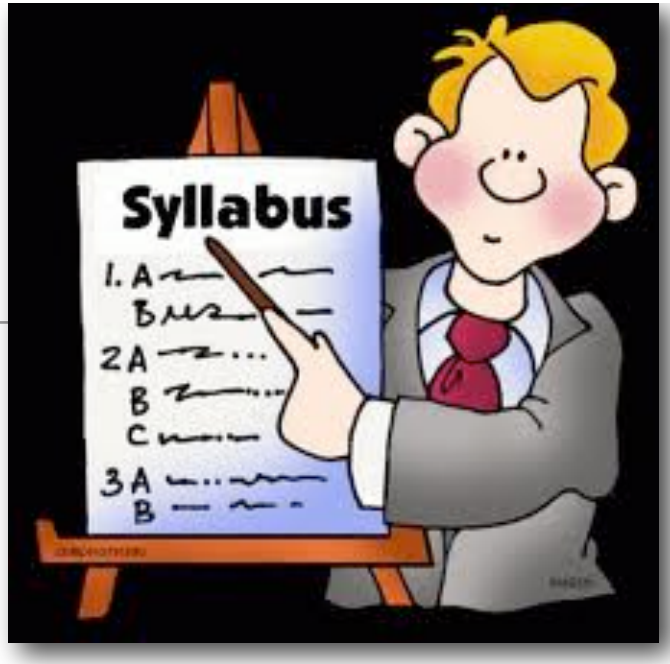
Machine Learning:
Clusters

Sebastián Moreno
Universidad Adolfo Ibañez

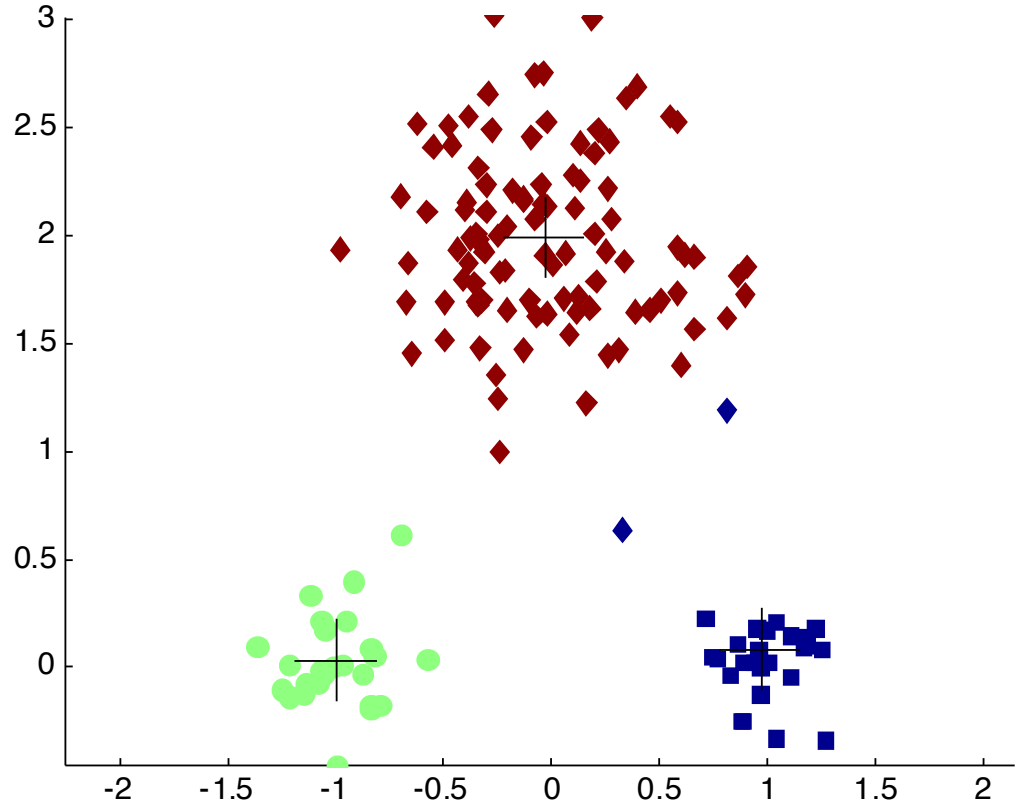
Contenidos

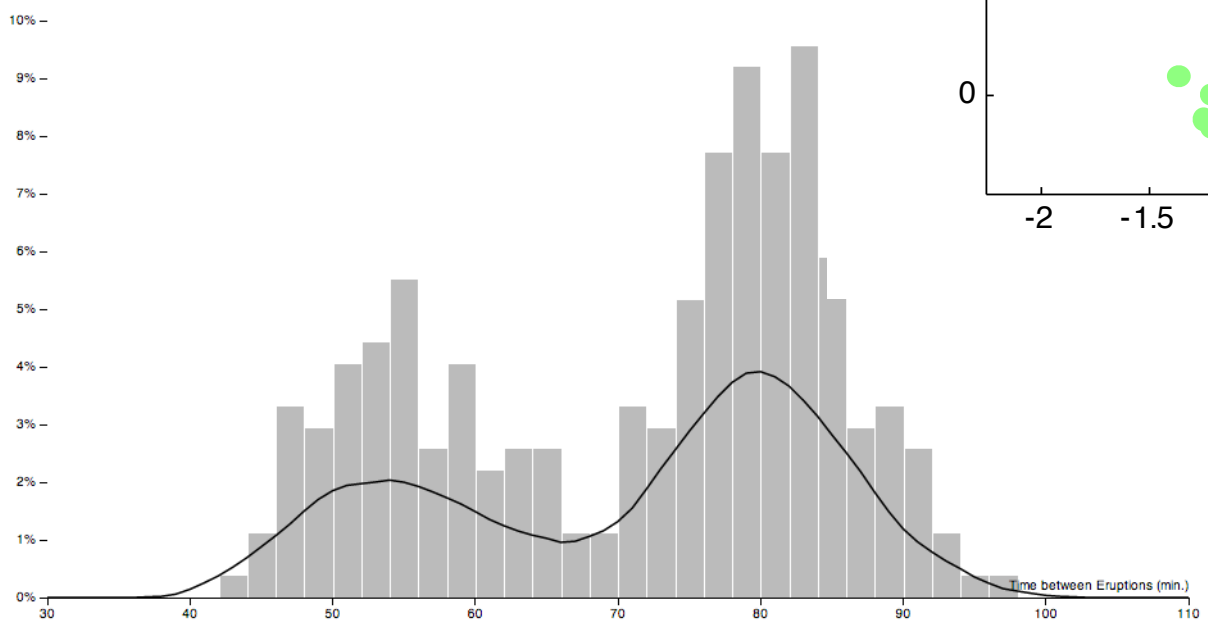


- Introducción
- Limpieza de datos / Feature Engineering
- **Clustering**
  - Introducción
  - Métodos de partición
  - Métodos jerárquicos
  - Métodos difusos
  - Evaluación

Modelos descriptivos

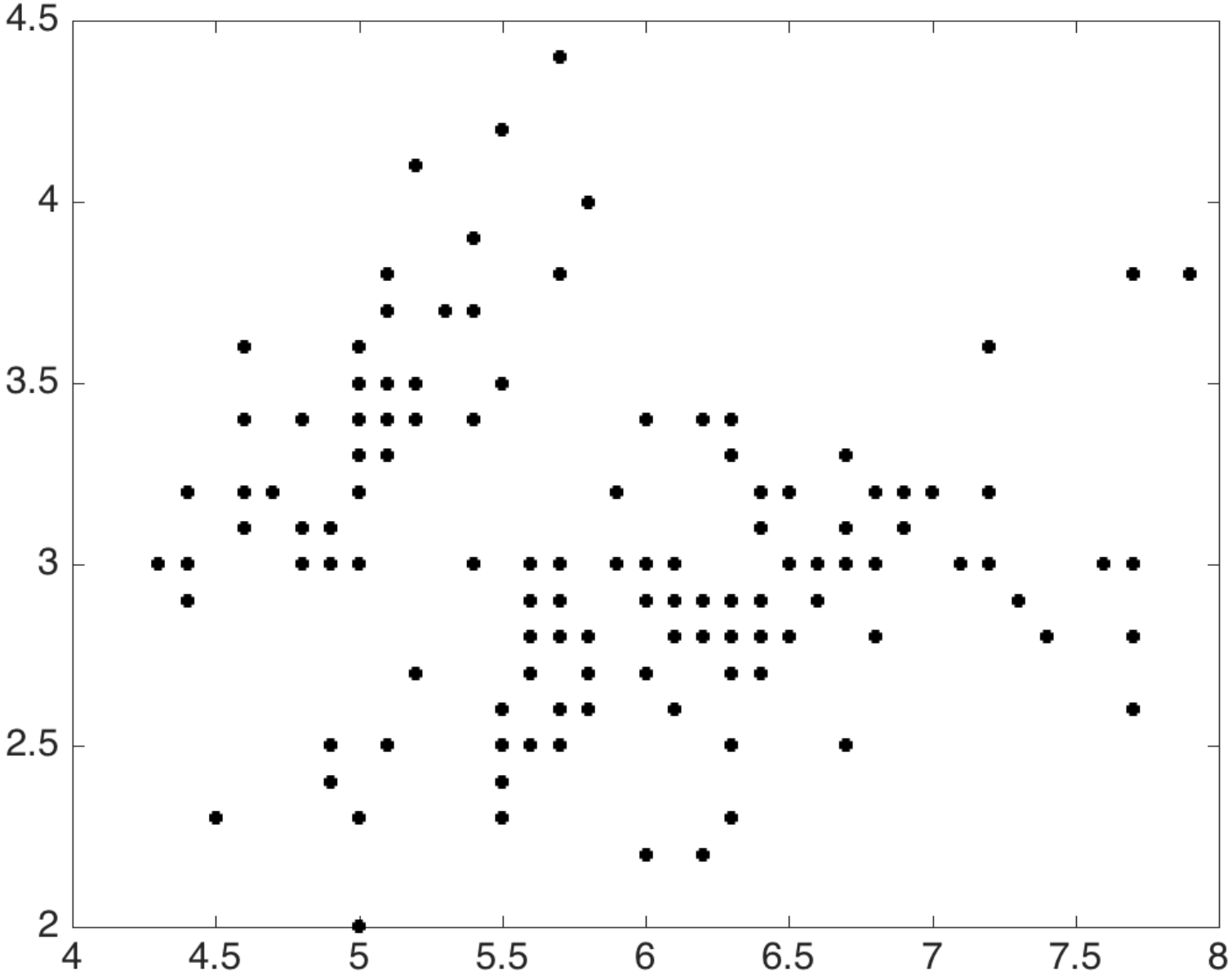
- **Objetivo:** Resumir los datos o replicar el proceso que los genera.
- **Técnicas:** Estimación de densidad, análisis de clusters.





Machine Learning
Clustering
Introducción

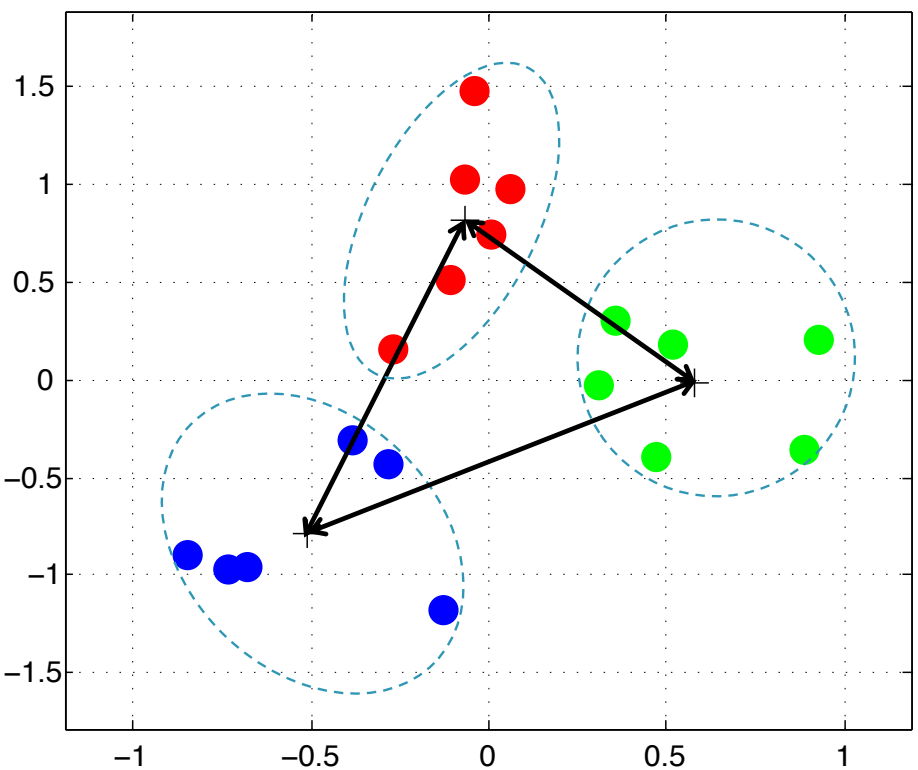
Clustering, ¿cuántos clusters/grupos de datos se pueden observar?



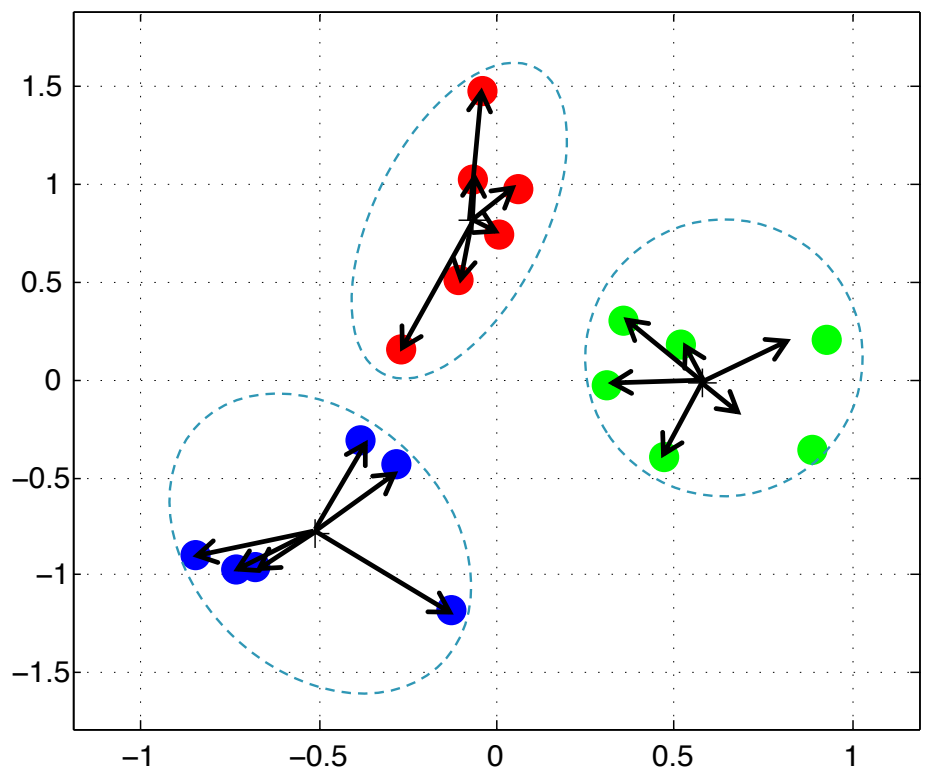
Modelo descriptivo: clustering

- **Clustering:** Encontrar grupos de objetos tal que la similitud dentro de los grupos es alta, mientras que la similitud entre grupos es baja.

Inter-grupo
Between distance



Intra-grupo
Within Distance

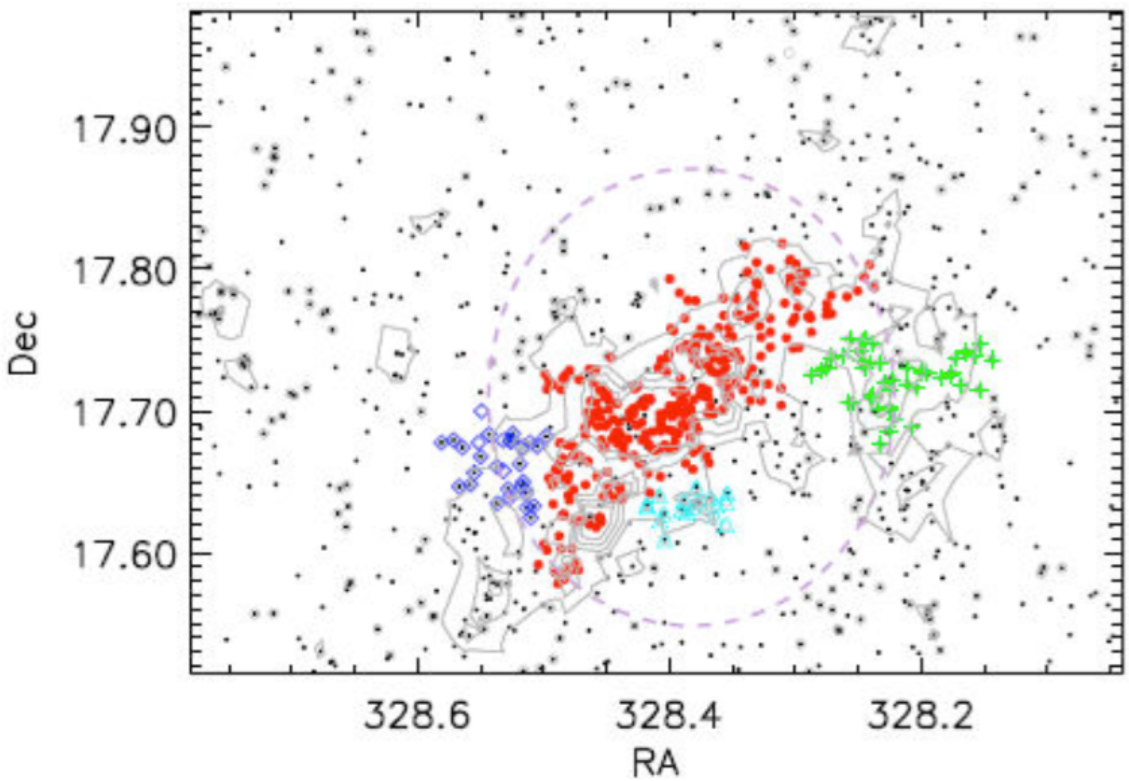


Modelo descriptivo: clustering

- **Clustering:** Encontrar grupos de objetos tal que la similitud dentro de los grupos es alta, mientras que la similitud entre grupos es baja.

- La medida de distancia o similitud es crucial para clustering.

- Difícil evaluar el éxito.
  -¿clusters “interesantes”? Son difíciles de evaluar.
  -¿clusters “similares”? Esto depende de la medida de distancia utilizada.



Modelo descriptivo: clustering

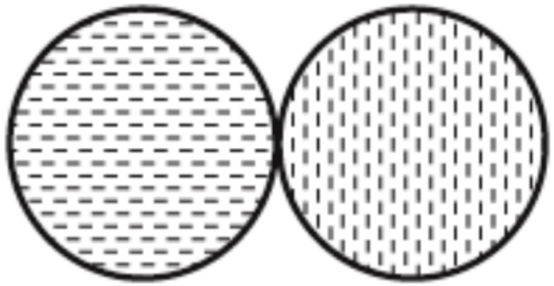
- La noción de cluster es ambigua

¿Cuántos clusters ven?

Modelo descriptivo: clustering

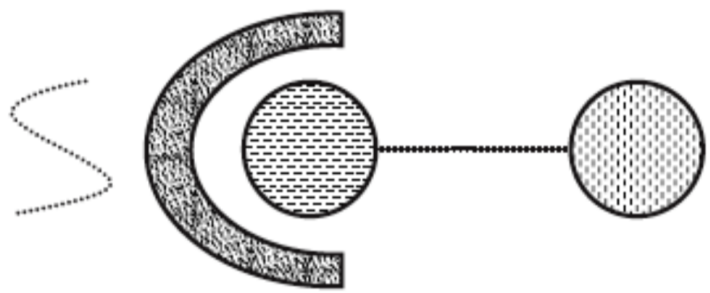
- La noción de cluster es ambigua
- **Clusters bien separados**: Cada dato es más cercano a **todos** los puntos de su cluster que a cualquier punto de otro cluster.

- **Clusters basados en centros**: Cada dato es más cercano al centro de su cluster que a cualquier centro de otro cluster.

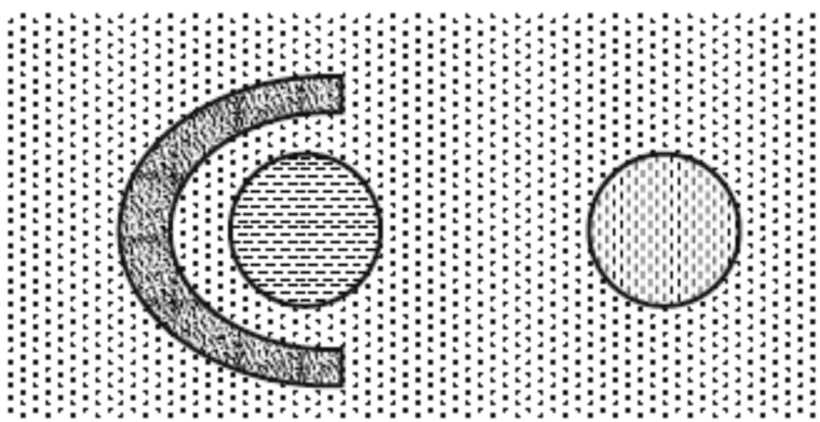


Modelo descriptivo: clustering

- La noción de cluster es ambigua
- **Clusters continuos**: Cada dato es más cercano al menos a un punto de su cluster que a cualquier punto de otro cluster.



- **Cluster basados en densidad**: Los clusters corresponden a regiones de datos con alta densidad separados por regiones de datos con baja densidad.

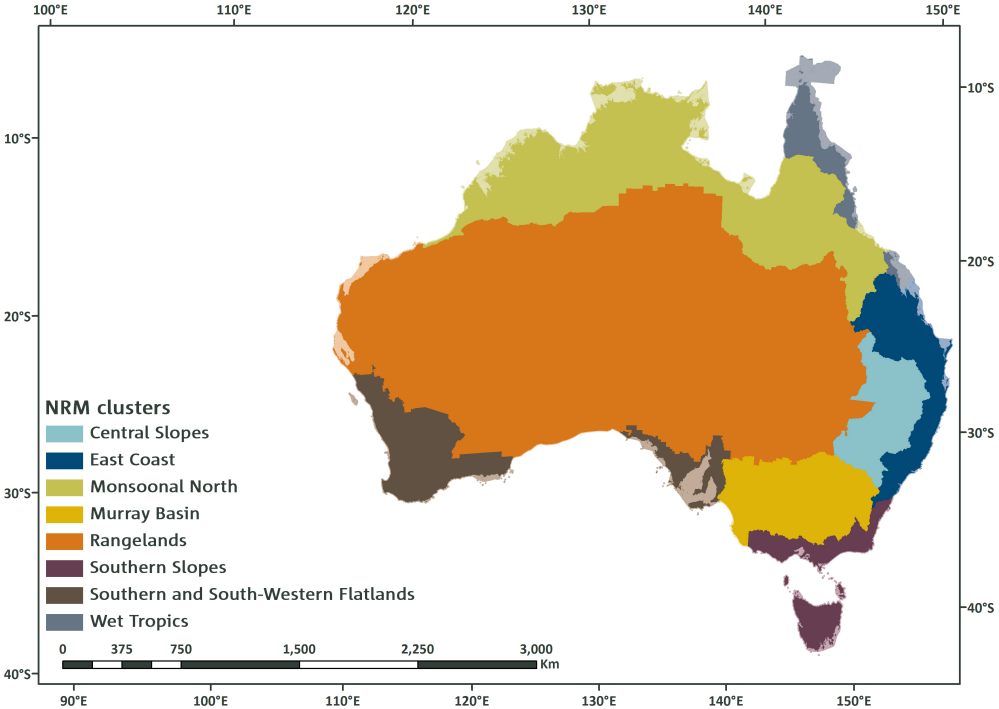


Modelo descriptivo: aplicaciones de clustering

- **Comprender:** Creación de grupos de documentos relacionados para lectura; genes y proteínas con funciones similares; acciones con cambio de precio similares.

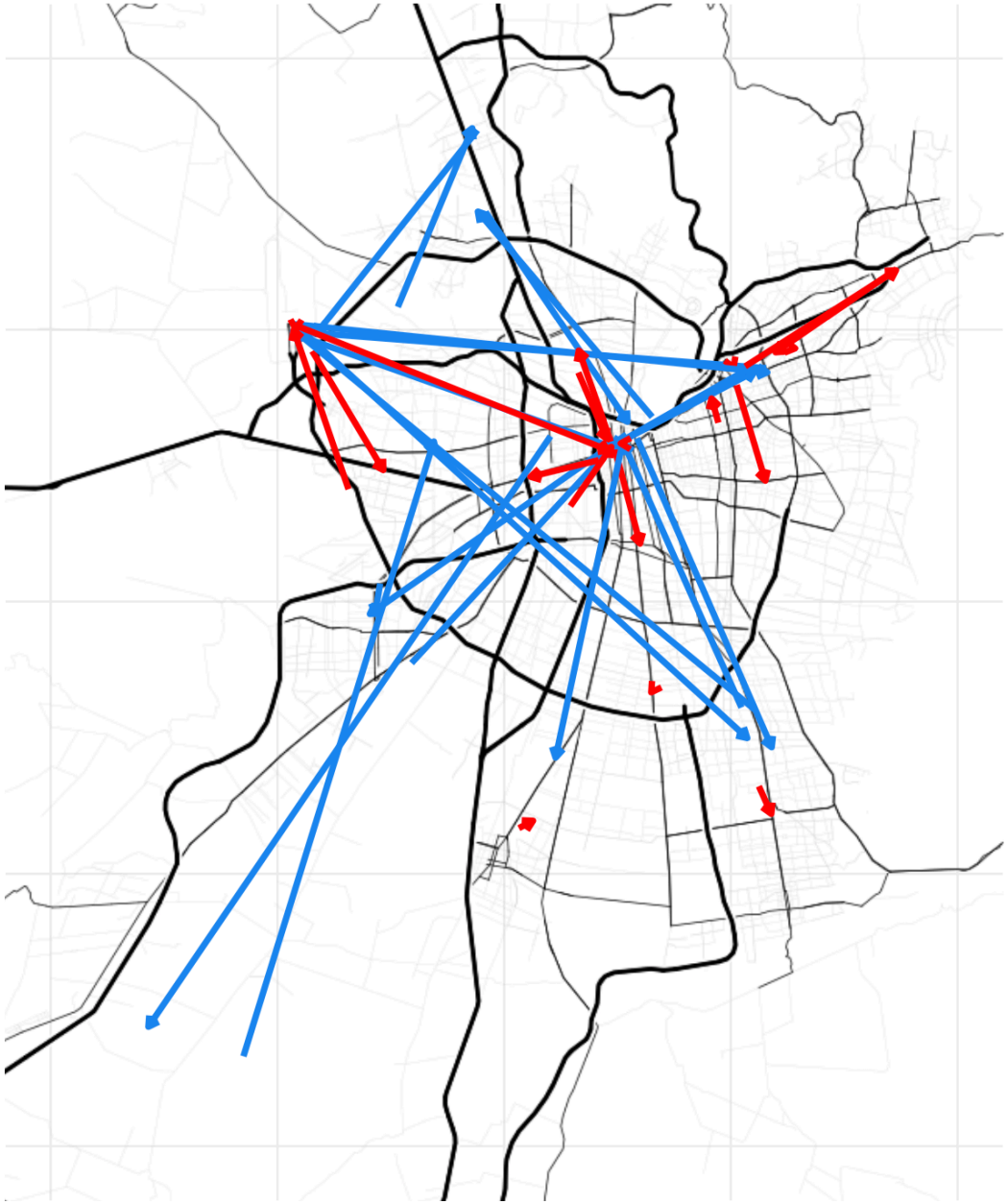
- **Resumir:** Reduce el tamaño de grandes set de datos.

	Discovered Clusters	*Industry Group*
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP



Modelo descriptivo: aplicaciones de clustering

- **Comprender** el comportamiento de radiotaxis (esto corresponde a la tarea 1).



Modelo descriptivo: aplicaciones de clustering

- **Marketing:** buscar grupos de clientes para el desarrollo de marketing dirigido.
- **Geografía:** Identificar areas geográficas similares para entender el desarrollo de la tierra.
- **Planificación urbana:** Agrupar casas en base a su tipo, valor, locación u otras características para identificar “vecindarios”.
- **Diseño de territorios:** Agrupar locales de compra para facilitar la entrega de productos.

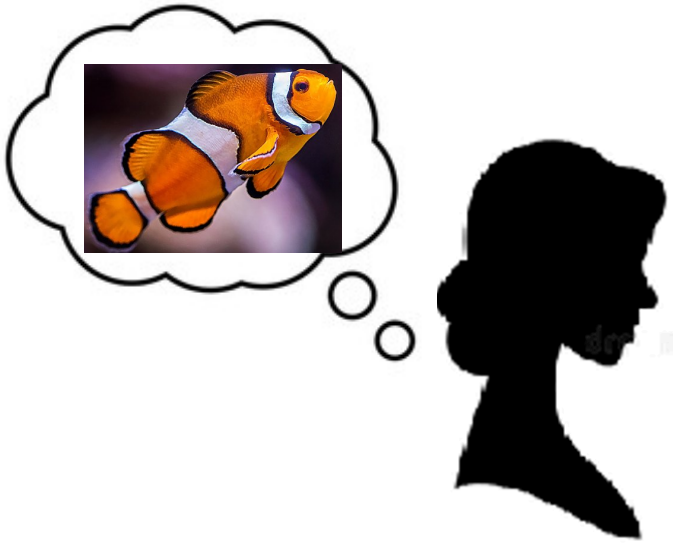
“Language processing differences between blind and sighted individuals and the abstract versus concrete concept difference”

Cognitive Science, WOS-Q2, 2021

E. Canessa, S. Chaigneau, & S. Moreno

Problema

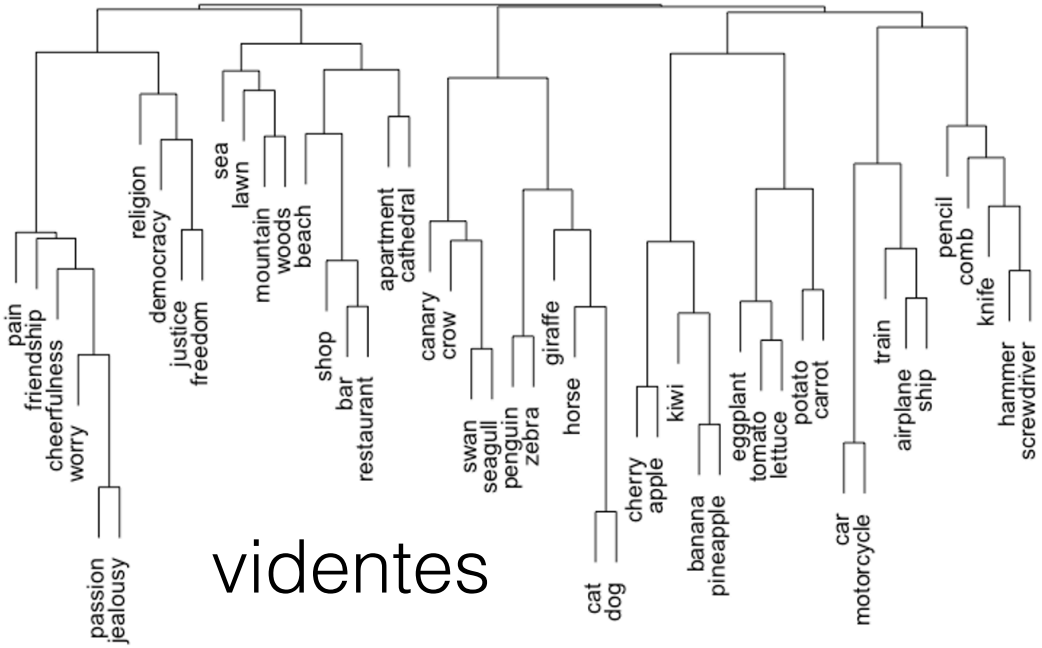
- Dada una palabra, personas la describen con distintas propiedades.
- Ejemplo:
  Perro
  Persona 1 => mejor amigo del hombre, cuadrúpedo, ladra
  Persona 2 => peludo, fiel, comilón, 4 patas, bullicioso
- Dado múltiples conceptos tanto abstractos (ejemplo paz) como concretos (ejemplo perro), ¿qué tan distintos describen estas palabras las personas videntes versus no videntes?



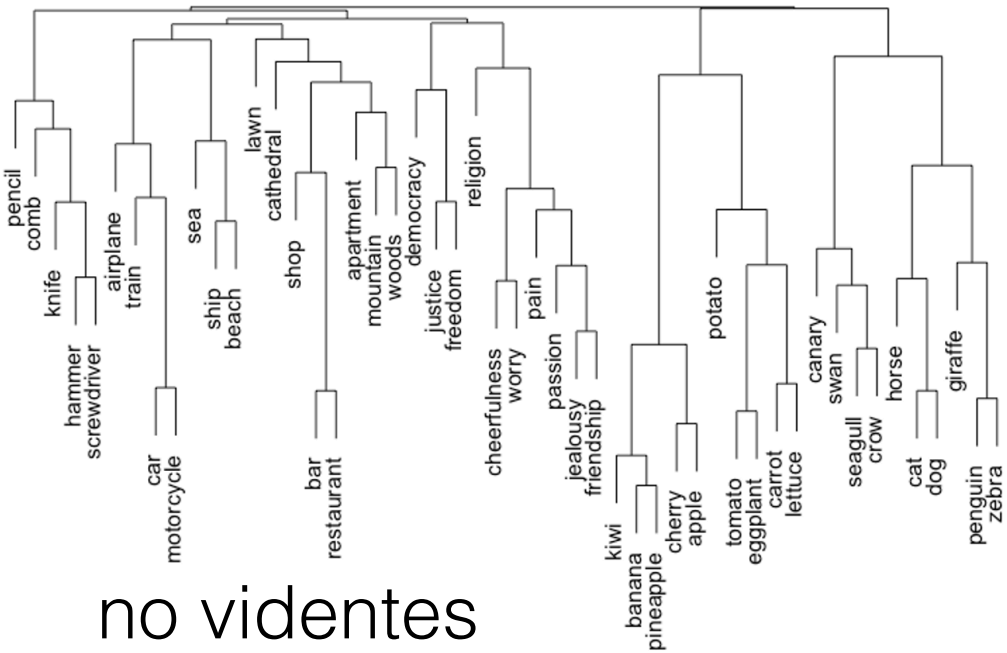
Mi mascota
es tan tierna
y silenciosa

Solución

- Se analizó un universo de 22 no videntes de nacimiento y 26 videntes, y se mencionaron 40 sustantivos concretos y 20 abstractos.
- Se agrupó las propiedades descritas. Sorprendentemente, las agrupaciones no muestran diferencias significativas.
- A pesar de lo esperado, tanto videntes como no videntes, tienen descripciones de propiedades similares de los conceptos.



videntes



no videntes

“A hybrid K-means and integer programming method for commercial territory design: a case study in meat distribution”

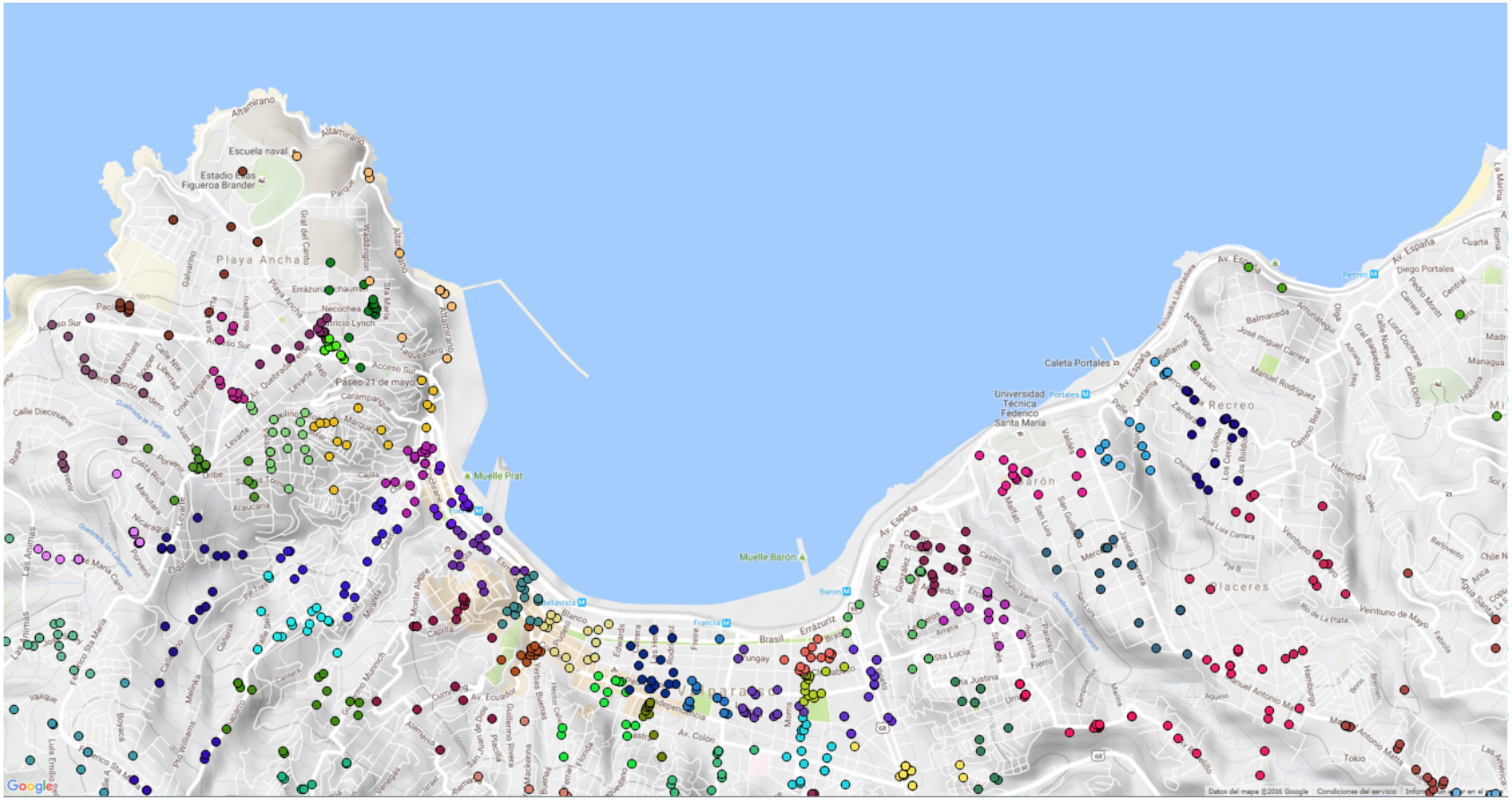
Annals of Operations Research, WOS-Q2, 2020
S. Moreno, J. Pereira, & W. Yushimito

Problema

- Empresa distribuidora de carne tiene 1976 clientes a los cuales tiene que agrupar para destinar camiones repartidores
  - Restricción 1: cada “camión” trabajador no puede superar la jornada laboral de 8 horas.
  - Suposiciones:
    - 35 km/h en ciudad
    - 50 km/h carretera
    - 30 minutos por cliente

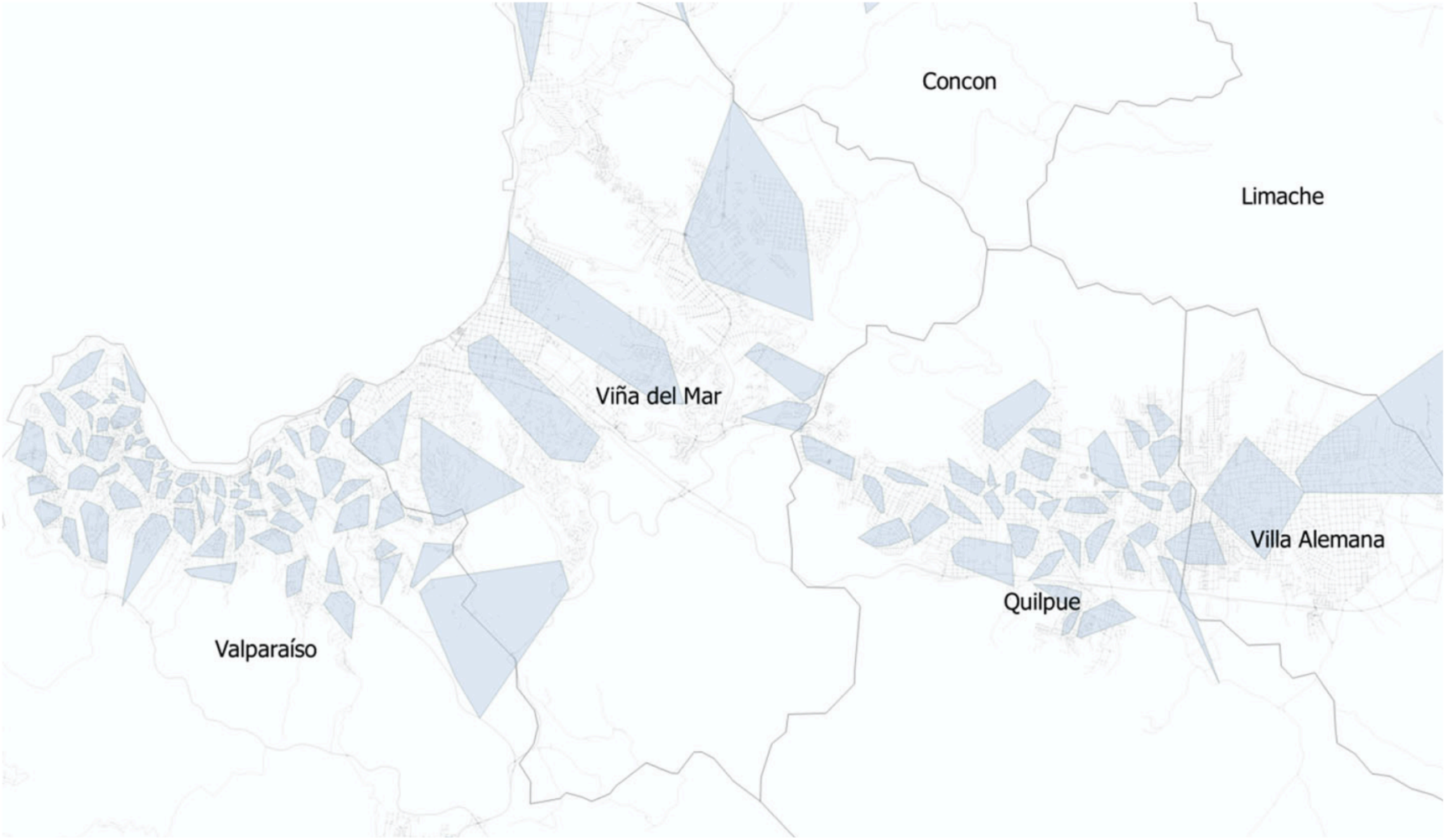
Solución

- Se propone un k-means dinámico, donde los clusters generados se adaptan a las restricciones solicitadas.
- Cluster generados para Viña y Valparaíso, cada cluster es un color.



Solución

- Una vez generados múltiples soluciones, se combinan entre ellas usando programación entera para generar asignaciones finales.

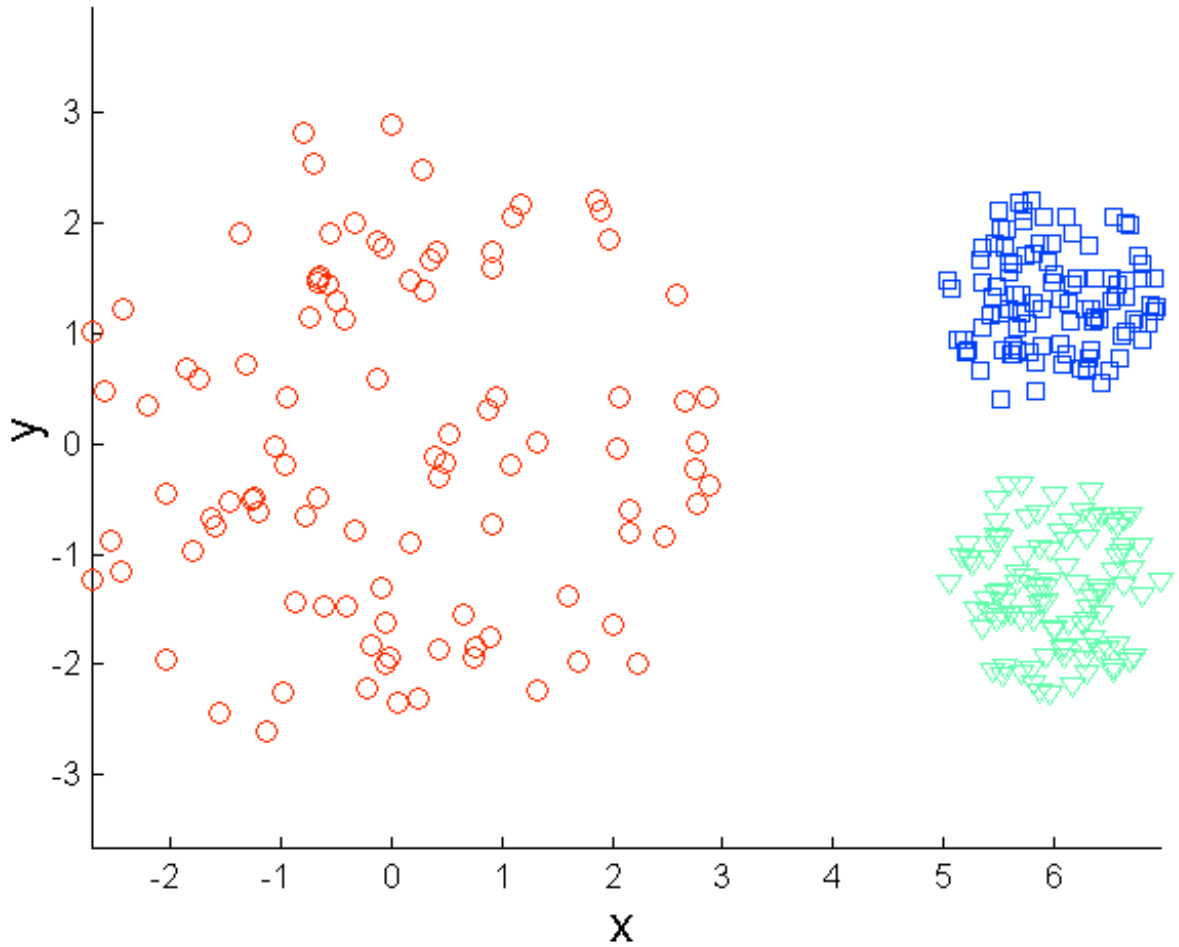


Modelo descriptivo: tipos de clustering

- Existen distintos tipos de métodos de clustering que encuentran cluster con distintas “formas”.
- Hard clustering
  - Métodos de partición
  - Métodos de densidad
  - Métodos jerárquicos (divisivo/aglomerativo)
- Soft clustering
  - Métodos probabilísticos

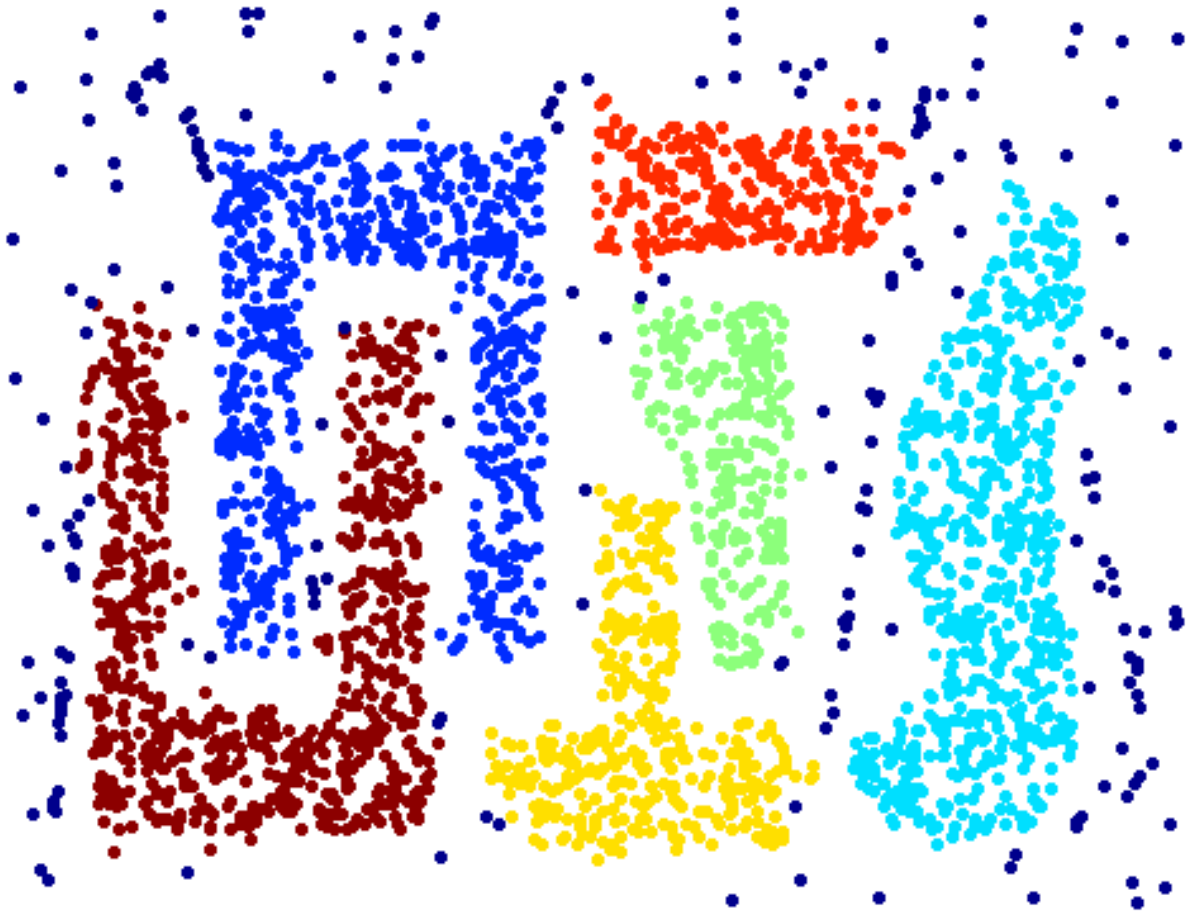
Modelo descriptivo: métodos de partición

- Los datos son separados en K clusters, donde cada punto pertenece en forma **exclusiva** a un solo cluster.



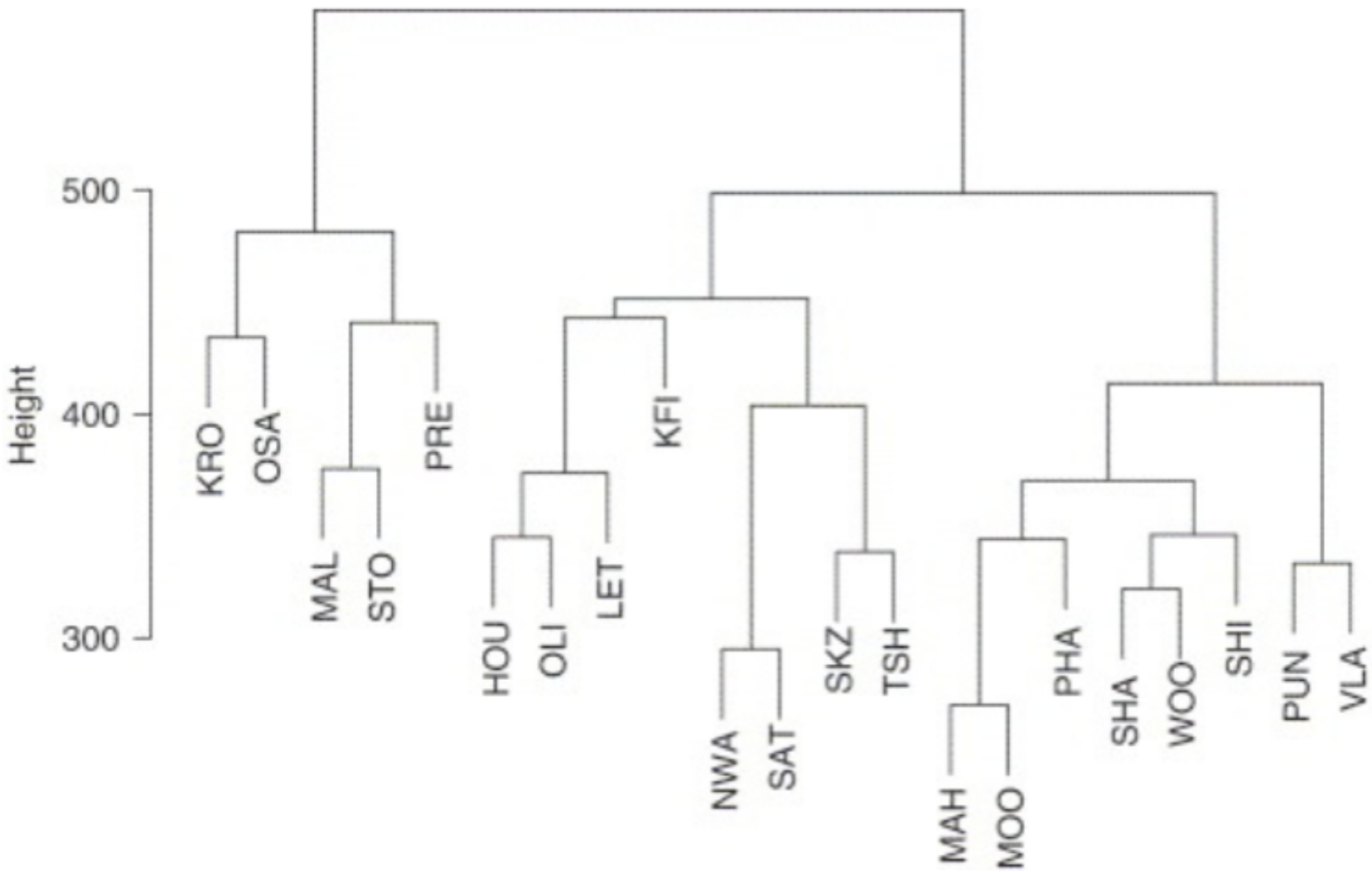
Modelo descriptivo: métodos de densidad

- Identificar regiones de puntos densas en los datos. Estas forman los “bloques originales” para la generación de cluster con formas arbitrarias.



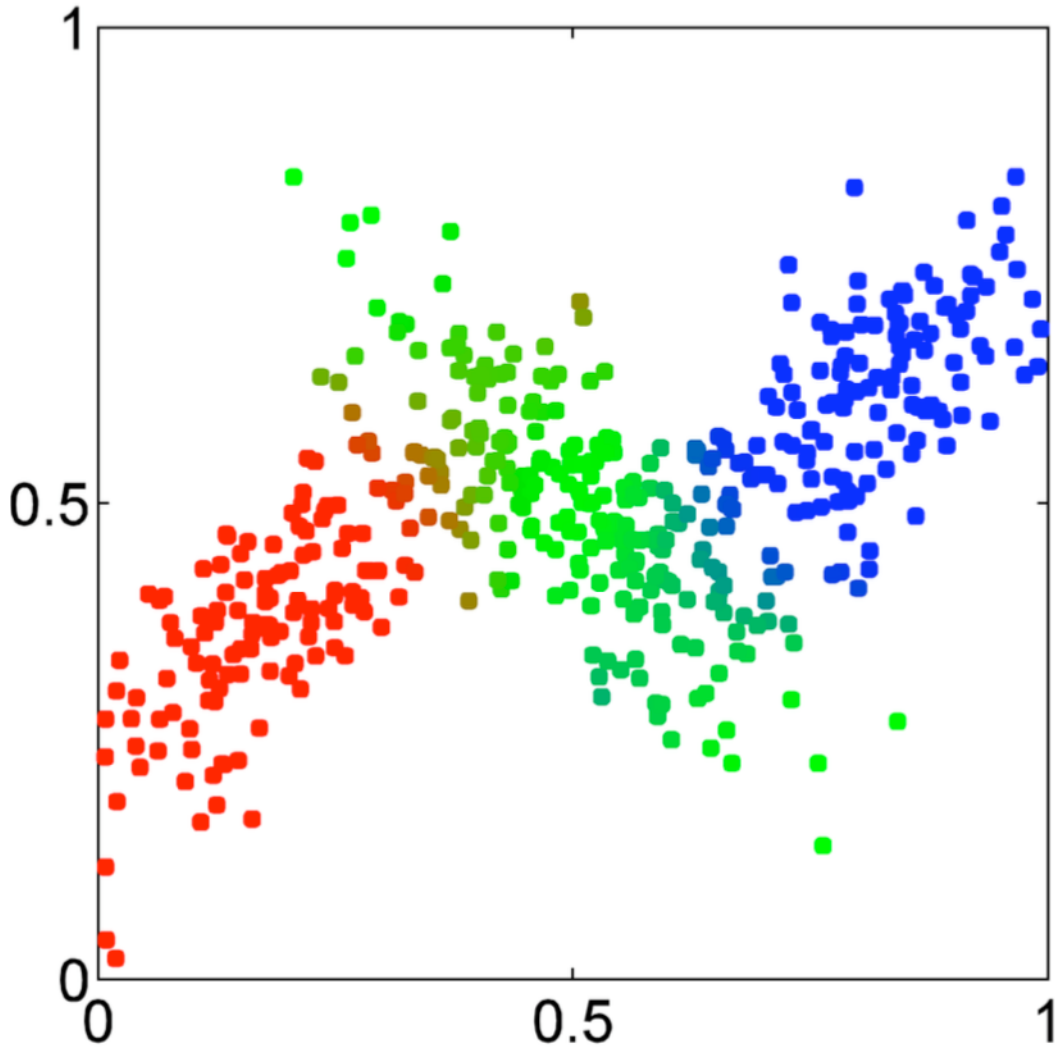
Modelo descriptivo: métodos jerárquicos

- Los datos son separados en una jerarquía de clusters.



Modelo descriptivo: métodos difusos/probabilísticos

- Cada dato tiene una “probabilidad” de pertenecer a un cluster en particular.

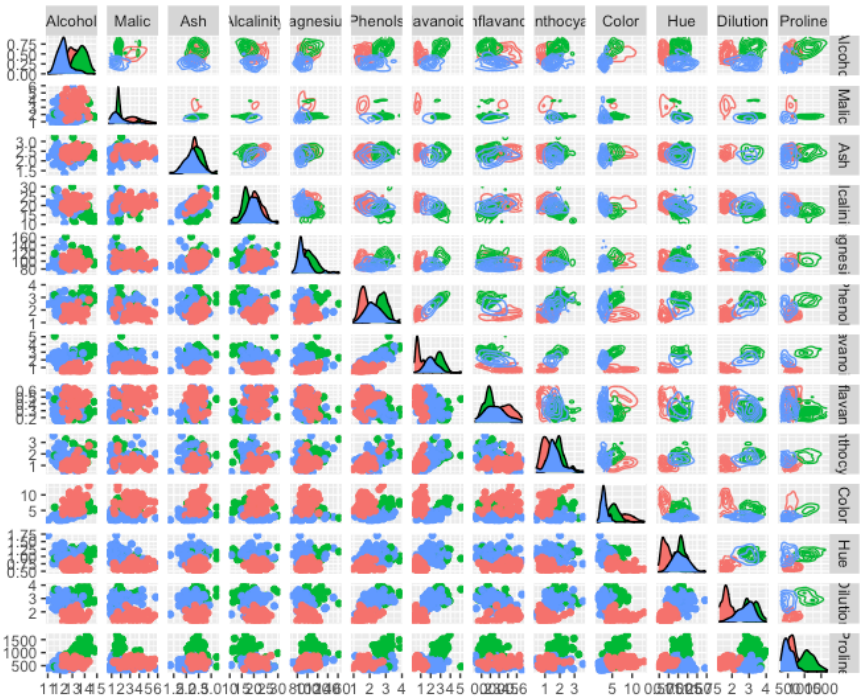


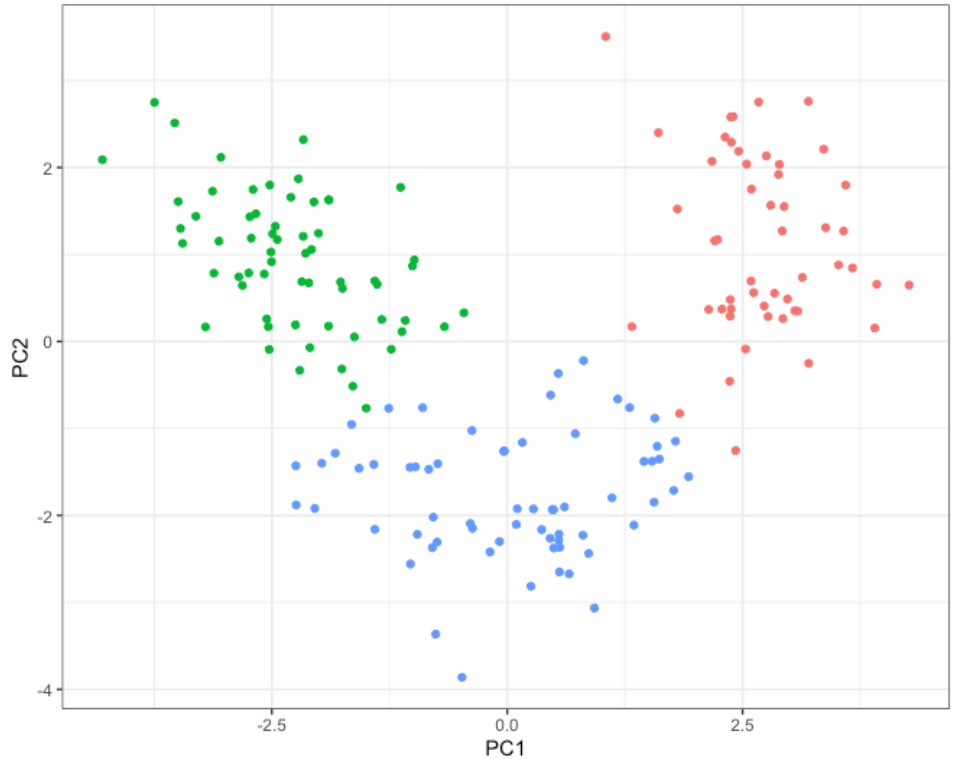
Modelo descriptivo: visualización

- Es prácticamente imposible visualizar los datos clusterizados debido a las altas dimensiones.

- La solución más sencilla es aplicar PCA y quedarse con las dos primeras dimensiones, pero existen otros análisis.

k-means -> 3 clusters



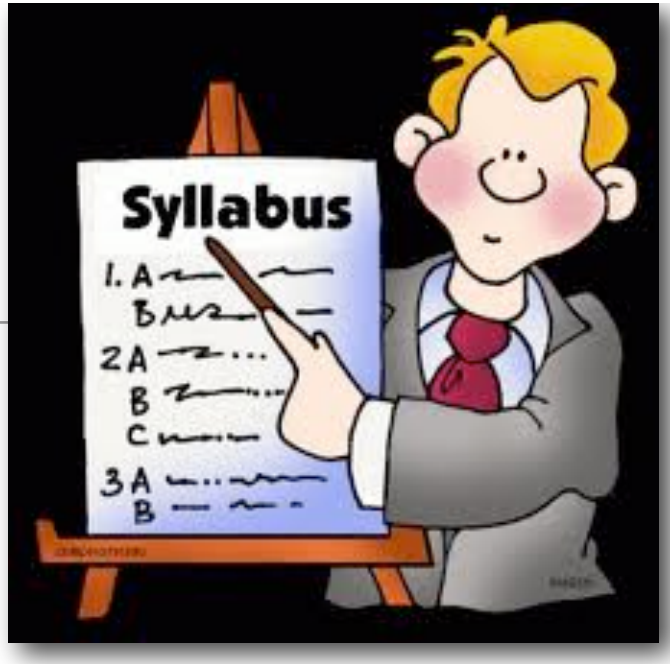


Modelo descriptivo: información extra

- En general, no se realiza selección de variables en los algoritmos de clustering.
- Si se ha demostrado que variables ruidosas pueden afectar a los modelos.
- Al eliminar variables correlacionadas podríamos estar acercando puntos que realmente no son cercanos entre ellos.
- Muchos algoritmos están basados en distancia, por lo cual sufren de la maldición de la dimensionalidad. Para evitar ello, aplique PCA.
    - Atención, esto puede eliminar clusters que sean importantes. A la vez, podría ayudar a tener mejores clusters.

Machine Learning
Clustering
Métodos de partición

Contenidos



- Introducción
- Limpieza de datos / Feature Engineering
- **Clustering**
  - Introducción
  - Métodos de partición
  - Métodos jerárquicos
  - Métodos difusos
  - Evaluación

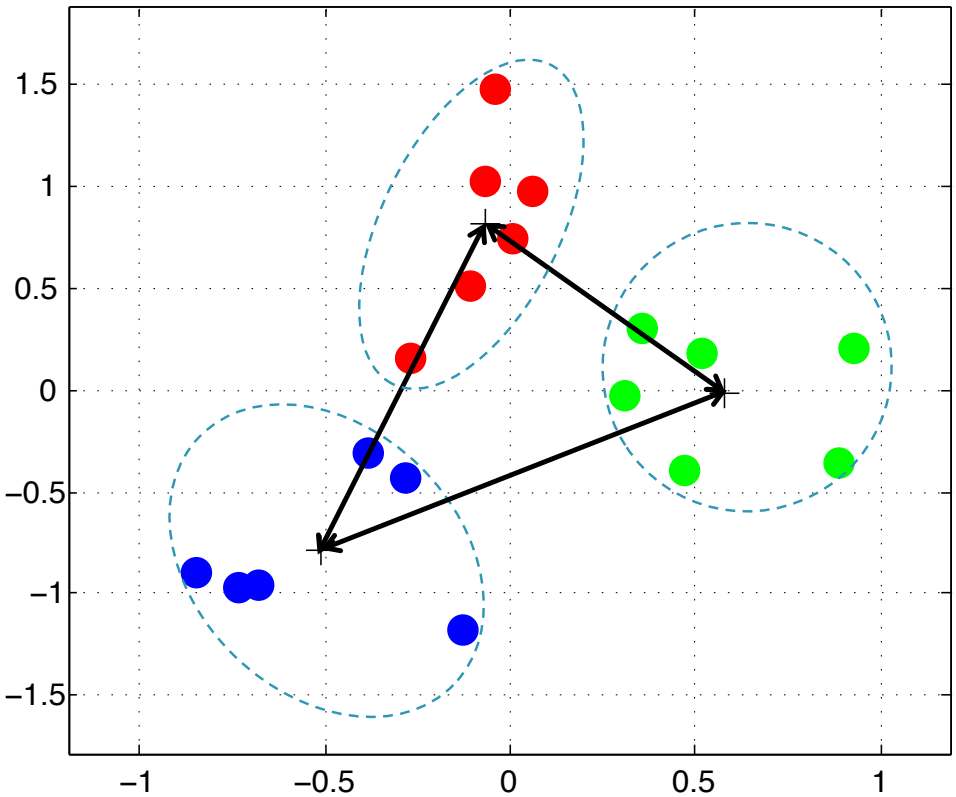
Métodos de partición, aprendizaje

- **Objetivo:**
  Clusters compactos: minimizar la distancia dentro del cluster (within)
  Clusters separados: maximizar la distancia entre clusters (between)

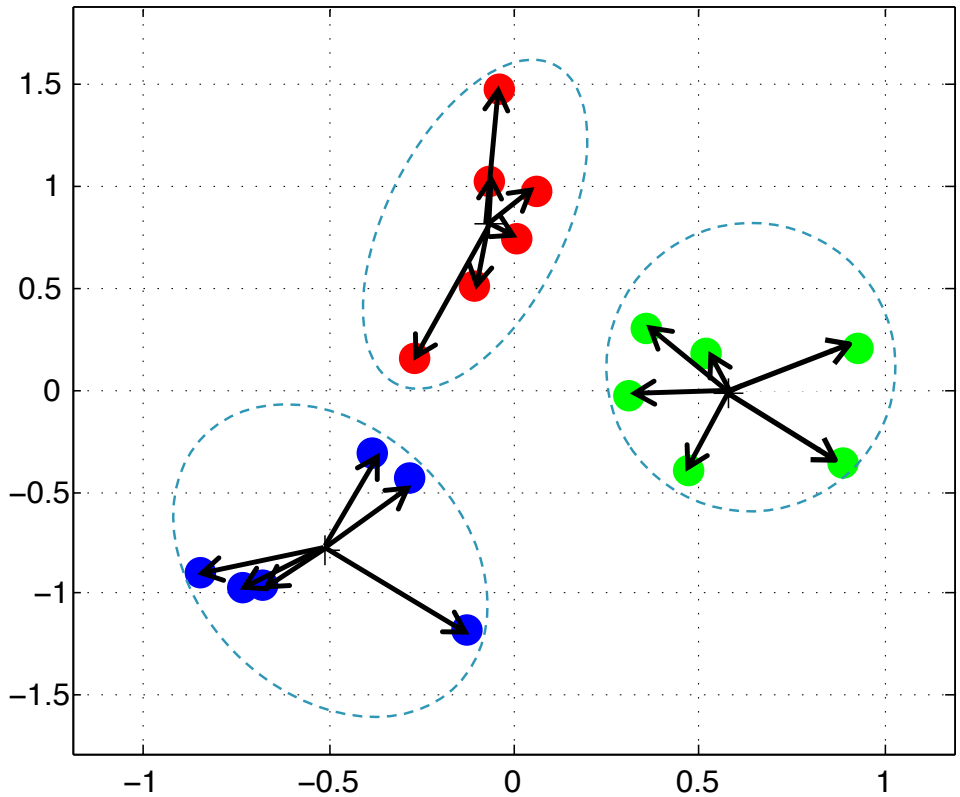
- **Score(C,D) = f(wc(C), bc(C))**
  Score mide la calidad de los clusters C para el set de datos D.
  Muchas funciones de score son una combinación de distancia dentro (wc) y distancia entre (bc) clusters
  - centroide del cluster: $$r_k = \frac{1}{n_k} \sum_{x(i) \in C_k} x(i)$$
  - distancia entre clusters: $$bc(C) = \sum_{1 \leq j \leq k \leq K} d(r_j, r_k)$$
  - distancia dentro clusters: $$wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{k=1}^{K} \sum_{x(i) \in C_k} d(x(i), r_k)$$

Métodos de partición, aprendizaje, ejemplo

Inter-grupo
Between distance



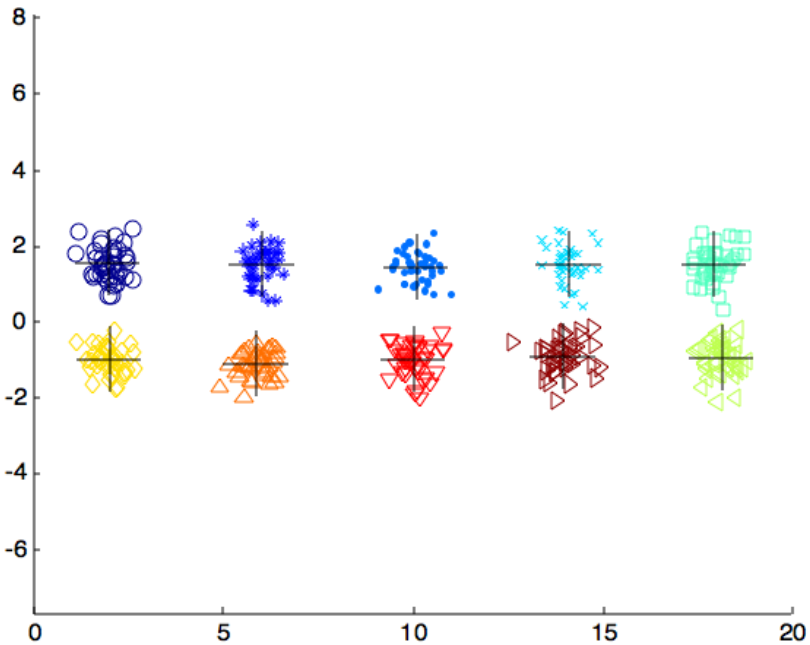
Intra-grupo
Within Distance

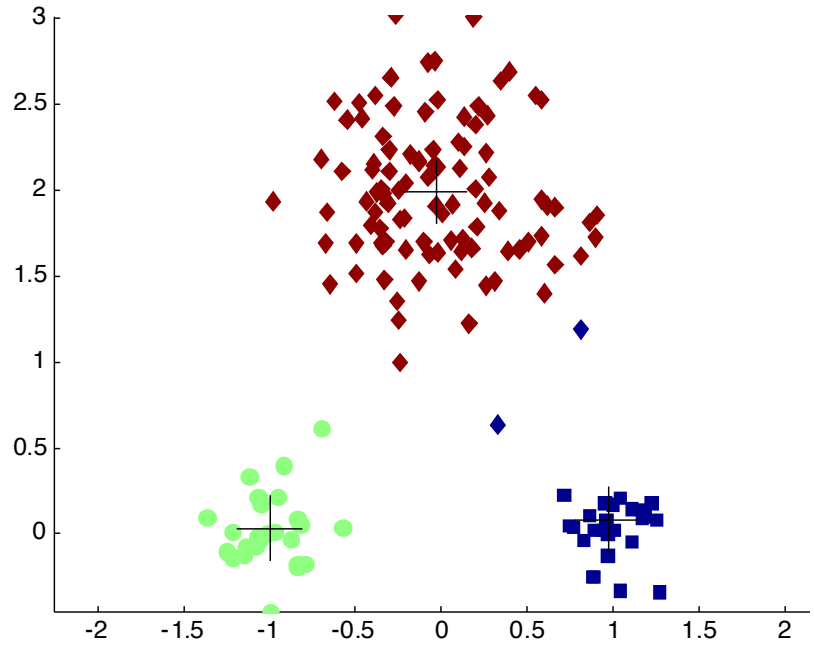


Segmentación de datos
Clustering
Métodos de partición: **K-medias (K-means)**

Métodos de partición, K-medias

- K-medias (K-means) es uno de los algoritmos de clustering más sencillo, importante, y utilizado. Dado un número **K** de centroides (definido por el usuario), cada centroide representa un cluster, y cada punto es asignado al centroide más cercano.





Métodos de partición, K-medias

- El objetivo de K-medias es minimizar la distancia de los puntos dentro de cada cluster (usualmente la distancia Euclideana al cuadrado)

$$wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{k=1}^{K} \sum_{x(i) \in C_k} d(x(i), r_k)$$

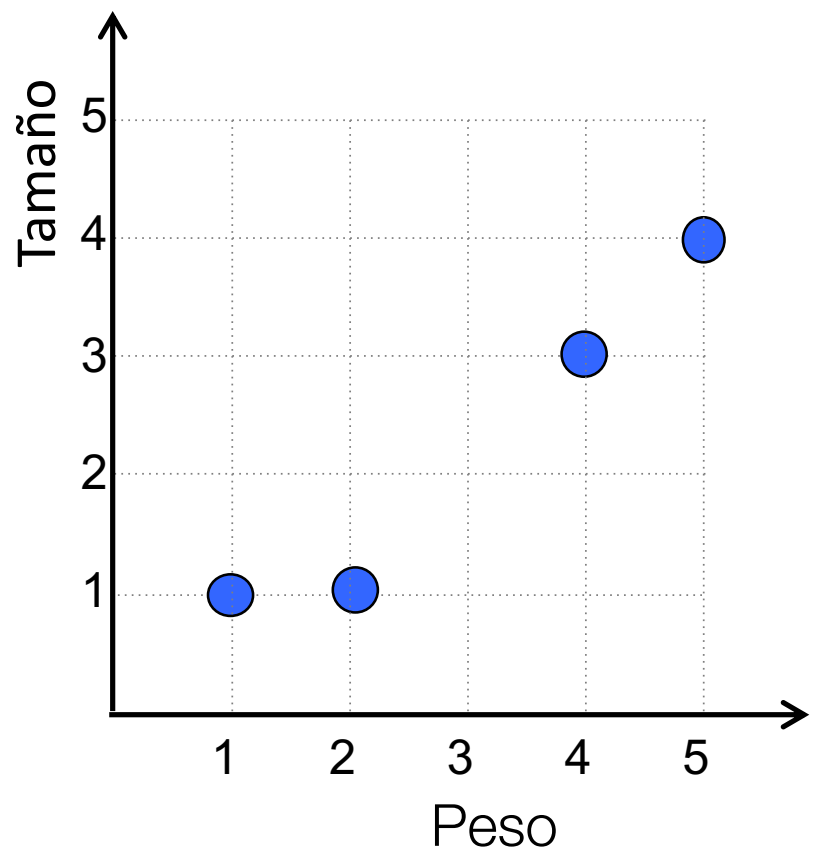
- Para ello, se hace un procedimiento iterativo que minimiza la función wc(C)

1. Seleccionar K centroides iniciales
2. **Repetir**
3. Generar K clusters asignando cada punto al centroide más cercano
4. Recalcular los centroides
5. **Hasta** los centroides no cambian

Métodos de partición, K-medias, ejemplo

- Supongamos que tenemos manzanas, y cada una de ellas tiene dos atributos o características (ver tabla). Nuestro objetivo es agrupar estos objetos en dos grupos de manzanas basadas en sus características (Peso y tamaño).

Objeto	Peso	Tamaño
Manzana A	1	1
Manzana B	2	1
Manzana C	4	3
Manzana D	5	4



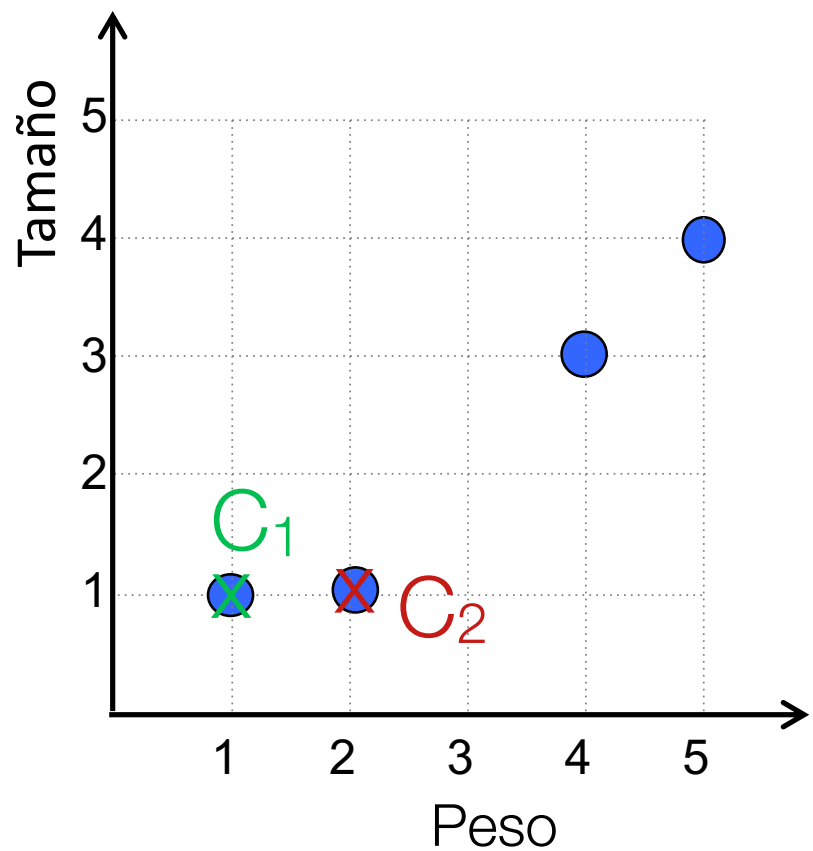
Métodos de partición, K-medias, ejemplo

- **Código línea 1: determinar los centroides**
  Supongamos que utilizamos la manzana A y B como los centroides.

- Sea C_1 y C_2 los coordenadas de los centroides. Usted puede definir como centroides una posición al azar.

- **Centroides:** Corresponden al punto central del grupo de puntos.

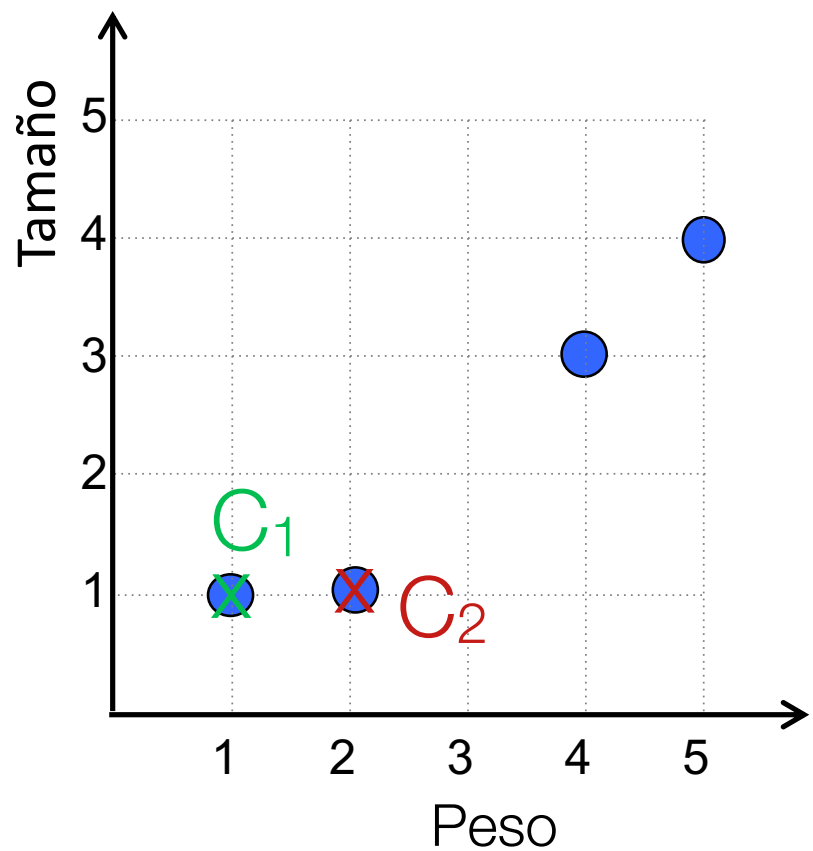
  Como hemos definido en nuestro problema, dado que k=2, entonces tenemos dos centroides C_1 y C_2.



Métodos de partición, K-medias, ejemplo

- **Código línea 3: distancia a los centroides**
  Calculamos la distancia de cada punto a cada uno de los centroides. En este ejemplo emplearemos la distancia euclidiana, y luego definimos la matriz de distancia.

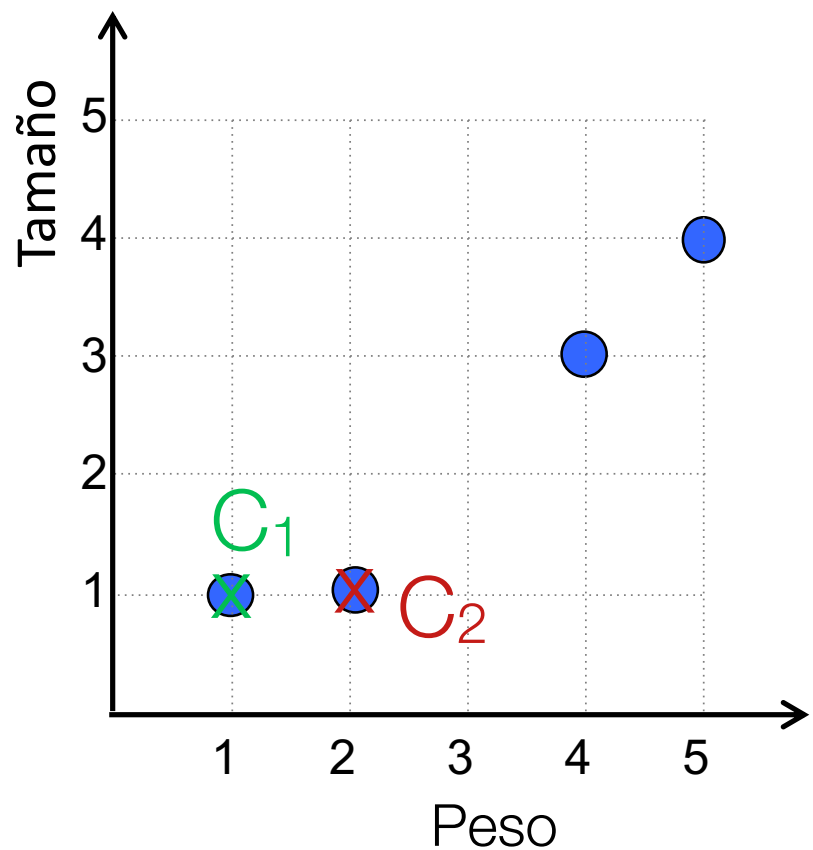
Objeto	Peso	Tamaño	Dist C_1	Dist C_2
Manzana A	1	1	0.00	1.00
Manzana B	2	1	1.00	0.00
Manzana C	4	3	3.61	2.83
Manzana D	5	4	5.00	4.24



Métodos de partición, K-medias, ejemplo

- **Código línea 3 (B): clustering de objetos a los centroides**
  Asignamos cada objeto a cada clase basado en la distancia mínima. Así la manzana A es asignada a la clase 1, la manzana B, C y D a la clase 2.

Objeto	Peso	Tamaño	Dist C_1	Dist C_2	A.
Manzana A	1	1	0.00	1.00	C_1
Manzana B	2	1	1.00	0.00	C_2
Manzana C	4	3	3.61	2.83	C_2
Manzana D	5	4	5.00	4.24	C_2

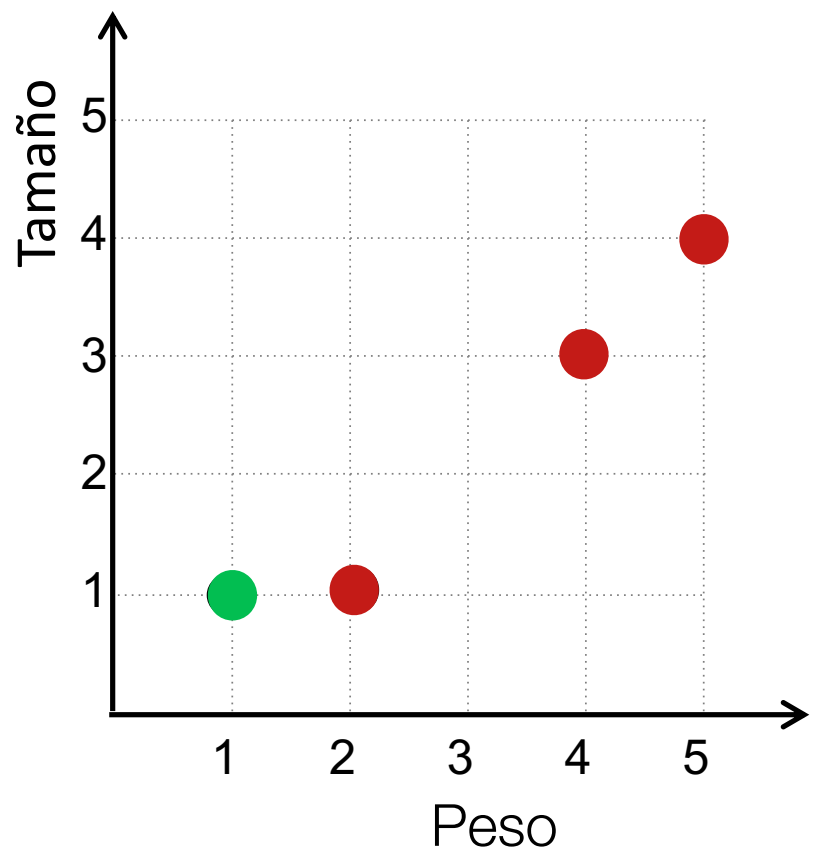


Métodos de partición, K-medias, ejemplo

- **Pausa**
  Si visualizáramos la clusterización actual, obtendríamos lo siguiente.

- Ahora debemos recalcular los centroides y verificar si tenemos que hacer una 2da iteración del algoritmo.

1. Seleccionar K centroides iniciales
2. **Repetir**
3. Generar K clusters asignando cada punto al centroide más cercano
4. Recalcular los centroides
5. **Hasta** los centroides no cambian

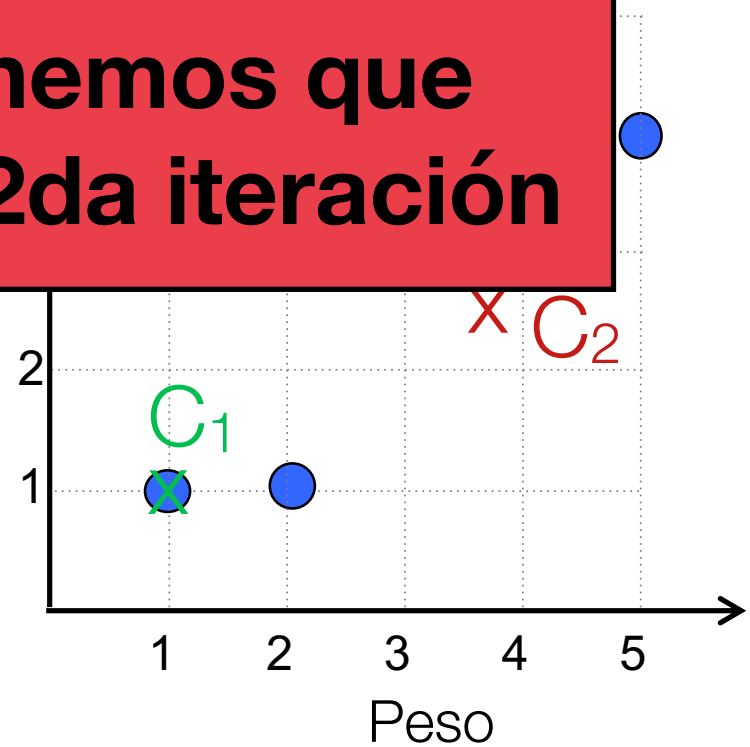


Métodos de partición, K-medias, ejemplo

- **Código línea 4: recalcular centroides**
  Una vez conocida los miembros de cada grupo, ahora calcularemos la posición de los nuevos centroides basado en la posición de dichos miembros.

Objeto	Peso
Manzana A	1
Manzana B	2
Manzana C	4
Manzana D	5

Los centroides cambiaron de lugar, tenemos que ejecutar una 2da iteración



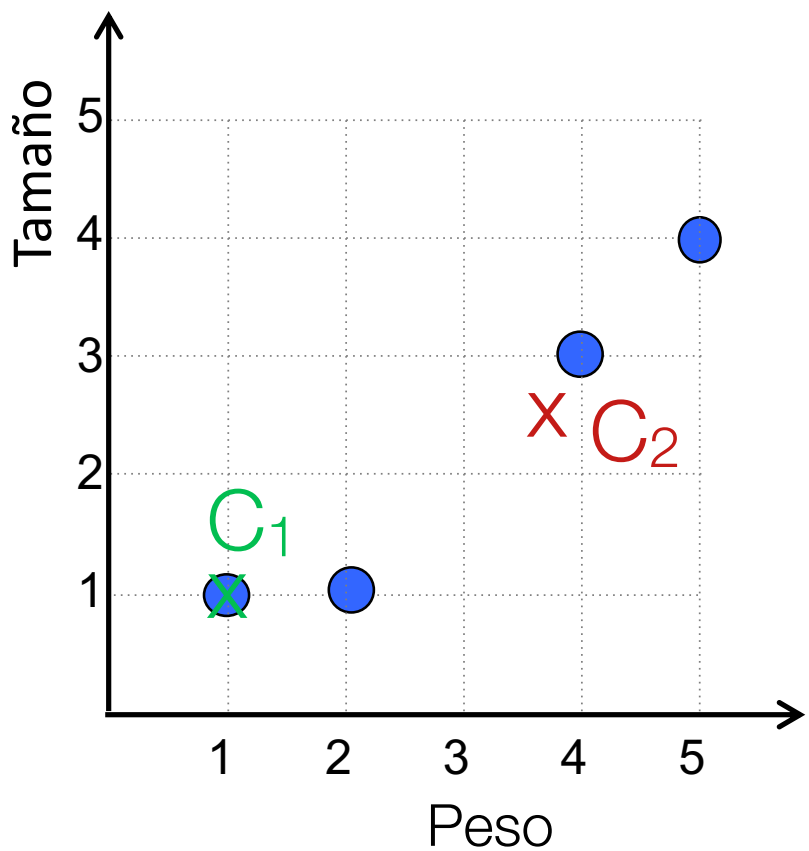
$$C_1 = (1, 1)$$

$$C_2 = \left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right) = \left(\frac{11}{3}, \frac{8}{3}\right)$$

Métodos de partición, K-medias, ejemplo, 2da iteración

- **Código línea 3: distancia a los centroides**
  Calculamos la distancia de cada punto a cada uno de los centroides. En este ejemplo emplearemos la distancia euclidiana, y luego definimos la matriz de distancia.

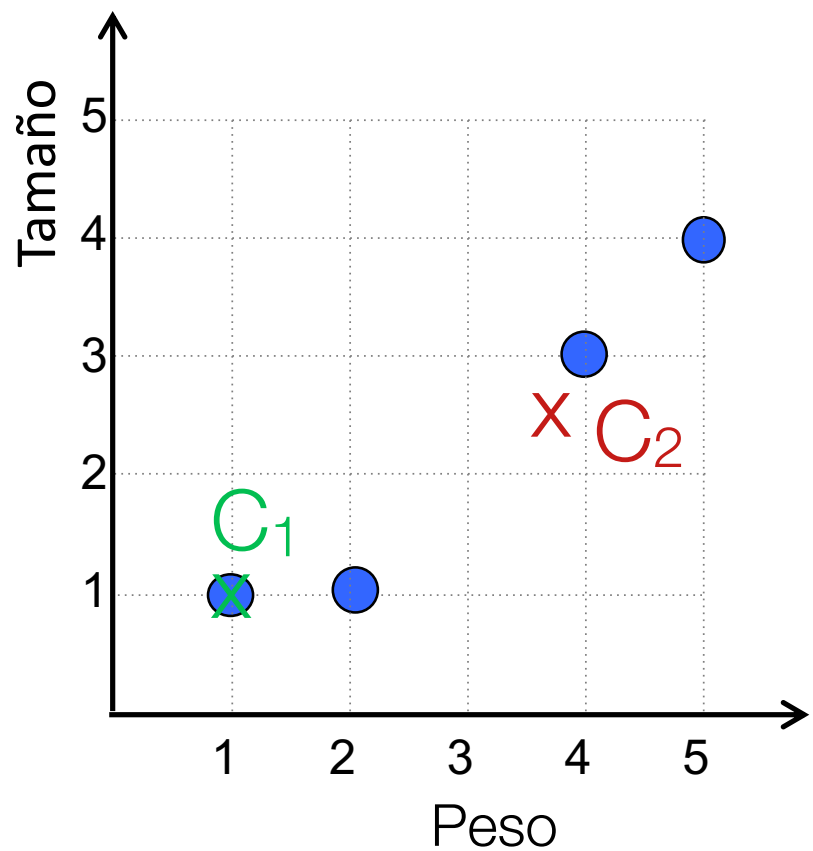
Objeto	Peso	Tamaño	Dist C_1	Dist C_2
Manzana A	1	1	0.00	3.14
Manzana B	2	1	1.00	2.26
Manzana C	4	3	3.61	0.47
Manzana D	5	4	5.00	1.89



Métodos de partición, K-medias, ejemplo, 2da iteración

- **Código línea 3 (B): clustering de objetos a los centroides (2da iteración)** Asignamos cada objeto a cada clase basado en la distancia mínima. Así la manzana A es asignada a la clase 1, la manzana B, C y D a la clase 2.

Objeto	Peso	Tamaño	Dist C_1	Dist C_2	A.
Manzana A	1	1	0.00	3.14	C_1
Manzana B	2	1	1.00	2.26	C_1
Manzana C	4	3	3.61	0.47	C_2
Manzana D	5	4	5.00	1.89	C_2

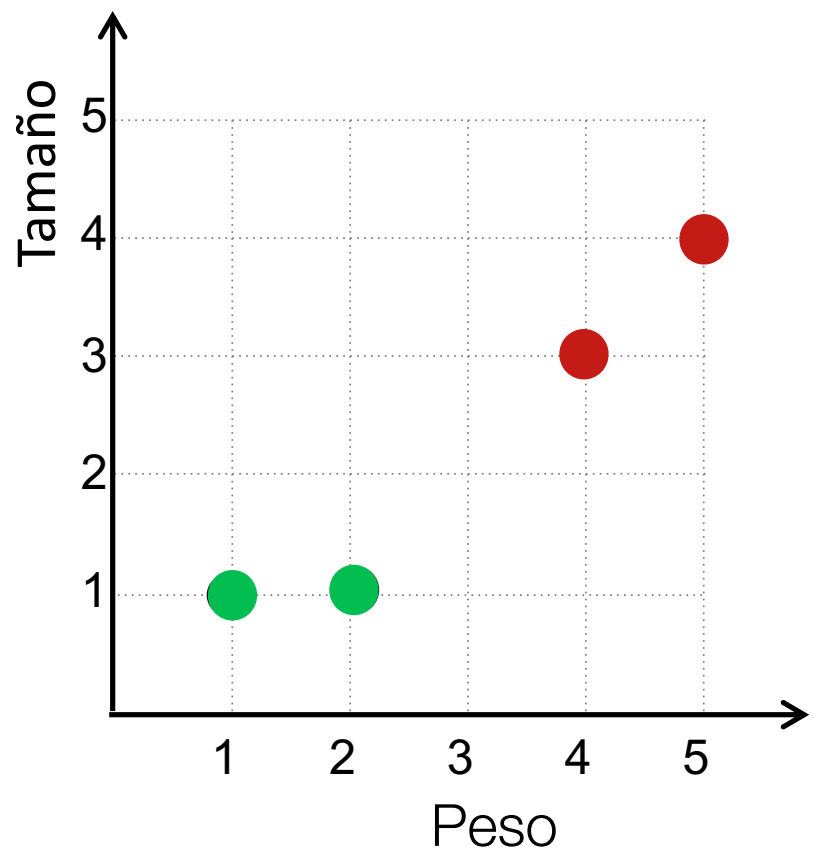


Métodos de partición, K-medias, ejemplo, 2da iteración

- **Pausa (2da iteración)**
  Si visualizáramos la clusterización actual, obtendríamos lo siguiente.

- Ahora debemos recalcular los centroides y verificar si tenemos que hacer una 3ra iteración del algoritmo.

1. Seleccionar K centroides iniciales
2. **Repetir**
3. Generar K clusters asignando cada punto al centroide más cercano
4. Recalcular los centroides
5. **Hasta** los centroides no cambian



Métodos de partición, K-medias, ejemplo, 2da iteración

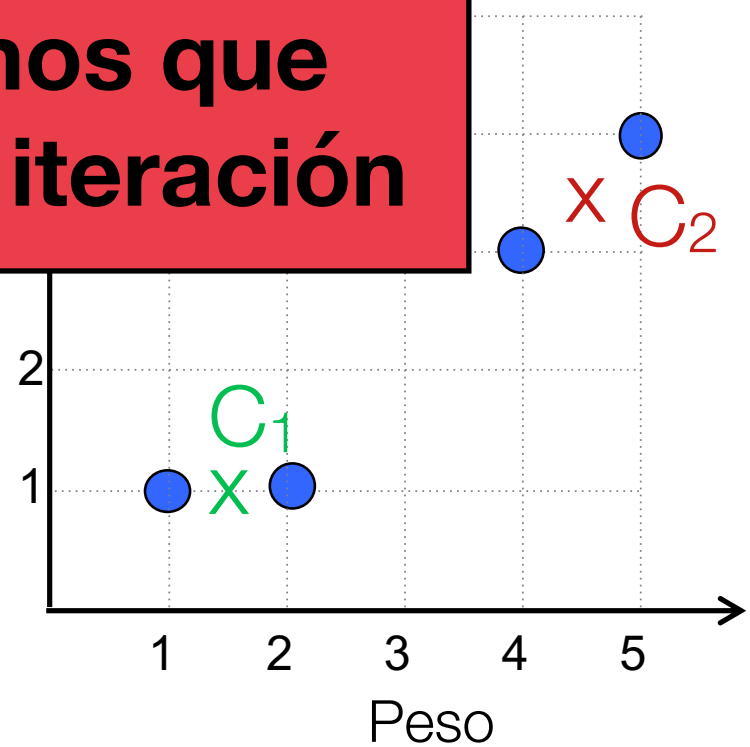
- **Código línea 4: recalcular centroides (2da iteración)**
  Una vez conocida los miembros de cada grupo, ahora calcularemos la posición de los nuevos centroides basado en la posición de dichos miembros

Los centroides cambiaron de lugar, tenemos que ejecutar una 3ra iteración

Objeto			
Manzana A			
Manzana B			
Manzana C			
Manzana D	5	4	C_2

$$C_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = \left(\frac{3}{2}, \frac{2}{2}\right)$$

$$C_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = \left(\frac{9}{2}, \frac{7}{2}\right)$$



Métodos de partición, K-medias, ejemplo, 3ra iteración

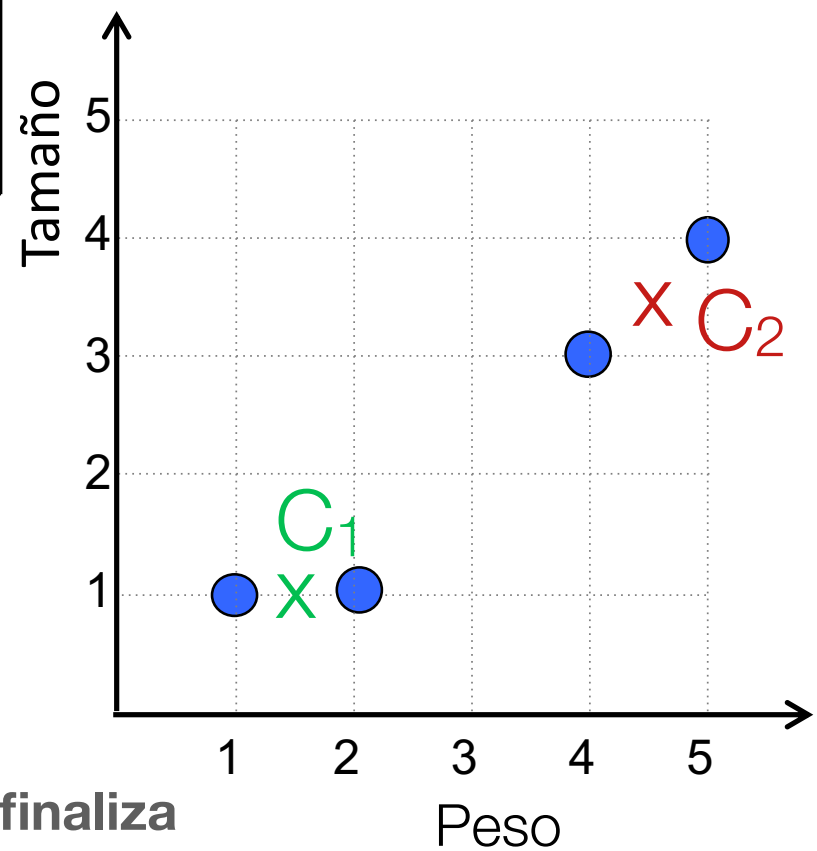
- **Código línea 3: distancia y asignación de centroides**

Objeto	Peso	Tamaño	Dist C_1	Dist C_2	A.
Manzana A	1	1	0.50	4.30	C_1
Manzana B	2	1	0.50	3.53	C_1
Manzana C	4	3	4.03	0.70	C_2
Manzana D	5	4	5.41	0.70	C_2

- **Código línea 4: recalcular centroides**

$$C_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = \left(\frac{3}{2}, \frac{2}{2}\right)$$

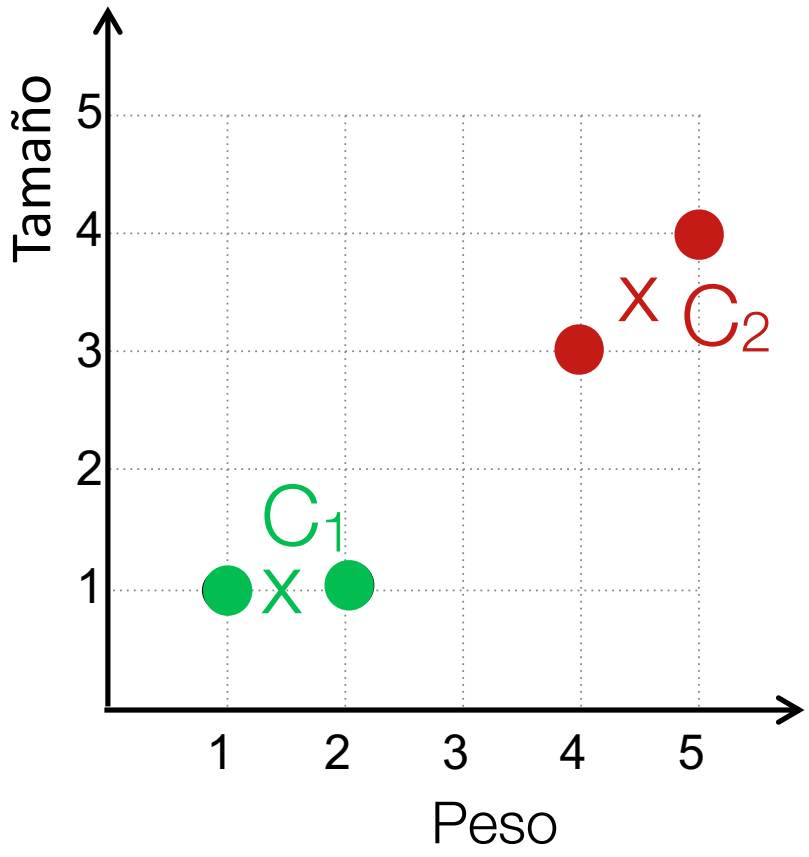
$$C_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = \left(\frac{9}{2}, \frac{7}{2}\right)$$



- **Los centroides NO cambiaron, K-means finaliza**

Métodos de partición, K-medias, ejemplo, fin de k-means

- **Fin del algoritmo**
  Los centroides no cambiaron, el algoritmo finaliza y entrega los centroides finales y la asignación de cada punto a su cluster correspondiente.



Métodos de partición, K-medias como algoritmo EM

- El objetivo de K-medias es minimizar la distancia de los puntos dentro de cada cluster

$$wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{k=1}^{K} \sum_{x(i) \in C_k} d(x(i), r_k)$$

- Sea $r_{ik} \in \{0,1\}$ una variable indicando si el punto i pertenece al cluster k (un punto debe pertenecer exclusivamente a un cluster), $\mathbf{x}_i$ el vector correspondiente al i-esimo punto, y $\boldsymbol{\mu}_k$ un vector representativo del cluster k. Si usamos la distancia Euclideana al cuadrado, obtenemos:

$$wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} ||\mathbf{x}_i - \boldsymbol{\mu}_k||^2$$

- Ahora, nuestro objetivo es encontrar los valores de la variable latente r_{ik} y $\boldsymbol{\mu}_k$ tal que minimice wc(C).

K-medias como algoritmo EM

$$wc(C) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} ||\mathbf{x}_i - \boldsymbol{\mu}_k||^2$$

- Dado valores iniciales para r_{ik} y $\boldsymbol{\mu}_k$ podemos minimizar wc(C) iterando:
  - Minimizar wc(C) dejando fijo $\boldsymbol{\mu}_k$ y buscando r_{ik}.
  - Minimizar wc(C) dejando fijo r_{ik} y buscando $\boldsymbol{\mu}_k$.
- El paso de buscar r_{ik} corresponde a encontrar el valor "esperado" de r_{ik} (E-step).
- El paso de buscar $\boldsymbol{\mu}_k$ corresponde a maximizar wc(C) (M-step)

K-medias como algoritmo EM

$$wc(C) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} ||\mathbf{x}_i - \boldsymbol{\mu}_k||^2$$

- **E-STEP:** queremos minimizar wc(C) para cada punto i, dado $\boldsymbol{\mu}_k$ fijo, específicamente, queremos encontrar el menor valor para cada $r_{ik} ||\mathbf{x}_i - \boldsymbol{\mu}_k||^2$

  La solución es bastante sencilla

$$r_{ik} = \begin{cases} 1 & \text{si } k = \arg\min_j ||\mathbf{x}_i - \boldsymbol{\mu}_j||^2 \\ 0 & \text{en otro caso} \end{cases}$$

  es decir, se asigna el i-ésimo punto al vector $\boldsymbol{\mu}_j$ más cercano

- **M-STEP:** queremos minimizar wc(C) para cada punto i, dado r_{ik} fijo. Derivando wc(C) con respecto a cada $\boldsymbol{\mu}_k$ e igualando a 0, obtenemos:

$$\frac{\partial wc(C)}{\partial \boldsymbol{\mu}_k} = 2 \sum_{i=1}^{N} r_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k) = 0 \Rightarrow \boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} r_{ik} \mathbf{x}_i}{\sum_{i=1}^{N} r_{ik}}$$

  $\boldsymbol{\mu}_k$ se minimiza como el promedio de los puntos que pertenecen al cluster.

Métodos de partición, K-medias, código en python

- Se utilizará python para los códigos.

- Principalmente se utilizará las bibliotecas numpy, pandas, sklearn y plotnine (ggplot2)

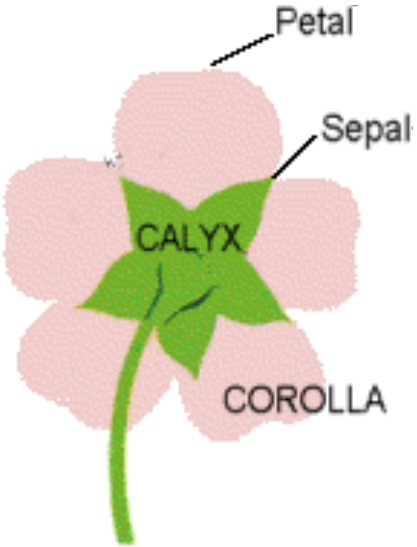
```
import numpy as np
import pandas as pd
from plotnine import *
```

- La mayoría de los modelos de sklearn corresponden a “objetos” de cierta “clase” con propiedades y funciones específicas.

```
from sklearn.xxx import modeloDeseado #Importación del modelo
nuevoModelo = modeloDeseado() #Creación del modelo
nuevoModelo.function() #Aplicar alguna función
nuevoModelo.attribute  #Ver los valores de un atributo
```

Métodos de partición, K-medias, datos flor de iris

- Los datos de la flor de iris corresponden a mediciones del largo y ancho del pétalo y el largo y ancho del sépalo de 3 tipos de flores (setosa, virginica, y versicolor)



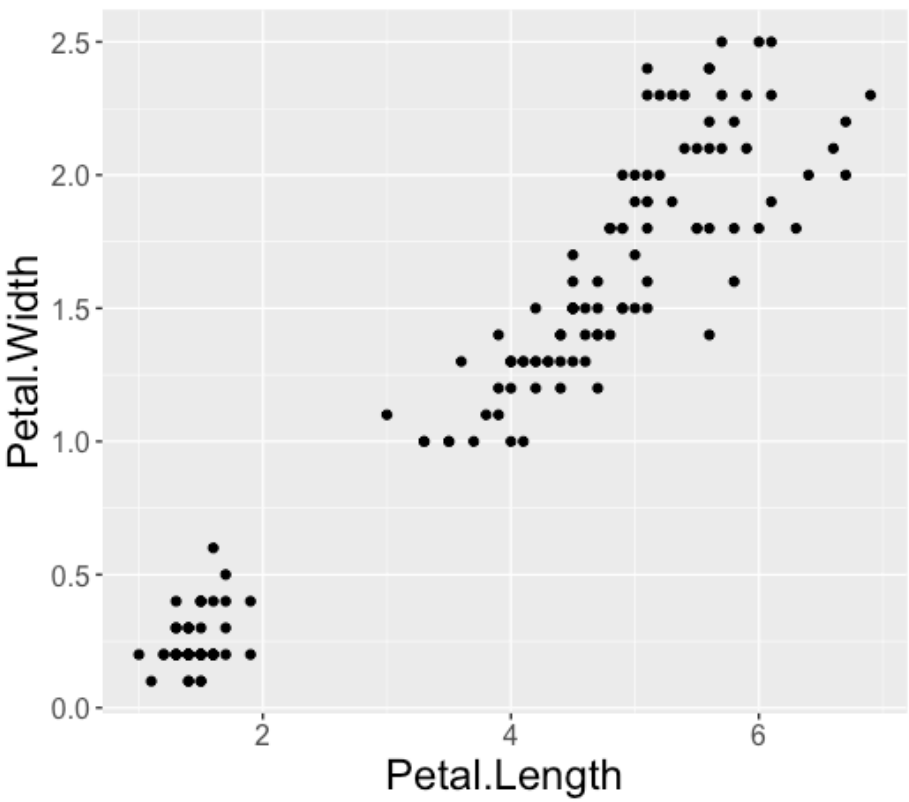
Setosa

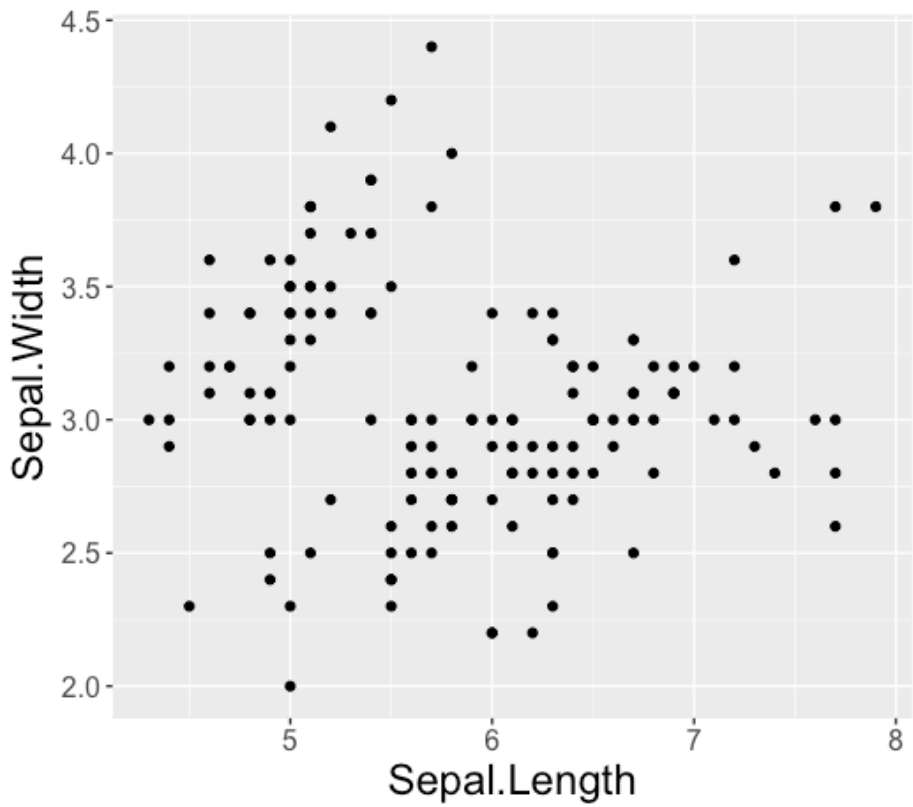
Virginica

Versicolor

Métodos de partición, K-medias, código en Python

- #Visualicemos los datos IRIS
  ggplot(tempDF)+aes(x="sepal length (cm)", y="sepal width (cm)")+ geom_point()
  ggplot(tempDF)+aes(x="petal length (cm)", y="petal width (cm)")+ geom_point()





Métodos de partición, K-medias, código en Python

- La clase KMeans de sklearn.cluster permite aplicar el algoritmo k-medias
  KMeans(n_clusters=8, init='k-means++', n_init=10, max_iter=300, tol=0.0001, random_state=None)
- n_clusters: número de clusters
- init: ‘k-means++’ inicialización inteligente, 'random' aleatoria
- n_init: Número de veces que aplicaremos k-means
- max_iter: Máximo número de iteraciones para cada ejecución
- tol: Tolerancia para la convergencia
- random_state: Semilla para inicializar los centroides.

Métodos de partición, K-medias, código en Python

- #Creemos un modelo de KMeans
  from sklearn.cluster import KMeans
  km = KMeans()

- km ha creado un objeto para k-means con 8 clusters, inicialización inteligente, 10 repeticiones del algoritmo por aplicación, un máximo de 300 iteraciones por aplicación y una tolerancia de cambio de 0.0001.

- #Aplicando k-means a los datos de iris
  km=km.fit(tempDF)

- Una vez entrenado el modelo, existen nuevas características que podemos observar

  - cluster_centers: Las coordenadas de los centroides. Si el algoritmo no converge estos no serán consistentes con los labels.

  - labels: Las etiquetas de cada punto (cluster al que pertenece)

  - inertia_float: la suma total de los within cluster distance

  - n_iter_int: número de iteraciones del algoritmo

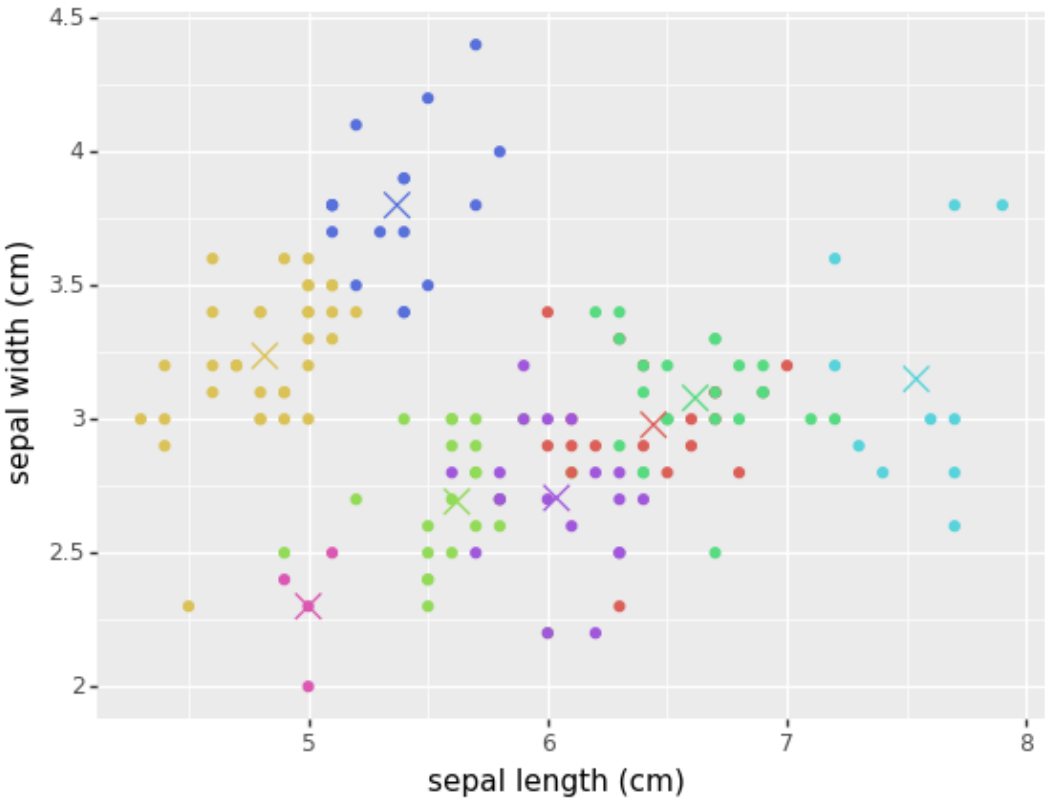
Métodos de partición, K-medias, código en Python

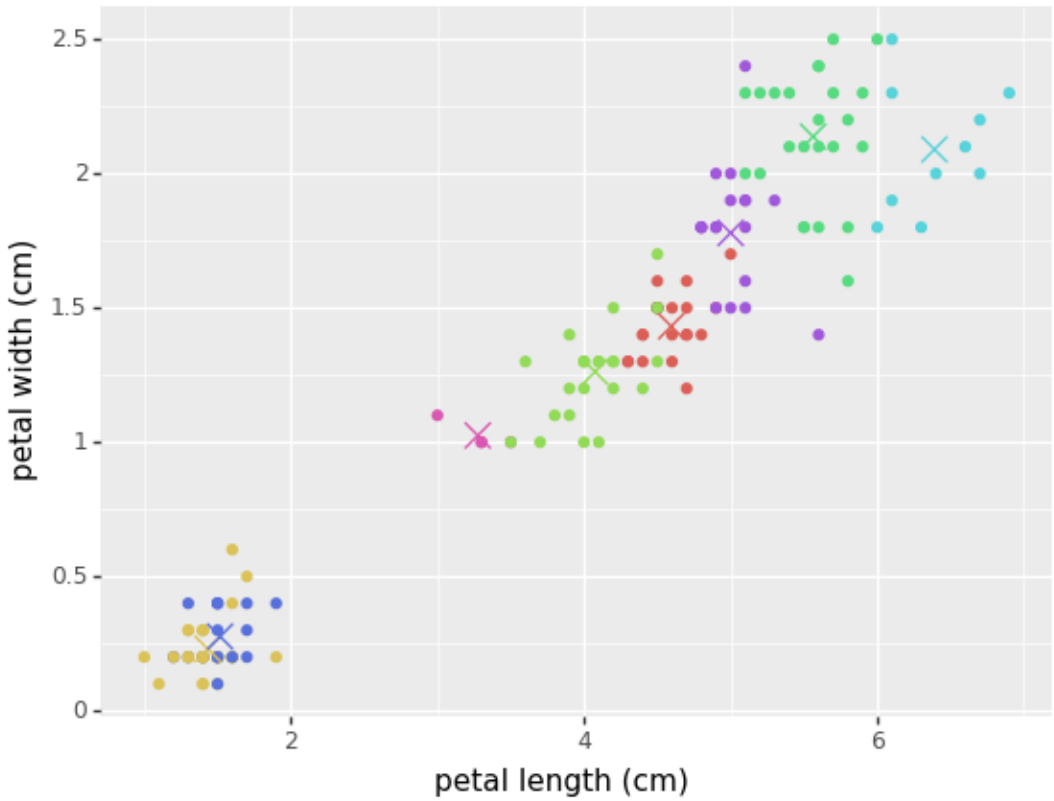
- #Veamos las características del modelo aprendido

```
print(km.cluster_centers_) #Mostrando los centros de los clusters
[[6.44210526  2.97894737  4.59473684  1.43157895]
 [4.81818182  3.23636364  1.43333333  0.23030303]
 [5.62083333  2.69166667  4.07500000  1.26250000]
 [6.61666667  3.07916667  5.56250000  2.13750000]
 [7.54000000  3.15000000  6.39000000  2.09000000]
 [5.37058824  3.80000000  1.51764706  0.27647059]
 [6.03684211  2.70526316  5.00000000  1.77894737]
 [5.00000000  2.30000000  3.27500000  1.02500000]]

print(km.inertia_) #Within cluster distance
30.499340627638613

print(km.labels_) #Etiquetas
[1 1 1 1 1 5 1 1 1 1 5 1 1 1 5 5 5 1 5 5 5 5 1 1 1 1 1 5 1 1 1 5 5 5 1 1 5
 1 1 1 1 1 1 1 5 1 5 1 5 1 0 0 0 2 0 2 0 7 0 2 7 2 2 0 2 0 2 2 6 2 6 2 6 0
 0 0 0 0 0 2 2 2 2 6 2 0 0 0 2 2 2 0 2 7 2 2 2 0 7 2 3 6 3 3 3 4 2 4 3 4 3
 6 3 6 6 3 3 4 4 6 3 6 4 6 3 4 6 6 3 3 4 4 3 6 6 4 3 3 6 3 3 3 6 3 3 3 6 3
 3 6]
```

Métodos de partición, K-medias, código en Python

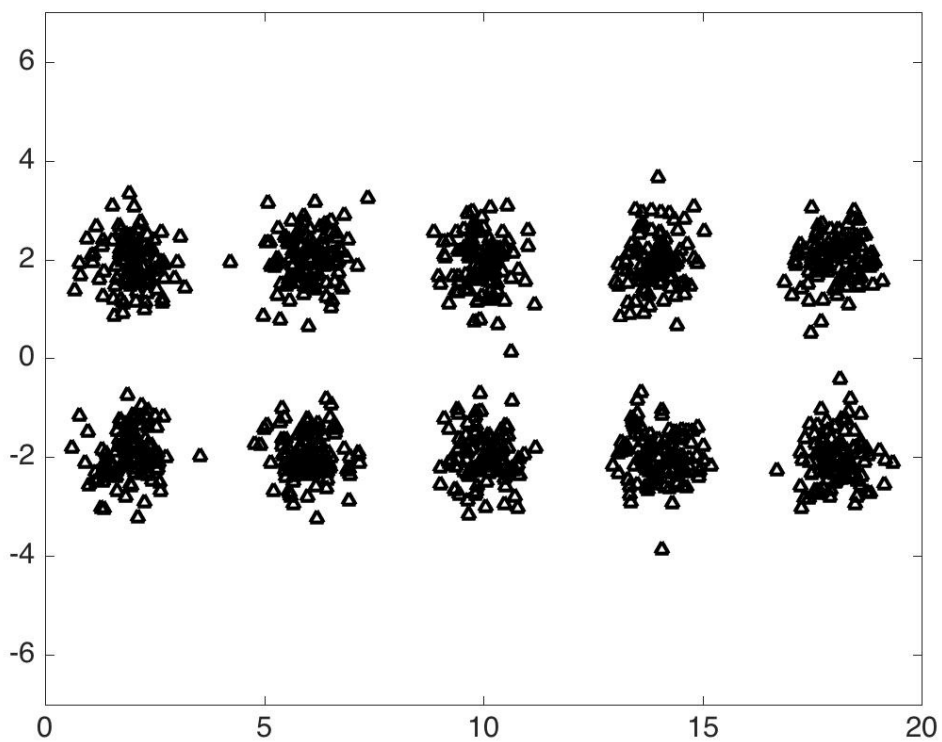
- #Veamos un plot de los cluster encontrados

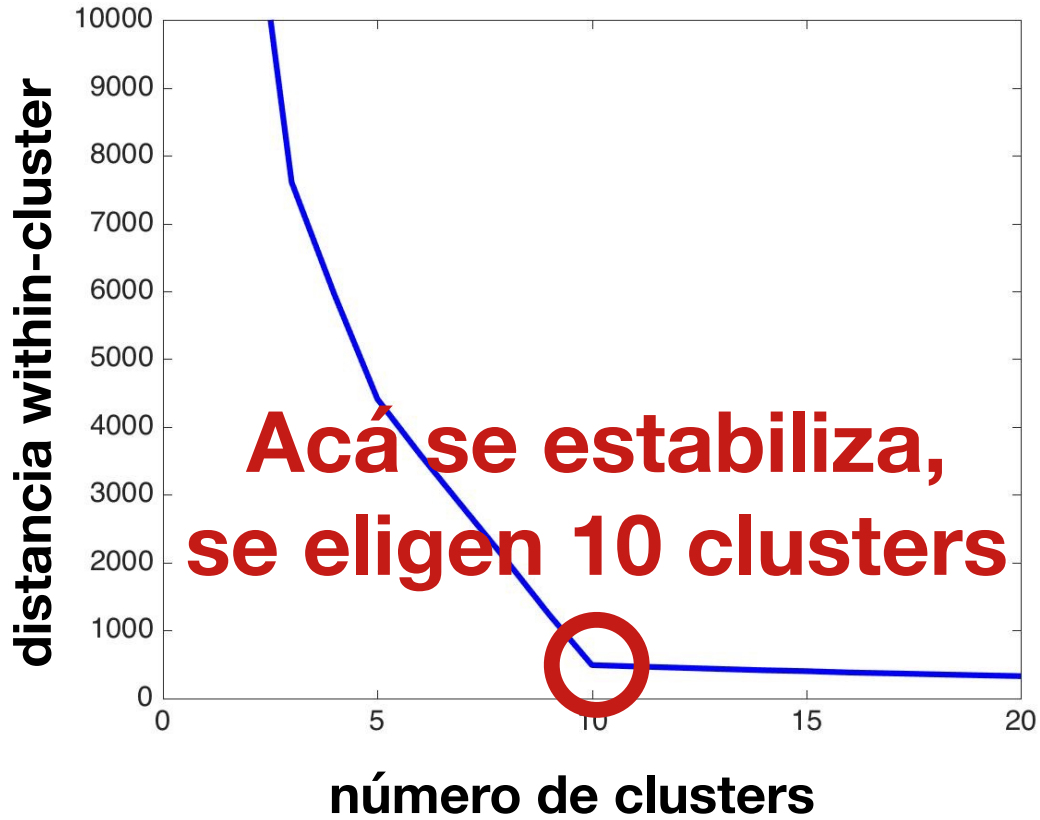




Métodos de partición, K-medias, selección de K

- Para seleccionar el valor de K es posible aplicar una heurística, aumentar el número de clusters hasta que la diferencia en la función de error (la suma de los within cluster para cada cluster) se estabilice.

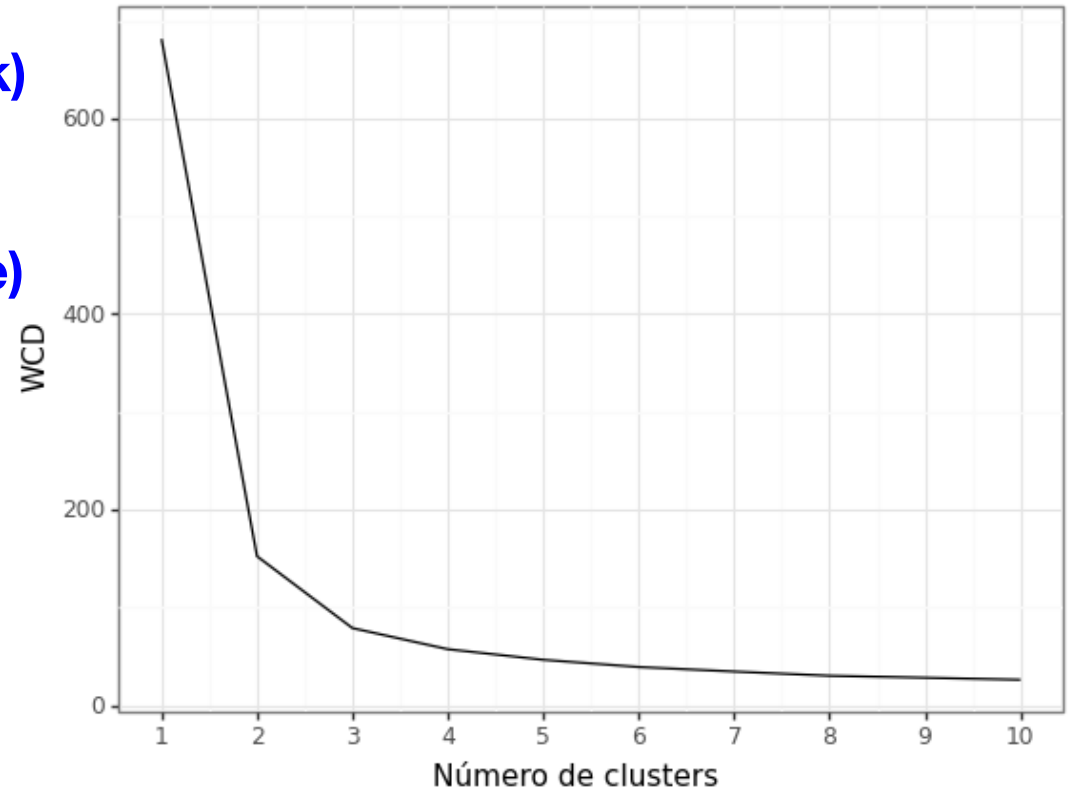




Métodos de partición, K-medias, selección de K

- #Calculemos los valores del total within-cluster distance

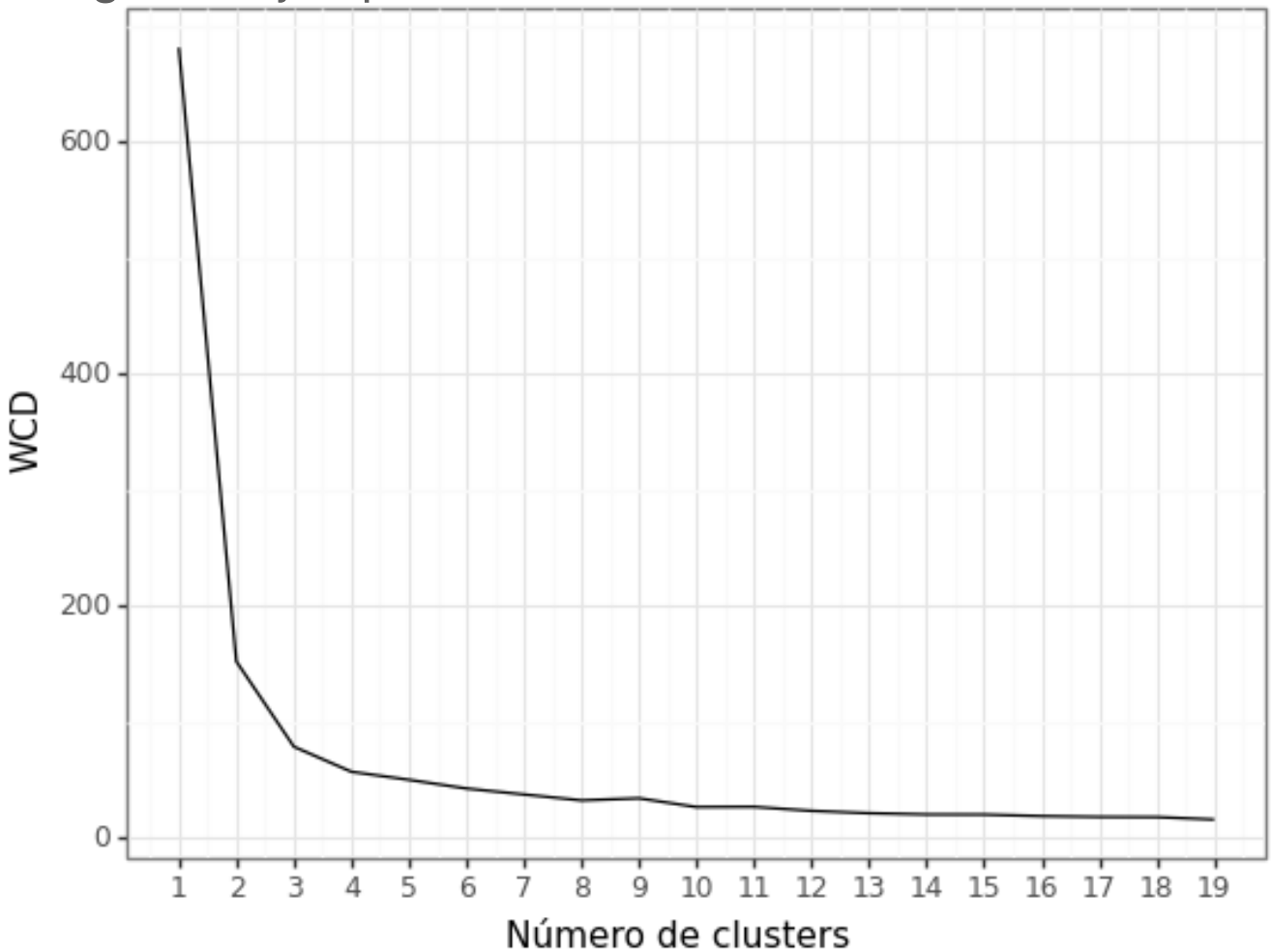
```
sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(iris.data)
    sse.append(kmeans.inertia_)
(ggplot()+aes(x=range(1, 11),y=sse)
 +theme_bw()+geom_line()
 +labs(x="Número de clusters",
   y="WCD")
 +scale_x_continuous(
      breaks=range(1,11)))
```



¿Cuántos clusters elegiría?

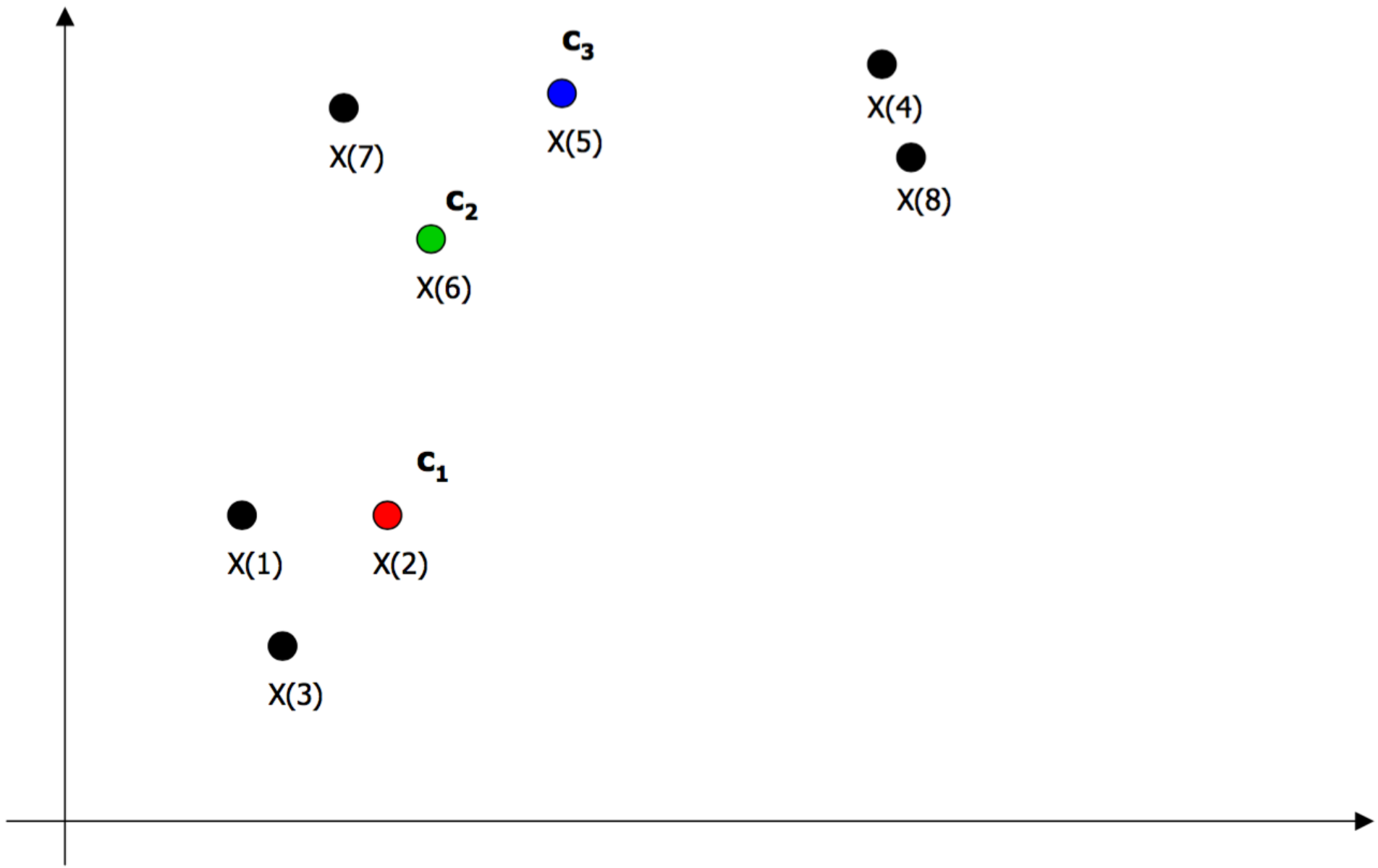
Métodos de partición, K-medias, selección de K

- Veamos un segundo ejemplo

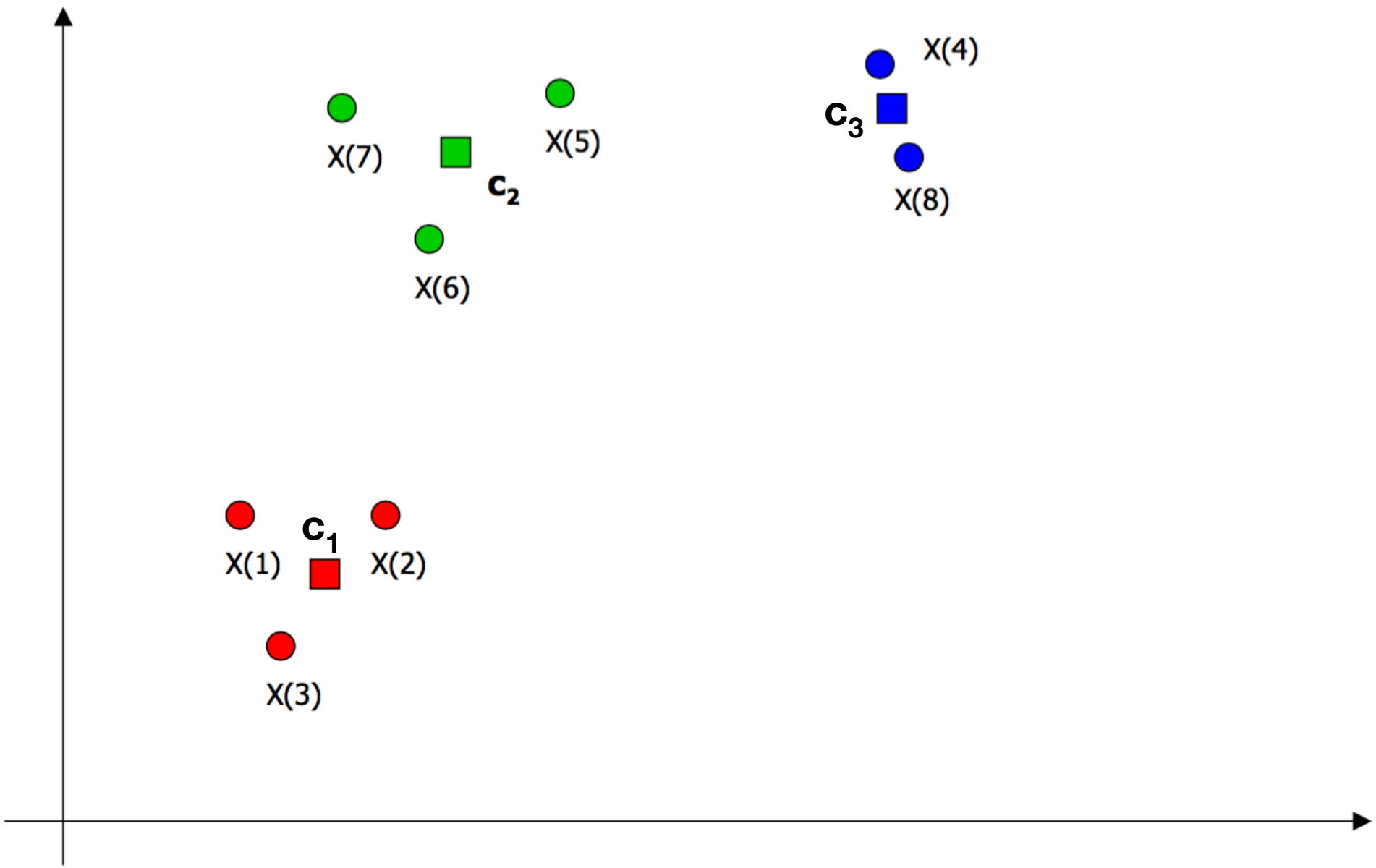


¿Cuál es el problema con esta curva?

Métodos de partición, K-medias, selección inicial de centroides

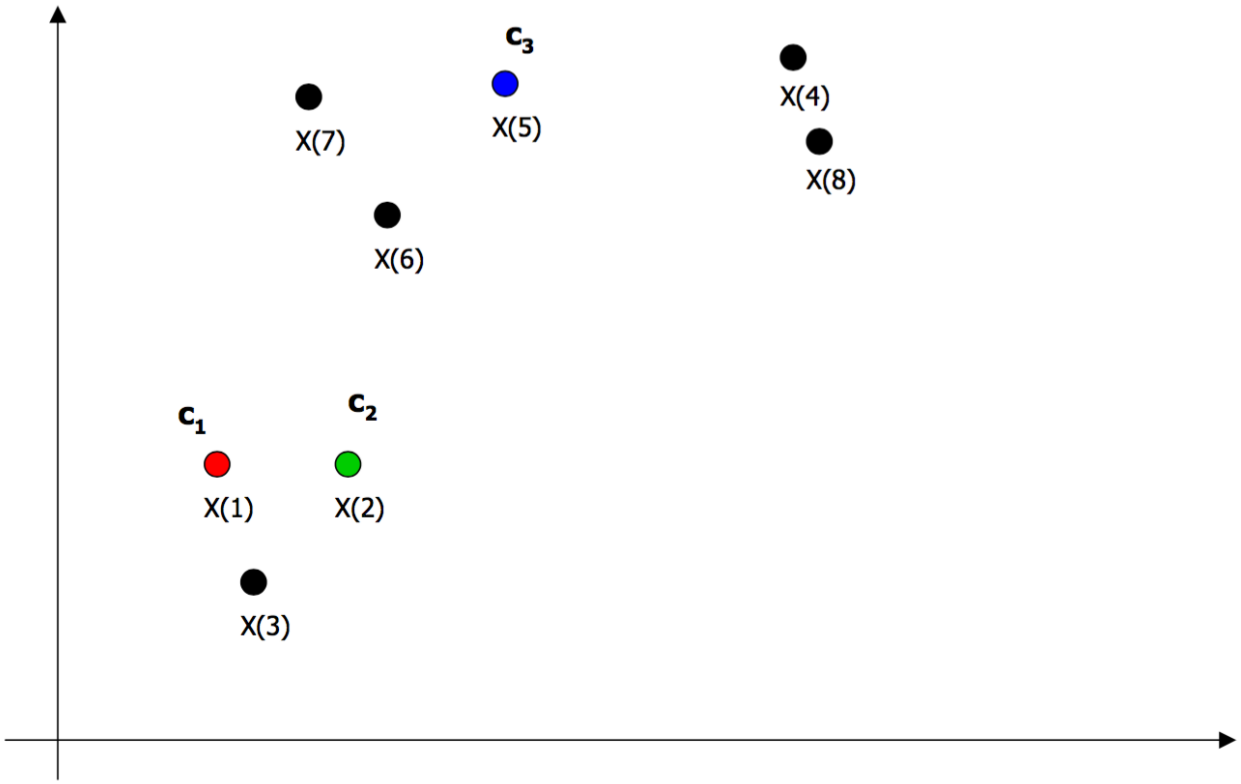


Métodos de partición, K-medias, selección inicial de centroides



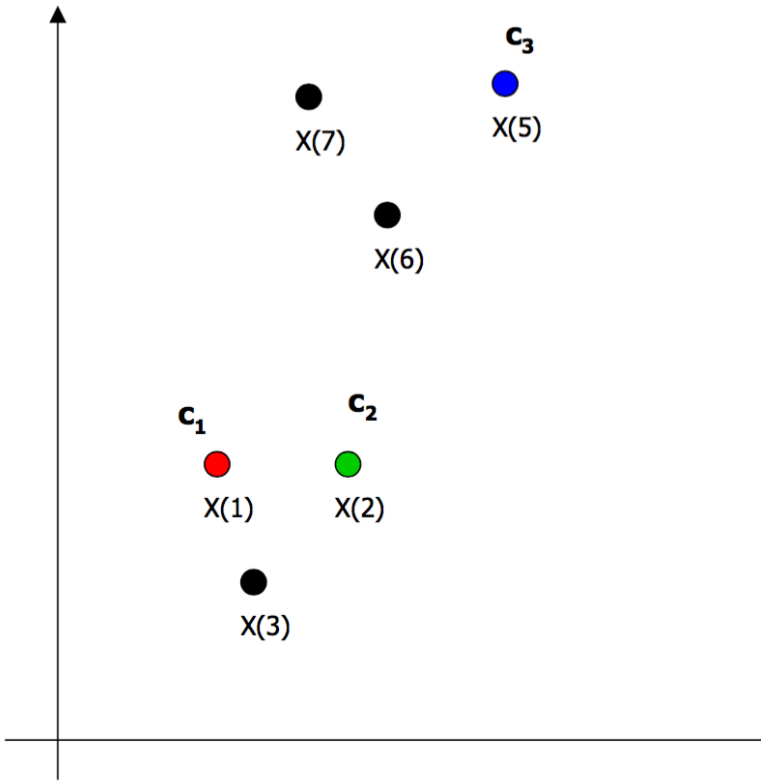
Métodos de partición, K-medias, selección inicial de centroides

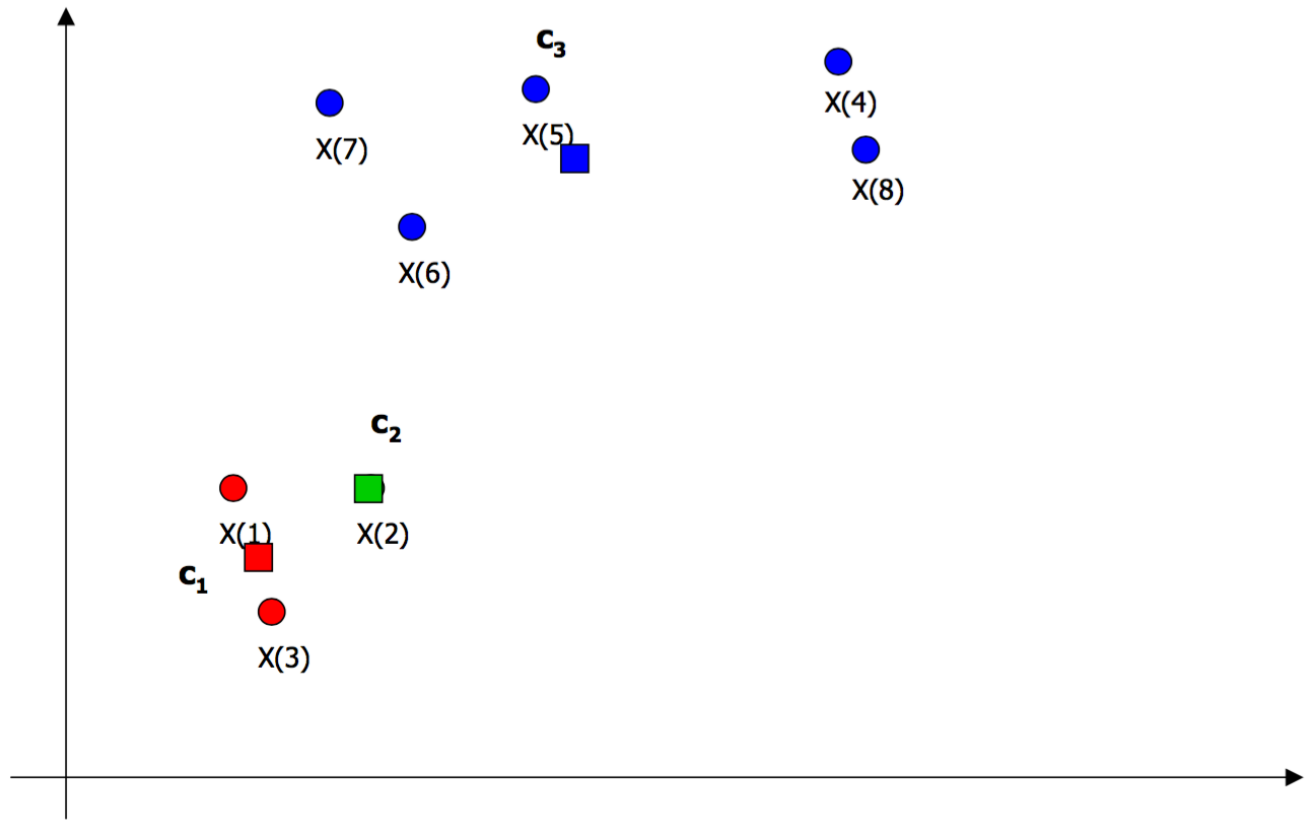
- La selección inicial de los centroides puede determinar los clusters finales.



Métodos de partición, K-medias, selección inicial de centroides

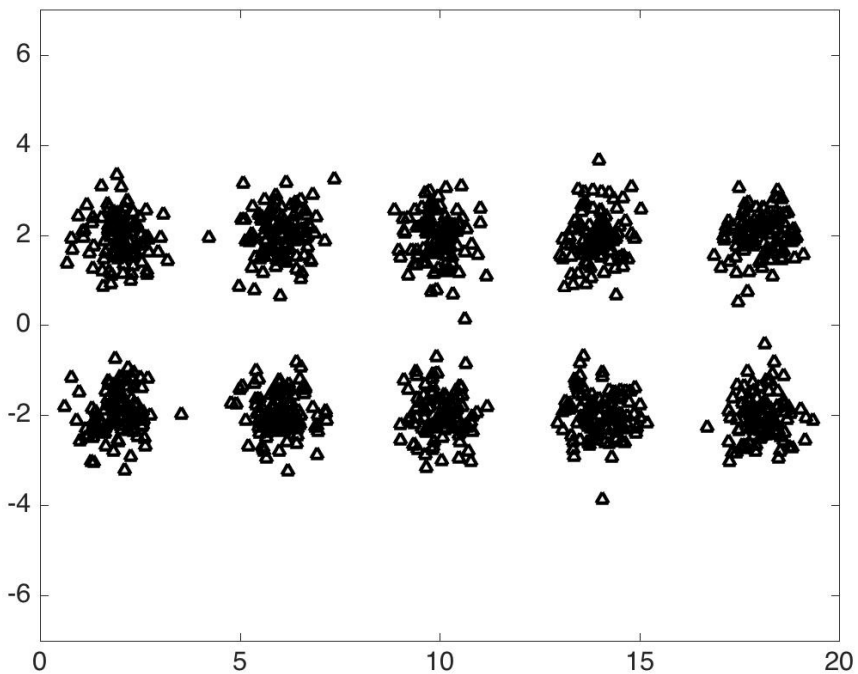
- La selección inicial de los centroides puede determinar los clusters finales.

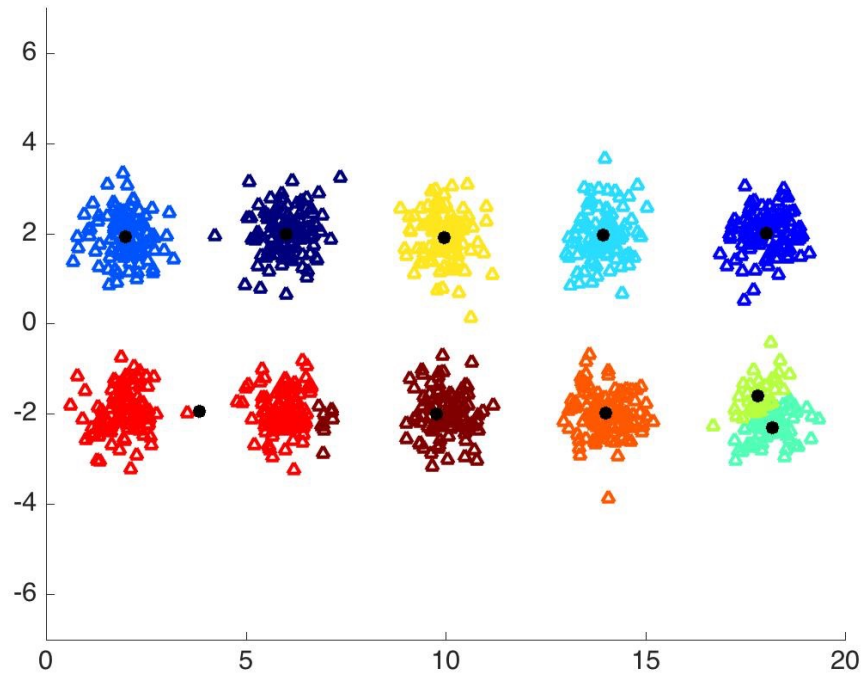




Métodos de partición, K-medias, selección inicial de centroides

- **Selección inicial de centroides:**
  - Correr K-medias con multiples selecciones aleatoria de centroids y seleccionar el resultado con mejor score (mínima distancia entre clusters)

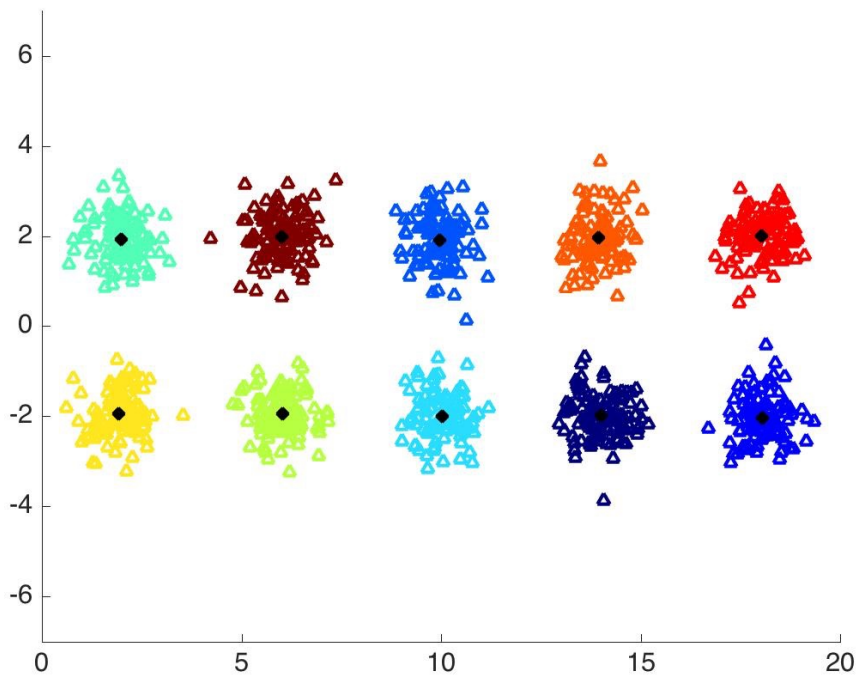




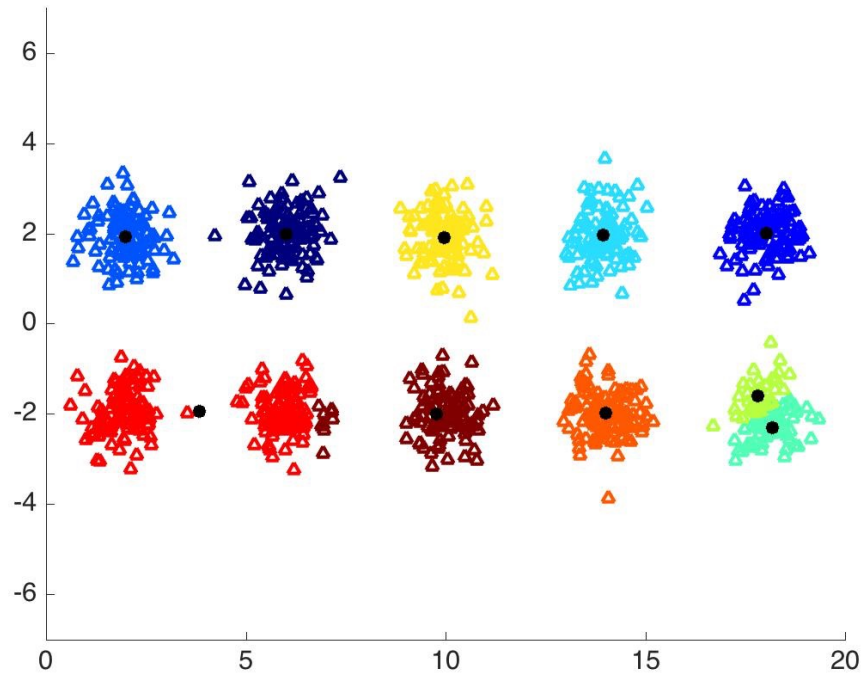
WC(C)=1304

Métodos de partición, K-medias, selección inicial de centroides

- **Selección inicial de centroides:**
  - Correr K-medias con multiples selecciones aleatoria de centroids y seleccionar el resultado con mejor score (mínima distancia entre clusters)



WC(C)=491



WC(C)=1304

Métodos de partición, K-medias, Python

- El código de Python permite realizar esto en forma bastante sencilla
  #Apliquemos K-medias 5 veces y elijamos el mejor
  km=KMeans(n_clusters=3,n_init=5)
  km=km.fit(tempDF)

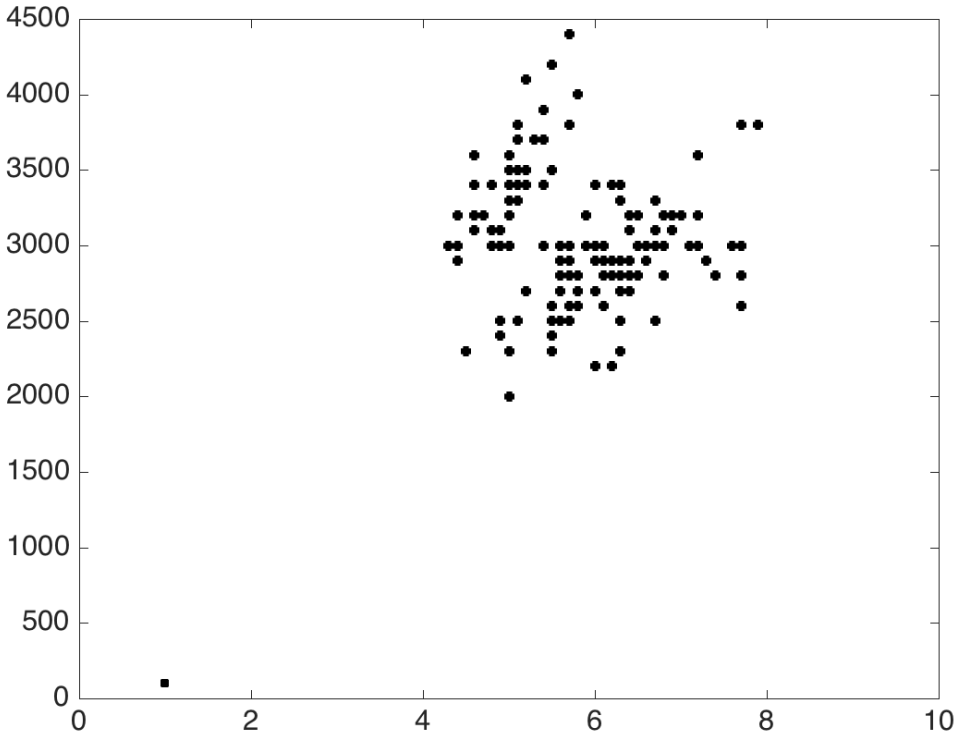
- #Otro parámetros del modelo K-means es el número máximo iteraciones (en caso de datos muy grandes)
  km=KMeans(n_clusters=3,n_init=5, max_iter=100)
  km=km.fit(tempDF)

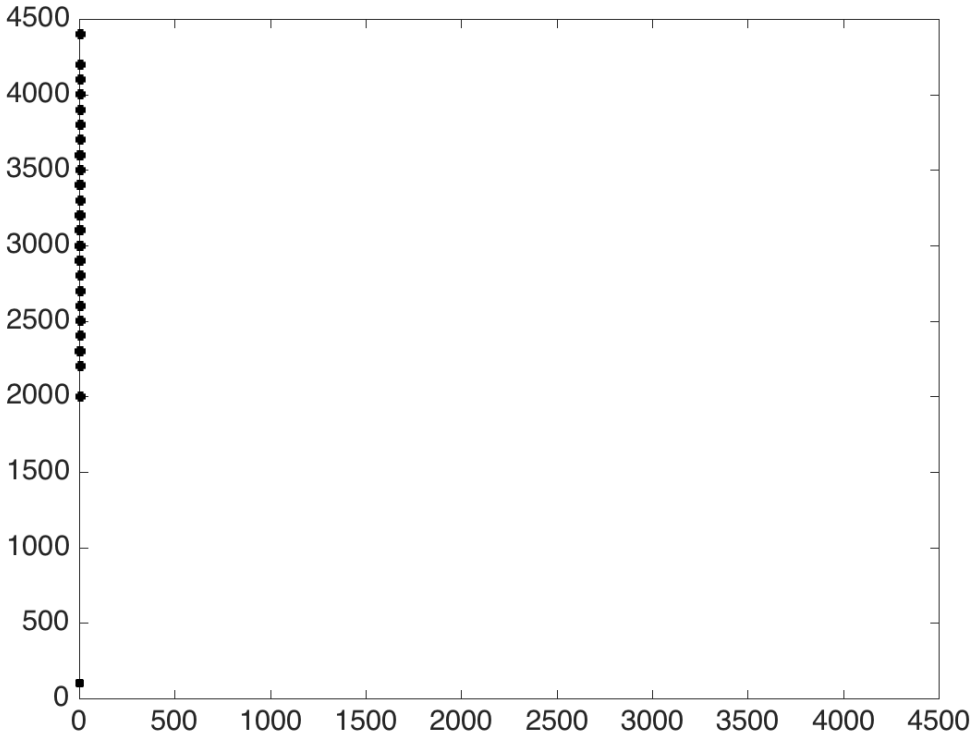
- #En caso que quiera replicar el resultado use random_state
  km=KMeans(n_clusters=3,n_init=5,max_iter=100,random_state=1)
  km=km.fit(tempDF)

Segmentación de datos
Clustering
Métodos de partición: **K-medias (K-means)**
Otros puntos a considerar

Métodos de partición, K-medias, pre procesamiento

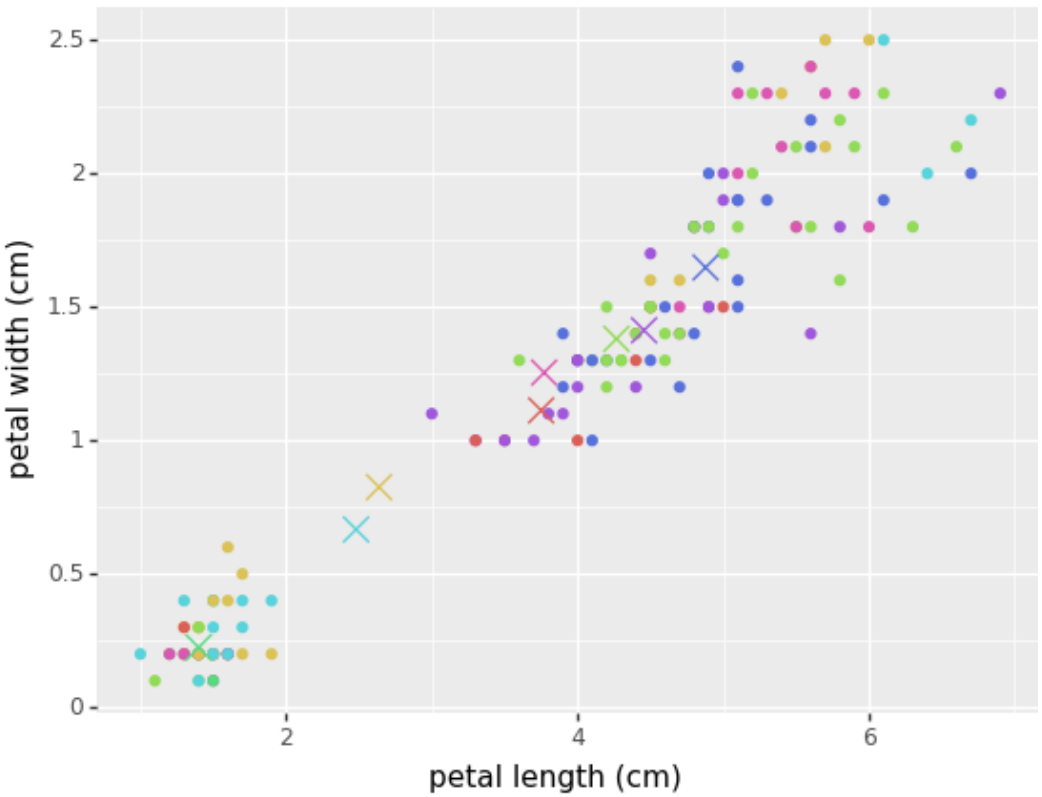
- Existen otros tipos de problemas en los algoritmos de clustering.
- ¿Cuántos clusters hay en el gráfico?
- ¿Cuántos clusters hay en el gráfico de la derecha?
- Sabiendo que son los mismos datos ¿cuál es el problema?

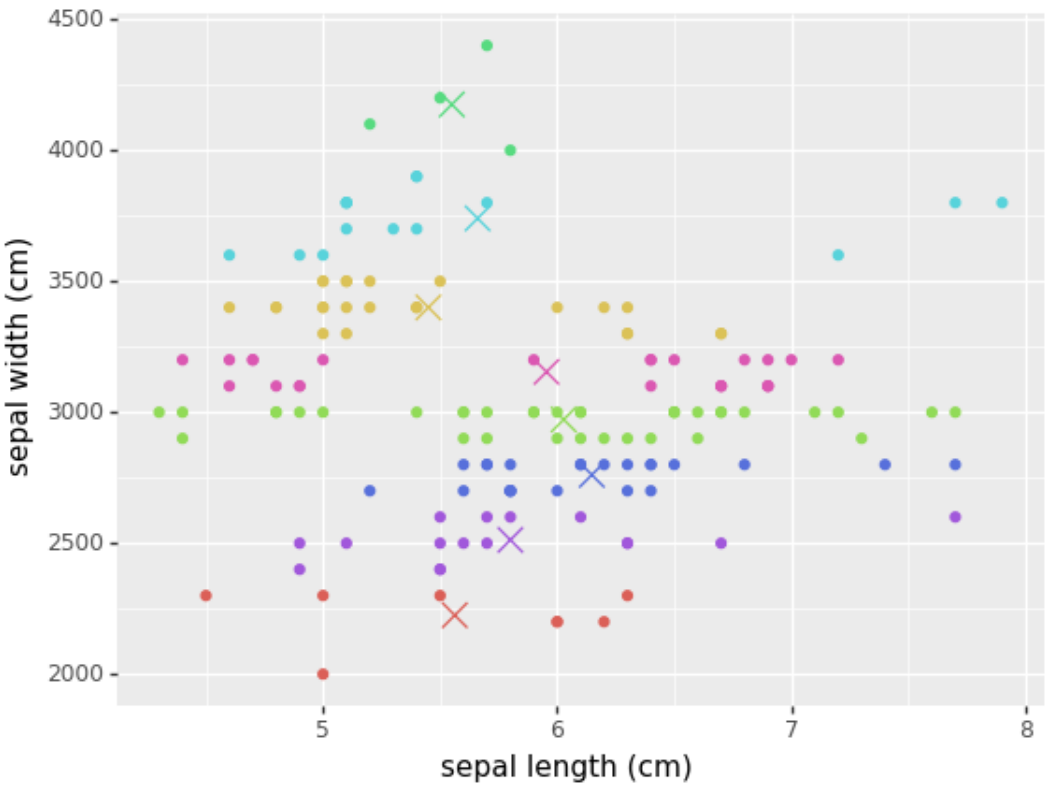




Métodos de partición, K-medias, pre procesamiento

- Este ejemplo extremo, se multiplicó una variable de los datos de iris por 1000.
- Esta variable pasa a ser la única importante en el proceso de clusterización.

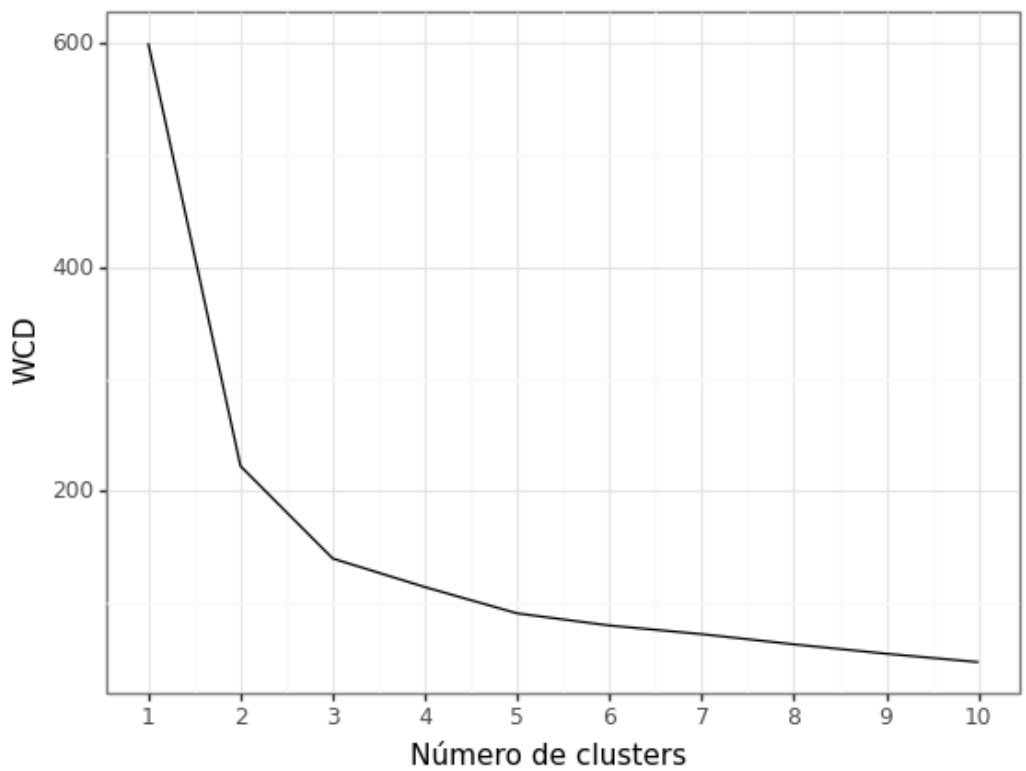




Métodos de partición, K-medias, pre procesamiento

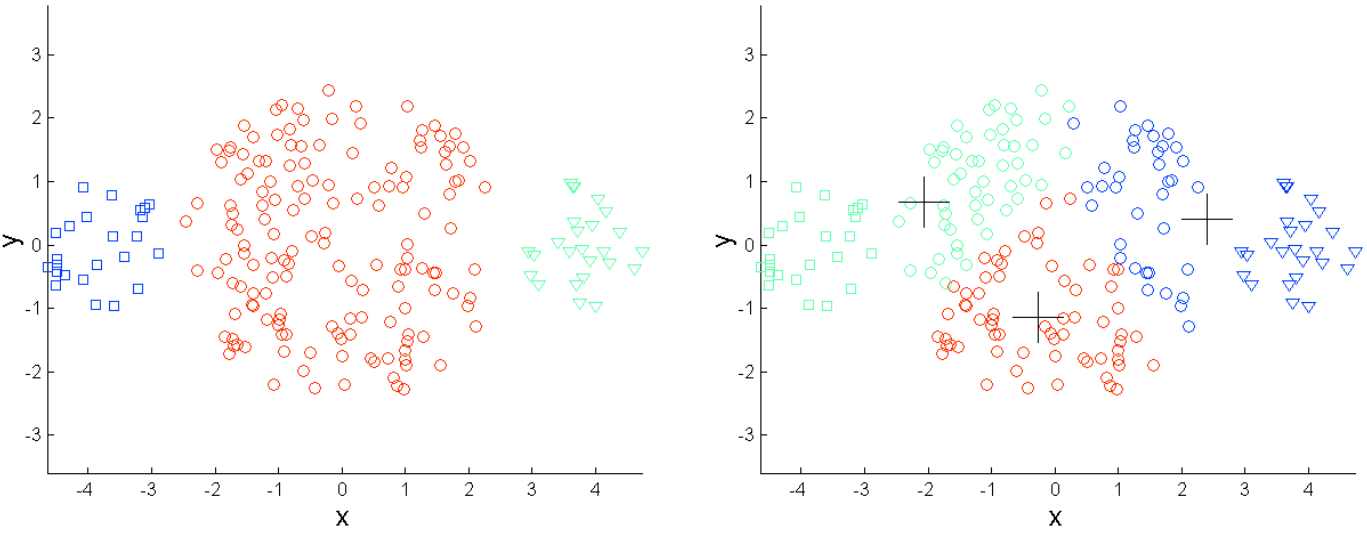
- Antes de aplicar los datos se recomienda pre-procesar los datos y escalarlos

```
from sklearn.preprocessing import StandardScaler
sse = []
scaler = StandardScaler() #estandarización (media 0 desviación estándar 1)
scaled_features = scaler.fit_transform(iris.data)
```



Métodos de partición, K-medias, post procesamiento

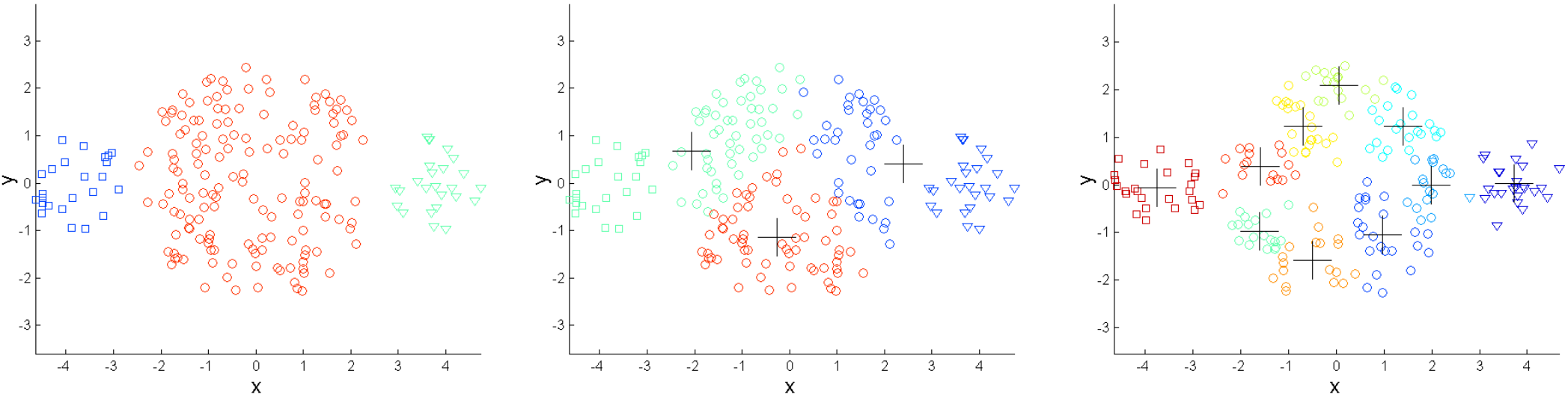
- ¿Como podríamos obtener el número de clusters deseados?



Clusters de tamaño distinto

Métodos de partición, K-medias, post procesamiento

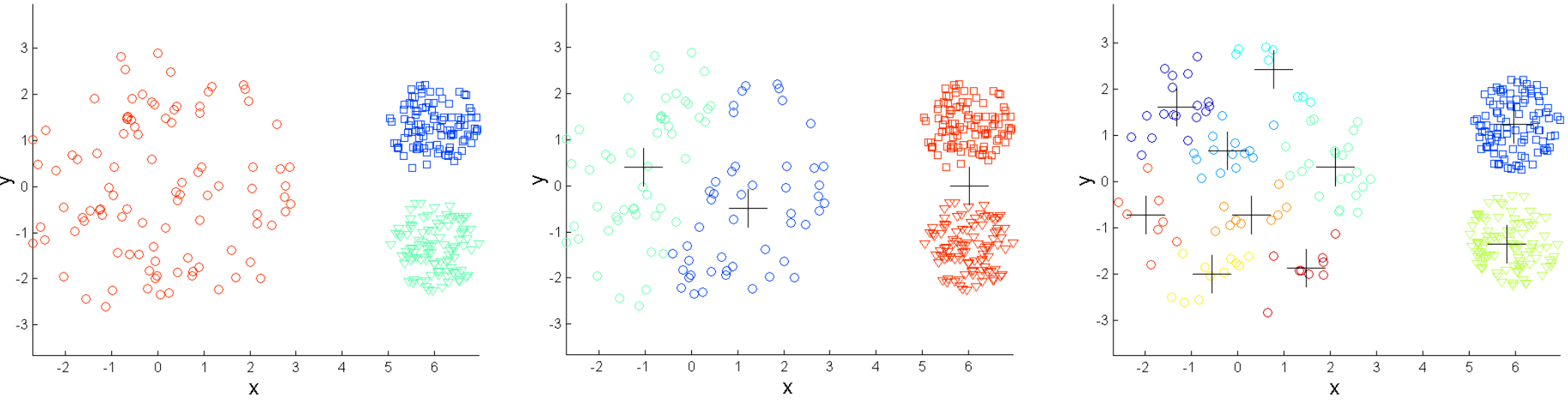
- K-medias tiene distintos problemas que se pueden solucionar generando más clusters de los necesarios y uniendo clusters con error bajo



Clusters de tamaño distinto

Métodos de partición, K-medias, post procesamiento

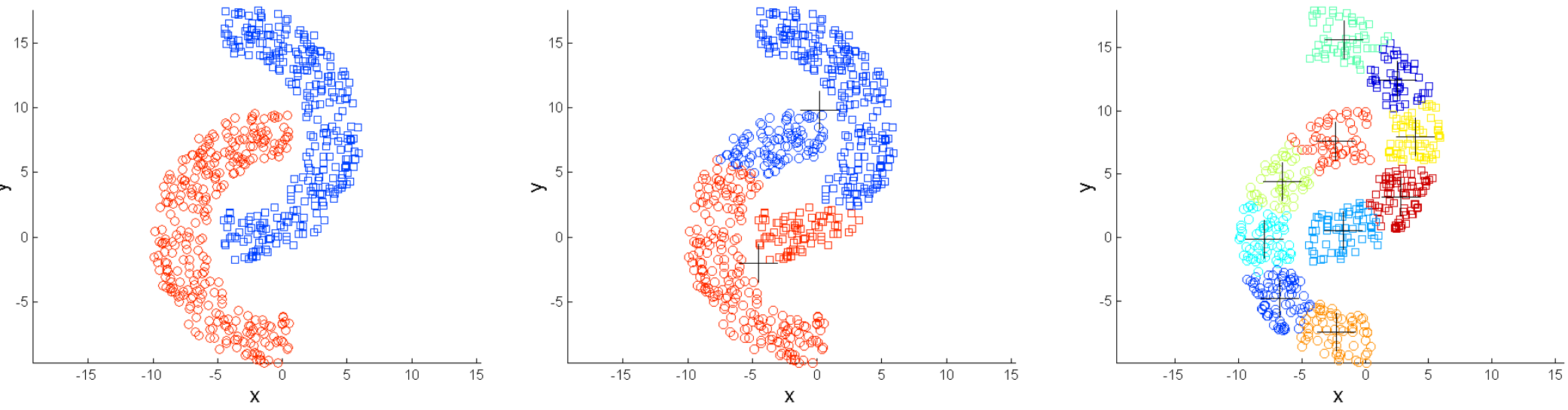
- K-medias tiene distintos problemas que se pueden solucionar generando más clusters de los necesarios y uniendo clusters con error bajo



Clusters con densidades distinta

Métodos de partición, K-medias, post procesamiento

- K-medias tiene distintos problemas que se pueden solucionar generando más clusters de los necesarios y uniendo clusters con error bajo



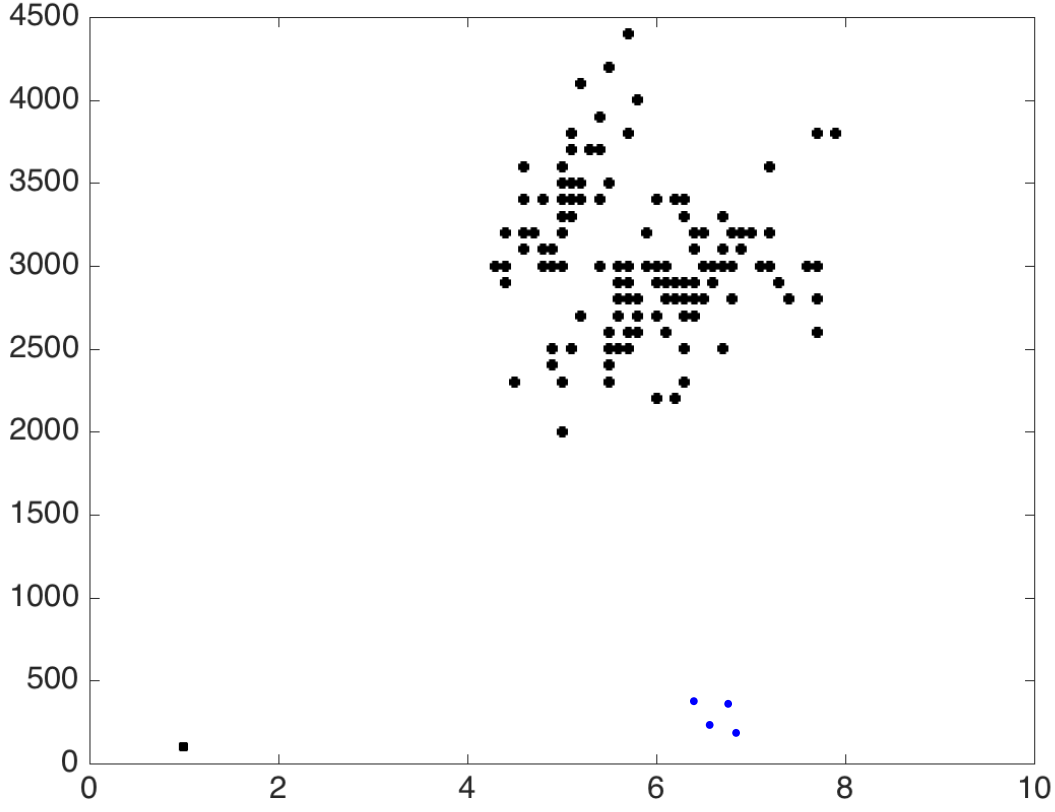
Cluster no globulares

Métodos de partición, K-medias, post procesamiento, unión

- ¿Por qué K-medias no puede encontrar los clusters anteriores?

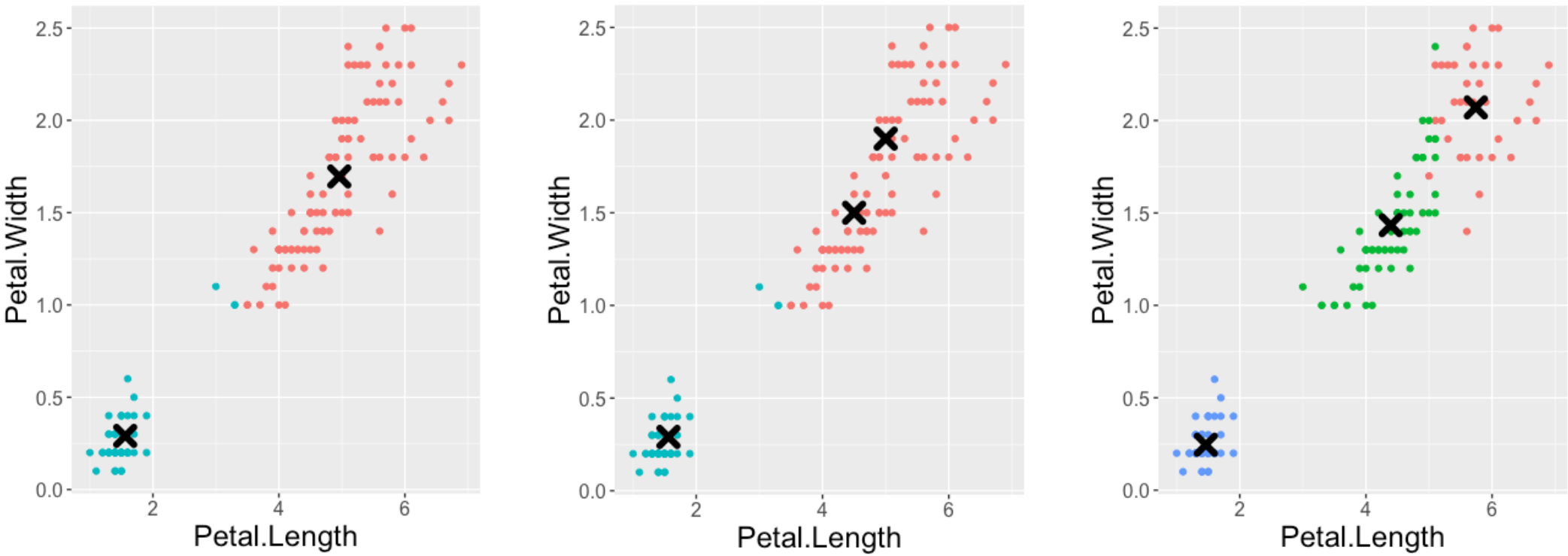
Métodos de partición, K-medias, post procesamiento, eliminación

- Clusters pequeños pueden representar set de datos atípicos los cuales no son interesantes para nuestros registros, **o son los más interesantes**.
- Con un modelo K-medias, estos datos se pueden eliminar de manera sencilla. Basta con eliminar los datos del dataset que pertenecen a ese dataset



Métodos de partición, K-medias, post procesamiento, separación

- Cluster que muestran un error muy alto pueden estar mal clusterizados.
- Separar un cluster es distinto a ejecutar el modelo con K+1 clusters.
- Al separar un cluster, tenemos que:
  1) Elegir dos nuevos centroides dentro del cluster a separar
  2) Ejecutar nuevamente el algoritmo de K-medias para ver la nueva convergencia



Métodos de partición, K-medias, aplicaciones

- **Color Quantization** es un proceso que reduce el número de colores distintos usados en una imagen, con la intención de que la nueva imagen sea visualmente cercana a la imagen original.

49633 colores

Métodos de partición, K-medias, aplicaciones

- **Image segmentation** segmentar la imagen basado en los colores que son similares.

49633 colores

Métodos de partición, K-medias, otros métodos

- ¿Es posible aplicar K-medias en datos nominales?

Segmentación de datos
Clustering
Métodos de partición: **k-modas (k-modes)**

Métodos de partición, K-modas, código en Python

- #instalando el paquete de K-modas
  import sys
  !{sys.executable} -m pip install kmodes
  from kmodes.kmodes import KModes

- Parámetros de KModes
  n_clusters: número de clusters.
  max_iter: máximo número de iteraciones.
  init: Método de inicialización ('Huang', 'Cao', 'random') default: 'Cao'
  * n_init: número de veces que K-modes se ejecuta

- Atributos de Kmodes
  cluster_centroids_: Centroide de los clusters
  labels_: Etiqueta de cada punto
  cost_: Suma de todos los puntos con respecto a los centroides
  n_iter_: número de iteraciones para el algoritmo

Métodos de partición, K-modas, código en Python

- #Cargando datos de clientes y preprocesamiento de datos
  bank = pd.read_csv('./data/bank.csv',sep=";")
  …

- #Transformando los datos a números (sklearn no trabaja con datos categóricos)
  from sklearn import preprocessing
  le = preprocessing.LabelEncoder()
  bank_cust = bank_cust.apply(le.fit_transform)

- #Ejecutando K-modas con 2 clusters
  km_cao = KModes(n_clusters=2, init = "Cao", n_init = 1, verbose=1)
  km_cao = km_cao.fit(bank_cust)
  print("Cluster 0",sum(km_cao.labels_==0))
  print("Cluster 1",sum(km_cao.labels_==1))
  Cluster 0: 3267
  Cluster 1: 1254

Métodos de partición, K-modas, código en Python

- #Veamos los centroides (no es eficiente)

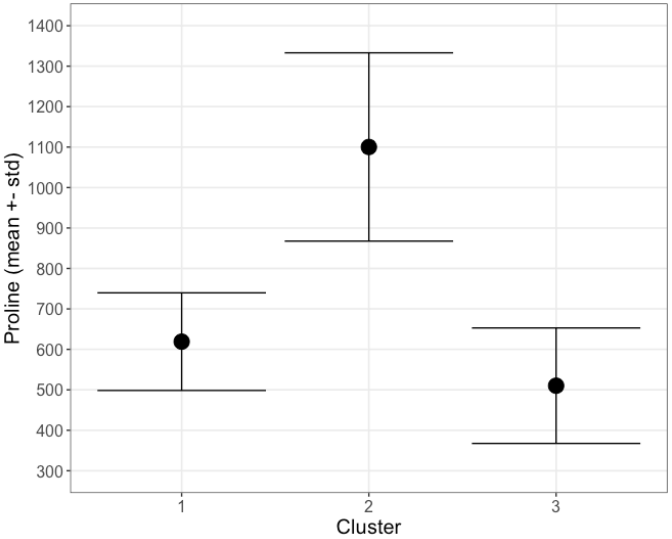
```
print(bank_cust.columns)
for i in range(km_cao.n_clusters):
    for j in range(len(km_cao.cluster_centroids_[i,])):
        le.fit(bank_cust_orig.iloc[:,j])
        print(list(le.inverse_transform([km_cao.cluster_centroids_[i,j]])),
                " ",end="")
    print(“")
```

- ['job', 'marital', 'education', 'default', 'housing', 'loan', ‘contact', 'month', 'poutcome', 'edad'],
  ['blue-collar'] ['married'] ['secondary'] ['no'] ['yes'] ['no'] ['cellular'] ['may'] ['unknown'] ['30-40']
  ['management'] ['single'] ['tertiary'] ['no'] ['no'] ['no'] ['cellular'] ['aug'] ['unknown'] ['50-60']

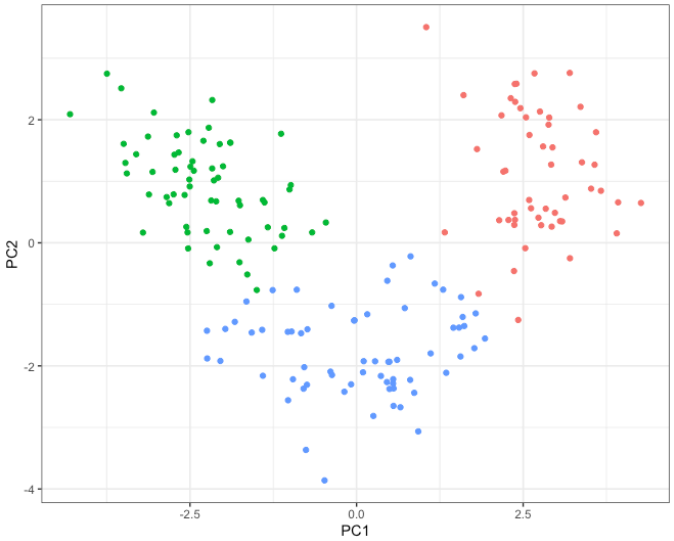
Machine Learning
Clustering
Interpretación

Interpretación

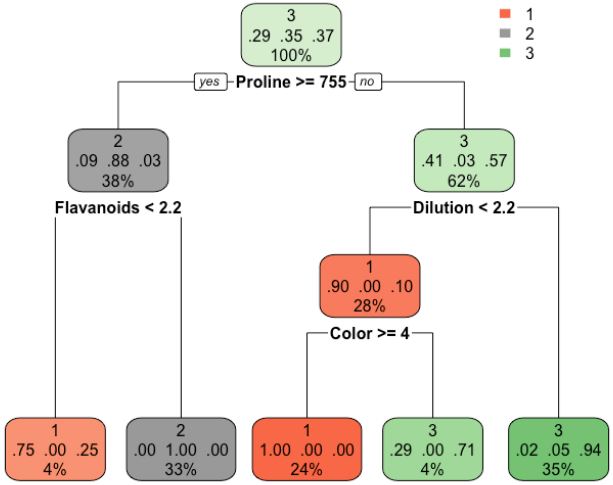
- Interpretar un cluster es tan importante como crear los clusters. Este es el proceso que nos genera información.
- Desafortunadamente, la interpretación de un cluster es una mezcla entre matemáticas e intuición (gran parte).
- Veremos 3 métodos para interpretar clusters:



Análisis de centros



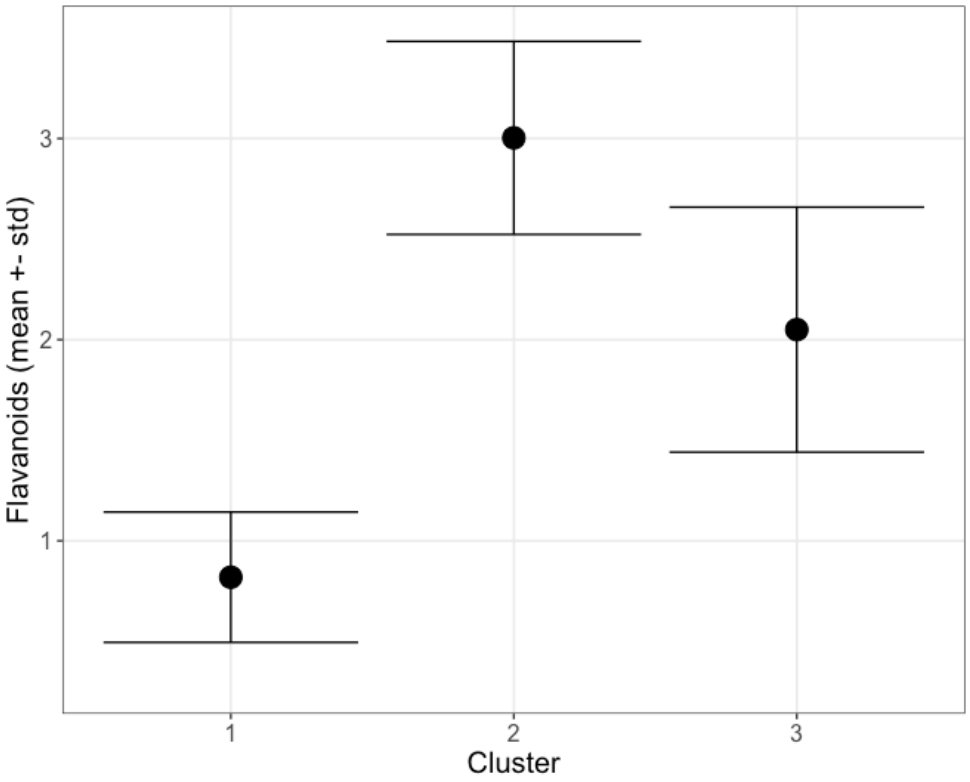
PCA



Modelos de clasificación

Interpretación, análisis de centros, ejemplo

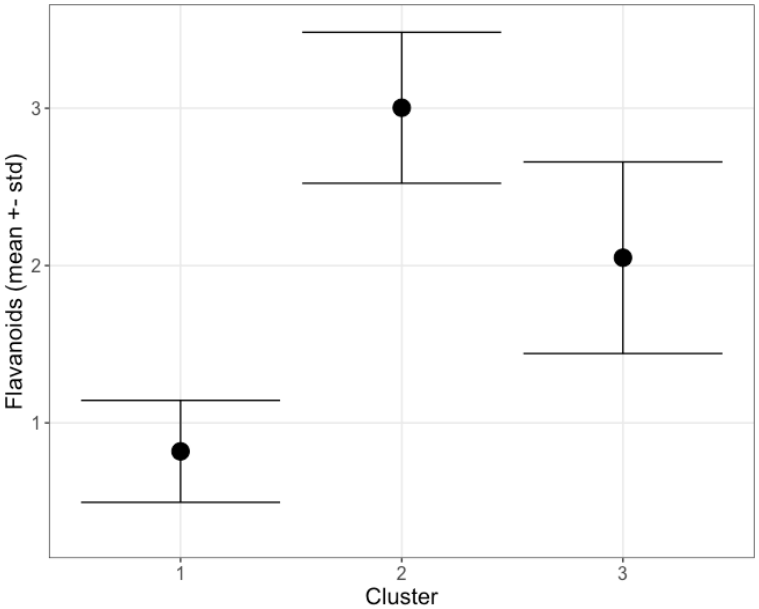
- Dada una clusterización, se analizan los centros y desviación estándar de los cluster obtenidos, verificando si existen diferencias significativas entre ellos.

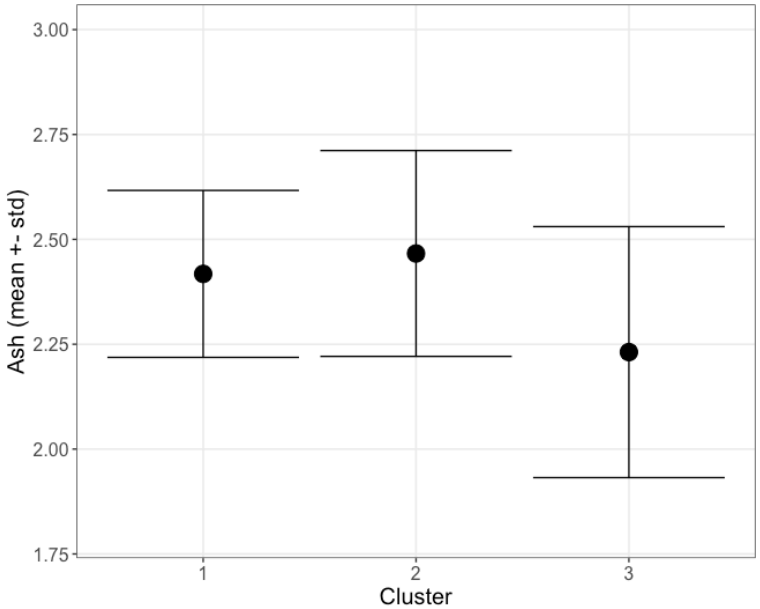


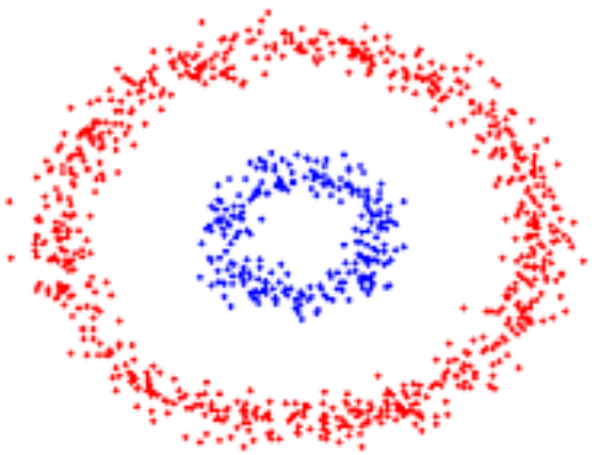
- Cluster 2 se caracteriza por tener puntos con Proline > 800.
- La variable Ash no es relevante en ningún cluster.
- Cluster 3 se caracteriza por tener Flavanoides cercanos a 2.0.
- Cluster 1 se caracteriza por tener bajos Flavanoides, cercanos a 0.9.

Interpretación, análisis de centros, ventajas y desventajas

- Permite determinar claramente las diferencias entre cada cluster.
- Es necesario analizar cada variable por separado.
- La mejor visualización es una tabla, pero es complejo de entender para personas no relacionadas al tema.
- Algunos métodos pueden tener centroides similares y modelar datos completamente diferentes (dataset tipo dona).

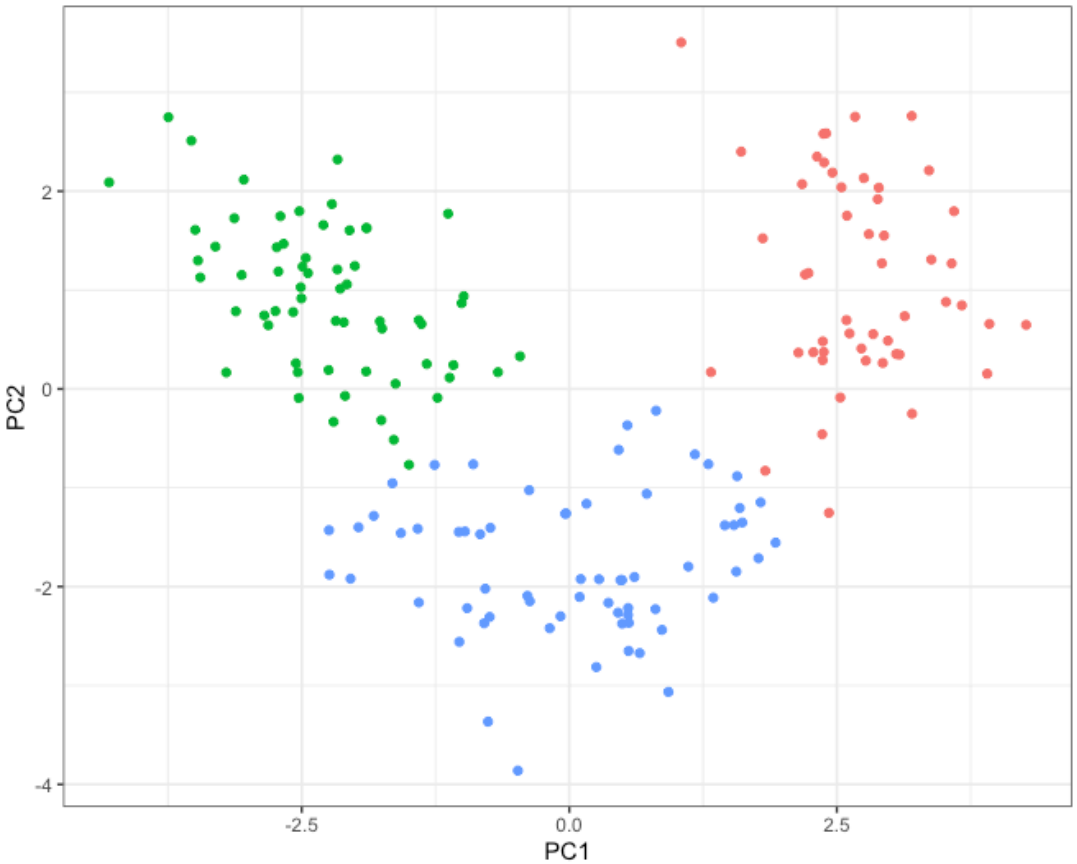


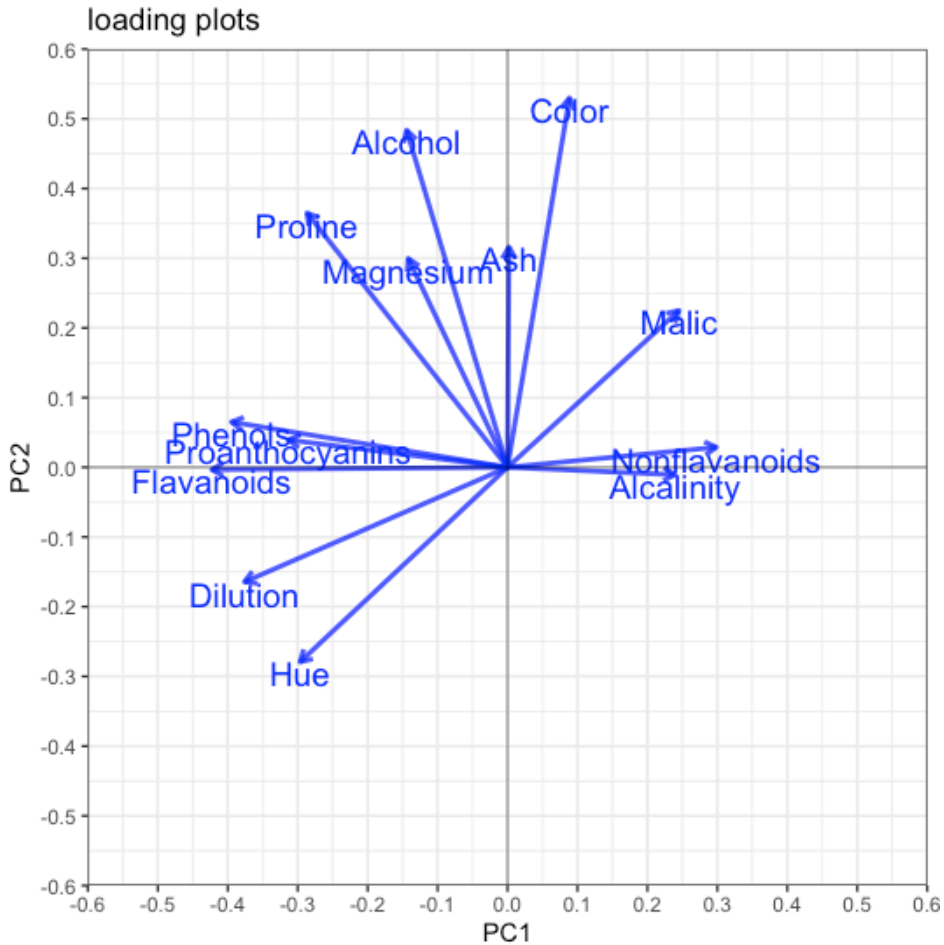




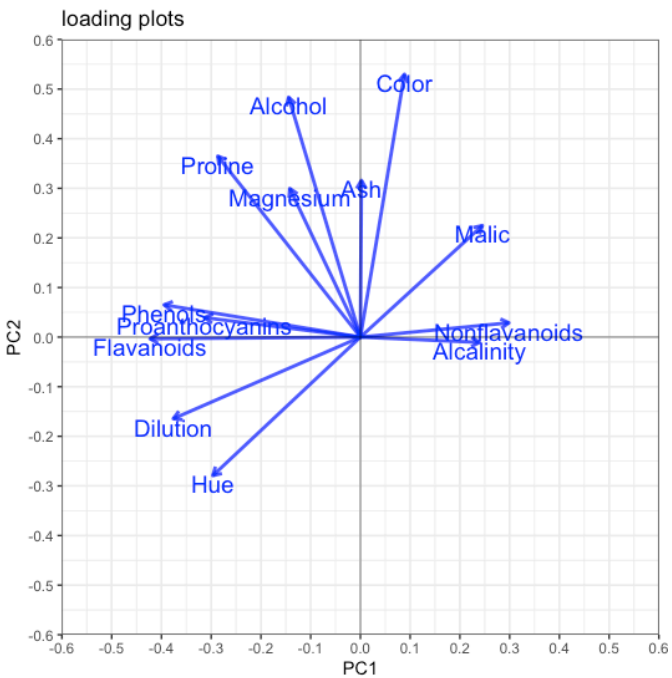
Interpretación, PCA

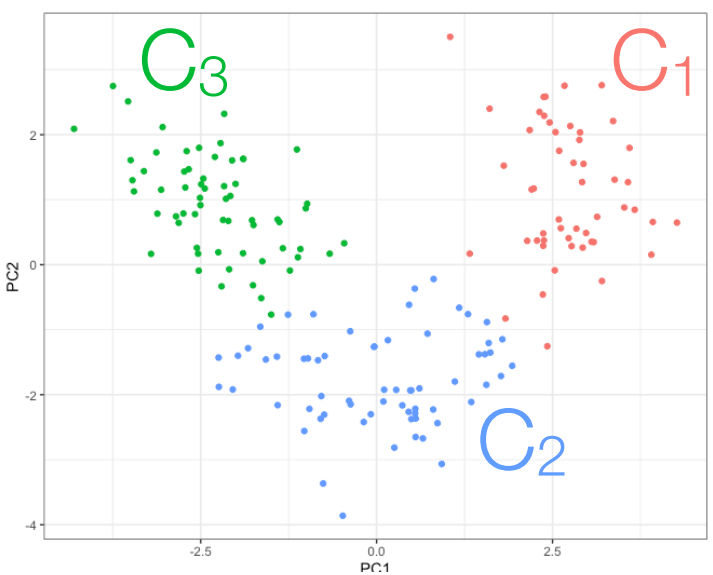
- Dada una clusterización, se aplica PCA a los datos originales y se analiza en forma visual y matemática las dos primeras componentes.
- El análisis se hace en conjunto a los loadings plots.





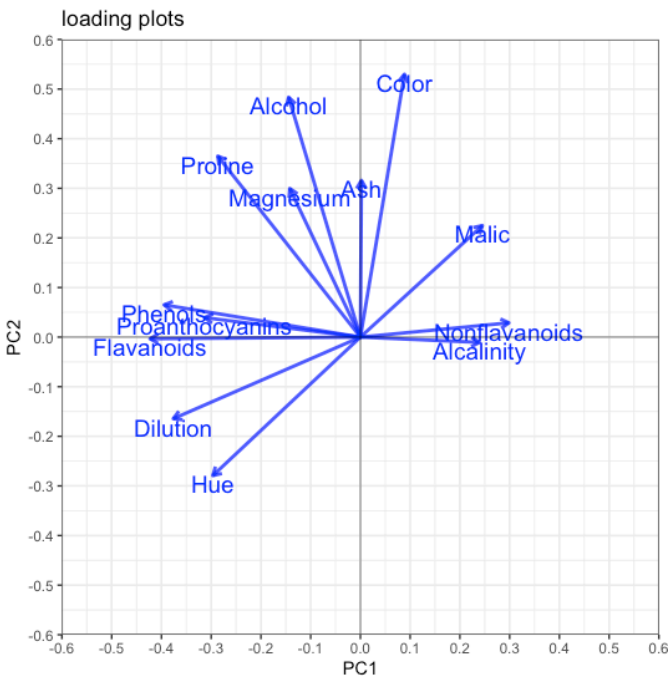
Interpretación, PCA, cluster C_1

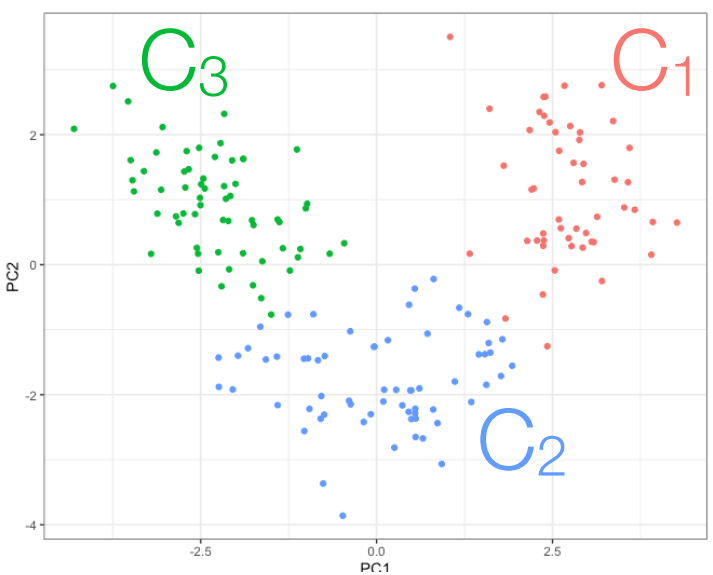




- C_1 tiene altos valores PC1, posibles explicaciones:
  - Altos valores de nonFlavanoids, Alcalinity, y Malic
  - Bajos valores de Proline, Phenols, Proanthocyanins, Flavanoids, Dilution, y Hue.
- C_1 tiene altos valores PC2, posibles explicaciones:
  - Altos valores de Color, Alcohol, Ash, Proline (contradicción), Magnesium, y Malic
  - Bajos valores de Dilution, y Hue.
- El proceso se repite para cada cluster

Interpretación, PCA, ventajas y desventajas





- Las mismas ventajas de PCA, incluyendo la visualización de todas las dimensiones en dos dimensiones.
- Es necesario entender y aplicar PCA.
- Es complejo de interpretar y se puede llegar a contradicciones.
- Mientras las primeras componentes son las más importantes, las otras componentes de igual manera podrían reflejar algo útil.

PCA, código en Python

- PCA(n_components=None)

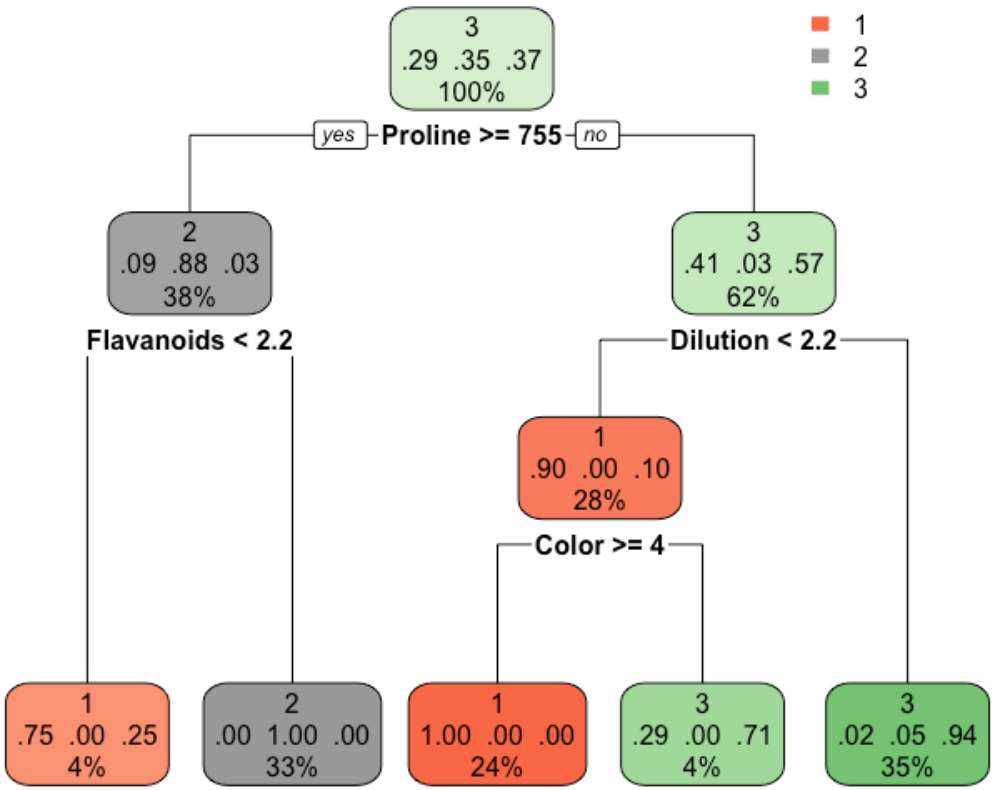
- Parámetros de PCA
  n_components: número de componentes.

- Atributos de PCA
  components_: vectores de transformación.
  explained_variance_: varianza explicada en cada componente.
  explained_variance_ratio_: % varianza explicada en cada componente.
  n_components_: número de componentes.
  n_features_: número de características.
  n_samples_: número de muestras.

- Funciones
  fit(X): Ajusta el modelo con X.
  fit_transform(X): Ajusta y aplica aplica el modelo a X.
  transform(X): Aplica la reducción de dimensionalidad a X.

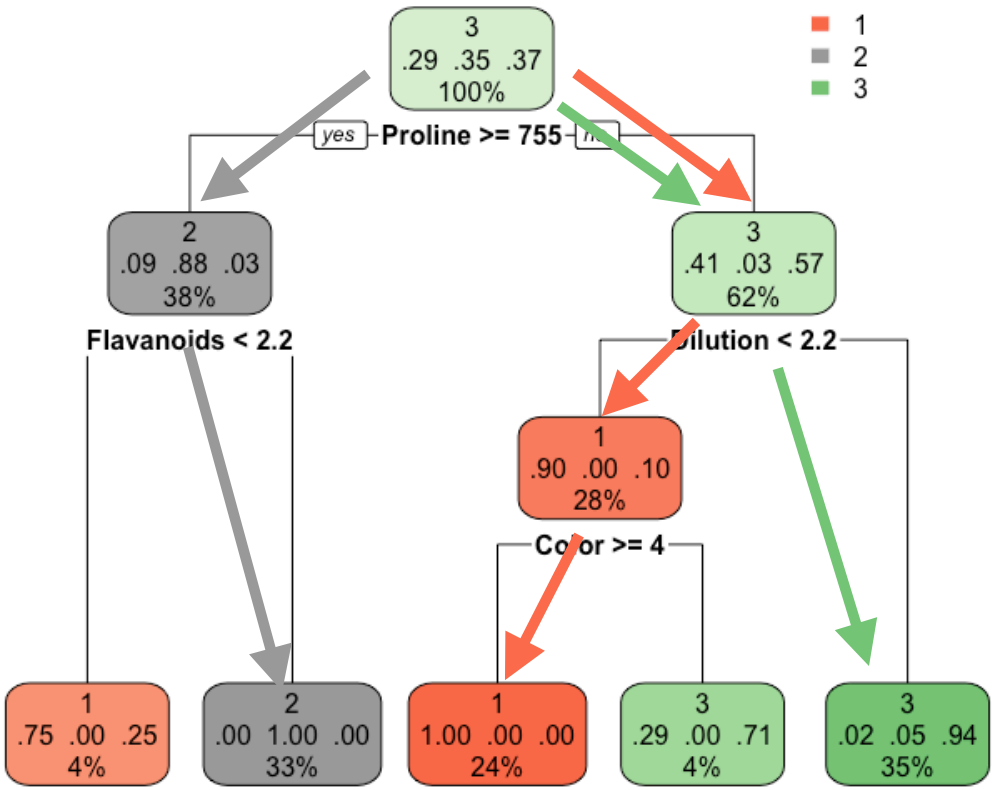
Interpretación, modelos de clasificación

- Dada una clusterización, se aplica un modelo de clasificación “informativo” para separar los datos en los clusters generados.
- El modelo obtenido se interpreta, obteniendo las variables más importantes.



Interpretación, modelos de clasificación

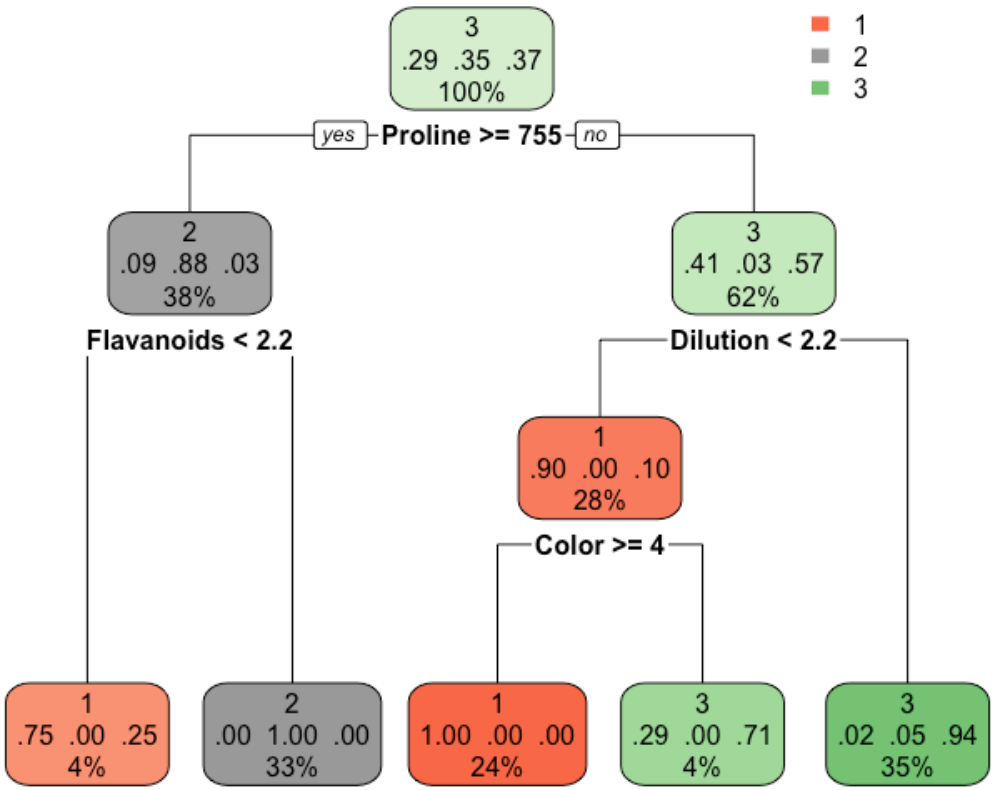
- Dada una clusterización, se aplica un modelo de clasificación “informativo” para separar los datos en los clusters generados.
- El modelo obtenido se interpreta, obteniendo las variables más importantes.



- Cluster 2 se caracteriza por puntos con Proline sobre 755 y Flavanoides mayores a 2.2.
- Cluster 3 se caracteriza por puntos con Proline bajo 755 y Dilution mayor a 2.2.
- Cluster 1 se caracteriza por puntos con Proline bajo 755, Dilution menor a 2.2, y Color mayor o igual a 4.

Interpretación, modelos de clasificación, ventajas y desventajas

- Dada una clusterización, se aplica un modelo de clasificación “informativo” para separar los datos en los clusters generados.
- El modelo obtenido se interpreta, obteniendo las variables más importantes.



- Los árboles de decisión son fácilmente interpretables (lo veremos más adelante).
- Debemos aplicar otro modelo “interpretativo” para la interpretación.
- Otras variables no son consideradas, sin embargo, también podrían ayudar a caracterizar a los clusters.