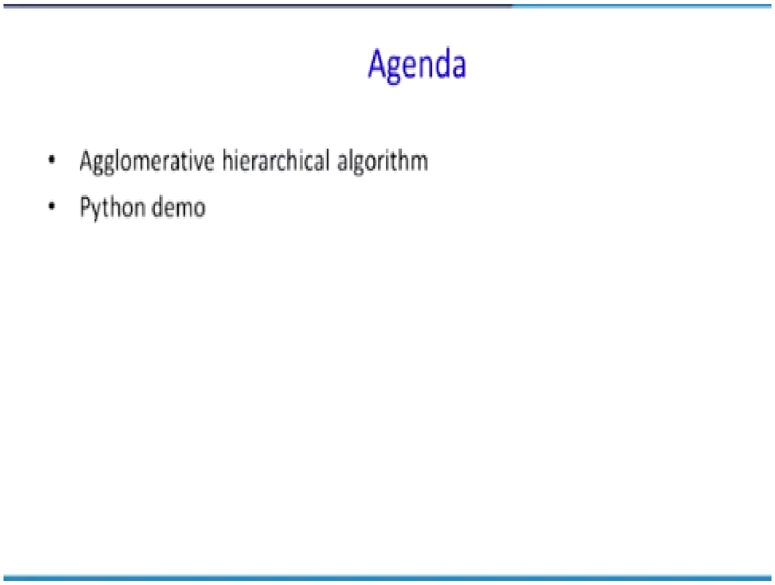
Data Analytics with Python
Prof. Ramesh Anbanandam
Department of Management Studies
Indian Institute of Technology - Roorkee

Lecture – 56
Hierarchical Method of Clustering - II

In a previous lecture, I have explained introduction to hierarchical clustering and different types of distance measures. In this lecture, I have taken a numerical example, with the help of the numerical example, I am going to explain how to do hierarchical clustering method, for that same problem, I am going to explain how to use Python for doing hierarchical clustering.

(Refer Slide Time: 00:53)

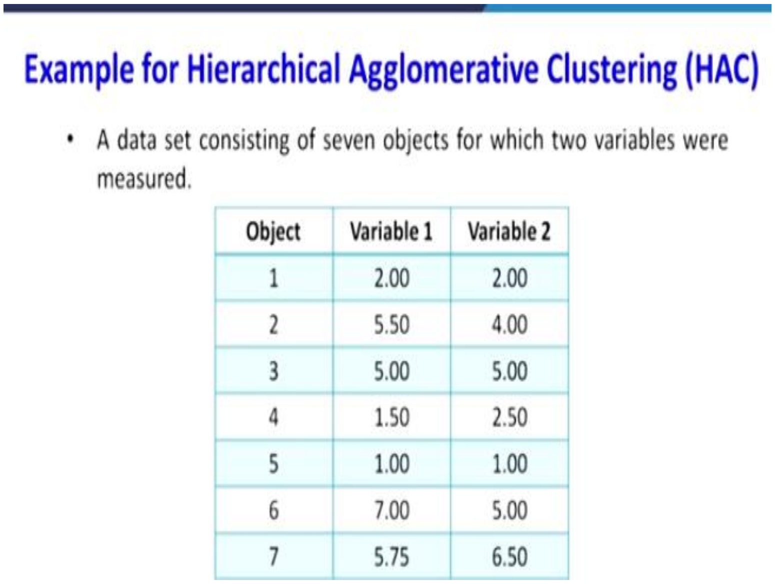


Agenda

- Agglomerative hierarchical algorithm
- Python demo

So, the agenda for this lecture is agglomerative hierarchical algorithm, the second one is python demo.

(Refer Slide Time: 00:57)



Example for Hierarchical Agglomerative Clustering (HAC)

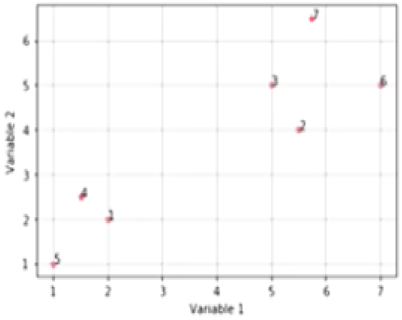
- A data set consisting of seven objects for which two variables were measured.

Object	Variable 1	Variable 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

So the example of hierarchical agglomerative clustering, HAC; a data set consists of 7 objects for which 2 variables were measured. There are 7 objects; 1, 2, 3, 4, 5, 6, 7, variable 1 and variable 2. So, variable 1 is 2, 5, 5.5, 5 and so on. Variable 2 is 2, 4, 5, 2, 1, 5, 6 and so on.

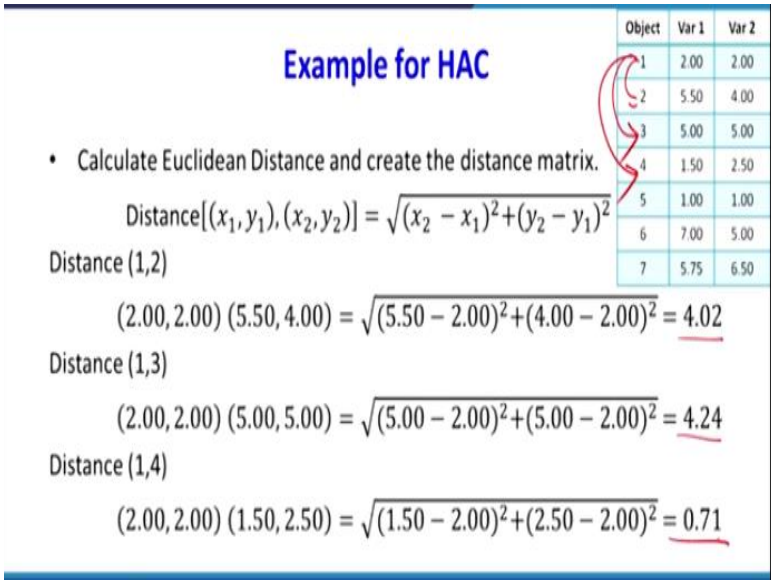
(Refer Slide Time: 01:24)

Scatter plot



When you plot in the scatterplot, it is appearing that there are 7 data set, variable 1 in x axis, variable 2 in y axis. Now, we are going to do hierarchical clustering for this data set.

(Refer Slide Time: 01:37)



Example for HAC

- Calculate Euclidean Distance and create the distance matrix.

$$\text{Distance}[(x_1,y_1),(x_2,y_2)] = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$

Distance (1,2)

$$(2.00,2.00)\ (5.50,4.00) = \sqrt{(5.50-2.00)^2+(4.00-2.00)^2} = \underline{4.02}$$

Distance (1,3)

$$(2.00,2.00)\ (5.00,5.00) = \sqrt{(5.00-2.00)^2+(5.00-2.00)^2} = \underline{4.24}$$

Distance (1,4)

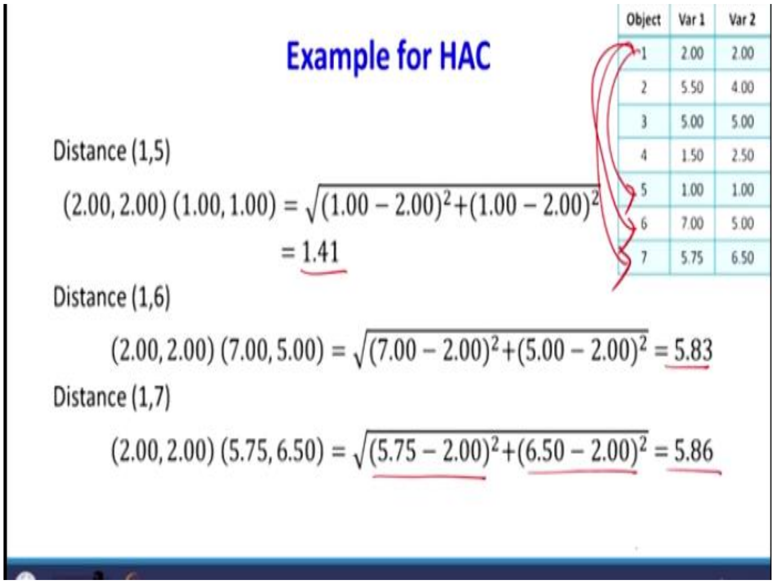
$$(2.00,2.00)\ (1.50,2.50) = \sqrt{(1.50-2.00)^2+(2.50-2.00)^2} = \underline{0.71}$$

Object	Var 1	Var 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

What is the first one; calculate the Euclidean distance and create a distance matrix, what we are going to do; first we are going to create a distance matrix, for that we are going to find out the distance between 1 and 2, 1 and 3, 1 and 4, 1 and 5, 1 and 6, 1 and 7 and 2 and 3, 2 and 4, 2 and 5, 2 and 6 and 2 and 7, then 3 and 4, 3 and 5, 3 and 6 and 3 and 7 then 4 and 5, 4 and 6 and 4 and 7, then 5 and 6 and 5 and 7 finally, 6 and 7.

First we will find the distance between object 1 and 2, so the object 1 is 2, 2, object 2 is 5.5, 4, after finding the Euclidean distance, we know the formula is x2 – x1 square + y2 – y1 whole square so, 5.5 – 2 whole square + 4 – 2 whole square that is your 4.02. Then, let us find the distance between; now we have seen 1 and 2, now the distance between 1 and 3. So, 1 and 3 is the position of 1 is 2, 2, position of 3 is 5, 5, so it is 5, 2 whole square + 5, 2 whole square. So, it is 4.24, now let us find the distance between 1 and 4, so this 1 and 4. So, 2, 2 and 1.5, 2.5, so the distance is 1.5 – 2 whole square + 2.5 – 2 whole square that is 0.71.

(Refer Slide Time: 03:35)



Example for HAC

Object	Var 1	Var 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

Distance (1,5)

$$(2.00, 2.00)\ (1.00, 1.00) = \sqrt{(1.00-2.00)^2+(1.00-2.00)^2}$$
$$= \underline{1.41}$$

Distance (1,6)

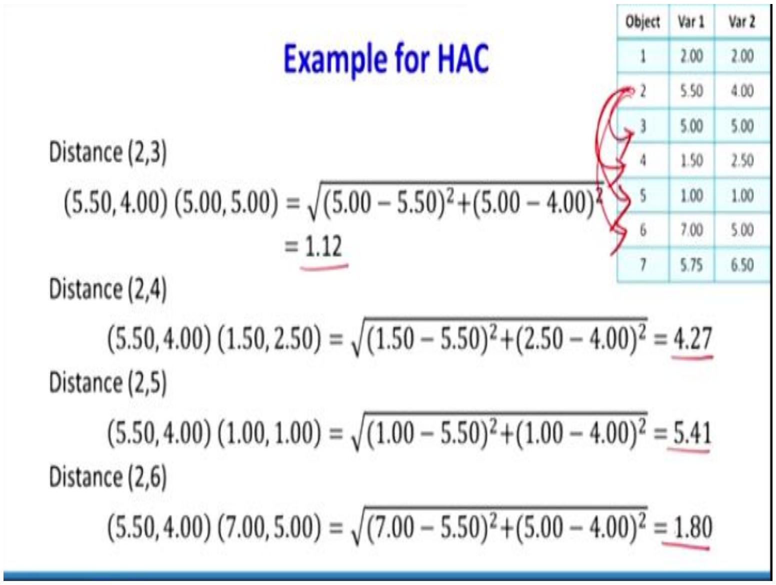
$$(2.00, 2.00)\ (7.00, 5.00) = \sqrt{(7.00-2.00)^2+(5.00-2.00)^2} = \underline{5.83}$$

Distance (1,7)

$$(2.00, 2.00)\ (5.75, 6.50) = \sqrt{\underline{(5.75-2.00)}^2+\underline{(6.50-2.00)}^2} = \underline{5.86}$$

Now, we will find the distance between 1 and 5, so this 1 and 5 because this we have done it, so 1 and 5 is 2, 2; the position of the 5th object is 1, 1, so it is 1 – 2 whole square + 1 – 2 whole square, it is 1.41. Now, we will find the distance 1 and 6 that is 2, 2, then 7, 5, so it is 7 – 2 whole square + 5 – 2 whole square that is 5.83. Similarly, we can find the distance between 1 and 7 that is 2, 2 versus 5.75 and 6.5.

(Refer Slide Time: 04:30)



Example for HAC

Object	Var 1	Var 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

Distance (2,3)

$$(5.50, 4.00)\ (5.00, 5.00) = \sqrt{(5.00-5.50)^2+(5.00-4.00)^2}$$
$$= \underline{1.12}$$

Distance (2,4)

$$(5.50, 4.00)\ (1.50, 2.50) = \sqrt{(1.50-5.50)^2+(2.50-4.00)^2} = \underline{4.27}$$

Distance (2,5)

$$(5.50, 4.00)\ (1.00, 1.00) = \sqrt{(1.00-5.50)^2+(1.00-4.00)^2} = \underline{5.41}$$

Distance (2,6)

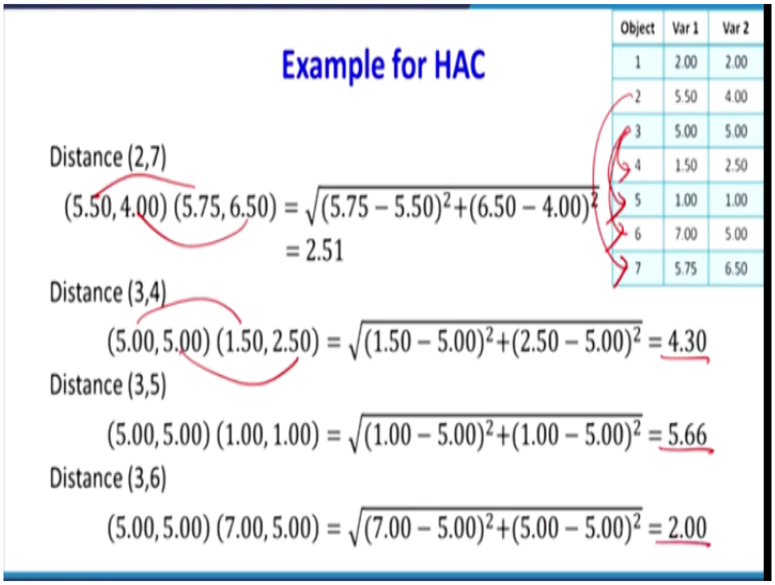
$$(5.50, 4.00)\ (7.00, 5.00) = \sqrt{(7.00-5.50)^2+(5.00-4.00)^2} = \underline{1.80}$$

So, it is a 5.75 – 2 whole square + 6.5 – 2 whole square, we got 5.86, now we have done 1 versus all the point, now we go second versus 3, the distance between 2 and 3, the position of 2 is 5, 5, 4; 3 is 5, 5, so the distance is 5 – 5.5 whole square + 5 – 4 whole square, it is 1.12. Now, 2 and 4;

5.5, 4 is the position of 2, 4 is the position of 2, the position of object 4 is 1.5, 2.5, so the distance is 1.5 - 5.5 whole square + 2.5 – 4 whole square that is 4.27.

Next we will find the distance between 2 and 5; we know the position of 2 is 5.5, 4, the position of object 5 is 1, 1, see this point, the distance is 1 – 5.5 whole square + 1 – 4 whole square that is 5.41. Now, we can find the distance between 2 and 6, so the position of object 6 is 7, 5, so the distance is 7 – 5.5 whole square – 5 – 4 whole square that is 1.81.

(Refer Slide Time: 06:04)



Example for HAC

Object	Var 1	Var 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

Distance (2,7)

$$(5.50, 4.00)\ (5.75, 6.50) = \sqrt{(5.75-5.50)^2+(6.50-4.00)^2}$$
$$= 2.51$$

Distance (3,4)

$$(5.00, 5.00)\ (1.50, 2.50) = \sqrt{(1.50-5.00)^2+(2.50-5.00)^2} = \underline{4.30}$$

Distance (3,5)

$$(5.00, 5.00)\ (1.00, 1.00) = \sqrt{(1.00-5.00)^2+(1.00-5.00)^2} = \underline{5.66}$$

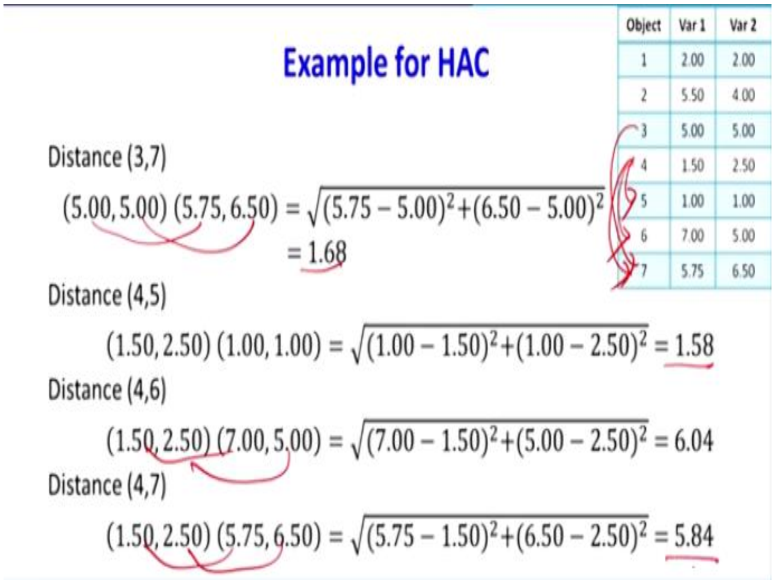
Distance (3,6)

$$(5.00, 5.00)\ (7.00, 5.00) = \sqrt{(7.00-5.00)^2+(5.00-5.00)^2} = \underline{2.00}$$

The next one; we are going to find the distance between 2 and 7, so 2 and 7, so the position of object 7 is 5.75, 6.5, so the distance is 5.75 – 5.5 whole square + 6.5 – 4 whole square, so this difference whole square and this difference whole square that is 2.51. Now, 2; the point 2, the object 2 and other all 7's we compare, now we will compare 3; 3 and 4, 3 and 5, 3 and 6 and 3 and 7.

So, the 3 and 4 distance is the position of 0.3 is 5, 5 and 4 is 1.5, 2.5, so this distance versus this distance, this difference, so 1.5 – 5 whole square + 2.5 – 5 whole square that is 4.30. Next 3 and 5, so 5, 5 and the 1; 1, 1 is the position of object 5, so what will happen; 1 – 5 whole square + 1 – 5 whole square, 5.66. Now, 3 and 6; so 3 is 5, 5; 6 is 7, 5, so the difference is 7 – 5 whole square + 5 – 5 whole square, the distance is 2.

(Refer Slide Time: 07:49)



Example for HAC

Object	Var 1	Var 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

Distance (3,7)

$$(5.00, 5.00)\ (5.75, 6.50) = \sqrt{(5.75-5.00)^2+(6.50-5.00)^2}$$
$$= 1.68$$

Distance (4,5)

$$(1.50, 2.50)\ (1.00, 1.00) = \sqrt{(1.00-1.50)^2+(1.00-2.50)^2} = 1.58$$

Distance (4,6)

$$(1.50, 2.50)\ (7.00, 5.00) = \sqrt{(7.00-1.50)^2+(5.00-2.50)^2} = 6.04$$

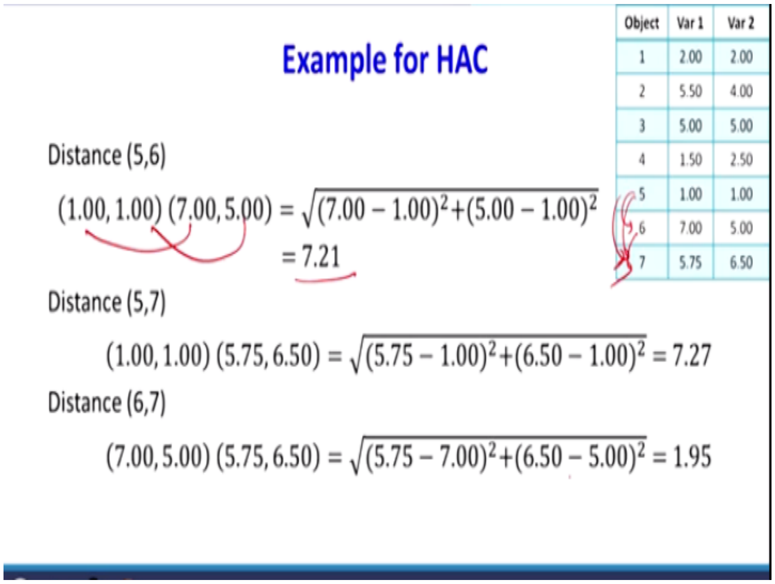
Distance (4,7)

$$(1.50, 2.50)\ (5.75, 6.50) = \sqrt{(5.75-1.50)^2+(6.50-2.50)^2} = 5.84$$

Now, we are going to find the next one; 3, 7, so 3, 7 what is the distance; we know position of 3 is 5, 5; 7 is 5.75 – 6.50, so this difference whole square versus this difference whole square, so 5.75 – 5 whole square + 6.5 – 5 whole square that is 1.68. Now, we have to find out 3 versus all the point, now we will take 4, 5 and 4, 6 and 4, 7. So, the distance between 4 and 5 is 1.5 – 2.5 is a the position of object 4, for the 5th one it is 1, 1.

When you look at the distance, it is 1.58, then 4 and 6 we have done that one, now we will go for 4 and 6. The 4 and 6 is the 4th 1.5 – 2.5, the 6th position is 7, 5, so the difference is 7 – 1.5 whole square + 5 – 2.5 whole square equal to 6.04. Now, we will find out 4 versus 7, the distance between 4 and 7, the 1.5, 2.5, the position of object 7 is 5.75 – 6.50, so the this difference whole square plus this difference whole square.

(Refer Slide Time: 09:25)



Example for HAC

Object	Var 1	Var 2
1	2.00	2.00
2	5.50	4.00
3	5.00	5.00
4	1.50	2.50
5	1.00	1.00
6	7.00	5.00
7	5.75	6.50

Distance (5,6)

$$(1.00, 1.00)\ (7.00, 5.00) = \sqrt{(7.00-1.00)^2+(5.00-1.00)^2}$$
$$= 7.21$$

Distance (5,7)

$$(1.00, 1.00)\ (5.75, 6.50) = \sqrt{(5.75-1.00)^2+(6.50-1.00)^2} = 7.27$$

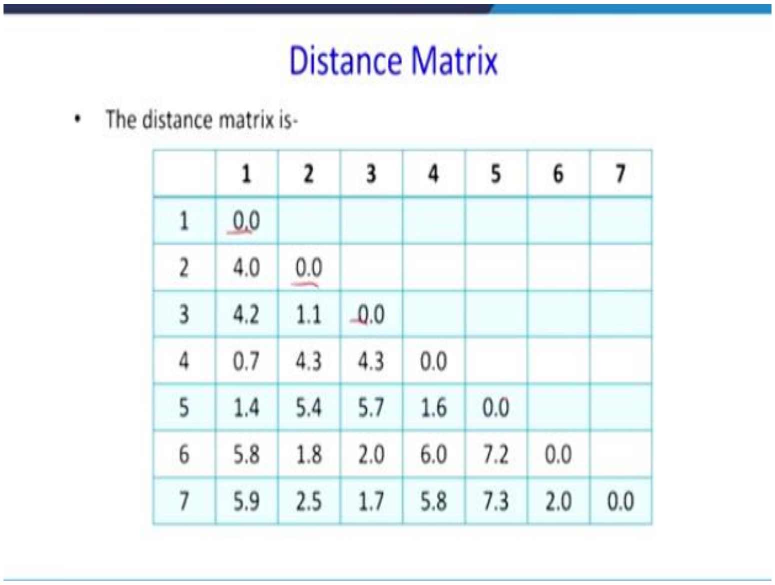
Distance (6,7)

$$(7.00, 5.00)\ (5.75, 6.50) = \sqrt{(5.75-7.00)^2+(6.50-5.00)^2} = 1.95$$

So, 5.5; 5.75 – 1.5 whole square + 6.5 – 2.5 whole square that is 5.84, 5 we have completed, now we will go 5 and 6 and 5 and 7, the distance between 5 and 6 is 1, 1 is a position of object 5, for 6th one, it is 7, 5, so it is 7 - 1 whole square + 5 – 1 whole square that is 7.21. Now, we will find out distance between 5 and 7 that is 1, 1 versus 5.75 – 6.5, so it is 5.75 – 1 whole square + 6.5 – 1 whole square, 7.27.

Now, 6 versus 7, we will find out the distance, so that is position of object 6 is 7, 5, position of object 7 is 5.75, 6.5, so the distance is 5.75 - 7 whole square + 6.5 – 5.00 whole square that is 1.95.

(Refer Slide Time: 10:24)



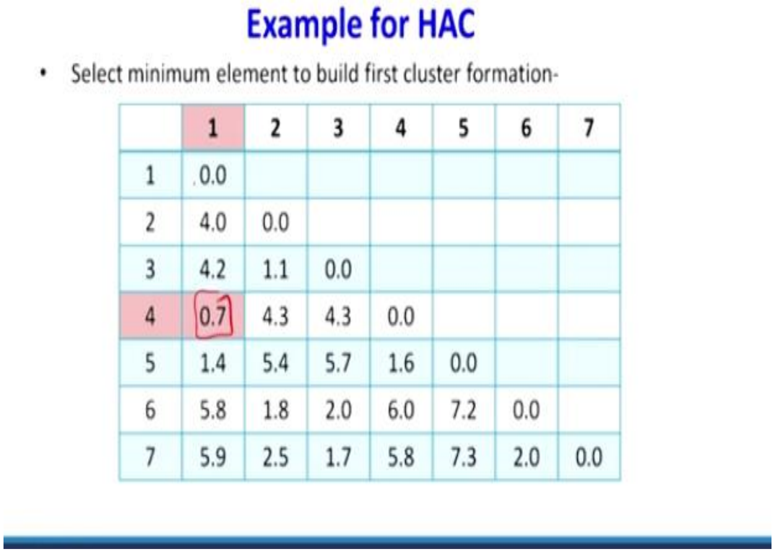
Distance Matrix

- The distance matrix is-

	1	2	3	4	5	6	7
1	0.0						
2	4.0	0.0					
3	4.2	1.1	0.0				
4	0.7	4.3	4.3	0.0			
5	1.4	5.4	5.7	1.6	0.0		
6	5.8	1.8	2.0	6.0	7.2	0.0	
7	5.9	2.5	1.7	5.8	7.3	2.0	0.0

Now, we have compared all the distance, as I told you this is a distance matrix, see that (()) (10:32) 0 because the distance is 0, we got distance between 2 and 1, 3 and 1, 4 and 1, 5 and 1, 6 and 1, 7 and 1, this value which we got from our previous slides, so we got the distance matrix.

(Refer Slide Time: 10:47)



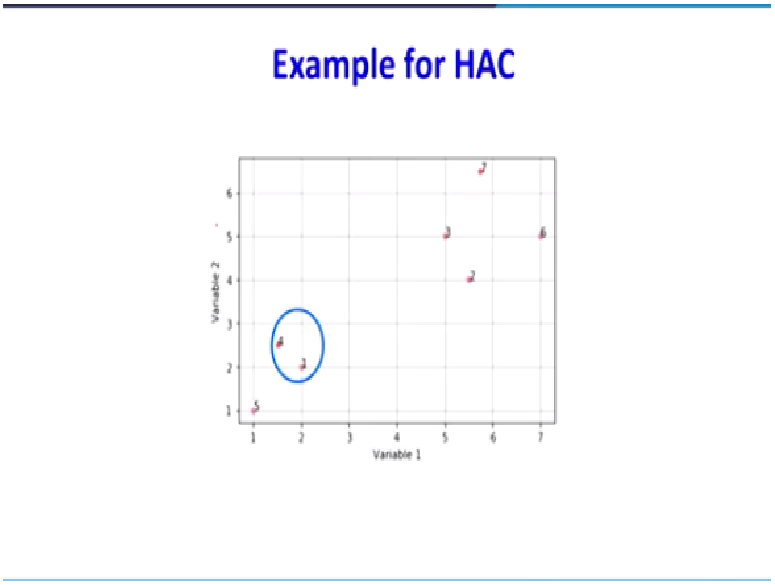
Example for HAC

- Select minimum element to build first cluster formation-

	1	2	3	4	5	6	7
1	0.0						
2	4.0	0.0					
3	4.2	1.1	0.0				
4	0.7	4.3	4.3	0.0			
5	1.4	5.4	5.7	1.6	0.0		
6	5.8	1.8	2.0	6.0	7.2	0.0	
7	5.9	2.5	1.7	5.8	7.3	2.0	0.0

After the distance matrix, for that hierarchical agglomerative method, select minimum element to build the first cluster formation, so among this, find out where there is a minimum distance, so the minimum distance is this one; 0.7. What are the 2 objects between 4 and 1, so what is going to do; we are going to form a cluster, in that there are 2 element is going to be there; 4 and 1.

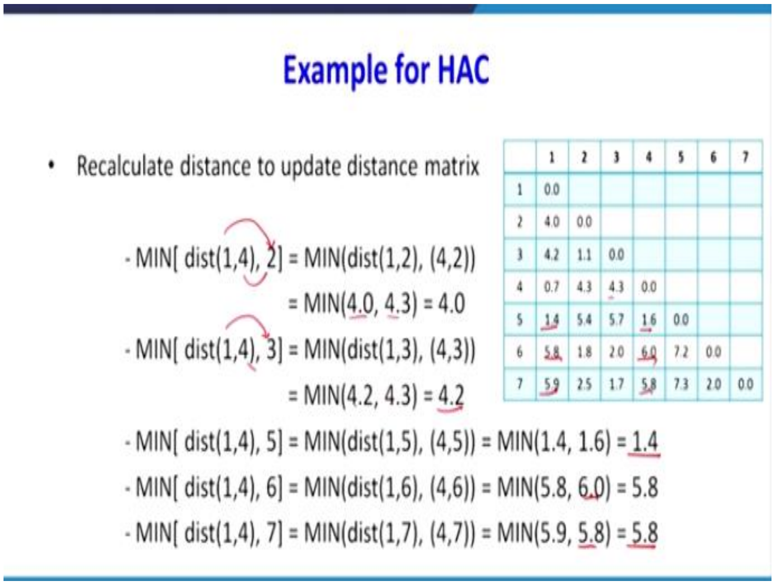
(Refer Slide Time: 11:23)



Example for HAC

So, the 4 and 1 will form a cluster, so we got this one, 4 and 1, this is our first cluster.

(Refer Slide Time: 11:29)



Example for HAC

- Recalculate distance to update distance matrix
  - MIN[dist(1,4), 2] = MIN(dist(1,2), (4,2))
    = MIN(4.0, 4.3) = 4.0
  - MIN[dist(1,4), 3] = MIN(dist(1,3), (4,3))
    = MIN(4.2, 4.3) = 4.2
  - MIN[dist(1,4), 5] = MIN(dist(1,5), (4,5)) = MIN(1.4, 1.6) = 1.4
  - MIN[dist(1,4), 6] = MIN(dist(1,6), (4,6)) = MIN(5.8, 6.0) = 5.8
  - MIN[dist(1,4), 7] = MIN(dist(1,7), (4,7)) = MIN(5.9, 5.8) = 5.8

	1	2	3	4	5	6	7
1	0.0						
2	4.0	0.0					
3	4.2	1.1	0.0				
4	0.7	4.3	4.3	0.0			
5	1.4	5.4	5.7	1.6	0.0		
6	5.8	1.8	2.0	6.0	7.2	0.0	
7	5.9	2.5	1.7	5.8	7.3	2.0	0.0

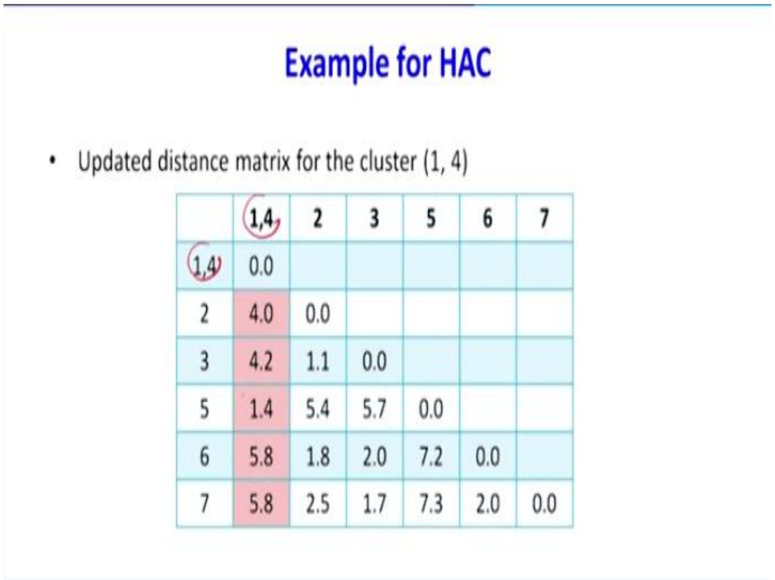
We are going to find the distance between that cluster 1, 4 forming a distance versus 2, so what we are going to do; we are going to find out distance between 1 and 2 and 4 and 2, whichever is minimum, we are going to take that distance because in cluster 1, already there are 2 object is there, we are going to consider the minimum distance. So, the distance between 1, 2, see that this value 4 and 4, 2 is 4.3, this value we take from the table.

So, the minimum distance is 4, the next one 1, 4 versus 3, similarly 1, 4 versus 5, 1, 4 versus 6, 1, 4 versus 7, so if you want to know this cluster versus 3, so minimum distance between 1, 3 and 4, 3; this 1, 3 and 4, 3, so 1, 3 it is a 4.2; 4, 3 it is a 4.3, this value, so minimum is 4.2. Now, 4th we cannot go, already the 4th one is already gone to that cluster, so we will go to 5th, so 1, 4 versus 5th object, for that we have to find out the minimum distance between 1, 5 and 4, 5.

So, 1, 5 is 1.4, this value and 4, 5 is your this value, 1.6 that 1.6, the minimum is 1.4, then 1, 4 with 6, now what happened here; we have to find out the minimum distance 1, 6 and 4, 6. So 1, 6 is 5.8 this value and 4, 6 that is 6, this value, so minimum is 5.8. Now, why we are taking 1.4 because this we formed one cluster, so from this cluster there are 2 point; 2 object 1 and 4. From 1 and 4, this 7 is how far away?

So, 2 way we have to do; 1 and 7 we have to find the distance and 4 and 7 we have to find the distance whichever minimum that has to be kept. So, 1 and 7, 4 and 7, so 1 and 7 is this one, 5.9, 4 and 7 that is your 5.8, so minimum value is 4.8.

(Refer Slide Time: 14:26)



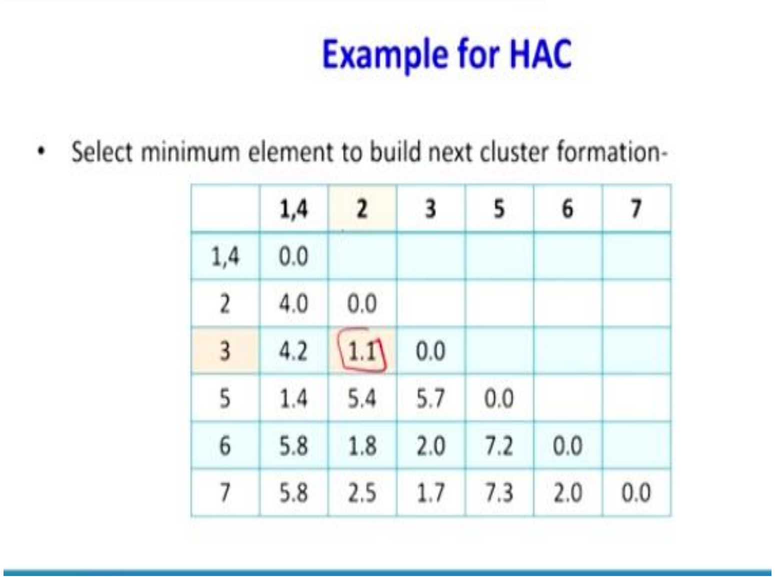
Example for HAC

- Updated distance matrix for the cluster (1, 4)

	1,4	2	3	5	6	7
1,4	0.0					
2	4.0	0.0				
3	4.2	1.1	0.0			
5	1.4	5.4	5.7	0.0		
6	5.8	1.8	2.0	7.2	0.0	
7	5.8	2.5	1.7	7.3	2.0	0.0

Now, we are going to update this value, so what update we have done that one; since 1 and 4 form a new cluster, so now we will find this value, how we got this value? So, 2, 1, 4; so 2, 1, 4 is 4, so we updated. Now, so updated the distance, this distance matrix is going to be used for were further steps.

(Refer Slide Time: 14:50)



Example for HAC

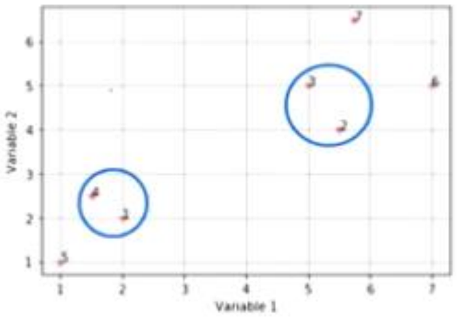
- Select minimum element to build next cluster formation-

	1,4	2	3	5	6	7
1,4	0.0					
2	4.0	0.0				
3	4.2	1.1	0.0			
5	1.4	5.4	5.7	0.0		
6	5.8	1.8	2.0	7.2	0.0	
7	5.8	2.5	1.7	7.3	2.0	0.0

So, now in that updated matrix, again you find out which is minimum, so this 1.1 is minimum that forms 3 and 2. So, now what is happening; the 3 and 2 is going to form one cluster.

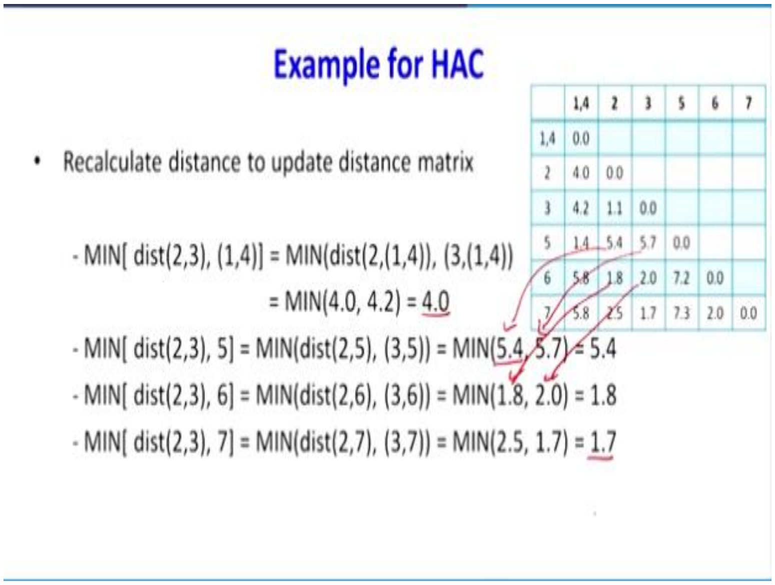
(Refer Slide Time: 15:06)

Example for HAC



Yeah, 3 and 2 is formed one cluster.

(Refer Slide Time: 15:09)



Example for HAC

- Recalculate distance to update distance matrix

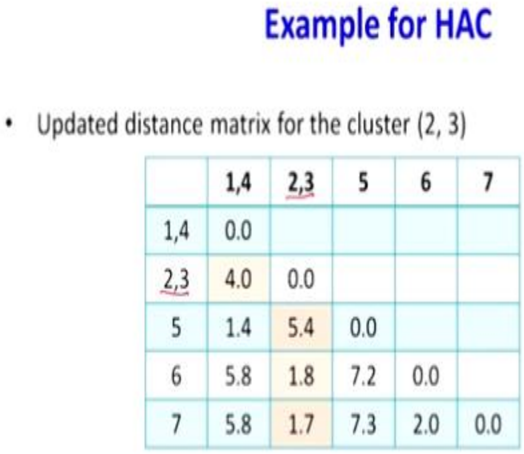
- MIN[dist(2,3), (1,4)] = MIN(dist(2,(1,4)), (3,(1,4))
= MIN(4.0, 4.2) = 4.0
- MIN[dist(2,3), 5] = MIN(dist(2,5), (3,5)) = MIN(5.4, 5.7) = 5.4
- MIN[dist(2,3), 6] = MIN(dist(2,6), (3,6)) = MIN(1.8, 2.0) = 1.8
- MIN[dist(2,3), 7] = MIN(dist(2,7), (3,7)) = MIN(2.5, 1.7) = 1.7

	1,4	2	3	5	6	7
1,4	0.0					
2	4.0	0.0				
3	4.2	1.1	0.0			
5	1.4	5.4	5.7	0.0		
6	5.8	1.8	2.0	7.2	0.0	
7	5.8	2.5	1.7	7.3	2.0	0.0

Now, since 2 and 3 is formed a distance and already there is 1 cluster; 1, 4, so we are going to find out the distance between 2 and 1, 4 and 3 and 1, 4. So, 2 and 1, 4 you can find out this is 4 because 1 and 4 formed a cluster, then 3 and 1, 4; 3 and 1, 4 is 4.2, in this minimum is 4. Similarly, the distance between this newly formed cluster 2, 3 versus 5, 5th point, we are not go to 4th one because 4 is already formed a cluster, so the 5th one, we have to find out the minimum distance of 2, 5 versus 3, 5.

So, in 2, 5, the distance is 5.4 this value, this value; 3, 5; 5.7, so we got this value, in between 5.4 and 5.7, minimum is 5.4. Now, 2, 3 versus 6, now we have to find out the distance between 2, 6 and 3, 6. So, 2, 6; where is 2, 6; 6, 2, this is 1.8, this value and 3, 6, 6, 3, see these 2 value, so in that minimum is 1.8. Now, the last point is 2, 3 versus 7, so what we have to do; we have to find the minimum distance is 2 and 7 and 3 and 7. So, between 2 and 7, minimum distance is 2.5, 3 and 7 minimum distance is 1.7, so out of this minimum is 1.7, now we are going to update this new distance.

(Refer Slide Time: 17:03)



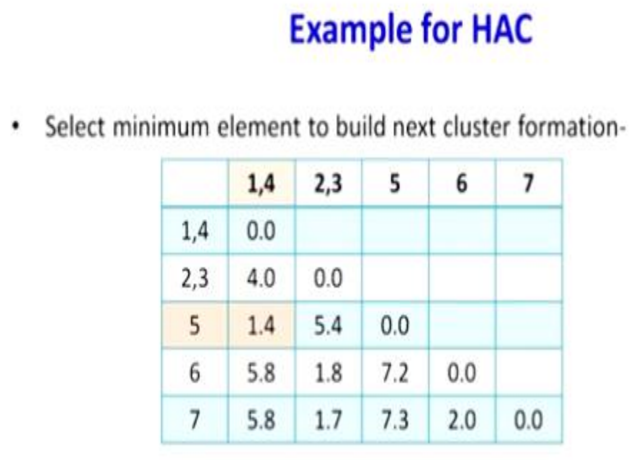
Example for HAC

- Updated distance matrix for the cluster (2, 3)

	1,4	2,3	5	6	7
1,4	0.0				
2,3	4.0	0.0			
5	1.4	5.4	0.0		
6	5.8	1.8	7.2	0.0	
7	5.8	1.7	7.3	2.0	0.0

So, this was updated distance but you see that 2 and 3 is formed 1 cluster, previously 1 and 4 will formed a cluster, now in the next slides what we are going to do; among these new updated distance matrix which is the lowest value.

(Refer Slide Time: 17:17)



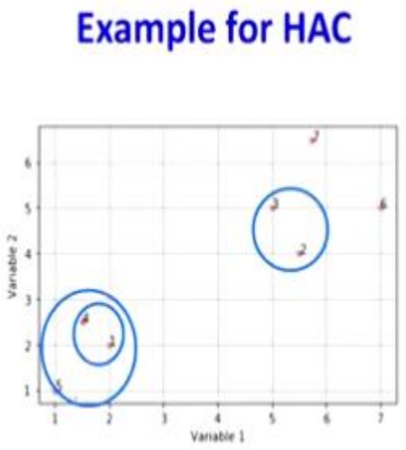
Example for HAC

- Select minimum element to build next cluster formation-

	1,4	2,3	5	6	7
1,4	0.0				
2,3	4.0	0.0			
5	1.4	5.4	0.0		
6	5.8	1.8	7.2	0.0	
7	5.8	1.7	7.3	2.0	0.0

So, select the minimum element to build the next cluster formation, in that the minimum point is this 1.4, so what is going to happen; this object 5 is going to join with this cluster 1, 4.

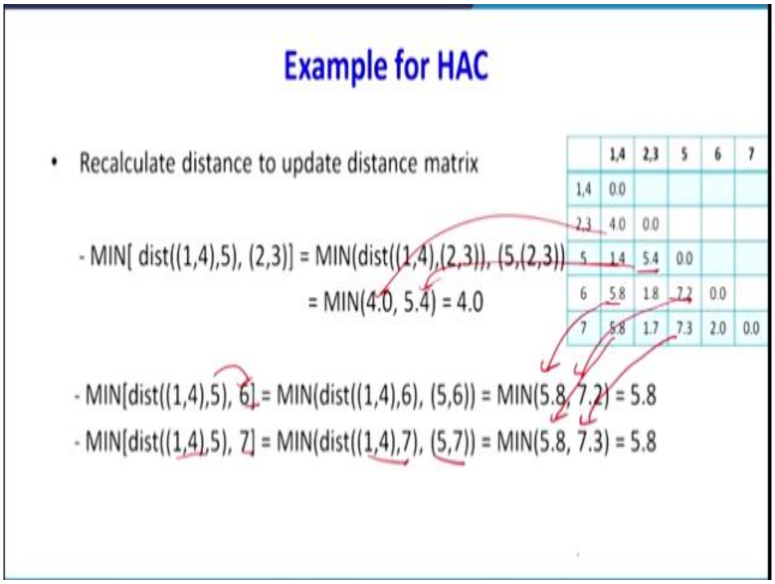
(Refer Slide Time: 17:35)



Example for HAC

You see that the object 5 is going to form with cluster 1, 4.

(Refer Slide Time: 17:41)



Example for HAC

- Recalculate distance to update distance matrix

  - MIN[dist((1,4),5), (2,3)] = MIN(dist((1,4),(2,3)), (5,(2,3))

    = MIN(4.0, 5.4) = 4.0

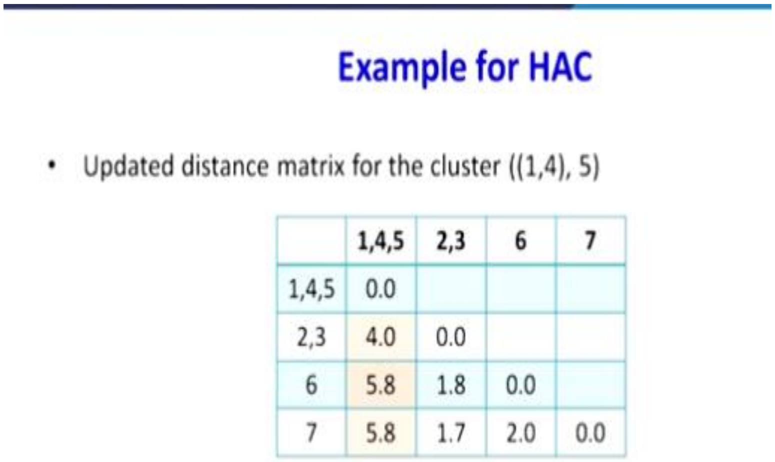
  - MIN[dist((1,4),5), 6] = MIN(dist((1,4),6), (5,6)) = MIN(5.8, 7.2) = 5.8
  - MIN[dist((1,4),5), 7] = MIN(dist((1,4),7), (5,7)) = MIN(5.8, 7.3) = 5.8

	1,4	2,3	5	6	7
1,4	0.0				
2,3	4.0	0.0			
5	1.4	5.4	0.0		
6	5.8	1.8	7.2	0.0	
7	5.8	1.7	7.3	2.0	0.0

Now, what we are going to do; you see that already there is; this is a new cluster, this is not new cluster, already 1, 4 is there, one cluster that again, the object 5 is added, so the distance between this cluster versus another cluster, in that there are 2 point; 2 object you see 2, 3. So, what we have to do; minimum distance between 1 and 4 and 2, 3 and 5 and 2, 3, so what is happening here; 1 and 4 is there, 2, 3 is there, the distance is 4, this value we got it.

The distance between 5, 2, 3 so this is 5.4, so this value we got it here, this value got it, the distant minimum is 4, similarly minimum distance between 1, 4, 5 versus 6, so what you have to do; 1, 4 versus 6 and 5 and 6, so 1, 4 versus 6, 1, 4 and 6, this is 5.8 right, so this value got here. So 5, 6; 7.2 we got this value here, the minimum is 5.8 now, there is a 7th object is there, let us see how far away the object 7. So, what you have to do; we have to find the minimum distance 1, 4 and 7; 1, 4 and 7 and 5, 7, so the distance between 1, 4 and 7 is 5.8 is this value and 5, 7 is our 7.3, so the minimum is 5.8.

(Refer Slide Time: 19:27)



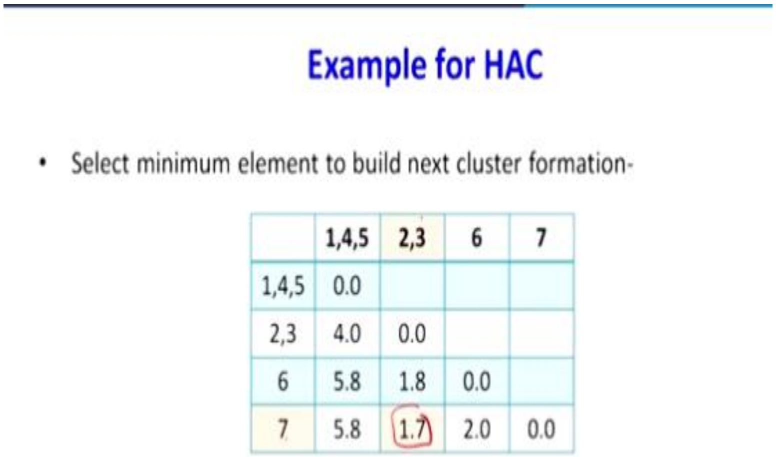
Example for HAC

- Updated distance matrix for the cluster ((1,4), 5)

	1,4,5	2,3	6	7
1,4,5	0.0			
2,3	4.0	0.0		
6	5.8	1.8	0.0	
7	5.8	1.7	2.0	0.0

Again, we will update our distance matrix, so what happen now, you see that the 5 has entered into this cluster because already there is 1, 4 is there, so this is our updated distance matrix.

(Refer Slide Time: 19:43)



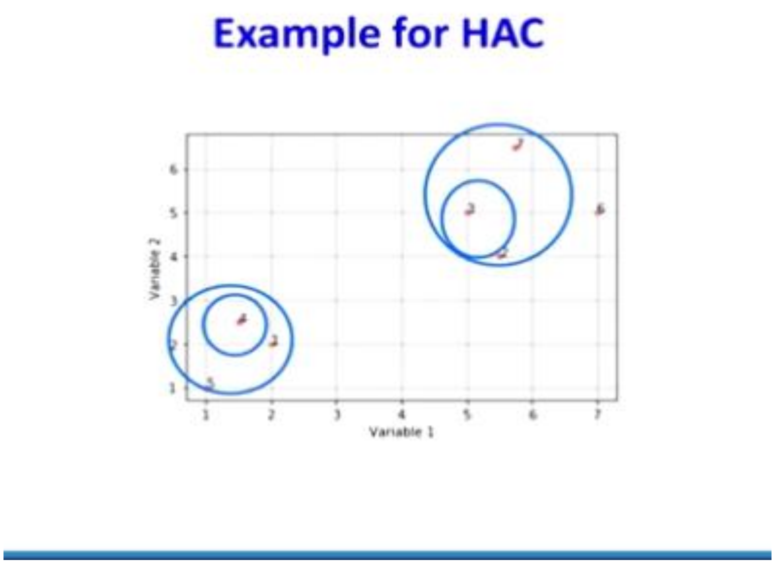
Example for HAC

- Select minimum element to build next cluster formation-

	1,4,5	2,3	6	7
1,4,5	0.0			
2,3	4.0	0.0		
6	5.8	1.8	0.0	
7	5.8	1.7	2.0	0.0

The next step what we are going to do; in the updated distance matrix look at where there is a minimum distance is there, so the 1.7 is a minimum distance. So, what is going to do that the object 7 is going to add in to the cluster 2, 3, so now we will update this distance.

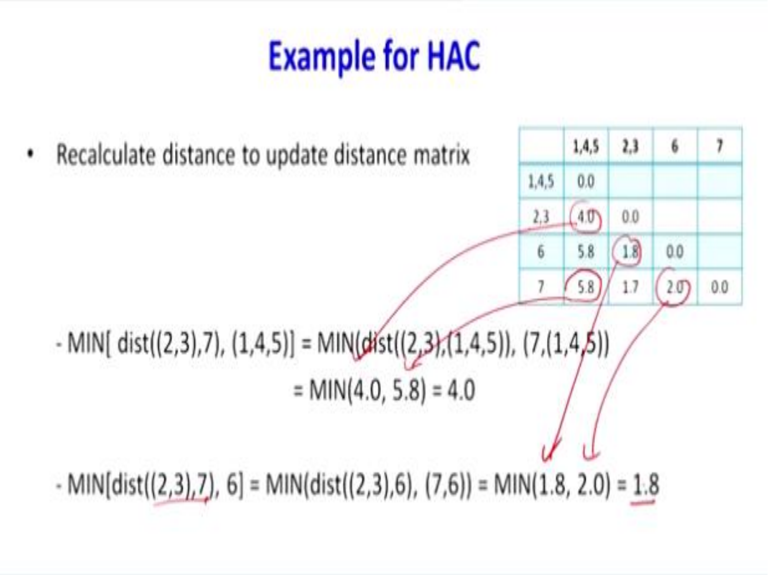
(Refer Slide Time: 20:05)



Example for HAC

What happen, see that this 7 is going to form in that cluster.

(Refer Slide Time: 20:09)



Example for HAC

- Recalculate distance to update distance matrix

	1,4,5	2,3	6	7
1,4,5	0.0			
2,3	4.0	0.0		
6	5.8	1.8	0.0	
7	5.8	1.7	2.0	0.0

- MIN[dist((2,3),7), (1,4,5)] = MIN(dist((2,3),(1,4,5)), (7,(1,4,5))

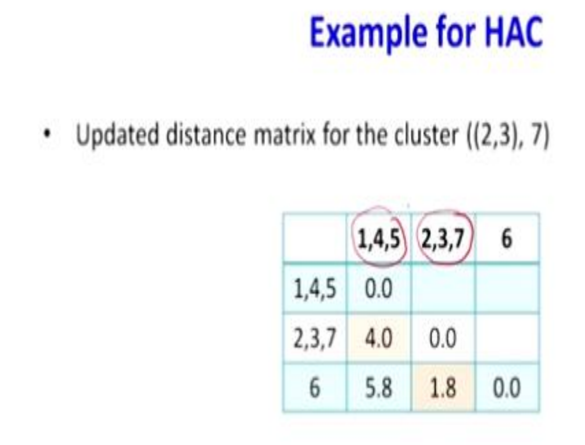
= MIN(4.0, 5.8) = 4.0

- MIN[dist((2,3),7), 6] = MIN(dist((2,3),6), (7,6)) = MIN(1.8, 2.0) = 1.8

Now, what happened, we are going to you see that already there is a cluster, in that 2 object is there because 7 also joined there, so the distance between this cluster versus 1, 4, 5, this is the another cluster. So, what we are going to do; so, 2, 3, 1, 4, 5 and 7, 1, 4, 5, so 2, 3 this 1; 2, 3, 1.45, yes the distance is 2, 3 1.45, this is; this 4 has come here. Second one; the distance between 7, 1, 4, 4, 5, so this one 5.8, so this distance has come here, out of this minimum is 4.

Now, this newly formed clustered versus 6, so here one point is 2, 3 versus 6 and 7, 6, so 2, 3 versus 6, what is a distance; 2, 3 versus 6 this is 1.8, so that distance is came here, so between 7 and 6, the distance is 2, so that distance has come here. Now, the minimum is 1.8, we cannot go next one because 7 is already gone in to that cluster.

(Refer Slide Time: 21:38)



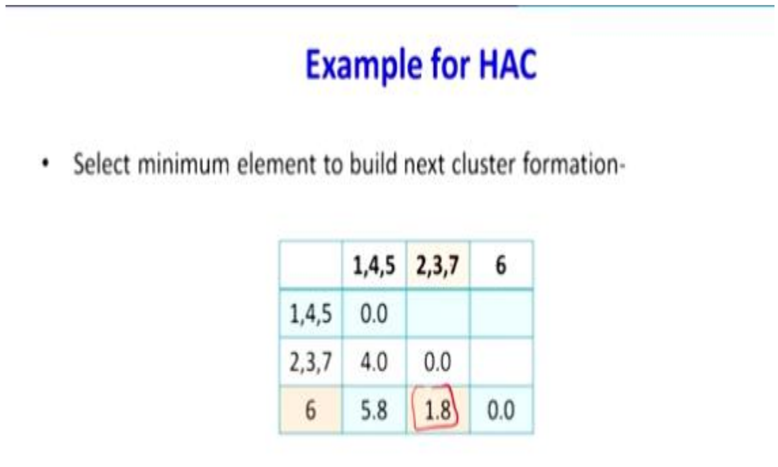
Example for HAC

- Updated distance matrix for the cluster ((2,3), 7)

	1,4,5	2,3,7	6
1,4,5	0.0		
2,3,7	4.0	0.0	
6	5.8	1.8	0.0

Now, again we will update that now, in the updation you see that we have formed 2 clusters, in that 1, 4, 5 is one group, 2, 3, 7 is another group, this is updated distance matrix. This value 4 we got from here, this 1.8 we got from here.

(Refer Slide Time: 22:01)



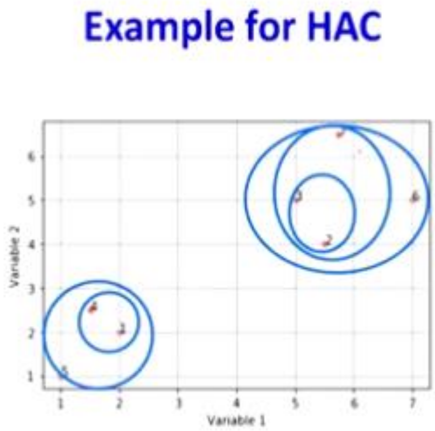
Example for HAC

- Select minimum element to build next cluster formation-

	1,4,5	2,3,7	6
1,4,5	0.0		
2,3,7	4.0	0.0	
6	5.8	1.8	0.0

Now, in the next stage select minimum element to build the next cluster formation, so after this the minimum value is 1.8, this gives the minimum distance between 6 and the cluster 2, 3, 7, so what is going to do now; the 7 is going to join with this cluster where 2, 3, 7 is there.

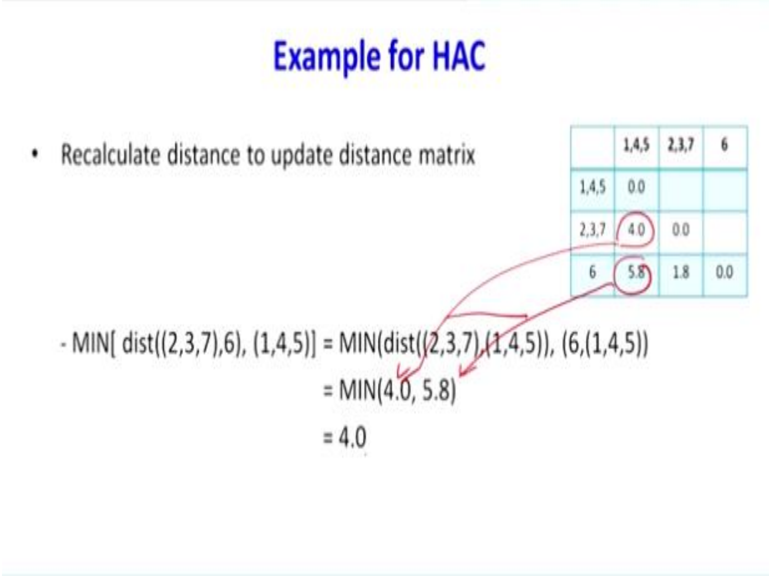
(Refer Slide Time: 22:24)



Example for HAC

See that it is gone to 2, 3, 7.

(Refer Slide Time: 22:26)



Example for HAC

- Recalculate distance to update distance matrix

	1,4,5	2,3,7	6
1,4,5	0.0		
2,3,7	4.0	0.0	
6	5.8	1.8	0.0

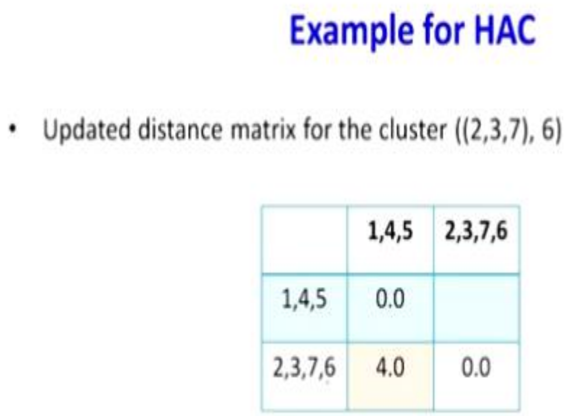
- MIN[dist((2,3,7),6), (1,4,5)] = MIN(dist((2,3,7),(1,4,5)), (6,(1,4,5))

= MIN(4.0, 5.8)

= 4.0

Now, recalculate the distance to update the distance matrix, so the distance between 2, 3, 7, 6 versus 1, 4, 5, so what you have to do the distance is 2, 3, 7 versus 1.45, you have to find out this distance, so 2, 3, 7, 1.45 the minimum distance is that one we brought it here. The next one 6, 1, 4.5, so 6, 1.45 that is a 5.8, so in that the minimum distance is 4.

(Refer Slide Time: 23:04)



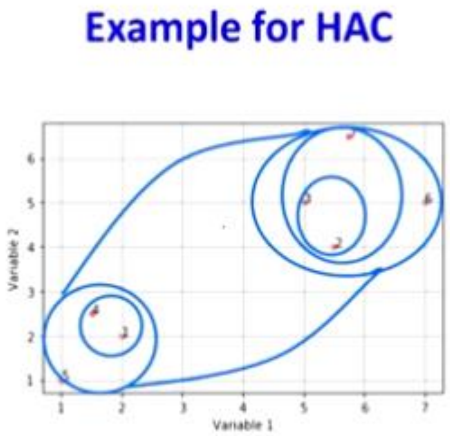
Example for HAC

- Updated distance matrix for the cluster ((2,3,7), 6)

	1,4,5	2,3,7,6
1,4,5	0.0	
2,3,7,6	4.0	0.0

This is our updated matrix, now what happened in that the minimum value is 4, so what happened this 2, 3, 7, 6 will join with 1, 4, 5.

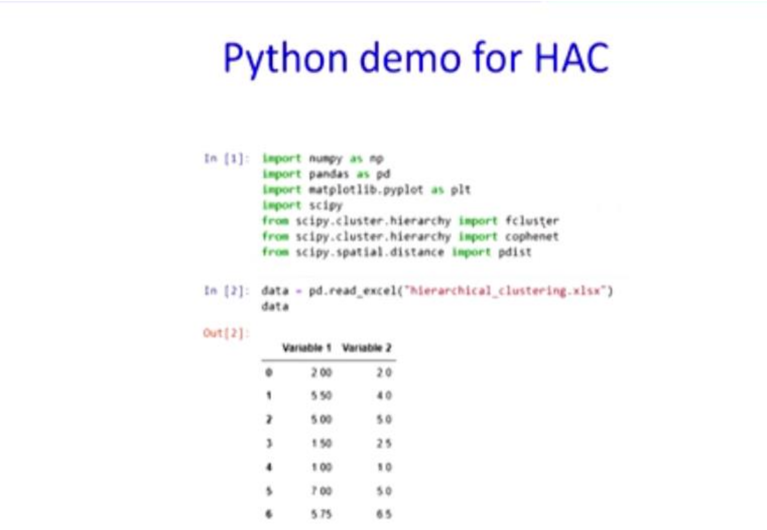
(Refer Slide Time: 23:17)



Example for HAC

So that is nothing but the everything is joined together.

(Refer Slide Time: 23:22)

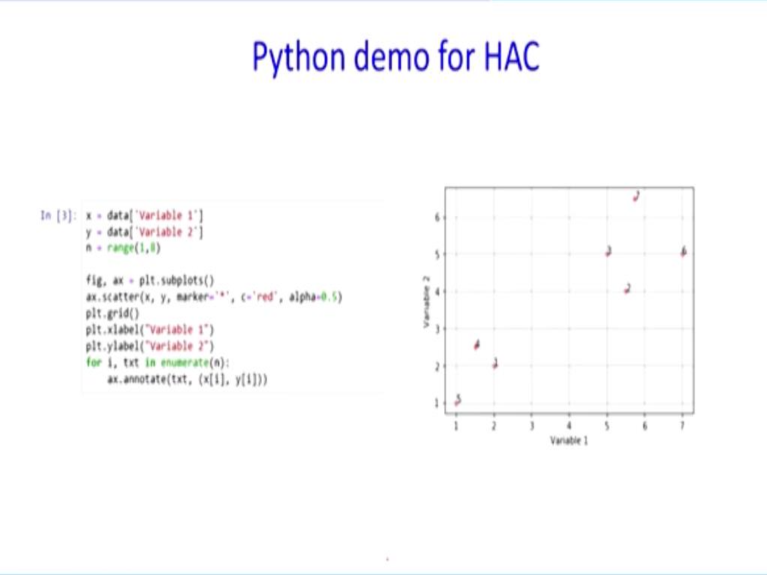


Python demo for HAC

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import scipy
        from scipy.cluster.hierarchy import fcluster
        from scipy.cluster.hierarchy import cophenet
        from scipy.spatial.distance import pdist

In [2]: data = pd.read_excel("hierarchical_clustering.xlsx")
        data

Out[2]:
           Variable 1  Variable 2
        0        2.00         2.0
        1        5.50         4.0
        2        5.00         5.0
        3        1.50         2.5
        4        1.00         1.0
        5        7.00         5.0
        6        5.75         6.5
```

Now, let us see how to do this agglomerative hierarchical clustering method with the help of python, so we have imported the data; import numpy as np, import pandas as pd, import matplotlib dot pyplot as plt, import scipy. From scipy dot cluster dot hierarchy import fcluster, from scipy dot cluster dot hierarchy import cophenet, from scipy dot spatial dot disatnce import pdist.

(Refer Slide Time: 24:07)

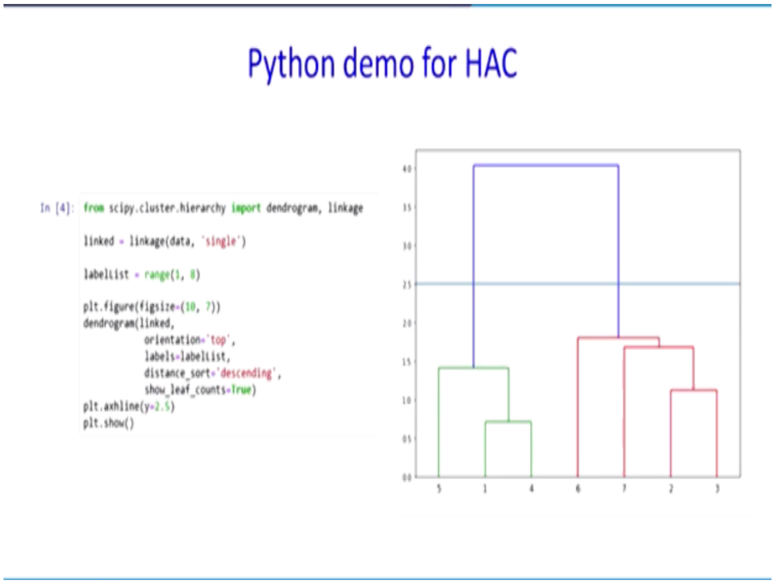


Python demo for HAC

```
In [3]: x = data['Variable 1']
        y = data['Variable 2']
        n = range(1,8)

        fig, ax = plt.subplots()
        ax.scatter(x, y, marker='*', c='red', alpha=0.5)
        plt.grid()
        plt.xlabel("Variable 1")
        plt.ylabel("Variable 2")
        for i, txt in enumerate(n):
            ax.annotate(txt, (x[i], y[i]))
```

So, I have data in hierarchical clustering, so what happened this was our data set, for this data set, we have plotted the 2 dimensional picture, so in that all the objects are displayed.

(Refer Slide Time: 24:14)



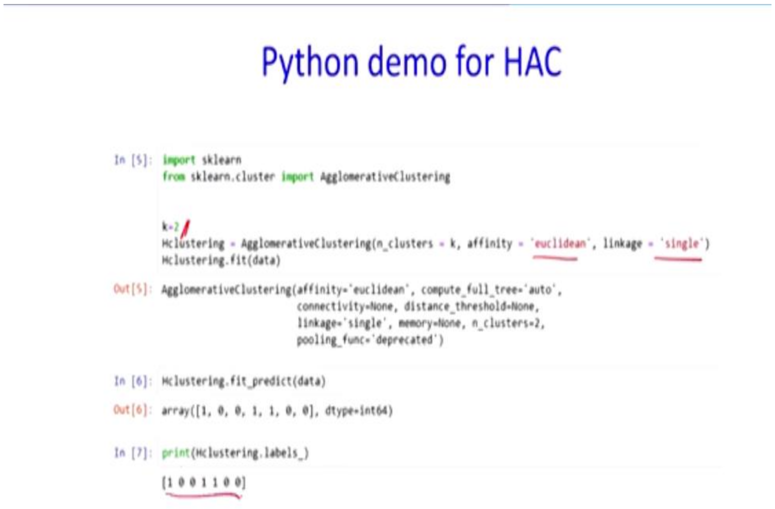
Python demo for HAC

```
In [4]: from scipy.cluster.hierarchy import dendrogram, linkage

        linked = linkage(data, 'single')

        labellist = range(1, 8)

        plt.figure(figsize=(10, 7))
        dendrogram(linked,
                   orientation='top',
                   labels=labellist,
                   distance_sort='descending',
                   show_leaf_counts=True)
        plt.axhline(y=2.5)
        plt.show()
```

Now, when you run this comment that is from scipy dot cluster dot hierarchy import dendrogram, linkage, you will get a this kind of pictures, So, what is happening you see that this is the way 1 and 4 are joining together this stage, after that along with 1 and 4, the 5 is joined, initially 2 and 4; 2 and 3 is a joined, so along with 2 and 3, 7 is joined, after sometime along with the 7, 2, 3, 6 also joined.

At the end, see that the blue line which says all are forming one clustering, this picture shows the dendrogram, so for that from scipy dot cluster dot hierarchy import dendrogram, linkage, so linked equal to linkage data, single, we are going to have single linkage, the label is this, range 1 to 8 that is a figure size. So, when you run that you are getting the dendrogram.

(Refer Slide Time: 25:17)

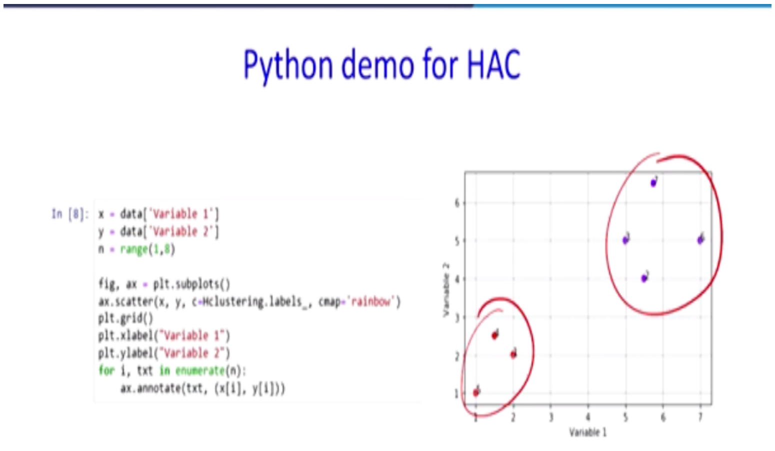


Python demo for HAC

```
In [5]: import sklearn
        from sklearn.cluster import AgglomerativeClustering


        k=2
        Hclustering = AgglomerativeClustering(n_clusters = k, affinity = 'euclidean', linkage = 'single')
        Hclustering.fit(data)

Out[5]: AgglomerativeClustering(affinity='euclidean', compute_full_tree='auto',
                                connectivity=None, distance_threshold=None,
                                linkage='single', memory=None, n_clusters=2,
                                pooling_func='deprecated')

In [6]: Hclustering.fit_predict(data)

Out[6]: array([1, 0, 0, 1, 1, 0, 0], dtype=int64)

In [7]: print(Hclustering.labels_)

        [1 0 0 1 1 0 0]
```

Now, here what it shows that, here we have entered k equal to 2, you look at this value from sklearn; from sklearn dot cluster import AgglomerativeClustering, when you put k equal to 2, we are going to say Euclidean distance and single linkage, so when you; k equal to 2, so the cluster name is into 2 category; one is 0 is one group, 1 is another group, so this was the labels; 1, 0.

(Refer Slide Time: 25:46)

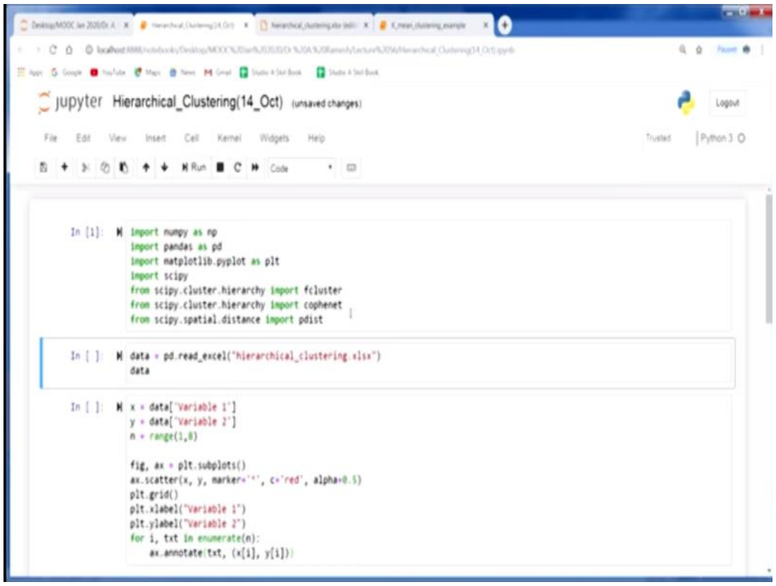


Python demo for HAC

```
In [8]: x = data['Variable 1']
        y = data['Variable 2']
        n = range(1,8)

        fig, ax = plt.subplots()
        ax.scatter(x, y, c=Hclustering.labels_, cmap='rainbow')
        plt.grid()
        plt.xlabel("Variable 1")
        plt.ylabel("Variable 2")
        for i, txt in enumerate(n):
            ax.annotate(txt, (x[i], y[i]))
```

So, when you run this, you see that there are 2 clustering, so this is forms 1 cluster, this is form another cluster, suppose if you write k equal to 3 here, you may get with 3 clusters now, I am going to the python demo for doing this agglomerative hierarchical clustering.

(Refer Slide Time: 26:11)

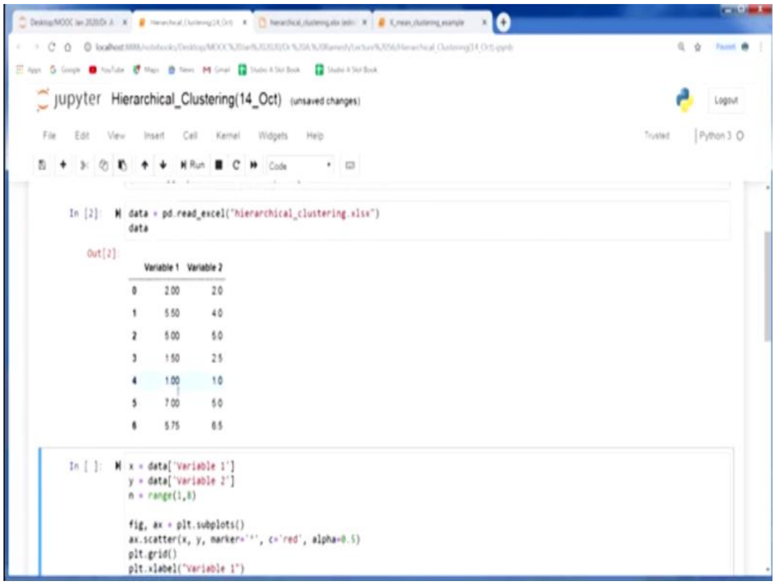


jupyter Hierarchical_Clustering(14_Oct) (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [1]:  import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import scipy
         from scipy.cluster.hierarchy import fcluster
         from scipy.cluster.hierarchy import cophenet
         from scipy.spatial.distance import pdist

In [ ]:  data = pd.read_excel("hierarchical_clustering.xlsx")
         data

In [ ]:  x = data['Variable 1']
         y = data['Variable 2']
         n = range(1,8)

         fig, ax = plt.subplots()
         ax.scatter(x, y, marker='*', c='red', alpha=0.5)
         plt.grid()
         plt.xlabel("Variable 1")
         plt.ylabel("Variable 2")
         for i, txt in enumerate(n):
             ax.annotate(txt, (x[i], y[i]))
```

Now, I am going to show how to do agglomerative hierarchical clustering in python, so import this necessary library; I an running this, I have stored the data in the file name hierarchical underscore clustering.

(Refer Slide Time: 26:26)

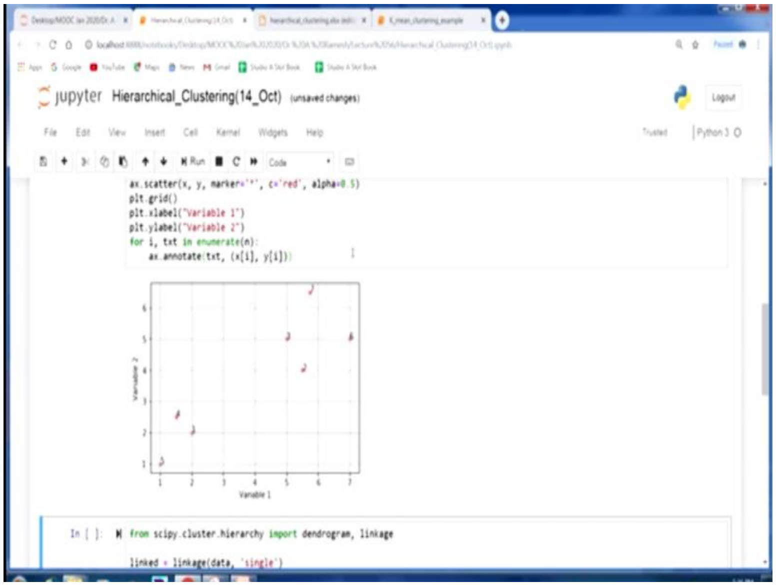


jupyter Hierarchical_Clustering(14_Oct) (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
In [2]:  data = pd.read_excel("hierarchical_clustering.xlsx")
         data

Out[2]:
            Variable 1  Variable 2
         0        2.00         2.0
         1        5.50         4.0
         2        5.00         5.0
         3        1.50         2.5
         4        1.00         1.0
         5        7.00         5.0
         6        5.75         6.5

In [ ]:  x = data['Variable 1']
         y = data['Variable 2']
         n = range(1,8)

         fig, ax = plt.subplots()
         ax.scatter(x, y, marker='*', c='red', alpha=0.5)
         plt.grid()
         plt.xlabel("Variable 1")
```

So, this is our data, for this I am going to do this hierarchical clustering.

(Refer Slide Time: 26:33)



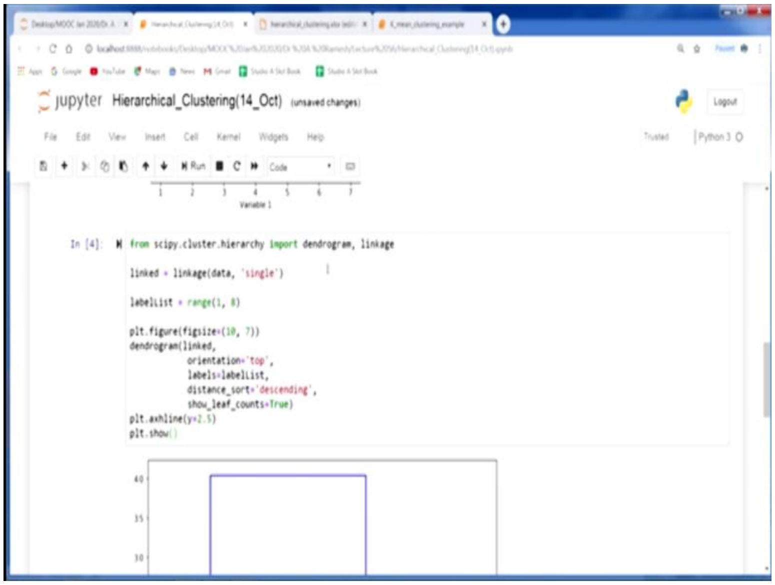
jupyter Hierarchical_Clustering(14_Oct) (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Variable 2
Variable 1

```
ax.scatter(x, y, marker='*', c='red', alpha=0.5)
plt.grid()
plt.xlabel("Variable 1")
plt.ylabel("Variable 2")
for i, txt in enumerate(n):
    ax.annotate(txt, (x[i], y[i]))
```

```
In [ ]:  from scipy.cluster.hierarchy import dendrogram, linkage

         linked = linkage(data, 'single')
```

So, first I am going to show the scatterplot, in that it is showing all the objects, while looking at the object itself, you see that if you go for 2 clustering that will be good.

(Refer Slide Time: 26:46)

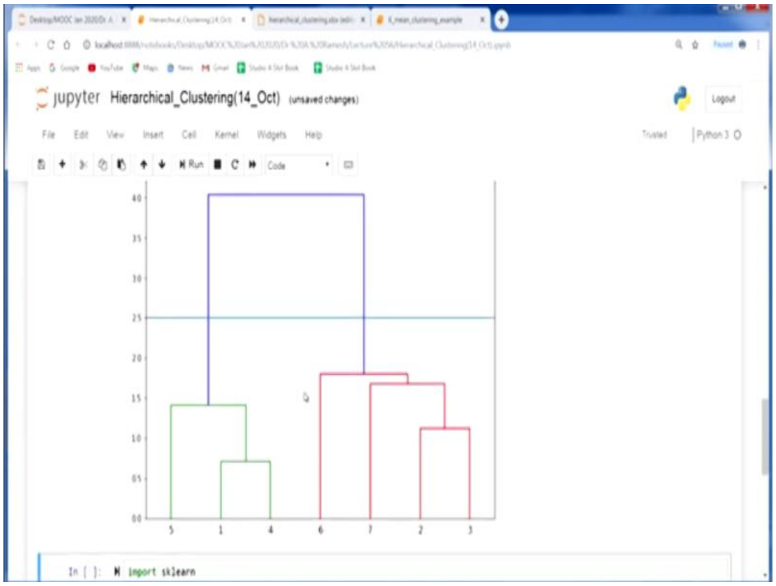


jupyter Hierarchical_Clustering(14_Oct) (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Variable 1
40
35
30

