UNSUPERVISED LEARNING PART 2: HIERARCHICAL CLUSTERING
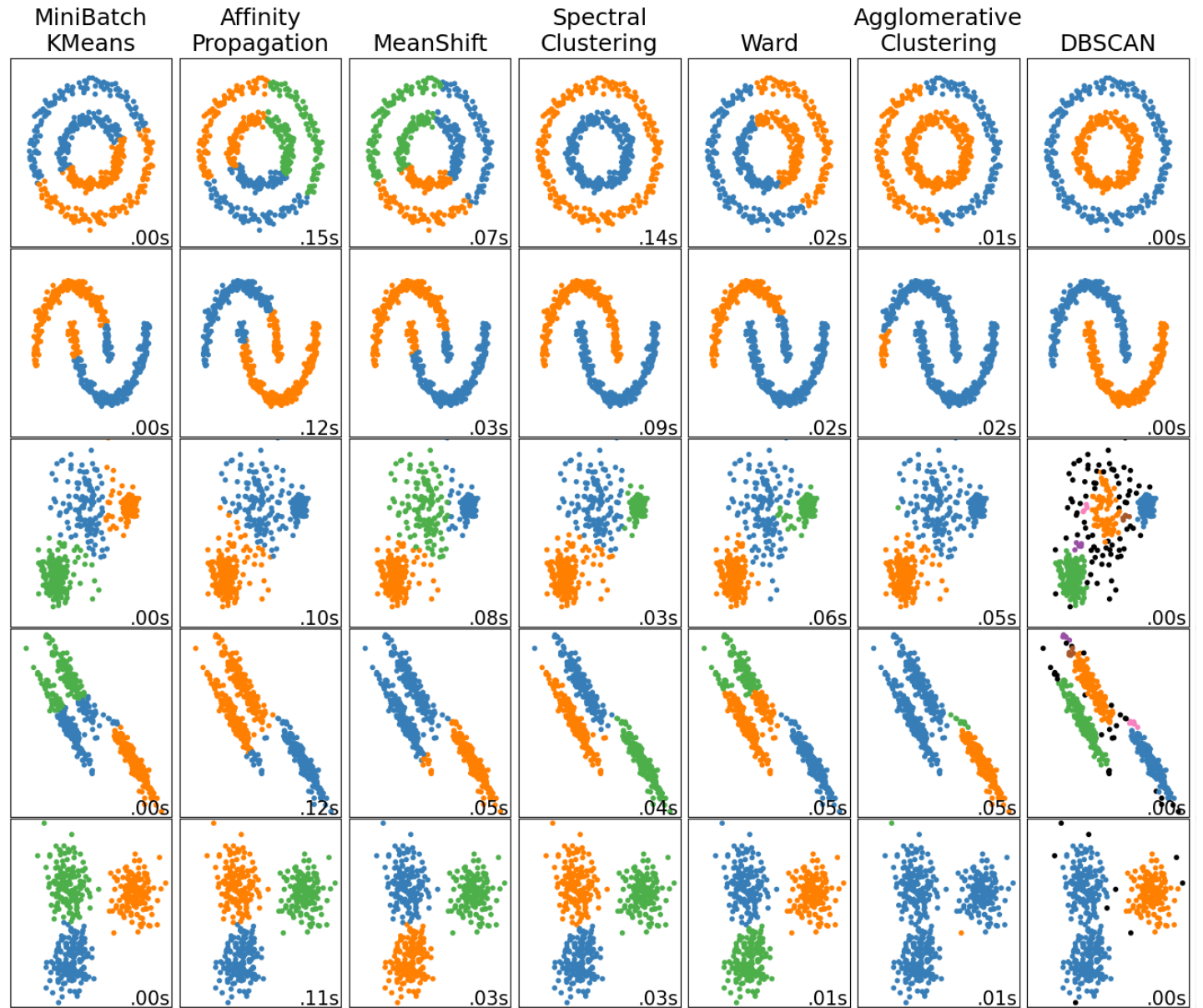
OUTLINE

- Hierarchical Clustering
  - Agglomerative
  - Divisive

- Density Based Clustering
  - DBSCAN

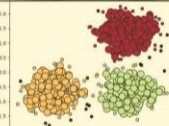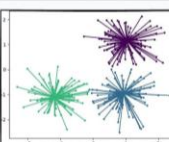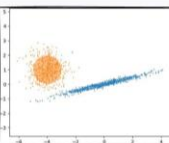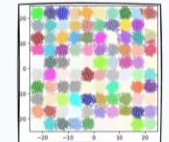| Clustering Algorithm Type		Clustering Methodology	Algorithm(s)
	Centroid-based	Cluster points based on proximity to centroid	KMeans KMeans++ KMedoids
	Connectivity-based	Cluster points based on proximity between clusters	Hierarchical Clustering (Agglomerative and Divisive)
	Density-based	Cluster points based on their density instead of proximity	DBSCAN OPTICS HDBSCAN
	Graph-based	Cluster points based on graph distance	Affinitiy Propagation Spectral Clustering
	Distribution-based	Cluster points based on their likelihood of belonging to the same distribution.	Gaussian Mixture Models (GMMs)
	Compression-based	Transform data to a lower dimensional space and then perform clustering	BIRCH

BYU

HIERARCHICAL CLUSTERING

There are two types of Hierarchical clustering techniques:

- **Agglomerative**

- **Divisive**

■ Agglomerative clustering is a **bottom-up** approach. Start with every point as a cluster and combine points till only a single cluster. (most common)

■ Divisive clustering is a **top-down** approach. Start with one large cluster and divide it into two, three, four, or more clusters.


