Practica 6

Barron Orozco Gabriela del Carmen

13/4/2020

Preparacion de los datos

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 3.6.3
```

```
Abandono <- read.csv("~/M@C/Semestre 8/Mineria de
Datos/Datos_de_Seguridad_Alimentaria_S_Todos_los_Datos/Tasa de abandono
escolar 2018-19.csv", row.names=1)
str(Abandono)
```

```
## 'data.frame':    32 obs. of  4 variables:
##  $ Primaria      : num  0.1 0.4 -0.3 1.2 0.2 1.3 1.1 0.2 1.4 0.6 ...
##  $ Secundaria    : num  5.8 4.4 2.6 5 3.5 6.7 5.6 4.9 3.7 6.7 ...
##  $ Media.superior: num  12.4 13 11 11.9 13.6 13.7 11.1 14.7 16 15.1
...
##  $ Superior      : num  7.5 6.4 12.7 10.3 7.4 10.5 8 6.8 10.3 7.1 ...
```

Vamos a preparas nuestros datos, para poder realizar el cluster y calcular nuestra matriz de distancia

```
abandono = na.omit(Abandono)
abandono= scale(abandono)
head(abandono, n=3)
```

```
##                      Primaria Secundaria Media.superior   Superior
## Aguascalientes      -0.6479375  0.9100554    -0.08135568 -0.2511215
## Baja California     -0.1503215 -0.1685288     0.15531540 -0.5773014
## Baja California Sur -1.3114255 -1.5552799    -0.63358820  1.2908199
```

Vamos a cargar nuestras librerias

```
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 3.6.3
```

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 3.6.3
```

```
## Loading required package: ggplot2
```

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

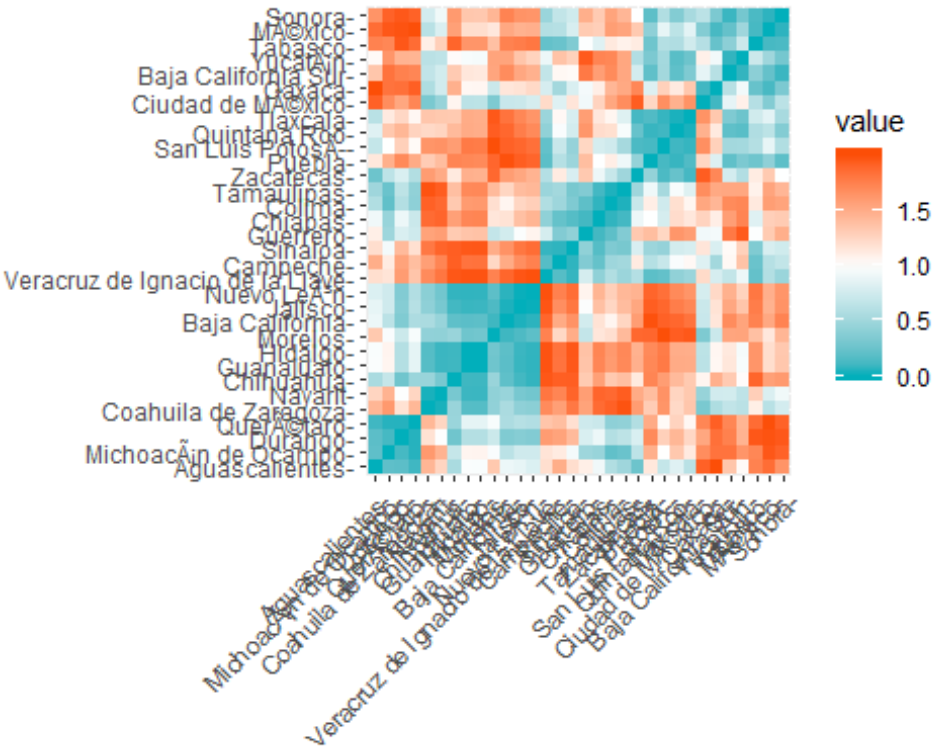
Matriz de distancias

Es sencillo calcular y visualizar la matriz de distancias utilizando las funciones get_dist () y fviz_dist () en el paquete factoextra r:

Get_dist (): para calcular una matriz de distancia entre las filas de una matriz de datos. Comparado con la función dist () estándar, soporta medidas de distancia basadas en la correlación incluyendo los métodos “pearson”, “kendall” y “spearman”.

