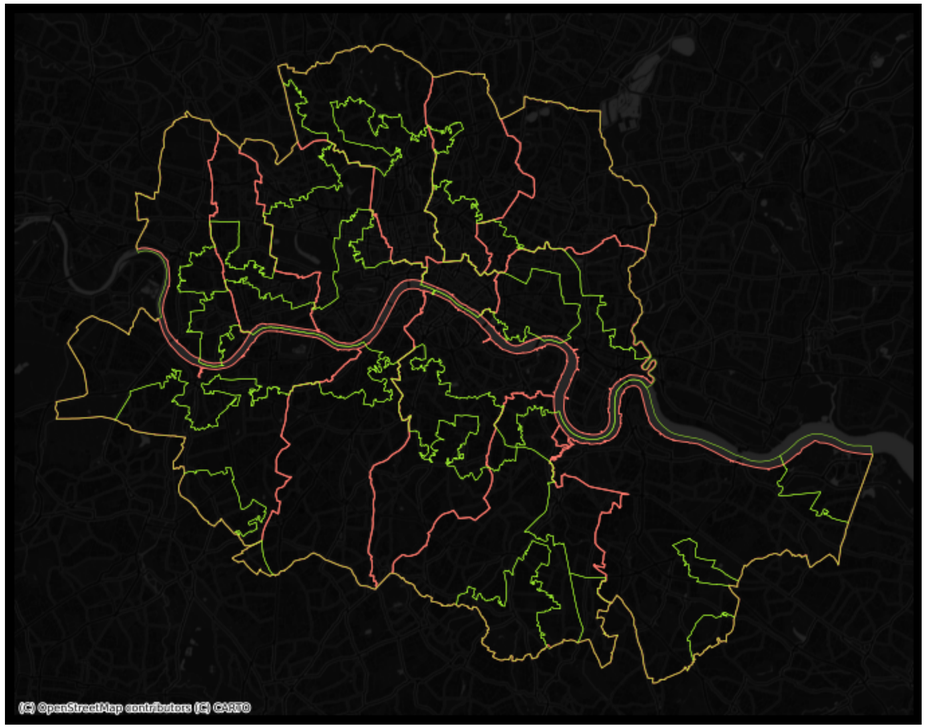
Lecture 6: Spatial clustering

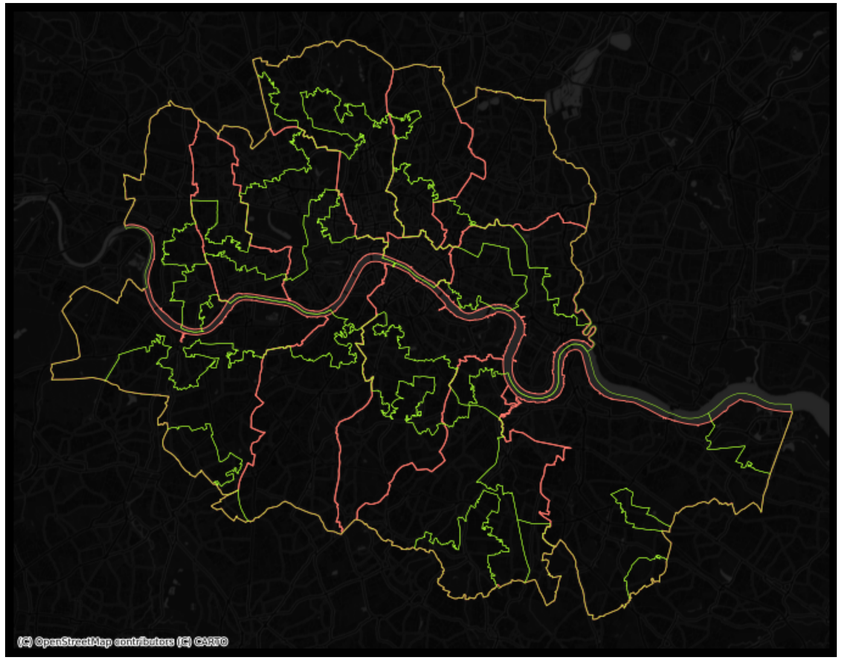
Instructor: Michael Szell
Mar 10, 2022



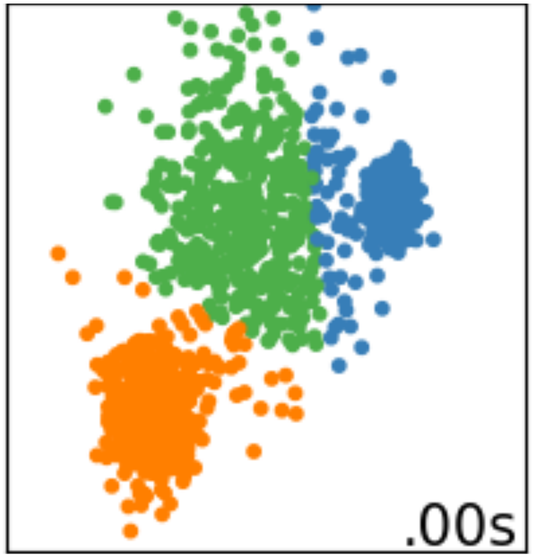
Today you will learn about spatial clustering

GDS QUIZ
8 questions
100s/question

Regionalization
with AirBnB data



Clustering



$$H = \frac{\sum_{i=1}^{p} w_i}{\sum_{i=1}^{p} u_i + \sum_{i=1}^{p} w_i}$$

Everything should be made as simple as possible, but not simpler.

Albert Einstein

From univariate to multivariate

The world is complex and multidimensional.

Univariate	Multivariate
Percent of foreign-born	Neighborhood
Years of schooling	Human development
Monthly income	Deprivation

