View(crime)

> scale\_crime <- scale(crime[,-1])

> View(scale\_crime)

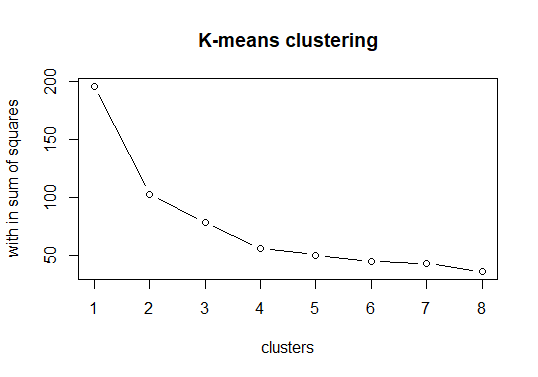
#finding number of clusters

> wss=(nrow(scale\_crime)-1)\*sum(apply(scale\_crime,2,var))

> for(i in 2:8) wss[i]=sum(kmeans(scale\_crime,centers=i)$withinss)

> plot(1:8,wss,type = "b",xlab = "clusters",ylab = "with in sum of squares",main = "K-means clustering")

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2 is the number of clusters \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

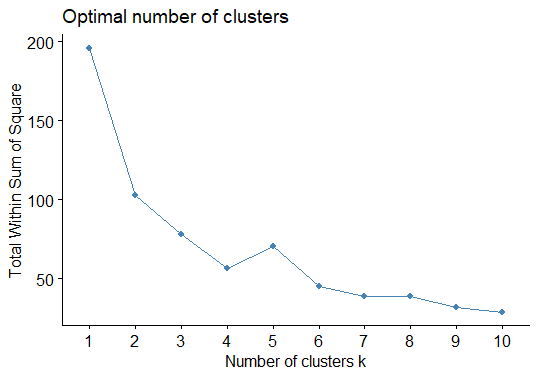


#to find optimal no of clusters

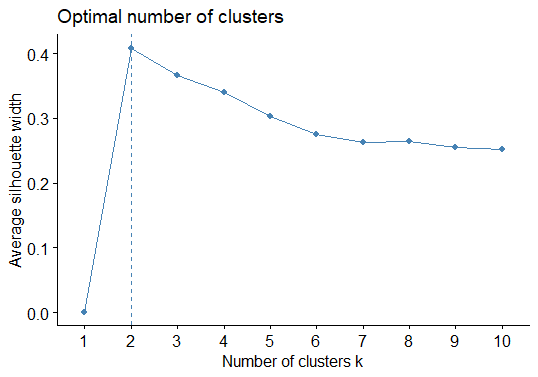
install.packages("factoextra")

library(factoextra)

> fviz\_nbclust(scale\_crime,method = 'wss',FUNcluster = kmeans)



> fviz\_nbclust(scale\_crime,method = 'silhouette',FUNcluster = kmeans)



> fviz\_nbclust(scale\_crime,method = 'gap\_stat',FUNcluster = kmeans)

Clustering k = 1,2,..., K.max (= 10): .. done

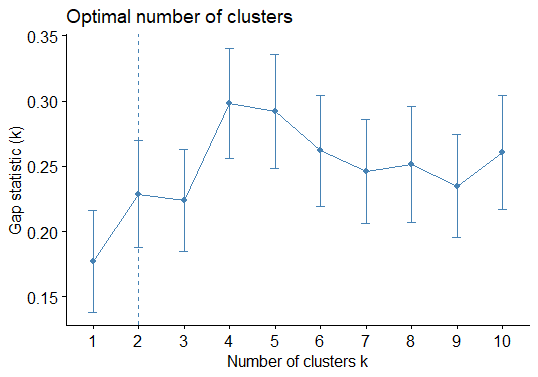
Bootstrapping, b = 1,2,..., B (= 100) [one "." per sample]:

.................................................. 50

.................................................. 100

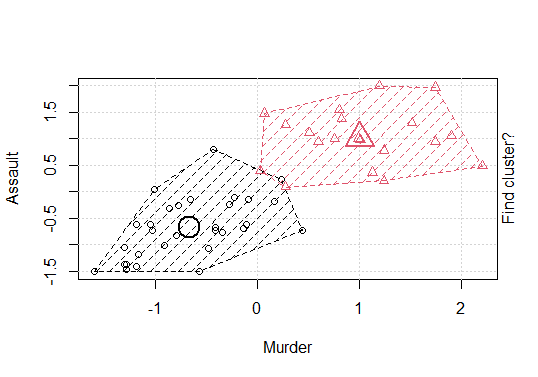
>

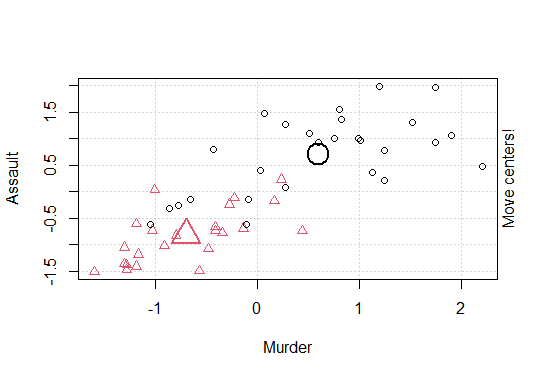
> # from all the elbow plots it is clear that optimal number of clusters is 2



