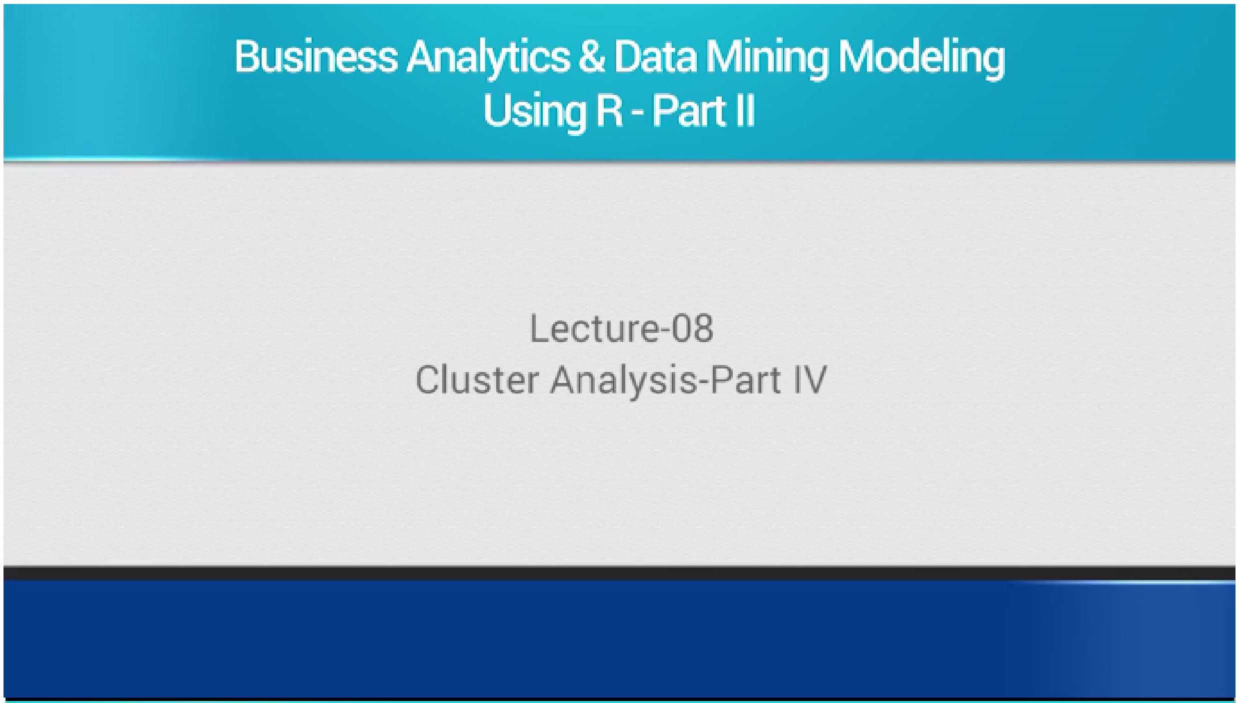
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
NPTEL
NPTEL ONLINE CERTIFICATION COURSE
Business Analytics & Data Mining Modeling
Using R – Part II
Lecture-08
ClusterAnalysis– Part IV
With
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology Roorkee



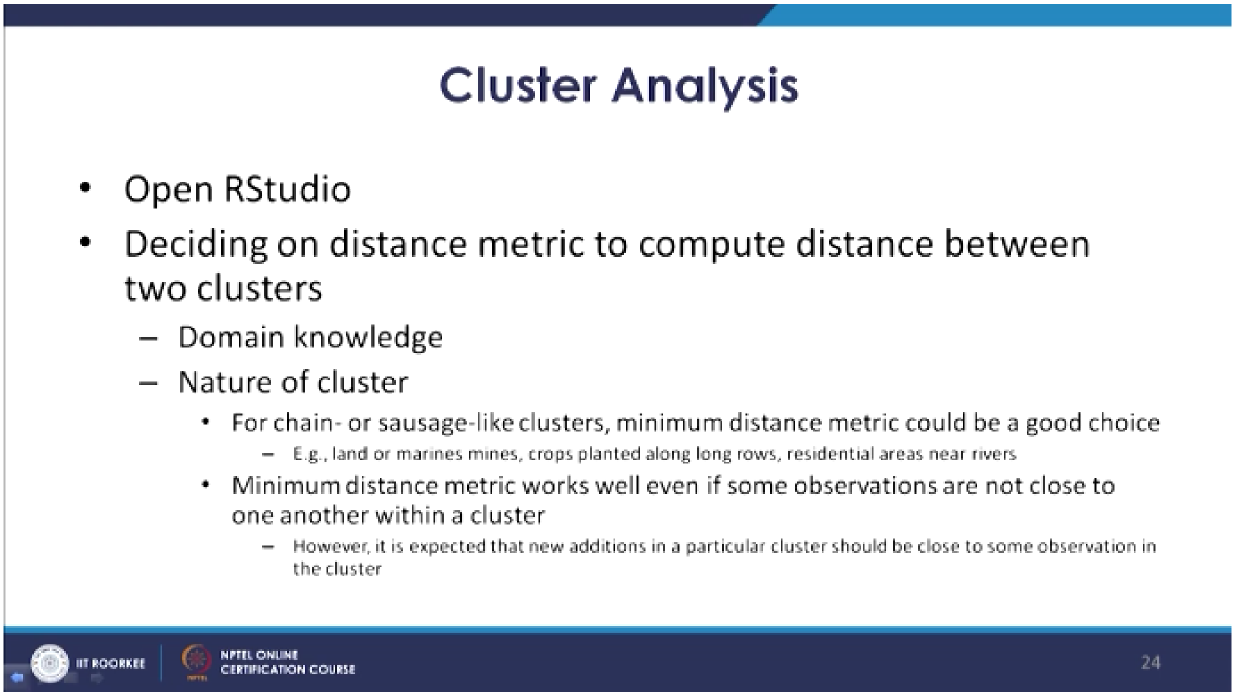
Business Analytics & Data Mining Modeling Using R - Part II

Lecture-08
Cluster Analysis-Part IV

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 discussion cluster analysisand specifically we talked about the distance matrixthat could be used to compute the distance between two clusters. We also did an exercise in R to understand those discussions.

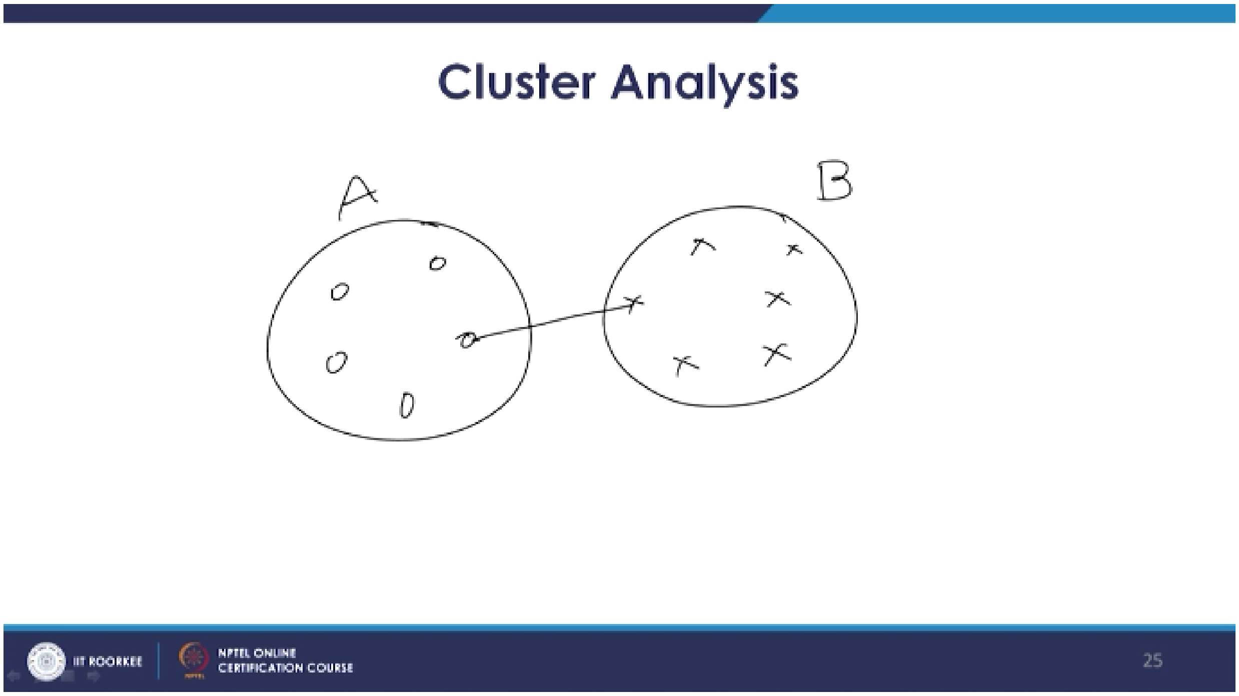
Now let’s continue our discussion on the same topic today in this lecture, so few important points about you know deciding, you know, which distance metric you know the distance metric to use to compute distance between two clusters, so there are certain points which could



