

```

%Question 1
data = readtable('seeds.csv');
kernel_width = data.Var5;
kernel_height = data.Var7;
variety = data.Var8;
disp('1a');
figure(1)
gscatter(kernel_width, kernel_height, variety, 'rgb', '*x+', 8, 'on', 'kernel width', 'kernel height')
title('Scatter Plot')
snapnow;
X = [kernel_width kernel_height];
Y = variety;

disp('1b');
% Initialisation
centroids = [X(1,:) ; X(2,:) ; X(3,:)];
K=3;
disp('Initial cluster centers = 1st, 2nd and 3rd datapoints.');
```

perform\_kmeans(X, centroids, K, 2);

```

disp('1d');
% Initialisation
centroids = [X(1,:) ; X(80,:) ; X(150,:)];
K=3;
disp('Initial cluster centers = 1st, 80th and 150th datapoints.');
```

perform\_kmeans(X, centroids, K, 4);

disp('K-means converges faster. This makes sense, in the initialistaion there is one centroid for each cluster, i.e. these centroids are spread ac

```

function cluster_idx = assign_clusters(centroids, X, K)
    temp = [];
    for i=1:K
        temp = [ temp vecnorm(X-centroids(i,:), 2, 2)];
    end
    [~, cluster_idx] = min(temp, [], 2);
end

function centroids = reassign_centroids(cluster_idx, X, K)
    centroids = [];
    for i=1:K
        points = X(cluster_idx==i, :);
        sz = size(points,1);
        centroids = [centroids ; [(1/sz) * sum(points(:,1)), (1/sz) * sum(points(:,2))]];
    end
end

function total_sum= calculate_cost(X, idx, centroids)
    total_sum = 0;
    for k=1:size(centroids,1)
        c = centroids(k,:);
        total_sum = total_sum + sum(vecnorm((X(idx==k,:)-c),2,2));
    end
end

function perform_kmeans(X, centroids, K, fig_no)
    kernel_width = X(:,1);
    kernel_height = X(:,2);
    max_iterations = 20;
    centroids_0_to_3_iter = [centroids];
    cluster_idx_0_to_3 = [];
    prev1=0;
    prev2=0;
    for i=1:max_iterations
        cluster_idx = assign_clusters(centroids, X, K);
        cur = calculate_cost(X, cluster_idx, centroids);
        if(i> 2 && cur == prev1 && cur == prev2)
            fprintf("k-means converged at i=%d \n", i-1);
            final_cluster_idx = cluster_idx;
            final_centroids = centroids;
            break
        end
        prev1 = prev2;
        prev2 = cur;
        centroids = reassign_centroids(cluster_idx, X, K);
        if i <=3
            centroids_0_to_3_iter = [centroids_0_to_3_iter; centroids];
            cluster_idx_0_to_3 = [cluster_idx_0_to_3; cluster_idx];
        end
    end
    cluster_idx_0_to_3 = [cluster_idx_0_to_3; cluster_idx];

    disp('Scatter plot');
    sz = size(X,1);
    figure(fig_no)
    for j=0:3
        subplot(2,2,j+1);
        start_idx = sz*j + 1;
        gscatter(kernel_width, kernel_height, cluster_idx_0_to_3(start_idx:start_idx+sz-1,:), 'rgb', '*x+', 8, 'on', 'kernel width', 'kernel heigh
        hold on;
        start_idx = 3*j + 1;
        plot(centroids_0_to_3_iter(start_idx:start_idx+2,1), centroids_0_to_3_iter(start_idx:start_idx+2,2), '.k', 'MarkerSize', 15);
        title(strcat('Iteration No.', int2str(j)));
        legend('1', '2', '3','centroids');
```

end

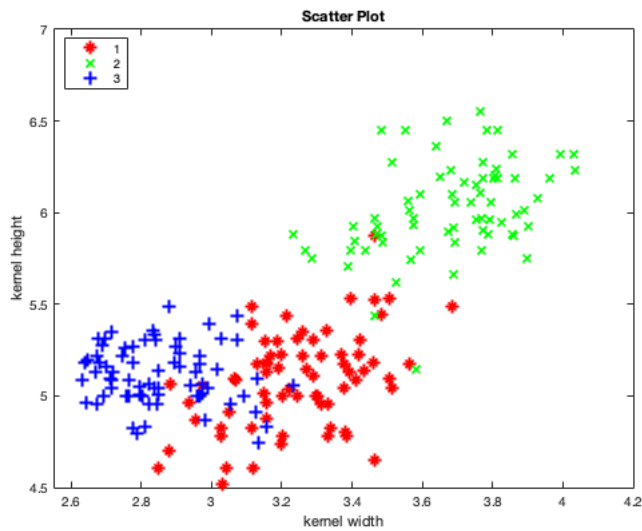
snapnow;

```

disp('Scatter plot');
sz = size(X,1);
figure(fig_no+1)
gscatter(kernel_width, kernel_height, final_cluster_idx, 'rgb', '*x+', 8, 'on', 'kernel width', 'kernel height');
hold on;
plot(centroids(:,1), centroids(:,2), '.k', 'MarkerSize', 15);
title('After convergence');
legend('1', '2', '3','centroids');
snapnow;
end

```

1a)

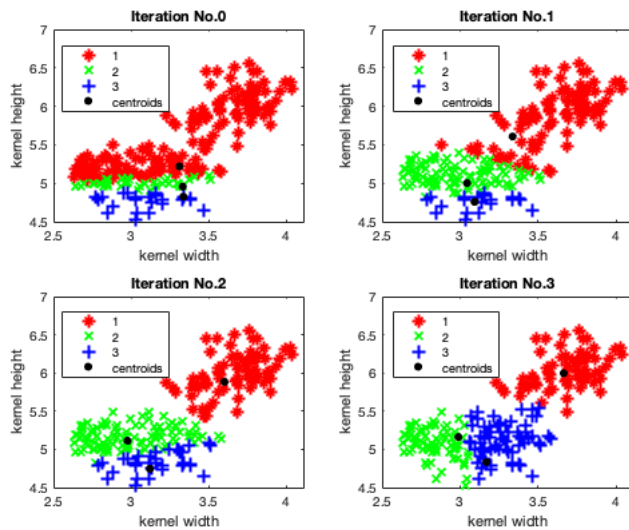


1b)

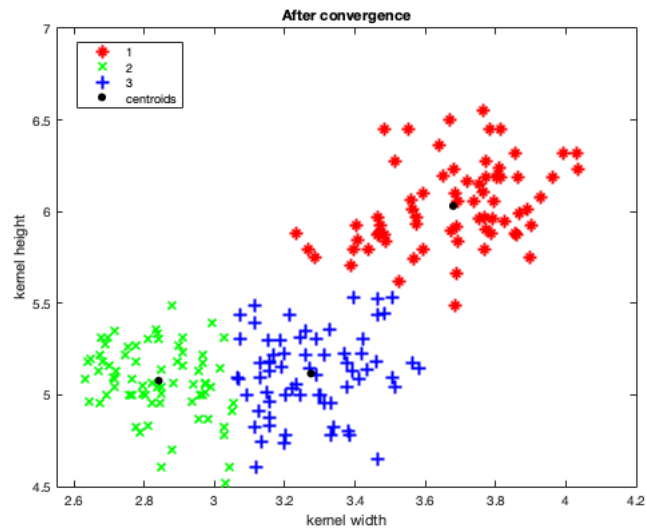
Initial cluster centers = 1st, 2nd and 3rd datapoints.

k-means converged at i=17

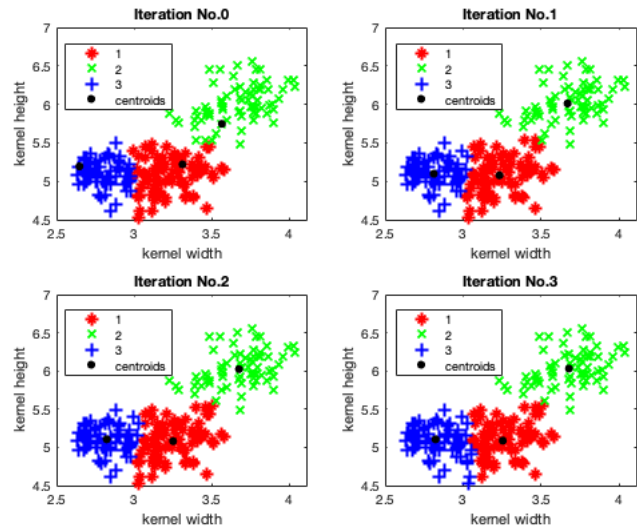
Scatter plot



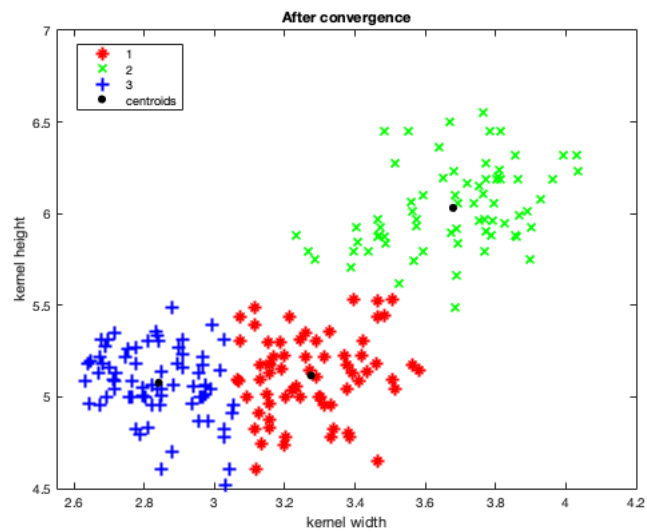
Scatter plot



1d)  
Initial cluster centers = 1st, 80th and 150th datapoints.  
k-means converged at i=8  
Scatter plot



