Machine Learning aplicado a las Ciencias Sociales

Análisis no supervisado

factor~data
EIDAES_UNSAM

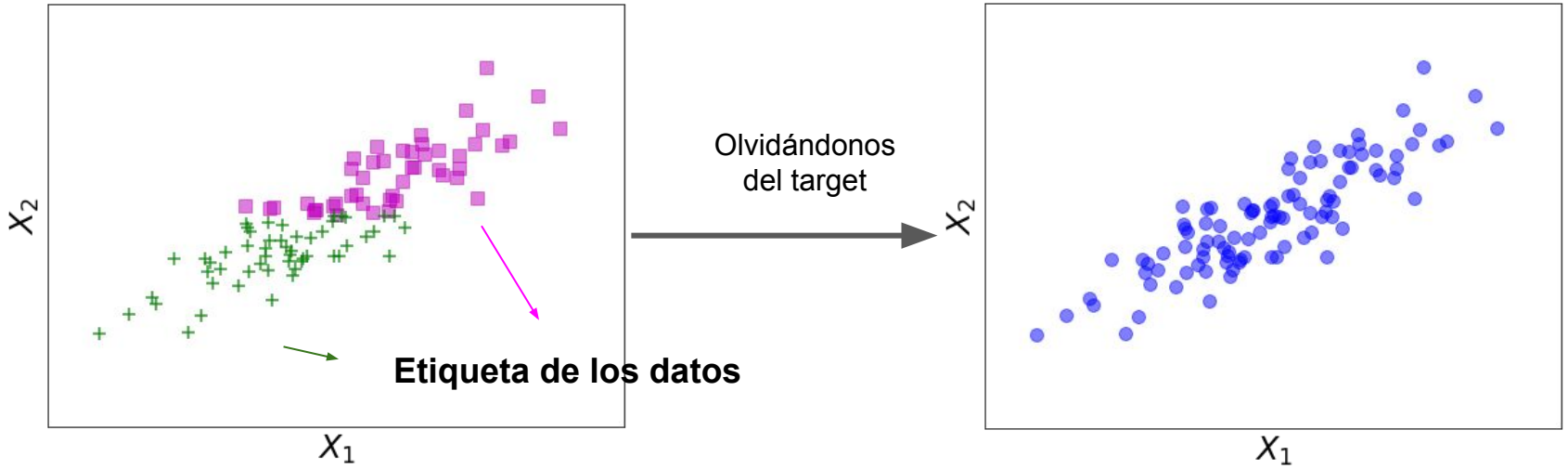
Tipos de problemas

- Aprendizaje Supervisado
  - Variable dependiente => Y, resultado, target
  - Matriz de predictores (p), X, features, variables independientes, etc.
  - Problemas de regresión: Y es cuantitativa
  - Problemas de clasificación: Y es cualitativa
  - Tenemos datos de entrenamiento (conjuntos de Xi,Yi), observaciones
  - Podemos definir una (o varias) métricas para evaluar los modelos

- **Aprendizaje no Supervisado**
  - **No hay variable target (Y)**
  - **Solo hay X**
  - **Es más difícil evaluar qué tan bien funciona el modelo**

¿Qué es el aprendizaje no supervisado?

- El estudio o la exploración de cómo están representados datos y qué conclusiones podemos sacar de dicha representación.



Aprendizaje no supervisado es todo lo que podemos hacer con solo la representación de los datos en el espacio de features.

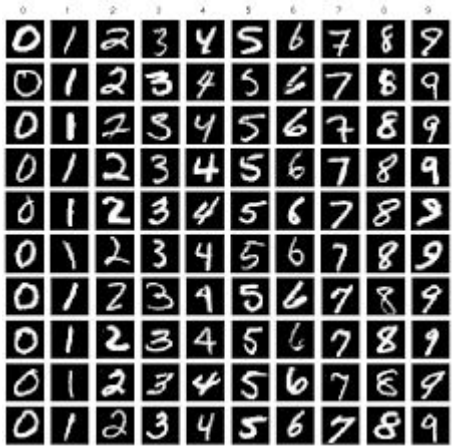
¿Qué podemos hacer solo con las features?

- **Reducción de la dimensionalidad:** encontrar combinaciones de features que reemplacen a los originales para reducir la dimensión del problema. “Reescribir” los datos en una menor cantidad de variables (o dimensiones)

¿Por qué reducir la dimensionalidad del problema?

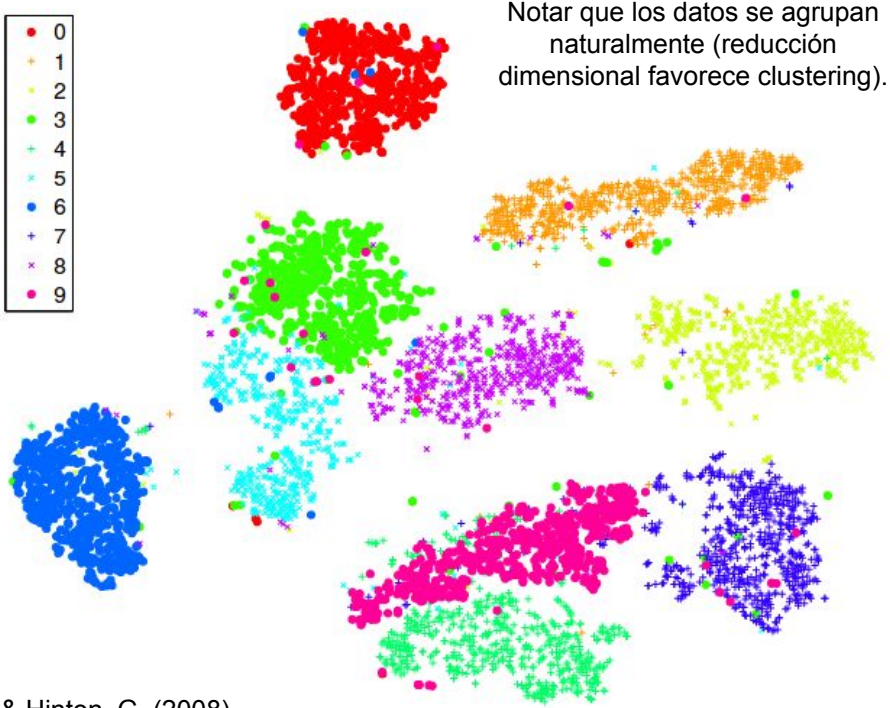
- ¿Todos los features aportan información relevante? ¿Es necesario trabajar con todos?
- Reduciendo la dimensión podemos:
  - Visualizar los datos en el espacio de dimensión reducida, más fácil de interpretar
  - Comprimir la información: nos permite separar la señal del ruido (abstracción)
  - Insumo para clustering: instancias parecidas en un espacio multidimensional son más parecidas en un espacio reducido (emergencia de estructuras).

¿Por qué reducir la dimensionalidad del problema?



MNIST dataset
Datos representados en un espacio de **748 píxeles**.

Visualización de datos multidimensionales en 2D



Notar que los datos se agrupan naturalmente (reducción dimensional favorece clustering).

(a) Visualization by t-SNE.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, *9*(11).

¿Por qué reducir la dimensionalidad del problema?

Compresión (con pérdida) de la información: separación de la señal del ruido.

Con 100 dimensiones (<< 4096) ya logramos una reproducción fiel de la imagen original.

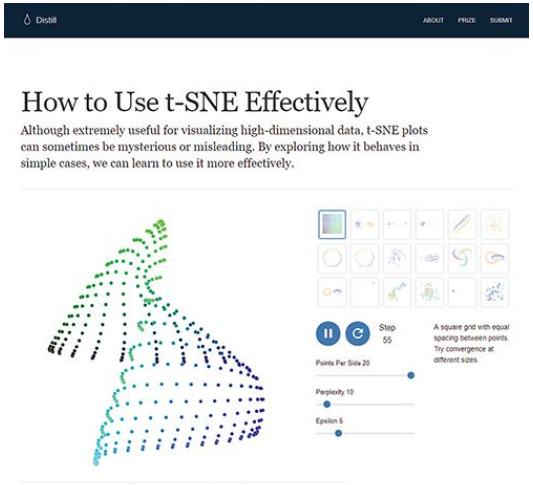
Datos representados en un espacio de **4096 píxeles**.

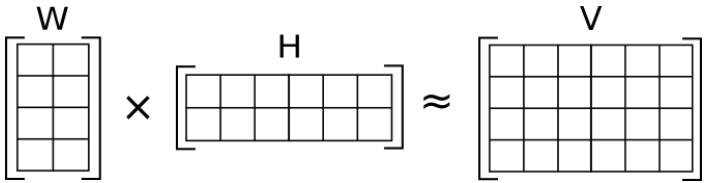
\+ dimensiones del espacio reducido

Imagen reconstruida de un espacio de dimensión reducida

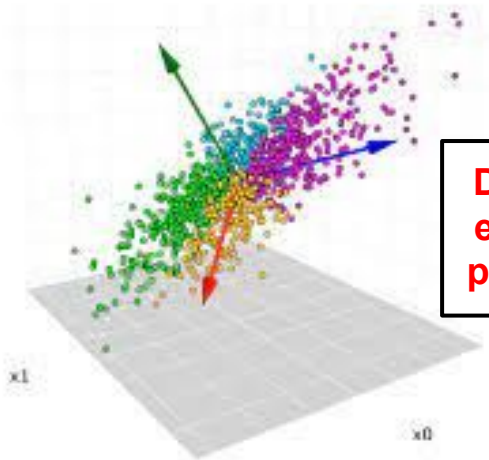
factor~data
EIDAES_UNSAM

Algunos métodos de reducción dimensional

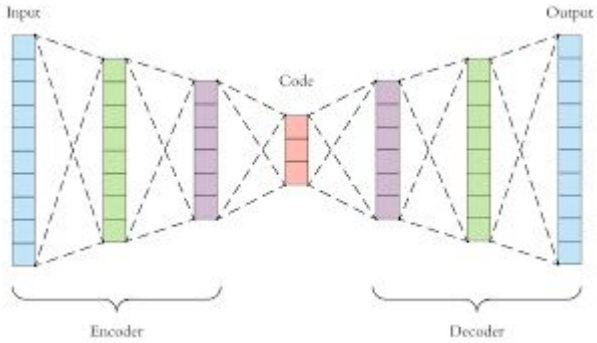




Non-negative matrix factorization (NMF)



Descomposición en componentes principales (PCA)

