Assignment 3
Simulation-based approximations

3.04.2023

Alperen Guncan
Mark Dragnev

Contents:

Bernoulli generator (1 point)

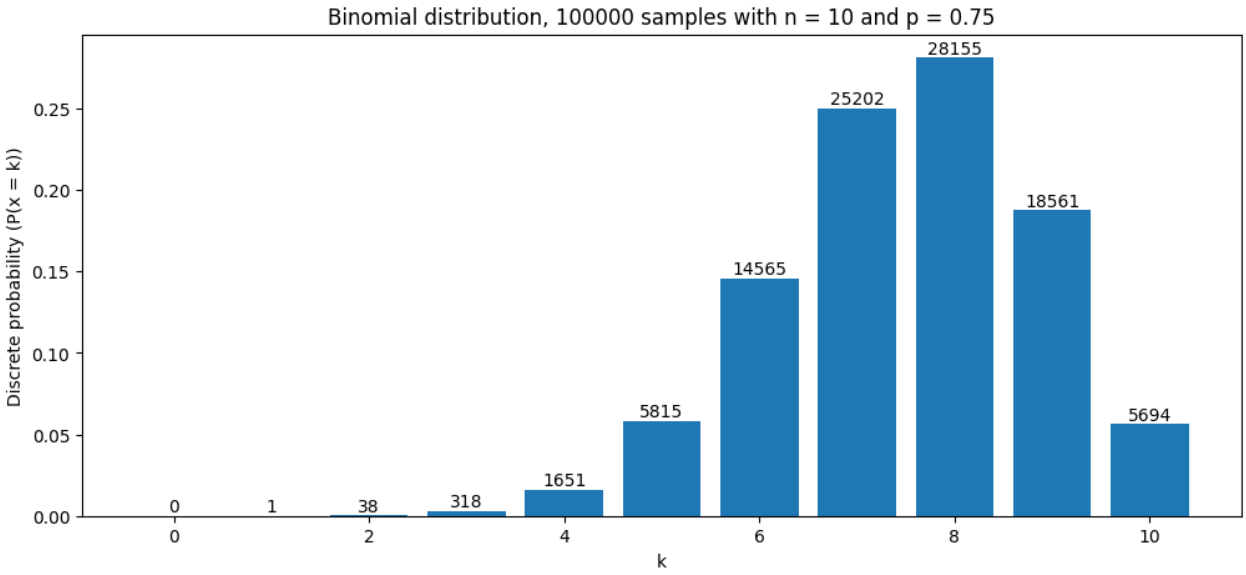
1. a. Bernoulli generator

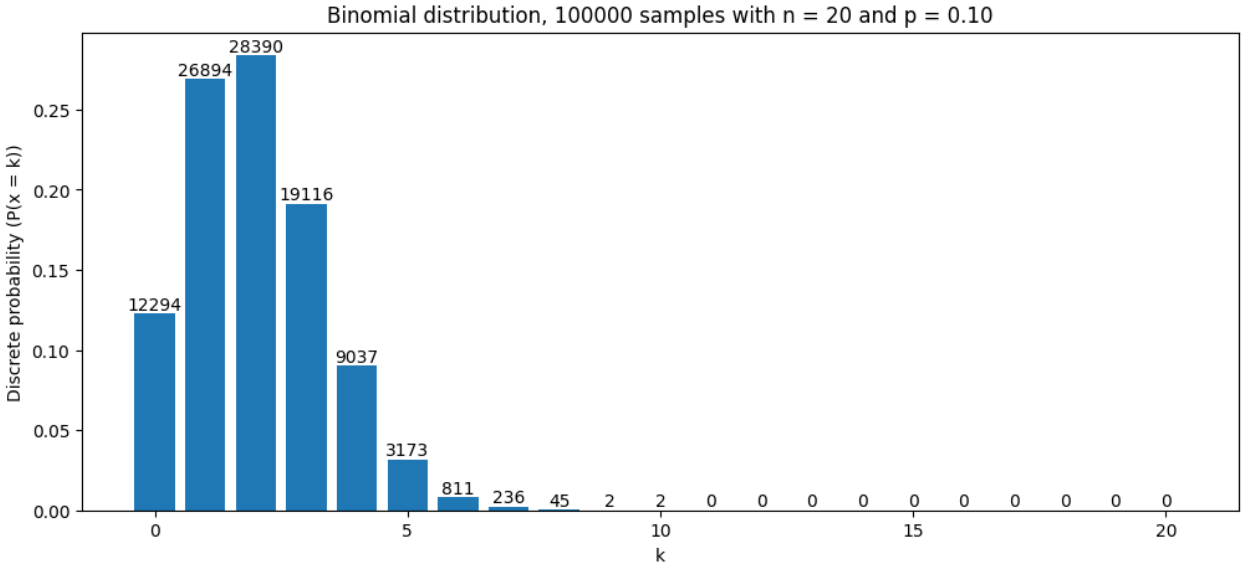
For the completion of this exercise, we start off with implementing a Bernoulli generator. Since the first point of the next exercise also requires the implementation of a Bernoulli generator, we will go into the details of how we do it there.

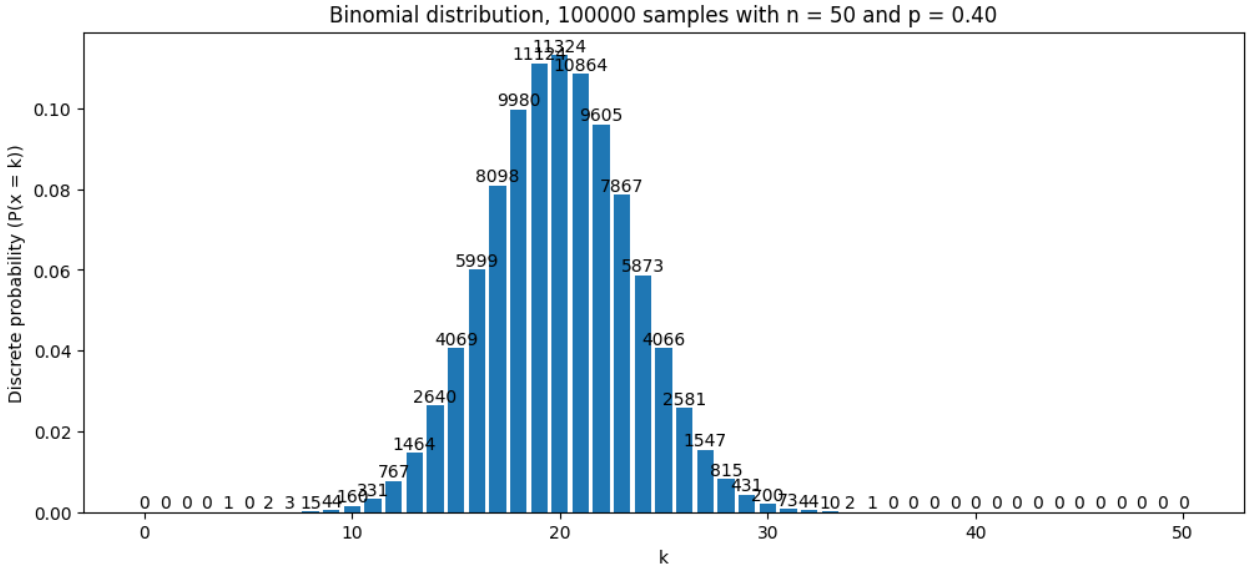
1. b. Binomial distribution

A binomial distribution, with two parameters n and p, is defined as the distribution that models the number of successes of n Bernoulli trials, each with probability p. In other words, a binomial distribution is a generalization of the Bernoulli distribution for more than one experiment. We denote such a variable as $X \sim B(n, p)$, where $Pr(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$ and $\binom{n}{k}$ is a binomial coefficient (from where also comes the name of the theorem that derives this distribution, the Binomial theorem).

The way we generate a binomial distribution is simple enough - just follow the definition. More specifically, we sample n random variables following a Bernoulli distribution and count the number of successes. Below we show sampled distributions of the generator with the aforementioned technique, for different values of n and p (noted on the title of the plots), each with *100,000* samples. A similar format for discrete distributions is to be expected throughout the report.





