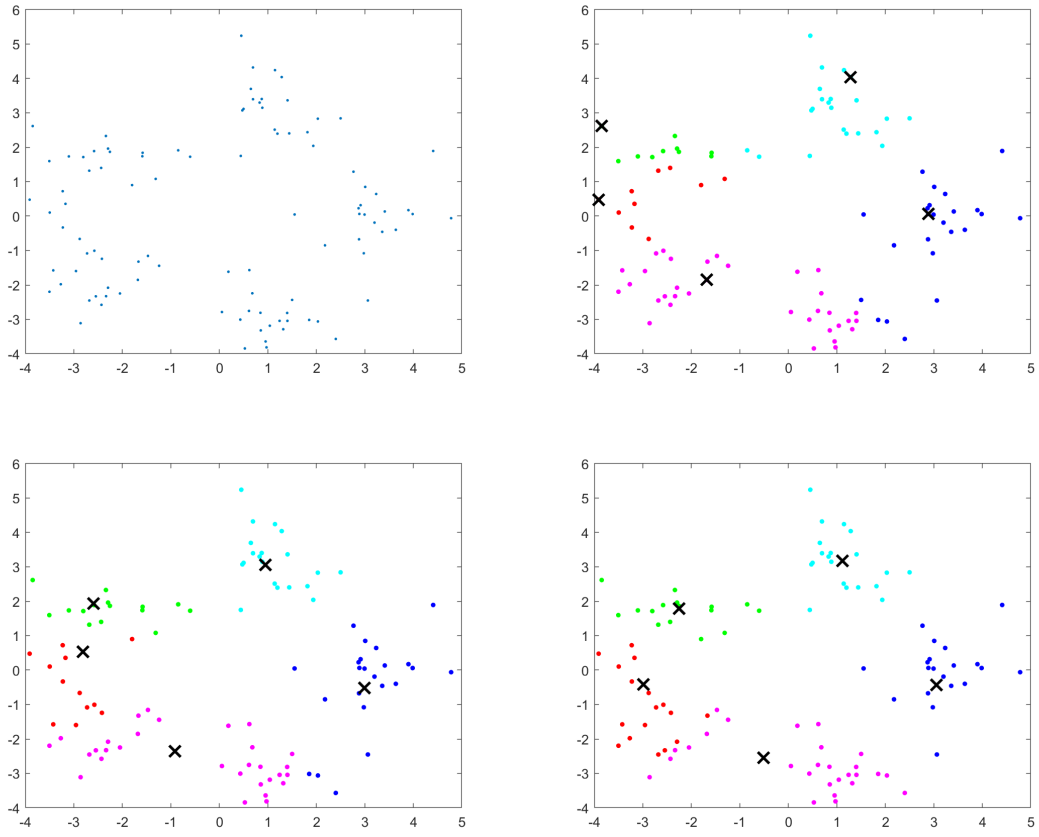
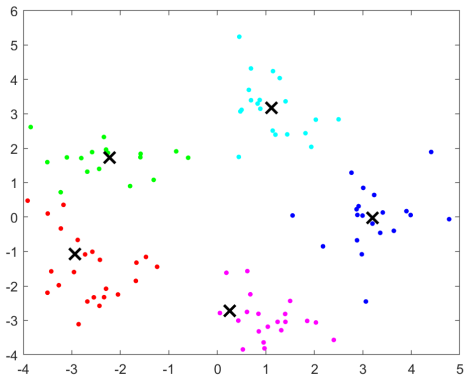
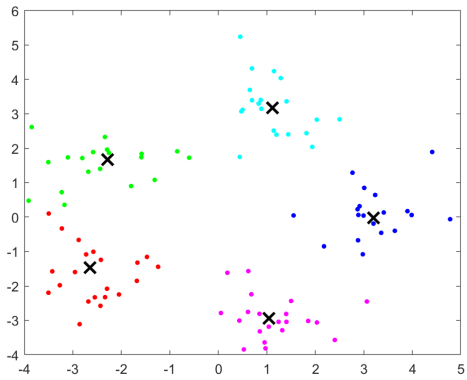
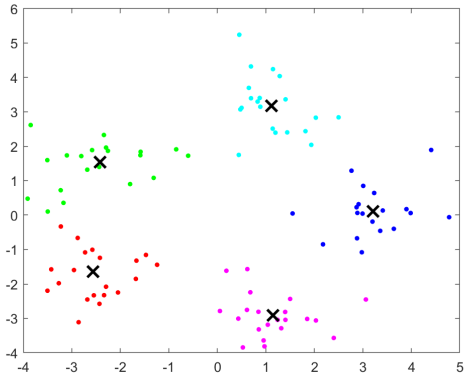
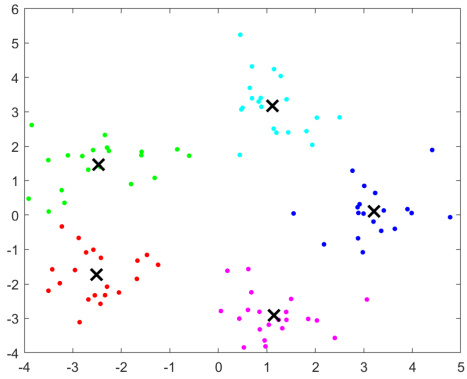
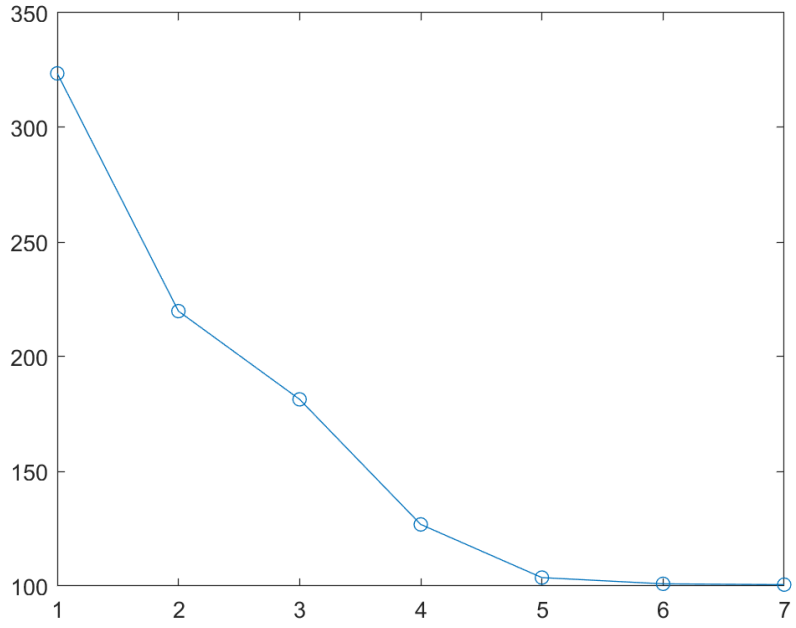
MATLAB Problems 1 and 2

Output:

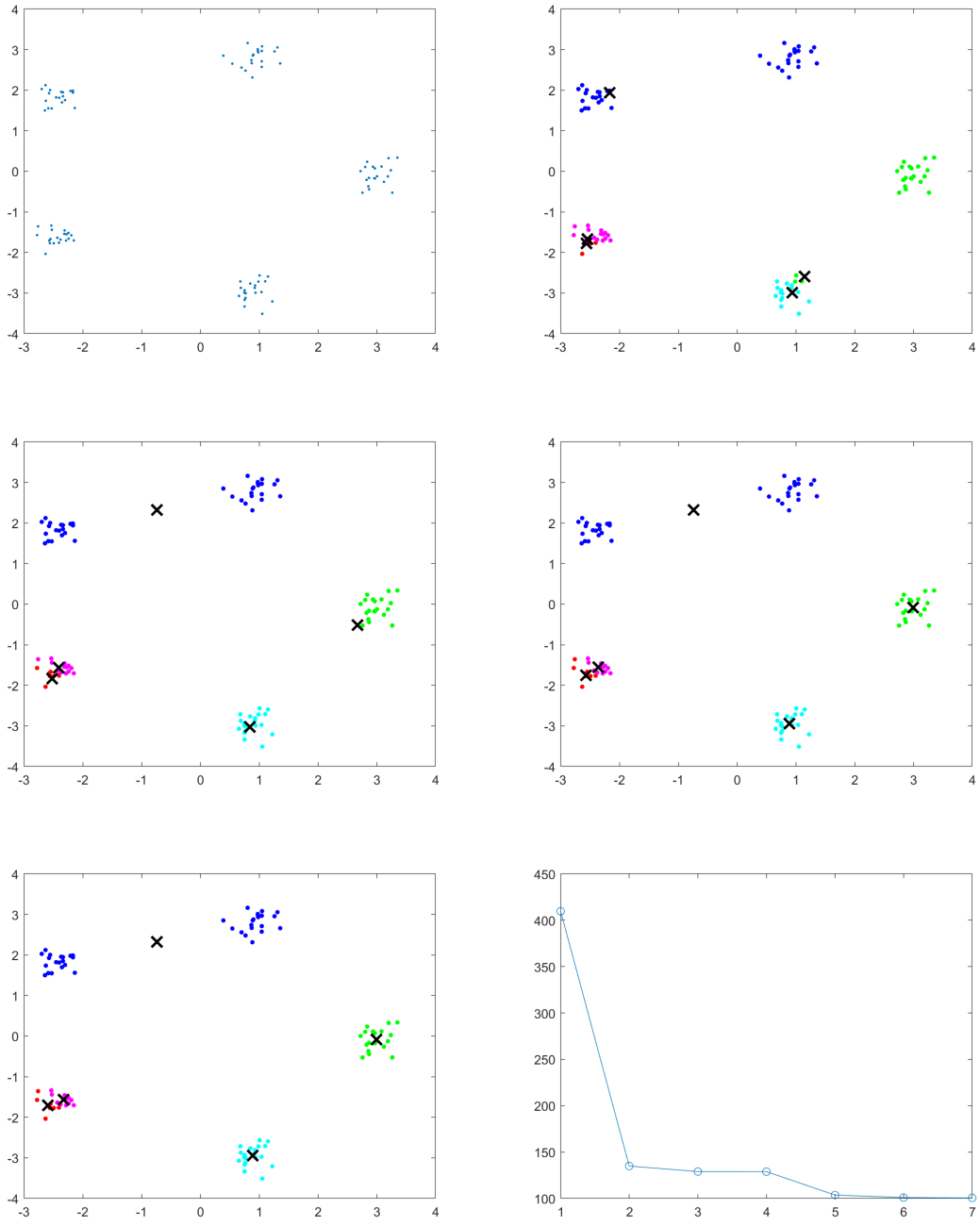
```
------------------------------------------------------------------------
         Oscar Martinez          HW 7          STA 5106
------------------------------------------------------------------------
-----------Problem 1----------

----------Dataset 1---------

----------Dataset 2---------

-----------Problem 2----------

----------Dataset 1---------
mu = (0.0254, 4.9720), sigma = (0.8458, 0.2694), alpha = (0.3240, 0.6760)

----------Dataset 2---------
mu = (3.1296, 1.7358), sigma = (0.1763, 0.4410), alpha = (0.7320, 0.2680)
```

