STAT40750 – Statistical Machine Learning (Online)

Daniel Williams 21203054

15/02/2022

Exercise 1

Question 1

let ozark = a, stranger things = b, daredevil = c, chernobyl = d

```
n <- 9690
a <- 1877
b <- 1070
c <- 910
d <- 506
ac <- 261
ab <- 242
ad <- 140
bc <- 117
bd <- 82
cd <- 65
abc <- 50
abd <- 27
acd <- 21
bcd <- 6
```

Looking for the probability that a given customer watched all 4 shows Using the Frechet bounds First looking for the upper bound Pr(A,B) =< min{Pr(A), Pr(B)}

We have many ways in which to construct A and B, as we have pairs data, triplet data, and singles data. For the upper bound we can just take the lowest probability from all of these data. It will result in the bcd result being our upper bound, this is the least likely result we have data for and for a customer to have watched all 4 shows, they much have watched bcd.

These probabilities which I am calculating here will be used throughout the question.

```
prob_a <- a / n
prob_b <- b / n
prob_c <- c / n
prob_d <- d / n
prob_ab <- ab / n
prob_ac <- ac / n
prob_ad <- ad / n
prob_bc <- bc / n
prob_bd <- bd / n
prob_cd <- cd / n
prob_abc <- abc / n
prob_abd <- abd / n
prob_acd <- acd / n
prob_bcd <- bcd / n
```

Here we can now use the Frechet Bound, and calcluate the minimum of these probabilities which will be our upper bound.

```
Upper_Bound <- min(prob_a, prob_b, prob_c, prob_d, prob_ab, prob_ac, prob_ad, prob_bc, prob_b
d, prob_cd, prob_abc, prob_abd, prob_acd, prob_bcd)

Upper_Bound
```

```
## [1] 0.000619195
```

This makes sense, the highest the probability could be is if the 6 people that watched bcd, also in fact watched a.

Now focusing on the lower bound Pr(A,B) >= max{Pr(A) + Pr(B) - 1, 0} We must test our possible A’s and B’s in this formula to see if their sum minus one, is larger than zero. Test sets of pairs, and triplets with singles

```
T1 <- prob_a + prob_bcd - 1
T2 <- prob_b + prob_acd - 1
T3 <- prob_c + prob_abd - 1
T4 <- prob_d + prob_abc - 1
T5 <- prob_ab + prob_cd - 1
T6 <- prob_ac + prob_bd - 1
T7 <- prob_ad + prob_bc - 1
T8 <- 0
```

Our lower bound will then be the maximum value of these 8 tests.

```
Lower_Bound <- max(T1, T2, T3, T4, T5, T6, T7, T8)
Lower_Bound
```

```
## [1] 0
```

This makes sense, intuitively we are testing… is it possible that all 506 people that watched d, could have feasibly not watched a, and yes as the total subscribers is 9690… it is possible, hence lower bound is 0.

Range of values 0 >= P(ABCD) <= 0.00062

Exercise 1, Question 2.

Focus on customers with daredevil & ozark, ac. Recommend whether to watch b, or d Some initial simple analysis, we can compare the probability of acd, and abc

```
acd
```

```
## [1] 21
```

```
abc
```

```
## [1] 50
```

This gives us initial thinking that b may be the correct answer, but lets progress more formally.

Using the association rules, quality measures - LIFT Lift is an estimate of the association between the two itemsets A and B. In our case, let A = ac, and B firstly be b. Lift_A_B = Prob_AB / (prob_A * Prob_B) We shall also compute confidence & support to add to the analysis.

```
Lift_ac_b <-  prob_abc / (prob_ac * prob_b)
Support_ac_b <- abc / n
Confidence_ac_b <- abc / ac

Lift_ac_b
```

```
## [1] 1.73488
```

```
Support_ac_b
```

```
## [1] 0.005159959
```

```
Confidence_ac_b
```

```
## [1] 0.1915709
```

Same calculation with B = d

```
Lift_ac_d <-  prob_acd / (prob_ac * prob_d)
Support_ac_d <- acd / n
Confidence_ac_d <- acd / ac

Lift_ac_d
```

```
## [1] 1.54082
```

```
Support_ac_d
```

```
## [1] 0.002167183
```

```
Confidence_ac_d
```

```
## [1] 0.08045977
```

We see that the lift, support & confidence figures are greater for ac_b, therefore we would recommend that a customer who has watched ac, should now watch b - Stranger things. This is because the association between these two itemsets is larger than with ac & d. We are most interested here in the LIFT quality measure.