**BYU**
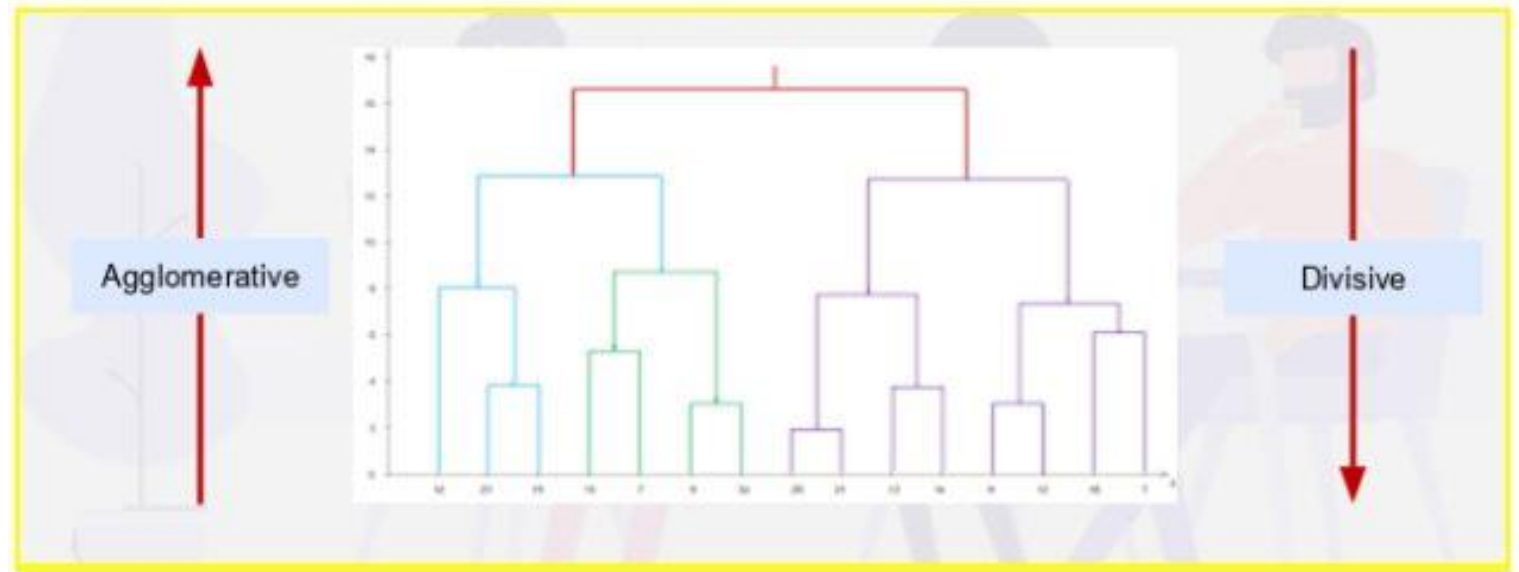
COMPARISON BETWEEN AGGLOMERATIVE AND DIVISIVE CLUSTERING

Parameters	Agglomerative Clustering	Divisive Clustering
Category	Bottom-up approach	Top-down approach
Approach	• Each data point starts in its own cluster, and the algorithm recursively merges the closest pairs of clusters until a single cluster containing all the data points is obtained.	• All data points start in a single cluster, and the algorithm recursively splits the cluster into smaller sub-clusters until each data point is in its own cluster.
Complexity Level	• Agglomerative clustering is generally more computationally expensive, especially for large datasets, as this approach requires the calculation of all pairwise distances between data points. $O(n^3)$–$O(n^2 log(n))$. But new versions come close to linear $O(n)$ for certain use cases.	• Comparatively less expensive as divisive clustering only requires the calculation of distances between sub-clusters, which can reduce the computational burden. $O(n^2)$
Outliers	• Agglomerative clustering can handle outliers better than divisive clustering since outliers can be absorbed into larger clusters	• Divisive clustering may create sub-clusters around outliers, leading to suboptimal clustering results.
Interpretability	• Agglomerative clustering tends to produce more interpretable results since the dendrogram shows the merging process of the clusters, and the user can choose the number of clusters based on the desired level of granularity.	• Divisive clustering can be more difficult to interpret since the dendrogram shows the splitting process of the clusters, and the user must choose a stopping criterion to determine the number of clusters.
Implementation	• Scikit-learn provides multiple linkage methods for agglomerative clustering, such as "single," "complete," "average," and "ward."	• Divisive clustering is not implemented in Scikit-learn.
Example	• Agglomerative Clustering is used in Image segmentation, Customer segmentation, Social network analysis, Document clustering, genomics, etc.	• Divisive Clustering is used in Market segmentation, Anomaly detection, Biological classification, Natural language processing, etc.

BYU

DENDROGRAM

- The output of a hierarchical clustering algorithm is a *dendrogram*. A **dendrogram** is a tree that shows the order in which clusters are grouped together and the distances between clusters.

- Parts of a dendrogram are listed below:

  - A **clade** is a branch of a dendrogram or a vertical line.

  - A **link** is a horizontal line that connects two clades, whose height gives the distance between clusters.

  - A **leaf** is the terminal end of each clade in a dendrogram, which represents a single instance.
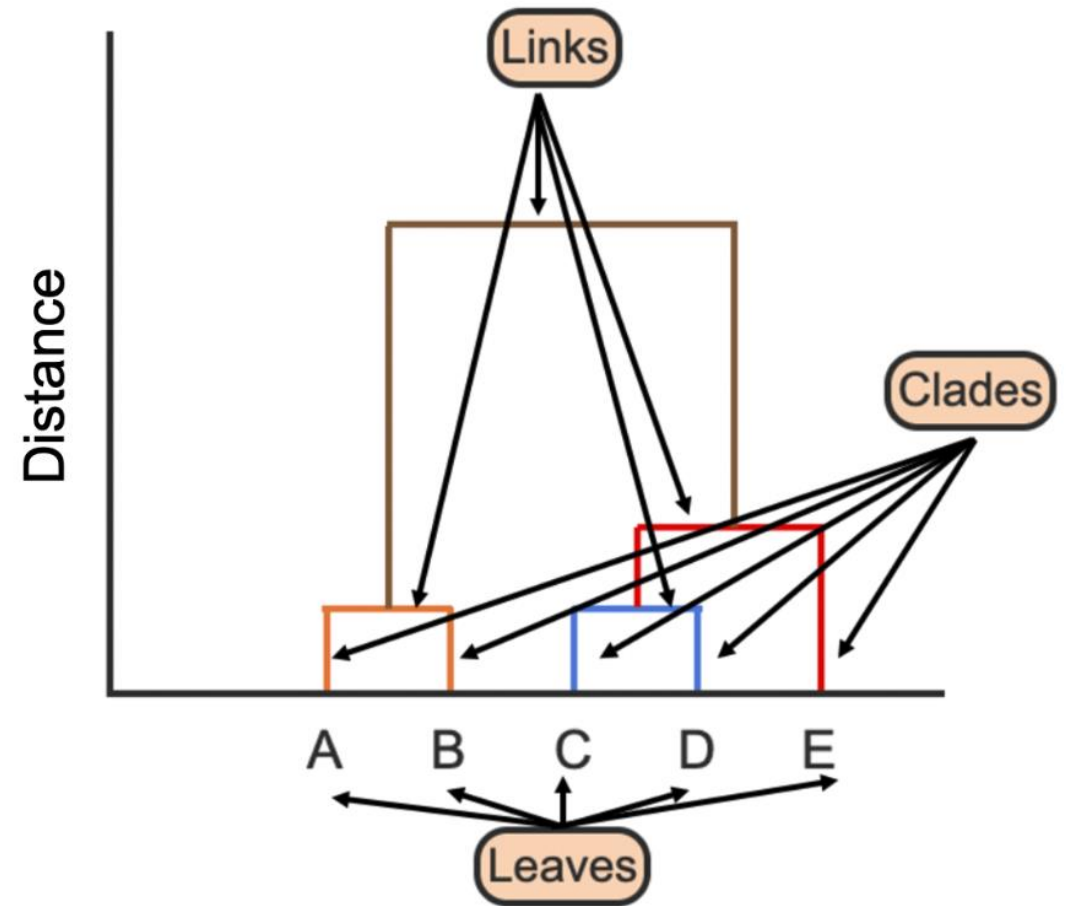
AGGLOMERATIVE CLUSTERING INTUITION

