**Clustering Workshop** **CS 480**

*When you are finished, share your Python code with me. If you are using Google Colab, share your code with:* `lins.brian@gmail.com`

In this workshop, you will write some helper functions to apply k-means clustering to a set of vectors. You can assume that numpy has been imported as `np`.

1. Complete the following function.

```
def getClusters(xs, zs):
    # xs is a list of n numpy arrays.
    # zs is a list of k representative arrays.

    # Loop through each x in xs.
    # Use the functions np.argmin and np.linalg.norm to find
    # the index of z in zs that is closest to x.

    # Return a list of clusters.
    # Each cluster is a list of numpy arrays.
```

2. Complete the following function.

```
def clusterMeans(clusters):
    # Return a list with the mean of each cluster in clusters.
```

3. Complete the following function.

```
def kmeans(xs,zs,k):
    # Apply the k-means algorithm to xs with initial representatives zs
    # Return a list of clusters.
```

4. Once you have completed the three functions above, apply them to the example data sets on the course website. You'll need to convert each data set into a list of numpy arrays `xs` and randomly select $k$ of those arrays as the initial representative arrays `zs`. Here is the code to do that.

```
import random
xs = list(A)
zs = random.sample(xs,k)
```

5. Print an image of the clusters when $k = 3$ and when $k = 6$ for each example. Here is code to get an image of the clusters. This is also on the course website (so you can cut & paste).

```
import matplotlib.pyplot as plt
for cluster in clusters:
    clusterArray = np.array(cluster)
    plt.plot(clusterArray[:,0], clusterArray[:,1],'o')
plt.show()
```

6. Write a function to compute the value of the objective function:

$$\sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - z_i\|^2.$$

Choose one of the example data sets. Compute the value of the objective function when $k = 3$ and $k = 6$ for that example after you run the k-means algorithm.