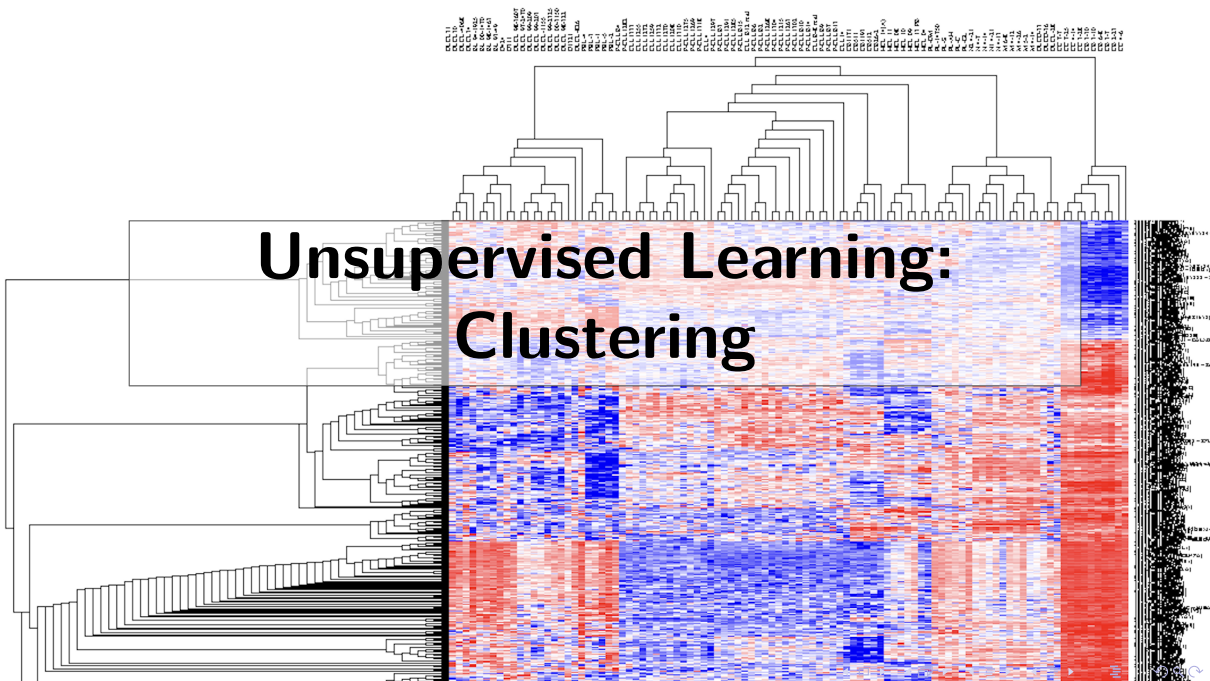


Unsupervised Learning: Clustering

Learning outcomes:

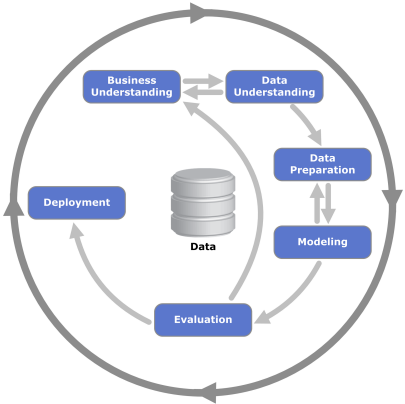


Figure: The CRISP-DM process.

- Define two alternative clustering paradigms;
- Define the k-means objective function;
- Describe two methods for choosing k;
- Define dendrograms and relate to hierarchical clustering.

What is clustering?

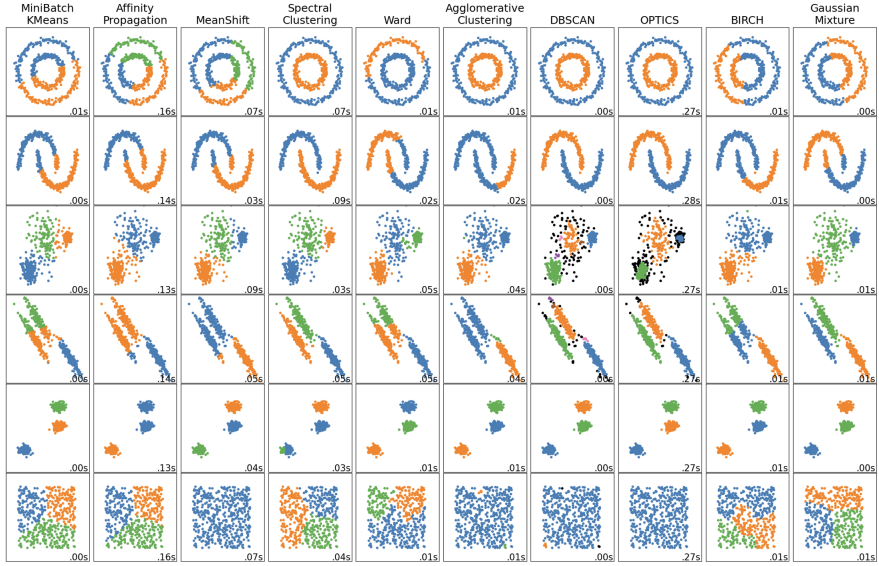
- Clustering is...
  - an **unsupervised** technique...
  - ...to divide data into different groups, where...
  - ...samples in each group are similar to each other and different from other groups;

- How groups can be used?
  - Exploratory analysis: groups can be analyzed in depth;
  - Feature engineering: groups can be used as a feature;
  - Labeling samples: groups can be used as a label/target/outcome.

- Examples:
  - Identifying similar documents;
  - Social network analysis: community detection;
  - Molecular biology: determine the group structure in a population of cells;
  - Consumer market segmentation: identify customers with similar profile in order to...
    - ...recommend personalized products;
    - ...send personalized advertisements;
    - ...align product messaging appropriately.

Clustering as Optimization

- The clustering problem is an optimization problem;
- To solve an optimization problem we define an “objective function” to mathematically encode what we consider to be ideal properties of clusters;
- Different clustering methods have different objective functions;
- These objective functions result in different ways of clustering the data and in different cluster shapes.


MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
OPTICS
BIRCH
Gaussian Mixture
.01s .16s .07s .07s .01s .01s .00s .27s .01s .00s
.00s .14s .03s .09s .02s .02s .00s .28s .01s .00s
.00s .13s .09s .03s .05s .04s .00s .27s .01s .01s
.00s .14s .05s .05s .05s .04s .06s .17s .01s .01s
.00s .13s .04s .03s .01s .01s .00s .27s .01s .00s
.00s .16s .07s .04s .01s .01s .00s .27s .01s .01s

Centroid-Based Clustering

k-Means Clustering

- k-means was one of the first clustering algorithms to be developed;
- It is a popular algorithm, and it is still widely used:
  - Fast;
  - Simple;
  - Intuitive;
- It is a distance-based clustering algorithm: points that are close to each other are similar.

