

---

```

% % AUTHOR: KWM
% Comment: writing my own for loop for k=2:10 or so proved difficult,
% where
% I ultimately gave up because starting at k=3 there were often a
% cluster
% centroid with no data points that were closer to it than any other
% centroid. This gave run time errors for empty variables and I didn't
% want to sink any more time in, instead using a community function
% to
% confirm that K=2, what was visually clear.

week1 = load('week1.mat'); week1 = week1.week1;
week2 = load('week2.mat'); week2 = week2.week2;
week3 = load('week3.mat'); week3 = week3.week3;
week4 = load('week4.mat'); week4 = week4.week4;
week5 = load('week5.mat'); week5 = week5.week5;

X = [week1; week2; week3; week4; week5];

y = X(:,5);
X = [X(:,2) X(:,3)];

minx = min(X(:,1));
maxx = max(X(:,1));
miny = min(X(:,2));
maxy = max(X(:,2));
center = [minx+(maxx-minx)/2, miny+(maxy-miny)/2];

[~,~,~,K_optimal_by_elbow] = kmeans_opt(X)

centroids = [[minx+(maxx-minx)*rand,miny+(maxy-miny)*rand],[minx
+(maxx-minx)*rand,miny+(maxy-miny)*rand]];

for iter=1:3
    dist_cent1 = X-centroids(1:2); dist_cent1 =
    sqrt(dist_cent1(:,1).^2+dist_cent1(:,2).^2);
    dist_cent2 = X-centroids(3:4); dist_cent2 =
    sqrt(dist_cent2(:,1).^2+dist_cent2(:,2).^2);

    cluster1_X = X(dist_cent1 < dist_cent2,:);
    cluster2_X = X(dist_cent1 >= dist_cent2,:);

    figure
    hold on;
    scatter(cluster1_X(:,1),cluster1_X(:,2),'red')
    scatter(cluster2_X(:,1),cluster2_X(:,2),'blue')
    scatter(centroids(1), centroids(2),'k','LineWidth',5)
    scatter(centroids(3), centroids(4),'k','LineWidth',5)
    hold off;

