DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Laboratorio de datos

Aprendizaje No Supervisado

Verano 2026

Aprendizaje Supervisado

Hasta ahora las observaciones tenían variables predictoras y un valor a predecir conocido (etiquetas).

Ejemplo:

Entrenamiento

Filter

	RU	DI
1	0	104.00
2	50	106.00
3	100	112.30
4	200	117.00
5		

Predicción

Filter

	RU	DI
1	25	
2	600	
3	1500	

Aprendizaje No Supervisado

No contamos datos de entrenamiento etiquetados. Se trata de **descubrir información** y **estructura implícita** en los datos **sin ninguna guía externa**.

Es decir, **descubrir patrones** o **estructuras ocultas** en **datos no etiquetados**. Sirve para **entender**, **resumir**, **relacionar** y **visualizar los datos**.

Posibles usos:

- **Clustering - agrupamiento**
- Reducción de la dimensiónalidad

Herramientas de aprendizaje no supervisado

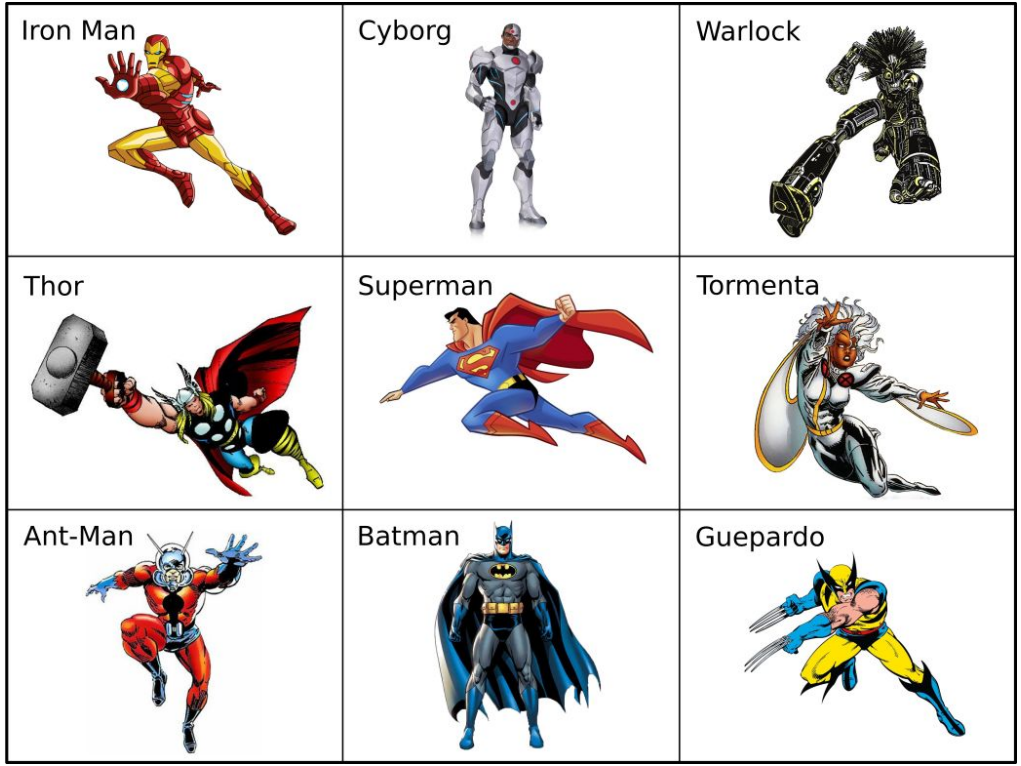
Clustering - Agrupamiento
Métodos para encontrar subgrupos homogéneos dentro del conjunto entero de los datos.

Reducción de dimensionalidad
Métodos para **proyectar los datos** -en general de dimensiones altas- **en un espacio de menor dimensión**, que haga posible su manipulación (o visualización) pero **preserve las características del conjunto original**.
Puede usarse como paso previo al clustering y al aprendizaje supervisado.

Iron Man	Cyborg	Warlock
Thor	Superman	Tormenta
Ant-Man	Batman	Guepardo

Actividad en grupos

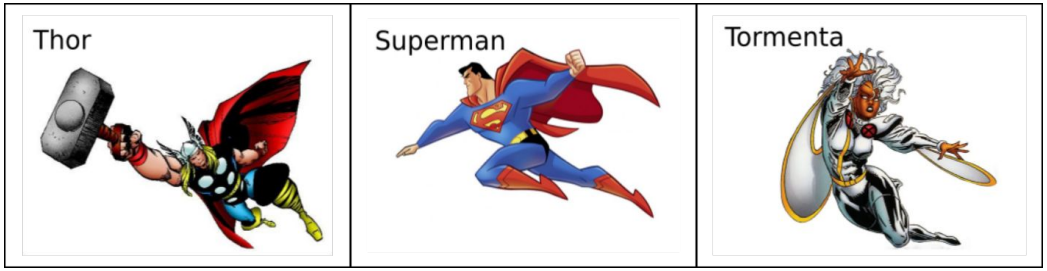
- Conformar equipos de 3 estudiantes
- Agrupar a los 9 superhéroes en grupos.
- Tienen 5 minutos
- Al finalizar, subir una foto de los grupos al siguiente link: <Carpeta Fotos>



Una forma

Robots

Voladores



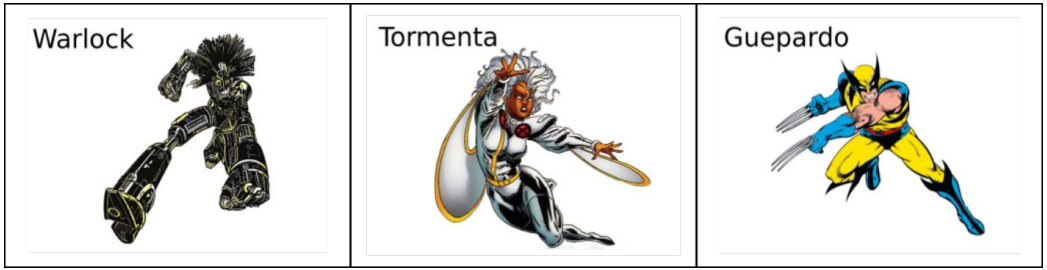
Animales

Otra forma

Avengers

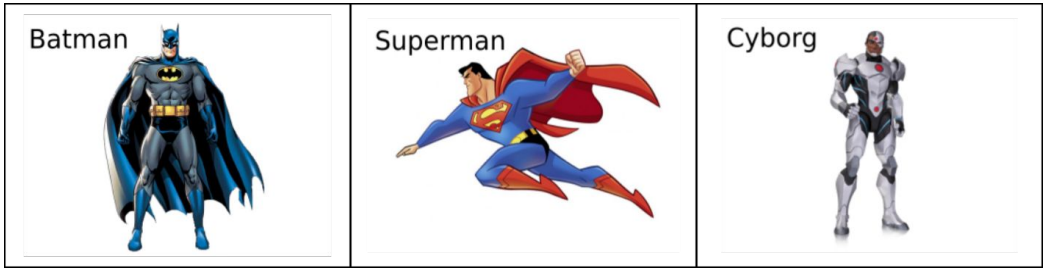
Liga de la Justicia

X-Men



Otra forma

DC Comics



Marvel

¿Qué tipos de clasificaciones tuvimos?

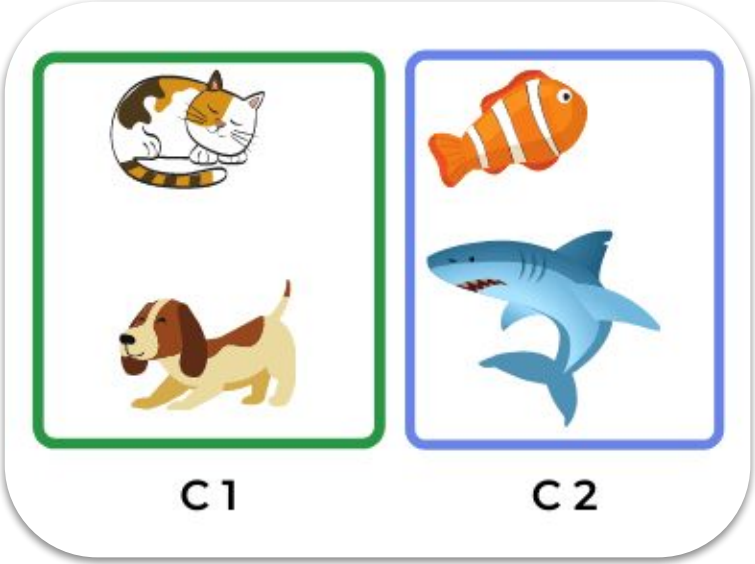
- por características de cada personaje
- por grupo de pertenencia
- por editorial

La selección de atributos nos condiciona el agrupamiento.

¿Cuál es el agrupamiento correcto?

Objetivo: Agrupar los datos en función de sus características de modo tal que

- Los datos que pertenecen a un mismo grupo son similares.
- Los de distintos grupos no son similares.

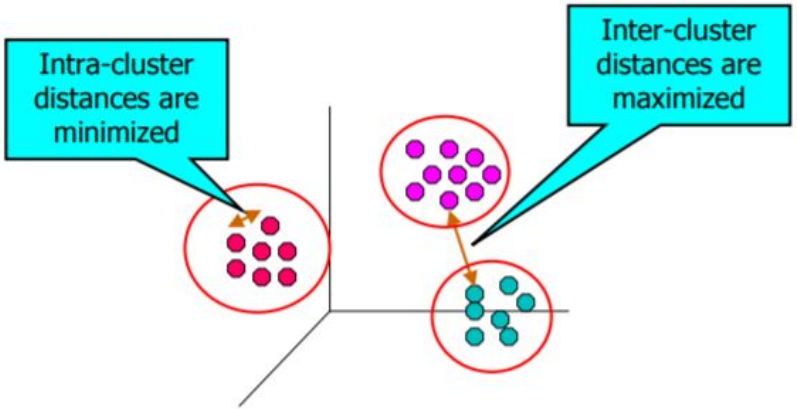


Clustering (agrupamiento)

Encontrar **grupos de instancias (clusters)** a partir de **información en los datos** que **describan objetos** y **sus relaciones**.

Instancias de un cluster tienen que ser:

- similares entre sí y
- diferentes a las de otros clusters

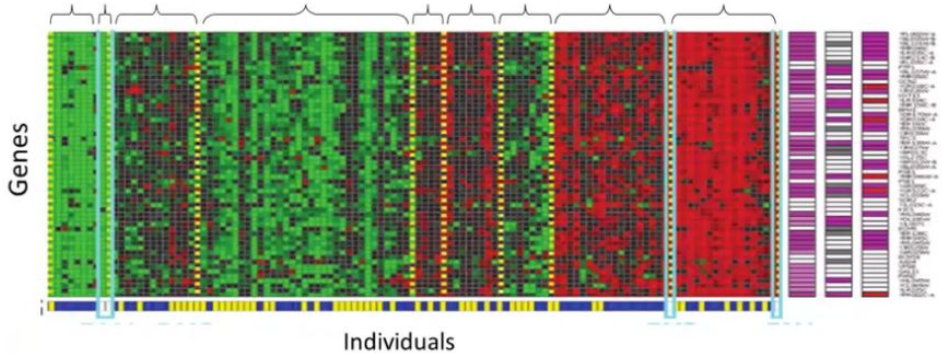


Tan, Steinbach & Kumar, Introduction to Data Mining
https://www-users.cs.umn.edu/~kumar001/dmbook/dmslides/chap8_basic_cluster_analysis.pdf

Algoritmos de clustering

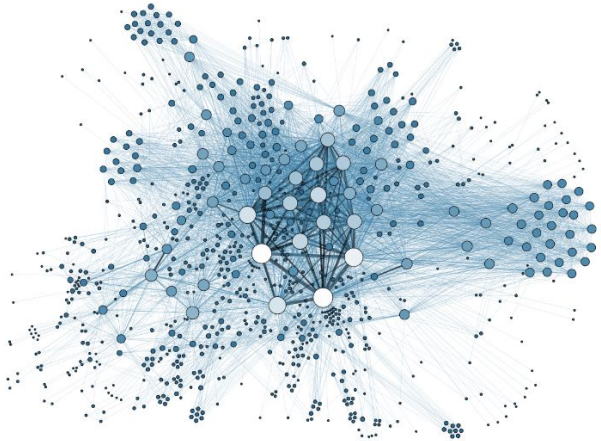
- **De partición:** se clasifican **m datos** en **k clusters**. Cada cluster satisface requerimientos de una partición:
  - cada dato está en un y sólo un cluster
  - cada cluster debe tener al menos un dato
- **Jerárquicos**
  - **Aglomerativos (bottom up):** empiezan con n clusters (cada dato es un cluster) y se combinan grupos.
  - **Divisorios (top down):** comienzan con un cluster de n observaciones y en cada paso se dividen.

Aplicaciones



Fuente: curso ML Stanford

Análisis de redes sociales



Fuente: Wikimedia commons

Segmentación del mercado.
Fuente: internet

Algoritmos de clustering

- **De partición:** se clasifican **m datos** en **k clusters**. Cada cluster satisface requerimientos de una partición:
  - cada dato está en un y sólo un cluster
  - cada cluster debe tener al menos un dato
- **Jerárquicos**
  - **Aglomerativos (bottom up):** empiezan con **m clusters** (cada dato es un cluster) y se combinan en grupos.
  - **Divisorios (top down):** comienzan con un cluster de n observaciones y en cada paso se dividen.
- **De densidad**

K- Means

K - medias

Algoritmo K-medias (K-means)

- Es un método iterativo:
  1. **Inicialización**: se elige la localización de los centroides de los K grupos aleatoriamente
  2. **Asignación**: se asigna cada dato al centroide más cercano
  3. **Actualización**: se actualiza la posición del centroide a la media aritmética de las posiciones de los datos asignados al grupo

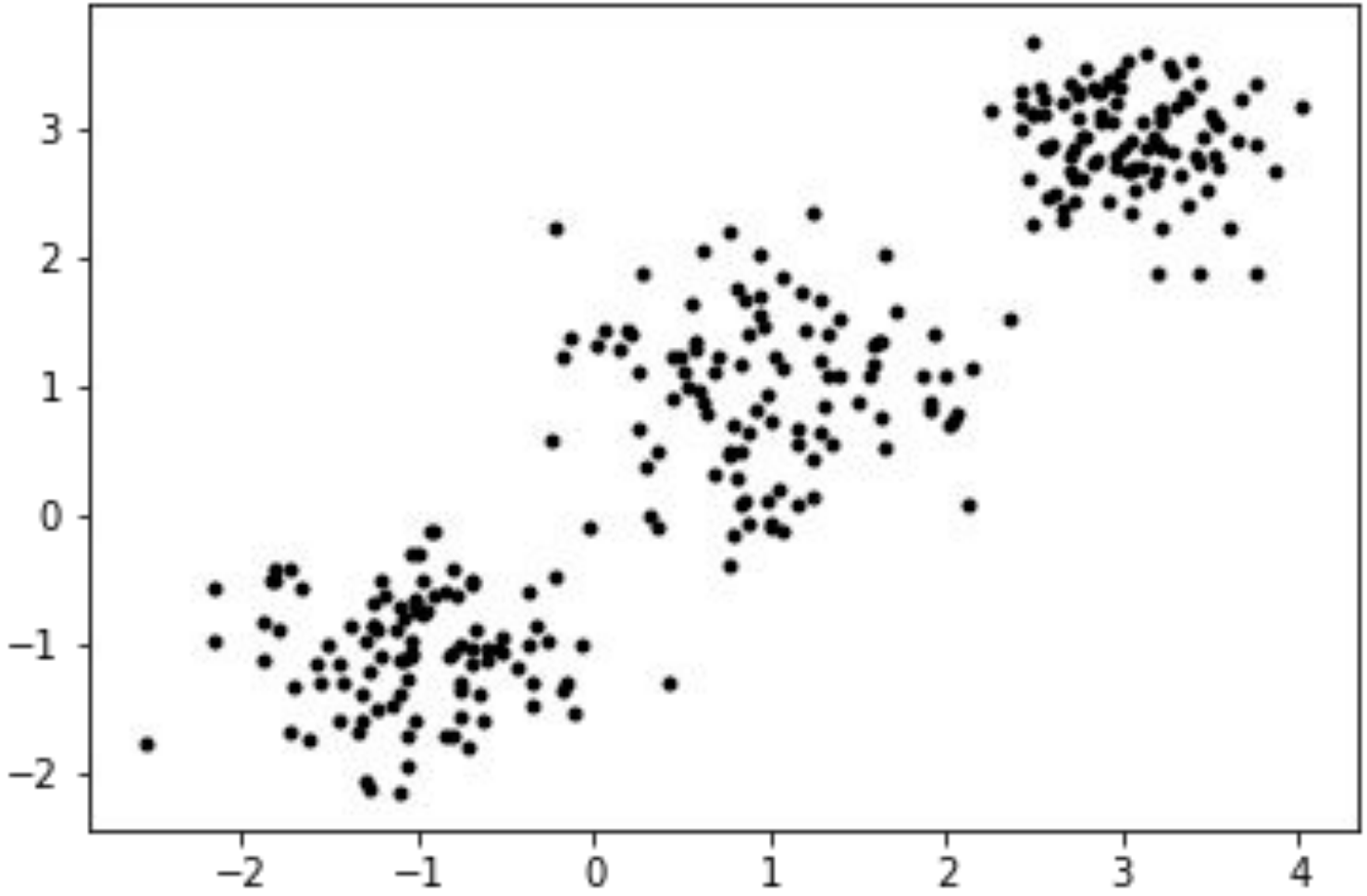
  Repite 2 y 3 hasta que la asignación es estable o se agotan las iteraciones permitidas.

