Assignment 3 - Unsupervised Learning: K-Means Clustering

Shubham Maheshwari

2024-10-14

## 1. Data Transformation and Descriptive Analysis

### 1. Rename all variables with your initials appended (just as was done in assignment 1)

getwd()

## [1] "C:/PROG8435 Data Analytics/Assignment 3"

df <- read.table("PROG8435-24F-Assign03.txt", sep=",", header=TRUE)  
df <- as.data.frame(df)  
  
colnames(df) <- paste(colnames(df),"SM",sep = "\_")  
  
head(df)

## Age\_SM BP\_SM Sq\_SM DL\_SM PU\_SM  
## 1 35.2 398 375 493 39  
## 2 63.7 261 531 514 14  
## 3 23.5 157 134 272 3  
## 4 37.2 160 114 236 10  
## 5 45.9 128 156 354 14  
## 6 47.7 313 350 430 35

str(df)

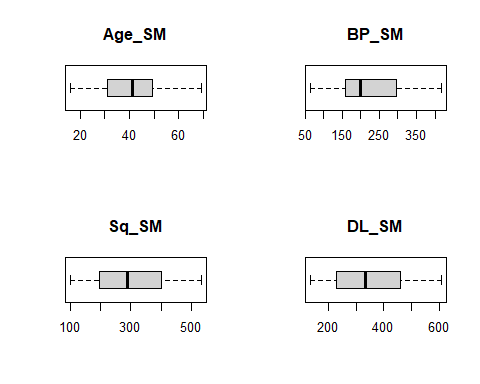
## 'data.frame': 709 obs. of 5 variables:  
## $ Age\_SM: num 35.2 63.7 23.5 37.2 45.9 47.7 50.1 32.7 29.4 27.1 ...  
## $ BP\_SM : int 398 261 157 160 128 313 278 371 381 151 ...  
## $ Sq\_SM : int 375 531 134 114 156 350 413 451 411 256 ...  
## $ DL\_SM : int 493 514 272 236 354 430 557 582 490 315 ...  
## $ PU\_SM : int 39 14 3 10 14 35 11 19 22 6 ...

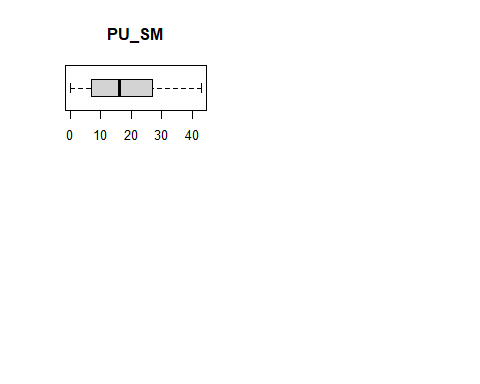
### 

### 2. Create graphical summaries of the data (as demonstrated in class: boxplots or histograms) and comment on any observations you make.

par(mfrow=c(2,2))  
  
for (i in 1:ncol(df)) {  
 if (is.numeric(df[,i])) {  
 boxplot(df[i], main=names(df)[i],  
 horizontal=TRUE, pch=10)  
 }  
}

par(mfrow=c(1,1))



Observation: All the boxplots above are clean without any visible outliers.

### 3.Standardize all of the variables using either of the two functions demonstrated in class. Describe why you chose the method you did.

sta01 <- function(x) {  
 return ((x - min(x)) / (max(x) - min(x)))  
}

***Reasoning: I chose the min-max function because there are no outliers in the dataset.***

df$Age\_Std\_SM <- sta01(df$Age\_SM)  
df$BP\_Std\_SM <- sta01(df$BP\_SM)  
df$Sq\_Std\_SM <- sta01(df$Sq\_SM)  
df$DL\_Std\_SM <- sta01(df$DL\_SM)  
df$PU\_Std\_SM <- sta01(df$PU\_SM)  
  
head(df)

## Age\_SM BP\_SM Sq\_SM DL\_SM PU\_SM Age\_Std\_SM BP\_Std\_SM Sq\_Std\_SM DL\_Std\_SM  
## 1 35.2 398 375 493 39 0.3609023 0.9461756 0.63425926 0.7568710  
## 2 63.7 261 531 514 14 0.8966165 0.5580737 0.99537037 0.8012685  
## 3 23.5 157 134 272 3 0.1409774 0.2634561 0.07638889 0.2896406  
## 4 37.2 160 114 236 10 0.3984962 0.2719547 0.03009259 0.2135307  
## 5 45.9 128 156 354 14 0.5620301 0.1813031 0.12731481 0.4630021  
## 6 47.7 313 350 430 35 0.5958647 0.7053824 0.57638889 0.6236786  
## PU\_Std\_SM  
## 1 0.90697674  
## 2 0.32558140  
## 3 0.06976744  
## 4 0.23255814  
## 5 0.32558140  
## 6 0.81395349

## 

## Clustering

### 1. Create segmentation/cluster schemes for k=2,3,4,5,6,7.

maxk <- 7  
nk <- c(2:maxk)  
wss <- rep(0,maxk-1)

k=6  
  
Clstr <- kmeans(df[,c("Age\_Std\_SM", "BP\_Std\_SM")], iter.max=10, centers=k, nstart=10)  
Clstr

