Zhiqian Zhou   Student Number: # 14303902

HW2

Question 1

a)

Sample x1, x2 are included in x in the form of [x1;x2], x is a 2*d matrix.

```
% Mahalanobis distance
function [ distance ] = mahalanobis(x, sigma, u)

[row_x,~] = size(x);
distance = sum((x - repmat(u,row_x,1)) * inv(sigma).*...
     (x - repmat(u,row_x,1)),2);

end
```

b)

Using the function above, the 'discriFunction' Function calculate the value of discriminate function.

```
% discriminant function
function [ discriValue ] = discriFunction(x, sigma, u, d, prior)

mahal = mahalanobis(x,sigma,u);
discriValue = -1/2*mahal - d/2*log(2*pi)- 1/2*log(det(sigma)) + log(prior);

end
```

c)

Compute mean and covariance.

```
% hw1 mean and covariance
function [ u, sigma ] = MeanCov(x)

[number,dimension] = size(x);
u = zeros(1,dimension);
sigma = zeros(dimension,dimension);

for i = 1:dimension
    for j = 1:number
        u(1,i) = u(1,i) + x(j,i)/number;
    end
end

for i=1:dimension
    for j=1:dimension
        sigma(i,j)=sum((x(:,i)-mean(x(:,i))).*(x(:,j)-mean(x(:,j))))/(number-1);
    end
end

end
```

Calculate the value of discriminate function using the function above, the result is listed below.

```
(1,3,2) is classified to class 1
(4,6,1) is classified to class 1
(7,-1,0) is classified to class 1
(-2,6,5) is classified to class 2
```

Question 2

a), b)

Generates the (say,1000) samples.

```
mu1 = [8,2];
mu2 = [2,8];
sigma = [4.1,0;0,2.8];
r1 = mvnrnd(mu1,sigma,1000);
r2 = mvnrnd(mu2,sigma,1000);
```

Calculate the scope using the mean and variance.

```
%scope
scop1_l = min(mu1(1,1),mu2(1,1))-sqrt(sigma(1,1))*4;
scop1_r = max(mu1(1,1),mu2(1,1))+sqrt(sigma(1,1))*4;
scop2_l = min(mu1(1,2),mu2(1,2))-sqrt(sigma(2,2))*4;
scop2_r = max(mu1(1,2),mu2(1,2))+sqrt(sigma(2,2))*4;
scope_x = scop1_l:0.2:scop1_r;
scope_y = scop2_l:0.2:scop2_r;
```

Derive the decision boundary.

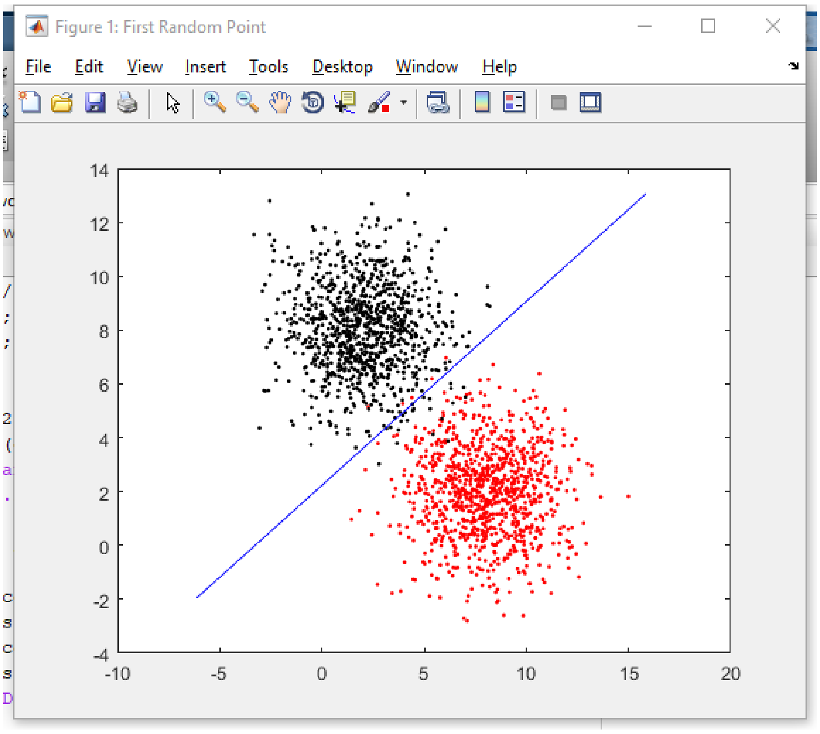
```
% solve the decision boundary
function [ g_solve ] = bound(mu1,mu2,sigma1,sigma2,prior1,prior2)
    syms x y;
    g_1 = (-0.5) * ([x-mu1(1,1),y-mu1(1,2)]) * inv(sigma1) * ([x-mu1(1,1);y-mu1(1,2)]) - 0.5 * log(det(sigma1)) + log(prior1);
    g_2 = (-0.5) * ([x-mu2(1,1),y-mu2(1,2)]) * inv(sigma2) * ([x-mu2(1,1);y-mu2(1,2)]) - 0.5 * log(det(sigma2)) + log(prior2);
    g = g_1 - g_2;
    g_solve = solve(g,'y');
end
```

Plot the boundary on top of the generated samples.

```
%boundary
syms x;
g_solve = bound(mu1,mu2,sigma,sigma,4/5,1/5);
boundary = double(subs(g_solve,x,scope_x));
figure('Name','First Random Point');
plot(r1(:,1),r1(:,2),'.r',r2(:,1),r2(:,2),'.k',scope_x,boundary,'-b');
```

