CS215 Assignment 1 Report

Arpon Basu
Shashwat Garg

Autumn 2021

Contents

1 Introduction

Hello and welcome to our report on 215 Assignment 1. We have tried to make this report as reader friendly as we could and we hope reading this would give you a proper flowing description of our work and the results obtained. Feel free to refer to the code to know the exact implementation of our tasks. The pictures included in the graphs folder as a part of this report as well.

We have referred to several sites on the web for finding the python implementation and general statistical knowledge of various parts of the assignment and most of the resources we consulted have been cited appropriately.

In many places, to better give context to the place from which the questions could have arisen, some theoretical discussions have been engaged in.

Hope you enjoy reading the report. Here we go!

2 Problem 1

2.1 What Theory Says

In this question we have been asked to plot the PDF and CDF of several different functions, along with calculating the variance whenever possible. The functions are-

- **Laplace Distribution**

  As per wikipedia, the Laplace distribution is the **distribution of the difference of two independent random variables with identical exponential distributions**.
  Also called the double exponential distribution, this distribution is often used to **model phenomena with heavy tails or when data has a higher peak than the normal distribution.**

- **Gumbel Distribution**

  The Gumbel distribution is used to **model the distribution of the maximum (or the minimum) of a number of samples** of various distributions. For modelling the minimum value, we use the negative of the original values.
  The PDF of the Gumbel distribution also has an almost constant shape and shifts depending on the location parameter.

- **Cauchy Distribution**

  Cauchy Distribution is a very interesting example of Statistical Distributions, It is an example of a "pathological" distribution since **both its expected value and its variance are undefined**.

2.2 Code Flow

The code does 3 things, plotting PDF, plotting CDF and calculating variance (whenever possible)

We first **define some generic variables**, specifying the location and scaling parameters for the various distributions.

Then we define a generic **pdf_plot()** function which given a function and some parameters, plots the function.

This is followed by **defining the three distributions** in terms of their pdf. Then we repeatedly pass the functions along with the parameters into the plotting functions to obtain the required graphs.

Next, we define a **generic function for plotting the CDF** of a distribution which uses PDF along with calculating Riemann sums for plotting the CDF graph. We have used the ranges from $\mu - 100b$ to $\mu + 100b$, to define $\pm\infty$.

Finally for calculating variance, we have again used Riemann sum method on the random variable defined by $(x - \mu)^2$.

Here we were intrigued by the behaviour of the Gumbel Distribution. We were hopelessly trying to match the variance when we found that the **location parameter of the Gumbel Distribution is not actually the mean**, even though we use the symbol μ to write both. That was a good learning event, which further led to us exploring more about statistical distributions. On the way, we also found that the **Cauchy distribution does not have finite mean and variance.** That is because although one might naively think that

$$\int_{-\infty}^{\infty} \frac{x}{\pi(1+x^2)}\, dx = 0$$

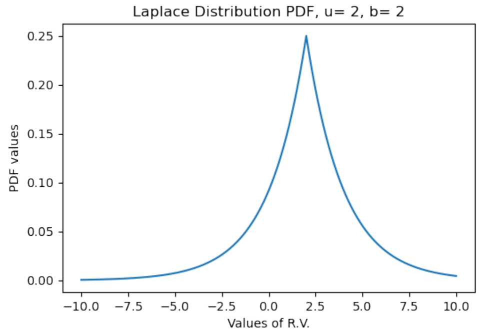
and hence the mean of the Cauchy distribution should be zero, **one should keep in mind that $\mathbb{E}[g(X)]$ is defined if and only if $\int_{-\infty}^{\infty} |g(x)| f_X(x)\, dx$ converges, and in our case $\int_{-\infty}^{\infty} \frac{|x|}{\pi(1+x^2)}\, dx$ does NOT converge, and hence the mean is NOT defined.**

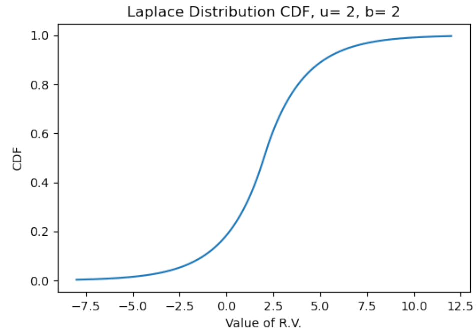
Since the mean doesn't exist, the variance, which presumes the existence of the mean in it's definition, doesn't exist either.

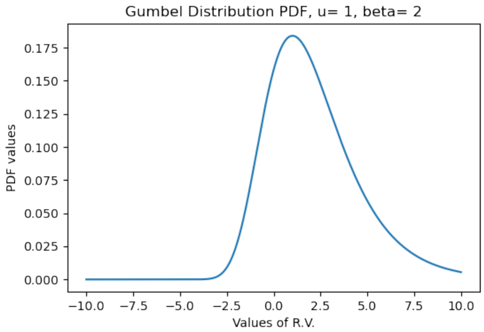
This finishes this question's coding part.

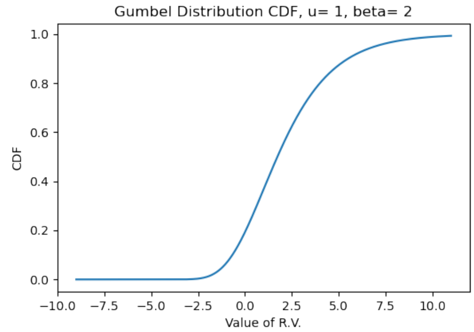
2.3 Graphs and the Results obtained

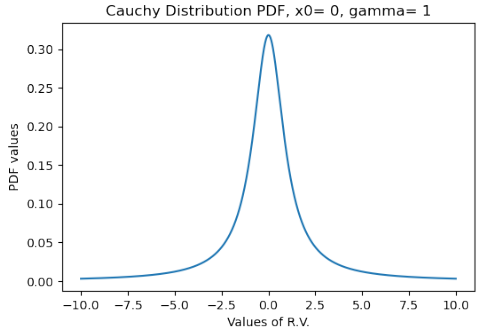
The results obtained are as follows-

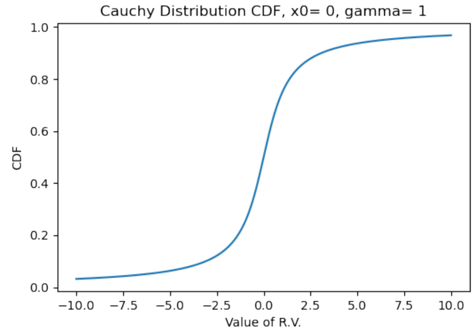












Difference in analytical and approximated variance-

- Laplace Distribution Error- 2.0000000812814278e-05 where true variance is 8
- Gumbel Distribution- 1.6449341313951038e-05 where true variance is 6.5797
- Cauchy Distribution- Undefined

