**Answer the following questions:**

1. For kMeans, how did you determine the number of clusters?  How can an elbow analysis help (see the readings this week)?

For HaireyeColor dataset we are having 4 attributes as Hair, Eye, Sex, Freq and in this by using system learning we would like to have system to identify how many clusters can this dataset can be made, and for that I had made a effective classification and clustering using elbow analysis and I had identified number of clusters identified are 3, here is the code added down with its identification.

R Code:   
  
set.seed(123)

# Compute and plot wss for k = 2 to k = 15

k.max <- 15 # Maximal number of clusters

wss <- sapply(1:k.max,

function(k){kmeans(he2, k, nstart=10 )$tot.withinss})

plot(1:k.max, wss,

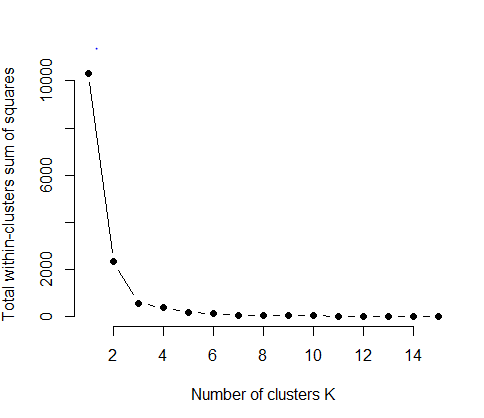
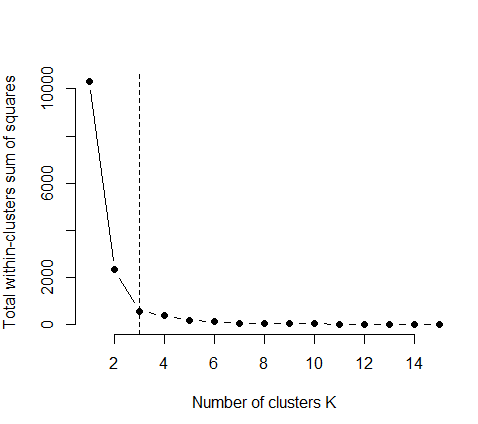
type="b", pch = 19, frame = FALSE,

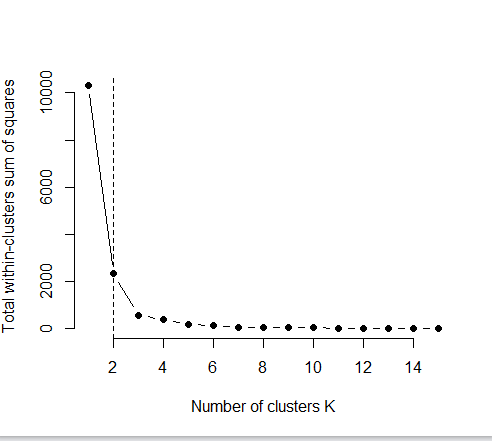
xlab="Number of clusters K",

ylab="Total within-clusters sum of squares")

abline(v = 2, lty =2)  
  
  
Output:

|  |
| --- |
| > set.seed(123)  > k.max <- 15 # Maximal number of clusters  > wss <- sapply(1:k.max,  + function(k){kmeans(he2, k, nstart=10 )$tot.withinss})  > plot(1:k.max, wss,  + type="b", pch = 19, frame = FALSE,  + xlab="Number of clusters K",  + ylab="Total within-clusters sum of squares")  > abline(v = 2, lty =2) |
|  |
| |  | | --- | |  | |

  
  
  
By above two graphs I had changed the v=2 and v=3 , as the identified attribute was Sex, and then as per changing the value we get 2 and 3 points, and then to get a identification in Sex attribute we have male and female as information which is of two values in it. By using Elbow analysis this is what the classification we get as information, and then this would be very much helpful for classifying the cluster to output for k means for best results.

1. Do all three (or 4, if you use both, pam() and pamk()) algorithms produce the same clusters?  Why do you see the differences?

Now identifying the clustering , re clarifying it with three other means , and pam and identifying the best output with the identification to it.