Exercise 1, Q3.

Lift of the rule… stranger things & daredevil… bc, to ozark a. Lift_A_B = Prob_AB / (prob_A * Prob_B) here we let A = bc , B = a

```
Lift_bc_a <- prob_abc / (prob_bc * prob_a)
Lift_bc_a
```

```
## [1] 2.206194
```

Lift figure of 2.20, this is well above 1, which indicates higher association. This is also above the two other lift figures in the previous question, again indicating high association for bc & a

Exercise 2

Question 1

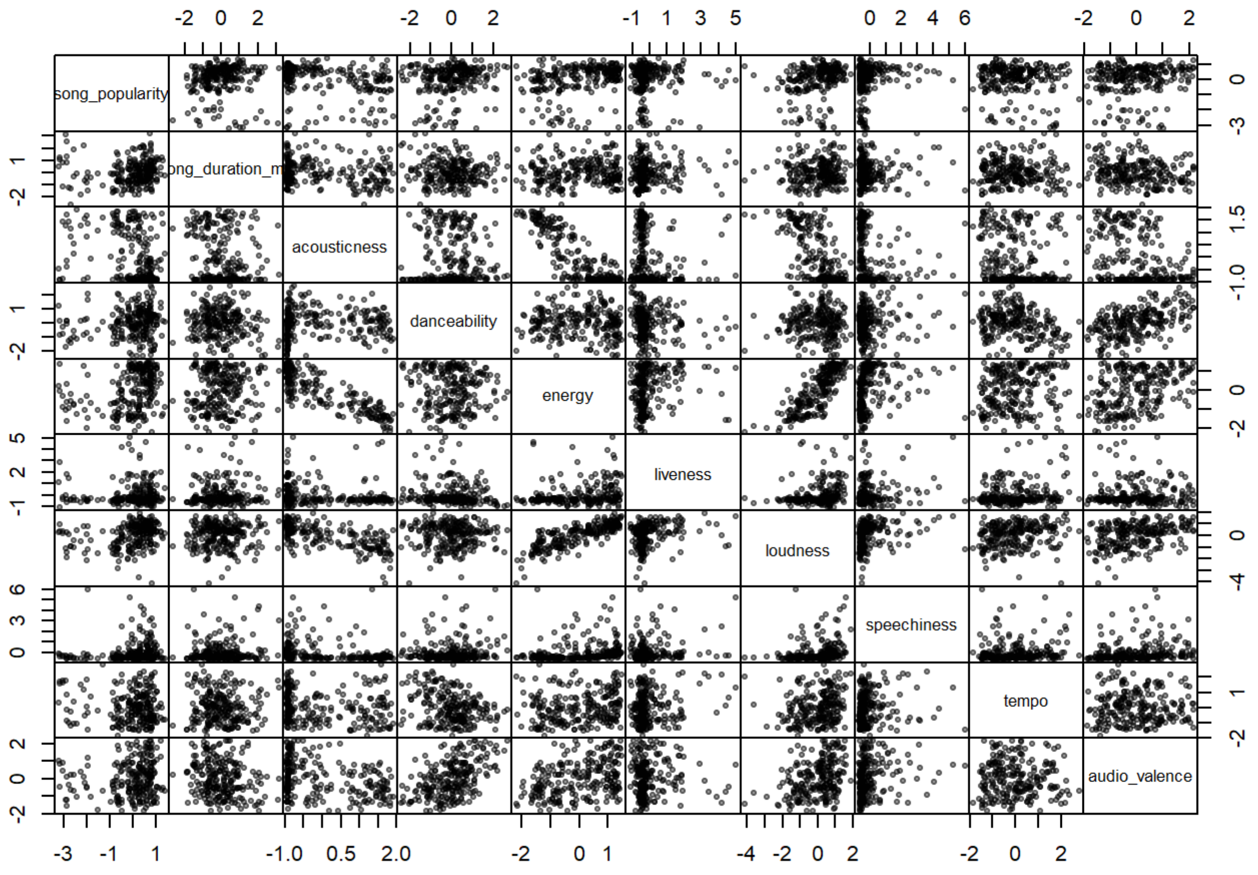
```
load("~/data_spotify_songs.rda")
View(spotify)
```

