CS109A Week 6 Notes

Ian Tullis

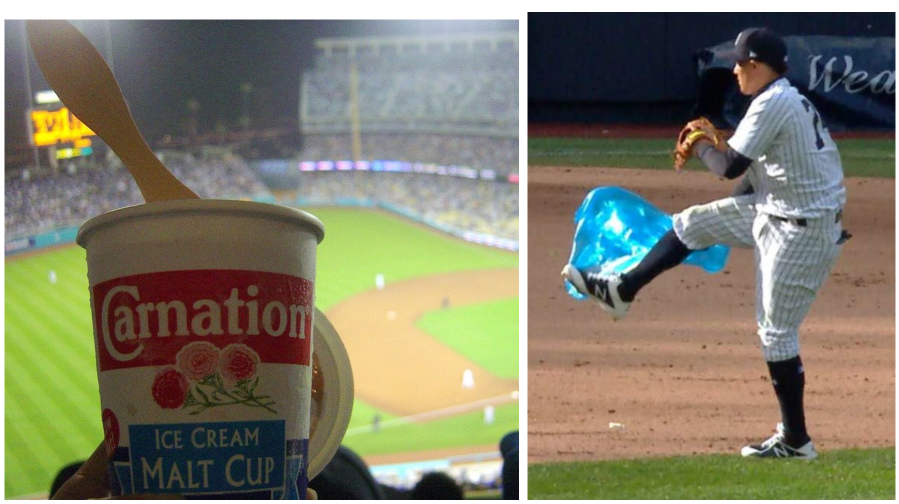
May 3, 2022

I. Making The World Series Fair

In honor of the return of baseball season, let's work through a baseball problem together!

Problem 1. *The World Series is only 7 games, which is not enough to decide which of two teams of similar ability is actually better. But the format is presumably designed to create drama and excitement, not to select the worthiest champion!*

Suppose that in any given World Series game, the Dodgers have a true probability of 0.51 of beating the Giants (apologies to Giants fans!), independently of the results of any other games. What is the minimum *odd* number of games that the World Series would need to have in order for there to be less than a 1% chance of the Giants winning a majority of the games? (The number of games must be odd to avoid an overall tie.) **Use a normal approximation.** You may (and should) use the fact that $\Phi(2.3263) = 0.99$.



The two most exciting things in baseball: 1. getting a chocolate malt, and 2. plastic bag on the field.

Solution to Problem 1. Let W be a random variable representing the number of wins the Giants (not the Dodgers) get in n games, for some value of n that we have yet to find. The true distribution of W is binomial:

$$P(W = w) = \binom{n}{w}(\frac{49}{100})^w(\frac{51}{100})^{n-w}$$

Per the usual formulas, the mean and variance are $np = \frac{49n}{100}$ and $np(1-p) = \frac{2499n}{10000}$, so we will use those as μ and σ^2 in our normal approximation.

Then the probability that the Giants win over half the games is

$$P(W > \frac{n}{2}) = 1 - \Phi(\frac{\frac{n}{2} - \frac{49n}{100}}{\sqrt{\frac{2499n}{10000}}})$$

Note the absence of a continuity correction, even though the number of wins is a discrete value (and there are also no ties in the World Series! See Rule 7.02(a) Comment in the MLB Official Baseball Rules.) This is because we are specifically assuming that n is odd. Suppose, for example, that $n = 7$; then values of 3.4 and 3.6 in continuous-land would already fall into the 3 and 4 buckets, respectively, in discrete-land, which is exactly what we want.

For the above probability to be 0.01, we need $\Phi(\frac{\frac{n}{2} - \frac{49n}{100}}{\sqrt{\frac{2499n}{10000}}}) = 0.99$. Using the provided piece of information about Φ, we know that $\frac{\frac{n}{2} - \frac{49n}{100}}{\sqrt{\frac{2499n}{10000}}} = 2.3263$. Solving this for n (which involves squaring to get rid of the $\sqrt{n}$ and then using the quadratic equation – again, you would probably not have to do this by hand on an exam, and I just used Wolfram Alpha), we find that $n \approx 13523.8$. Because we are requiring n to be odd, the next largest odd value is $\boxed{13525}$. Since Ian eats at least one chocolate malt per baseball game, that's a lot of chocolate malts, especially if the Series makes it all the way to that crucial game 13525...

