

Actividad 1.10 Conglomerados no jerárquicos

Franco Mendoza Muraira A01383399

2023-11-22

1. Lea los datos con `read.csv` y cree una nueva matriz sin las variables fecha y hora, finalmente, cree una nueva matriz estandarizada.

```
library(factoextra)

## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

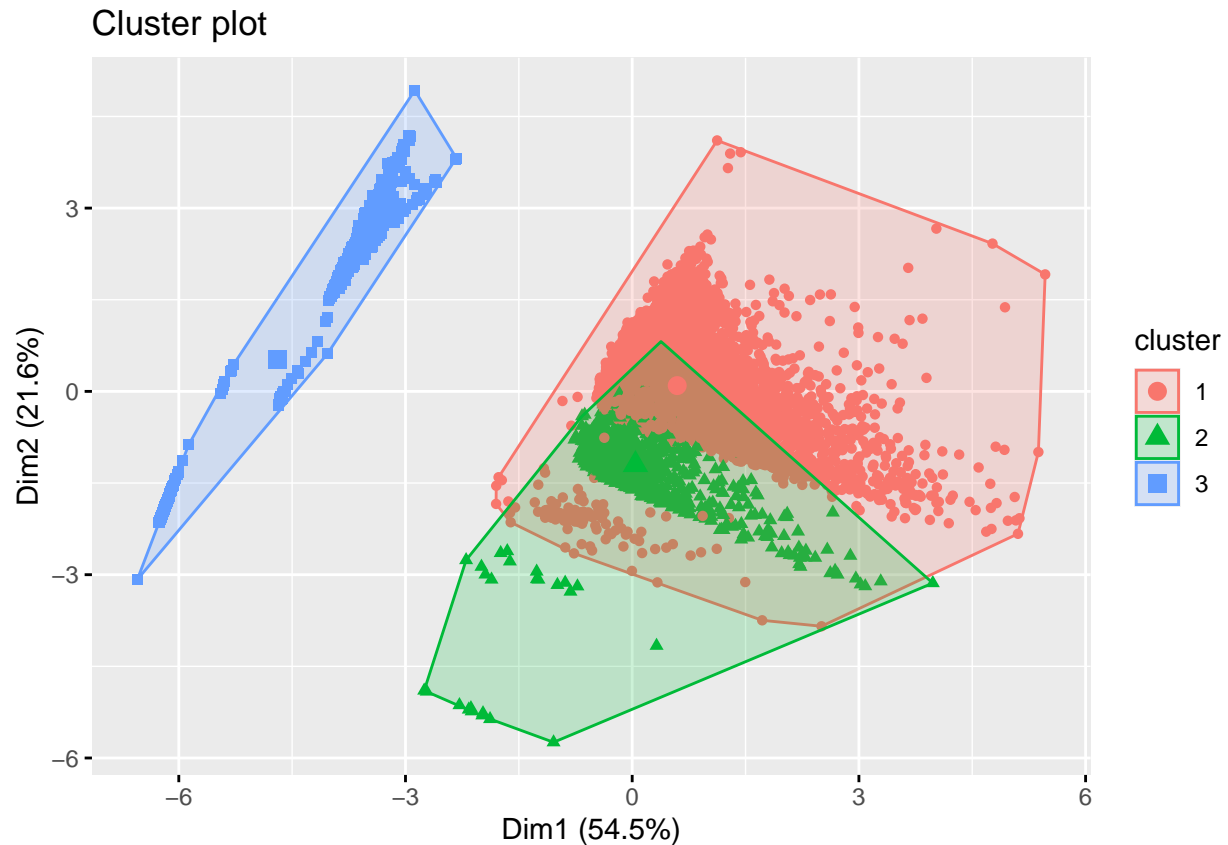
M = read.csv("TLA2021.csv")
names(M)

## [1] "FECHA" "CO"      "NO"      "NO2"     "NOX"     "O3"      "PM10"   "SO2"

M1 = M[, -1:-2] # quitando las columnas de fecha y hora
Mstand = scale(x = M1, center = TRUE, scale = TRUE)
```