Figures for Problem 1, Dataset 1



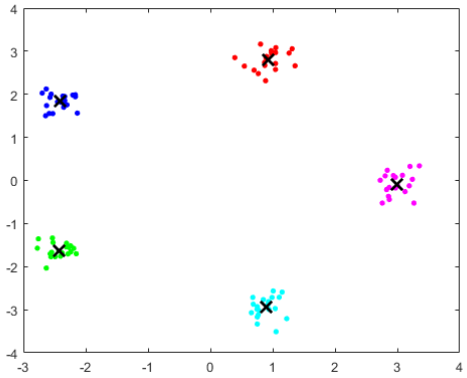





350
300
250
200
150
100
1
2
3
4
5
6
7

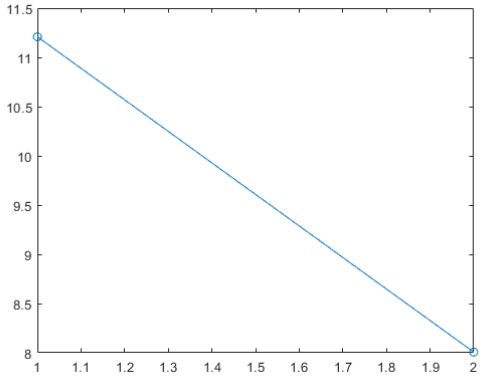
Figures for Problem 1, Dataset 2



Figures for Problem 1, Dataset 2, nonrandom seeds

Here, I picked one member of each cluster and used them as seeds. These members were: (0.9852, 2.935), (-2.4, 1.814), (-2.193, -1.576), (0.8488, -2.765), (3.123, -0.259). In this case, there was faster convergence (two iterations) and less squared error.



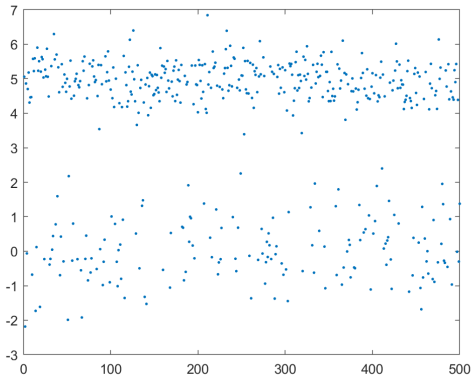
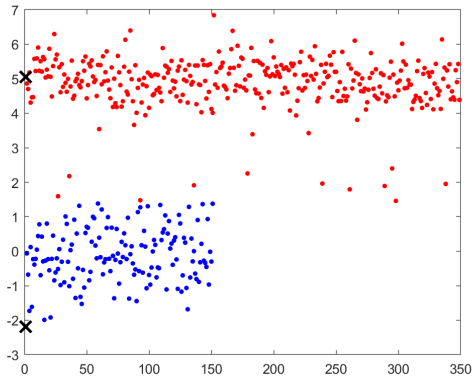
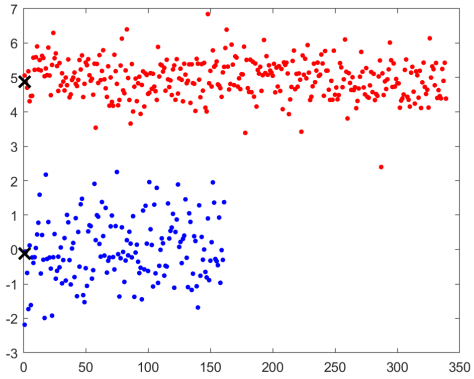
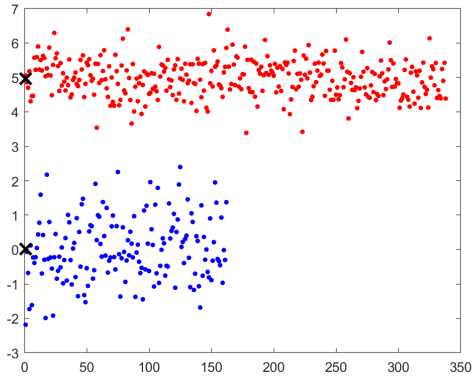
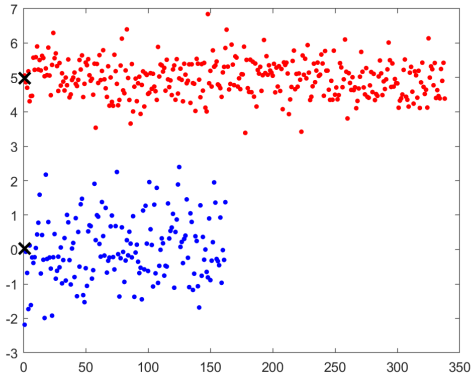
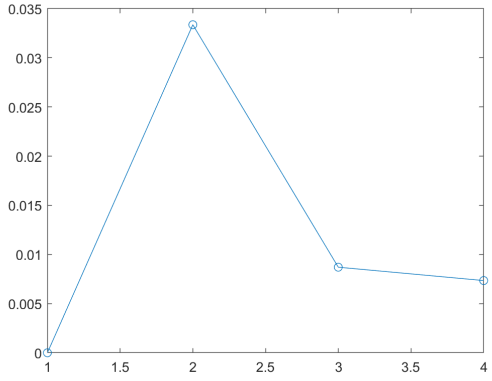


