Anexo XI. Clustering con PAM (k-medoides) y clusters de clusters

Vamos a realizar clustering sobre los datos 'nc_data' con el método de clustering k-medoides o PAM.

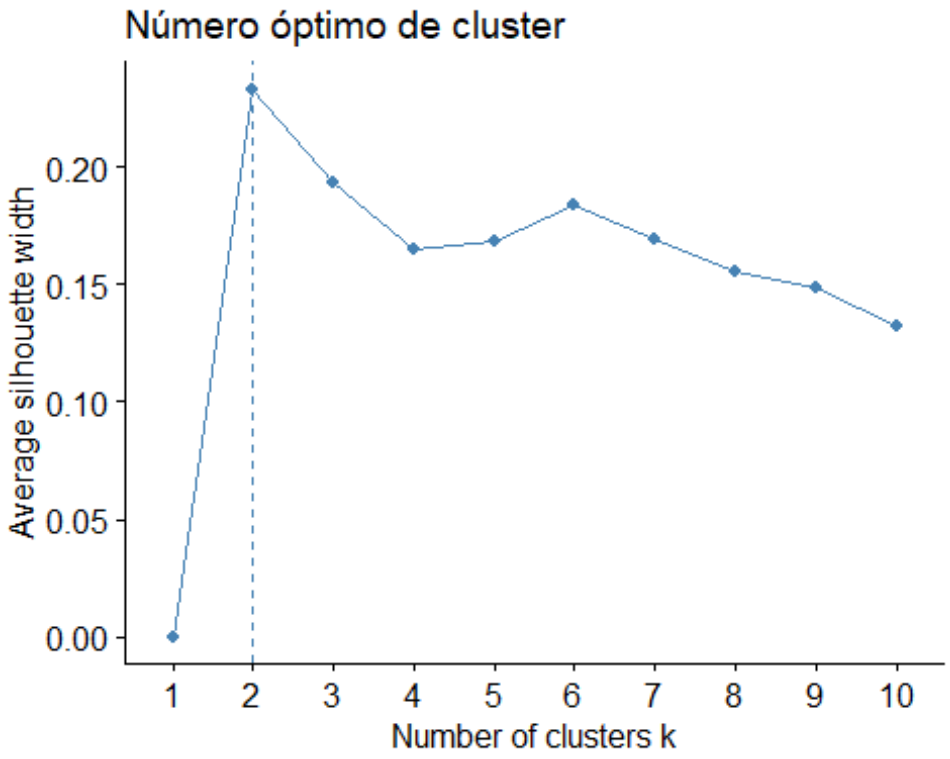
```
load('C:/Users/belan/Desktop/UNIVERSIDAD/2CD1/PROYECTO
II/DATOS/ncdata.RData')
load('C:/Users/belan/Desktop/UNIVERSIDAD/2CD1/PROYECTO
II/DATOS/desc_data.RData')
```

Tras múltiples intentos de clustering (Anexos VIII, IX) los resultados poco deseables han hecho que podamos comprender que existen demasiados factores para obtener un agrupamiento "limpio". Uno de ellos es la popularidad: al no estar restringiendo el rango de esta variable, muchas canciones poco relevantes (quizás por su combinación inusual de valores en los atributos relacionados con el sonido) pueden estar alterando o distorsionando nuestros intentos de agrupamiento.

Es por eso por lo que a partir de ahora vamos a trabajar con dos nuevos filtros:

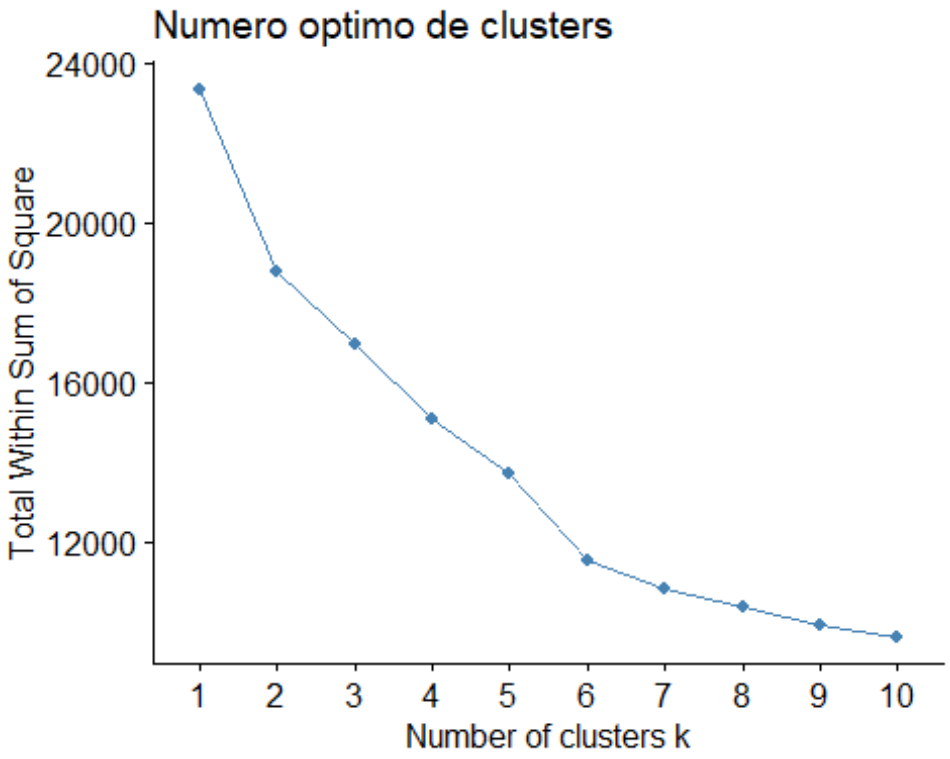
- Las canciones deben estar en 30 segundos y 5 minutos: fragmentos de 10 segundos no son útiles para un recomendador, así como canciones pistas muy largas, que probablemente son podcasts, entrevistas o recopilaciones de muchas canciones.
- Las canciones deben tener un nivel de popularidad mayor o igual a 70, con el fin de reducir la variabilidad de nuestros datos.

```
ncdata_aux = ncdata
ncdata_aux = ncdata_aux[ncdata_aux$duration_ms > 18000,]
ncdata_aux = ncdata_aux[ncdata_aux$duration_ms < 300000,]
ncdata_aux = ncdata_aux[ncdata_aux$popularity > 70,]

ncdata2= ncdata_aux[,desc_data$type=='numerical']
View(ncdata2)

ncdata2 = ncdata2[,-c(3,8,10,12)]
ncdata2 = scale(ncdata2, center = TRUE, scale = TRUE)

fviz_nbclust(x=ncdata2, FUNcluster = pam, method = 'silhouette', k.max = 10,
verbose = FALSE) +
  labs(title = 'Número óptimo de cluster')
```



Como podemos observar en el gráfico anterior, el número óptimo de cluster son 2. Sin embargo, como este número de cluster es trivial, vamos a realizar un gráfico de la Suma de Cuadrados:

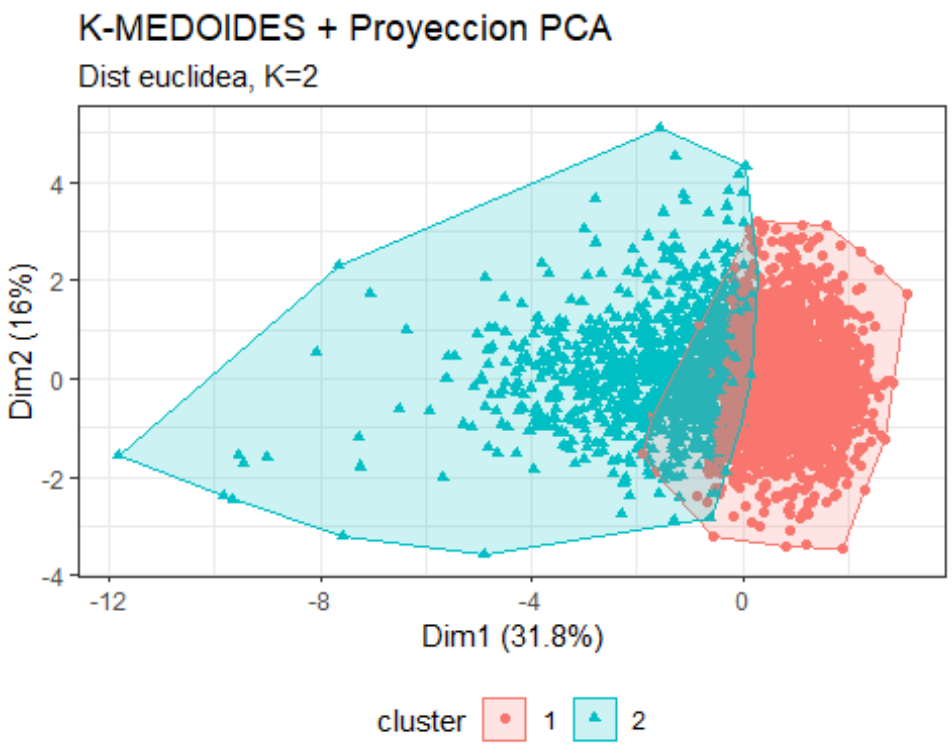
```
fviz_nbclust(x = ncdata2, FUNcluster = pam, method = "wss",
             k.max = 10, verbose = FALSE) +
  labs(title = "Numero optimo de clusters")
```



Ahora podemos afirmar que realizar 2 clusters tendría gran error. Teniendo en cuenta la orientación del codo (elbow), vamos a elegir 6 clusters, que tienen un gran coeficiente de Silhouette también.

No obstante, al realizar su correspondiente gráfico, pudimos observar que 6 clusters no era nada bueno: los clusters se solapaban unos encima de otros, así que vamos a rectificar y realizaremos 2 clusters.

```
cluster1 <- pam(ncdata2, k = 2)
table(cluster1$clustering)

##
##    1    2
## 1957  961

fviz_cluster(object = list(data=ncdata2, cluster=cluster1$clustering), stand
= FALSE,
             ellipse.type = "convex", geom = "point", show.clust.cent =
FALSE,
             labelsize = 8, axes = (1:2))  +
  labs(title = "K-MEDOIDES + Proyeccion PCA",
       subtitle = "Dist euclidea, K=2") +
  theme_bw() +
  theme(legend.position = "bottom")
```

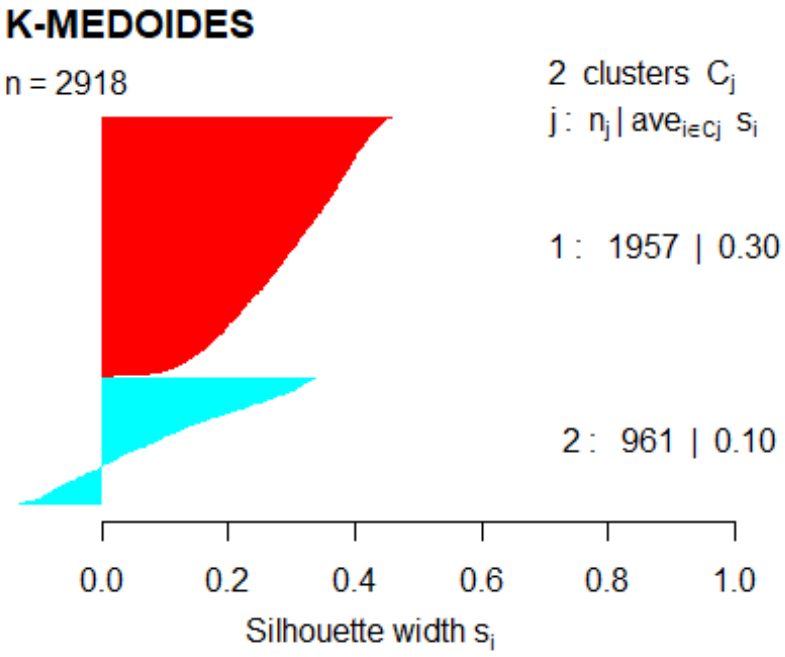



```
#View(desc_data)
#canciones = ncdata[which(ncdata$popularity > 70),]

##kable(ncdata_aux[which(cluster1$clustering == 1),c('artists','name')])
```

Con 2 clusters, vemos que hay unas pocas observaciones que se solapan pero se distinguen las dos nubes de puntos.

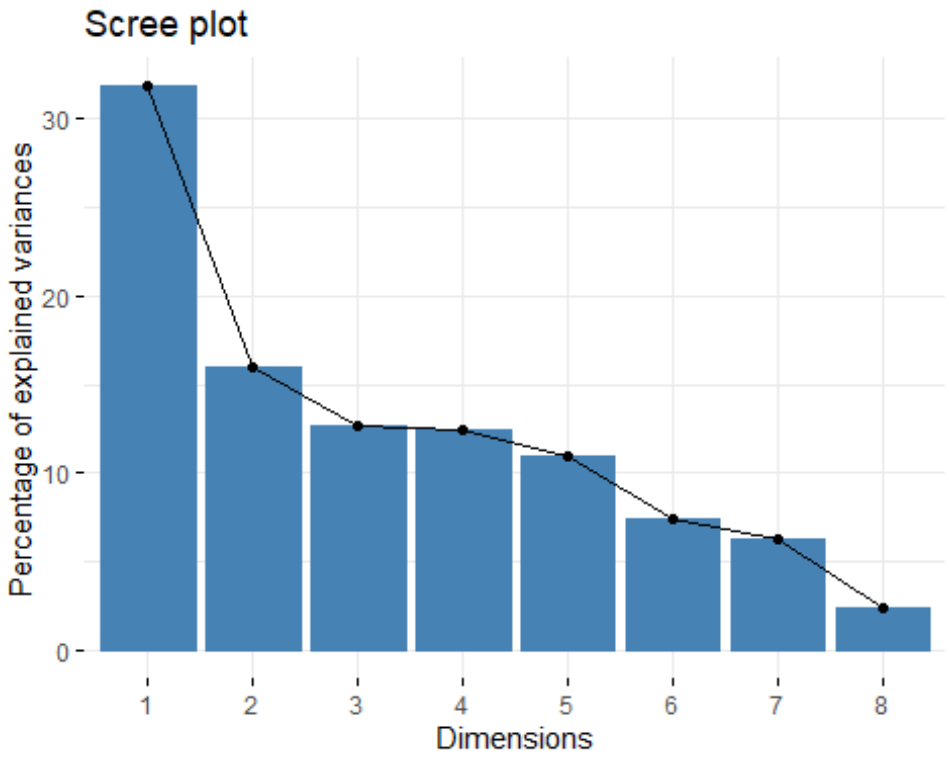
Como dentro de cada cluster tenemos una enorme cantidad de individuos, vamos a realizar otro clustering sobre el grupo 1, el cluster con más observaciones.

```
midist <- get_dist(ncdata2, stand = FALSE, method = "euclidean")
plot(silhouette(cluster1$clustering, midist), col=rainbow(2), border=NA, main
= "K-MEDOIDES")
```

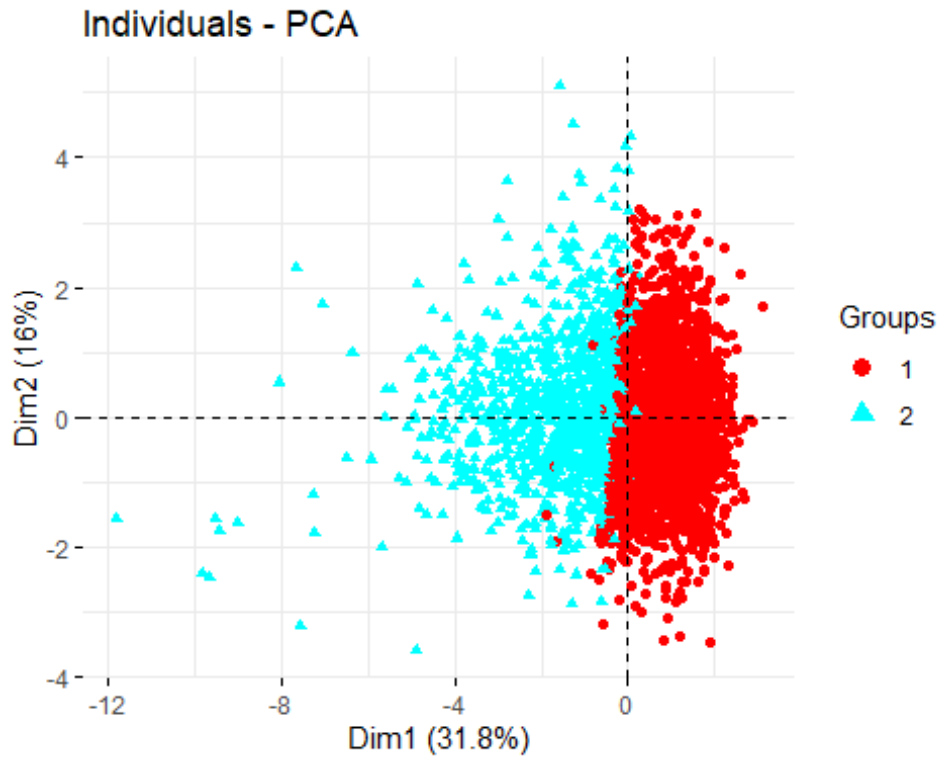


Average silhouette width : 0.23

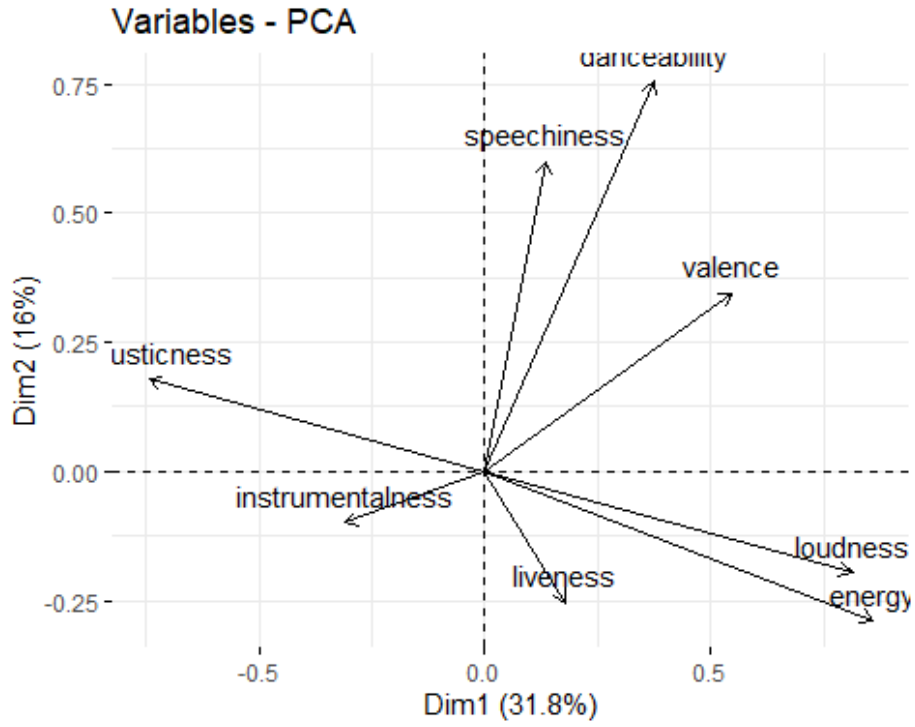
```
misclust = factor(cluster1$cluster)
miPCA = PCA(ncdata2, scale.unit = FALSE, graph = FALSE)
fviz_eig(miPCA)
```



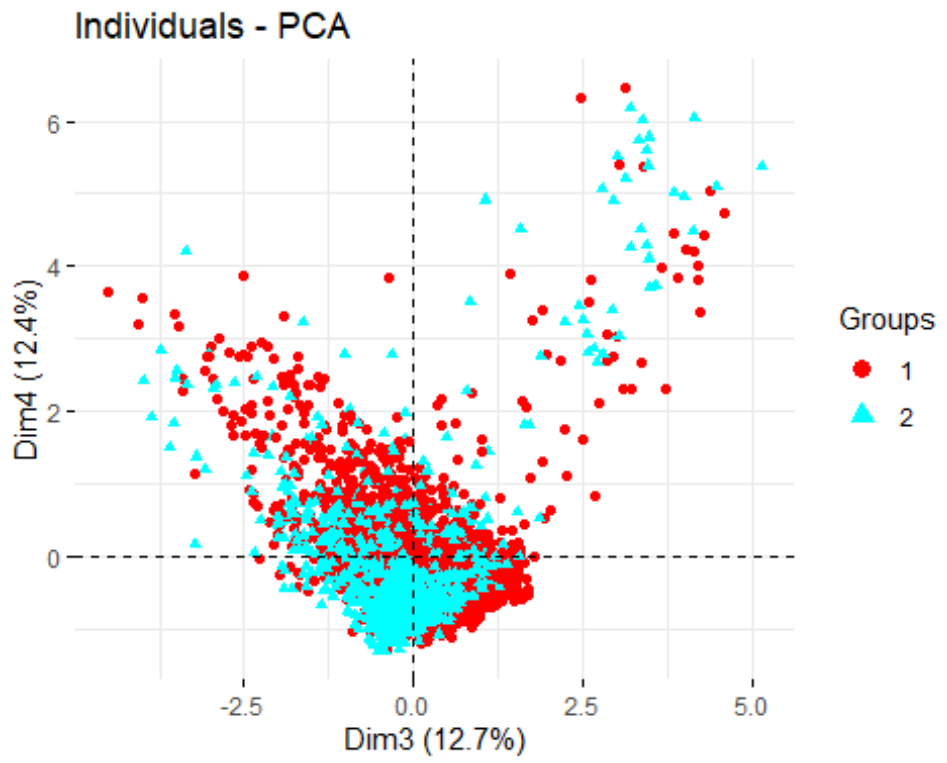
```
fviz_pca_ind(miPCA, geom = "point", habillage = misclust, addEllipses =
FALSE,
             palette = rainbow(2))
```



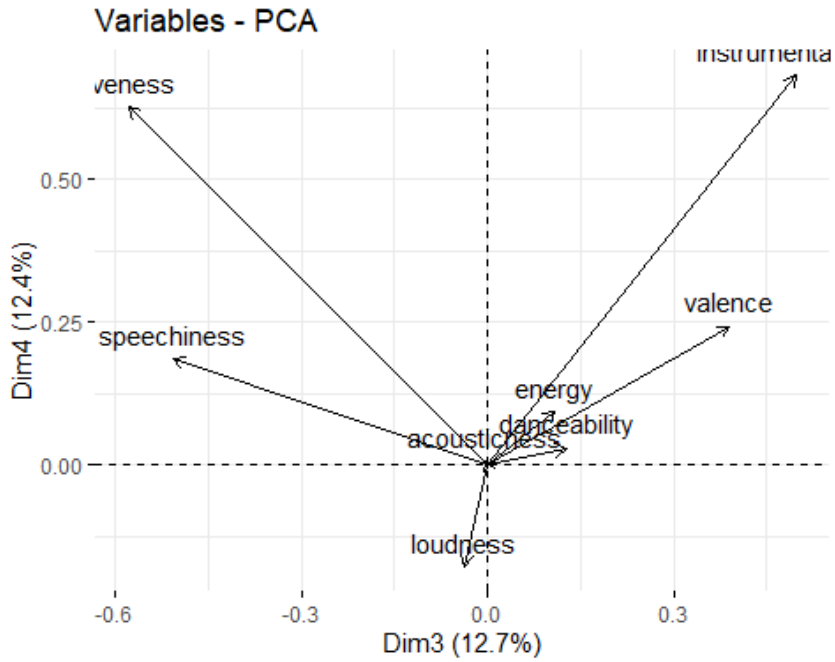
```
fviz_pca_var(miPCA)
```



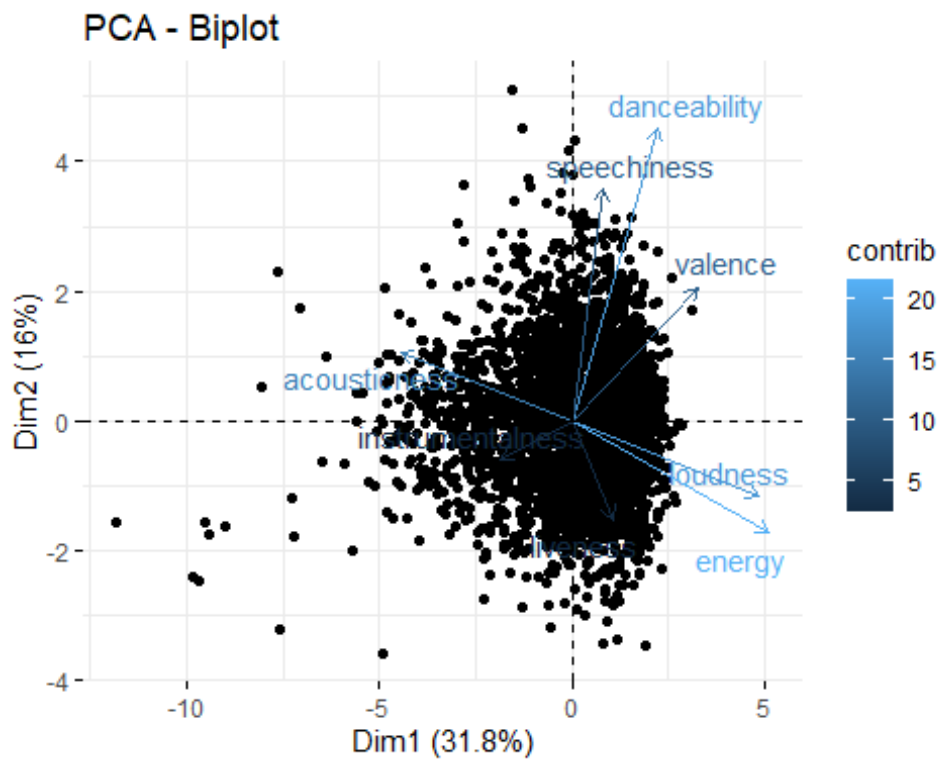

```
fviz_pca_ind(miPCA, geom = "point", habillage = misclust, addEllipses =
FALSE, axes = 3:4,
             palette = rainbow(2))
```




```
fviz_pca_var(miPCA, axes = 3:4)
```



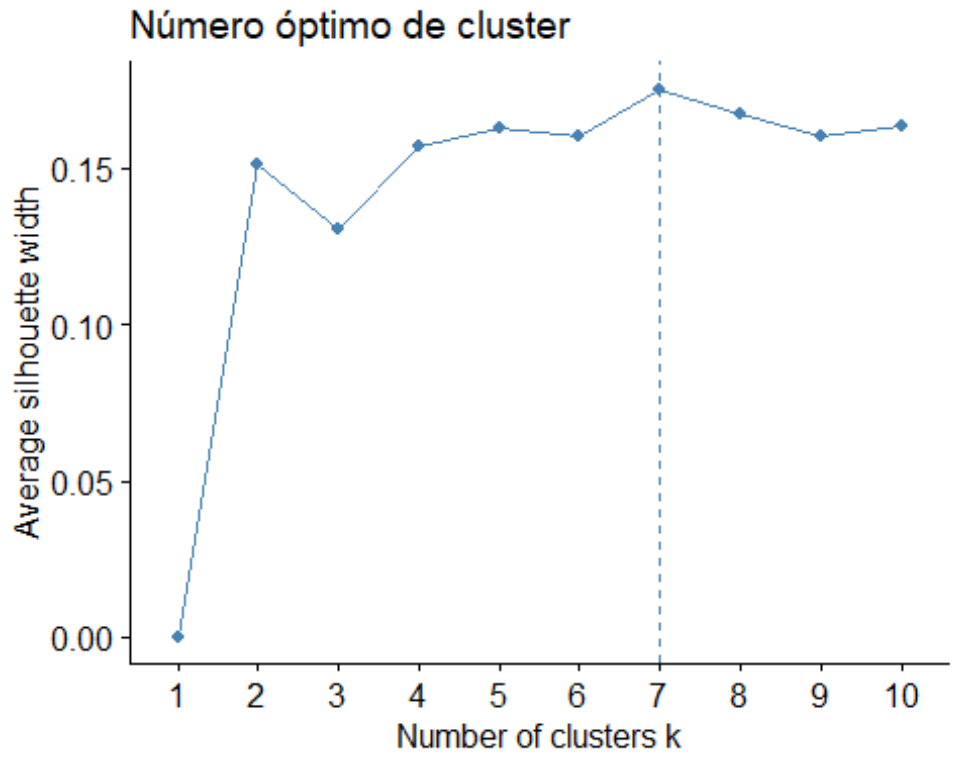

```
fviz_pca_biplot(miPCA, axes = c(1,2), col.var = "contrib", label = "var",
                repel = TRUE)
```




```
ncdata_aux3 = ncdata_aux[which(cluster1$clustering == 1),]
ncdata3 = ncdata2[which(cluster1$clustering == 1),]
View(ncdata3)
ncdata3 = scale(ncdata3, center = TRUE, scale = TRUE)
```

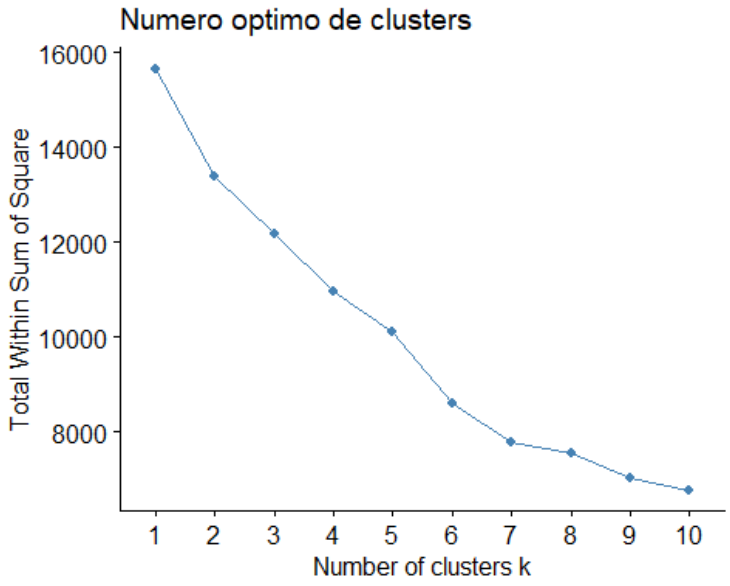
Vamos a realizar otra vez los gráficos para comprobar el nº óptimo de clusters y decidiremos.

```
fviz_nbclust(x=ncdata3, FUNcluster = kmeans, method = 'silhouette', k.max =
10, verbose = FALSE) +
  labs(title = 'Número óptimo de cluster')
```




```
fviz_nbclust(x = ncdata3, FUNcluster = kmeans, method = "wss",
             k.max = 10, verbose = FALSE) +
  labs(title = "Numero optimo de clusters")
```

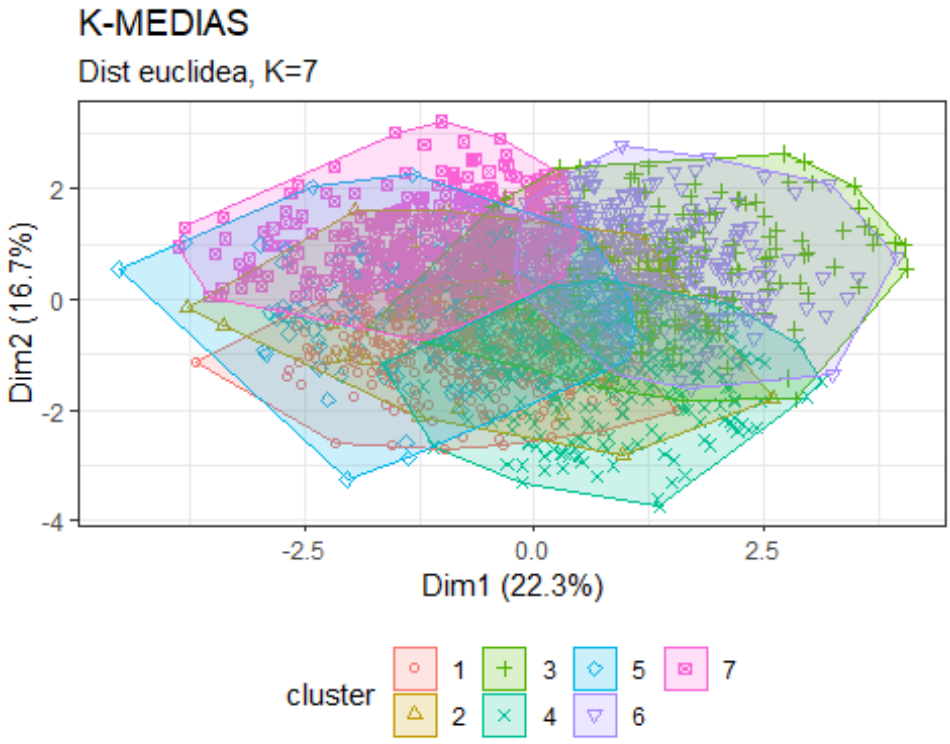
```
## Warning: did not converge in 10 iterations
```



