ABDULLAH GÜL
UNIVERSITY

MATH 301

PROBABILITY & STATISTICS

PROJECT I

SIMULATING PROBABILITY DISTRIBUTIONS

Bilal KABAŞ
110110320

April 2020

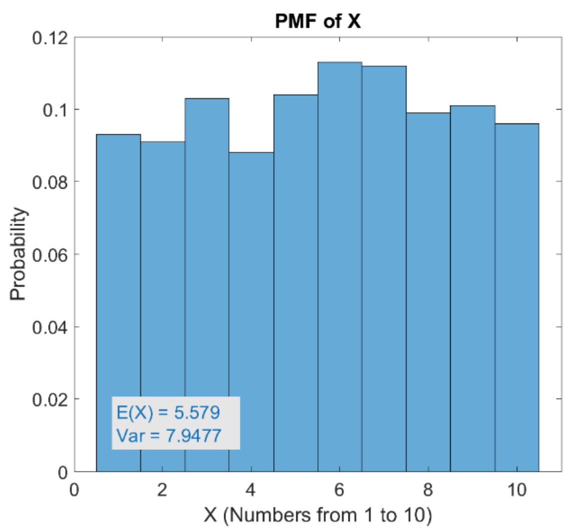
AIM

The aim of this report is to present MATLAB codes simulating uniform, normal, binomial and Poisson distributions. Subsequently, a PDF will be fit to a probability distribution of a built-in dataset in MATLAB. The interpretation of the simulation results will also be provided.

TASK I

Simulate drawing of N cards from a deck of cards which include integers between 1 and 10. The random variable X is the number on the cards drawn randomly.

```
deck = [1 2 3 4 5 6 7 8 9 10];
N = 1000;
% Simulate drawing N cards from the deck
S = deck(randi([1 10],1,N));

% Plot PMF
histogram(S,'normalization','probability')
title("PMF of X")
xlabel("X (Numbers from 1 to 10)")
ylabel("Probability")

% Expected Value
mu = 0;
for x=1:10
    p = sum(S(:) == x)/N;
    mu = mu + x*p;
end

% Variance
var = 0;
for i=1:length(S)
    var = var + (S(i) - mu)^2;
end
var = var/(N-1);
```



(a) (b)

Fig. 1. (a) MATLAB code for simulating drawing N cards from a deck. (b) Probability mass function of random variable X along with the expected value and the variance.

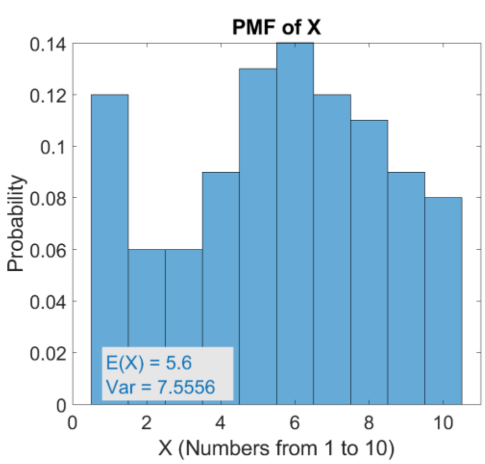


Fig. 2. Probability mass function for N = 100.

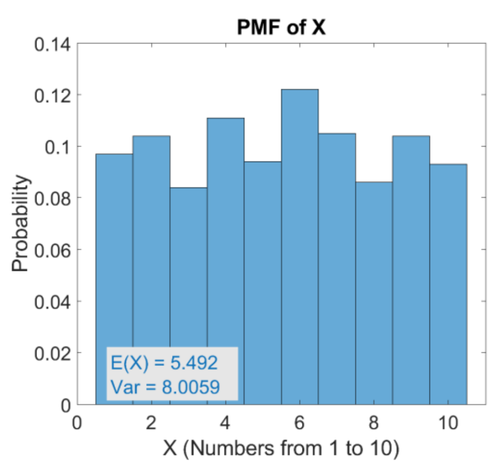


Fig. 3. Probability mass function for N = 10^3.

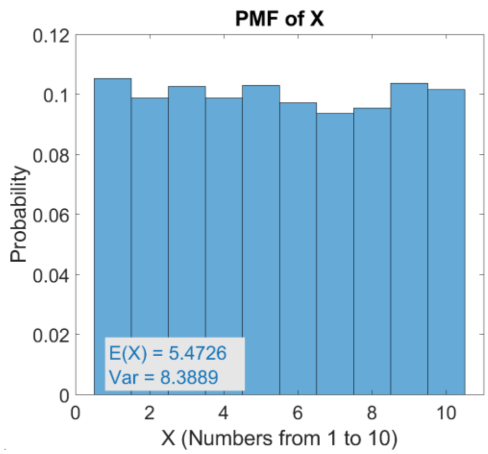


Fig. 4. Probability mass function for N = 10^4.