Fviz_dist (): para visualizar una matriz de distancia

```
res.dist <- get_dist(abandono, stand = TRUE, method = "pearson")
fviz_dist(res.dist, gradient = list(low = "#00AFBB", mid = "white", high
= "#FC4E07"))
```



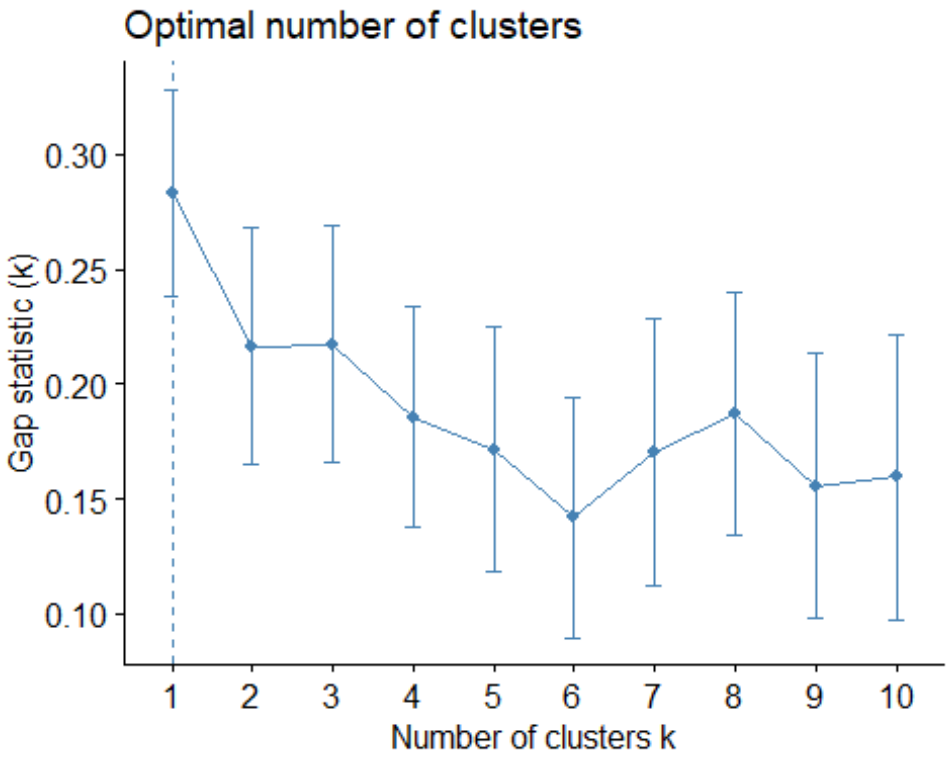
El nivel de color es proporcional al valor de la diferencia entre las observaciones: rojo si d(x,y)=0 y azul si dist(x,y)=1. Los objetos que pertenecen al mismo cluster se visualizan en orden consecutivo.

Particionando el cluster

Los algoritmos de partición son enfoques de agrupamiento que dividen los conjuntos de datos, que contienen n observaciones, en un conjunto de k grupos (es decir, conglomerados). Los algoritmos requieren que el analista especifique el número de clústeres que se generarán.