A diferencia del anterior estudio, en este caso 7 clusters es la mejor decisión porque tiene un coeficiente de Silhouette muy alto a diferencia de los demás y su Suma de Cuadrados es aceptable.

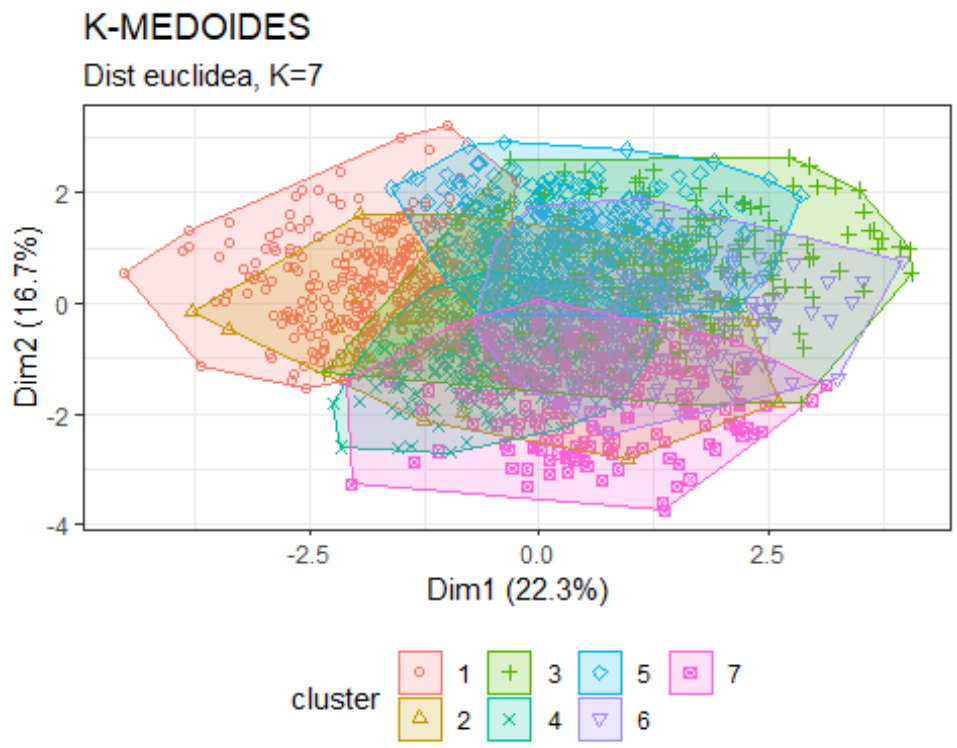
Esta vez vamos a hacer uso tanto de PAM como de kmeans para observar diferencias:

```
cluster_data3 <- kmeans(ncdata3, centers = 7, nstart = 20)
table(cluster_data3$clustering)

## < table of extent 0 >

fviz_cluster(object = list(data=ncdata3, cluster=cluster_data3$cluster),
stand = FALSE,
             ellipse.type = "convex", geom = "point", show.clust.cent =
FALSE,
             labelsize = 8, axes(7:8))  +
  labs(title = "K-MEDIAS",
       subtitle = "Dist euclidea, K=7") +
  theme_bw() +
  theme(legend.position = "bottom")
```




```
clusterpam_data3 <- pam(ncdata3, k = 7)
```

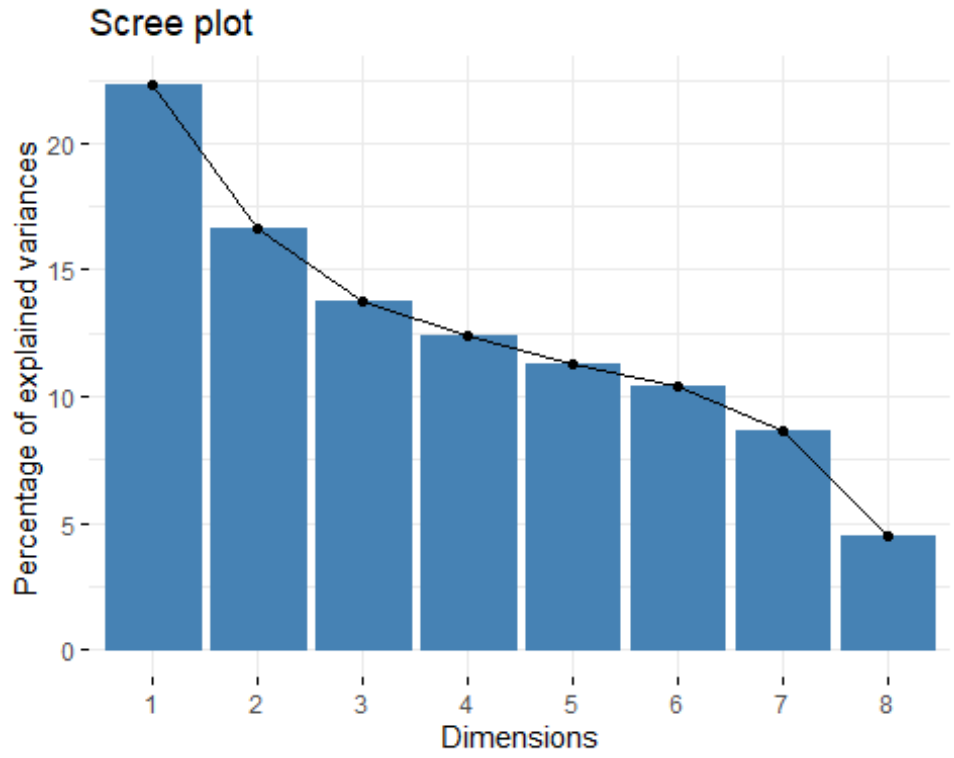
```
fviz_cluster(object = list(data=ncdata3,
cluster=clusterpam_data3$clustering), stand = FALSE,
             ellipse.type = "convex", geom = "point", show.clust.cent =
FALSE,
             labelsize = 8,axes(1:2))  +
  labs(title = "K-MEDOIDES",
       subtitle = "Dist euclidea, K=7") +
  theme_bw() +
  theme(legend.position = "bottom")
```