Cluster Analysis

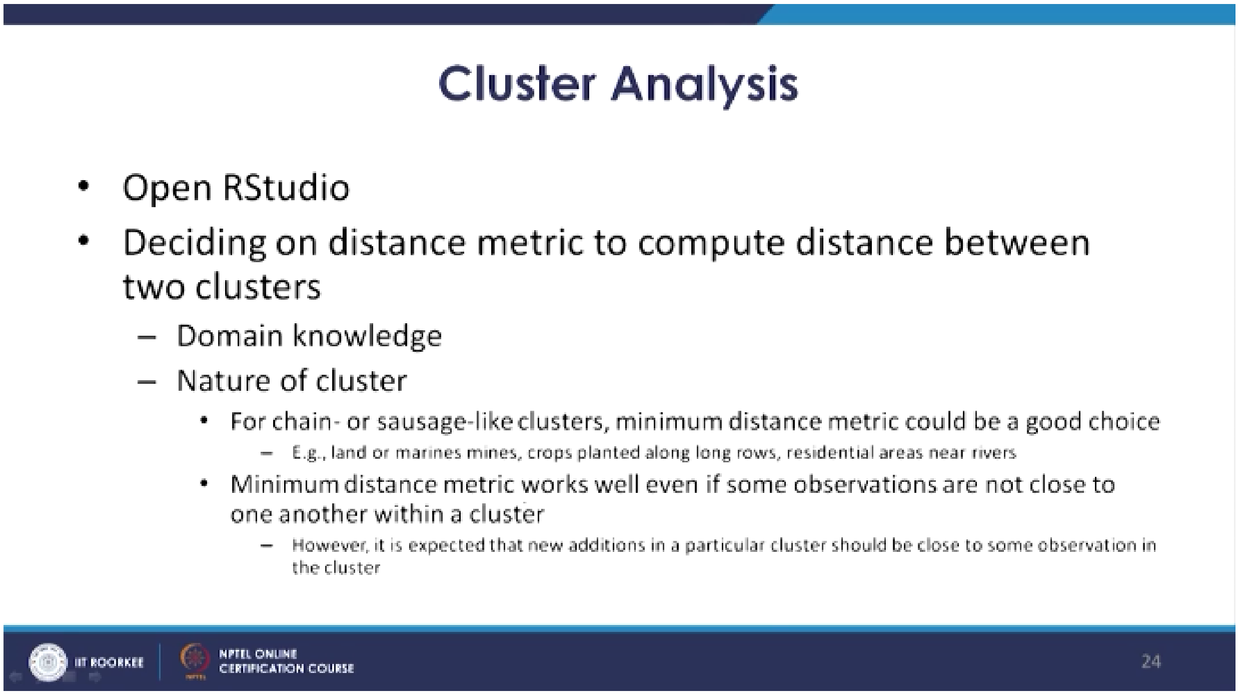
- Open RStudio
- Deciding on distance metric to compute distance between two clusters
  - Domain knowledge
  - Nature of cluster
    - For chain- or sausage-like clusters, minimum distance metric could be a good choice
      - E.g., land or marines mines, crops planted along long rows, residential areas near rivers
    - Minimum distance metric works well even if some observations are not close to one another within a cluster
      - However, it is expected that new additions in a particular cluster should be close to some observation in the cluster

be helpful in making this particular decision, so one is certain, first one is certainly the domain knowledge, so it is the domain knowledge the problems at hand you know the dataset, the problem that we are trying to you know analyze that for which we are trying to build model and analyze, so that particular domain, the problem domain will actually in a way can guide us in terms of what distance metric would be more suitable for computing distances between clusters, so then the second thing is the nature of cluster you know, nature of cluster will also drive this computation. For example, if the nature of cluster is like chain or sausage like then probably the minimum distance metric could be a good choice, because this particular distance metric the way the distances are you know, the way distances are measured it tries to identify the closest observations you know from two clusters.



Cluster Analysis

And in a way it is you know linking of those chains, so this particular metric is like friend of friends kind of scenario and because of this the kind of you know cluster shifts for which this metric could be suitable or like chain or sausage like clusters, so the same thing we can understand here. For example, so if this is one cluster and we have the another cluster here, and we have certain observations in each of this clusters, so let's call this cluster as A, and this cluster as B, if we look at the minimum distance metric probably we'll find out the pair of observation which are closest to each other, and you know these are the two observations, each one belonging to each of these clusters which would be identified, and then this distance is going to be taken as the distance between these two clusters, in this fashion if we look at you know these clusters, if there are more than two clusters then you would see a chain kind of thing forming, so because of this the you know, the clusters which are you know chain like which are more of a chain like clusters, for those or sausage like clusters for those situations this minimum distance metric could be a better choice. Examples, land or marine mines, so

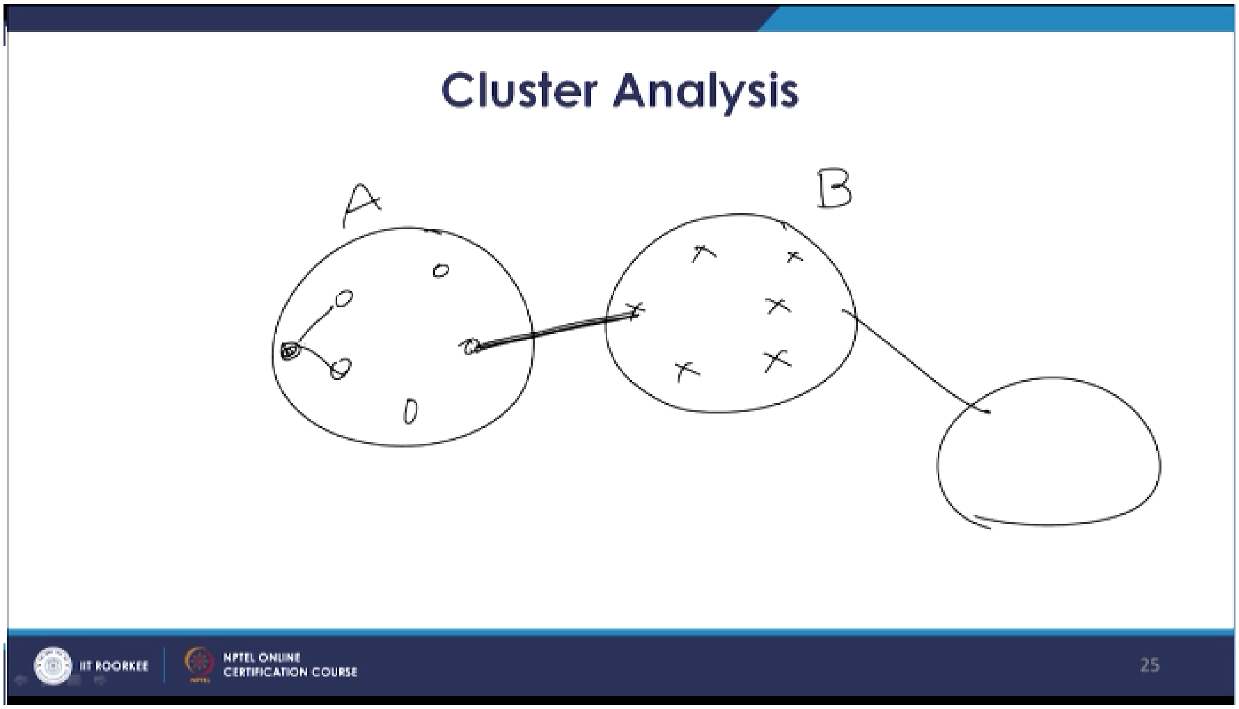


Cluster Analysis

- Open RStudio
- Deciding on distance metric to compute distance between two clusters
  - Domain knowledge
  - Nature of cluster
    - For chain- or sausage-like clusters, minimum distance metric could be a good choice
      - E.g., land or marines mines, crops planted along long rows, residential areas near rivers
    - Minimum distance metric works well even if some observations are not close to one another within a cluster
      - However, it is expected that new additions in a particular cluster should be close to some observation in the cluster

typically if we look at the minings how they are you know, how they are laid down or how they are you know, whether it is about laying the minings or it is about finding out the mines, you know, in whether in the land or in marine, typically you know they are around a particular border area, so the mines or typically laid down along that area, so therefore if we are looking to identify the cluster of mines then probably the minimum distance metric would be a better choice, because they would be along something and because of that the connecting them would be easier for us or understanding this particular of, this kind of observations would be better for us if we use the minimum distance metric, because the nature of those clusters, because of their formation along something, right, because of their formation along something they would form like chain like or sausage like you know clusters.

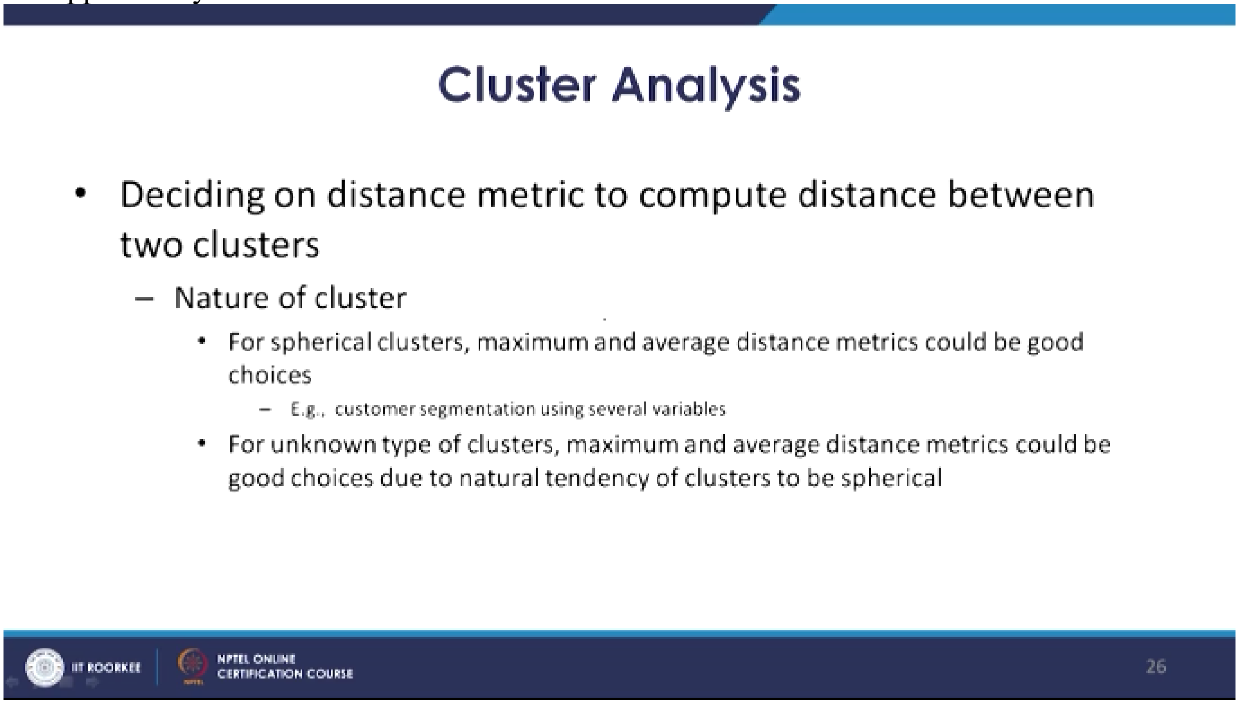
Similarly, you know, one more point, similarly we'll talk about other distance metrics, but before we move ahead one more point about minimum distance metric, so minimum distance metrics works well, even if some observations are not closed to one another within a cluster, so this is another scenario where this particular metric could be useful. However, one important point that it is expected that new additions in a particular clusters, cluster should be closed to some observation in the cluster, so to understand this particular point let's see here that you know the observations for example if you look at this particular cluster, cluster A there could be a distance observation here, and here we have this observation, though these observation are distance from each other, still they are part of the same cluster, so this is the point that was being made here, so even if some observation or not close to one another, you know still they can be part of the same thing and minimum distance metric could make that happened, because distance between clusters or being you know or is being at you know computed in this fashion, so distance observation belonging to this cluster A is, can still be there, however as the you know point also notes here that the observation, that distance observation should be close to you know some other observations, so you can see here this particular observation is closed to some other observation so that it can become part of this cluster, cluster A however if we look



Cluster Analysis

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE 25

at you know its distance from this particular observation then we can see here, this is the distance observation but still part of cluster A, and the minimum distance metric is allowing this to happen in a you know suitable manner.