R Code:   
  
#Running the executable file   
  
install.packages("h2o") ## Should already be installed

install.packages("cluster") ## Should be installed by default

install.packages("fpc") ## For density-based clustering

library(readr)

he1 <- data.frame(HairEyeColor)

he2 <- HairEyeColor

he2$sex <- NULL  
  
Output:

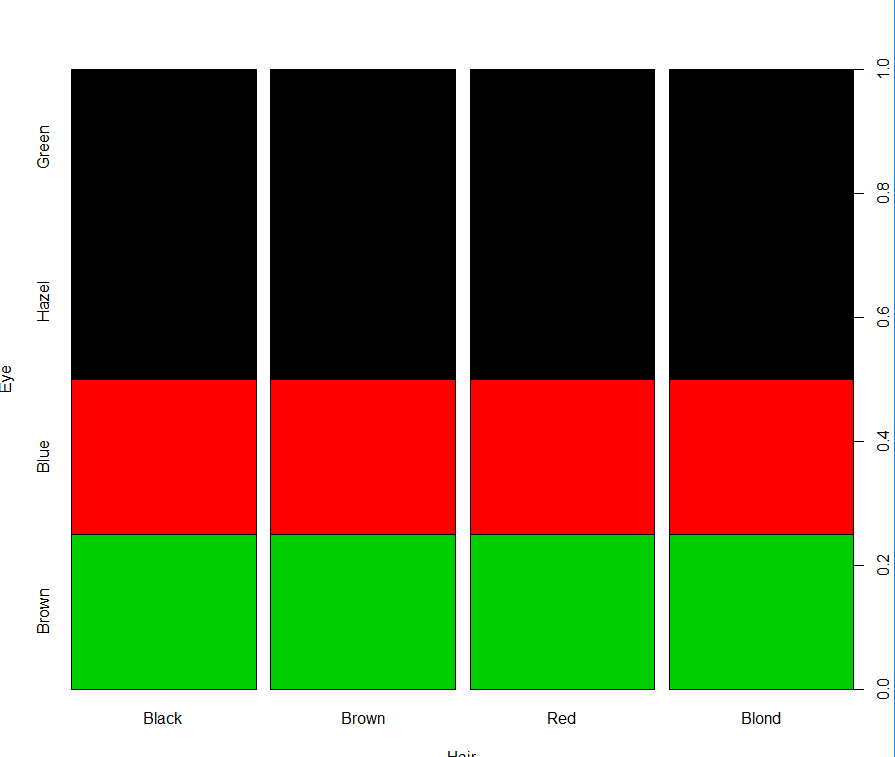
|  |
| --- |
| > library(readr)  > he1 <- data.frame(HairEyeColor)  > he2 <- HairEyeColor  > he2$sex <- NULL  Warning message:  In he2$sex <- NULL : Coercing LHS to a list |
|  |
| |  | | --- | | > | |

K means:   
  
R Code:   
  
(kmeans.result <- kmeans(he1, 3))

table(he1$Sex, kmeans.result$cluster)

plot(he1[c("Hair", "Eye")], col = kmeans.result$cluster)  
  
  
Output:

|  |
| --- |
| > table(he1$Sex, kmeans.result$cluster)    1 2 3  Male 11 2 3  Female 11 2 3  > plot(he1[c("Hair", "Eye")], col = kmeans.result$cluster) |
|  |
| |  | | --- | | > | |

  
  
Based on the hair color , and we are having 4 colors, and then graph is been identified with number of first cluster with male and brown color and and second cluster with female with other colors.   
  