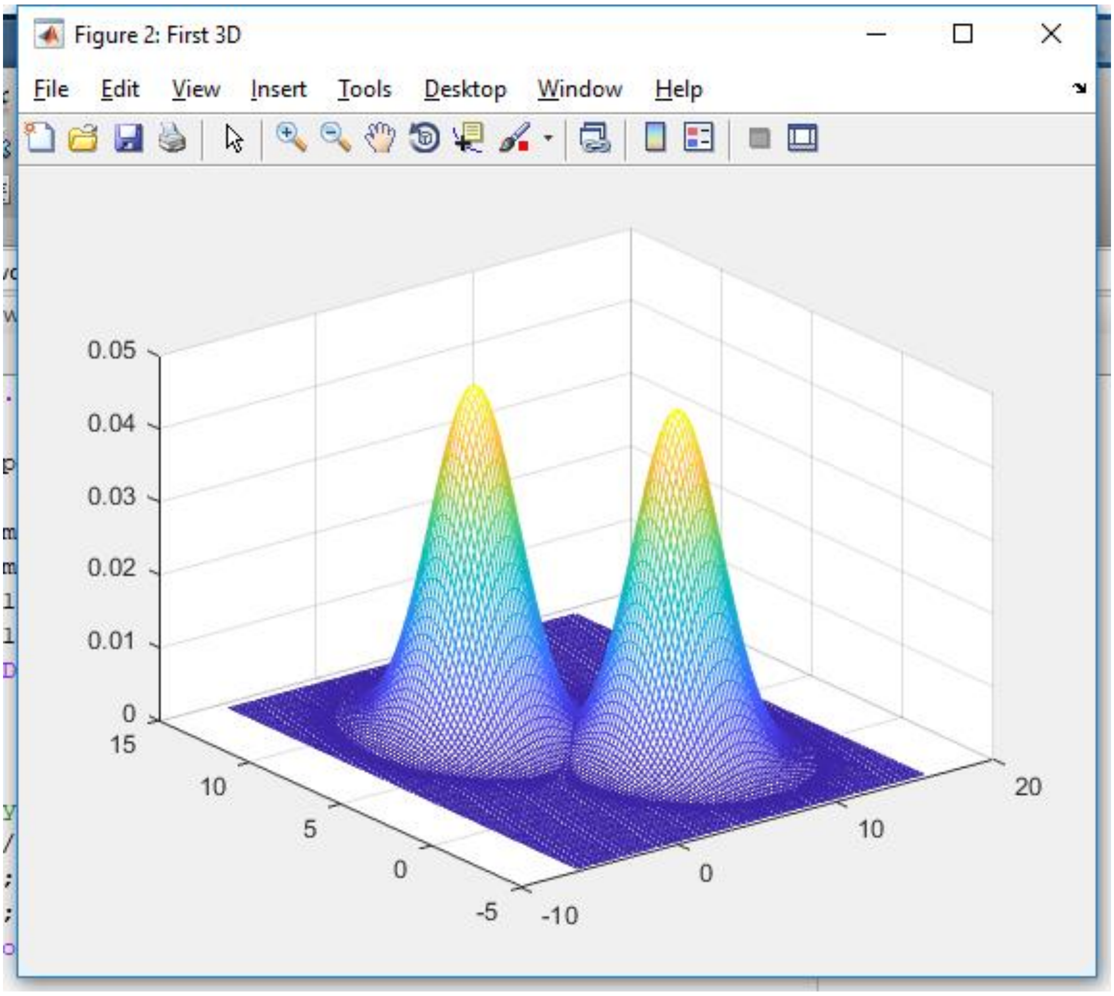
Boundary for the first distribution.

```
>> g_solve1

g_solve1 =

(28*x)/41 + 6182840975149691633/2769713770832855040
```



Plot the two classes in 3D.

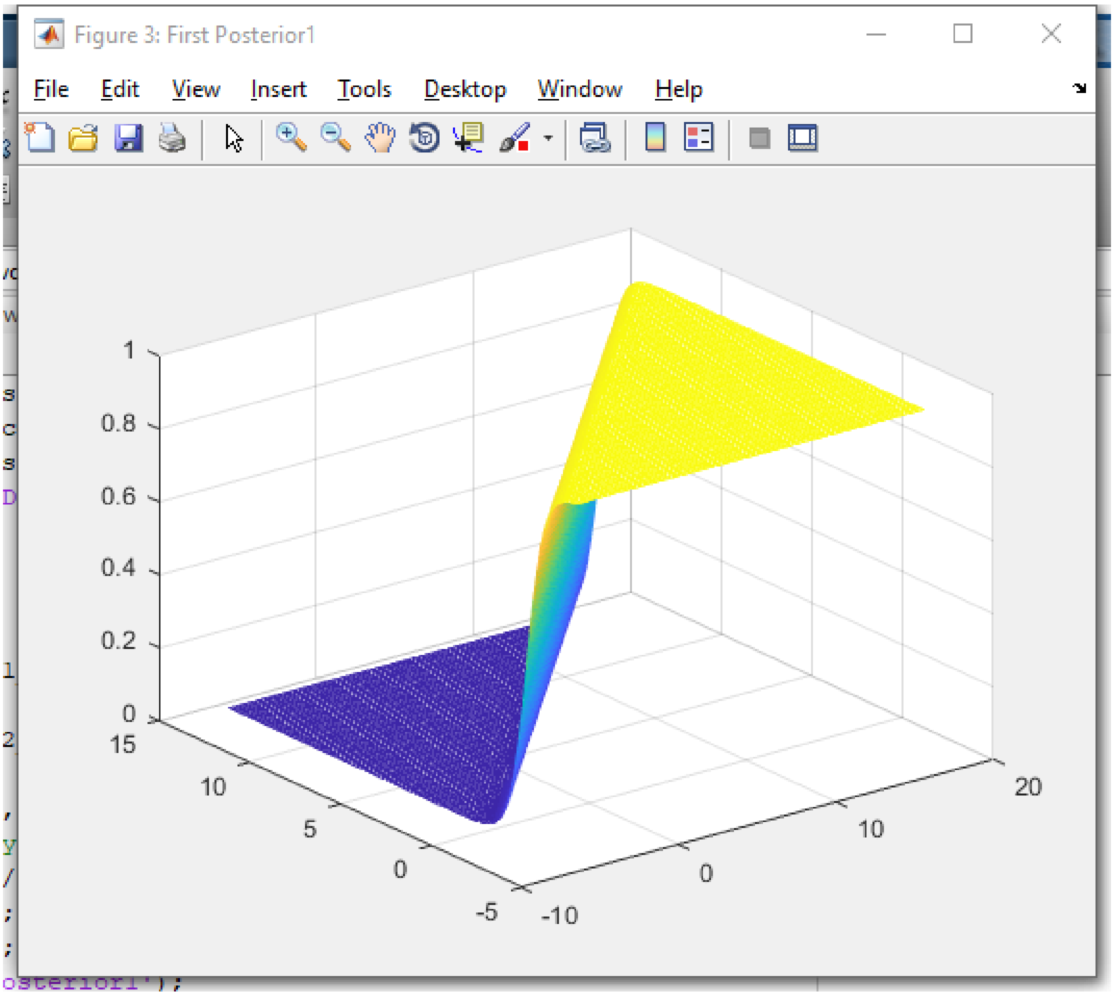
```
%3D
[x1,y1] = meshgrid(scope_x,scope_y);
xy = [x1(:), y1(:)];
p1 = mvnpdf(xy,mu1,sigma);
p2 = mvnpdf(xy,mu2,sigma);
P1 = reshape(p1,size(x1));
P2 = reshape(p2,size(x1));
figure('Name','First 3D')
mesh(x1,y1,P1);
hold on;
mesh(x1,y1,P2);
```

