CS-215 Assignment-1 Report

Abhijit Amrendra Kumar (210050002)
Prerak Contractor (210050124)

dynamight@cse.iitb.ac.in
prerak@cse.iitb.ac.in

Contents

PART

I

Question 1

Formulae

Laplace Distribution

$$P(X = x; \mu, b) = \frac{1}{2b} \cdot e^{\frac{-|x-\mu|}{b}}$$

Gumbel Distribution

$$P(X = x; \mu, \beta) = \frac{1}{\beta} \cdot e^{-(k+e^k)} \text{ with } k = \frac{x-\mu}{\beta}$$

Cauchy Distribution

$$P(X = x; x_0, \gamma) = \frac{\gamma}{\pi(\gamma^2 + (x-x_0)^2)}$$

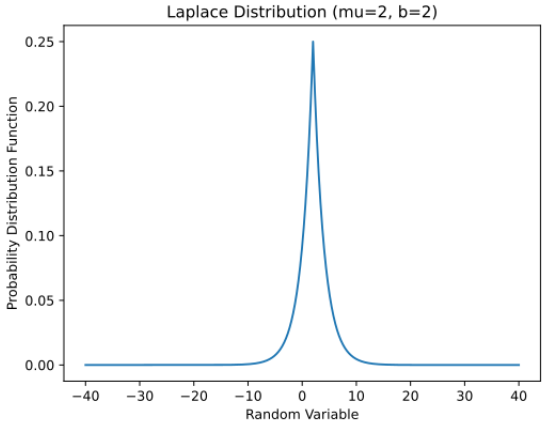
Section 1

Plotting the PDF

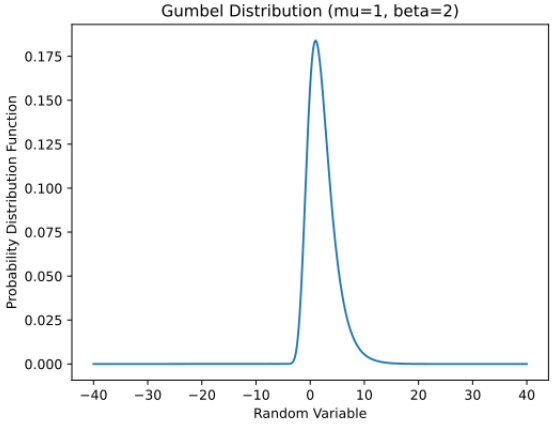
For each function, we first split a unit length on x-axis in `EPSILON=1000` parts, and created a array of these points. For each point, we calculated the PDF value, and stored it in another array. The graph of these arrays was then plotted using in `matplotlib` library of Python.

Subsection 1.1

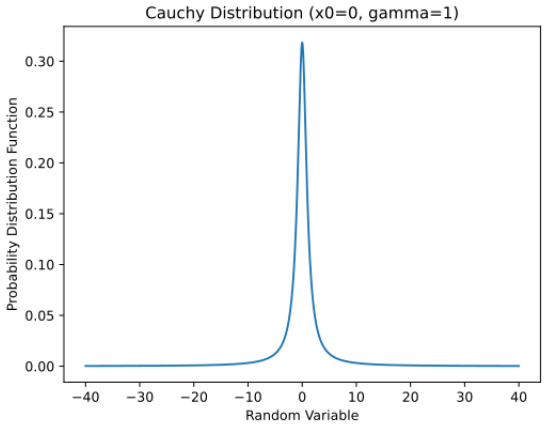
Laplace Distribution



Subsection 1.2
Gumbel Distribution



Subsection 1.3
Cauchy Distribution



Section 2
Plotting the CDF

Subsection 2.1
Riemann Sum Approximation

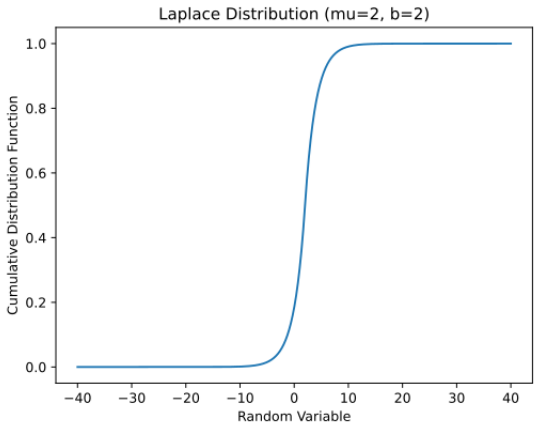
We use the following Riemann Sum approximation to compute integrals.

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n-1} f(x_i)\delta x, \quad \delta x = \frac{b-a}{n} \quad \text{and} \quad x_i = a + \frac{(b-a)i}{n} \quad \text{with} \quad n \to \infty$$

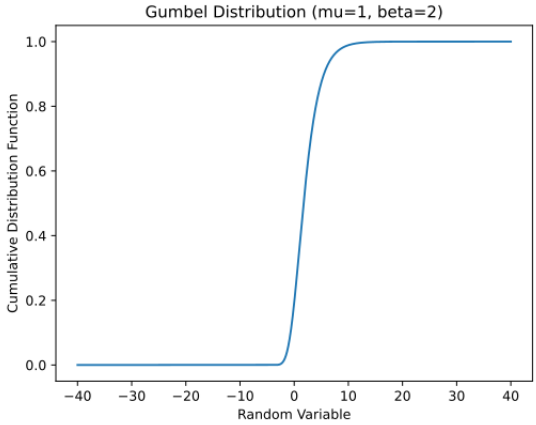
We defined a function which takes a PDF (or any function), a start point, an end point, and the parameters of PDF as input. The function then generates an array of points of x-axis from start to end, dividing each unit length into `EPSILON=1000` parts. The value of PDF is then computed for each point and summed up. The sum is divided by `EPSILON` to get the approximate value of integral.

