**Computational Statistics: Assignment 1**

**By: Blesson Issac(202485606) and Michael Bimal(202389499)**

**Github repository:** [**https://github.com/michael1999997/Computational-Statistics-Assignment-1**](https://github.com/michael1999997/Computational-Statistics-Assignment-1)

1. **A**) Here we only have access to uniform random samples, our goal is to generate binomial random samples (Binomial (10, 1/3)) using from this uniform samples (U (0,1)) using inversion method.

**Inversion Method**

The inversion method is a technique used to generate random variates from any distribution, given a source of uniform random numbers. The core idea of the inversion method is to use the inverse of the cumulative distribution function (CDF) of the target distribution. If F is the CDF of the target distribution and U is uniformly distributed over [0,1], then the variable X=F −1(U) has the distribution with the CDF F**.**

**Binomial Distribution and its Pseudo-Inverse**

For a binomial distribution with parameters n and p, the CDF is given by : F(k;n,p)=∑ i=0 k​( i n​)p i (1−p) n−i where ( i n​) is the binomial coefficient. The pseudo-inverse of the CDF is found by determining the smallest k such that F(k;n,p)≥U.

**Methodology**

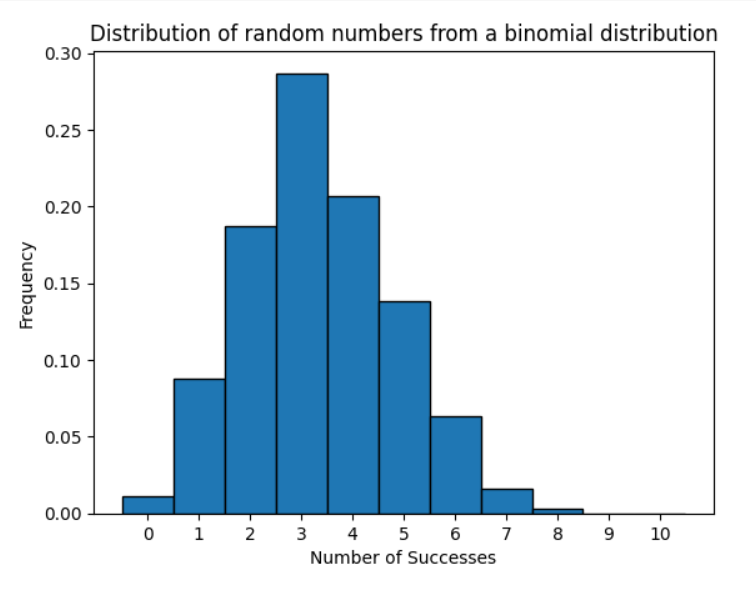
Here we took from binomial distribution and computed it CDF then it would be a random variable in U(0,1), Which means CDF maps any random variable to a uniform variable in the range 0 to 1.Here we need to generate random variables from binomial distribution by using a uniform distribution. So, we can do this just by reversing the above-mentioned process. That is by taking the inverse CDF of random variable from U(0,1).

Let X be a random variable from Binomial(10, 1/3) and CDF Fx(X) gives a random number in the range 0 to 1 since CDF takes value from 0 to 1, that is, it’s from U(0,1).

That is, Fx(X) = u, where u~U(0,1) Here we need X from u, we will get that by taking inverse

That is, Fx^-1(u) = X

The plot of this is shown below



The most frequent outcome is around 3 successes, which aligns with the expected mean of a Binomial distribution calculated as 𝑛 × 𝑝 = 10 × 1/ 3 ≈ 3.33 . This peak suggests that the most common scenario is achieving roughly one-third successes per 10 trials, reflecting the probability of success per trial.

**PART -B**

Here we are making use of both inversion and transformation method to generate samples form a binomial distribution. We know that the binomial distribution is a Bernoulli distribution repeated multiple times. That is, a Binomial random variable is the sum of n independent Bernoulli trials. If we perform n independent Bernoulli trials, each with success probability p, and sum the outcomes, the result will follow a Binomial distribution.