Output for Homework 6

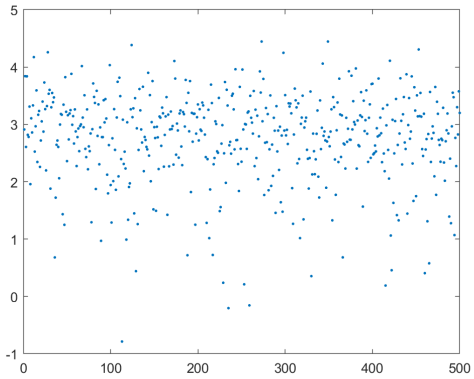
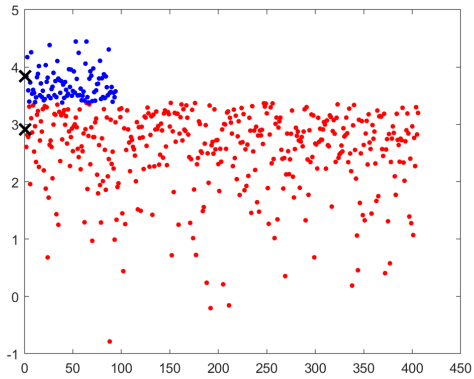
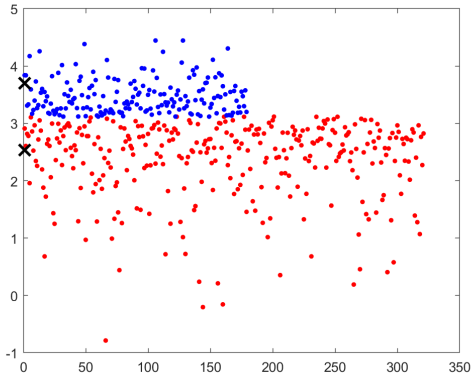
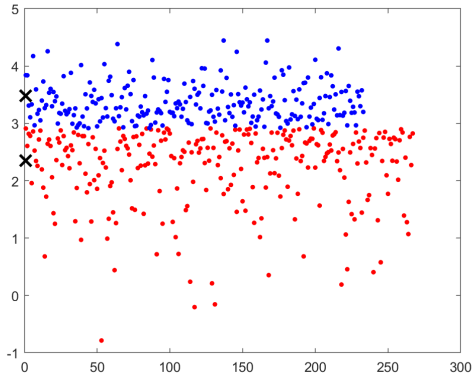
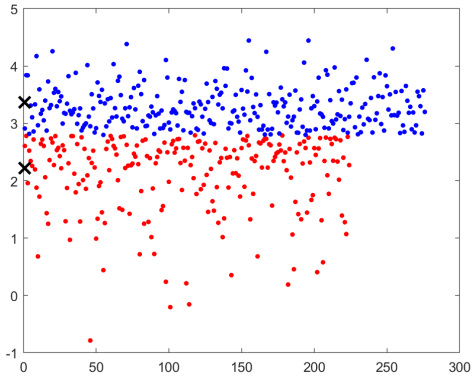
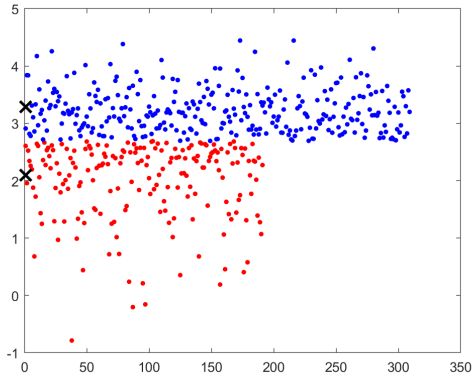
```
------------------------------------------------------------
Oscar Martinez          Homework 6: Problems 1 and 2          STA 5106
------------------------------------------------------------
----------Problem 1----------

theta =

  Columns 1 through 5

    0.1000    0.2226    0.2369    0.2384    0.2385

  Columns 6 through 8

    0.2385    0.2385    0.2385

----------Problem 2----------
----------Dataset 1---------

m =

     7

mu = (0.023, 4.970), sigma = (0.914, 0.522), alpha = (0.324, 0.676)

----------Dataset 2---------

m =

    29

mu = (0.550, 2.818), sigma = (0.722, 0.700), alpha = (0.027, 0.973)
```

