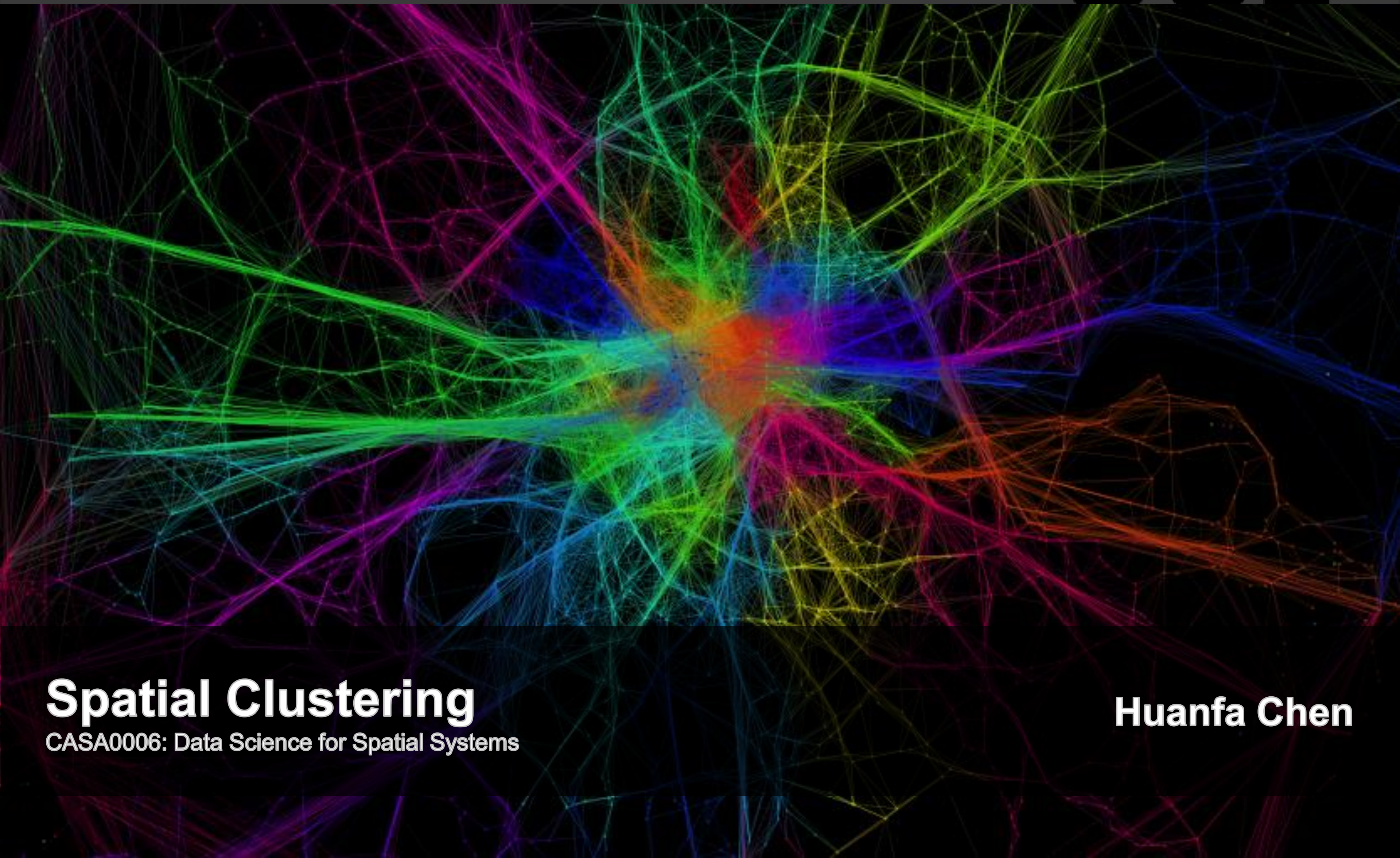


Spatial Clustering

CASA0006: Data Science for Spatial Systems

Huanfa Chen

CASA0006

1. Introduction to Module
2. Supervised Machine Learning
3. Tree-based Methods
4. Artificial Neural Networks
5. Analysis Workflow
6. Spatial Clustering
7. Panel Regression
8. Difference in Difference
9. Regression Discontinuity
10. Dimensionality Reduction

Connecting with CASA0007 (T1) Clustering : Plan of Attack

Standardisation Methods
Z-Score (roughly symmetrical data)
Min-Max rescaling (asymmetric data)
IDR rescaling (data with significant outliers)
Explicit rescaling

Clustering Methods
K-Means
Hierarchical

Clustering Quality
SSE
Silhouette Analysis

Visualisation
Elbow Diagram
Silhouette Plot
Dendrogram
Scatter Plots

Follow Up
Examine cluster centroids
Describe cluster characteristics
Compare against unconsidered variables / categories / geography
Consider analysing clusters separately

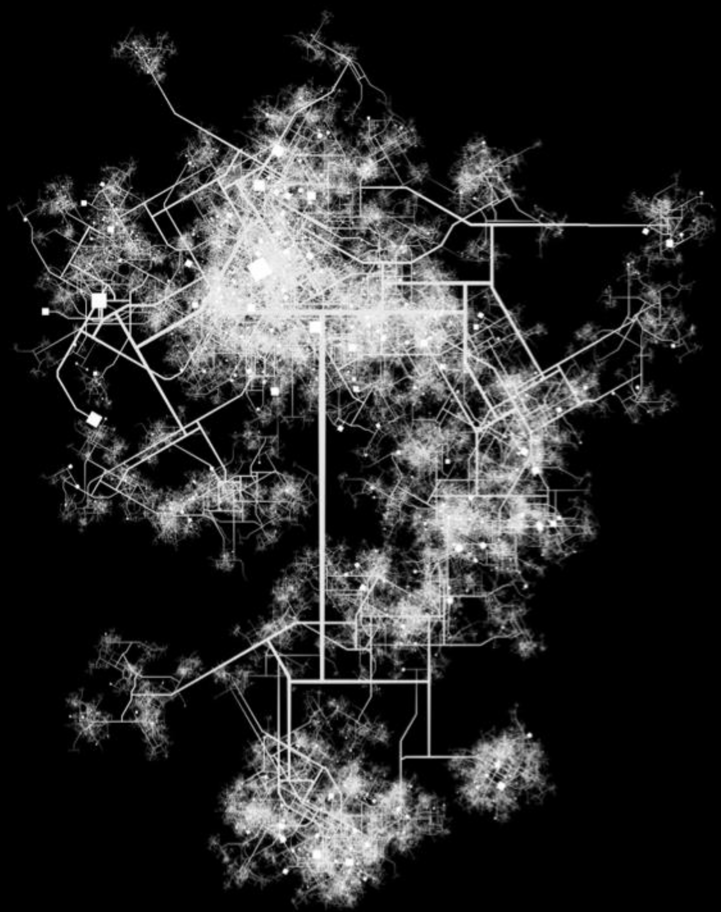
Connecting with CASA0013 (T1, W10)

- Geodemographics
  - Booth map
  - London Output Area Classification
- Clustering methods

Different Approaches

Algorithm	Pros	Cons	Geographically Aware?
k-Means	Fast. Deterministic.	Every observation to cluster.	N.
DBSCAN	Allows for clusters *and* outliers.	Slower. Choice of $$\epsilon$$ critical. Can end up with all outliers.	N, but implicit in $$\epsilon$$.
OPTICS	Fewer parameters than DBSCAN.	Even slower.	N, but implicit in $$\epsilon$$.
Hierarchical	Can cut at any number of clusters.	No 'ideal' solution.	Y, with connectivity parameter
ADBSCAN	Scales. Confidence levels.	May need large data set to be useful. Choice of $$\epsilon$$ critical.	Y.
Max-p	Coherent regions returned.	Very slow if model poorly specified.	Y.

Outline



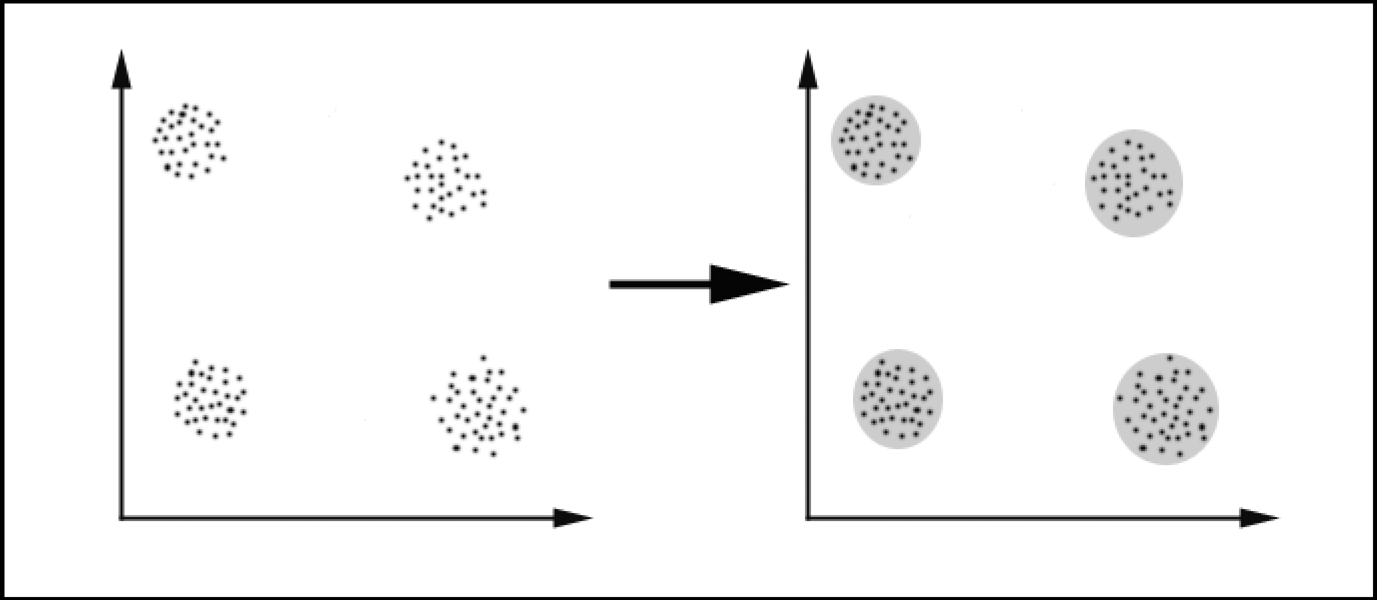
1. Definition and workflow
2. Clustering Methods
    a. K-Means
    b. Hierarchical
    c. DBSCAN
    d. Choosing clustering methods
3. Spatial Clustering
4. Measuring Clustering Quality
    a. SSE/Elbow Method
    b. Silhouette Analysis

4. Next steps

Clustering

Definition

Type of analysis that divides data points into groups based on some similarity criteria



Clustering

- Purpose of clustering
  - Discover groups of similar data points
  - Extract ‘knowledge’ from data

- What is a cluster?
  - A group of similar data points

Standardisation

Z score

(for not highly skewed data)

$$z = \frac{x - \mu}{\sigma}$$

Min-Max Rescaling

(for highly skewed data)

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

IDR Standardisation

(Non-normal data with significant outliers)

$$x^{\mathrm{IDR}} = \begin{cases} \dfrac{x - P_{50}}{P_{90} - P_{50}}, x \geq P_{50} \\ \dfrac{x - P_{50}}{P_{50} - P_{10}}, x < P_{50} \end{cases}$$

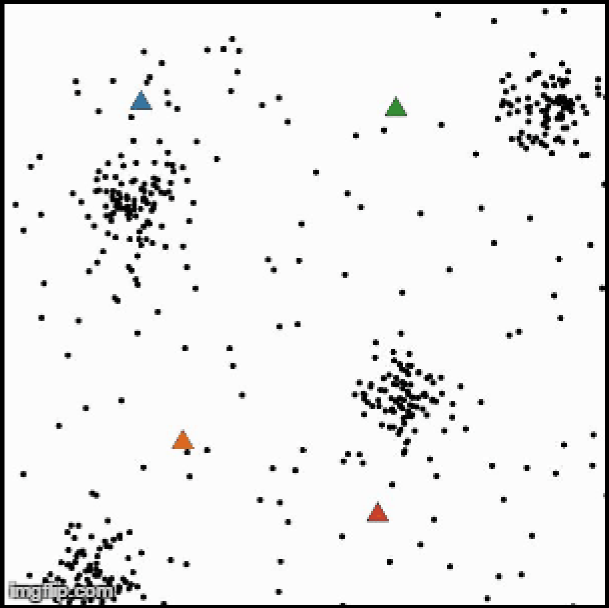
Criteria

1. Highly skewed distribution?
2. Significant outliers?

(See: https://www.ons.gov.uk/methodology/geography/geographicalproducts/areaclassifications/2011areaclassifications/methodologyandvariables)

Clustering
K-Means Clustering

K-Means clustering **breaks down** a dataset into groups, based on proximity of points within a multidimensional space.



