Hierarchical Clustering …

Hierarchical Clustering

- ***Hierarchical clustering***, also known as *hierarchical cluster analysis*, is an algorithm that groups similar objects into groups called *clusters*.

- The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

- Hierarchical clustering can be performed with either a *distance matrix* or *raw data.*

- When raw data is provided, the software will automatically compute a distance matrix in the background.

Hierarchical Clustering Working

- Hierarchical clustering starts by treating each observation as a separate cluster.

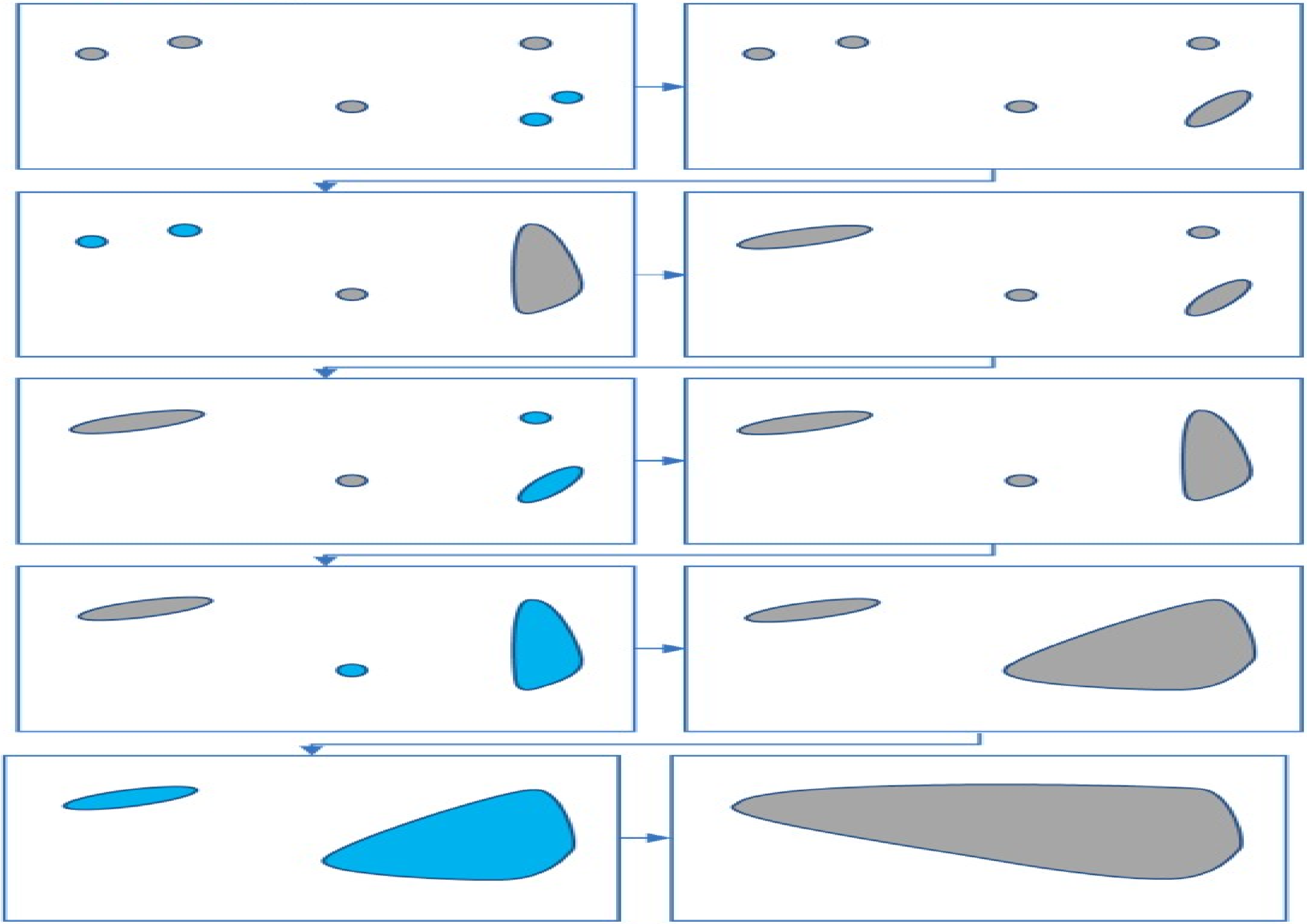
- Then, it repeatedly executes the following two steps:
  (1) identify the two clusters that are closest together
  (2) merge the two most similar clusters.

- This iterative process continues until all the clusters are merged together.

Example

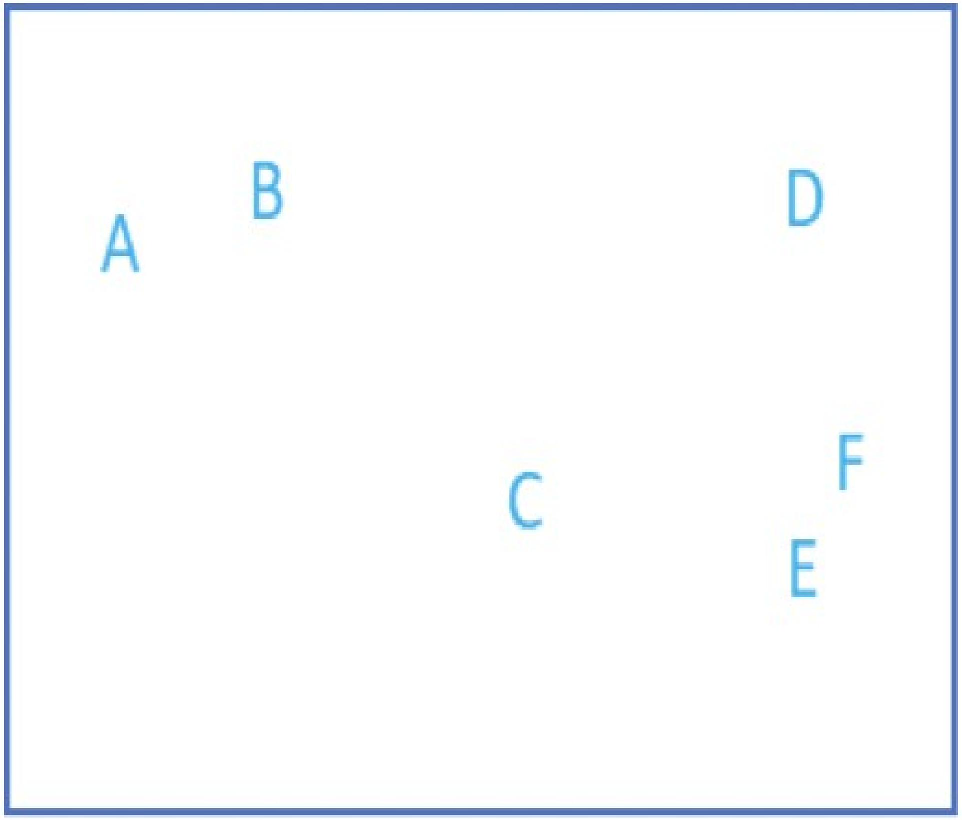
Identify the two clusters that are closest together

Merge the two most similar clusters



Final Output

- The final output of Hierarchical Clustering is a ***dendrogram***, which shows the hierarchical relationship between the clusters.



Measures of distance (similarity)

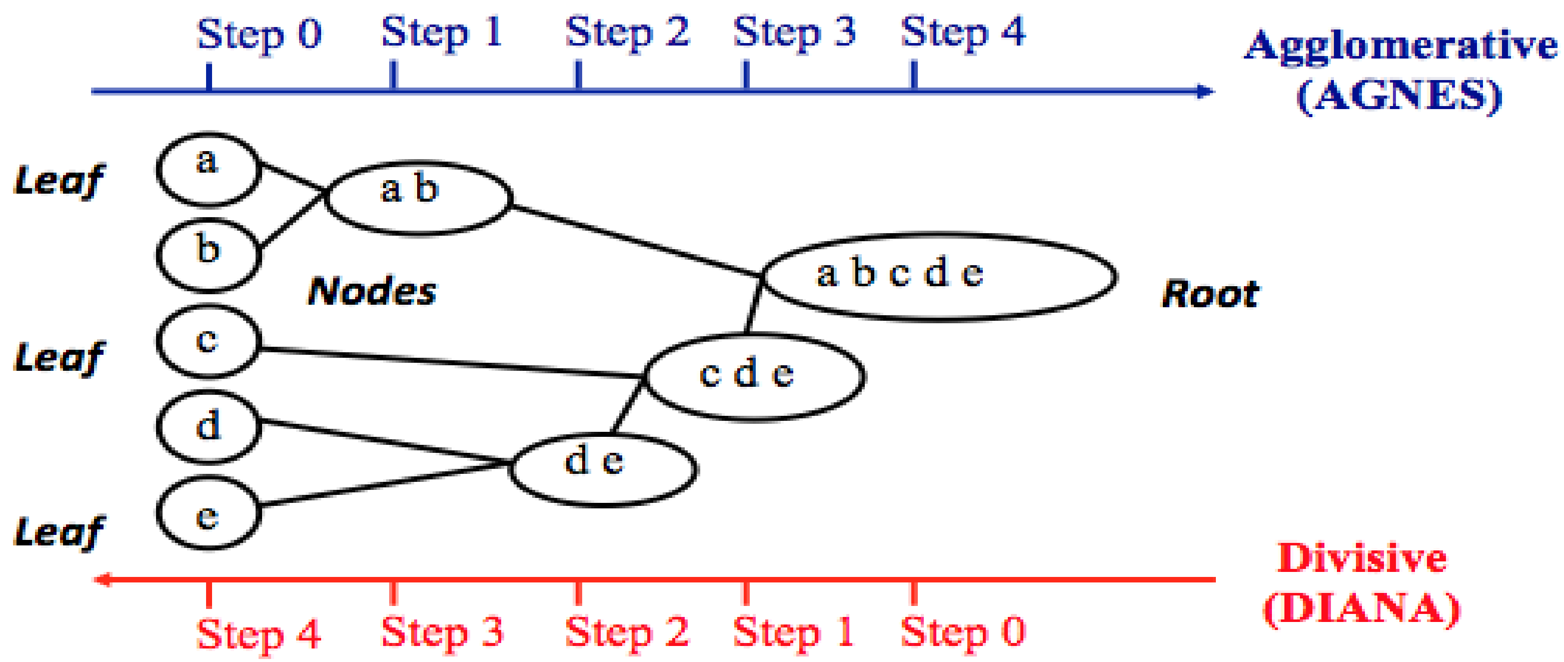
- The ***distance*** between two clusters has been computed based on the length of the straight line drawn from one cluster to another.
- This is commonly referred to as the *Euclidean distance.*
- Many other *distance metrics* have been developed.

- **Euclidean distance formula :**

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$

Types of Hierarchical Clustering

- There are two different types:
  - Agglomerative Clustering (Bottom – up)
  - Divisive Clustering (Top – Down)

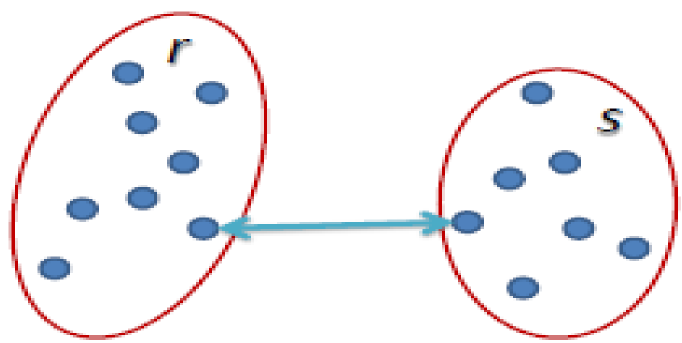


Types of Methods

- There are three different types:
  - Single Linkage
  - Complete Linkage
  - Average Linkage

1. Single Linkage

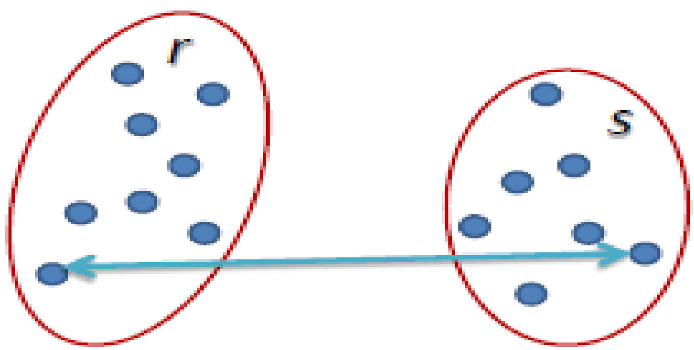
- In single linkage hierarchical clustering, the distance between two clusters is defined as the **shortest distance** between two points in each cluster.
- For example, the distance between clusters "r" and "s" to the left is equal to the length of the arrow between their two closest points.



$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$

2. Complete Linkage

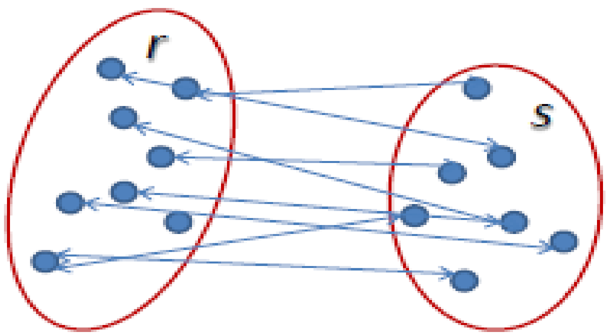
- In complete linkage hierarchical clustering, the distance between two clusters is defined as the **longest distance** between two points in each cluster.
- For example, the distance between clusters "r" and "s" to the left is equal to the length of the arrow between their two furthest points.



$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

2. Average Linkage

- In average linkage hierarchical clustering, the distance between two clusters is defined as the **average distance** between each point in one cluster to every point in the other cluster.
- For example, the distance between clusters "r" and "s" to the left is equal to the average length each arrow between connecting the points of one cluster to the other.



$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

DIFFERENCES

Single Linkage	This is the distance between the closest members of the two clusters.
Complete Linkage	This is the distance between the members that are farthest apart.
Average Linkage	This method involves looking at the distances between all pairs and averages all of these distances. This is also called Unweighted Pair Group Mean Averaging.

EXAMPLE

Find the clusters using single link technique. Use Euclidean distance, and draw the dendrogram.

	X	Y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

- Calculate Euclidean distance, create the distance matrix.

$$\text{Distance } [(x,y),(a,b)] = \sqrt{(x-a)^2+(y-b)^2}$$

$$\text{Distance (P1,P2)} = \sqrt{(0.40-0.22)^2+(0.53-0.38)^2}$$

$$(0.40,0.53),(0.22,0.38) = \sqrt{(0.18)^2+(0.15)^2}$$

$$= \sqrt{0.0324+0.0225}$$

$$= \sqrt{0.0549}$$

$$= 0.23$$

Here upper diagonal and lower diagonal have the same value.

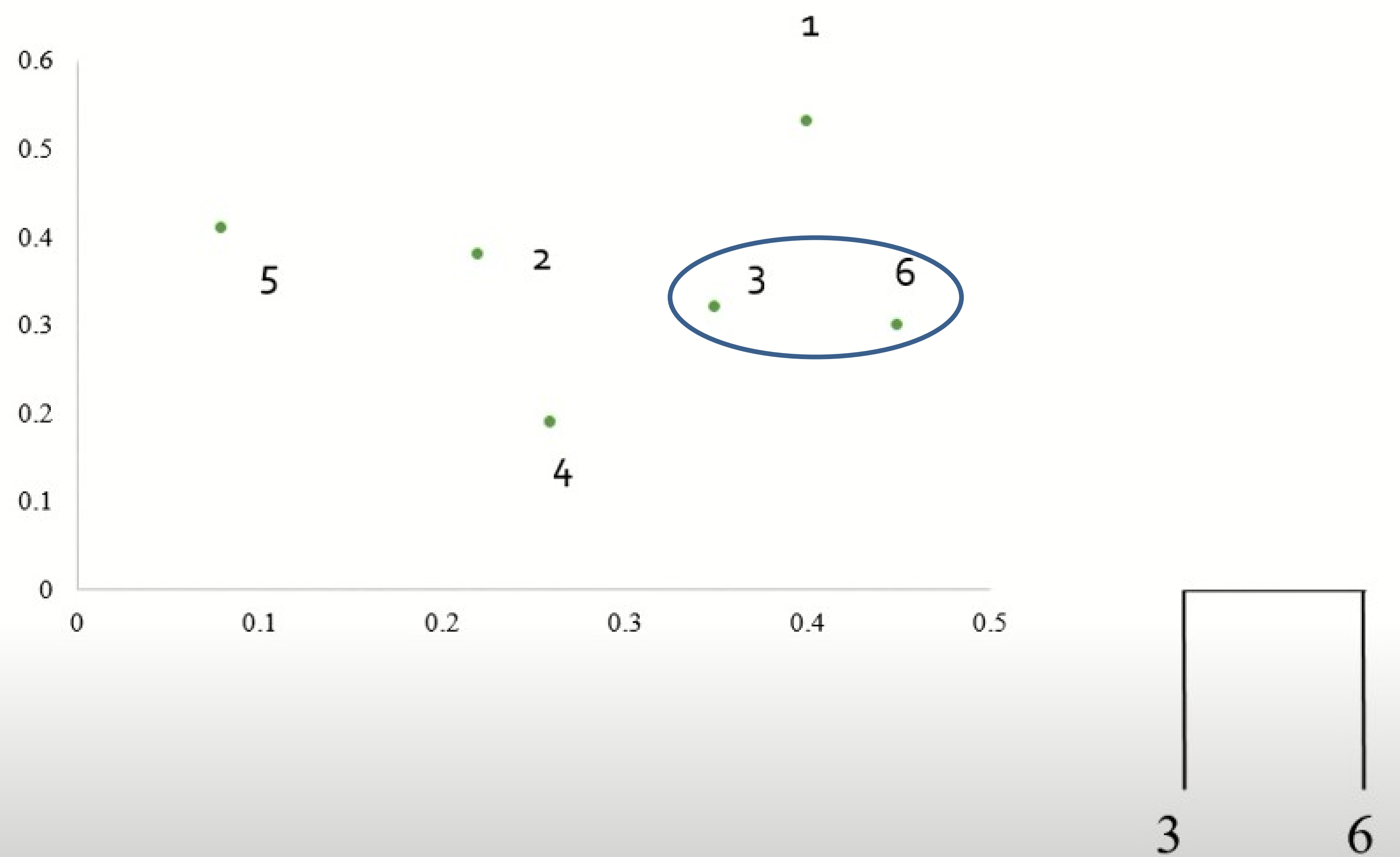
The distance matrix is

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

Choose the minimum value in distance matrix

The distance matrix is

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.24	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0


0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
1
2
3
4
5
6
3
6

To update the distance matrix MIN[dist(P3,P6),P1)]

MIN(dist(P3,P1), (P6,P1))

= min[(0.22,0.23)]

= 0.22

To update the distance matrix MIN[dist(P3,P6),P2)]

MIN(dist(P3,P2), (P6,P2))

= min[(0.15,0.25)]

= 0.15

The distance matrix is

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.24	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

To update the distance matrix MIN[dist(P3,P6),P4)]

MIN(dist(P3,P4), (P6,P4))

= min[(0.15,0.22)]

= 0.15

To update the distance matrix MIN[dist(P3,P6),P5)]

MIN(dist(P3,P5), (P6,P5))

= min[(0.28,0.39)]

= 0.28

The distance matrix is

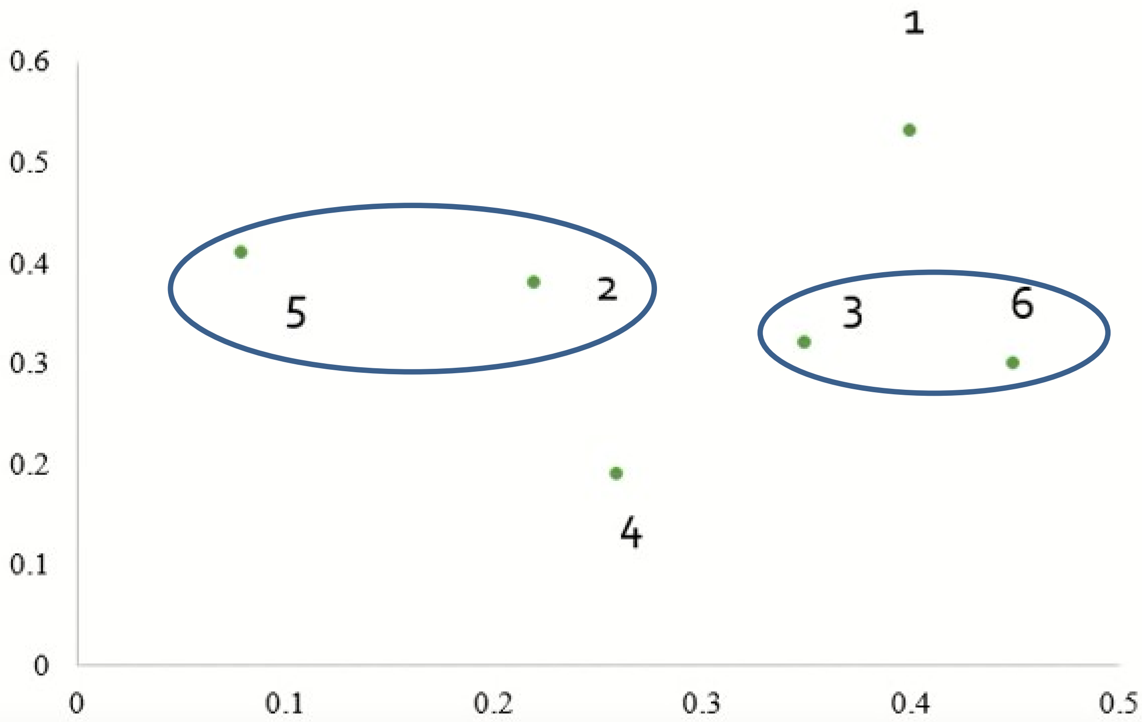
	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.24	0				
P3	0.22	0.15	0			
P4	0.37	0.20	0.15	0		
P5	0.34	0.14	0.28	0.29	0	
P6	0.23	0.25	0.11	0.22	0.39	0

The updated distance matrix for cluster P3, P6

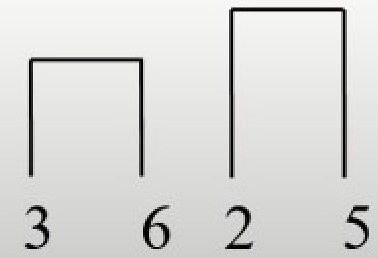
	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

Choose the minimum value from the distance matrix

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0



1
0.6
0.5
0.4
0.3
0.2
0.1
0
5
2
3
6
4
0
0.1
0.2
0.3
0.4
0.5



3
6
2
5

To update the distance matrix MIN[dist(P2,P5),P1)]

MIN[dist(P2,P1), (P5,P1)]

= min[(0.23,0.34)]

= 0.23

To update the distance matrix MIN[dist(P2,P5),(P3,P6)]

MIN[dist(P2,(P3,P6)), (P5,(P3,P6))]

= min[(0.15,0.28)]

= 0.15

	P1	P2	P3,P6	P4	P5
P1	0				
P2	0.23	0			
P3,P6	0.22	0.15	0		
P4	0.37	0.20	0.15	0	
P5	0.34	0.14	0.28	0.29	0

To update the distance matrix MIN[dist(P2,P5),P4)]

MIN[dist(P2,P4), (P5,P4)]

= min[(0.20,0.29)]

= 0.20

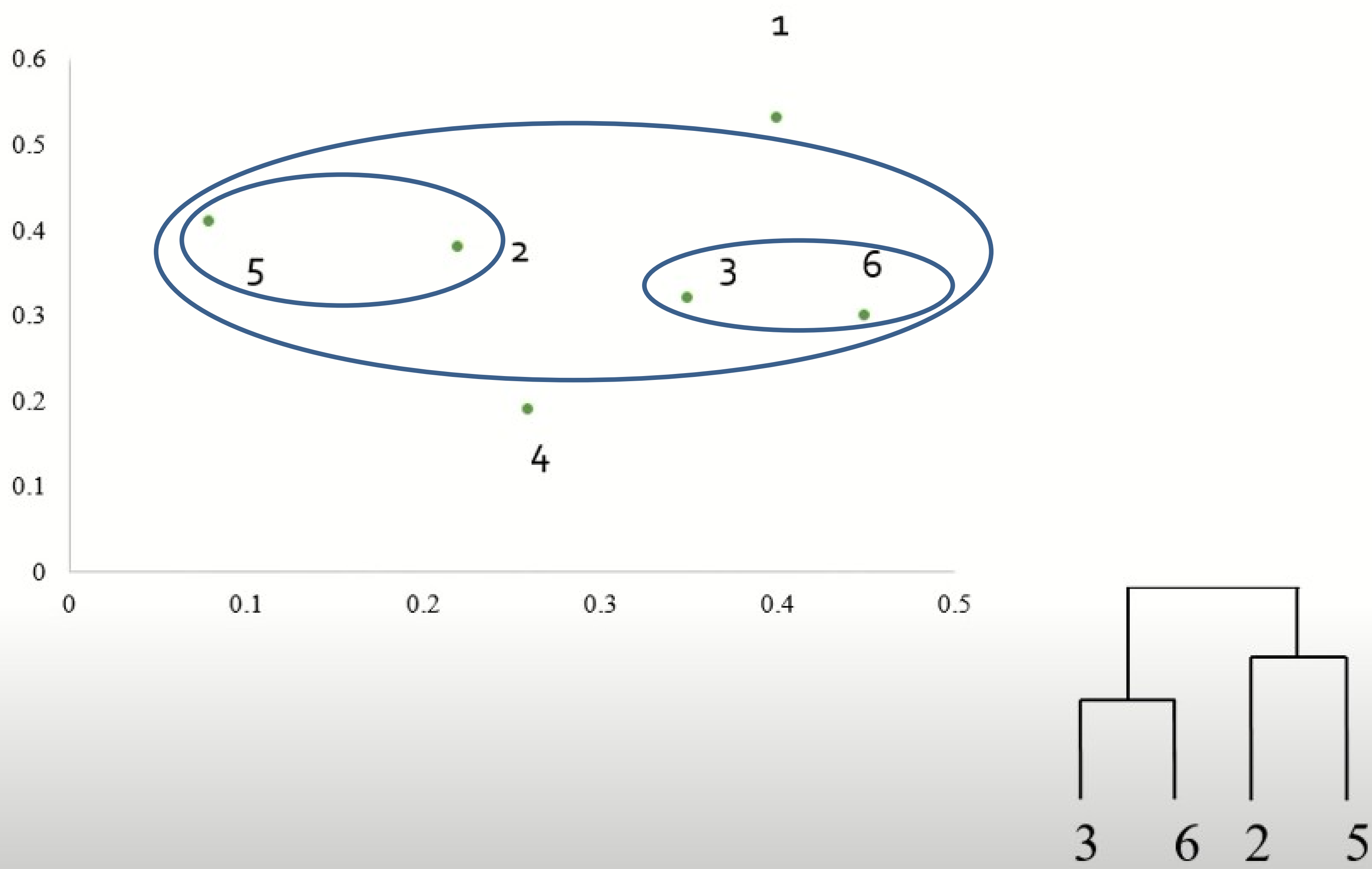
The updated distance matrix for cluster P2,P5

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

Choose minimum value from distance matrix
(Has 2 values same, choose the first)

The distance matrix is

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0



1
0.6
0.5
0.4
0.3
0.2
0.1
0
5
2
3
6
4
0
0.1
0.2
0.3
0.4
0.5
3
6
2
5

To update the distance matrix MIN[dist((P2,P5),(P3,P6)),P1]

MIN[dist((P2,P5),P1), ((P3,P6),P1)]

= min[(0.23,0.22)]

= 0.22

The distance matrix is

	P1	P2,P5	P3,P6	P4
P1	0			
P2,P5	0.23	0		
P3,P6	0.22	0.15	0	
P4	0.37	0.20	0.15	0

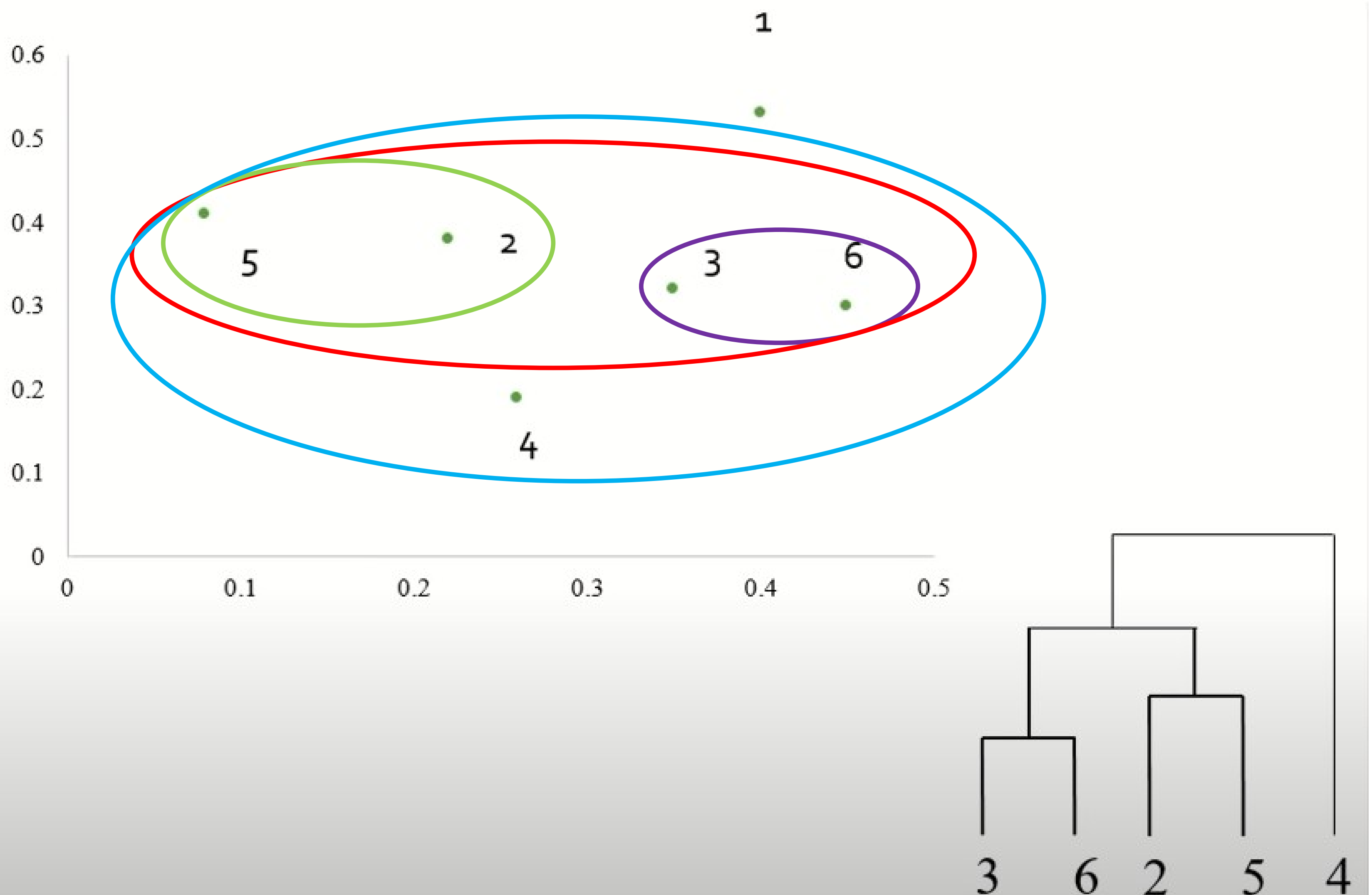
The updated distance matrix for cluster P2,P5,P3,P6

	P1	P2,P5,P3,P6	P4
P1	0		
P2,P5,P3,P6	0.22	0	
P4	0.37	0.15	0

Choose minimum value

The distance matrix is

	P1	P2,P5,P3,P6	P4
P1	0		
P2,P5,P3,P6	0.22	0	
P4	0.37	0.15	0



1
0.6
0.5
0.4
0.3
0.2
0.1
0
5
2
3
6
4
0
0.1
0.2
0.3
0.4
0.5
3
6
2
5
4

- To update the distance matrix MIN[dist(P2,P5,P3,P6),P4]
- MIN[dist((P2,P5,P3,P6),P1), (P4,P1)]

= min[(0.22,0.37)]

= 0.22

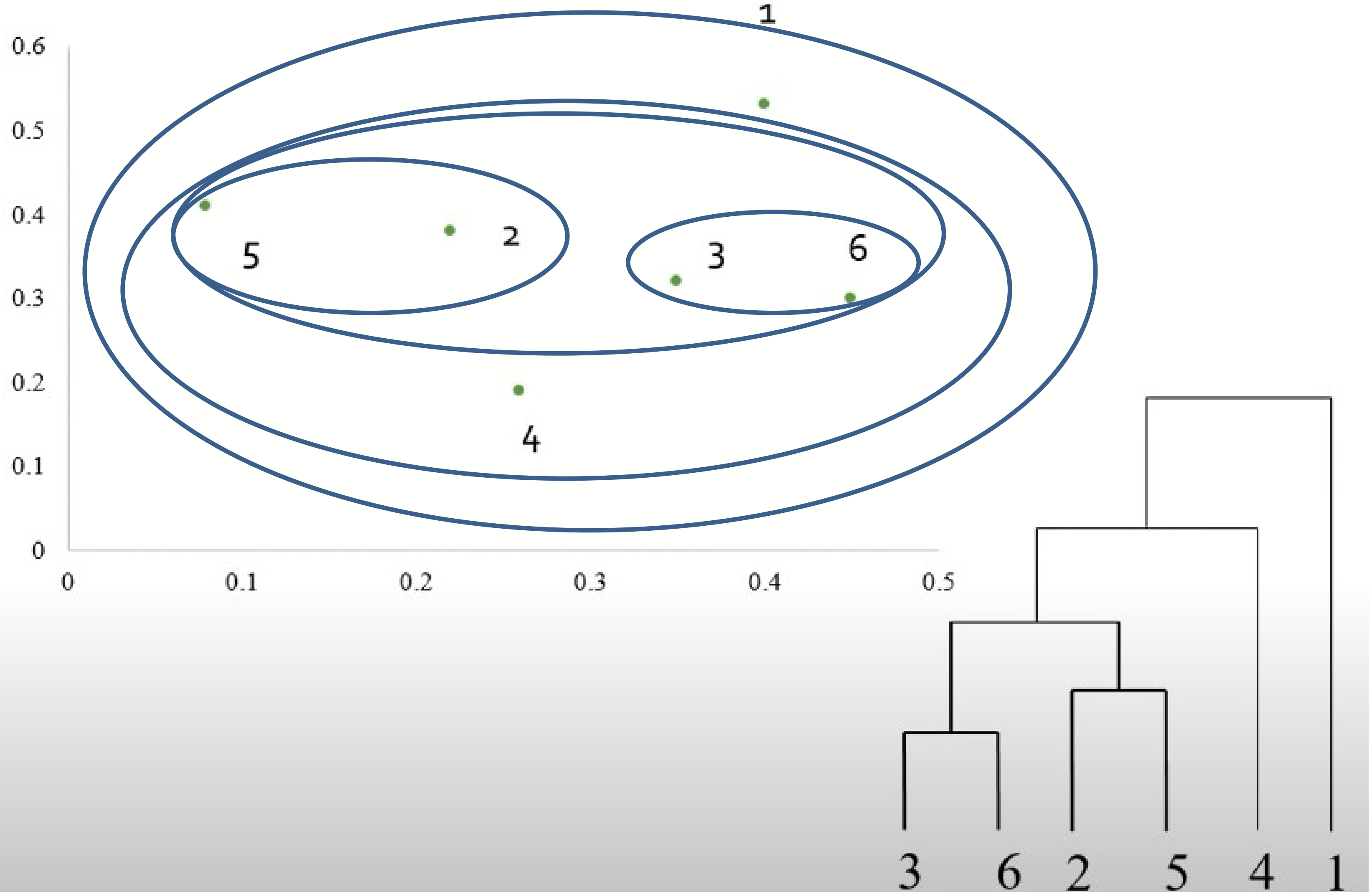
The distance matrix is

	P1	P2,P5,P3,P6	P4
P1	0		
P2,P5,P3,P6	0.22	0	
P4	0.37	0.15	0

The updated distance matrix for cluster P2,P5,P3,P6,P4

	P1	P2,P5,P3,P6,P4
P1	0	
P2,P5,P3,P6,P4	0.22	0

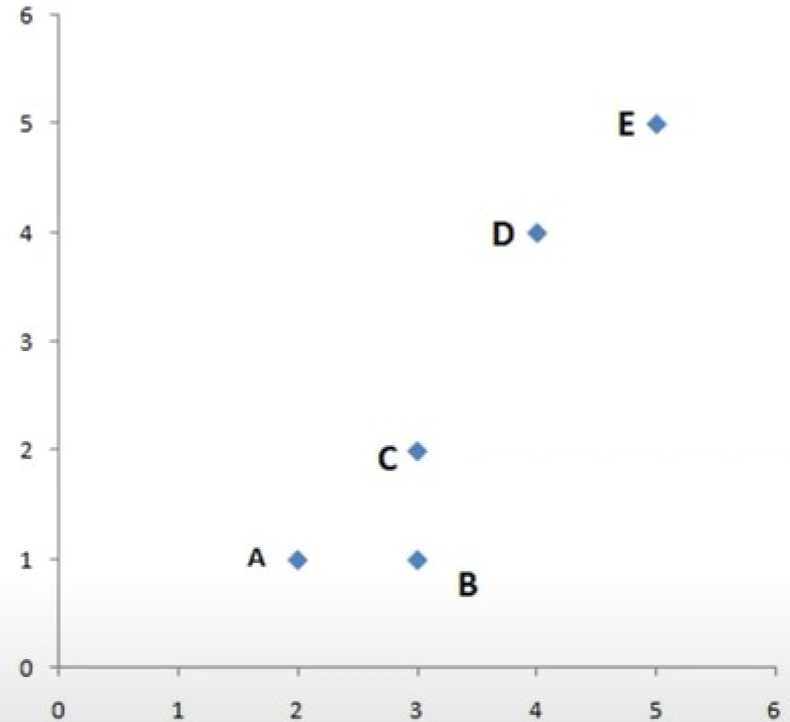
Final Cluster



Example 2

Apply Agglomerative with **Single, Complete and Average Linkage** on following data.

	X	Y
A	2	1
B	3	1
C	3	2
D	4	4
E	5	5



Using Euclidean distance Adjacency Matrix is created

	X	Y
A	2	1
B	3	1
C	3	2
D	4	4
E	5	5

→

A	0				
B	1.00	0			
C	1.41	1.00	0		
D	3.60	3.16	2.24	0	
E	5.00	4.47	3.60	1.41	0
	A	B	C	D	E

A	0				
B	1.00	0			
C	1.41	1.00	0		
D	3.60	3.16	2.24	0	
E	5.00	4.47	3.60	1.41	0
	A	B	C	D	E

In the original matrix , **A & B and B & C**

are located closed to each other at distance 1.

Select any one option.

Merge them into single cluster.

Here A & B merged

a) Single Linkage: Minimum Function

A	0				
B	1.00	0			
C	1.41	1.00	0		
D	3.60	3.16	2.24	0	
E	5.00	4.47	3.60	1.41	0
	A	**B**	**C**	**D**	**E**

AB	0			
C		0		
D		2.24	0	
E		3.60	1.41	0
	AB	**C**	**D**	**E**

min[(C,A), (C,B)]	min[(D,A), (D,B)]	min[(E,A), (E,B)]
min[1.41,1]	min[3.60,3.16]	min[5,4.47]
1	**3.16**	**4.47**

AB	0			
C	1	0		
D	3.16	2.24	0	
E	4.47	3.60	1.41	0
	AB	**C**	**D**	**E**

AB	0			
C	1	0		
D	3.16	2.24	0	
E	4.47	3.60	1.41	0
	AB	**C**	**D**	**E**

AB	0			
C	1	0		
D	3.16	2.24	0	
E	4.47	3.60	1.41	0
	AB	**C**	**D**	**E**

ABC	0		
D		0	
E		1.41	0
	ABC	**D**	**E**

min[(D,AB), (D,C)]	min[(E,AB), (E,C)]
min[3.16,2.24]	min[4.47,3.60]
2.24	**3.60**

ABC	0		
D	2.24	0	
E	3.60	1.41	0
	ABC	**D**	**E**

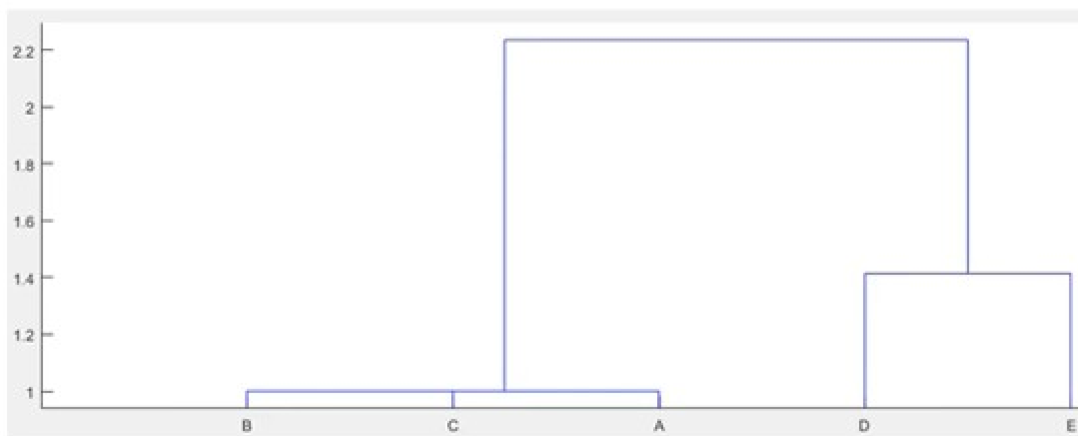
ABC	0		
D	2.24	0	
E	3.60	1.41	0
	ABC	D	E

ABC	0		
D	2.24	0	
E	3.60	1.41	0
	ABC	D	E

ABC	0	
DE		0
	ABC	DE

min[(ABC,D), (ABC,E)]
min[2.24,3.60]
2.24

ABC	0	
DE	**2.24**	0
	ABC	DE



Dendrogram

b) Complete Linkage: Maximum Function

A	0				
B	1.00	0			
C	1.41	1.00	0		
D	3.60	3.16	2.24	0	
E	5.00	4.47	3.60	1.41	0
	A	**B**	**C**	**D**	**E**

AB	0			
C		0		
D		2.24	0	
E		3.60	1.41	0
	AB	**C**	**D**	**E**

max[(C,A), (C,B)]	max[(D,A), (D,B)]	max[(E,A), (E,B)]
max[1.41,1]	max[3.60,3.16]	max[5,4.47]
1.41	**3.60**	**5.00**

AB	0			
C	1.41	0		
D	3.60	2.24	0	
E	5.00	3.60	1.41	0
	AB	**C**	**D**	**E**

AB	0			
C	1.41	0		
D	3.60	2.24	0	
E	5.00	3.60	1.41	0
	AB	**C**	**D**	**E**

AB	0			
C	1.41	0		
D	3.60	2.24	0	
E	5.00	3.60	1.41	0
	AB	**C**	**D**	**E**

ABC	0		
D	3.60	0	
E	5.00	1.41	0
	ABC	**D**	**E**

max[(D,AB), (D,C)]	max[(E,AB), (E,C)]
max[3.60,2,24]	max[5.00,3.60]
3.60	**5.00**

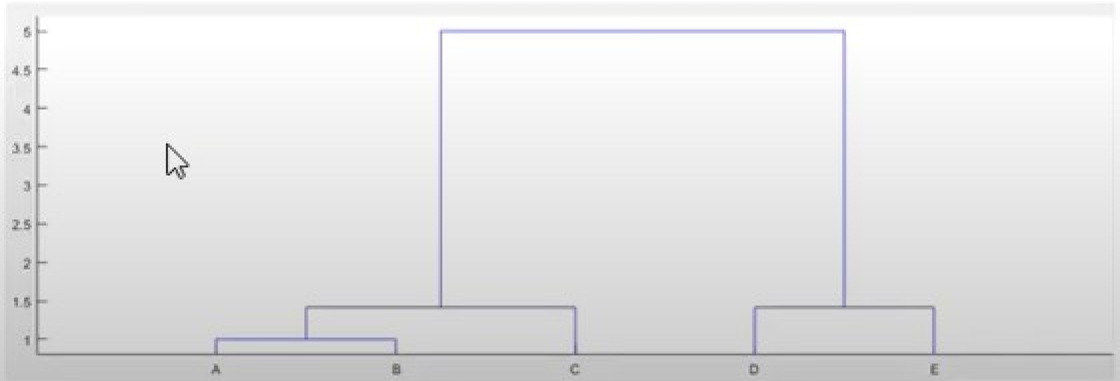
ABC	0		
D	3.60	0	
E	5.00	1.41	0
	ABC	D	E

ABC	0		
D	3.60	0	
E	5.00	1.41	0
	ABC	D	E

ABC	0	
DE		0
	ABC	DE

max[(ABC,D), (ABC,E)]
max[3.60,5.00]
5.00

ABC	0	
DE	**5.00**	0
	ABC	DE



Dendrogram

c) Average Linkage: Average Function

A	0				
B	1.00	0			
C	1.41	1.00	0		
D	3.60	3.16	2.24	0	
E	5.00	4.47	3.60	1.41	0
	A	**B**	**C**	**D**	**E**

AB	0			
C		0		
D		2.24	0	
E		3.60	1.41	0
	AB	**C**	**D**	**E**

(A, B) (C)	(A,B) (D)	(A, B) (E)
avg[(C,A), (C,B)]	avg[(D,A), (D,B)]	avg[(E,A), (E,B)]
avg[1.41,1]	avg[3.60,3.16]	avg[5,4.47]
1.21	**3.38**	**4.74**

AB	0			
C	1.21	0		
D	3.38	2.24	0	
E	4.74	3.60	1.41	0
	AB	**C**	**D**	**E**

AB	0			
C	1.21	0		
D	3.38	2.24	0	
E	4.74	3.60	1.41	0
	AB	**C**	**D**	**E**

AB	0			
C	1.21	0		
D	3.38	2.24	0	
E	4.74	3.60	1.41	0
	AB	**C**	**D**	**E**

ABC	0		
D		0	
E		1.41	0
	ABC	**D**	**E**

ABC	0		
D	3.00	0	
E	4.36	1.41	0
	ABC	**D**	**E**

(A,B,C)(D)	(A,B,C)(E)
avg[(A,D),(B,D), (C,D)]	avg[(A,E), (B,E), (C,E)]
avg[3.6,3,16,2.24]	avg[5,4.47,3.6]
3	**4.36**

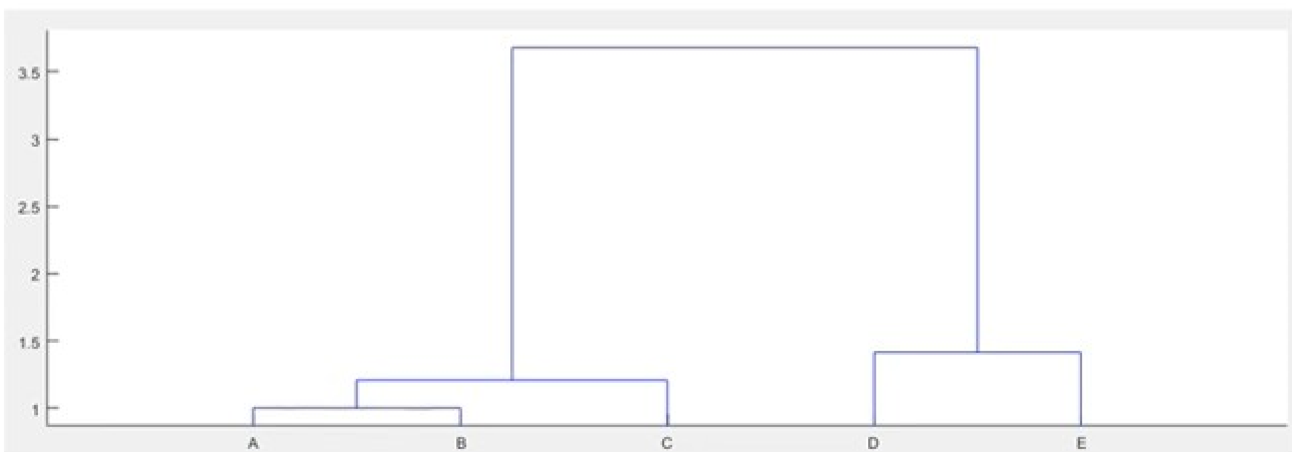
ABC	0		
D	3.0	0	
E	4.36	1.41	0
	ABC	**D**	**E**

ABC	0		
D	3.0	0	
E	4.36	1.41	0
	ABC	**D**	**E**

ABC	0	
DE		0
	ABC	**DE**

(A,B,C) (D,E)
avg[(A,D), (A,E), (B,D), (B,E), (C,D), (C,E),]
avg[3.6,5,3.16,4.47,2.24,3.60]
3.68

ABC	0	
DE	**3.68**	0
	ABC	**DE**



Dendrogram

Function used in R

1. dist( ) – For distance calculation

Syntax :

dist(x, method = "euclidean“)

Where,

- **x** - a numeric matrix, data frame or "dist" object.
- **method** - the distance measure to be used. This must be one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski".

2. hclust( ) – Hierarchial clustering

Syntax :

hclust(d, method = "complete")

Where,

- **d** **-** a dissimilarity structure as produced by dist.
- **method** **-** the agglomeration method to be used. This can be one of "single", "complete", "average“.