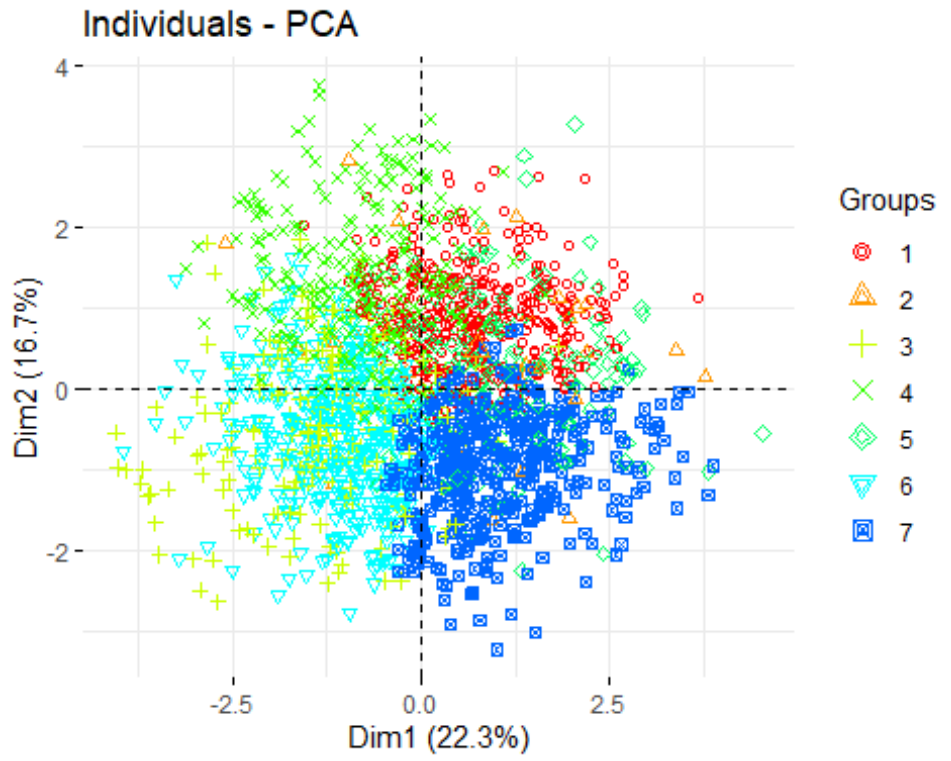

```
cluster_data3
```

```
## K-means clustering with 7 clusters of sizes 449, 35, 228, 217, 181, 430,
417
##
## Cluster means:
##   acousticness danceability     energy instrumentalness   liveness
loudness
## 1  -0.37871210   0.07038226  0.8020375       -0.1253214 -0.2945399
0.6557696
## 2  -0.06217078  -0.15308404  0.5149622        6.8213228 -0.1478089
-0.5203324
## 3  -0.10590762   0.59270112 -0.5050279       -0.1631016 -0.2096925
-0.3161127
## 4   2.11701044   0.31928099 -0.1697148       -0.1400876 -0.2084059
```

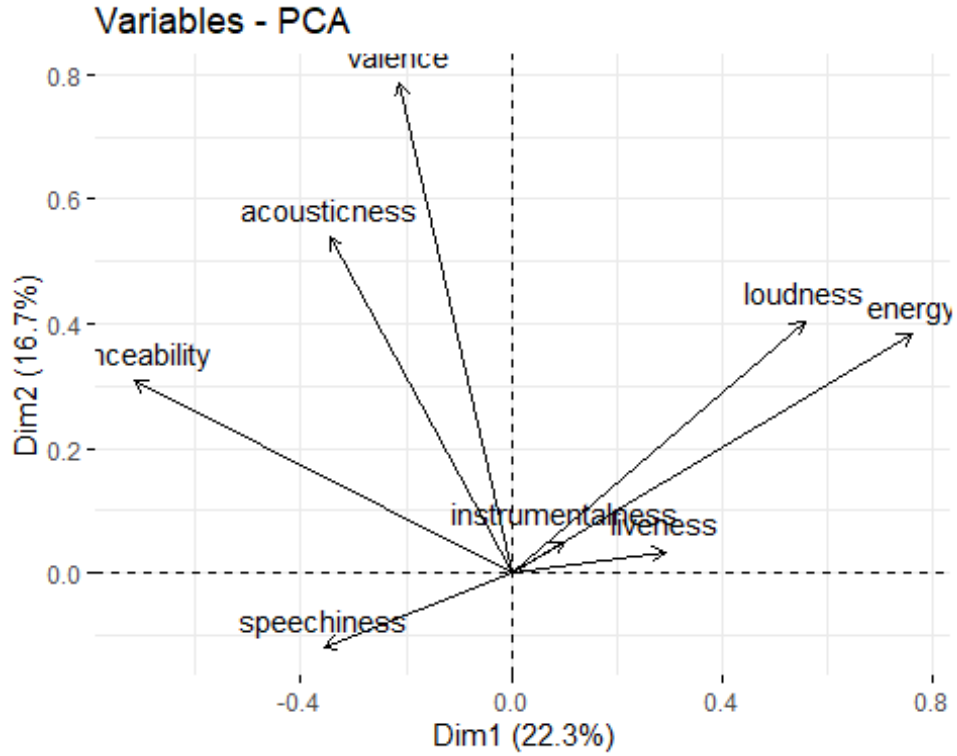
```
0.1519955
## 5  -0.20355619  -0.24619411  0.4013967       -0.1023553  2.2965104
0.2154879
## 6  -0.15737221   0.53596570 -0.8966930       -0.1268538 -0.2767653
-0.8434703
## 7  -0.38012686  -0.99896316  0.2080604       -0.1002815 -0.1587616
0.2075555
##   speechiness     valence
## 1  -0.3225566  0.72366206
## 2  -0.5165601  0.31588689
## 3   2.2110932 -0.06559031
## 4  -0.2366226  0.63571919
## 5  -0.1016328  0.09292016
## 6  -0.2669277 -0.08346549
## 7  -0.3757798 -1.05492847
miPCA2 = PCA(ncdata3, scale.unit = FALSE, graph = FALSE)
misclust2 = factor(cluster_data3$cluster)
fviz_eig(miPCA2)
```



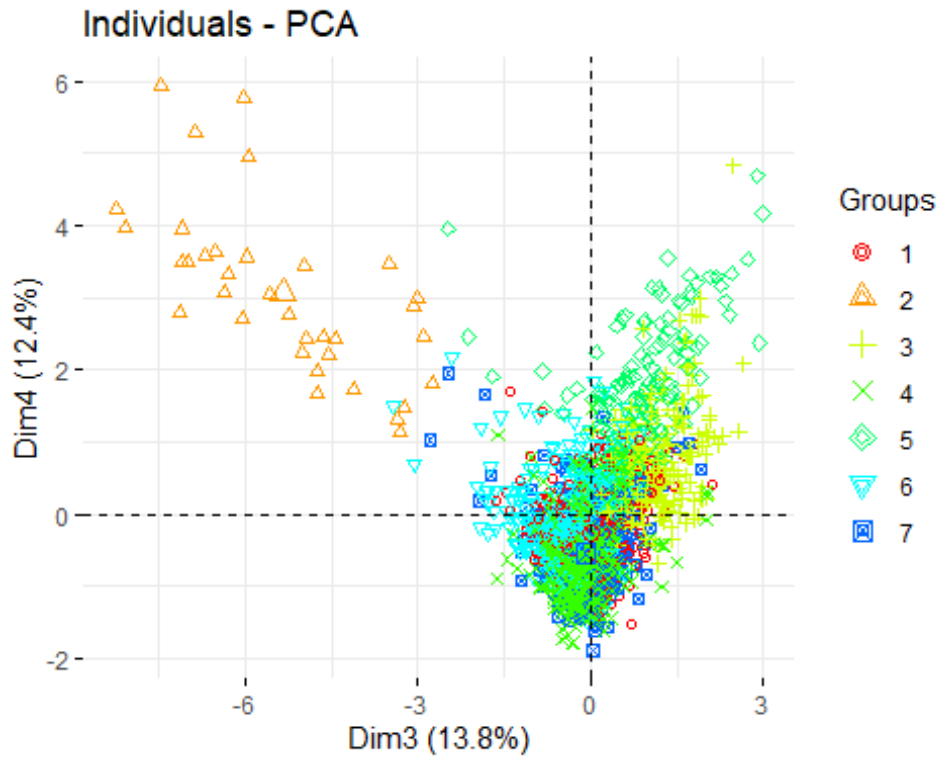

```
fviz_pca_ind(miPCA2, geom = "point", habillage = misclust2, addEllipses =
FALSE,
             palette = rainbow(10))
```



