Using Machine Learning Tools

University of Adelaide

Unsupervised ML

So far we have looked at supervised methods:

- classification
- regression

Both involve having data values ***and labels***

When labels are not available we use unsupervised methods

Tasks done without labels include:

- clustering
- anomaly detection
- visualisation
- dimensionality reduction

Unsupervised ML: Overview

Clustering Methods

Hierarchical Clustering

Data Visualisation

Dimensionality Reduction

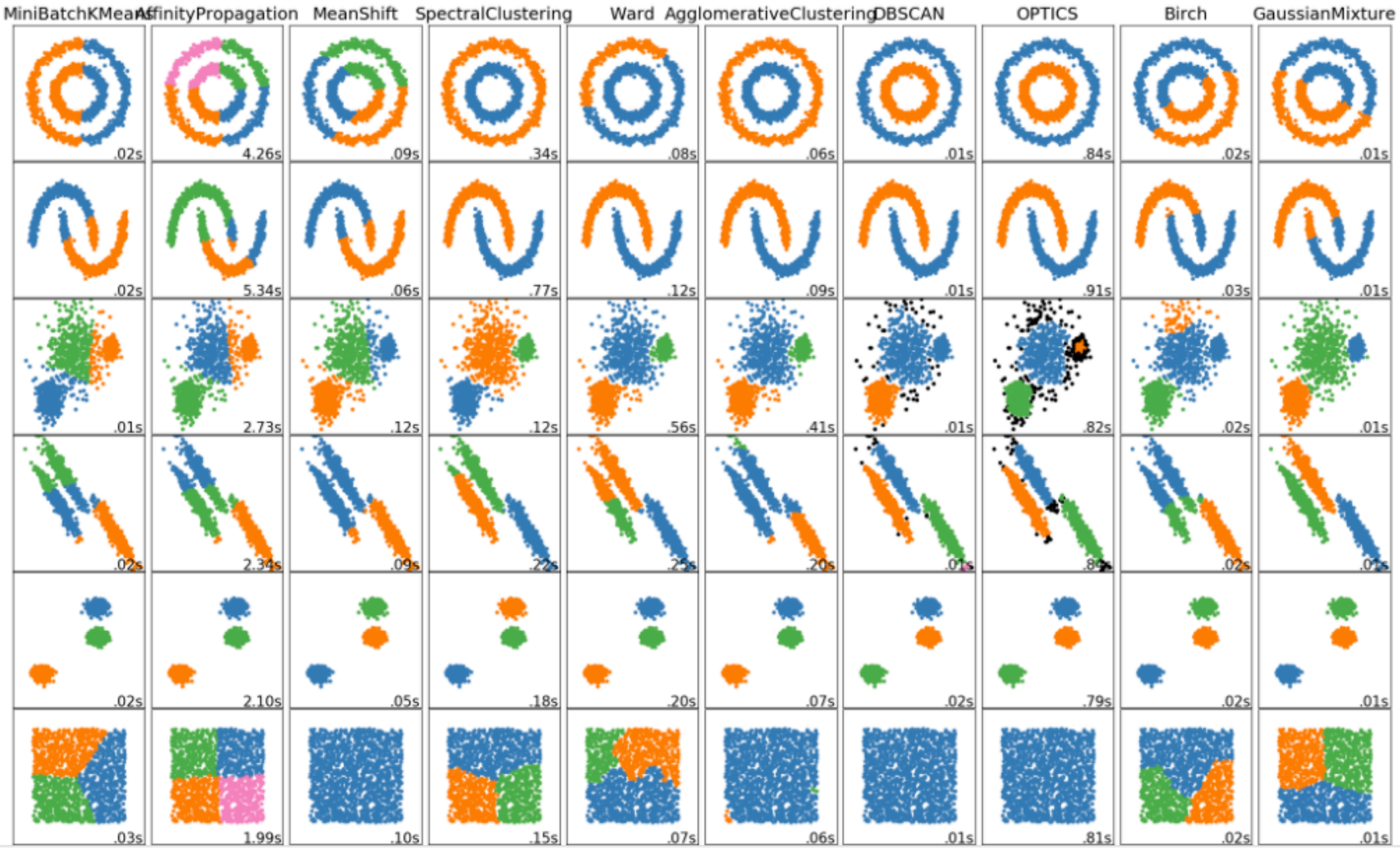
Clustering

- There are many different types of clustering method

- They are based on a number of different underlying principles:
  - explicit models of probability density
  - within vs between group distances
  - progressive splitting
  - progressive joining (agglomeration)
  - local connectivity

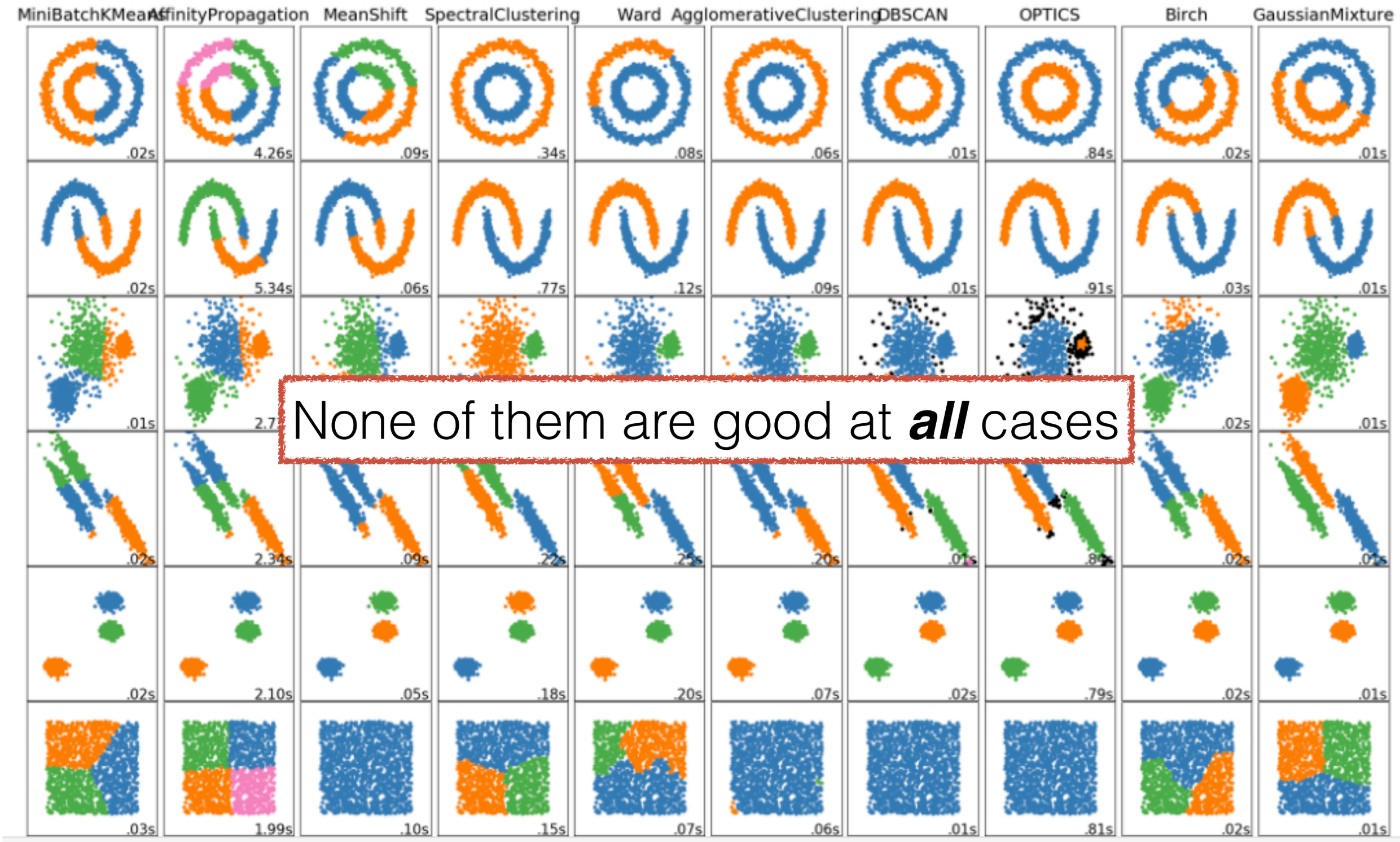
- Each has different strengths and weaknesses
- Difficult to know what to choose to begin with
- Good performance will depend on the task/application

Clustering



https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

Clustering



https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

Clustering

- We will consider a couple of methods in more detail to get some intuition into the principles.

- We will not be able to look at all methods or to go into mathematical detail.

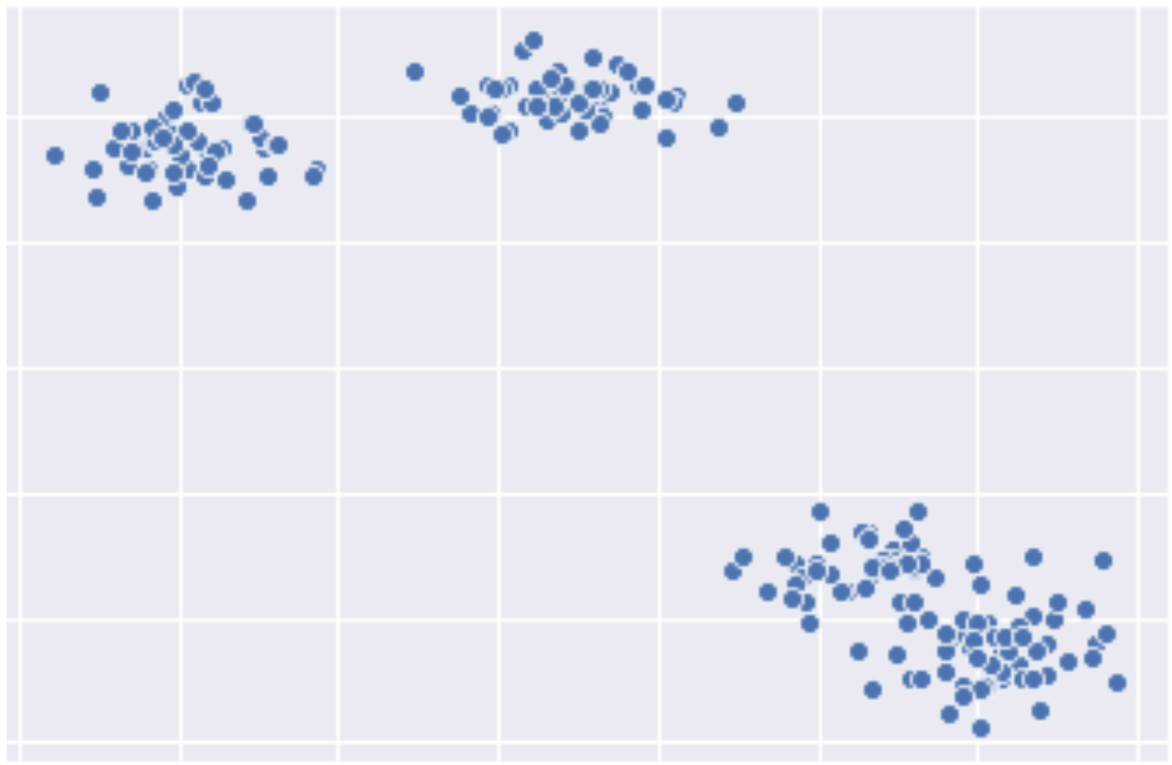
- It is not necessary to understand the inner workings to use the methods.

- The trick when using clustering is to be able evaluate performance in order to choose the best method and appropriate parameters.

- Cannot necessary use the mathematical function within any one method to evaluate, as that is circular.

K-Means

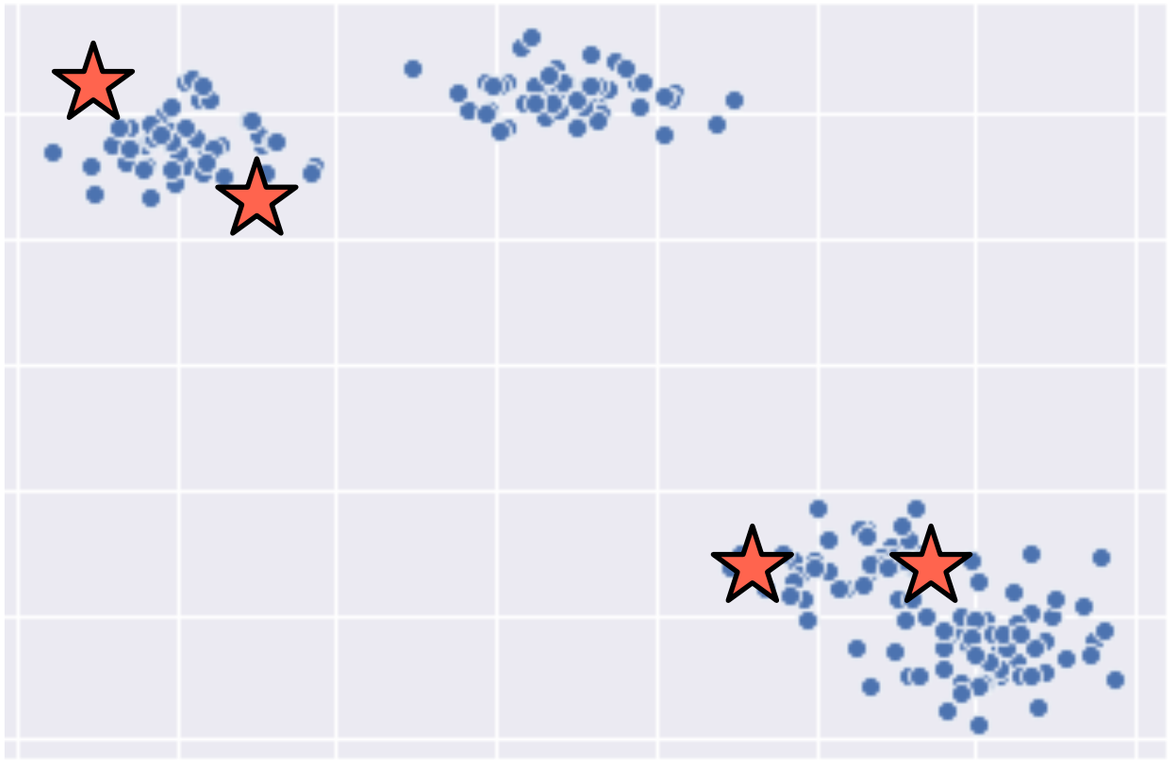
Based on minimising within-cluster distances (to centroids)



K-Means

Based on minimising within-cluster distances (to centroids)

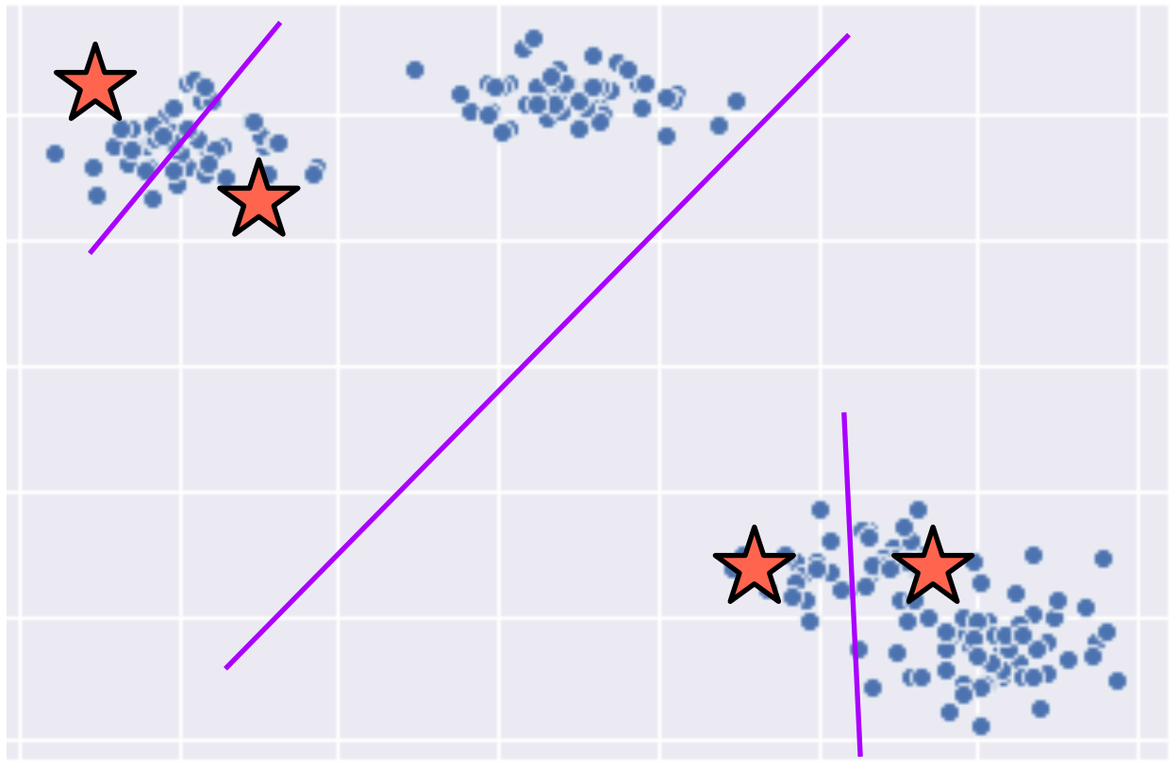
Start by choosing centroids (initialisation)



K-Means

Based on minimising within-cluster distances (to centroids)