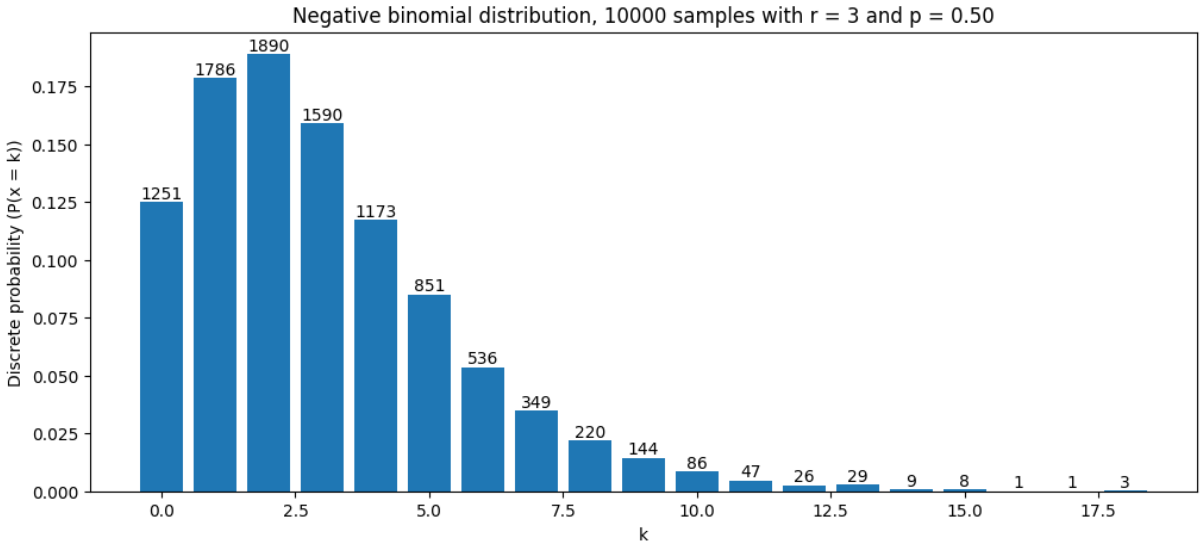


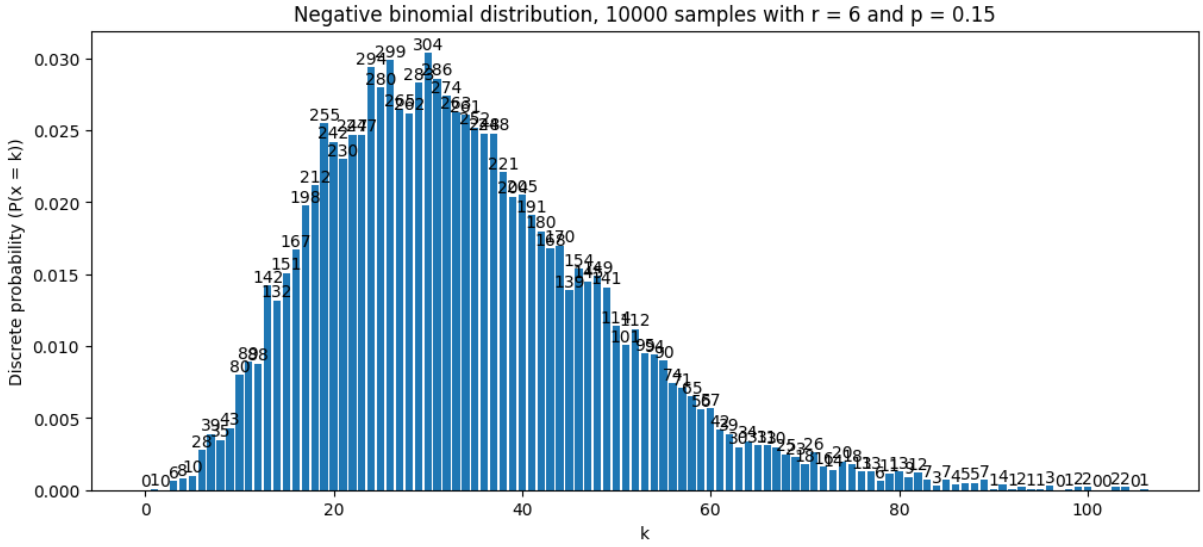
1. c. Negative binomial distribution

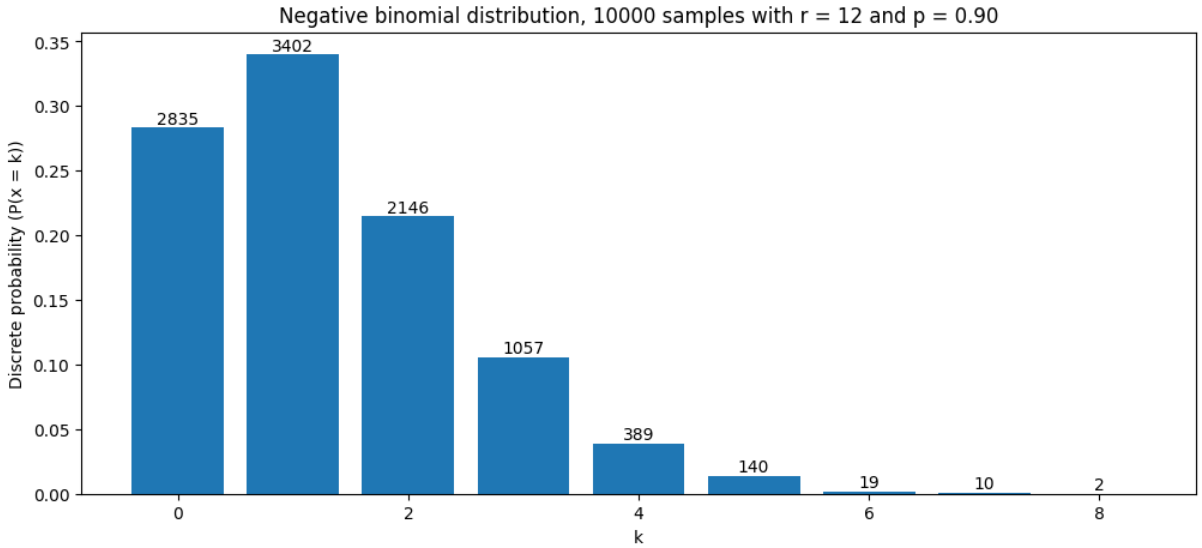
A negative binomial distribution is quite similar conceptually to a binomial one. This distribution, with parameters r and p, models the number of failures until r number of successes, each with probability p. A negative binomial distribution is denoted as $X \sim NB(r, p)$, where $Pr(X = k) = \binom{k+r-1}{k} p^r (1 - p)^k$, and $\binom{n}{k}$ is again a binomial coefficient.

Similarly to a binomial generator, we generate this one by simply following the definition - we generate random Bernoulli trials with probability p until we reach r successes, keeping track of the number of failures by then. This is not the best way of implementing a generator for this distribution, by any stretch of the

imagination, but it works. Below we show the generators in action, with *10 000* samples, and different parameters (again shown on plot titles).

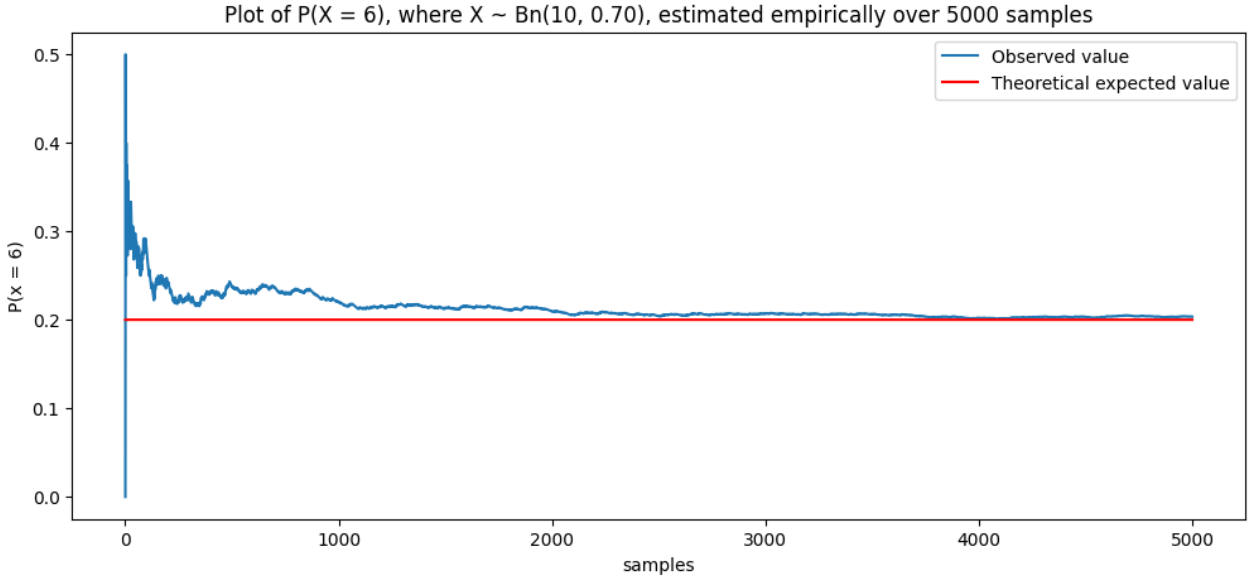






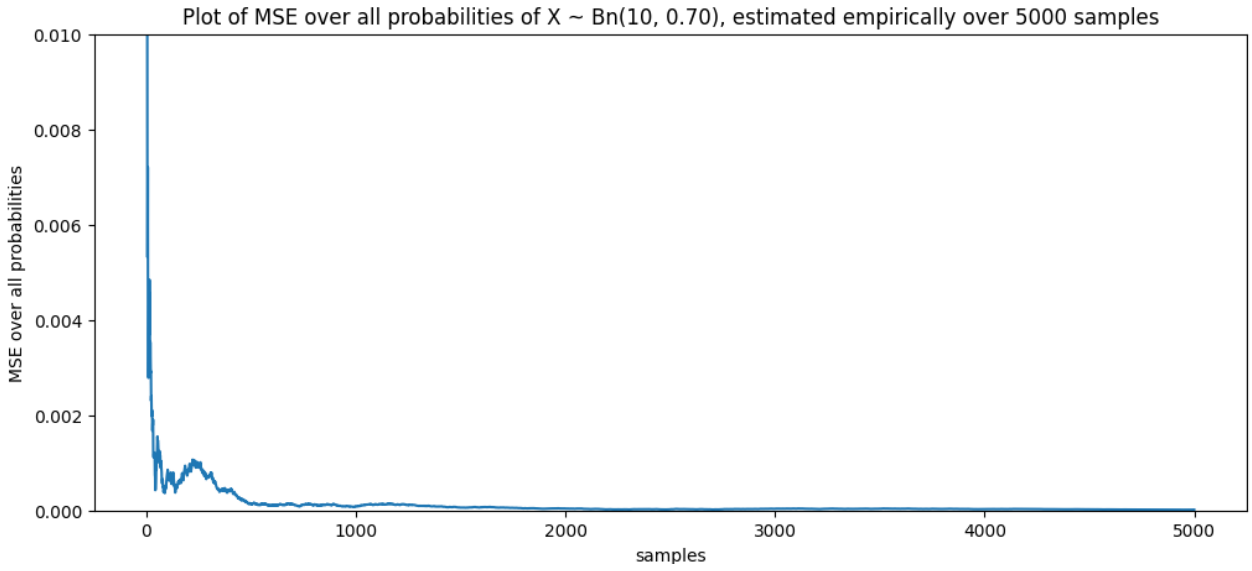
2. a. Empirical evidence for LLT

While LLT is usually used to show that the mean/expected value of a sampled distribution converges to its theoretical value, it can just as easily be extended to show this for any specific probability from the mass/density function of a distribution. For example, here we show how the value of $P(X = 6),\ where\ X \sim Bn(10, 0.7)$, through *5000* samples reaches its theoretical value of $\binom{10}{6} 0.7^6 (1 - 0.7)^4 \approx 0.2$.



We can also plot the mean-square error of the sampled distribution as per our generator over the theoretical distribution, defined as

$$MSE(s,n) = \frac{\sum_{i=0}^{s} \frac{\sum_{k=0}^{n} (P(X=k) - P(X_i=k))^2}{n+1}}{s}$$, where X_i is the random variable generated from i samples of our generator, with parameters n and p, and $X \sim Bn(n, p)$. We are basically doing the above procedure, but for all discrete probabilities simultaneously. We expect this MSE to converge to 0 by this theorem, meaning that the sampled distribution converges to the theoretical one - this is indeed what is shown empirically. Below are shown the results of plotting *MSE* with the same distribution $X \sim Bn(10,\ 0.7)$:



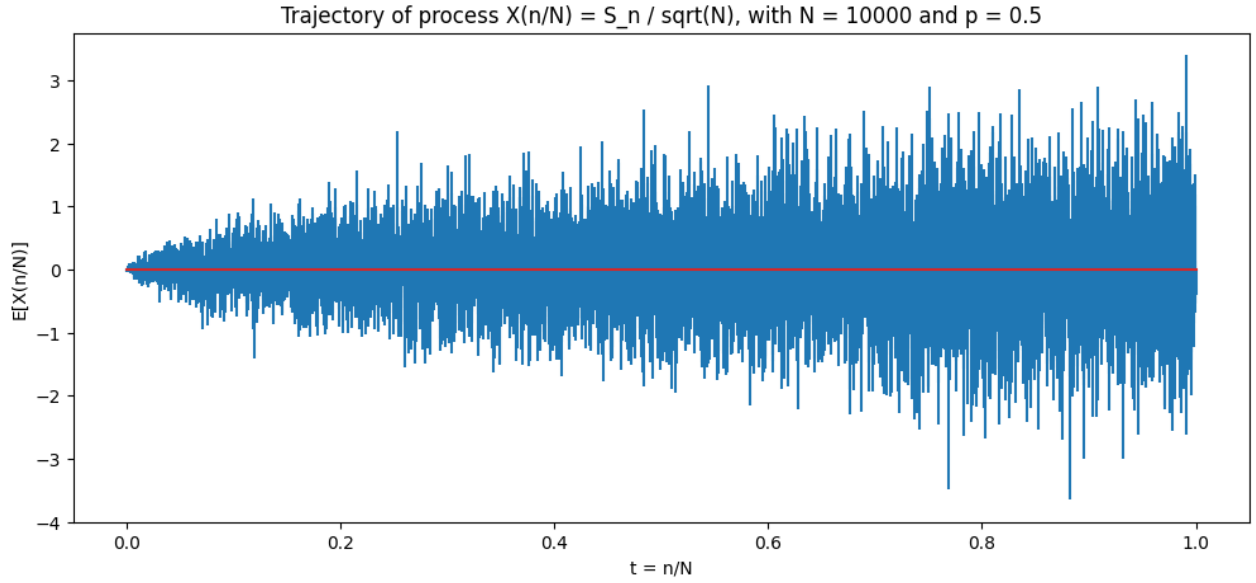
3. a. Trajectory process

We define the trajectory process in the same manner as the exercise puts it. Let $\forall i,\ X_i \sim Ber(1/2)$, scaled to {-1,1}

$$S(n) = \sum_{i=1}^{n} X_i$$

$$X(t) = X(n/N) = S(n)/\sqrt{N},\ where\ t \in [0,1],\ n \in [1,N].$$

Below we show the plot of the resulting function:

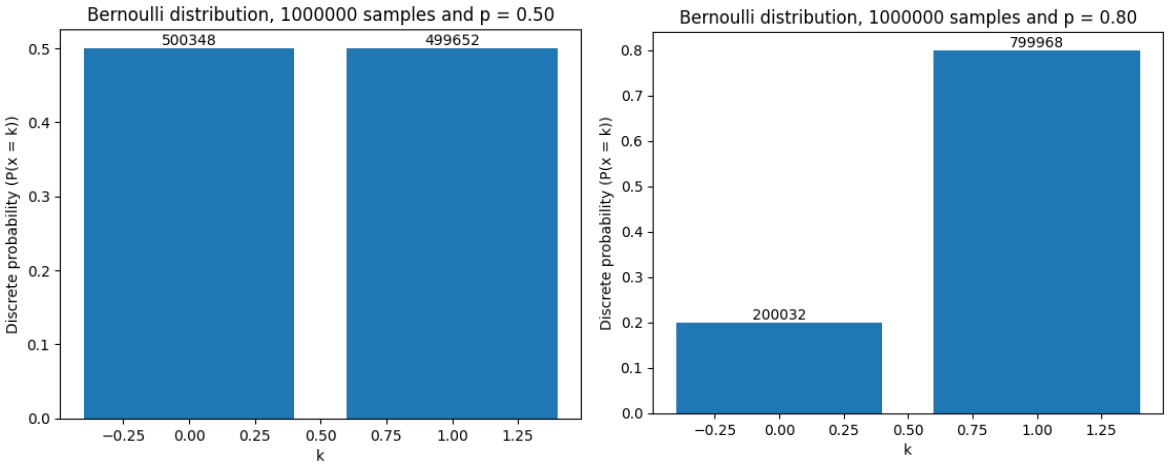


What is important to note here is how the trajectory oscillates around $X(t) = 0$, which is to be expected - after all the Bernoulli random variables we use to calculate the process have $p = 0.5$ to pick between -1 and 1, so $\forall i,\ E[X_i] = 0$.

