

EE 511
Fall 2018
Prof. John Silvester

Project 0 - Coin Toss Experiment

Pavan Athreya Narasimha Murthy
USC ID: 9129210968
E-mail: pavanatn@usc.edu

A) Simulate tossing a coin 100 times and record

1. The number of heads NH
2. The length of the longest run of heads LH

Description

A coin toss is simulated by using the “rand” function in Matlab and checking to see if the value is less than 0.5 assuming that this is a fair coin.

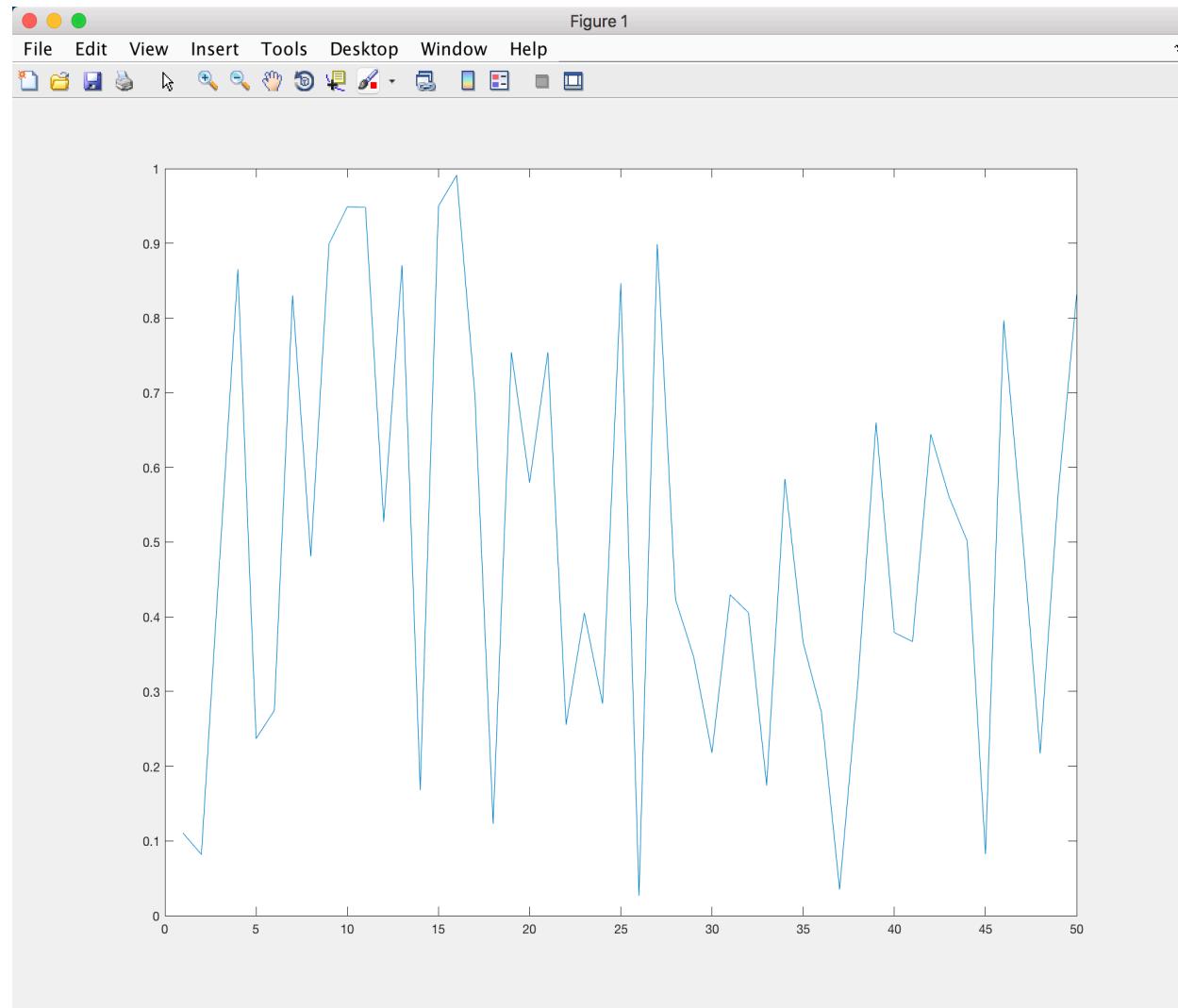
The user is asked to enter the number of tosses required to perform in the experiment.

