bc

**MEMO Number** CMPE320-S21-PROJ1 CODE

**DATE:** 02/27/2021

**TO:** EFC LaBerge and CMPE320 classmates

**FROM:** Nem Negash

**SUBJECT: Project 1 Code**

# matlab project code

## PMF for a single fair die

function diePMF()

%120 trials

figure(1)

list120 = randi(6,1,120);

hist = histogram(list120,'Normalization', 'probability');

xlabel('Role of the fair die');

ylabel('Probability of occurances');

title('PMF for Occurances for 120 trials');

M = mean(list120);

V = var(list120);

fprintf('The sample mean for 120 trials: %.3f\n',M);

fprintf('The sample variance for 120 trials: %.3f\n\n',V);

%1200 trials

figure(2)

list1200 = randi(6,1,1200);

hist = histogram(list1200,'Normalization', 'probability');

xlabel('Role of the fair die');

ylabel('Probability of occurances');

title('PMF for Occurances for 1,200 trials');

M = mean(list1200);

V = var(list1200);

fprintf('The sample mean for 1,200 trials: %.3f\n',M);

fprintf('The sample variance for 1,200 trials: %.3f\n\n',V);

%12000 trials

figure(3)

list12000 = randi(6,1,12000);

hist = histogram(list12000,'Normalization', 'probability');

xlabel('Role of the fair die');

ylabel('Probability of occurances');

title('PMF for Occurances for 12,000 trials');

M = mean(list12000);

V = var(list12000);

fprintf('The sample mean for 12,000 trials: %.3f\n',M);

fprintf('The sample variance for 12,000 trials: %.3f\n\n',V);

%120000 trials

figure(4)

list120000 = randi(6,1,120000);

hist = histogram(list120000,'Normalization', 'probability');

xlabel('Role of the fair die');

ylabel('Probability of occurances');

title('PMF for Occurances for 120,000 trials');

M = mean(list120000);

V = var(list120000);

fprintf('The sample mean for 120,000 trials: %.3f\n',M);

fprintf('The sample variance for 120,000 trials: %.3f\n\n',V);

end

## PMF for binary strings

function binarystrings()

%for 20 trials

figure(1)

trial05\_20 = trial(0.5,20);

disp(trial05\_20);

hist = histogram(trial05\_20,'Normalization', 'probability');

hist.BinEdges = [0:100];

hist.NumBins = 100;

xlabel('Indices');

ylabel('Probability of first occurances at given index');

title('Indices vs. Number of first occurances for 20 trials with p = 0.5');

M = mean(trial05\_20);

V = var(trial05\_20);

fprintf('Mean for 20 trials @ p = 0.5 = %0.4f\n',M);

fprintf('Variance for 20 trials @ p = 0.5 = %0.4f\n',V);

figure(2)

trial09\_20 = trial(0.9,20);

hist = histogram(trial09\_20,'Normalization', 'probability');

hist.BinEdges = [0:100];

hist.NumBins = 100;

xlabel('Indices');

ylabel('Probability of first occurances at given index');

title('Indices vs. Number of first occurances for 20 trials with p = 0.9');

M = mean(trial09\_20);

V = var(trial09\_20);

fprintf('Mean for 20 trials @ p = 0.9 = %0.4f\n',M);

fprintf('Variance for 20 trials @ p = 0.9 = %0.4f\n',V);

figure(3)

trial01\_20 = trial(0.1,20);

hist = histogram(trial01\_20,'Normalization', 'probability');

hist.BinEdges = [0:100];

hist.NumBins = 100;

xlabel('Indices');

ylabel('Probability of first occurances at given index');

title('Indices vs. Number of first occurances for 20 trials with p = 0.1');

M = mean(trial01\_20);

V = var(trial01\_20);

fprintf('Mean for 20 trials @ p = 0.1 = %0.4f\n',M);

fprintf('Variance for 20 trials @ p = 0.1 = %0.4f\n\n',V);

%for 200 trials

figure(4)

trial05\_200 = trial(0.5,200);

hist = histogram(trial05\_200,'Normalization', 'probability');

hist.BinEdges = [0:100];

hist.NumBins = 100;

xlabel('Indices');

ylabel('Probability of first occurances at given index');

title('Indices vs. Number of first occurances for 200 trials with p = 0.5');

M = mean(trial05\_200);