R code:  
  
library(reshape2)

barplot(he,col=Hair)

mm = melt(HairEyeColor)

mm <- within(mm, {

color <- tolower(Hair)

color <- ifelse(color == 'blond', 'yellow', color)

color1 <- tolower(Eye)

color1 <- ifelse(color1 == 'hazel', 'gold', color1)

value <- value / 2

value1 <- value

})

mm <- melt(mm, id.vars = -(4:5))

cols <- c(apply(mm[1:16, c('color','color1')], 1, c))

library(ggplot2)

ggplot(data = mm, aes(x = interaction(Hair, Eye), y = value, fill = interaction(variable, interaction(Hair, Eye)))) +

geom\_bar(stat = 'identity') + facet\_grid(Sex ~ .) +

theme(legend.position = 'none') +

scale\_fill\_manual(values = cols)  
  
Output:  
  
> library(reshape2)

> barplot(he,col=Hair)

Error in barplot.default(he, col = Hair) : object 'Hair' not found

> mm = melt(HairEyeColor)

> mm <- within(mm, {

+ color <- tolower(Hair)

+ color <- ifelse(color == 'blond', 'yellow', color)

+ color1 <- tolower(Eye)

+ color1 <- ifelse(color1 == 'hazel', 'gold', color1)

+ value <- value / 2

+ value1 <- value

+ })

> mm <- melt(mm, id.vars = -(4:5))

> cols <- c(apply(mm[1:16, c('color','color1')], 1, c))

> plot(he1[c("Hair", "Eye")], col = kmeans.result$cluster)

R Code:   
  
library(fpc)

pamk.result <- pamk(HairEyeColor)

pamk.result$nc

## Let's check the clusters against the actual species

table(pamk.result$pamobject$clustering, he1$Sex)

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

plot(pamk.result$pamobject)

layout(matrix(1)) ## change back to one graph per page

library(cluster)

pam.result <- pam(he1, 3)

table(pam.result$clustering, he1$Sex)

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

plot(pam.result)

layout(matrix(1)) ## change back to one graph per page

Output:  
  
> library(fpc)

> pamk.result <- pamk(HairEyeColor)

> pamk.result$nc

[1] 3

> table(pamk.result$pamobject$clustering, he1$Sex)

Male Female

1 3 3

2 2 2

3 11 11

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

> plot(pamk.result$pamobject)

> layout(matrix(1)) ## change back to one graph per page

> library(cluster)

> pam.result <- pam(he1, 3)

> table(pam.result$clustering, he1$Sex)

