19201513-Akanseoluwa-Adegoke-SML-Assignment-1-copy.R

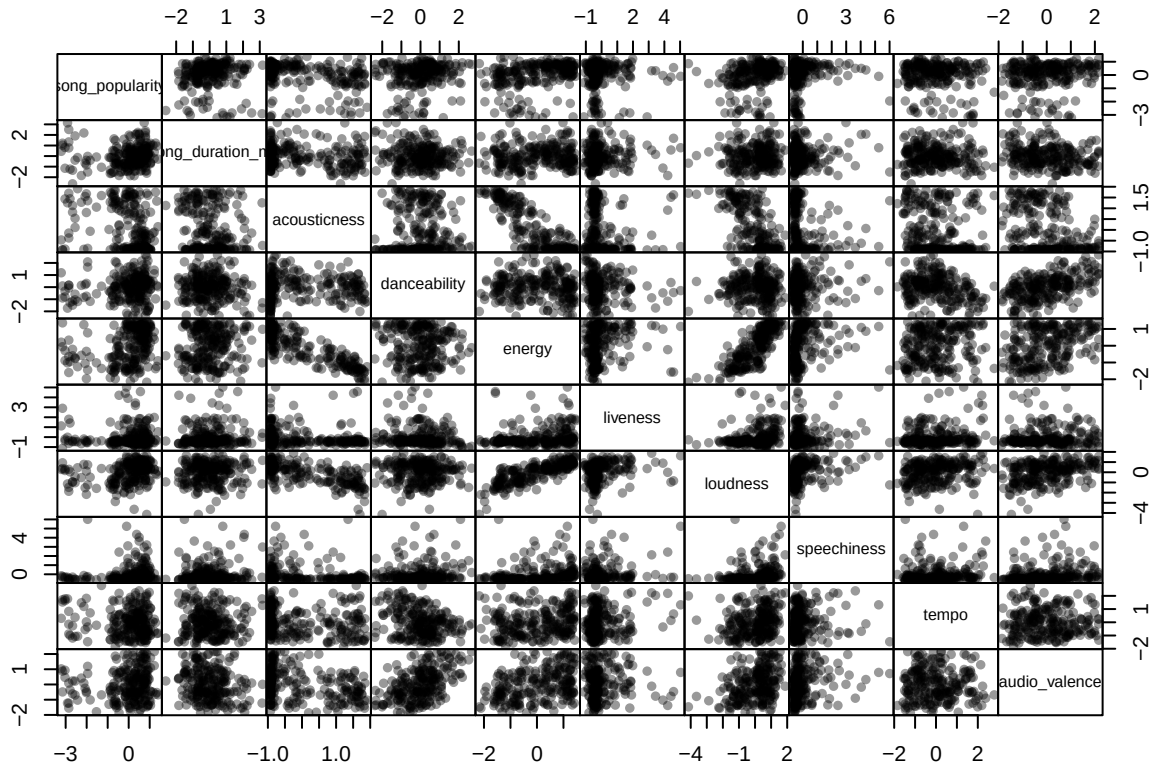
Akanseoluwa

2020-08-19