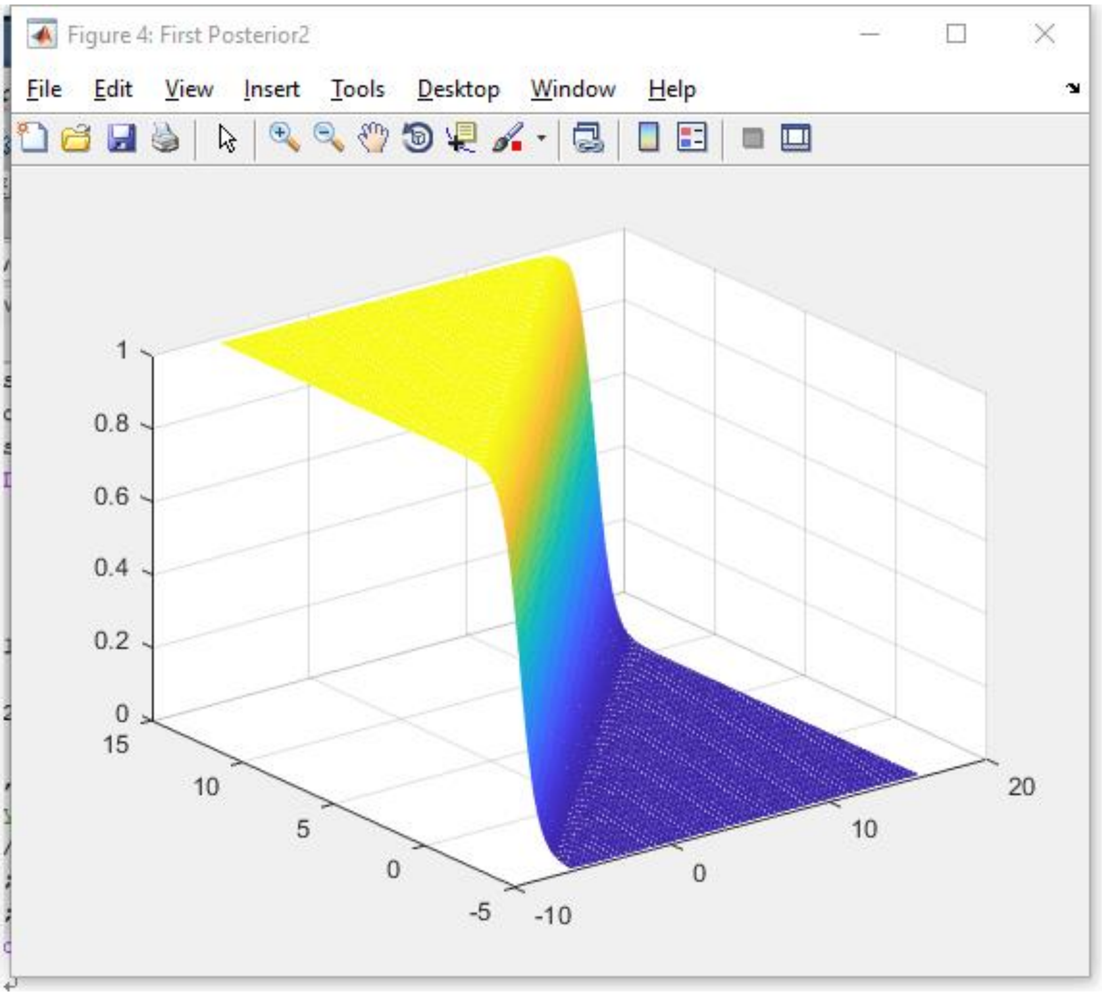


c)

Plot the posterior probabilities. Using the total probability formula to calculate the evidence P(x) and the Bayes Formula to calculate the posterior probability. P(w1) = 4/5, P(w2) = 1/5.

```
% posterior probability
px = P1 * 4/5 + P2 * 1/5;
post1 = (P1*4/5) ./ px;
post2 = (P2*1/5) ./ px;
figure('Name','First Posterior1');
mesh(x1,y1,post1);
figure('Name','First Posterior2');
mesh(x1,y1,post2);
```





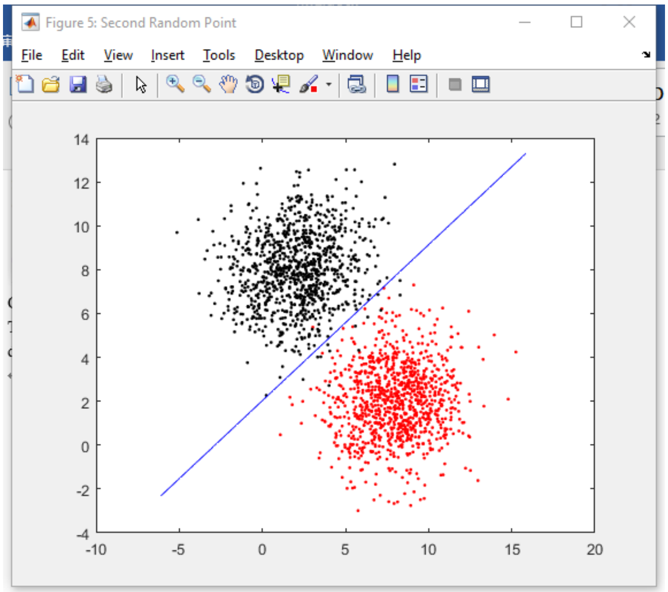
Code are nearly the same for the rest of them.

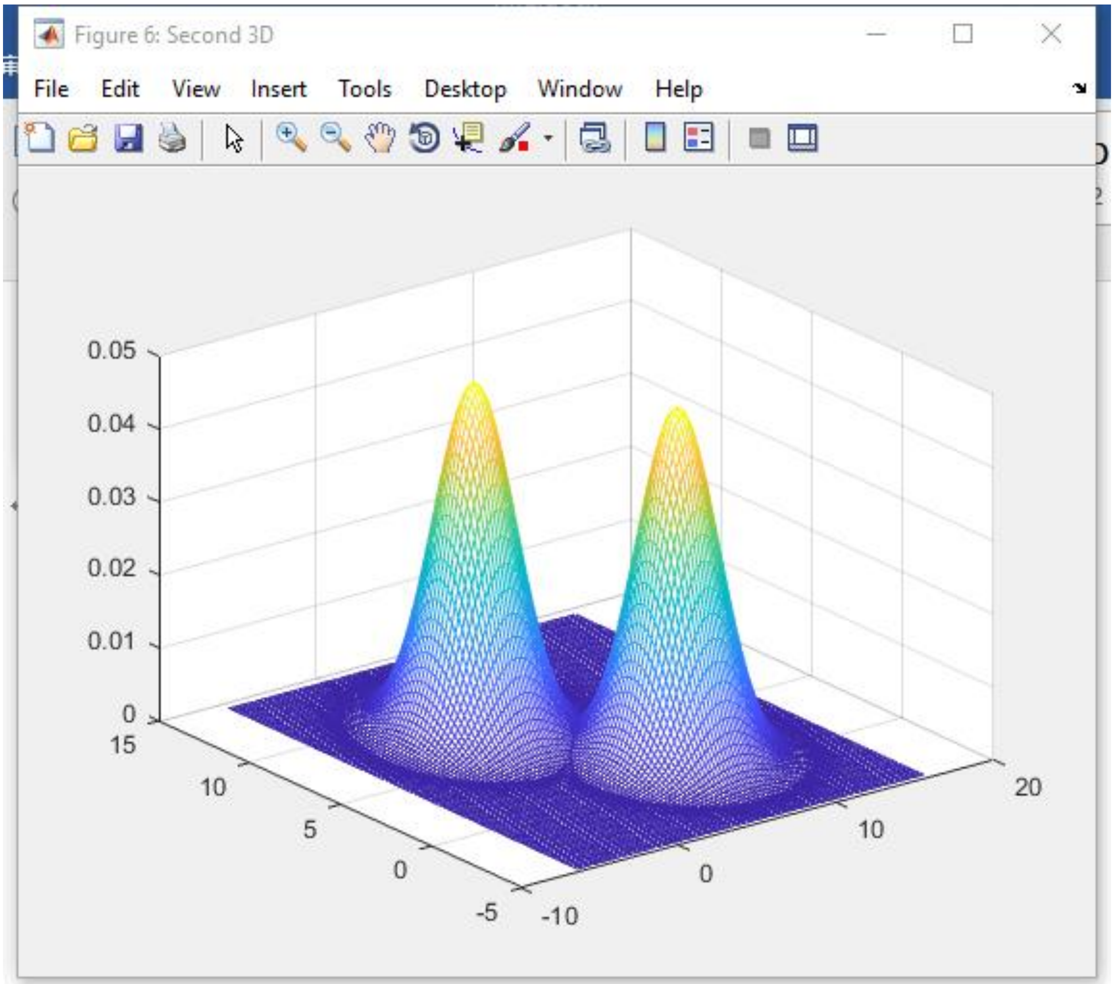
Therefore, only figure will be showed below.

d)

boundary for the second distribution

```
>> g_solve2

g_solve2 =

(32*x)/45 + 6157867716077030197/3039929748475084800
```





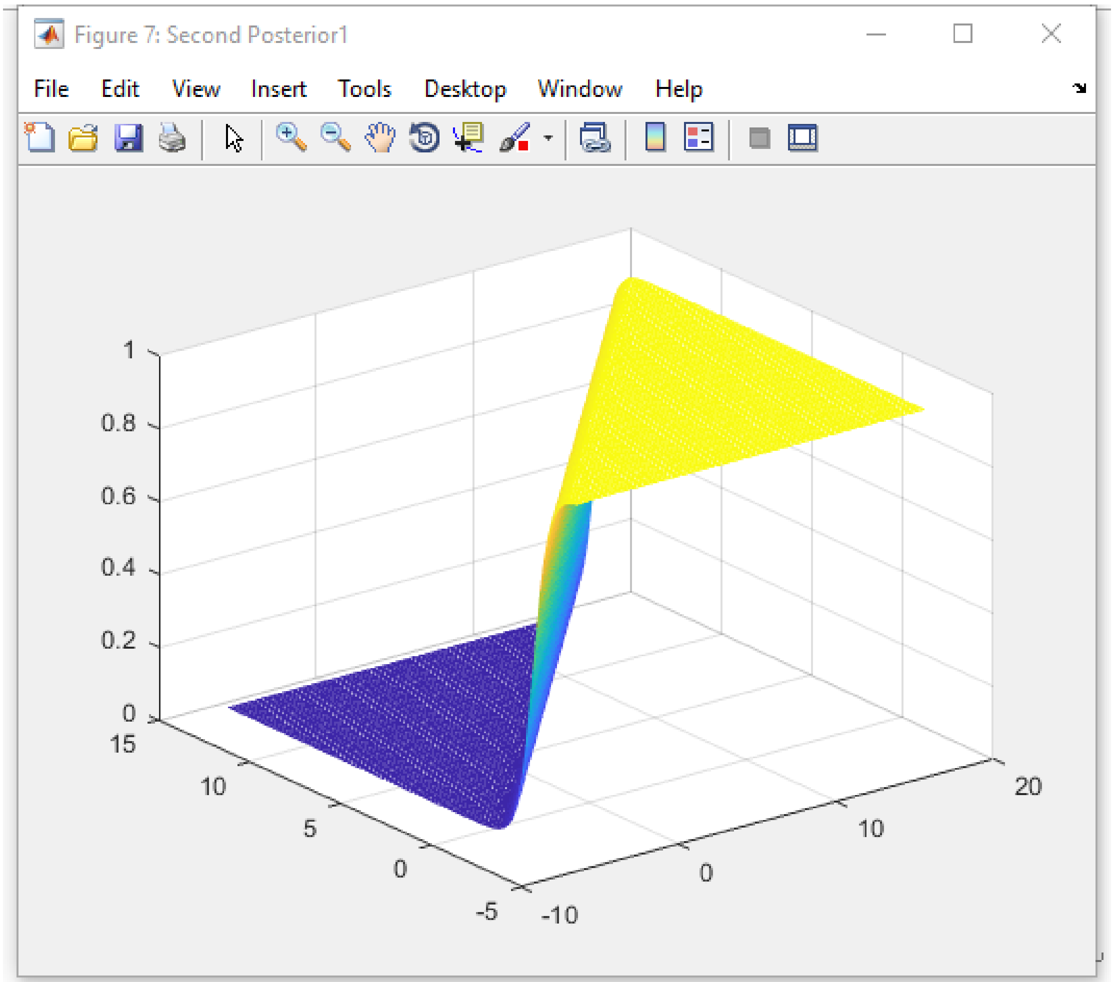


Figure 7: Second Posterior1
File Edit View Insert Tools Desktop Window Help
1
0.8
0.6
0.4
0.2
0
15
10
5
0
-5
-10
0
10
20

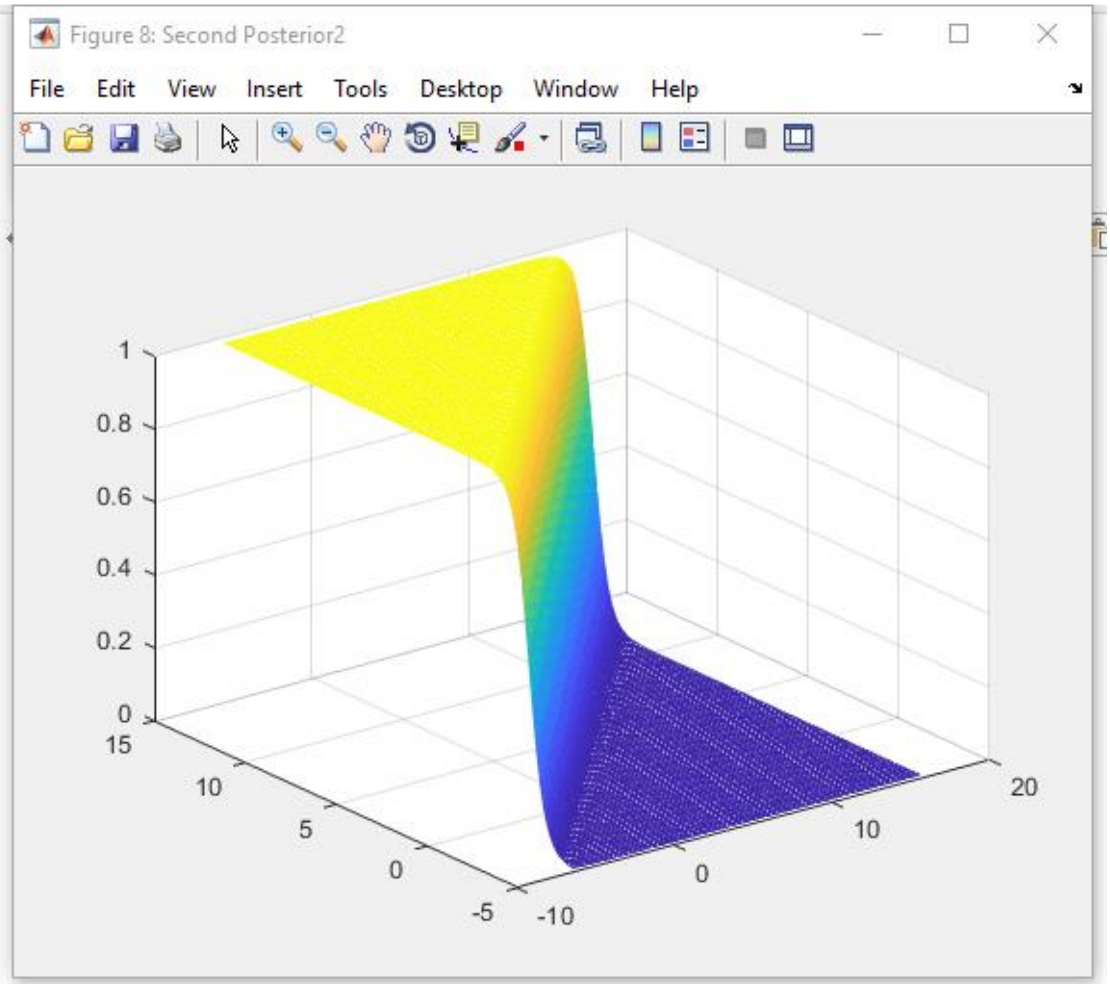
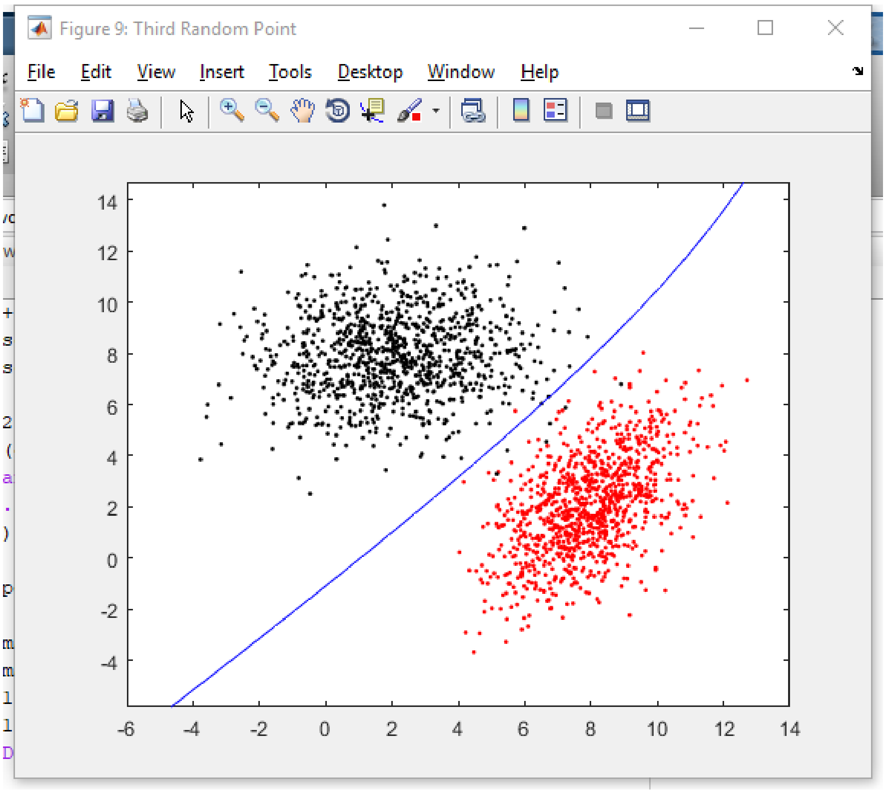


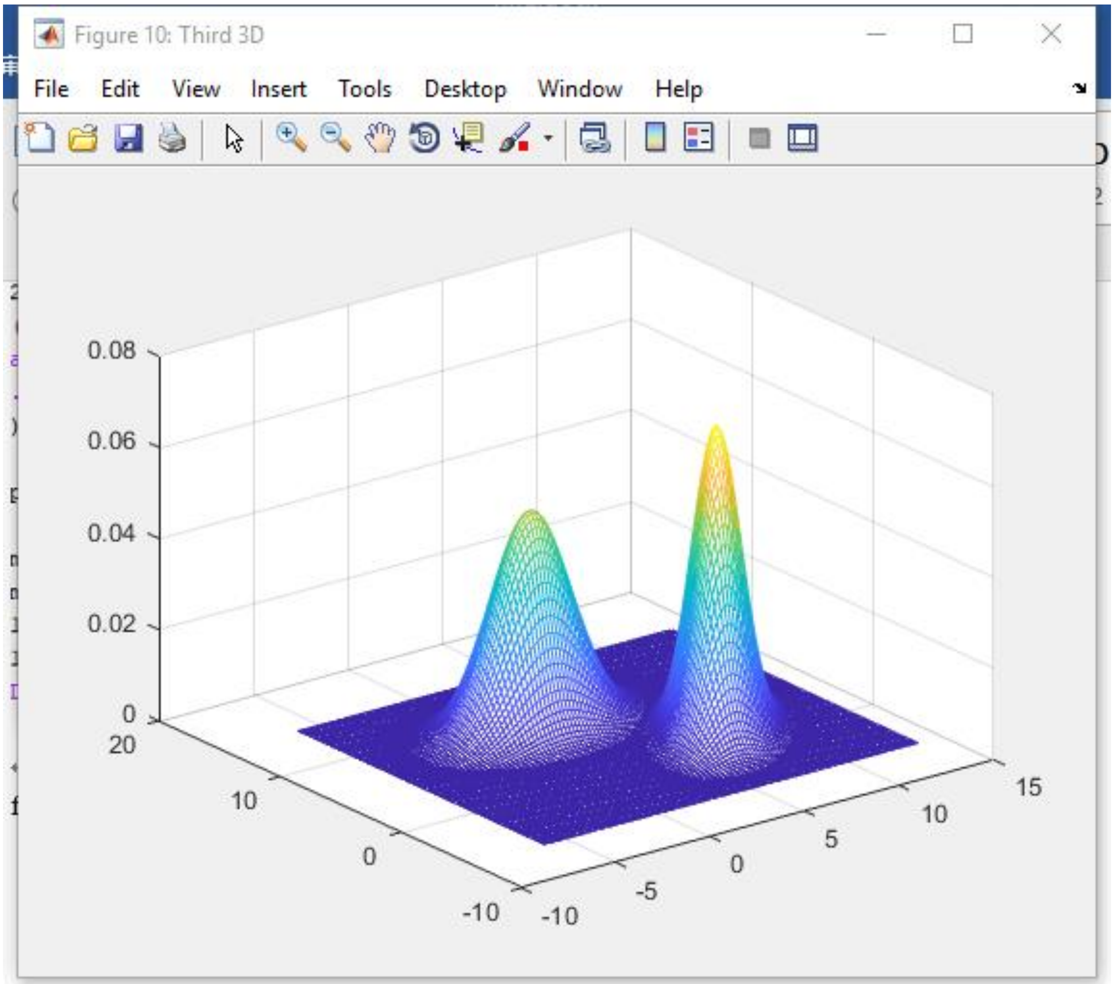
Figure 8: Second Posterior2
File Edit View Insert Tools Desktop Window Help
1
0.8
0.6
0.4
0.2
0
15
10
5
0
-5
-10
0
10
20

e)

For this graph, there is two boundaries, but one of them is two far away to plot in this graph.
Boundary for the third distribution

```
>> g_solve3

g_solve3 =

 (1632*x)/31 - (108106*((2675*x^2)/54053 - (100400*x)/54053 + 8520836848171519372705/486866141316514840576)^(1/2))/465 - 30184/31
 (1632*x)/31 + (108106*((2675*x^2)/54053 - (100400*x)/54053 + 8520836848171519372705/486866141316514840576)^(1/2))/465 - 30184/31
```





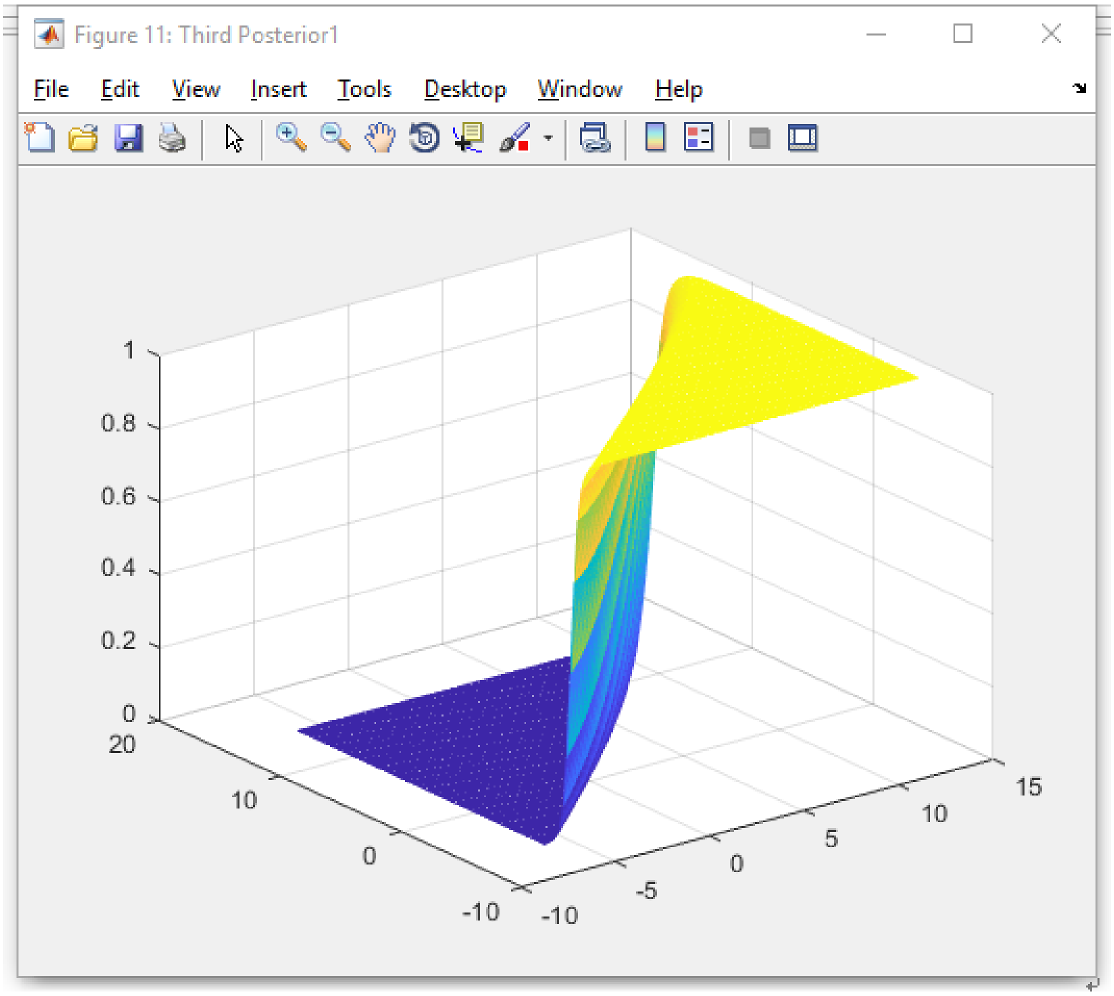


Figure 11: Third Posterior1
File Edit View Insert Tools Desktop Window Help
1
0.8
0.6
0.4
0.2
0
20
10
0
-10
-10
-5
0
5
10
15

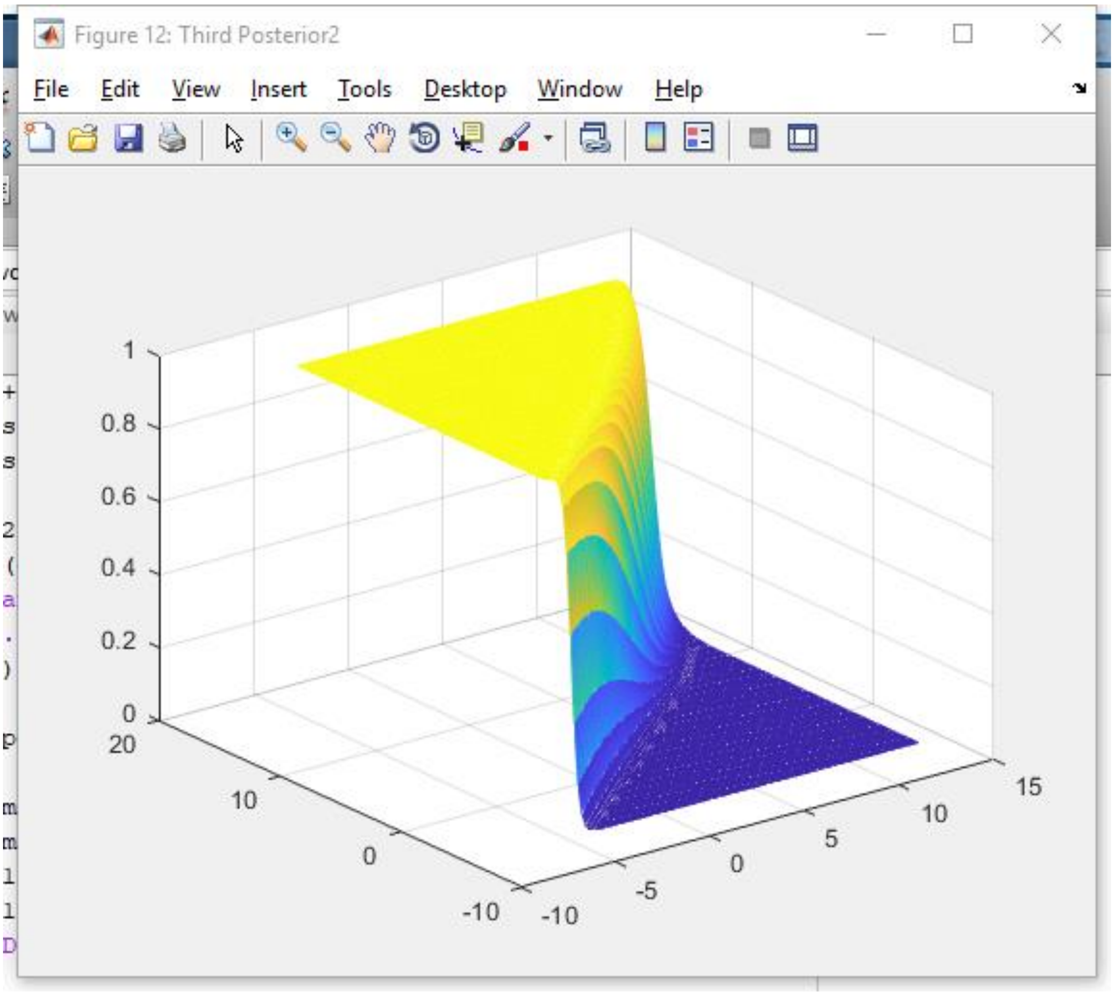


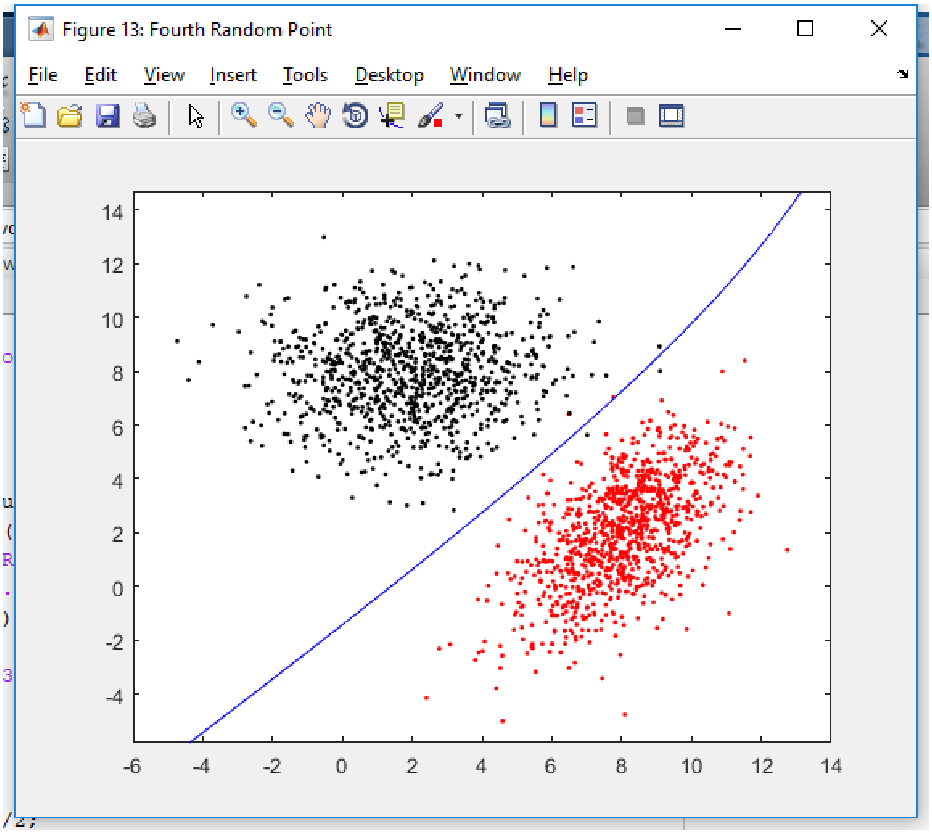
Figure 12: Third Posterior2
File Edit View Insert Tools Desktop Window Help
1
0.8
0.6
0.4
0.2
0
20
10
0
-10
-10
-5
0
5
10
15

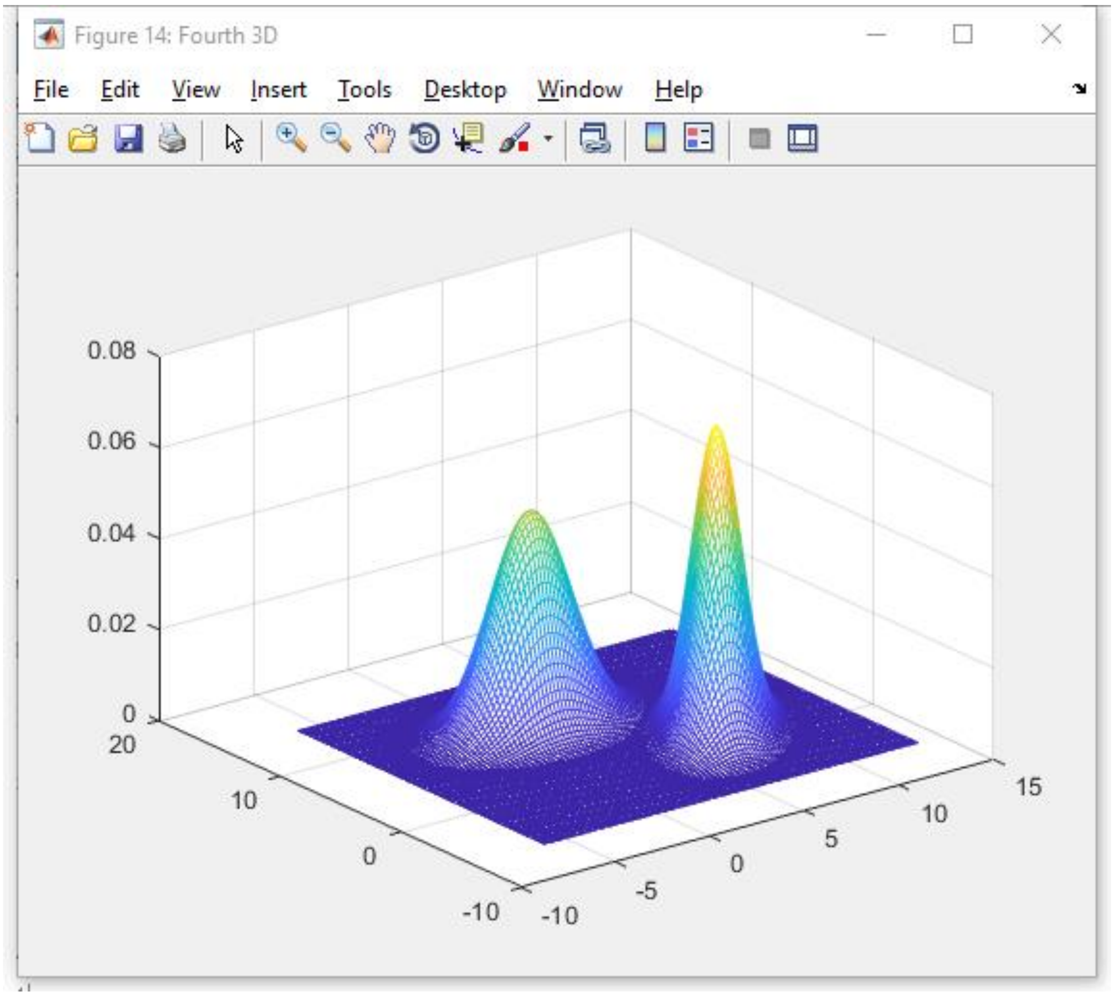
f)

The second boundary line cannot show either.