## K-means clustering with 6 clusters of sizes 110, 111, 117, 104, 140, 127  
##   
## Cluster means:  
## Age\_Std\_SM BP\_Std\_SM  
## 1 0.6464457 0.1141643  
## 2 0.2634288 0.8714494  
## 3 0.4360902 0.2654657  
## 4 0.7589286 0.6054424  
## 5 0.2113050 0.3049170  
## 6 0.5422710 0.6387098  
##   
## Clustering vector:  
## [1] 2 4 5 3 1 6 6 2 2 5 3 4 6 1 3 6 5 5 3 3 6 1 5 4 4 3 1 6 2 2 3 5 6 6 6 3 5  
## [38] 6 5 2 3 4 2 5 3 5 6 3 3 1 6 6 3 2 6 5 6 2 2 1 3 6 6 4 2 5 4 6 6 3 4 6 3 5  
## [75] 1 1 6 3 4 5 2 1 1 2 3 4 1 1 3 1 2 6 2 2 5 6 1 6 6 1 6 4 5 2 5 4 4 4 3 6 3  
## [112] 5 6 2 1 4 1 1 4 1 1 6 6 4 2 6 6 5 5 2 4 2 1 6 3 4 5 6 2 4 2 4 5 2 4 6 4 5  
## [149] 5 6 3 5 6 5 4 1 4 6 3 5 6 5 3 6 5 5 1 6 3 1 5 4 1 6 2 4 2 6 6 6 1 3 6 3 5  
## [186] 1 1 3 6 1 3 3 6 1 3 1 6 1 5 2 6 3 5 1 3 3 6 1 4 4 3 3 5 2 2 4 4 5 6 5 1 6  
## [223] 1 3 6 6 2 3 6 6 4 5 6 3 1 1 1 1 4 2 2 5 2 5 1 5 3 2 4 2 3 5 5 5 4 2 6 5 5  
## [260] 3 2 2 2 5 4 6 2 1 5 4 4 1 1 4 3 5 5 3 4 3 5 2 6 2 2 1 3 2 4 2 5 6 4 4 6 1  
## [297] 1 1 5 1 1 6 2 2 5 6 6 4 5 4 6 2 1 1 5 5 2 4 2 5 4 2 3 2 3 5 3 2 4 5 5 5 6  
## [334] 3 6 5 1 2 2 6 5 3 6 5 5 5 6 5 3 6 3 6 5 1 6 6 5 2 5 3 2 3 6 3 6 2 5 2 1 2  
## [371] 3 6 6 2 2 4 2 1 3 5 2 2 5 5 3 5 4 3 1 1 2 1 4 2 1 6 3 1 1 6 4 4 5 2 5 5 3  
## [408] 5 5 3 4 4 2 1 4 2 1 4 2 4 3 4 1 5 4 4 5 1 4 6 1 6 6 6 3 2 2 3 1 5 3 6 2 4  
## [445] 3 3 3 2 6 1 5 4 1 1 5 5 4 3 6 2 3 6 6 1 6 5 6 5 6 5 4 2 6 4 5 2 6 1 3 1 2  
## [482] 4 4 2 1 3 2 1 5 6 6 6 5 5 6 5 4 4 2 3 1 4 6 6 4 2 1 4 2 1 1 6 4 1 6 4 5 2  
## [519] 5 5 5 2 4 5 1 5 6 5 6 5 2 3 3 5 3 1 3 6 3 1 4 5 1 3 6 6 3 3 5 5 1 4 4 3 4  
## [556] 1 5 3 2 5 3 6 3 3 2 3 3 4 2 5 4 3 2 1 1 5 2 5 5 5 6 2 4 3 5 5 5 1 2 3 6 6  
## [593] 5 3 3 5 4 2 5 5 2 4 5 5 4 1 4 4 2 3 4 1 6 1 2 2 6 1 5 3 3 2 3 6 1 1 1 2 4  
## [630] 4 1 3 1 5 1 5 3 6 5 1 2 1 4 3 5 1 3 1 2 1 2 6 2 6 3 6 4 6 2 5 4 4 4 5 6 6  
## [667] 1 5 4 3 4 4 2 3 2 3 2 3 1 5 3 5 5 4 6 5 5 4 5 3 6 4 2 1 4 3 3 5 6 3 5 6 2  
## [704] 1 3 1 3 3 5  
##   
## Within cluster sum of squares by cluster:  
## [1] 1.357328 1.546545 1.300671 1.352010 1.276998 1.878145  
## (between\_SS / total\_SS = 89.3 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

df$cluster <- factor(Clstr$cluster)   
head(df)