Types of clustering

Non-spatial

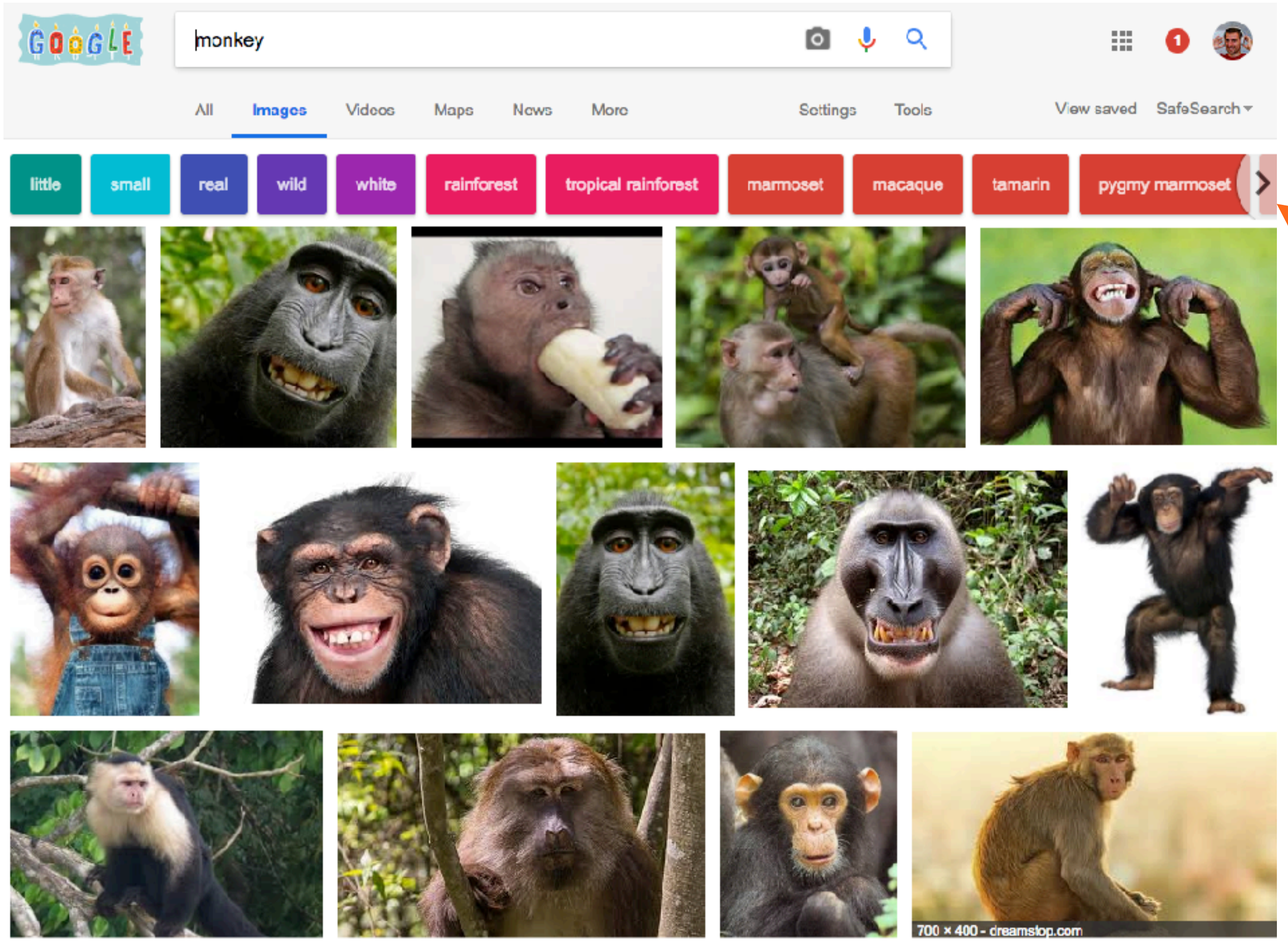
Spatial / Regionalization

Cluster analysis

Cluster analysis is the division of data into groups that are meaningful, useful, or both.

It can be the starting point for other purposes like data summarization.

Cluster analysis: useful for understanding/handling data

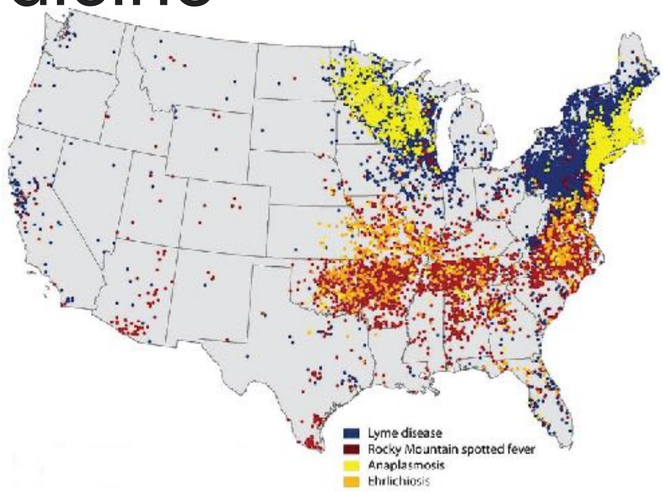


Automatic clustering to refine search!

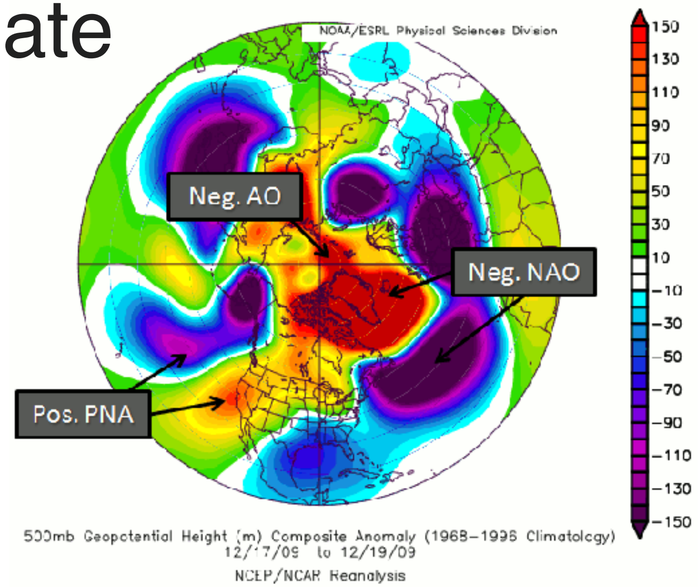
Images should not be too similar, but have some diversity

Cluster analysis: useful for understanding/handling data

Medicine



Climate



Business

Segment customers for additional analysis and to target for marketing activities

Facebook

Facebook allowed advertisers to target 'Jew haters'

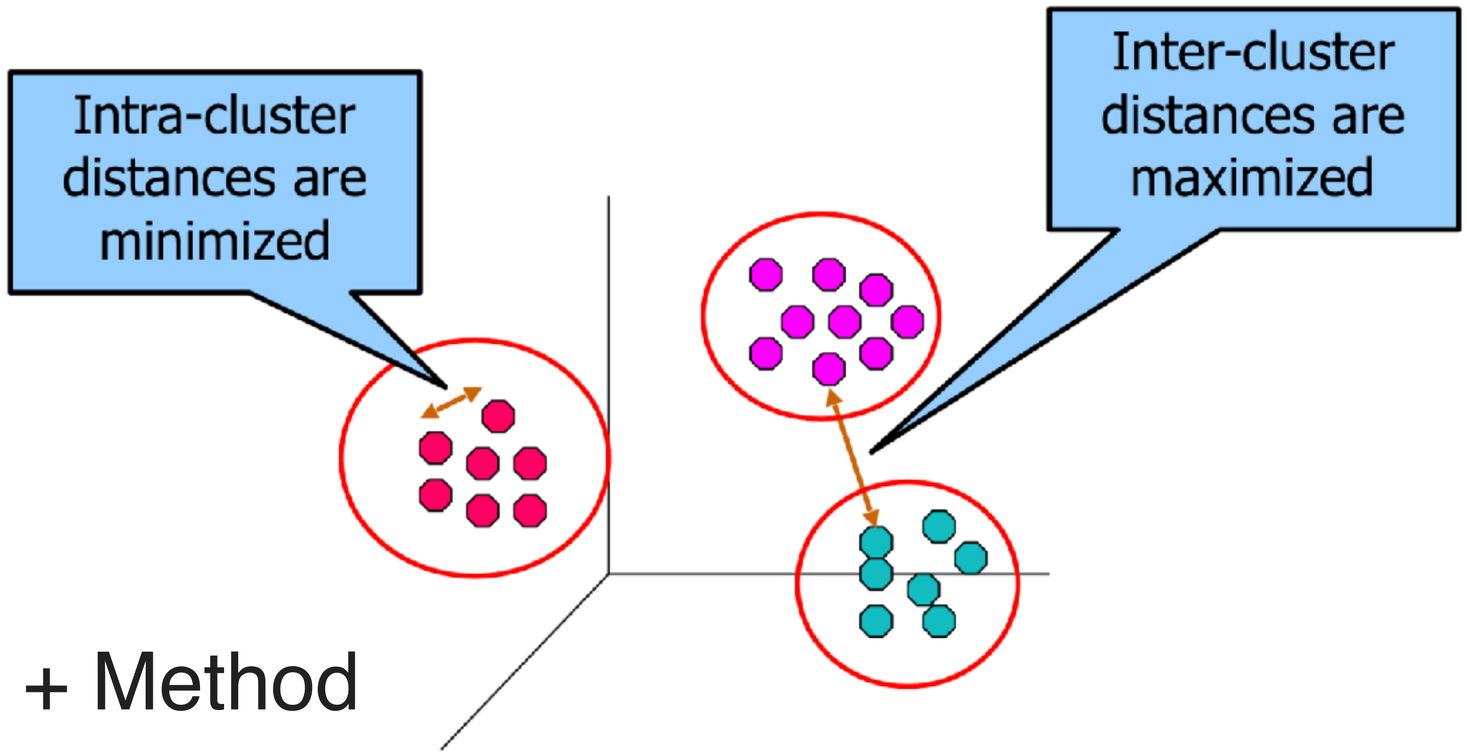
Embarrassing discovery that Facebook let advertisers target users interested in antisemitic topics comes as the social network's ad practices are under scrutiny

