Exercise 5.4

```
#####################################################
# Exercise 5.4 - 27th november 2015
#
# Davide Pedranz (362504)
# Jakub Amanowicz (362594)
# Hongmei Liang (292520)
#####################################################


# calculate the distance between 2 points
distance <- function(p1, p2) {
  return(sqrt(sum((p2 - p1) ^ 2)))
}

# X: n x d data-matrix
# k: number of clusters
# means: k x d matrix
assignCluster <- function(X, k, means) {
  # returns the cluster assignment vector

  # for each row in the matrix (= each data point)
  clusters <- apply(X, 1, function(x) {

    # calculate the distances from each mean point
    dists <- apply(means, 1, function(m) {
      return(distance(m, x))
    })

    # return the index of the nearest cluster's center
    return(match(min(dists), dists))
  })

  return(clusters)
}

# X: n x d data-matrix
# k: number of clusters
# clusters: cluster assignment vector (1 dimension)
updateMean <- function(X, k, clusters) {
  # update the mean points
```

```
  # sum of values for each cluster (cluster indexed by row)
  sums <- matrix(nrow = k, ncol = dim(X)[2], data = 0)
  counts <- seq(from=0, to=0, length.out=k)

  # sum values for cluster
  for(i in 1:length(clusters)) {
    cluster <- clusters[i]
    sums[cluster, ] <- sums[cluster, ] + X[i,]
    counts[cluster] <- counts[cluster] + 1
  }

  # returns the means
  return(sums / counts)
}

# X: n x d data-matrix
# k: number of clusters
# maxIter: maximum number of iterations
#
# return values
# cluster: cluster assignment vector
# means: k x d matrix. The i-th row represents the mean of the
    i-th cluster
myKmeans <- function(X, k, maxIter){
  #initialize the centers
  #in this implementation, we choose k points from X as the
      initial means
  means <- X[sample(1:dim(X)[1],k),]

  #loop for clustering
  for (i in 1:maxIter){
    #assign each data point to the closest cluster
    #based on the distance to the mean point of the cluster
    cluster <- assignCluster(X, k, means)

    #update the mean points
    newMeans <- updateMean(X, k, cluster)

    #if the difference between the new and previous means is
        small enough we stop
    difference <- abs(newMeans - means) < c(rep(0.0001, k))
    if(all(difference)){
      break
    }
```

```
    else{ #we use the new means for next iteration
      means <- newMeans
    }
    #NOTE: this stopping criterion is just an example, you can
        also compare the differences between
    #the new and previous clusters.
  }

  result <- list(cluster, means)
  return(result)
}
```