Symmetry property (4 points)

1. a. Bernoulli generator

A Bernoulli distribution models the success of a Bernoulli trial, or in other words, an experiment with only two outcomes, a *positive* and a *negative* one, with probabilities of occurring *p* and *1 - p* respectively. The way we implement this generator is by taking a uniformly distributed variable between 0 and 1 - if the pseudo-randomly generated number is smaller than *p*, we label it as 1, and 0 otherwise. Additionally, we have extended the generator so that the high and low values are modifiable - this will help for the Cantor distribution later. Below we show how the generator sampled a million numbers, once with $p = 0.5$ and once more with $p = 0.8$.



2. a. Cantor distribution

To generate a cantor distribution, we have to start off with the definition of a Cantor function. The function is defined in the following way:

$$c(x) = 2\sum_{n=1}^{\infty}\frac{a_n}{3^n},\ when\ x \in C,\ else\ sup_{y \le x,\ y \in C} c(y$$

, where $a_n \in \{0,\ 1\}\ \forall n \in \mathbb{N}$, and *C* denotes the Cantor set, recursively defined as:

$$C_0 = [0,\ 1]$$

$$C_n = \frac{C_{n-1}}{3} \cup \left(\frac{2}{3} + \frac{C_{n-1}}{3}\right)$$
$$C := \lim_{n\to\infty} C_n$$

An intuitive way to conceptualize this is that the Cantor function equals twice the sum of "3-terms" if *x* is in the Cantor set, else it's simply the last value that was in the set, creating the staircase function that one can observe when plotting the function (shown later). The way the Cantor set is defined, where it recursively "chops" the range in thirds and discards the middle one, allows for some very neat properties and symmetries, which we will get back to in later subquestions. The way we create a generator for the singular Cantor distribution is precisely derived from the definition of the Cantor function, where we swap a_n with Bernoulli random variables (with *p=½)* and have an upper bound on the number of "3-terms", because we cannot calculate an infinite sum, after all. Therefore, the way we define the generator is as follows: $X_n = 2\sum_{i=1}^{n} \frac{Y_i}{3^i}$, where X_n is a Cantor random variable, approximated with *n* terms, and $\forall i,\ Y_i \sim Ber(0.5)$. A higher number of terms leads to a better estimation, but Python's floats can only go so far that we did not bother increasing it over 15-20-ish. Because the 4th subquestion of this exercise requires empirical evidence for the distribution (which should've been put here for logical textual flow and consistency…), we will move the plots showing the distributions there.

3. a Anti-symmetry over "halves" and symmetry over "thirds" - self-similarity

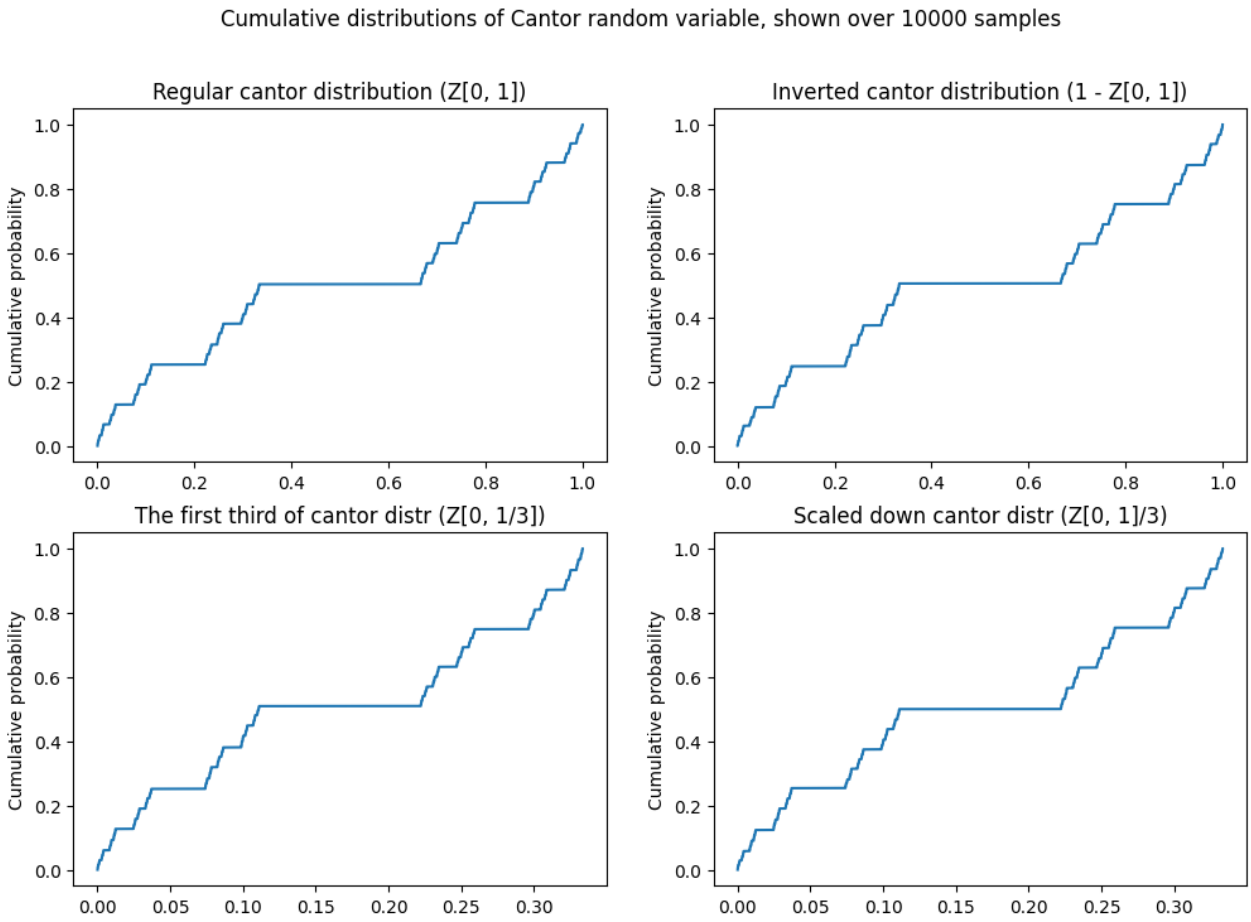
It should be evident from the histogram that the Cantor distribution is mirrored over the axis defined by *x = 0.5*. What should be even more evident from the way we define the Cantor set is that the distribution is also self-similar over its end-thirds. To showcase the validity of these observations, we will plot 4 separate Cantor distributions, and modify each one to show that the symmetries hold. Let's generalize the random variable in the following manner: Define a random variable $Z_n^{[a,b]}$ as $Z_n^{[a,b]} = X_n$, where X_n is the Cantor generator with *n*-terms over the Cantor set $C^{[a,b]}$, defined recursively as:

$$C_0^{[a,b]} = [a,\ b]$$

$$C_n^{[a,b]} = \frac{C_{n-1}^{[a,b]}}{3} \cup \left(\frac{2}{3} + \frac{C_{n-1}^{[a,b]}}{3}\right)$$

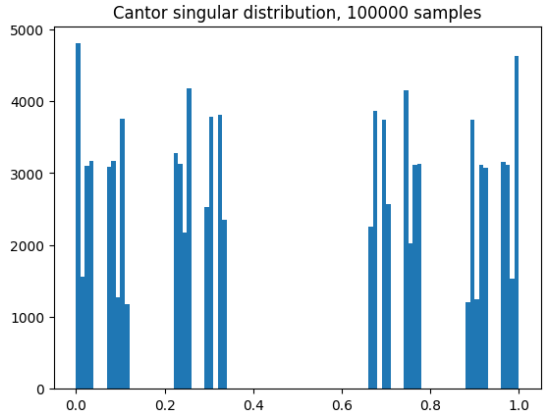
$$C^{[a,b]} := \lim_{n\to\infty} C_n^{[a,b]}$$

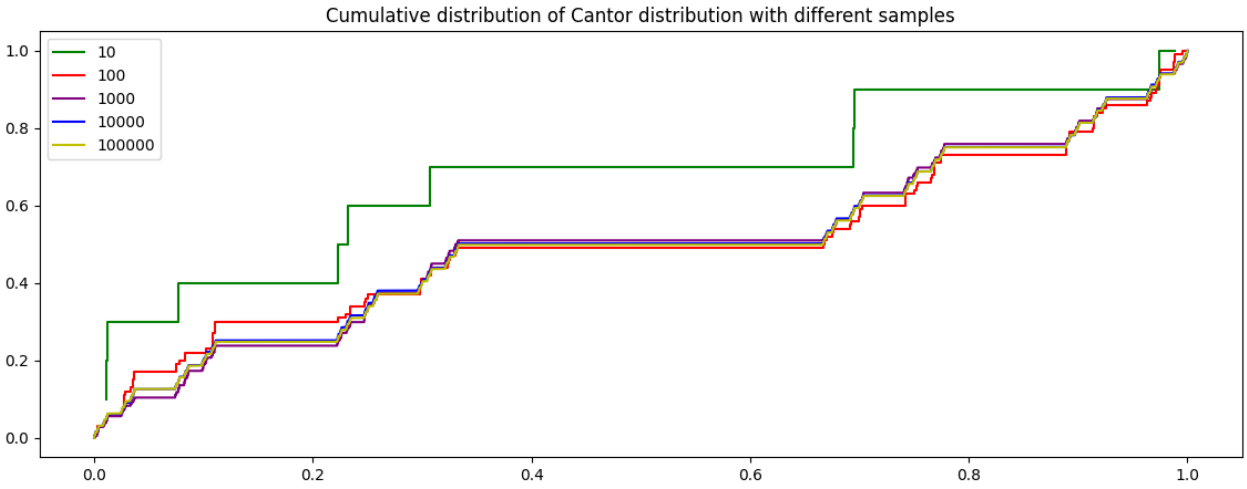
We basically extend the definition of a Cantor set to give it an arbitrary starting range, and the new Cantor random variable just uses said Cantor set. Now, to show the anti-symmetry over its halves, we have to show that $Z_n^{[0,1]} = 1 - Z_n^{[0,1]}$, and for the self-similarity over thirds it is sufficient to show that $Z_n^{[0,1/3]} = Z_n^{[0,1]} / 3$. This is when we plot the 4 variables next to each other, with *10000* samples:



4. a. An empiric evaluation of the Cantor generator

Below we show the singular distribution of the variable, approximated with 20 terms, along with the probably more interesting part - the cumulative distribution of the random variable, which is an approximation of the analytic Cantor function as defined above:





From one distribution to another (3 points)

