Alex Te (861215867)
CS 171 Spring 2018
Professor Papalexakis
Late Days used for this assignment: 1
Total Late Days used so far: 2

**Question 0: Getting real data**

I imported the data from the Iris and wine database by creating a script in python that would read in the data, store it into an array, and parse it. Which is the same for Assignment 1.

**Question 1: K-Means Clustering**

To implement the K-Means Algorithm, first I prompted the user to specify the number of clusters, k. Then what the algorithm does is it chooses k random points in the data and uses the chosen points as centroids. The points that are not centroids are data to be classified to a centroid.

How I classified what datapoint is in what cluster is by taking the distance from the datapoint to each of the k neighbors. The shortest distance is the cluster that it belongs to.

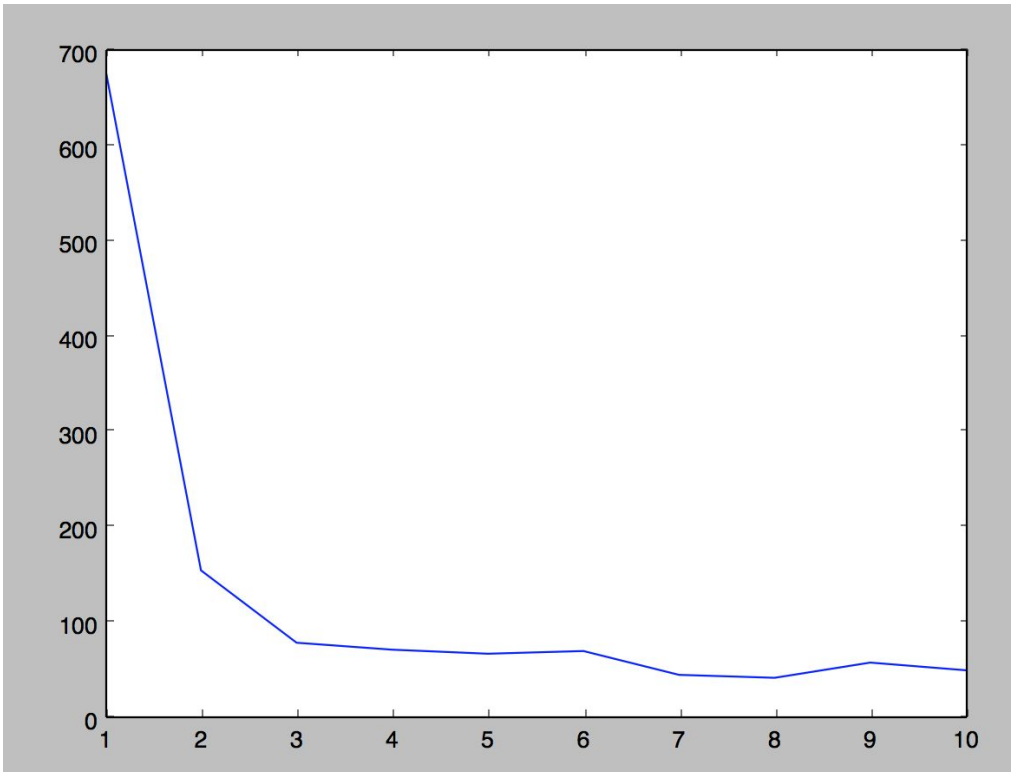
I repeat this algorithm until the cluster assignments stop, meaning that all the points are in the “correct” cluster. “Correct” because we are randomly initializing points as clusters, so each iteration will yield different results.

I put the algorithm in a function called “k_means” which does all the work. I also created a distance function called “distance” to calculate the Euclidean distance from each point to its centroid. The return values for k_means is a tuple of cluster_assignments, cluster_centroids, and distance

**Question 2: Evaluation:**

(next page)

Knee plot:

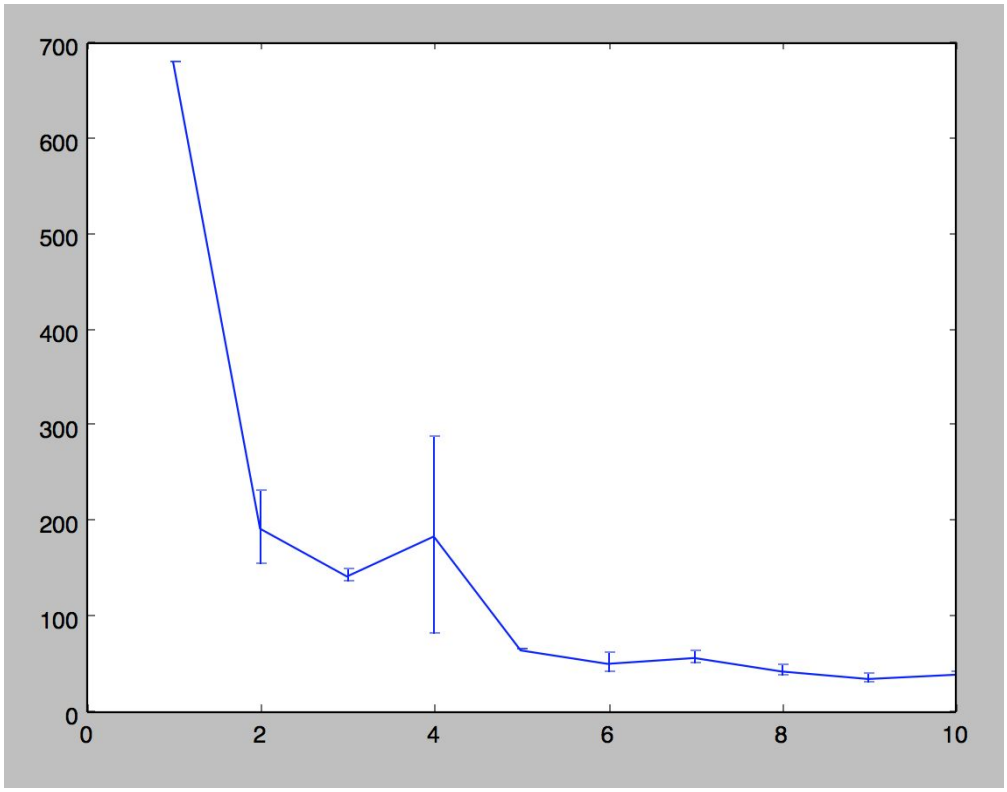



We use a knee plot to see how good our clustering algorithm is working. We look at where the errors stop improving dramatically and it is at k = 3. This is also where the “knee” appears and is what is expected because the original dataset has 3 classes. This shows that the clustering algorithm had a correct outcome.

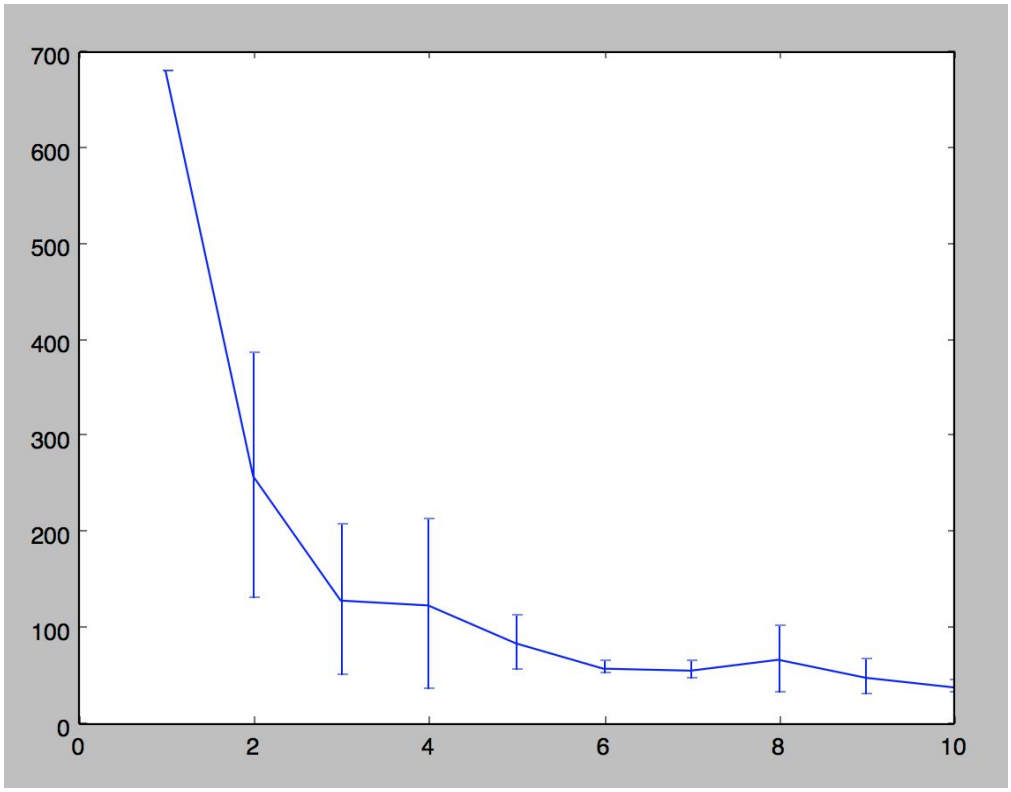
The x axis is the number for k ( 1 - 10 ) and the y-axis is the Sum of Squared Errors (SSE)