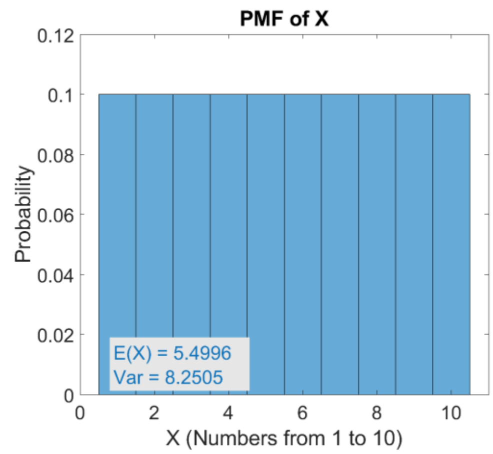


Fig. 5. Probability mass function for N = 10^8.

TABLE I. EXPECTED VALUE AND VARIANCE FOR DIFFERENT N VALUES

N	Expected Value	Variance
100	5.6	7.56
10^3	5.49	8.01
10^4	5.47	8.39
10^8	5.5	8.25

Interpreting the Results

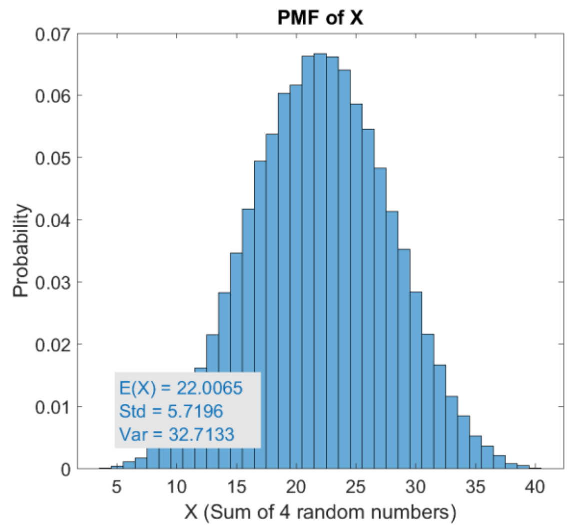
As it is seen in Fig. 1, *randi* function produces array indices between 1 and 10 uniformly. The main conclusion that can be drawn from these results is that the distribution of numbers converges to the uniform distribution as the number of cards drawn from the deck (N) increases. Additionally, the expected value also converges to 5.5 for the increasing values of N. It should be also noted that as the expected value becomes closer to 5.5, the variance converges to 8.25.

As it can be seen in the MATLAB code shown in Fig. 1, the expected value and the variance are calculated by using their actual formulas. Even though MATLAB has *mean* and *var* functions which are calculating the expected value and the variance of a given dataset respectively, this approach has been found more convenient to get better understanding of the topic.

TASK II

Simulate the sum of 4 randomly selected integers between 1 and 10. The random variable X is that sum.

```
deck = [1 2 3 4 5 6 7 8 9 10];
N = 100000;
S = zeros(1,N);

% Iterate N times drawing 4 random cards
and sum the numbers on them
for i=1:N
    S(i) = sum(deck(randi([1 10],1,4)));
end

% Plot PMF
histogram(S,'normalization','probability');
title("PMF of X")
xlabel("X (Sum of 4 numbers)")
ylabel("Probability")

% Expected Value
mu = 0;
for i=4:40
    x = sum(S(:) == i)/N;
    mu = mu + i*x;
end

% Variance
var = 0;
for i=1:length(S)
    var = var + (S(i) - mu)^2;
end
var = var/(N-1);
```



(a) (b)

Fig. 6. (a) MATLAB code for simulating drawing 4 cards from a deck and summing up the numbers on them. (b) Probability mass function of random variable X along with the expected value and the variance.

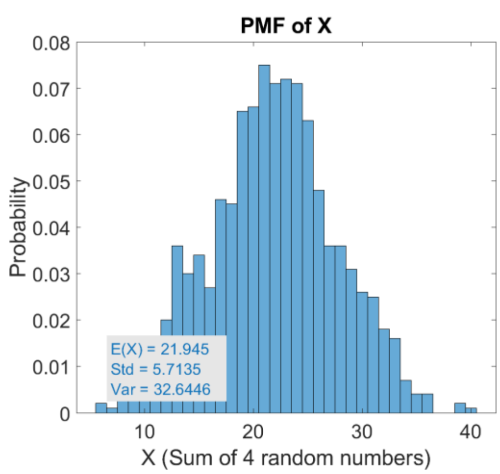


Fig. 7. Probability mass function for N = 10^3.