```
# Akanseoluwa Stephen Adegoke
# Master's Student, Data and Computational Science
# University College Dublin
# © 2020. Akanseoluwa Adegoke.


# Task #
# - Complete a cluster analysis of the Spotify audio features data

# Experience
# - Unsupervised Learning

# Performance Measure
# - Internal Validation : Calinski Harabasz Index
#                       : Silhouete
# - External Validation : - Rand Index


#Reading Spotify Songs data
spotify = read.csv("data_spotify_songs.csv")

#Viewing first parts of the dataset
head(spotify)
```

```
##   genre         song_name               artist song_popularity
## 1  rock        In The End          Linkin Park              66
## 2  rock Seven Nation Army     The White Stripes              76
## 3  rock        By The Way Red Hot Chili Peppers              74
## 4  rock How You Remind Me           Nickelback              56
## 5  rock  Bring Me To Life          Evanescence              80
## 6  rock       Last Resort           Papa Roach              81
##   song_duration_ms acousticness danceability energy liveness loudness
## 1           216933     0.010300        0.542  0.853    0.108   -6.407
## 2           231733     0.008170        0.737  0.463    0.255   -7.828
## 3           216933     0.026400        0.451  0.970    0.102   -4.938
## 4           223826     0.000954        0.447  0.766    0.113   -5.065
## 5           235893     0.008950        0.316  0.945    0.396   -3.169
## 6           199893     0.000504        0.581  0.887    0.268   -3.659
##   speechiness   tempo audio_valence
## 1      0.0498 105.256         0.370
## 2      0.0792 123.881         0.324
## 3      0.1070 122.444         0.198
## 4      0.0313 172.011         0.574
## 5      0.1240 189.931         0.320
## 6      0.0624  90.578         0.724
```

```
# Removing categorical data from our data
spotify_new = spotify[, -1 : -3]

# Scaling our dataset
spotify_new = scale(spotify_new)

# Visualising our data
pairs(spotify_new, gap = 0, pch = 16, col=adjustcolor(1,0.4))
```

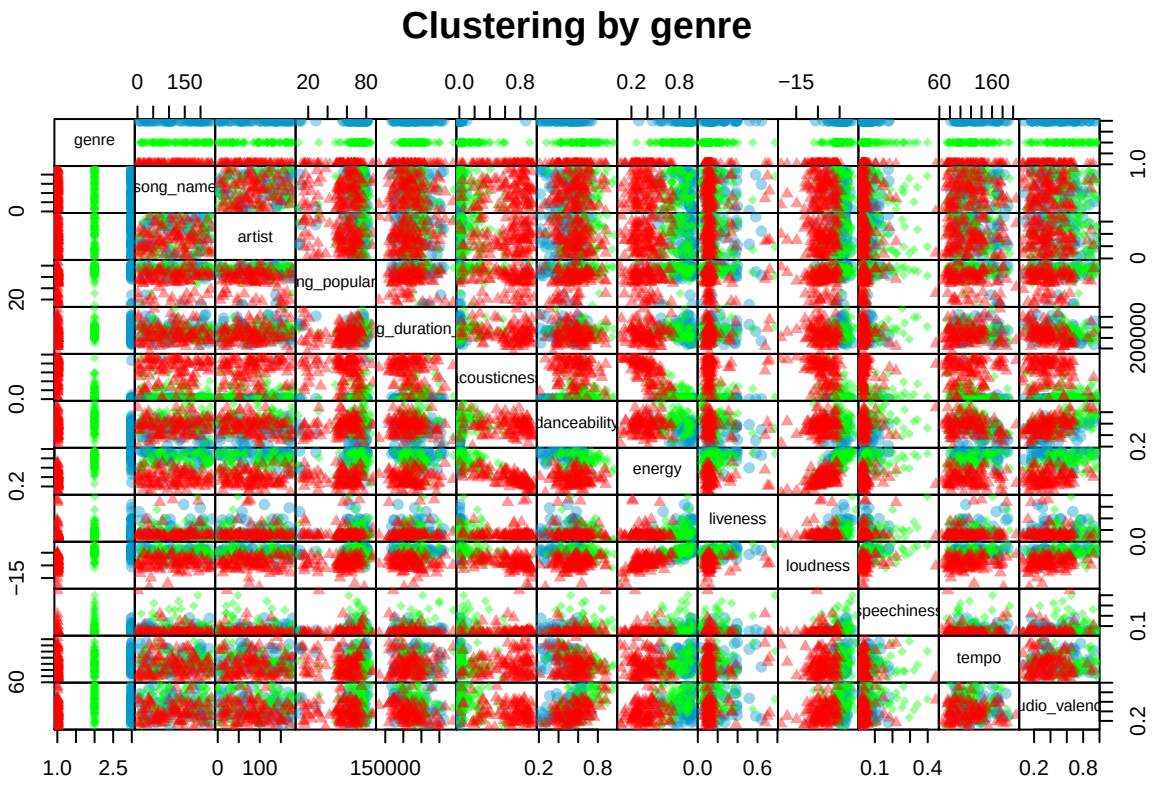



```
#K-means Clustering wth 3 clusters
fitkm = kmeans(spotify_new, centers = 3, nstart = 20)
fitkm
```

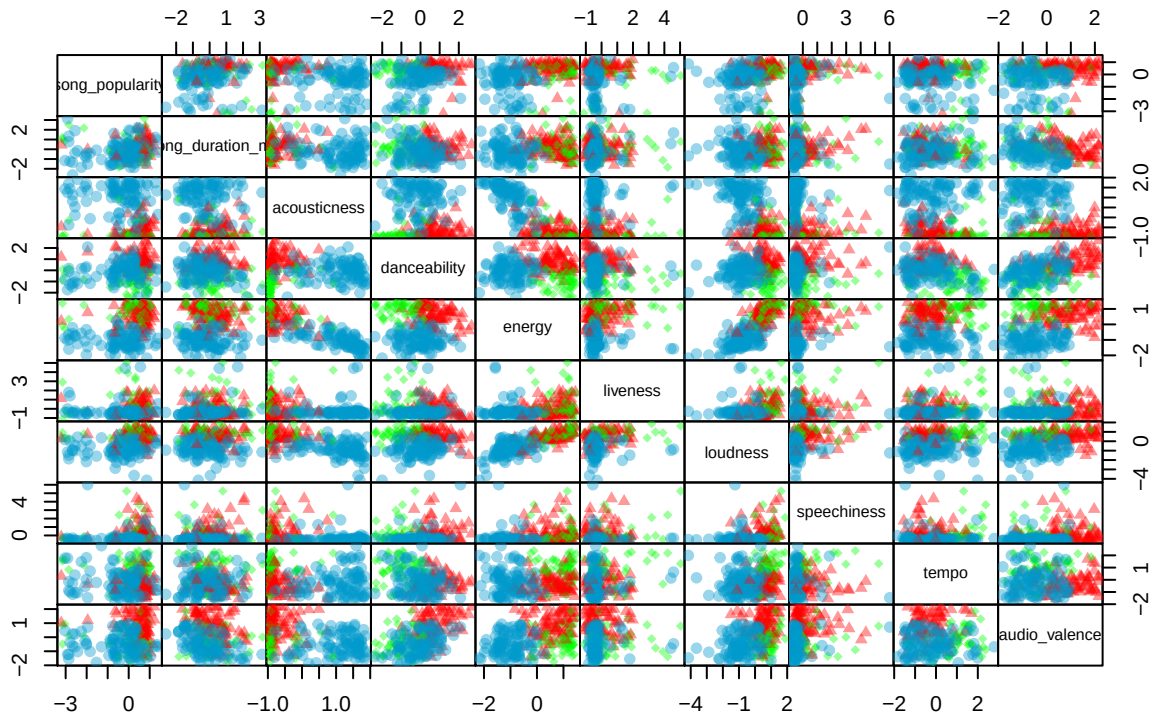
```
## K-means clustering with 3 clusters of sizes 85, 58, 96
##
## Cluster means:
##   song_popularity song_duration_ms acousticness danceability    energy
## 1      0.46253591        0.1399628   -0.6632780   0.73828817  0.574176
## 2      0.07828932        0.1595962   -0.7678231  -0.91719112  0.866714
## 3     -0.45683680       -0.2203481    1.0511705  -0.09955636 -1.032025
##      liveness   loudness speechiness      tempo audio_valence
## 1 -0.05483045  0.4725582   0.2393707 -0.3146682     0.7845552
## 2  0.47520390  0.7545638   0.2191265  0.8254328    -0.2792268
## 3 -0.23855456 -0.8742932  -0.3443318 -0.2200865    -0.5259587
##
```