2.4 Observations and their Rationalization

We observed how increasing the number of sample points makes the graph smoother. By adjusting various values of the location and scaling parameters, **we observed and understood the behaviour of parameters on the functions.**

We also understood how both location and scale parameters can affect mean and variance in some pathological cases, or that there might not even exist a mean or variance at all.

We also learnt basic implementation strategies to plot different functions and their PDF, CDF and other corresponding values easily.

3 Problem 2

3.1 What Theory Says

This problem deals with operating on the **Poisson distribution variable** and **understanding the behaviour of sum of random variables** and Poisson thinning.

The process of summing the random variables, say $Z = X + Y$ is carried out analytically by combining the different possibilities in which the combination of x and y could lead to the required value of z.

Implementing this in code is simpler since we can just take a large amount of samples and then combine the outcomes of the corresponding individual outcomes, to get the overall random variable. No need of the double summation (Σ).

Poisson thinning is basically a **filtering process**. Suppose Y is the thinned R.V. obtained from X, with probability p. So we require to keep each value of Y as a value from the **Binomial Distribution with parameters as values taken by X and the probability factor of the binomial distribution as p.**

3.2 Code Flow

The code begins with specifying the parameters for the Poisson random variables, X and Y.

We then **obtain a distribution of $Z = X + Y$** by adding the distributions of the random variables X and Y.

We then construct a plot of the obtained (experimental) distribution of Z and compare it with the actual distribution (analytical) of Z, ie. the Poisson R.V. with λ_z as $\lambda_x + \lambda_y$

Both plots seem to match near perfectly. We then list out the errors between the plots which is also of the order 10^{-4} on average.

Next, we implement the Poisson thinning process. We begin with specifying the original Poisson random variable Y and obtain a distribution containing 100,000 instances of the same.

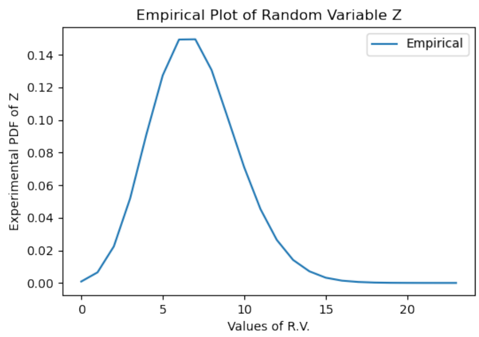
We then use the **binomial distribution with probability p on Y** to obtain the distribution of the variable Z.

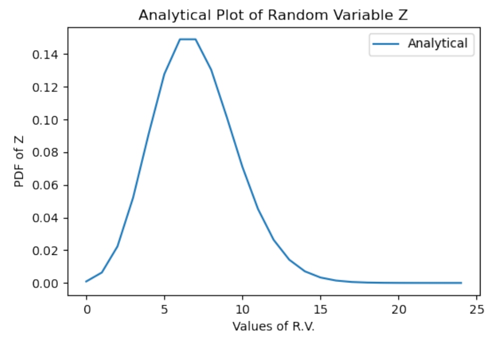
Comparing the obtained values with the actual distribution of Z which is actually the Poisson distribution with $\lambda_z = \lambda_y * p$, gives a near perfect match, confirming the correctness of the procedure.

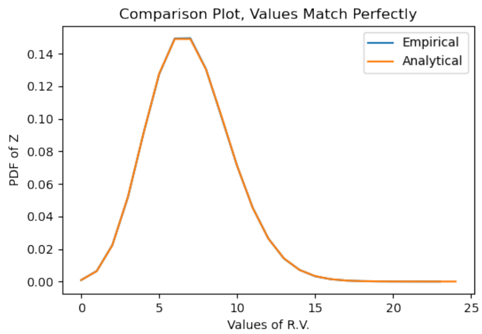
Again the error is of the order 10^{-4} on average.

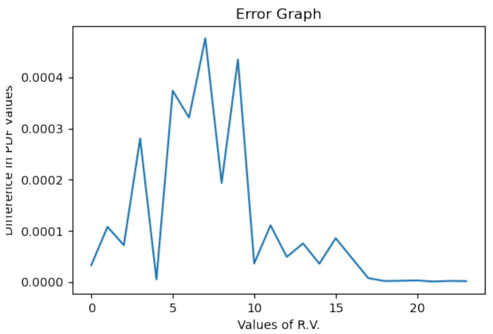
3.3 Graphs and the Results obtained

3.3.1 Sums of Random Variables

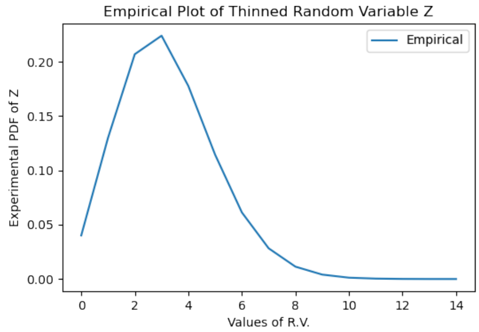


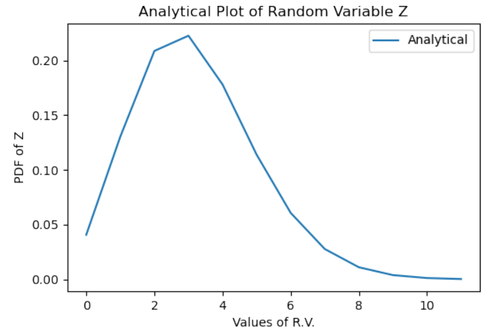


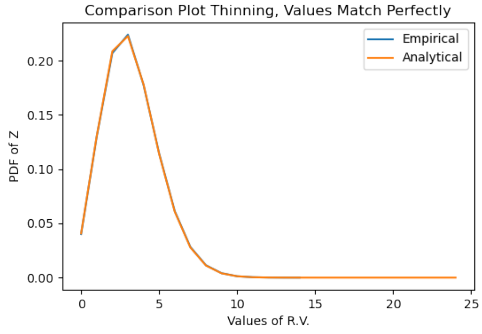


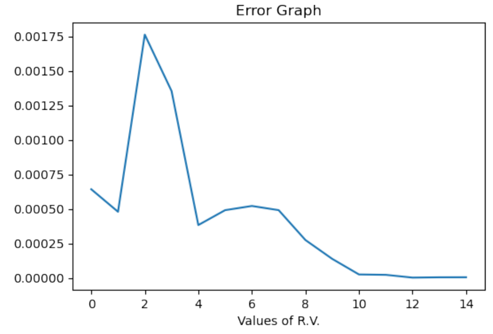


3.3.2 Poisson Thinning









3.4 Observations and their Rationalization

We learned about the concept of Sums of Random Variables. We saw how taking enough eventually leads us back to the original distribution.

We observed how Poisson thinning starts with applying the binomial distribution on the random variable Y to obtain Z, which finally is a Poisson distribution only.

Apart from the mathematical proof of the same, this behaviour can be understood by realising that **the λ parameter of the Poisson distribution is actually the average number of hits per unit time**, so to speak. Thus, is we allow only p fraction to come through the effective distribution would stay the same, just get scaled down in magnitude of values by p.

4 Problem 3

4.1 Introduction

This problem deals with the classical topic of random walks (for our case, one-dimensional), the importance of whom can't be underestimated. So we'll analyse random walks, in two sections, first the coding implementation and then a theoretical one, to answer all the questions posed in the assignment.

Also, a significant portion of the assignment involves proving results associated with the Law of Large Numbers. Thus, that too will be presented in a different section.

Finally, **we have carried out an extra computation on our own, ie:- to theoretically calculate and then show the envelope of the random walk plots that are obtained**. Since the calculations are a bit involved, they too are put in a different subsection. Finally, we present all our results in the results subsection.

4.2 Code Flow

The code for this too is not that involved : Firstly, we *generate* a random walk (through the function **generateRandomWalk**) by initializing an array filled with zero, and probabilistically adding either -0.001 or 0.001 (the step size) to each element in the array to obtain the next element.

After we're done with this, we create 10,000 random walks (through the function **generateSetOfWalks**), and store them. Then we develop two visualization functions (**plotHistogram** and **plotWalks**), for rendering the above data in terms of histograms and plots, as required by the assignment.

Then two more functions are used to calculate the empirical mean and variance of the terminal positions of the above walks. They print the error b/w themselves and the true mean/variances too.

Finally, in the final cell of our notebook, we call the functions we have written, one by one, to demonstrate everything that had been asked for.

4.3 Random Walk Analysis

So basically, what is a random walk in one dimensions? A person starts at the origin, and and for each unit interval of time δt, takes either a unit step (δz) forwards, or backwards, with probabilities p and $q = (1-p)$ respectively. In the context of this problem, we shall assume that $p = \frac{1}{2}$.

Even without any mathematical justification, from the above formulation itself it's intuitively clear that over a large number of steps, the person will, on an average, take as many steps forward as backward, and thus the **expected** position of the person after a long time t ($= n\delta t$, where n is the number of steps taken) will be at $z = 0$.

Now, suppose we want to mathematically analyze the probability distribution of a random walk. How do we do so? We first observe that if x is the number of forward steps taken, then:

$$z = x\delta z - (n-x)\delta z = (2x-n)\delta z$$

Also, the probability that exactly x forward steps will be taken can be modeled via the **binomial distribution**, ie:-

$$P(x) = \binom{n}{x} p^x q^{n-x}$$

Considering n, x to be much much greater than 1 (so as to allow us to apply the Stirling's approximation, as in $n! \sim \sqrt{2\pi n}(n/e)^n \mathcal{O}(1 + \frac{1}{n})$), and after some mathematical jugglery, we obtain:

$$P(x) = \frac{1}{(2\pi npq)^{1/2}} e^{-\frac{(x-np)^2}{2npq}}$$

Applying the transformations $t = n\delta t$, $z = (2x-n)\delta z$, $P(z;t) \cdot 2\delta z = P(x)$, and most importantly, defining the **diffusion coefficient** D as $\frac{(\delta z)^2}{2\delta t}$, we obtain $P(z;t) = \frac{1}{(4\pi Dt)^{1/2}} e^{-\frac{z^2}{4Dt}}$. Thus, as asked for in the report,

This resembles the Gaussian PDF and thus we can compare the mean and variance of the two directly.

We have $P(z;t) = \frac{1}{(4\pi Dt)^{1/2}} e^{-\frac{z^2}{4Dt}}$ and the Gaussian Distribution as $P(x) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

Thus, the mean is 0 and the variance is $2Dt$. Hence,
The true mean of the terminal positions for a random walk with probability $p = \frac{1}{2}$ is zero.
The true variance of the terminal positions for a random walk with probability $p = \frac{1}{2}$ is $2Dt = 2\frac{(\delta z)^2}{2\delta t}(n\delta t) = n(\delta z)^2 = 0.001$, since in our case $n = 10^3$ steps and $\delta z = 0.001$.

