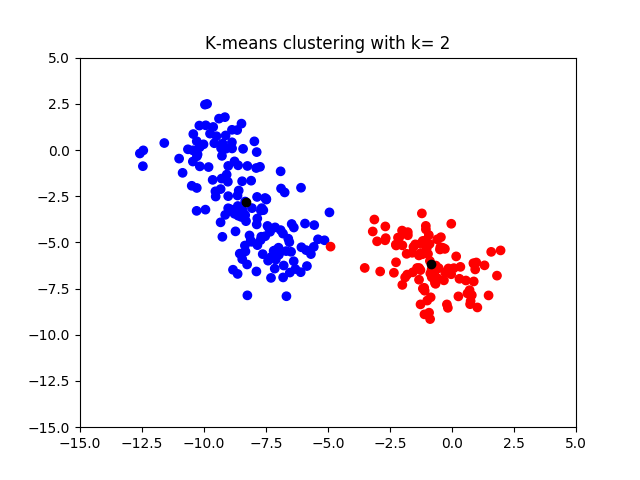
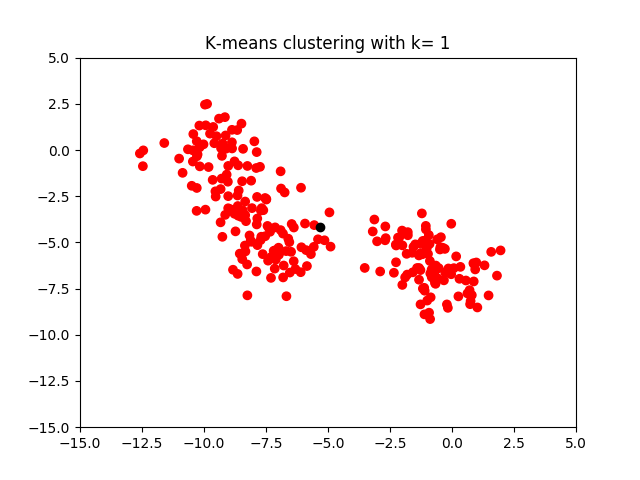
Milo Knowles

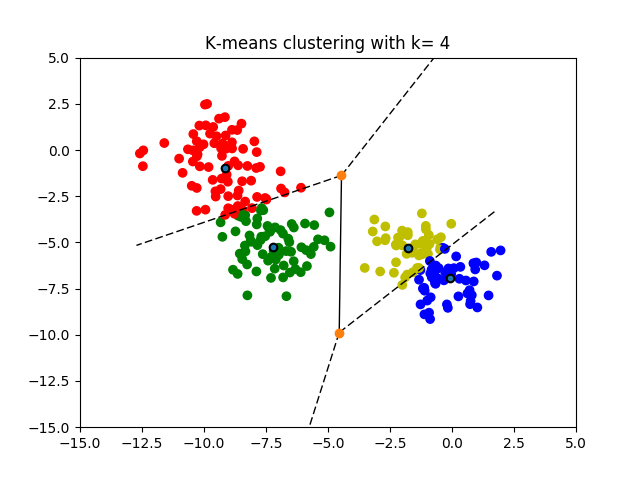
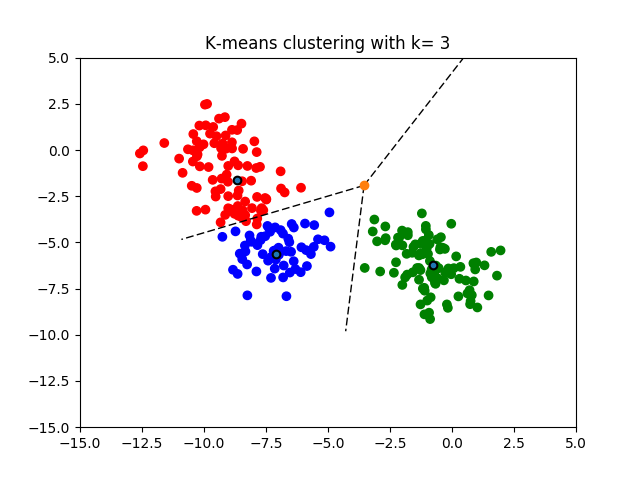
6.036

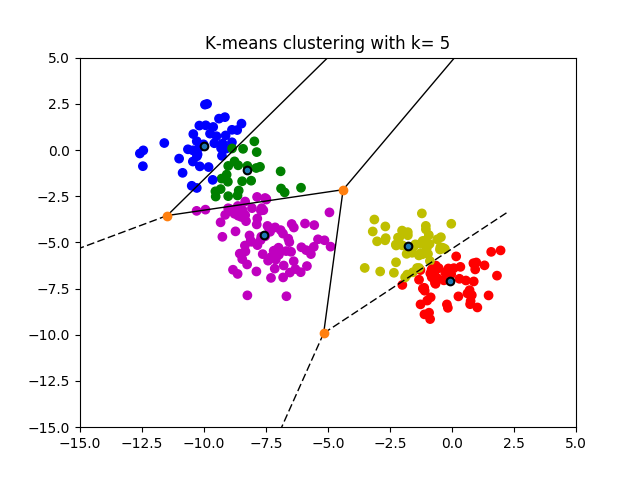
Project 3

Project 3 Writeup

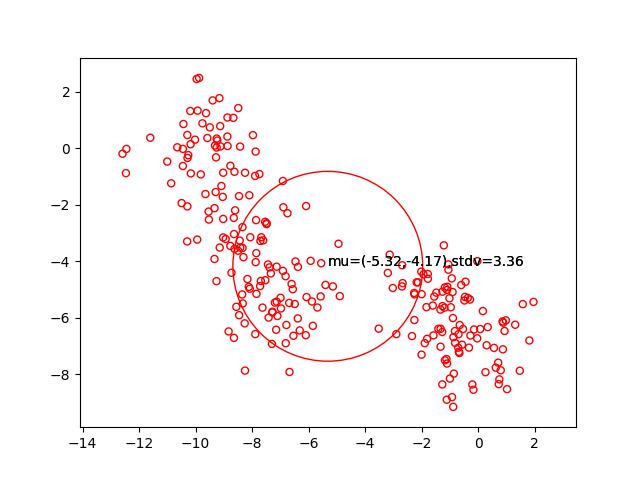
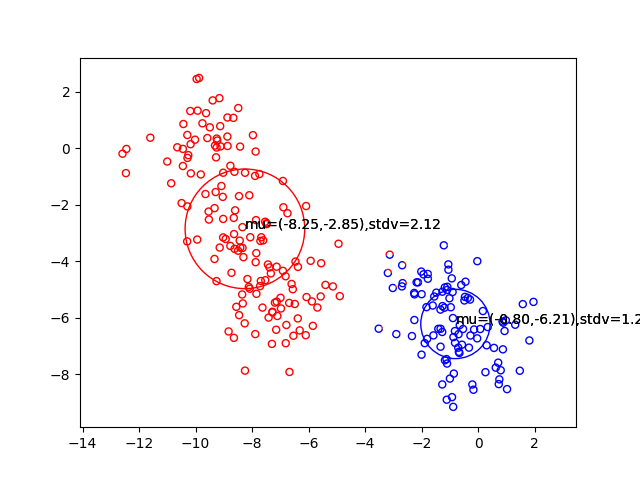
**Part 1: K-Means Results (with k=1, 2, 3, 4, 5)**

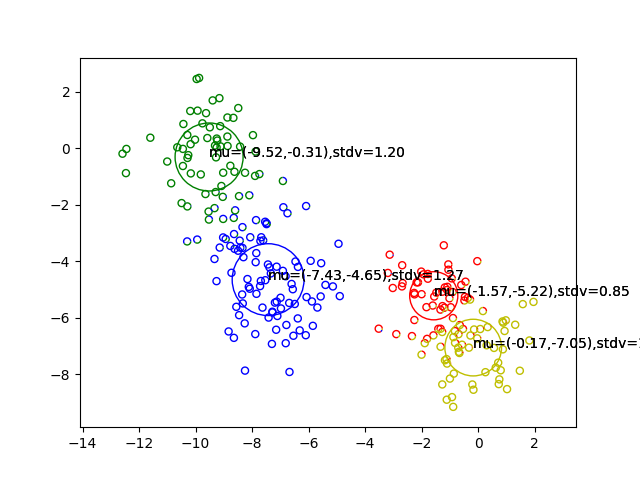
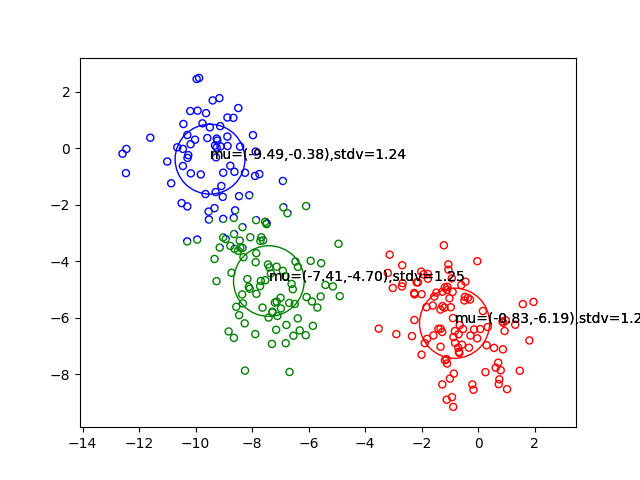


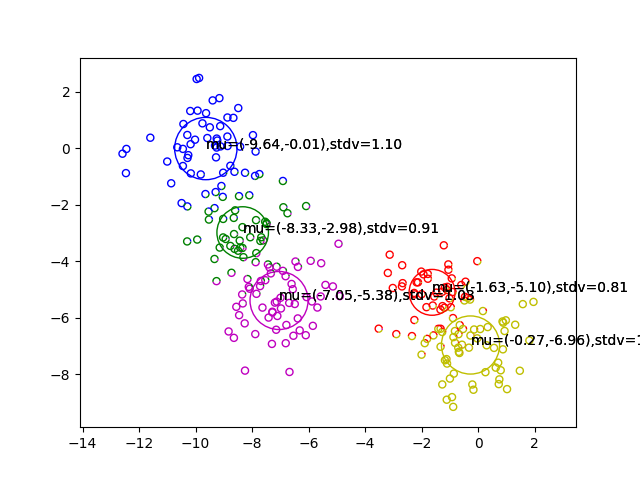




**Part 1: EM Results (with k=1, 2, 3, 4, 5)**

****





**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 seems to cluster points similarly. # TODO

**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. **Posterior Distribution**

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

2-3. b. **Maximum Likelihood Estimate for . (Using definition of above).**

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.