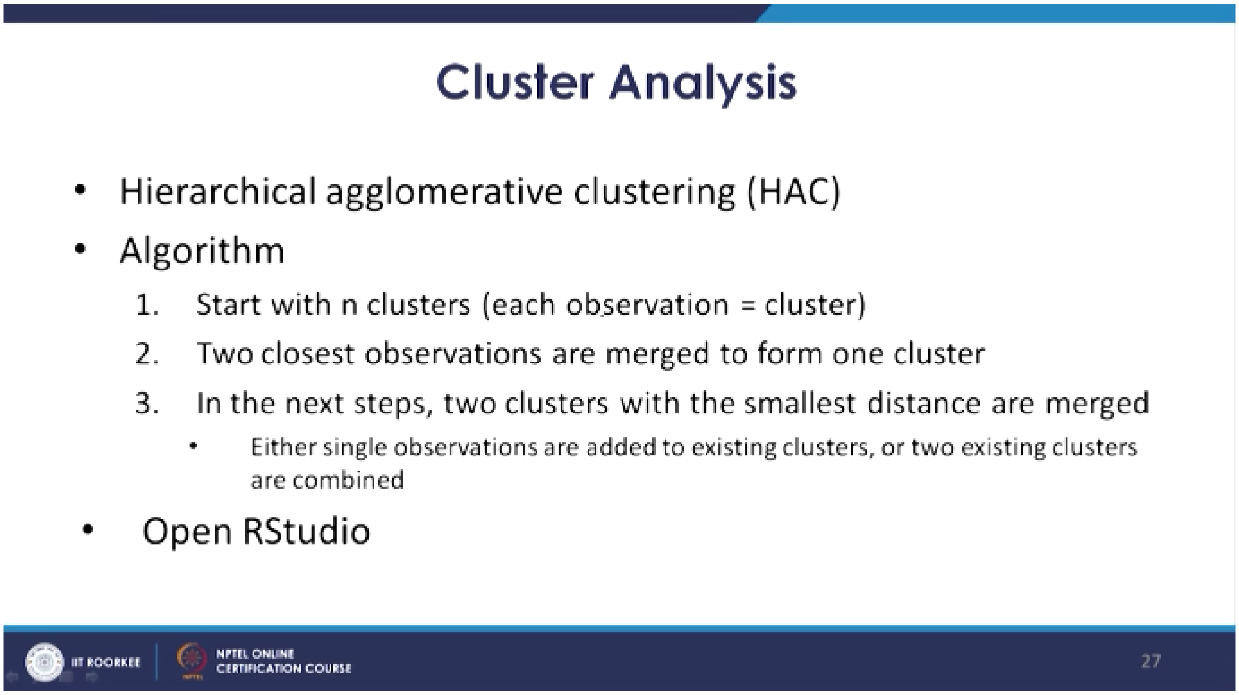
Cluster Analysis

- Deciding on distance metric to compute distance between two clusters
  - Nature of cluster
    - For spherical clusters, maximum and average distance metrics could be good choices
      - E.g., customer segmentation using several variables
    - For unknown type of clusters, maximum and average distance metrics could be good choices due to natural tendency of clusters to be spherical

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE 26

There are some other you know cluster types for which different metrics would be useful, for example spherical clusters, so spherical clusters the maximum and average distance metrics could be good choices, for example customer segmentation using several variables, so if we look at the maximum and average distance metric, so the way distance between two cluster is computed either the farthest pair of observation are involved or all possible you know pair of observation are involved, so essentially the way this computation happen this is going to be more suitable for spherical clusters.

Customer segmentation because typically this is done using a number of variables, when we use a number of variables then essentially it is a long, many dimension and distance computation is along many dimension and the kind of values that we get, because of this, the observations are going to form a cluster in a spherical fashion, so because of this if we are encountering such clusters then probably these are the maximum and average distance metrics are the one that could be used. And if we talk about some other situations where the nature of cluster, the shape of cluster is not well-known or not understood even in those situations these metrics, maximum and average distance metrics can be used because this is the natural tendency, because if we are going to do distance computation using a number of variables, so because of this the natural tendency of clusters is to be spherical, and because of this maximum and average distance metric could be found to be more suitable.



Cluster Analysis

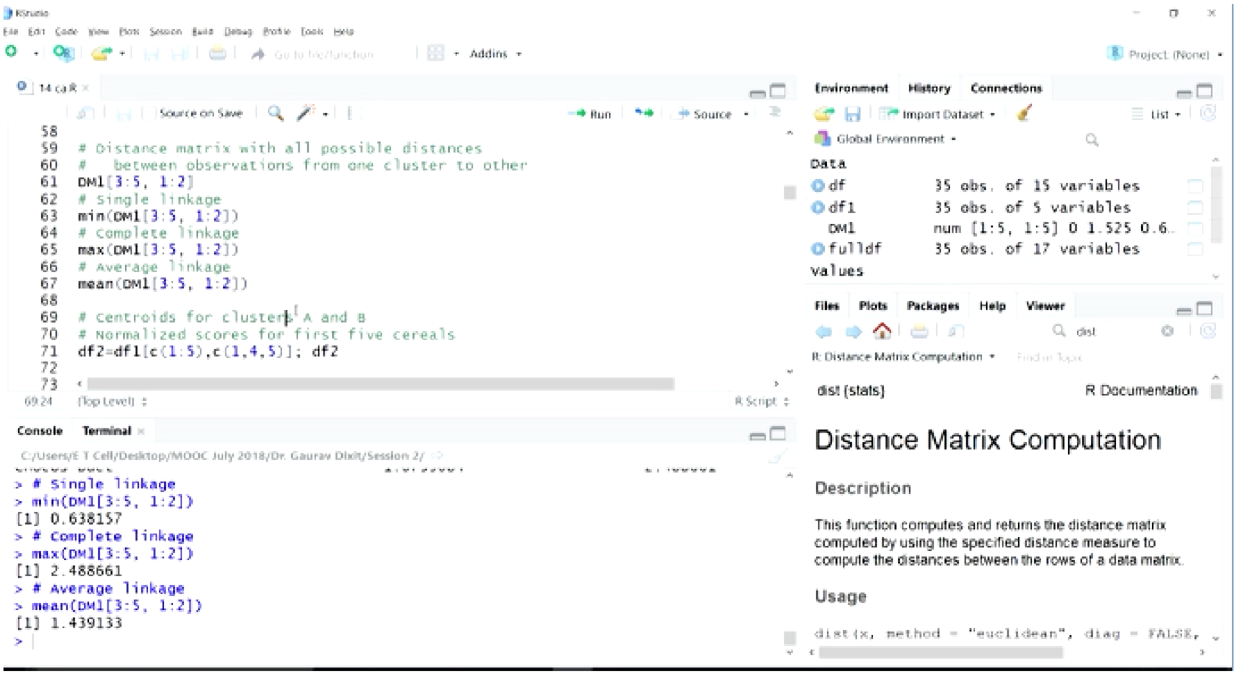
- Hierarchical agglomerative clustering (HAC)
- Algorithm
  1. Start with n clusters (each observation = cluster)
  2. Two closest observations are merged to form one cluster
  3. In the next steps, two clusters with the smallest distance are merged
     - Either single observations are added to existing clusters, or two existing clusters are combined
- Open RStudio

So these were some of the points which could be helpful in terms of deciding, which metric is going to be useful for what kind of, for different kind of in a situations, so this discussion brings us to our next part, so that is as we talked about in previous lecture that we would be covering a hierarchical agglomerative clustering and K means clustering one technique from each of those two types hierarchical methods and non-hierarchical methods, so within hierarchical method we are covering this one hierarchical agglomerative clustering within non-hierarchical methods we would be talking about K means clustering.

So let's start with our discussion on hierarchical agglomerative clustering, so the main idea behind the agglomerative clustering we have already talked about in previous lecture, so let's understand the algorithm and it's steps, so in this particular algorithm the way clustering happens, so typically we start with the N clusters, so each observation is going to be seen as 1 cluster, so each observation is going to be equivalent of one cluster, so we will start with, if there are N number of observations, so each observation would be considered as a cluster, a cluster having a single observation, so we will start with, essentially we'll start with N clusters.

Now out of these N clusters as you can see from the step number 2, two closest observations are merged to form 1 cluster, so out of these N clusters that we start with, so these all N clusters are nothing but N observations, so we tried to find out the observation which are closest to each other and then joint them or merge them, so this merger leads to creation of one cluster, having two observations, so in this fashion we move forward, so if we look at the third step in this algorithm it says that in the next steps 2 clusters with the smallest distance are merged, so now after this step number 2 what we have is one cluster having two observations and the remaining N- 2 clusters, so now within these total, so now total we have N-1 cluster and one of them having a you know, one of them having two observation and the, then we have the N-2 other clusters.

Now within all these clusters N-2 clusters we again try to find out the two clusters which are closest and then merge them, so either, so what is going to happen now is, either single observations are added to existing clusters, so for example we have one cluster with 2 you know, 2 observation, so either some other observation can also be added to this cluster or our you know single observation clusters, 2 of them might be merged together and one cluster could be formed. So either single observations are added to existing clusters or 2 existing clusters are combined, so these are the two scenarios so in this fashion we keep on moving, so we keep on finding either the two closest observations or you know, two closest you know clusters or you know one observation being closed to you know some other cluster, so in this fashion we move ahead and essentially what will happen is as we talked about in previous lecture, in the you know agglomerative clustering, essentially we will end up with just one cluster having all the observations.



```
# Distance matrix with all possible distances
#   between observations from one cluster to other
DM1[3:5, 1:2]
# Single linkage
min(DM1[3:5, 1:2])
# Complete linkage
max(DM1[3:5, 1:2])
# Average linkage
mean(DM1[3:5, 1:2])

# Centroids for clusters A and B
# Normalized scores for first five cereals
df2=df1[c(1:5),c(1,4,5)]; df2
```

```
> # Single linkage
> min(DM1[3:5, 1:2])
[1] 0.638157
> # Complete linkage
> max(DM1[3:5, 1:2])
[1] 2.488661
> # Average linkage
> mean(DM1[3:5, 1:2])
[1] 1.439133
> 
```

```
Data
df        35 obs. of 15 variables
df1       35 obs. of 5 variables
 DM1      num [1:5, 1:5] 0 1.525 0.6...
fulldf    35 obs. of 17 variables
Values
```

dist {stats}   R Documentation

Distance Matrix Computation

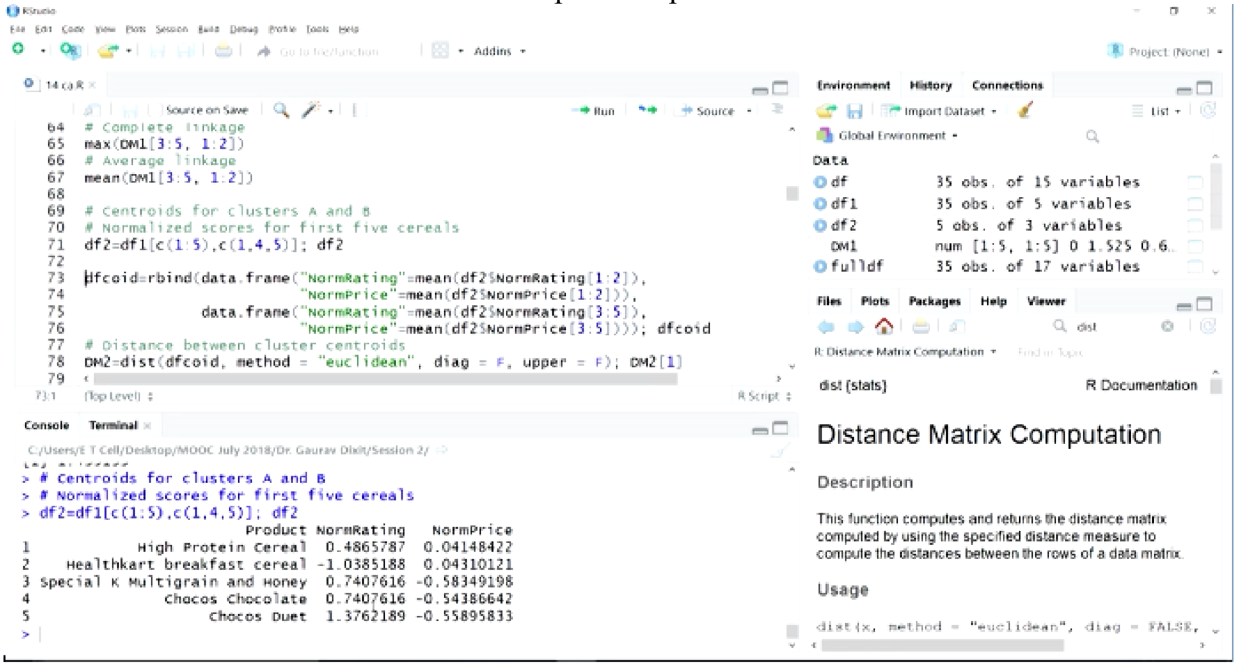
Description

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,
```

So let’s go back to R studio and let’s go through a few exercises to understand these concepts, so before we move ahead in the previous exercise in R, we were able to cover the, we were able to you know compute distances using single linkage, complete linkage and average linkage, but we also talked about one another metric that is centroid distance to compute the distances between clusters, so let’s also go through this exercise first and then we will you know move further, so if we have two centroids, if we have two clusters, cluster A and B and we want to compute the centroids for this two, so what we will do, you know first we'll use this particular information DF1 where we had the normalize this course, so if we are interested in first five cereals so we can, we run this code and we will get this information, so first five cereals and normalized this course, and rating and price we can see here, so why we need this information? Because we need to compute the centroids, so far you know computation on the centroid we need values for all the observation that are part of a particular cluster.



```
# Complete linkage
max(DM1[3:5, 1:2])
# Average linkage
mean(DM1[3:5, 1:2])

# Centroids for clusters A and B
# Normalized scores for first five cereals
df2=df1[c(1:5),c(1,4,5)]; df2

dfcoid=rbind(data.frame("NormRating"=mean(df2$NormRating[1:2]),
                        "NormPrice"=mean(df2$NormPrice[1:2])),
             data.frame("NormRating"=mean(df2$NormRating[3:5]),
                        "NormPrice"=mean(df2$NormPrice[3:5]))); dfcoid
# Distance between cluster centroids
DM2=dist(dfcoid, method = "euclidean", diag = F, upper = F); DM2[1]
```

```
> # Centroids for clusters A and B
> # Normalized scores for first five cereals
> df2=df1[c(1:5),c(1,4,5)]; df2
                      Product NormRating   NormPrice
1          High Protein Cereal  0.4865787  0.04148422
2    Healthkart breakfast cereal -1.0385188  0.04310121
3 Special K Multigrain and Honey  0.7407616 -0.58349198
4             Chocos Chocolate  0.7407616 -0.54386642
5                  Chocos Duet  1.3762189 -0.55895833
> 
```

```
Data
df       35 obs. of 15 variables
df1      35 obs. of 5 variables
df2      5 obs. of 3 variables
DM1      num [1:5, 1:5] 0 1.525 0.6...
fulldf   35 obs. of 17 variables
```

dist {stats}   R Documentation

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,
```

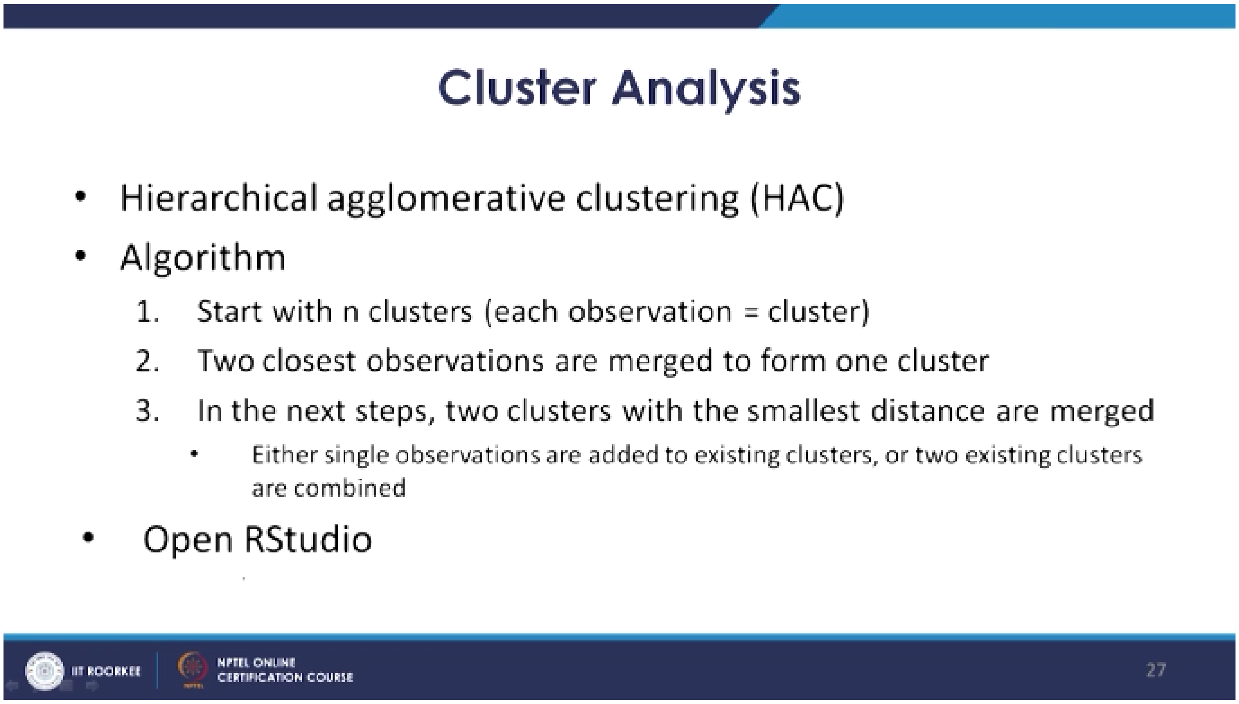
So if we look at the next code, so here we see that the, what we have is the mean values here, mean values are being computed using you know the observations, you know 1 and 2, so because first two rows high protein cereal and health kart breakfast cereal they are part of the cluster A as we talked about in the previous lecture, so these two observations are to be used to compute the you know average value for this two dimensions, rating and price, this two variables, right, so you can see mean is being taken for these two values 1 to 2 that means there is just two values.

If we look at the you know for the next you know centroid we can see here is that for two, for you know this particular centroid, the second centroid we are considering three observations, we have 3 cereals, special K, multigrain and honey, then chocos, chocolate and chocos duet, so 3 to 5, so these 3 rows are being used to, you know take the, compute the average value, so for these two variables rating and price will get the mean values, and these values will, this probably you know computation will actually give us the centroid for the two clusters, so let's run this code.

Now you can see, the row number 1 is giving us the centroid for cluster A, and the row number 2 is giving us the centroid for cluster B, so if we want to use the, if we want to compute the distance between clusters to clusters using the centroid distance method, then essentially what we need is we need to compute the distance between these two points, so for this we can use the distance function again, and just like we had computed the distance metrics for this 5 cereals, so here again we have two points, and we need to compute the distance between these two, so we can use the dist function and we will get this value, so let's run this code.

Now you can see the values has been computed, so this is the distance between these two centroids that we have just computed and essentially this is going to be taken as the distance between 2 cluster, so this is the centroid distance method for us, if we want to use this particular metric to compute the distance between 2 clusters.

Now let’s start our discussion of hierarchical you know agglomerative clustering that this algorithm that we have discussed just now, so again for understanding this particular algorithm further again let’s consider these two variables, these two variables once again customer rating and price, and we’ll consider first five cereals only to understand this discussion, so again before we move ahead we need to have the distances between you know, distance value between different cereals, so for this we are going to use this particular metrics which we have already computed, so this is using Euclidean distance metric on normalize scales, so this particular you know metrics we had computed before, we have seen this before in the previous lecture as well, so let’s run this. And in this metrics as we saw in the previous lecture we can see each of these cereals and the distance between each of this cereal with the remaining cereals, so these distance values are available with us.

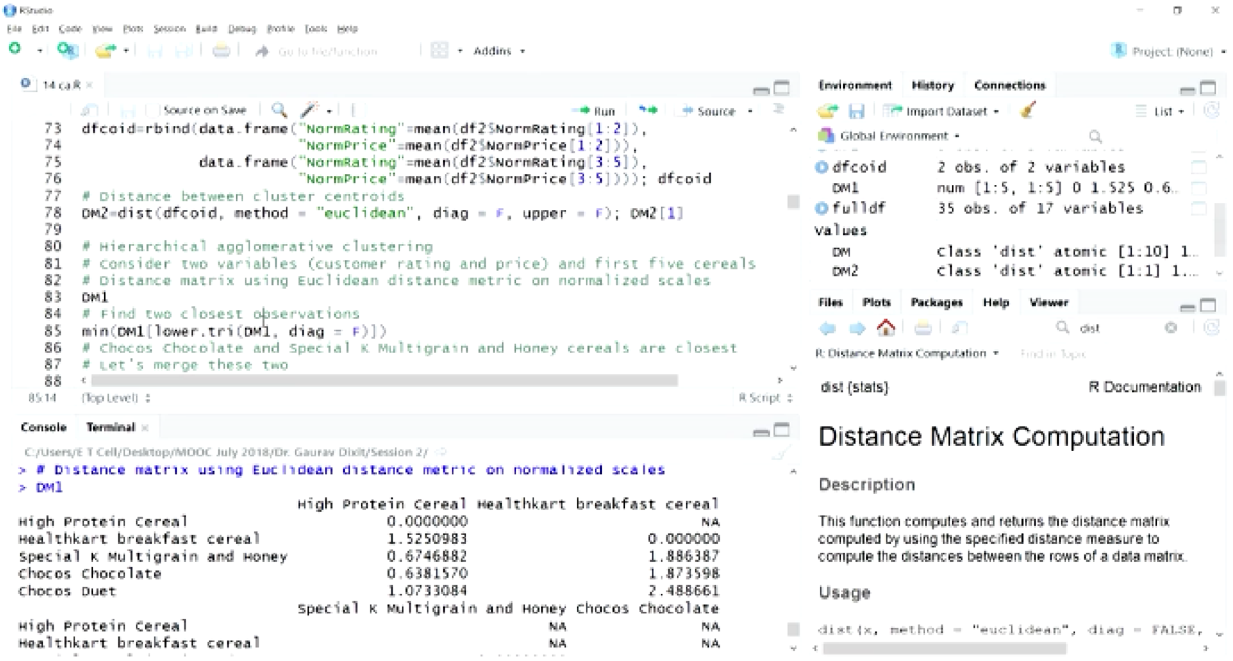


Cluster Analysis

- Hierarchical agglomerative clustering (HAC)
- Algorithm
  1. Start with n clusters (each observation = cluster)
  2. Two closest observations are merged to form one cluster
  3. In the next steps, two clusters with the smallest distance are merged
     - Either single observations are added to existing clusters, or two existing clusters are combined
- Open RStudio

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE 27

Now once this distance values are available with us remember the first step in the hierarchical agglomerative clustering, so let’s go back to our slide, so you can see start with N clusters and then the second step is two closest observations are merged to form one cluster, so we need to identify two closest observation, so that is going to be the important task here, so to find out the two closest observation we can learn this code, so what we are doing is we are taking the minimum value of you know, minimum value of this distance metrics DM1 and within this because the matrix is symmetrical we are just taking the lower triangular and we’ll get the value, so let’s run this code.



```
dfcoid=rbind(data.frame("NormRating"=mean(df2$NormRating[1:2]),
                        "NormPrice"=mean(df2$NormPrice[1:2])),
             data.frame("NormRating"=mean(df2$NormRating[3:5]),
                        "NormPrice"=mean(df2$NormPrice[3:5]))); dfcoid
# Distance between cluster centroids
DM2=dist(dfcoid, method = "euclidean", diag = F, upper = F); DM2[1]

# Hierarchical agglomerative clustering
# Consider two variables (customer rating and price) and first five cereals
# Distance matrix using Euclidean distance metric on normalized scales
DM1
# Find two closest observations
min(DM1[lower.tri(DM1, diag = F)])
# Chocos Chocolate and Special K Multigrain and Honey cereals are closest
# Let's merge these two
```

```
C:/Users/E T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 2/
> # Distance matrix using Euclidean distance metric on normalized scales
> DM1
                                High Protein Cereal Healthkart breakfast cereal
High Protein Cereal                       0.0000000                          NA
Healthkart breakfast cereal               1.5250983                    0.000000
Special K Multigrain and Honey            0.6746882                    1.886387
Chocos Chocolate                          0.6381570                    1.873598
Chocos Duet                               1.0733084                    2.488661
                                Special K Multigrain and Honey Chocos Chocolate
High Protein Cereal                                         NA               NA
Healthkart breakfast cereal                                 NA               NA
```

```
dfcoid      2 obs. of 2 variables
  DM1       num [1:5, 1:5] 0 1.525 0.6…
fulldf      35 obs. of 17 variables
Values
  DM        class 'dist' atomic [1:10] 1…
  DM2       class 'dist' atomic [1:1] 1…
```

dist {stats} R Documentation

Distance Matrix Computation

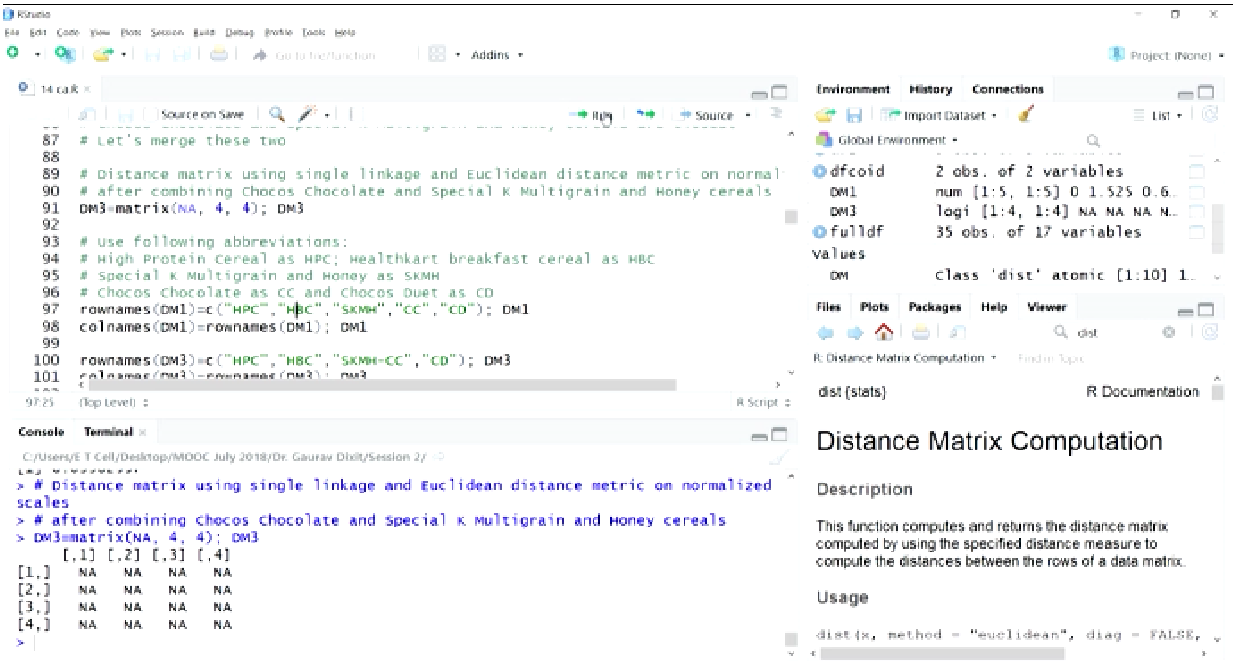
Description

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,
```

So this is the value 0.03962557 this is the lowest you know distance between you know two observations, so let's find out you know which observations have this value, so if we go back, so we will see that where you know, so this is the value we can see, this was the value, so this value is between this two cereals choco chocolate and special K multigrain and honey, so these two are the cereals which are closest, so as we understood from the algorithm, two closest observation are merged to form one clusters, so as indicated in the algorithm step, now we will plan to merge this two observation, choco chocolate and special K multigrain, so these two observation will be merged to form one clusters, and then will have the remaining observations as clusters you know individual you know clusters with single item, right, so essentially what we left with, so one particular cluster having 2 observation and 3 other clusters with having single observation, so essentially now if in terms of distance values, you know distance values for clusters we will have to compute this 4 x 4 metrics, so you can see here, so let's initialize this 4 x 4 metrics, so you can see here initialization has been done, and now we are going to use because the names of these cereals are you know they're quite long so we'll take, we'll do some abbreviation here and you can see here a high protein cereal now is going to be used as HBC, the health kart breakfast cereals is going to be used as HBC, and similarly other cereals they have been abbreviated and these abbreviated names are going to be use now, so these 4 x 4 metrics that we have just now created will change the row names and column names, so that we are able to understand which cereals, which particular cereals we are talking about.



```
# Let's merge these two

# Distance matrix using single linkage and Euclidean distance metric on normal
# after combining Chocos Chocolate and Special K Multigrain and Honey cereals
DM3=matrix(NA, 4, 4); DM3

# Use following abbreviations:
# High Protein Cereal as HPC; Healthkart breakfast cereal as HBC
# Special K Multigrain and Honey as SKMH
# Chocos Chocolate as CC and Chocos Duet as CD
rownames(DM1)=c("HPC","HBC","SKMH","CC","CD"); DM1
colnames(DM1)=rownames(DM1); DM1

rownames(DM3)=c("HPC","HBC","SKMH-CC","CD"); DM3
```

```
> # Distance matrix using single linkage and Euclidean distance metric on normalized
scales
> # after combining Chocos Chocolate and Special K Multigrain and Honey cereals
> DM3=matrix(NA, 4, 4); DM3
     [,1] [,2] [,3] [,4]
[1,]   NA   NA   NA   NA
[2,]   NA   NA   NA   NA
[3,]   NA   NA   NA   NA
[4,]   NA   NA   NA   NA
>
```