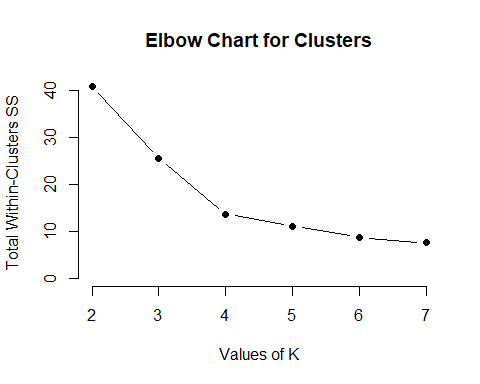
## Age\_SM BP\_SM Sq\_SM DL\_SM PU\_SM Age\_Std\_SM BP\_Std\_SM Sq\_Std\_SM DL\_Std\_SM  
## 1 35.2 398 375 493 39 0.3609023 0.9461756 0.63425926 0.7568710  
## 2 63.7 261 531 514 14 0.8966165 0.5580737 0.99537037 0.8012685  
## 3 23.5 157 134 272 3 0.1409774 0.2634561 0.07638889 0.2896406  
## 4 37.2 160 114 236 10 0.3984962 0.2719547 0.03009259 0.2135307  
## 5 45.9 128 156 354 14 0.5620301 0.1813031 0.12731481 0.4630021  
## 6 47.7 313 350 430 35 0.5958647 0.7053824 0.57638889 0.6236786  
## PU\_Std\_SM cluster  
## 1 0.90697674 2  
## 2 0.32558140 4  
## 3 0.06976744 5  
## 4 0.23255814 3  
## 5 0.32558140 1  
## 6 0.81395349 6

centers <- data.frame(cluster=factor(1:k), Clstr$centers)  
  
  
wss[k-1] <- Clstr$tot.withinss

### 

### 2. Create the WSS plots as demonstrated in class and select a suitable k value based on the “elbow”. [NOTE – Use the code that I provided to do this. Using other functions will yield different results.]

for (k in nk) {  
 Clstr <- kmeans(df[, c("Age\_Std\_SM", "BP\_Std\_SM")], centers = k, nstart = 10)  
 wss[k - 1] <- Clstr$tot.withinss  
}  
  
plot(2:maxk, wss,  
 type="b", pch = 19, frame = FALSE,  
 main="Elbow Chart for Clusters",  
 xlab="Values of K",  
 ylab="Total Within-Clusters SS",  
 ylim=c(0,max(wss)))



***Observation: I’m choosing the elbow to be at K=4 since visible drastic change in the data points occur after that point.***

## Evaluation of Clusters

### 

### 1. Based on the “k” chosen above, create a scatter plot showing the clusters and colour-coded datapoints for each of “k-1”, “k”, “k+1”. For example, if you think the “elbow” is at k=5 create the charts for k=4, k=5 and k=6

