Clustering

Rocío Vecino Torres

2/1/2022

Clustering

La agrupación es una técnica para agrupar puntos de datos similares en un grupo y separar las diferentes observaciones en diferentes grupos.

Antes de empezar con el análisis del clustering, cargamos las librerias necesarias y el dataset de observaciones.

Cargamos las librerias necesarias:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cluster)
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(ggthemes)
library(purrr)
library(ggplot2)
```

Cargamos los datos del fichero .csv:

```
#En un primer lugar cargaremos los datos:
students_data <- read.csv("StudentsPerformancePreprocesado.csv", sep = ',', head = TRUE)
students_data$id <- NULL
colnames(students_data)
```

```
## [1] "gender"                      "race.ethnicity"
## [3] "parental.level.of.education" "lunch"
## [5] "test.preparation.course"     "math.score"
## [7] "reading.score"               "writing.score"
```

```
view(students_data)
```

Estudio 1: Clustering Jerárquico.

Los clústers pueden crearse de arriba a abajo o viceversa. Por lo tanto, son dos tipos: Divisivo y Aglomerativo. En este estudio veremos ambos tipos y se hará una comparativa entre los dos.

- El tipo devisivo consiste en: suponer que todas las observaciones pertenecen a un único grupo y luego dividimos el clúster en dos grupos menos similares. Esto se repite recursivamente en cada grupo hasta que haya un grupo para cada observación.
- El tipo aglomerativo consiste en: que cada observación se asigna a su propio clúster. Luego, se calcula la similitud (o distancia) entre cada uno de los clusters y los dos clusters más similares se fusionan en uno. Finalmente, los pasos 2 y 3 se repiten hasta que solo quede un grupo.

Agrupamiento jerárquico aglomerativo:

Hay dos funciones interesantes para el clustering agromerativo: la función hclust(), que es la que hemos usado en clase, y la función agnes(). En este estudio se hara uso de la función agnes().

La función agnes además de que nos permite conocer el coeficiente de aglomeración, mide la cantidad de estructura de agrupamiento encontrada (los valores más cercanos a 1 sugieren una estructura de agrupación fuerte).

En un primer lugar, vamos a probar agnes con un link de tipo complete.

```
cluster_aglomerativo <- agnes(students_data, method = "complete")
#mostramos el coeficiente aglomerativo
cluster_aglomerativo$ac
```

```
## [1] 0.9785117
```

Vemos que el coeficiente aglomerativo para un cluster de tipo complete link es de: 0,978. Muy cercana a 1.

Ahora, vamos a comparar con los distintos tipos vistos en clase y nos quedamos con el que tenga mayor coeficiente aglomerativo. Esos métodos son:

- Average Link
- Single Link
- Complete Link

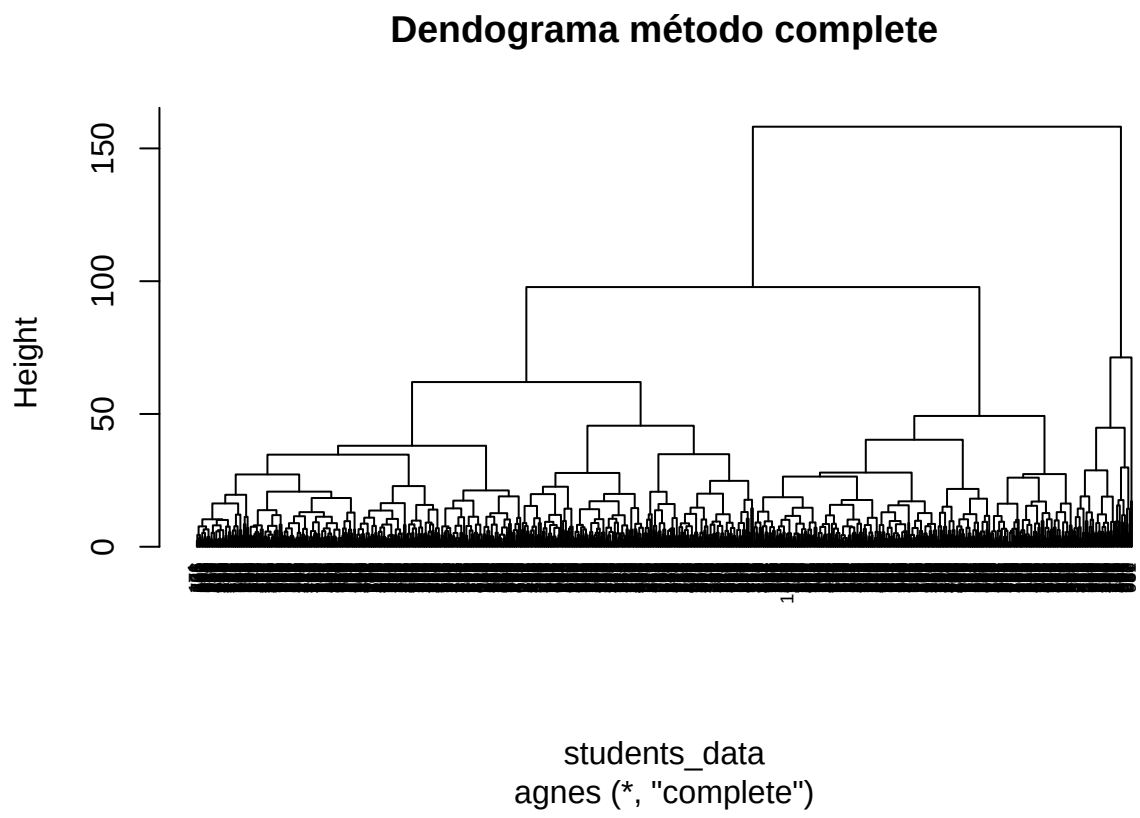
```r
#vector con los metodos a comparar
m <- c("average","single","complete")
names(m) <- c("average","single","complete")