Hierarchical Clustering Dendrogram
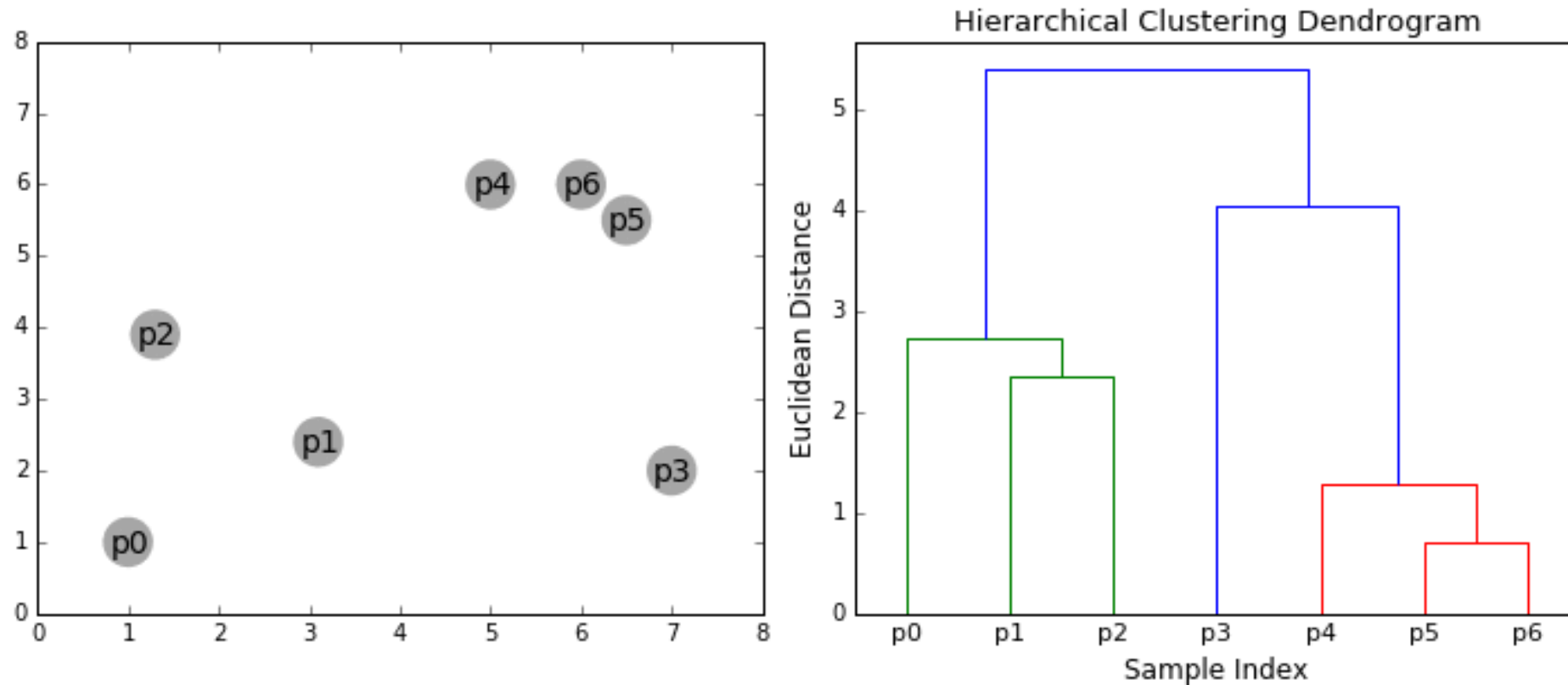
AGGLOMERATIVE CLUSTERING ALGORITHM

Steps:

1. Compute the proximity (distance) matrix between each point or cluster

2. Initialize: Let each data point be a cluster

3. **Repeat**

4. Merge the two closest clusters

5. Update the proximity matrix

6. **Until** only a single cluster remains

The key operation is the computation of the proximity (distance) of two clusters.

To do this, we define different approaches to measuring inter-cluster distance. These approaches are called "**Linkages**."

BYU

PROXIMITY MATRIX

Clusters	x_1	x_2	x_3	...	x_n
x_1	0	$d(x_1, x_2)$	$d(x_1, x_3)$	...	$d(x_1, x_n)$
x_2	$d(x_2, x_1)$	0	$d(x_2, x_3)$	...	$d(x_2, x_n)$
x_3	$d(x_3, x_1)$	$d(x_3, x_1)$	0	...	$d(x_3, x_n)$
...	...	...	...	0	...
x_n	$d(x_n, x_1)$	$d(x_n, x_1)$	$d(x_n, x_1)$	...	0

- Calculate all distances between clusters and/or points
- Symmetric Matrix
- Distances to self = 0
- Use Euclidean, Manhattan, or other
- Update the Proximity Matrix after each iteration
- Merge clusters using Linkage Function