How does k-means clustering work?

- Given k (number of clusters), the goal of k-means is to find best assignment of observations to clusters;
  - Uses distance to measure intra-cluster "coherence";
  - Finding the optimal assignment is a very difficult task: k-means uses a simple iterative algorithm, which finds a **local optimum**;

- Building blocks of the algorithm:
  - A **measure of central tendency** (e.g. the mean) to create centroids;
  - A **distance metric** (e.g. Euclidean distance) to measure distance between samples/centroids;
  - A **clustering metric** – sum of squares;

How does k-means clustering work?

- Given k (number of clusters), the goal of k-means is to find best assignment of observations to clusters;
  - Uses distance to measure intra-cluster "coherence";
  - Finding the optimal assignment is a very difficult task: k-means uses a simple iterative algorithm, which finds a **local optimum**;

- Building blocks of the algorithm:
  - A **measure of central tendency** (e.g. the mean) to create centroids;
  - A **distance metric** (e.g. Euclidean distance) to measure distance between samples/centroids;
  - A **clustering metric** – sum of squares;

- Steps of the algorithm:
  1. Specify the number of clusters k;
  2. Initialize: randomly initialize the k centroids;

How does k-means clustering work?

- Given k (number of clusters), the goal of k-means is to find best assignment of observations to clusters;
  - Uses distance to measure intra-cluster "coherence";
  - Finding the optimal assignment is a very difficult task: k-means uses a simple iterative algorithm, which finds a **local optimum**;

- Building blocks of the algorithm:
  - A **measure of central tendency** (e.g. the mean) to create centroids;
  - A **distance metric** (e.g. Euclidean distance) to measure distance between samples/centroids;
  - A **clustering metric** – sum of squares;

- Steps of the algorithm:
  1. Specify the number of clusters k;
  2. Initialize: randomly initialize the k centroids;
  3. Assign each data sample to the cluster whose centroid is the closest;

How does k-means clustering work?

- Given k (number of clusters), the goal of k-means is to find best assignment of observations to clusters;
  - Uses distance to measure intra-cluster "coherence";
  - Finding the optimal assignment is a very difficult task: k-means uses a simple iterative algorithm, which finds a **local optimum**;

- Building blocks of the algorithm:
  - A **measure of central tendency** (e.g. the mean) to create centroids;
  - A **distance metric** (e.g. Euclidean distance) to measure distance between samples/centroids;
  - A **clustering metric** – sum of squares;

- Steps of the algorithm:
  1. Specify the number of clusters k;
  2. Initialize: randomly initialize the k centroids;
  3. Assign each data sample to the cluster whose centroid is the closest;
  4. For each of the k clusters, update the centroid based on the new cluster assignment (e.g. use within-cluster mean). If the centroids don't change stop. Else go to step 3.

How does k-means clustering work?

- Given k (number of clusters), the goal of k-means is to find best assignment of observations to clusters;
  - Uses distance to measure intra-cluster "coherence";
  - Finding the optimal assignment is a very difficult task: k-means uses a simple iterative algorithm, which finds a **local optimum**;

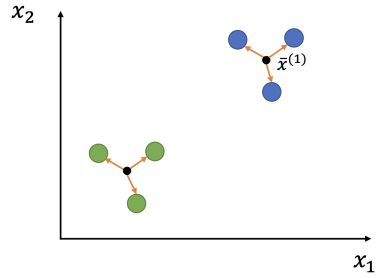
- Building blocks of the algorithm:
  - A **measure of central tendency** (e.g. the mean) to create centroids;
  - A **distance metric** (e.g. Euclidean distance) to measure distance between samples/centroids;
  - A **clustering metric** – sum of squares;

- Steps of the algorithm:
  1. Specify the number of clusters k;
  2. Initialize: randomly initialize the cluster assignment;
  3. For each of the k clusters, update the centroid based on the new cluster assignment (e.g. use within-cluster mean). If the centroids don't change stop. Else go to step 3.
  4. Assign each data sample to the cluster whose centroid is the closest.

The sum of squares

- k-means works with the sum of squares which is similar to standard deviation;
- Given cluster C for sample $x_i = (x_{i,1}, \ldots, x_{i,m})$:

$$SS(i, C) = \sum_{j=1}^{m} (x_{i,j} - \overline{x}_j(C))^2 \text{ where } \overline{x}_j(C) = \underbrace{\frac{1}{|C|} \sum_{i \in C} x_{i,j}}_{\substack{\text{mean of } j\text{th} \\ \text{feature in } C}};$$



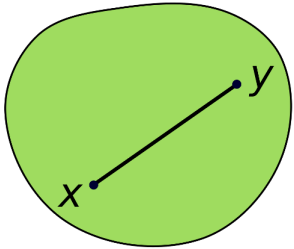
- A globally optimal clustering for k would find:

$$\min_{\mathcal{P}(k)} \left\{ \sum_{C \in \mathcal{P}(k)} \sum_{i \in C} SS(i, C) \right\}$$

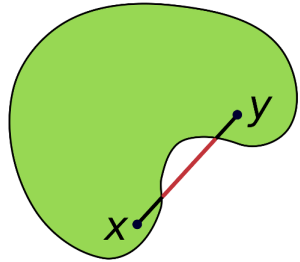
where $\mathcal{P}(k) =$ partition of k groups.

The sum of squares: modeling assumption

Using sum of squares as the objective function means that k-means assumes that the clusters are convex:

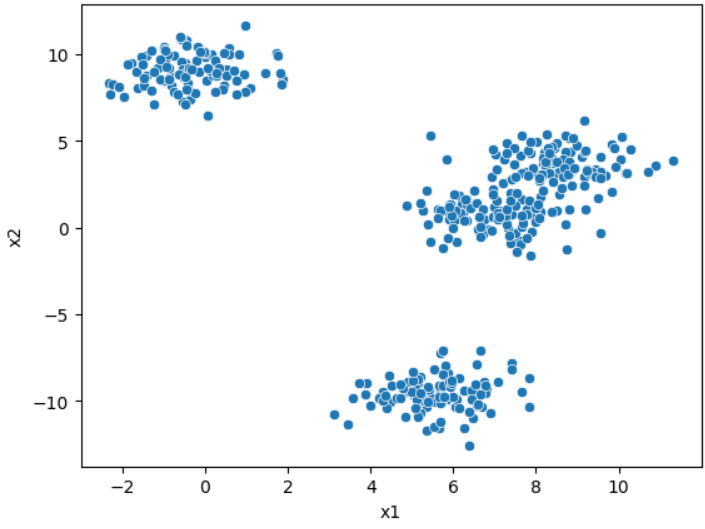


An example of convex set.

