**[K-Means Clustering in R –](https://www.r-exercises.com/2018/04/13/11256/)** [**Exercises**](https://www.r-exercises.com/2018/04/13/11256/)



K-means clustering algorithm (MacQueen, 1967) es eficiente, y quizás, el método de agrupamiento más popular. Es un algoritmo utilizado para encontrar subgrupos homogéneos en una población. K-means agrupa las observaciones minimizando las distancias euclidianas entre ellas. Las distancias euclidianas pueden extenderse a n dimensiones con cualquier número n, y las distancias se refieren a diferencias numéricas en cualquier variable continúa medida, no solo distancias espaciales o geométricas. Esta definición de distancia euclidiana, por lo tanto, requiere que todas las variables utilizadas para determinar la agrupación usando k-means deben ser continuas. Requerimientos:

* Necesitamos los datos (sin valores NA, y escalados para comparación).
* Número de centros o grupos
* Número de corridas (runs). Comienza al asignar puntos a los grupos de manera aleatoria y puedes encontrar mínimos locales, de modo que ejecutarlo varias veces te ayuda a encontrar el mínimo global.
* Se puede ejecutar K-medias muchas veces para estimar el número de subgrupos cuando no se conoce a priori la forma de encontrar grupos naturales en datos no etiquetados.
* El agrupamiento jerárquico y el no jerárquico pueden ser complementarios a la hora de seleccionar el número de clusters/conglomerados.

Se requerirán las siguientes librerías: **car**, **cluster**, **fpc** y **kableExtra**.

En los siguientes ejercicios trataremos el análisis de conglomerados. Usaremos R para implementar el algoritmo k-means (k-medias) para el análisis de conglomerados y el conjunto de datos DavisThin. Este set de datos tiene 191 filas y 7 columnas y está incluido en el paquete **car**. Estos datos son parte de un conjunto de datos más grande para un estudio de trastornos de alimentación. Las siete variables en el marco de datos comprenden una escala de "unidad para delgadez", que se conformó sumando ciertos elementos durante una investigación.

**Exercise 1**

Cargar los datos DavisThin del paquete **car**. Verificar la presencia de entradas NA y escalar/estandarizar los datos.

**Exercise 2**

Inicialmente lleve a cabo un agrupamiento jerárquico usando la función **hclust()**, utilizar distancias euclidianas y el método “**complete**”, el cual produce árboles más balanceados, graficar el dendograma.

**Exercise 3**

De acuerdo con la interpretación del dendograma, establezca una línea de corte para seleccionar el número de conglomerados posibles. Utilice la función **rect.hclust()** para dibujar rectángulos alrededor de los posibles conglomerados (por ejemplo 5 o 6 clusters). Cortar el árbol por número de clusters e imprimir en pantalla los resultados.

**Exercise 4**

Utilice el siguiente script para determinar el número de conglomerados. Graficar el número de cluster versus el error cuadrático intra-grupos. ¿Cuántos clusters utilizar?

**Exercise 5**

Generar un primer modelo con la función **k-means()** con el número de clusters seleccionado en el ejercicio anterior. Utilizar el algoritmo "**Hartigan-Wong**" por defecto en la función k-means. Imprima en pantalla los resultados. Adicionar una nueva columna al dataframe original con el número del cluster.

**Exercise 6**

Utilizar la función **clusplot()** del paquete **cluster** para visualizar e interpretar los resultados del algoritmo k-means.

**Exercise 7**

Comparar los resultados de los algoritmos hclust() y kmeans(). Utilizar el mismo cut-off (clusters) para ambos.