Cluster analysis: useful for understanding/handling data

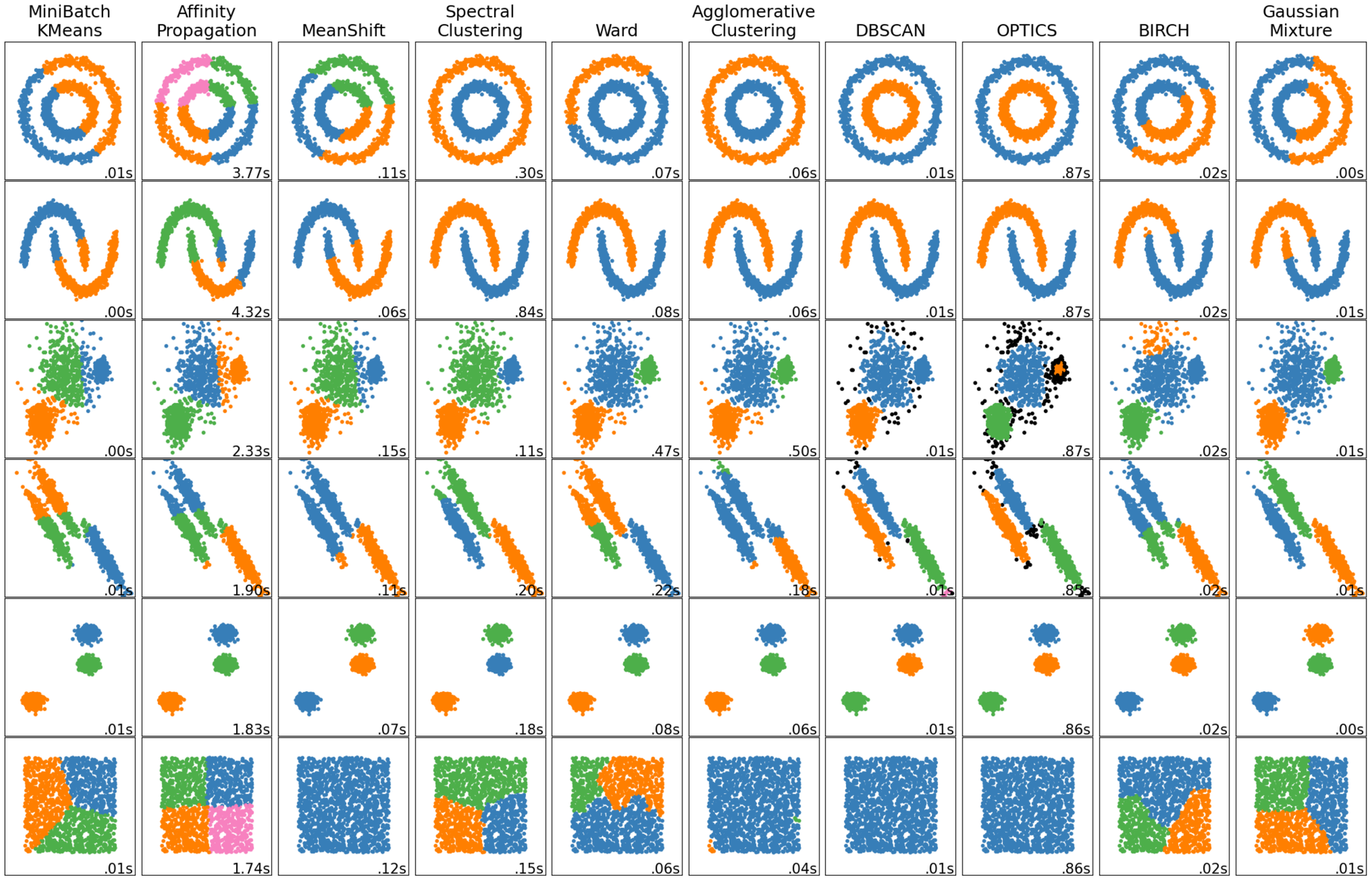
Sometimes a representative object, the cluster prototype,
is identified to simplify data analysis/processing techniques.

Cluster analysis: General concept

Finding groups of objects such that the objects in a group will be similar or related to one another and dissimilar or unrelated to the objects in other groups



Clustering = Distance + Method



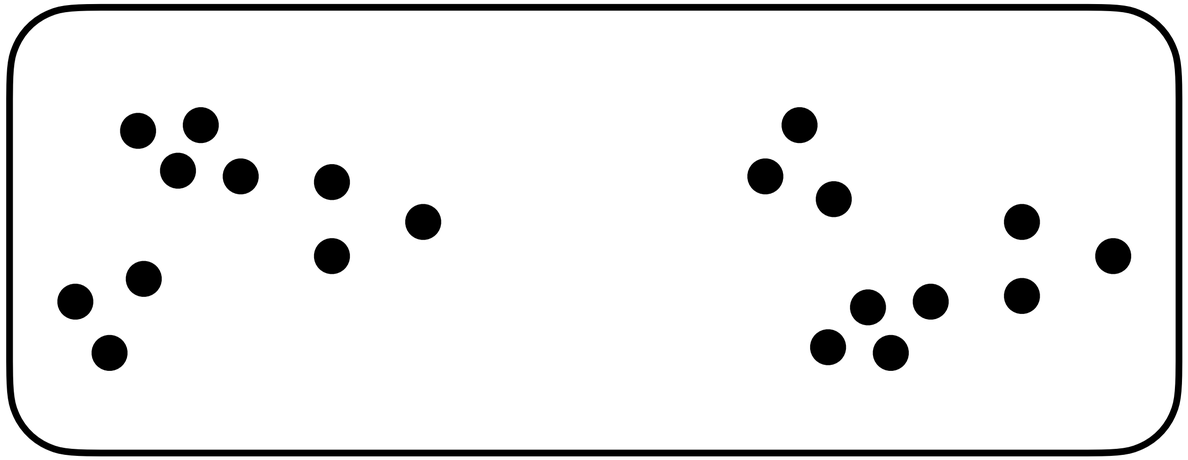
https://scikit-learn.org/stable/modules/clustering.html#overview-of-clustering-methods

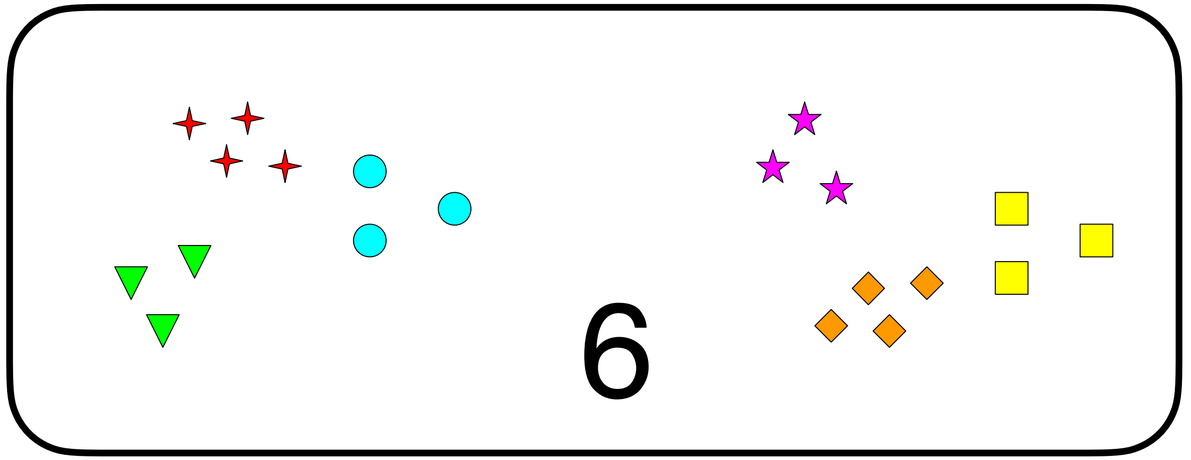
Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters, inductive	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry, inductive	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry, inductive	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry, transductive	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters or distance threshold	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, transductive	Distances between points
Agglomerative clustering	number of clusters or distance threshold, linkage type, distance	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances, transductive	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes, outlier removal, transductive	Distances between nearest points
OPTICS	minimum cluster membership	Very large `n_samples`, large `n_clusters`	Non-flat geometry, uneven cluster sizes, variable cluster density, outlier removal, transductive	Distances between points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation, inductive	Mahalanobis distances to centers
BIRCH	branching factor, threshold, optional global clusterer.	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction, inductive	Euclidean distance between points