BYU

LINKAGE FUNCTION INTUITION
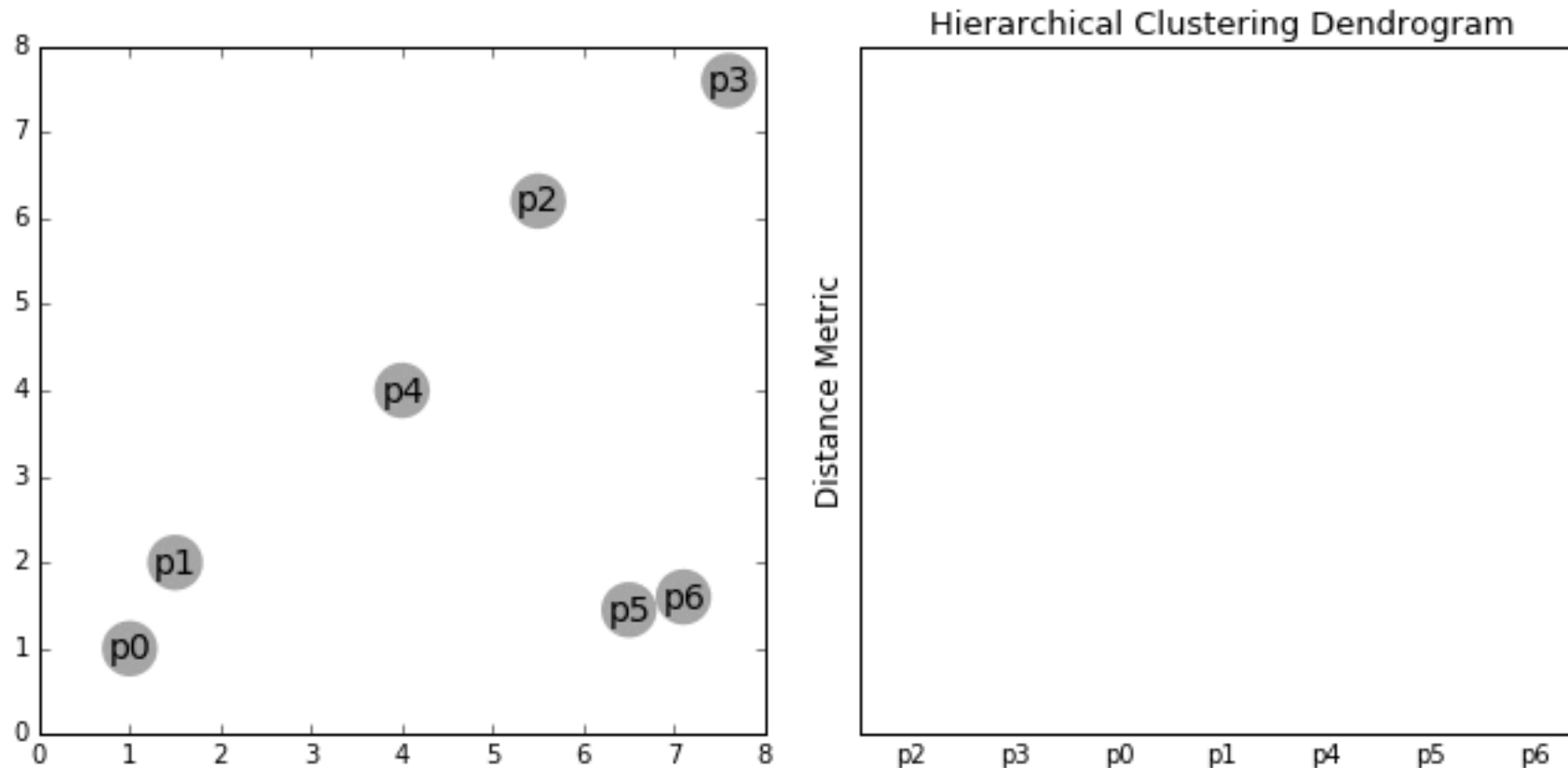
Hierarchical clustering is heavily influenced by two factors…

LINKAGE METHODS

Linkage methods are crucial as they determine how clusters are formed. Here are some standard linkage methods used in hierarchical clustering:

1. **Single Linkage:** (aka Nearest Neighbor) This method joins clusters based on the minimum distance between their data points. It tends to create long, "stringy" clusters and is sensitive to outliers.

2. **Complete Linkage:** Connects clusters by their maximum pairwise distance, resulting in compact, spherical clusters. However, it can be sensitive to outliers as well.

3. **Average Linkage:** This method calculates the average distance between data points in two clusters. It produces balanced clusters and is less sensitive to outliers than single or complete linkage.

4. **Ward's Linkage:** Ward's method minimizes the increase in variance within the clusters when merging them. It often produces equally sized clusters and is suitable for minimizing the variance in the data.

5. **Centroid Linkage**: Combines clusters with minimum distance between centroids (Mean, Median, Medoid, Mode)

These linkage methods play a crucial role in shaping the hierarchical clustering output, and the choice depends on the specific characteristics of your data and your analysis goals.

BYU
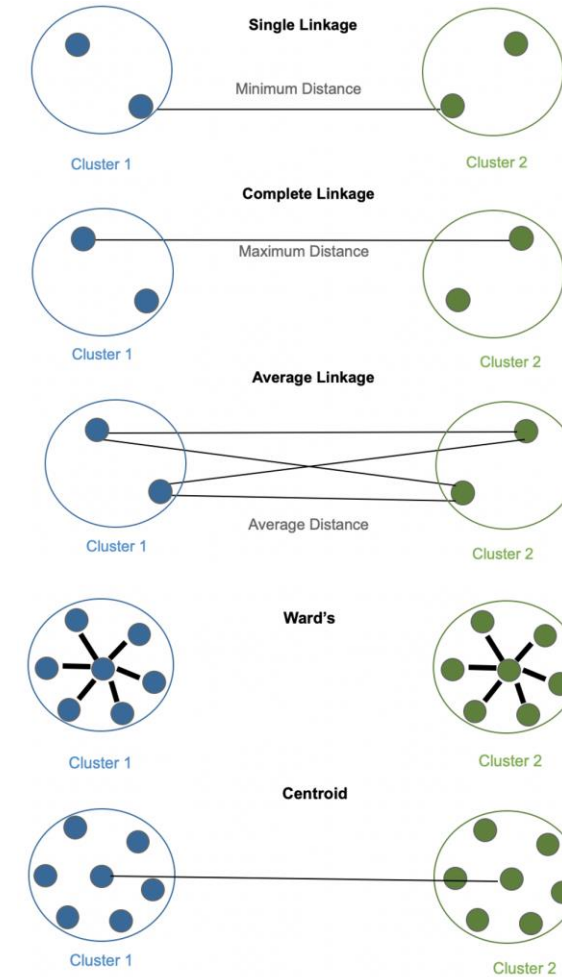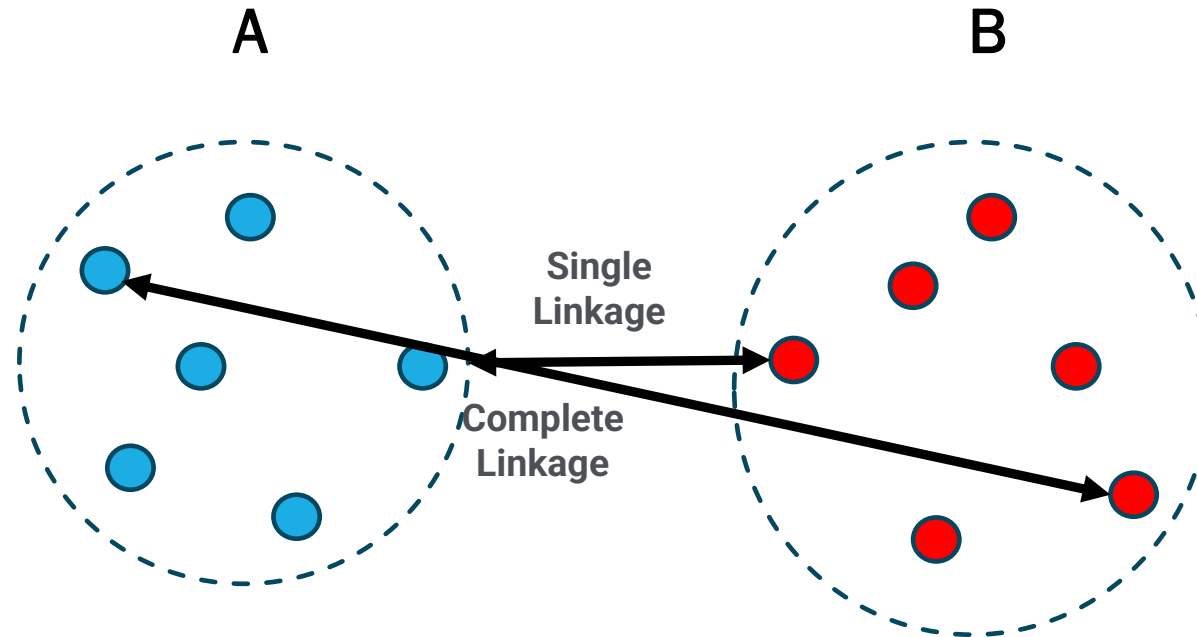
LINKAGE METHOD EXAMPLES