Let's double-check that that makes sense. The mean is 6627.5 and the variance is 3379.8975. The probability that the Giants win more than half the games in the series – i.e. 6762.5 – is $1 - \Phi(\frac{6762.5-6627.5}{\sqrt{3379.8975}}) = 1 - \Phi(2.322) = 0.01$.

What is the real answer without the approximation? We want the smallest odd n such that

$$P(W_{\text{Giants}} > \lceil \frac{n}{2} \rceil) = \sum_{w=\lceil \frac{n}{2} \rceil}^{n} \binom{n}{w}(\frac{49}{100})^w(\frac{51}{100})^{n-w} < 0.01$$

This was extremely painful for my computer to check in Python (check out the `Decimal` library if you need to work with gigantic factorials, and consider using PyPy), but it found the answer 13527. So the normal approximation was very close! (and the error probably came from how I provided 2.3263 in a truncated form, not from the failure of the Central Limit Theorem)

II. Beepworld

The geometric, negative binomial, Poisson, and exponential distributions are all intimately related. To better understand them, let's imagine a very annoying place: Beepworld.

Beepworld looks like the blank expanse from The Matrix or The Good Place, except that there is a single smoke detector. Fire safety is important, even in the featureless Beepworld! But it must not be *that* important, because the battery is dying and the smoke detector is beeping.

It's hard to tune out; it's not even beeping at fixed intervals. As we listen, we conjecture that each second, the smoke detector decides whether to beep: with probability 0.1, it beeps, and with probability $1 - 0.1 = 0.9$, it does not.

Because we are stuck in Beepworld and literally have nothing to do but study the beeping smoke detector, we figure we might as well ask some CS109 questions about it. If this is Hell, we might as well do probability!

How long an expected window until the next beep?

First, let's find the expected amount of time until the next beep. This will consist of some nonnegative integer number of milliseconds with no beep, followed by one millisecond with a beep. So this amount of time follows a geometric distribution: if X is the time in seconds until (and including) the next beep, $P(X = x) = (1-p)^{x-1} \cdot p$. We know from the Week 4 109A notes that the expectation of a geometric distribution is $\frac{1}{p}$, so in this case, it is $\frac{1}{0.1} = 10$.

Notice that this same logic holds *no matter when we ask the question*! The *expected* amount of seconds until the next beep is *always* 10.

Remember that "expected amount" means that if we were to repeat this experiment infinitely many times, the average number of seconds until the next beep would be 10. But this is **not a guarantee** that we hear a beep every 10 seconds, nor does it mean that the probability of hearing a beep in the next 10 seconds is $\frac{1}{2}$. We know from the geometric distribution that the probability that we hear at least one beep in the next 10 seconds is $P(X = 1) + ... + P(X = 10)$, because that first beep has to occur in one of those 10 seconds, regardless of whether there end up being any other beeps later. This turns out to be ≈ 0.6513.

What is the expected amount of time until we hear a total of *two* more beeps? One way to find this is to harness the awesome power of linearity of expectation (see the Week 3 109A notes). Let X be the time in seconds until the next beep, and let Y be the time in seconds from that first beep until the next beep after that. Both X and Y have the same (geometric) distribution – again, that expectation is always the same no matter when we ask the question – so $\mathbf{E}[X + Y] = \mathbf{E}[X] + \mathbf{E}[Y] = 10 + 10 = 20$.

Another option is to use the negative binomial distribution, which exists to answer exactly this sort of question. I won't dwell on the details here since the negative binomial is not a huge focus in CS109, but you can find this distribution (and its expectation, etc.) on slide 17 of lecture 9. The amount of time X until two beeps has a negative binomial distribution with $p = 0.1, r = 2$, and so the expectation is $\frac{r}{p} = \frac{2}{0.1} = 20$, the same answer as before.