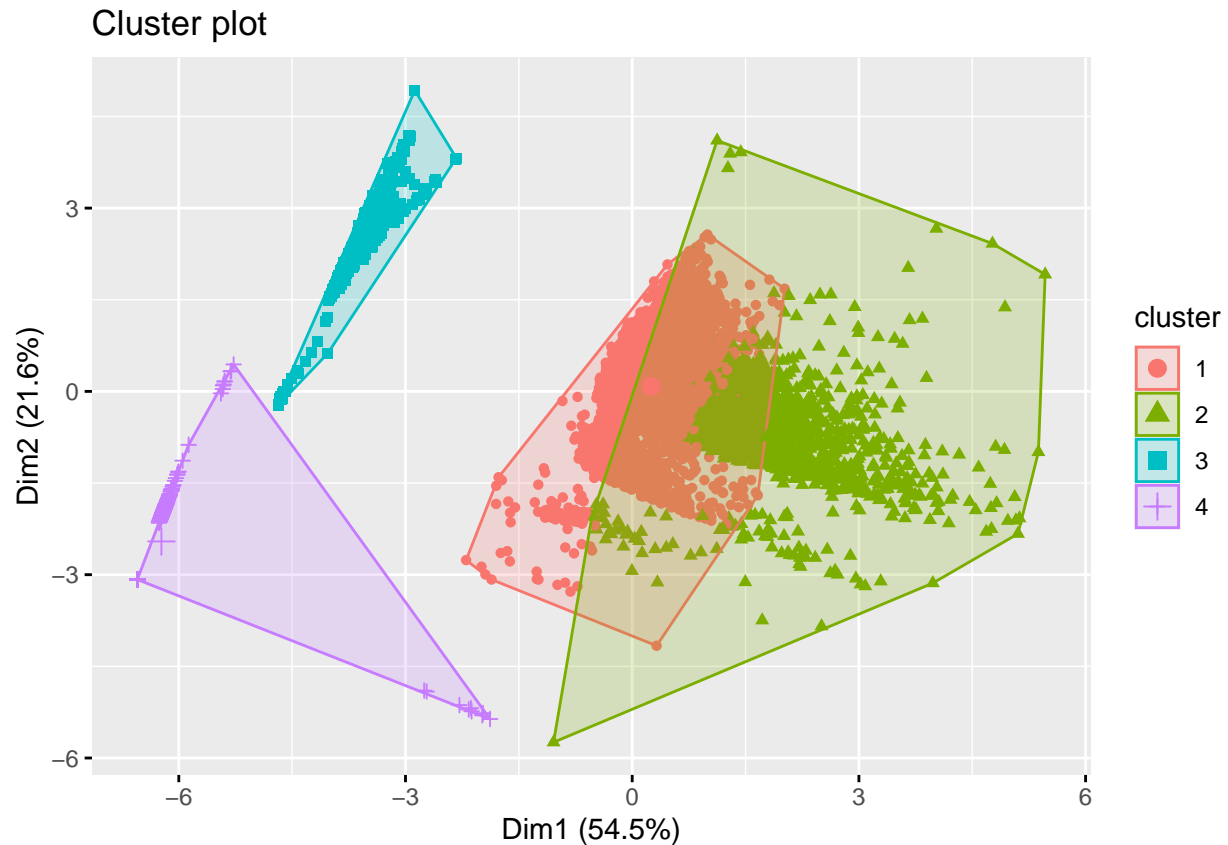
2. Aplique el algoritmo de k-means y su gráfico con las funciones `kmeans` y `fviz_cluster`. Podría serles útil:

```
M1f = data.frame(Mstand) # Se convierte la matriz Mstand a un data.frame porque así lo requiere la fun
km_clusters = kmeans(M1f, centers = 3)
fviz_cluster(object = km_clusters, data = M1f, show.clust.cent = TRUE, ellipse.type = "convex", star.pl
```



En nuestro primer grafico intentamos hacer 3 clusters con los datos proporcionados, con 2 dimensiones que explican 76% de la varianza total de los datos. Podemos observar que los clusters 1 y 2 se cruzan muchos datos entre sí, sin excluirse unos de los otros. Esto nos muestra que la decisión de usar 3 clusters es una mala sin buena definición entre ellos, ya que causa confusión en el gráfico. El cluster 2 se puede ver que no es malo, ya que se separa bien de los demás datos, aunque toma una forma curiosa debido a algunos datos más alejados del centroide.

