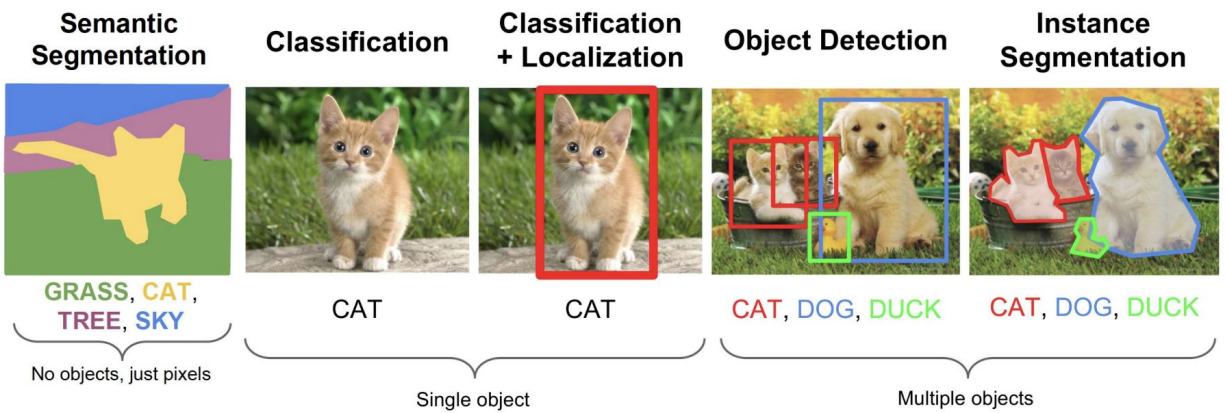
UNIVERSIDAD DE LA REPUBLICA
URUGUAY

# Clustering

## Introducción a la Ciencia de Datos 2025

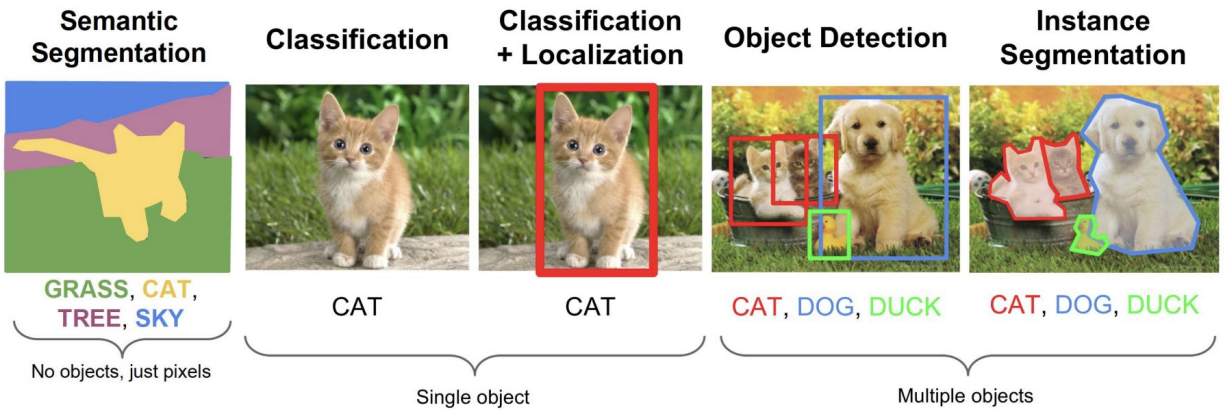
(slides prestadas del curso “Laboratorio de Datos” de la UBA y del curso CICADA 2023)

# Repaso: Aprendizaje supervisado




Slides tomadas de curso DLVIS 2024

# Repaso: Aprendizaje supervisado




- Datos etiquetados: $(\mathbf{x}, y)$
- Objetivo: aprender una función $f(\mathbf{x}) \approx y$
- Probabilísticamente: aprender $p(y \mid \mathbf{x})$

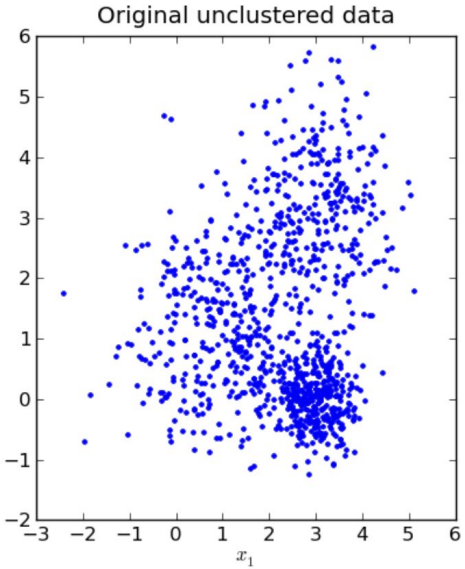
Slides tomadas de curso DLVIS 2024

# Aprendizaje NO supervisado

- Datos **no etiquetados**: $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- **Objetivo:** aprender estructura de datos

**Ejemplos:**

- Clustering

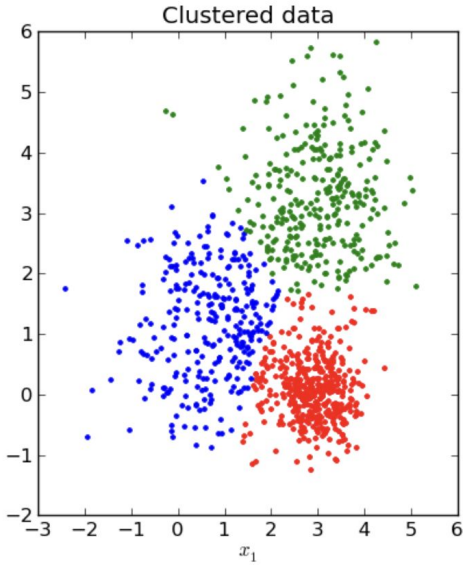


# Aprendizaje NO supervisado

- Datos **no etiquetados**: $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- **Objetivo:** aprender estructura de datos

**Ejemplos:**

- Clustering

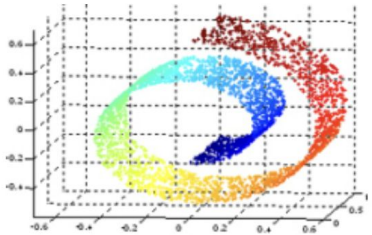


# Aprendizaje NO supervisado

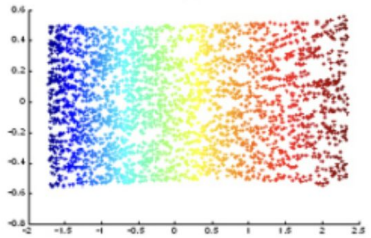
- Datos **no etiquetados**: $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- **Objetivo:** aprender estructura de datos

**Ejemplos:**

- Clustering
- Reducción de dimensión



(a)



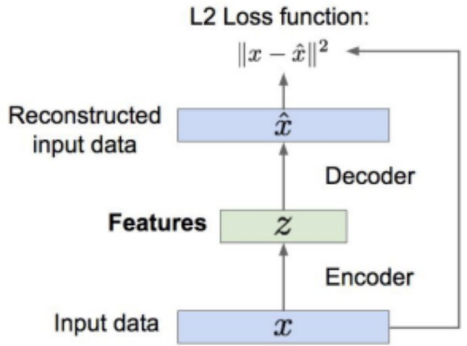
(c)

# Aprendizaje NO supervisado

- Datos **no etiquetados**: $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- **Objetivo:** aprender estructura de datos

**Ejemplos:**

- Clustering
- Reducción de dimensión
- *Feature representation*



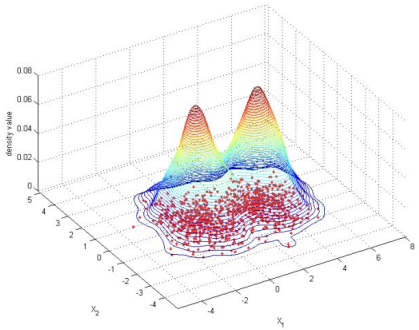

Wikipedia

# Aprendizaje NO supervisado

- Datos **no etiquetados**: $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- **Objetivo:** aprender estructura de datos

**Ejemplos:**

- Clustering
- Reducción de dimensión
- *Feature representation*
- Estimación de densidad subyacente $p_\theta(\mathbf{x})$



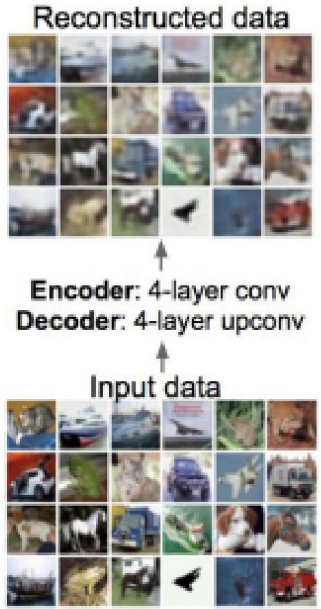

cs231n (Stanford)

# Aprendizaje NO supervisado

- Datos **no etiquetados**: $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- **Objetivo:** aprender estructura de datos

**Ejemplos:**

- Clustering
- Reducción de dimensión
- *Feature representation*
- Estimación de densidad subyacente $p_\theta(\mathbf{x})$
- Síntesis de datos (muestras de la densidad subyacente)




cs231n (Stanford)

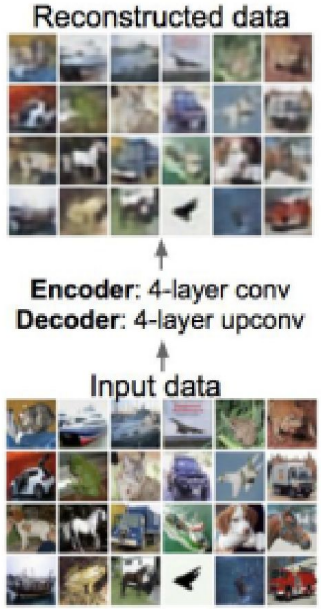
Slides tomadas de curso DLVIS 2024

# Aprendizaje NO supervisado

- Datos **no etiquetados**: $\mathbf{x}_1, \ldots, \mathbf{x}_n$
- **Objetivo:** aprender estructura de datos

**Ejemplos:**

- Clustering
- Reducción de dimensión
- *Feature representation*
- Estimación de densidad subyacente $p_\theta(\mathbf{x})$




Gran ventaja del aprendizaje no supervisado:

- No necesito etiquetas: datos abundan (en general)

Slides tomadas de curso DLVIS 2024

# Motivación: ¿Por qué estudiamos el aprendizaje no supervisado?

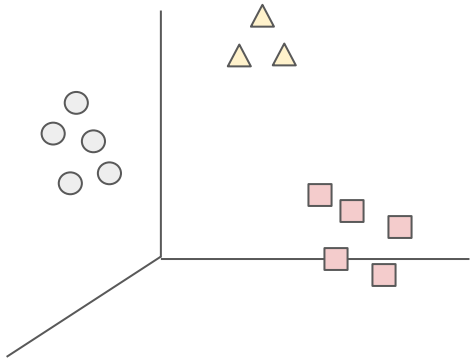
🌍 Mayoría de los datos no están etiquetados: es caro y a menudo inviable.
🧠 Descubrir estructuras subyacentes: revelar patrones y agrupaciones en datos.
⚙️ Base para el aprendizaje autosupervisado.
🚀 Escala mejor: Eficaz para conjuntos de datos a gran escala sin curar.

**Ejemplos de aplicaciones:**

- Reconocimiento de imágenes a gran escala (por ejemplo, Facebook/Google)
- Agrupamiento del comportamiento de los clientes en comercio electrónico
- Clasificación del uso del suelo a partir de imágenes satelitales

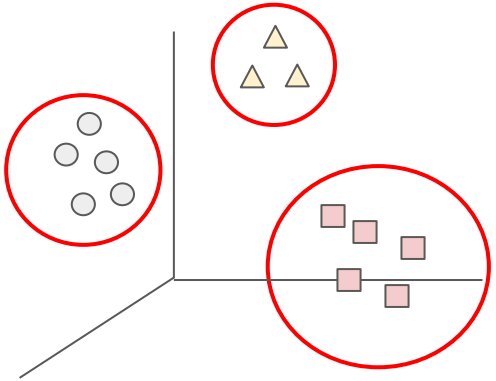
# Clustering

Encontrar **subgrupos** (*clústers*) en los datos



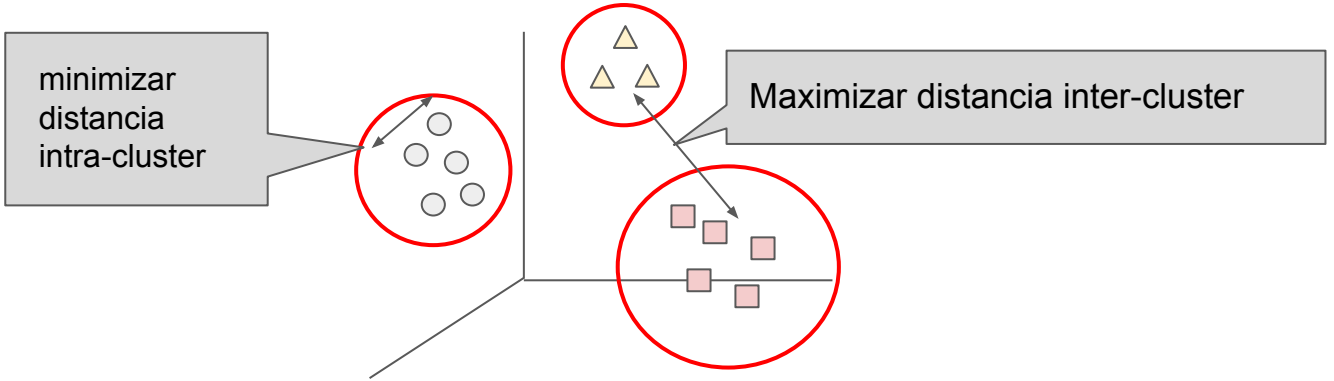
# Clustering

Encontrar **subgrupos** (*clústers*) en los datos



# Clustering

Encontrar **subgrupos** (*clústers*) en los datos

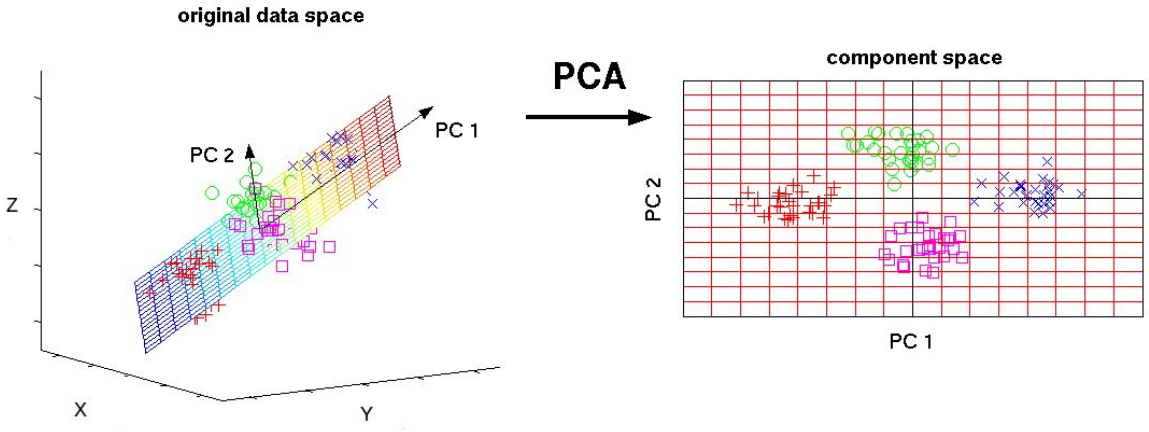



Observaciones dentro de un cluster **similares**
Observaciones entre clusters **no similares**

# Clustering

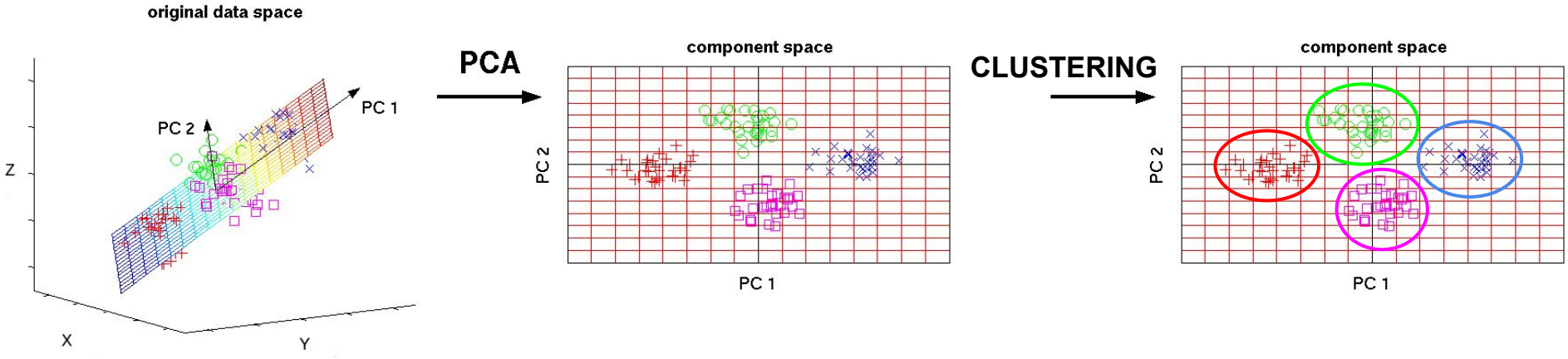
- Objetivo: descubrir estructura dentro de un conjunto de datos, dividiéndolo en subgrupos que muestran una cierta coherencia.
    - *Coherencia*: muestras dentro de un mismo grupo o *cluster* son más parecidas entre sí que a las muestras de otros clusters.
    - *Muestras parecidas:* noción de similitud o de distancia entre muestras.
- La mayor parte de los métodos de clustering son de dos tipos:
    - *Particionales*: producen una única partición que optimiza una función criterio
    - *Jerárquicos*: jerarquía de particiones anidadas; cada nivel de la jerarquía es en sí mismo una partición, obtenida por unión de *clusters* de la jerarquía inferior.
- **Importante**:
  Un método de clustering siempre produce clusters, aunque éstos no existan realmente ⇒ todo método de clustering debe ser seguido de una etapa de validación de los clusters obtenidos.

