Lab 4

Thomas Haase

September 19, 2025

Table of contents

Part 1 - Taste clustering and influence

In the first part of this lab, we will consider a (simulated) data set which contains information about a sample of (fictive) individuals' music tastes as well as a measure of their influence on others.

Task 1

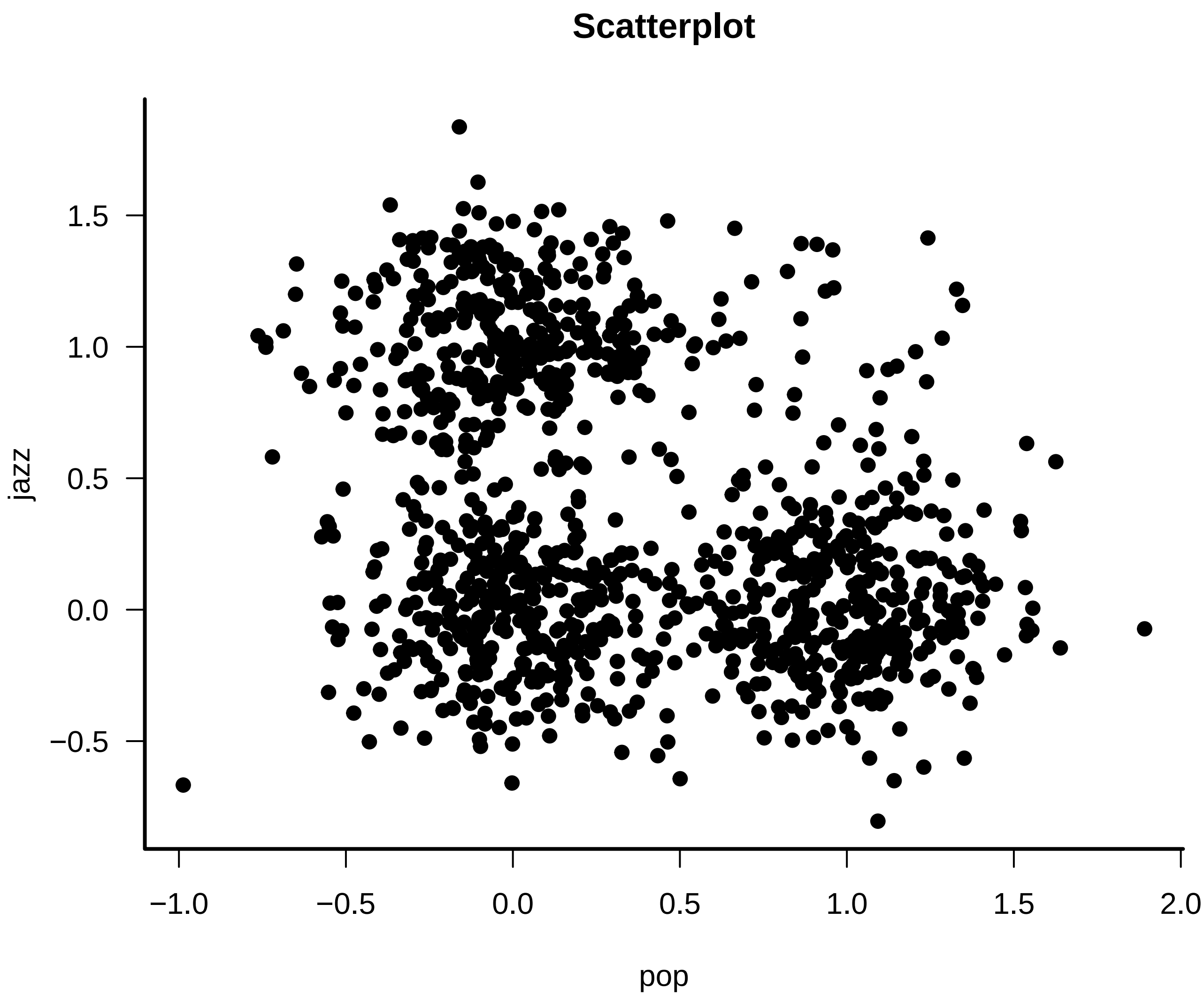
> Begin by importing the file "taste_influence.csv". Report the number of rows and columns of the data set, and the genres contained in it. Create a scatter-plot of two combinations of genres of your choice. Based on this, do you get any indication that the data is clustered along musical tastes?

```r
library(data.table)

library(scatterplot3d)

library(tibble)
library(kableExtra)

setwd("~/Github/ML-Labs/4")
d <- fread("taste_influence.csv")

tribble(
  ~Name,     ~Value,
  "Rows",    nrow(d) |> as.character(),
```

```r
  "Columns", ncol(d) |> as.character(),
  "Genre 1", names(d)[1],
  "Genre 2", names(d)[2],
  "Genre 3", names(d)[3]
) |> kable()
```

Name	Value
Rows	1075
Columns	4
Genre 1	jazz
Genre 2	pop
Genre 3	hiphop

```r
d[,c("pop","jazz")] |>
  plot(type = "p", bty = "n", pch = 20, cex = 1.5, las = 1,
       main = "Scatterplot")
  box("plot", bty = "l", lwd = 2)
```



There are defined clusters visible in the plot. The main observation is that individuals that like pop a lot do not seem to like jazz and the other way around.

Task 2

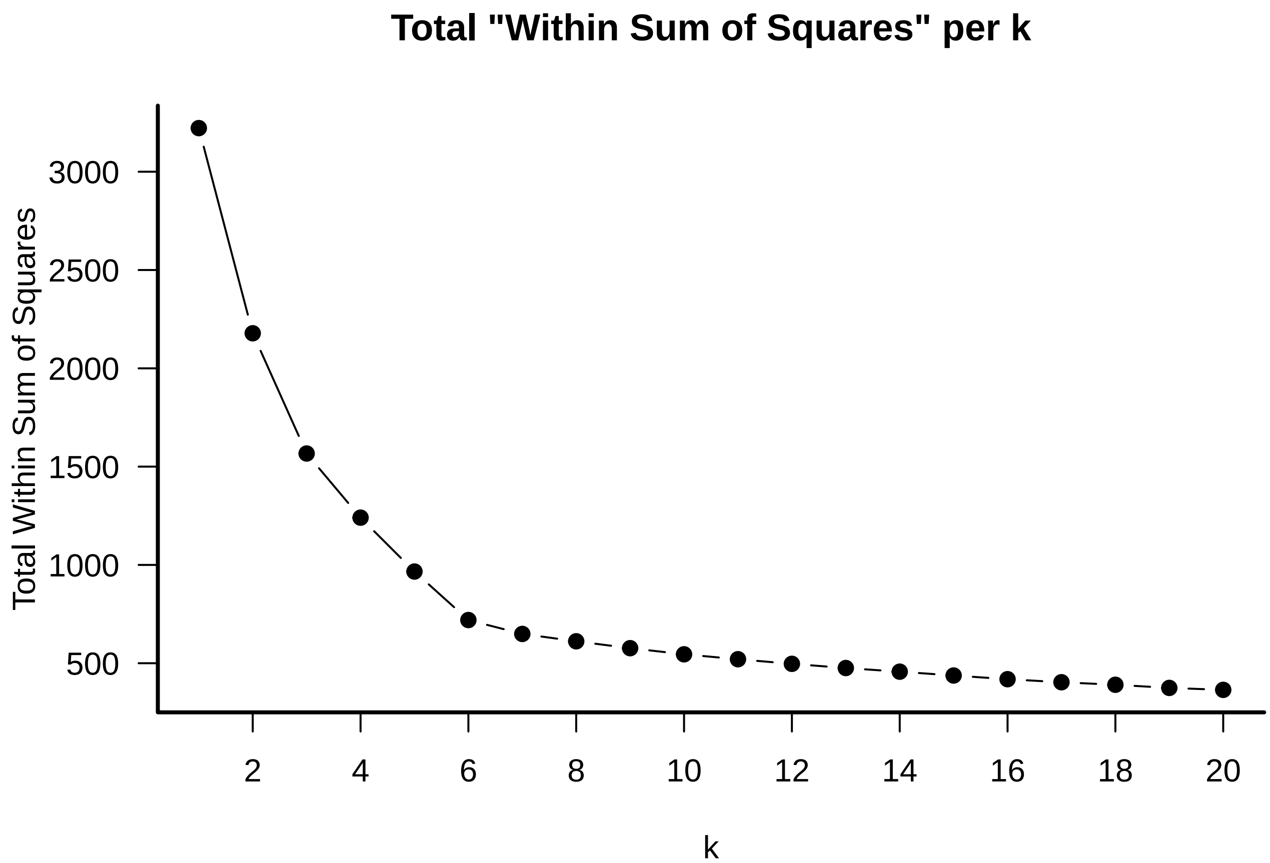
Now you shall do some clustering. To prepare the data, do the following: (i) store/copy the data to a new R object, and subset it so that it only contains the three "taste columns"—these are the columns you will cluster based upon, (ii) standardize this data table (hint: you can e.g., use scale() for this purpose), (iii) transform it into a matrix (hint: e.g., by using as.matrix()).

```r
dc <- d[,-"influence"] |>
  scale() |>
  as.matrix()
```

Task 3

Having formatted the data according to #2, you shall now use the kmeans algorithm to cluster your data. Recall that a requisite for running kmeans is that the parameter k has been specified. In practice—and as is the case here—we often do not know the appropriate number of clusters a priori. Therefore, you shall implement a loop that, at every iteration, runs kmeans with a different number of clusters, and extracts the total within cluster sum of squares (hint 1: which can be extracted using $tot.withinss | hint 2: set the argument nstart=100 to ensure robustness of the local optima you find). Consider no. clusters ranging from 1 to 20, with an interval of 1. Plot k against tot.withinss. Which number of clusters do you find appropriate? Motivate.

```r
set.seed(5)
k <- 1:20
wss <- c()

for(i in 1:length(k)){
  temp <- kmeans(x = dc,
                 centers = k[i],
                 nstart = 100)

  wss[i] <- temp$tot.withinss
}


tibble(k = 1:20,`Total Within Sum of Squares` = wss) |>
plot(type = "b", bty = "n", pch = 20, cex = 1.5, las = 1,
     main = 'Total "Within Sum of Squares" per k',
     xaxt = "n", xlab = "k", ylab =)
axis(1, at = seq(0, 20, 2))
box("plot", bty = "l", lwd = 2)
```

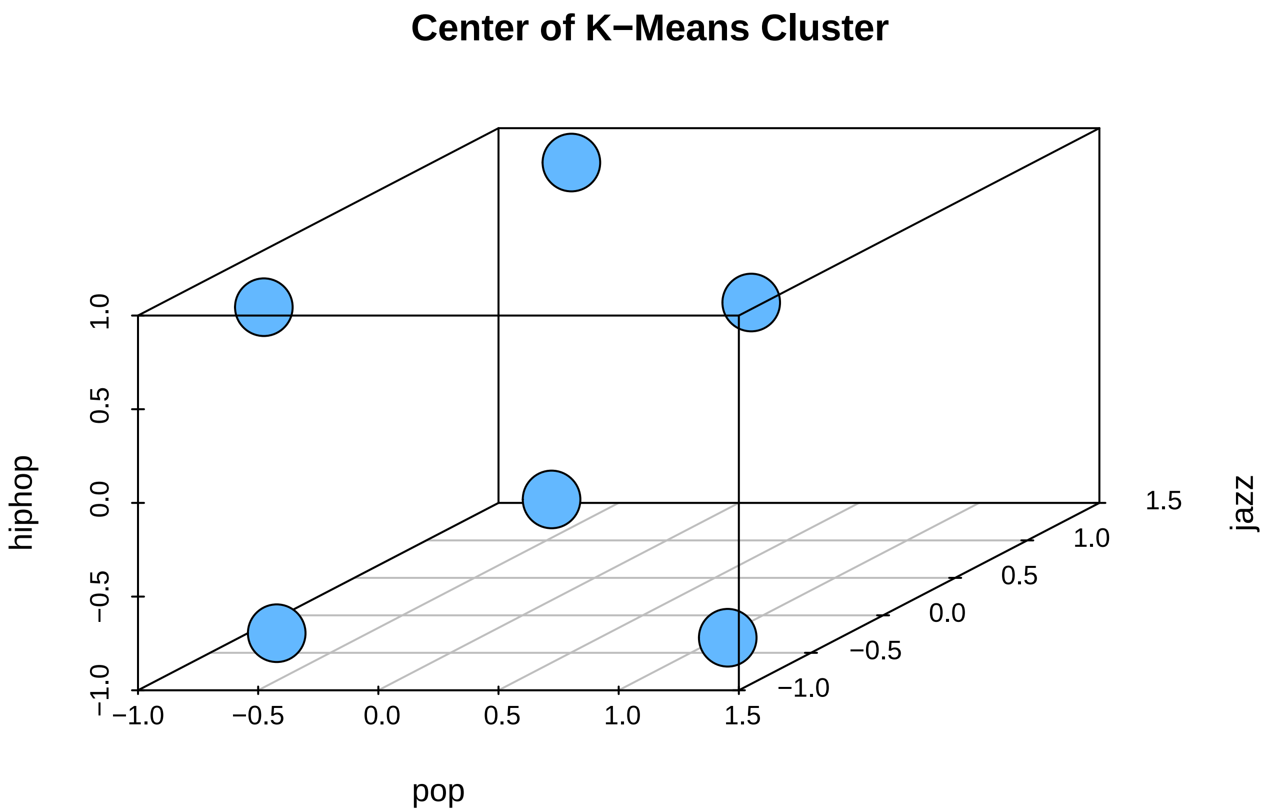


At $k = 6$ the plot seems to have an elbow, indicating decreasing gain in the total within sum of squares.

Task 4

For the specification (of k) that you decided on in #3, extract the centroids and interpret each cluster in terms of what distinguishes it from the rest. Do the clusters seem meaningfully distinct?

```
km6 <- kmeans(x = dc,
              centers = 6,
              nstart = 100)

km6c <- km6$centers |> as.data.frame()

scatterplot3d(x = km6c$pop, y = km6c$jazz, z = km6c$hiphop,
              xlab = "pop", ylab = "jazz", zlab = "hiphop",
              main = "Center of K-Means Cluster",
              type="p", pch = 21, bg = "steelblue1",
              cex.symbols = 4   ,angle = 36)
```


Center of K−Means Cluster
hiphop
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
1.5
pop
jazz
1.5
1.0
0.5
0.0
−0.5
−1.0