Datos de entrada

Ejecución ( )

Datos de entrada

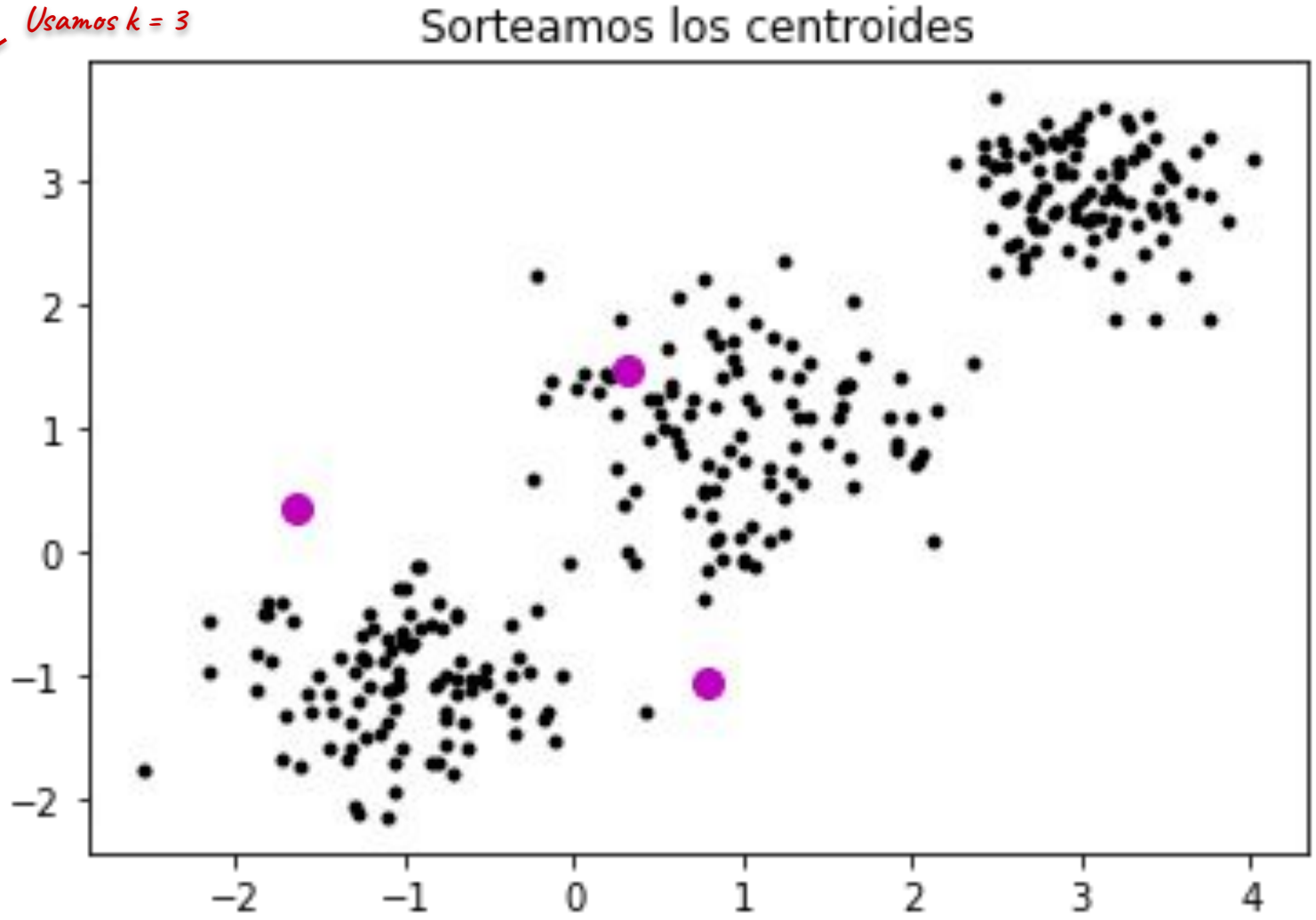
Ejecución (Dataset)



Usamos k = 3

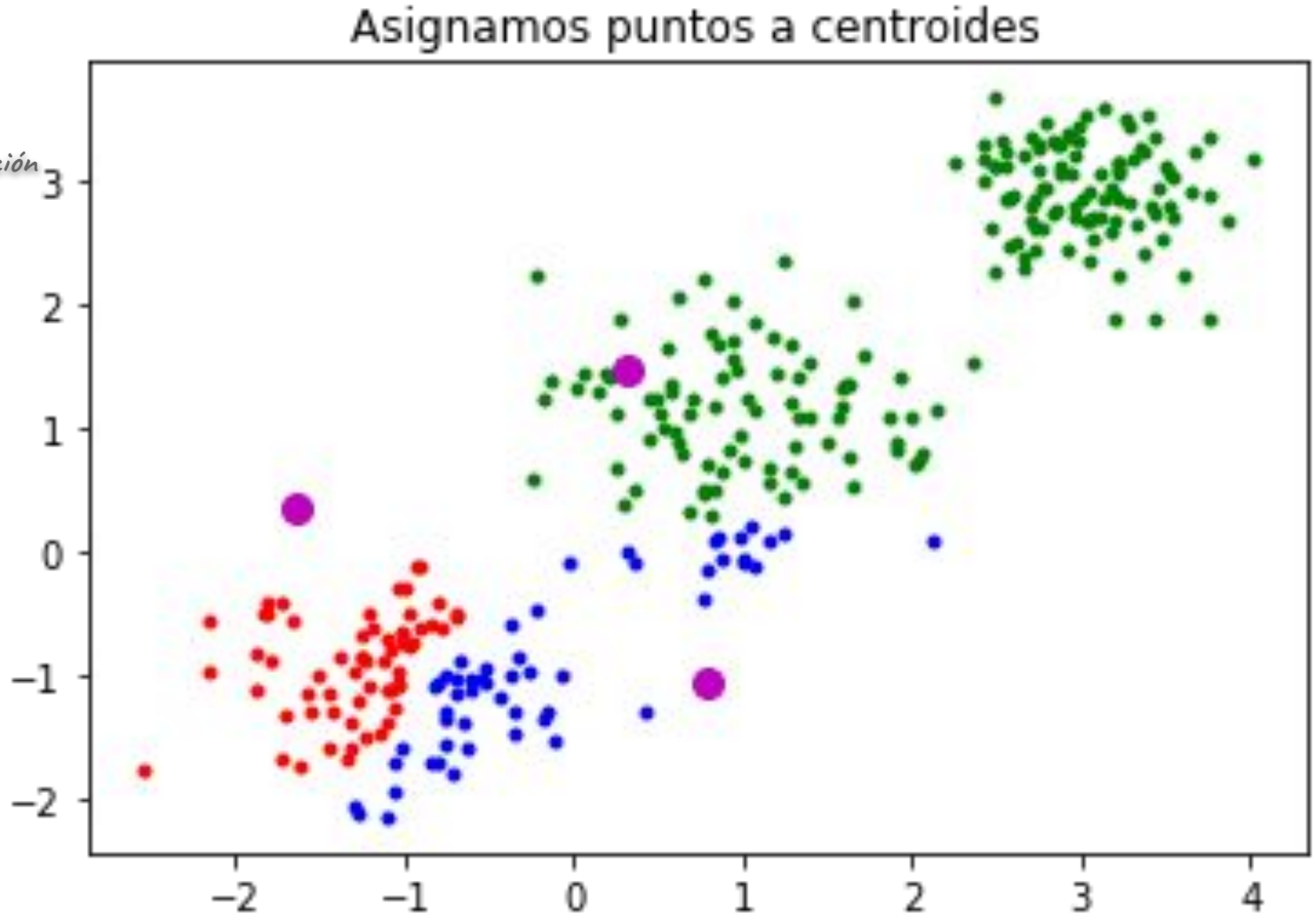
Ejecución (Dataset)

1. Inicialización



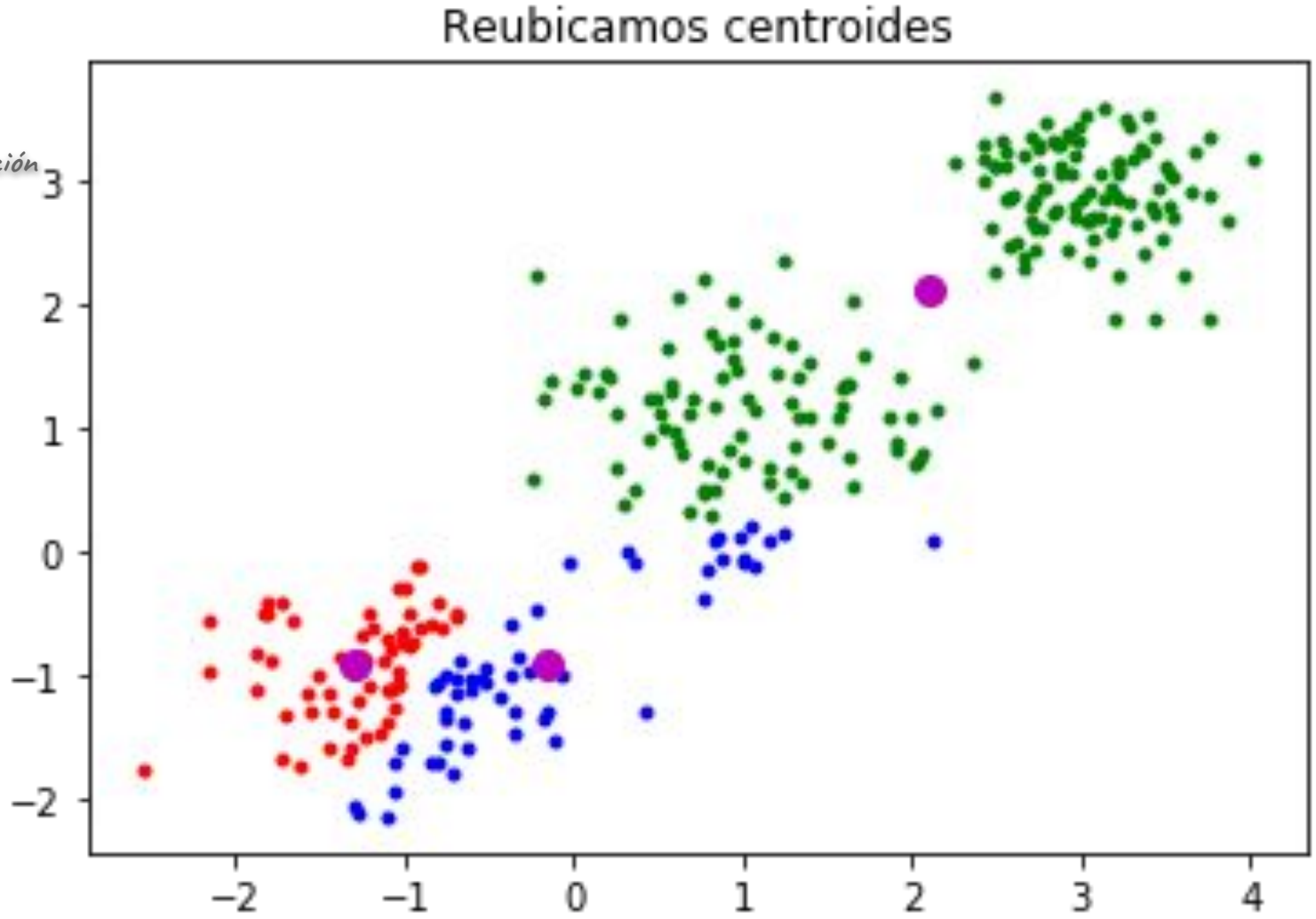
Ejecución (Dataset)

1. Inicialización y Asignación



Ejecución (Dataset)

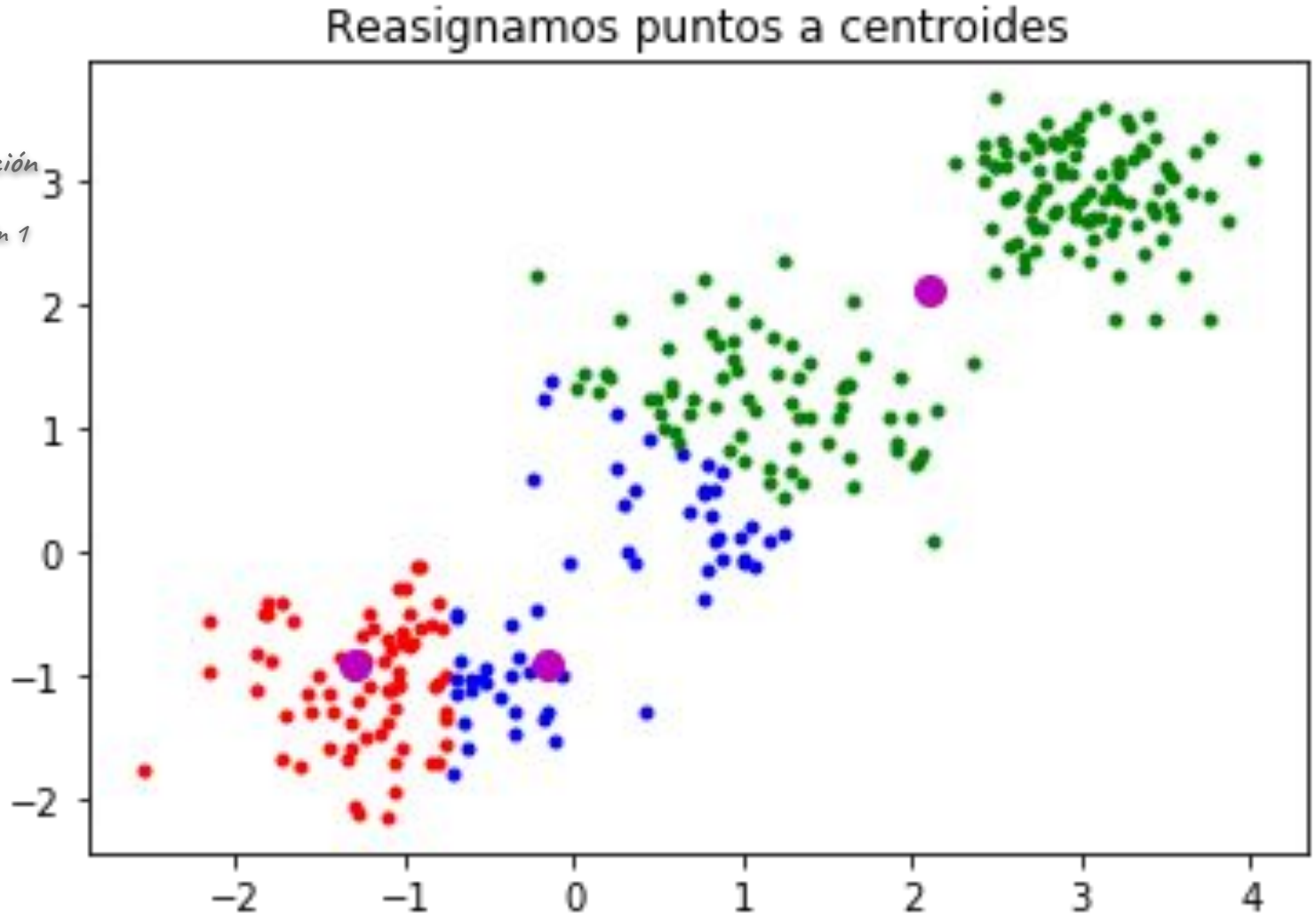
1. Inicialización y Asignación
2. Actualización



Ejecución (Dataset)

1. Inicialización y Asignación
2. Actualización
3. Asignación

} Iteración 1

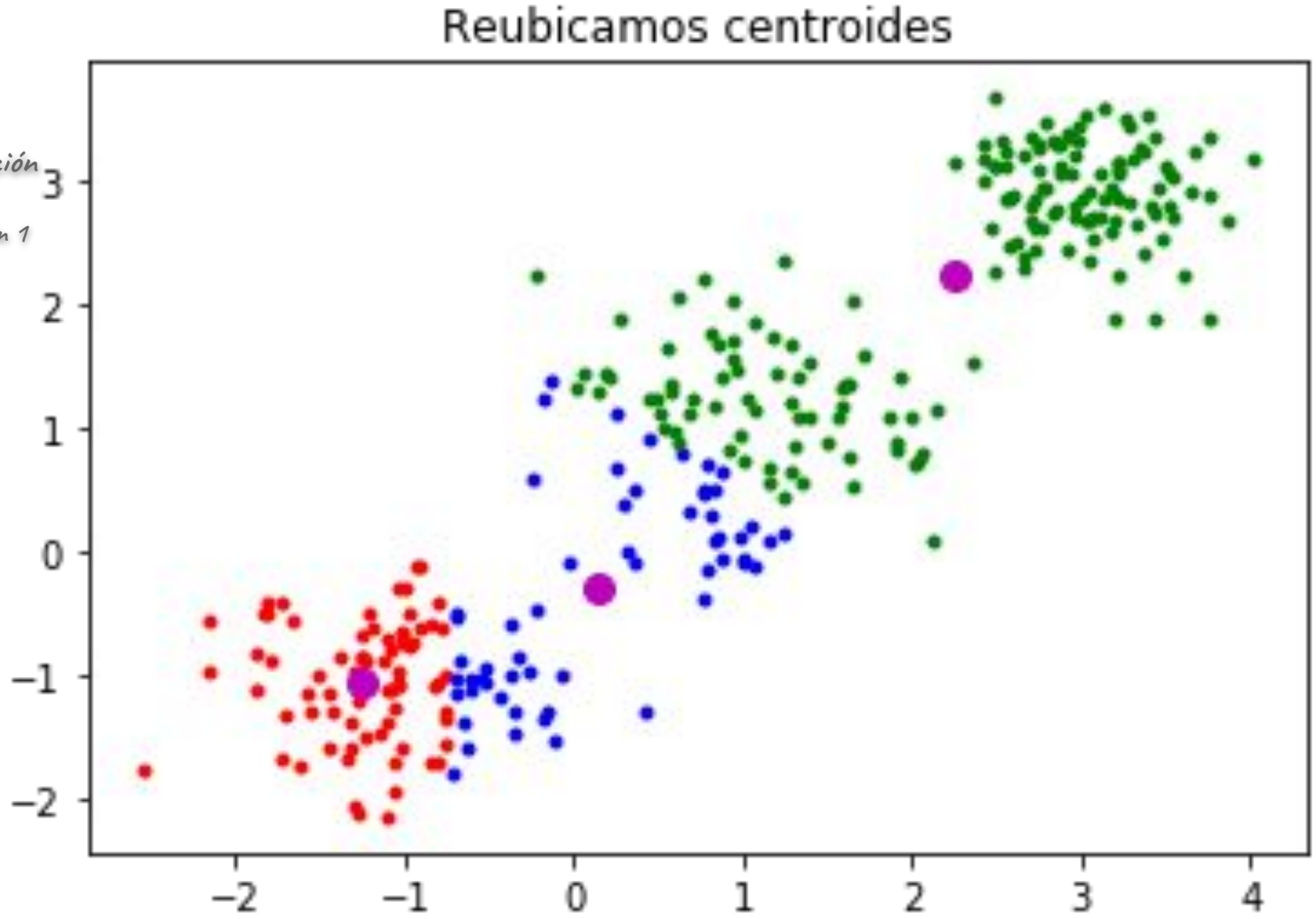


Ejecución (Dataset)