# (k-1)  
  
k=3  
Clstr<- kmeans(df[,c(6:7)], iter.max=10, centers=k, nstart=10)  
Clstr

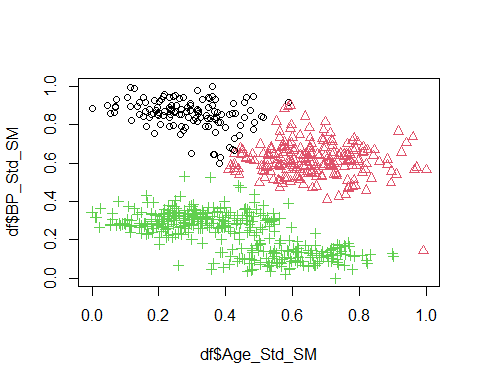
## K-means clustering with 3 clusters of sizes 126, 214, 369  
##   
## Cluster means:  
## Age\_Std\_SM BP\_Std\_SM  
## 1 0.2846998 0.8549620  
## 2 0.6584920 0.6156575  
## 3 0.4118222 0.2375151  
##   
## Clustering vector:  
## [1] 1 2 3 3 3 2 2 1 1 3 3 2 2 3 3 2 3 3 3 3 2 3 3 2 2 3 3 2 1 1 3 3 2 2 1 3 3  
## [38] 2 3 1 3 2 1 3 3 3 2 3 3 3 2 2 3 1 2 3 3 1 1 3 3 2 2 2 1 3 2 2 1 3 2 2 3 3  
## [75] 3 3 2 3 2 3 1 3 3 1 3 2 3 3 3 3 1 2 1 1 3 2 3 2 1 3 2 2 3 1 3 2 2 2 3 2 3  
## [112] 3 1 1 3 2 3 3 2 3 3 2 2 2 1 2 1 3 3 1 2 1 3 2 3 2 3 2 1 2 1 2 3 1 2 2 2 3  
## [149] 3 2 3 3 2 3 2 3 2 2 3 3 2 3 3 2 3 3 3 2 3 3 3 2 3 2 1 2 1 2 2 2 3 3 2 3 3  
## [186] 3 3 3 2 3 3 3 2 3 3 3 2 3 3 1 1 3 3 3 3 3 2 3 2 2 3 3 3 1 1 2 2 3 2 3 3 2  
## [223] 3 3 2 2 1 3 2 2 2 3 2 3 3 3 3 3 2 1 1 3 1 3 3 3 3 1 2 1 3 3 3 3 2 1 2 3 3  
## [260] 3 1 1 1 3 2 2 1 3 3 2 2 3 3 2 3 3 3 3 2 3 3 1 2 1 1 3 3 1 2 1 3 2 2 2 1 3  
## [297] 3 3 3 3 3 2 1 1 3 2 2 2 3 2 2 1 3 3 3 3 1 2 1 3 2 1 3 1 3 3 3 1 2 3 3 3 2  
## [334] 3 2 3 3 1 1 2 3 3 3 3 3 3 1 3 3 2 3 2 3 3 1 2 3 1 3 3 1 3 2 3 2 1 3 1 3 1  
## [371] 3 2 2 1 1 2 1 3 3 3 1 1 3 3 3 3 2 3 3 3 1 3 2 1 3 2 3 3 3 2 2 2 3 1 3 3 3  
## [408] 3 3 3 2 2 1 3 2 1 3 2 1 2 3 2 3 3 2 2 3 3 2 2 3 2 2 1 3 1 1 3 3 3 3 2 1 2  
## [445] 3 3 3 1 2 3 3 2 3 3 3 3 2 3 2 1 3 2 2 3 2 3 2 3 1 3 2 1 1 2 3 1 2 3 3 3 1  
## [482] 2 2 1 3 3 1 3 3 2 1 2 3 3 2 3 2 2 1 3 3 2 2 2 2 1 3 2 1 3 3 2 2 3 1 2 3 1  
## [519] 3 3 3 1 2 3 3 3 2 3 2 3 1 3 3 3 3 3 3 2 3 3 2 3 3 3 2 2 3 3 3 3 3 2 2 3 2  
## [556] 3 3 3 1 3 3 2 3 3 1 3 3 2 1 3 2 3 1 3 3 3 1 3 3 3 2 1 2 3 3 3 3 3 1 3 2 2  
## [593] 3 3 3 3 2 1 3 3 1 2 3 3 2 3 2 2 1 3 2 3 2 3 1 1 2 3 3 3 3 1 3 2 3 3 3 1 2  
## [630] 2 3 3 3 3 3 3 3 2 3 3 1 3 2 3 3 3 3 3 1 3 1 2 1 2 3 2 2 2 1 3 2 2 2 3 2 2  
## [667] 3 3 2 3 2 2 1 3 1 3 1 3 3 3 3 3 3 2 3 3 3 2 3 3 2 2 1 3 2 3 3 3 2 3 3 1 1  
## [704] 2 3 3 3 3 3  
##   
## Within cluster sum of squares by cluster:  
## [1] 2.400921 5.085429 17.949512  
## (between\_SS / total\_SS = 68.9 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

df$cluster <- factor(Clstr$cluster) # Adding Cluster tags to variables  
head(df)

## Age\_SM BP\_SM Sq\_SM DL\_SM PU\_SM Age\_Std\_SM BP\_Std\_SM Sq\_Std\_SM DL\_Std\_SM  
## 1 35.2 398 375 493 39 0.3609023 0.9461756 0.63425926 0.7568710  
## 2 63.7 261 531 514 14 0.8966165 0.5580737 0.99537037 0.8012685  
## 3 23.5 157 134 272 3 0.1409774 0.2634561 0.07638889 0.2896406  
## 4 37.2 160 114 236 10 0.3984962 0.2719547 0.03009259 0.2135307  
## 5 45.9 128 156 354 14 0.5620301 0.1813031 0.12731481 0.4630021  
## 6 47.7 313 350 430 35 0.5958647 0.7053824 0.57638889 0.6236786  
## PU\_Std\_SM cluster  
## 1 0.90697674 1  
## 2 0.32558140 2  
## 3 0.06976744 3  
## 4 0.23255814 3  
## 5 0.32558140 3  
## 6 0.81395349 2

centers <- data.frame(cluster=factor(1:k), Clstr$centers)

plot(df$Age\_Std\_SM, df$BP\_Std\_SM,  
 col=df$cluster, pch=as.numeric(df$cluster))