```

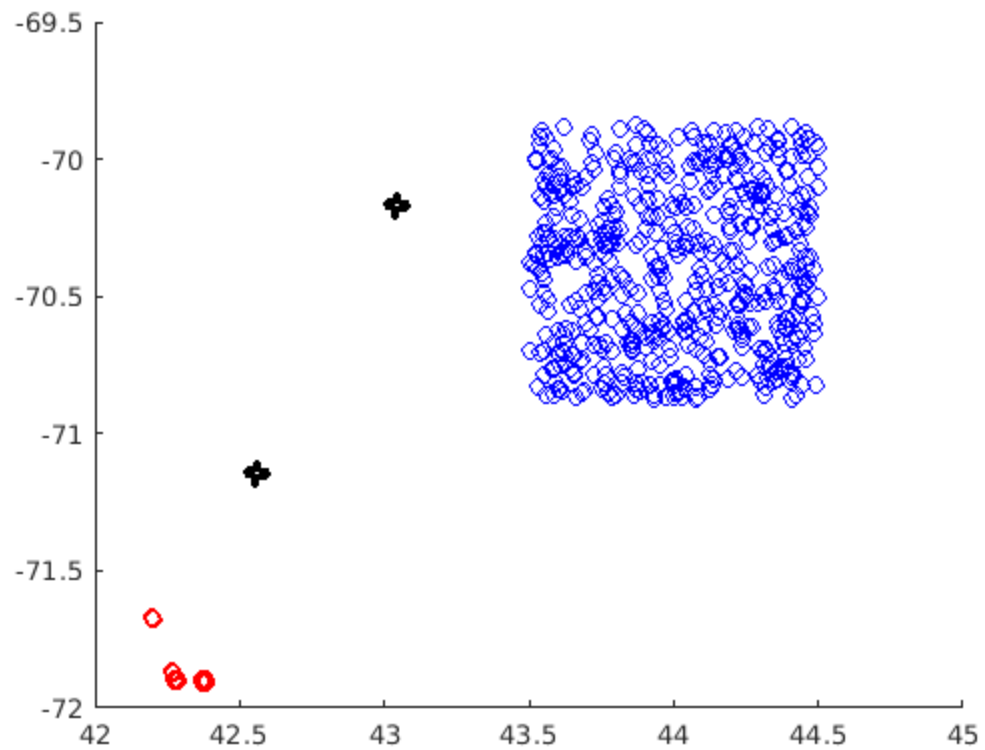
---

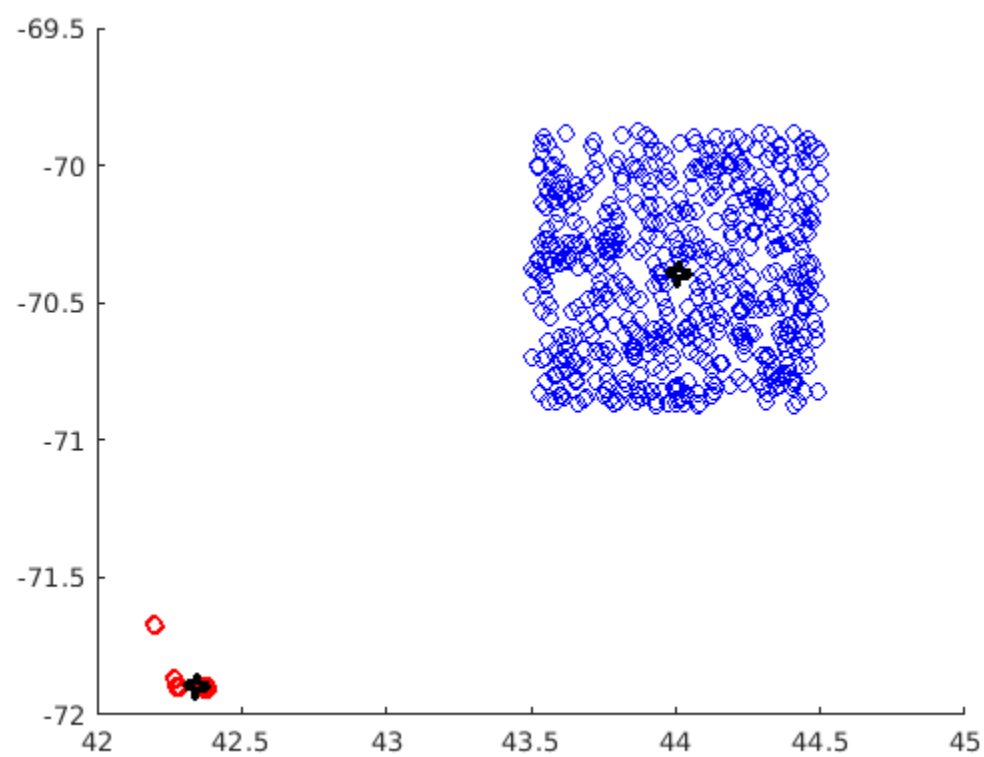
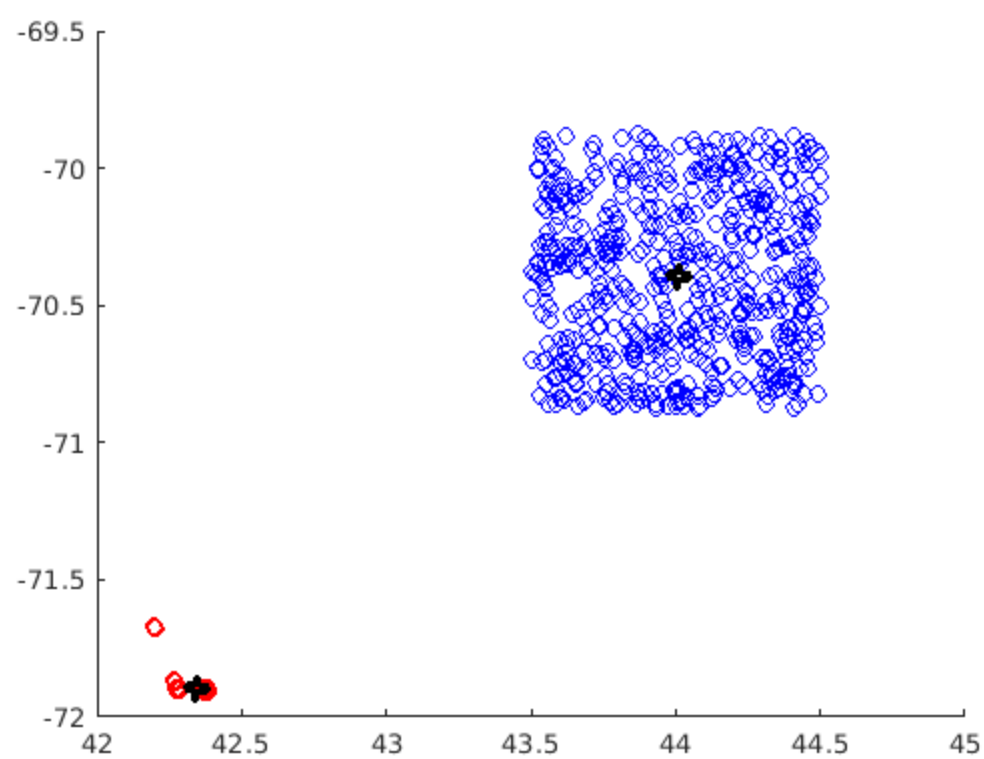
---

```
    centroids = [mean(cluster1_X(:,1)), mean(cluster1_X(:,2)),  
    mean(cluster2_X(:,1)), mean(cluster2_X(:,2))];  
end
```

```
K_optimal_by_elbow =
```

```
    2
```





---

*Published with MATLAB® R2017b*