Spotify Clustering

2022-12-23

# Uncomment if the code will look for 'cluster' package  
# install.packages("cluster")  
  
library("tidyverse")

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.6 v dplyr 1.0.7  
## v tidyr 1.1.4 v stringr 1.4.0  
## v readr 2.1.1 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

setwd("C:/Users/Joseph/Documents/Codes/2022/mvtec-2022/mvtec-statsprogramming/statsprog-09")  
raw <- read.csv("data/spotify.csv", sep=",");  
  
set.seed(1)  
sample <- sample\_n(raw, 1000)  
  
dd <- sample %>% select(popularity, duration\_ms, danceability,  
 energy, key, loudness,  
 speechiness, acousticness, # mode, # excluded as considered binary  
 instrumentalness, liveness, valence,  
 tempo, time\_signature)  
names(dd)

## [1] "popularity" "duration\_ms" "danceability" "energy"   
## [5] "key" "loudness" "speechiness" "acousticness"   
## [9] "instrumentalness" "liveness" "valence" "tempo"   
## [13] "time\_signature"

dim(dd)

## [1] 1000 13

summary(dd)

## popularity duration\_ms danceability energy   
## Min. : 0.00 Min. : 34878 Min. :0.0000 Min. :0.0000202   
## 1st Qu.:17.00 1st Qu.:177256 1st Qu.:0.4610 1st Qu.:0.4437500   
## Median :33.00 Median :215973 Median :0.5735 Median :0.6750000   
## Mean :32.70 Mean :227874 Mean :0.5680 Mean :0.6296990   
## 3rd Qu.:49.25 3rd Qu.:263265 3rd Qu.:0.6943 3rd Qu.:0.8482500   
## Max. :85.00 Max. :751106 Max. :0.9570 Max. :0.9990000   
## key loudness speechiness acousticness   
## Min. : 0.000 Min. :-35.644 Min. :0.00000 Min. :0.0000014   
## 1st Qu.: 2.000 1st Qu.:-10.308 1st Qu.:0.03600 1st Qu.:0.0183500   
## Median : 5.000 Median : -7.176 Median :0.04930 Median :0.1920000   
## Mean : 5.258 Mean : -8.411 Mean :0.08453 Mean :0.3332612   
## 3rd Qu.: 8.000 3rd Qu.: -5.119 3rd Qu.:0.08682 3rd Qu.:0.6377500   
## Max. :11.000 Max. : 0.698 Max. :0.95200 Max. :0.9960000   
## instrumentalness liveness valence tempo   
## Min. :0.0000000 Min. :0.0133 Min. :0.0000 Min. : 0.00   
## 1st Qu.:0.0000000 1st Qu.:0.0979 1st Qu.:0.2530 1st Qu.: 99.97   
## Median :0.0000403 Median :0.1305 Median :0.4490 Median :122.52   
## Mean :0.1529858 Mean :0.2078 Mean :0.4708 Mean :123.00   
## 3rd Qu.:0.0428500 3rd Qu.:0.2625 3rd Qu.:0.6843 3rd Qu.:141.90   
## Max. :0.9910000 Max. :0.9900 Max. :0.9840 Max. :207.48   
## time\_signature   
## Min. :0.000   
## 1st Qu.:4.000   
## Median :4.000   
## Mean :3.892   
## 3rd Qu.:4.000   
## Max. :5.000

attach(dd)

# Set a list of numerical variables  
names(dd)

## [1] "popularity" "duration\_ms" "danceability" "energy"   
## [5] "key" "loudness" "speechiness" "acousticness"   
## [9] "instrumentalness" "liveness" "valence" "tempo"   
## [13] "time\_signature"

dcon <- dd  
dim(dcon)

## [1] 1000 13

# CLUSTERING

## KMEANS RUN, BUT HOW MANY CLASSES?

k1 <- kmeans(dcon,5)  
names(dcon)

## [1] "popularity" "duration\_ms" "danceability" "energy"   
## [5] "key" "loudness" "speechiness" "acousticness"   
## [9] "instrumentalness" "liveness" "valence" "tempo"   
## [13] "time\_signature"

print(k1)