How many beeps do we expect in a certain window?

Now let's ask a different question: given a time window of a particular size – say, 20 seconds – what is the expected number of beeps?

One approach is to take the expectation of a sum of indicator random variables (109A Notes 3). Let $B_1, ..., B_{20}$ be the events corresponding to beeps at seconds 1 through 20, and $I_1, ..., I_{20}$ be the corresponding indicator RVs. Then the total number of beeps is $I_1 + ... + I_{20}$, and the expected number of beeps is $\mathbf{E}[I_1 + ... + I_{20}]$. By linearity of expectation, this is $\mathbf{E}[I_1] + ... + \mathbf{E}[I_{20}]$. Since the expectation of an indicator variable is just the probability of its corresponding event, each of these $\mathbf{E}[I_1]$ values is 0.1, so the total is 2. This is consistent with what we found before! If the expected amount of seconds needed to hear 2 beeps is 20, the expected number of beeps in 20 seconds should be 2.

Another approach is to observe that the number of beeps in a given window has a binomial distribution: each of the 20 seconds is an independent "trial" ($n = 20$), and the "success" probability (p) for each one is 0.1. Recalling (or double-checking) that the expectation of a binomial distribution is $n \cdot p$, we again get $20 \cdot 0.1 = 2$.

III. Millisecond Beepworld

It occurs to us that a second is a pretty large chunk of time for an electronic device. What if the smoke alarm actually decides every *millisecond* whether to go off, but with a new beep probability p' that keeps the overall rate of beeping the same? (That is, 1 beep per 10 seconds, in expectation.)

Now it is possible that the smoke alarm might beep multiple times in the same second, which could not have happened in Original Beepworld with its one-second granularity. But the rules of the system are otherwise the same. In particular, the number of beeps in 10 seconds is still governed by a binomial distribution, but one with different parameters...

Problem 2.

(a) What is the value n' for our new binomial distribution (for the number of beeps in 10 seconds), now that our granularity is a millisecond?

(b) What is the value p' for our new binomial (or, equivalently, geometric) distribution, if the expected number of beeps per 10 seconds is still 1?

(c) What is the expected amount of time needed to hear one beep? Two beeps? Give your answers in seconds, just for easier comparison with Original Beepworld.

(d) Using the geometric distribution, write (but don't evaluate) an expression for the probability that we hear at least one beep in the next 10 seconds. Your expression should be in terms of p' and should include a summation.

(e) The expression in part (d) turns out to evaluate to ≈ 0.6321, which is different from the value of ≈ 0.6513 that we found for Original Beepworld. Why might we expect this value for Original Beepworld to be different from – but not *that* different from – the value for Millisecond Beepworld? (Feel free to skip this one – it's tough to reason about!)

(f) Using the binomial distribution (or a similar method), find the probability that we hear at least one beep in the next 10 seconds. (Hint: Rather than a summation, use our typical trick for "at least one", and a calculator.)

(g) Would you expect the answer to (d) and the answer to (f) to be the same? Are they?

Solutions to Problem 2.

(a) There are 10000 milliseconds (and thus 10000 separate "trials") in 10 seconds, so $n = \boxed{10000}$.

(b) For the expected number of beeps per 10 seconds (10000 milliseconds) to equal 1, we need $n'p' = 1$. Therefore $p' = \frac{1}{n'} = \boxed{\dfrac{1}{10000}}$. This also makes sense intuitively: if we are allowing 1000 times as many trials, each one needs to succeed $\frac{1}{1000}$-th as often for the expected number of successes to stay the same.

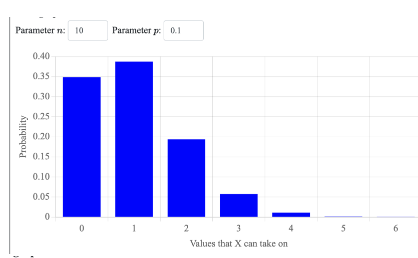
(c) By the geometric distribution, the expected amount of milliseconds needed to hear one beep is $\frac{1}{p'} = 10000$, i.e. $\boxed{10 \text{ seconds}}$. The expected amount of milliseconds needed to hear two beeps is $\frac{1}{p'} + \frac{1}{p'} = 20000$, i.e. $\boxed{20 \text{ seconds}}$. So these values have not changed, even though we changed the granularity.