For every value of the sample, the code tries to check for heads and all keeps track of the consecutive head count.

In the end, the results are displayed in the command window.

Data:

The plot shows the random sample data used for the experiment.



Name	Value
list_of_consecutive_heads	<i>1x50 double</i>
longest_heads	6
number_heads	26
number_toss	50
randomSamples	<i>1x50 double</i>
sample	50
temp_heads	1

Result:

Command Window

```
Enter the number of tosses: 50
Number of heads:
26

Length of the longest run:
6

fx >>
```

Commented Source Code:

```
%Name: Pavan Athreya Narasimha Murthy
%USC ID: 9129210968
%E-mail: pavanatn@usc.edu
%Ph: +1(323)-684 5715
%Term: Fall 2018
%Course: EE511
%Professor: John Silvester

%Clear the variables and command window
clear all;
clc;

%Get number of coin flips to be performed from the user
number_toss = input('Enter the number of tosses: ');

%Initialize variable to keep tracks of the results needed
number_heads = 0;
longest_heads = 0;
temp_heads = 1;
list_of_consecutive_heads = zeros(1,number_toss);

%Random values within 0 and 1 for our coin flip experiments
randomSamples = rand(1,number_toss);

%Looping the random values to get our experiment result
for sample = 1:number_toss
    %For this experiment we are considering <= 0.5 as Head and > 0.5 as Tails
    if(randomSamples(sample) <= 0.5)
        number_heads = number_heads + 1;
        if(sample < number_toss)
            if(randomSamples(sample) < 0.5 && randomSamples(sample + 1) < 0.5)
                %calculating the consecutive head sequence
                temp_heads = temp_heads + 1;
            else
                list_of_consecutive_heads(1,sample) = temp_heads;
                temp_heads = 1;
            end
        end
    end
end

%Get the longest head sequence from the array with head sequence count
longest_heads = max(list_of_consecutive_heads);

%display all the results needed
disp('Number of heads: ');
disp(number_heads);
disp('Length of the longest run: ');
disp(longest_heads);
```

*B) Simulate repeatedly tossing a coin and record the number of tosses until: the first head occurs S1;
 the first time 2 heads in sequence occurs S2;
 the first time 3 heads in sequence occurs S3;
 the first time 4 heads in sequence occurs S4
 e.g. if the observed sequence is: TTHTHTHHTTHHTHHH
 you would record S1 = 3, S2 = 8, S3 = 15, S4 = 16*

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

Description:

The same coin toss is modified to run indefinitely until 4 consecutive heads are found.

This function is called 5000 times to check the number of runs to get the sequence required.

A coin toss is simulated by using the “rand” function in Matlab and checking to see if the value is less than 0.5 assuming that this is a fair coin.

For every value of the sample, the code tries to check for heads and all keeps track of the consecutive head count.

In the end, the results are displayed in Plots of the occurrences and iterations which are shown in the results column.

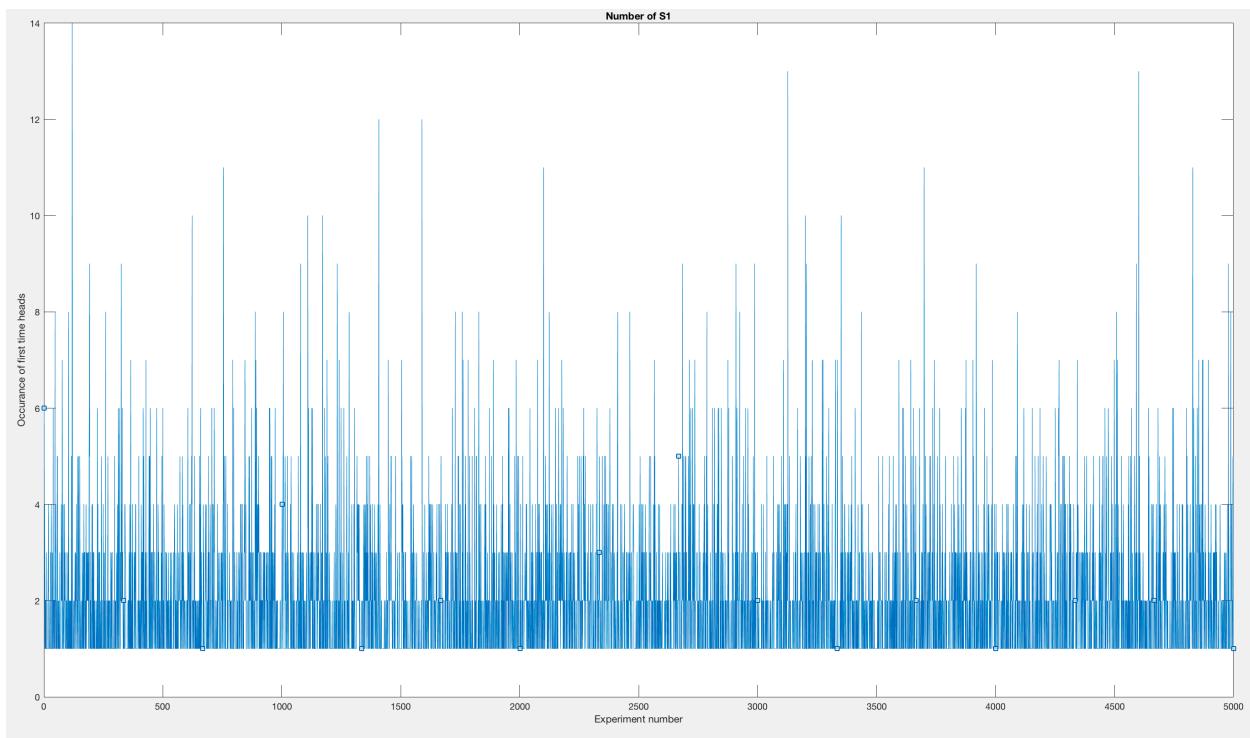
Data:

Name	Value
exp_number	5000
heads_numbers	1x5000 double
iterations	1x5000 double
s1s	1x5000 double
s2s	1x5000 double
s3s	1x5000 double
