Universidad Nacional Autónoma de México
Facultad de Ingeniería

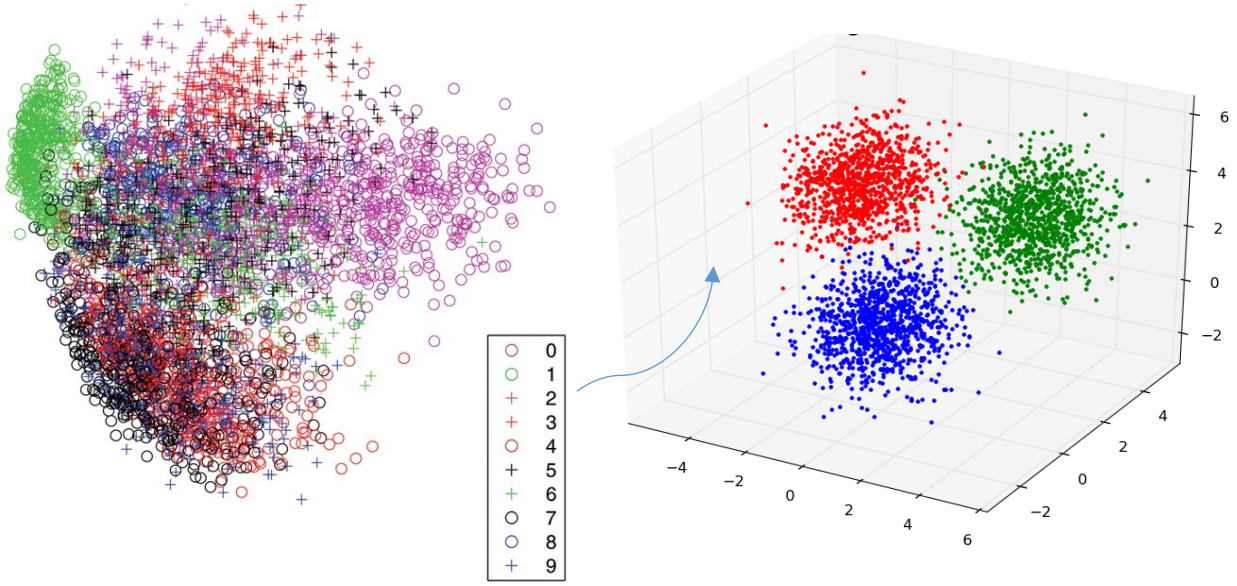
Clustering Particional

Guillermo Molero-Castillo
guillermo.molero@ingenieria.unam.edu

Noviembre, 2020

Contexto

- La **Inteligencia Artificial** aplicada a la definición de cluster consiste en la segmentación y delimitación de grupos de elementos de manera automática, que son unidos por características comunes que éstos comparten.
- El objetivo es dividir una población heterogénea en un número de grupos naturales (regiones o segmentos homogéneos), de acuerdo a la **similitud de los elementos.** Es un tipo de aprendizaje automático no-supervisado.



Los grupos nacen a partir de los datos y se descubre una serie de patrones ocultos en éstos.

Aplicaciones

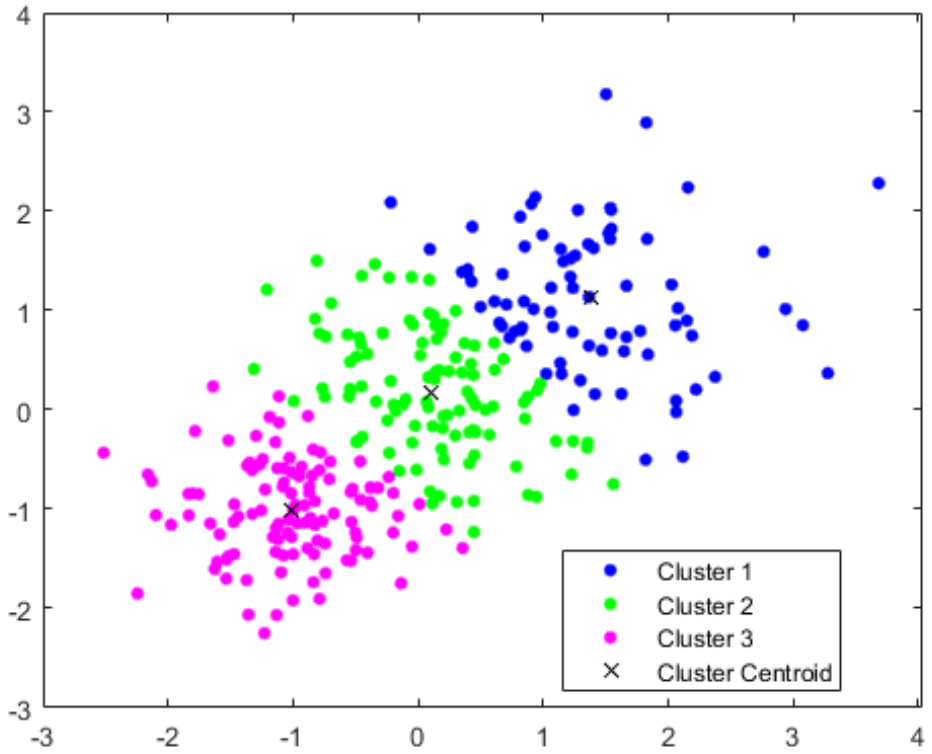
- **Marketing.** Para caracterizar y descubrir segmentos de clientes con fines de marketing.
- **Biología.** Para organizar diferentes especies de plantas y animales.
- **Bibliotecas.** Para agrupar libros a través de temas o autores.
- **Seguro.** Para reconocer a los clientes, sus pólizas e identificar los fraudes.
- **Urbanismo.** Para organizar tipos de viviendas y analizar sus valores en función de su ubicación geográfica.
- **Otras.** Estudios demográficos, regiones afectadas por terremotos, identificación de zonas peligrosas, regionalizaciones climáticas, comunidades de usuarios para los sistemas de recomendación, entre otros.

Clustering Particional

Clustering Particional

El **algoritmo particional**, conocido también como de particiones, organiza los registros dentro de k clústeres. Tiene ventajas en aplicaciones que involucran gran cantidad de datos.

Particional



Clustering Particional

Pasos para formar clústeres:

1. Utilizar un método para medir la similitud de los elementos.
2. Utilizar un método para agrupar a los elementos.
3. Utilizar un método para decidir la cantidad adecuada de grupos.

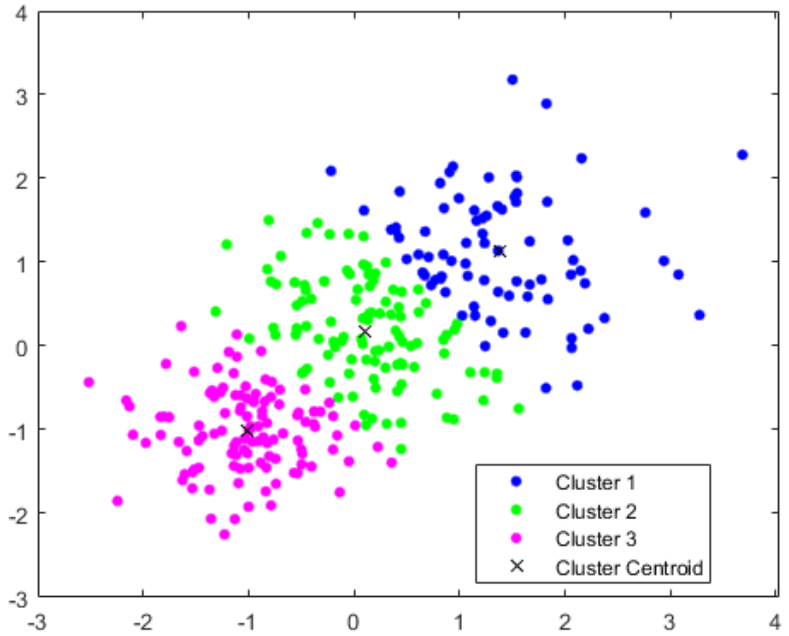
Algoritmo K-means

K-means

- K-means es uno de los algoritmos ampliamente utilizado en el mundo académico y la industria.
- Crea ***k*** clústeres a partir de un conjunto de elementos (objetos), de modo que los miembros de un grupo sean similares.
- Ejemplo: Pacientes por edad, pulso, presión arterial, colesterol, entre otros.