To increase efficiency, the CDF value, defined as $\mathrm{CDF}(X = k) = \int_{-\infty}^{k} P(X)\mathrm{d}x$ is computed not by evaluating the integral for each k but adding the term $P(X)\delta x$ to the value of CDF at $X = K$ to get value of $CDF(X = k + \delta x)$. The same procedure as PDF is now used to create two arrays and then plot.

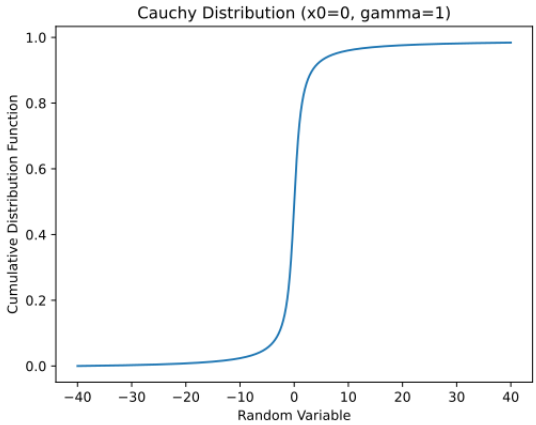
Subsection 2.2
Laplace Distribution



Subsection 2.3
Gumbel Distribution



Subsection 2.4
Cauchy Distribution



Section 3
Calculating Variance

Definition 1

$$\begin{aligned}
\mathrm{Var}(X) &:= \int_{-\infty}^{\infty} (x-\mu)^2 P(x)dx, \text{ where } \mu = \text{mean} \\
&= \int_{-\infty}^{\infty} x^2 P(x)dx + \int_{-\infty}^{\infty} \mu^2 P(x)dx - 2\mu \int_{-\infty}^{\infty} xP(x)dx \\
&= \int_{-\infty}^{\infty} x^2 P(x)dx + \mu^2 - 2\mu^2 \\
&= \int_{-\infty}^{\infty} x^2 P(x)dx - \mu^2
\end{aligned}$$

This is simplified to

$$\mathrm{Var}(X) = \mathbb{E}\left[X^2\right] - (\mathbb{E}[X])^2$$

Subsection 3.1

Mean And RMS of PDF

We defined two functions to calculate the Mean and RMS of PDF. Each function evaluates the riemann sum for the function $xP(x)$ and $x^2P(x)$ from `-LENGTH` to `LENGTH`, where `LENGTH=40`. We used this interval instead of summing up from $-\infty$ to ∞, because further part of the PDF contributes negligibly to the required sum.

The above two functions are used to estimate the variance $\mathrm{Var}(X)$ for each PDF.

Formulae

Laplace Distribution:
Mean:

$$\int_{-\infty}^{\infty} \frac{x}{2b} \cdot e^{\frac{-|x-\mu|}{b}} dx = \mu$$

Variance:

$$\int_{-\infty}^{\infty} \frac{(x-\mu)^2}{2b} \cdot e^{\frac{-|x-\mu|}{b}} dx = 2b^2$$

Gumbel Distribution:
Mean:

$$\int_{-\infty}^{\infty} \frac{x}{\beta} \cdot e^{-\left(k+e^k\right)} dx = \mu$$

Variance:

$$\int_{-\infty}^{\infty} \frac{(x-\mu)^2}{\beta} \cdot e^{-\left(k+e^k\right)} dx = \frac{\pi^2\beta^2}{6}$$

Cauchy Distribution:
Mean:

$$\int_{-\infty}^{\infty} \frac{x\gamma}{\pi(\gamma^2 + (x-x_0)^2)} dx = \text{unbound}$$

Variance:

$$\int_{-\infty}^{\infty} \frac{(x-\mu)^2\gamma}{\pi(\gamma^2 + (x-x_0)^2)} dx = \text{unbound}$$

The error between estimated variance and theoretical variance was found to be of order 10^{-6}.

PART

II

Question 2

Definition 2 **Poisson Distribution**

$$P(X = k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Given that the average number of hits/events per unit time is λ, poisson random variable X describes the probability of k hits/events occuring in a given time.

Section 4

Sum of Poisson Random Variables

The first task of the question was to generate two poisson random variables X and Y with $\lambda_X = 3$ and $\lambda_Y = 4$. A new random variable $Z := X + Y$ was then introduced. We have to then generate values for this new random variable.

This was achieved by creating two arrays consisting of values of random variables X and Y using `numpy.random.poisson(lambda, size_array)`. A new array Z representing values from new random variable was created by summing up the values from X and Y.

Subsection 4.1

Empirical Estimation for PMF P(Z)

`N=1e6` instances of values each were created for X and Y and their sum was used to create instances of Z.

Then, for each of $k = 0, 1, \cdots, 25$, the frequency of k in Z was calculated and divided by N to estimate the probability of $Z = k$.