LINKAGE METHOD EXAMPLES

A
B

Average
Linkage

BYU

LINKAGE METHOD EXAMPLES

A

B

Centroid
Linkage

BYU

LINKAGE METHOD EXAMPLES

A   B

Ward's
Linkage

+
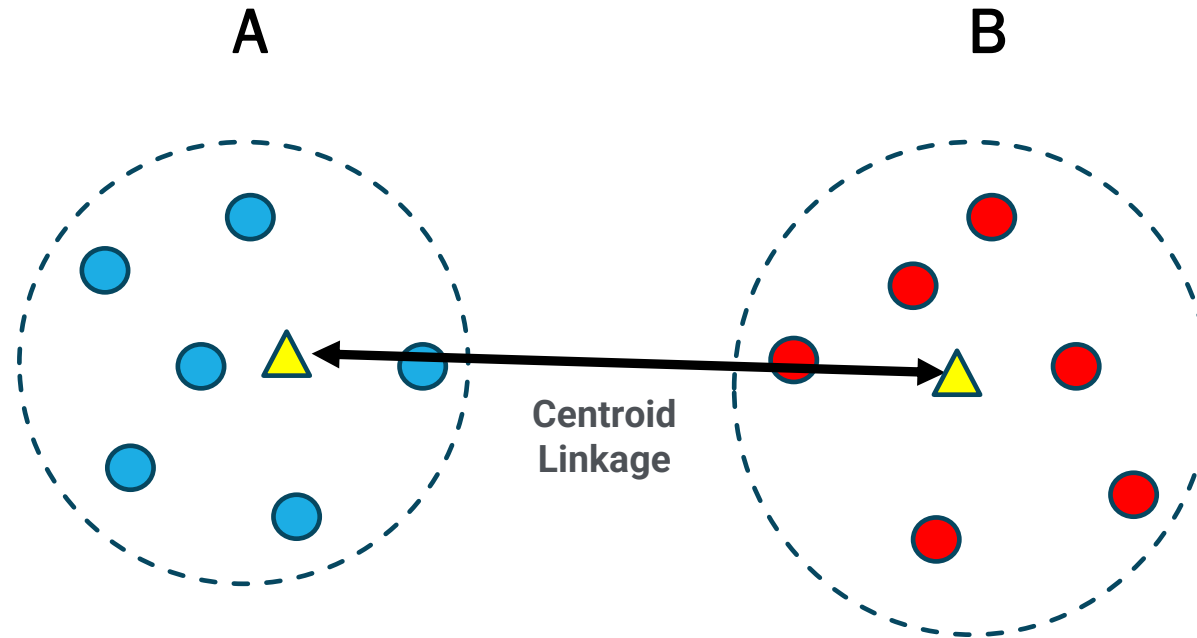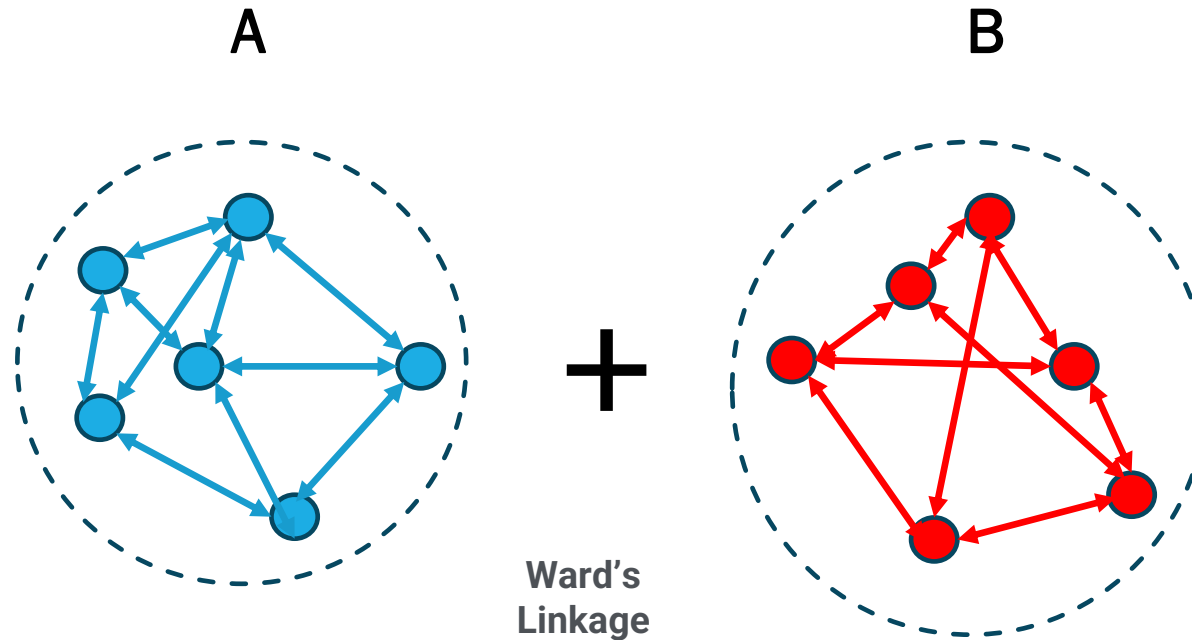
- Compute intra-cluster Sum-of-Squared Errors (ESS) for each cluster individually
- Combine clusters if combined ESS is less than sum of each individual cluster ESS
- Objective is to minimize the ESS of joint clusters

BYU

IMPACT OF DIFFERENT LINKAGE METHODS

Single Linkage · Average Linkage · Complete Linkage · Ward Linkage
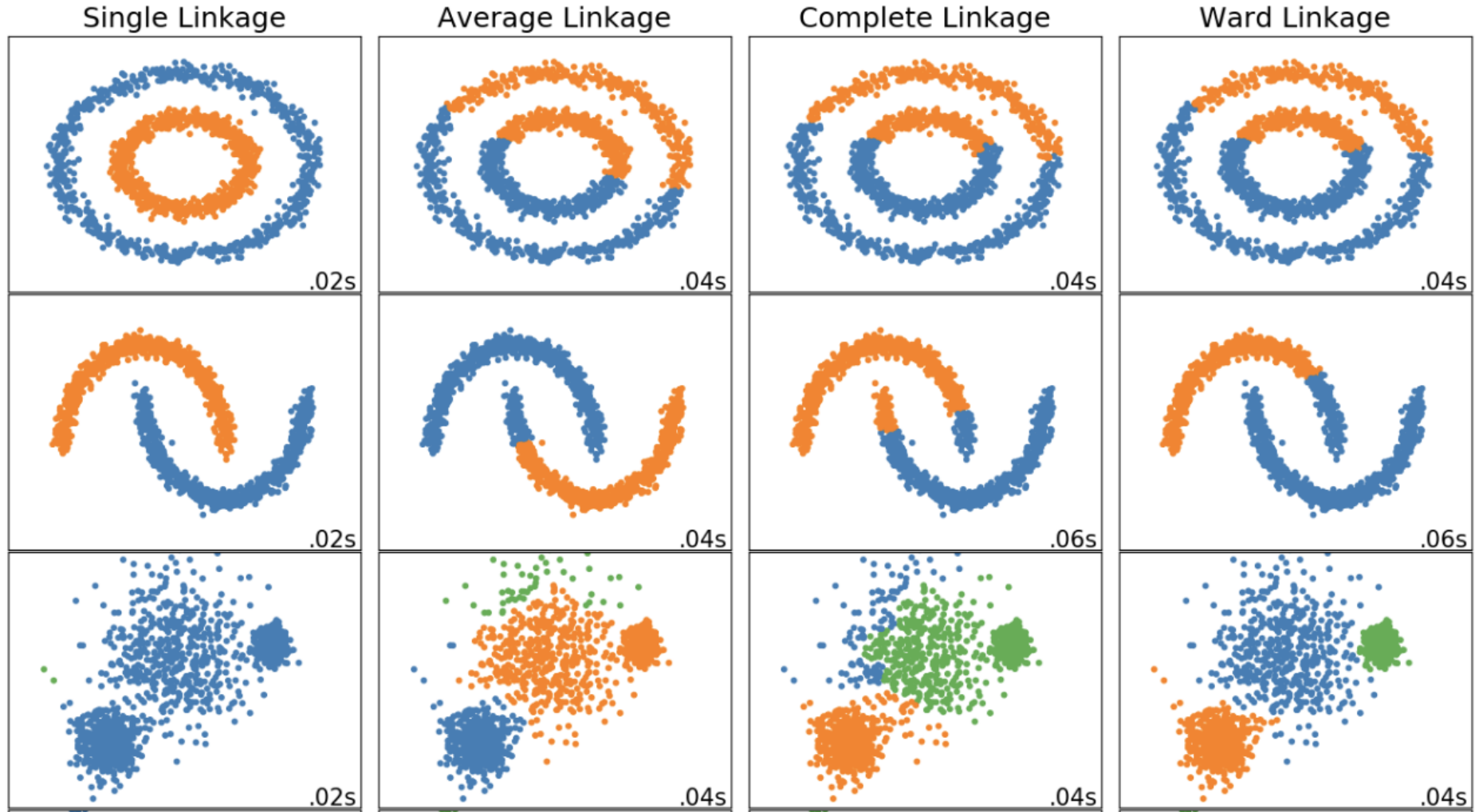
HIERARCHICAL AGGLOMERATIVE CLUSTERING EXAMPLE

Let's say a clothing store has collected the ages of 9 of its customers, labeled C1-C9, and the amount each spent at the store last month.

We start by making every single data point a cluster. This forms 9 clusters:

Take the two closest clusters and make them one cluster. Since C2 and C3 are closest, they form a cluster.

Take the next two closest clusters and make them one cluster. Since C4 and C5 are closest, they form a cluster.
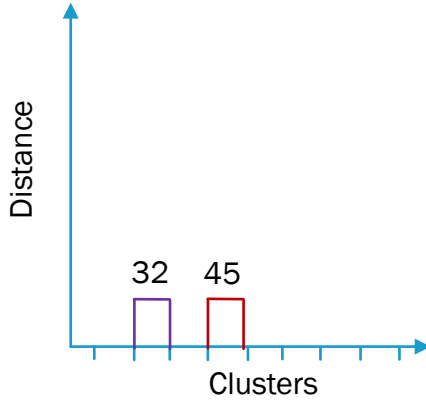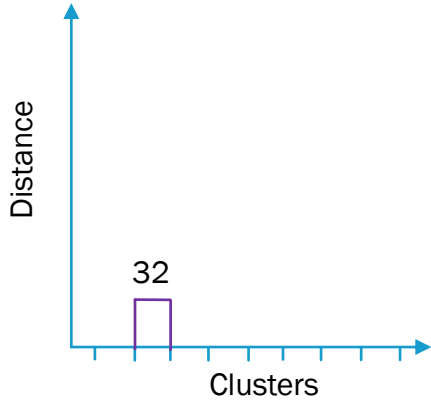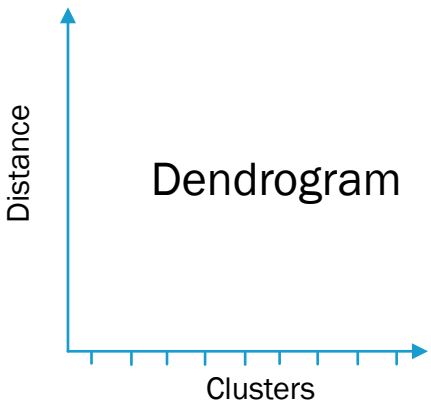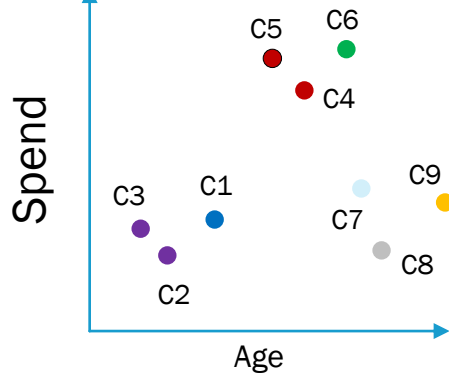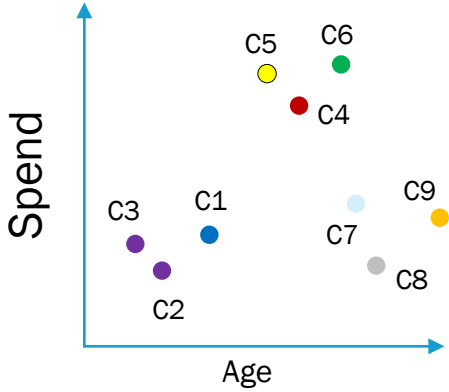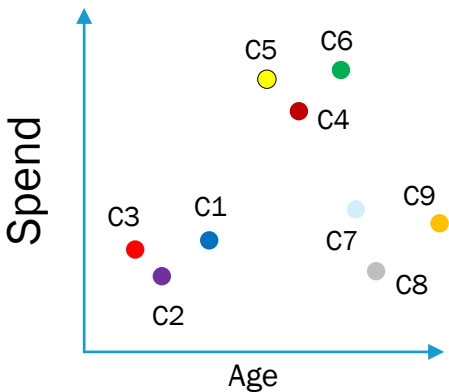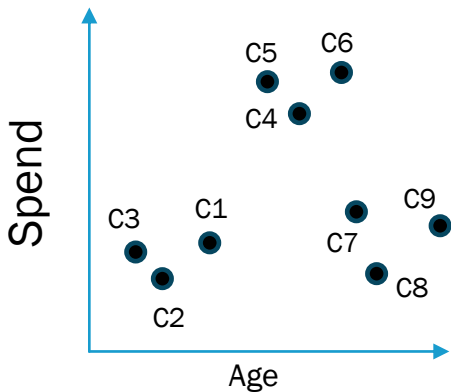
HAC EXAMPLE CONTINUED
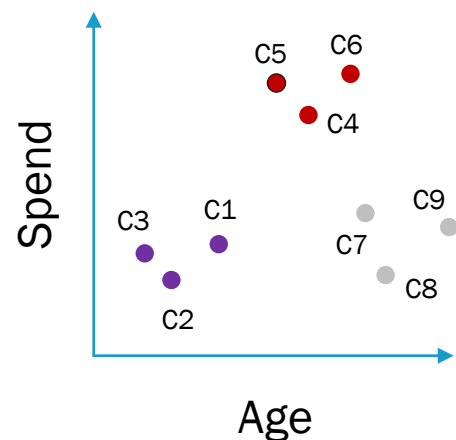
Based on Single Linkage, merge C45 with C6.
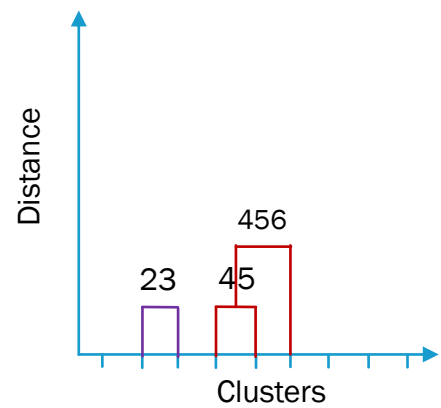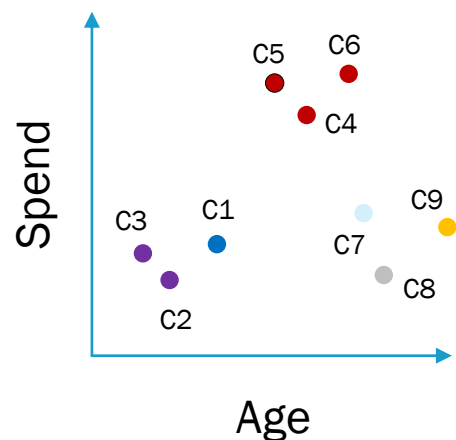
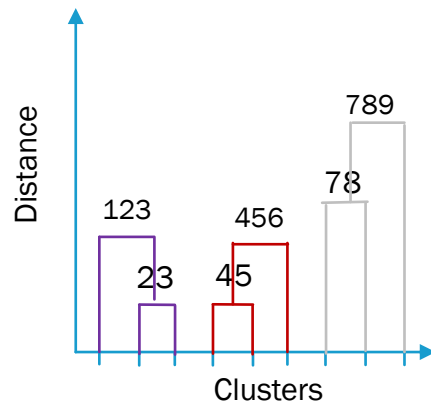Based on Single Linkage, merge C23 with C1
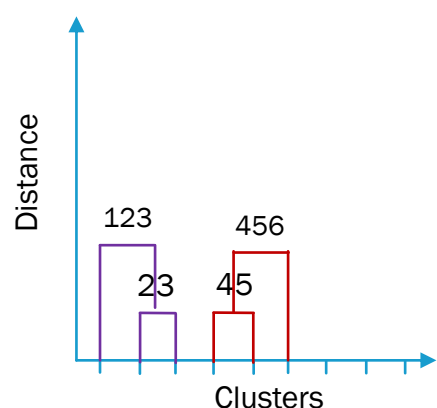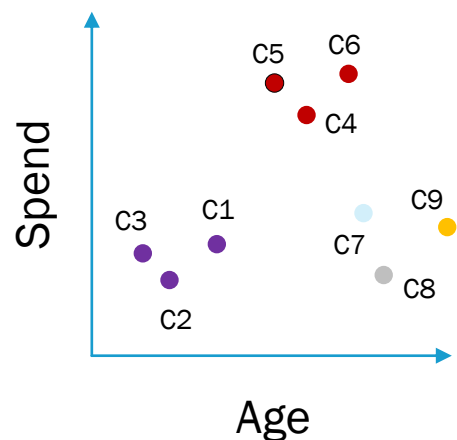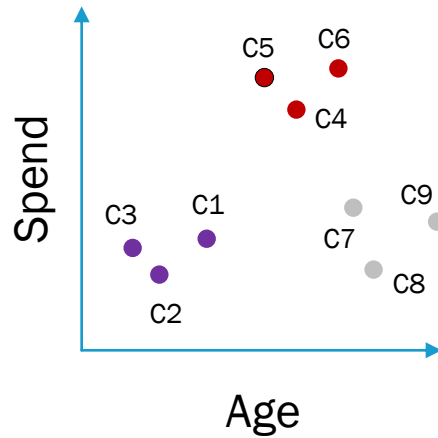
Based on distance, merge C7 and C8
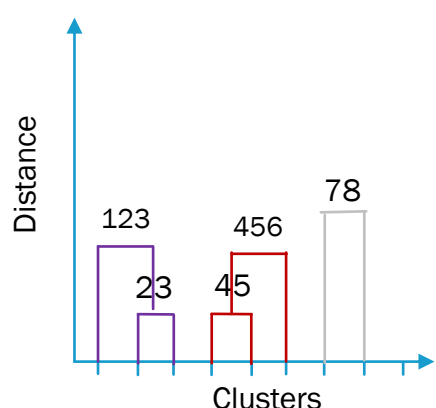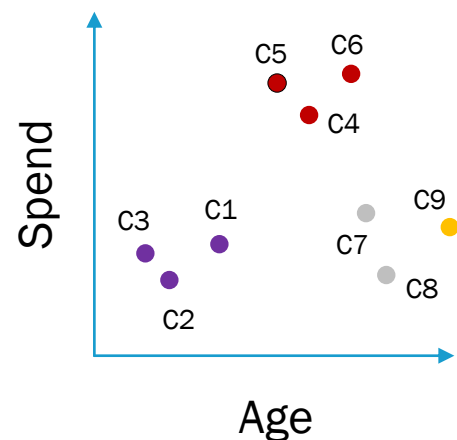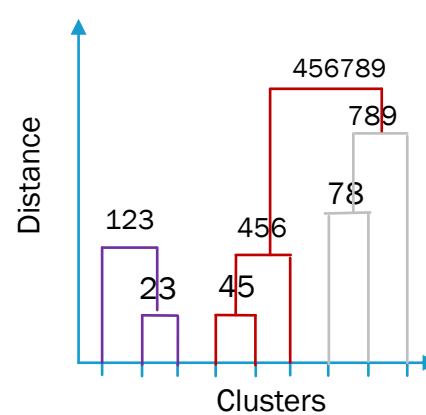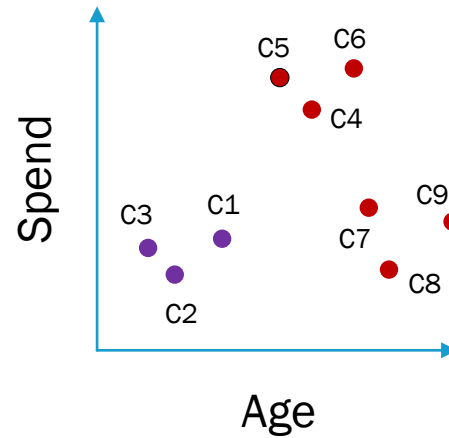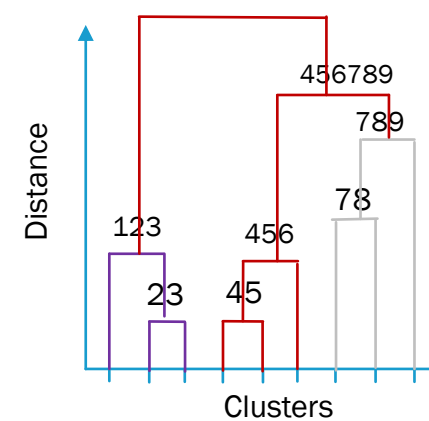
Merge C78 with C9
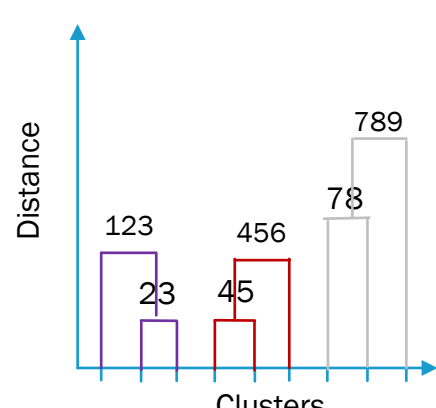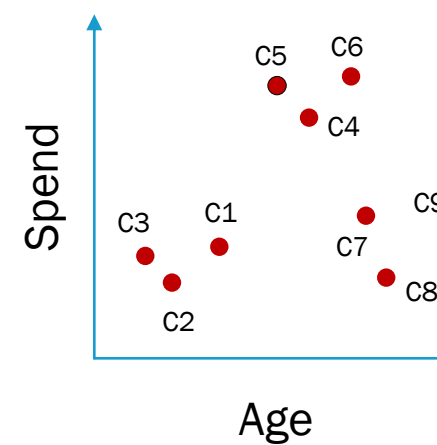
HAC EXAMPLE END

Starting from previous page

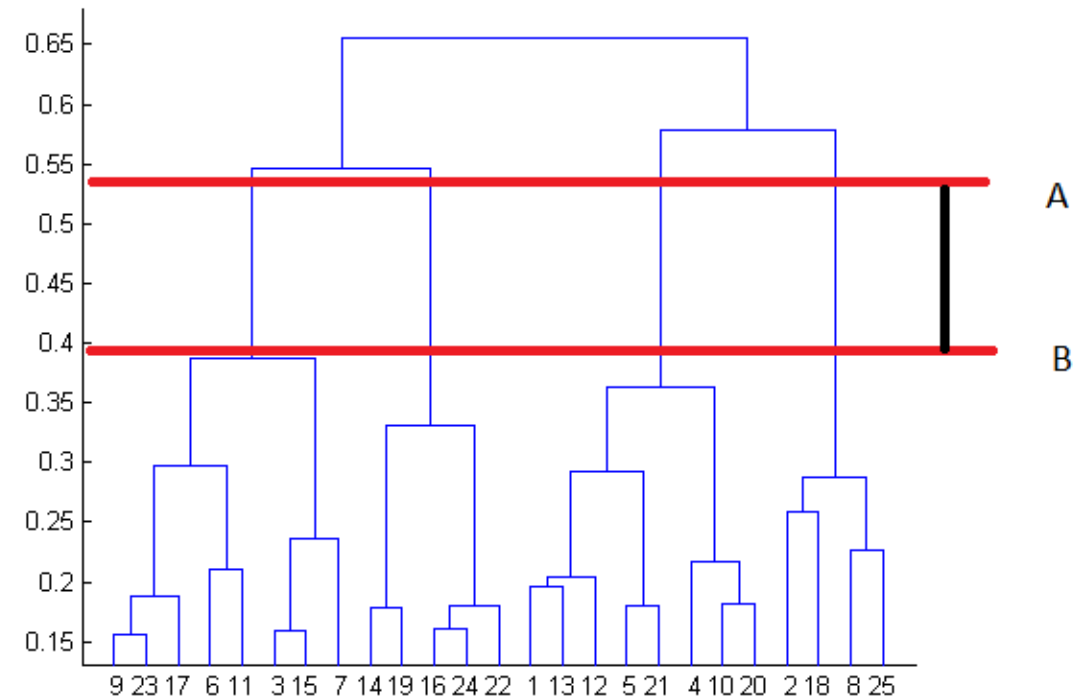Using Single Linkage, merge C456 with C789

Finally, merge C123 with C456789 to form one cluster.

USING DENDROGRAMS TO FIND OPTIMAL CLUSTER NUMBER

We can use a dendrogram to visualize the history of groupings and figure out the optimal number of clusters.

1. Determine the largest vertical distance that doesn't intersect any of the other clusters

2. Draw a horizontal line across the diagram

3. The optimal number of clusters is equal to the number of vertical lines going through the horizontal line

AGGLOMERATIVE CLUSTERING PYTHON EXAMPLE

REFERENCES

- https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68

- https://www.geeksforgeeks.org/ml-k-medoids-clustering-with-example/

- https://www.analyticsvidhya.com/blog/2021/06/kmodes-clustering-algorithm-for-categorical-data/#h-what-is-kmodes

- https://towardsdatascience.com/the-k-prototype-as-clustering-algorithm-for-mixed-data-type-categorical-and-numerical-fe7c50538ebb

- http://scikit-learn.org/stable/modules/clustering.html

- https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.15.4028&rep=rep1&type=pdf

- https://github.com/nicodv/kmodes/blob/master/kmodes/kprototypes.py

- https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/

- https://www.geeksforgeeks.org/difference-between-agglomerative-clustering-and-divisive-clustering/

- https://stackabuse.com/dbscan-with-scikit-learn-in-python/

BYU