## K-means clustering with 5 clusters of sizes 156, 402, 27, 303, 112  
##   
## Cluster means:  
## popularity duration\_ms danceability energy key loudness speechiness  
## 1 31.53846 120402.8 0.5657103 0.5233124 5.032051 -10.877936 0.12285577  
## 2 32.72886 193152.2 0.5820398 0.6386258 5.199005 -8.014968 0.08335249  
## 3 28.55556 531501.6 0.5397407 0.6353593 5.925926 -10.395778 0.11681852  
## 4 34.39934 256385.3 0.5564323 0.6625330 5.343234 -7.311119 0.06239406  
## 5 30.66071 351860.6 0.5588750 0.6556473 5.392857 -8.897098 0.08750893  
## acousticness instrumentalness liveness valence tempo time\_signature  
## 1 0.4953245 0.24465698 0.2156103 0.5266917 117.6768 3.730769  
## 2 0.3412623 0.10342647 0.2002463 0.4921473 124.2408 3.920398  
## 3 0.2399277 0.41255923 0.2535222 0.3619889 121.4872 3.888889  
## 4 0.2825102 0.09326735 0.2055188 0.4552974 123.8637 3.924092  
## 5 0.2386117 0.30216777 0.2195339 0.3846679 123.9692 3.928571  
##   
## Clustering vector:  
## [1] 3 1 2 1 2 4 2 2 2 2 2 4 4 1 2 2 2 5 1 4 2 2 2 2 2 4 1 1 2 4 2 4 2 4 5 4 2  
## [38] 3 1 1 1 4 3 2 2 2 2 2 1 4 2 5 4 5 2 2 5 1 5 1 2 1 2 5 4 1 4 4 1 5 1 4 4 4  
## [75] 4 2 4 2 2 4 4 4 2 2 2 4 2 1 4 2 5 4 5 4 2 4 2 4 4 4 2 2 2 5 2 2 4 1 4 3 4  
## [112] 4 4 4 2 4 5 2 5 3 1 1 2 4 5 4 2 5 2 5 5 2 1 4 4 5 2 2 5 4 4 4 2 4 2 4 4 2  
## [149] 5 4 2 2 5 5 2 2 1 2 1 4 5 3 2 2 5 4 2 2 1 2 4 2 1 2 2 4 4 5 2 4 2 2 2 2 4  
## [186] 2 2 4 1 2 2 1 2 1 1 2 1 1 1 1 2 1 2 5 4 4 5 1 1 4 1 2 2 2 2 2 2 4 2 1 2 2  
## [223] 4 4 5 4 4 2 2 1 4 2 4 5 5 2 2 2 2 5 2 2 2 4 2 4 5 1 1 2 1 5 3 2 1 4 2 4 5  
## [260] 1 2 2 2 4 2 2 5 2 4 2 1 2 1 5 4 2 2 2 3 4 2 2 1 2 2 4 4 2 2 2 4 4 5 2 2 4  
## [297] 4 5 3 4 2 2 5 2 1 2 4 4 5 2 1 2 1 1 2 2 2 5 2 2 2 5 2 4 2 2 2 2 4 5 2 5 5  
## [334] 5 2 4 5 1 4 2 2 4 2 2 2 4 4 2 3 2 4 4 2 2 2 4 2 4 2 2 2 1 2 2 2 2 2 4 1 2  
## [371] 2 2 2 5 2 2 1 2 4 4 1 2 4 4 2 5 1 1 4 2 4 2 4 4 4 2 4 4 4 2 3 2 1 2 2 1 4  
## [408] 5 2 1 4 4 2 2 2 4 1 2 2 2 3 2 1 2 1 5 1 1 4 1 4 2 4 1 2 4 4 2 5 4 2 2 4 4  
## [445] 2 2 2 4 2 1 5 1 2 1 2 4 4 5 1 4 1 4 4 3 2 5 4 1 2 2 2 4 5 4 2 2 2 4 1 3 4  
## [482] 2 2 1 2 1 1 2 2 2 2 1 2 4 2 4 1 2 4 1 4 4 3 2 4 2 4 2 2 2 1 4 5 4 4 2 4 2  
## [519] 2 4 1 4 5 5 2 4 1 2 2 5 1 5 2 2 2 2 1 1 4 4 2 4 4 1 2 4 4 4 4 2 4 1 4 2 2  
## [556] 2 4 2 4 2 2 5 4 2 2 2 4 1 4 4 1 1 5 4 3 4 4 2 4 2 5 2 1 5 3 5 2 5 2 4 1 2  
## [593] 2 2 2 2 2 4 2 4 2 4 5 5 2 2 4 2 4 4 1 4 1 1 2 1 4 2 4 2 2 4 4 1 1 2 2 4 4  
## [630] 4 2 2 3 2 4 2 5 2 5 2 2 2 4 4 5 5 3 4 4 5 2 4 4 2 4 4 2 3 1 2 1 4 2 3 4 5  
## [667] 4 4 2 5 5 1 4 3 2 4 1 4 2 2 5 1 1 5 5 2 2 5 4 2 2 4 5 2 5 2 1 4 4 5 4 5 2  
## [704] 4 4 5 2 2 2 4 5 5 5 2 2 4 2 4 1 2 2 5 2 1 2 4 4 2 1 4 2 2 4 1 2 1 2 4 4 2  
## [741] 4 4 4 1 4 1 4 2 3 4 2 1 2 2 2 1 4 2 4 2 4 1 1 4 2 4 4 1 2 1 2 2 2 2 4 4 4  
## [778] 5 1 4 4 2 2 4 1 4 4 1 2 2 2 1 2 4 4 2 1 4 5 5 4 2 2 1 4 4 2 4 2 4 2 2 4 2  
## [815] 5 4 2 1 5 1 4 1 2 2 4 1 4 4 4 5 2 2 4 2 4 4 4 2 2 1 4 4 1 2 4 1 2 4 2 2 4  
## [852] 4 4 2 1 1 2 4 2 2 2 5 2 2 4 5 5 2 3 1 4 2 2 1 2 4 2 1 4 2 4 4 5 1 2 1 4 4  
## [889] 1 2 4 4 5 5 2 5 2 2 2 2 5 2 4 4 1 2 2 4 2 1 4 1 2 2 4 3 2 5 4 2 4 2 4 4 4  
## [926] 2 2 2 4 1 4 5 2 2 4 4 5 4 2 4 4 4 2 1 4 2 2 2 4 2 2 1 1 4 2 3 3 4 4 4 1 2  
## [963] 4 1 2 5 2 5 1 4 2 4 4 2 4 5 2 2 4 1 1 2 4 2 5 4 1 4 2 2 1 2 1 2 2 2 2 2 4  
## [1000] 5  
##   
## Within cluster sum of squares by cluster:  
## [1] 133403243646 148004709168 179069333429 144634467283 138183120609  
## (between\_SS / total\_SS = 90.1 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

attributes(k1)

## $names  
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"   
##   
## $class  
## [1] "kmeans"

k1$size

## [1] 156 402 27 303 112

k1$withinss

## [1] 133403243646 148004709168 179069333429 144634467283 138183120609

k1$centers

## popularity duration\_ms danceability energy key loudness speechiness  
## 1 31.53846 120402.8 0.5657103 0.5233124 5.032051 -10.877936 0.12285577  
## 2 32.72886 193152.2 0.5820398 0.6386258 5.199005 -8.014968 0.08335249  
## 3 28.55556 531501.6 0.5397407 0.6353593 5.925926 -10.395778 0.11681852  
## 4 34.39934 256385.3 0.5564323 0.6625330 5.343234 -7.311119 0.06239406  
## 5 30.66071 351860.6 0.5588750 0.6556473 5.392857 -8.897098 0.08750893  
## acousticness instrumentalness liveness valence tempo time\_signature  
## 1 0.4953245 0.24465698 0.2156103 0.5266917 117.6768 3.730769  
## 2 0.3412623 0.10342647 0.2002463 0.4921473 124.2408 3.920398  
## 3 0.2399277 0.41255923 0.2535222 0.3619889 121.4872 3.888889  
## 4 0.2825102 0.09326735 0.2055188 0.4552974 123.8637 3.924092  
## 5 0.2386117 0.30216777 0.2195339 0.3846679 123.9692 3.928571

## LETS COMPUTE THE DECOMPOSITION OF INERTIA

Bss <- sum(rowSums(k1$centers^2)\*k1$size)  
Bss

## [1] 5.867007e+13

Wss <- sum(k1$withinss)  
Wss

## [1] 743294874136

Tss <- k1$totss  
Tss

## [1] 7.486924e+12

Bss+Wss

## [1] 5.941336e+13

Ib1 <- 100\*Bss/(Bss+Wss)  
Ib1

## [1] 98.74894

# LETS REPEAT THE KMEANS RUN WITH K=5  
  
k2 <- kmeans(dcon,5)  
k2$size

## [1] 156 112 303 402 27

Bss <- sum(rowSums(k2$centers^2)\*k2$size)  
Bss

## [1] 5.867007e+13

Wss <- sum(k2$withinss)  
Wss

## [1] 743294874136

Ib2 <- 100\*Bss/(Bss+Wss)  
Ib2

## [1] 98.74894

Ib1

## [1] 98.74894

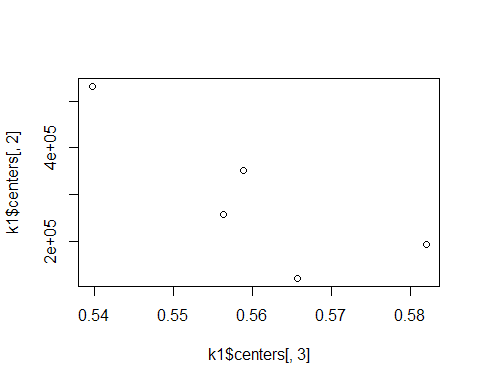
k2$centers

## popularity duration\_ms danceability energy key loudness speechiness  
## 1 31.53846 120402.8 0.5657103 0.5233124 5.032051 -10.877936 0.12285577  
## 2 30.66071 351860.6 0.5588750 0.6556473 5.392857 -8.897098 0.08750893  
## 3 34.39934 256385.3 0.5564323 0.6625330 5.343234 -7.311119 0.06239406  
## 4 32.72886 193152.2 0.5820398 0.6386258 5.199005 -8.014968 0.08335249  
## 5 28.55556 531501.6 0.5397407 0.6353593 5.925926 -10.395778 0.11681852  
## acousticness instrumentalness liveness valence tempo time\_signature  
## 1 0.4953245 0.24465698 0.2156103 0.5266917 117.6768 3.730769  
## 2 0.2386117 0.30216777 0.2195339 0.3846679 123.9692 3.928571  
## 3 0.2825102 0.09326735 0.2055188 0.4552974 123.8637 3.924092  
## 4 0.3412623 0.10342647 0.2002463 0.4921473 124.2408 3.920398  
## 5 0.2399277 0.41255923 0.2535222 0.3619889 121.4872 3.888889

k1$centers

## popularity duration\_ms danceability energy key loudness speechiness  
## 1 31.53846 120402.8 0.5657103 0.5233124 5.032051 -10.877936 0.12285577  
## 2 32.72886 193152.2 0.5820398 0.6386258 5.199005 -8.014968 0.08335249  
## 3 28.55556 531501.6 0.5397407 0.6353593 5.925926 -10.395778 0.11681852  
## 4 34.39934 256385.3 0.5564323 0.6625330 5.343234 -7.311119 0.06239406  
## 5 30.66071 351860.6 0.5588750 0.6556473 5.392857 -8.897098 0.08750893  
## acousticness instrumentalness liveness valence tempo time\_signature  
## 1 0.4953245 0.24465698 0.2156103 0.5266917 117.6768 3.730769  
## 2 0.3412623 0.10342647 0.2002463 0.4921473 124.2408 3.920398  
## 3 0.2399277 0.41255923 0.2535222 0.3619889 121.4872 3.888889  
## 4 0.2825102 0.09326735 0.2055188 0.4552974 123.8637 3.924092  
## 5 0.2386117 0.30216777 0.2195339 0.3846679 123.9692 3.928571

plot(k1$centers[,3],k1$centers[,2])



table(k1$cluster, k2$cluster)

##   
## 1 2 3 4 5  
## 1 156 0 0 0 0  
## 2 0 0 0 402 0  
## 3 0 0 0 0 27  
## 4 0 0 303 0 0  
## 5 0 112 0 0 0

## WHY WE HAVE OBTAINED DIFFERENT RESULTS?, AND WHICH RUN IS BETTER?

### NOW TRY K=8

k3 <- kmeans(dcon,8)  
k3$size

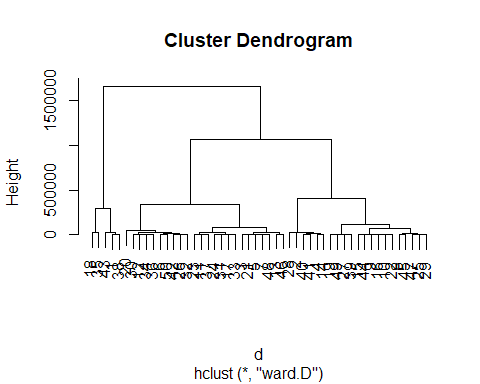
## [1] 167 52 241 15 42 163 235 85

Bss <- sum(rowSums(k3$centers^2)\*k3$size)  
Wss <- sum(k3$withinss)  
  