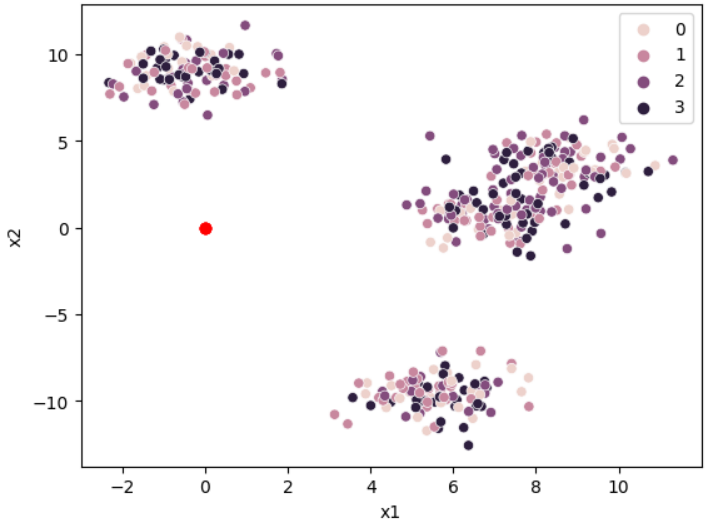


An example of non-convex set.

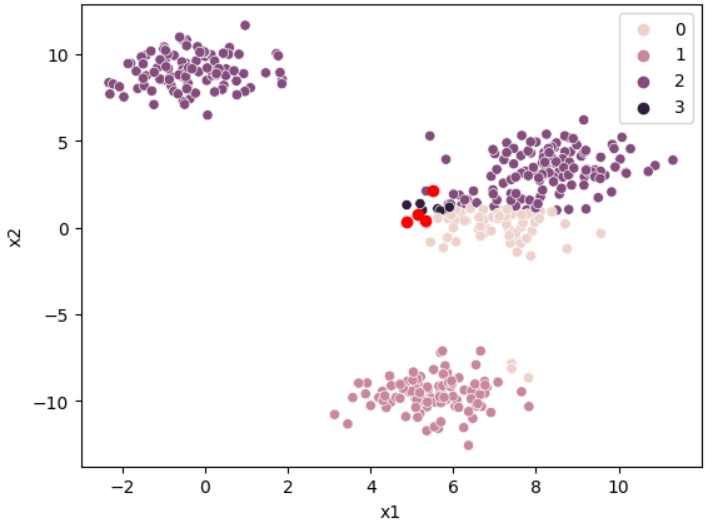
k-means example data



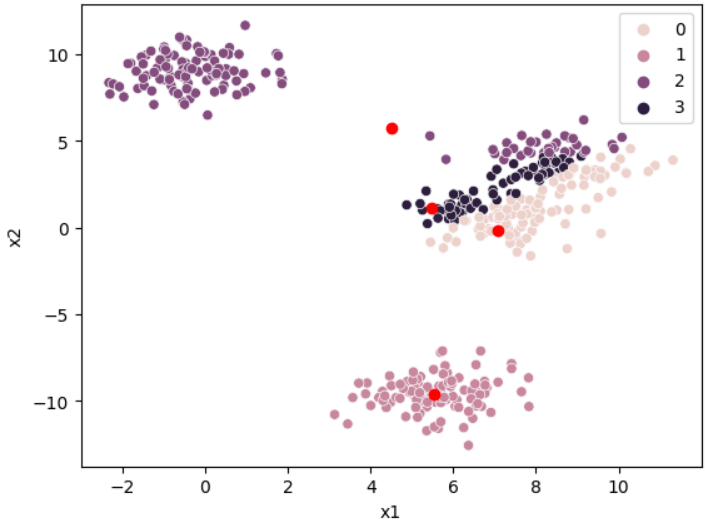
k-means initialization



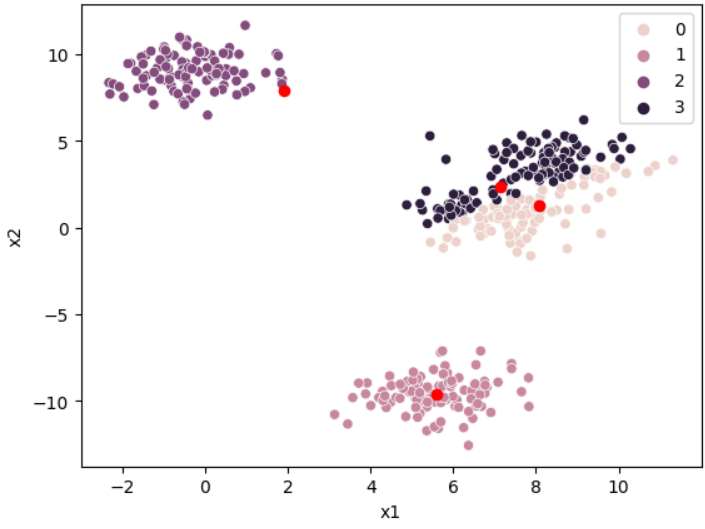
k-means iteration 1



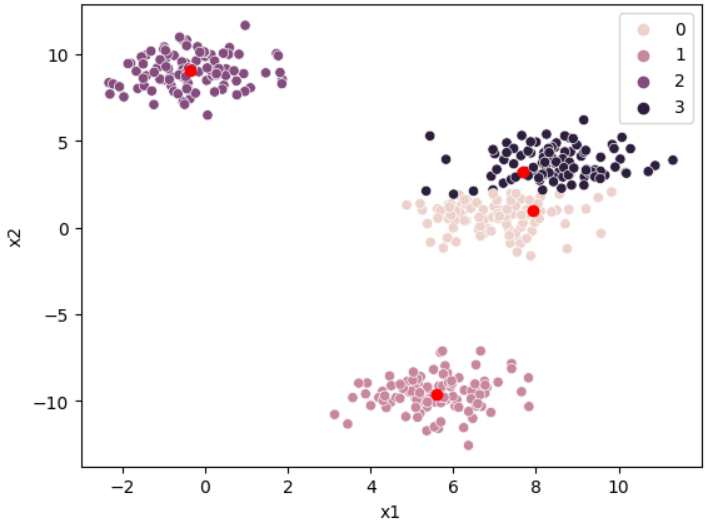
k-means iteration 2



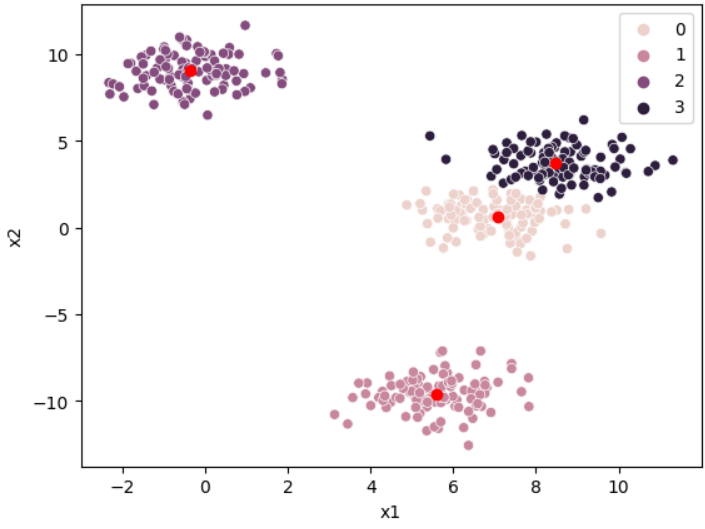
k-means iteration 3



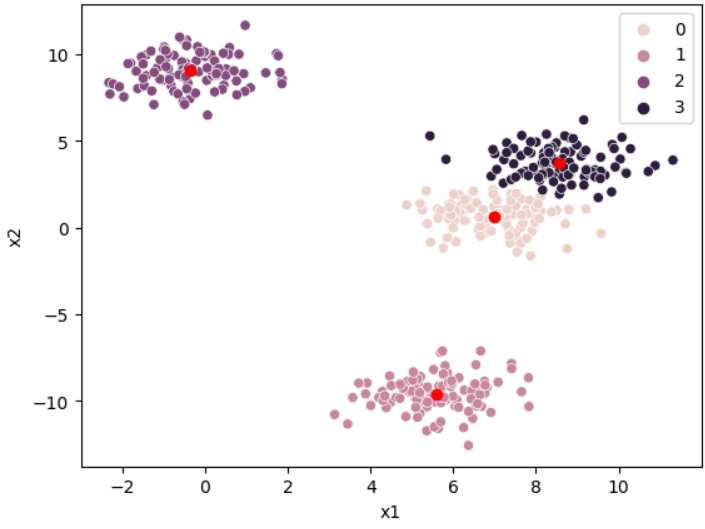
k-means iteration 4



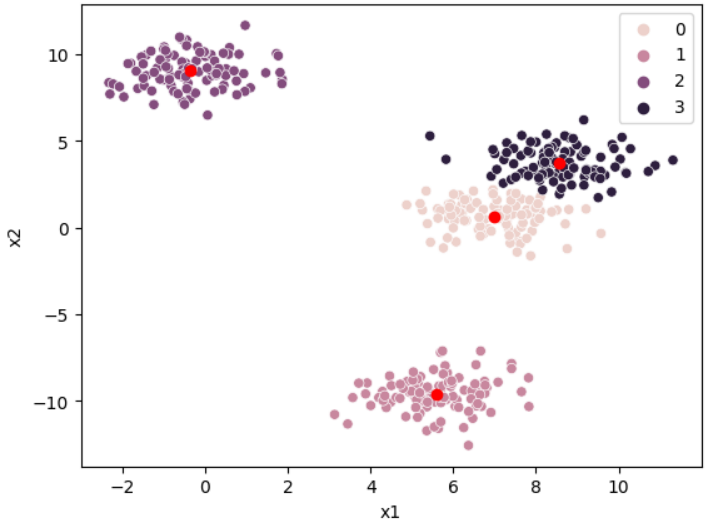
k-means iteration 5



k-means iteration 6



k-means iteration 7



How do we choose k? Method 1: the elbow

- Loop over different choices for k;
- Choose k as the point at which the sum of squares starts descending much more slowly;

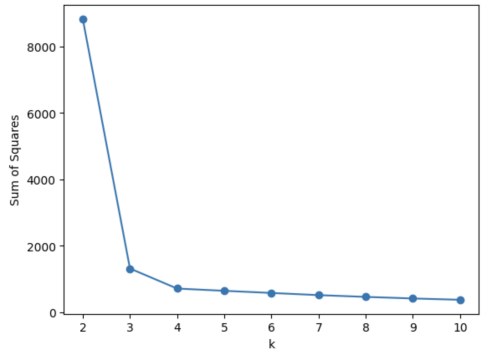


Figure: Elbow method can be misleading. It IDs 3 but there are 4 clusters in our data.

How do we choose k? Method 1: the elbow

- Loop over different choices for k;
- Choose k as the point at which the sum of squares starts descending much more slowly;
- IRL, may not be useful...

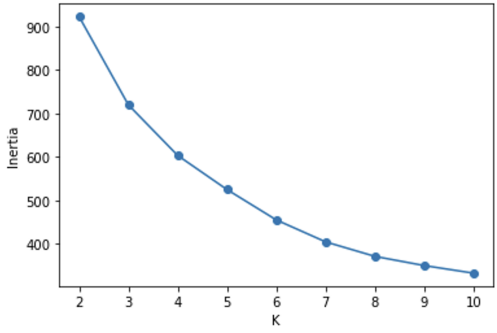