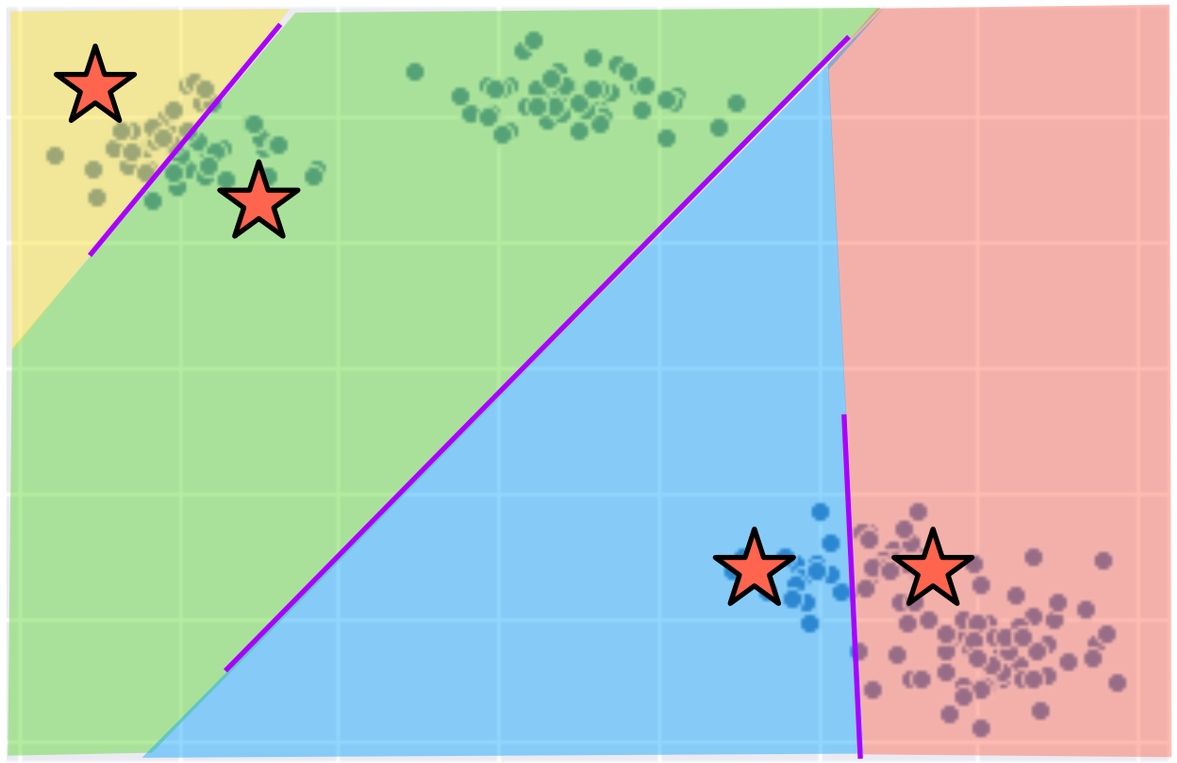
Assign points to clusters according to closest centroid



K-Means

Based on minimising within-cluster distances (to centroids)

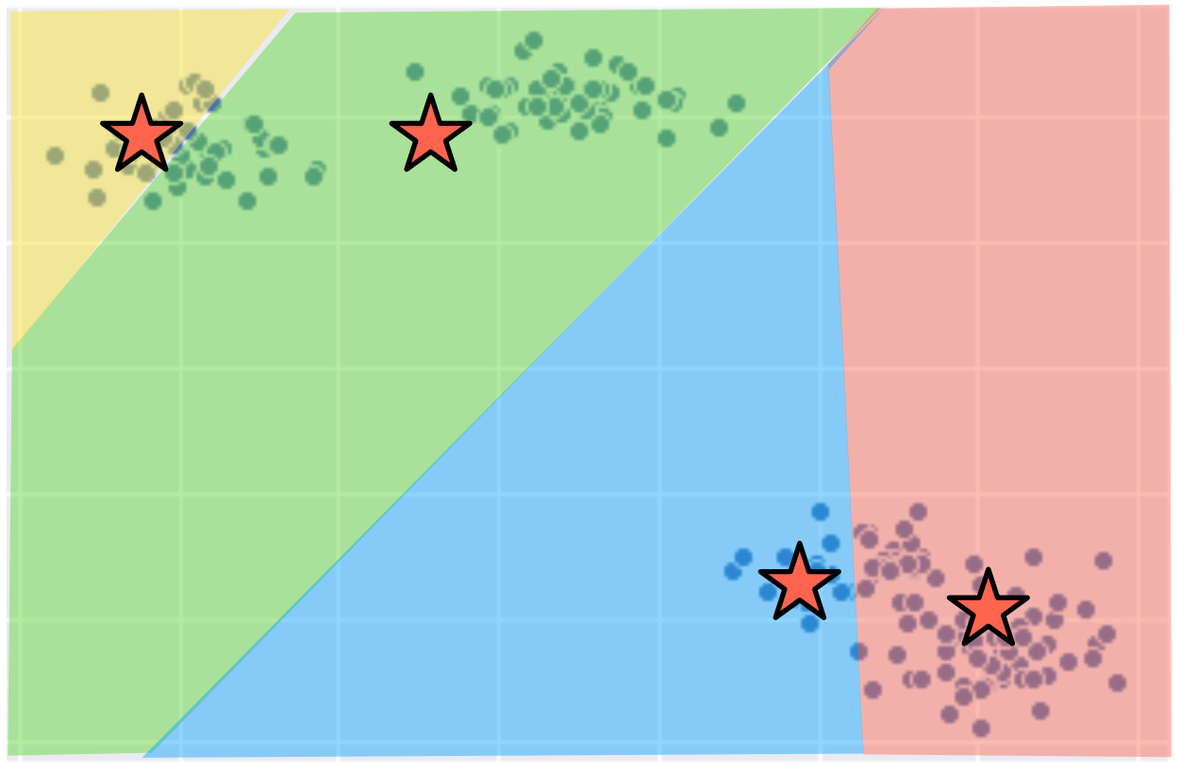
Assign points to clusters according to closest centroid



K-Means

Based on minimising within-cluster distances (to centroids)

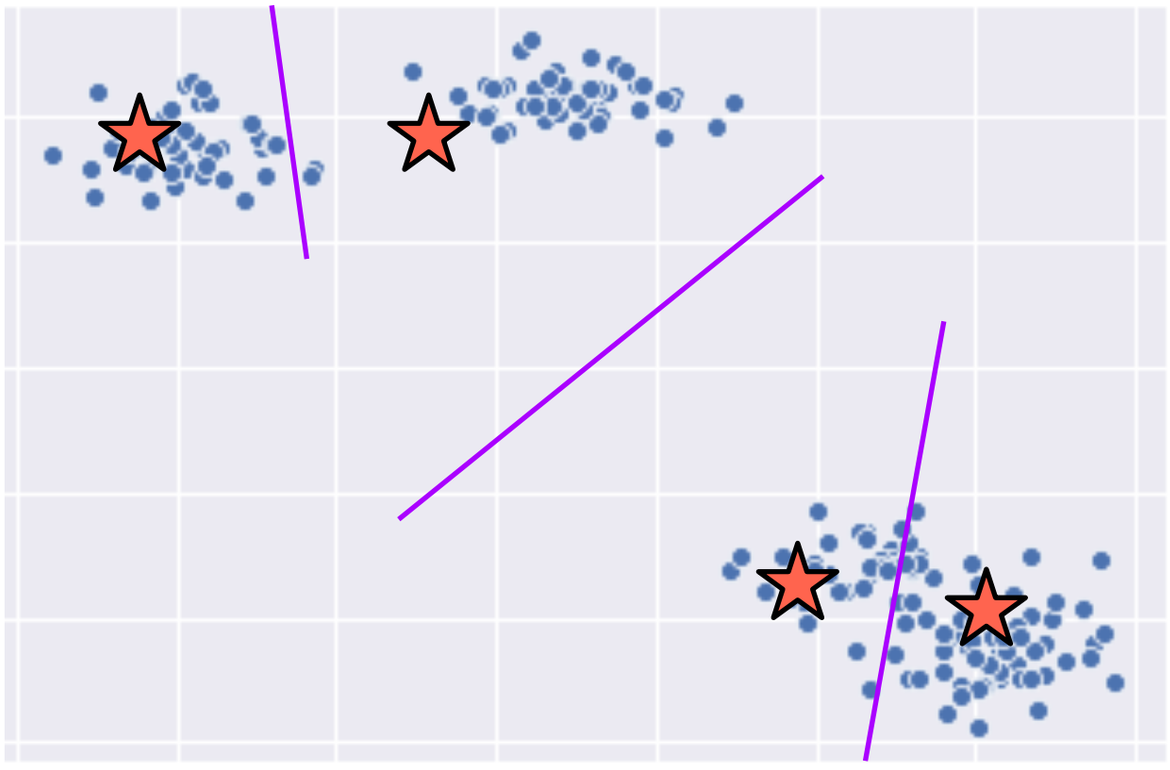
Recalculate centroids based on current cluster assignment



K-Means

Based on minimising within-cluster distances (to centroids)

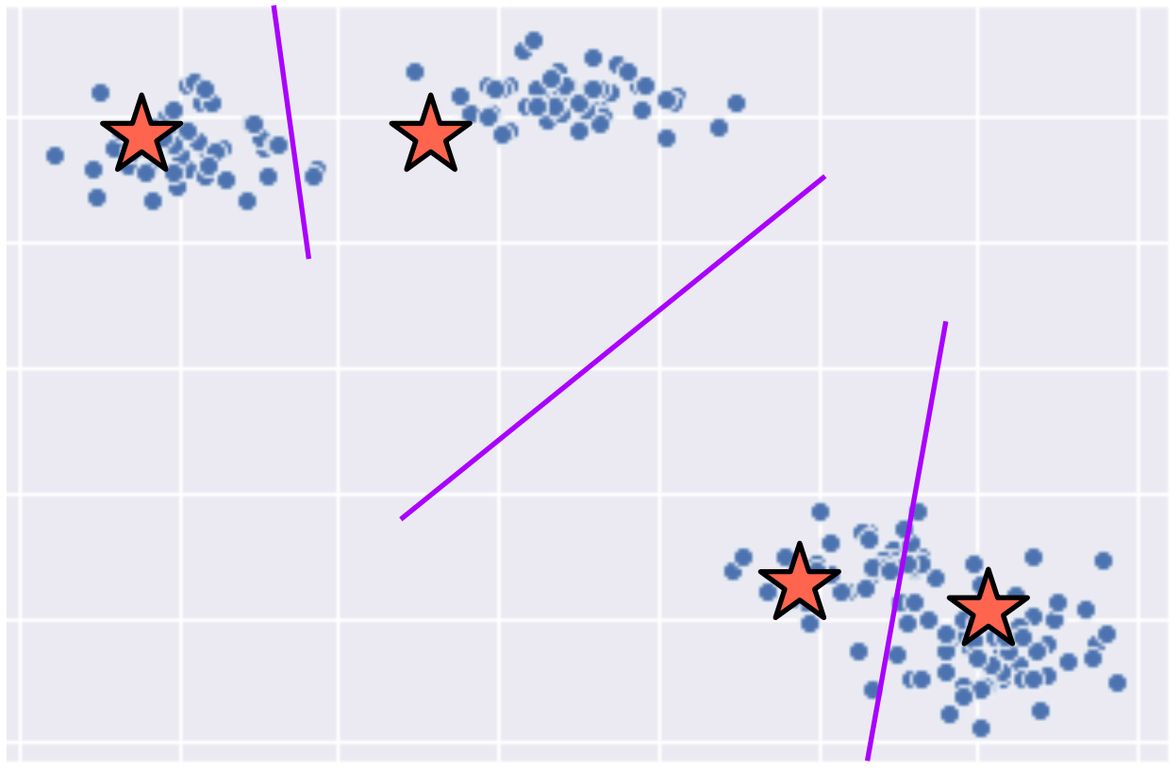
Recalculate centroids based on current cluster assignment



K-Means

Based on minimising within-cluster distances (to centroids)

Recalculate centroids based on current cluster assignment

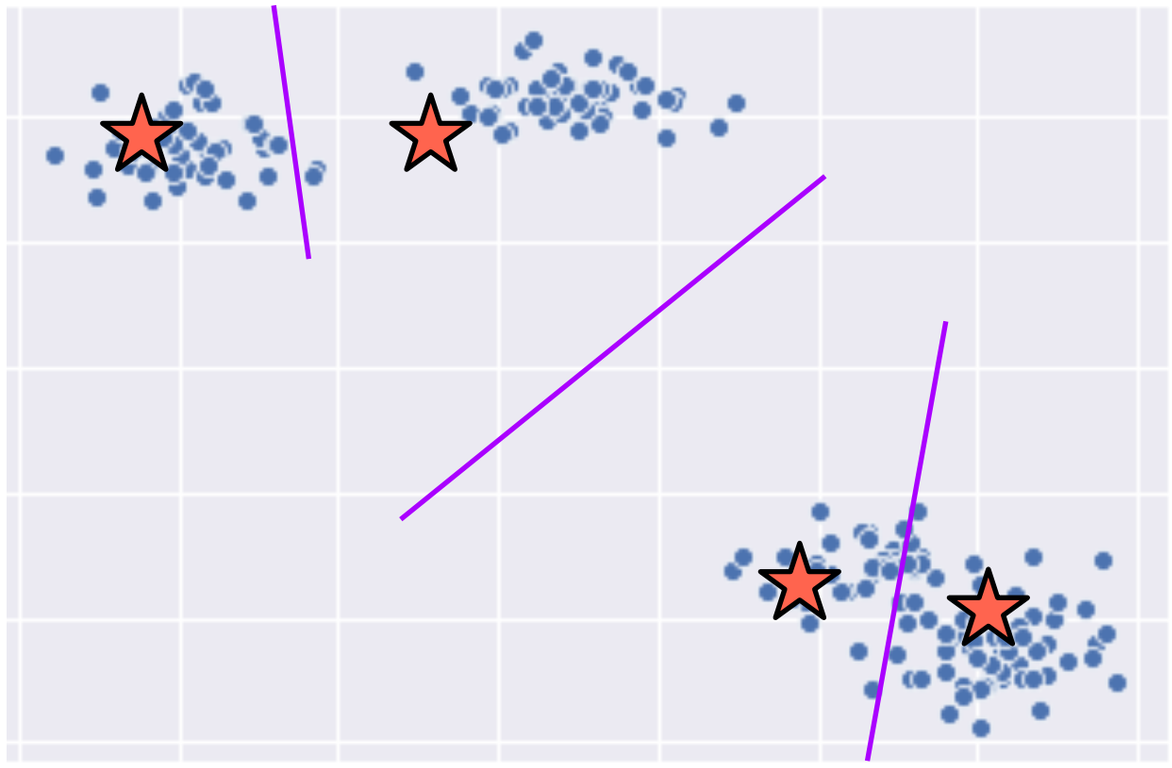


… and iterate …

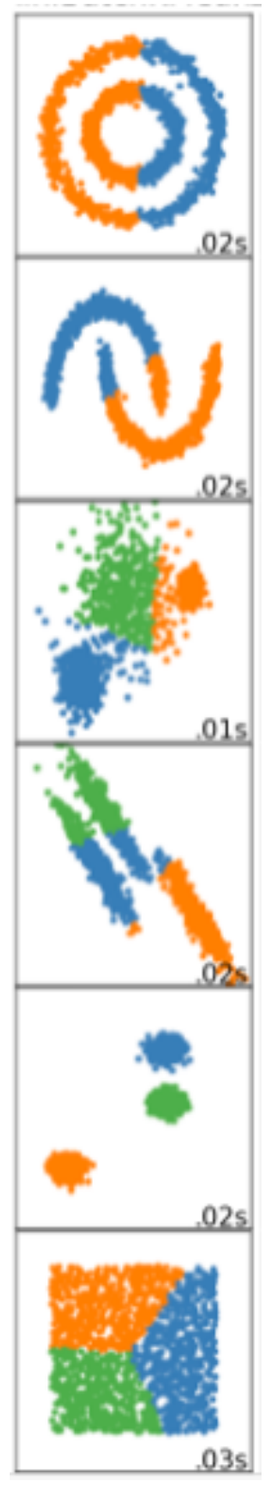
K-Means

Based on minimising within-cluster distances (to centroids)

Recalculate centroids based on current cluster assignment



… and iterate …



K-Means

Must choose:

N clusters

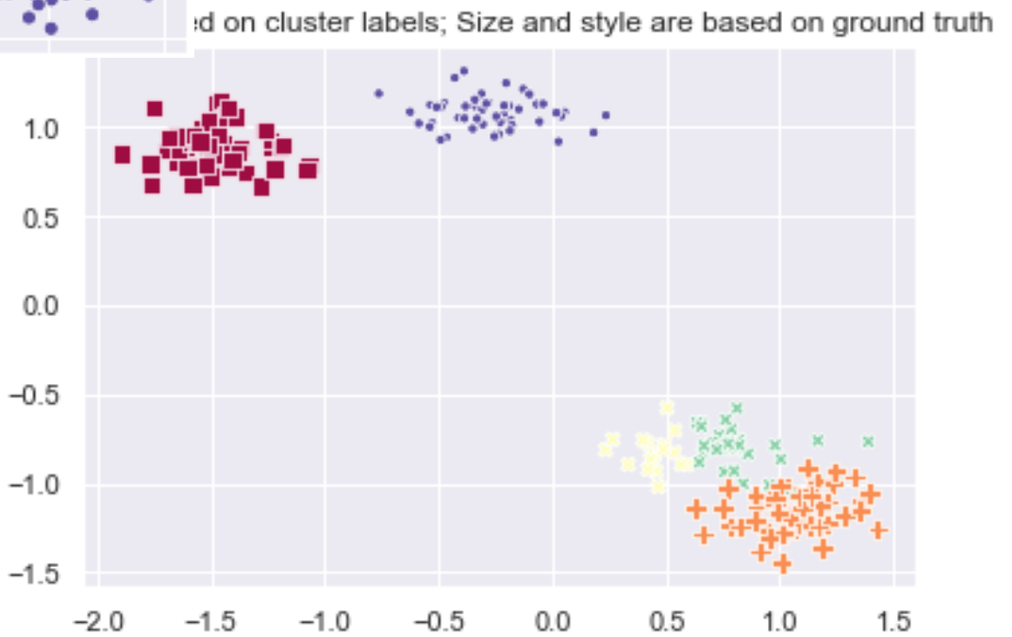
Initial centroids

2 Clusters

3 Clusters

4 Clusters

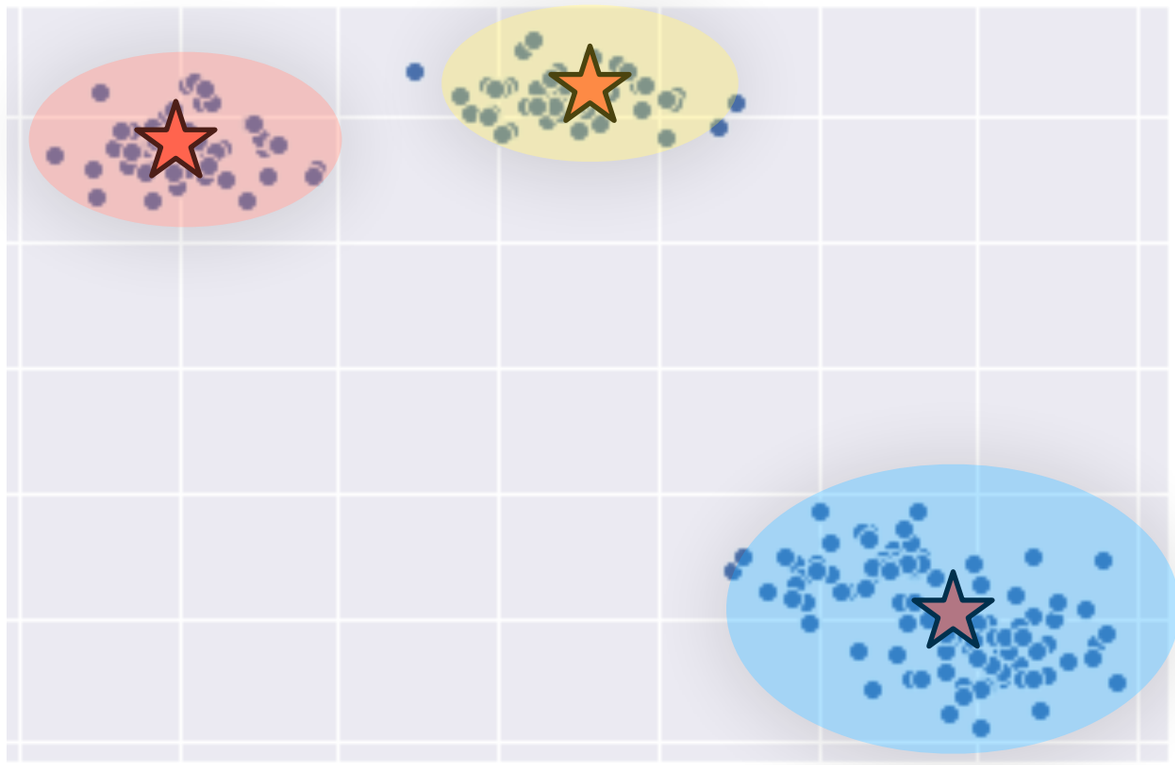
5 Clusters



Gaussian Mixture Model

Explicit probability density modelling

Parameterised by centres and covariances

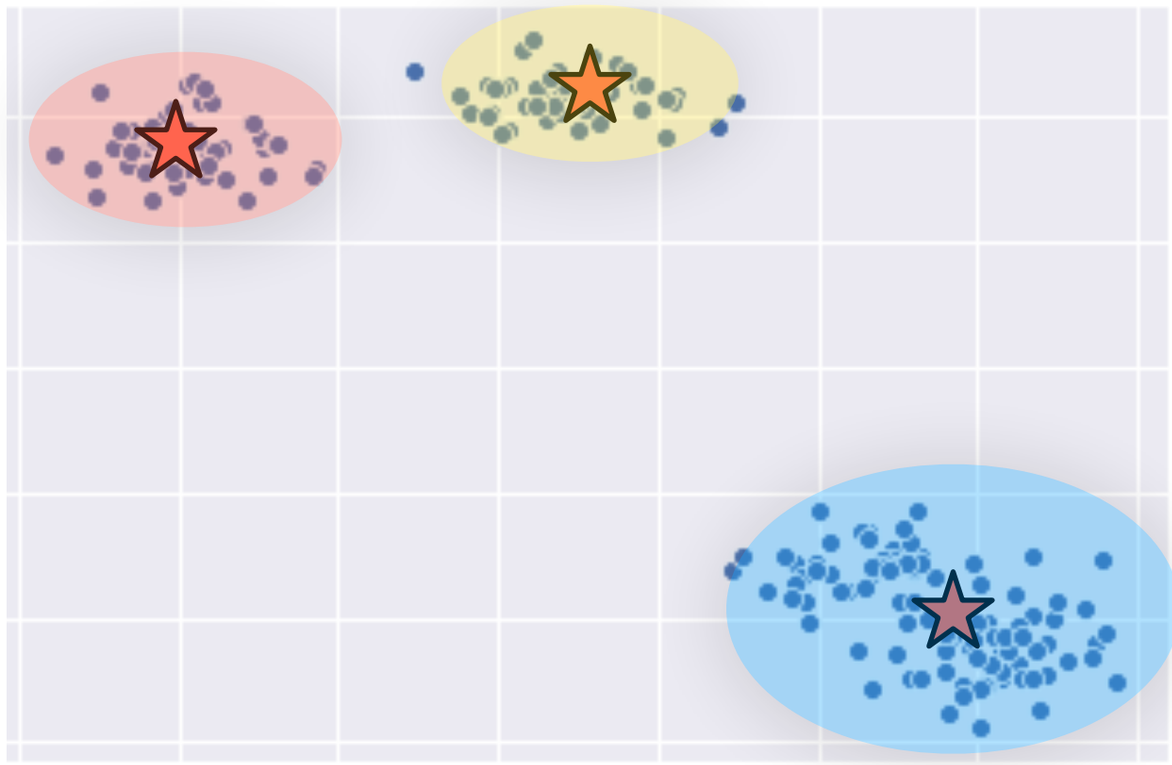


Gaussian Mixture Model

Explicit probability density modelling

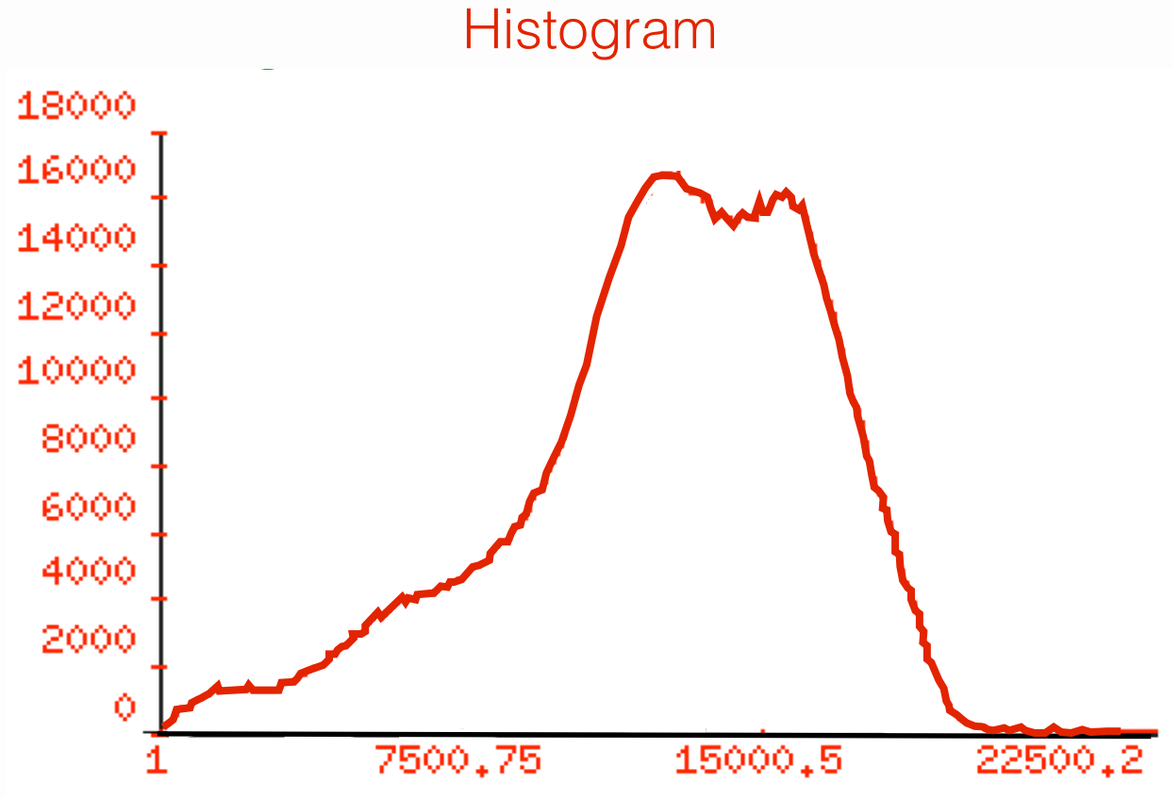
Parameterised by centres and covariances

… aka “blobs”



Also works in 1D …

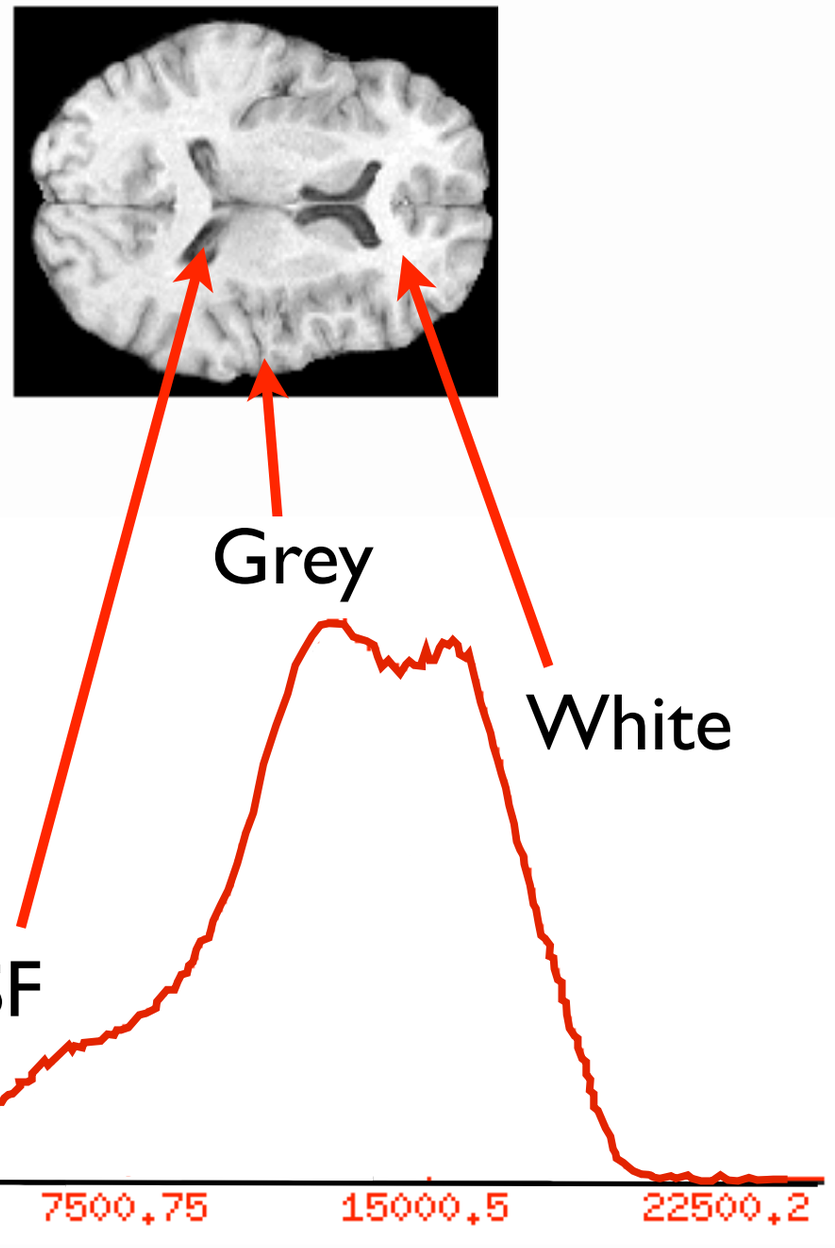
Gaussian Mixture Model



Also works in 1D …

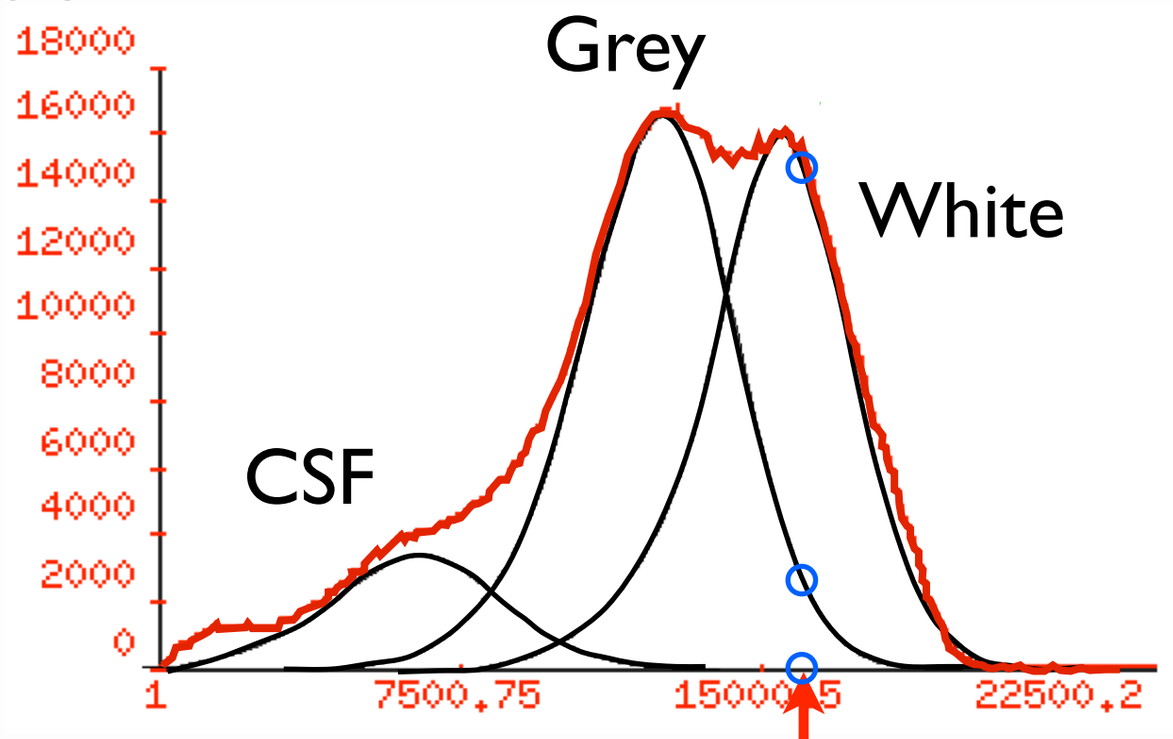
Gaussian Mixture Model
brain tissue segmentation

- Histogram = voxel count vs. intensity
- Model = mixture of Gaussians
- If well separated, have clear peaks; then segmentation easy
- Overlap worsened by:
  - Bias field
  - Blurring
  - Low resolution
  - Head motion
  - Noise



Gaussian Mixture Model

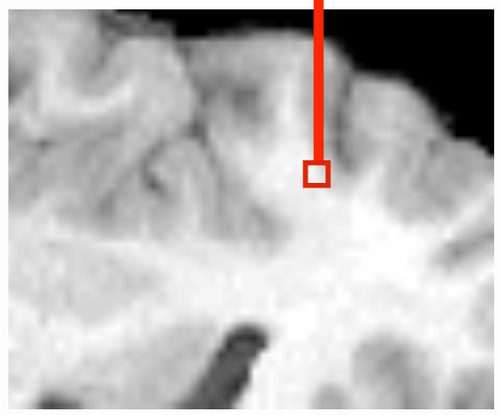
- Histogram = probability distribution function
- Model = mixture of Gaussians
- Probability determined for each tissue class



For example:

Voxel near WM/GM border

Intensity = 17203

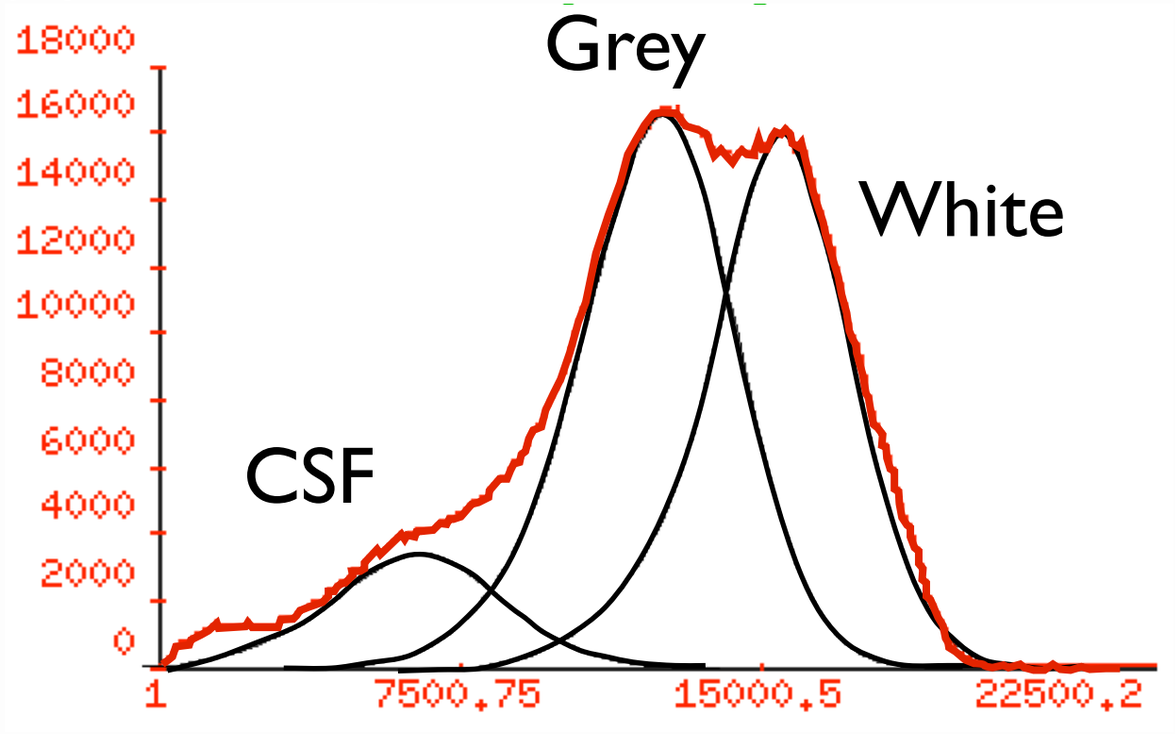


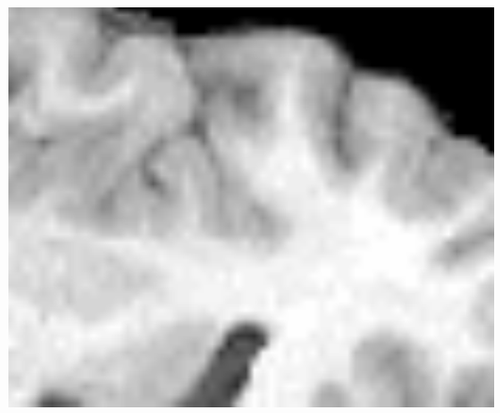
P(CSF) near zero
P(GM) low
P(WM) moderate
} $p(c \mid I, \theta)$

Gaussian Mixture Model

- Histogram = probability distribution function
- Model = mixture of Gaussians
- Probability determined for each tissue class

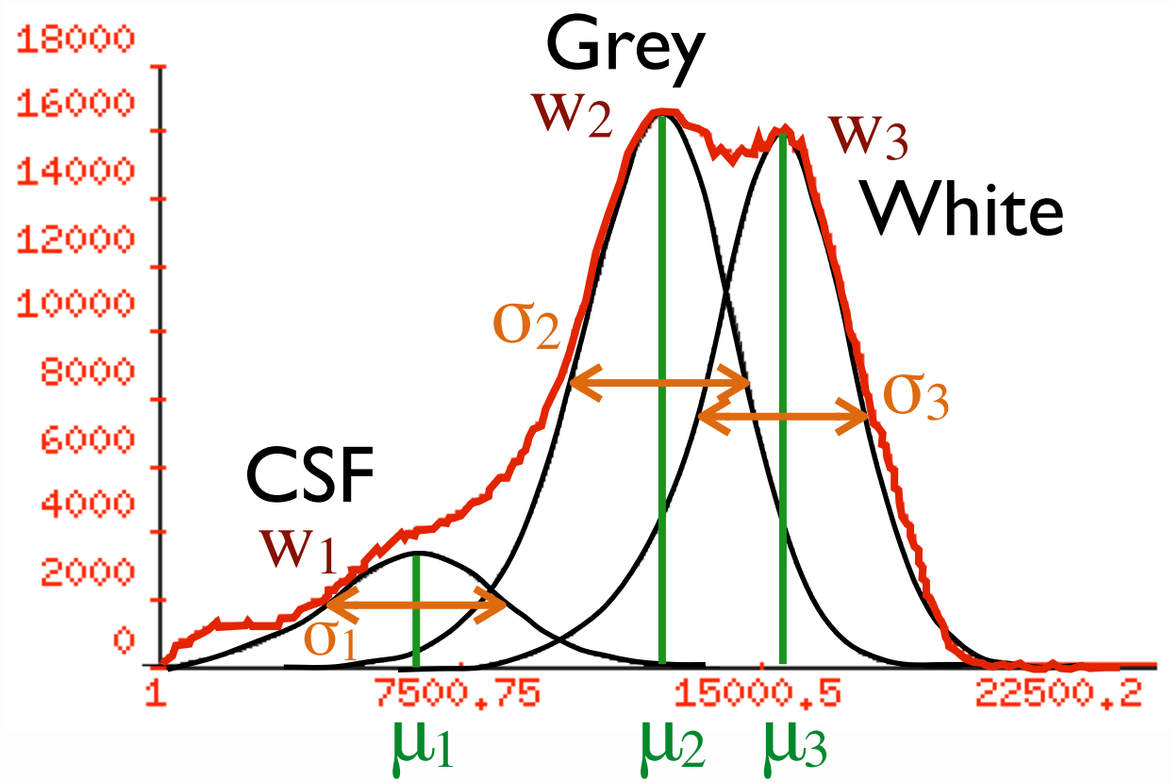
$$p(I) = \Sigma_c \, p(I \mid c) \; . \; p(c)$$





Gaussian Mixture Model

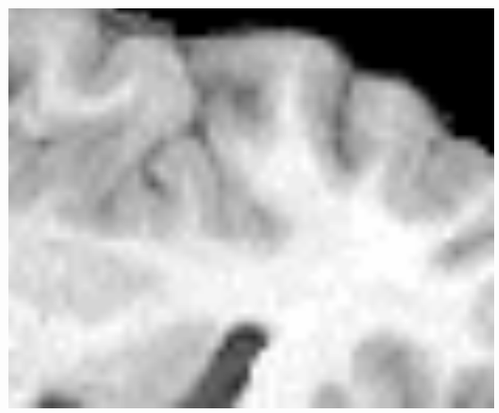
- Histogram = probability distribution function
- Model = mixture of Gaussians
- Probability determined for each tissue class



$$p(I) = \Sigma_c \, p(I \mid c) \; . \; p(c)$$
$$= \Sigma_j \, g(I \mid \mu_j, \sigma_j) \; . \; w_j$$

$$\theta = \{ \mu_j, \sigma_j, w_j \}$$
$$\Sigma_j \, w_j = 1$$

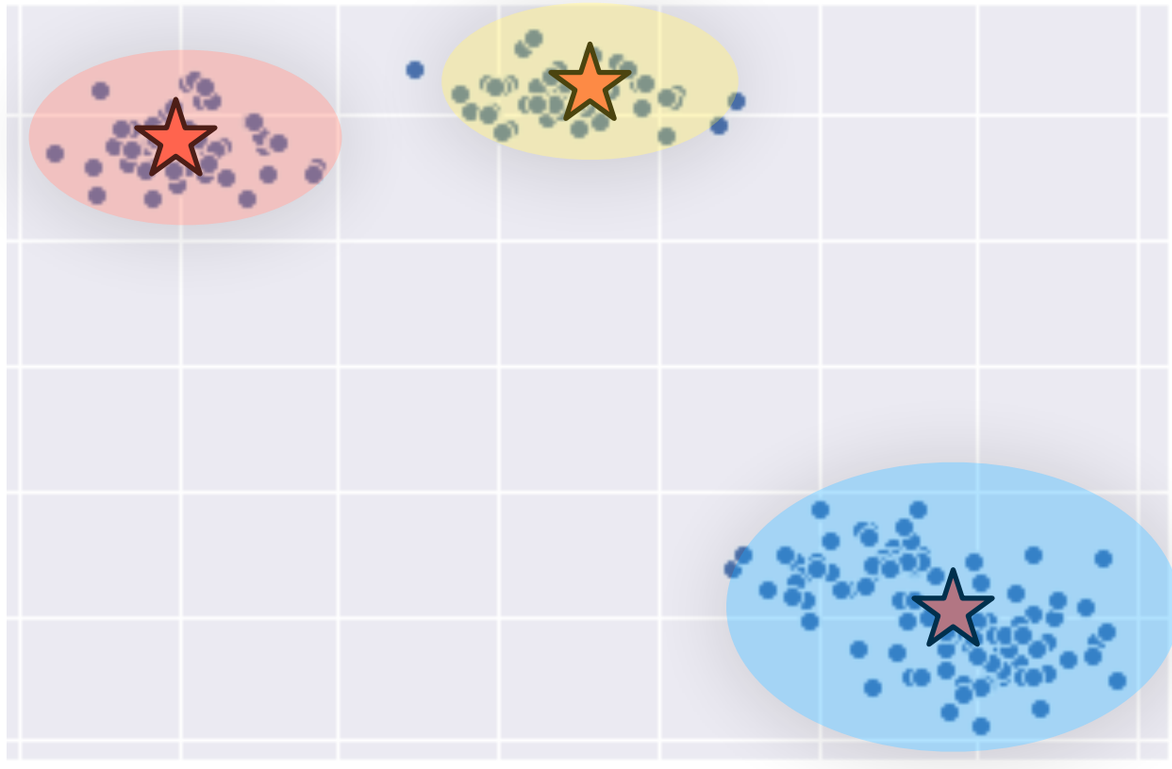
$$p(c=j \mid I, \theta) = g(I \mid \mu_j, \sigma_j) \; . \; w_j \; / \; p(I)$$

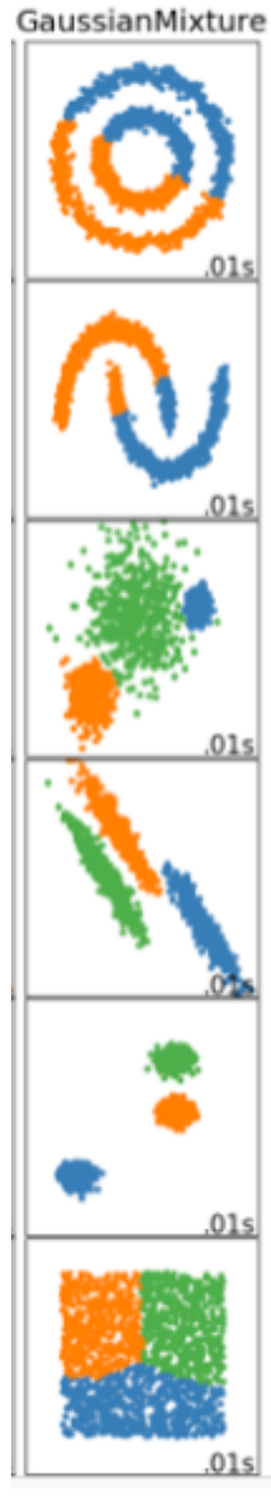


Gaussian Mixture Model

Explicit probability density modelling

Need to ***initialise*** means and covariances

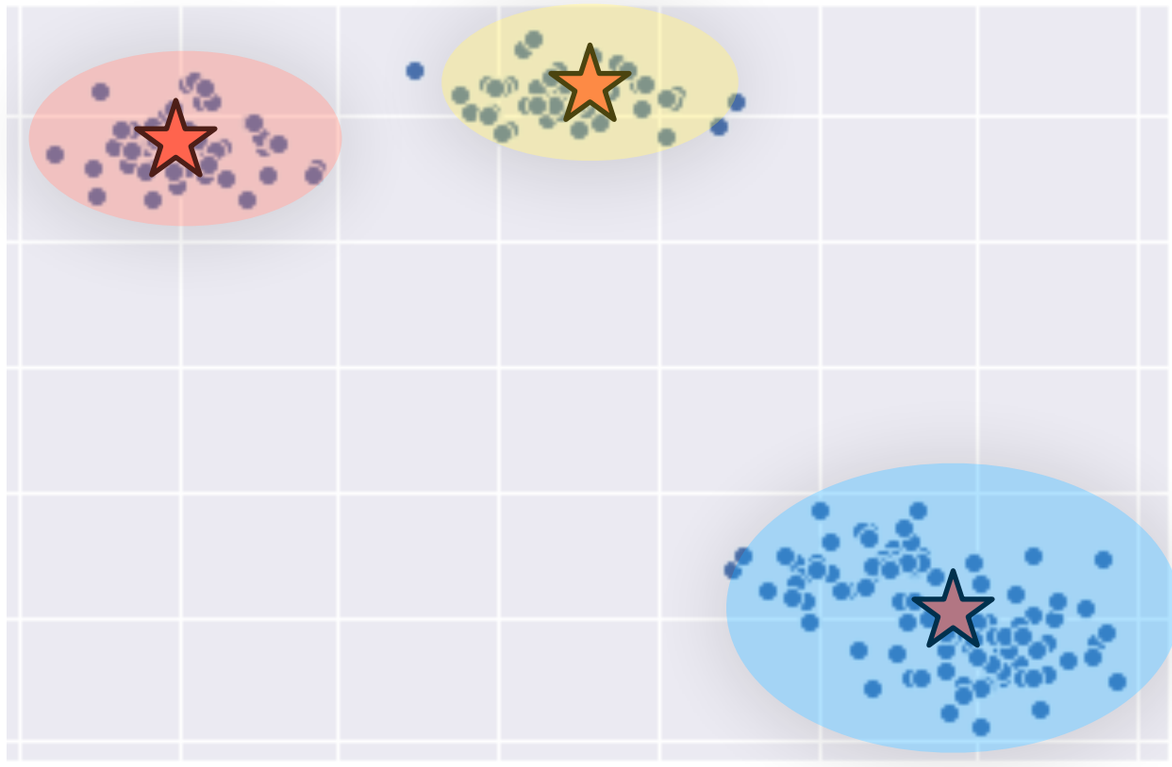




Gaussian Mixture Model

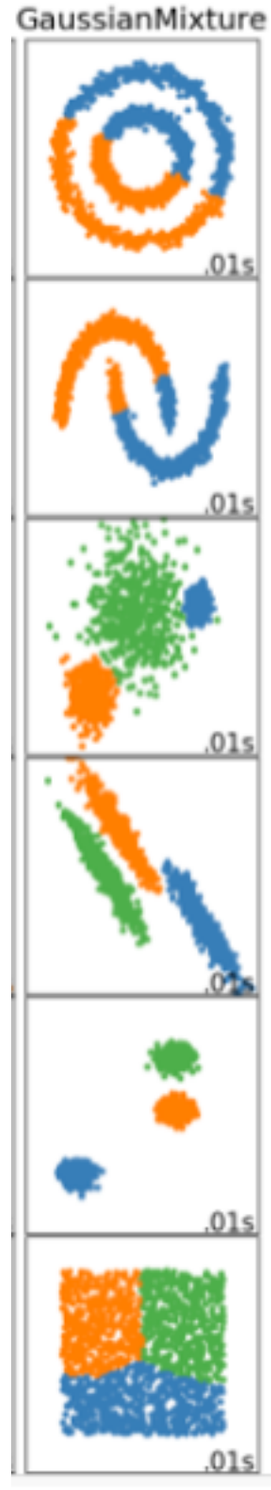
Explicit probability density modelling

Need to ***initialise*** means and covariances

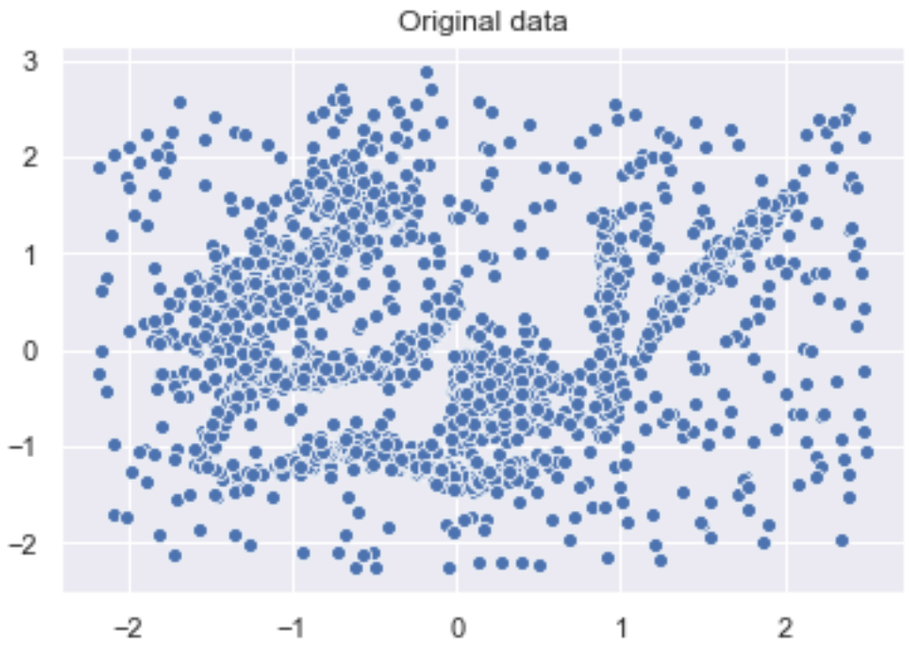


A lot of the complexity is in the details

You don’t need to know all the details to use it



Other Examples

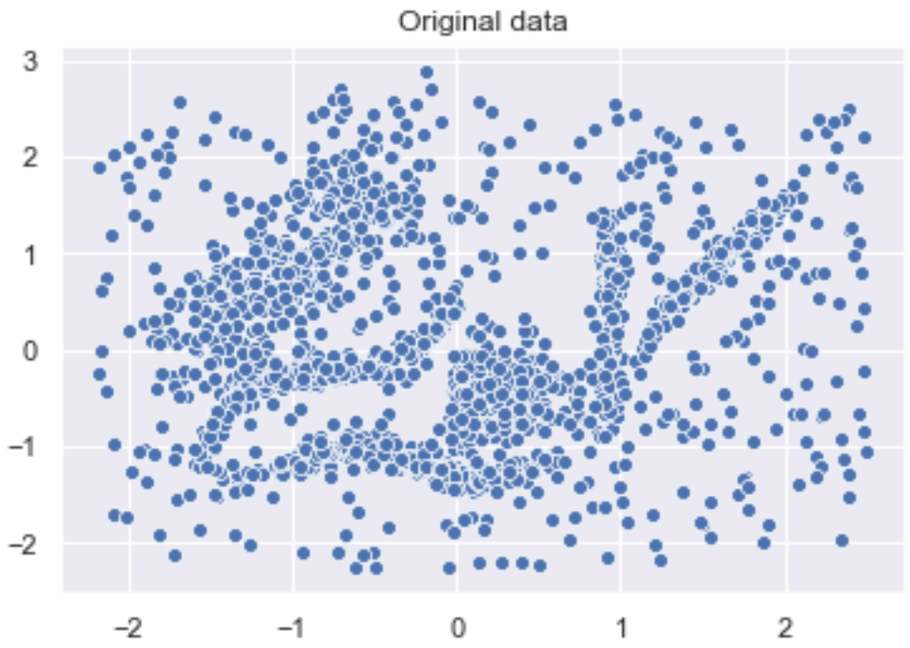


In some cases it is hard to decide what is “right”

