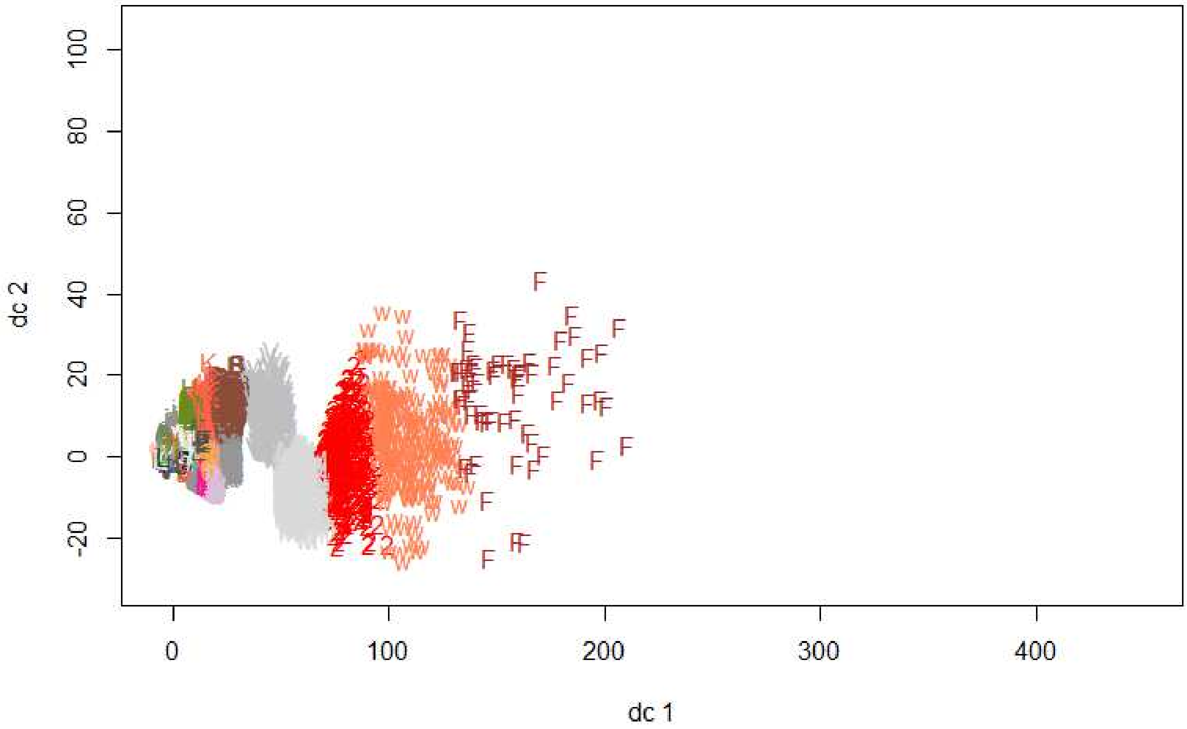
David Smith
STAT 4630
Homework 4

We aim to group users together based on the genres of movies that they like, in order to help understand their favorite kinds of movies and to offer better recommendations. Thus, k-means and hierarchal clustering will be the ideal algorithms to try. We choose k-means clustering over hierarchal clustering as viewing the dendrogram would be very difficult due to the size and complexity of the dataset, and we would not know where to make the cut. Furthermore, we have a rough idea of how many clusters to use in k-means, something large enough to sufficiently separate the large number of users but not so large as to have too many clusters, so hierarchal clustering is not necessary. Since we have 20 different genres, we will use PCA to represent the data by its first couple of principal components, making it less noisy.