Subsection 4.2

Theoretical values for PMF P(Z)

Derivation Z take the value k when $X = j$ and $Y = k - j$ for some k in range $[0, 1, \ldots, k]$. Hence the probability of $Z = k$ will be the sum of probabilities for when

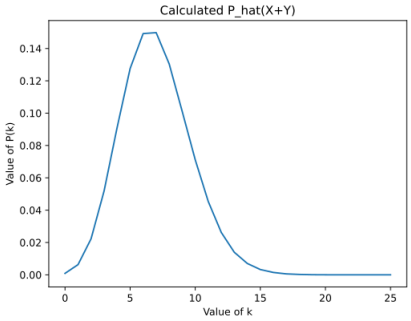
$X = j$ **and** $Y = k - j$ for all possible values of j.

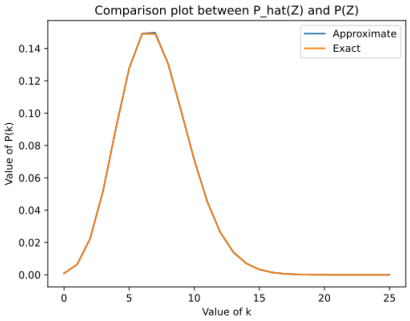
$$
\begin{aligned}
P(Z = k) = \sum_{j=0}^{k} P(X = j; Y = k - j) &= \sum_{j=0}^{k} \frac{e^{-\lambda}.\lambda^{j}}{j!} \cdot \frac{e^{-\mu}.\mu^{k-j}}{(k-j)!} \quad \text{(Since X and Y are independent)} \\
&= \frac{e^{-(\lambda+\mu)}}{k!} \sum_{j=0}^{k} \frac{k!}{j!(k-j)} \lambda^{j} \mu^{k-j} \\
&= \frac{e^{-(\lambda+\mu)}}{k!} \sum_{j=0}^{k} \binom{k}{j} \lambda^{j} \mu^{k-j} \\
&= \frac{e^{-(\lambda+\mu)}(\lambda+\mu)^{k}}{k!} \\
&= P_{poisson}(Z = k; \lambda + \mu)
\end{aligned}
$$

Comparison between Calculated and Actual Values

The estimated and actual values for PDF $P(Z)$, with the error we obtained are:

k	$\hat{P}(Z = k)$	$P(Z = k)$	error = abs$(P(Z = k) - P(\hat{Z} = k))$
0	0.000903	0.0009118819655545162	$8.881965554516195e-06$
1	0.006325	0.006383173758881614	$5.8173758881613705e-05$
2	0.022276	0.022341108156085646	$6.510815608564563e-05$
3	0.051859	0.052129252364199845	0.0002702523641998425
4	0.091072	0.09122619163734973	0.00015419163734972652
5	0.127581	0.12771666829228961	0.00013566829228961463
6	0.149209	0.1490027796743379	0.00020622032566211534
7	0.149799	0.1490027796743379	0.0007962203256620948
8	0.130449	0.13037743221504564	$7.156778495437388e-05$
9	0.101062	0.10140466950059107	0.0003426695005910724
10	0.071183	0.07098326865041375	0.0001997313495862435
11	0.045372	0.045171170959354204	0.00020082904064579882
12	0.026231	0.02634984972628995	0.00011884972628994905
13	0.013956	0.014188380621848434	0.00023238062184843464
14	0.007028	0.007094190310924218	$6.619031092421762e-05$
15	0.003278	0.0033106221450979684	$3.262214509796831e-05$
16	0.001467	0.0014483971884803612	$1.8602811519638756e-05$
17	0.000585	0.0005963988423154428	$1.1398842315442793e-05$
18	0.00024	0.0002319328831226722	$8.067116877327805e-06$
19	$7.4e-05$	$8.544895693993188e-05$	$1.144895693993188e-05$
20	$2.9e-05$	$2.990713492897615e-05$	$9.071349289761511e-07$
21	$1.3e-05$	$9.969044976325385e-06$	$3.0309550236746138e-06$
22	$6e-06$	$3.1719688561035315e-06$	$2.8280311438964686e-06$
23	$3e-06$	$9.653818257706398e-07$	$2.0346181742293605e-06$
24	0.0	$2.8156969918310334e-07$	$2.8156969918310334e-07$
25	0.0	$7.883951577126893e-08$	$7.883951577126893e-08$





Section 5

Poisson Thinning Process

The second task was to simulate a Poisson thinning process with $p = 0.8$.

This was achieved by first creating a array of values for the random variable Y which represents total hits in one unit time. Then each value N was passed to numpy's `numpy.random.binomial(N, p)` to get successful hits (each had a probability p of succeeding) from total hits. This array represents the thinned Poisson random variable.

Subsection 5.1

Empirical Estimation for PDF P(Z)

A total of `N=1e5` values of Poisson distribution were generated. The above mentioned procedure was used to generate the random variable Z. The frequency for each of k in $[0, 1, \ldots, 25]$ was calculated and divide by N to get an estimate of $P(Z = k)$.

Subsection 5.2

Theoretical value of P(Z=k)

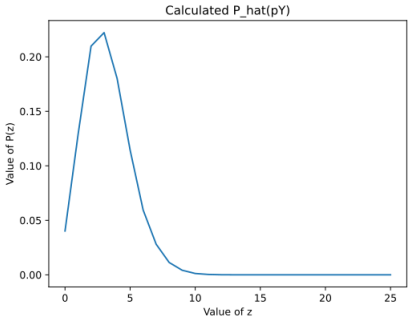
Derivation Number of successful hits is k when a total of $j \geq k$ hits occurred and k of them became successful. Hence probability $P(Z = k)$ will be the sum of probabilities for all values of k, with a binomial distribution for a total of k successful events out of n.