V = var(trial05\_200);

fprintf('Mean for 200 trials @ p = 0.5 = %0.4f\n',M);

fprintf('Variance for 200 trials @ p = 0.5 = %0.4f\n',V);

figure(5)

trial09\_200 = trial(0.9,200);

hist = histogram(trial09\_200,'Normalization', 'probability');

hist.BinEdges = [0:100];

hist.NumBins = 100;

xlabel('Indices');

ylabel('Probability of first occurances at given index');

title('Indices vs. Number of first occurances for 200 trials with p = 0.9');

M = mean(trial09\_200);

V = var(trial09\_200);

fprintf('Mean for 200 trials @ p = 0.9 = %0.4f\n',M);

fprintf('Variance for 200 trials @ p = 0.9 = %0.4f\n',V);

figure(6)

trial01\_200 = trial(0.1,200);

hist = histogram(trial01\_200,'Normalization', 'probability');

hist.BinEdges = [0:100];

hist.NumBins = 100;

xlabel('Indices');

ylabel('Probability of first occurances at given index');

title('Indices vs. Number of first occurances for 200 trials with p = 0.1');

M = mean(trial01\_200);

V = var(trial01\_200);

fprintf('Mean for 200 trials @ p = 0.1 = %0.4f\n',M);

fprintf('Variance for 200 trials @ p = 0.1 = %0.4f\n\n',V);

%for 2000 trials

figure(7)

trial05\_2000 = trial(0.5,2000);

hist = histogram(trial05\_2000,'Normalization', 'probability');

hist.BinEdges = [0:100];

hist.NumBins = 100;

xlabel('Indices');

ylabel('Probability of first occurances at given index');

title('Indices vs. Number of first occurances for 2,000 trials with p = 0.5');

M = mean(trial05\_2000);

V = var(trial05\_2000);

fprintf('Mean for 2,000 trials @ p = 0.5 = %0.4f\n',M);

fprintf('Variance for 2,000 trials @ p = 0.5 = %0.4f\n',V);

figure(8)

trial09\_2000 = trial(0.9,2000);

hist = histogram(trial09\_2000,'Normalization', 'probability');

hist.BinEdges = [0:100];

hist.NumBins = 100;

xlabel('Indices');

ylabel('Probability of first occurances at given index');

title('Indices vs. Number of first occurances for 2,000 trials with p = 0.9');

M = mean(trial09\_2000);

V = var(trial09\_2000);

fprintf('Mean for 2,000 trials @ p = 0.9 = %0.4f\n',M);

fprintf('Variance for 2,000 trials @ p = 0.9 = %0.4f\n',V);

figure(9)

trial01\_2000 = trial(0.1,2000);

hist = histogram(trial01\_2000,'Normalization', 'probability');

hist.BinEdges = [0:100];

hist.NumBins = 100;

xlabel('Indices');

ylabel('Probability of first occurances at given index');

title('Indices vs. Number of first occurances for 2,000 trials with p = 0.1');

M = mean(trial01\_2000);

V = var(trial01\_2000);

fprintf('Mean for 2,000 trials @ p = 0.1 = %0.4f\n',M);

fprintf('Variance for 2,000 trials @ p = 0.1 = %0.4f\n\n',V);

%for 200,000 trials

figure(10)

trial05\_200000 = trial(0.5,200000);

hist = histogram(trial05\_200000,'Normalization', 'probability');

hist.BinEdges = [0:100];

hist.NumBins = 100;

xlabel('Indices');

ylabel('Probability of first occurances at given index');

title('Indices vs. Number of first occurances for 200,000 trials with p = 0.5');

M = mean(trial05\_200000);

V = var(trial05\_200000);

fprintf('Mean for 200,000 trials @ p = 0.5 = %0.4f\n',M);

fprintf('Variance for 200,000 trials @ p = 0.5 = %0.4f\n',V);

figure(11)

trial09\_200000 = trial(0.9,200000);

hist = histogram(trial09\_200000,'Normalization', 'probability');

hist.BinEdges = [0:100];

hist.NumBins = 100;

xlabel('Indices');

ylabel('Probability of first occurances at given index');

title('Indices vs. Number of first occurances for 200,000 trials with p = 0.9');

M = mean(trial09\_200000);

V = var(trial09\_200000);

fprintf('Mean for 200,000 trials @ p = 0.9 = %0.4f\n',M);

fprintf('Variance for 200,000 trials @ p = 0.9 = %0.4f\n',V);

figure(12)

trial01\_200000 = trial(0.1,200000);

hist = histogram(trial01\_200000,'Normalization', 'probability');

hist.BinEdges = [0:100];

hist.NumBins = 100;

xlabel('Indices');

ylabel('Probability of first occurances at given index');

title('Indices vs. Number of first occurances for 200,000 trials with p = 0.1');