4.4 Calculations involving averages of Random Variables, the Law of Large Numbers

Coming to the next question raised by the assignment, which digresses a bit and asks us to prove that the random variable $\hat{M} := \frac{1}{n}\sum_{j=1}^{n} X_j$ converges to it's mean $M := \mathbb{E}[X]$ (ie:- becomes a one point random variable), where X_1, X_2, ..., X_n are independent draws from a random variable X. Note that $\mathbb{E}[\hat{M}] = \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}[X_j] = \frac{1}{n}\sum_{j=1}^{n} M = M$. Thus, we just need to prove that as $n \longrightarrow \infty$, **the probability that $\hat{M}$ takes any other value than M tends to zero**.

For proving this, note that by Chebyshev's inequality,

$$P(|\hat{M} - M| \geq \epsilon) \leq \frac{\sigma_n^2}{\epsilon},$$

where σ_n^2 is the variance of $\hat{M}$. But then also note that since X_1, X_2, ..., X_n are all mutually independent (and hence the variance of their sum is the sum of their variances),

$$Var(\frac{1}{n}\sum_{j=1}^{n} X_j) = \frac{1}{n^2}\sum_{j=1}^{n} Var(X_j) = \frac{\sigma^2}{n} \implies \sigma_n = \frac{\sigma}{\sqrt{n}}$$

Thus,
$P(|\hat{M} - M| \geq \epsilon) \leq \frac{\sigma_n^2}{\epsilon} = \frac{\sigma^2}{n\epsilon} \longrightarrow 0$ as $n \longrightarrow \infty$; $\forall \epsilon > 0$, as desired (Note that this effectively was a proof of the **weak version of the Law of Large Numbers**. See Problem 5 for more details).

For the random variable $\hat{V} := \frac{1}{n}\sum_{j=1}^{n}(X_j - M)^2 = \frac{1}{n}\sum_{j=1}^{n}(X_j - \mathbb{E}[X])^2$, we have, after expanding the square-

$$\hat{V} = \frac{1}{n}\sum_{j=1}^{n}(X_j - M)^2 = M^2 + \frac{1}{n}\sum_{j=1}^{n} X_j^2 - \frac{2M}{n}\sum_{j=1}^{n} X_j$$

As $n \longrightarrow \infty$, $\frac{2M}{n}\sum_{j=1}^{n} X_j \longrightarrow 2M^2$, borrowing from the above proof.

Also, in the above proof (for $\hat{M}$), let X^2 be our random distribution instead of X. Then we also have

$$\frac{1}{n}\sum_{j=1}^{n} X_j^2 \longrightarrow \mathbb{E}[X^2] \ as\ n \ \longrightarrow \infty$$

Finally, to finish the proof, consider the following argument : If $Q_{1,n}$ and $Q_{2,n}$ are 2 random variables which tend to their means as $n \longrightarrow \infty$, then

P($Q_{1,n} + Q_{2,n} \not\longrightarrow \mathbb{E}[Q_{1,n}] + \mathbb{E}[Q_{2,n}]; n \longrightarrow \infty) \leq P(Q_{1,n} \not\longrightarrow \mathbb{E}[Q_{1,n}]; n \longrightarrow \infty) + P(Q_{2,n} \not\longrightarrow \mathbb{E}[Q_{2,n}]; n \longrightarrow \infty) \longrightarrow 0$

This is because both $P(Q_{1,n} \not\longrightarrow \mathbb{E}[Q_{1,n}]; n \longrightarrow \infty)$ and $P(Q_{2,n} \not\longrightarrow \mathbb{E}[Q_{2,n}]; n \longrightarrow \infty) \longrightarrow 0$.

Hence $\hat{V}$ equals $M^2 + \mathbb{E}[X^2] - 2M^2 = \mathbb{E}[X^2] - \mathbb{E}[X]^2 =Var(X)$, for large enough n, with probability 1.

4.5 An Extra Derivation: Envelope Calculation of the Random Walk plots

Having answered all the questions in the problem, we now return to random walks for some concluding remarks.

As shown by the formula of $P(z;t)$, for a fixed time t (ie:- fixed n), we have that z, **which is the terminal position of the walk**, will have a Gaussian distribution for large enough n. Also, even though we saw that the expected **position** of a random walker is zero, the expected **translation**, which is the absolute value of position, ie:- $|z|$, is not zero.

In fact, the PDF of z is what is known as the **folded normal distribution** in literature (click here to obtain reference). Directly using the result given in the source, we get

$$\mathbb{E}[|Z|] = \sigma\sqrt{\frac{2}{\pi}}e^{-\frac{\mu^2}{2\sigma^2}} + \mu(1 - 2\Phi(-\frac{\mu}{\sigma}))$$

Where Φ is the CDF of the Gaussian distribution. Since $\mu = \mathbb{E}[Z] = 0$, much of the terms vanish, leaving behind

$$\mathbb{E}[|Z|] = \sigma\sqrt{\frac{2}{\pi}} = \sqrt{2Dt}\sqrt{\frac{2}{\pi}}$$

Putting $2Dt = n(\delta z)^2$, we get

$$\mathbb{E}[|Z|] = \sqrt{\frac{2n}{\pi}}\delta z$$

Also,

$$\mathbb{E}[|Z|^2] = \mathbb{E}[Z^2] = Var(Z) + \mathbb{E}[Z]^2 = Var(Z) = 2Dt$$

Thus Var$(|Z|) = \mathbb{E}[|Z|^2] - \mathbb{E}[|Z|]^2 = \sigma^2(1 - \frac{2}{\pi})$.

