one via your simulations. For extra credit, run one of your simulations 100,000 times and plot a histogram of return values.

2. Lyft line gets 2 requests every 5 minutes, on average, for a particular route. A user requests the route and Lyft commits a car to take her. All users who request the route in the next five minutes will be added to the car as long as the car has space. The car can fit up to three users. Lyft will make $6 for each user in the car (the revenue) minus $7 (the operating cost).
    a. How much does Lyft expect to make from this trip?
    b. Lyft has one space left in the car and wants to wait to get another user. What is the probability that another user will make a request in the next 30 seconds?

3. Suppose it takes at least 9 votes from a 12-member jury to convict a defendant. Suppose also that the probability that a juror votes that an actually guilty person is innocent is 0.25, whereas the probability that the juror votes that an actually innocent person is guilty is 0.15. If each juror acts independently and if 70% of defendants are actually guilty, find the probability that the jury renders a correct decision. Also determine the percentage of defendants found guilty by the jury.

4. Let $X$ be a continuous random variable with probability density function:

$$f(x) = \begin{cases} c(2 - 2x^2) & -1 < x < 1 \\ 0 & \text{otherwise} \end{cases}$$

    a. What is the value of $c$?
    b. What is the cumulative distribution function (CDF) of $X$?
    c. What is $E[X]$?

5. Scores on the SAT maths (out of 800) are normally distributed with a mean of 500 and a standard deviation of 100.
    a. What fraction of students receive a score within 1.5 standard deviations of the mean?
    b. Irina scores 750. What percent of students scored lower than 750? (Irina's percentile)

6. You are testing software and discover that your program has a non-deterministic bug that causes catastrophic failure (aka a "hindenbug"). Your program was tested for 400 hours and the bug occurred **twice**.
    a. Each user uses your program to complete a three hour long task. If the hindenbug manifests they will immediately stop their work. What is the probability that the bug manifests for a given user?
    b. Your program is used by one million users. Use a normal approximation to estimate the probability that more than 10,000 users experience the bug. Use your answer from part (a).

7. Say we have a cable of length $n$. We select a point (chosen uniformly randomly) along the cable, at which we cut the cable into two pieces. What is the probability that the shorter of the two pieces of the cable is less than 1/3 the size of the longer of the two pieces? Explain formally how you derived your answer.

## Dithering

8. **[Coding]** Below are two sequences of 300 "coin flips" (H for heads, T for tails). One of these is a true sequence of 300 independent flips of a fair coin. The other was generated by a person typing out H's and T's and trying to *seem* random. Which sequence is truly composed of coin flips?

   We'll save you a bit of time by telling you that both sequences have 148 heads, two less than the expected number for a 0.5 probability of heads. It won't be as simple as finding out which one is closer to half heads! Make an argument that is justified with probabilities calculated on the sequences. This problem is solvable without code, but it would require some tedious counting. You're encouraged to put your computer to good use by looking at these sequences in the accompanying `pset3.zip`.

   Sequence 1:

   ```
   TTHHTHTTHTTTHTTTHTTTHTTHTHHTHHTHTHHTTTHHTHTHTTHTHH
   TTHTHHTHTTTHHTTHHTTHHHTHHTHTTHTHTTHHTHHHTTHTHTTTHH
   TTHTHTHTHTHTTHTHTHHHTTHTHTHHTHHHTHTHTTHTTHHTHTHTHT
   THHTTHTHTTHHHTHTHTHTTHTTHHTTHTHHTHHHTTHHTHTTHTHTHT
   HTHTHTHHHTHTHTHTHHTHHTHTHTTHTTTHHTHTTTHTHHTHHHHTTT
   HHTHTHTHTHHHTTHHTHTTTHTHHTHTHTHHTHTTHTTHTHHTHTHTTT
   ```

   Sequence 2:

   ```
   HTHHHTHTTHHTTTTTTTTHHHTTTHHTTTTHHTTHHHTTHTHTTTTTTH
   THTTTTHHHHTHTHTTHTTTHTTHTTTTTHTHHTHHHHTTTTTHHHHTHHH
   TTTTHTHTTHHHHTHHHHHHHHTTHHTHHTHHHHHHHTTHTHTTTHHTTT
   THTHHTTHTTHTHTHTTHHHHHTTHTTTHTHTHHTTTTHTTTTTHHTHTH
   HHHTTTTHTHHHTHHTHTHTHTHHHTHTTHHHTHHHHHHTHHHTHTTTHH
   HTTTHHTHTTHHTHHHTHTTHTTHTTTHHTHTHTTTTHTHTHTTHTHTHT
   ```

## Hashing

9. Say there are $k$ buckets in a hash table. Each new string added to the table is hashed to bucket $i$ with probability $p_i$, where $\sum_{i=1}^{k} p_i = 1$. If $n$ strings are hashed into the table, find the expected number of buckets that have at least one string hashed to them. (Hint: Let $X_i$ be a binary variable that has the value 1 when there is at least one string hashed to bucket $i$ after the $n$ strings are added to the table (and 0 otherwise). Compute $E\left[\sum_{i=1}^{k} X_i\right]$.)