Some methods may use density or build up local groupings into global clusters

Some methods separate out “anomalies” from core data points

Other Examples



In some cases it is hard to decide what is “right”

Some methods may use density or build up local groupings into global clusters

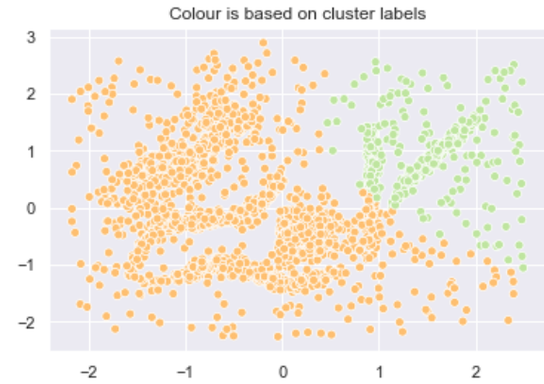
Some methods separate out “anomalies” from core data points

KMeans N=2

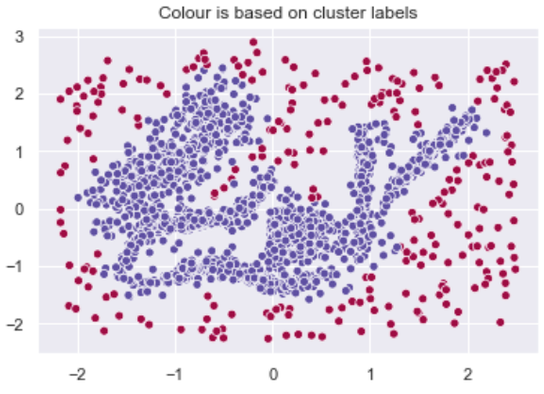
GMM N=2

Ward N=2

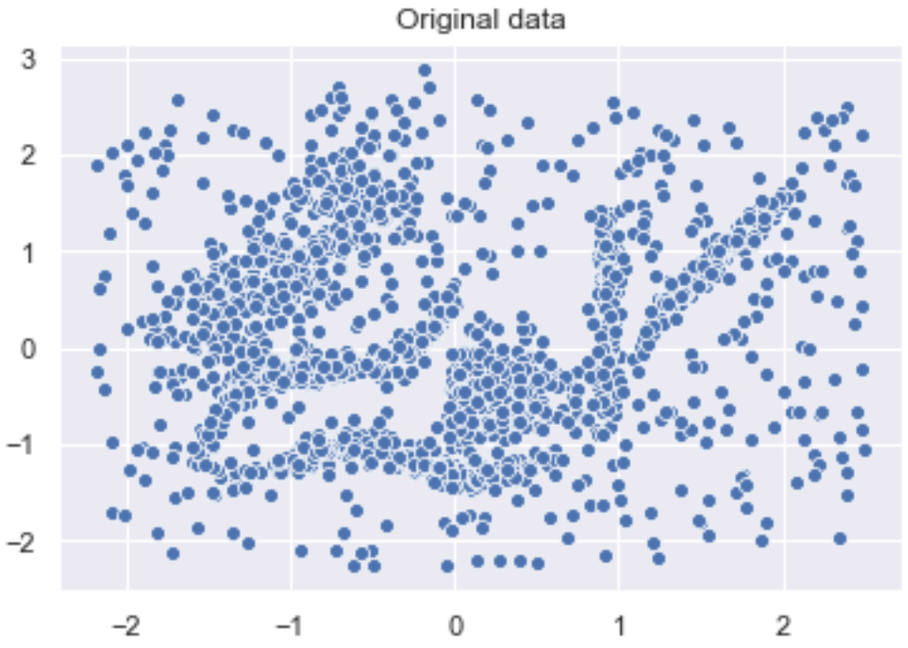
Spectral N=2



DBSCAN (esp=0.5)



Other Examples

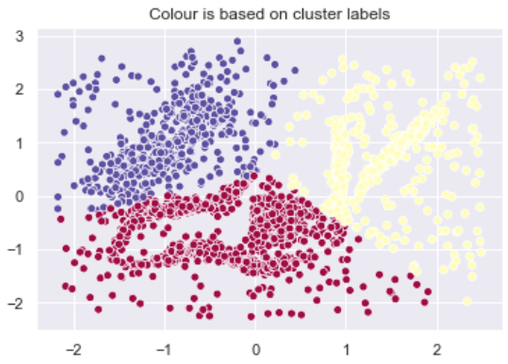


In some cases it is hard to decide what is “right”

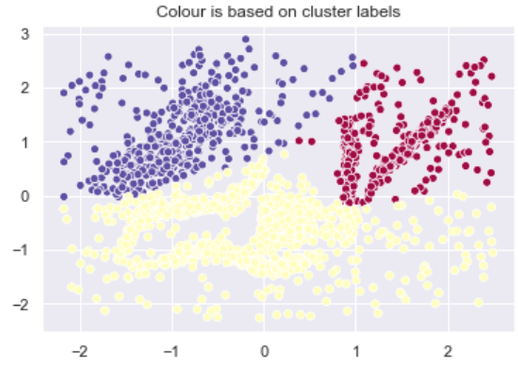
Some methods may use density or build up local groupings into global clusters

Some methods separate out “anomalies” from core data points

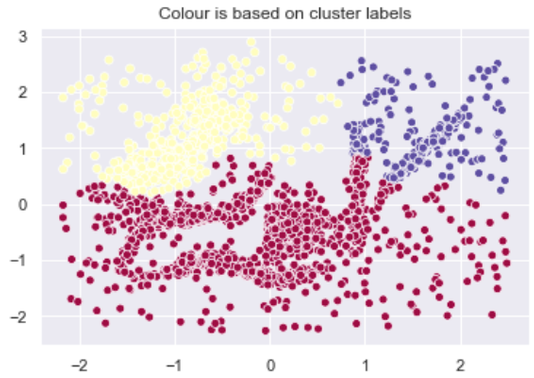
KMeans N=3



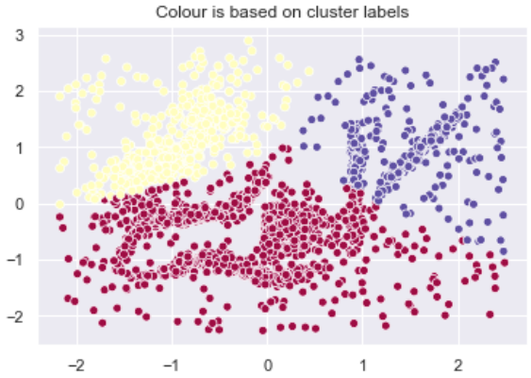
GMM N=3



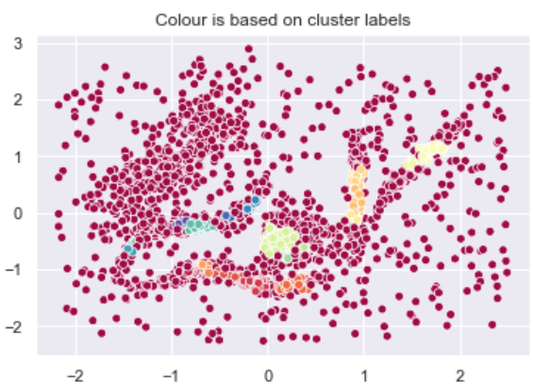
Ward N=3



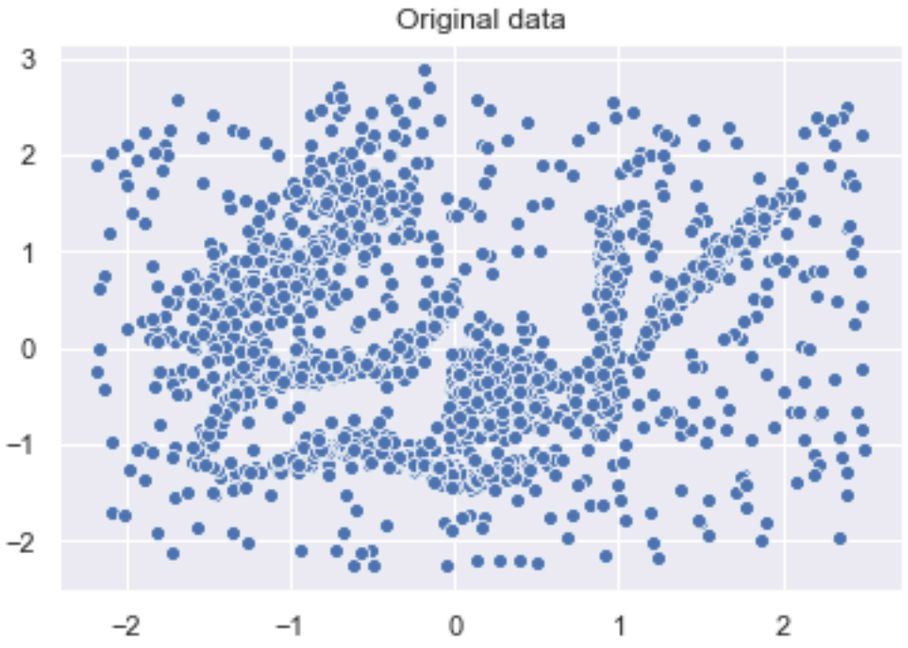
Spectral N=3



DBSCAN (esp=0.05)

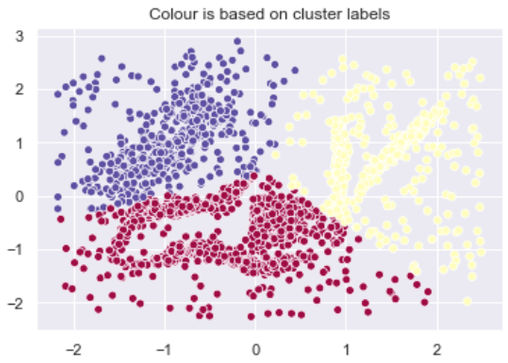


Other Examples

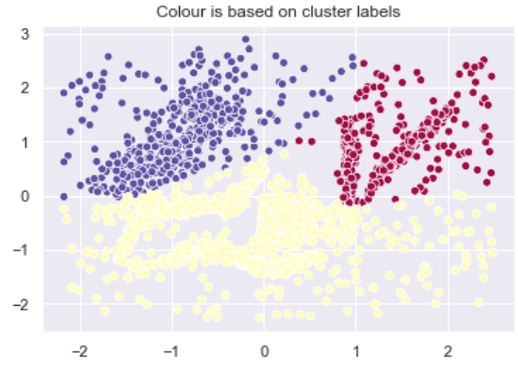


Is there a mathematical formula for what is good?

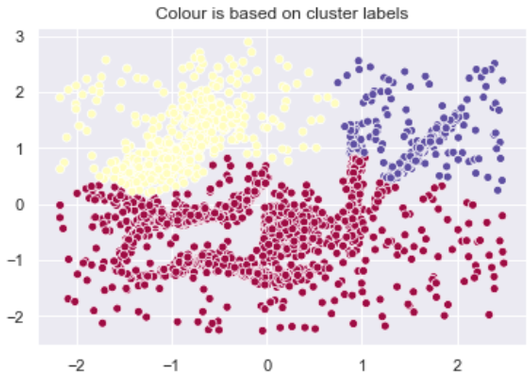
KMeans N=3



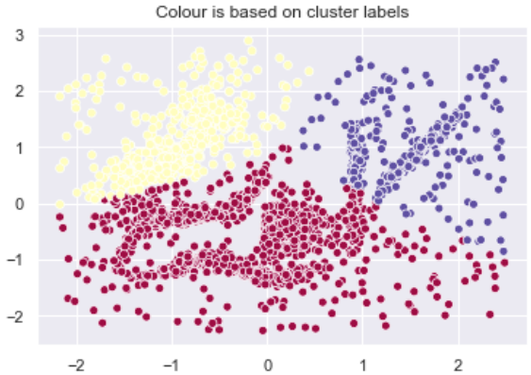
GMM N=3



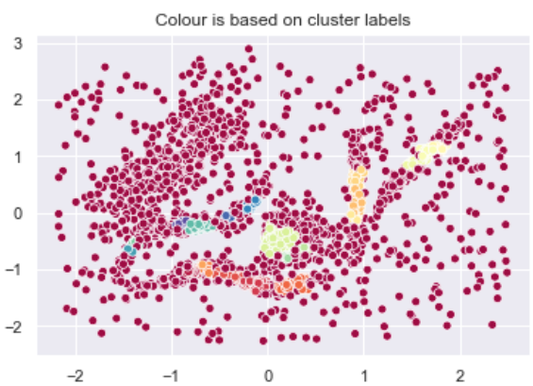
Ward N=3



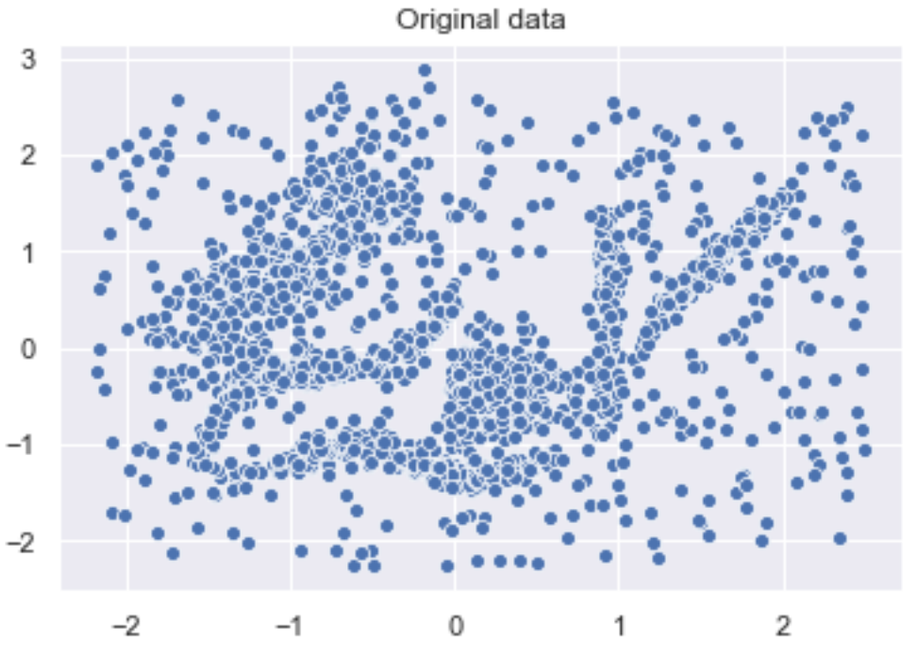
Spectral N=3



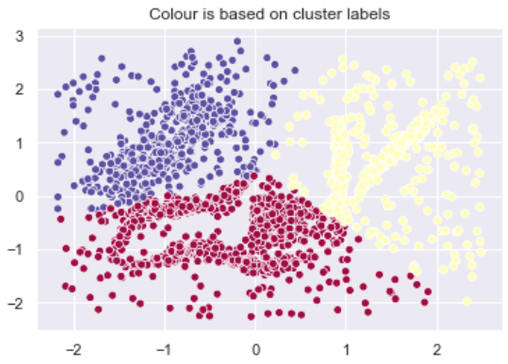
DBSCAN (esp=0.05)



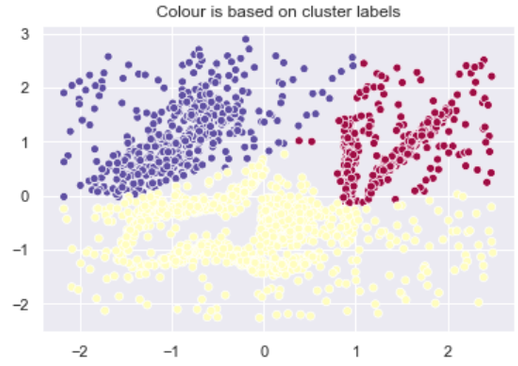
Other Examples



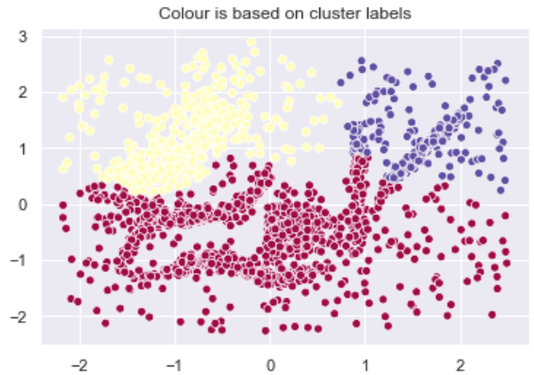
KMeans N=3



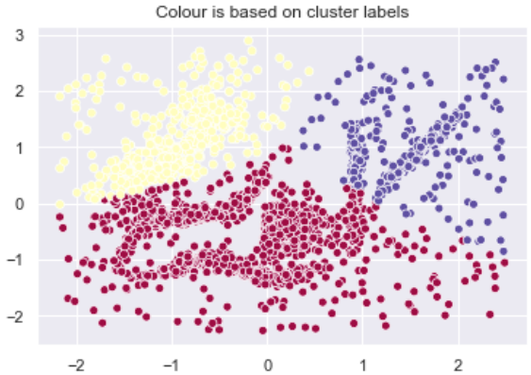
GMM N=3



Ward N=3



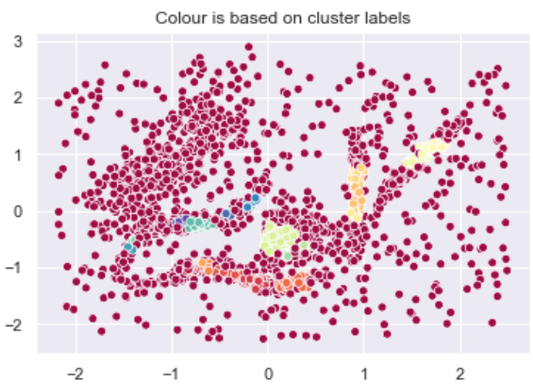
Spectral N=3



Is there a mathematical formula for what is good?

