CS215 Assignment 1
Random Variables & Expectation

Josyula Venkata Aditya - 210050075
Kartik Sreekumar Nair - 210050083

October 1, 2022

Preface

This assignment is done in a group. Group members:

1. Josyula Venkata Aditya - 210050075
2. Kartik Sreekumar Nair - 210050083

The submission folder contains four folders(in addition to this file and the problem statement file, out of which data is an empty folder):

1. **code** - contains 5 MATLAB LiveScript(`.mlx`) files, each of which corresponds to one of the 5 problems.
2. **results/fig** - contains 5 folders; each folder corresponds to a problem and contains the plots of that problem. This is the main folder of plots, please perform the evaluations of plots using this folder. The `plots/eps` folder contains `.eps` files which are used in the report, since it is easier to attach `.eps` files than `.fig` files
3. **results/eps** - contains all the plots, but in `.eps` format. For evaluation purposes, please check the plots in `plots/fig` folder and **NOT** this folder.
4. **report** - contains the reports of the 5 problems

*__Note:__ We have seeded the random number generator in the first section in every `.mlx` file. So, please select the first section and "Run to End" to replicate the results.

Problem 1

For each of the following distributions, do:

1. Plot the probability density function (PDF) based on the analytical expression. The PDF must appear smooth enough and without apparent signs of discretization.
2. Plot the cumulative distribution function (CDF) using Riemann-sum approximation. The CDF must appear smooth enough and without apparent signs of discretization.
3. Use Riemann-sum approximations to compute the approximate variance (if finite) within a tolerance of 0.01 of its true value known analytically.

Approach

(i) The respective analytical function was applied to the entire range at once to obtain the PDF. We took a step size of `int_size = 0.1` for the input values.

(ii) With Riemann-sum approximation,

$$P(X \leq x) = \int_{-\infty}^{x} p(t)dt \sim \sum_{i=-a}^{x} p(i) \cdot h \tag{1.1}$$

Where h is the rectangle width, and a an appropriately chosen lower bound.

For getting the cumulative sum for each x together we used the `cumsum()` function. Then multiplied the entire array with `int_size` ($= h$) to get the CDF function.

(iii) For the mean and variance, the Riemann-sum approximations were

$$\text{Mean} = E[X] = \int_{-\infty}^{\infty} x \cdot p(x)dx \sim \sum_{i=-a}^{b} i \cdot p(i)h \tag{1.2}$$

$$E[X^2] = \int_{-\infty}^{\infty} x^2 \cdot p(x)dx \sim \sum_{i=-a}^{b} i^2 \cdot p(i)h \tag{1.3}$$

$$\text{Var}(X) = E[X^2] - E[X]^2 \tag{1.4}$$

With h again taken as `int_size` , we computed `x.*pdf` and `x.*x.*pdf`, took their sums, and multiplied with `int_size` to get the respective expectations.

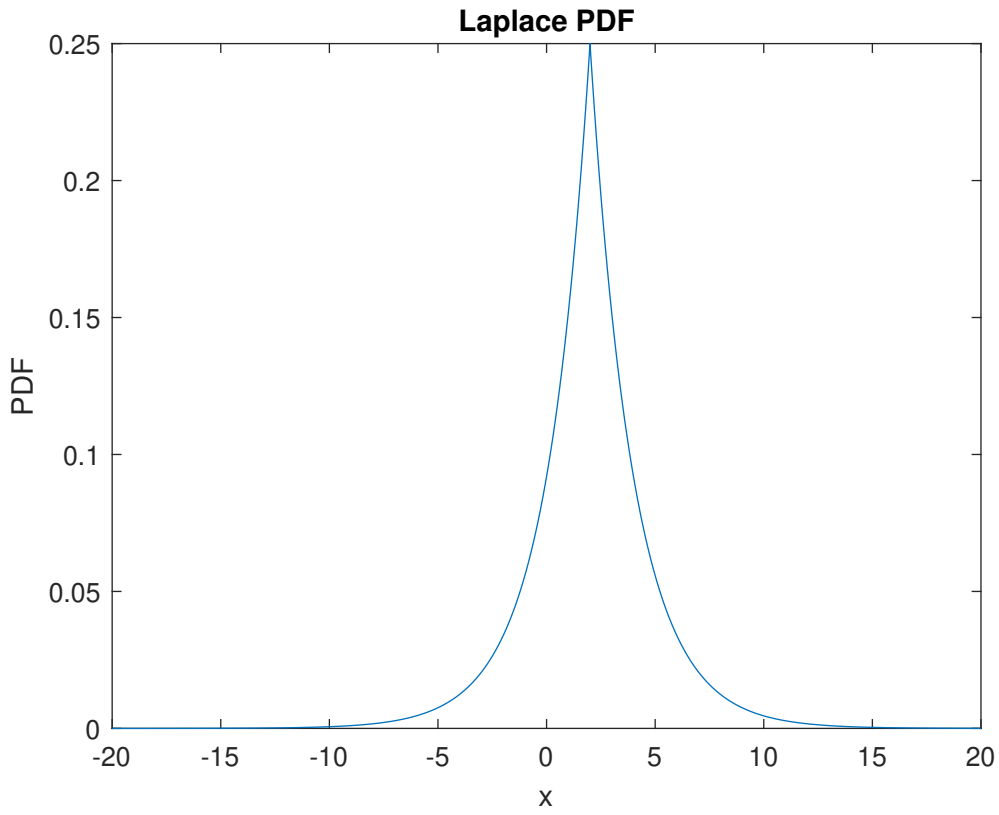
Refer `problem1.mlx` for the code and data.

1.1 Laplace PDF

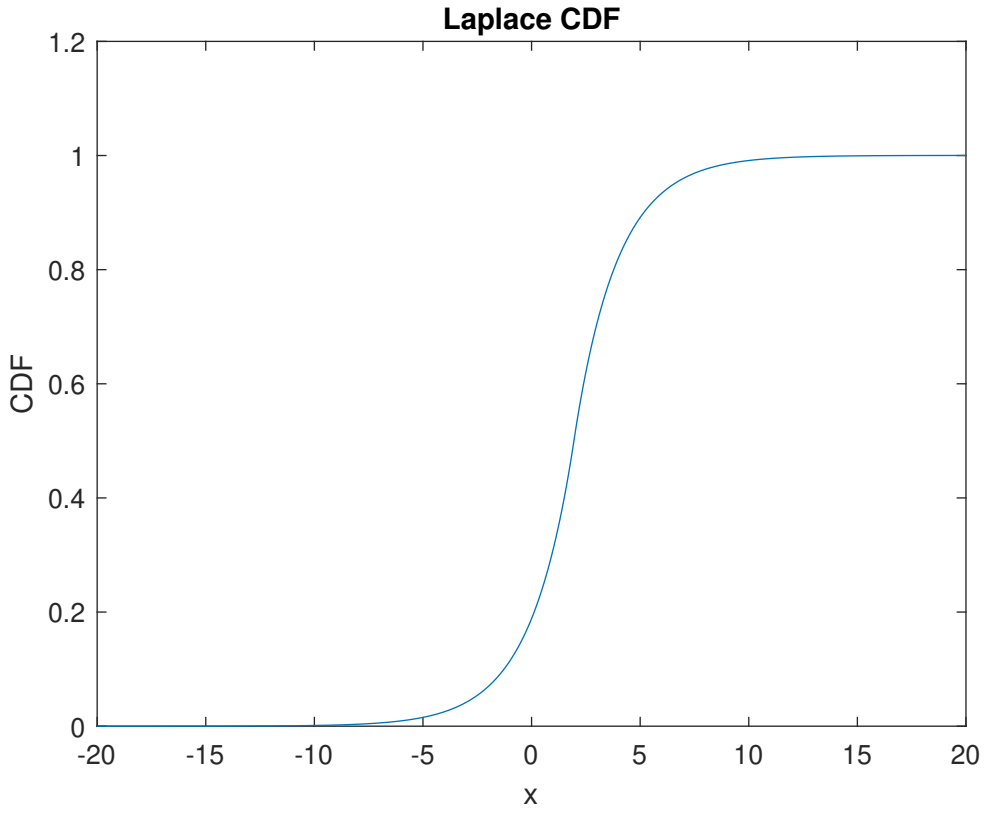
with location parameter $\mu := 2$ and scale parameter $b := 2$.
The analytical expression for the Laplace PDF is

$$P_X(x) = \frac{1}{2b} e^{-|x-\mu|} \tag{1.5}$$

For the estimation we take discrete points at intervals of `int_size = 0.1` We got the following plot:



For the CDF we used the same `int_size` as the width of the rectangles for Riemann summing. The following plot is the calculated CDF



Analytically, the variance would be (location parameter is inconsequential, so consider $\mu = 0$)

$$\begin{aligned} \text{Var}(X) &= E[X^2] - E[X]^2 && (1.6) \\ &= \int_{-\infty}^{\infty} x^2 \cdot \frac{1}{2b} e^{-|x|/b} dx - \left(\int_{-\infty}^{\infty} x \cdot \frac{1}{2b} e^{-|x|/b} dx \right)^2 && (1.7) \\ &= 2b^2 && (1.8) \end{aligned}$$

For our given parameters, we get: 8
The value we obtained empirically was: 7.9992
Relative Error: 0.0001

1.2 Gumbel PDF

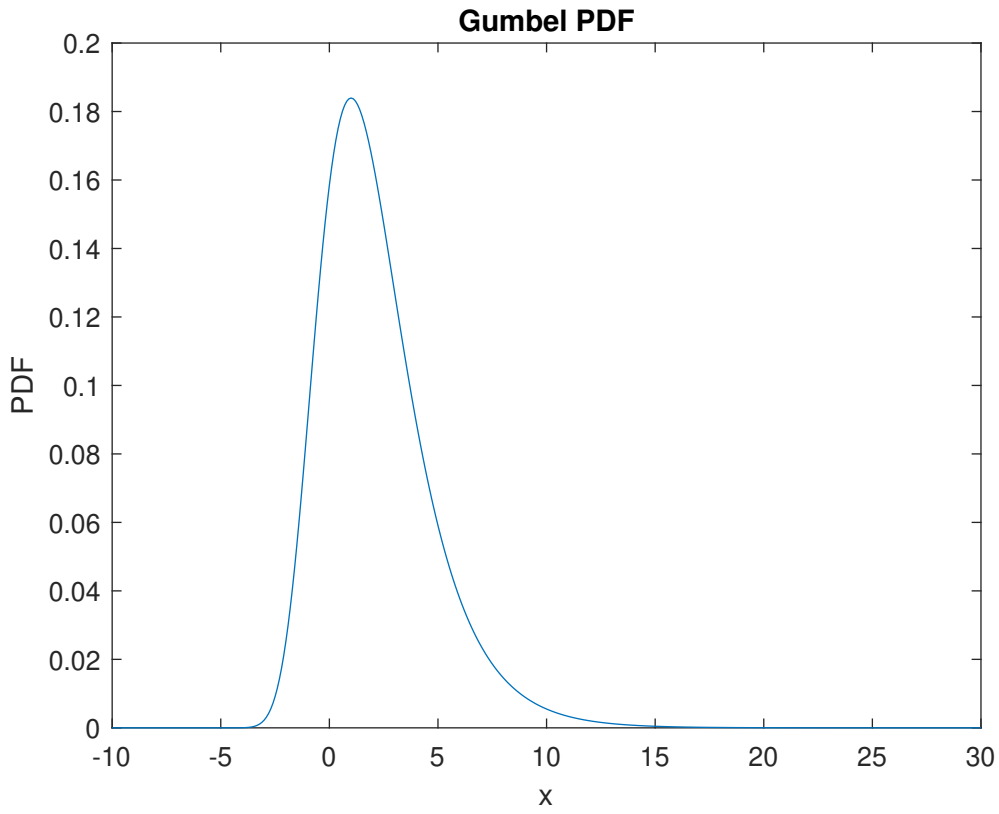
with location parameter $\mu := 1$ and scale parameter $\beta := 2$.
The analytical expression for the Gumbel PDF is

$$P_X(x) = e^{-\left(\tilde{x}+e^{-\tilde{x}}\right)} \tag{1.9}$$

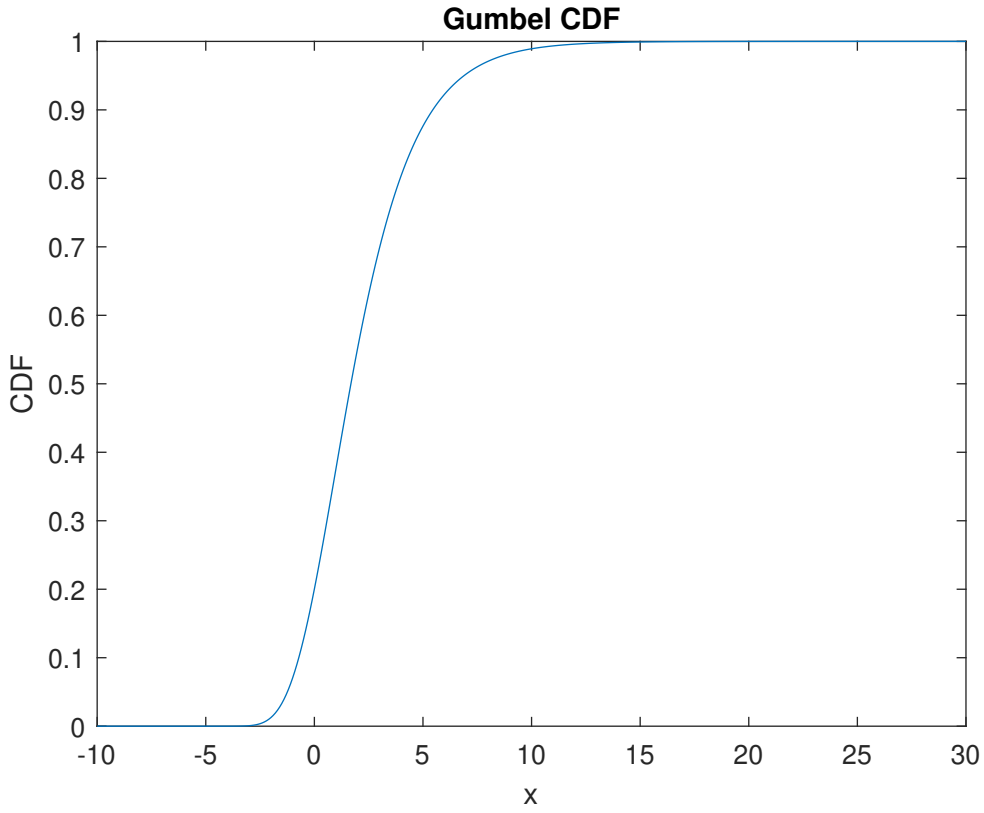
where

$$\tilde{x} = \frac{x-\mu}{\beta} \tag{1.10}$$

For the estimation we take discrete points at intervals of `int_size = 0.1` again. We got the following plot:



For the CDF we use the same `int_size` as the width of the rectangles for Riemann summing. The following plot



Analytically variance would be

$$\text{Var}(X) = E[X^2] - E[X]^2 \tag{1.11}$$

$$= \int_{-\infty}^{\infty} x^2 \cdot e^{-\left(\frac{x-\mu}{\beta} + e^{\frac{x-\mu}{\beta}}\right)} dx - \left(\int_{-\infty}^{\infty} x \cdot e^{-\left(\frac{x-\mu}{\beta} + e^{-\frac{x-\mu}{\beta}}\right)} dx\right)^2 \tag{1.12}$$

$$= \frac{\pi^2}{6}\beta^2 \tag{1.13}$$

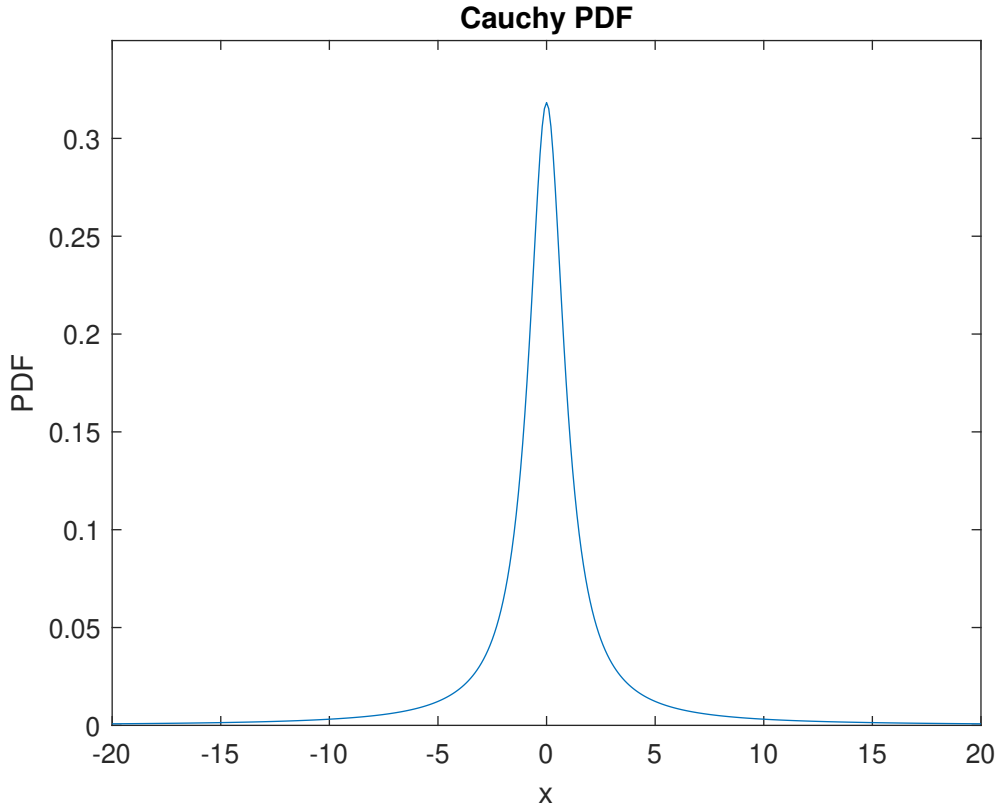
For our given parameters, we get: 6.5797
The value we obtained empirically was: 6.5797

