

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE  
NPTEL  
NPTEL ONLINE CERTIFICATION COURSE  
Business Analytics & Data Mining Modeling  
Using R – Part II  
Lecture-04  
Association Rules – 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-04 Association Rules-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 been discussing association rules, so let's do a small review of what we have discussed so far, so we started with our discussion on association rules what it is about the different names for this like affinity analysis, market basket analysis, we also talked

## Association Rules

- Also called
  - Affinity Analysis
  - Market Basket Analysis
    - Due to its origin from the studies of customer purchase transactions databases
- Main Idea is
  - To identify item associations in transaction-type databases and
  - Formulate probabilistic association rules for the same
  - “what goes with what”



about the main idea is to find to identify item associations in transaction type databases and formulate probabilistic association rules for the same, so what goes with what, if you know a particular, typically the examples that we have been taking or in the you know marketing domain for you know when the customer is going for purchase in a supermarket or retail store, if a particular item X is being purchased then what else is also being purchased along with that item, so what goes with what kind of you know analysis we typically do in association rules. All these things we have talked about, so we talked about the if then statements that are typically use in association rules, we talked about two stage process, so first stage being rule generation where we apply Apriori algorithm, so we had discussed Apriori algorithm in previous lecture.

## Association Rules

- Association rules
  - “if-then” statements computed from data
  - Example: online recommendation systems or recommender systems in online shopping websites of e-commerce companies like Amazon, Flipkart, and Snapdeal
- Two-stage process
  - Rule generation
    - Apriori Algorithm
  - Assessment of rule strength



We also talked about the rules that are generated in the first stage, how the strength of those rules are assessed, so that we can you know select the most important rules for implementation, so these things have been discussed so far. We have also taken one example for mobile phone

## Association Rules

- Example: mobile phone cover purchase
  - What colors of covers customers are likely to purchase together?
  - Database of ten transactions
  - Open RStudio
- Candidate Rules generation
  - Examine all possible rules between items in “if-then” format
  - Select rules which are most likely to capture the true association



cover, we discussed two transaction data formats in the last lecture item list format, and binary matrix format so those things were also covered, so antecedent, consequent item sets and the concept related to these you know these item sets we have discussed in previous lectures.

## Association Rules

- Apriori Algorithm
  - Generate frequent item sets with one-item sets
    - Compute support for one-item sets
    - Drop the sets having support below user specified minimum support
    - Remaining sets are the frequent one-item sets
  - Recursively generate frequent item sets with two items
    - Use frequent one-item sets to generate two-item sets
      - Since larger size item sets containing non-frequent one-item sets will also be non-frequent item sets
    - Compute support for two-item sets
    - Drop the sets having support below user specified minimum support
    - Remaining sets are the frequent two-item sets



We also talked about the frequent item set, after that you know previous lectures we have also discussed Apriori algorithm, the different steps for it, for example first we start with the generation of frequent item sets having just single item, then this particular list of frequent item

set with single item is taken for generating frequent item sets with two items and so on and so forth frequent item sets with three items, frequent item sets with four items, so in this fashion we keep on going, we also did a small you know generalization where we say that to generate K

## Association Rules

- Apriori Algorithm
  - To generate k-item sets, use frequent (k-1)-item sets
    - Then with three items
    - And so on for all sizes
- Apriori algorithm is quite fast even for a large no. of unique items
  - Each step requires a single run through the database

item sets we use frequent K-1 item sets, we also talked about the Apriori algorithm and the you know lower computational intensity that is required because of you know, in each step we just need to go through once through the database, so all those things we have discussed so far. We talked about few matrix for assessing rule strength, so in terms of rule generation we talked about support and its importance in generating rules, then we talked about these two matrix confidence and lift ratio which could be used to assess the strength of association rules, so we

## Association Rules

- Assessing rule strength
  - Idea is to identify rules which capture strong association between antecedent and consequent item sets
  - Metrics to measure strength of this association as implied by a rule
    - Confidence
      - Ratio of no. of transactions with antecedent and consequent item sets to the no. of transactions with antecedent item set
    - Lift ratio

discussed and defined these two matrix in previous lectures confidence and lift ratio, so we then thought of support and confidence matrix in probability terms, so this part we were able to

## Association Rules

- Revisit Support and Confidence
  - Support as  $P(\text{antecedent and consequent})$
  - Confidence as  $\frac{P(\text{antecedent and consequent})}{P(\text{antecedent})}$   
or  $P(\text{consequent} | \text{antecedent})$
- Typically, high value of confidence means strong association rule
  - It might fail in cases where antecedent and/or consequent item sets have high support leading to high confidence despite no real association

discuss that confidence can be defined as the conditional probability of you know occurrence of consequent item set given that antecedent item sets has occurred, so these things we were able to discuss.

## Association Rules

- Lift ratio
    - Compare the confidence of a rule with its benchmark value
    - Benchmark value of a rule is confidence value computed by assuming no association between antecedent and consequent item sets
      - We assume that antecedent and consequent item sets occur independently  
 $P(\text{antecedent and consequent}) = P(\text{antecedent}) \times P(\text{consequent})$
- Confidence (benchmark) =  $\frac{P(\text{antecedent}) \times P(\text{consequent})}{P(\text{antecedent})} = P(\text{consequent})$
- Benchmark confidence =  $\frac{\text{No. of transactions with consequent item set}}{\text{total no. of transactions}}$

Then while our discussion on lift ratio we talked about the benchmark value, the benchmark confidence value and its importance, how it is calculated so we discussed the how the expressions for benchmark confidence can be you know, can be defined, so you know discussed in the previous lecture that **we talked about** benchmark confidence value can be determined as the probability of occurrence of consequent item sets, so those formulas were also discussed, we defined lift ratio, we also talked about the usefulness of a you know rule, using lift ratio value, so it should be greater than 1, so for any association rule to be useful it's should be, the lift ratio value should be greater than 1.

## Association Rules

- Transaction Data formats
  - Item list format
    - Each row contains a list of purchased items and represents a transaction
  - Binary matrix format
    - Rows represent transactions
    - Columns represent items
    - Cells have either a 1 or a 0 indicating presence or absence of the item in the transaction
  - Open RStudio



In you know exercise in R studio environment we also went through some of the examples where we saw how the rules can be generated, and then how we can assess the strength of the rules, we sorted the list in particular you know lift ratio order for all the rules were sorted using lift ratio orders, so all those things we did, as I talked about transaction data format, item list format and binary matrix format so these things we discussed in previous lectures.

## Association Rules

- Rule Selection
  - Select rules which have confidence higher than user specified cut-off value for confidence
- Open RStudio
- Results Interpretation
  - Support of a rule indicates its impact w.r.t database
    - What proportion of transactions is affected?

We also talked about the results interpretation part and how you know importance of support, confidence and lift value, we talked about that higher value of support and confidence value and

## Association Rules

- Results Interpretation
  - Lift ratio indicates efficiency of rule in finding consequents in comparison to random selection
  - Confidence indicates the rate of finding consequents
- Statistical significance and chance occurrence of rules
  - How sure are we about the meaningfulness of the rules?
  - Are we ending up with associations which are just chance occurrences?

lift ratio are desired for you know selection of you know top association rules and later on implementation, so then we also talked about the statistical significance of these rules, the



chance occurrence so we did an exercise where data was randomly generated and then association rule mining was applied, and we found that even with the randomly generated data we are able to find some strong association rules, so what about the significance? Whether you know these rules can be taken as at phase value or whether we should look for something more, so we discussed a few rules of thumb few points that could be consider, for example more number of transactions for a rule, and you know if there are more number of distinct rules then chance of spuriousness is going to increase, so all those things we talked about.

## Association Rules

- Open RStudio
- Assessing rules for spuriousness due to chance effects
  - More no. of transactions for a rule, less chance of spuriousness
    - Large no. of transactions yield margins of error to a small range occurring due to sampling variations
    - Look to statistical confidence intervals on proportions
  - More no. of distinct rules, higher chance of spuriousness
    - Limit the no. of rules that can be considered from topdown to a no. which can be reasonably incorporated in human decision making process to guard against automated review of rules
- Open RStudio



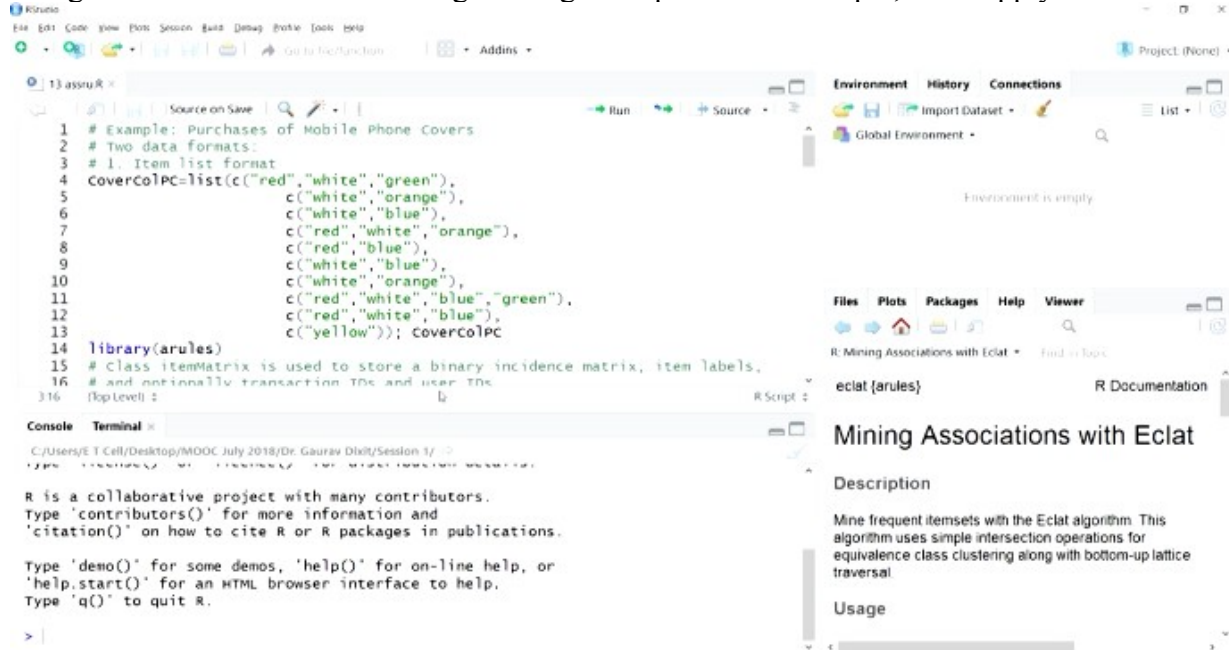
## Association Rules

- Further Comments on Association Rules
  - Transparent and easy to understand method
  - Generates clear and simple rules  
“IF X is purchased THEN Y is also likely to be purchased.”
  - Data-driven modeling process leads to generation of too many rules
    - Non-automated method can be used to select a small set of useful and strong rules
  - Rare combinations of items might be ignored due to high minimum support level specified by user
    - Use items having almost equal frequency
      - Use higher level hierarchies to group items



Before we move on to you know further comments on association rules, we talked about that you know one exercise that we are going to do using a, you know, a particular you know