Boundary for the fourth distribution

```
g_solve4 =

 (1632*x)/31 - (108106*((2675*x^2)/54053 - (100400*x)/54053 + 851503056543712558883 5/4868661413165148405 76)^(1/2))/465 - 30184/31
 (1632*x)/31 + (108106*((2675*x^2)/54053 - (100400*x)/54053 + 851503056543712558883 5/4868661413165148405 76)^(1/2))/465 - 30184/31
```





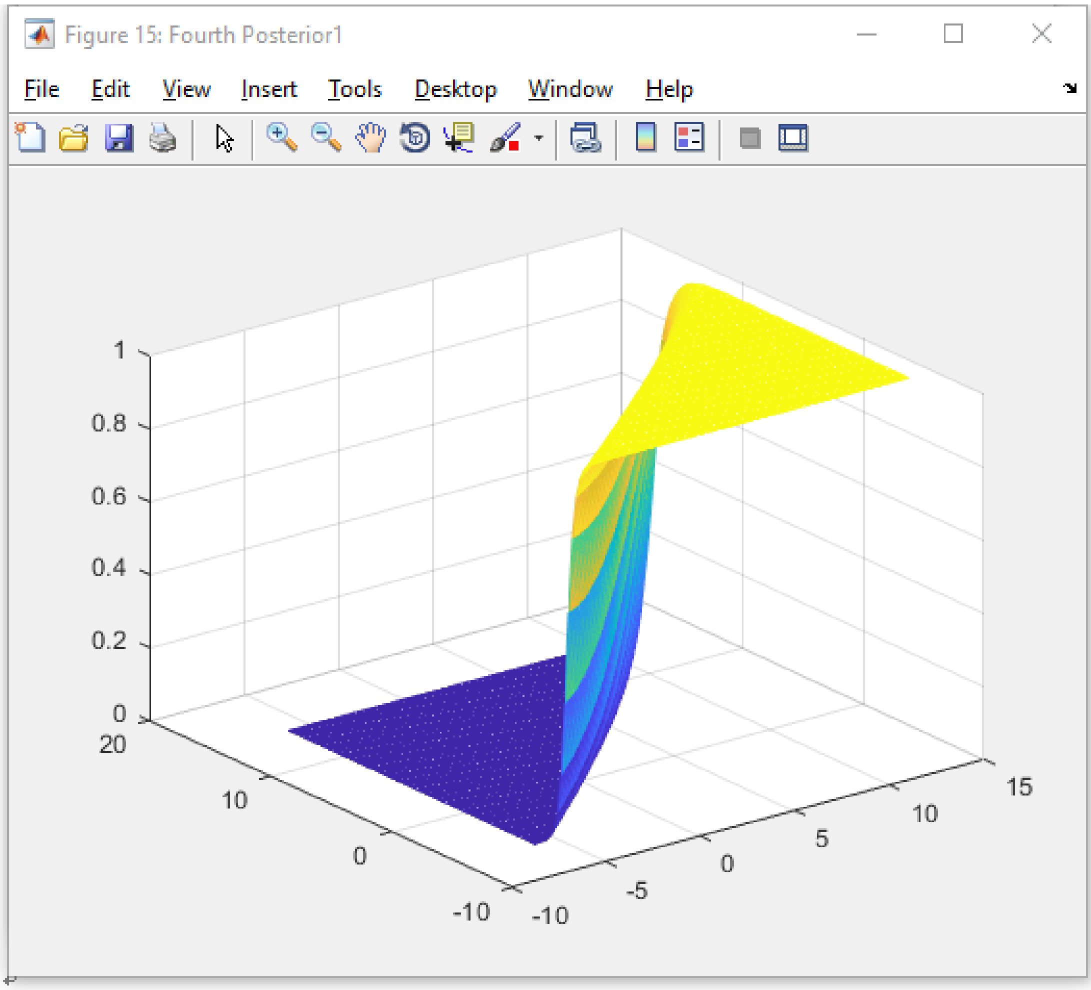


Figure 15: Fourth Posterior1
File Edit View Insert Tools Desktop Window Help
1
0.8
0.6
0.4
0.2
0
20
10
0
-10
-10
-5
0
5
10
15

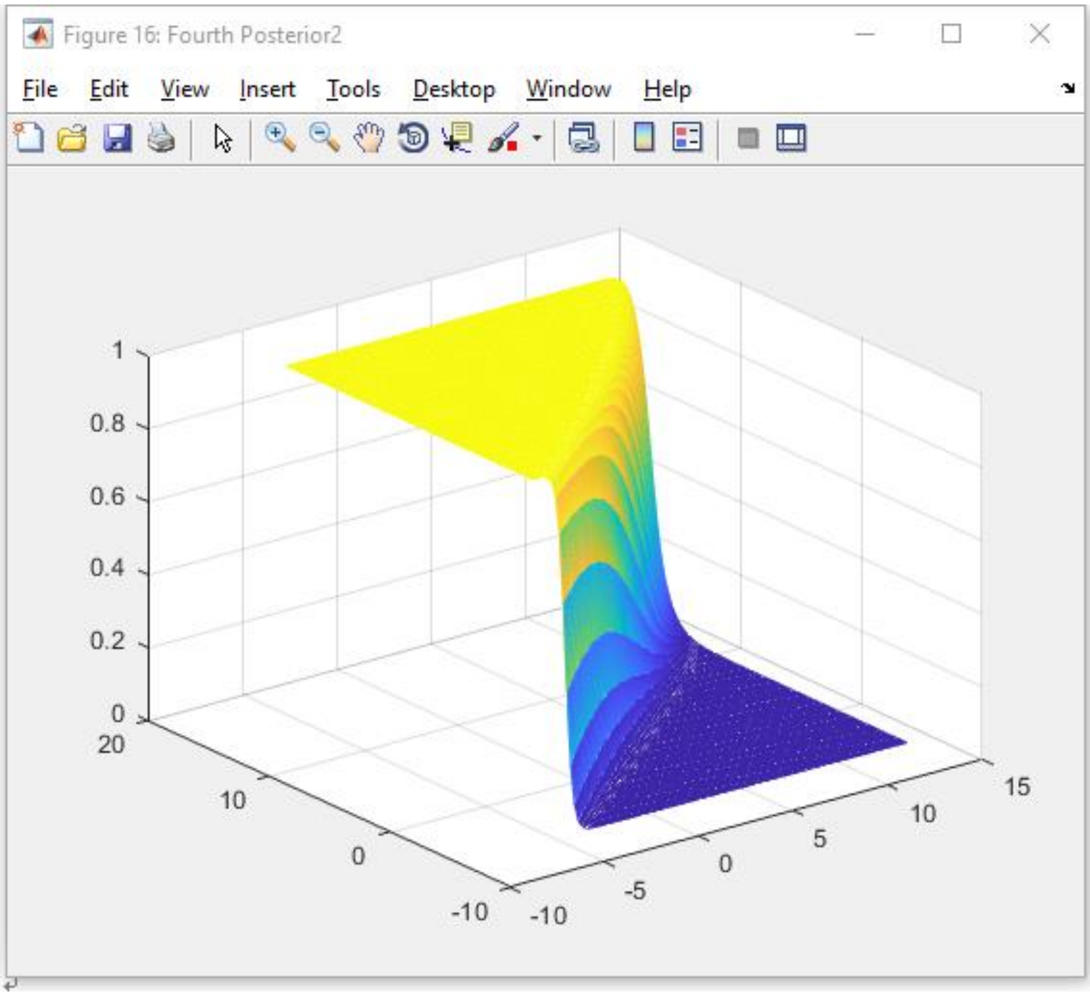


Figure 16: Fourth Posterior2
File Edit View Insert Tools Desktop Window Help
1
0.8
0.6
0.4
0.2
0
20
10
0
-10
-10
-5
0
5
10
15