By using this idea, we can do this question. First, we can generate random samples from Bernoulli distribution from U (0,1) using Inversion Method. And then we can transform our Bernoulli samples to binomial samples using transformation method based on the idea that sum of Bernoulli gives binomial.

Here we know that Bernoulli distribution has only two out comes, success (1) or failure(0).

P(X=1) = p = 1/3

P(x=0) = 1-p = 2/3

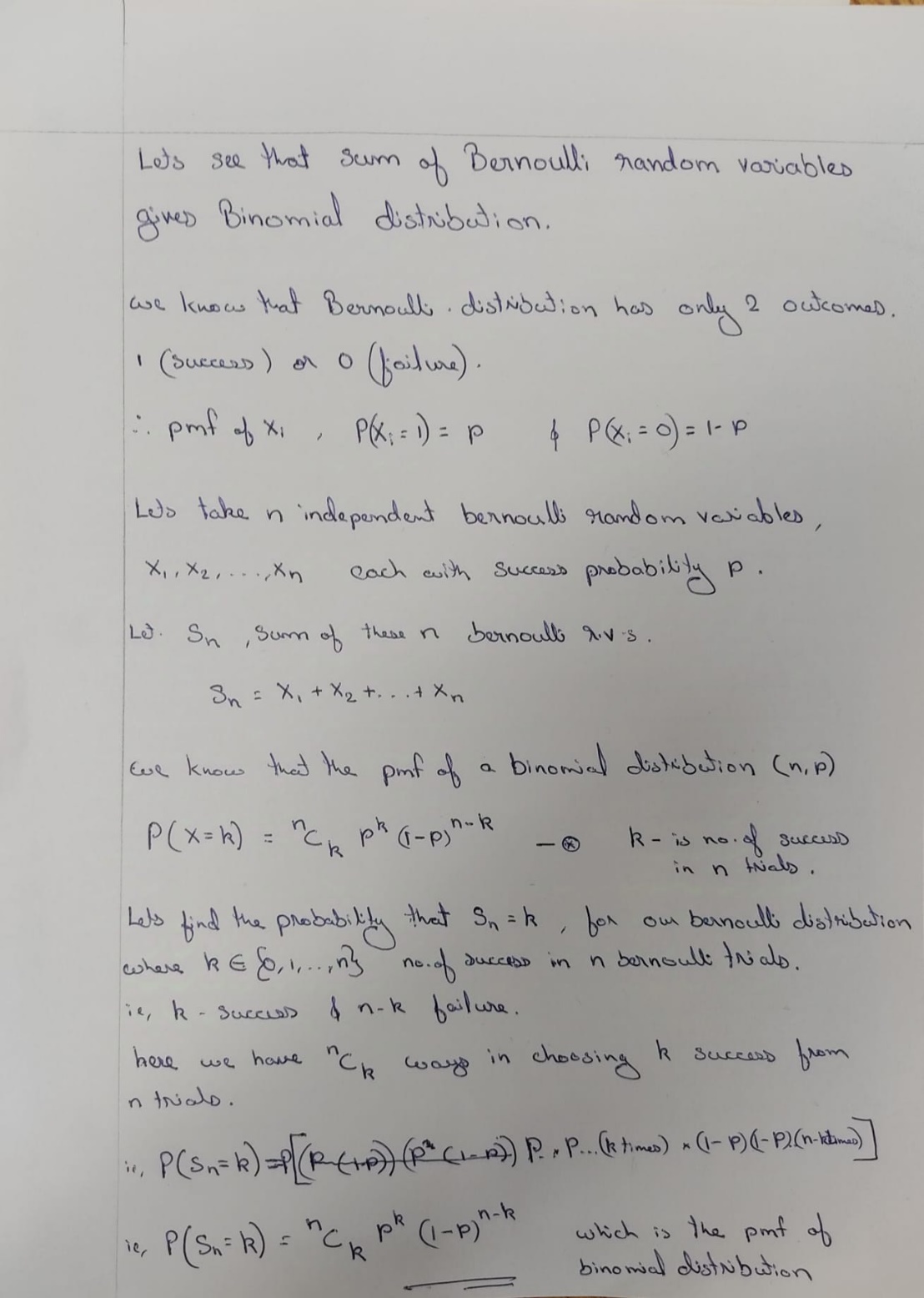
That is success is considered in the range 0 to 1/3 and above 1/3 it’s failure

U < p is considered as success. So we identified which all random numbers from U(0,1) are less than or equal to p and considered those as the success (a sample of Bernoulli Distribution). Then we transform these Bernoulli Distribution to Binomial by adding all Bernoulli trails.

**The transformation** **method**

It generates random samples from a given distribution by utilizing established distributions. Specifically, a binomial distribution with parameters n and p (binomial(n,p)) can be represented as the aggregate of n independent Bernoulli trials, each with a success probability p. In this context, each Bernoulli random variable Yi can take a value of 1 with probability p and a value of 0 with probability 1−p. By leveraging the transformation method, samples from a Binomial distribution can be produced by independently sampling from the Bernoulli distribution and then summing these outcomes. This approach allows for the synthesis of a Binomial distribution by sequentially building it from individual Bernoulli events.

Proof to show that the sum of Bernoulli random variables gives Binomial distribution is given below.



**Inversion Method**

The inversion method is a process used to create random samples from a probability distribution by converting uniform samples. This technique primarily relies on the cumulative distribution function (CDF), which, for a discrete random variable, manifests as a step function allocating probabilities to each possible outcome. To generate samples from a Binomial distribution via the inversion method, a uniform random variable U is produced from U[0,1]. The specific outcome k is then identified by referencing the CDF of the Binomial distribution to see where U falls within its range.

**Methodology**

We first generate a uniform random variable U from U[0,1]. We then identify the smallest integer k for which: U≤F X(k) This procedure aligns the uniform random variable U with the specific discrete outcomes of X, reflecting the probabilities delineated in the cumulative distribution function (CDF).

In essence, the inversion method is effective because it capitalizes on the characteristics of uniform random variables and the CDF of the intended distribution. By converting a uniform sample U into a sample from the Binomial distribution with X=F −1 (U), we guarantee that the sampled values adhere to the probability structure specified by the Binomial distribution.

**Combination of Inversion and Transformation Methods**

The transformation method represents a Binomial (n, p) random variable as the sum of n independent Bernoulli(p) random variables. It converts the results of independent Bernoulli trials into a Binomial distribution by summing their outcomes.

The inversion method, on the other hand, transforms a uniform random variable into a desired distribution using the cumulative distribution function (CDF). The key idea is that a uniform random variable U∼U(0,1) can be mapped to the target distribution by applying the inverse of its CDF.

In practice, the inversion method is used to sample from a Binomial distribution by generating uniform random variables and converting them into Bernoulli outcomes. Each Bernoulli trial reflects the probability of success within the Binomial distribution. By summing these trials, we generate samples that accurately represent the characteristics of the Binomial distribution. This approach effectively merges the inversion method with the transformation of Bernoulli trials to ensure the final results conform to the Binomial distribution’s theoretical properties.

A graph of a number of success

Description automatically generated

The most common number of successes is around 3, which aligns with the expected mean of the Binomial(10, 1/3) distribution. The mean of a Binomial distribution is calculated as n pn×p, where n=10 and p=1/3 ​. This gives a mean of approximately 3.33, which explains why the peak of the distribution is centered around 3 successes.

**PART -C**

Central limit theorem states that the distribution of sample means of a sufficiently large number of i.i.d random variables follow a normal distribution regardless of the original distribution if the original distribution has a finite mean and variance.

Here if we are not aware of the original distribution, CLT is very helpful in finding the confidence intervals because according to CLT we know that the distribution of sample means follow a Normal distribution. So we can determine Confidence intervals,

CI = E(X) + Z (SE)

Where E(X) is the mean

Z – is the z value for the Confidence level

SE – Standard error = Sigma/ Sq root of sample size

Here lets choose 95% as confidence level, therefore the z value will be 1.96

That is, CI = E(X) + 1.96 (SE)

Which means we are 95% certain that our true population mean lies between sample mean + 1.96 times SE and sample mean - 1.96 times SE.