1. Inicialización y Asignación
2. Actualización
3. Asignación

} Iteración 1

2. Actualización



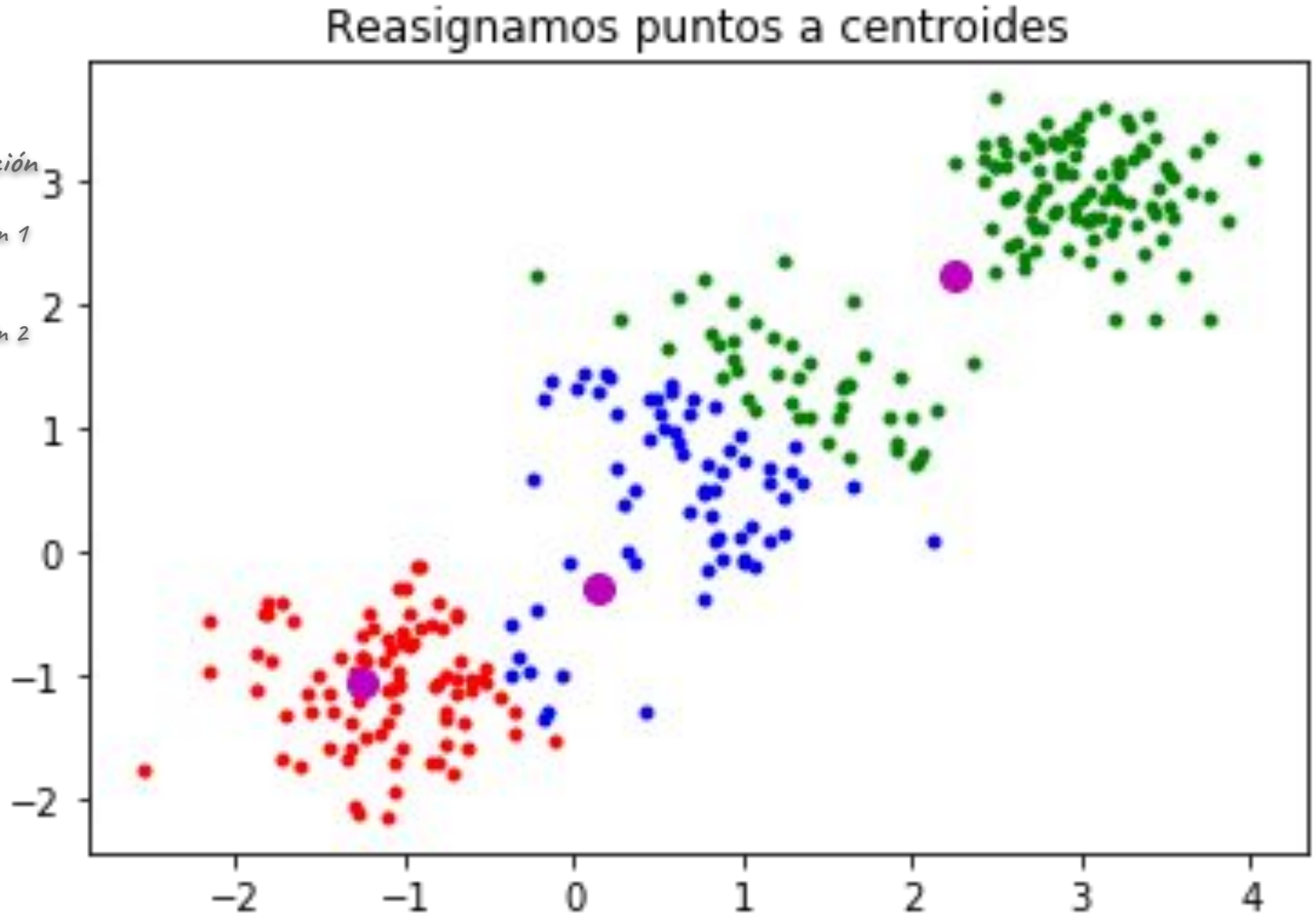
Ejecución (Dataset)

1. Inicialización y Asignación
2. Actualización
3. Asignación

} Iteración 1

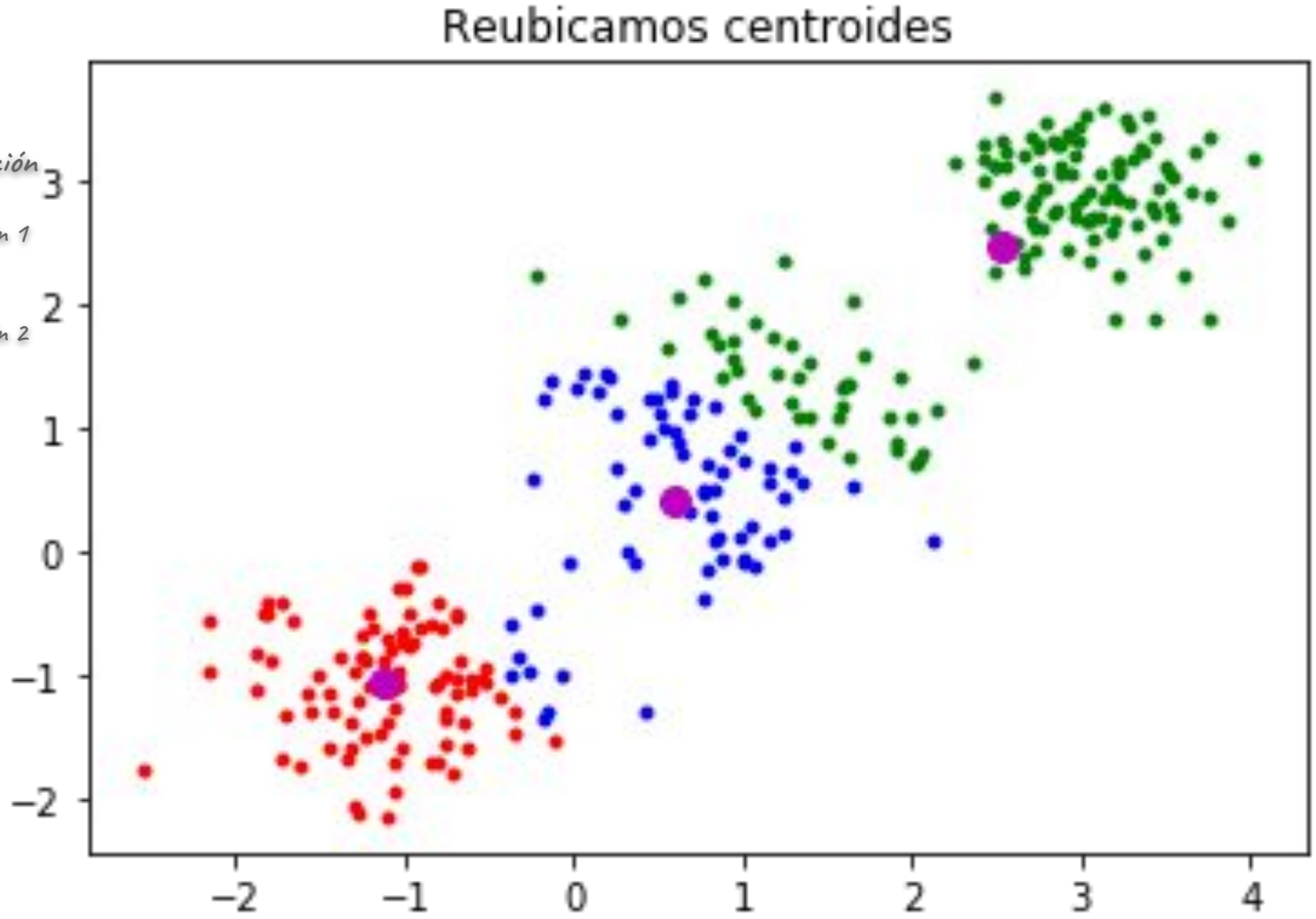
2. Actualización
3. Asignación

} Iteración 2



Ejecución (Dataset)

1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
4. Actualización



Ejecución (Dataset)

1. Inicialización y Asignación
2. Actualización
3. Asignación

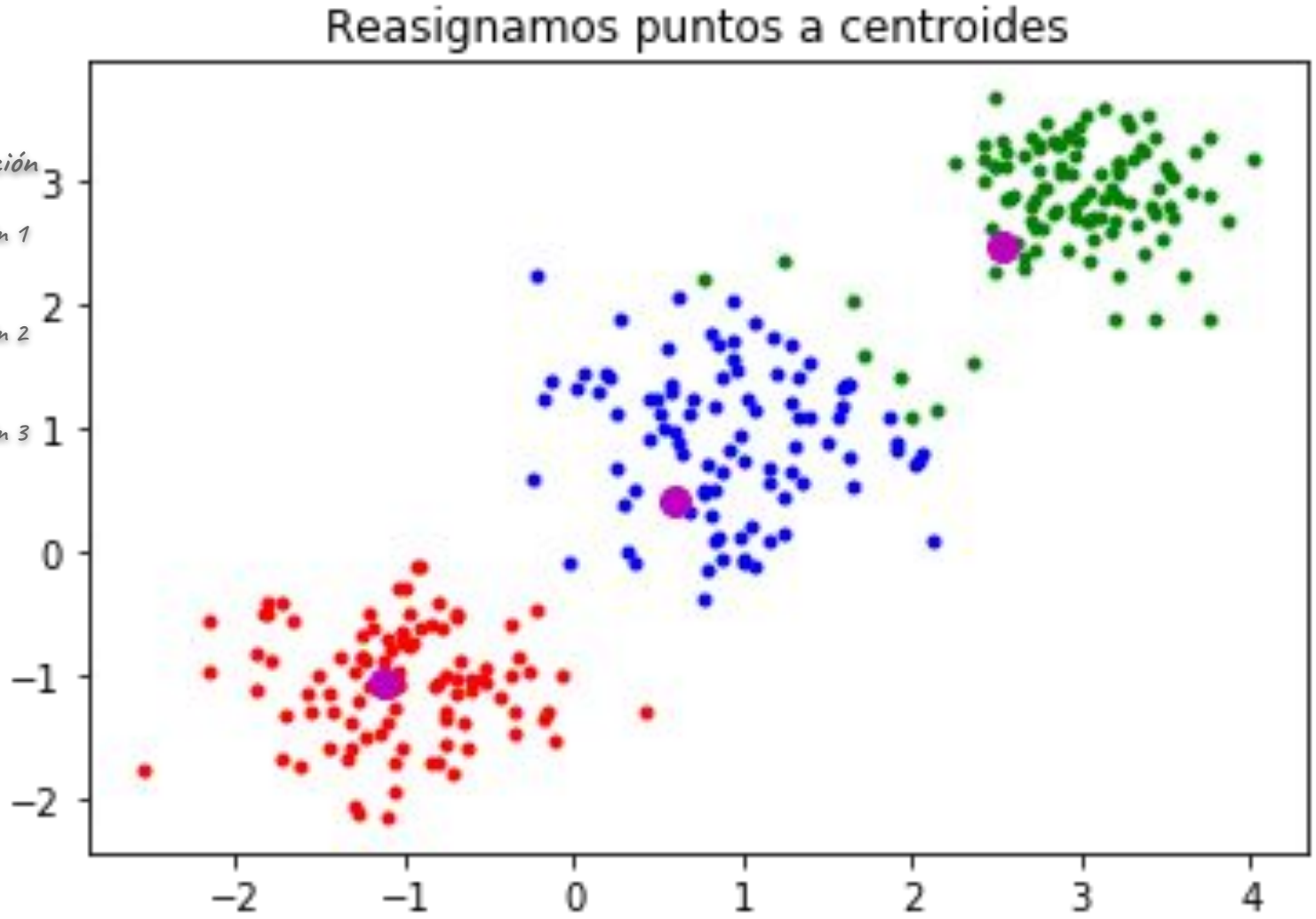
Iteración 1

2. Actualización
3. Asignación

Iteración 2

4. Actualización
5. Asignación

Iteración 3



Ejecución (Dataset)