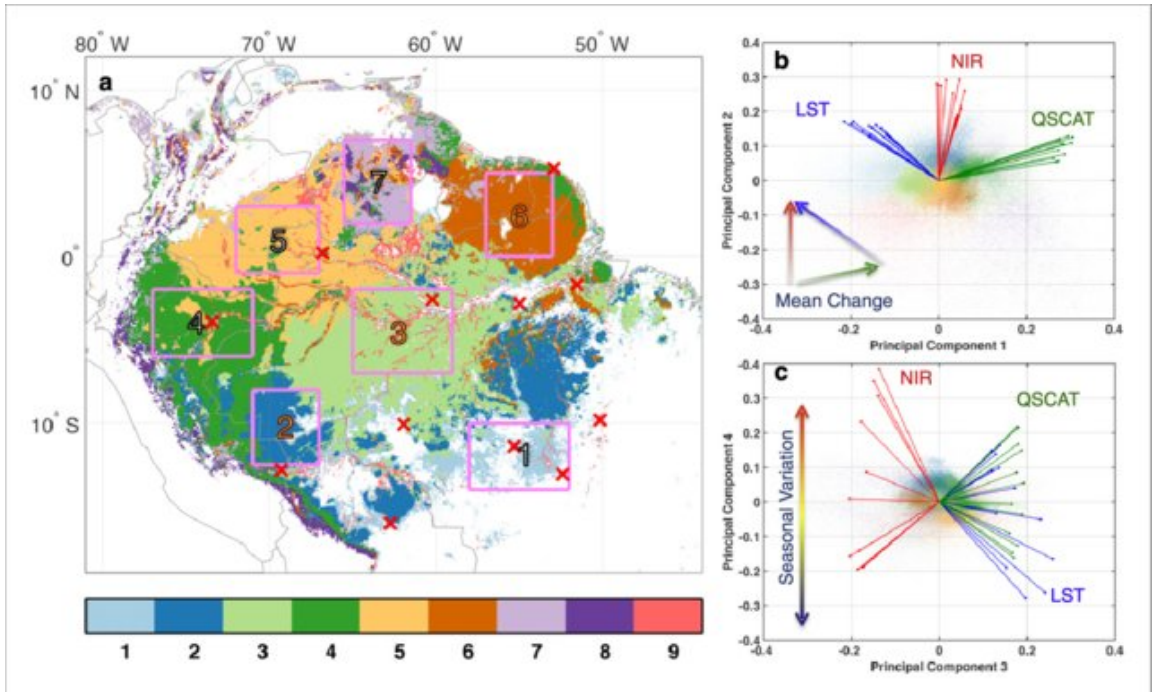
Pulso	una dimensión
Presión arterial	otra dimensión
Colesterol	otra dimensión
…	

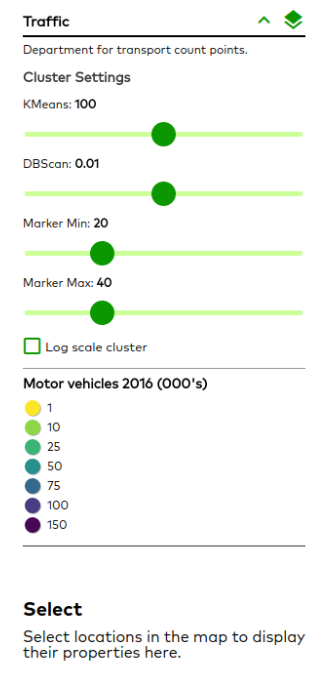
Estas mediciones sobre el paciente representan un vector de datos.

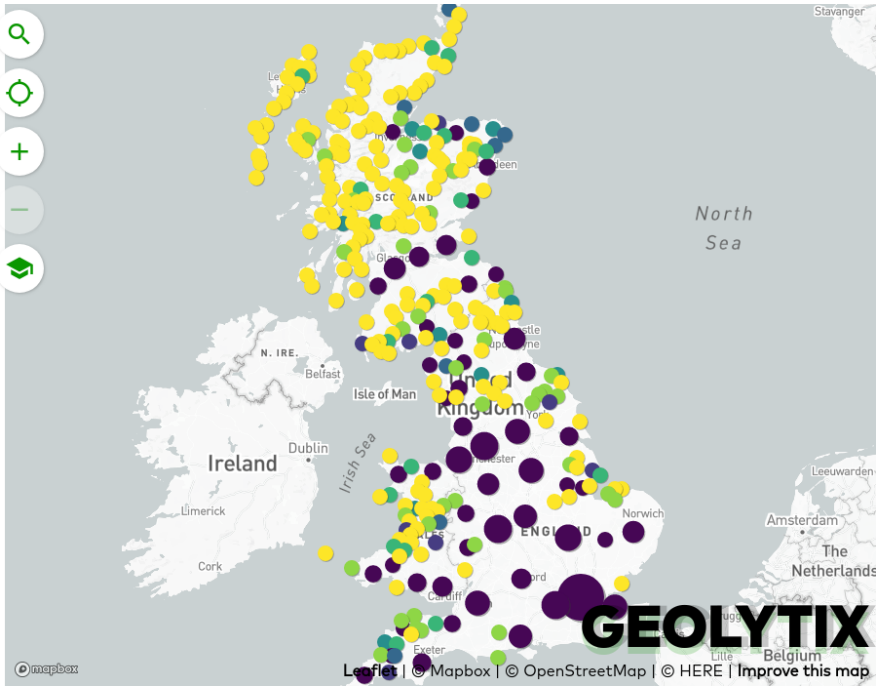


Clustering Particional

El algoritmo *k-means* resuelve **problemas de optimización**, dado que la función es minimizar (optimizar) la suma de las distancias de cada elemento al centroide de un cluster.



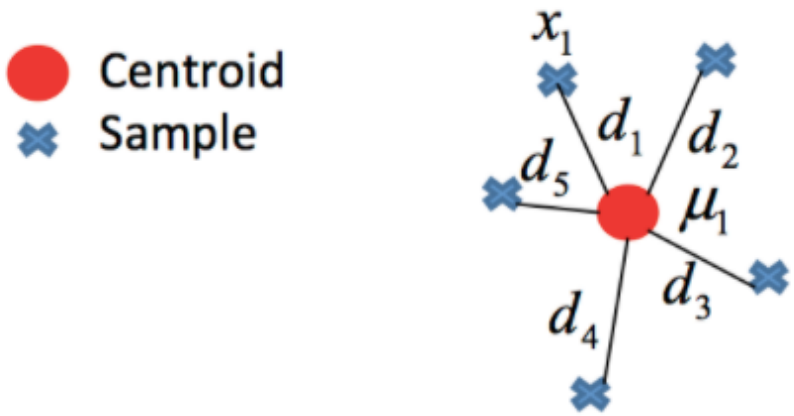




Clustering Particional

Cetroide:

- El centroide es el punto que ocupa la posición media en un cluster.
- Al inicio, cuando se empieza a definir el cluster, es probable que el centroide no tenga relación con algunos de los elementos.
- Posteriormente, la ubicación del centroide se calcula de manera iterativa.



Clustering Particional

Pseudocódigo

1. **Inicio:** Se establecen *k* centroides para la formación de *k* grupos. Estos centroides (elementos) se eligen aleatoriamente.
2. **Asignación:** Cada elemento es asignado al centroide más cercano.
3. **Actualización**: Se actualiza la posición del centroide con base en la media de los elementos asignados en el cluster.
4. **Repetir:** Se repiten los pasos 2 y 3 de manera iterativa hasta que los centroides no cambien más.

K-MEANS(P, k)

Input: a dataset of points $P = \{p_1, \ldots, p_n\}$, a number of clusters k
Output: centers $\{c_1, \ldots, c_k\}$ implicitly dividing P into k clusters

```
1  choose k initial centers C = {c_1, ..., c_k}
2  while stopping criterion has not been met
3      do ▷ assignment step:
4          for i = 1, ..., N
5              do find closest center c_k ∈ C to instance p_i
6                  assign instance p_i to set C_k
7          ▷ update step:
8          for i = 1, ..., k
9              do set c_i to be the center of mass of all points in C_i
```

Pseudocódigo

Para la asignación: Se asigna cada objeto al cluster más cercano, aplicando alguna medida de distancia (por ejemplo, distancia euclidiana, Manhattan, Chebyshev, y otros) entre el objeto y el centroide del cluster.

$$d_e(X,\mu) = \sqrt{\sum_{i=1}^{n}(x_i - \mu_i)^2}$$

$$d_{Manh}(p,q) = \sum_{i=1}^{n}|p_i - q_i| \qquad d_{Cheb}(p,q) = \max|p_i - q_i|$$

Para la actualización: Se calcula los nuevos centroides con base en la media de los elementos asignados en el cluster.

