Lab 4

Thomas Haase

September 23, 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.

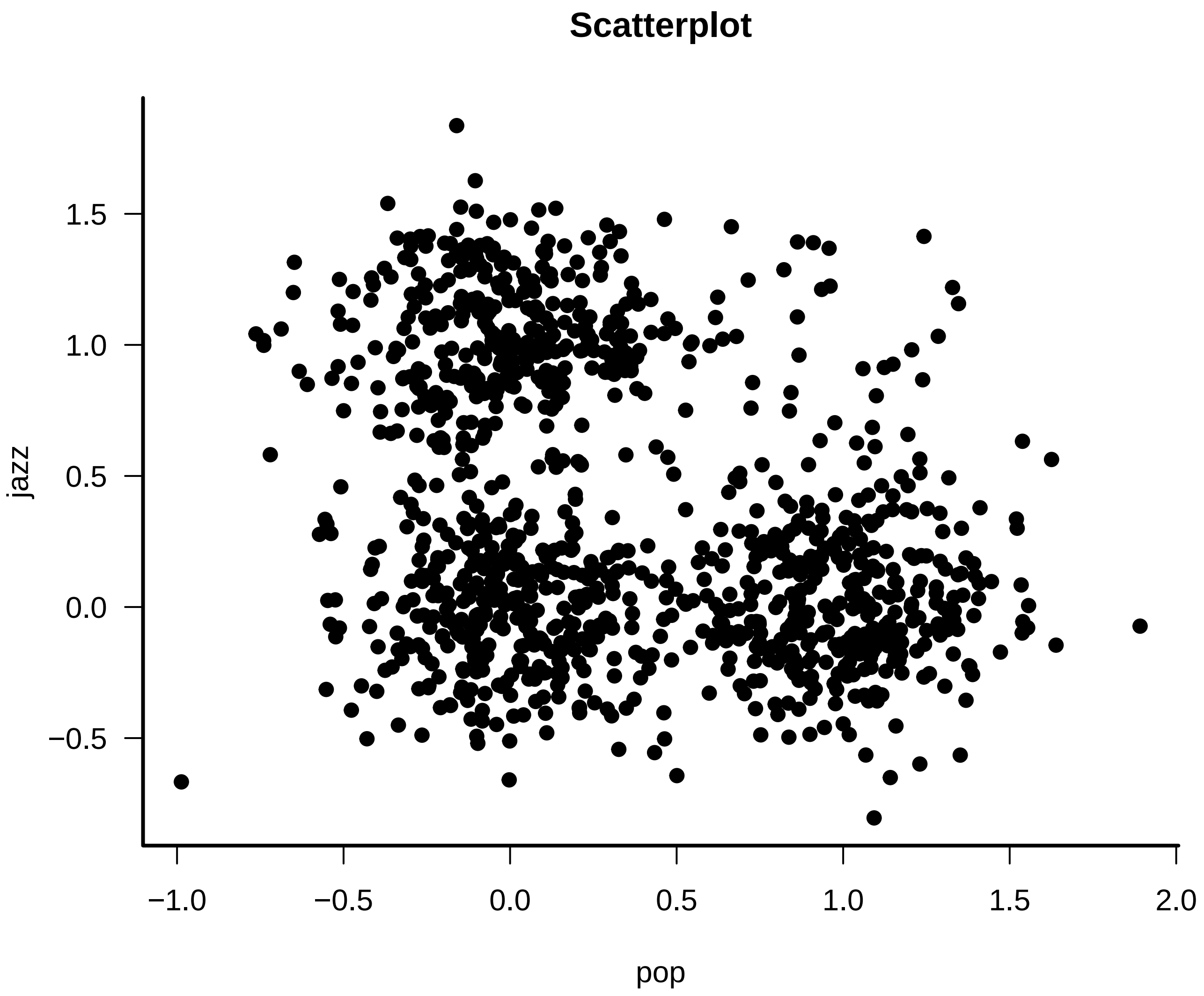
Task 1

> 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(mclust)
library(elasticnet)

library(scatterplot3d)

library(tibble)
library(easystats)
library(kableExtra)

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

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

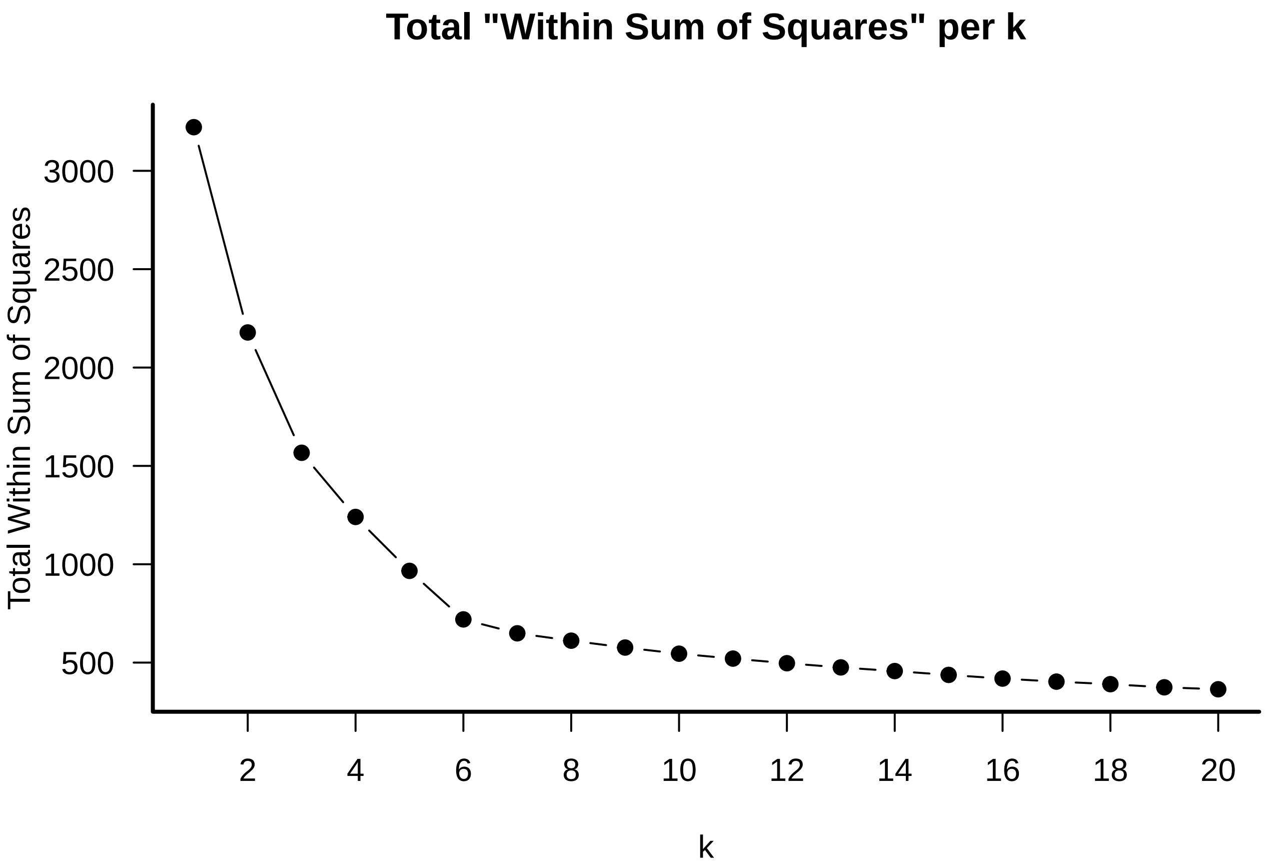
> Now you shall do some clustering. To prepare the data, do the following: (i) store/copy the data to a new R object, and subset it so that it only contains the three "taste columns"—these are the columns you will cluster based upon, (ii) standardize this data table (hint: you can e.g., use scale() for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

```
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)
```

