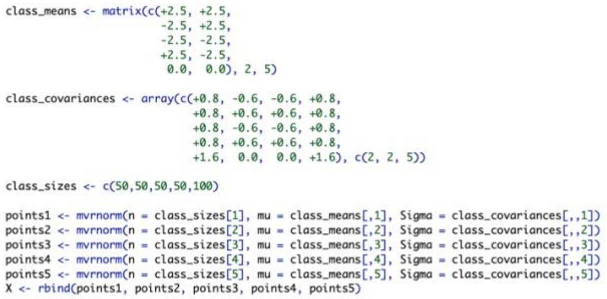
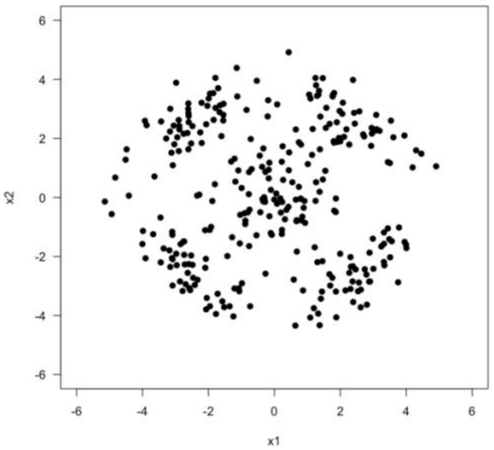
ENGR421: Introduction to Machine Learning
Fall 2020 – Homework 6
Ege Yelken – 61742

The aim of this assignment was to implement 2 clustering algorithms, namely k-means algorithm and Expectation-Maximization algorithm. I have followed the steps below:

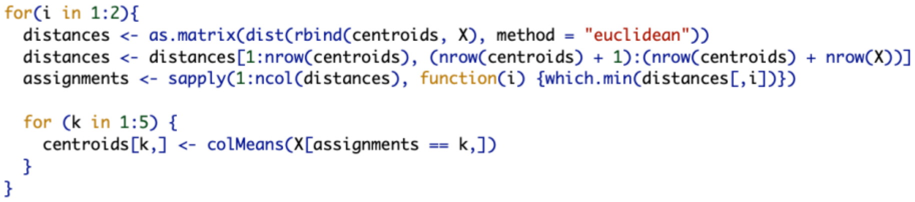
1. Generated random samples with given parameters and plotted them:



```
class_means <- matrix(c(+2.5, +2.5,
                        -2.5, +2.5,
                        -2.5, -2.5,
                        +2.5, -2.5,
                         0.0,  0.0), 2, 5)

class_covariances <- array(c(+0.8, -0.6, -0.6, +0.8,
                             +0.8, +0.6, +0.6, +0.8,
                             +0.8, -0.6, -0.6, +0.8,
                             +0.8, +0.6, +0.6, +0.8,
                             +1.6,  0.0,  0.0, +1.6), c(2, 2, 5))

class_sizes <- c(50,50,50,50,100)

points1 <- mvrnorm(n = class_sizes[1], mu = class_means[,1], Sigma = class_covariances[,,1])
points2 <- mvrnorm(n = class_sizes[2], mu = class_means[,2], Sigma = class_covariances[,,2])
points3 <- mvrnorm(n = class_sizes[3], mu = class_means[,3], Sigma = class_covariances[,,3])
points4 <- mvrnorm(n = class_sizes[4], mu = class_means[,4], Sigma = class_covariances[,,4])
points5 <- mvrnorm(n = class_sizes[5], mu = class_means[,5], Sigma = class_covariances[,,5])
X <- rbind(points1, points2, points3, points4, points5)
```

