Clustering

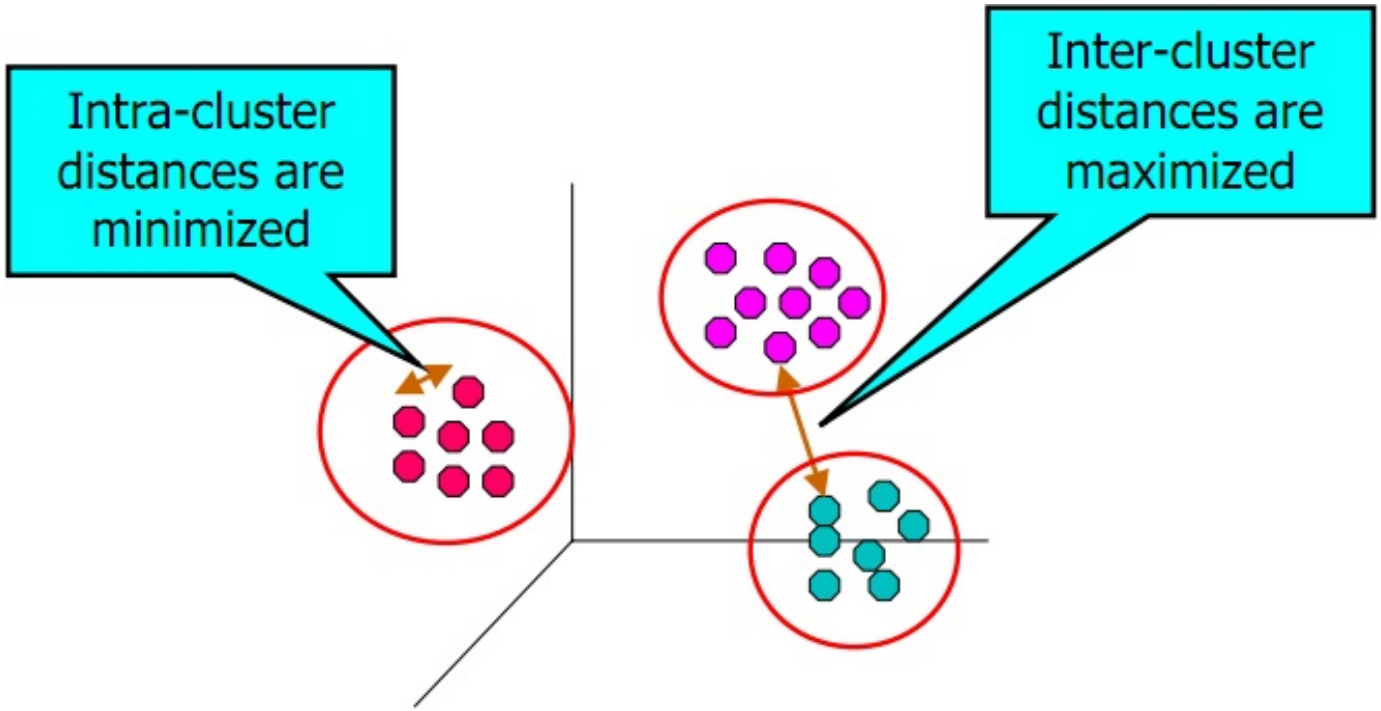
Dra. Amparo López Gaona

Fac. Ciencias, UNAM
Mayo 2018

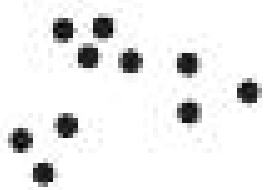
Introducción

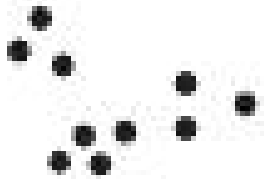
- Supón que tienes 5 ejecutivos trabajando para ti y quieres dividir a los clientes de una compañía en 5 grupos, para asignar un ejecutivo a cada grupo, con la intensión de crear campañas específicas basadas en las características comunes que comparten como grupo.
- ¿Cuál de las técnicas estudiadas utilizarías?

- Clustering es una técnica usada para agrupar objetos similares en el mismo grupo (cluster).
- Cluster: colección de objetos/datos.
  - Similares a otros objetos dentro del mismo cluster.
  - Diferentes a los objetos de otros clusters.



- Análisis de clusters
  - Encontrar similitudes entre datos de acuerdo a las características de ellos y agrupar los similares en clusters.





- Análisis de clusters
  - Encontrar similitudes entre datos de acuerdo a las características de ellos y agrupar los similares en clusters.

How many clusters?

Six Clusters

Two Clusters

Four Clusters

... Introducción

- Aprendizaje no-supervisado: no hay clases predefinidas.
- Utilizado como:
  - Herramienta, independiente, para conocer distribución de datos.
  - Un paso de preprocesamiento para otros algoritmos.
- Cada cluster puede verse como una clase de la cual pueden derivarse reglas.
- Haciendo clustering, se pueden descubrir patrones de distribución y correlaciones interesantes entre los atributos de los datos.
- Ejemplo: Realizar análisis de cluster para identificar subpoblaciones homogéneas de clientes, quizá para hacer marketing.

Aplicaciones del análisis de clusters

- **Entender.** Clases o grupos de objetos que comparten características comunes juegan un papel importante en cómo la gente analiza y describe el mundo.
  - Grupos de documentos relacionados para hacer browsing.
  - Grupos de genes y proteínas que tienen funcionalidad similar,
  - Grupos de artículos con fluctuaciones similares de precio.
- **Resumir.** Reducir el tamaño de grandes conjuntos.

Ejemplos de aplicaciones de clustering

- Ventas. Ayuda a descubrir distintos grupos en sus BD de clientes considerando patrones de compra, con la intensión de crear campañas de venta efectivas.
- Detección de comportamiento anómalo, tal como intrusos en redes.
- Simplificar datasets extremadamente grandes agrupando.
- Aseguradoras. Identifican grupos de titulares de pólizas de seguros de autos que tienen un alto promedio de costo de reclamaciones.
- Planeación de ciudades. Identifican grupos de casas de acuerdo a su tipo, valor y ubicación geográfica.
- Estudios de terremotos. Observaciones de epicentros de temblores podrían agruparse a lo largo fallas.

En general, el clustering es útil cuando gran cantidad de datos variados puede caracterizarse por un número pequeño de grupos.

¿Qué no es análisis de cluster?

- Clasificación supervisada.
  - Aprende un método para predecir la clase de tuplas pre-etiquetadas.
  - Las clases necesitan una etiqueta de clase.
- Segmentación sencilla.
  - Dividir a los estudiantes en diferentes grupos de registro de acuerdo a su apellido paterno.
- El resultado de una consulta (que utiliza `GROUP BY`)

Requerimientos de clustering en MD

- Escalabilidad.
- Habilidad de tratar con diferentes tipos de atributos.
- Requerimientos mínimos de conocimiento del dominio para determinar los parámetros de entrada.
- Habilidad para tratar con ruido en los datos.
- Alta dimensionalidad.

Pasos básicos para hacer agrupamientos

- Seleccionar.
  - Seleccionar los datos concernientes a las tareas de interés.
  - Minimizar los datos redundantes.
- Elegir una medida de proximidad.
  - Similitud de dos vectores de características.
- Elegir criterios de agrupación (clustering).
  - Expresados vía una función de costo o algunas reglas.
- Elegir un algoritmo de clustering.
- Validar los resultados.

Características de un buen cluster

- Un buen clustering produce clusters con:
  - Alta similitud intra clases, y
  - Baja similitud inter clases.
- La calidad del clustering:
  - Depende de la medida de similitud usada y su implementación.
  - Se mide por la habilidad de descubrir algunos o todos los patrones ocultos.
- Sin embargo, una evaluación objetiva es problemática: usualmente hecha por humanos/ inspección de expertos.
- El análisis de cluster se ha estudiado por años, principalmente basado en distancia

Similitud y disimilitud

- Similitud
  - Medida numérica de qué tan parecidos son dos objetos.
  - Un valor es alto cuando los objetos son más parecidos.
  - Casi siempre cae en el rango de [0,1]
- Disimilitud.
  - Medida numérica de qué tan diferentes son dos objetos.
  - Un valor es bajo cuando los objetos son más parecidos.
  - Con mínima diferencia, es casi siempre 0.
  - El limite superior varía.
- Proximidad, se refiere a la similitud o disimilitud.
- Es difícil definir "suficientemente similar" o "suficientemente bueno"
  - La respuesta típicamente es altamente subjetivo.

Matriz de datos y de disimilitud/disparidad

- Matriz de datos: n objetos con p dimensiones o atributos

$$\begin{pmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{pmatrix}$$

- Matriz de disimilitud (diferencias). Son n puntos pero sólo se registra la distancia entre ellos en una matriz triangular.

$$\begin{pmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{pmatrix}$$

$d_{ij} \geq 0$ cercano a cero si son muy similares mayor si ellos difieren.

Medidas de distancia

- Existen funciones de distancia para los diferentes tipos de variables.
- Las funciones de distancia tienen las siguientes propiedades:
  - $d(i,j) \geq 0$
  - $d(i,i) = 0$
  - $d(i,j) = d(j,i)$
  - $d(i,j) \leq d(i,k) + d(k,j)$
- Una distancia que satisface estas propiedades se conoce como **métrica**.

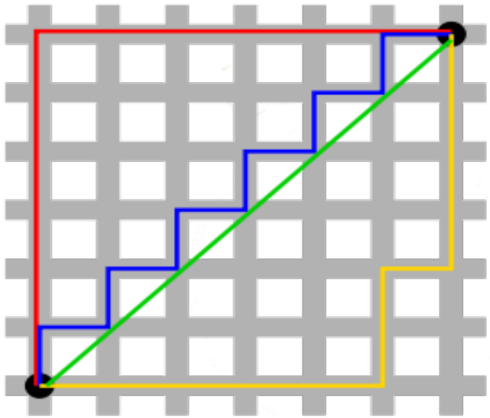
Similitud de datos numéricos

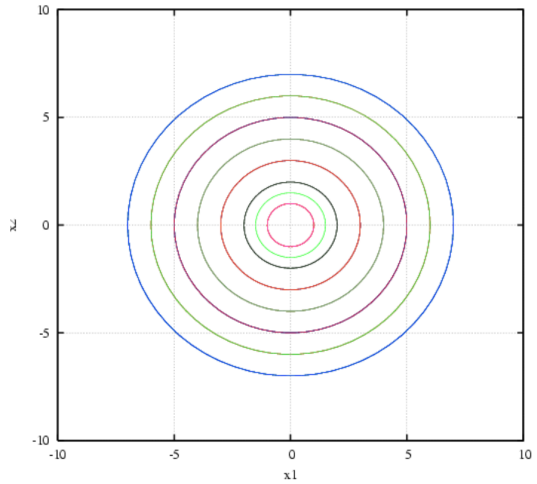
- Distancia de Manhattan (del taxista)
  Cantidad de unidades horizontales y verticales que se requieren para ir de un punto (valor real) a otro.

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + ... + |x_{ip} - x_{jp}|$$

- Distancia Euclidiana.

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ... + (x_{ip} - x_{jp})^2)}$$





Ejemplo de distancia

punto	atributo1	atributo2
x1	1	2
x2	3	5
x3	2	0
x4	4	5

Manhattan (L_1)

L	x1	x2	x3	x4
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Euclidiana (L_2)

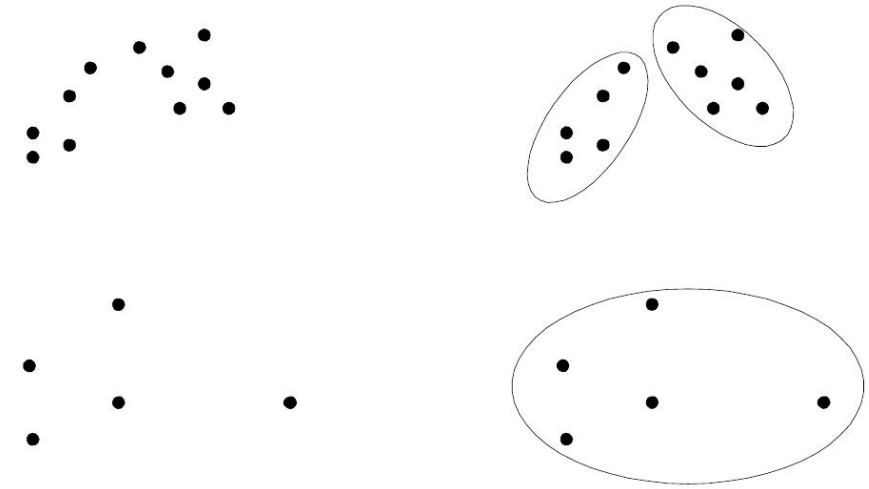
L_2	x1	x2	x3	x4
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Principales enfoques para clustering

- Particionamiento.
  - Se construyen varias particiones y luego se evalúan por algún criterio. Por ejemplo, minimizar la suma de los cuadrados de los errores.
  - Métodos típicos: k-media, k-medoides,
- Métodos jerárquicos.
  - Se crea un conjunto de clusters anidados organizados como un árbol jerárquico.
- Métodos basados en la densidad.
  - Continúan creciendo hasta que la densidad (cantidad de objetos) en el vecino exceda algún umbral.

Métodos de particionamiento

- Método de particionamiento: Construir una partición de una BD con n objetos en un conjunto de k clusters.



- Juntos deben satisfacer los siguientes criterios:
  - Cada grupo debe contener al menos un objeto, y
  - Cada objeto debe pertenecer exactamente a un grupo.
- Varios métodos
  - k-medias: Cada cluster se representa por el centro del mismo.
  - k-medoides o PAM (Partition around medoids): Cada cluster se representa por uno de los objetos del mismo.

Algoritmo k-medias

In: El número de clusters k y una BD con n objetos.
Out: Un conjunto de k clusters

1. Elegir arbitrariamente k objetos como centros iniciales (centroide) de los clusters.
2. Repetir
   1. (Re) asignar cada objeto al cluster en el cual sea más similar, considerando la distancia entre el objeto y el centroide del cluster.
   2. Actualizar la media del cluster, es decir, calcular el valor medio de los objetos de cada cluster;

   hasta que no haya cambios.

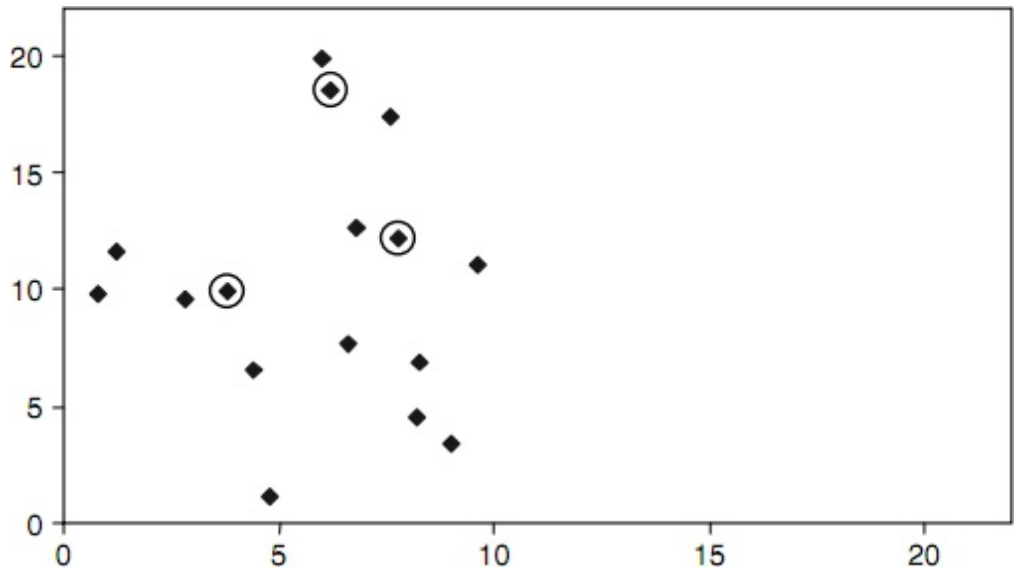
La calidad del cluster C_i se mide por la variación dentro del mismo, la cual es la suma del error al cuadrado entre los objetos en el cluster y el centroide.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, m_i)^2$$

p = punto en el espacio, ie, un objeto dado; m_i es la media del cluster C_i

... Algoritmo k-medias (Ejemplo)

x	y
6.8	12.6
0.8	9.8
1.2	11.6
2.8	9.6
3.8	9.9
4.4	6.5
4.8	1.1
6.0	19.9
6.2	18.5
7.6	17.4
7.8	12.2
6.6	7.7
8.2	4.5
8.4	6.9
9.0	3.4
9.6	11.1

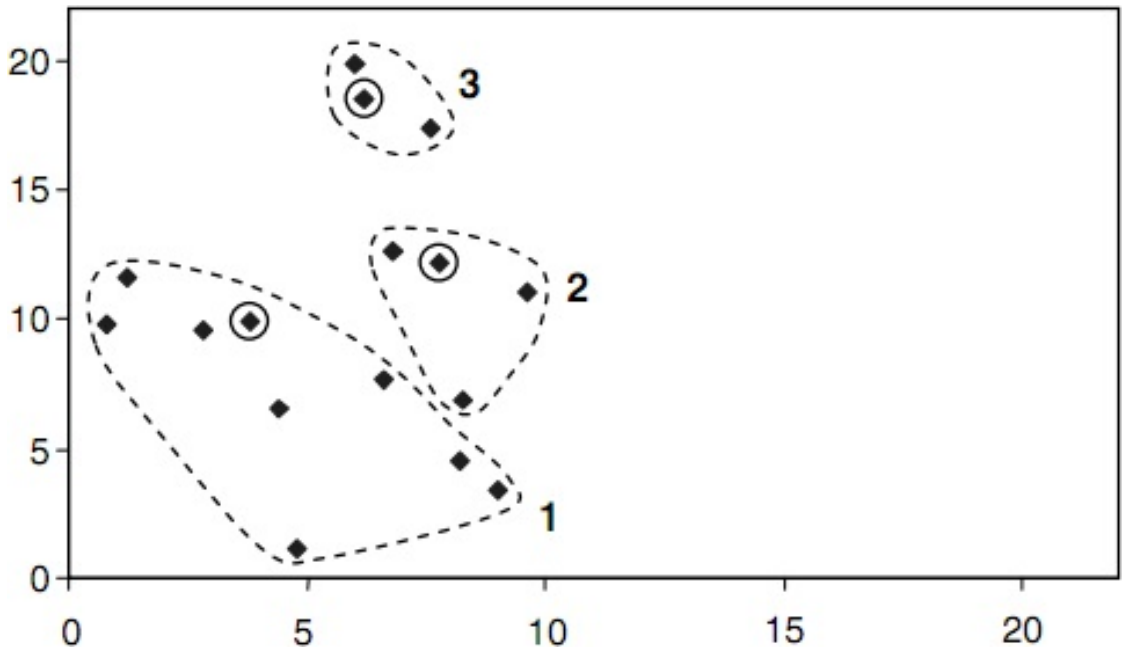


... Algoritmo k-medias (Ejemplo)

	Initial	
	x	y
Centroid 1	3.8	9.9
Centroid 2	7.8	12.2
Centroid 3	6.2	18.5

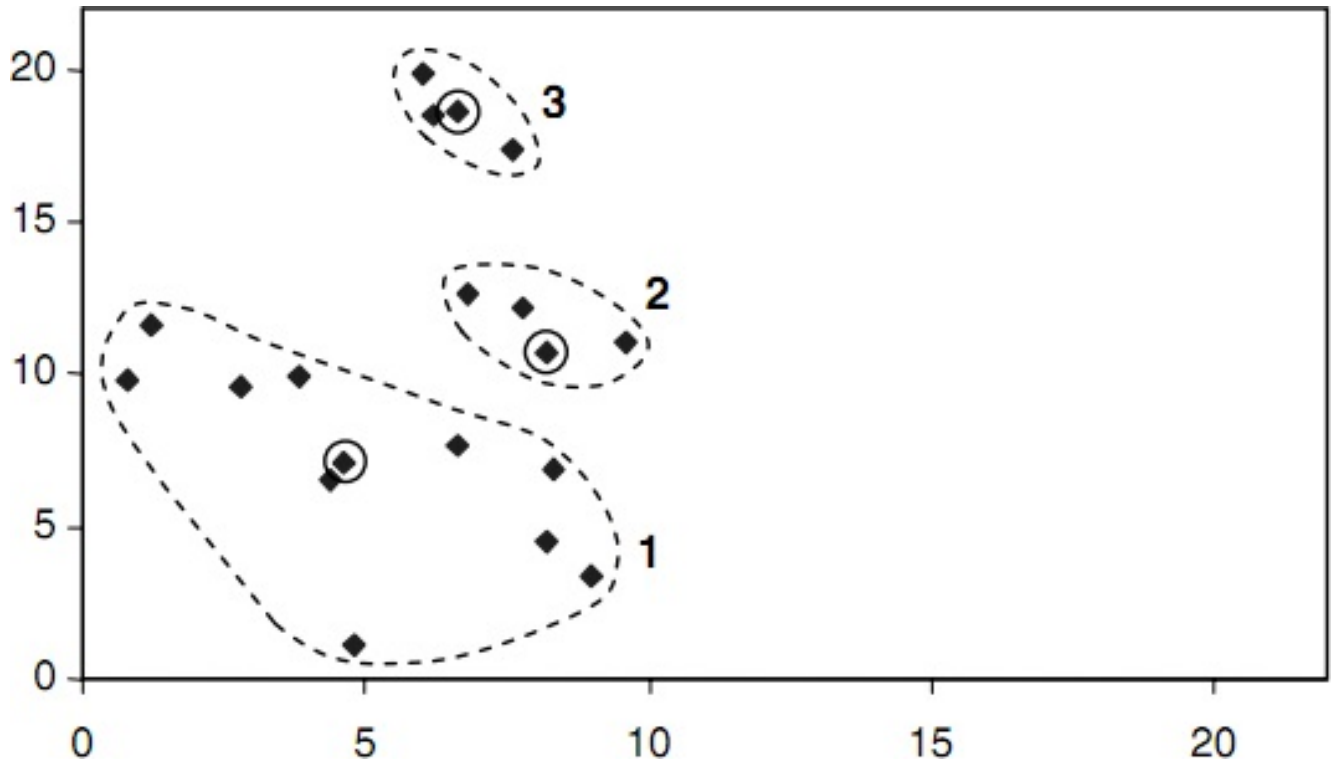
x	y	$d1$	$d2$	$d3$	cluster
6.8	12.6	4.0	1.1	5.9	2
0.8	9.8	3.0	7.4	10.2	1
1.2	11.6	3.1	6.6	8.5	1
2.8	9.6	1.0	5.6	9.5	1
3.8	9.9	0.0	4.6	8.9	1
4.4	6.5	3.5	6.6	12.1	1
4.8	1.1	8.9	11.5	17.5	1
6.0	19.9	10.2	7.9	1.4	3
6.2	18.5	8.9	6.5	0.0	3
7.6	17.4	8.4	5.2	1.8	3
7.8	12.2	4.6	0.0	6.5	2
6.6	7.7	3.6	4.7	10.8	1
8.2	4.5	7.0	7.7	14.1	1
8.4	6.9	5.5	5.3	11.8	2

... Algoritmo k-medias (Ejemplo)

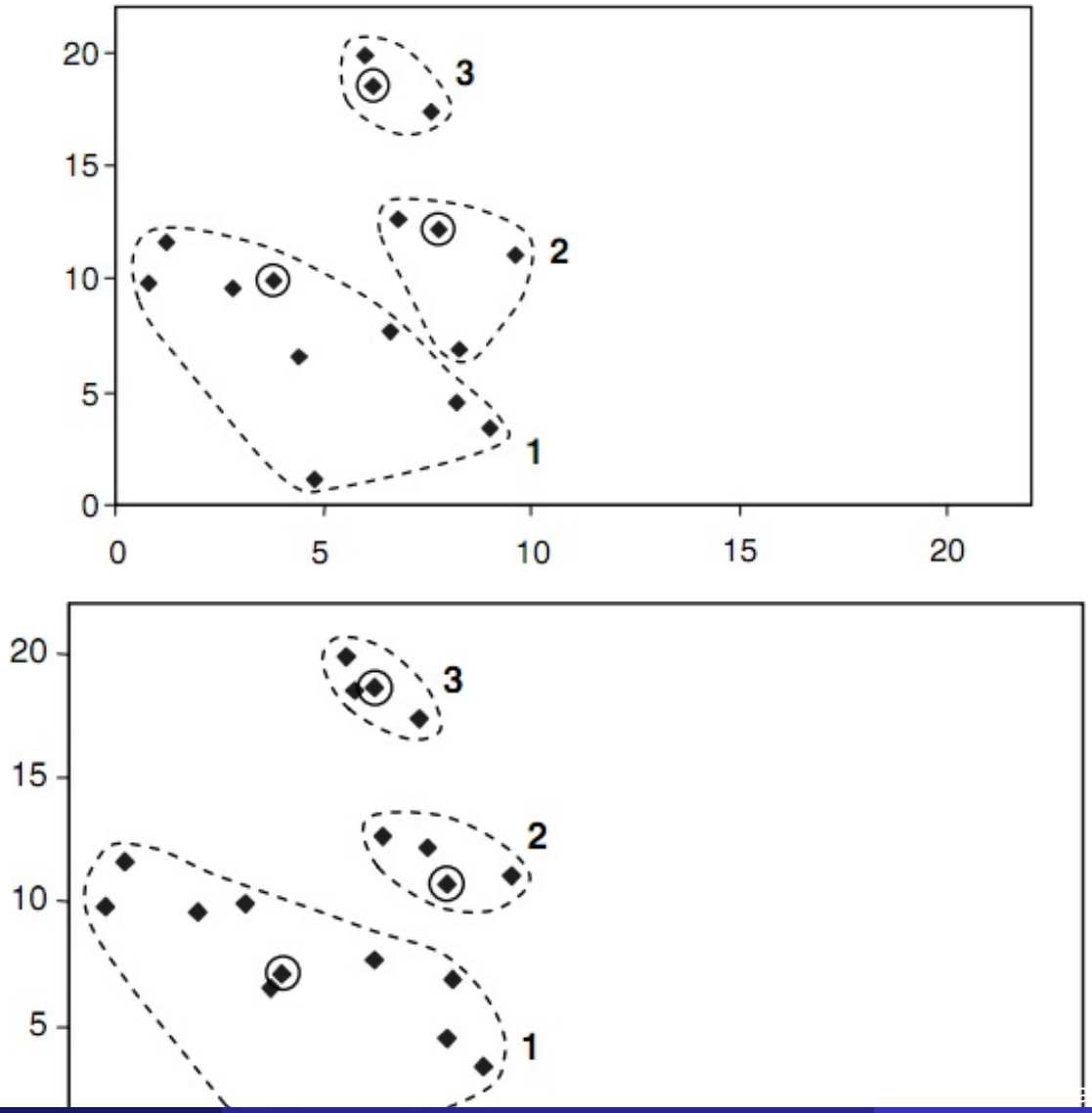


... Algoritmo k-medias (Ejemplo)

	Initial		After first iteration	
	x	y	x	y
Centroid 1	3.8	9.9	4.6	7.1
Centroid 2	7.8	12.2	8.2	10.7
Centroid 3	6.2	18.5	6.6	18.6

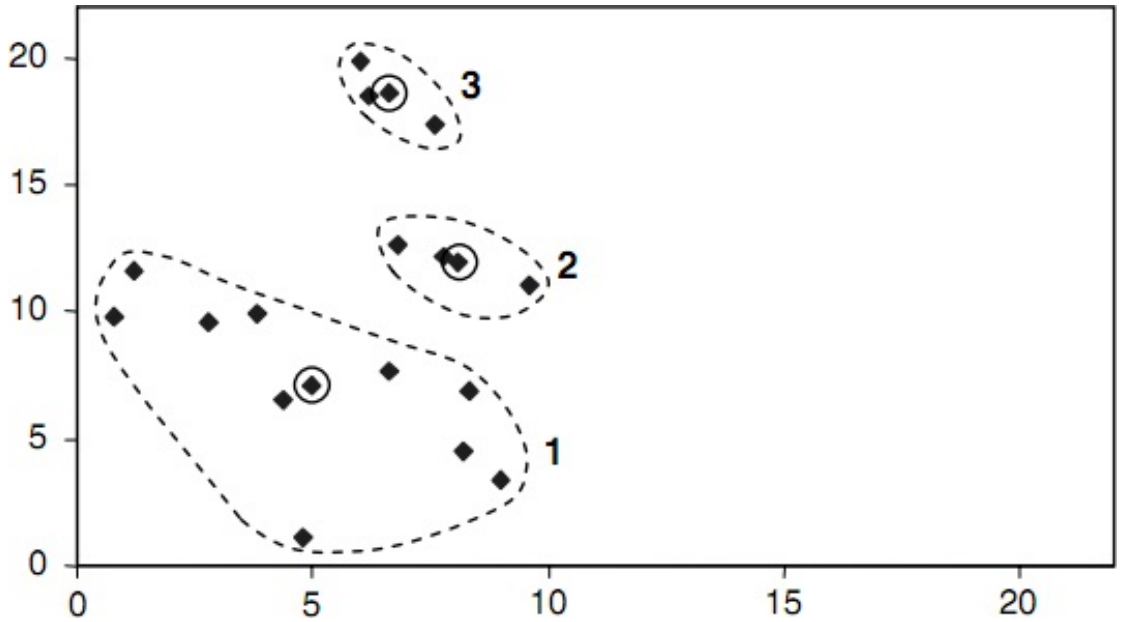


Comparación de las dos iteraciones:



... Algoritmo k-medias (Ejemplo)

	Initial		After first iteration		After second iteration	
	x	y	x	y	x	y
Centroid 1	3.8	9.9	4.6	7.1	5.0	7.1
Centroid 2	7.8	12.2	8.2	10.7	8.1	12.0
Centroid 3	6.2	18.5	6.6	18.6	6.6	18.6



Ventajas y desventajas

- Ventajas:
  - Casi siempre termina con un óptimo local.
- Desventajas:
  - Sólo es aplicable cuando la media está definida.
  - Necesita especificar la cantidad de clusters desde un principio.
  - El resultado depende de la selección inicial de centros.
  - Es sensible a datos con ruido y outliers.

Ejemplo en R con el dataset Iris

```
kmeans(x, centers, iter.max = 10)
```

- `x` = matriz de datos numéricos.
- `centers` = número de clusters requeridos.
- `iter.max` = número máximo de iteraciones permitidas.

Setosa

Versicolor

Virginica

K-MEDIAS CON EL DATASET IRIS

```
> dim(iris)
[1] 150 5

> names(iris)
[1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
    "Species"

> str(iris)
'data.frame':                    150 obs. of 5 variables:
$ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
$ Sepal.Width: num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
$ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
$ Petal.Width: num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
$ Species                   : Factor w/ 3 levels "setosa","versicolor
    1 1 1 1 1 1 1 ...
```

... k-medias con el dataset Iris

```
> summary(iris)
Sepal.Length  Sepal.Width   Petal.Length  Petal.Width   Species
Min.:4.300    Min.:2.000    Min.:1.000    Min.:0.100    setosa:50
1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300 versicolor:5
Median:5.800  Median:3.000  Median:4.350  Median:1.300  virginica:5
Mean:5.843    Mean:3.057    Mean:3.758    Mean:1.199
3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800
Max.:7.900    Max.:4.400    Max.:6.900    Max.:2.500
```

Eliminar la variable nominal

```
> iris2 <- iris
> iris2$Species <- NULL
```

... k-medias con el dataset Iris

Se aplica el algoritmo de k-medias.

```
> irisC <- kmeans(iris2, 3)
K-means clustering with 3 clusters of sizes 38, 50, 62
Cluster means:
    Sepal.Length      Sepal.Width     Petal.Length      Petal.Width
 1  6.850000          3.073684        5.742105          2.071053
 2  5.006000          3.428000        1.462000          0.246000
 3  5.901613          2.748387        4.393548          1.433871
Clustering vector:
  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [75] 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 1 1 1 3 3 3 3 3 3 3
[112] 1 1 3 3 1 1 1 1 3 1 3 1 3 1 1 3 3 1 1 1 1 1 3 1 1 1 1 3 1 1 1 3 3 3 3 3 3 3
[149] 1 3
Within cluster sum of squares by cluster:
[1] 23.87947 15.15100 39.82097
  (between_SS / total_SS = 88.4%)
```

... k-medias con el dataset Iris

Obtenemos información de las especies en cada cluster

```
> table(iris$Species, irisC$cluster)

             1  2  3
  setosa     0 50  0
  versicolor 2  0 48
  virginica 36  0 14
```

Con 4 clusters:

```
> irisC <-kmeans(iris2, 4)
K-means clustering with 4 clusters of sizes 45, 27, 28, 50
....
> table(iris$Species, irisC$cluster)
             1  2  3  4
  setosa     0  0  0 50
  versicolor 23  0 27  0
  virginica  22 27  1  0
```

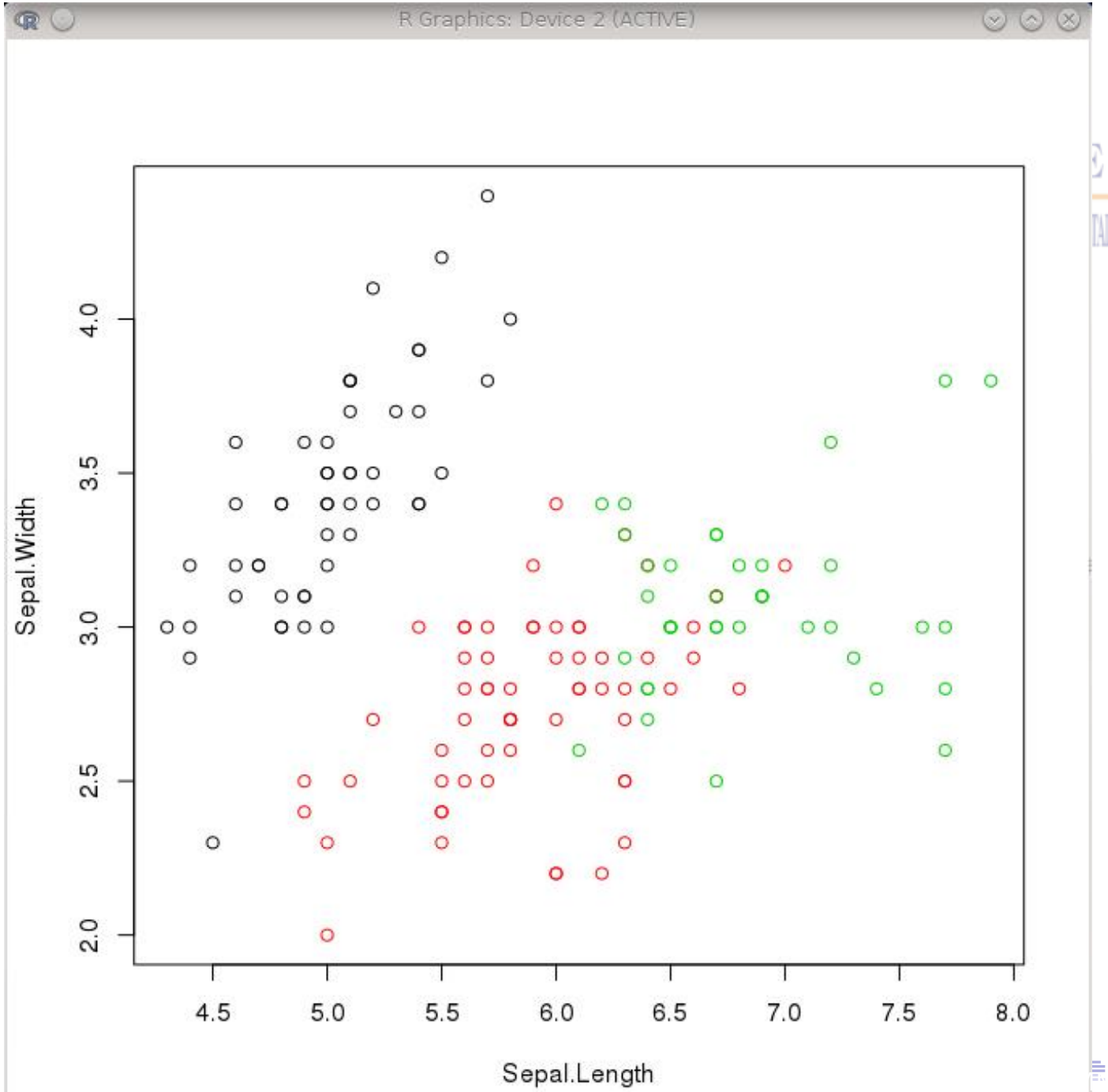
... k-medias con el dataset Iris

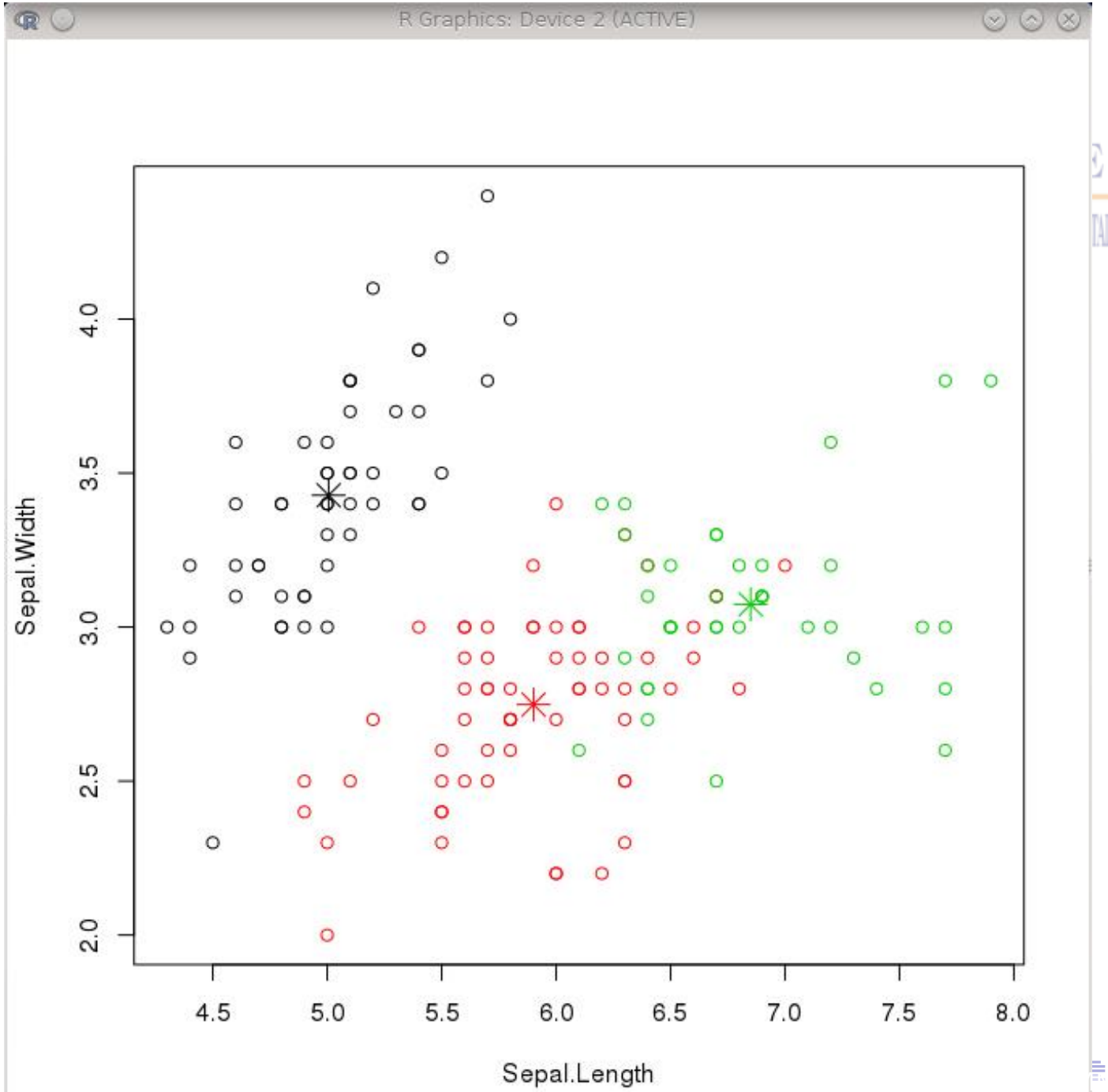
Con 2 clusters:

```
> irisC <-kmeans(iris2, 2)
K-means clustering with 2 clusters of sizes 53, 97
....

> table(iris$Species, irisC$cluster)
              1  2
  setosa     50  0
  versicolor  3 47
  virginica   0 50
```

... k-medias con el dataset Iris

```
> (irisC <- kmeans(iris2, 3))
K-means clustering with 3 clusters of sizes 38, 50, 62
...
> table(iris$Species, irisC$cluster)
             1  2  3
  setosa     0 50  0
  versicolor  2  0 48
  virginica  36  0 14

#
# Grafica los cluster
#
> plot(iris2[c("Sepal.Length","Sepal.Width")],col=irisC$cluster)

#
# Grafica los centros del cluster
#
> points(irisC$centers[,c("Sepal.Length", "Sepal.Width")], col=1:3,
```


R Graphics: Device 2 (ACTIVE)
Sepal.Width
Sepal.Length
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


R Graphics: Device 2 (ACTIVE)
Sepal.Width
Sepal.Length
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

Problema del método k-medias

- EL algoritmo k-medias es sensible a los outliers!!
  - Debido a que un objeto con valor extremadamente grande puede distorsionar substancialmente la distribución de los datos.
- Ejemplo. Los valores $\{1,2,3,8,9,10,25\}$
  - Intuitivamente vemos 1,2,3 en un cluster y 8,9,10 en otro y 25 como un outlier.
  - Con k = 2: $\{\{1,2,3\}, \{8,9,10,25\}\}$ con media 2 y 13 respectivamente SS = $(1-2)^2+(2-2)^2+(3-2)^2+(8-13)^2+(9-13)^2+(10-13)^2+(25-13)^2 = 196$
  - Si se tuviera: $\{\{1,2,3,8\}, \{9,10,25\}\}$ con media 3.5 y 14.67 SS = 189.67
    Como éste tiene menor variación dentro de los clusters asigna el 8 al primer cluster, pero el centro del segundo cluster está lejos de sus elementos.

Variaciones del método de las k-medias

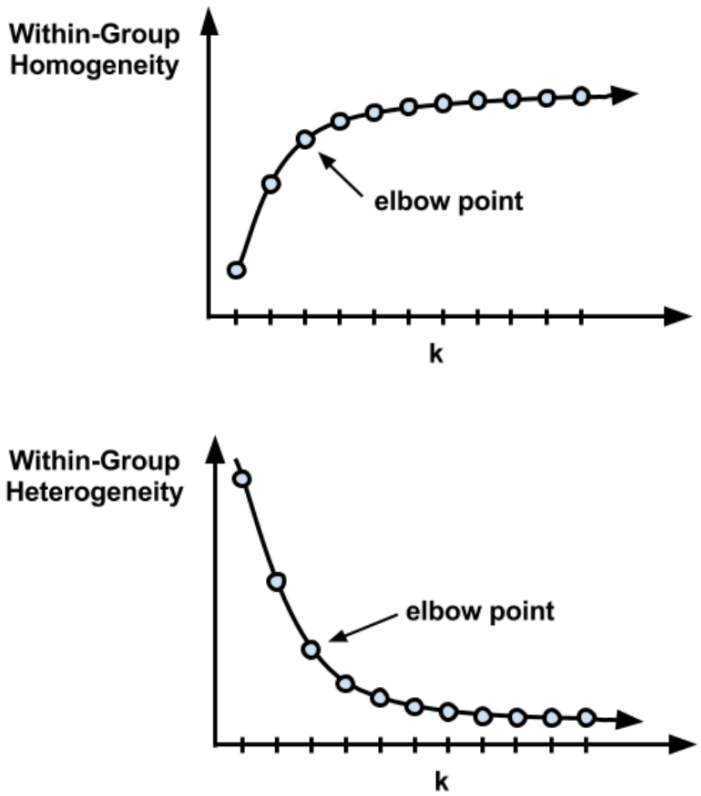
Variantes del método de k-medias, consideran:

- Selección de los centros.
  - k-medoides. En lugar de la media se toman medoides.
    Un medoide es el punto más cercano al centro en un cluster.
    - `pam(x, k, metric = "euclidean")`
    - `pamk(x)`, ésta se encuentra en la biblioteca `(fpc)`
- Cálculo de no similitudes.
- Estrategias para calcular las medias. PAM (Partition around medoids), CLARA, (Clustering Large Applications)
- Tratamiento de datos categóricos/nominales.
  - Reemplazar media por moda
  - Usar las medidas de similitud para datos categóricos.
    - Distancia de Hamming.
      Se define 0 si los dos puntos pertenecen a la misma categoría y 1 si no es el caso.

$$d(i,j) = (x_{i1} \neq x_{j1}) + (x_{i2} \neq x_{j2}) + \ldots + (x_{ip} \neq x_{jp})$$

Selección de la k

- Idealmente, se tiene un conocimiento a priori acerca de ciertos grupos (iris, películas, etc.)
- La k, puede ser determinada por las necesidades de la organización.
- Técnica del codo. Se intenta calibrar cómo la homogeneidad o heterogeneidad dentro de los clusters varía de acuerdo al valor de k.



Uso de PAM con Iris

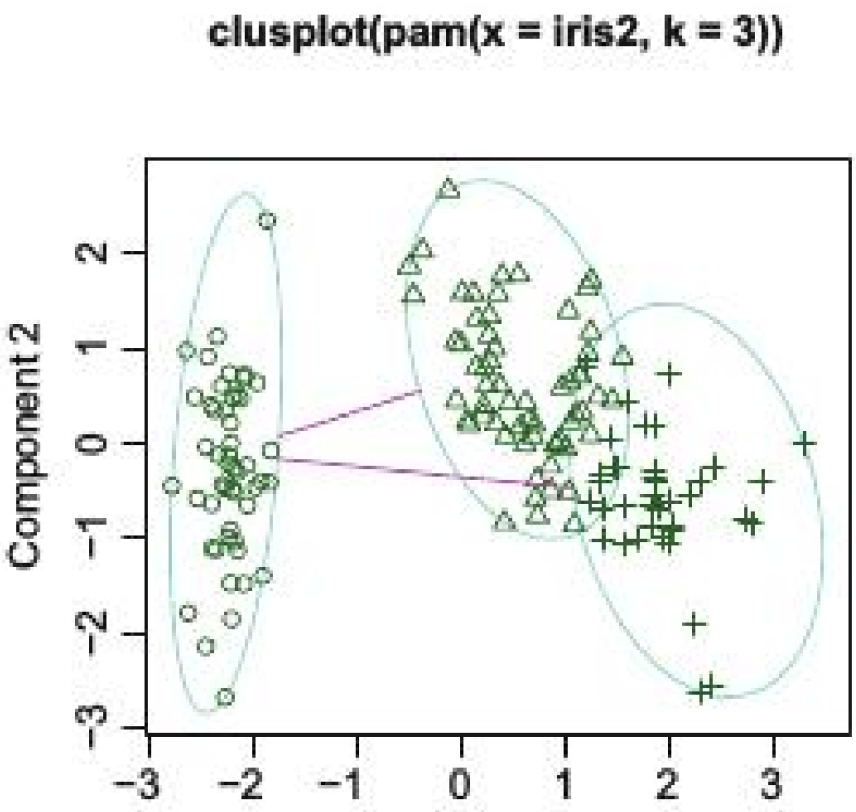
```
 > dim(iris)
[1] 150 5

> summary(iris)
Sepal.Length  Sepal.Width    Petal.Length   Petal.Width   Species
Min.:4.300    Min.:2.000     Min.:1.000     Min.:0.100    setosa:50
1st Qu.:5.100 1st Qu.:2.800  1st Qu.:1.600  1st Qu.:0.300 versicolo
Median:5.800  Median:3.000   Median:4.350   Median:1.300  virginica
Mean:5.843    Mean:3.057     Mean:3.758     Mean:1.199
3rd Qu.:6.400 3rd Qu.:3.300  3rd Qu.:5.100  3rd Qu.:1.800
Max.:7.900    Max.:4.400     Max.:6.900     Max.:2.500

> #Eliminar la variable nominal

> iris2 <- iris
> iris2$Species <- NULL
```

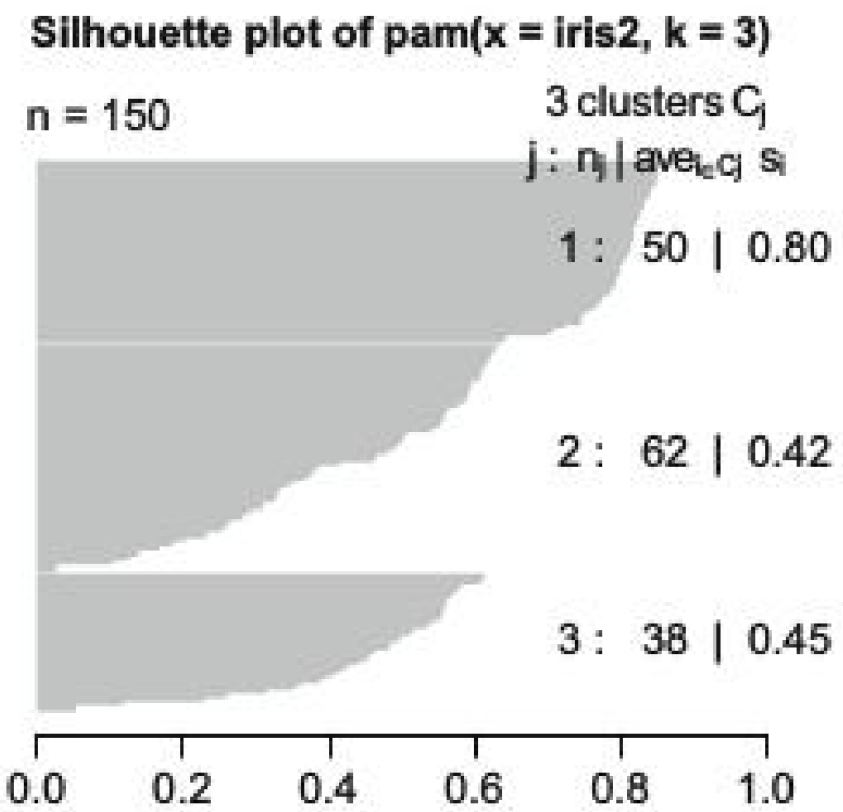
... Uso de PAM con Iris

```
> library(cluster)
> pam.result <- pam(iris2, 3)

> table(pam.result$clustering, iris$Species)

   setosa versicolor virginica
1      50          0         0
2       0         48        14
3       0          2        36

> pam.result$id.med
[1] 8 79 113

>iris2[8,]
   Sepal.Length Sepal.Width Petal.Length Petal.Width Specie
8             5         3.4          1.5         0.2  setosa
>iris2[79,]
79            6         2.9          4.5         1.5 versicolor
```

... Uso de PAM con Iris

```
> layout(matrix(c(1,2),1,2)) # 2 graficas por pag.
> plot(pam.result)
> layout(matrix(1))          # 1 grafica por pag.
```

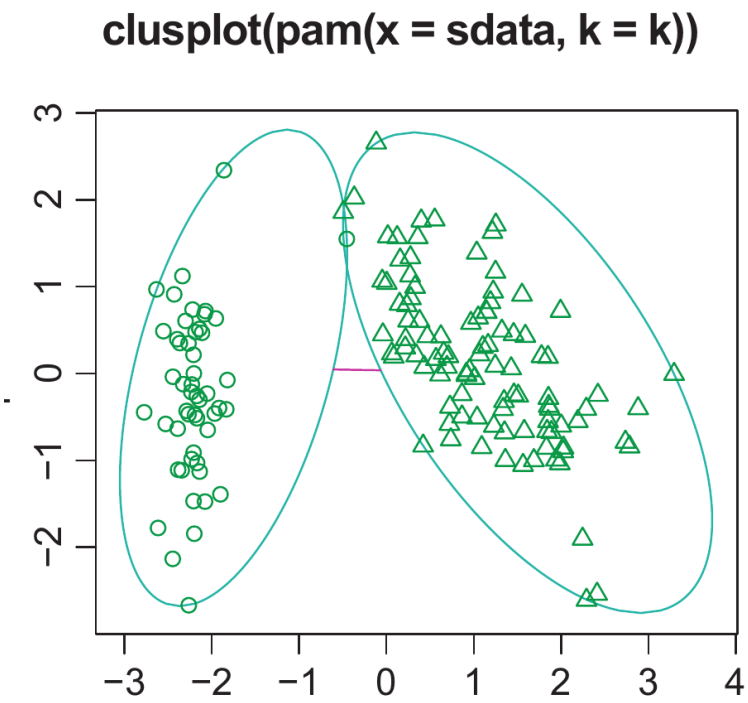


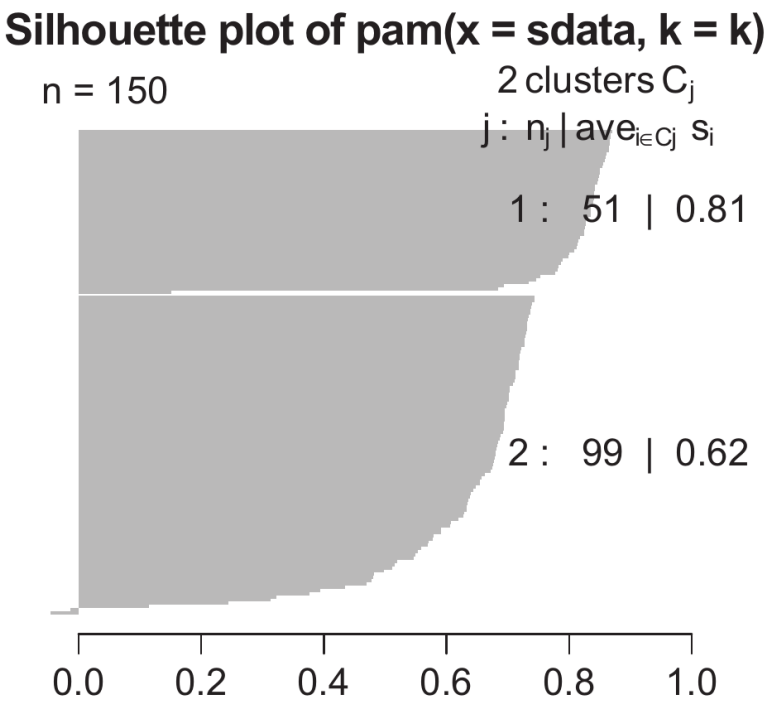


... Uso de PAMK con Iris

```
> library(fpc)
> pamk.result <- pamk(iris2)

> # cantidad de clusters
> pamk.result$nc
[1] 2

> table(pamk.result$clustering, iris$Species)

  setosa versicolor virginica
1     50          1         0
2      0         49        50

> plot(pamk.result$pamobject)
```

... Uso de PAMK con Iris





Ejemplo segmentación de mercado

- Los adolescentes que interactúan mediante redes sociales se han convertido en nicho de oportunidades para ventas.
- Han captado la atención de vendedores que luchan por encontrar un eje en cual incrementar su mercado competitivo.
- Una forma de lograrlo es identificar segmentos de adolescentes que compartan gustos similares, para que evitar llenarlos con publicidad que no sea de su interés. Ejemplo, camisetas de futbolistas puede ser difícil de vender a adolescentes sin interés en deportes.
-
- EL objetivo es encontrar 10 segmentos de mercado para adolescentes.

Colección y exploración de datos

- Colección: En un archivo llamado: `snsdata.csv` se tienen 30,000 registros de estudiantes de preparatoria registrados en un servicio de redes sociales en USA entre los años 2006 y 2009.
- Exploración:

```
> teens <- read.csv("snsdata.csv")

#Conocimiento de los datos
> str(teens)
'data.frame':  30000 obs. of 40 variables:
 $ gradyear    : int 2006 2006 2006 2006 2006 2006 2006 2006 ...
 $ gender      : Factor w/ 2 levels "F","M": 2 1 2 1 NA 1 1 2 ...
 $ age         : num 19 18.8 18.3 18.9 19 ...
 $ friends     : int 7 0 69 0 10 142 72 17 52 39 ...
 $ basketball  : int 0 0 0 0 0 0 0 0 0 0 ...
 $ football    : int  0 1 1 0 0 0 0 0 0 0 ...
 $ soccer      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ softball    : int  0 0 0 0 0 0 0 1 0 0 ...
 $ volleyball  : int  0 0 0 0 0 0 0 0 0 0
```

... Segmentación estudiantes

```
> summary(teens)
    gradyear       gender           age            friends      
 Min.   :2006   F   :22054   Min.   :  3.086   Min.   :  0.00  
 1st Qu.:2007   M   : 5222   1st Qu.: 16.312   1st Qu.:  3.00  
 Median :2008   NA's: 2724   Median : 17.287   Median : 20.00  
 Mean   :2008                Mean   : 17.994   Mean   : 30.18  
 3rd Qu.:2008                3rd Qu.: 18.259   3rd Qu.: 44.00  
 Max.   :2009                Max.   :106.927   Max.   :830.00  
                             NA's   :5086                      
   basketball         football          soccer          softball      
 Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.00  
 1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.00  
 Median : 0.0000   Median : 0.0000   Median : 0.0000   Median : 0.00  
 Mean   : 0.2673   Mean   : 0.2523   Mean   : 0.2228   Mean   : 0.16  
 3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.0000   3rd Qu.: 0.00  
 Max.   :24.0000   Max.   :15.0000   Max.   :27.0000   Max.   :17.00  

   volleyball        swimming       cheerleading       baseball      
```

... Segmentación estudiantes

```
#Cuantos adolescentes hay de cada sexo
> table(teens$gender)
    F     M
22054  5222

> table(teens$gender, useNA = "ifany")
    F     M  <NA>
22054  5222  2724
```

Veamos la edad

```
> summary(teens$age)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  NA's
3.086  16.310  17.290  17.990  18.260  106.900  5086

> teens$age <- ifelse(teens$age >= 13 & teens$age < 20,teens$age,NA)

> summary(teens$age)
   Min. 1st Qu. Median   Mean 3rd Qu.   Max.  NA's
  13.03  16.30   17.26  17.25  18.22  19.00  5523
```

Valores perdidos.

```
> teens$female <- ifelse(teens$gender == "F" &
                           !is.na(teens$gender), 1, 0)
> teens$no_gender <- ifelse(is.na(teens$gender), 1, 0)

> table(teens$gender, useNA = "ifany")
    F     M  <NA>
22054  5222  2724

> table(teens$female, useNA = "ifany")
    0     1
 7946 22054

> table(teens$no_gender, useNA = "ifany")
    0     1
27276  2724
```

... Segmentación (Imputación de v. perdidos)

```
 > mean(teens$age)
[1] NA

 > mean(teens$age, na.rm = TRUE)
[1] 17.25243

> aggregate(data = teens, age ~ gradyear, mean, na.rm = TRUE)
  gradyear       age
1     2006  18.65586
2     2007  17.70617
3     2008  16.76770
4     2009  15.81957

> ave_age <- ave(teens$age, teens$gradyear, FUN =
                   function(x) mean(x, na.rm = TRUE))

> teens$age <- ifelse(is.na(teens$age), ave_age, teens$age)

> summary(teens$age)
   Min. 1st Qu. Median   Mean 3rd Qu.    Max.
  13.03   16.28  17.24  17.24   18.21   20.00
```

... Segmentación estudiantes

Modelo de entrenamiento:

```
library(stats)
```

```
miCluster <- kmeans(datos, k)
```

Resultados:

- miCluster$cluster es un vector con la asignación de cada registro a su correspondiente cluster.
- miCluster$center es una matriz con centros.
- miCluster$size cantidad de registros asignados a cada cluster.

... Segmentación estudiantes

```
#Normalización

> interests_z <- as.data.frame(lapply(interests, scale))

>  teen_clusters <- kmeans(interests_z, 5)

> teen_clusters$size
[1] 3376 601 1036 3279 21708
```

... Segmentación estudiantes

```
> teen_clusters$centers
   basketball    football      soccer    softball
1  0.02447191  0.10550409  0.04357739 -0.02411100
2 -0.09442631  0.06927662 -0.09956009 -0.04697009
3  0.37669577  0.38401287  0.14650286  0.15136541
4  1.12232737  1.03625113  0.53915320  0.87051183
5 -0.18869703 -0.19317864 -0.09245172 -0.13366478

   volleyball    swimming cheerleading    baseball
1  0.04803724  0.31298181   0.63868578 -0.03875155
2 -0.07806216  0.04578401  -0.10703701 -0.11182941
3  0.09157715  0.24413955   0.18678448  0.28545186
4  0.78664128  0.11992750   0.01325191  0.86858544
5 -0.12850235 -0.07970857  -0.10728007 -0.13570044
```

... Segmentación estudiantes

Analizando los clusters se puede crear una tabla similar a la siguiente:

Cluster 1 (N = 3,376)	Cluster 2 (N = 601)	Cluster 3 (N = 1,036)	Cluster 4 (N = 3,279)	Cluster 5 (N = 21,708)
swimming cheerleading cute sexy hot dance dress hair mall hollister abercrombie shopping clothes	band marching music rock	sports sex sexy hot kissed dance music band die death drunk drugs	basketball football soccer softball volleyball baseball sports god church Jesus bible	???
Princesses	Brains	Criminals	Athletes	Basket Cases

Mejorando el modelo

```
> teens$cluster <- teen_clusters$cluster

> teens[1:5, c("cluster", "gender", "age", "friends")]
  cluster gender     age friends
1       5      M  18.982       7
2       1      F  18.801       0
3       5      M  18.335      69
4       5      F  18.875       0
5       3   <NA>  18.995      10

> aggregate(data = teens, age ~ cluster, mean)
  cluster      age
1       1 16.99678
2       2 17.38765
3       3 17.10022
4       4 17.09634
5       5 17.29841
```

... Segmentación estudiantes

Asignamos a cada cluster el promedio de edades, aunque no se usó para hacer los grupos.

```
> aggregate(data = teens, female ~ cluster, mean)
  cluster     female
1       1  0.8942536
2       2  0.7221298
3       3  0.8001931
4       4  0.7130223
5       5  0.7109821
```

... Segmentación estudiantes

Trataremos de determinar algo acerca de la cantidad de amigos.

```
> aggregate(data = teens, friends ~ cluster, mean)
  cluster   friends
1       1  38.74733
2       2  32.88186
3       3  30.57046
4       4  36.14029
5       5  27.85314
```

Resumen

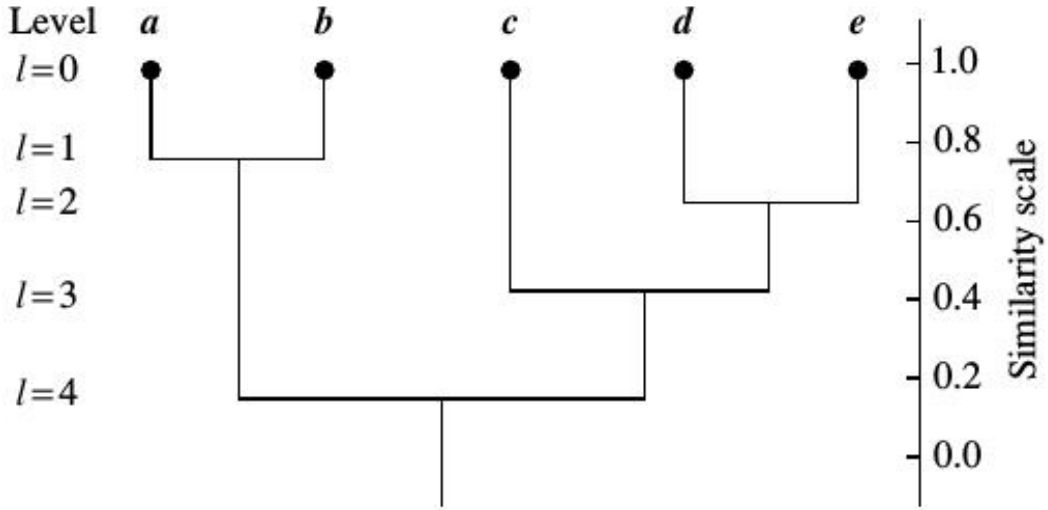
- Se empieza por datos no-etiquetados y por medio de clustering se crean etiquetas para clases.
- A partir de aquí se puede aplicar aprendizaje supervisado para encontrar los predictores más importantes de estas clases.
- Esto se conoce como aprendizaje semi-supervisado.
- En el ejemplo, lo que se mostró es que los clusters pueden ser predictores útiles.

Métodos jerárquicos

- Producen un conjunto de clusters anidados organizados como un árbol jerárquico.
- Útil para resumir y visualizar datos. Ejemplos:
  - Reconocimiento de manuscritos. Grupos para cada caracter y luego subgrupos por la forma de escritura.
  - Estudio de la evolución, se pueden agrupar los animales de acuerdo a sus características biológicas y descubrir trayectorias de evolución, las cuales son una jerarquía de especies.
  - Agrupar configuraciones de un juego de estrategia en forma jerárquica puede ayudar a desarrollar estrategias de juego que pueden usarse para entrenar a jugadores.
- Utilizan la matriz de distancias.
- No requieren especificación del número de clusters como entrada.

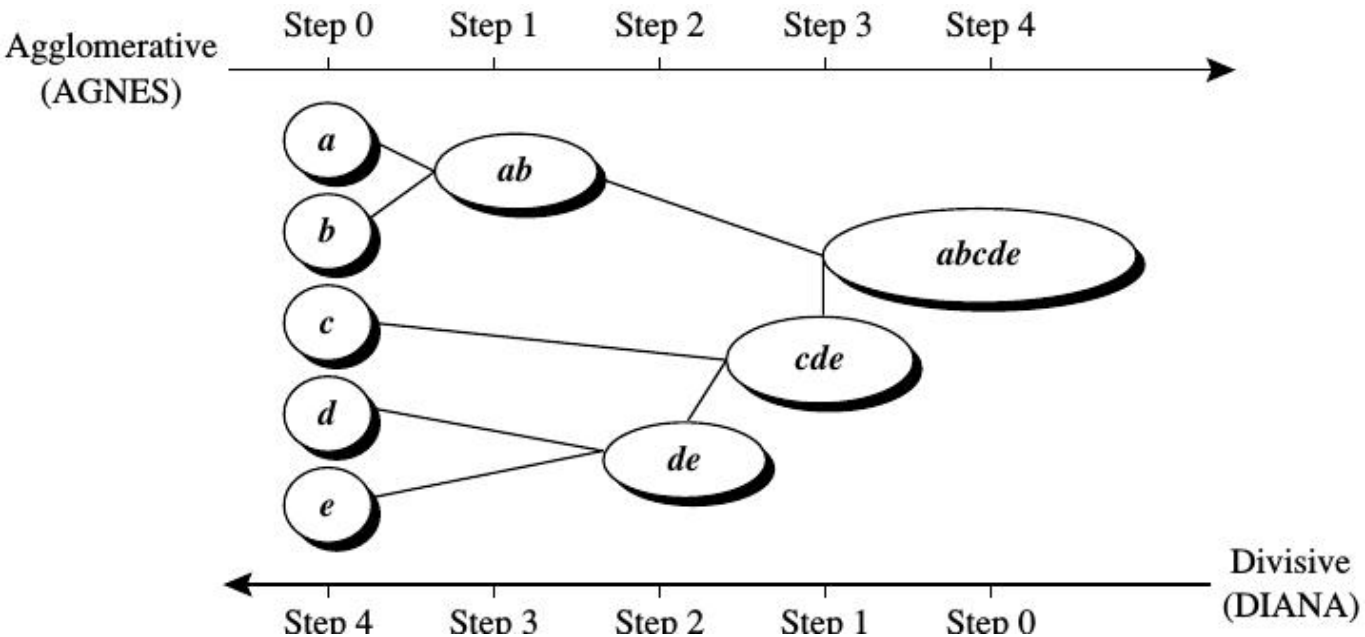
... Métodos jerárquicos

- Pueden visualizarse como un dendrograma:
  - Un diagrama como árbol que registra la secuencias de divisiones/agrupamientos.



- Fortalezas:
  - No se necesita especificar una cantidad de clusters
    - Cualquier número de clusters puede obtenerse cortando el dendrograma a nivel apropiado.
  - Pueden corresponder a taxonomías con significado.

... Métodos jerárquicos (Enfoques)



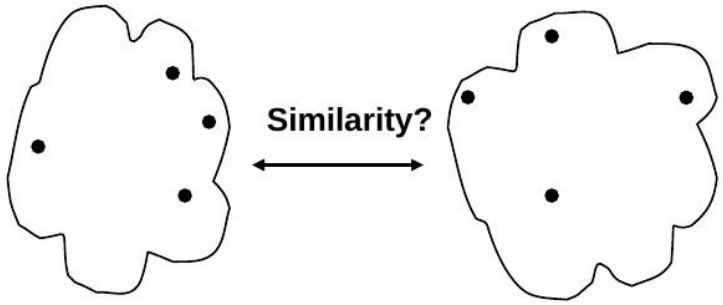
... Métodos jerárquicos (Enfoques)

- Dos principales enfoques:
  - Aglomerativo:
    - Empieza con los puntos como clusters individuales.
    - En cada paso, junta pares cercanos de clusters hasta tener un sólo cluster.
    - AGNES (AGglomerative NESting)
  - Divisivo:
    - Empieza con un cluster que contiene todo.
    - En cada paso, divide un cluster hasta que cada uno contiene un punto (o hay k clusters).
    - DIANA (DIvisive ANAlysis)
- Tradicionalmente usan una matriz de distancia o similitud.
  - Dividen o unen un cluster a la vez.

Algoritmo de clustering aglomerativo

- Es la técnica más popular para clustering jerárquico.
- El algoritmo básico es sencillo:
  1. Calcular la matriz de proximidad.
  2. Cada punto será un cluster.
  3. Repetir hasta que sólo hay un cluster:
     1. Unir los dos clusters más cercanos.
     2. Actualizar la matriz de proximidad.
- La operación clave es el cálculo de la proximidad de dos clusters:
  - Diferentes enfoques para definir la distancia entre cluster es lo que distingue a los diferentes algoritmos.

Medidas de similitud entre clusters



- Distancia mínima: Basado en los dos elementos, en cluster diferentes, más parecidos.
  - Algoritmo de clustering del vecino cercano. Cluster de una liga/enlace (`single`).
  - $dist_{min}(C_i, C_j) = min_{p \in C_i, p' \in C_j}\{|p - p'|\}$
    Con $|p - p'|$ la distancia entre dos objetos: p y p'
- Distancia máxima: Se basa en los dos elementos, en distintos clusters, que son menos parecidos.
  - Algoritmo de clustering del vecino lejano. Cluster de ligado completo (`complete`).
  - $dist_{max}(C_i, C_j) = max_{p \in C_i, p' \in C_j}\{|p - p'|\}$

- Distancia promedio: Promedio de proximidad por parejas entre puntos en los dos clusters.

$$dist_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{p \in C_i, p' \in C_j} |p - p'|$$

- Centroide: Distancia entre los centroides de dos clusters.
- Medoide: Distancia entre los medoides de dos clusters.
- Ward: Minimiza el error al cuadrado, interno, cuando dos cluster se integran.
  - Similar al promedio de grupo si la distancia entre los puntos es la distancia al cuadrado.
  - Es menos susceptible al ruido y atípicos.

Requerimientos de tiempo y espacio

Necesita:

- $O(N^2)$ espacio para la matriz de proximidad. Con N = número de puntos.
- Complejidad $O(N^3)$ en muchos casos:
  - Hay N pasos y cada uno es de tamaño N^2, la matriz de proximidad debe buscar y actualizarse.
  - La complejidad puede reducirse a $O(N^2 log(N))$ para algunas implementaciones.

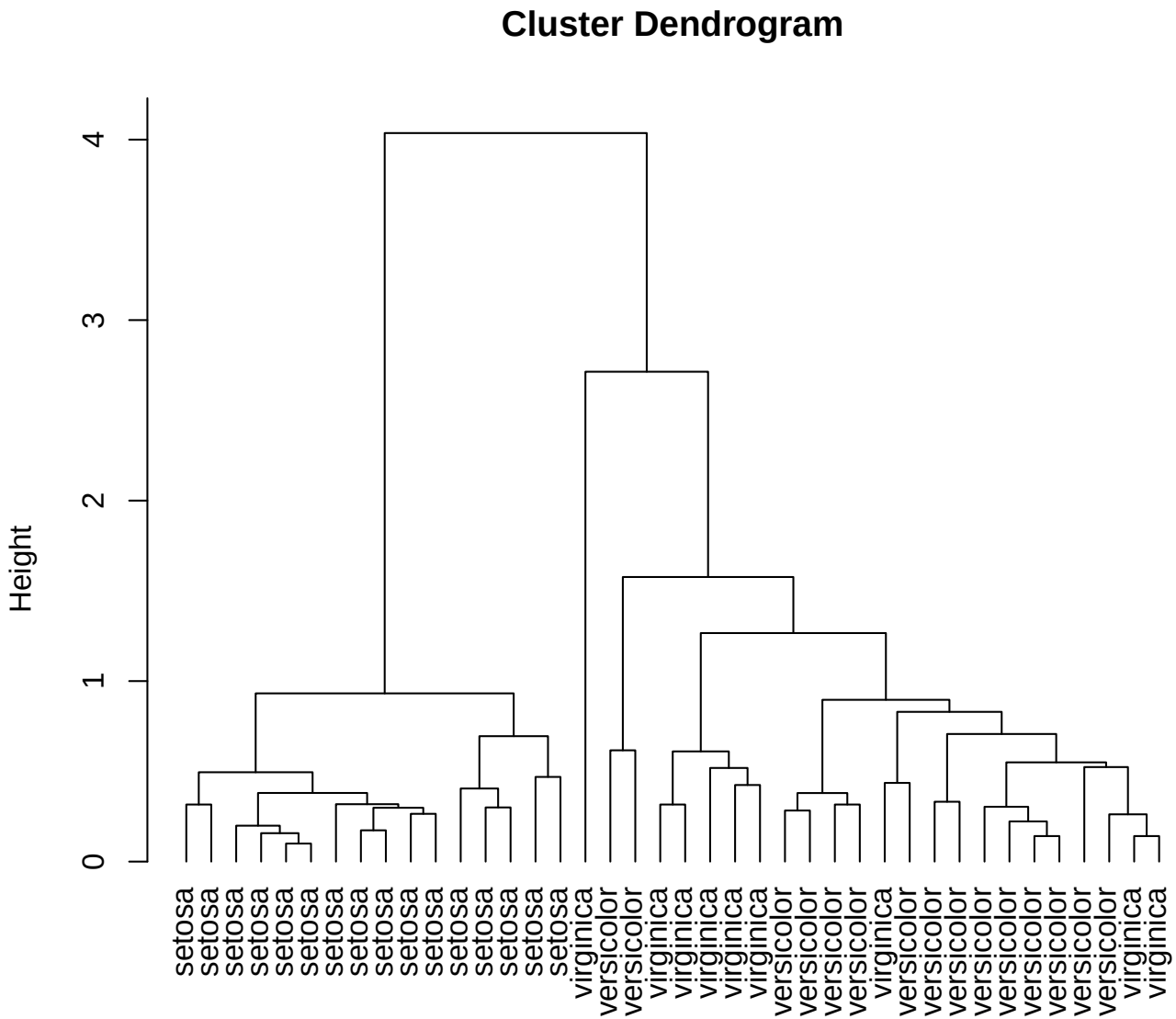
Cluster jerárquico en R

- `hclust(d, method = çomplete")` Con:
  - d = matriz de no-similitud probablemente creada con la función `dist`
  - método de aglomeración usado (`single, complete, average, ...`)
- `plot(x, labels = NULL, hang = 0.1)`

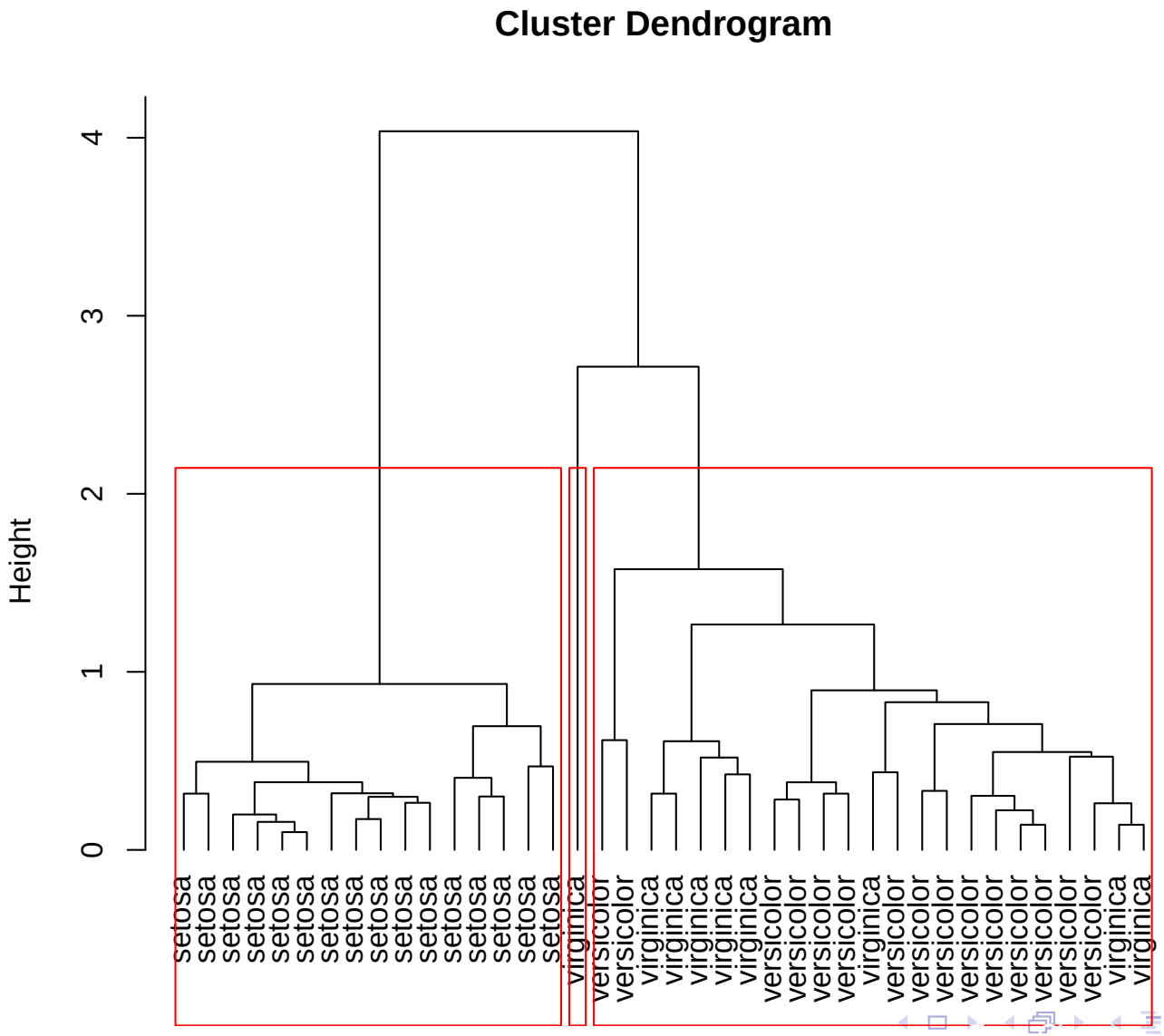
Ejemplo Iris

```
#Se toman 40 registros de iris para trabajar
> idx <- sample(1:dim(iris)[1], 40)
> iris.hc <- iris[idx,]
> iris.hc$Species <- NULL

#Realiza el clustering jerarquico y grafica

> hc <- hclust(dist(iris.hc), method="ave")
> plot(hc, labels=iris$Species[idx])

> rect.hclust(hc, k=3)
> grupos <- cutree(hc, k=3)
> grupos
```

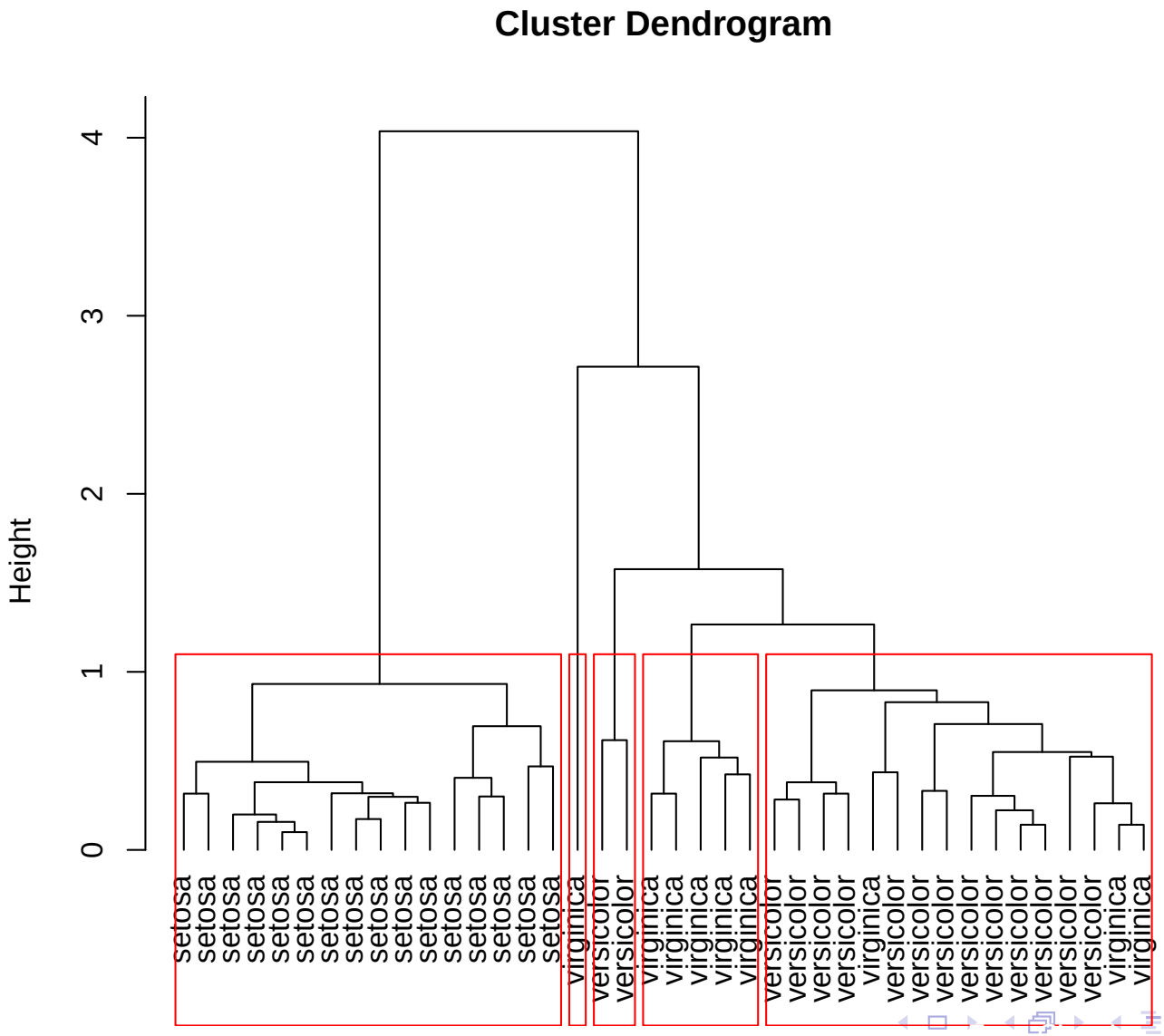
... Ejemplo Iris



... Ejemplo Iris



... Ejemplo Iris



Ejemplo Proteínas

```
> comida <- read.csv("protein.csv")

> str(comida)
'data.frame': 25 obs. of  10 variables:
 $ Country  : Factor w/ 25 levels "Albania","Austria",..: 1 2 3 4 5
 $ RedMeat  : num  10.1 8.9 13.5 7.8 9.7 10.6 8.4 9.5 18 10.2 ...
 $ WhiteMeat: num  1.4 14 9.3 6 11.4 10.8 11.6 4.9 9.9 3 ...
 $ Eggs     : num  0.5 4.3 4.1 1.6 2.8 3.7 3.7 2.7 3.3 2.8 ...
 $ Milk     : num  8.9 19.9 17.5 8.3 12.5 25 11.1 33.7 19.5 17.6 ..
 $ Fish     : num  0.2 2.1 4.5 1.2 2 9.9 5.4 5.8 5.7 5.9 ...
 $ Cereals  : num  42.3 28 26.6 56.7 34.3 21.9 24.6 26.3 28.1 41.7
 $ Starch   : num  0.6 3.6 5.7 1.1 5 4.8 6.5 5.1 4.8 2.2 ...
 $ Nuts     : num  5.5 1.3 2.1 3.7 1.1 0.7 0.8 1 2.4 7.8 ...
 $ Fr.Veg   : num  1.7 4.3 4 4.2 4 2.4 3.6 1.4 6.5 6.5 ...
```

Ejemplo Proteínas

```
> head(comida)

          Country RedMeat WhiteMeat Eggs Milk Fish Cereals Starch Nu
1         Albania    10.1       1.4  0.5  8.9  0.2    42.3    0.6  5
2         Austria     8.9      14.0  4.3 19.9  2.1    28.0    3.6  1
3         Belgium    13.5       9.3  4.1 17.5  4.5    26.6    5.7  2
4        Bulgaria     7.8       6.0  1.6  8.3  1.2    56.7    1.1  3
5 Czechoslovakia     9.7      11.4  2.8 12.5  2.0    34.3    5.0  1
6         Denmark    10.6      10.8  3.7 25.0  9.9    21.9    4.8  0
```

Ejemplo Proteínas

```
> summary(comida)
          Country      RedMeat        WhiteMeat         Eggs
 Albania       : 1   Min.   : 4.400   Min.   : 1.400   Min.   :0.50
 Austria       : 1   1st Qu.: 7.800   1st Qu.: 4.900   1st Qu.:2.70
 Belgium       : 1   Median : 9.500   Median : 7.800   Median :2.90
 Bulgaria      : 1   Mean   : 9.828   Mean   : 7.896   Mean   :2.93
 Czechoslovakia: 1   3rd Qu.:10.600   3rd Qu.:10.800   3rd Qu.:3.70
 Denmark       : 1   Max.   :18.000   Max.   :14.000   Max.   :4.70
 (Other)       :19
      Milk            Fish            Cereals          Starch
 Min.   : 4.90   Min.   : 0.200   Min.   :18.60   Min.   :0.600
 1st Qu.:11.10   1st Qu.: 2.100   1st Qu.:24.30   1st Qu.:3.100
 Median :17.60   Median : 3.400   Median :28.00   Median :4.700
 Mean   :17.11   Mean   : 4.284   Mean   :32.25   Mean   :4.276
 3rd Qu.:23.30   3rd Qu.: 5.800   3rd Qu.:40.10   3rd Qu.:5.700
 Max.   :33.70   Max.   :14.200   Max.   :56.70   Max.   :6.500
```

Ejemplo Proteínas

```
     Nuts             Fr.Veg
Min.   :0.700   Min.   :1.400
1st Qu.:1.500   1st Qu.:2.900
Median :2.400   Median :3.800
Mean   :3.072   Mean   :4.136
3rd Qu.:4.700   3rd Qu.:4.900
Max.   :7.800   Max.   :7.900
```

Ejemplo Proteínas

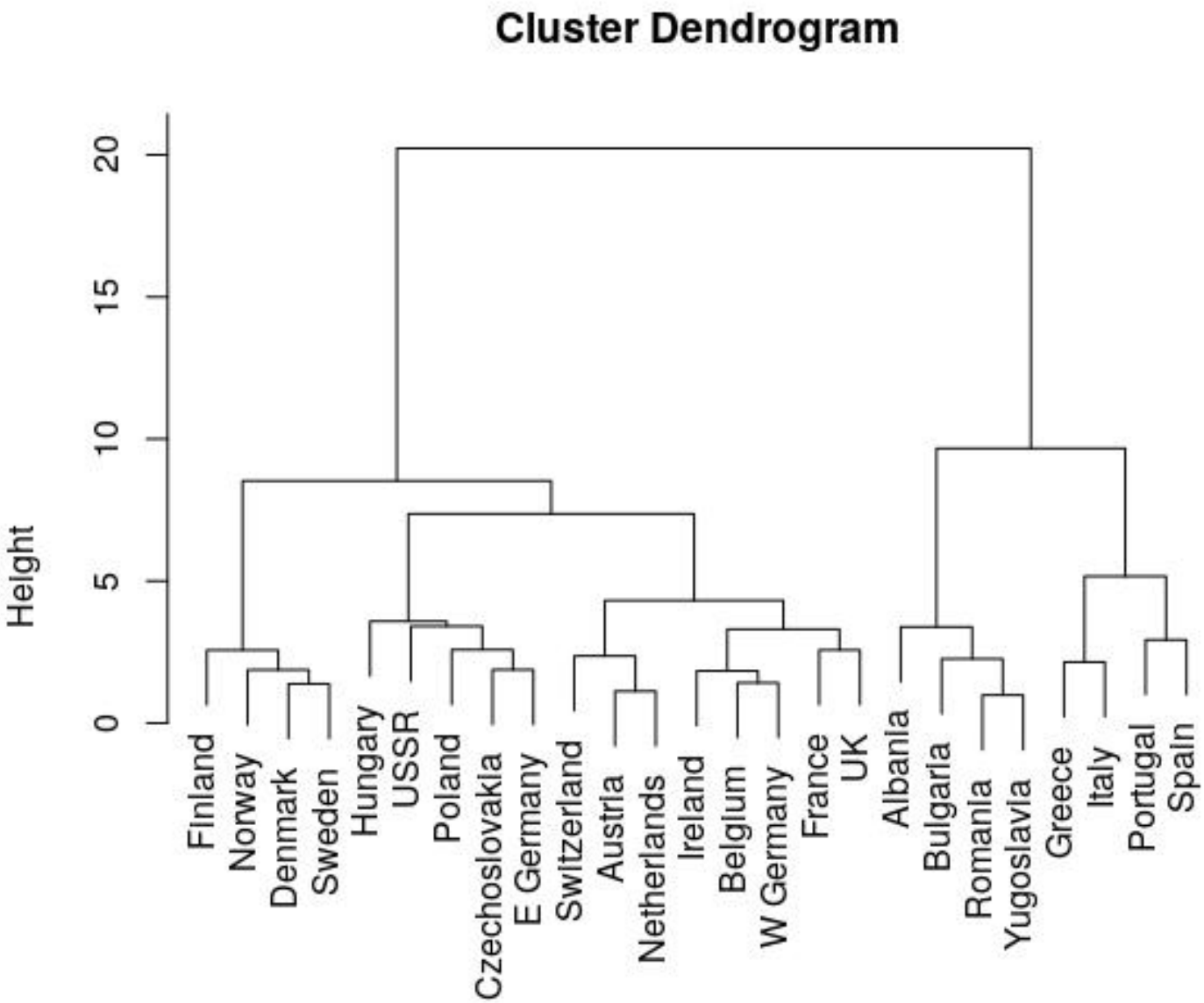
Preparación de los datos.

```
vars.usadas <- colnames(comida)[-1]  #Elimino el país
pmatrix <-scale(comida[,vars.usadas])
```

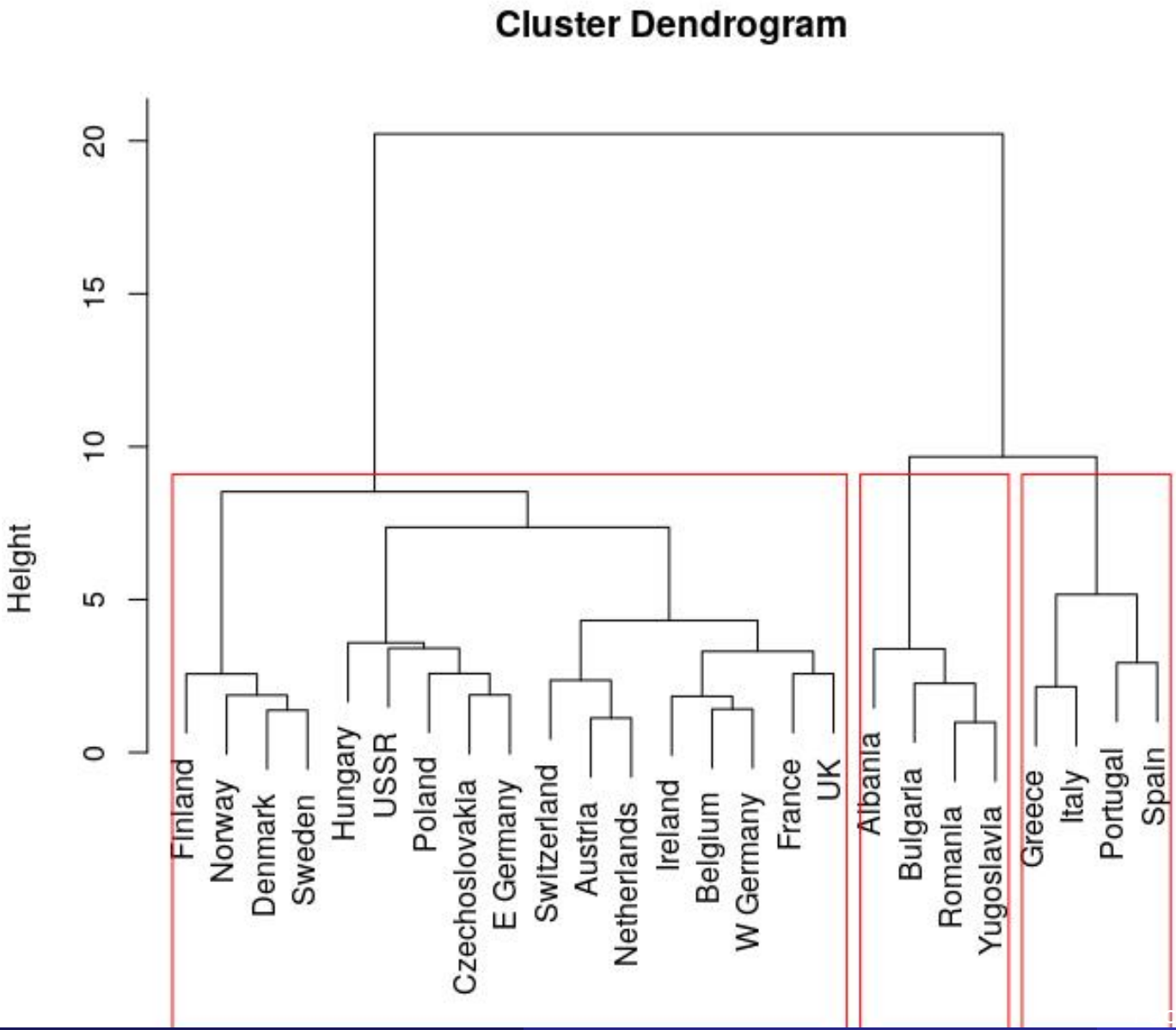
Generación de modelo.

```
> d <-dist(pmatrix, method="euclidean")
> pfit <- hclust(d, method="ward")
> plot(pfit, labels=comida$Country)
> rect.hclust(pfit, k=3)
> rect.hclust(pfit, k=5)
```

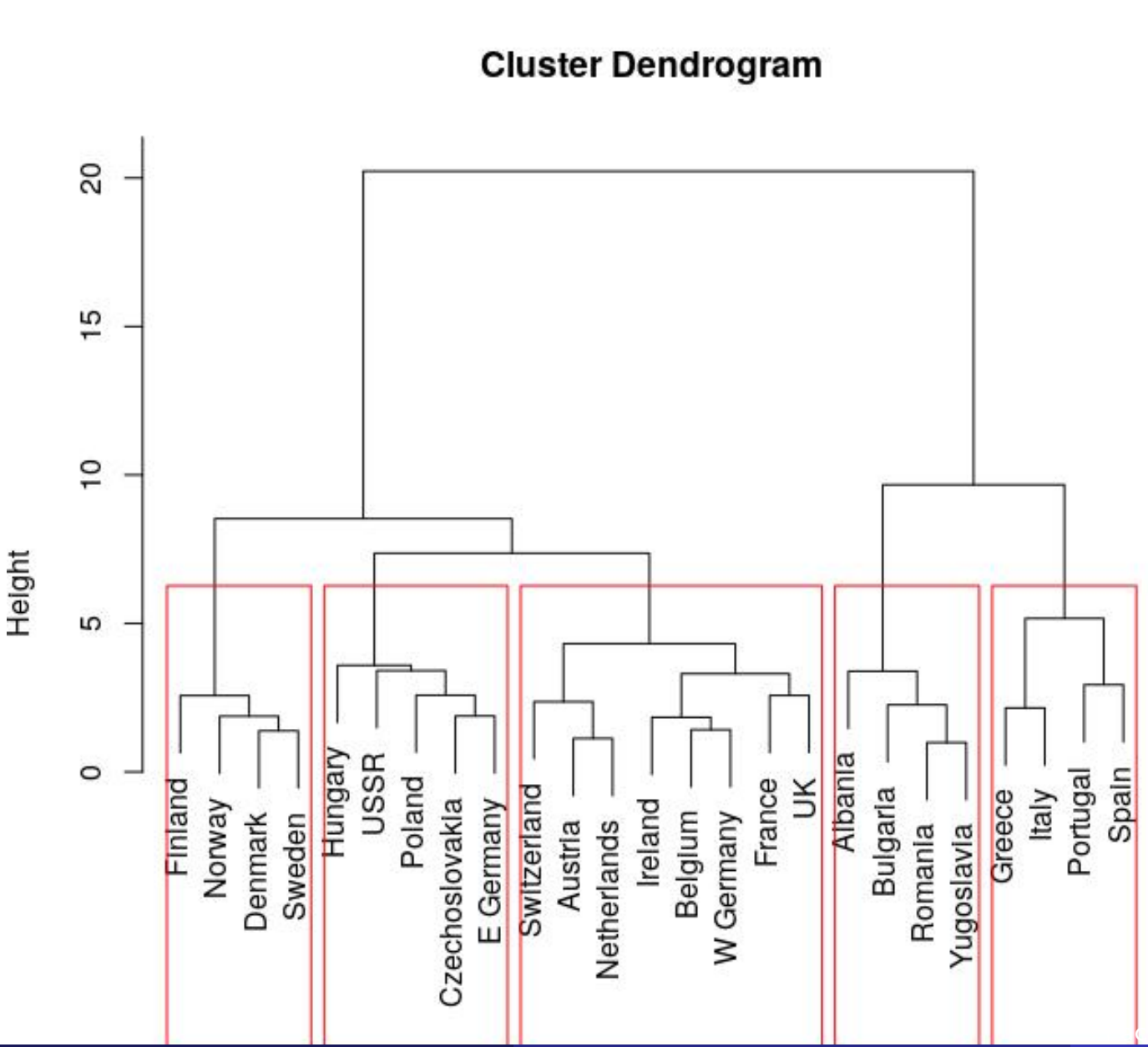
Ejemplo Proteínas



Ejemplo Proteínas



Ejemplo Proteínas



Ejemplo Proteínas

- Uso de la función `cutree` para obtener los miembros de cada cluster.
- Sintaxis: `cutree(tree, k)`

```
> grupos <-cutree(pfit, k=5)
>
> imprime_clusters <- function(labels, k) {
+   for(i in 1:k) {
+     print(paste("cluster", i))
+     print(comida[labels== i, c("Country","RedMeat","Fish","Fr.Veg
+   }
+ }
>
> imprime_clusters(grupos,5)
```

Ejemplo Proteínas

```
[1] "cluster 1"
      Country RedMeat Fish Fr.Veg
1     Albania    10.1  0.2    1.7
4    Bulgaria     7.8  1.2    4.2
18    Romania     6.2  1.0    2.8
25 Yugoslavia     4.4  0.6    3.2
[1] "cluster 2"
       Country RedMeat Fish Fr.Veg
2      Austria     8.9  2.1    4.3
3      Belgium    13.5  4.5    4.0
9       France    18.0  5.7    6.5
12     Ireland    13.9  2.2    2.9
14 Netherlands     9.5  2.5    3.7
21 Switzerland    13.1  2.3    4.9
22          UK    17.4  4.3    3.3
24   W Germany    11.4  3.4    3.8
```

Ejemplo Proteínas

```
[1] "cluster 3"
          Country RedMeat Fish Fr.Veg
5  Czechoslovakia     9.7  2.0    4.0
7       E Germany     8.4  5.4    3.6
11        Hungary     5.3  0.3    4.2
16         Poland     6.9  3.0    6.6
23           USSR     9.3  3.0    2.9
[1] "cluster 4"
   Country RedMeat Fish Fr.Veg
6  Denmark    10.6  9.9    2.4
8  Finland     9.5  5.8    1.4
15  Norway     9.4  9.7    2.7
20  Sweden     9.9  7.5    2.0
[1] "cluster 5"
10   Greece    10.2  5.9    6.5
13    Italy     9.0  3.4    6.7
17 Portugal     6.2 14.2    7.9
19    Spain     7.1  7.0    7.2
```