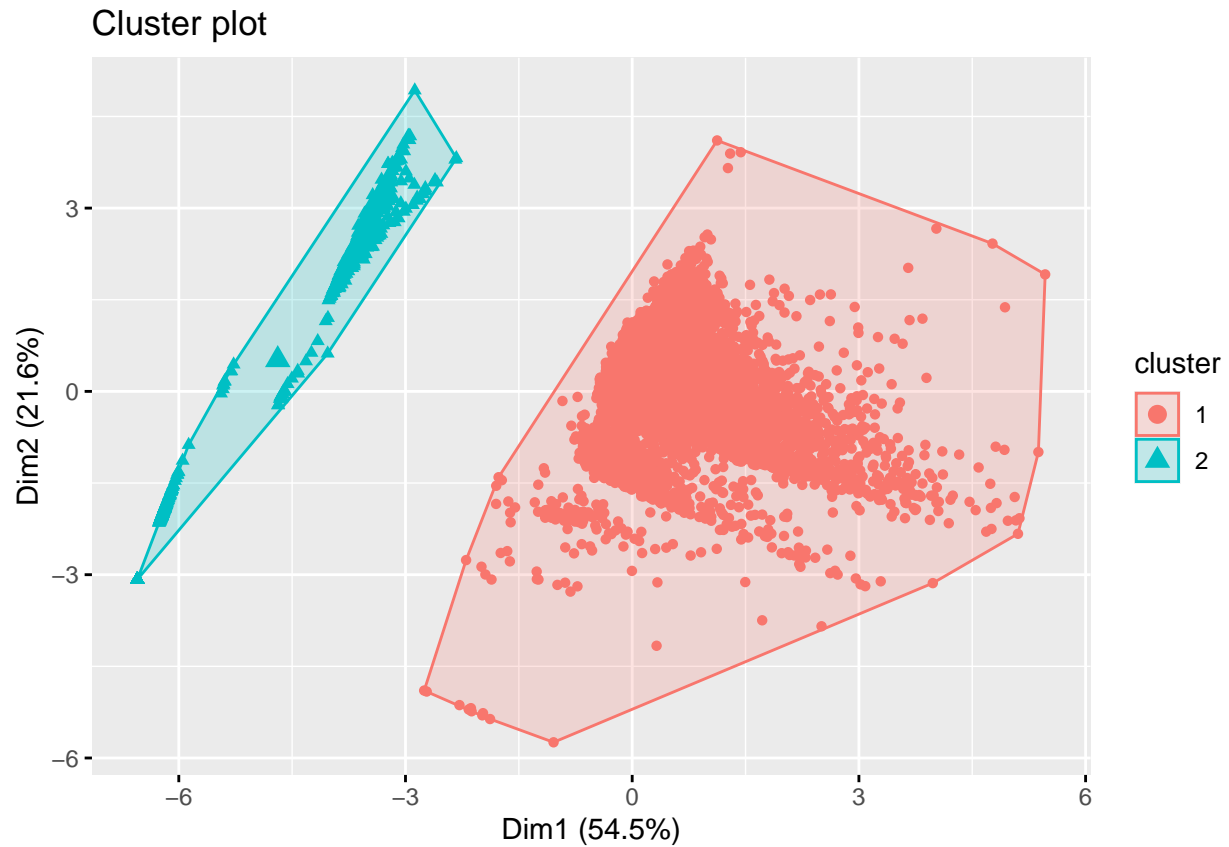
```
M1f = data.frame(Mstand) # Se convierte la matriz Mstand a un data.frame porque así lo requiere la fun
km_clusters = kmeans(M1f, centers = 4)
fviz_cluster(object = km_clusters, data = M1f, show.clust.cent = TRUE, ellipse.type = "convex", star.pl
```



Ahora, podemos ver un intento con 4 clusters, podemos ver que ahora el cluster 2 agarra una forma con los datos más centrados, y el nuevo cluster más que nada agarra los datos que eran atípicos del cluster 1 y 2 de la gráfica pasada.

Aún así, en esta gráfica aún podemos observar que el número de clusters no es el óptimo debido a que los clusters 1 y 2 se siguen cruzando mucho, con datos confusos en los diferentes clusters.

```
M1f = data.frame(Mstand) # Se convierte la matriz Mstand a un data.frame porque así lo requiere la fun
km_clusters = kmeans(M1f, centers = 2)
fviz_cluster(object = km_clusters, data = M1f, show.clust.cent = TRUE, ellipse.type = "convex", star.pl
```



Ahora con 2 clusters, podemos ver que los datos se separan de buena forma, y se encontró la cantidad de clusters óptima para el problema, se puede ver que en los 2 clusters se tienen dato atípicos, pero los grupos de datos están bien definidos y diferenciados de los demás.

Conclusión

Pudimos observar en este análisis la evaluación de datos usando clustering, y pudimos evaluar que la cantidad de clusters era de 2 con solo observar las diferentes gráficas, y usando diferentes valores para poder observar las diferencias.