Finally, assuming translation varies normally (we can do so due to Lippmann's joke:)), we obtain that the **envelope of the random walk plots would be** (note that we assume that 3 times the standard deviation in a normal distribution encompasses most (99.7%) of the data)

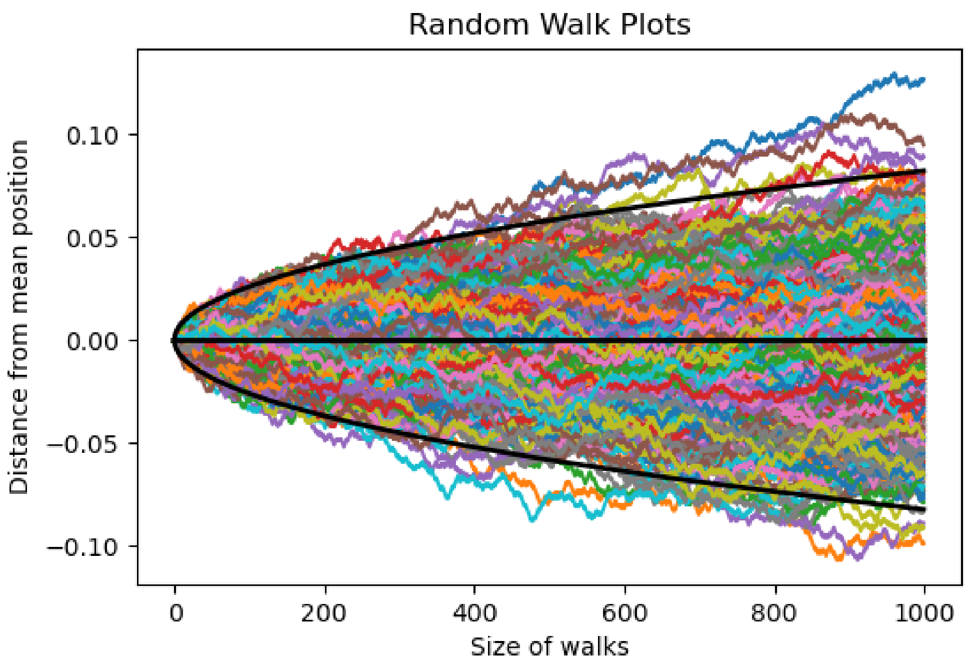
$$env(n) = \pm(\mathbb{E}[|Z|] + 3(Var(|Z|))^{\frac{1}{2}}) = \pm\sigma(\sqrt{\frac{2}{\pi}} + 3(1 - \frac{2}{\pi})^{\frac{1}{2}}) \approx \pm 2.606(\delta z)\sqrt{n}$$

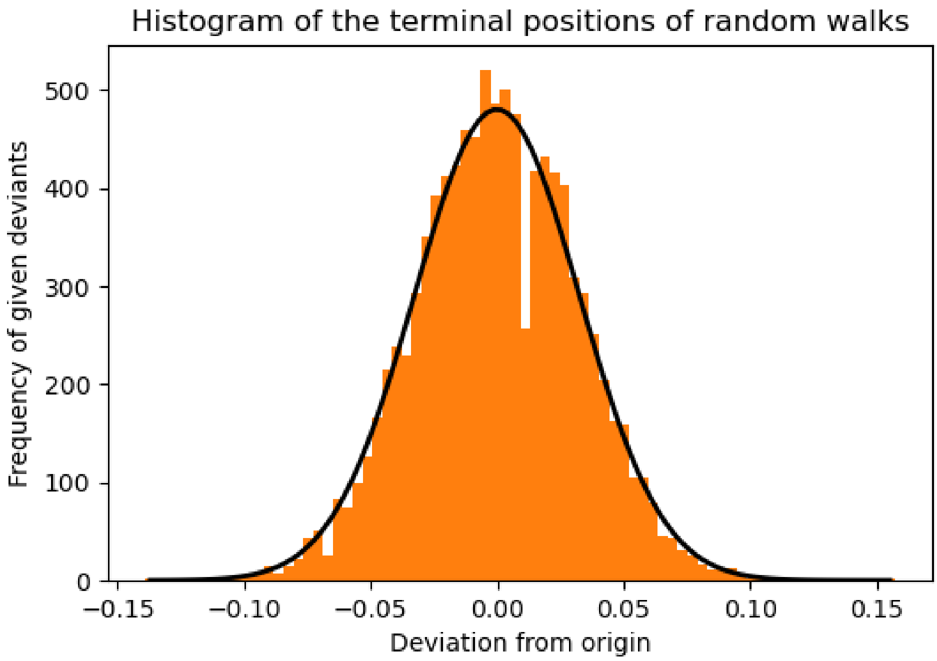
On plotting these curves over the random walk plots, one obtains excellent encompassing by the envelopes.

4.6 Results

Based on the discussion above, we present the histogram, random walk plots and the errors between the empirical and theoretical means and variances below for the graders perusal. Wherever curve fitting (A normal distribution fit to the terminal position histogram) or enveloping (a parabolic envelope to the random walk plots) has been done, one can see the curves as thick black lines.

The equations of those have not been included in the image for aesthetic reasons. The grader can check for their implementation though, in the code.





The empirical mean of the last locations of the random walkers is 3.7200000000000064e-05

The error between true mean and empirical mean is 0.0037200000000000063%

The empirical variance of the last locations of the random walkers is 0.0010203058161600013

The error between true variance and empirical variance is 0.0020305816160000132%

5 Problem 4

5.1 What Theory Says

This is a fairly intriguing question asking us to explore how the combination of enough of independent random variables from any distribution will **lead to a Gaussian** being formed.

5.2 Code Flow

The code begins with defining the M function, which has value $|M|$ for $|M| < 0$ and 0 otherwise.

We then define the inverse CDF of the M distribution and then use the uniform distribution alongwith the inverse CDF to generate the random variable sample array from scratch. The method uses a map to calculate the distribution values effectively based on the slope of the CDF values, which directly relates to the PDF. More on this later in the observations.

The next part is drawing the histogram of the array obtained, which resembles the M function as expected.

We then use the same array of the drawn random variable M values to obtain a CDF of the distribution, which again matches the expected shape, two parabolas joined together.

The remaining part of the question is just scaling up the model from one M distribution, to an average of a large number of such distributions.

