```
# Created by Fabio Vitor at 19-05-2017
# K-means algorithm

getDistance <- function(a, b) {
  return (sqrt(Reduce("+", (a-b) ^ 2)))
}

numberOfGroups = 7
numberOfDots   = 1000
dimensionOfDots= 2
minCoord = 0
maxCoord = 100

for(i in c(1:dimensionOfDots)) {
  if(i == 1) {
    dots <- runif(numberOfDots, minCoord, maxCoord)
  } else {
    dots <- rbind(dots, runif(numberOfDots, minCoord, maxCoord))
  }
}

centroids <- dots[,c(1:numberOfGroups)]

previousGrouping <- -1
grouping <- -1

while((numberOfDots * numberOfGroups) != length(which(grouping == previousGrouping))) {
  previousGrouping = grouping
  for(i in c(1:numberOfGroups)) {
    for(j in c(1:numberOfDots)) {
      if(j == 1) {
        dist <- getDistance(dots[,j], centroids[,i])
      } else {
        dist <- cbind(dist, getDistance(dots[,j], centroids[,i]))
      }
    }
    if(i == 1) {
      distances <- dist
    } else {
      distances <- rbind(distances, dist)
    }
  }
  for(i in c(1:numberOfDots)) {
    if(i == 1) {
      grouping <- ifelse(distances[,i] == min(distances[,i]), 1, 0) ^ 2
    } else {
      grouping <- cbind(grouping, ifelse(distances[,i] == min(distances[,i]), 1, 0) ^ 2)
    }
  }
  for(i in c(1:numberOfGroups)) {
    for(j in c(1:dimensionOfDots)) {
      centroids[j,i] <- (Reduce("+", dots[j, which(grouping[i,] == 1)]) / length(which(grouping[i,] == 1)))
    }
  }
}

# Printing ------------------------------------------------------------

if(dimensionOfDots == 2) {
  for(i in c(1:numberOfGroups)) {
    plot(dots[1,which(grouping[i,] == 1)], dots[2,which(grouping[i,] == 1)], main="Visualization",
         sub="visualization of the K-means algorithm", xlab="X values", ylab="Y values",
         xlim=c(minCoord, maxCoord), ylim=c(minCoord, maxCoord), pch=20, col=i)
    par(new=T)
  }
}
```