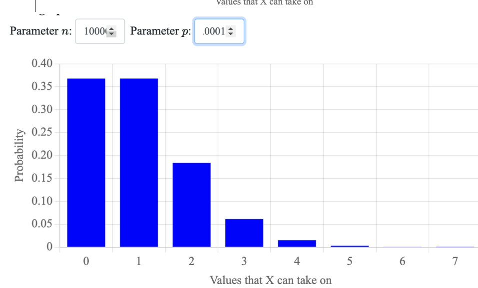
(d) Using the geometric distribution, $P(\text{at least 1 beep in the next 10 seconds})$

$$= P(\text{at least 1 beep in the next 10000 milliseconds})$$

$$= P(X = 1) + ... + P(X = 10000) = \boxed{\sum_{x=1}^{10000} (1 - p')^{x-1} p'}$$

(e) If we compare the binomial distributions from Original Beepworld and Millisecond Beepworld, for example, we notice that they are similar but not identical:





As a more concrete example, think of the distributions for flipping two vs. four fair coins. Both have an expected value of $np = \frac{n}{2}$ heads, but the distributions are not the same. As we flip more and more coins, the distribution gets more and more of a classic binomial curve shape, and the distributions with smaller n are almost like clunky, Minecraft-esque approximations thereof.

(f) $P(X \geq 1) = 1 - P(X = 0) = 1 - \binom{10000}{0}(p')^0(1-p')^{10000} = 1 - 1 \cdot 0.9999^{10000} \approx \boxed{0.6321}$.

(g) Because the geometric and binomial distributions are expressions of the same underlying model of "trials", with the same success probability, they should (and do) give exactly the same result; neither is an approximation.

IV. Continuous Beepworld

Many real-world phenomena, like the decay of radioactive atoms, do not operate on such large discrete time scales. We can try to model this by making the Beepworld time granularity even smaller, all while keeping the expected number of beeps at 1 per 10 seconds.

As we make this granularity arbitrarily tiny, the binomial distribution of the number of beeps within a time window gets hard to work with: p becomes vanishingly small, n becomes enormous, and $\binom{n}{x}$ becomes very hard to evaluate. Fortunately, as we saw in lecture, this distribution approaches a limit that is much easier to work with: a Poisson distribution.

Let's try to use the Poisson to find the probability that we hear at least one beep in the next 10 seconds. As always with Poissons, our first step is to find λ; as detailed in the Week 4 109A notes, this depends on our choice of time window. Here we are using a window of 10 seconds, and we already conveniently know the rate is 1 beep per 10 seconds, so $\lambda = 1$.

Then the probability of no beeps ($X = 0$) is $\frac{e^{-\lambda}\lambda^0}{0!} = e^{-1} = \frac{1}{e}$, so the probability of at least one beep is $1 - \frac{1}{e} \approx 0.6321$, the same answer we kept seeing in Millisecond Beepworld!

What does this tell us? The millisecond-level granularity was a pretty close approximation to continuous time! So we could have just as well used a Poisson distribution for Millisecond Beepworld. But it would *not* have been a perfect approximation for Original Beepworld.

What about our geometric distribution for the amount of time X until the next beep? Fortunately, this also approaches a limit as the time unit grows smaller: the **exponential distribution**. Specifically, we have

$$f(X = x) = \lambda e^{-\lambda x}$$

where λ is (conveniently enough) the same λ from the Poisson distribution, with the same meaning.

Remember that unlike a probability mass function (for a discrete variable), evaluating a probability *density* function (for a continuous variable) at some value $X = x$ does *not* give the probability that $P(X = x)$. This is because for a continuous variable, it does not even make sense to ask, for example, "what is the probability that $X = 5$?" If we imagine arbitrarily small time units, the chance of it taking *exactly* this amount of time becomes vanishingly close to 0.

Rather, to get a probability out of a PDF, we have to integrate under a region of the function. We can only ask questions like "what is the probability that X is between 4.5 and 5.5?" or "what is the probability that X is greater than 7?", and answer them by taking a definite integral of the PDF from 4.5 to 5.5, or from 7 to infinity.

Now our running example question of "what is the probability that we hear at least one beep in the next 10 seconds?" is equivalent to "what is the probability that an exponential random variable is between 0 and 10?"[1] But just as with the Poisson distribution, we have to be careful with how we define λ. If we are looking at an interval of 1 second, then our rate λ is only 0.1 expected beep per second, and so the answer is

$$\int_0^{10} \lambda e^{-\lambda x} dx = \int_0^{10} 0.1 e^{-0.1x} dx$$

In general, the definite integral of $e^{kx}dx$ (for some constant k) is $\frac{1}{k}e^{kx}$, so our answer here is

$$0.1[\frac{1}{-0.1}e^{-0.1x}]_0^{10} = 0.1(-10e^{-0.1\cdot 10} - (-10e^{-0.1\cdot 0})) = 0.1(-10e^{-1} + 10e^{0}) = 1 - \frac{1}{e} \approx 0.6321$$

as ever.

On the other hand, if we decide use a window of 10 seconds – that is, if we say that "1" means 10 seconds – then $\lambda = 1$, and the expression is $\int_0^1 e^{-x}dx$. This yields the same answer, but it is perhaps easier on our brains to keep 1 as meaning "1 second"!

Let's also think about the expectation of the exponential distribution. The expectation of a discrete random variable is a weighted average over the values

[1] Normally I am careful about specifying inclusive or exclusive limits on ranges, but for continuous variables, it doesn't matter. Because $P(X = 0) = 0$ and $P(X = 10) = 0$, $P(0 \leq X \leq 10)$ is the same as $P(0 < X < 10)$.

it can take on, where the weights are the probabilities of those values. The expectation of a continuous random variable is conceptually the same, except now there are (probably) arbitrarily many values it can take on. So instead of $\sum_{\text{all possible } x} x \cdot p(x)$, we now have $\int_{\text{support of } x} x \cdot f(x)dx$.

The exponential distribution is supported on (i.e., can take on values in) the range from 0 to infinity, so its expectation is

$$\int_0^\infty x \cdot f(x)dx = \int_0^\infty \lambda x e^{-\lambda x} dx$$

This is not an integral that Jerry would expect you to handle on an exam, so feel free to skip this paragraph! We can solve it via "integration by parts", i.e. the $\int udv = uv - \int vdu$ trick. Perhaps after some trial and error, we pick $u = \lambda x, dv = e^{-\lambda x}dx$, and therefore $du = \lambda dx$ and $v = -\frac{1}{\lambda}e^{-\lambda x}$. Now we have $uv - \int vdu = -xe^{-\lambda x} + \int e^{-\lambda x}dx = -(x + \frac{1}{\lambda})e^{-\lambda x}$. Evaluating this from 0 to infinity gives $0 - -(0 + \frac{1}{\lambda}) = \frac{1}{\lambda}$.

So the expectation of an exponential distribution with parameter λ is $\frac{1}{\lambda}$. It makes sense that if we have a rate of $\frac{1}{10}$ beeps per minute, for instance, then we expect to have to wait 10 minutes to hear a beep. Again, note the strong parallel with the geometric distribution with parameter p and expectation $\frac{1}{p}$.

Problem 3. Here's some practice relating exponential and Poisson distributions. Suppose the random variables X_1, X_2, X_3, X_4 are independent, and each has an exponential distribution with the same parameter $\lambda = 2$. Let $Y = X_1 + X_2 + X_3 + X_4$. First let's warm up with some properties of this new distribution:

(a) Is it true that $f_Y(y) = f_X(\frac{y}{4})$? (i.e. that the value of the PDF of Y at $Y = y$ is the value of the PDF of X at $X = \frac{y}{4}$)

(b) What is $\mathbf{E}[Y]$? (Hint: do not try to use an integral!)

