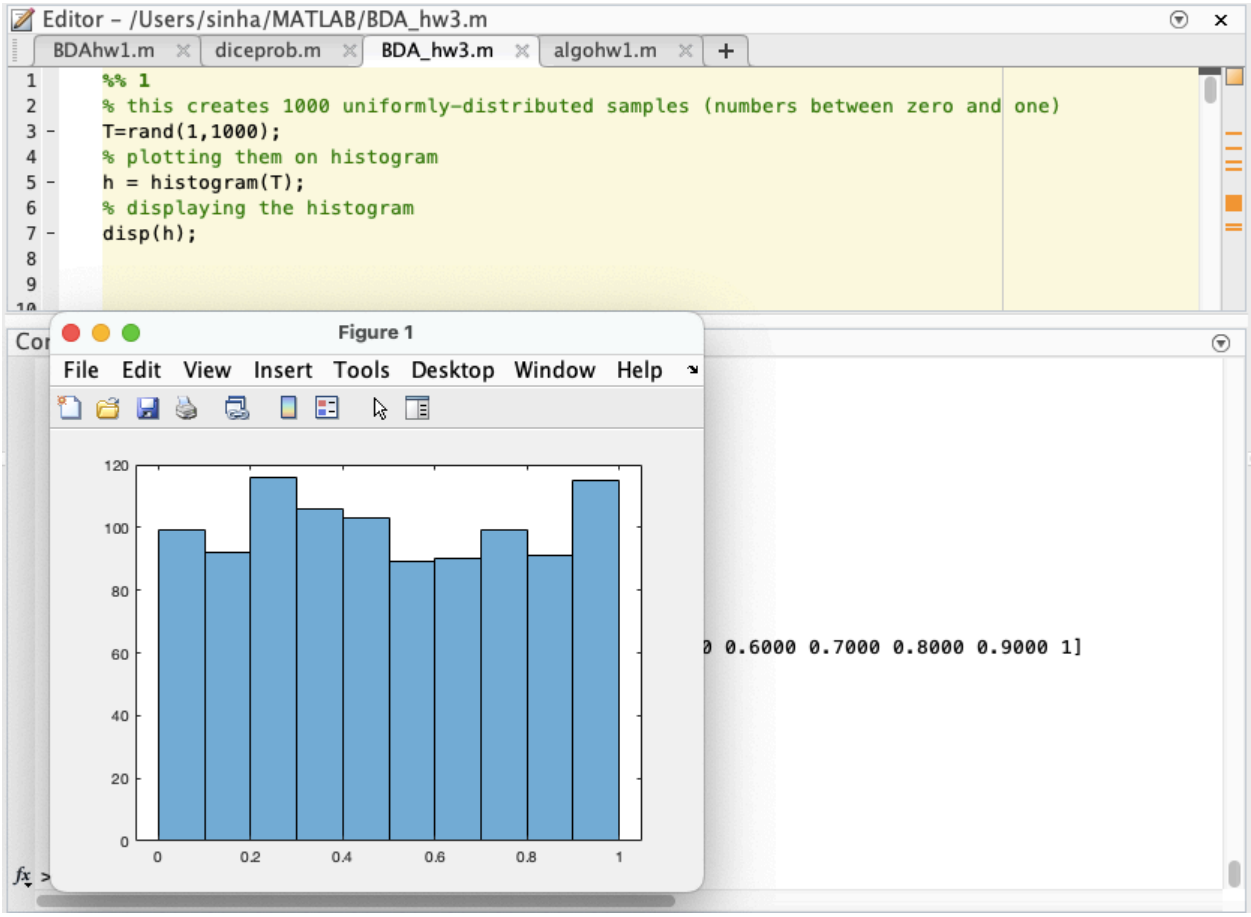
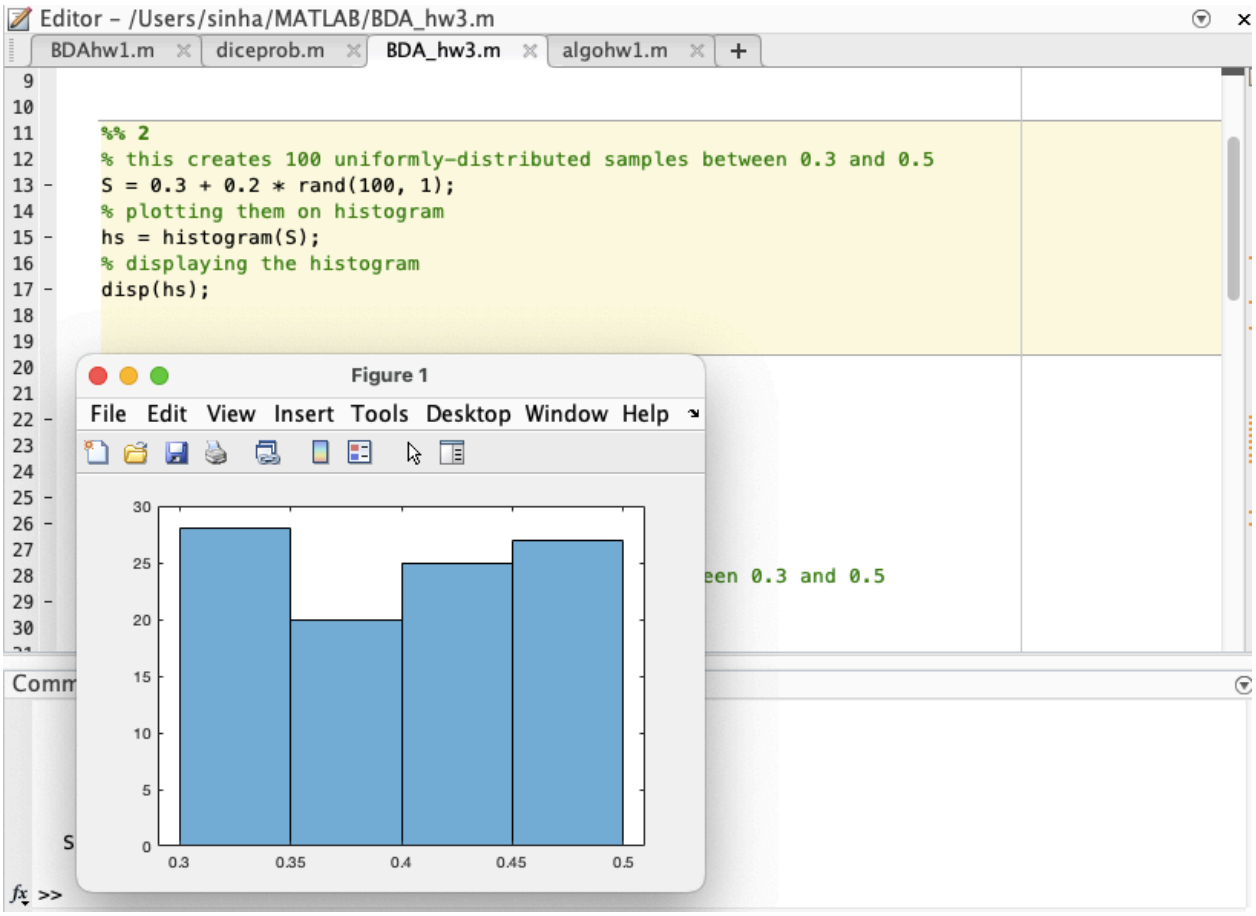
Bayesian Data Analysis. HW3p: Programming Homework #3
NAME : RAGINI RANI (ID: RR335214)

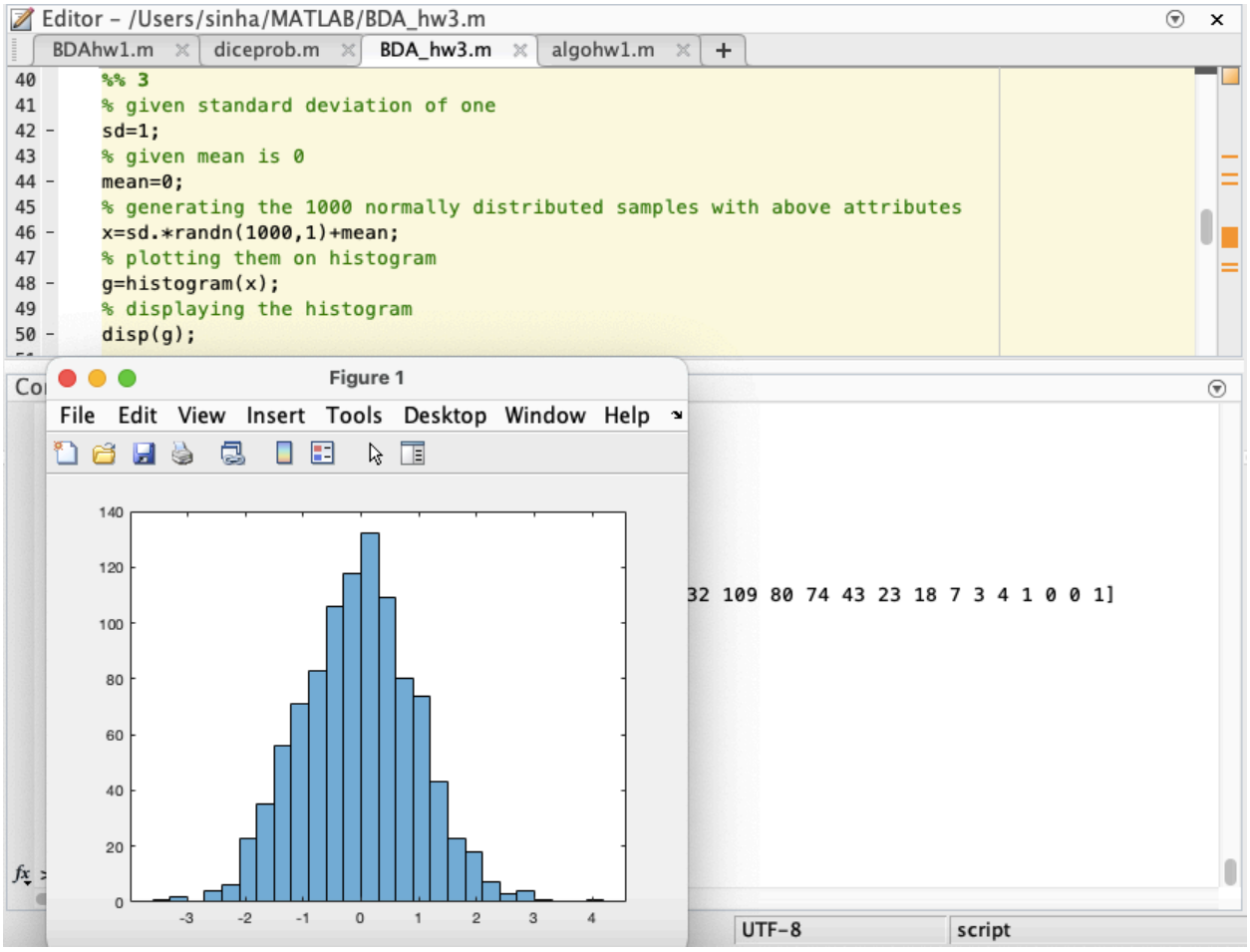
Q1.



Q2.


Editor - /Users/sinha/MATLAB/BDA_hw3.m
BDAhw1.m
diceprob.m
BDA_hw3.m
algohw1.m
%% 2
% this creates 100 uniformly-distributed samples between 0.3 and 0.5
S = 0.3 + 0.2 * rand(100, 1);
% plotting them on histogram
hs = histogram(S);
% displaying the histogram
disp(hs);
Figure 1
File Edit View Insert Tools Desktop Window Help
30
25
20
15
10
5
0
0.3
0.35
0.4
0.45
0.5
>>

Q3.


Editor - /Users/sinha/MATLAB/BDA_hw3.m
BDAhw1.m
diceprob.m
BDA_hw3.m
algohw1.m
40 %% 3
41 % given standard deviation of one
42 - sd=1;
43 % given mean is 0
44 - mean=0;
45 % generating the 1000 normally distributed samples with above attributes
46 - x=sd.*randn(1000,1)+mean;
47 % plotting them on histogram
48 - g=histogram(x);
49 % displaying the histogram
50 - disp(g);
Figure 1
File Edit View Insert Tools Desktop Window Help
140
120
100
80
60
40
20
0
-3 -2 -1 0 1 2 3 4
32 109 80 74 43 23 18 7 3 4 1 0 0 1]
UTF-8
script

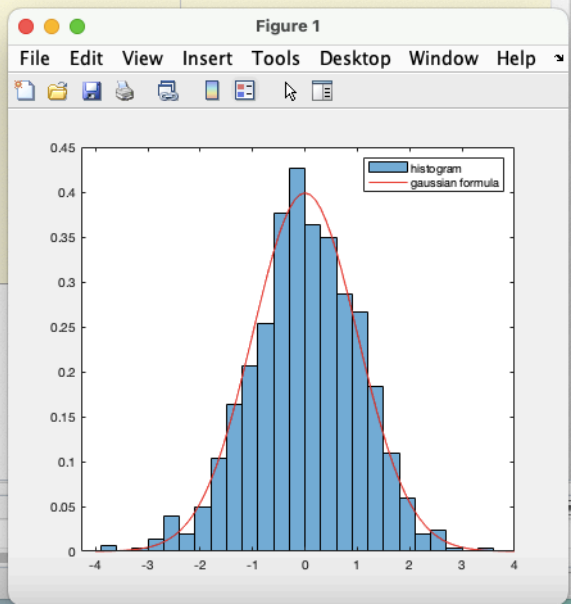
Q4.

BDAhw1.m | diceprob.m | BDA_hw3.m | algohw1.m | fun1.m | untitled2*

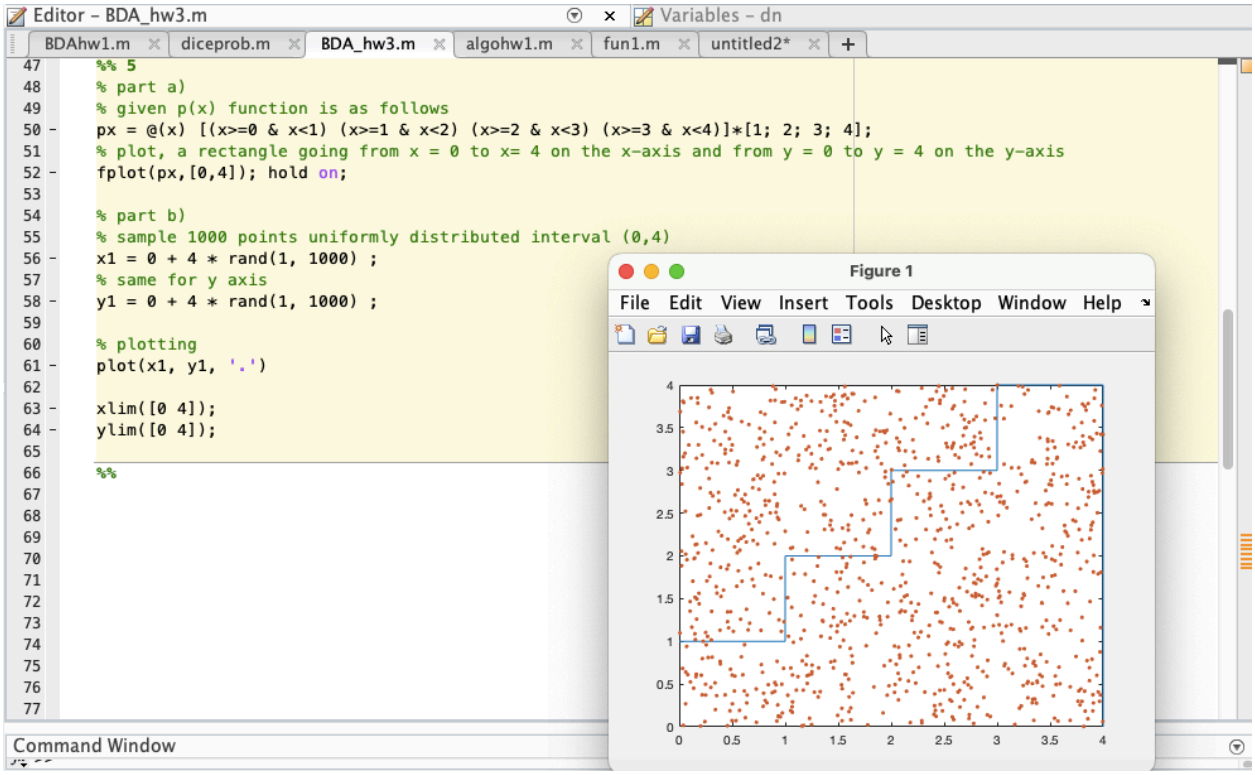
```
%% 4
% given standard deviation of one
sd=1;
% given mean is 0
mean=0;
% generating the 1000 normally distributed samples with above attributes
A = randn(1000,1)*sd + mean;
%plot histogram after normalization
histogram(A,'Normalization','pdf'); hold on;

%range of x to evaluate through gaussian formula
x = -4:0.1:4;
% Gaussian curve equation
gaussian = 1/sqrt(2*pi*sd^2)*exp(-1*(x-mean).^2./2*sd^2);
% plot of the gaussian curve
plot(x,gaussian,'r');
% labels of the plot
legend('histogram','gaussian formula')

%%
```

Command Window

fx >>



Q5.


Editor - BDA_hw3.m
Variables - dn
BDAhw1.m
diceprob.m
BDA_hw3.m
algohw1.m
fun1.m
untitled2*
47 %% 5
48 % part a)
49 % given p(x) function is as follows
50 - px = @(x) [(x>=0 & x<1) (x>=1 & x<2) (x>=2 & x<3) (x>=3 & x<4)]*[1; 2; 3; 4];
51 % plot, a rectangle going from x = 0 to x= 4 on the x-axis and from y = 0 to y = 4 on the y-axis
52 - fplot(px,[0,4]); hold on;
53
54 % part b)
55 % sample 1000 points uniformly distributed interval (0,4)
56 - x1 = 0 + 4 * rand(1, 1000) ;
57 % same for y axis
58 - y1 = 0 + 4 * rand(1, 1000) ;
59
60 % plotting
61 - plot(x1, y1, '.')
62
63 - xlim([0 4]);
64 - ylim([0 4]);
65
66 %%
67
68
69
70
71
72
73
74
75
76
77
Command Window
Figure 1
File Edit View Insert Tools Desktop Window Help
0 0.5 1 1.5 2 2.5 3 3.5 4