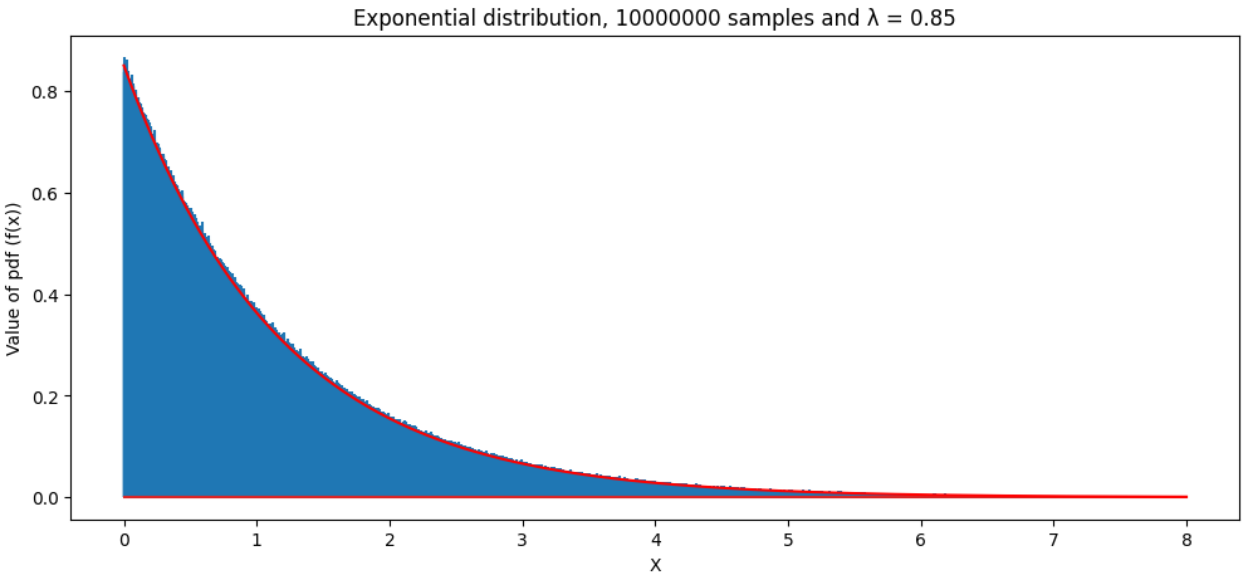
1. a. Exponential distribution

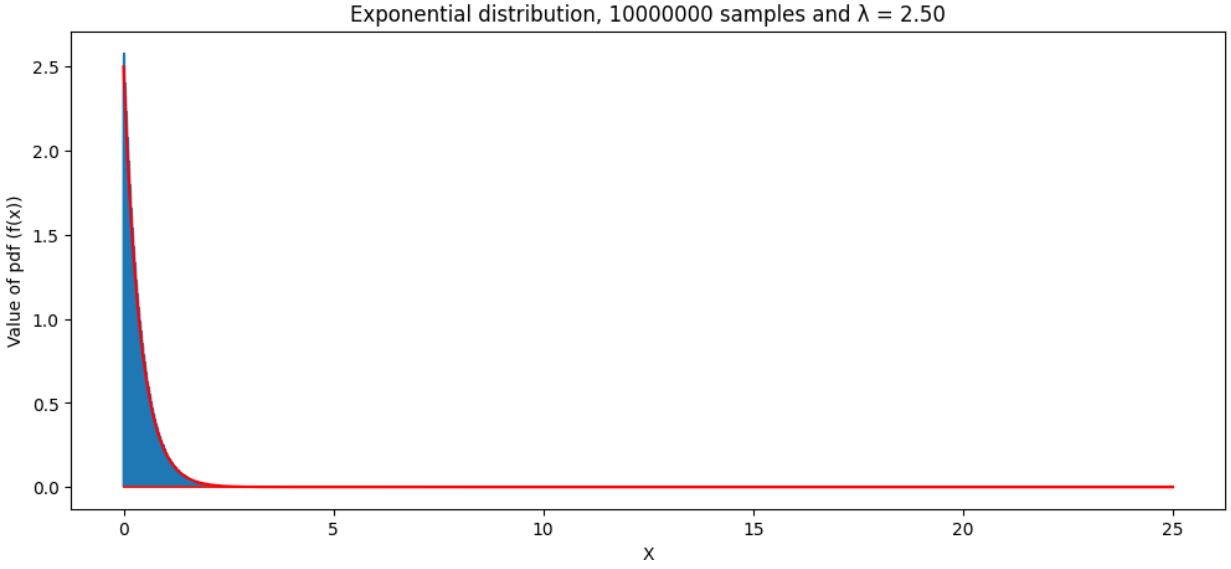
We define the exponential distribution as the distribution with a pdf of $f(x, \lambda) = \lambda e^{-x\lambda}$, $where\ x \in [0, \infty]$, and usually denoted as $X \sim Exp(\lambda)$. The exponential distribution plots the time between two events in a Poisson point process, which is a process by which events occur on average at the same rate and are independent of each other. The way we implement such a generator makes use of inverse transform sampling, which allows us to create any random variable so long as we know its inverse cumulative distribution, with the given relation:

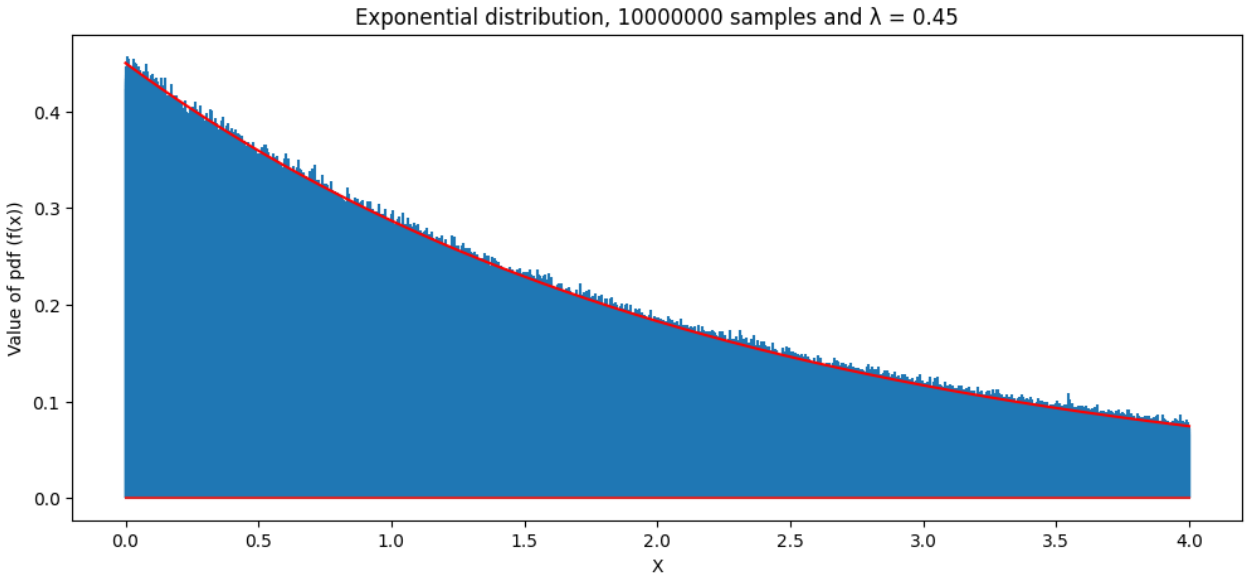
$X = F_X^{-1}(U)$, for any random variable X, where $U \sim Uniform(0, 1)$.

While finding the inverse is usually a difficult task, for the exponential distribution it is quite simple - its cumulative distribution can be shown to be

$F_{Exp}(x) = 1 - e^{-\lambda x}$, so its inverse would be $F_{Exp}^{-1}(x) = -\ln(1-x)/\lambda$. When we plug in a uniform random variable, the whole thing simplifies to: $F^{-1}(U) = -\ln(U)/\lambda$. This allows for a simple and performant implementation of an exponential random variable. Below we plot the final generator over 10 million samples, along with the analytical closed form of the pdf (shown as the red function), with different values of lambda:







1. b. Poisson distribution

Verifying the Poisson distribution from an exponential distribution is quite straightforward once the link is established between the two distributions and the aptly named Poisson process. While the exponential distribution models the time between events, the Poisson process models the number of events given a certain timeframe - its conceptual inverse. We generate it as follows:

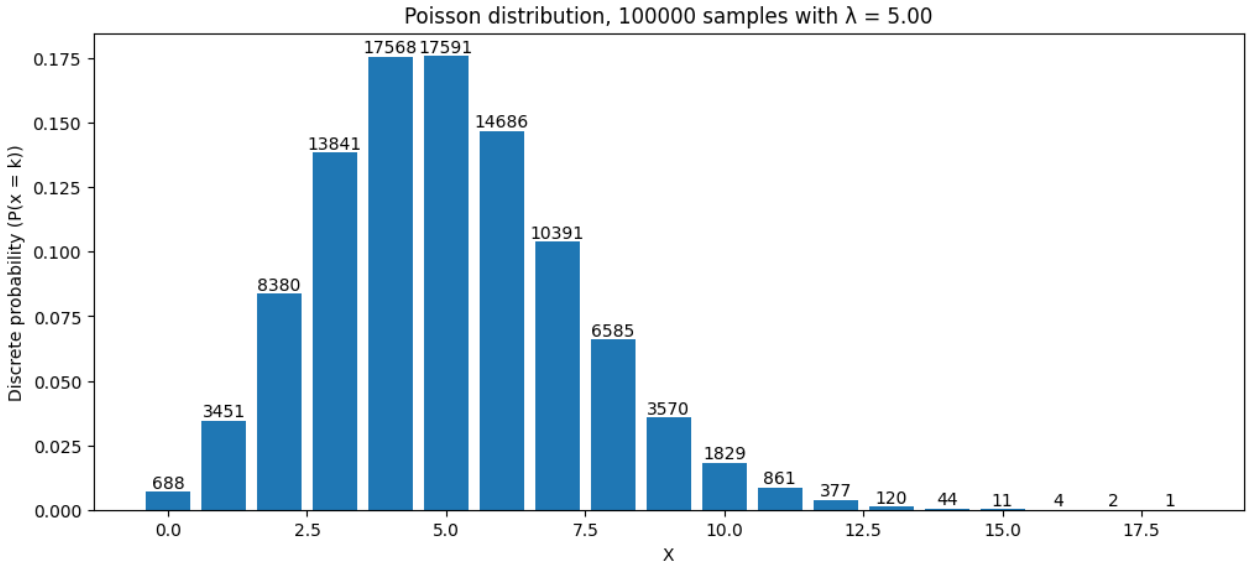
> *Initialize a counter and a total elapsed time variable to 0. Keep generating samples from an exponential generator with the same lambda as the Poisson generator, while keeping track of their count. Once the total elapsed time reaches [over] 1, terminate the process and return the counter - 1.*

The resulting distribution should be a Poisson distribution, and this is the case when compared to the closed-form analytical distribution, defined as:

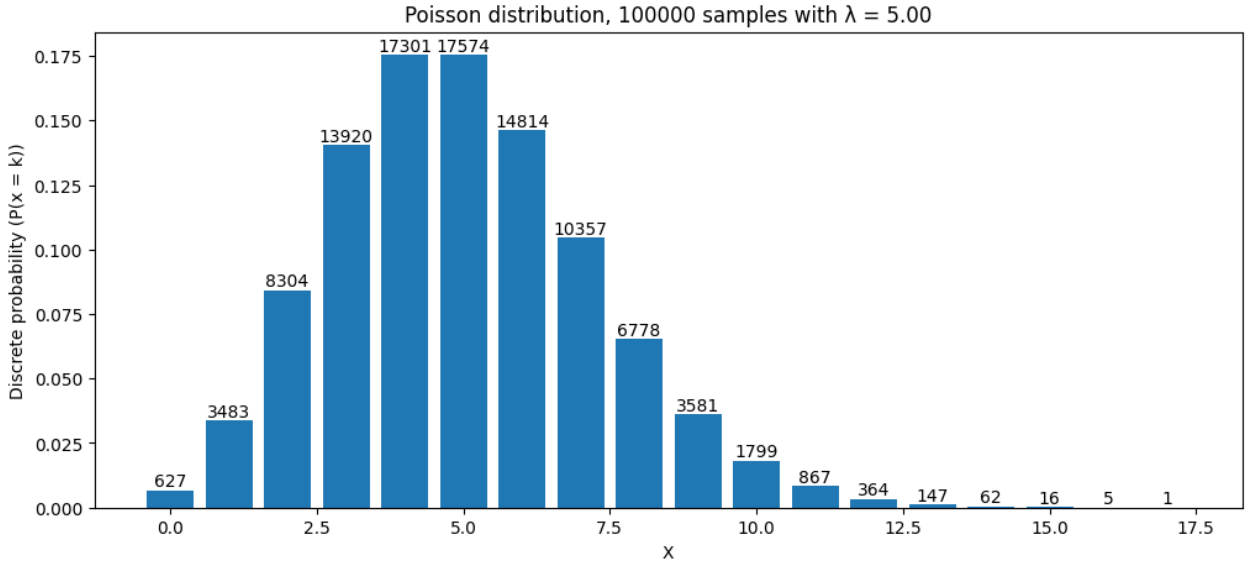
$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, \text{ where } X \sim Pois(\lambda)$$

Below we show two bar charts to compare the empirical with analytical solutions over *100,000* samples, with $\lambda = 5.0$:

Empiric/sampled:

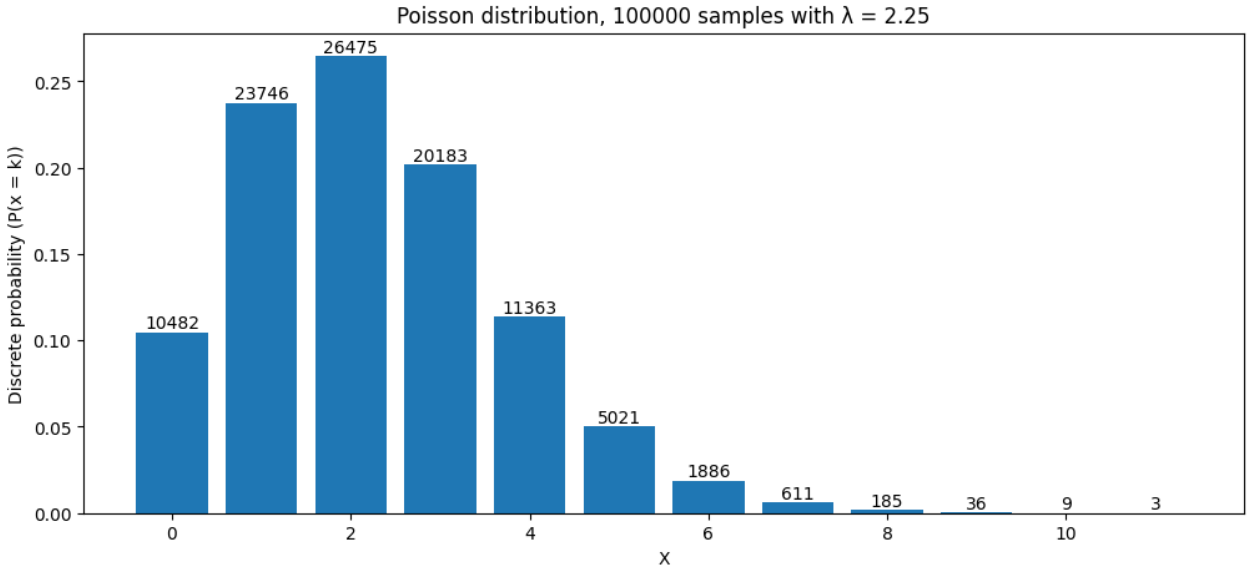


Analytic/closed form:

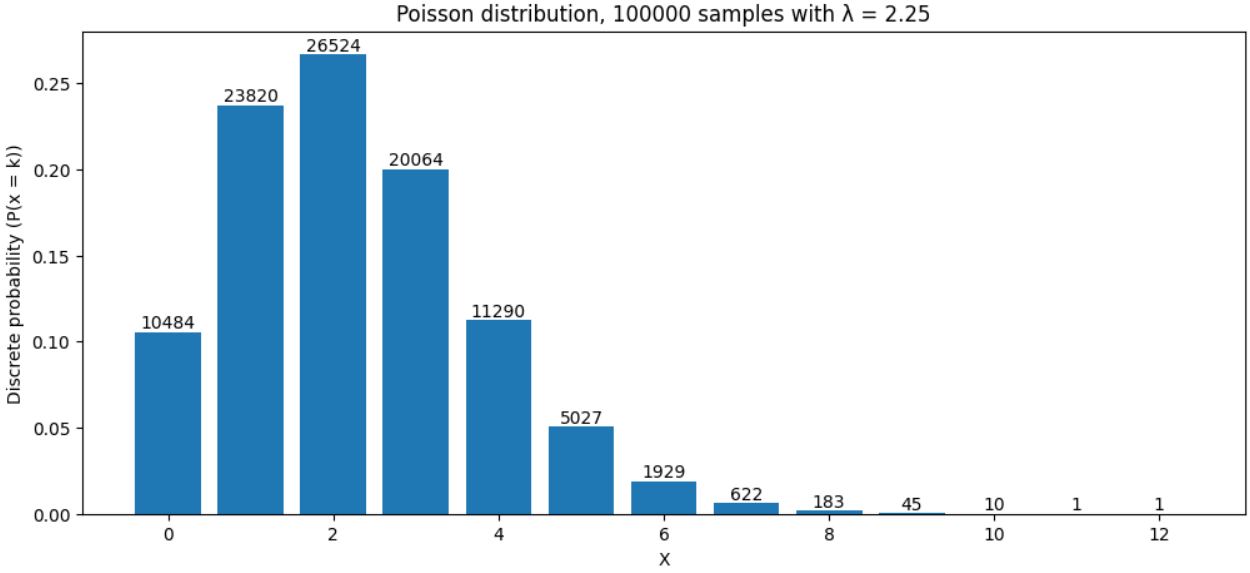


And here is the same comparison with $\lambda = 2.25$, the same number of samples:

Empiric/sampled:



Analytic/closed-form:



2. a. Normal distribution

Now onto the cornerstone of all distributions - the normal distribution. A normal random variable is denoted as $X \sim N(\mu, \sigma^2)$, with a pdf function:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Due to its ubiquity in probability theory, there are probably tens of ways of generating such a distribution - this time we will make use of polar coordinates

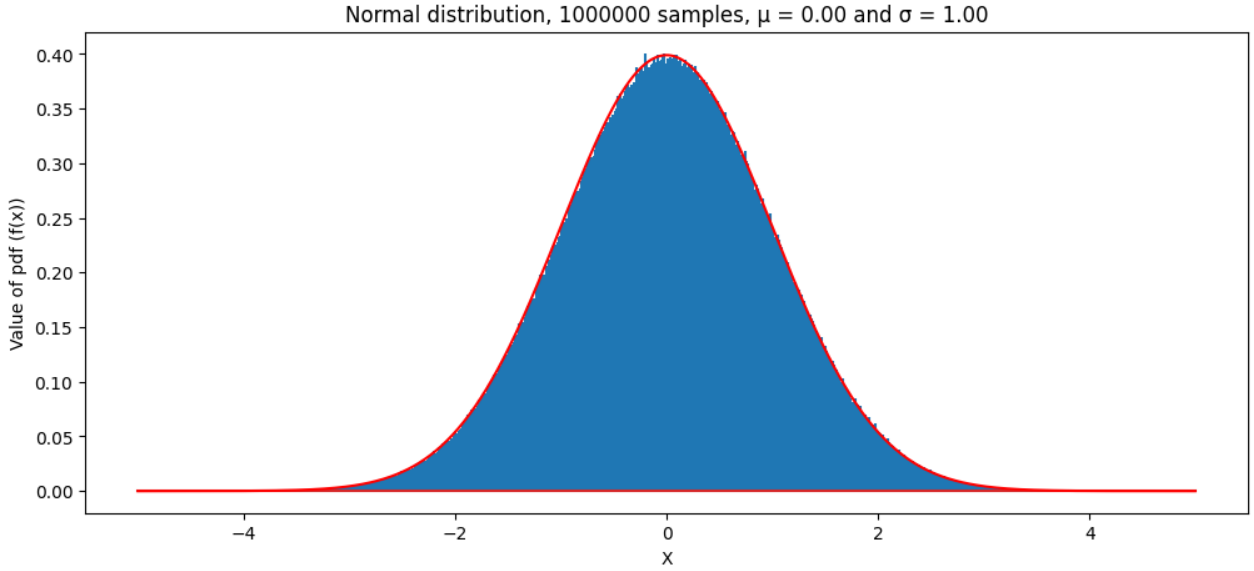
generated in a certain fashion. More specifically, if we generate $r^2 \sim Exp(1/2)$, and $\varphi \sim U(0, 2\pi)$, then the final polar coordinates generated from these pairs will have a standard normal distribution, or

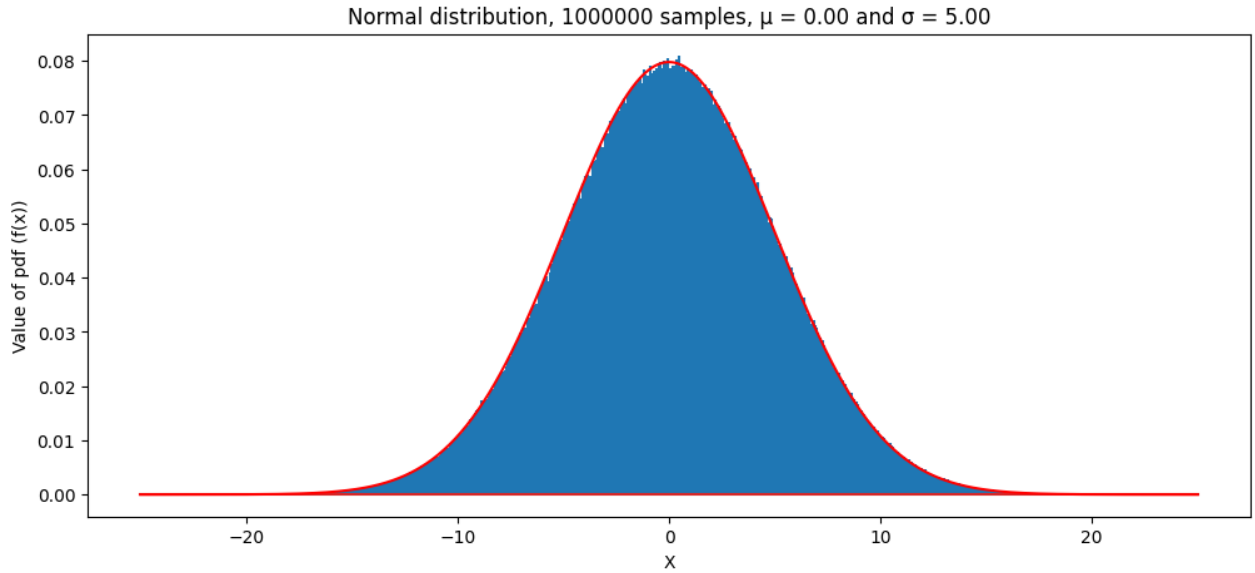
$$r\sin(\varphi) \sim N(0,\ 1) \cap\ r\cos(\varphi) \sim N(0,\ 1)$$

Due to its amazing properties, the standard normal distribution can be scaled to any other normal distribution by very simple operations, particularly:

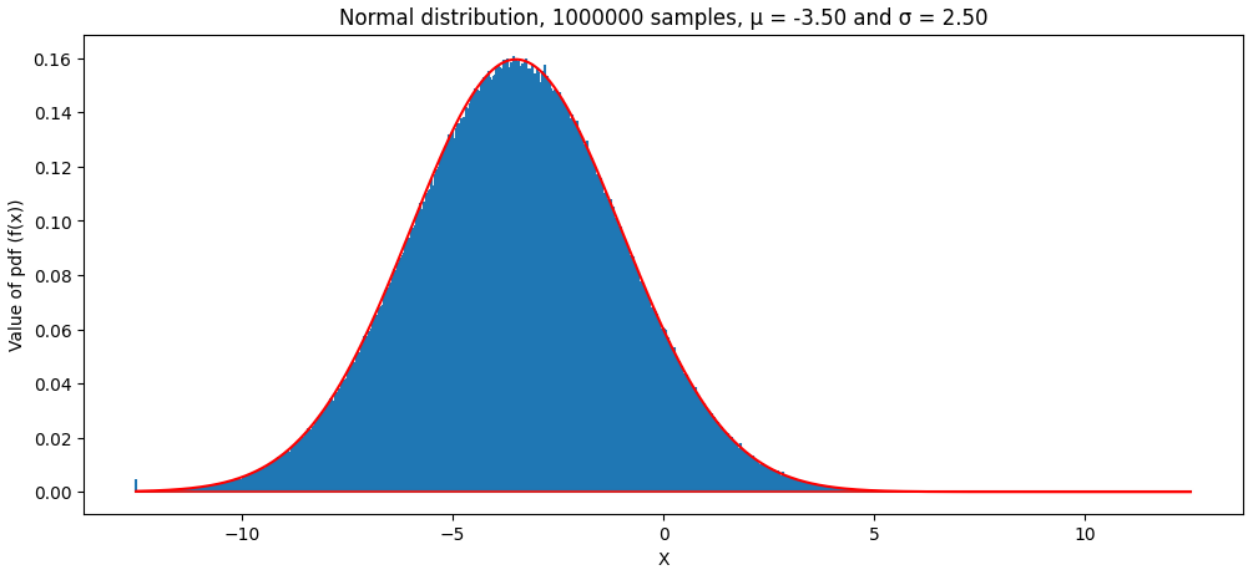
$$\sigma r\sin(\varphi) + \mu \sim N(\mu,\ \sigma^2) \cap\ \sigma r\cos(\varphi) + \mu \sim N(\mu,\ \sigma^2).$$