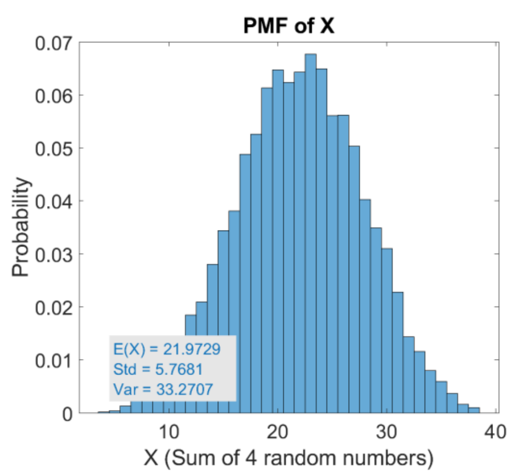


Fig. 8. Probability mass function for N = 10^4.

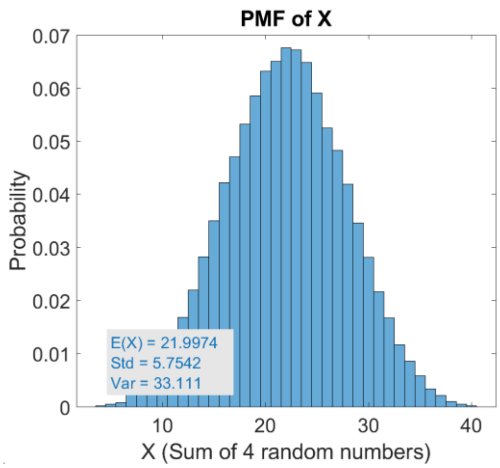


Fig. 9. Probability mass function for N = 10^5.

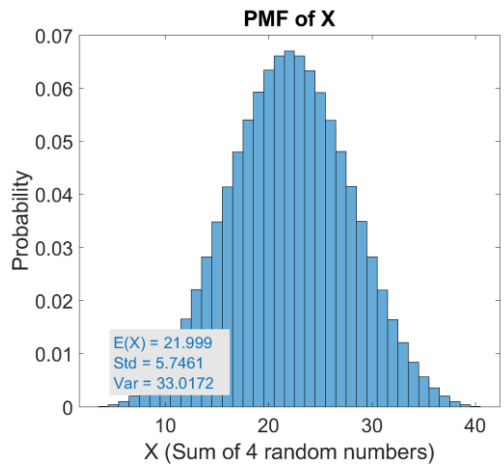


Fig. 10. Probability mass function for N = 10^7.

TABLE II. EXPECTED VALUE, STANDARD DEVIATION AND VARIANCE FOR DIFFERENT N VALUES

N	Expected Value	Standard Deviation	Variance
10^3	21.95	5.71	32.64
10^4	21.97	5.77	33.27
10^5	22	5.75	33.1
10^7	22	5.71	33.01

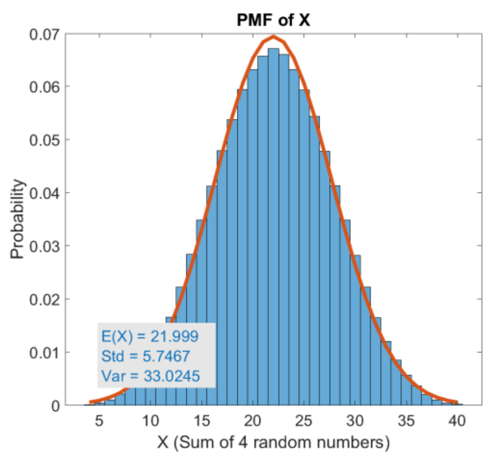


Fig. 11. The orange line indicates a normal distribution with $\mu = 22$ and $\sigma^2 = 33.02$.

Interpreting the Results

The code for simulating the task 1 has been slightly modified. Now, 4 cards from a deck are chosen randomly to be added and this action is iterated N times.

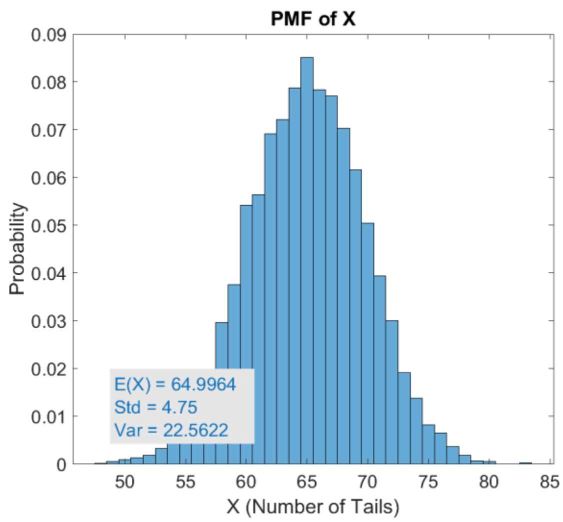
As N increases, the expected value gets closer to 22 and the PMF forms a bell curve shape. Fig. 11 shows how this distribution is similar to the normal distribution. Comparing P(X=4) and P(X=22), it is obvious that P(X=22) > P(X=4) since there are many combinations of 4 random integers that their sum makes 22, whereas the only way of obtaining X=4 is to get 4 consecutive ones which is the least probable case. Another important point is that according to the Central Limit Theorem, when independent random variables are added constituting another random variable which is that sum, the distribution of that sum converges to a normal distribution. However, it should be noted that the distribution of sum in this case is not continuous as X can be any integer between 4 and 40.