s4s	1x5000 double

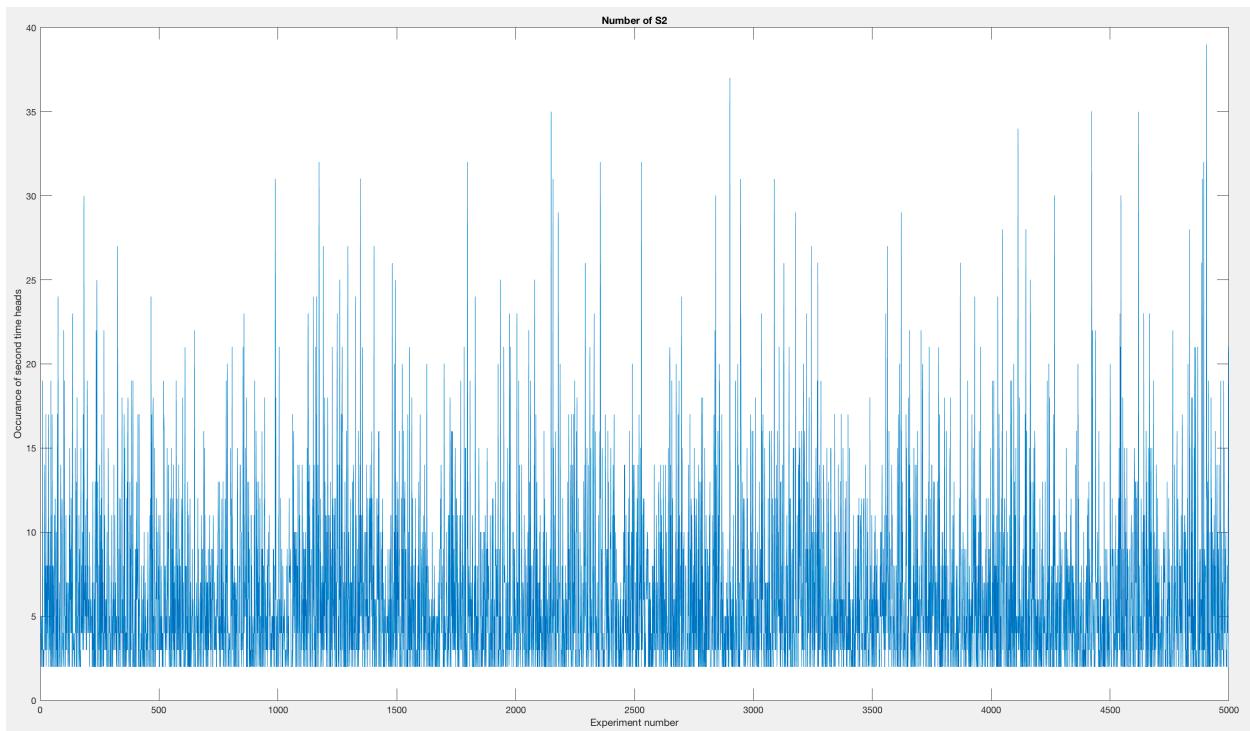
Data is too large to be shown here. The plot below shows all the variables above which is a result of the simulation which ran 5000 times

