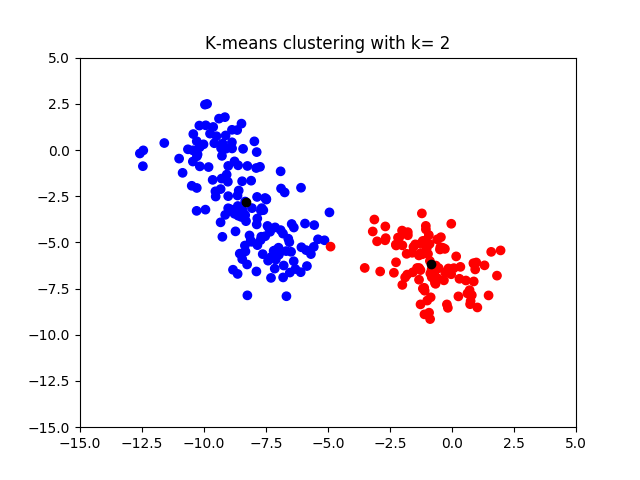
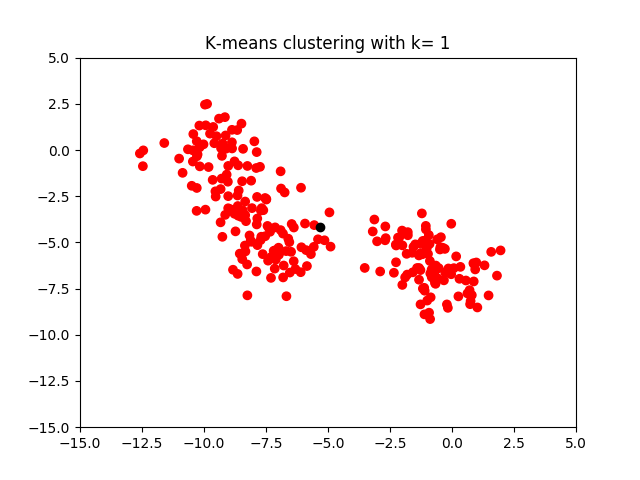
Milo Knowles

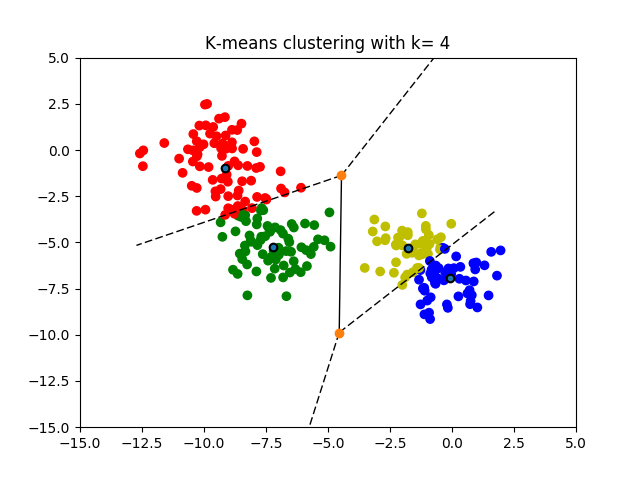
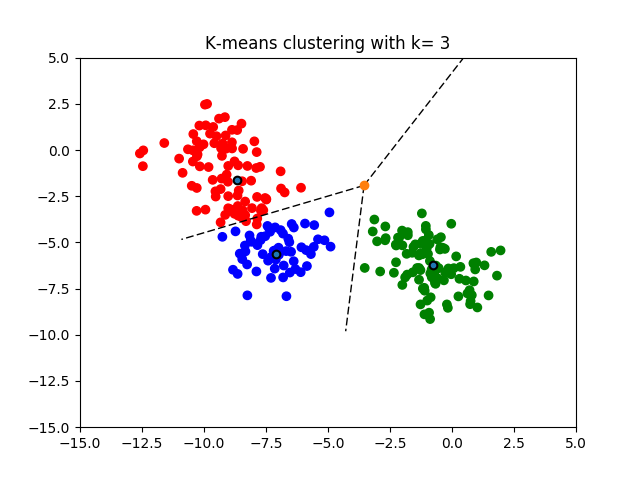
6.036 Project 3

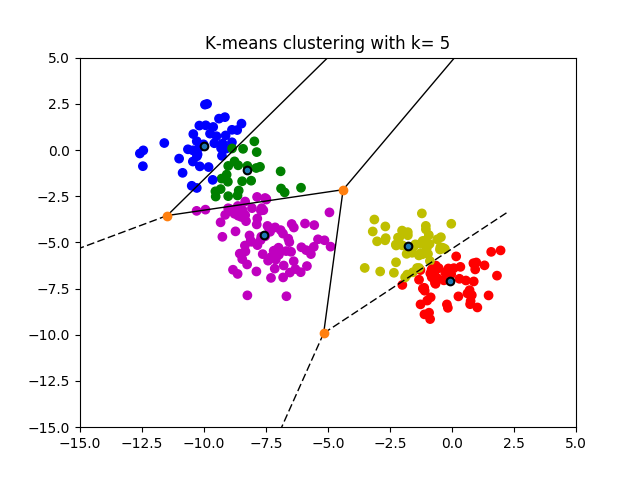
May 5, 2017

**Project 3**

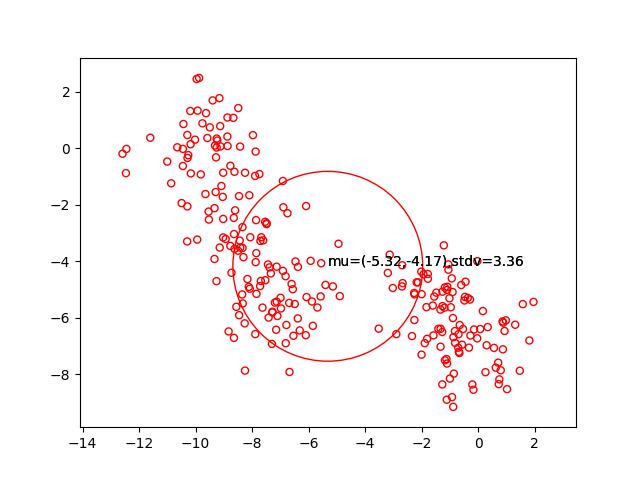
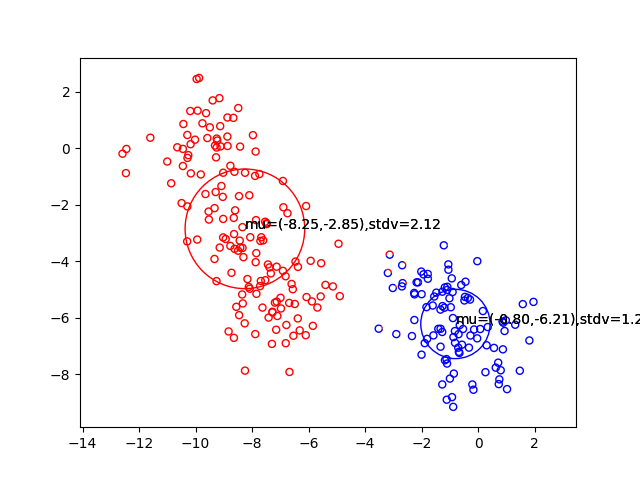
**1.1: K-Means Results (with k=1, 2, 3, 4, 5)**

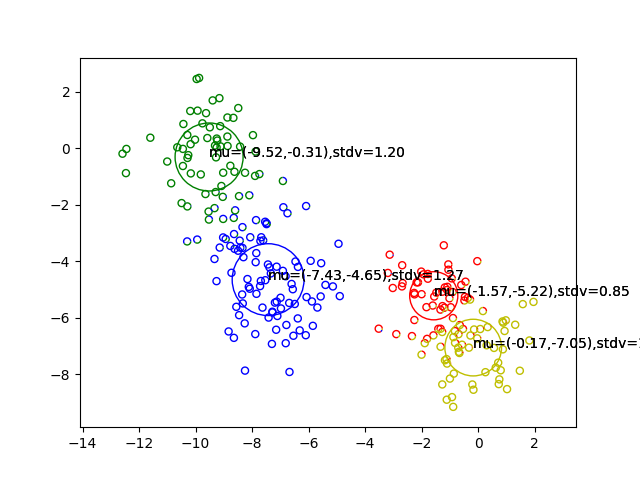
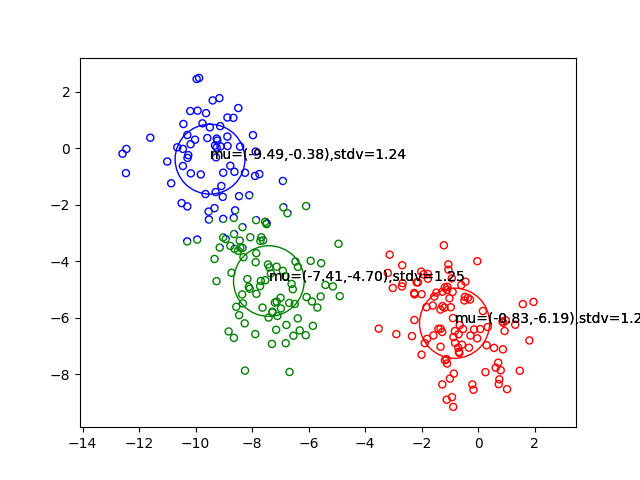


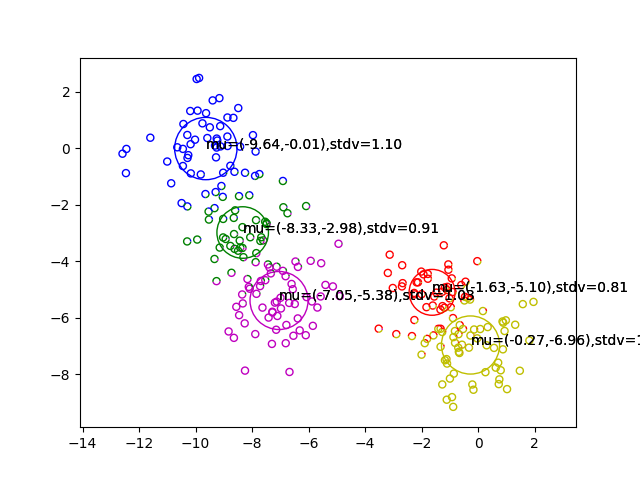




**1.4: EM Results (with k=1, 2, 3, 4, 5)**

****





**Log Likelihoods (for plots above)**

K=1: -1315.32

K=2: -1139.73

K=3: -1072.59

K=4: -1058.96

K=5: -1045.62

**1-5.** For the most part, K-Means and EM produced very similar results for each K=1, 2, 3, 4, 5. Up to an arbitrary coloring, both algorithms seem to cluster points similarly. The difference in coloring is due to the fact that K-Means and EM are non deterministic. Depending on the random initialization of the algorithms, clusters can end up with different labeling, even if they partition points in the same way.

**Part 2: Clustering Census Data**

**2-1.** Our independence assumption will produce overconfident cluster assignments because as more redundant features are added to the data, clusters will appear farther and farther apart. If two features are highly correlated, then points within a cluster are very likely to have similar values of those features. This means that by adding more correlated features, we are simply adding another dimension in which points from the same cluster are close, but points in different clusters are relatively far apart. The Euclidean distance between clusters will be higher in the larger feature space, causing the model to be overconfident in its cluster assignments.

**2-2.** **CMM: Posterior Distribution**

**2-3a.** **CMM:** **Maximum Likelihood Estimate for .**

**2-3b.** **Maximum Likelihood Estimate for .**

**2-5.** Over the iterations, log likelihood increases. This makes sense, because our model is finding the maximum likelihood estimates of the parameters. Because log is a monotonic function, as we increase the likelihood of our model, the log likelihood also increases. For example, with *k=2*, the model begins with a log-likelihood score of -4.656e6 but ends with a log-likelihood score of *-2.736e6* after *7* iterations. In addition, as the *k* value increases, and more clusters are used, the log likelihood improves more, since the model is better able to fit to the data.

**2-6b.**

**Best K Value (LL):**

**Best K Value (BIC):**

These results do not agree because…

**7a.** Describe the clusters found by model

**7b.** Using a smaller value of K, the clusters are …

**7c.** If the model was trained on census data for each state, we would end up with some *K* clusters, each represented a demographic. Each state would likely have some number of it’s citizens in each of the clusters (or perhaps would not be represented in some of the clusters). By examining the number of people from each state that are represented in each of the clusters, we could put states into a *K* dimensional “cluster space”, where each dimension of the space represents a cluster. States with a high representation in some cluster would have a large value in the dimension corresponding to that cluster. Once we represent each state as a vector in the *K* dimensional cluster space, we can compare the “closeness” of two states based on their cosine similarity. This is analogous to the way that we compared documents using cosine similarity. If two states are close in the cluster space, this means that they are represented similarly among demographics; this means that the states have similar populations.