TASK III

Simulating tossing a coin N times with P(tails) = p. The random variable X indicates how many times the coin comes up tails.

```
P_tails = 0.65; % Prob. of obtaining tails
N = 100;
% Create a weighted array according to 'p'
W = [ones(1,P_tails*100) zeros(1,(100-
P_tails*100))];
S = zeros(1,10000);
I = 100000;     % # of iteration

% Toss coin for N times in each iteration
for x=1:I
   S(x) = sum(W(randi([1 100],1,N)));
end

% Plot PMF
histogram(S,'normalization', 'probability');
title("PMF of X")
xlabel("X (Sum of 4 random numbers)")
ylabel("Probability")

% Expected Value
mu = 0;
for x=1:N
    p = sum(S(:) == x)/I;
    mu = mu + x*p;
end

% Variance
var = 0;
for i=1:length(S)
    var = var + (S(i) - mu)^2;
end
var = var/(I-1);
```

(a)



(b)

Fig. 12. (a) MATLAB code for simulating flipping a coin N times with P(tails) = p. (b) Probability mass function of random variable X along with the expected value, standard deviation and variance.

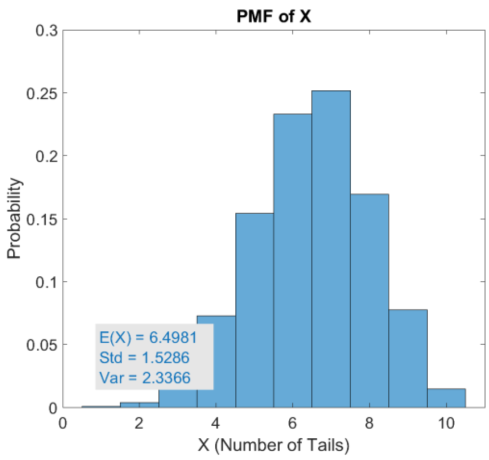


Fig. 13. Probability mass function for N = 10 and P(tails) = 0.65.

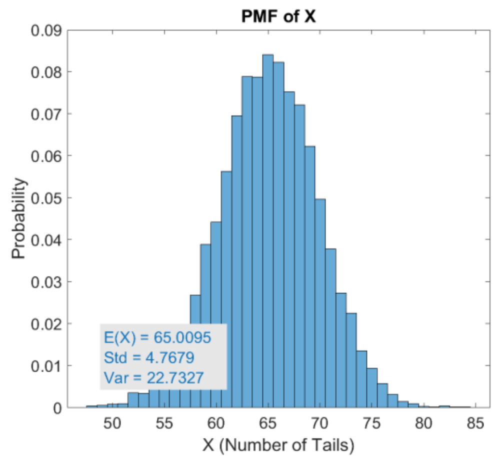


Fig. 14. Probability mass function for N = 100 and P(tails) = 0.65.

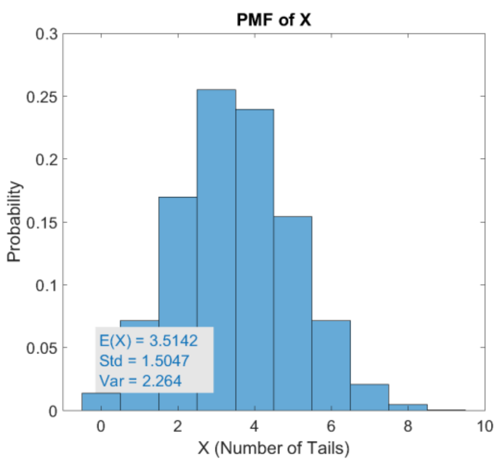


Fig. 15. Probability mass function for N = 10 and P(tails) = 0.35.

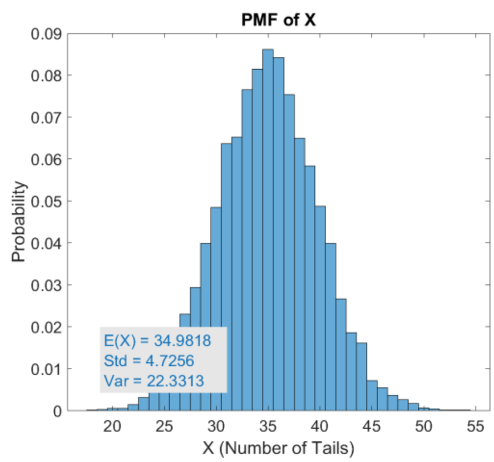


Fig. 16. Probability mass function for N = 100 and P(tails) = 0.35.