#funcion para calcular el coeficiente por metodo del vector
ac <- function(x){
  agnes(students_data, method = x)$ac
}
#show
map_dbl(m,ac)
```

```
##   average    single  complete
## 0.9531943 0.8204931 0.9785117
```

Viendo los tres coeficientes vemos que efectivamente, el que mejor coeficiente de aglomeración tiene es el cluster usando el método de **Complete.**

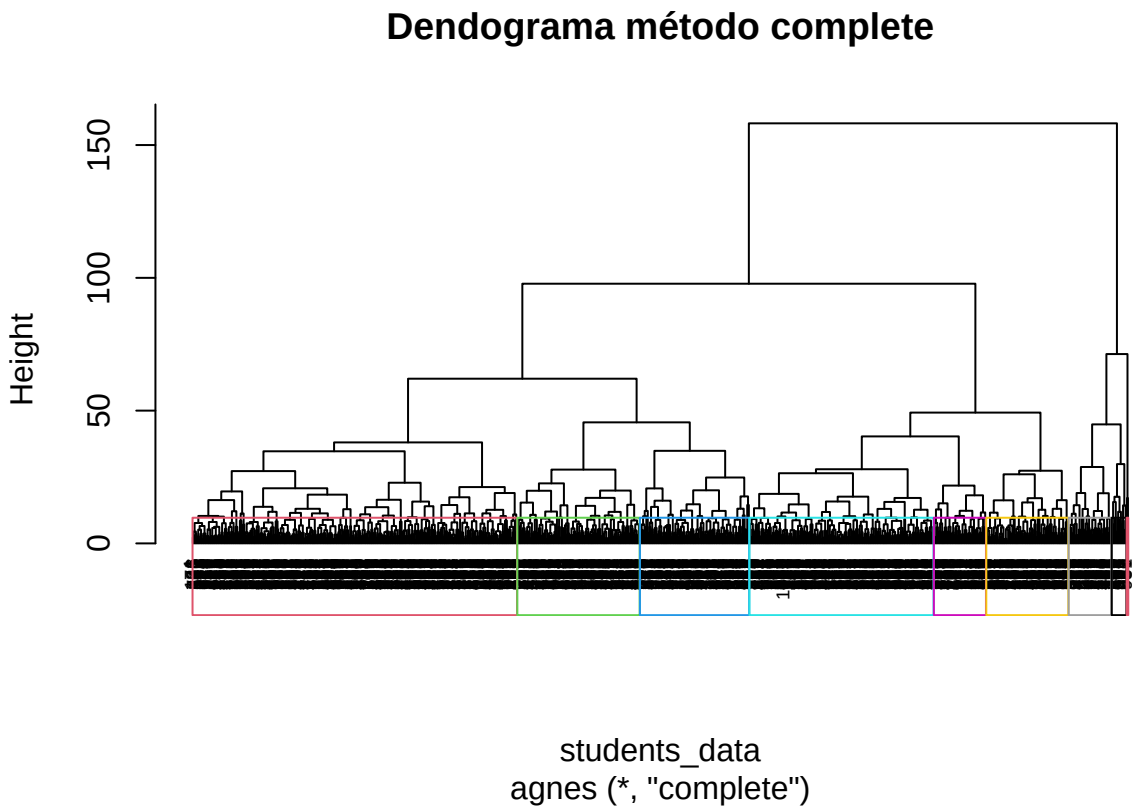
Visualizamos el dendograma:

```r
#dendograma
pltree(cluster_aglomerativo, cex = 0.6, hang = -1, main = "Dendograma método complete")
```



El dendograma podemos diferenciar los cluster mediante un parametro k, colocandole unos bordes para diferenciarlos. Mediante la siguiente manera:

```
#dendograma
pltree(cluster_aglomerativo, cex = 0.6, hang = -1, main = "Dendograma método complete")
rect.hclust(cluster_aglomerativo, k = 9, border = 2:10)#bordes y division de clusters en k = 9
```



Agrupamiento Jerárquico divisivo.

Para este estudio se hará uso de la función diana, diferenciando de que en este caso no tendremos coeficientes aglomerativos sino coeficientes divisivos.

Uno de los parámetros de entrada de la función diana es metrics, que hace referencia a la distancia que utilizaremos para las desimilitudes de los clusters. Vamos a comparar dos distancias: euclidea y manhattan, y dependiendo del mejor resultado nos quedaremos con ese cluster.

Cluster usando la distancia euclídea:

```
cluster_diana_euclidea <- diana(students_data, metric = "euclidean")
cluster_diana_euclidea$dc
```

```
## [1] 0.976041
```

Cluster usando la distancia Manhattan:

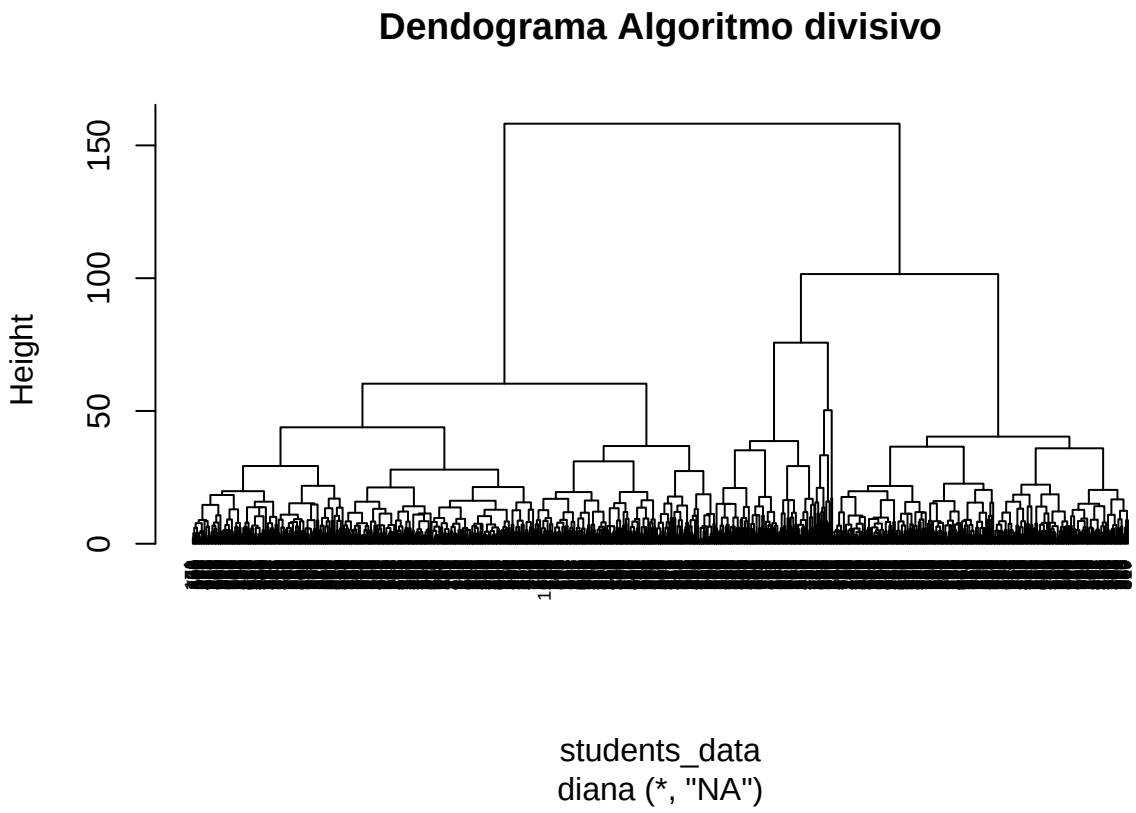
```
cluster_diana_manh <- diana(students_data, metric = "manhattan")
cluster_diana_manh$dc
```

