BIOINF 580 - Assignment 4

Due on April 7, 2019 at 10pm

After implementing each step, copy the output, the resulting figure and the description of your results into this document. To copy a Matlab figure, click on *Copy Figure* in the edit menu of your figure. Make sure that your *x*-axis has the appropriate unit, all your axes are labeled and your plot has a title. **When you plot the data, pay attention to the values on the *x*- and *y*-axes. If you stretch or squeeze one of the axes, you can artificially make the clusters/classes look wider or narrower. You should be aware of that fact when you investigate the shape of your clusters/classes.** Submit this document as part of your assignment along with your code(s).

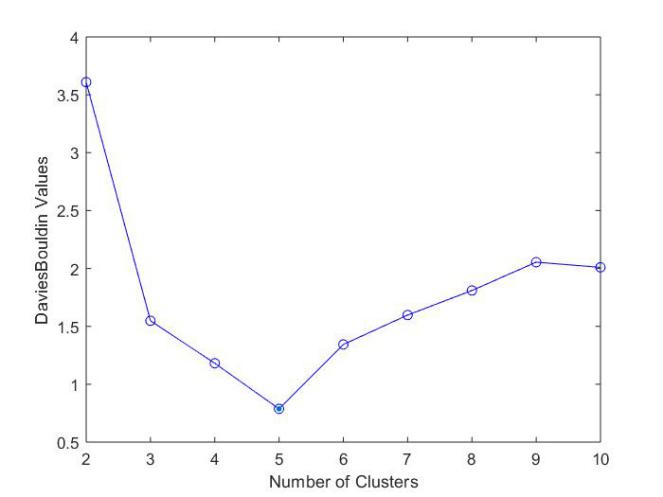
1. Download *data1.mat* from the assignment folder and open it in Matlab.
   1. Use a scatter plot to display the data. Comment on the patterns that you see in the data. How many clusters can you identify? (5 points)

***5 clusters***

* 1. What distance metric do you think is most appropriate to use for clustering this data? Use this metric in the rest of this question. (5 points)  
     *Hint: notice the scales of the x- and y-axes. Use xlim() and ylim() functions with the same input values to set the x- and y-axes to have the same scales. This would help better identify the structure of the data.*

***mahalanobis distance***

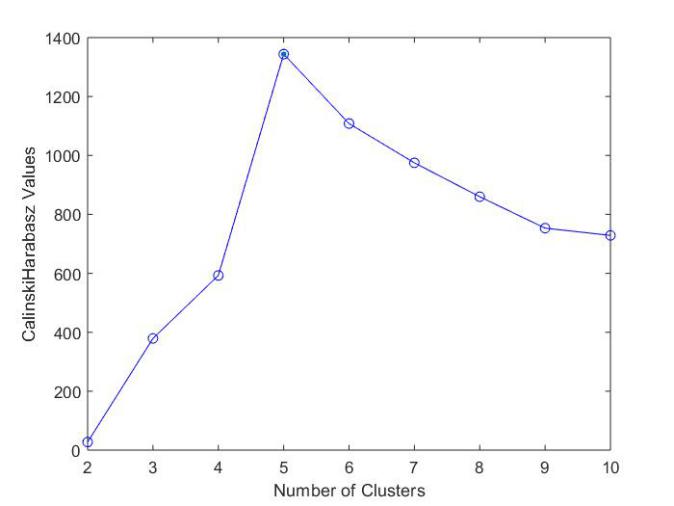
* 1. The k-means algorithm can lead to different results when different initial values are used. One way to resolve this issue is to run the clustering algorithm several times and use a criterion to select the best results. Revise the *k\_means()* function available in assignment 4 package to do this. Add a new function input called ‘*replications*’ that determines how many times the clustering algorithm should run. The new function signature should be *k\_means(X, k, distance, replications)*. Add a *for* loop inside the function to run the k-means algorithm *replications* times. Compute the sum of the distances between the data points and the center of the cluster that they are assigned to (can be calculated using *sum(min\_distance)*. The vector *min\_distance* is calculated in line 20 of the *k\_means* function) and return the clustering results that have the lowest sum as output. (20 points)
  2. Next we use evaluation metrics for the clusters to determine the best value for *k* (number of clusters). For each *k* ranging from 2 to 10, use the *evalclusters()* function in Matlab with Davies-Bouldin criterion to evaluate the compactness of the clusters (look at the Matlab help entry for the *evalclusters* function to call *k\_means()* function using *function handle*). Run the k-means algorithm 10 times by setting replications = 10. Plot the results with *k* on the *x-*axis and the criterion values on the *y-*axis. Comment on the results. How many clusters do you think there are in the data? (10 points)