Scatter plot



K-means converges faster. This makes sense, in the initialistaion there is one centroid for each cluster, i.e. these centroids are spread across.



```

% Question 2
data = readtable('seeds.csv');
kernel_width = data.Var5;
kernel_height = data.Var7;
variety = data.Var8;
X = [kernel_width kernel_height];

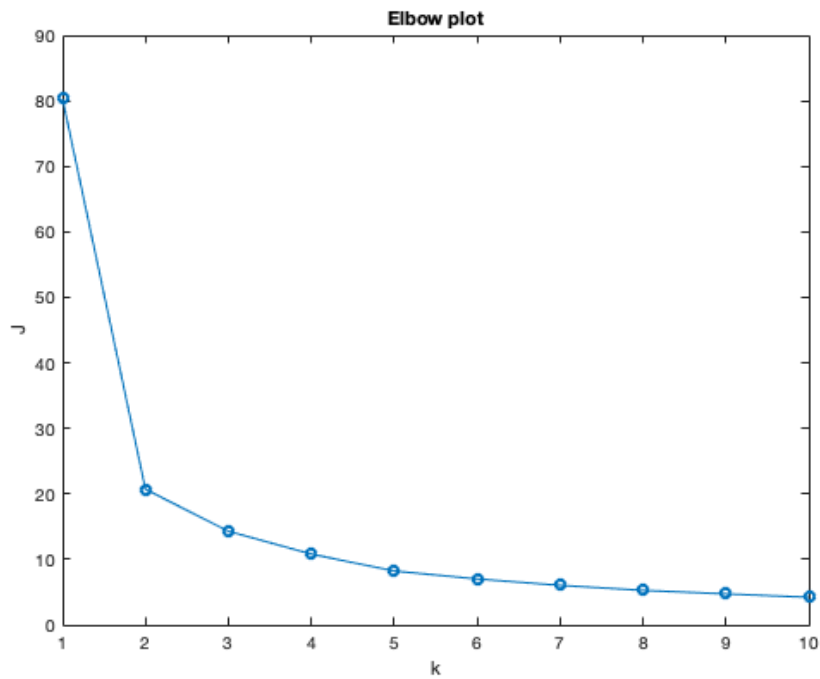
disp('2a');
figure(1);
J = [];
for k=1:10
    [idx, C, sumd] = kmeans(X, k, 'Start', 'Sample', 'Replicates',100);
    J = [J; sum(sumd)];
end
plot(1:10, J, '-o');
title('Elbow plot');
xlabel('k');
ylabel('J');
snapnow;
disp('I would consider points k=2 or 3 to be the elbow.');
```

```

disp('2b')
disp('In the figure the legend data corresponds to centroids')
figure(2);
K = [2, 3, 4, 5, 6];
for i=1:size(K,2)
    subplot(2, 3, i);
    [idx, C] = kmeans(X, K(1,i), 'Start', 'Sample', 'Replicates',100);
    gscatter(kernel_width, kernel_height, idx);
    xlabel('Kernel Width');
    ylabel('Kernel Height');
    hold on;
    plot(C(:,1), C(:,2), '.k', 'MarkerSize', 15);
    title(strcat('K=', int2str(K(i))));
end
snapnow;
disp('2c')
figure(3);
for i=1:size(K,2)
    subplot(2, 3, i);
    [idx, C] = kmeans(X, K(1,i), 'Start', 'Sample', 'Replicates',100);
    [s, h] = silhouette(X, idx);
    title(strcat('k=',int2str(K(1,i))));
    fprintf("K=%d , Avg. silhouette score=%f \n", K(1,i), sum(s)/size(s, 1));
end
disp('We see that the silhouette score for K=2 is the maximum. We know that the clusters are well seperated');
disp('when the score is close to 1. This result matches the conclusion from the elbow plot. Hence optimal K=2');
snapnow;

```

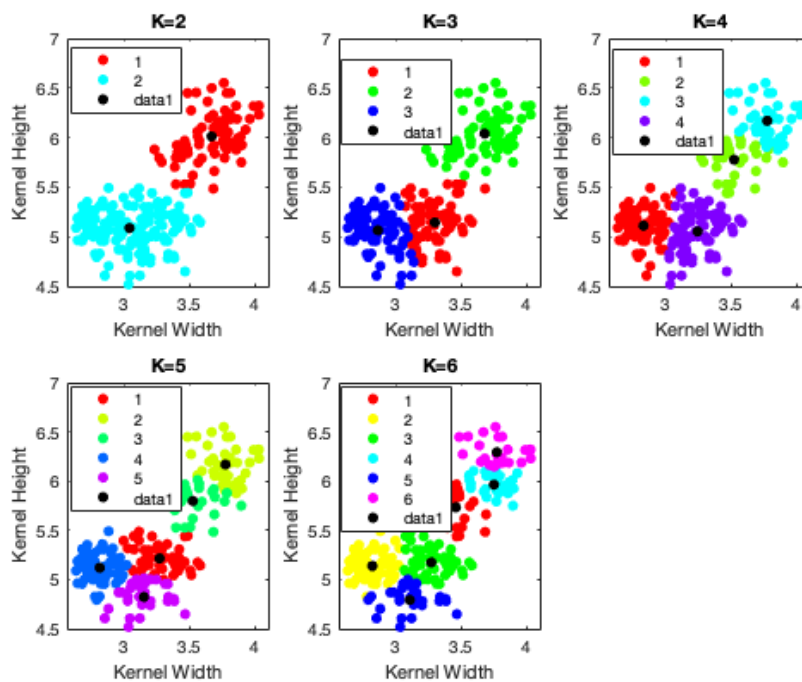
2a)



I would consider points k=2 or 3 to be the elbow.

2b)

In the figure the legend data1 corresponds to centroids



2c)

K=2 , Avg. silhouette score=0.831273

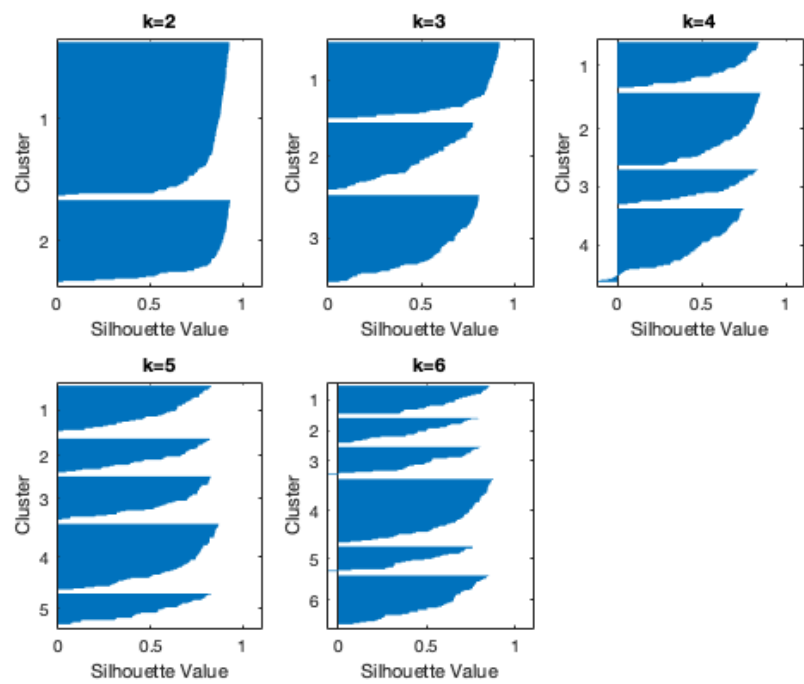
K=3 , Avg. silhouette score=0.642039

K=4 , Avg. silhouette score=0.579551

K=5 , Avg. silhouette score=0.589436

K=6 , Avg. silhouette score=0.578406

We see that the silhouette score for K=2 is the maximum. We know that the clusters are well separated when the score is close to 1. This result matches the conclusion from the elbow plot. Hence optimal K=2