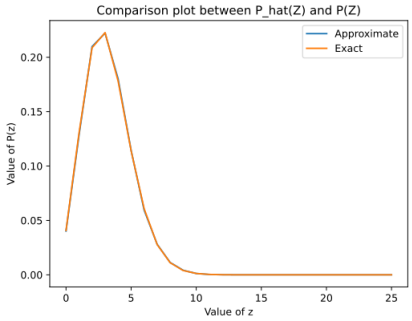
$$
\begin{aligned}
P(Y=k) = \sum_{j=k}^{\infty} P_{\text{Binomial}}(X=j|Y=k) P_{\text{Poisson}}(X=j) &= \sum_{j=k}^{\infty} \frac{e^{-\lambda}.\lambda^j}{j!} \cdot \binom{j}{k} p^k (1-p)^{j-k} \\
&= e^{-\lambda} \sum_{j=k}^{\infty} \frac{\lambda^j}{j!} \cdot \frac{j!}{k!(j-k)!} p^k (1-p)^{j-k} \\
&= \frac{e^{-\lambda}(\lambda p)^k}{k!} \sum_{j=k}^{\infty} \frac{(\lambda(1-p))j-k}{(j-k)!} \\
&= \frac{e^{-\lambda}(\lambda p)^k}{k!} \cdot e^{\lambda(1-p)} \\
&= \frac{e^{-\lambda p}(\lambda p)^k}{k!} \\
&= P_{poisson}(Y=k;\lambda p)
\end{aligned}
$$

Comparison of Estimated and Theoretical Values

The estimated and theoretical values of $P(Z)$ and the error are:

k	$\hat{P}(Z=k)$	$P(Z=k)$	error = abs$(P(Z=k) - P(\hat{Z}=k))$
0	0.04021	0.04076220397836621	0.0005522039783662086
1	0.12849	0.1304390527307719	0.0019490527307718941
2	0.2098	0.20870248436923505	0.0010975156307649336
3	0.22219	0.22261598332718405	0.0004259833271840485
4	0.18014	0.17809278666174724	0.0020472133382527513
5	0.11443	0.11397938346351824	0.0004506165364817627
6	0.05946	0.0607890045138764	0.0012900045138764014
7	0.02808	0.027789259206343498	0.00029074079365650277
8	0.01131	0.011115703682537401	0.00019429631746259966
9	0.00419	0.00395225019823552	0.0002377498017644801
10	0.00124	0.0012647200634353665	$2.4720063435366545e-05$
11	0.00035	0.0003679185639084703	$1.791856390847028e-05$
12	$8e-05$	$9.811161704225874e-05$	$1.8111617042258736e-05$
13	$3e-05$	$2.415055188732523e-05$	$5.849448112674772e-06$
14	0.0	$5.520126145674339e-06$	$5.520126145674339e-06$
15	0.0	$1.1776269110771923e-06$	$1.1776269110771923e-06$
16	0.0	$2.3552538221543848e-07$	$2.3552538221543848e-07$
17	0.0	$4.433418959349431e-08$	$4.433418959349431e-08$
18	0.0	$7.8816337055101e-09$	$7.8816337055101e-09$
19	0.0	$1.327433045138543e-09$	$1.327433045138543e-09$
20	0.0	$2.1238928722216692e-10$	$2.1238928722216692e-10$
21	0.0	$3.2364081862425436e-11$	$3.2364081862425436e-11$
22	0.0	$4.707502816352791e-12$	$4.707502816352791e-12$
23	0.0	$6.549569135795188e-13$	$6.549569135795188e-13$
24	0.0	$8.732758847726919e-14$	$8.732758847726919e-14$
25	0.0	$1.1177931325090456e-14$	$1.1177931325090456e-14$


Calculated P_hat(pY)
Value of P(z)
Value of z
0.00
0.05
0.10
0.15
0.20
0
5
10
15
20
25


Comparison plot between P_hat(Z) and P(Z)
Approximate
Exact
Value of P(z)
Value of z
0.00
0.05
0.10
0.15
0.20
0
5
10
15
20
25

PART

III

Question 3

Section 6

Random Walk

Definition 3 A random walk is a random process that describes a path that consists of a series of random steps on a line, each having equal probability.

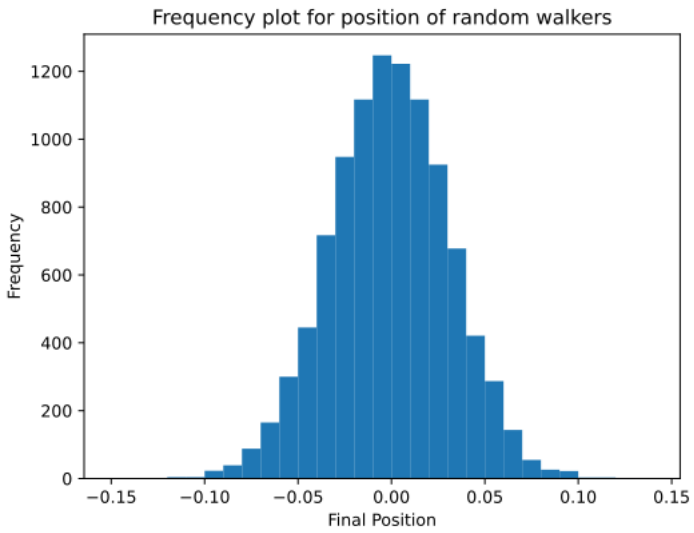
Subsection 6.1

Plotting the histogram

First, we simulate a large number `N=1e4` of random walkers, each taking either a right step or a left step in each iteration.