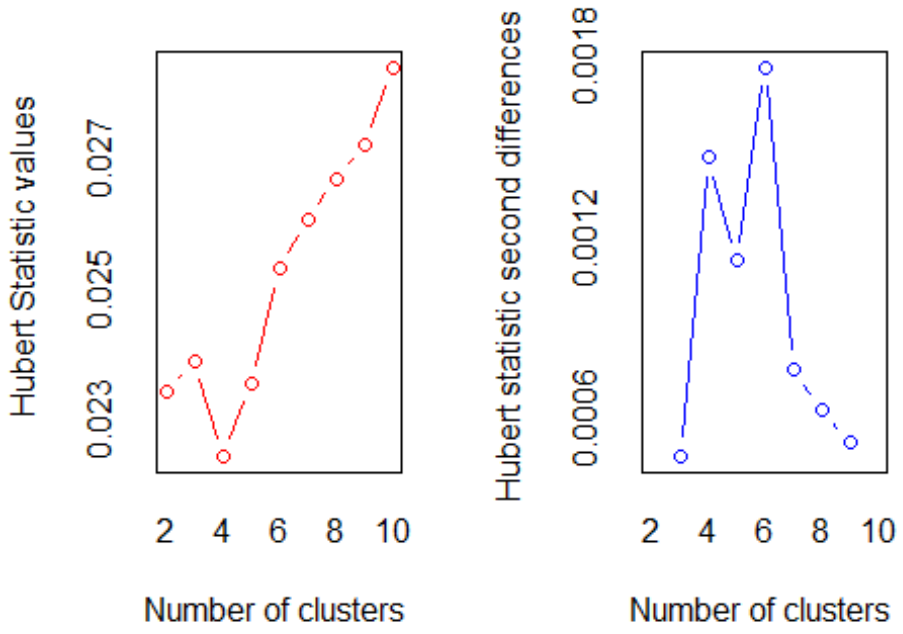
Determinacion optima del numero de cluster

```
fviz_nbclust(abandono, kmeans, method = "gap_stat")
```

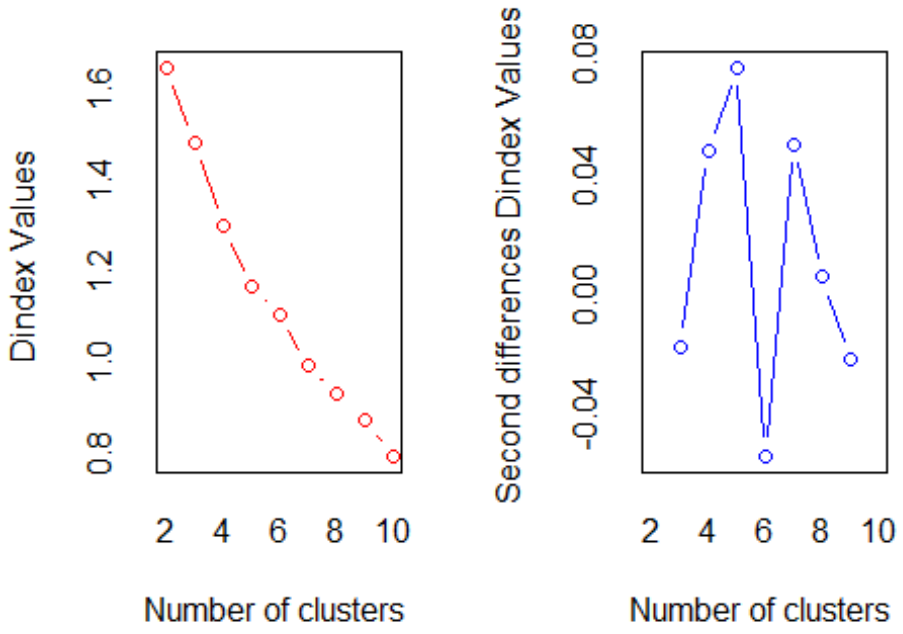


Nbclust: un paquete que proporciona 30 índices para determinar el mejor número de clusters

```
library(NbClust)
set.seed(123)
res.nbclust <- NbClust(abandono, distance = "euclidean",
                  min.nc = 2, max.nc = 10,
                  method = "complete", index ="all")
```




```
## *** : The Hubert index is a graphical method of determining the number
of clusters.
##                 In the plot of Hubert index, we seek a significant
knee that corresponds to a
##                 significant increase of the value of the measure i.e
the significant peak in Hubert
##                 index second differences plot.
##
```