Published with MATLAB® R2018a

```
%Question 3
data = readtable('seeds.csv');
kernel_width = data.Var5;
kernel_height = data.Var7;
variety = data.Var8;

X = [kernel_width kernel_height];
X = [X; [-1 -1]; [10 0]];

disp('3a');
disp('Scatter Plot');
figure(1);
[idx, C] = kmeans(X, 3, 'Start', 'Sample', 'Replicates',100);
gscatter(X(:,1), X(:,2), idx);
xlabel('Kernel Width');
ylabel('Kernel Height');
hold on;
plot(C(:,1), C(:,2), 'ok', 'MarkerSize', 10);
title('kmeans, K=3');
legend('1', '2', '3','centroids')
snapnow;

disp('3b');
disp('Scatter Plot');
figure(2);
[idx, C] = kmeans(X, 3, 'Start', 'Sample', 'Replicates',100, 'Distance','cityblock');
gscatter(X(:,1), X(:,2), idx);
xlabel('Kernel Width');
ylabel('Kernel Height');
hold on;
plot(C(:,1), C(:,2), 'ok', 'MarkerSize', 10);
title('kmedians, K=3');
legend('1', '2', '3','centroids')
snapnow;

disp('3c');
disp('Scatter Plot');
figure(3);
[idx, C] = kmedoids(X, 3, 'Start', 'Sample', 'Replicates',100, 'Distance','cityblock');
gscatter(X(:,1), X(:,2), idx);
xlabel('Kernel Width');
ylabel('Kernel Height');
hold on;
plot(C(:,1), C(:,2), 'ok', 'MarkerSize', 10);
title('kmediods, K=3');
legend('1', '2', '3','centroids')
snapnow;

disp('4d')
disp('In k-means, we see one cluster that contains just one of the outliers. We can see that k-means is sensitive to outliers.');
```

As we know, in case of l2-norm calculation, one very distant outlier can modify the mean to a large extent.');

In case of k-medians, clusters seem right - here the outlier doesn't have a huge impact on centroids and the clusters, as we';

use the manhattan distance. k-medians is more robust than kmeans. Even in-case of k-mediods using manhattan distance, clusters';

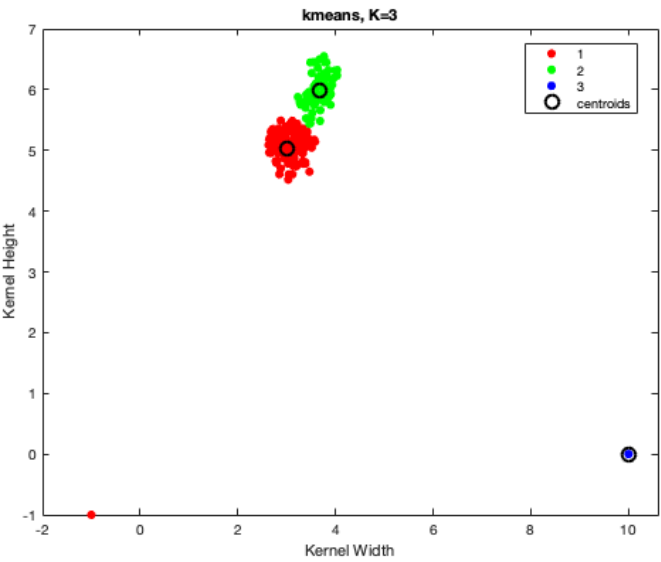
seem right - the outlier doesn't have a huge impact. This is more robust than k-means. Also in general, k-mediods is more robust';

than k-medians as well. It chooses cluster centres among the data points as opposed to k-medians. On zooming into the figures,');

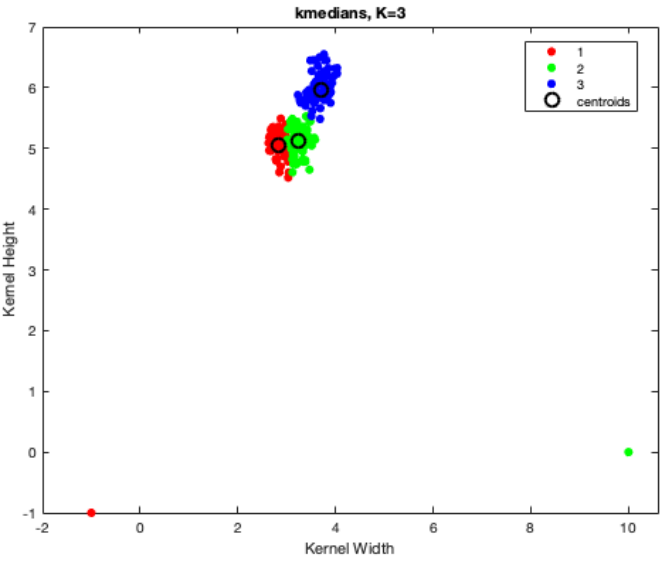
this can be clearly observed.')

3a)  
Scatter Plot

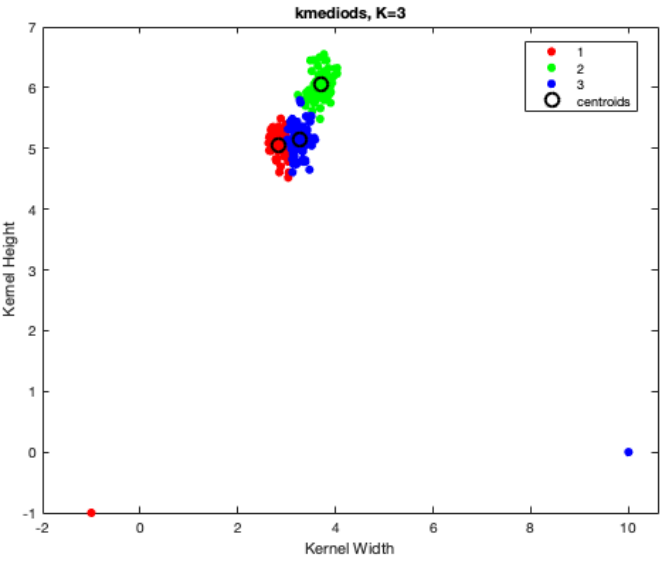




3b)  
Scatter Plot



3c)  
Scatter Plot



4d)  
In k-means, we see one cluster that contains just one of the outliers. We can see that k-means is sensitive to outliers. As we know, in case of l2-norm calculation, one very distant outlier can modify the mean to a large extent. In case of k-medians, clusters seem right - here the outlier doesn't have a huge impact on centroids and the clusters, as we use the manhattan distance. k-medians is more robust than kmeans. Even in-case of k-medoids using manhattan distance, clusters seem right - the outlier doesn't have a huge impact. This is more robust than k-means. Also in general, k-medoids is more robust than k-medians as well. It chooses cluster centres among the data points as opposed to k-medians. On zooming into the figures, this can be clearly observed.

