> A <- read.xlsx(file.choose(),2)

> View(A)

> a <- A[,-1]

> View(a)

> ################# K-means clustering #########################

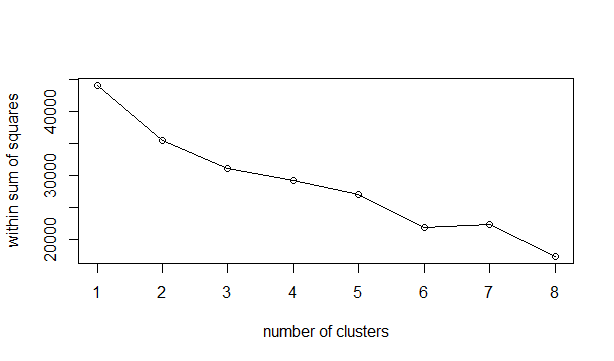
> norm\_a <- scale(a)

> View(norm\_a)

> wss=(nrow(norm\_a)-1)\*sum(apply(norm\_a,2,var))

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

> plot(1:8,wss,type = "o",xlab = "number of clusters",ylab = "within sum of squares")



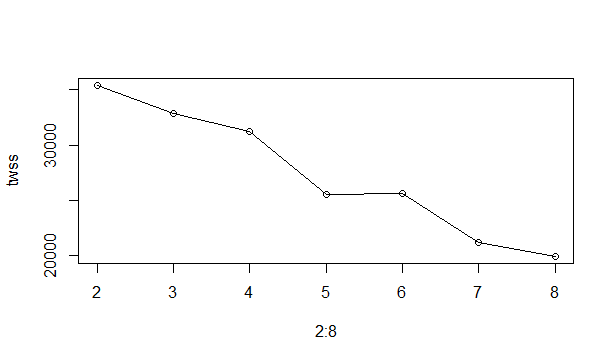
> twss <- NULL

> for(i in 2:8){ + twss <- c(twss,kmeans(norm\_a,i)$tot.withinss) + }

> twss

[1] 35400.65 32881.99 31250.32 25534.86 25671.43 21193.96 19950.31

> plot(2:8,twss,type = "o")



> #from elbow curve it is clear that k = 3 or 5

> km1 <- kmeans(norm\_a,3)

> str(km1) #twss=31001 ,bss=12977

List of 9

$ cluster : Named int [1:3999] 1 1 1 1 2 1 2 1 3 2 ...

..- attr(\*, "names")= chr [1:3999] "1" "2" "3" "4" ...

$ centers : num [1:3, 1:11] -0.29821 0.40677 1.19476 -0.05928 0.00899 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:3] "1" "2" "3"

.. ..$ : chr [1:11] "Balance" "Qual\_miles" "cc1\_miles" "cc2\_miles" ...

$ totss : num 43978

$ withinss : num [1:3] 12404 8874 9724

$ tot.withinss: num 31001

$ betweenss : num 12977

$ size : int [1:3] 2522 1285 192

$ iter : int 3

$ ifault : int 0

- attr(\*, "class")= chr "kmeans"

> km2 <- kmeans(norm\_a,5)

> str(km2) #twss=26308,bss=17670

List of 9

$ cluster : Named int [1:3999] 3 3 3 3 5 3 5 1 2 5 ...

..- attr(\*, "names")= chr [1:3999] "1" "2" "3" "4" ...

$ centers : num [1:5, 1:11] -0.154 1.218 -0.136 -0.388 0.655 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ : chr [1:5] "1" "2" "3" "4" ...

.. ..$ : chr [1:11] "Balance" "Qual\_miles" "cc1\_miles" "cc2\_miles" ...

$ totss : num 43978

$ withinss : num [1:5] 4910 4675 3925 3411 10049

$ tot.withinss: num 26969

$ betweenss : num 17009

$ size : int [1:5] 843 144 990 1185 837

$ iter : int 3

$ ifault : int 0

- attr(\*, "class")= chr "kmeans"

> #the best cluster have less totwss and high betweenss

> #hence choosing number of clusters as 5

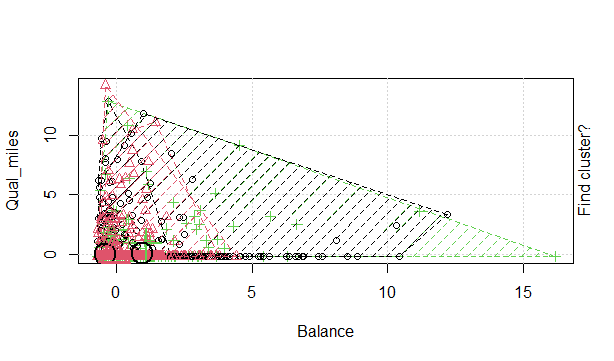
> library(animation)