1. Inicialización y Asignación
2. Actualización
3. Asignación

Iteración 1

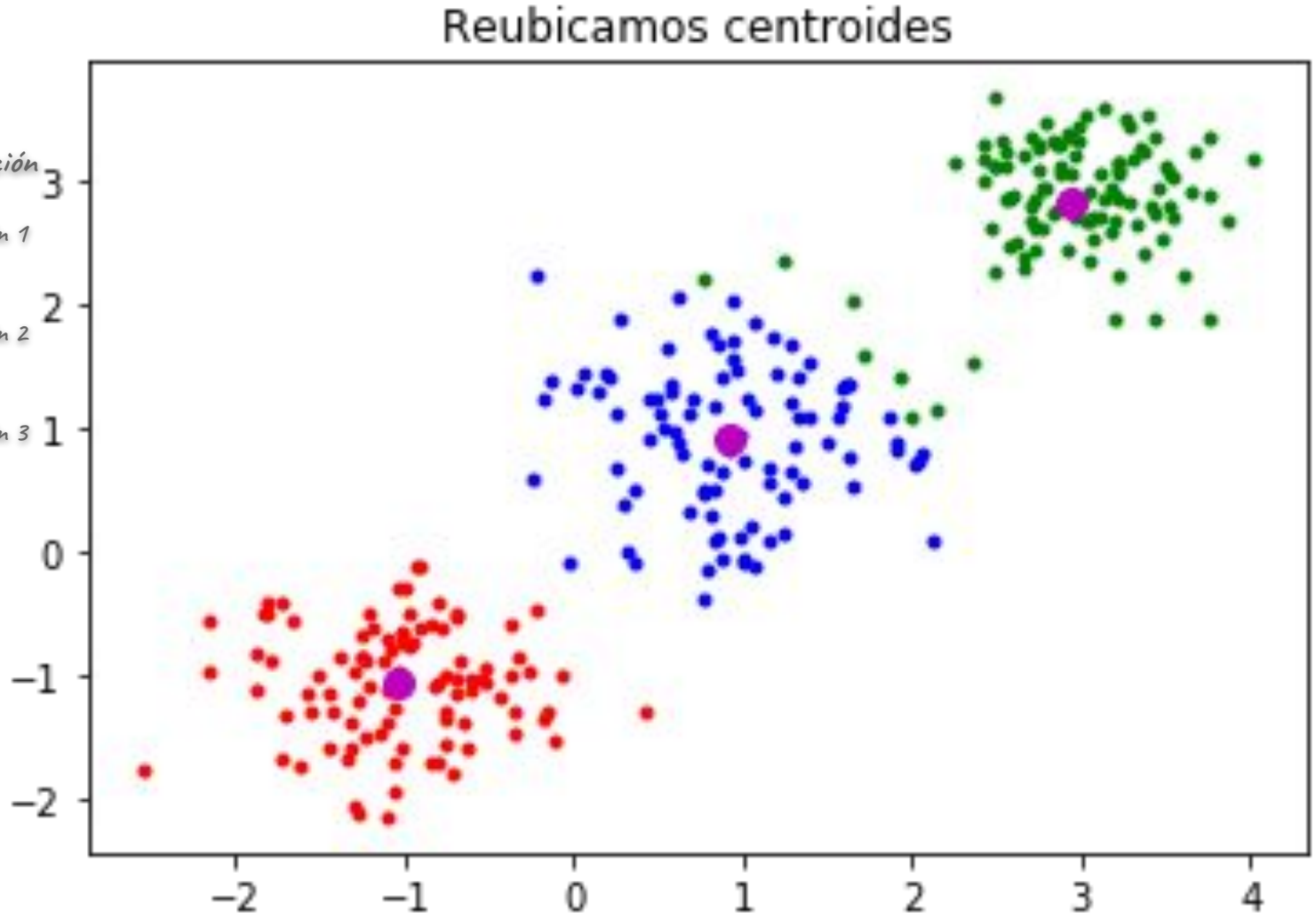
2. Actualización
3. Asignación

Iteración 2

4. Actualización
5. Asignación

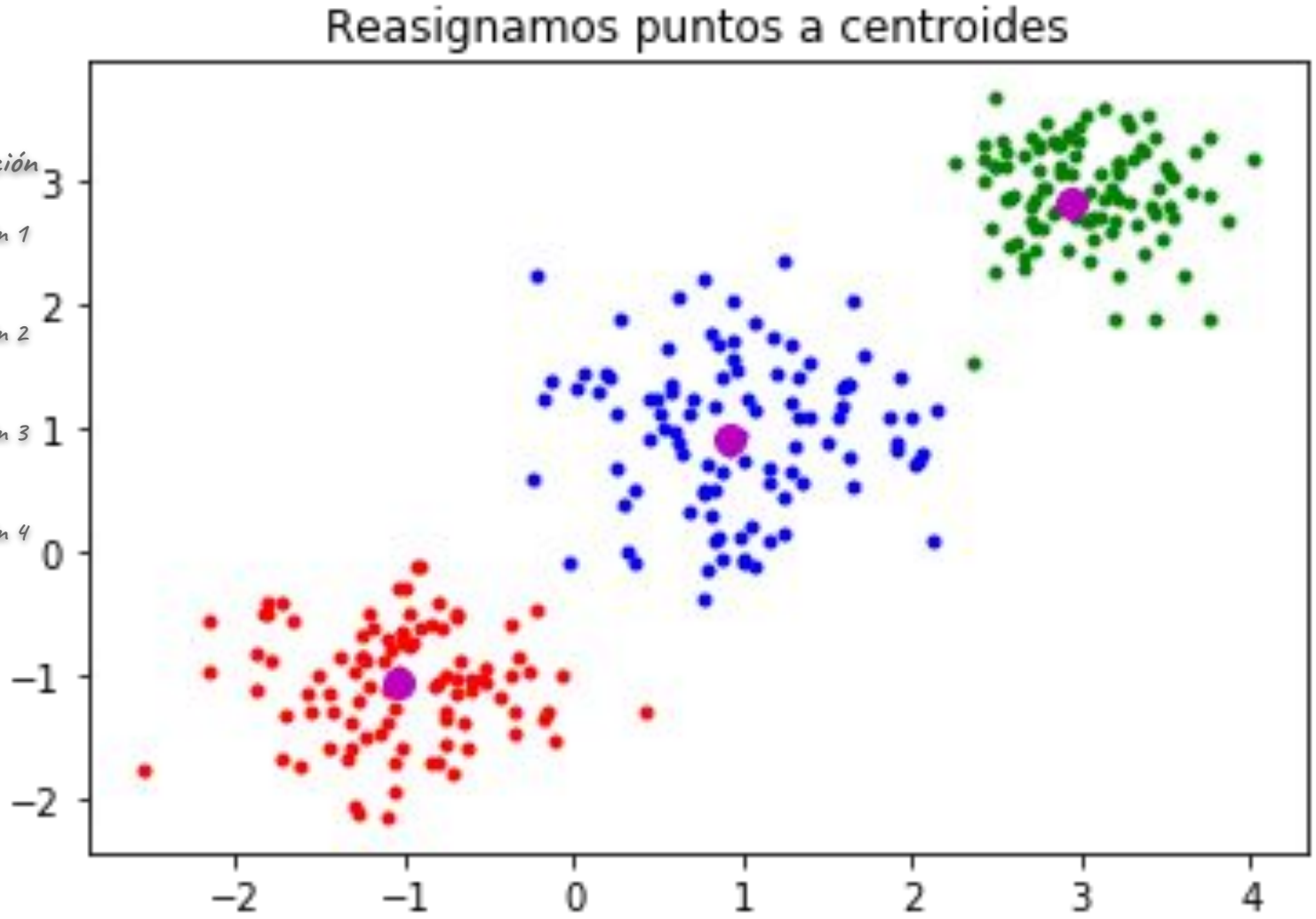
Iteración 3

6. Actualización



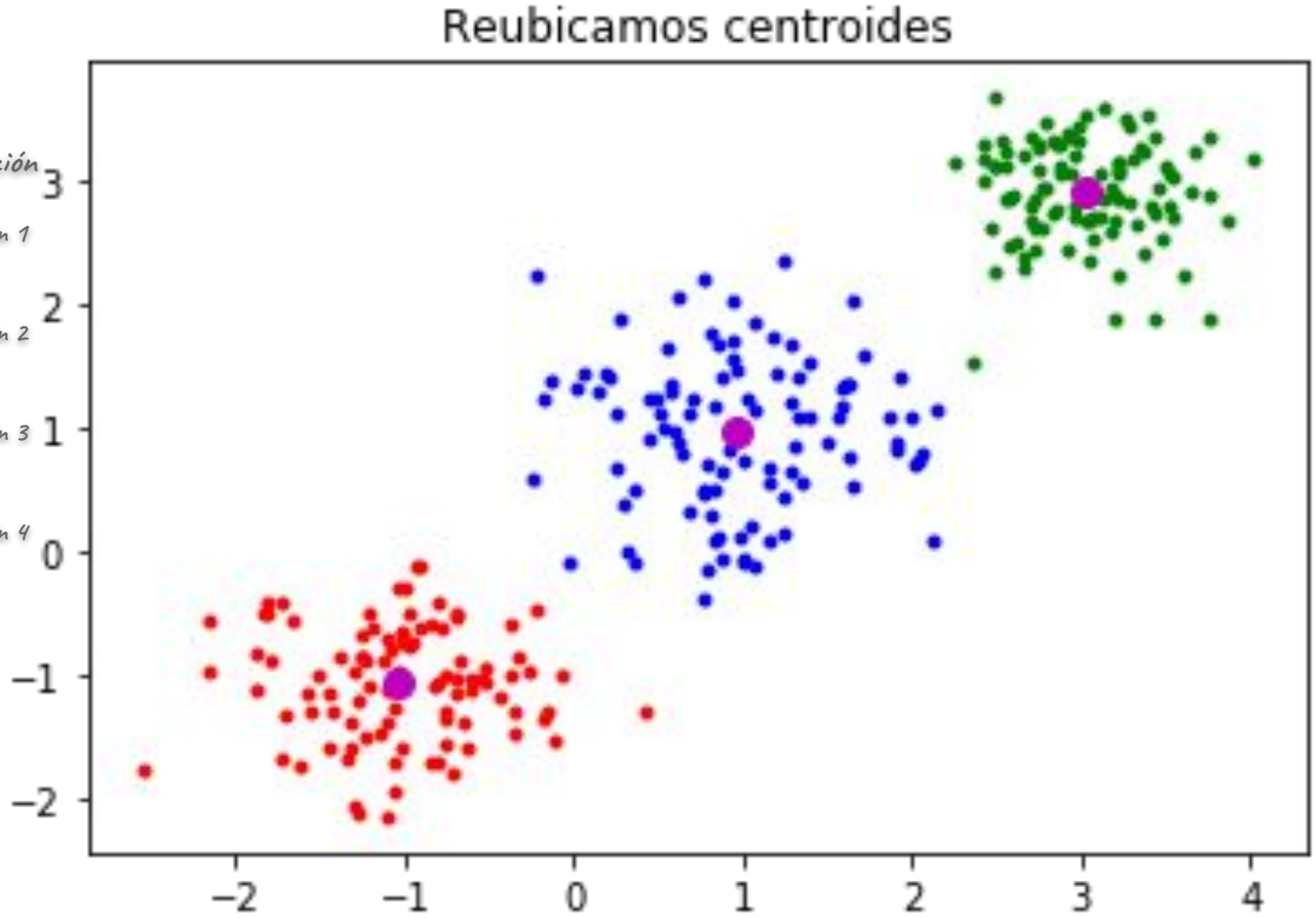
Ejecución (Dataset)

1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
4. Actualización } Iteración 3
5. Asignación
6. Actualización } Iteración 4
7. Asignación



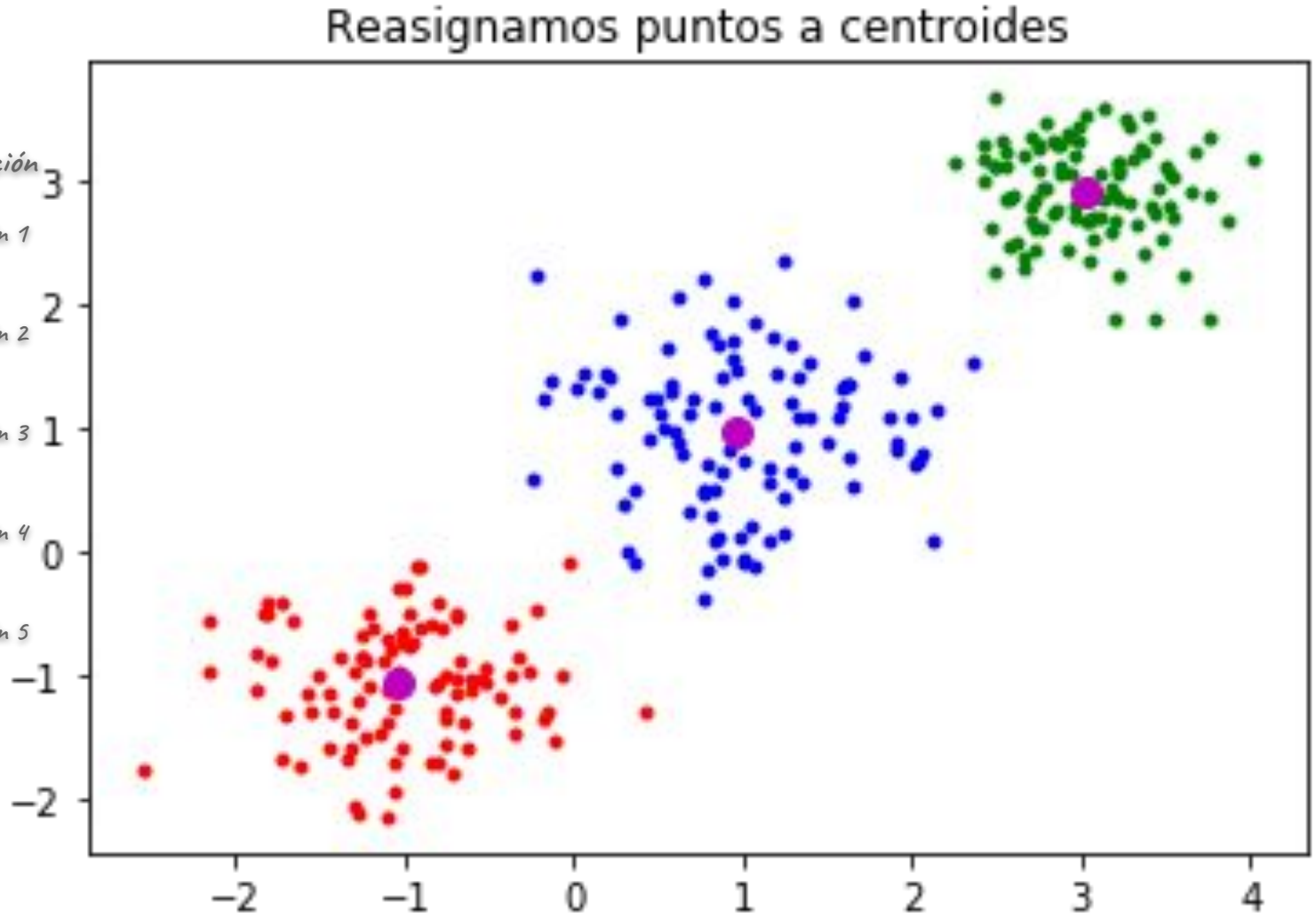
Ejecución (Dataset)

1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
4. Actualización } Iteración 3
5. Asignación
6. Actualización } Iteración 4
7. Asignación
8. Actualización



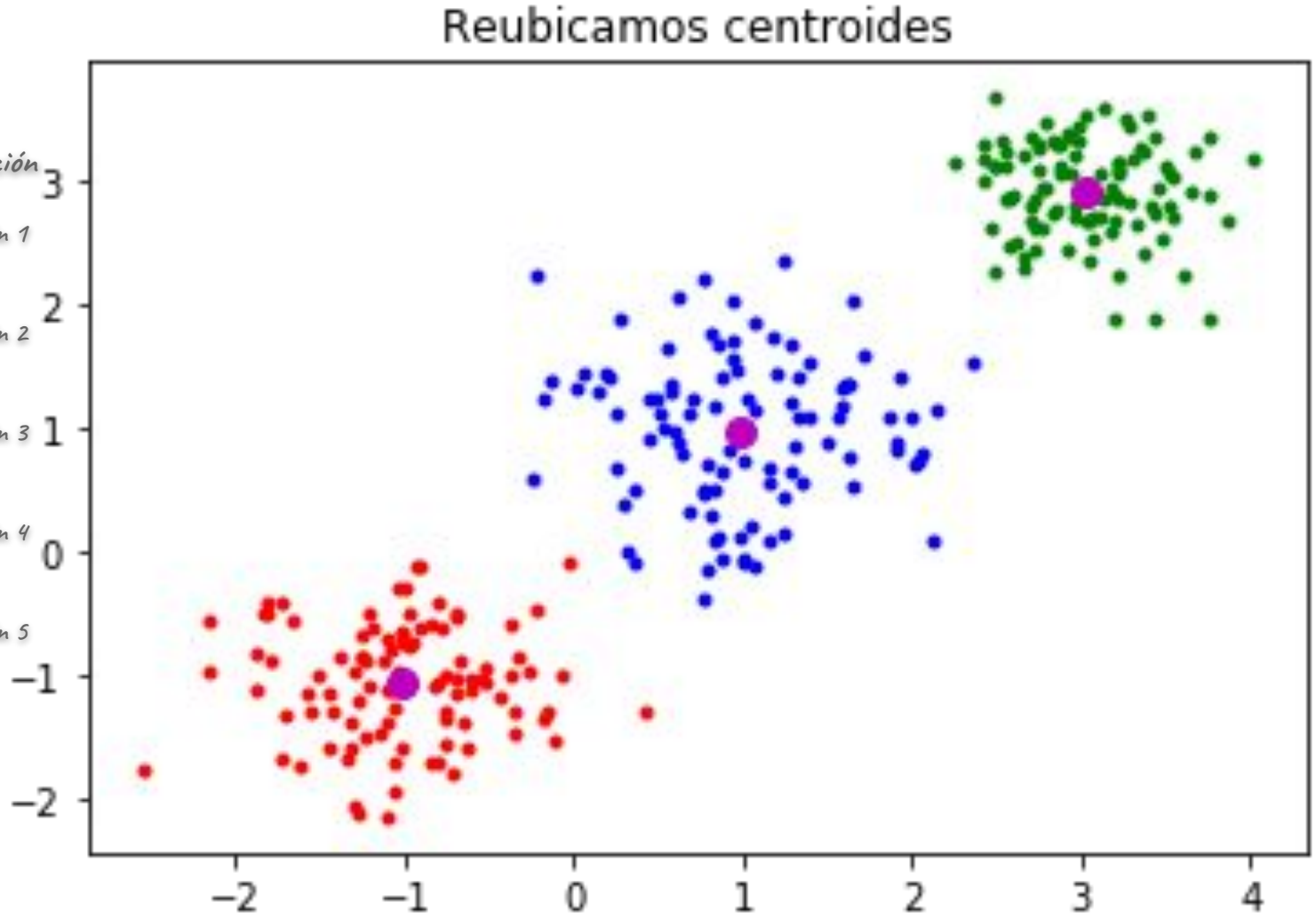
Ejecución (Dataset)

1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
4. Actualización } Iteración 3
5. Asignación
6. Actualización } Iteración 4
7. Asignación
8. Actualización } Iteración 5
9. Asignación



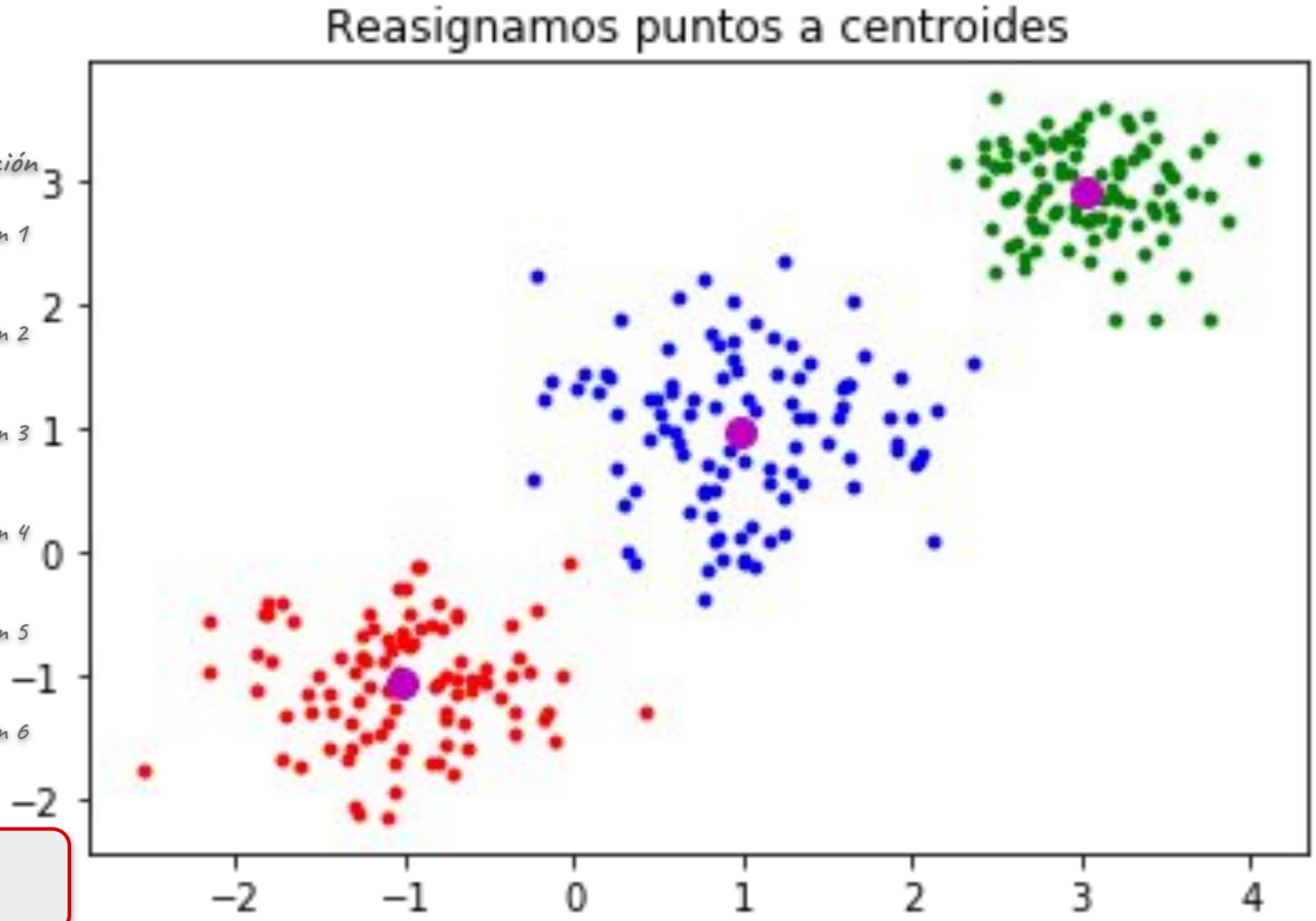
Ejecución (Dataset)

1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
4. Actualización } Iteración 3
5. Asignación
6. Actualización } Iteración 4
7. Asignación
8. Actualización } Iteración 5
9. Asignación
10. Actualización



Ejecución (Dataset)

1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
4. Actualización } Iteración 3
5. Asignación
6. Actualización } Iteración 4
7. Asignación
8. Actualización } Iteración 5
9. Asignación
10. Actualización } Iteración 6
11. Asignación



Criterio de parada

K-means

Si un cluster no tiene puntos:

- se elimina el centroide o
- se ubica nuevamente el centroide al azar

K-means

Optimización, función de costo a minimizar. Función de **distorsión**.

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} a_{ik} \cdot ||x^{(i)} - \mu_k||^2$$