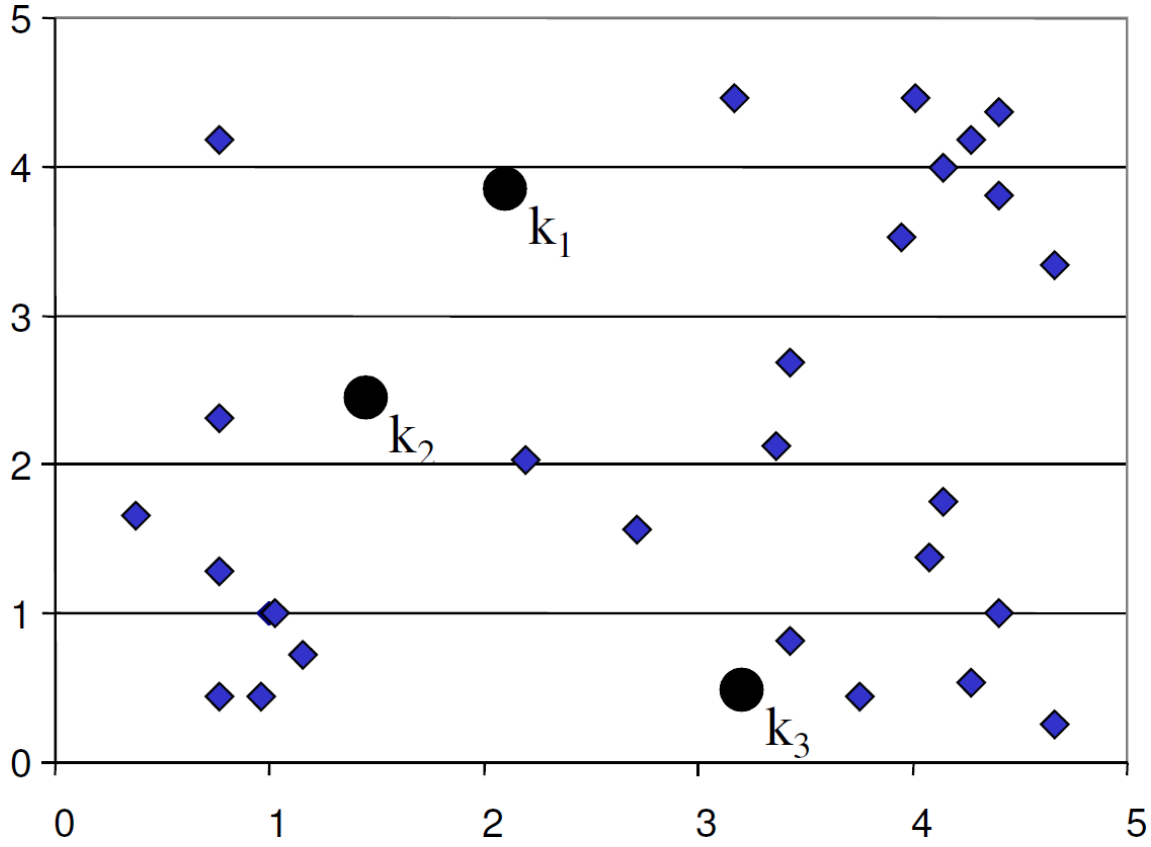
$$\mu = \frac{1}{N}\sum_{j=1}^{N} x_j$$

Procedimiento

Paso previo: Se elije el número de **K de grupos** en los que se asignarán los elementos**.**

1

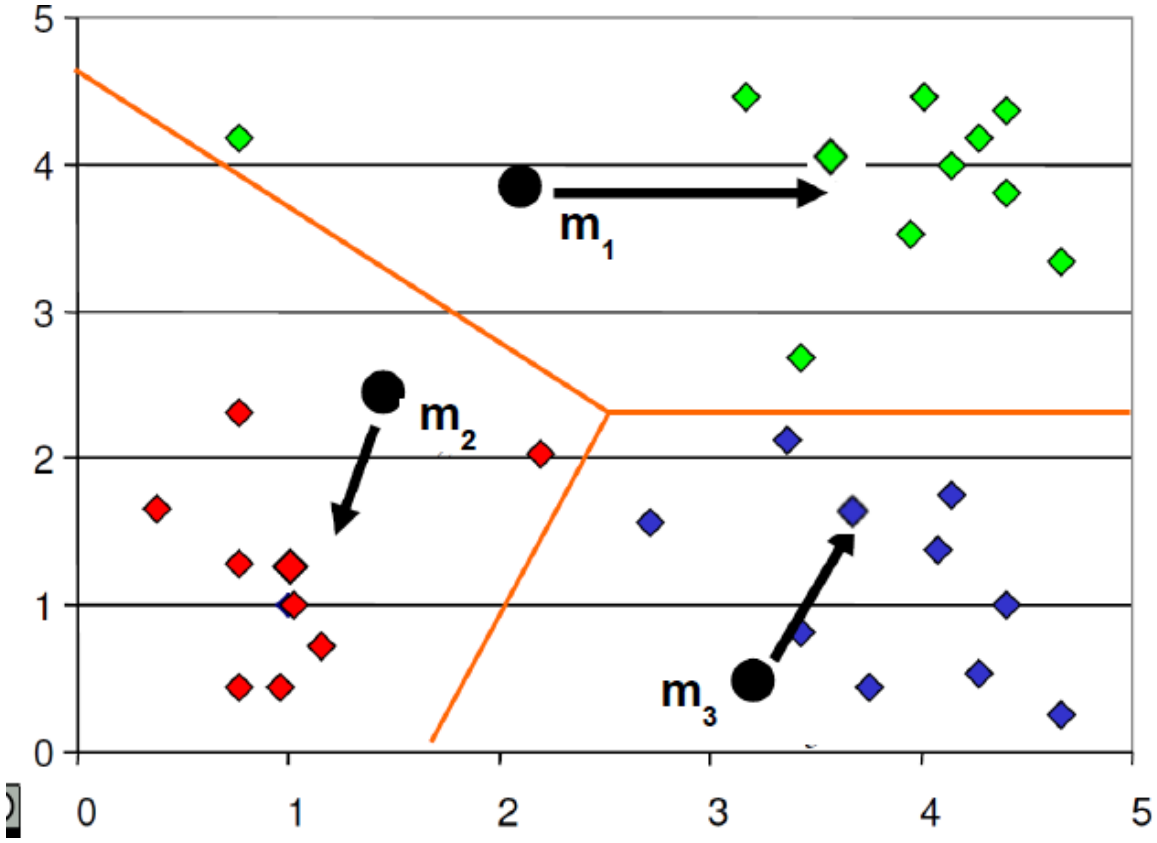
Paso 1: Seleccionar *k centroides* aleatoriamente. Estos serán los centros iniciales en los k grupos. Por ejemplo, 3 centroides (elementos).



Procedimiento

2

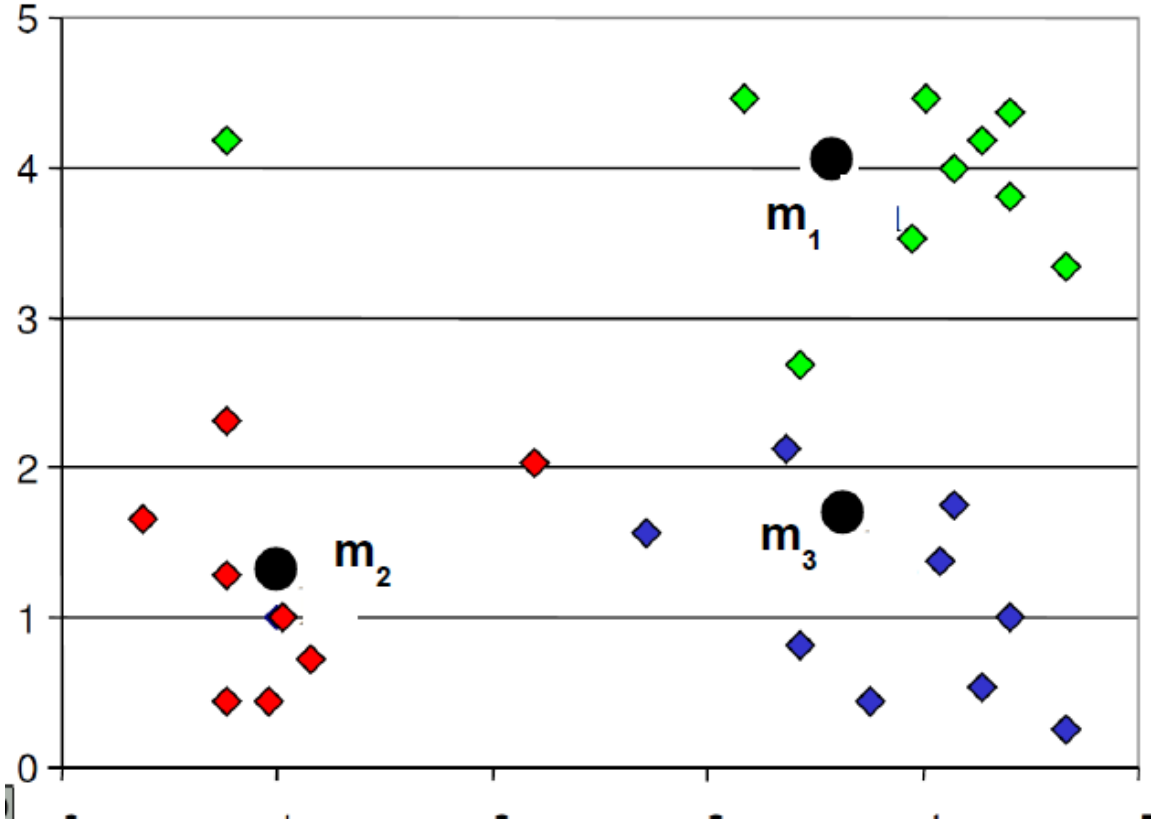
Paso 2: Se asigna cada elemento al centroide más cercano, creando así k clústeres.
Para la asignación se utiliza mediciones de **distancia mínima** entre el elemento y el centroide.



Procedimiento

3

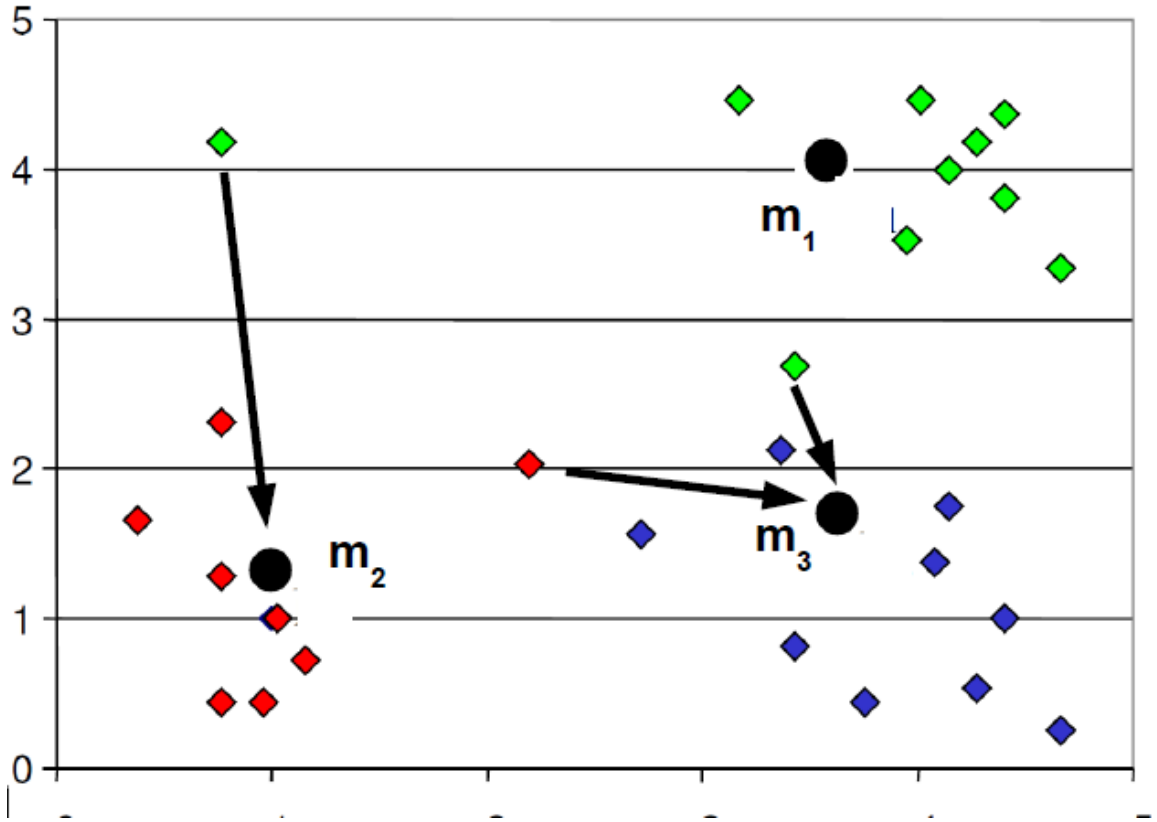
Paso 3: Una vez asignados todos los elementos, se actualiza la posición de los **centroides**, tomando como nuevo centro la posición del promedio de los elementos pertenecientes a cada cluster.



Procedimiento

4

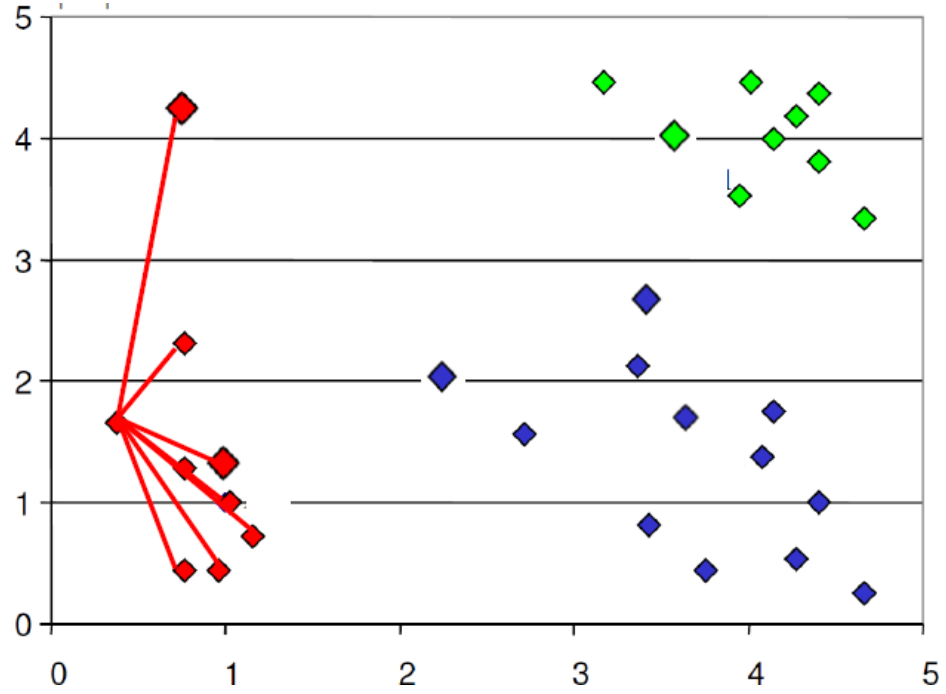
Paso 4: Se repiten los pasos **2 y 3**, se vuelven a asignar los elementos y se recalculan los centroides, hasta que éstos (centroides) no se modifiquen más, o se alcance un número máximo de iteraciones.



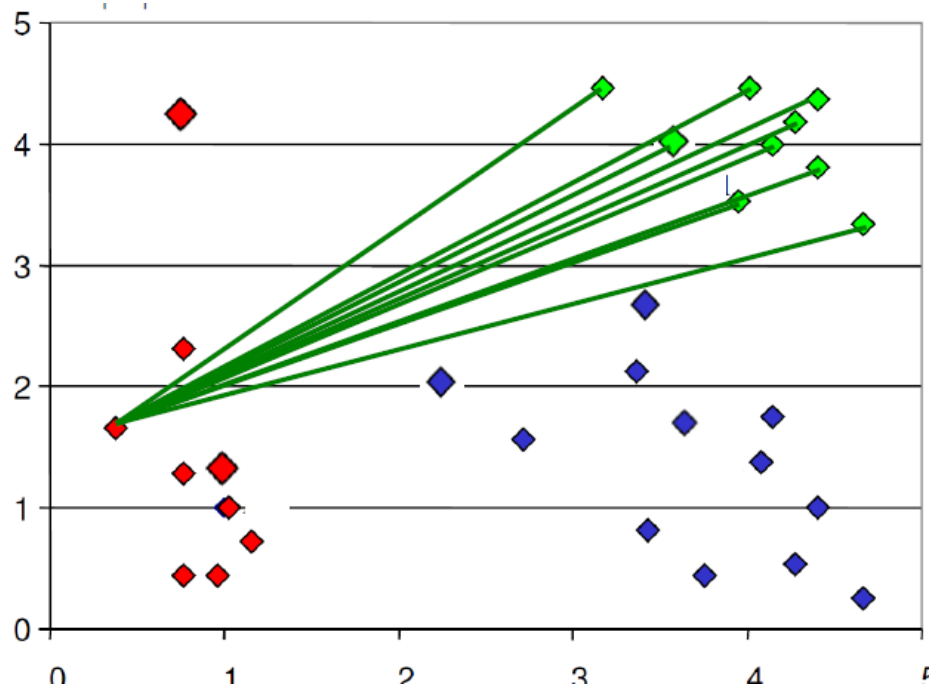
Clustering Particional

Lo que se busca

La similitud entre los elementos del mismo clúster sea alta. **Similitud intraclúster alta**.



La similitud entre los elementos de distintos clústeres sea baja. **Similitud interclúster baja**.



Clustering Particional

Procedimiento con una matriz de datos

Se quiere dividir una población de usuarios de un determinado sitio Web (Netflix) con base en sus edades: **n = 18**

15, 15, 16, 19, 19, 20, 20, 21, 22, 28, 35, 40, 41, 42, 43, 44, 60, 61

Distancia Euclidiana: $dist(p,q) = di_j = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$

Distancia de Manhattan: $d_{Manh}(p,q) = \sum_{i=1}^{n}|p_i - q_i|$

Distancia de Chebyshev: $d_{Cheb}(p,q) = \max|p_i - q_i|$

***k* = 3**

$c_1 = 16$
$c_2 = 22$
$c_3 = 60$

Procedimiento con una tabla de datos

Iteración 1

ID	x_i	c_1	c_2	c_3	Distancia 1	Distancia 2	Distancia 3	Cluster Cercano	Nuevo Centroide
1	15	16	22	60	1	7	45	1	
2	15	16	22	60	1	7	45	1	
3	16	16	22	60	0	6	44	1	**16.8**
4	19	16	22	60	3	3	41	1	
5	19	16	22	60	3	3	41	1	
6	20	16	22	60	4	2	40	2	
7	20	16	22	60	4	2	40	2	
8	21	16	22	60	5	1	39	2	
9	22	16	22	60	6	0	38	2	
10	28	16	22	60	12	6	32	2	**28.4**
11	35	16	22	60	19	13	25	2	
12	40	16	22	60	24	18	20	2	
13	41	16	22	60	25	19	19	2	
14	42	16	22	60	26	20	18	3	
15	43	16	22	60	27	21	17	3	
16	44	16	22	60	28	22	16	3	**50.0**
17	60	16	22	60	44	38	0	3	
18	61	16	22	60	45	39	1	3	

$k = 3$

$c_1 = 16$
$c_2 = 22$
$c_3 = 60$

$$d_{Manh}(p, q) = \sum_{i=1}^{n} |p_i - q_i|$$

$$Distancia\ 1 = |x_i - c_1|$$
$$Distancia\ 2 = |x_i - c_2|$$
$$Distancia\ 3 = |x_i - c_3|$$