```
In [4]:  from scipy.cluster.hierarchy import dendrogram, linkage

         linked = linkage(data, 'single')

         labelList = range(1, 8)

         plt.figure(figsize=(10, 7))
         dendrogram(linked,
                     orientation='top',
                     labels=labelList,
                     distance_sort='descending',
                     show_leaf_counts=True)
         plt.axhline(y=2.5)
         plt.show()
```

So, what we are going to do; we are going to do the single linkage, here I am going to show the figure size here also, so this picture you know we are going to get the dendrogram.

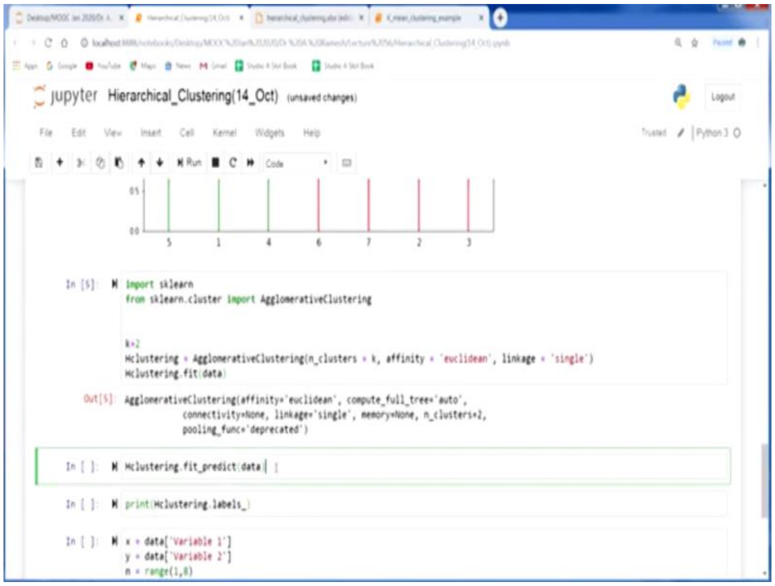
(Refer Slide Time: 26:56)



```
In [ ]:  import sklearn
```

So, this is the dendrogram, so 2 and 3 is forming one cluster, out of that this 7 is joining, out of that 6 is joining. Here at level 1, it is 1 and 4 is joining, then 5 is joining there at the end, it is going to; all are going to be in the same cluster.

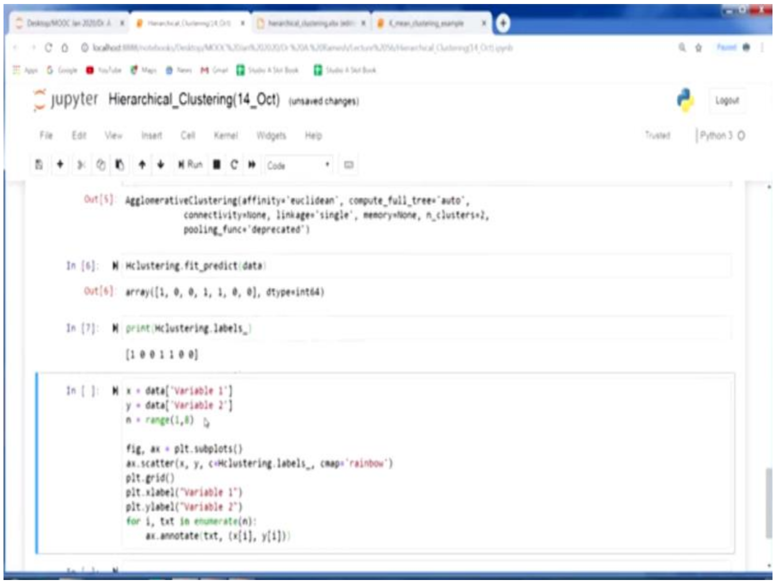
(Refer Slide Time: 27:12)



```
In [5]:  import sklearn
         from sklearn.cluster import AgglomerativeClustering


         k=2
         Hclustering = AgglomerativeClustering(n_clusters = k, affinity = 'euclidean', linkage = 'single')
         Hclustering.fit(data)

Out[5]:  AgglomerativeClustering(affinity='euclidean', compute_full_tree='auto',
                     connectivity=None, linkage='single', memory=None, n_clusters=2,
                     pooling_func='deprecated')

In [ ]:  Hclustering.fit_predict(data)

In [ ]:  print(Hclustering.labels_)

In [ ]:  x = data['Variable 1']
         y = data['Variable 2']
         m = range(1,8)
```

Now, from sklearn dot cluster import AgglomerativeClustering, suppose we will start with 2, so 2, so run this, let us see hierarchical clustering, how we are doing, see that there are 2 clustering, it is one is named as 1, another one is 0.

(Refer Slide Time: 27:38)

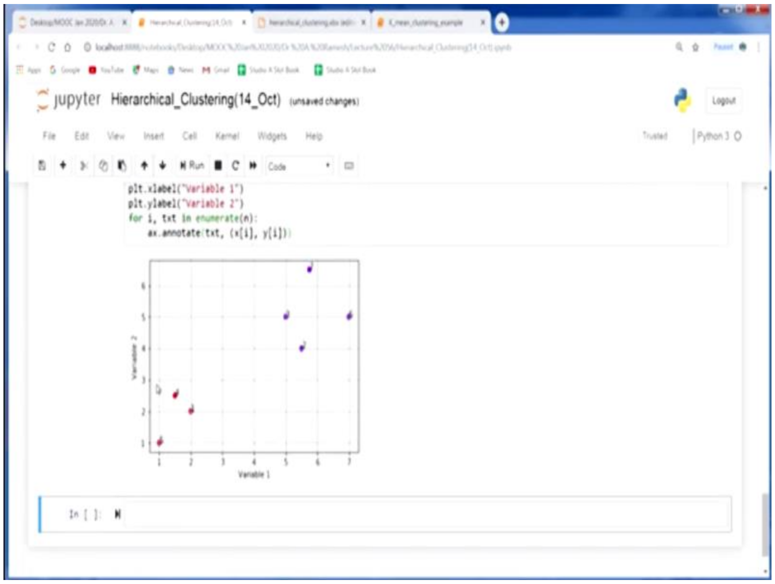


jupyter Hierarchical_Clustering(14_Oct) (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
Out[5]: AgglomerativeClustering(affinity='euclidean', compute_full_tree='auto',
                    connectivity=None, linkage='single', memory=None, n_clusters=2,
                    pooling_func='deprecated')

In [6]: Hclustering.fit_predict(data)

Out[6]: array([1, 0, 0, 1, 1, 0, 0], dtype=int64)

In [7]: print(Hclustering.labels_)

        [1 0 0 1 1 0 0]

In [ ]: x = data['Variable 1']
        y = data['Variable 2']
        n = range(1,8)

        fig, ax = plt.subplots()
        ax.scatter(x, y, c=Hclustering.labels_, cmap='rainbow')
        plt.grid()
        plt.xlabel("Variable 1")
        plt.ylabel("Variable 2")
        for i, txt in enumerate(n):
            ax.annotate(txt, (x[i], y[i]))
```

So, let us see what are the labels so, this is labels; 1, 0, 0, 1, 1, 0, 0.

(Refer Slide Time: 27:48)

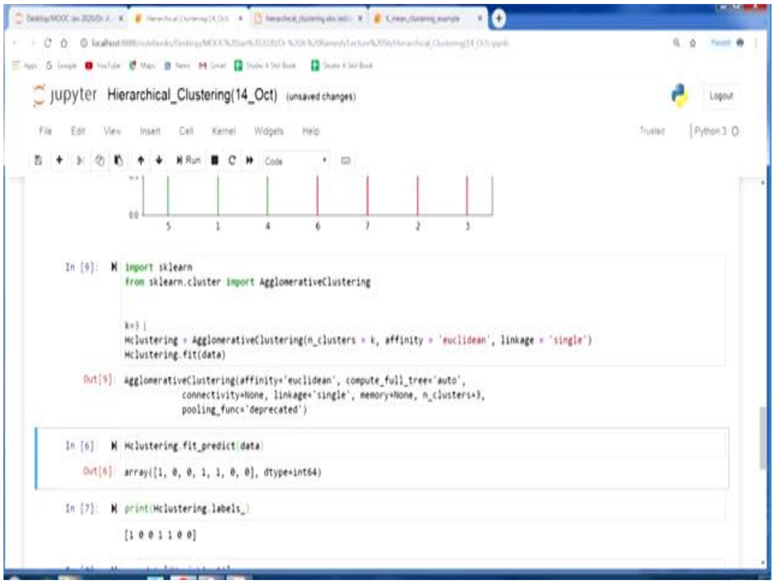


jupyter Hierarchical_Clustering(14_Oct) (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
1
2
3
4
5
6
Variable 2
1
2
3
4
5
6
7
Variable 1

```
        plt.xlabel("Variable 1")
        plt.ylabel("Variable 2")
        for i, txt in enumerate(n):
            ax.annotate(txt, (x[i], y[i]))

In [ ]:
```

So, now if you run this, what you are getting; now there are 2 clusters, so the red colour point, say 1 cluster, the purple is another cluster. Suppose, if we go for k equal to 3, let us see what kind of answer we are getting.

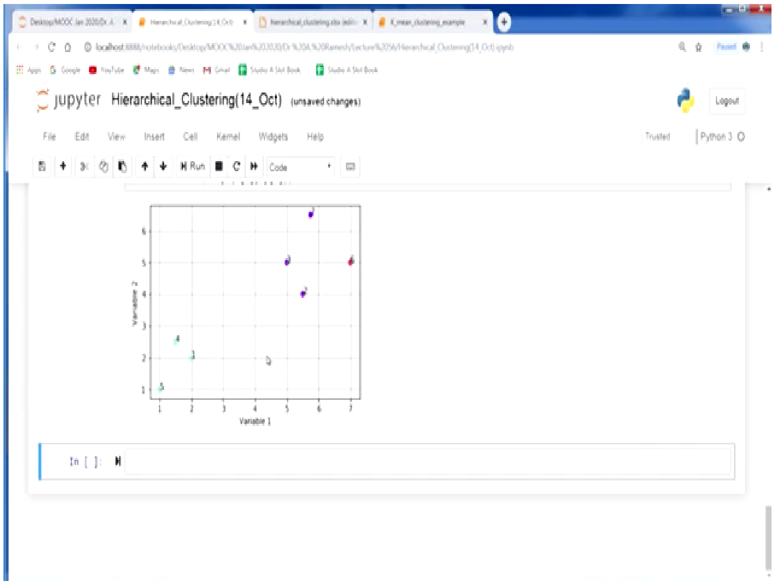
(Refer Slide Time: 28:04)


jupyter Hierarchical_Clustering(14_Oct) (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
0.0
5
1
4
6
7
2
3

```
In [9]:  import sklearn
         from sklearn.cluster import AgglomerativeClustering


         k=3
         Hclustering = AgglomerativeClustering(n_clusters = k, affinity = 'euclidean', linkage = 'single')
         Hclustering.fit(data)

Out[9]:  AgglomerativeClustering(affinity='euclidean', compute_full_tree='auto',
                    connectivity=None, linkage='single', memory=None, n_clusters=3,
                    pooling_func='deprecated')

In [6]:  Hclustering.fit_predict(data)

Out[6]:  array([1, 0, 0, 1, 1, 0, 0], dtype=int64)

In [7]:  print(Hclustering.labels_)

         [1 0 0 1 1 0 0]
```

Suppose, if we go for k equal to 3, again you run it, you see that the level is 0, 1, 2, again you run it, level is see that.

(Refer Slide Time: 28:18)


jupyter Hierarchical_Clustering(14_Oct) (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Variable 2
Variable 1
1
2
3
4
5
6
7
In [ ]:

Now, if you run this, you see that there are 3 cluster; green, purple and red because red is only one cluster, there is only one element, so the optimal number of cluster for this kind of data set is k equal to 2 that is a purpose we can visualise how the cluster is formed and quality of cluster also. In this lecture, what I have done, I have started agglomerative hierarchical algorithm with the help of a numerical example.

In that example, I have first I found the distance matrix, after finding the distance matrix, I formed a cluster wherever there is a minimum value is there, I connected that 2 objects, then I have updated the distance matrix, again I have gone to where there is a minimum point is there, so that point and that objects are clubbed together. At the end, for the same data set, I have explained how to do python programming.

And I also shown how the result is appearing, so here the number of cluster I have initially started with k equal to 2, then again I changed k equal to 3 but when I changed k equal to 3, I get some other result that is not looking good, so I kept only k equal to 2 is the right number of clusters, so optimal number of clusters.