*Within-Cluster Sum of Squares (**WCSS**)*

K: nº de clusters, m: cant. datos

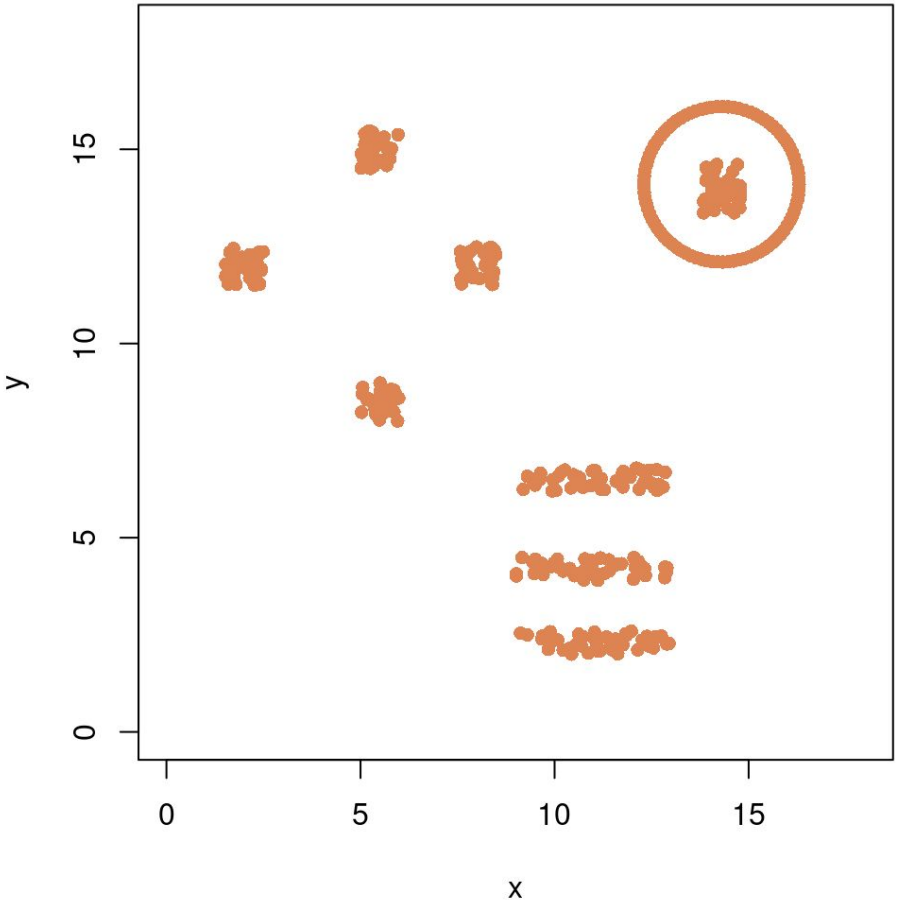
μ_k: centroide de cluster k

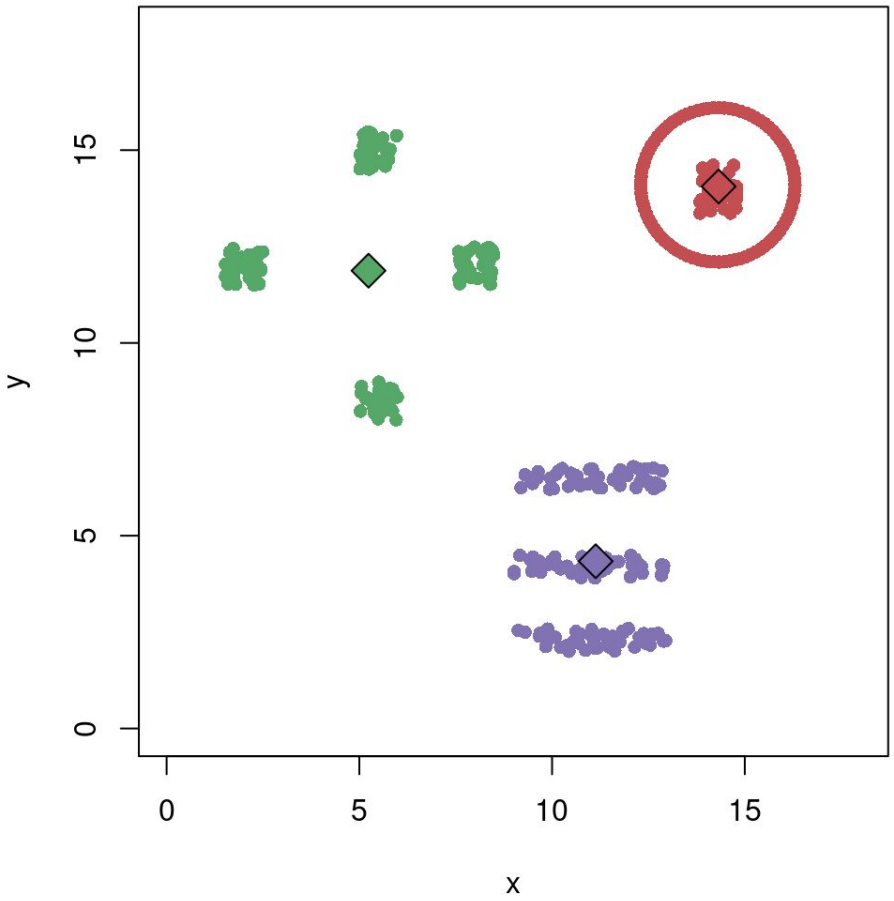
$x^{(i)}$: dato nro i

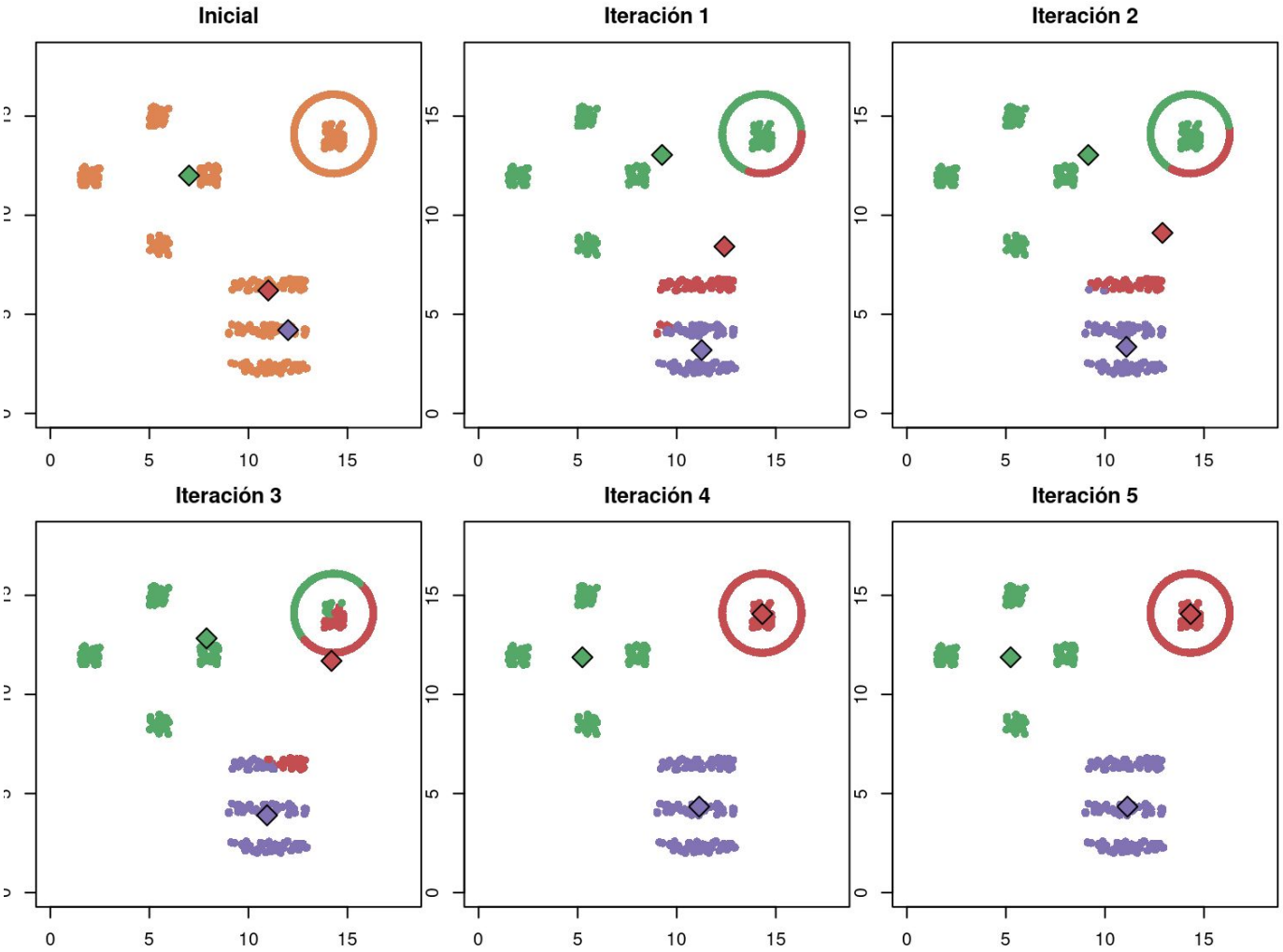
$$a_{ik} \begin{cases} 1 & \text{si } x^{(i)} \text{ está asignado al cluster k} \\ 0 & \text{en otro caso} \end{cases}$$

K-means intenta encontrar μ_k y a_{ik} que minimicen J

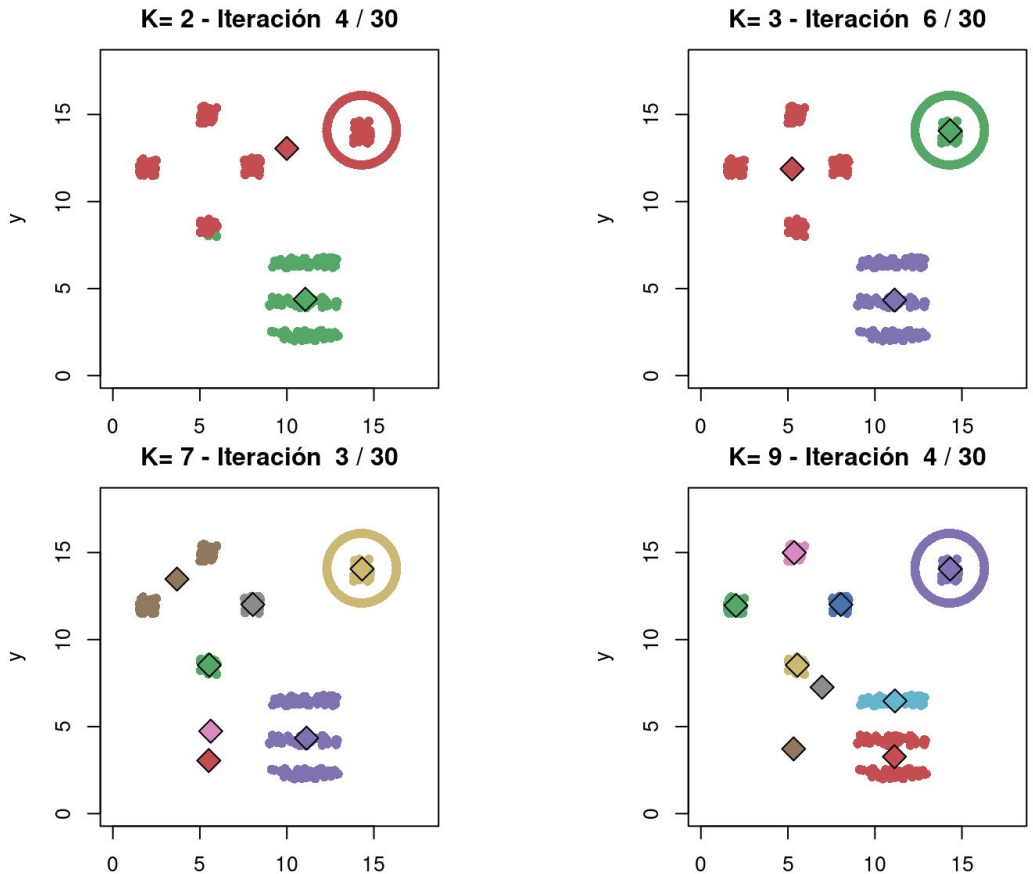
- En **asignación de cluster**: minimiza J con respecto a a_{ik} (asignando los puntos al centroide más cercano, que está fijo)
- En **actualización de centroide**: minimiza J con respecto a μ_k (la ubicación de los centroides)


15
10
5
0
y
0
5
10
15
x

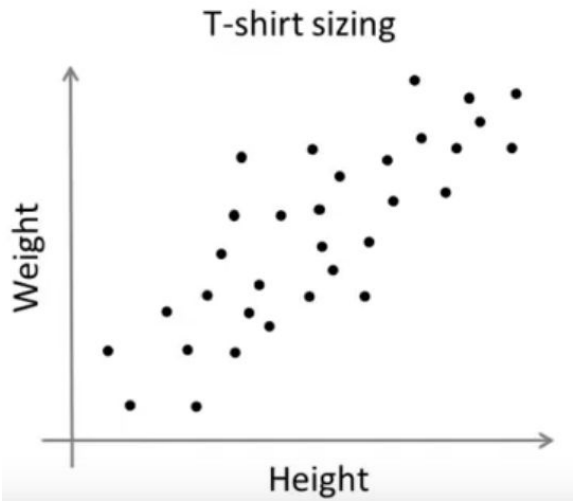

y
0
5
10
15
x
0
5
10
15


Inicial
Iteración 1
Iteración 2
Iteración 3
Iteración 4
Iteración 5
0
5
10
15

¿Qué sucede si lo ejecutamos con otros valores de k?



K-means



Fuente: curso ML Stanford

Los problemas **no necesariamente están bien separados en clusters**.

Además, muchas veces la **dimensión > 3** (técnicas de reducción de la dimensionalidad)

K-means - Inicialización

K: cantidad de clusters
m: cantidad de datos

Requisito: $K<m$

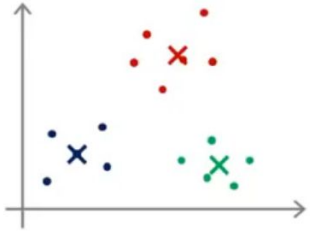
Inicialización de centroides:

- al azar
- **elegir K datos cualesquiera**

K-means - Inicialización

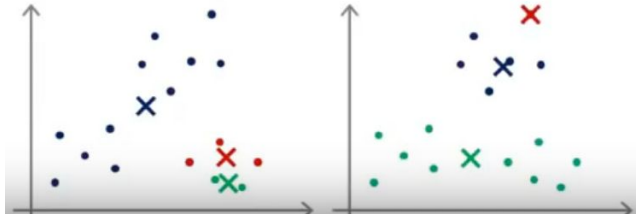
K: cantidad de clusters
m: cantidad de datos

Requisito: K<m



Inicialización de centroides:

- al azar
- **elegir K datos cualesquiera**



Puede converger a distintas soluciones dependiendo de cómo lo inicializo

K-means - Inicialización

Inicialización de centroides:

- al azar
- elegir K datos cualesquiera
- **múltiples inicializaciones al azar (para evitar óptimos locales)**

  Hacer entre 50 y 1000 veces
  - **inicializar centroides**
  - **correr k-means y calcular la función de costo**

  elegir el clustering que tuvo la menor función de costo

Útil en casos con K chicos (<10)

- usar k clusters de un método jerárquico e inicializar con sus centroides
- ...

¿Cómo evaluamos el modelo?

¿Qué sería una métrica de que algo es bueno (o malo)?

Proponemos la distancia de cada punto al centro que le asignamos (Within-Cluster Sum of Squares (WCSS)):

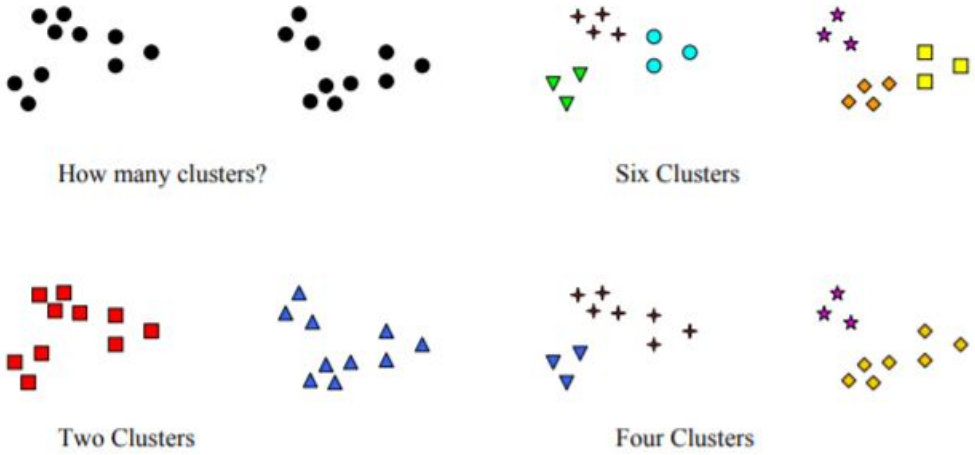
$$WCSS = \sum_{c_i \in C} \sum_{p_{i,j} \in C_i} distancia(p_{i,j} - c_i)^2$$

Siendo C el conjunto de los centros y C_i el conjunto de puntos del cluster i.

K-means - Cómo elegimos el valor de K

¿Cómo elegimos el K? ¿Manualmente?

Ambigüedad



Tan, Steinbach & Kumar, Introduction to Data Mining. Cap 8

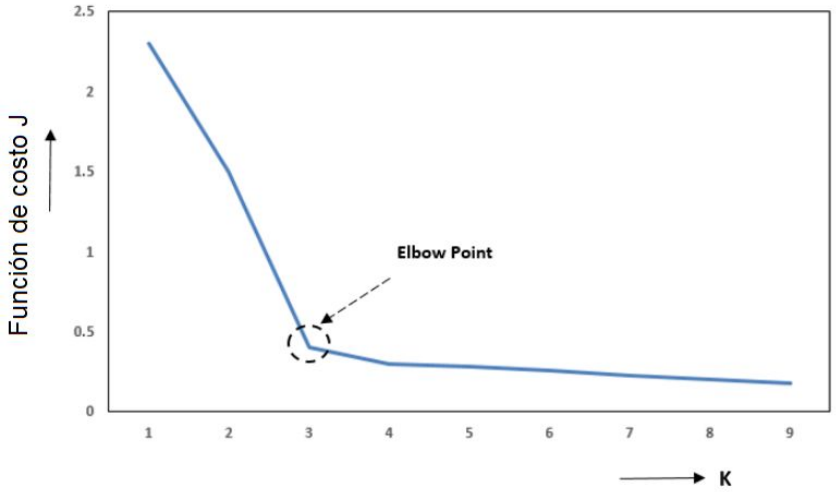
Elección del K

- **manualmente**. Problema: ambigüedad.
- **elbow method**
- **evaluar con una métrica** y ver cuán bien funciona para un propósito posterior

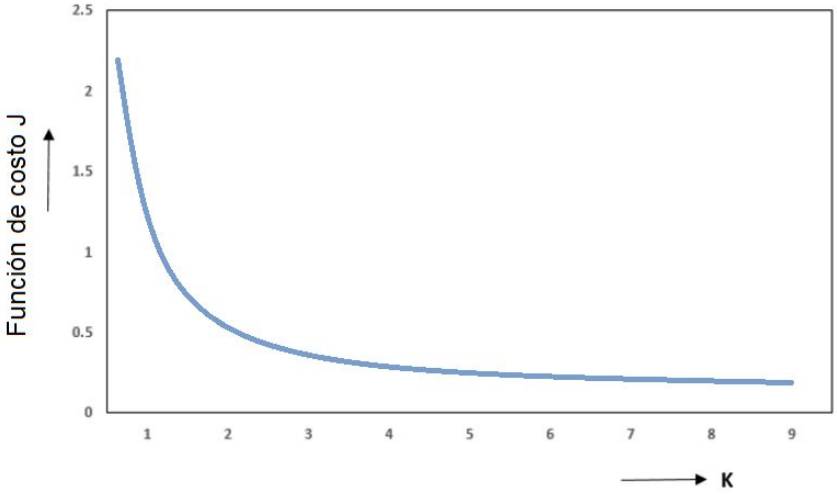
K-means - Elección del K

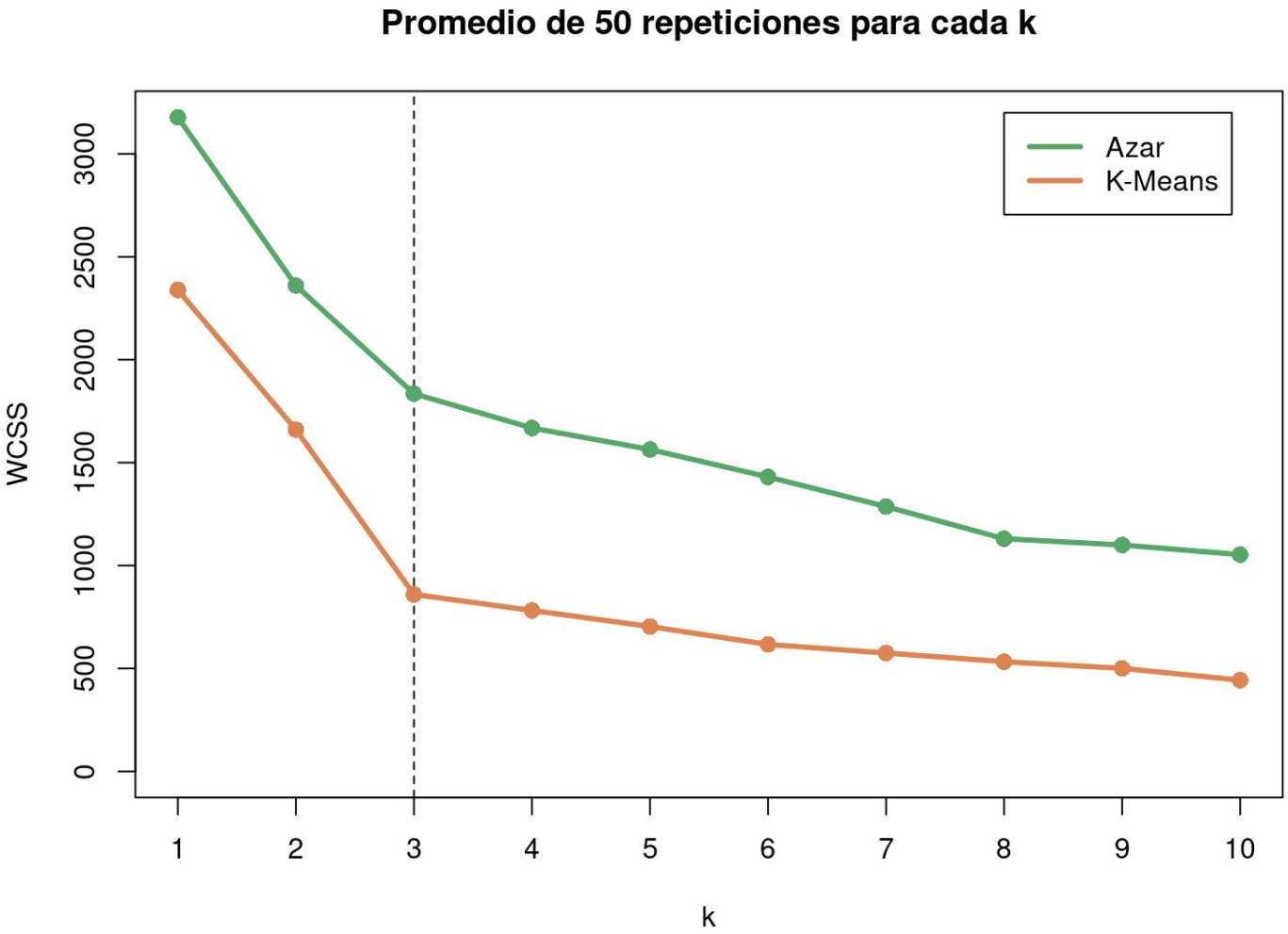
Graficamos WCSS u otra función de costo para un rango de k y usamos el *<u>método del codo</u>*

