K means:

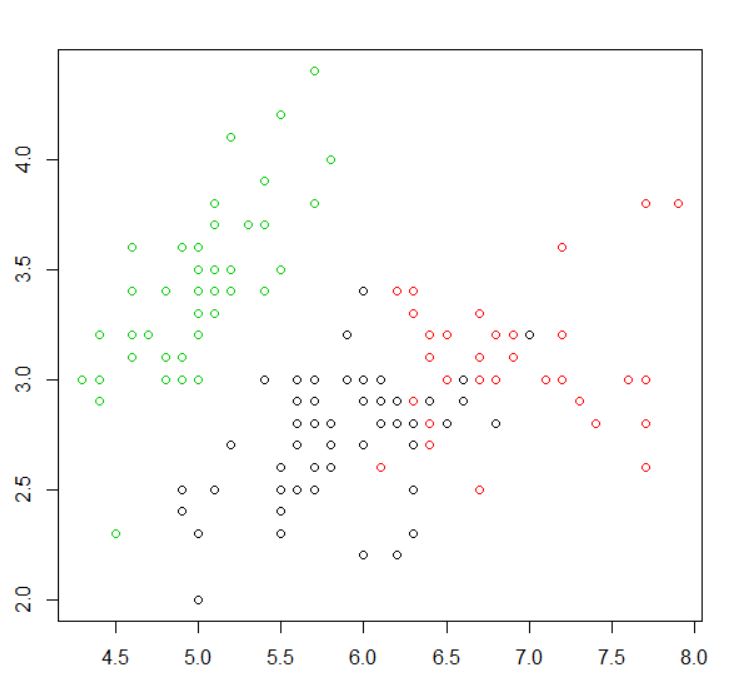
> iris3 <- iris

> iris2$Species <- NULL

> (kmeans.result <- kmeans(iris2, 3))

> plot(iris2[c("Petal.Length", "Petal.Width")], col = kmeans.result$cluster)

> points(kmeans.result$centers[,c("Petal.Length", "Petall.Width")], col = 1:3, + pch = 8, cex=2)



Kmedians:

> library(fpc)

> pamk.result <- pamk(iris2)

> # number of clusters

> pamk.result$nc

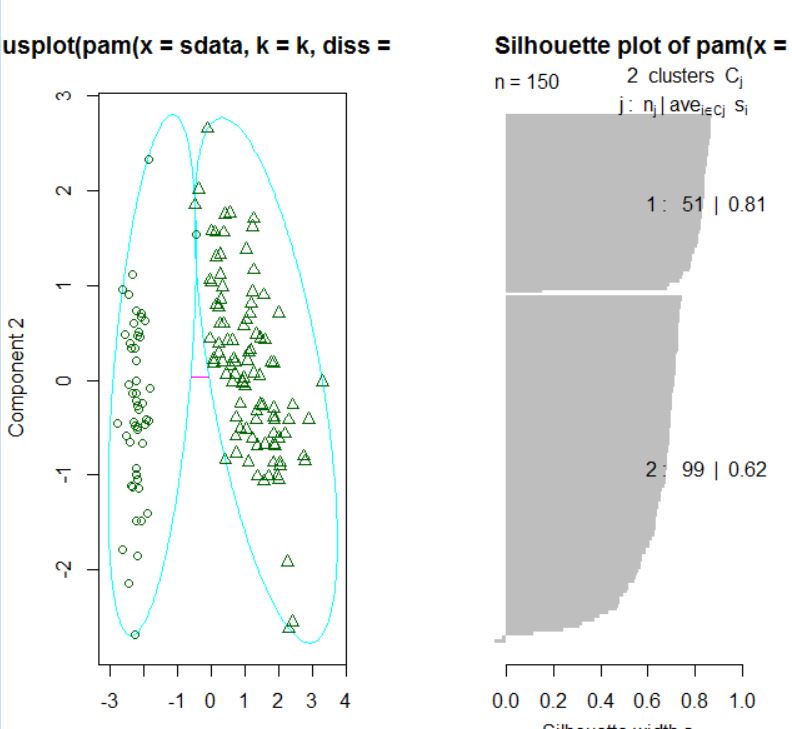
> # check clustering against actual species

> table(pamk.result$pamobject$clustering, iris$Species)

> layout(matrix(c(1,2),1,2)) # 2 graphs per page

> plot(pamk.result$pamobject)

> layout(matrix(1))



Hierchial Clustering

> layout(matrix(c(1,2),1,2))

> plot(pamk.result$pamobject)

> q()

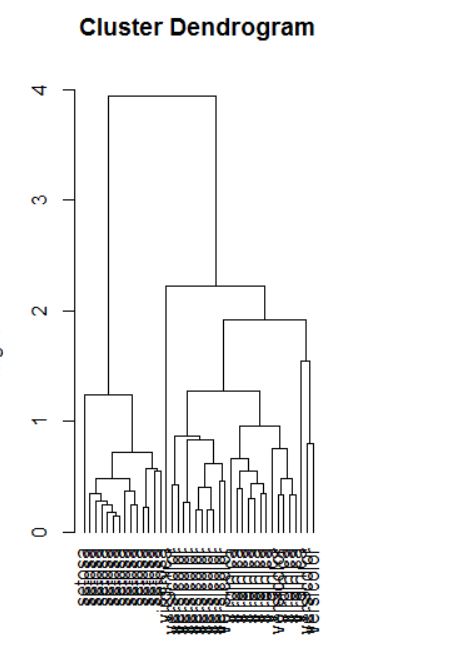
> idx <- sample(1:dim(iris)[1], 40)

> irisSample <- iris[idx,]

> irisSample$Species <- NULL

> hc <- hclust(dist(irisSample), method="ave")

> plot(hc, hang = -1, labels=iris$Species[idx]



Expectation Maximization:

Expectation Maximization

-----------------------------------------------

library(mclust) # load mclust library

x1 = runif(20) # generate 20 random random numbers for x axis (1st class)

y1 = runif(20) # generate 20 random random numbers for y axis (1st class)

x2 = runif(20) # generate 20 random random numbers for x axis (2nd class)

y2 = runif(20) # generate 20 random random numbers for y axis (2nd class)

rx = range(x1,x2) # get the axis x range

ry = range(y1,y2) # get the axis y range

plot(x1, y1, xlim=rx, ylim=ry) # plot the first class points

points(x2, y2 ) # plot the second class points

mix = matrix(nrow=40, ncol=2) # create a dataframe matrix

mix[,1] = c(x1, x2) # insert first class points into the matrix

mix[,2] = c(y1, y2) # insert second class points into the matrix

mixclust = Mclust(mix) # initialize EM with hierarchical clustering, execute BIC and EM