```
## [1] 0.9739787
```

Vemos que el coeficiente de división para el cluster usando distancia euclídea es de 0,976 mientrás que usando la distancia Manhattan es de 0,973. Por lo tanto, usamos la distancia euclídea.

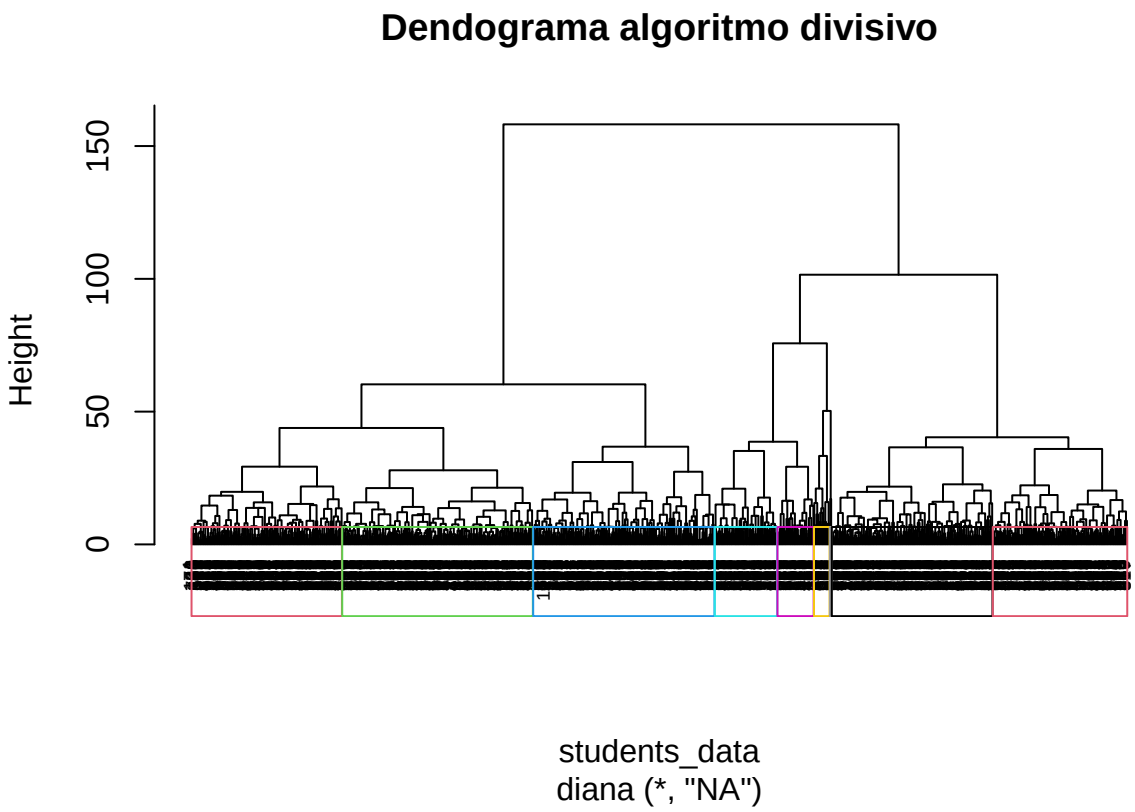
Lo siguiente que haremos es visualizar el dendograma del cluster mediante algoritmo divisivo.

```
#dendograma
pltree(cluster_diana_euclidea, cex = 0.6, hang = -1, main = "Dendograma Algoritmo divisivo")
```



Y, tal y como se hizo anteriormente, se divide en k = 9 cluster marcandolos mediante borders. Tal y como se puede apreciar:

```
#dendograma
pltree(cluster_diana_euclidea, cex = 0.6, hang = -1, main = "Dendograma algoritmo divisivo")
rect.hclust(cluster_diana_euclidea, k = 9, border = 2:10)#bordes y division de clusters en k = 9
```



Podemos comparar los dos dendogramas uno frente al otro y con sus etiquetas relacionadas entre si (algoritmo aglomerativo y divisivo). Para ello, haremos uso de la libreria dendextend.

