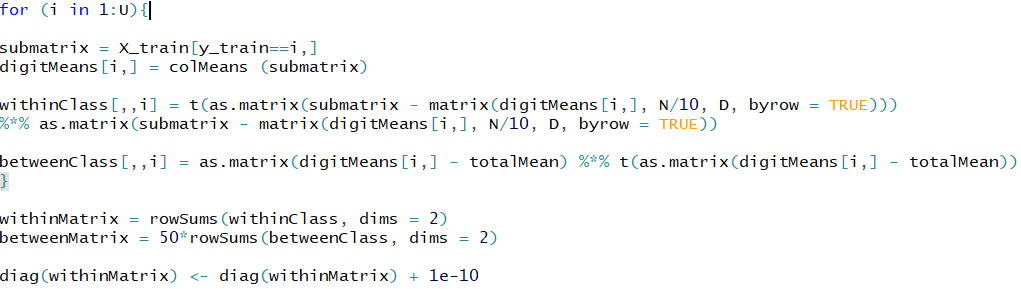
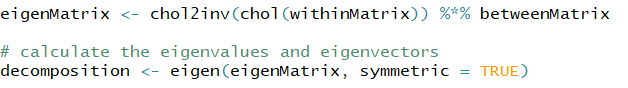
**INDR421 HW#6, Prepared by Arda Kırkağaç, 49799**

In this homework, our assignment was to apply Linear Discriminant Analysis on MNIST dataset, which are basically pixelwise intensity values of handwritten digits. Original data points are 784-dimensional, and we want to reduce the dimensions while protecting the information as much as possible.

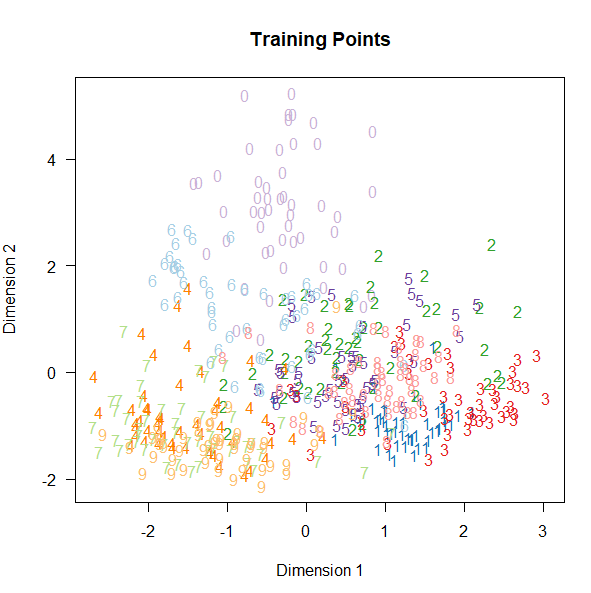
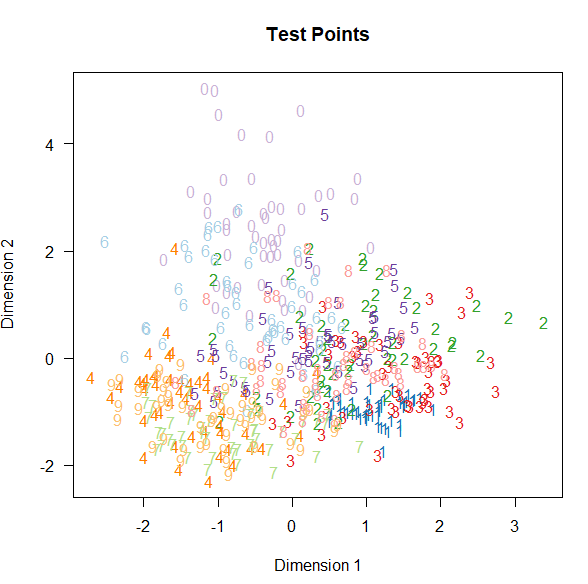
For each class, I extracted the submatrices to calculate within class and between class matrices. This extraction simply holds for the binary value, which is 1 if the data point belongs to that class, and 0 if it is not. To calculate within-class scatter matrix, the distances to the mean is calculated and then multiplied with its transpose. The matrices for each class are then added together to reach within-class scatter matrix. 1e-10 is added to avoid singularity since we will use inverse operation on it.