Here we make use of Monte Carlo integration to find the expectation of the distribution by taking the mean of the random samples instead of calculating the true expectation directly (10/3).

Monte Carlo integration is a statistical technique used to approximate the value of definite integrals through random sampling.

Here using Monte Carlo Approach, we generate 100 random samples from Binomial (10,1/3) each of these sample represents an outcome from 10 trials with p=1/3. And then we take average of these random numbers and which serves as your Monte Carlo estimate of the true mean.

**Monte Carlo Integration**

Monte Carlo integration relies on random sampling to estimate mathematical functions or to compute integrals. Here, we will be using it to estimate the expectation of our Binomial random variable. We generate random samples from the distribution and then we calculate the sample mean, which will act as our estimate of the expectation.

The expectation E[X] of a random variable X can be estimated using the sample mean x̄ of n independent samples x1,x2,…,xn drawn from the distribution Expectation of X is equal to average of the sum of random variables xi

**Central Limit Theorem (CLT)**:

The CLT states that for a large sample size(n), the sampling distribution of the sample mean will be approximately normally distributed, regardless of the original distribution’s shape, provided that the samples are independent and identically distributed.1

The above property allows us to make inferences about the population mean based on sample means and thus is fundamental in Monte Carlo methods.

According to the Central Limit Theorem (CLT), for large n, the distribution of the sample mean approaches a normal distribution,

x̄ ∼ N (μ, σ/√n)

̄ X sampling distribution of the sample means

μ - the mean of the population

σ - standard deviation of the population

*n* - sample size

The standard error is,

SE = SD(X)/√n

Using the normal approximation, the **confidence interval** (CI) for the mean, is:

CI = x̄ ± z \* SE

where z is the z-score corresponding to the desired confidence level.

For 95% confidence, α = 1 – 0.95

= 0.05

Z0.05 = 1.96

**Application: Estimation Using Monte Carlo Integration**

First we defined the number of trials (n\_trials) = 10,

the probability of success (p\_success) = 1/3, and

the number of samples (n\_samples) = 100.

Then we used the runif () function to generate 100 uniform random numbers between 0 and 1and then to transform these uniform samples into Binomial samples using the qbinom() function with the specified size and probability. Here we are using the inversion method.

Later, we calculated the sample mean of the generated binomial samples which is an estimate of the expectation.

After which we computed the standard deviation of the Binomial samples and the standard error by dividing the standard deviation by the square root of the number of samples.

And then we calculated the z-score for a 95% confidence interval using the qnorm() function. Then using the estimated mean and standard error, we found the lower and upper bounds of the confidence interval

Estimated Expectation: **3.45**

This is the average number of successes in 100 trials of the Binomial (10, 1/3) distribution based on the samples. It is an estimate of the

theoretical expectation, which is 10/3 ≈ 3.33. We can see that the estimated mean from the sample is close to the theoretical mean, indicating that the sampling method is effective in approximating the expectation.

Standard Error: **0.1513**

The standard error from the Monte Carlo integration was found to be approximately 0.1309599. This value indicates the expected variation in the sample mean due to sampling variability. A smaller standard error suggests that the sample mean is a more precise estimate of the true population mean.

95% Confidence Interval: **3.153 to 3.746**

The confidence interval gives a range in which we expect the true mean of the Binomial distribution to lie with 95% confidence. This means that if we were to take many samples and calculate the confidence intervals for each, approximately 95% of those intervals would contain the true expectation 10/3 ≈ 3.33. The calculated interval suggests that we can be confident that the true mean falls within this range, providing a degree of assurance about the accuracy of our estimate.

Theoretical confidence interval can be calculated as below

The theoretical mean

E[X] = n \* p

=10 \* 1/3

≈ 3.33

**QUESTION 2**

Here we need to draw samples from Poisson distribution using U(0,1) by making use of transformation method.

We know that the Poisson distribution counts the number of events in a fixed interval, while the exponential distribution describes the time between those events.