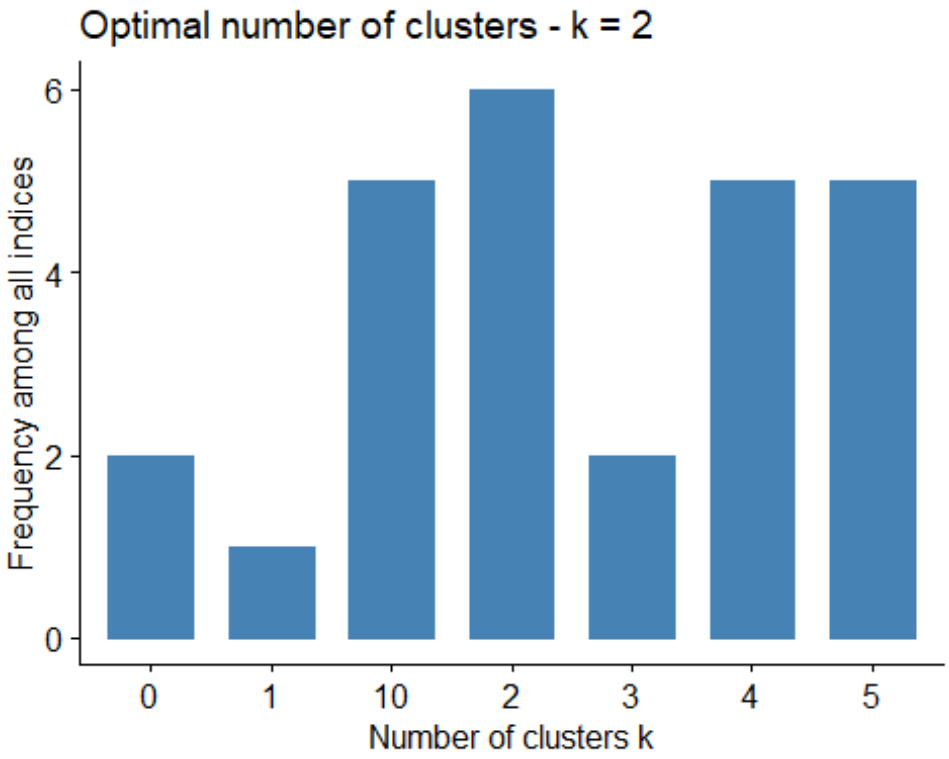



```
## *** : The D index is a graphical method of determining the number of
clusters.
##                 In the plot of D index, we seek a significant knee
(the significant peak in Dindex
##                 second differences plot) that corresponds to a
significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 6 proposed 2 as the best number of clusters
## * 2 proposed 3 as the best number of clusters
## * 5 proposed 4 as the best number of clusters
## * 5 proposed 5 as the best number of clusters
## * 5 proposed 10 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

```
factoextra::fviz_nbclust(res.nbclust)
```

```
## Among all indices:
## ==================
```

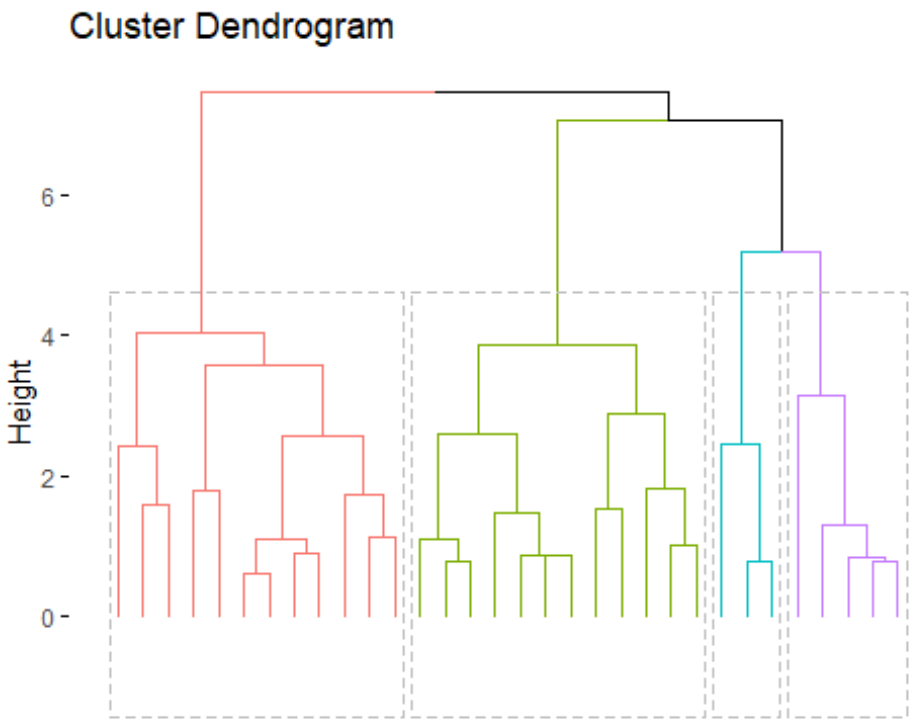
```
## * 2 proposed  0 as the best number of clusters
## * 1 proposed  1 as the best number of clusters
## * 6 proposed  2 as the best number of clusters
## * 2 proposed  3 as the best number of clusters
## * 5 proposed  4 as the best number of clusters
## * 5 proposed  5 as the best number of clusters
## * 5 proposed  10 as the best number of clusters
##
## Conclusion
## =========================
## * According to the majority rule, the best number of clusters is  2 .
```



Estadisticas de validacion de clusters

Se ha propuesto una variedad de medidas en la literatura para evaluar los resultados de la agrupación. El término validación de agrupamiento se utiliza para diseñar el procedimiento de evaluación de los resultados de un algoritmo de agrupación.