Figure: Where is the elbow?!

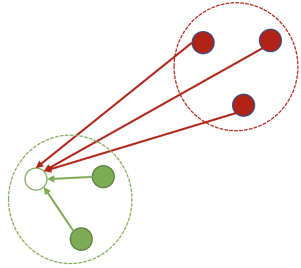
How do we choose k? Method 2: Silhouette Scores

- **Silhouette Scores**: metric for evaluating any clustering (not only to choose the best k for k-means);
- Silhouette Coefficient is defined for each sample:

$$s = \frac{b - a}{\max\{a, b\}};$$

  - a = mean distance between a sample and all other points in the same cluster;
  - b = mean distance between a sample and all other points in the next nearest cluster;

- The sklearn version returns the average of silhouette coefficients over all samples.

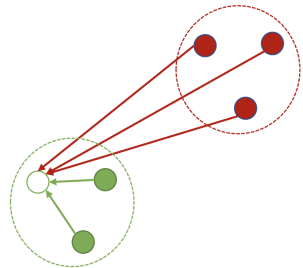


How do we choose k? Method 2: Silhouette Scores

- Silhouette Coefficient is defined for each sample:

$$-1.0 < s = \frac{b - a}{\max\{a, b\}} < 1.0;$$

- Rules of thumb for interpreting:
  - $-1 \Rightarrow$ clustering is incorrect;
  - $0 \Rightarrow$ clusters are overlapping;
  - $< 0.25 \Rightarrow$ no substantial structure;
  - $0.26 - -0.5 \Rightarrow$ weak structure (might be artificial);
  - $0.51 - -0.70 \Rightarrow$ reasonable structure;
  - $0.71 - -1.0 \Rightarrow$ strong structure.