The pmf of Poisson distribution is,

A black math equation with a white background

Description automatically generated with medium confidence

where ‘t’ is the average number of events in the given interval

We know that if Xi are i.i.d. Exponential (1) random variables and Sn = Pn i=1 Xi , then P(Sn ≤ t ≤ Sn+1) = P(X=n).

Therefore, here we can generate Poisson samples from Exponential random variables through these steps:

1. First, we set a fixed time interval (t=1)
2. Then we generate Exponential random variables
3. Sum the Exponential random numbers until the sum exceeds the fixed interval
4. Now these sum gives a Poisson random variable.

Here we generate exponential random samples from U(0,1) using the inversion method

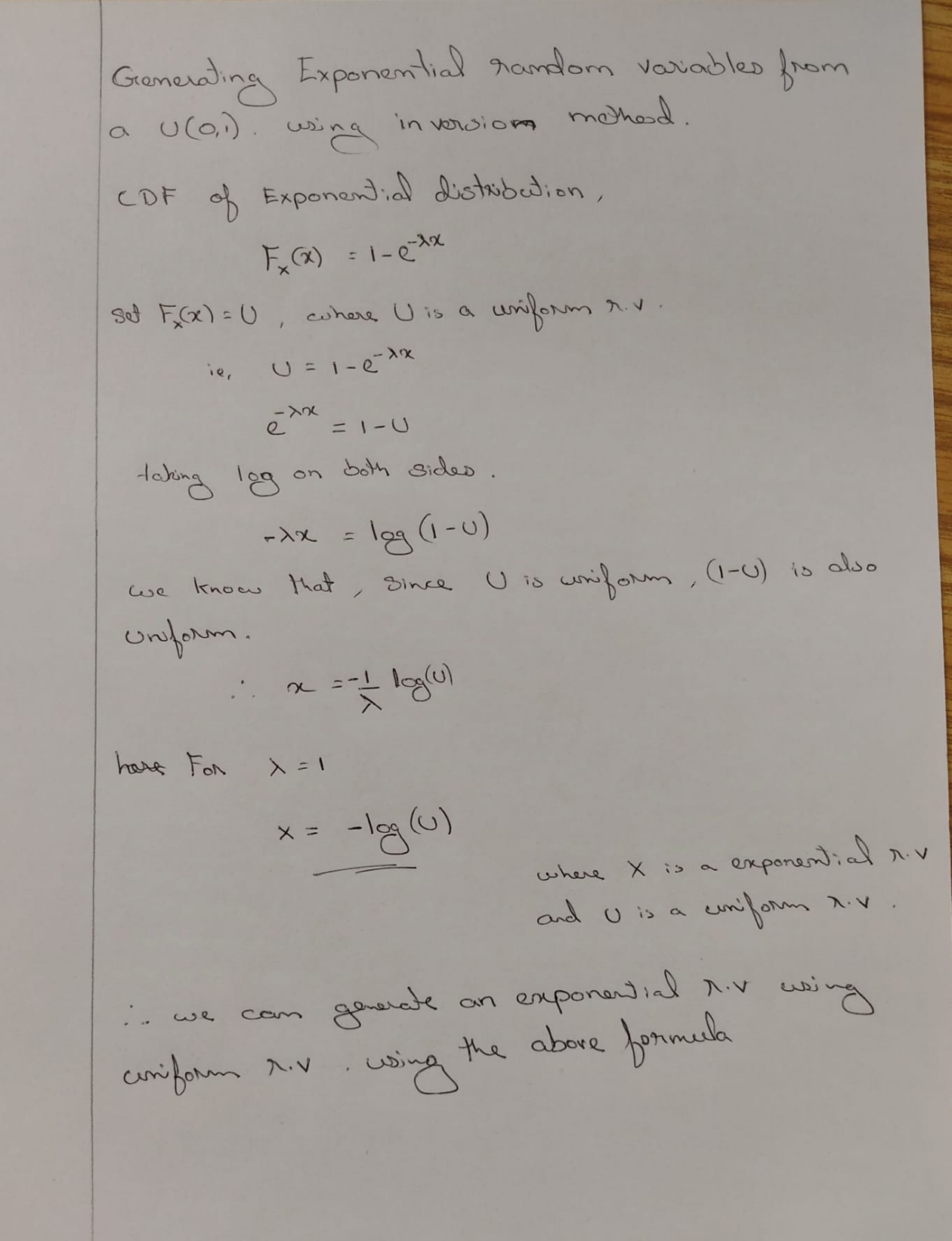
Equation

X = -log(U)