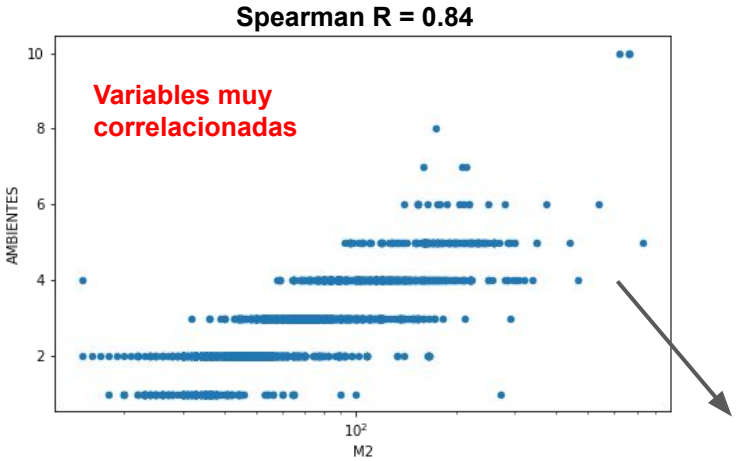


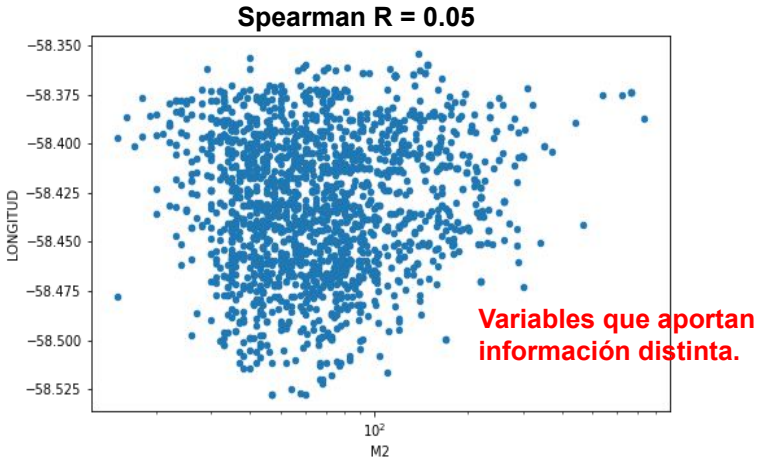
Autoencoders

factor~data
EIDAES_UNSAM

Análisis de componentes principales (PCA)

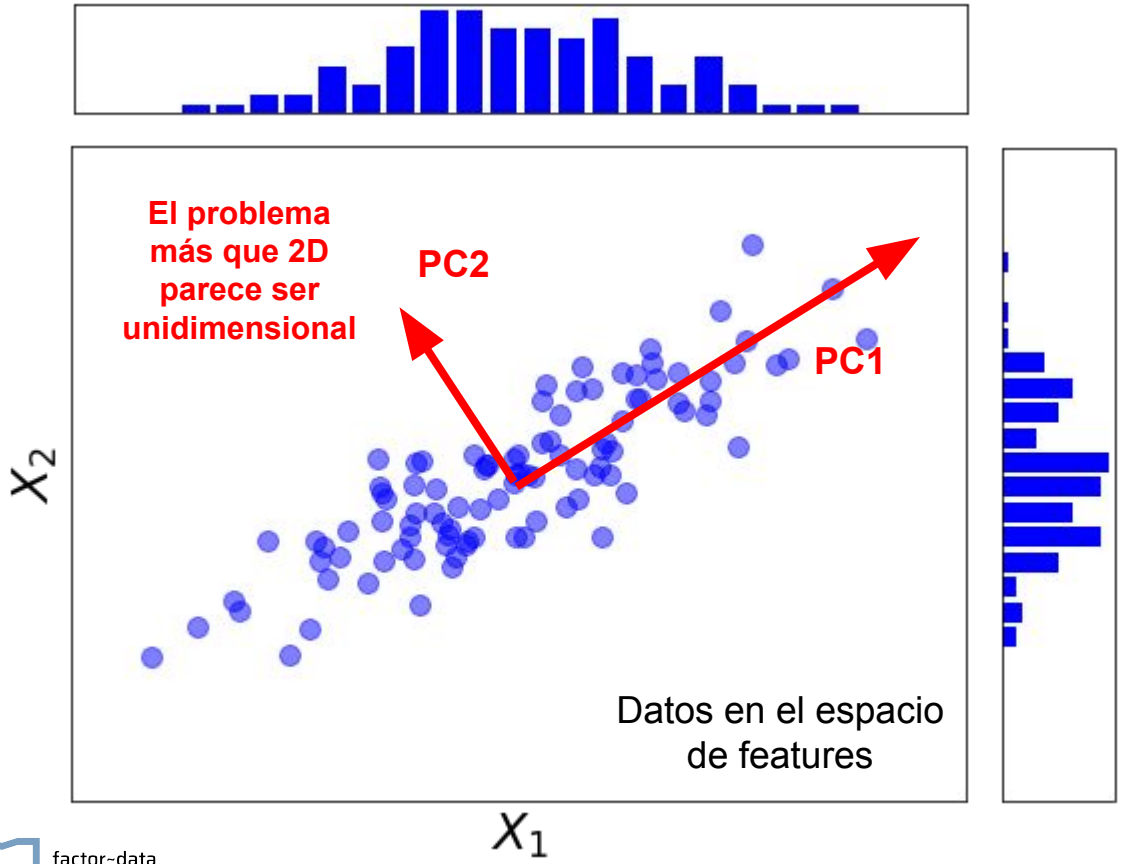
Problema: muchas veces hay variables que contienen prácticamente la misma información que otras (están muy correlacionadas entre sí), por lo que agregan una dimensión más al problema sin aportar muchas más información.





Una opción para solucionar el problema sería tirar una de las dimensiones por redundante, sin embargo ¿estamos seguros que toda la información que tiramos no es necesaria? ¿Hay alguna otra forma más inteligente de tirar dimensiones?

Esquema general

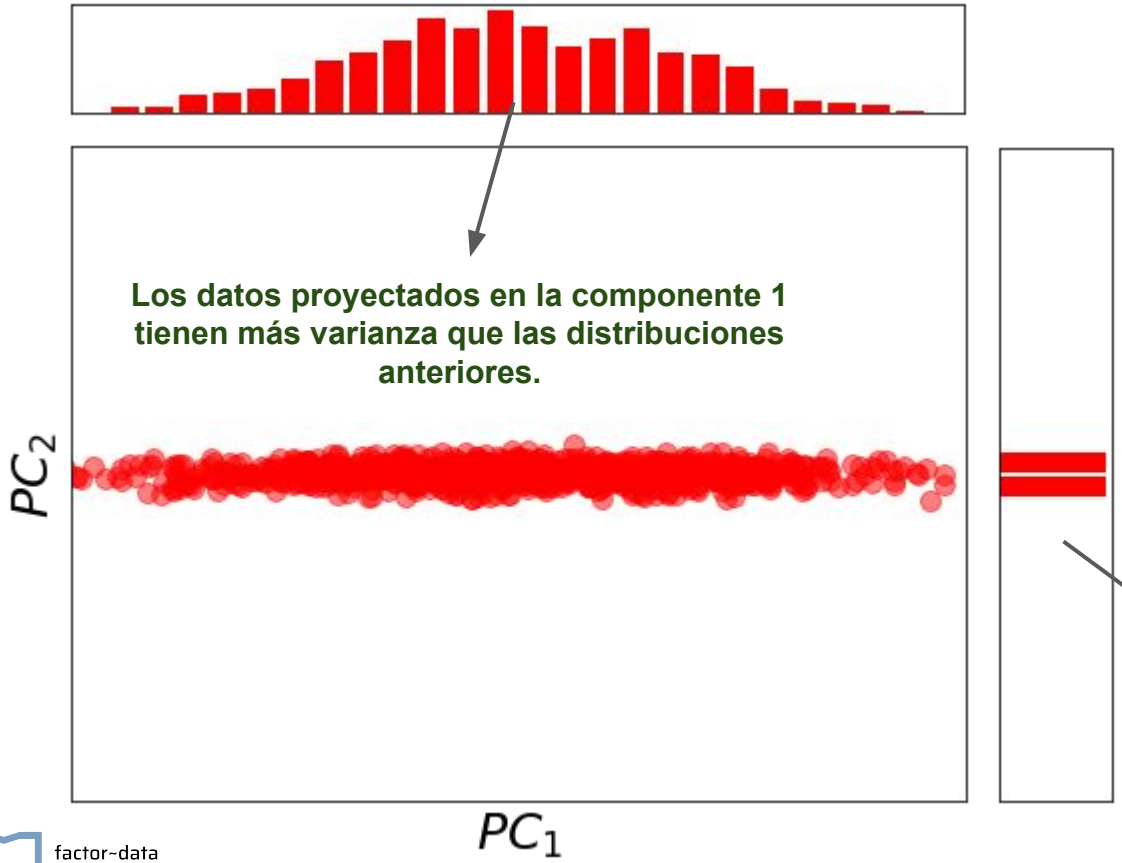


Podemos ver cómo se distribuyen los datos proyectando sobre cada uno de los features.

De las distribuciones podemos calcular su varianza, como una medida de qué tan dispersos están los datos en esa dirección.

Sin embargo, notamos direcciones (combinación lineal de features) donde los datos parecen variar más.

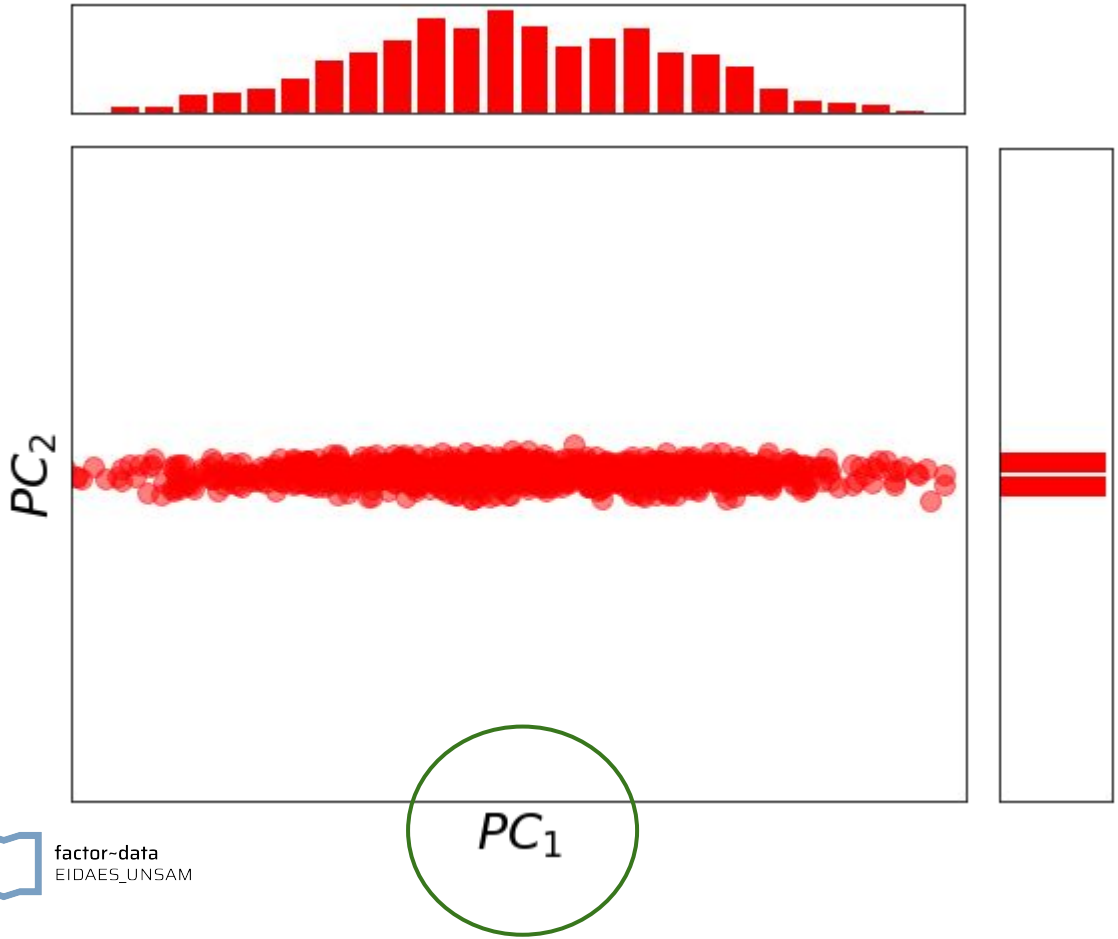
Esquema general



Proyectando en esas nuevas direcciones (ortogonales) volvemos a ver la distribución de los datos proyectadas en las mismas.