Iterative Algorithm

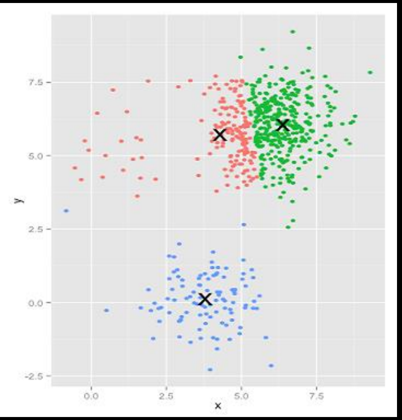
1. Place *k* centroids randomly within space
2. Assign points to nearest centroid
3. Recalculate centroids as the new mean of the cluster
4. Continue until centroid assignments no longer change

Interactive demo of kmeans:
https://jeff3dx.github.io/kmc

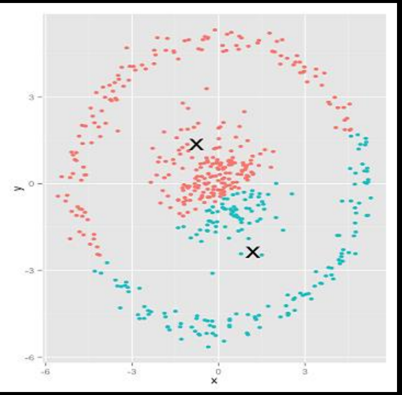
Clustering
Problems with K-Means Clustering

- Requires knowledge of the number of clusters, which you may not know in advance (solution: Elbow method);
- Sensitive to initialisation, which can lead to poor solutions (solution: try different random initialisation and pick up the best one);
- Sensitive to outliers, which can result in inaccurate clusters (solution: use another clustering method, or remove outliers);
- Incapable of handling clusters of a non-convex shape (no solution);
- Inapplicable to categorical data (solution: k-modes or k-prototypes).

Choose k wisely



Non-convex shape

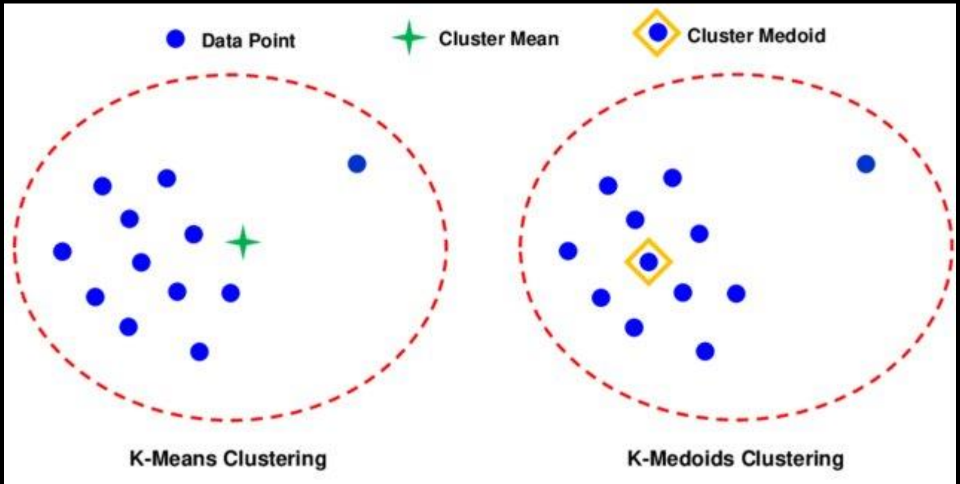


Extension of kmeans
K-modes and K-prototypes

method	Input variables
K-means	numerical
K-modes	categorical
K-prototypes	numerical and categorical

https://pypi.org/project/kmodes/

K-medoids



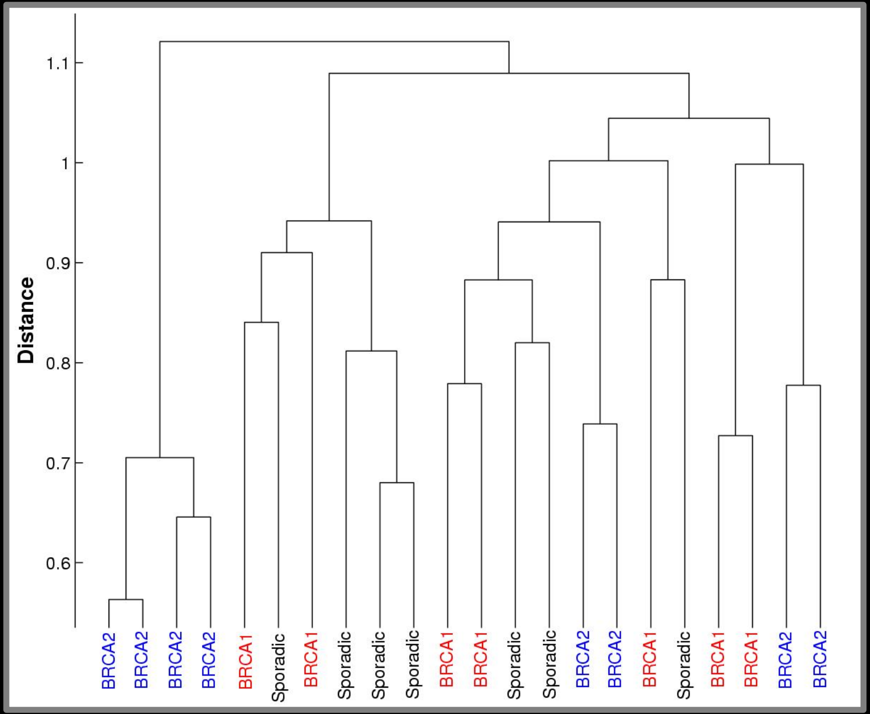
	Cluster ‘centre’	Distance metric	Robustness to outlier	Computation cost
K-means	Mean of points in a cluster	Distance to the cluster mean	Not robust	Usually low
K-medoids	One of the points in a cluster	Any similarity measure	Robust	Much higher

https://scikit-learn-extra.readthedocs.io/en/stable/modules/cluster.html#k-medoids

https://www.researchgate.net/publication/342871651_An_innovative_hybrid_strategy_for_structural_health_monitoring_by_modal_flexibility_and_clustering_methods/figures?lo=1

Hierarchical Clustering Agglomerative

Hierarchical clustering **builds up** clusters based on proximity of instances, ending on reaching predefined number of points

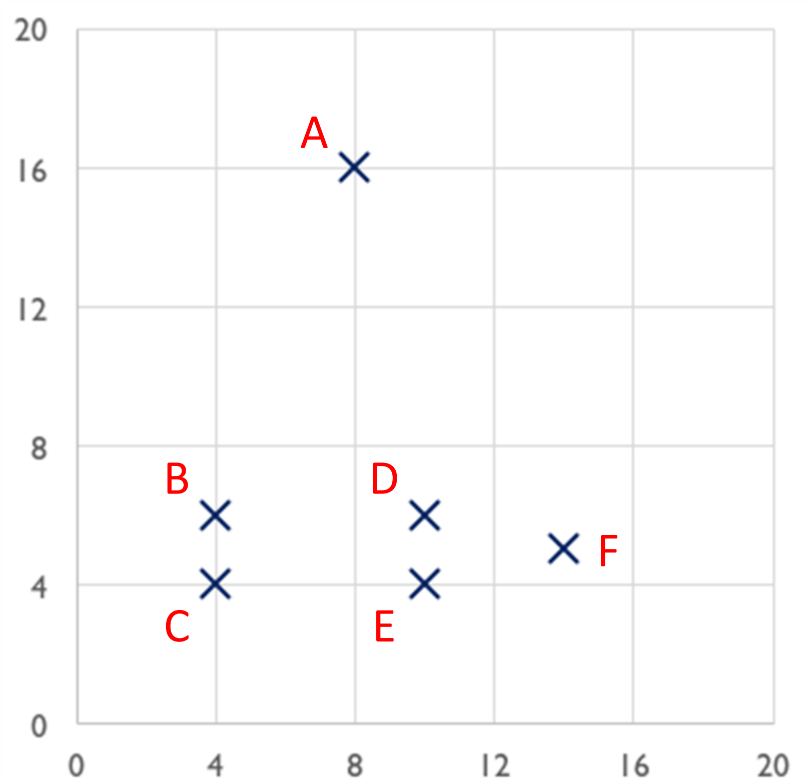


Iterative Algorithm

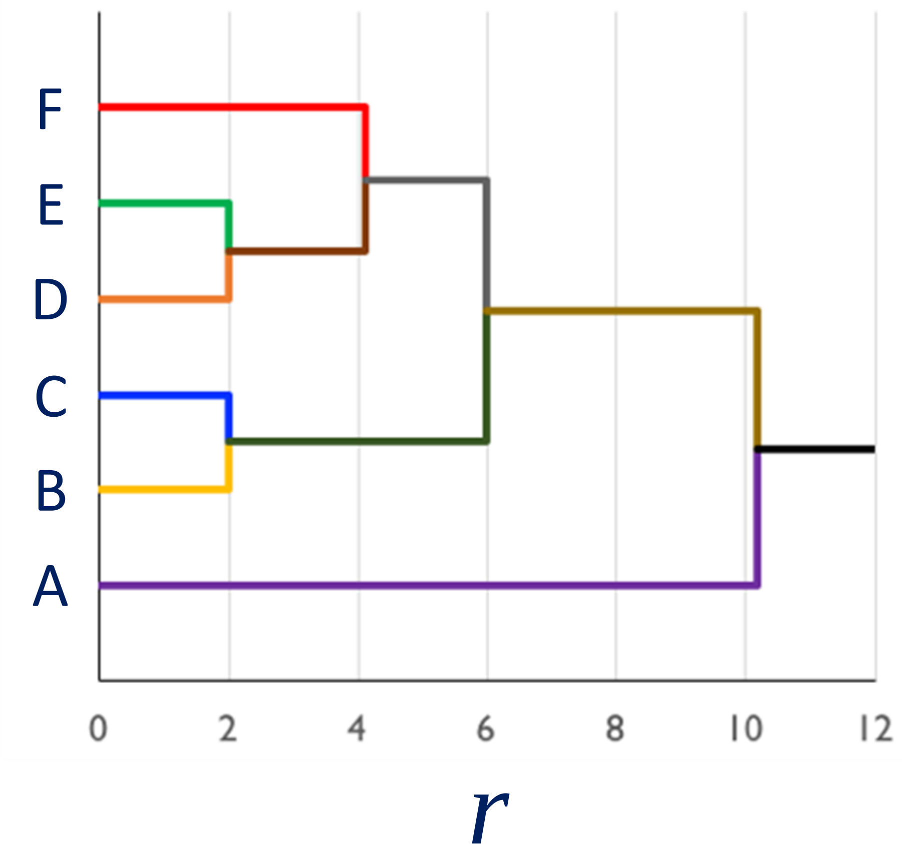
1. Start with every point in its own cluster
2. Merge points according to a *linkage criterion (or distance)*
3. Compute centroid of new clusters
4. Expand linkage threshold and continue until all points in one cluster

