Instalación y carga de paquetes de R packages

1. Instalación de paquetes

- **fpc, dbscan**: para utilizar algunos algoritmos de clustering
- **factoextra**: para visualización de clusters

2. Lectura de datos

a. Lee el conjunto de datos multishapes del paquete factoextra
b. Quédate con las dos primeras variables.

3. Agrupación con el algoritmo dbscan

a. Utiliza la función dbscan de los paquetes fpc y dbscan para realizar clustering. Observa el objeto que retornan. ¿Producen el mismo agrupamiento?

b. Dibuja los resultados utilizando la función fviz_cluster (En el siguiente ejemplo db es el objeto que retorna dbscan y df un data.frame con los datos)

```
# Plot DBSCAN results
library("factoextra")
fviz_cluster(db, data = df, stand = FALSE,
       ellipse = FALSE, show.clust.cent = FALSE,
       geom = "point",palette = "jco", ggtheme = theme_classic())
```

c. Normaliza los datos con la función scale(), agrúpalos nuevamente con dbscan y observa los resultados.
    a. ¿Se produce el mismo agrupamiento?
    b. ¿Cuál crees que es mejor? ¿Por qué?

4. Optimización de parámetros

a. Existen 3 funciones en la librería dbscan para analizar las distancias de los k-vecinos más próximos.
    a. kNN que retorna, para cada ejemplo, la distancia a sus k-vecinos más próximos (campo dist) y el identificador de esos k-vecinos (campo id).
    b. kNNdist, que calcula las distancias que se utilizan en kNNplot.
    c. kNNplot, que dibuja la distribución de distancias entre datos considerando el k-vecino más cercano. Utilizando kNNplot, estima los valores de épsilon para k=5, 10,20,30,40. ¿Observas alguna tendencia, alguna dependencia entre épsilon y k? Elige para cada k valores de épsilon "no óptimos" (por ejemplo 0.5 y 0.4 para k=5) y observa las

agrupaciones que se producen utilizando la función fviz_cluster (pon el parámetro ellipse a TRUE para que agrupe los clusters)

d. Comprueba que agrupaciones se producen para los valores de épsilon y MinPts estudiados. En el campo *cluster* del objeto que retorna la función dbscan tienes las asignaciones de cada ejemplo al cluster al que pertenece. Comprueba que ejemplos cambian de cluster o pasan de ser clasificados como ruido a pertenecer a un cluster.

5. Comparación de resultados

a. Ahora vamos a comparar los resultados de este clustering con los de otros métodos. Vamos a utilizar el siguiente algoritmo para realizar esta comparación:
   a. Utiliza kmeans o hclust para obtener el mismo número de clusters que se obtuvieron con dbscan.
   b. Dibuja los clusters obtenidos utilizando
b. Plot clusters using the fviz_cluster function (as in the previous exercise).
c. Retorna la asignación de clusters obtenida por k-means, y calcula la silueta mediante la función silhouette del paquete (cluster).

```
Sil=silhouette(db1$cluster, dist(df)), #db1 objeto que retorna un
método de clustering, dist(df), la distancia entre los ejemplos del
data set
```

d. Dibuja el objeto que retorna (función fviz_silhouette del paquete factoextra).
e. Repite el proceso para los grupos obtenidos con dbscan del apartado 3ª.
f. Compara las siluetas. ¿Cuál es el mejor clustering?