Procedimiento con una tabla de datos

Iteración 2

ID	x_i	c_1	c_2	c_3	Distancia 1	Distancia 2	Distancia 3	Cluster Cercano	Nuevo Centroide
1	15	16.8	28.4	50	1.8	13.4	35	1	
2	15	16.8	28.4	50	1.8	13.4	35	1	
3	16	16.8	28.4	50	0.8	12.4	34	1	
4	19	16.8	28.4	50	2.2	9.4	31	1	
5	19	16.8	28.4	50	2.2	9.4	31	1	**18.6**
6	20	16.8	28.4	50	3.2	8.4	30	1	
7	20	16.8	28.4	50	3.2	8.4	30	1	
8	21	16.8	28.4	50	4.2	7.4	29	1	
9	22	16.8	28.4	50	5.2	6.4	28	1	
10	28	16.8	28.4	50	11.2	0.4	22	2	**31.5**
11	35	16.8	28.4	50	18.2	6.6	15	2	
12	40	16.8	28.4	50	23.2	11.6	10	3	
13	41	16.8	28.4	50	24.2	12.6	9	3	
14	42	16.8	28.4	50	25.2	13.6	8	3	
15	43	16.8	28.4	50	26.2	14.6	7	3	**47.3**
16	44	16.8	28.4	50	27.2	15.6	6	3	
17	60	16.8	28.4	50	43.2	31.6	10	3	
18	61	16.8	28.4	50	44.2	32.6	11	3	

$k = 3$

$c_1 = 16.8$
$c_2 = 28.4$
$c_3 = 50.0$

$$d_{Manh}(p,q) = \sum_{i=1}^{n} |p_i - q_i|$$

$$Distancia\ 1 = |x_i - c_1|$$
$$Distancia\ 2 = |x_i - c_2|$$
$$Distancia\ 3 = |x_i - c_3|$$

Procedimiento con una tabla de datos

Iteración 3

ID	x_i	c_1	c_2	c_3	Distancia 1	Distancia 2	Distancia 3	Cluster Cercano	Nuevo Centroide
1	15	18.6	31.5	47.3	3.6	16.5	32.3	1	
2	15	18.6	31.5	47.3	3.6	16.5	32.3	1	
3	16	18.6	31.5	47.3	2.6	15.5	31.3	1	
4	19	18.6	31.5	47.3	0.4	12.5	28.3	1	
5	19	18.6	31.5	47.3	0.4	12.5	28.3	1	**18.6**
6	20	18.6	31.5	47.3	1.4	11.5	27.3	1	
7	20	18.6	31.5	47.3	1.4	11.5	27.3	1	
8	21	18.6	31.5	47.3	2.4	10.5	26.3	1	
9	22	18.6	31.5	47.3	3.4	9.5	25.3	1	
10	28	18.6	31.5	47.3	9.4	3.5	19.3	2	**31.5**
11	35	18.6	31.5	47.3	16.4	3.5	12.3	2	
12	40	18.6	31.5	47.3	21.4	8.5	7.3	3	
13	41	18.6	31.5	47.3	22.4	9.5	6.3	3	
14	42	18.6	31.5	47.3	23.4	10.5	5.3	3	
15	43	18.6	31.5	47.3	24.4	11.5	4.3	3	**47.3**
16	44	18.6	31.5	47.3	25.4	12.5	3.3	3	
17	60	18.6	31.5	47.3	41.4	28.5	12.7	3	
18	61	18.6	31.5	47.3	42.4	29.5	13.7	3	

$k = 3$

$c_1 = 18.6$
$c_2 = 31.5$
$c_3 = 47.3$

$$d_{Manh}(p,q) = \sum_{i=1}^{n} |p_i - q_i|$$

$$Distancia\ 1 = |x_i - c_1|$$
$$Distancia\ 2 = |x_i - c_2|$$
$$Distancia\ 3 = |x_i - c_3|$$

Procedimiento con una tabla de datos

Iteración 4

ID	x_i	c_1	c_2	c_3	Distancia 1	Distancia 2	Distancia 3	Cluster Cercano	Nuevo Centroide
1	15	18.6	31.5	47.3	3.6	16.5	32.3	1	
2	15	18.6	31.5	47.3	3.6	16.5	32.3	1	
3	16	18.6	31.5	47.3	2.6	15.5	31.3	1	
4	19	18.6	31.5	47.3	0.4	12.5	28.3	1	
5	19	18.6	31.5	47.3	0.4	12.5	28.3	1	**18.6**
6	20	18.6	31.5	47.3	1.4	11.5	27.3	1	
7	20	18.6	31.5	47.3	1.4	11.5	27.3	1	
8	21	18.6	31.5	47.3	2.4	10.5	26.3	1	
9	22	18.6	31.5	47.3	3.4	9.5	25.3	1	
10	28	18.6	31.5	47.3	9.4	3.5	19.3	2	**31.5**
11	35	18.6	31.5	47.3	16.4	3.5	12.3	2	
12	40	18.6	31.5	47.3	21.4	8.5	7.3	3	
13	41	18.6	31.5	47.3	22.4	9.5	6.3	3	
14	42	18.6	31.5	47.3	23.4	10.5	5.3	3	
15	43	18.6	31.5	47.3	24.4	11.5	4.3	3	**47.3**
16	44	18.6	31.5	47.3	25.4	12.5	3.3	3	
17	60	18.6	31.5	47.3	41.4	28.5	12.7	3	
18	61	18.6	31.5	47.3	42.4	29.5	13.7	3	

$k = 3$

$c_1 = 18.6$
$c_2 = 31.5$
$c_3 = 47.3$

$$d_{Manh}(p,q) = \sum_{i=1}^{n} |p_i - q_i|$$

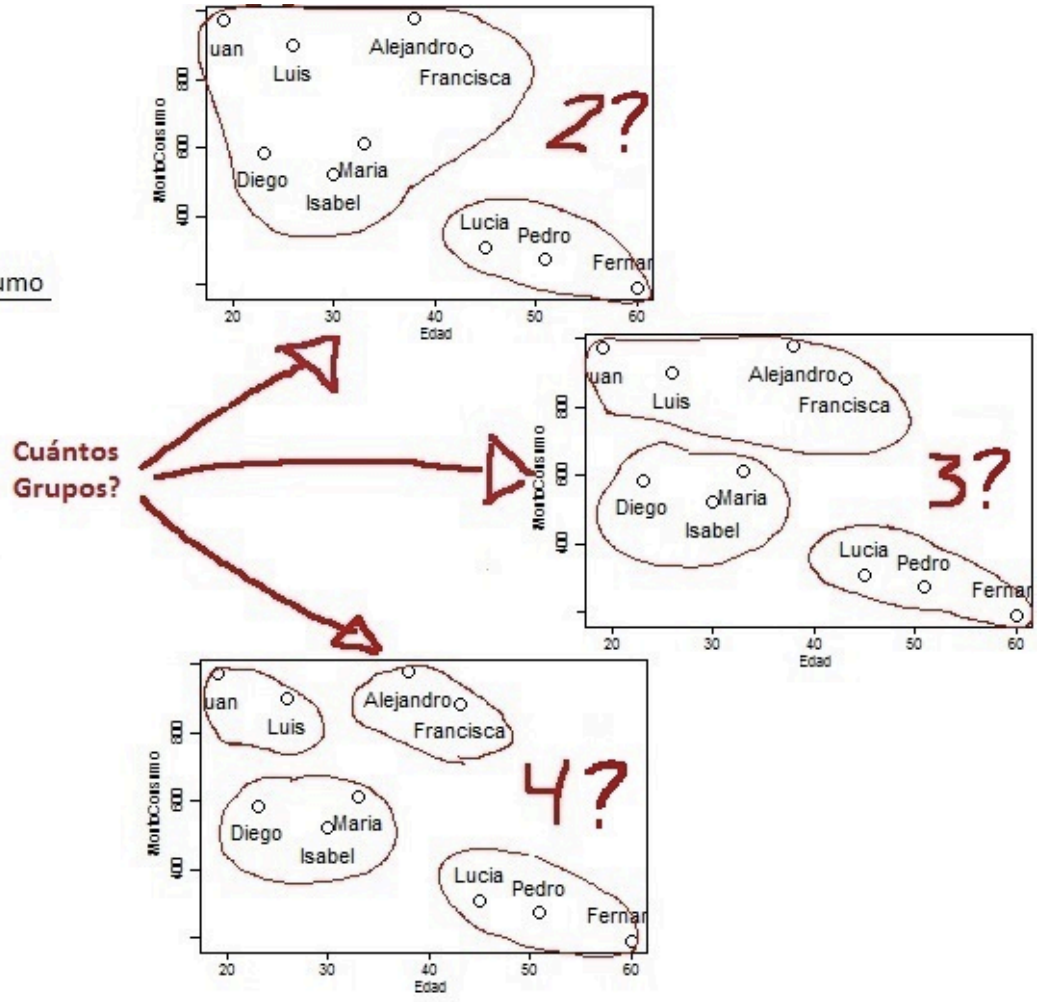
$$Distancia\ 1 = |x_i - c_1|$$
$$Distancia\ 2 = |x_i - c_2|$$
$$Distancia\ 3 = |x_i - c_3|$$

Método para decidir la cantidad de grupos

Clustering Particional

La idea básica de los algoritmos de partición, como k-means, es definir el número de grupos.

Nombre	Edad	MontoConsumo
Juan	19	971
Pedro	51	271
Maria	33	614
Isabel	30	521
Diego	23	585
Luis	26	898
Lucia	45	310
Francisca	43	884
Alejandro	38	979
Fernando	60	189



Elbow method

Elbow method es una herramienta gráfica útil para estimar el número óptimo de grupos.
El propósito es identificar el valor de k donde la distorsión (efecto del codo) cambia de manera significativa.

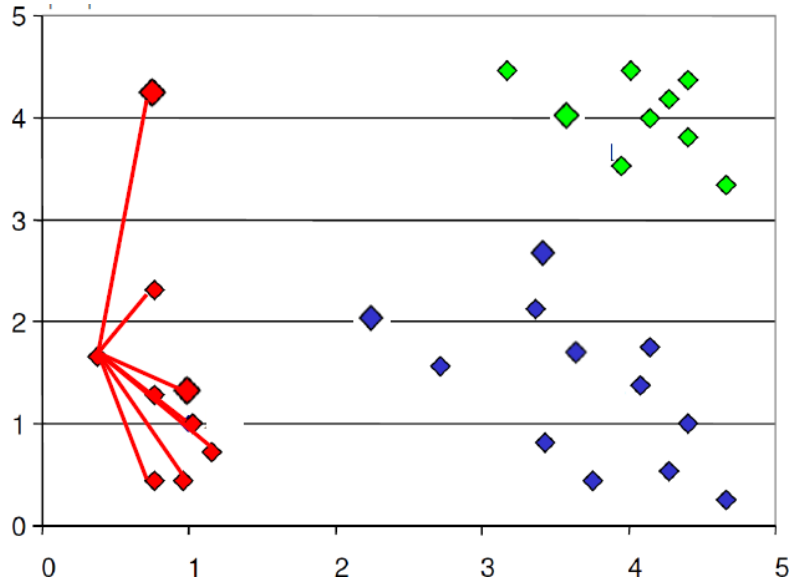
Para esto

- Para aplicar este método se debe calcular **SSE *–tot.withiness–*** (suma de la distancia al cuadrado entre cada elemento del cluster y su centroide) para varias configuraciones de k. Por ejemplo, k = 2, 3, 4, 5, 6, 7, 8 … n.

Número de cluster

$$SSE = \sum_{k=1}^{k} \text{dist}(x_i, uk) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - u_k)^2$$

Elemento del cluster Centroide



Clustering Particional

SSE

$$SSE = \sum_{k=1}^{k} \text{dist}(x_i, uk) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - u_k)^2$$

ID	x_i	c_1	c_2	c_3	Distancia 1	Distancia 2	Distancia 3	Cluster Cercano	Nuevo Centroide	$\lvert x_i - u_k \rvert$	$\lvert x_i - u_k \rvert^2$	SSE
1	15	18.6	31.5	47.3	3.6	16.5	32.3	1	18.6	3.6	12.96	54.24
2	15	18.6	31.5	47.3	3.6	16.5	32.3	1		3.6	12.96	
3	16	18.6	31.5	47.3	2.6	15.5	31.3	1		2.6	6.76	
4	19	18.6	31.5	47.3	0.4	12.5	28.3	1		0.4	0.16	
5	19	18.6	31.5	47.3	0.4	12.5	28.3	1		0.4	0.16	
6	20	18.6	31.5	47.3	1.4	11.5	27.3	1		1.4	1.96	
7	20	18.6	31.5	47.3	1.4	11.5	27.3	1		1.4	1.96	
8	21	18.6	31.5	47.3	2.4	10.5	26.3	1		2.4	5.76	
9	22	18.6	31.5	47.3	3.4	9.5	25.3	1		3.4	11.56	
10	28	18.6	31.5	47.3	9.4	3.5	19.3	2	31.5	3.5	12.25	24.50
11	35	18.6	31.5	47.3	16.4	3.5	12.3	2		3.5	12.25	
12	40	18.6	31.5	47.3	21.4	8.5	7.3	3	47.3	7.3	53.29	499.4
13	41	18.6	31.5	47.3	22.4	9.5	6.3	3		6.3	39.69	
14	42	18.6	31.5	47.3	23.4	10.5	5.3	3		5.3	28.09	
15	43	18.6	31.5	47.3	24.4	11.5	4.3	3		4.3	18.49	
16	44	18.6	31.5	47.3	25.4	12.5	3.3	3		3.3	10.89	
17	60	18.6	31.5	47.3	41.4	28.5	12.7	3		12.7	161.29	
18	61	18.6	31.5	47.3	42.4	29.5	13.7	3		13.7	187.69	
												578.17

Elbow method

Algoritmo

1. Calcular el agrupamiento para **diferentes valores de k**. Por ejemplo, *k* de 2 a 10 grupos.
2. Para cada *k*, calcular la suma total de la distancia al cuadrado dentro de cada grupo (**SSE,** conocido también como **WSS o *tot.withiness***).
3. **Trazar la curva de SSE** de acuerdo con el número de grupos k.
4. La ubicación de una curva (efecto del codo) en el gráfico se considera como un indicador del número adecuado de grupos.

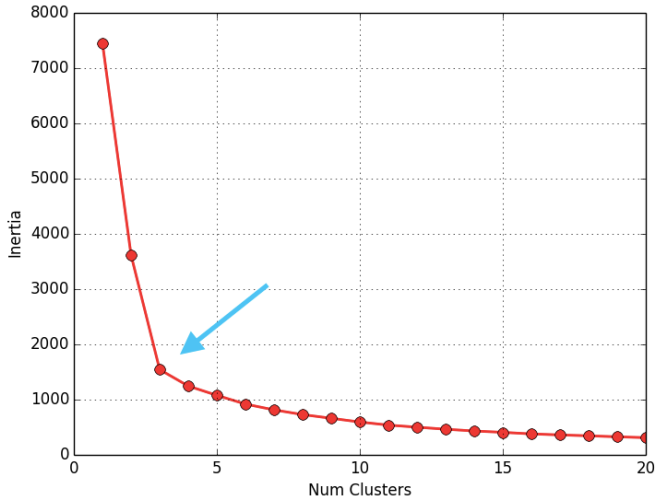
Número de cluster

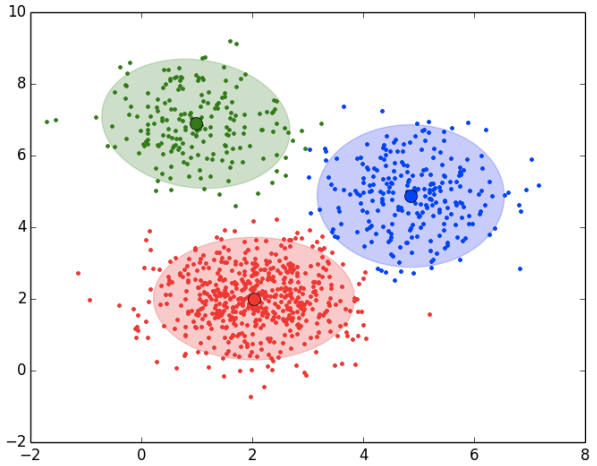
$$SSE = tot.withiness = \sum_{k=1}^{k} \text{dist}(x_i, uk) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - u_k)^2$$

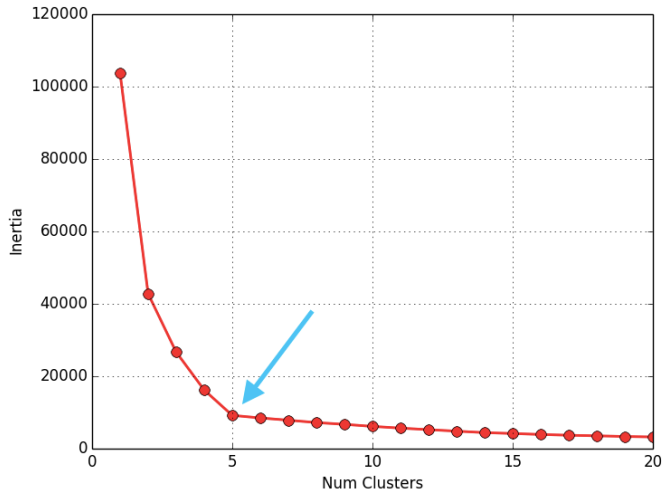
Elemento del cluster Centroide

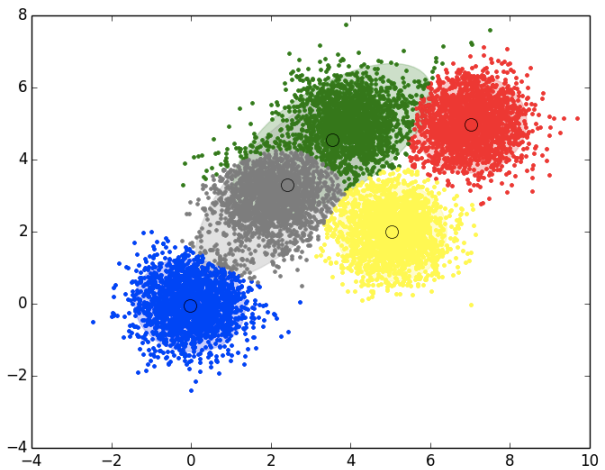
Elbow method

La idea básica de los métodos de partición, como k-medias, es definir el número de grupos.