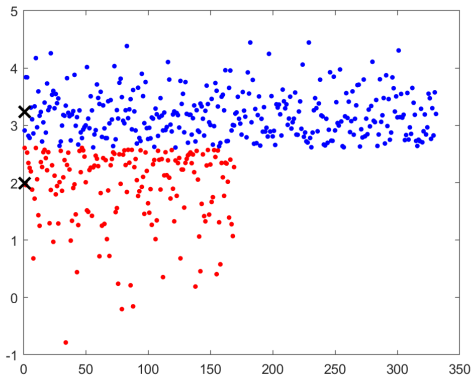
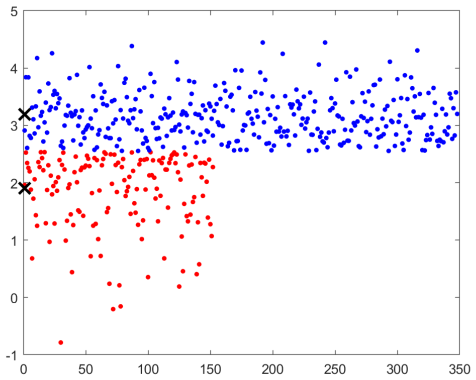
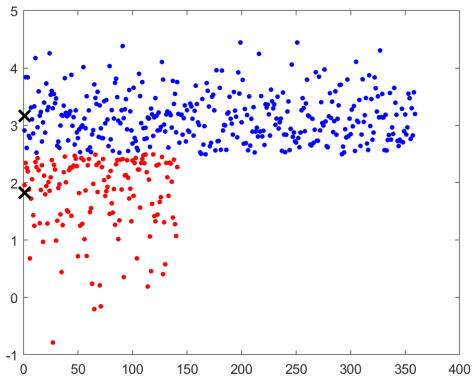
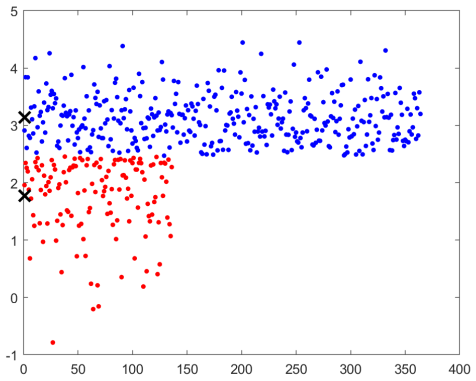
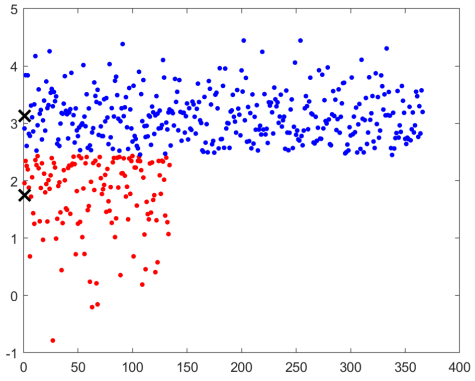
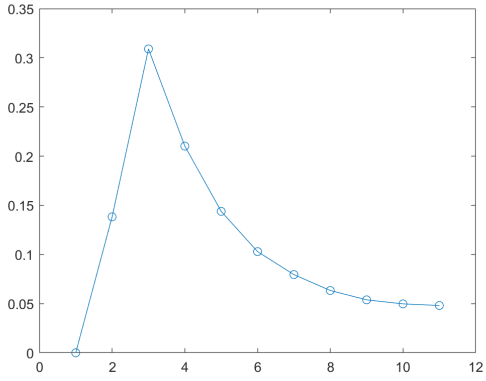
As can be seen, the mu's and alpha's for data set 1 are similar, however the K-Mean variances are much smaller. In contrast, the results for the second data set were not similar at all. This was after using several different `seeds` (initial values) such as extremal elemens, using the proposed means from homework 6, or using the means plus or minus 0.25 of the entire dataset.

