EE - 511 PROJECT 0 - COIN TOSSING EXPERIMENT

Name - Tamoghna Chattopadhyay
Student ID # - 8541324935

1. PROBLEM STATEMENT

A) Simulate tossing a coin 100 times and record
   i) the number of heads N_H
   ii) the length of the longest run of heads L_H

B) Simulate repeatedly tossing a coin and record the number of tosses until : the first head occurs S_1 ; the first time 2 heads occur in sequence S_2 ; the first time 3 heads occur in sequence S_3 ; the first time 4 heads occur in sequence S_4

Do 5000 repetitions (samples) to find the frequency distribution of the items recorded (random variable).

2. THEORETICAL ANALYSIS

Tossing of a coin multiple times is a Bernoulli Experiment. There are only two possible outcomes :

- Success - Occurrence of Head
- Failure - Occurrence of Tail

Coin tossing is simulated using Matlab. If we say that X denotes the random variable which denotes the total number of heads in an experiment, then,

$$X = \{1, 2, \cdots, 100\} \quad \text{for 100 coin tosses}$$

and,

$$P(X = x) = \binom{100}{x} p^x (1-p)^{1-x}$$

Also, let Random Variable Y represent the number of tosses until the first head occurs.

$$Y = \{ H, TH, TTH, TTTH, \ldots \}$$
$$= \{ 1, 2, 3, 4, \ldots \}$$

It can be seen that if $Y = 4$, then there are 3 cases of failures. In general, if $Y = y$, then there are $y-1$ failures. Hence, Y can be modeled as a geometric distribution. The probability distribution of Y is given by

$$P(Y = y) = (1-p)^{y-1} \cdot p.$$

3. SIMULATION METHODOLOGY

Coin tossing is simulated by generating a random sequence of 0's and 1's. 1 indicates occurrence of head whereas 0, on the other hand, indicates tail. The number of heads in a coin tossing experiment, is the total number of 1's in the random sequence.

In order to find longest run of heads, we consider the random sequence as a digital sequence / signal consisting of Highs (sequence of heads) and Lows (sequence of Tails). The longest run of heads is the longest duration for which the signal stays 'High'. The high duration of the signal is evaluated as the duration between the rising and falling edge of the random sequence.

The number of coin tosses until the N^{th} Head occurs

is calculated by counting the number of 0s and 1s until N^{th} 1 occurs

The number of coin tosses until N Heads occur in sequence is evaluated by generating a random sequence and then counting the length of random sequence until the longest run of Heads is N

4. EXPERIMENTS & RESULTS

The coin tossing experiment can be easily simulated using the randi function in MATLAB

For part A, answers:

Number of heads : 54
Longest run of heads : 5

For part B, answers:

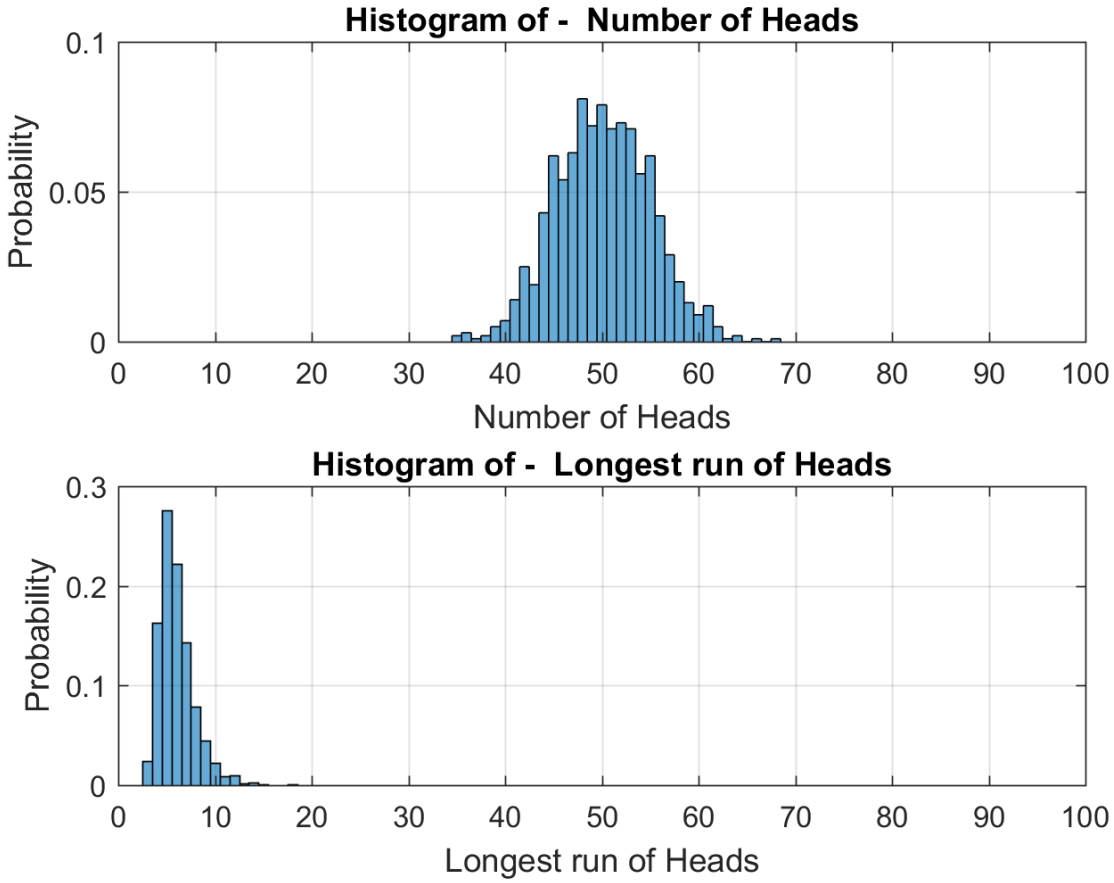
Number of tosses for 1 Head : 3
Number of tosses for 2 Head : 4
Number of tosses for 3 Head : 5
Number of tosses for 4 Head : 11

