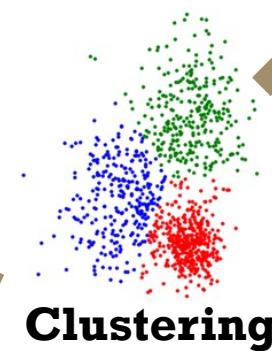


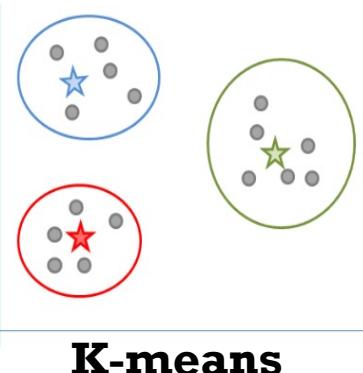
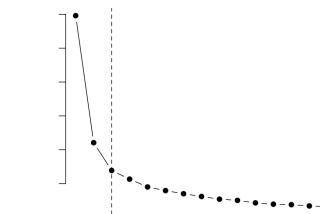
# **APRENDIZAJE NO SUPERVISADO**



# AGENDA



Evaluación del clustering



# APRENDIZAJE AUTOMÁTICO

## Aprendizaje supervisado

- Aprender a partir de un “experto”
- Datos de entrenamiento **etiquetados** con una clase o valor:

$(\underline{x_1}, \underline{x_2}, \dots, \underline{x_n}, y)$

Predictores, explicativos, independientes      Dependiente, objetivo, salida



- **Meta:** predecir una clase o valor

## Aprendizaje no supervisado

- Sin conocimiento de una clase o valor objetivo
- Datos **no** están **etiquetados** ( $x_1, x_2, \dots, x_n$ )
- **Meta:** descubrir factores no observados, estructura, o una representación mas simple de los datos



# APRENDIZAJE AUTOMÁTICO

## Aprendizaje supervisado

Edad	Ingresos	Tiene carro?
24	1'200.000	NO
23	4'500.000	SI
45	1'250.000	SI
32	1'100.000	NO

Factores/atributos/variables independientes, predictores, explicativos      Dependiente, objetivo, respuesta, salida

34	3'500.000
----	-----------

?

¿Cuál es el valor predicho para una instancia dada?

## Aprendizaje no supervisado

Edad	Ingresos
24	1'200.000
23	4'500.000
45	1'250.000
32	1'100.000

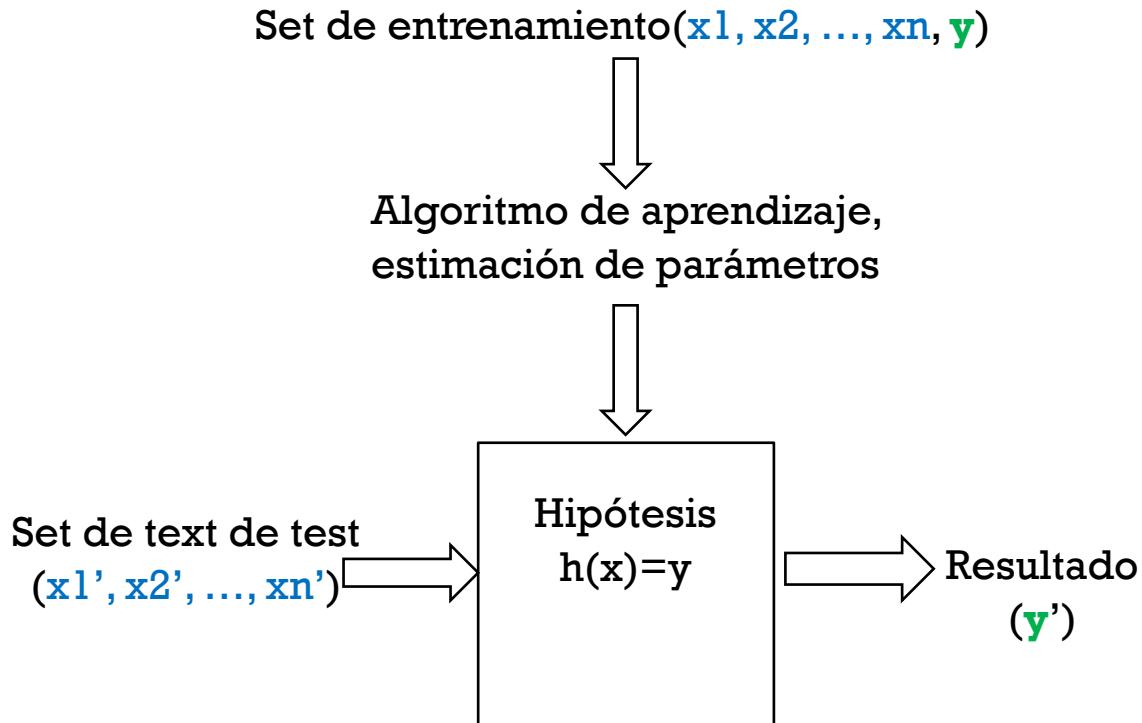
Factores/atributos/variables  
Datos no etiquetados:  
“¿Qué me puede decir de mis datos?”

¿Se puede encontrar alguna estructura en los datos?

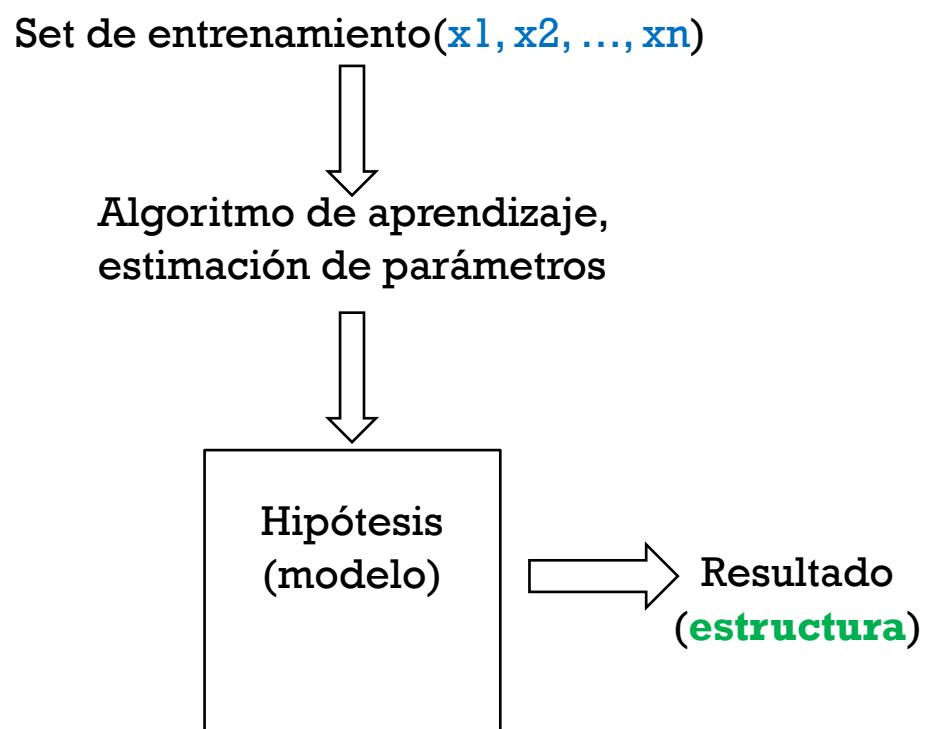


# APRENDIZAJE AUTOMÁTICO

## Aprendizaje supervisado



## Aprendizaje no supervisado



# APRENDIZAJE NO SUPERVISADO

- No se interesa por la predicción sino por encontrar una estructura, un nuevo punto de vista, una simplificación o un resumen de los datos
- Usualmente se incluye en la fase exploratoria de datos
- Tipos de tareas:
  - Segmentación (clustering)
  - Cambio de representación (e.g. reducción de dimensiones, selección de factores)
  - Reglas de asociación
  - Detección de anomalías (i.e. excepciones)
- Difícil de validar los resultados, ya que no se cuenta con un “gold standard”

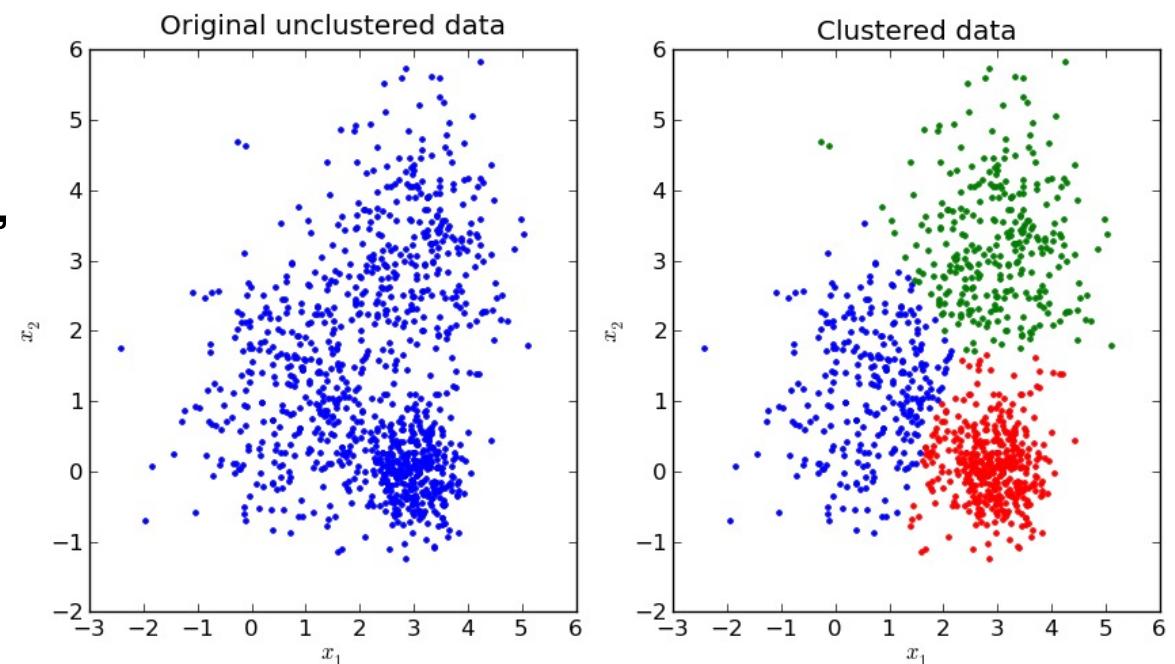


# CLUSTERING



# CLUSTERING

- No se tiene una variable objetivo
- Se busca agrupar los datos similares para encontrar patrones globales de los datos
- Agrupamiento por **similitud, proximidad, densidad**
- Particionar un conjunto heterogéneo en grupos, de forma que elementos en un grupo sean similares entre sí y tan diferentes como sea posible de elementos en otros grupos.

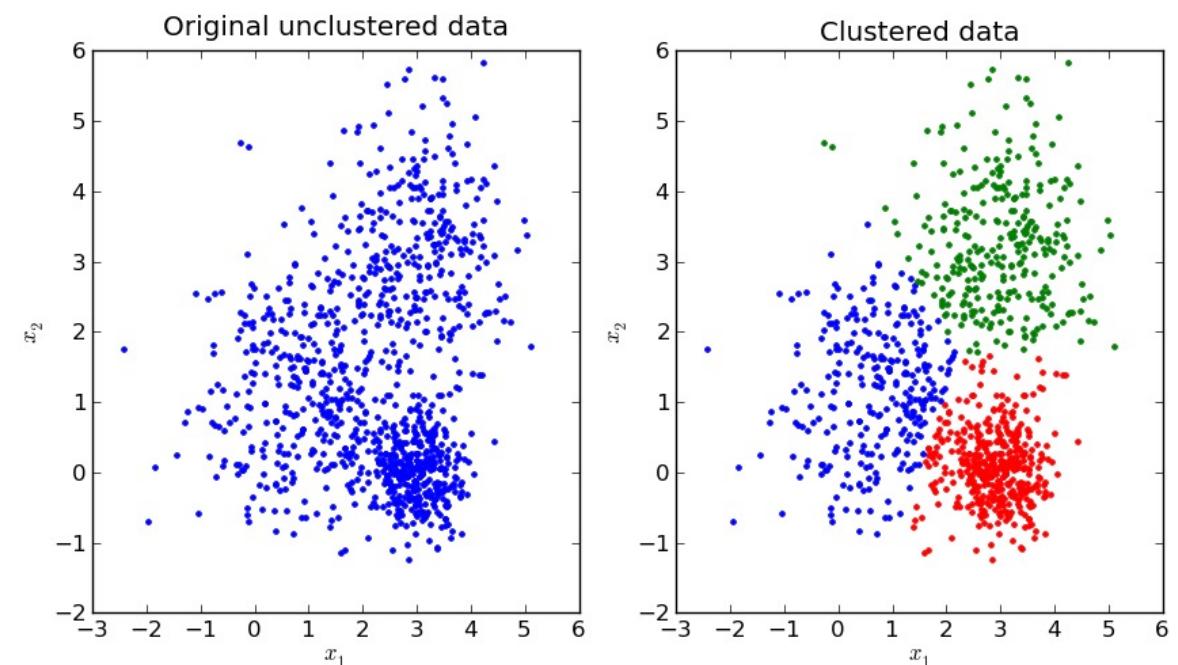


<http://pypr.sourceforge.net/kmeans.html>



# CLUSTERING POR DISTANCIA

- Objetivo: descubrir **k grupos o segmentos** desconocidos que
  - Minimicen la distancia dentro de los grupos
  - Maximicen la distancia por fuera de los grupos
- Se basan en una noción de **distancia**
  - Definición de la medida a utilizar
  - Unidades de los atributos tienen gran influencia
- **Normalizar**
- **Estandarizar**



<http://pypr.sourceforge.net/kmeans.html>



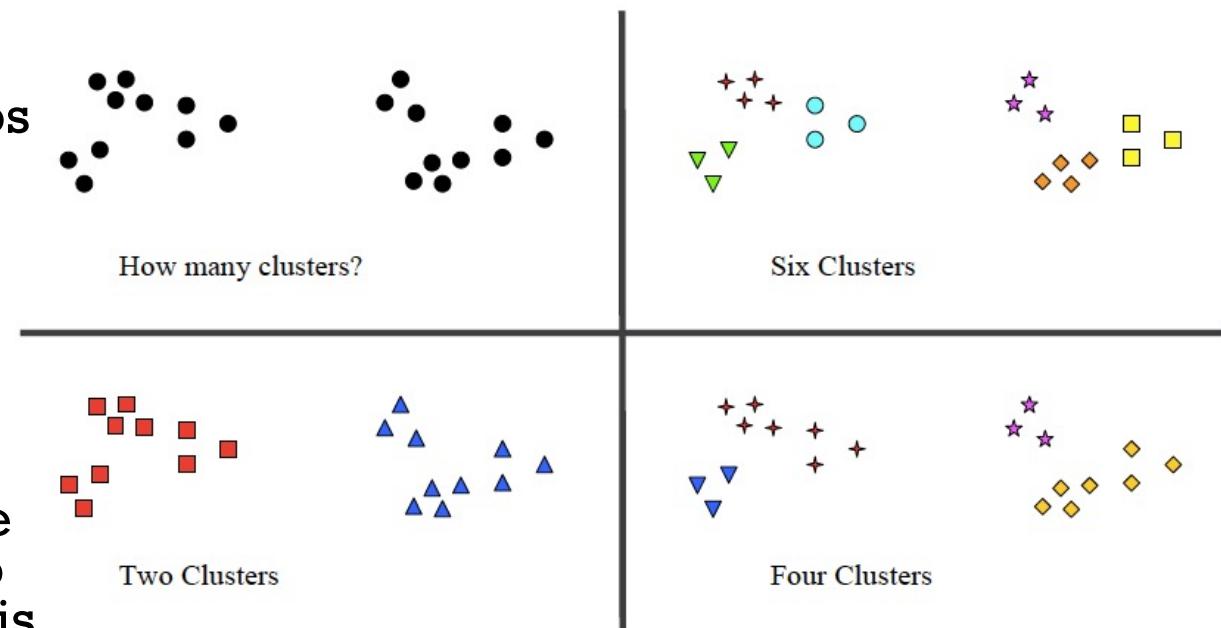


How many clusters?



# CLUSTERING POR DISTANCIA

- Se pueden buscar segmentos de observaciones o de atributos (usando los mismos algoritmos)
- No existe un método universal absoluto para establecer **k**, solo heurísticos
- Requiere juicio humano, más difícil de automatizar
- La interpretación de los resultados no se debe de hacer de manera absoluta, sino como un punto de partida para el análisis
- Los datos puede que no contengan estructura, por lo que su segmentación no va a tener tanto sentido



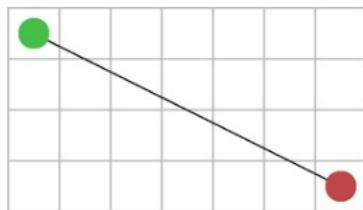
<http://governingstochastic.weebly.com/blog/category/clustering>



# CLUSTERING – DISTANCIAS

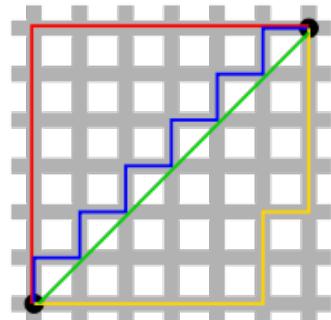
- Medidas de **similitud o distancia**:

- **Euclíadiana**: tamaño del segmento linear que une las dos instancias comparadas.



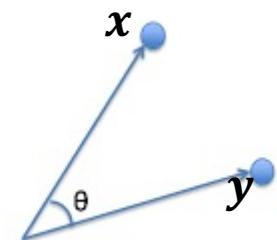
$$\begin{aligned} d(\mathbf{p}, \mathbf{q}) &= d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \\ &= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}. \end{aligned}$$

- **Manhattan**: basada en una organización en bloques rectilíneos

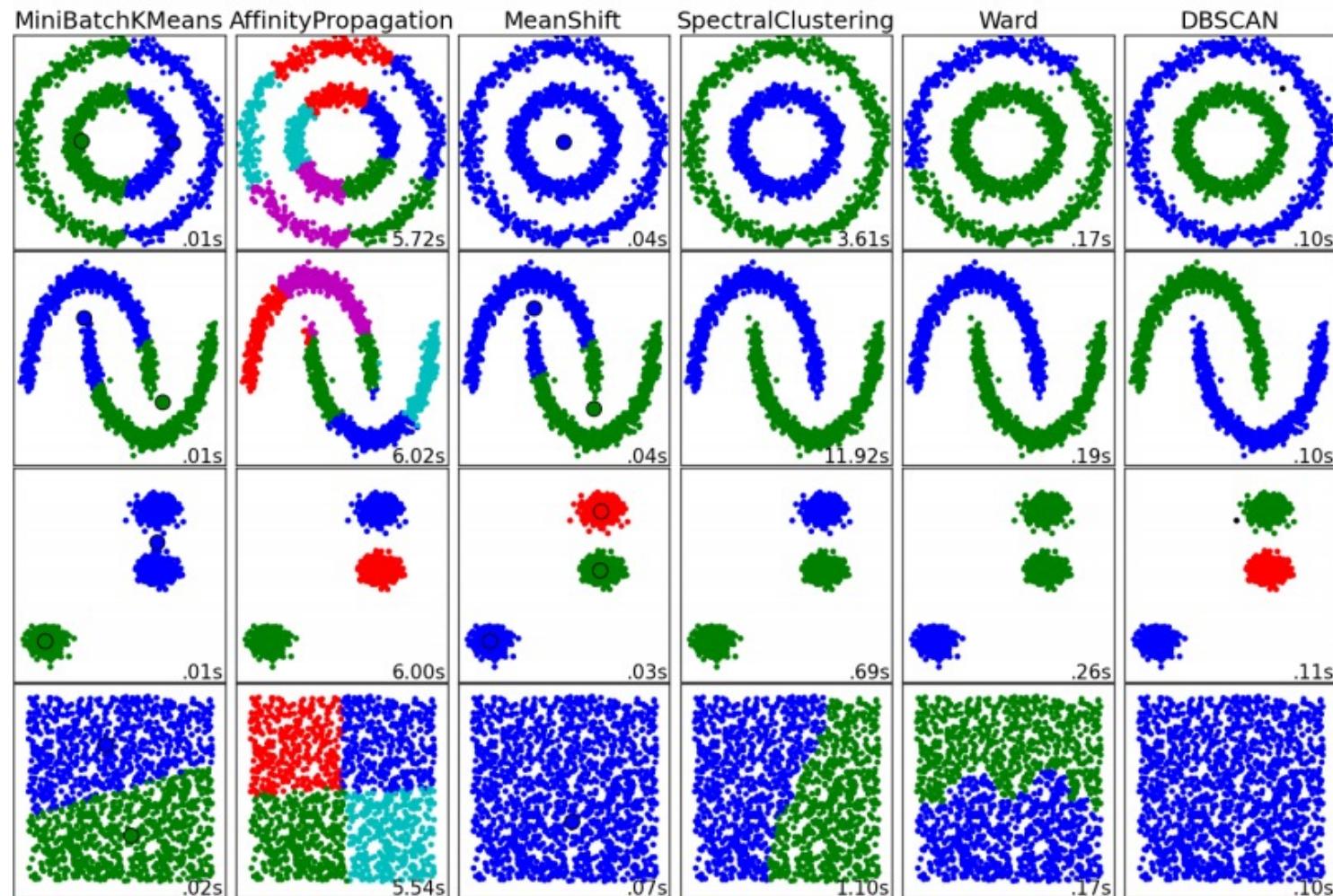


- **Coseno**: coseno del ángulo entre las dos instancias comparadas → Alta dimensionalidad y **big data**

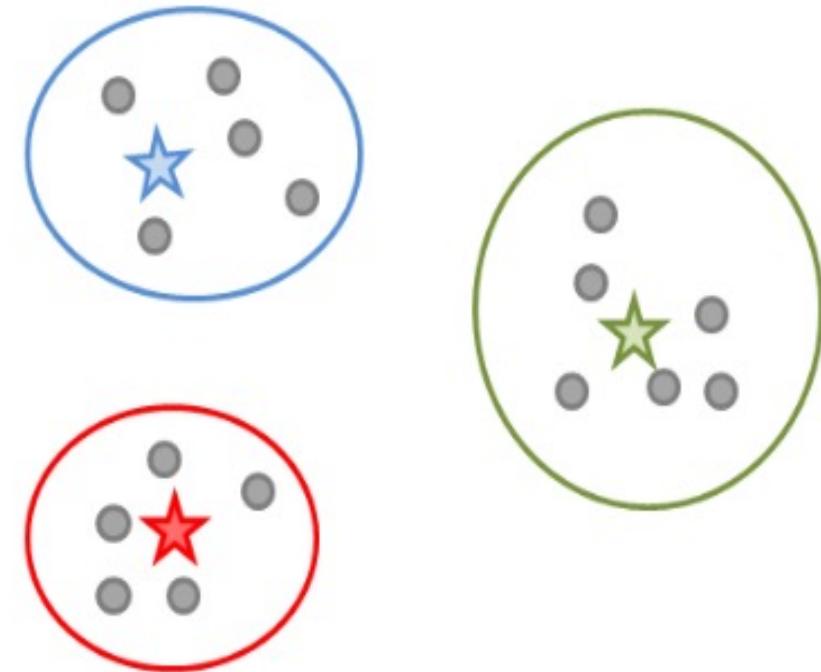
$$sim(x, y) = \cos(\theta_{x,y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\sum_i x_i * y_i}{\sqrt{(\sum_i x_i * x_i) * (\sum_i y_i * y_i)}}$$



# CLUSTERING



# K-MEANS



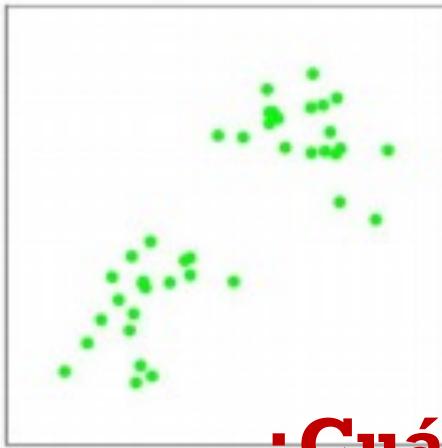
# K-MEANS



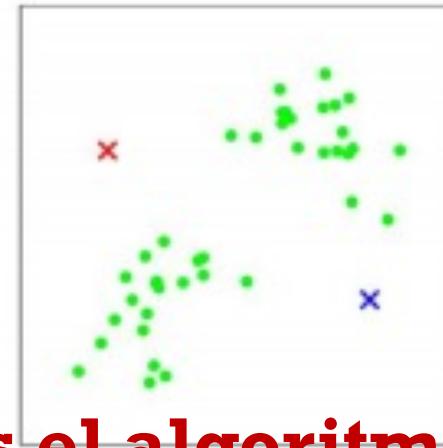
(a)



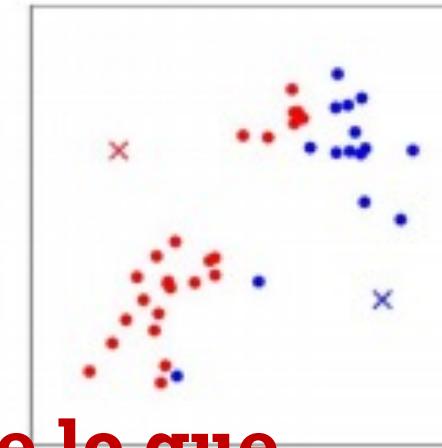
# K-MEANS



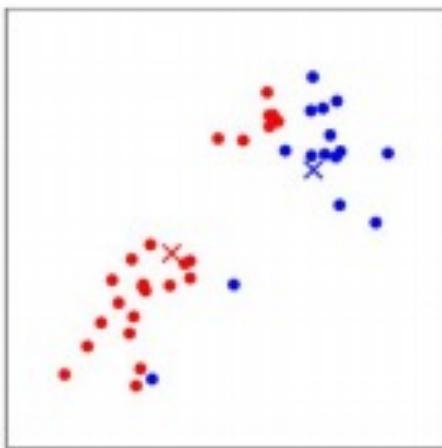
(a)



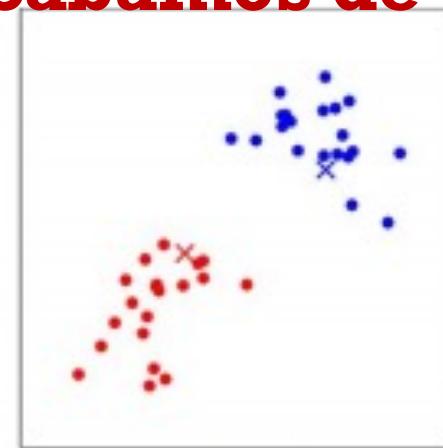
(b)



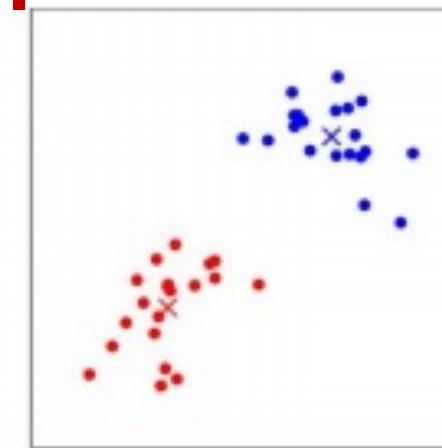
(c)



(d)



(e)



(f)

**¿Cuál es el algoritmo de lo que acabamos de ver?**



# K-MEANS

- Algoritmo:
  1. Inicializar los K centroides
  2. Asignar cada instancia al cluster del centroide más cercano
  3. Recalcular los centroides de cada cluster (el baricentro/promedio)
  4. Repetir pasos 2 y 3 hasta convergencia (hasta que los centroides permanezcan estáticos)
- Cada observación se asigna a un solo cluster, de manera absoluta
- Los clusters no se sobrelapan
- Objetivo: minimizar la variación dentro de los clusters (Within Sum of Squares - WSS):

$$WSS = \sum_{i=1}^{\#instancias} distancia(x_i - centroide(x_i))^2$$



# K-MEANS

- Consideraciones:

- ¿Cómo estimar el número de clusters (K)?
  - Mardia (1979):  $\sqrt{n}/2$
  - Método “del codo”
  - Método Silhouette
  - Medida de CH
- ¿Cómo inicializar los centroides de los clusters?
  - Escoger centros completamente aleatorios
  - Escoger puntos existentes aleatoriamente
  - Escoger los centroides utilizando K-Means ++



# K-MEANS

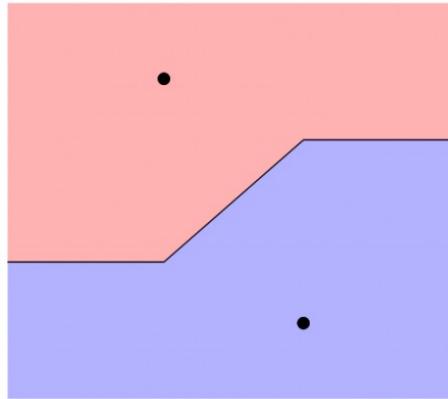
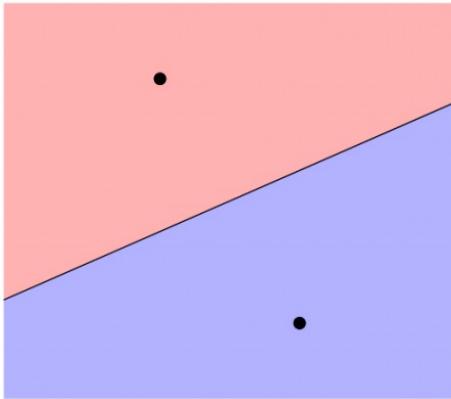
## K-Means++

- La idea es inicializar los centros lo más lejanos los unos de los otros
- Algoritmo K-Means++:
  1. Se comienza con un conjunto M de centroides vacío
  2. Se escoge aleatoriamente una instancia que no sea ya un centroide y se agrega a M.
  3. Se calcula la distancia mínima de las instancias que quedan con los centroides en M
  4. Se escoge una nueva instancia como centroide de manera aleatoria asociando una probabilidad a cada instancia dada por su distancia mínima calculada anteriormente
  5. Repetir pasos 3 y 4 hasta haber seleccionado K centroides
  6. Continuar con el algoritmo K-Means clásico.



# K-MEANS

- Consideraciones:
  - ¿Qué distancia escoger?
    - Depende del problema
    - e.g. Euclidiana, Manhattan, correlación



**Diagrama de Voronoi:** particionamiento espacial basado en la distancia con los centroides



Euclidean



Manhattan

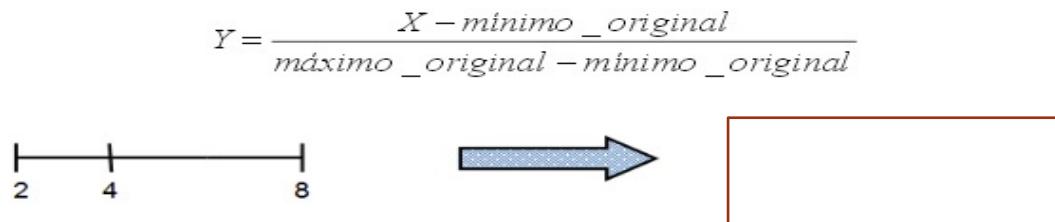
<http://math.stackexchange.com/>



# K-MEANS

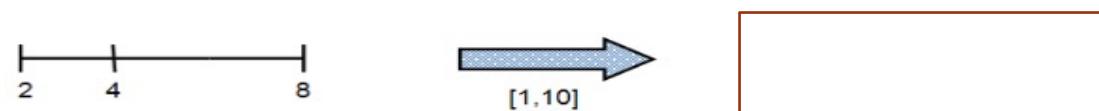
## Consideraciones:

- Normalización Min-Max [0, 1]



- Normalización [newmin, newmax] → Generalización, cambio de escala a otro intervalo cualquiera, no necesariamente [0,1], ni [oldmin, oldmax]

$$Y = \text{min} + \frac{X - \text{minimo\_original}}{\text{máximo\_original} - \text{minimo\_original}} (\text{max} - \text{min})$$



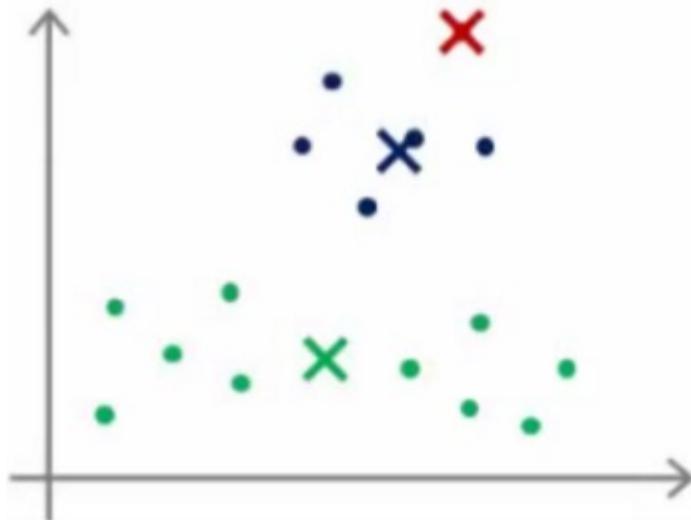
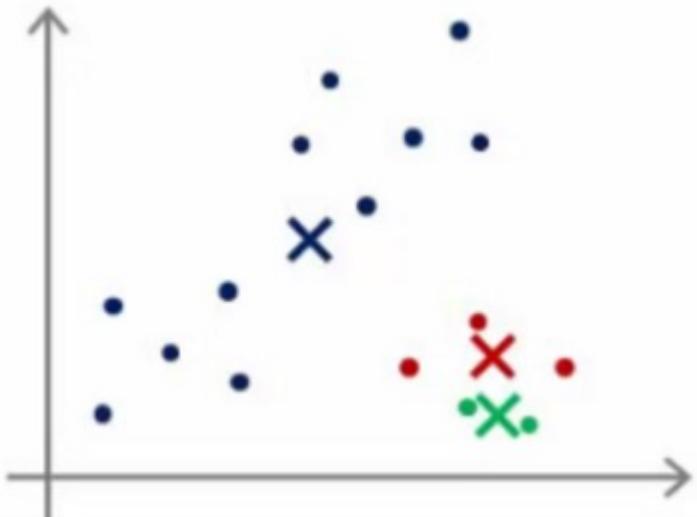
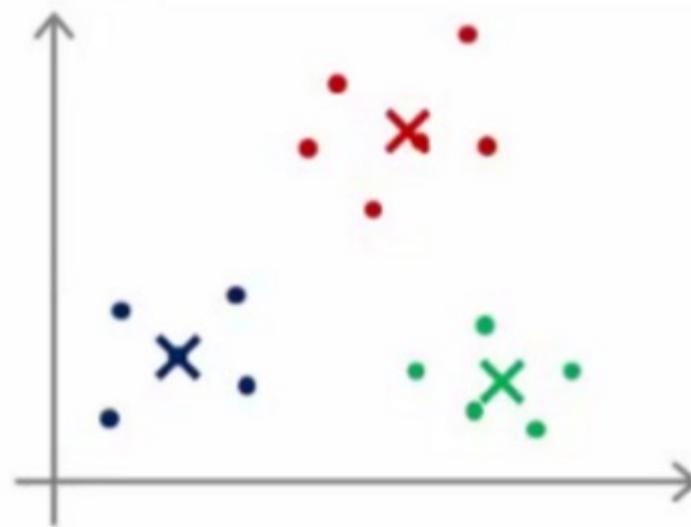
- Normalización z-score (estandarización)
  - Supuesto de distribución normal
  - Sea Z la representación estandarizada del dato
  - X la representación actual del dato
  - $\mu$  el valor promedio de los datos
  - $\sigma$  la desviación estándar del campo

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



# K-MEANS

- Consideraciones:
  - ¿Cómo evitar los óptimos locales?
    - Ejecutar varias veces el algoritmo con diferentes inicializaciones, seleccionar el clustering con el mínimo WSS



# K-MEANS

- Consideraciones:

- Algoritmo de particionamiento
- Muy fácil de implementar
- Más rápido que clustering jerárquico
- Sólo trabaja con atributos numéricos (noción de promedio)
- Muy sensible a excepciones
- Funciona muy bien con datos generados siguiendo un proceso Gaussiano, cuyas fronteras son lineales en el caso de la distancia euclídea (Voronoi)
- No funciona para identificar clusters con formas no convexas

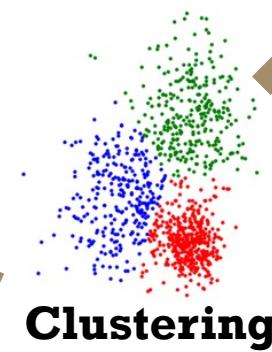


# TALLER: K-MEANS – CLIENTES SUPERMERCADOS

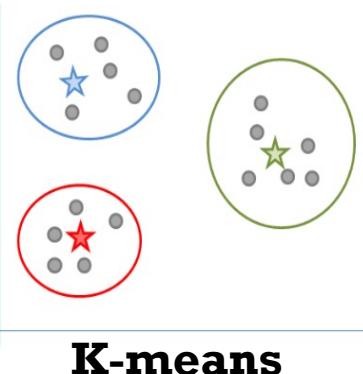
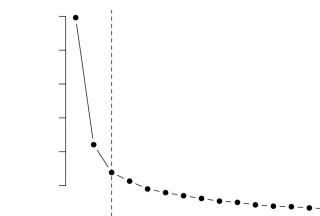
Desarrollar el taller de clustering de clientes de supermercados  
09-SUPERMERCADOS-K-Means-STUD.html (hasta antes de la determinación del k).



# AGENDA



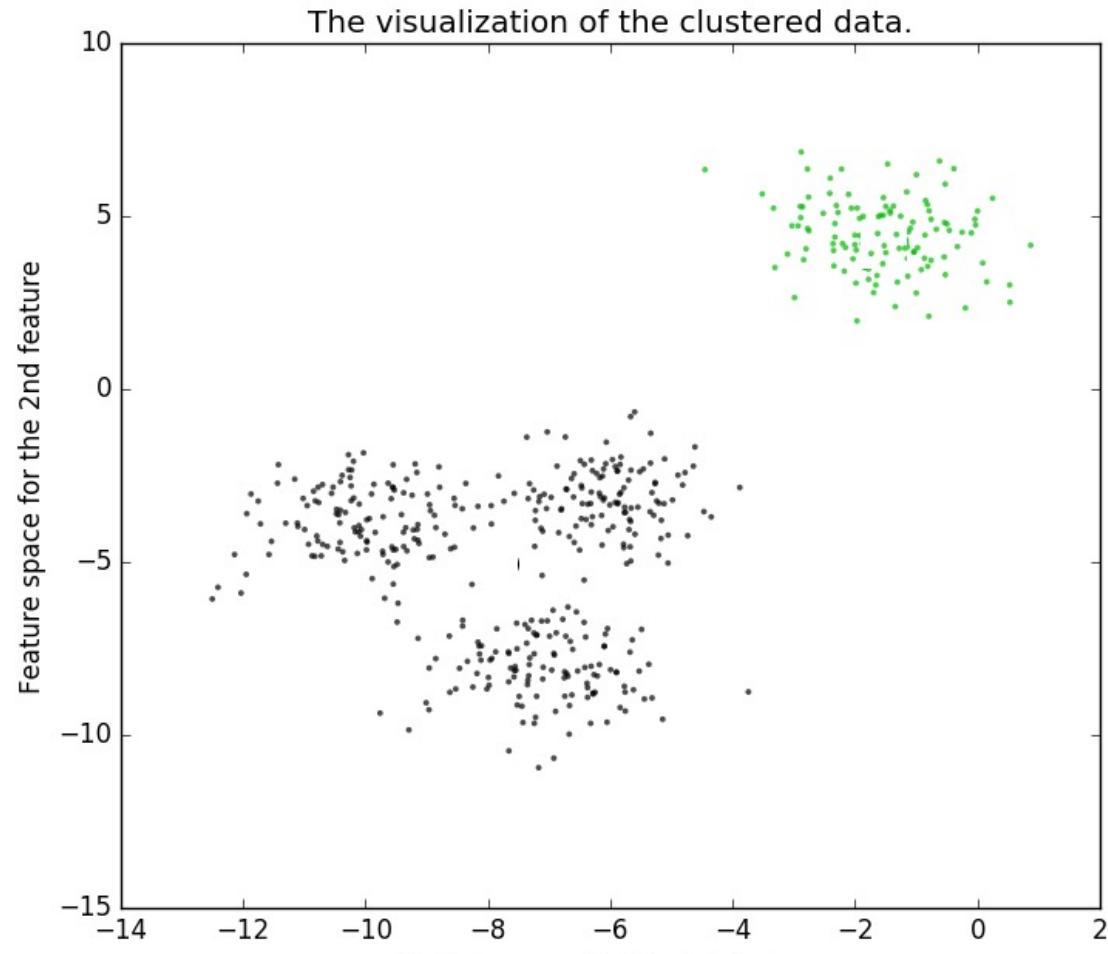
Evaluación del clustering



# EVALUACIÓN DE CLUSTERING



# ESCOGENCIA DEL K



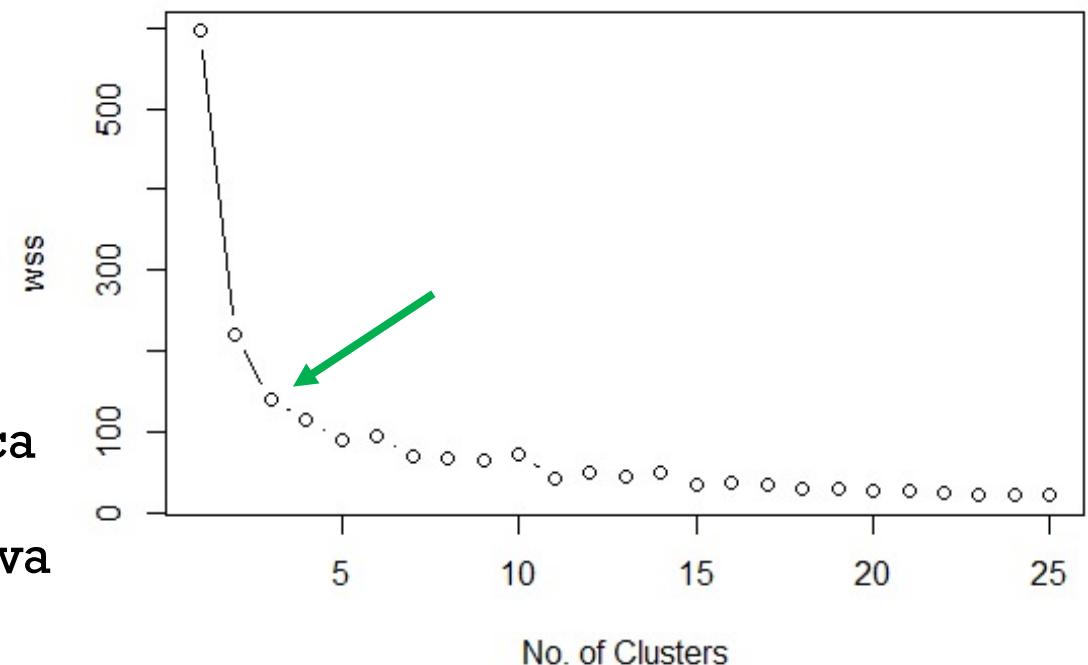
¿Cuántos clusters ven uds aquí?

[http://scikit-learn.org/stable/auto\\_examples/cluster/plot\\_kmeans\\_silhouette\\_analysis.html](http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html)



# ESCOGENCIA DEL K – CODO

- Heurísticos:
  - No hay un método absoluto
  - Dependen del juicio del analista, se requiere conocimiento del negocio
- Método “del codo”:
  - Plotear WSS para cada valor de K
  - Escoger el último valor de K que implica una reducción “considerable” del WSS del clustering resultante, cuando la curva se vuelve aproximadamente lineal

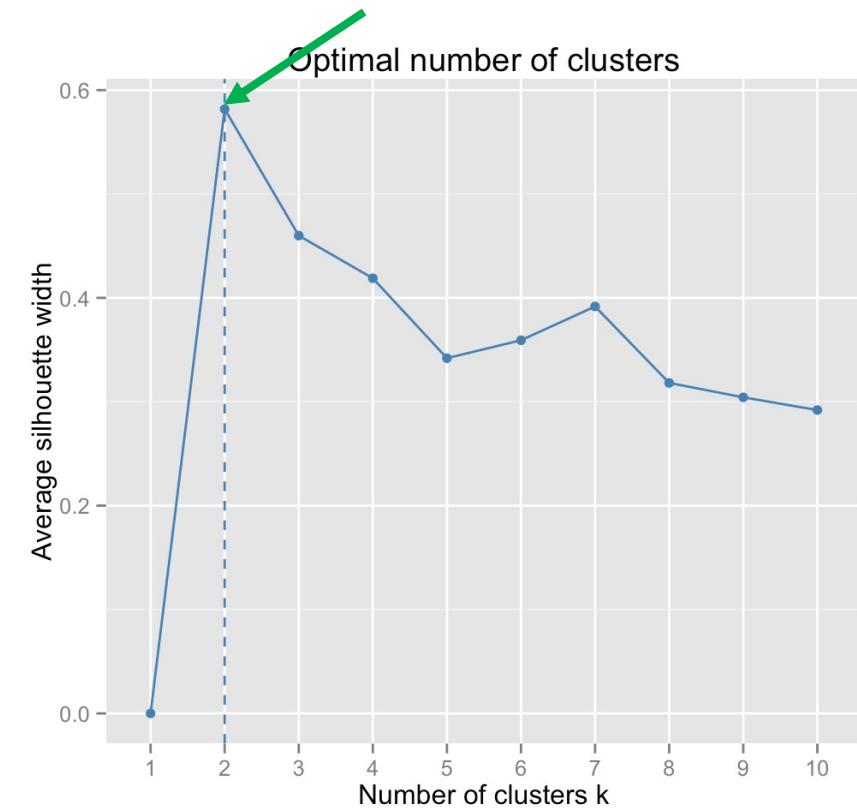


$$WSS = \sum_{i=1}^{\#instancias} \text{distancia}(\mathbf{x}_i - \text{centroide}(\mathbf{x}_i))^2$$



# ESCOGENCIA DEL K – SILHOUETTE

- Método Silhouette
  - Se busca el K que maximice la **separación** entre clusters, con clusters lo más **compactos** posibles
  - Analizar el ajuste de cada instancia al cluster al que fue asignado
  - Qué tan cerca está cada observación de las demás de su propio cluster
    - 0,7-1,0: el cluster es fuertemente robusto
    - 0,5-0,7: el cluster es razonablemente robusto
    - 0,25-0,5: el cluster puede ser artificial y puede no denotar una noción de estructura necesariamente
    - Inferior a 0,25: el cluster debería descartarse, no indica estructura
  - Se busca la maximización del valor Silhouette promedio de los clusters



# ESCOGENCIA DEL K – SILUETA

- Método Silueta (Silhouette)

- Calcular el valor de silueta de cada punto:

- Cohesión del punto con su cluster  $C_i$  (promedio de distancias con puntos de su mismo cluster):

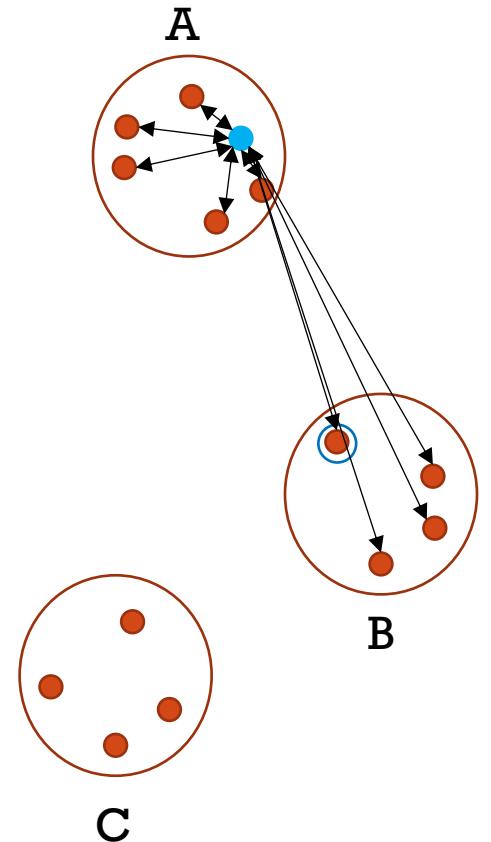
$$cohésion(p) = a(p) = \frac{\sum_{p' \in C_i, p' \neq p} distancia(p, p')}{|C_i| - 1}$$

- Separación de los puntos de otros clusters (distancia promedio con los puntos del cluster más cercano):

$$separación(p) = b(p) = \min_{C_j: 1 \leq j \leq k, j \neq i} \left( \frac{\sum_{p' \in C_j} distancia(p, p')}{|C_j|} \right)$$

- El valor de silueta del punto es entonces:

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



# ESCOGENCIA DEL K – SILUETA

- Método Silueta (Silhouette)

- Calcular el valor de silueta de cada cluster (promedio de las siluetas de sus puntos).

$$\text{silueta}(C_i) = \frac{1}{|C_i|} \sum_{p \in C_i} s(p)$$

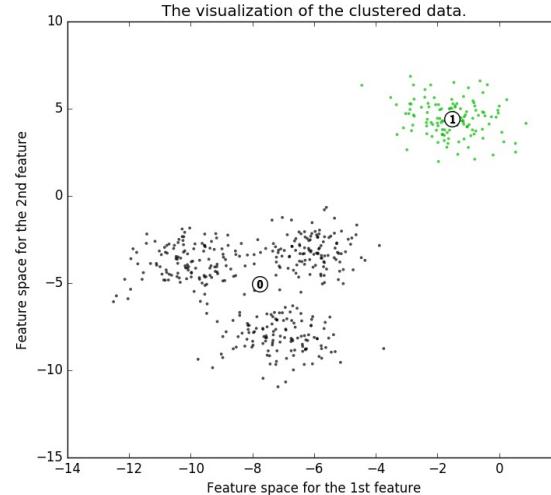
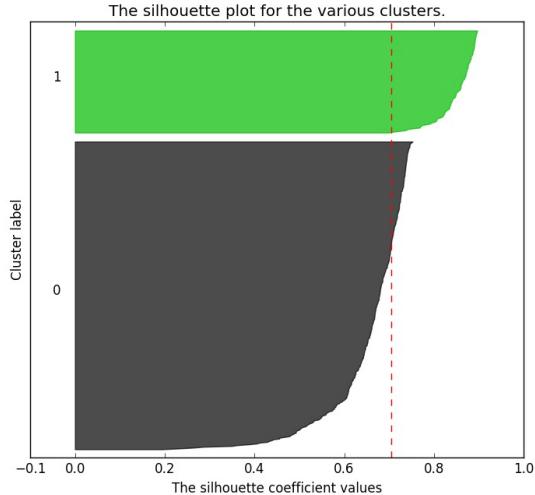
- Analizar los puntos y clusters, buscando posibles problemas de asignación dados por el valor del K:

- El rango de la silueta está entre -1 y 1
  - Una silueta de 0 implica que la asignación de un punto a su cluster es indiferente
  - Se espera que los puntos del mismo cluster estén más cercanos al punto en cuestión: para que la silueta sea positiva tenemos que  $a(p) < b(p)$

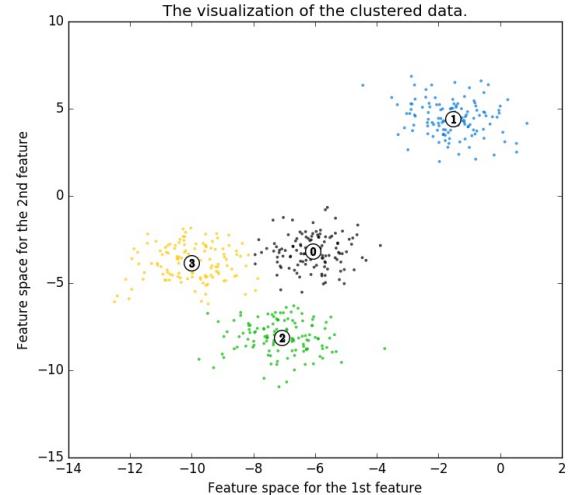
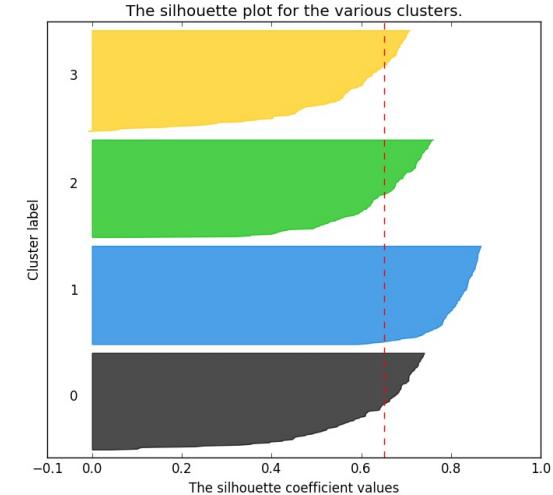


# ESCOGENCIA DEL K – SILHOUETTE

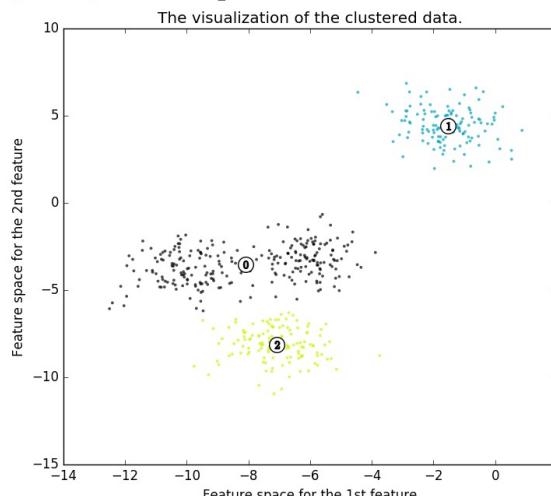
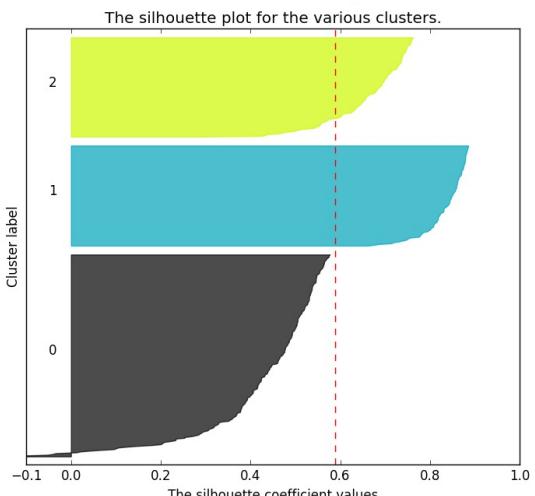
Silhouette analysis for KMeans clustering on sample data with n\_clusters = 2



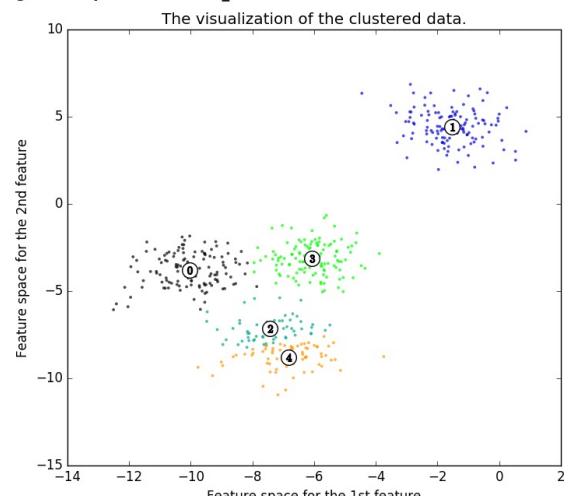
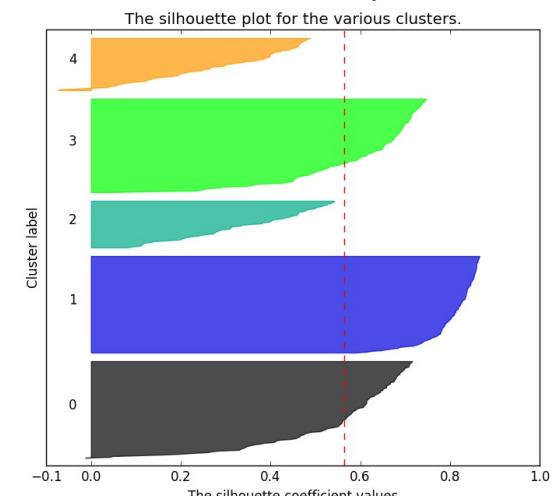
Silhouette analysis for KMeans clustering on sample data with n\_clusters = 4



Silhouette analysis for KMeans clustering on sample data with n\_clusters = 3



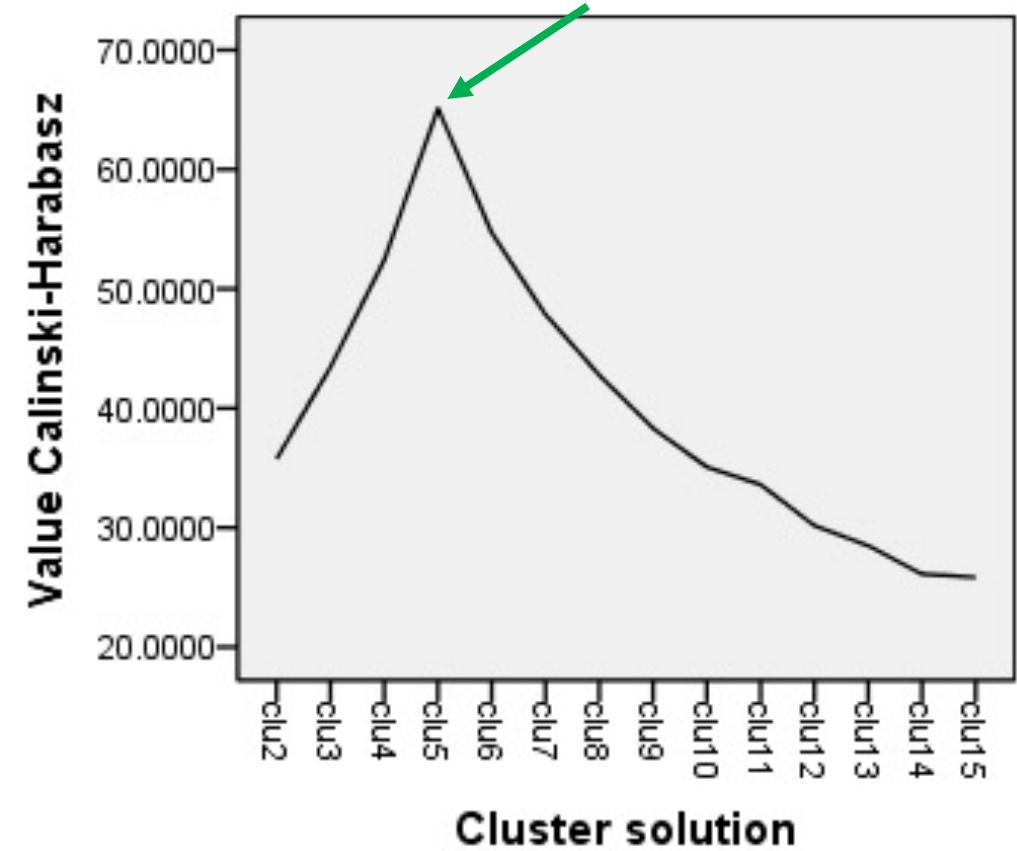
Silhouette analysis for KMeans clustering on sample data with n\_clusters = 5



# ESCOGENCIA DEL K – CALINSKI-HARABASZ

- Método de Calinski-Harabasz:
  - Se busca el K que maximice la **separación** entre clusters, con clusters lo más **compactos** posibles
  - TSS = variación total (entre todos los datos y el centro global)
  - WSS = variación intra-cluster (entre los puntos de cada cluster y sus centroides)
  - BSS = variación inter-cluster (entre los centroides de los clusters y el centro global).  
 $BSS = TSS - WSS$
  - CH = ratio entre la variación entre clusters (BSS) y el promedio de la variación interna de los clusters (WSS). Se busca maximizar CH:

$$CH = \frac{BSS}{WSS} * \frac{N - k}{k - 1}$$



# BOOTSTRAP DE LOS CLUSTERS CREADOS

- La robustez del clustering depende de la escogencia adecuada del **k**
  - Algunos clusters encontrados representan una estructura bien determinada
  - Otros terminan siendo repositorios de las instancias que no encajan en ninguno de los clusters estructurales
- Clusters estructurales: estables y resistentes a cambios de los datos
  - Repetición del clustering a partir de técnicas de resampleo (e.g. bootstrapping)
  - Algoritmo de clustering bootstrap
    1. Crear un clustering del dataset original
    2. Crear una nuevo dataset del mismo tamaño a partir del original, con repeticiones, y realizar su clustering
    3. Para cada cluster original encontrar el cluster resampleado más similar, utilizando la medida de Jaccard:  $Jaccard(A, B) = \frac{A \cap B}{A \cup B}$ . Si esta medida es inferior a 0,5, el cluster correspondiente se considera inestable y se disuelve. Esto es una indicación de que no es buen cluster.
    4. Se repiten los pasos 2 y 3 varias veces



# TALLER: EVALUACIÓN DE CLUSTERING

Continuar con el taller de clustering de clientes de supermercado 09-SUPERMERCADOS- K-Means-STUD.html con la parte dedicada a la determinación y evaluación del número de clusters.



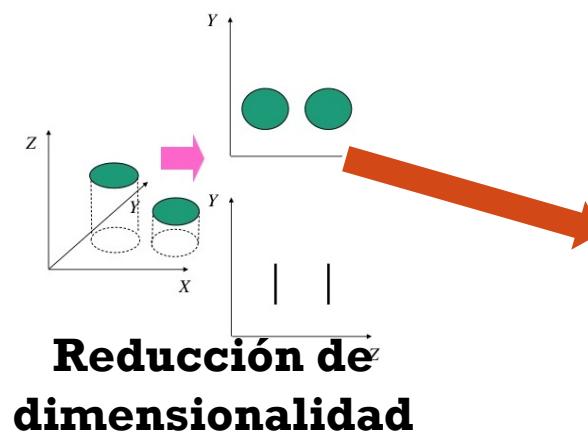
# APRENDIZAJE NO SUPERVISADO



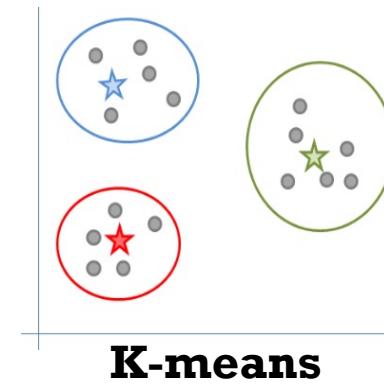
# AGENDA



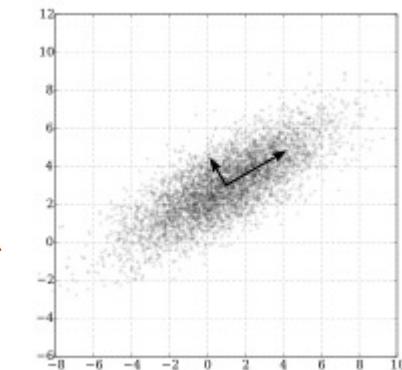
Clustering



Reducción de  
dimensionalidad



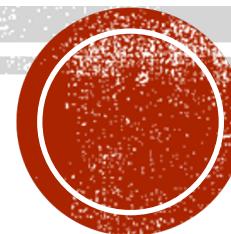
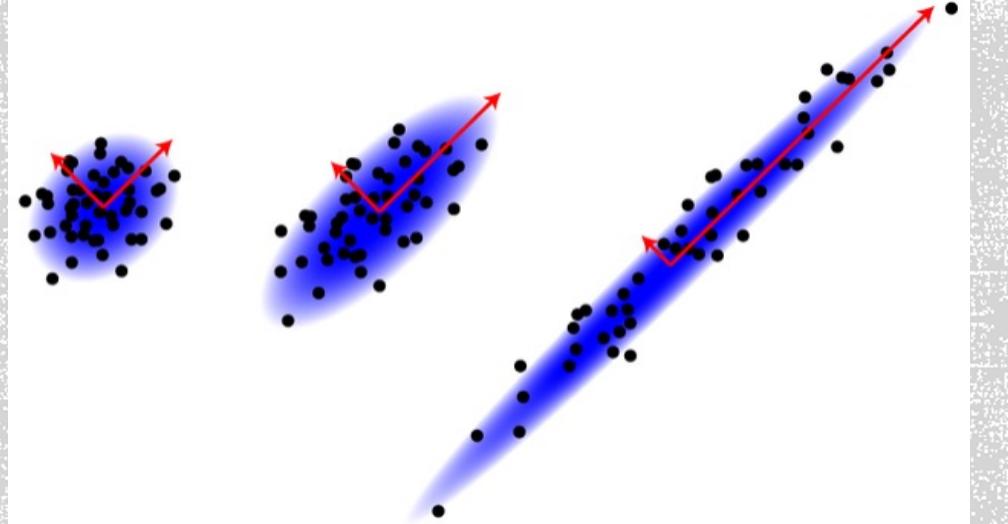
K-means



Componentes  
principales



# COMPONENTES PRINCIPALES

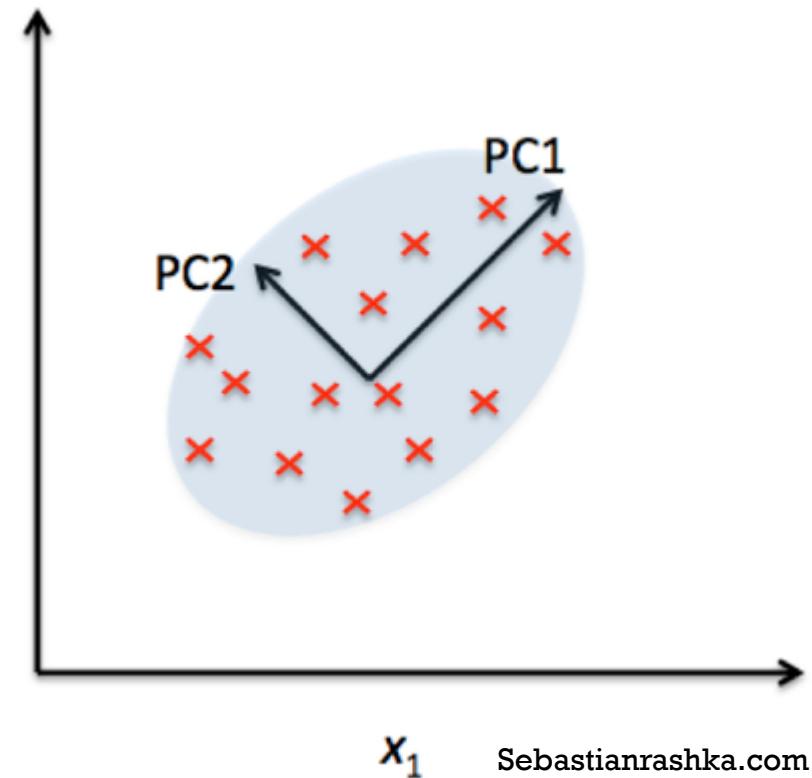


# COMPONENTES PRINCIPALES

## PCA: Principal Component Analysis

**Objetivo:** Simplificar el dataset, encontrando una representación de **baja dimensionalidad** que conserva la mayor parte de la información

- **Combinación lineal** de las dimensiones (atributos) originales del dataset que maximiza la varianza
- **Rotación** de los ejes originales
- Permite una **visualización** los datos en problemas de aprendizaje supervisado y no supervisado
- Se limitan las dimensiones que estén altamente **correlacionadas** entre ellas
- PCA permite encontrar la superficie lineal de menos dimensiones más cercana a los puntos en el espacio original (en distancia Euclidiana)



$x_1$

Sebastianrashka.com



# COMPONENTES PRINCIPALES

- Hay tantos componentes principales (PCs) como dimensiones, ortogonales entre ellos
- Cada PC es una combinación lineal normalizada de los atributos del dataset ( $X_1, X_2, \dots, X_N$ ), se buscan los vectores que maximicen la varianza

$$PC_i = \Phi_{1i}X_1 + \Phi_{2i}X_2 + \dots + \Phi_{Ni}X_N, \text{ sujeto a } \sum_{j=1}^N \Phi_{ji}^2 = 1$$

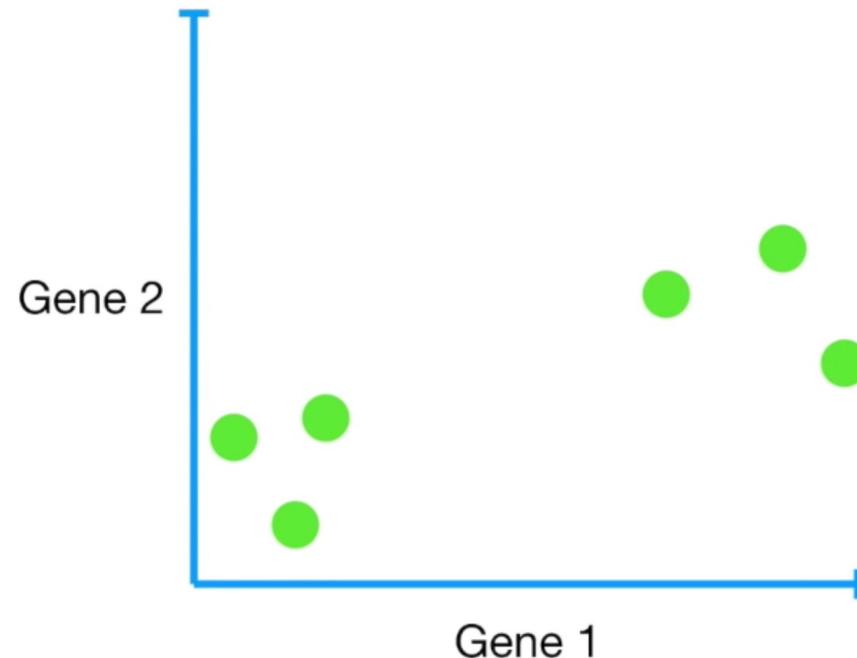
$$\text{Además } \forall j < i, PC_i \perp PC_j$$

- Cada PC tiene asociada una carga o **loading** de cada una de las dimensiones originales (los  $\Phi_{ji}$ ). El vector de loadings de una variable original indica su dirección en el espacio de los PC
- Solo existe una solución posible de espacio de componentes principales, conservando siempre las direcciones aunque puede que el sentido sea el contrario.
- A cada PC se le puede establecer la cantidad de información original especificada (Proporción de varianza explicada). Esta va decreciendo con cada PC considerado, por lo que los primeros  $p$  PCs van a representar mucha más información que las primeras  $p$  dimensiones originales
- Las instancias originales se proyectan en el espacio dado por los primeros  $p$  PCs