# k   
k=4  
Clstr<- kmeans(df[,c(6:7)], iter.max=10, centers=k, nstart=10)  
Clstr

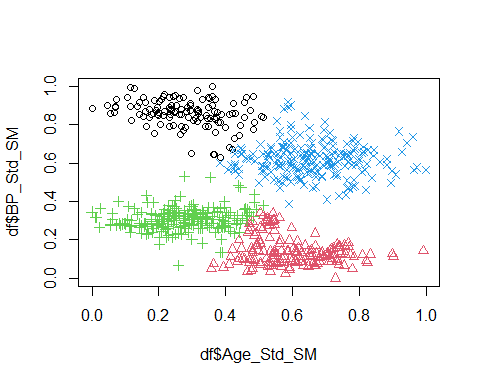
## K-means clustering with 4 clusters of sizes 123, 157, 212, 217  
##   
## Cluster means:  
## Age\_Std\_SM BP\_Std\_SM  
## 1 0.2803197 0.8580345  
## 2 0.5980078 0.1422926  
## 3 0.2758370 0.3068844  
## 4 0.6539534 0.6183470  
##   
## Clustering vector:  
## [1] 1 4 3 3 2 4 4 1 1 3 2 4 4 2 2 4 3 3 3 3 4 2 3 4 4 2 2 4 1 1 2 3 4 4 1 2 3  
## [38] 4 3 1 3 4 1 3 2 3 4 3 3 2 4 4 2 1 4 3 3 1 1 2 2 4 4 4 1 3 4 4 1 2 4 4 2 3  
## [75] 2 2 4 3 4 3 1 2 2 1 3 4 2 2 3 2 1 4 1 1 3 4 2 4 4 2 4 4 3 1 3 4 4 4 3 4 3  
## [112] 3 1 1 2 4 2 2 4 2 2 4 4 4 1 4 1 3 3 1 4 1 2 4 2 4 3 4 1 4 1 4 3 1 4 4 4 3  
## [149] 3 4 2 3 4 3 4 2 4 4 3 3 4 3 3 4 3 3 2 4 3 2 3 4 2 4 1 4 1 4 4 4 2 3 4 2 3  
## [186] 2 2 2 4 2 3 3 4 2 3 2 4 2 3 1 4 3 3 2 3 3 4 2 4 4 2 2 3 1 1 4 4 3 4 3 2 4  
## [223] 2 2 4 4 1 2 4 4 4 3 4 3 2 2 2 2 4 1 1 3 1 3 2 3 2 1 4 1 2 3 3 3 4 1 4 3 3  
## [260] 3 1 1 1 3 4 4 1 2 3 4 4 2 2 4 2 3 3 3 4 2 3 1 4 1 1 2 2 1 4 1 3 4 4 4 1 2  
## [297] 2 2 3 2 2 4 1 1 3 4 4 4 3 4 4 1 2 2 3 3 1 4 1 3 4 1 3 1 3 3 2 1 4 3 3 3 4  
## [334] 2 4 3 2 1 1 4 3 3 3 3 3 3 1 3 3 4 3 4 3 2 1 4 3 1 3 2 1 3 4 2 4 1 3 1 2 1  
## [371] 2 4 4 1 1 4 1 2 3 3 1 1 3 3 3 3 4 2 2 2 1 2 4 1 2 4 2 2 2 4 4 4 3 1 3 3 2  
## [408] 3 3 3 4 4 1 2 4 1 2 4 1 4 2 4 2 3 4 4 3 2 4 4 2 4 4 1 3 1 1 3 2 3 3 4 1 4  
## [445] 3 3 4 1 4 2 3 4 2 2 3 3 4 3 4 1 3 4 4 2 4 3 4 3 4 3 4 1 1 4 3 1 4 2 2 2 1  
## [482] 4 4 1 2 2 1 2 3 4 1 4 3 3 4 3 4 4 1 3 2 4 4 4 4 1 2 4 1 2 2 4 4 2 1 4 3 1  
## [519] 3 3 3 1 4 3 2 3 4 3 4 3 1 3 3 3 2 2 2 4 2 2 4 3 2 3 4 4 2 3 3 3 2 4 4 2 4  
## [556] 2 3 2 1 3 3 4 2 2 1 3 2 4 1 3 4 3 1 2 2 3 1 3 3 3 4 1 4 3 3 3 3 2 1 3 4 4  
## [593] 3 3 2 3 4 1 3 3 1 4 3 3 4 2 4 4 1 3 4 2 4 2 1 1 4 2 3 3 3 1 3 4 2 2 2 1 4  
## [630] 4 2 3 2 3 2 3 3 4 3 2 1 2 4 3 3 2 3 2 1 2 1 4 1 4 3 4 4 4 1 3 4 4 4 3 4 4  
## [667] 2 3 4 3 4 4 1 2 1 3 1 3 2 3 3 3 3 4 3 3 3 4 3 2 4 4 1 2 4 3 2 3 4 3 3 1 1  
## [704] 2 3 2 3 3 3  
##   
## Within cluster sum of squares by cluster:  
## [1] 2.173901 2.954582 3.478288 5.034602  
## (between\_SS / total\_SS = 83.3 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

df$cluster <- factor(Clstr$cluster) # Adding Cluster tags to variables  
head(df)

## Age\_SM BP\_SM Sq\_SM DL\_SM PU\_SM Age\_Std\_SM BP\_Std\_SM Sq\_Std\_SM DL\_Std\_SM  
## 1 35.2 398 375 493 39 0.3609023 0.9461756 0.63425926 0.7568710  
## 2 63.7 261 531 514 14 0.8966165 0.5580737 0.99537037 0.8012685  
## 3 23.5 157 134 272 3 0.1409774 0.2634561 0.07638889 0.2896406  
## 4 37.2 160 114 236 10 0.3984962 0.2719547 0.03009259 0.2135307  
## 5 45.9 128 156 354 14 0.5620301 0.1813031 0.12731481 0.4630021  
## 6 47.7 313 350 430 35 0.5958647 0.7053824 0.57638889 0.6236786  
## PU\_Std\_SM cluster  
## 1 0.90697674 1  
## 2 0.32558140 4  
## 3 0.06976744 3  
## 4 0.23255814 3  
## 5 0.32558140 2  
## 6 0.81395349 4

centers <- data.frame(cluster=factor(1:k), Clstr$centers)

plot(df$Age\_Std\_SM, df$BP\_Std\_SM,  
 col=df$cluster, pch=as.numeric(df$cluster))