1.3 Cauchy PDF

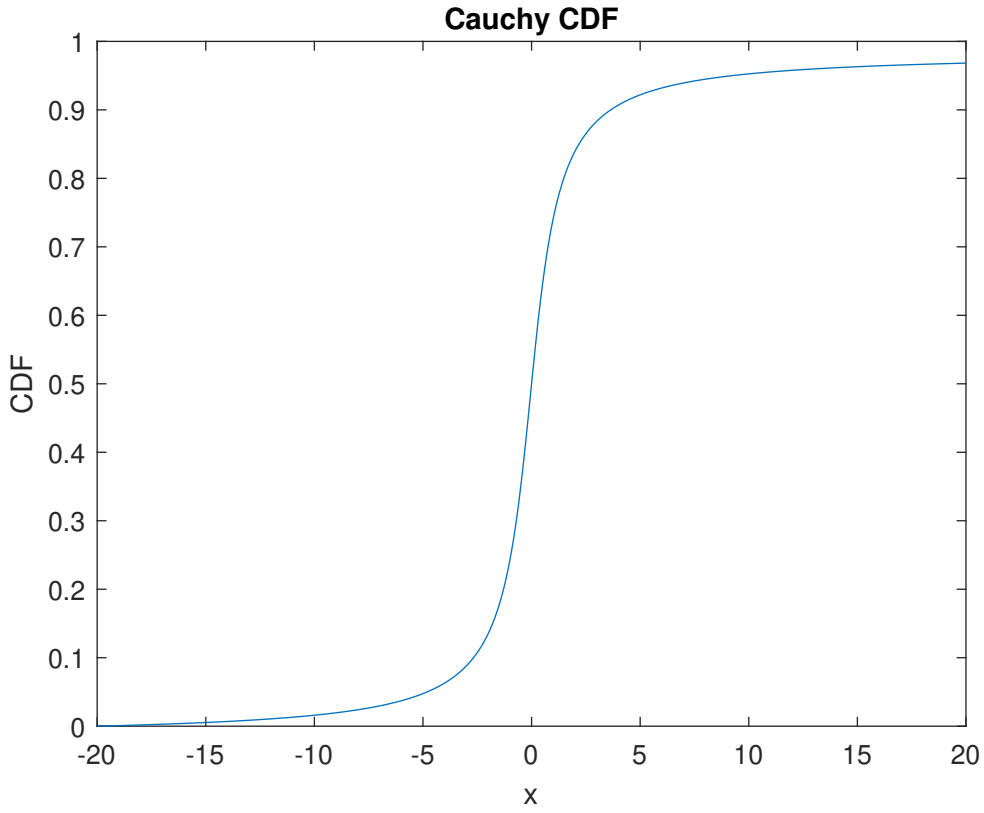
with location parameter $x_0 := 0$ and scale parameter $\gamma := 1$.
The analytical expression for the Laplace PDF is

$$P_X(x) = \frac{1}{\pi\gamma\left(1 + \frac{x-x_0}{\gamma}\right)^2} \tag{1.14}$$

For the estimation we take discrete points at intervals of the same `int_size = 0.1` We got the following plot:



For the CDF we use the same `int_size` as the width of the rectangles for Riemann summing. The following plot



Analytically variance would be

$$\mathrm{Var}(X) = E[X^2] - E[X]^2 \tag{1.15}$$

$$= \int_{-\infty}^{\infty} x^2 \cdot \frac{1}{\pi\gamma\left(1+\frac{x-x_0}{\gamma}\right)^2} dx - \left(\int_{-\infty}^{\infty} x \cdot \frac{1}{\pi\gamma\left(1+\frac{x-x_0}{\gamma}\right)^2} dx\right)^2 \tag{1.16}$$

$$\longrightarrow \infty. \tag{1.17}$$

The above integral expression is a diverging quantity, hence the variance is not finite. This is also confirmed by our code.

Problem 2

Consider two independent Poisson random variables X and Y , with parameters $\lambda_X := 3$ and $\lambda_Y := 4$. The PMF for the poisson distribution is given as

$$p_\lambda(X = k) = e^{-\lambda}\frac{\lambda^k}{k!} \tag{2.1}$$

- Define a random variable $Z := X + Y$, having a probability mass function (PMF) $P(Z)$.

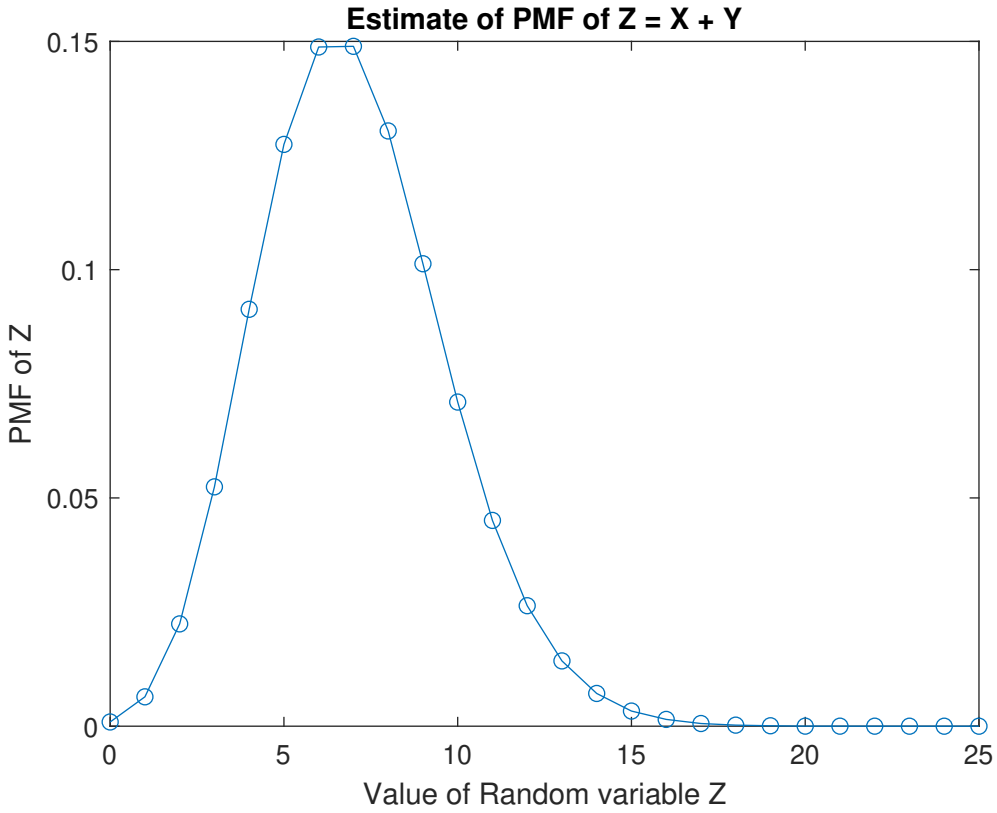
  1. Empirically obtain an estimate $\hat{P}(Z)$ of the PMF $P(Z)$, by drawing $N = 10^6$ instances (sample points) of X and Y both. You may use the poissrnd$(\cdot)$ function in Matlab. Report the values of $\hat{P}(Z = k)$ for $k = 0, 1, 2, \cdots, 25$.

     We first generated the N instances as a size N array of `poissrnd(`λ_x`)+poissrnd(`λ_y`)`. To count the frequencies of these values we simply used the `hist()` function. Divide by N to get the empirical probability.