**

***There are 5 clusteres in total as when cluster number =5, the DaviesBouldin Values***

***is the minimum.***

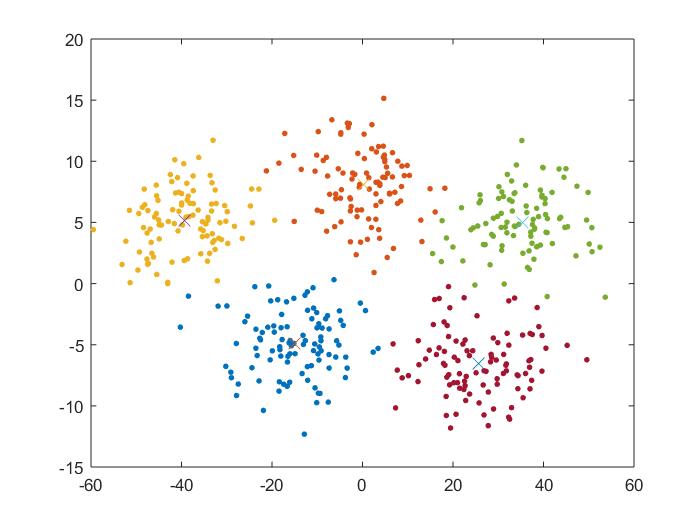
* 1. Repeat part *d* with Calinski-Harabasz criteria to create similar plots and comment on the results. Notice that the best clustering results are achieved when Calinski-Harabasz reaches its maximum value while the best clustering results are achieved when the Davies-Bouldin criterion reaches its minimum value. (5 points)

**

***There are 5 clusteres in total as when cluster number =5, the Calinski-Harabasz***

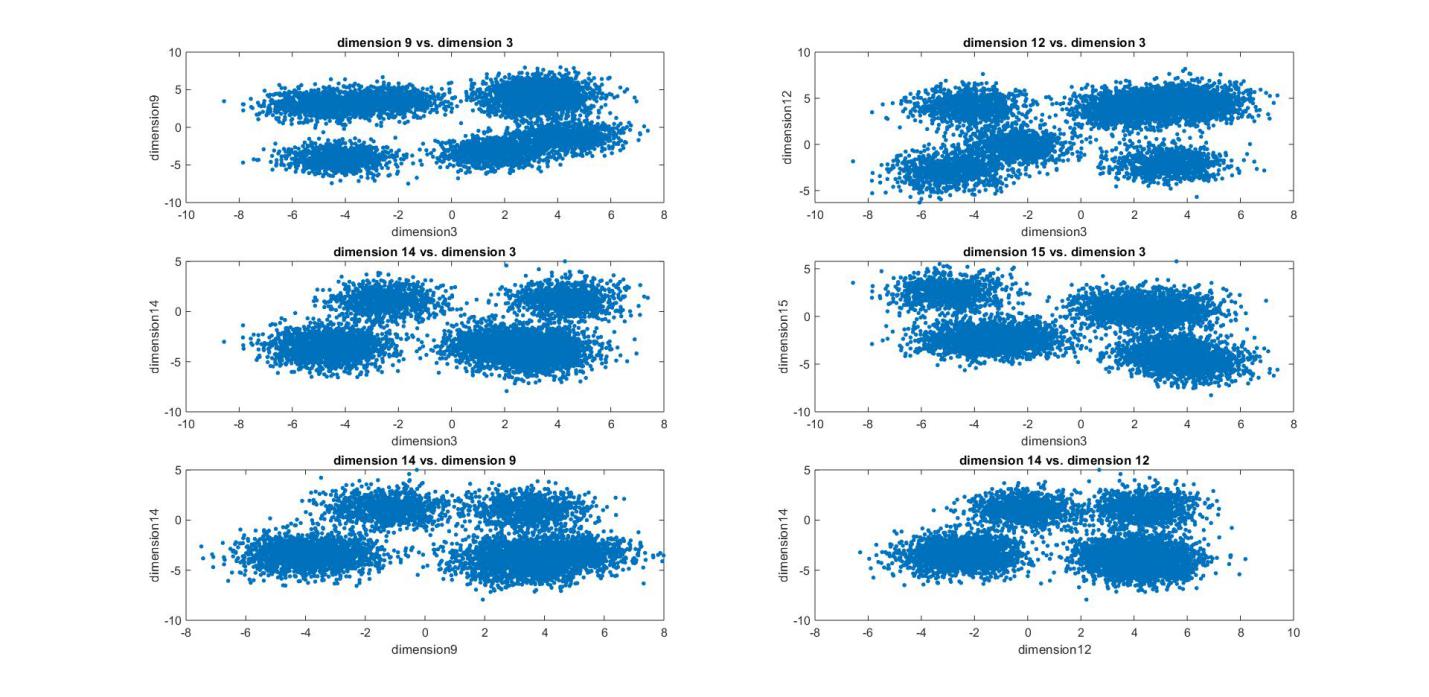
***values is the maximum.***

* 1. Use the best *k* from parts *d* and *e* to cluster the data and plot the results. Comment on the resulting clusters. Do you think the clusters correspond to the true structure of the data? (5 points)

**

***From the graph, I believe the clusters do represent the structure of the data.***

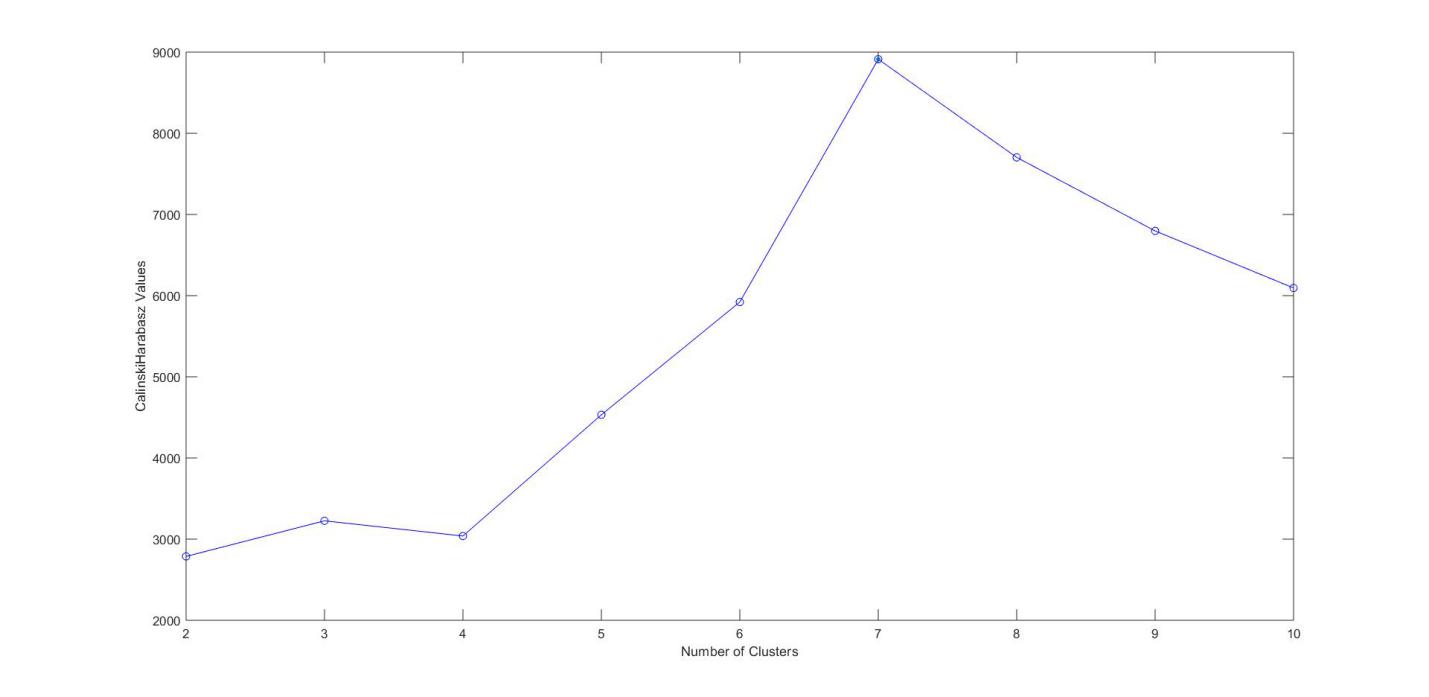
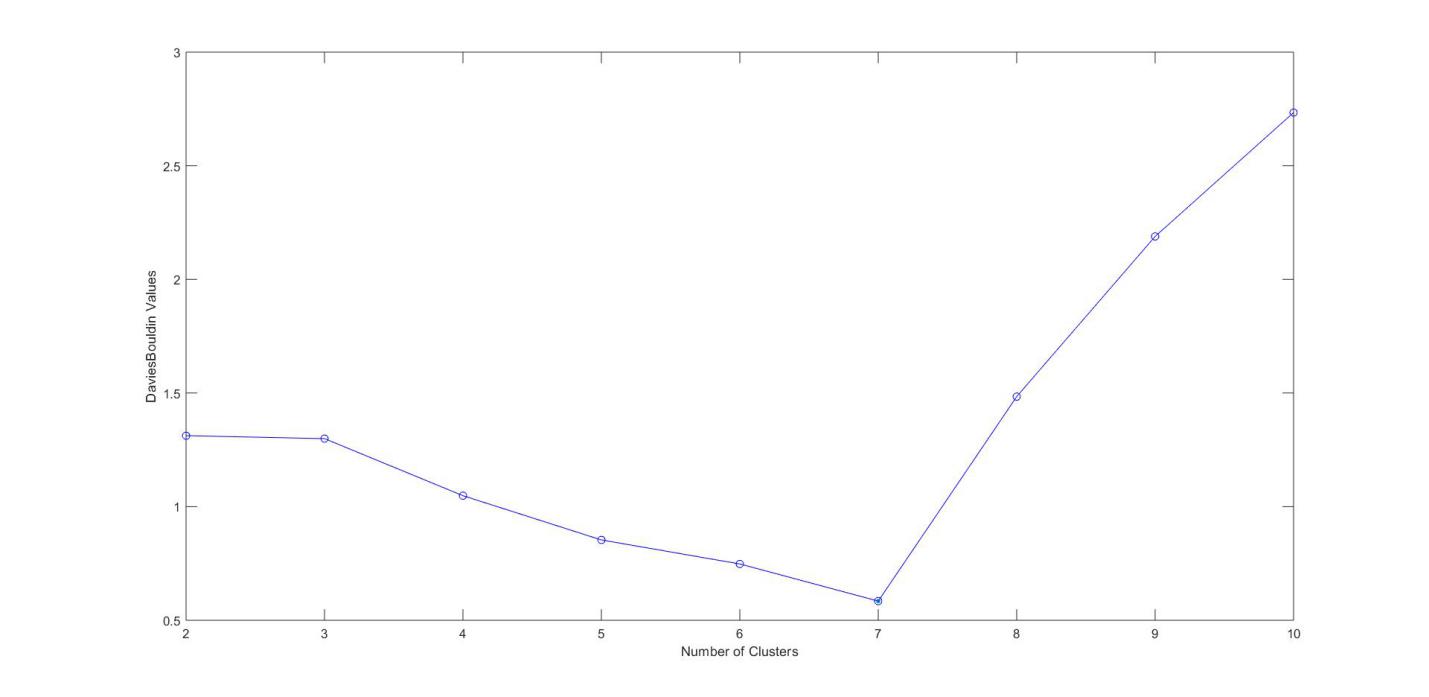
1. Download *data2.mat* from the assignment folder and open it in Matlab.
   1. Plot different combinations of pairs of variables in Matlab, e.g., plot variable 1 vs. 2, 1 vs. 3, …, 2 vs. 3, 2 vs. 4, …, etc. Choose a few of the plots that better illustrate the clusters with most amount of separation and include them in your submission. These are the projections of this 15-dimensional data into 2-dimensional spaces. Comment on the number of clusters that you can identify in the data. (5 points)