M = mean(trial01\_200000);

V = var(trial01\_200000);

fprintf('Mean for 200,000 trials @ p = 0.1 = %0.4f\n',M);

fprintf('Variance for 200,000 trials @ p = 0.1 = %0.4f\n',V);

end

function o\_list = trial(p,m)

%first 20

list = [];

for x = 1:m

temp = [];

for y = 1:100

z = rand;

if z < p

temp(y) = 1;

else

temp(y) = 0;

end

end

list = [list;temp];

end

%disp(list);

o\_list = [];

indx = 1;

while indx <= (m\*100)

if list(indx) == 1

o\_list = [o\_list,mod(indx,100)];

indx = indx +(100 - mod(indx,100));

end

indx = indx + 1;

end

end

## PDF for an exponentially distributed random variable

function PDF\_expo()

%10 trials

figure(1)

list = randx(1,10,0.5);

hist = histogram(list,'Normalization', 'pdf');

hist.BinWidth = 2;

hold on

x = [0:1:10];

y = 0.5 \* exp(-0.5\*x);

plot(x,y,'LineWidth',4);

xlabel('x');

ylabel('f(x)');

title('PDF of exponentially distributed random variable for 10 trials');

legend('Histogram','Analytical')

M = mean(list);

V = var(list);

fprintf('Mean for 10 trials = %0.4f\n',M);

fprintf('Variance for 10 trials = %0.4f\n',V);

%1000 trials

figure(2)

list = randx(1,1000,0.5);

hist = histogram(list,'Normalization', 'pdf');

hold on

x = [0:1:10];

y = 0.5 \* exp(-0.5\*x);

plot(x,y,'LineWidth',4);

xlabel('x');

ylabel('f(x)');

title('PDF of exponentially distributed random variable for 1,000 trials');

legend('Histogram','Analytical')

M = mean(list);

V = var(list);

fprintf('Mean for 1,000 trial s= %0.4f\n',M);

fprintf('Variance for 1,000 trials = %0.4f\n',V);

%100000 trials

figure(3)

list = randx(1,100000,0.5);

hist = histogram(list,'Normalization', 'pdf');

hold on

x = [0:1:10];

y = 0.5 \* exp(-0.5\*x);

plot(x,y,'LineWidth',4);

xlabel('x');

ylabel('f(x)');

title('PDF of exponentially distributed random variable for 100,000 trials');

legend('Histogram','Analytical')

M = mean(list);

V = var(list);

fprintf('Mean for 100,000 trials = %0.4f\n',M);

fprintf('Variance for 100,000 trials = %0.4f\n',V);

end

function rexp = randx(n,k,lambda)

%

%function rexp = randx(n,k,lambda)

% Generates samples of an exponentially distributed random variable with

% parameter lambda.

% Calling parameters

% n: number of columns in output array rexp

% m: number of rows in output array rexp

% lambda: exponential distribution parameter, lambda > 0.

% Returned parameters

% rexp an n x k array containing independent samples from an

% exponential distribution with pdf f(x) = lambda exp( -lambda\*x)

%

% Help comments updated 2/13/2021 EFCL

% Original code EFCL ~1989

%

Z = rand(n,k); % compute a uniformly distributed random variable

% Now treating the Z value as the CDF of the desired exponential random variable,

% invert the CDF ( F(x) = 1 - exp(-lambda\*x) ) to find the equivalent x

% value. exp(-lambda x) = 1 - F(x) = 1 - Z

% -lambda x = log(1 - Z)

% x = -log(1-Z)/lambda

rexp=zeros(n,k); % establish the memory

rexp=-log(1-Z)/lambda; % invert the CDF.

end

## PDF for a normal or Gaussian distributed random variable

function part2\_4()

%10 trial

figure(1)

list = randn(1,10);

histogram(list,'Normalization', 'pdf');

hold on

x = [-3:0.0001:3];

y = ((2\*pi).^-0.5)\*exp((-x.^2)/2);

plot(x,y,'LineWidth',4);

xlabel('x');

ylabel('f(x)');

title('PDF for normal distributed random variable for 10 trials');

legend('Histogram','Analytical')

M = mean(list);

V = var(list);

fprintf('Mean for 10 trials= %0.4f\n',M);

fprintf('Variance for 10 trials = %0.4f\n',V);