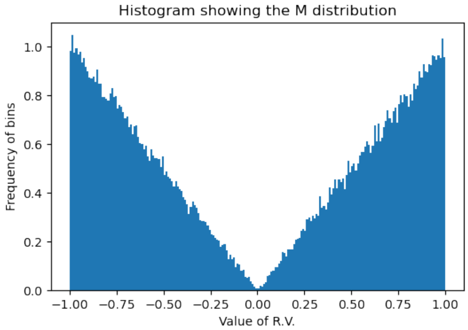
We define a new variable $Y_N = 1/N * \sum_1^n X_i$ which modulates the behaviour of average large number of such M functions. As expected the plots all tend towards the Gaussian Distribution.

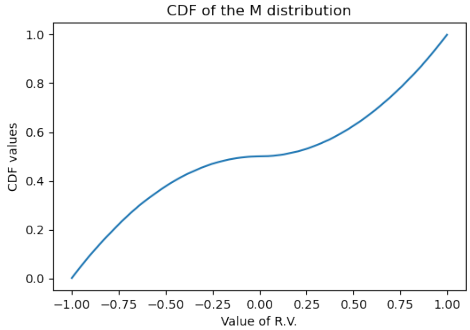
We have used iterative procedure for plotting since we wanted to get a deeper insight into the actual functioning of these concepts.

Hence, we have defined our own CDF plotter and Random Variable Generators, rather than using the inbuilt library functions, wherever we could.

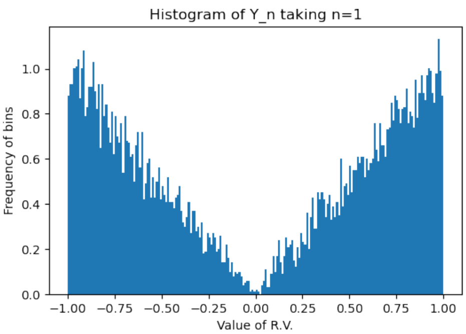
Finally, we get the combined CDF of all these Y_i which briefly shows the transitions happening in the random variables.

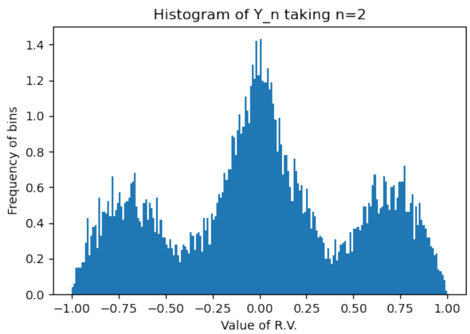
5.2.1 Plotting X_i

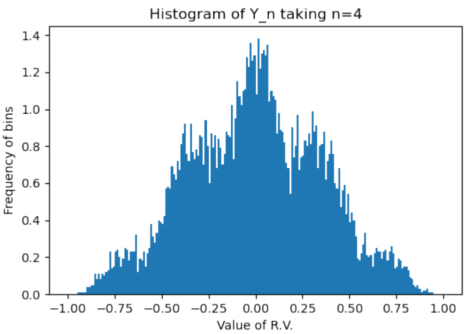


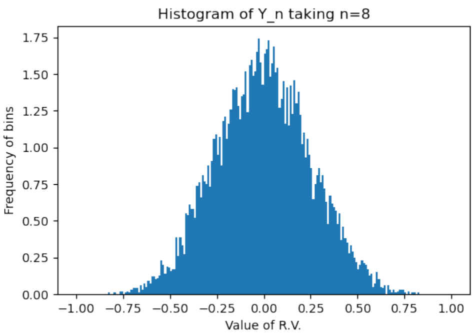


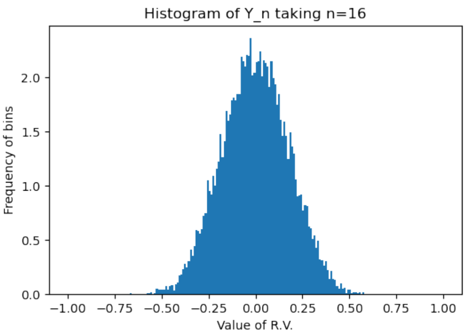
5.2.2 Plotting Y_i for various values of i

