

A collection of approximately 18 squares in three shades of blue, grey, and light blue, arranged in a sparse, abstract pattern across the top half of the slide.

MUBD

Màster Universitari en Enginyeria de Dades Massives (Big Data)

Estadística



Índice – Clusterización jerárquica

1. Introducción
- 2. Clusterización jerárquica**
 1. Distancia
 2. Inercia
 3. Criterio de Ward
 4. Algoritmo aglomerativo
 5. Partición
- 3. K-means**
 1. Algoritmo
 2. Parámetros
 3. Convergencia
 4. Métodos/variantes
 5. Medidas de rendimiento
- 4. Sistemas mixtos**

Introducción

Objetivo

- Se desea formar grupos de observaciones similares entre sí y distintas entre grupos según unas características determinadas.
- Es un problema de clasificación de instancias.
- Hay distintas metodologías (algoritmos) de alcanzar este objetivo
 - Clusterización jerárquica
 - K-means
 - ...

Introducción

Aplicabilidad

- Biología
 - Agrupar organismos en especies
 - Agrupar en familias genéticas
- Medicina
 - Análisis de imágenes
- Marketing
 - Crear segmentos de consumidores
 - Clasificación de productos
- Sociología
 - Organizar comunidades a través de relaciones en las redes sociales
 - Identificar grupos de estudiantes dentro de una comunidad educativa
- Otros...

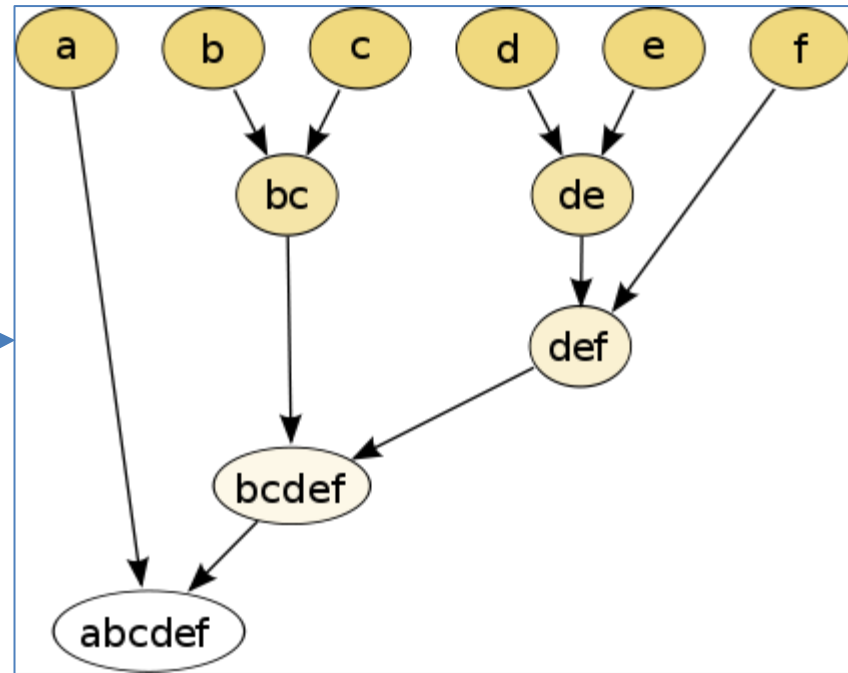
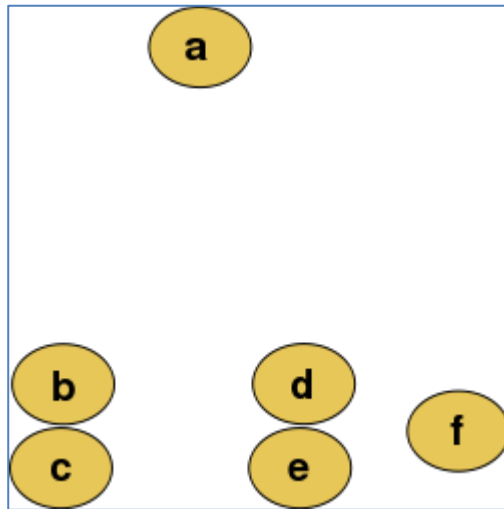
Introducción

Tipos de problemas

- No supervisado (Clustering analysis)
 - Sin variable respuesta
 - Evaluación de la calidad del algoritmo complicada
 - Ej: clusterización jerárquica, K-means
- Supervisado (Discriminant analysis)
 - Se dispone de una variable respuesta
 - Fácil evaluación de la capacidad predictiva
 - Ej: conditional trees, random forest, KNN, Naive-Bayes, SVM

Clusterización jerárquica

Ejemplo



	A	B	C	D	E	F
A						
B	6					
C	7	1				
D	6	3	4			
E	7	4	3	1		
F	8	5	5	2	2	



	A	B-C	D-E	F
A				
B-C	7			
D-E	7	4		
F	8	5	2	



	A	B-C	D-E-F
A			
B-C	7		
D-E-F	8	5	

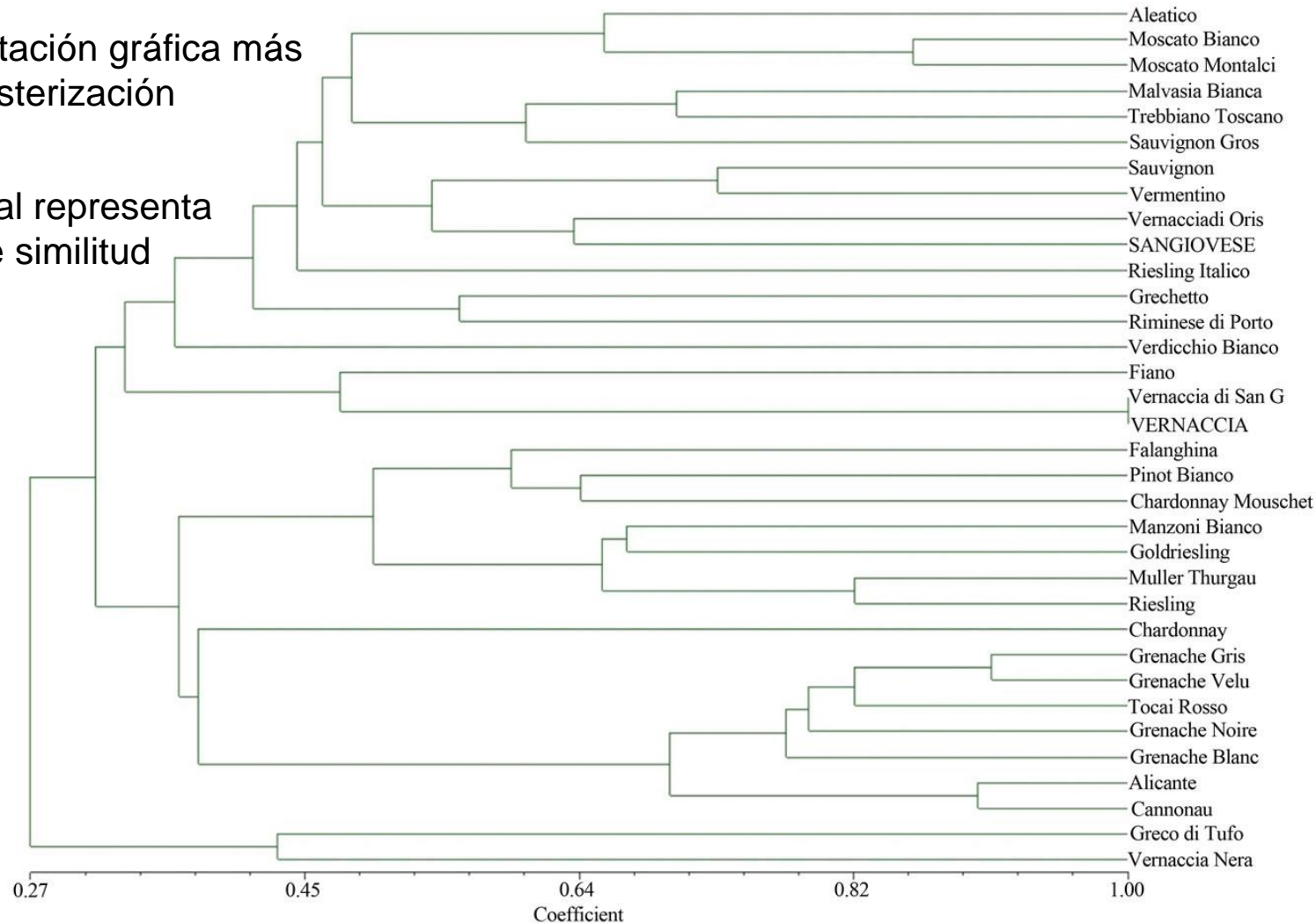


	A	B-C-D-E-F
A		
B-C-D-E-F	8	

Clusterización jerárquica

Dendograma

- Es la representación gráfica más usual de la clusterización jerárquica
- El eje horizontal representa una medida de similitud



Clusterización jerárquica

Introducción

■ Ventajas

- No es imprescindible tener variables cuantitativas (aunque es más usual) ya que se pueden definir distancias para variables categóricas
- No requiere definir el número de agrupaciones a priori. Se calcula un árbol jerárquico independientemente de los grupos
- El árbol permite visualizar de forma intuitiva distancias entre individuos para tamaños muestrales no muy grandes

■ Inconvenientes

- Requiere definir un tipo de distancia
- Requiere el cálculo de todos los pares de distancias. Mayor tiempo de computación para muestras grandes (>500)

Clusterización jerárquica

Proceso

Al realizar una clusterización jerárquica, se deben realizar los siguientes pasos (en azul, decisiones a tomar):

1. Decidir el tipo de distancia a emplear (euclídea, manhattan...)
2. Construir la matriz de disimilitudes
3. Decidir el método de agrupación (aglomerativo o divisivo)
4. Decidir el sub-método de agrupación (Ward, completo...)
5. Aplicar el algoritmo de clusterización
6. Decidir el criterio para realizar la partición de los clústeres
7. Particionar la muestra

NOTA: Con los pasos 1 a 5 construimos el dendograma

Distancias entre observaciones

Tipos

■ **Minkowski**

■ **Manhattan** ($p = 1$)

■ **Euclídea** ($p = 2$)

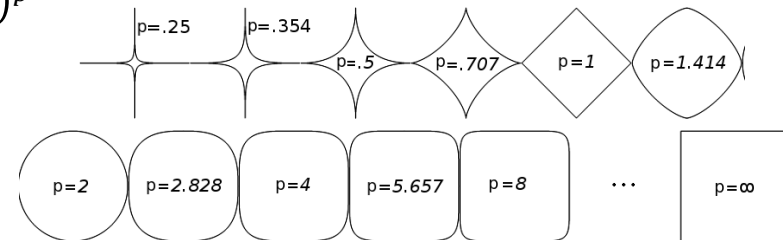
■ **Máxima** ($p = \infty$)

$$d(i, l) = \left(\sum_{k=1}^K |x_{ik} - x_{lk}|^p \right)^{\frac{1}{p}}$$

$$d(i, l) = \sum_{k=1}^K |x_{ik} - x_{lk}|$$

$$d^2(i, l) = \sum_{k=1}^K (x_{ik} - x_{lk})^2$$

$$d(i, l) = \max_k \|x_{ik} - x_{lk}\|$$



■ **Mahalanobis**

$$d^2(i, l) = (x_i - x_l)^T \Sigma^{-1} (x_i - x_l) \rightarrow \Sigma \text{ es la matriz de var-covar}$$

■ **Canberra**

$$d(i, l) = \sum_{k=1}^K \frac{|x_{ik} - x_{lk}|}{|x_{ik} + x_{lk}|}$$

■ **Gower**

$$d_k(i, l) = \sum_{k=1}^n d_k(i, l) \rightarrow d_k(i, l) = \begin{cases} 1 & \text{si } i_k \neq j_k \\ 0 & \text{si } i_k = j_k \end{cases} \text{ (variables categóricas)}$$

■ **Índice de Jaccard**

$$d(i, l) = \frac{n_{++}}{n_{++} + n_{+-} + n_{-+}} \text{ (variables categóricas)}$$

Distancias entre grupos de observaciones

Tipos

- **Simple.** Mínima distancia entre dos puntos pertenecientes uno a cada grupo
- **Completa.** Máxima distancia entre dos puntos pertenecientes uno a cada grupo
- **Entre centros de gravedad.** Distancia entre los centros de gravedad
- **Criterio de Ward.** No es una distancia en sí, sino un criterio para hacer la agrupación que consiste en minimizar la inercia.

Inercia

Definición

- La inercia es una medida de la heterogeneidad existente en un conjunto de datos.
- Dada una partición en Q grupos formados por I_q elementos en un espacio K-dimensional (K variables) se define:

Inercia Total $\sum_{k=1}^K \sum_{q=1}^Q \sum_{i=1}^{I_q} (y_{iqk} - \bar{y}_k)^2$ *[Distancia de cada punto al centro de gravedad]*

Inercia Entre-grupo $\sum_{k=1}^K \sum_{q=1}^Q I_q (\bar{y}_{qk} - \bar{y}_k)^2$ *[Distancia del centro de los grupos al centro global]*

Inercia Intra-grupo $\sum_{k=1}^K \sum_{q=1}^Q \sum_{i=1}^{I_q} (y_{iqk} - \bar{y}_{qk})^2$ *[Distancia de cada punto al centro de su grupo]*

- Se cumple que:

$$Inercia\ Total = Inercia\ Entre-grupo + Inercia\ Intra-grupo$$

Inercia

Propiedades

- La calidad de una partición puede medirse como el % de variabilidad explicada por los grupos (similar al R^2 del modelo lineal), es decir:

$$\text{Variabilidad explicada} = \frac{\text{Inercia entre - grupos}}{\text{Inercia total}}$$

- El porcentaje de variabilidad explicada siempre crece a medida que se incrementan el número de grupos:

Mayor número de grupos → **Más** variabilidad explicada → **Más** inercia entre-grupos

Menor número de grupos → **Menos** variabilidad explicada → **Menos** inercia entre-grupos

Método de agrupación

Aglomerativo vs. Divisivo

- El agrupamiento **aglomerativo** es el más común: AGNES (AGglomerative NESTing).
 - Funciona de una manera **ascendente**
 - El algoritmo comienza tratando cada objeto como un único clúster.
 - En cada iteración, los 2 clústeres más similares se combinan en un nuevo clúster (nodo). Este proceso se repite hasta que todos los puntos pertenezcan a un único gran grupo (raíz)
- El proceso inverso de la agrupación aglomerativa es la agrupación **divisiva**, también conocida como DIANA (Divisve ANAlysis)
 - Funciona de manera **descendente**
 - Comienza con la raíz, en la cual todos los objetos están incluidos en un solo grupo.
 - En cada iteración, el clúster más heterogéneo se divide en dos. El proceso se itera hasta que todos los objetos sean un único clúster.
- La agrupación aglomerativa es buena para identificar pequeños grupos y la divisiva sirve para identificar grandes agrupaciones.

Criterio de Ward

Agrupación

- Se parte de Q clústeres
- Se desea una agrupación ideal que pase a $Q-1$ clústeres
- Sea el clúster p (con centro de gravedad g_p y tamaño I_p) y el clúster q (con centro de gravedad g_q y tamaño I_q). El incremento en la inercia intra-grupo se cuantifica por:

$$\Delta(p, q) = \frac{I_p I_q}{I_p + I_q} d^2(g_p, g_q)$$

- El criterio de Ward consiste en juntar aquellos clústeres tales que minimicen el incremento en la inercia intra-grupo al pasar de Q a $(Q-1)$ clústeres.
- Este criterio favorece juntar:
 - Los clústeres cercanos
 - Los clústeres pequeños

Clusterización jerárquica

Algoritmo aglomerativo

1. Construir la matriz de distancias
2. Se escogen los dos puntos (o agrupaciones de puntos) más próximos o según algún criterio (p.ej, Ward)
3. Se representa la agrupación uniendo ambos puntos (o agrupaciones) a una altura equivalente a la distancia (o al cambio en la inercia).
4. Se actualiza la matriz de distancias agrupando las filas y las columnas correspondientes a los puntos agrupados y recalculando las distancias de todos los puntos al nuevo conglomerado.
5. Se vuelve al punto 2 mientras queden agrupaciones posibles

Calidad del árbol jerárquico

Distancia y correlación cofenética

- Para evaluar la calidad del árbol (dendograma) construido se deben comparar cuán similares son las distancias (alturas) producidas por el dendograma respecto a las distancias originales.
- Las distancias obtenidas del dendograma se denominan **distancias cofenéticas**.
- Para evaluar la calidad, se calcula la **correlación cofenética** entre estas distancias y las distancias originales.
- Si la agrupación es válida, debe existir una fuerte correlación entre estas distancias. Se considera que los valores superiores a 0.75 son aceptables.
- Escoger distancias “medias” para la agrupación produce valores altos de esta correlación.

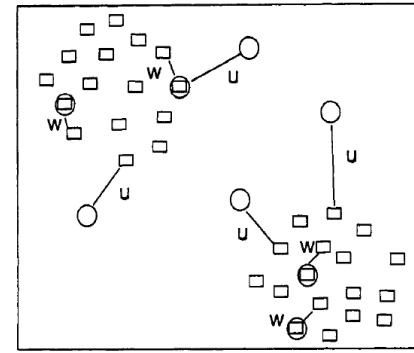
Clusterización jerárquica

Número de clústeres

- Una vez se tiene el dendograma, se debe definir una partición en K clústeres.
- Criterios para escoger una partición:
 - **Visual.** A partir del dendograma
 - **Cambio de inercia**
 - [Regla del codo.](#)
 - Minimizar el cociente $\Delta(q) / \Delta(q+1)$ donde $\Delta(q)$ es el cambio en la inercia intra al pasar de q a $q-1$ clústeres
 - **Parsimonia.** No es conveniente escoger un gran número de clústeres
 - **Interpretabilidad.** Los grupos deben tener algún sentido

Tendencia a la agrupación

Estadístico de Hopkins



- Escoger n puntos aleatorios $D'=(p_1, \dots, p_n)$ de nuestro conjunto de datos (D)
- Para cada punto $p_i \in D'$, buscar su vecino más próximo $p_j \in D$ y calcular su distancia (w_i)
- Generar un conjunto de datos simulado (R) con distribución uniforme con n puntos (q_1, \dots, q_n) y la misma variabilidad que el conjunto de datos D .
- Para cada punto $q_i \in R$, buscar su vecino más próximo dentro del conjunto real de puntos y calcular su distancia (u_i)
- El estadístico de *Hopkins* será:

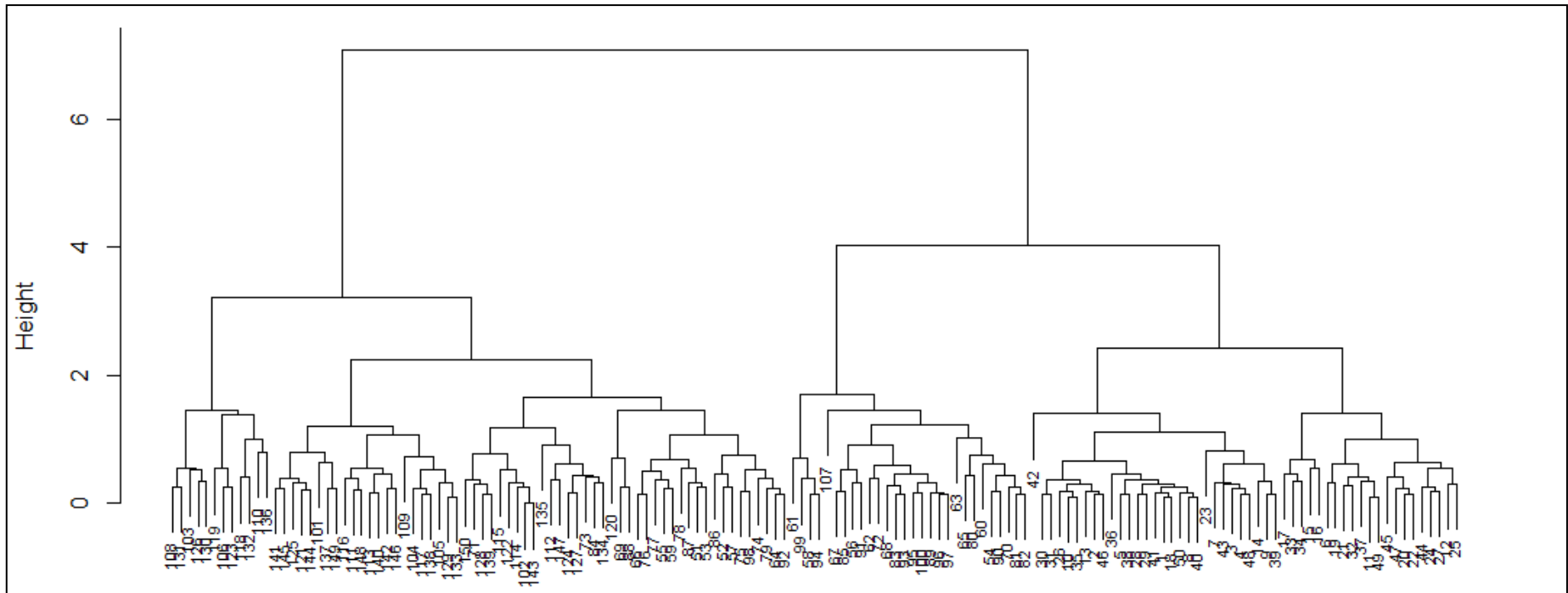
$$H = \frac{\sum u_i}{\sum w_i + \sum u_i}$$

- Un valor de 0.5 indicará que los clústeres son parecidos y que por tanto, no existen clústeres. Un valor próximo a 1 indicará presencia de clústeres.
- **Nota:** la función `hopkins{clustertend}` de R proporciona el estadístico $1-H$

Clusterización jerárquica

Ejemplo con R (iris)

```
d <- dist(iris2, method = "euclidean") # matriz de distancias
View(as.matrix(d))                     # ver matriz
hc <- hclust(d, method = "complete")   # jerarquización
windows(14, 7)                         # representar jerarquía
plot(hc, cex=0.5)
```



Ejemplo R (iris)

- Se ha establecido una jerarquía sin fijar el número de grupos
- La instrucción *cutree* permite fijar un número de clusters determinando haciendo un corte transversal del árbol a la altura que proporcione dicho número de clusters

```
cutree(hc, k = 3)
```

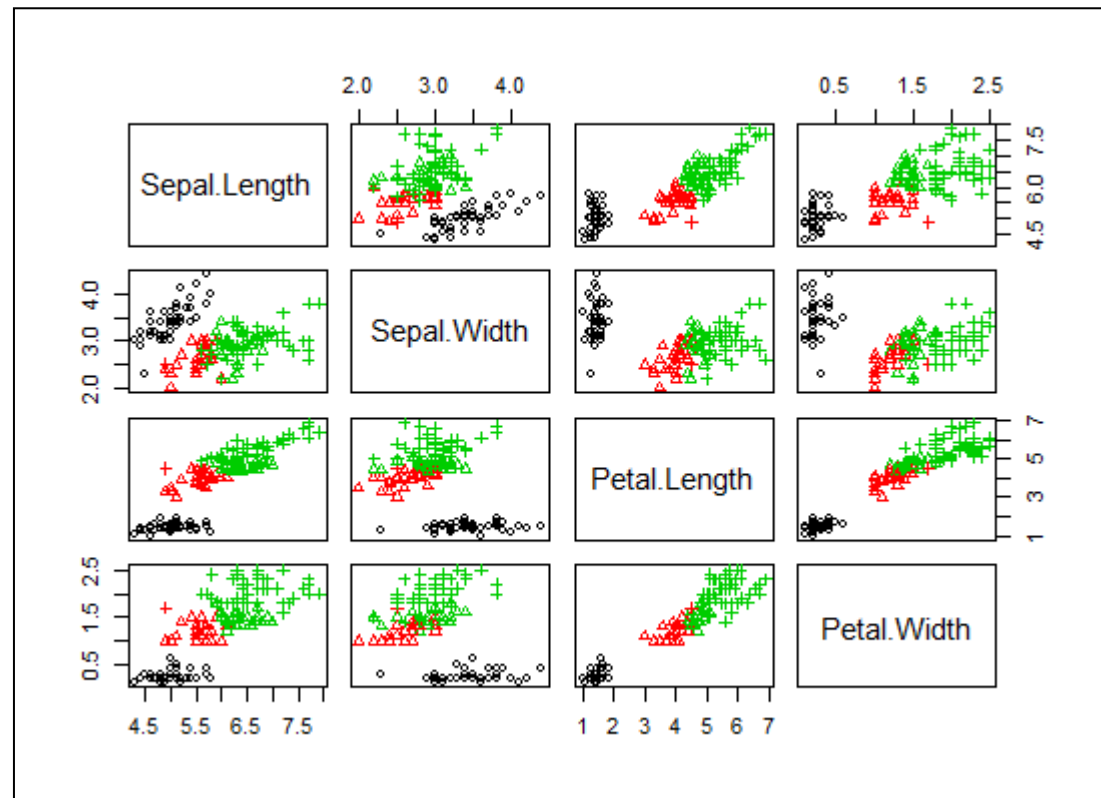
[illegible]

Clusterización jerárquica

Ejemplo R (iris)

- Correspondencia entre clústeres y especies da un 84% de acierto

	Clúster		
	1	2	3
setosa	50	0	0
versicolor	0	27	23
virginica	0	1	49



K-means

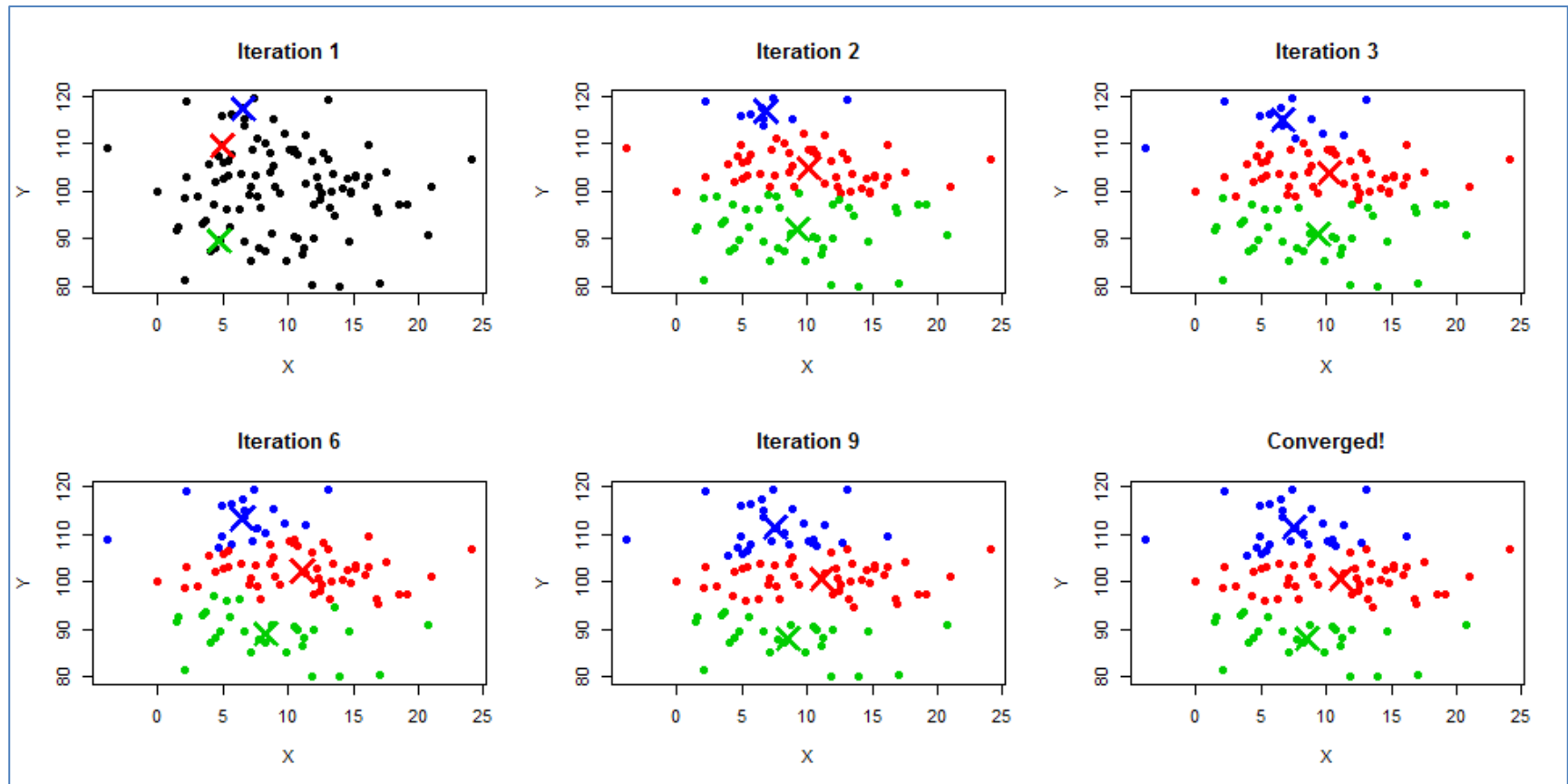
Introducción

- Tipos de variables: idealmente se precisan variables cuantitativas o ordinales con un gran número de categorías (aunque pueden implementarse distancias para otro tipo de variables)
- Ventajas
 - No requiere del cálculo de todos los pares de distancias
 - Requiere menos tiempo de computación que la clusterización jerárquica
- Inconvenientes
 - Resultados inaceptables para según qué tipo de clústeres (sobre todo NO convexos). Sol: usar versiones de K-means que minimicen este problema.
 - Distintos resultados dependiendo de los parámetros iniciales (sobre todo para conjuntos de datos pequeños). Sol: fijar distintos puntos iniciales.
 - Requiere fijar el número de clústeres a priori. Sol: si el coste computacional no es muy alto, probar diversos números de clústeres.
 - Al basarse en distancias medias, los *outliers* pueden afectar drásticamente al resultado. Sol: probar algoritmos más robustos (K-medians, K-mediods).

K-means

Ejemplo

- En cada iteración los puntos se van agrupando según sus similitudes



K-means

Proceso

- Objetivo: definir los clústeres de tal manera que se minimice la variabilidad intra-cluster.
- Determinar los parámetros iniciales
 - Número de clústeres
 - Número máximo de iteraciones
 - Número de ejecuciones con distintos puntos iniciales
- Ejecución del algoritmo

K-means

Algoritmo

Inicialización:

1. Especificar el número de clústeres (k)
2. Seleccionar aleatoriamente k elementos del conjunto de datos como centro de los clústeres

Repetir iterativamente:

3. Asignar cada elemento al clúster cuyo centro este más cercano
4. Recalcular el centro de gravedad para cada clúster

Se deja de iterar si se cumple algún criterio:

- Se alcanza el máximo número de iteraciones
- El cambio en la variabilidad-intra entre 2 iteraciones consecutivas es menor que un determinado umbral (o incluso nulo)

K-means

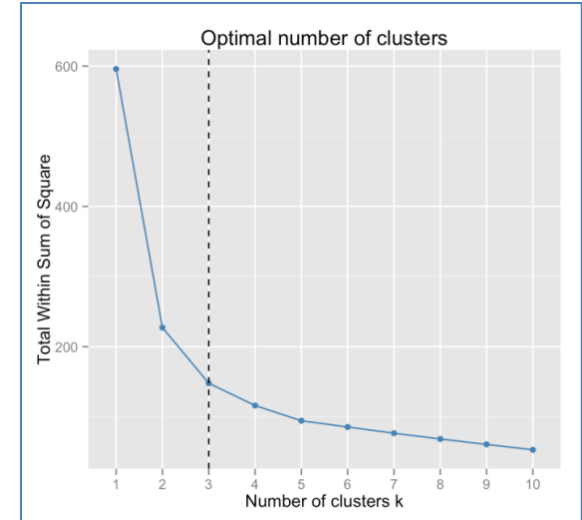
Parámetros iniciales

- ***Número máximo de iteraciones.*** K-means suele converger con pocas iteraciones. Ajustar según capacidad computacional, pero no debería ser un problema.
- ***Número de ejecuciones con distintos puntos iniciales.*** K-means tiene una componente aleatoria en la elección de los centroides iniciales. Conviene:
 - Realizar más de una ejecución del algoritmo (*nstart*) y quedarnos con la mejor
 - Poner una semilla para hacer los resultados reproducibles

K-means

Número de grupos

- **Regla del codo.** Comparar el porcentaje de variabilidad explicada (o variabilidad intra) para cada número de clúster. El punto en que se encuentre un codo (variación constante a partir de él) indicará el número idóneo de clústeres



- Coger un **número de clústeres k en función del número de observaciones**

$$k = \sqrt{n/2}$$

P.ej. para 800 observaciones, se obtendrían 20 clústeres

- **Usar otra indicador.** Hay múltiples indicadores aparte de la inercia para evaluar el rendimiento del algoritmo.

K-means

R

- `kmeans(x, centers, iter.max = 10, nstart = 1, algorithm = c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"), trace=FALSE)`
 - `x`: datos
 - `centers`: número de clústeres
 - `iter.max`: número máximo de iteraciones
 - `nstart`: número de puntos iniciales distintos
 - `algorithm`: método refinado del algoritmo
 - `trace`: imprimir proceso

K-means

Convergencia

- La convergencia hacia una solución está garantizada dado que la inercia-intra disminuye en cada iteración. Sin embargo, no está garantizado que converja hacia el óptimo.
- La convergencia es rápida (generalmente menos de 5 iteraciones, incluso, para grandes cantidades de datos).
- En general, se ejecuta el algoritmo con diferentes particiones iniciales y se retiene la solución más satisfactoria.
- Debido a la componente aleatoria, conviene poner una semilla antes de ejecutar el algoritmo.

K-means

Métodos

R implementa 4 métodos para hacer la partición:

- **Hartigan-Wong** (método por defecto). Optimiza el cálculo de distancias dividiendo los puntos pertenecientes a clústeres actualizados y no actualizados en cada iteración.
- **Lloyd**. Explicado en diapositivas previas. Método más simple.
- **Forgy** (o Forgy-Lloyd). Igual que el de Lloyd pero para los centroides iniciales se eligen puntos aleatorios dispersos (más eficiente con el algoritmo común)
- **MacQueen**. Actualiza los centros en cada punto que se mueve. Usa probabilidades y convergencias asintóticas.

K-means

Variantes. Paquetes de R

Diversos paquetes implementan métodos alternativos del k-means:

- ***clustMixType***. Variables categóricas y numéricas
- ***kml***. Datos longitudinales
- ***skmeans***. Spherical K-means
- ***trimcluster***. Métodos robustos ante la presencia de outliers
- ***Biganalytics***. Contiene la función *bigkmeans* para grandes conjuntos de datos

K-means

Variantes

- ***K-medians***. Usa medianas en vez de medias (más robusto)
- ***K-mediods***. Usa la instancia más representativa dentro del clúster. Puede usarse cualquier distancia (se pueden usar variables categóricas)
- ***Fuzzy C-Means***. Cada punto tiene un grado difuso de pertenecía a cada grupo.
- ***Esperanza-maximización***. Modelos de mezclas gaussianas. Emplean una asignación probabilística a cada grupo, en vez de asignaciones deterministas.
- ***K-means++***. Cambia el método de elección de los centroides iniciales.
- ***KD-trees***. Filtrado para mejorar la eficiencia en cada paso del algoritmo.
- ***Spherical k-means***. Para datos direccionales (ángulos en vez de distancias).
- ***Minkowski metric weighted k-means***. Soluciona el problema del ruido asignando pesos a las componentes de los vectores por grupos

Variantes

K-mediods

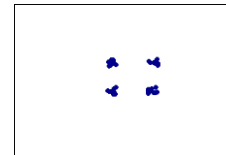
- La diferencia básica con el K-means es que en vez de escoger el centro de gravedad como centro del clúster, se escoge el **elemento (*mediod*) más representativo**
- No confundir con el algoritmo K-medians que se basa en las medianas univariantes.
- El *mediod* es aquel punto cuya **distancia media al resto de puntos del clúster es mínima**
- Es un algoritmo **más robusto** que el K-means ya que no es tan sensible a la presencia de outliers
- Una variante del algoritmo de **K-mediods** es **CLARA** (*CLustering LARge Applications*) que mejora el rendimiento para grandes volúmenes de datos
- En *R* existen funciones para K-mediods [*pam (cluster)*] y para su variante [*clara (cluster)*]

Medidas de rendimiento

Tipos

- **Internas.** Evalúa la calidad de la agrupación sin ninguna referencia externa. Hay distintas propiedades deseables:

- **Compacidad.** Baja variabilidad-intra



- **Separación.** Alta variabilidad-entre



- **Conectividad.** Hasta qué punto los elementos conexos pertenecen al mismo clúster.



- **Externas.** Comparan la agrupación con un resultado externo (otra agrupación o un conjunto que ya haya sido etiquetado)

Clustering

Medidas de rendimiento internas (1 agrupación)

- **Inercia intra** (suma de las distancias euclídeas al cuadrado de cada punto a su respectivo centroide). Valores entre 0 y ∞ . Decrece monótonamente al aumentar el número de clústeres:

$$I = \sum_{i=1}^k \sum_{j=1}^{n_k} (x_{ij} - c_j)^2$$

- **Dunn index** (cociente entre distancia mínima entre puntos de distintos clústeres y la distancia máxima entre puntos del mismo clúster). Valores entre 0 y ∞ . Se debe maximizar. R: *dunn(clValid)*

$$D = \frac{d_{\min}}{d_{\max}}$$

- **Silhouette coefficient**. Valores entre -1 y 1. Se debe maximizar. Para un punto concreto i , $a(i)$ es la media de las distancias a los puntos del mismo clúster y $b(i)$, la mínima distancia a puntos de otro clúster. Este coeficiente es la media de todos los $s(i)$. R: *silhouette(cluster)*

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Clustering

Medidas de rendimiento externas (similitud de 2 agrupaciones)

- Para cada pareja de puntos en 2 agrupaciones (C1 y C2), se define:
 - A:** parejas en mismo clúster en ambas agrupaciones C1 y C2
 - B:** parejas en distintos clústeres en ambas agrupaciones C1 y C2
 - C:** parejas en mismo clúster en C1 y distinto clúster en C2
 - D:** parejas en distinto clúster en C1 y mismo clúster en C2
- **Rand index** (Proporción de parejas igual de agrupadas en ambas agrupaciones). Toma valores entre 0 y 1. Cuánto mayor, más similitud. R: *randIndex (flexclust)*

$$R = \frac{\# \text{coincidencias por parejas}}{\# \text{parejas}} = \frac{A + B}{A + B + C + D}$$

- **Jaccard index** (Proporción de parejas en el mismo clúster). Toma valores entre 0 y 1. Cuánto mayor, más similitud. R: *jaccard_indep (clusteval)*

$$J = \frac{\# \text{parejas en el mismo clúster}}{\# \text{parejas} - \# \text{parejas en distintos clústeres en ambas}} = \frac{A}{A + C + D}$$

Sistemas mixtos

K-means + Clusterización jerárquica

■ **Opción 1:** Útil con un gran número de observaciones

- **Etapas 1**. Se realiza un *k-means* con un elevado número de grupos (p.ej, 100)
- **Etapas 2**. Se realiza una clusterización jerárquica de los grupos usando sus centros de gravedad.

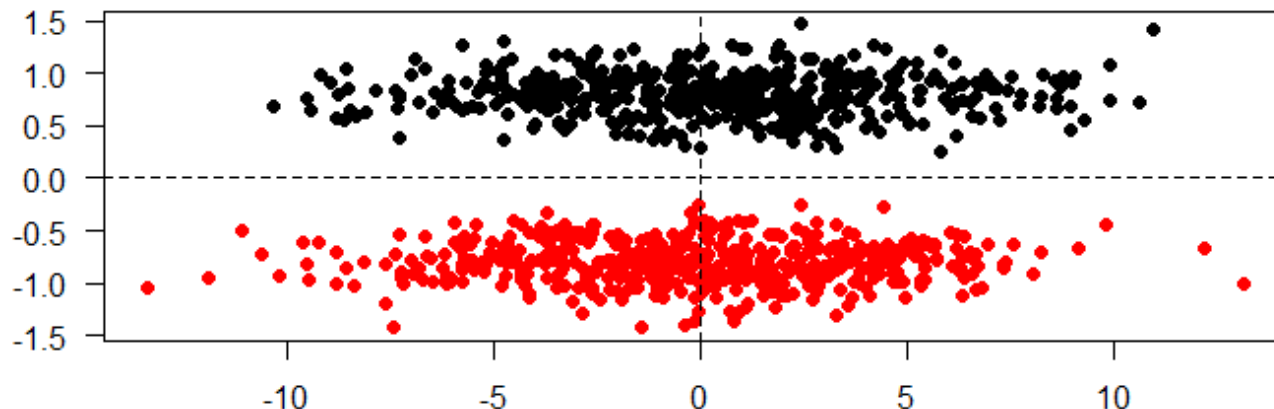
■ **Opción 2:** Útil con pocas observaciones. En ciertos casos, mejora el rendimiento

- **Etapas 1**. Se realiza una clusterización jerárquica
- **Etapas 2**. Partiendo de una partición inicial de esta clusterización se pueden reasignar los puntos según el algoritmo de k-means

Sistemas mixtos

(ACP o ACM) + (K-means o clusterización jerárquica)

- Útil para eliminar ruido (variables irrelevantes) o para tratar con todo tipo de variables (categóricas inclusive).
 - Se realiza un ACP y/o ACM y se retiene las componentes principales con varianza no nula
 - Se aplica una clusterización jerárquica/*k-means* con dichas componentes usando la distancia euclídea
- Inconveniente: En algunos casos, el ACP puede ser contraproducente

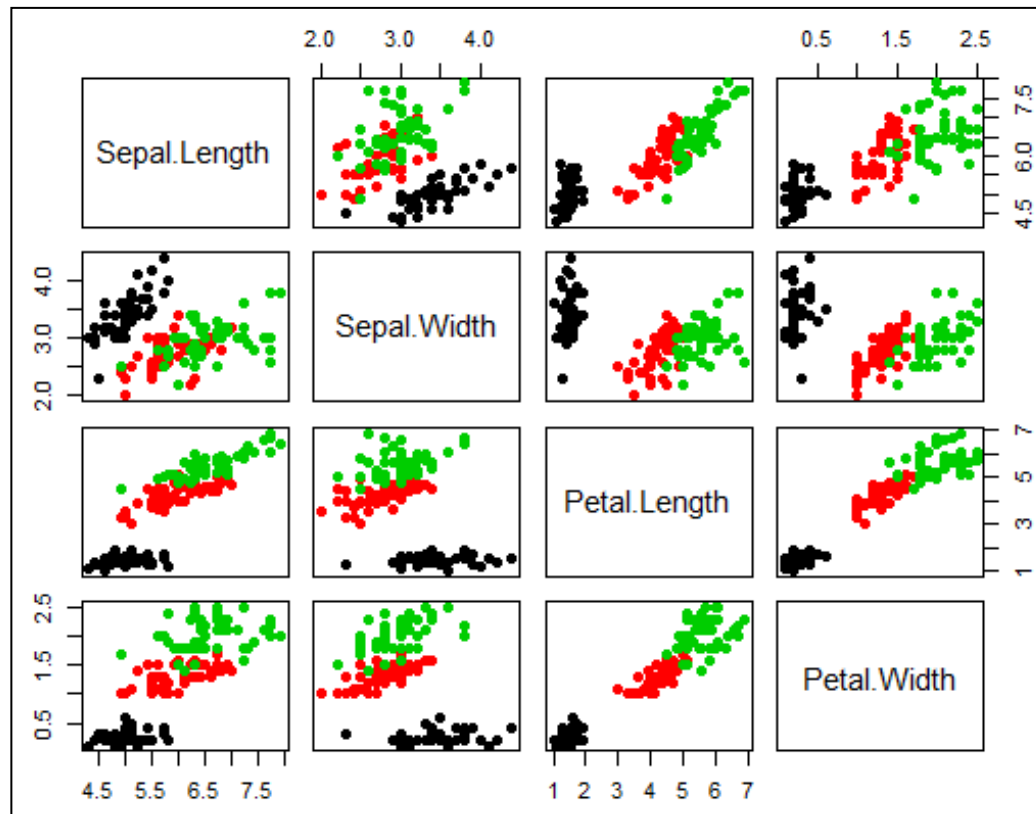


K-means

Ejemplo R

Datos de longitud y anchura de sépalos y pétalos de 3 tipos de especie de plantas

```
iris2 <- scale(iris[,1:4]) # Se elimina especie  
pairs(iris2,col=iris$Species,pch=19) # Descriptiva bivariante
```



K-means

Ejemplo R – 2 grupos

- En un principio, se desconoce el número de grupos. Se prueba 2, 3 y 4

```
km2 <- kmeans(iris2,2,nstart=10) # Algoritmo de k-means para 2 grupos
km2
```

K-means clustering with 2 clusters of sizes 97, 53

Cluster means:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	6.301031	2.886598	4.958763	1.695876
2	5.005660	3.369811	1.560377	0.290566

Clustering vector:

```
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 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
```

Within cluster sum of squares by cluster:

```
[1] 123.79588 28.55208
```

```
(between_SS / total_SS = 77.6 %)
```

K-means

Ejemplo R – 3 grupos

```
km3 <- kmeans(iris2,3,nstart=10) # k-means para 3 grupos
km3
```

K-means clustering with 3 clusters of sizes 62, 50, 38

Cluster means:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.901613	2.748387	4.393548	1.433871
2	5.006000	3.428000	1.462000	0.246000
3	6.850000	3.073684	5.742105	2.071053

Clustering vector:

```
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 2 2 2 2 2 2 2 2 2 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 3 3 3
3 1 3 3 3 3 3 3 1 1 3 3 3 3 1 3 1 3 1 3 3 1 1 3 3 3 3 1 3 3 3 1
3 3 3 1 3 3 3 1 3 3 1
```

Within cluster sum of squares by cluster:

```
[1] 39.82097 15.15100 23.87947
(between_SS / total_SS = 88.4)
```

Ejemplo R – 4 grupos

```
km4 <- kmeans(iris2,4,nstart=10) # k-means para 4 grupos
```

km4

K-means clustering with 4 clusters of sizes 32, 28, 40, 50

Cluster means:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	6.912500	3.100000	5.846875	2.131250
2	5.532143	2.635714	3.960714	1.228571
3	6.252500	2.855000	4.815000	1.625000
4	5.006000	3.428000	1.462000	0.246000

[illegible]

Within cluster **sum** of squares **by** cluster:

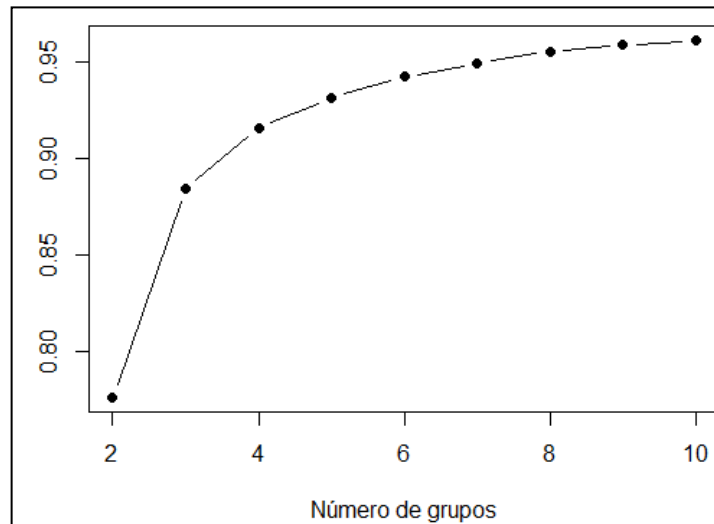
```
[1] 18.703437 9.749286 13.624750 15.151000
```

(between SS / total SS = 91.6 %)

K-means

Ejemplo R

- Variabilidad explicada:
 - 2 grupos: 77.6%
 - 3 grupos: 88.4%
 - 4 grupos: 91.6%
- La ganancia de variabilidad explicada por los grupos respecto al total al pasar de 2 a 3 personas es suficientemente importante como para considerarla. No ocurre lo mismo con el paso de 3 a 4. Por tanto, parece que la mejor opción es quedarse con 3 grupos.



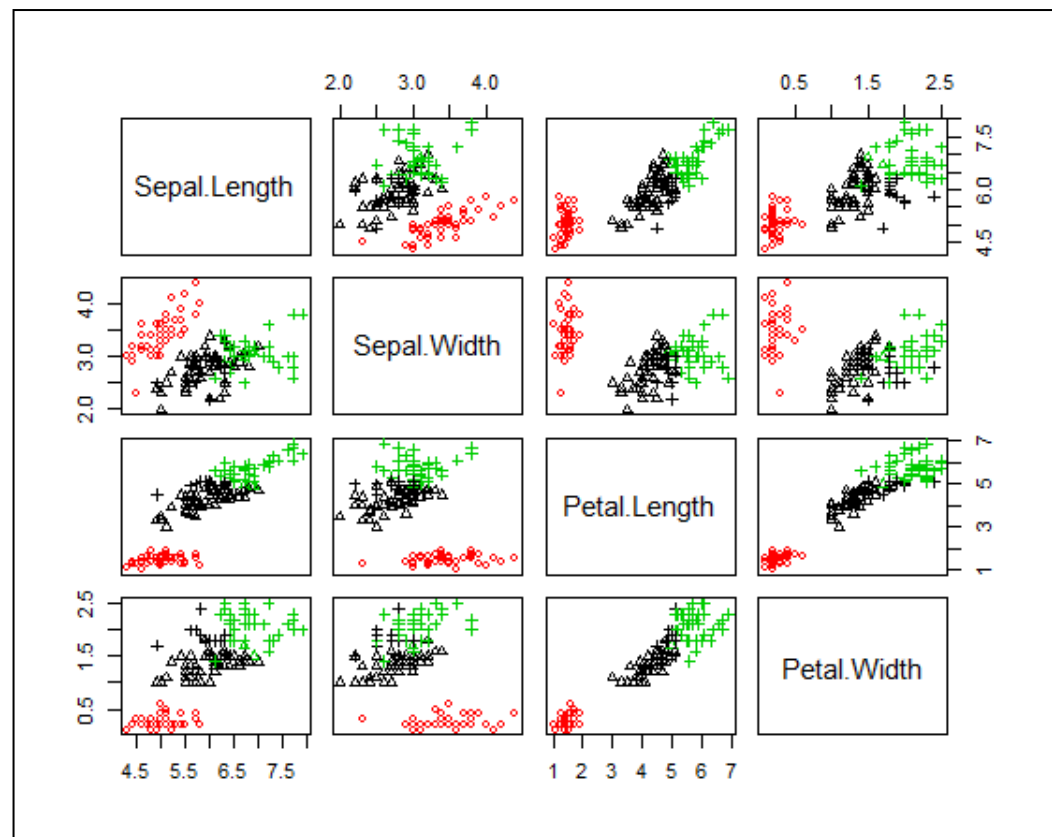
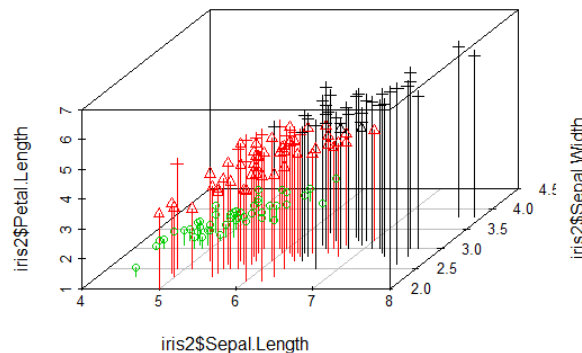
K-means

Ejemplo R

- El algoritmo de k-means es no supervisado. No obstante, en este caso, podemos chequear si concuerdan los grupos con las especies (acierto: 89%)

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

- Podría ser que una 3ª dimensión no visible en el plano aportase más información



A collection of approximately 18 squares in three shades of blue, grey, and light blue, arranged in a sparse, abstract pattern across the top half of the slide.

MUBD

Màster Universitari en Enginyeria de Dades Massives (Big Data)

Estadística