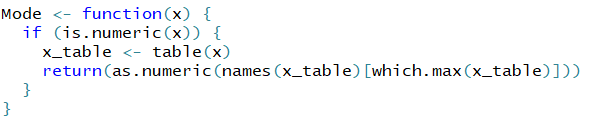
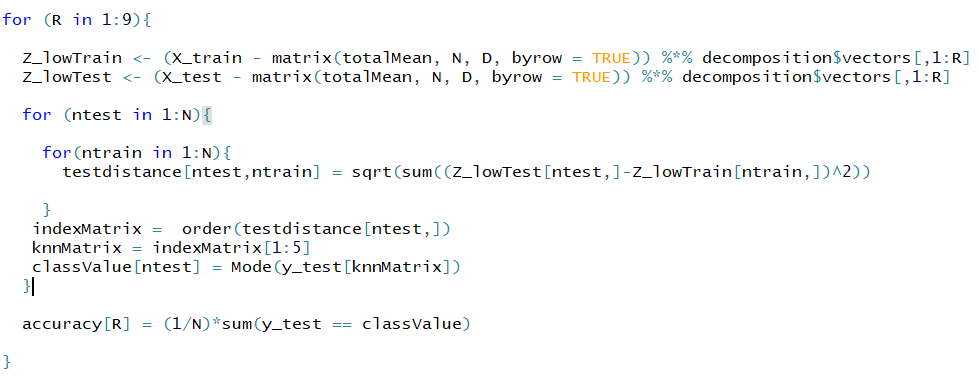
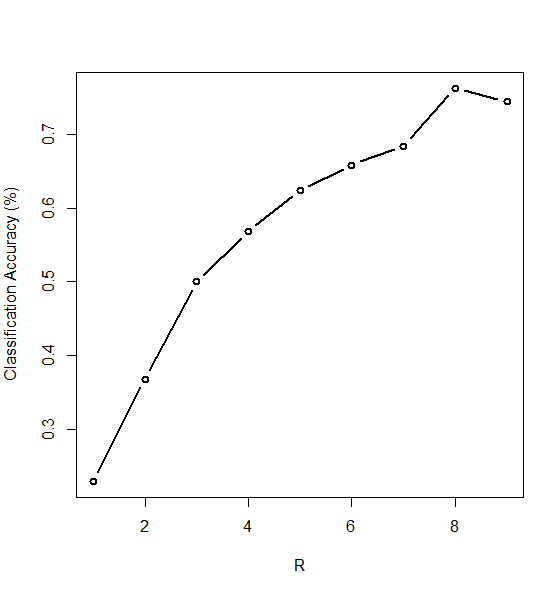
To calculate between-class scatter matrix, the total mean is calculated for each pixel, and subtracted from the means of each class to calculate submatrices, then added together. My code is below:

The vectors that solve this problem can simply be found using:



When R=2, first two vectors with largest eigenvalues will be used for projection. Here are my plots when training and test data are projected onto those vectors in 2-dimension:



For the next step, R value will be changed between 1 and 9, to calculate different low-dimensional projections and to check their classification accuracy. 5-nearest-neighbour algorithm is simply implemented by calculating the distance between a test point and every training points. This is done for each test point. The mode of the classes of the 5-nearest training data points will be our prediction. Then, the prediction is compared with the actual classes of test data points. Since R does not have a built-in mode function, one can define it in several ways. You can see my code for mode function, my code for classification, and my accuracy plot below: