# Clustering

Dr. Rahul Kottath

# Clustering (Unsupervised Learning)

**Given:** Examples: $< x_1 , x_2 , \ldots x_n >$

**Find:** A natural clustering (grouping) of the data

**Example Applications:**

Identify similar energy use customer profiles

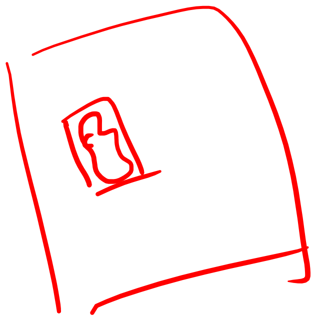
<**x**> = time series of energy usage

Identify anomalies in user behavior for computer security

<**x**> = sequences of user commands

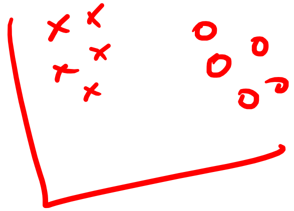
# Why cluster?

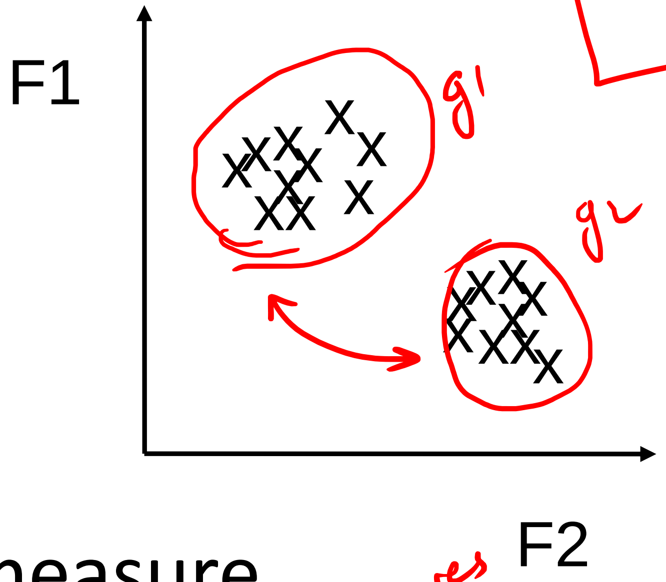
- Labeling is expensive
- Gain insight into the structure of the data
- Find prototypes in the data



# Goal of Clustering

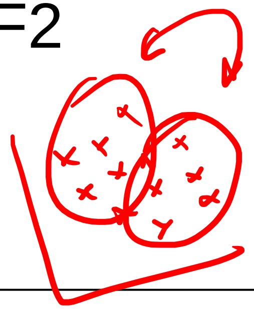
- Given a set of data points, each described by a set of attributes, find clusters such that:
  - Inter-cluster similarity is maximized
  - Intra-cluster similarity is minimized



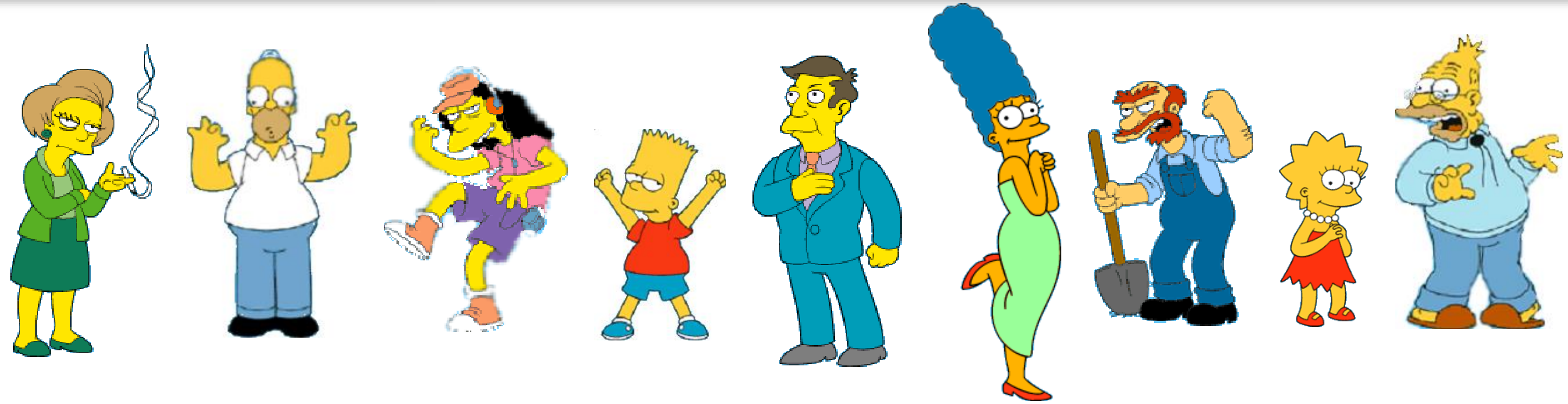



- Requires the definition of a similarity measure

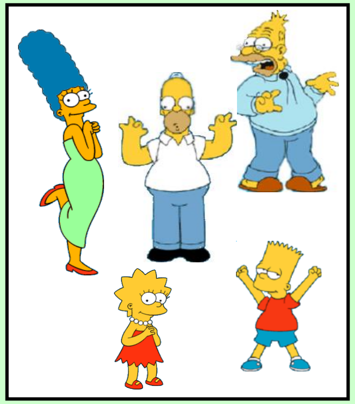
distance measures



# What is a natural grouping of these objects?

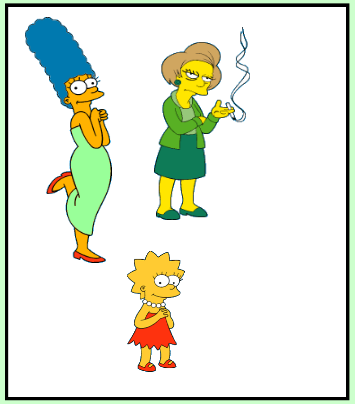


## Clustering is subjective



Simpson's Family

School Employees



Females

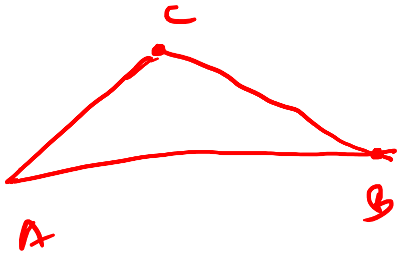
Males

# What is Similarity?

Similarity is hard to define, but…
"*We know it when we see it*"

# What properties should a distance measure have?

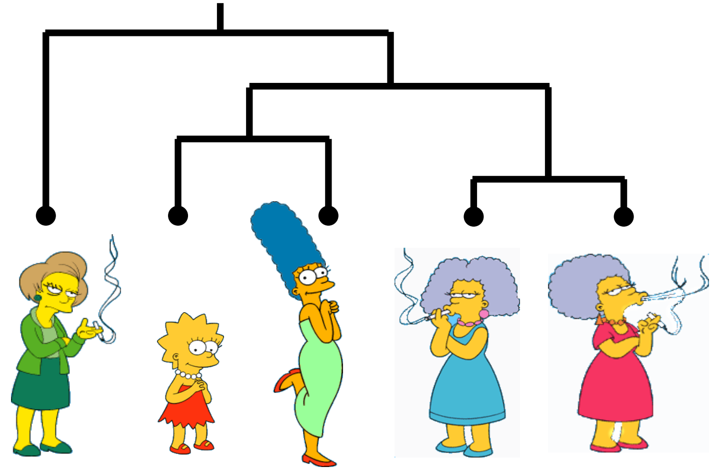
- $D(A,B) = D(B,A)$ — *Symmetry*
- $D(A,A) = 0$ — *Constancy of Self-Similarity*
- $D(A,B) = 0$ iif $A = B$ — *Positivity (Separation)*
- $D(A,B) \leq D(A,C) + D(B,C)$ — *Triangular Inequality*



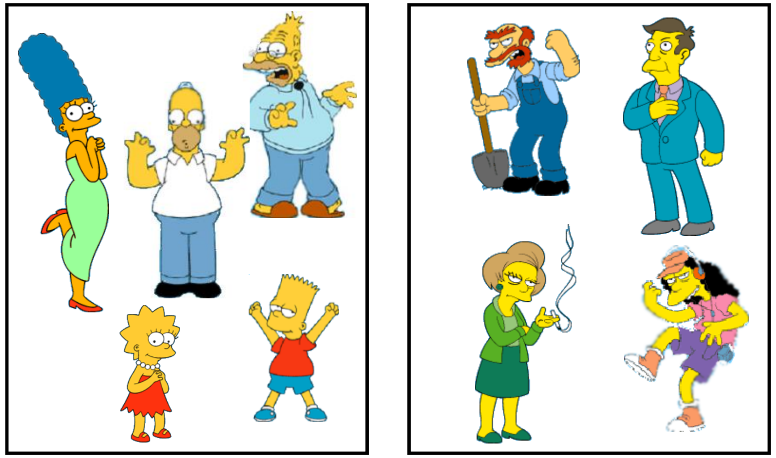
# Two Types of Clustering

- **Partitional algorithms:** Construct various partitions and then evaluate them by some criterion
- **Hierarchical algorithms:** Create a hierarchical decomposition of the set of objects using some criterion