Ib3 <- 100\*Bss/(Bss+Wss)  
Ib3

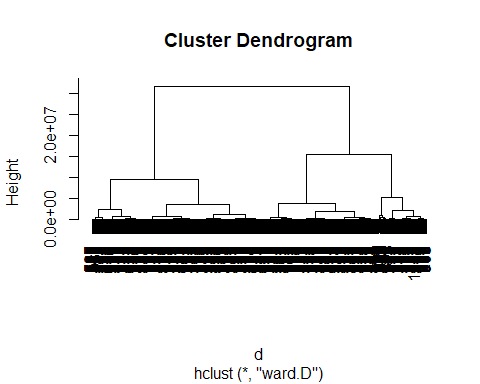
## [1] 99.41543

## HIERARCHICAL CLUSTERING

d <- dist(dcon[1:50,])  
h1 <- hclust(d,method="ward.D") # NOTICE THE COST  
plot(h1)



d <- dist(dcon)  
# h1 <- hclust(d,method="ward") # NOTICE THE COST  
# The "ward" method has been renamed to "ward.D"; note new "ward.D2"  
h1 <- hclust(d,method="ward.D") # NOTICE THE COST  
plot(h1)



## BUT WE ONLY NEED WHERE THERE ARE THE LEAPS OF THE HEIGHT

## WHERE ARE THER THE LEAPS? WHERE WILL YOU CUT THE DENDREOGRAM?, HOW MANY CLASSES WILL YOU OBTAIN?

nc = 3  
  
c1 <- cutree(h1,nc)  
  
c1[1:20]

## [1] 1 2 2 2 2 3 2 2 2 2 2 3 3 2 2 2 2 1 2 3

nc = 5  
  
c5 <- cutree(h1,nc)  
  
c5[1:20]

## [1] 1 2 3 2 3 4 3 3 3 3 3 4 4 2 3 3 3 5 2 4

table(c1)

## c1  
## 1 2 3   
## 142 524 334

table(c5)

## c5  
## 1 2 3 4 5   
## 27 152 372 334 115

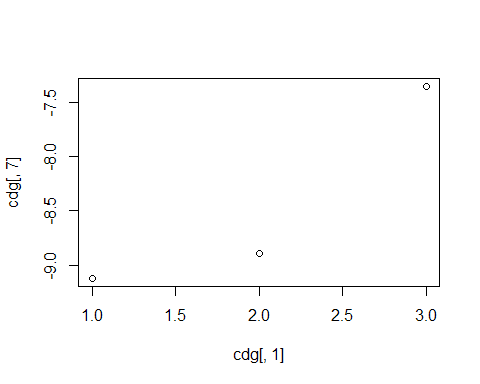
table(c1,c5)

## c5  
## c1 1 2 3 4 5  
## 1 27 0 0 0 115  
## 2 0 152 372 0 0  
## 3 0 0 0 334 0

cdg <- aggregate(as.data.frame(dcon),list(c1),mean)  
cdg

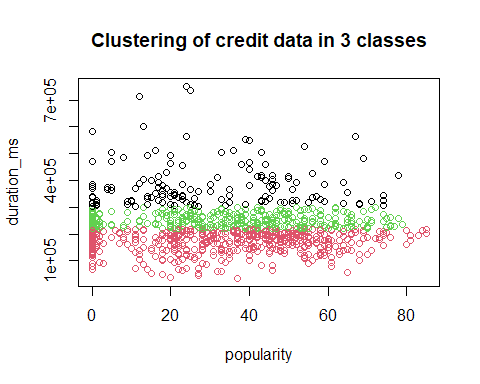
## Group.1 popularity duration\_ms danceability energy key loudness  
## 1 1 29.88028 384994.1 0.5547465 0.6566282 5.500000 -9.121606  
## 2 2 32.04198 169628.5 0.5778660 0.6006756 5.169847 -8.893502  
## 3 3 34.94611 252452.7 0.5581467 0.6637838 5.293413 -7.353120  
## speechiness acousticness instrumentalness liveness valence tempo  
## 1 0.09197183 0.2363933 0.31838352 0.2245951 0.3891373 123.7114  
## 2 0.09529542 0.3892775 0.14205565 0.2056000 0.5046720 122.2489  
## 3 0.06448772 0.2865627 0.09981498 0.2042287 0.4524344 123.8692  
## time\_signature  
## 1 3.922535  
## 2 3.864504  
## 3 3.922156

plot(cdg[,1], cdg[,7])