Using the Calinski-Harabasz index to decide a suitable number for K First we take out the categorical variables from the data in cols 1:3. We have a look at the pairs plot for all variables, and see if we can initially see clusters.

```
str(spotify)
```

```
## 'data.frame':    239 obs. of  13 variables:
##  $ genre           : Factor w/ 3 levels "rock","pop","acoustic": 1 1 1 1 1 1 1 1 1 1 ...
##  $ song_name       : Factor w/ 237 levels "Acoustic","Adore - Piano Unplugged",..: 99 171 
31 89 29 109 11 137 172 200 ...
##  $ artist          : Factor w/ 178 levels "*NSYNC","3 Doors Down",..: 111 169 141 123 55 1
33 81 164 100 83 ...
##  $ song_popularity : int  66 76 74 56 80 81 76 80 81 78 ...
##  $ song_duration_ms: int  216933 231733 216933 223826 235893 199893 213800 222586 203346 1
68253 ...
##  $ acousticness    : num  0.0103 0.00817 0.0264 0.000954 0.00895 0.000504 0.00148 0.00108 
0.00172 0.0424 ...
##  $ danceability    : num  0.542 0.737 0.451 0.447 0.316 0.581 0.613 0.33 0.542 0.629 ...
##  $ energy          : num  0.853 0.463 0.97 0.766 0.945 0.887 0.953 0.936 0.905 0.897 ...
##  $ liveness        : num  0.108 0.255 0.102 0.113 0.396 0.268 0.152 0.0926 0.136 0.263 ...
##  $ loudness        : num  -6.41 -7.83 -4.94 -5.07 -3.17 ...
##  $ speechiness     : num  0.0498 0.0792 0.107 0.0313 0.124 0.0624 0.0855 0.0917 0.054 0.04
83 ...
##  $ tempo           : num  105 124 122 172 190 ...
##  $ audio_valence   : num  0.37 0.324 0.198 0.574 0.32 0.724 0.537 0.234 0.374 0.93 ...
```

```
spot_kmeans <- scale(spotify[,-c(1:3)])
pairs(spot_kmeans, gap = 0, pch = 20, col = adjustcolor(1, 0.4))
```



Setting the value of k to be 10 as specified in the question. WSS will measure the variation within the clusters in terms of sum of squares. BSS will measure the variation between different clusters in the same manor.

Using a loop to iterate through the kmeans fitting for 1 up to 10 clusters. Adds the WSS & BSS to a vector for each value of K. Uses that within the Calinski Harabaz formula. A high value of CH indicates successful clustering, high variation between clusters compared to within clusters.