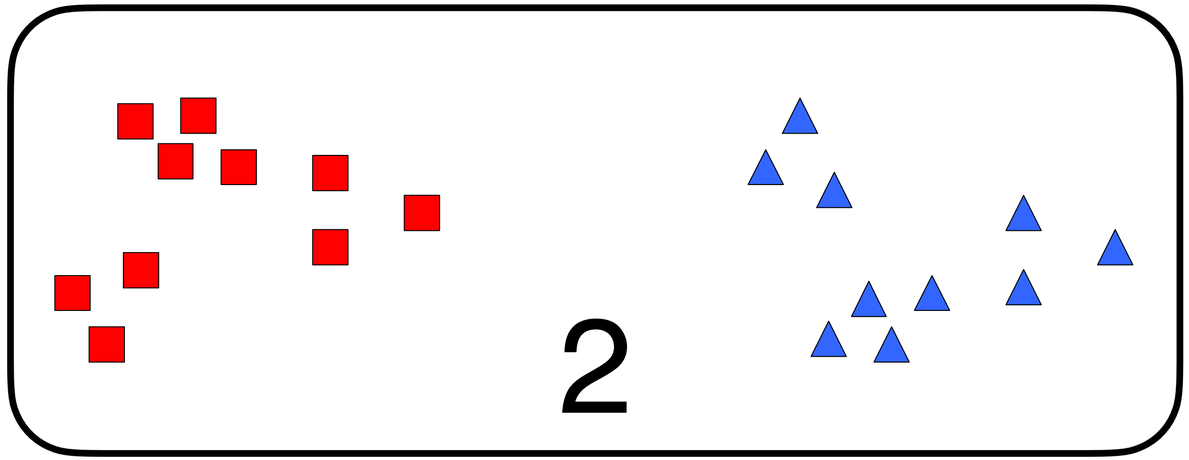
https://scikit-learn.org/stable/modules/clustering.html#overview-of-clustering-methods

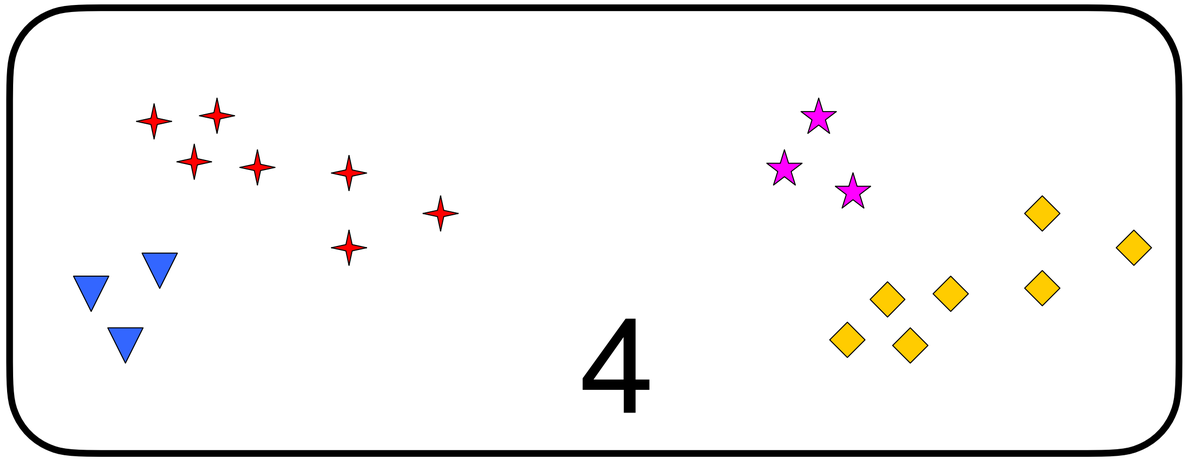
How many clusters are there?

Clustering is ambiguous







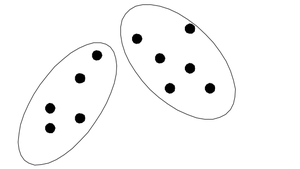


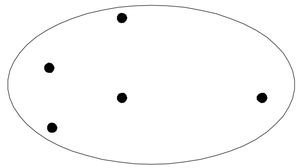
There is no clustering technique that is universally applicable in uncovering the variety of structures present in multidimensional data sets

Partitional vs hierarchical

Partitional clustering

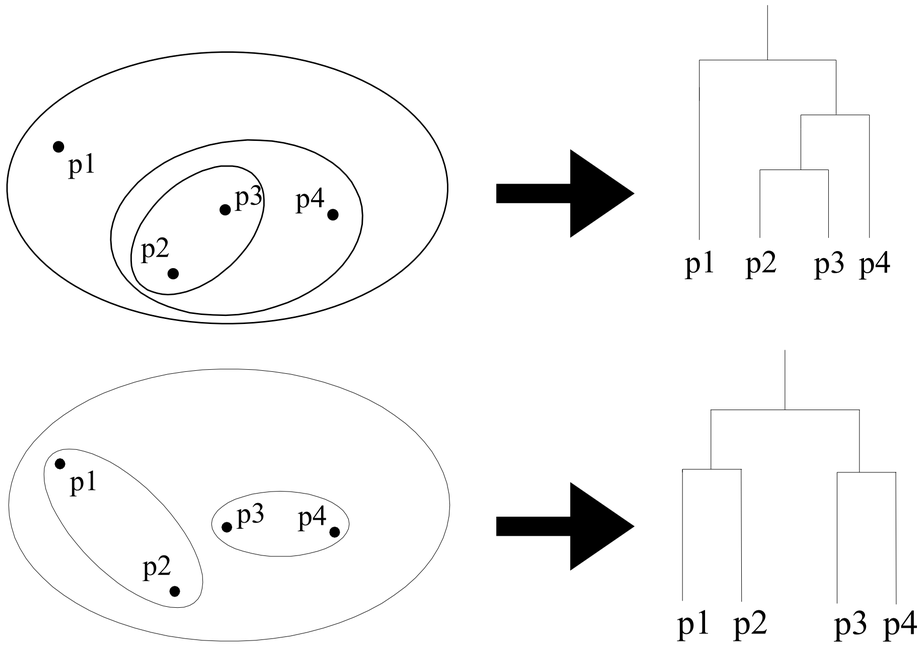
A division into non-overlapping subsets. Each data object is in exactly one subset.





Hierarchical clustering

A set of nested clusters organized as a tree



Partitional vs hierarchical

Agglomerative clustering (bottom-up)

Start with points being individual clusters

At each step merge the closest pair of clusters.

Needs: Similarity measure

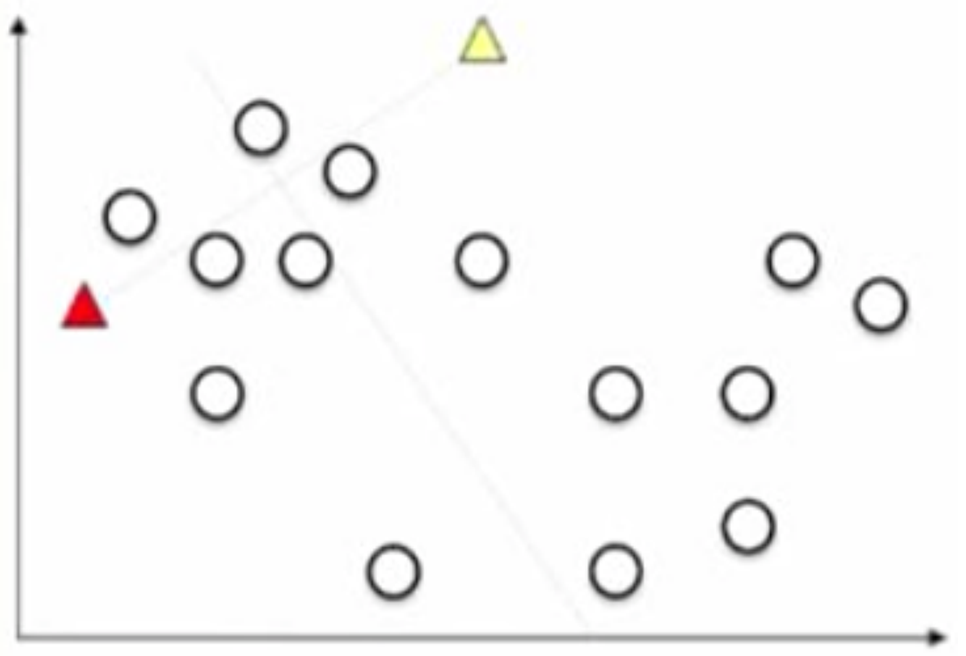
Divisive clustering (top-down)

Start with one all-inclusive cluster.

At each step split a cluster until clusters of single points remain.

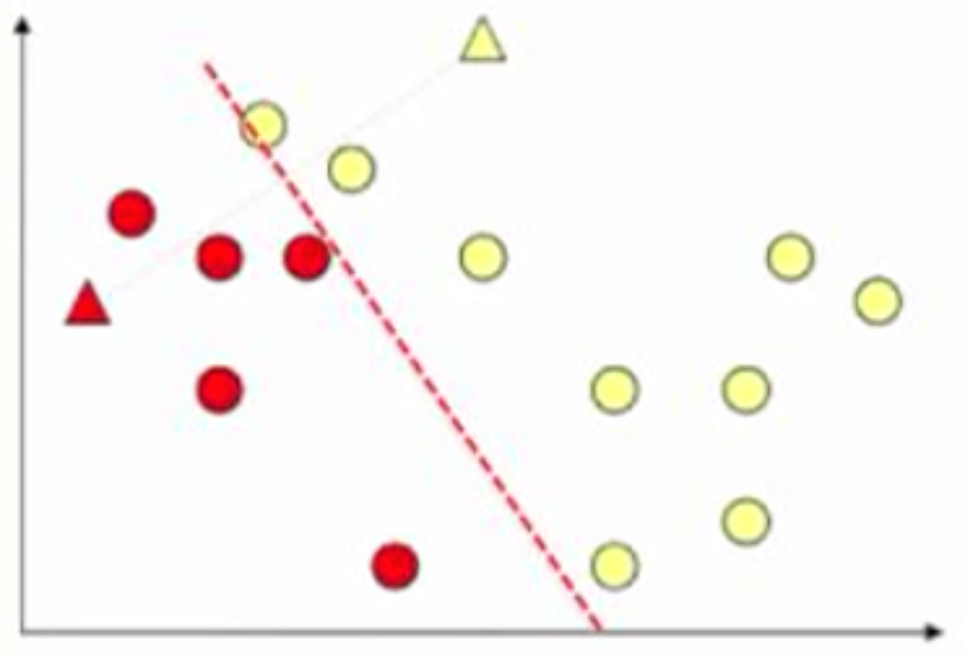
Needs: Way to decide a split

K-means



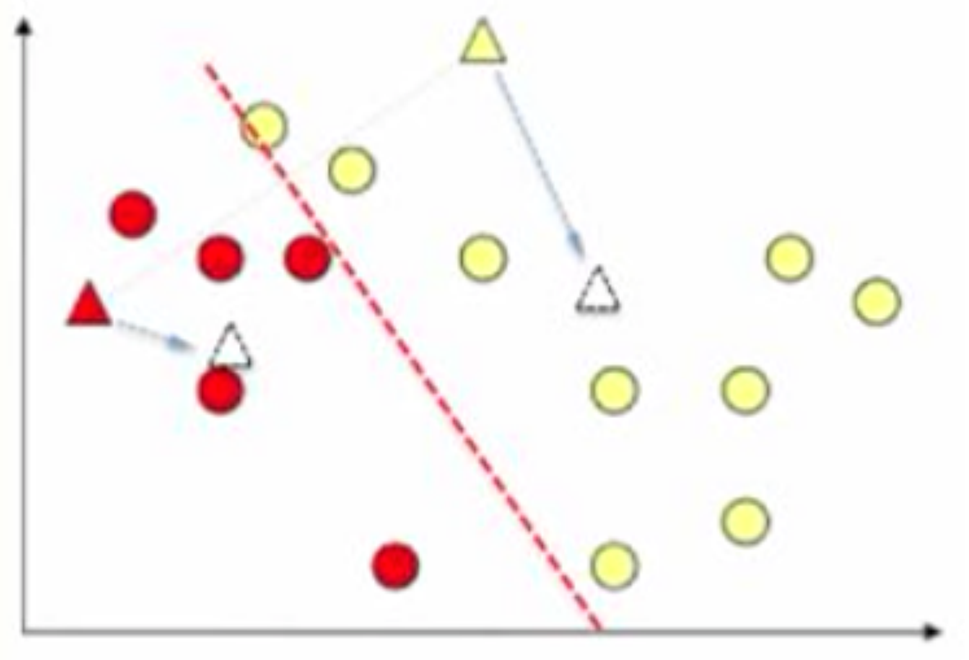
Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.



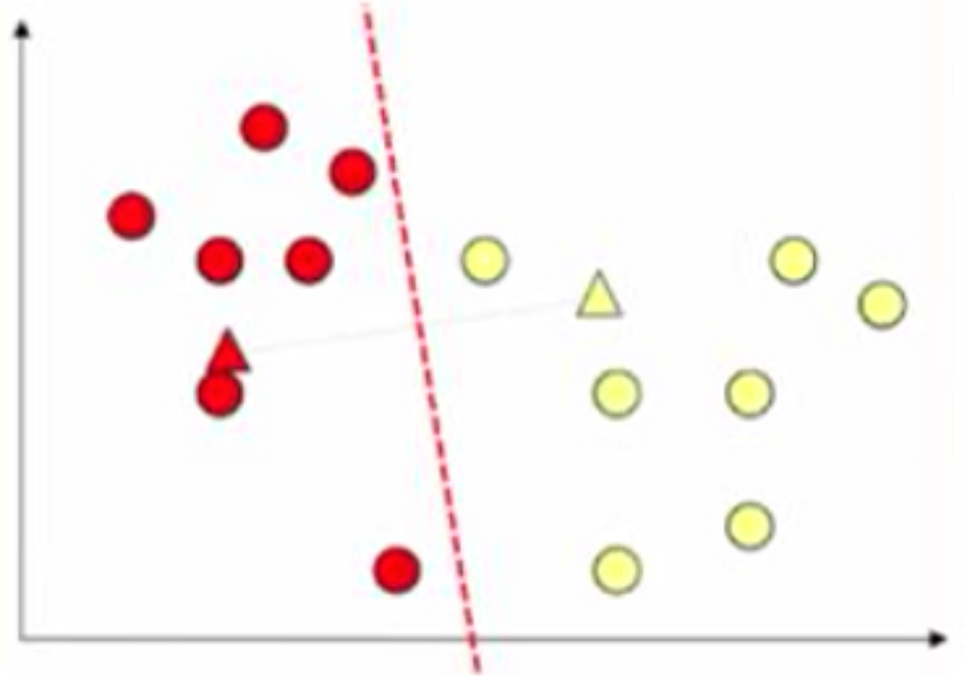
Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.



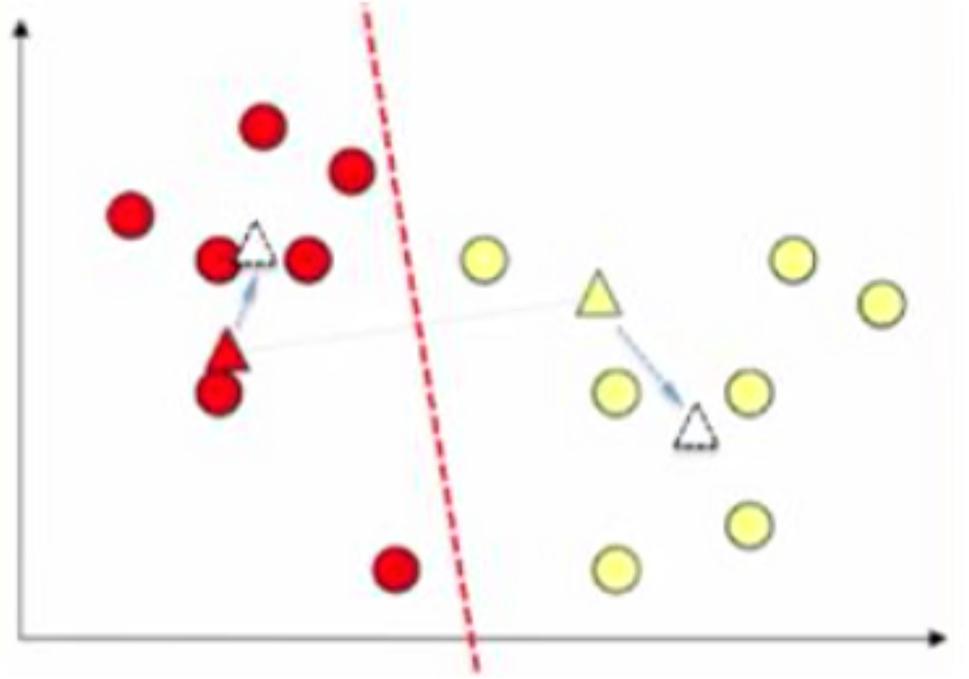
Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.



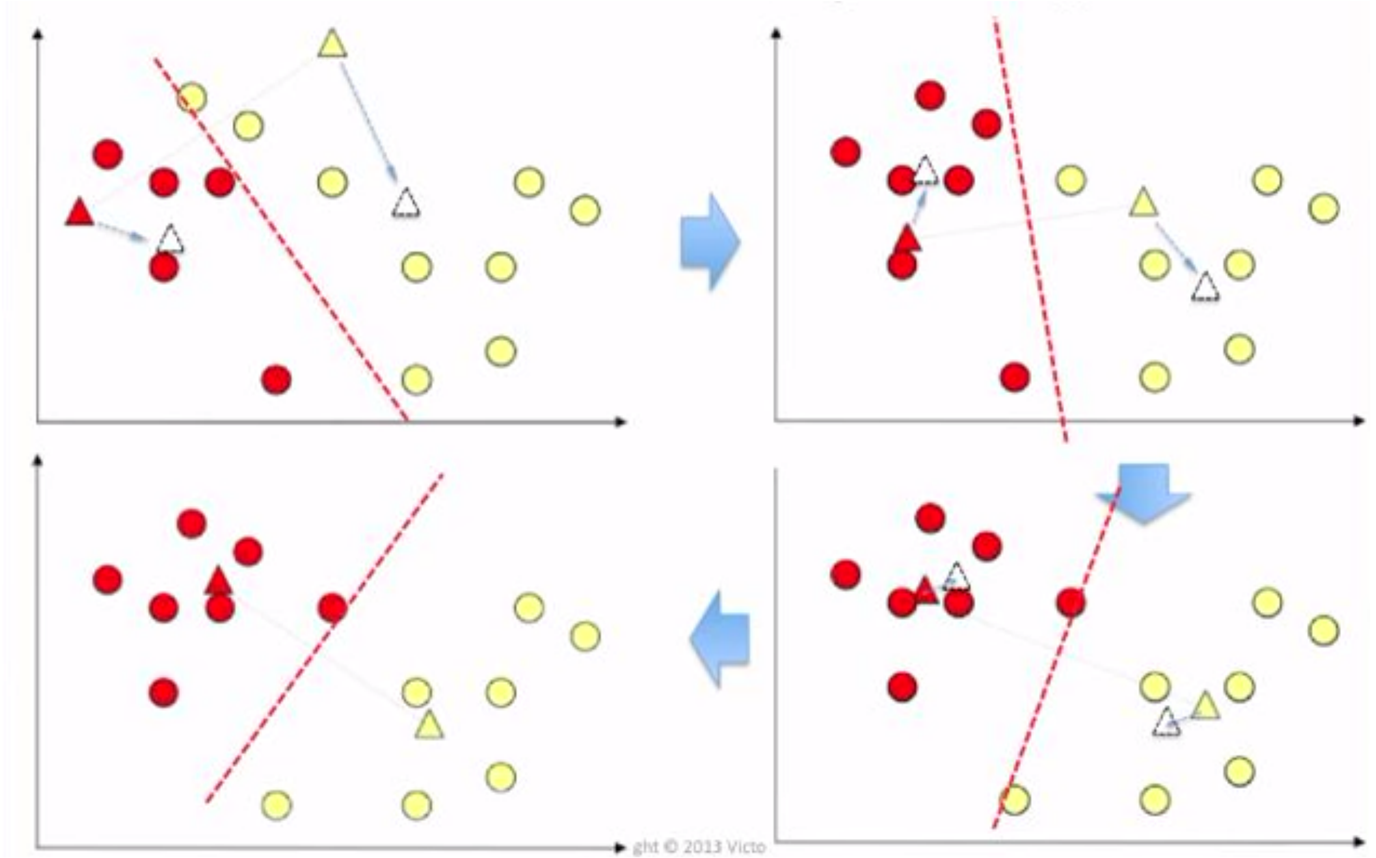
Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.



Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.



Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.

https://www.youtube.com/watch?v=_aWzGGNrcic

K-means: Math background

We want to minimize the sum of squared errors (SSE)

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} dist(\mathbf{c}_i, \mathbf{x})^2$$

The centroid of the ith cluster is just the mean of the points

$$\mathbf{c}_i = \frac{1}{m_i} \sum_{\mathbf{x} \in C_i} \mathbf{x}$$

Distance can be Euclidian, Manhattan, Minkowski,..

K-means: Pros and Cons

Advantages

Easy to implement

Fast: $O(n)$

Disadvantages

Number of partitions needs to be known

Sensitivity to initial conditions

Not effective under several conditions

Often K-means is not effective

K-means works well when data is partitioned into globular clusters of same size and density

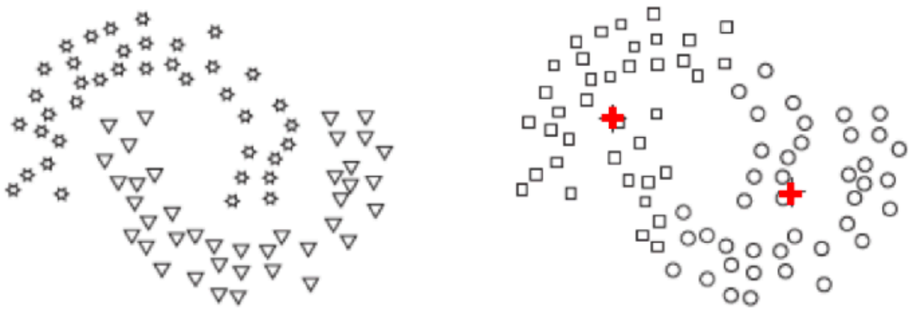
(a) Original points. (b) Three K-means clusters.

Figure 8.9. Clusters of different size

Otherwise:

(a) Original points. (b) Three K-means clusters.

Figure 8.10 Clusters of different density

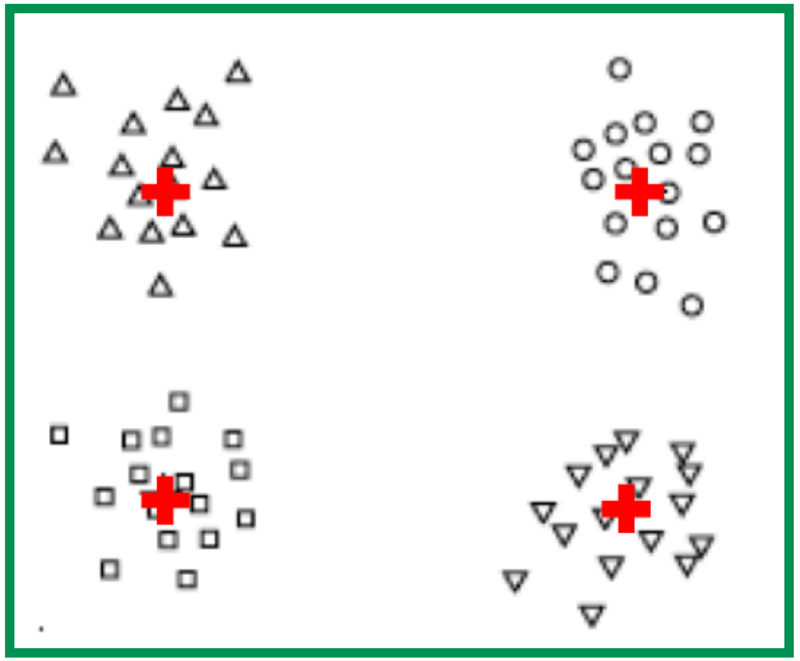


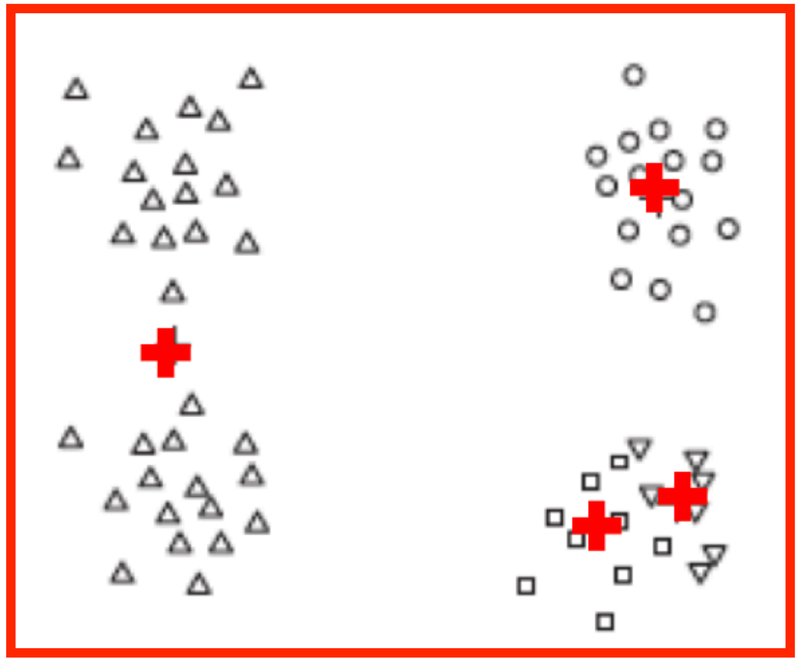
(a) Original points. (b) Two K-means clusters.