```
library(dendextend)
```

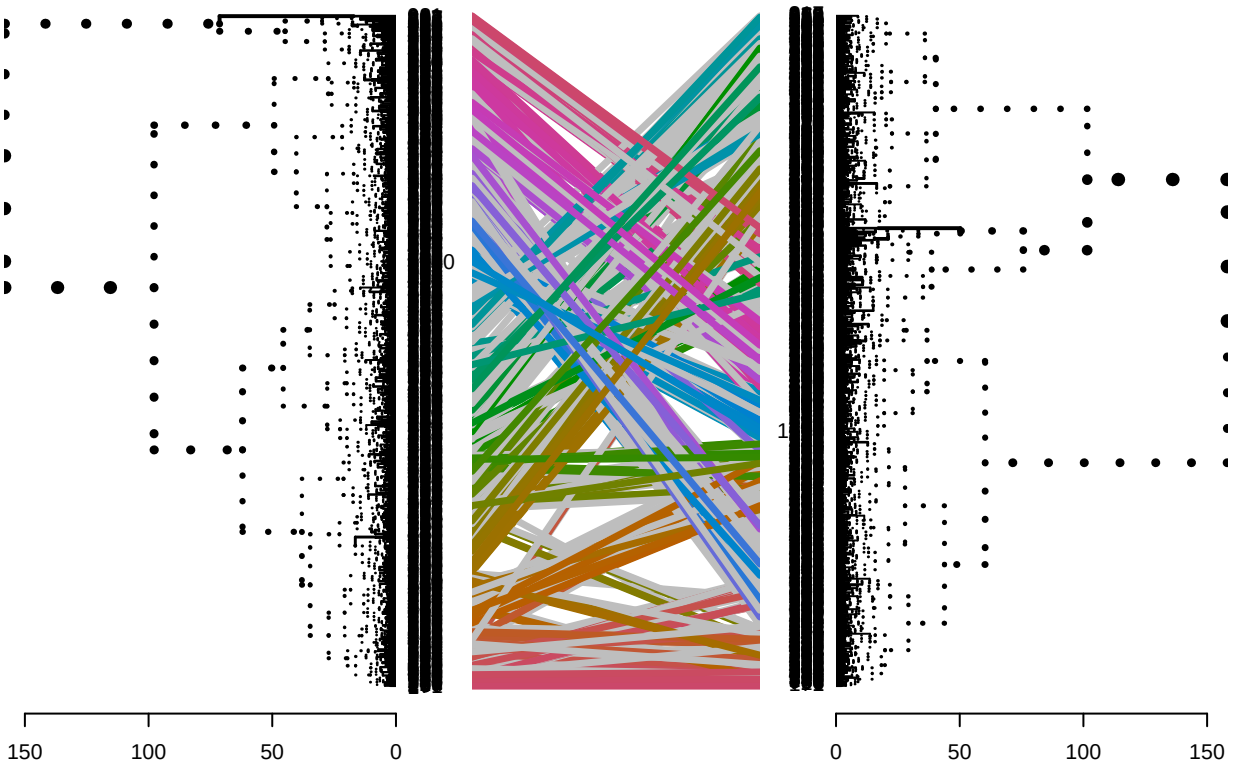
```
##
## ---------------------
## Welcome to dendextend version 1.15.2
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------

##
## Attaching package: 'dendextend'

## The following object is masked from 'package:stats':
##
##     cutree
```

Convertimos los objetos clusters a objetos dendograma.

```
algoritmo_aglomerativo <- as.dendrogram(cluster_aglomerativo)
algoritmo_divisivo <- as.dendrogram(cluster_diana_euclidea)
```

Usamos la funcion tanglegram() de R, para visualizar esta comparativa.

```
tanglegram(algoritmo_aglomerativo, algoritmo_divisivo )
```



Como podemos ver, a simple vista no son muy similares los dos árboles. Para compararlo un poco más, calculamos la matriz de correlación de Baker y Cophenetic. Esto simplemente es para ver la similitud entre árboles. El valor que devuelvan como resultado puede variar en un intervalo de -1 a 1. Cuando los valores son cercanos a 0 significa que los árboles no son estadísticamente similares.

Coeficiente de correlación Cophenetic:

```
cor_cophenetic(algoritmo_aglomerativo, algoritmo_divisivo)
```

```
## [1] 0.3953647
```

Coeficiente de correlación Baker:

```
cor_bakers_gamma(algoritmo_aglomerativo, algoritmo_divisivo)
```

```
## [1] 0.4223358
```

Como podemos ver, el valor que devuelven ambas correlaciones son cercanas al valor de 0. Por lo que, confirmamos lo visto con la función tanglegram de que ambos árboles son muy diferentes entre sí.

Referencias.

https://www.datanovia.com/en/lessons/comparing-cluster-dendrograms-in-r/

https://rpubs.com/mjimcua/clustering-jerarquico-en-r

https://rpubs.com/jaimeisaacp/760355