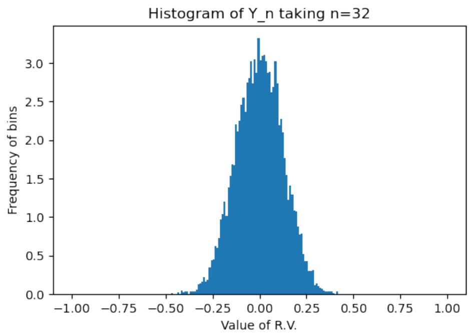


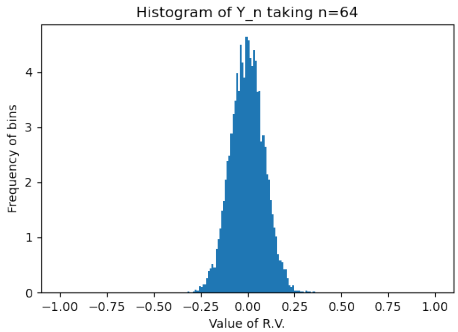




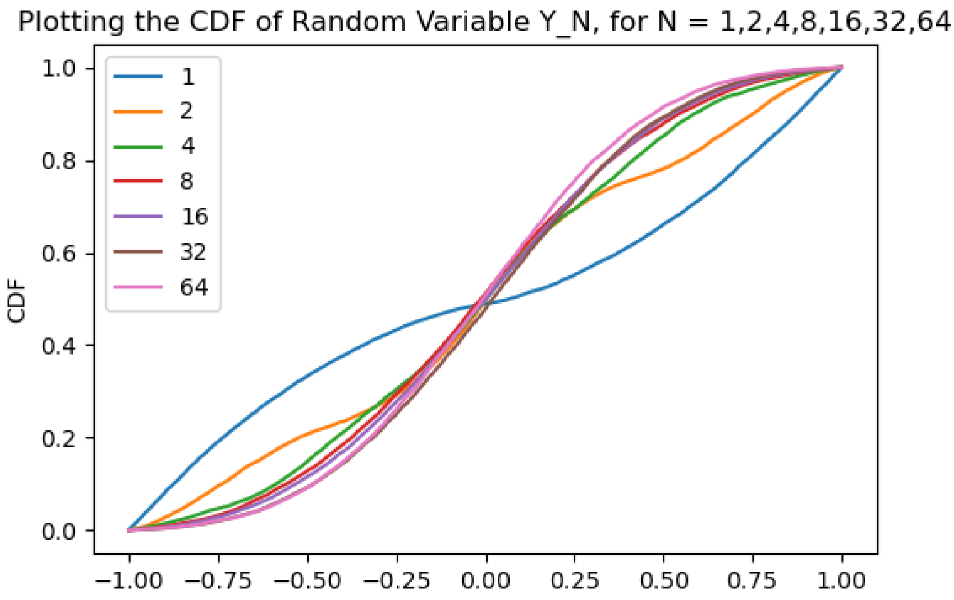








5.2.3 Final CDF Plot



5.3 Observations and their Rationalization

Thus, we observe how the Combination of several random variables of any distribution finally gives a Gaussian Plot as output.

We observe that the Gaussian width also decreases as the number of M distributions increases. This is for the same reason as described in the 4.4 where the variance is proportional to $1/\sqrt{n}$ and thus the graph becomes sharper as we add more and more X_i to the distribution.

We also found an implementation method to create a random variable generator based solely upon the CDF of the random variable, ie:- the **inverse transform method**. Here's how it works (refer here):

Let $F_X(x)$ be a *strictly monotonic continuous* CDF of a distribution X, and let F_X^{-1} be it's inverse function (which exists due to the monotonicity of the function). Also, let U be the uniform random variable on $(0, 1)$. Then

$P(F_X^{-1}(U) \leq x) = P(U \leq F_X(x))$ (since F_X is monotonic, $x \leq y \implies F_X(x) \leq F_X(y)$)

$= F_X(x)$.

Thus, for random sampling from the distribution of X, it's enough to take a random uniform sample in $(0, 1)$ and map it by the function $F_X^{-1}(x)$.

Now, for the M-distribution (for which the PDF is $f_M(x) = |x| \cdot \mathbf{1}_{|x|\leq 1}$), we can get the CDF as

$$F_M(x) = \int_{-\infty}^{x} |x| \cdot \mathbf{1}_{|x|\leq 1}\, dx =$$

$$\begin{cases} 0 & x < -1 \\ \int_{-1}^{min(x,1)} |x| \cdot \mathbf{1}_{|x|\leq 1}\, dx & -1 \leq x \end{cases}$$

Simplifying the integral yields

$$\text{CDF}(X) = \begin{cases} 0 & x < -1 \\ \frac{1-x^2}{2} & -1 \leq x < 0 \\ \frac{1+x^2}{2} & 0 \leq x < 1 \\ 1 & 1 \leq x \end{cases}$$

Thus, by simple inversion we get

$$\text{CDF}^{-1}(X) = \begin{cases} -\sqrt{1-2x} & 0 \leq x \leq 0.5 \\ \sqrt{2x-1} & 0.5 \leq x \leq 1 \end{cases}$$

Now, if U is a uniform random variable between 0 and 1,

$$CDF^{-1}(U) = F_X(x)$$

Thus, we obtain the result.

6 Problem 5

6.1 Introduction

The motivation behind this problem was to actually demonstrate through the proper choice (ie:- independent and identically distributed) of random variables that the conclusions of the Law of Large Numbers holds. Thus the grader may also wish to refer the contents of Problem 3 for some more discussion regarding the Law of Large numbers.

Thus, after a brief explanation of the code used to implement the questions asked, we present our results and interpretations for the grader's perusal.

6.2 Code Flow

The code for this problem is not much involved. So, we first write a function named **plot_Unif()**, which generates an array of errors (of size $M = 100$), where each error is the difference between the mean of N uniformly chosen random variables (the choosing of random variables is by the method **np.random.uniform**) with the theoretical mean, ie:- $\frac{1}{2}$.

Using these function we generate error arrays for $N = 5$, 10, 20, 40, 60, 80, 100, 500, 1000, 10000, stack up those arrays, and then use python's box plot utility within matplotlib to generate the box plot.

The exact same procedure as above is used to generate the Gaussian Random Variable case too, this time the choice of normal random variables being made by **np.random.normal**.

6.3 Interpretations and Results

The box and whisker plot shows us the **median of the error distribution** (as opposed to the mean) and it's spread in terms of **percentiles** (as opposed to the standard deviation, though the two can be linked through Chebyshev's inequality), the general trend of predictions by the Law of Large Numbers is apparent from these metrics too.

(Note that by the inequality $|\mu_{error} - median_{error}| \leq \sigma_{error}$, median is mathematically bound to tend to zero since both mean of errors and standard deviations of errors tend to zero), ie:- the median goes to zero, though not monotonically, as expected : **Although the Law of Large numbers guarantees that as $N \longrightarrow \infty$ the errors go to zero, being a stochastic quantity, some fluctuations in the median are only to be expected**. But as N grows larger, and the distribution of errors becomes more and more concentrated around the central tendency as N grows large.

One may also cursorily note that the median (of deviations) in the Gaussian box plot are larger than the corresponding medians of the uniform box plot.

This is again to be expected since the uniform distribution is effectively confined to a fixed region while Gaussian distribution brings a non-zero probability to the whole real line. Hence the Gaussian plot is bound to be more spread out.

Mathematically, the variance of uniform distribution is given as

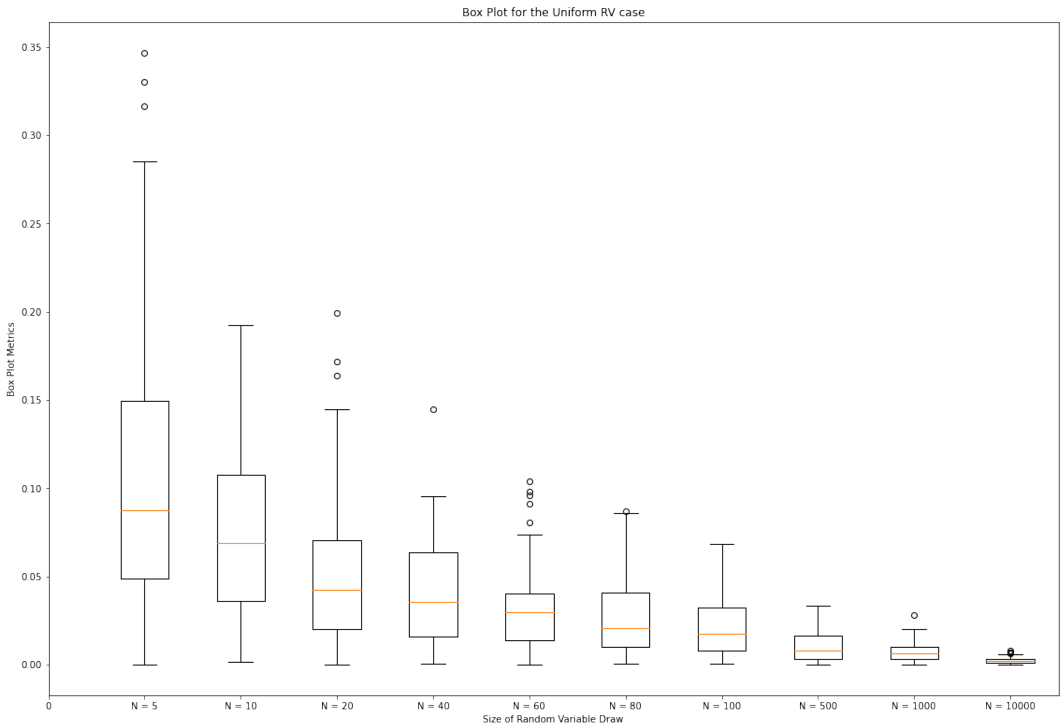
$$Var(X) = (b-a)^2/12$$

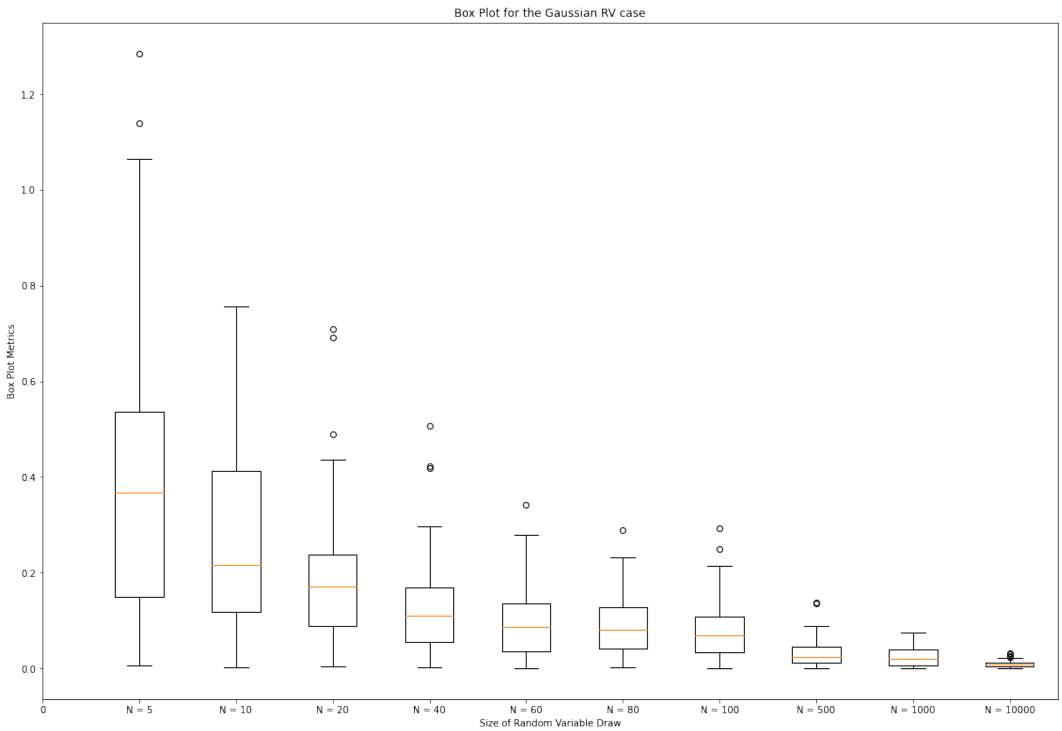
whereas the variance of Gaussian distribution is given as

$$Var(X) = \sigma^2$$

Since, here $\sigma = 1$ and $b - a = 1$ as well, the variance and thus, spread of uniform distribution is less than the Gaussian distribution.

The copies of the plots are attached below for the grader's perusal :


Box Plot for the Uniform RV case
Box Plot Metrics
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
N = 5
N = 10
N = 20
N = 40
N = 60
N = 80
N = 100
N = 500
N = 1000
N = 10000
Size of Random Variable Draw


Box Plot for the Gaussian RV case
Box Plot Metrics
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
N = 5
N = 10
N = 20
N = 40
N = 60
N = 80
N = 100
N = 500
N = 1000
N = 10000
Size of Random Variable Draw

7 References

1. https://numpy.org/doc/stable/reference/generated/numpy.histogram.html
2. https://stackoverflow.com/questions/17779316/un-normalized-gaussian-curve-on-histogram
3. https://www.geeksforgeeks.org/generate-random-numbers-from-the-uniform-distribution-using-numpy/
4. https://www.geeksforgeeks.org/box-plot-in-python-using-matplotlib/
5. https://numpy.org/doc/stable/reference/random/generated/numpy.random.normal.html
6. https://stats.stackexchange.com/questions/3476/how-to-name-the-ticks-in-a-python-matplotlib-boxplot
7. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.rv_continuous.html
8. https://matplotlib.org/stable/tutorials/introductory/pyplot.html
9. https://www.statisticshowto.com/gumbel-distribution/
10. https://numpy.org/doc/stable/reference/random/generated/numpy.random.poisson.html