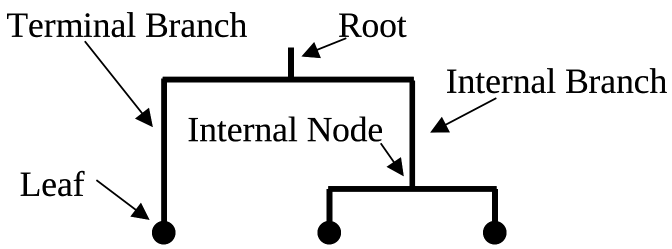
**Hierarchical**



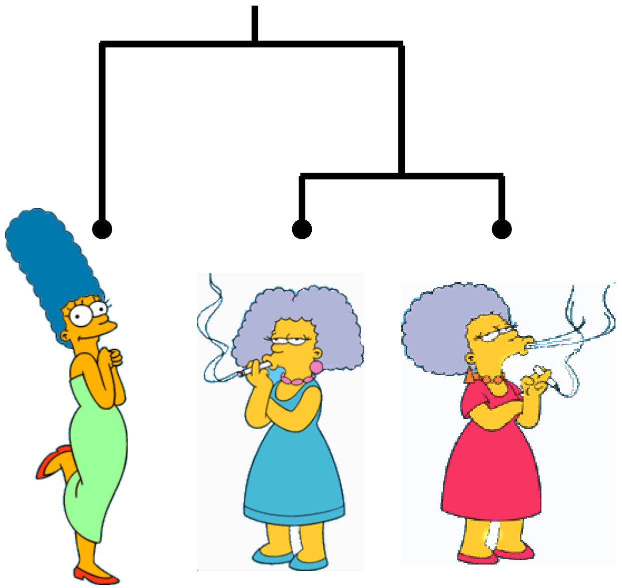
**Partitional**

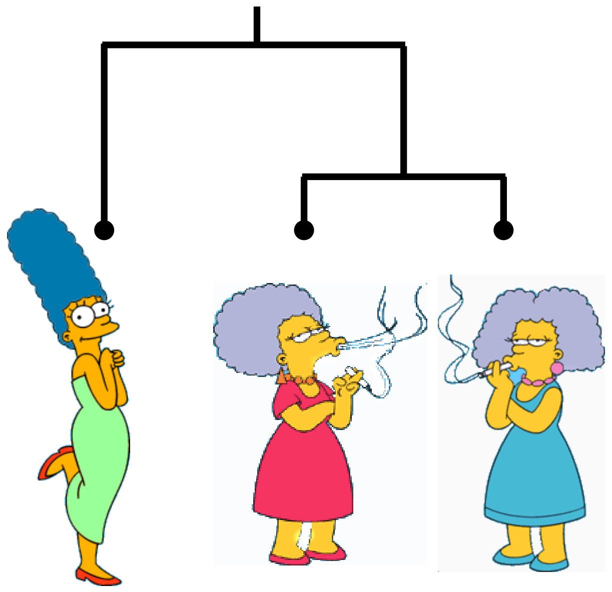


# Dendogram: A Useful Tool for Summarizing Similarity Measurements




The similarity between two objects in a dendrogram is represented as the height of the lowest internal node they share.



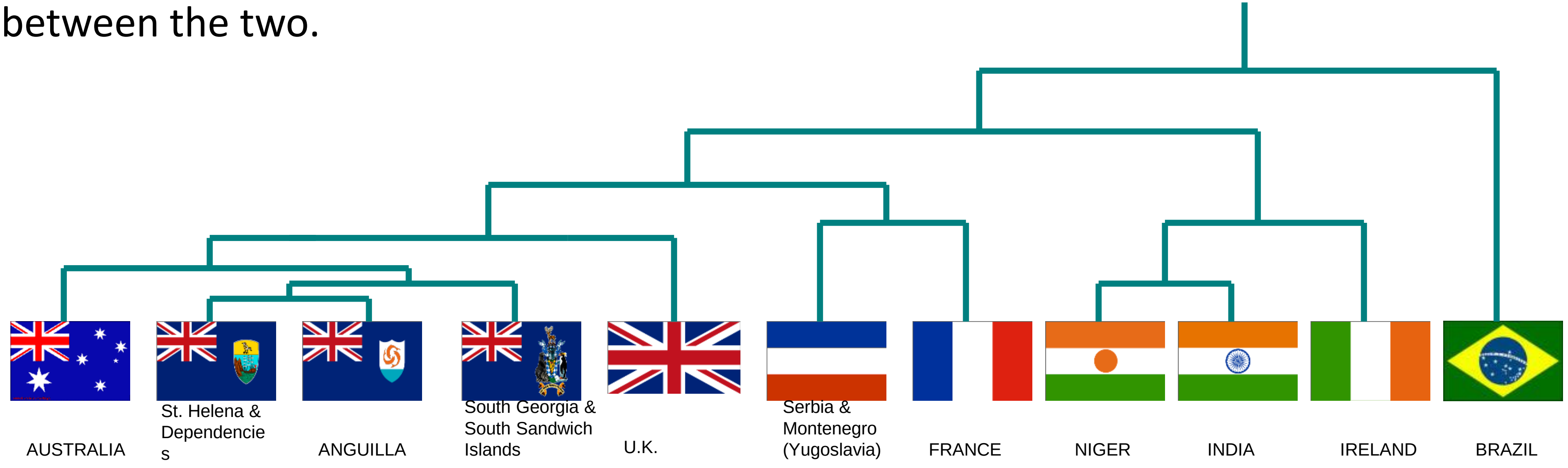


# Types of hierarchical clustering

- Agglomerative (bottom up) clustering: It builds the dendrogram (tree) from the bottom level, and
  - merges the most similar (or nearest) pair of clusters
  - stops when all the data points are merged into a single cluster (i.e., the root cluster).
- Divisive (top down) clustering: It starts with all data points in one cluster, the root.
  - Splits the root into a set of child clusters. Each child cluster is recursively divided further
  - stops when only singleton clusters of individual data points remain, i.e., each cluster with only a single point