%1000 trials

figure(2)

list = randn(1,1000);

histogram(list,'Normalization', 'pdf');

hold on

x = [-3:0.0001:3];

y = ((2\*pi\*(1.^2)).^-0.5)\*exp((-x.^2)/2);

plot(x,y,'LineWidth',4);

xlabel('x');

ylabel('f(x)');

title('PDF for normal distributed random variable for 1,000 trials');

legend('Histogram','Analytical')

M = mean(list);

V = var(list);

fprintf('Mean for 1,000 trials= %0.4f\n',M);

fprintf('Variance for 1,000 trials= %0.4f\n',V);

%100000 trials

figure(3)

list = randn(1,100000);

histogram(list,'Normalization', 'pdf');

hold on

x = [-3:0.0001:3];

y = ((2\*pi).^-0.5)\*exp((-x.^2)/2);

plot(x,y,'LineWidth',4);

xlabel('x');

ylabel('f(x)');

title('PDF for normal distributed random variable for 100,000 trials');

legend('Histogram','Analytical')

M = mean(list);

V = var(list);

fprintf('Mean for 100,000 trials= %0.4f\n',M);

fprintf('Variance for 100,000 trials= %0.4f\n',V);

end

## PDF for a normal or Gaussian distributed random variable

function part2\_5()

%10 trial

figure(1)

list = (2\*randn(1,10)) + 1;

histogram(list,'Normalization', 'pdf');

hold on

x = [-8:0.0001:8];

y = ((8\*pi).^-0.5)\*exp(-((x-1).^2)/8);

plot(x,y,'LineWidth',4);

xlabel('x');

ylabel('f(x)');

title('PDF for normal distributed random variable for 10 trials', '?^2 = 4, m = 1');

legend('Histogram','Analytical')

M = mean(list);

V = var(list);

fprintf('Mean for 10 trials = %0.4f\n',M);

fprintf('Variance for 10 trials = %0.4f\n',V);

%1000 trials

figure(2)

list = (2\*randn(1,1000)) + 1;

histogram(list,'Normalization', 'pdf');

hold on

x = [-8:0.0001:8];

y = ((8\*pi).^-0.5)\*exp(-((x-1).^2)/8);

plot(x,y,'LineWidth',4);

xlabel('x');

ylabel('f(x)');

title('PDF for normal distributed random variable for 1,000 trials', '?^2 = 4, m = 1');

legend('Histogram','Analytical')

M = mean(list);

V = var(list);

fprintf('Mean for 1,000 trials = %0.4f\n',M);

fprintf('Variance for 1,000 trials = %0.4f\n',V);

%100000 trials

figure(3)

list = (2\*randn(1,100000)) + 1;

histogram(list,'Normalization', 'pdf');

hold on

x = [-8:0.0001:8];

y = ((8\*pi).^-0.5)\*exp(-((x-1).^2)/8);

plot(x,y,'LineWidth',4);

xlabel('x');

ylabel('f(x)');

title('PDF for normal distributed random variable for 100,000 trials', '?^2 = 4, m = 1');

legend('Histogram','Analytical')

M = mean(list);

V = var(list);

fprintf('Mean for 100,000 trials = %0.4f\n',M);

fprintf('Variance for 100,000 trials = %0.4f\n',V);

end

## Computing probabilities from the pdf

function part2\_6()

%Unscaled histogram

list = (2\*randn(1,1000000)) + 1;

figure(1)

hist1 = histogram(list);

hist1.BinWidth = 0.1;

x\_values1 = find(hist1.BinEdges >= 1.0 & hist1.BinEdges < 3.0);

pdf\_sum1 = sum(hist1.Values(x\_values1));

prob = pdf\_sum1 / 1000000;

fprintf('Unscaled Sample probability: %0.4f\n',prob);

%Scaled histogram

figure(2)

hist2 = histogram(list,'Normalization', 'pdf');

hist2.BinWidth = 0.1;

x\_values2 = find(hist2.BinEdges >= 1.0 & hist1.BinEdges < 3.0);

pdf\_sum2 = sum(hist2.Values(x\_values2));

numeric\_integ = pdf\_sum2 \* hist2.BinWidth;

fprintf('Scaled Sample probability: %0.4f\n',numeric\_integ);

%True PDF probability

f\_x = @(x) ((8\*pi).^-0.5)\*exp(-((x-1).^2)/8);

integ = integral(f\_x,1,3);

fprintf('True PDF probability: %0.4f\n',integ);

end