Elbow method



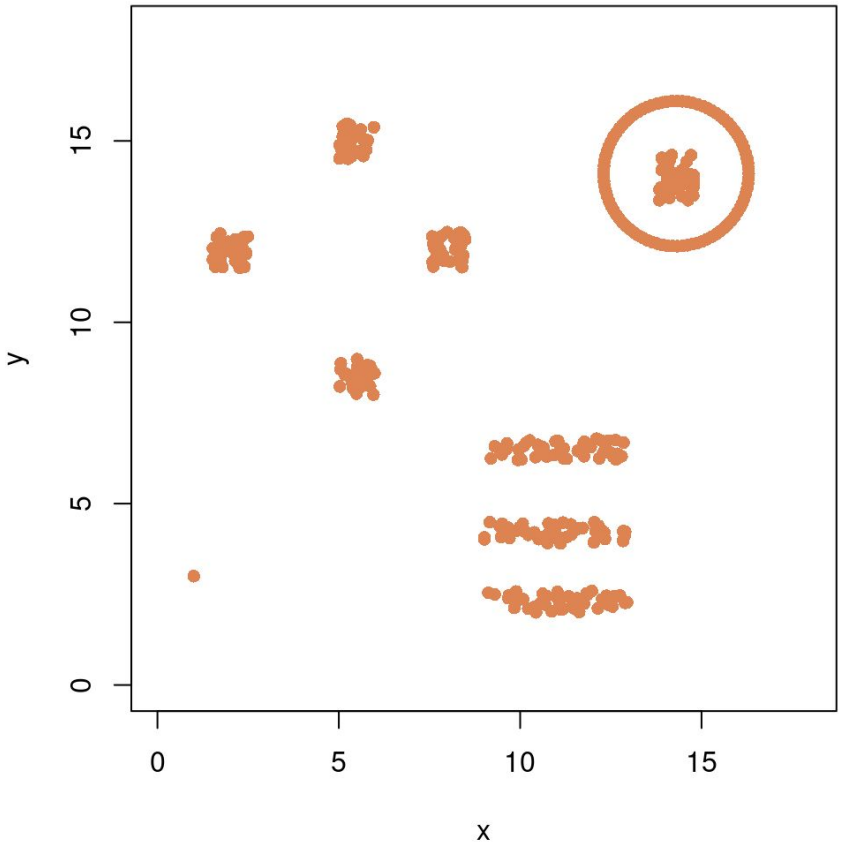
no siempre es útil...



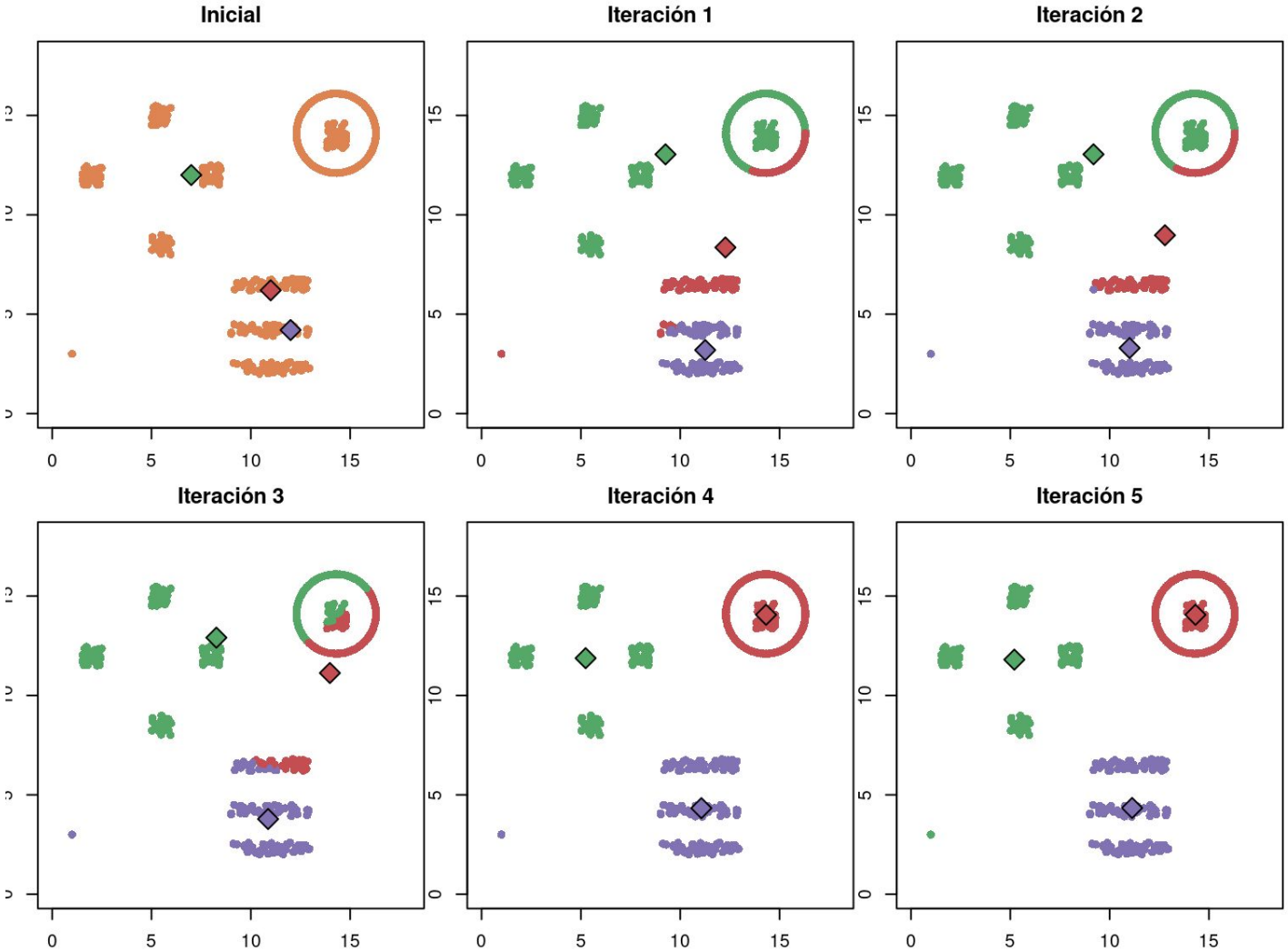

Promedio de 50 repeticiones para cada k
Azar
K-Means
WCSS
0
500
1000
1500
2000
2500
3000
1
2
3
4
5
6
7
8
9
10
k

¿Qué pasa con los datos atípicos (outliers)?

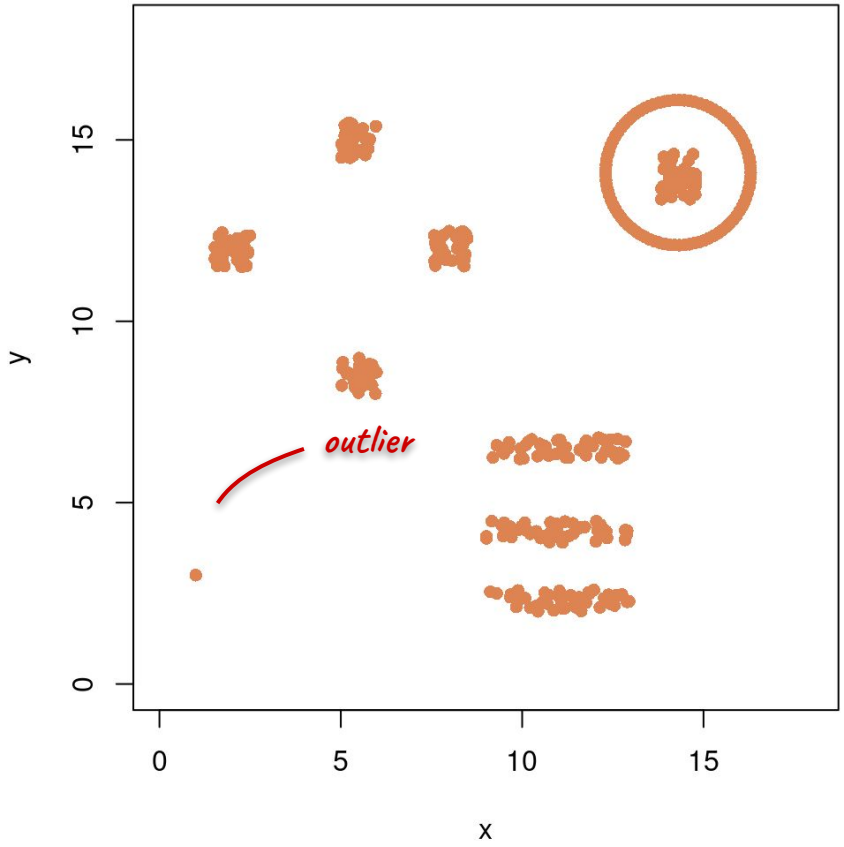
Outliers



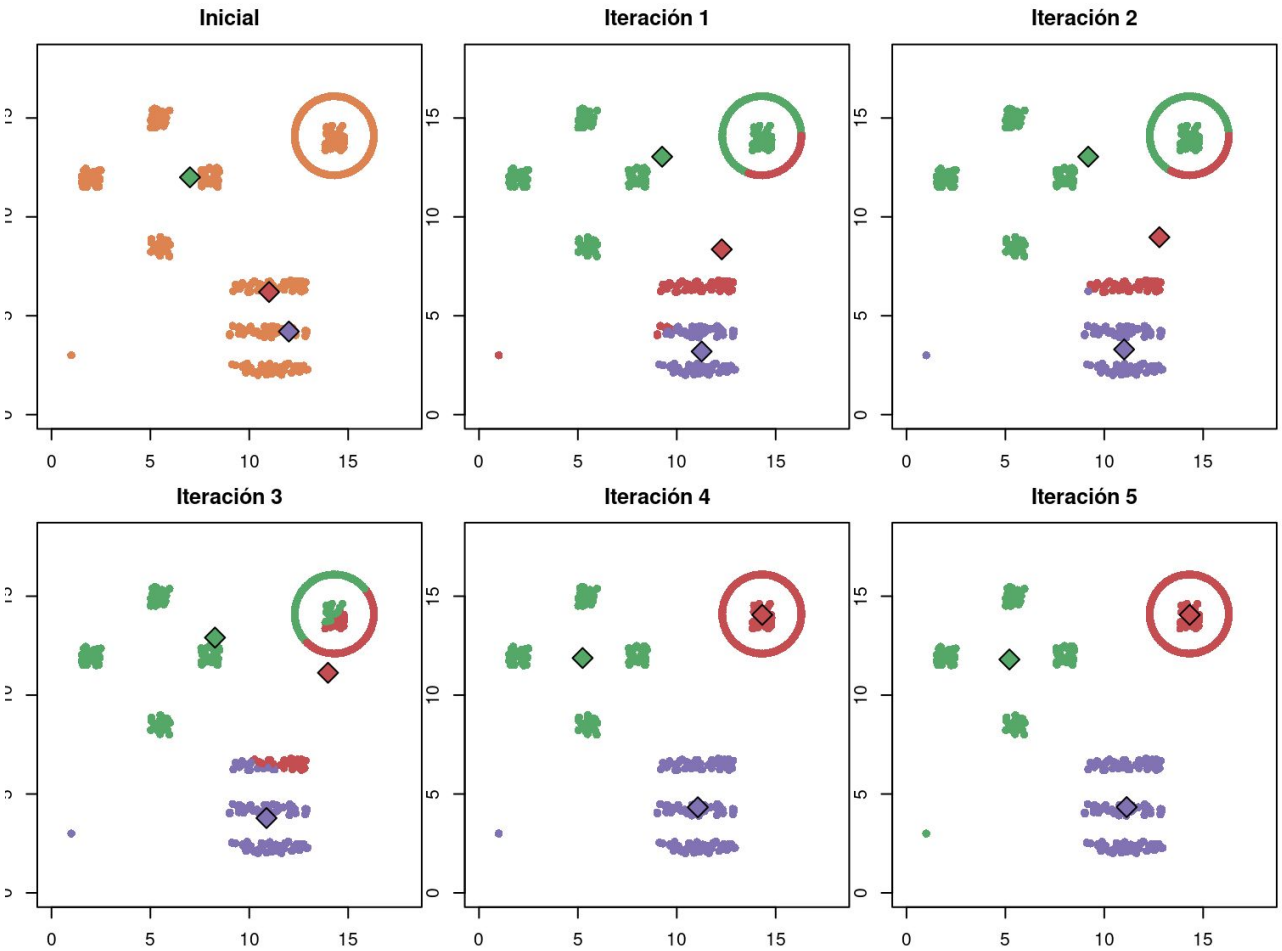
Outliers


Inicial
Iteración 1
Iteración 2
Iteración 3
Iteración 4
Iteración 5
0
5
10
15

¿Qué sucede con los outliers?



¿Qué sucede con los outliers?



K-Means con scikit-learn

```
from sklearn.cluster import KMeans
```

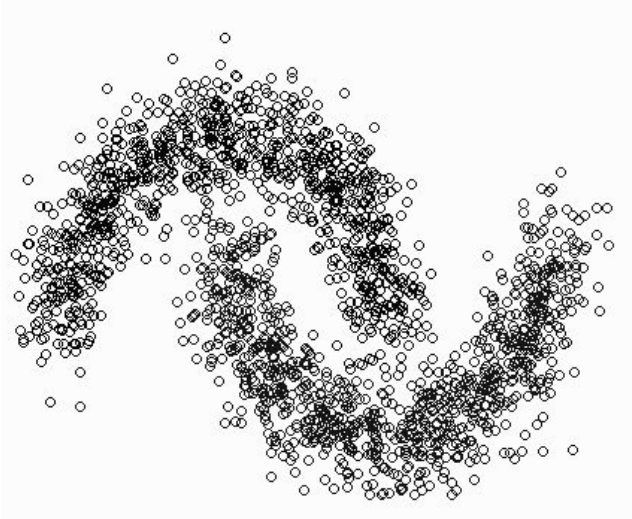
```
kmeans = KMeans(n_clusters = 2, random_state = 2, n_init = 20)
```

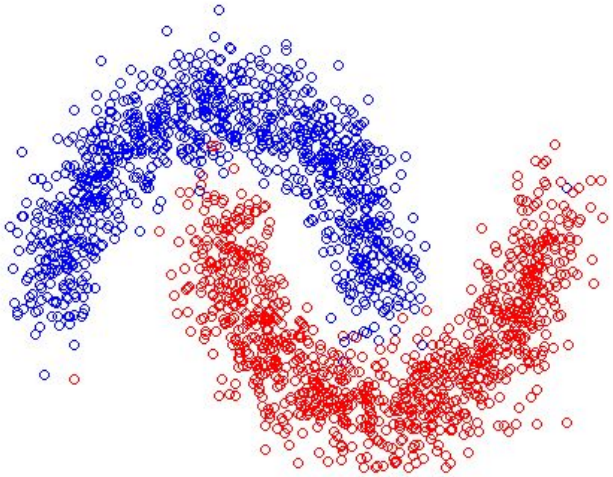
```
kmeans.fit(X)
```

```
kmeans.labels_
```

Problemas para encontrar clusters cuando:

- no tienen forma esférica
- tienen tamaños o densidades muy distintas

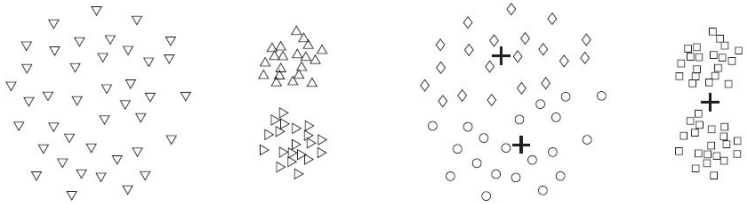




K-means

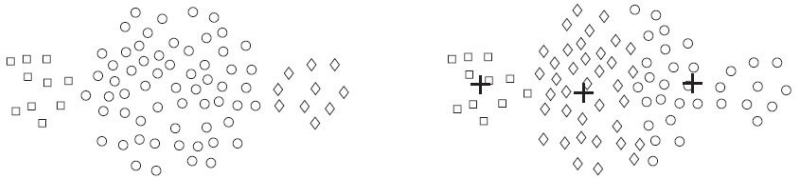
Problemas para encontrar clusters cuando:

- no tienen forma esférica
- tienen tamaños o densidades muy distintas



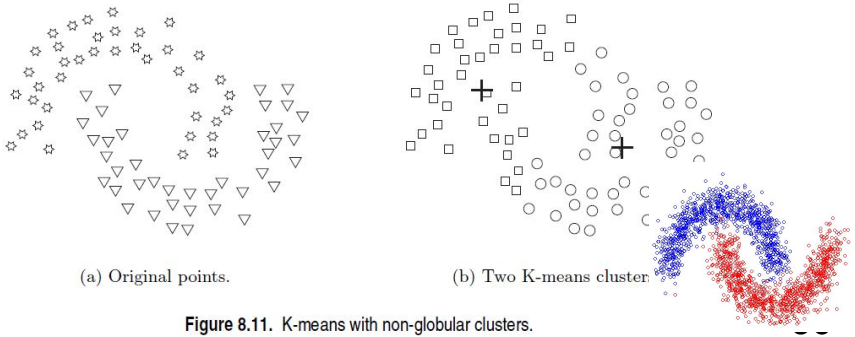
(a) Original points. (b) Three K-means clusters.