We now showcase this generator in the same manner as the previous continuous distribution, with several parameter pairs (titled) and over a million samples:





(Has same shape, but note how the axes' scales changed)

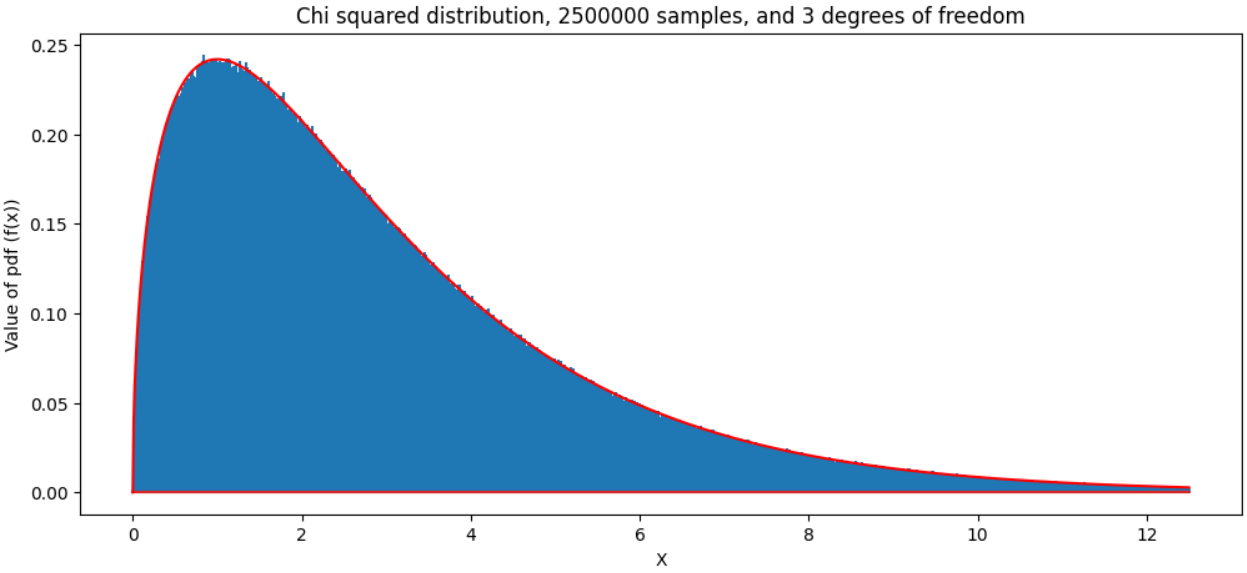


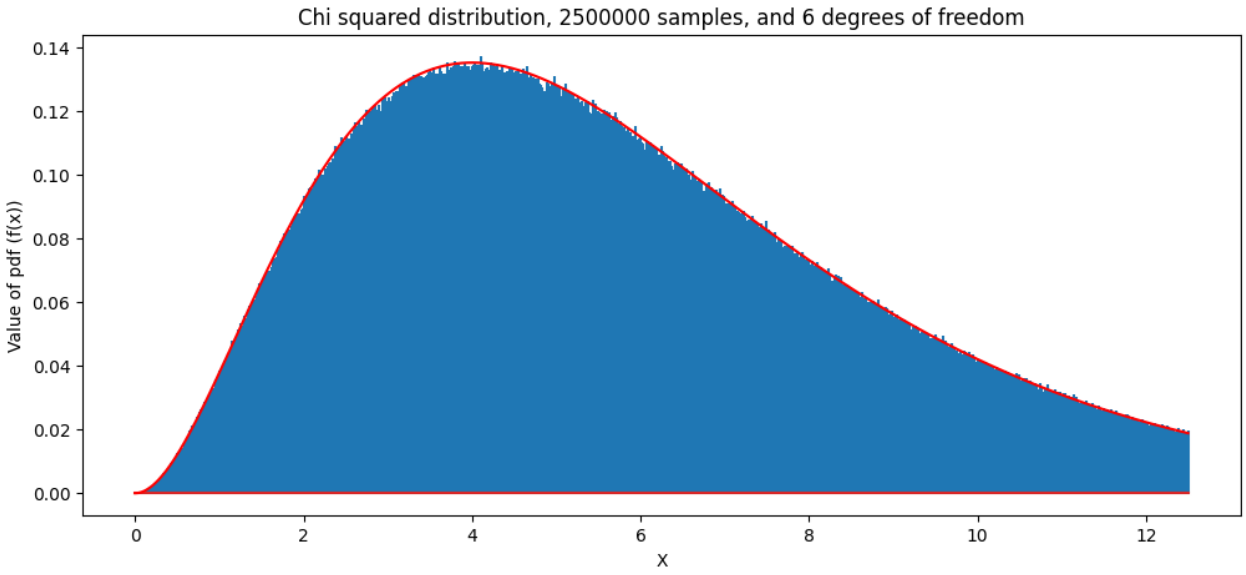
3. a. Chi-squared distribution

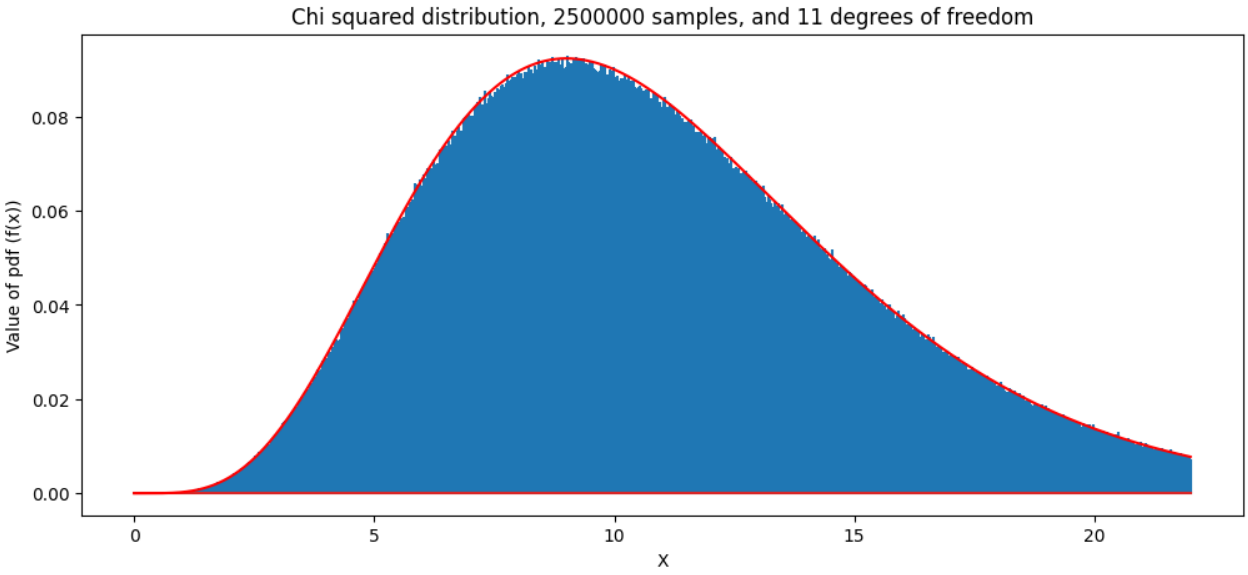
A chi-squared distribution is defined as the sum of many squared normal distributions, where the num is called *degrees of freedom*. More specifically:

$$X \sim \chi^2(k) \rightarrow X = \sum_{i=1}^{k} Z_i^2, \; where \; \forall i, \; Z_i \sim N(0,1)$$

Here is how the distribution looks like with several choices of *k*/degrees of freedom (titled), over 2.5 million samples:







LLN and CLT (2 points)

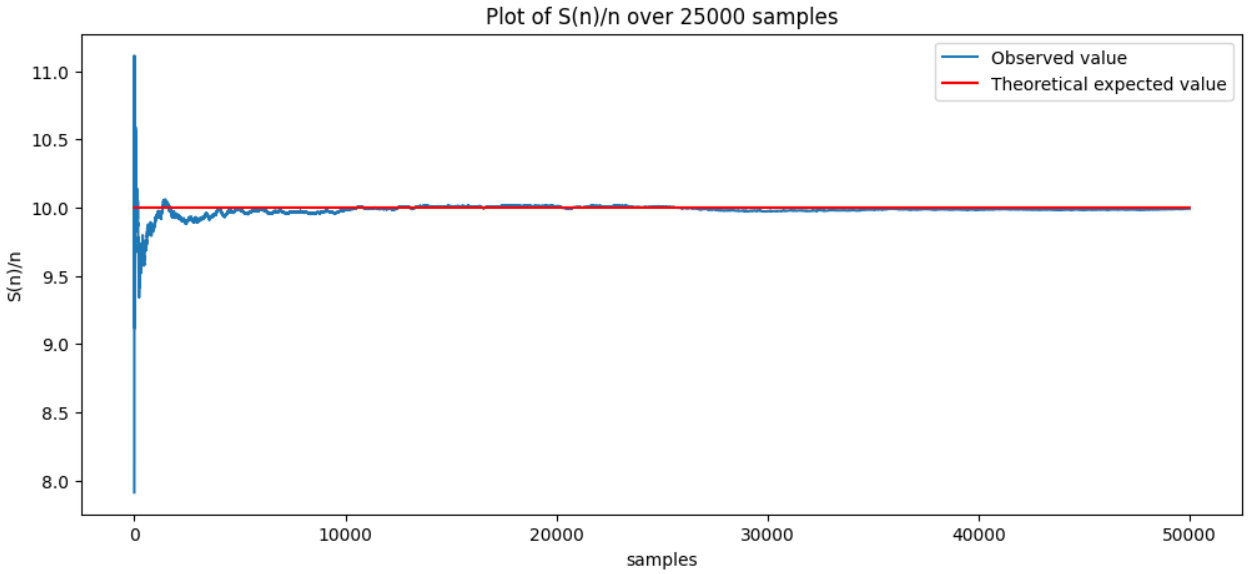
1. a. LLN

We will start off by stating the theorem we try to empirically show. The Law of Large Numbers states that

$$\lim_{n\to\infty} \sum_{i=1}^{n} \frac{X_i}{n} = \bar{X}_n.$$

For simplicity when plotting, we will define the running average $S(n) = \sum_{i=1}^{n} X_i$.

Furthermore, we will show this empirical proof with $X \sim N(a,\ b^2)$, as per the exercise's request. Therefore, we only have to show that $\lim_{n \to N} \frac{S(n)}{n} = a$, for some very large sample size *N*. We will now plot $S(n)\ /\ n$ over *25000* samples:



From this, we clearly see the convergence of the expected value of the random variable.

2. a. CLT

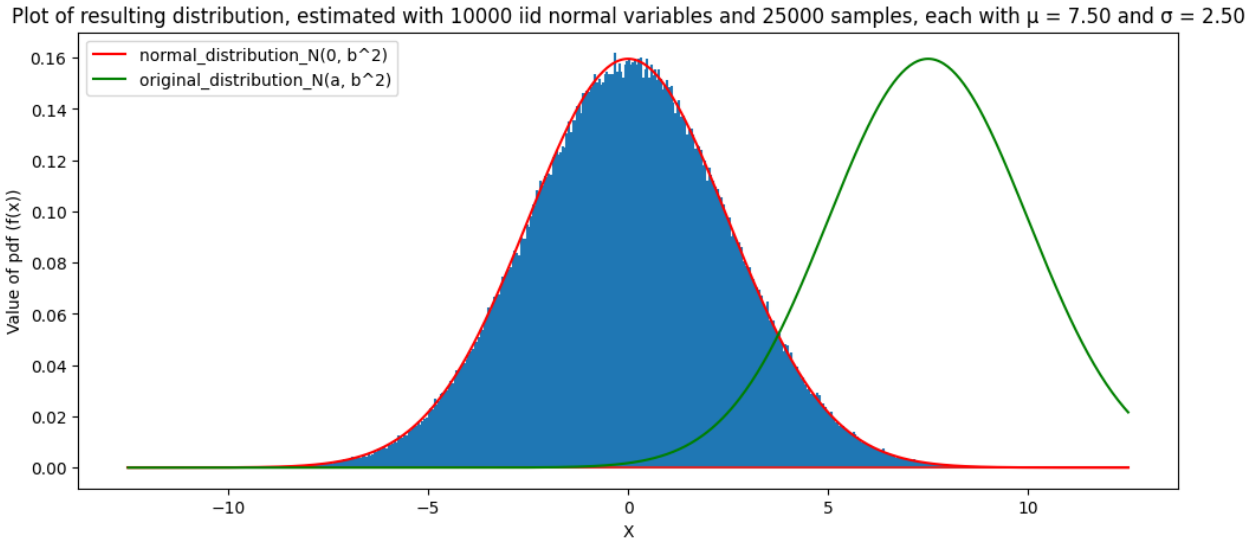
Another important theorem in probability theory is the Central Limit Theorem, which is one of the distinguishing features and describers of the importance of the normal distribution in probability theory. It states that the sum of many i.i.d random variables with the same mean and *finite* standard deviation is normally distributed, even if these random variables are not themselves normally distributed. Specifically (the classical way CLT is stated),

$$\lim_{n \to \infty} \sqrt{n}(\bar{X}_n - \mu) \sim N(0,\ \sigma^2).$$

Again, due to the nice standardization property of the normal distribution, this theorem can be slightly extended to:

$$\lim_{n \to \infty} \frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} \sim N(0,\ 1).$$

We show the result of the **former** random variable (because exercise says so) when estimated with $n = 10000$ and sampling the final random variable with *25000* samples:



We can see that the distribution of $\sqrt{n}(\bar{X}_n - \mu)$ is clearly converging to a normal distribution with std of b, empirically showing the validity of the theorem.