```
## Clustering vector:
##   [1] 1 1 2 2 2 1 1 2 2 1 2 1 1 2 2 1 2 2 1 2 2 2 1 1 3 2 2 2 1 2 1 2 1 1 1
##  [36] 2 2 1 2 1 2 2 1 2 1 2 2 1 1 2 1 2 2 2 2 2 2 2 2 1 1 1 1 1 1 2 1 1 1 1
##  [71] 1 3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 2 1 1 2 1 1 1 1 1 2 1 1 1 1
## [106] 1 1 3 2 1 1 1 1 2 1 1 2 2 1 2 1 2 2 1 1 3 1 2 1 2 1 2 2 1 2 2 2 1 3 3
## [141] 3 3 3 1 3 3 3 3 3 2 1 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [176] 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3
## [211] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 1 3 3 3 3 1 3 3 3 3
##
## Within cluster sum of squares by cluster:
## [1] 462.4538 441.5809 682.7856
##  (between_SS / total_SS =  33.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
# split the plot window in 2 screens
par( mfrow = c(1,2) )

#Setting different shapes for the plot
symb <- c(17, 18, 19)

#Setting different colours for the clusters
col = c("red", "green", "deepskyblue3")


#Clustering according to Music Genre
pairs(spotify, gap = 0, pch = symb[spotify$genre],
      col= adjustcolor(col[spotify$genre], 0.4), main = "Clustering by genre")
```

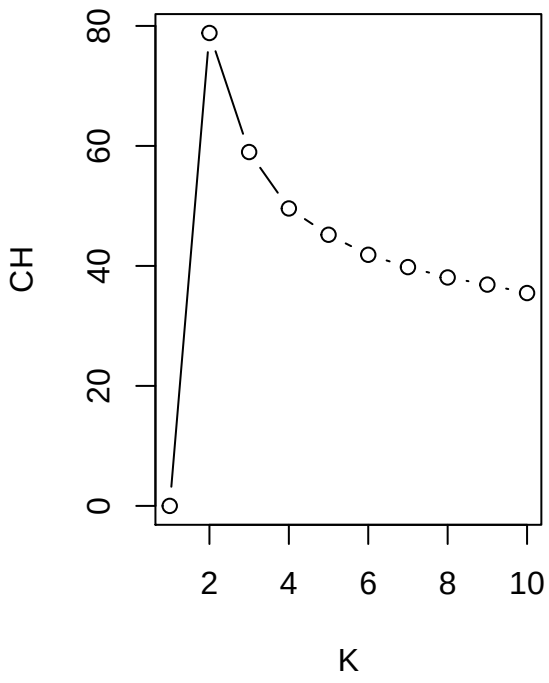



```
# Plot of Clustering Solution with Three(3) clusters
pairs(spotify_new, gap = 0, pch = symb[fitkm$cluster],
      col=adjustcolor(col[fitkm$cluster], 0.4), main = "Clustering of three components")
```

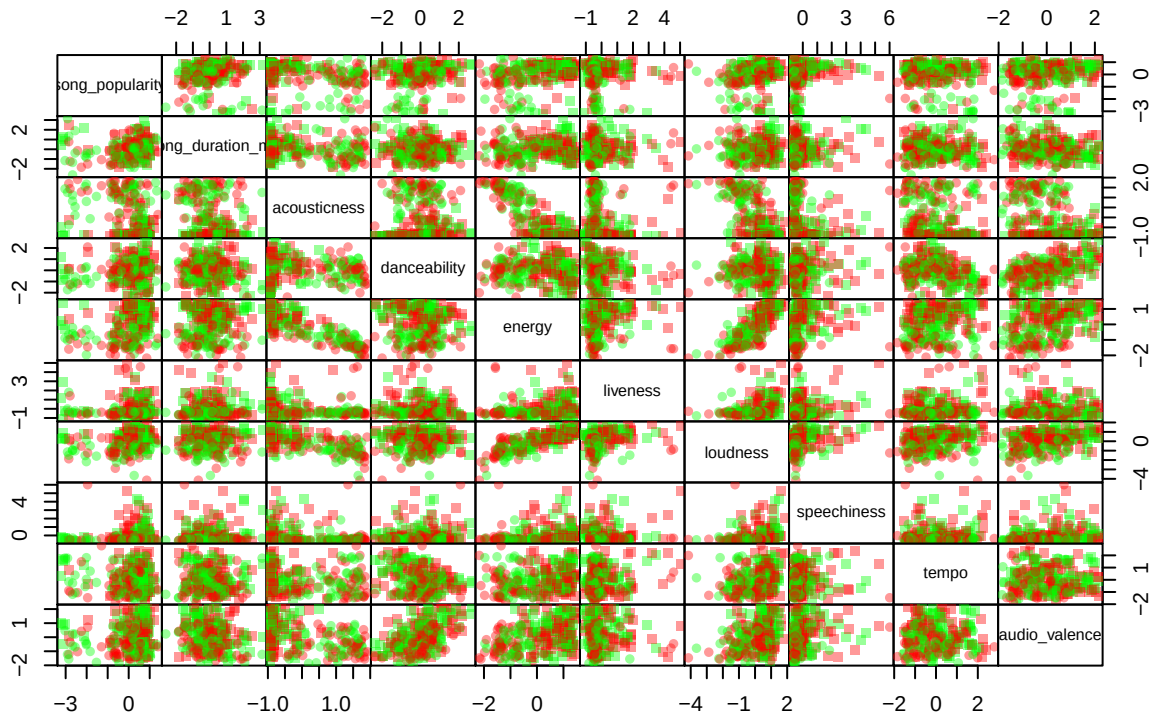



```
######## CLUSTERING VALIDATION  ##########

## INTERNAL VALIDATION #
# Calinski-Harabasz Index #

#Setting our maximum value of K
K = 10

#Initialise Empty Vector
wss = bss = rep(NA, K)


for ( k in 1:K ) {
  # run kmeans for each value of k
  fit <- kmeans(spotify_new, centers = k, nstart = 50)
  wss[k] <- fit$tot.withinss
  # store total within sum of squares
  bss[k] <- fit$betweenss

}

# Computing Calinski-Harabasz Index
N <- nrow(spotify_new)
ch <- ( bss/(1:K - 1) ) / ( wss/(N - 1:K) )
ch[1] <- 0
```

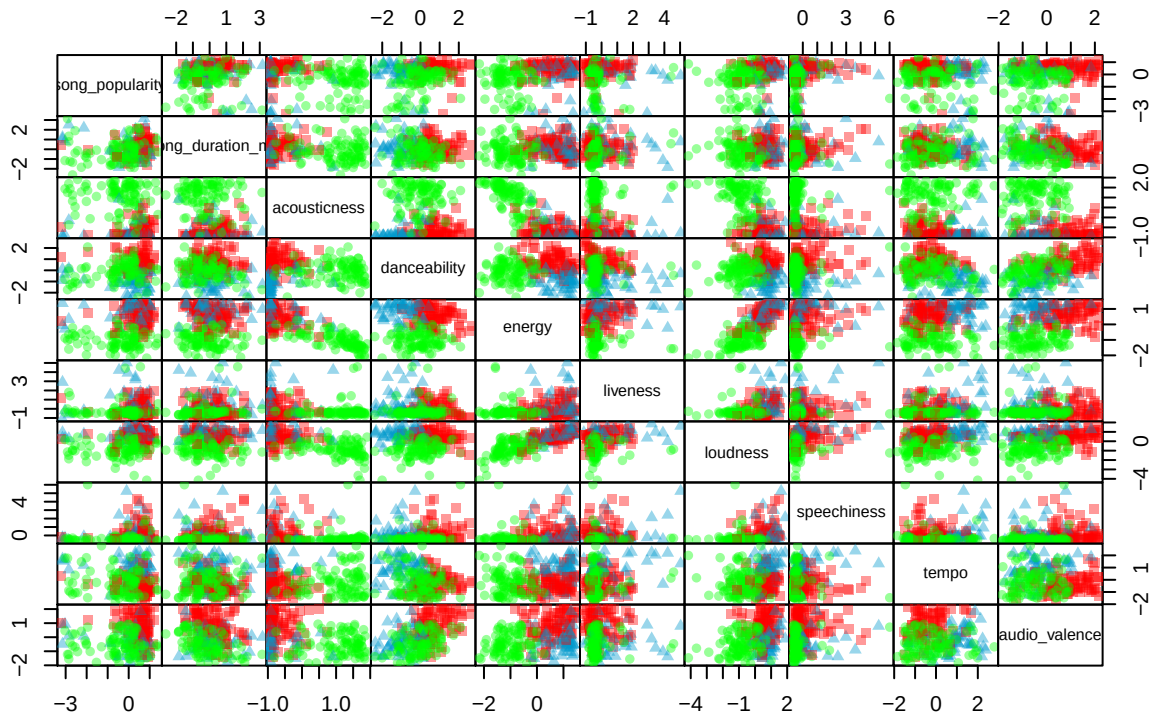
```r
# Plot of the CH Index
plot(1:K, ch, type = "b", ylab = "CH", xlab = "K")


fit2 <- kmeans(spotify_new, centers = 2, nstart = 50)

fit3 <- kmeans(spotify_new, centers = 3, nstart = 50)

symb <- c(15, 16, 17)
col = c("red", "green", "deepskyblue3")

# split the plot window in 2 screens
par( mfrow = c(1,2) )
```



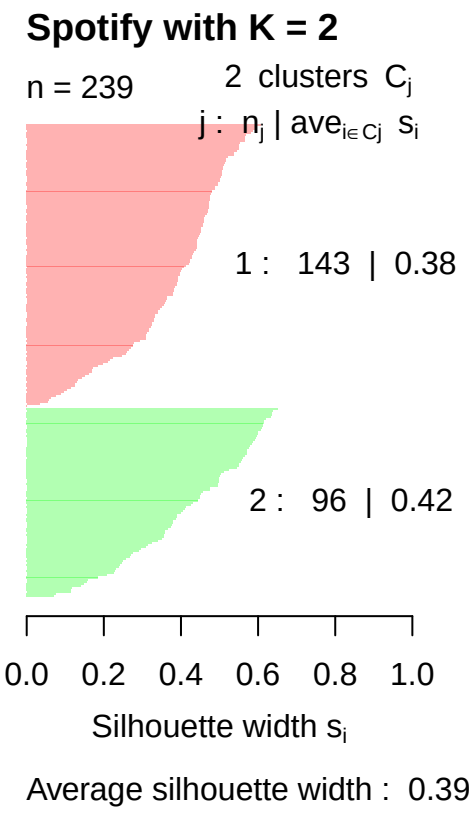

```r
# plot corresponding to 2 clusters
pairs(spotify_new, gap = 0, pch = symb[fit2$cluster],
      col = adjustcolor(col[1:2], 0.4), main = "Clustering Result - K = 2")
```

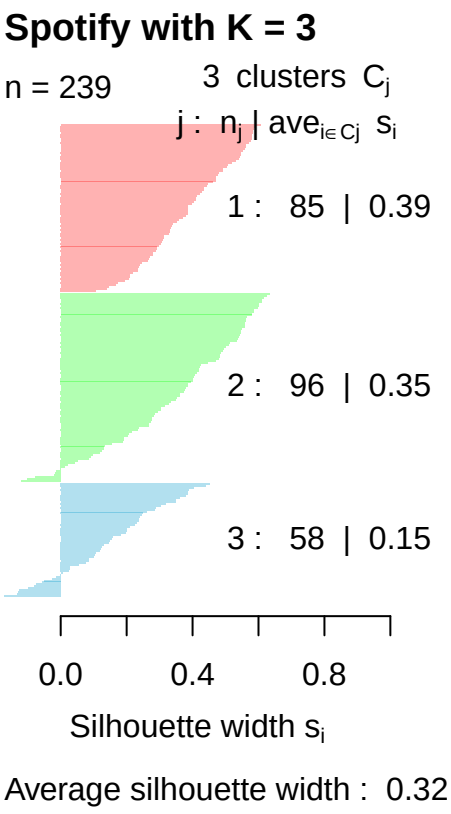



```