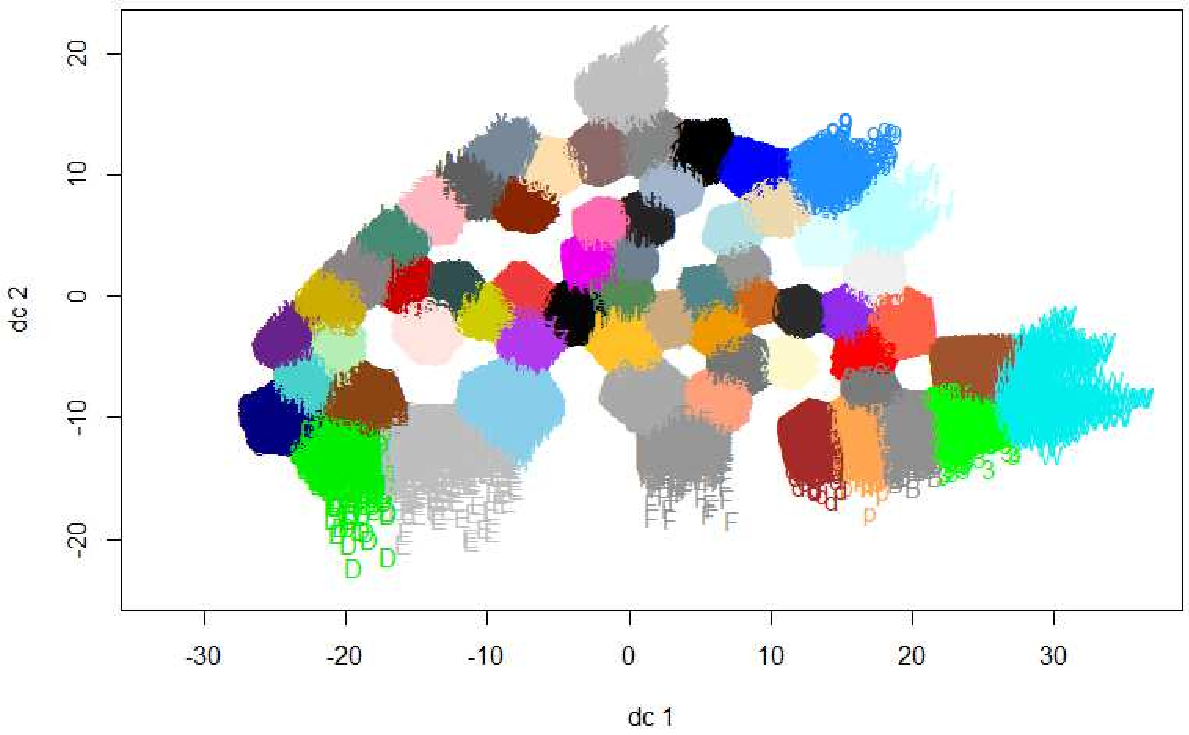
Both the number of movies a user watches that fall into a certain genre and the rating given to movies in those genres seem to be important information for classifying users by genre. Thus, we will consider several functions of these two variables. We will first simply count the number of movies each user watches that fall into each genre, and thus get a data point for each genre that a user watches, using a value of 0 if the user has not watched any movies in that genre. Then, we can apply k-means clustering on the complete collection of genre data for the users, with each user being an observation and the total number of genres being the dimension of the data. We will use 100 clusters, which will give an average cluster size of about 2650 users, given that there are around 265,000 users total. We will then repeat the same process using the average rating a user gives to each genre for our genre data value, and then use the average rating multiplied by the number of movies in that genre watched, which we suspect will perform the best, as it encapsulates both important pieces of information in a simple way, rather than just one. We will compare the three algorithms based on their total within-cluster sum of squares, which we aim to minimize. We will standardize the data before applying each k-means algorithm, as the actual values we will work with are not important in and of themselves. This way, the within-cluster error comparisons will be valid as well.

Having done all this for the number of movies that each user watches within each genre, and plotting the clusters on the first two principal components, we see that there are many small clusters that are close together, with clusters in the positive direction on the first principal component being larger and more distinguishable, but too spread out near the right:



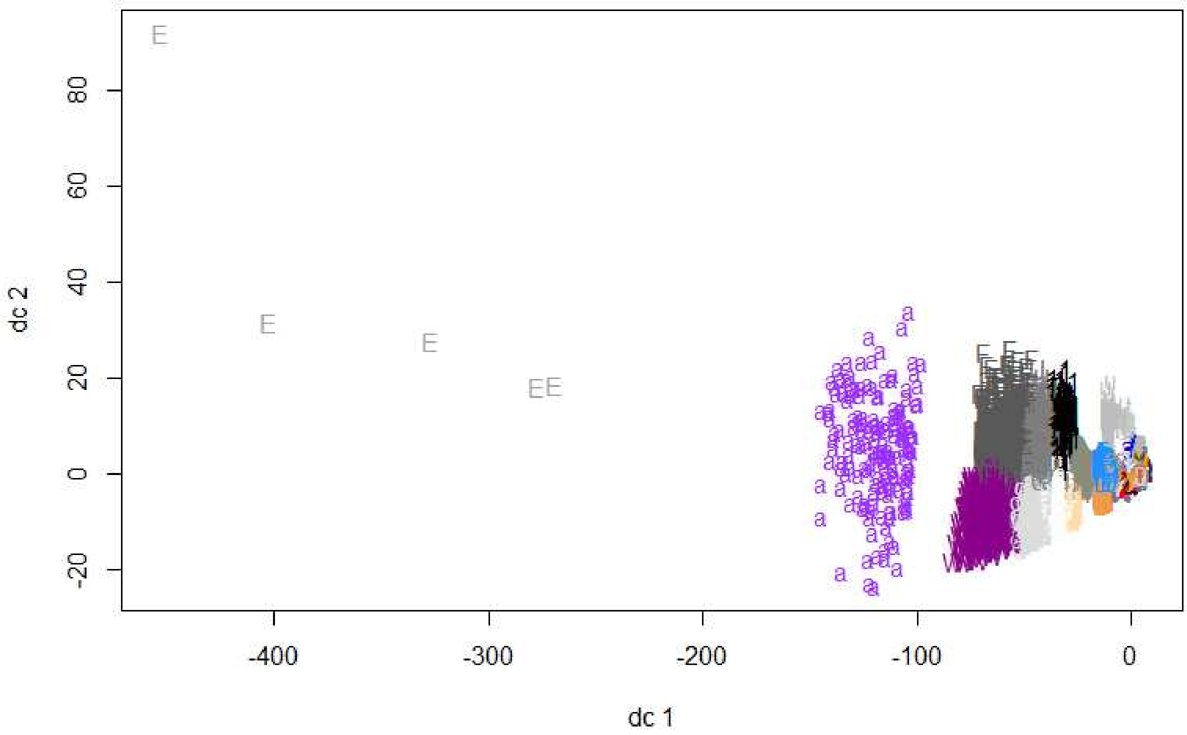
Using a smaller number of clusters was attempted, 50 and 75, but in this case, they were even more densely packed, and virtually indistinguishable. Looking at user 1, they were placed into cluster 22, and watched the most movies in genre ID 18, with a few other genres here and there. There are 2226 users in cluster 22, not too small of a number. The total within-cluster sum of squares is 89,735.53, which seems high.

Next, we used R to repeat k-means for the average rating users gave to movies in each genre. Plotting the clusters on the first two principal components, we obtained the following:



The clusters now appear to be more evenly sized, and more distinguishable from each other. There is also much more activity on the second principal component. It is suspected that the total within-cluster sum of squares will be smaller. Calculating this, we see that this is indeed the case, as it is 21,785.88, a significantly smaller error. Thus, using the average rating of movies to represent a user's preference for certain genres seems preferable to using the number of movies watched. Looking at user 1 again, they gave high average ratings to genres 18, 35, 80, and 99, a low average rating to genre 9648, and medium average ratings to a couple of other genres. They were placed into cluster 13, which has 2788 users, close to the expected cluster size.

Finally, we apply k-means to the product of the number of movies watched in each genre and the average rating given to movies in that genre. Plotting the clusters on the first two principal components, we get:



Somewhat surprisingly, this actually looks much worse than using only the average rating. The clusters for values of the first principal component are indistinguishable and densely packed, many being very small and/or dense, and the two clusters on the left, particularly cluster "E", are spread out too much. The within-cluster sum of squares is 89,575.69, much worse than using the average rating alone, confirming that this is a poor clustering to use. Looking at the first user again, they had high genre calculations for genres 18 and 35, and low calculations for genres 28, 80, 99, and 9648. They were placed into cluster 53, which contains 2458 users, not too small of a size.

As is clear, the best clustering to use is to calculate the average rating given to movies of each genre, and then use k-means on that data. This should be helpful towards the company's initiative to build a new recommendation system based on the genres of movies users like. Using an average rating is simple and easily interpretable, and avoids introducing noise that may arise from using a more complex formula.