---

*Published with MATLAB® R2018a*

```

%Question 4
[data, TEXT, raw] = xlsread('nutrients.xlsx');
food_items = TEXT(2:end, 1:2);
X = data(:,1:end-1);

disp('4a');
figure(1);
J = [];
cluster_centres = [];
for k=2:10
    [idx, C, sumd] = kmeans(X, k, 'Start', 'sample', 'Replicates',100, 'MaxIter',200);
    J = [J; sum(sumd)];
    cluster_centres = [cluster_centres ; C];
end
plot(2:10, J, '-o');
xlabel('k');
ylabel('J');
title('Elbow Plot')
snapnow;
disp('Based on the elbow plot, I would consider K=4, 5 or 6 as the elbow');

disp('4b')
figure(2);
i=1;
for k=3:7
    subplot(2, 3, i);
    [idx, C] = kmeans(X, k, 'Start', 'sample', 'Replicates',100, 'MaxIter',200);
    [s, h] = silhouette(X, idx);
    title(strcat('k=',int2str(k)));
    fprintf("K=%d , Silhouette Score=%f \n", k, sum(s)/size(s, 1));
    i = i+ 1;
end
snapnow;

disp("4c")
start_idx = 1;
for k=2:10
    indices = [];
    centroids = cluster_centres(start_idx: start_idx+k-1, :);
    for c=1:size(centroids,1)
        [~, idx] = min(vecnorm(X-centroids(c,:),2,2));
        indices = [indices; idx];
    end
    fprintf("K=%d, Food Representatives : \n", k);
    disp(food_items(indices,:));
    start_idx = start_idx +k;
end

disp('4d');

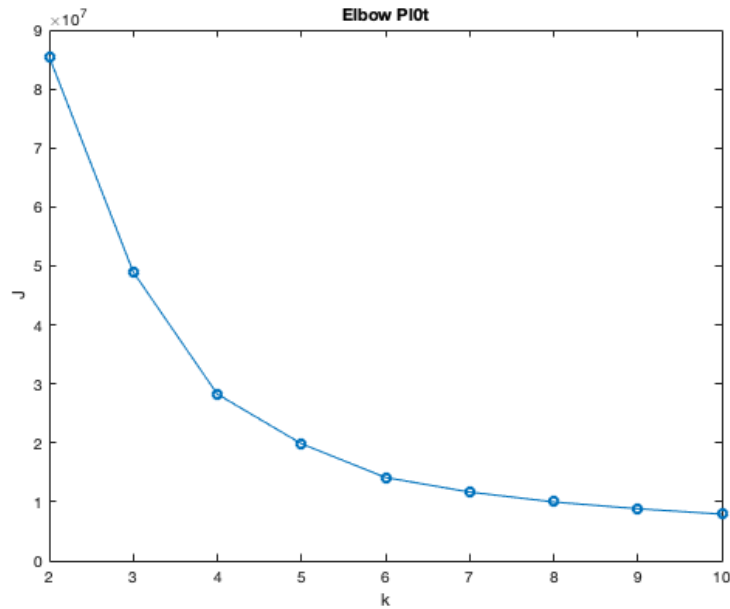
disp(' From elbow plot, the conclusion was the choose K=4, 5, or 6. ');
disp('From the silhouette values, though the average value is maximum when K=3, clusters don't seem to be well distributed');
disp('- most of the items are in cluster 1 and extremely few in cluster 3. Similar observation with K=4 and K=5. ');
disp('For K=6 and K=7, the cluster size distribution seems better. ');
disp('However, by looking at the average silhouette score for varying K, the score for K=7 is much lower. ');

disp(' From food domain knowledge, we can see that if we take K=6, we have food items from 6 different groups. ');
disp('(Whereas for K=4, 5 or 7, some food groups are repeated). ');

disp('Hence, I think we should select K=6. ');

```

4a)



Based on the elbow plot, I would consider K=4, 5 or 6 as the elbow  
4b)

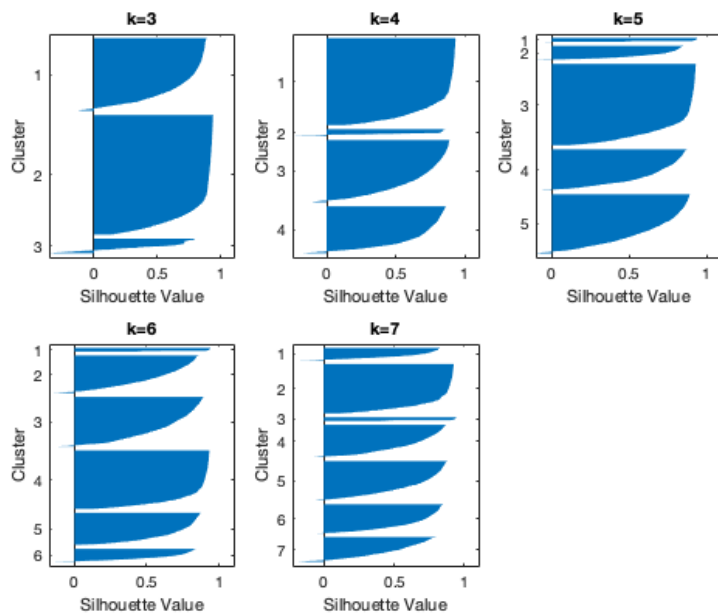
K=3 , Silhouette Score=0.742971

K=4 , Silhouette Score=0.716766

K=5 , Silhouette Score=0.706845

K=6 , Silhouette Score=0.680982

K=7 , Silhouette Score=0.650313



4c)

K=2, Food Representatives :  
 'Edamame, frozen, prepared' 'Vegetables and Vegetable Products'  
 'Croissants, cheese' 'Baked Products'

K=3, Food Representatives :  
 'Biscuits, plain or buttermilk, ...' 'Baked Products'  
 'Stew, mutton, corn, squash (Nav...)' 'Ethnic Foods'  
 'Nuts, formulated, wheat-based, ...' 'Nut and Seed Products'

K=4, Food Representatives :  
 'Fast foods, miniature cinnamon ...' 'Fast Foods'  
 'WENDY'S, CLASSIC SINGLE Hamburg...' 'Fast Foods'  
 'Soup, pea, split with ham, cann...' 'Soups, Sauces, and Gravies'  
 'Pork, cured, salt pork, raw' 'Pork Products'

K=5, Food Representatives :  
 'WORTHINGTON Stakelets, frozen, ...' 'Legumes and Legume Products'  
 'Bread, pan dulce, sweet yeast b...' 'Baked Products'

'Pie crust, cookie-type, prepare...'	'Baked Products'
'Soup, vegetable chicken, canned...'	'Soups, Sauces, and Gravies'
'Pork, fresh, variety meats and ...'	'Pork Products'

K=6, Food Representatives :

'Cereals ready-to-eat, POST, Gre...'	'Breakfast Cereals'
'Peppered loaf, pork, beef'	'Sausages and Luncheon Meats'
'McDONALD'S, DOUBLE QUARTER POUN...'	'Fast Foods'
'CAMPBELL Soup Company, CAMPBELL...'	'Soups, Sauces, and Gravies'
'Pork, fresh, variety meats and ...'	'Pork Products'
'Pie crust, cookie-type, prepare...'	'Baked Products'

K=7, Food Representatives :

'Pork, fresh, variety meats and ...'	'Pork Products'
'Turkey sausage, reduced fat, br...'	'Sausages and Luncheon Meats'
'Fast foods, bagel, with egg, sa...'	'Fast Foods'
'Soup, tomato rice, canned, prep...'	'Soups, Sauces, and Gravies'
'Cereals ready-to-eat, POST, Gre...'	'Breakfast Cereals'
'Sauce, alfredo mix, dry'	'Soups, Sauces, and Gravies'
'Honey loaf, pork, beef'	'Sausages and Luncheon Meats'

K=8, Food Representatives :

'Cereals ready-to-eat, granola, ...'	'Breakfast Cereals'
'Nuts, formulated, wheat-based, ...'	'Nut and Seed Products'
'BURGER KING, Cheeseburger'	'Fast Foods'
'Turkey sausage, reduced fat, br...'	'Sausages and Luncheon Meats'
'Honey loaf, pork, beef'	'Sausages and Luncheon Meats'
'Soup, tomato rice, canned, prep...'	'Soups, Sauces, and Gravies'
'Cereals ready-to-eat, POST, Gre...'	'Breakfast Cereals'
'Shortening, vegetable, househol...'	'Fats and Oils'

K=9, Food Representatives :

'Cereals ready-to-eat, QUAKER, Q...'	'Breakfast Cereals'
'Shortening, vegetable, househol...'	'Fats and Oils'
'Nuts, formulated, wheat-based, ...'	'Nut and Seed Products'
'McDONALD'S, QUARTER POUNDER wit...'	'Fast Foods'
'WENDY'S, Chicken Nuggets'	'Fast Foods'
'MORNINGSTAR FARMS Hot and Spicy...'	'Legumes and Legume Products'
'Turkey, white, rotisserie, deli...'	'Sausages and Luncheon Meats'
'CAMPBELL Soup Company, CAMPBELL...'	'Soups, Sauces, and Gravies'
'Cereals ready-to-eat, granola, ...'	'Breakfast Cereals'

K=10, Food Representatives :

'Turkey sausage, reduced fat, br...'	'Sausages and Luncheon Meats'
'CAMPBELL Soup Company, CAMPBELL...'	'Soups, Sauces, and Gravies'
'Fast foods, cheeseburger; singl...'	'Fast Foods'
'Shortening, vegetable, househol...'	'Fats and Oils'
'Nuts, formulated, wheat-based, ...'	'Nut and Seed Products'
'LOMA LINDA Tender Bits, canned,...'	'Legumes and Legume Products'
'KENTUCKY FRIED CHICKEN, Fried C...'	'Fast Foods'
'Cereals ready-to-eat, QUAKER, Q...'	'Breakfast Cereals'
'Okra, frozen, cooked, boiled, d...'	'Vegetables and Vegetable Products'
'Cereals ready-to-eat, granola, ...'	'Breakfast Cereals'

4d)

From elbow plot, the conclusion was the choose K=4, 5, or 6.

From the silhouette values, though the average value is maximum when K=3, clusters don't seem to be well distributed - most of the items are in cluster 1 and extremely few in cluster 3. Similar observation with K=4 and K=5.

For K=6 and K=7, the cluster size distribution seems better.

However, by looking at the average silhouette score for varying K, the score for K=7 is much lower.

From food domain knowledge, we can see that if we take K=6, we have food items from 6 different groups.

(Whereas for K=4, 5 or 7, some food groups are repeated).

Hence, I think we should select K=6.

```
%Question 5
data = readtable('spectral.csv');
x1 = data.Var1;
x2 = data.Var2;
Y = data.Var3;
Y(Y==1) = 2;
Y(Y==0) = 1;
X = [x1 x2];

disp('5a');
figure(1)
disp('Scatter Plot')
gscatter(x1, x2, Y);
title('Scatter Plot')
snapnow;

disp('5b')
figure(2);
[idx, C] = kmeans(X, 2, 'Start', 'Sample', 'Replicates',100, 'MaxIter', 200);
disp('Scatter Plot')
gscatter(x1, x2, idx);
xlabel('x1');
ylabel('x2');
title('Scatter Plot')
hold on;
plot(C(:,1), C(:,2), '.k', 'MarkerSize', 15);
title(strcat('kmeans, K=2'));
legend('1','2','centroids')
snapnow;

print_mis_classification(idx, Y);
disp("5c) Spectral Clustering");

figure(3);
A_1 = build_adjacency_matrix(X, 0.5);
G = graph(A_1);
plot(G, 'XData',X(:,1), 'YData',X(:,2));
xlabel('x1');
ylabel('x2');
title('e-neighbourhood graph , e=0.5. ');
snapnow;

figure(4);
A_2 = build_adjacency_matrix(X, 1);
G = graph(A_2);
plot(G, 'XData',X(:,1), 'YData',X(:,2));
xlabel('x1');
ylabel('x2');
title('e-neighbourhood graph , e=1. ');
snapnow;

figure(5);
A_3 = build_adjacency_matrix(X, 2);
G = graph(A_3);
plot(G, 'XData',X(:,1), 'YData',X(:,2));
xlabel('x1');
ylabel('x2');
title('e-neighbourhood graph , e=2. ');
snapnow;

figure(6);
[idx, C] = perform_spectral_clustering(A_1, 2);
```

```
gscatter(x1, x2, idx);
xlabel('x1');
ylabel('x2');
hold on;
title(strcat('Scatter Plot, K=2, e=0.5'));
legend('1','2');
snapnow;
print_mis_classification(idx, Y);

figure(7);
[idx, C] = perform_spectral_clustering(A_2, 2);
gscatter(x1, x2, idx);
xlabel('x1');
ylabel('x2');
hold on;
title(strcat('Scatter Plot, K=2, e=1'));
legend('1','2');
snapnow;
print_mis_classification(idx, Y);

figure(8);
[idx, C] = perform_spectral_clustering(A_3, 2);
gscatter(x1, x2, idx);
xlabel('x1');
ylabel('x2');
hold on;
title(strcat('Scatter Plot, K=2, e=2'));
legend('1','2');
snapnow;
print_mis_classification(idx, Y);

disp('5d')
A_1 = build_gaussian_sim_graph(X, 0.1);

figure(9);
[idx, C] = perform_spectral_clustering(A_1, 2);
gscatter(x1, x2, idx);
xlabel('x1');
ylabel('x2');
hold on;
title(strcat('spectral clustering, K=2, sigma=0.1'));
legend('1','2');
snapnow;
print_mis_classification(idx, Y);

A_2 = build_gaussian_sim_graph(X, 0.5);

figure(10);
[idx, C] = perform_spectral_clustering(A_2, 2);
gscatter(x1, x2, idx);
xlabel('x1');
ylabel('x2');
hold on;
title(strcat('spectral clustering, K=2, sigma=0.5'));
legend('1','2');
snapnow;
print_mis_classification(idx, Y);

figure(11);
[idx, C] = perform_spectral_clustering(A_3, 2);
gscatter(x1, x2, idx);
xlabel('x1');
ylabel('x2');
```

```

hold on;
title(strcat('spectral clustering, K=2, sigma=1'));
legend('1','2');
snapnow;
print_mis_classification(idx, Y);

figure(12);
A_3 = build_gaussian_sim_graph(X, 1);
G = graph(A_3);
plot(G, 'XData', X(:,1), 'YData', X(:,2));
xlabel('x1');
ylabel('x2');
title('e-neighbourhood graph');
snapnow;

function [idx, C] = perform_spectral_clustering(W, k)
    D = diag(sum(W,2));
    L = D - W;
    %[V, D] = eig(L);
    %eigenvalues = diag(D);
    %[~, id] = mink(eigenvalues, k)
    [U,~] = eigs(L,k,1e-13);
    %U = V(:,id);
    [idx, C] = kmeans(U, k, 'Start', 'Sample', 'Replicates', 100, 'MaxIter', 500);
end

function A=build_adjacency_matrix(X, e)
    sz = size(X,1);
    A = zeros(sz, sz);
    for i=1:sz
        for j=i+1:sz
            dist = norm(X(i,:)-X(j,:));
            if dist <= e
                A(i,j) = 1;
                A(j,i) = 1;
            end
        end
    end
end

function A=build_gaussian_sim_graph(X, sigma)
    sz = size(X,1);
    A = zeros(sz, sz);
    for i=1:sz
        for j=i+1:sz
            w = exp(-(norm(X(i,:)-X(j,:))^2/(2*sigma^2)));
            A(i,j) = w;
            A(j,i) = w;
        end
    end
end

function print_mis_classification(pred, Y)
    sz_1 = sum(Y==1);
    sz_2 = sum(Y==2);
    C(1) = sum(pred(1:sz_1,')==1);
    C(2) = sum(pred(1:sz_1,')==2);
    [max_val, id] = max(C);
    mishit_1 = abs(sz_1-max_val)/sz_1;
    if id == 1
        mishit_2 = abs(sz_2 - sum(pred(sz_2:end,')==2))/sz_2;
    else
        mishit_2 = abs(sz_2 - sum(pred(sz_2:end,')==1))/sz_2;
    end

    fprintf("Incorrectly classified fraction for category 1 = %f \n", mishit_1);

```

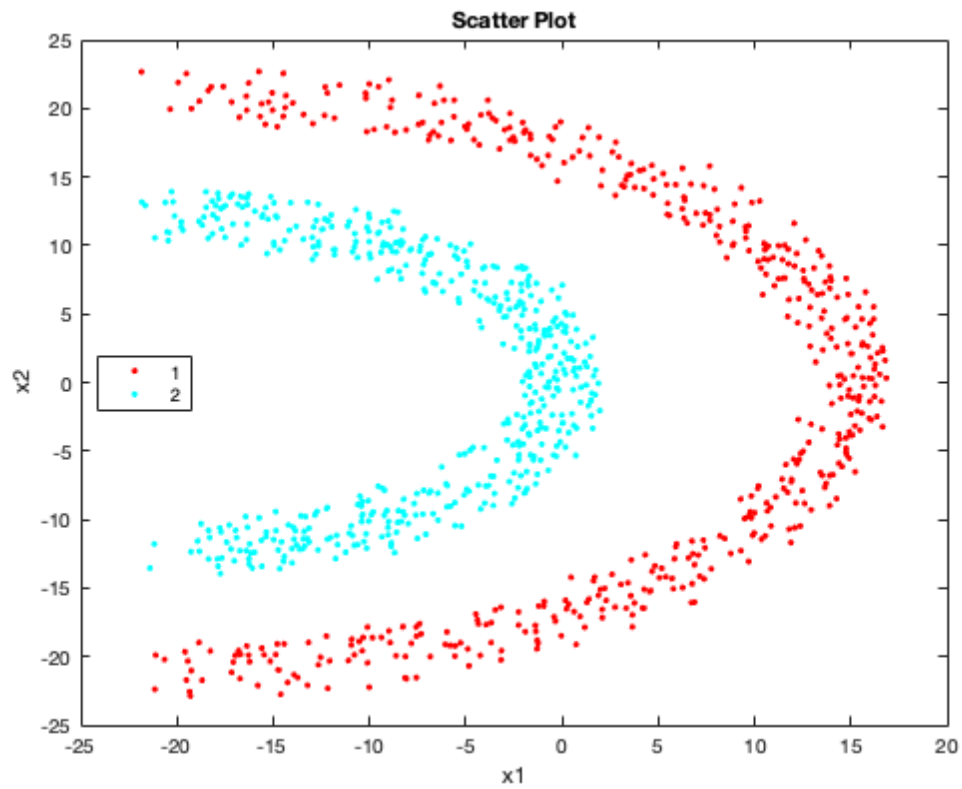


```
fprintf("Incorrectly classified fraction for category 2 = %f \n", mishit_2);
```

```
end
```

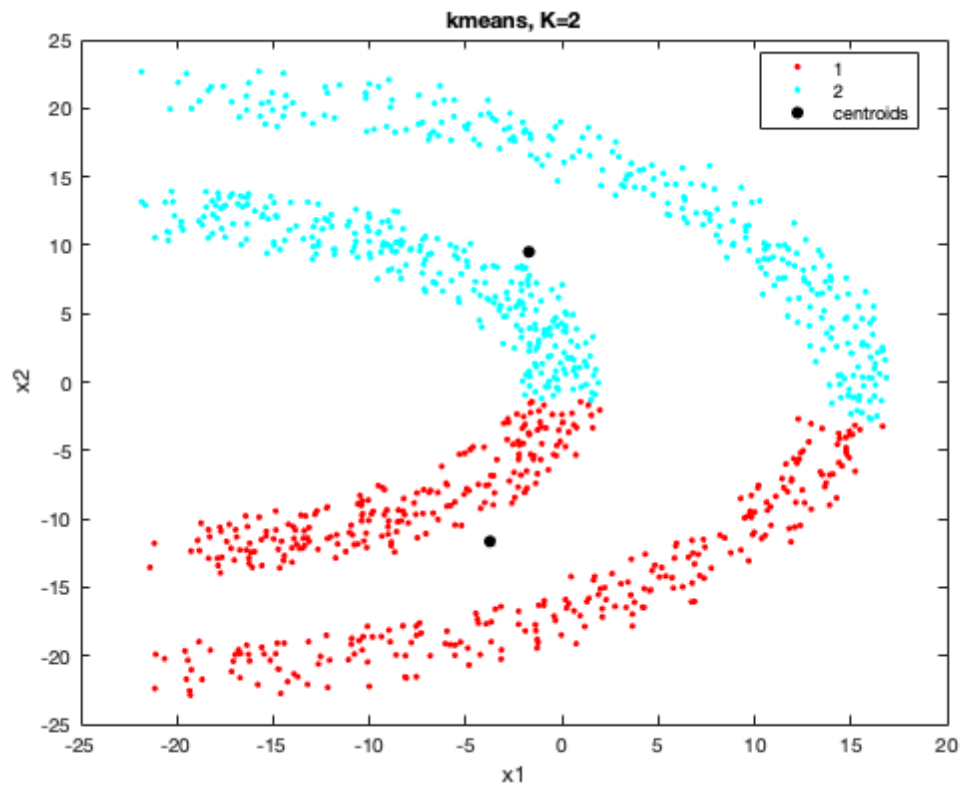
5a)

Scatter Plot



5b)

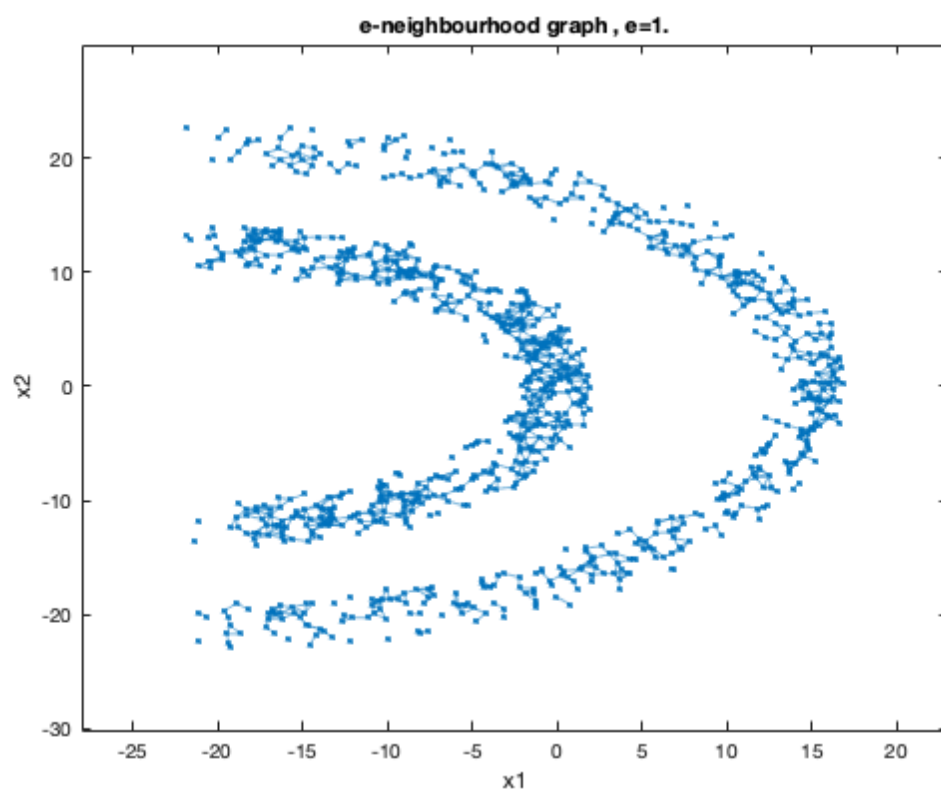
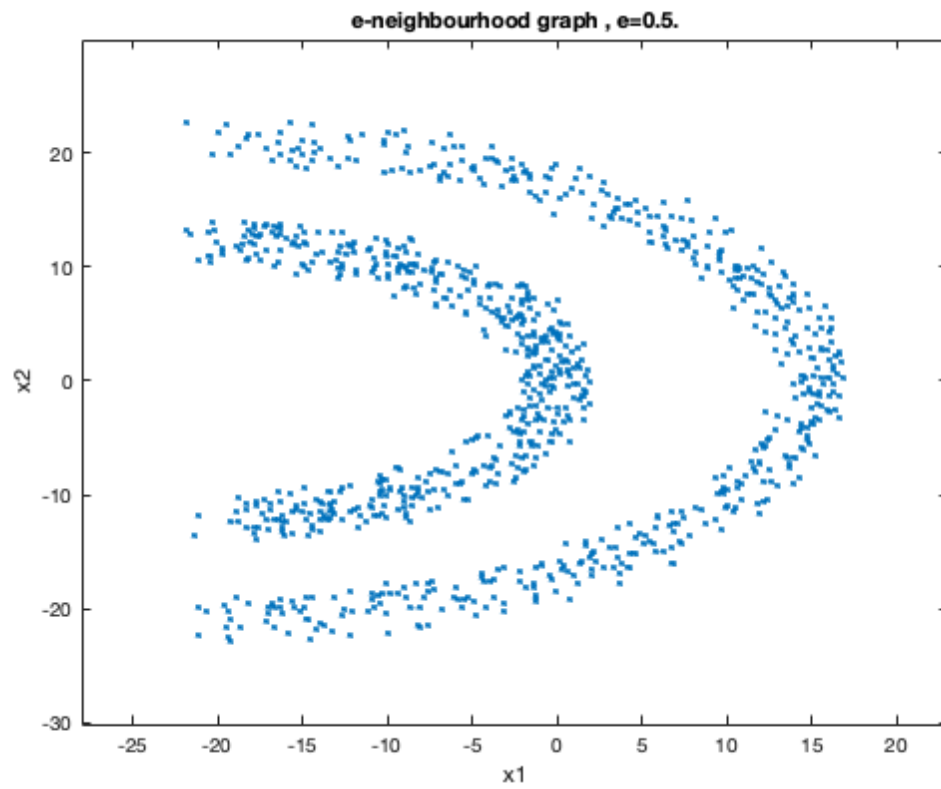
Scatter Plot