Integral (2 points)

0. Representation

We first spend some time actually analyzing the function to be integrated. The specific instance we are asked to solve is

$$\int_0^{\infty}\dots\int_0^{\infty} exp\{-\ (x_1 + \dots + x_{10} + \frac{1}{2^{20}x_1\dots x_{10}})\}x_1^{\frac{1}{11}-1}x_2^{\frac{2}{11}-1}\dots x_{10}^{\frac{10}{11}-1}dx_1\dots dx_{10}$$

An intuitive thing to do with multivariate integrals is to generalize them based on the number of dimensions. Let us define a function *I(n)* in which we want to parametrize the dimensions of the integral based on *n*. This can be achieved by defining *I* the following way:

$$I(n) = \underbrace{\int_0^{\infty}\cdots\int_0^{\infty}}_{n \text{ times}} e^{-\frac{1}{2^{2n}\prod_{j=1}^{n}x_j}} \prod_{i=1}^{n} e^{-x_i}x_i^{\frac{i}{n+1}-1}dx_1\cdots dx_n$$

Therefore, the problem turns into "solve *I(10)*". This also will allow for empirical checks with other methods when dimensionality is low. Another essential thing to note about this integral is how quickly the values converge to 0 as one evaluates

the integrand. For instance, evaluating the integrand of *I(1)*, a single dimension, with, for example $x_1 = 10$, we get

$e^{-\frac{1}{2^2 10}} e^{-10} 10^{-1/2} \approx 1.17 \times 10^{-6}$. Now one can imagine that this value will get exponentially smaller as we increase the dimensions, to a point where they are negligible in reaching a good numerical approximation for the integral, while being a heavy burden to compute explicitly. In the end, if our goal is a simple numerical approximation of *I(n)*, then there is no need to integrate from $[0, \infty]$, but to a much smaller domain since most of the important terms are concentrated in the small initial range within the first few positive integers.

1. a. Monte Carlo integration

The first way we will calculate said integral is through the Monte Carlo integration numerical method. What it tries to do is to sample points randomly around the function based on a certain distribution, and the method bases its validity on the fact that finding the average of all these samples, weighted over the probability of picking them, will be an approximation of the integral in question. Equivalently, the percentage of i.i.d. randomly distributed variables over the area closed by the axes and the function, over all samples, will be an approximation of the percentage of the said area over the sampling space area. Even more formally, if we denote $\langle F^N \rangle$ as a Monte Carlo approximator of an integral F of function f with N samples, the Monte Carlo integration method works based on this equation:

$$\langle F^N \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{pdf(X_i)}.$$

Here is a quick rundown of why this estimator works as a whole:

$$\lim_{N \to \infty} \langle F^N \rangle = E[\langle F^N \rangle] \text{ (by LLN)}$$

$$= E[\frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{pdf(X_i)}]$$

$$= \frac{1}{N} \sum_{i=1}^{N} E[\frac{f(X_i)}{pdf(X_i)}]$$

$$= \frac{1}{N} \sum_{i=1}^{N} \int_{\Omega} \frac{f(x)}{pdf(x)} pdf(x) dx$$

(by general definition of E[X] for arbitrary random variable)

$$= \frac{1}{N}\sum_{i=1}^{N}\int_{D} f(x)dx,$$ where D is full domain of $f(x)$

$$= \int_{D} f(x)dx$$

$$= F$$

It is important to note that while this might looks very different from the formula given on the lecture slides, it is not only the same formula, it is actually a generalized version of it - the one on the slides is just a special case of reordering and using uniform sampling, and has the following form:

$$\frac{\sum \text{[illegible]}}{N} \approx p = \frac{\int_0^{\pi} f(x)dx}{\pi/2}.$$

Knowing that the sampling on the example is uniform (so pdf $f(x) = 1/(b-a)$), and replacing terms with the ones used wherever possible, we get

$$\frac{\sum_{i=1}^{N} g(Y_i, f(X_i))}{N} = pdf(X)pdf(Y)\langle F^N \rangle, \text{ where}$$

$$g(x,y) = 1 \; if \; (x < y) \; else \; 0$$

$$\rightarrow \langle F^N \rangle = \frac{\sum_{i=1}^{N} g(Y_i, f(X_i))}{N pdf(X) pdf(Y)}$$

$$\rightarrow \langle F^N \rangle = \frac{1}{N}\sum_{i=1}^{N} \frac{\frac{g(Y_i, f(X_i))}{pdf(Y_i)}}{pdf(X_i)}$$

$$\rightarrow \langle F^N \rangle = \frac{1}{N}\sum_{i=1}^{N} \frac{f(X_i)}{pdf(X_i)}, \; \forall i \; Y_i \sim U$$

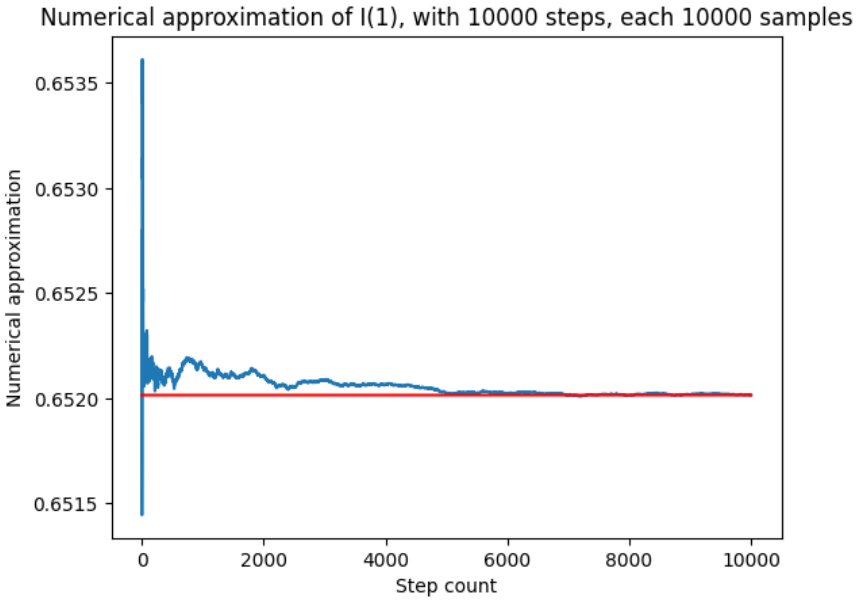
And we get to the original Monte Carlo theorem as defined above. What's very powerful about our derivation of the equation, however, is that X_i can be any arbitrary distribution, and it's all a matter of how quickly the approximation converges to the theoretical value. For our problem in particular, as already mentioned, most values are concentrated at the beginning of the positive domain axis, therefore something like exponential distribution is better suited for sampling

and will converge faster than uniform (even though the function does not decrease monotonically like an exponential distribution, it still has the same behavior overall). The exponential distribution also does not require predefining the possible range of values, like with uniform, so we will not run into the problem of plugging in infinity in the calculations (since the integral here goes to infinity). Lastly, the range of the integrand does not include negative values (it's the multiplication of exponentials, all of which are positive), also shown by the fact that the integral is evaluated from 0, therefore the entire function is "sample-able" by an exponential distribution. In the end, what we do is that we sample as many exponential random variables as dimensions of the integral, and we assume i.i.d. between them so that the probability of sampling a certain point in space is the product of all these variables.

Before we solve the actual problem, we will check our methodology with fewer dimensions, where classical analytical methodologies can also be used to reach a closed-form solution. For instance, for 1 dimension it evaluates nicely to

$$I(1) = \frac{\sqrt{\pi}}{e} \approx 0.65204933\ldots$$

And here is the Monte Carlo approximation with *1000* steps, each with *1000* samples:

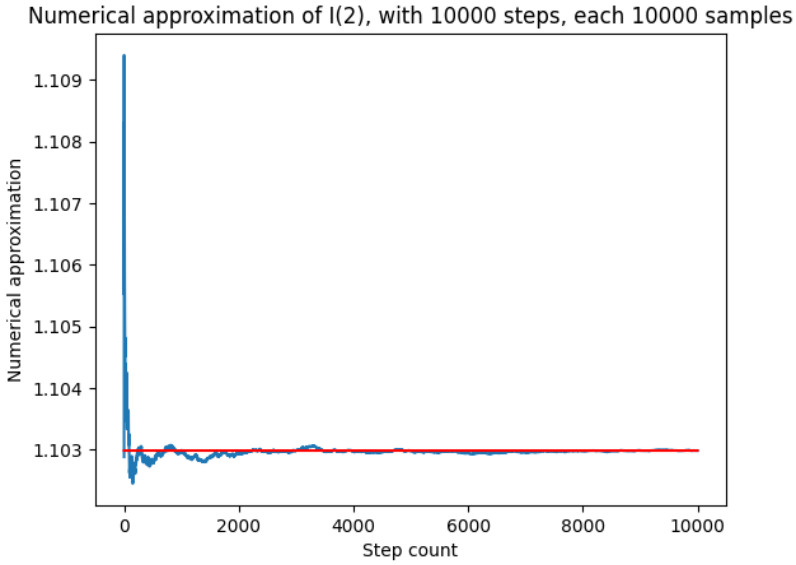


It converges to a final value of 0.65204567**,** accurate to 5 decimal places.

For 2 dimensions, we have

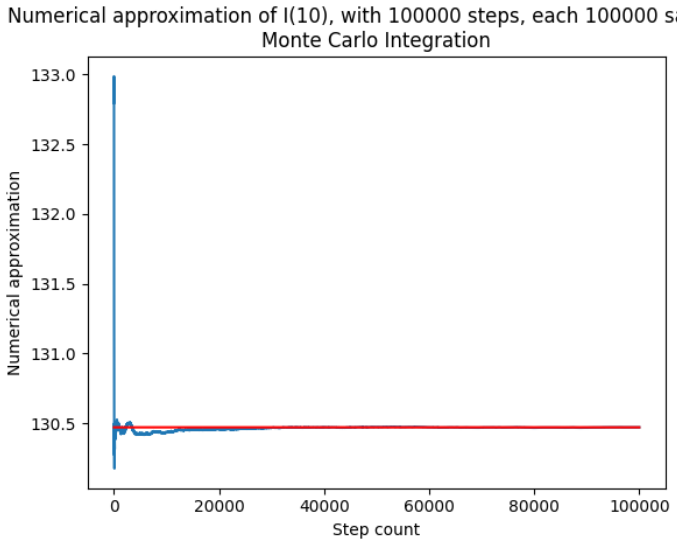
$$I(2) = \frac{2e^{-\frac{3}{2^{4/3}}}\pi}{\sqrt{3}} \approx 1.10298499\ldots$$

(closed-form found using WolframAlpha). One can notice how cumbersome the closed-forms get on 2 dimensions already, let alone 10. The Monte Carlo approximation of the same integral is given below, same sampling used:



the final value being 1.10298227, again accurate on 5 decimal places.

We can safely say that the Monte Carlo method as we have implemented it is definitely usable, and it's all a matter of how many samples it will take to reach a satisfactory result. Now onto the actual calculation, with 10 dimensions. Here we use more samples and steps to ensure maximal accuracy.



