Minería de datos

Sesión 3: Análisis de agrupamiento

Paloma Botella Rocamora
(Paloma.Botella@gmail.com)

Estructura de la sesión.

Estructura de la sesión

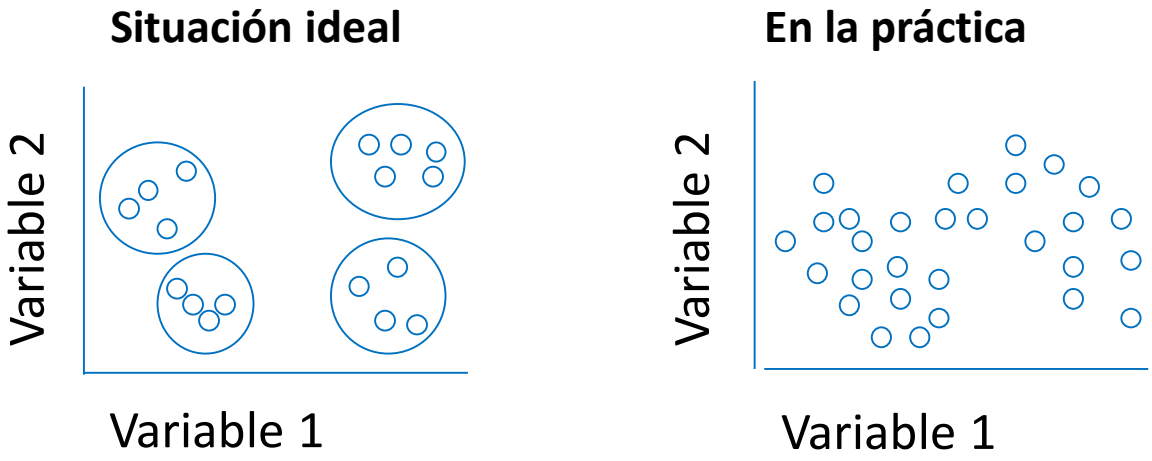
1. *Introducción*
2. *Distancias o Medidas de similitud/disimilitud*
3. *Métodos de Partición (no jerárquicos)*
4. *Métodos jerárquicos*
5. *Resumen final*

1. Introducción

- **Objetivo**: Deseamos **agrupar** objetos o individuos **similares,** basándonos en las variables o características consideradas, de forma que:
  - Los individuos del mismo grupo sean lo más similares posible entre sí.
  - Los individuos de grupos distintos sean lo más diferentes posible entre sí.
- Se trata de un **método no supervisado**, puesto que no hay una variable que nos indique a qué grupo pertenece realmente cada individuo (a diferencia de otras técnicas como el ***análisis discriminante***).
- A los grupos homogéneos que crearemos los denominaremos **agrupaciones** o **clusters**.

1. Introducción

Agrupaciones o clusters:



- La **selección de las variables** en las que se basa la agrupación es también muy importante.
- La inclusión de una o más **variables irrelevantes** puede distorsionar una solución de agrupación que de otra forma podría ser útil.

1. Introducción

- Existen diferentes técnicas para realizar un **análisis cluster**.
- En todas ellas es necesario definir cómo se va a cuantificar la similitud o disimilitud entre las observaciones, es decir, lo "**similares**" o no que son dos individuos.
- Se suele hablar, en general, de **distancia** para definir la forma de medir la idea de similitud/disimilitud entre observaciones.
- El investigador **puede escoger la distancia más adecuada** en función del estudio en cuestión.

1. Introducción

- Existen **diferentes medidas de distancia** que tendremos que elegir en función de:
  - nuestro objetivo
  - tipo de variables en nuestro banco de datos
  - características de los individuos (existencia de outliers, …)

- También existen **diferentes tipos de métodos** y diferentes métodos dentro de cada tipo para obtener los clusters. Usar diferentes métodos nos permitirá comparar los resultados de unos y otros y nos permitirá tener mayor seguridad en los resultados obtenidos.

1. Introducción

Los tipos de métodos más populares para crear grupos:

- **Métodos de partición**: Requieren que el usuario proponga previamente el número de clusters. **Dividen** los individuos en grupos disjuntos.

- **Métodos jerárquicos**: En la modalidad *ascendente* o *aglomerativa*, parten de tantos clusters como individuos y van agrupando casos similares, formando una estructura jerárquica hasta llegar a un solo grupo. Existe también la modalidad *descendente* o *divisiva*.

- ***Otros métodos***: algunos combinan o modifican los anteriores, métodos basados en modelos de mixturas,…

1. Introducción

Secuencia lógica al realizar un análisis de agrupación:

1. Partimos de un banco de datos con **n individuos** y **p variables**.
2. Establecemos un **indicador** que nos diga **en qué medida se parece cada par de individuos** en base a sus observaciones (*distancia*)
3. Elegimos **un método** y se **crean los grupos** con **aquellas observaciones que más se parezcan entre sí**.
4. Una vez obtenidos los grupos **el investigador debe tratar de describir los grupos que ha obtenido** y comparar los unos con los otros a partir de los valores de las variables en cada uno de ellos (*por ejemplo obtener los valores promedio de las variables en cada grupo*)

2. Distancias o medidas de similitud/disimilitud

Si tenemos **p** variables (supongamos cuantitativas), cada individuo es **un punto** en un espacio de **p dimensiones**.

¿Cómo podemos cuantificar lo "cerca" o "lejos" que están los diferentes puntos (individuos)?

Grado de proximidad entre dos puntos en un espacio de dimensión p

Distancia (o **métrica**)

2. Distancias o medidas de similitud/disimilitud

Dados dos vectores x_i y x_j del espacio R^p se define una **función distancia** d entre ellos como cualquier función que cumpla las siguientes propiedades:

1. $d(x_i, x_j) \geq 0$ [valor positivo o 0]
2. $d(x_i, x_i) = 0$ [la distancia entre un elemento y sí mismo es 0]
3. $d(x_i, x_j) = d(x_j, x_i)$ [simétrica]
4. $d(x_i, x_j) \leq d(x_i, x_p) + d(x_p, x_j)$ [propiedad triangular]

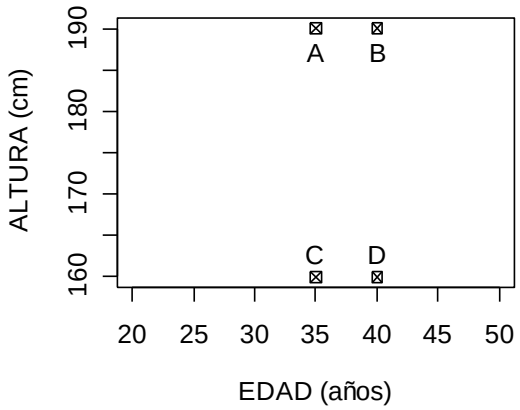
La **elección** de la **escala** de las variables y la **distancia** entre los individuos es **crucial** en el **análisis cluster**.

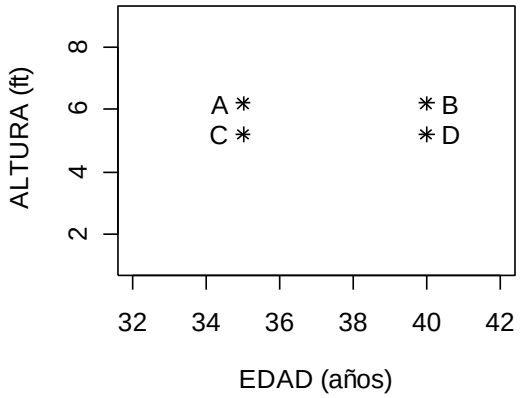
2. Distancias o medidas de similitud/disimilitud

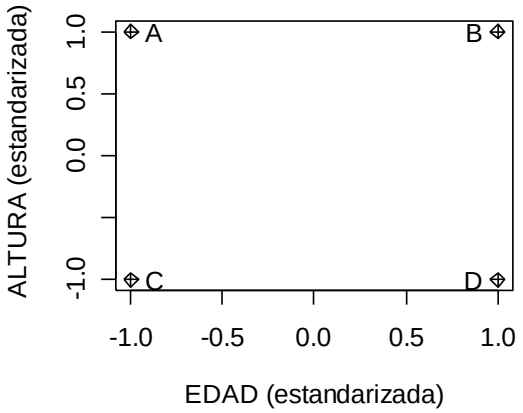
La **elección** de la **escala** de las variables y la **distancia** entre los individuos es **crucial** en el **análisis cluster**.

(*Recordad ejemplo*)

Persona	*Edad (años)*	*Altura (cm)*	*Altura (ft)*	*Edad (estand)*	*Altura (estand)*
A	35	190	6.2	-1	1
B	40	190	6.2	1	1
C	35	160	5.2	-1	-1
D	40	160	5.2	1	-1







2. Distancias o medidas de similitud/disimilitud

Para solucionar los problemas de **escala**:

*Posibles **transformaciones** de las variables cuantitativas:*

- Pasar a puntuaciones z.

$$z_v = \frac{y_v - m_v}{s_v} \quad con\ m_v = \frac{1}{n}\sum_{i=1}^{n} y_{iv} \quad s_v^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_{iv} - m_v)^2$$

Conseguimos media 0 y desviación típica 1

Puedes usar la función `scale` de R.

- Máximo valor en 1

Dividiendo por el valor máximo

- Rango de 0 a 1

Restando el mínimo y dividiendo por el valor máximo

2. Distancias o medidas de similitud/disimilitud

Como en el ACP podemos preguntarnos:

¿Conviene estandarizar los datos o no?

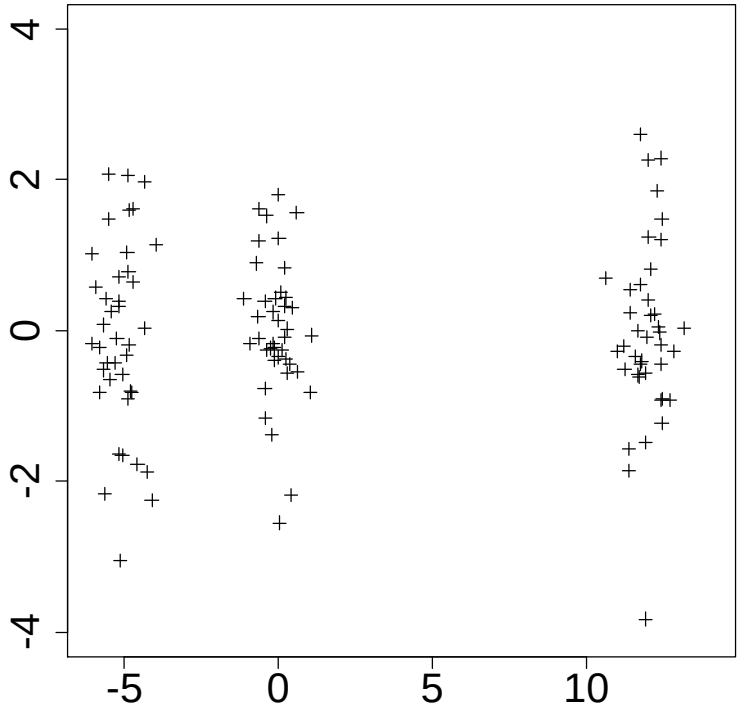
Si no estandarizamos la **distancia euclídea dependerá** sobre todo de las **variables con valores más grandes** y el resultado puede cambiar radicalmente al modificar la escala de medida.

Si estandarizamos damos un **peso semejante** (a priori) a **todas las variables**, independientemente de su variabilidad original, lo que puede no ser adecuado.

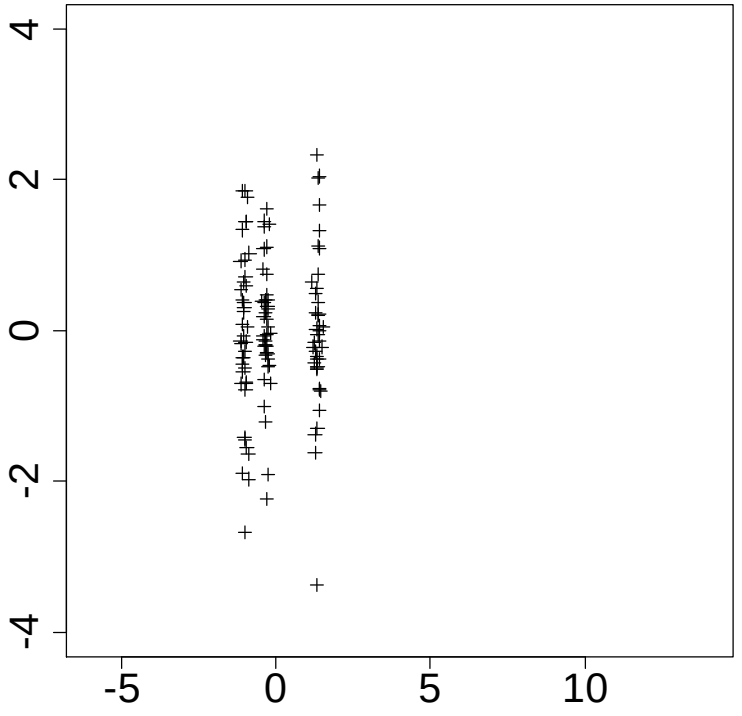
Conviene **tener en cuenta el objetivo del estudio** y, en general, si las variables están medidas en las mismas unidades se suele recomendar no estandarizar.

2. Distancias o medidas de similitud/disimilitud

¿Conviene estandarizar?



No estandarizados



Estandarizados

2. Distancias o medidas de similitud/disimilitud

Ejemplos de **distancias** (para **variables cuantitativas**):

Euclídea	$d_E(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{v=1}^{p}(x_v - y_v)^2} = \sqrt{(\mathbf{x}-\mathbf{y})'(\mathbf{x}-\mathbf{y})}$
Euclídea al cuadrado	$d_{E^2}(\mathbf{x},\mathbf{y}) = \sum_{v=1}^{p}(x_v - y_v)^2 = (\mathbf{x}-\mathbf{y})'(\mathbf{x}-\mathbf{y})$
Chebychev ó Dominante	$d_{m_\infty}(\mathbf{x},\mathbf{y}) = \max\left(\lvert x_1 - y_1\rvert, \lvert x_2 - y_2\rvert, .., \lvert x_p - y_p\rvert\right)$
Manhattan ó Ciudad ó City-block	$d_{m_1}(\mathbf{x},\mathbf{y}) = \sum_{v=1}^{p}\lvert x_v - y_v\rvert$
Potencia ó Power	$d_{P_{t,r}}(\mathbf{x},\mathbf{y}) = \left(\sum_{v=1}^{p}\lvert x_v - y_v\rvert^t\right)^{1/r}$

$t = r \rightarrow Minkowski$

$t = r = 1 \rightarrow Ciudad$

$t = r = 2 \rightarrow Euclídea$

$t = 2, r = 1 \rightarrow Euclídea\ al\ cuadrado$

$t = \infty, r = \infty \rightarrow Chebychev$

2. Distancias o medidas de similitud/disimilitud

Minkowski	$d_{m_q}(\mathbf{x},\mathbf{y}) = \left(\sum_{v=1}^{p} \lvert x_v - y_v \rvert^q \right)^{1/q}$
Canberra	$d_C(\mathbf{x},\mathbf{y}) = \sum_{v=1}^{p} \frac{\lvert x_v - y_v \rvert}{\lvert x_v \rvert + \lvert y_v \rvert}$
K. Pearson	$d_{K^2}(\mathbf{x},\mathbf{y}) = \sum_{v=1}^{p} \frac{(x_v - y_v)^2}{s_v^2} \quad con \;\; s_v^2 = \mathrm{var}(X_v)$
Mahalanobis	$d_{M^2}(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})'\mathbf{S}^{-1}(\mathbf{x}-\mathbf{y}) \;\; con \;\; \mathbf{S} = \mathrm{var}(\mathbf{X})$

- Muchas de estas distancias están en la función `dist` de R.
- Y en la función `distance` de la librería `philentropy` puedes encontrar muchas más, todas las que te muestra la función `getDistMethods()`

2. Distancias o medidas de similitud/disimilitud

¿Qué **distancias** podemos usar?

Para variables continuas (estandarizadas univariantemente o no) la distancia más utilizada es la **distancia euclídea**.

$$d_E(x, y) = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}$$

Si queremos magnificar las distancias entre puntos más alejados se puede usar también la **distancia euclídea al cuadrado** (menos exigente computacionalmente).

2. Distancias o medidas de similitud/disimilitud

¿Qué **distancias** podemos usar?

