Ayachit Madhukar Exercise16

## Exercise 16

### reading file to create data frame

setwd("~/MadR/Workspaces/dsc520")  
data.cluster=read.csv("data/clustering-data.csv")

### a. Creating initial plot and observing data /structure using str() and summary()

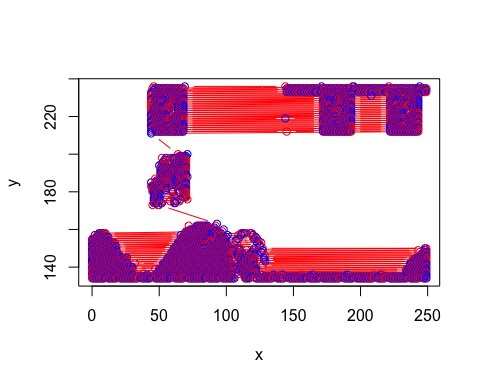
summary(data.cluster)

## x y   
## Min. : 0.0 Min. :134.0   
## 1st Qu.: 56.0 1st Qu.:141.0   
## Median : 82.0 Median :154.0   
## Mean :109.6 Mean :175.7   
## 3rd Qu.:180.0 3rd Qu.:218.0   
## Max. :249.0 Max. :236.0

str(data.cluster)

## 'data.frame': 4022 obs. of 2 variables:  
## $ x: int 46 69 144 171 194 195 221 244 45 47 ...  
## $ y: int 236 236 236 236 236 236 236 236 235 235 ...

plot(data.cluster,col=c("red","blue"),type="b")



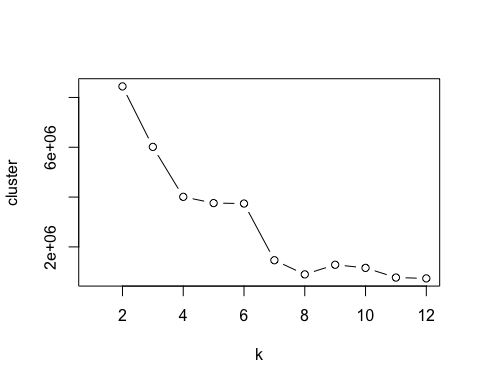
### b. Preparing clusters for k=2 to 12 and plotting all the results for visualization

library(factoextra)

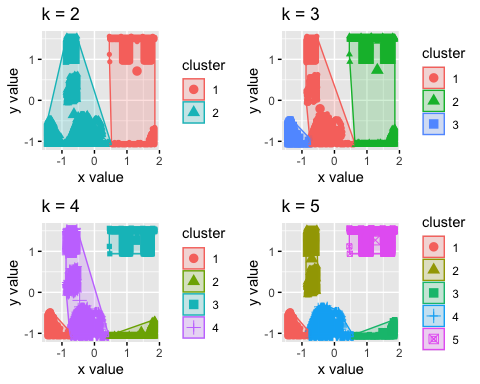
## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

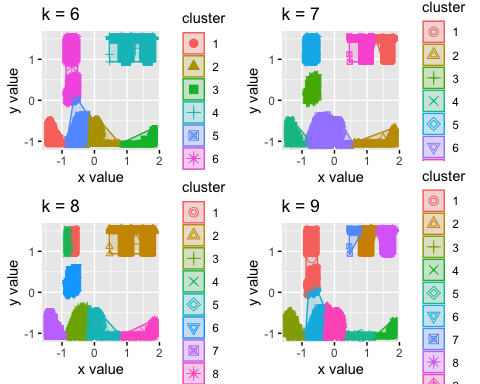
library(gridExtra)  
  
  
  
result<-NULL  
for (i in 2:12) result[i]<-sum(kmeans(data.cluster,centers = i)$tot.withinss)  
plot(1:12,result,type="b",xlab = "k",ylab = "cluster")



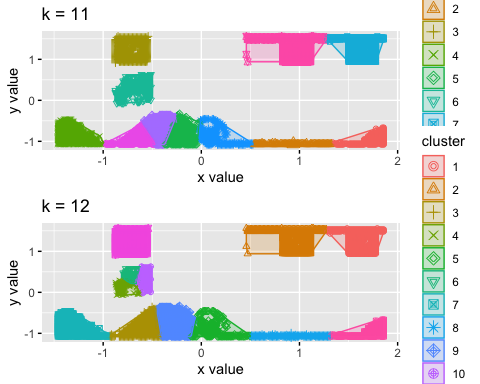
results2<-kmeans(data.cluster,2)  
results3<-kmeans(data.cluster,3)  
results4<-kmeans(data.cluster,4)  
results5<-kmeans(data.cluster,5)  
results6<-kmeans(data.cluster,6)  
results7<-kmeans(data.cluster,7)  
results8<-kmeans(data.cluster,8)  
results9<-kmeans(data.cluster,9)  
results10<-kmeans(data.cluster,10)  
results11<-kmeans(data.cluster,11)  
results12<-kmeans(data.cluster,12)  
  