TABLE III. EXPECTED VALUE AND VARIANCE FOR DIFFERENT N AND P VALUES

N	P(tails)	Expected Value	Variance
10	0.1	1.00	0.90
100	0.1	10.01	8.96
1000	0.1	99.99	90.08
10	0.3	3.00	2.09
100	0.3	30.02	20.98
1000	0.3	300.06	210.33
10	0.5	5.00	2.52
100	0.5	49.98	24.97
1000	0.5	500.05	249.94
10	0.8	8.00	1.60
100	0.8	80.01	15.98
1000	0.8	799.95	159.81

Interpreting the Results

Flipping a coin N times with P(tails) = p is an example of the Bernoulli process as we have two possible outcomes for each trial which are tails or heads. In this task, this Bernoulli process has been repeated and the PMF showing how many times the coin comes up tails has been demonstrated. Consequently, in the end of his Bernoulli process, a binomial distribution has been obtained. The probability of the random variable of a binomial distribution is given by

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

It should be noted that each flipping event is independent. The expected value and the variance for a binomial distribution are found by

$$E(X) = Np$$

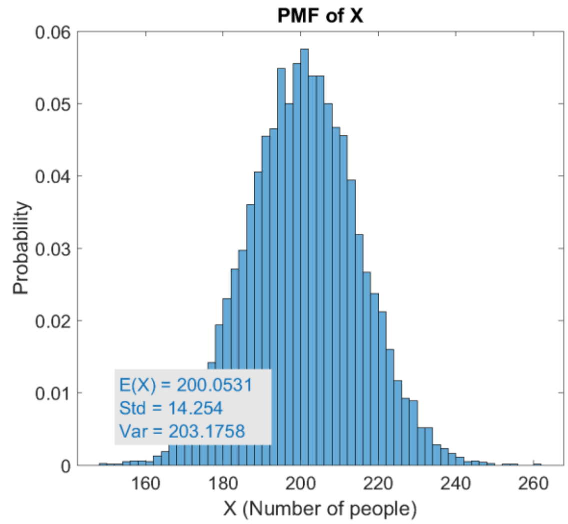
$$\sigma^2 = Np(1 - p)$$

The results shown in Table III are in an agreement with these equations that also mathematically shows that this distribution is exactly an example of binomial distribution.

TASK IV

Simulate the number of people entering a shop in one hour in Kayseri Park. The random variable X is the number of people. Simulate this RV for different lambdas.

```
L = 200;           % Avg # of people per hour
% Smallest time interval is 1/36000 h
precision = 36000;
% Create a weighted array according to 'L'
W = [ones(1,L) zeros(1,precision-L)];
I = 10000;          % # of iterations
S = zeros(1,I);

% Get 3600 samples from W in each iteration
for i=1:I
   S(i) = sum(W(randi([1
precision],1,precision)));
end

% Plot PMF
histogram(S,'normalization','probability');
title("PMF of X")
xlabel("X (Number of people)")
ylabel("Probability")

% Expected Value
mu = 0;
for x=min(S):max(S)
    p = sum(S(:) == x)/I;
    mu = mu + x*p;
end

% Variance
var = 0;
for i=1:length(S)
    var = var + (S(i) - mu)^2;
end
var = var/(I-1);
```



(a) (b)

Fig. 17. (a) MATLAB code for simulating number of people entering a shop within 1 hour for given value of lambda which indicates the average number of people. (b) Probability mass function of random variable X along with the expected value, standard deviation and variance.

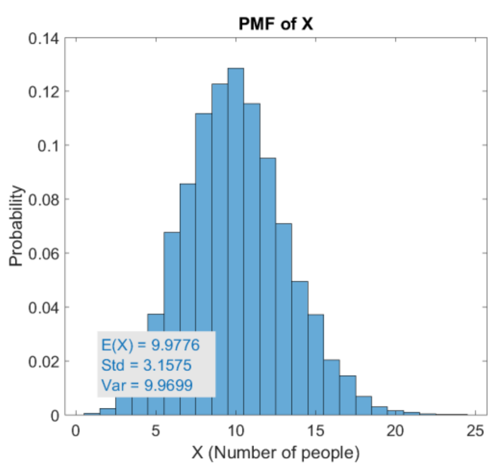


Fig. 18. Probability mass function for $\lambda = 10$.