Figures for Problem 2, Dataset 1

Figures for Problem 2, Dataset 2

Code:

```
clc
clear
%Diary
dfile ='MATLAB_Output_OM.txt';
if exist(dfile, 'file') ; delete(dfile); end
diary(dfile)
diary on
```

```

%Introduction
fprintf('
   ------------------------------------------------------------------------\n'
   );
fprintf('\t Oscar Martinez \t HW 7 \t STA 5106\n');
fprintf('
   ------------------------------------------------------------------------\n'
   );

%------Problem 1:------
fprintf('----------Problem 1----------\n');

fprintf('\n----------Dataset 1----------\n');
%Load Data
load hw7_1_data1.mat %Loads the first dataset
X=Yn'; %X is 100*2
[N, I] = size(X); %N=100,n=I

%Visualize original data
figure(1);
plot(X(:,1), X(:,2), '.');
pause;

K = 5;    % number of clusters
C(:,:,1) = X(1:K,:);  % assign the first K points of X as the means

E = 1;  % update error
m = 1;
while (E > 1e-3)
    for n = 1:N %Find closest K-Mean for each n \in N of X = N x I
        dis = sqrt(sum((ones(K,1)*X(n,:) - C(:,:,m)).^2,2));
        [min_dis(m,n), ind(m,n)] = min(dis);
    end
    for k = 1:K
        C(k,:,m+1) = mean(X(ind(m,:)==k,:)); %Update Means
    end
    E = norm(C(:,:,m+1)-C(:,:,m)); %Difference between mean iterations

    % plot the process
    figure(2); clf;
    color = 'rbgcmyk';
    for k = 1:K
        plot(X(ind(m,:)==k,1),X(ind(m,:)==k,2),[color(k) '.'],'MarkerSize'
           ,12);
```

```
        hold on;
        plot(C(k,1,m),C(k,2,m),'kx', 'MarkerSize',12,'LineWidth',2)
    end

    % compute the sum of squares
    ss(m) = sum(min_dis(m,:).^2);

    pause;
    m = m+1;
end

figure(3);
plot(ss, 'o-');

%---Load Second Data---
fprintf('\n----------Dataset 2----------\n');
load hw7_1_data2.mat %Loads the second dataset
X=Yn'; %X is 100*2
[N, I] = size(X); %m=100,n=2

%Visualize original data
figure(4);
plot(X(:,1), X(:,2), '.');
pause;

K = 5;    % number of clusters
C(:,:,1) = X(1:K,:);  % assign the first K points of X as the means
%C(:,:,1) = [0.9852 2.935; -2.4 1.814; -2.193 -1.576; 0.8488 -2.765; 3.123
%-0.259] non-random seed

E = 1;  % update error
m = 1;
while (E > 1e-3)
    for n = 1:N %Find closest K-Mean for each n \in N of X = N x I
        dis = sqrt(sum((ones(K,1)*X(n,:) - C(:,:,m)).^2,2));
        [min_dis(m,n), ind(m,n)] = min(dis);
    end
    for k = 1:K
        C(k,:,m+1) = mean(X(ind(m,:)==k,:)); %Update Means
    end
    E = norm(C(:,:,m+1)-C(:,:,m)); %Difference between mean iterations

    % plot the process
    figure(5); clf;
    color = 'rbgcmyk';
```

```
    for k = 1:K
        plot(X(ind(m,:)==k,1),X(ind(m,:)==k,2),[color(k) '.'],'MarkerSize'
            ,12);
        hold on;
        plot(C(k,1,m),C(k,2,m),'kx', 'MarkerSize',12,'LineWidth',2)
    end

    % compute the sum of squares
    ss(m) = sum(min_dis(m,:).^2);

    pause;
    m = m+1;
end

figure(6);
plot(ss, 'o-'); %Plot Error

%------Problem 2:------
fprintf('\n----------Problem 2----------\n');
clear;
fprintf('\n----------Dataset 1----------\n');
%Load the first dataset
load hw6_2_data1.mat;
X = Y'; %500 x 1

[N, I] = size(X); %N=500,I=1

%Visualize original data
figure(7);
plot(X(:), '.');
pause;

K = 2;    % number of clusters
C(:,1) = X(1:K);  % assign the first K points of X as the means

E = 1;  % update error
m = 1;  %Iteration
while (E > 1e-3)
    for n = 1:N %Find closest K-Mean for each n \in N of X = N x I
        dis = sqrt(sum((ones(K,1)*X(n) - C(:,m)).^2,2)); %distance between
            members of X and proposed means
        [min_dis(m,n), ind(m,n)] = min(dis); %Min dist and corresponding
            index between element n of X for iteration n
    end
    for k = 1:K
```

```
135         C(k,m+1) = mean(X(ind(m,:)==k)); %Update Means via responsibilities
                for K
136     end
137     E = norm(C(:,m+1)-C(:,m)); %Difference between mean iterations
138
139     % plot the process
140     figure(8); clf;
141     color = 'rbgcmyk';
142     for k = 1:K
143         plot(X(ind(m,:)==k),[color(k) '.'],'MarkerSize',12);
144         hold on;
145         plot(C(k,m),'kx', 'MarkerSize',12,'LineWidth',2)
146     end
147
148     % compute the sum of squares
149     ss(m) = sum(min_dis(m).^2);
150
151     pause;
152     m = m+1;
153 end
154
155 figure(9);
156 plot(ss, 'o-'); %Plot Error
157
158 %Mean, Variance,and Proportion
159 Mu1 = C(1,m-1);
160 Mu2 = C(2,m-1);
161 Var1 = var(X(ind(m-1,:)==1));
162 Var2 = var(X(ind(m-1,:)==2));
163 Alpha1 = sum(ind(m-1,:)==1)/N;
164 Alpha2 = sum(ind(m-1,:)==2)/N;
165 theta = [Mu2 Mu1 Var2 Var1 Alpha2 Alpha1];
166 fprintf('mu = (%2.4f, %2.4f), sigma = (%2.4f, %2.4f), alpha = (%2.4f, %2.4f)
        \n', theta(1), theta(2), theta(3), theta(4), theta(5), theta(6));
167
168
169
170 fprintf('\n----------Dataset 2----------\n');
171 clear;
172 %Load the second dataset
173 load hw6_2_data2.mat;
174 X = Y'; %500 x 1
175
176 [N, I] = size(X); %N=500,I=1
177
```

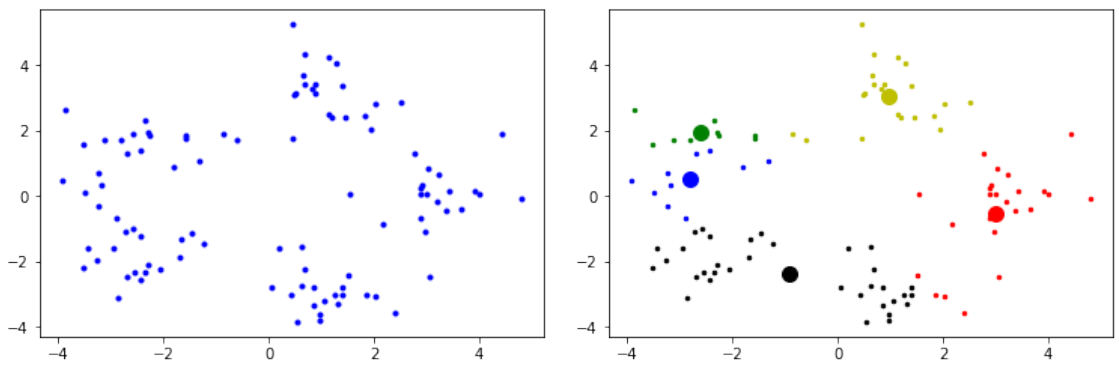
```
%Visualize original data
figure(10);
plot(X(:), '.');
pause;

K = 2;    % number of clusters
C(:,1) = X(1:K);  % assign the first K points of X as the means
%C(:,1) = [min(X); max(X)]; %Extremal Seeds
%C(:,1) = [mean(X)+0.25; mean(X)-0.25]; %Mean seeds
%C(:,1) = [2.818; 0.550]; %HW6 Seeds

E = 1;  % update error
m = 1;  %Iteration
while (E > 1e-6)
    for n = 1:N %Find closest K-Mean for each n \in N of X = N x I
        dis = sqrt(sum((ones(K,1)*X(n) - C(:,m)).^2,2)); %distance between
           members of X and proposed means
        [min_dis(m,n), ind(m,n)] = min(dis); %Min dist and corresponding
           index between element n of X for iteration n
    end
    for k = 1:K
        C(k,m+1) = mean(X(ind(m,:)==k)); %Update Means via responsibilities
           for K
    end
    E = norm(C(:,m+1)-C(:,m)); %Difference between mean iterations

    % plot the process
    figure(11); clf;
    color = 'rbgcmyk';
    for k = 1:K
        plot(X(ind(m,:)==k),[color(k) '.'],'MarkerSize',12);
        hold on;
        plot(C(k,m),'kx', 'MarkerSize',12,'LineWidth',2)
    end

    % compute the sum of squares
    ss(m) = sum(min_dis(m).^2);

    pause;
    m = m+1;
end

figure(12);
plot(ss, 'o-'); %Plot Error

