Homework 5: Political k-Means

Your job is to use k-means to discover natural clusters within the House of Representatives data, that we used earlier. Your program should be called `kmeans.py`.

The program will take two command-line arguments. The first will be the name of the file containing the data. The second will be the number of groups to split it into. (If this number is less than 2, just exit the program with a short error message.)

To place all of the Representatives in feature space, assume that each "yea" has a value of 1, each "nay" has a value of -1, and each abstention has a score of 0. The dimensionality is the same as the number of issues (so 10, in the given file). Do not normalize the data in any of the dimensions.

To determine the initial centroids, perform the following procedure:

1. Start by choosing the two Representatives that are farthest from each other in feature space, and give the centroids the values of those two. (If there is a tie, choose the one with the lowest-number Representative. If there is still a tie, choose the lowest-number second Representative.)
2. Each subsequent centroid should coincide with the Representative who has the largest sum of squared Euclidean distances from all previously existing centroids. (Again, if there is a tie, choose the Rep with the lowest number.)

Proceed with k-means until the system stabilizes, using Euclidean distance to create each new centroid. Centroids are allowed to take non-integer values. If ever you have a tie, place the Representative in the lower-numbered group. At the end, you must print out the following:

- The initial centroids that you chose. You should display the names of the Representatives on whom you based the centroids.
- The number of steps taken. Count both the E and M stages as a single step. Don't count the last step, in which nothing changes.
- For each final group, print its size, and the percent composition Democrat and Republican. (For readability, do one per line and indent the lines.)

Here is what your output might look like:

```
Initial centroids based on:  Rep-37, Rep-282, Rep-152, Rep-280, Rep-36
Converged after 5 rounds of k-means.
    Group 1:  size 40 (100.000% D, 0.000% R)
    Group 2:  size 77 (72.727% D, 27.273% R)
    Group 3:  size 116 (8.621% D, 91.379% R)
    Group 4:  size 91 (100.000% D, 0.000% R)
    Group 5:  size 106 (2.830% D, 97.170% R)
```

Hint: instead of calculating the distances between Representatives, or distances between Representatives and centroids, calculate only the squared distances. This “surrogate calculation” corresponds perfectly with calculating distances: the centroid with the least distance *must* also be the surrogate with the least squared distance. It will also save you computation time (square roots are expensive), and mean that you won’t be dealing with any irrational numbers (which might round strangely).