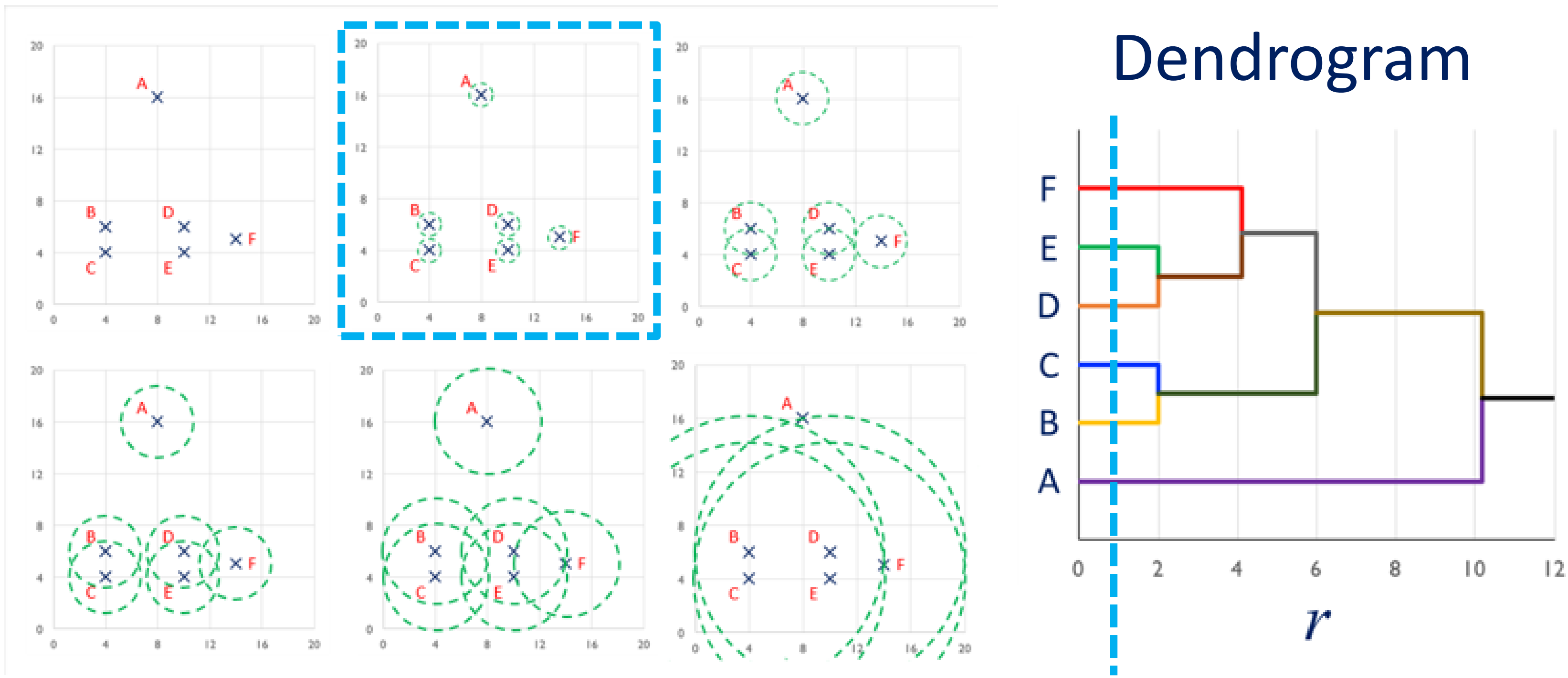
Pros

- Users can choose the level in a hierarchy structure;
- No prior knowledge of data required



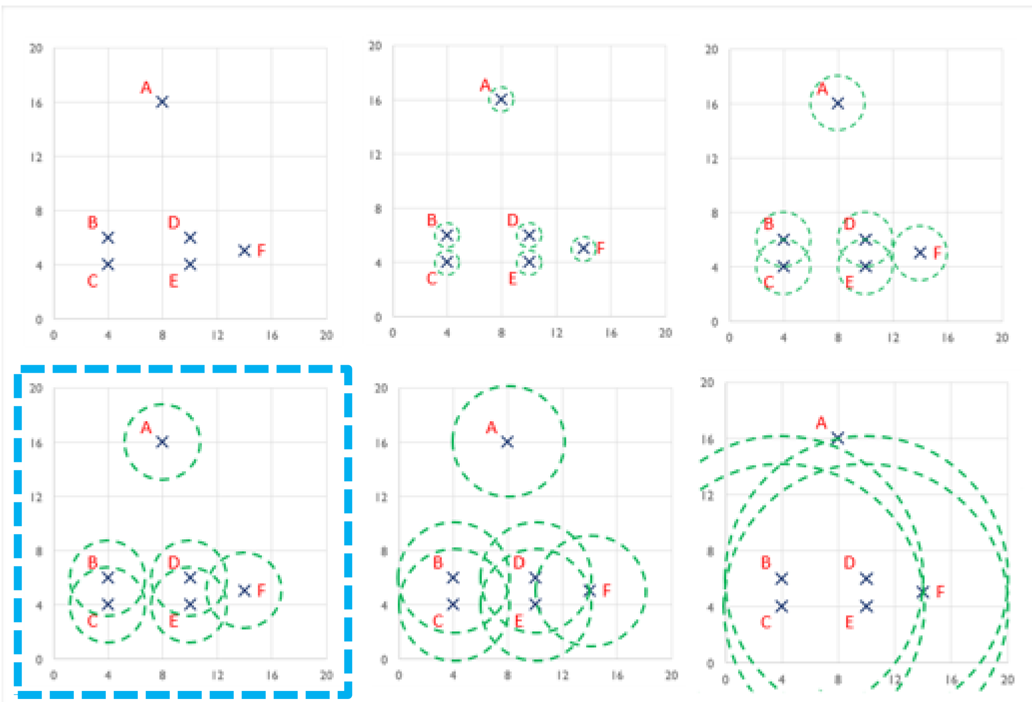
Dendrogram

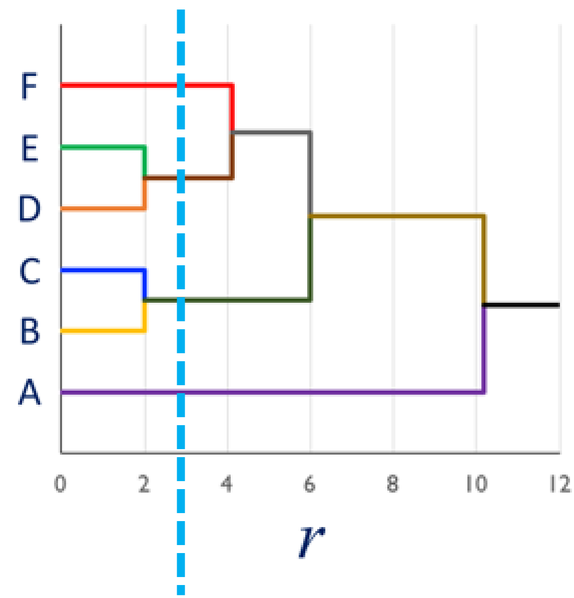




Dendrogram
F
E
D
C
B
A
0
2
4
6
8
10
12
r

Dendrogram





Hierarchical Clustering

Agglomerative

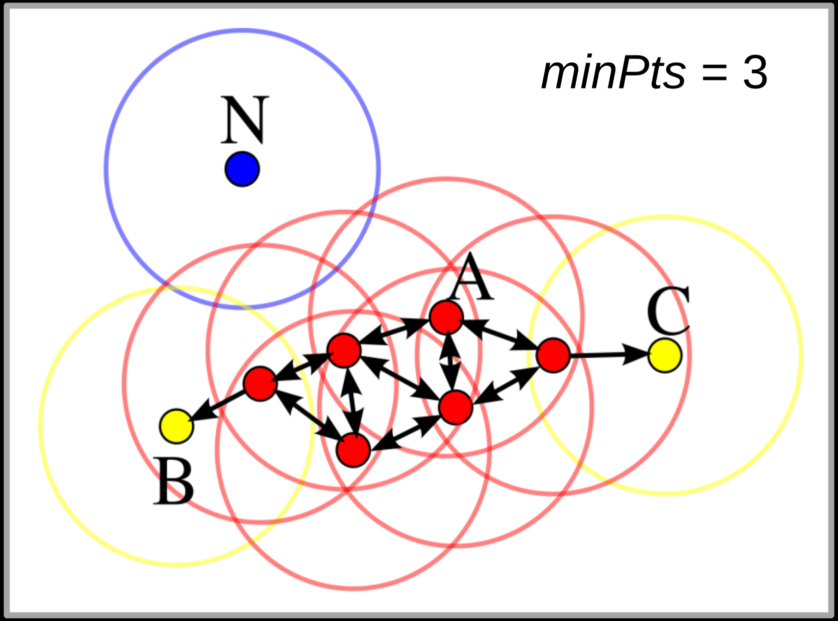
Bottom Up:	Begins with one cluster per data point; Gradually merge into larger clusters.

Divisive

Top Down:	Begins with one big cluster; Gradually split into smaller clusters.

Clustering
Density-based – DBSCAN Clustering

DBSCAN builds clusters of points based on local proximity, considering neighbours within a maximum distance threshold

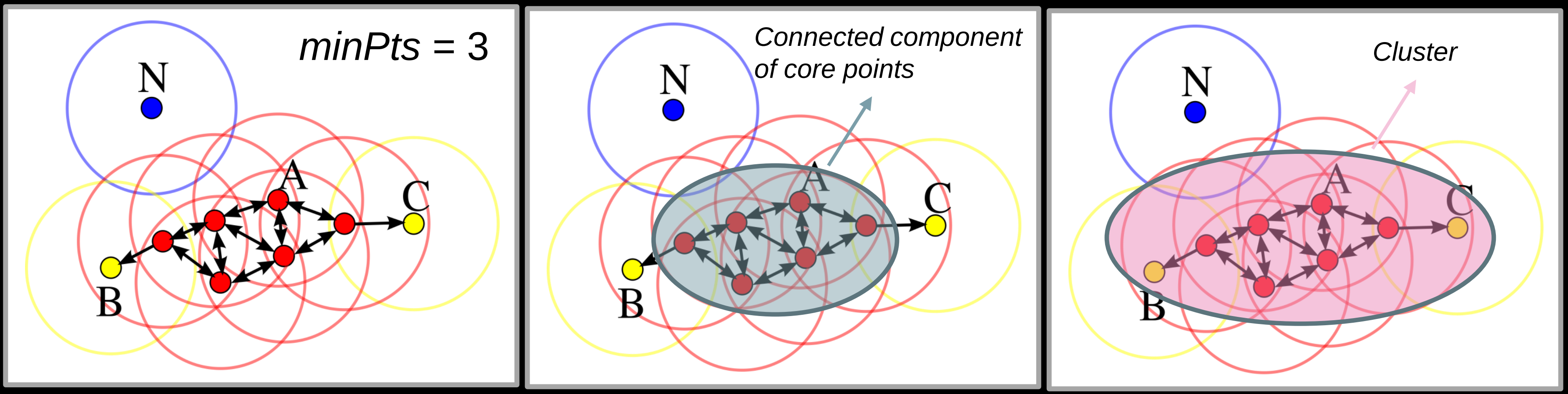


Given ε (search radius) and *minPts*, points are classified into three classes:

1. Point p is **core point**: if at least *minPts* points are within distance ε of it (including p)
2. Point p is **edge point**: if p is not a core point but it is reachable from a core point
3. Point p is **outlier**: all points not reachable from any core points

Clustering
Density-based – DBSCAN Clustering

Step 1 Step 2 Step 3



Process (given ε and *minPts*)

1 Identify core points (with at least *minPts* neighbours)

2 Connect core points while ignoring non-core points (forming connected components)

3 Assign each non-core point to a nearby cluster if it is within ε of a cluster, otherwise assign it to noise

Summary
Three clustering methods

method	required parameters	Extensions
kmeans	*k* (number of clusters)	K-modes, k-medoids, K-prototypes
hierarchical	No required parameter before clustering, but you should decide number of clusters afterwards	*NA*
DBSCAN	ε and *minPts*	*NA*

Choosing a cluster method …

There is no best way. Some issues are important:

1 Ability to cluster at speed for the given data size (the larger data, the fewer choices)

2 Accommodating the data types (numerical, categorical, or mixed)

3 Ability to cope with outliers (if the data have many outliers, then choose a method with robustness to outliers)

4 You can compare different methods and choose the best one

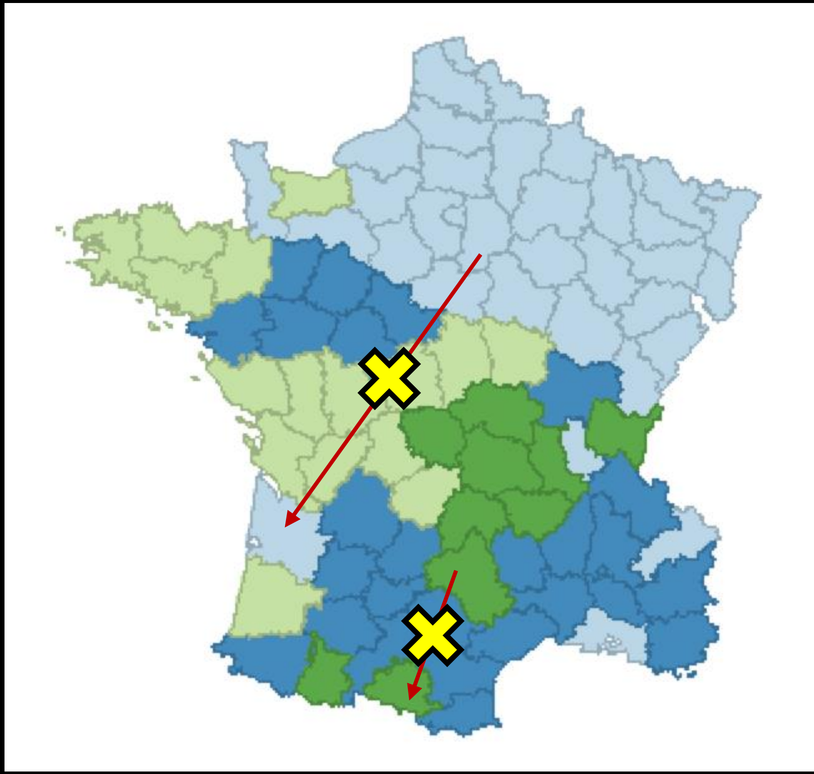
Spatial clustering

- The methods above are *general-purpose* clustering.
- They are applicable to non-spatial variables (e.g. house price, # bedrooms), or spatial variables (e.g. long, lat).
- Note the difference -
  - If you use non-spatial variables for clustering, a cluster represents a type of ‘observations’ with similar attributes (<u>*feature homogeneity*</u> or <u>*attribute similarity*</u>)
  - If you explicitly consider spatial variables in clustering, a cluster represents a ‘place’ or a region (<u>*geographic cohesion*</u> or <u>*spatial contiguity*</u>)

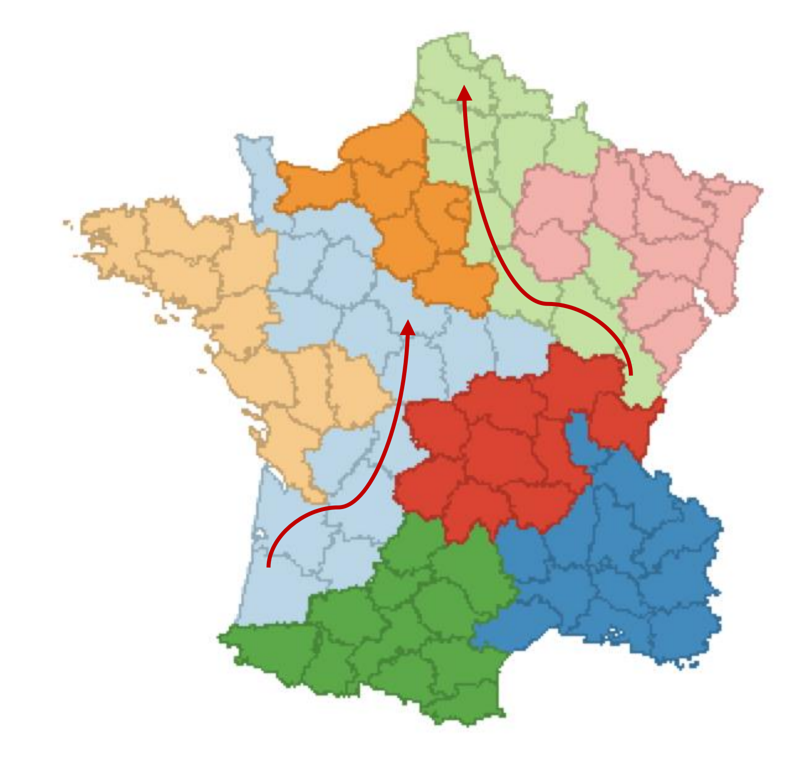
Spatial contiguity / geographic cohesion

A regions is spatial contiguous, if all parts are spatially connected to each other, or if you can travel from one part to another within the region.

None of the clusters are contiguous



All clusters are spatially contiguous



Spatial clustering

Given data points with both non-spatial and spatial variables, there are three approaches to cluster these points:

1. Clustering on only non-spatial variables, and then exploring the geography of clusters;
2. Clustering on non-spatial variables but with constraints of ‘geographic cohesion’;
3. Clustering on both non-spatial and spatial variables*

- Note the trade-off between them.

* Reference: https://doi.org/10.1080/13658816.2021.1934475

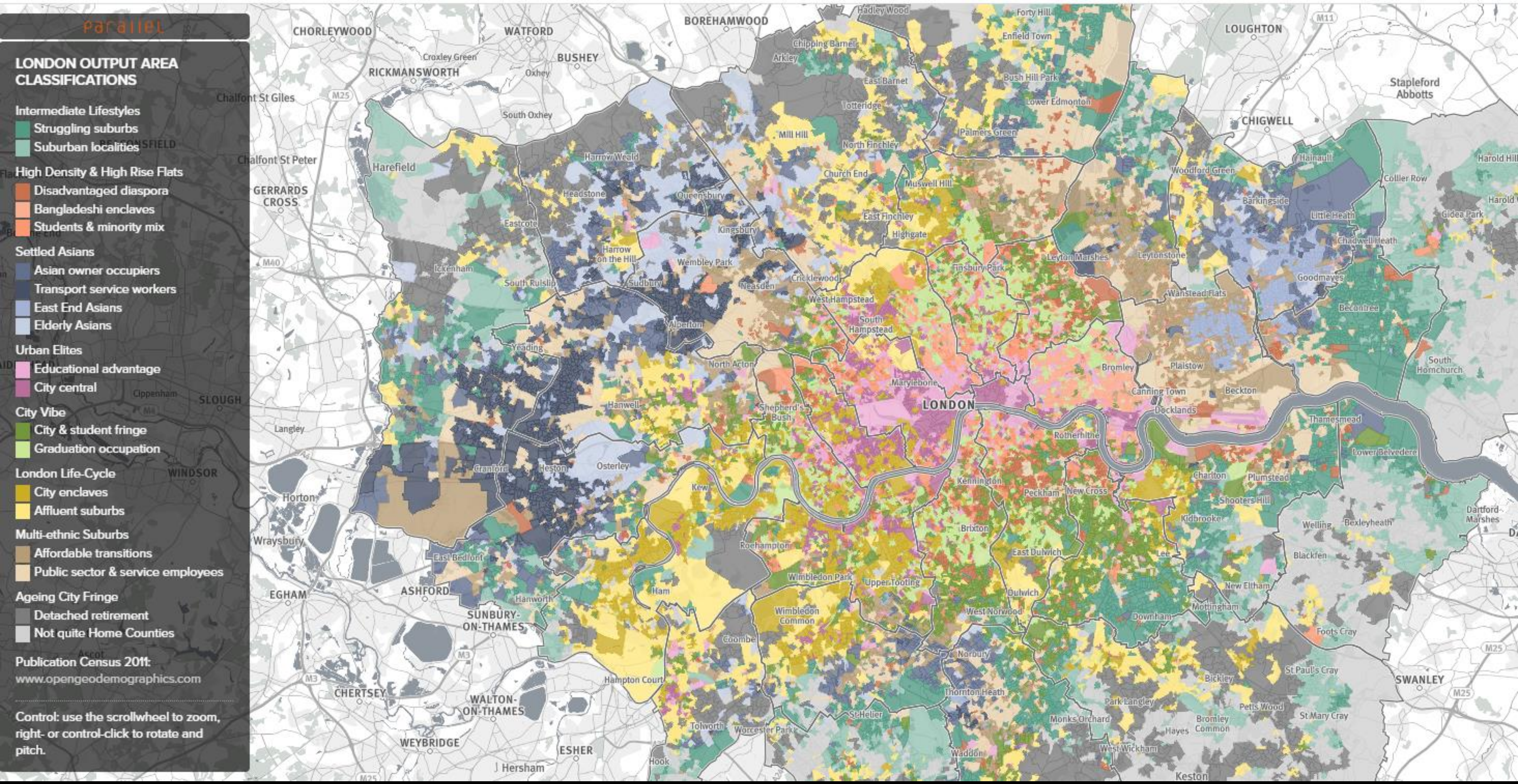
Approach One

- Clustering on only non-spatial variables, and then exploring the geography of clusters;
- *“usually ignores geographical coherence at the outset, but then explores the geography of uncovered solutions” (Wolf, 2021)*
- Example: geodemographic analysis (London OA classification)

- Pros: it works well for geodemographics
- Cons: geographic cohesion is not sufficiently accounted for.

