

# Problemas de Clustering

1. Consideremos la tabla de datos `worldcup` del paquete `faraway` que nos información sobre los jugadores de Fútbol que participaron en el Mundial de Fútbol celebrado el año 2010 en Sudáfrica. Esta tabla de datos da información de 595 jugadores y tiene 7 variables:
  - **Team:** el país del jugador.
  - **Position:** la posición en que juega el jugador. Tiene 4 valores:
    - **Defender:** defensa.
    - **Forward:** delantero.
    - **Goalkeeper:** portero.
    - **Midfielder:** medio.
  - **Time:** tiempo jugado en minutos.
  - **Shots:** número de tiros que ha realizado el jugador.
  - **Passes:** número de pases del jugador.
  - **Tackles:** número de entradas del jugador.
  - **Saves:** número de paradas del jugador.
  - a) Seleccionar una muestra de 25 jugadores usando la función `sample`. Escribir `set.seed(2020)` antes de elegir la muestra.
  - b) Aplicar el algoritmo k-means a la muestra anterior usando las variables cuantitativas para clasificar a los 25 jugadores en 4 grupos usando el algoritmo de MacQueen. Aplicar la función `kmeans` unas cuantas veces con el fin de que la suma de los cuadrados de todos los clusters sea mínima.
  - c) Queremos estudiar hasta qué punto la clasificación anterior coincide con la clasificación de los 25 jugadores según la posición que ocupan. Calcular la tabla bidimensional que nos dé el cluster a qué pertenece el jugador por un lado y la posición a la que juega. ¿Qué porcentaje de aciertos ha tenido el algoritmo k-means?

## Solución

- a) Primero cargamos el paquete y seleccionamos los 25 jugadores:

```
library(faraway)
```

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

```
set.seed(2020)
jugadores.elegidos = sample(1:dim(worldcup)[1],25)
muestra.jugadores = worldcup[jugadores.elegidos,]
```

- b) Aplicamos el algoritmo k-means a la muestra anterior unas 50 veces y vemos cuál es el mínimo de la suma de cuadrados de todos los clusters:

```
veces=100
SSCs=c()
for (i in 1:veces){
  SSCs=c(SSCs,kmeans(muestra.jugadores[,3:7],4,algorithm = "MacQueen")$tot.withinss)
}
(min(SSCs))
```

```
## [1] 54005.42
```

Vemos que el valor mínimo es 53981.01. Ahora realizamos el algoritmo k-means hasta obtener dicho valor:

```
minimo = 53981.01
estudio.kmeans = kmeans(muestra.jugadores[,3:7],4,algorithm = "MacQueen")
while (estudio.kmeans$tot.withinss > minimo +10){
  estudio.kmeans = kmeans(muestra.jugadores[,3:7],4,algorithm = "MacQueen")
}
```

La clasificación de los jugadores ha sido la siguiente:

```
estudio.kmeans$cluster
```

```
## Oh Beom-Seok      Iniesta      Toulalan      Lampard      Kahlenberg      Fabregas
##           4           1           3           1           3           3
## FernandezUF      Birsa      GonzalezC      VeronP      Sim~ao      ColeA
##           4           2           3           4           2           1
## Mun In-Guk      Larsen      Frei      Burdisso      Tamada      Smeltz
##           3           4           4           1           4           2
## Halliche      Bendtner      Parker      Kirm      Doumbia      Bravo
##           2           2           4           2           4           2
## Bassong
##           3
```

c) La tabla bidimensional pedida es la siguiente:

```
table(estudio.kmeans$cluster,muestra.jugadores$Position)
```

```
##
##      Defender Forward Goalkeeper Midfielder
## 1           2           0           0           2
## 2           1           2           1           3
## 3           1           0           0           5
## 4           2           6           0           0
```

Mirando la tabla anterior, podemos clasificar correctamente 14 jugadores clasificando correctamente el 56% de los jugadores de la muestra.

2. Consideremos la tabla de datos `worldcup` del paquete `faraway` que nos información sobre los jugadores de Fútbol que participaron en el Mundial de Fútbol celebrado el año 2010 en Sudáfrica. Esta tabla de datos da información de 595 jugadores y tiene 7 variables:

- **Team:** el país del jugador.
  - **Position:** la posición en que juega el jugador. Tiene 4 valores:
    - **Defender:** defensa.
    - **Forward:** delantero.
    - **Goalkeeper:** portero.
    - **Midfielder:** medio.
  - **Time:** tiempo jugado en minutos.
  - **Shots:** número de tiros que ha realizado el jugador.
  - **Passes:** número de pases del jugador.
  - **Tackles:** número de entradas del jugador.
  - **Saves:** número de paradas del jugador.
- a) Seleccionar una muestra de 25 jugadores usando la función `sample`. Escribir `set.seed(2020)` antes de elegir la muestra.
  - b) Calcular la matriz de distancias de los 25 jugadores anteriores usando la distancia euclídea entre las variables cuantitativas.
  - c) Usando el método jerárquico aglomerativo del **enlace promedio** hallar el dendrograma para clasificar los 25 jugadores anteriores.
  - d) Clasificar los 25 jugadores en 4 clusters a partir del dendrograma anterior.
  - e) Queremos estudiar hasta qué punto la clasificación anterior coincide con la clasificación de los 25 jugadores según la posición que ocupan. Calcular la tabla bidimensional que dos dé el cluster a qué pertenece el jugador por un lado y la posición a la que juega. ¿Qué porcentaje de aciertos ha tenido el algoritmo aplicado?

## Solución

- a) Primero cargamos el paquete y seleccionamos los 25 jugadores:

```
library(faraway)
set.seed(2020)
jugadores.elegidos = sample(1:dim(worldcup)[1],25)
muestra.jugadores = worldcup[jugadores.elegidos,]
```

- b) La matriz de distancias será: (mostramos sólo las 5 primeras filas y columnas)

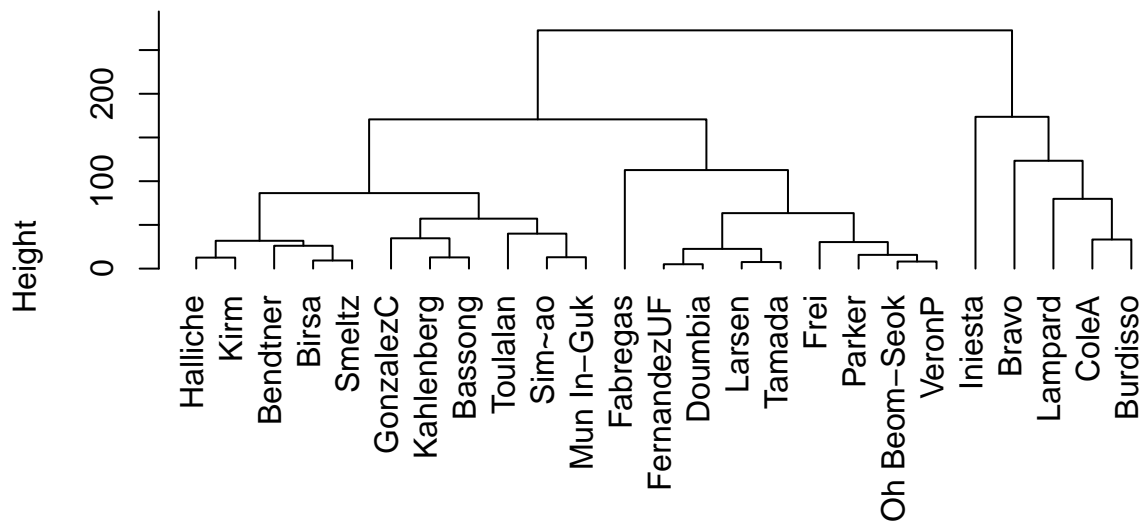
```
distancias.jugadores = as.matrix(dist(muestra.jugadores[,3:7]))
round(distancias.jugadores[1:5,1:5],3)
```

```
##           Oh Beom-Seok Iniesta Toulalan Lampard Kahlenberg
## Oh Beom-Seok      0.000 436.746 110.598 334.969    73.396
## Iniesta          436.746   0.000 326.348 102.426    367.402
## Toulalan         110.598 326.348   0.000 224.704    44.710
## Lampard          334.969 102.426 224.704   0.000    265.351
## Kahlenberg       73.396 367.402  44.710 265.351     0.000
```

- c) El dendrograma pedido será:

```
estudio.clustering = hclust(dist(muestra.jugadores[,3:7]),method="average")
plot(estudio.clustering,hang=-1)
```

## Cluster Dendrogram



```
dist(muestra.jugadores[, 3:7])
hclust (*, "average")
```

d) Los clusters pedidos son los siguientes:

```
(clusters = cutree(estudio.clustering,k=4))
```

```
## Oh Beom-Seok      Iniesta      Toulalan      Lampard      Kahlenberg      Fabregas
##           1           2           3           4           3           1
## FernandezUF      Birsa      GonzalezC      VeronP      Sim~ao      ColeA
##           1           3           3           1           3           4
## Mun In-Guk      Larsen      Frei      Burdisso      Tamada      Smeltz
##           3           1           1           4           1           3
## Halliche      Bendtner      Parker      Kirm      Doumbia      Bravo
##           3           3           1           3           1           4
## Bassong
##           3
```

e) La tabla bidimensional pedida es la siguiente:

```
table(clusters,muestra.jugadores$Position)
```

```
##
## clusters Defender Forward Goalkeeper Midfielder
##           1           2           6           0           1
##           2           0           0           0           1
##           3           2           2           0           7
##           4           2           0           1           1
```

Mirando la tabla anterior, podemos clasificar correctamente 15 jugadores clasificando correctamente el 60% de los jugadores de la muestra.