# Hierarchical Clustering

계층적 군집분석

21.01.20

Seunghan Lee (이승한)

# Contents

# 1. What is Hierarchical Clustering (HC)?

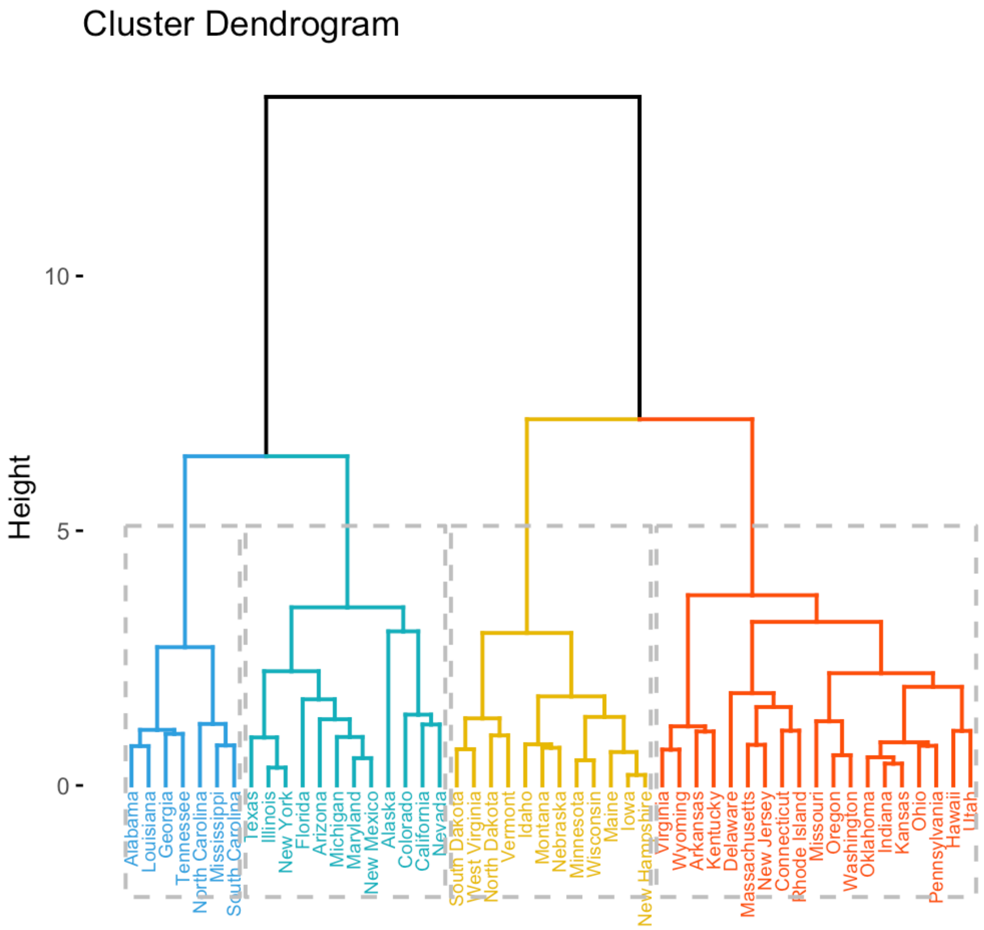
## **“계층적 군집 분석”**

생성되는 cluster들은 **“계층”**을 가지고 있다.

데이터들이 어느 단계(계층)에서 서로 묶이는지

(clustering이 되는지)를 확인할 수 있다.

( 생물학/고객군 분류에서 자주 사용 )




계층적 군집분석은 크게 2가지로 나뉜다

1) **Agglomerative**

2) **Divisive**

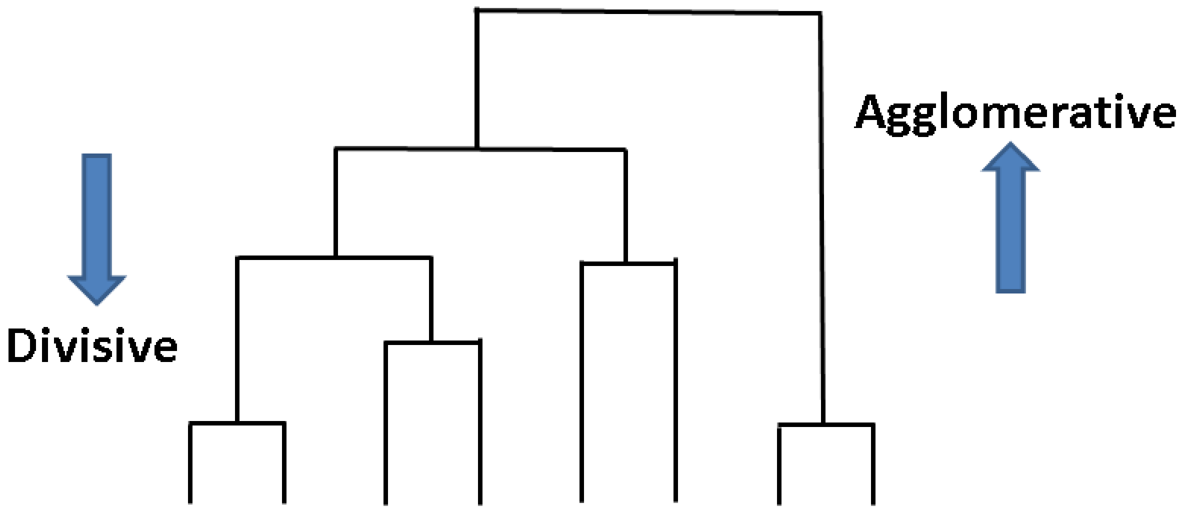
# 1. What is Hierarchical Clustering (HC)?

## “계층적 군집 분석”

계층적 군집분석은 크게 2가지로 나뉜다

1) **Agglomerative** : bottom-up approach

2) **Divisive** : top-down approach



# 1. What is Hierarchical Clustering (HC)?

1) **Agglomerative :** bottom-up approach

- 각각의 data가 하나의 cluster로써 시작을 한다

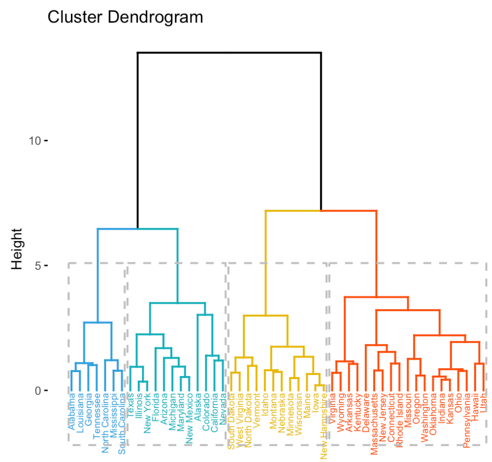
- (위로 올라가면서) 각각의 data (cluster)가 **서로 합쳐지면서(merge)** 더 큰 cluster를 이루어 나간다

2) **Divisive** : top-down approach

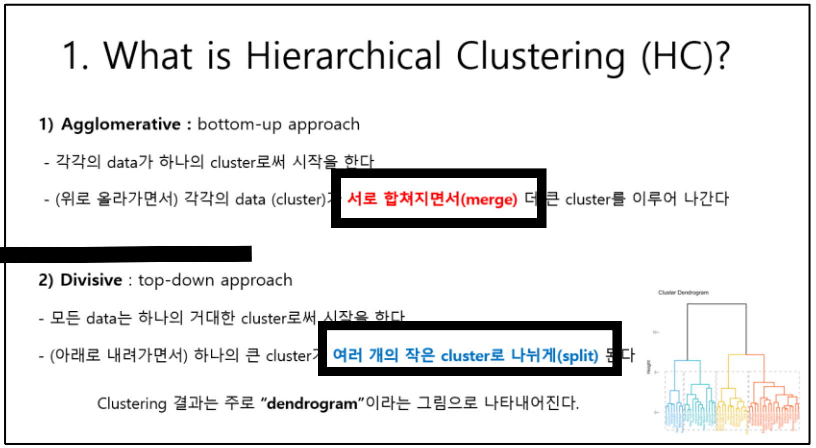
- 모든 data는 하나의 거대한 cluster로써 시작을 한다

- (아래로 내려가면서) 하나의 큰 cluster가 **여러 개의 작은 cluster로 나뉘게(split)** 된다

위 두 방법을 통한 Clustering 결과는 주로 **"dendrogram"**으로 나타내어진다.



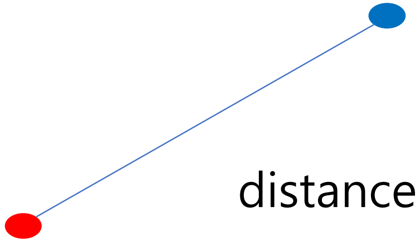
# 2. Clustering Dissimilarity




어떠한 기준으로 합쳐지고(merge), 나뉘게(split)되는가?

## (1) Distance metric

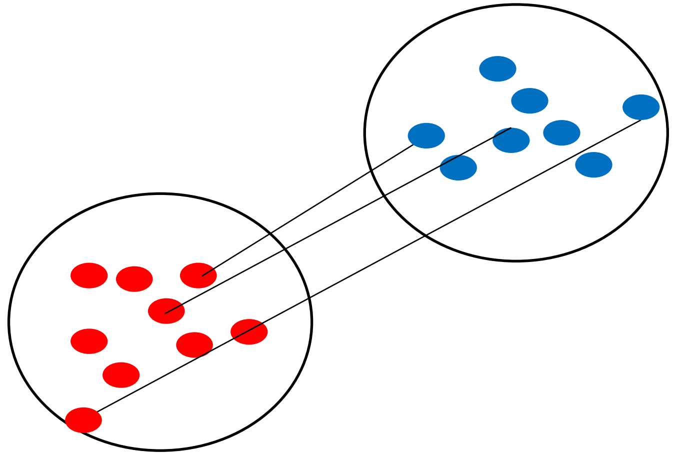
| Names | Formula |
|---|---|
| Euclidean distance | $\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$ |
| Squared Euclidean distance | $\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$ |
| Manhattan distance | $\|a-b\|_1 = \sum_i \lvert a_i - b_i \rvert$ |
| Maximum distance | $\|a-b\|_\infty = \max_i \lvert a_i - b_i \rvert$ |
| Mahalanobis distance | $\sqrt{(a-b)^\top S^{-1}(a-b)}$ where $S$ is the Covariance matrix |



# 2. Clustering Dissimilarity

## (2) Linkage criteria

| Names | Formula |
|---|---|
| Maximum or complete-linkage clustering | $\max\{\, d(a,b) : a \in A,\, b \in B \,\}.$ |
| Minimum or single-linkage clustering | $\min\{\, d(a,b) : a \in A,\, b \in B \,\}.$ |
| Unweighted average linkage clustering (or UPGMA) | $\frac{1}{\lvert A\rvert \cdot \lvert B\rvert} \sum_{a \in A} \sum_{b \in B} d(a,b).$ |
| Weighted average linkage clustering (or WPGMA) | $d(i \cup j, k) = \frac{d(i,k) + d(j,k)}{2}.$ |
| Centroid linkage clustering, or UPGMC | $\lVert c_s - c_t \rVert$ where $c_s$ and $c_t$ are the centroids of clusters $s$ and $t$, respectively. |
| Minimum energy clustering | $\frac{2}{nm} \sum_{i,j=1}^{n,m} \lVert a_i - b_j \rVert_2 - \frac{1}{n^2} \sum_{i,j=1}^{n} \lVert a_i - a_j \rVert_2 - \frac{1}{m^2} \sum_{i,j=1}^{m} \lVert b_i - b_j \rVert_2$ |
| Ward | Minimum Variance |



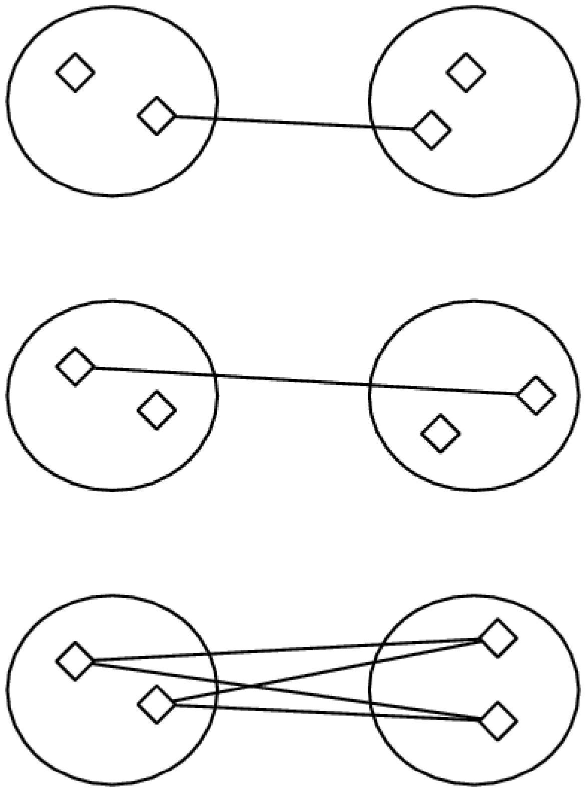
# 2. Clustering Dissimilarity

## (2) Linkage criteria

Single Linkage
( = minimum-linkage )

Complete Linkage
( = maximum-linkage )

Group Average Linkage



많이 사용되는 3가지 Linkage Critera

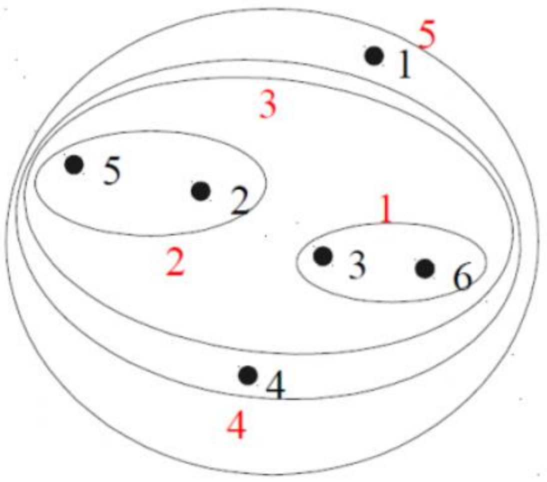
- 1) Single(Minimum) Linkage : **최단 연결법**
- 2) Complete (Maximum) Linkage : **최장 연결법**
- 3) Group Average : **평균 연결법**

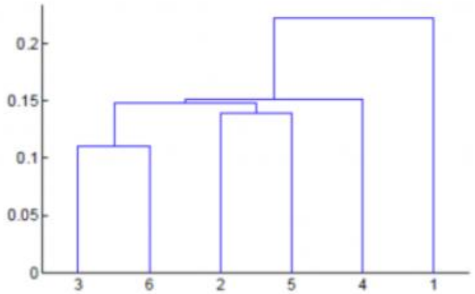
# 2. Clustering Dissimilarity

### (2) Linkage criteria

1) Single(Minimum) Linkage : **최단 연결법**

- 두 군집 간에 “가장 가까운 점 " 사이의 거리로서 구함



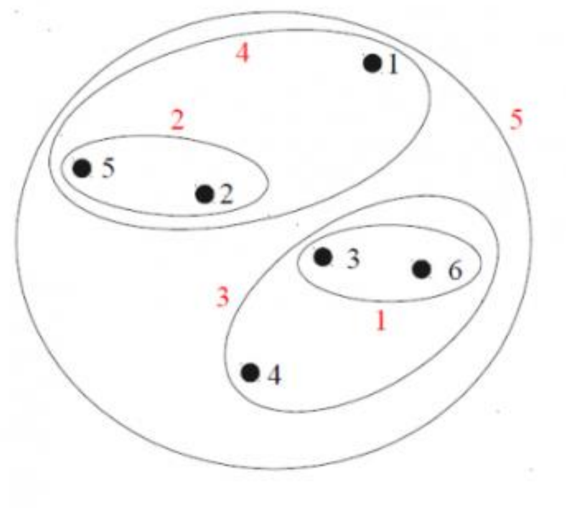



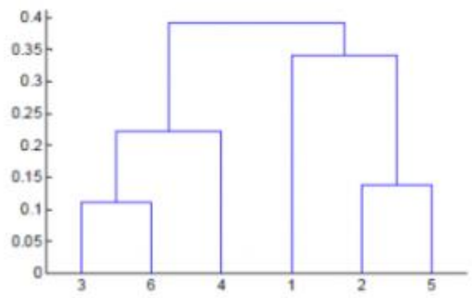
# 2. Clustering Dissimilarity

## (2) Linkage criteria

2) Complete (Maximum) Linkage : **최장 연결법**

- 두 군집 간에 “가장 먼 점 " 사이의 거리로서 구함



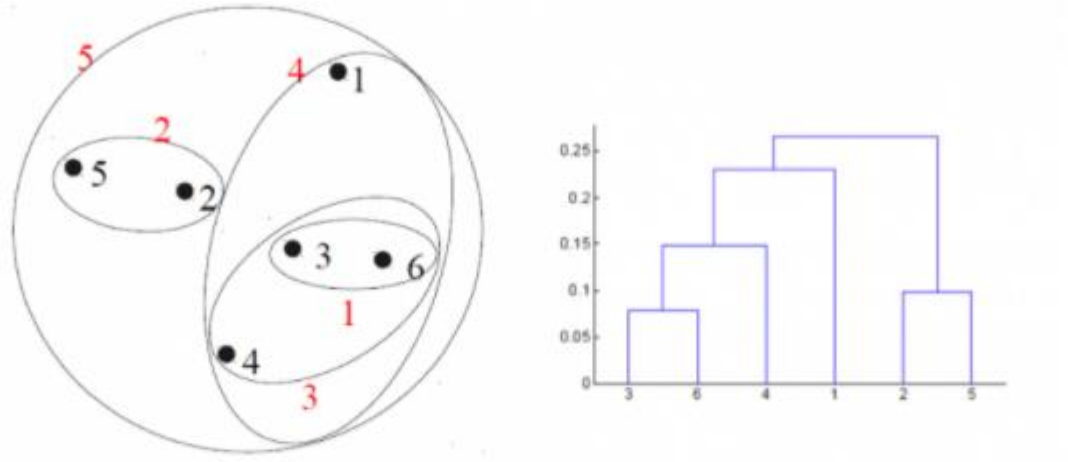



# 2. Clustering Dissimilarity

## (2) Linkage criteria

3) Group Average : **평균 연결법**

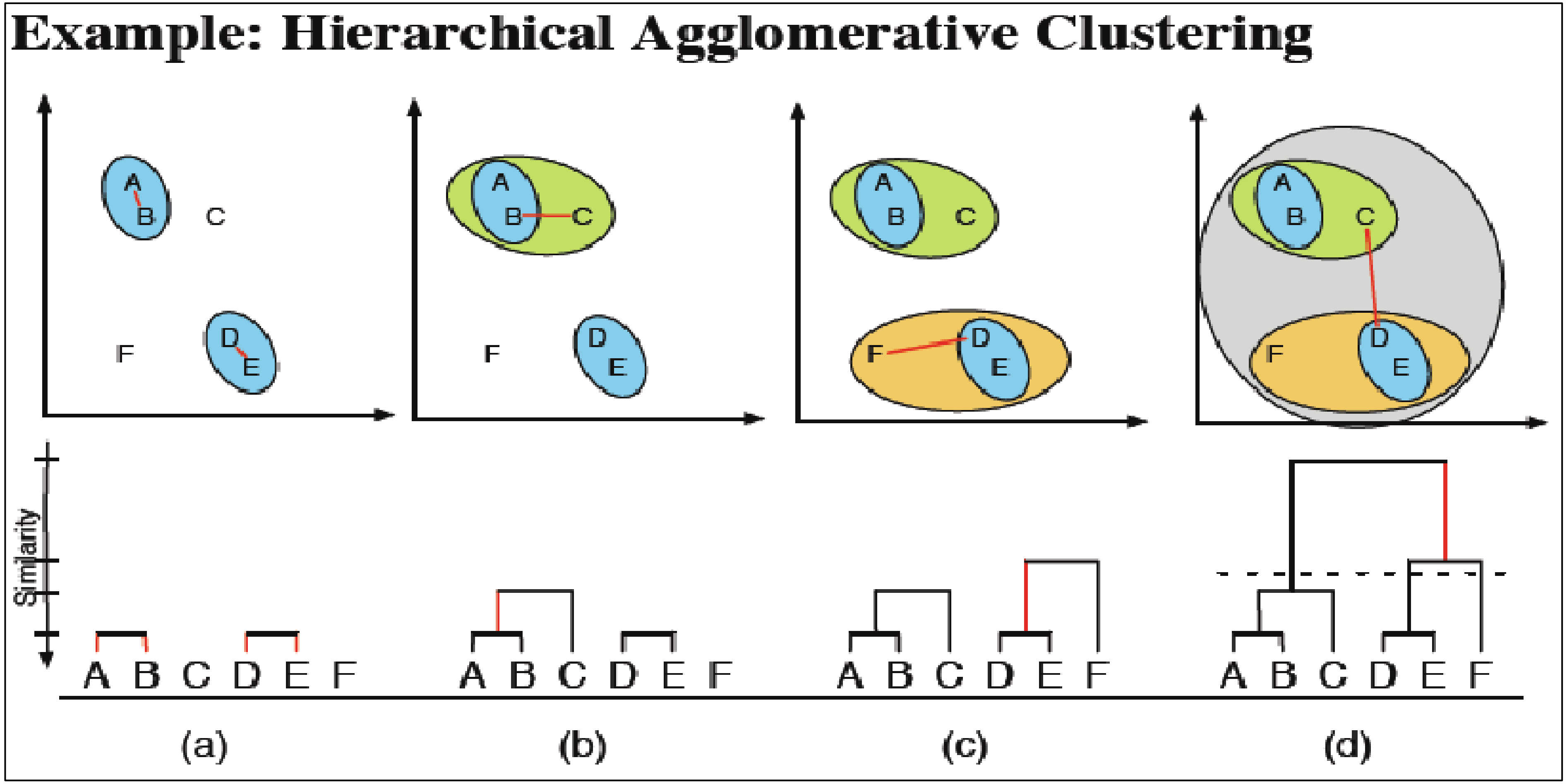
- 두 군집 간에 “점들 사이의 평균 거리 " 로서 구함




최단연결법 & 최장연결법의 trade-off 관계를 절충

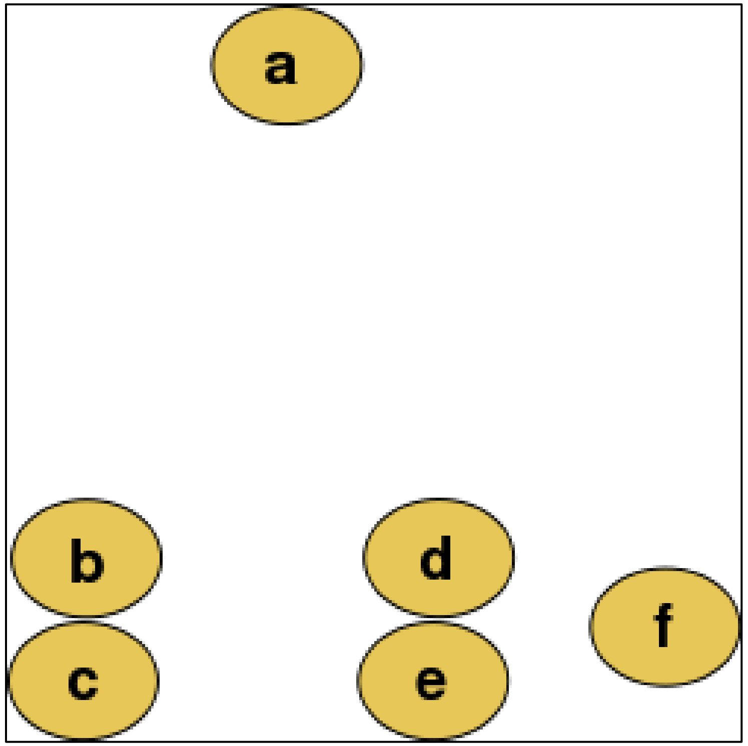
BUT 계산 비용이 높다는 단점!

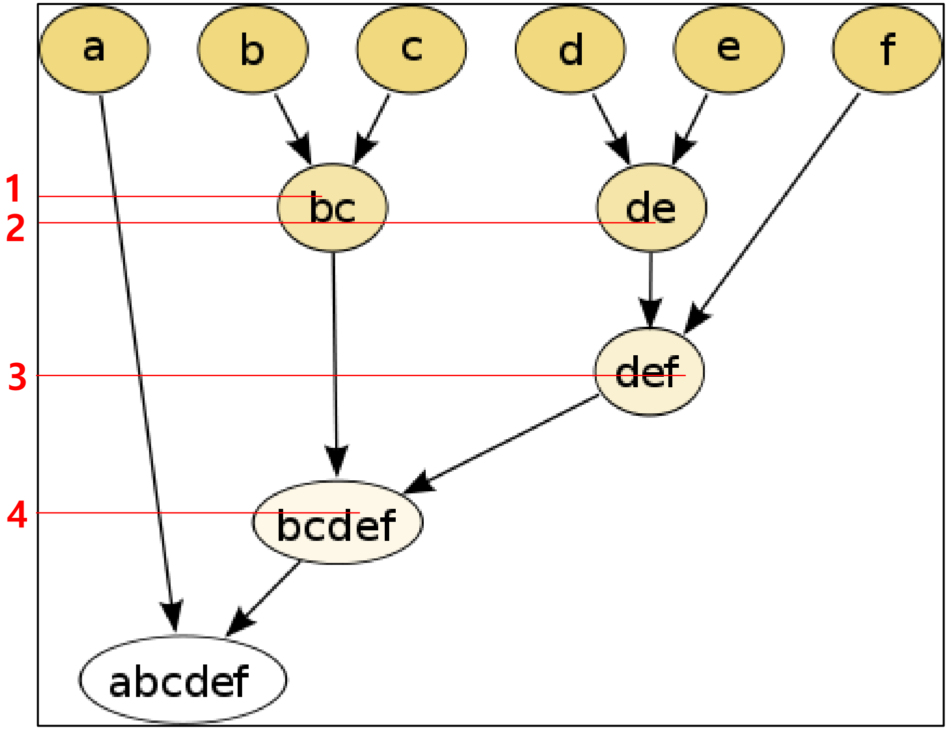
# 3. Agglomerative Clustering



# 3. Agglomerative Clustering

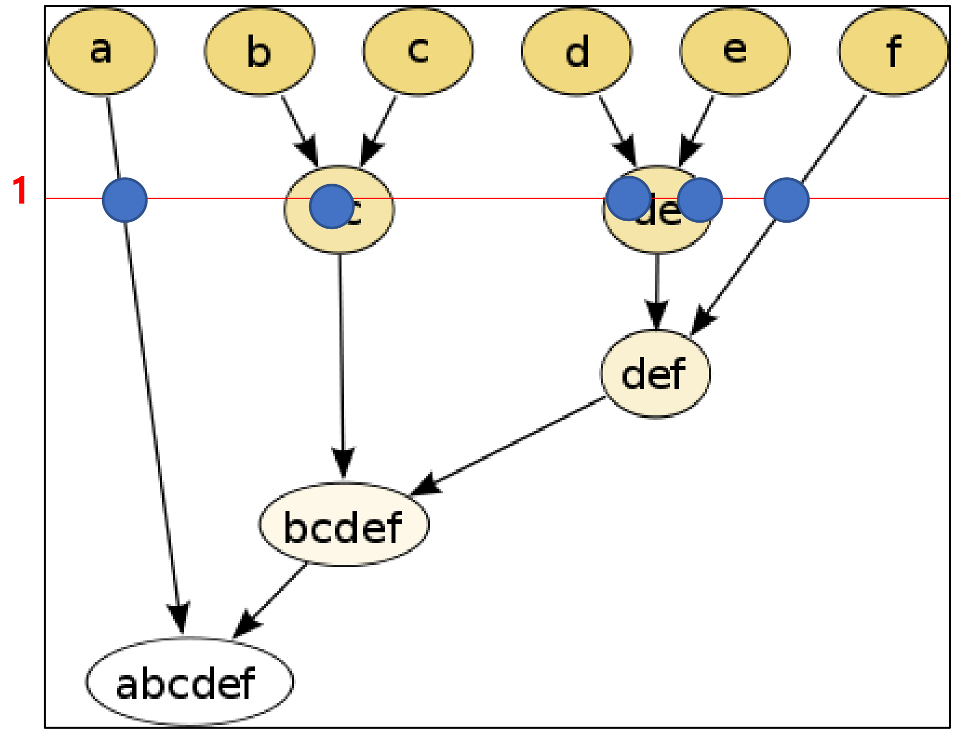
장점 : 눈으로 상황을 보아가며 **Cluster의 개수를 직접 정할 수 있다!**



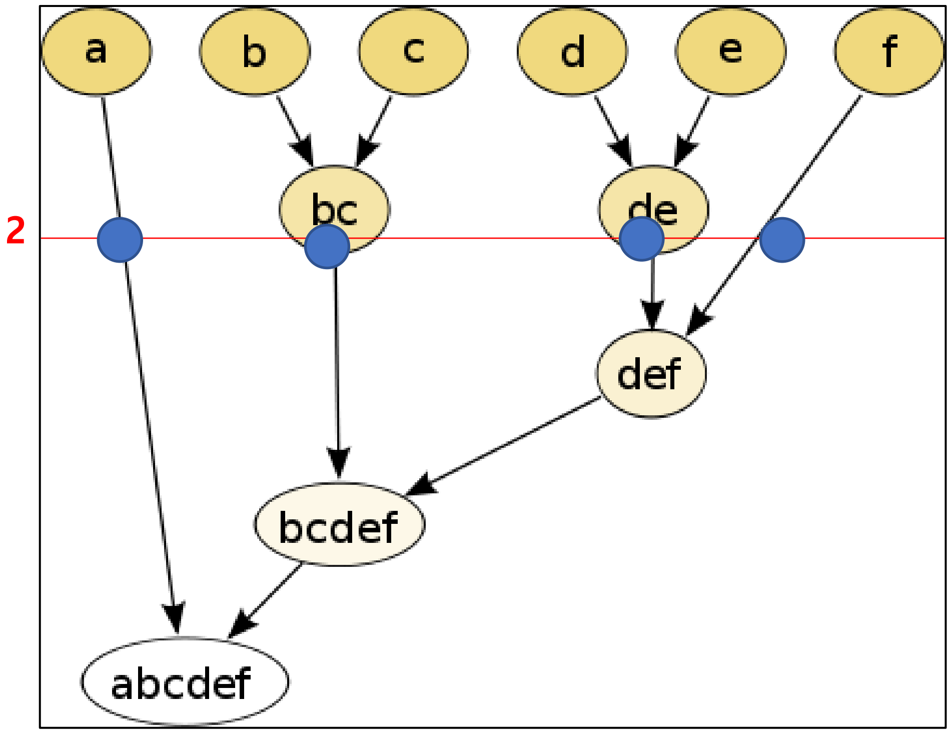



# 3. Agglomerative Clustering

장점 : 눈으로 상황을 보아가며 **Cluster의 개수를 직접 정할 수 있다!**



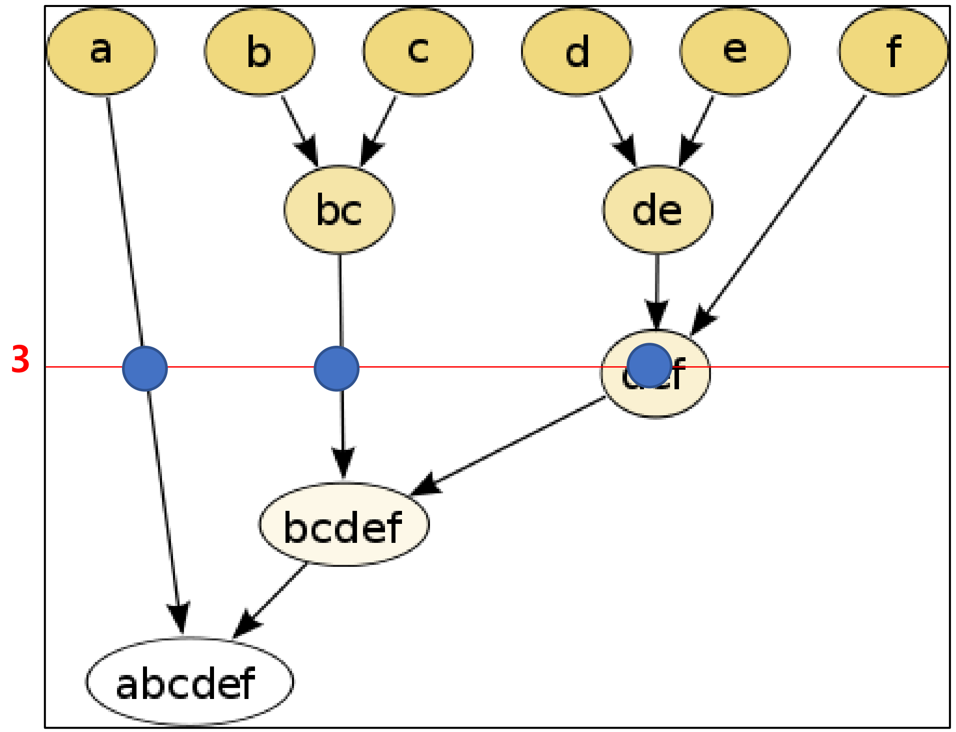

**Number of clusters = 5**



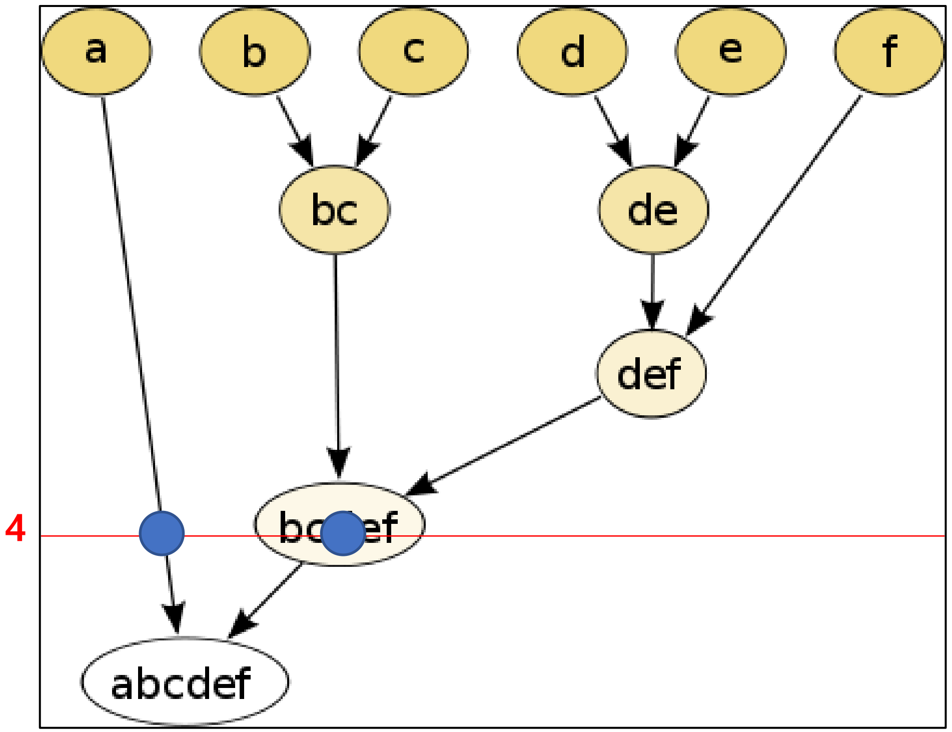

**Number of clusters = 4**

# 3. Agglomerative Clustering

장점 : 눈으로 상황을 보아가며 **Cluster의 개수를 직접 정할 수 있다!**



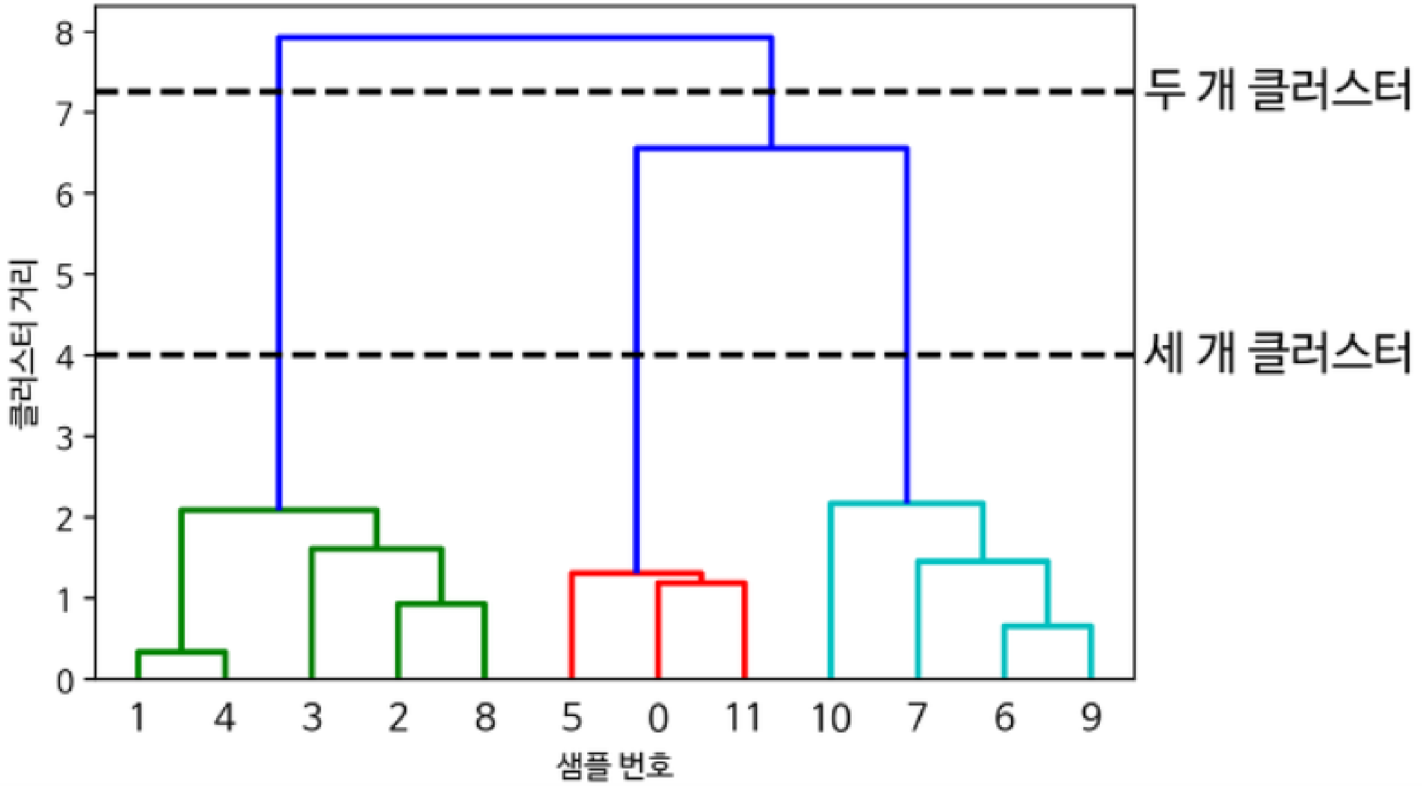

Number of clusters = 3




Number of clusters = 2

# 3. Agglomerative Clustering

장점 : 눈으로 상황을 보아가며 **Cluster의 개수를 직접 정할 수 있다!**



# 4. Problems with Hierarchical Clustering

**Greedy algorithm (탐욕 알고리즘)**

- 미래 생각 X...ONLY 현재 단계에서 최선의 것을 선택하는 “이기적 " 인 기법!

**높은 연산량**

- 각각의 data 사이의 distance matrx를 계산해야 하기 때문에, 연산량이 높다.

  ( ex. 1만개의 data : 1만x1만 = 1억번 계산 )

- Time complexity : O(n^3)

# 5. Python Code for Hierarchical Clustering

Package

- (1) Sklearn : for clustering
- (2) Scipy : for dendrogram

Sklearn은 dendrogram 시각화 기능을 제공하지 않음

따라서 Scipy 패키지를 사용!

# 5. Python Code for Hierarchical Clustering

## (1) sklearn

### sklearn.cluster.AgglomerativeClustering

*class* sklearn.cluster.**AgglomerativeClustering**(*n_clusters=2, *, affinity='euclidean', memory=None, connectivity=None, compute_full_tree='auto', linkage='ward', distance_threshold=None, compute_distances=False*) [source]

지정해줘야할 핵심 변수 3가지

- 1) n_cluster : 클러스터의 개수
- 2) **affinity** : distance metric
- 3) **linkage** : linkage criterion

# 5. Python Code for Hierarchical Clustering

## (1) sklearn

```
agg = AgglomerativeClustering(n_clusters=3,linkage='ward')
assignment = agg.fit_predict(pca_df_temp)
```

( 거리지표는 주로 "Euclidean"을 사용 )

**linkage : *{'ward', 'complete', 'average', 'single'}, default='ward'***

Which linkage criterion to use. The linkage criterion determines which distance to use between sets of observation. The algorithm will merge the pairs of cluster that minimize this criterion.

- 'ward' minimizes the variance of the clusters being merged.
- 'average' uses the average of the distances of each observation of the two sets.
- 'complete' or 'maximum' linkage uses the maximum distances between all observations of the two sets.
- 'single' uses the minimum of the distances between all observations of the two sets.

# 5. Python Code for Hierarchical Clustering

## (2) scipy

```
linkage_array = linkage(pca_df_temp,'ward')
dendrogram(linkage_array)
```

- method='single' assigns

$$d(u,v) = \min(dist(u[i], v[j]))$$

for all points $i$ in cluster $u$ and $j$ in cluster $v$. This is also known as the Nearest Point Algorithm.

- method='complete' assigns

$$d(u,v) = \max(dist(u[i], v[j]))$$

for all points $i$ in cluster u and $j$ in cluster $v$. This is also known by the Farthest Point Algorithm or Voor Hees Algorithm.

- method='average' assigns

$$d(u,v) = \sum_{ij} \frac{d(u[i], v[j])}{(|u| * |v|)}$$

for all points $i$ and $j$ where $|u|$ and $|v|$ are the cardinalities of clusters $u$ and $v$, respectively. This is also called the UPGMA algorithm.

- method='weighted' assigns

$$d(u,v) = (dist(s,v) + dist(t,v))/2$$

where cluster u was formed with cluster s and t and v is a remaining cluster in the forest (also called WPGMA).

- method='centroid' assigns

$$dist(s,t) = ||c_s - c_t||_2$$

where $c_s$ and $c_t$ are the centroids of clusters $s$ and $t$, respectively. When two clusters $s$ and $t$ are combined into a new cluster $u$, the new centroid is computed over all the original objects in clusters $s$ and $t$. The distance then becomes the Euclidean distance between the centroid of $u$ and the centroid of a remaining cluster $v$ in the forest. This is also known as the UPGMC algorithm.

- method='median' assigns $d(s,t)$ like the `centroid` method. When two clusters $s$ and $t$ are combined into a new cluster $u$, the average of centroids s and t give the new centroid $u$. This is also known as the WPGMC algorithm.
- method='ward' uses the Ward variance minimization algorithm. The new entry $d(u,v)$ is computed as follows,

$$d(u,v) = \sqrt{\frac{|v|+|s|}{T} d(v,s)^2 + \frac{|v|+|t|}{T} d(v,t)^2 - \frac{|v|}{T} d(s,t)^2}$$

where $u$ is the newly joined cluster consisting of clusters $s$ and $t$, $v$ is an unused cluster in the forest, $T = |v| + |s| + |t|$, and $|*|$ is the cardinality of its argument. This is also known as the incremental algorithm.

# 참고 자료

참고 자료

https://en.wikipedia.org/wiki/Hierarchical_clustering#cite_note-15

https://lucy-the-marketer.kr/ko/growth/hierarchical-clustering/

https://www.zerocho.com/category/Algorithm/post/584ba5c9580277001862f188

https://www.youtube.com/watch?v=7xHsRkOdVwo

# Thank You