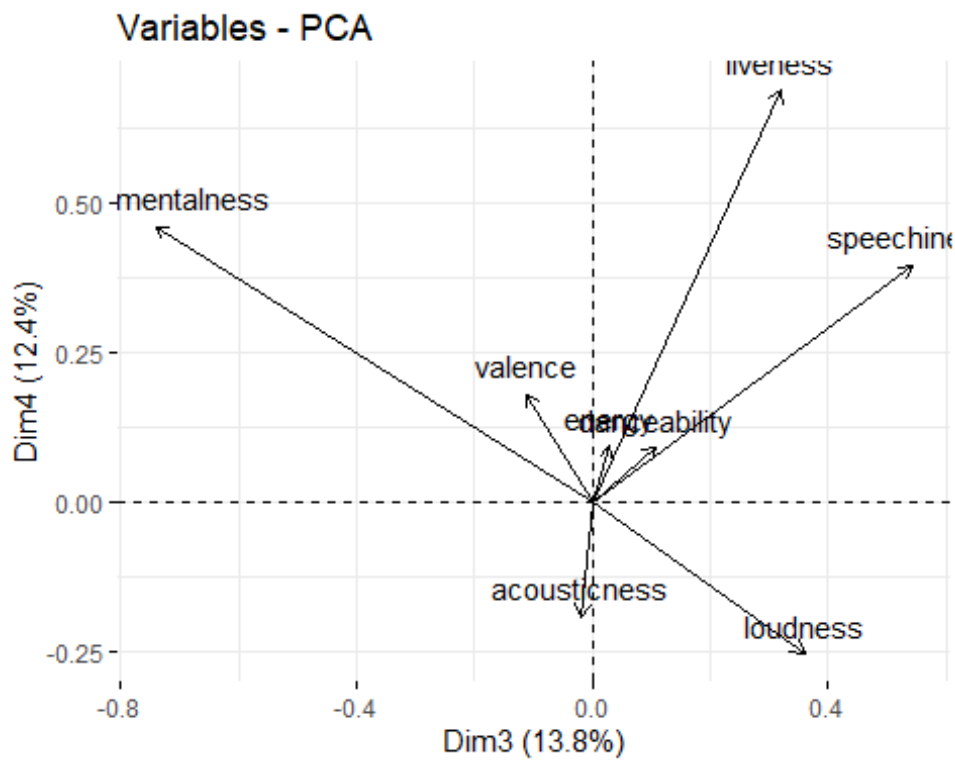

```
fviz_pca_var(miPCA2)
```



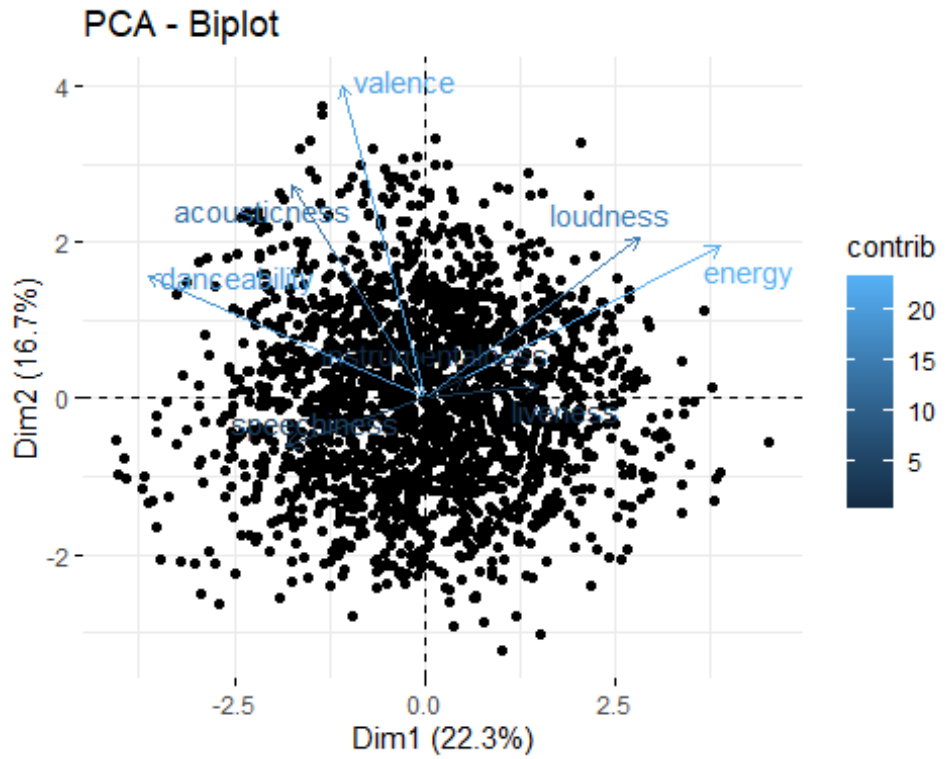
```
fviz_pca_ind(miPCA2, geom = "point", habillage = misclust2, addEllipses =
FALSE, axes = 3:4,
             palette = rainbow(10))
```




```
fviz_pca_var(miPCA2, axes = 3:4)
```



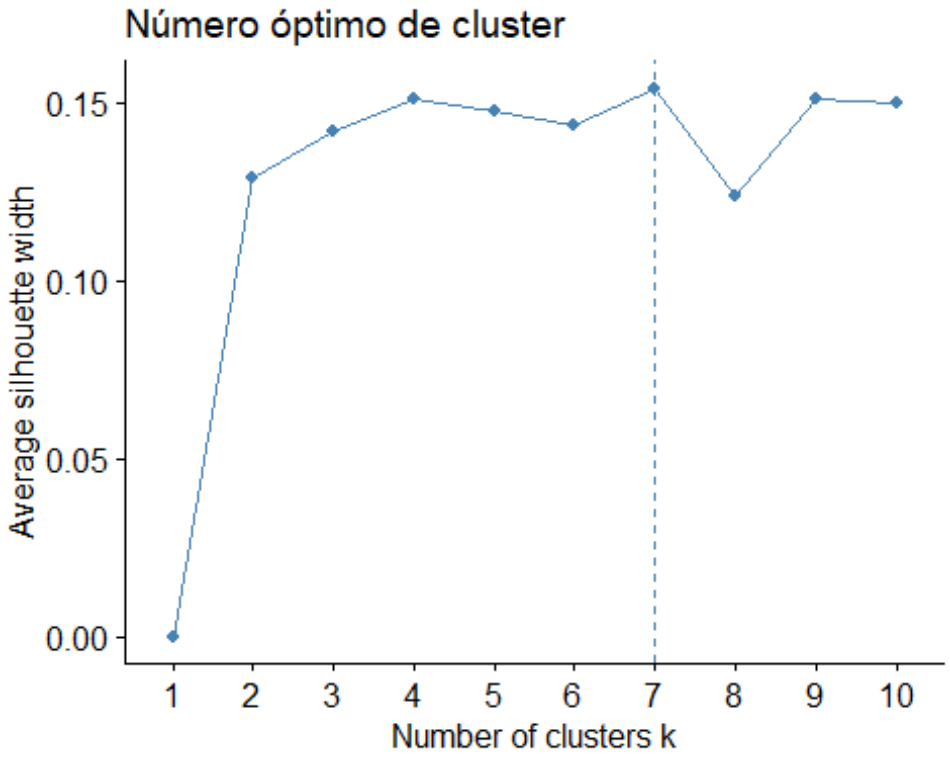

```
fviz_pca_biplot(miPCA2, axes = c(1,2), col.var = "contrib", label = "var",
                repel = TRUE)
```



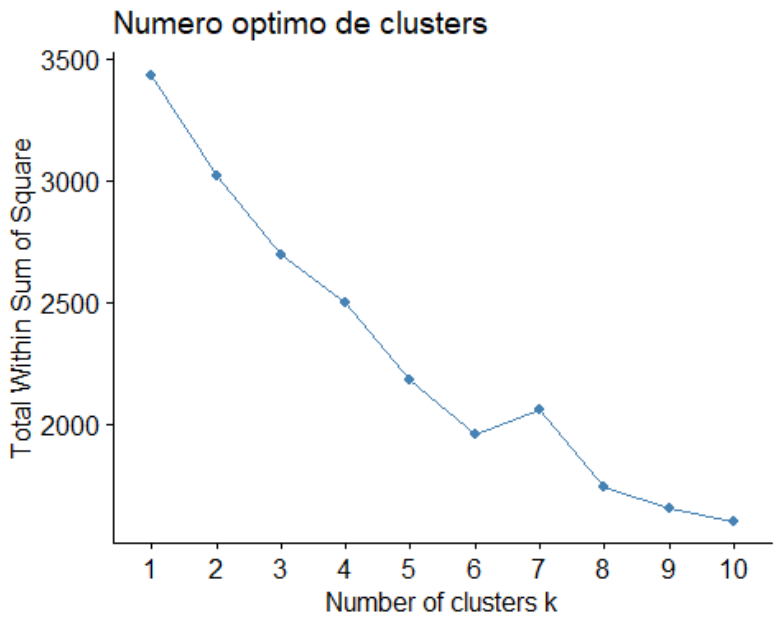
```
ncdata_aux4 = ncdata_aux3[which(cluster_data3$cluster == 6),]
ncdata4 = ncdata3[which(cluster_data3$cluster == 6),]
View(ncdata4)
ncdata4 = scale(ncdata4, center = TRUE, scale = TRUE)
```

Vamos a realizar otra vez los gráficos para comprobar el nº óptimo de clusters y decidiremos.

