A close up of text on a white background

Description generated with high confidence

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