# (k+1)  
k=5  
Clstr<- kmeans(df[,c(6:7)], iter.max=10, centers=k, nstart=10)  
Clstr

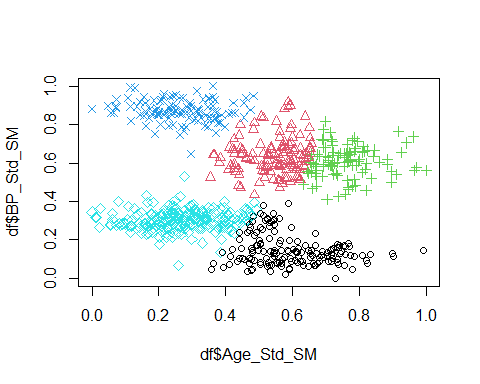
## K-means clustering with 5 clusters of sizes 162, 128, 104, 111, 204  
##   
## Cluster means:  
## Age\_Std\_SM BP\_Std\_SM  
## 1 0.5944491 0.1469066  
## 2 0.5418233 0.6371282  
## 3 0.7589286 0.6054424  
## 4 0.2634288 0.8714494  
## 5 0.2692577 0.3042687  
##   
## Clustering vector:  
## [1] 4 3 5 5 1 2 2 4 4 5 1 3 2 1 1 2 5 5 5 5 2 1 5 3 3 1 1 2 4 4 1 5 2 2 2 1 5  
## [38] 2 5 4 5 3 4 5 1 5 2 5 5 1 2 2 1 4 2 5 2 4 4 1 1 2 2 3 4 5 3 2 2 1 3 2 1 5  
## [75] 1 1 2 5 3 5 4 1 1 4 5 3 1 1 5 1 4 2 4 4 5 2 1 2 2 1 2 3 5 4 5 3 3 3 5 2 5  
## [112] 5 2 4 1 3 1 1 3 1 1 2 2 3 4 2 2 5 5 4 3 4 1 2 1 3 5 2 4 3 4 3 5 4 3 2 3 5  
## [149] 5 2 1 5 2 5 3 1 3 2 5 5 2 5 5 2 5 5 1 2 5 1 5 3 1 2 4 3 4 2 2 2 1 1 2 1 5  
## [186] 1 1 1 2 1 5 5 2 1 5 1 2 1 5 4 2 5 5 1 5 5 2 1 3 3 1 1 5 4 4 3 3 5 2 5 1 2  
## [223] 1 1 2 2 4 1 2 2 3 5 2 5 1 1 1 1 3 4 4 5 4 5 1 5 1 4 3 4 1 5 5 5 3 4 2 5 5  
## [260] 5 4 4 4 5 3 2 4 1 5 3 3 1 1 3 1 5 5 5 3 1 5 4 2 4 4 1 1 4 3 4 5 2 3 3 2 1  
## [297] 1 1 5 1 1 2 4 4 5 2 2 3 5 3 2 4 1 1 5 5 4 3 4 5 3 4 5 4 5 5 1 4 3 5 5 5 2  
## [334] 1 2 5 1 4 4 2 5 5 2 5 5 5 2 5 5 2 5 2 5 1 2 2 5 4 5 1 4 5 2 1 2 4 5 4 1 4  
## [371] 1 2 2 4 4 3 4 1 5 5 4 4 5 5 5 5 3 1 1 1 4 1 3 4 1 2 1 1 1 2 3 3 5 4 5 5 1  
## [408] 5 5 1 3 3 4 1 3 4 1 3 4 3 1 3 1 5 3 3 5 1 3 2 1 2 2 2 5 4 4 5 1 5 5 2 4 3  
## [445] 5 5 1 4 2 1 5 3 1 1 5 5 3 5 2 4 5 2 2 1 2 5 2 5 2 5 3 4 2 3 5 4 2 1 1 1 4  
## [482] 3 3 4 1 1 4 1 5 2 2 2 5 5 2 5 3 3 4 2 1 3 2 2 3 4 1 3 4 1 1 2 3 1 2 3 5 4  
## [519] 5 5 5 4 3 5 1 5 2 5 2 5 4 5 5 5 1 1 1 2 1 1 3 5 1 5 2 2 1 5 5 5 1 3 3 1 3  
## [556] 1 5 1 4 5 5 2 1 1 4 5 1 3 4 5 3 5 4 1 1 5 4 5 5 5 2 4 3 5 5 5 5 1 4 5 2 2  
## [593] 5 5 1 5 3 4 5 5 4 3 5 5 3 1 3 3 4 1 3 1 2 1 4 4 2 1 5 5 5 4 5 2 1 1 1 4 3  
## [630] 3 1 5 1 5 1 5 5 2 5 1 4 1 3 5 5 1 5 1 4 1 4 2 4 2 5 2 3 2 4 5 3 3 3 5 2 2  
## [667] 1 5 3 5 3 3 4 1 4 5 4 5 1 5 5 5 5 3 2 5 5 3 5 1 2 3 4 1 3 5 1 5 2 5 5 2 4  
## [704] 1 5 1 5 1 5  
##   
## Within cluster sum of squares by cluster:  
## [1] 3.195834 1.922069 1.352010 1.546545 3.060257  
## (between\_SS / total\_SS = 86.4 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

df$cluster <- factor(Clstr$cluster) # Adding Cluster tags to variables  
head(df)