***3vs9 3vs12 3vs14 3vs15 9vs14 12vs 14***

***4 clusters***

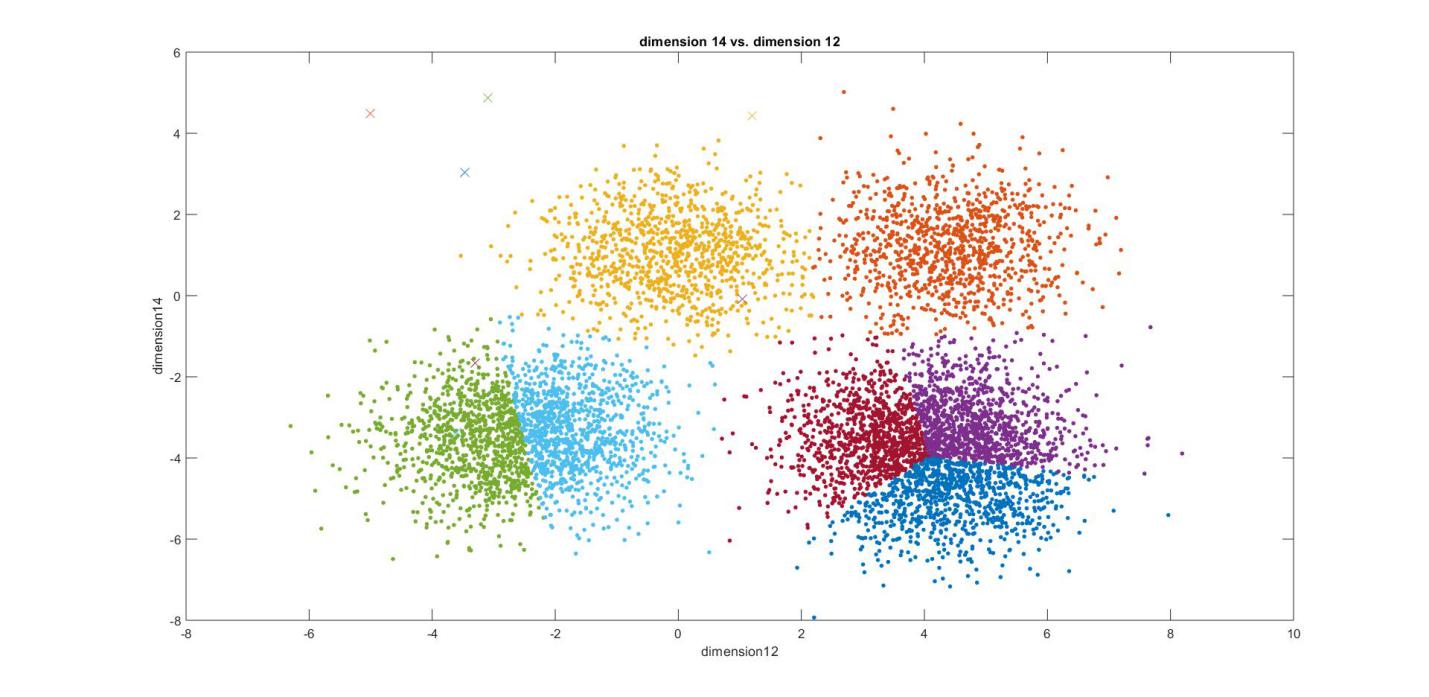
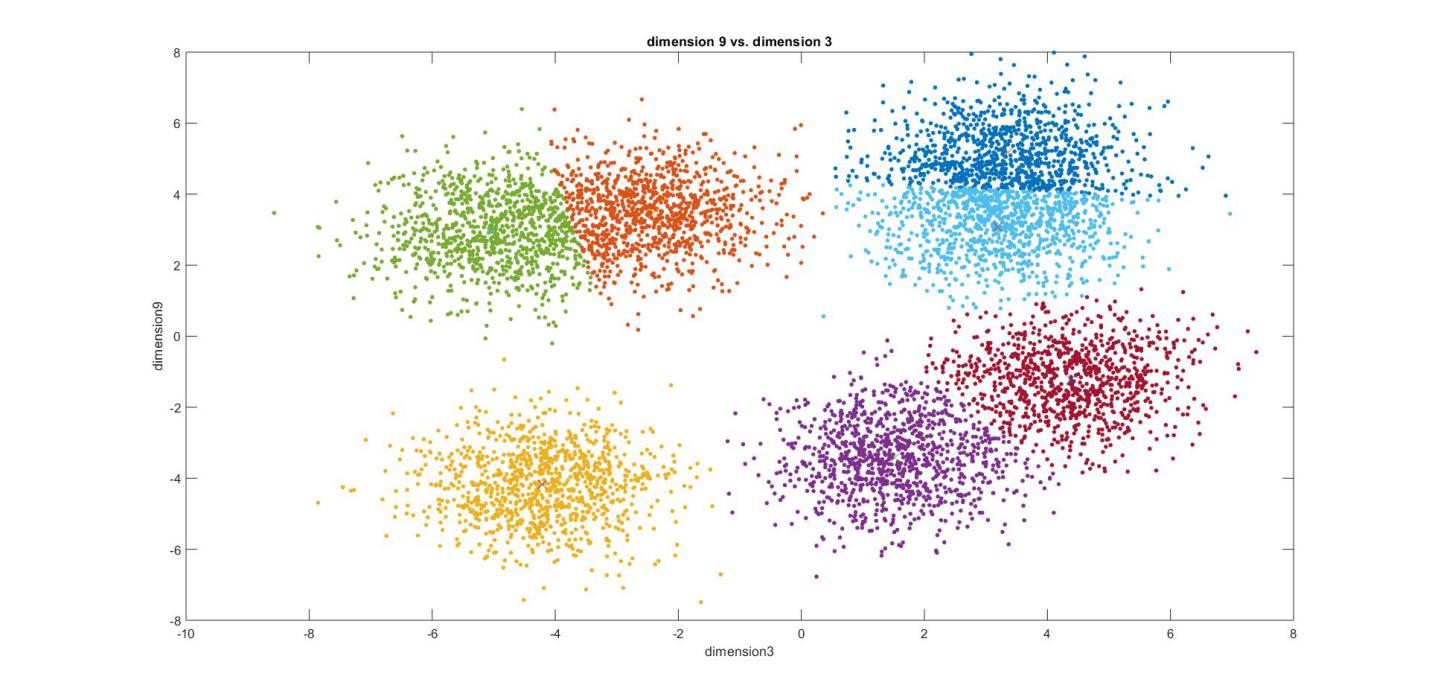
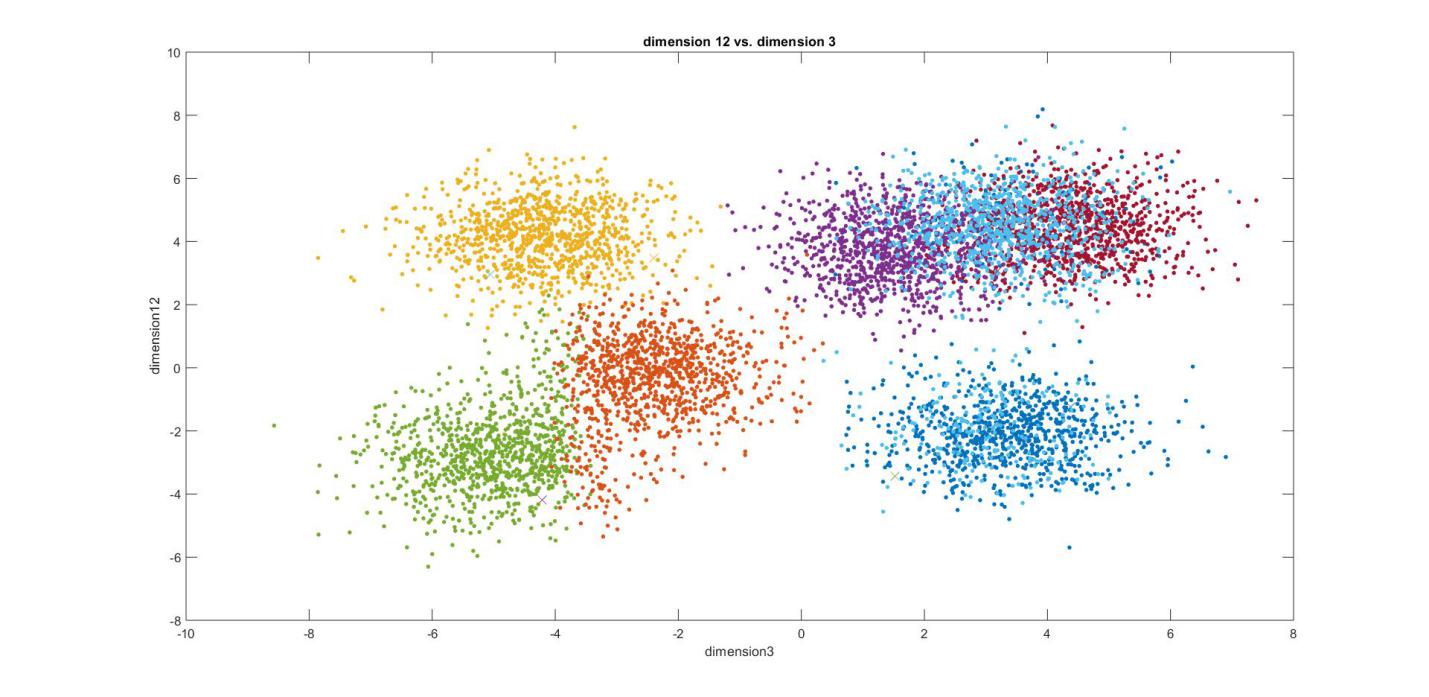
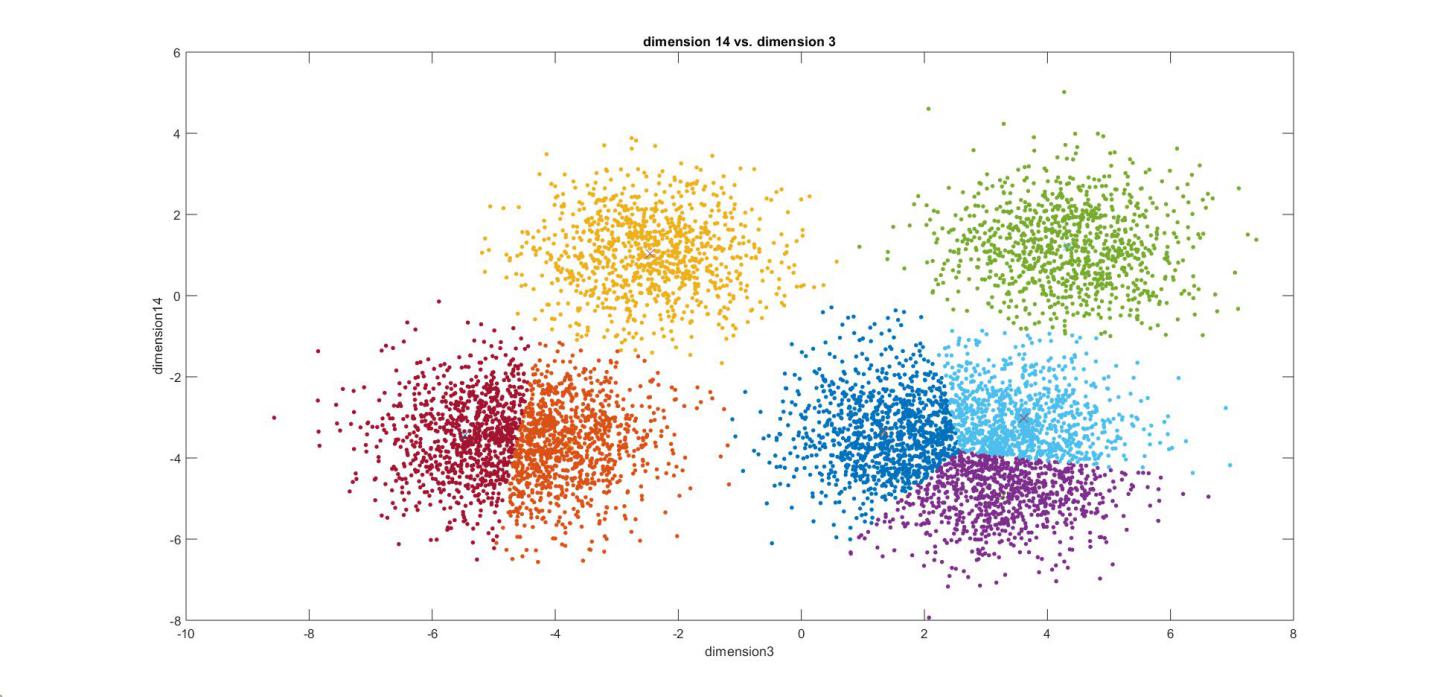
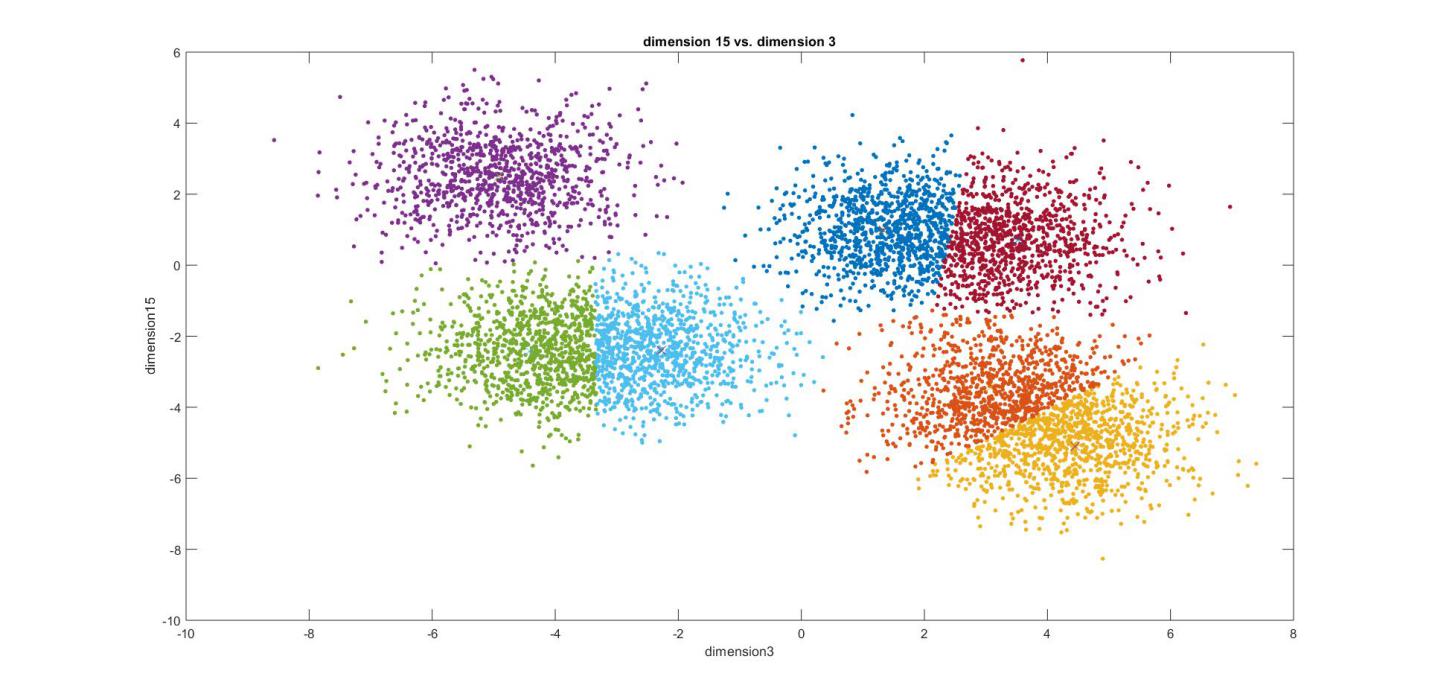
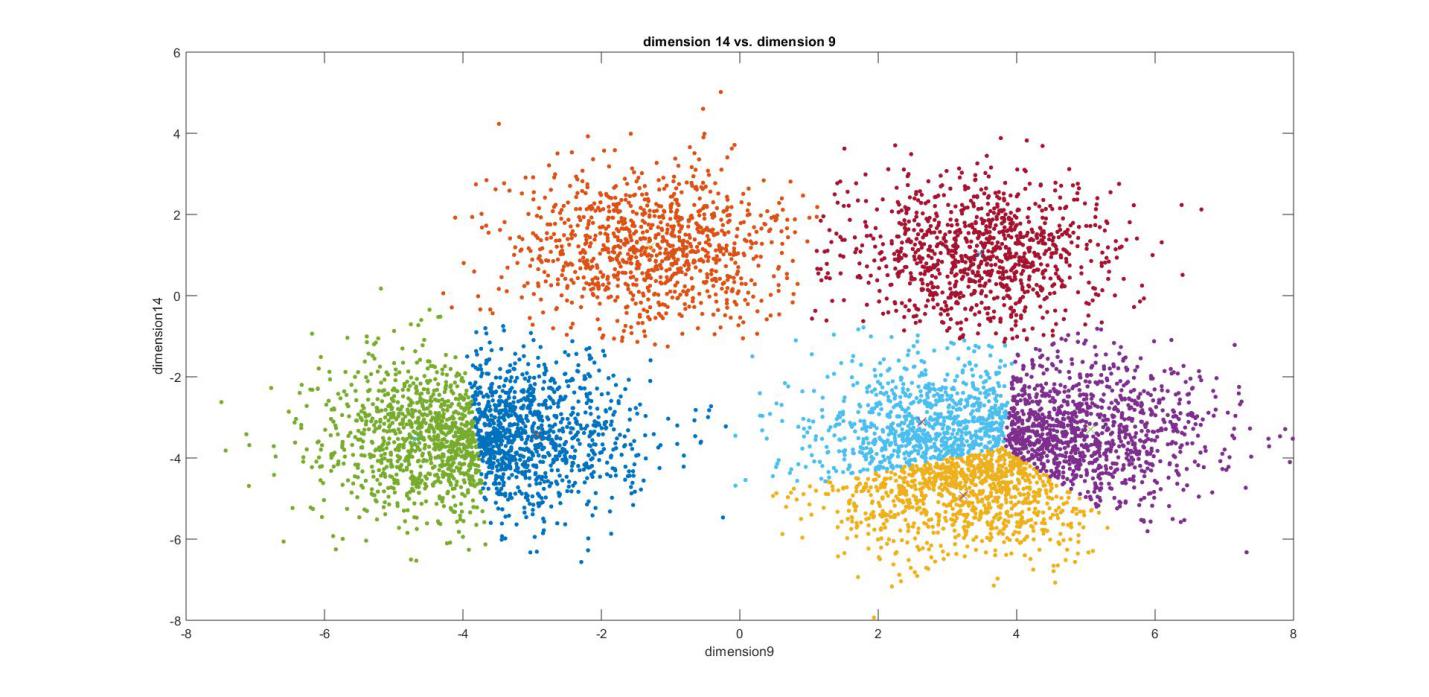
* 1. Use a similar approach as in question 1 to systematically determine the number of clusters in the data. This dataset is much larger than the one used in question 1. Therefore, our *k\_means()* function can take a long time to run since it is not implemented efficiently. Instead, use Matlab’s *kmeans()* function with Euclidean distance and 5 replications (look at the *k\_means* ‘Replicates’ option to specify the number of replications). Use the two metrics that we discussed in question 1 to create plots showing the cluster compactness for different values of *k* (similar to parts d and e in question 1). (5 points)



***Number of clusters is 7 as DaviesBouldin Values is minimum and Calinski-Harabasz***

***value is maximum when number of clusters is 7.***

* 1. Choose the best value for *k* according to the graphs that you created in part b and use it to cluster the data. Create the same plots as in part 2.a and use the clustering results to color the data points. Comment on the quality of the clusters. (5 points)



***The clustering did not fully represented the structure of the data from the plots due to the***

***reason that the best number of clustering centers is predicted under 15-dimensions while the***

***plot is clustering a 2D data into 7 clustering center. Need different visualization methods such***

***as T-SNE to better visualize the structure and shape of the data.***

Turn in the following files:

* *main\_function\_4.m and k\_means.m*
* This report named youruniquename\_hw4.dox