```
my_data <- scale(iris[, -5])
res.hc <- eclust(abandono, "hclust", k = 4, graph = FALSE)
# Visualize
fviz_dend(res.hc, rect = TRUE, show_labels = FALSE)
```

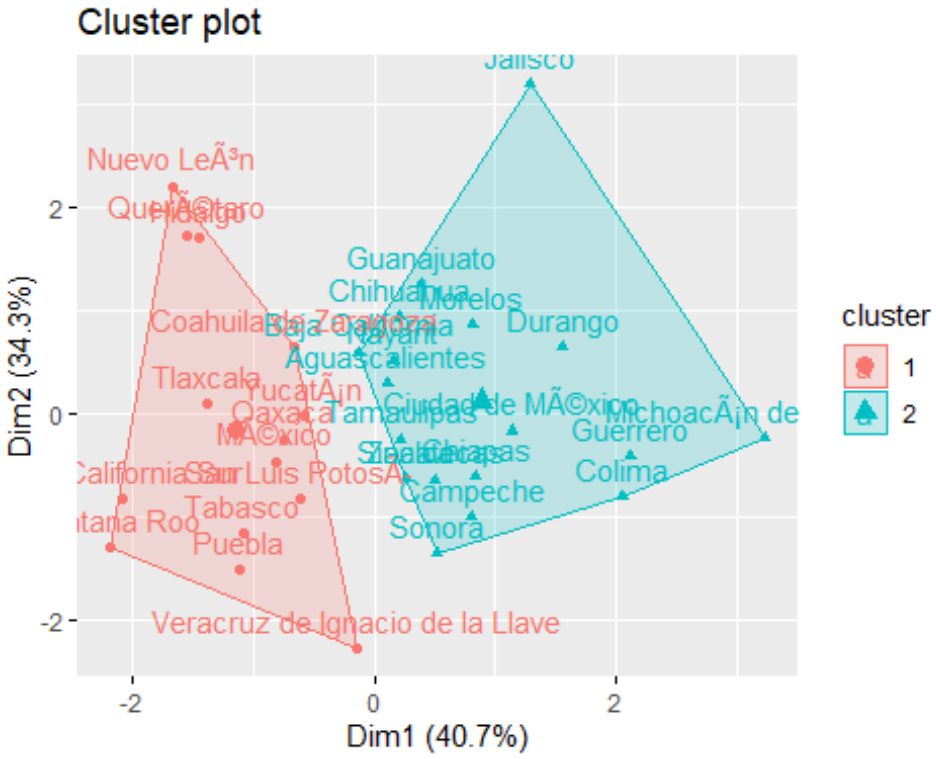


Cluster kmeans

```
km.res <- kmeans(abandono, 2, nstart = 25)
fviz_cluster(km.res, data = abandono, frame.type = "convex")

## Warning: argument frame is deprecated; please use ellipse instead.

## Warning: argument frame.type is deprecated; please use ellipse.type
instead.
```



Cluster PAM

```
pam.res <- pam(abandono, 4)
fviz_cluster(pam.res)
```