Práctica 6
Clustering Particional

Práctica

Retomando el ejemplo sobre 'Empleados'

```
DatosEmp <- read.table("/Users/guille/Documents/1 FI-UNAM/1 Cursos/2021-1/1 IA2021-1/2 CasosPracticos/3 Similitudes/Empleados.txt", header=T, sep="\t")

DatosEmp
```

	ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAno	Antiguedad	Sexo
1	E1	10000	1	0	0	0	0	7	15	1
2	E2	20000	0	1	1	0	1	3	3	0
3	E3	15000	1	1	2	1	1	5	10	1
4	E4	30000	1	1	1	0	0	15	7	0
5	E5	10000	1	1	0	1	1	1	6	1
6	E6	40000	0	1	0	0	1	3	16	0
7	E7	25000	0	0	0	0	1	0	8	1
8	E8	20000	0	1	0	1	1	2	6	0
9	E9	20000	1	1	3	1	0	7	5	1
10	E10	30000	1	1	2	1	0	1	20	1
11	E11	45000	0	0	0	0	0	2	12	0
12	E12	8000	1	1	2	1	0	3	1	1
13	E13	20000	0	0	0	0	0	27	5	0
14	E14	10000	0	1	0	0	1	0	7	1
15	E15	8000	0	1	0	0	0	3	2	1

Práctica

1 Método para segmentar elementos (K-means)

```
k2 <- kmeans(DatosEmp[2:10], centers = 2, nstart = 10)
k3 <- kmeans(DatosEmp[2:10], centers = 3, nstart = 10)
k4 <- kmeans(DatosEmp[2:10], centers = 4, nstart = 10)
...
```

nstart = 10, generará 10 posibles centroides iniciales (configuraciones iniciales). Este debe ser mayor a 1.

Práctica

1 Método para segmentar elementos (K-means)

k3

```
K-means clustering with 3 clusters of sizes 2, 7, 6

Cluster means:
   Salario    Casado     Coche     Hijos  Vivienda Sindicato FaltasAnno Antiguedad      Sexo
1 42500.00 0.0000000 0.5000000 0.0000000 0.0000000 0.5000000   2.500000  14.000000 0.0000000
2 23571.43 0.4285714 0.7142857 1.0000000 0.4285714 0.4285714   7.857143   7.714286 0.4285714
3 10166.67 0.6666667 0.8333333 0.6666667 0.5000000 0.5000000   3.166667   6.833333 1.0000000

Clustering vector:
 [1] 3 2 3 2 3 1 2 2 2 2 1 3 2 3 3

Within cluster sum of squares by cluster:
[1]  12500010 135715078  32833511
 (between_SS / total_SS =  90.2 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"         "ifault"
```

Práctica

1 Método para segmentar elementos (K-means)

```
str(k3)      # str muestra de forma compacta la estructura interna del objeto
```

```
List of 9
 $ cluster     : int [1:15] 3 2 3 2 3 1 2 2 2 2 ...
 $ centers     : num [1:3, 1:9] 4.25e+04 2.36e+04 1.02e+04 0.00 4.29e-01 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:3] "1" "2" "3"
  .. ..$ : chr [1:9] "Salario" "Casado" "Coche" "Hijos" ...
 $ totss       : num 1.85e+09
 $ withinss    : num [1:3] 1.25e+07 1.36e+08 3.28e+07
 $ tot.withinss: num 1.81e+08
 $ betweenss   : num 1.67e+09
 $ size        : int [1:3] 2 7 6
 $ iter        : int 2
 $ ifault      : int 0
 - attr(*, "class")= chr "kmeans"
```

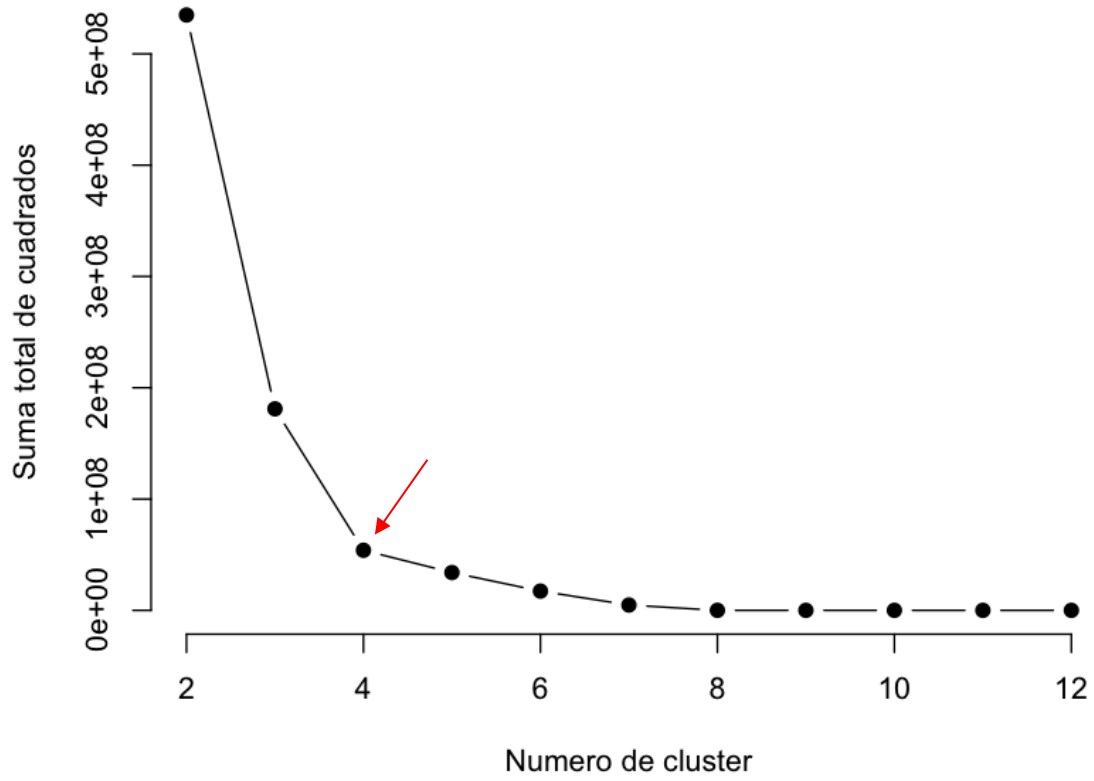
2 Obtención de número de grupos (Elbow method)

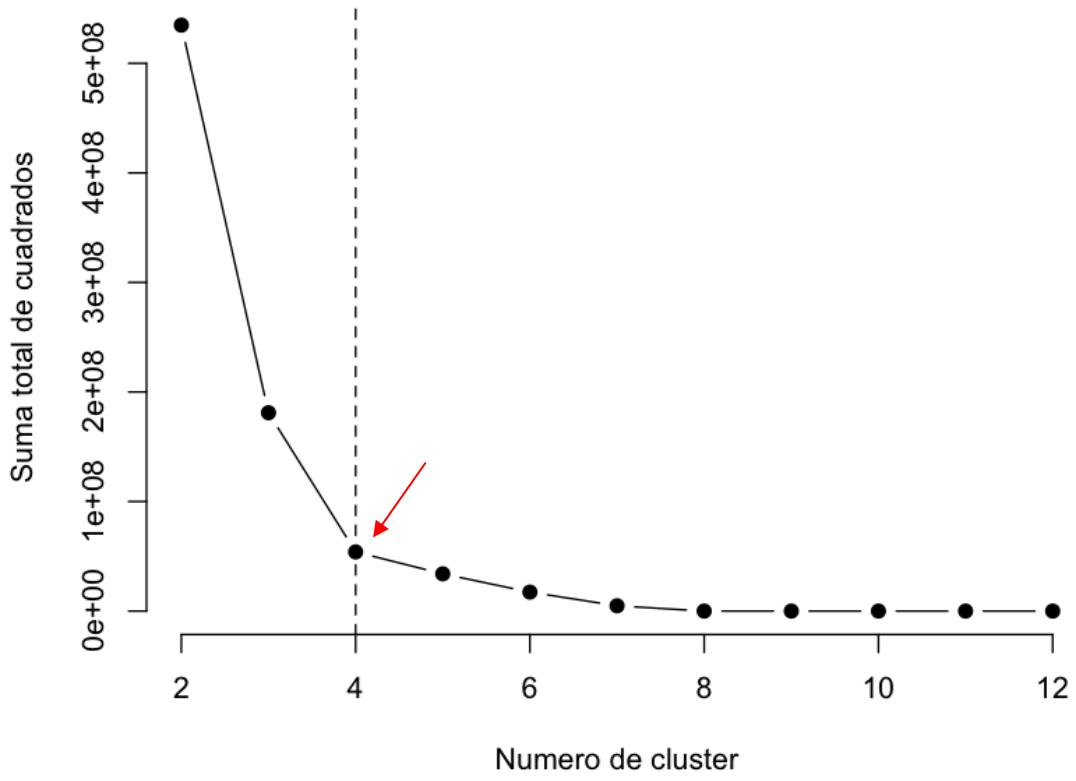
```
set.seed(123)
SSE <- sapply(2:12,
              function(k)
              {kmeans(DatosEmp[2:10], k, nstart=10)$tot.withinss})
plot(2:12, SSE, type = "b", pch = 19, frame = FALSE, xlab = "Numero de cluster",  ylab = "Suma total de cuadrados")

# La función sapply simplifica la salida de los resultados de un vector o matriz
# nstart = 10, generará 10 posibles centroides iniciales (configuraciones iniciales).
```