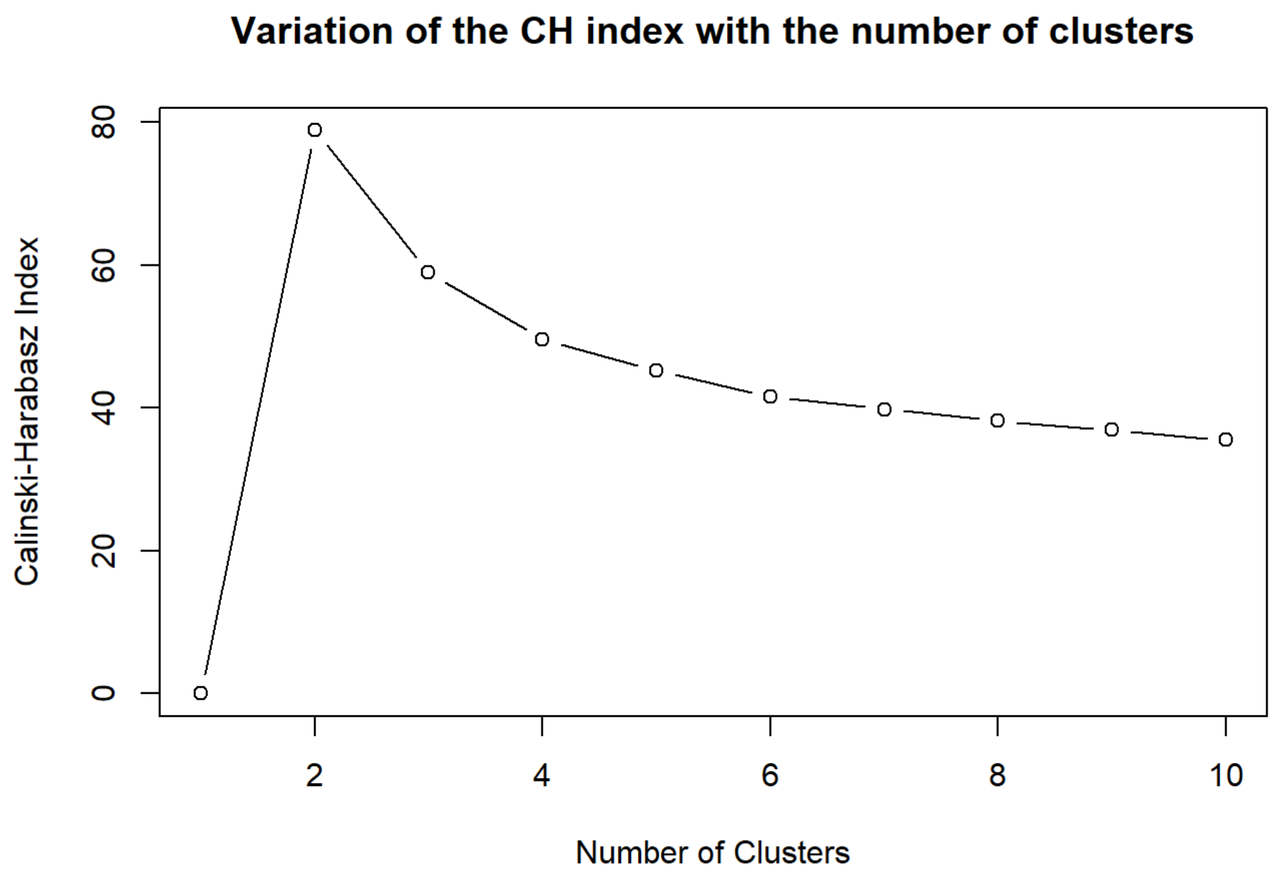
CH = (BSS/(k-1)) / (WSS / (n-k))

```
K <- 10
wss <- bss <- rep(NA, K)

for ( k in 1:K ) {
  fit <- kmeans(spot_kmeans, centers = k, nstart = 40)
  wss[k] <- fit$tot.withinss
  bss[k] <- fit$betweenss
}
N <- nrow(spot_kmeans)
ch <- ( bss/(1:K - 1) ) / ( wss/(N - 1:K) )
ch[1] <- 0
```

Plotting the data of the CH index, against number of clusters to identify the optimum segregation.

```
plot(1:K, ch, type = "b", ylab = "Calinski-Harabasz Index", xlab = "Number of Clusters", main
= 'Variation of the CH index with the number of clusters')
```