  
p2 <- fviz\_cluster(results2, geom = "point", data = data.cluster) + ggtitle("k = 2")  
p3 <- fviz\_cluster(results3, geom = "point", data = data.cluster) + ggtitle("k = 3")  
p4 <- fviz\_cluster(results4, geom = "point", data = data.cluster) + ggtitle("k = 4")  
p5 <- fviz\_cluster(results5, geom = "point", data = data.cluster) + ggtitle("k = 5")  
p6 <- fviz\_cluster(results6, geom = "point", data = data.cluster) + ggtitle("k = 6")  
p7 <- fviz\_cluster(results7, geom = "point", data = data.cluster) + ggtitle("k = 7")  
p8 <- fviz\_cluster(results8, geom = "point", data = data.cluster) + ggtitle("k = 8")  
p9 <- fviz\_cluster(results9, geom = "point", data = data.cluster) + ggtitle("k = 9")  
p10 <- fviz\_cluster(results10, geom = "point", data = data.cluster) + ggtitle("k = 10")  
p11 <- fviz\_cluster(results11, geom = "point", data = data.cluster) + ggtitle("k = 11")  
p12 <- fviz\_cluster(results12, geom = "point", data = data.cluster) + ggtitle("k = 12")  
  
  
grid.arrange(p2,p3,p4,p5,nrow = 2)



grid.arrange(p6,p7,p8,p9,nrow = 2)

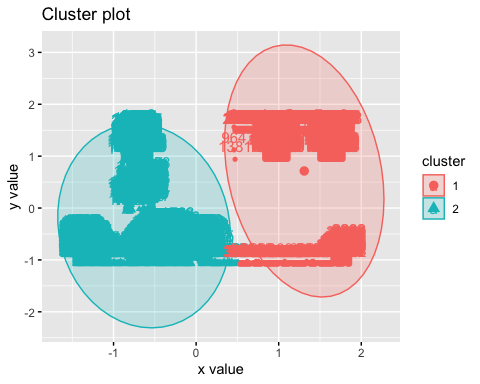


grid.arrange(p11,p12,nrow = 2)

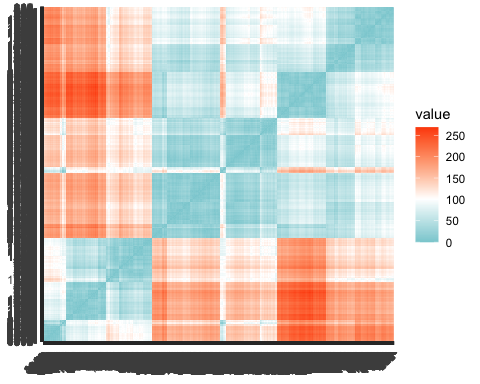


### c. measuring distance and depicting in the graph

fviz\_cluster(results2, data.cluster, ellipse.type = "norm")



distance2 <- get\_dist(data.cluster)  
fviz\_dist(distance2, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07") )



### Euclidean Distance

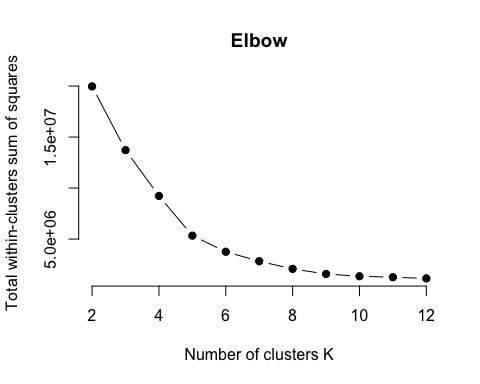
model <- kmeans(data.cluster, centers = 2)  
data.cluster$clustered <-model$cluster  
  
head(data.cluster)

## x y clustered  
## 1 46 236 1  
## 2 69 236 1  
## 3 144 236 2  
## 4 171 236 2  
## 5 194 236 2  
## 6 195 236 2

data.cluster$x.dist <- model$centers[data.cluster$clustered,][1] - data.cluster$x  
data.cluster$y.dist <- model$centers[data.cluster$clustered,][2] - data.cluster$y  
  
  
data.cluster$euclid <- sqrt((data.cluster$x.dist \*\* 2) + (data.cluster$y.dist \*\* 2))  
  
#data.cluster$euclid

### Elbow

library(purrr)  
library(cluster)  
  
  
# function to compute total within-cluster sum of square   
wss <- function(k) {  
 kmeans(data.cluster, k, nstart = 10 )$tot.withinss  
}  
  
# Compute and plot wss for k = 1 to k = 15  
k.values <- 2:12  
  
# extract wss for 2-12 clusters  
wss\_values <- map\_dbl(k.values, wss)  
  
  
plot(k.values, wss\_values,main="Elbow",  
 type="b", pch = 19, frame = FALSE,   
 xlab="Number of clusters K",  
 ylab="Total within-clusters sum of squares")



fviz\_nbclust(data.cluster, kmeans, method = "silhouette")