and,

Number of tosses until 2 consecutive Heads occur : 7
Number of tosses until 3 consecutive Heads occur : 43
Number of tosses until 4 consecutive Heads occur : 67

If we plot the histogram for 10000 repetitions for number of heads and probability, we can see that the histogram almost fits a gaussian distribution.

Similarly, in the case of Longest run of Heads, if we plot the histogram, we can see that it almost fits the gaussian distribution again.



5. SOURCE CODES

Part A:

```
clc;
clear all;
close all;

% Number of coin tosses
N = 100;

% Generate a uniformly distributed random vector of 1's and 0's
trial = randi([0,1],[1,N]);

% Evaluate number of heads and the longest run of heads
[n_heads, n_head_max] = number_of_heads(trial);

% Display the output on command window
```

```
fprintf('Number of heads : %d\n',n_heads);
fprintf('Longest run of heads : %d\n',n_head_max);
```

And the function is given by :

```
function [ n_heads, n_heads_max ] = number_of_heads( trial )
% This function helps to count the number of heads in an experiment.
% It also counts the maximum number of heads that occured
consecutively
%
% input arguements:
%     trial: A random sequence of 1's and 0's
%
% output arguements:
%     n_heads: Number of heads in the random sequence
%     n_heads_max: max number of consecutive 1's (heads) in the
sequence

% Count the number of 1's (heads) from trial
n_heads = sum(trial==1);

% Find the longest sequence of consecutive 1's (heads)
trial_pad = [0, trial, 0];
rising_edge = find(diff(trial_pad)==1);
falling_edge = find(diff(trial_pad)==-1);

n_heads_max = max(falling_edge - rising_edge);

if (isempty(n_heads_max));
    n_heads_max = 0;
end

end
```

For plotting, the code is:

```
% PLOTTING DISTRIBUTIONS

clc;
clear;
close all;

N = 100;
n_trials = 1000;

for m = 1:n_trials
    trial = randi([0,1],[1,N]);
    [n_heads(m), n_head_max(m)] = number_of_heads(trial);
end
```

```
figure('Name', 'Project 0 (Part A)');
subplot(2,1,1);
histogram(n_heads, 'Normalization','probability');
xlim([0 100]);
grid on;
title('Histogram of -  Number of Heads');
xlabel('Number of Heads');
ylabel('Probability');

subplot(2,1,2);
histogram(n_head_max, 'Normalization','probability');
xlim([0 100]);
grid on;
title('Histogram of -  Longest run of Heads');
xlabel('Longest run of Heads');
ylabel('Probability');
```

Part B:

```
clc;
clear;
close all;

% Number of tosses until nth head occurs
n_toss = zeros(1,4);
for k = 1:4
    n_toss(k) = toss_count(k);
    fprintf('Number of tosses for %d Head: %d\n',k,n_toss(k));
end

fprintf('\n');

% Number of tosses until n consecutive heads occur
for k = 2:4
    fprintf('Number of tosses until %d Consecutive Heads occur:
%d\n',k,toss_sequence(k));
end
```

And the first function is given by :

```
function [ t_cnt ] = toss_count ( heads )
% This function counts the number of tosses until n heads occur
%
% input arguement:
%       n_heads: Number of heads to be counted
%
% output arguement:
%        t_cnt: Number of tosses until n_heads occur

n_heads = 0;
```

```
t_cnt = 0;

while(n_heads ~= heads)
    if (randi([0 1]))
        n_heads = n_heads + 1;
    end

    t_cnt = t_cnt + 1;
end

end
```

And the second function is given by :

```
function [ t_cnt ] = toss_sequence ( heads )
% This function counts the number of tosses until n heads occur in
sequence
%
% input arguement:
%       n_heads: Number of heads to be counted
%
% output arguement:
%        t_cnt: Number of tosses until n_heads occur

n_heads_seq = 0;
t_cnt = 0;
sequence = [];

while(n_heads_seq ~= heads)
    sequence = [sequence randi([0 1])];
    [n_heads, n_heads_seq] = number_of_heads(sequence);
    t_cnt = t_cnt + 1;
end

end
```