## Age\_SM BP\_SM Sq\_SM DL\_SM PU\_SM Age\_Std\_SM BP\_Std\_SM Sq\_Std\_SM DL\_Std\_SM  
## 1 35.2 398 375 493 39 0.3609023 0.9461756 0.63425926 0.7568710  
## 2 63.7 261 531 514 14 0.8966165 0.5580737 0.99537037 0.8012685  
## 3 23.5 157 134 272 3 0.1409774 0.2634561 0.07638889 0.2896406  
## 4 37.2 160 114 236 10 0.3984962 0.2719547 0.03009259 0.2135307  
## 5 45.9 128 156 354 14 0.5620301 0.1813031 0.12731481 0.4630021  
## 6 47.7 313 350 430 35 0.5958647 0.7053824 0.57638889 0.6236786  
## PU\_Std\_SM cluster  
## 1 0.90697674 4  
## 2 0.32558140 3  
## 3 0.06976744 5  
## 4 0.23255814 5  
## 5 0.32558140 1  
## 6 0.81395349 2

centers <- data.frame(cluster=factor(1:k), Clstr$centers)

plot(df$Age\_Std\_SM, df$BP\_Std\_SM,  
 col=df$cluster, pch=as.numeric(df$cluster))



### 2.Based on the WSS plot (3.2) and the charts (4.1) choose one set of clusters that best describes the data.

k=4  
Clstr<- kmeans(df[,c(6:7)], iter.max=10, centers=k, nstart=10)  
Clstr

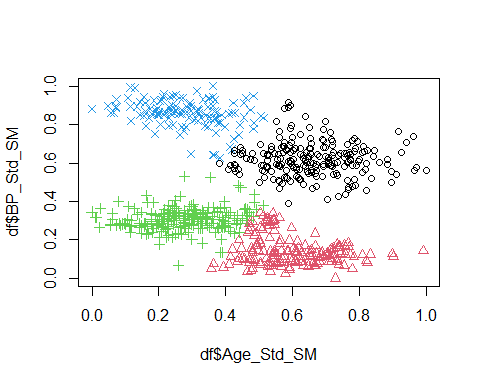
## K-means clustering with 4 clusters of sizes 217, 157, 212, 123  
##   
## Cluster means:  
## Age\_Std\_SM BP\_Std\_SM  
## 1 0.6539534 0.6183470  
## 2 0.5980078 0.1422926  
## 3 0.2758370 0.3068844  
## 4 0.2803197 0.8580345  
##   
## Clustering vector:  
## [1] 4 1 3 3 2 1 1 4 4 3 2 1 1 2 2 1 3 3 3 3 1 2 3 1 1 2 2 1 4 4 2 3 1 1 4 2 3  
## [38] 1 3 4 3 1 4 3 2 3 1 3 3 2 1 1 2 4 1 3 3 4 4 2 2 1 1 1 4 3 1 1 4 2 1 1 2 3  
## [75] 2 2 1 3 1 3 4 2 2 4 3 1 2 2 3 2 4 1 4 4 3 1 2 1 1 2 1 1 3 4 3 1 1 1 3 1 3  
## [112] 3 4 4 2 1 2 2 1 2 2 1 1 1 4 1 4 3 3 4 1 4 2 1 2 1 3 1 4 1 4 1 3 4 1 1 1 3  
## [149] 3 1 2 3 1 3 1 2 1 1 3 3 1 3 3 1 3 3 2 1 3 2 3 1 2 1 4 1 4 1 1 1 2 3 1 2 3  
## [186] 2 2 2 1 2 3 3 1 2 3 2 1 2 3 4 1 3 3 2 3 3 1 2 1 1 2 2 3 4 4 1 1 3 1 3 2 1  
## [223] 2 2 1 1 4 2 1 1 1 3 1 3 2 2 2 2 1 4 4 3 4 3 2 3 2 4 1 4 2 3 3 3 1 4 1 3 3  
## [260] 3 4 4 4 3 1 1 4 2 3 1 1 2 2 1 2 3 3 3 1 2 3 4 1 4 4 2 2 4 1 4 3 1 1 1 4 2  
## [297] 2 2 3 2 2 1 4 4 3 1 1 1 3 1 1 4 2 2 3 3 4 1 4 3 1 4 3 4 3 3 2 4 1 3 3 3 1  
## [334] 2 1 3 2 4 4 1 3 3 3 3 3 3 4 3 3 1 3 1 3 2 4 1 3 4 3 2 4 3 1 2 1 4 3 4 2 4  
## [371] 2 1 1 4 4 1 4 2 3 3 4 4 3 3 3 3 1 2 2 2 4 2 1 4 2 1 2 2 2 1 1 1 3 4 3 3 2  
## [408] 3 3 3 1 1 4 2 1 4 2 1 4 1 2 1 2 3 1 1 3 2 1 1 2 1 1 4 3 4 4 3 2 3 3 1 4 1  
## [445] 3 3 1 4 1 2 3 1 2 2 3 3 1 3 1 4 3 1 1 2 1 3 1 3 1 3 1 4 4 1 3 4 1 2 2 2 4  
## [482] 1 1 4 2 2 4 2 3 1 4 1 3 3 1 3 1 1 4 3 2 1 1 1 1 4 2 1 4 2 2 1 1 2 4 1 3 4  
## [519] 3 3 3 4 1 3 2 3 1 3 1 3 4 3 3 3 2 2 2 1 2 2 1 3 2 3 1 1 2 3 3 3 2 1 1 2 1  
## [556] 2 3 2 4 3 3 1 2 2 4 3 2 1 4 3 1 3 4 2 2 3 4 3 3 3 1 4 1 3 3 3 3 2 4 3 1 1  
## [593] 3 3 2 3 1 4 3 3 4 1 3 3 1 2 1 1 4 3 1 2 1 2 4 4 1 2 3 3 3 4 3 1 2 2 2 4 1  
## [630] 1 2 3 2 3 2 3 3 1 3 2 4 2 1 3 3 2 3 2 4 2 4 1 4 1 3 1 1 1 4 3 1 1 1 3 1 1  
## [667] 2 3 1 3 1 1 4 2 4 3 4 3 2 3 3 3 3 1 3 3 3 1 3 2 1 1 4 2 1 3 2 3 1 3 3 4 4  
## [704] 2 3 2 3 3 3  
##   
## Within cluster sum of squares by cluster:  
## [1] 5.034602 2.954582 3.478288 2.173901  
## (between\_SS / total\_SS = 83.3 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