Práctica

2 Obtención de número de grupos (Elbow method)

```
# Línea en el número deseado de grupos
abline(v = 4, lty =2)
```





3 Interpretación

```
k4 <- kmeans(DatosEmp[2:10], centers = 4, nstart = 10)
k4
```

```
K-means clustering with 4 clusters of sizes 3, 2, 5, 5

Cluster means:
   Salario    Casado     Coche Hijos  Vivienda Sindicato FaltasAnno Antiguedad      Sexo
1 28333.33 0.6666667 0.6666667   1.0 0.3333333 0.3333333   5.333333   11.66667 0.6666667
2 42500.00 0.0000000 0.5000000   0.0 0.0000000 0.5000000   2.500000   14.00000 0.0000000
3  9200.00 0.6000000 0.8000000   0.4 0.4000000 0.4000000   2.800000    6.20000 1.0000000
4 19000.00 0.4000000 0.8000000   1.2 0.6000000 0.6000000   8.800000    5.80000 0.4000000

Clustering vector:
 [1] 3 4 4 1 3 2 1 4 4 1 2 3 4 3 3

Within cluster sum of squares by cluster:
[1] 16666917 12500010  4800159 20000468
 (between_SS / total_SS =  97.1 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"         "ifault"
```

Práctica

3 Interpretación

```
Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"         "ifault"
```

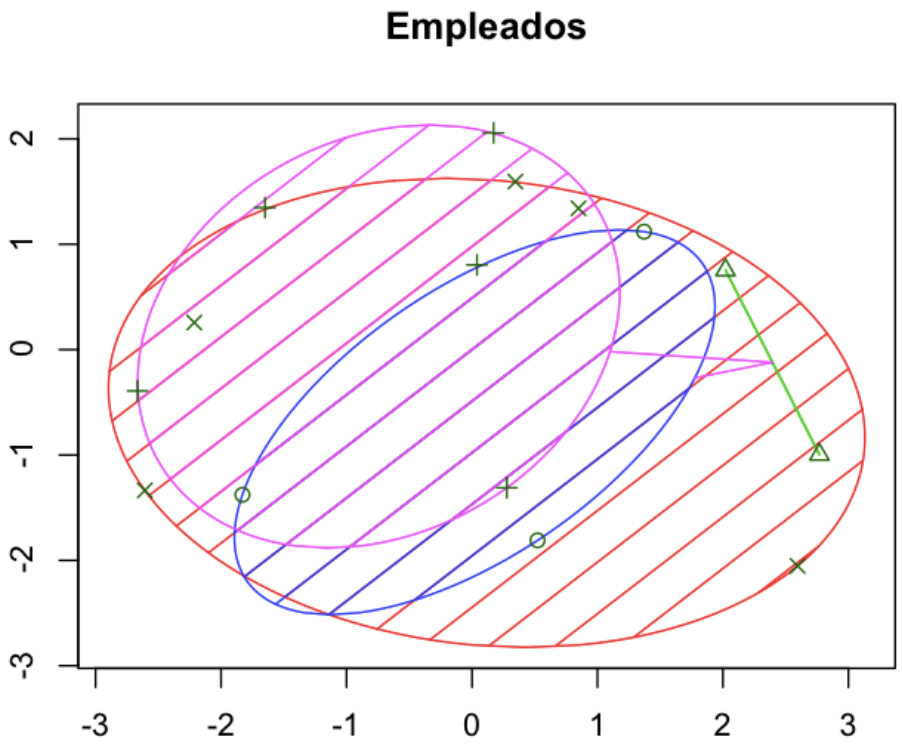
k4$cluster
k4$centers
k4$totss
k4$withinss
k4$tot.withinss
k4$size

```
> k4$size
[1] 3 2 5 5
> k4$withinss
[1] 16666917 12500010  4800159 20000468
> k4$tot.withinss
[1] 53967554
```

$$SSE = tot.withiness = \sum_{k=1}^{k} \text{dist}(x_i, uk) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - u_k)^2$$

3 Interpretación

```
cluster::clusplot(DatosEmp[2:10], k4$cluster, color=T, shade=T, main='Empleados')
```



3 Interpretación

	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAnno	Antiguedad	Sexo
1	28333.33	0.6666667	0.6666667	1.0	0.3333333	0.3333333	5.333333	11.66667	0.6666667
2	42500.00	0.0000000	0.5000000	0.0	0.0000000	0.5000000	2.500000	14.00000	0.0000000
3	9200.00	0.6000000	0.8000000	0.4	0.4000000	0.4000000	2.800000	6.20000	1.0000000
4	19000.00	0.4000000	0.8000000	1.2	0.6000000	0.6000000	8.800000	5.80000	0.4000000

Cluster 1: 3 empleados

Salario :	28333
Casado :	Si = 0.67 / No = 0.33
Coche :	Si = 0.67 / No = 0.33
Hijos :	1
Vivienda :	Prop = 0.33 Alquiler = 0.67
Sindicato :	Si = 0.33 / No = 0.67
Faltas/Año :	5.3 (5)
Antigüedad :	11.6 (12)
Sexo :	M = 0.67 / F = 0.33

...

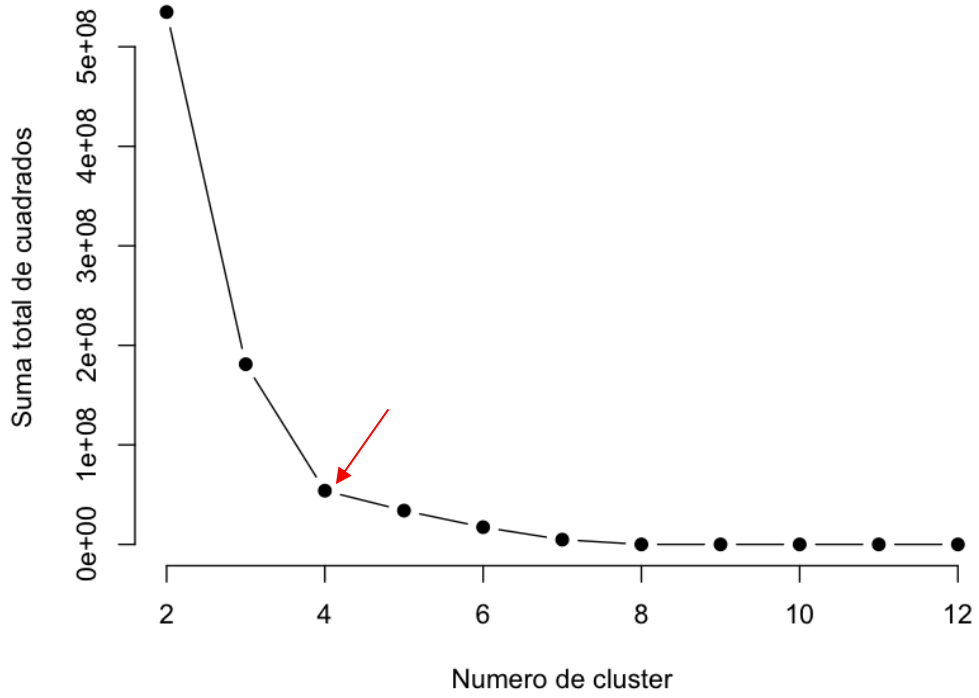
	ID	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAno	Antiguedad	Sexo
1	E1	10000	1	0	0	0	0	7	15	1
2	E2	20000	0	1	1	0	1	3	3	0
3	E3	15000	1	1	2	1	1	5	10	1
4	E4	30000	1	1	1	0	0	15	7	0
5	E5	10000	1	1	0	1	1	1	6	1
6	E6	40000	0	1	0	0	1	3	16	0
7	E7	25000	0	0	0	0	1	0	8	1
8	E8	20000	0	1	0	1	1	2	6	0
9	E9	20000	1	1	3	1	0	7	5	1
10	E10	30000	1	1	2	1	0	1	20	1
11	E11	45000	0	0	0	0	0	2	12	0
12	E12	8000	1	1	2	1	0	3	1	1
13	E13	20000	0	0	0	0	0	27	5	0
14	E14	10000	0	1	0	0	1	0	7	1
15	E15	8000	0	1	0	0	0	3	2	1

- **Cluster 1 [3 elementos –4, 7, 10–]**. Empleados con salario promedio de $28333, casados en su mayoría (67%), con coche en su mayoría (67%) y con un hijo. No tienen vivienda propia en su mayoría (67%), no sindicalizados en su mayoría (67%), con varias faltas al año (5), con una antigüedad promedio de 12 años y la mayoría varones (67%).

Práctica

K-means

	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAnno	Antiguedad	Sexo
1	28333.33	0.6666667	0.6666667	1.0	0.3333333	0.3333333	5.333333	11.66667	0.6666667
2	42500.00	0.0000000	0.5000000	0.0	0.0000000	0.5000000	2.500000	14.00000	0.0000000
3	9200.00	0.6000000	0.8000000	0.4	0.4000000	0.4000000	2.800000	6.20000	1.0000000
4	19000.00	0.4000000	0.8000000	1.2	0.6000000	0.6000000	8.800000	5.80000	0.4000000



Jerárquico Ascendente

	Salario	Casado	Coche	Hijos	Vivienda	Sindicato	FaltasAnno	Antiguedad	Sexo
[1,]	9200.00	0.6000000	0.8000000	0.4	0.4000000	0.4000000	2.800000	6.20000	1.0000000
[2,]	19000.00	0.4000000	0.8000000	1.2	0.6000000	0.6000000	8.800000	5.80000	0.4000000
[3,]	28333.33	0.6666667	0.6666667	1.0	0.3333333	0.3333333	5.333333	11.66667	0.6666667
[4,]	42500.00	0.0000000	0.5000000	0.0	0.0000000	0.5000000	2.500000	14.00000	0.0000000