Figure 8.11. Non globular clusters

Also problems with outliers

K-means can get stuck in a local minimum





Does our data set even have clusters?

A clustering algorithm will do its job and give us clusters.

But does it make sense?

The Hopkins statistic H measures clustering tendency

A bit like Moran's I, but usually used in a non-spatial context

The Hopkins statistic H measures clustering tendency

Generate p points randomly distributed across the data space, and sample p data points.

For both sets, find the distance to the nearest neighbors in the original data set.

The Hopkins statistic H measures clustering tendency

Generate p points randomly distributed across the data space, and sample p data points.

For both sets, find the distance to the nearest neighbors in the original data set.

nn-distance of synthetic data

Hopkins statistic $$H = \frac{\sum_{i=1}^{p} w_i}{\sum_{i=1}^{p} u_i + \sum_{i=1}^{p} w_i}$$

nn-distance of real data

H=1: real data is highly clustered

H>0.75: real data is clustered at 90% confidence level

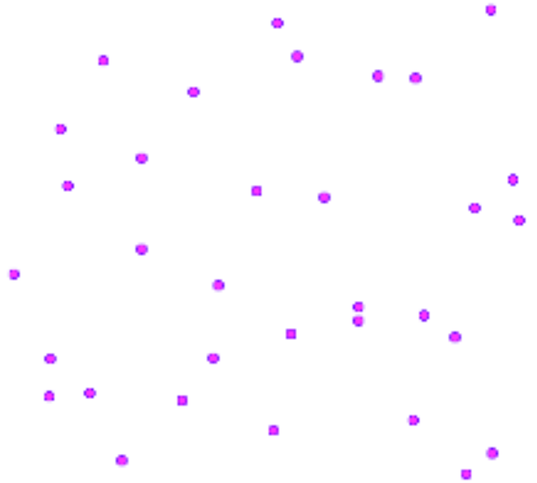
H=0.5: random and real points have same distance distributions