The final result we get is **130.4721701285** after 10 billion exponentially distributed samples in total.
Note: The red plot on each of the illustrations is simply the last value observed, not the theoretical value of a certain value of I, considering that for 10 dimensions we do not even have access to a closed-form solution.

2. a. Quadrature method

While this exercise is quite unclear from the mention of the method *solely* (the term "quadrature method" can refer to any numerical approximation methodology), the specification right afterward helps with explaining the problem - it mentions the reduction to a Riemann integral. Therefore, the equation we want to make use of is based on this formula (the one provided below is for a single integral, easily extended for more dimensions):

$$R(f) = \lim_{n\to\infty} \sum_{i=0}^{n-1} \sup_{t\in[x_i, x_{i+1}]} f(t)\Delta x$$

Where *R(f)* is an *upper bound* approximator of the Riemann integral - basically we split the domain into *n* splices, build the proper d-dimensional volume with height equal to the maximal value of the integrand in the range of the splice in question, and combine all the volumes (or areas in a 2d setting). The problem with using a

Riemann integral is that for the simple/useful case of the calculation, the splices that approximate the integral need to be evenly spaced (by Δx), along all the dimensions of the integral. Therefore, even if we want to split each axis by even the minuscule amount of 5 splices, this will result in 5^{10} overall elements to be summed up. We will need many more samples than that if we want to have a good approximation of the integral. This is why Monte Carlo Integration is simply the better option for this 10-dimensional integral if huge amounts of memory are not present to the user. The maximal number of samples we managed to use for the calculation of the 10-integral is *np.linspace(0.0001, 5, 7)* overall 10 dimension, or $7^{10} \approx 282mil$ *samples and got* **131.7255274622.**

Note: Since this procedure is not probabilistic like the others on the report, there is not really much to illustrate here (this method is incomplete without going through all bins in the range, so you cannot divide the work in steps and look at area convergence either.

Unit simplex (2 points)

1. a. Uniform simplex / Flat Dirichlet Distribution

We are now tasked with sampling from a random variable *X*, defined as

$$X = \{(x_1, x_2, x_3 | \forall i,\ 0 \leq x_i \leq 1 \cap x_1 + x_2 + x_3 = 1\}.$$

This distribution precisely describes what can be plotted on a *standard simplex,* a (k-1)-dimensional polytope, where its k vertices have the special property of adding up to 1 - it is no coincidence that such a simplex is also called a *probabilistic simplex*. A distribution defined by the possible points of such a simplex is equivalent to what can be modeled by a Dirichlet distribution, denoted as $X \sim Dir(\alpha)$ and its pdf defined as:

$$f(x, \alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{k} x^{\alpha_i - 1}$$

, where *B(x)* is the beta function and dim(*x*) = dim(α).

When all elements of α equal to 1, one can see that the product on the right collapses to a 1, which means that the *pdf* no longer depends on what *x'*s we plug in, or in other words, it is uniform over the simplex. We call this a special case of a Dirichlet distribution *flat*.

To be able to easily implement a generator for this special case of the distribution, we will make use of one more property of the Dirichlet distribution, namely its relation to the Gamma distribution. It can be shown that *k* sum-normalized Gamma random variables together are equivalent to a Dirichlet distribution with *k* dimensions, or more specifically,

$$\forall i,\ Y_i \sim Gamma(\alpha_i,\ \beta), \text{ and } S = \sum_{i=1}^{k} Y_i,$$

$$\left(\frac{Y_1}{S},\ldots,\frac{Y_k}{S}\right) \sim Dir(\alpha)$$

This is useful because the Gamma distribution can be further simplified when $\alpha_i = 1$ as its the case with a flat Dirichlet distribution. First, we write the definition of the pdf of a Gamma distribution: for $X \sim Gamma(\alpha, \beta)$

$$f(x) = \frac{x^{\alpha-1}e^{-\beta x}\beta^{\alpha}}{\Gamma(\alpha)}$$

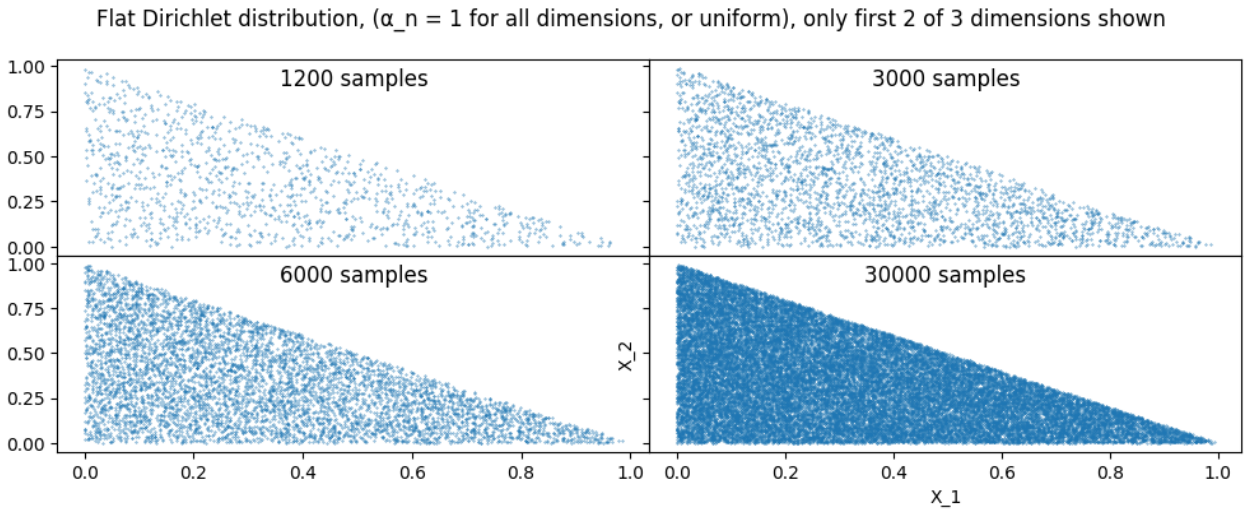
Plugging in $\alpha = 1$, we get

$$f(x) = \frac{x^{1-1}e^{-\beta x}\beta^{1}}{\Gamma(1)} = \beta e^{-\beta x} \qquad (\Gamma(1) = 0! = 1)$$

And would you look at that, we end up with the pdf of an exponential distribution! In the end, we have that for a flat Dirichlet distribution,

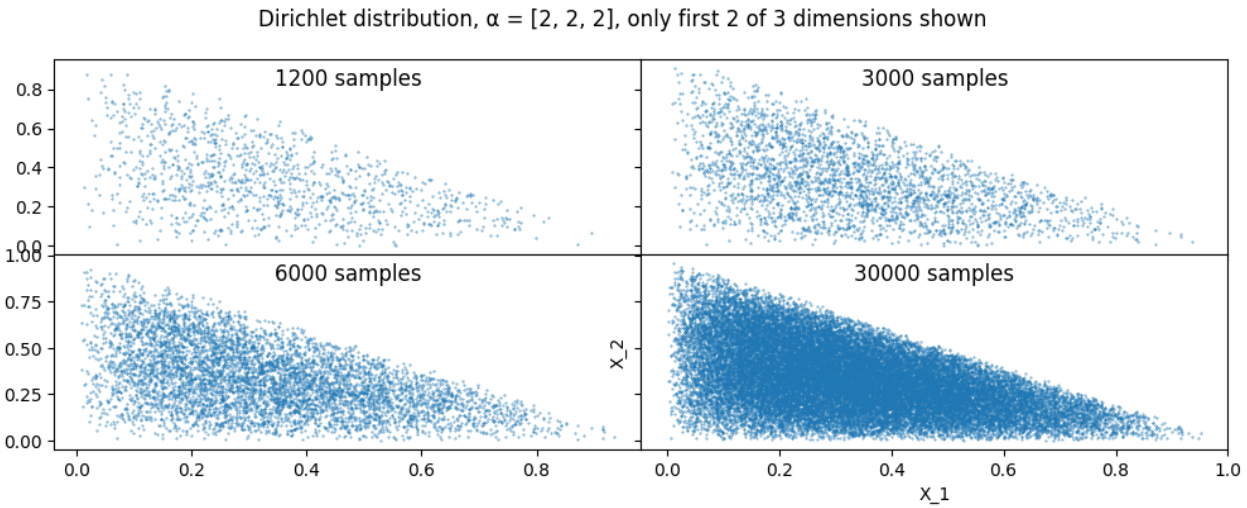
$$Dir(\alpha) \sim \left(\frac{Exp(\beta)}{S},\ldots,\frac{Exp(\beta)}{S}\right),\ \forall \beta \in \mathbb{R}_*^{+}.$$

Therefore, for a *k*-dimensional flat Dirichlet distribution, we can just sample *k* exponential random variables with any β of our choice (we will pick 1 for simplicity) and normalize them so that they all sum up to 1 (by dividing by their sum). The plot below visualizes the first 2 of three dimensions of a Dirichlet distribution over 3 variables, as per the task of the exercise, over several samples:



2. a. “Centered” simplex / Centered/Normalized Dirichlet Distribution

If we want our samples to be “centered” in the simplex, we have to adjust the alphas to be bigger than 1, but still all equal to each other. With this change, the pdf of the distribution now also depends on the *x*s but is weighted evenly on all of them, so that the center is, well, centered. A trivial way of doing it is to use the generalized Dirichlet distribution as described above and set α it in this manner. This is the result of doing this with $\alpha = [2,\ 2,\ 2]$



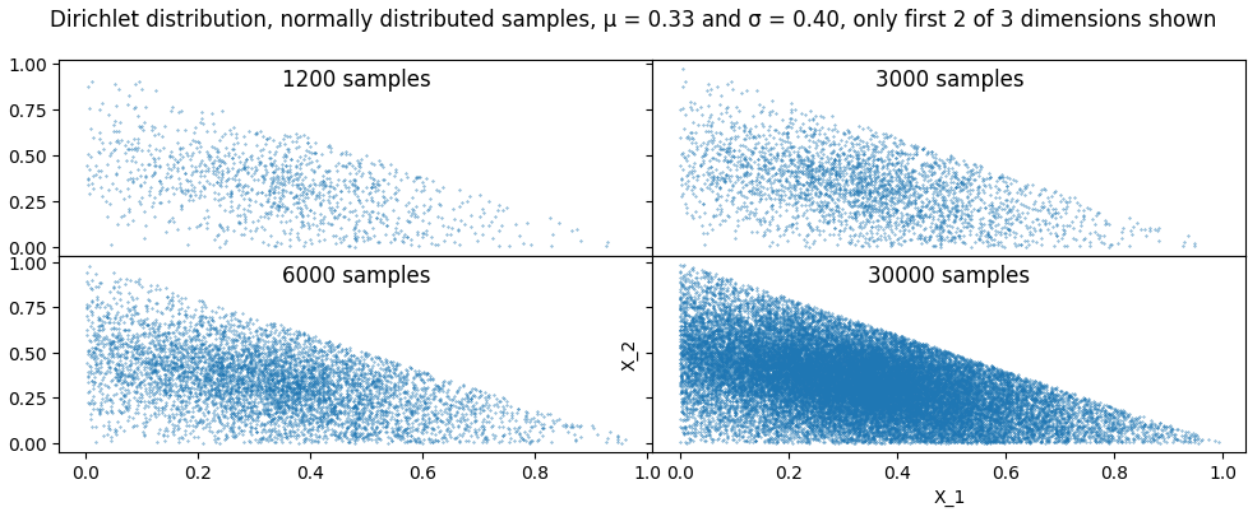
While this does the job, it is not as centered as some might want it…

Another solution to this is to pick a different sampling method for the points that will force the points to be centered - one candidate is to use the absolute value of a normal distribution that is centered around the mean value of the Dirichlet

variables (which is $1/k$ for a k-dimensional Dirichlet distribution), with any standard deviation that is deemed sufficient. The random variables will then again have to be sum-normalized to ensure the combination follows the probabilistic simplex property of "total sum to 1".

It is worth noting that on a purely mathematical/rigor level, using normal distributions for the variables will **not** result in a Dirichlet distribution specifically, but it will be a valid distribution overall, and with proper standard deviations it can cover the entire available simplex area while being "centered" with most samples around the variable mean of the distribution - the ambiguity of this subquestion does not eliminate this as an option nor does it enforce a technical Dirichlet distribution.

Below we show how this sampling looks with $\mu = 0.33,\ \sigma = 0.4$:



.