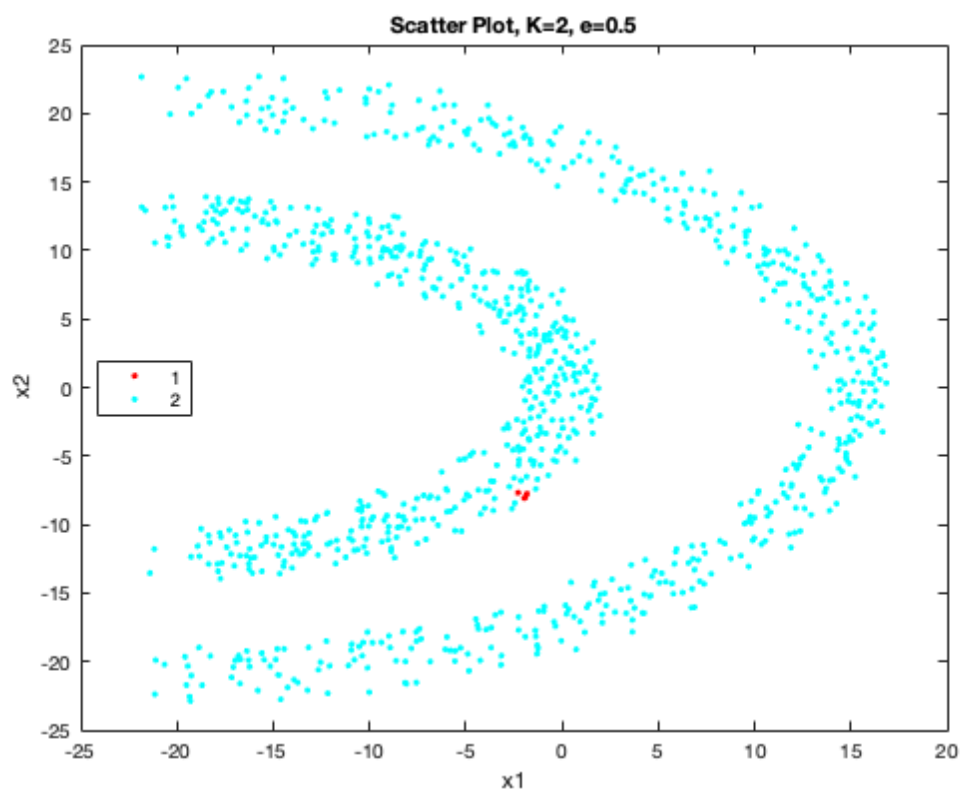
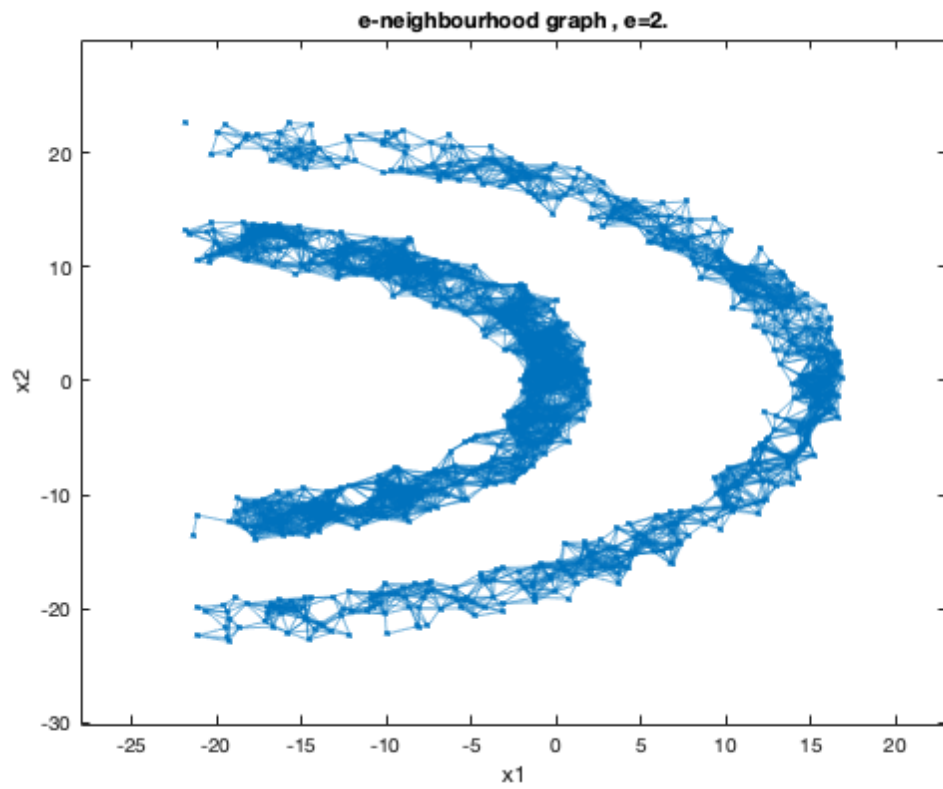


Incorrectly classified fraction for category 1 = 0.430000

Incorrectly classified fraction for category 2 = 0.540000

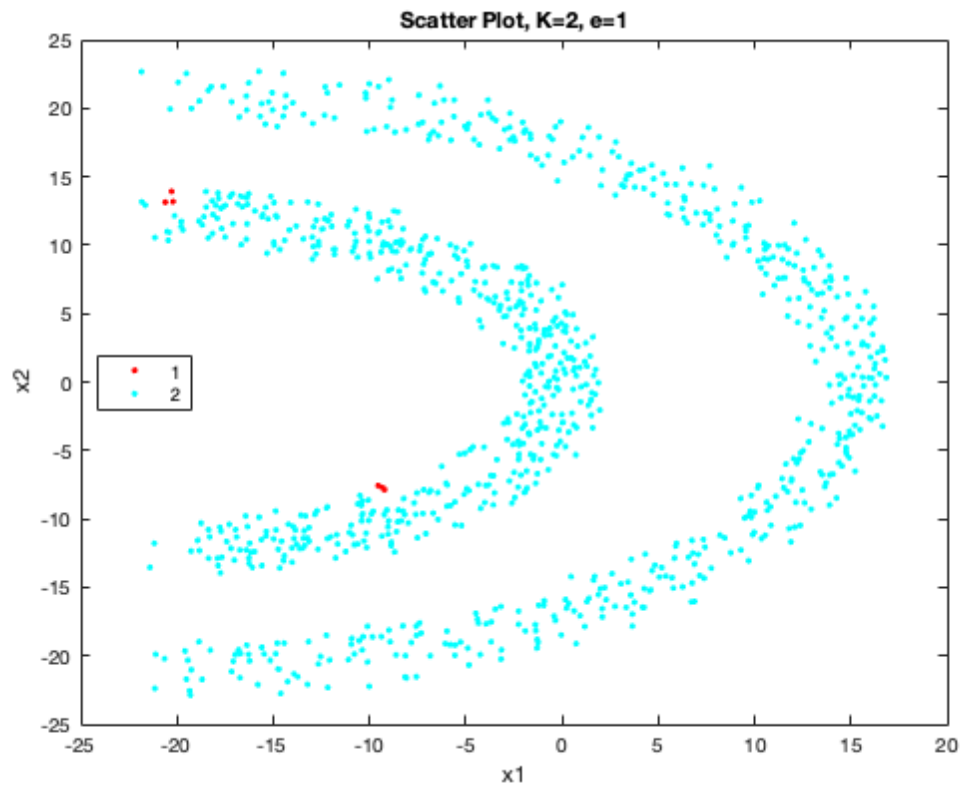
5c) Spectral Clustering



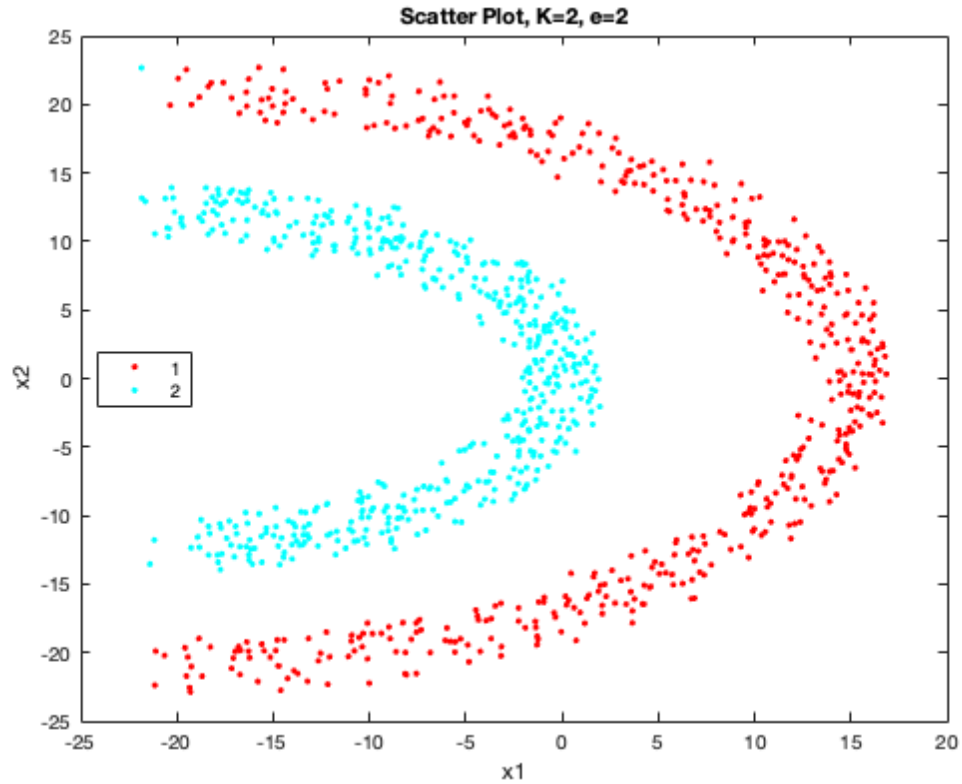


Incorrectly classified fraction for category 1 = 0.008000

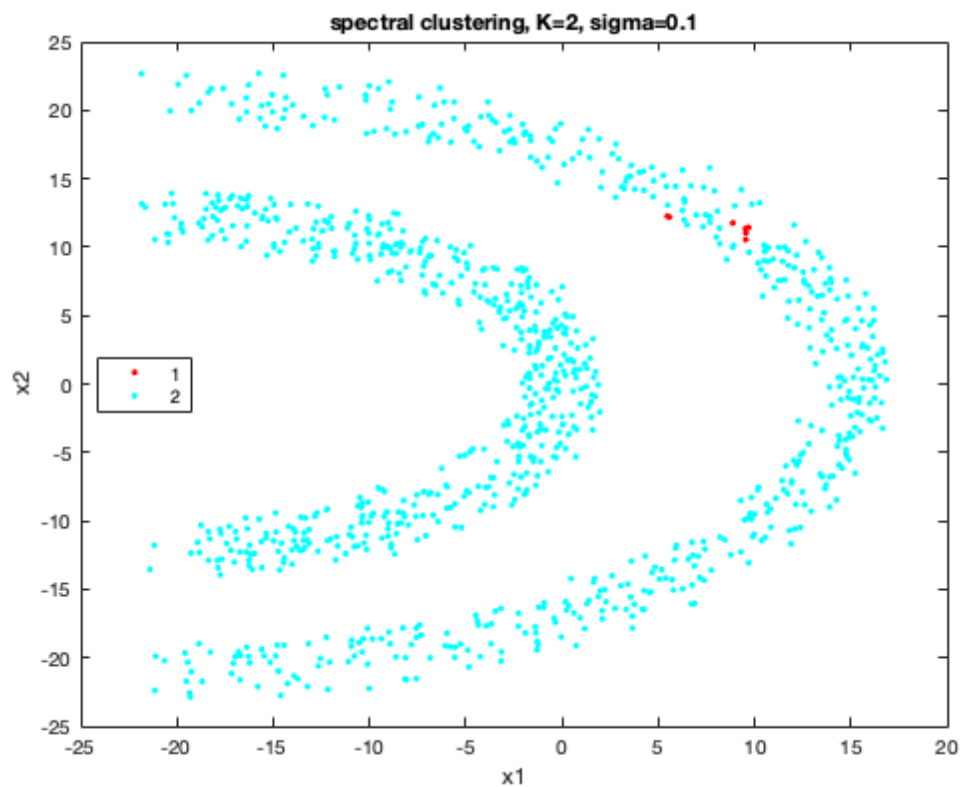
Incorrectly classified fraction for category 2 = 1.000000



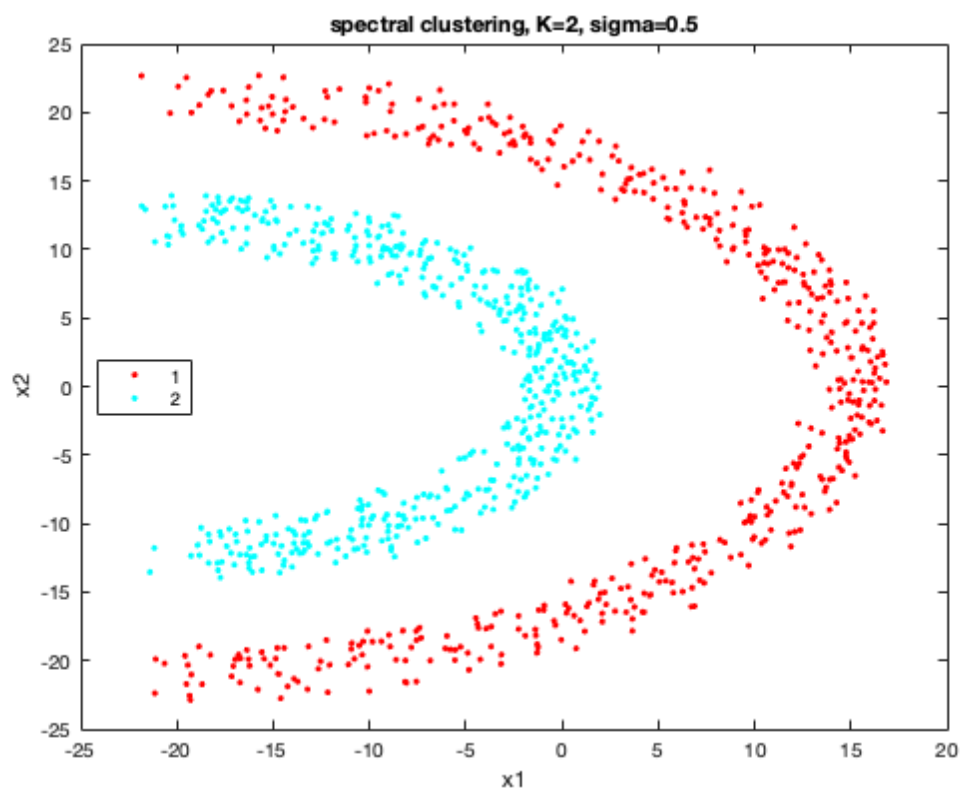
Incorrectly classified fraction for category 1 = 0.012000  
 Incorrectly classified fraction for category 2 = 1.000000



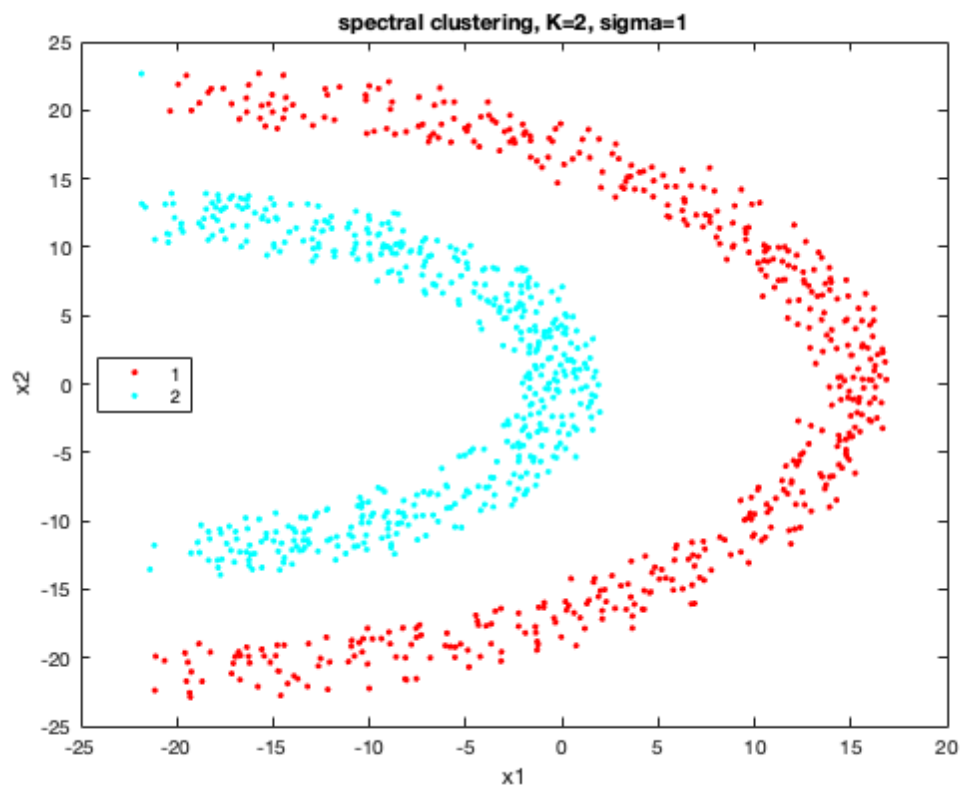
Incorrectly classified fraction for category 1 = 0.000000  
 Incorrectly classified fraction for category 2 = 0.002000  
 5d)



Incorrectly classified fraction for category 1 = 0.000000  
Incorrectly classified fraction for category 2 = 0.986000



Incorrectly classified fraction for category 1 = 0.000000  
Incorrectly classified fraction for category 2 = 0.000000



Incorrectly classified fraction for category 1 = 0.000000  
Incorrectly classified fraction for category 2 = 0.002000