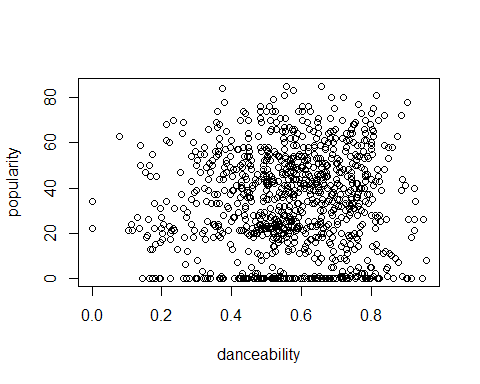
## LETS SEE THE PARTITION VISUALLY

plot(popularity ,duration\_ms ,col=c1,main="Clustering of credit data in 3 classes")

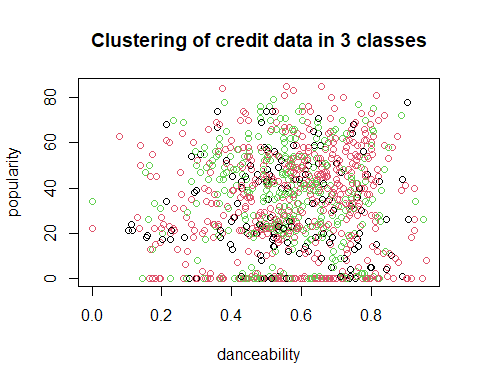


# Commented out the legend below and the consecutive temporarily  
# legend("topright",c("class1","class2","class3"),pch=1,col=c(1:3))

plot(danceability, popularity)

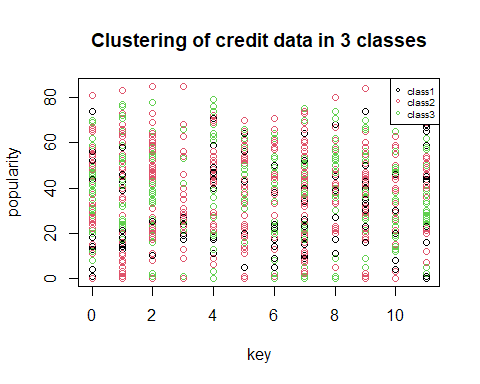


plot(danceability, popularity,col=c1,main="Clustering of credit data in 3 classes")

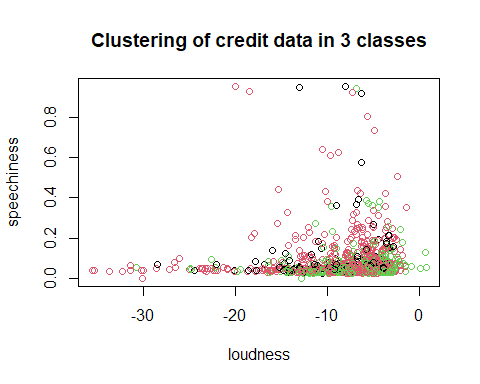


# legend("topright",c("class1","class2","class3"),pch=1,col=c(1:3), cex=0.6)

plot(key, popularity, col=c1,main="Clustering of credit data in 3 classes")  
legend("topright",c("class1","class2","class3"),pch=1,col=c(1:3), cex=0.6)

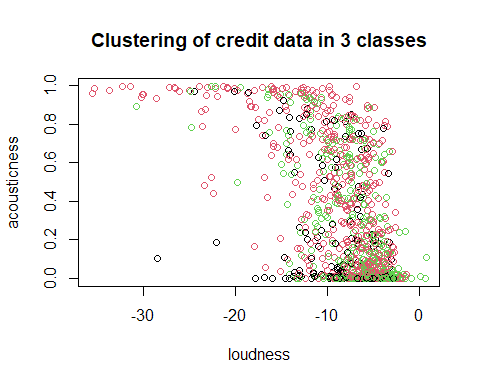


plot(loudness, speechiness,col=c1,main="Clustering of credit data in 3 classes")



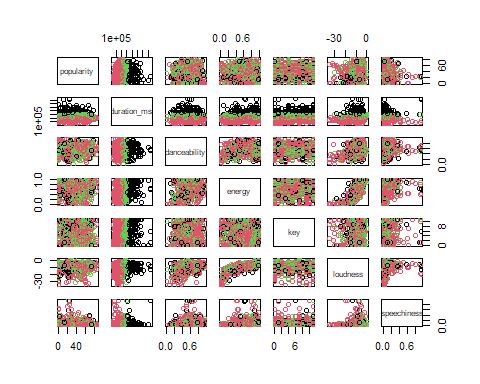
# legend("topright",c("class1","class2","class3"),pch=1,col=c(1:3), cex=0.6)

plot(loudness, acousticness,col=c1,main="Clustering of credit data in 3 classes")