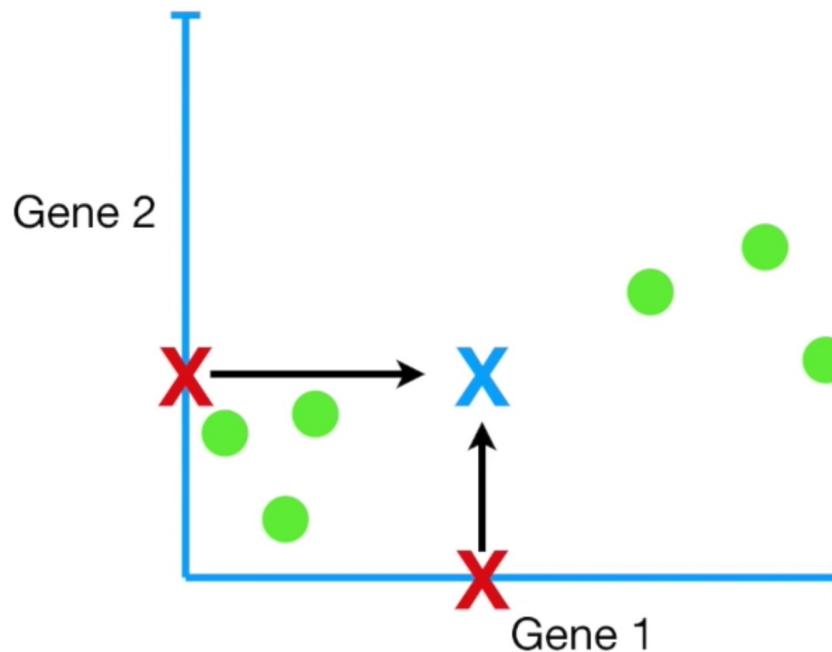
# COMPONENTES PRINCIPALES

	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1



# COMPONENTES PRINCIPALES

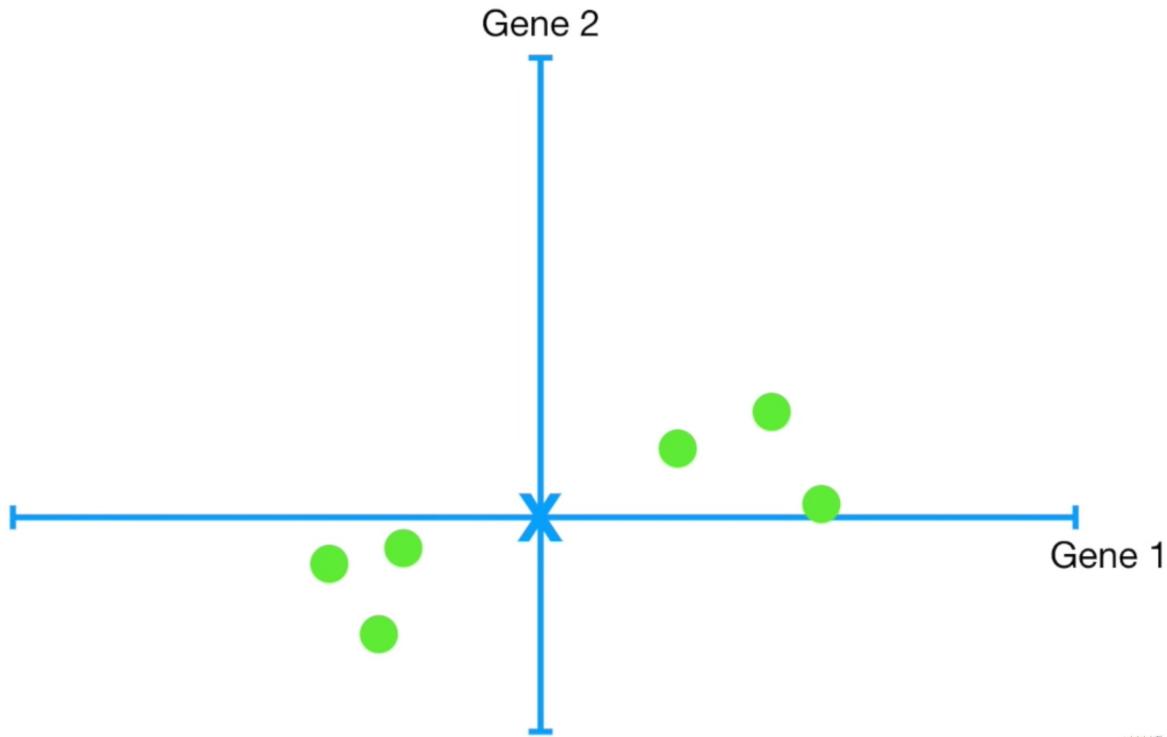
	Mouse					
	1	2	3	4	5	6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1



<https://www.youtube.com/watch?v=FgakZw6K1OQ>



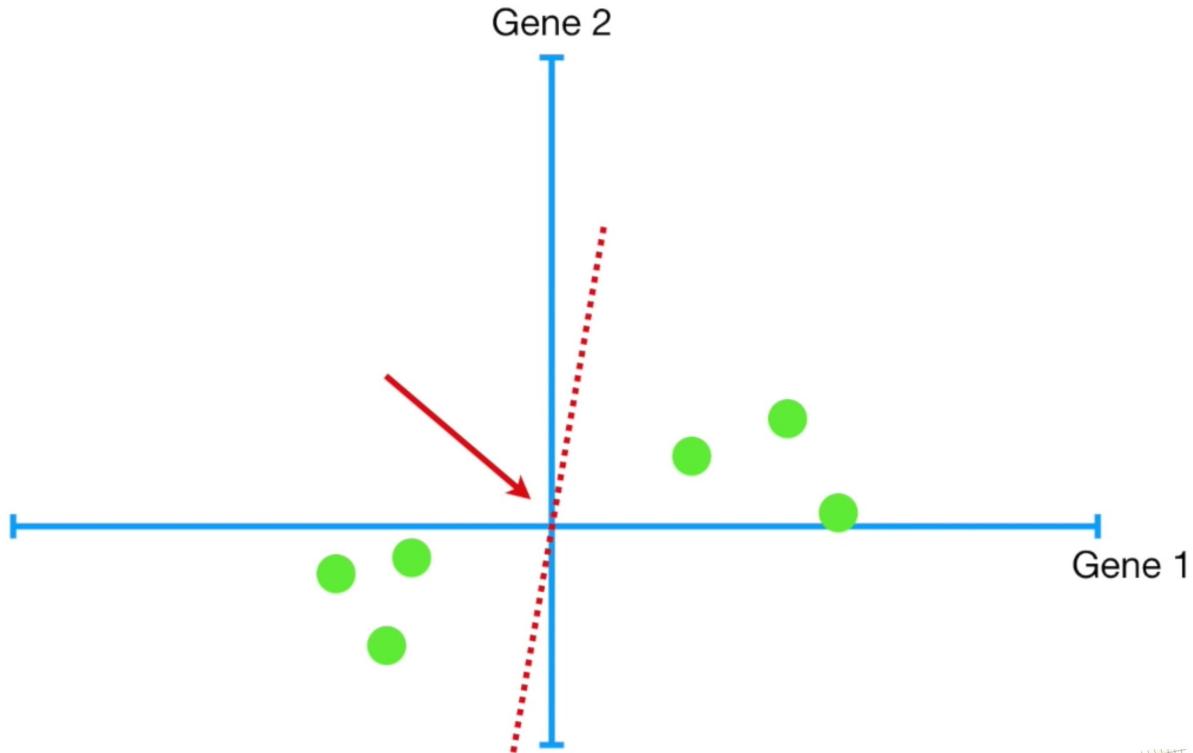
# COMPONENTES PRINCIPALES



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



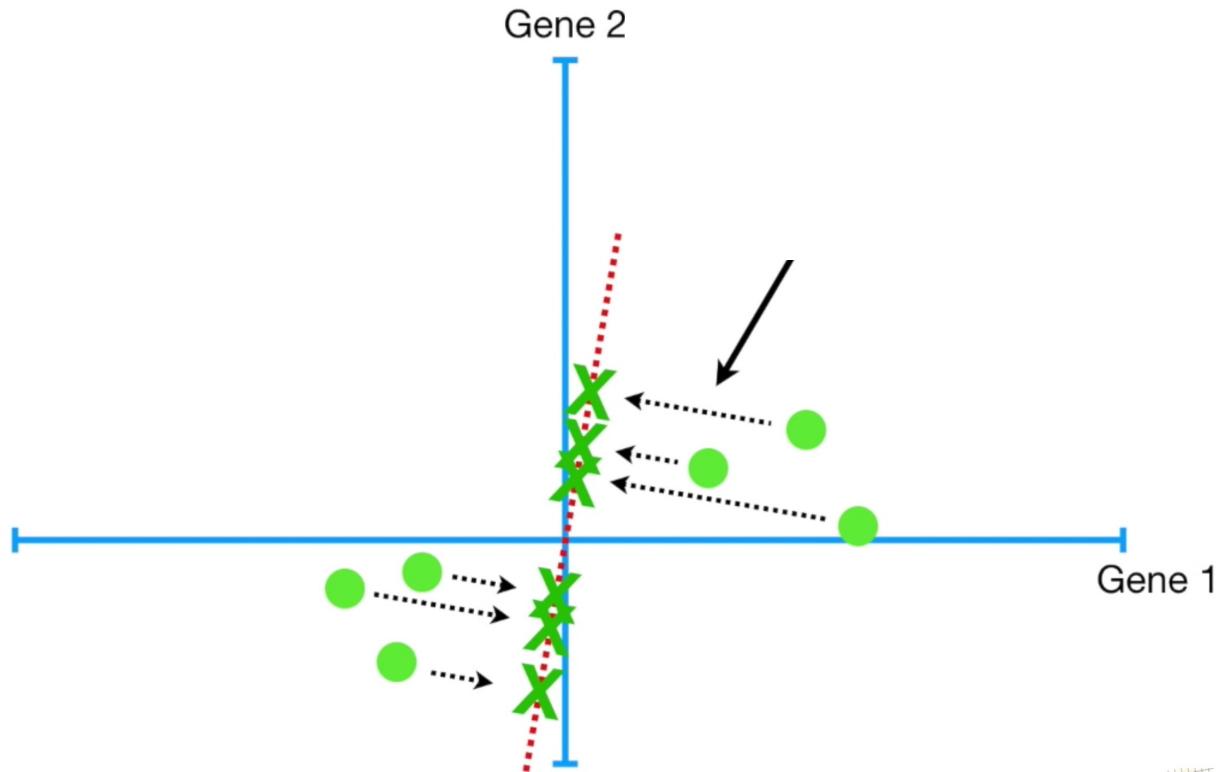
# COMPONENTES PRINCIPALES



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



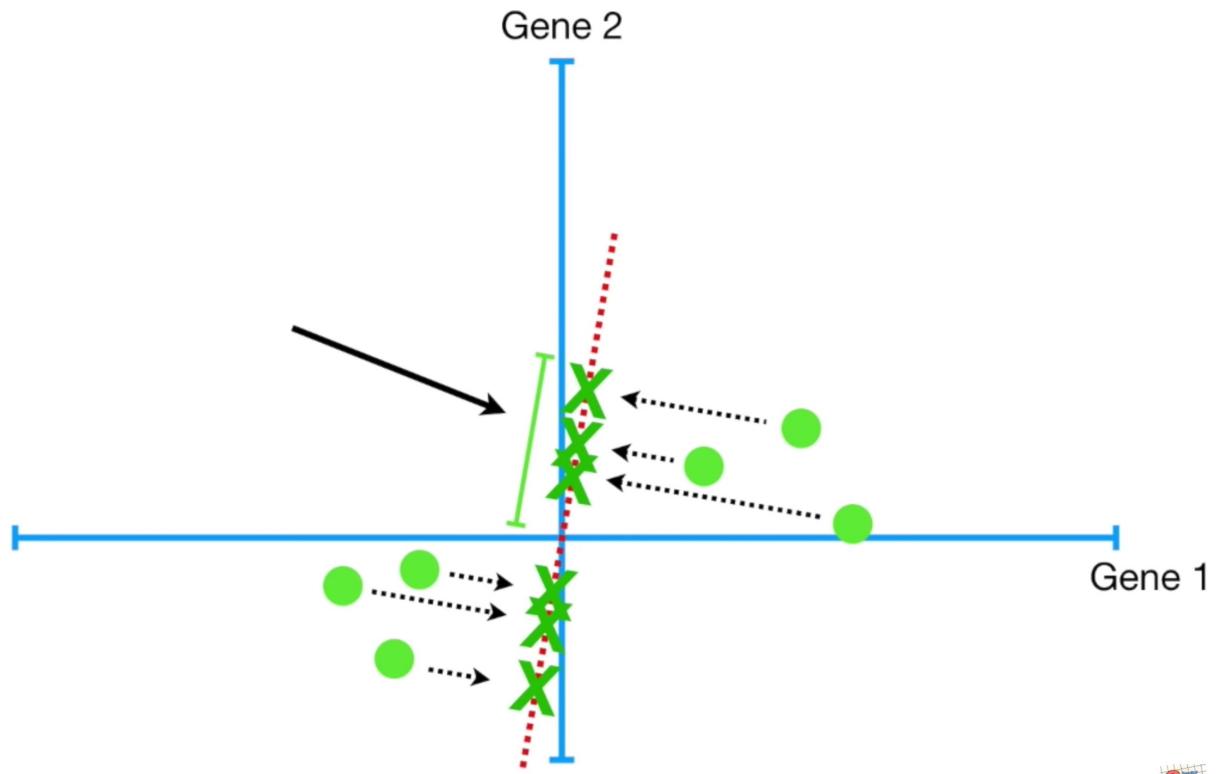
# COMPONENTES PRINCIPALES



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

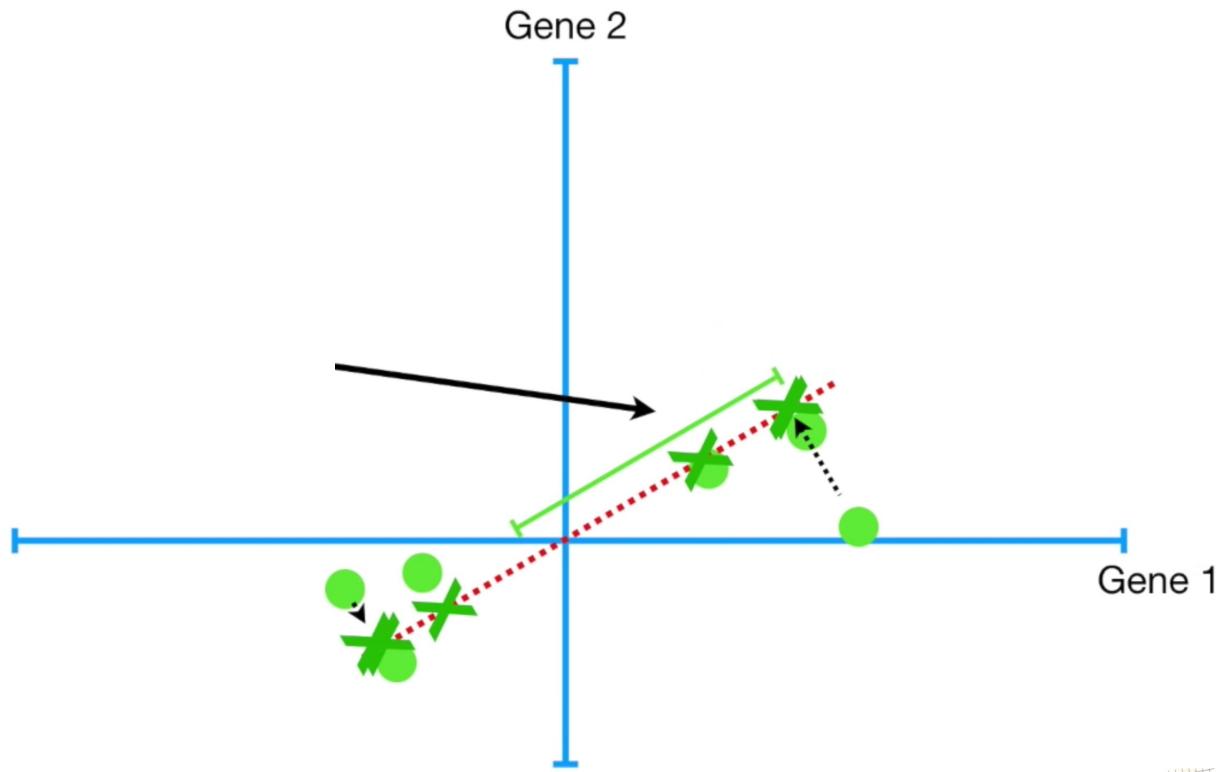


WIKI

<https://www.youtube.com/watch?v=FgakZw6K1QQ>



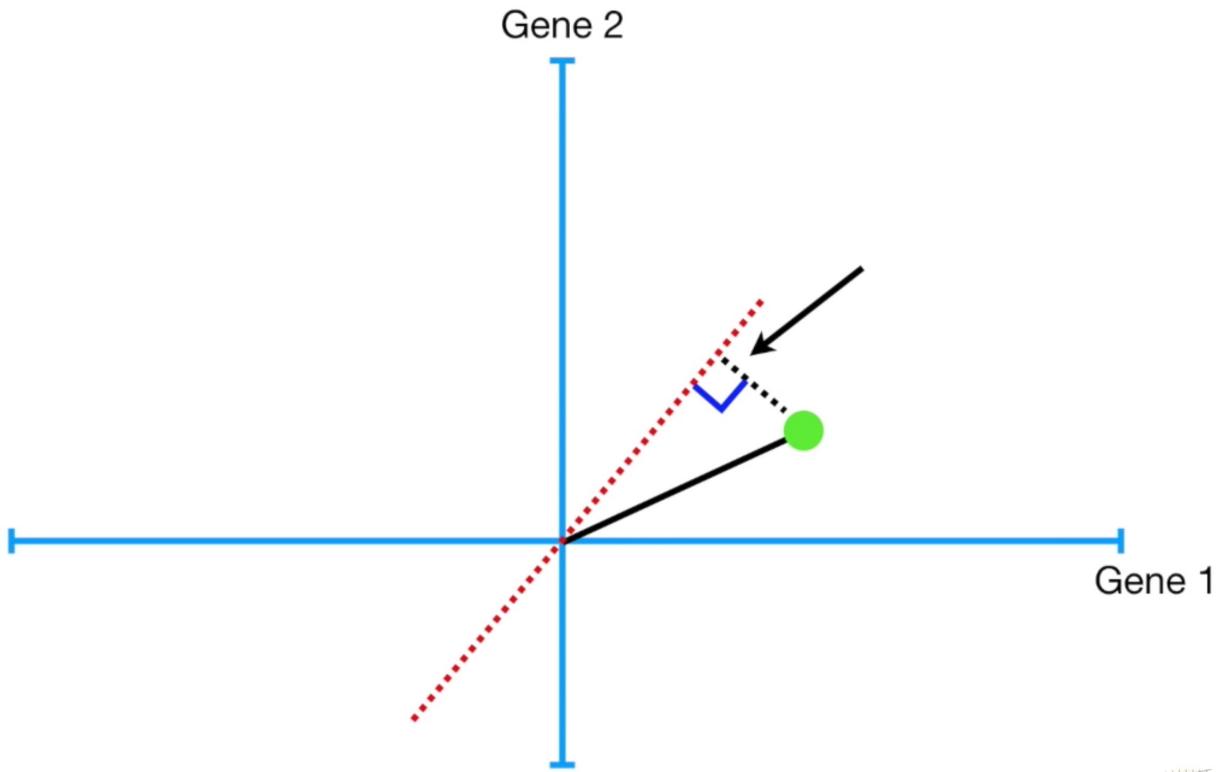
# COMPONENTES PRINCIPALES



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES



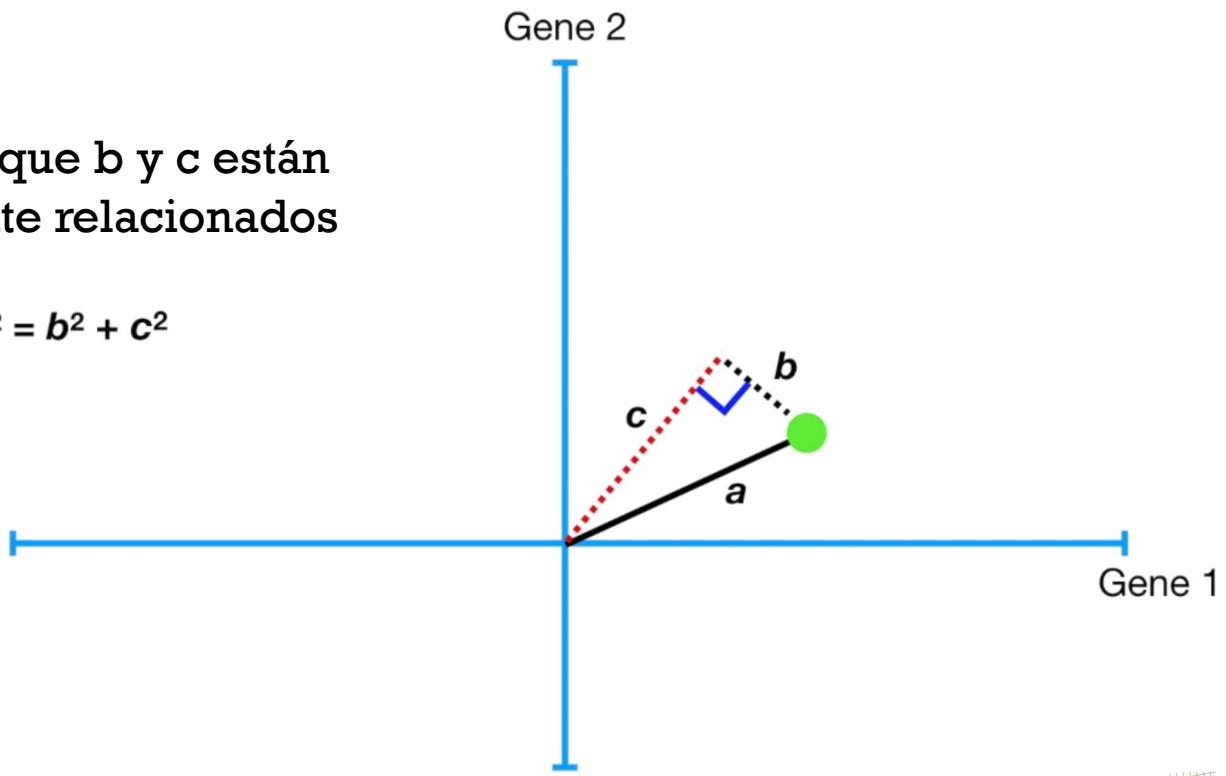
<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

Se muestra que b y c están inversamente relacionados

$$a^2 = b^2 + c^2$$

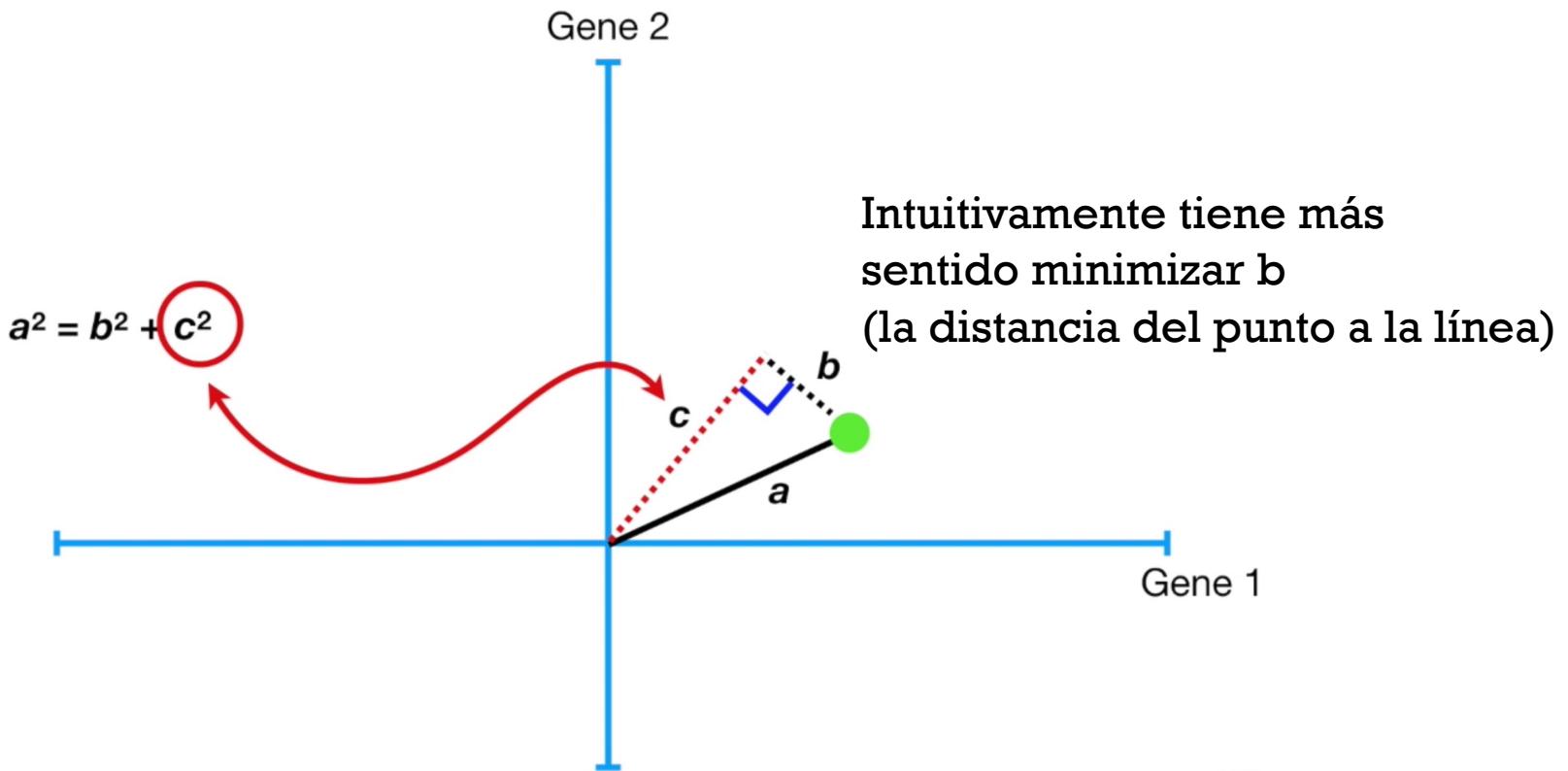


1/1

<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES



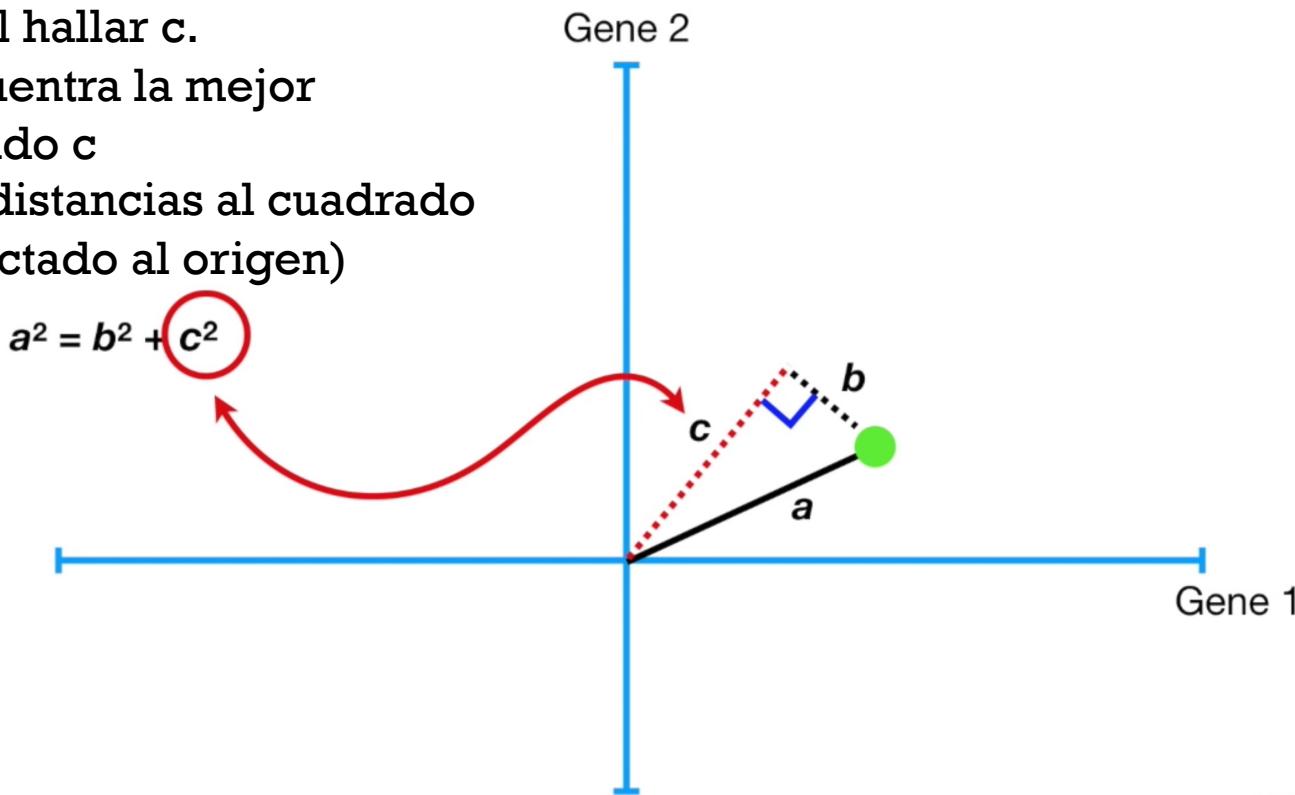
<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

Pero es más fácil hallar c.

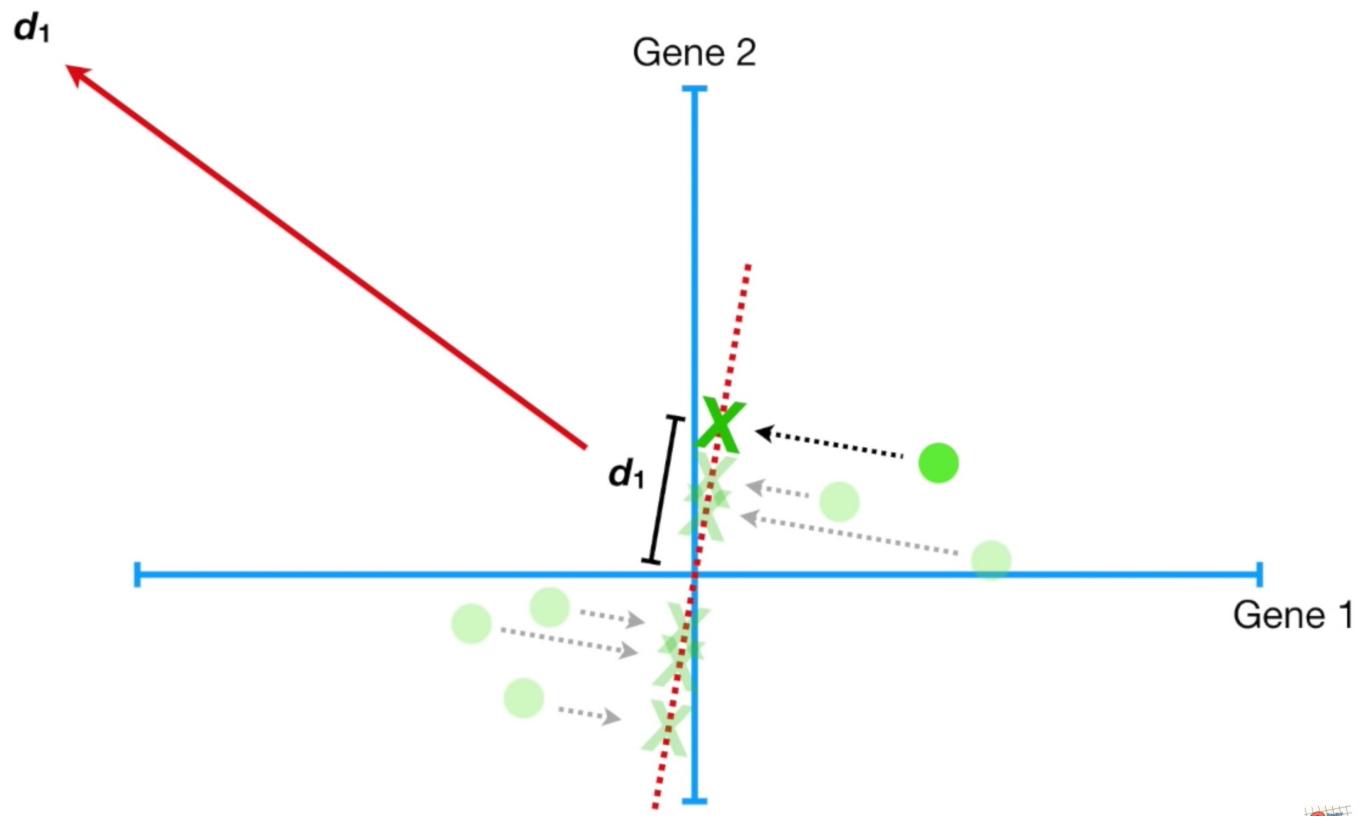
Luego PCA encuentra la mejor  
línea maximizando c  
(la suma de las distancias al cuadrado  
Del punto proyectado al origen)



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



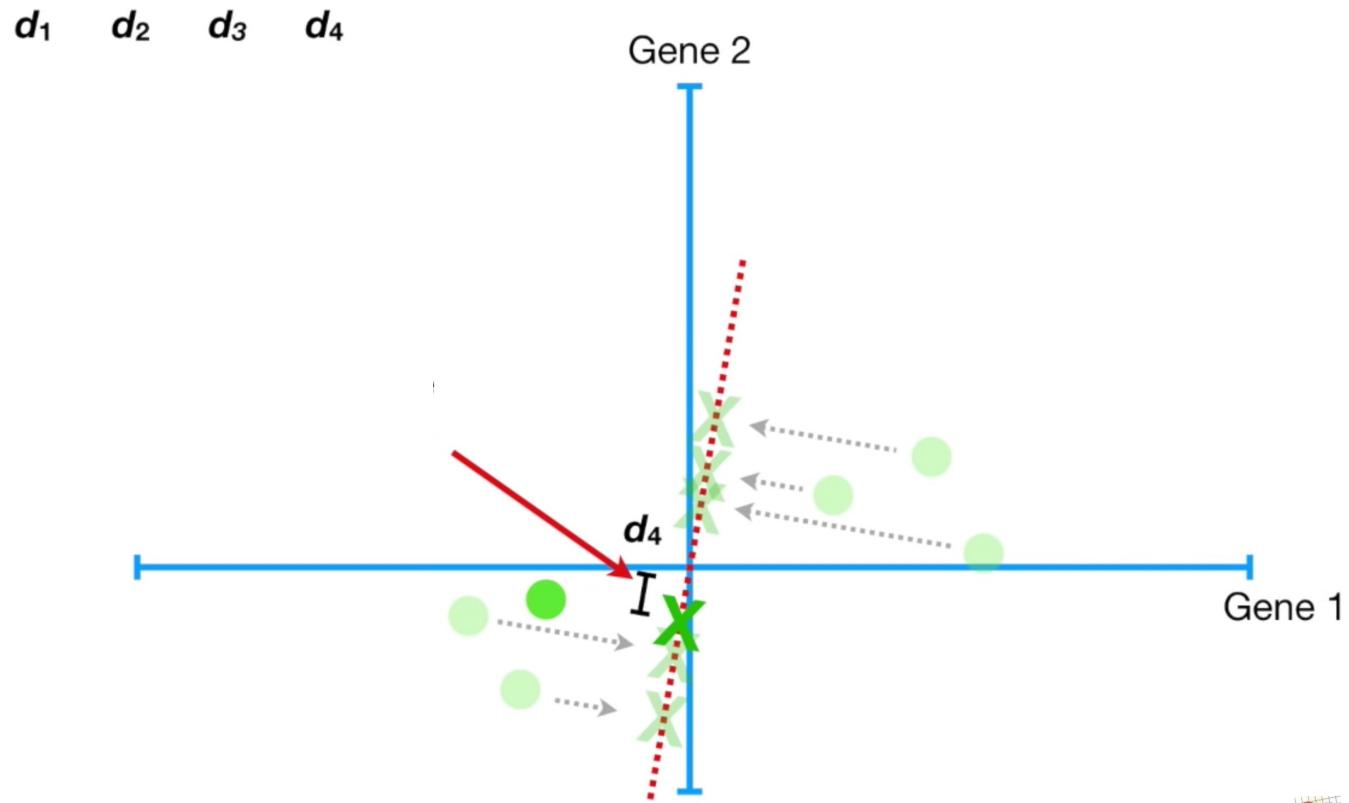
# COMPONENTES PRINCIPALES



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

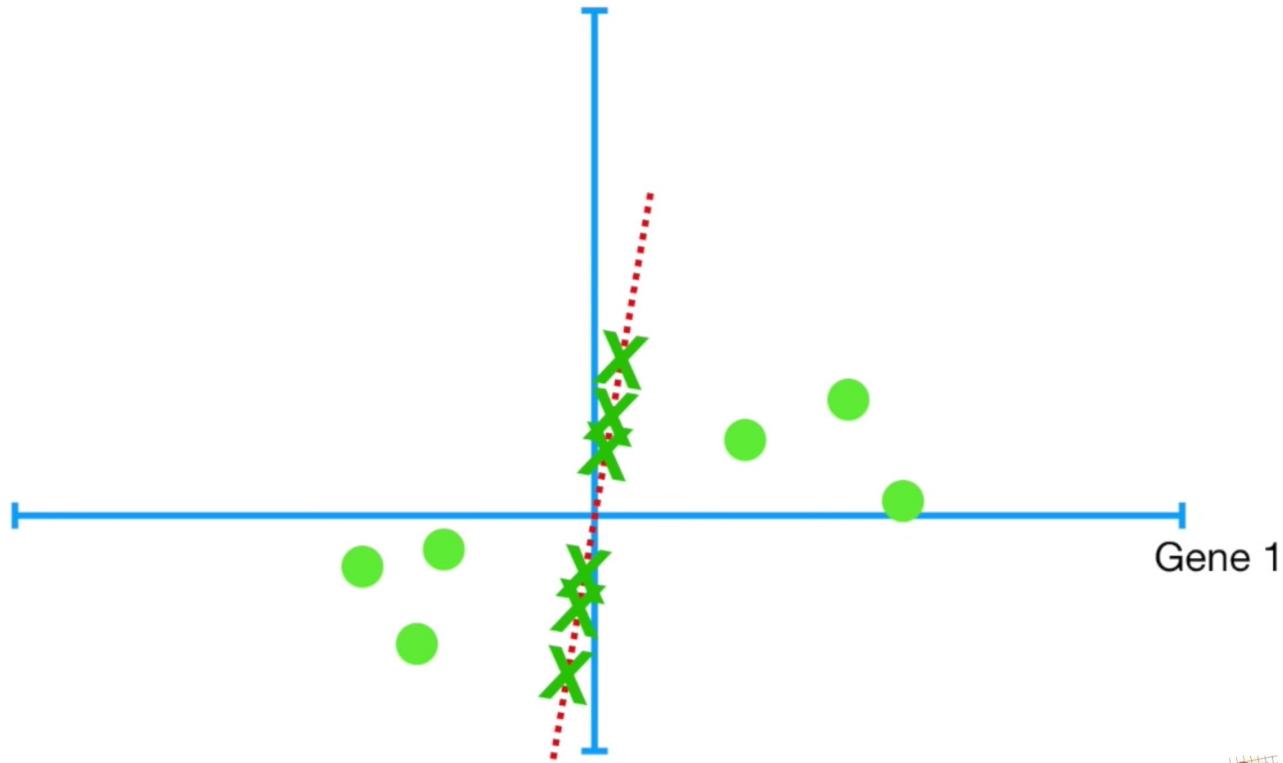


<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

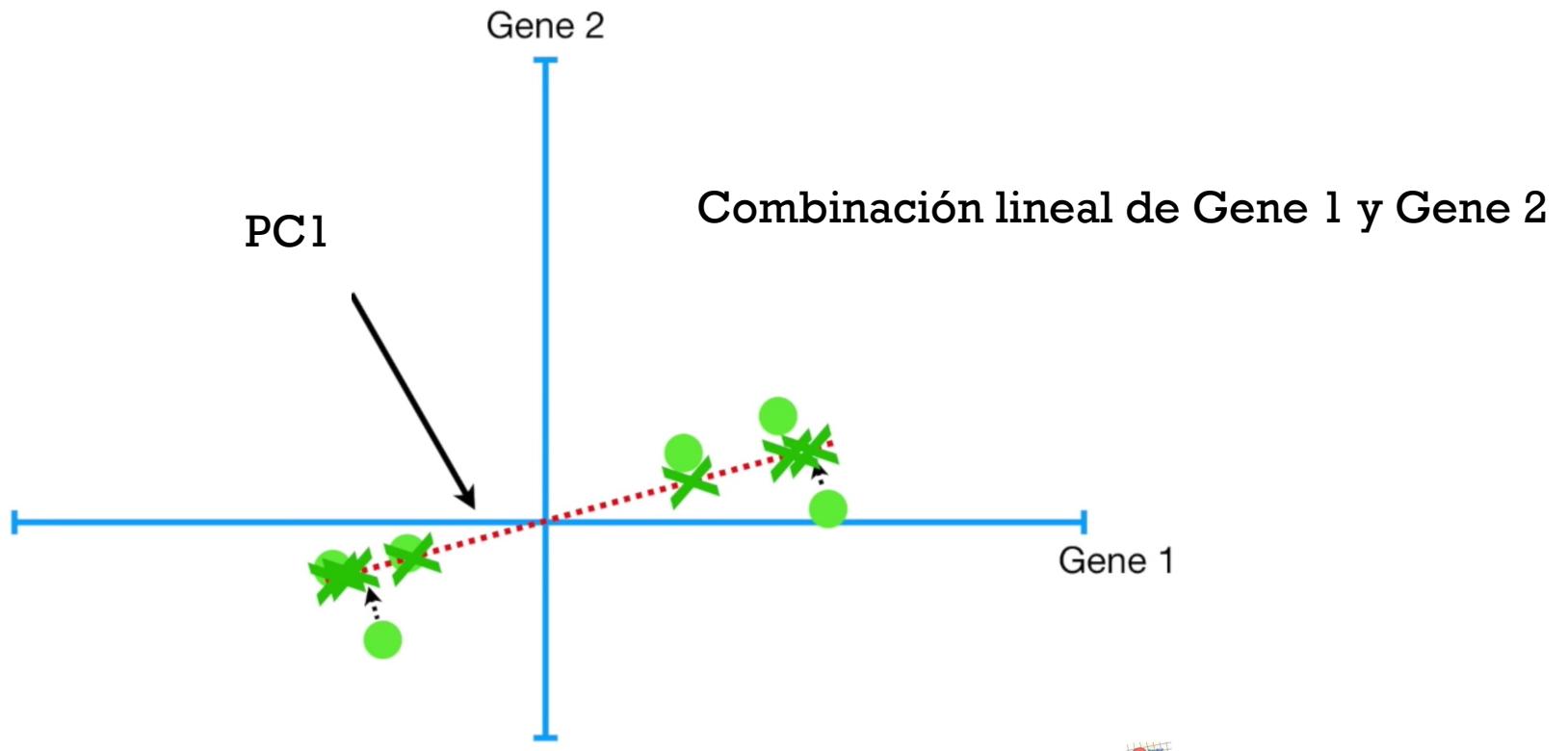
$$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2 = \text{sum of squared distances} = \text{SS(distances)}$$



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



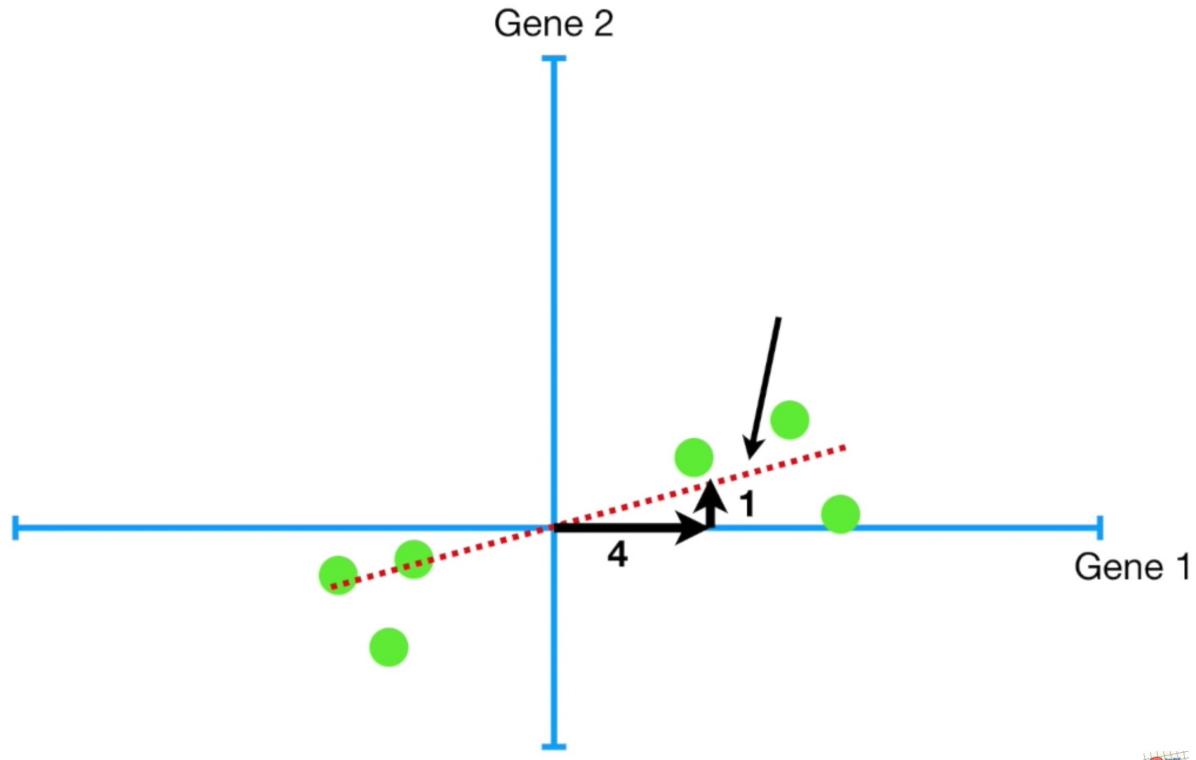
# COMPONENTES PRINCIPALES



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



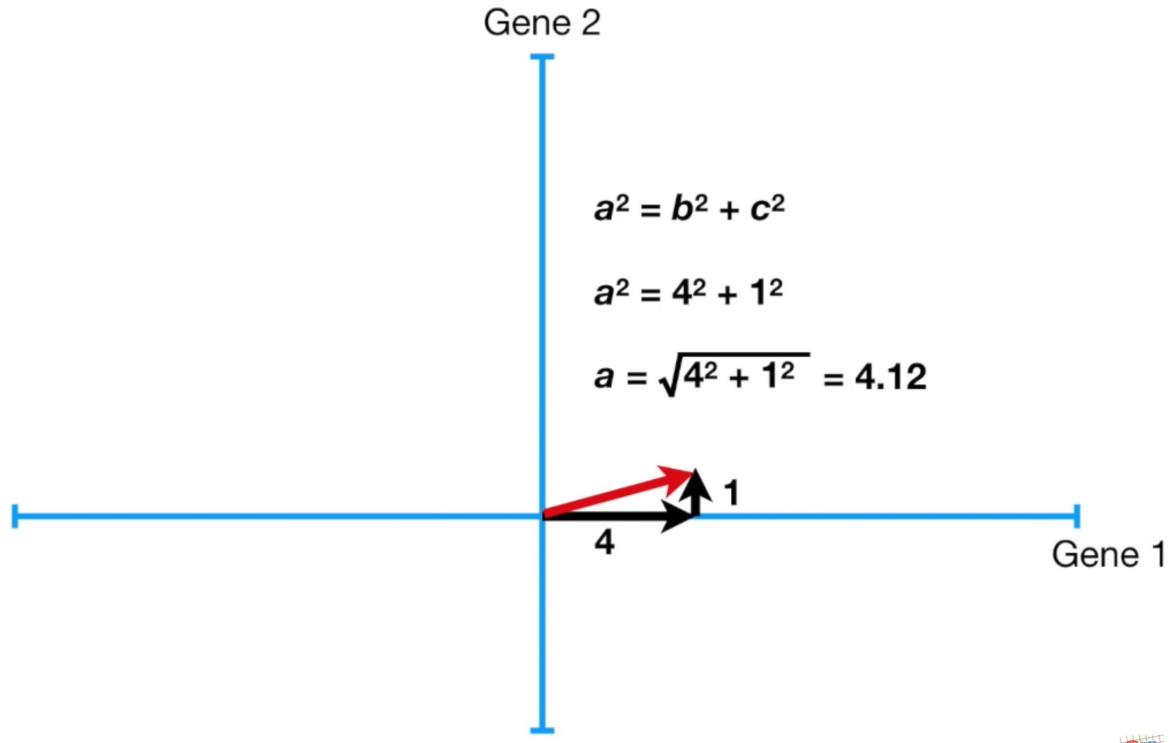
# COMPONENTES PRINCIPALES



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

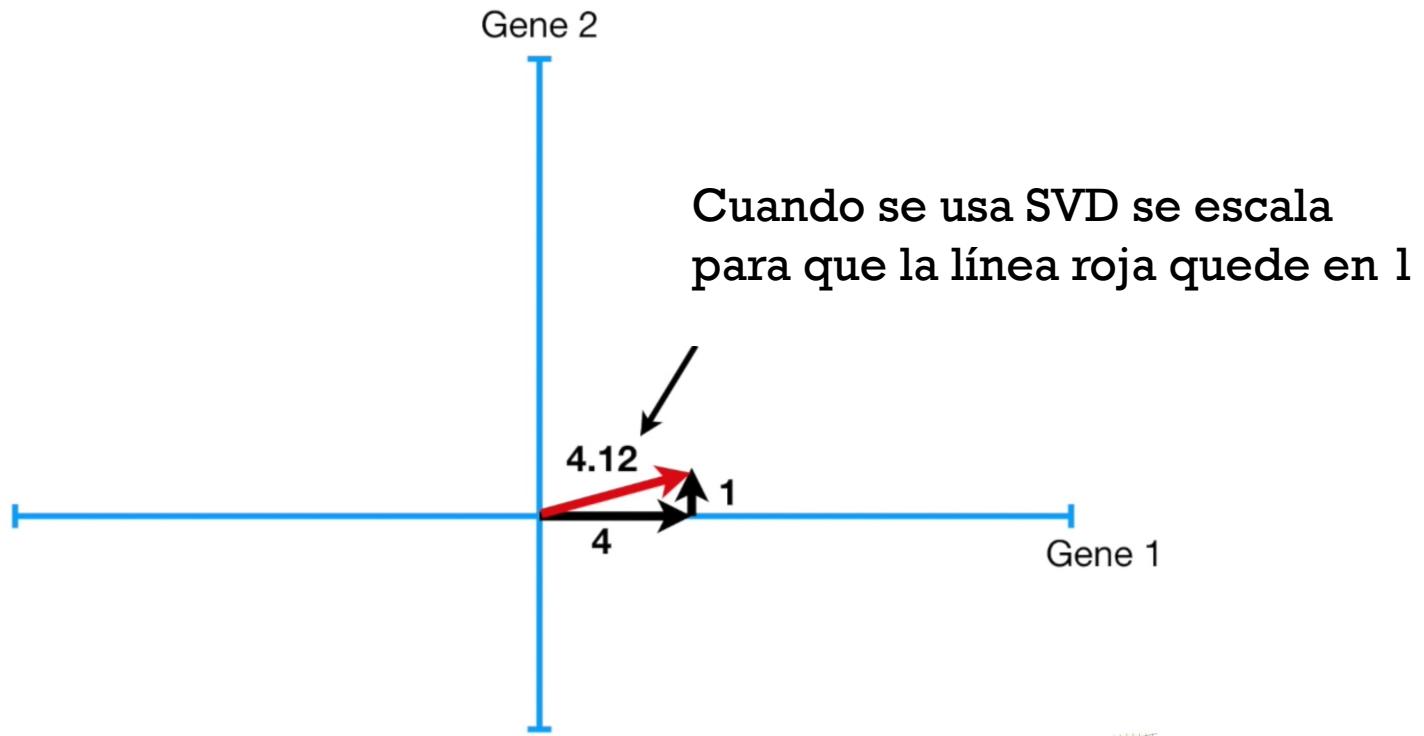


L10-1

<https://www.youtube.com/watch?v=FgakZw6K1QQ>



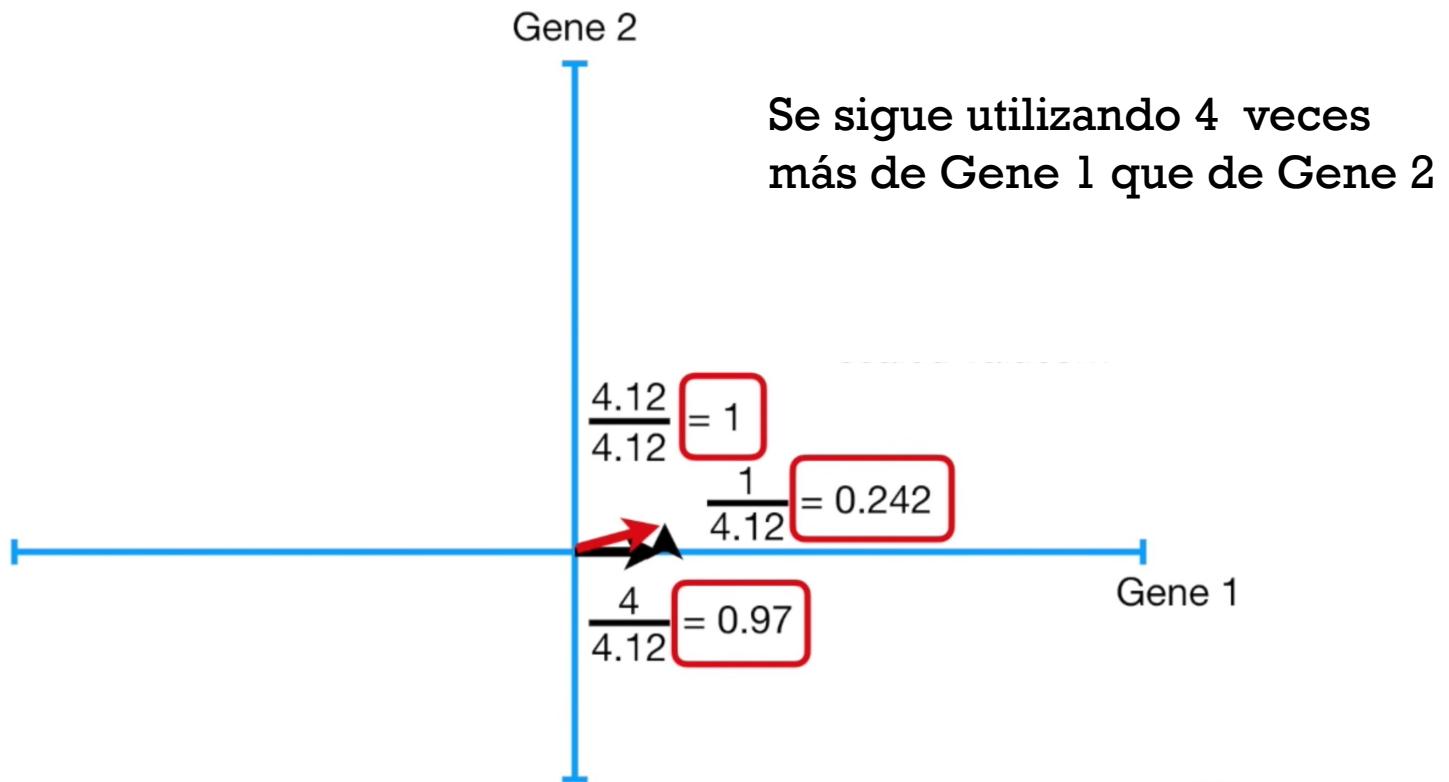
# COMPONENTES PRINCIPALES



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



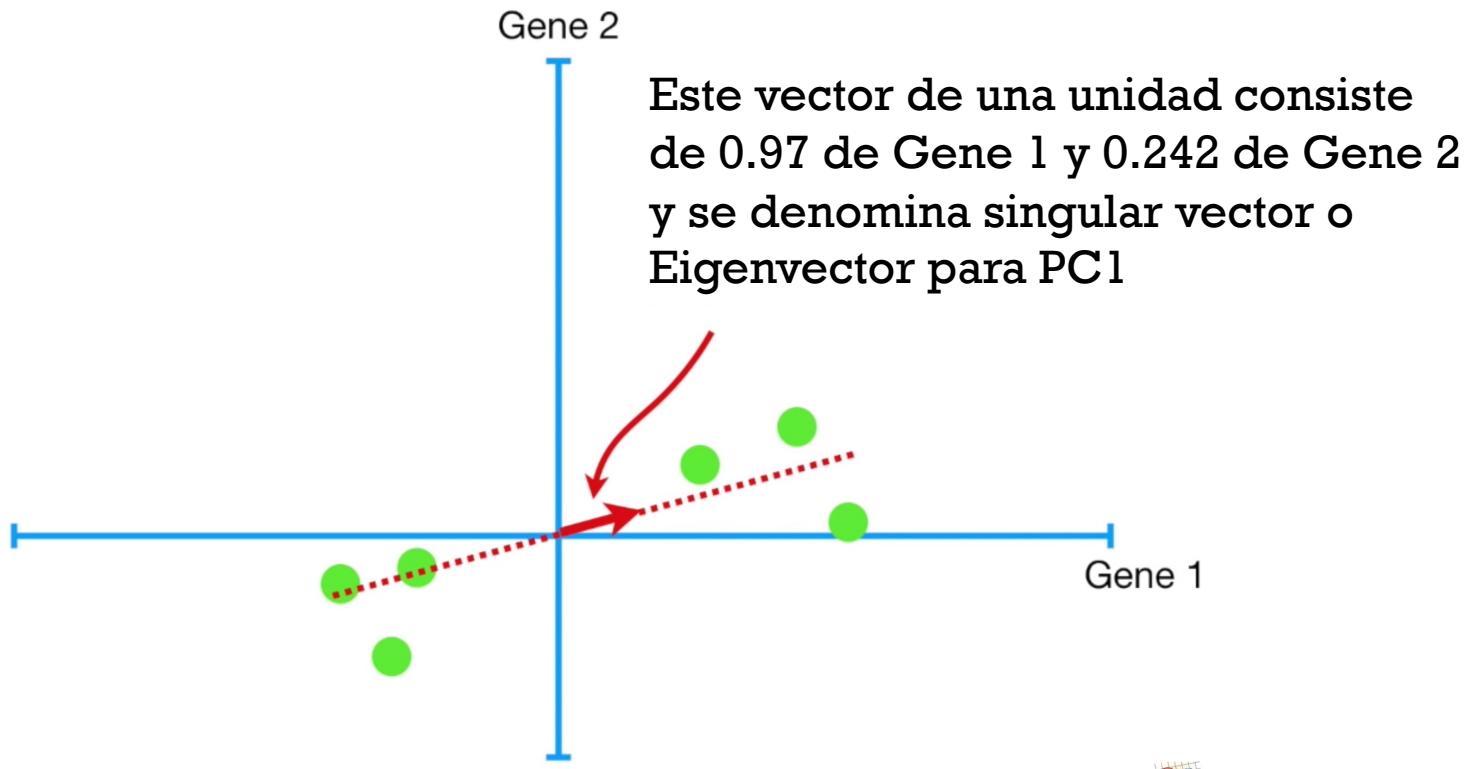
# COMPONENTES PRINCIPALES



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

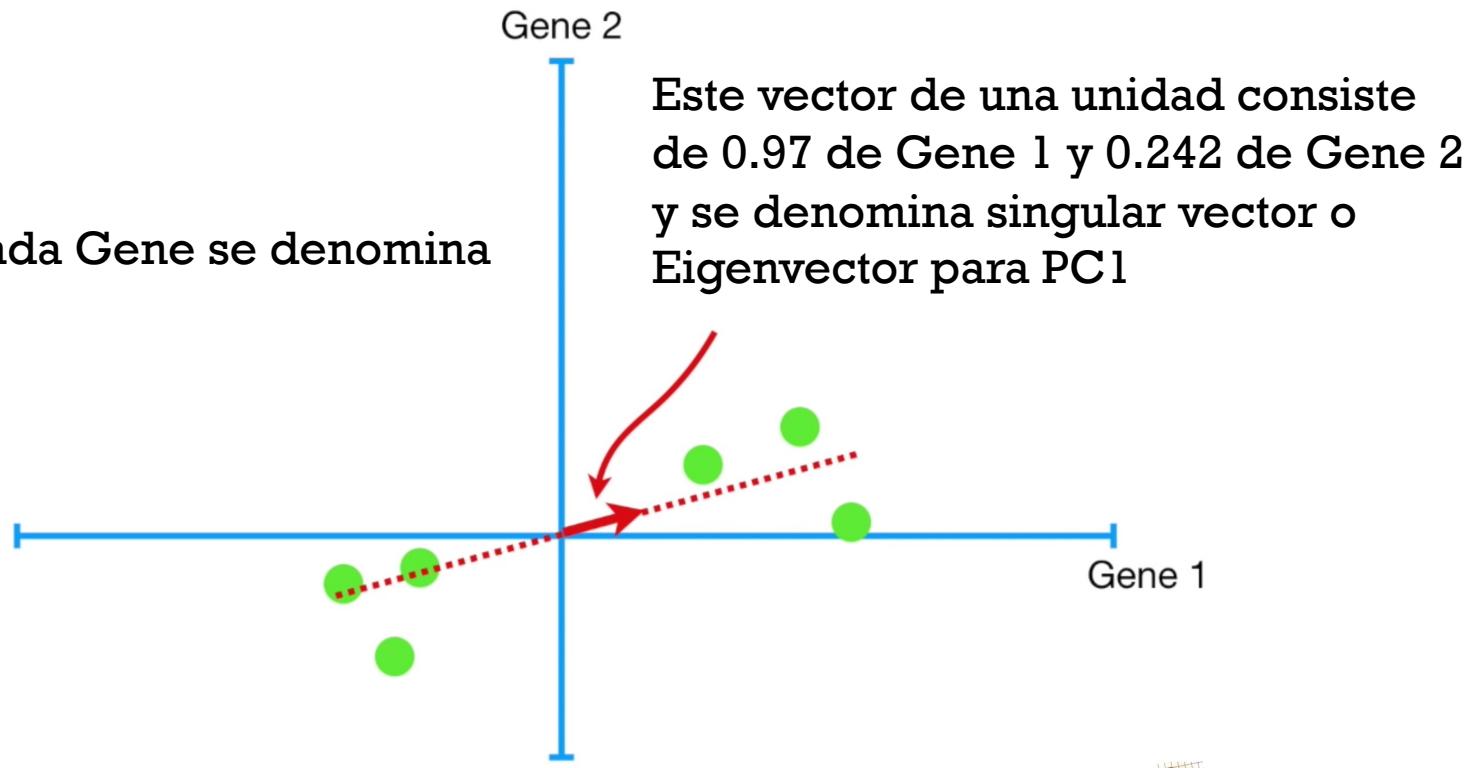


<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

La proporción de cada Gene se denomina loading o carga

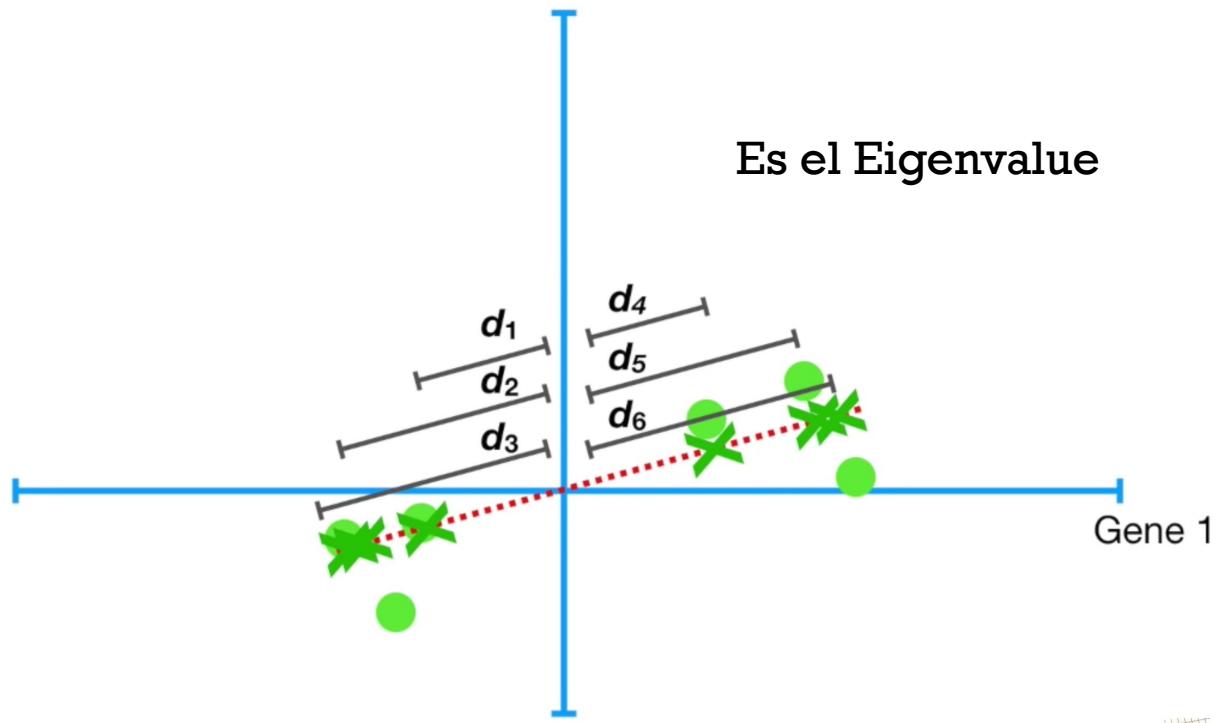


<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

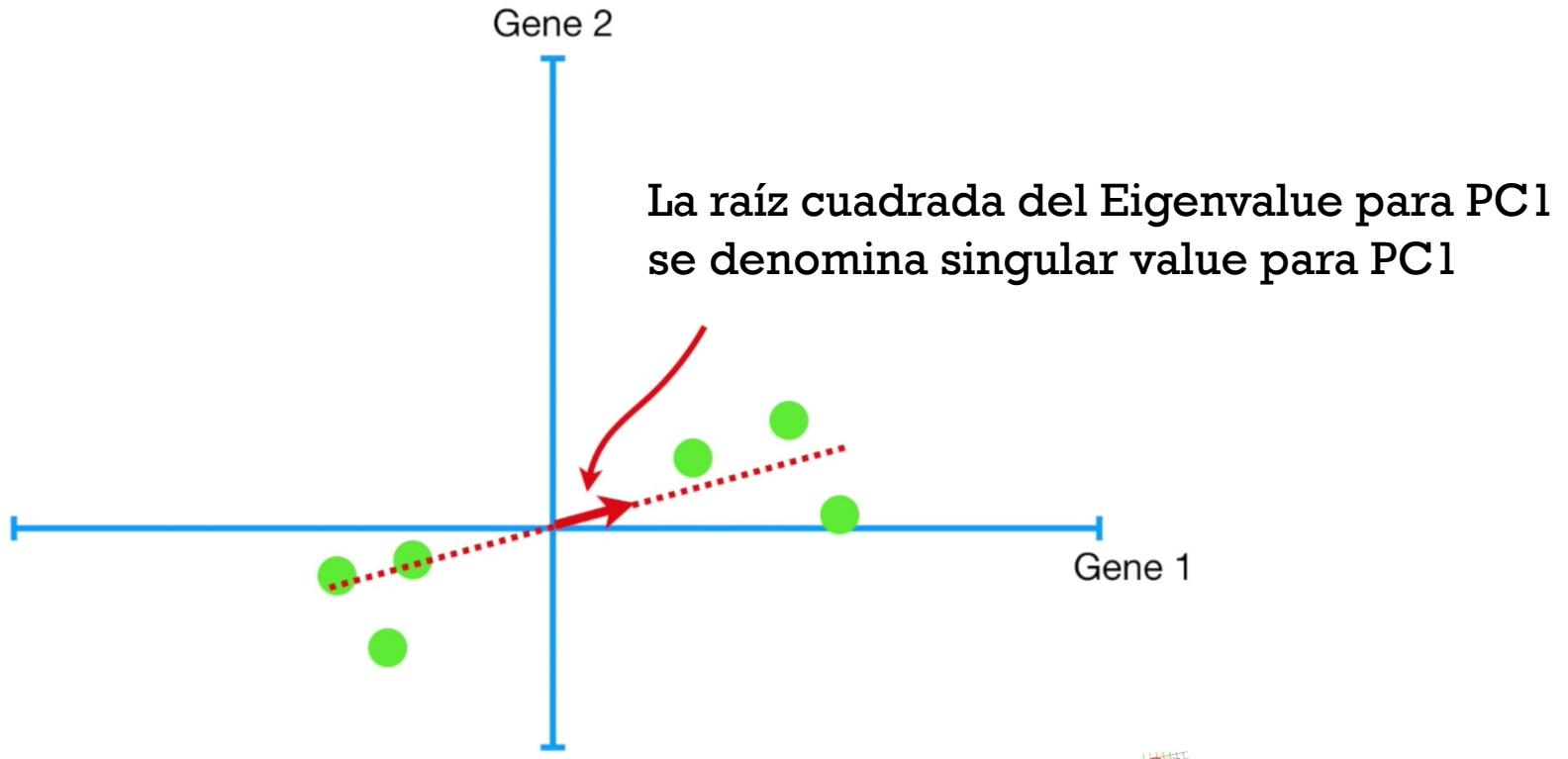
$$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2 = \text{sum of squared distances} = \text{SS(distances)}$$



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

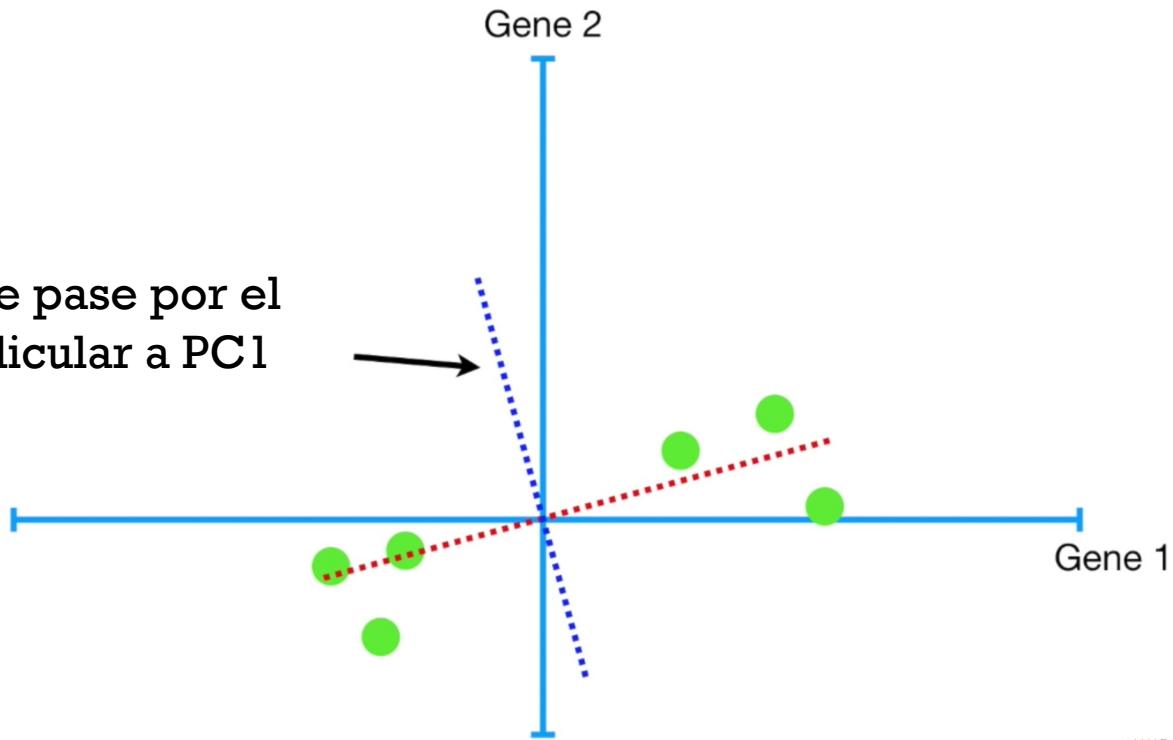


<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

Se escoge la línea que pase por el origen y sea perpendicular a PC1

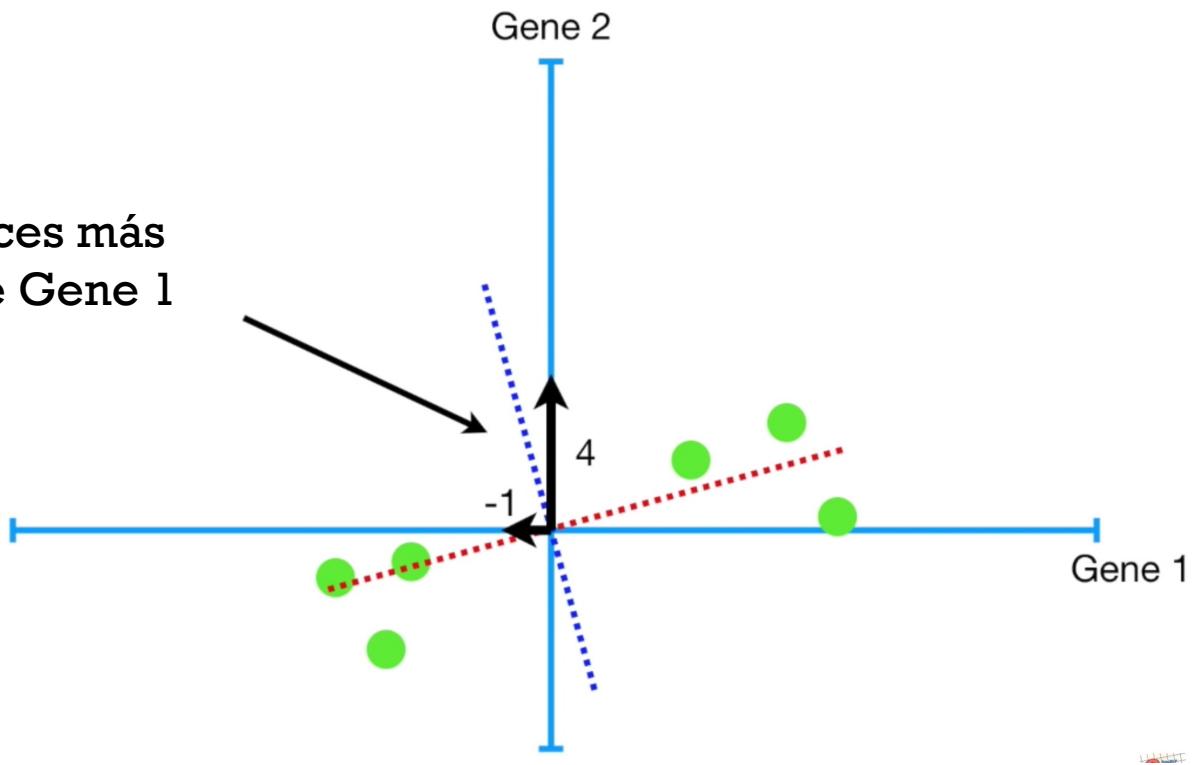


<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

Gene 2 es 4 veces más importante que Gene 1

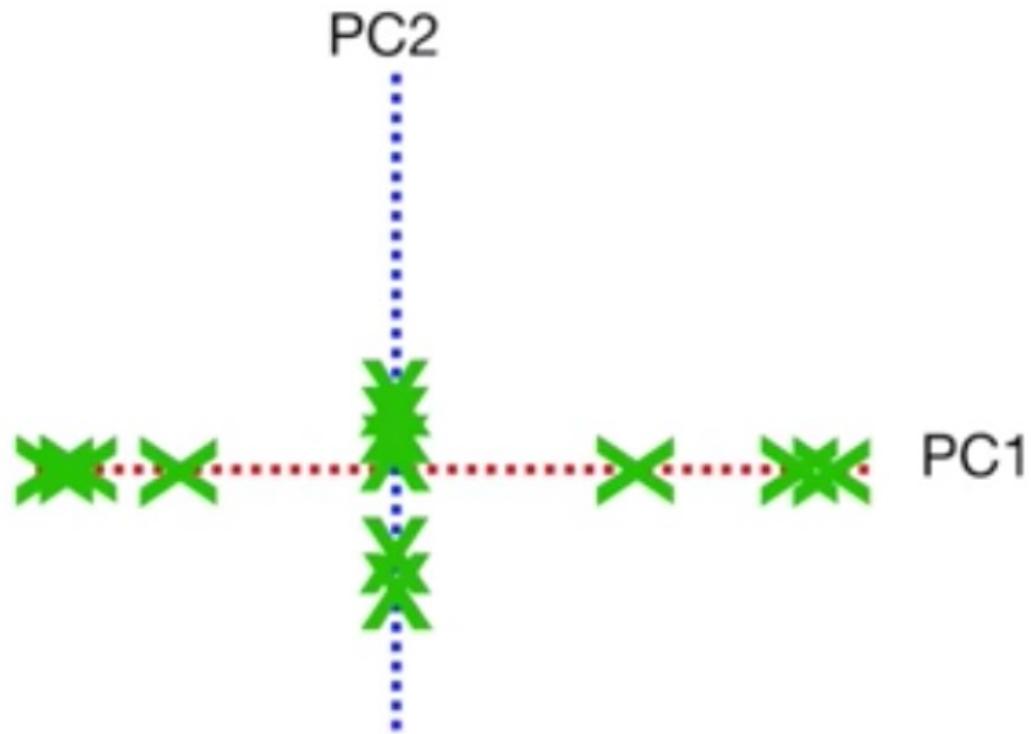


<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

Rotamos y utilizamos los puntos  
Proyectados para encontrar las  
muestras

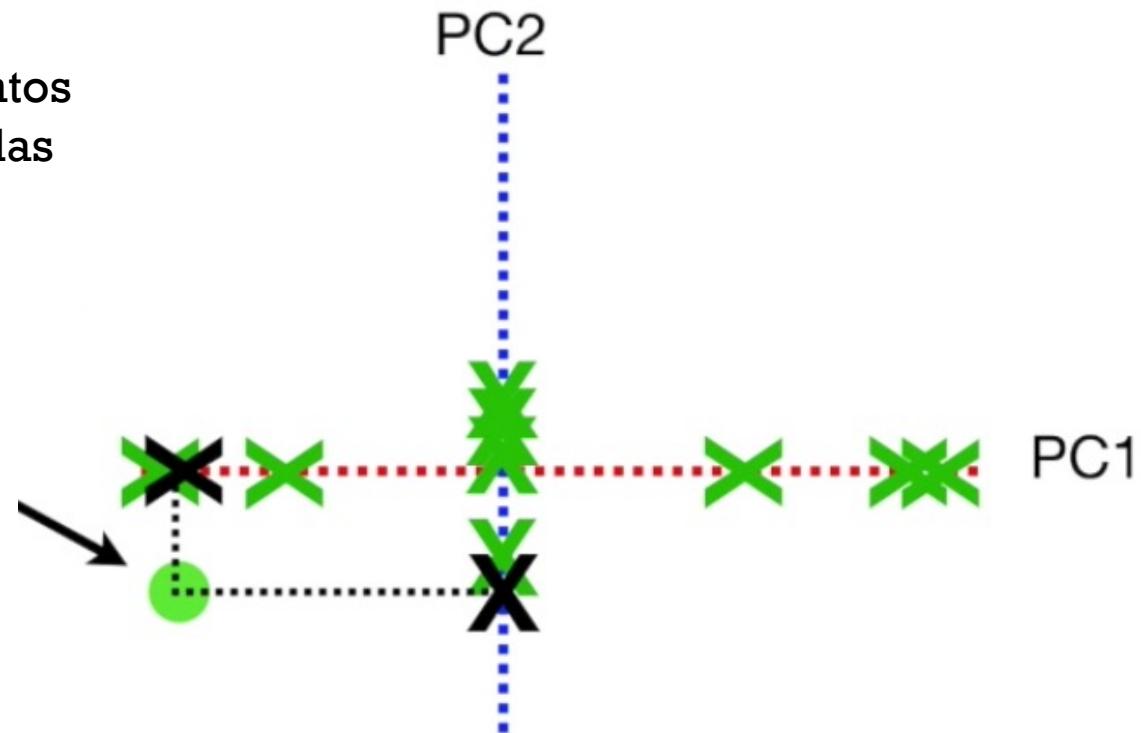


<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

Rotamos y utilizamos los puntos  
Proyectados para encontrar las  
muestras

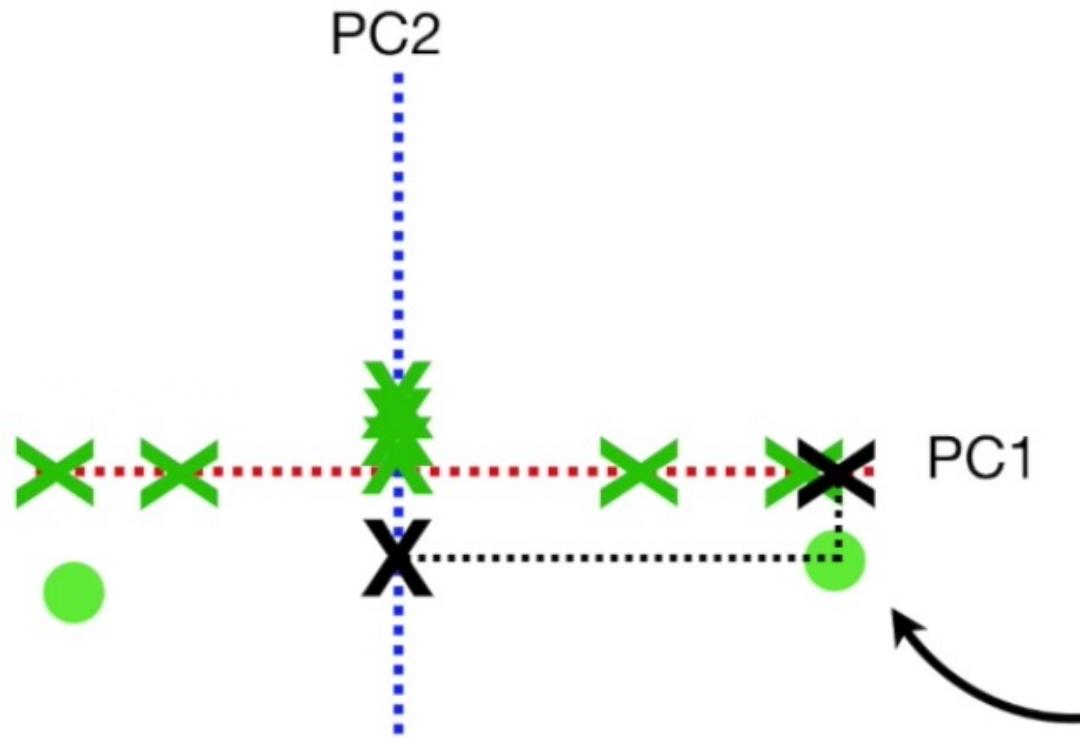


<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

Rotamos y utilizamos los puntos  
Proyectados para encontrar las  
muestras

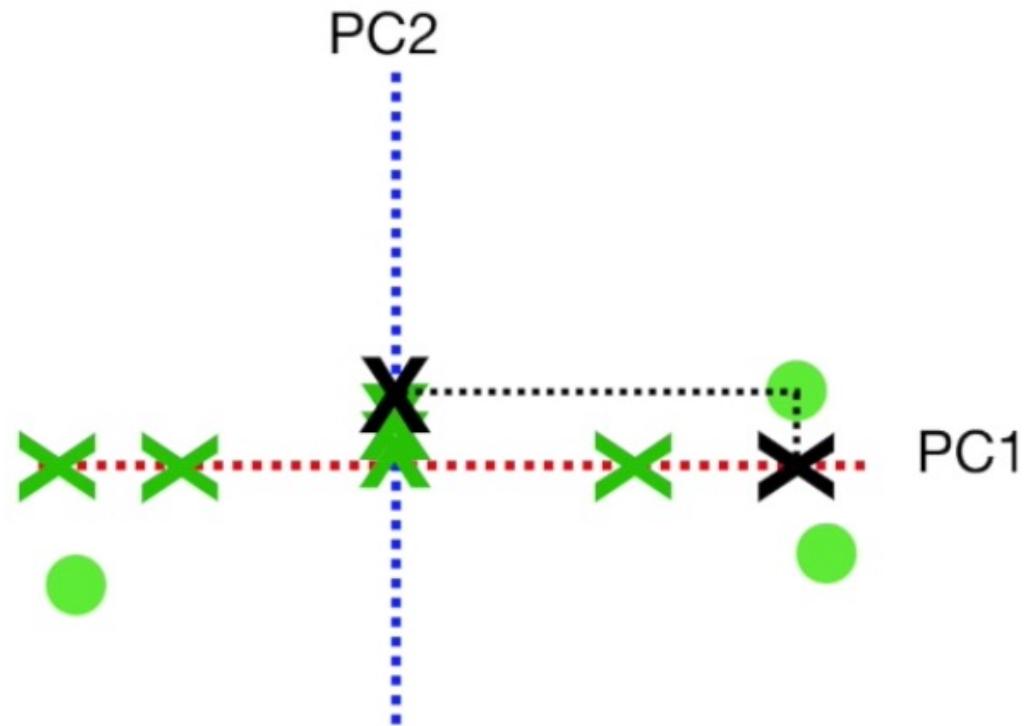


<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

Rotamos y utilizamos los puntos  
Proyectados para encontrar las  
muestras

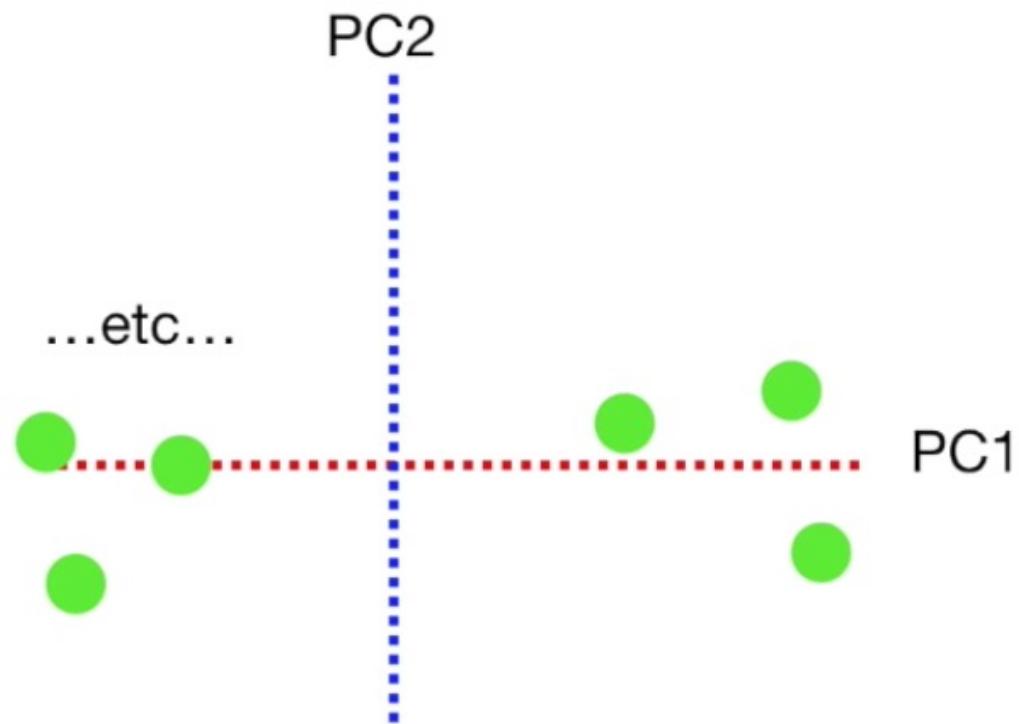


<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES

Rotamos y utilizamos los puntos  
Proyectados para encontrar las  
muestras



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



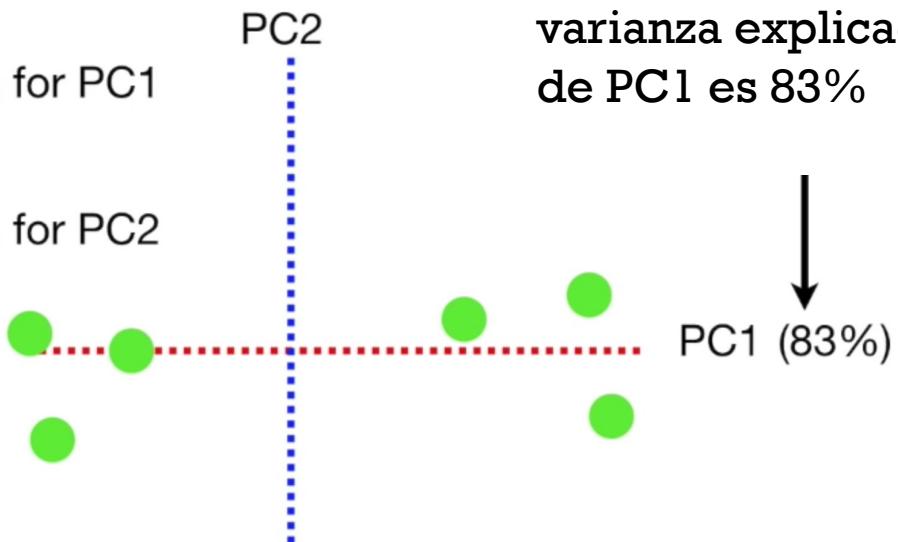
# COMPONENTES PRINCIPALES

Si la variación de PC1  
fuese 15 y la de PC2 3

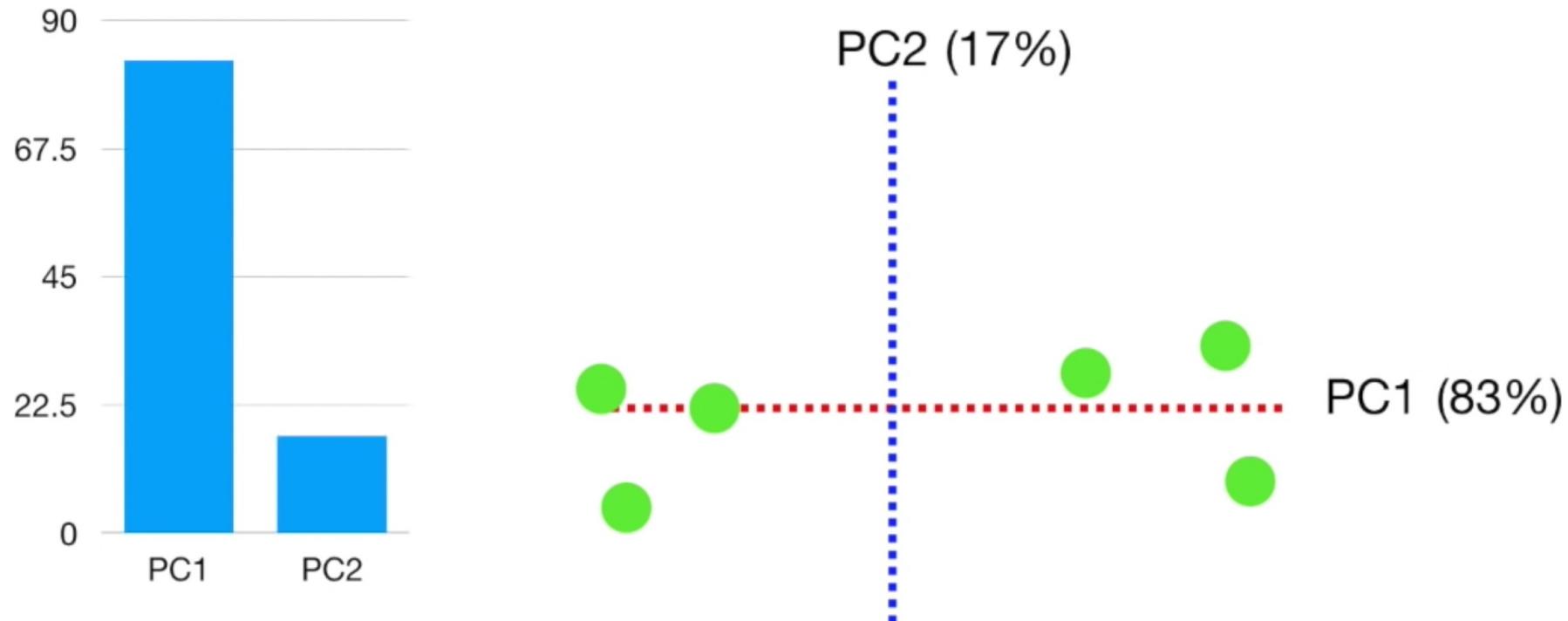
$$\frac{SS(\text{distances for PC1})}{n - 1} = \text{Variation for PC1}$$

$$\frac{SS(\text{distances for PC2})}{n - 1} = \text{Variation for PC2}$$

La variación total en  
ambos PCs es 18 y la  
proporción de  
varianza explicada  
de PC1 es 83%



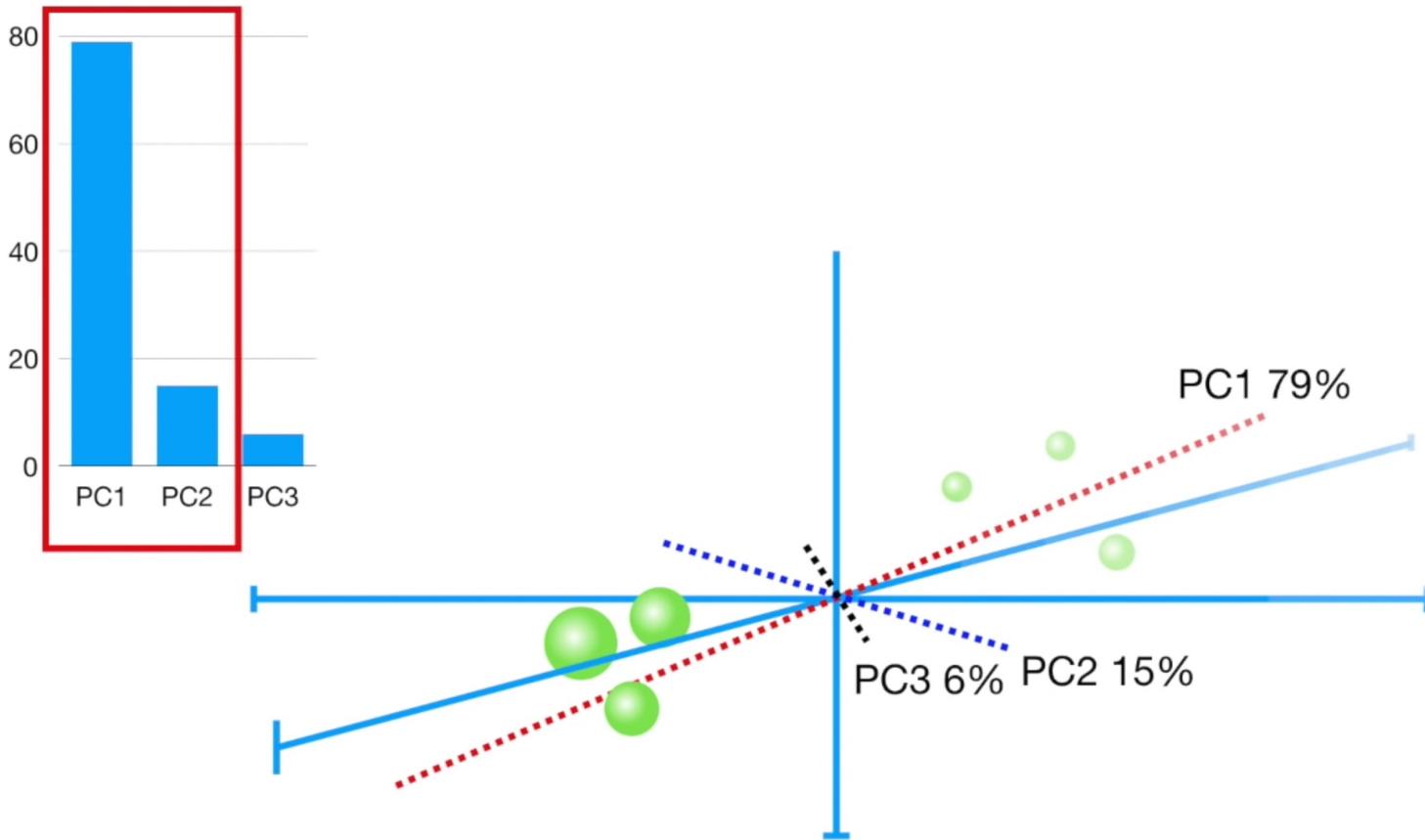
# COMPONENTES PRINCIPALES



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



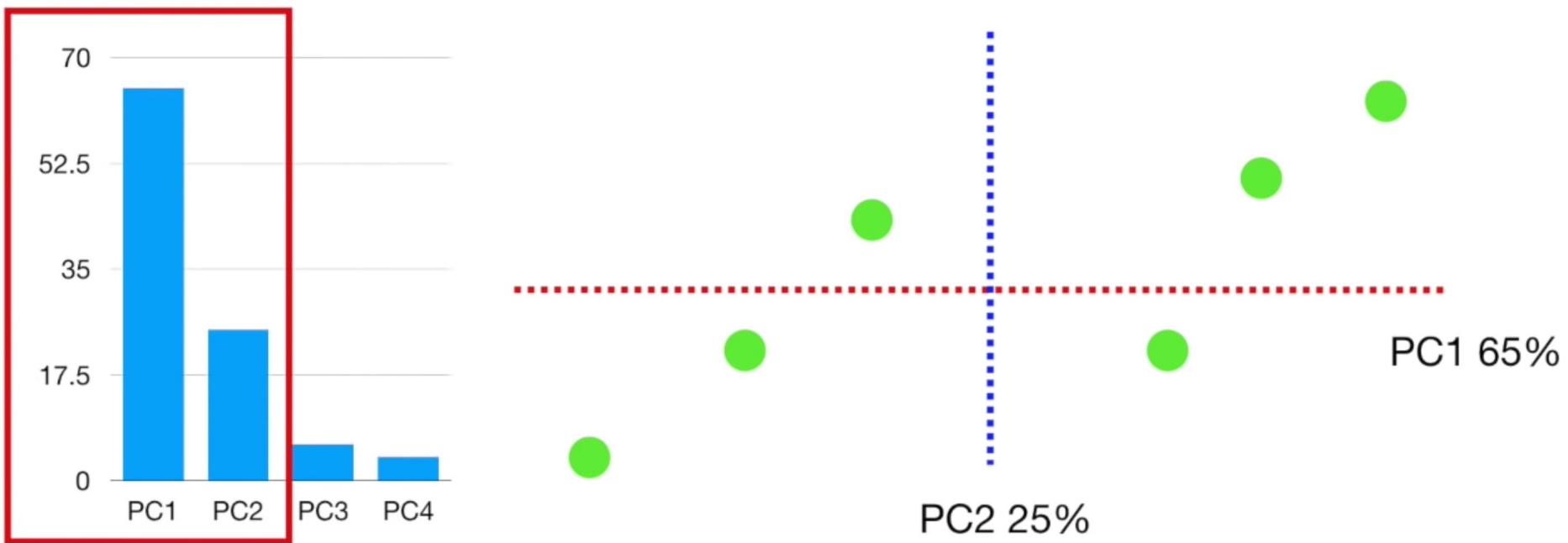
# COMPONENTES PRINCIPALES



<https://www.youtube.com/watch?v=FgakZw6K1QQ>



# COMPONENTES PRINCIPALES



# COMPONENTES PRINCIPALES

## Consideraciones

- La varianza de cada uno de los atributos (dimensiones) originales depende de su escala, por lo que se debe **normalizar** los datos originales
- El número de dimensiones originales puede ser superior al número de instancias del dataset, pero limitaría el número de PCs al número de instancias - 1
- Puede que la varianza esté bien distribuida en los atributos originales, por lo que aplicar PCA no tendría efecto
- El considerar solo los componentes principales más importantes permite ignorar el posible ruido que puedan tener los datos, enfocándose solo en la información más importante. A partir de una transformación inversa del espacio de los PCs hacia el espacio original podemos entonces **filtrar el ruido**.

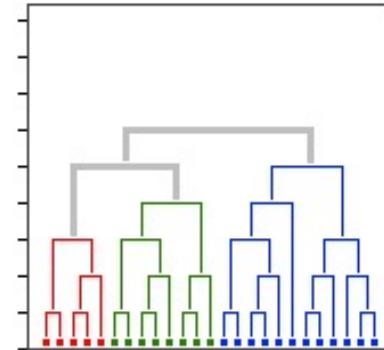
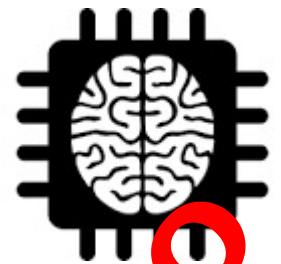


# TALLER: COMPONENTES PRINCIPALES

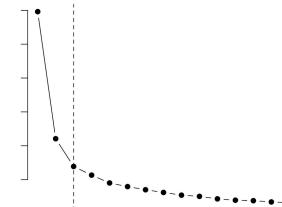
1. Realizar el taller de clustering de clientes de supermercado 10-SUPERMERCADOS-PCA-STUD.html con la parte dedicada a la reducción de dimensionalidad a partir de PCA
2. Continuar el taller de clustering de clientes que desertan aplicando reducción de dimensionalidad a partir de PCA



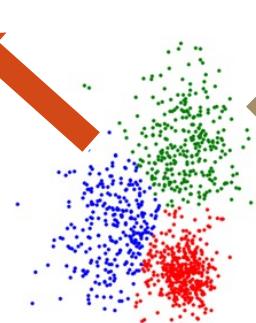
# AGENDA



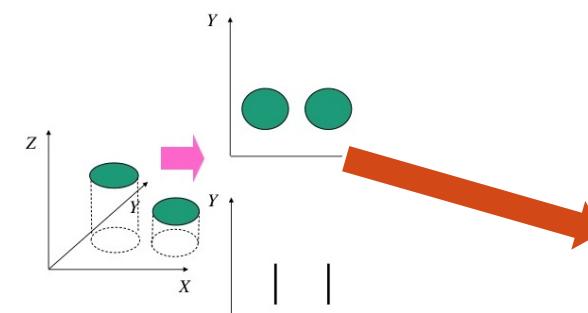
Clustering jerárquico



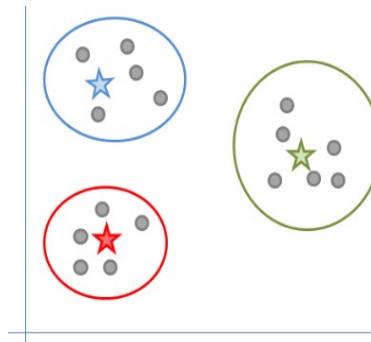
Evaluación del clustering



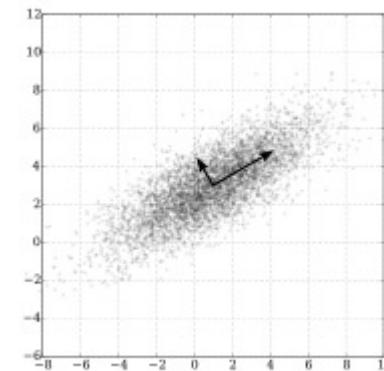
Clustering



Reducción de dimensionalidad



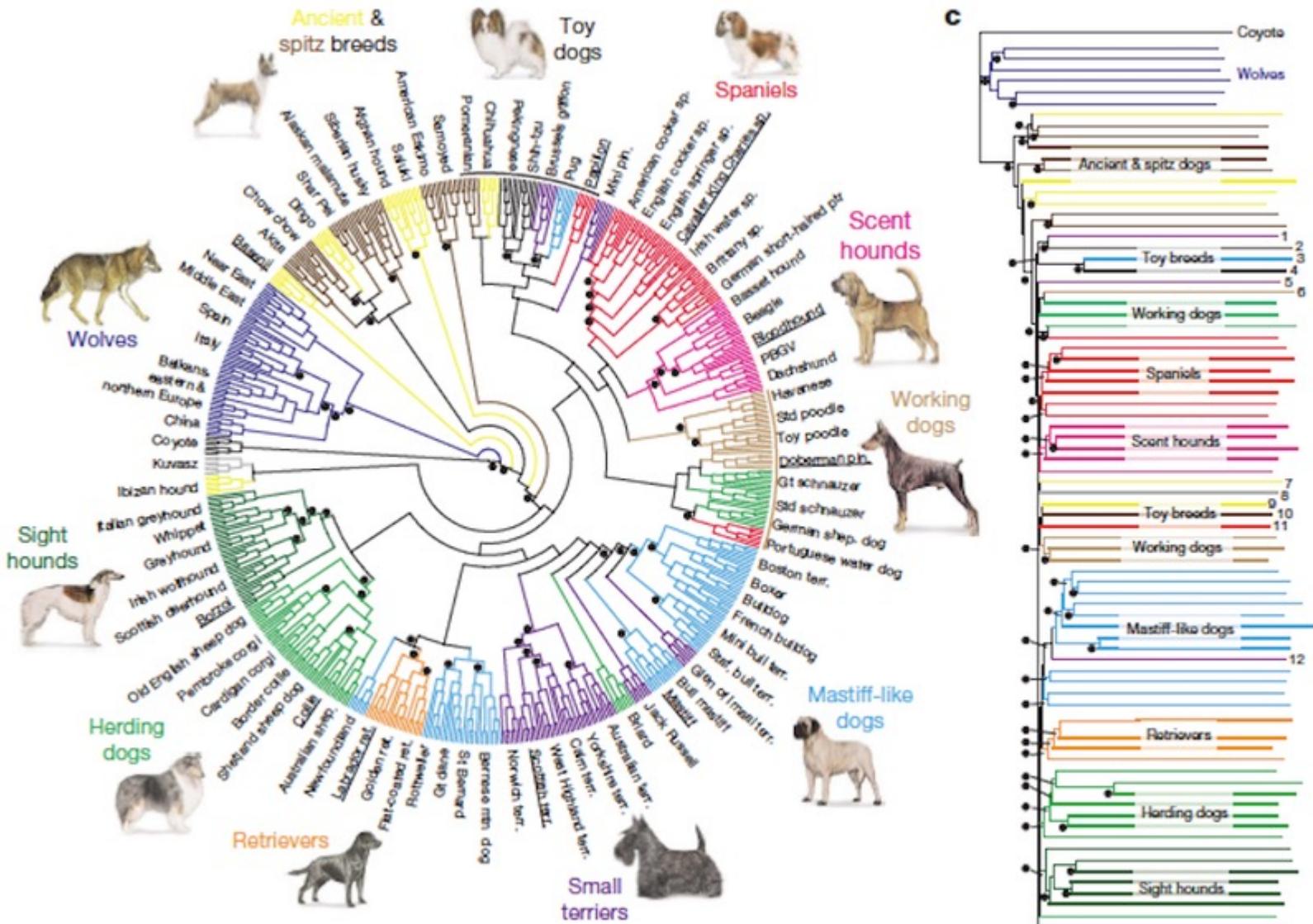
K-means



Componentes principales

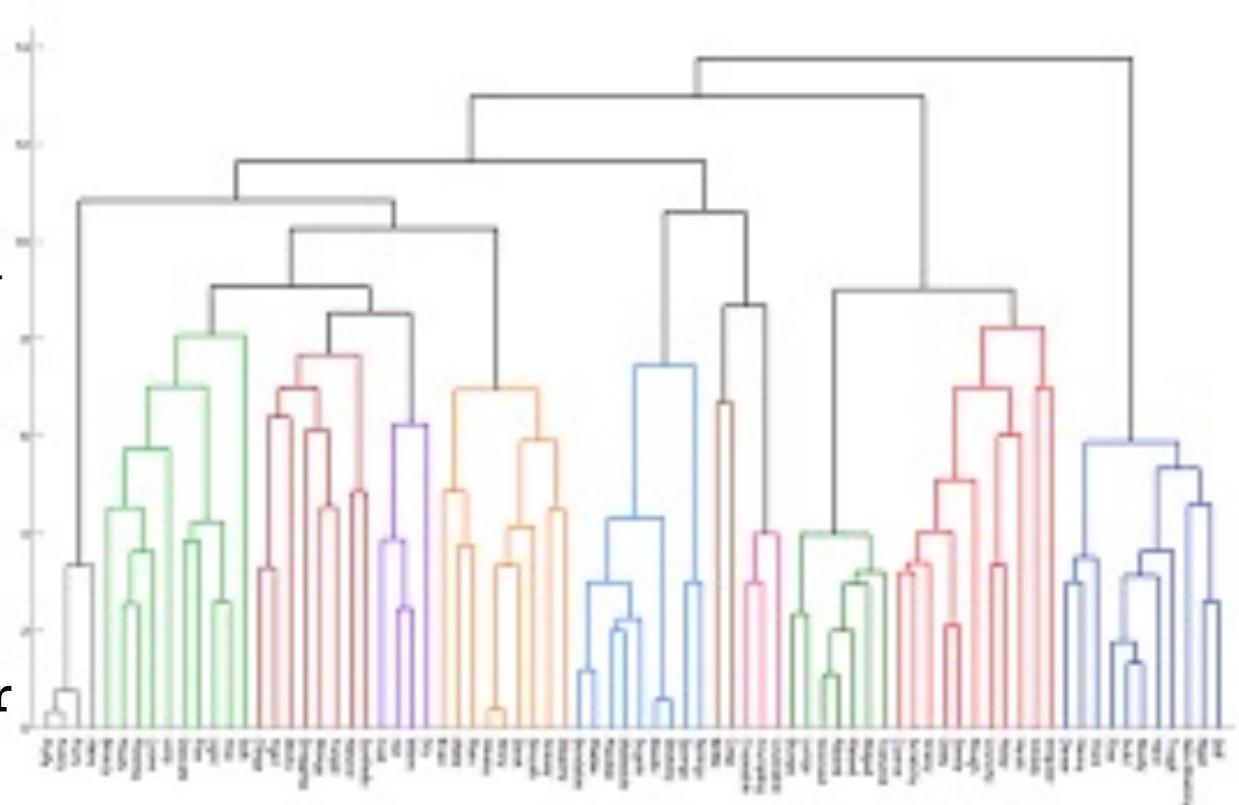


# CLUSTERING JERÁRQUICO



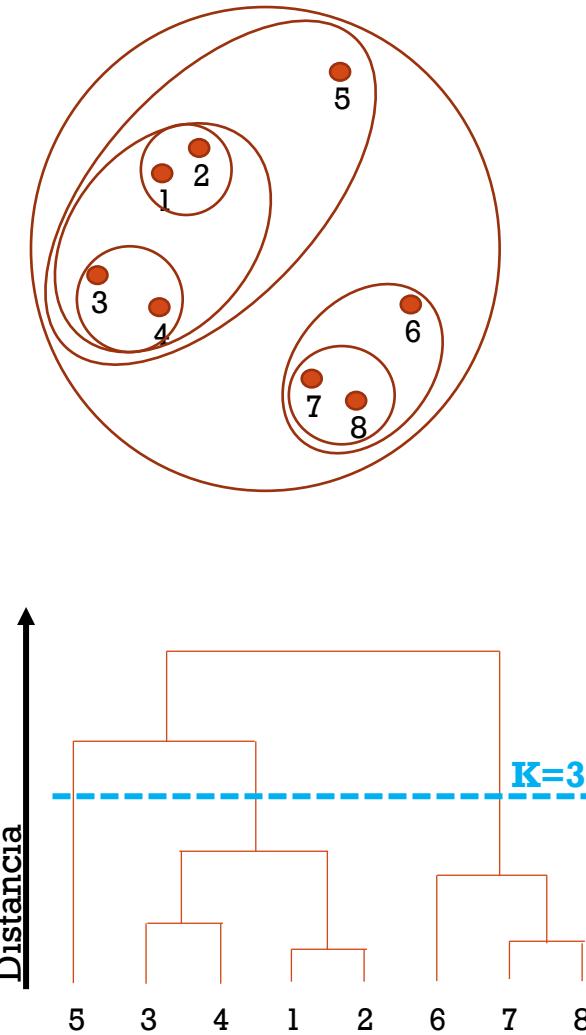
# CLUSTERING JERÁRQUICO

- Aproximación **bottom-up**
- Produce como resultado un **dendrograma**
  - Basado en las **distancias** entre instancias y entre clusters
  - Determina todas las segmentaciones posibles, permitiendo su visualización
- No se necesita repetir el proceso para diferentes valores de **K**
- Las instancias **excepcionales** pueden ser rápidamente identificadas



# CLUSTERING JERÁRQUICO

- Algoritmo (iterativo):
  1. Al inicio cada instancia es un cluster ( $n$  clusters)
  2. Se identifica el par de clusters más cercanos y se fusionan ( $n-1$  clusters)
  3. Se repite el paso anterior hasta que queda un solo cluster con todas las instancias
  4. Se escoge un punto de corte
- Los clusters se pueden organizar en forma de **dendrograma**
- Es necesario definir como **fusionar** clusters y la **distancia** a utilizar



# CLUSTERING JERÁRQUICO

- **Fusión entre clusters** basadas en el cómputo de las distancias entre todos los pares de puntos de cada cluster:

- **Single linkage:**

- Distancia mínima entre dos puntos de los dos clusters.
    - Resultan clusters formados por “cadenas” de puntos, usualmente con fusiones consecutivas entre un cluster y un punto cercano
    - Sensible al ruido y a las excepciones

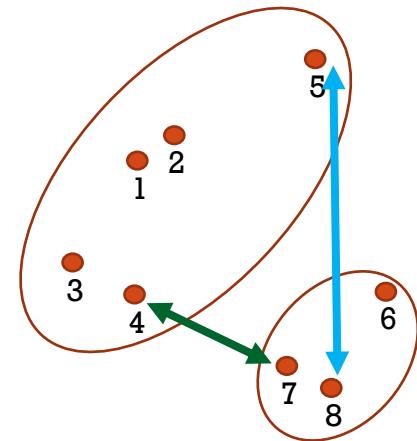
- **Complete linkage:**

- Distancia máxima entre dos puntos de los dos clusters.
    - Tiende hacia clusters esféricos con diámetros consistentes

- **Average linkage:**

- Promedio de las distancias entre todos los pares de puntos
    - Punto intermedio entre single y complete linkage
    - Menos afectado por las excepciones

→ Complete y average se prefieren sobre single linkage



# CLUSTERING JERÁRQUICO

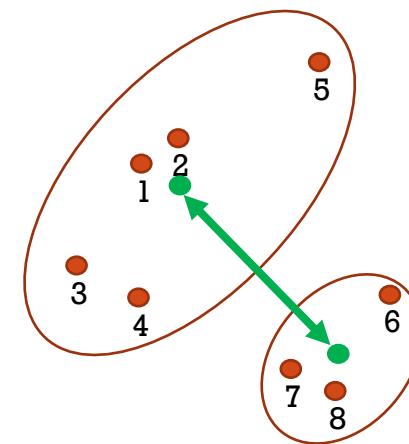
- Otros tipos de **fusión entre clusters**

- **Centroide:**

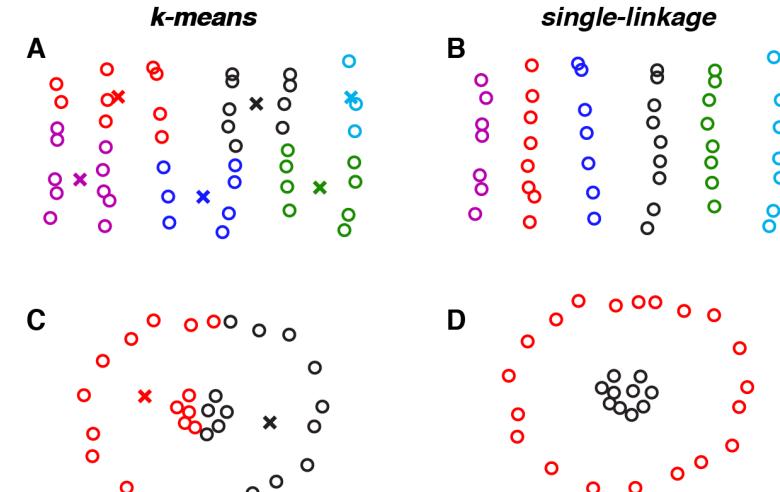
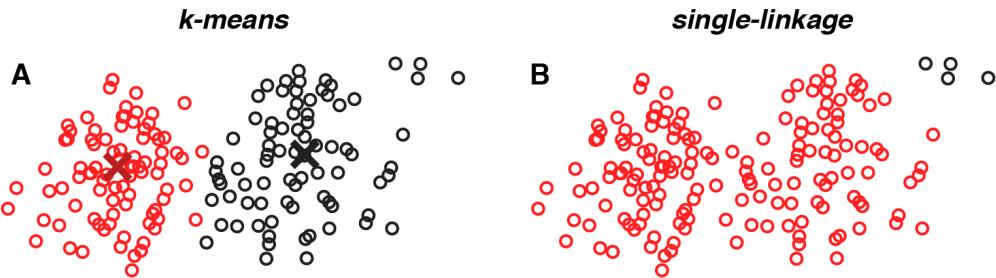
- Distancia entre los centroides de los clusters
    - Sufre de *inversiones*, cuando el punto de fusión de dos clusters en el dendrograma es inferior al de alguno de los clusters fusionados

- **Ward:** Para cada fusión se analiza el cambio en la varianza

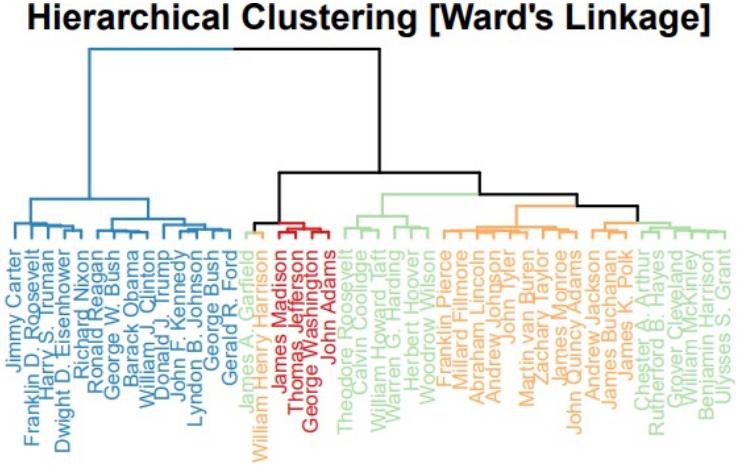
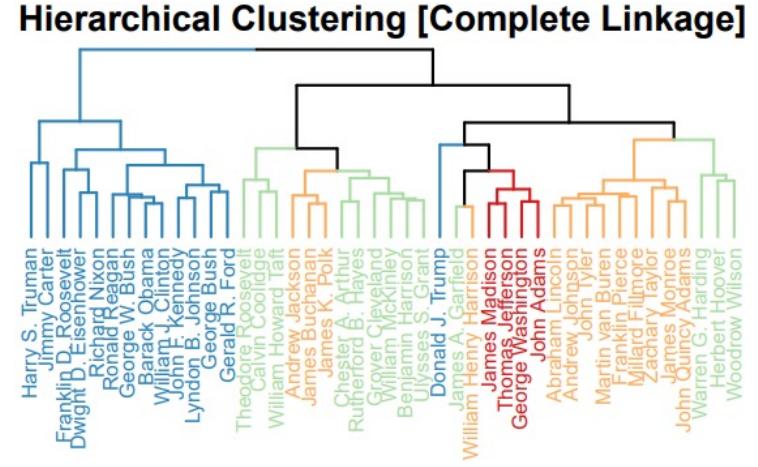
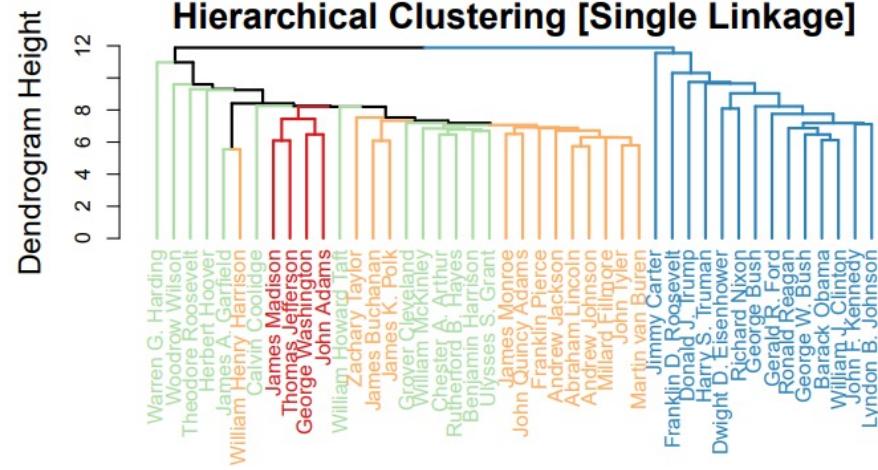
- Con cada fusión, la varianza global del conjunto de clusters aumenta
    - Se escoge la fusión cuyo aumento de varianza es mínimo



# CLUSTERING JERÁRQUICO



<http://alexhwilliams.info/itsneuronalblog/2015/09/11/clustering1/>



<https://arxiv.org/pdf/1901.01477.pdf>



# CLUSTERING JERÁRQUICO

## ■ Consideraciones

- La pertenencia de las instancias a los clusters es absoluta
- Requiere poder de cálculo computacional grande
- Una vez una fusión se decide, no hay vuelta atrás
- Dependiendo de la distancia utilizada y al tipo de fusión:
  - Sensible al ruido y a excepciones
  - Dificulta gestionar clusters de tamaños diferentes o no convexos
  - Puede llegar a particionar clusters grandes
- Influencia de las unidades de los atributos utilizados → estandarización
- Qué punto de corte ( $k$ ) escoger?



# **TALLER: CLUSTERING JERÁRQUICO**

Realizar el taller 10-SYNTH- HClust-STUD.html con datos sintéticos