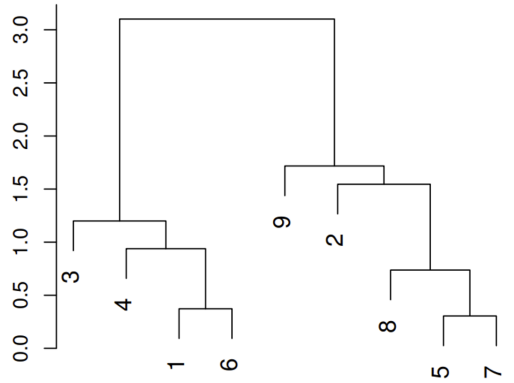
Hierachical Clustering

Agglomerative Hierarchical Clustering

- Agglomerative hierarchical clustering is a non-centroid alternative to k-means – works in a "bottom-up" fashion;
- Added advantages:
  - Does not require a pre-specified choice for the number of clusters (k) to run;
  - Results in attractive tree-based representation of the observations: dendrogram (easier interpretation of the clusters);
- However, it is expensive in terms of compute and memory – does not scale well with the number of samples.

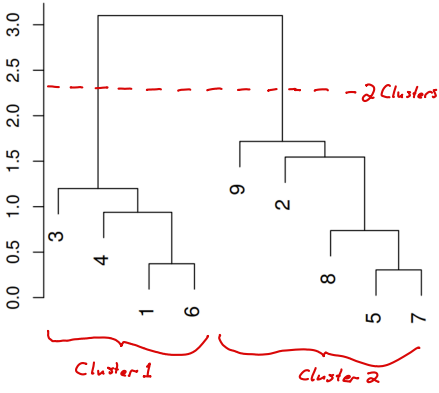
The Dendrogram

- **Dendrogram**: a tree-like visualization of how data samples were combined by the algorithm;
- At the bottom each terminal node (leaf) represents one row in your data;
- Each non-terminal node is a cluster containing all terminal nodes;
- Height indicates the degree of "dissimilarity" between clusters:
  - The clusters that merge at the very bottom of the tree are quite similar;
  - The clusters that merge at the top of the tree will tend to be different.



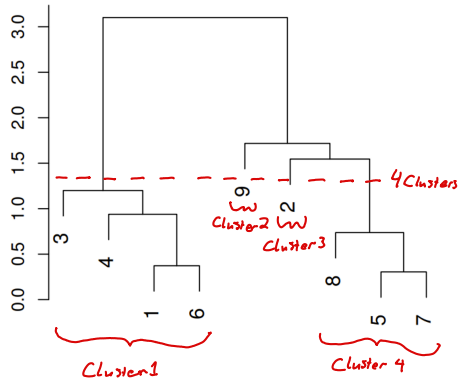
The Dendrogram

- Unlike k-means, hierarchical clustering doesn't require the user to specify the number of clusters beforehand;
- Can decide from the dendogram the appropriate number of clusters (either manually or algorithmically):
  - Make a horizontal cut across the dendrogram;
  - Take the first set of nodes beneath the cut as your clusters;



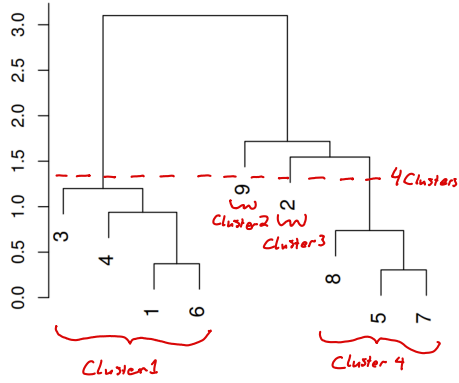
The Dendrogram

- Unlike k-means, hierarchical clustering doesn't require the user to specify the number of clusters beforehand;
- Can decide from the dendogram the appropriate number of clusters (either manually or algorithmically):
  - Make a horizontal cut across the dendrogram;
  - Take the first set of nodes beneath the cut as your clusters;



The Dendrogram

- Unlike k-means, hierarchical clustering doesn't require the user to specify the number of clusters beforehand;
- Can decide from the dendogram the appropriate number of clusters (either manually or algorithmically):
  - Make a horizontal cut across the dendrogram;
  - Take the first set of nodes beneath the cut as your clusters;
- Can draw the cut based on:
  - User interpretability;
  - Algorithmic analysis, e.g. silhouette scores.



How does Agglomerative Hierarchical Clustering work?

- Agglomerative (bottom-up approach):
  - we start with each data sample constituting its own single-sample cluster;
  - We then build up larger and larger clusters;

- Building blocks of the algorithm:
  - A **distance metric** (e.g. Euclidean distance) to measure distance between samples;
  - A **dissimilarity metric** to measure the difference between clusters;

How does Agglomerative Hierarchical Clustering work?

- Agglomerative (bottom-up approach):
  - we start with each data sample constituting its own single-sample cluster;
  - We then build up larger and larger clusters;

- Building blocks of the algorithm:
  - A **distance metric** (e.g. Euclidean distance) to measure distance between samples;
  - A **dissimilarity metric** to measure the difference between clusters;

- Steps of the algorithm:
  1. Create an initial set of n clusters with each cluster consisting of a single sample/record/row;

How does Agglomerative Hierarchical Clustering work?

- Agglomerative (bottom-up approach):
  - we start with each data sample constituting its own single-sample cluster;
  - We then build up larger and larger clusters;

- Building blocks of the algorithm:
  - A **distance metric** (e.g. Euclidean distance) to measure distance between samples;
  - A **dissimilarity metric** to measure the difference between clusters;

- Steps of the algorithm:
  1. Create an initial set of n clusters with each cluster consisting of a single sample/record/row;
  2. Compute all pairwise dissimilarities among the clusters and identify the pair of clusters that are least dissimilar;

How does Agglomerative Hierarchical Clustering work?