# legend("topright",c("class1","class2","class3"),pch=1,col=c(1:3), cex=0.6)

pairs(dcon[,1:7], col=c1)



#plot(FI[,1],FI[,2],col=c1,main="Clustering of credit data in 3 classes")  
#legend("topleft",c("c1","c2","c3"),pch=1,col=c(1:3))

## LETS SEE THE QUALITY OF THE HIERARCHICAL PARTITION

Bss <- sum(rowSums(cdg^2)\*as.numeric(table(c1)))  
  
Ib4 <- 100\*Bss/Tss  
Ib4

## [1] 766.8226

#move to Gower mixed distance to deal   
#simoultaneously with numerical and qualitative data  
  
library(cluster)

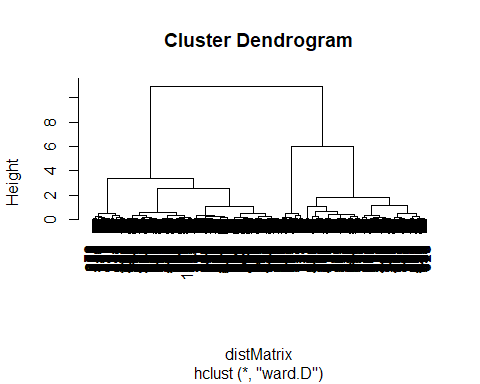
## Warning: package 'cluster' was built under R version 4.1.3

## Dissimilarity matrix (Main task)

dissimMatrix <- daisy(dd, metric = "gower", stand=TRUE)   
  
distMatrix<-dissimMatrix^2

## INSTRUCTION THAT RUN THE HEIRARCHICAL CLUSTERING

h1 <- hclust(distMatrix,method="ward.D") # NOTICE THE COST  
plot(h1)



## CUT THE TREE

c2 <- cutree(h1, 4)  
  
#class sizes   
table(c2)

## c2  
## 1 2 3 4   
## 104 361 466 69

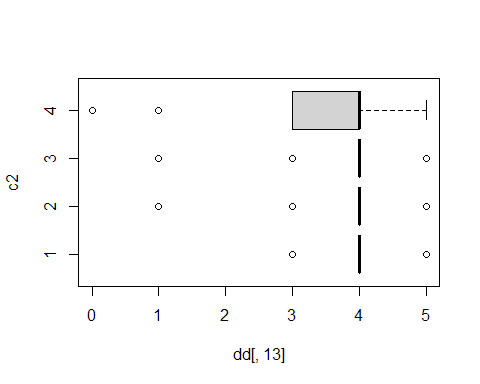
#comparing with other partitions  
table(c1,c2)

## c2  
## c1 1 2 3 4  
## 1 40 37 54 11  
## 2 34 211 232 47  
## 3 30 113 180 11

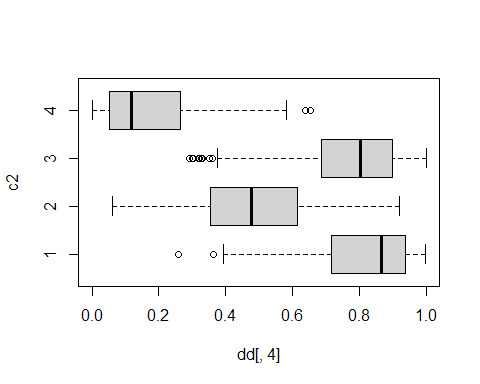
names(dd)

## [1] "popularity" "duration\_ms" "danceability" "energy"   
## [5] "key" "loudness" "speechiness" "acousticness"   
## [9] "instrumentalness" "liveness" "valence" "tempo"   
## [13] "time\_signature"

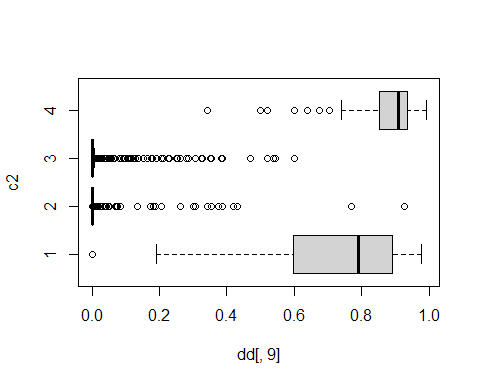
# time\_signature  
boxplot(dd[,13]~c2, horizontal=TRUE)