# plot corresponding to 3 clusters
pairs(spotify_new, gap = 0, pch = symb[fit3$cluster],
      col = adjustcolor(col[fit3$cluster], 0.4), main = "Clustering Result - K = 3")
```




```
# SILHOUETTE #
# Calling cluster library for Silhouette
library(cluster)

# Calculating the dissimilarity matrix
d = dist(spotify_new, method = "euclidean") ^ 2

# Silhouette for a two(2) cluster solution
sil2 = silhouette(fit2$cluster, d)
# Silhouette for a three(3) cluster solution
sil3 = silhouette(fit3$cluster, d)

# Producing two Silhouette Plots
col = c("red", "green", "deepskyblue3")

par(mfrow = c(1,2) )

#Plot of Silhouette for a 2 cluster solution
plot(sil2, col = adjustcolor(col[1:2], 0.3), main ="Spotify with K = 2")

#Plot of Silhouette for a 3 cluster solution
plot(sil3, col = adjustcolor(col, 0.3), main ="Spotify with K = 3")
```






```
# EXTERNAL VALIDATION #

#Using Rand Index and External Rand Index

#Calling library e1071 for calculating Rand and Adjusted Rand Index
library(e1071)

tab = table(fit2$cluster, spotify[,1])
tab
```

```
##
##     acoustic pop rock
##   1       10  75   58
##   2       90   5    1
```

```
tab2 = table(fit3$cluster, spotify[,1])
tab2
```

```
##
##     acoustic pop rock
##   1        6  56   23
##   2       90   5    1
##   3        4  19   35
```

```r
classAgreement(tab)
```

```
## $diag
## [1] 0.06276151
##
## $kappa
## [1] -0.5234626
##
## $rand
## [1] 0.7342569
##
## $crand
## [1] 0.4741481
```

```r
classAgreement(tab2)
```

```
## $diag
## [1] 0.1924686
##
## $kappa
## [1] -0.2294304
##
## $rand
## [1] 0.7744805
##
## $crand
## [1] 0.5007643
```

```r
### REPORT ###
### MOTIVATION and COMMENTS ###


# From the CH index, there is an elbow at k =2 and from 2 downwards there is a downward movement on the
# till it  fades off. From this , I will suggest a
# two cluster solution.

# From the Silhouette , the two cluster solution has a greater Silhouette number than that
# of the 3 cluster solution. So I will suggest a two cluster solution
# from the Silhouette Index.
#
# From the Visulaisation of the plots of the two and three clusters,
# the plot of the two clusters seems to be more clear and pass more
#information than that of the three clusters. I will also suggest a two
# cluster solution from the visualisations.
#
# From the Rand Index, Most of our data in cluster 1 belong to the acoustic genre with
# little representation of the pop and rock genre while
# most of our data in Cluster two belong to the pop genre, followed by the rock genre and
# little representation of the acoustic genre.

# The Rand Index gives a value of 0.73 which is quite large
# and the adjusted Rand Index gives a value of 0.47 which is okay and
```

```
# still suggests a fairly good fit.
#
```