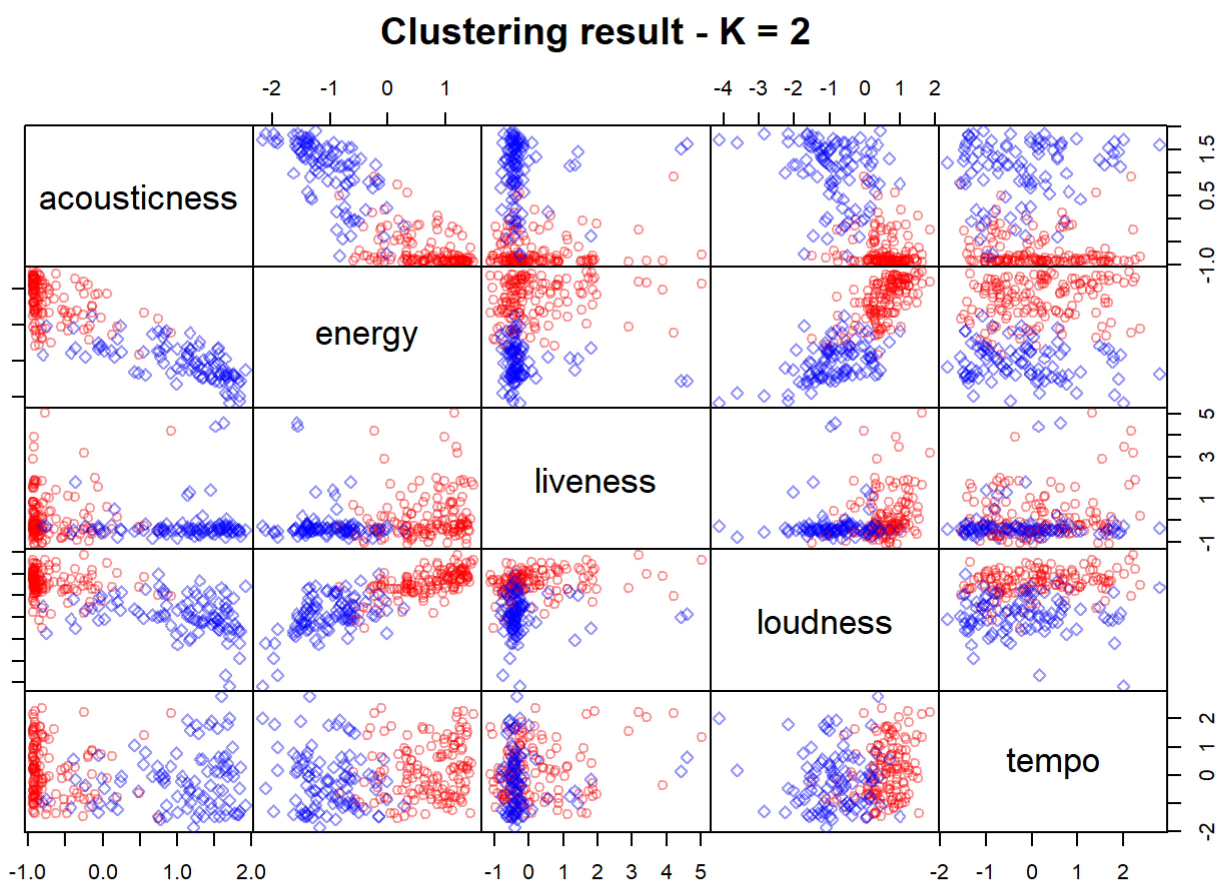
We want to maximize the CH index, as this shows a large value of difference between clusters when compared to the difference within clusters… thus the best partitioning. We see from the graph that this occurs with K = 2, then K = 3 is another candidate however at a lower CH. Other values of k are not really to be considered based on this graph.

We now model the two & 3 cluster scenarios to allow us to inspect them visually.