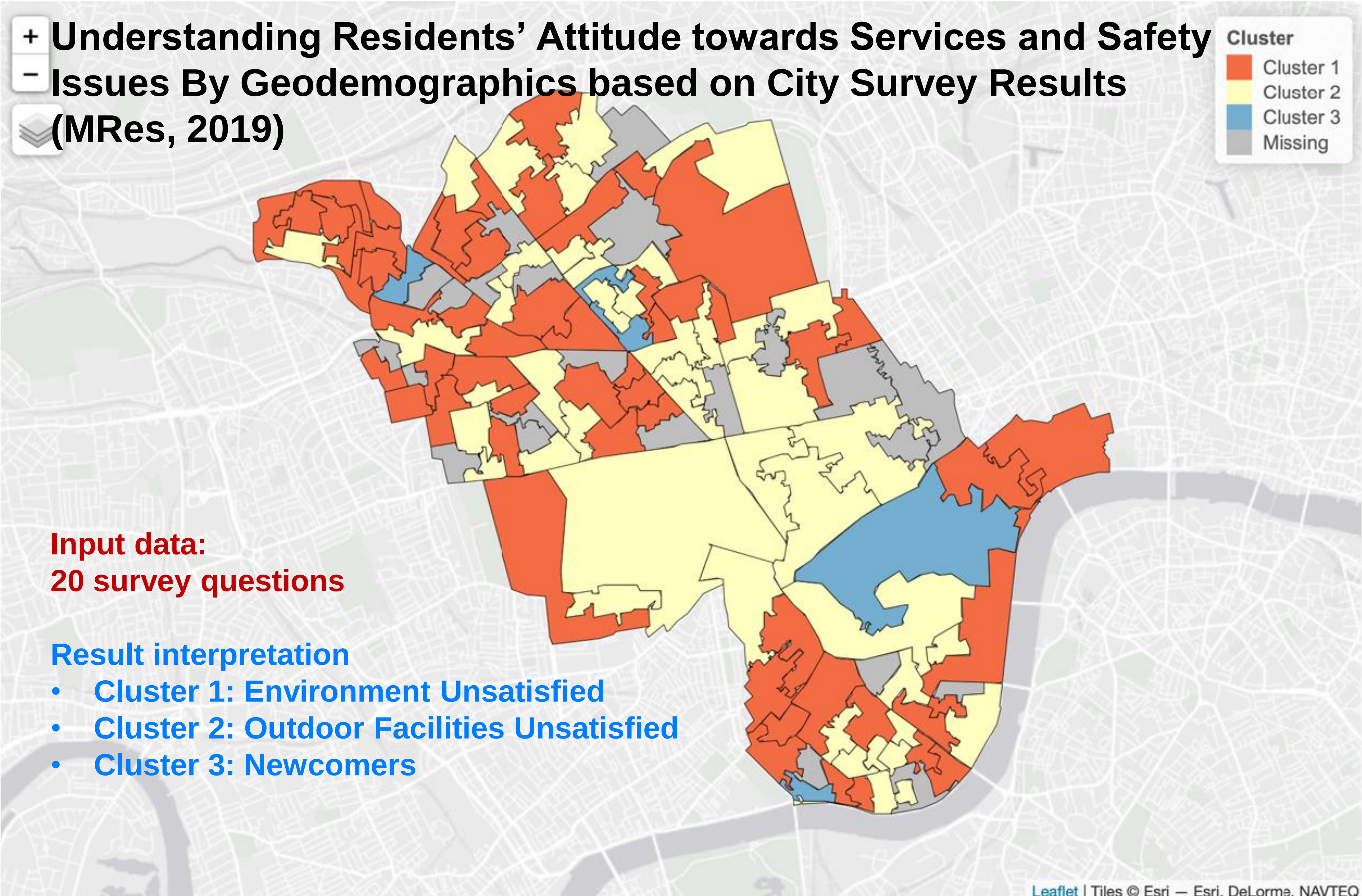
LOAC

1. Clustering OAs on 70+ socio-economic variables (non-spatial); 32000 OAs are clustered into 8 groups
2. Clusters have obvious spatial patterns but aren't spatially contiguous(!!)



https://parallel.co.uk/loac

Understanding Residents' Attitude towards Services and Safety Issues By Geodemographics based on City Survey Results (MRes, 2019)



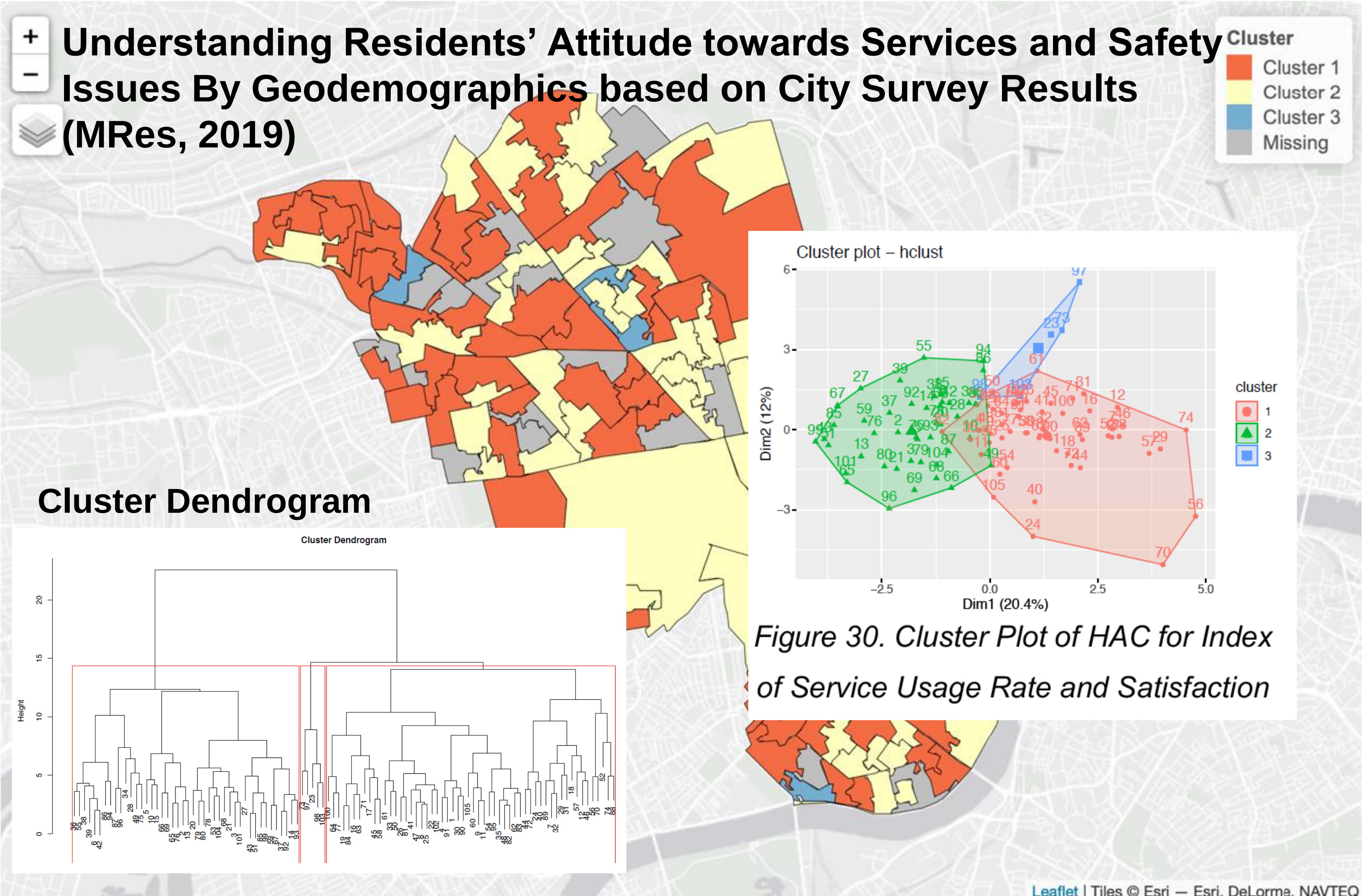
Input data:
20 survey questions

Result interpretation

- **Cluster 1: Environment Unsatisfied**
- **Cluster 2: Outdoor Facilities Unsatisfied**
- **Cluster 3: Newcomers**

Figure 33. Cluster Map of HAC for Index of Service Usage Rate and Satisfaction

Understanding Residents' Attitude towards Services and Safety Issues By Geodemographics based on City Survey Results (MRes, 2019)



Cluster Dendrogram

Figure 30. Cluster Plot of HAC for Index of Service Usage Rate and Satisfaction

Figure 33. Cluster Map of HAC for Index of Service Usage Rate and Satisfaction

Approach Two

- Clustering on non-spatial variables while adding constraints of ‘geographic cohesion’

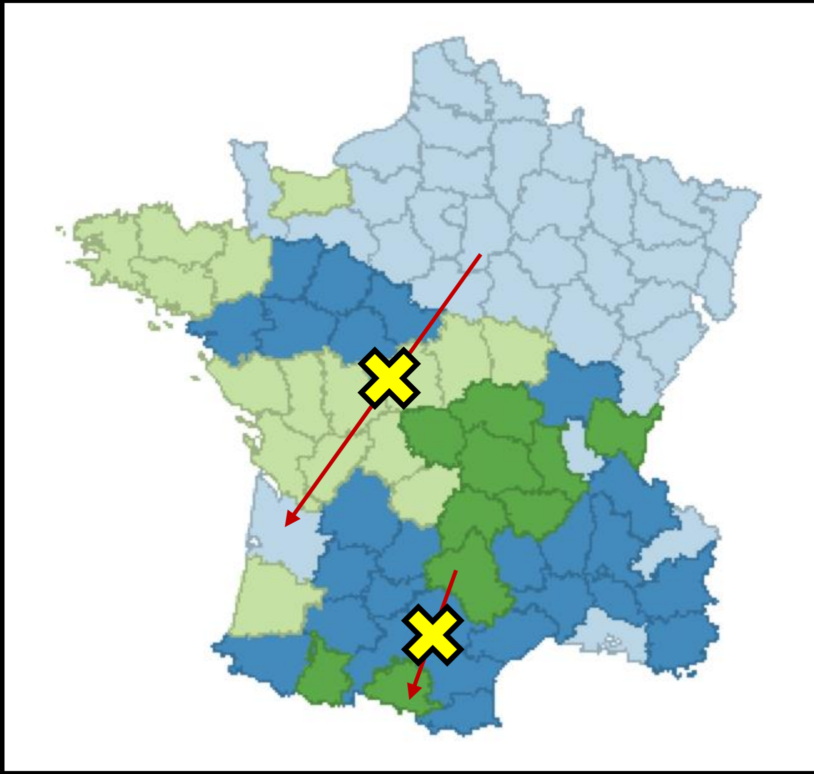
- Also called *Regionalization, districting, spatially constrained clustering* in literature.

- Pros: it simultaneously considers feature homogeneity and geographic cohesion
- Cons: computationally expensive

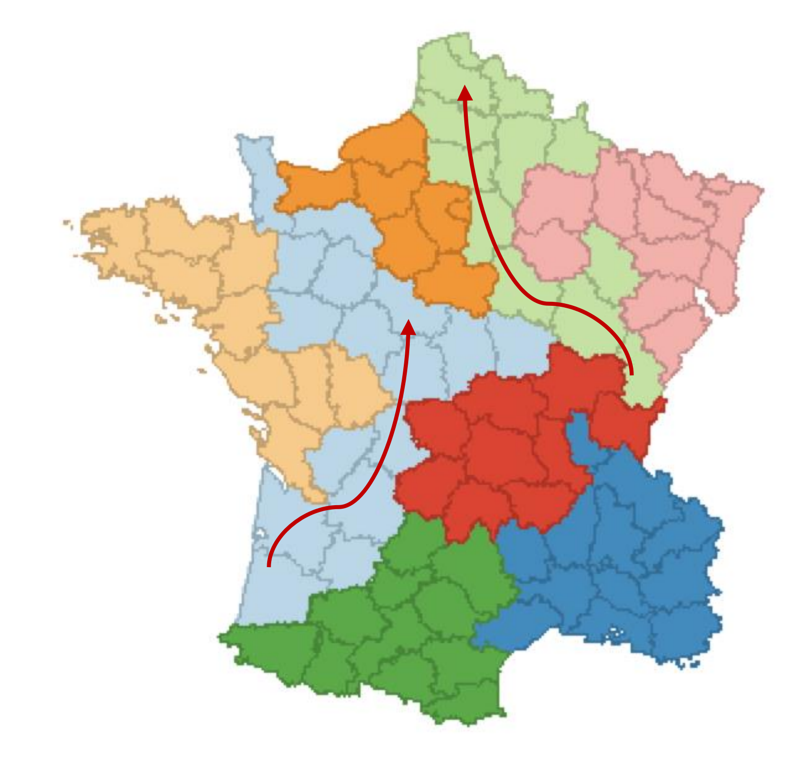
Spatial contiguity / geographic cohesion

A regions is spatial contiguous, if all parts are spatially connected to each other, or if you can travel from one part to another within the region.

None of the clusters are contiguous



All clusters are spatially contiguous



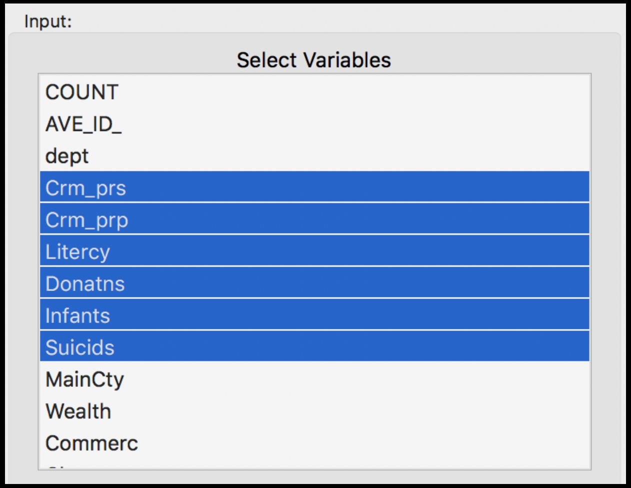
The max-p method

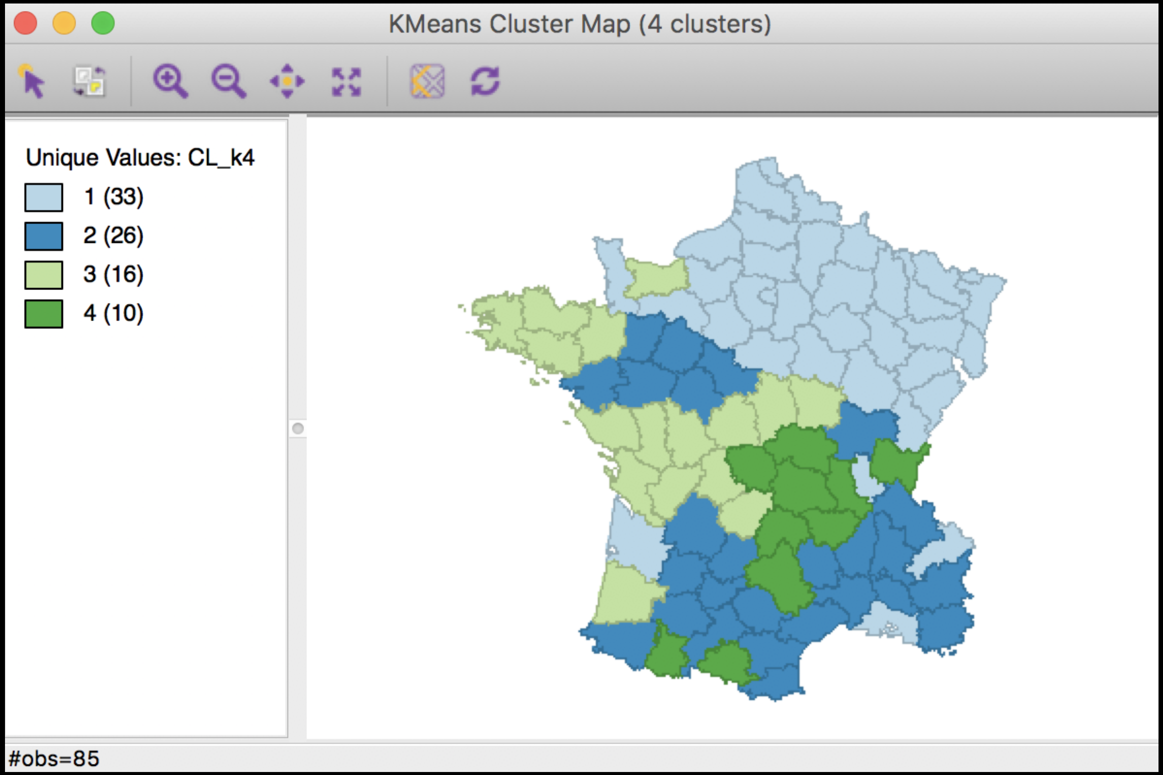
- Max-p: clustering of a set of geographic areas into the maximum number of regions such that the value of each region (e.g. population) is above a predefined threshold value

- Hyperparameter: the predefined threshold value
- The number of clusters or the maximum number of clusters is not predefined

Comparing kmeans and max-p

- Case study: Guerry data set on moral statistics in 1830 France
- Method 1: generic kmeans (6 attributes, without considering geometric centroids), k=4. None of the clusters is spatially contiguous.

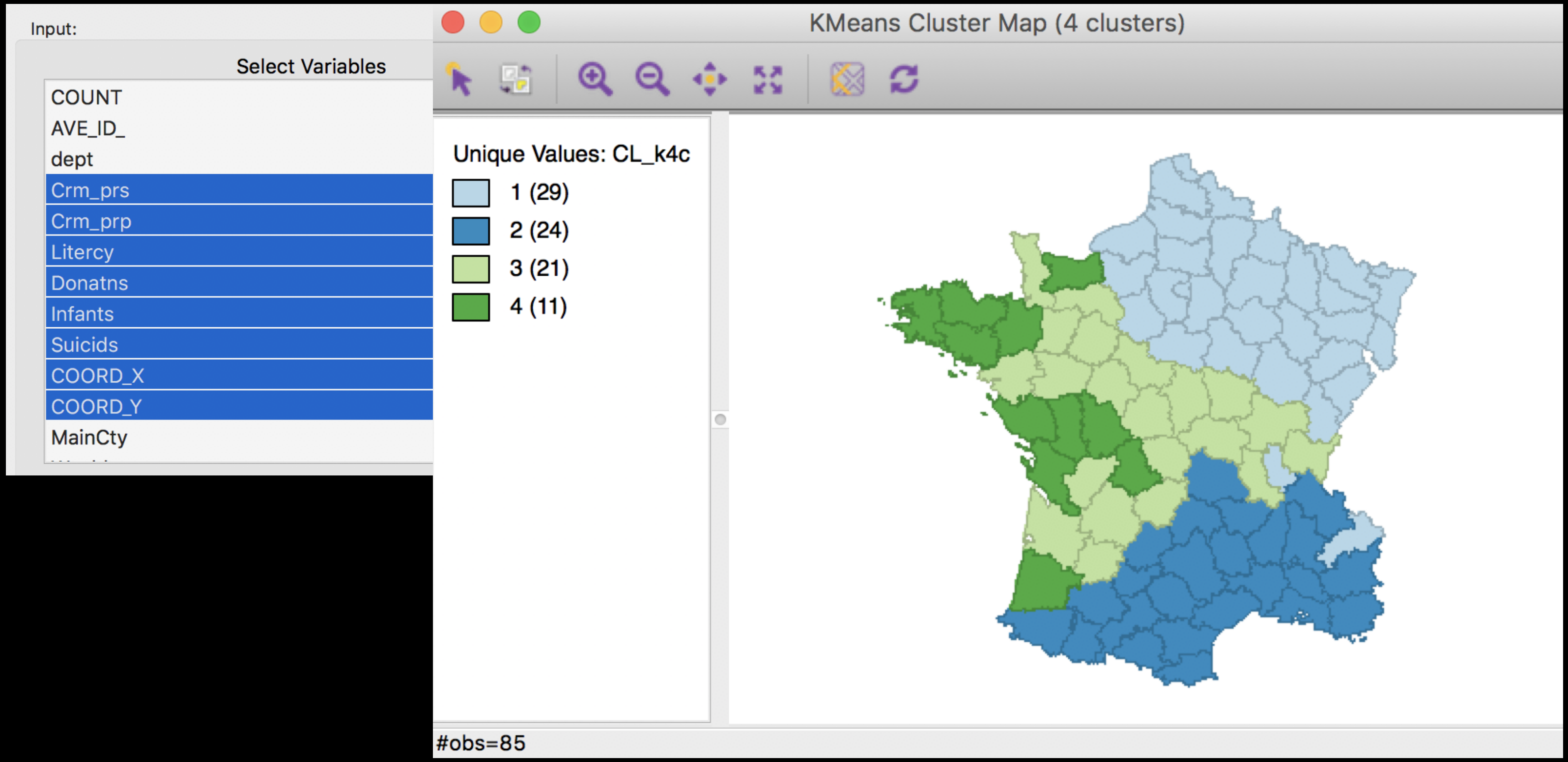




https://geodacenter.github.io/workbook/8_spatial_clusters/lab8.html

Comparing kmeans and max-p

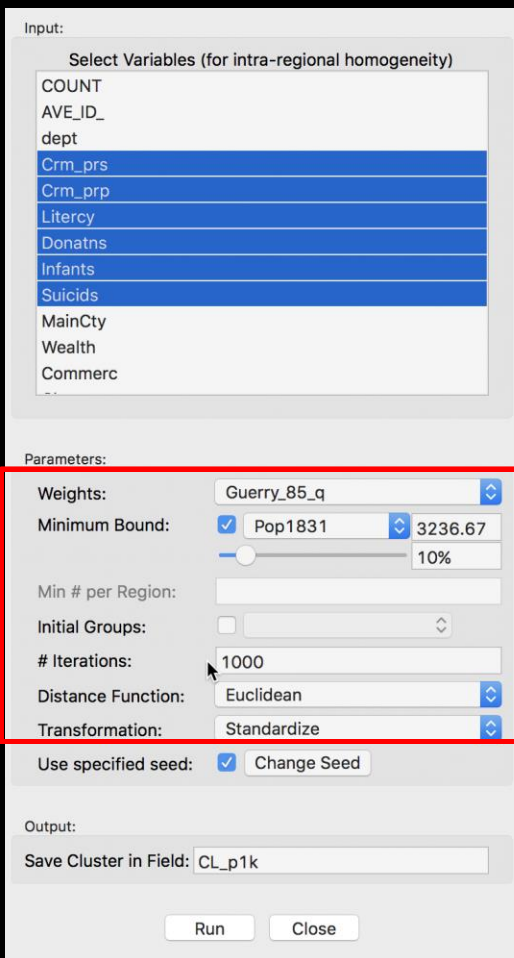
- Method 2: kmeans with centroids included as variables (8 attributes).
- Group 2 and 3 achieve contiguity. group 1 consists of three parts (including two singletons), and group 4 consists of four parts (including two singletons)



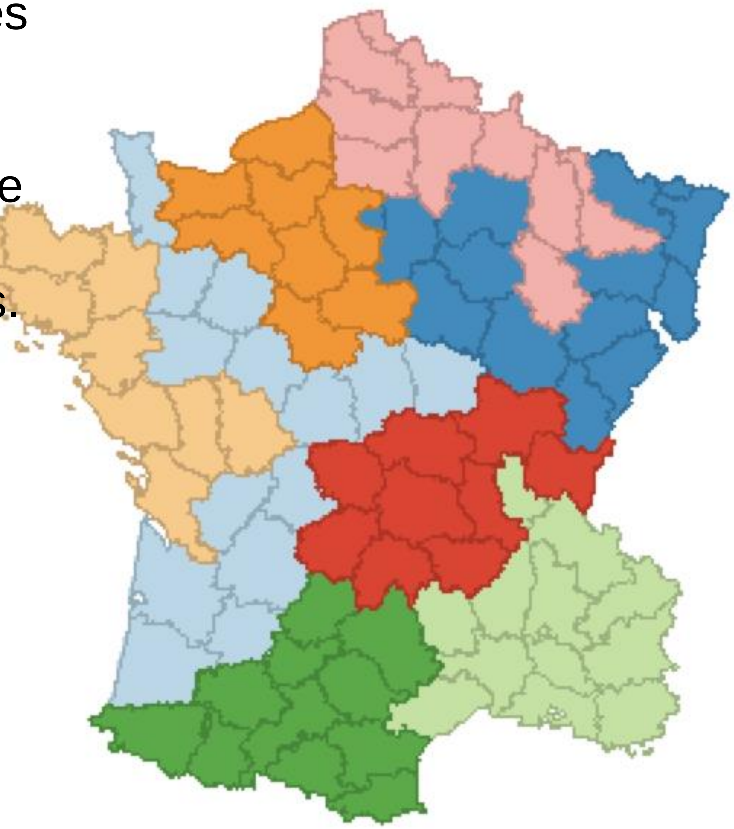
UCL

Comparing kmeans and max-p

- Method 3: max-p method (each region has at least 10% of total pop)



It generates 8 clusters, and all clusters are spatially contiguous.



Measuring Clustering Quality

Necessary when...

- **Comparing different implementations of a clustering method with randomness (e.g., k-means)**
- **Comparing clustering with different parameters (e.g., numbers of clusters)**
- **Comparing different clustering methods**

Method 1: SEE / Elbow Method

- SSE: Sum of Squared Errors

$$SSE = \sum_{i=1}^{n}\sum_{j=1}^{k} w^{(i,j)} \mathrm{dist}(x^{(i)}, \mu^{(j)})$$

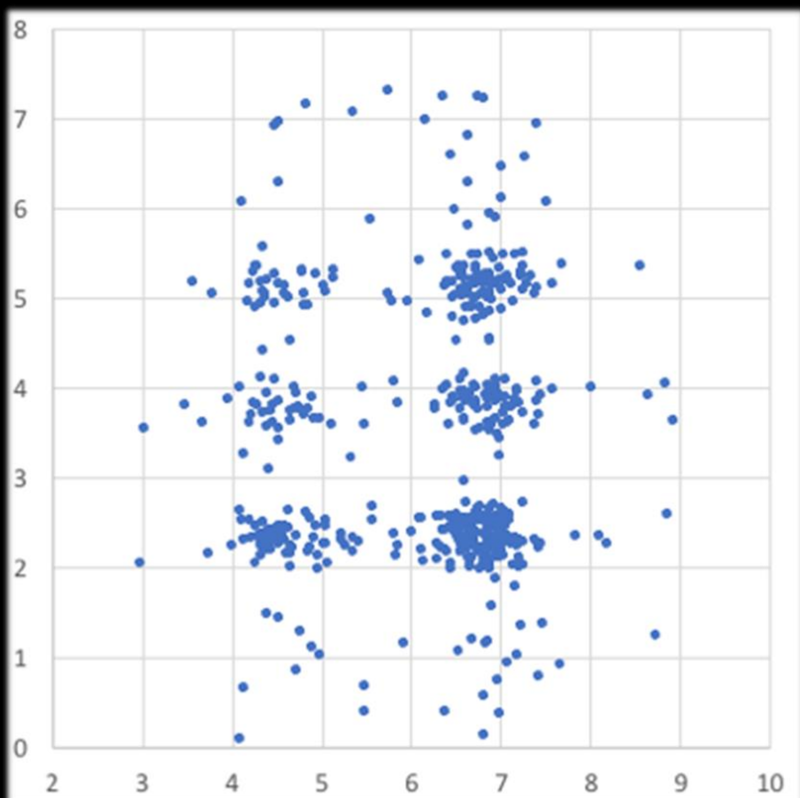
Where: *i* is a data point, *j* is a cluster, and $\mu^{(j)}$ is the centre of a cluster. w(*i*,*j*)=1 when *i* is in cluster *j*, otherwise 0.

- The range of SSE? [0, infinity)
- A small SSE means that the data points are close to cluster centre and the clustering has good performance.

Method 1: SEE / Elbow Method

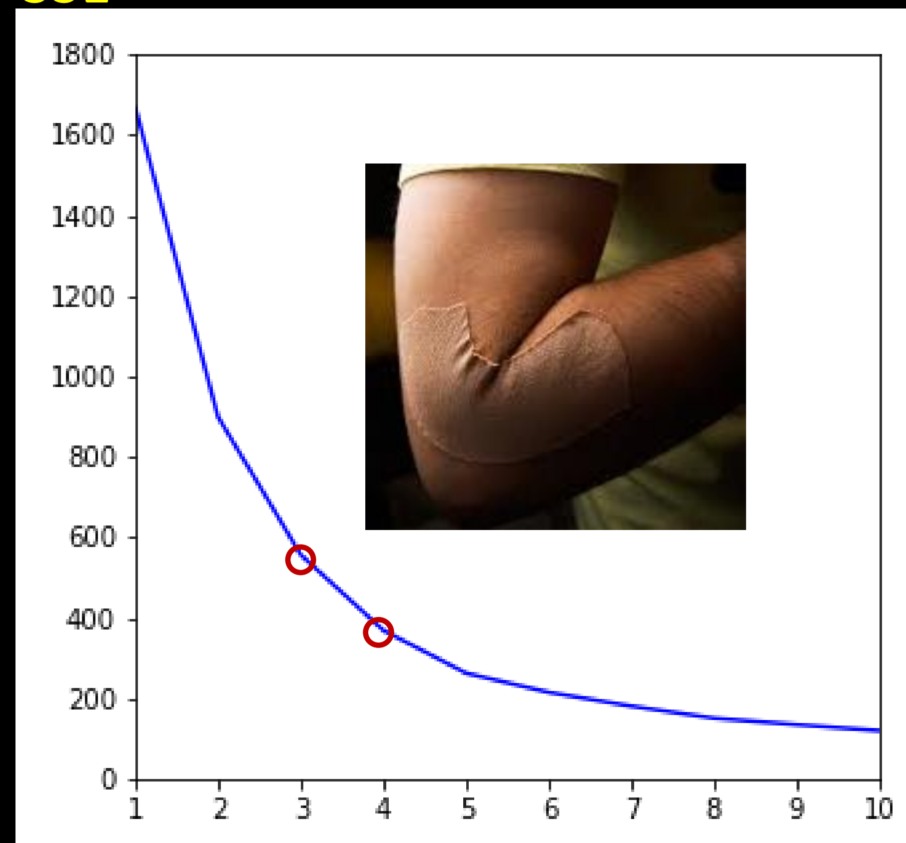
Elbow diagram: help choose k for k-means

Y



X

SSE



k (Number of Clusters)

Method 2: Silhouette Analysis

Silhouette of a point

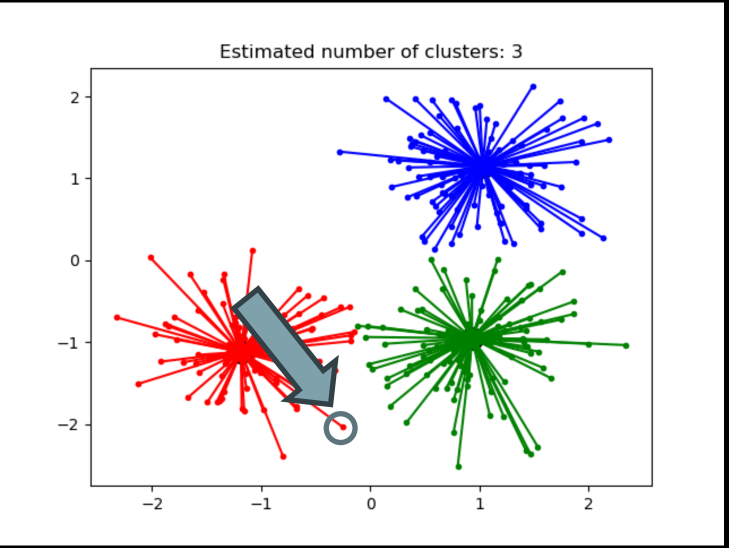
"Is this point closer to points of the same cluster, or another cluster?"

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

$a(i)$: mean distance to points of the same cluster
$b(i)$: minimum mean distance to points of another cluster

$$-1 \leq s(i) \leq 1$$

The larger s(*i*), the higher clustering quality



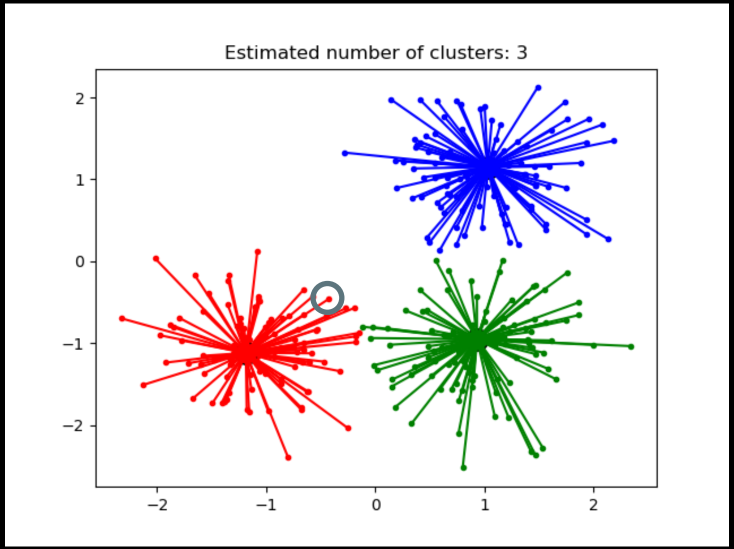
$a(i)$: mean distance to points in RED
$b(i)$: mean distance to points in green (as this point is closer to GREEN cluster than BLUE cluster)

Method 2: Silhouette Analysis

Silhouette of a point

"Is this point closer to points of the same cluster, or any other cluster? "