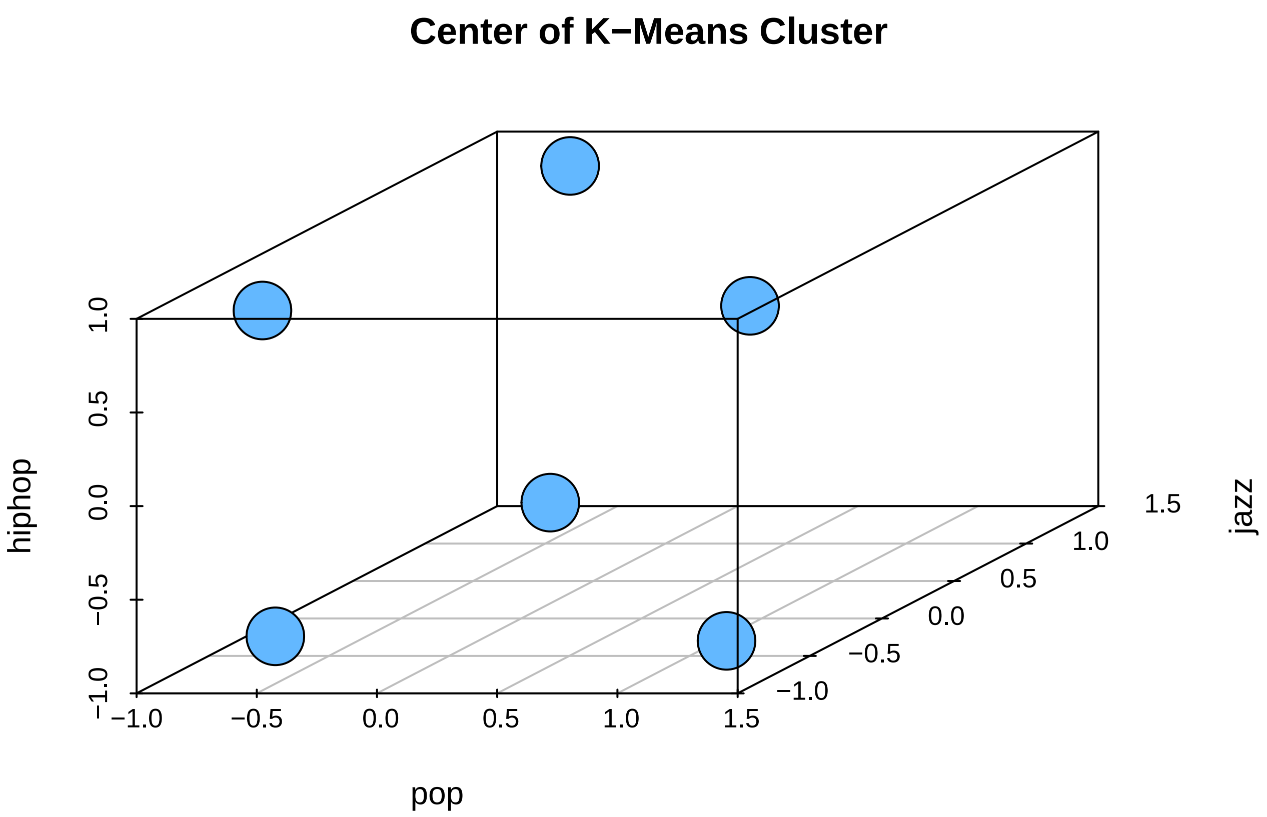


At $k = 6$ the plot seems to have an elbow, indicating decreasing gain in the total within sum of squares.

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?

```r
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)
```

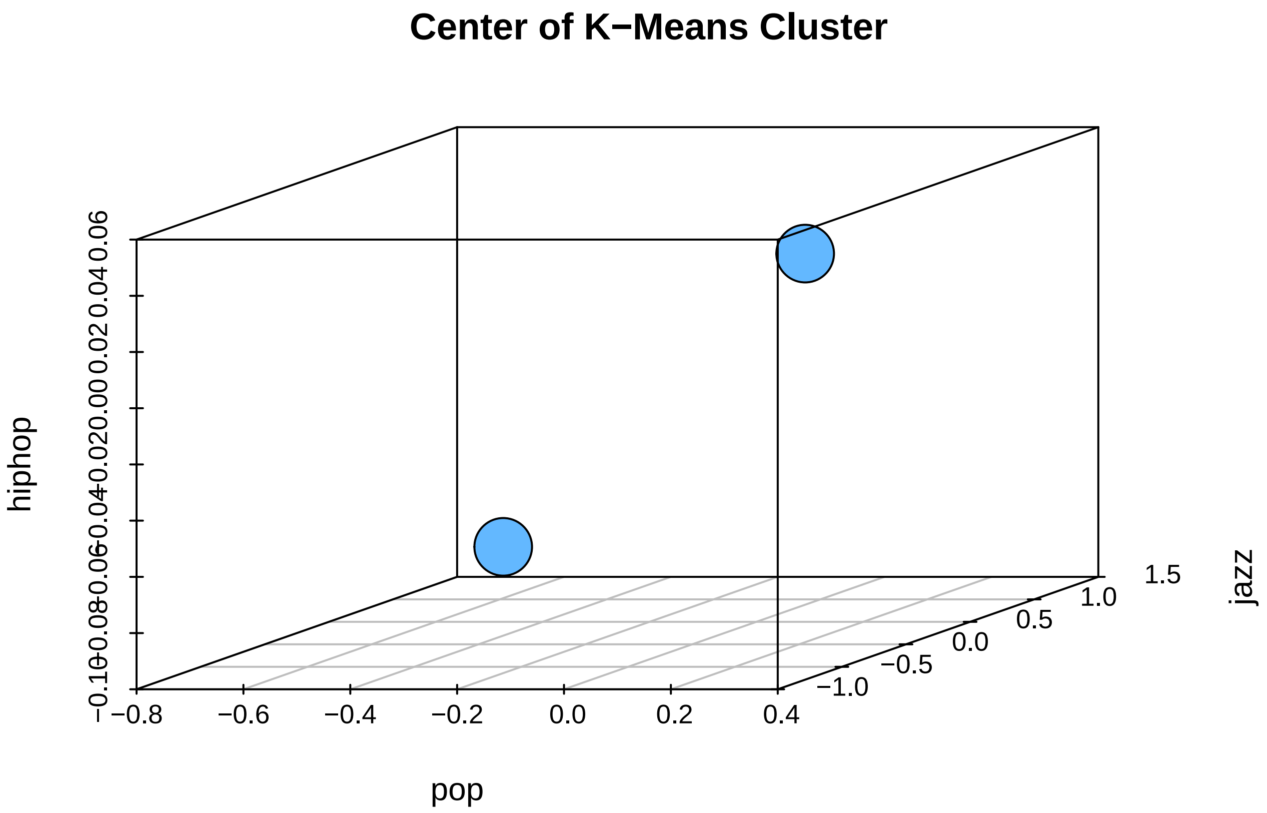


The clusters have very distinct places. There are listeners of almost all combinations of genres, except jazz-fans. There are no clusters of people with high jazz and hiphop or pop scores - the people that like jazz dislike pop and hiphop.

Task 5

To get a feeling for the role that the choice of k plays, estimate another kmeans model but this time with k = 2. Inspecting the centroids, how does your clustering change; how does it alter your understanding of the population?

```r
km2 <- kmeans(x = dc,
              centers = 2,
              nstart = 100)

km2c <- km2$centers |> as.data.frame()

scatterplot3d(x = km2c$pop, y = km2c$jazz, z = km2c$hiphop,
              xlab = "pop", ylab = "jazz", zlab = "hiphop",
              main = "Center of K-Means Cluster",
              type="p", pch = 21, bg = "steelblue1",
              cex.symbols = 4   ,angle = 36)
```



The centers of the two clusters have similar positions as two of the six clusters before. In the space of musical tastes these two clusters were the furthest apart from the six before. Since kmeans tries to find very distinct clusters it finds the two clusters that are the furthest apart. The six cluster plot from before also was able to capture the groups inbetween.

Task 6

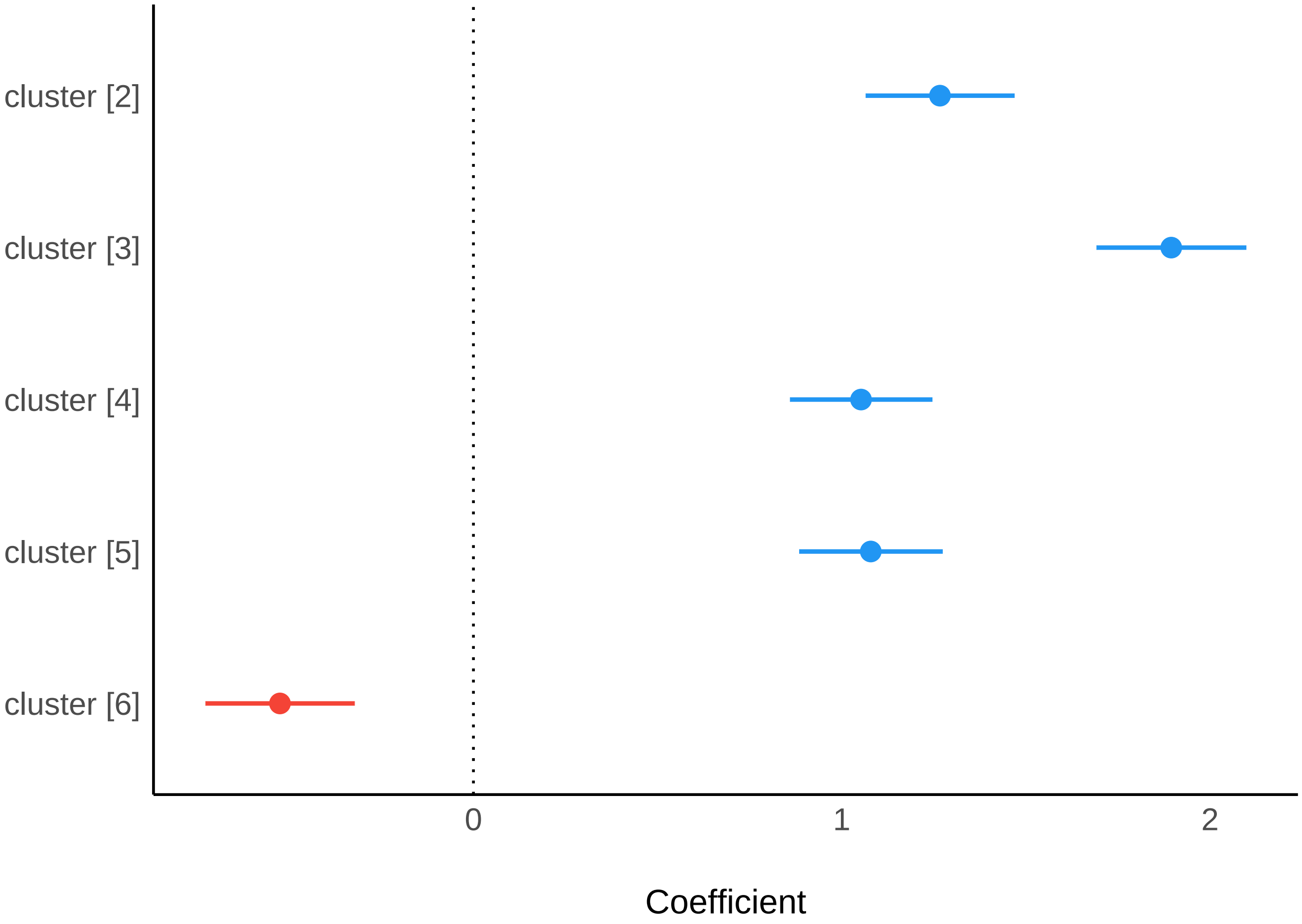
> Clustering provides a tool for discovering underlying structures in our data. Once these structures have been discovered, they can be studied in separate analyses. That is what you shall do now. We want to examine whether different "taste types" have differential degree of influence on others. To do so, (i) create a new column in your original data set storing the the retrieved cluster assignments (hint: you find the cluster assignments using $cluster). Then (ii) estimate a linear regression with the influence score (infuence) as the outcome variable, and the clusters (formatted as a factor) as predictors. Interpret the results: are there any difference in influence between the clusters?

```
d$cluster <- km6$cluster |> as.factor()

lm <- lm(influence ~ cluster, d)

lm |> report() |> summary() -> report
```

We fitted a linear model to predict influence with cluster. The model's explanatory power is substantial (R2 = 0.40, adj. R2 = 0.40). The model's intercept is at 0.72 (95% CI [0.58, 0.86]). Within this model:

- The effect of cluster [2] is statistically significant and positive (beta = 1.27, 95% CI [1.07, 1.47], t(1069) = 12.34, p < .001, Std. beta = 1.02) - The effect of cluster [3] is statistically significant and positive (beta = 1.89, 95% CI [1.69, 2.10], t(1069) = 18.34, p < .001, Std. beta = 1.52) - The effect of cluster [4] is statistically significant and positive (beta = 1.05, 95% CI [0.86, 1.25], t(1069) = 10.72, p < .001, Std. beta = 0.84) - The effect of cluster [5] is statistically significant and positive (beta = 1.08, 95% CI [0.88, 1.27], t(1069) = 10.90, p < .001, Std. beta = 0.87) - The effect of cluster [6] is statistically significant and negative (beta = -0.53, 95% CI [-0.73, -0.32], t(1069) = -5.10, p < .001, Std. beta = -0.42)

```
lm |> parameters() |> plot()
```

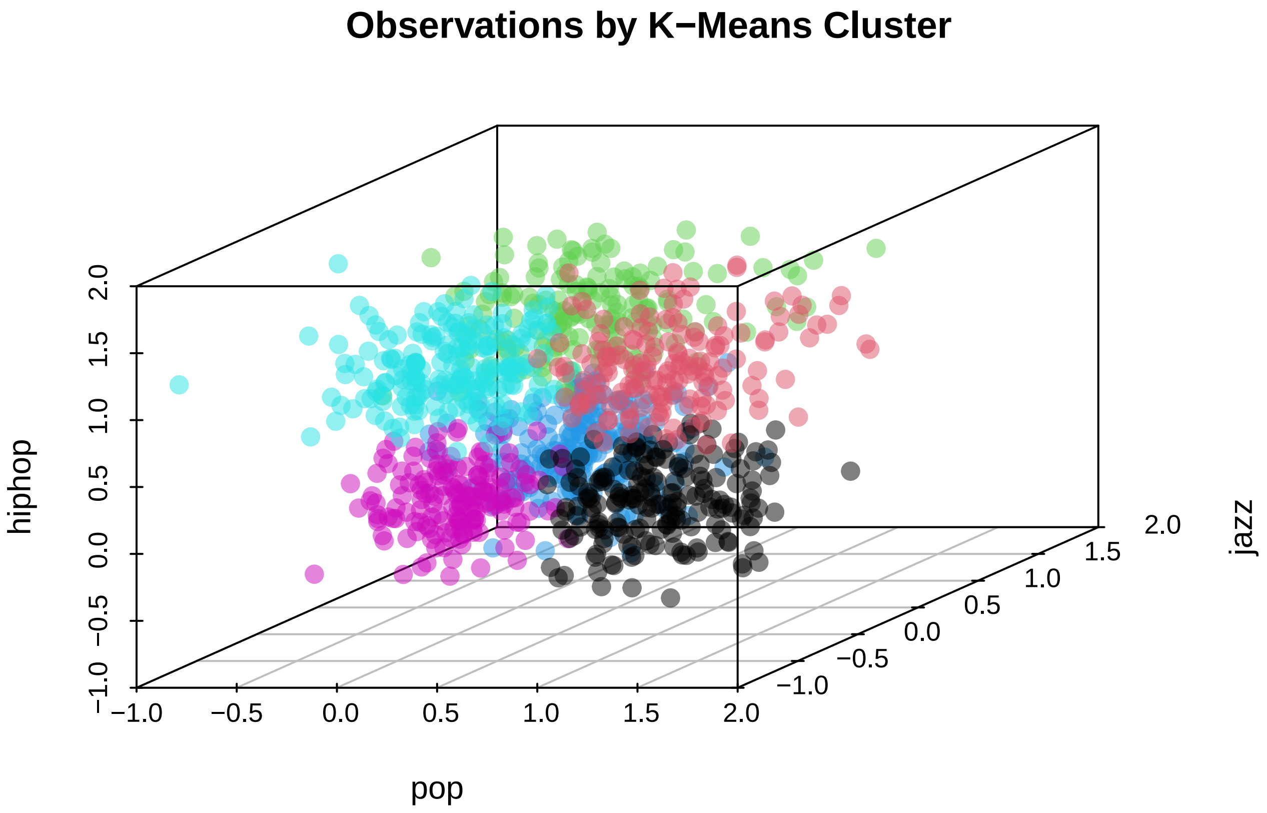


In comparison to cluster 1 all other clusters are associated with an increase in influence, except cluster 6, which is associated with a decrease in influence compared to cluster 1.

Task 7

Now that you have merged the cluster assignments to the original data, produce the same plots as you did in #1, but now colored by the cluster assignments. Does it look like kmeans have picked up on the patterns you observed in #1? Further—what you think of the separation between the clusters? Is there clear spacing betweeen the clusters, or are the borders almost touching each other (note that there will be certain overlap due to plotting the data in 2D)?

```
scatterplot3d(x = d$pop, y = d$jazz, z = d$hiphop,
              xlab = "pop", ylab = "jazz", zlab = "hiphop",
              main = "Observations by K-Means Cluster",
              type="p", pch = 19, cex.symbols = 1.2, angle = 36,
              color = d$cluster |> adjustcolor(alpha.f = 0.5))
```

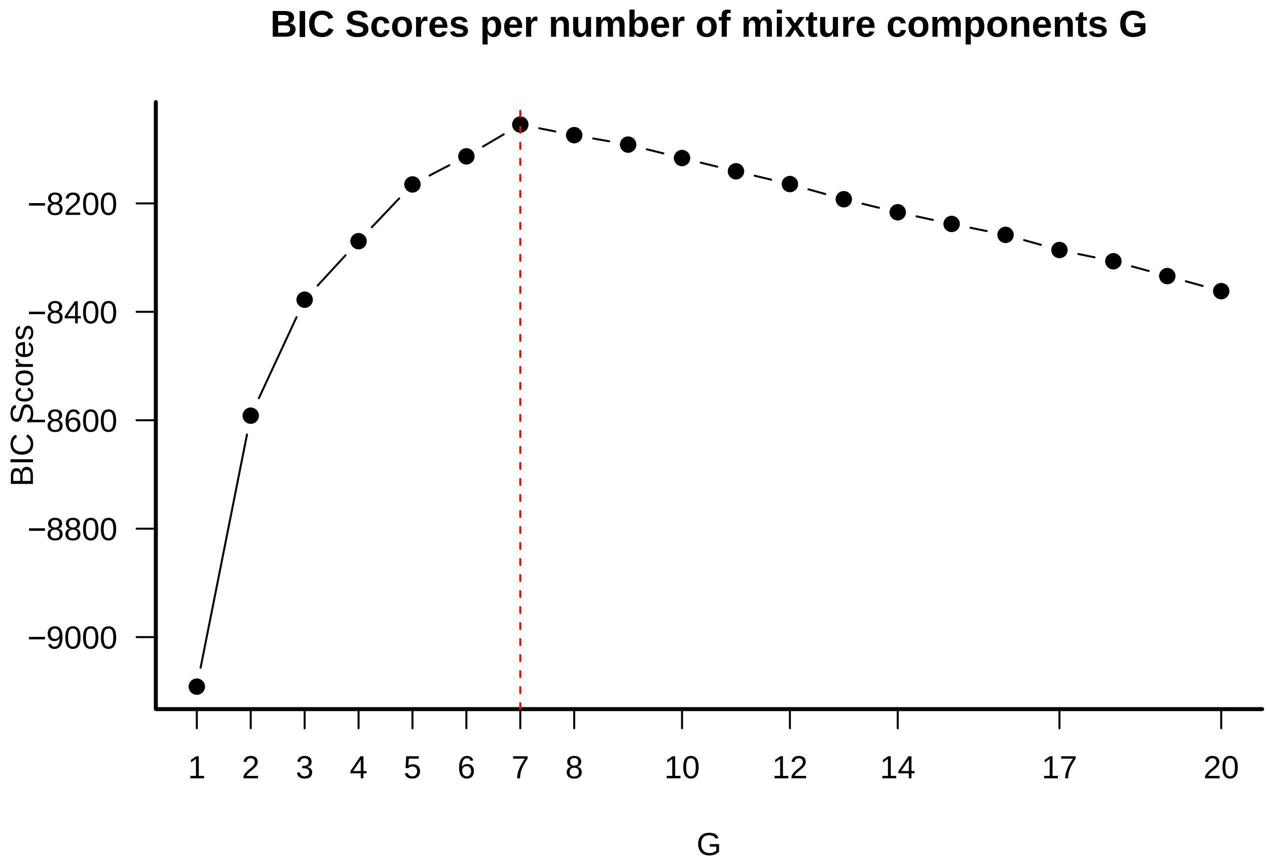


The Kmeans definitly picked up the distinct groups/clouds observed in Task 1. There are clear spaces between the cluster, which is very promising!