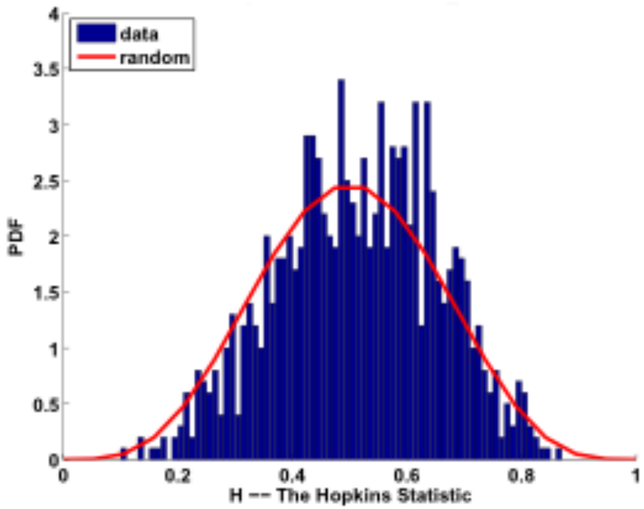
H≅0: real data is uniformly distributed

Typical results from the Hopkins Spatial Statistics Test

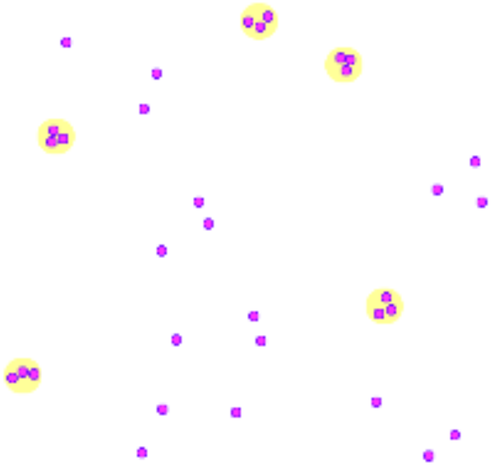
https://stmc.unm.edu/tools-and-data/

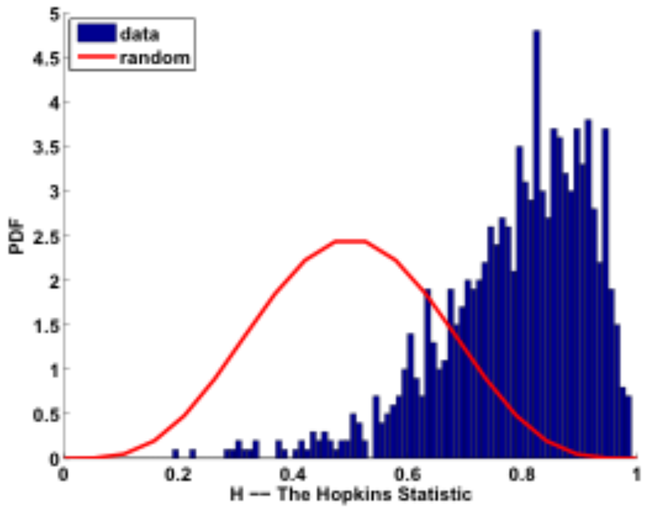
Random



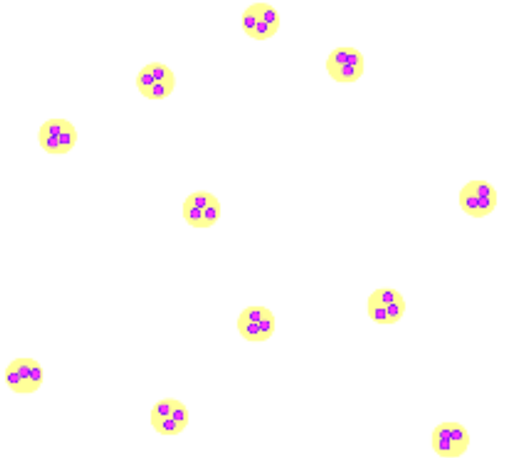


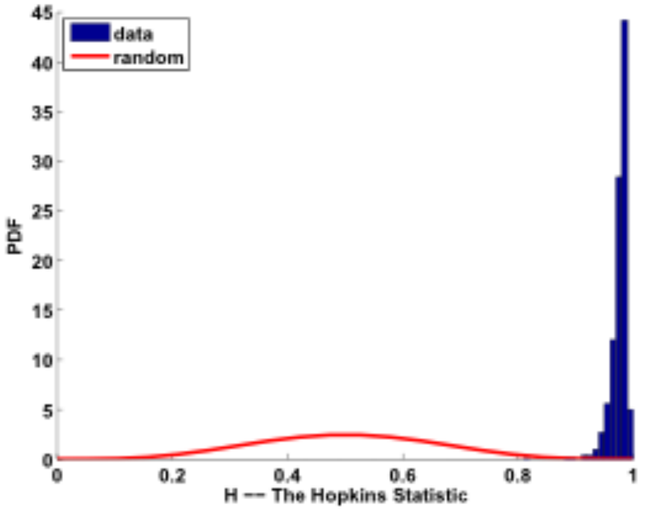
Partially Clustered





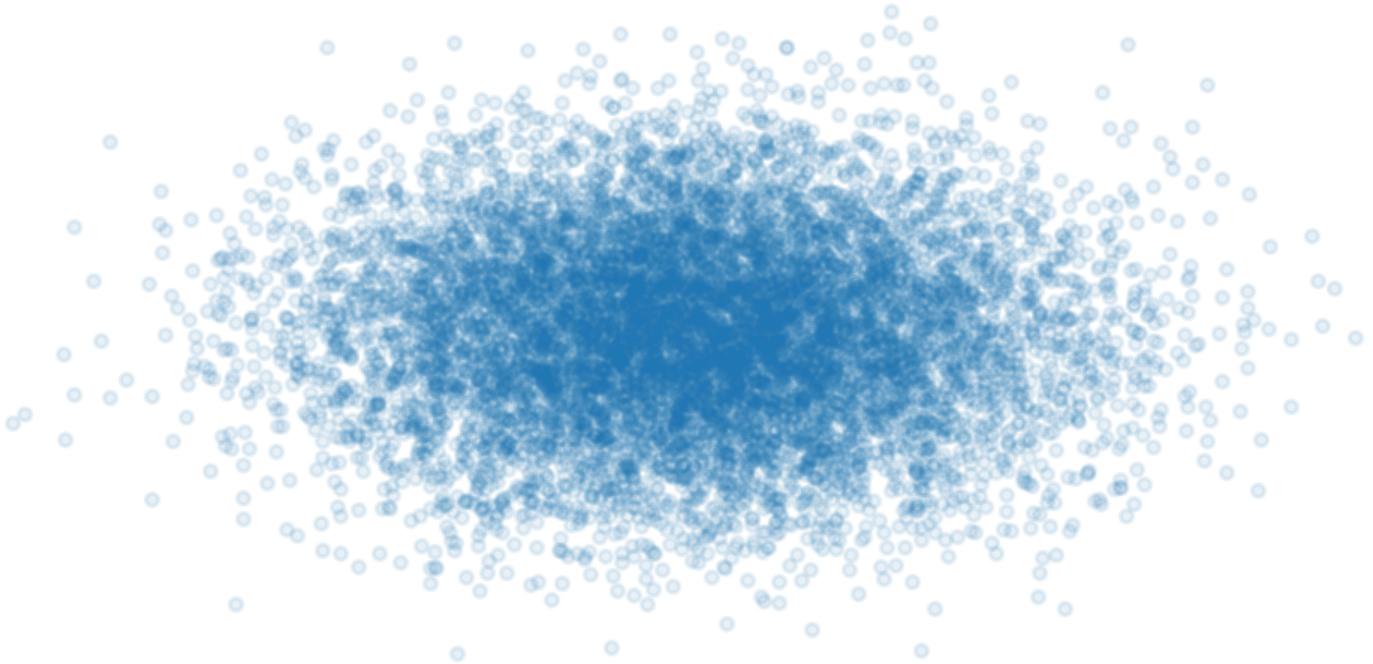
Clustered





The Hopkins statistic compares to a uniform distribution

Criticism of H:
If there is just one "cluster", H is also close to 1



Regionalization is "geographic clustering"

Regionalization is clustering where observations must be geographical neighbors to be able to be in the same category

Jupyter

Measures for quantifying the match between partitions

Many measures that quantify the match between two partitions X and Y build on these 4 numbers:

n_{11} number of pairs of elements in the same community under both X and Y

n_{00} number of pairs of elements not in the same community under both X and Y

n_{01} number of pairs of elements not in the same community under X but in the same community under Y

n_{10} number of pairs of elements in the same community under X but not in the same community under Y

Measures for quantifying the match between partitions

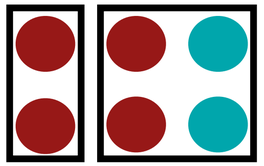
Many measures that quantify the match between two partitions X and Y build on these 4 numbers:

n_{11} number of pairs of elements in the same community under both X and Y

n_{00} number of pairs of elements not in the same community under both X and Y

n_{01} number of pairs of elements not in the same community under X but in the same community under Y

n_{10} number of pairs of elements in the same community under X but not in the same community under Y



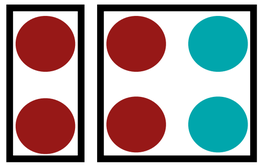
X Colors

Y Boxes

Measures for quantifying the match between partitions

Many measures that quantify the match between two partitions X and Y build on these 4 numbers:

3	n_{11}	number of pairs of elements in the same community under both X and Y
4	n_{00}	number of pairs of elements not in the same community under both X and Y
4	n_{01}	number of pairs of elements not in the same community under X but in the same community under Y
4	n_{10}	number of pairs of elements in the same community under X but not in the same community under Y



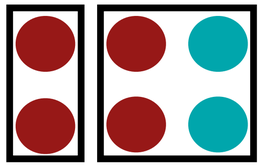
X Colors

Y Boxes

Measures for quantifying the match between partitions

Many measures that quantify the match between two partitions X and Y build on these 4 numbers:

3	n_{11}	number of pairs of elements in the same community under both X and Y
4	n_{00}	number of pairs of elements not in the same community under both X and Y
4	n_{01}	number of pairs of elements not in the same community under X but in the same community under Y
4	n_{10}	number of pairs of elements in the same community under X but not in the same community under Y

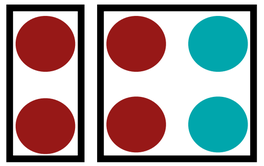


0.47 Rand index $\mathcal{R} = \dfrac{n_{00} + n_{11}}{n_{00} + n_{11} + n_{01} + n_{10}}$ is the accuracy of the pair classification

Measures for quantifying the match between partitions

Many measures that quantify the match between two partitions X and Y build on these 4 numbers:

3	n_{11}	number of pairs of elements in the same community under both X and Y
4	n_{00}	number of pairs of elements not in the same community under both X and Y
4	n_{01}	number of pairs of elements not in the same community under X but in the same community under Y
4	n_{10}	number of pairs of elements in the same community under X but not in the same community under Y



0.43

Fowlkes-Mallows index

$$\mathcal{F} = \sqrt{\frac{n_{11}}{n_{11}+n_{10}} \cdot \frac{n_{11}}{n_{11}+n_{01}}}$$

Measures for quantifying the match between partitions

1971

$$\mathcal{R} = \frac{n_{00} + n_{11}}{n_{00} + n_{11} + n_{01} + n_{10}}$$

1983

$$\mathcal{F} = \sqrt{\frac{n_{11}}{n_{11} + n_{10}} \cdot \frac{n_{11}}{n_{11} + n_{01}}}$$

Both give 1 for a perfect match, 0 when no pair agrees in their clustering.

Problems:

For completely unrelated data, $\mathcal{R}$ goes towards 1.
Not so $\mathcal{F}$ which goes towards 0.

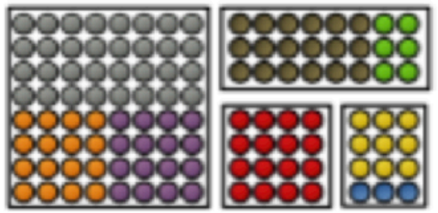
If you reshuffle the data randomly, you wont get 0,
but a baseline > 0 for the random expectation.

Measures for quantifying the match between partitions

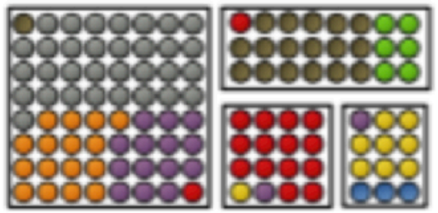
To correct for this baseline, there is an adjusted Rand index

$$\mathcal{ARI} = \frac{n_{00} + n_{11} - \mathbf{E}[n_{00} + n_{11}]}{n_{00} + n_{11} + n_{01} + n_{10} - \mathbf{E}[n_{00} + n_{11}]}$$

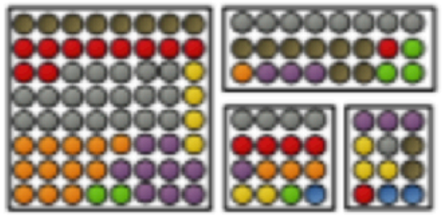
where $\mathbf{E}[n_{00} + n_{11}]$ are the values expected by chance.
This index can be negative.



$\mathcal{ARI} = 1.00$



$\mathcal{ARI} = 0.88$



$\mathcal{ARI} = 0.03$

Sources and further materials for today's class

Geographic Data Science with Python

https://geographicdata.science/book/notebooks/10_clustering_and_regionalization.html

https://darribas.org/gds_course/content/bG/concepts_G.html