*Continued on the next page...*

10. A Bloom filter is a probabilistic implementation of the *set* data structure, an unordered collection of unique objects. In this problem we are going to look at it theoretically. Our Bloom filter uses 3 different independent hash functions $H_1$, $H_2$, $H_3$ that each take any string as input and each return an index into a bit-array of length n. Each index is equally likely for each hash function.

    To add a string into the set, feed it to each of the 3 hash functions to get 3 array positions. Set the bits at all these positions to 1. For example, initially all values in the bit-array are zero. In this example $n = 10$:

| Index: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Value: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

    After adding a string "pie", where $H_1$("pie") = 4, $H_2$("pie") = 7, and $H_3$("pie") = 8:

| Index: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Value: | 0 | 0 | 0 | 0 | **1** | 0 | 0 | **1** | **1** | 0 |

    Bits are never switched back to 0. Consider a Bloom filter with $n = 9,000$ buckets. You have added $m$ = 1,000 strings to the Bloom filter. Provide a **numerical answer** for all questions.

    a. What is the (approximated) probability that the first bucket has 0 strings hashed to it?

    To *check* whether a string is in the set, feed it to each of the 3 hash functions to get 3 array positions. If any of the bits at these positions is 0, the element is not in the set. If all bits at these positions are 1, the string *may* be in the set; but it could be that those bits are 1 because some of the other strings hashed to the same values. You may assume that the value of one bucket is independent of the value of all others.

    b. What is the probability that a string which has *not* previously been added to the set will be misidentified as in the set? That is, what is the probability that the bits at all of its hash positions are already 1? Use approximations where appropriate.

    c. Our Bloom filter uses three hash functions. Was that necessary? Repeat your calculation in (b) assuming that we only use a single hash function (not 3).

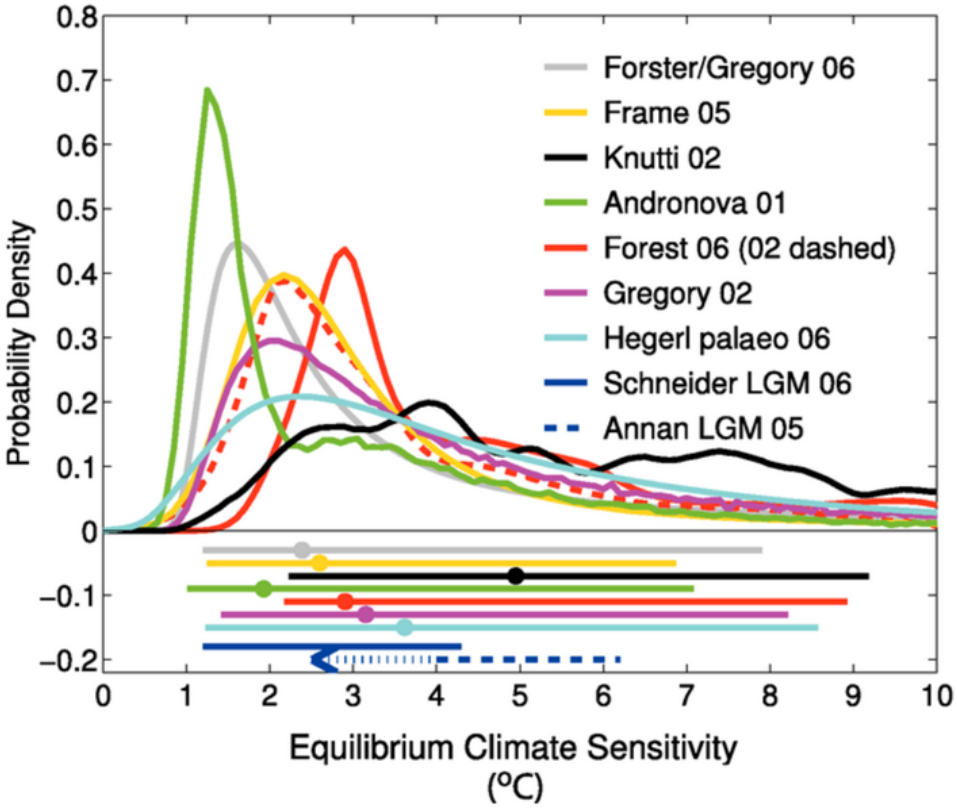
    (Chrome uses a Bloom filter to keep track of malicious URLs. Questions such as this allow us to compute appropriate sizes for hash tables in order to get good performance with high probability in applications where we have a ballpark idea of the number of elements that will be hashed into the table.)

## Climate Change

11. This summer (May 2019) the concentration of $CO_2$ in the atmosphere was 414 parts per million (ppm) which is substantially higher than the pre-industrial concentration: 275 ppm. $CO_2$ is a greenhouse gas and as such increased $CO_2$ corresponds to a warmer planet.

    Absent some pretty significant policy changes, we will reach a point within the next 50 years (i.e., well within your lifetime) where the $CO_2$ in the atmosphere will be double the pre-industrial level. In this problem we are going to explore the following question: What will happen to the global temperature if atmospheric $CO_2$ doubles?

    The measure, in degrees Celsius, of how much the global average surface temperature will change (at the point of equilibrium) after a doubling of atmospheric $CO_2$ is called "Climate Sensitivity." Since the earth is a complicated ecosystem climate scientists model Climate Sensitivity as a random variable, $S$. The IPPC Fourth Assessment Report had a summary of 10 scientific studies that estimated the PDF of $S$:




    In this problem we are going to treat $S$ as part-discrete and part-continuous. For values of $S$ less than 7.5, we are going to model sensitivity as a discrete random variable with PMF based on the average of estimates from the studies in the IPCC report. Here is the PMF for $S$ in the range 0 through 7.5:

| Sensitivity, $S$ (degrees C) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Expert Probability | 0.00 | 0.11 | 0.26 | 0.22 | 0.16 | 0.09 | 0.06 | 0.04 |

The IPCC fifth assessment report notes that there is a non-negligible chance of S being greater than 7.5 degrees but didn't go into detail about probabilities. In the paper "Fat-Tailed Uncertainty in the Economics of Catastrophic Climate Change" Martin Weitzman discusses how different models for the PDF of Climate Sensitivity (S) for large values of S have wildly different policy implications.

For values of S greater than or equal to 7.5 degrees Celsius, we are going to model S as a continuous random variable. Consider two different assumptions for S when it is at least 7.5 degrees Celsius: a fat tailed distribution ($f_1$) and a thin tailed distribution ($f_2$):

$$f_1(x) = \frac{K}{x} \text{ s.t. } 7.5 \leq x < 30$$

$$f_2(x) = \frac{K}{x^3} \text{ s.t. } 7.5 \leq x < 30$$

For this problem assume that the probability that $S$ is greater than 30 degrees Celsius is 0.

a. Compute the probability that Climate Sensitivity is at least 7.5 degrees Celsius.
b. Calculate the value of $K$ for both $f_1$ and $f_2$.
c. It is estimated that if temperatures rise more than 10 degrees Celsius, all the ice on Greenland will melt. Estimate the probability that S is greater than 10 under both the $f_1$ and $f_2$ assumptions.
d. Calculate the expectation of S under both the $f_1$ and $f_2$ assumptions.
e. Let $R = S^2$ be a crude approximation of the cost to society that results from $S$. Calculate $E[R]$ under both the $f_1$ and $f_2$ assumptions.

Notes: (1) Both $f_1$ and $f_2$ are "power law distributions". (2) Calculating expectations for a variable that is part discrete and part continuous is as simple as: use the discrete formula for the discrete part and the continuous formula for the continuous part.

*Continued on the next page. . .*

## Helicommuting

12. **[Coding]** Let's say you operate a helicopter commuting business with once-a-day flights each way between Oakland and Palo Alto. You notice that not all of your paid-up passengers show up for their flights. So you may be able to get away with selling more tickets than there are seats. In this problem, you will determine how much you should overbook your flights to maximize your revenue.

    Your helicopter has 6 passenger seats and you sell tickets for $195 each. Your data tell you that the demand for tickets for each flight is a Poisson random variable with expectation of 7.5. You also assume that each paid-up passenger will show up on time independently with probability 65%. If a passenger does not show up on time, you get to keep the $195 fare. On the other hand, if more than 6 passengers show up for a flight, you compensate the ones who don't get to fly with $950 each (which includes a refund of the fare).

    Write a program to determine the number of tickets $n$ you should offer for sale for each flight in order to maximize your expected revenue, where revenue is ticket sales minus compensation for overbooking. (Notice that if you offer $n$ tickets for sale but the demand turns out to be less than $n$, then you are left with unsold tickets. If the demand turns out to be greater than $n$, some potential customers do not get to purchase tickets.)

    Your program should simulate the value of revenue for at least 100,000 trials for each value of $n$. Plot expected revenue vs. $n$ (for $6 \leq n \leq 15$) and report the optimal $n$.

    This problem was inspired by `https://blade.flyblade.com/p/san-francisco-bay-area`.

    **Python tips:** We recommend that you use NumPy library functions such as `numpy.minimum()`, `numpy.random.poisson()` (or SciPy's `scipy.stats.poisson()`, etc.

    If your Python code uses nested loops (one for $n$ and one for the trials), your simulation may take minutes to complete. If you have a loop for $n$ and use vectorized operations instead of the other loop, your simulation may finish in seconds (or even less than a second). In this problem, a vectorized solution means that most lines of code will operate on arrays of size 100,000. That is, for each value of $n$ the program processes all 100,000 trials together.

    We will accept vectorized or non-vectorized programs for full credit. While the speed-up may not make a difference here, vectorization will be more important when you train ML models later in the semester. More info: `https://realpython.com/numpy-array-programming`