Task 8

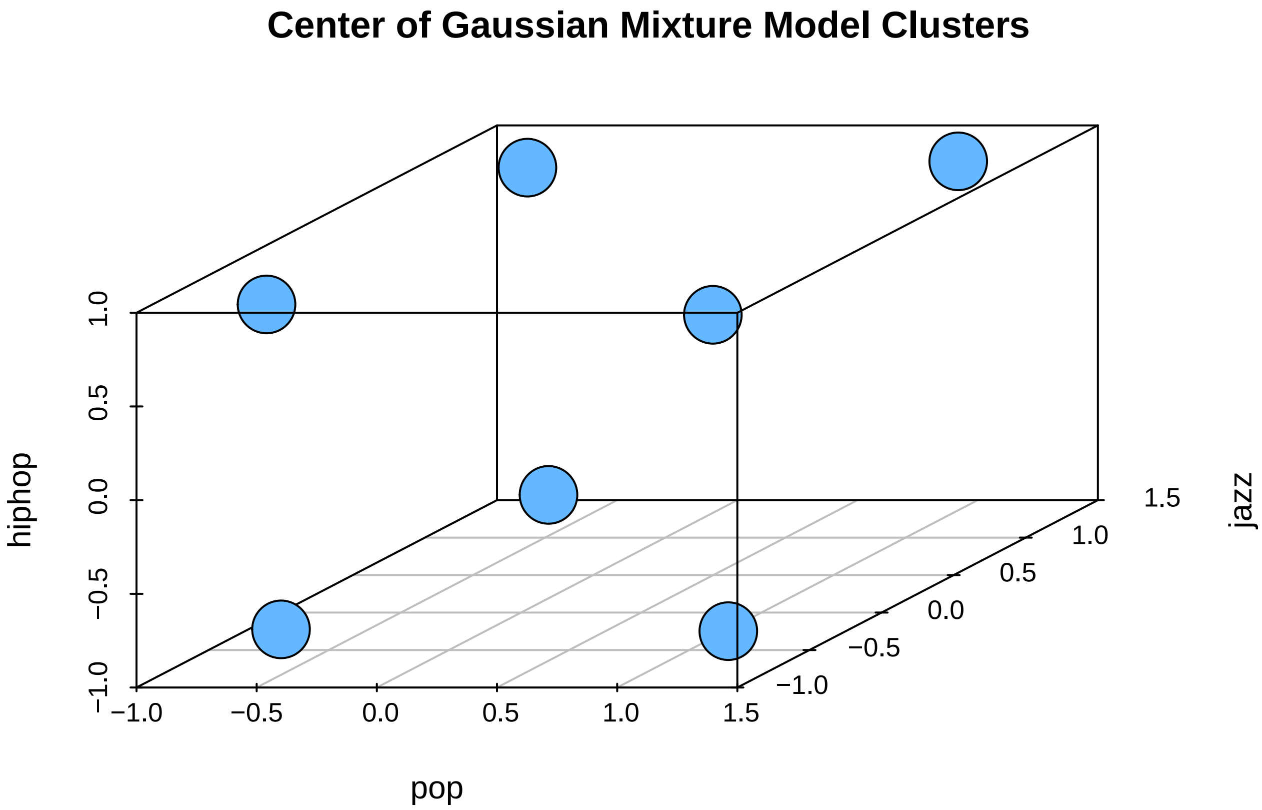
Repeat step 3–7 (but skip #5) using now instead a Gaussian mixture model. For this, you may use mclust's function Mclust() (specifying the number of components with the argument G). For #3: Note that, because this is a probabilistic model, we retrieve a likelihood score (or, more specifically BIC which is based upon the likehliood score but also penalizes for complexity) to measure its performance instead of total within cluster sum of squares (hint: you can extract the BIC by `$bic` on the model object). For #4, you can use `$parameters$mean` to extract the means/centroids of each cluster. For #6, you shall extract the hard cluster assignments (which you can do using '$classification´)

```r
set.seed(5)
g <- 1:20
bic <- c()

for(i in 1:length(k)){
  temp <- Mclust(data = dc, G = g[i])

  bic[i] <- temp$bic
}


tibble(k = 1:20,`BIC Scores` = bic) |>
plot(type = "b", bty = "n", pch = 20, cex = 1.5, las = 1,
     main = 'BIC Scores per number of mixture components G',
     xaxt = "n", xlab = "G", ylab =)
axis(1, at = c(seq(0, 8, 1),seq(10, 14, 2),17,20))
box("plot", bty = "l", lwd = 2)
abline(v = c(1:20)[which.max(bic)], col = "red", lty = 2)
```



The highest BIC is reached for 7 Gaussian Mixture Model Components.

```
gmm7 <- Mclust(data = dc, G = 7)

gmm7c <- gmm7$parameters$mean |> t() |> as.data.frame()

scatterplot3d(x = gmm7c$pop, y = gmm7c$jazz, z = gmm7c$hiphop,
              xlab = "pop", ylab = "jazz", zlab = "hiphop",
              main = "Center of Gaussian Mixture Model Clusters",
              type="p", pch = 21, bg = "steelblue1",
              cex.symbols = 4   ,angle = 36)
```



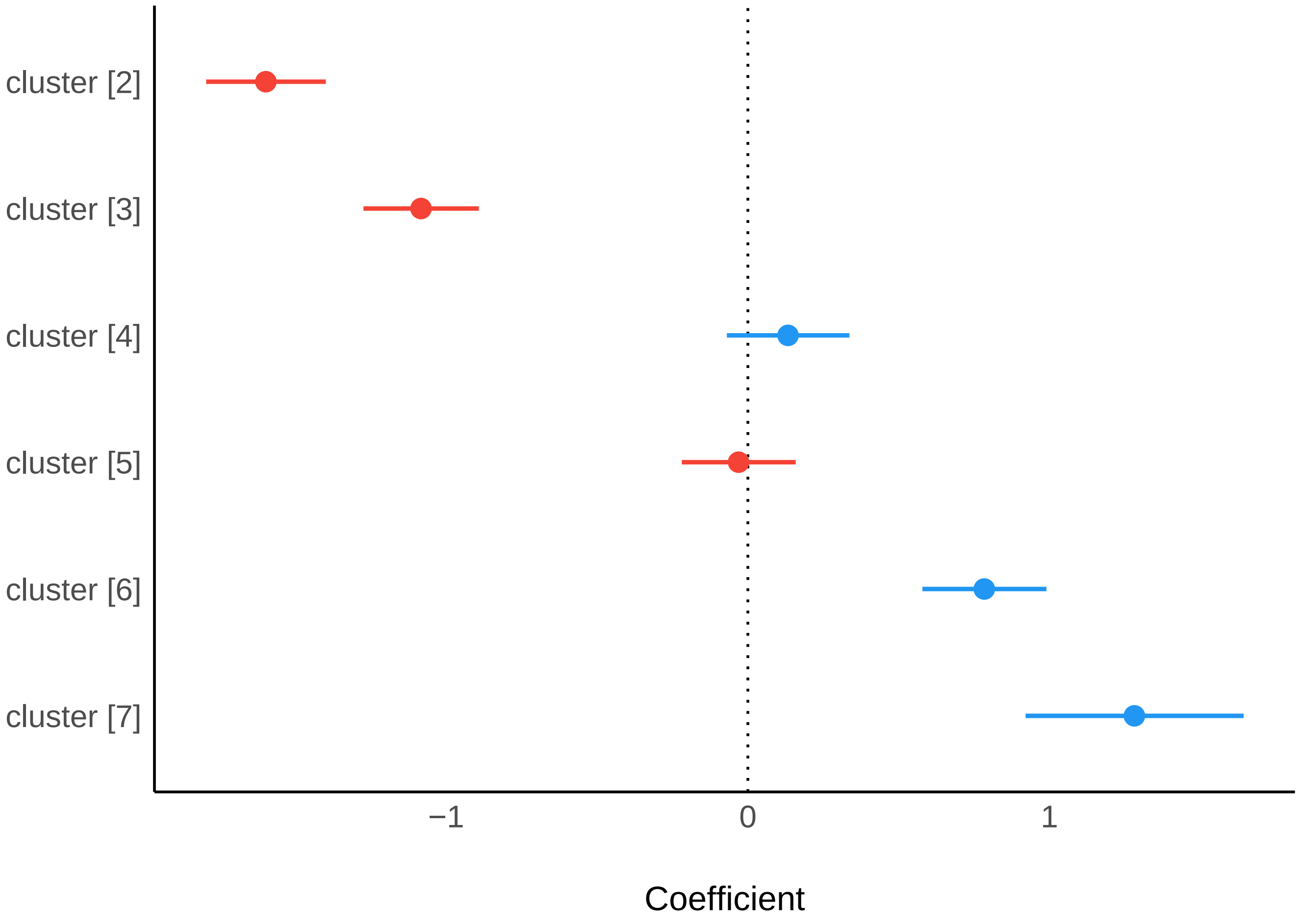
The clusters seem meaningfully distinct, since they are all located in distinct corners of the space of music-tastes. It seems like the GMM identified a cluster which has not been detected by kmeans before. There seem to be a few Jazz-fans that also like hiphop.

```
d$cluster <- gmm7$classification |> as.factor()

lm <- lm(influence ~ cluster, d)

lm |> report() |> summary() -> report
```

We fitted a linear model to predict influence with cluster. The model's explanatory power is substantial (R2 = 0.42, adj. R2 = 0.42). The model's intercept is at 1.80 (95% CI [1.66, 1.93]). Within this model:

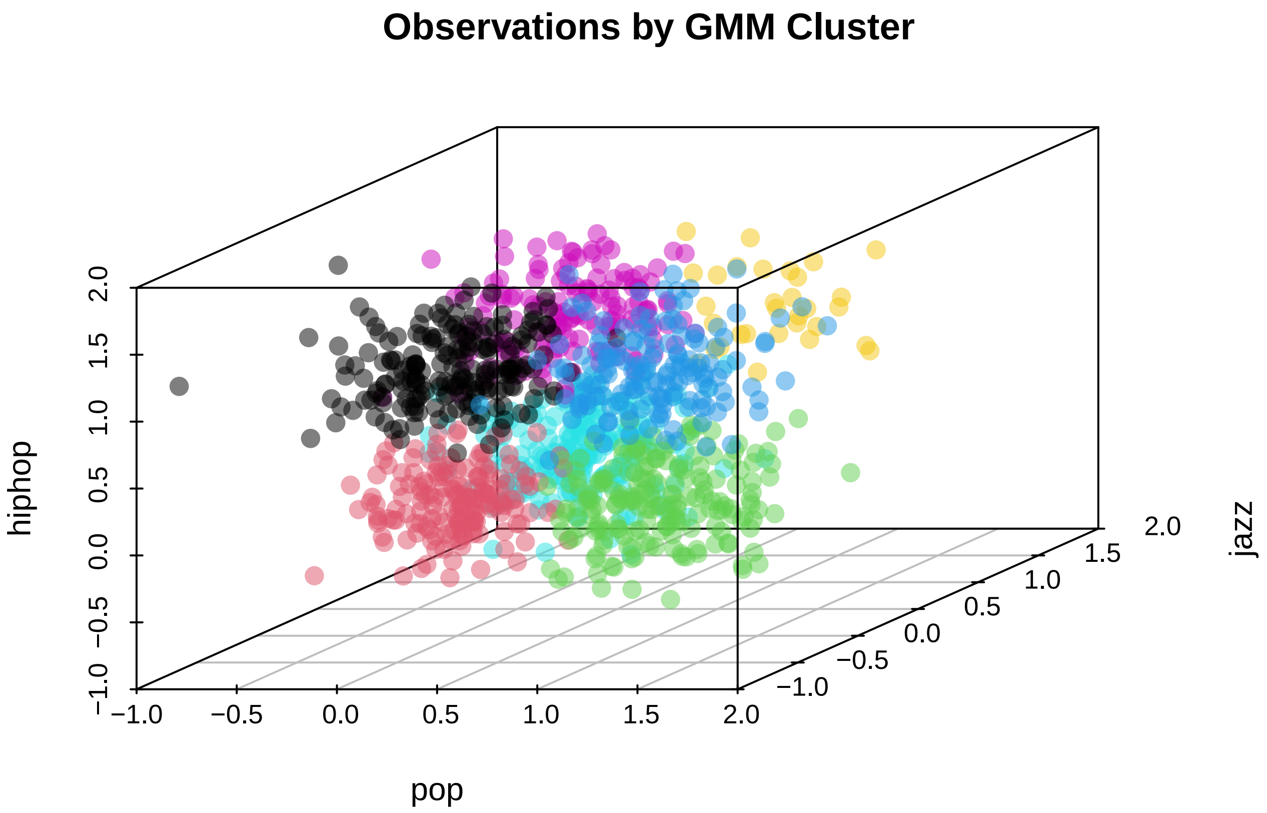
- The effect of cluster [2] is statistically significant and negative (beta = -1.60, 95% CI [-1.80, -1.40], t(1068) = -15.88, p < .001, Std. beta = -1.28) - The effect of cluster [3] is statistically significant and negative (beta = -1.08, 95% CI [-1.27, -0.89], t(1068) = -11.17, p < .001, Std. beta = -0.87) - The effect of cluster [4] is statistically non-significant and positive (beta = 0.13, 95% CI [-0.07, 0.34], t(1068) = 1.29, p = 0.197, Std. beta = 0.11) - The effect of cluster [5] is statistically non-significant and negative (beta = -0.03, 95% CI [-0.22, 0.16], t(1068) = -0.32, p = 0.748, Std. beta = -0.02) - The effect of cluster [6] is statistically significant and positive (beta = 0.78, 95% CI [0.58, 0.99], t(1068) = 7.51, p < .001, Std. beta = 0.63) - The effect of cluster [7] is statistically significant and positive (beta = 1.28, 95% CI [0.92, 1.64], t(1068) = 6.98, p < .001, Std. beta = 1.03)

```r
lm |> parameters() |> plot()
```



In comparison to cluster 1, cluster 6 and 7 are associated with an increase in influence. Cluster 2 and 3 are associated with a decrease in influence. Cluster 4 and 5 do not affect influence significantly compared to cluster 1.

```r
scatterplot3d(x = d$pop, y = d$jazz, z = d$hiphop,
              xlab = "pop", ylab = "jazz", zlab = "hiphop",
              main = "Observations by GMM Cluster",
              type="p", pch = 19, cex.symbols = 1.2, angle = 36,
              color = d$cluster |> adjustcolor(alpha.f = 0.5))
```

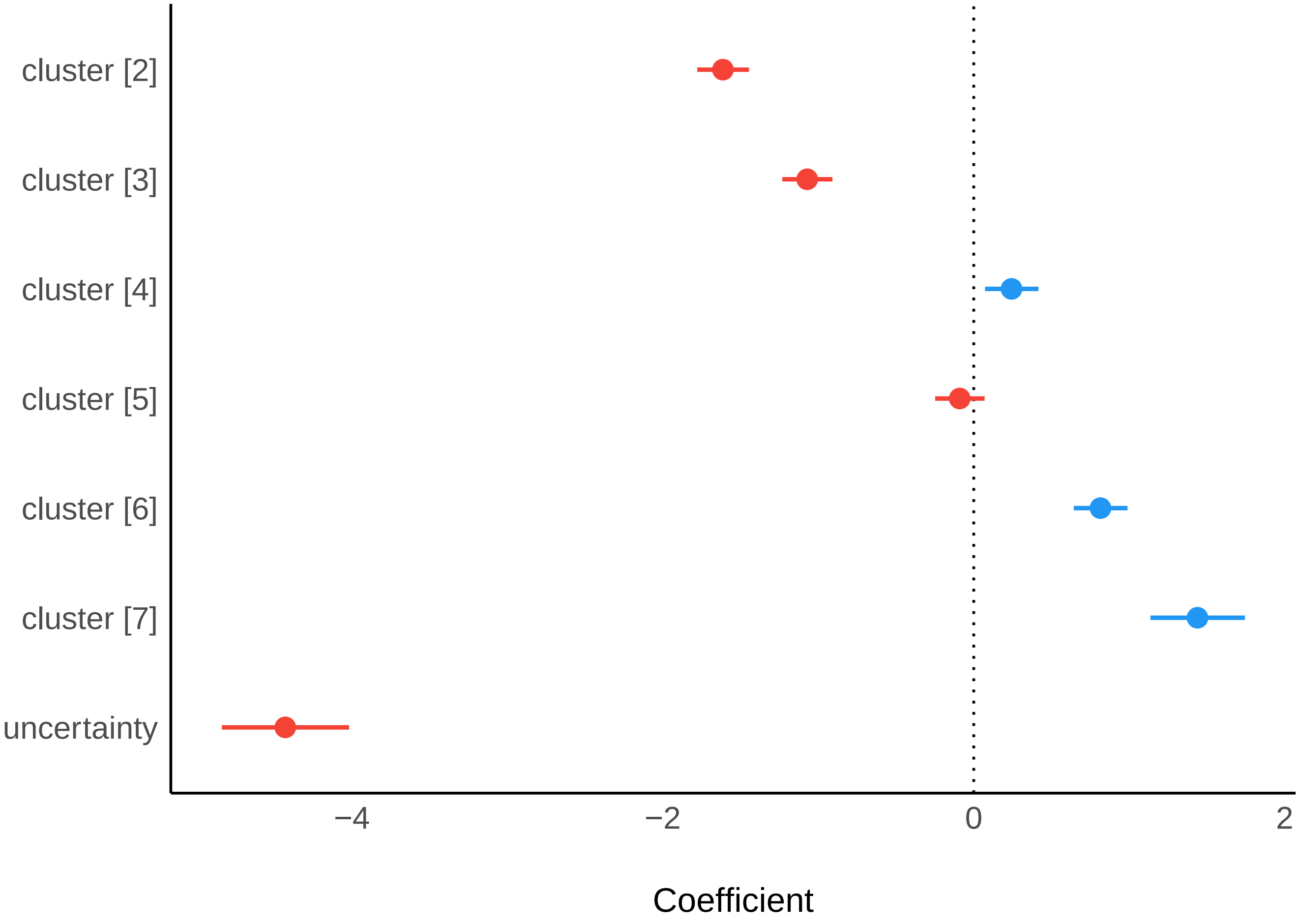


This plot highlights the few jazz-fans that also enjoy hiphop.

Task 9

Something which Mclust() also provides is a score for each observation how uncertain we are about its assignment. As mentioned during the lecture, "border-observations" can sometimes be substantively meaningful to study. You shall do so here. Extract the vector $uncertainity from the Gaussian mixture model fit, and store it in the original data. Then yet again fit a linear regression (together with the taste variables), but this time additionally with the uncertainity variable.

```
d$uncertainty <- gmm7$uncertainty

lm <- lm(influence ~ cluster + uncertainty, d)

lm |> parameters() |> plot()
```



It looks like uncertainty is highly negative associated with influence.

Part 2

```r
rm(list = ls())
```

In the second part of the lab, we will consider another simulated data set. This time, containing information about both the (fictive) individuals themselves, but also their social environments.

Task 1

Begin by importing the file "neighborhood.csv". Report the number of rows and columns of the data set, and make a brief note on the types of columns contained in it.

```r
d <- fread("neighborhood.csv")

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

Name	Value
Rows	200
Columns	25

```
names(d)
```

```
 [1] "taste_jazz"                "taste_classical"
 [3] "taste_blues"               "taste_pop"
 [5] "taste_country"             "taste_raegge"
 [7] "income"                    "nbhood_avg_income"
 [9] "education"                 "nbhood_avg_education"
[11] "nhood_crime"               "nbhood_unemployment"
[13] "nhbood_avg_temp"           "nhbood_pop"
[15] "nhbood_nr_lights"          "nhbood_nr_pizzerias"
[17] "city_avg_taste_jazz"       "city_avg_taste_classical"
[19] "city_avg_taste_blues"      "city_avg_taste_pop"
[21] "city_avg_taste_country"    "city_avg_taste_raegge"
[23] "taste_film_action"         "taste_film_romcom"
[25] "taste_film_documentary"
```

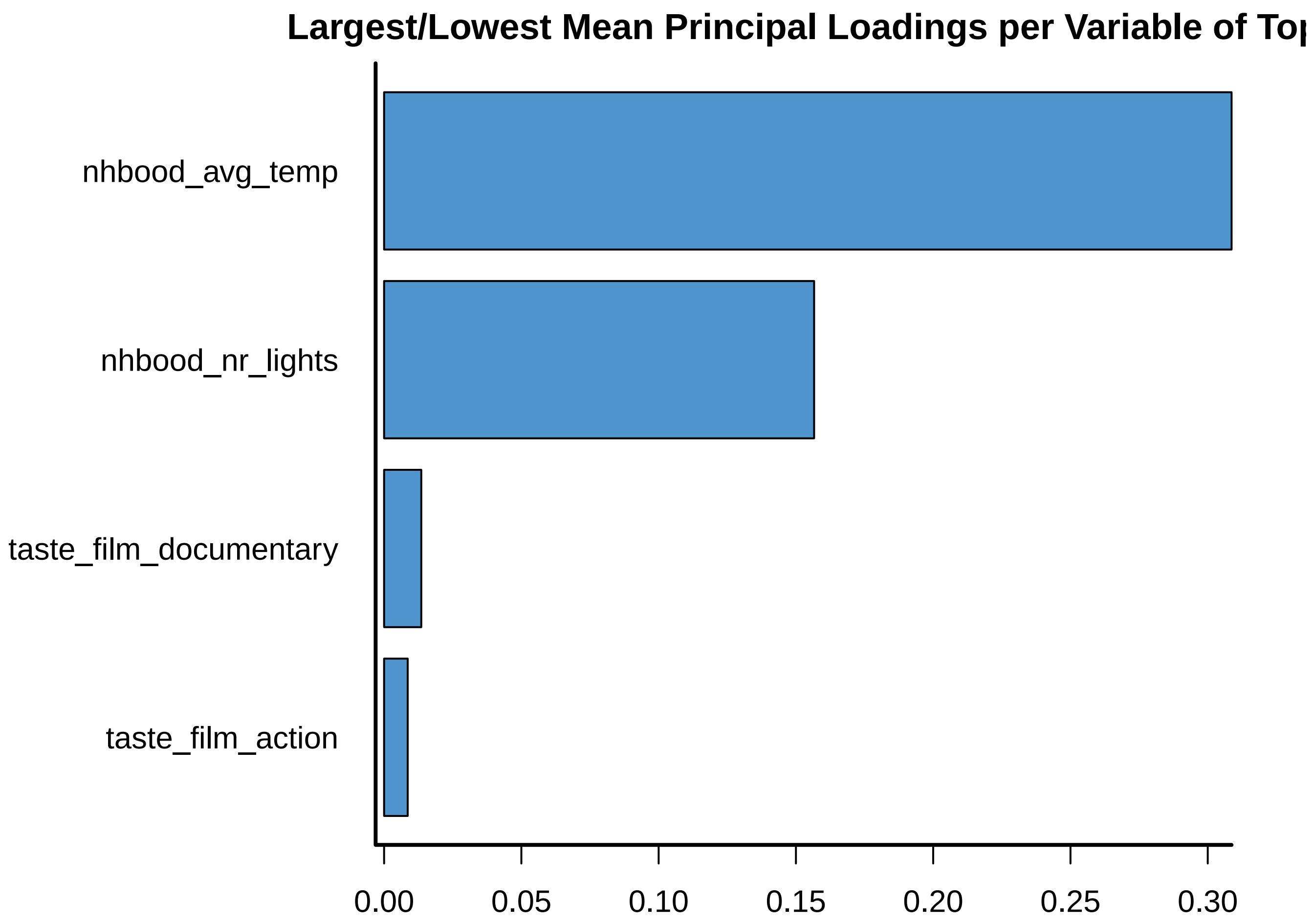
The dataset contains 200 observations at the individual level. It contains information about music and film taste, aswell as social demographic variables aggregated at a neihborhood level and the cities average taste.

Task 2

Based on the types of variables we find, we have some suspicion that there may exist considerable correlation between different variables in this data set. To explore whether we can capture key aspects of our data using fewer dimensions, we will use PCA and its extensions. Begin by estimating a principal components model without doing any standardization (hint: to estimate a PCA, use prcomp()). Why is this problematic (hint: examine the principal loadings)

```
headtail <- function(x, n = 2){
    c(head(x, n), tail(x, n))
}

pca <- prcomp(d)

par(mar = c(2, 10, 2, 2))
```

```r
apply(pca$rotation[,1:5],1,mean) |>
  abs() |>
  sort() |>
  headtail() |>
  barplot(horiz = T, las = 1,
          col = "steelblue3",
          main = "Largest/Lowest Mean Principal Loadings per Variable of Top")
box("plot", bty = "l", lwd = 2)
```

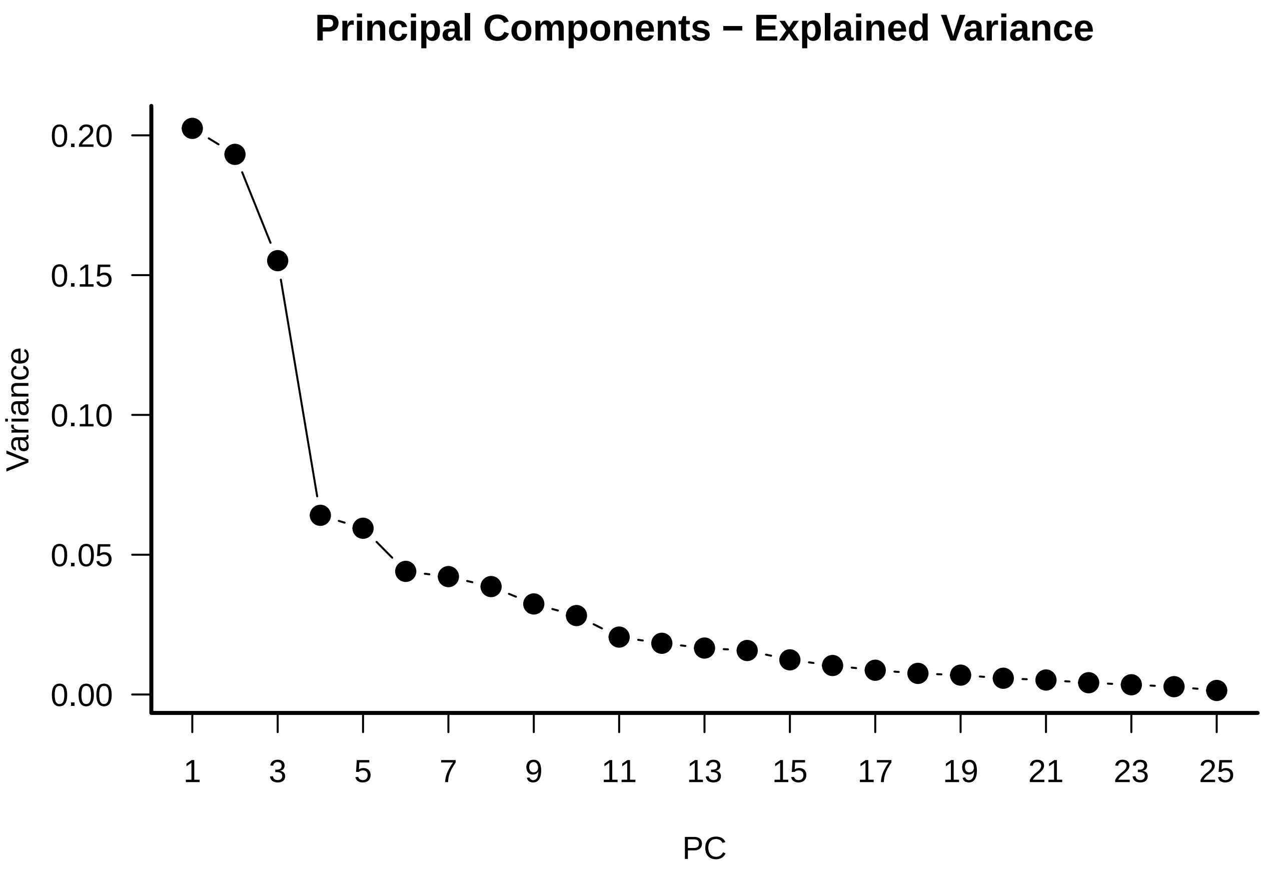


When inspecting the principal loadings it becomes obvious that the neighborhoodvariables and taste variables are measured on very different scales.

Task 3

Now, standardize your data, and then fit a PCA on this standardized data set. Plot the proportion variance explained. Interpret and decide on an appropriate number of principal components.

```r
pca <- d |>
  prcomp(scale. = T)

summary(pca)$importance[2,] |>
plot(type = "b", bty = "n", pch = 20, cex = 2, las = 1,
     main = 'Principal Components - Explained Variance',
     xaxt = "n", xlab = "PC", ylab = "Variance")
axis(1, at = seq(1, 25, 2))
box("plot", bty = "l", lwd = 2)
```

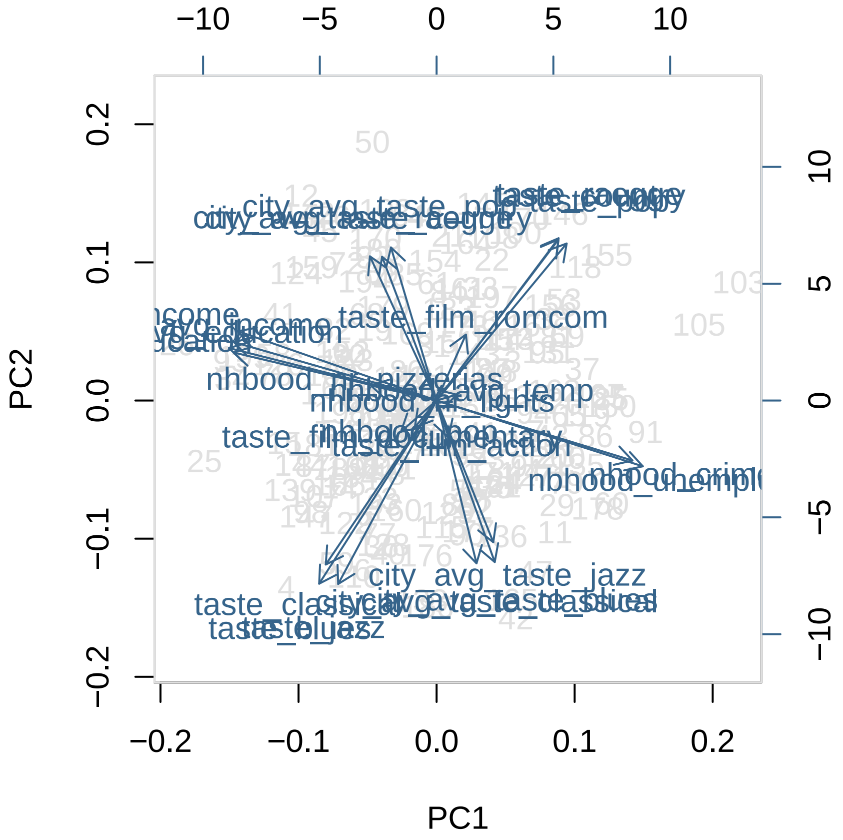


The first dimension explains ~20% of the total variance. From the variance that is still left to explain the second principal component is again explaining 20%, which is very strong. After the third principal component the explained variance falls off strongly.

Task 4

Interpret the retrieved principal components based on their loadings. Do they provide easy and substantively expected interpretations?

```r
biplot(pca,choices = c(1,2),
       col = c("gray88","steelblue4"))
```



```r
summary(pca)$rotation[,c(1,2,3)] |>
  round(2) |>
  kable()
```

	PC1	PC2	PC3
taste_jazz	-0.17	-0.32	0.19
taste_classical	-0.19	-0.28	0.22
taste_blues	-0.20	-0.32	0.17
taste_pop	0.22	0.27	-0.19
taste_country	0.21	0.28	-0.21
taste_raegge	0.21	0.28	-0.21
income	-0.33	0.12	-0.20
nbhood_avg_income	-0.34	0.10	-0.18
education	-0.34	0.08	-0.21
nbhood_avg_education	-0.35	0.09	-0.18
nhood_crime	0.33	-0.10	0.22
nbhood_unemployment	0.35	-0.11	0.19
nhbood_avg_temp	0.03	0.01	-0.05
nhbood_pop	-0.04	-0.05	-0.01
nhbood_nr_lights	-0.01	0.00	0.02
nhbood_nr_pizzerias	-0.11	0.03	0.06
city_avg_taste_jazz	0.10	-0.24	-0.31

	PC1	PC2	PC3
city_avg_taste_classical	0.07	-0.28	-0.29
city_avg_taste_blues	0.10	-0.28	-0.27
city_avg_taste_pop	-0.08	0.26	0.32
city_avg_taste_country	-0.09	0.25	0.30
city_avg_taste_raegge	-0.11	0.25	0.29
taste_film_action	0.02	-0.06	0.00
taste_film_romcom	0.05	0.11	-0.01
taste_film_documentary	-0.06	-0.05	0.03

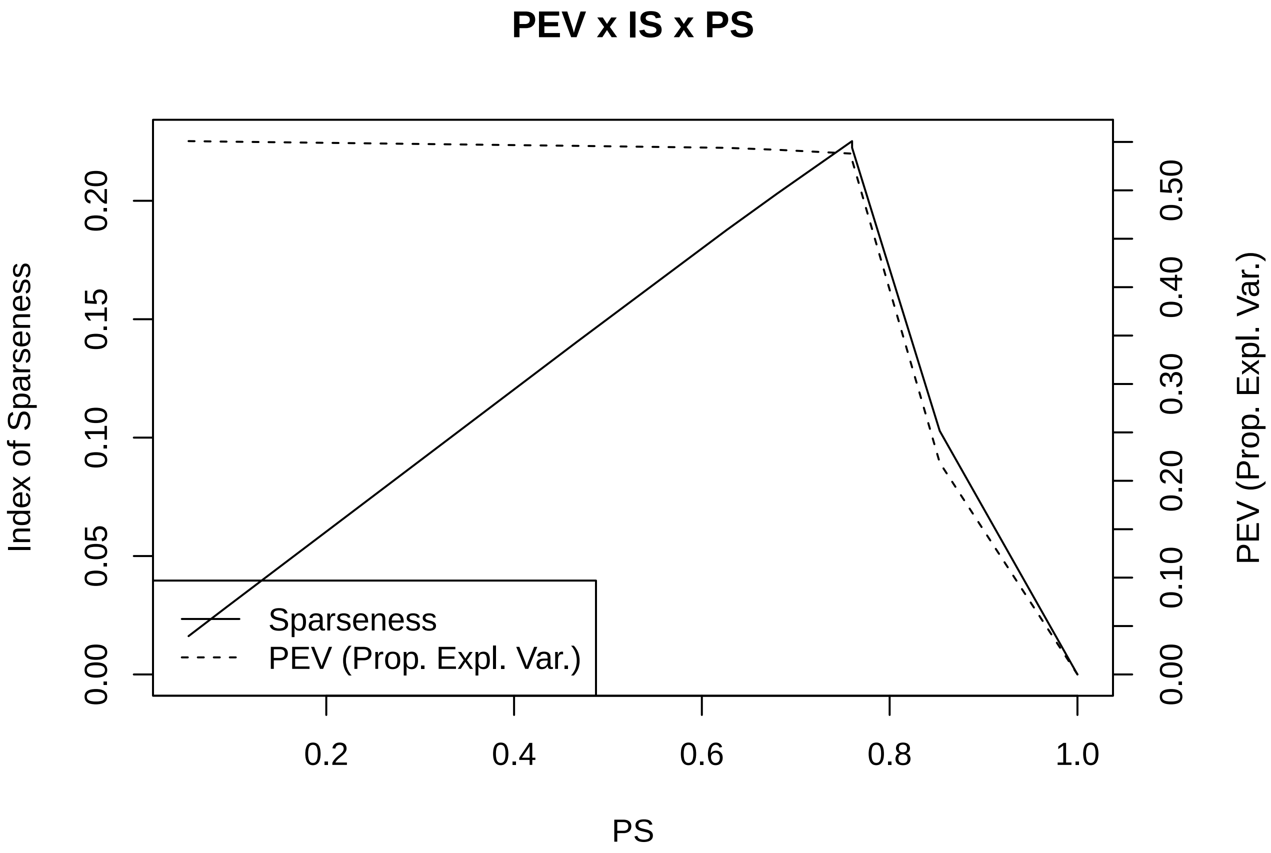
The first principal component captures mainly the distinction between neighborhoods of priviliged social economic standard and low social economic standard. the second axis captures the taste-distinction between popular and classical/jazz music at the individual and city level. This is also captured by the third axis. A lot of the variables end up in the middle without much contribution to any of the axis.

Task 5

Because of the conclusions in #4, we will now consider the sparse PCA. Use the same number of principal components that you did for the standard PCA in #2. In comparison to the standard PCA, we have an additional parameter λ in the sparse PCA. Use the IS index to determine an appropriate λ. Inspect the principal loadings for the resulting configuration. Interpret each dimension. Which do you think was easier to interpret; the sparse PCA or the standard PCA? Are there any downsides to sparse PCA?

```r
lasso <- c(0,1,5,10,20,80,100,200,500,1000)
spca_pevs <- c() # explained variance by axis
spca_ps <- c()   # sparseness

for(i in 1:length(lasso)){
  temp <- spca(x = d |> scale(), K = 3,
               type = 'predictor',
               para = c(rep(lasso[i],ncol(d))),
               sparse = 'penalty')
  # summed percentage of explained variance by axis
  spca_pevs[i] <- sum(temp$pev)
  spca_loadings <- temp$loadings # var x PC1 & PC2
  # sparseness
  ps <- length(spca_loadings[abs(spca_loadings)<=0.01]) / length(spca_loadings)
  spca_ps[i] <- ps
  if(length(ps)==0){ps <- 0}
}

is_pev_dt <- data.table(lasso=lasso,
                        spca_PEV=spca_pevs,
                        spca_PS=spca_ps)
# how much variance can be explained by axis without penalty
standard_PCA_PEV <- is_pev_dt[lasso==0]$spca_PEV
```

```r
# calculate Index of Sparseness
is_pev_dt[,IS:=standard_PCA_PEV * spca_PEV * spca_PS]
par(mar = c(5, 5, 5, 5))

# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

with(is_pev_dt, {
  plot(spca_PS, IS, type="l", xlab="PS",
       ylab="Index of Sparseness",
       main = "PEV x IS x PS")
  par(new=TRUE)
  plot(spca_PS, spca_PEV, type="l", lty=2,
       axes=FALSE, xlab="", ylab="")
  axis(side=4, at=pretty(spca_PEV, 8))
  mtext("PEV (Prop. Expl. Var.)", side=4, line=3)
  legend("bottomleft", legend = c("Sparseness", "PEV (Prop. Expl. Var.)"),
       lty = c(1, 2), col = c("black", "black"))
})
```



```r
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

spca_final <- spca(x = d |> scale(),
                   K = 3,
                   type = 'predictor',
                   para = rep(80, ncol(d)),
                   sparse = 'penalty')

spca_final$loadings |>
  as.data.frame() |>
  data_filter("PC1|PC2|PC3 != 0") |>
  round(2) |>
  kable()
```

	PC1	PC2	PC3
taste_jazz	0.00	-0.42	0.00
taste_classical	0.00	-0.38	0.00
taste_blues	0.00	-0.42	0.00
taste_pop	0.00	0.40	0.00
taste_country	0.00	0.41	0.00
taste_raegge	0.00	0.42	0.00
income	-0.43	0.00	0.00
nbhood_avg_income	-0.38	0.00	0.00
education	-0.40	0.00	0.00
nbhood_avg_education	-0.39	0.00	0.00
nhood_crime	0.38	0.00	0.00
nbhood_unemployment	0.45	0.00	0.00
city_avg_taste_jazz	0.00	0.00	-0.40
city_avg_taste_classical	0.00	0.00	-0.40
city_avg_taste_blues	0.00	0.00	-0.41
city_avg_taste_pop	0.00	0.00	0.46
city_avg_taste_country	0.00	0.00	0.39
city_avg_taste_raegge	0.00	0.00	0.39

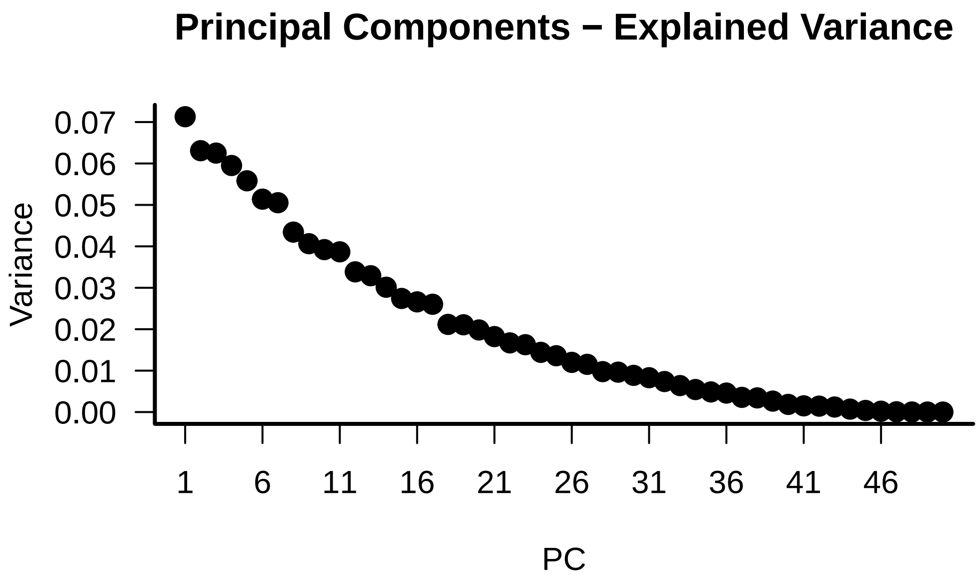
The sparse PCA is way easier to interpret, since many the variables that are not contributing meaningfully to an axis are set to 0. This is great for isolating signal, but if the goal is to keep as much explained variance as possible it could be seen as a downside.

PC1 captures the distinction between the socioeconomic status of the neighborhoods of the observed individuals. PC2 captures the distinction of an individuals musical taste between more jazz, classical music and blues oriented people versus pop, country and reagge oriented people. PC3 captures the city level variables while making a distinction between the same genres that were distinctive at an individual level.

Task 6

As a last exercise for today, you shall simulate your own data. Generate a dataset of 50 observations and 50 independent variables using the function provided below. Once you have generated the data, process your data as you did above for the neighborhood data set (standardize, making into a matrix). Then, estimate a standard PCA. What do you find: could the PCA help us effecively redue the dimensionality of our data or not? Why?

```r
gen_data <- function(n,p){
  df <- c()
  for(i in 1:p){
    ith_var <- rnorm(n = n, mean = 0, sd = 1)
    df <- cbind(df,ith_var)
  }
  return(df)
}

sim <- gen_data(50,50) |>
  scale() |>
  as.matrix()


pca <- prcomp(sim)

summary(pca)$importance[2,] |>
  plot(type = "b", bty = "n", pch = 20, cex = 2, las = 1,
       main = 'Principal Components - Explained Variance',
       xaxt = "n", xlab = "PC", ylab = "Variance")
  axis(1, at = seq(1, 50, 5))
  box("plot", bty = "l", lwd = 2)
```



The pca could not help reduce the dimensionality, since the simulated data is just a 50 dimensional cloud of perfectly normal distributed points. Since there are no meaningful relationships between the variables, the PCA is not able to find a meaningful axis containing the most variance/information in the high dimensional cloud of points.

Quiz

1. Which of the following are true about the relation between supervised learning and unsupervised learning (1p):

   a. The task of assigning observations to predefined categories is exclusive to supervised learning.

   b. Discovery of previously unknown patterns/relations is exclusive to unsupervised learning.

   c. Supervised learning problems have a ground-truth. Unsupervised learning problems do not.

   d. The problem of models picking up noise and spurious patterns in data is exclusive to supervisedlearning.

2. We can use PCA on a matrix X in order to:

   a. Discover latent organizations of the variables in X.

   b. Partition observations into distinct clusters.

   c. Predict some outcome variable Y .

   d. Reduce the dimensionality of X.

3. When using a quantitative approach to select the number of clusters (or dimensions) in unsupervised learning, which two competing forces do we usually seek to balance? > First the amount of variance accounted for by cluster and second the complexity degree (number of clusters).

4. For which of the following scenarios do you have a good reason to make a choice—about the number of clusters/dimensions—that contradicts the decision based on the so-called elbow criterion:

   a. You have no domain knowledge and no hypothesis; you are just interested in exploring the data.

   b. You have substantial domain knowledge and a clear hypothesis.

   c. You do not care about interpretability. Your goal is to use the principal scores in an supervised learning model to predict some outcome as well as possible.

   d. Your purpose for using PCA is to visualize your data.