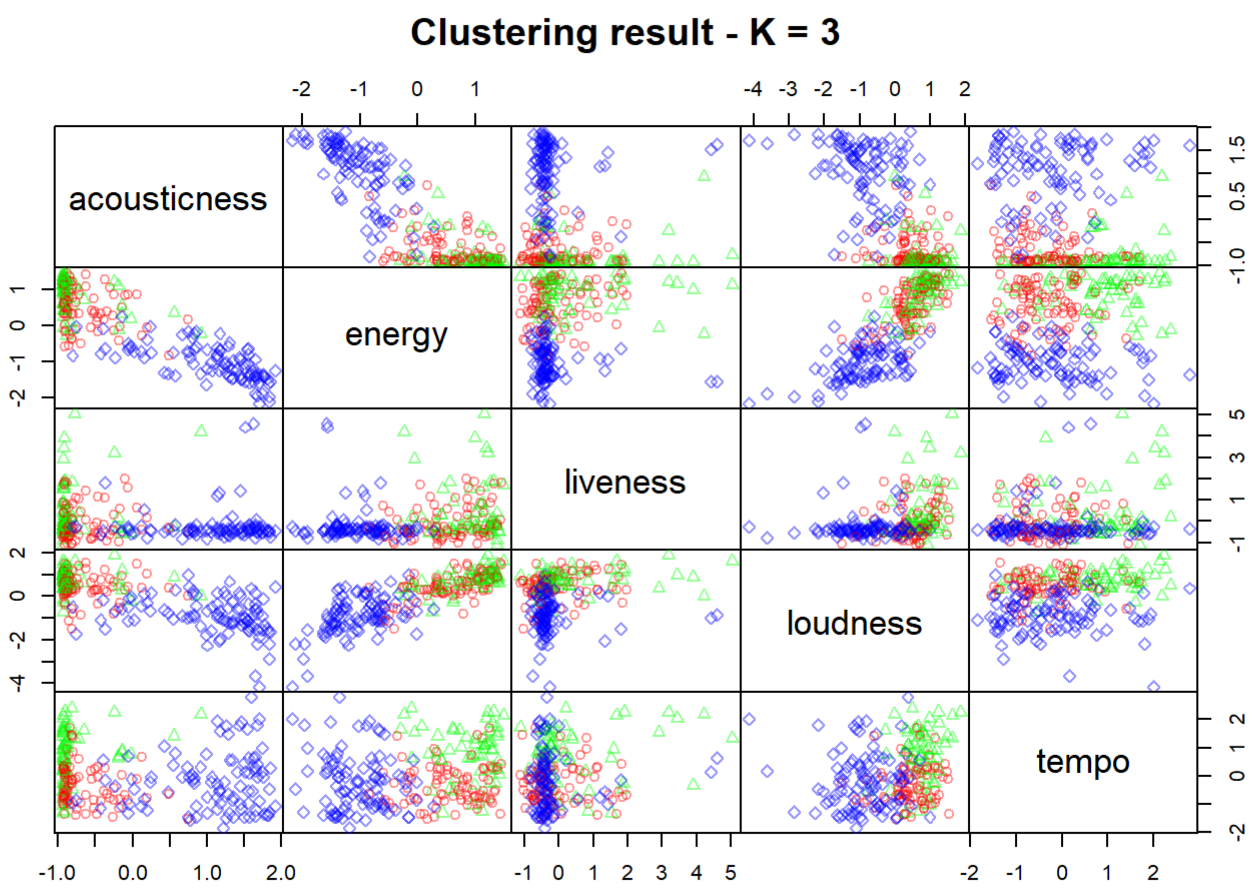
```
two_clust <- kmeans(spot_kmeans, centers = 2, nstart = 40)
three_clust <- kmeans(spot_kmeans, centers = 3, nstart = 40)
```

Plotting 2 & 3 next to each other for a few different pairs, to inspect the clustering visually.

```
plot_markers <- c(21, 23, 24)
colours <- c("red", "blue", "green")
par( mfrow = c(1,2) )
pairs(spot_kmeans[,c(3,5,6,7,9)], gap = 0, pch = plot_markers[two_clust$cluster],
      col = adjustcolor(colours[two_clust$cluster], 0.4), main = "Clustering result - K = 2")
```



```
pairs(spot_kmeans[,c(3,5,6,7,9)], gap = 0, pch = plot_markers[three_clust$cluster],
      col = adjustcolor(colours[three_clust$cluster], 0.4), main = "Clustering result - K =
 3")
```