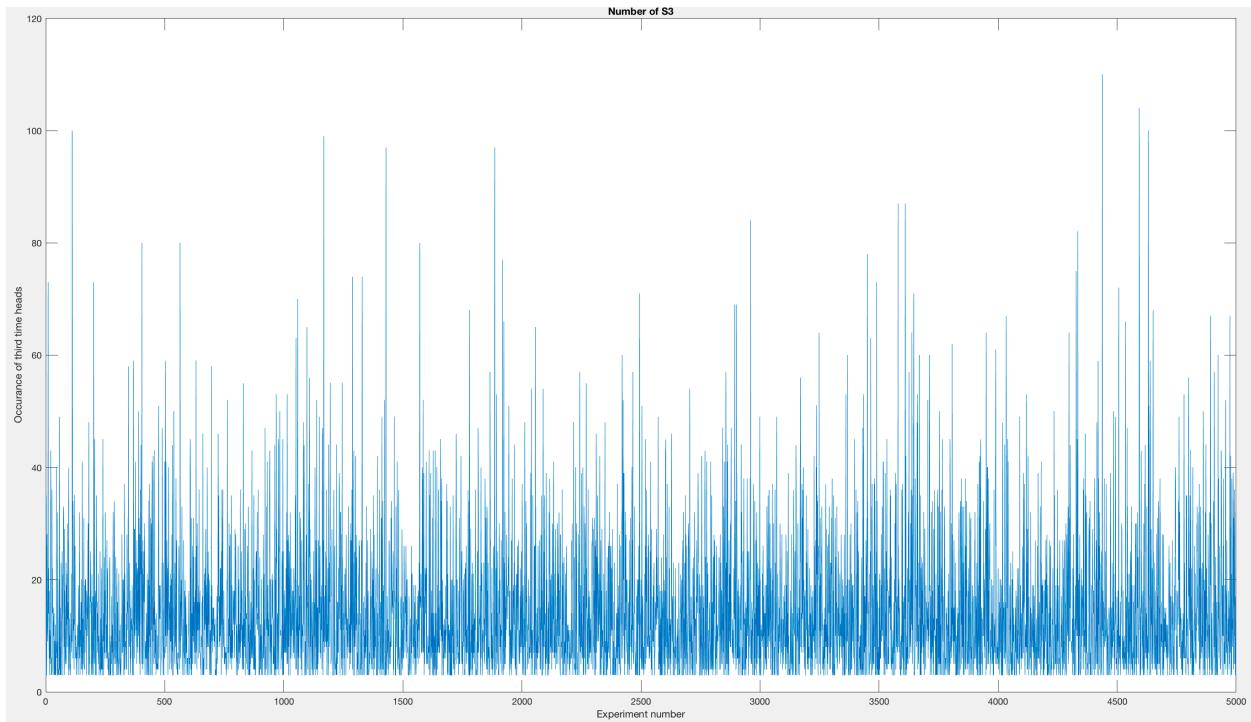
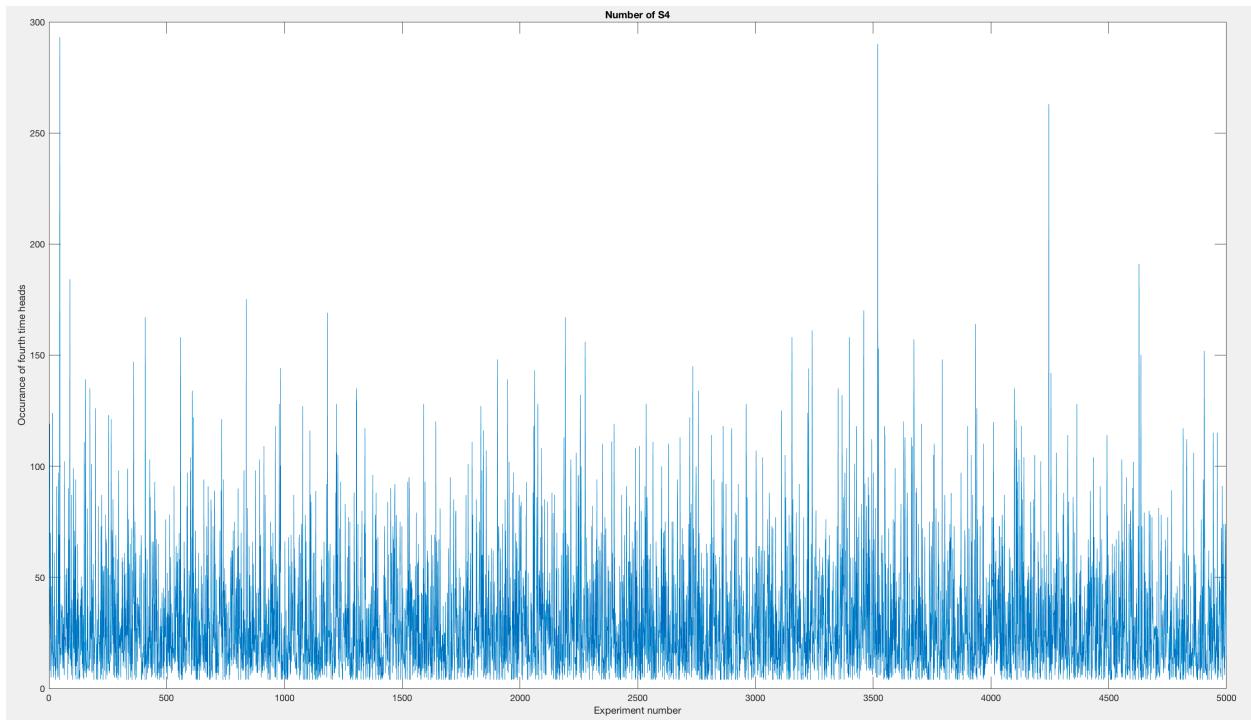
Result:

S1:

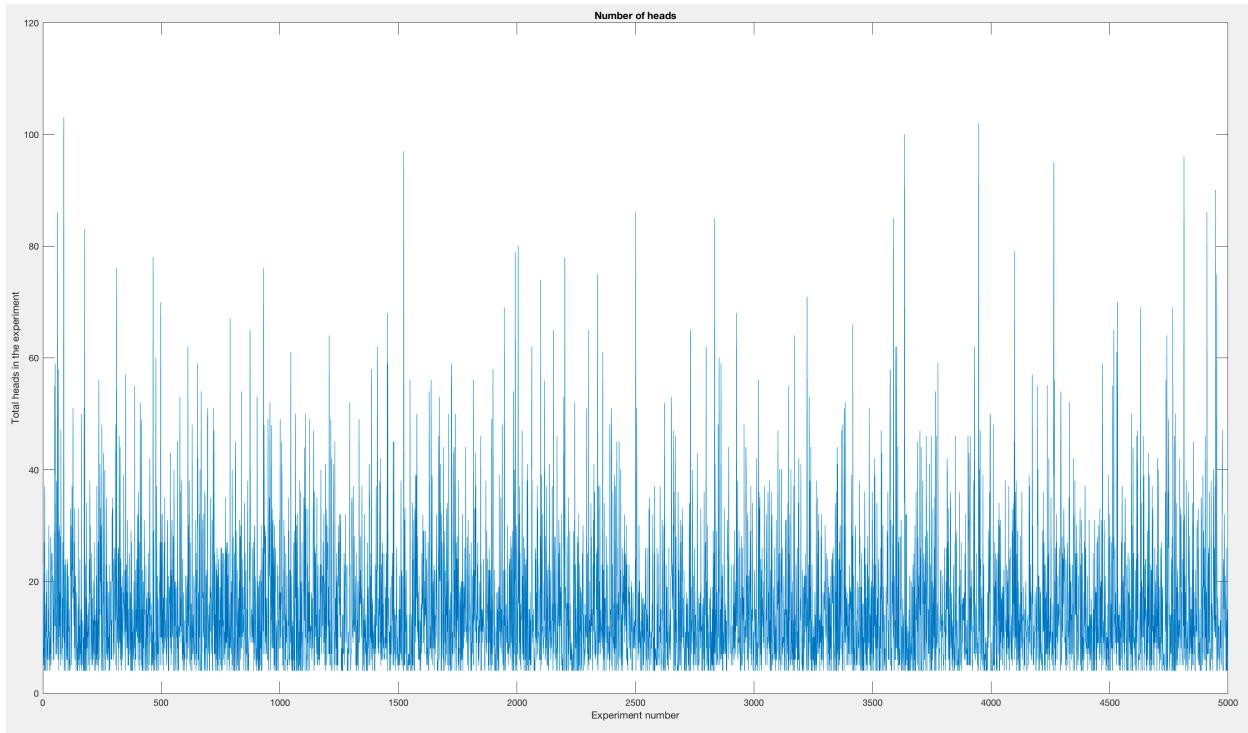


S2:

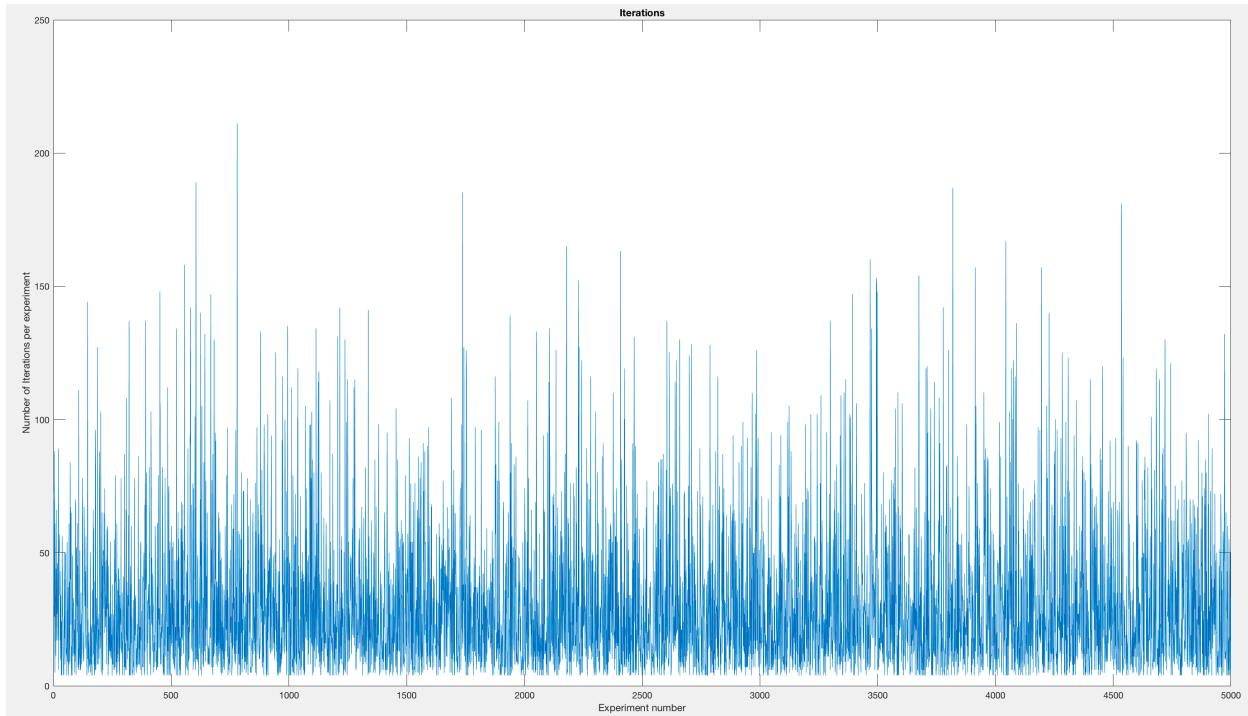


S3:**S4:**

Number of heads:



Number of runs per iteration to get the required sequence:



Commented Source Code:

```
%Name: Pavan Athreya Narasimha Murthy
%USC ID: 9129210968
%E-mail: pavanatn@usc.edu
%Ph: +1(323)-684 5715
%Term: Fall 2018
%Course: EE511
%Professor: John Silvester

%Clear the variables and command window
clear all;
clc;

%Global variables for getting the result of the experiments that runs 5000 times
global s1s;% = zeros(1,5000);
global s2s;% = zeros(1,5000);
global s3s;% = zeros(1,5000);
global s4s;% = zeros(1,5000);
%global sequences;% = ['AA'];
global heads_numbers;% = zeros(1,5000);
global iterations;% = zeros(1,5000);

%Function call to run the experiment 5000 times
for exp_number = 1:5000
    GenerateSequence(exp_number);
end

%Plotting all the six results captured
subplot(3,2,1);
plot(s1s);
title('Number of S1');
xlabel('Experiment number');
ylabel('Occurrence of first time heads');
subplot(3,2,2);
plot(s2s);
title('Number of S2');
xlabel('Experiment number');
ylabel('Occurrence of second time heads');
subplot(3,2,3);
plot(s3s);
title('Number of S3');
xlabel('Experiment number');
ylabel('Occurrence of third time heads');
subplot(3,2,4);
plot(s4s);
title('Number of S4');
xlabel('Experiment number');
ylabel('Occurrence of fourth time heads');
```

```

subplot(3,2,5);
plot(heads_numbers);
title('Number of heads');
xlabel('Experiment number');
ylabel('Total heads in the experiment');
subplot(3,2,6);
plot(iterations);
title('Iterations');
xlabel('Experiment number');
ylabel('Number of Iterations per experiment');

%Function to perform the experiment
function experiment = GenerateSequence(expNum)
global s1s;% = zeros(1,5000);
global s2s;% = zeros(1,5000);
global s3s;% = zeros(1,5000);
global s4s;% = zeros(1,5000);
%global sequences;% = ['AA'];
global heads_numbers;% = zeros(1,5000);
global iterations;% = zeros(1,5000);
number_heads = 0;
locations = zeros(1,4);
temp_heads = 0;
sequence = ['A'];
iterationCount = 0;
%Iterate until s4 is found
while(1)
    iterationCount = iterationCount + 1;
    value = rand;
    if(value <= 0.5)
        sequence(1,iterationCount) = 'H';
        if(0 == locations(1,1))
            %s1
            locations(1,1) = iterationCount;
        end
        number_heads = number_heads + 1;
        temp_heads = temp_heads + 1;
        switch(temp_heads)
            case {2}
                if(0 == locations(1,2))
                    %s2
                    locations(1,2) = iterationCount;
                end
            case {3}
                if(0 == locations(1,3))
                    %s3
                    locations(1,3) = iterationCount;
                end
            case {4}

```

```
if(0 == locations(1,4))
    %s4
    locations(1,4) = iterationCount;
end
break;
end
else
    sequence(1,iterationCount) = 'T';
    temp_heads = 0;
end
end
s1s(1,expNum) = locations(1,1);
s2s(1,expNum) = locations(1,2);
s3s(1,expNum) = locations(1,3);
s4s(1,expNum) = locations(1,4);
% sequences(1,expNum) = sequence;
heads_numbers(1,expNum) = number_heads;
iterations(1,expNum) = iterationCount;
end
```