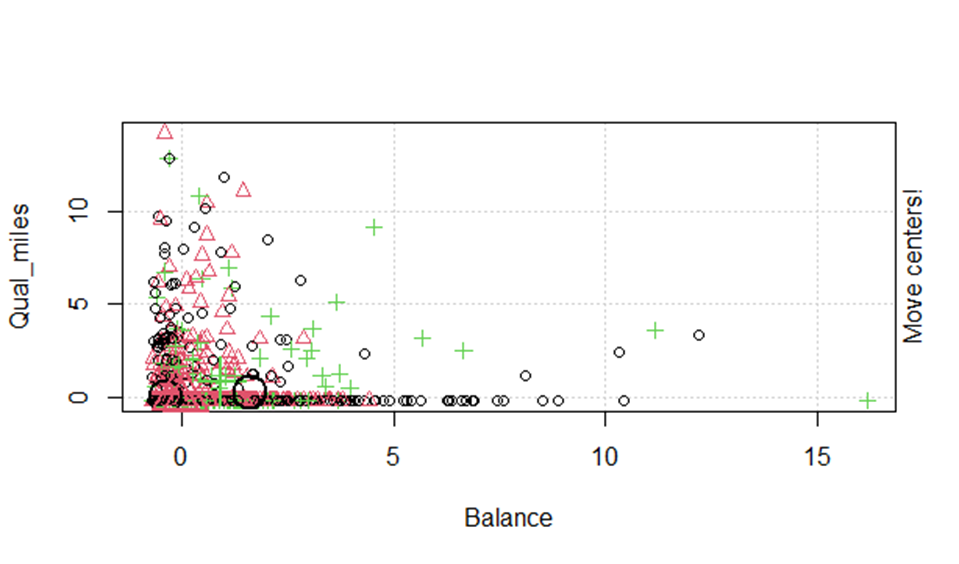
Warning message:

package ‘animation’ was built under R version 4.0.5

> km <- kmeans(norm\_a,5)

> kmanimation <- kmeans.ani(norm\_a,5)





> kmfinal <- data.frame(km$cluster,A)

> View(kmfinal)

> aggregate(kmfinal[,-c(1,2)],by=list(km$cluster),FUN = mean)

Group.1 Balance Qual\_miles cc1\_miles cc2\_miles cc3\_miles Bonus\_miles

1 1 68876.58 23.25581 1.139535 2.348837 1.000000 14689.837

2 2 216411.25 246.72316 4.573446 1.000000 1.135593 70933.031

3 3 81335.18 119.72127 3.059006 1.000000 1.000000 24646.307

4 4 162175.55 808.62000 1.993333 1.000000 1.000000 27515.640

5 5 39590.75 98.18854 1.076248 1.000000 1.000462 3210.949

Bonus\_trans Flight\_miles\_12mo Flight\_trans\_12 Days\_since\_enroll Award.

1 17.534884 582.6279 2.2093023 3968.930 0.3953488

2 22.607345 603.2203 1.9209040 5511.494 0.7401130

3 16.983696 270.8688 0.8128882 4558.483 0.5427019

4 26.180000 5607.2533 16.1800000 4537.093 0.7866667

5 5.469963 190.0199 0.5748614 3602.817 0.1779113

> View(a)

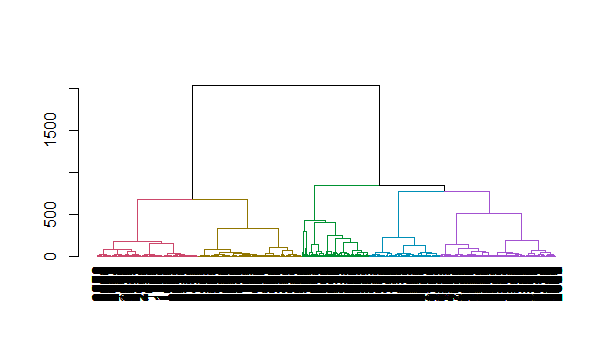
>library(dendextend)

> d <- dist(norm\_a,method = "euclidean")

> clust <- hclust(d,method = "ward.D") #choosen ward.D for better visualisation among (complete,centroid,average,ward.D,ward.D2,median)

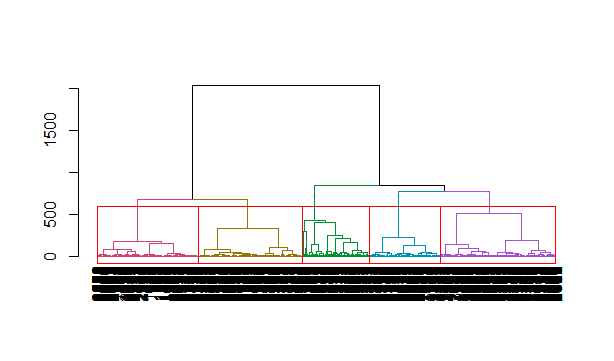
> clust\_col=color\_branches(clust,k=5)

> plot(clust\_col)



> groups <- cutree(clust,k=5)

> rect.hclust(clust,k=5,border = "red")



> cluster\_no <- as.matrix(groups)

> finaldata <- data.frame(cluster\_no,A)

> View(finaldata)

> aggregate(finaldata,by=list(cluster\_no),FUN = mean)

Group.1 cluster\_no ID. Balance Qual\_miles cc1\_miles cc2\_miles cc3\_miles

1 1 1 1389.463 46828.15 5.673839 1.029445 1.000000 1.000000

2 2 2 1847.787 161109.75 934.347530 2.540034 1.098807 1.083475

3 3 3 1461.878 96738.31 6.653693 3.665669 1.000000 1.000000

4 4 4 2115.293 50783.50 9.087379 1.593851 1.000000 1.000000

5 5 5 3271.356 33107.86 11.616062 1.295930 1.000000 1.000000

Bonus\_miles Bonus\_trans Flight\_miles\_12mo Flight\_trans\_12 Days\_since\_enroll

1 2479.873 5.383918 145.51642 0.4597961 5245.408

2 34913.136 20.597956 2279.06985 6.6149915 4462.295

3 35506.402 16.830339 129.05389 0.4071856 5126.157

4 9546.762 9.676375 255.05825 0.8252427 3955.833

5 4841.790 7.378438 95.18152 0.3146315 1801.917

Award.

1 0.0000000

2 0.6780239

3 0.4640719

4 1.0000000

5 0.0000000