Can you even get people to agree on what is good?

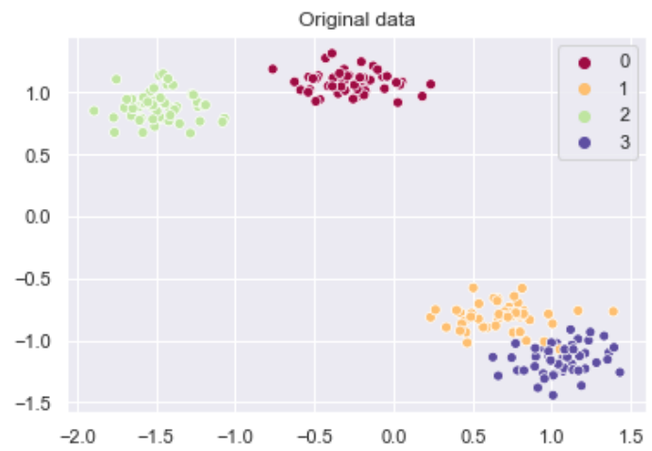
DBSCAN (esp=0.05)

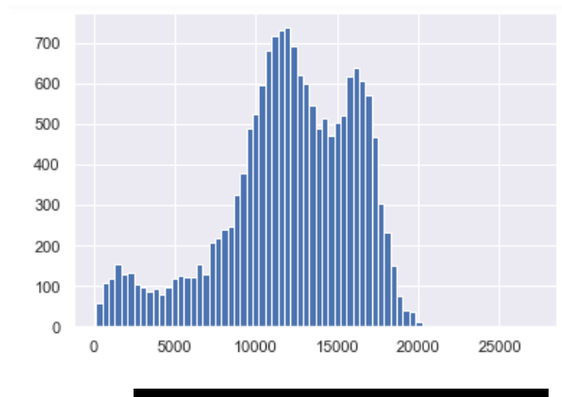


Other Examples

In the practical we will explore a range of methods

… on a range of datasets



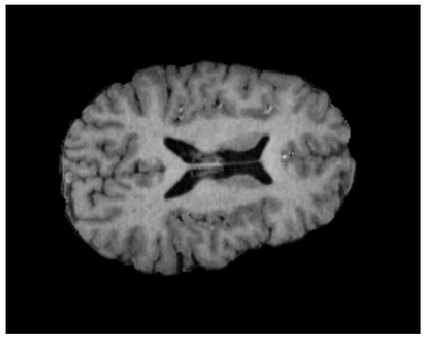


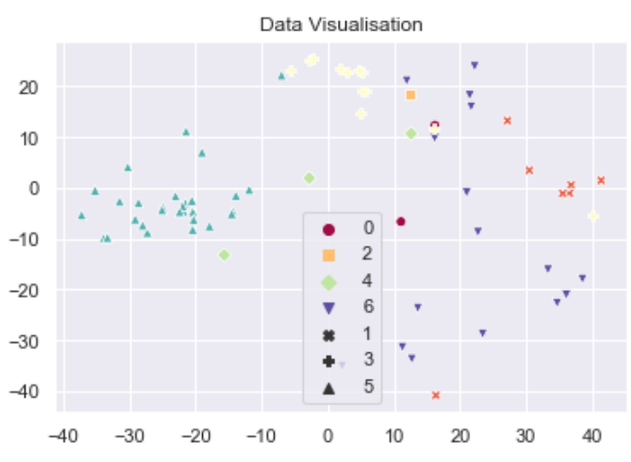
Real and not

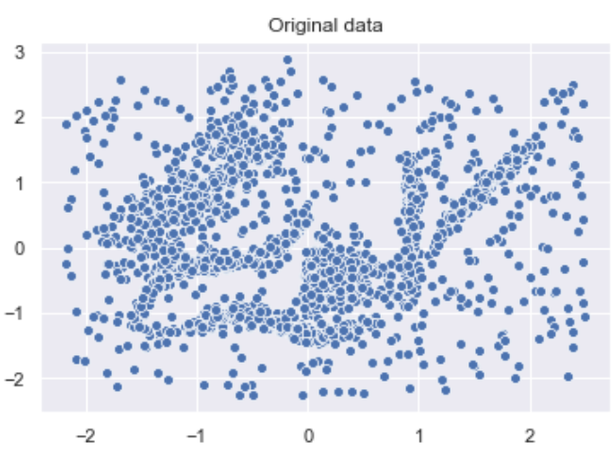
Different dimensionalities

Different numbers of true clusters

Different numbers of datapoints and density







Overview

Clustering Methods

Hierarchical Clustering

Data Visualisation

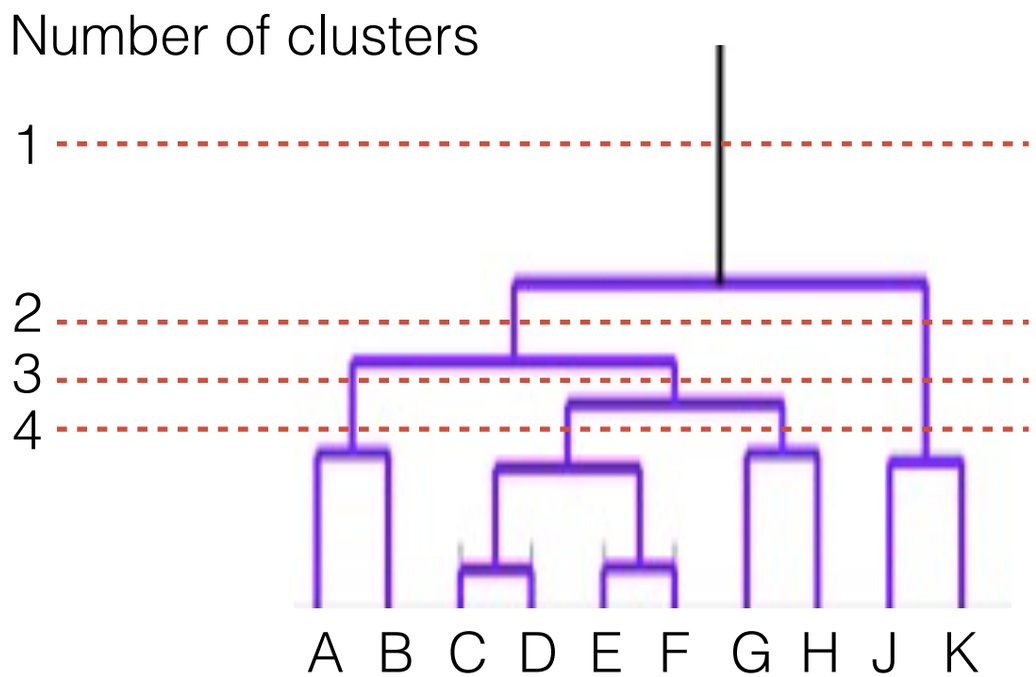
Dimensionality Reduction

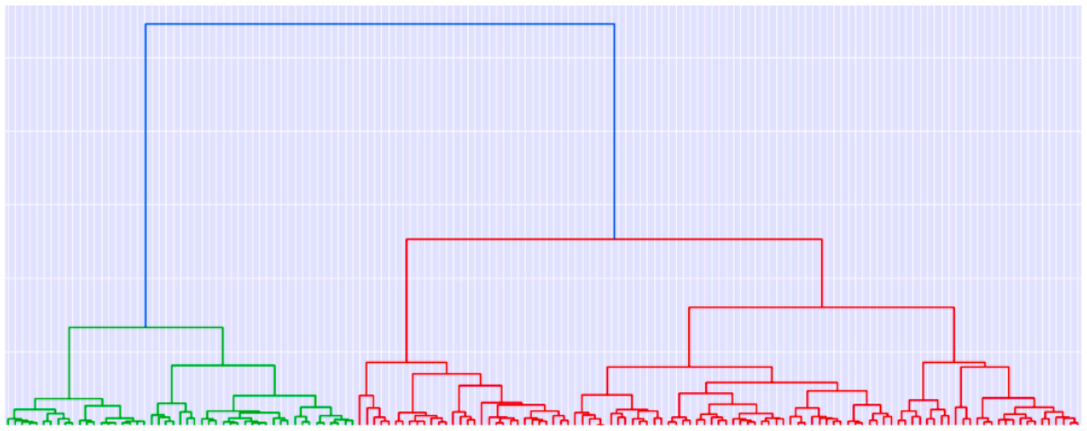
Hierarchical Clustering

- Bottom-up (agglomerative) or top-down (divisive) approaches
- Agglomerative is more commonly used
- Idea is simple:
  - Start with all datapoints separate (each is a “set” of 1 point)
  - Sequentially merge “sets” based on pairwise point distances and linkage criteria (measuring cost/benefit of merging sets)
  - Visualise the series of merges as a dendrogram
    - generally only useful for limited dataset sizes

Dendrograms

- Original datapoints shown at bottom
- Drawing a horizontal line shows number of clusters at that point
- Can look at different options for number of clusters easily
- Height of connection represents distance/separation





Example

- C & D very close, like E & F
- 4 cluster version is:
  - A + B
  - C + D + E + F
  - G + H
  - J + K
- Cluster J + K is most isolated