Male Female

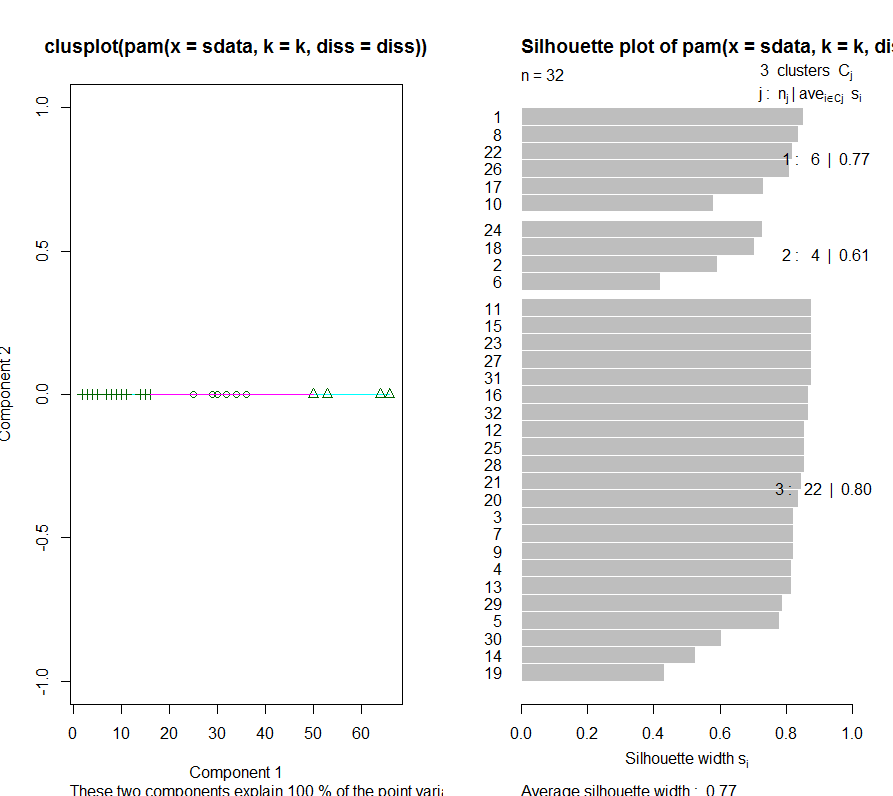
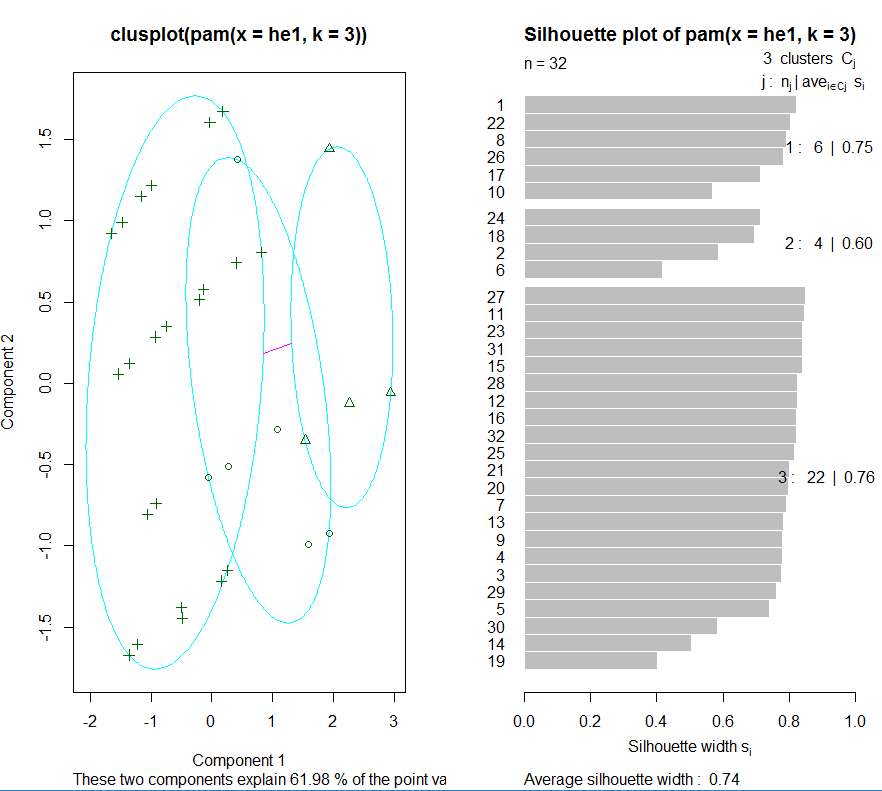
1 3 3

2 2 2

3 11 11

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

> plot(pam.result)

  
  
  
  
  
  
Hierarchial Clustering:  
  
  
R Code:   
  
  
########################### Hierarchical Clustering ##########################

## We will perform hierarchical clustering with hclust()

## We first draw a sample of 40 records from the iris data, so that the clustering plot will

## not be overcrowded. Same as before, variable Species is removed from the data. After that,

## we apply hierarchical clustering to the data.

idx <- sample(1:dim(he1)[1], 32)

hesample <- he1[idx,]

he1$Sex <- NULL

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

plot(hc, hang = -1, labels=he1$Sex[idx])

## Let's cut the tree into 3 clusters

rect.hclust(hc, k=3)

groups <- cutree(hc, k=3)   
  
Output:

> idx <- sample(1:dim(he1)[1], 32)

> hesample <- he1[idx,]