     Values generated are also reported in `problem2.mlx` too.

```
P_emp_1 = 1x26
  0 to  4
    0.0009    0.0064    0.0224    0.0524    0.0913
  5 to  9
    0.1274    0.1488    0.1489    0.1304    0.1013
 10 to 14
    0.0710    0.0451    0.0264    0.0143    0.0072
 15 to 19
    0.0033    0.0015    0.0006    0.0002    0.0001
 20 to 24
    0.0000    0.0000    0.0000    0         0
 25
    0
```

     The following plot was generated from the data–



2. What will the PMF $P(Z)$ be theoretically/analytically ?
   The poisson distribution is

$$P_\lambda(a) = \frac{\lambda^a}{a!} e^{-\lambda} \tag{2.2}$$

   $P(Z = z)$ is

$$P(Z = z) = \sum_{x=0}^{z} P(X = x) \cdot P(Y = z - x) \tag{2.3}$$

$$= \sum_{x=0}^{z} \frac{\lambda_x^x}{x!} \frac{\lambda_Y^{z-x}}{(z-x)!} e^{-\lambda_X - \lambda_Y} \tag{2.4}$$

$$= \sum_{x=0}^{z} \frac{(\lambda_X/\lambda_Y)^x \cdot {}^zC_x}{z!} \lambda_y^z e^{-\lambda_X - \lambda_Y} \tag{2.5}$$

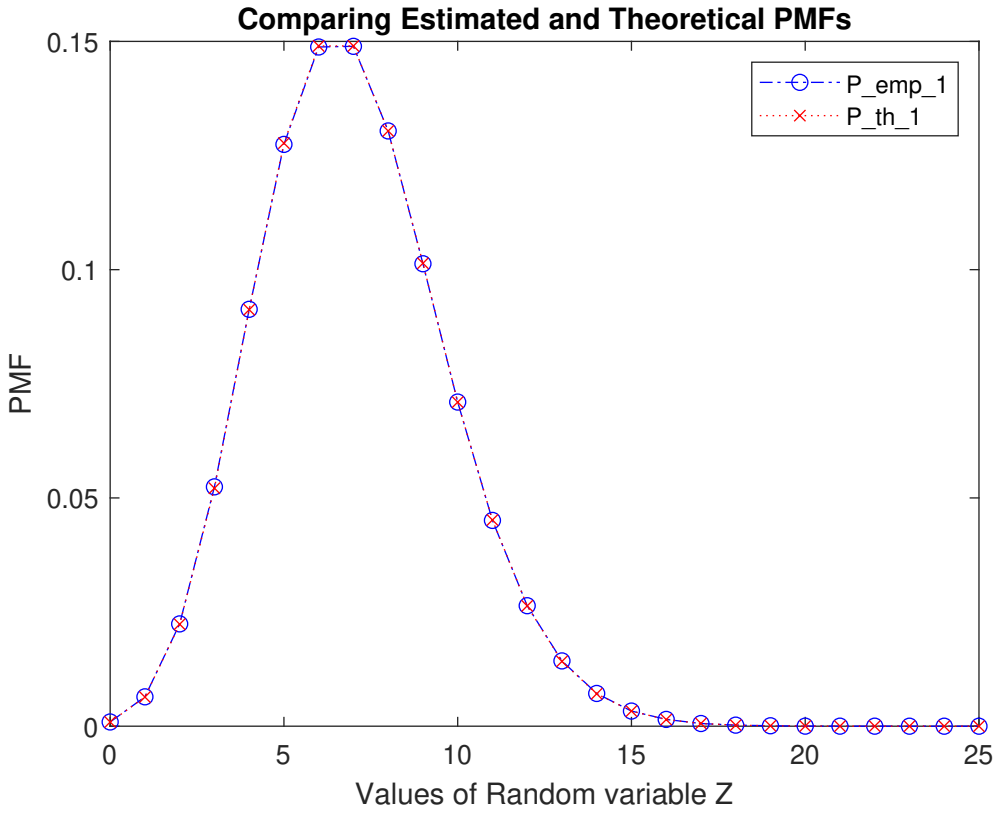
$$= \frac{(\lambda_X + \lambda_Y)^z}{z!} e^{-(\lambda_X + \lambda_Y)} \tag{2.6}$$

   This is again a poisson distribution, which also agrees with the general intuition.

3. Show and compare the values for $\hat{P}(Z = k)$ and for $k = 0, 1, 2, \cdots, 25$.

	k	Theoretical	Empirical
1	0	0.0009	0.0009
2	1	0.0064	0.0064
3	2	0.0223	0.0224
4	3	0.0521	0.0524
5	4	0.0912	0.0913
6	5	0.1277	0.1274
7	6	0.1490	0.1488
8	7	0.1490	0.1489
9	8	0.1304	0.1304
10	9	0.1014	0.1013
11	10	0.0710	0.0710
12	11	0.0452	0.0451
13	12	0.0263	0.0264
14	13	0.0142	0.0143
15	14	0.0071	0.0072
16	15	0.0033	0.0033
17	16	0.0014	0.0015
18	17	0.0006	0.0006
19	18	0.0002	0.0002
20	19	0.0001	0.0001
21	20	0	0
22	21	0	0
23	22	0	0
24	23	0	0
25	24	0	0
26	25	0	0

Values generated are also reported in `problem2.mlx`. For comparing the true and estimated PMFs, we plotted the two on a graph–.


Comparing Estimated and Theoretical PMFs
P_emp_1
P_th_1
PMF
Values of Random variable Z
0.15
0.1
0.05
0
0
5
10
15
20
25

- Implement a Poisson thinning process (as discussed in class) on the random variable Y , where the thinning process uses probability parameter 0.8. Let the thinned random variable be Z.

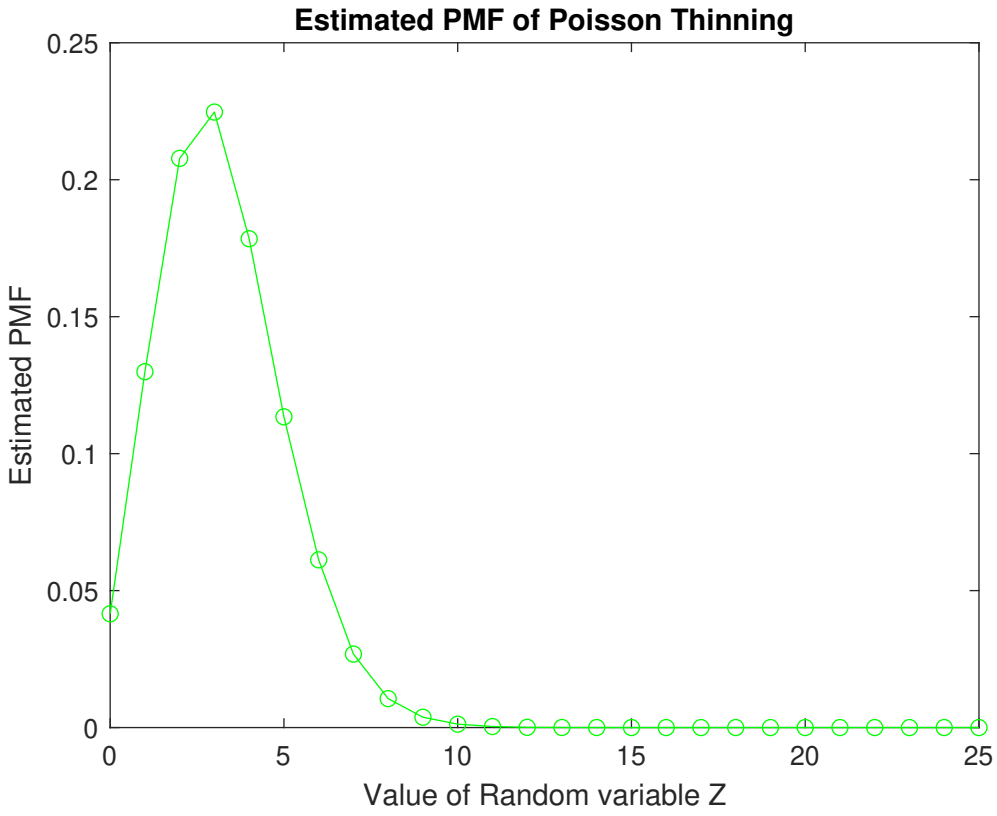
  1. Empirically obtain an estimate $\hat{P}(Z)$ of the PMF $P(Z)$, by drawing $N := 10^5$ instances (sample points) from Y . You may use the poissrnd$(\cdot)$ and binornd$(\cdot)$ functions in Matlab. Report the values of $\hat{P}(Z = k)$ for $k = 0, 1, 2, \cdots, 25$.

     We first generated the pre-thinned variables as an array of size N, then applying binornd to this array to get our thinned variables. To count the frequencies of these values we again used the `hist()` function and then divided by N to get the empirical probability.

```
P_emp_2 = 1x26
  0 to  5
    0.0415    0.1299    0.2078    0.2247    0.1784    0.1134
  6 to 11
    0.0612    0.0268    0.0106    0.0038    0.0012    0.0004
 12 to 17
    0.0001    0.0000    0.0000    0         0         0
 18 to 23
    0         0         0         0         0         0
 24 to 25
    0         0
```

     Values generated are also reported in `problem2.mlx`.

     The following plot was generated from the data–



2. What will the PMF $P(Z)$ be theoretically/analytically ?
   Poisson thinning is a composition of a Poisson and Binomial experiments.

$$P(Z=k) = \sum_{j=k}^{\infty} P(Y=j, Z=k) \tag{2.7}$$

$$= \sum_{j=k}^{\infty} P(Z=k|Y=j)P(Y=j) \tag{2.8}$$

$$= \sum_{j=k}^{\infty} \binom{j}{k} p^k (1-p)^{j-k} \cdot \frac{\lambda_Y^j}{j!} e^{-\lambda} \tag{2.9}$$

$$= e^{-\lambda} \frac{p^k \lambda^k}{k!} \sum_{(j-k)=0}^{\infty} \frac{(1-p)^{j-k}}{(j-k)!} \tag{2.10}$$

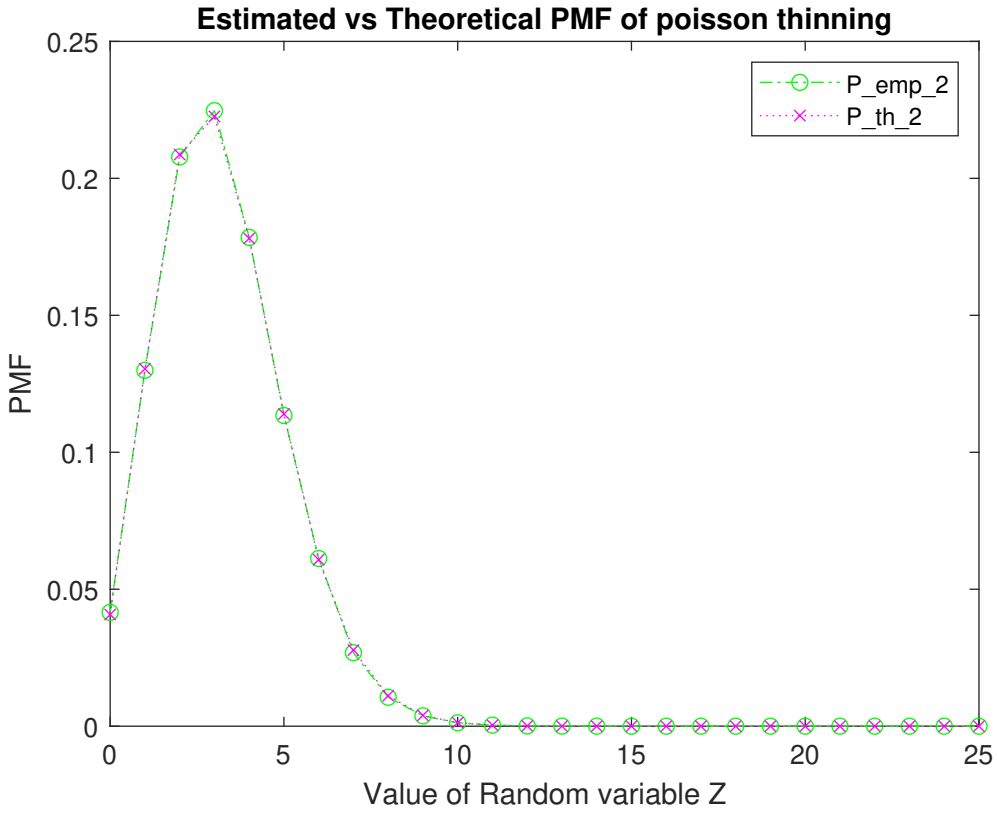
$$= \frac{e^{-\lambda p}(\lambda p)^k}{k!} \tag{2.11}$$

   This is again a poisson distribution which agrees with the general intuition.

3. Show and compare the values for $\hat{P}(Z=k)$ and $P(Z=k)$ for $k = 0, 1, 2, \cdots, 25$.

	k	Theoretical	Empirical
1	0	0.0408	0.0415
2	1	0.1304	0.1299
3	2	0.2087	0.2078
4	3	0.2226	0.2247
5	4	0.1781	0.1784
6	5	0.1140	0.1134
7	6	0.0608	0.0612
8	7	0.0278	0.0268
9	8	0.0111	0.0106
10	9	0.0040	0.0038
11	10	0.0013	0.0012
12	11	0.0004	0.0004
13	12	0.0001	0.0001
14	13	0	0
15	14	0	0
16	15	0	0
17	16	0	0
18	17	0	0
19	18	0	0
20	19	0	0
21	20	0	0
22	21	0	0
23	22	0	0
24	23	0	0
25	24	0	0
26	25	0	0

   Values generated are also reported in `problem2.mlx`. For comparing the true and estimated PMFs, we plotted the two on a graph–.


Estimated vs Theoretical PMF of poisson thinning
P_emp_2
P_th_2
PMF
Value of Random variable Z
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
25

Problem 3

Simulate $N := 10^4$ independent random walkers (as discussed in class) along the real line, each walker starting at the origin, and each walker taking 10^3 steps each of length 10^{-3} .

Approach

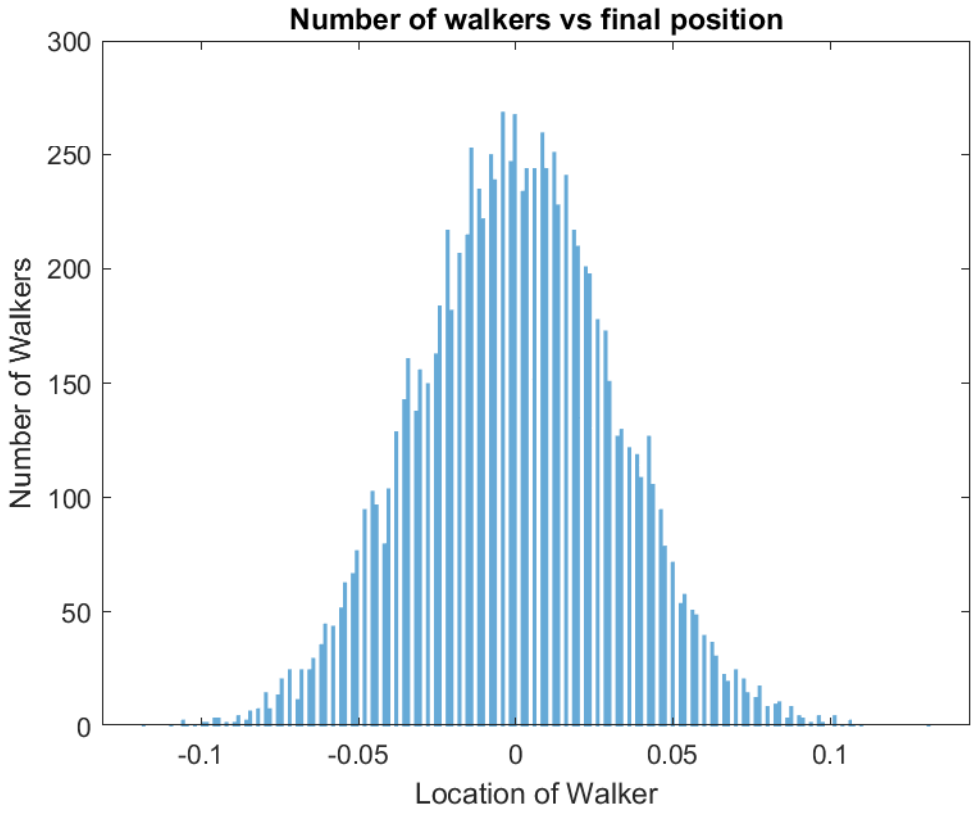
Each random walker has a 0.5 chance of taking a step in the +ve direction and a 0.5 chance of a step in the negative direction. To model this we used the `randi()` function to generate steps of $\{+10^{-3}$ or $-10^{-3}\}$ with equal probability.

We generated all the steps for the 10^4 walkers and 10^3 steps of each of them together as a matrix.

To get the final positions of each walker we just had to sum the matrix along the second dimension.

Similarly, to get the position of the walker at each time step we used the cumulative sum (`cumsum()`) function along the second dimension (here we only took the first 10^3 random walkers as required by the question).

- Plot a histogram of the final locations of all the random walkers. We used the `histogram(·)` function to plot the histogram from the array of the final positions of the random walkers.



Central limit theorem - The random walkers have the same distribution for their final locations and the paths taken by one walker is independent of the other walkers; it becomes easier to believe from this graph that the distribution as $N \to \infty$, will actually be Gaussian.

- For the first 10^3 walkers, plot space-time curves that show the path taken by each walker (as depicted in the class slides). On the graph, draw each path in a different randomly-chosen color for better clarity.

- Consider a random variable X. Consider a dataset that comprises N independent draws (e.g., modeled by $X_1, \cdots, X_N$) from the distribution of X.

  Use the law of large numbers to show that the random variable

$$\hat{M} := \frac{(X_1 + \cdots + X_N)}{N} \tag{3.1}$$

  converges to the true mean

$$M := E[X] \tag{3.2}$$

  as $N \to \infty$. Prove that the expected value of the random variable

$$\hat{V} := \sum_{i=1}^{N} \frac{(X_i - \hat{M})^2}{N} \tag{3.3}$$

  tends to the true variance

$$V := Var(X) \tag{3.4}$$

  as $N \to \infty$.

  The result for $\hat{M}$ follows directly from the law of large numbers. Proof of the law is the following: Let

$$X_l = \frac{X_1 + X_2 \cdots X_N}{N} \tag{3.5}$$

is a random variable denoting the average of X_is where X_is follow the same distribution as X. We know that $E[X + l] = E[X]$ from the linearity property of expectation. Also, since the trials are independent,

$$Var(X_l) = \frac{Var(X_1) + Var(X_2) \cdots Var(X_N)}{N^2} = \frac{\cdot Var(X)}{N} \tag{3.6}$$

From Chebyshev's inequality, we know that

$$P(|X_l - E[X]| \geq \epsilon) \leq \frac{Var(X_l)}{\epsilon^2} = \frac{Var(X)}{N\epsilon^2} \tag{3.7}$$

Assuming that $Var(X)$ is finite, we can always find a sufficiently large N for a given ϵ such that

$$P(|X_l - E[X]| < \epsilon) \geq 1 - \frac{Var(X)}{N\epsilon^2} \tag{3.8}$$

In other words, $X_l \to E[X]$ as $N \to \infty$ (choose arbitrarily small epsilon). Hence $\hat{M}$ converges to M. Now, we proceed to show it for $\hat{V}$.

$$\hat{V} = \sum_{i=1}^{N} \frac{(X_i - \hat{M})^2}{N} \tag{3.9}$$

$$= \sum_{i=1}^{N} \frac{1}{N} \left(X_i^2 - 2X_i\hat{M} + \hat{M}^2 \right) \tag{3.10}$$

$$= \sum_{i=1}^{N} \frac{X_i^2}{N} - 2\hat{M} \sum_{i=1}^{N} \frac{X_i}{N} + \hat{M}^2 \tag{3.11}$$

$$= \sum_{i=1}^{N} \frac{X_i^2}{N} - \hat{M}^2 \tag{3.12}$$

$$\longrightarrow E\left[X^2\right] - M^2 \tag{3.13}$$

$$= Var(X) \tag{3.14}$$

Where we used the law of large numbers on the first term $\sum_{i=1}^{N} \frac{X_i^2}{N}$, as X_i^2 is also a random variable, to show that it converges to $E[X^2]$ in 3.13. And $\hat{M} \to M \implies \hat{M}^2 \to M^2$

- Report the empirically-computed mean $\hat{M}$ and the empirically-computed variance $\hat{V}$ of the final locations of the random walkers.

  Empirical mean: $1.386 \cdot 10^{-4}$
  Empirical variance: $1 \cdot 10^{-3}$

- What should the values of the true mean and the true variance be for the random variable that models the final location of the random walker, as function of the step length and the number of steps ?

  True mean is 0 since there is an equal probability to step either left or right.
  Variance $= 2Dt = \frac{t(\Delta z)^2}{\Delta t} = n \cdot (\Delta z)^2 = 10^3 \cdot (10^{-3})^2 = 10^{-3}$

  Alternatively, variance in number of steps$(V_s) = npq = 0.25 \cdot 10^3 \implies$
  variance in the locations $= V_s \cdot (2\Delta z)^2 = 10^{-3}$

- Report the error between the empirically-computed mean and the true mean. Report the error between the empirically-computed variance and the true variance.

  Error in mean(absolute): $1.386 \cdot 10^{-4}$
  Error in variance(absolute): $4.8 \cdot 10^{-6}$
  Error in variance(relative): 0.0048

  Refer `problem3.mlx` for code and data.

Problem 4

Consider a continuous random variable X that has an M-shaped probability density function (PDF) $P_X(\cdot)$ as follows:

$$P_X(x) := 0 \text{ for } |x| > 1, \text{ and } P_X(x) := |x| \text{ for } x \in [-1, 1] \tag{4.1}$$

Consider independent continuous random variables $\{X_i : i = 1, 2, \cdots, \infty\}$ with PDFs identical to that of X. Define random variables

$$Y_N := \frac{1}{N} \sum_{i=1}^{N} X_i, \tag{4.2}$$

for $N = 1, 2, \cdots, \infty$, which have associated distributions $P_{Y_N}(\cdot)$.

- Write code to generate independent draws from $P_X(\cdot)$. Your code can use only the uniform random number generator rand() (no other generator). Submit this code.

  Let x be the variable representing the values of rand() (X, uniform distribution on $[0, 1]$).

  Now, we have to find a random variable Y, $y = f(x)$ such that $P(y) = |y|, -1 \leq y \leq 1$.

  Every interval Δx should map to a corresponding Δy such that the probabilities of X being in $x \to x + \Delta x$ and Y being in $y \to y + \Delta y$ are the same.

  Which implies

  $$P_Y(y)dy = P_X(x)dx \tag{4.3}$$

  As

  $$P(x) = 1, 0 \leq x \leq 1 \text{ and } P(y) = |y|, -1 \leq y \leq 1, \tag{4.4}$$

  $$P_Y(y)dy = P_X(x)dx \tag{4.5}$$

  $$\implies |y|dy = dx \tag{4.6}$$

  $$\implies y = \sqrt{2(x + c_1)}, -\sqrt{2(c_2 - x)} \tag{4.7}$$

  We assume that the solution y is continuous. We know that the range of y is $[-1, 1]$ and the two solution functions are continuous. Let us call the first solution y_1 and the second solution y_2. $y_1 \geq 0$ and $y_2 \leq 0$. Hence, $\exists\ \alpha \in [0, 1]$ such that $y_1(\alpha) = y_2(\alpha) = 0$. We also know that the mass to the left of $y = 0$ is the same as the mass to the right of $y = 0$. Hence, $\alpha = 0.5$ by symmetry.

  $$\implies c_1 = -0.5 \text{ and } c_2 = 0.5 \tag{4.8}$$

  The transformation function $y = y(x)$ can be written in a simplified form as:

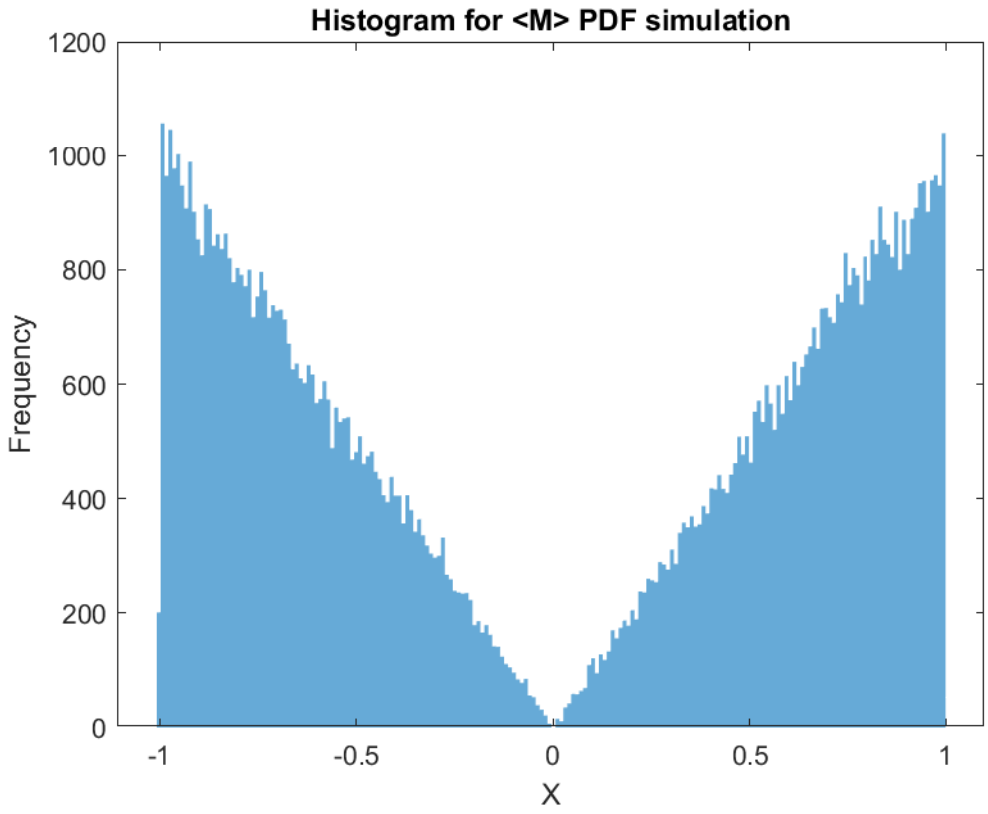
  $$y = \text{sgn}(2x - 1) \cdot \sqrt{|2x - 1|} \tag{4.9}$$

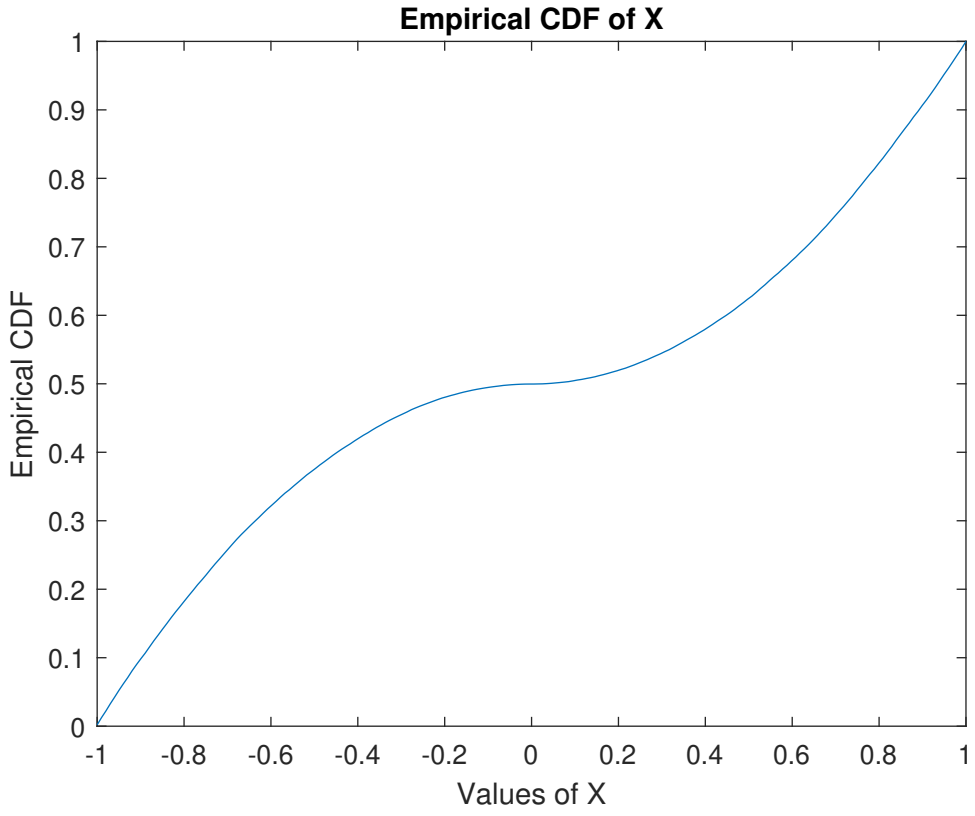
To generate on instance of this random variable –

```
mprob = sign(2*rand()-1).*sqrt(abs(2*rand()-1));
```

Refer `problem4.mlx`.

- Show plots of
  1. The histogram (with 200 bins)





2. Cumulative distribution function (CDF), both using $M := 10^5$ draws from the PDF $P_X(\cdot)$.

- Use the code written in the previous sub-question to write code to generate independent draws from $P_{Y_N}(\cdot)$. Submit this code.

  Refer `problem4.mlx`. We used this function to generate values of Y_N (as a function of N)–

```
function pyn = avg_M(N)
    pyn = mean(sign(2*rand(N,1)-1).*sqrt(abs(2*rand(N,1)-1)));
end
```

- Show plots (separately) of histograms using draws from each of the PDFs $P_{Y_N}(\cdot)$ for $N = 2, 4, 8, 16, 32, 64$.

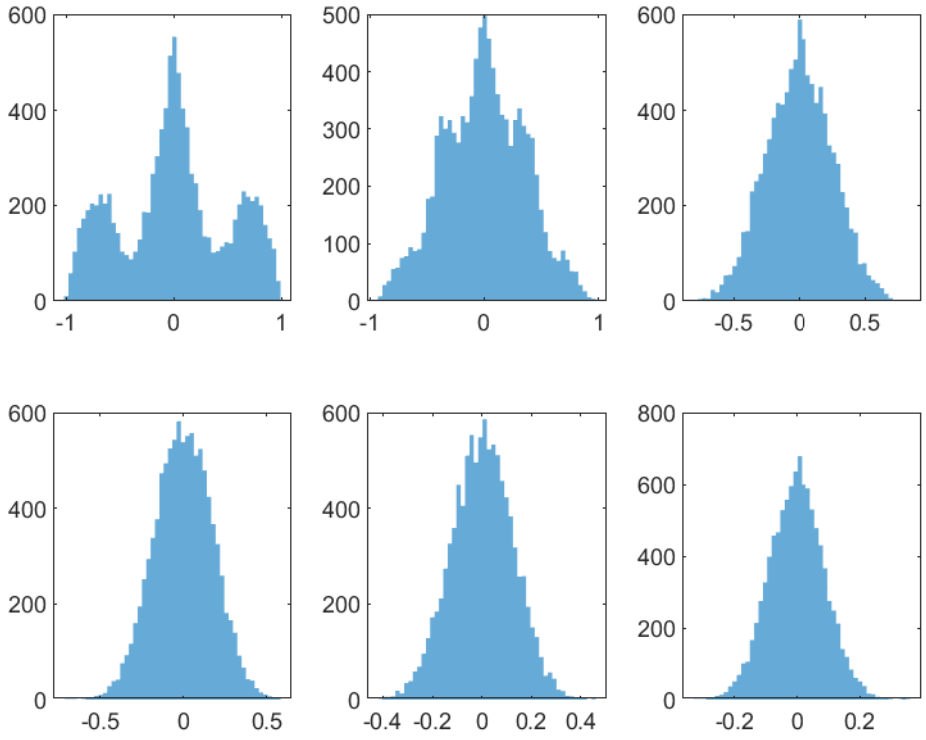
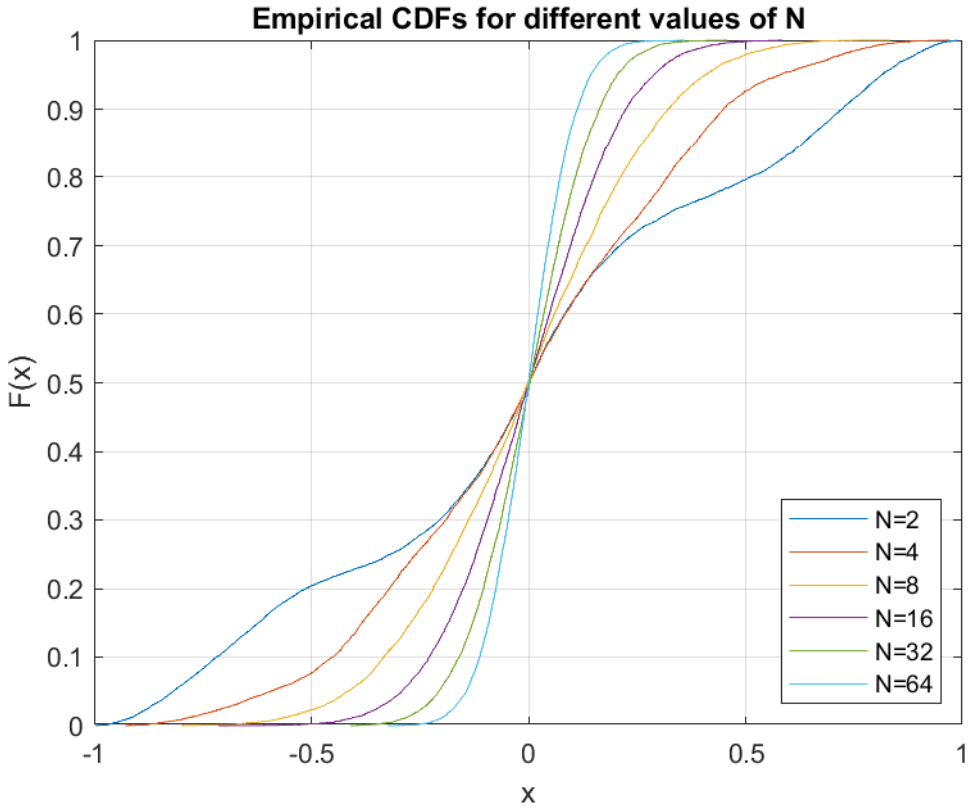


Figure 4.1: Histograms corresponding to $N = 2, 4, 8, 16, 32, 64$ respectively (L-R T-B)

Show plots, on the same graph, of all the CDFs associated with Y_N for $N = 1, 2, 4, 8, 16, 32, 64$, computed using 10^4 draws from each $P_{Y_N}(\cdot)$. Plot each CDF curve using a different color. You may use the cdfplot(·) function in Matlab.



Problem 5

Generate a dataset comprising a set of real numbers drawn from the uniform distribution on $[0, 1]$. Consider various dataset sizes

$$N = 5, 10, 20, 40, 60, 80, 100, 500, 10^3, 10^4$$

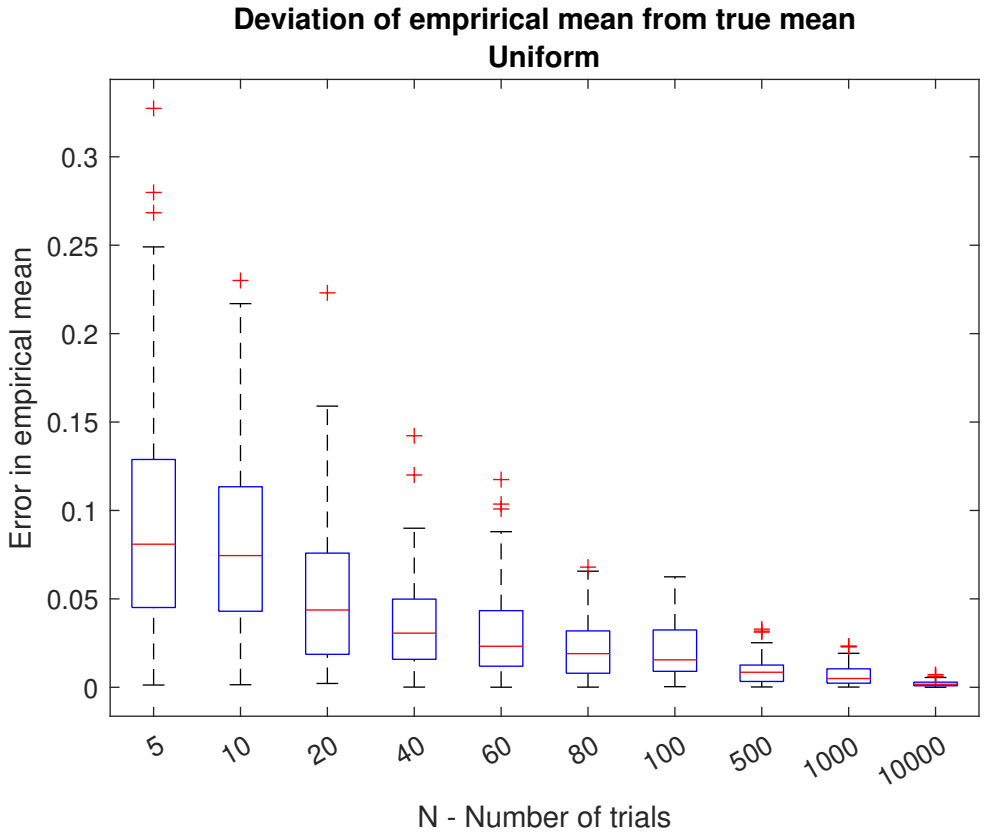
For each N, repeat the following experiment $M := 100$ times:

1. generate the data
2. compute the average $\hat{\mu}$
3. measure the error between the computed average $\hat{\mu}$ and the true mean μ as $|\hat{\mu} - \mu_{true}|$.
   Refer to `problem5.mlx` for the generated data, and implementation.

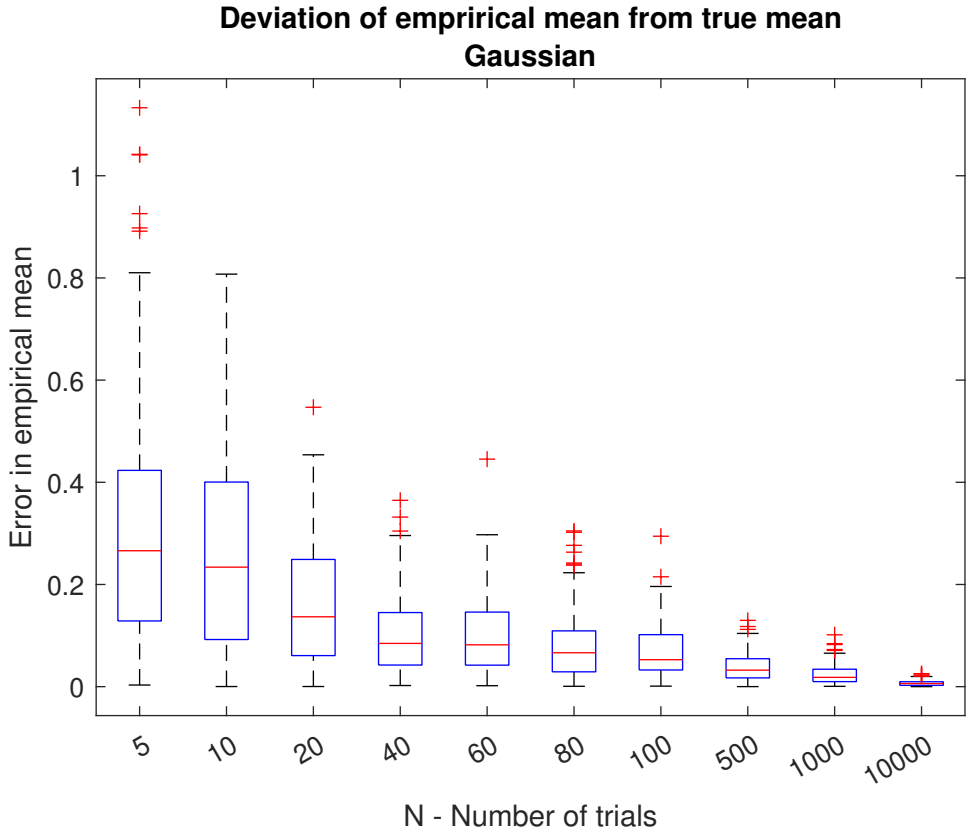
Approach

We generated the M experiments for each value of N as a single matrix, with each column corresponding to a particular N (the boxplot also takes input in this way). Then looped over each value of N, for which we computed the error M times , stored as a 1D array. And concatenated the arrays to get the desired matrix. We used the same procedure for both parts, with just the random generator functions differing.

- For the uniform distribution, plot a single graph that shows the distribution of errors (across M repeats) for all values of N using a box-and-whisker plot. You may use the boxplot($\cdot$) function in Matlab.



- Repeat the above question by replacing the uniform distribution by a Gaussian distribution with $\mu := 0$ and $\sigma^2 := 1$.



- Interpret what you see in the graphs. What happens to the distribution of error as N increases ? This graph is a direct reflection of the law of large numbers. i.e. the mean converges to the theoretical mean as $N \to \infty$
  When the value of N is small, the data that is generated is more 'random' than the data generated when N is large. So for small N, the spread in the deviations from the true mean are relatively high. As N increases to large enough values(say 1000 and 10000), the mean values obtained over the $M(100)$ iterations are almost equal to the true mean value. In fact, there is barely any data set that is conspicuously farther from 0 when $N = 10000$; as is evident from the graph.

  In conclusion, the distribution of error decreases as N increases.