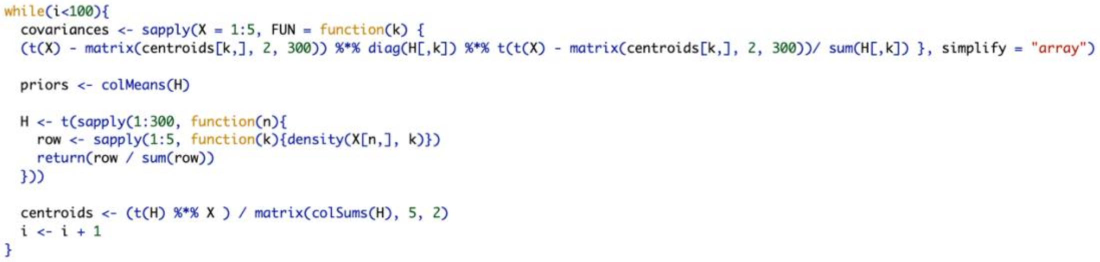


2. Ran the k-means algorithm twice with k = 5:



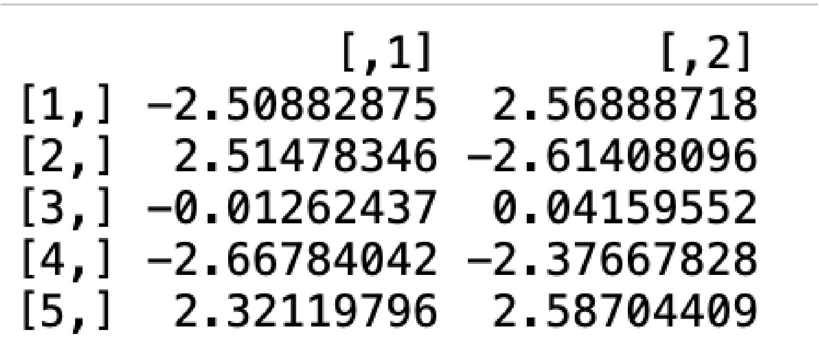
```
for(i in 1:2){
  distances <- as.matrix(dist(rbind(centroids, X), method = "euclidean"))
  distances <- distances[1:nrow(centroids), (nrow(centroids) + 1):(nrow(centroids) + nrow(X))]
  assignments <- sapply(1:ncol(distances), function(i) {which.min(distances[,i])})

  for (k in 1:5) {
    centroids[k,] <- colMeans(X[assignments == k,])
  }
}
```

3. I calculated the most likelihood (Gaussian) density of the hidden variable of the EM algorithm and updated the centroids as the initial mean values, then iterated over the algorithm 100 times:



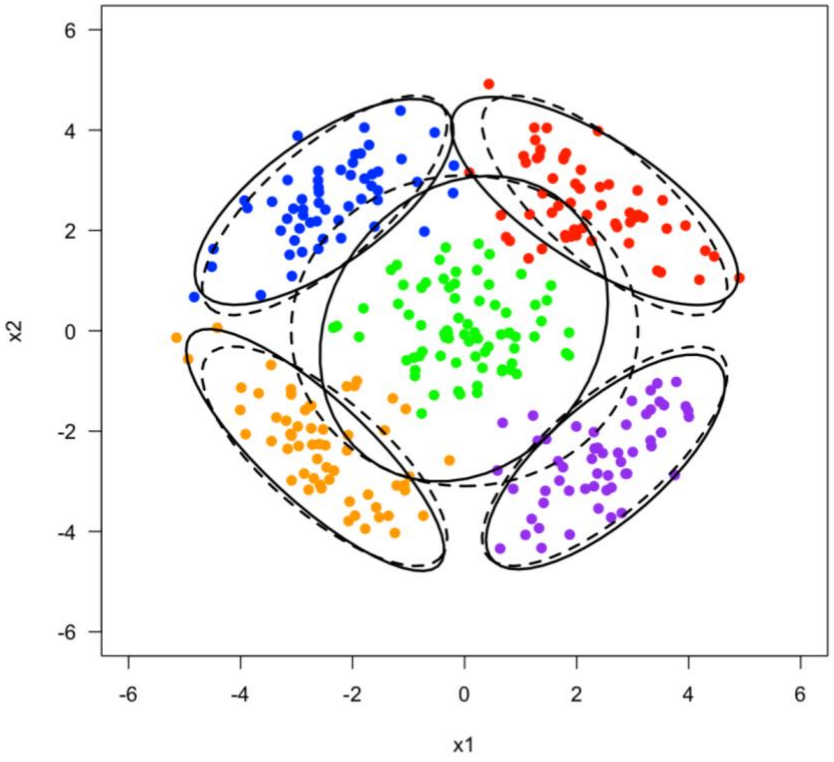
```
while(i<100){
  covariances <- sapply(X = 1:5, FUN = function(k) {
  (t(X) - matrix(centroids[k,], 2, 300)) %*% diag(H[,k]) %*% t(t(X) - matrix(centroids[k,], 2, 300))/ sum(H[,k]) }, simplify = "array")

  priors <- colMeans(H)

  H <- t(sapply(1:300, function(n){
    row <- sapply(1:5, function(k){density(X[n,], k)})
    return(row / sum(row))
  }))

  centroids <- (t(H) %*% X ) / matrix(colSums(H), 5, 2)
  i <- i + 1
}
```

4. Printed the centroid values:



```
             [,1]        [,2]
[1,] -2.50882875  2.56888718
[2,]  2.51478346 -2.61408096
[3,] -0.01262437  0.04159552
[4,] -2.66784042 -2.37667828
[5,]  2.32119796  2.58704409
```

5. Plotted the clustering results along with the original densities, results were as shown below:


6
4
2
0
-2
-4
-6
x2
-6
-4
-2
0
2
4
6
x1