# Clustering




Reducir dimensión maximizando la varianza

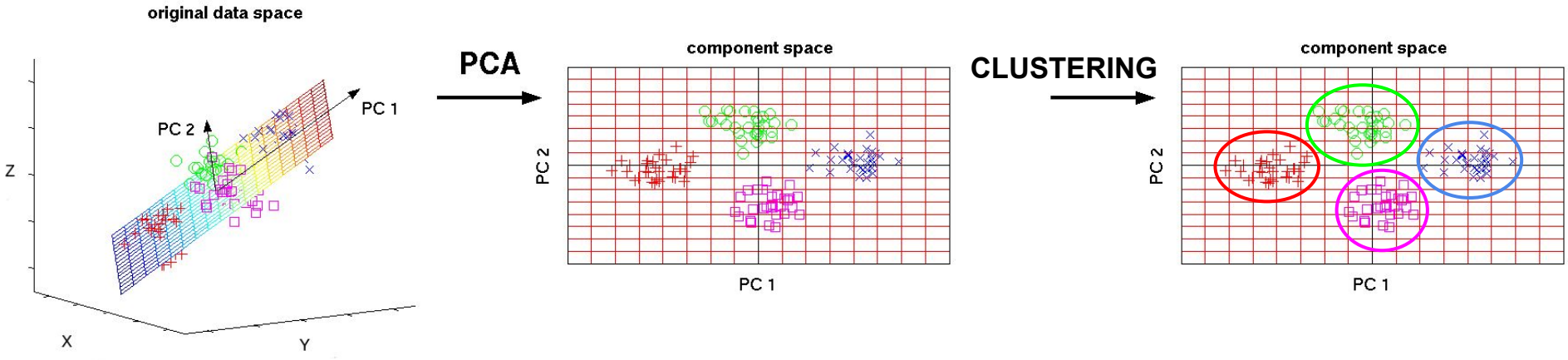
# Clustering




Reducir dimensión maximizando la varianza

Encontrar grupos homogéneos

# Clustering




Reducir dimensión maximizando la varianza

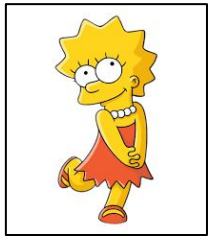
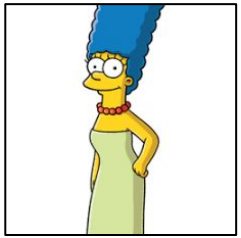
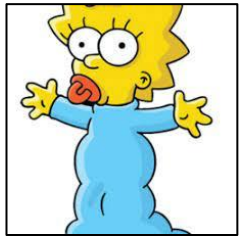
Encontrar grupos homogéneos

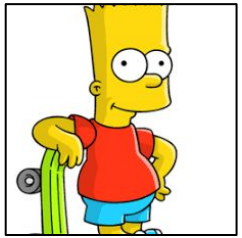
Se puede encontrar grupos en el espacio de features original

Si son muchos -> podría ser costoso computacionalmente

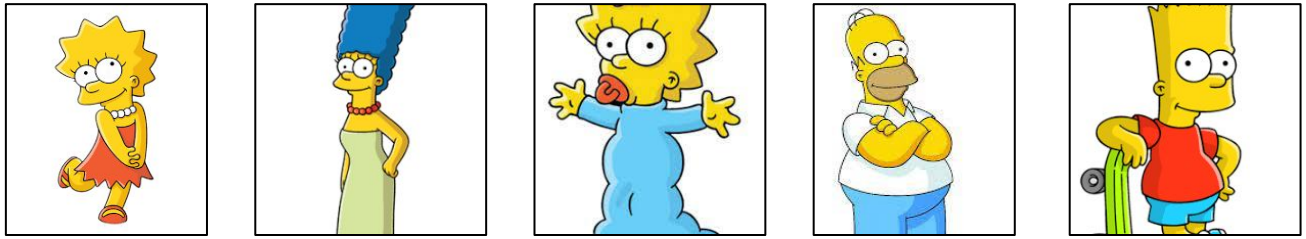
-> podrían esconderse las características que mejor agrupan los datos

# Clustering - forma natural de agrupar los datos

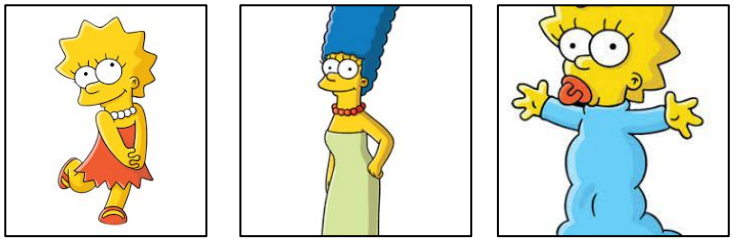






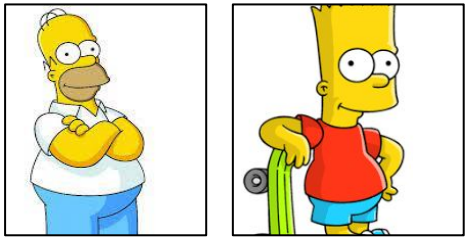
# Clustering - forma natural de agrupar los datos



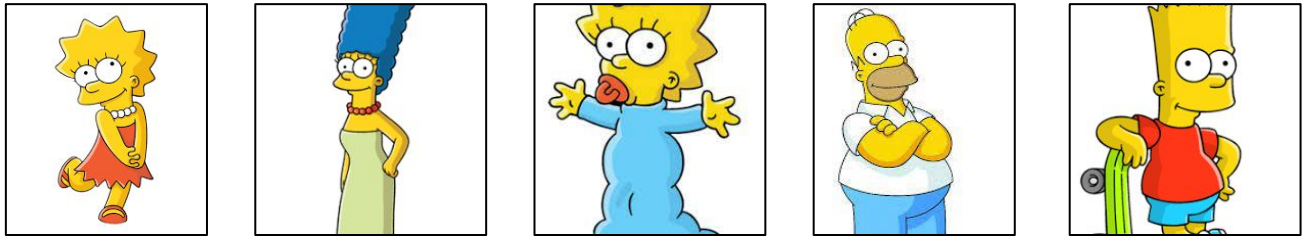
Mujeres



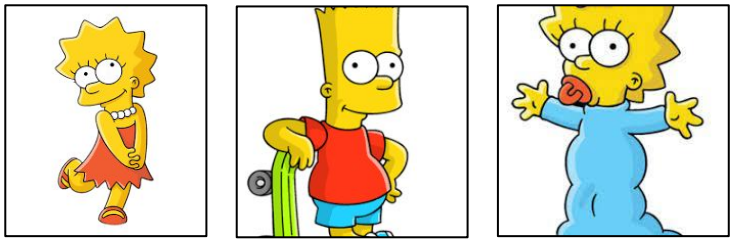
Hombres



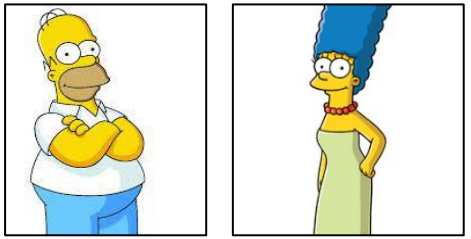
# Clustering - forma natural de agrupar los datos



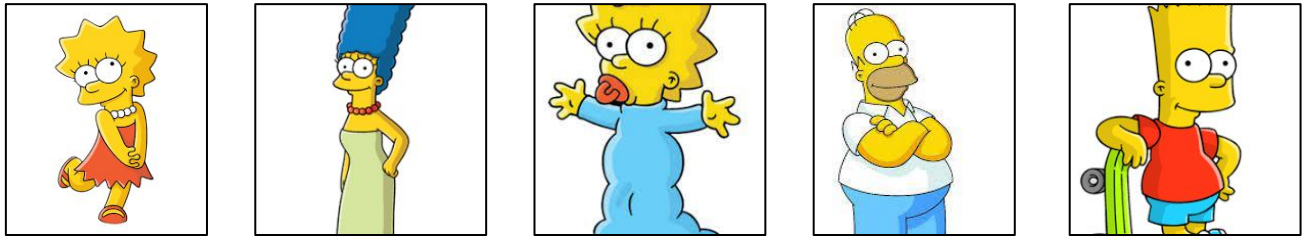
Niñes



Adultos

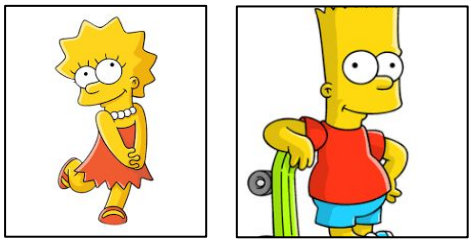


# Clustering - forma natural de agrupar los datos

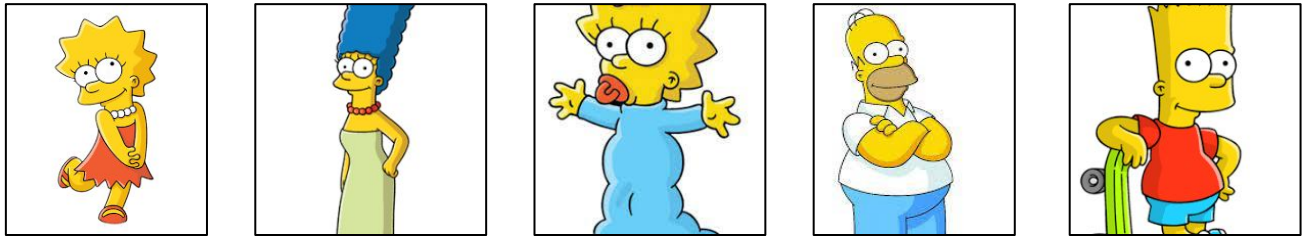


No van a la primaria

Van a la primaria



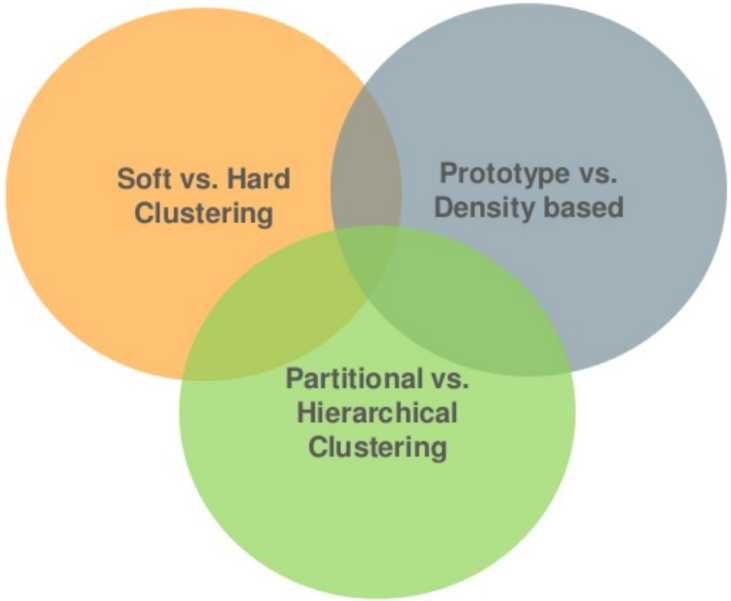
# Clustering - forma natural de agrupar los datos



No hay una forma natural de agrupar los datos, el clustering es **subjetivo**, la mejor elección de grupos depende de qué le queremos preguntar a los datos

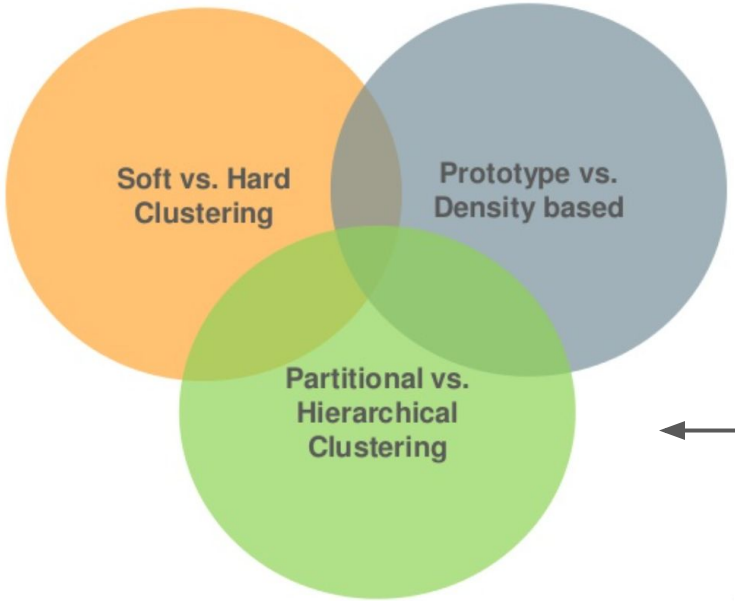
# Clustering - estrategias

Hay muuuuchos métodos de clusterización y distintos criterios de división

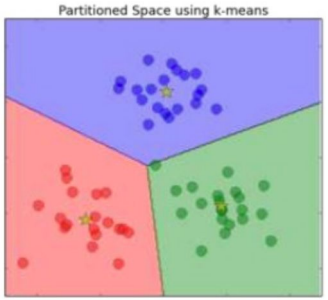


# Clustering - estrategias

Hay muuuuchos métodos de clusterización y distintos criterios de división.

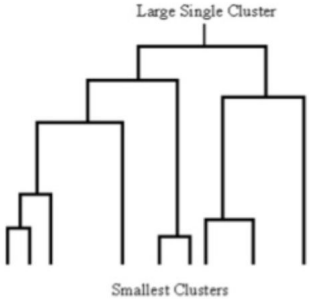



Partición



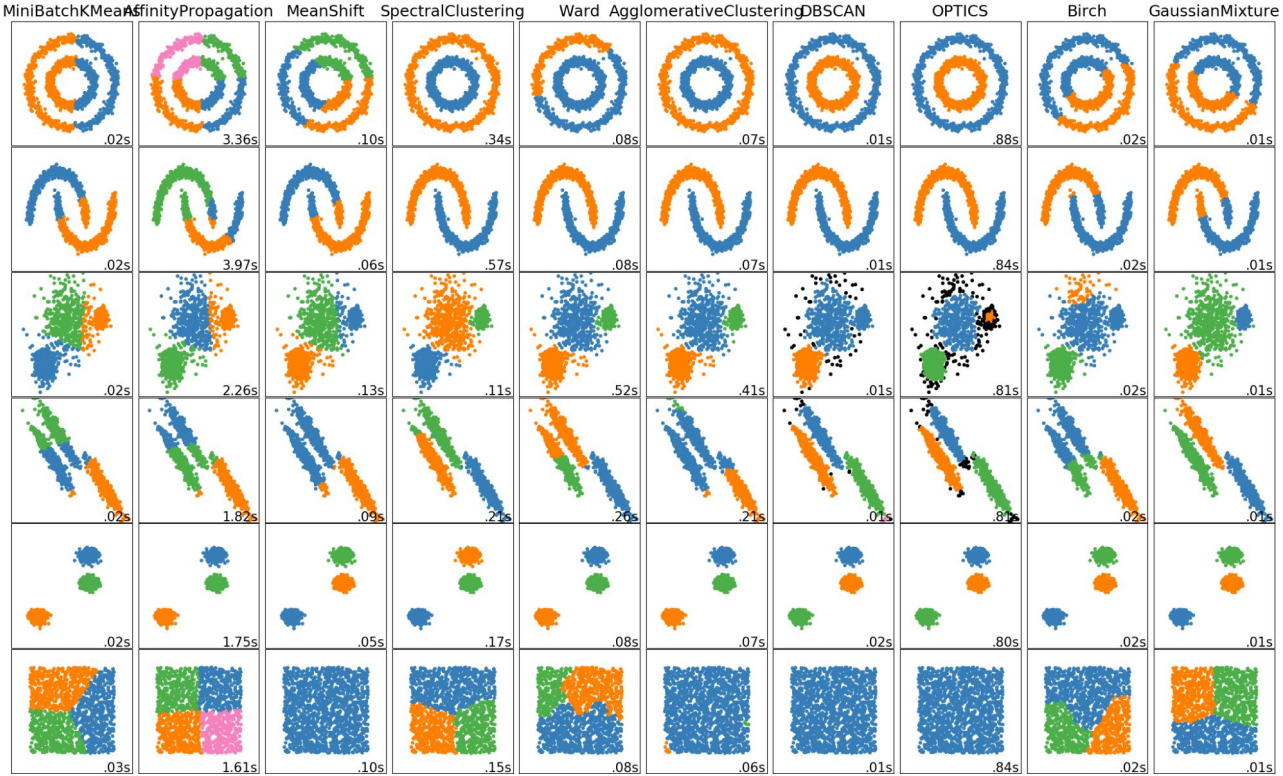

-particiona el espacio
-encuentra todos los clusters simultáneamente

Jerárquico




-genera una jerarquía de clusters anidados

# Clustering - ejemplos y desafíos



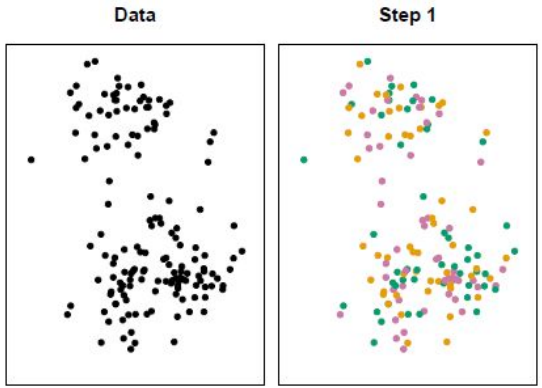
# K-means: Esquema

Damos el número de clusters ***k*** que queremos obtener

# K-means: Esquema

Damos el número de clusters ***k*** que queremos obtener

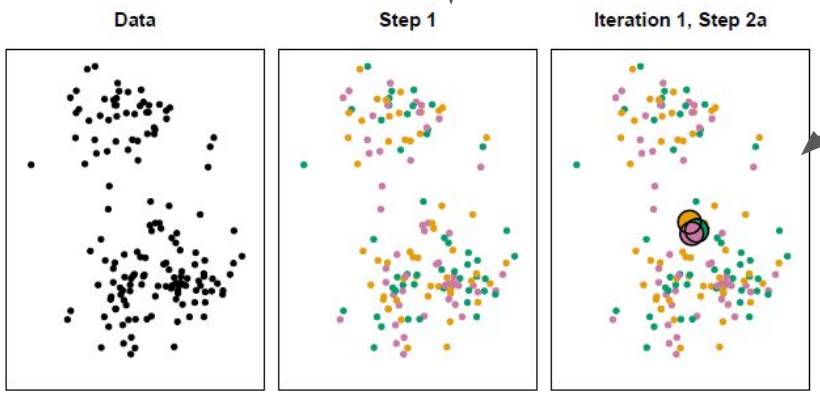
Inicialización random



# K-means: Esquema

Damos el número de clusters ***k*** que queremos obtener

Inicialización random

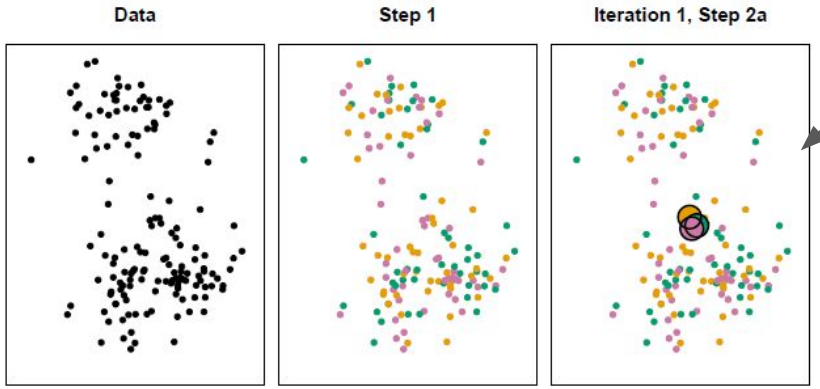



Computa los ***centroides*** (centros) de cada cluster como el promedio de las features de sus samples

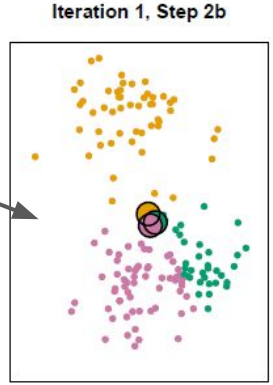
# K-means: Esquema

Inicialización random

Damos el número de clusters ***k*** que queremos obtener




Computa los ***centroides*** (centros) de cada cluster como el promedio de las features de sus samples

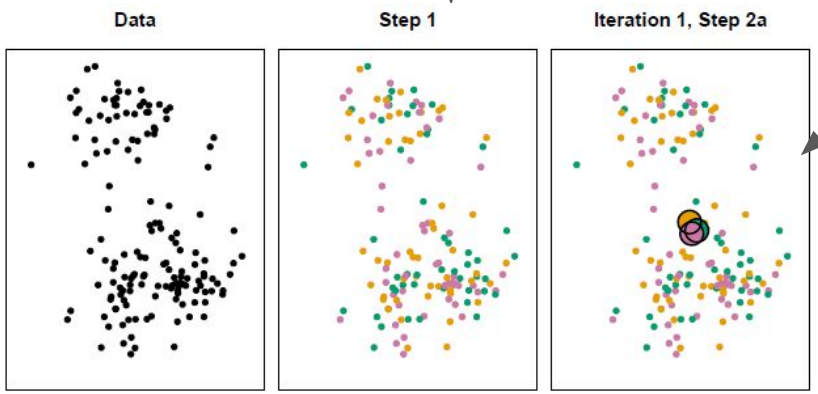



le asigna a cada sample la etiqueta del cluster cuyo centroide es más cercano (distancia euclídea al cuadrado)

# K-means: Esquema

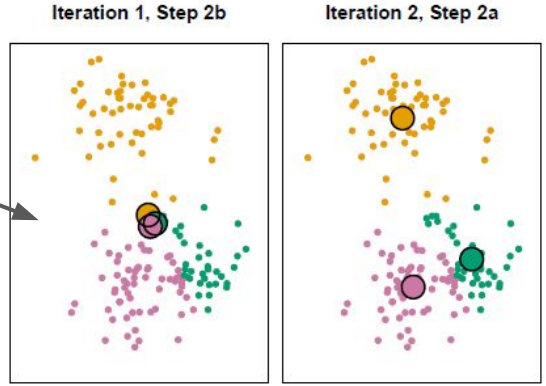
Inicialización random

Damos el número de clusters ***k*** que queremos obtener




Computa los ***centroides*** (centros) de cada cluster como el promedio de las features de sus samples

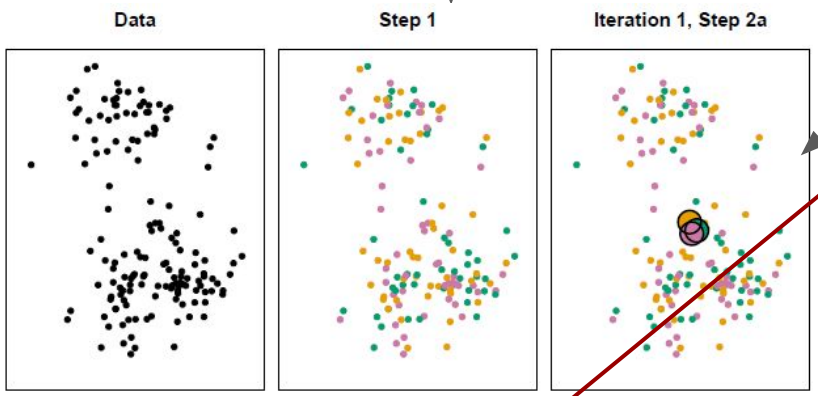
le asigna a cada sample la etiqueta del cluster cuyo centroide es más cercano (distancia euclídea al cuadrado)



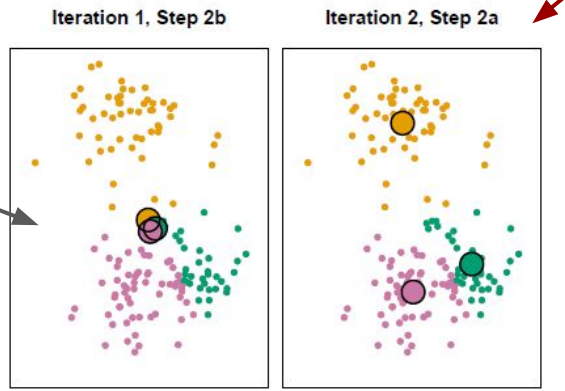
# K-means: Esquema

Inicialización random

Damos el número de clusters ***k*** que queremos obtener




Computa los ***centroides*** (centros) de cada cluster como el promedio de las features de sus samples

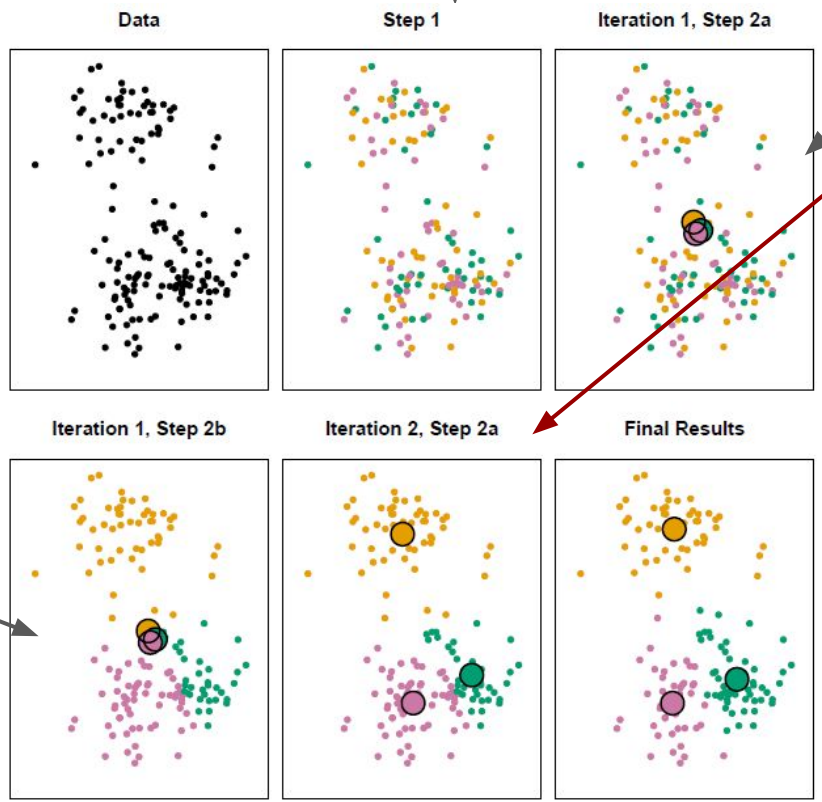



le asigna a cada sample la etiqueta del cluster cuyo centroide es más cercano (distancia euclídea al cuadrado)

# K-means: Esquema

Inicialización random

Damos el número de clusters ***k*** que queremos obtener




Computa los ***centroides*** (centros) de cada cluster como el promedio de las features de sus samples

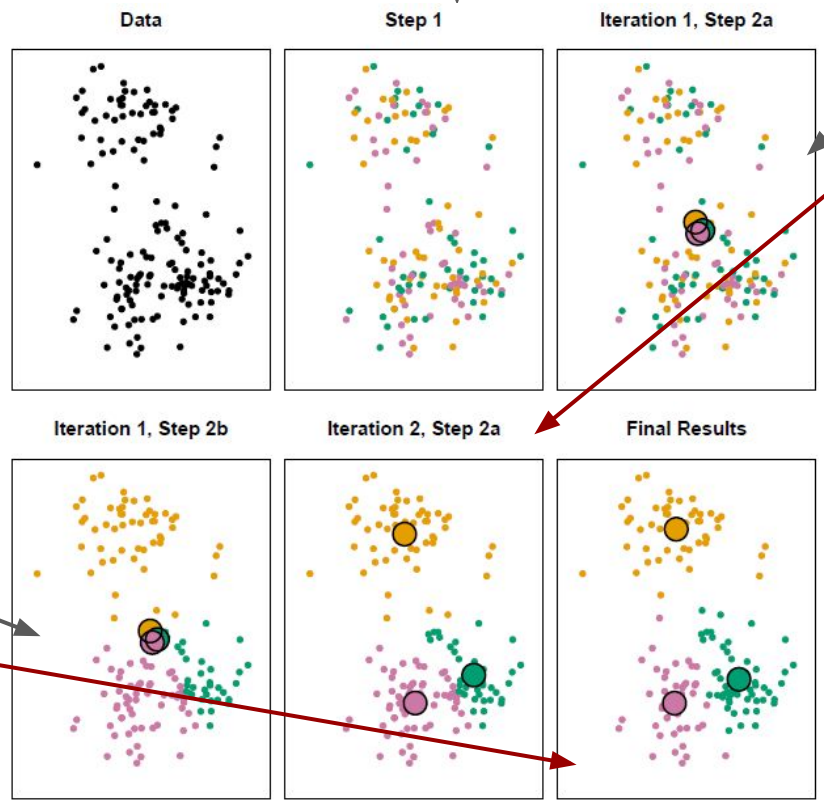
le asigna a cada sample la etiqueta del cluster cuyo centroide es más cercano (distancia euclídea al cuadrado)

# K-means: Esquema

Inicialización random

Computa los ***centroides*** (centros) de cada cluster como el promedio de las features de sus samples

Damos el número de clusters ***k*** que queremos obtener

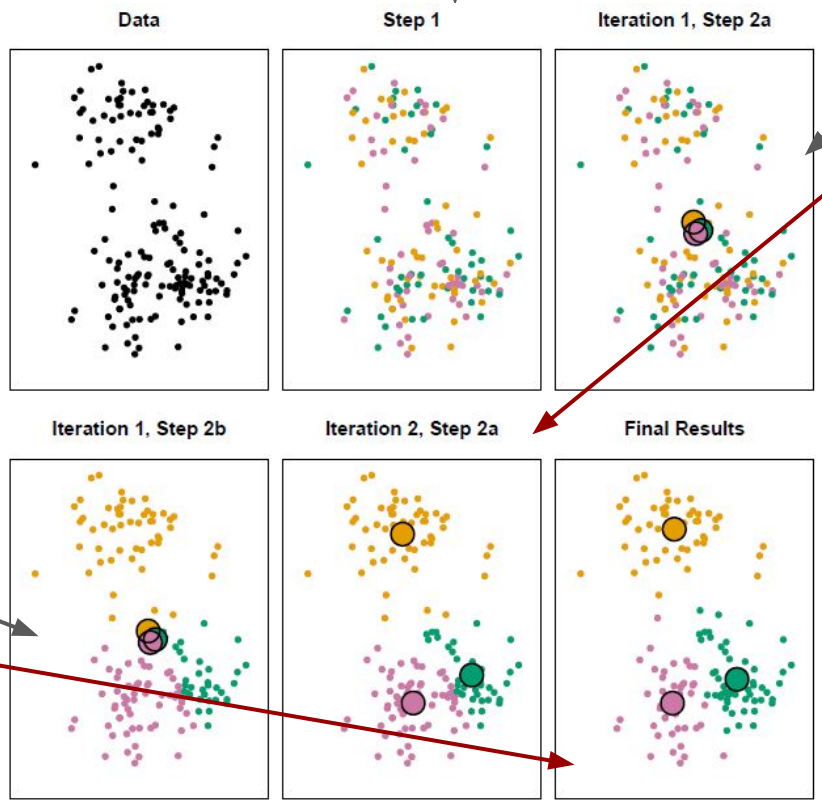



le asigna a cada sample la etiqueta del cluster cuyo centroide es más cercano (distancia euclídea al cuadrado)

# K-means: Esquema

Inicialización random

Damos el número de clusters ***k*** que queremos obtener




Computa los ***centroides*** (centros) de cada cluster como el promedio de las features de sus samples

le asigna a cada sample la etiqueta del cluster cuyo centroide es más cercano (distancia euclídea al cuadrado)

Termina cuando en una iteración no hay cambio de etiqueta o se llega a un máximo de iteraciones 'max_iter'

# K-means: Esquema

Inicialización random

Damos el número de clusters ***k*** que queremos obtener

# K-means: Esquema

Damos el número de clusters ***k*** que queremos obtener


Inicialización random

sklearn.cluster.KMeans

init='random'

**init='k-means++'**

# K-means: Esquema

Damos el número de clusters ***k*** que queremos obtener


Inicialización random

sklearn.cluster.KMeans

init='random'

elige aleatoriamente ***k*** samples como centroides

**init='k-means++'**

# K-means: Esquema

Inicialización random

sklearn.cluster.KMeans

Damos el número de clusters ***k*** que queremos obtener


init='random'

elige aleatoriamente ***k*** samples como centroides

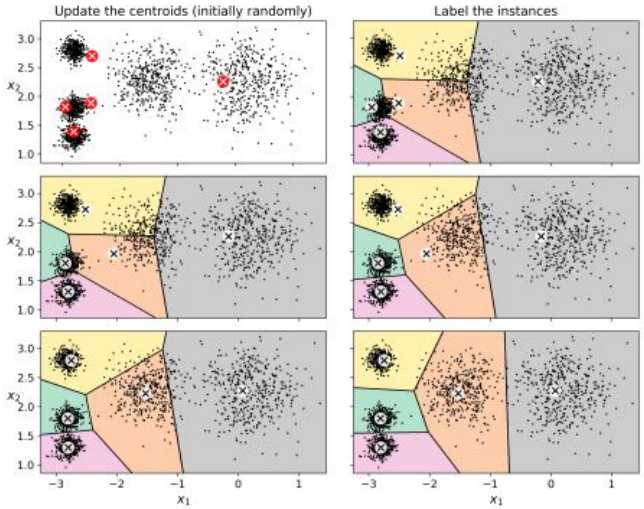
**init='k-means++'**

mayor velocidad de convergencia

1. Selecciona aleatoriamente un dato y lo asigna como centroide
2. Para los otros datos x, calcula D(x), distancia entre x y el centro más cercano que ya ha sido seleccionado.
3. Escoge un nuevo punto al azar como nuevo centroide, utilizando una distribución de probabilidad ponderada donde un punto x es escogido con la probabilidad proporcional a $D(x)^2$.
4. Repite paso 2 y 3 hasta que se hayan seleccionado k centroides.

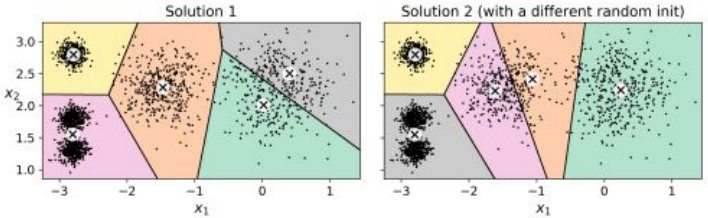
# K-means: inicialización

Evolución con un buena inicialización.




En este caso converge en tres iteraciones.

Otras inicializaciones pueden conducir a soluciones sub-óptimas.




Cómo resolvemos esto:

- Métodos de inicialización.
- Métricas de calidad de la partición.

# K-means: Función objetivo

Buena clusterización es la que minimiza la varianza entre datos de un mismo cluster

# K-means: Función objetivo

Buena clusterización es la que minimiza la varianza entre datos de un mismo cluster

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

# K-means: Función objetivo

Buena clusterización es la que minimiza la varianza entre datos de un mismo cluster

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

distancia euclídea al cuadrado - lo más usual

# K-means: Función objetivo

Buena clusterización es la que minimiza la varianza entre datos de un mismo cluster

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

# K-means: Función objetivo

Buena clusterización es la que minimiza la varianza entre datos de un mismo cluster

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

Básicamente K-means es un algoritmo de optimización de esta función objetivo

# K-means: Función objetivo

Buena clusterización es la que minimiza la varianza entre datos de un mismo cluster

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

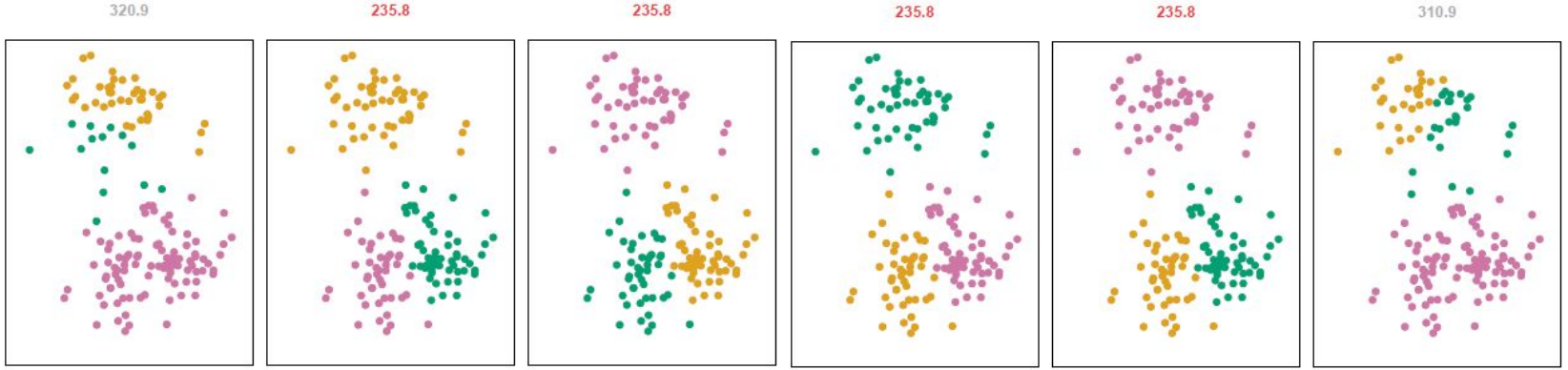
Básicamente K-means es un algoritmo de optimización de esta función objetivo
Depende de la inicialización -> **método no determinista**

# K-means: Función objetivo

Buena clusterización es la que minimiza la varianza entre datos de un mismo cluster

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

Básicamente K-means es un algoritmo de optimización de esta función objetivo
Depende de la inicialización -> **modelo no determinista**




distintas inicializaciones del mismo modelo con los mismos datos

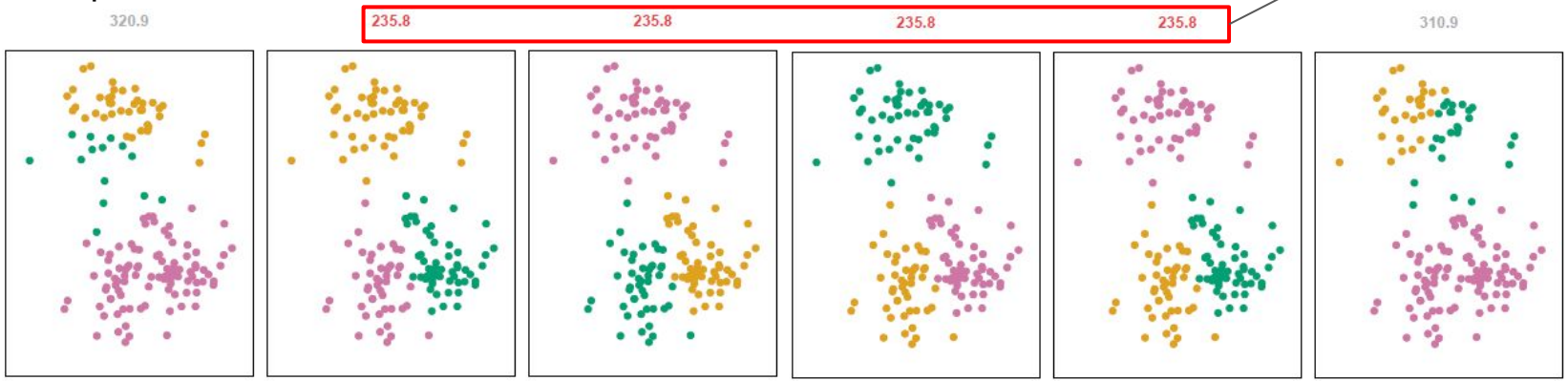
# K-means: Función objetivo

Buena clusterización es la que minimiza la varianza entre datos de un mismo cluster

$$\text{SSE} = \underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

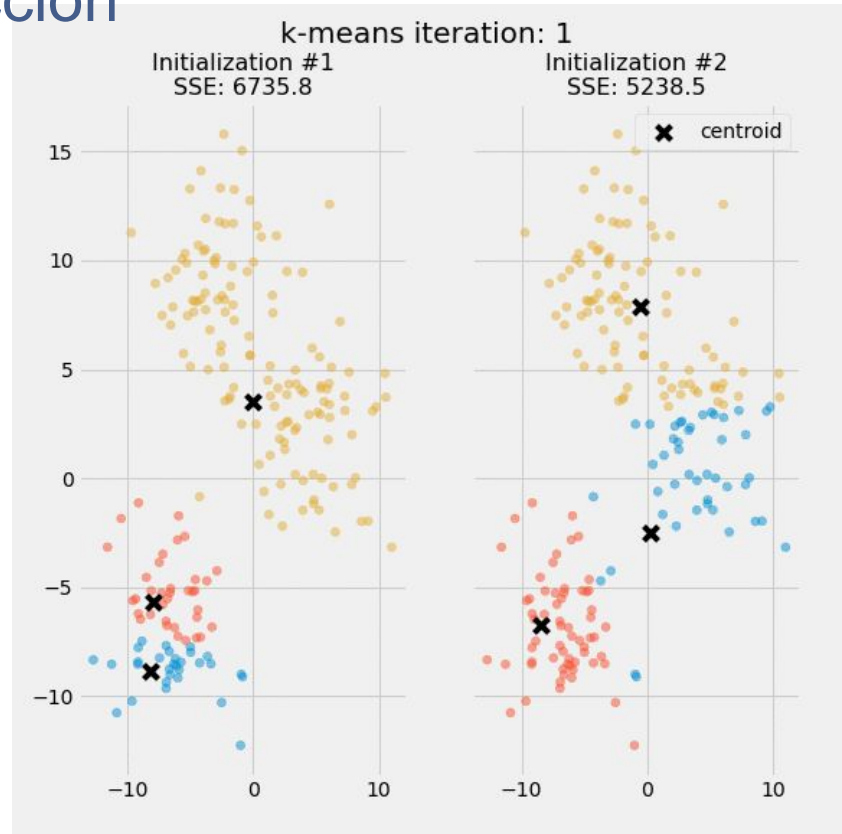
Básicamente K-means es un algoritmo de optimización de esta función objetivo
Depende de la inicialización -> **modelo no determinista**

Elige alguna de estas
4 inicializaciones




distintas inicializaciones del mismo modelo con los mismos datos ‘n_init’

# K-means en acción



# K-means: Función objetivo

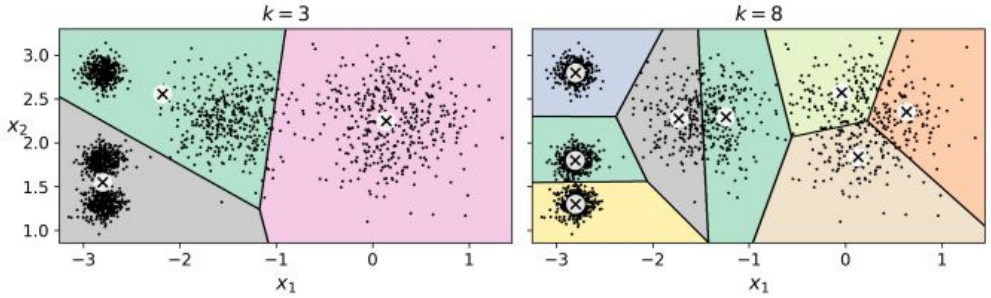
Una solución posible es iterar muchas veces el algoritmo con diferentes inicializaciones conservando el clustering que minimiza el valor de la función objetivo.

# K-means: Elección de k

No es trivial elegir ***k*** en la mayoría de los dataset reales. No hay algún método que funcione siempre.

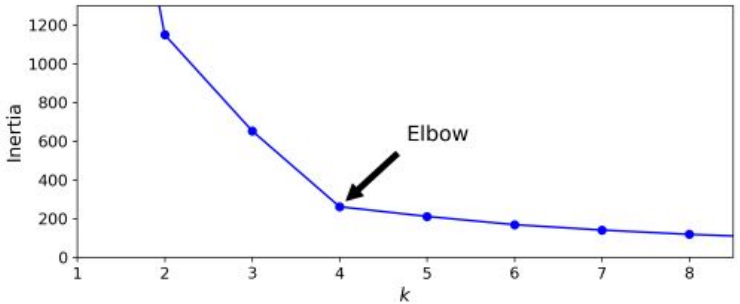
# K-means: Elección de k

En general desconocemos cuántos clusters hay. En el ejemplo anterior:




El valor de la inercia no es una medida adecuada: tiende a bajar cuanto más grande es $k$.

Método gráfico basado en comportamiento de la inercia con K: en general la cantidad de clusters óptima se encuentra cerca del codo (cambio de comportamiento).



# K-means: Elección de k

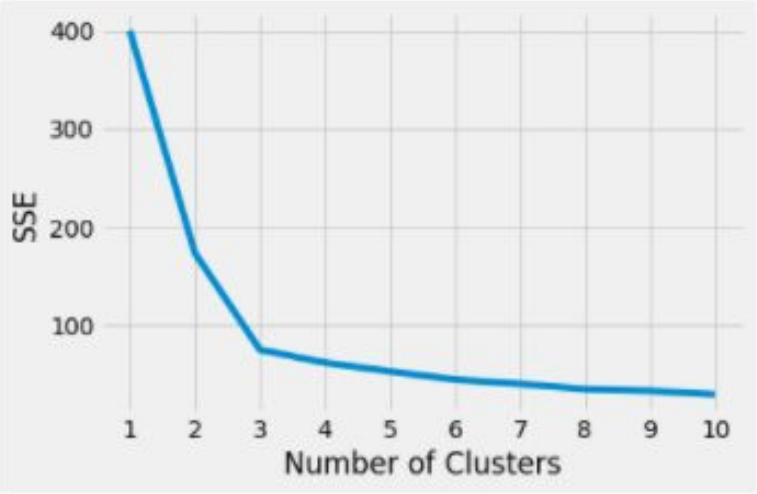
No es trivial elegir ***k*** en la mayoría de los dataset reales. No hay algún método que funcione siempre.

“Método del codo”

# K-means: Elección de k

No es trivial elegir ***k*** en la mayoría de los dataset reales. No hay algún método que funcione siempre.

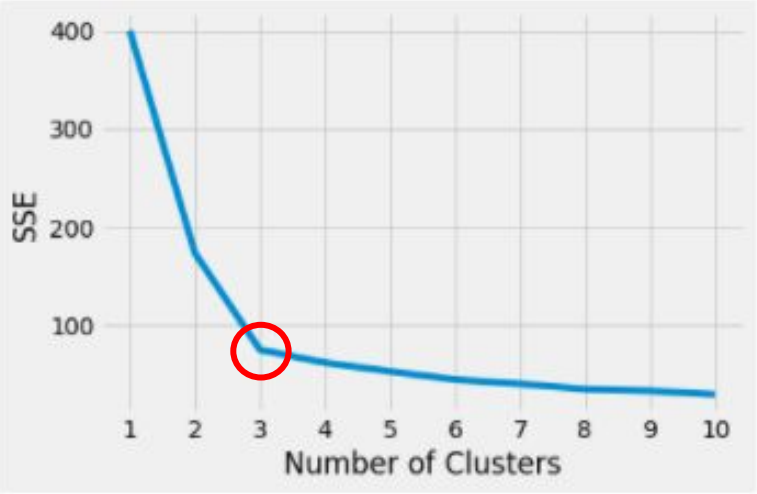
"Método del codo"



# K-means: Elección de k

No es trivial elegir ***k*** en la mayoría de los dataset reales. No hay algún método que funcione siempre.

"Método del codo"



# K-means: Elección de k

No es trivial elegir ***k*** en la mayoría de los dataset reales. No hay algún método que funcione siempre.

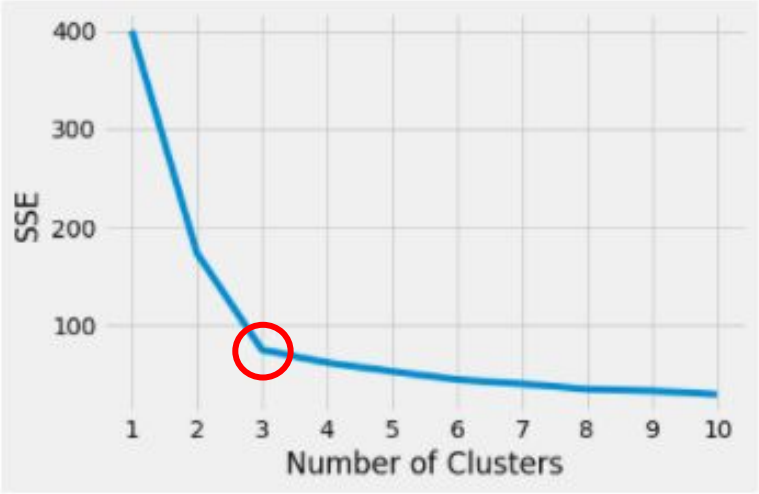
“Método del codo”

“coeficiente de Silhouette”

# K-means: Elección de k

No es trivial elegir ***k*** en la mayoría de los dataset reales. No hay algún método que funcione siempre.

“Método del codo”



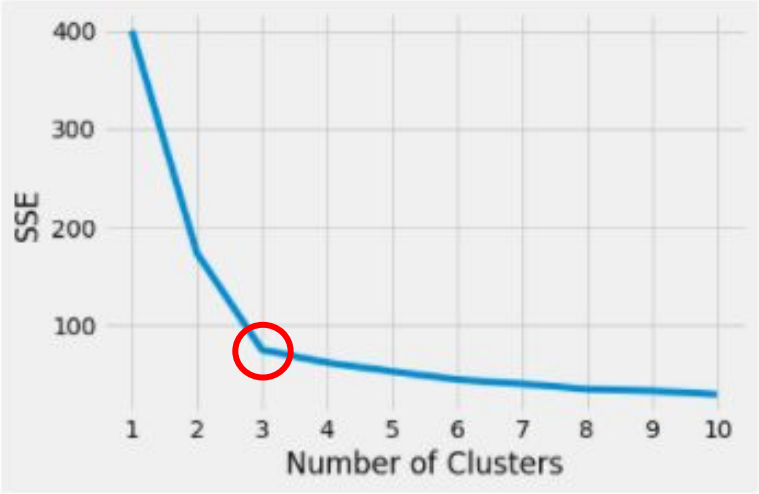

“coeficiente de Silhouette”

medida de cuán similar es un dado dato a los datos de su cluster en comparación a los datos del cluster más cercano

# K-means: Elección de k

No es trivial elegir ***k*** en la mayoría de los dataset reales. No hay algún método que funcione siempre.

"Método del codo"




"coeficiente de Silhouette"

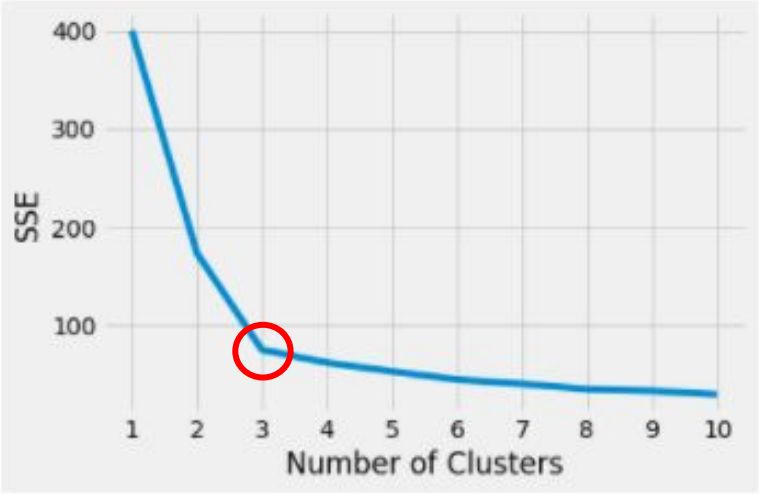
medida de cuán similar es un dado dato a los datos de su cluster en comparación a los datos del cluster más cercano

su valor va [-1,1]

# K-means: Elección de k

No es trivial elegir ***k*** en la mayoría de los dataset reales. No hay algún método que funcione siempre.

"Método del codo"




"coeficiente de Silhouette"

medida de cuán similar es un dado dato a los datos de su cluster en comparación a los datos del cluster más cercano

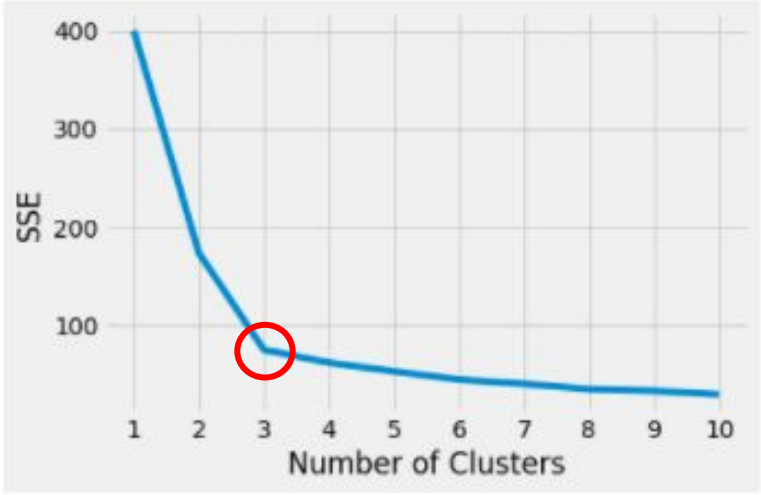
su valor va [-1,1]

1 indica que el dato está bien emparejado en su propio cluster y mal emparejado con los datos de otros clusters

# K-means: Elección de k

No es trivial elegir ***k*** en la mayoría de los dataset reales. No hay algún método que funcione siempre.

"Método del codo"




"coeficiente de Silhouette"

medida de cuán similar es un dado dato a los datos de su cluster en comparación a los datos del cluster más cercano

su valor va [-1,1]

1 indica que el dato está bien emparejado en su propio cluster y mal emparejado con los datos de otros clusters

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_i| > 1$$

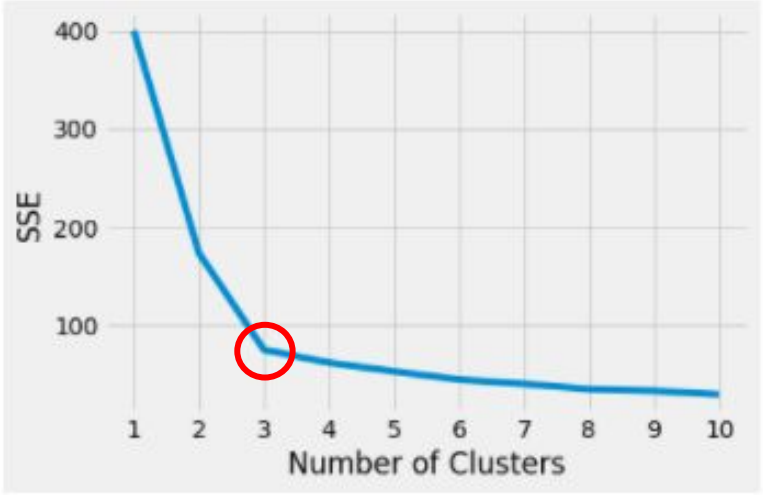
$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j)$$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

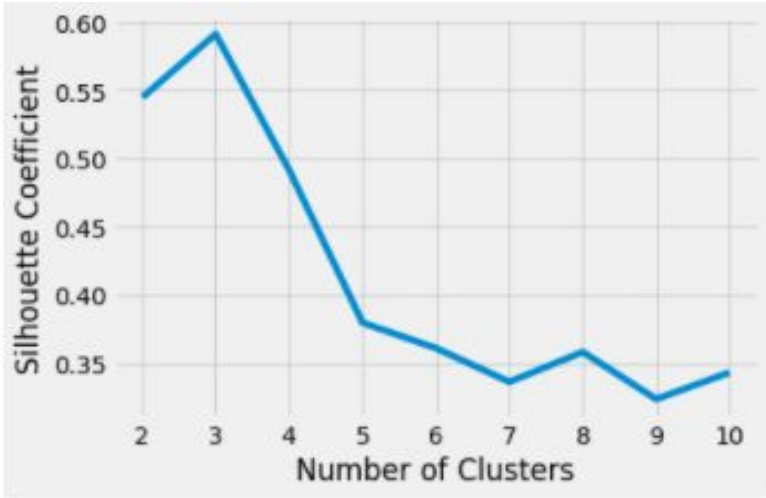
# K-means: Elección de k

No es trivial elegir ***k*** en la mayoría de los dataset reales. No hay algún método que funcione siempre.

“Método del codo”



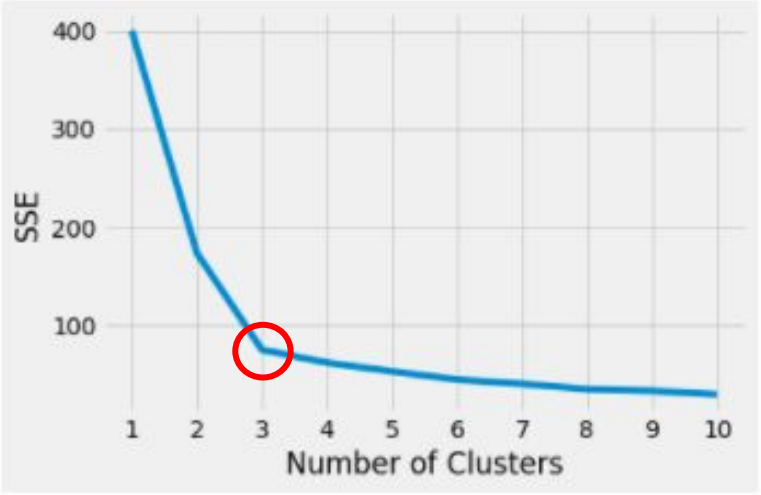

“coeficiente de Silhouette”



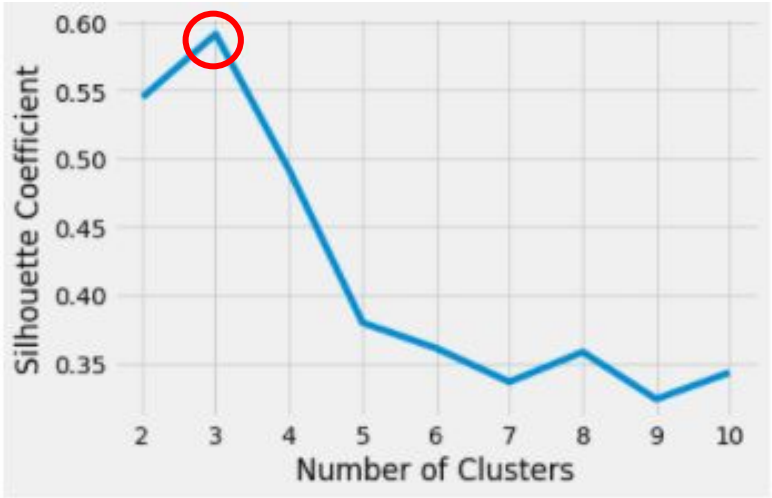
# K-means: Elección de k

No es trivial elegir ***k*** en la mayoría de los dataset reales. No hay algún método que funcione siempre.

“Método del codo”

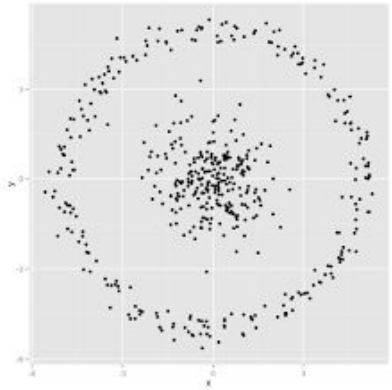


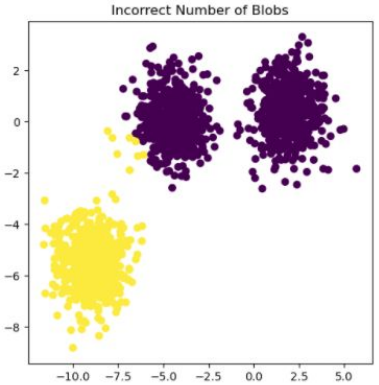

“coeficiente de Silhouette”

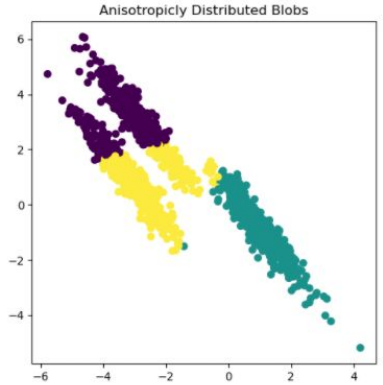


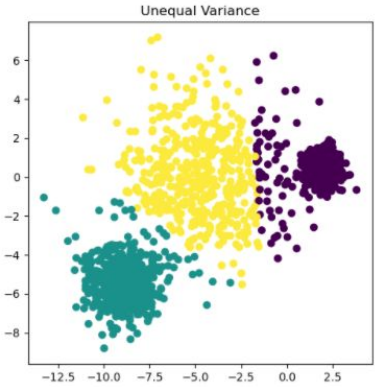
# K-means: Pros y Cons

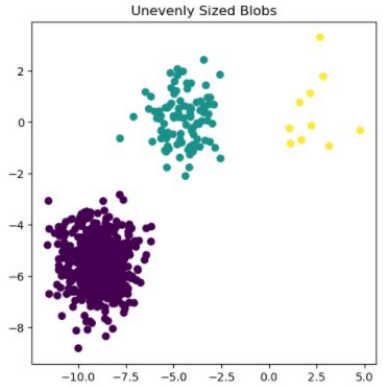
+ Simple y Fácil de implementar
+ Orden del algoritmo es lineal
- Depende de la inicialización
- Tiende a caer en un mínimo local
- Sensible a outliers
- Los clusters tienen que tener forma esférica
- No se puede aplicar a datos categóricos





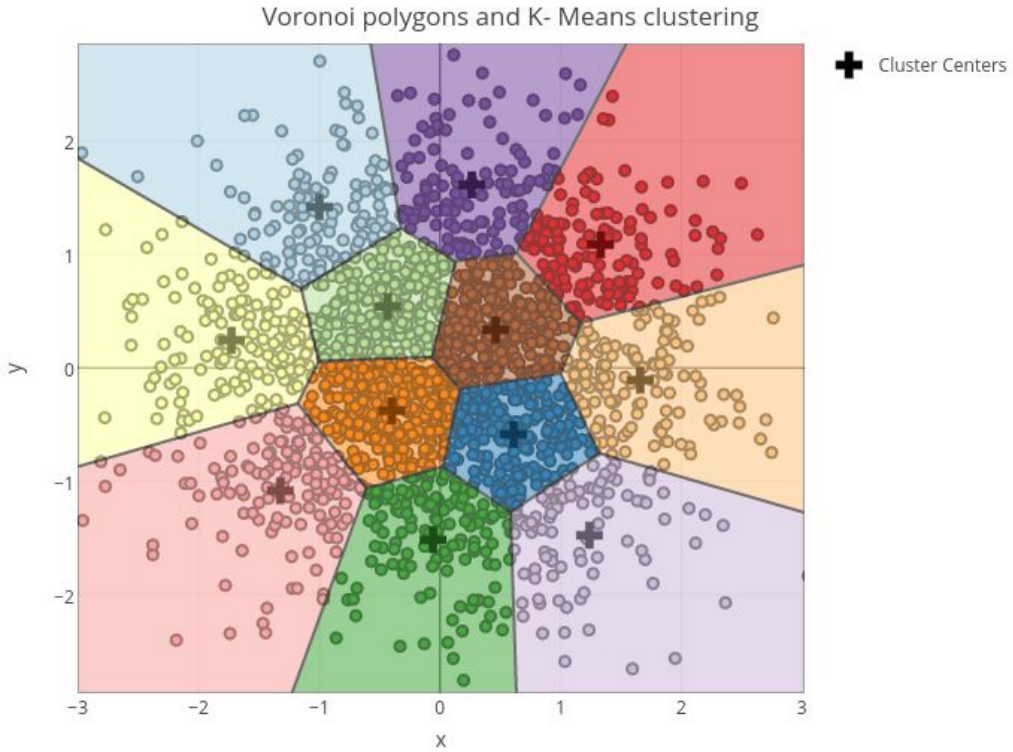



# K-means: Pros y Cons

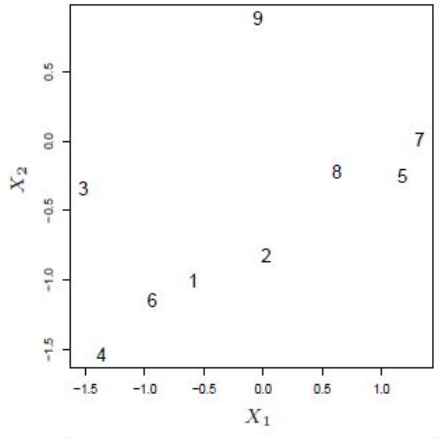
- Da un clustering de los datos aún si los datos no están “clusterizados”



# Clustering Jerárquico: Dendrograma

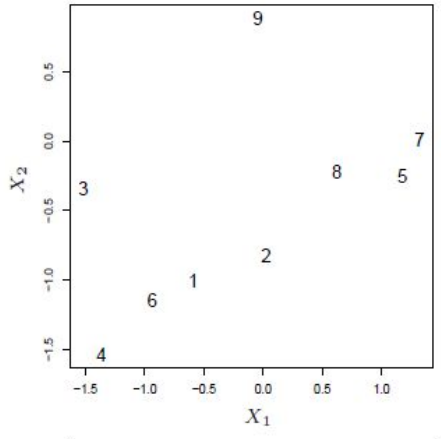
# Clustering Jerárquico: Dendrograma

**n** samples
**n** clusters



# Clustering Jerárquico: Dendrograma

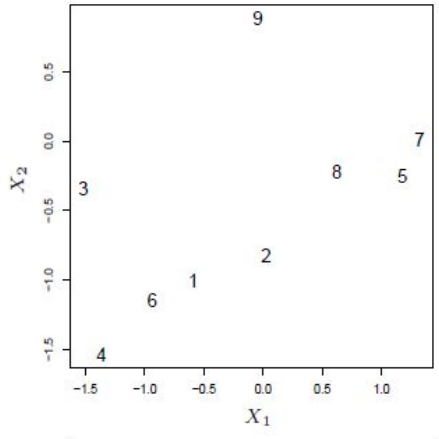
**n** samples
**n** clusters




medida de distancia entre samples (‘affinity’), usualmente la euclídea

# Clustering Jerárquico: Dendrograma

**n** samples
**n** clusters

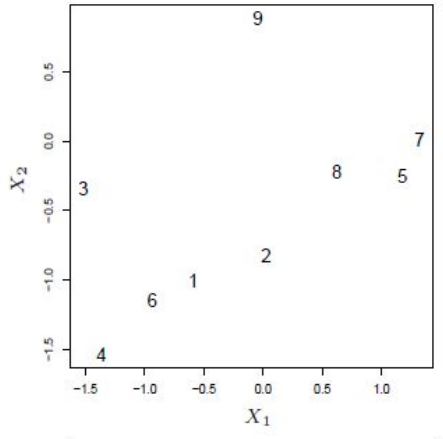



medida de distancia entre samples (‘affinity’), usualmente la euclídea

junto los dos cluster que están a menor distancia

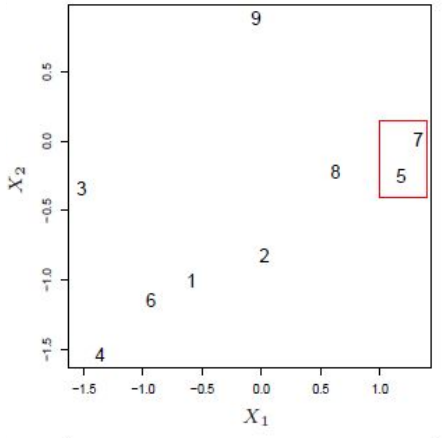
# Clustering Jerárquico: Dendrograma

**n** samples
**n** clusters




medida de distancia entre samples (‘affinity’), usualmente la euclídea

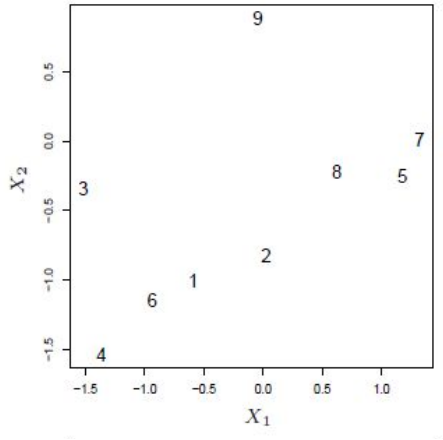
junto los dos cluster que están a menor distancia




**n-1** clusters

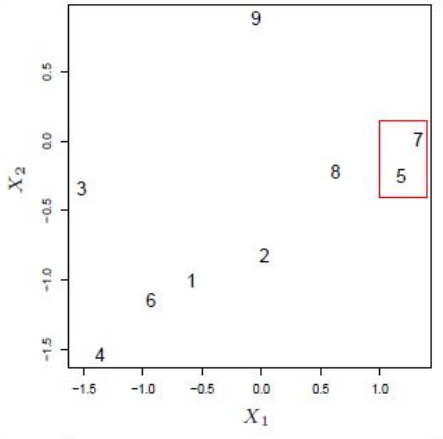
# Clustering Jerárquico: Dendrograma

**n** samples
**n** clusters




medida de distancia entre samples (‘affinity’), usualmente la euclídea

junto los dos cluster que están a menor distancia

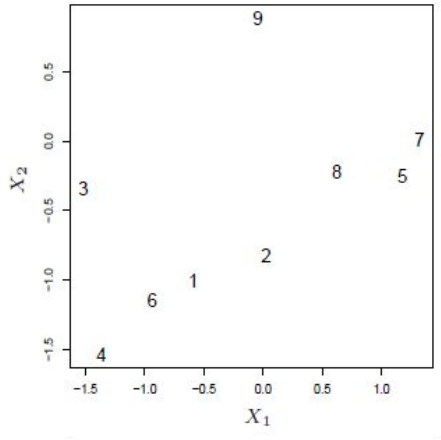



**n-1** clusters

distancia entre clusters de >=1 elementos (‘linkage’)

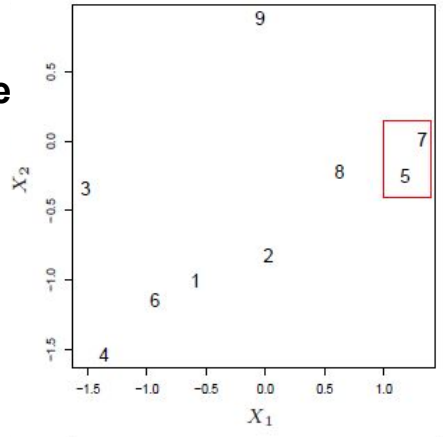
# Clustering Jerárquico: Dendrograma

**n** samples
**n** clusters




medida de **distancia entre samples** (‘affinity’), usualmente la euclídea

junto los dos cluster que están a menor distancia

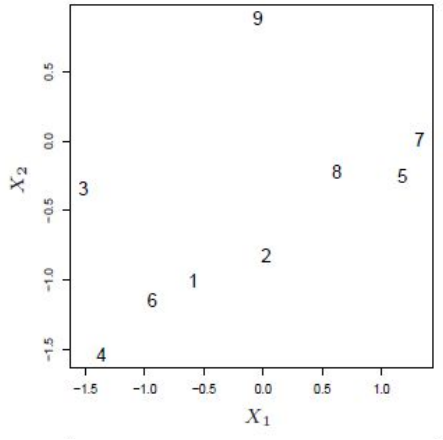



**n-1** clusters

**distancia entre clusters** de >=1 elementos (‘linkage’)

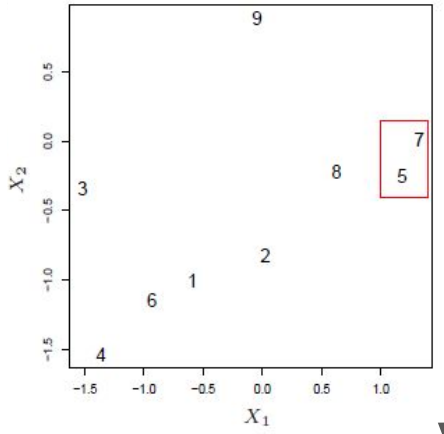
# Clustering Jerárquico: Dendrograma

**n** samples
**n** clusters



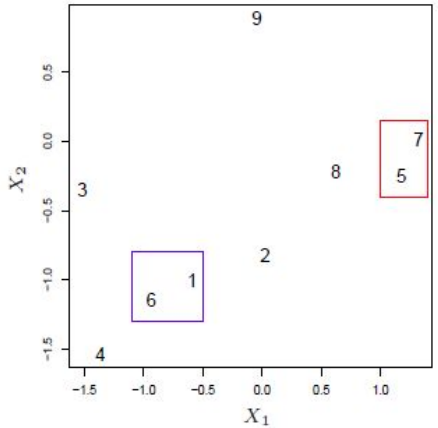

medida de distancia entre samples (‘affinity’), usualmente la euclídea

junto los dos cluster que están a menor distancia



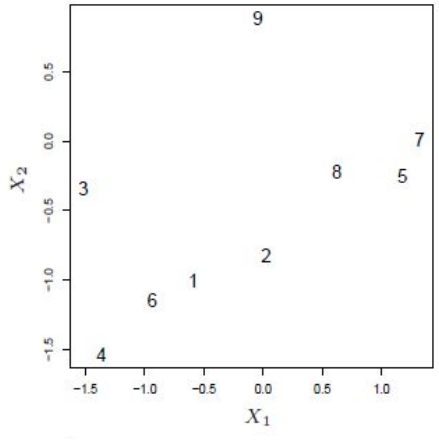

**n-1** clusters

distancia entre clusters de >=1 elementos (‘linkage’)



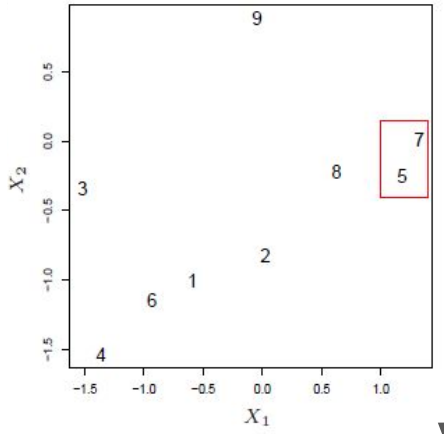
# Clustering Jerárquico: Dendrograma

**n** samples
**n** clusters



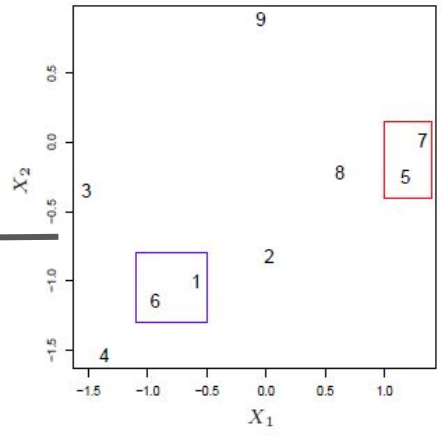

medida de distancia entre samples (‘affinity’), usualmente la euclídea

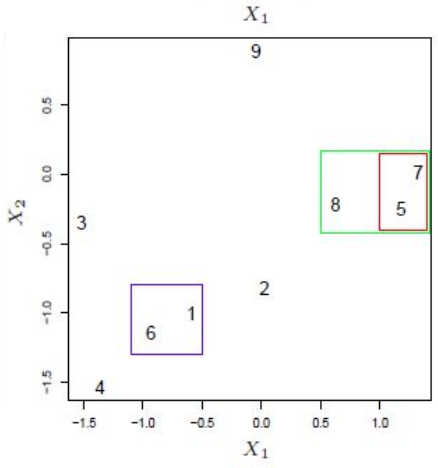
junto los dos cluster que están a menor distancia




**n-1** clusters

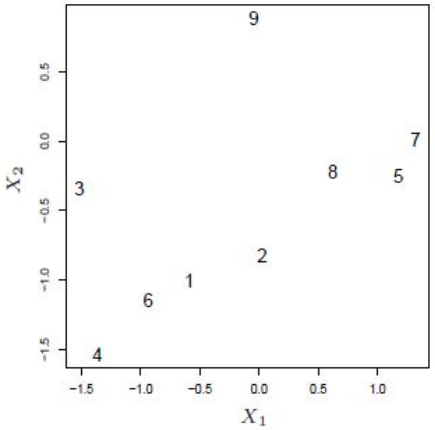
distancia entre clusters de >=1 elementos (‘linkage’)



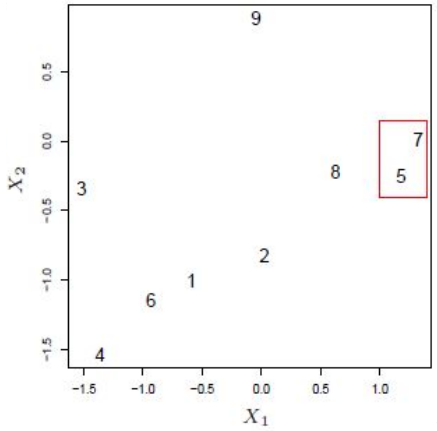
# Clustering Jerárquico: Dendrograma

**n** samples
**n** clusters



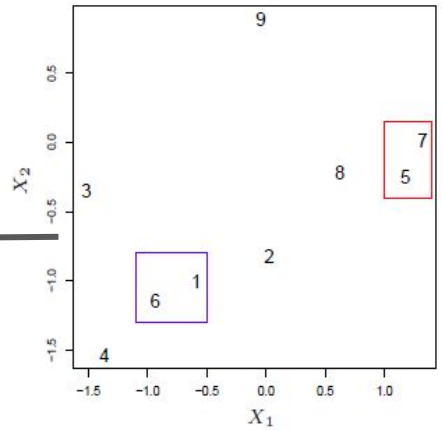

medida de distancia entre samples (‘affinity’), usualmente la euclídea

junto los dos cluster que están a menor distancia

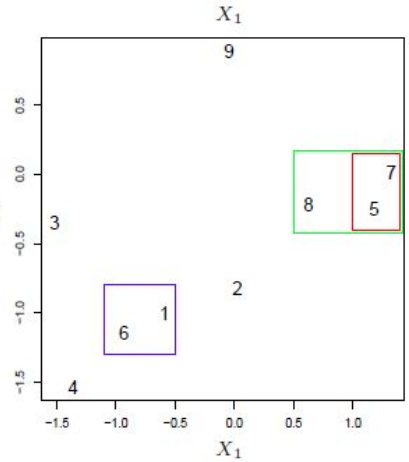


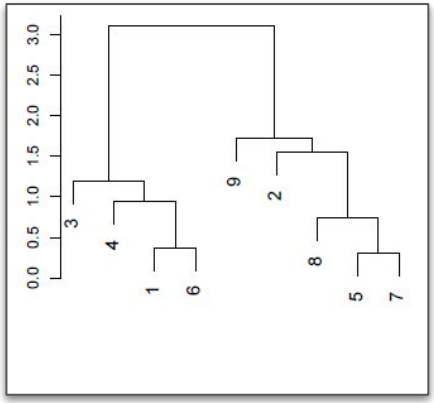

**n-1** clusters

distancia entre clusters de >=1 elementos (‘linkage’)



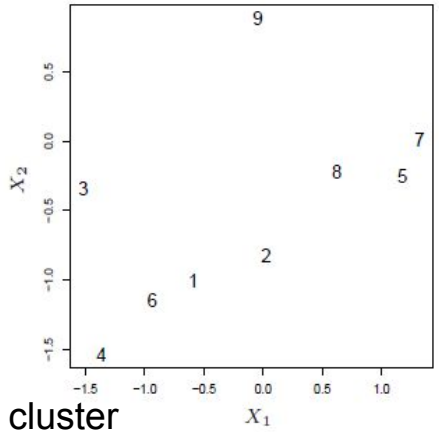

(...)



# Clustering Jerárquico: Dendrograma

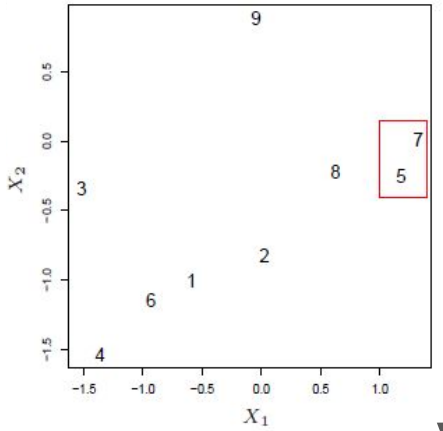
**n** samples
**n** clusters




**1** cluster

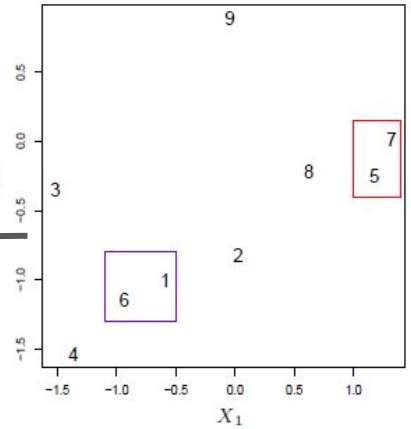
medida de distancia entre samples (‘affinity’), usualmente la euclídea

junto los dos cluster que están a menor distancia

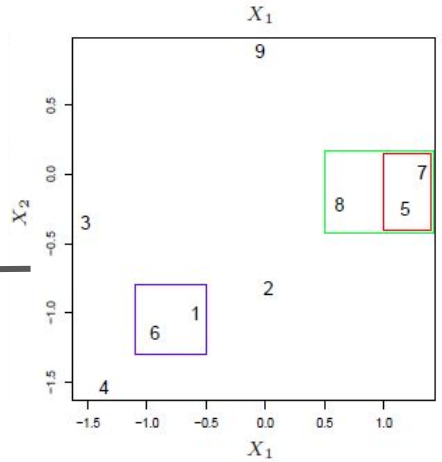


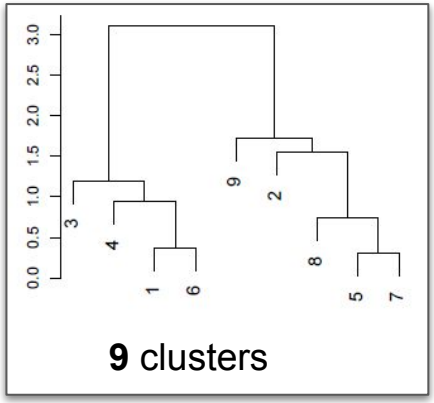

**n-1** clusters

distancia entre clusters de >=1 elementos (‘linkage’)




(...)

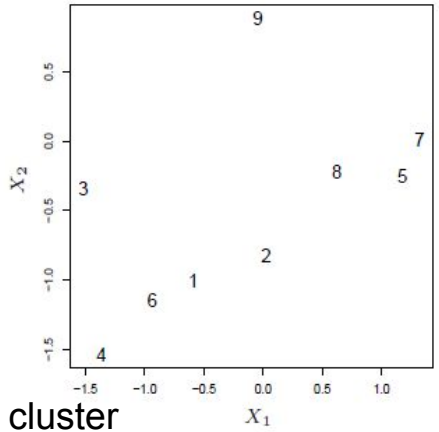



**9** clusters

# Clustering Jerárquico: Dendrograma

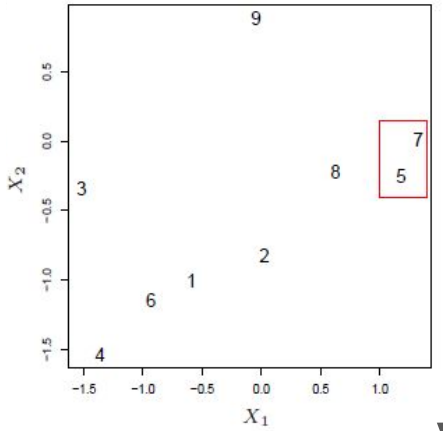
**n** samples
**n** clusters




**1** cluster

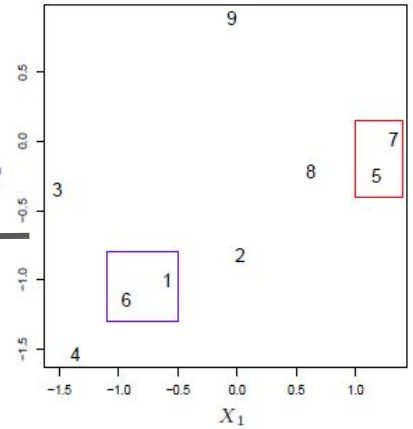
medida de distancia entre samples ('affinity'), usualmente la euclídea

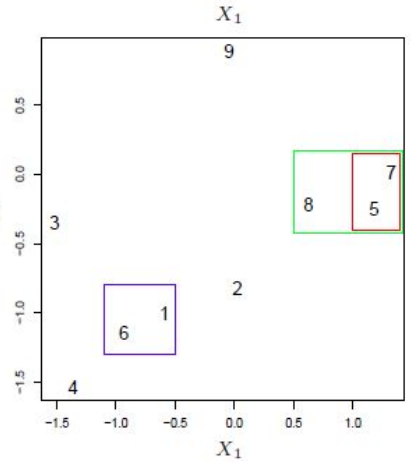
junto los dos cluster que están a menor distancia



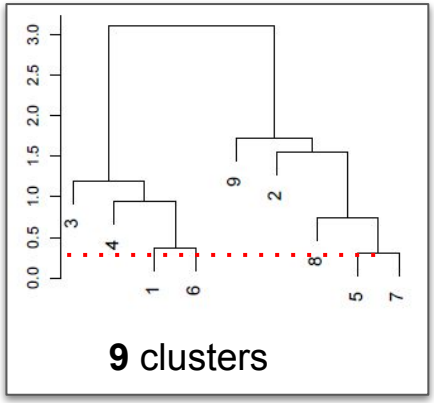

**n-1** clusters

distancia entre clusters de >=1 elementos ('linkage')



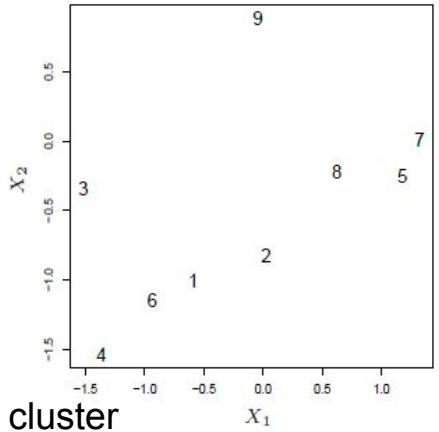

(...)




**9** clusters

# Clustering Jerárquico: Dendrograma

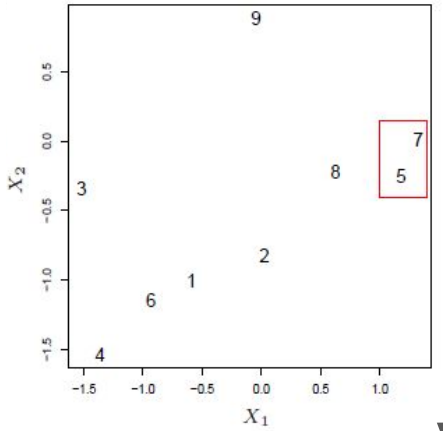
**n** samples
**n** clusters




**1** cluster

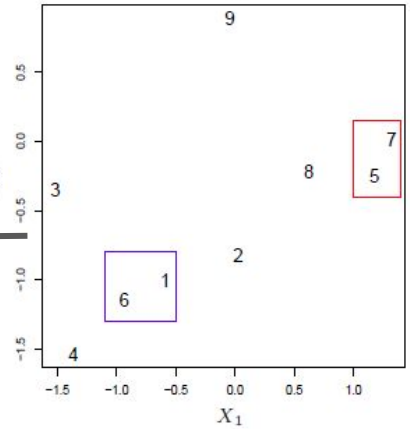
medida de distancia entre samples (‘affinity’), usualmente la euclídea

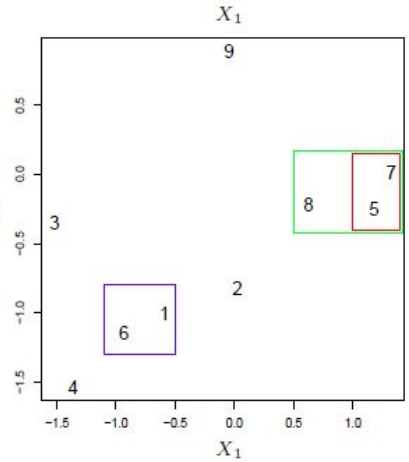
junto los dos cluster que están a menor distancia



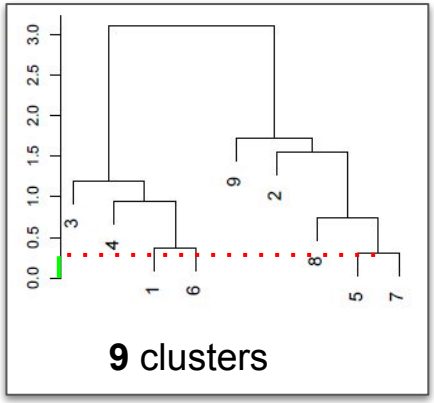

**n-1** clusters

distancia entre clusters de >=1 elementos (‘linkage’)



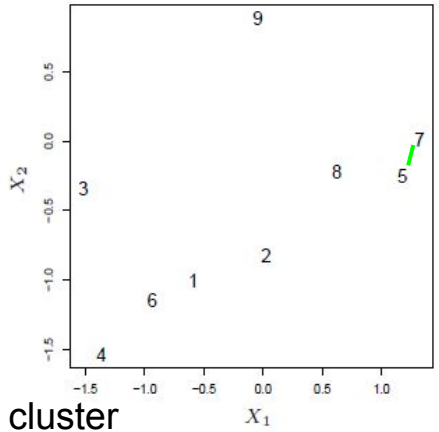

(...)




**9** clusters

# Clustering Jerárquico: Dendrograma

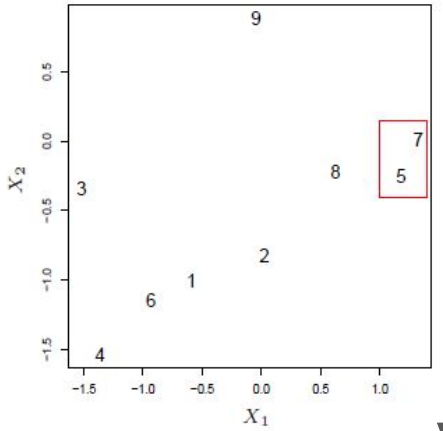
**n** samples
**n** clusters




**1** cluster

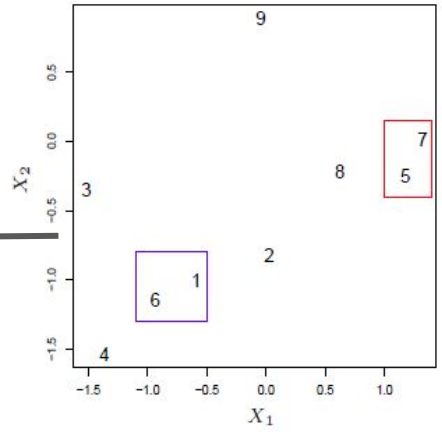
medida de distancia entre samples (‘affinity’), usualmente la euclídea

junto los dos cluster que están a menor distancia

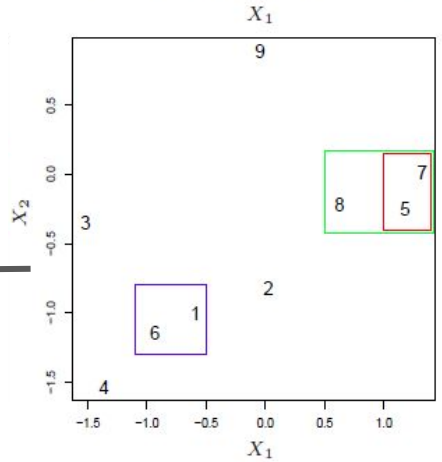


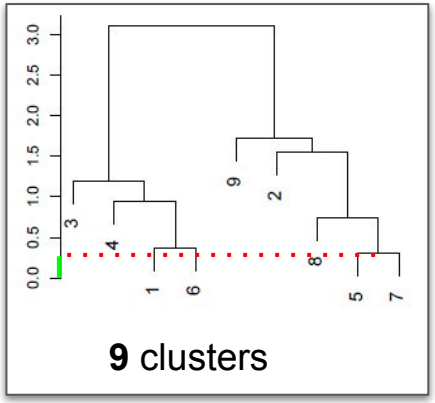

**n-1** clusters

distancia entre clusters de >=1 elementos (‘linkage’)




(...)

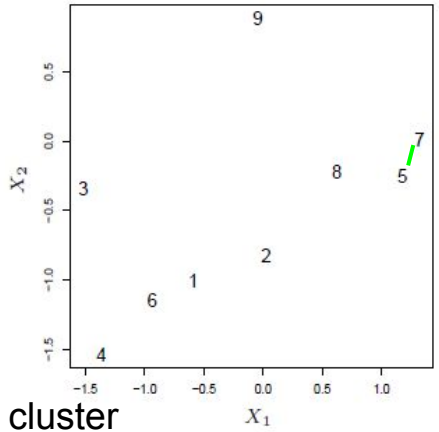



**9** clusters

# Clustering Jerárquico: Dendrograma

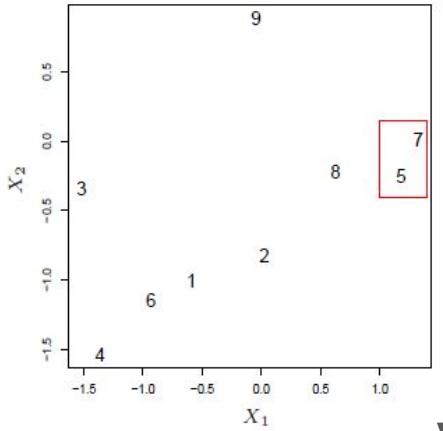
**n** samples
**n** clusters




**1** cluster

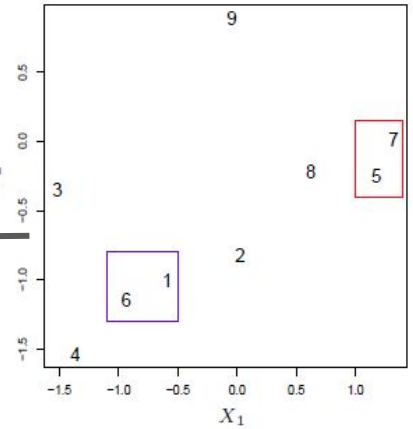
medida de distancia entre samples ('affinity'), usualmente la euclídea

junto los dos cluster que están a menor distancia

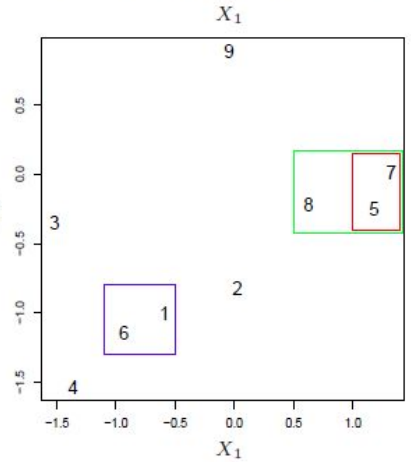



**n-1** clusters

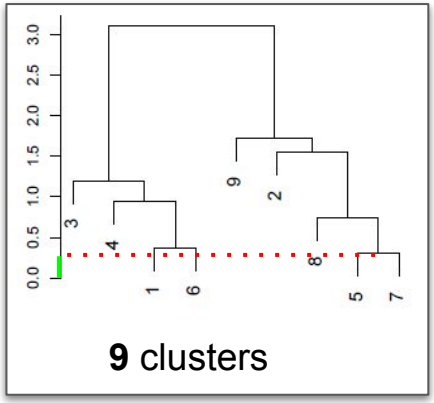
distancia entre clusters de >=1 elementos ('linkage')




(...)

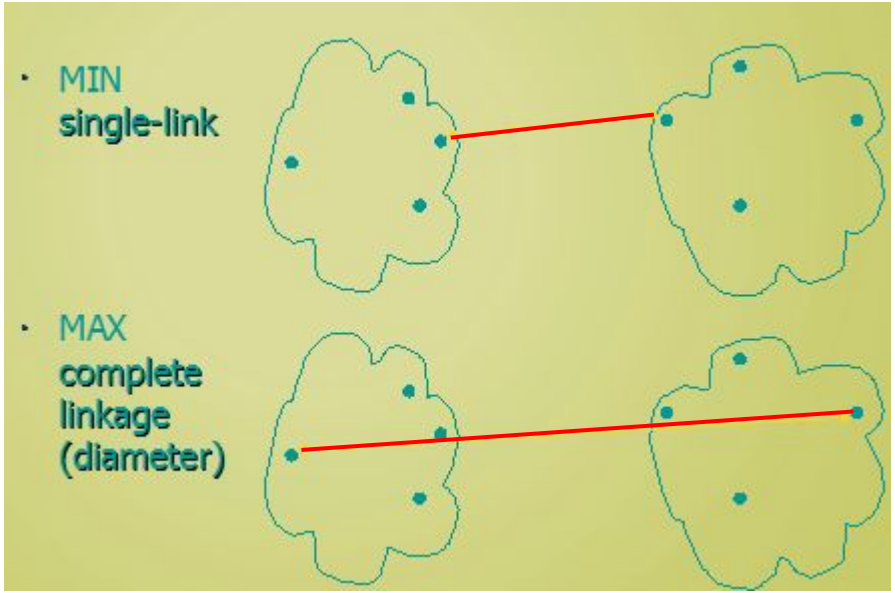


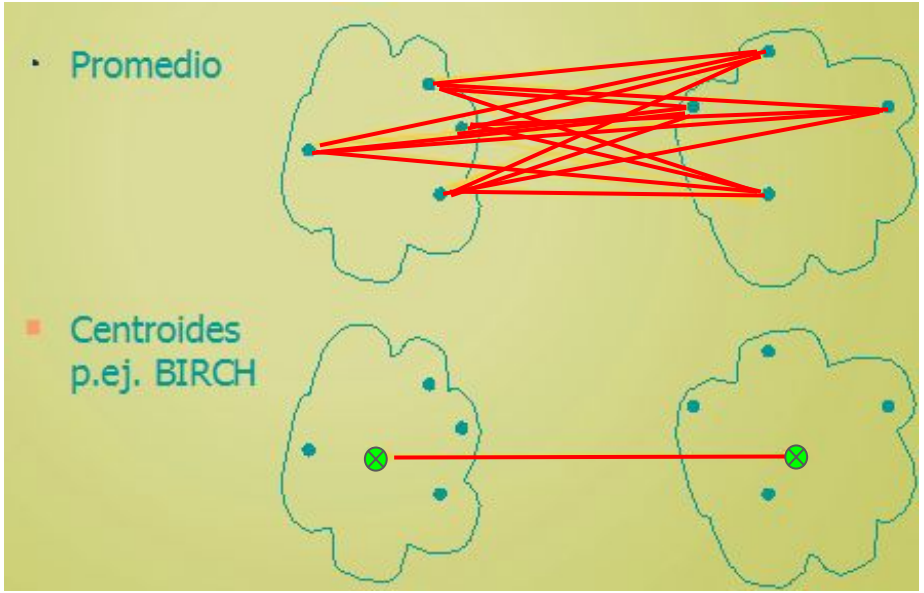

linkage




**9** clusters

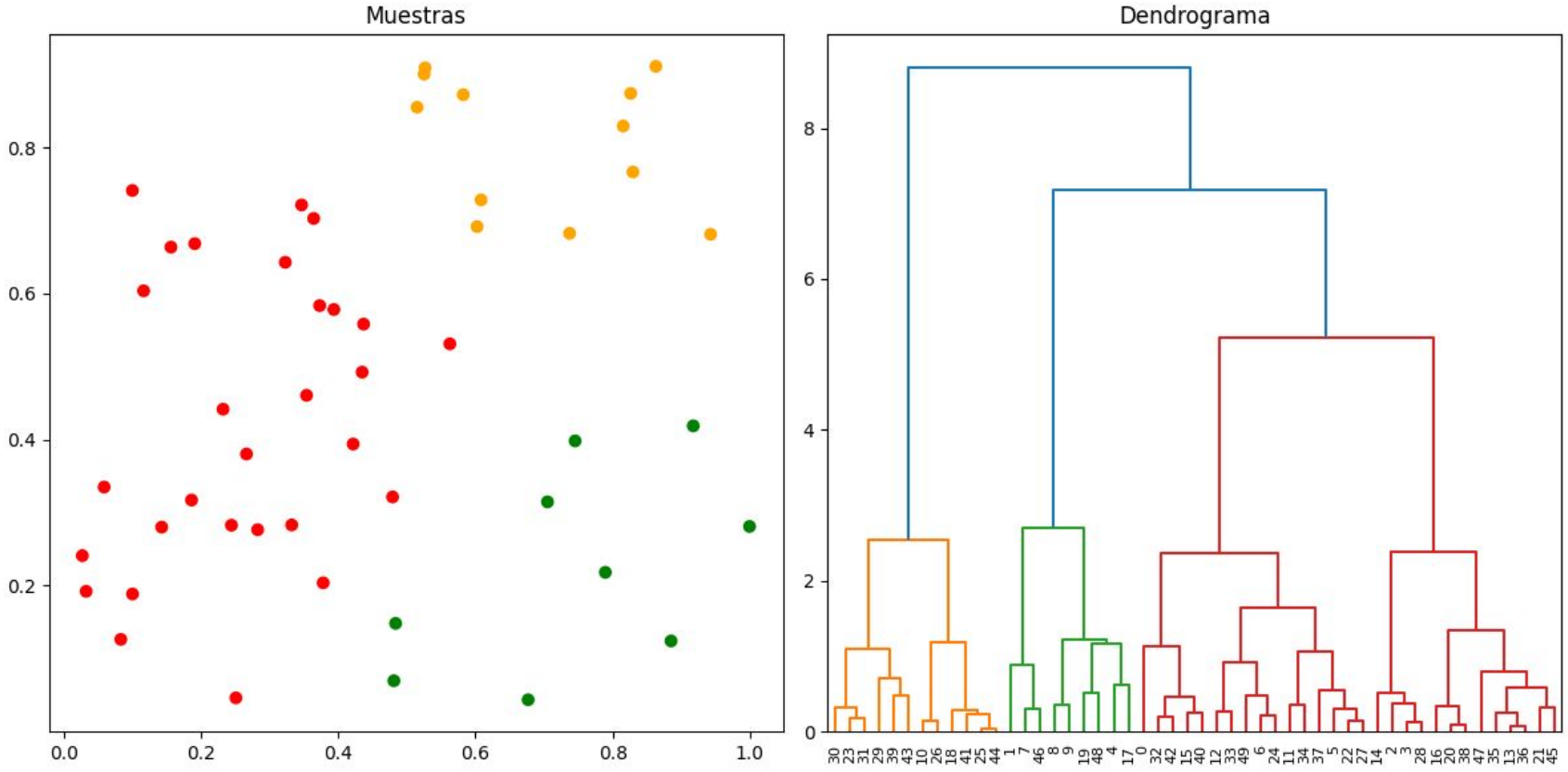
# Clustering Jerárquico: Linkage



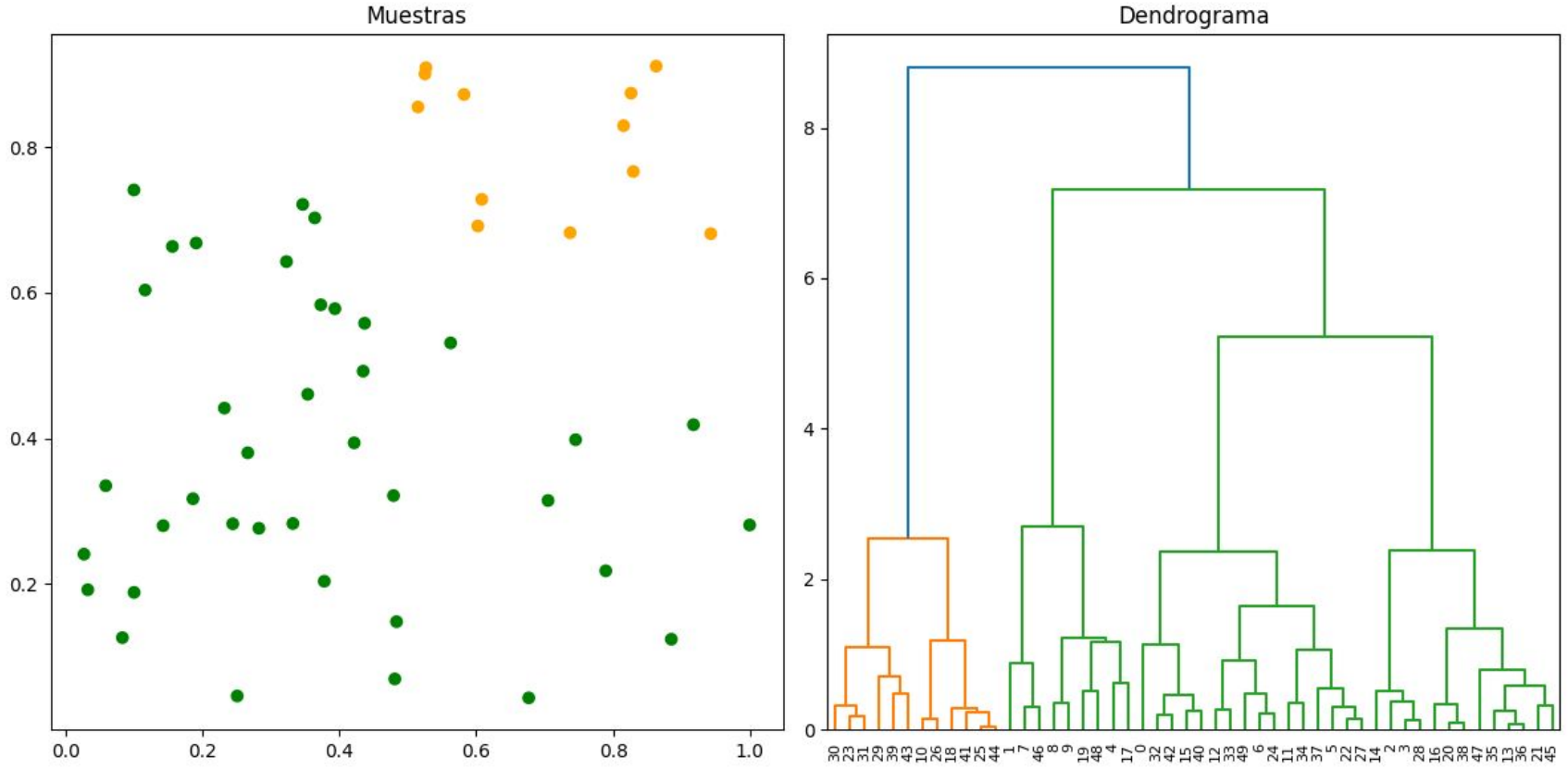




- **Ward linkage**: minimiza la varianza del cluster que se va a *mergear*

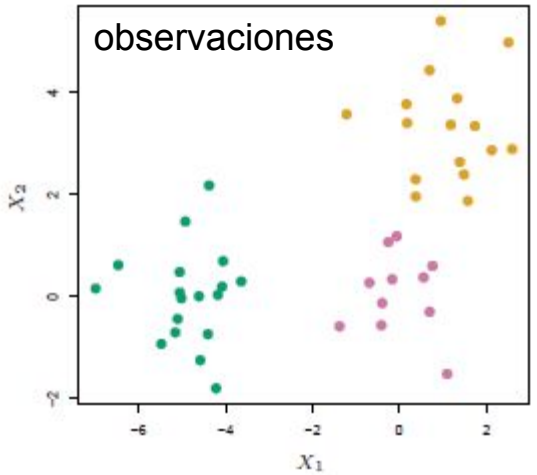
# Clustering Jerárquico: De vuelta al dendrograma

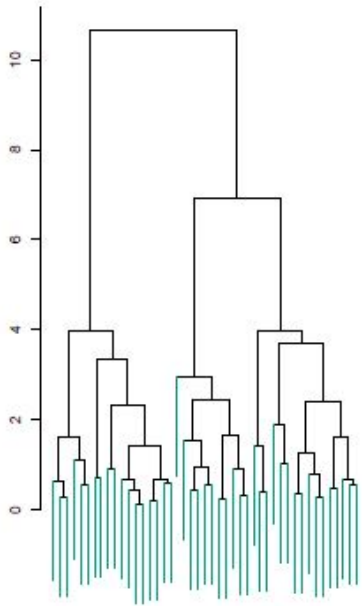


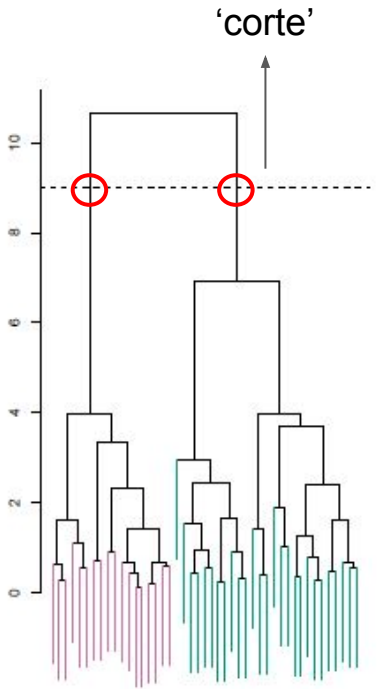
# Clustering Jerárquico: De vuelta al dendrograma



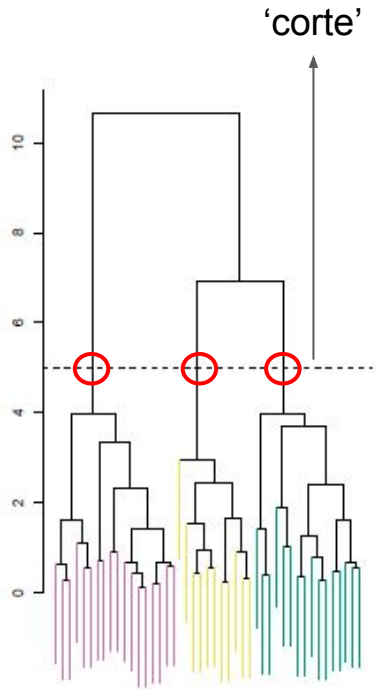
# Clustering Jerárquico: De vuelta al dendrograma



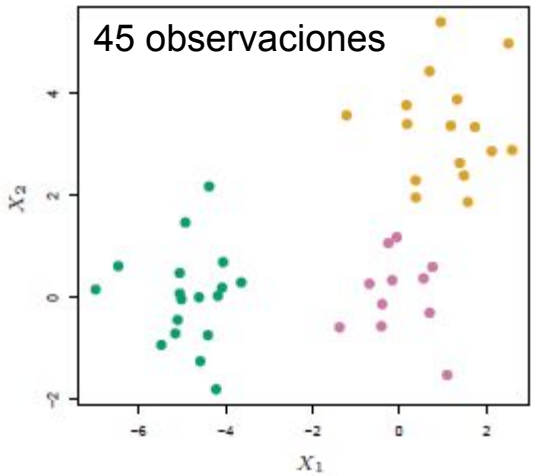



**dos** clusters

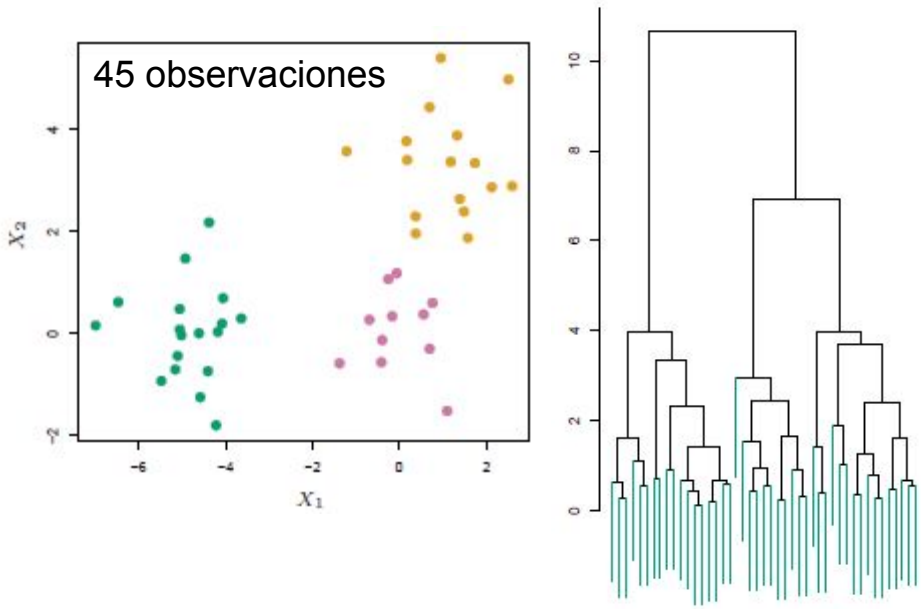



**tres** clusters

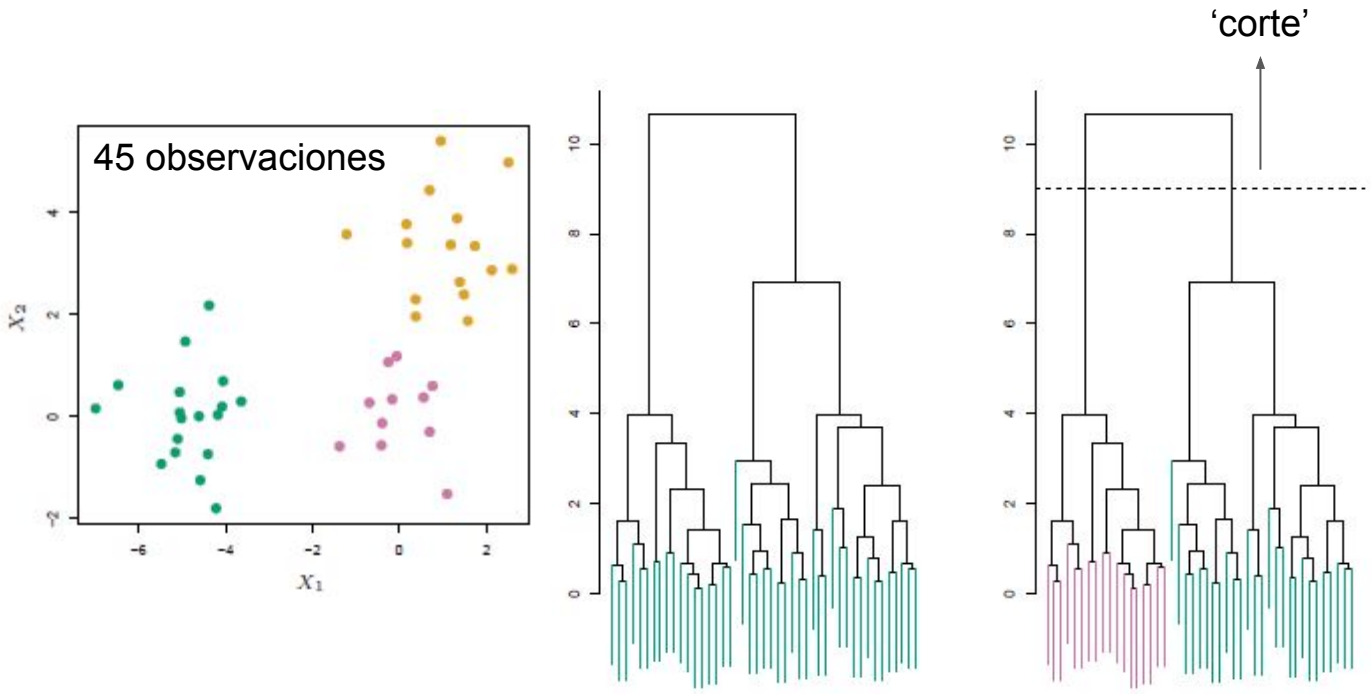
# Clustering Jerárquico: De vuelta al dendrograma



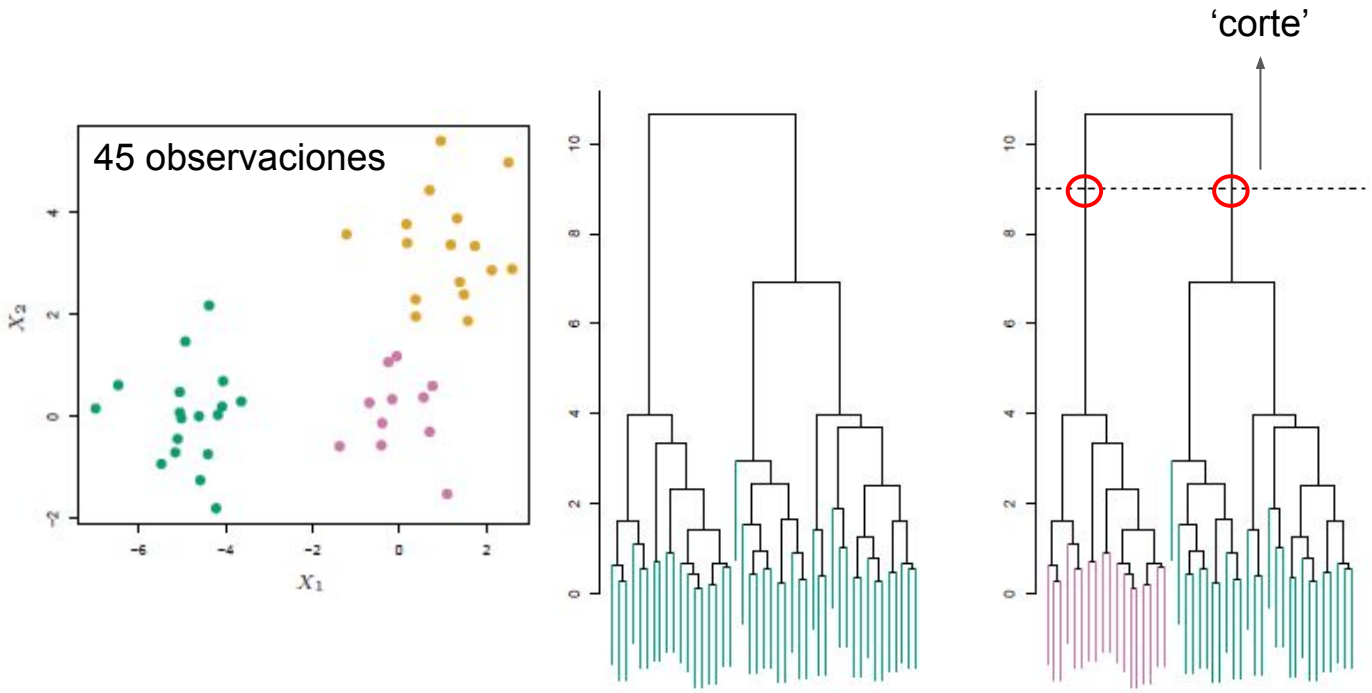
# Clustering Jerárquico: De vuelta al dendrograma



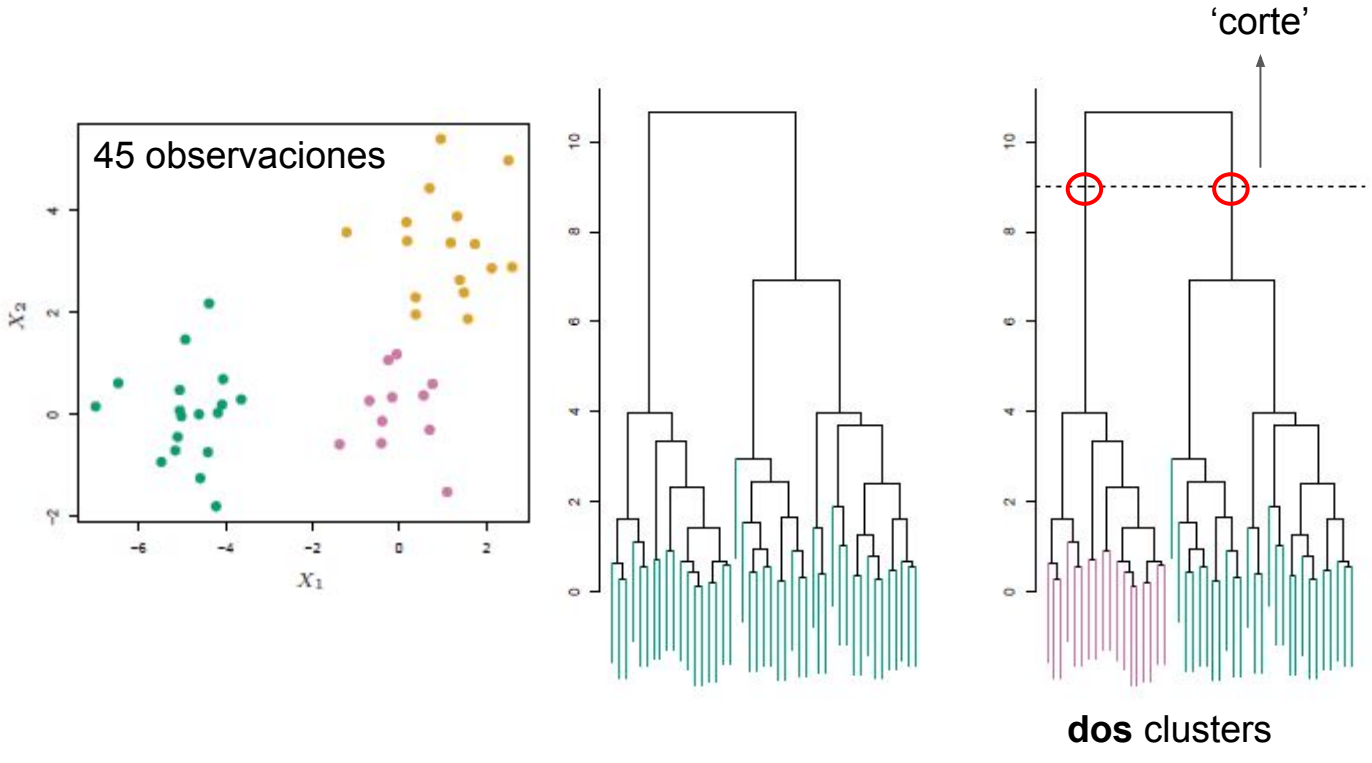
# Clustering Jerárquico: De vuelta al dendrograma



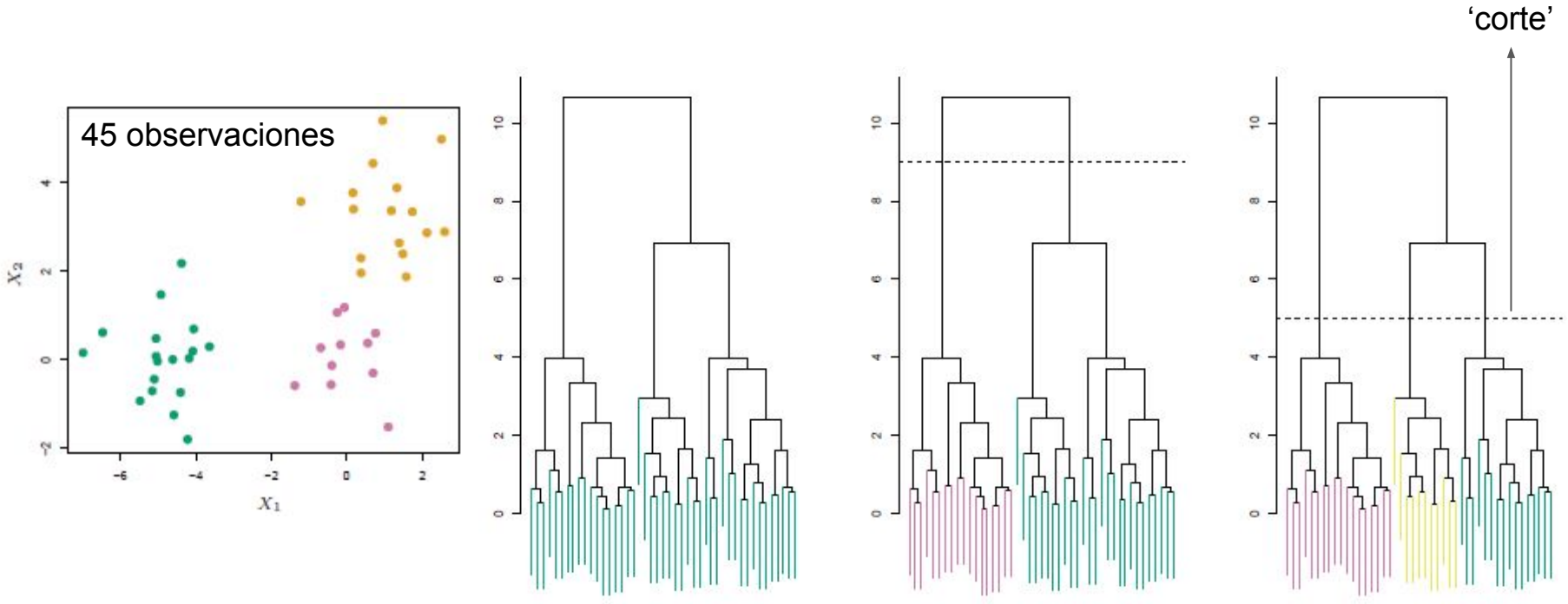
# Clustering Jerárquico: De vuelta al dendrograma



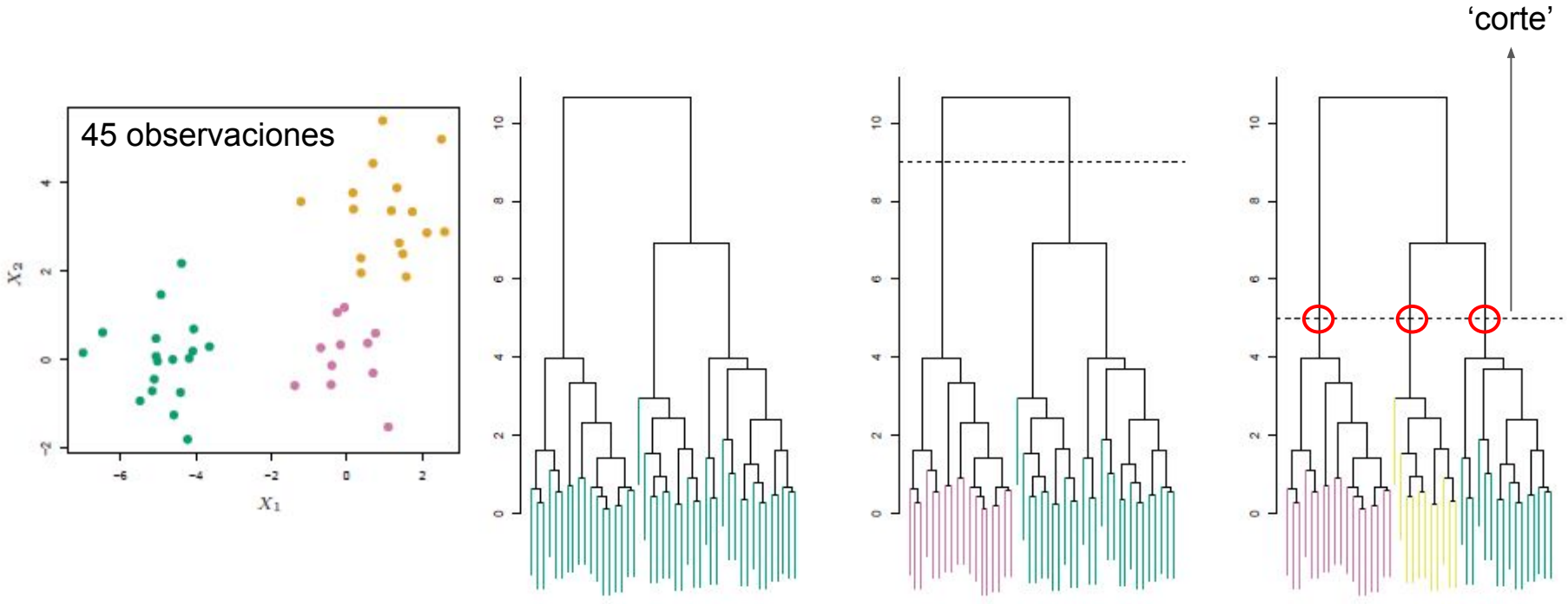
# Clustering Jerárquico: De vuelta al dendrograma



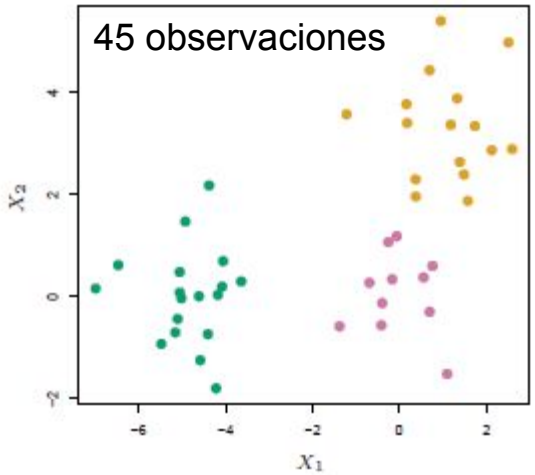
# Clustering Jerárquico: De vuelta al dendrograma

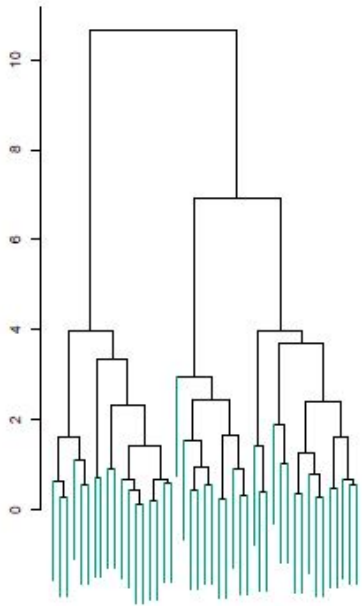


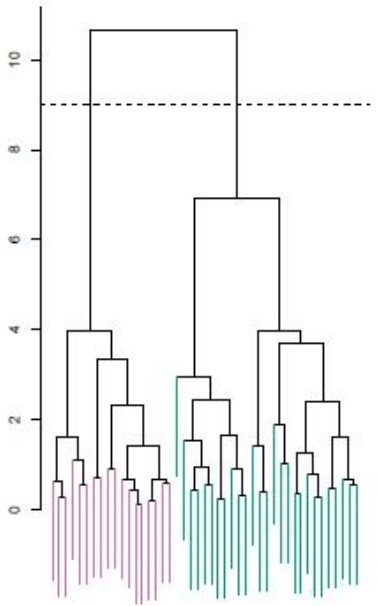
# Clustering Jerárquico: De vuelta al dendrograma

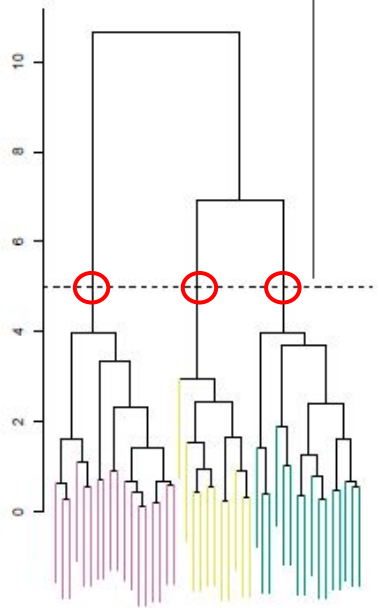


# Clustering Jerárquico: De vuelta al dendrograma



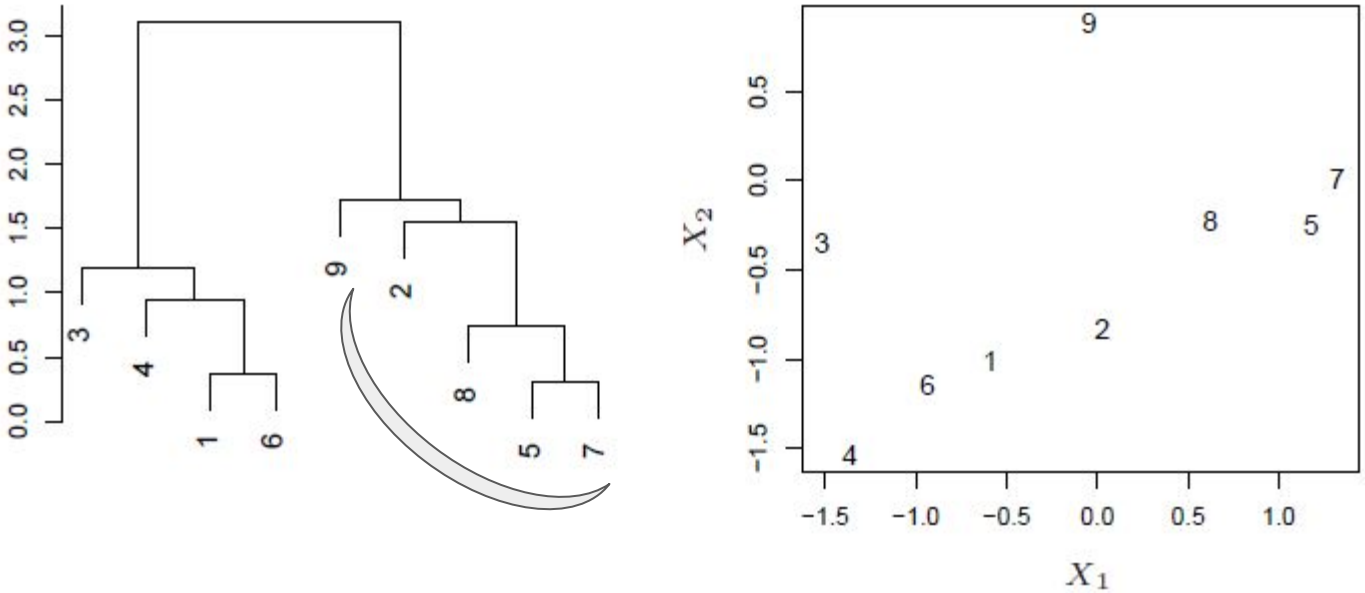








**tres** clusters

# Clustering Jerárquico: De vuelta al dendrograma

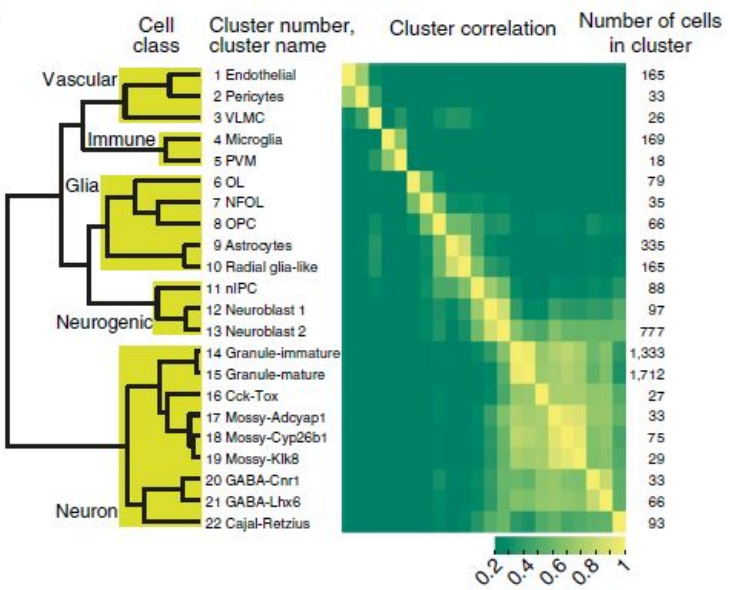



Los clusters que se obtienen cortando el dendograma a una dada altura están anidados con los clusters que se obtienen al cortar el dendograma en una altura superior.

Este puede ser un requisito fuerte para un dataset real donde no necesariamente la mejor clusterización de los datos en 3 grupos resulta de tomar la mejor clusterización de los datos en dos grupos y separarlos

# Clustering Jerárquico: Pros y Cons

+ Pueden revelar detalles finos en la relación de los datos
+ Proveen un dendrograma interpretable
+ Son determinísticos - producen el mismo resultado si se corre el mismo modelo con el mismo input
- Son computacionalmente costosos



# Consideraciones

- Tanto K-means como clustering jerárquico asignan un cluster a cada sample, por lo que los clusters encontrados pueden **distorsionarse fuertemente debido a la presencia de outliers** que no pertenecen a algún determinado grupo -> modelos mixtos están pensados para lidiar con outliers.
- Los algoritmos de clusterización **no suelen ser robustos ante permutaciones en los datos**. Si computamos un set de k clusters en nuestros datos, removemos algunas muestras de forma aleatoria y volvemos a clusterizar los resultados pueden ser muy distintos.
- Los resultados de una clusterización **no deberían ser vistos como una verdad absoluta** sino que deberían dar un puntapié inicial al desarrollo de una hipótesis científica e investigación futura, preferentemente, en un dataset independiente.

# Sugerencias

- Realizar varias inicializaciones de los modelos cambiando los parámetros para ver las estructuras que emergen consistentemente

- Testear la robustez del método aplicándolo sobre subsets aleatorios de los datos

# (sklearn) K-means clustering

## sklearn.cluster.KMeans

*class* `sklearn.cluster.KMeans`(*n_clusters=8, *, init='k-means++', n_init=10, max_iter=300, tol=0.0001, precompute_distances='deprecated', verbose=0, random_state=None, copy_x=True, n_jobs='deprecated', algorithm='auto'*) [source]

El parámetro es *n_clusters* con el cual indicamos al modelo el número de clusters que queremos obtener

*init* - es el método de inicialización de los k centroids (random, k-means++)

*n_init* - cuantas veces queremos inicializar el modelo ya que no es determinista (output es el de menor SSE)

*max_iter* - el número máximo de iteraciones si es que no converge antes

# (sklearn) Clustering Jerárquico

## sklearn.cluster.AgglomerativeClustering

*class* `sklearn.cluster.` **`AgglomerativeClustering`**(*n_clusters=2, *, affinity='euclidean', memory=None, connectivity=None, compute_full_tree='auto', linkage='ward', distance_threshold=None, compute_distances=False*) [source]

*affinity* - medida de disimilaridad de para el primer paso del algoritmo (distancia euclídea o basada en correlación)

*linkage* - medida de disimilaridad cuando un cluster tiene más de un elemento

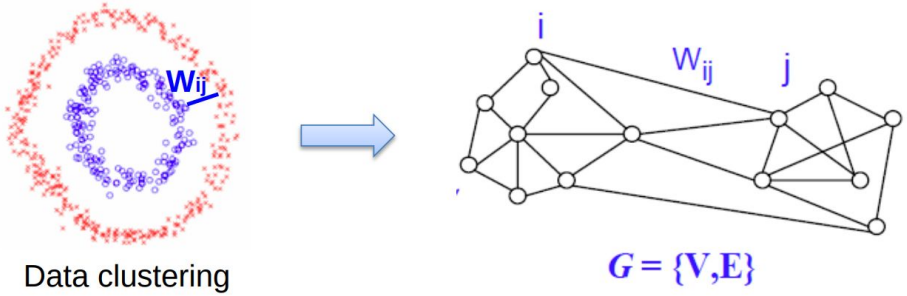
# Spectral clustering

## Similarity graph construction

Similarity Graphs: Model local neighborhood relations between data points

E.g. Gaussian kernel similarity function

$$W_{ij} = e^{\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$ → Controls size of neighborhood




Data clustering

# Spectral clustering

**¿Qué hace exactamente?**

1. **Construye un grafo**: cada punto de datos es un nodo, y las conexiones (aristas) se basan en la similitud entre puntos (por ejemplo, con una función de afinidad tipo gaussiana).
2. **Calcula el Laplaciano del grafo**, una matriz que codifica la estructura del grafo.
3. **Obtiene los vectores propios** del Laplaciano (espectro del grafo).
4. **Proyecta los datos** en un espacio de menor dimensión usando esos eigenvectores.
5. **Aplica K-means** (u otro método de clustering) en ese nuevo espacio reducido.

**Intuición general:**

- El Laplaciano "lee" la conectividad del grafo.
- Los vectores propios del Laplaciano extraen la estructura global.
- Al proyectar los datos usando esos vectores propios, obtenemos una representación que revela los clústeres ocultos.

**¿Por qué usar Spectral Clustering?**

- Detecta **clústeres no convexos** o con formas arbitrarias (por ejemplo, en forma de anillos).
- Muy útil cuando la estructura de los datos **no se ajusta bien a K-means**.

# Spectral clustering

Luego particionamos el grafo usando los valores y vectores propios de la matriz Laplaciana.