Si hay outliers en los datos, la distancia euclídea se verá afectada más que la **distancia Manhattan**.

$$d_M(x, y) = \sum_{i=1}^{p} |x_i - y_i|$$

2. Distancias o medidas de similitud/disimilitud

¿Qué **distancias** podemos usar?

Otro tipo de distancia utiliza el **coeficiente de correlación** entre los valores de todas las variables de cada par de individuos.

$$d_{cor}(x, y) = 1 - cor(x, y)$$

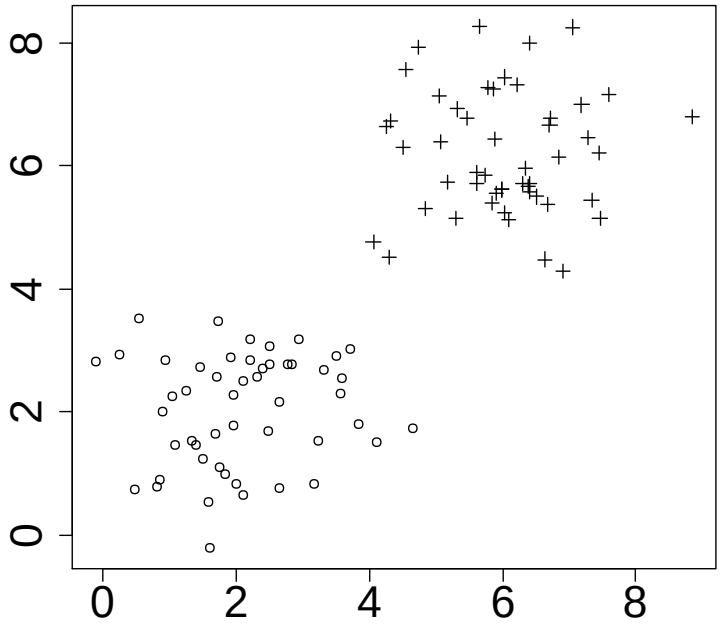
Este tipo de distancias valora la similitud de patrón y no la magnitud en sí.

$$d_{cor}\big((1,2,3,4,5), (10,20,30,40,50)\big) = 1 - 1 = 0$$

2. Distancias o medidas de similitud/disimilitud

¿Qué **distancias** podemos usar?

En cuanto a la **distancia de Mahalanobis** no en todos los casos es recomendable su uso (las correlaciones entre variables de la matriz de datos pueden no corresponderse con esas mismas correlaciones en cada grupo), aunque en otras ocasiones puede funcionar mejor.



En este caso las variables están incorreladas en cada grupo, aunque al trabajar con la nube de puntos completa se obtiene una alta correlación.

2. Distancias o medidas de similitud/disimilitud

¿Distancia de Mahalanobis?

Si las variables no han sido estandarizadas, la distancia de mahalanobis puede ser adecuada, ya que a partir de la varianza controla las diferencias de escala de unas y otras.

Además, cuando hay mucha correlación entre las variables, la distancia euclídea puede magnificar la distancia entre observaciones utilizando las variables con información redundante. La distancia de mahalanobis puede funcionar mejor en este caso (esta solución sería equivalente a trabajar directamente con distancia euclídea sobre las componentes principales).

2. Distancias o medidas de similitud/disimilitud

Hasta ahora hemos dado por hecho que nuestras variables eran cuantitativas. ¿Qué **distancias** podemos usar cuando no es así?

- La problemática se complica cuando en la muestra existen **variables continuas** y **cualitativas**: la **distancia euclídea** dará **mayor peso** a las **variables continuas** (a pesar de estar estandarizadas) que a las binarias.

- Este **hecho puede ser aceptable** en muchos casos, **pero cuando** teniendo en cuenta la naturaleza del problema de estudio **no lo sea** la solución es trabajar con **distancias / disimilaridades** adecuadas para variables binarias, categóricas,…

2. Distancias o medidas de similitud/disimilitud

Ejemplos de **distancias** (para **variables binarias 0-1**):

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}
x	0	0	1	0	1	1	1	0	1	0	0	1
y	0	1	1	0	0	1	1	0	0	0	1	0

		y	
		1	0
x	1	A=3	B=3
	0	C=2	D=4

2. Distancias o medidas de similitud/disimilitud

- En las variables ***binarias simétricas***, el **0** y el **1** representan dos estados de forma indistinta.

Por ejemplo sexo (0=Chico, 1=Chica) o tipo de diálisis (0=Peritoneal, 1=Hemodiálisis)

Dos individuos serán más diferentes (disimilares) cuantas más variables de este tipo tienen distintas. El valor 0 y el valor 1 tienen la misma importancia.

% coincidencias	$s_{\%c}(\mathbf{x},\mathbf{y}) = \frac{A+D}{A+B+C+D}$	; $d_{\%c}(\mathbf{x},\mathbf{y}) = 1 - s_{\%c}(\mathbf{x},\mathbf{y})$
Roger y Tanimoto	$s_{RT}(\mathbf{x},\mathbf{y}) = \frac{A+D}{A+D+2(B+C)}$	; $d_{RT}(\mathbf{x},\mathbf{y}) = 1 - s_{RT}(\mathbf{x},\mathbf{y})$
Sokal y Sneath	$s_{SSs}(\mathbf{x},\mathbf{y}) = \frac{2(A+D)}{2(A+D)+B+C}$	; $d_{SSs}(\mathbf{x},\mathbf{y}) = 1 - s_{SSs}(\mathbf{x},\mathbf{y})$

2. Distancias o medidas de similitud/disimilitud

- Para variables **binarias asimétricas:** el **1** representa la presencia del carácter. Cuando los dos individuos son 1 indica mayor semejanza entre los mismos que cuando los dos son cero.

(<<Una cabra y un libro no se parecen por NO tener ruedas>>).

Jaccard	$s_{Jac}(\mathbf{x},\mathbf{y}) = \frac{A}{A+B+C}$	; $d_{Jac}(\mathbf{x},\mathbf{y}) = 1 - s_{Jac}(\mathbf{x},\mathbf{y})$
Dice y Sorensen	$s_{DS}(\mathbf{x},\mathbf{y}) = \frac{2A}{2A+B+C}$	; $d_{DS}(\mathbf{x},\mathbf{y}) = 1 - s_{DS}(\mathbf{x},\mathbf{y})$
Sokal y Sneath	$s_{SSa}(\mathbf{x},\mathbf{y}) = \frac{A}{A+2(B+C)}$	; $d_{SSa}(\mathbf{x},\mathbf{y}) = 1 - s_{SSa}(\mathbf{x},\mathbf{y})$

2. Distancias o medidas de similitud/disimilitud

- Para variables ***nominales*** con varios posibles estados (ojos: 1=marrones, 2=azules, 3=verdes, 4=otro color.) se utiliza:

  (1- proporción de coincidencias entre todas las variables nominales)

- Para variables ***ordinales*** se suele ***pasar a un valor cuantitativo*** y se utiliza una distancia para este tipo de variables. Se puede pasar el rango del valor estandarizado para que queden valores entre 0 y 1:

$$z_{iv} = \frac{r_{iv} - 1}{M_v - 1} \quad con \begin{cases} r_{iv} = Rango\,de\,la\,categoría\,del\,caso\,i\,en\,X_v \\ M_v = Max\,del\,rango\,de\,las\,categorías\,de\,X_v \end{cases}$$

2. Distancias o medidas de similitud/disimilitud

- Muchas de las distancias comentadas están en la función `dist` de la librería `stats` de R.

```
dist(x, method = "euclidean", diag = FALSE, upper =
FALSE, p = 2)
```

("euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski")

- Otra función interesante es la función `get_dist()` de la librería `factoextra`.

("euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski", "pearson", "spearman" or "kendall")

Interesante la función `fviz_dist` que nos muestra gráficamente la matriz de distancias entre individuos agrupando aquellos más cercanos

2. Distancias o medidas de similitud/disimilitud

- Y en la función `distance` de la librería `philentropy` puedes encontrar muchas más, todas las que te muestra la función `getDistMethods()`

```
("euclidean", "manhattan", "minkowski", "chebyshev",
"sorensen", "gower", "soergel", "kulczynski_d", "canberra",
"lorentzian", "intersection", "non-intersection",
"wavehedges", "czekanowski", "motyka", "kulczynski_s",
"tanimoto", "ruzicka", "inner_product", "harmonic_mean",
"cosine",  "hassebrook", "jaccard", "dice", "fidelity",
"bhattacharyya", "hellinger", "matusita", "squared_chord",
"squared_euclidean", "pearson",  "neyman", "squared_chi",
"prob_symm", "divergence", "clark", "additive_symm",
"kullback-leibler", "jeffreys", "k_divergence", "topsoe",
"jensen-shannon", "jensen_difference", "taneja", "kumar-
johnson", "avg")
```

2. Distancias o medidas de similitud/disimilitud

Hemos visto ejemplos de disimilaridades o distancias para cada tipo de variable.

Pero…***¿Cómo podemos combinar algunas de ellas en una única medida de disimilaridad?.***

Gower (1971) propone una medida de disimilaridad que combina medidas de distancia/disimilaridad para distintas variables y que además tiene en cuenta los posibles datos faltantes.

Por ejemplo, la función `daisy` del paquete `cluster` la tiene implementada.

2. Distancias o medidas de similitud/disimilitud

Ejemplos de **distancias** (para bancos de datos con diferentes tipos de **variables: variables cuantitavias y/o categóricas**):

Gower (1971) propone la siguiente medida de disimilaridad que tiene en cuenta los posibles datos faltantes:

$$d_G(\mathbf{x},\mathbf{y}) = \frac{\sum_{v=1}^{p} w_{x,y}^{v} \cdot d(x_v, y_v)}{\sum_{v=1}^{p} w_{x,y}^{v}}$$

$$\begin{cases} w_{x,y}^{v} = 1 \;\; si \;\; x_v, y_v \;\; son\ datos\ NO\ \ faltantes. \\ w_{x,y}^{v} = 0 \;\; si \;\; x_v \;\; ó \;\; y_v \;\; son\ datos\ \ faltantes. \\ w_{x,y}^{v} = 0 \;\; si \;\; v = binaria\ asimétrica\ y\ x_v = 0 \;\; y \;\; y_v = 0 \end{cases}$$

$$\begin{cases} si \;\; v = binaria\ ó\ nominal \begin{cases} d(x_v, y_v) = 1\ si \;\; x_v \neq y_v \\ d(x_v, y_v) = 0\ si \;\; x_v = y_v \end{cases} \\ si \;\; v = cuantitativa \begin{cases} d(x_v, y_v) = \dfrac{|x_v - y_v|}{R_v} \quad con \quad R_v = \max\limits_{i=1,..,n}(x_{i,v}) - \min\limits_{i=1,..,n}(x_{i,v}) \end{cases} \\ si \;\; v = ordinal, \;\; pasar \;\; X_v \;\; a\ su\ rango\ r_v \;\; y\ luego\ como\ cuantitativa. \end{cases}$$

3. Métodos de partición

- **Agrupan los individuos** en un número de clusters o agrupaciones, **k**, previamente fijado.
- Se parte de un **conjunto inicial de k clusters** que van **cambiando de modo iterativo** (existen diferentes propuestas para seleccionar los clusters iniciales) hasta un número máximo de iteraciones o hasta la estabilización de las agrupaciones.
- El **método más habitual** es el de las **k-medias**, conocido como ***k-means***.
- Trata de obtener los **k mejores clusters**, entendiendo que son mejores aquellos que tienen **menor varianza interna** (diferentes métodos establecen diferentes formas de medir esta varianza interna).

3. Métodos de partición

Algoritmo de K-means *(variables cuantitativas)*

Preestablecido el número *k* de clases (**clusters**) que se quieran formar, se obtiene una **partición del conjunto** de individuos (filas en el banco de datos) **en *k* grupos** disjuntos y exhaustivos.

- *Eficiente cuando hay un gran número de casos.*
- *Las variables consideradas deben tener un carácter cuantitativo y unas varianzas similares.*

Partimos de un banco de datos, con ***n*** individuos y ***p*** variables.

$$\mathbf{X}_{n\times p} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{pmatrix}$$

3. Métodos de partición

Fundamentos de los algoritmos de k-medias:

1) Seleccionar k puntos como centros de los grupos iniciales:

¿Cómo?

a) considerando k centroides iniciales al azar en el espacio p dimensional

b) asignando aleatoriamente los objetos a los grupos y tomando los centros de los grupos así formados

c) tomando los k primeros individuos como centros iniciales

d) tomando como centros los k puntos más alejados entre sí

e) construyendo unos grupos iniciales con información a priori y calculando sus centros

f) …

3. Métodos de partición

Fundamentos del algoritmo de k-medias:

2) Calcular **las distancias euclídeas** de cada elemento a los centros de los k grupos, y asignar cada elemento al grupo de cuyo centro esté más próximo.

La asignación puede realizarse en bloque o iterativamente (on-line), es decir, secuencialmente, de forma que al introducir un nuevo elemento en un grupo se recalculan las coordenadas del nuevo centro del grupo.

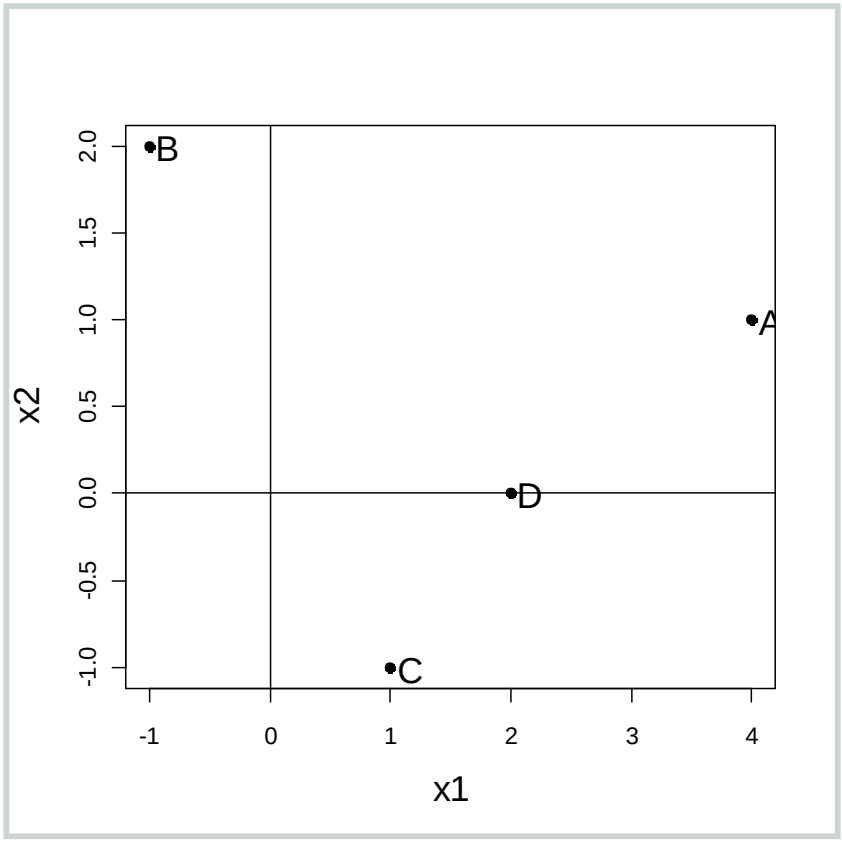
3) Definir un criterio de optimalidad y comprobar si reasignando alguno de los elementos mejora el criterio. Si no es posible mejorar el criterio de optimalidad, terminar el proceso.

3. Métodos de partición

Ejemplo 1: algoritmo de K-Medias (*online*).

Supongamos que tenemos dos variables x1 y x2 y 4 elementos: A, B, C, D:

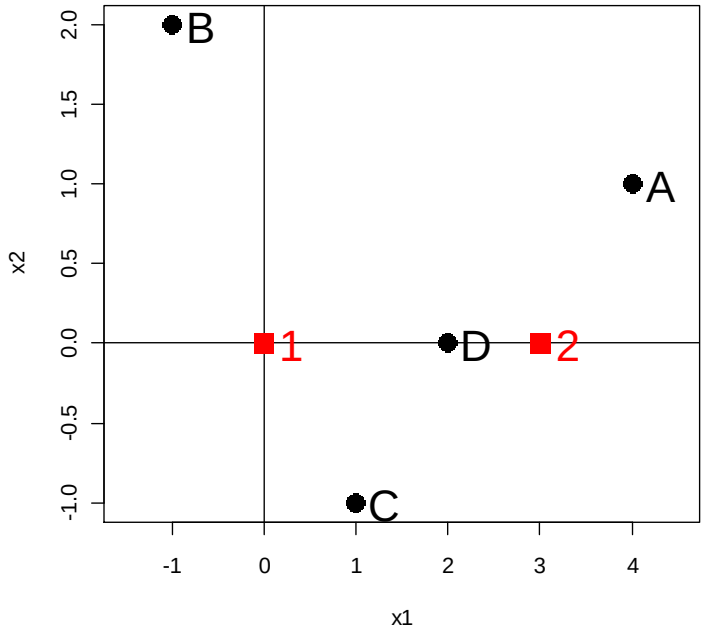
	X1	X2
A	4	1
B	-1	2
C	1	-1
D	2	0



3. Métodos de partición

Ejemplo 1: algoritmo de K-Medias (*online*).