> he1$Sex <- NULL

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

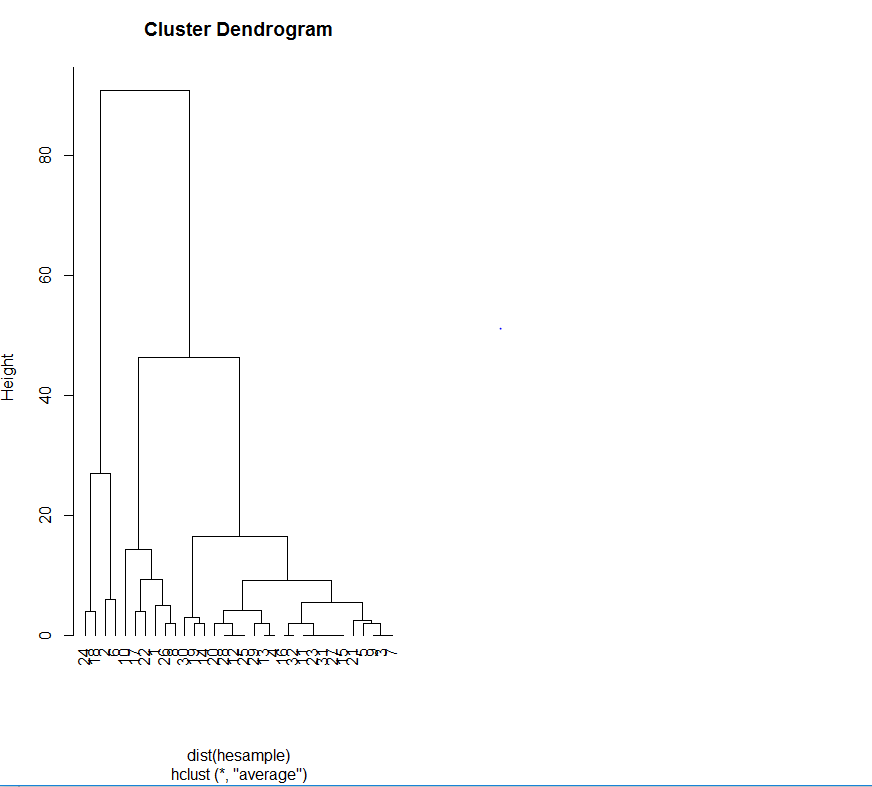
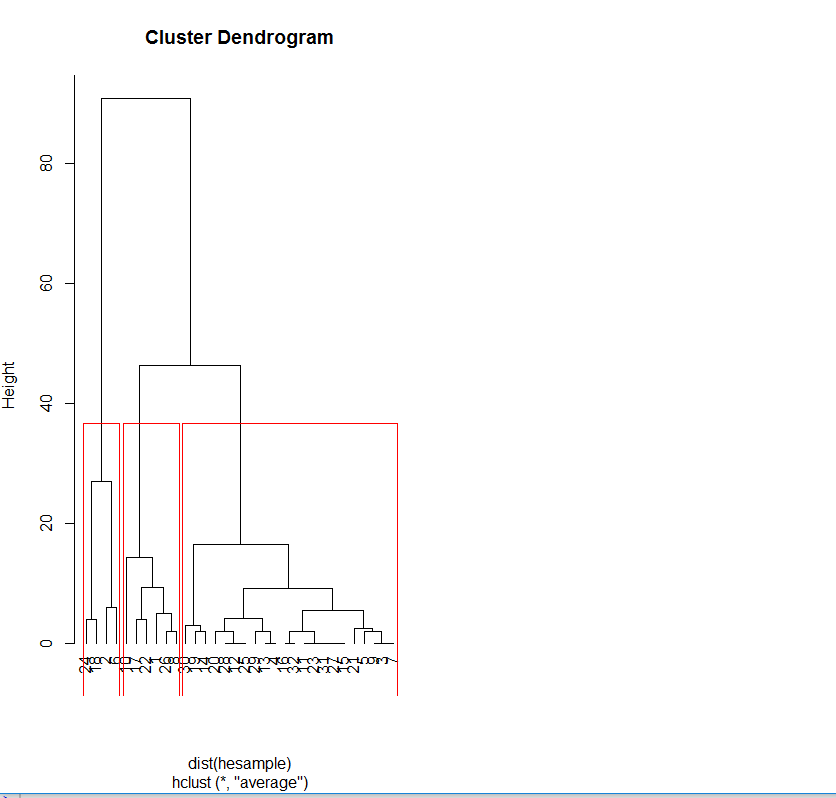
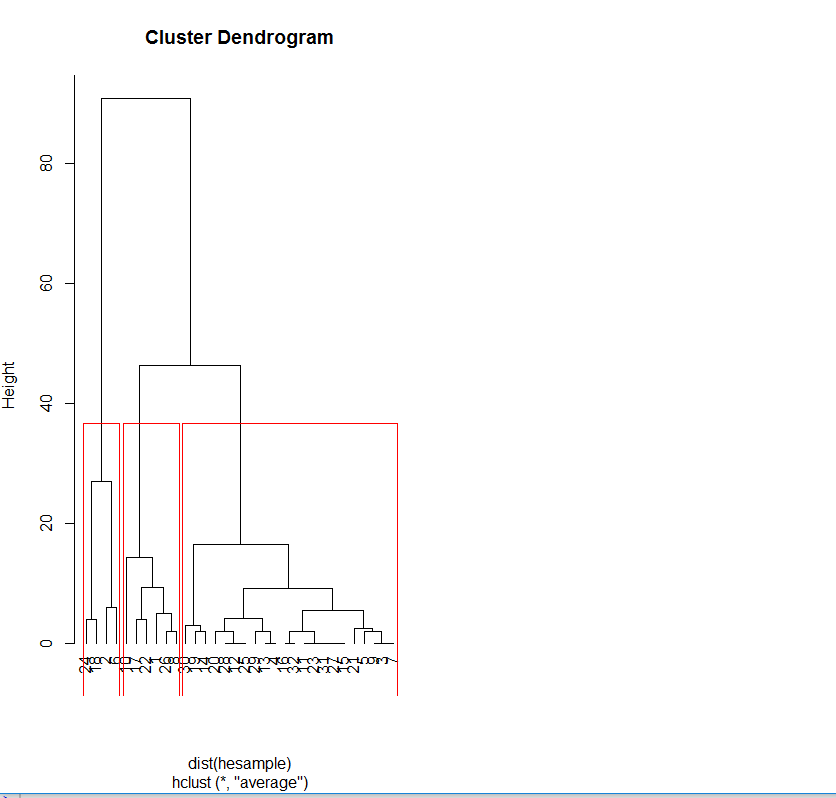
Warning message:

In dist(hesample) : NAs introduced by coercion

> plot(hc, hang = -1, labels=he1$Sex[idx])

> rect.hclust(hc, k=3)

> groups <- cutree(hc, k=3)

  
  
  
  
  
  
  
  
  
  
Density :   
  
R Code:   
  
######################## Density-based Clustering #########################

install.packages("fpc")

library(fpc)

he3 <- HairEyeColor[-5] ## Remove class

ds <- dbscan(he3, eps=0.32, MinPts=5)

## Let's compare the clusters with the original class labels

table(ds$cluster, he2$Sex)

## In the table, "1" to "3" in the first column are three identified clusters, while "0" stands

## for noises or outliers, i.e., objects that are not assigned to any clusters. The noises are

## shown as black circles in the plot below:

plot(ds, he3)

## Or let's display the clusters in a scatter plot using the first and fourth columns of the data.

plot(ds, he3[c(1,4)])

## Even better, let's use plotcluster. Yeah!

plotcluster(he3, ds$cluster)

## Now let's get crazy and do some predicting! The clustering model can be used to label new data,

## based on the similarity between new data and the clusters. The following example draws a sample

## of 10 objects from iris and adds small noises to them to make a new dataset for labeling.

## The random noises are generated with a uniform distribution using function runif().

## Create a new dataset for labeling

set.seed(435)

idx <- sample(1:nrow(he1), 10)

newData <- he1[idx,-5]

newData <- newData + matrix(runif(10\*4, min=0, max=0.2), nrow=10, ncol=4)

## Label new data

myPred <- predict(ds, he1, newData)

## Plot the result with new data as asterisks

plot(he1[c(1,4)], col=1+ds$cluster)

points(newData[c(1,4)], pch="\*", col=1+myPred, cex=3)

## Check cluster labels

table(myPred, iris$Species[idx])

Output:   
  
> set.seed(435)

> idx <- sample(1:nrow(he1), 10)

> newData <- he1[idx,-5]