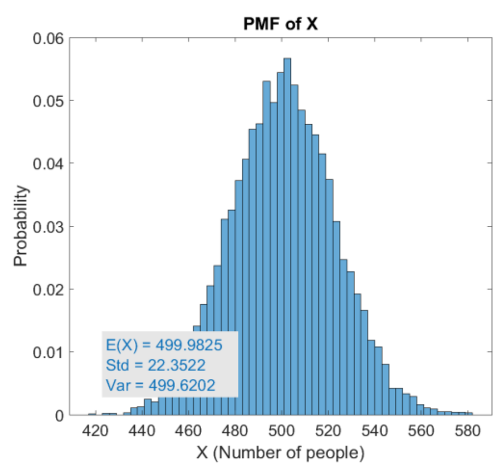


Fig. 19. Probability mass function for $\lambda = 500$.

TABLE IV. EXPECTED VALUE AND VARIANCE FOR DIFFERENT VALUES OF LAMBDA

Lambda (λ)	Expected Value	Variance
10	10.02	10.1
50	49.89	49.81
200	200.16	198
500	500.36	481.9
1000	1000.33	964.17
2500	2499.95	2331.3

Interpreting the Results

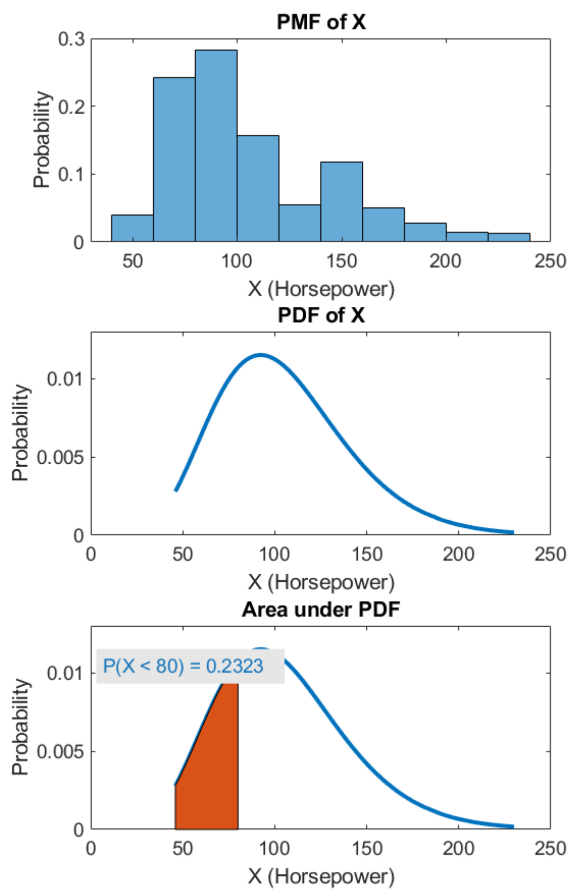
In order to simulate number of people entering a shop within one hour, the MATLAB code written to simulate the task 3 has been modified. Now, we are sampling from 36000 data. There is again a Bernoulli process, exactly a person enters, or nobody enters within a specific interval. Lambda is the number of ones in the weighted array and the number of zeros is (36000 – L). This distribution is an example of the Poisson distribution as we have average number of people and the random variable is the number of people **within one hour.** In the Poisson distribution, the expected value and the variance are equal to lambda (λ). As it can be seen in Table IV, the expected value and the variance are close to lambda.

TASK V

Choose one built-in dataset in MATLAB, plot the histogram and fit a probability density function (pdf) to it. Compute the probability of a random value being less than a predetermined threshold of your choice.

```
% Measurements of cars, 1970-1982
load carbig.mat
S = Horsepower;
S = sort(S(~isnan(S)));

% Plot PMF
subplot(3,1,1)
histogram(S,'normalization','probability')
title("PMF of X")
xlabel("X (Horsepower)")
ylabel("Probability")

% Fit 'Gamma' distribution
subplot(3,1,2)
pd = fitdist(S,'gamma');
y = pdf('gamma',S,pd.a,pd.b);
plot(S, y)
ylim([0 0.013])
title("PDF of X")
xlabel("X (Horsepower)")
ylabel("Probability")

threshold = 80;     % Threshold
T = find(S==threshold);
T = T(1);

% Calculate area under the curve
A = 0;               % Area under the curve
for i=1:T
   A = A + (S(i+1)-S(i))*y(i);
end

% Plot the area
subplot(3,1,3)
plot(S,y);
title("Area under PDF")
xlabel("X (Horsepower)")
ylabel("Probability")
hold on
area(S(1:T), y(1:T));
```

(a)



(b)

Fig. 20. (a) MATLAB code for fitting a probability density function to a built-in dataset. In this case, Gamma distribution has been fit to the actual distribution of horsepowers. (b) PMF, PDF and area under the curve giving the probability.