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

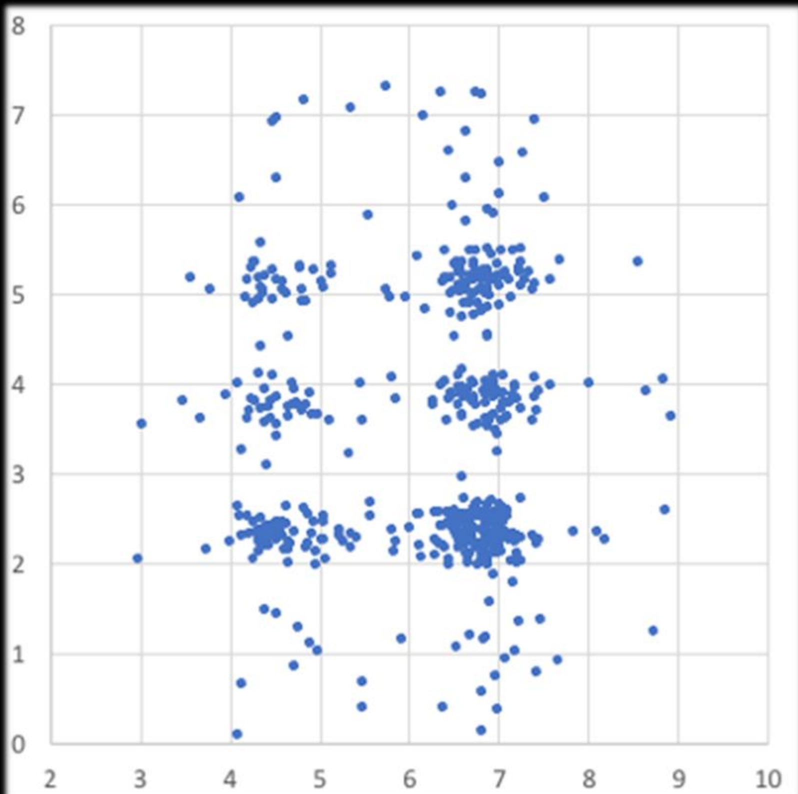


Silhouette Score for a Clustering = *Average of s(i) for all points i*

Method 2: Silhouette Analysis

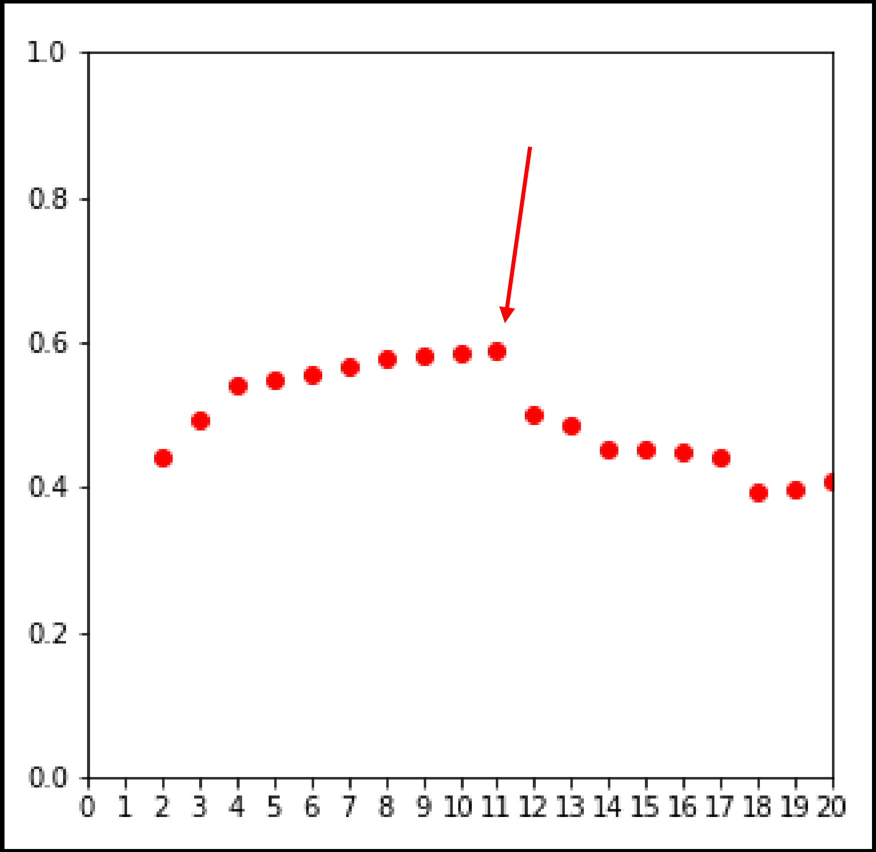
Choose k for k-means

Y



X

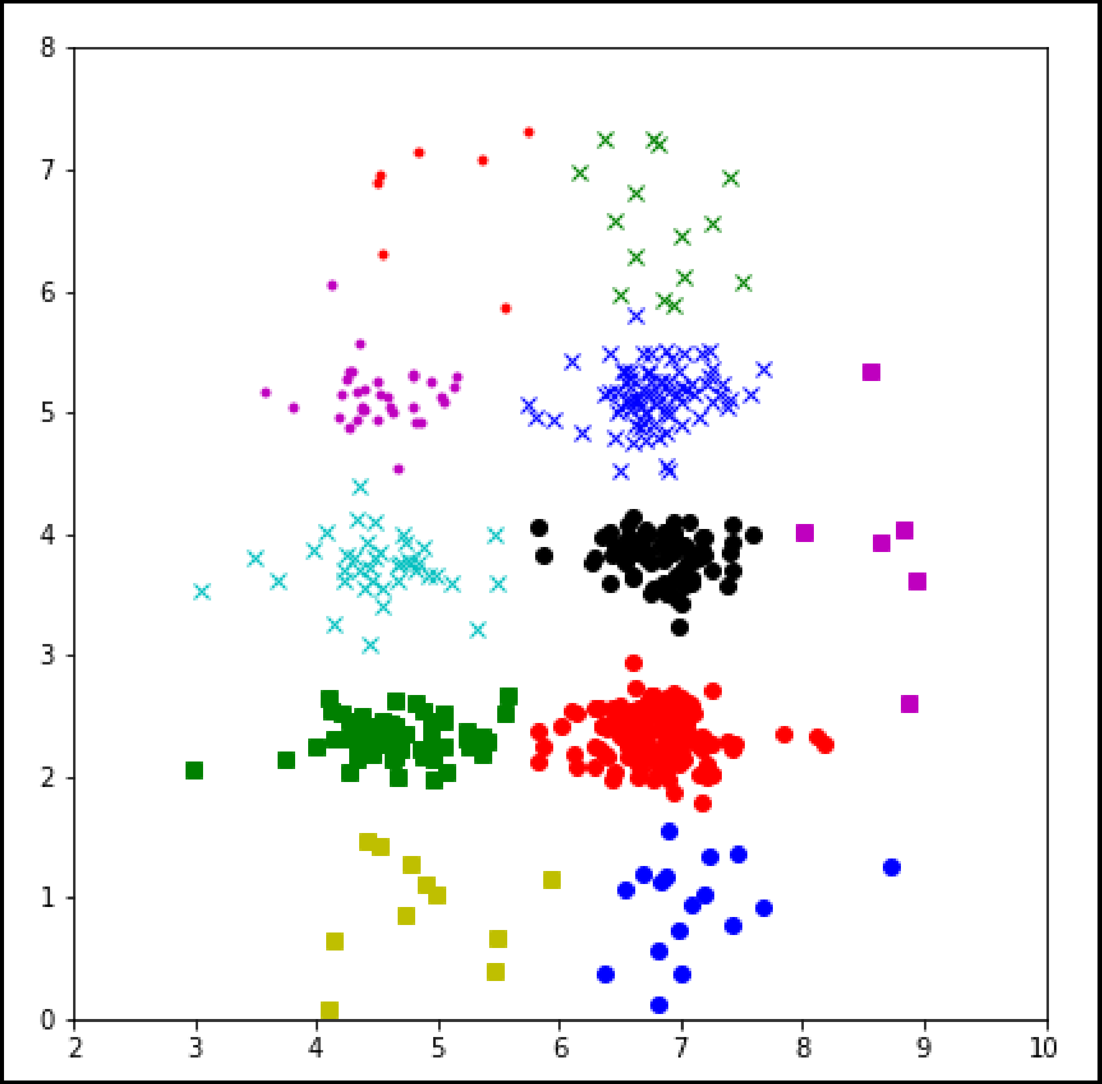
Silhouette Score



k (Number of Clusters)

Method 2: Silhouette Analysis

Y



X

'Optimal' k-Means

k = 11
S. Score = 0.59

Selecting hyperparameters

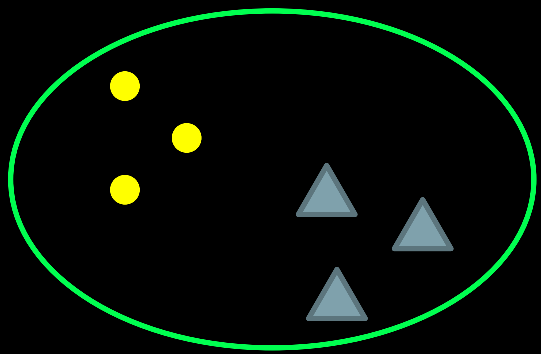
- You might get different k values if you use SSE/Elbow Method and Silhouette analysis to select the k hyperparameter of k-means
- This is quite normal
- You can decide which k value to use, depending on your preference on these two methods, or the observations.

Method 3: Comparing against ‘ground truth’

Homogeneity All clusters contain only points from a single observed class – expressed as a proportion of clusters for which this is true

Completeness All members of given class are within the same cluster – expressed as a proportion of classes for which this is true

Scenario 1



	C1	C2	Average
Homogeneity	1	1	1
	OC1		**Average**
Completeness	0		0

Clustering
C1
C2

Observed class (ground truth)
OC1

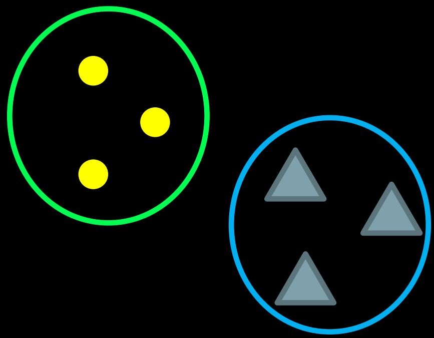
https://scikit-learn.org/stable/modules/clustering.html#homogeneity-completeness

Method 3: Comparing against 'ground truth'

Homogeneity All clusters contain only points from a single observed class – expressed as a proportion of clusters for which this is true

Completeness All members of given class are within the same cluster – expressed as a proportion of classes for which this is true

Scenario 2



	C1	C2	Average
Homogeneity	1	1	1
	OC1	**OC2**	**Average**
Completeness	1	1	1

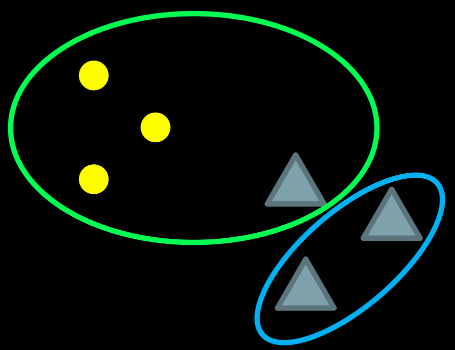
https://scikit-learn.org/stable/modules/clustering.html#homogeneity-completeness

Method 3: Comparing against 'ground truth'

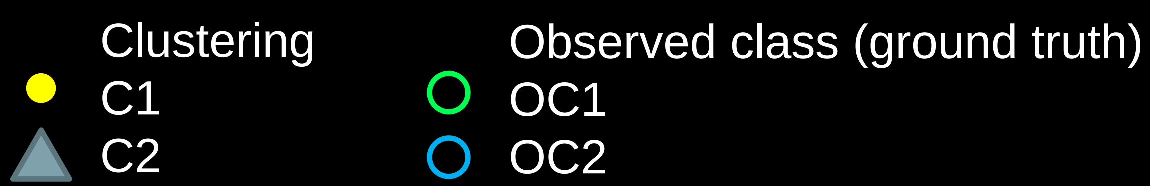
Homogeneity — All clusters contain only points from a single observed class – expressed as a proportion of clusters for which this is true

Completeness — All members of given class are within the same cluster – expressed as a proportion of classes for which this is true

Scenario 3



	C1	C2	Average
Homogeneity	1	0	0.5
	OC1	**OC2**	**Average**
Completeness	0	1	0.5



https://scikit-learn.org/stable/modules/clustering.html#homogeneity-completeness

Method 3: Comparing against ‘ground truth’

- Where is the ‘ground truth’ from?
  - You have some ground truth available;
  - The ‘ground truth’ can come from a different but relevant task. Should prove these tasks are relevant.
  - You can ask some domain experts for their opinions
    - *Do you think these results make sense in practice?*
    - *What improvements are possible or needed?*

Next steps of clustering

- Visualisation (often combined with dimension reduction, e.g. PCA)
- Qualitatively describe cluster characteristics
- Mapping the clusters
- Compare against expert knowledge

Thank You

Questions?

Huanfa Chen

huanfa.chen@ucl.ac.uk

Workshop
Dimension reduction

- Weekly quiz on Moodle: please finish them before the workshop and we will discuss the quiz in the workshop
- Python notebooks for workshop: will be ready by 5pm Thursday.
- See you in the workshop on Friday 1-3pm