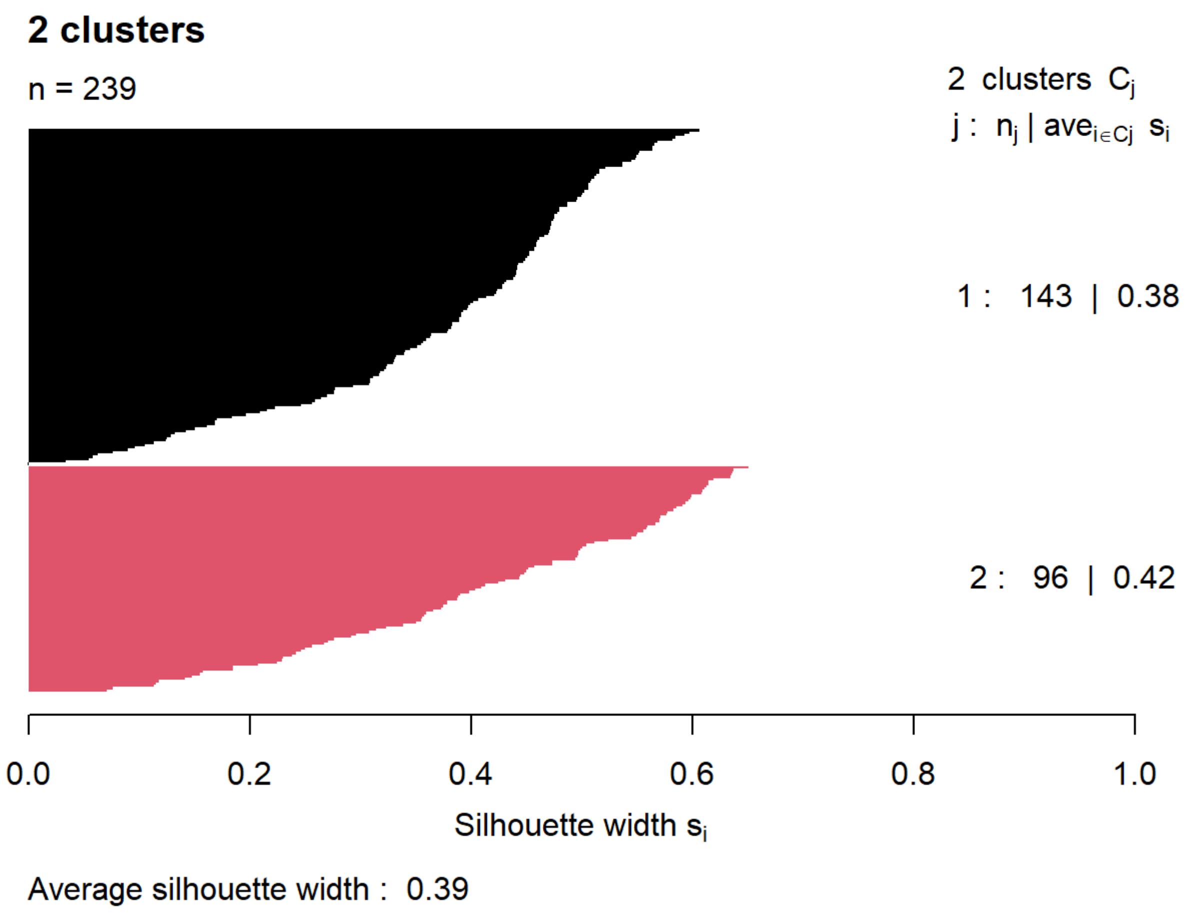
We see from this, that the K=2 scenario seems to have better separation, which is inkeeping with the CH index result being larger for this case too.

Now looking at the Silhouettes for both results too.