TABLE V. PROBABILITIES FOR DIFFERENT THRESHOLD VALUES

Threshold Value	P(X < x)
60	0.059
80	0.23
100	0.46
200	0.97
220	0.98

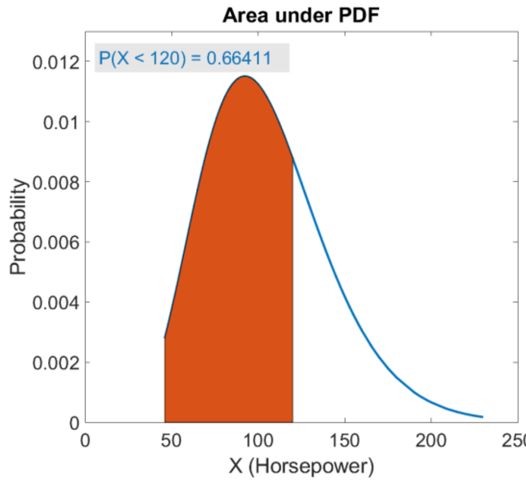


Fig. 21. Probability of the horsepower being less than 120 is the area under the curve which is 0.66.

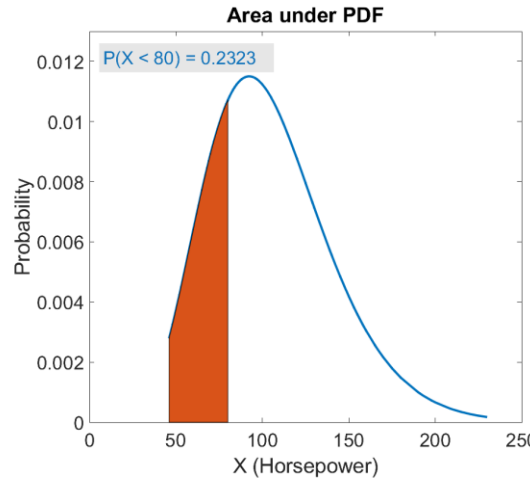


Fig. 22. Probability of the horsepower being less than 80 is the area under the curve which is 0.23.

Interpreting the Results

For the task 5, a built-in dataset which includes the distribution of horsepower between 1970 and 1982 has been used. In order to fit an appropriate probability density function to this distribution, MATLAB's built-in *Distribution Fitter* app has been initially used. The *Distribution Fitter* app has given the most accurate result for the Gamma distribution which is shown in Fig. 23. Consequently, the Gamma distribution has been fit to the actual distribution using MATLAB's built-in *fitdist* function which returns shape and scale coefficients. Then *pdf* function has been used to obtain probability values.

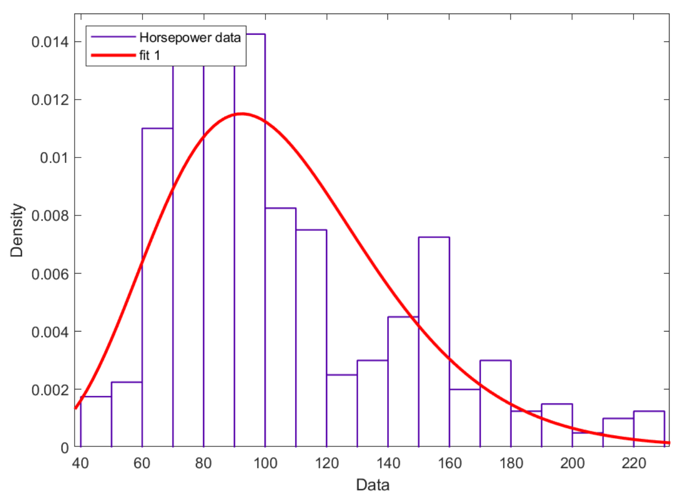


Fig. 23. Fitting Gamma distribution to the actual distribution using the Distribution Fitter app.

The area under curve is calculated by applying the Riemann Sum in the MATLAB code shown in Fig. 20. The area under PDF curve gives a specific probability. For example, in Fig. 21, the probability of horsepower being less than 120 is the area shaded by orange color and can be denoted as P(X<120).

ATTACHMENTS

Complete MATLAB Code for the Task I

```
deck = [1 2 3 4 5 6 7 8 9 10];
N = 1000;
% Simulate drawing N cards from the deck
S = deck(randi([1 10],1,N));

% Plot PMF
histogram(S,'normalization','probability')
title("PMF of X")
xlabel("X (Numbers from 1 to 10)")
ylabel("Probability")

% Expected Value
mu = 0;
for x=1:10
    p = sum(S(:) == x)/N;
    mu = mu + x*p;
end

% Variance
var = 0;
for i=1:length(S)
    var = var + (S(i) - mu)^2;
end
var = var/(N-1);

% Annotation
str = sprintf("E(X) = " + mu + "\nVar = " + var);
annotation('textbox',[0.19 .1 .1 .15],'String',str,'EdgeColor','none',...
    'FitBoxToText','on','background','#e6e6e6','color','#0f72ba')
```