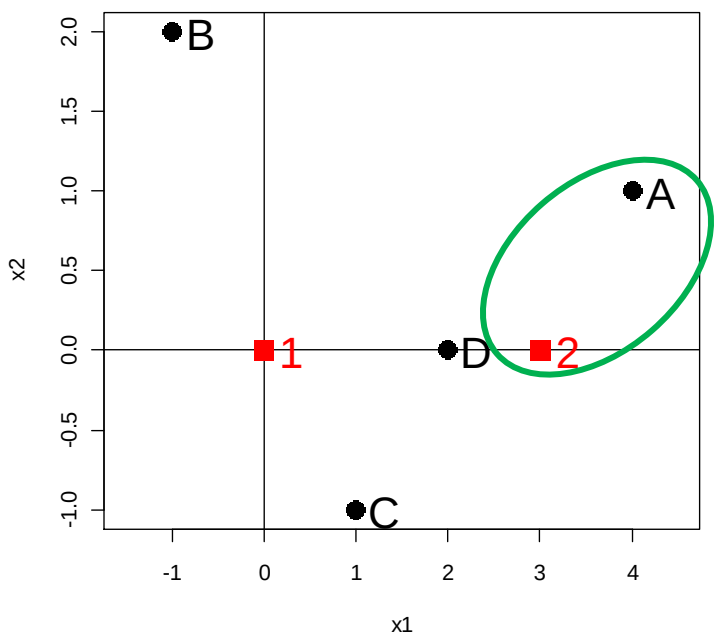
Consideramos dos centroides iniciales: $C_1^0 = (0,0)\,;\; C_2^0 = (3,0)$



3. Métodos de partición

Ejemplo 1: algoritmo de K-Medias (*online*).

Consideramos dos centroides iniciales: $C_1^0 = (0,0)\ ;\ C_2^0 = (3,0)$



Paso -1-

Punto A:

$d(A,c1)^2=17$ y $d(A,c2)^2=2$

Asignamos punto A al centro 2.

Y recalculamos el centro 2.

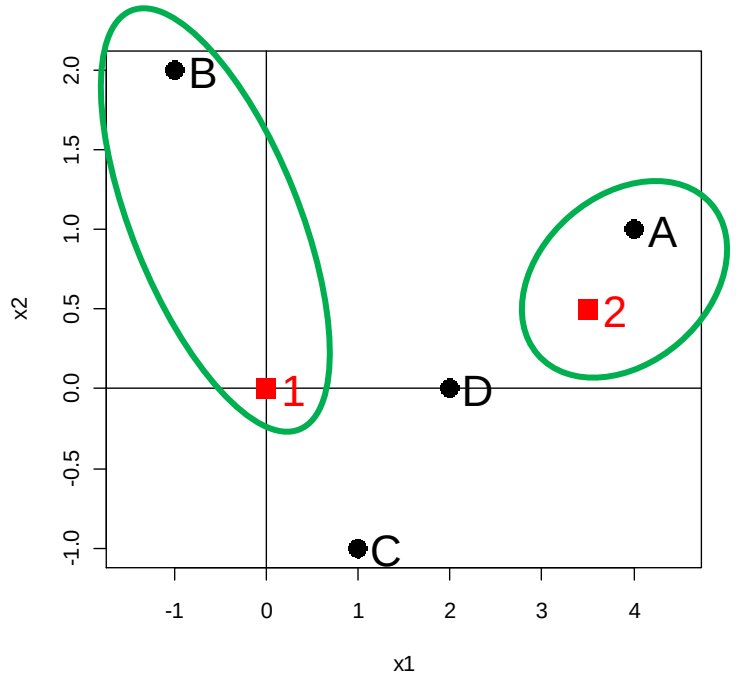
$$C_2^0 = \left(\frac{4+3}{2}, \frac{1+0}{2}\right) = (3.5,0.5)$$

3. Métodos de partición

Ejemplo 1: algoritmo de K-Medias (*online*).

Nuevos centroides:

$$C_1^0 = (0,0)\ ;\ C_2^0 = (3.5,0.5)$$



Paso -1-

Punto B:

$d(B,c1)^2=5$ y $d(B,c2)^2=22.5$

Asignamos punto B al centro 1.

Y recalculamos el centro 1.

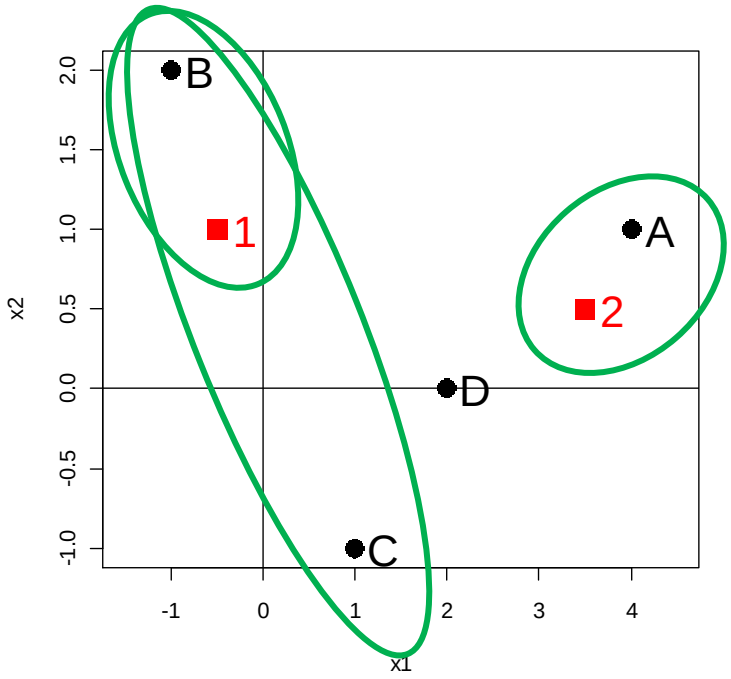
$$C_1^0 = \left(\frac{-1+0}{2}, \frac{2+0}{2}\right) = (-0.5,1)$$

3. Métodos de partición

Ejemplo 1: algoritmo de K-Medias (*online*).

Nuevos centroides:

$$C_1^0 = (-0.5,1)\ ;\ C_2^0 = (3.5,0.5)$$



Punto C:

Paso -1-

$d(C,c1)^2=6.25$ y $d(C,c2)^2=8.5$

Asignamos punto C al centro 1.

Y recalculamos el centro 1.

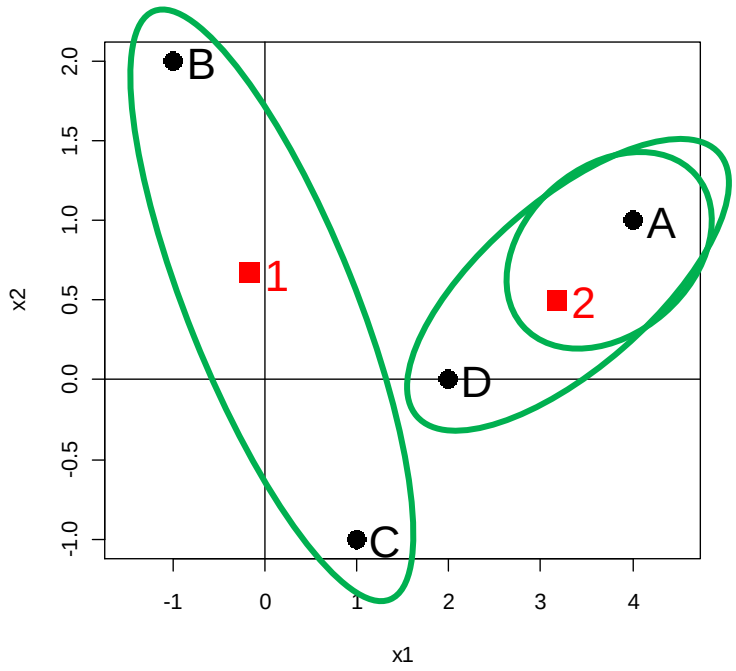
$$C_1^0 = \left(\frac{-0.5-1+1}{3}, \frac{1+2-1}{3}\right) = (-0.17,0.67)$$

3. Métodos de partición

Ejemplo 1: algoritmo de K-Medias (*online*).

Nuevos centroides:

$$C_1^o = (-0.17, 0.67); C_2^o = (3.5, 0.5)$$



Punto D:

Paso -1-

$d(D,c1)^2=5.16$ y $d(D,c2)^2=2.5$

Asignamos punto D al centro 2.

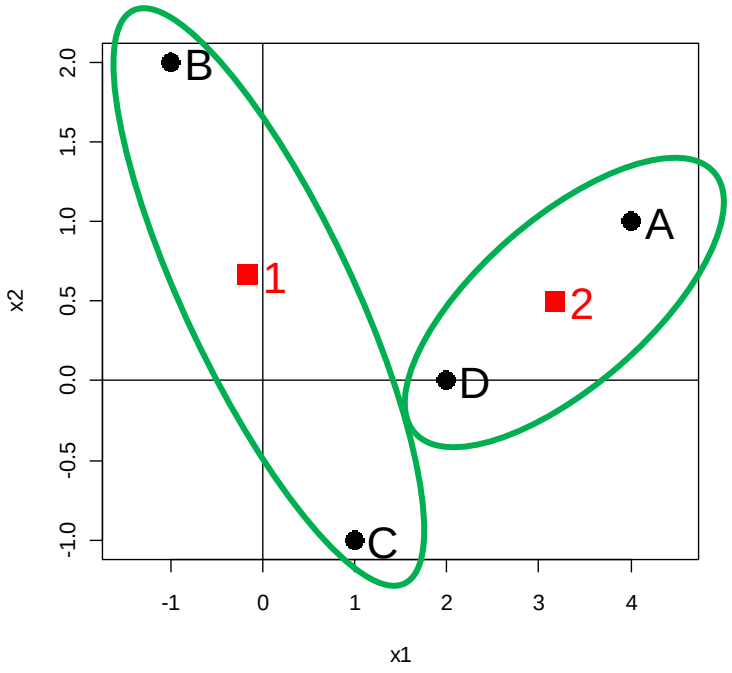
Y recalculamos el centro 2.

$$C_2^0 = \left(\frac{3.5+4+2}{3}, \frac{0.5+1+0}{3}\right) = (3.17, 0.5)$$

3. Métodos de partición

Ejemplo 1: algoritmo de K-Medias (*online*).

Centroides al final paso 1: $C_1^1 = (-0.17, 0.67); C_2^1 = (3.17, 0.5)$



Fin

Paso -1-

3. Métodos de partición

Ejemplo 1: algoritmo de K-Medias (*online*).

Paso -2-

Centroides:

$$C_1^1 = (-0.17, 0.67); C_2^1 = (3.17, 0.5)$$

Calculamos distancias (al cuadrado) de todos los puntos a los centroides y comprobamos que ningún punto cambia de cluster:

	C1	C2
A	4.18	0.96
B	1.57	4.43
C	2.03	2.64
D	2.27	1.27

Centroides finales:

$$C_1^1 = \left(\frac{-1+1}{2}, \frac{2-1}{2}\right) = (0, 0.5)$$

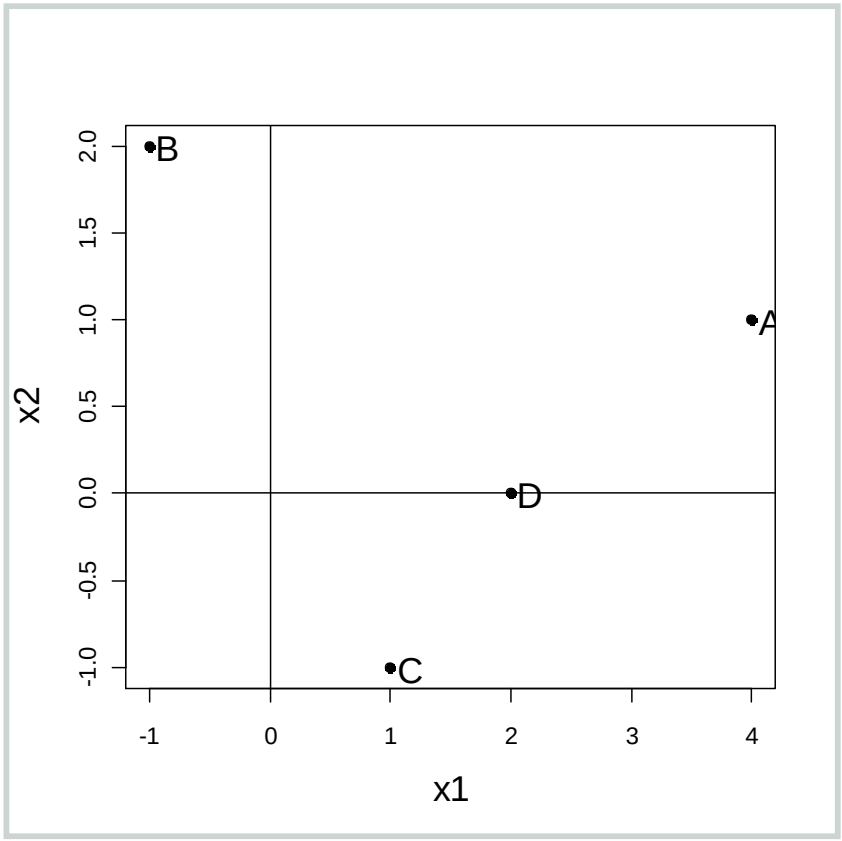
$$C_2^1 = \left(\frac{4+2}{2}, \frac{1+0}{2}\right) = (3, 0.5)$$

3. Métodos de partición

Ejemplo 2: otro algoritmo de K-Medias

Supongamos que tenemos dos variables x1 y x2 y 4 elementos: A, B, C, D:

	X1	X2
A	4	1
B	-1	2
C	1	-1
D	2	0



3. Métodos de partición

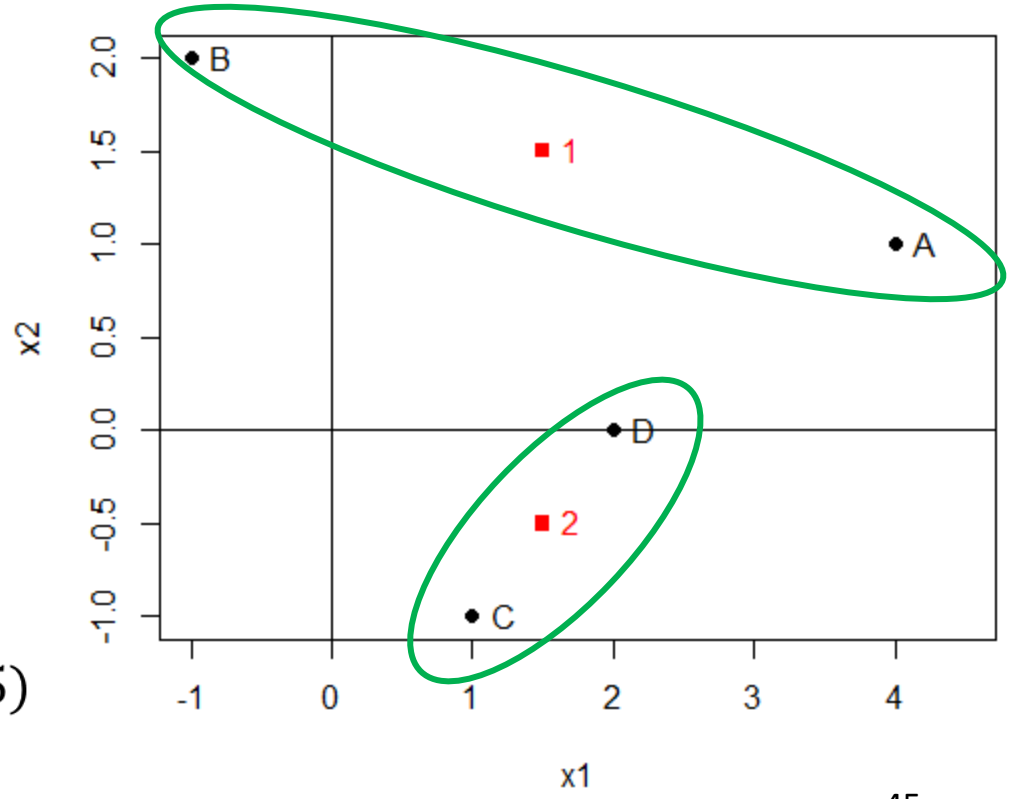
Ejemplo 2: otro algoritmo de K-Medias (asignación aleatoria 1)

Paso 1: Asignamos cada observación a un cluster aleatoriamente y calculamos el centroide de cada cluster.

	X1	X2	G
A	4	1	G1
B	-1	2	G1
C	1	-1	G2
D	2	0	G2

$$C_1^1 = \left(\frac{4-1}{2}, \frac{1+2}{2}\right) = (1.5, 1.5)$$

$$C_2^1 = \left(\frac{1+2}{2}, \frac{-1+0}{2}\right) = (1.5, -0.5)$$

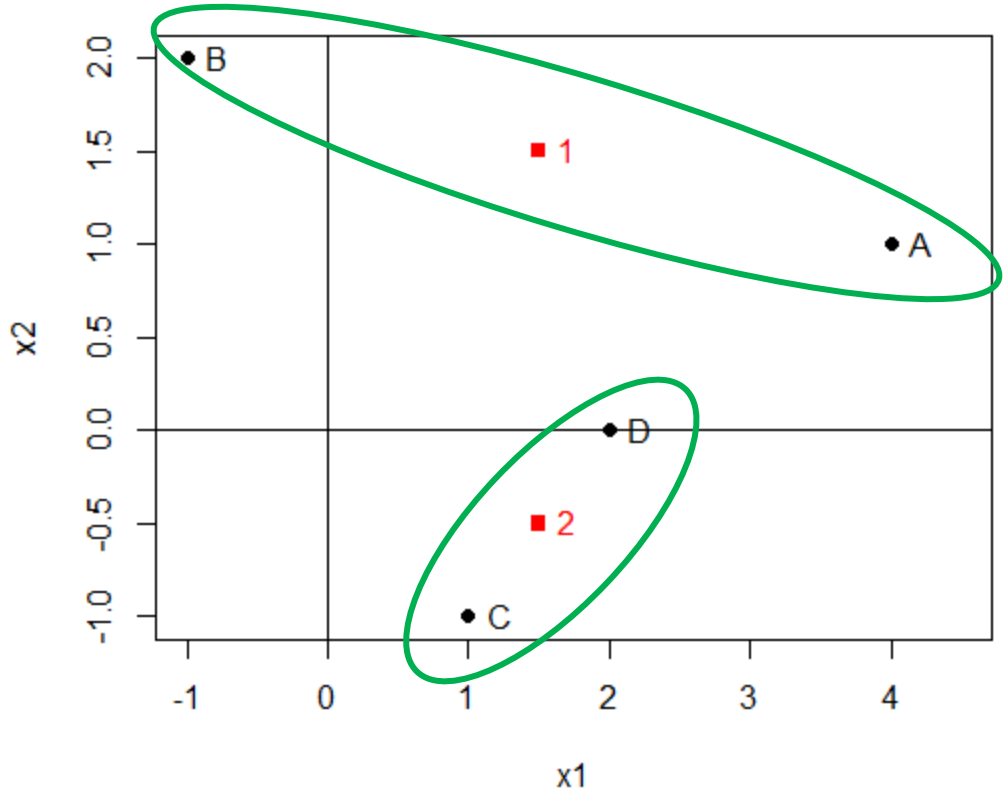


3. Métodos de partición

Ejemplo 2: otro algoritmo de K-Medias (asignación aleatoria 1)

Paso 2: Calculamos la distancia de cada punto a cada centroide y comprobamos que ningún punto se cambiaría de cluster.

	C1	C2
A	2.5	2.9
B	2.5	3.5
C	2.5	0.7
D	1.6	0.7



3. Métodos de partición

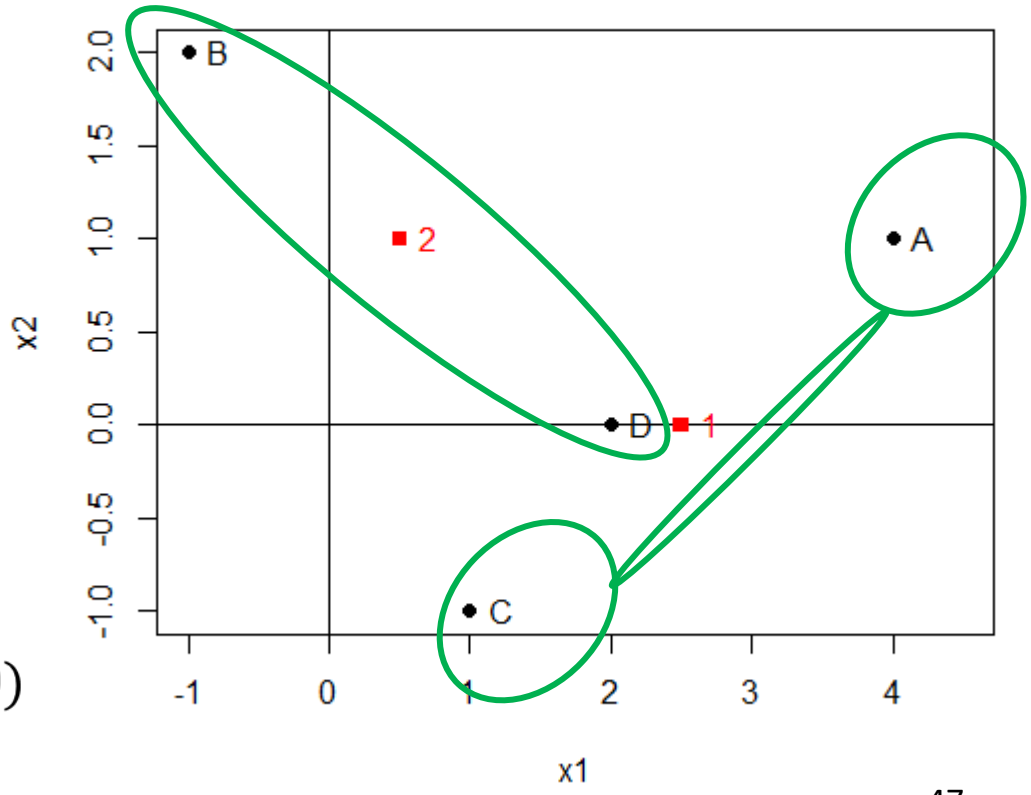
Ejemplo 2: algoritmo de K-Medias (asignación aleatoria 2)

Paso 1: Asignamos cada observación a un cluster aleatoriamente y calculamos el centroide de cada cluster.

	X1	X2	G
A	4	1	G1
B	-1	2	G2
C	1	-1	G1
D	2	0	G2

$$C_1^1 = \left(\frac{4+1}{2}, \frac{1-1}{2}\right) = (2.5, 0)$$

$$C_2^1 = \left(\frac{-1+2}{2}, \frac{2+0}{2}\right) = (0.5, 1.0)$$

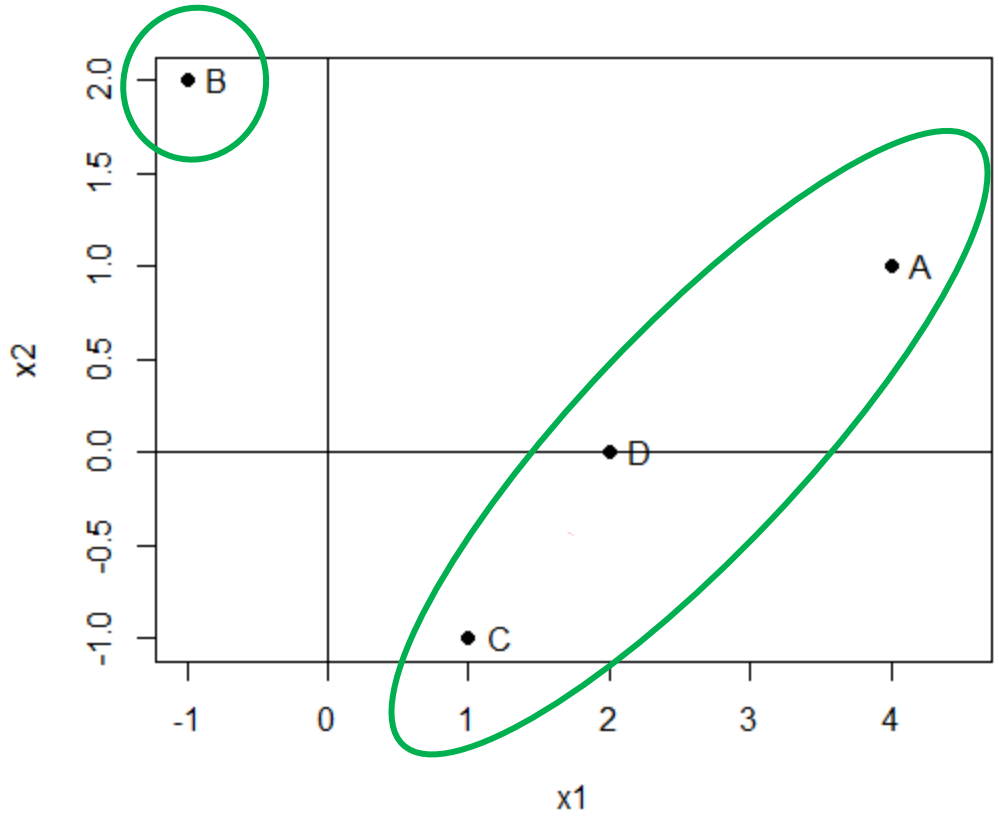


3. Métodos de partición

Ejemplo 2: algoritmo de K-Medias (asignación 2)

Paso 2: Calculamos la distancia de cada punto a cada centroide y comprobamos que la observación D cambiaría de cluster. Si recalculáramos veríamos que esta configuración sería la final

	C1	C2
A	1.8	3.5
B	4.0	1.8
C	1.8	2.1
D	0.5	1.8



3. Métodos de partición

Algoritmo de K-Medias.

- El hecho de **fijar k cluster inciales** lleva consigo determinados problemas:
  - Si dos centroides iniciales caen por casualidad en un único cluster natural, entonces los clusters que resultan están poco diferenciados entre sí.
  - Si aparecen outliers, se obtiene por lo menos un cluster con sus objetos muy dispersos.
  - Si se imponen previamente k clusters puede dar lugar a grupos artificiales o bien a juntar grupos distintos.

3. Métodos de partición

Algoritmo de K-Medias.

- Si se quiere comprobar la **estabilidad de los grupos**, es conveniente volver a correr el algoritmo con otros clusters iniciales.
  - Se recomienda repetir el proceso entre 20 y 50 veces y seleccionar la configuración con mejor suma de cuadrados (menor varianza interna). *Lo veremos a continuación…*
  - Este tipo de métodos presenta un problema de robustez cuando hay outliers, por lo que una revisión previa de los mismos y el conocimiento de su existencia es muy importante para entender los resultados.

3. Métodos de partición

Algoritmo de K-Medias.

- Una vez **considerados los clusters finales** es conveniente interpretarlos.
  - Puede ayudar ordenar los individuos de forma que los del primer cluster aparezcan al principio y los del último al final, y comprobar las similitudes entre individuos del mismo cluster.
  - También puede ayudar calcular el valor medio de cada variable en cada grupo definido.
  - …
  - Cualquier análisis que ayude a interpretar la composición de los grupos obtenidos.

3. Métodos de partición

Algoritmo de K-Medias.

***Sumas de Cuadrados Dentro de cada Grupo* (SCDG)**

- Dentro de cada grupo podemos calcular las Sumas de Cuadrados, sumando, por ejemplo, las distancias al cuadrado de cada punto al centroide del grupo (otra alternativa podría ser sumar las distancias al cuadrado entre todos los puntos del cluster) . Así, para cada cluster formado disponemos de su **Suma de Cuadrados**.
- Si sumamos las Sumas de Cuadrados de cada cluster obtenemos la **SCDG**. Cuanto menor sea este valor mejor agrupados han resultado los datos. Por tanto, queremos tratar de minimizar la SCDG. Repetiremos el proceso varias veces y nos quedaremos con la configuración con menor SCDG.

3. Métodos de partición

Algoritmo de K-Medias.

Elección del número de grupos:

- Objetivo de los grupos: que los **centroides** estén lo **más separados** entre sí como sea posible y que las **observaciones** dentro de cada cluster estén **muy próximas** al centroide.

- Para valorar con qué numero de clusters quedarnos, podemos utilizar la ***Suma de Cuadrados Dentro de los Grupos*** (**SCDG**), ya que la forma de obtener grupos homogéneos es **minimizar la SCDG**.

- Podemos aplicar el algoritmo de k-medias con k=2,3,4,5,… y comprobar el valor de SCDG. Cuando el paso de k a k+1 reduzca la SCDG en una cantidad despreciable nos quedaremos con k clusters (y no con k+1).

3. Métodos de partición

Algoritmo de K-Medias.

Elección del número de grupos:

Criterio de Hartigan:

Supongamos que hemos planteado obtener G grupos y me pregunto si introduzco uno más (es decir, G+1). Llamemos SCDG(G) a la suma de cuadrados con G grupos y SCDG(G+1) a la suma de cuadrados con G+1 grupos.

Calculamos F: $$F = \frac{SCDG(G) - SCDG(G+1)}{SCDG(G+1)/(n-G-1)}$$

Si F>10 escogeremos una partición con G+1 grupos (si no lo es nos quedaremos con G grupos)

Existen más métodos para seleccionar el mejor número de grupos *(Exploraremos la función* ***NbClust*** *de la librería* ***NbClust*** *al final de la sesión)*

3. Métodos de partición

Diferentes algoritmos para realizar el proceso (k-medias):

- **Lloyd (1957)**
- **Forgy (1965)**
- **MacQueen (1967)**
- **Hartigan and Wong (1979)**

Abordan el proceso de búsqueda de los k grupos de individuos de forma diferente. Basados en

- minimizar de la distancia euclídea de cada individuo al centroide de cada *cluster* en cada iteración,
- minimizar las sumas de cuadrados de las posibles asignaciones de individuos a grupos en cada iteración,…

3. Métodos de partición

Algoritmo de K-Medias con R.

Ejemplo

(ver libro *Análisis Multivariante* de **D.Peña** (2001))

Datos **MEDIFIS**: Ocho variables físicas tomadas a 27 estudiantes:

sex [Sexo] (0=mujer, 1=hombre)

est [Estatura] (en cm)

pes [Peso] (en kg)

lpie [Longitud del pie] (en cm)

lbr [Longitud del brazo] (en cm)

aes [Anchura de la espalda] (en cm)

dcr [Diámetro del cráneo] (en cm)

lrt [Longitud entre la rodilla y el tobillo] (en cm)

3. Métodos de partición

Algoritmo de K-Medias con R.

En R una función que realiza el algoritmo de las k-medias se llama `kmeans` (de la librería básica **`stats`**)

Datos **medifis** con, por ejemplo: k=3 clusters

`Km3<-kmeans(medifis,3)`

(Consultar ayuda de R> **`? kmeans`**)

```
kmeans(x, centers, iter.max = 10, nstart = 1,
       algorithm = c("Hartigan-Wong", "Lloyd", "Forgy",
                     "MacQueen"))
```

3. Métodos de partición

Algoritmo de K-Medias con R.

Parámetros importantes de la función `kmeans` :

`x` : Matriz de datos

`centers`: Número de grupos o vector con los centros iniciales de cada cluster (si se proporciona únicamente el número de centros la función selecciona aleatoriamente individuos de la matriz x como centros iniciales)

`iter.max`: Máximo número de iteraciones permitido (para que tenga un criterio de parada aunque no consiga su objetivo de convergencia)

`algorithm`: Algoritmo que será usado para obtener los grupos (“Hartigan-Wong”, “Lloyd”, “Forgy” o “MacQueen”)

`nstart`: Número de veces que se va a repetir el proceso, cada vez con una asignación aleatoria inicial distinta (recomendable un número elevado, de 20 a 50 veces)

3. Métodos de partición

Algoritmo de K-Medias con R.