La componente 2 tiene mucha menos variación (parece ser más ruido que información importante)

Esquema general



Notar que si nos olvidamos de la componente 2, no perdemos tanta información como si hubiésemos tirado alguno de los features originales.

Podemos reducir nuestro problema de 2 dimensiones a una sola.

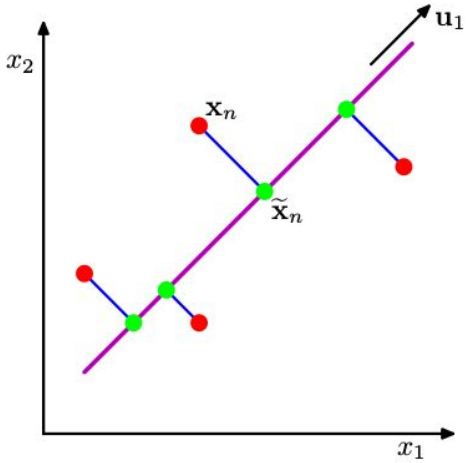
Las direcciones que se llevan las mayor cantidad de varianza de los datos se llaman **componentes principales**

PCA: descripción matemática

Dado un conjunto {xn} de N datos en un espacio de dimensión D (cada xn es un vector en el espacio de D dimensiones).

Las M < D componentes principales son:

- las M direcciones que maximizan la varianza de las proyecciones en ese subespacio,
- o, equivalentemente, las M direcciones que minimizan el error en la proyección (figura).



PCA: descripción matemática

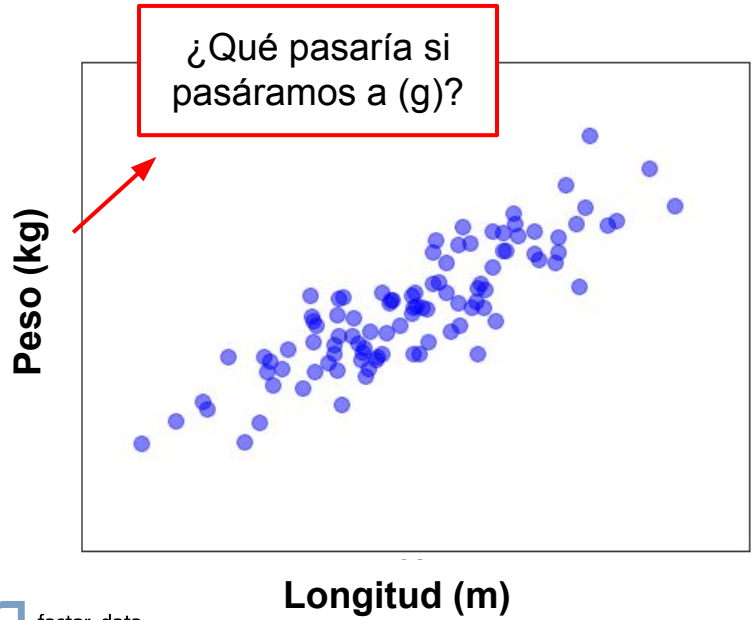
No nos vamos a detener en detalle en los diferentes algoritmos que se utilizan para estimar los componentes. Solo diremos que uno de ellos lleva a que podemos obtener los componentes principales a partir de la descomposición de la matriz de covarizanza **S**:

$$\bar{\bar{S}}\bar{u} = \lambda\bar{u}$$

- Las componentes principales son ortogonales.
- Ordenando las componentes desde el autovalor más grande al más chico, las primeras componentes llevan la mayor varianza de los datos (es lo que efectivamente hace PCA).

Dependencia de las unidades y escaleo

¿Qué pasa con la variabilidad si cambiamos las unidades de alguna de las variables?

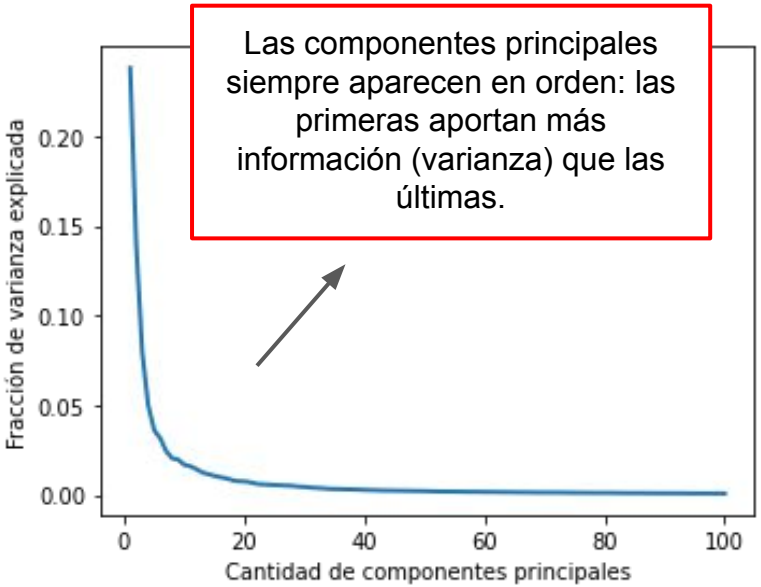


Para evitar introducir o sacar variabilidad por cambiar las unidades de alguna de las variables, una práctica habitual antes de hacer PCA es estandarizar las variables:

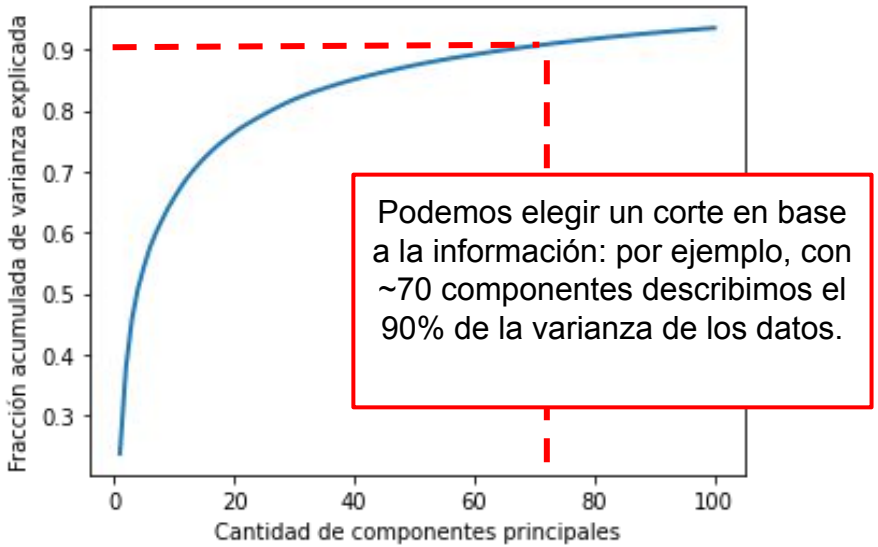
$$Z = \frac{X - \mu}{\sigma}$$

¿Cuántas componentes?

Al ser un problema de aprendizaje no-supervisado, no tenemos un conjunto de test para validar el número de componentes que elijamos. ¿Con cuántas nos quedamos?



Fracción de varianza que aporta cada componente



Fracción de varianza acumulada

Reducción dimensional como input de otros modelos

Podemos usar las primeras componentes principales (Zm) como variables independientes en modelos de regresión o clasificación.

$$y \sim \beta_0 + \beta_1 Z_1 + \ldots + \beta_M Z_M$$

Ventajas:

- Resolvemos el problema de colinealidad: En PCA las variables son por defecto ortogonales y por lo tanto buenas candidatas.
- Podemos validar nuestro modelo y por lo tanto tomar la cantidad de componentes principales como un hiperparámetro.

Resumen de PCA

- Las componentes principales son una combinación lineal de los features originales y se corresponden con los autovectores de la matriz de covarianza de los datos.
- Las componentes están ordenadas de mayor a menor, en el sentido de la información (varianza) que se llevan. Si tiramos las últimas componentes, estamos reduciendo la dimensión de nuestro problema.
- Una práctica usual es estandarizar las variables antes de aplicar PCA (fundamental si los features corresponden a distintas magnitudes).
- Podemos usar las componentes principales para: comprimir la información, visualizar datos multidimensionales en bajas dimensiones, y como input para modelos de aprendizaje supervisado.

¿Qué podemos hacer solo con las features?

- **Clustering:** agrupar casos “parecidos”, que tengan una combinación de valores **similar** en las variables independientes (espacio de features) (próxima clase). Buscamos subconjuntos de datos muy parecidos entre sí.

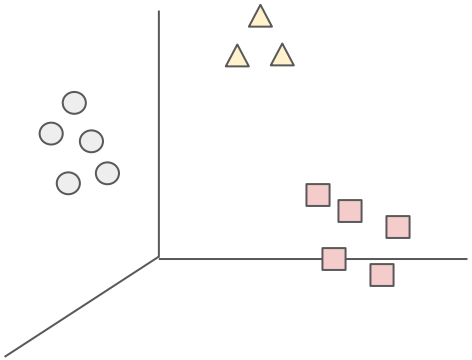
Clustering

- Amplia clase de métodos que buscan detectar grupos y subgrupos en los datos

    - Dado un conjunto de clientes, ¿es posible encontrar segmentos de clientes similares, según su historial de compras?
    - Agrupar películas en función de sus clasificaciones
    - Encontrar grupos de jurisdicciones en función de características sociodemográficas
    - Etc...

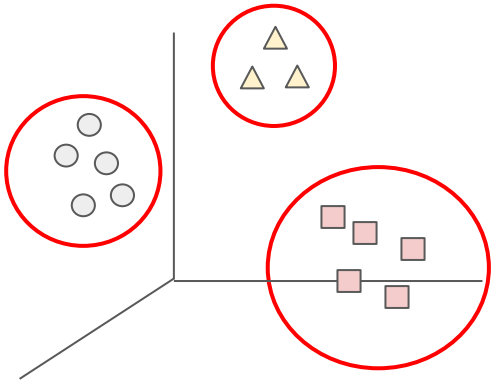
Clustering

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



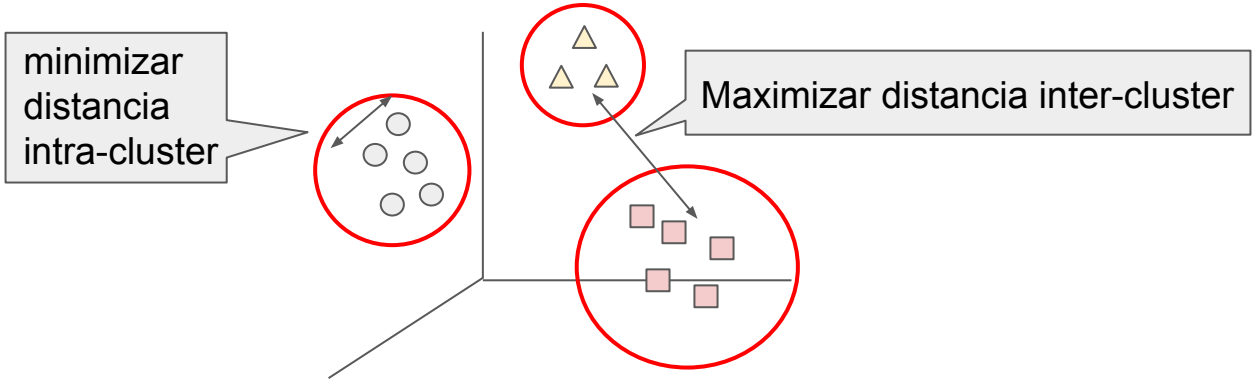
Clustering

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



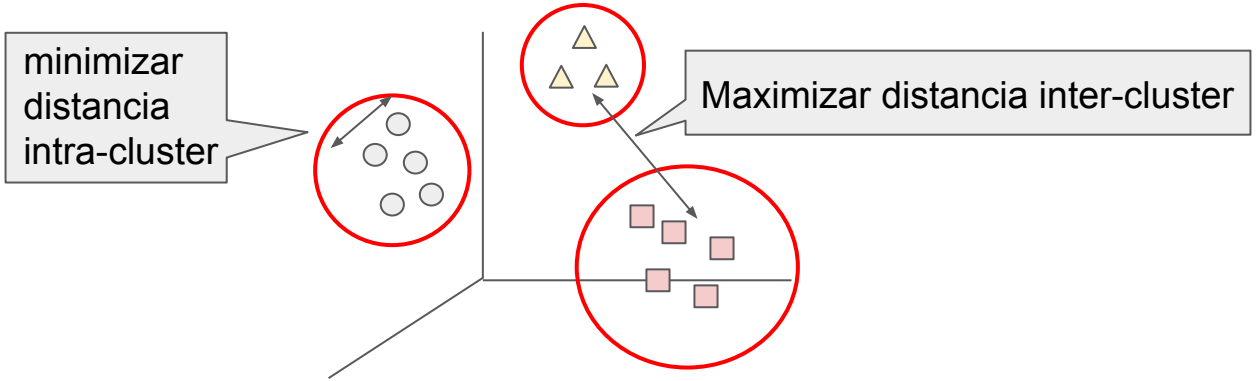
Clustering

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



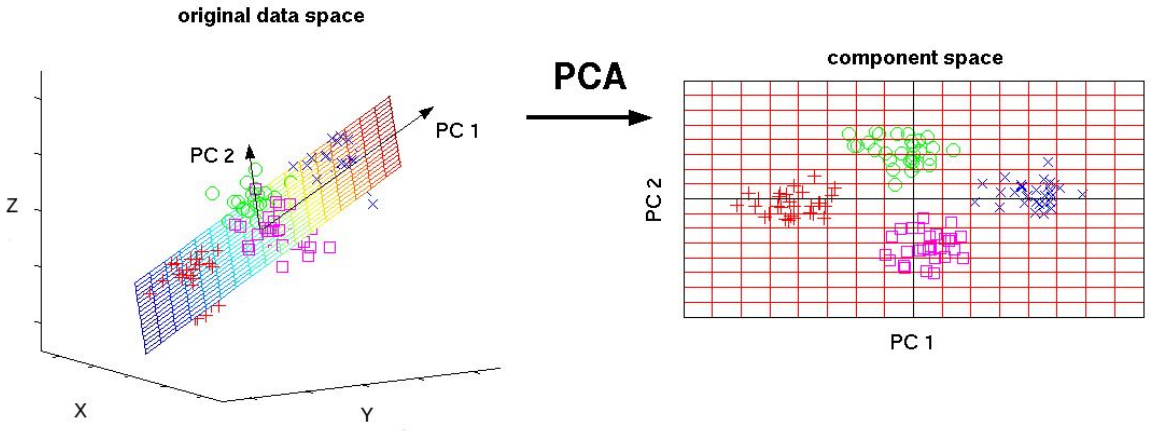
Clustering

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



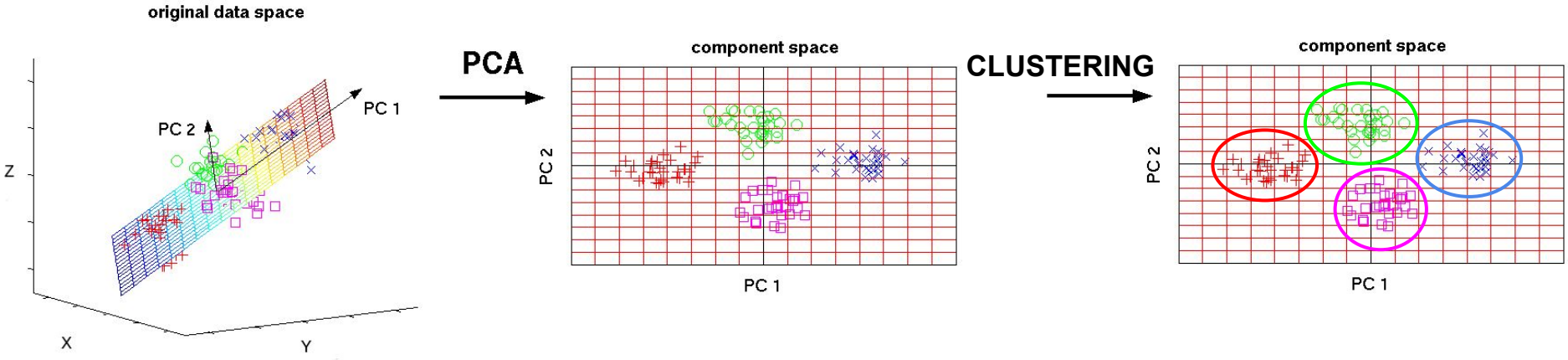
Observaciones dentro de un cluster **similares**
Observaciones entre clusters **no similares**

Clustering - Diferencia con PCA



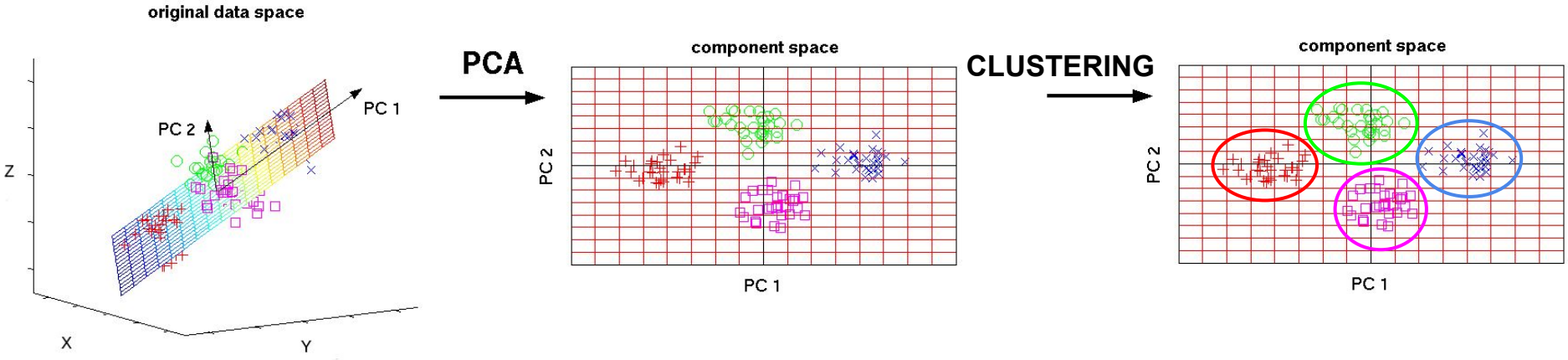
Reducir dimensión maximizando la varianza

Clustering - Diferencia con PCA



Reducir dimensión maximizando la varianza

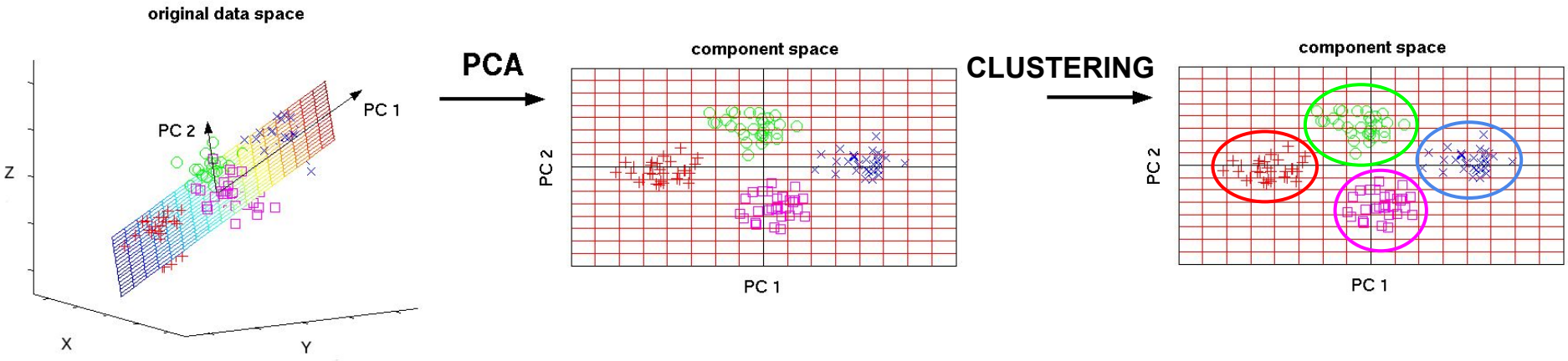
Clustering - Diferencia con PCA



Reducir dimensión maximizando la varianza

Encontrar grupos homogéneos

Clustering - Diferencia con PCA



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

Tipos de problemas

- Buscamos encontrar grupos de casos que sean parecidos entre sí
- Es necesario definir qué significa *similitud* o *diferencia*
- El conocimiento del dominio es fundamental en esta definición

Dos métodos (entre muchos otros)

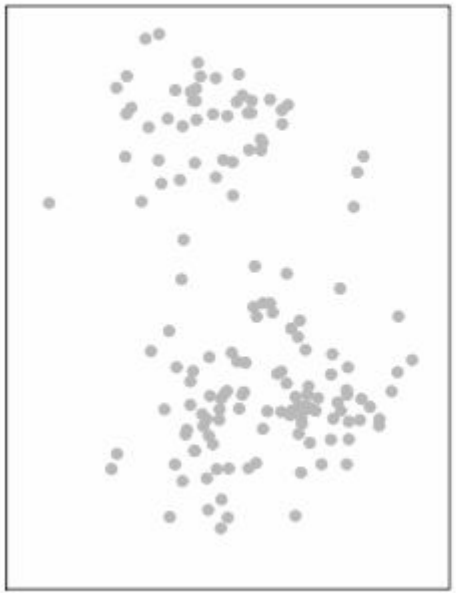
- K-medias

Encontrar una partición de las observaciones según un número pre-definido de clusters

Dos métodos (entre muchos otros)

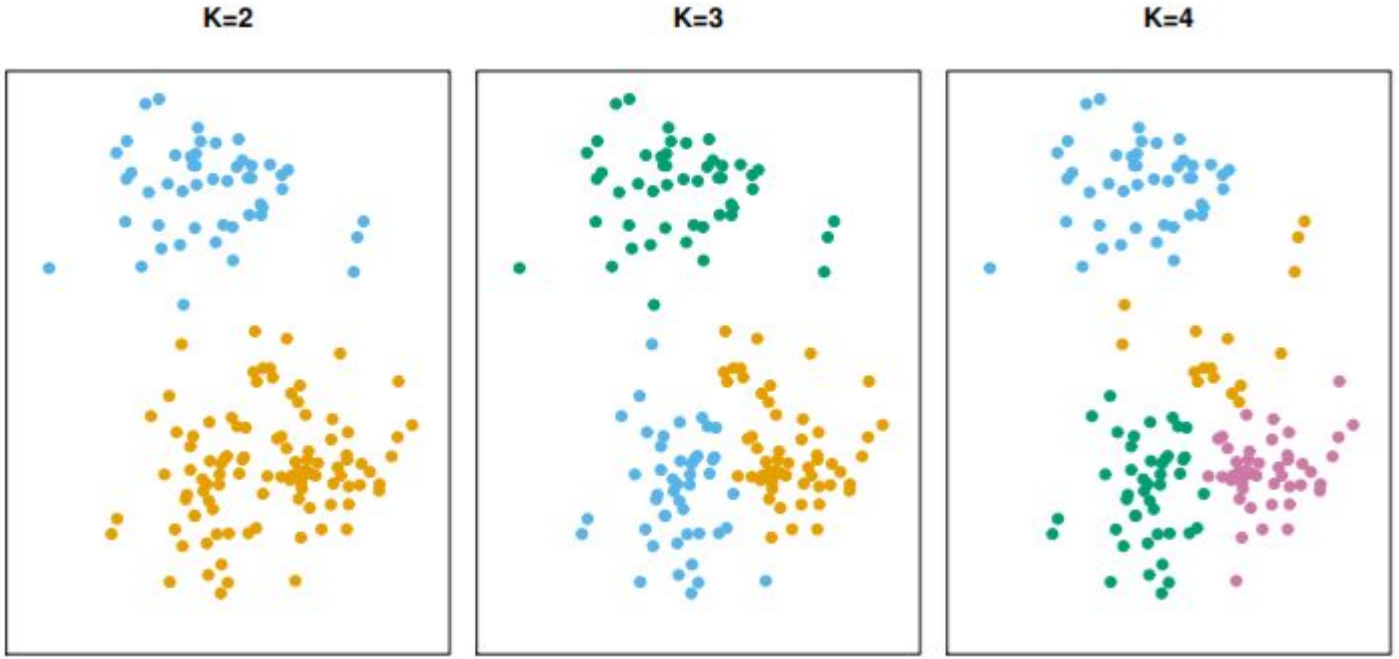
- Clustering jerárquico
  - No conocemos a priori el número de clusters
  - Usamos el *dendograma* para definirlo

K-Medias



- Cada puntito es una persona, caracterizada por su experiencia laboral (eje X) y su ingreso (eje Y)
- ¿Cuántos grupos existen?

K-Medias



K-Medias

- Tenemos K grupos C_1, C_2,..., C_K
- Cada grupo tiene dos propiedades:
  - $C_1 \cup C_2 \cup \ldots \cup C_K = \{1, 2, 3, \ldots, n\}$, es decir que **cada unidad pertenece a un cluster**
  - $C_1 \cap C_2 \cap \ldots \cap C_K, = \emptyset$ es decir, que **cada elemento queda clasificado en un solo cluster**

K-Medias

- La idea es que un buen esquema de clustering implica que adentro del cluster existe la menor variabilidad posible
- Variabilidad intra-cluster WCV(C_k): medida que indica cuánto se diferencian los elementos al interior de un cluster

$$\underset{C_1, C_2, \ldots, C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} WCV(C_k) \right\}$$

K-Medias

- ¿Cómo definimos la medida de variación intra cluster -WCV-? Típicamente, la distancia euclidiana:

$$WCV(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} \left(x_{i,j} - x_{i',j} \right)^2$$

- Entonces, queremos minimizar (**nuestra función objetivo**)

$$\underset{C_1, C_2, \ldots, C_k}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} \left(x_{i,j} - x_{i',j} \right)^2 \right\}$$

K-Medias - Algoritmo

1. Asignar aleatoriamente cada observación a un cluster (entre 1 y K). Esto sirve como un punto de inicialización
2. Repetir hasta que la asignación deje de cambiar:
    2.1 Para cada uno de los K clusters, computar el centroide (el vector que contiene el promedio de cada una de las variables a considerar)
    2.2 Computar la distancia euclidiana de cada caso a cada centroide y asignar cada caso al cluster con el centroide más cercano

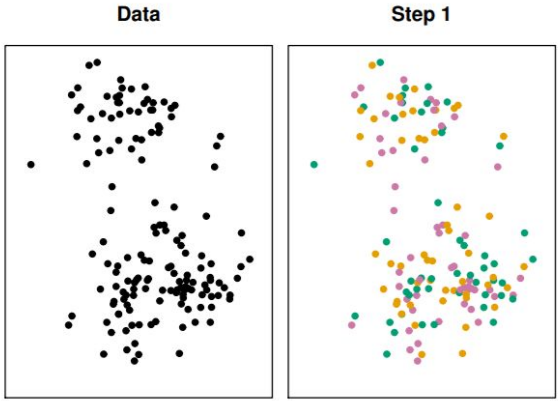
K-Medias - Algoritmo

Definimos el número de clusters que buscamos (k)

K-Medias - Algoritmo

Inicializamos los centroides de forma aleatoria

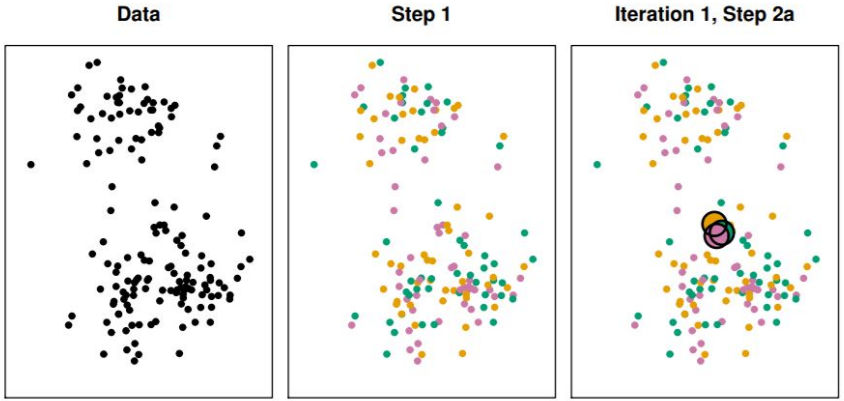
Definimos el número de clusters que buscamos (k)



K-Medias - Algoritmo

Inicializamos los centroides de forma aleatoria

Definimos el número de clusters que buscamos (k)

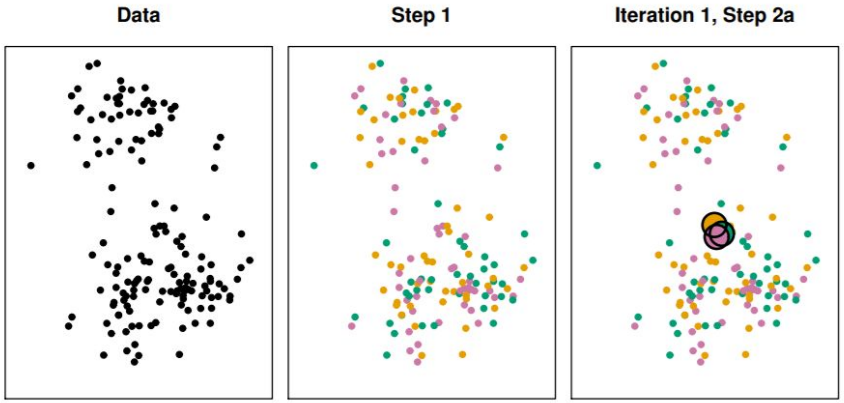


Calculamos los **centroides** (centros) de cada cluster como el promedio de las features de sus samples

K-Medias - Algoritmo

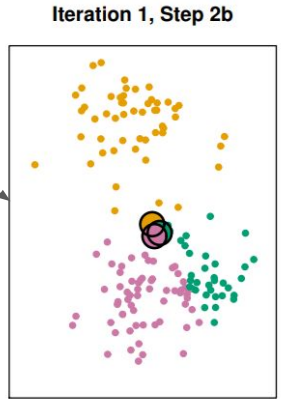
Inicializamos los centroides de forma aleatoria

Definimos el número de clusters que buscamos (k)



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

Asignamos cada caso al cluster con centroide más cercano

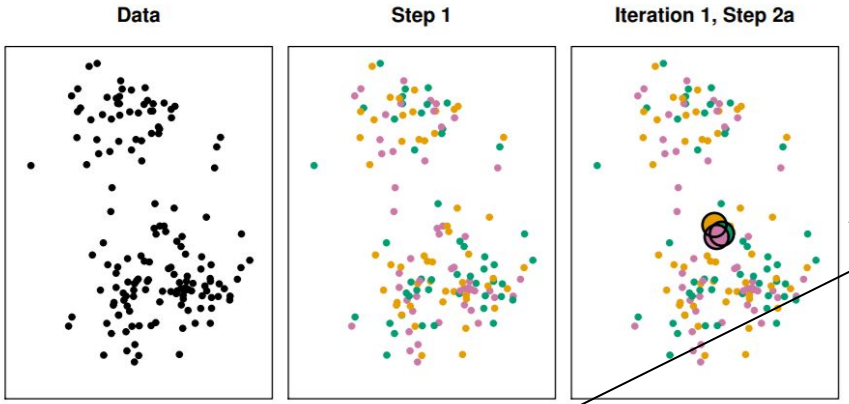


factor~data
EIDAES_UNSAM

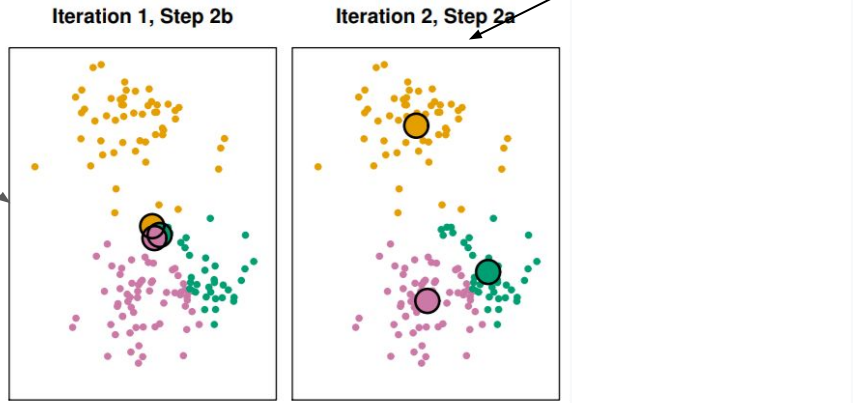
K-Medias - Algoritmo

Inicializamos los centroides de forma aleatoria

Definimos el número de clusters que buscamos (k)



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



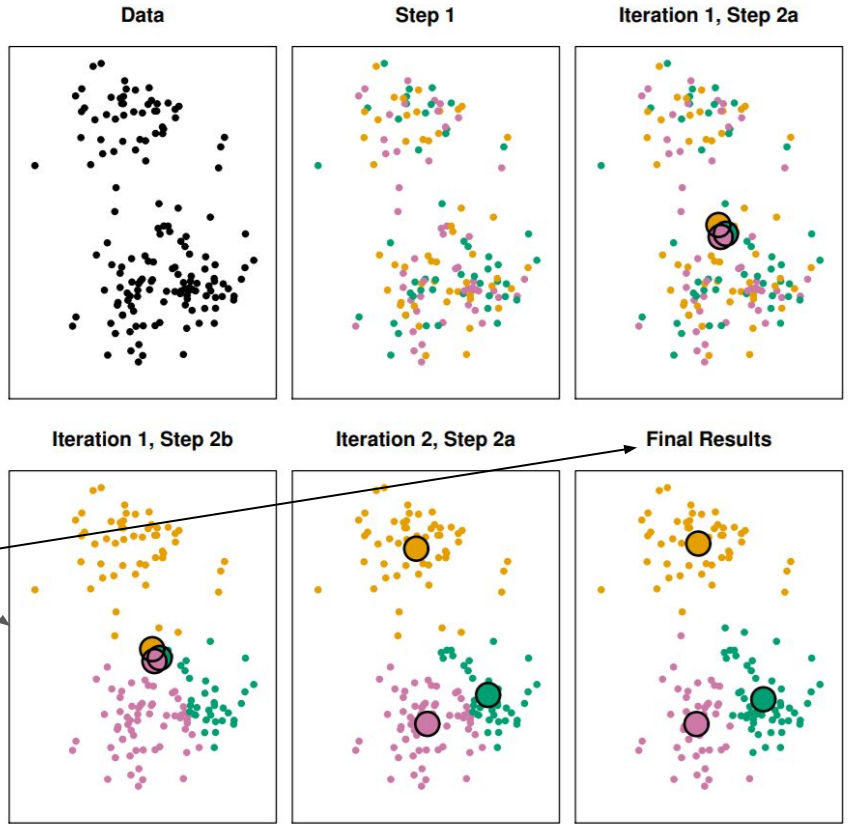
Asignamos cada caso al cluster con centroide más cercano

factor~data
EIDAES_UNSAM

K-Medias - Algoritmo

Inicializamos los centroides de forma aleatoria

Definimos el número de clusters que buscamos (k)

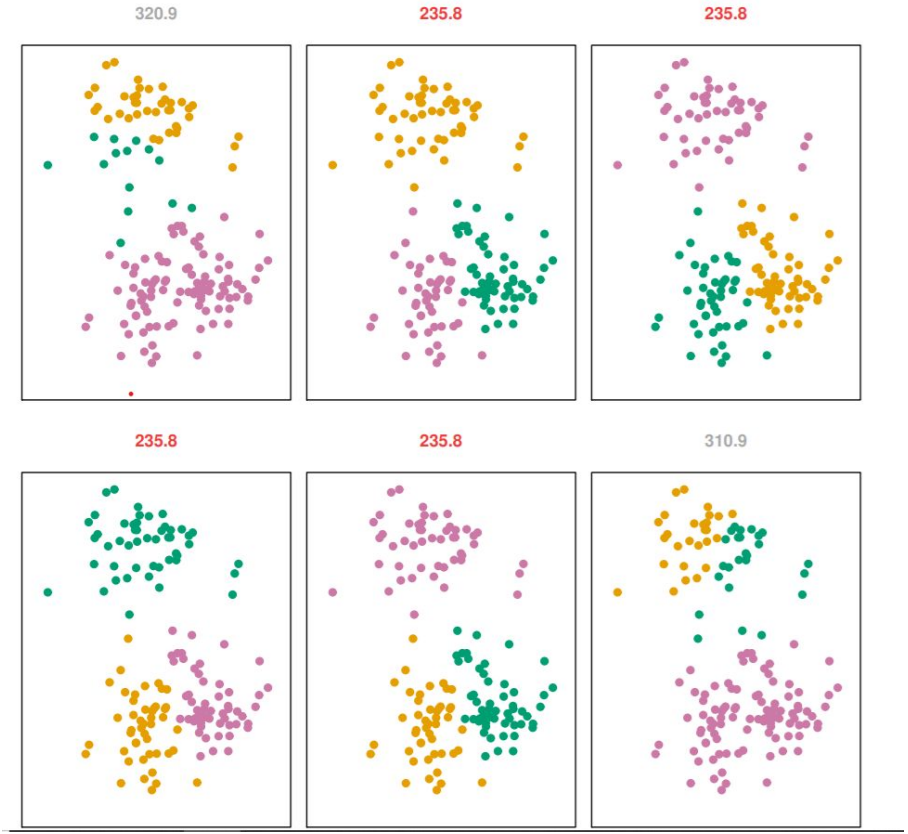


Calculamos los **centroides** (centros) de cada cluster como el promedio de las features de sus samples

Asignamos cada caso al cluster con centroide más cercano

factor~data
EIDAES_UNSAM

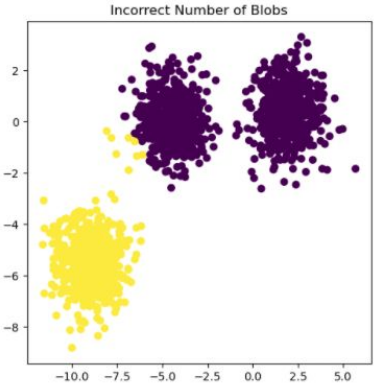
K-Medias - Diferentes inicializaciones

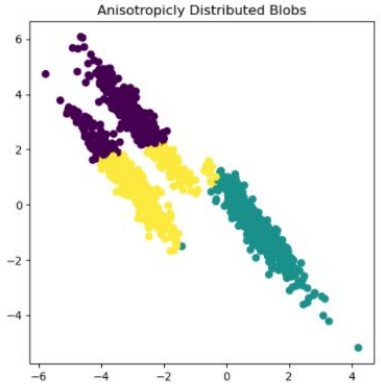


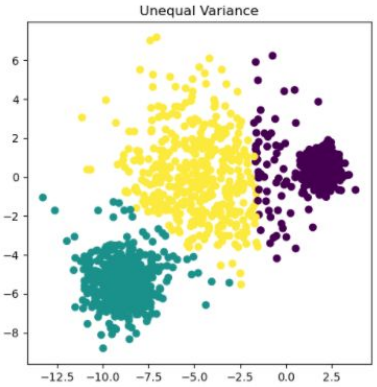
- Inicialización aleatoria: modelo no determinista
- Los resultados dependen de esa inicialización

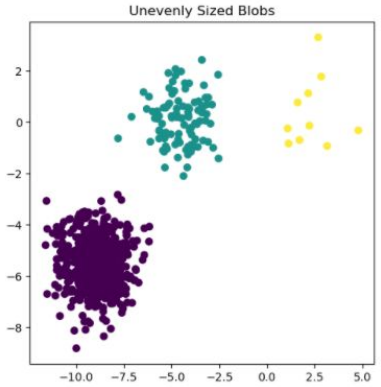
K-Medias - Ventajas y límites

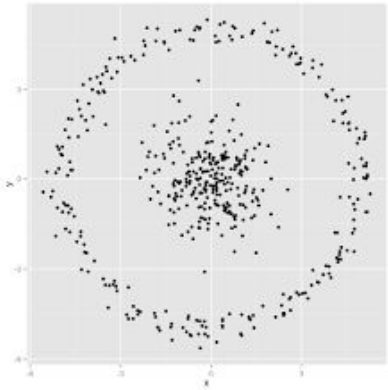
+ Simple y Fácil de implementar
+ Orden del algoritmo es lineal
- Depende de la inicialización
- Tiende a caer en un mínimo local
- Sensible a outliers
- Los clusters tienen que tener forma esférica
- No se puede aplicar a data categórica







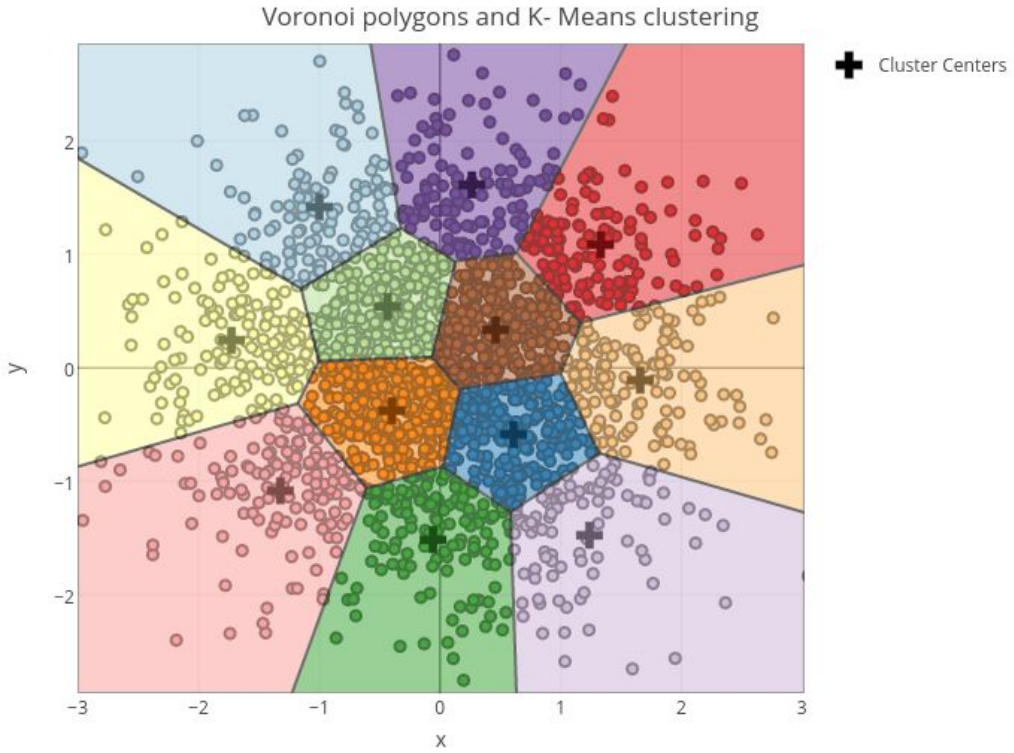




factor~data
EIDAES_UNSAM

K-Medias - Ventajas y límites

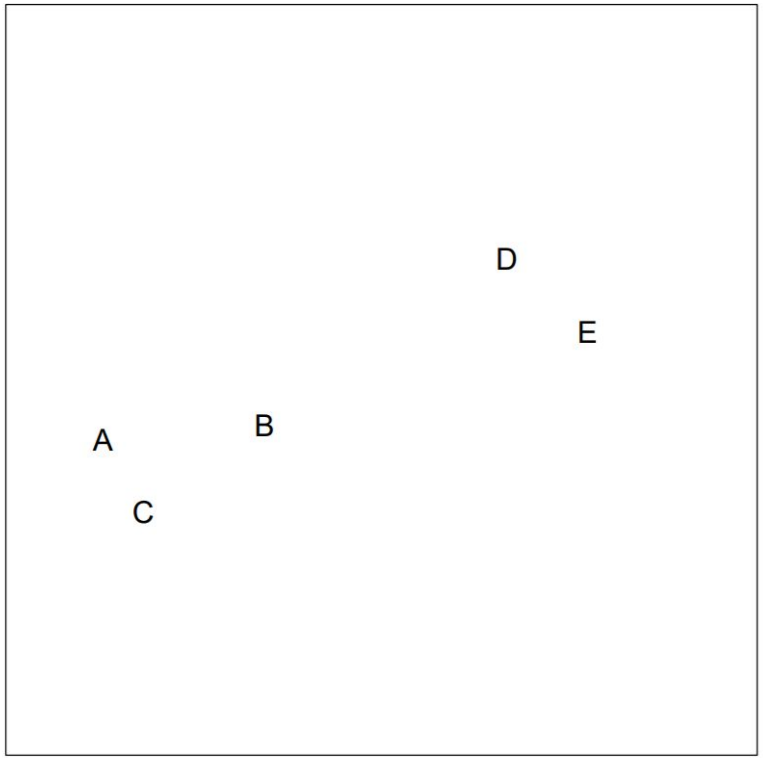
- Da un clustering de los datos aún si los datos no están “clusterizados”



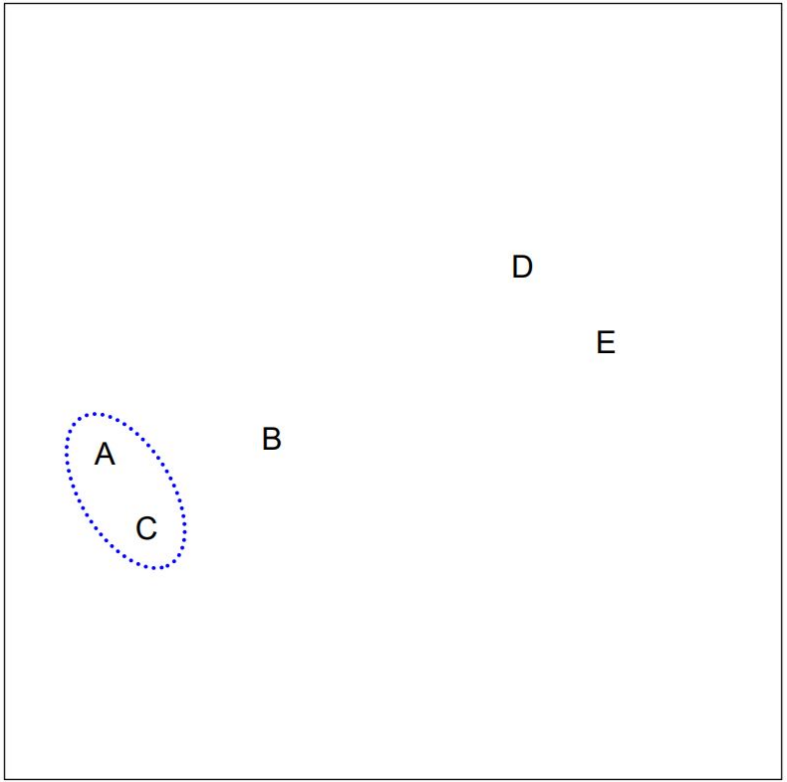
Clustering jerárquico

- K-medias requiere especificar de antemano la cantidad de clusters buscados
- El clustering jerárquico ofrece un método para tratar de estimarlo
- Veremos el método más común de clustering jerárquico: bottom-up o “aglomerativo”

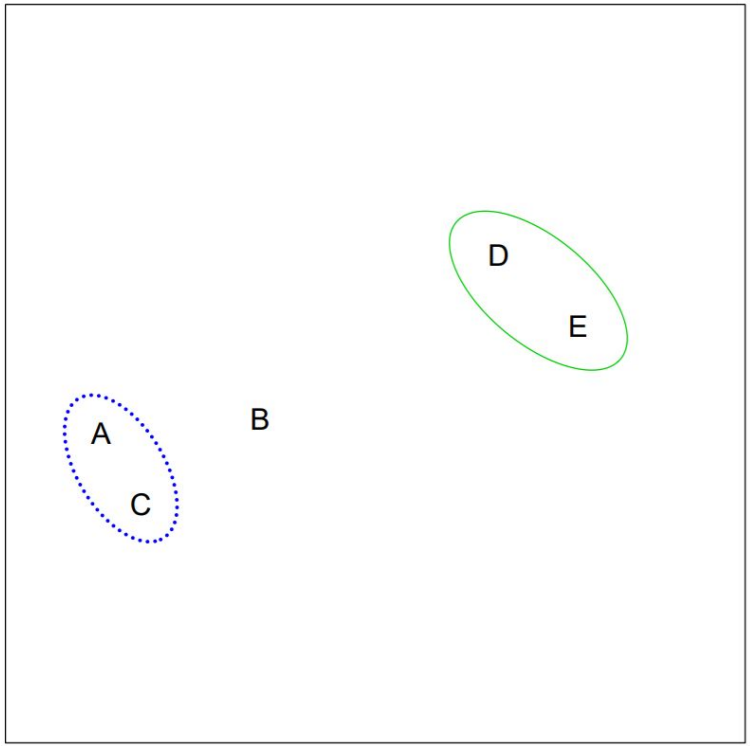
Clustering jerárquico



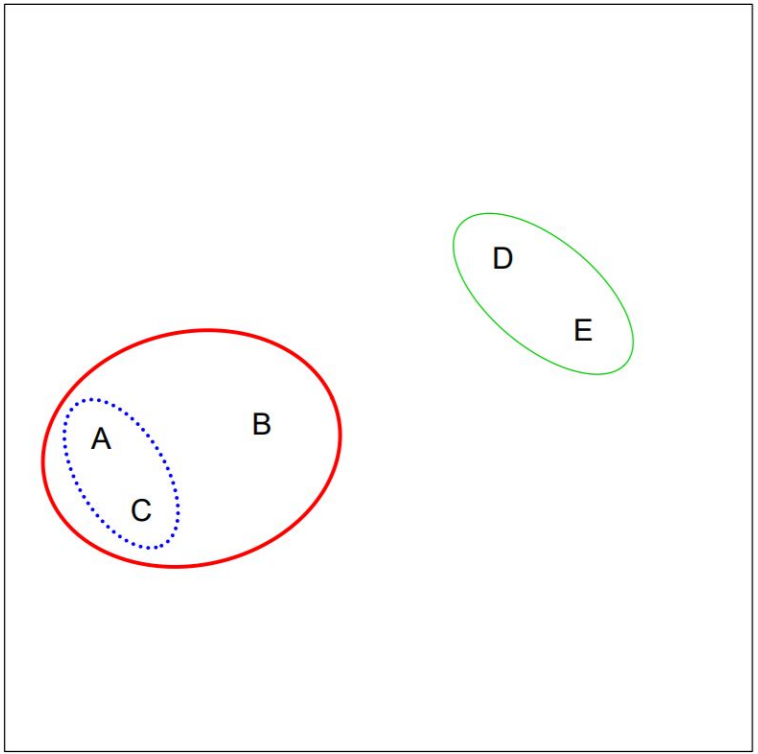
Clustering jerárquico



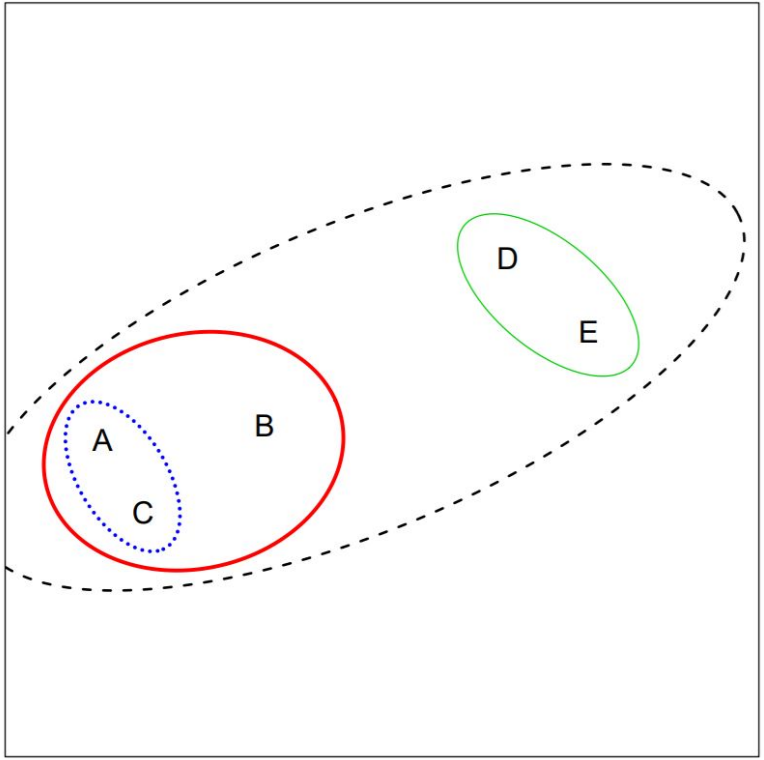
Clustering jerárquico



Clustering jerárquico



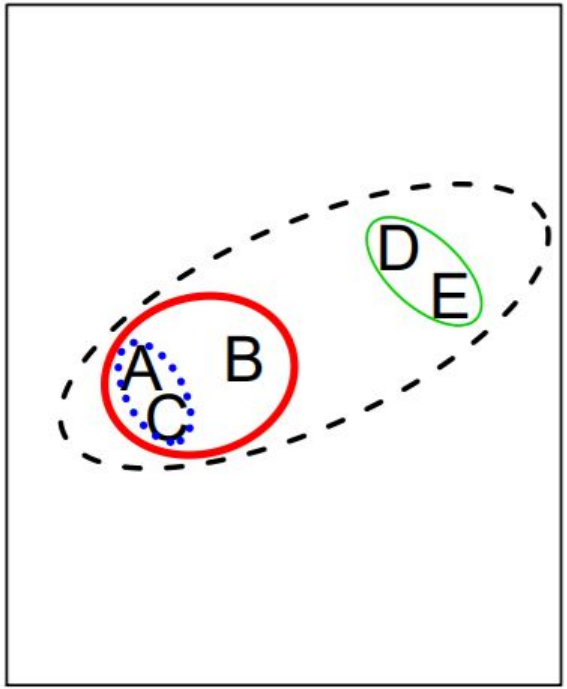
Clustering jerárquico

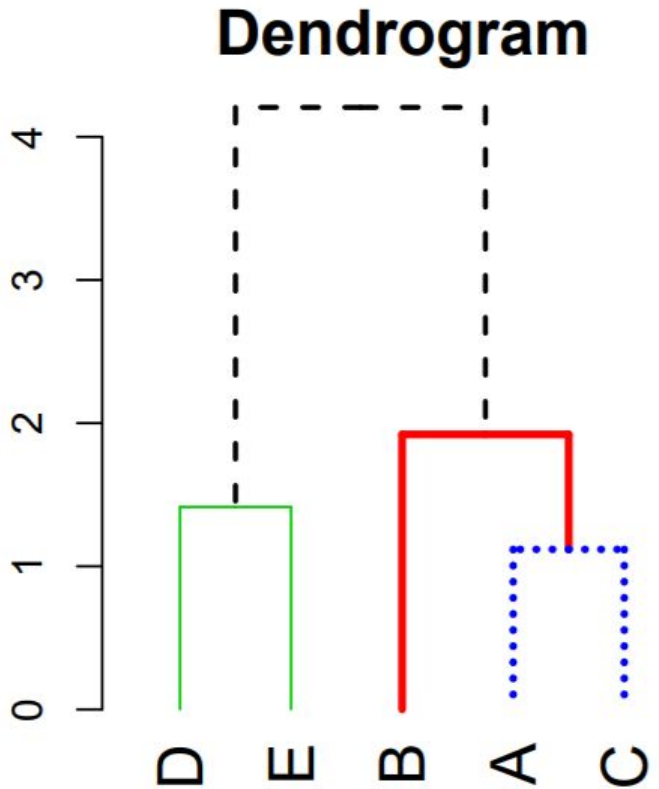


Clustering jerárquico

- Empieza con cada caso como un cluster único
- Identifica los clusters más cercanos y los une
- Repite el procedimiento
- Finaliza cuando todos los puntos han sido asignados a un único cluster

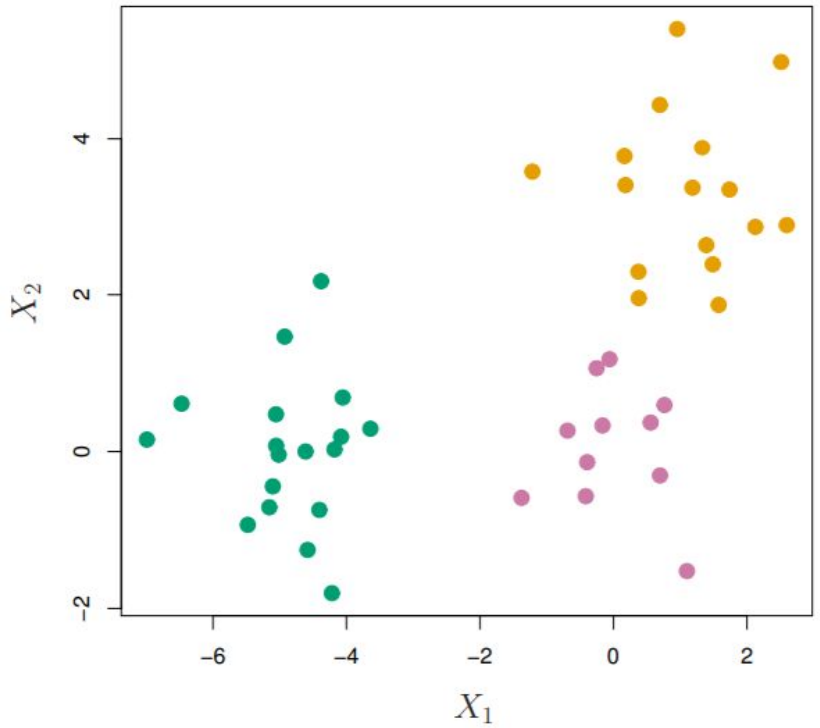
Clustering jerárquico



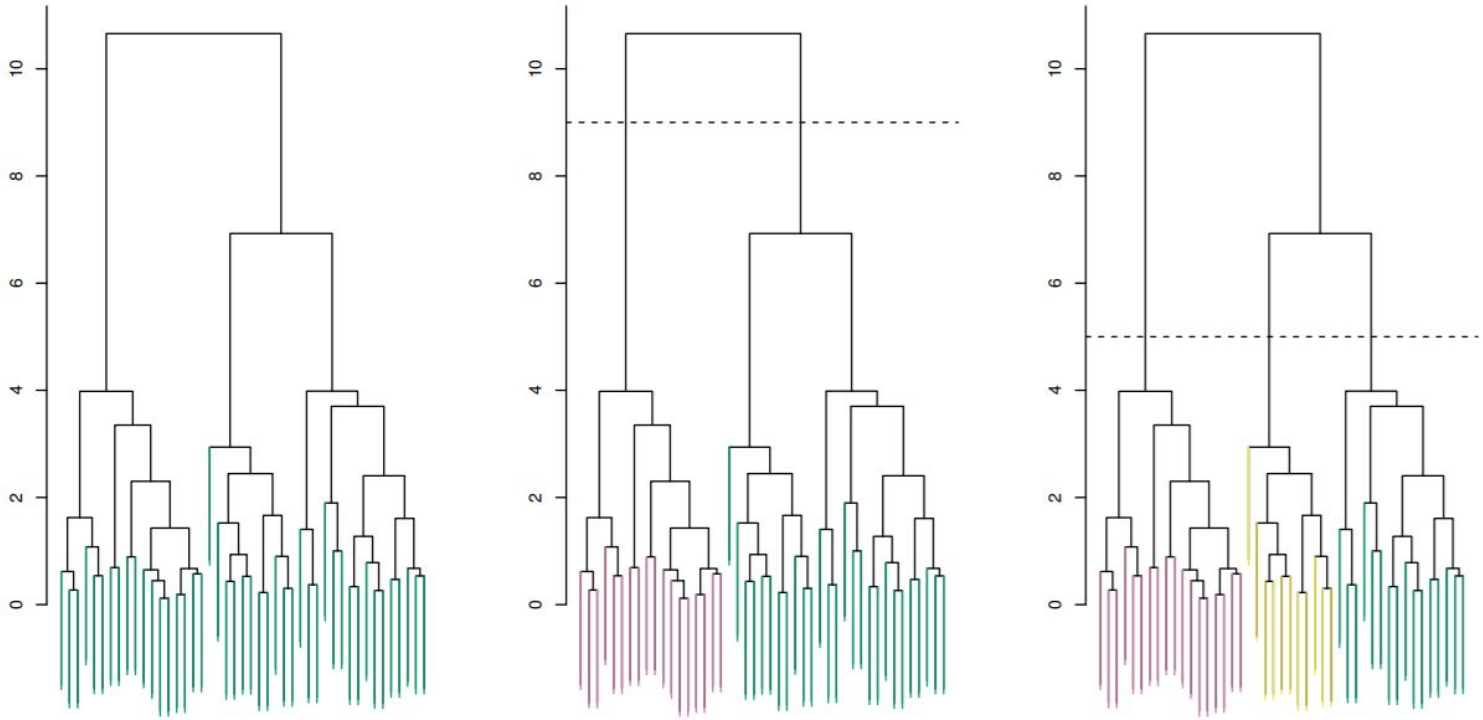


Clustering jerárquico

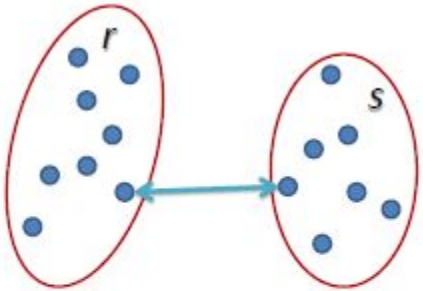
- Otro ejemplo



Clustering jerárquico

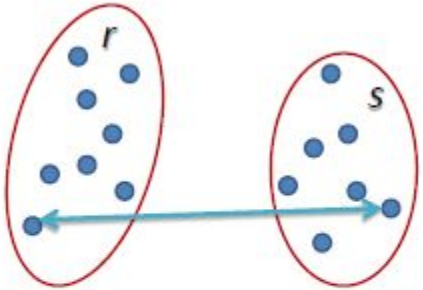


¿Cómo definir el método de aglomeración?



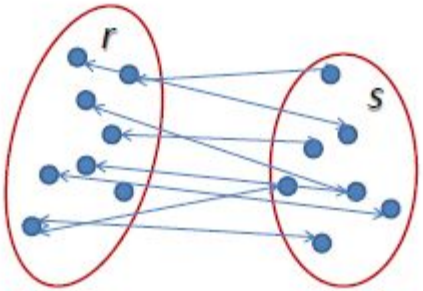
$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$

Single Linkage



$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

Complete Linkage

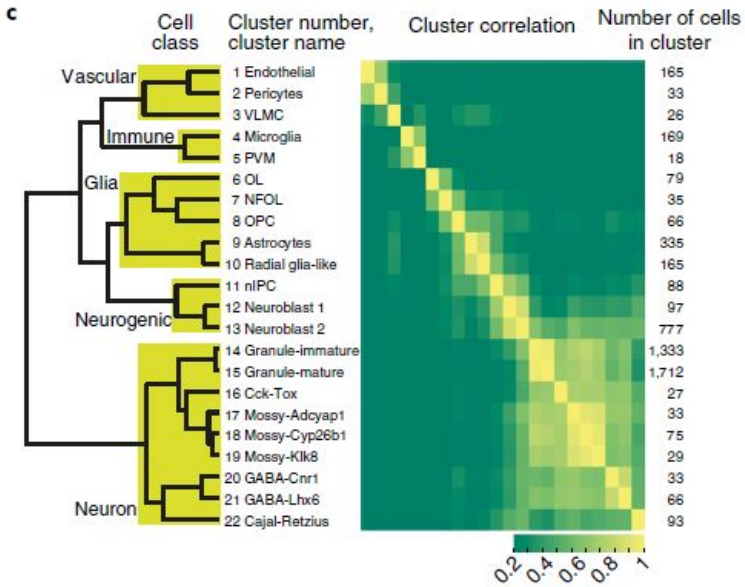


$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

Average Linkage

Clustering jerárquico. Ventajas y limitaciones

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



Problemas prácticos

- ¿Es necesario normalizar las escalas de las variables?
- ¿Qué métrica de similaridad usar?
- ¿Qué método de aglomeración?
- ¿Cuál es el número óptimo de clusters a utilizar?

Vamos al Notebook