Where X is exponential random variable

U uniform random variable

Proof:



A graph of a number of events

Description automatically generated

**Results**

Sample size: 10

Estimated Mean: 1.2

Standard Error: 0.3

95% Confidence Interval: 0.33325 to 2.066744

Sample size: 100

Estimated Mean: 1.0

Standard Error: 0.0932033

95% Confidence Interval: 0.817321 to 1.82678

Sample size: 1000

Estimated Mean: 1.03

Standard Error: 0.0315609

95% Confidence Interval: 0.968140 to 1.0918569

Sample size: 10000

Estimated Mean: 1.0028

Standard Error: 0.010035

95% Confidence Interval: 0.983130 to 1.022469

**Interpretations**

As the sample size increases, the estimated mean becomes more accurate , and closer to the true mean of the Poisson distribution which is 1. The standard error decreases with increasing sample size, which aligns with the expectation that larger samples provide more precise estimates of the mean. The 95% confidence intervals narrow with larger sample sizes, reflecting increased certainty about the estimated mean. For the largest sample size of 10,000 , the confidence interval is very tight , showing that the estimate is very precise . Also we can clearly see that the histogram bars at x=0 and x=1 is nearly equal, this is because both probabilities are approximately 0.3679. We can conclude that, the bars X=0 and X=1 are nearly identical because the probabilities for these outcomes re mathematically equal**.**

**QUESTION 3**

**PART A**

Rejection sampling is a Monte Carlo technique used to generate samples from a target probability distribution by utilizing a proposal distribution. This method is particularly useful when direct sampling from the target distribution is challenging. The basic idea involves generating samples from a proposal distribution and accepting or rejecting them based on a specific criterion

.

**Target Distribution**

Here, we have a target distribution defined as a mixture of two normal distributions:

N(1,0.5) with weight α1=0.2

N(2,0.1) with weight α2=0.8

The target distribution can be expressed as:

f(x) = α1⋅N (1,0.5) + α2⋅N (2,0.1)

**Proposal Distribution**

To perform rejection sampling, we need to choose a proposal distribution g(x) such that it captures as much of the mass of the original distribution as possible. A suitable choice for the proposal distribution is a normal distribution:

Mean, μg = 2: This is chosen because the majority of the distribution's mass is centered around the second component, N(2,0.1).

Standard Deviation, σg = 0.5: This value reflects the spread of both components without underestimating the narrower second component.

**Acceptance Rate**

In rejection sampling, the acceptance rate is defined as the ratio of the areas under the target distribution f(x) and the scaled proposal distribution M⋅g(x), where M is a constant ensuring

f(x)≤M⋅g(x) for all x. The acceptance rate can be calculated as:

Acceptance rate = 1/M

To find M, we determine the maximum value of the ratio f(x) / g(x) across all x

**Justification for the Proposal**

We are setting mean, μg = 2, since the majority of the probability mass in the target mixture is centered around the second normal component, which has a higher weight α2 = 0.8.

Setting standard deviation, σg = 0.5 balances the wider spread of the first component and the narrower spread of the second component.

**To Estimate the Acceptance Rate**

To estimate the acceptance rate numerically, We can follow these steps:

Simulate samples from the proposal distribution g(x).

Apply the rejection criterion based on the acceptance ratio f(x)/M⋅g(x)

The acceptance rate is approximately the fraction of accepted samples out of the total generated samples.

**Sanity Check Using the Composition Method**

To verify the correctness of the rejection sampling, we can use the composition method:

With probability α1 = 0.2, sample from N(1,0.5)

With probability α2 = 0.8, sample from N(2,0.1).

**Code**

To implementing rejection sampling:

* Target Distribution: This is the mixture of two normal distributions N(1,0.5) and N(2,0.1), weighted by α1 = 0.2 and α2 = 0.8

The function target\_distribution(x) calculates the probability density of the target mixture at a given value of x.

* Proposal Distribution: A single normal distribution N(2,0.5) is used as the proposal, capturing most of the mass of the mixture.
* Rejection Sampling Process: For each sample, propose a value from the proposal distribution.

Generate a uniform random number u∈[0,1].

Accept the proposed sample if the condition u<f(x)/ M⋅g(x) holds, where f(x) is the target distribution and g(x) is the proposal distribution. M is the scaling factor (set as 3 here, but it could be optimized).

* Plotting: The code visualizes the target distribution, proposal distribution, and the histogram of accepted samples.
* Acceptance Rate: The estimate\_acceptance\_rate() function simulates the rejection process multiple times to estimate the acceptance rate of the rejection sampler.

A green and blue graph

Description automatically generated

**Target Distribution :** Represents the probability density function of a mixture of two normal distributions, one likely centered around 1 and the other around 2, each with different variances.

**Proposal Distribution :** Illustrates the chosen proposal distribution N(2,0.5), centered at 2 with a standard deviation of 0.5. This distribution is selected to adequately cover major portions of both components in the mixture.

**Rejection Samples :** Showcases the samples that were successfully accepted through the rejection sampling process. These samples should, ideally, resemble the target distribution closely if the proposal distribution is appropriately suited and the acceptance rate is sufficiently high.

**Estimated Acceptance Rate (30.20%):** Reflects the percentage of samples drawn from the proposal distribution that were accepted as valid representations of the target distribution. This rate evaluates the efficiency of the sampling process, where higher rates indicate less computational waste on generating non-useful samples.

**Interpretation**

**Acceptance Rate:** An acceptance rate of 28.30% is considered relatively efficient for this method, given the straightforward nature of rejection sampling. Nonetheless, further optimization of the proposal distribution might enhance this rate.The central peak of the accepted samples closely aligns with the highest peak of the target distribution, suggesting that the proposal distribution is effectively capturing the most probable values of the target. However, the tails of the target distribution seem to be less represented by the samples, likely due to the proposal's narrower range compared to the broader spread of the target's distribution.

**COMPOSITION METHOD**

**Now implementing the composition method as a sanity check.**

Composition method is a technique used in statistics to generate samples form a mixture of distributions. Here we have mixture of 2 normal distributions with different weights 0.2 and 0.8.

So lets generate random samples from uniform distribution (0,1) and sample among the 2 normal distributions based their weights

That is ,

* If u<0.2, sample from N(1,0.5).
* If u≥0.2, sample from N(2,0.1).

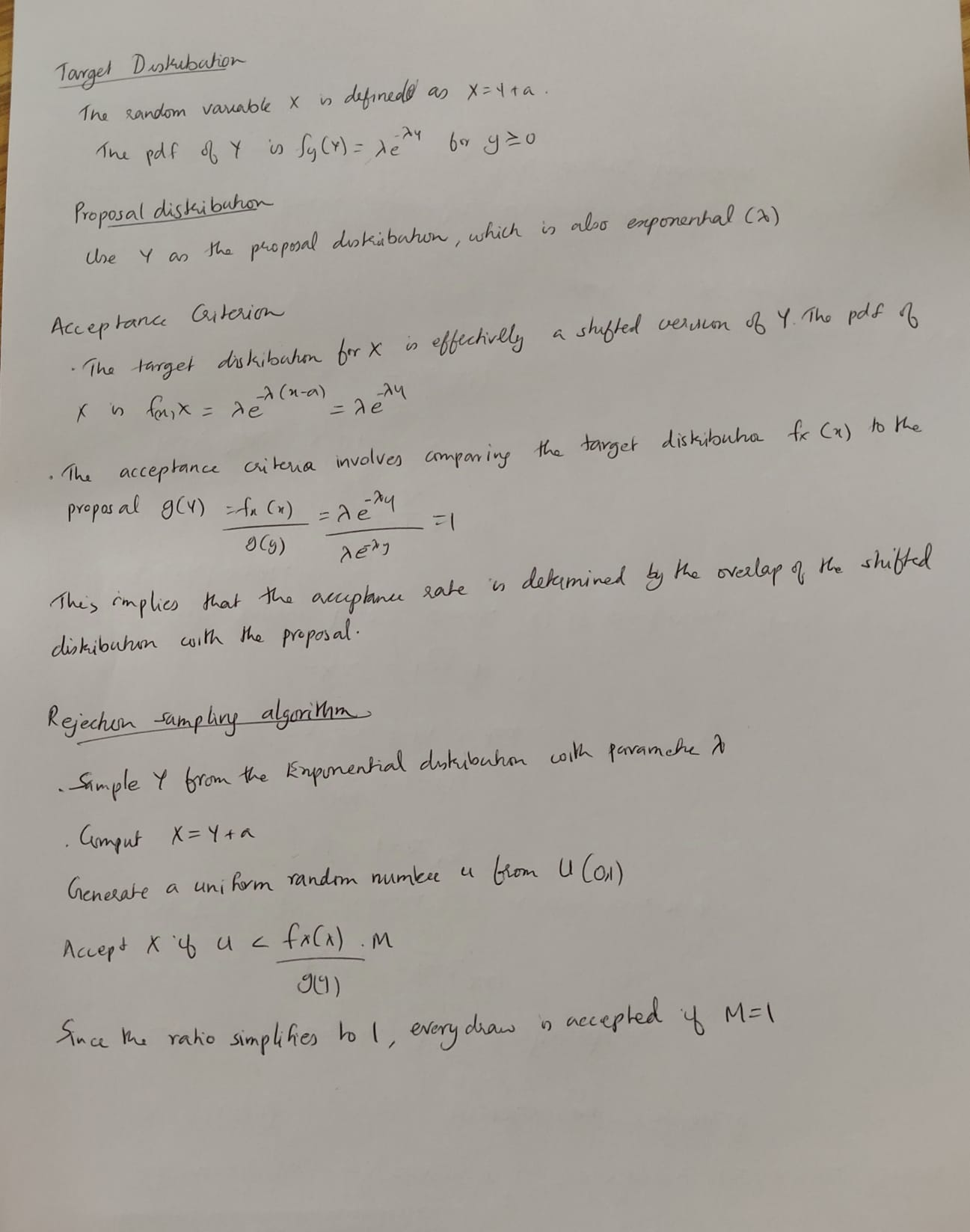
And now we can compare the distribution of these samples with the previous one.

A graph of a normal distribution

Description automatically generated

This graph demonstrates that the composition method has successfully captured the essential characteristics of the mixture of normal distributions. Theres a similar pattern that can be observed on comparison with the rejection sampling graph which indicates a correctness of our approach. When compared with the histogram from the previous rejection sampling method, both approaches seem to deliver comparable outcomes in terms of capturing the central peak, which indicates a consistency in methodological execution. However, the smoother and broader spread observed in the composition method may be preferable for applications that require a more detailed representation of the distribution, particularly in capturing the behavior of its tails.

**PART B**

`

**When ‘a’ grows**

As a increases, X values (adjusted by a) necessitate correspondingly higher Y values to fulfill the equation X=Y+a. Consequently, securing sufficiently large Y values becomes progressively unlikely because such values are rare within the Exponential distribution. Although every Y sample drawn is accepted, the process to find appropriate Y values for X becomes increasingly resource-intensive due to the infrequency of large Y values, leading to more draws required. Thus, the sampling method, while maintaining an acceptance probability of 1, becomes inefficient and computationally demanding, necessitating a substantial number of iterations to collect enough valid samples. Furthermore, as 'a' increases and the X distribution shifts rightward, capturing the necessary range of values may require even more samples, thereby amplifying the computational burden.