La función `kmeans` devuelve un objeto con los siguientes componentes :

`cluster`: Vector de valores enteros que indica el cluster al que pertenece cada individuo

`centers`: Matriz que contiene los centroides de cada cluster

`withinss`: La suma de cuadrados dentro de cada cluster

`size`: Tamaño de cada cluster (número de individuos en cada grupo)

`tot.withinss`: Suma de **`withinss`** de todos los clusters (suma de cuadrados dentro de los grupos)

`betweenss`: La suma de cuadrados entre-cluster (entre grupos)

3. Métodos de partición

```
set.seed(1234)
km3 <- kmeans(x = scale(medifis), centers = 3, nstart = 25)
km3
```

```
## K-means clustering with 3 clusters of sizes 11, 8, 8
##
## Cluster means:
##          sex        est          pes        pie        lbr        aes
## 1  1.0971343  0.9847854  0.89637767  1.0222679  0.9518872  0.8959398
## 2 -0.6308522 -0.3705330 -0.05967303 -0.4081870 -0.3077163 -0.1963008
## 3 -0.8777075 -0.9835470 -1.17284627 -0.9974313 -1.0011286 -1.0356164
##          dcr        lrt
## 1  0.5603545  0.8491372
## 2  0.1407742 -0.1877489
## 3 -0.9112617 -0.9798148
##
## Clustering vector:
##  [1] 3 2 2 2 3 2 3 1 1 3 2 1 3 1 1 1 1 2 3 3 1 1 3 1 2 1 2
##
## Within cluster sum of squares by cluster:
## [1] 30.97231 18.65857 13.00783
##  (between_SS / total_SS =  69.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

3. Métodos de partición

Algoritmo de K-Medias con R.

- Podemos observar el número de individuos por cluster:

```
table(km3$cluster)
```

```
##
##  1  2  3
## 11  8  8
```

- Podríamos identificar los individuos de cada grupo

```
which(km3$cluster == 1)
```

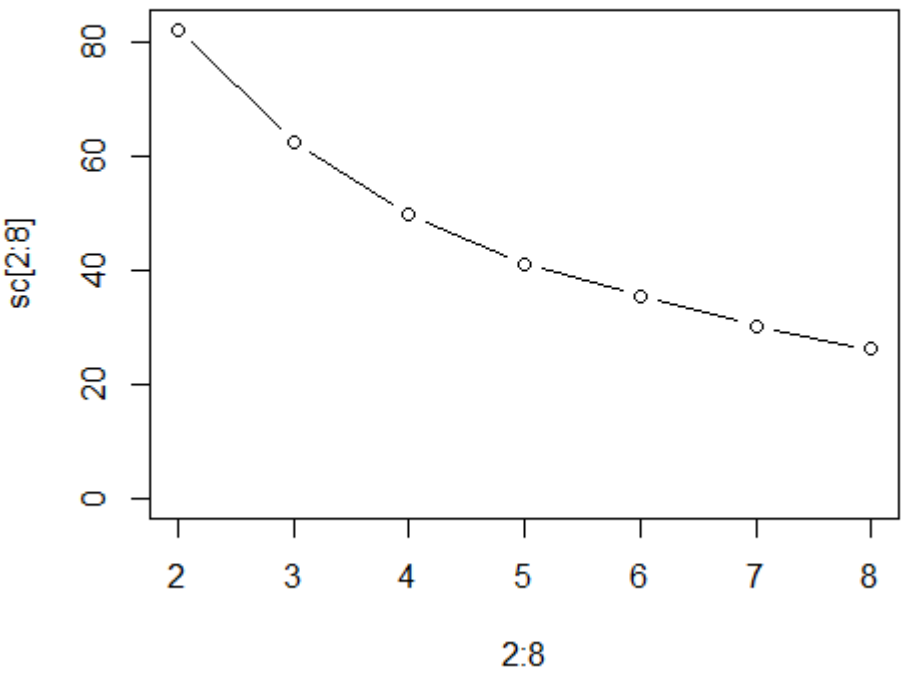
```
##  [1]  8  9 12 14 15 16 17 21 22 24 26
```

(si tuviéramos un identificador, como por ejemplo el nombre de un país, de cada individuo podríamos mostrarlo)

3. Métodos de partición

Número de clusters a elegir

```
# comprobación de la SCDG para cada valor de k
sc <- c()
for (k in 2:8) {
    kk <- kmeans(x = scale(medifis), centers = k, nstart = 25)
    sc[k] <- kk$tot.withinss
}
plot(2:8, sc[2:8], type = "b")
```

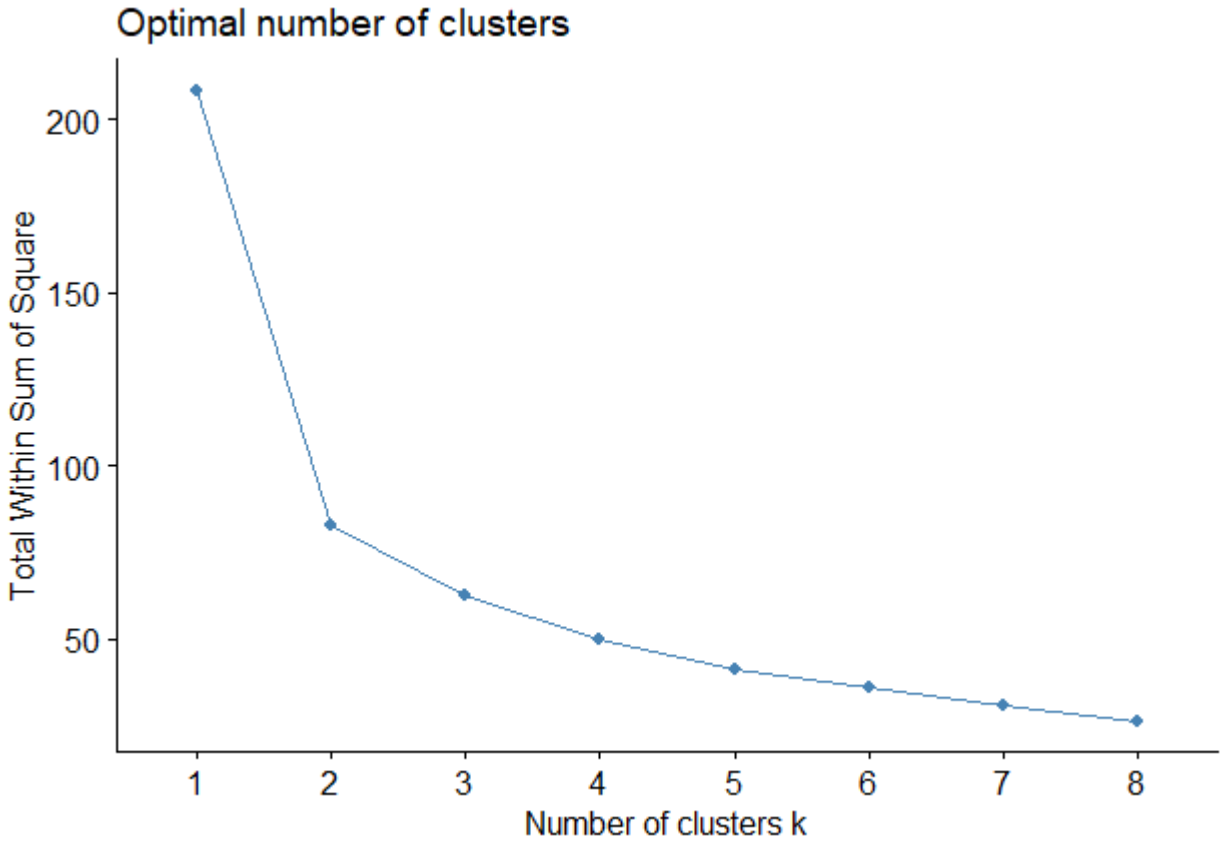


- En este ejemplo no está claro en qué valor de k deja de disminuir la SCDG, tomaremos como valor k=5 clusters, pues la SCDG ya no disminuye sustancialmente, y para 26 individuos no parece sensato hacer muchos más grupos.

3. Métodos de partición

Número de clusters a elegir (una función equivalente)

```
library(factoextra)
fviz_nbclust(x = scale(medifis[, 1:8]), FUNcluster = kmeans, method = "wss",
    k.max = 8, diss = get_dist(scale(medifis[, 1:8]), method = "euclidean"),
    nstart = 50)
```



- Reproduce el análisis anterior directamente (incluye la posibilidad de k=1)

3. Métodos de partición

Análisis de los resultados

Sabiendo a qué cluster se ha asignado cada individuo, podríamos hacer un análisis exploratorio para definir las diferencias entre las agrupaciones realizadas en cuanto a las variables del banco de datos.

Podríamos crear una nueva columna en el banco de datos que indicara a cuál de los 5 grupos pertenece cada individuo, y realizar un resumen según esa agrupación de las variables del banco de datos.

```
km5 <- kmeans(x = scale(medifis), centers = 5, nstart = 25)
medifis$cluster <- km5$cluster
head(medifis)
```

```
##   sex est pes pie lbr  aes dcr lrt cluster
## 1   0 159  49  36  68 42.0  57  40       5
## 2   1 164  62  39  73 44.0  55  44       2
## 3   0 172  65  38  75 48.0  58  44       1
## 4   0 167  52  37  73 41.5  58  44       1
## 5   0 164  51  36  71 44.5  54  40       5
## 6   0 161  67  38  71 44.0  56  42       1
```

3. Métodos de partición

Análisis de los resultados

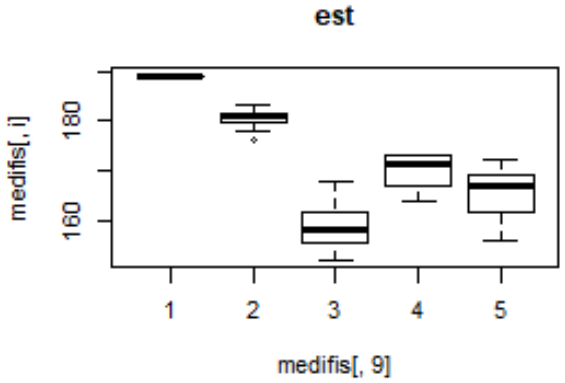
Respecto a la variable Sexo:

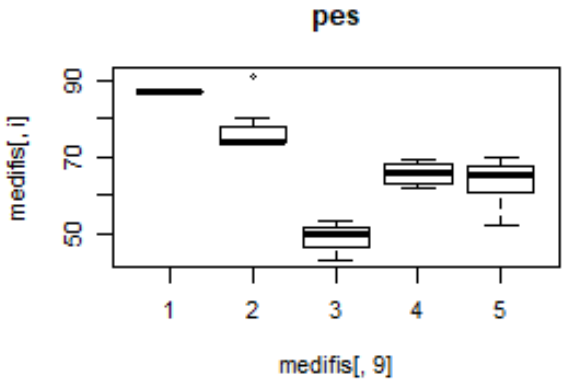
```
table(medifis$sex, medifis$cluster)
```

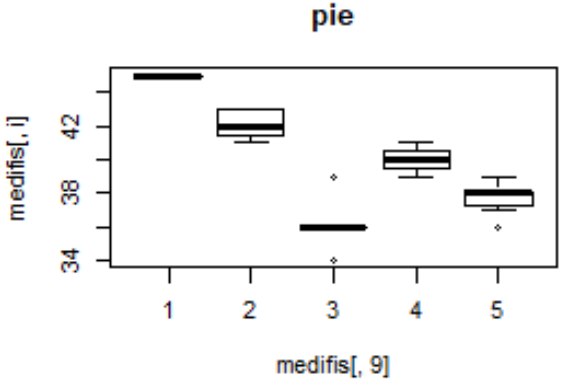
```
  1 2 3 4 5
0 7 0 0 0 8
1 0 4 7 1 0
```

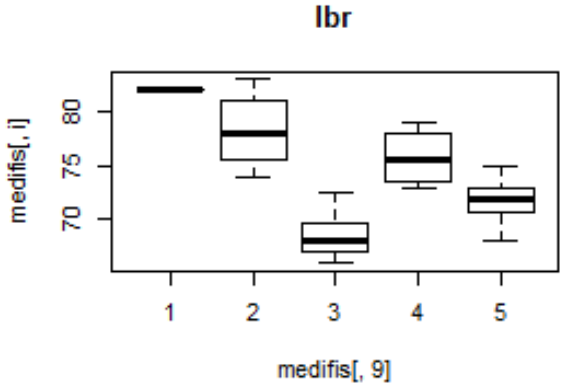
Respecto a las variables cuantitativas *(boxplot por variable y/o dispersión por variable*):

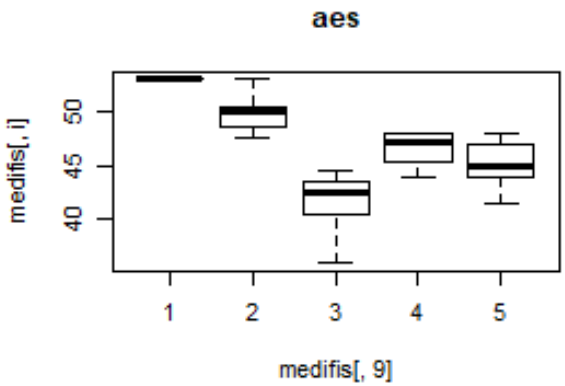
```
par(mfrow = c(3, 3))
for (i in 2:8) {
    boxplot(medifis[, i] ~ medifis[, 9], main = colnames(medifis)[i])
}
par(mfrow = c(3, 3))
for (i in 2:8) {
    plot(medifis[, 9], medifis[, i], col = medifis[, 9], pch = medifis[,
        1] + 1, xlab = "Grupo", ylab = "", main = colnames(medifis)[i])
}
```

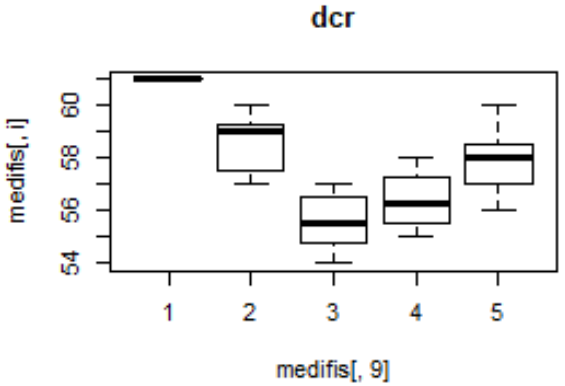

est
medifis[, i]
160
180
1
2
3
4
5
medifis[, 9]

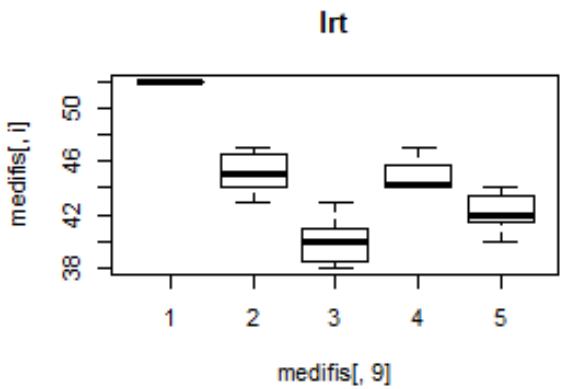

pes
medifis[, i]
50
70
90
1
2
3
4
5
medifis[, 9]

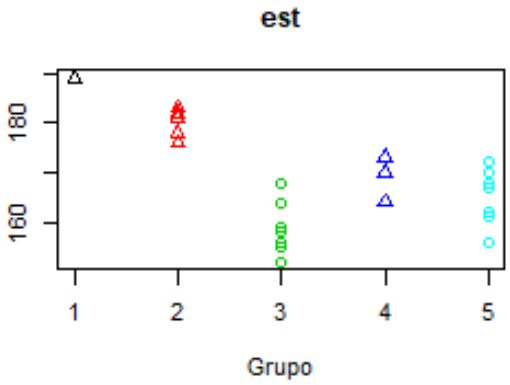

pie
medifis[, i]
34
38
42
1
2
3
4
5
medifis[, 9]

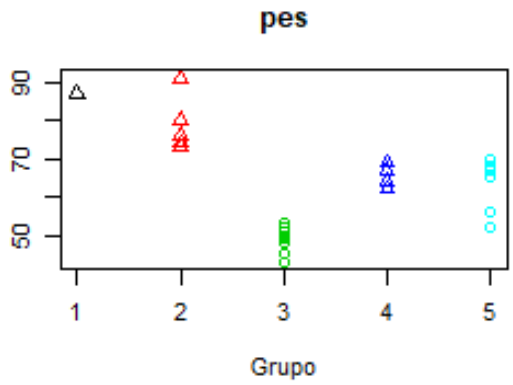

lbr
medifis[, i]
70
75
80
1
2
3
4
5
medifis[, 9]

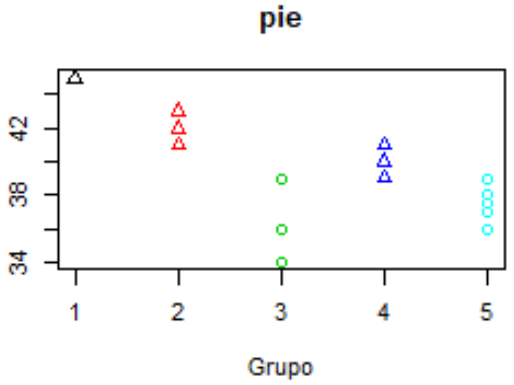

aes
medifis[, i]
40
45
50
1
2
3
4
5
medifis[, 9]

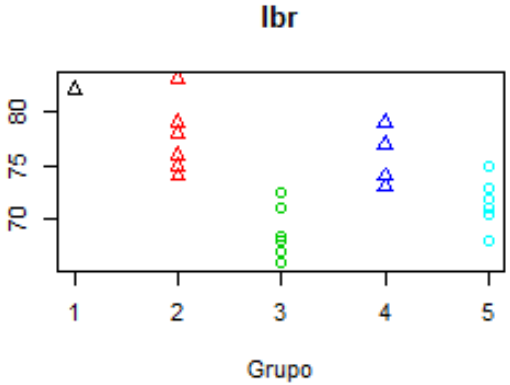

dcr
medifis[, i]
54
56
58
60
1
2
3
4
5
medifis[, 9]

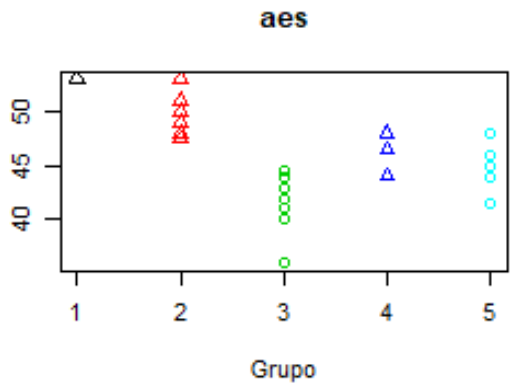

lrt
medifis[, i]
38
42
46
50
1
2
3
4
5
medifis[, 9]

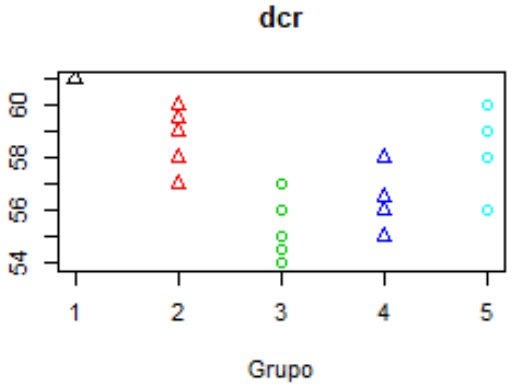

est
160
180
1
2
3
4
5
Grupo

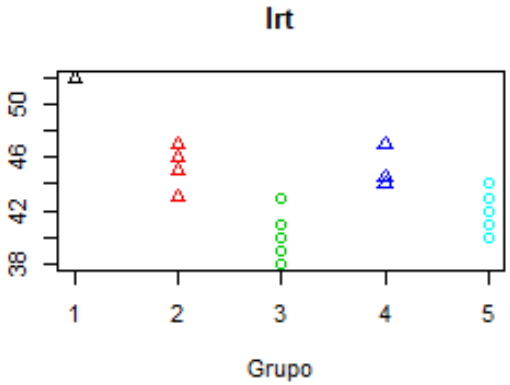

pes
50
70
90
1
2
3
4
5
Grupo


pie
34
38
42
1
2
3
4
5
Grupo

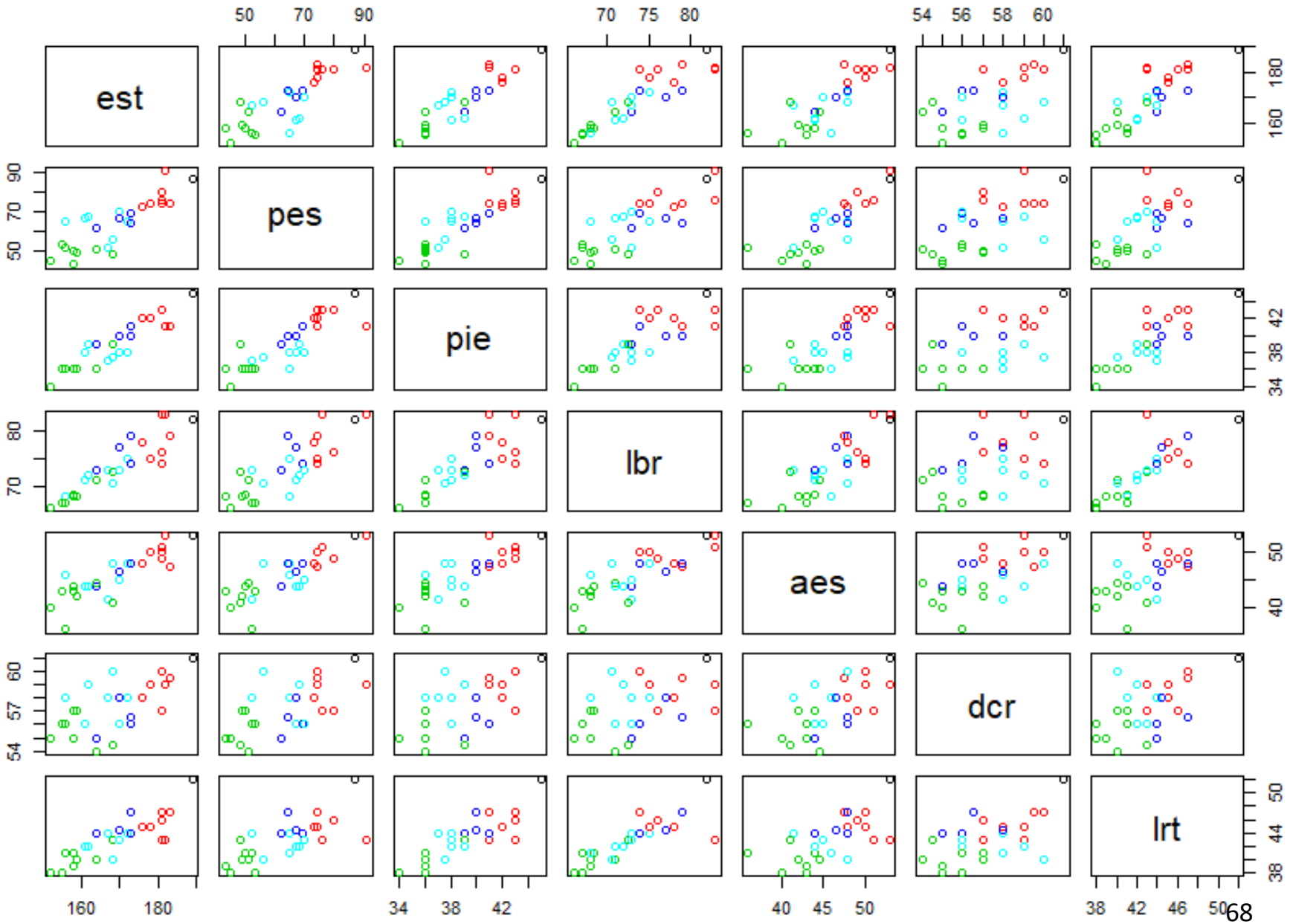

lbr
70
75
80
1
2
3
4
5
Grupo


aes
40
45
50
1
2
3
4
5
Grupo


dcr
54
56
58
60
1
2
3
4
5
Grupo


lrt
38
42
46
50
1
2
3
4
5
Grupo

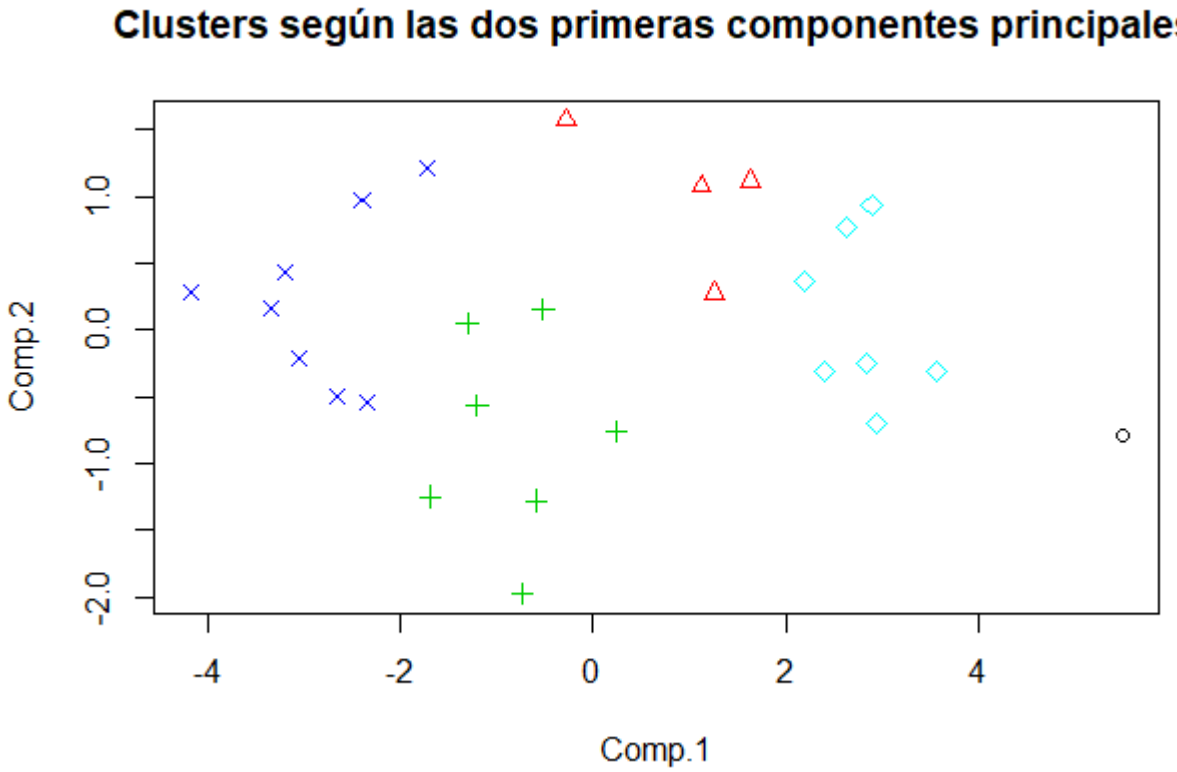
```
pairs(medifis[, -c(1, 9)], col = medifis[, 9])
```



3. Métodos de partición

Para una visualización completa, podemos representar los clusters según las dos primeras CP (en 2 dim).

```
km5 <- kmeans(x = scale(medifis[, 1:8]), centers = 5, nstart = 25)
acp1 <- princomp(medifis[, 1:8], cor = TRUE)
plot(acp1$scores[, 1:2], pch = km5$cluster, col = km5$cluster)
title("Clusters según las dos primeras componentes principales")
```

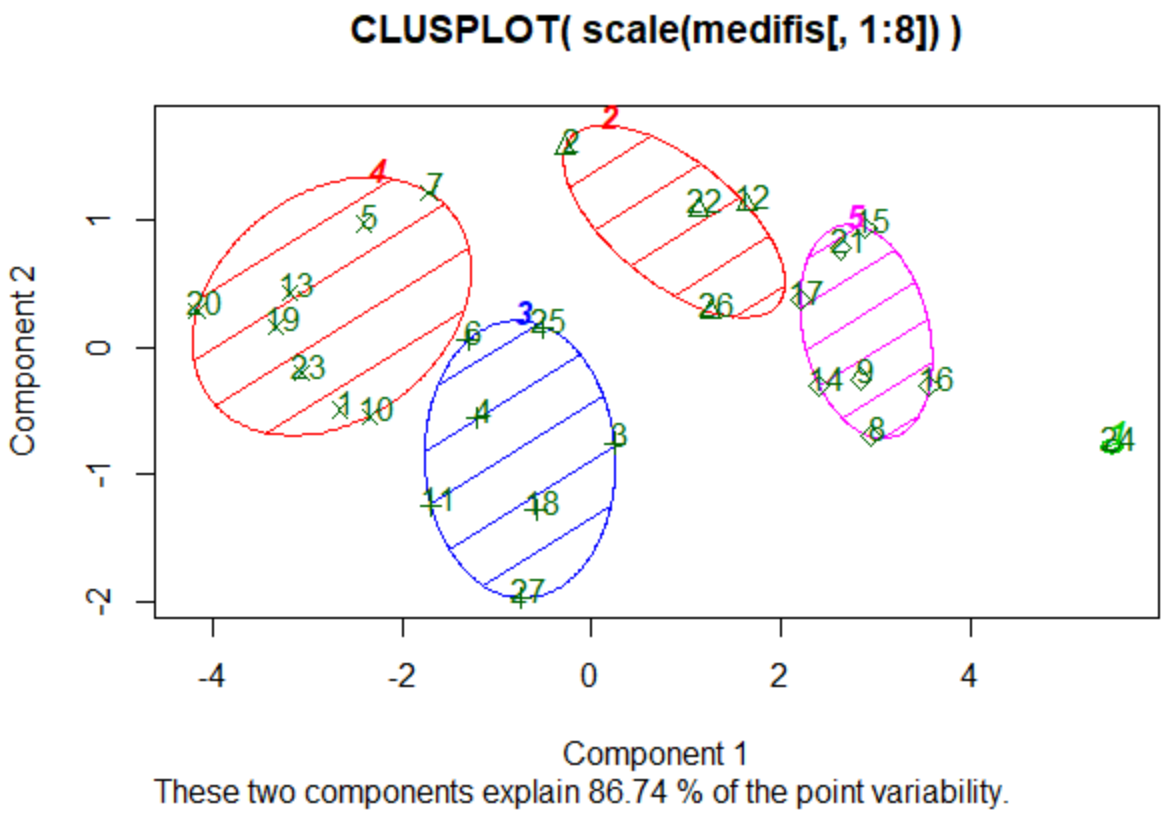


Podríamos representar identificadores de individuos,…

3. Métodos de partición

Otras funciones para visualizar resultados: Realiza un ACP sobre las variables estandarizadas y representa los grupos sobre las 2 primeras Comp. Principales

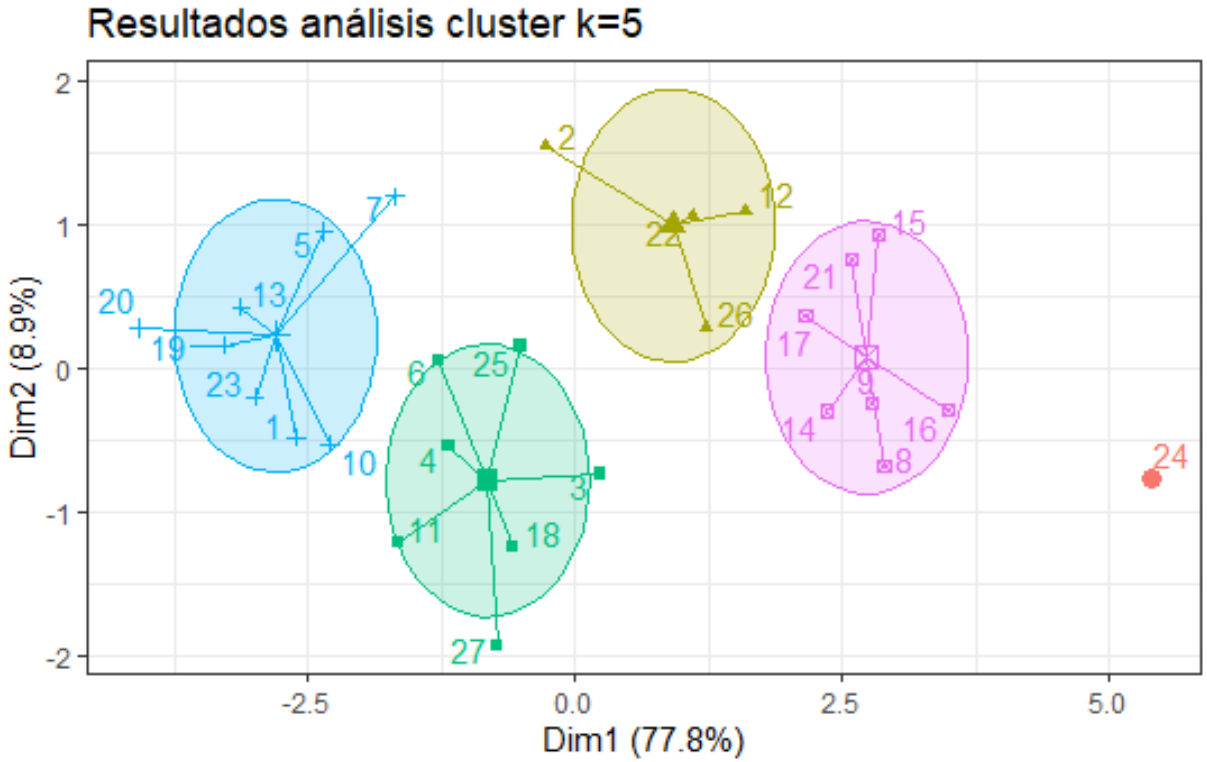
```
library(cluster)
clusplot(scale(medifis[, 1:8]), km5$cluster, color = TRUE, shade = TRUE,
    labels = 2, lines = 0)
```



3. Métodos de partición

Otra utilidad del paquete `factoextra` permite visualizar los clusters obtenidos. Si hay más de dos variables realiza un PCA directamente