> newData <- newData + matrix(runif(10\*4, min=0, max=0.2), nrow=10, ncol=4)

Warning messages:

1: In Ops.factor(left, right) : ‘+’ not meaningful for factors

2: In Ops.factor(left, right) : ‘+’ not meaningful for factors

> myPred <- predict(ds, he1, newData)

Error in knn1(data, newdata, 1:n.orig) : no missing values are allowed

> plot(he1[c(1,4)], col=1+ds$cluster)

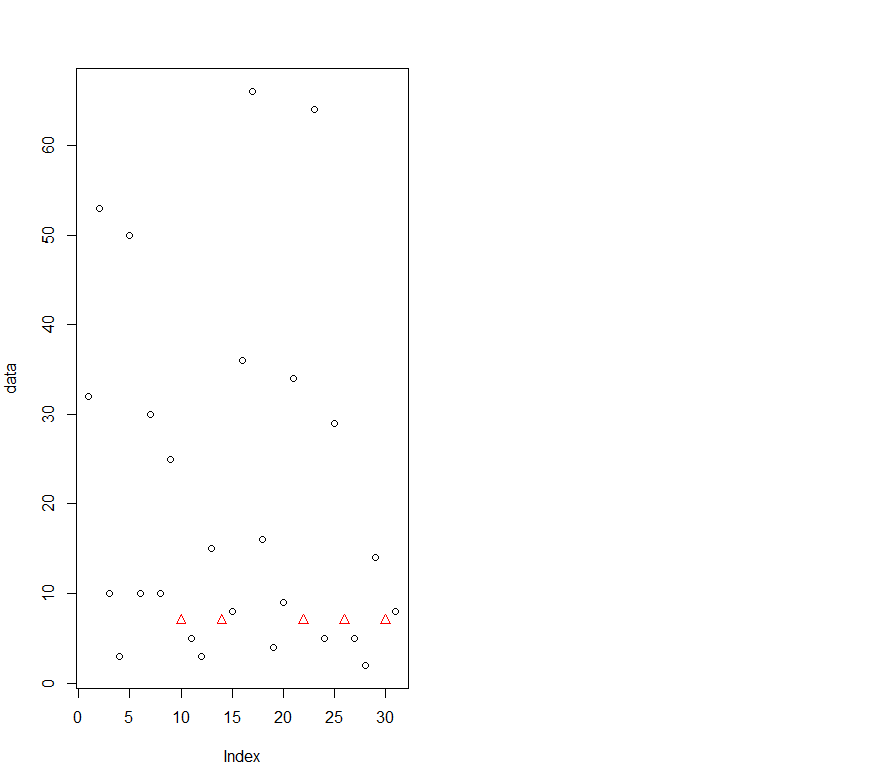
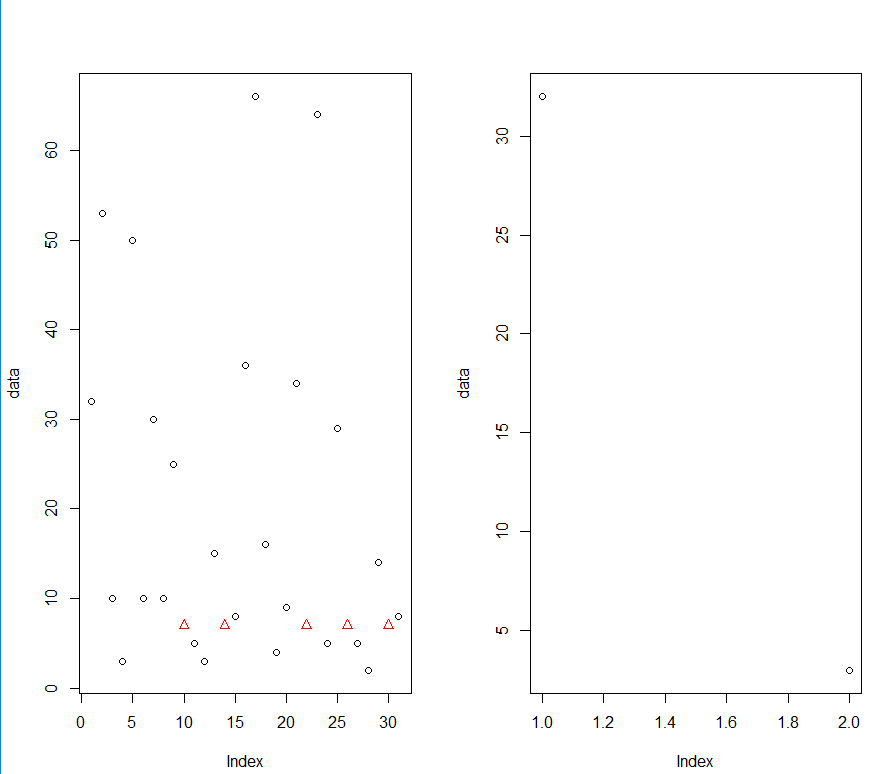
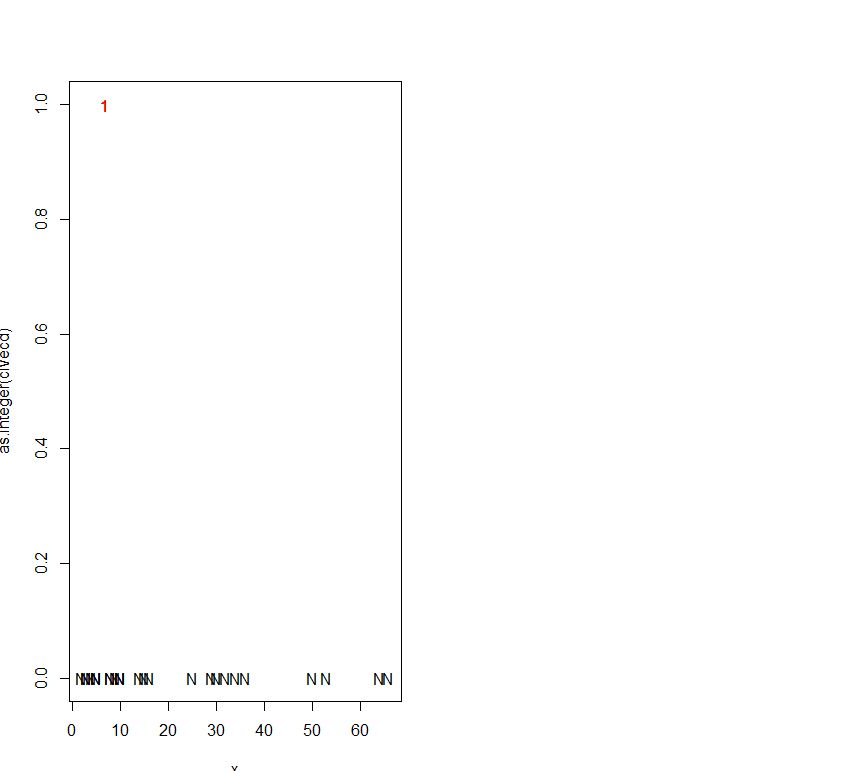
Error in `[.data.frame`(he1, c(1, 4)) : undefined columns selected

> points(newData[c(1,4)], pch="\*", col=1+myPred, cex=3)

Error in `[.data.frame`(newData, c(1, 4)) : undefined columns selected

> table(myPred, iris$Species[idx])

Error in table(myPred, iris$Species[idx]) : object 'myPred' not found

  
  
  
  
  
  
  
  
  
  
3. What does each of the algorithms tell you about your assigned dataset?  Which algorithms produces the most logical results given the specific conditions of your dataset?

My algorithms are defines that number of cluster identified that are 3, but each datapoints identified in each cluster are different in different cluster, so it would have three different clusters then it would be a different as per the different samples get produced, and then to have it with different values. Number of data points and outliers are chnagn because it is having many outliers, taking that in consideration I would say density would and should generally as a good results than compared with other dataset, and in then to have good logical results with density than compared with other modules used.

Mean value resolved between the datapoints are compared to be a different data points in a single cluster, as per the sample dataset with k value varied.