(c) Show that $f_Y(0) = 16$. (Hard, and involves thinking about joint distributions, which we'll cover later – feel free to skip.)

(d) The answer to (c) is larger than 1. Why does it not violate rules for probability distributions when a PDF exceeds 1?

(e) What is $P(Y = 0)$?

Now, in parts (f)-(h), suppose we want to find the *exact* probability that $Y \leq 2$, without using approximations.

(f) Explain how to envision a Poisson-distributed variable Z such that $Y \leq 2$ corresponds exactly to $Z \geq 4$.

(g) What would be the rate λ' of that Poisson distribution?

(h) Using that Poisson distribution, what is the exact value of $P(Y \leq 2)$?

Solutions to Problem 3. If we try to solve this just in terms of the sum of exponentials, it gets pretty hairy to think about! Nor is a normal distribution a good approximation for the sum of a few exponentials. A sum of independent and identically distributed exponentials *does* have something called a gamma distribution, but this is a niche topic in CS109.

(a) $\boxed{\text{No}}$. Y is the sum of four independent exponential random variables, not the result of taking one exponential random variable and multiplying it by 4. However, if Y were defined as $4X$, this statement would be true.

(b) By linearity of expectation, $\mathbf{E}[Y] = \mathbf{E}[X_1 + X_2 + X_3 + X_4] = \mathbf{E}[X_1] + \mathbf{E}[X_2] + \mathbf{E}[X_3] + \mathbf{E}[X_4] = \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = \boxed{2}$.

(c) Since exponential distributions cannot take on negative values, we can only have $Y = 0$ if X_1, X_2, X_3, X_4 all equal 0. The X_is are independent of each other, so $f_Y(0) = f_{X_1}(0)f_{X_2}(0)f_{X_3}(0)f_{X_4}(0)$. Since all of $X_1, ..., X_4$ are identically distributed, this equals $(f_{X_1}(0))^4$. By the exponential distribution, $f_{X_1}(0) = \lambda e^{-\lambda x} = 2e^{-2 \cdot 0} = 2$, so $f_Y(0) = 16$.

(d) A PDF can take on values larger than 1, since values of PDFs are not probabilities. However, the PDF must *integrate* to 1, and be nonnegative everywhere.

(e) For a PDF, for any specific value x, $P(X = x) = \boxed{0}$.

(f) The exponential distributions X_1, ..., X_4 can each be viewed as measuring the time until some occurrence happens (following the previous one, in the case of X_2 through X_4.) Y is the total time for all four to happen back-to-back. So $Y \leq 2$ means that all four occurrences happened by (or before) the 2 second mark.

Consider a Poisson distribution for the number Z of these occurrences in a 2-second window, with the "clock" for one occurrence starting when the previous one stops. (This is called a "Poisson process".) Then $Y \leq 2$ means that four occurrences happened within the 2-second window. And maybe some other occurrences happened after that as well! So $Y \leq 2$ corresponds exactly to $Z \geq 4$ for this Poisson distribution.

(g) Since $\lambda = 2$, the expected time for each of $X_1, ..., X_4$ is $\frac{1}{2} = 0.5$ seconds. Then we would expect 4 events in 2 seconds, so $\lambda' = \boxed{4}$ for our Poisson distribution.

(h) Since $P(Z \geq 4) = P(Z = 4) + P(Z = 5) + ...$ ad infinitum, it is easier to find $P(Z = 0) + P(Z = 1) + P(Z = 2) + P(Z = 3)$. So the final answer is

$$1 - \sum_{k=0}^{3}\left(\frac{e^{-4}4^k}{k!}\right) = 1 - e^{-4}\left(1 + 4 + 8 + \frac{32}{3}\right) = \boxed{1 - \frac{71}{3e^4}}.$$

For many of us, including myself, coding is believing. I got the result 0.56657, which is very close to the true value of 0.56653.

```
# https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.expon.html
from scipy.stats import expon
total = 0
TRIALS = 100000000
for i in range(TRIALS):
    # in scipy, scale = 1/lambda
    r = expon.rvs(scale=0.5, size=4)
    if sum(r) <= 2.0:
        total += 1
print(total / TRIALS)
```

V. CDFs and Inverse CDFs

Every PDF $f(x)$ has a corresponding CDF $F(y)$, which is defined as $\int_{\text{smallest supported value}}^{y} f(x)dx$. (We use y to make it clear that we are referring to a specific value on the x-axis, and not to the x variable that we are integrating over.) For example, for the exponential distribution, the smallest supported value is 0, so the CDF is

$F(y) = \int_0^y \lambda e^{-\lambda x} dx = \lambda[-\frac{1}{\lambda}e^{-\lambda x}]_0^y = -e^{\lambda \cdot y} + e^{\lambda \cdot 0} - = 1 - e^{\lambda y}$.

The point of the CDF is to do the integration once so that we never need to do it again. If we want $P(0 \leq X \leq 3)$, now we can simply take $F(3)$. If we want $P(1 \leq X \leq 2)$, we have to be a bit sneakier: first take $F(2)$, then get rid of the unwanted area below $X = 1$ by subtracting off $F(1)$.

Although it is not a major part of CS109, it is often useful to find the **inverse CDF** as well. As with other inverse functions, the inverse CDF, $F^{-1}(y)$ tells us: what value of y do I need to pick so that $F(y) = y$? We can find an expression for $F^{-1}(y)$ by swapping $F(y)$ and y in the CDF and rearranging. For example, for the exponential distribution:

$F(y) = 1 - e^{-\lambda y}$ swap!

$y = 1 - e^{-\lambda F^{-1}(y)}$

$1 - y = e^{=\lambda F^{-1}(y)}$

$-\lambda F^{-1}(y) = \log(1 - y)$

$F^{-1}(y) = -\frac{\log(1-y)}{\lambda}$

So what? We can use this to quickly draw a random value from an exponential distribution! All we need to do is choose a uniformly random value y in [0, 1] – which you can think of as being a percentile value – and take $F^{-1}(y)$. For instance, if we happen to pick $y = 0.5$, we find that $F^{-1}(y) = -\frac{\log(1-0.5)}{\lambda} = \frac{\log(2)}{\lambda}$. Non-coincidentally, this is the median of the distribution; 0.5 is the 50th percentile.

Consider the alternative of how we would sample a random value from a continuous distribution. With a discrete distribution, we can find the probability of every value, but here there are infinitely many values and we can't enumerate them all! We still have to choose x proportional to $f(x)$, so essentially we want to throw a dart uniformly at random at the area under the curve of the PDF. That is intractable, so we instead imagine integrating along the curve and stopping at some fraction of the way, where that fraction is chosen uniformly in [0, 1]. This is exactly what the inverse CDF does for us.

Unfortunately, it is not always tractable, or even mathematically possible, to compute an inverse CDF... but when one exists and is easy to find, it can be a valuable tool.

Problem 4. Let's practice with a new distribution of the form $f(x) = kx^{1/2}$, with k being some unknown constant. Suppose that we declare that the distribution is supported on the range [0, 1].

(a) $f(x)$ is defined on $[0, \infty)$, so why can't we support this distribution over that entire range?

(b) What must k be in order for $f(x)$ to be a valid probability distribution? (i.e. for $f(x)$ to integrate, over its support, to 1.) Use this value in the distribution in the remaining parts.

(c) Let X be a variable with this distribution. What is $\mathbf{E}[X]$?

(d) What is $\mathbf{E}[X^2]$?

(e) What is $\mathbf{Var}[X]$? (Hint: use the results of (c) and (d).)

(f) What is the CDF, $F(y)$?

(g) What is the inverse CDF, $F^{-1}(y)$?

(h) What is the median of the PDF? Is it above or below 0.5, or exactly 0.5? Does this match your intuition based on the shape of $f(x)$?

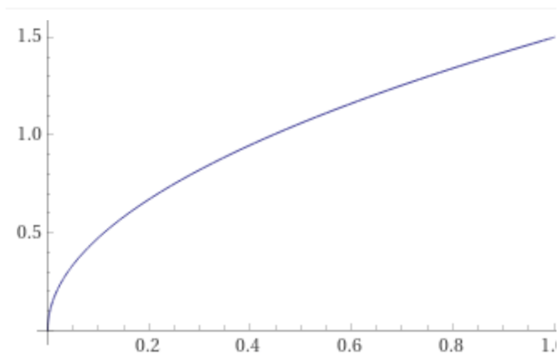
(i) Write Python code to sample a value from $f(x)$. As in Homework 3, Problem 2, your only call should be to `random.uniform(0, 1)`.

(j) What does this distribution look like? Can you think of some potential real-world application for this distribution, given its qualities? (This is open-ended; I don't have a specific answer in mind.)

Solutions to Problem 4.

(a) As x gets arbitrarily large, $f(x)$ also gets arbitrarily large, albeit at a slower rate because of the square root. So we can't integrate over this entire range; the area under the curve would be infinite, and regardless of the value of k, that infinite area could never equal 1.

(b) We want $\int_0^1 kx^{1/2}dx = 1$. The left side is $[\frac{2k}{3}x^{3/2}]_0^1 = \frac{2k}{3}1^{3/2} - \frac{2k}{3}0^{3/2} = \frac{2k}{3}$. For this to equal 1, we need $k = \boxed{\frac{3}{2}}$. So we will use $f(x) = \frac{3}{2}x^{1/2}$ from here on out. Here's what our distribution looks like:



(c) To get $\mathbf{E}[X]$, we take $\int_0^1 xf(x)dx = \int_0^1 x\cdot\frac{3}{2}x^{1/2}dx = \int_0^1 \frac{3}{2}x^{3/2} = [\frac{3}{5}x^{5/2}]_0^1 = \boxed{\frac{3}{5}}$.

(d) Similarly, to get $\mathbf{E}[X^2]$, we take $\int_0^1 xf(x)dx = \int_0^1 x^2\cdot\frac{3}{2}x^{1/2}dx = \int_0^1 \frac{3}{2}x^{5/2} = [\frac{3}{7}x^{7/2}]_0^1 = \boxed{\frac{3}{7}}$.

(e) Using $\mathbf{Var}[X] = \mathbf{E}[X^2] - (\mathbf{E}[X])^2$: $\frac{3}{7} - (\frac{3}{5})^2 = \boxed{\frac{12}{175}}$.

(f) We get the CDF by integrating the PDF up to an arbitrary point y. $F(y) = \int_0^y \frac{3}{2}x^{1/2} = [x^{3/2}]_0^y = \boxed{y^{3/2}}$.

(g) To find the inverse of $F(y) = y^{3/2}$, we swap $F(y)$ and y and replace $F(y)$ by $F^{-1}(y)$: $y = F^{-1}(y)^{3/2}$. Then, raising both sides to the 2/3 power, we have $F^{-1}(y) = \boxed{y^{2/3}}$.

(h) An easy way to get the median is to evaluate the inverse CDF at 0.5. This gives us $0.5^{2/3}$, i.e. $\boxed{\frac{1}{2^{2/3}}}$, which is about 0.63. Looking at the graph above, this is plausible: most of the area under the curve is toward the right, so in order to cut the area exactly in half, the median also needs to be toward the right.

(i) Look how nice this is, thanks to the magic of the inverse CDF!

```
import random
def sample():
  r = random.uniform(0, 1)
  return r**(2/3)
```

(j) The distribution assigns more weight to values closer to 1. One possible use of this would be for, e.g., modeling scores (as a fraction of 1) on a problem set. But there are probably more interesting natural situations where the probability of x (in the range (0, 1)) is proportional to the square root of x.

Appendix: Beepworld Simulator

This works on my Mac. `beep.wav` is some file in the same directory. This is not a perfect simulation since it takes a nontrivial amount of time to actually play the sound, but it simulates the irritating sporadic nature of the beeping pretty well!

```
import os, random, time
while True:
  time.sleep(1)
  if random.uniform(0, 1) <= 0.1:
    os.system("afplay beep.wav")
```