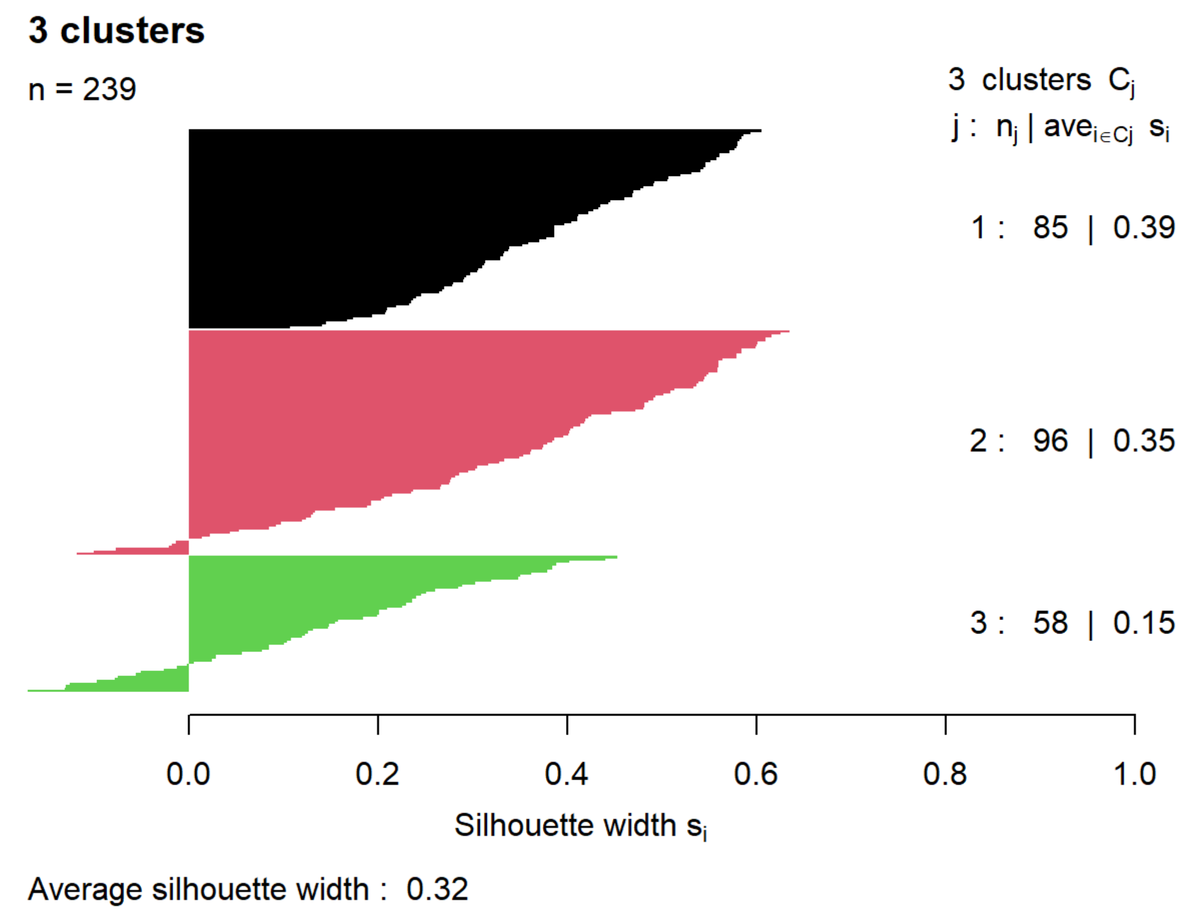
```r
library(cluster)
euc_dist <- dist(spot_kmeans, method = "euclidean")^2

two_clust_silhouette <- silhouette(two_clust$cluster, euc_dist)
three_clust_silhouette <- silhouette(three_clust$cluster, euc_dist)

plot(two_clust_silhouette, col=1:2, border=NA, main = '2 clusters')
```



```
plot(three_clust_silhouette, col=1:3, border=NA, main = '3 clusters')
```



We see some negative values in the Silhouette plot for k=3, this suggests that these values may have been assigned to the wrong cluster. Again this suggets k=2 is optimum. We see larger average cluster scores for k=2, especially when compared to the green cluster for k=3, this is particuarly low.

Exercise 2, Question 2.

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.1.1
```

```
Two_Cluster_Performance <- table(two_clust$cluster, spotify$genre)
Two_Cluster_Performance
```

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

```
classAgreement(Two_Cluster_Performance)
```

```
## $diag
## [1] 0.2635983
##
## $kappa
## [1] -0.02585114
##
## $rand
## [1] 0.7342569
##
## $crand
## [1] 0.4741481
```

```
Three_Cluster_Performance <- table(three_clust$cluster, spotify$genre)
Three_Cluster_Performance
```

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

```
classAgreement(Three_Cluster_Performance)
```

```
## $diag
## [1] 0.1338912
##
## $kappa
## [1] -0.2808212
##
## $rand
## [1] 0.7744805
##
## $crand
## [1] 0.5007643
```

The rand index, and the adjusted rand index are actually both higher for K=3. This suggests that K=3 is optimum. As a value of 1 for this metric would represent perfect alignment between the external class information, and the cluster results.

Looking at the K=3 table, It seems to have successfully located the acoustic music into group 2, however it seems to have a difficult time ‘correctly’ differentiating between rock & pop.

Again with the K=2 table, the acoustic songs are clustered effectively, and again the rock & pop songs seem to be grouped together, which aligns with our K=3 results.