

# 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.

```
numericData = spotify %>% #Add data  
  dplyr::select(where(is.numeric)) #finds where is.numeric is true  
#numericData <- head(numericData, 10000) # gets only 10000 observations  
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.

```
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

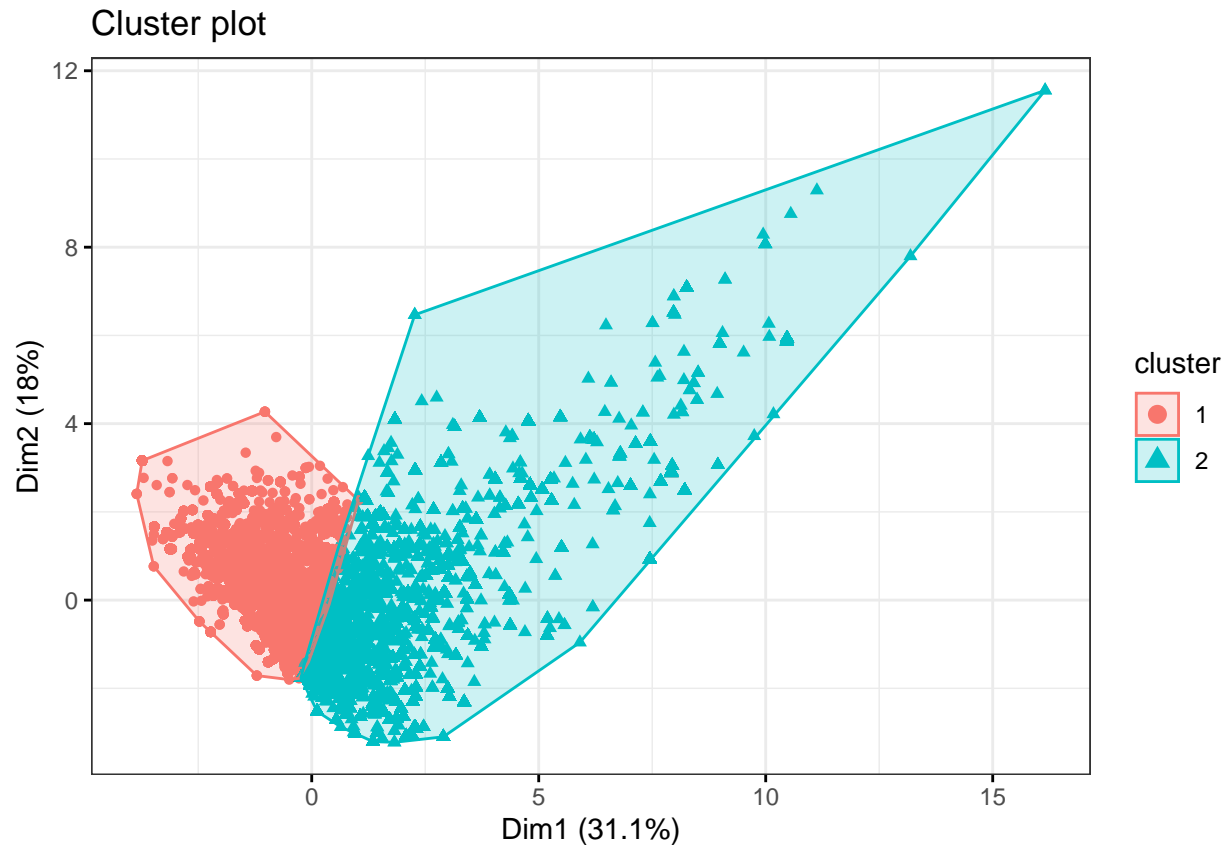
```
kmeansize <- kmean$size  
kmeansize
```

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

We have 2 clusters, with size 36053 and 20157.