Example: Brain Networks

Medial Visual

Lateral Visual

Auditory

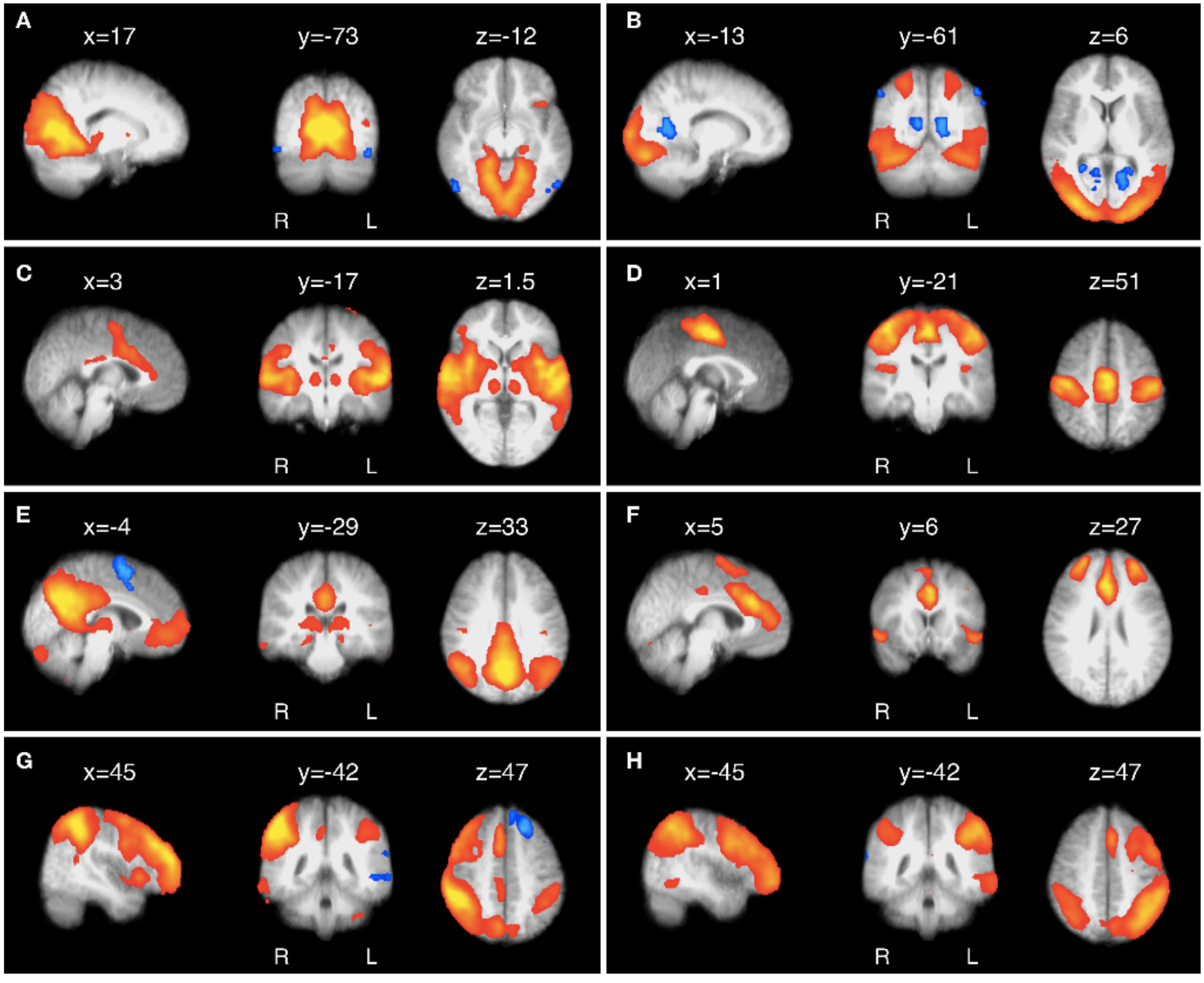
Sensory-Motor

Visuo-Spatial

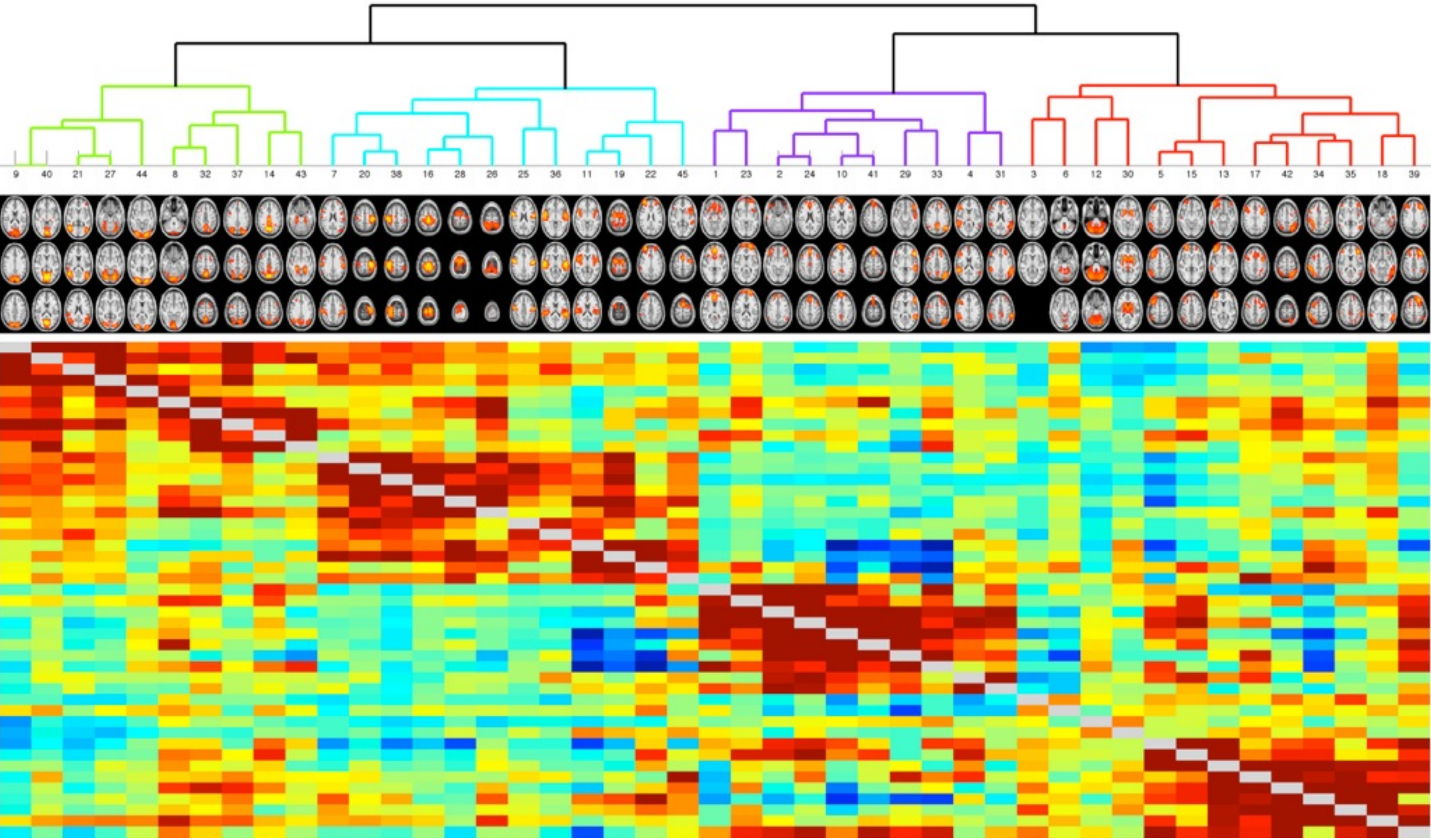
Executive Control

Dorsal Visual Stream

Dorsal Visual Stream



Hierarchy of clusters



Overview

Clustering Methods

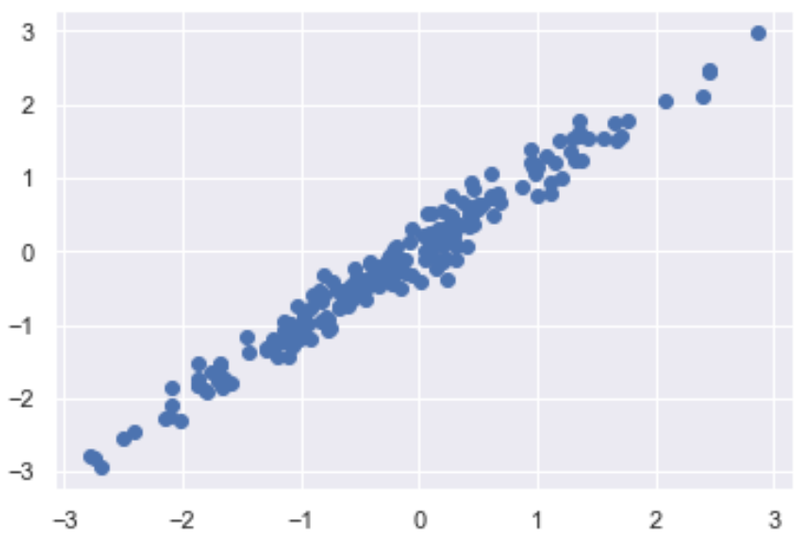
Hierarchical Clustering

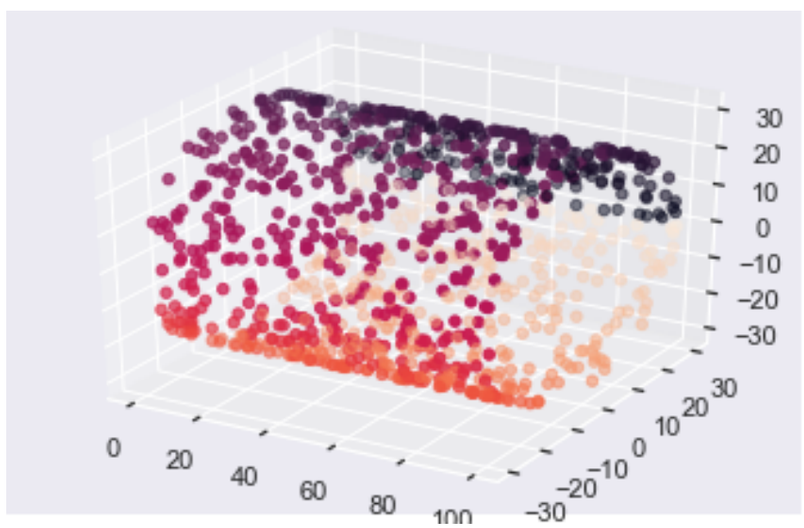
Data Visualisation

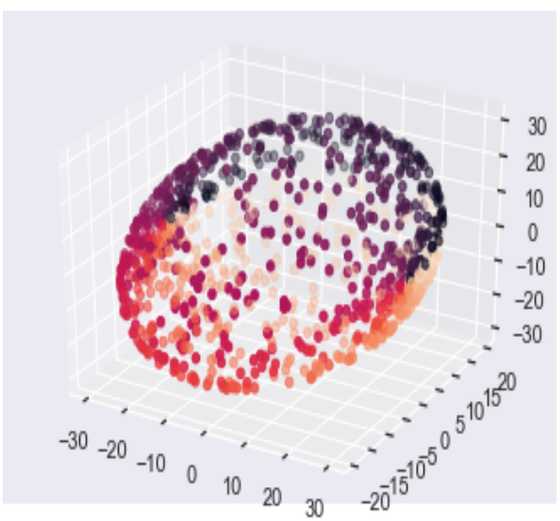
Dimensionality Reduction

Data Visualisation

- High dimensional data cannot simply be inspected
- Visualising data is a key skill!
- Even data with more than 4 dimensions can be tricky to get some intuition for
- Many methods exist for data visualisation
  - Most are based on finding local subspaces of interest
  - PCA finds linear subspaces (hyperplanes)
  - Manifold learning finds a non-linear embedded space (like the 2D surface of a sphere or cylinder in 3D)

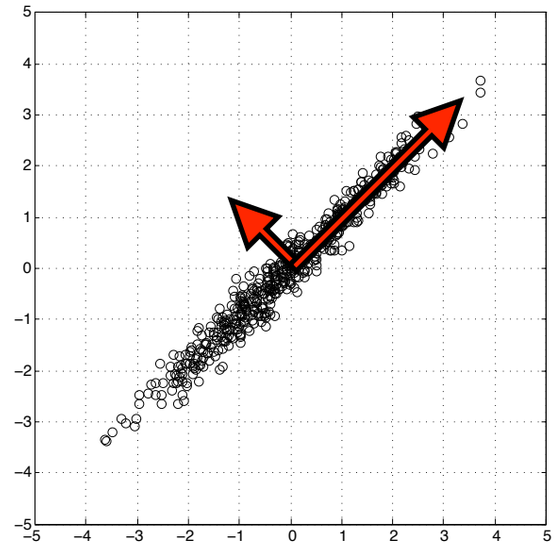




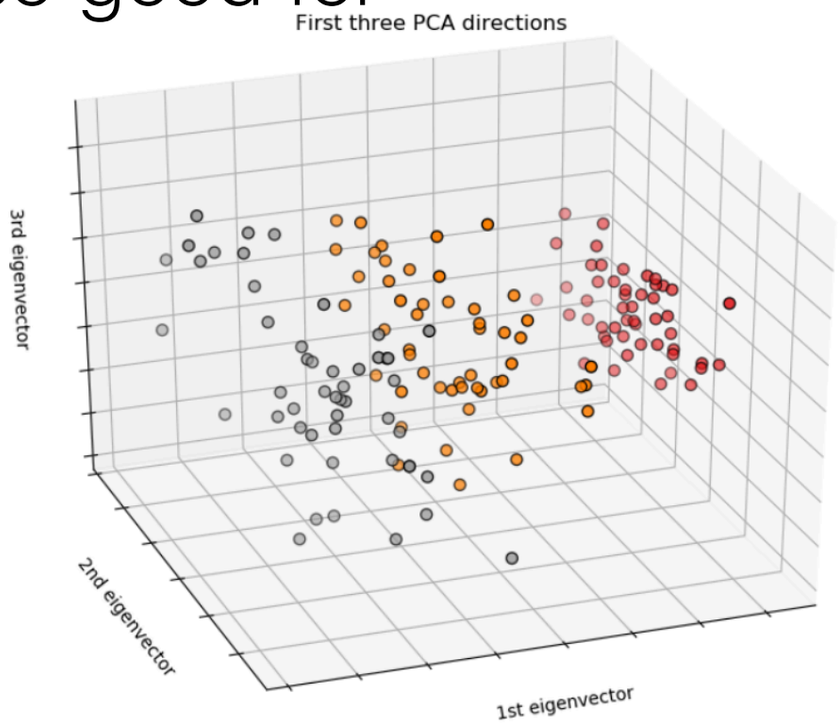


Data Visualisation: PCA

- Principal Component Analysis
- Find directions in the data of maximum variance
- Can find as many directions as are in the data
- Each direction is orthogonal to all other directions



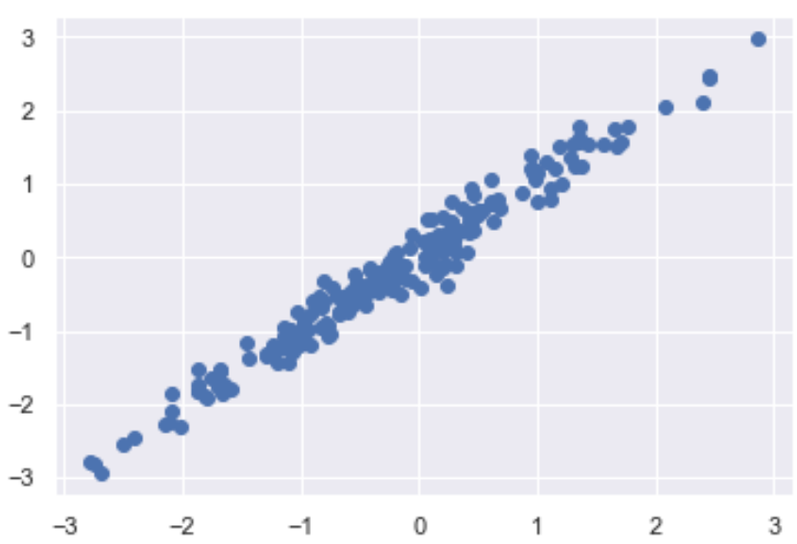
- These allow directions/dimensions that only capture small changes to be ignored for visualisation (also good for dimensionality reduction)
- Based on principles of eigen-vectors / singular value decomposition (SVD) of covariance matrix

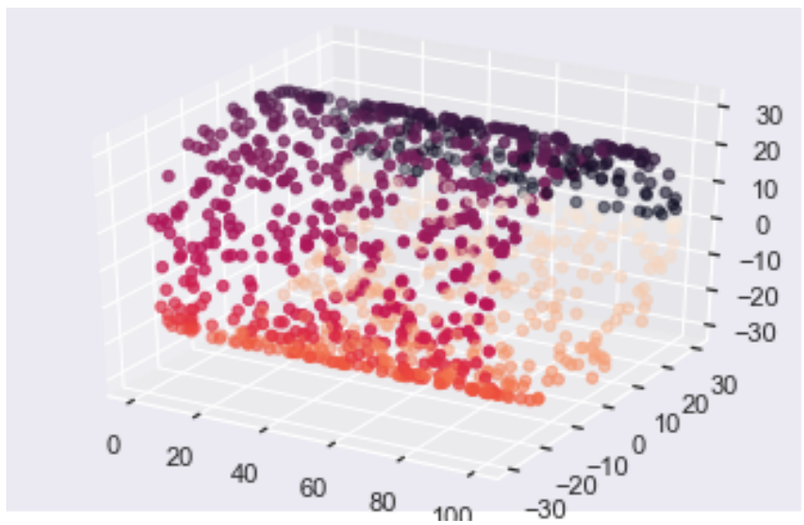


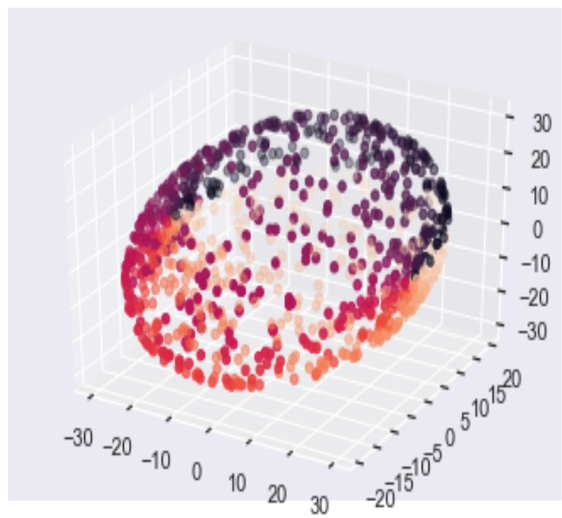
https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html

Data Visualisation: Manifold Embedding

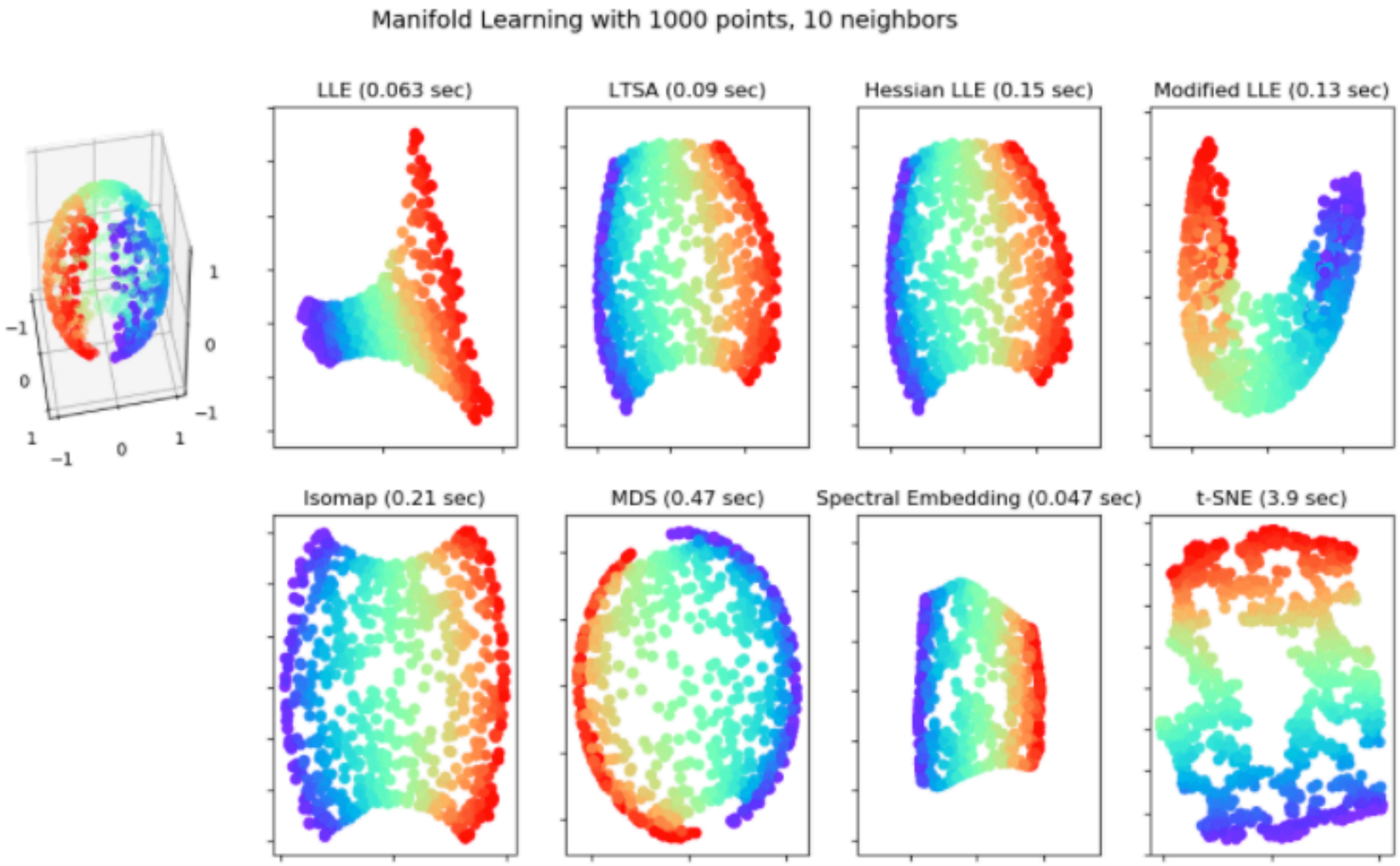
- Need to map down to 2D or 3D space
- Uses local parts of space (hyperplanes) that explain variation
- Overall mapping is nonlinear (e.g. sphere)
- Distances cannot be exactly maintained
  - projection / lossy mapping
  - often tries to maintain as much local and global similarity
  - e.g. local linear embedding