```

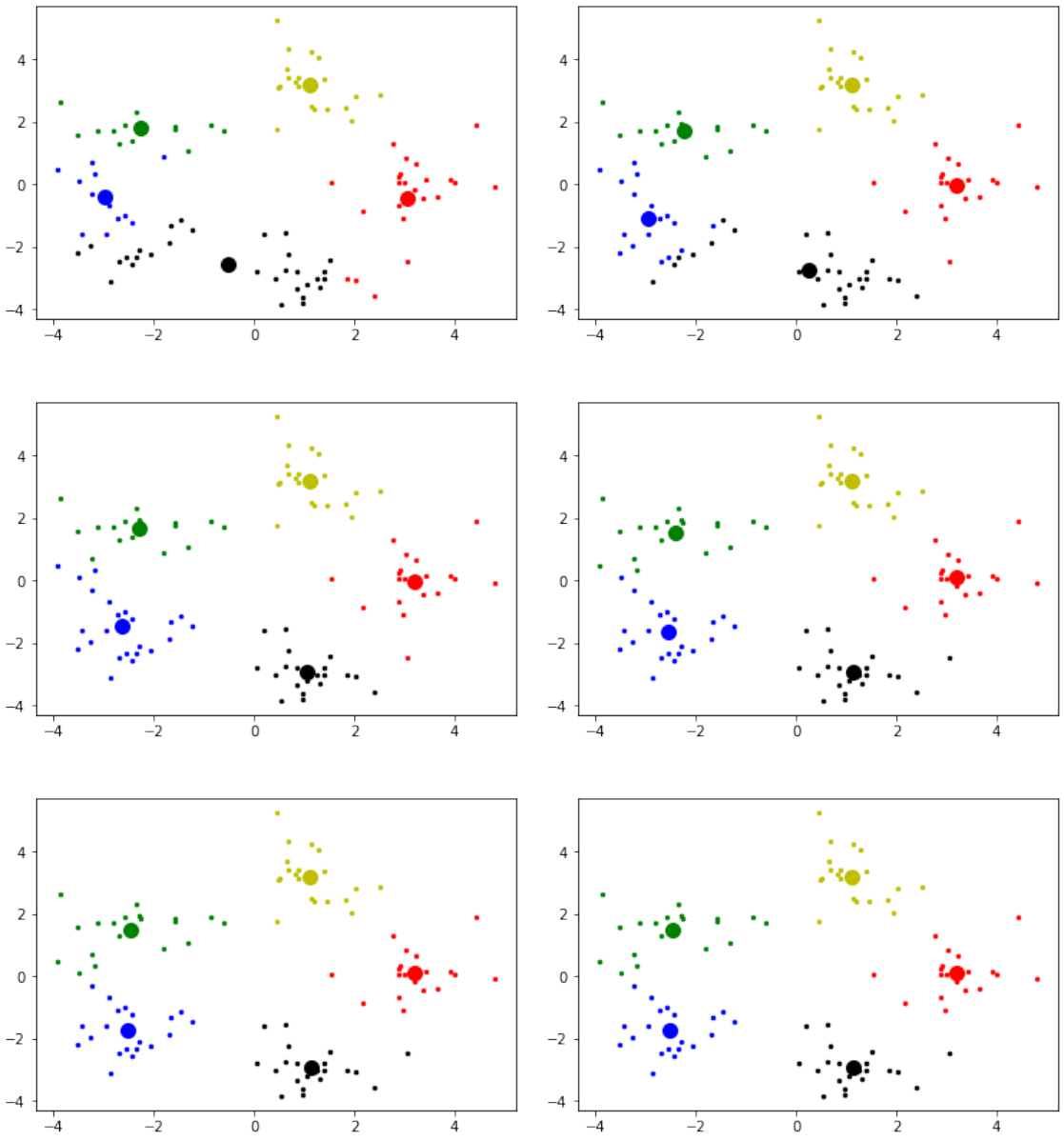
```
%Mean, Variance,and Proportion
Mu1 = C(1,m-1);
Mu2 = C(2,m-1);
Var1 = var(X(ind(m-1,:)==1));
Var2 = var(X(ind(m-1,:)==2));
Alpha1 = sum(ind(m-1,:)==1)/N;
Alpha2 = sum(ind(m-1,:)==2)/N;
theta = [Mu2 Mu1 Var2 Var1 Alpha2 Alpha1];
fprintf('mu = (%2.4f, %2.4f), sigma = (%2.4f, %2.4f), alpha = (%2.4f, %2.4f)
    \n', theta(1), theta(2), theta(3), theta(4), theta(5), theta(6));

diary off
```