```
dfcoid    2 obs. of 2 variables
DM1       num [1:5, 1:5] 0 1.525 0.6...
DM3       logi [1:4, 1:4] NA NA NA N...
fulldf    35 obs. of 17 variables
values
DM        class 'dist' atomic [1:10] 1...
```

dist {stats} R Documentation

Distance Matrix Computation

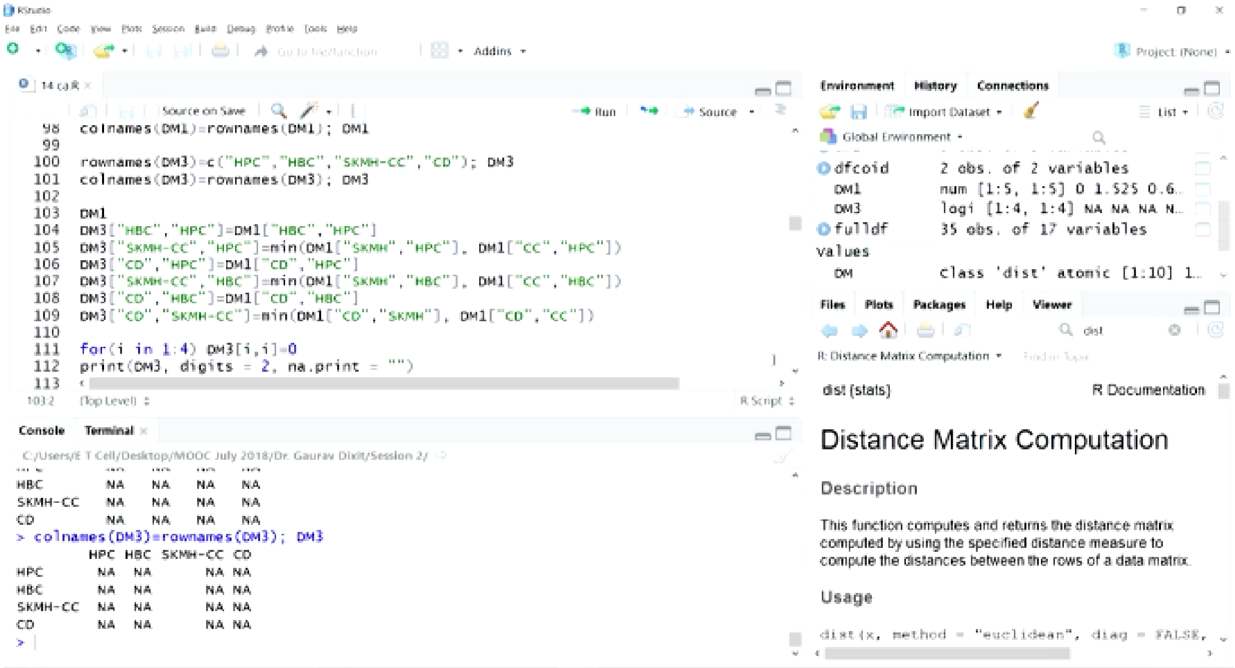
Description

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,`

So let’s change the row names, so first we are changing the row names for DM1 our original you know 5 x 5 metrics, and then we’ll do this the same for the 4 x 4 metrics, so DM1 let’s run this code, now you can see in the DM1 where we had the you know 5 cereals, the original metrics, distance metric and we had identified, this is the, these two are the closest cereals CC and SKMH, these were identified to be the closest.



```
colnames(DM1)=rownames(DM1); DM1

rownames(DM3)=c("HPC","HBC","SKMH-CC","CD"); DM3
colnames(DM3)=rownames(DM3); DM3

DM1
DM3["HBC","HPC"]=DM1["HBC","HPC"]
DM3["SKMH-CC","HPC"]=min(DM1["SKMH","HPC"], DM1["CC","HPC"])
DM3["CD","HPC"]=DM1["CD","HPC"]
DM3["SKMH-CC","HBC"]=min(DM1["SKMH","HBC"], DM1["CC","HBC"])
DM3["CD","HBC"]=DM1["CD","HBC"]
DM3["CD","SKMH-CC"]=min(DM1["CD","SKMH"], DM1["CD","CC"])

for(i in 1:4) DM3[i,i]=0
print(DM3, digits = 2, na.print = "")
```

```
HBC         NA   NA   NA   NA
SKMH-CC     NA   NA   NA   NA
CD          NA   NA   NA   NA
> colnames(DM3)=rownames(DM3); DM3
        HPC HBC SKMH-CC CD
HPC      NA  NA      NA NA
HBC      NA  NA      NA NA
SKMH-CC  NA  NA      NA NA
CD       NA  NA      NA NA
>
```

```
dfcoid    2 obs. of 2 variables
DM1       num [1:5, 1:5] 0 1.525 0.6...
DM3       logi [1:4, 1:4] NA NA NA N...
fulldf    35 obs. of 17 variables
values
DM        class 'dist' atomic [1:10] 1...
```

dist {stats} R Documentation

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.

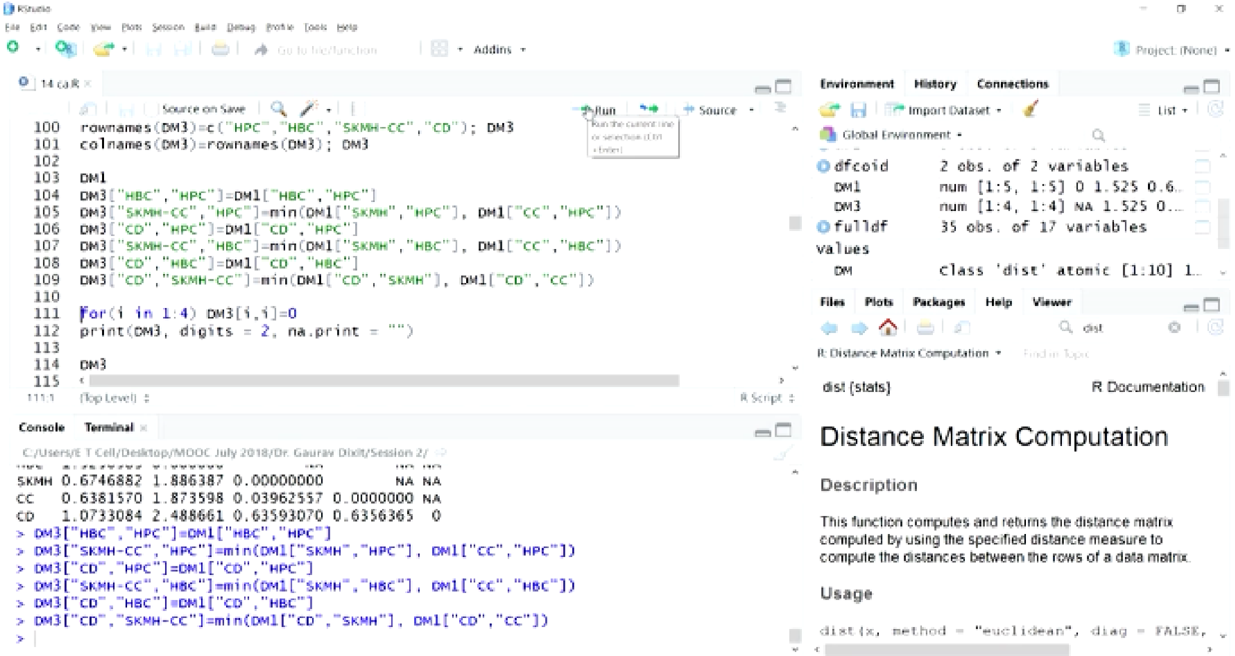
Usage

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

Now we’ll need the 4 x 4 metrics which we have already initialized, so let’s change the row names and column names for this particular metrics as well, and you can see here. Now you can see the third cluster we have renamed this one as SKMH-CC, so this is because these two observation have been clubbed or merged to form 1 cluster, you can see here, the cluster has been, these two observation have been merged into 1 cluster, and the cluster has been renamed as SKMH-CC, so now what we need to do is we need to find the distances between these clusters, so some of this distances can be you know directly taken from the 5 x 5 metrics for other distances, now we need to see out of which two observations, so in the cluster SKMH-CC out of you know there are two observation, one is SKMH, another one is CC, so which observation is closest to the you know other clusters observation, and then that is to be that minimum value because we are using here minimum distance metric formula, so therefore that particular value is going to be used as the distance between those clusters.

So let’s have a relook at this DM1, so this is the DM1 5 x 5 metrics having 5 cereals and the distances between them, now if we look at the our 4 x 4 metrics, first distance that we need to compute is look at the metrics, first distance HBC between this cluster HBC and HPC, so we need to compute this value, so this value can be directly taken from DM1 as you can see here, so let’s run this.

Then this is important the next one, so next value is between if we go back next value is between this cluster of two observation SKMH-CC and HPC, so this is the value that we need to compute now, so if you look at this because we are using minimum distance metric, I’m using the mean function and there are two possible values, right DM1 you know from SKMH to HPC and then CC to HPC, so out of these two values the value which is minimum is going to be taken as the distance, right, so as discussed let’s run this.



```
rownames(DM3)=c("HPC","HBC","SKMH-CC","CD"); DM3
colnames(DM3)=rownames(DM3); DM3

DM1
DM3["HBC","HPC"]=DM1["HBC","HPC"]
DM3["SKMH-CC","HPC"]=min(DM1["SKMH","HPC"], DM1["CC","HPC"])
DM3["CD","HPC"]=DM1["CD","HPC"]
DM3["SKMH-CC","HBC"]=min(DM1["SKMH","HBC"], DM1["CC","HBC"])
DM3["CD","HBC"]=DM1["CD","HBC"]
DM3["CD","SKMH-CC"]=min(DM1["CD","SKMH"], DM1["CD","CC"])

for(i in 1:4) DM3[i,i]=0
print(DM3, digits = 2, na.print = "")

DM3
```

```
C:/Users/E T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 2/
SKMH 0.6746882 1.886387 0.00000000        NA NA
CC   0.6381570 1.873598 0.03962557 0.0000000 NA
CD   1.0733084 2.488661 0.63593070 0.6356365  0
> DM3["HBC","HPC"]=DM1["HBC","HPC"]
> DM3["SKMH-CC","HPC"]=min(DM1["SKMH","HPC"], DM1["CC","HPC"])
> DM3["CD","HPC"]=DM1["CD","HPC"]
> DM3["SKMH-CC","HBC"]=min(DM1["SKMH","HBC"], DM1["CC","HBC"])
> DM3["CD","HBC"]=DM1["CD","HBC"]
> DM3["CD","SKMH-CC"]=min(DM1["CD","SKMH"], DM1["CD","CC"])
> 
```

