Don Liu

The dataset was read. Each line was split and placed into an array. The first 7 elements of the array were used to instantiate wheat objects, all of which were placed inside of an array list. Picking a centroid randomly was a matter of using the next integer method of the random class, with parameter being 220 because a total of 220 wheat objects were read. This produced three random integers, which were used to get random elements from the array list of wheat objects. In step six, instead of using the random class, a data set was manually chosen. My selection process was choosing the datasets that weren’t too similar in data, nor were too obscene that it’d be considered outliers.

Aside from the test driver, wheat, k means, and wheat clusters class was created. The wheat class is used to objectify wheat, and it takes in its seven respective attributes. This class also handles arithmetic between wheat classes, with methods like division and addition. These operations are later used to find distance, ev, iv, etc. It also features a print and equal class, which are mostly used for testing. The wheat clusters class is a representation of clusters of wheat. It contains an array of three clusters as its field, along with its convergence state, and three centroids. Three wheat objects are needed to instantiate the wheat clusters class, so that they could be used as centroids. The closest cluster method iterates through the entire data set, comparing each data point’s distance to each cluster. The number of the cluster that is closest is returned. The reset centroid method recalculates centroids after datasets are distributed to their respective closest cluster. It also resets the clusters, so that the next iteration of dataset can reassign to blank clusters. This method returns true is reset occurred, and false otherwise. This is used to determine if convergence occurred. The k means class is where the k means algorithm actually takes place. The kmeansAlg method iterates through the dataset, and puts the data points into the closest cluster. It then determines if convergence happened. If it does, end loop, otherwise continue.

I found that the best results came from centroids that were relatively distant from one another. Also, sometimes I notice similar results for my calculations of iv and ev, despite centroids differing; I wonder if it’s a coincidence?