Visually, the clusters look like so

```
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 acoustictness instrumentallness
## 1  -0.4209279  0.4310019  0.3185938  0.3690182    -0.250197    -0.04458292
## 2   0.7219741 -0.7392529 -0.5464510 -0.6329388     0.429137     0.07646846
##   liveness    valence    tempo kmean.size
## 1  0.2119527 -0.3736084  0.2590891     35508
## 2 -0.3635405  0.6408118 -0.4443888     20702
```

Here are the explanation of the traits.

acoustictness — how acoustic

danceability — self-explanatory

energy — how 'fast, loud an noisy'

instrumentallness — the less vocals, the higher

liveness — whether there is audience in the recording

loudness — self-explanatory

speechiness — the more spoken words, the higher

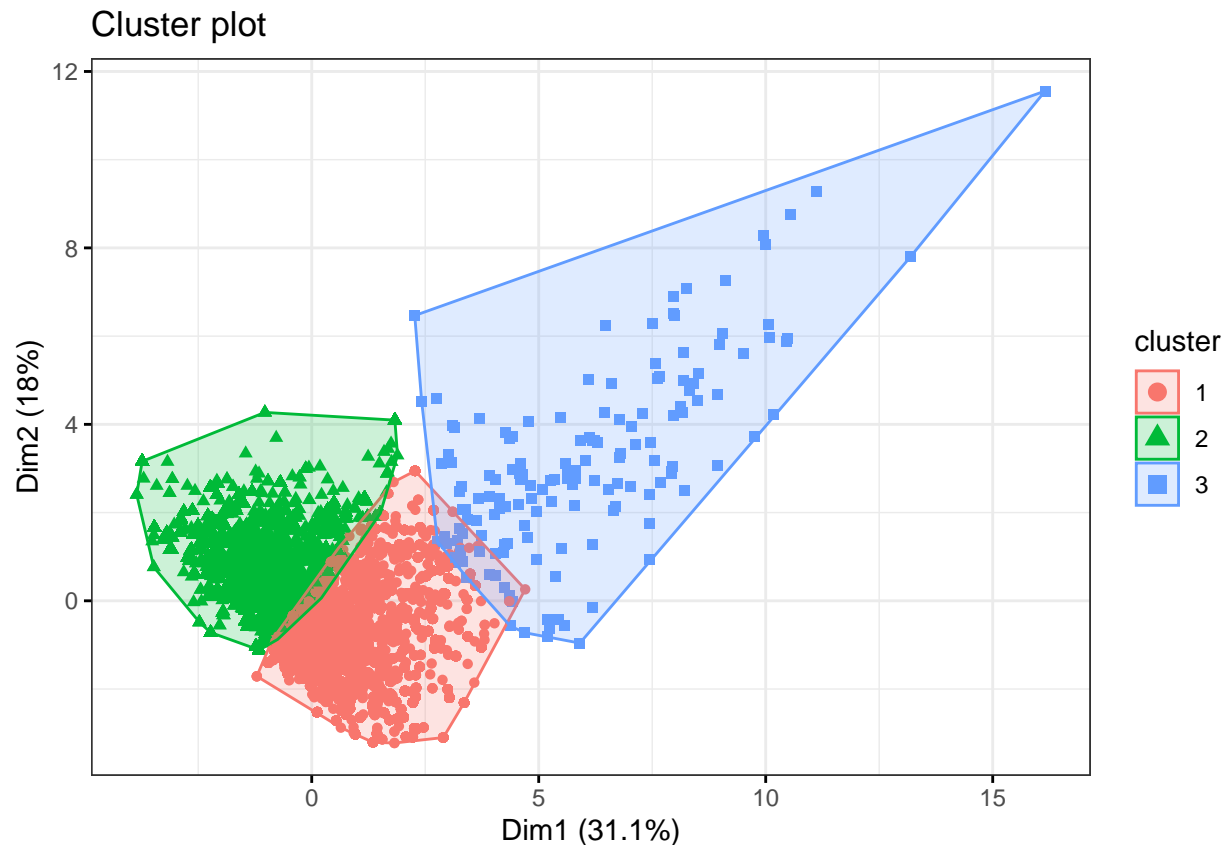
valence — whether the track sounds happy or sad

tempo — the bpm

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	0.5838525	-0.1482326	-0.05809338	-0.4858961	-0.0610521	-0.08428329
## 2	-0.6309338	0.4227500	0.26684764	0.5679374	-0.2442500	0.05047940
## 3	0.2863125	-3.7874760	-2.91568780	-0.8446513	4.3591616	0.53308803
##	liveness	valence	tempo	kmean2.size		
## 1	-0.2610920	0.5743555	-0.3983282	27777		
## 2	0.2999122	-0.5804305	0.4542926	26556		
## 3	-0.3793901	-0.2876731	-0.5326758	1877		