Complete MATLAB Code for the Task II

```
deck = [1 2 3 4 5 6 7 8 9 10];
N = 1000000;
S = zeros(1,N);

% Iterate N times drawning 4 random cards and sum the numbers on them
for i=1:N
    S(i) = sum(deck(randi([1 10],1,4)));
end

% Plot PMF
histogram(S,'normalization','probability');
title("PMF of X")
xlabel("X (Sum of 4 random numbers)")
ylabel("Probability")

% Expected Value
mu = 0;
for x=4:40
    p = sum(S(:) == x)/N;
    mu = mu + x*p;
end

% Variance
var = 0;
for i=1:length(S)
    var = var + (S(i) - mu)^2;
end
var = var/(N-1);
```

```
hold on

y = pdf('normal',sort(S),mu,sqrt(var));
plot(sort(S),y,'linewidth',3);

% Annotation
str = sprintf("E(X) = " + mu + "\nStd = " + sqrt(var) + "\nVar = " + var);
annotation('textbox',[0.19 .14 .1 .15],'String',str,'EdgeColor','none',...
    'FitBoxToText','on','background','#e6e6e6','color','#0f72ba')
```

Complete MATLAB Code for the Task III

```
P_tails = 0.35;     % Probability of obtaining tails
N = 100;

% Create a weighted array according to 'p'
W = [ones(1,P_tails*100) zeros(1,(100-P_tails*100))];
S = zeros(1,10000);
I = 10000;          % # of iteration

% Toss the coin for N times in each iteration
for x=1:I
   S(x) = sum(W(randi([1 100],1,N)));
end

% Plot PMF
histogram(S,'Normalization', 'Probability');
title("PMF of X")
xlabel("X (Number of Tails)")
ylabel("Probability")

% Expected Value
mu = 0;
for x=1:N
    p = sum(S(:) == x)/I;
    mu = mu + x*p;
end

% Variance
var = 0;
for i=1:length(S)
    var = var + (S(i) - mu)^2;
end
var = var/(I-1);

% Annotation
str = sprintf("E(X) = " + mu + "\nStd = " + sqrt(var) + "\nVar = " + var);
annotation('textbox',[0.19 .14 .1 .15],'String',str,'EdgeColor','none',...
    'FitBoxToText','on','background','#e6e6e6','color','#0f72ba')
```

Complete MATLAB Code for the Task IV

```
L = 500;            % Average # of people per hour
precision = 36000;   % Smallest time interval is 1 second
% Create a weighted array according to 'L'
W = [ones(1,L) zeros(1,precision-L)];
I = 10000;           % # of iterations
S = zeros(1,I);

% Get 3600 samples from weighted array in each iteration
for i=1:I
   S(i) = sum(W(randi([1 precision],1,precision)));
end

% Plot PMF
```

```
histogram(S,'Normalization','Probability');
title("PMF of X")
xlabel("X (Number of people)")
ylabel("Probability")

% Expected Value
mu = 0;
p = zeros(1,I);
for x=min(S):max(S)
    p = sum(S(:) == x)/I;
    mu = mu + x*p;
end

% Variance
var = 0;
for i=1:length(S)
    var = var + (S(i) - mu)^2;
end
var = var/(I-1);

% Annotation
str = sprintf("E(X) = " + mu + "\nStd = " + sqrt(var) + "\nVar = " + var);
annotation('textbox',[0.19 .14 .1 .15],'String',str,'EdgeColor','none',...
    'FitBoxToText','on','background','#e6e6e6','color','#0f72ba')
```

Complete MATLAB Code for the Task V

```
% Measurements of cars, 1970-1982
load carbig.mat
S = Horsepower;
S = sort(S(~isnan(S)));

% Plot PMF
subplot(3,1,1)
histogram(S,'Normalization','Probability')
title("PMF of X")
xlabel("X (Horsepower)")
ylabel("Probability")

% Fit 'Gamma' distribution
subplot(3,1,2)
pd = fitdist(S,'gamma');
y = pdf('gamma',S,pd.a,pd.b);
plot(S, y,'linewidth',3)
ylim([0 0.013])
title("PDF of X")
xlabel("X (Horsepower)")
ylabel("Probability")

threshold = 220;    % Threshold
T = find(S==threshold);
T = T(1);

A = 0;              % Area under the curve
for i=1:T
   A = A + (S(i+1)-S(i))*y(i);
end

% Plot the area
subplot(3,1,3)
plot(S,y,'linewidth',3);
title("Area under PDF")
xlabel("X (Horsepower)")
ylabel("Probability")
ylim([0 0.013])
hold on
area(S(1:T), y(1:T));
```

```
% Annotation
str = sprintf("P(X < " + threshold + ") = " + A);
annotation('textbox',[0.15 0.15 .1 .15],'String',str,'EdgeColor','none',...
    'FitBoxToText','on','background','#e6e6e6','color','#0f72ba')
```