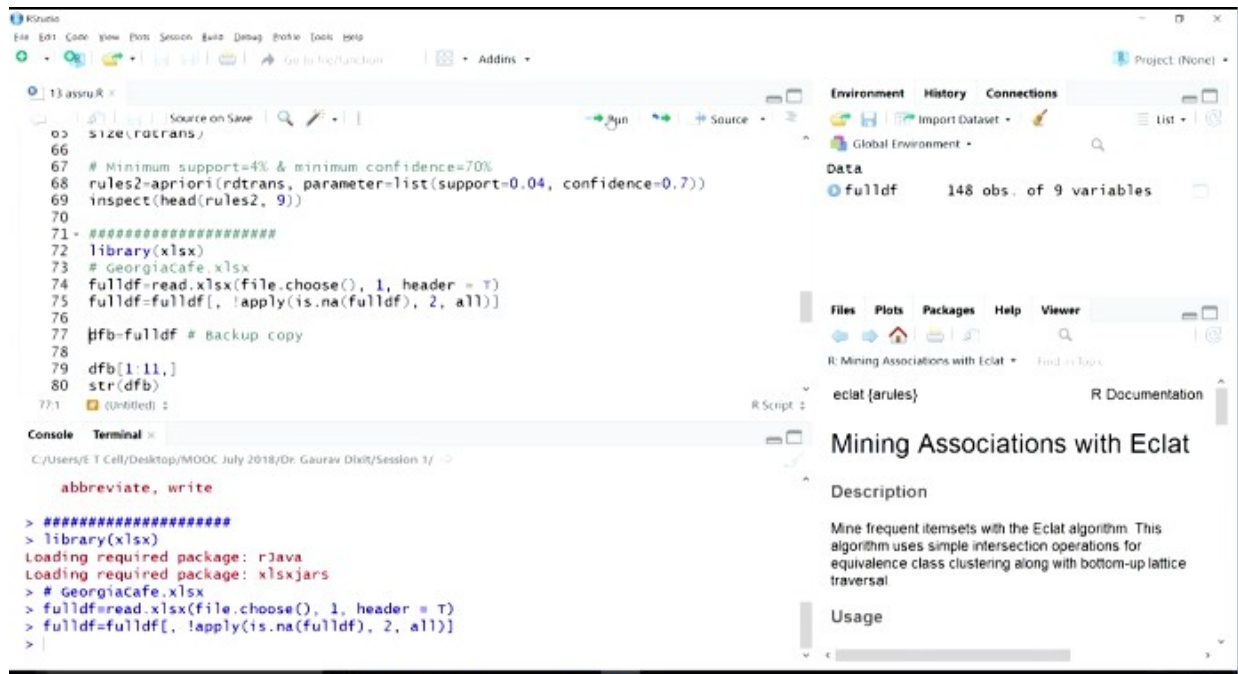
dataset, this dataset is also manufactured, but it is based on a more real life like situation, so we will go back to R studio and we'll go through this particular example, we'll apply association



rules and whatever concepts and exercises that we have done in previous lecture, we'll try to repeat that and discuss in more detail.

So let's scroll through the part which we want to cover, so let's load this library, so A rules is the package that we require to you know develop our models to do our association rules planning in R environment, so once this is done we'll move forward, so in the previous lecture we were trying to load another library, so were facing some problems installing that particular package, so this one, so the problem that we faced was that java was not installed in this particular system and that is why that R java library, though we were able to install it you were not able to load it into R studio environment, so since now we have already installed java in this particular system , now we should be able to load this particular library.

As you can see you know in this session we are able to load this library, now we talked about this Georgia cafe dataset, so let's import this dataset into the R environment, so let's run this code, so we have talked about this particular piece of code in previous code on business analytics and data mining modeling using R, so for more detail you know about these functions and how we are using them here you can always find in those video lectures, of course you have the advantage of help section here in this R studio environment and you can always look for the you know manuals for these functions read.xlsx or any other functions that we are using, so let's run this code and import this particular data, so we can see here Georgia café this, this one, so you can see in the environment section this particular file has been imported into R environment and you can see 148 observations of 9 variables.



So now let's remove any you know, any redundant columns here, so let's run this code. Now let's take a backup of this dataset that we've just imported, so in case you know if we planned to do some modification transformation in some of the variables, then it is always recommended to take back up before proceeding with those transformations.

So now the next code is about the few observations, so let's look at these first few 11 observation of this dataset, so as you can see here first column is transaction number, so this particular dataset is about this Georgia café where customers can buy different you know snacks and beverage items as you can see in the columns itself, tea, coffee, frappe, patties samosa, soft drinks, burgers and chips, so these are some of the items that are being sold by this café and different customers they have purchased different items, so those transactions along with their transaction, number transaction ID's have been recorded, so each row is representing a transaction, so for example first row is transaction number is 561, and the you know items that they have purchased, so you can see they have purchased just the one item that is chips, one packet of it, and the numbers that you see in this particular dataset they are you know representing the number of you know units that they have purchased, right, so but however typically what we require is just the you know presence or absence of that item, so later on as we will see that we would be converting this particular data set into a format of ones and zeros, where one would be representing whether that item was part of that transaction or not, and you know 0 would be you know absence of it, and 1 would be presence of it.

Right now these numbers that you see 3, 2, 1 or 0 they are representing the number of these items that have been purchased, so this also has the information whether they have been purchased or not.

```
71- #####
72 library(xlsx)
73 # Georgiacafe.xlsx
74 fulldf=read.xlsx(file.choose(), 1, header = T)
75 fulldf=fulldf[, !apply(is.na(fulldf), 2, all)]
76
77 dfb=fulldf # Backup copy
78
79 dfb[1,11.]
80 str(dfb)
81
82 df=apply(dfb[,-1], c(1,2), as.logical)
83 head(df)
84 dftrans=as(df, "transactions")
85
```

```
> str(dfb)
'data.frame': 148 obs. of 9 variables:
 $ Txn.no. : num 561 368 668 549 381 456 506 569 607 351 ...
 $ Tea : num 0 0 3 1 3 3 2 0 3 0 ...
 $ coffee : num 0 3 2 1 3 0 0 2 2 2 ...
 $ Frappe : num 0 0 0 0 0 0 0 0 0 0 ...
 $ Patties : num 0 2 0 0 0 3 1 3 0 0 ...
 $ Samosa : num 0 0 1 1 2 0 0 0 1 0 ...
 $ Soft.drinks: num 0 0 0 0 0 0 0 1 0 0 ...
 $ Burgers : num 0 0 0 0 1 0 0 0 1 0 ...
 $ chips : num 1 0 0 0 0 0 0 1 0 1 ...
```

**Mining Associations with Eclat**

**Description**

Mine frequent itemsets with the Eclat algorithm. This algorithm uses simple intersection operations for equivalence class clustering along with bottom-up lattice traversal.

**Usage**

Let's look at the structure of this particular dataset, all the variables that are part of this you know data frame, so if we look at this particular data frame, as I talked about 148 observations, 9 variables, you can see transaction ID tea, coffee, so because all these are number of units that have been purchased, so you can see all these variables are numerical, so we are not required to do any transformation as such here, because as I talked about we would just be requiring ones and zeros.

So right now this particular dataset as you can see is in the numerical format, now we can use the apply function to convert all these values into the logical format, so the values would be true and false, so more detail about this particular function apply you can find out in the help section, so you can see the first argument is the you know data frame on which we want to apply you know this particular function, third argument is the name of the function that we want to apply, on the first argument that is the you know data frame, so as.logical so this function would actually colds the points into you know logical you know variables and you can see the second argument is talking about 1 and 2, 1 representing rows, and 2 representing columns, that means all the cells, on all the cells this particular function is going to be applied, so let's go through this.



```

72 library(xlsx)
73 # GeorgiaCafe.xlsx
74 fulldf=read.xlsx(file.choose(), 1, header = T)
75 fulldf=fulldf[, !apply(is.na(fulldf), 2, all)]
76
77 dfb=fulldf # Backup copy
78
79 dfb[1:11,]
80 str(dfb)
81
82 df=apply(dfb[,-1], c(1,2), as.logical)
83 head(df)
84 dftrans=as(df, "transactions")
85
86 summary(dftrans)
87 inspect(head(dftrans))

```

Console Terminal

```

C:/Users/E T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 1/
> df=apply(dfb[,-1], c(1,2), as.logical)
> head(df)
  Tea Coffee Frappe Patties Samosa Soft.drinks Burgers Chips
[1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
[2,] FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE
[3,] TRUE TRUE FALSE TRUE TRUE FALSE FALSE FALSE
[4,] TRUE TRUE FALSE FALSE TRUE FALSE FALSE FALSE
[5,] TRUE TRUE FALSE FALSE TRUE FALSE TRUE FALSE
[6,] TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE

```

Environment History Connections

Data

- df logi [1:148, 1:8] FALSE FA...
- dfb 148 obs. of 9 variables
- fulldf 148 obs. of 9 variables

Files Plots Packages Help Viewer

R Mining Associations with Eclat Find in Page

eclat (arules) R Documentation

### Mining Associations with Eclat

Description

Mine frequent itemsets with the Eclat algorithm. This algorithm uses simple intersection operations for equivalence class clustering along with bottom-up lattice traversal.

Usage

We are doing this, because we are interested in just the presence or absence of a particular item in a particular transaction, so let's run this, and let's look at the first 6 observations here, now you can see we have just the you know items that are available, so all the items are distinct, 1, 2, 3, 4, 5, 6, 7, 8, so we have 8 items, 8 distinct items and whether they are present in a particular transaction or absent that is being represented using these you know values, true or false.

```

76
77 dfb=fulldf # Backup copy
78
79 dfb[1:11,]
80 str(dfb)
81
82 df=apply(dfb[,-1], c(1,2), as.logical)
83 head(df)
84 dftrans=as(df, "transactions")
85
86 summary(dftrans)
87 inspect(head(dftrans))
88
89 # Minimum support=20 & minimum confidence=50%
90 rules3=apriori(dftrans, parameter=list(support=20/148, confidence=0.5))
8710

```

Console Terminal

```

C:/Users/E T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 1/
most frequent items:
  Tea Patties Samosa Coffee Frappe (Other)
  116   100   97    90    26    46

element (itemset/transaction) length distribution:
sizes
 1  2  3  4  5  6
6 29 56 43 13  1

  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 1.000  3.000   3.000   3.209  4.000   6.000

```

Environment History Connections

Data

- df logi [1:148, 1:8] FALSE FA...
- dfb 148 obs. of 9 variables
- dftrans Formal class transactions
- fulldf 148 obs. of 9 variables

Files Plots Packages Help Viewer

R Mining Associations with Eclat Find in Page

eclat (arules) R Documentation

### Mining Associations with Eclat

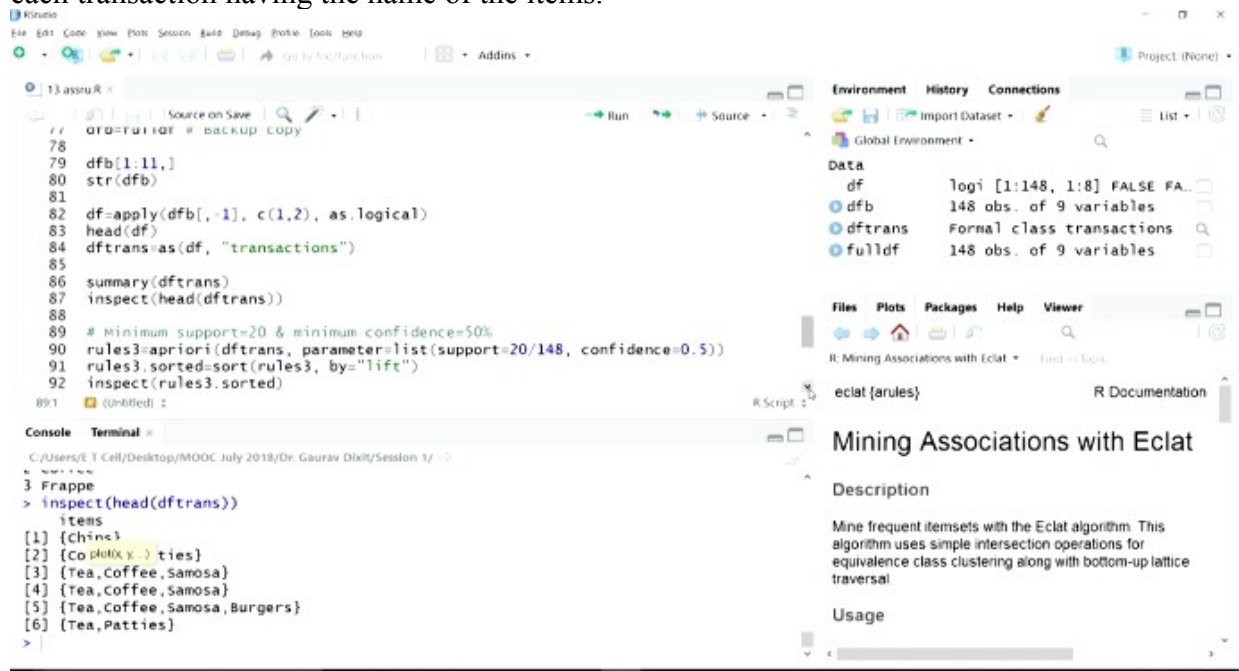
Description

Mine frequent itemsets with the Eclat algorithm. This algorithm uses simple intersection operations for equivalence class clustering along with bottom-up lattice traversal.

Usage

Now we can always convert this particular data frame into transactions format, so for this we can use as a function, so let's run this and we'll get to convert this particular dataset into transactions format. Now this is done, we can look at the summary of this, so you can see here, so the few details about this particular you know format is available, you can see most frequent items, you can see tea is the most frequent item followed by patties, samosa and all those, other

all other things, so as we talked about in previous lecture inspect is the function that could be used to look at the, look at the transactions database, so you can see. Now this format is the, you know item list format where each transaction is representing the name of the items that is there, so you can see in this we have just first 6 transaction, so first transaction is having just chips, second transaction is having coffee and patties, third transaction we have tea, coffee, samosa, so each of these transaction now, what we are seeing right now is the item list format, each transaction having the name of the items.



Now to proceed with our association rules mining, what we will do is we'll call this Apriori function, but before calling this function we need to establish our you know support level, minimum support level and minimum confidence level, so as you can see in the commented line I was specified support as 20 that is the number of transaction and the minimum confidence as 50 percentage, that is the percentage minimum accepted confidence value in percentage terms.

So now you can see in the next line I'm calling the function Apriori and I'm passing on the first argument, this transaction database that we have just created, then the parameter I'm specifying the you know, the minimum support level that is 20 divided by 148 and the confidence value, so you can see these values are in proportion terms, so appropriately they have been specified, so let's call this function.

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

```

80  str(df)
81
82  df=apply(dfb[, -1], c(1,2), as.logical)
83  head(df)
84  dftrans=as(df, "transactions")
85
86  summary(dftrans)
87  inspect(head(dftrans))
88
89  # Minimum support=20 & minimum confidence=50%
90  rules3=apriori(dftrans, parameter=list(support=20/148, confidence=0.5))
91  rules3.sorted=sort(rules3, by="lift")
92  inspect(rules3.sorted)
93
94  # Find redundant rules
95  subset3=is.subset(rules3.sorted)
96
97  (untitled)

```

The console output shows the execution of the Apriori algorithm:

```

C:/Users/E T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 1/ >
set transactions ... [8 item(s), 148 transaction(s)] done [0.00s].
sorting and recoding items ... [6 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [34 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
warning message:
In apriori(dftrans, parameter = list(support = 20/148, confidence = 0.5)) :
Mining stopped (maxlen reached). Only patterns up to a length of 4 returned!

```

The Environment pane on the right shows the following objects:

- df: logi [1:148, 1:8] FALSE FA...
- dfb: 148 obs. of 9 variables
- dftrans: Formal class transactions
- fulldf: 148 obs. of 9 variables
- rules3: Formal class rules

The Files pane shows the R script 'eclat {rules}' and R Documentation for 'Mining Associations with Eclat'.

Now once this is done we can always sort this particular you know output of Apriori function into you know, by lift values so let's run this, next line. And now we can use the inspect function to look at the details of the output, so now as you can see there are number of association rules have been displayed in this output, and you can see that all these association rules have been sorted in decreasing order of the lift value, so the first association rule is having the highest lift value, as you can see the lift value is 1.17 so from this database that we have Georgia café database that we have, we are not getting you know many assessment rules with you know higher lift values, so even the top association rules are having lift values around you know 1, and this is the only the first transaction is having just you know a bit higher lift value that is also not much high, it is just 1.17, so you can see that in this first transaction we have you know frappe and if the frappe is being purchased then samosa is also being purchased, so this is you know the first, the strongest association rule that we have from this database.



```

82 df=apply(df[,1:9], M(1,2), as.logical)
83 head(df)
84 dftrans=as(df, "transactions")
85
86 summary(dftrans)
87 inspect(head(dftrans))
88
89 # Minimum support=20 & minimum confidence=50%
90 rules3=apriori(dftrans, parameter=list(support=20/148, confidence=0.5))
91 rules3.sorted=sort(rules3, by="lift")
92 inspect(rules3.sorted)
93
94 # Find redundant rules
95 subset3=is.subset(rules3.sorted)
96 subset3[lower.tri(subset3, diag=T)]=NA
97 rdd3=colSums(subset3, na.rm=T) >= 1
94:1 (Unfiled) :

```

```

> rules3.sorted=sort(rules3, by="lift")
> inspect(rules3.sorted)
      lhs      rhs support confidence lift count
[1] {Frappe} => {Samosa} 0.1351351 0.7692308 1.1736717 20
[2] {Coffee,Samosa} => {Tea} 0.3175676 0.8245614 1.0520266 47
[3] {Frappe} => {Tea} 0.1418919 0.8076923 1.0305040 21
[4] {Tea,Samosa} => {Coffee} 0.3175676 0.6184211 1.0169591 47
[5] {Tea,Coffee} => {Samosa} 0.3175676 0.6619718 1.0100189 47
[6] {Coffee} => {Tea} 0.4797297 0.7888889 1.0065134 71
[7] {Tea} => {Coffee} 0.4797297 0.6120690 1.0065134 71

```

**Environment** History Connections

Global Environment

Data

- df logi [1:148, 1:8] FALSE FA...
- dfb 148 obs. of 9 variables
- dftrans Formal class transactions
- fulldf 148 obs. of 9 variables
- rules3 Formal class rules

**Files** Plots Packages Help Viewer

R Mining Associations with Eclat

R Script : eclat (arules) R Documentation

### Mining Associations with Eclat

**Description**

Mine frequent itemsets with the Eclat algorithm. This algorithm uses simple intersection operations for equivalence class clustering along with bottom-up lattice traversal.

**Usage**

So as you can see is you know, frappe and samosa is being purchased, this is not something you know that we would ideally expect as to be a strong association rule, so all this is because of the manufactured dataset and so the real relationships might not anyway be captured. However, for an exercise we can see here other assessment rules so in this you would also see that some association rules are involving the same items, for example you would see that association rule number 4 and 5, tea, samosa, if tea and samosa are being purchased then coffee is being purchased, so the lift value is higher than 1 just above 1, and in the transaction, in the association rule number 5 you see that if tea and coffee are being purchased then samosa is also being purchased, so this is also having lift value of just above 1, so you would see that you know on the same items, you know are being used, so this is not unusual, so you know rather this is more favorable thing because you'll have to deal with, you can have more number of association rules and still it will be the you know, fewer number of items, so we can have this kind of situation where this one item set is leading to more than 1 association rules.

```

82 df=apply(df[,1:4], c(1,2), as.logical)
83 head(df)
84 dftrans=as(df, "transactions")
85
86 summary(dftrans)
87 inspect(head(dftrans))
88
89 # Minimum support=20 & minimum confidence=50%
90 rules3=apriori(dftrans, parameter=list(support=20/148, confidence=0.5))
91 rules3.sorted=sort(rules3, by="lift")
92 inspect(rules3.sorted)
93
94 # Find redundant rules
95 subset3=is.subset(rules3.sorted)
96 subset3[lower.tri(subset3, diag=T)]=NA
97 rdd3=colSums(subset3, na.rm=T) >= 1
98
99 (Untitled) :

```

```

C:/Users/E T Cell/Desktop/MOOC July 2018/Dr. Gaurav Dixit/Session 1/
In apriori(dftrans, parameter = list(support = 20/148, confidence = 0.5)) :
Mining stopped (maxlen reached). Only patterns up to a length of 4 returned!
> rules3.sorted=sort(rules3, by="lift")
> inspect(rules3.sorted)

```

lhs	rhs	support	confidence	lift	count
[1] {Frappe}	=> {Samosa}	0.1351351	0.7692308	1.1736717	20
[2] {coffee,Samosa}	=> {Tea}	0.3175676	0.8245614	1.0520266	47
[3] {Frappe}	=> {Tea}	0.1418919	0.8076923	1.0305040	21
[4] {Tea,Samosa}	=> {Coffee}	0.3175676	0.6184211	1.0169591	47
[5] {Tea,Coffee}	=> {Samosa}	0.3175676	0.6619718	1.0100189	47
[6] {coffee}	=> {Tea}	0.4797297	0.7888889	1.0065134	71

So other things also you can look at, so since this output has been sorted using lift values, so strongest you know association rules will come on top, so in terms of, as we talked about in the previous lectures we should always you know select the rules which can be you know reasonably incorporated in human decision making process, so probably if you have to select the rules from this output, the rules which can be implemented for their business and operational you know applicability, we will have to select few rules from the top, right, so if we look at the some of the rules from the top, then we see that as we talked about that we would prefer, all these values support, confidence and lift values to be on the higher side. So we also talked about that we took a wide spuriousness to get more meaningful rules, we would like to have support count, support that is count, more number of transactions supporting particular rule, so we look at the first association rules the count is 20, however it is having higher lift value quite good you know high value of confidence, but slightly lower support, but if we look at the you know, if we look at the transaction association rule number 2 that is if coffee and samosa are being purchased then tea is also being purchased, so there we look at that 47 transactions, 47 you know transactions are supporting this particular rules, so in total we have 148 transactions and out of that about you know 31% of the transactions are supporting this rule, and we are also having the second highest lift value, we are also having a very high confidence value, it might be the highest also, if we are able to look at the you know, the full table, and sorted it out in confidence values, and then we can find out, so high confidence value, high support for the second association rules and of course you know high count, high lift value, relatively high lift value and higher you know confidence and support value, so from this we can see it is the second association rule that we might be willing to invest a bit more implement.

```

82 df=apply(DFD[, -1], C(1,2), as.logical)
83 head(df)
84 dftrans=as(df, "transactions")
85
86 summary(dftrans)
87 inspect(head(dftrans))
88
89 # Minimum support=20 & minimum confidence=50%
90 rules3=apriori(dftrans, parameter=list(support=20/148, confidence=0.5))
91 rules3.sorted=sort(rules3, by="lift")
92 inspect(rules3.sorted)
93
94 # Find redundant rules
95 subset3=is.subset(rules3.sorted)
96 subset3[lower.tri(subset3, diag=T)]=NA
97 rdd3=colSums(subset3, na.rm=T) >= 1
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Console Output:

```

Mining stopped (maxlen reached), only patterns up to a length of 4 returned!
> rules3.sorted=sort(rules3, by="lift")
> inspect(rules3.sorted)
  lhs                rhs      support confidence lift  count
[1] {Frappe}          => {Samosa} 0.1351351 0.7692308 1.1736717 20
[2] {Coffee,Samosa} => {Tea}   0.3175676 0.8245614 1.0520266 47
[3] {Frappe}          => {Tea}   0.1418919 0.8076923 1.0305040 21
[4] {Tea,Samosa}     => {Coffee} 0.3175676 0.6184211 1.0169591 47
[5] {Tea,Coffee}     => {Samosa} 0.3175676 0.6619718 1.0100189 47
[6] {Coffee}          => {Tea}   0.4797297 0.7888889 1.0065134 71
[7] {Coffee}          => {Tea}   0.4797297 0.6120690 1.0065134 71

```

Now if we go down this table we see that there are few association rules like 6, 7, and 8, 9, so and even if we go down further we see that support level is increasing around this, so we see that association rule number 14, if samosa is being purchased then tea is also being purchased, not this one just above this, tea is being purchase, so you can see most of the transactions involve that tea is being purchased, so 116, patties 100 transactions, but if we are not interested in single item association rules then probably, then probably this transaction if samosa is being purchased then tea is being purchased, 76 transactions are supporting this association rules, however lift value is lower, so probably lift value is less than 1, so probably we'll not go with this, so we'll have to stick to the association rules which are having lift value higher than 1, so if we take that as the criteria, then we will have to consider the association rules up to 9, so association rules from 1 to 9 are to be considered if P follow this criteria that lift value has to be higher than 1, and within this if we see that rule number 6 and 7 they are having the highest support, right, they're having highest count 71 and 71, and we can see you know higher confidence value, high support value, higher count and also the lift value is greater than 1, so probably association rule number 6 and 7, these are the rules that we might be interested into implementing first, right, even though they do not have the highest lift value.

So if we look at these association rules, if coffee is being purchased then tea is being purchased, and if tea is being purchased then coffee is being purchased, if we look at this, so then essentially you know in terms of implementation we just get at the one decision rule, so you know this coffee and tea they are being purchased together so this particular information we can use in our business and operational you know implementation in our decisions.

After this we can look at the rule number you know 4 and 5, rather we'll prefer the rule number 2 for it being the having the higher lift value, so after rule number 6 and 7 we'll probably go with rule number 2, that if coffee and samosa are being purchased then tea is also being purchased, and then after that if we still are looking for more rules to implement then probably 4 and 5, however again you can see 4 and 5 effectively they also converge to the same thing, so these rules also, you know, these rules involve just three items, tea, samosa and coffee, so if we are looking for just top three rules that we would like to implement, then first one would be

based on rule number 6 and 7, tea and coffee that association then tea, samosa and coffee that association would be third, the second one would be you know coffee, samosa, and tea, so if we look at the rule number 2 is also involving the same item, so essentially top things that we would like to implement from this output would be, first thing would be tea and coffee, then the second thing would be tea, coffee, and samosa, so based on this output as you can see we can restrict ourselves to you know some fewer number of rules, some you know few rules which are you know, you know satisfying all the criteria that we want, higher transaction support, higher number of transaction supporting them, higher support value, higher confidence value, and higher lift value, and then some of these you know rules will involve the same items, so using all this we would be able to get some you know few implementable you know information, implementable associations, It mean items, and move ahead.

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

```

0.2
86 summary(dftrans)
87 inspect(head(dftrans))
88
89 # Minimum support=20 & minimum confidence=50%
90 rules3=apriori(dftrans, parameter=list(support=20/148, confidence=0.5))
91 rules3.sorted=sort(rules3, by="lift")
92 inspect(rules3.sorted)
93
94 # Find redundant rules
95 subset3=is.subset(rules3.sorted)
96 subset3[lower.tri(subset3, diag=T)]=NA
97 rdd3=colsums(subset3, na.rm=T) >= 1
98 which(rdd3)
99 # Prune them
100 rules3.pruned=rules3.sorted[!rdd3]
94.1 (Untitled) 1

```

The console output shows the following table of association rules:

lhs	rhs	support	confidence	lift	count
[1] {Frappe}	=> {Samosa}	0.1351351	0.7692308	1.1736717	20
[2] {Coffee,Samosa}	=> {Tea}	0.3175676	0.8245614	1.0520266	47
[3] {Frappe}	=> {Tea}	0.1418919	0.8076923	1.0305040	21
[4] {Tea,Samosa}	=> {Coffee}	0.3175676	0.6184211	1.0169591	47
[5] {Tea,Coffee}	=> {Samosa}	0.3175676	0.6619718	1.0100189	47
[6] {Coffee}	=> {Tea}	0.4797297	0.7888889	1.0065134	71
[7] {Tea}	=> {Coffee}	0.4797297	0.6120690	1.0065134	71
[8] {Coffee}	=> {Patties}	0.4121622	0.6777778	1.0031111	61
[9] {Patties}	=> {Coffee}	0.4121622	0.6100000	1.0031111	61
[10] {}	=> {Coffee}	0.6081081	0.6081081	1.0000000	85

So let's move forward, so some of these rules as we have seen that, some of these rules might be redundant, so some of this rules might be subset of some other rules, so how do we eliminate these redundant rules which are subset of some other rule, right, so we have some code for this, so you can see we are calling here is.subset function, so this function can be used to identify the you know item sets which are subset of some other set, so let's run this code.

Now in the next line as you can see that because we'll get a you know so lower triangular values and upper triangular value would be same, so we would like to you know strike you know lower triangular using any values, the second line is doing precisely that, so let's run this. Then as you can see we are using colsums function, so it will sum all the you know subsets, all the item sets which are subset of 1 or more than 1 set, so all those you know item sets would be identified, so let's run this, so we can also, from here we can also see which of these you know subsets or which of these item sets are you know subsets of other item sets, so we can see big list is here, so we can always pruned this list and the remaining, we'll just have the remaining items which are distinct, so we run this, let's look at this, let's call this inspect and so right now probably you know, there was, in this particular run, there, we were not able to find some unique items, otherwise we would have got an output where some remaining subset would be there.



The screenshot shows an RStudio session with the following components:

- Source Editor:** Contains R code for pruning rules and visualizing them.
 

```

91 rules3.sorted=sort(rules3, by="lift")
92 inspect(rules3.sorted)
93
94 # Find redundant rules
95 subset3=is.subset(rules3.sorted)
96 subset3[lower.tri(subset3, diag=T)]=NA
97 rdd3=colSums(subset3, na.rm=T) >= 1
98 which(rdd3)
99 # Prune them
100 rules3.pruned=rules3.sorted[!rdd3]
101 inspect(rules3.pruned)
102
103 library(arulesviz)
104 plot(rules3.pruned)
105 plot(rules3.pruned, method="graph", control=list(type="items"))
106
103:14 (Untitled)

```
- Console:** Shows the output of the R code, displaying association rules with their support and confidence values.
 

```

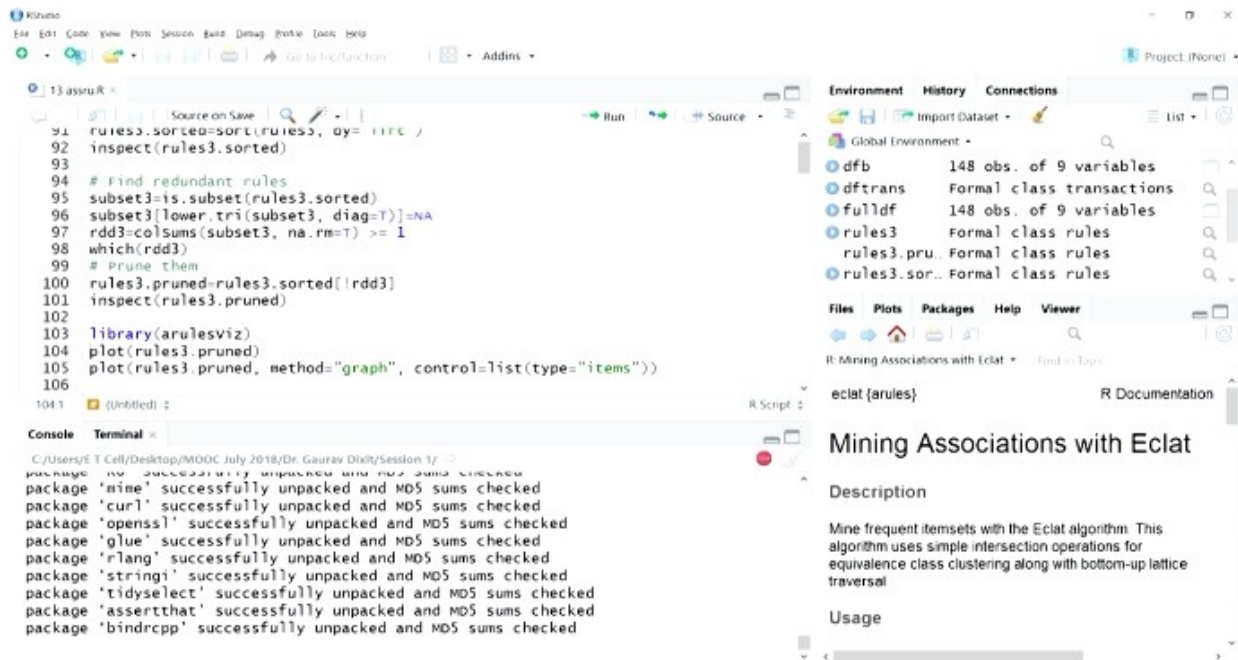
{Tea,Patties,Samosa} {Coffee,Patties,Samosa}
29 30
{Tea,Coffee,Patties,Samosa} {Tea,Patties,Samosa}
31 32
{Coffee,Patties,Samosa} {Tea,Coffee,Patties,Samosa}
33 34
> # Prune them
> rules3.pruned=rules3.sorted[!rdd3]
> inspect(rules3.pruned)
>

```
- Environment:** Lists loaded objects: dfb (148 obs. of 9 variables), dftrans (Formal class transactions), fulldf (148 obs. of 9 variables), rules3 (Formal class rules), rules3.pru. (Formal class rules), and rules3.sor.. (Formal class rules).
- Files Plots Packages Help Viewer:** Shows the 'arulesviz' package documentation, including a description of the Eclat algorithm and its usage.

So then there is once these remaining rules are there then we can use this particular library *arulesviz*, so this is only for visualization purpose, so the rules they can be you know a graphic can be created which can, convey the information about these shortlisted rules in terms of you know the different shape of you know different shapes that are being used, their size of those shape and color of those shapes can be used to convey different kind of information, so for this we would be requiring this library, so let's try and load this, so this is not installed, so let's run this.

So we will have, we'll try and install this particular library `install.packages("arulesviz")`, so within double quotes we will like to enter the name of it, so once this particular library is installed and then we can load it and you'll see in this run we have not been able to shortlist rules, we do not have an real output in this particular rules, otherwise we would be able to see that shortlist rule, they can be depicted using certain shapes and size of those shapes and color can convey certain information, right, so for example size you can convey the number of transaction that are involved, and the color can convey the you know lift value, so support, confidence and lift value can be you know depicted using some of these you know graphics and different visualization, so essentially it would be a multi-dimensional graphic.

So what we can do is, we can do one more run of this particular, you know we can apply association rules one more time and if some output is after pruning we get few rules shortlisted rule, then we can of course visualize them using these functions.



So right now this package is still being installed, so this is about to be completed I think this package installation is almost complete. Just about there the packages are being unpacked and installed, so it seems to be stuck somewhere, so as I talked about that, once this library is loaded and we have shortlisted some rules they can be visually seen the importance of those rules, their strength using different matrix that we have talked about, support, confidence, and lift value, so using shapes and colors we can create a multidimensional graphic using this particular code and we would be able to see it, so we'll stop here and we'll start our discussion on cluster, analysis in the next lecture. Thank you.

The slide has a blue header with the text "For Further Details Contact". Below the header is the IIT Roorkee logo, which is a circular emblem with the text "INDIAN INSTITUTE OF TECHNOLOGY ROORKEE" and "संस्कृतम्" and "सर्वज्ञानं सर्वभूतानां" in Sanskrit. To the right of the logo, the contact information is listed: "Coordinator, Educational Technology Cell, Indian Institute of Technology Roorkee, Roorkee- 247 667, E Mail: etcell@iitr.emet.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](mailto:etcell@iitr.ernet.in), [iitrke@gmail.com](mailto:iitrke@gmail.com)  
Website: [www.nptel.iitm.ac.in](http://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