```
fviz_cluster(object = km5, data = scale(medifis[,1:8]),
             show.clust.cent = TRUE,ellipse.type = "euclid",
             star.plot = TRUE, repel = TRUE) +
  labs(title = "Resultados análisis cluster k=5") +
  theme_bw() +
  theme(legend.position = "none")
```



3. Métodos de partición

Para finalizar esta sección considerar:

Antes de aceptar los resultados de un análisis de conglomerados mediante el algoritmo de k-medias conviene probar distintos puntos de partida y distintos algoritmos.

Existen un métodos similares al k-means, denominados k-medoids, en el que cada cluster queda representado por una observación en lugar de por su centroide. Esa observación que representa cada cluster es aquella cuya distancia media al resto de elementos del cluster es lo más pequeña posible (similar a la idea de media y mediana). Estos métodos, aunque más costosos computacionalmente, pueden funcionar mejor en presencia de outliers. Uno de estos métodos es el método **PAM** (Partitioning Around Miedoids), disponible en la función `pam` de la librería `cluster` y en la función `fviz_nbclust` de `factoextra`.

4. Métodos jerárquicos

Los **métodos jerárquicos** son una alternativa a los métodos de partición que no requiere que el usuario especifique a priori el número de clusters.

Estos métodos pueden ser:

- **Aglomerativos**: Se parte de que cada observación es un cluster inicial y se van uniendo clusters iterativamente hasta obtener un único cluster. Nos centraremos en esta opción.
- **Divisivos**: Se parte de todas las observaciones contenidas en un único cluster y se va dividiendo iterativamente hasta que cada observación está es un cluster diferente.

4. Métodos jerárquicos

- El **resultado** se suele representar mediante un **dendograma**.

- A partir del resultado (dendograma) se puede obtener la composición de los clusters para k=2,3,…grupos. **Decidir por dónde cortar** supone el problema similar al que se produce en los métodos de partición (pero en este caso a posteriori).

4. Métodos jerárquicos

Fundamentos de los algoritmos jerárquicos (aglomerativos):

- Partimos de una "disimilitud /similitud" entre los *"n"* puntos a clasificar.
- Consideramos **cada punto como un cluster**. Tenemos *"n"* grupos.
- Buscamos los dos puntos más próximos y los agregamos en un grupo. Tenemos *"n-1"* clusters.
- Calculamos la ***"distancia"*** entre el nuevo grupo y los restantes grupos (es necesario indicar cómo se va a hacer este *linkage*, se debe *extender* el concepto de *distancia entre invidiuos a distancia entre grupos*).
- Buscamos otra vez los dos clusters más próximos, que agruparemos y repetiremos el proceso anterior hasta que tengamos un solo grupo.

4. Métodos jerárquicos

Fundamentos de los algoritmos jerárquicos:

Debemos detallar los elementos del siguiente esquema:

Matriz de disimilaridades *ó similaridades*

Método aglomerativo *de clusters*

Presentación**:**

- **Dendograma**
- Carámbanos
- Esquema

4. Métodos jerárquicos

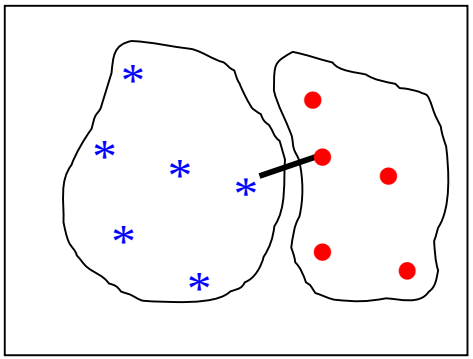
Métodos aglomerativos de cluster.

- M1: **Unión smple** o **Vecino más próximo** (*SINGLE o MINIMUM*)

$$d(C_1, C_2) = \min_{i,j} d(x_i, y_j) \quad x_i \in C_1, y_j \in C_2$$

La distancia entre dos clusters será la distancia **entre los puntos más próximos de ambos clusters**.

M1



4. Métodos jerárquicos

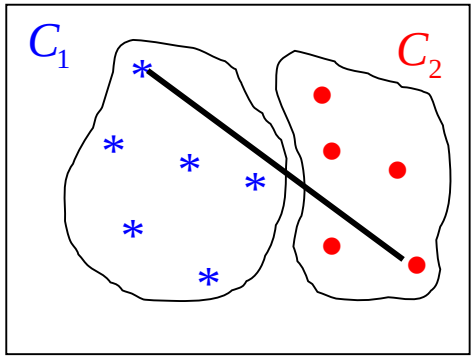
Métodos aglomerativos de cluster.

- M2: **Unión completa** o **Vecino más alejado** (*COMPLETE o MAXIMUM*)

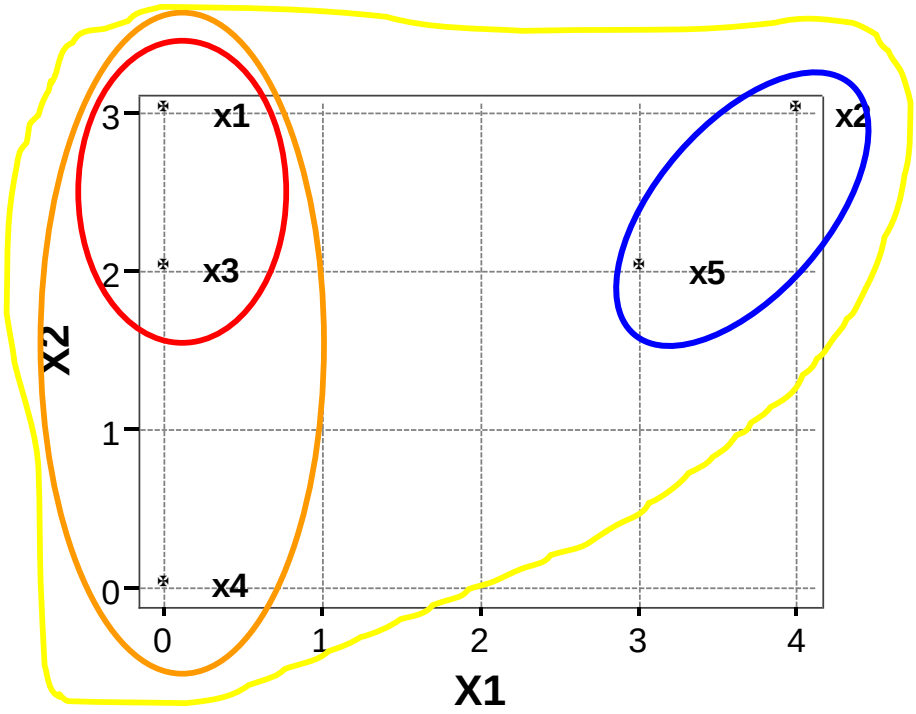
$$d(C_1, C_2) = \max_{i,j} d(x_i, y_j) \quad x_i \in C_1, y_j \in C_2$$

La distancia entre dos clusters será la distancia **entre los puntos más alejados de ambos clusters**.

M2



4. Métodos jerárquicos



	(1)	(2)	(3)	(4)
(2)	16			
(3)	1	17		
(4)	9	25	4	
(5)	10	2	9	13

$(1+3)$

	(1)	(2)	(4)
(2)	16		
(4)	4	25	
(5)	9	2	13

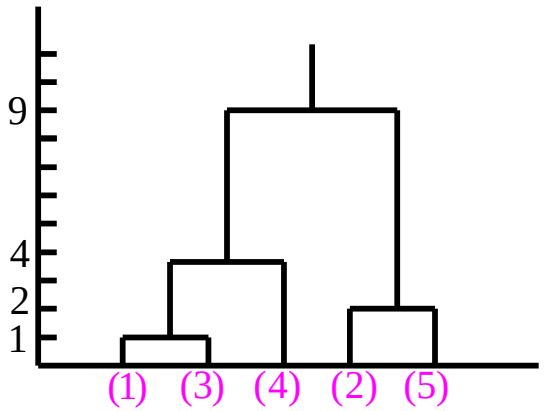
$(1+3)$ $(2+5)$

	(1)	(2)
(2)	9	
(4)	4	13

$(1+3+4)$ $(2+5)$

	(1)
(2)	9

DENDOGRAMA



4. Métodos jerárquicos

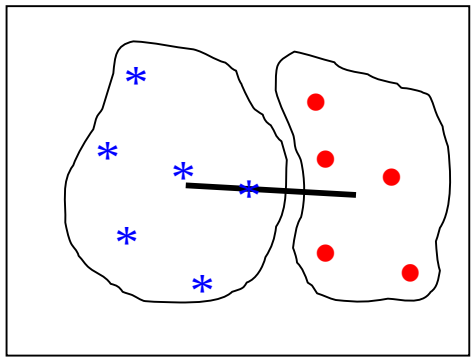
Métodos aglomerativos de cluster.

- M3: **Método del centroide** (*CENTROID*)

$$d(C_1, C_2) = d(\bar{x}, \bar{y})$$

La distancia entre dos clusters será la distancia **entre los centroides de ambos clusters.** El centro del nuevo cluster será $\frac{(n_1\bar{x}+n_2\bar{y})}{(n_1+n_2)}$

M3



4. Métodos jerárquicos

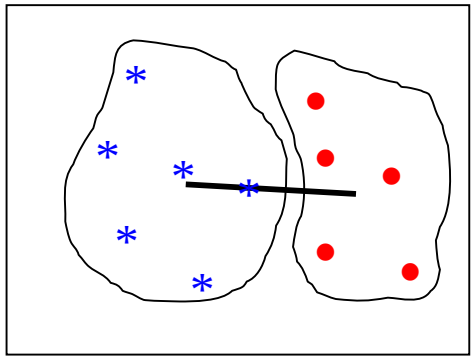
Métodos aglomerativos de cluster.

- M4: **Método de la mediana** (*MEDIAN*)

$$d(C_1, C_2) = d(\bar{x}, \bar{y})$$

La distancia entre dos clusters será la distancia **entre los centroides de ambos clusters.** El centro del nuevo cluster será $\frac{(\bar{x}+\bar{y})}{2}$

M4



4. Métodos jerárquicos

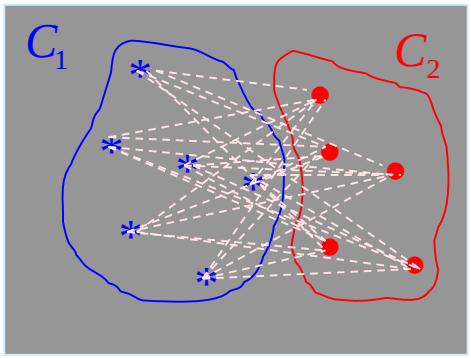
Métodos aglomerativos de cluster.

- M5: **Método de la distancia media entre grupos** (BAVERAGE)

$$d(C_1, C_2) = \frac{\sum_i^{n_1} \sum_{j=1}^{n_2} d(x_i, y_j)}{n_1 \cdot n_2}; x_i \in C_1, y_j \in C_2$$

Se calcula el promedio de las distancias entre los elementos de un cluster y los elementos del otro.

M5



4. Métodos jerárquicos

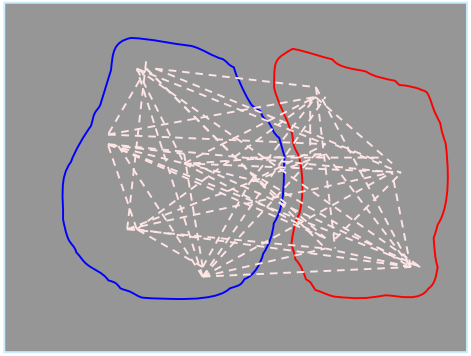
Métodos aglomerativos de cluster.

- M6: **Método de la distancia media dentro de grupos** (WAVERAGE)

$$d(C_1, C_2) = \frac{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} d(x_i, y_j) + \sum_{i=1}^{n_1}\sum_{j=1}^{n_1} d(x_i, x_j) + \sum_{i=1}^{n_2}\sum_{j=1}^{n_2} d(y_i, y_j)}{n_1 \cdot n_2}; x_i \in C_1, y_j \in C_2$$

Se calcula el promedio de las distancias entre todos los elementos del cluster resultante si se unen C_1 y C_2

M6



4. Métodos jerárquicos

Métodos aglomerativos de cluster.

- M7: **Método del incremento de la suma de cuadrados** (WARD)

Se calcula la suma de cuadrados dentro de cada clusters C_1 y C_2, SS_1 y SS_2. A continuación se calcula la suma de cuadrados del cluster que se obtendría de la unión de ambos clusters, SS_{12}. A continuación se calcula el incremento de la suma de cuadrados que resulta ser:
$\mathrm{I}(C_1, C_2) = SS_{12} - SS_1 - SS_2$.

Se unirán aquellos clusters cuya unión produzca menos incremento.

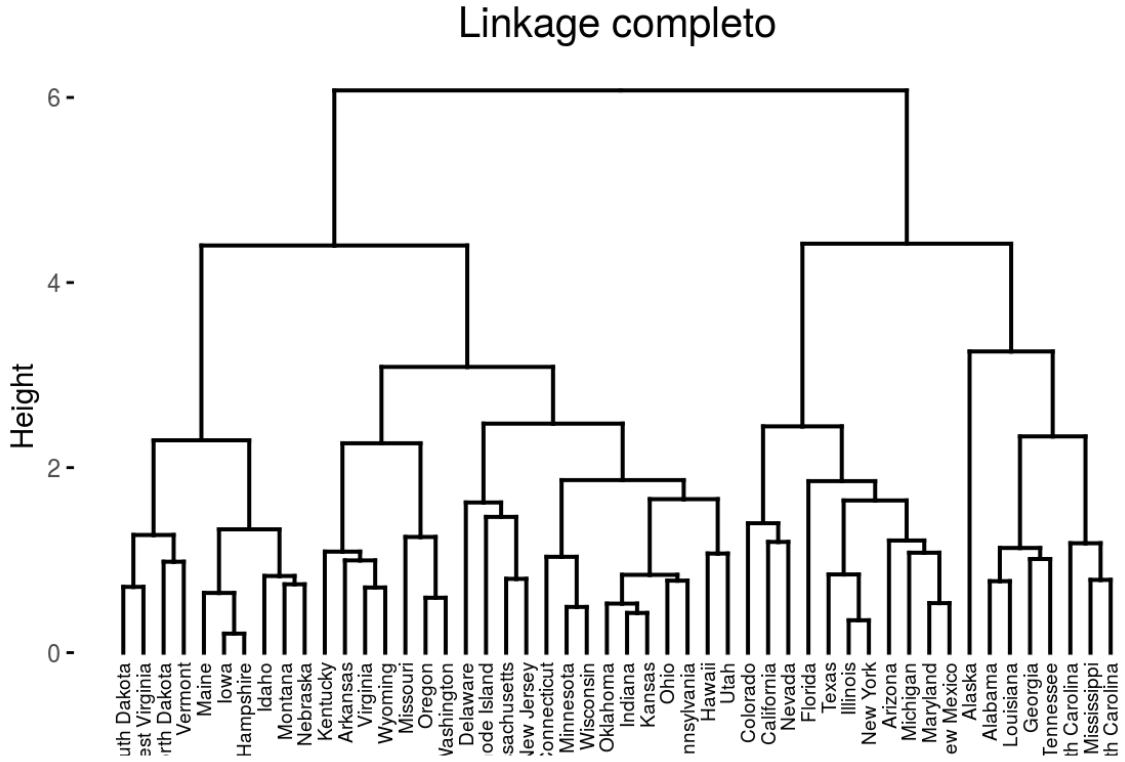
Para el dendograma se utiliza $\mathrm{d}(C_1, C_2) = SS_{12}$

4. Métodos jerárquicos

- Los métodos de unión *COMPLETE, WARDS y BAVERAGE* son los más utilizados debido a que genera dendogramas más equilibrados o compensados.
- En genómica, por ejemplo, se suelen utilizar los métodos *CENTROID*.
- **Los resultados pueden variar en función de la distancia empleada y del tipo de linkage**, por este motivo es necesario indicar, junto con los resultados, los criterios utilizados.
- La **selección del número óptimo** puede valorarse de forma visual, tratando de identificar las ramas principales en base a la altura a la que ocurren las uniones.

4. Métodos jerárquicos

- La **selección del número óptimo** de clusters puede explorarse de forma visual a partir del dendograma, tratando de identificar las ramas principales en base a la altura a la que ocurren las uniones. *Por ejemplo, en este gráfico sería razonable seleccionar 4.*



https://rpubs.com/Joaquin_AR/310338

4. Métodos jerárquicos

Algoritmo jerárquico con R

En primer lugar necesitamos una matriz de distancias:

Función `dist` *(o daisy)*