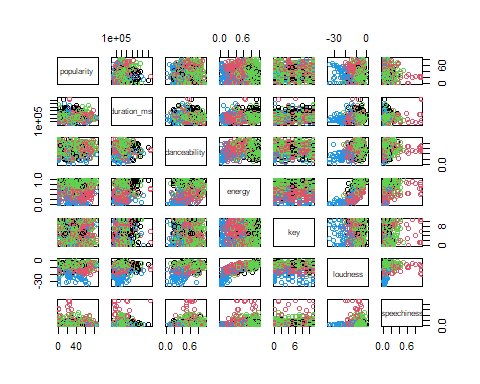
# energy  
boxplot(dd[,4]~c2, horizontal=TRUE)



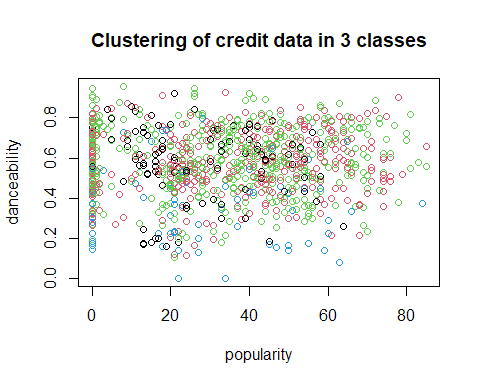
# instrumentalness  
boxplot(dd[,9]~c2, horizontal=TRUE)



pairs(dcon[,1:7], col=c2)



plot(popularity,danceability,col=c2,main="Clustering of credit data in 3 classes")

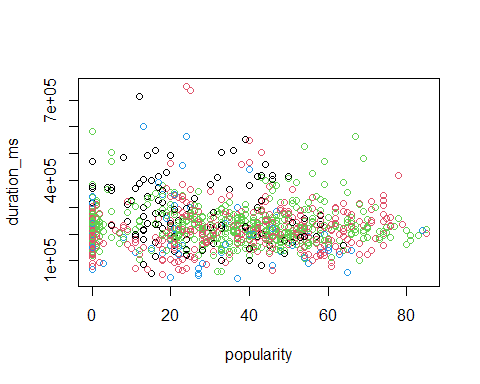


# legend("topright",levels(c2),pch=1,col=c(1:4), cex=0.6)

cdg <- aggregate(as.data.frame(dcon),list(c2),mean)  
cdg

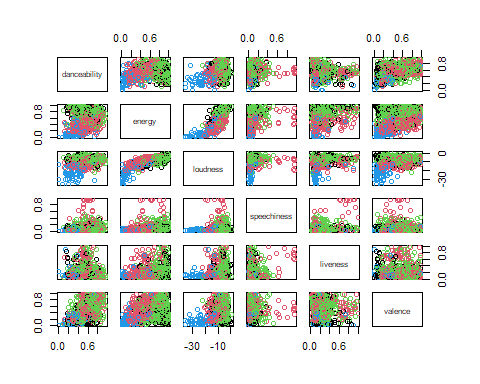
## Group.1 popularity duration\_ms danceability energy key loudness  
## 1 1 27.45192 281015.6 0.5718269 0.8022788 5.288462 -7.488192  
## 2 2 33.85042 215732.5 0.5738698 0.4822856 5.240997 -9.417305  
## 3 3 33.66524 229490.0 0.5942339 0.7712532 5.324034 -6.089751  
## 4 4 28.14493 200381.4 0.3543014 0.1848251 4.855072 -20.219739  
## speechiness acousticness instrumentalness liveness valence tempo  
## 1 0.07446154 0.03611830 0.71950964 0.2310010 0.3731615 128.3631  
## 2 0.09186150 0.62660080 0.01892382 0.2098496 0.4786626 117.3942  
## 3 0.08655215 0.09562238 0.02527967 0.2102099 0.5272509 128.6594  
## 4 0.04774493 0.85133333 0.86297101 0.1464014 0.1958523 105.9920  
## time\_signature  
## 1 3.990385  
## 2 3.839335  
## 3 3.957082  
## 4 3.579710

# plot(Edad, Gastos, col= c2)  
plot(popularity, duration\_ms, col= c2)



# points(cdg[,4],cdg[,5],pch=16,col="orange")  
# text(cdg[,4],cdg[,5], labels=cdg[,1], pos=2, font=2, cex=0.7, col="orange")

potencials<-c(3,4,6,7,10,11)  
pairs(dcon[,potencials],col=c2)



# Profiling plots