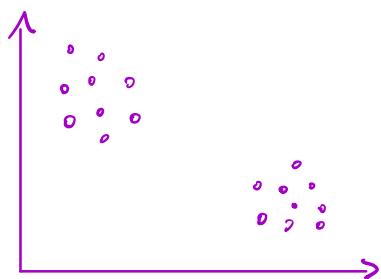


## Clustering

Hasta ahora solo hemos visto técnicas de aprendizaje supervisado, esto es, tenemos la etiqueta de cada instancia o le hace de entrenar. ¿Pero qué pasa si no tenemos etiquetas? En ese caso tendremos que hacer Aprendizaje no Supervisado para descubrir nosotros las etiquetas. Este proceso de clasificar instancias en clases sin saber cuáles son las etiquetas se llame clustering, y es una de las posibles tareas del aprendizaje no supervisado.

## K-means

K-means es uno de los algoritmos de clustering más famosos. Supongamos el siguiente dataset:

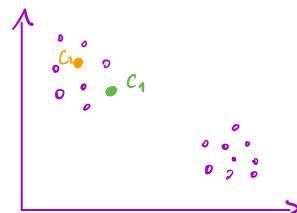


Donde claramente hay dos clusters. Entonces, ¿Cómo funciona el algoritmo?

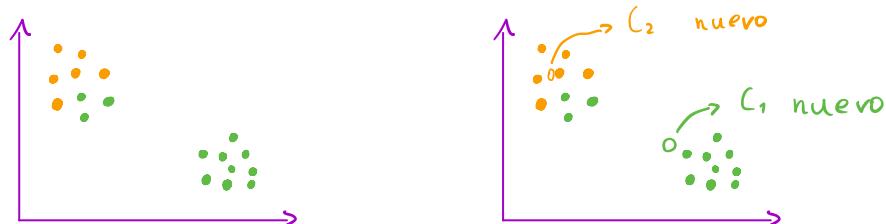
Primero tenemos que indicarle el número de clusters que queremos encontrar. En este caso son 2, pero a veces el número de clusters  $K$  no es claro. El  $K$  de K-means es el número de clusters a encontrar. Discutiremos cómo encontrar  $K$  en el código.

Una vez fijo el  $K$ , buscamos clusters por distancia (trabajaremos con distancia euclídea).

Para el ejemplo presentado, tomamos dos puntos al azar que serán nuestros centroides, cada uno represente a un cluster.

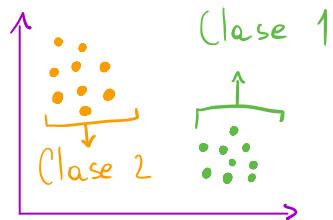


Luego, agregamos cada elemento a su centroide más cercano y luego actualizamos los centroides



Ojo, los centroides se calculan como calcularíamos un centro de gravedad.

Luego repetimos la clasificación con los nuevos centroides, e iteramos hasta que los centroides no cambien en dos iteraciones consecutivas.



Es importante notar que el resultado final depende de la inicialización de los centroides. Así que en general corremos el algoritmo varias veces y nos quedamos con "la mejor solución" que es la que tiene menor inercia: es el promedio de las distancias al cuadrado de cada elemento a su centroide. Ojo, este método sirve para cualquier número de dimensiones.

### Encontrando el K

A veces la separación no es tan clara como en el ejemplo, o no la vamos a poder visualizar. Entonces, ¿Cómo escogemos el K?

Una opción es minimizar la inercia, pero a medida que el K crece, la inercia es menor. De hecho si  $K = \text{número de datos}$  la inercia será 0.

Una buena forma es usar el método del codo. Esto es graficar la inercia para distintos  $K$ , y escoger el  $K$  que se ve como un punto de inflexión:

- $K-1$  tiene una inercia mucho más alta.
- $K+1$  no mejora casi nada.

Veremos esto en el código.

## Otros algoritmos de Clustering

Vamos a discutir dos útiles técnicas de clustering que vale la pena conocer: DBSCAN y clustering jerárquico.

### DBSCAN

Este algoritmo define un cluster como regiones continuas de alta densidad. El algoritmo funciona así:

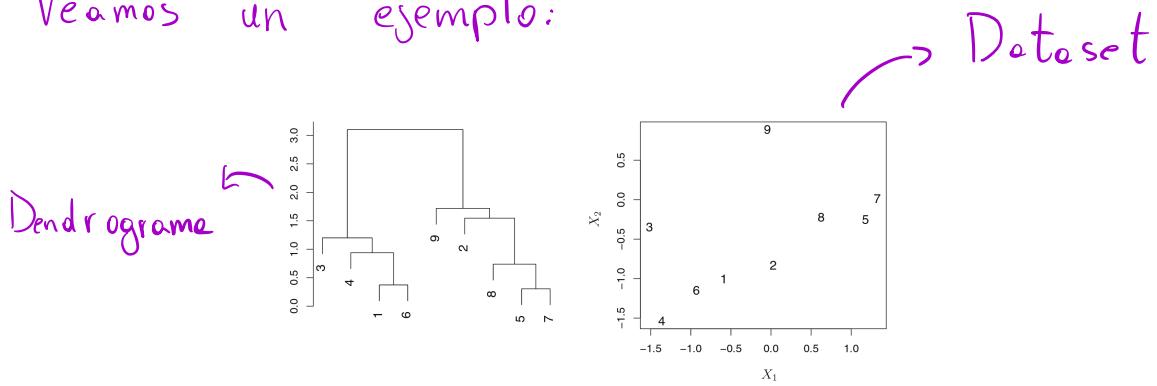
1. Para cada instancia, contamos cuantas instancias están dentro de un rango  $\epsilon$  (E).
2. Si una instancia tiene al menos  $m$  instancias cerca según E, se considere una instancia "core".
3. Todas las instancias en el vecindario de una instancia core pertenecen al mismo cluster. Esta vecindad puede contener otras instancias core. Así, una secuencia larga de instancias core adyacentes forman un cluster.
4. Una instancia no core y que no tiene un core en su vecindad se considere una anomalía.

Este algoritmo no requiere entregar el número de clusters que queremos y además funciona bien si tenemos áreas densas separadas por áreas poco densas.

### Clustering Jerárquico

La última técnica a discutir es clustering jerárquico. La idea es entender la similaridad entre instancias mediante un dendrograma.

Veamos un ejemplo:



El dendrograma de la izquierda representa el dataset de la derecha.

El dendrograma se lee de la siguiente manera:

- Mientras más abajo se junten los elementos, más similares. Por ejemplo la instancia 5 se parece a la 7, la 1 a la 6. También la 8 se parece a la 5 y 7, porque el 8 se junta abajo con el cluster  $\{5, 7\}$ .
- La instancia 9 no se parece a la 2. Es un error común pensar esto ya que estas instancias están al lado, pero esto pasa arriba en el dendrograma. Además podemos decir del dendrograma que la instancia 9 se parece tanto al 2 como al 5, 7 y 8.
- Abajo los elementos se juntan de a pares para formar clusters. Luego, estos clusters se

juntan con otros elementos o clusters para formar más clusters.

Pero, ¿Cómo se forma un dendrograma?

La idea es la siguiente:

1. Partimos con  $n$  observaciones, cada una es un "cluster" de un elemento. También definimos una medida de similaridad (por ej. cuan cerca están según la distancia euclídea).
2. Para cada par de elementos (hay  $\frac{n(n-1)}{2}$  pares) juntamos en el mismo cluster los dos más cercanos. Ahora tenemos  $n-1$  elementos por juntar (los  $n-2$  iniciales restantes y el nuevo cluster de 2 elementos).
3. Iteramos de la misma forma hasta quedar con un cluster.

Ahora bien, ¿Cómo se calcula la similaridad entre un elemento y un cluster o dos clusters?

Hay varias formas:

-Tomar la mayor distancia entre dos elementos de distinto cluster.

- Tomar la menor.
- Tomar el centroide.
- Tomar un promedio.

Así, podemos armar un dendrograma que nos muestre los elementos más similares y menos similares de forma jerárquica.