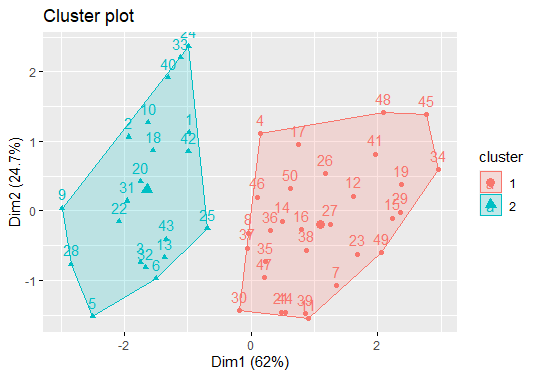
> final <- kmeans(scale\_crime,2)

> finalanim <- kmeans.ani(scale\_crime,2)





fviz\_cluster(final,data = crime[-1])



View(finaldata)

> aggregate(crime[,-1],by=list(final$cluster),FUN = mean)

Group.1 Murder Assault UrbanPop Rape

1 1 4.870 114.4333 63.63333 15.94333

2 2 12.165 255.2500 68.40000 29.16500

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Hierarchical Clustering\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> setDT(crime)

> head(crime)

X Murder Assault UrbanPop Rape

1: Alabama 13.2 236 58 21.2

2: Alaska 10.0 263 48 44.5

3: Arizona 8.1 294 80 31.0

4: Arkansas 8.8 190 50 19.5

5: California 9.0 276 91 40.6

6: Colorado 7.9 204 78 38.7

> #lets check missing values

> colSums(is.na(crime))

X Murder Assault UrbanPop Rape

0 0 0 0 0

> #some of the variables have missing values. Let's impute the missing values with median.

> #scale the variables

> scaled\_cm<-scale(crime[,-1])

> #we'll first calculate a distance matrix based on Euclidean measure. Then using the hclust function, we can implement hierarchical clustering.

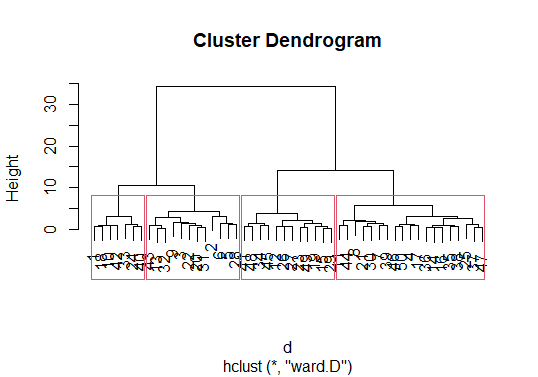
> #Hierarchical Clustering

> d <- dist(scaled\_cm,method = "euclidean") #distance matrix

> h\_clust <- hclust(d, method = "ward") #clustering

The "ward" method has been renamed to "ward.D"; note new "ward.D2"

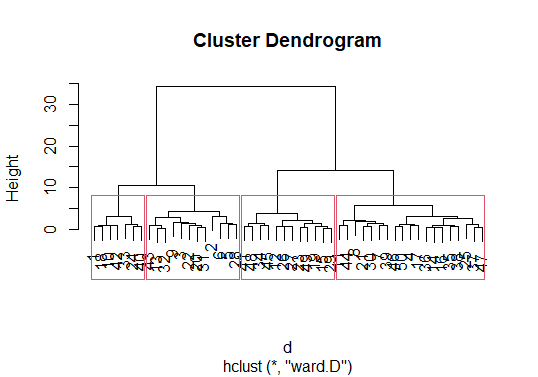
> plot(h\_clust,labels = crime$V1) #dendrogram



> # how can you estimate the number of clusters? Going by the logic of horizontal cut, four clusters are evident. Let's see!

>

> rect.hclust(h\_clust,k=4)



> #To look at which observation went into which cluster, you can write:

> #extract clusters

> groups <- cutree(h\_clust,k=4)

> groups

[1] 1 2 2 3 2 2 3 3 2 1 3 4 2 3 4 3 3 1 4 2 3 2 4 1 3 4 4 2 4 3 2 2 1 4

[35] 3 3 3 3 3 1 4 1 2 3 4 3 3 4 4 3

> groups <- cutree(h\_clust,k=4)

> groups

[1] 1 2 2 3 2 2 3 3 2 1 3 4 2 3 4 3 3 1 4 2 3 2 4 1 3 4 4 2 4 3 2 2 1 4

[35] 3 3 3 3 3 1 4 1 2 3 4 3 3 4 4 3

> #o implement PCA, we'll use princomp base function. For our convenience, we'll take only the first two components.

>

> #pca

> pcmp <- princomp(scaled\_cm)

> pred\_pc <- predict(pcmp, newdata=scaled\_cm)[,1:2]

> #Now, we'll create a data frame having pc values and their corresponding clusters. Then, using ggplot2 we'll create the plot.

> comp\_dt <- cbind(as.data.table(pred\_pc),cluster = as.factor(groups), Labels = crime$V1)

> ggplot(comp\_dt,aes(Comp.1,Comp.2))+

+ geom\_point(aes(color = cluster),size=3)

>