To do this, we take a random number from $[0, 1)$ using python's `numpy.random.random()`, and if it is greater than 0.5, we map it to +step, else we map it to -step. We repeat this for `iters=1e3` iterations of steps, and further repeat the random walk for N number of random walkers.

For each walker, we sum the steps taken to get its final position. A frequency histogram with 30 bins was then plotted to show the distribution of final positions of all random walkers.



Subsection 6.2
Plotting paths of random walkers

For each random walker, we first created an array of positions at each time by cumulatively adding the steps before it (using the `numpy.cumsum(array)` function). The plot was then created for each walk with position on x-axis and time on y-axis.

Section 7
Law of Large Numbers

Theorem 1 Define a random variable $\hat{M} := (\sum_i X_i)/N$ for N independent random variables X_i. The law of large number states that:

$$\forall \epsilon > 0, \lim_{n\to\infty} P(|\hat{M} - \mu| \geq \epsilon) = 0$$

where μ is the mean of each of the random variable X_i.

Proof Let the mean of X_i be μ and variance v. Then, by linearity of expectation,

$$\mathbb{E}\left[\hat{M}\right] = \frac{\sum_i \mu}{N} = \mu$$

And variance of $\hat{M}$,

$$\begin{aligned}\mathrm{Var}\left(\hat{M}\right) &= \sum_i \left(\mathrm{Var}\left(\frac{X_i}{N}\right)\right) \\ &= \sum_i \frac{v}{N^2} \\ &= \frac{v}{N}\end{aligned}$$

Using Chebyshev's inequality,

$$\begin{aligned}P(|\hat{M} - \mu| \geq \epsilon) &\leq \mathrm{Var}(\hat{M})/\epsilon^2 \\ &= v/(N\epsilon^2)\end{aligned}$$

Thus, $\lim_{N\to\infty} P(|\hat{M} - \mu| \geq \epsilon) = 0$ □

This shows that the value of $\hat{M}$ converges to the true mean μ for $N \to \infty$ as the probability of occurrence of value of $\hat{M}$ outside an ϵ neighbour of μ tends to zero.

Theorem 2 Define a quantity $\hat{V} := \left(\sum_i \left(X_i - \hat{M}\right)^2\right)/N$, with $\hat{M}$ defined as before. Then,

$$\boxed{\mathbb{E}[\hat{V}] \to \mathrm{Var}(X) \quad \text{as N tends to infinity}}$$

Proof Simplifying $\hat{V}$:

$$\begin{aligned}\hat{V} &= \frac{\sum_i \left(X_i^2 + \hat{M}^2 - 2X_i\hat{M}\right)}{N} \\ &= \sum_i \frac{X_i^2}{N} + \hat{M}^2 - \frac{2\hat{M}}{N}\sum_i X_i \\ &= \sum_i \frac{X_i^2}{N} + \hat{M}^2 - 2\hat{M}^2 \\ &= \sum_i \frac{X_i^2}{N} - \hat{M}^2\end{aligned}$$

Now, calculating the expected value,

$$\begin{aligned}\mathbb{E}\left[\hat{V}\right] &= \mathbb{E}\left[\sum_i \frac{X_i^2}{N} - \hat{M}^2\right] \\ &= \mathbb{E}\left[\sum_i \frac{X_i^2}{N}\right] - \mathbb{E}\left[\hat{M}^2\right] \\ &= \mathbb{E}[X^2] - \mathbb{E}\left[\hat{M}^2\right]\end{aligned}$$

Using law of large numbers, $\hat{M} \to \mu$ as $N \to \infty$. Hence, expected value $\mathbb{E}[\hat{M}^2] = \mu^2$ as $N \to \infty$. Hence,

$$\mathbb{E}\Big[\hat{V}\Big] \to \mathbb{E}\Big[X^2\Big] - \mu^2 = \mathrm{Var}(X)$$

□

Subsection 7.1

Comparing calculated and true values of mean and variance

Derivation

Let the random walker take k steps to right and $N-k$ steps to left. Final position is $(k-(N-k))\delta x = (2k-N)\delta x$. Modelling this as binomial distribution with $p = 0.5$, we have

$$P((2k-N)\delta x) = \binom{N}{k}\left(\frac{1}{2}\right)^N$$

$$\implies P(x) = \binom{N}{\frac{x+N\delta x}{2\delta x}}\left(\frac{1}{2}\right)^N$$

Where $P(x)$ is the probability of final position being x.
Mean of position is

$$\begin{aligned}
\mathbb{E}[X] = \sum_i x_i P(x_i) &= \delta x \sum_{k=0}^{N} \frac{(2k-N)\binom{N}{k}}{2^N} \\
&= \delta x \sum_{k=0}^{N} \frac{(2(N-k)-N)\binom{N}{k}}{2^N}, \text{ substituting k with N-k} \\
&= \delta x \sum_{k=0}^{N} \frac{(N-2k)\binom{N}{k}}{2^N} \\
&= 0
\end{aligned}$$

Variance of position is,

$$\begin{aligned}
\mathrm{Var}(X) = \sum x^2 P(x) &= \delta x^2 \sum_{k=0}^{N} \frac{(2k-N)^2\binom{N}{k}}{2^N} \\
&= \delta x^2 \sum_{k=0}^{N} \frac{(4k^2+N^2-4Nk)\binom{N}{k}}{2^N} \\
&= \delta x^2 \left(4\sum_{k=0}^{N} \frac{k^2\binom{N}{k}}{2^N} + N^2 - 4N\sum_{k=0}^{N} \frac{k\binom{N}{k}}{2^N}\right) \\
&= \delta x^2 \Big(N(N+1) + N^2 - 2N^2\Big) \\
&= N\delta x^2
\end{aligned}$$

The mean and variance was then calculated for the data of final positions generated on previous step using numpy's `numpy.mean(array)` and `numpy.var(array)`. The comparison of theoretical and empirical values is shown in the table below:

	Calculated	True	Error
Mean	$9.4200000000000008e-05$	0	$9.4200000000000008e-05$
Variance	0.0010068147263600001	0.001	$6.8147263600000094e-06$

This supports law of large numbers as the the calculated mean is converging to theoretical mean for large N.

PART

IV

Question 4

Section 8

Random variable with PDF = |x|

Subsection 8.1

Generating independent draws and plotting histogram

To generate draws for the given distribution, we first calculated a number t as the max of two randomly chosen numbers in $[0, 1)$. Then, we chose another random number in $[0, 1)$ (say q), using which we assigned the sign to t ($t > 0$ if $q < 0.5$, else $t < 0$. The final value of t is used as a draw, which follows the given M-shaped distribution.

Proof Consider three random variables X_1, X_2 (uniform distribution in [0,1]) and Y_1 (Bernoulli distribution for {-1,1} with $p = 0.5$).

Define the random variable $Z := \max(X_1, X_2)Y$.
Since Z is independent of $\max(X_1, X_2)$, Z will take positive and negative values with equal probabilities. Hence, all we need to prove is that PDF of $Z' := \max(X_1, X_2)$ is proportional to k for $Z' = k$.

The CDF for Z', $P(Z' \leq k)$ will be

$$P(Z' \leq k) = P(X_1 \leq k) \cdot P(X_2 \leq k) = k \cdot k$$

as CDF of uniform distribution is $P(x \leq k) = k$.

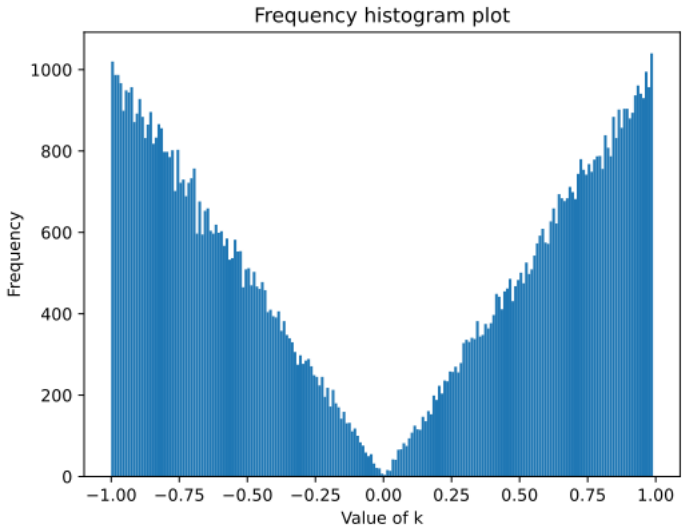
Hence, PDF for Z' will be

$$P(Z' = k) = \frac{\partial P(Z' \leq k)}{\partial k} = 2k$$

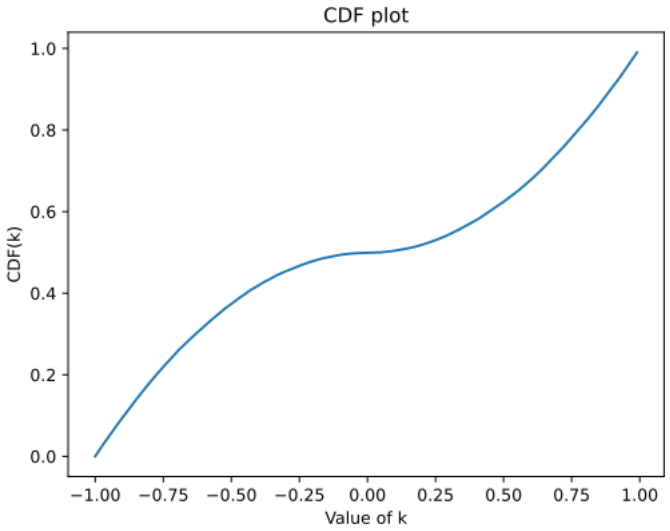
□

Subsection 8.2
Plotting the distribution

`M=1e5` draws were simulated for the distribution and stored in array. The frequency histogram was plotted with `bins=200`.



The CDF was plotted by first dividing a unit length on x-axis in 50 parts, and storing the points in an array. For each point, the number of draws with values less than or equal to the value of point is found and stored in another array. The plot of these two arrays is given below.

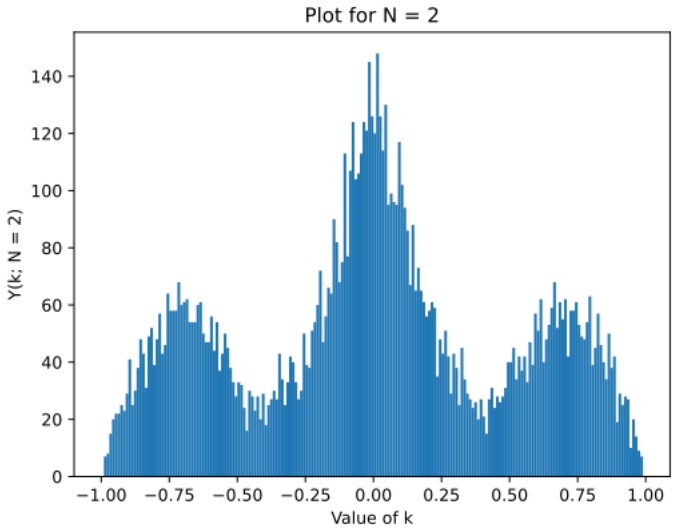


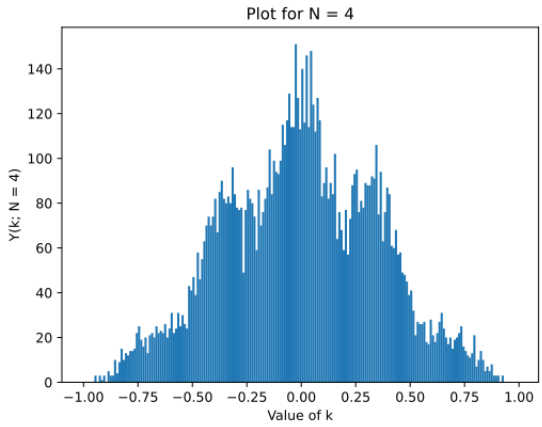
Section 9

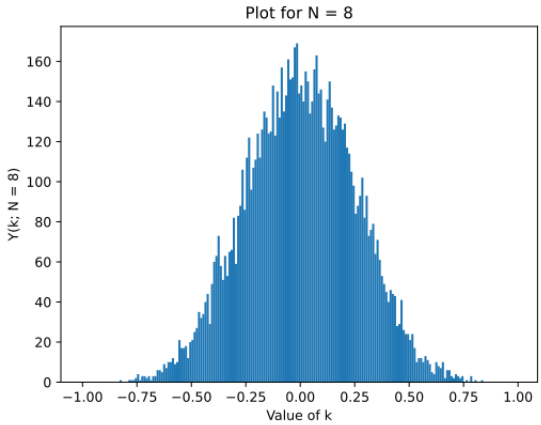
Random Variable representing the mean of N Random Variables

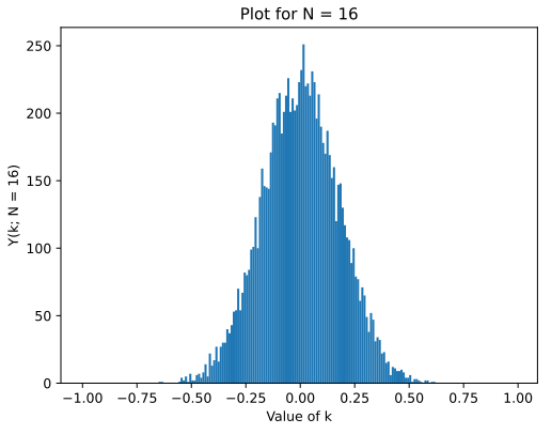
A function was defined to return the mean of k (given as input) randomly generated draws from the above mentioned distribution. Let the random variable to model this be defined as $Y_N := (\sum_{i=1}^{n} X_i)/N$.

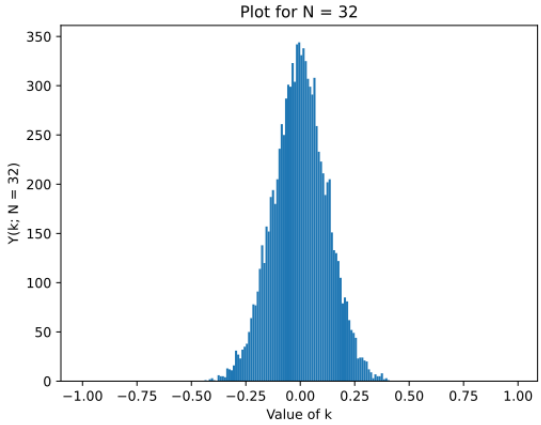
`1e4` draws of random variables were stored for different values of `N in [2, 4, 8, 16, 32, 64]`. The frequency histogram for each N was plotted with `bins=200`:

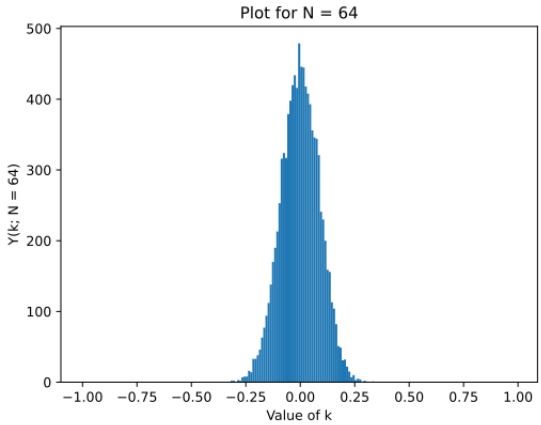





Plot for N = 8
Y(k; N = 8)
Value of k
0
20
40
60
80
100
120
140
160
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