df$cluster <- factor(Clstr$cluster) # Adding Cluster tags to variables  
head(df)

## Age\_SM BP\_SM Sq\_SM DL\_SM PU\_SM Age\_Std\_SM BP\_Std\_SM Sq\_Std\_SM DL\_Std\_SM  
## 1 35.2 398 375 493 39 0.3609023 0.9461756 0.63425926 0.7568710  
## 2 63.7 261 531 514 14 0.8966165 0.5580737 0.99537037 0.8012685  
## 3 23.5 157 134 272 3 0.1409774 0.2634561 0.07638889 0.2896406  
## 4 37.2 160 114 236 10 0.3984962 0.2719547 0.03009259 0.2135307  
## 5 45.9 128 156 354 14 0.5620301 0.1813031 0.12731481 0.4630021  
## 6 47.7 313 350 430 35 0.5958647 0.7053824 0.57638889 0.6236786  
## PU\_Std\_SM cluster  
## 1 0.90697674 4  
## 2 0.32558140 1  
## 3 0.06976744 3  
## 4 0.23255814 3  
## 5 0.32558140 2  
## 6 0.81395349 1

centers <- data.frame(cluster=factor(1:k), Clstr$centers)

plot(df$Age\_Std\_SM, df$BP\_Std\_SM,  
 col=df$cluster, pch=as.numeric(df$cluster))

 ***This set best describes the data as there were couple of significances visible in other clusters at different k points, The lowermost red cluster were together in k=3 and they seem two different clusters all together as shown in this cluster via blue cluster and green cluster.***

***The black cluster in k=4 seems to give a good representation of the data as   
in k=5, there were quite many overlapping of data points seem in red and green clusters***

### 3. Create summary tables for the segmentation/clustering scheme (selected in step 4.2).

SummClusters <- aggregate(cbind(Age\_SM, BP\_SM, Sq\_SM,DL\_SM,PU\_SM) ~ cluster,  
 df, FUN=function(x) round(mean(x), 0))  
SummClusters

## cluster Age\_SM BP\_SM Sq\_SM DL\_SM PU\_SM  
## 1 1 51 282 380 428 25  
## 2 2 48 114 187 207 6  
## 3 3 31 172 233 277 13  
## 4 4 31 367 428 502 30

### 

### 4. Create suitable descriptive names for each cluster.

Cluster 1 - Higest Strength

Cluster 2 - Oldest with Moderate Strength

Cluster 3 - Least Strength

Cluster 4 - Youngest

### 

### 5. Suggest possible uses for this clustering scheme.

1. Fitness Programs:

Tailored workouts: The clusters could be used to design fitness programs that are specifically tailored to the individual’s age and strength level. For example, younger

Individuals with high strength might benefit from more challenging workouts, while older individuals with lower strength might need more gradual progression.

1. Medical Research and Insights:

The scheme could help identify individuals who may be at risk for muscle loss or other age-related health issues, allowing for targeted interventions to improve their overall health and fitness.

It can also be used to identify populations that may be at higher risk for certain health conditions related to age and strength, such as osteoporosis or cardiovascular disease as seen in the cluster 3.

1. Marketing:

The scheme could also be benefited by identifying specific market segments for fitness products or equipment, such as age-appropriate strength training equipment or supplements.