Figure 8.10. K-means with clusters of different density.



(a) Original points. (b) Three K-means clusters.

Figure 8.9. K-means with clusters of different size.



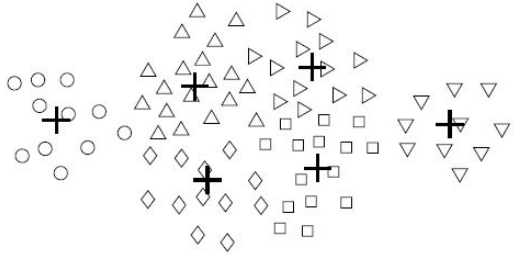
(a) Original points. (b) Two K-means cluster

Figure 8.11. K-means with non-globular clusters.

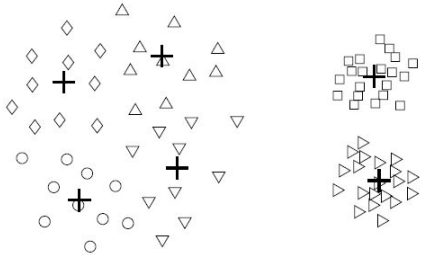
K-means

Función objetivo minimizada por clusters globulares de igual tamaño y densidad.

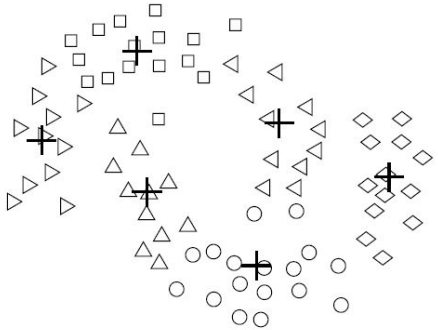
Posible solución: dividir clusters en sub clusters.



(a) Unequal sizes.



(b) Unequal densities.



(c) Non-spherical shapes.

K-medias (K-means)

Ventajas:

- Rápido
- Fácil de implementar
- Útil en general

Desventajas:

- Sensible a ruido y outliers
- Necesita especificar el número de clusters de antemano
- No es bueno cuando los datos tienen formas raras, o cuando hay mucha variabilidad en los tamaños de los clusters
- Sensibilidad a los valores iniciales de los centroides

DBSCAN

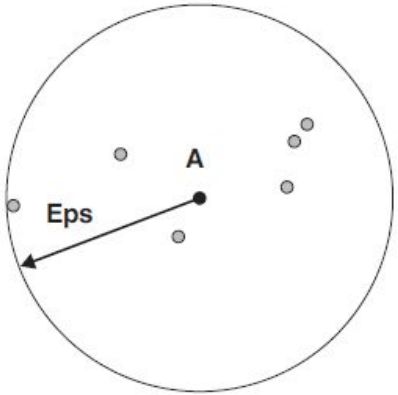
Density-based spatial clustering of applications with noise

DBSCAN

"Density-based spatial clustering of applications with noise"

- **Vecindad** de un punto A: esfera centrada en A y radio Eps.

- **Densidad** de un punto A: cantidad de puntos dentro de la vecindad de A incluyendo a A.



En el ejemplo, Densidad = 7

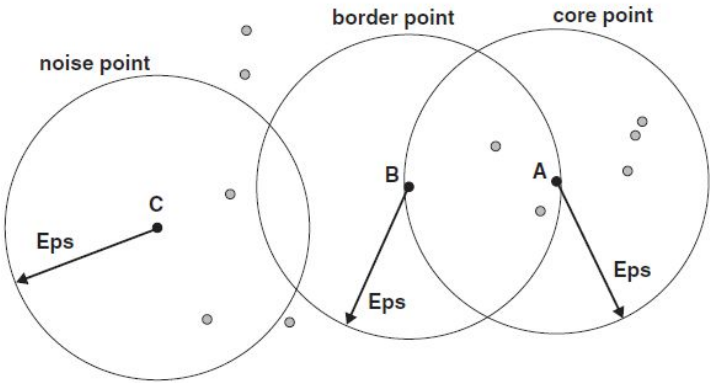
La densidad depende del radio.

DBSCAN

Se clasifica a cada punto como

- **Core point:** puntos con densidad alta (> a una umbral dado: **MinPts**).
- **Border point:** puntos que no son core, pero son vecinos de algún punto core.
- **Noise points:** Puntos que no son core ni border

ej. con minpoints <=7



Algoritmo DBSCAN

Entrada: conjunto de puntos D, Eps, minPts

Algoritmo:

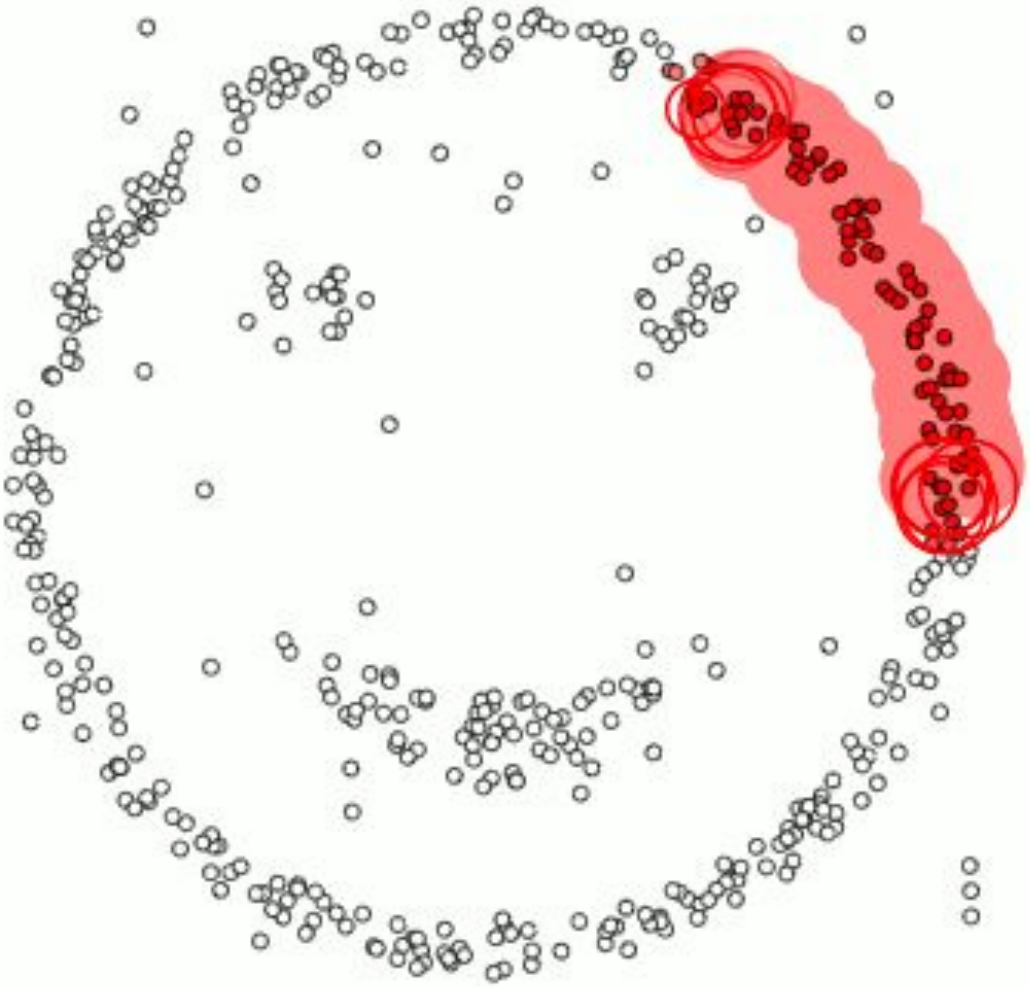
1. Inicializar todos los puntos como **no visitados**.
2. Para cada punto p no visitado:
   Marcar p como visitado.
   - Encontrar vecinos dentro de Eps
   - Si tiene cant. vecinos ≥ minPts →p es **core point** → iniciar cluster y expandirlo agregando todos los vecinos densamente conectados.
   - Si tiene cant vecinos < minPts → marcar temporalmente como **noise**.
   - Durante la expansión:
     i. Puntos que no son core pero están dentro de la vecindad de un core → **border points** → se agregan al cluster.
     ii. Puntos que no se conectan a ningún core → permanecen como **noise**.
3. Repetir hasta que todos los puntos hayan sido procesados..

Algoritmo DBSCAN

Parámetros:

Eps - Distancia para la vecindad
minPts - Cantidad de vecinos requeridos

- Un punto p es un punto *núcleo* si al menos **minPts** puntos están a una distancia menor a **Eps** de él.
- Un punto q es *alcanzable* desde p si hay un camino de puntos alcanzables que va de p a q.
- Un punto que no sea alcanzable desde al menos **minPts** es considerado *ruido*.



DBSCAN con scikit-learn

```
from sklearn.cluster import DBSCAN
```

```
dbclust = DBSCAN(eps=0.5, min_samples=3)
```

```
dbclust.fit(X)
```

Algoritmo DBSCAN

Ventajas:

- No asume ningún número de clusters.
- Puede encontrar clusters con formas geométricas arbitrarias.
- Es robusto detectando outliers.
- No es susceptible al orden en que se encuentran los puntos dentro de la base de datos, ni a la inicialización.

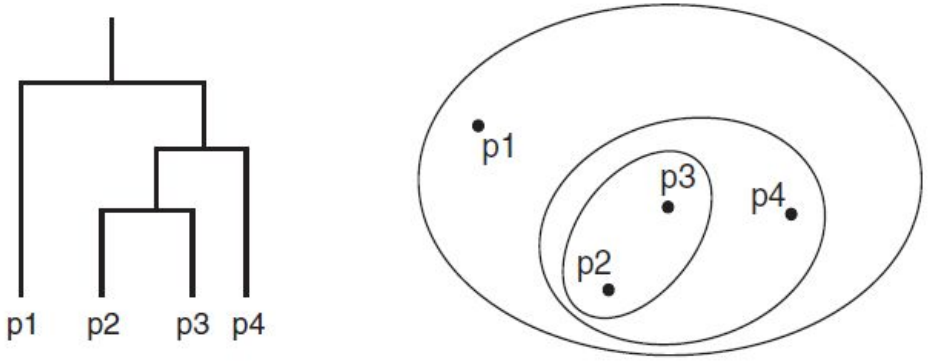
Desventajas:

- No puede agrupar bien conjuntos de datos con grandes diferencias en las densidades.
- Es muy sensible a los parámetros (minPts y Eps) y a veces es difícil determinarlos.
- No es bueno para datasets muy grandes o en muchas dimensiones.

Clustering Jerárquico

Clustering jerárquico

Se suele mostrar en un **diagrama en forma de árbol**, llamado **dendograma**. Muestra clusters, subclusters y el orden en que fueron unidos.



(a) Dendrogram. (b) Nested cluster diagram.

Tan, Steinbach & Kumar, Introduction to Data Mining (Cap. 8)

Algoritmo Clustering Jerárquico

aglomerativo - bottom up

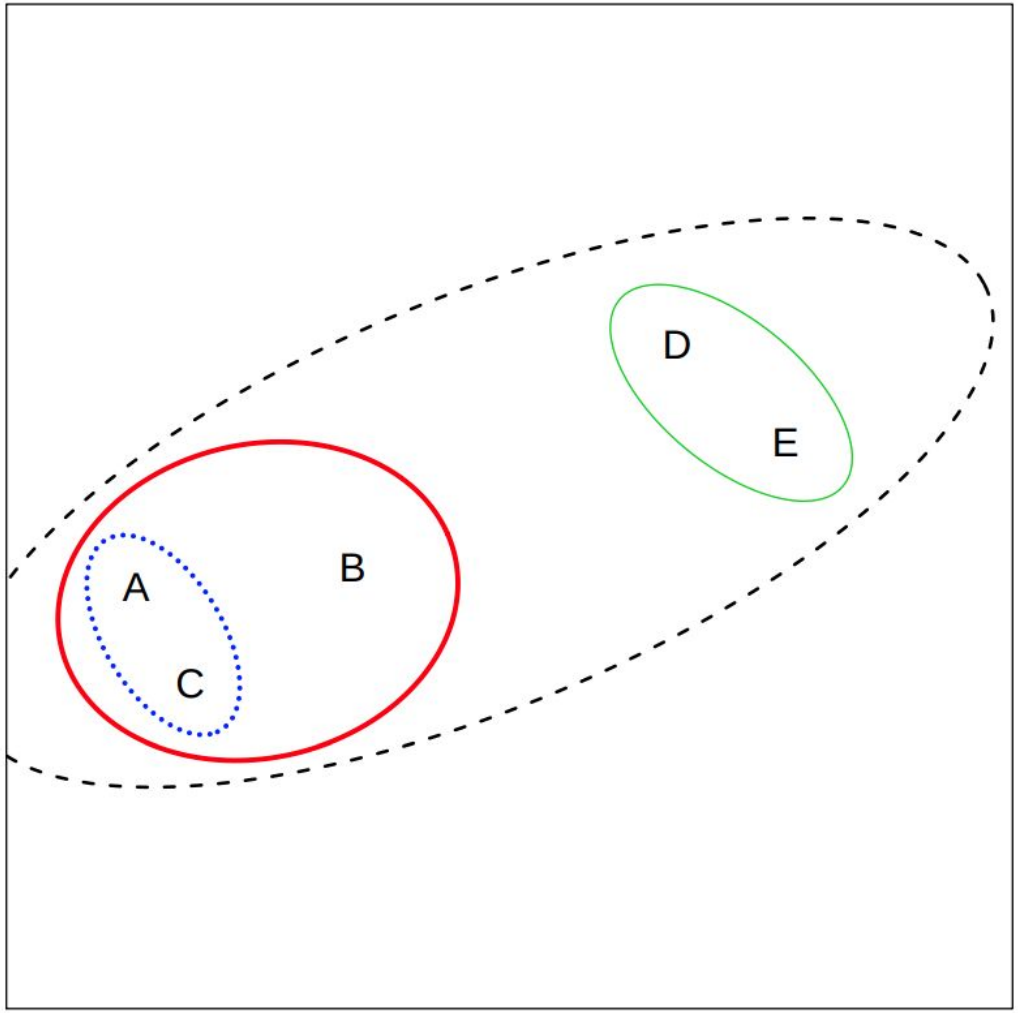
1. Cada punto forma un cluster
2. Computar matriz de *cercanía*
3. Repetir:
   a. Buscar el par de clusters más similar y hacer un merge
   b. Actualizar la matriz de proximidad

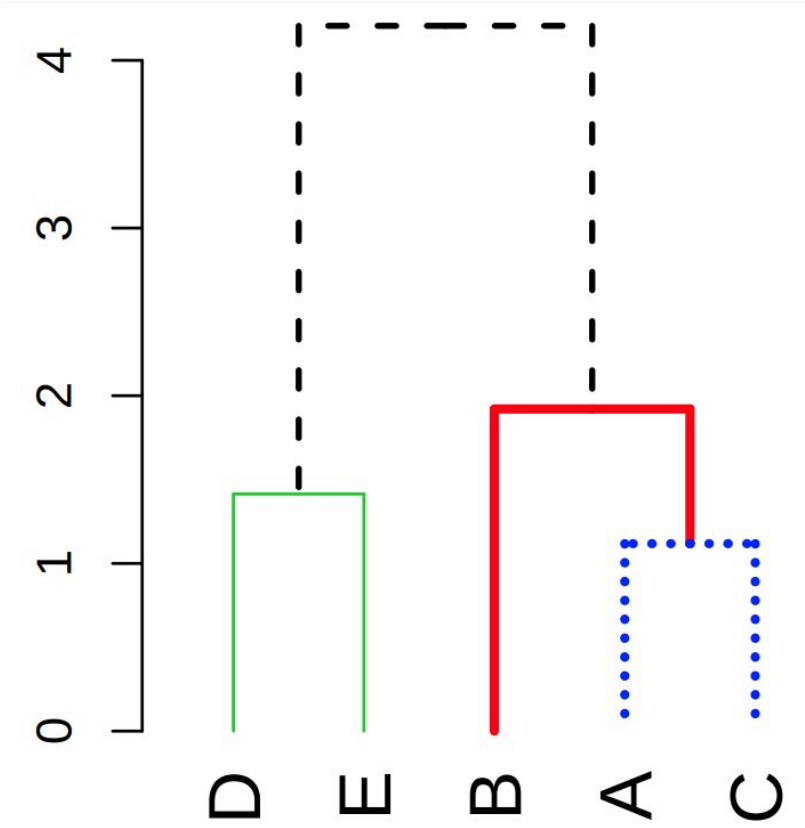
   hasta que haya un solo cluster

Este proceso genera un *dendrograma*.

Ejemplo

del libro ISLP





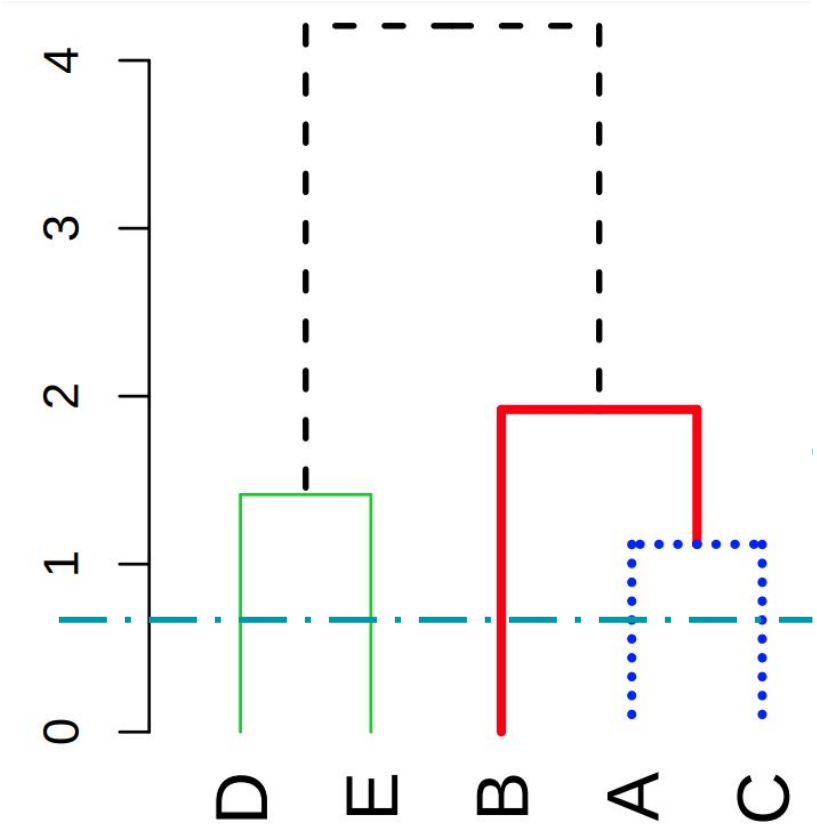
Dendrograma

La *altura* representa la escala a la que se unen los clusters.