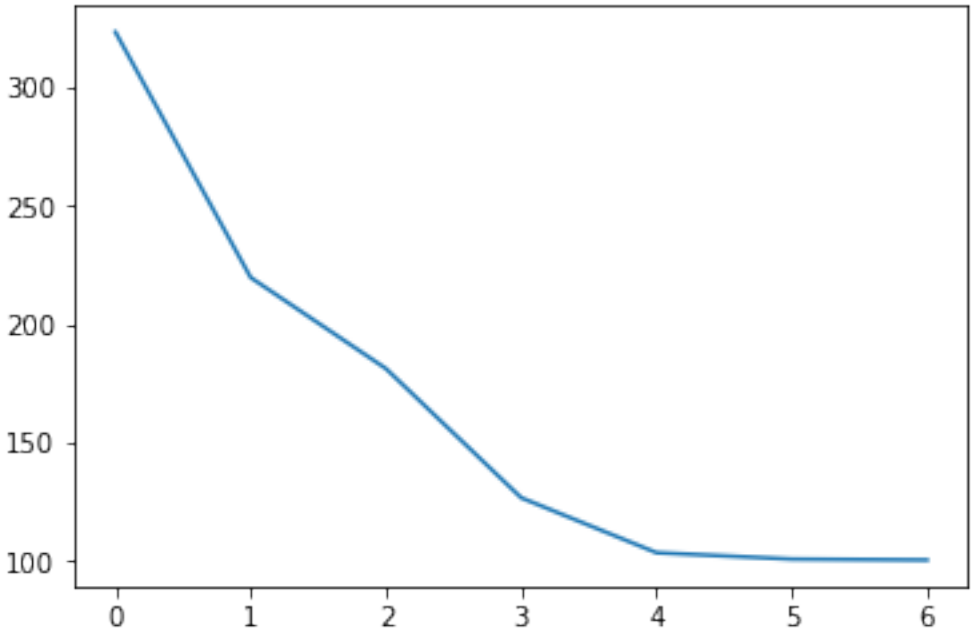
Problem 3

```
In [12]: ### K-Means Algorithm

from numpy import *
from matplotlib import pyplot
import time
import scipy.io

mat_contents=scipy.io.loadmat('hw7_1_data1.mat')
Y=mat(mat_contents['Yn'])
X=Y.T

(N,I)=shape(X)

pyplot.ion()    # allow to show figures without holding command lines

pyplot.figure(1)
pyplot.plot(X[:,0], X[:,1], 'b.')

K = 5    # number of clusters
C = X[0:K,:].copy()  # assign the first K points as the means

E = 1  # update error
m = 0
itr_max = 20

min_dis = zeros((itr_max,N))
ind = zeros((itr_max, N))
ss = zeros((itr_max))
```

```
CC = zeros((K, I, itr_max))
CC[:,:,0] = C

while (E > 1e-3):
for n in range(0,N):
dis = sqrt(sum(array(ones((K,1))*X[n] - C)**2, axis=1))
min_dis[m,n] = amin(dis)
ind[m,n] = argmin(dis)

for k in range(0,K):
C[k,:] = mean(X[ind[m,:] == k,:], axis=0)

CC[:,:,m+1] = C

E = linalg.norm(CC[:,:,m+1] - CC[:,:,m])
ss[m] = sum(min_dis[m,:]**2)

pyplot.figure(m+2)
#pyplot.clf()
cr = 'brgyk'
for k in range(0,K):
pyplot.plot(X[ind[m,:]==k,0], X[ind[m,:]==k,1], '.', \
color = cr[k], markersize = 5)
pyplot.plot(C[k,0], C[k,1], '*', color = cr[k], markersize = 10)

#        pyplot.show()

m = m+1

pyplot.figure(m+2)
pyplot.plot(range(0,m), ss[0:m])

pyplot.show()
```




4
2
0
−2
−4
−4
−2
0
2
4




```
In [13]: ### K-Means Algorithm

from numpy import *
from matplotlib import pyplot
import time
import scipy.io

mat_contents=scipy.io.loadmat('hw7_1_data2.mat')
Y=mat(mat_contents['Yn'])
X=Y.T

(N,I)=shape(X)

pyplot.ion()    # allow to show figures without holding command lines

pyplot.figure(1)
pyplot.plot(X[:,0], X[:,1], 'b.')

K = 5    # number of clusters
C = X[0:K,:].copy()  # assign the first K points as the means

E = 1  # update error
m = 0
itr_max = 20
```

```
min_dis = zeros((itr_max,N))
ind = zeros((itr_max, N))
ss = zeros((itr_max))

CC = zeros((K, I, itr_max))
CC[:,:,0] = C

while (E > 1e-3):
for n in range(0,N):
dis = sqrt(sum(array(ones((K,1))*X[n] - C)**2, axis=1))
min_dis[m,n] = amin(dis)
ind[m,n] = argmin(dis)

for k in range(0,K):
C[k,:] = mean(X[ind[m,:] == k,:], axis=0)

CC[:,:,m+1] = C

E = linalg.norm(CC[:,:,m+1] - CC[:,:,m])
ss[m] = sum(min_dis[m,:]**2)

pyplot.figure(m+2)
#pyplot.clf()
cr = 'brgyk'
for k in range(0,K):
pyplot.plot(X[ind[m,:]==k,0], X[ind[m,:]==k,1], '.', \
color = cr[k], markersize = 5)
pyplot.plot(C[k,0], C[k,1], '*', color = cr[k], markersize = 10)

#        pyplot.show()

m = m+1

pyplot.figure(m+2)
pyplot.plot(range(0,m), ss[0:m])

pyplot.show()
```