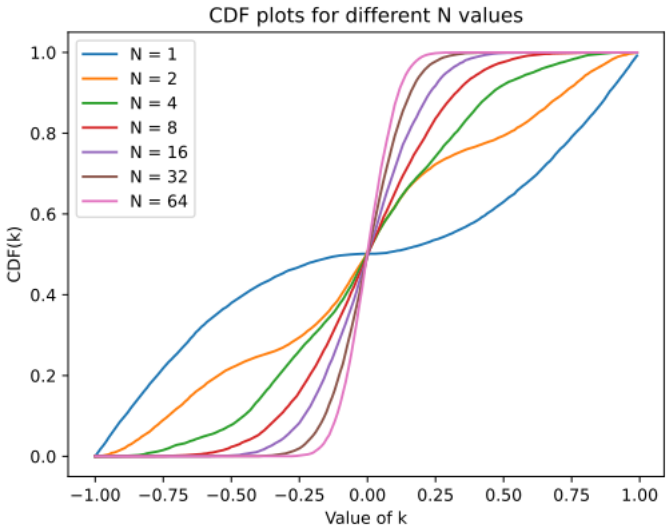

Plot for N = 16
Y(k; N = 16)
Value of k
0
50
100
150
200
250
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


Plot for N = 32
Y(k; N = 32)
Value of k
0
50
100
150
200
250
300
350
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00



We can see that the value of random variable converges to a Gaussian Distribution around the mean of the random variable X. Also, as $N \to \infty$, the value of random variable converges to the mean of X itself.

For each array, the CDF was computed by first dividing a unit length into 100 parts, and the points are stored in an array. For each point, the number of values in array with value less than or equal to the value of point is calculated, and stored in another array. The resultant CDF plots for different values of N is shown below:



PART

V

Question 5

Section 10

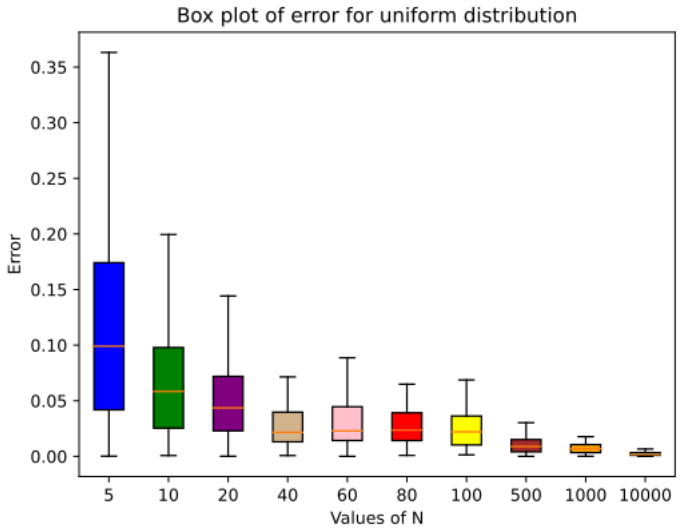
Generating data set for uniform distribution

For each value of `N in [5, 10, 20, 40, 60, 80, 100, 500, 1e3 , 1e4]` as size of data set, the following experiment is performed `M=100` times.

First, an array of size N is generated using uniform distribution.

The mean of these N numbers is then calculated and its absolute error from the true value $\mu = 0.5$ is calculated.

The errors in these M experiments are plotted as a box plot for different values of N. The result obtained is:



Section 11

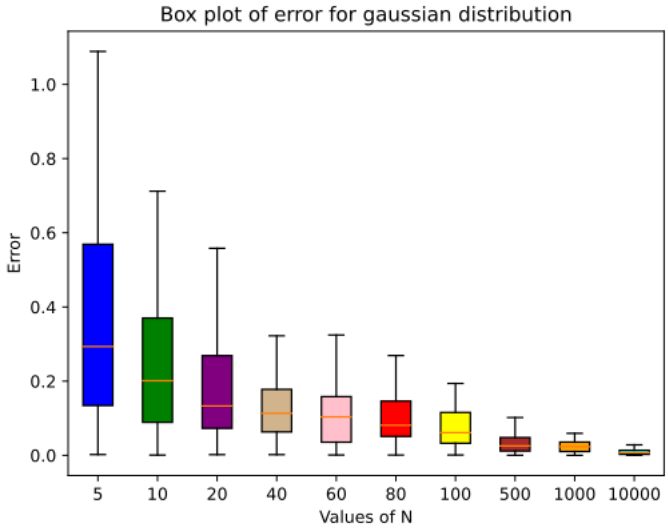
Generating dataset for Gaussian distribution

For each value of `N in [5, 10, 20, 40, 60, 80, 100, 500, 1e3 , 1e4]` as size of data set, the following experiment is performed `M=100` times.

First, an array of size N is generated using Gaussian distribution.

The mean of these N numbers is then calculated and its absolute error from the true value $\mu = 0$ is calculated.

The errors in these M experiments are plotted as a box plot for different values of N. The result obtained is:



Section 12

Interpretation of Result

We can see that as the size of dataset increases, the mean of the dataset approaches the true mean of the distribution, and the error between these two approaches zero. This is in agreement with the **Law of Large Numbers**.