```
dfcoid      2 obs. of 2 variables
  DM1       num [1:5, 1:5] 0 1.525 0.6...
  DM3       num [1:4, 1:4] NA 1.525 0...
fulldf      35 obs. of 17 variables
values
  DM        class 'dist' atomic [1:10] 1...
```

dist {stats}   R Documentation

Distance Matrix Computation

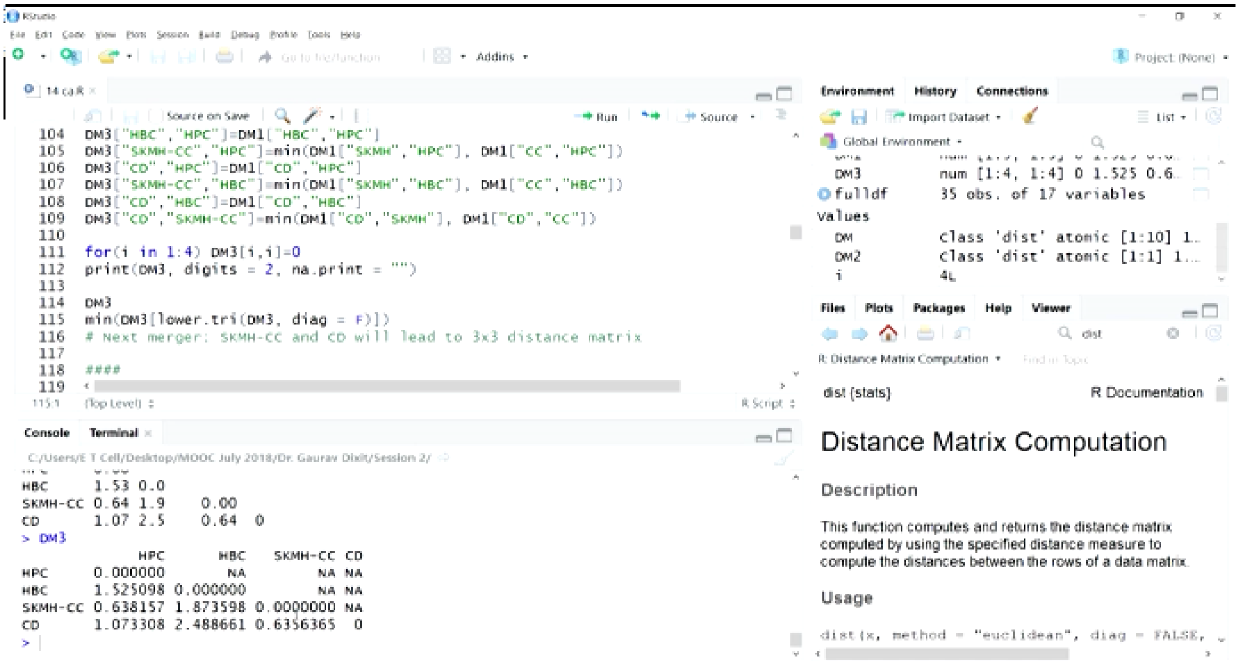
Description

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 similarly for other distances we can run this code and we’ll get these values. So it is only 3 distance computations where this cluster with 2 observations that is SKMH-CC this cluster with 2 observation and the remaining you know 3 clusters so those two distances are the one where this minimum distance metric is going to be used.


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
14 ca.R
Source on Save
Run
Source
Environment History Connections
Import Dataset
List
Global Environment
Files Plots Packages Help Viewer
R: Distance Matrix Computation
Find in Topic
Console Terminal

```
DM3["HBC","HPC"]=DM1["HBC","HPC"]
DM3["SKMH-CC","HPC"]=min(DM1["SKMH","HPC"], DM1["CC","HPC"])
DM3["CD","HPC"]=DM1["CD","HPC"]
DM3["SKMH-CC","HBC"]=min(DM1["SKMH","HBC"], DM1["CC","HBC"])
DM3["CD","HBC"]=DM1["CD","HBC"]
DM3["CD","SKMH-CC"]=min(DM1["CD","SKMH"], DM1["CD","CC"])

for(i in 1:4) DM3[i,i]=0
print(DM3, digits = 2, na.print = "")

DM3
min(DM3[lower.tri(DM3, diag = F)])
# Next merger: SKMH-CC and CD will lead to 3x3 distance matrix

####
```

```
C:/Users/E T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 2/
HBC     1.53 0.0
SKMH-CC 0.64 1.9     0.00
CD      1.07 2.5     0.64  0
> DM3
             HPC      HBC   SKMH-CC CD
HPC     0.000000       NA        NA NA
HBC     1.525098 0.000000        NA NA
SKMH-CC 0.638157 1.873598 0.0000000 NA
CD      1.073308 2.488661 0.6356365  0
>
```

```
DM3       num [1:4, 1:4] 0 1.525 0.6...
fulldf    35 obs. of 17 variables
Values
DM        class 'dist' atomic [1:10] 1...
DM2       class 'dist' atomic [1:1] 1...
i         4L
```

dist {stats}   R Documentation

Distance Matrix Computation

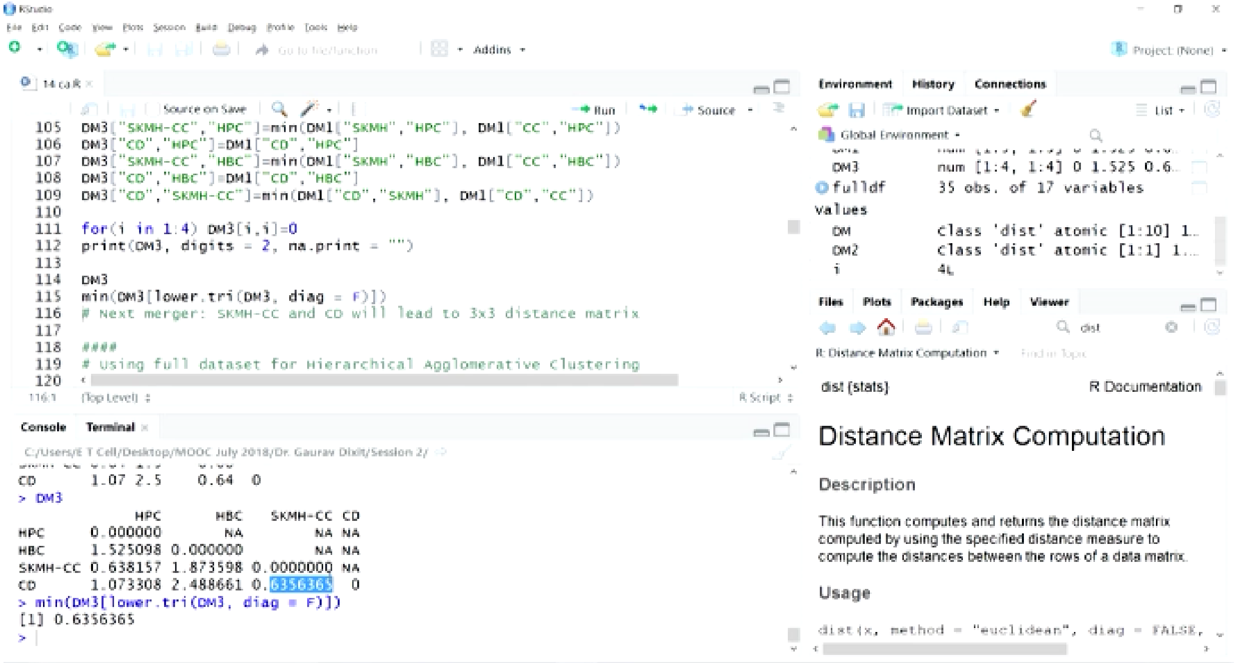
Description

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 complete this particular metrics let’s run this part of the code and we can filled this now so you see, in this 4 x 4 metrics we have all the values now up to two decimal places, I can see here and let’s look at the values with the more decimal points, so this is the metrics, now you can see. Now if you know as the next step if we go back to the algorithm here again in the next steps two clusters with the smallest distance are merge, so again our task is quite similar, within this 4 x 4 metrics we need to identify which two clusters are closest and then merge them together, so first we need to find out the value, so distance values we already have, now we need to just find out the minimum value here, so let’s run this code and we’ll get the value, so this is the value which is minimum 0.6356365, we try to find this value in this metrics this is


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
14 ca.R
Source on Save
Run
Source
Environment History Connections
Import Dataset
List
Global Environment
Files Plots Packages Help Viewer
R: Distance Matrix Computation
Find in Topic
Console Terminal

```
DM3["SKMH-CC","HPC"]=min(DM1["SKMH","HPC"], DM1["CC","HPC"])
DM3["CD","HPC"]=DM1["CD","HPC"]
DM3["SKMH-CC","HBC"]=min(DM1["SKMH","HBC"], DM1["CC","HBC"])
DM3["CD","HBC"]=DM1["CD","HBC"]
DM3["CD","SKMH-CC"]=min(DM1["CD","SKMH"], DM1["CD","CC"])

for(i in 1:4) DM3[i,i]=0
print(DM3, digits = 2, na.print = "")

DM3
min(DM3[lower.tri(DM3, diag = F)])
# Next merger: SKMH-CC and CD will lead to 3x3 distance matrix

####
# using full dataset for Hierarchical Agglomerative Clustering
```

```
C:/Users/E T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 2/
CD      1.07 2.5     0.64  0
> DM3
             HPC      HBC   SKMH-CC CD
HPC     0.000000       NA        NA NA
HBC     1.525098 0.000000        NA NA
SKMH-CC 0.638157 1.873598 0.0000000 NA
CD      1.073308 2.488661 0.6356365  0
> min(DM3[lower.tri(DM3, diag = F)])
[1] 0.6356365
>
```

```
DM3       num [1:4, 1:4] 0 1.525 0.6...
fulldf    35 obs. of 17 variables
Values
DM        class 'dist' atomic [1:10] 1...
DM2       class 'dist' atomic [1:1] 1...
i         4L
```

dist {stats}   R Documentation

Distance Matrix Computation

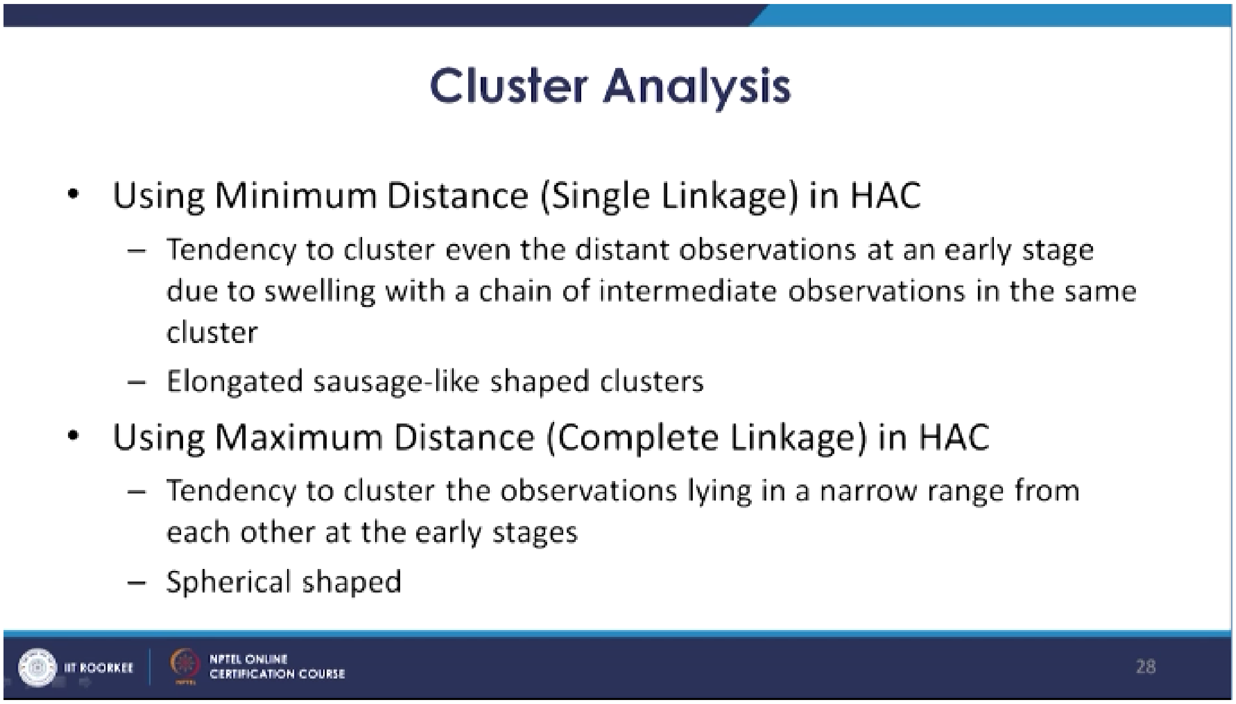
Description

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,
```