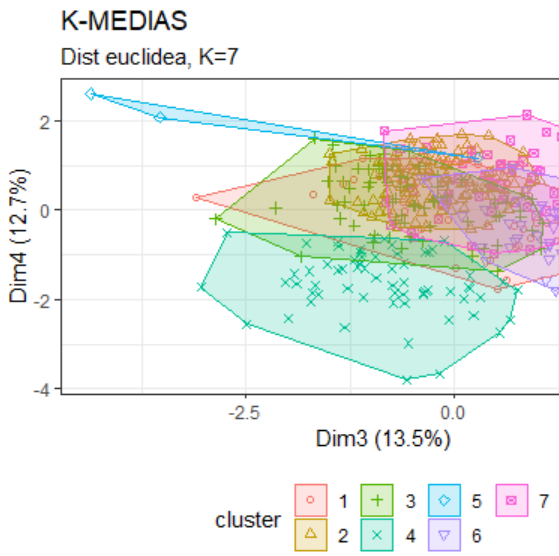
```
fviz_nbclust(x=ncdata4, FUNcluster = kmeans, method = 'silhouette', k.max =
10, verbose = FALSE) +
  labs(title = 'Número óptimo de cluster')
```




```
fviz_nbclust(x = ncdata4, FUNcluster = kmeans, method = "wss",
             k.max = 10, verbose = FALSE) +
  labs(title = "Numero optimo de clusters")
```



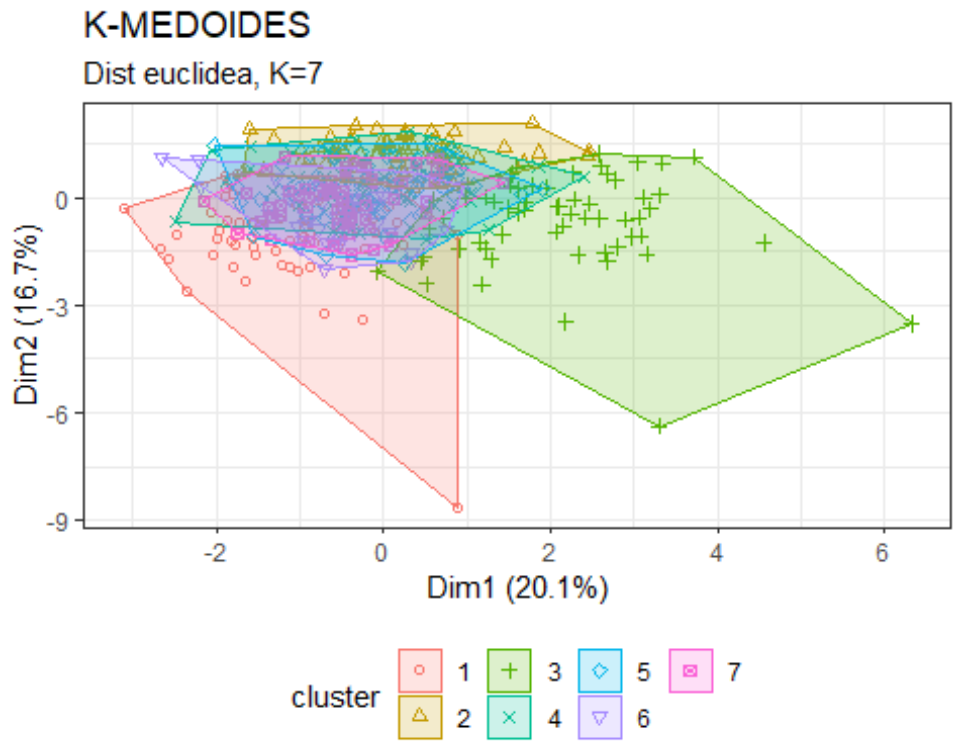

```
cluster_data4 <- kmeans(ncdata4, centers = 7, nstart = 20)


fviz_cluster(object = list(data=ncdata4, cluster=cluster_data4$cluster),
stand = FALSE,
             ellipse.type = "convex", geom = "point", show.clust.cent =
FALSE,
             labelsize = 8,axes= (3:4))  +
  labs(title = "K-MEDIAS",
       subtitle = "Dist euclidea, K=7") +
  theme_bw() +
  theme(legend.position = "bottom")
```



```
clusterpam_data4 <- pam(ncdata4, k = 7)
table(clusterpam_data4$clustering)
```

```
##
##  1  2  3  4  5  6  7
## 63 65 61 54 81 46 60
```

```
fviz_cluster(object = list(data=ncdata4,
cluster=clusterpam_data4$clustering), stand = FALSE,
             ellipse.type = "convex", geom = "point", show.clust.cent =
FALSE,
             labelsize = 8)  +
  labs(title = "K-MEDOIDES",
       subtitle = "Dist euclidea, K=7") +
  theme_bw() +
  theme(legend.position = "bottom")
```




```
cluster_data4
```

```
## K-means clustering with 7 clusters of sizes 50, 117, 72, 67, 3, 57, 64
##
## Cluster means:
##   acousticness danceability     energy instrumentalness   liveness
loudness
## 1   0.46524192   -0.7786970  1.0024387       0.02018484 -0.4851994
-1.46157223
```

```
## 2  -0.62083172   -0.5088678  0.1810722      -0.08966402 -0.2991351
0.28780397
## 3  -0.16839177    0.4900375 -0.3236632      -0.06906328 -0.3531460
-0.21172236
## 4  -0.08554912    0.1114310  0.3106113      -0.14522548  1.9258363
0.06527926
## 5   0.61635938    0.7670115  0.6722573      10.14208369 -0.6894351
-1.72856280
## 6   1.43550898    0.2006360 -0.3718753      -0.06157458 -0.1520624
0.43294279
## 7  -0.25690432    0.6560394 -0.7755385      -0.04269355 -0.5251543
0.48099683
##    speechiness     valence
## 1 -0.652298449  1.09360977
## 2 -0.510295144 -0.58115563
## 3  1.642540234 -0.33928742
## 4  0.090790076  0.07221301
## 5 -0.004505373  1.66326829
## 6 -0.337726538 -0.53085129
## 7 -0.199413263  0.90896659
##
##
## Within cluster sum of squares by cluster:
## [1] 284.90232 361.34154 345.96624 324.78667  71.59524 180.41820 248.56114
##  (between_SS / total_SS =  47.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

En conclusión, se puede comprobar que el algoritmo de k-medoides no proporciona resultados mucho mejores que el de k-medias (Anexo VIII), existiendo mucho solapamiento cuando el nº de clusters es mayor que 2, y por lo tanto creando un agrupamiento muy pobre (solo dos grupos).

La otra alternativa explorada ha sido volver realizar clustering dentro de un cluster muy grande (proporcionado por PAM), con el fin de aumentar la granularidad de nuestro agrupamiento: sin embargo, hemos comprobado que dentro de este grupo, el solapamiento es terrible.

La conclusión, después de aplicar varios métodos de clustering diferentes (Anexos VIII, IX y este informe), parece que contamos con demasiadas variables correlacionadas entre sí, pudiendo afectar dramáticamente a la calidad del agrupamiento. Por lo tanto, una alternativa a explorar es realizar esta técnica pero únicamente con aquellas variables que estén muy poco correlacionadas. Esta opción es explorada en el Anexo X.