However, the knee plot is subject to changes based on the randomly initialized clusters, and if we have a bad random round, then the knee plot would look “dirtier”. Which is why for the next part, we graph the mean on the y_axis and standard deviation for different number of max_iterations, namely 2,10, and 100.

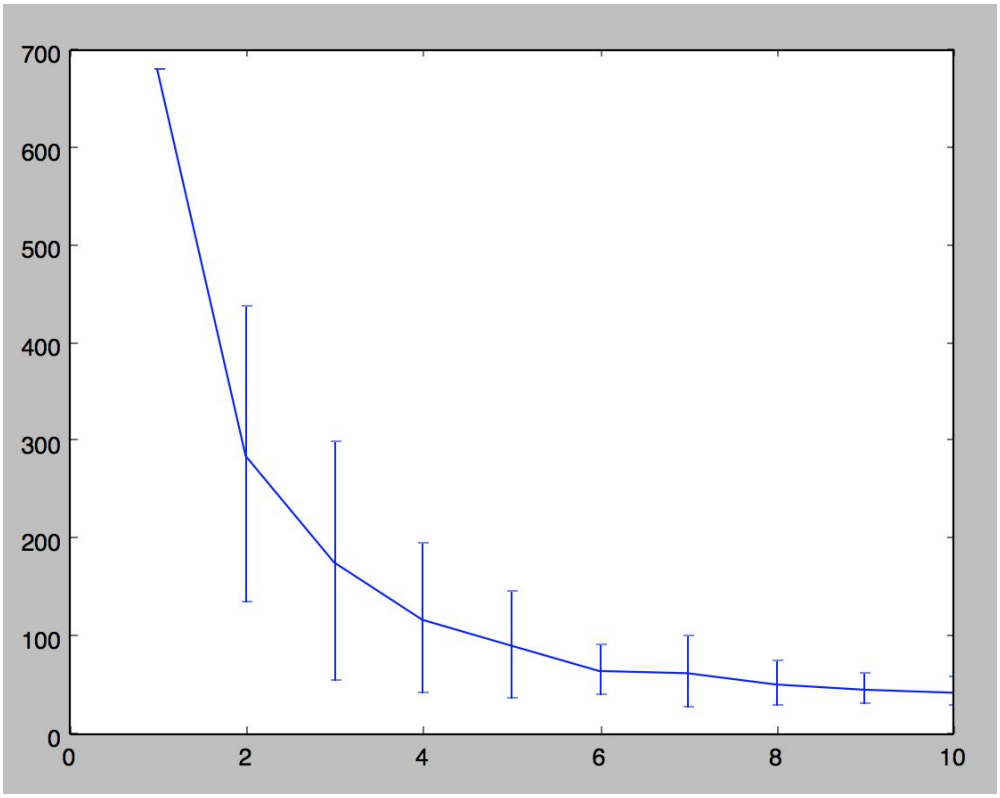
For max iterations = 2, I get:
(next page)




For max iterations = 10, I get:



For max iterations = 100, I get:




Each max iterations ran the K-means algorithm that many times and for every k value (1 - 10). This time, instead of just the knee plots, we have the error bar to indicate the spread of the data.