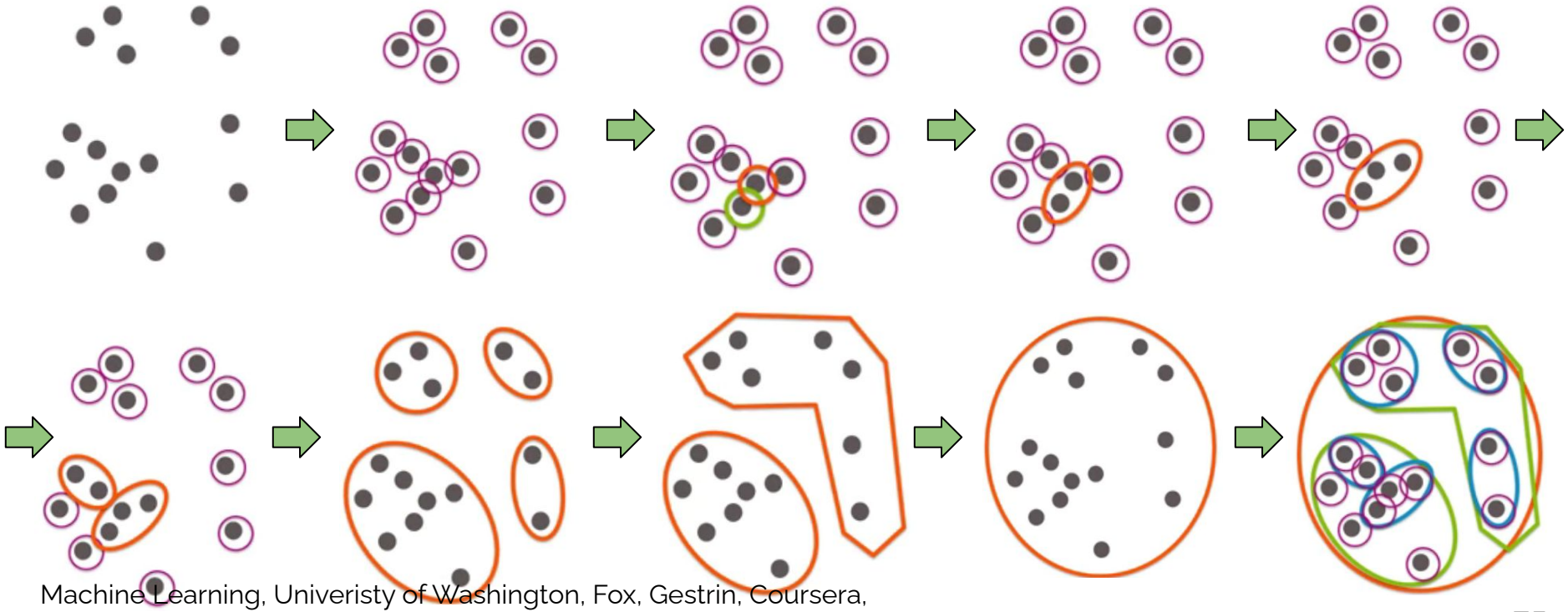
Una vez hecho el dendrograma, se lo puede cortar en una altura y así generar el clustering.

Ejemplo

del libro ISLP

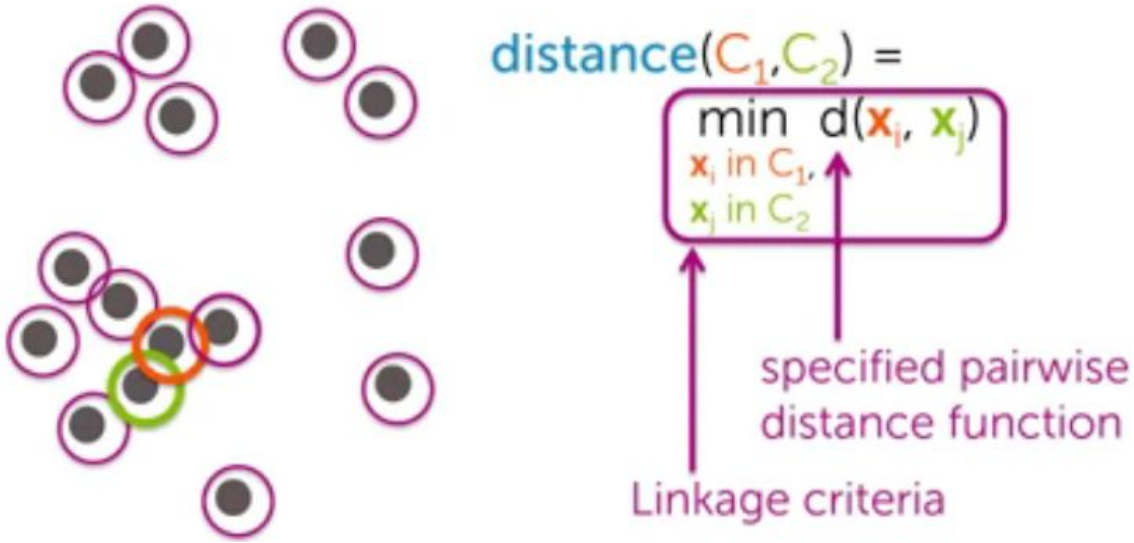


Aglomerativo: single linkage



Machine Learning, Univeristy of Washington, Fox, Gestrin, Coursera,
https://www.coursera.org/lecture/ml-clustering-and-retrieval/agglomerative-clustering-bsFBT

Aglomerativo: single linkage



Single linkage:

uno clusters de menor distancia.

distancia: distancia entre cualesquiera dos puntos más cercanos (pertenecientes a distintos clusters).

Machine Learning, Univeristy of Washington, Fox, Gestrin, Coursera, https://www.coursera.org/lecture/ml-clustering-and-retrieval/agglomerative-clustering-bsFBT

Criterios de cercanía

Entre dos puntos - la cercanía es la distancia (que depende de los atributos). Puede ser la distancia euclídea u otra.

¿Pero entre dos clusters que tienen varios puntos?

Hay que extender la noción de distancia entre puntos, a conjuntos de puntos.

Ejemplo con single linkage

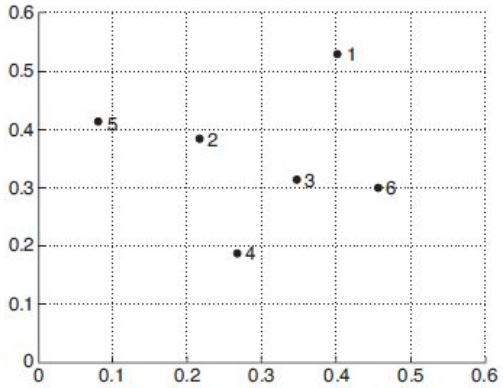


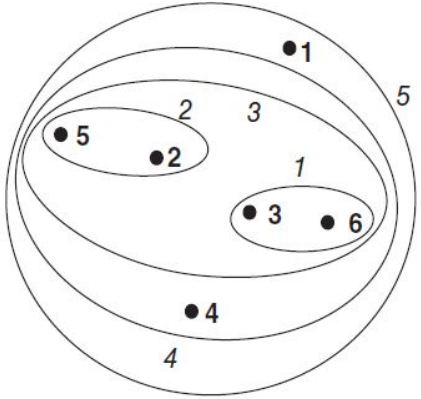
Figure 8.15. Set of 6 two-dimensional points.

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

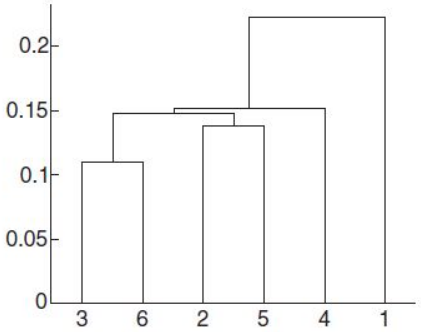
Table 8.4. Euclidean distance matrix for 6 points.

Point	x Coordinate	y Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

Table 8.3. xy coordinates of 6 points.



(a) Single link clustering.



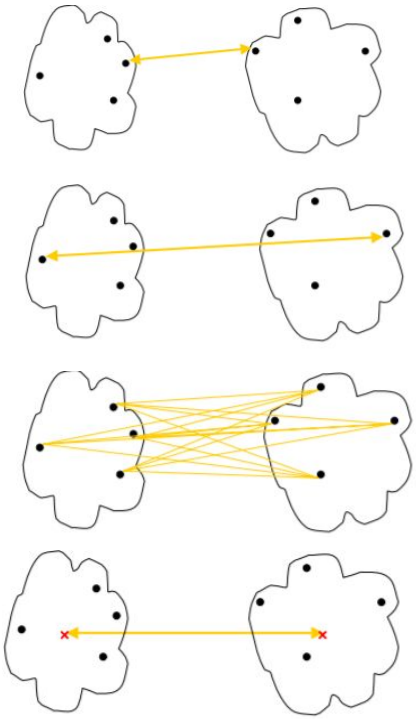
(b) Single link dendrogram.

$$
\begin{aligned}
dist(\{3,6\},\{2,5\}) &= \min(dist(3,2), dist(6,2), dist(3,5), dist(6,5)) \\
&= \min(0.15, 0.25, 0.28, 0.39) \\
&= 0.15.
\end{aligned}
$$

Tan, Steinbach & Kumar, Introduction to Data Mining

Medición de similaridad entre clusters

- **Single linkage:** distancia mínima entre dos puntos de los dos distintos clusters.
- **Complete linkage:** distancia máxima entre dos puntos de los distintos clusters
- **Average linkage:** promedio de la distancia entre los puntos de los clusters
- **Centroid linkage:** distancia entre centroides



Tan, Steinbach & Kumar, Introduction to Data Mining
https://www-users.cs.umn.edu/~kumar001/dmbook/dmslides/chap8_basic_cluster_analysis.pdf

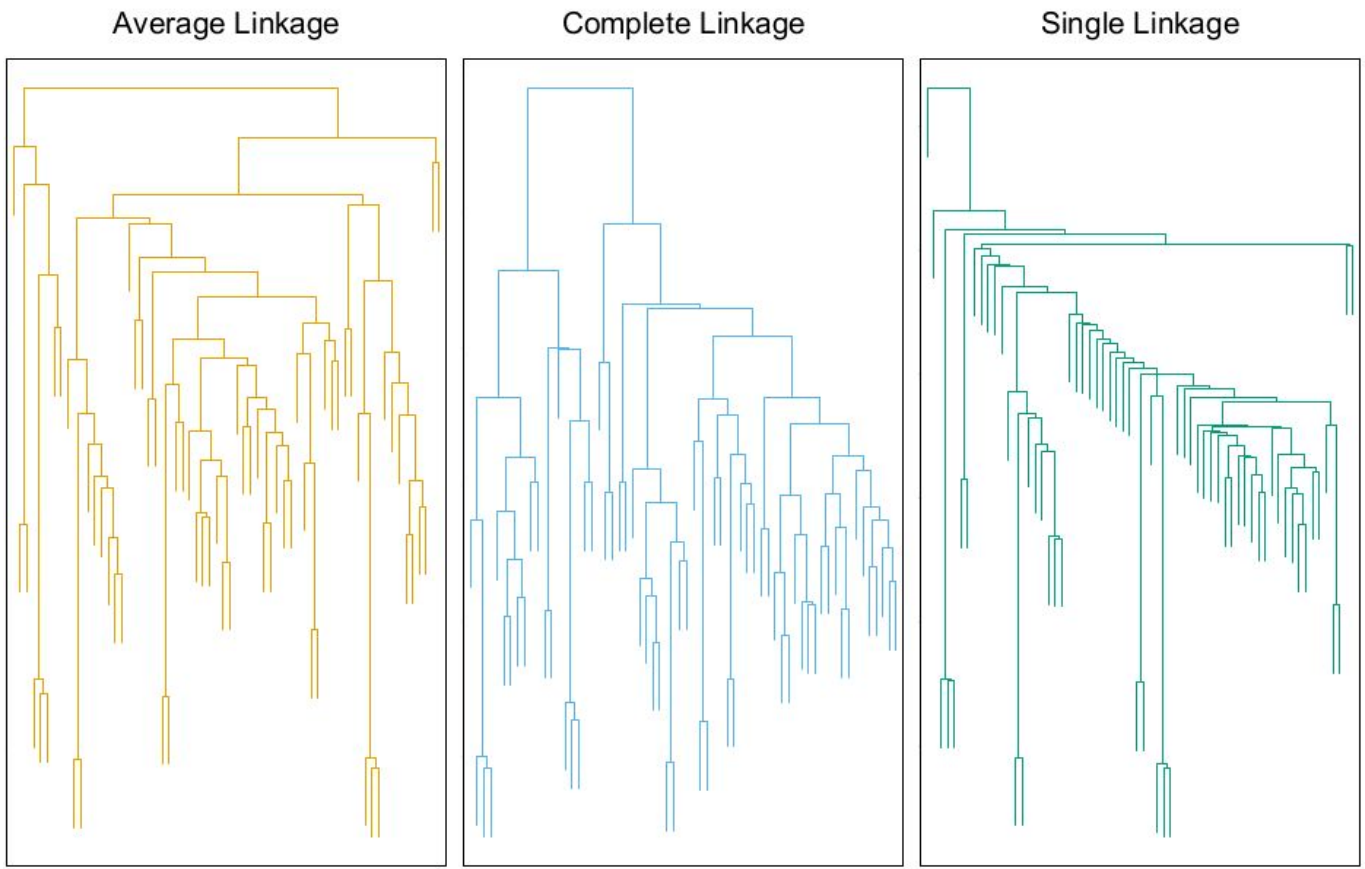


FIGURE 12.14. *Average, complete, and single linkage applied to an example data set. Average and complete linkage tend to yield more balanced clusters.*

del libro
ISLP

Clustering jerárquico

Ventajas:

- No asume ningún número de clusters (se pueden obtener cortando el dendograma en el nivel deseado)
- Genera un dendrograma: puede ser útil para interpretar
- Pueden corresponder a taxonomías (ej. reino animal)

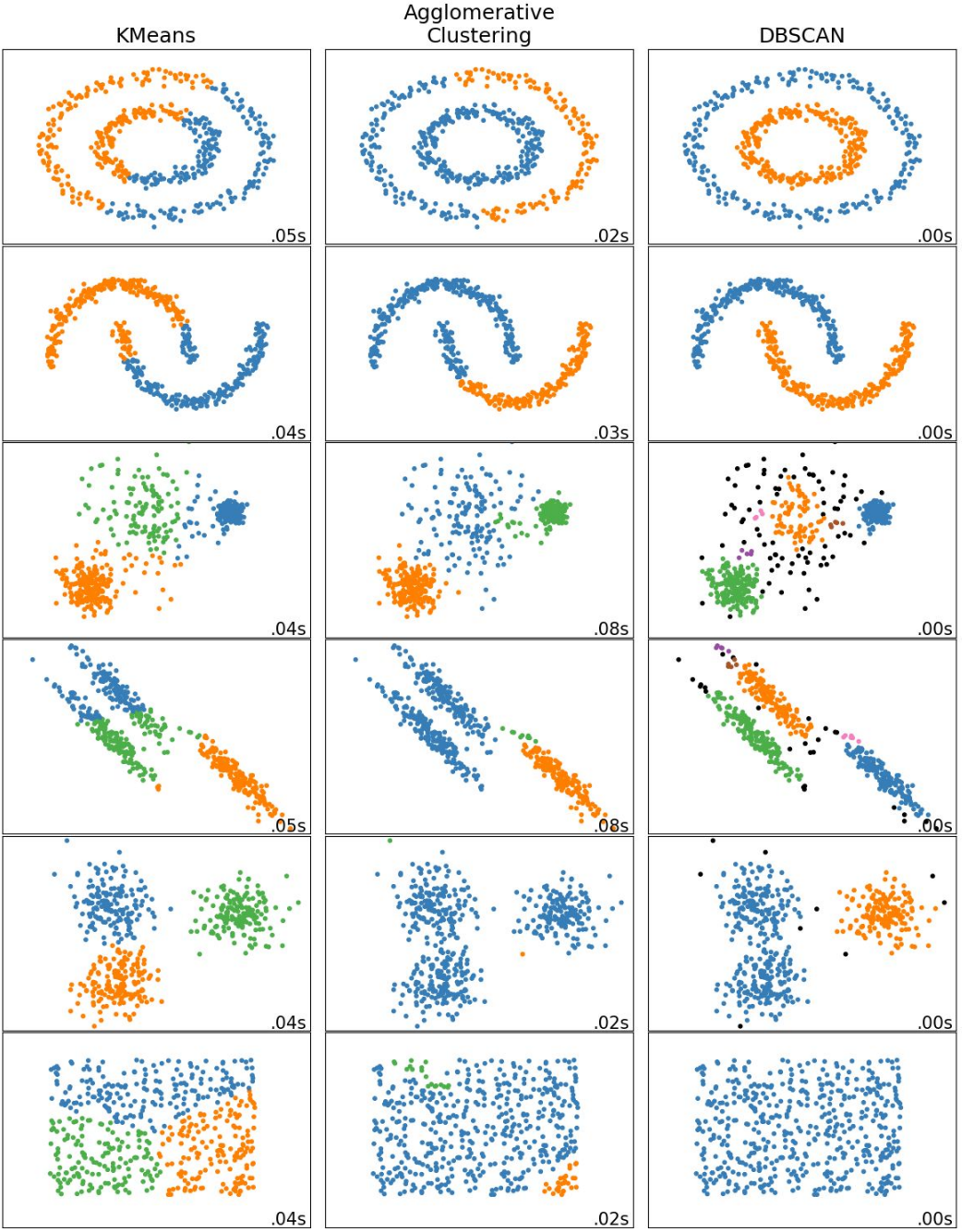
Desventajas:

- Sensible a ruido y outliers
- Computacionalmente caro en tiempo y en espacio
- No siempre la estructura jerárquica es la más adecuada
- Optimiza localmente, no de manera global

Clustering jerárquico con scikit-learn

```
from sklearn.cluster import AgglomerativeClustering
```

```
HClust = AgglomerativeClustering
hc_comp = HClust(distance_threshold =0, n_clusters=None ,
linkage='complete')
hc_comp.fit(X)
```


KMeans
Agglomerative Clustering
DBSCAN
.05s
.02s
.00s
.04s
.03s
.00s
.04s
.08s
.00s
.05s
.08s
.00s
.04s
.02s
.00s
.04s
.02s
.00s

Cierre

1. Aprendizaje No Supervisado
2. KMeans
3. DBSCAN
4. Jerárquico