- Agglomerative (bottom-up approach):
  - we start with each data sample constituting its own single-sample cluster;
  - We then build up larger and larger clusters;

- Building blocks of the algorithm:
  - A **distance metric** (e.g. Euclidean distance) to measure distance between samples;
  - A **dissimilarity metric** to measure the difference between clusters;

- Steps of the algorithm:
  1. Create an initial set of n clusters with each cluster consisting of a single sample/record/row;
  2. Compute all pairwise dissimilarities among the clusters and identify the pair of clusters that are least dissimilar;
  3. Merge these two clusters into one cluster (dissimilarity between these two clusters indicates the height in the dendrogram at which to merge them)

How does Agglomerative Hierarchical Clustering work?

- Agglomerative (bottom-up approach):
  - we start with each data sample constituting its own single-sample cluster;
  - We then build up larger and larger clusters;

- Building blocks of the algorithm:
  - A **distance metric** (e.g. Euclidean distance) to measure distance between samples;
  - A **dissimilarity metric** to measure the difference between clusters;

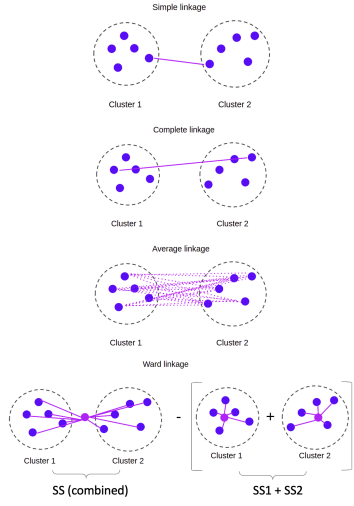
- Steps of the algorithm:
  1. Create an initial set of n clusters with each cluster consisting of a single sample/record/row;
  2. Compute all pairwise dissimilarities among the clusters and identify the pair of clusters that are least dissimilar;
  3. Merge these two clusters into one cluster (dissimilarity between these two clusters indicates the height in the dendrogram at which to merge them)
  4. The total number of clusters is now decreased by 1. If the updated number of cluster is more than 2, return to step 2. Otherwise, stop.

Measures of distance/dissimilarity

- Distance metric between $x_1 = (x_{1,1}, \ldots, x_{1,m})$ and $x_2 = (x_{2,1}, \ldots, x_{2,m})$:

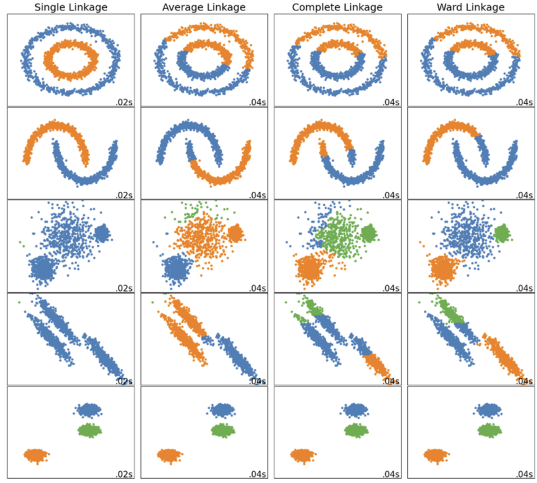
$$d(x_1, x_2) = \sqrt{\sum_j (x_{1,j} - x_{2,j})^2};$$

- Lots of dissimilarity metrics – some examples:
  - **Simple Linkage**: distance between the closest elements;
  - **Complete Linkage**: distance between the furthest elements;
  - **Average Linkage**: average distance between all elements;
  - **Ward Linkage**: If we were to combine to clusters, what is the increase in SS (sum of squares).

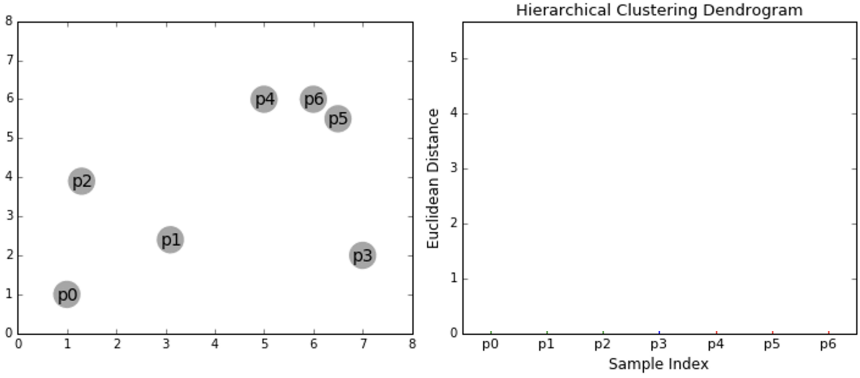


Measures of distance/dissimilarity

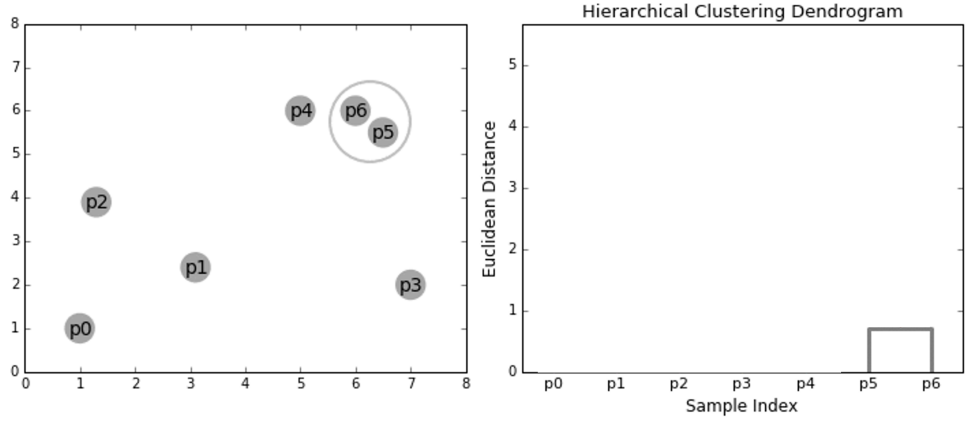
- Simple Linkage can lead to unbalanced trees but can perform well on non-globural data;
- Other types of linkage lead to more balanced trees but can perform worse on non-globular data;
- Can add constraints to make this work better on non-globular data.



