Brief Intro To Clustering in R (And Also R Markdown)

Khoi Trinh

2023-02-22

Before working with R markdown, we need a few packages

```
# The rmarkdown package
#install.packages('rmarkdown')
```

If you want to generate pdf files, you will need to install LaTeX

If you don't plan on using LaTeX anywhere outside of R markdown, I suggest TinyTex

```
#install.packages('tinytex')
#tinytex::install_tinytex()
```

Note that you can and should run these above commands in the RStudio console

Let's Explore Clustering

```
library(cluster)
library(NbClust)
library(factoextra)
library(dplyr)
library(kmed)
```

First, we need some data

Data description

The data I chose is my own Spotify streaming history for the past year; you can find how to get your own Spotify data here

Then, follow these instructions to obtain the song traits.

Misc data processing:

Read in the data

```
spotify = read.csv("final.csv")
```

The data had almost 60,000 observations, out of those, only the numeric data will be considered.

And out of the numeric columns, we will drop columns 1, 2, 5, 7, 14, and 15 as they are not song traits.

```
numericData = spotify %>% #Add data
  dplyr::select(where(is.numeric)) #finds where is.numeric is true

# drop the mentioned columns
numericData <- subset(numericData, select=-c(1,2,5,7,14,15))
```

Time To Create Some Clusters

One last data processing step

Scale the data, and use that to create our clusters. We need to scale the data as most of the traits are < 1; but tempo are not

So scaling is needed to not skewed the clusters.

```
clusterData <- scale(numericData)
```

K means

For this clustering method, the `kmeans()` function from the stats package is used. Let's start with 2 clusters. Normally, there are ways to determine an optimal number of clusters, but for the sake of simplicity, let's stick to 2, maybe we can change it later.

```
kmean <- kmeans(clusterData,2, nstart=10)
```

Clusters Analysis

First, let's see the size of the clusters

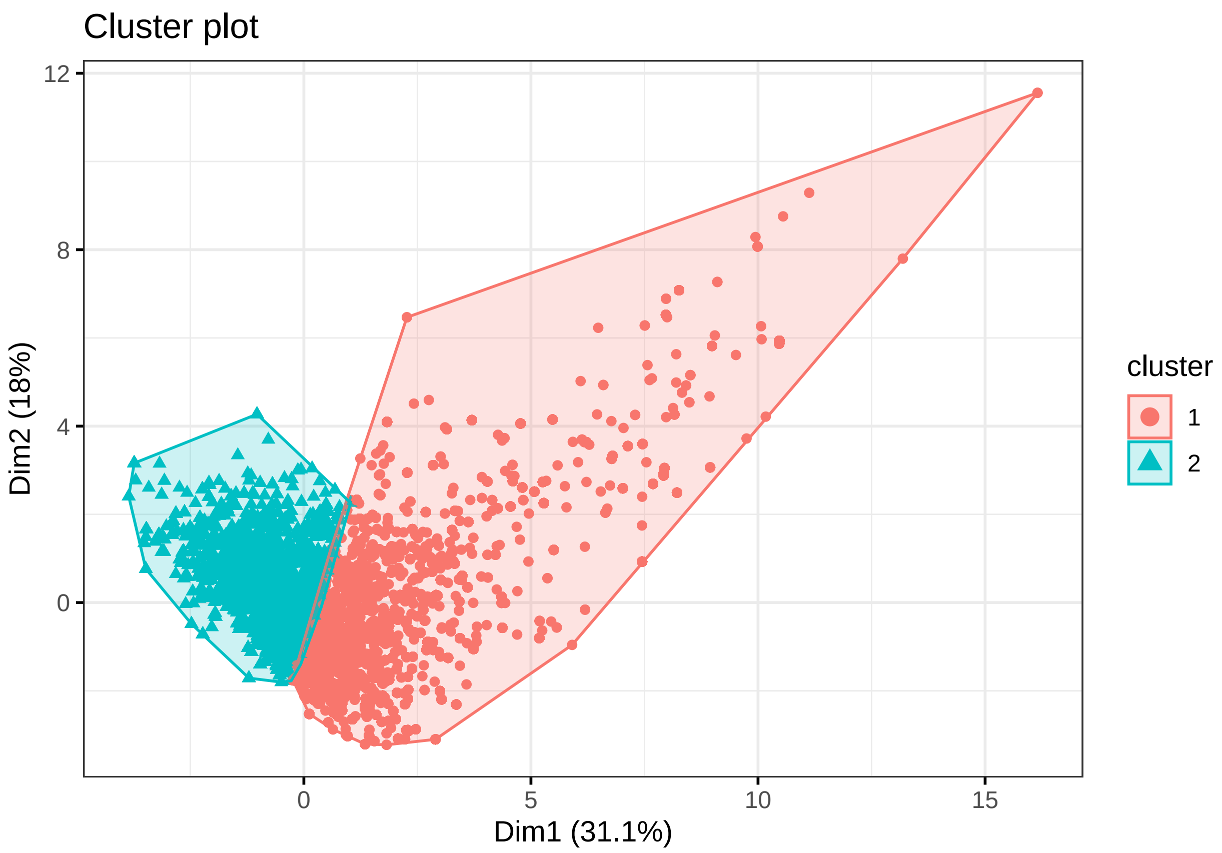
```
kmeansize <- kmean$size
kmeansize
```

```
## [1] 20702 35508
```

We have 2 clusters, with size 36053 and 20157.