the value, so it is actually the distance between CD that is the cluster with single observation CD, and our cluster with 2 observation SKMH-CC which is the closest, right, so the CD you know cluster with single observation, CD is going to be merged with cluster with 2 observations SKMH-CC, so this is going to be the next merger and it will lead us to a 3 x 3 distance metrics, so in this fashion, in this fashion we can keep on following the steps of algorithm and you would see that we'll have the you know full crossing of this clusters, and finally we'll end up with one cluster having all the observations, so this was the exercise that would have given us, the understanding of about how this algorithm is going to work, so let's move ahead.

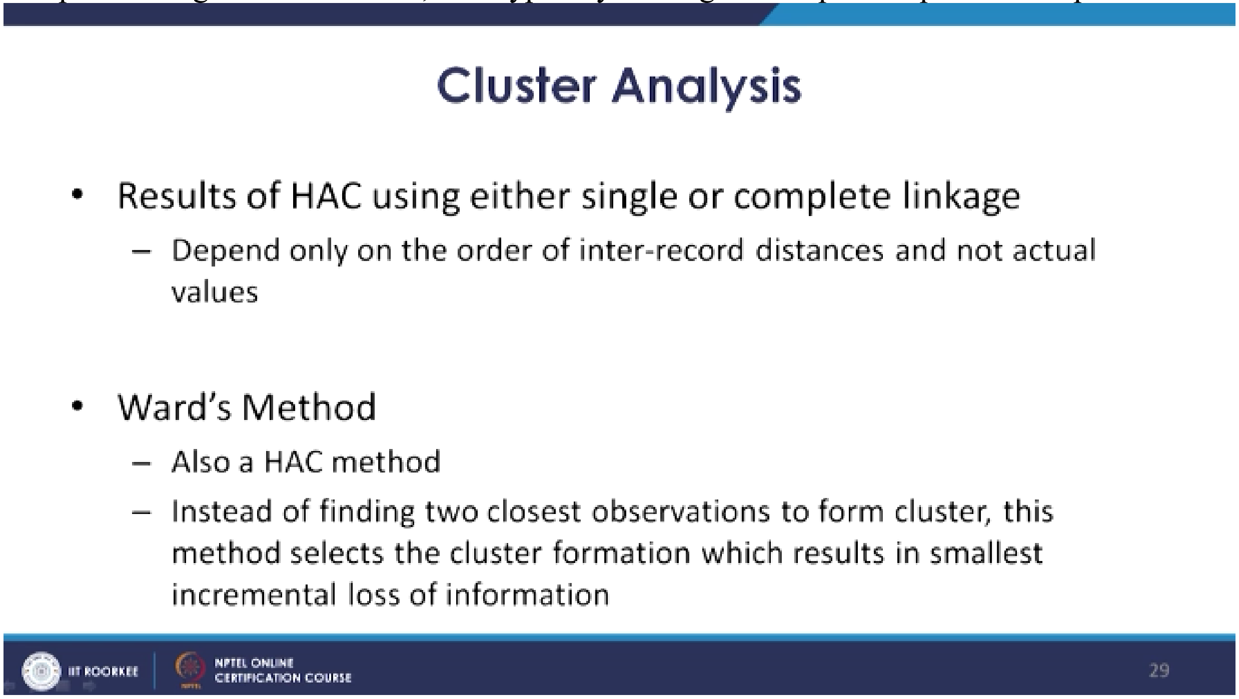


Cluster Analysis

- Using Minimum Distance (Single Linkage) in HAC
  - Tendency to cluster even the distant observations at an early stage due to swelling with a chain of intermediate observations in the same cluster
  - Elongated sausage-like shaped clusters
- Using Maximum Distance (Complete Linkage) in HAC
  - Tendency to cluster the observations lying in a narrow range from each other at the early stages
  - Spherical shaped

So let's talk about the difference distance metric and how the influence the HAC algorithm, so if we are using minimum distance metric or single linkage metrics in HAC, so what are the important points, so let's discuss them, so what happens when we use this minimum distance metric is there is this tendency to cluster even the distant observations at an early stage due to swelling with a chain of intermediate observations in the same clusters, so as we talked about because the observations even the distance observations can be cluster, can be part of the same cluster because of the minimum distance know metric that we are using there, because intermediate distance, intermediate points with you know are going to be there and they will allow this to happen, so what we'll essentially, this will essentially lead to elongate sausage like saved clusters, so as we talked about that the nature of clusters that we have, they are chain like or sausage like then minimum distance metric is one suitable metric, similarly if we are using minimum distance metric in our clustering you know method then also we can end up with these elongated sausage like shaped clusters, so it goes both ways, so far this kind of you know cluster, cluster with this kind of shapes you know minimum distance metric would be you know suitable, and similarly if we are using the minimum distance metric then we might end up with having clusters with this kind of shapes.

Similarly for maximum distance metrics so there is a tendency to cluster the observations lying in a narrow range from each other at the early stages because of the you know that farthest distance but within a you know, farthest distance, because of the distance that we are using is between the farthest observations from each of you know those clusters, so because of this, for this particular you know distance is going to be a narrow range, because it is the closest clusters which are going to be merged, right, so all the clusters and the distance between those clusters even though we would be computing them using maximum distance and we'll you know get the distance values based on the observation which are farthest but when we apply the algorithm it is the you know clusters which are having, which are closest they are going to be merged, so therefore we'll get a narrow range, you know observations which are lying in a narrow range they are probably going to be you know merged together, and therefore the kind of you know shape that we get is typically spherical, so if we are going to use maximum distance or complete linkage metric in HAC, then typically we might end up with spherical shaped clusters.



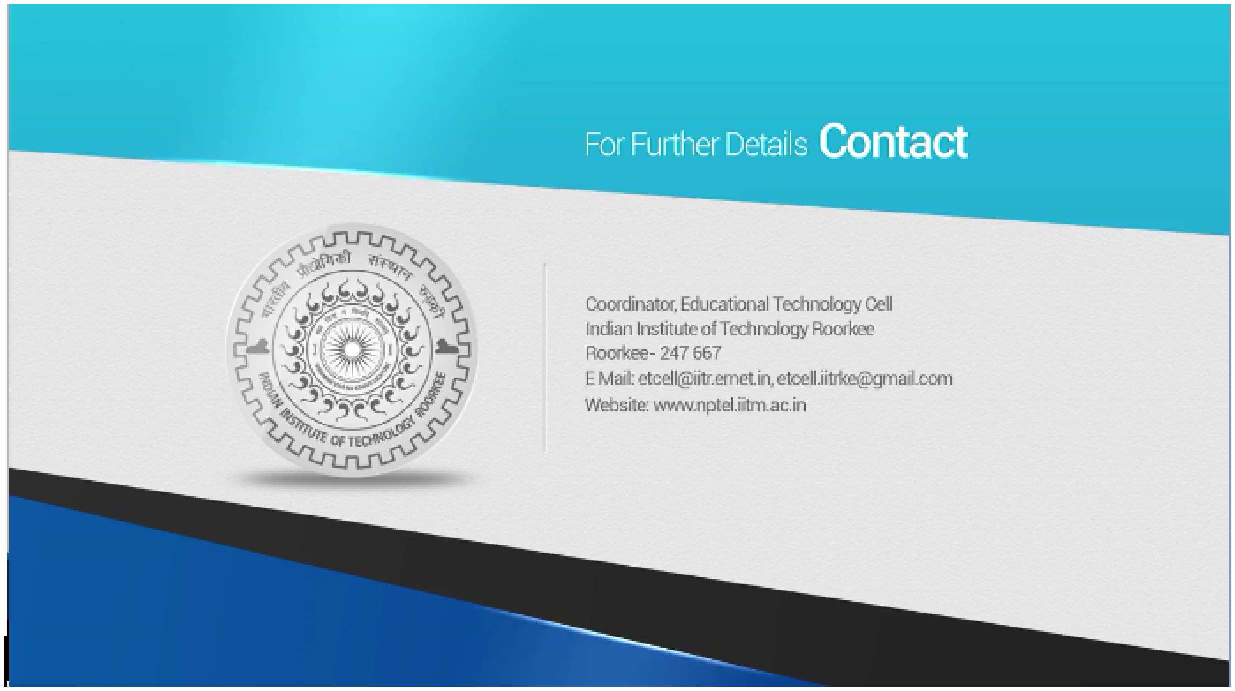
Cluster Analysis

- 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

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE 29

Now you know if we are you know using HAC if you're you know HAC, in HAC in our implementation of HAC, if we are using single or complete linkage then it is important one point that is mentioned here that it is the order of inter record distances that is the whole process depends on, depends on this particular order and not the actual values as you can see here even though the distance competition or you know, you know using you know minimum distance or maximum distance, right, but if we look at how those clusters are joint it is the clusters which are closest, so whether you have, whether you have measured your distance using minimum or maximum distance metric the clusters which are closest they're going to be identified and joint, so therefore it is the order of you know those inter-record distances which are going to matter and not the actual values, so even if you know certain linear transformation or some other kind of transformation happens and distance values change a bit, it might not matter because the order might still remain same, therefore we might end up with this same clusters, so it is important to understand that it is the order, because we need to identify to closest clusters you know, and if the order remains same irrespective of the metric that we are using minimum, maximum then probably we'll end up with same kind of results.

So with this we'll stop our discussion on cluster analysis and in the next class we'll discuss the another important methodagglomerative clustering method that is worse method, so we'll discuss this one in the next lecture. 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. NibeditaBisoyi
Neetesh Kumar
Jitender Kumar
Vivek Kumar
Dharamveer Singh
Gaurav Kumar
An educational Technology cell