```
dist(x, method = "euclidean", diag = FALSE,
     upper = FALSE,…)
```

Algunos parámetros de la función `dist`:

`x` : Matriz de datos

`method`: Distancia a elegir entre "**`euclidean`**", "**`maximum`**", "**`manhattan`**", "**`canberra`**","**`binary`**" o "**`minkowski`**"

`diag`: Indicamos TRUE si queremos que muestre la diagonal con 0's

`upper`: Indicamos TRUE si queremos que muestre la triangular superior

…

4. Métodos jerárquicos

Algoritmo jerárquico con R

Para aplicar el método de conglomerados jerárquico:

Función `hclust`

```
hclust(d, method = "complete", members=NULL,…)
```

Algunos parámetros de la función `hclust`:

`d` : Matriz de distancias (o medidas de similitud)

`labels`: Etiquetas que identificarán a los individuos. Por defecto tomará el número de cada fila o los nombres de cada fila.

…

4. Métodos jerárquicos

Algoritmo jerárquico con R

Algunos parámetros de la función `hclust`:

`d` : Matriz de distancias (o medidas de similitud)

`method`: Método aglomerativo a elegir entre

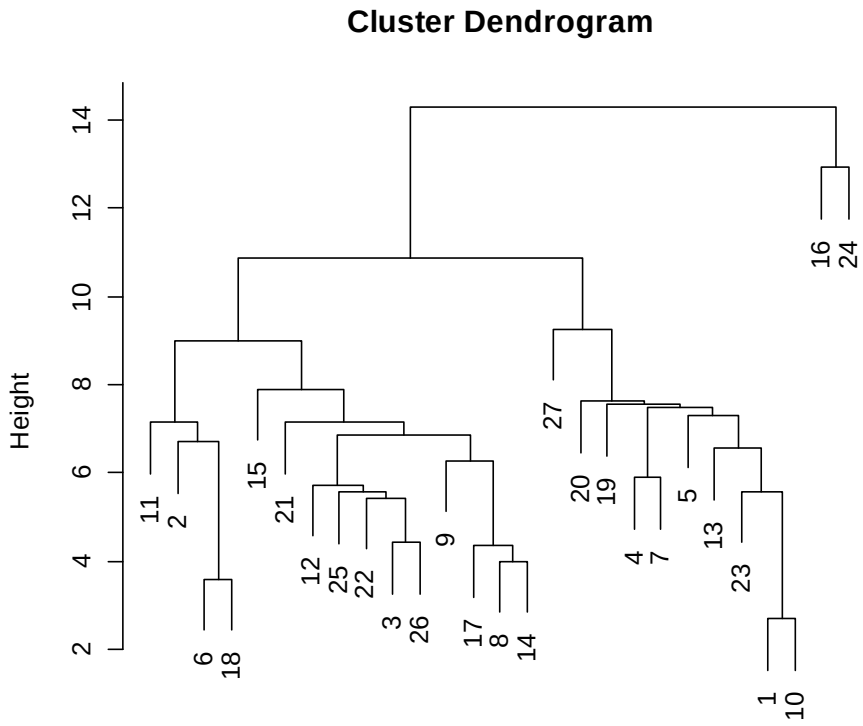
- **`"ward.D"`** * **M7**
- **`"ward.D2"`** * *Variación/mejora de M7 (Murtagh & Legendre 2014)*
- **`"single"`** **M1**
- **`"complete"`** **M2**
- **`"average"`** **M5**
- **`"mcquitty"`** **M6**
- **`"median"`** **M4**
- **`"centroid"`** **M3**

* *En versiones antiguas de R (anteriores a la v.3.0.3 aparece únicamente el método "ward", **M7***

4. Métodos jerárquicos

Ejemplo (Medifis):

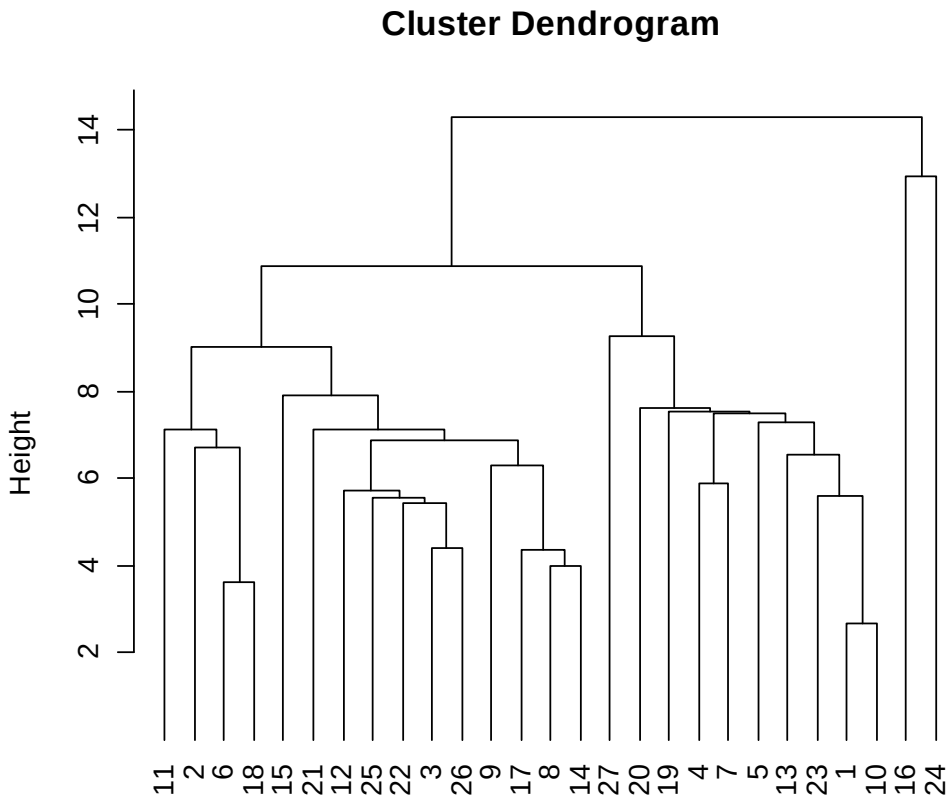
```
> d<-dist(medifis,method="euclidean")
> hc1<-hclust(d,method="single")
> plot(hc1)
```



4. Métodos jerárquicos

Ejemplo (Medifis): Consideramos en primer lugar las variables sin estand.

```
> plot(hc1,hang=-1)
```



4. Métodos jerárquicos

Algoritmo jerárquico con R

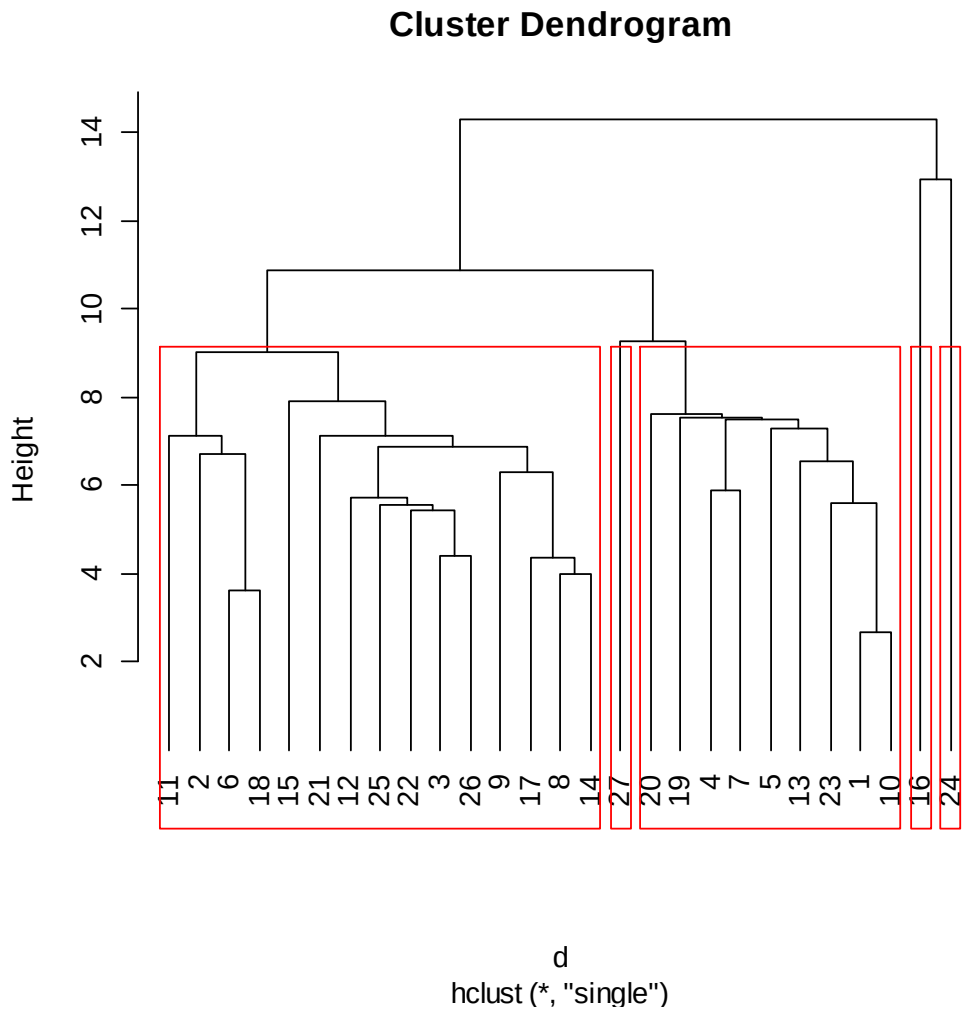
- Los **métodos jerárquicos proporcionan todos los posibles niveles** de clasificación.
- Normalmente **estamos interesados** en **agrupar los individuos** en un número (aprox.) determinado de grupos.
- Para visualizar los grupos a determinado nivel de clasificación podemos utilizar las **funciones `rect.hclust` y `cutree`.**

```
> grupos5<-cutree(hc1,k=5)
> grupos5
 [1] 1 2 2 3 3 2 3 4 4 1 2 2 1 4 4 5 4 2 1 1 4 2 1
5 2 2 3
> table(grupos5)
grupos5
 1  2  3  4  5
 9 15  1  1  1
```

4. Métodos jerárquicos

Ejemplo (Medifis)

```
> plot(hc1,hang=-1)
> rect.hclust(hc1,
k=5,border="red")
```



4. Métodos jerárquicos

Validación de los conglomerados obtenidos.

- El **análisis cluster jerárquico** parte de una **matriz de distancias** (o de medidas de similaridad).
- El **resultado** de estos métodos es un **dendograma** que representa todos los niveles de clasificación.
- **En un dendograma**, la **distancia estimada entre dos puntos** es el **nivel al cuál esos dos puntos se unen**. A esta distancia se le llama **distancia cofenética**.
- Un **buen método reproduce** en las **distancias cofenéticas** la **relación** de los puntos en la **matriz de distancias original**.
- Una forma de **validar los resultados** obtenidos es obtener la **correlación cofenética** : **correlación** entre las **distancias reales** (de partida) y las **distancias cofenéticas** (obtenidas a partir del dendograma)

4. Métodos jerárquicos

Correlación cofenética

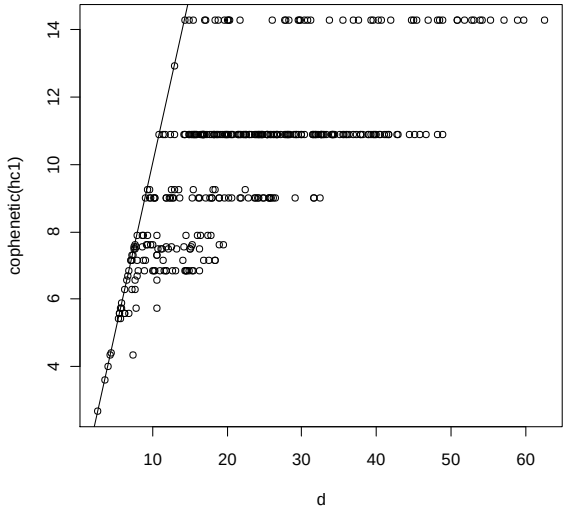
Hay diferentes criterios respecto a lo que se considera un buen valor para la correlación cofenética. Valores superiores a 0.75 suelen considerarse como buenos, aunque cuanto más altos mejor se considera la clasificación jerárquica. Una correlación cofenética de 0.6 a 0.7 indica una clasificación pobre o cuestionable.

Correlación cofenética en R

En **R** se pueden obtener las **distancias cofenéticas** a partir de un **dendograma** con la función **`cophenetic`**.

Ejemplo (Medifis)

```
> cor(d,cophenetic(hc1))
[1] 0.7151927
> plot(d,cophenetic(hc1))
> abline(0,1)
```



Resumen final (I)

- El **objetivo** del ***Análisis Cluster*** es **clasificar individuos en grupos homogéneos** (en función de la información [variables] disponible)

- Esta técnica tiene **carácter exploratorio**

- Consiste en **asignar individuos a grupos** por "*algún criterio de homogeneidad*".

Resumen final (II)

- Consideraciones importantes **previas al análisis**:
  - Debemos **conocer bien** (escala, correlaciones,…) y **seleccionar** las **variables** que describen a los individuos (análisis exploratorio previo)
  - Se debe definir una **medida de similitud/disimilitud** para ir clasificando a los individuos en unos grupos u otros.
    - ✓ Basados en la **distancia** (considerando que los individuos son puntos [vectores] en el espacio k-dimensional que definen las variables.
    - ✓ Basados en **coeficientes de correlación.**
    - ✓ Basados en tablas de variables que definen **posesión o no de una serie de atributos**.

Resumen final (III)

- Podemos abordar este problema desde **dos perspectivas**:
  - **Definir un número de grupos** y agregar en ese número de grupos a los individuos
  - **Partir de tantos grupos como individuos e ir agrupando** los más similares (seleccionar posteriormente el número de clusters)

- Las **distancias** deben calcularse no sólo entre los individuos inicialmente, sino también **entre grupos** o **entre individuos y grupos**.

Resumen final (IV)

- Existen otros métodos, como se ha comentado al inicio de la sesión, que no son objeto de esta asignatura introductoria a la Minería de datos: métodos que asignan a cada elemento una probabilidad de pertenencia a un cluster (no finalizan con cada elemento asignado a un único cluster, métodos basados en modelos,…)

- Los métodos de clustering siempre ofrecen una solución de agrupación de las observaciones en diferente número de clusters. Sin embargo, es posible que en la realidad esas agrupaciones no existan.

- También existen estadísticos que ayudan a elegir el número óptimo de clusters en nuestro banco de datos (*lo veremos a continuación*).

Resumen final (IV)

Análisis de los resultados

- Es recomendable **aplicar diferentes algoritmos** y **comparar sus resultados** para obtener una buena clasificación de los individuos.

- Una vez **obtenida una clasificación aceptable**, se debe realizar un **análisis de los resultados** (tal y como hemos comentado tras la aplicación del algoritmo de k-medias)

- Para llevar a cabo este análisis se pueden utilizar todas las técnicas descriptivas e inferenciales disponibles.

- Los **métodos de conglomerados** revisados en esta sesión, **pueden ser utilizado**s para **agrupar variables** de la misma forma que han sido presentados para agrupar individuos.