Visually, the clusters look like so. We can see that the clusters have a little bit of overlap, but overall, it looks good.

```
fviz_cluster(kmean, data = clusterData, geom = "point", ellipse.type = "convex",
             ggtheme = theme_bw())
```



Cluster Intepretation

```
kmeaninfo <- data.frame(kmean$centers, kmean$size)
kmeaninfo
```

```
##   danceability     energy   loudness speechiness acousticness instrumentalness
## 1    0.7219741 -0.7392529 -0.5464510  -0.6329388     0.429137       0.07646846
## 2   -0.4209279  0.4310019  0.3185938   0.3690182    -0.250197      -0.04458292
##     liveness    valence      tempo kmean.size
## 1 -0.3635405  0.6408118 -0.4443888      20702
## 2  0.2119527 -0.3736084  0.2590891      35508
```

Here are the explanation of the traits.

acousticness — how acoustic

danceability — self-explanatory

energy — how ‘fast, loud an noisy’

instrumentalness — the less vocals, the higher

liveness — whether there is audience in the recording

loudness — self-explanatory

speechiness — the more spoken words, the higher valence — whether the track sounds happy or sad

tempo — the bpm

We can see that cluster 1 have songs that are mixed louder, have higher energy and liveness, but lower valence

Maybe we can call this cluster “Sad Workout Songs”?

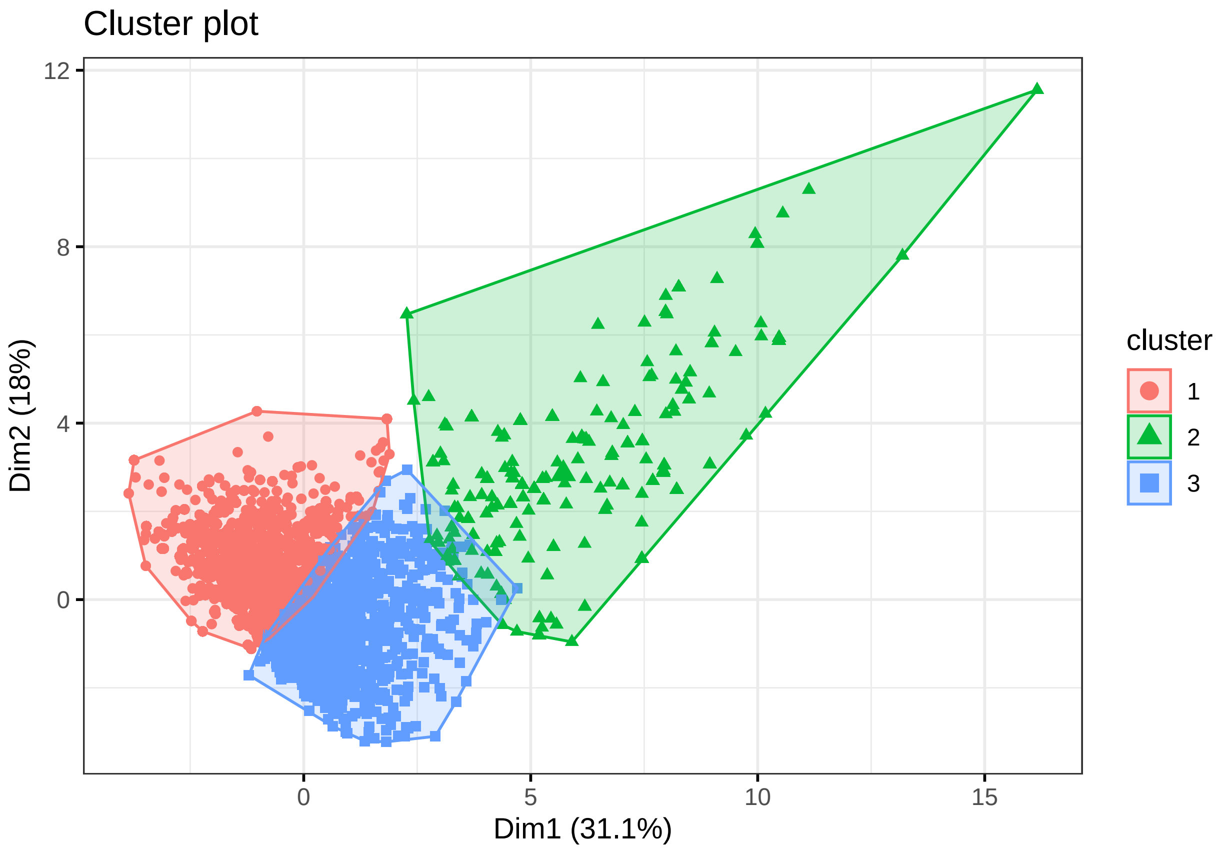
Cluster 2 have songs that are higher in valence, acousticness, danceability, but lower energy(?)

Maybe we can call this cluster “Happy Coffehouse Acoustic Songs To Dance To”???

Let’s increase the number of clusters, to see if we get any more clear separation.

```
kmean2 <- kmeans(clusterData,3, nstart=10)
```

```
fviz_cluster(kmean2, data = clusterData, geom = "point", ellipse.type = "convex",
             ggtheme = theme_bw())
```



```
kmeaninfo2 <- data.frame(kmean2$centers, kmean2$size)
kmeaninfo2
```

```
##   danceability     energy    loudness speechiness acousticness instrumentalness
## 1   -0.6305388  0.4235014  0.26672088   0.5692154  -0.24421672       0.05079510
## 2    0.2863125 -3.7874760 -2.91568780  -0.8446513   4.35916164       0.53308803
## 3    0.5825141 -0.1484980 -0.05771523  -0.4862819  -0.06122882      -0.08447799
##     liveness    valence      tempo kmean2.size
## 1  0.3005374 -0.5807230  0.4546128       26534
## 2 -0.3793901 -0.2876731 -0.5326758        1877
## 3 -0.2612448  0.5737209 -0.3979591       27799
```

We can do the same analysis as with the 2 clusters. See if you can try it for yourself!