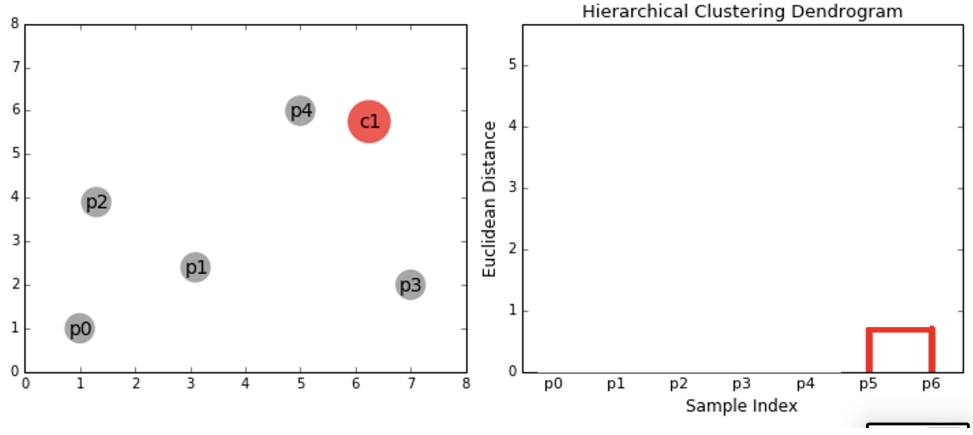
Agglomerative Hierarchical Clustering Example:



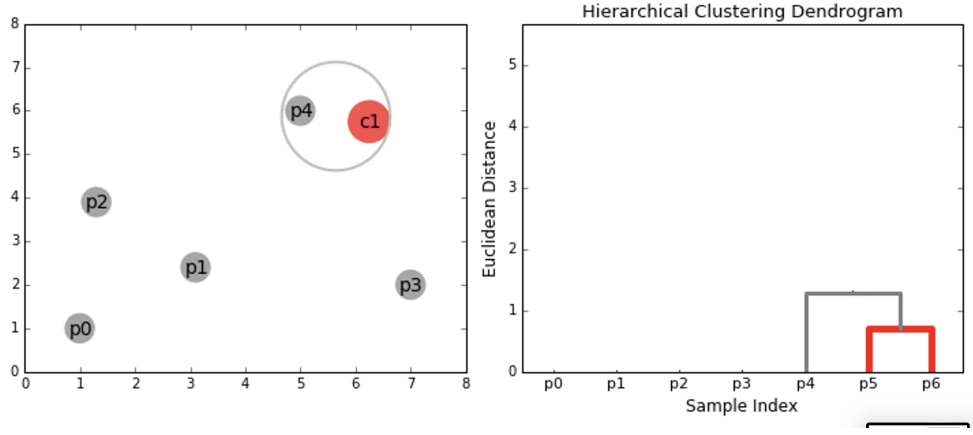
Agglomerative Hierarchical Clustering Example:



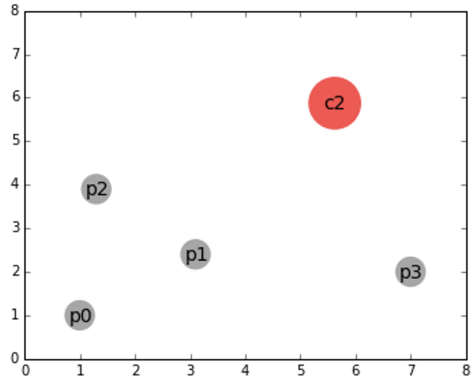
Agglomerative Hierarchical Clustering Example:

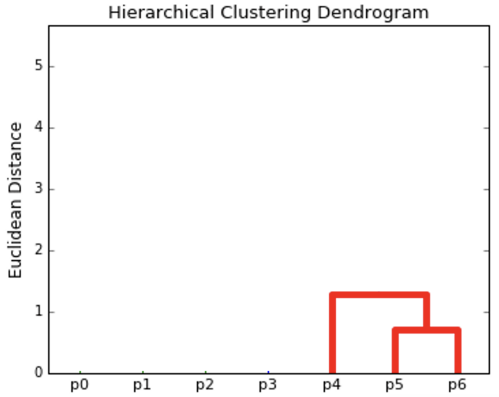


Agglomerative Hierarchical Clustering Example:

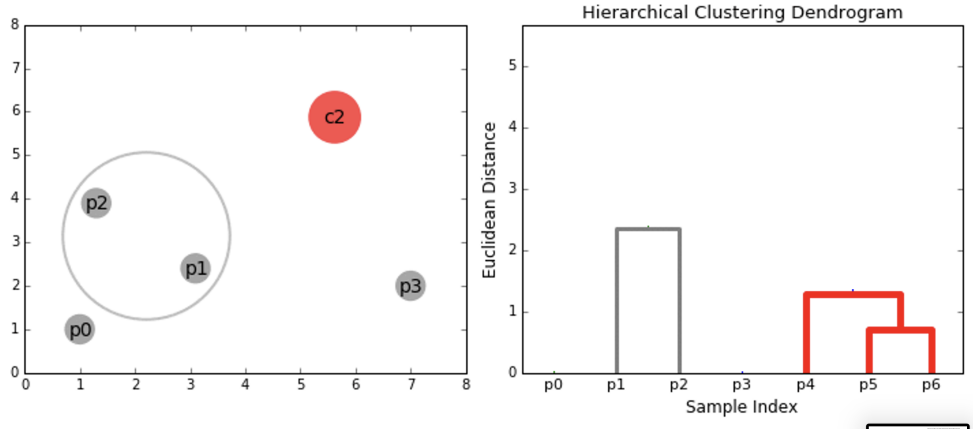


Agglomerative Hierarchical Clustering Example:

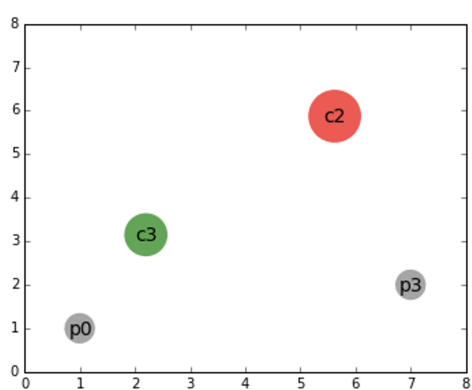


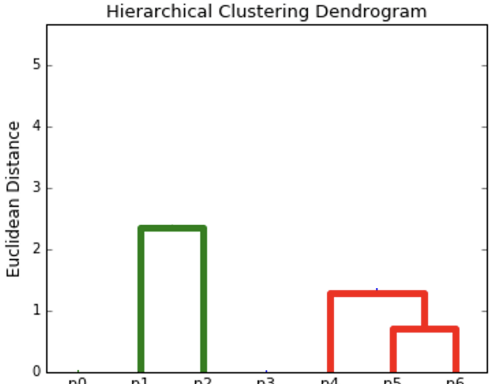


Agglomerative Hierarchical Clustering Example:

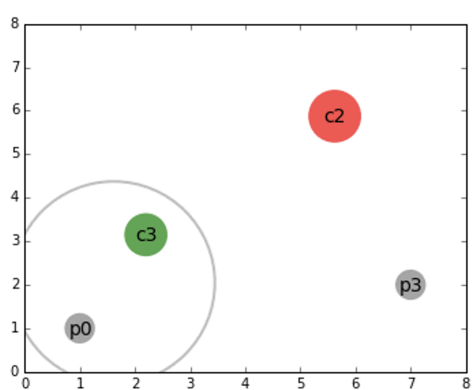


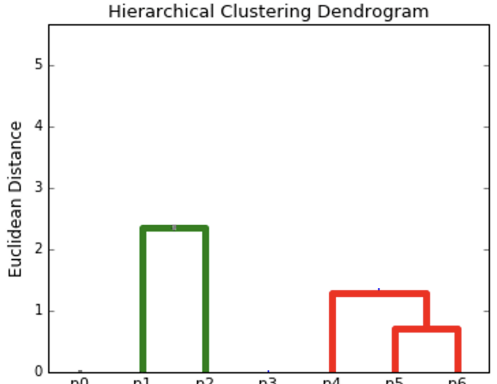
Agglomerative Hierarchical Clustering Example:



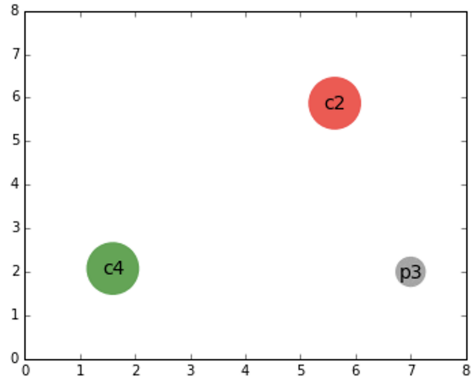


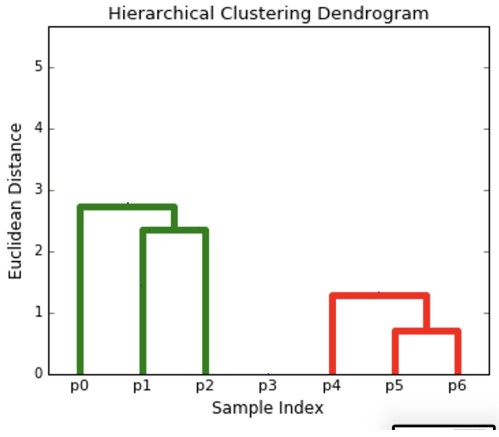
Agglomerative Hierarchical Clustering Example:



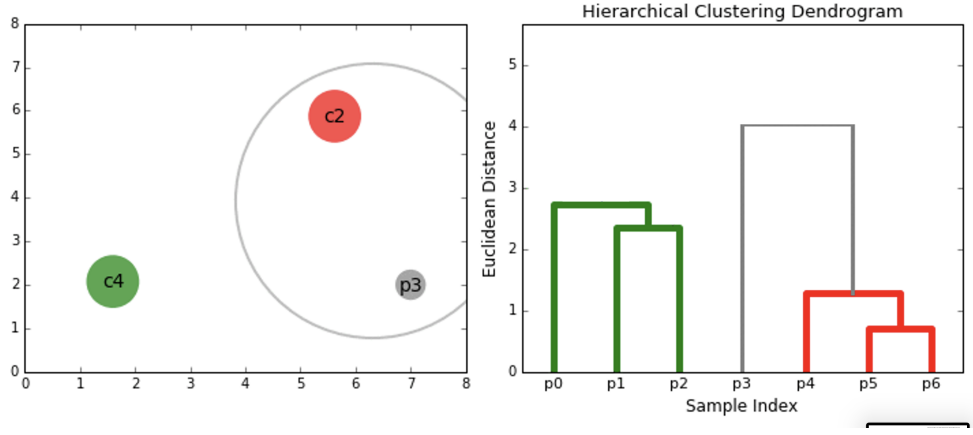


Agglomerative Hierarchical Clustering Example:

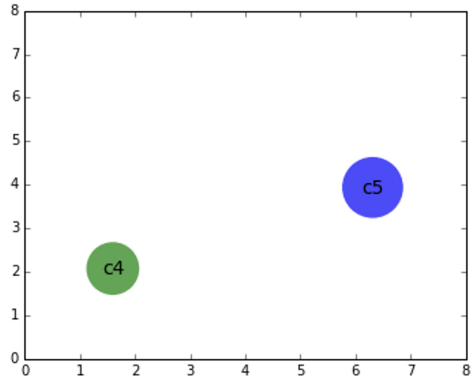


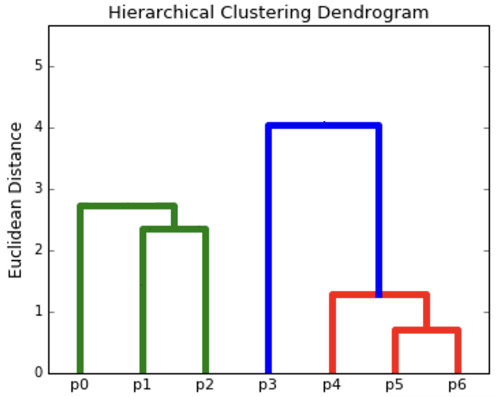


Agglomerative Hierarchical Clustering Example:

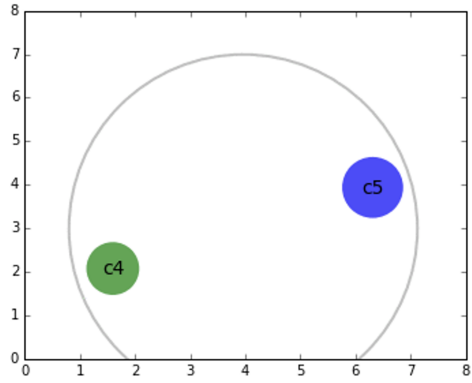


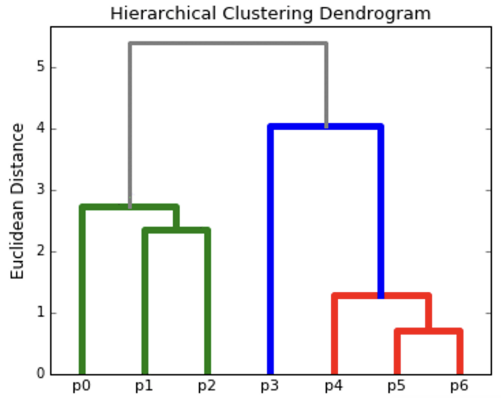
Agglomerative Hierarchical Clustering Example:





Agglomerative Hierarchical Clustering Example:





Agglomerative Hierarchical Clustering Example: