

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
NPTEL
NPTEL ONLINE CERTIFICATION COURSE
Business Analytics & Data Mining Modeling
Using R – Part II
Lecture-09
ClusterAnalysis– Part V
With
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology Roorkee

Business Analytics & Data Mining Modeling Using R - Part II

Lecture-09 Cluster Analysis-Part V



With
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology Roorkee

Welcome to the course Business Analytics and Data Mining Modeling Using R – Part 2, so in previous few lectures we have been discussing cluster analysis, so specifically in the previous lecture we stopped at this point and we wanted to discuss Ward’s method, so which is also part of hierarchical agglomerative clustering approaches, so let’s start our discussion with this particular method.

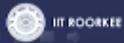
Cluster Analysis

- Results of HAC using either single or complete linkage
 - Depend only on the order of inter-record distances and not actual values
- Ward's Method
 - Also a HAC method
 - Instead of finding two closest observations to form cluster, this method selects the cluster formation which results in smallest incremental loss of information

So as I said Ward's method is slightly different from what we have discussed so far, other approaches that we have discussed so far within the HAC, hierarchical agglomerative clustering you know methods, so how it is different so let's see, so the main point of difference is mentioned here in the slide, that instead of finding two closest observations to form cluster, this method selects the cluster formation which results in smallest incremental loss of information, so as you can see in other methods you know and the distance metrics that we use in HAC, we typically look to identify the two closest records, and later on two closest you know clusters, two you know to build our clusters, however this particular method Ward's method it actually you know select the cluster formation, that means the cluster which are formed, the clusters which in comparison to the previous formation in comparison with the previous steps it tries to minimize the loss of information, right, so new cluster formation is going to replace the previous the cluster formation, so whether there has been you know a loss of information and to minimize that loss of information is the main idea behind this particular method, Ward's method.

Cluster Analysis

- Ward's Method
 - Main idea being that
Merging of observations to form clusters can be thought of
As
Information about an individual observation being replaced by the
information about the cluster where it will be merged
 - It might lead to loss of information which is measured using
“error sum of squares” (ESS)
 - Measures the difference between individual observations and a group mean



So let's discuss further, so few more details about Ward's methods so you can see here, main idea being that merging of observations to form clusters can be thought of as information about and individual observation being replaced by the information about the cluster where it will be merged, so we can think about an individual observation which will be you know representing a certain piece of information and when it becomes part of, as part of the clustering process when it becomes part of some cluster then the information you know, it's information is going to be replaced by the information about cluster, so because of this there could be some loss of information, so that is measured and then minimized in this particular method, the same points are being conveyed here.

Now how this loss of information can be measured, so the metric the popular common metric that is typically used is mentioned here, error sum of squares, ESS, so what it does it measures the difference between individual observations and a group mean, so you know when a particular observation becomes you know part of a cluster then of course the cluster mean is, you know that group mean might be representing you know the information about the cluster, so the difference between this individual observation and group mean so that is being measured by error sum of squares.

Cluster Analysis

- For example: consider following univariate data (10 observations, one variable)

2, 6, 5, 6, 2, 2, 2, 0, 0, 0

Mean = 2.5

If we cluster all of these observations into a single group, loss of information would be

$$ESS = (2-2.5)^2 + (6-2.5)^2 + (5-2.5)^2 + \dots + (0-2.5)^2 = 50.5$$

If we cluster these observations into following four groups:

(0, 0, 0), (2, 2, 2, 2), (5), and (6, 6), ESS = 0



So let's understand this whole thing with a simple example that we have in the slide, so let's consider a following univariate data, so 10 observations 1 variable, so we have 10 observations that we want to cluster and then we have just 1 variable, right, so these you know values could be, values for this particular just one variable could be this one 2, 6, 5, 6, 2, 2, 2, and then 0, 0 and 0, so these are the 10 observations that we have, so if we look for the mean value of these observations that is going to be 2.5, so if we cluster all of these observations into a single group, right, so single group so this group will have a mean value of 2.5, so all these observations, instead of representing their individual information now they are going to be represented by this mean value, because they are going to be part of this cluster, right, so how we are going to apply this particular metric error sum of the squared that we talked about, so loss of information can be computed in this fashion, for first observation you can see $2 - 2.5$, so this is square of this value then for second observation $6 - 2.5$ square, then for third observation $5 - 2.5$ is square and so on. So in this fashion we can compute the error sum of squares which comes out to be 50.5.

So the loss of information can be expressed using this particular number in this fashion, this metric error sum of squares and this is the loss of information that we have, however so as I said we had 10 observations and you know we wanted, if we you know wanted to club them in one group and that group mean we understood and we calculated the ESS value. Now let's change this you know cluster formation, now if we want to cluster these observations into following four groups, now you see the first group containing all the values you know which are 0, then the second cluster containing all the observations which are having value of 2, and then this singleton cluster having just one observation of you know value 5, and then two observations of having value 6, right, so we have these four observations. If you look at key clusters you know this particular cluster configuration you can see we are clubbing all the observations which are having the same value, therefore that mean value is also going to be the same.

Now we have four clusters right now, and we apply the same formula ESS here then our ESS value is going to be, come out to be 0, so therefore loss of information as per this metric is going to be 0, so probably the next step in this you know, in this particular method, Ward's

method is going to be this one, so these observation if they are you know the cluster analysis using the Ward's approach is to be applied then probably this is going to be the next step, because we incremental loss of information is minimal in this particular cluster, for this particular cluster formation, so you can understand how this is, this approach is different from the earlier approaches, so in the earlier approaches under HAC that we talked about, we look to identify the closest observation so that we can create a cluster, right.

Cluster Analysis

- Ward's Method
 - Tendency to form convex clusters of approx. equal sizes
 - E.g., employee performance ratings, student grades, customer segments
- Display of clustering results
 - By a treelike diagram called Dendrogram
 - Summarizes the clustering process

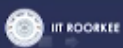
Here in this case Ward's method we'll look to identify cluster formation which is going to have the minimum loss of information as per the metric that we are going to use, few more important points about Ward's method, so for example the kind of shapes that are produced in the shapes of clusters, produced cluster so we can see here tendency to form convex clusters of approximately equal sizes, so why this happen also you can easily you know see from the previous example that we have discussed, so in the previous example we saw that the particular cluster formation which you know, which were combining you know value, visual combining observation with the you know, you know same values, so there is the loss of information about 0, so therefore from that you know observation we can see that this particular method would actually lead into convex clusters and the sizes would also be approximately roughly might be equal, so where this particular kind of method could be suitable, for example employ performance ratings, so students creates customer segments, so these are some of the problems, some of the areas where this particular method can be applied and could be more suitable because the way the performance ratings happen, the way rating happens and sometimes you might also want to have customer segments of equal sizes and which are you know, you know we would look to minimize the loss of information as advocated by Ward's method, then this is more suitable approach.

Till now under HAC we have discussed the more common approaches, now after discussing you know how the clustering happens let's understand few more details about the cluster analysis, so how the clustering measures are displayed, so in this particular you know technique also just like in the classification and regression keys that we discussed in the previous course we have a tree like diagram here which is called Dendrogram, this particular Dendrogram, this

particular you know tree diagram or structure is actually used to display the clustering results and it also in a way summarizes the clustering process, how it does this particular thing we will see in this lecture.

Cluster Analysis

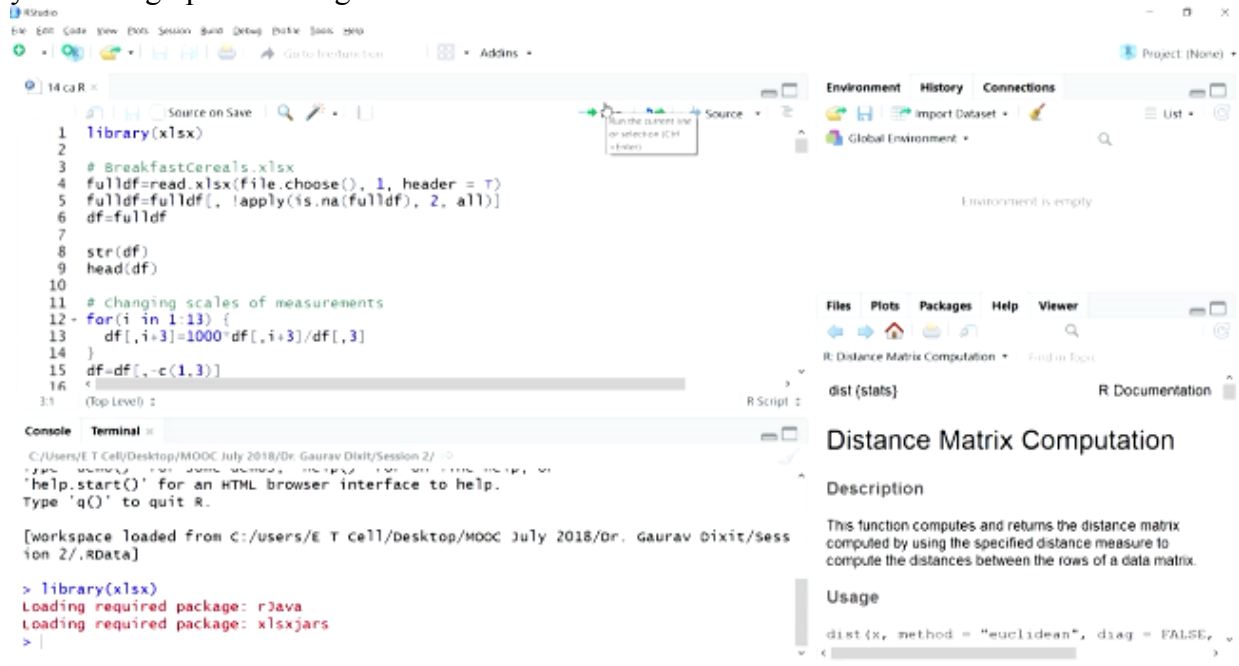
- Dendogram
 - Bottom level represents all the observations
 - When two observations are merged, vertical lines representing the distance between them are depicted to join together
 - It can be used to find out the no. of observations in each cluster if no. of clusters to be created is specified
 - By sliding a horizontal line up and down to achieve the no of intersections equal to the no. of desired clusters
- Open RStudio



So before we move ahead let's understand this particular you know this particular graphic Dendogram, so what this is about? So in Dendogram this is a tree like structure so we will see when we do an exercise in R, we'll see how this Dendogram looks like, but we'll start with, it is a tree like structure, tree like diagram and the bottom level, bottom level of this particular you know Dendogram, this particular tree like structure are going to represent all the observations, because we are you know, we are talking about, right now we are talking about the HAC hierarchical agglomerative clustering where we start with all the observations you know being considered as clusters, right, so bottom level of the tree is going to represent all the observations.

When two observations are merged you know vertical lines we'll see that through an exercise in R that vertical lines will represent the distance between the observation which are going to be merged, and then they would be depicted to, depicted to joint together so we might have you know at the bottom level will have certain number of observation and you know depending on their distance, if they are found to be closest they are going to be merged, so that in the tree diagram you would see that joining happening, and that is how this tree structure is going to be created bottom up, right, so because of this bottom up process all the observation then joining and then you know this structure will keep on building till we end up with just you know, just like in tree diagram just one note so where in this case we say just one cluster having all the observation, so the whole process would also be depicted in this particular diagram, so it can also be used this, Dendogram can also be used to find out the number of observations in each cluster, if number of clusters to be created is specified, so if we have you know certain expectation that number of clusters in a particular dataset, in a particular problems are you know, are required to be you know 6 or 5, so this particular Dendogram this will also allow us to you know get those number of clusters and we'll also able to see through this Dendogram, which are the observation, which are going to be part of this clusters, so for each cluster will get

to know the observation which are going to be part of those clusters and you know how those you know clusters are going to be created, so all that would be clearly visible in this particular you know graphic Dendrogram.



So let's open R studio and we'll do a small exercise to find out more about what we have discussed, so let's go through some of the code that we have, understood in previous lecture, let's import the dataset. So few things, few lines of code we'll just go through so that we are able to reach to the point what we have discussed so far, so all these lines of code we have discussed in previous lecture so we won't discuss them we'll just go through this part of code, and we'll just run through some of these things.

So in the previous lecture we had gone through this, so where we created, where we computed the centroids for two clusters, so all this we have gone through, so let's quickly go through this, so this part also we covered where we can see how two observations are merged and how the distance metrics where computed.

Now so we have reached to this point so till now in the previous lecture the exercise that we have been doing, we selected 5 cereals from the dataset that we have out of 35 observation we picked 5, first 5 observations, first 5 cereals and we went through an exercise in R and we applied our hierarchical agglomerative clustering, now what we are going to do is will use the full dataset and see how some of these approaches, HAC approaches can be applied.

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for computing a distance matrix. The code includes:


```

113 DM3
114 min(DM3[lower.tri(DM3, diag = F)])
115 # Next merger: SKMH-CC and CD will lead to 3x3 distance matrix
116
117 #####
118 # Using full dataset for Hierarchical Agglomerative Clustering
119 head(df)
120
121 # Normalization of all variables
122 dfn=as.data.frame(scale(df[, -c(1)], center = T, scale = T)); head(dfn)
123
124 # Distance matrix
125 DMn=dist(dfn, method = "euclidean"); DMn
126
127
128
129 (Top Level)

```
- Console:** Shows the output of the R script, including a table of normalized data and the resulting distance matrix.

	Customer.Rating
1	4.2
2	3.0
3	4.4
4	4.4
5	4.9
6	3.3

5	0.2666667	0	0.0240000	11.20000
6	NA	0	1.2000000	23.46667
- Environment Pane:** Shows the objects created in the global environment:
 - `DM3`: numeric matrix [1:4, 1:4] with values 0, 1.525, 0.6...
 - `fulldf`: data frame with 35 observations and 17 variables.
 - `DM`: distance matrix of class "dist" atomic [1:10] 1..
 - `DM2`: distance matrix of class "dist" atomic [1:1] 1..
 - `i`: numeric value 4L.
- Help Pane:** Displays the documentation for the `dist` function, titled "Distance Matrix Computation". It includes a description: "This function computes and returns the distance matrix computed by using the specified distance measure to compute the distances between the rows of a data matrix." and a usage example: `dist(x, method = "euclidean", diag = FALSE, ...)`.

Right so let's have a relook at the dataset that we have, so this is the dataset that we have, so in this we can see the variables which we have discussed already, and so now in this, because we are going to use all the observations here so we have 35 observation, 70 variables, so as you know we have discussed in previous lectures, before we move ahead we need to normalize all these variables so that the scale dependency is you know taking care of because typically we use Euclidean you know distance matrix, so for that we need to normalize you know all the variables that we have numerical variables, so this is the code that can be used you can see we are using a scale function, so this particular function we have used before also in previous course as well, and so this will create the normalized scales for us you can see, for 6 observation you can see the normalized values here, so all the variables have been normalized, now with this we can compute our distance matrix so here we can, we are going to have the distance between all possible pair of observations, so we have 35 of them so this matrix 35 by 35 matrix is going to be created, you can see here, so all the distance value that have been computed.

The screenshot shows the RStudio interface. The main editor contains the following R code:

```

120 head(df)
121
122 # Normalization of all variables
123 dfn=as.data.frame(scale(df[,-c(1)], center = T, scale = T)); head(dfn)
124
125 # Distance matrix
126 DMn=dist(dfn, method = "euclidean"); DMn
127
128 # Ward's method
129 mod=hclust(DMn, method="ward.D2")
130
131 plot(mod, xlab = "", main = "", hang = -1, sub = "", frame.plot = T)
132
133 # Single linkage
134 mod1=hclust(DMn, method="single")
135 <
12913 (Top Level) :

```

The Environment pane on the right shows the following objects:

Object	Class	Attributes
DM1	num	[1:4, 1:4] 0 1.525 0.6...
DM3	num	[1:4, 1:4] 0 1.525 0.6...
fulldf	data.frame	35 obs. of 17 variables
DM	class 'dist' atomic	[1:10] 1...
DM2	class 'dist' atomic	[1:1] 1...
DMn	class 'dist' atomic	[1:595] ..

The Help pane on the right shows the documentation for the `dist` function:

Distance Matrix Computation

Description

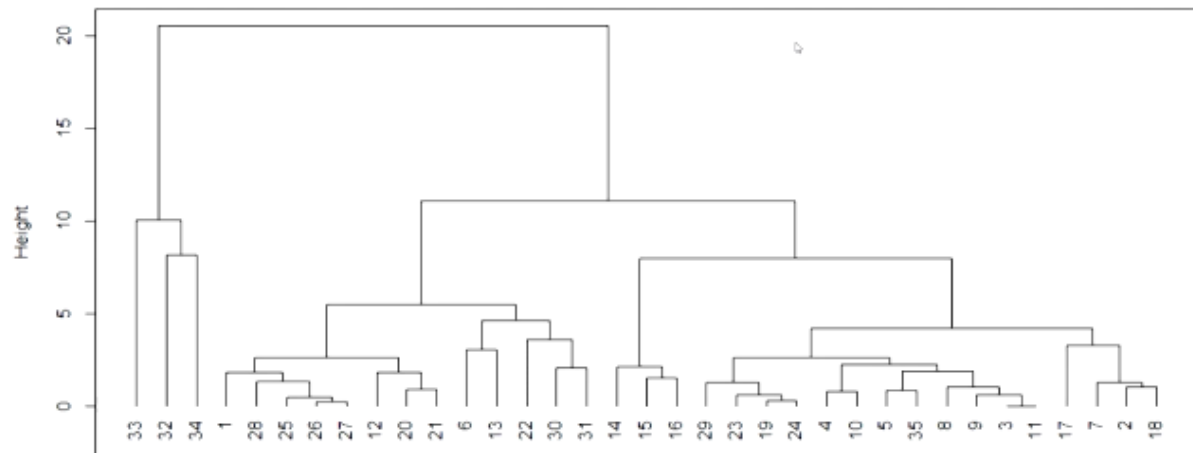
This function computes and returns the distance matrix computed by using the specified distance measure to compute the distances between the rows of a data matrix.

Usage

```
dist(x, method = "euclidean", diag = FALSE, ...)
```

Now to apply some of the methods that we have discussed so far, for example Ward's method single linkage and other methods that we have discussed, you know now we are going to apply each of them on full data set, so you can see we'll start off with the Ward's method so here we are calling this function H clust, so this is for hierarchical clustering and in the first argument is, first argument is the distance matrix that we have just computed, this distance matrix is going to be used for further processing and implementation of these different HAC approaches, so first one we are starting with the Ward approach so there are two version which are available in R, so this is the more latest version which we are going to use here, so let's run this code.

And once this model has been built, we can plot this using plot function and some other arguments I've also been specified and we'll see the Dendrogram for this particular model, so the 35 observation that we have, all the observation that we have in the dataset and all the variables 17 of them have been used, they were first normalized as we you know just, you know saw, and now you know we have just build the model now we are going to see the output using Dendrogram, so you can see here the Dendrogram.



Now this particular Dendrogram has been created using Ward's method, so you can see all 35 observations, all these observations at the bottom level they have been indexed so their names you can see here and so this is on X axis, and on Y axis we have the height so that is going to display at which point you know these observations have been merged, so these heights are also indicating the relative occurrence of these observations joining on mergers, so you can find out probably which observations were merged first, so from this you know vigilance inspection if we are correct, so it seems that observation 3 and 11 were merged first, right, so you can see here the same thing you can also verify using the distance matrix, so in the distance matrix if you see that the distance between these two observations, observation 3 and 11 is happens to be the minimum distance then probably what we just saw through this Dendrogram was correct, because there are few other observations also which are also having very small height, so we can see for example 26, 27 and then the observation 19 and 24, they also seem to be merging you know in the initial part of this clustering process, but it seems that 3 and 11 this merger happens first, so as we talked about Ward's method so instead of finding two closest observations we'll look to compare the cluster formation, the previous one and the new one and you know try to minimize the loss of information, so using this approach this particular cluster, this particular Dendrogram has been created.

Now using this Dendrogram as you can see you know you can see the clusters are nested, right, so eventually we end up with one cluster but if we go down you know from the tree then we can see the nesting of clusters, so if we are going for, for example if we are going to, if we want 6 clusters then we can, then we can, all the time we can draw a horizontal line here, so we can draw a horizontal line here and in this paragraph and the intersections you know the line will intersect this particular you know Dendrogram at certain points and those intersection points will get, will give us the equal number of clusters, and if this particular new line we go up then the number of clusters are going to be fewer, so we might you know have, you know from 6 to 3, just 3 clusters and but interesting thing that is you know we would find that the clusters now would be nested, right.

The screenshot shows an RStudio window with the following R code in the editor:

```

122 # Normalization of all variables
123 dfn=as.data.frame(scale(df[,-c(1)], center = T, scale = T)); head(dfn)
124
125 # Distance matrix
126 DMn=dist(dfn, method = "euclidean"); DMn
127
128 # ward's method
129 mod=hclust(DMn, method="ward.D2")
130
131 plot(mod, xlab = "", main = "", hang = -1, sub = "", frame.plot = T)
132
133 # Single linkage
134 mod1=hclust(DMn, method="single")
135
136 plot(mod1, xlab = "", main = "", hang = -1, sub = "", frame.plot = T)
137
133.1 (Top Level) :

```

The console shows the execution of the code:

```

> # ward's method
> mod=hclust(DMn, method="ward.D2")
> plot(mod, xlab = "", main = "", hang = -1, sub = "", frame.plot = T)

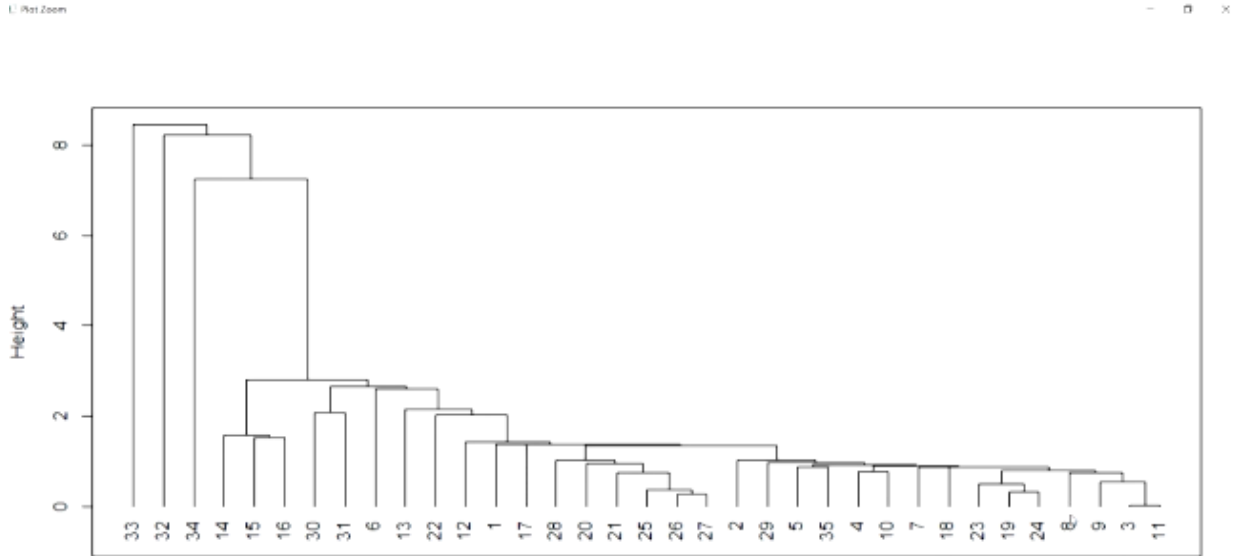
```

The Environment pane shows the following objects:

- Global Environment:
 - dfn: num [1:4, 1:4] 0 1.525 0.6...
 - DM3: num [1:4, 1:4] 0 1.525 0.6...
 - fulldf: 35 obs. of 17 variables
 - mod: List of 7
- Values:
 - DM: class 'dist' atomic [1:10] 1...
 - DM2: class 'dist' atomic [1:1] 1...

A small dendrogram plot is visible in the bottom right corner of the RStudio window.

So let's go back, now let's apply the other approach this is a single linkage approach, so in the single linkage approach as we talked about in the previous lecture we look you know distance between clusters to clusters is computed using the minimum distance between pair of observation, one from each of those clusters this part we have discussed, so we'll just call this function H clust and the method has been specified as single and we'll get this model and then we'll plot the Dendrogram, so let's execute this.



So this is the Dendrogram of this approach, single linkage approach and as you can see this is you know quite different from what we have from Ward's method, again here also we can clearly identify which two observations were merged first, so it seems that 3 and 11, so if you

remember that in Ward's method also these were the two observation ID's that were merged first, so 3 and 11 so these observations seem to be merging first, so from this we can see the approach is different, we are just looking at you know identifying two closest observation and then you know two closest clusters, right, and in this fashion this particular clustering process happens.

The screenshot shows the RStudio environment with the following components:

- Source Editor:** Contains R code for hierarchical clustering:


```

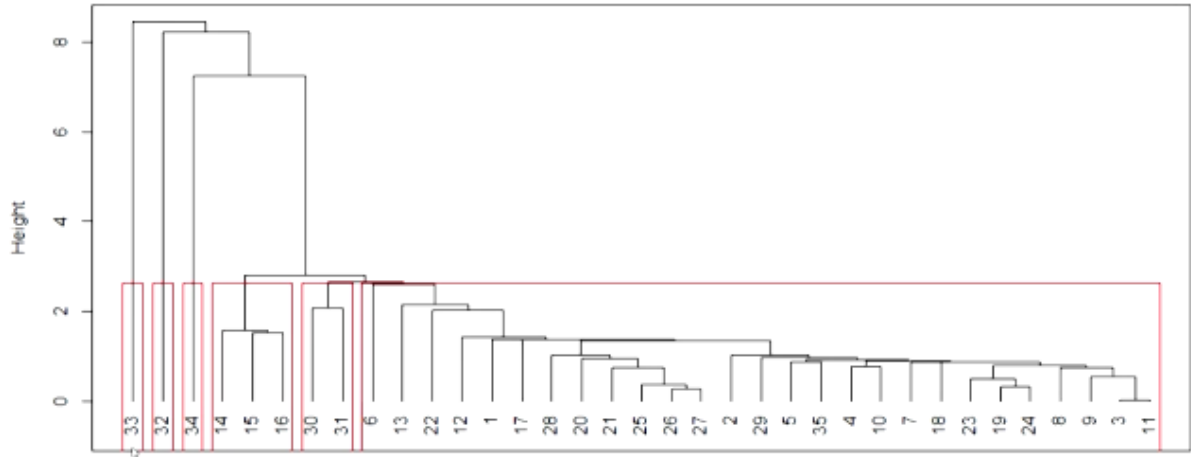
129 mod=hclust(DMn, method="ward.D2")
130
131 plot(mod, xlab = "", main = "", hang = -1, sub = "", frame.plot = T)
132
133 # Single linkage
134 mod1=hclust(DMn, method="single")
135
136 plot(mod1, xlab = "", main = "", hang = -1, sub = "", frame.plot = T)
137
138 # Example: No. of desired clusters=6
139 # To cut a single linkage dendrogram into 6 clusters
140 groups=cutree(mod1, k=6)
141 # To redraw dendrogram with red borders around the 6 clusters
142 rect.hclust(mod1, k=6, border="red")
143
144 <
145
146 (Top Level) :
```
- Environment:** Shows the current environment with variables:

DMn	num [1:4, 1:4]	0 1.525 0.6...
fulldf	35 obs. of 17 variables	
mod	List of 7	
mod1	List of 7	
values		
DM	Class 'dist' atomic fl:101 1...	
- Console:** Shows the execution of the code:

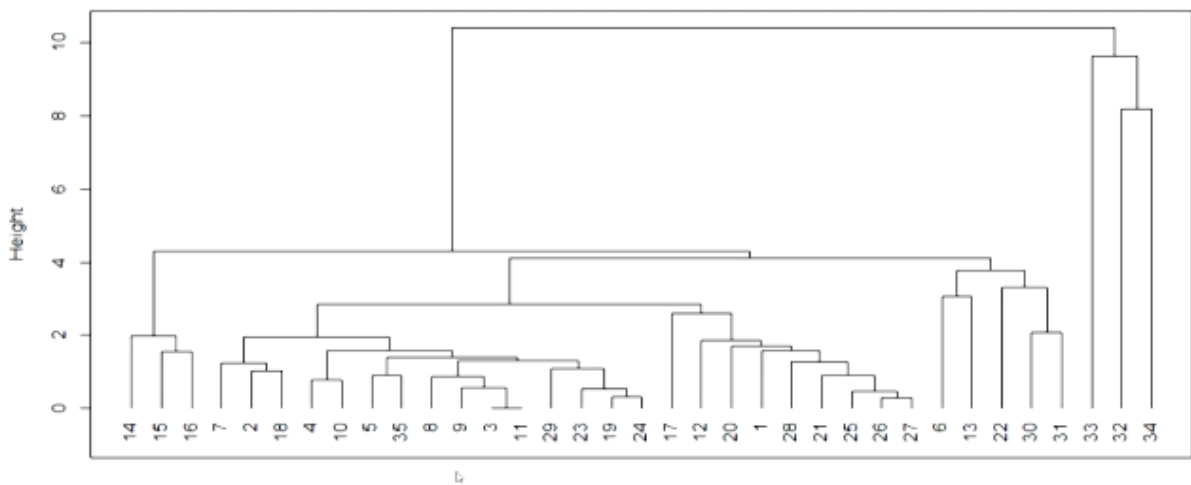

```

[ reached getOption("max.print") -- omitted $ rows ]
> # ward's method
> mod=hclust(DMn, method="ward.D2")
> plot(mod, xlab = "", main = "", hang = -1, sub = "", frame.plot = T)
> # Single linkage
> mod1=hclust(DMn, method="single")
> plot(mod1, xlab = "", main = "", hang = -1, sub = "", frame.plot = T)
>
```
- Plots:** A dendrogram plot is visible in the bottom right corner, showing the hierarchical clustering process with a y-axis labeled 'Height'.

So let's go back, so as we talked about for Ward's method, if we want to have you know, if we have this that we need 6 clusters out of our clustering process, then you can see the example number of desired cluster 6, then what we can do is to cut a single linkage Dendrogram into 6 clusters we can call this function, cut tree function so here we just need to pass on the model object and the number of clusters that we desire, so this particular function is going to cut the model into these 6 clusters, then we have another function rect.hclust which can actually redraw the Dendrogram you know, if we want 6 clusters so those you know this particular Dendrogram is going to be redrawn and we can create the borders to see which observations are going to be combined in particular clusters, so let's run this part of the code, so this groups is just for the you know cutting the single linkage Dendrogram into 6 clusters, so you can see here all the 35 observations(DMn you can see in the environment section, groups variables has been created, we have 35 observations and we are looking to, if we are interested in looking at the values, then we can just write this, we'll just run this, you can see the observations in the output you can see the observations and their cluster ID, so which cluster they belong to, so we created 6 clusters and how they have been assigned, so any you know single, any Dendrogram that we might have, we can cut it as per the requirement of the number of clusters, and using this particular function we'll get the cluster ID's.



Now if we want to visualize this process then we can call this function `rect.hclust` so let's run this, you can see here this particular Dendrogram you can see that borders here, because we had specified this red colour and we can see how these six clusters have been you know formed, so you can see the cluster in this region, in this right part of the region is the biggest cluster having most of the observation, then we have the next cluster is this one having two observation, then we have this cluster having 3 observation, then we have 3 cluster with just one observation, so if we want 6 clusters then this is how, these are the cluster that we are going to have, and we can easily see the observations which are going to be part of you know each of these 6 clusters, so the process, the results are very clearly you know they can be very clearly understood using this particular Dendrogram.



Now we can also apply average linkage approach in HAC, so the same function, now HC clust the same data matrix that we have computed, and method we have to just specify average and we'll get there, so let's run this. Now we can again plot this one as well, you can see a new Dendrogram has been created, now this was created following average linkage approach, so here you can see the way this Dendrogram, the way clusters have formed, the clusters have been merged, observations have been merged, it is quite different from the previous two approaches, so if we just go back and do the same thing, if we are interested in having 6 cluster then again

The screenshot shows an RStudio window with the following R code in the script editor:

```

137
138 # Example: No. of desired clusters=6
139 # To cut a single linkage dendrogram into 6 clusters
140 groups=cutree(mod1, k=6)
141 # To redraw dendrogram with red borders around the 6 clusters
142 rect.hclust(mod1, k=6, border="red")
143
144 # Average linkage
145 mod2=hclust(DMn, method="average")
146
147 plot(mod2, xlab = "", main = "", hang = -1, sub = "", frame.plot = T)
148 # To redraw dendrogram with red borders around the 6 clusters
149 rect.hclust(mod2, k=6, border="red")
150
151 #####
152
1481 (Top Level)

```

The console output shows the execution of the code:

```

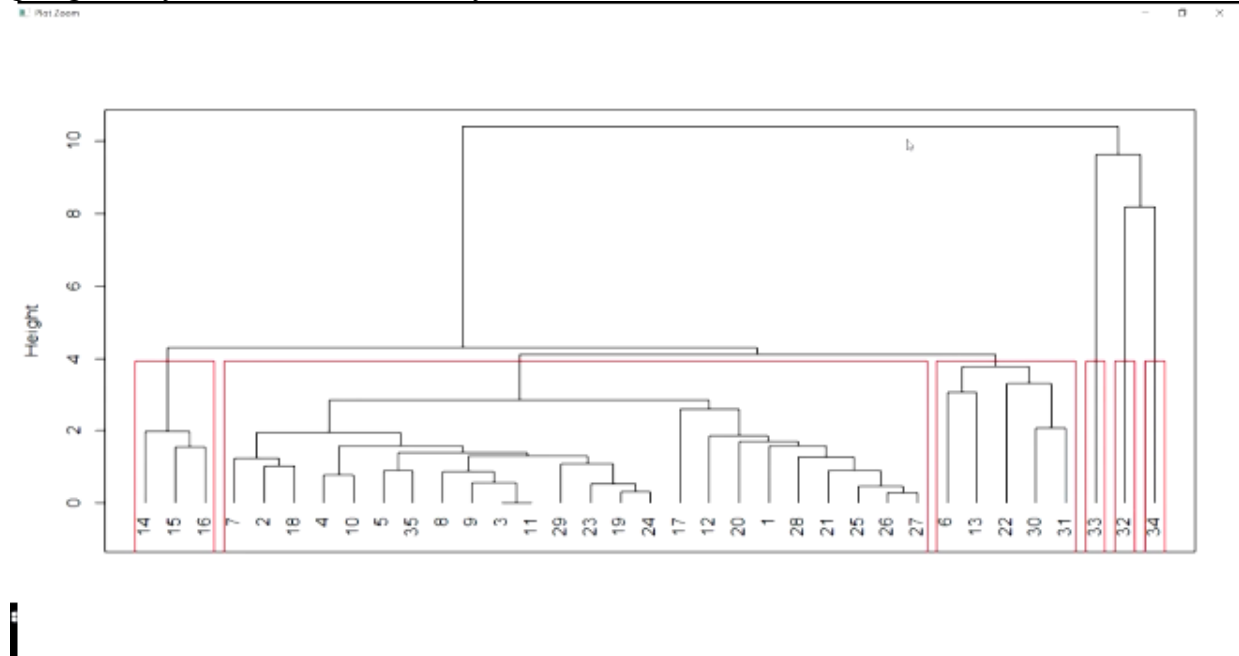
> # To cut a single linkage dendrogram into 6 clusters
> groups=cutree(mod1, k=6)
> groups
[1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 4 5 6 1
> # To redraw dendrogram with red borders around the 6 clusters
> rect.hclust(mod1, k=6, border="red")
> # Average linkage
> mod2=hclust(DMn, method="average")
> plot(mod2, xlab = "", main = "", hang = -1, sub = "", frame.plot = T)
>

```

The environment pane shows variables: mod1 (List of 7), mod2 (List of 7), DM (class 'dist' atomic [1:10] 1...), DM2 (class 'dist' atomic [1:1] 1...), and DMn (class 'dist' atomic [1:595] ...).

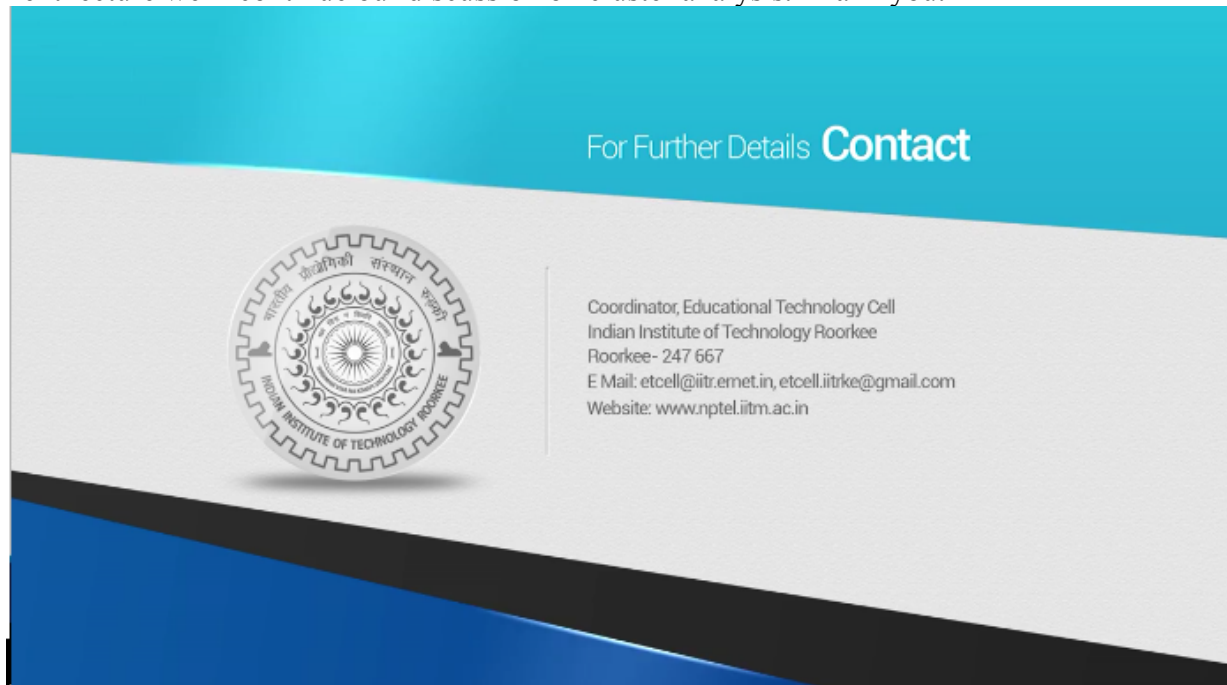
The plot shows a dendrogram with a y-axis labeled 'Height' ranging from 0 to 10. The x-axis labels are 14, 15, 16, 7, 2, 18, 4, 10, 5, 35, 8, 9, 3, 11, 29, 23, 19, 24, 17, 12, 20, 1, 28, 21, 25, 26, 27, 6, 13, 22, 30, 31, 33, 32, 34. Red vertical lines are drawn at x=14, x=16, x=27, x=31, x=33, and x=34 to indicate 6 clusters.

we can call this function rect.hclust and run this, and we'll again see how these clusters are going to be you know identified, so you can see here in the left most cluster, we have 3




observation then the second cluster in the middle, this is the largest cluster that we have, more number of observation most of the observation is cluster, then we have one more cluster with

high observation and then 3 cluster with single observation, if you remember in the previous output that we had, these 3 singleton clusters were there even in the case of single linkage method, and even the average linkage method these three clusters can be clearly seen here, right, so in this way we can easily apply you know hierarchical clustering approach in R and you know we can see through an Dendrogram how the results are going to be, and how the clustering process is happened, all this can be clearly seen. So we'll stop at this point and in the next lecture we'll continue our discussion on cluster analysis. Thank you.



For Further Details **Contact**



Coordinator, Educational Technology Cell
Indian Institute of Technology Roorkee
Roorkee- 247 667
E Mail: etcell@iitr.ernet.in, etcell.iitrke@gmail.com
Website: www.nptel.iitm.ac.in

For Further Details Contact
Coordinator Educational Technology Cell
Indian Institute of Technology Roorkee
Roorkee – 247 667
E Mail:-etcell@iitr.ernet.in, iitrke@gmail.com
Website: www.nptel.iitm.ac.in

Acknowledgement

Prof. Ajit Kumar Chaturvedi
Director, IIT Roorkee

NPTEL Coordinator

IIT Roorkee

Prof. B. K Gandhi

Subject Expert

Dr. Gaurav Dixit

Department of Management Studies
IIT Roorkee

Produced by

Mohan Raj.S

Graphics

Binoy V.P

Web Team

Dr. Nibedita Bisoyi

Neetesh Kumar

Jitender Kumar

Vivek Kumar

Dharamveer Singh

Gaurav Kumar

An educational Technology cell

IIT Roorkee Production

© Copyright All Rights Reserved

WANT TO SEE MORE LIKE THIS

SUBSCRIBE