Data Visualisation



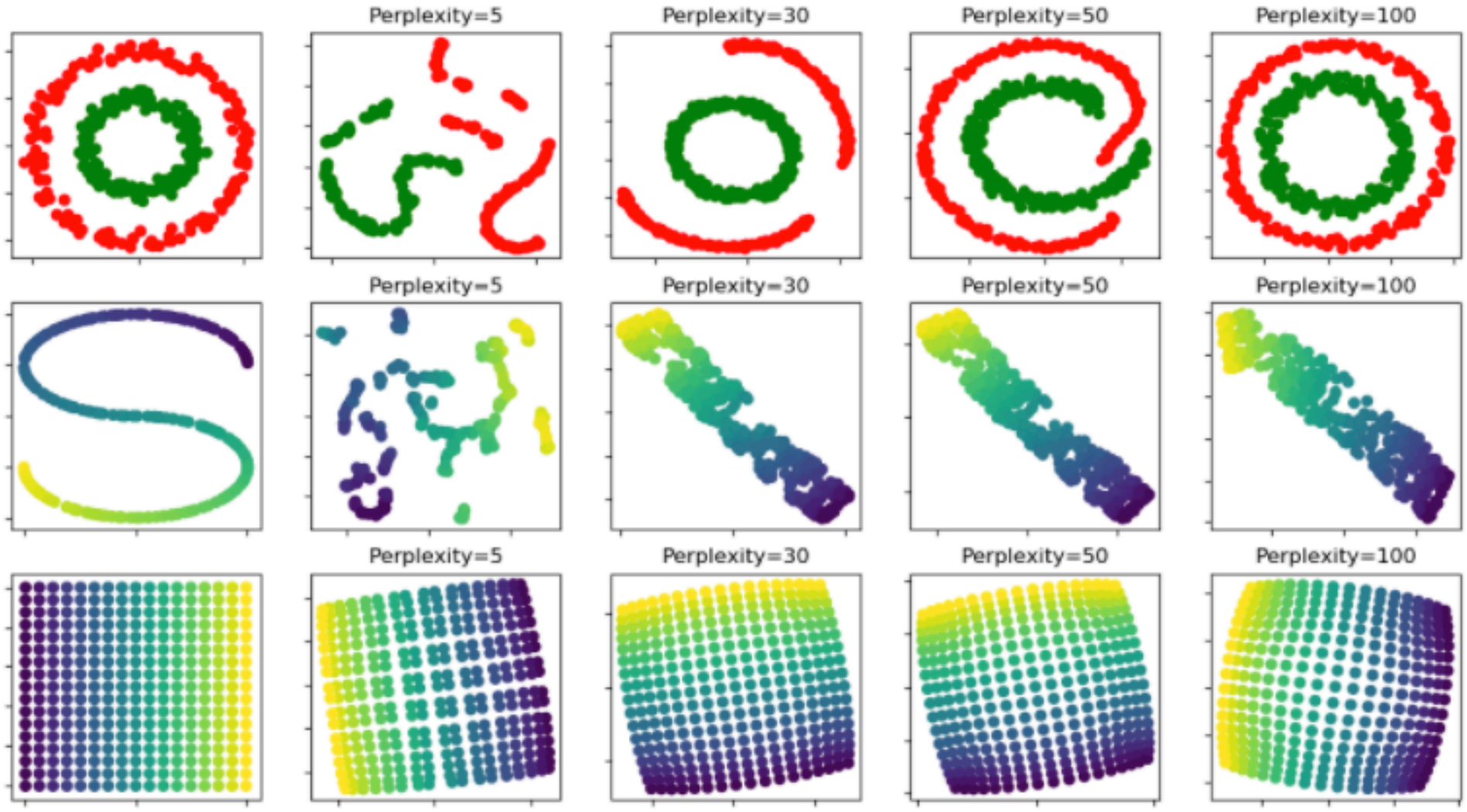
https://scikit-learn.org/stable/auto_examples/manifold/plot_manifold_sphere.html#sphx-glr-auto-examples-manifold-plot-manifold-sphere-py

Data Visualisation: tSNE

t-distributed Stochastic Neighbour Embedding

- Maps the values to the new space using stochastically estimated local nonlinearities
- Stochastic algorithm (different each time)
- Distances are not maintained
  - small clusters can appear large and vice versa
  - nearby clusters might be shown far apart and vice versa
- Perplexity parameter can radically change the look
- Not ideal for use with very high dimensional data

Data Visualisation



https://scikit-learn.org/stable/auto_examples/manifold/plot_t_sne_perplexity.html#sphx-glr-auto-examples-manifold-plot-t-sne-perplexity-py

Overview

Clustering Methods

Hierarchical Clustering

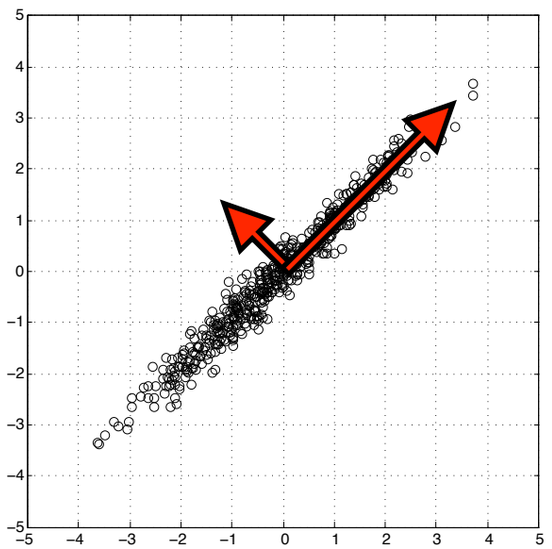
Data Visualisation

Dimensionality Reduction

Dimensionality Reduction

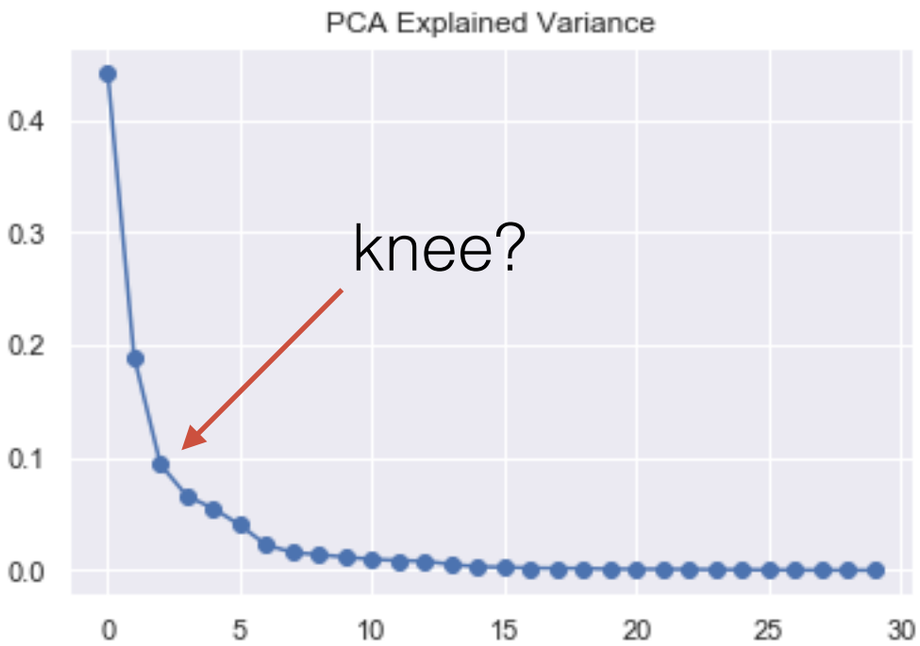
- Methods discussed for data visualisation are often useful for dimensionality reduction too
- Usually want to keep more than 2 or 3 dimensions
- General principle is to project/deform to lower dimension
- Lose some information, but minimise curse of dimensionality

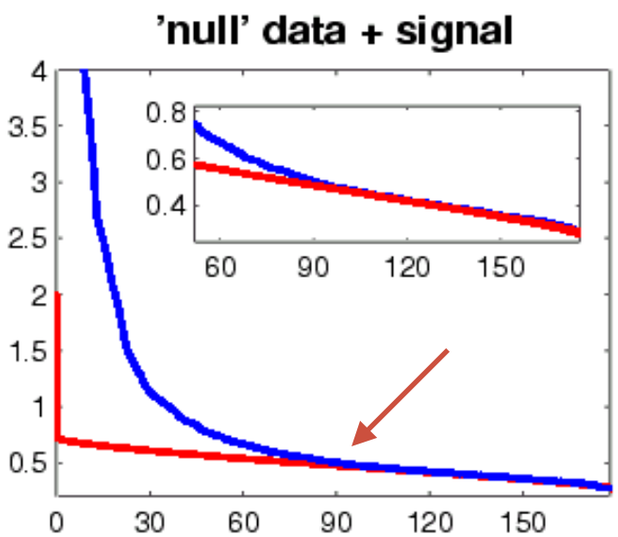
- How many dimensions to keep? That is the key question!
- PCA is one of the most commonly used methods
  - Scales extremely well for very high dimensions
  - Easier to understand (well studied)
  - Maintains linear data structure
    - not always good - e.g. manifold embedding



Dimensionality Reduction: PCA

- Only keep N components
- Can calculate how much variance each component explains
- These correspond to eigenvalues or singular values
- Expect small ones to only capture noise
- Can try to decide on how many components based on:
  - “knee”
  - % cumulative variance
  - statistical model

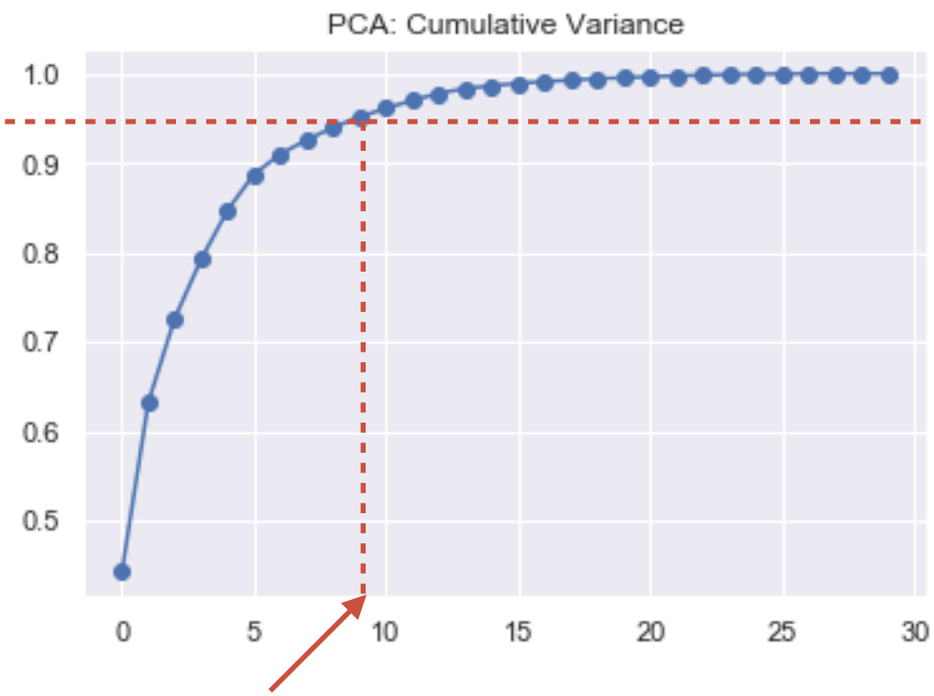




Expected from noise alone

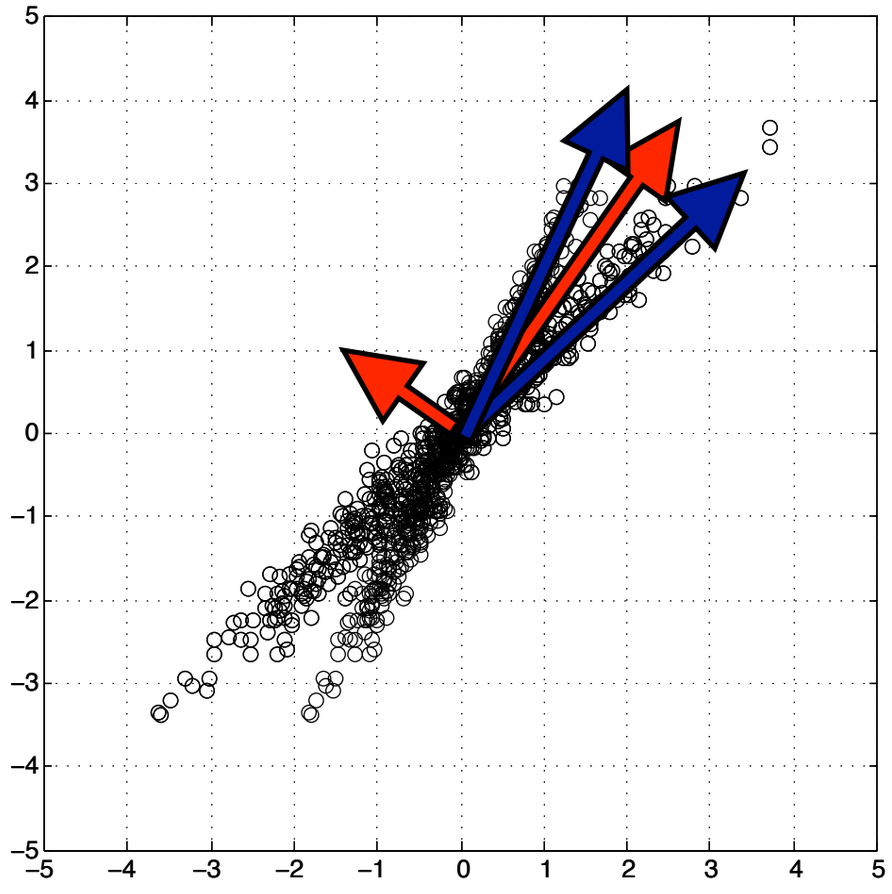
Observed Eigen-values

95% variance



Independent Component Analysis (ICA)

- ICA vs PCA
  - statistical indepedence vs variance
- In ICA do not need different directions to be orthogonal
- Can capture some information better
- Still uses PCA for dimensionality reduction in high dimensions
- ICA is more useful for clustering/ decomposition



non-Gaussian data

Example: Brain Networks

Medial Visual

Lateral Visual

Auditory

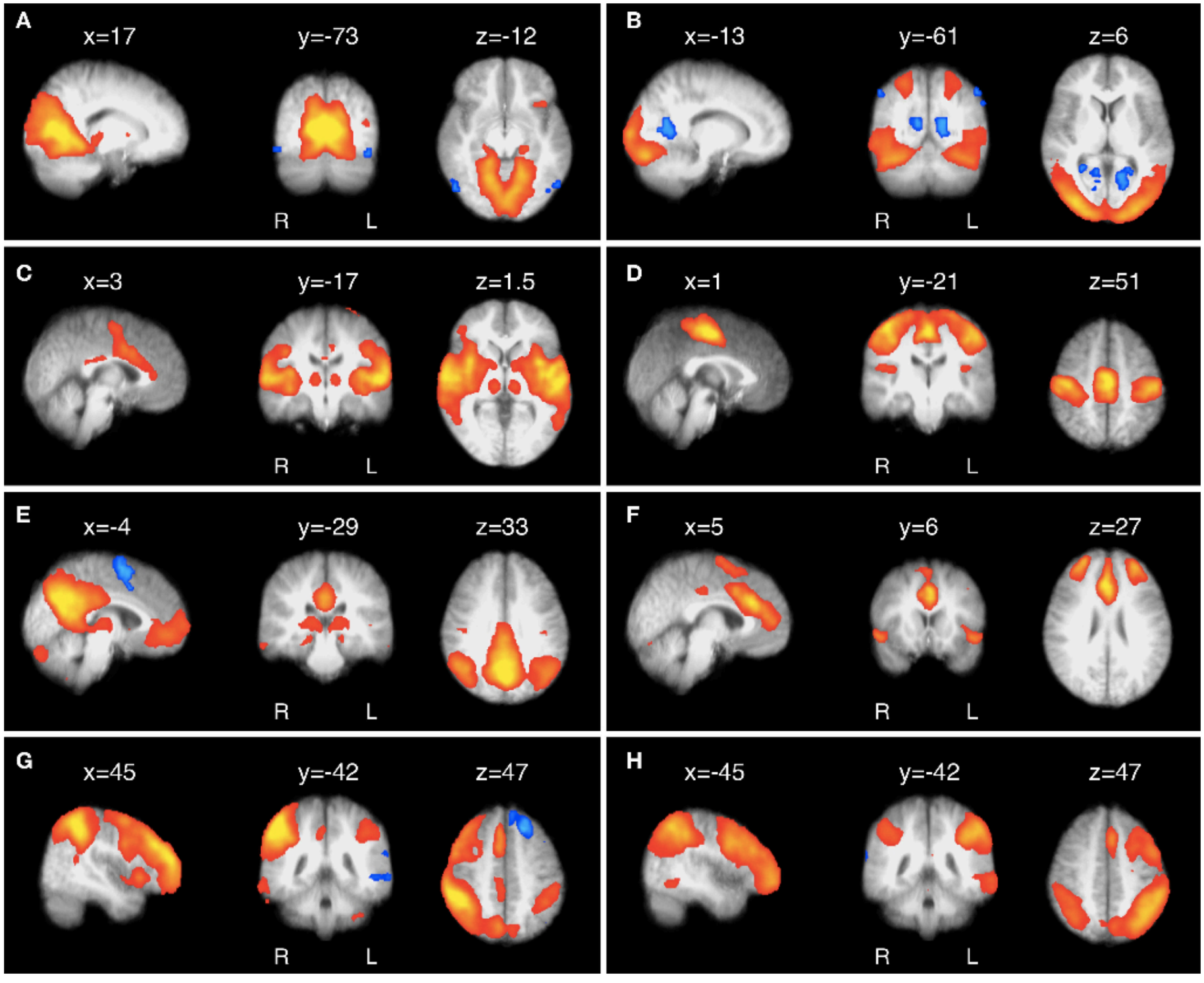
Sensory-Motor

Visuo-Spatial

Executive Control

Dorsal Visual Stream

Dorsal Visual Stream



Examples

- Let us consider some examples of how to select: methods, number of clusters, and parameters in practice
  - resting-state fMRI
  - tissue-type segmentation
  - disease clustering
  - consumer grouping
  - semi-supervised
    - few labels, not enough for supervised: label propagation
  - search engines
  - image segmentation
  - cleaning up ground truth labels

Summary

- There are many different types of clustering method
- They are based on a number of different underlying principle
- Each has different strengths and weaknesses
- Good performance will depend on the task/application
- Data visualisation is an unsupervised task and useful for clustering and for supervised learning
- Dimensionality reduction are data visualisation methods are often closely related
- Choosing the “right” method and “right” parameters is a skill that is often subjective and/or requires domain expertise

Further Reading

- Chapters 8 and 9 of Geron's book cover dimensionality reduction and clustering
  - More detail than necessary - skim these parts
  - Methods for selecting number of clusters do not generalise well
  - Skim over many methods that are often used in practice

- Documentation pages of sklearn are highly informative
  - Useful example code
  - Good range of up-to-date methods
  - Detail provided is variable, some good practical advice