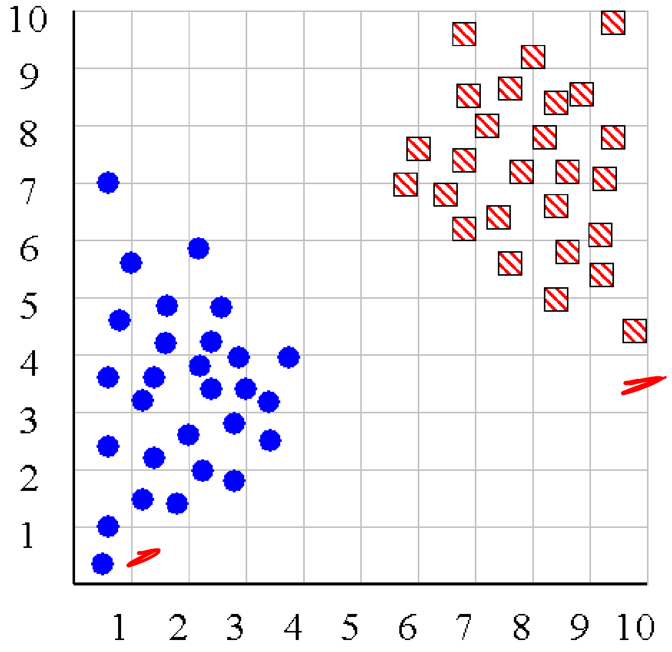
# Hierarchal clustering

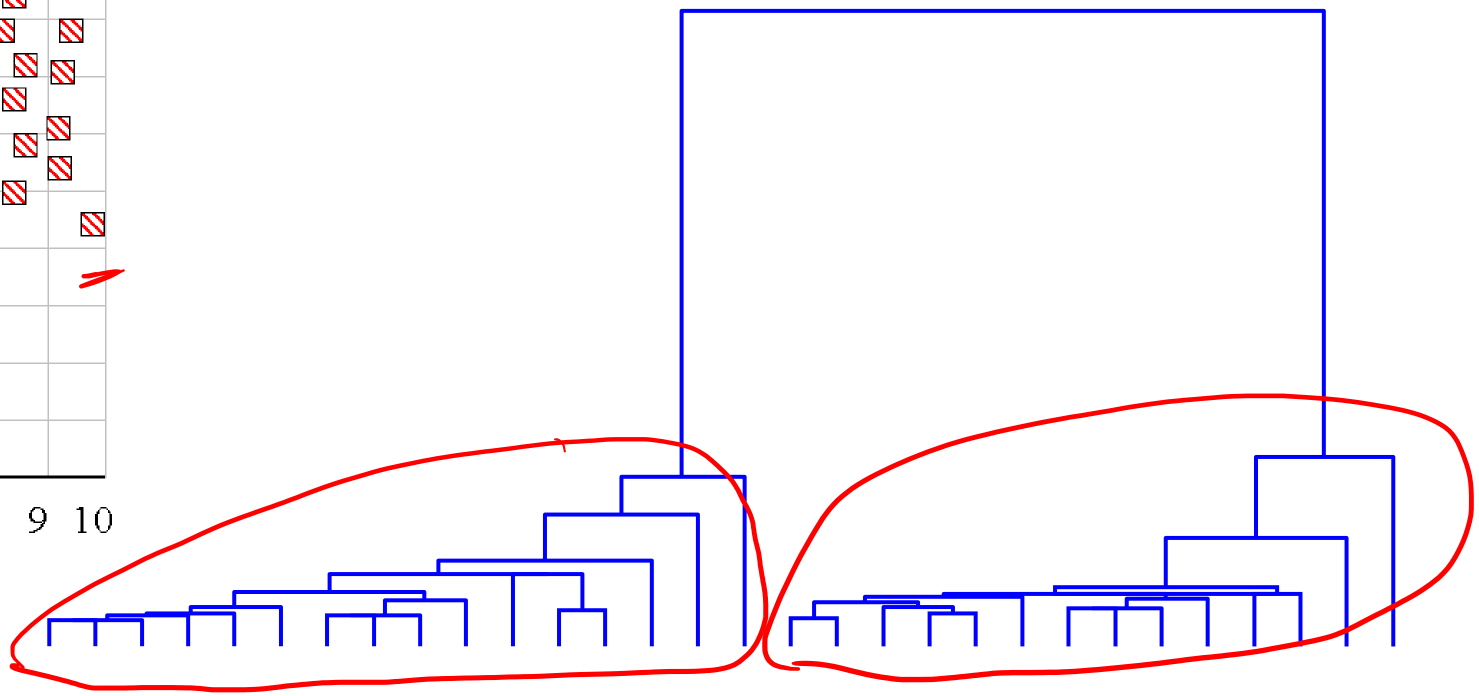
- Hierarchal clustering can sometimes show patterns that are meaningless or spurious
  - The tight grouping of Australia, Anguilla, St. Helena etc is meaningful; all these countries are former UK colonies
  - However the tight grouping of Niger and India is completely spurious; there is no connection between the two.



# Hierarchal clustering

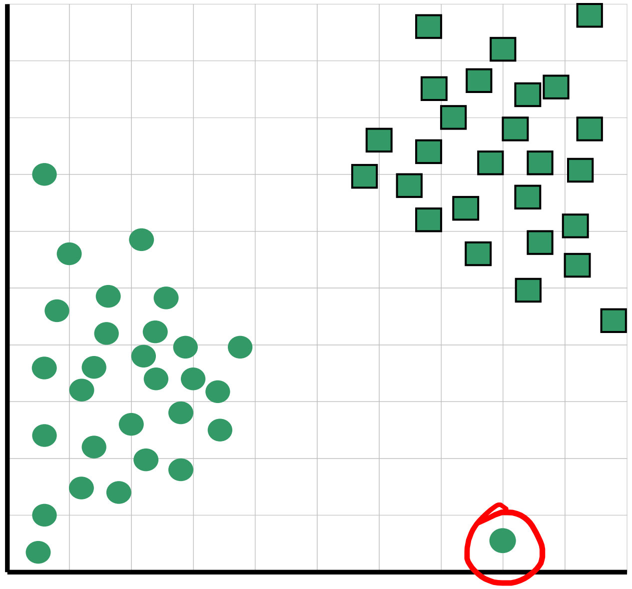
We can look at the dendrogram to determine the "correct" number of clusters.

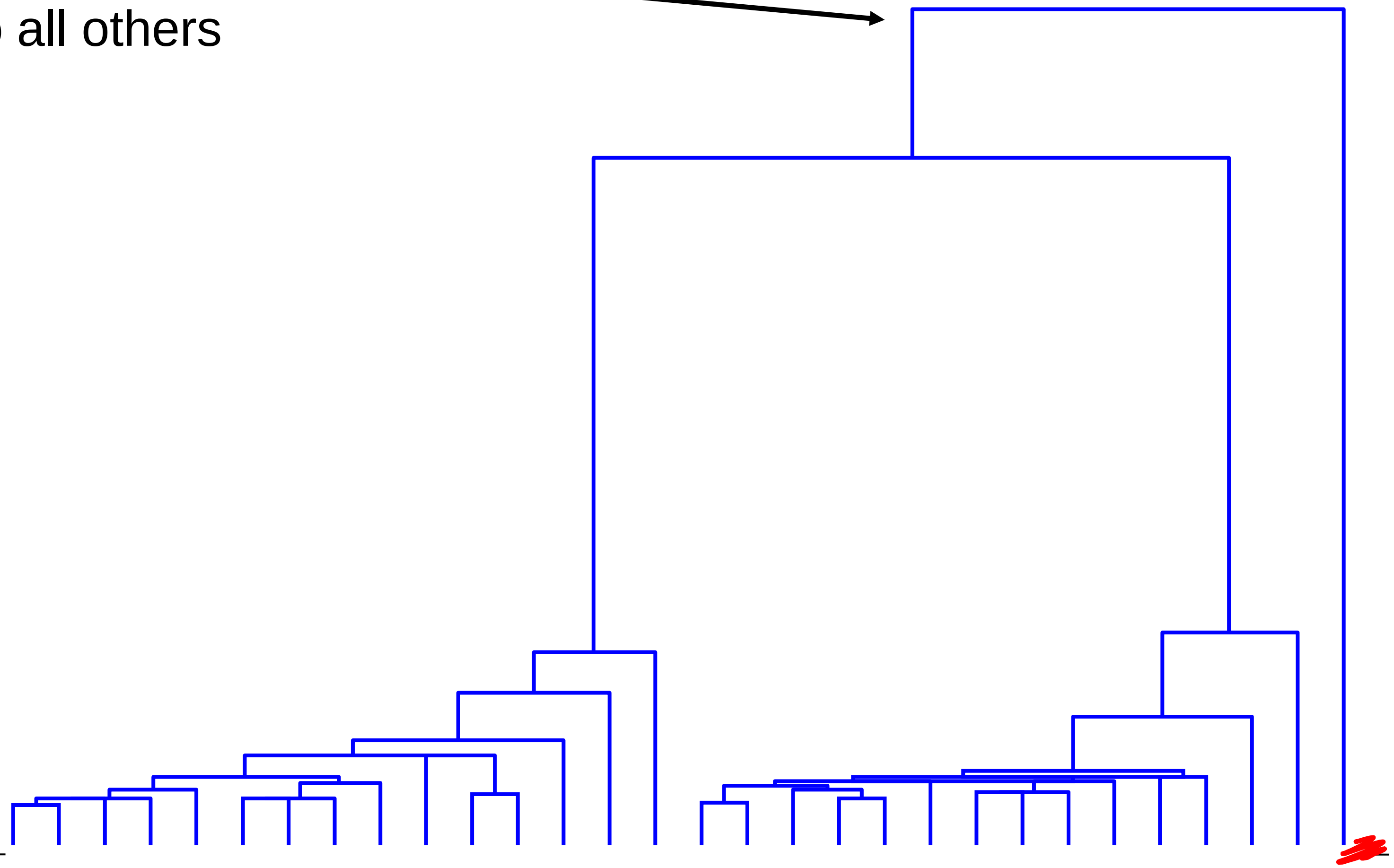


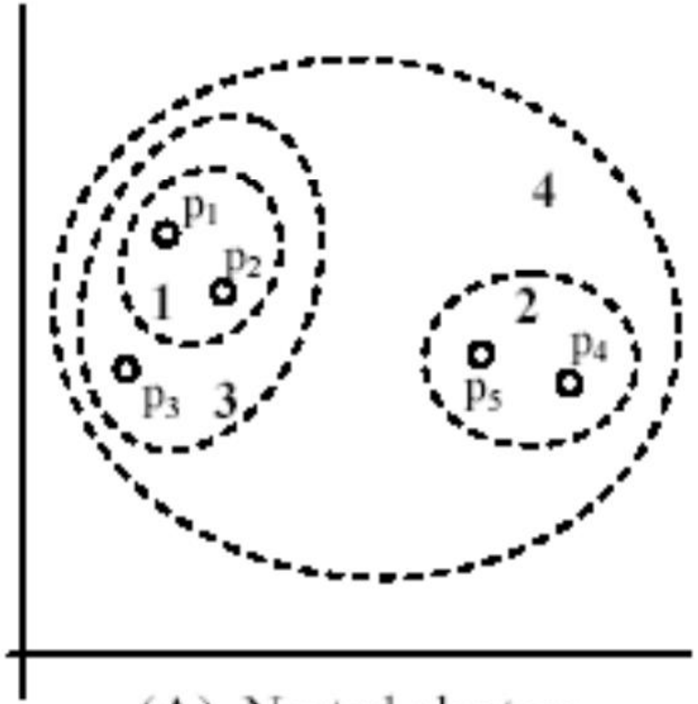
# One potential use of a dendrogram: detecting outliers

The single isolated branch is suggestive of a data point that is very different to all others

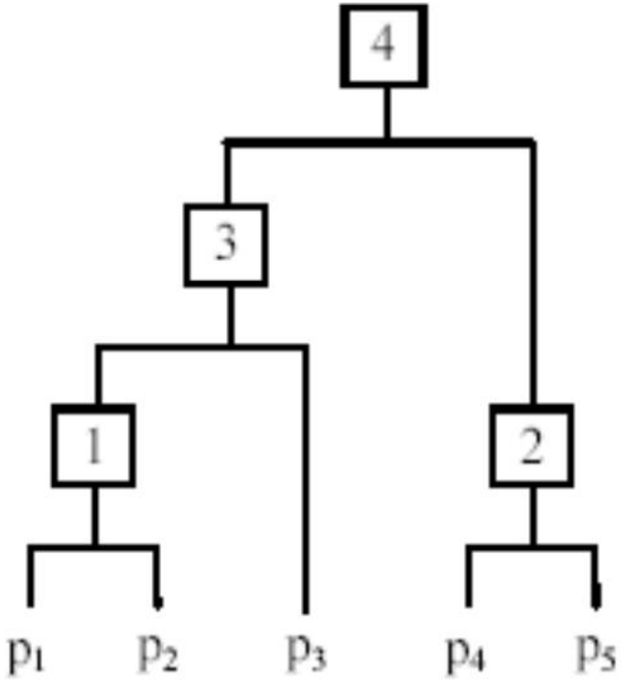




# An example: working of the algorithm



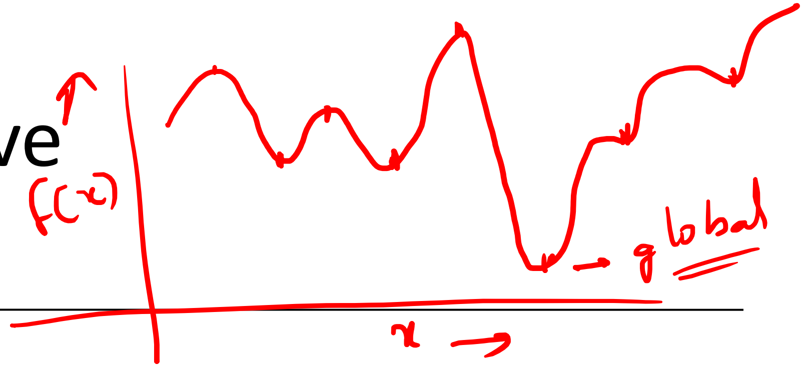

(A). Nested clusters




(B) Dendrogram

# Hierarchal Clustering Methods Summary

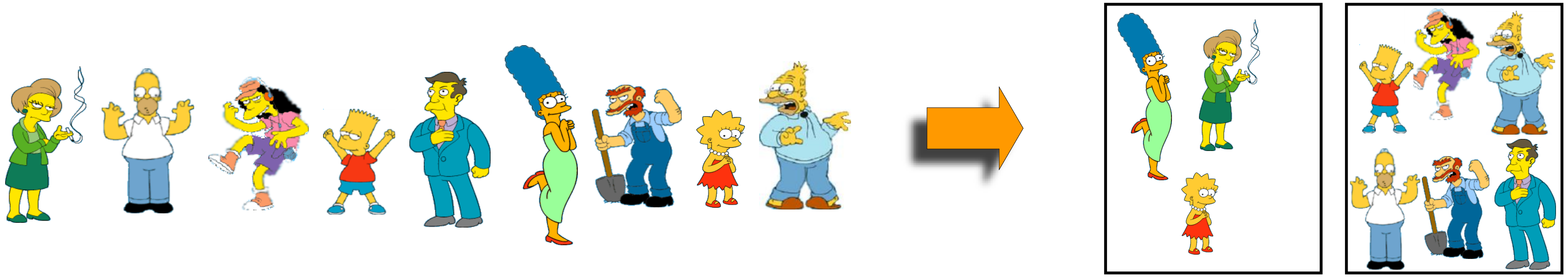
- No need to specify the number of clusters in advance
- Hierarchal nature maps nicely onto human intuition for some domains
- They do not scale well
- Like any heuristic search algorithms, local optima are a problem
- Interpretation of results is (very) subjective



# Partitional Clustering

data point

- Nonhierarchical, each instance is placed in exactly one of K non-overlapping clusters.
- Since only one set of clusters is output, the user normally has to input the desired number of clusters K.

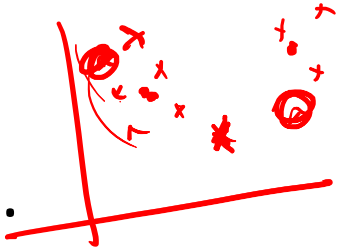


# K-Means Clustering

# The K-Means Clustering Method: for numerical attributes

Given k, the k-means algorithm is implemented in five steps:

1. Decide on a value for k.
2. Initialize the k cluster centers (randomly, if necessary).

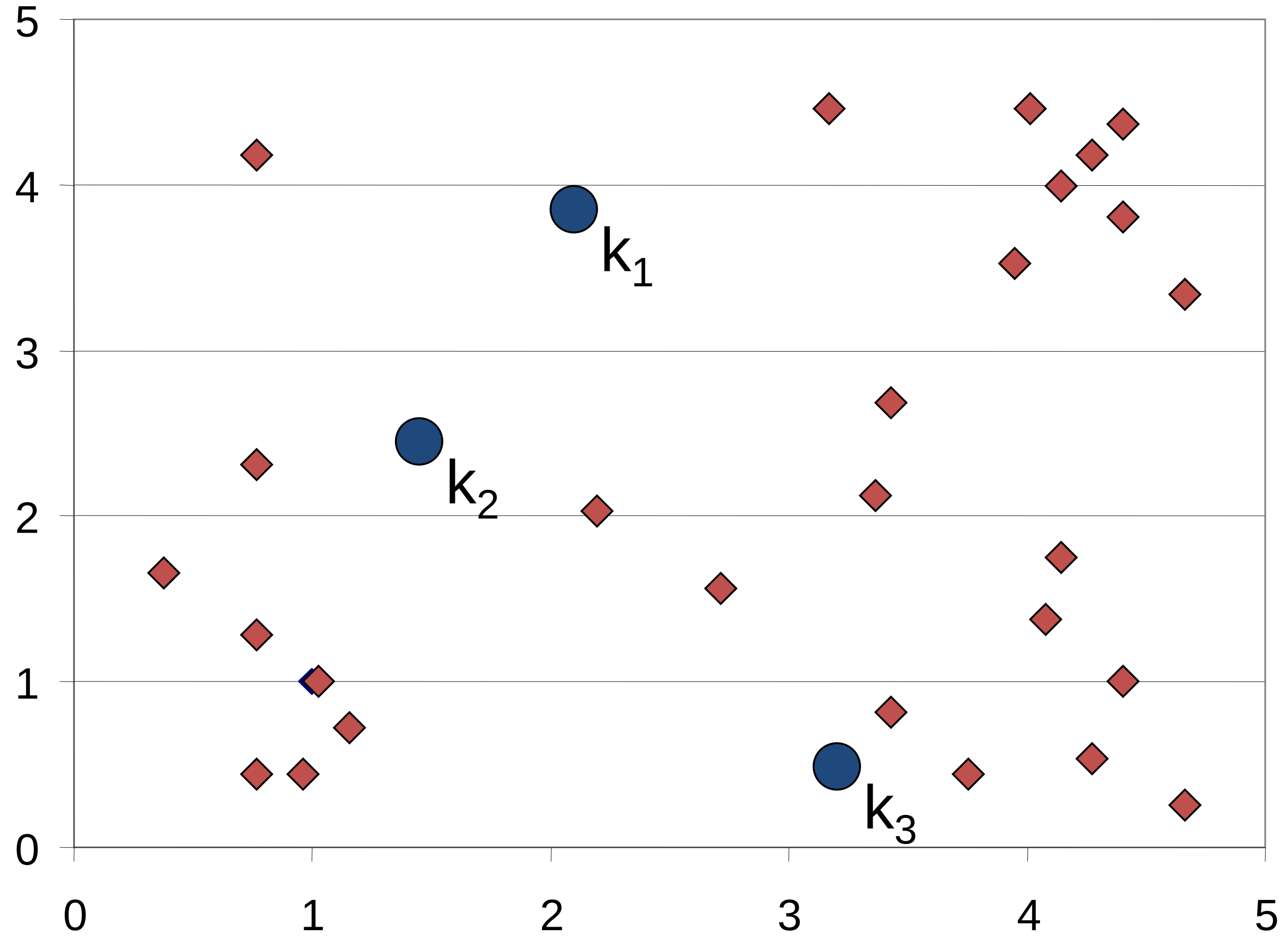


3. Decide the class memberships of the N objects by assigning them to the nearest cluster center.
4. Re-estimate the k cluster centers, by assuming the memberships found above are correct.
5. If none of the N objects changed membership in the last iteration, exit. Otherwise go to 3.

# K-means Clustering: Step 1

Algorithm: k-means,
Distance Metric: Euclidean Distance

Initialized k = 3



# K-means Clustering: Step 2

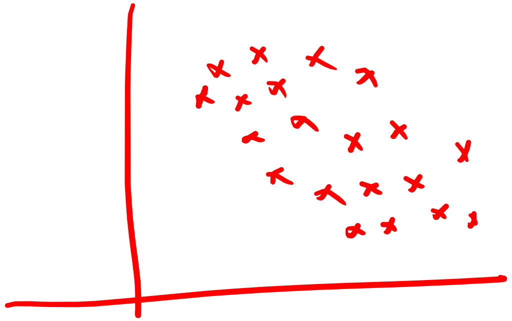
Algorithm: k-means,
Distance Metric: Euclidean Distance

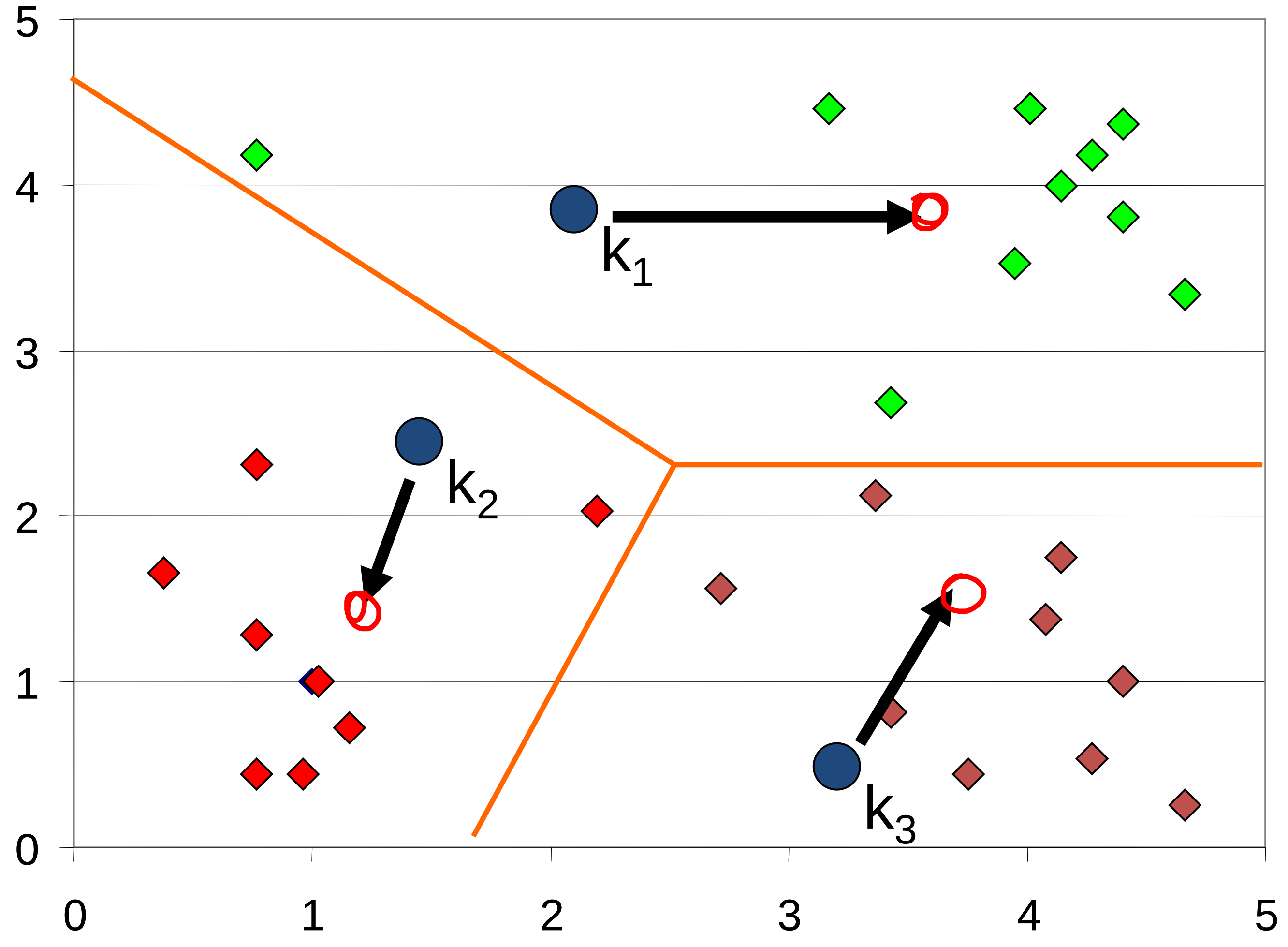
100 → 10

19 33

97 67

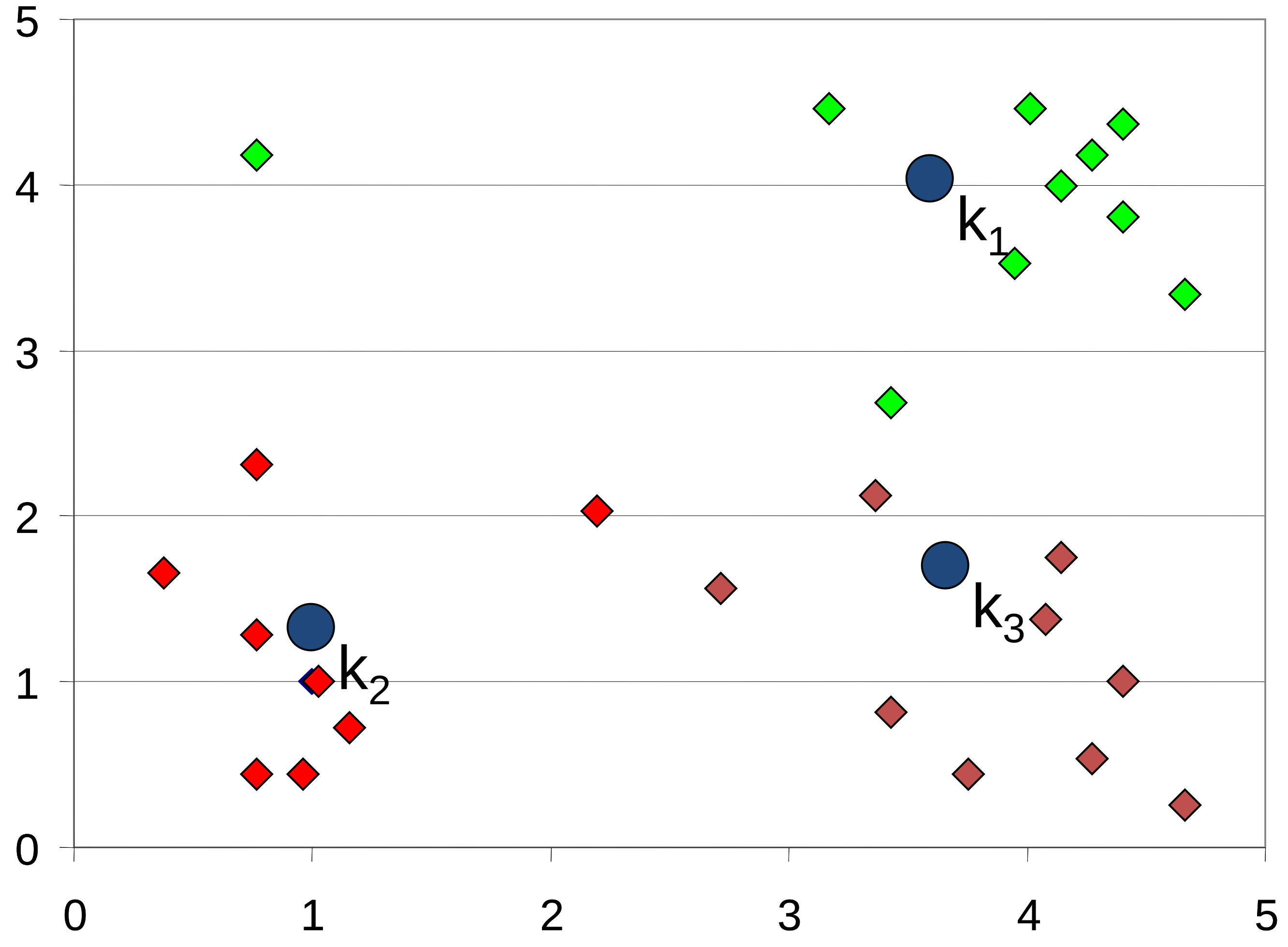
23





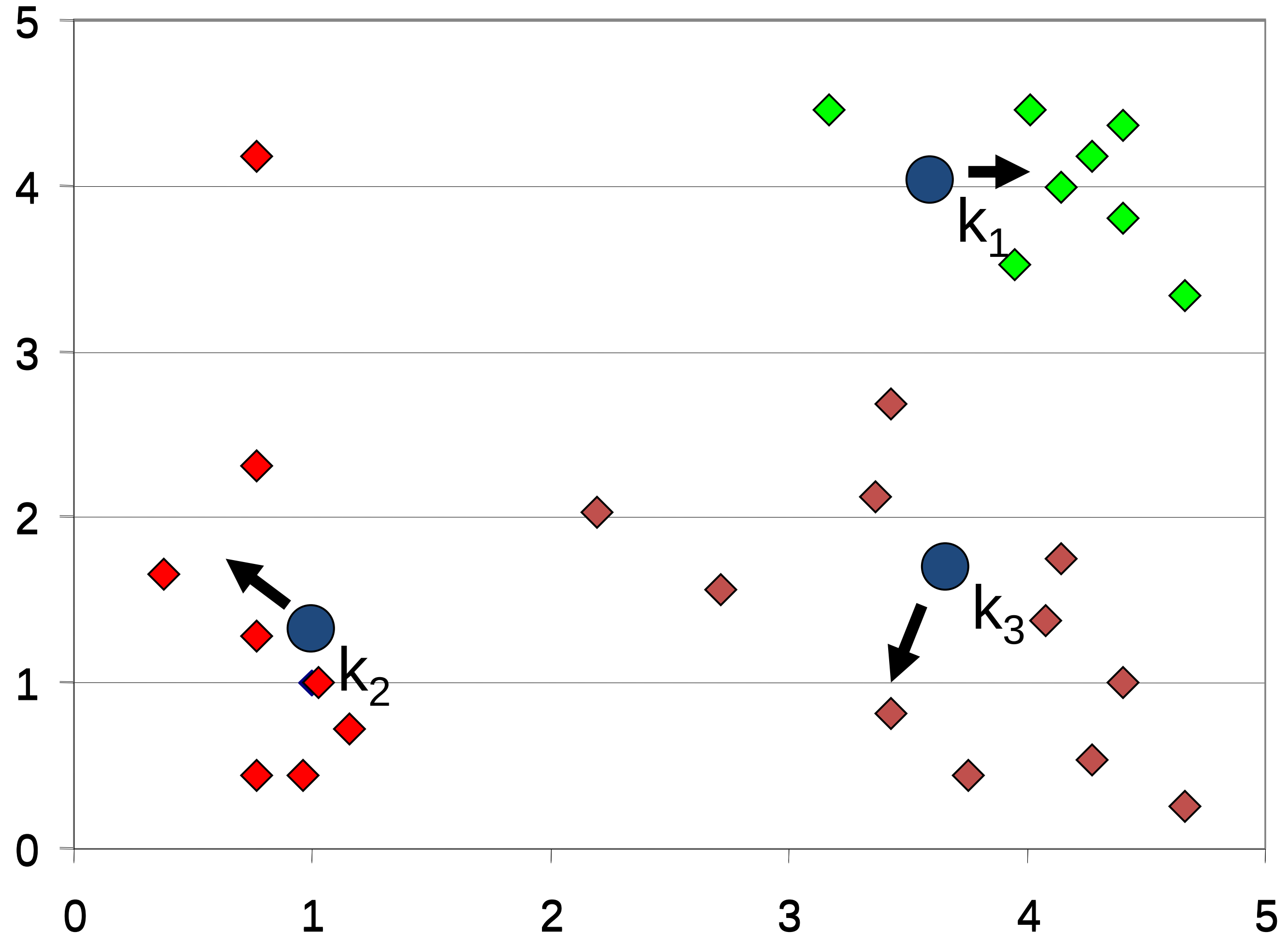
# K-means Clustering: Step 3

Algorithm: k-means,
Distance Metric: Euclidean Distance



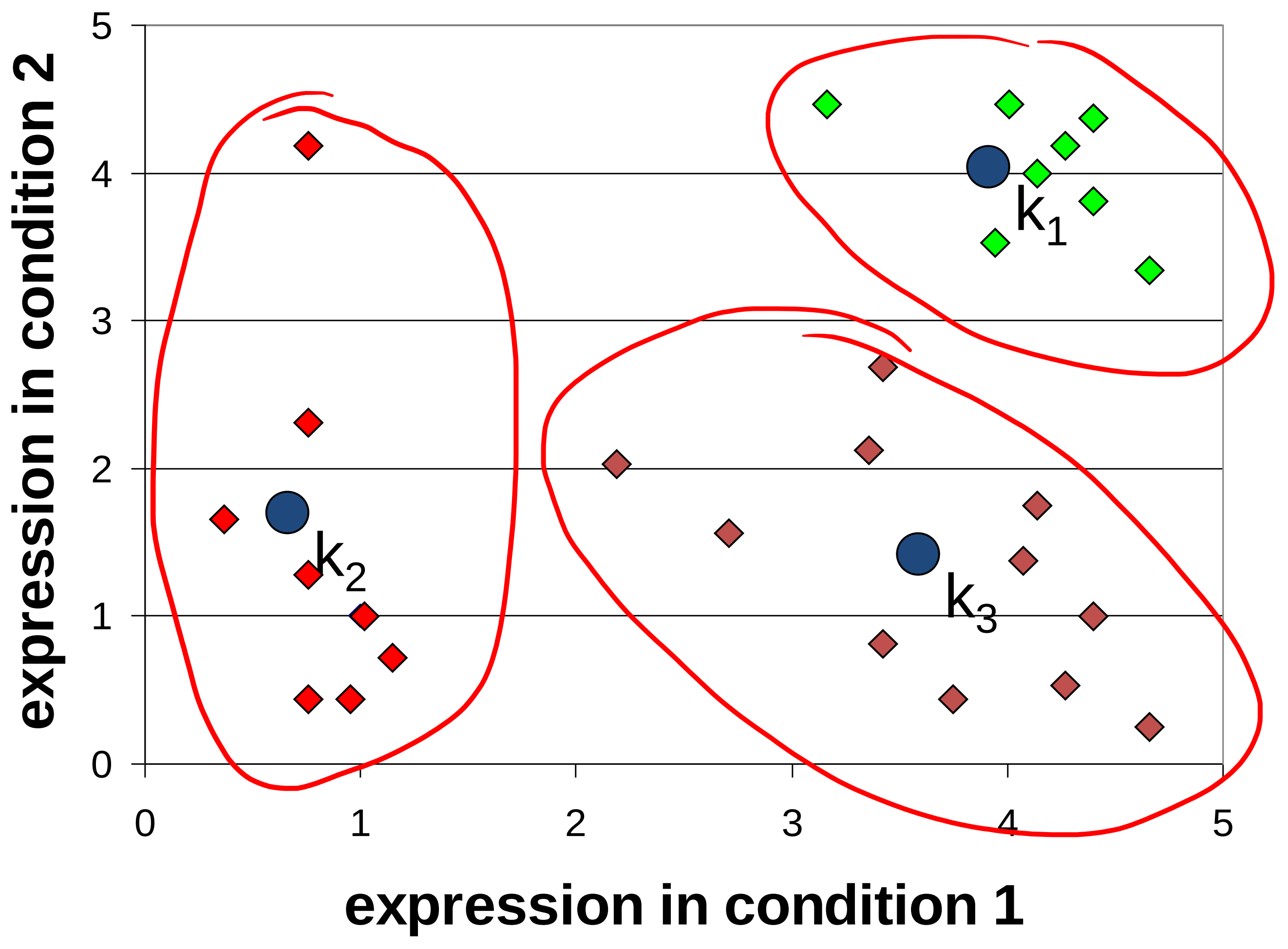
# K-means Clustering: Step 4

Algorithm: k-means,
Distance Metric: Euclidean Distance



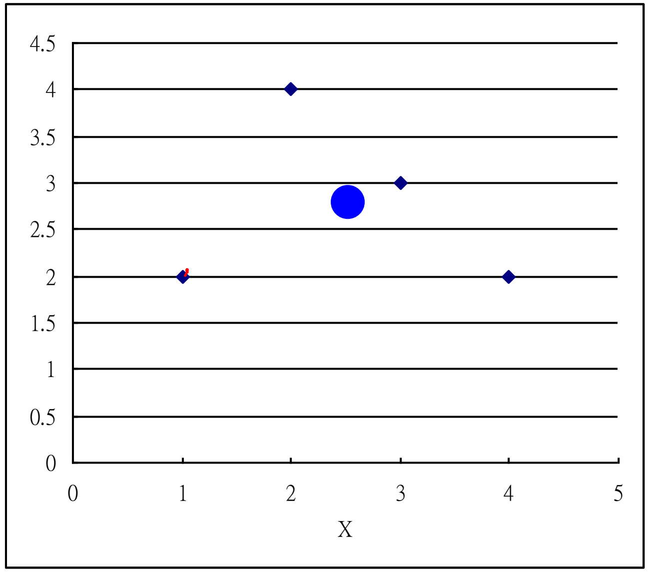
# K-means Clustering: Step 5

Algorithm: k-means,
Distance Metric: Euclidean Distance



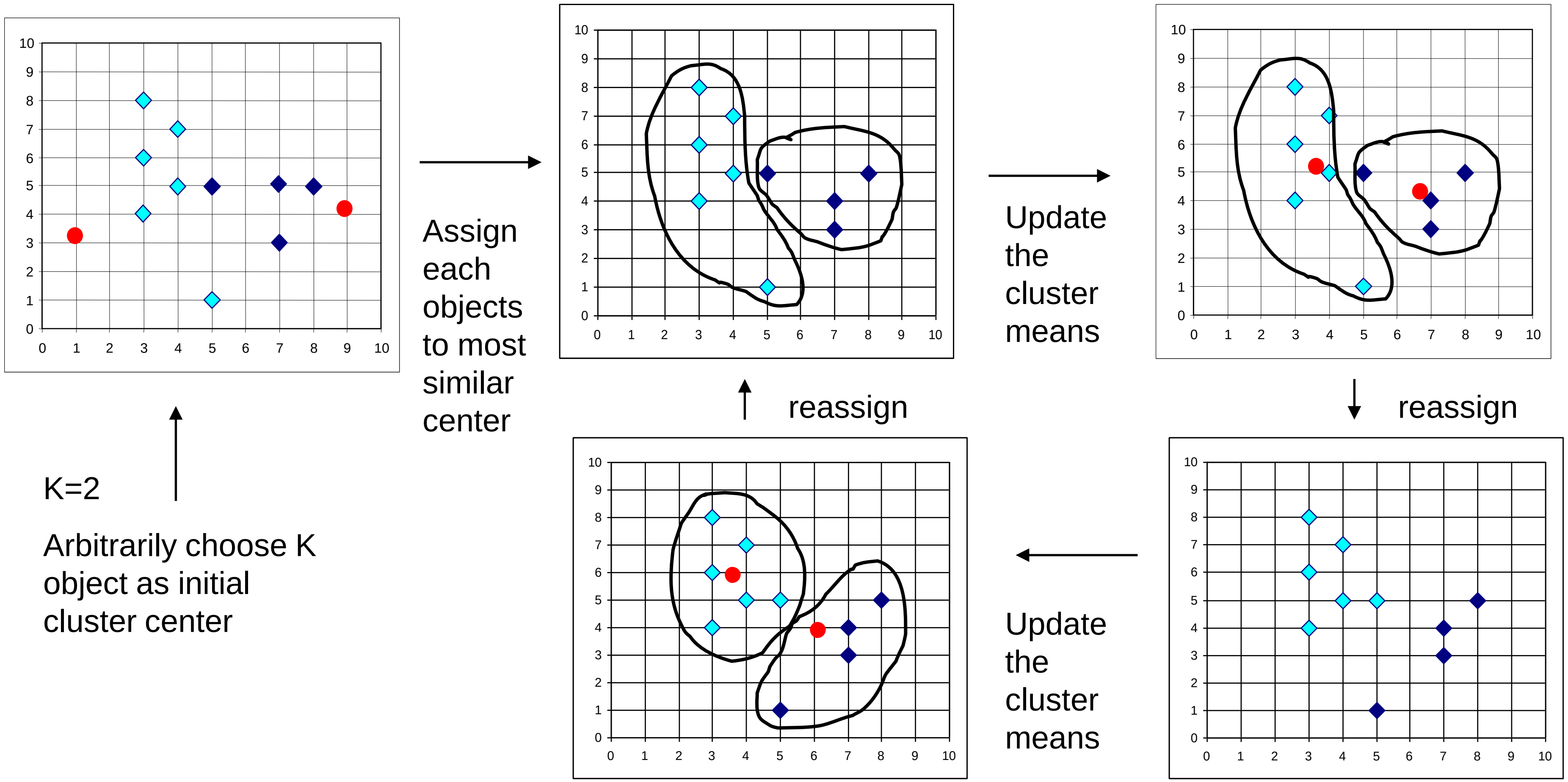
# The mean point can be influenced by an outlier

| X | Y |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 3 |
| 4 | 2 |
| 2.5 | 2.75 |

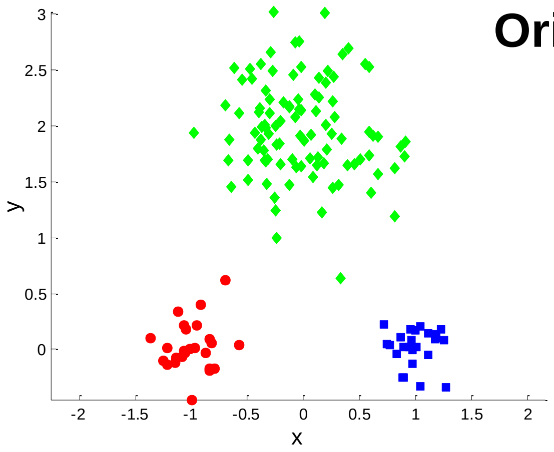



(The mean point can be a virtual point)

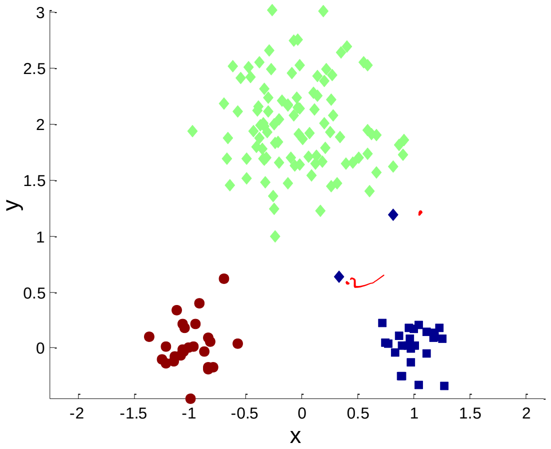
# The K-Means Clustering Method



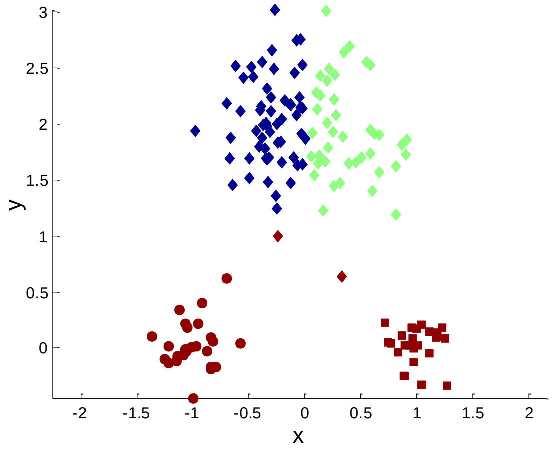
# K-means Clustering




**Original Points**

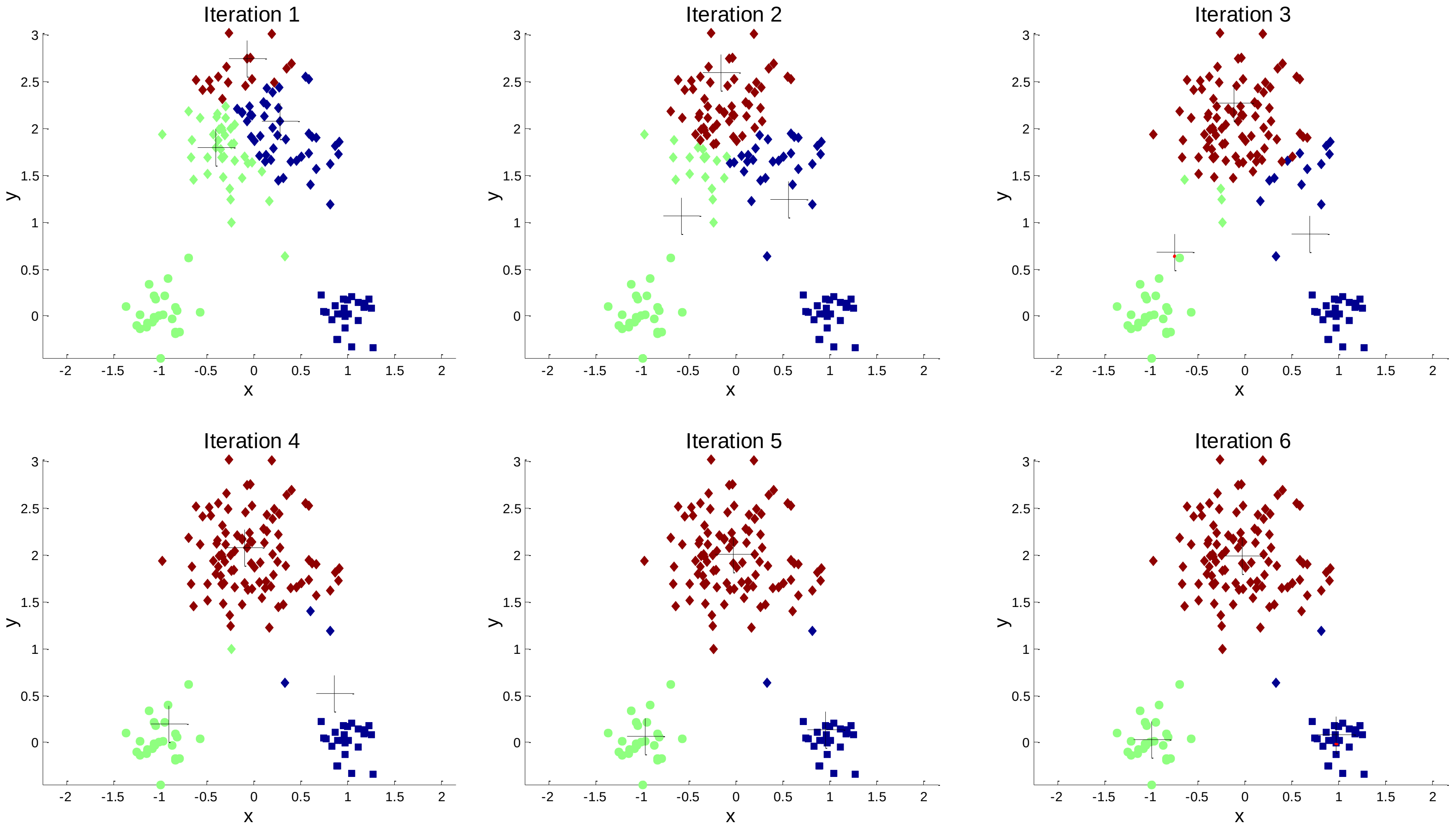



**Optimal Clustering**




**Sub-optimal Clustering**

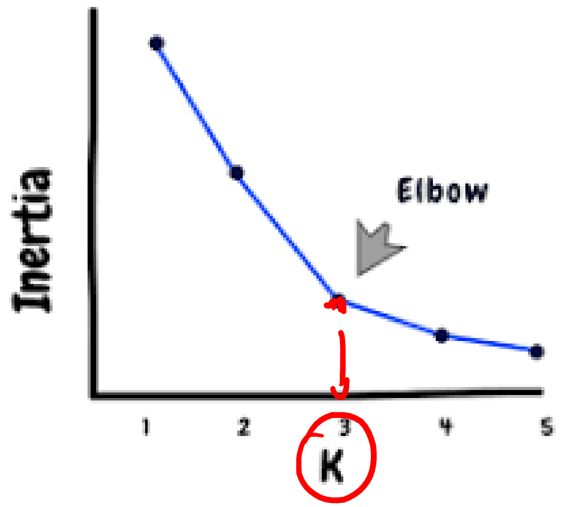
# Importance of Choosing Initial Centroids

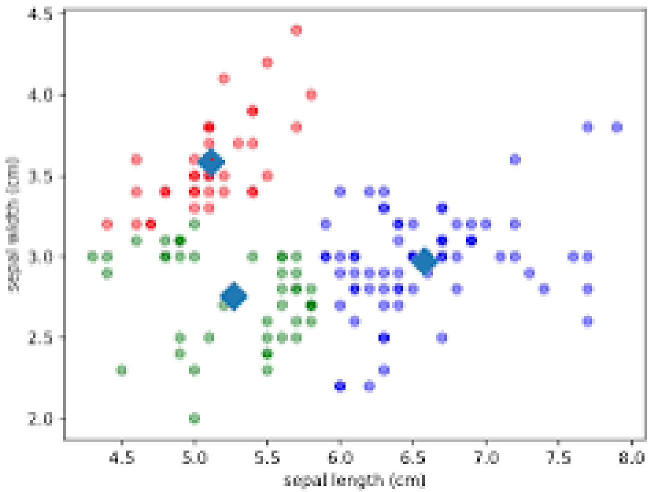


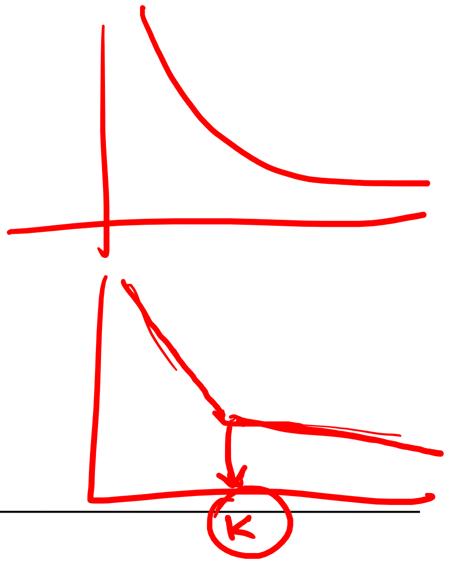
# K-Means: Inertia

k = 3 & k = 20

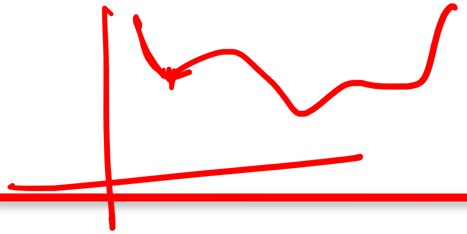
- It is calculated by measuring the distance between each data point and its centroid, squaring this distance, and summing these squares across one cluster.
- A good model is one with low inertia AND a low number of clusters ( K ).
- However, this is a tradeoff because as K increases, inertia decreases.

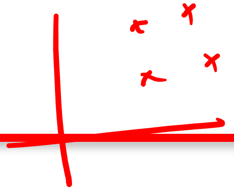


# Comments on k-Means

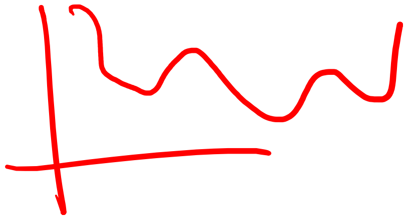




## Strengths

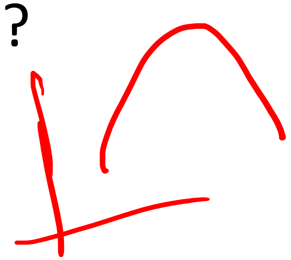
*Relatively efficient*: $O(tkn)$, where $n$ is # objects, $k$ is # clusters, and $t$ is # iterations. Normally, $k$, $t$ << $n$.

Often terminates at a local optimum.



## Weakness

Applicable only when mean is defined, then what about categorical data?



Need to specify *k,* the number of clusters, in advance

Unable to handle noisy data and outliers

Not suitable to discover clusters with non-convex shapes

A A A B B B B B

C C

# Categorical Values

- Handling categorical data: k-modes (Huang'98)
  - Replacing means of clusters with modes
    - Mode of an attribute: most frequent value
    - Mode of instances: for an attribute A, mode(A)= most frequent value
    - K-mode is equivalent to K-means
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: k-prototype method

# Python Packages needed

- pandas
  - Data Analytics
- numpy
  - Numerical Computing
- matplotlib.pyplot
  - Plotting graphs
- Sklearn, Scipy
  - Clustering Classes

# Implementation Using sklearn

Let's go to Jupyter Notebook!